Department of Computer Science and Engineering
The Chinese University of Hong Kong

Mobile Application Using NFC

LYU1301 Final Year Project Report (Spring)

Chan Sin Yee 1155006665

Supervised by Michael R. Lyu

NFC gives opportunities to connect people between virtual world and physical world. In this project, we do research on NFC and made use of this powerful technology to build an attractive Android app. With the idea of social network, our app, Latia, is finally implemented.

LYU1301
Mobile Application Using NFC

Contents

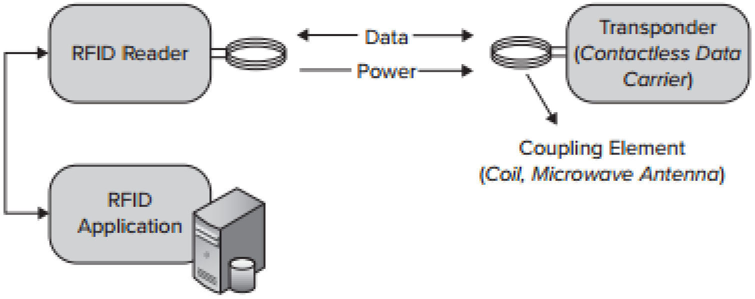
1 INTRODUCTION

1.1 Background

Near Field Communication (NFC) is one of the enabler for ubiquitous computing. This is a new technology begins to deploy on new smartphones. Users can use NFC to pass data wirelessly through devices or between device and a tag by touching them together. Since the design of NFC consists of security, the NFC enabled mobile devices are good replacement of credit cards, tickets, car keys and hotel room access card. It also simplifies the interactions of users with intelligent environment. The technology that NFC used is based on Radio Frequency Identification (RFID), but focus on consumer smartphone.

1.1.1 Radio Frequency Identification

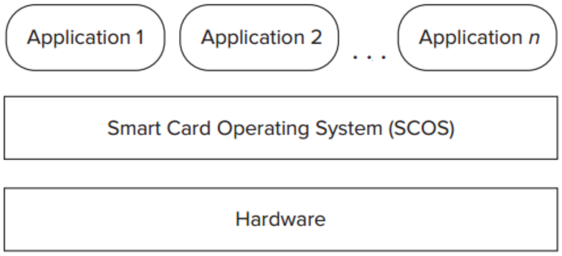
Radio Frequency Identification (RFID) is a wireless communication technology for identification and tracking. The data transmission results from electromagnetic waves, which can have different ranges (20 – 400 cm)


RFID Reader
Data
Power
Transponder (Contactless Data Carrier)
Coupling Element (Coil, Microwave Antenna)
RFID Application

The RFID application is assigned specific information after transmission then the application can do some operation. A typical example is the Octopus system in MTR station. Passenger uses his Octopus card to validate e-money using RFID, the RFID application then receives the data inside the Octopus card. If the card contains enough money, then unlock the gate otherwise keep locking the gate. The RFID technology is being used all over the world for different applications. Including inventory system, Human implants, Animal Identification, Casino Chip-tracking, etc. The main defect of RFID is that the tag is so passive such that people who own a tag cannot read the data inside directly. Presumably, normal people do not own an RFID reader. For example, passengers cannot check their e-money inside their Octopus card at home; they have to go to the MTR station to check it using specific machine or device.

1.1.2 Smart Card Technology

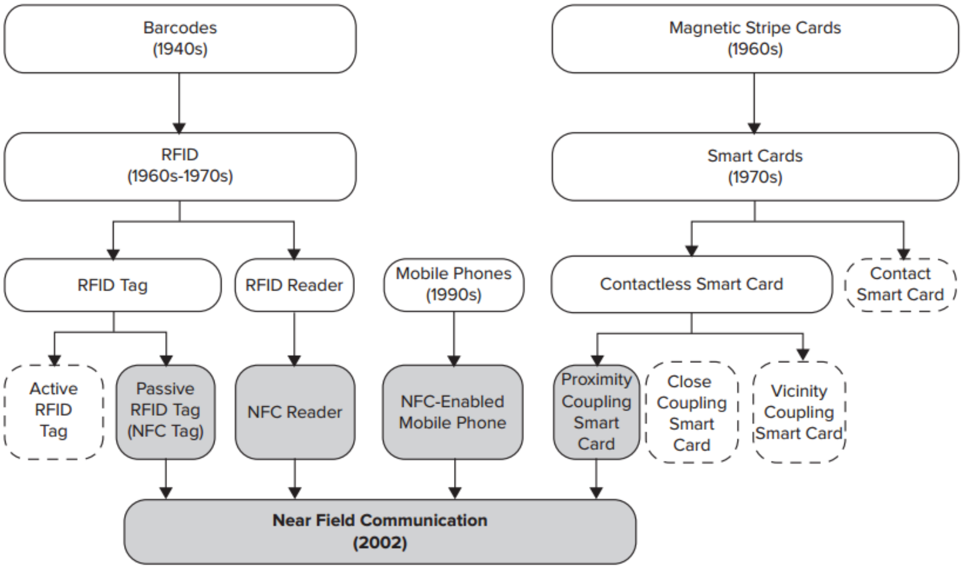
Smart card is similar to a tag but it contains an embedded microprocessor to enhance the secure ability. In order to generalize the usage of smart card, the microprocessor may have installed a Smart Card Operating System (SCOS). Upon the operating system, applications can then be installed.



However, the communication of smart card relies on two methods, physical contact through the conductive contact plate on the surface of the smart card and magnetic strip. Examples are EPS, HKID Card and CULink. Although it has high security, it requires user to insert or slide the card into a machine or device which may not be so convenient. Besides, likewise, user cannot read the data inside the tag unless suitable public machine is nearby.

1.1.3 Near Field Communication

NFC integrates the technology of RFID and Smart Card putting it into a mobile phone. It standardized for consumer smartphone. Since mobile phone has a monitor, user can read the data from a tag through the phone in anywhere. With an NFC-enabled mobile phone, user can even turn the phone to an active tag containing personal information, credit cards and set of keys. Moreover, because of the mobility of NFC device, devices can communicate using peer to peer mode, it, on the other hand, enhance the interactions of mobile phone usage.



1.2 Motivation

Living in this era, we cannot imagine what our lives will become if mobile phone disappears. Originally, mobile phone was used for voice communication between two people apart from each other. Yet, mobile phone evolves so quickly such that it becomes a mobile computer today. Now, people not only use phones to make phone call, they use their phone for searching, gaming, socializing and messaging. Since the mobile phones attach to them everywhere, the power of the phone becomes the power of them as long as the battery is charged. This special power makes them easy to work on anything; they can plan a party by just several clicks. The bonds between people should become increasingly stronger, but is this happening in reality?

We observe that many people get addicted on their phones. In a study of 1,600 managers and professionals, Professor Perlow found that 58% of smartphone users don't go 1 hour without checking their phones [1]. People stuck in the world of smartphone, communicating through the screen. In our experience, people use their phone during transportation, dining or even walking. They become so immersed in the virtual world that they end up not paying attention to the real world, people and conversation around them. This could harm the relationship between people.

To create a connection between the virtual world and the physical world we live in. Near Field Communication (NFC) might be the technology that makes this vision comes true. In today’s smartphones, manufacturers are starting to immerse the NFC technology into new phones, e.g. BlackBerry Bold 9790(2011) Nexus 7(2013). Increasingly many people own a smartphone with an NFC service, which leaves a question: What can developers make use of this treasure? People are required to go to a specific location in order to use NFC; this is the key to connect people to the real world.

We believe NFC can saves people from the immersive virtual world by creating a unique culture. We believe that if we can create a popular app that encourages people to use NFC, people will pay more attention to the physical world.

1.3 Project Objective

The goal of our project is to write an attractive Android application using NFC service. We have set the following objectives for our app to achieve the goal:

The app should

- Be Interactive with the physical world using NFC. The beauty of NFC is the connection between virtual world and the reality; we have to focus on the function that NFC brought to the app.
- Be Interesting. The idea has to be interesting which makes user to share with their friends, and finally addicted to it.
- Contains user-generated content (UGC). UGC helps the app to grow by users themselves, a fruitful content attracts new users and keep the old users alive.
- Has well and fluent user-experience. User may feel inconvenient if it has bad user-experience, which may stop user to continue using the app.

2 Studies On NFC

NFC is a set of standard for smartphones based on RFID technology created by Philip and Sony. Standards include Near Field Communication Interface and Protocol (NFCIP-1/2), Near Field Communication Wired Interface (NFC-WI), Front-End Configuration Command for NFC-WI, NFC Data Exchange Format (NDEF), etc. NFC enabled devices can establish data transmission by touching 2 NFC enabled devices together or by touching a tag to the device over a few centimeters.

Since NFC is inherited by RFID, the operating frequency is the subset of that of RFID which makes NFC-enabled mobile device supports some of the RFID tag. Besides, NFC is invented partially for replacing credit cards and access keys, security is a major issue. Although RFID supports operating range up to 4 meters, NFC restricts to only 0-4 centimeters which greatly enhance the security.

2.1 Comparison with Bluetooth

In the spectrum of wireless communication, we categorize them according to their operating range. There are wireless wide area network (WWAN), wireless local area network (WLAN) and wireless personal area network (WPAN). For the WPAN technologies, Bluetooth is one of them and is also available in mobile phones. Since NFC and Bluetooth are both short-range communication technologies and integrated into mobile phones. We can give a simple comparison between them.

Aspect	NFC	Bluetooth
RFID compatible	ISO 18000-3	active
Standardization body	ISO/IEC	Bluetooth SIG
Network Standard	ISO 13157 etc.	IEEE 802.15.1
Network Type	Point-to-point	WPAN
Cryptography	not with RFID	Available
Range	< 0.2 m	~100 m (class 1)
Frequency	13.56 MHz	2.4–2.5 GHz
Bit rate	424 kbit/s	2.1 Mbit/s
Set-up time	< 0.1 s	< 6 s
Power consumption	< 15mA (read)	varies with class

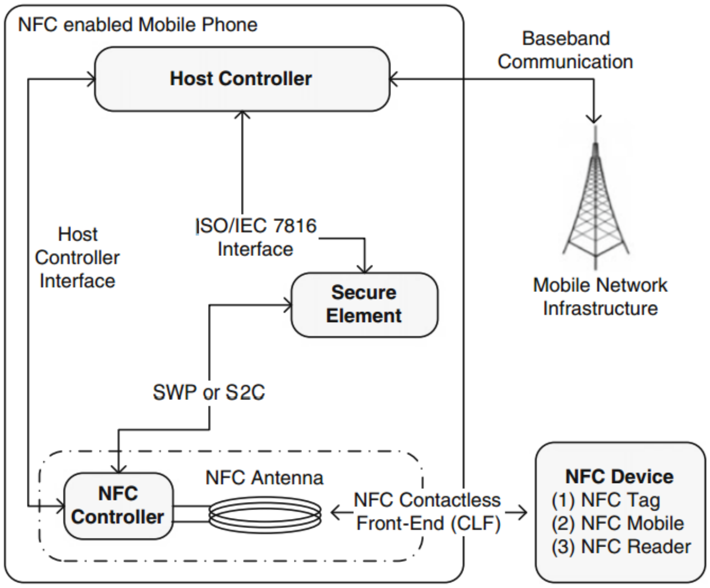
The operating range of NFC is so short, which reduces the likelihood of unwanted interception during any transmission. Yet, NFC sets up in less than 0.1 seconds, the short handshaking time allows devices the begin transmission just after the touch.

Compare with Bluetooth, first requires searching for targets, after selection then handshaking with authentication. In practice, it takes up more than 10 seconds in Set-up time.

NFC requires comparatively low power; hence it works with an unpowered NFC tag. In contrast to Bluetooth, more power is needed to illuminate through a long range of distance. Operating with passive device is difficult.

2.2 NFC Mobile Architecture

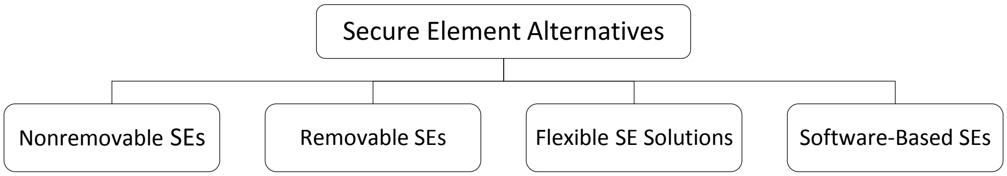
NFC-enabled mobile devices are typically composed of various circuits such as host controller, secure element(s) (SE) and NFC contactless front-end (NFC CLF). The NFC CLF includes an NFC controller and an NFC Antenna. NFC controller is the driver to convert host controller interface (HCI) commands to analog signal for NFC antenna. Host controller is the main controller of the mobile phone, which is connected to the NFC controller using HCI. The host controller sets the operating modes of the NFC controller then processes the data that is sent and received. On the other hand, SE provides a secure storage environment for private data such as credit card information. It is possible for a mobile phone to contain more than one secure element.



2.2.1 SE

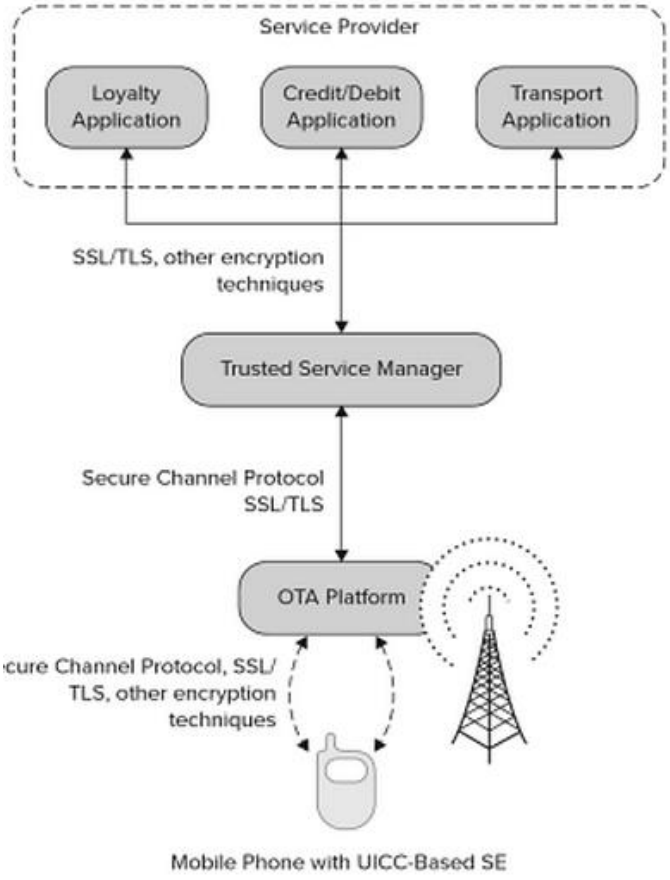
NFC enabled devices must ensure customer the credit card information and transaction takes place in a secure environment. The protection method is very similar to that of smart card technology. An SE, itself, is a separate microcontroller that has its own operating system, such as MULTOS and JavaCard OS. If such sensitive data is needed to transmit E.g. Payment, the secure element will directly access the NFC controller.

There are various SE alternatives available in the market. We can divide them into 4 parts: Nonremovable SEs, Removable SEs, Flexible SE Solutions and Software-Based SEs. Nonremovable SE is an embedded hardware which gives a relatively high security. The chip embedded in the mobile phone is inserted during the manufacturing phase; hence it has the same security level as smart card. Removable SE such as Secure Memory Card (SMC) and Universal Integrated Circuit Card (UICC), these allow user to change after purchasing the mobile phone, thus manufacturers only need to produce one generic phone for public. Furthermore, because the NFC enabled mobile phones are not very common todays. Thus, for SIM-based SEs and SMC-based SEs, SE manufacturers need to produce a generic SIM card that can support both NFC-enabled mobile phone and common mobile phone. Therefore, we have a new class of SE, Flexible SE Solutions.



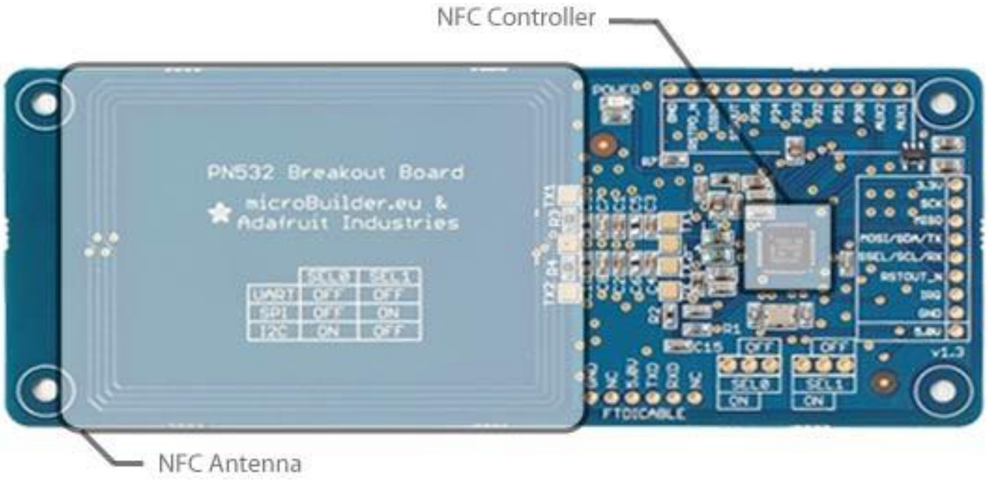
SE Management

Average user does not have the ability to alter the SE, but management is still needed. The authority is given to the Over-The-Air technology (OTA). Upon approval of user, They can install and load new NFC applications on SEs, remotely through UICCs. Besides, they can activate / deactivate SEs, remote service managing, PIN reset of NFC applications on SEs.


Service Provider
Loyalty Application
Credit/Debit Application
Transport Application
SSL/TLS, other encryption techniques
Trusted Service Manager
Secure Channel Protocol SSL/TLS
OTA Platform
cure Channel Protocol, SSL/ TLS, other encryption techniques
Mobile Phone with UICC-Based SE

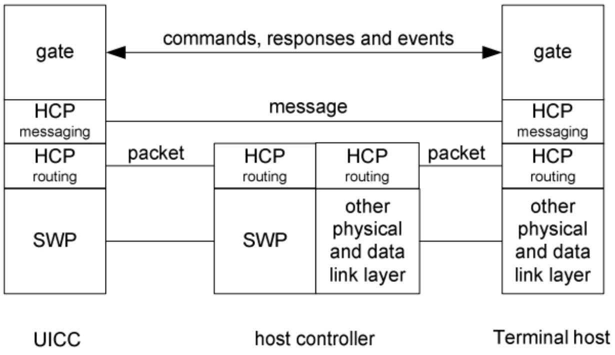
2.2.2 NFC Interface

The NFC Interface is composed of an NFC controller, an Antenna and NFC Contactless Front-End (NFC CLF). The NFC interface is able to communicate with host controller, SEs and other NFC interface through RF signal. It is the core component of NFC. The NFC controller works as a modulator and demodulator between the analog RF signal and NFC antenna. Besides, the NFC controller receives operations the host controller via HCI. The NFC CLF is the analog front end of the NFC controller. It is the logical interafce defines the protocol on top of the data link layer.



HCI

HCI is a logical interface allowing NFC interface to communicate directly with the host controller. Host controller sends command and data as a package to the NFC controller under Host Controller Protocol (HCP). Examples of HCP are SPI, I2C and HSU (High Speed UART) serial links.



At the beginning, the Host controller has to initialize the NFC controller by setting up configurations. After configuration, the NFC controller will wait for RF signal from other NFC devices. If RF signal is detected, the host controller is allowed to receive ID of the NFC tag / NFC device. Then host controller can use its ID to communicate with the NFC tag / NFC device using its protocol. For different type of target, the NFC controller will use different communication interface standards.

Parameters	ISO/IEC 14443	ISO/IEC 15693	ISO/IEC 18092
Operating Mode	Reader to card	Reader to card	Peer-to-Peer
Communication Mode	Passive	Passive	Active and Passive
Range	Proximity	Vicinity	Proximity
Data Rate	106 Kbps	Up to 26 Kbps	106, 212, 424 Kbps

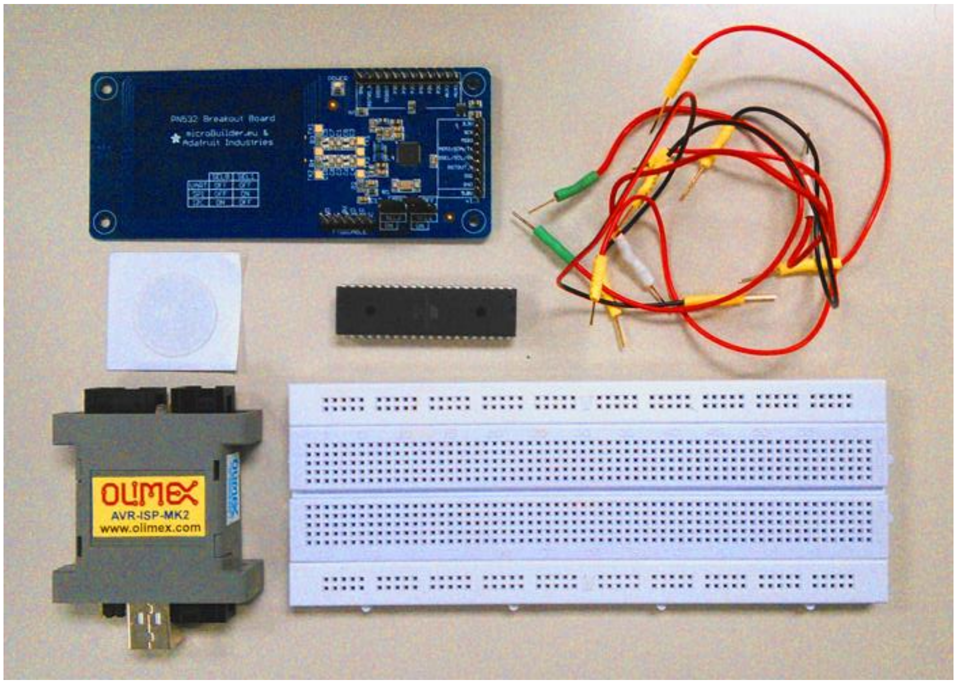
Serial Peripheral Interface Bus (SPI)

SPI is the transport layer between microcontrollers. Devices communicate in Master/Slave mode where the master initiates the data frame. A common clock is required to be established between master and slave. The Host controller, in this case, act as the master and the NFC controller act as a slave. All the commands in HCP will be converted into packets then be transmitted to the NFC controller.

2.2.3 Experiment

In order to investigate the operations of NFC in details, we have made a simple prototype to read and write NFC tags. In the prototype, it contains a simple host controller, NFC controller and an NFC antenna. The host controller tells the NFC controller to write an URL into an NFC tag. Then we will test it using an ordinary NFC-enabled mobile phone.

Components



- Breadboard
- Connectors
- ATMEGA324A microcontroller (Host Controller)
- PN532 NFC/RFID controller breakout board (NFC Interface)
- Mifare Ultralight (NFC Tag)

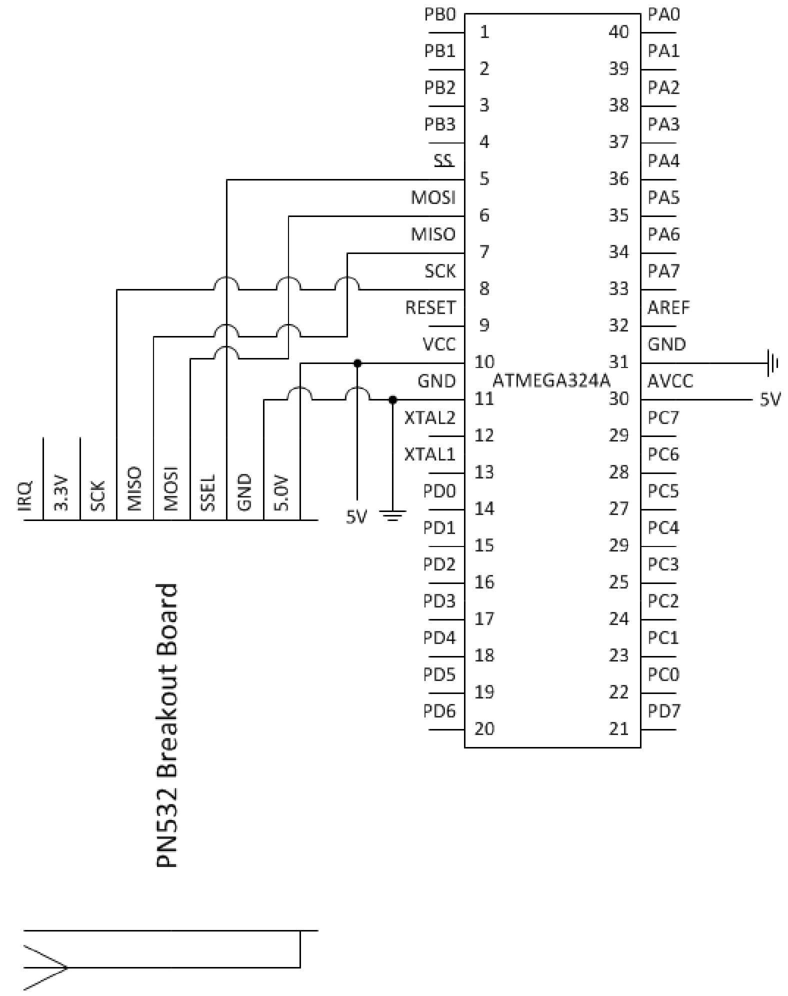
Equipment

- Computer
- Olimex AVR-ISP-MK2 Programmer
- Atmel Studio

Design

After the host controller initializes the NFC controller, the host-controller will keep monitor the incoming RF signal. If an anonymous NFC tag is moved in within a range, the host controller will write a hardcoded data into the tag. We have chosen to use SPI bus as the HCI, since the microcontroller use SPI as the default IO interface.

We connected the circuit as follows:

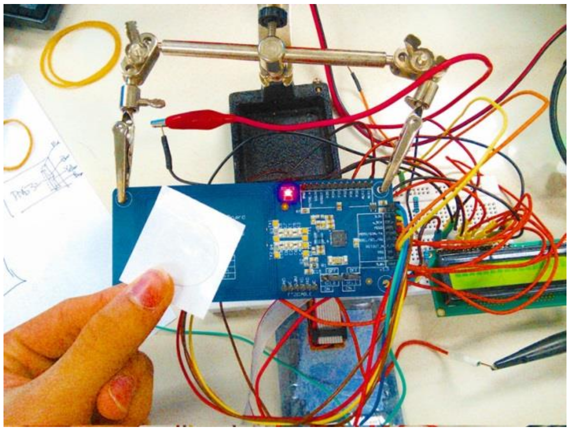


The controller writes an URL (http://www.cuhk.edu.hk) into a Mifare Ultralight Tag using ISO14443 standard whenever an NFC Tag is detected. This code is written as follows:

```
uint8_t data[20]={  0x03, 0x10, 0xd1, 0x01, 0x0c, 'U', 0x01,
                    'c', 'u', 'h', 'k', '.', 'e', 'd', 'u', '.', 'h', 'k', 0, 0};

// Initialize NFC Device
setupPins();
begin();
SAMConfig();

// Main loop
while(1)
{
        while(!readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uid_length));

        // When a tag is found
        for(i = 0; i < 5; i++)
        {
                // Authenticate block
                if(authenticateBlock(1, uid, uid_length, 4 + i, KEY_A, key))

                // Write data into the block
                writeMemoryBlock(1, 4 + i, data + i*4);
        }
}
```

Testing

We put a tag in front of the NFC antenna for a while, then put back to an NFC enabled mobile device. We can successfully initiates the CUHK website.



2.3 NFC Operating Mode

There are three classes of NFC devices: NFC enabled mobile device, NFC tag and NFC Reader. By pairing up these devices, we have different interaction styles. Each NFC operation has an initiator as well as a target. Initiator is an active device that initiates the communication with the target. Target, on the other hand, can be either active or passive. Hence, an NFC tag cannot be an initiator. In summary, the NFC operating modes are the reader/writer, peer-to-peer and card emulation modes with different interaction styles.

Operating Mode	Initiator	Target
Reader / Writer	NFC Mobile	NFC Tag
Peer-to-Peer	NFC Mobile	NFC Mobile
Card emulation	NFC Reader	NFC Mobile

Reader/Writer Mode enables NFC mobile device to exchange data with an NFC tag. Peer-to-Peer mode enables two NFC mobile devices to exchange data with each other. Card emulation mode allows Mobile device use its SEs to emulate a smart card such as credit card and ID card.

2.3.1 Reader/Writer Operating Mode

In reader/writer mode, the mobile device initiates the NFC tag by providing power through EM energy. Upon authentication, the mobile device is free to choose reader mode or writer mode. In reader mode, the mobile device can request a read operation on specific block of data in the tag. In writer mode, the mobile device overwrites any data on the block, modification is not supported. The only possible data rate in this mode is 106 Kbps.

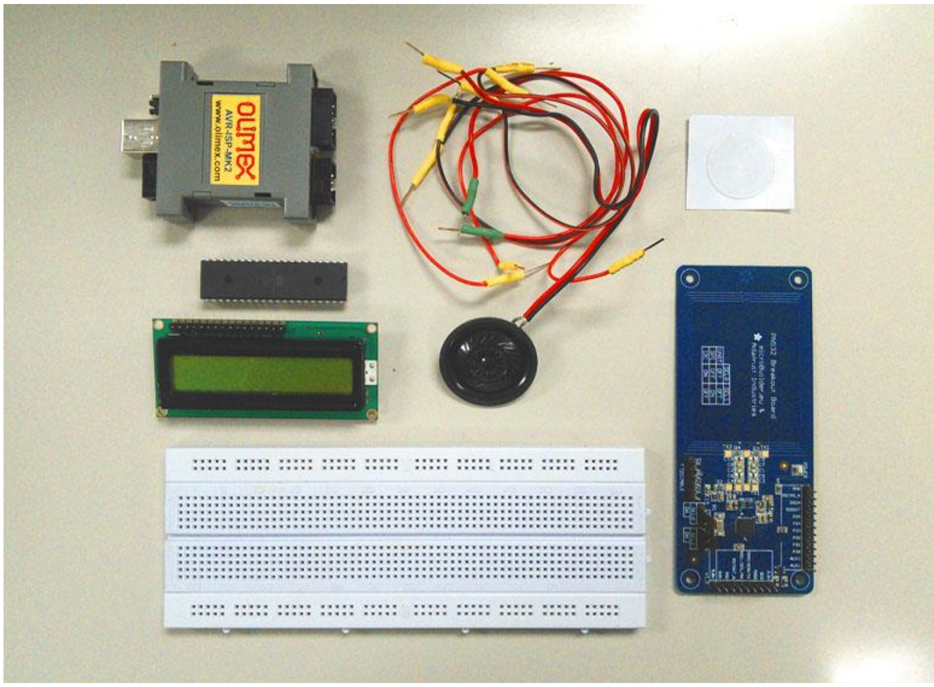
NFC enabled phone
Tag

Experiment 1

We have implemented a simple Octopus System. The NFC tag acts as an octopus card that stores the amount of money. On the other side, the NFC reader acts as the payment system in the minibus. Initially, the tag contains 300 dollars. When the tag is touched on the payment device, the money will be decremented by 3.

The process can be repeated, illustrating the read / write mode in NFC communication.

Components



- Breadboard
- Connectors
- ATMEGA324A microcontroller (Host Controller)
- PN532 NFC/RFID controller breakout board (NFC Interface)
- TS1620-1 LCD Monitor

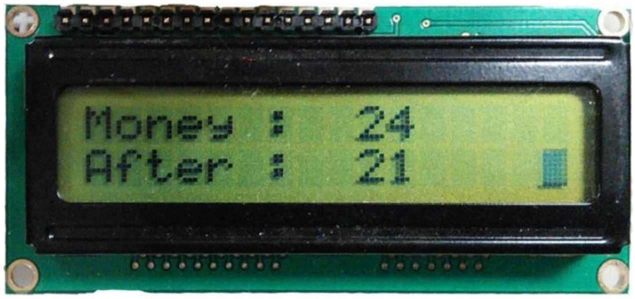
- Speaker
- Mifare Ultralight (NFC Tag)

Equipment

- Computer
- Olimex AVR-ISP-MK2 Programmer
- Atmel Studio

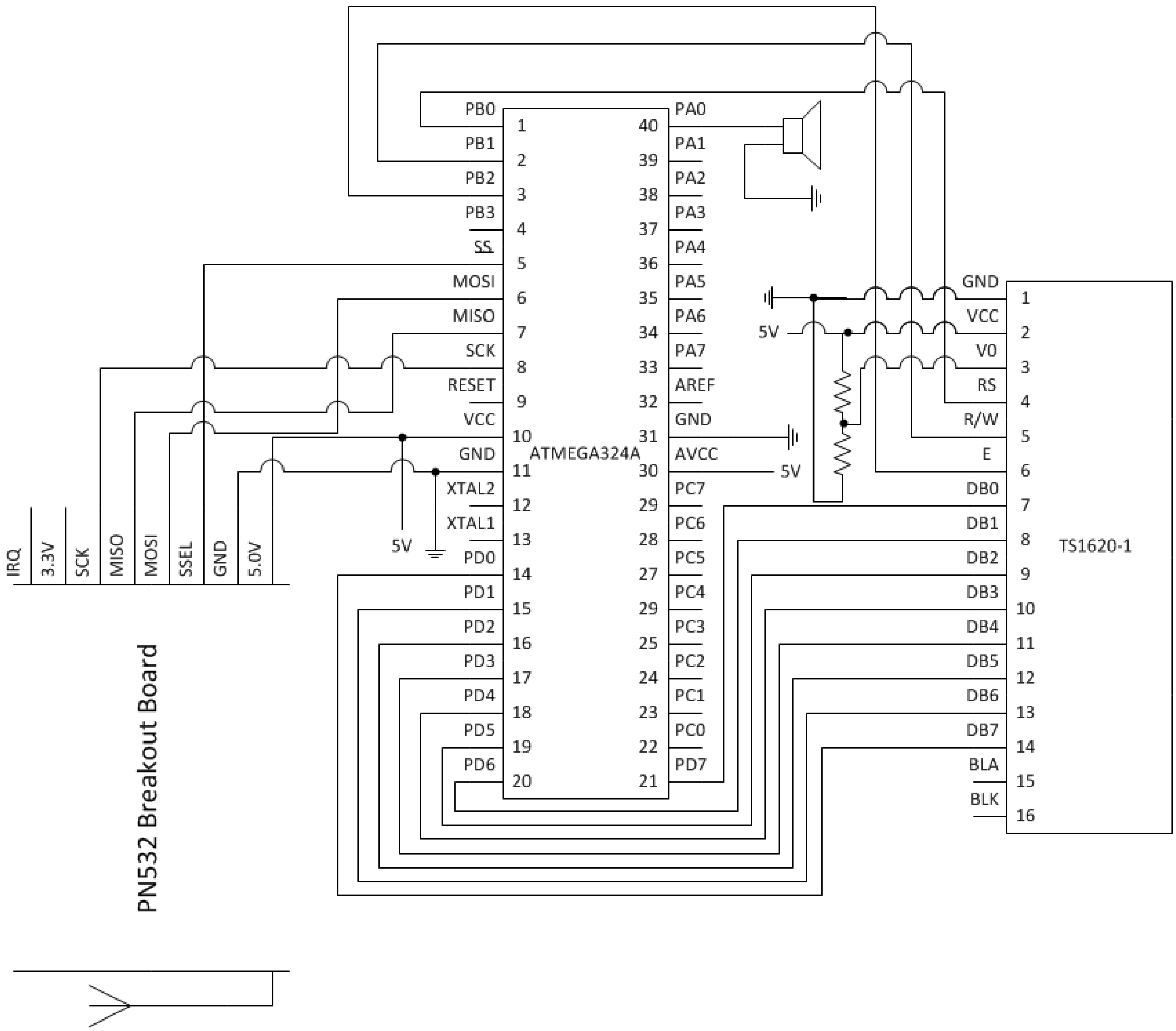
Design

When a valid NFC tag is detected, the host controller will read the data in the tag. Then it will decrement the value by 3 and write it back on the tag. At the same time, the speaker will gives a ‘beep’ sound indicating the transaction is done (It also simulates the sound effect on real octopus system). On the LCD monitor it shows the amount of money before and after.



In addition to the original circuit, we have added an LCD monitor and a speaker.

The circuit is modified as follows:



The host controller reads the value from the tag, and then parses the value into integer. After parsing, the value will be incremented by 3 then parse it back. Since the operation is so short, the read and write processes can be done in only 1ms. The possibility of failure is decreased dramatically.

The main part of the code is shown in the following

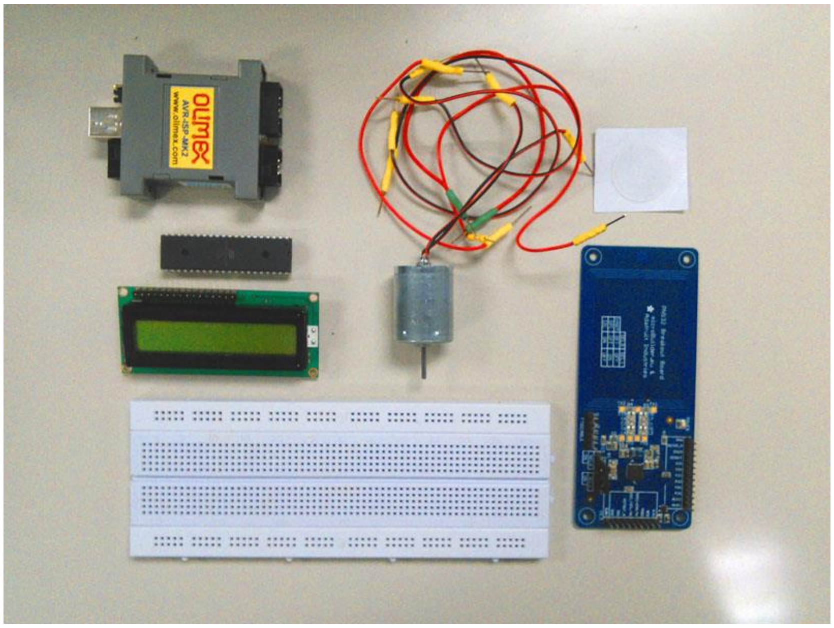
```
    // Initialize NFC Device
    setupPins();
    begin();
    SAMConfig();

    // Initialize 1602
    LCMInit();

    // Main loop
    while(1)
    {
             // Wait for a tag
             DisplayListChar(0, 0, "Put a tag      ");
             while(!readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uid_length));
```

```
// Authenticate Block
if(authenticateBlock(1, uid, uid_length, 6, KEY_A, key))
{
    // Read memory block
    if(readMemoryBlock(1, 6, data))
    {
        // Update the value
        putValue(data, getValue(data) - 3);
        writeMemoryBlock(1, 6, data);

        // Display the amount of money for user
        sprintf(str, "Money : %3d    ", getValue(data)+3);
        sprintf(str, "After : %3d    ", getValue(data));
        DisplayListChar(0,1, str);

        // Play the "beep" sound
        for(i = 0; i < 800; i++)
        {
            PORTA = 0x01;
            _delay_us(625);
            PORTA = 0x00;
            _delay_us(625);
        }
        DisplayListChar(0,1,"               ");
    }
}
```

Experiment 2

In this experiment, we will demonstrate using NFC to control hardware devices such as door lock. The NFC tag simulates a door key, and a motor simulates the door lock. Because each NFC tag has its own unique tag ID which cannot be modified, we use this key to verify the identity of the tag.

Components



- Breadboard
- Connectors
- ATMEGA324A microcontroller (Host Controller)

- PN532 NFC/RFID controller breakout board (NFC Interface)
- TS1620-1 LCD Monitor
- DC motor
- Mifare Ultralight (NFC Tag)

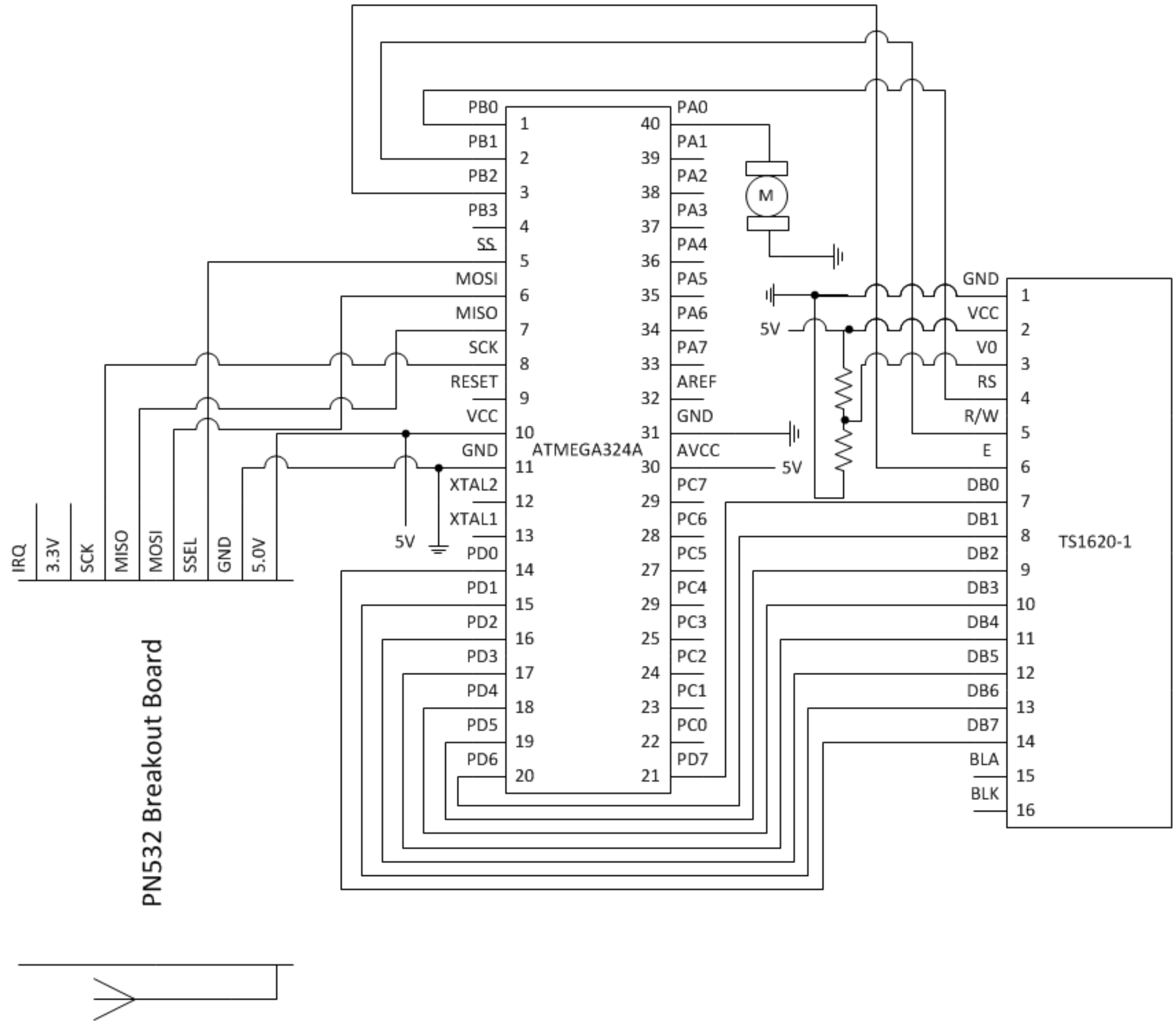
Equipment

- Computer
- Olimex AVR-ISP-MK2 Programmer
- Atmel Studio

Design

Initially, in the host controller, we have hardcoded a key as the unlocking key. When an NFC tag is detected, the NFC controller pass the tag ID to the host controller, then the host controller will compare the tag ID with the hardcoded key. If they are the same, the motor will spin indicating that the door unlocks.

The circuit is similar to that of the previous experiment except that the speaker is replaced by a DC motor. The circuit is connected as follows.



The code snippet is shown in the following, indicating the structure in the single loop processor.

```
    // Initialize NFC Device
    setupPins();
    begin();
    SAMConfig();

    // Initialize 1602
    LCMInit();

    // Main Loop
    while(1)
    {
        // Get card UID
        id = readPassiveTargetID(PN532_MIFARE_ISO14443A);
        while(id == 0)
            id = readPassiveTargetID(PN532_MIFARE_ISO14443A);

        // If card UID is the key
        if((((id >> 16) & 0xffff)==0x7a43) && ((id & 0xffff)==0x2f80))
        {
            // Show unlock message
            DisplayListChar(0,1,"Unlocked!   ");

            // Run motor
            PORTA = 1;
            _delay_ms(2000);
            PORTA = 0;
        }
        // If the card UID is not the key
        else
        {
            DisplayListChar(0,1,"Not this one");
            _delay_ms(2000);
        }

    }
```

Application

In reader/writer mode, there are various possibilities of applications. The data format can be URL, text, multimedia or even a self-defined format. After transmission, the app can use the data to do any kinds of operations. Hence, a wide range of applications in education, remote services, medical and entertainment potentially may be generated using the reader/writer mode.

Smart Poster

The most famous use of reader/writer mode is smart poster. The idea of smart poster is to make the poster alive. An NFC tag is prepared with an URI link, when an NFC enabled mobile device touches the tag, a corresponding website will pop up. This tag will then be stuck on a poster.

Business Card

Business card is another application using the NFC tag. People save his personal information into an NFC tag, whoever wants to get the contact information; he can simply use its mobile phone to touch the tag. Then the contact information will be automatically inserted into his phone.

NFC Shopping

Recently, an electronics retailer opens an NFC shopping wall in Russian subway. On the wall, it listed the photos of products. Customers are required to use its NFC mobile phone to touch the tag on the wall. Then the corresponding website pops up with the product in the online shopping cart. By using the mobile phone to touch the tags, customer will finally get a shopping cart with a list of items. After that, the customer can submit the shopping cart and pay for it online.

2.3.2 Peer-to-Peer Operating Mode (P2P)

P2P mode is the operating mode which allows two mobile devices to communicate with each other. Since, this mode is new from RFID, NFC has its own standards; i.e. NFC Interface and Protocol-1 (NFCIP-1) and Logical Link Control Protocol (LLCP). Besides, both devices are active and one of them has to be the initiator, they communicate in bidirectional half-duplex mode. However, the data rate is only up to 424 Kbps; transmission of large data is not practical. The high security nature of NFC allows mobile phones to exchange sensitive information. Since NFC has short handshaking time, it facilitates the transmission process. This feature also gives another set of applications. NFC can help other connection to finish the handshaking process. For example, user can use NFC to pair up

Bluetooth devices. With the help of NFC, it improves the security of the pair up process and shortens the authentication time.

Application

The transmission data format of Peer-to-peer mode is very similar to that of Reader/Writer mode. Typical data format are text, image, file and business card. Since both are active devices, they have higher flexibility with respect to Reader/Writer mode. Device can change the content of transmission as they wish. Developer is allowed to develop their own protocol on top of P2P mode depends on the application. On the other hand, hardware can use NFC P2P mode to pair up with the mobile phone such as Bluetooth earphone.

S-Beam

One of the largest limitations in NFC is its low transfer rate. It is not capable to transfer large file such as video. However, Samsung invented a new usage of NFC which enhances the transmission to another level. S-Beam allows users to pair up their phones using NFC, then it connects the phones using Wi-Fi-Direct and transfer the file/data through Wi-Fi. Since Wi-Fi has much higher data rate, the transmission speed is improved.

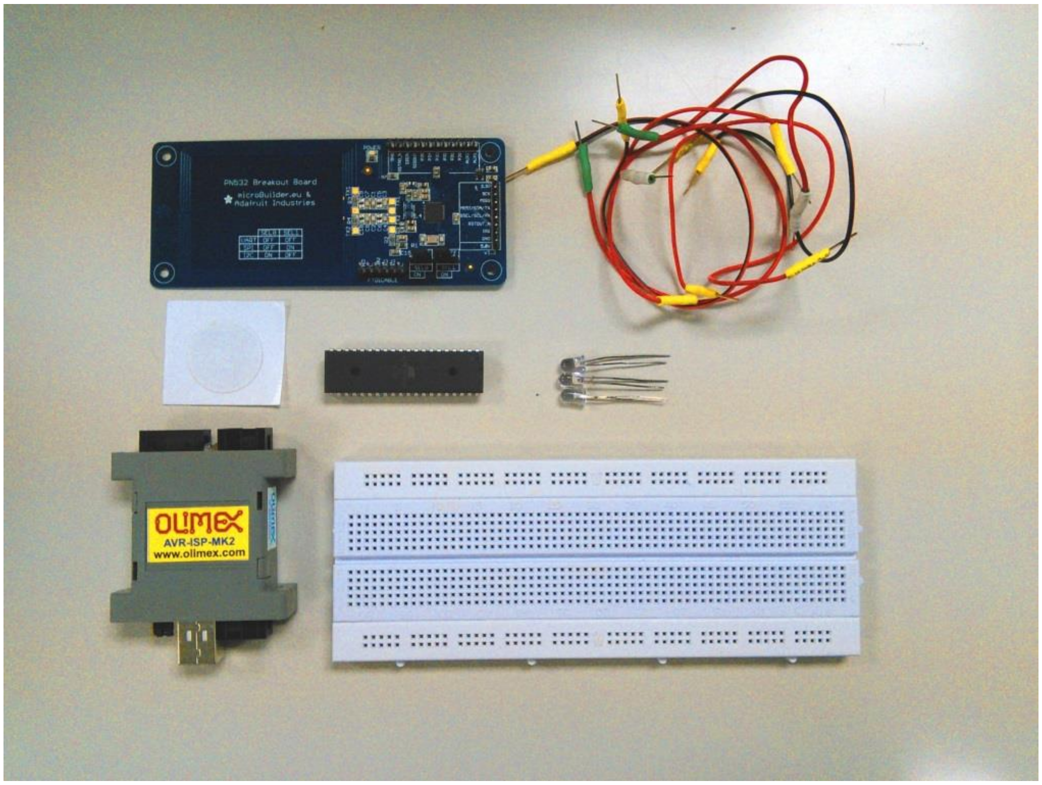
2.3.3 Card Emulation Operating Mode

Card emulation mode allows NFC-enabled mobile devices to function as contactless smart cards such as credit cards and loyalty cards. In this mode, the mobile device acts as a passive tag and the card reader act as an active initiator. The reader generates RF field to “power up” the NFC interface in mobile device (similar to the communication in RFID), then the mobile device sends RF signal using that field. This transmission mode enhances the security. Besides, with the SEs embedded in mobile devices, NFC reader directly communicates with the SEs, hence, the host controller is not allowed to eavesdrop the data in between.

Experiment

This experiment simulates the card emulation mode using the NFC reader and an NFC-enabled mobile device. The mobile device act as a light color controller and a circuit will simulates an NFC enabled lamp. Because individual SE can only owned by the manufacturer for security reason, we have to use an NFC tag to simulate the SE in the phone.

Components



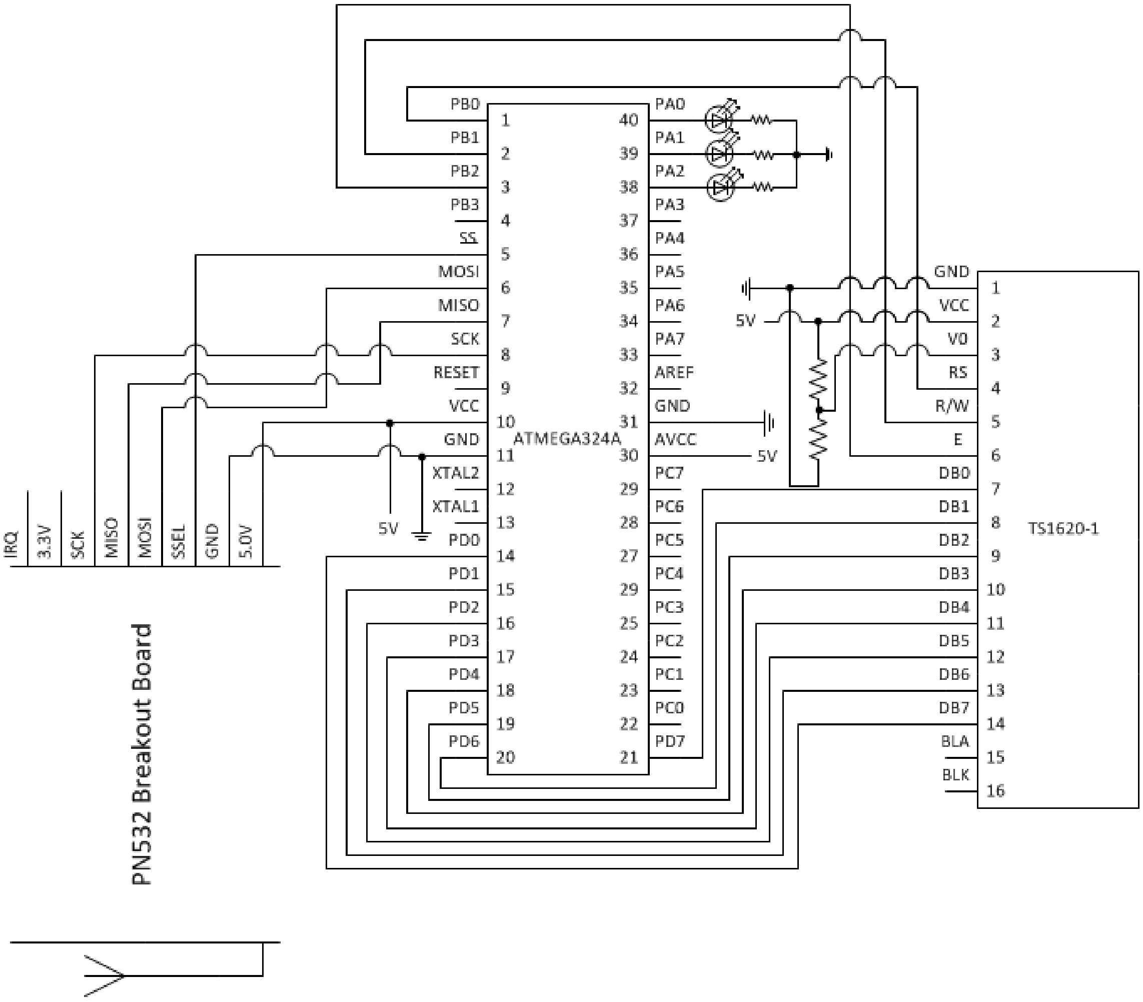
- Breadboard
- Connectors
- Samsung Mega (NFC-enabled mobile device)
- ATMEGA324A microcontroller (Host Controller)
- PN532 NFC/RFID controller breakout board (NFC Interface)
- TS1620-1 LCD Monitor
- Red LED
- Green LED
- Blue LED
- Mifare Ultralight (NFC Tag)

Equipment

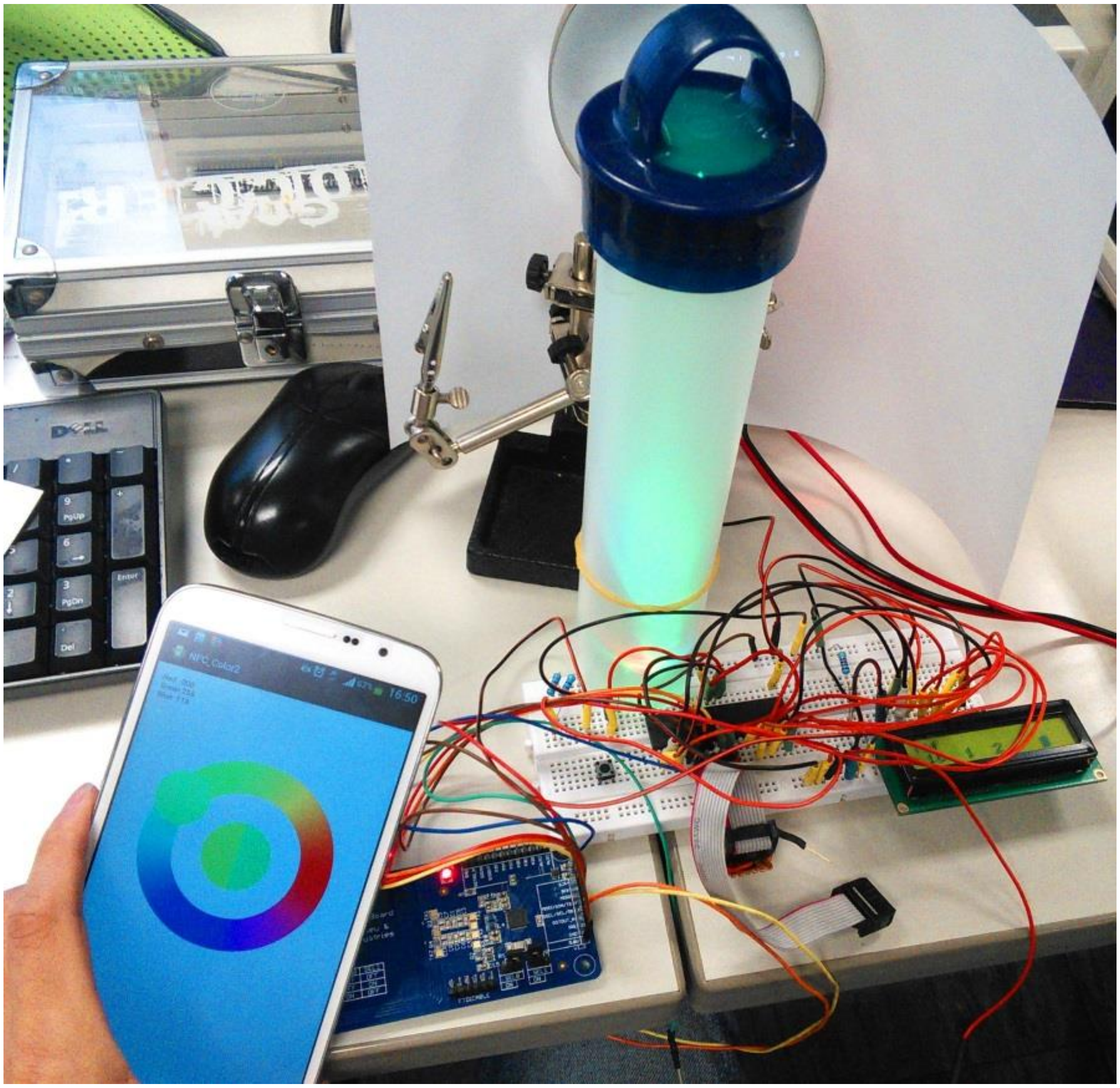
- Computer
- Olimex AVR-ISP-MK2 Programmer
- Atmel Studio

Design

The experiment composed of two devices and an NFC tag. The NFC reader controls the LEDs, each LED represents one color. By varying the intensity of the LEDs, different color can be represented.



The NFC tag contains the information of present color. It sticks on the NFC reader, and the NFC reader continuously monitor the content in the tag then updates the color of the LEDs. In the NFC-enabled mobile device, we have implemented an app which shows a color picker and writes the data into the NFC tag.



Since, the microcontroller cannot output analog signal. Hence, we use frequency to control the intensity of LEDs. Also, single thread nature of the processor makes it difficult to monitor the tag. Therefore, in high level implementation like C, we have to schedule different operation in a while loop. The code is given as follows

```
    // Initialize NFC Device
    setupPins();
    begin();
    SAMConfig();

    // Initialize 1602
    LCMInit();

    // Open LEDs ports
    PORTA = 0x07;
    DDRA = 0x00;

    // Main loop
    while(1)
    {
    // When NFC tag is found
        if(readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uid_length)){
            // Authenticate Block
            if(authenticateBlock(1, uid, uid_length, 6, KEY_A, key)){
                if(readMemoryBlock(1, 6, data)){
                    // Discretize the intensity into 1 to 16
                    r = 16 - getValue(data+1) / 16;
                    g = 16 - getValue(data+4) / 16;
                    b = 16 - getValue(data+7) / 16;
                }
            }
            releaseTag();
        }
        // Blink the LEDs
        for(i = 0; i < DIV; i++){
            if(i%r == 0) DDRA |= 0x01;
            else                DDRA &= 0xfe;

            if(i%g == 0) DDRA |= 0x02;
            else                DDRA &= 0xfd;

            if(i%b == 0) DDRA |= 0x04;
            else                DDRA &= 0xfb;
        }
    }
```

Application

The application of card emulation is very strict forward; it simulates different kinds of cards. By making use of the SE in the mobile device, the mobile device can serve as a credit card, debit card, loyalty card, ticket, access control unit and identity card.

Door Key

Hotel commonly uses an RFID card to unlock the door, however, a passive RFID card is not flexible and guest can steal it easily. NFC technology is proposed to use in hotel’s door lock system. When a guest books a room from hotel, room information and digital key are sent to the guest’s mobile phone. Since, the mobile device is active; a timer can be set inside the mobile phone. We can also extend the application to house door lock. One can use mobile phone to unlock its door, or even transfer a temporary key to a friend through the internet.

Payment

One of the big focuses on NFC is the replacement of credit card. By adding applications into SEs, one mobile device can emulates a set of credit cards and debit cards. When an NFC reader triggers the mobile phone, only one corresponding application in SE is allowed to communicate.

2.4 NFC Tag

NFC tag is an RFID tag that has no integrated power source. It is a low-cost and low-capacity passive device. Since it does not contain any power source, communication needs to be triggered by an active device. The active device use its RF fields to power up the NFC tag, then the communication can be start.

2.4.1 NFC Forum Tag Types

According to the NFC forum, NFC tag is defined into four tag types. Each tag types has different capacity and format.

Parameter	Type 1	Type 2	Type 3	Type 4
Based On	ISO/IEC 14443 Type A	ISO/IEC 14443 Type A	FeliCa	ISO/IEC 14443 Type A, Type B
Chip Name	Topaz	MIFARE	FeliCa	DESFire, SmartMX-JCOP
Memory Size	Up to 1 KB	Up to 2 KB	Up to 1 MB	Up to 64 KB
Data Rate	106	106	212	106-424
Cost	Low	Low	High	Medium/High
Security	16-bit or 32-bit digital signature	Insecure	16-bit or 32-bit digital signature	Variable

Type 1

Type 1 utilizes a simple memory model with communication speed 106 Kbps. Since is the most simple type, this tag type is cost effective and can be used in most NFC applications. The tag belongs to this type is readable and writable. It can be reformatted as read-only tag. The memory size is only up to 1KB; hence it is only just enough to save URLs and text messages. For larger data such as contact information and images, type 1 tag is not capable to store it. The structure of the tag is rather simple; it has two memory model mappings, static and dynamic, depending on the size of the tag. The tag is divided into blocks; each block contains 8 bytes of data.

Type 2

Type 2 tag is very similar to Type 1 tag. This tag type is also readable and writable, but the only difference is that the memory size is expanded to 2KB and no digital signature embedded inside the tag.

Type 3

This tag type is defined base on Sony FeliCa contactless smart card interface, hence it is more suitable for complex applications. It is also writable but with block size 16 bytes. Hence, its maximum memory size is 2KB. Due to the technology backed by contactless smart card, the capacity can be expanded into 1MB. The data communication speed is doubled to 212 Kbps, it is the most expensive among four.

Type 4

NFC tag of type 4 can be preconfigured into read-only in the manufacturing phase. This can guarantee the security of the NFC tag, since no one including the buyer can modify the data inside the tag. The capacity can be up to 64KB which has a larger capacity than Type 1 and Type 2.

2.4.2 Major Proximity Contactless Smart Card Technologies

NFC tag is developed on contactless smart card, hence they have the same fundamental structure. Up to now, the most famous and competing contactless smart cards are Mifare, Calypso and FeliCa.

MIFARE

Mifare is a widely used contactless proximity smart card system owned by NXP Semiconductors. The Mifare family contains different types of cards, such as Mifare Classics, Mifare DESFire, Mifare Ultralight and Mifare Ultralight C. Mifare-based smart card is widely used in range of applications, access management, e-payment and loyalty applications such as student card (CULink) in The Chinese University of Hong Kong.

FeliCa

FeliCa is a high-speed proximity smart card system from Sony. It is designed for electronics money card. Primary application of FeliCa smart card is public transport ticketing, for example the Octopus card in Hong Kong Mass Transit Railway (MTR).

2.4.3 NFC Tag vs. QR Tag

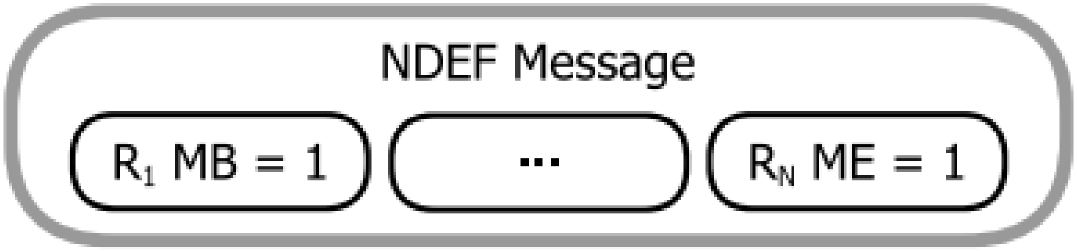
QR code is a 2-dimensional bar code that stores a short piece of information. The data type of QR code is usually URL website address. Common mobile phones contain cameras, hence mobile phone can use the camera to capture the QR code then decode it into URL message. The usage of QR code is similar to that of NFC tag, comparison between them can emphasize the characteristic of NFC tag.

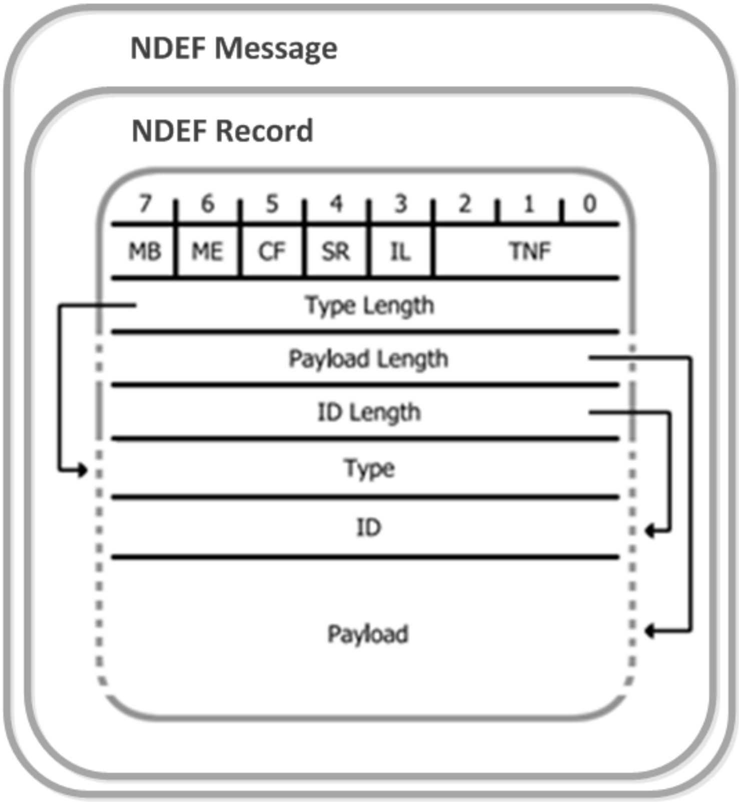
QR code requires user to use a camera app to focus on the barcode, the whole process (launch the app, focus on the QR code, wait until the app recognize the QR code) used approximately 10 seconds. But using NFC only needs 2 seconds, the corresponding app will be launched after the transmission. However, QR code is very easy to create. Only a black ink printer is needed to make a QR code. [2]But NFC tag requires a special circuit embedded inside the tag, which is not easily available. In the issue of security, NFC tag offer a unique manufacture ID number and specialist tags are also support encryption. For the QR code, it offers no security.

	NFC Tag	QR Code
User Experience	Immediate, no additional software	~10s, requires launching an app before scanning
Cost	HKD$ 1.00 – 10.00	~HKD$ 0.007
Size	10-30mm (diameter)	>20mm
Product Integration	Can be embedded inside a product	Must be printed visibly
Print and Customization	Full color custom print	Limited space without affecting the barcode
Availability in Mobile Phones	10% of smartphones	All smartphones
Programming	Mobile apps available on NFC-enabled smartphones	Wide range of websites with QR generation service
Security	UID, Encryption support	No

2.4.4 NFC Data Exchange Format (NDEF)

NDEF is the data format to exchange information between NFC devices. It is also the data structure inside an NFC tag. The NDEF message can be sent either in NFCIP-1 or LLCP. In one transmission, it can include more than one NDEF message. In each NDEF message, there are some NDEF records. Each record contains one piece of information in any defined format.





In an NDEF record, it has a definite data structure with varies fields. One of the fields, Type Name Format (TNF) defines the format of the data type. This field is a 3-bit field with values defined in the table

Type Name Format	Value
Empty	0x00
NFC Forum well-known type	0x01
Media-type	0x02
Absolute URI	0x03
NFC Forum external type	0x04
Unknown	0x05
Unchanged	0x06
Reserved	0x07

The NFC Forum well-known type defines many useful data types, such as smart poster, text, URI and signature. NFC Forum also allows developer to define his own data type in NFC Forum external type. The NFC Tag used in Latia is defined from this type.

The Payload is the real content of the data. Since payload length is a 32-bit unsigned integer, the payload consists up to 2^{32}-1 octets in size, which should be more than enough compare to the maximum data rate 424Kbps.

3 Studies On Android NFC Package

In Android SDK, NFC is already supported at API level 9. Since Latia is only used Reader/Writer mode, we only discuss on the operations in this mode.

3.1 Android NFC Package

Android SDK provides two packages for different usages. android.nfc package is used in high level operation such as reading NDEF contents in known structure. android.nfc.tech package focuses on low level operation, it provides classes for different tag types, in each class it also provides methods to do low level tag operations.

3.2 NFC Dispatch System

There are two NFC dispatch systems in Android Reader/Writer mode. Intent dispatch system and Foreground dispatch system, each has its own characteristic in the application level. The tag intent dispatch system is used to launch applications when the predefined tags or NDEF data are identified in tags. After scanning an NFC tag, the corresponding registered application will be launched. For example, after installation of Latia, the application is automatically registered for our NFC tag. Hence, when the mobile phone touches to the tag, Latia will pop up and shows the corresponding content.

On the other hand, the foreground dispatch system is designed to handle tags when the application is running. After the foreground dispatch system is enabled, it will stand by and wait for any registered NFC tag. When the Android device touches the NFC tag, the applications registered as in Intent dispatch system will be ignored, since the priority of foreground dispatch system is greater than that of Intent dispatch system.

In coding, tag intent dispatch system requires application registered the NFC tag types or NDEF data types in the manifest file using intent filter. While the foreground dispatch system registers inside an activity.

3.2.1 Tag Intent Dispatch System

When an NFC tag is scanned, the NFC controller notifies the Host controller of the mobile phone, then the OS will reads the content and UID of the NFC tag. Depends on the content, the OS will look for any registered application. If an activity of an application is registered to this content, OS will intent to that activity. If more than application is registered to the same content, an application chooser will pop up showing the registered applications; in this case user has to choose their preferred application. On the other hand, no application registered to this content, and then the NFC action will be ignored.

In registration, application can register to a specific NDEF data type such as URI and text. However, due to different situations, the content inside the NFC tag may not be concise enough. For example, the scanned tag may contain a valid NDEF structure and a known NDEF payload data type. A tag can also contain a valid NDEF structure but an unknown NDEF payload data type. Even worse, the scanned tag does not contain a valid NDEF structure but it is a known tag technology. Due to the above options, three different intents are developed:

ACTION_NDEF_DISCOVERED, ACTION_TECH_DISCOVERED and ACTION_TAG_DISCOVERED.

ACTION_NDEF_DISCOVERED

This intent has the highest priority among three. When a tag contains a valid NDEF data type, the OS will try to run the registered application. For instance, Latia registers this intent for our NFC tag. When an NFC tag is touched and it contains a valid NDEF format with Latia data type, OS will launch the corresponding activity in Latia.

ACTION_TECH_DISCOVERED

This intent has the second highest priority. If a valid NDEF format is detect from the scanned tag, but no activity has registered to the data type, the OS will call the ACTION_TECH_DISCOVERED intent. In this intent, application has to register to a tag type, such as Mifare Classic and Mifare Ultralight.

ACTION_TAG_DISCOVERED

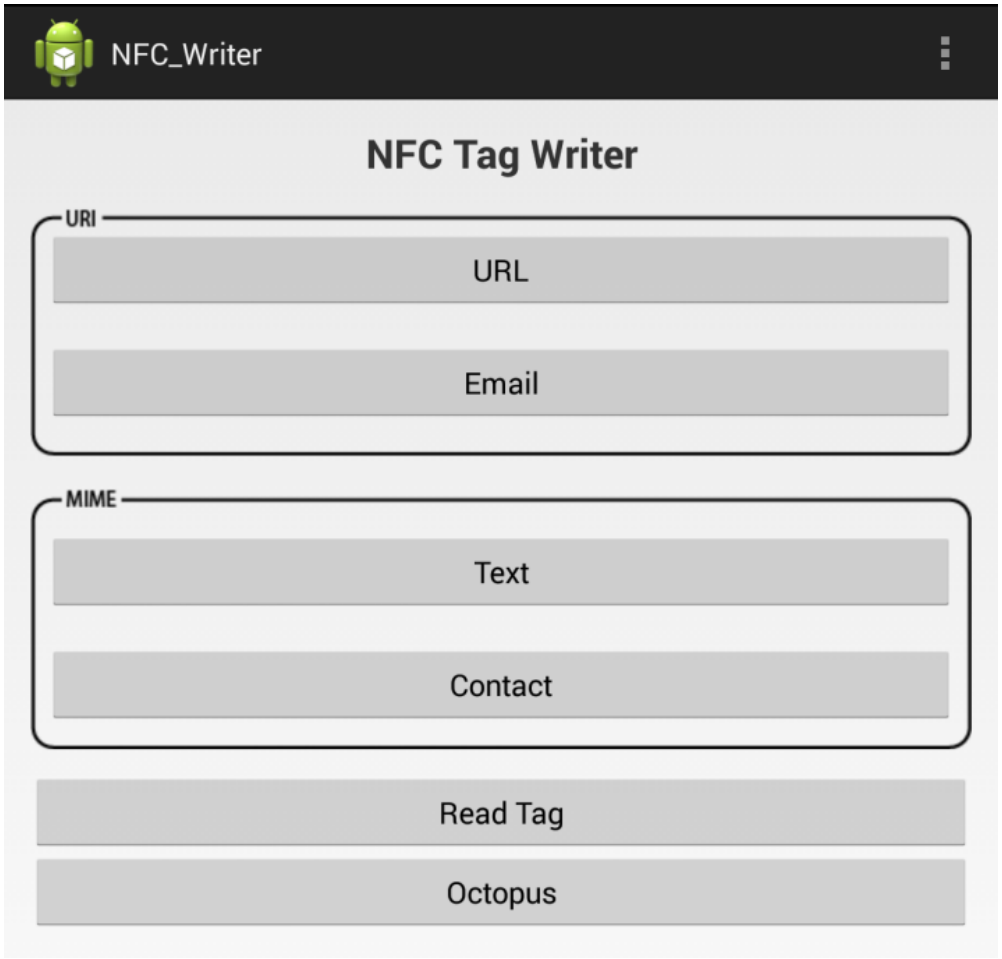
This intent has the lowest priority. If an NFC tag is detected but the NDEF data type is not recognized, or even the tag type is not recognized, then ACTION_TAG_DISCOVERED intent will be called. In this case, the programmer has to write the communication protocol by himself. This intent is rarely used.

3.2.2 Foreground Dispatch System

Foreground Dispatch system handles the NFC tag when the application is in active states. The priority of Foreground dispatch system is greater than Tag intent dispatch system. Hence, the active application has the full control of the incoming NFC tag. This is useful in writing operation.

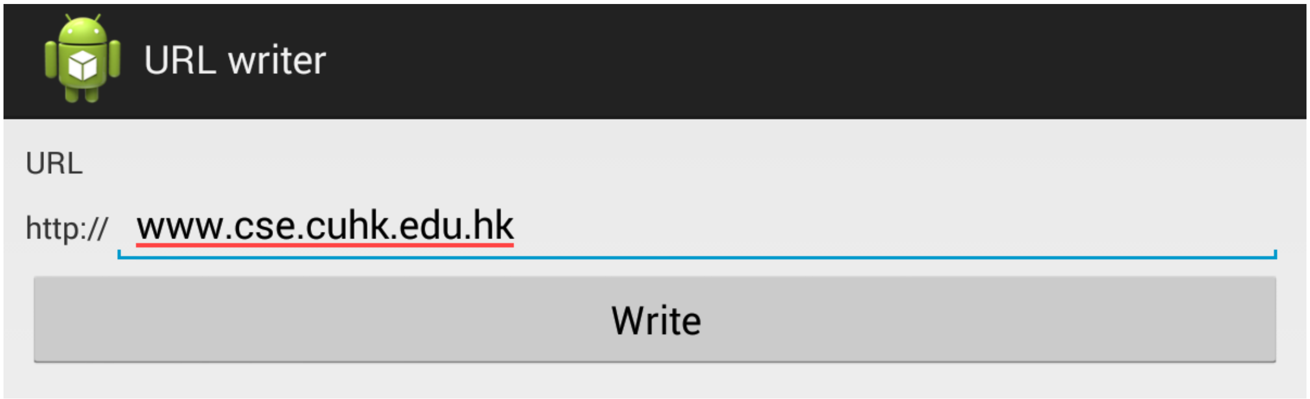
3.3 Experiment

Before we begin to develop our Android application using NFC (Latia), we have written an NFC tag writer. This application can writes variety data types into a tag. We also learned UI interface through developing this app. This application not only demonstrates the usage of NFC Foreground dispatch system, it also demonstrates the intents between activities as well as the 9-patch image.



Specifically, the application can writes two URI formats and two MIME (Multipurpose Internet Mail Extensions) formats. It can also read the NDEF record in the NFC tag. It uses the Foreground Dispatch system to do all the read/write operations.

3.3.1 URI Writer

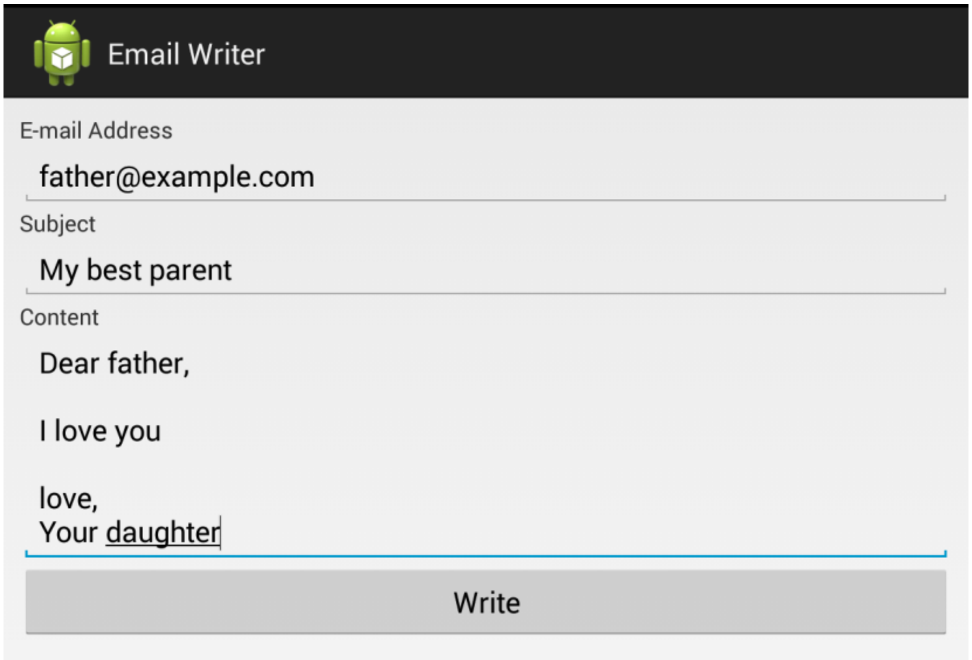


URL

http:// www.cse.cuhk.edu.hk

The URI writer allows user to write an HTTP link into the tag. User inputs the URL such as http://www.cse.cuhk.edu.hk/ then press write. The NDEF message of the link will be constructed and will be sent to the activity of Foreground dispatch system. When an NFC enabled android device touches the NFC tag, the browser will pop up with the link stored in the tag.

3.3.2 Email Writer



Email Writer

E-mail Address

father@example.com

Subject

My best parent

Content

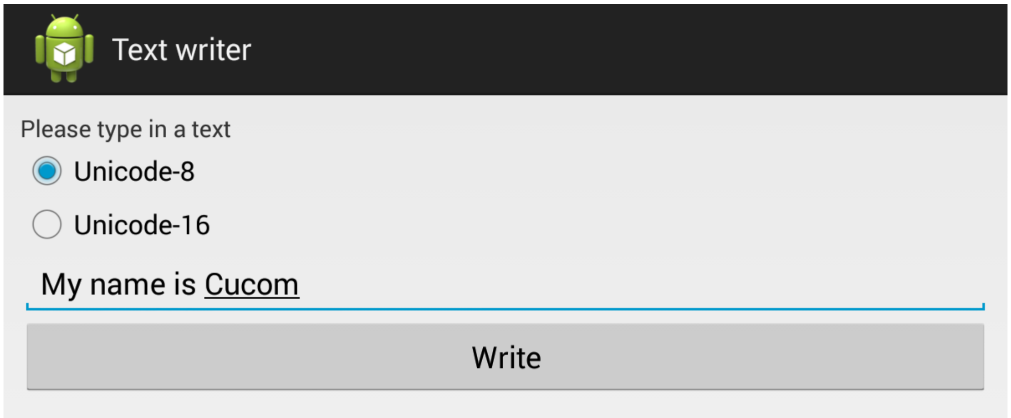
Dear father,

I love you

love,
Your daughter

Email writer allows user to input the target email address, subject and content. The NFC tag stores all the details, when the phone touches the tag, the default email application will pop up. In the application it automatically creates a new email filled with the information stored in the tag. It is useful to create a framework for emails.

3.3.3 Text Writer



Please type in a text

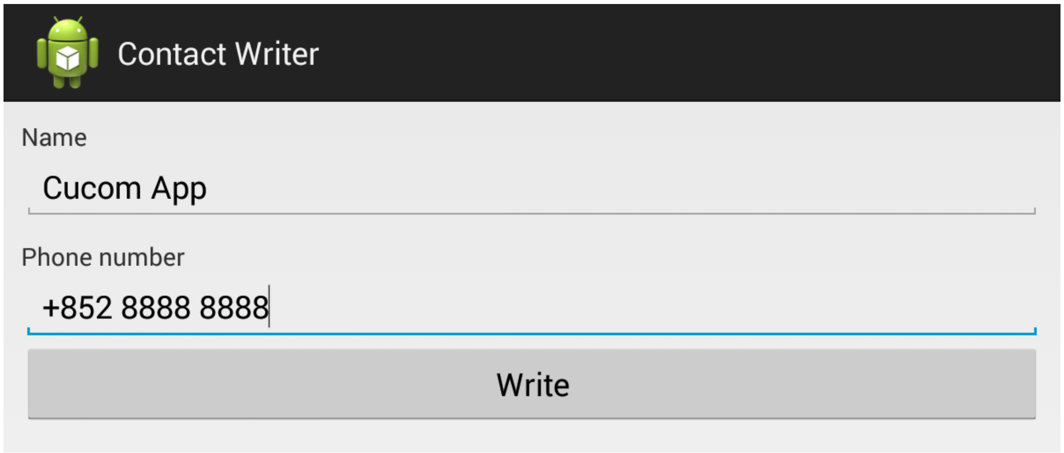
● Unicode-8

○ Unicode-16

My name is Cucom

Text writer simply writes a text into an NFC tag. User is free to choose between Unicode-8 and Unicode-16 input format. There is no default application that handles text message. Hence, user needs to install a specific application for text message.

3.3.4 Contact Writer



Contact Writer

Name

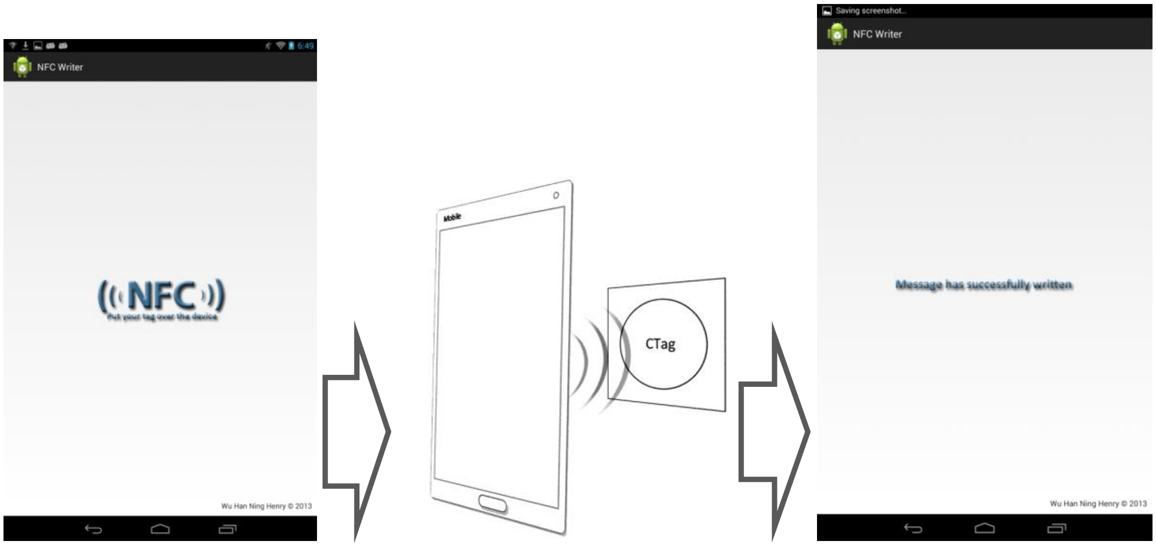
Cucom App

Phone number

+852 8888 8888

This writer enables user to store simple contact information into a tag. The tag immediately turns into a business card. When a mobile phone touches the tag, it automatically adds the contact into the phone.

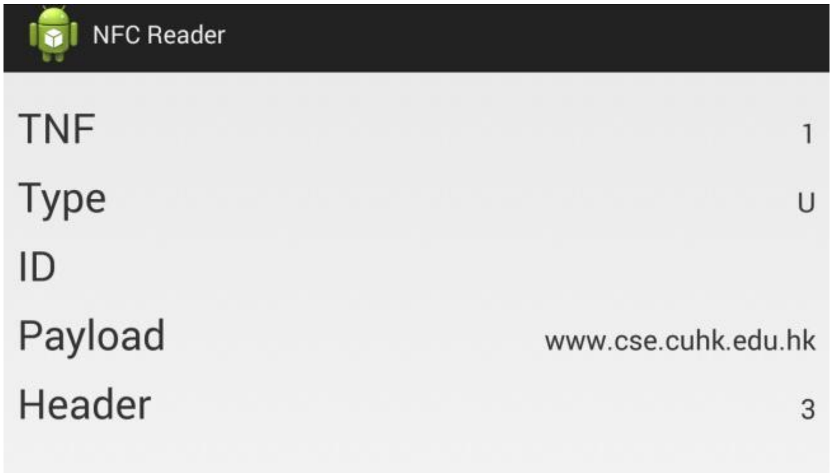
3.3.5 Foreground Dispatch System Activity



Message has successfully written

When the user press write from the above activities, it will intent to here. In this activity, it enables a foreground dispatch system, user needs to put the phone on the NFC tag and wait for writing operation. When the operation is done, an acknowledgement will be displayed on the screen.

3.3.6 NFC Tag Reader



TNF	1
Type	U
ID	
Payload	www.cse.cuhk.edu.hk
Header	3

This activity also equips a Foreground dispatch system. When a tag with a valid NDEF is detected, it reads the content in the tag, and then shows it on the screen.

4 USER EXPERIENCE

In the definition of ISO 9241-210, user experience is defined as ‘a person's perceptions and responses that result from the use or anticipated use of a product, system or service’. The user experience is an important role for a system since user experience just like the information architecture. The bad information architecture will lead to ineffective, inefficient and unsatisfying experience of using the system for the users. Note that using mobile device is different from using desktop computer. With a smaller display and different styles of user interaction, the design for mobile applications becomes strong influence in application development. The paradigm of mobile user interface is based on widgets, touch, motion and keyboards, which is different to the common WIMP (Windows, Icons, Menus, Pointer) interface style of Apple’s iOS and Microsoft Windows. With the limited screen size, UI gesture becomes an important role on app development.

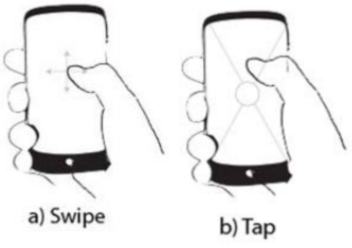
4.1 UI Gesture

Most of the people are distracted by another task when using smartphone. For example, walking or taking MTR. Thus, there are two principles of the user interfaces for apps when users are distracted:

1. The interface should limit the need for visual attention during interaction.

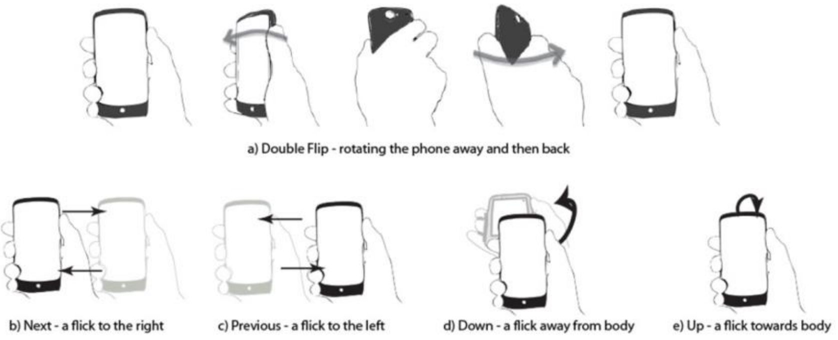
2. The interface should provide streamlined commands for the most common tasks

These principles can be take benefit by applying suitable gesture. There are three alternative input forms supported by smartphone: tap, swipe and physical move.



a) Swipe

b) Tap



a) Double Flip - rotating the phone away and then back

b) Next - a flick to the right

c) Previous - a flick to the left

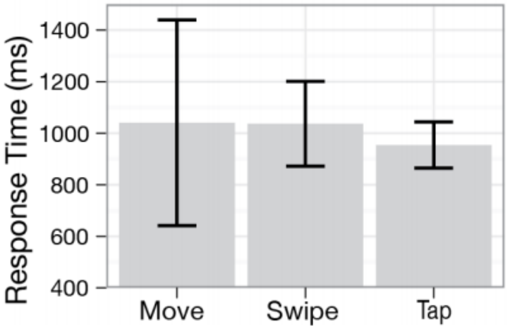
d) Down - a flick away from body

e) Up - a flick towards body

Each of them has their advantages and disadvantages. Since our project aims to encourage users to be interactive with the physical world, we mainly focus on the performance of moving scenario. The following are the different statistic measured when the participants are walking. The statistics are response time, success rate, screen gaze, walking speed and the times lost.

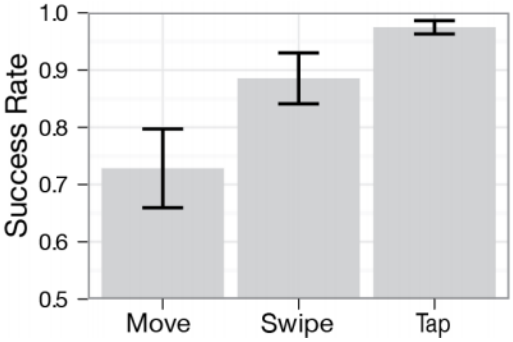
4.1.1 Response time

The response time is related to the difficulty of taking corresponding action after given the commands. The graph shows that the response time of Move and Swipe are slightly longer than the Tap.



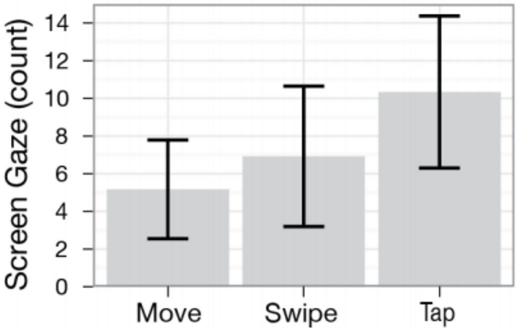
4.1.2 Success rate

The success rate is the percentage of performs the command successfully. The success rate of Tap and Swipe is higher than that of move.



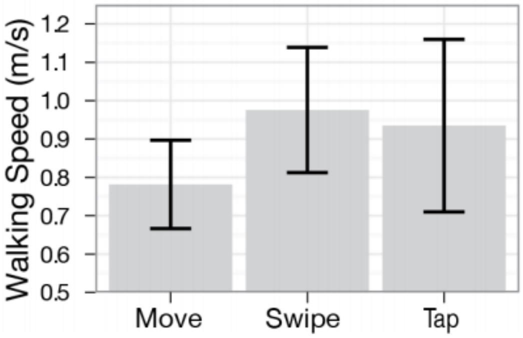
4.1.3 Screen gaze

For the screen gaze, the number of times that participants looked at the smartphone screen for Move and Swipe are less than that of Tag.



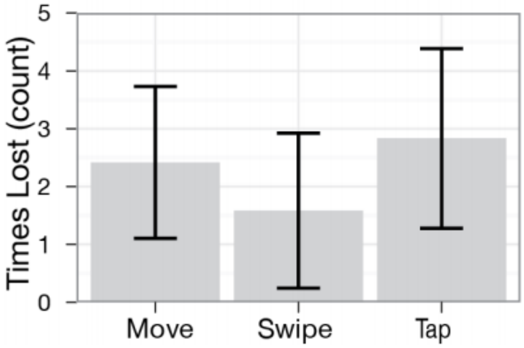
4.1.4 Walking speed

It is the walking speed when the participants perform these actions. The result is that the walking speed of Swipe is slightly higher than that of Move and Tap.



4.1.5 Times Lost

The times lost are the number of times participants had to stop and accommodate themselves. The number count of Swipe is the least among others.



According the above figures, we found that the performance of Swipe in different aspects are the most stable one on moving scenario.

4.2 Classification

Classification may refer to the categorization, which is the way that how the ideas or objects to be recognized, differentiated and understood. For a search engine, it is important to use classification to classify the data in order to search information efficiently. A good classification not only can reduce the group of data for searching, but also provides an overview of the data so that the users can view it clearly.

The classification has many types, but we will focus on two types: Hierarchical and Tagging.

4.2.1 Hierarchy

Hierarchical classification is a method of grouping in which terms are arranged from general to specific that the structure is initially arranged in broad groups which are then mutual exclusivity subdivided into narrower groups. It can be applied to the parent-child hierarchies. The examples of the application of taxonomy are animate objects, places, concepts, events, properties, and relationships.

The advantage on using a hierarchical classification is efficient in both learning and representation. Each sub-problem is smaller than the original problem, and it is sometimes possible to use a much smaller set of features for each.

4.2.2 Tagging

Tag is a non-hierarchical keyword or term assigned to a piece of information. It is generally chosen informally and personally by the item's creator or by its viewer, depending on the system.

4.2.3 Hierarchy vs. Tagging

Hierarchy gives a better structure and efficiency in searching. It also has a high consistency. With hierarchies user is limited to one category in which to place information. Classifying the NFC tag using location is not satisfied. For example, some may want to look for restaurants; others may want to look for teaching buildings of Medicine. Depends on the usage, Hierarchical classification does not suit this kind of query. However, tag can fit into many ‘categories’ and the results are much more contextual.

5 Overview

In the previous semester, we have brought up an idea in letting people to get some useful information, comment and vote on some physical environment. Base on this idea, we have developed a social app, Cucom. So, the users can comment and voting on the CTag (an NFC tag encoded with some information) located in different places on The Chinese University of Hong Kong through our app.

We decided to deploy the app but we have discovered some weakness:

1. The usage of Cucom is not flexible enough. The usage of our app is limited on some kind of tourism app for The Chinese University of Hong Kong. Actually, this idea can be applied in different situation. For example, if this idea applies on exhibition, the NFC tag can be used for displaying some additional information that cannot simply use poster to present it. Or if applies in a company, the NFC tag can be used as reporting issue like reporting lack of printing paper. Therefore, we need to increase the flexibility of our app so that administrator can make use of our app to apply in different situation.

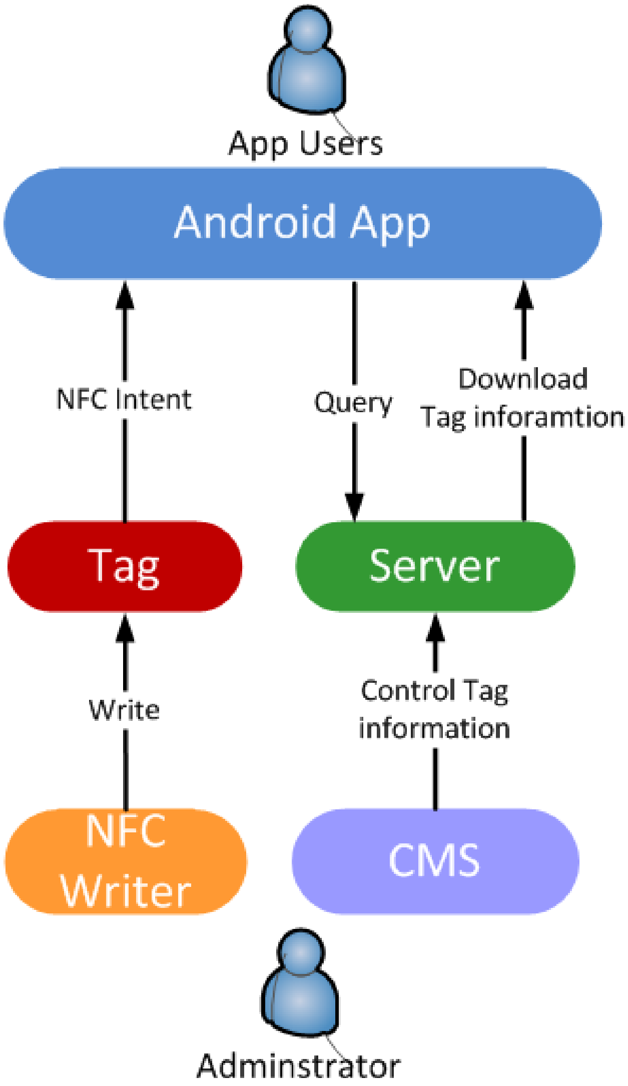
2. Cucom is not customizable. The layout of each tag is fixed. The fragments in each tag are fixed to be comment and voting pages. If we want to increase the

flexibility and applies our app in different situation, the voting or comment page may not necessary in some tags. For example, the tag for reporting issue in company may not need to include voting page. Hence, we have to improve the customizability of the app in order to let administrators design the solution for their case.

3. The changes of app are not updated in the app synchronously. If the app is deployed, users must update the app in the app store to download the most current version. In some situation like exhibition, the administrators may modify the layout in the last second and hope that the visitor can have the newest version of app. But it depends on whether the user update app or not. If the users do not update the version, it may introduce some version inconsistence. So, we have to consider this issue and solve the problem.

4. It does not have any content management system for administrators to support modification of the app. We need to change the information in the database directly without a content management system. In this way, it is not user friendly for administrators who does not have any database administration experience before. Thus, we need to develop some solution that provides a simple user interface for administrators to modify the app.

Initially, there are numbers of empty tags and the administrator will use the NFC writer to arrange them by writing the tag id on it. After assigning the tag id to the tags, administrator will add the tag information into server through the CMS. They can also use the CMS to control the tag information afterward. When the tag and the database are ready, users can start using our app. Each time when the app is opened, the app will query the database to check if its version is up-to-date. If the version is outdated, it will update the version by downloading all the new tag information from the database. Once the app users enable the NFC function in their mobile and scan the tag, the app will get the tag id and display the corresponding information to them.



So, we can see Latia mainly includes four parts: Server, Android app, Tag, and CMS. Each of them will communicate with others to transfer different information for their jobs. Their purposes are shown as follow:

5.2 Server

The server aims to manipulate the information that related to the Tag. In order to achieve this purpose efficiently and effectively, we use a server to be our database. Since the server is easier for the synchronization on mobile devices, it gives the app users the update-to-date information. Also, it provides larger memory storage space than that of the mobile phone, which is comparatively small. Besides, the server ensures reliability. If any Tag is damaged, malfunctioned or lost, it can be replaced by a new one and retrieves the information from server.

The server divides into two parts, database and interface. The database is used for storing the Tag information, statistic of voting, layout, and resources. Behind this we needs an interface to handles the action on the database, so that client side can query and update the database through the interface.

5.3 CMS

The main usage of CMS is to provide a simple equipped user interface to the administrator for supporting manipulation on tag or layout information, upload resource, and update version. It will query the server to make changes base on the administrators' request. The CMS needs to provide a convenient way for administrator to modify the information in the database, for it we build it in form of website. Hence, it can be used without installing any package and administrators can do the modification in everywhere with internet.

It divides into two parts, frontend and backend. The frontend provides the user interface for administrators to get the tag information, layout information, resource, or version. Then, it will pass the information to the backend. The backend will connect to the database and query it base on the information.

5.4 Android App

The android app has three modules, NFC, updating and layout rendering.

Latia NFC Tag will be located in different places and objects where people will visit, for example, restaurant, posters and Lecture Theater etc. After the users installed Latia, the users need to enable the NFC in their mobile setting. Then, they can scan the NFC tag whenever they want to interact with the object represented. The app will be opened automatically when detecting Latia NFC Tag and read the information inside the tag.

The action that a user can do is completely determined by the tag. For example, if the tag represents a study room, user can book the room through Latia. If the tag represents an event, by attaching the mobile phone user can add the event to the calendar automatically. If the tag represents a restaurant, user can get a ticket and queue up for a seat. Latia becomes an interface between user and object. It promotes the power of NFC to an extent that user feels very intuitive to use it.

5.5 XML

In order to make the layout flexible, Latia downloads and renders the layout in runtime. Designers only need to update the layout remotely. After user attached to the NFC tag, Latia automatically looks for the list of layouts corresponds to the tag. Each layout will be rendered on the screen with customized images, text and button functions.

5.6 Tag

Tag is an NFC Tag which have assigned the ID that used by Latia. Each Tag represents a physical object or environment, which is specified by the administrator. It acts like a small database. The administrators can use the NFC Writer to write ID on it and on the other hand, the app can read the ID and display the relevant information.

The administrator can arrange the Tags to suit their needs. For example, if our app applies on the Expo, each tag can represent an exhibit. The administrators can set numbers of layout and information shown by the tag to present the exhibit. Also, they can add a voting button in the tag layout to collect the visitor reviews. Once the users scan the tag using our app, it can retrieve the related information efficiently without using network, as long as the NFC is on.

5.7 Logo Design

To impress user at first glance, we have design an eye-catching logo for our application - Latia. The emblem contains the apps name, a smartphone, an NFC signal and some colorful liquid.

The liquid splashes from the smartphone screen signify user can have an innovative and fruitful experience by attaching the smartphone to the NFC tag through our application.

6 LATIA DESIGN

6.1 Server

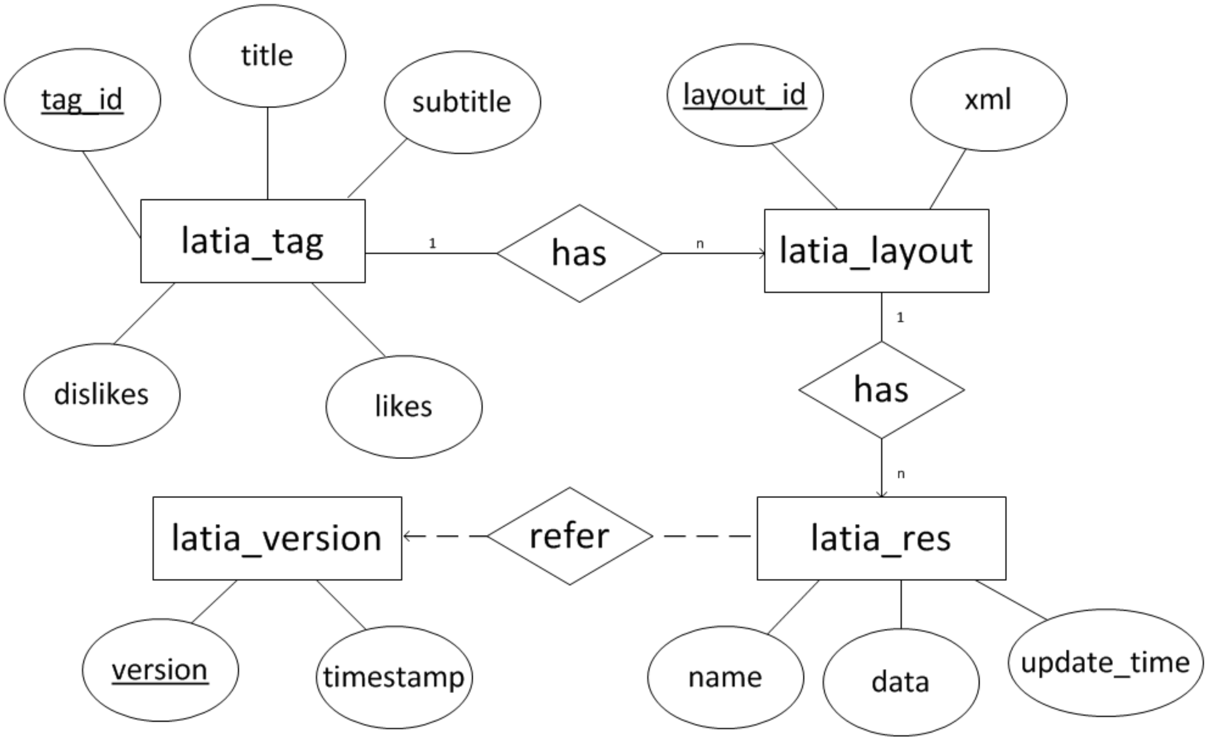
The server divides into two parts, database and interface. The database is used for storage while interface is used for query the database according to the request of the app.

6.1.1 Database

To increase the flexibility, Latia will get most of the information from database. The information is about the tag information, basic layout of each tag, current version and admin information.

The tag information includes the title, subtitle and number of likes/dislikes of the object that be represented. The basic layout includes the number of pages, the xml of each page and some other resource like image. For updating the application, modified time of resource needs to be record in order to differentiate the outdated data. For the CMS, administrators need to login the website in order to manage the database. The admin information includes username and password.

ER Diagram


tag_id
title
subtitle
layout_id
xml
latia_tag
1
has
n
latia_layout
1
has
n
dislikes
likes
latia_version
refer
latia_res
version
timestamp
name
data
update_time

Scheme

tag_information(tag_id:Integer, title:String, subtitle:String, likes:Integer, dislikes:Integer)

layout(tag_id:Integer, layout_id:Integer, xml:String)

resource(name:String, data:Binary, update_time:Date/Time)

version(version:Integer, timestamp:Date/Time)

admin(username:String, password:String)

The tag information table stores the title and subtitle of the object that be represented, the voting statistic (the number of likes and dislikes).The layout stores the xml layout for each page on the tag. The tag_id is created so that the application can display the pages for each tag by joining tag information and layout table using it. Besides, the layout_id is added for indicating page order.

The resource stores the name of resource, data. For updating the data on the mobile application, the field update time is added in the resource table. Also, the version table is created to store the version number and the timestamp of releasing the version. Then, the application can compare its version number and current version number and check if update is needed. The admin table stores the username and password for login validation.

The tag information table

The tag information table includes tag_id, title, subtitle, likes and dislikes. Each NFC tag contains a Tag ID. So, the tag_id in this table indicates which tag the information belongs to. The title and subtitle is the information of object that be represented. The likes and dislikes store the voting statistic of object that be represented.

The data type of tag_id is 11-digit integer so that the application can support at most 10,000,000,000 tags simultaneously. The title and subtitles use string to store the text information. Similar to tag_id, likes and dislikes use integer to store the voting statistic.

Attribute	Description	Data Type
tag_id	The Tag id used by Latia	Integer
title	The title of the object	String
subtitle	The subtitle of the object	String
likes	The number of likes	Integer
dislikes	The number of dislikes	Integer

The layout table

The layout table includes tag_id, layout_id and xml. Each tag can have more than one page in the layout. So, the layout_id indicates the display order of pages when the apps users flip over the screen. The xml store the code using our own defined syntax for displaying the layout.

Similar to tag_id, the data type of layout_id is 11-digit integer. The data type of xml is String to store the xml code written in our own defined syntax.

Attribute	**Description**	**Data Type**
tag_id	The Tag ID used by Latia	Integer
layout_id	The Layout ID indicate display order of pages	Integer
xml	The subtitle of the object	String

The resource table

The resource table includes name, update_time and data. The name is the name of uploaded resource. The xml includes the name of resource, for example, image. So, the application can refer the name of resource and download. The data is the binary data of the resource. The update_time records the modified date and time.

After the administrators updated the resource, the update_time will be updated to indicate if the application needs to download the updated data.

The data type of name uses String to store the filename. The update_time use Data/Time data type to store the time that accurate to a second. The data type of data uses binary to store.

Attribute	Description	Data Type
name	The name of the resource	String
update_time	The modified date and time	Date/Time
data	The data of resource	Binary

The version table

The version table includes version and timestamp. The version is the version number. The timestamp is the date and time that release the corresponding version. So the application will check the timestamp. If its own version timestamp is older than the current version timestamp, the application will check it and download all the resource between these two timestamp.

The data type of the version is integer. Similar to the update_time, the timestamp uses Data/Time data type to store the released time that accurate to a second.

Attribute	Description	Data Type
version	The version number	Integer
timestamp	The released date and time	Date/Time

The admin table

The admin table includes username and password. When the administrators want to use the CMS to monitor the tag, they need to login using the username and password. So, the admin table stores the login information and CMS can have validation. Also, the password is encrypted to increase confidentiality.

The data type of the username and password is string.

Attribute	Description	Data Type
username	The username for login	String
password	The encrypted password for login	String

6.1.2 Interface

The interface is used to query the database and returns the relevant data from database to app. The interface involves six activities: get tag overview, get tag information, get layout information, get resource overview, get resource data, get vote, and update vote.

Get Tag Overview

Get Tag Information

Get Layout Information

Get Resource Overview

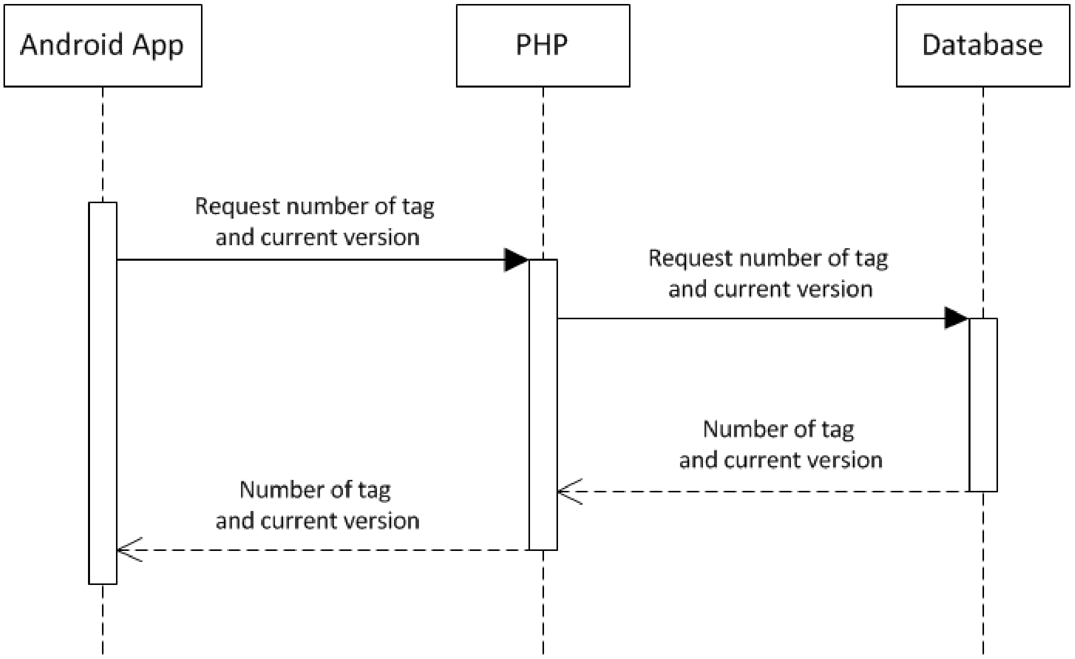
Get Resource Data

Get Vote

Update Vote

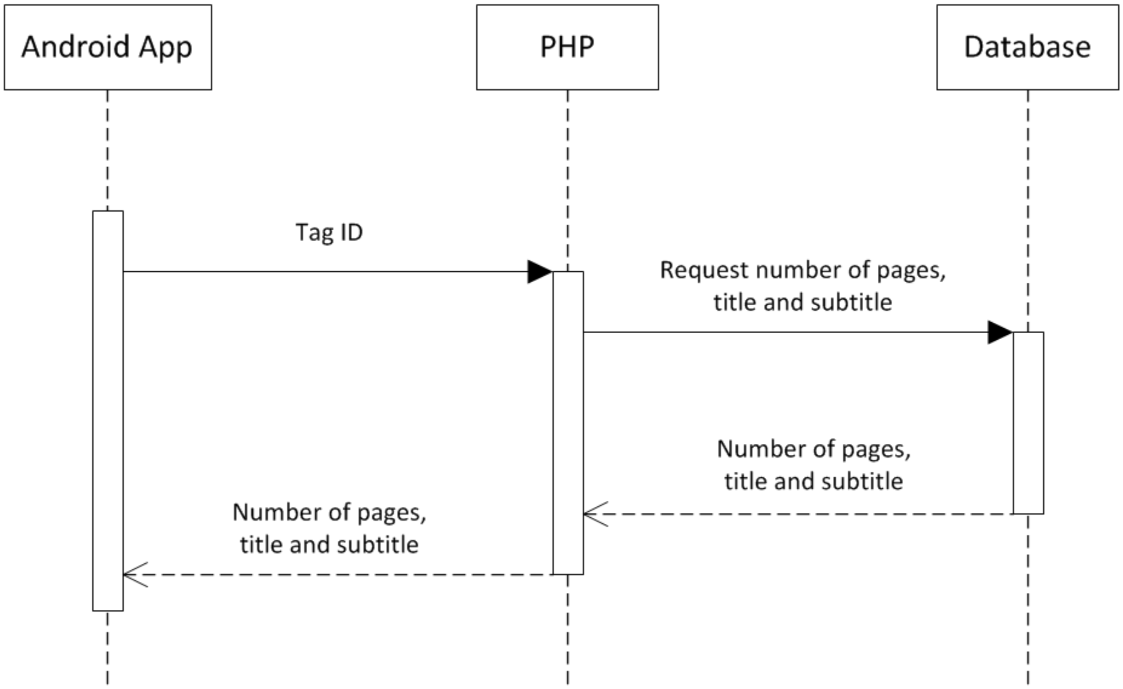
Get Tag Overview

When the app is opened, it will send a request to get the number of tags in order to use it for downloading the xml layouts for each tag. Also, it will send a request to get the current version to check if there are any resources that need to be updated. After receiving the request of getting number of tags and current version, the PHP will query the database to retrieve the overview. If it is success, PHP will return the results to the app.


Android App
PHP
Database
Request number of tag and current version
Request number of tag and current version
Number of tag and current version
Number of tag and current version

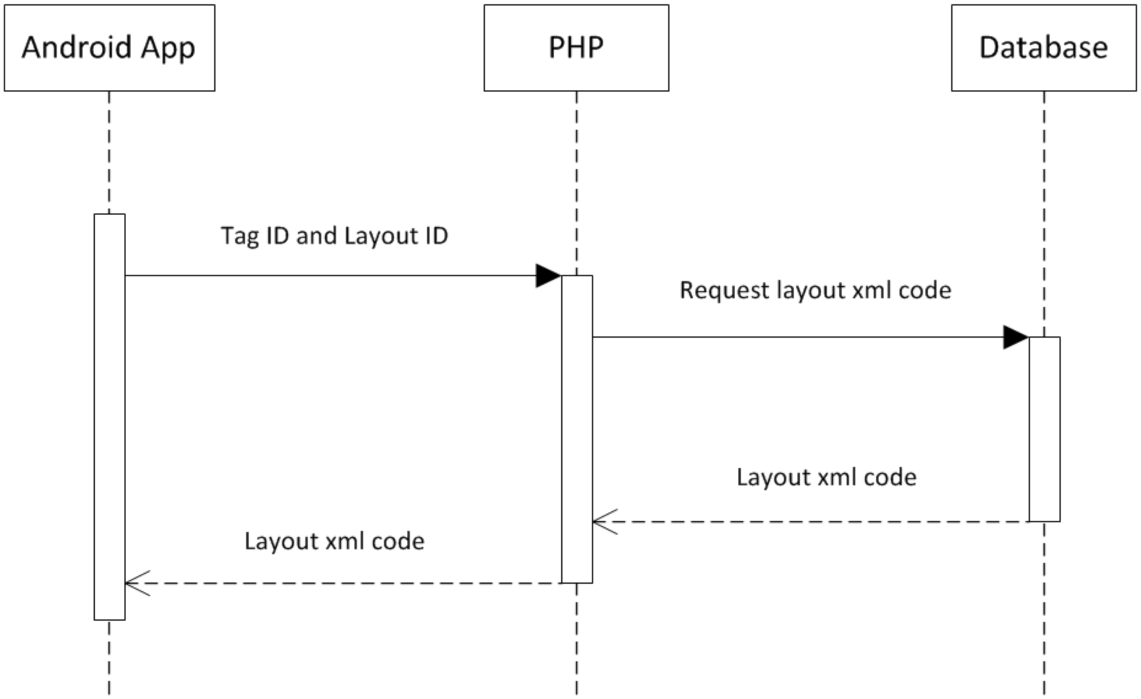
Get Tag Information

After the Get Tag Overview activity, the app will receive the number of tags. The app can use it to generate the Tag ID. For each Tag ID, the app sends it to request getting the tag information, number of pages, title and subtitle of the tag. After receiving the Tag ID and request of getting tag information, the PHP will query the database to retrieve the number of pages, title and subtitle using the Tag ID. If it is success, PHP will return the results to the app. If the Tag ID is not found, PHP will return null back to the app.



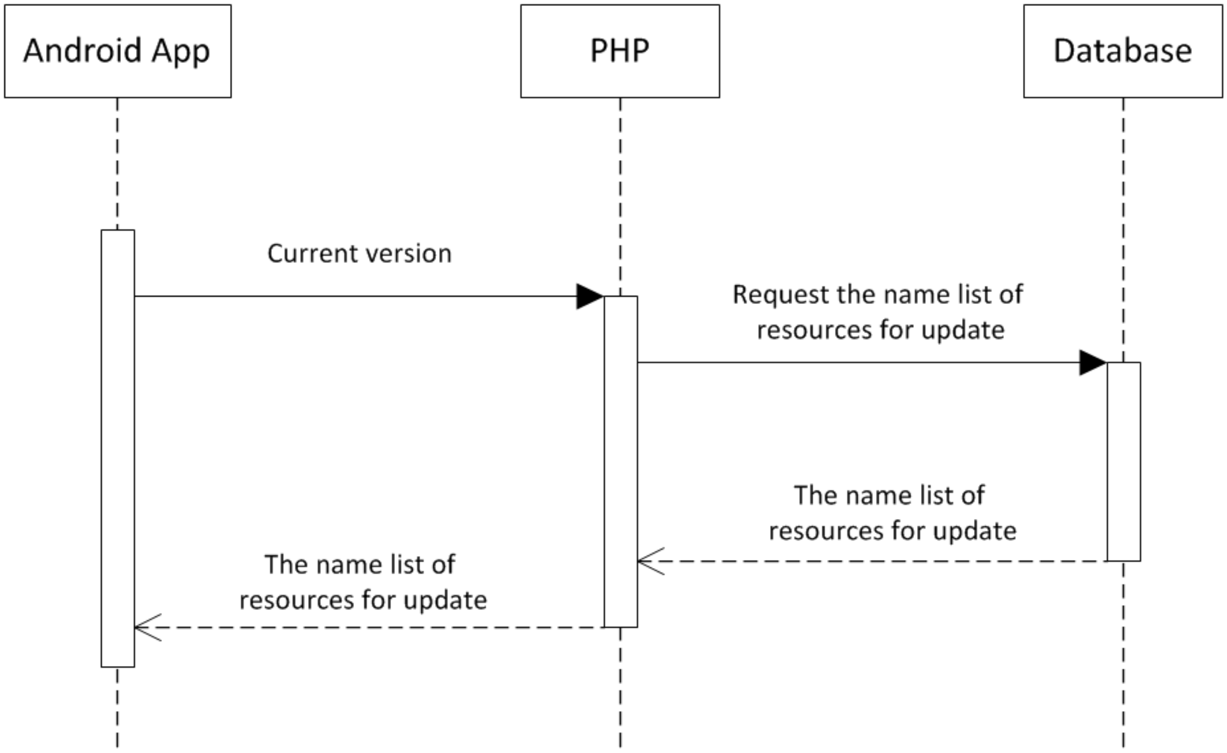
Get Layout Information

The app gets the number of pages in the Get Tag Information activity using a particular Tag ID. The app can use it to generate the Layout ID. For the specified Tag ID, each of the Layout ID will be sent to PHP and the app will request getting the layout xml code. Then, PHP will query the database to retrieve the layout xml code using the Tag ID and Layout ID after receive it. If it is success, PHP will return the xml results to the app. If the Tag ID or Layout ID is not found, PHP will return not xml error to the app.



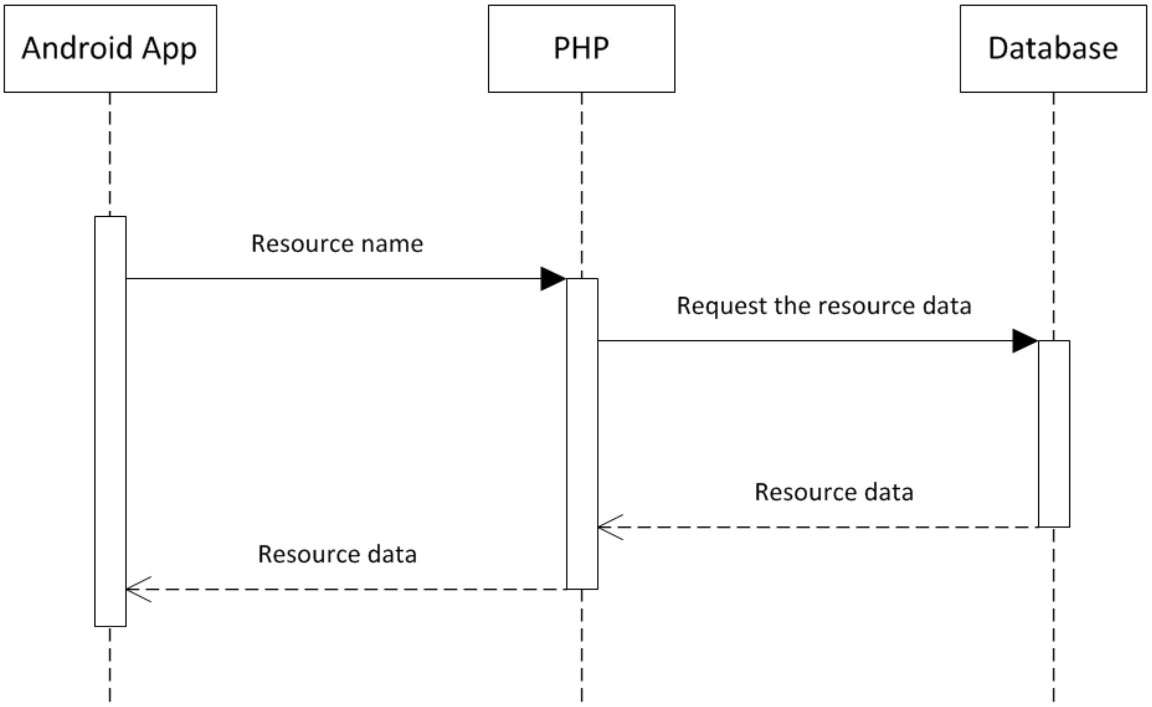
Get Resource Overview

In order update the resource, the app has to know the name list of resources which need to be updated. The app sends its own version to the PHP for requesting the name list of resources which need to be updated. PHP will use it to query the database to retrieve the name of resource which its update time in between its own version released time and the current version released time. If it is success, PHP will return the list of results to the app. If it is not found, PHP will return an empty list to the app.


Android App
PHP
Database
Current version
Request the name list of resources for update
The name list of resources for update
The name list of resources for update

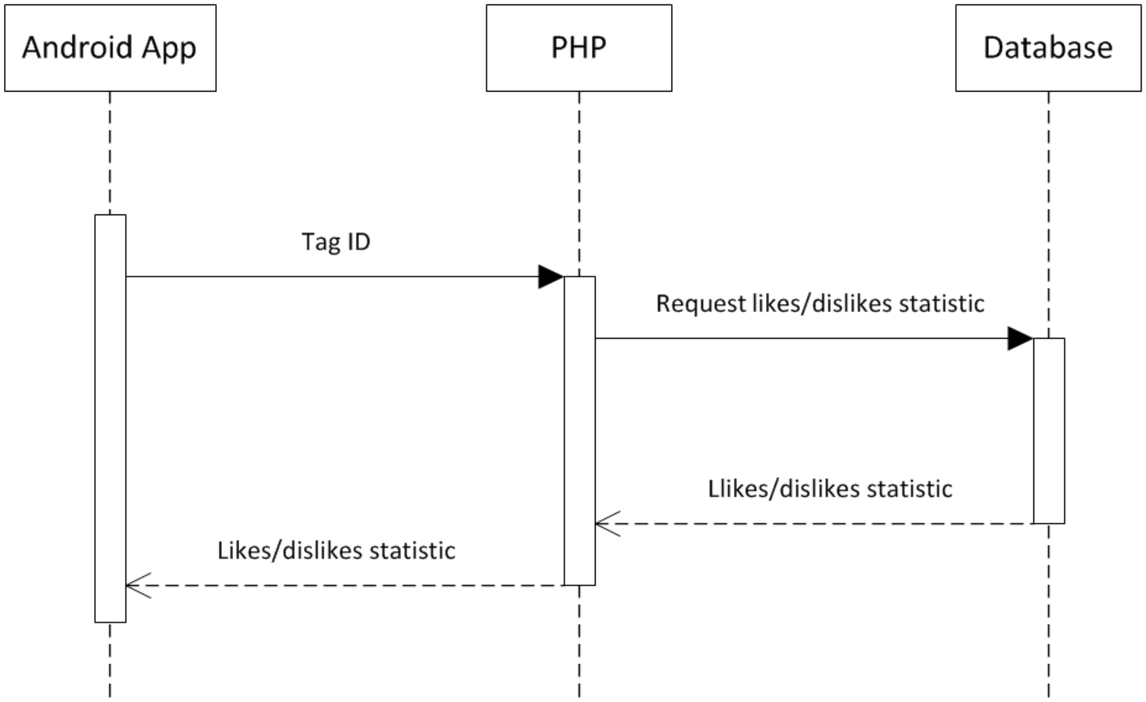
Get Resource Data

After getting the name list of resource, the app will send each of the resource names to PHP requesting the resource data. PHP will use the resource name to query the database for retrieving the resource data. If it is success, PHP will return the resource data. If the resource is not found, PHP will return null to the app.



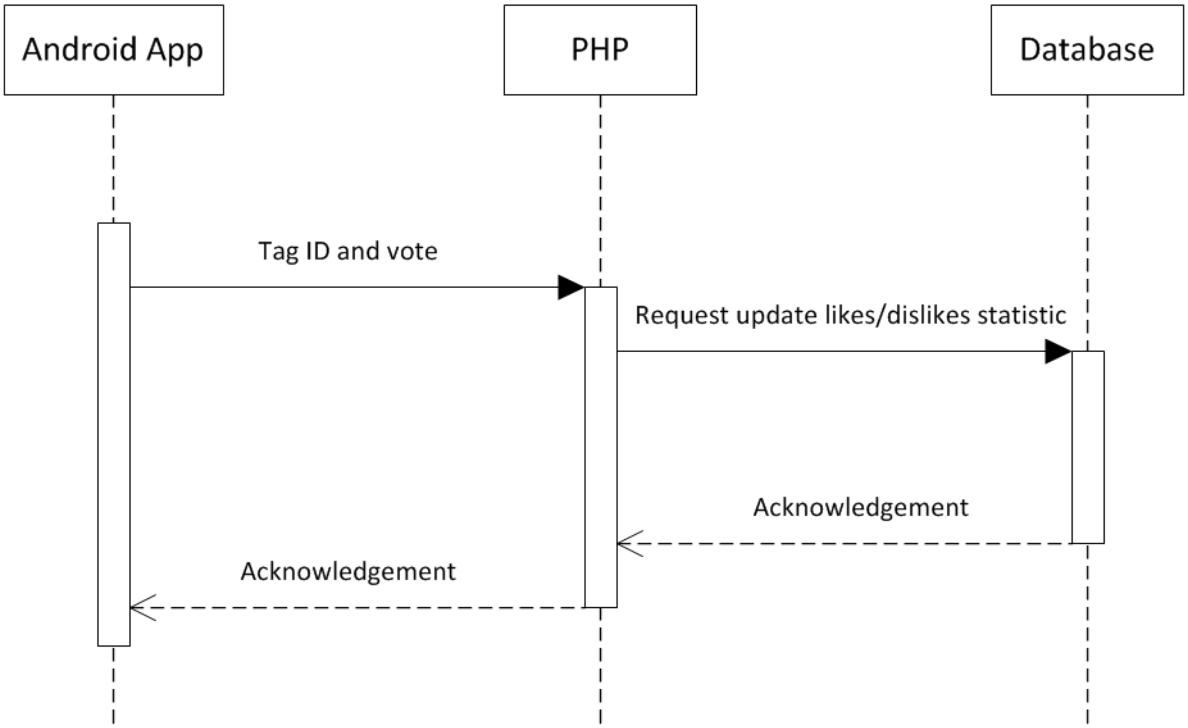
Get Vote

The app will display the vote statistic if the administrators included in layout xml code. The app will send the Tag ID to PHP that requests getting likes/dislikes statistic. Then, PHP will query the database for retrieving the likes/dislikes statistic using the Tag ID. If it is success, PHP will return the likes/dislikes statistic.



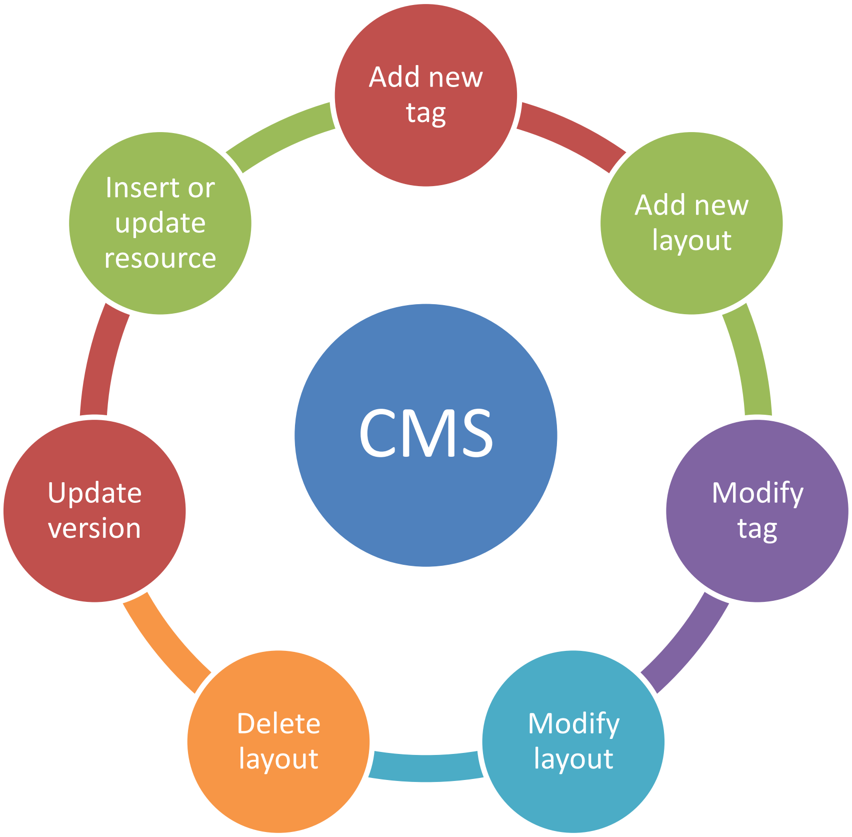
Update Vote

When the users vote on the tag, the app sends the Tag ID and vote to the PHP. After receiving it, PHP will request the database to update the vote using the Tag ID. Then, it will send an acknowledgement to the app.

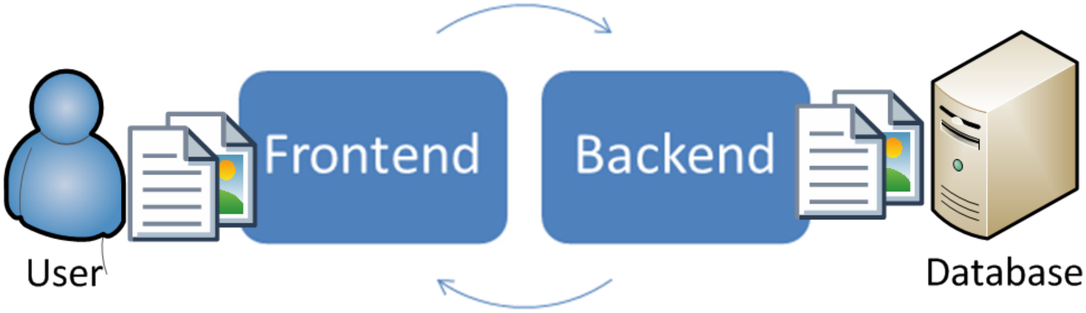

Android App
PHP
Database
Tag ID and vote
Request update likes/dislikes statistic
Acknowledgement
Acknowledgement

6.2 CMS

The CMS provide a user interface for administrator to perform seven functions cooperate with database: add new tag information, add layout information, modify tag information, modify layout information, delete layout information, insert or update resource and update version. Administrator can use these function to create a customized Tag structure. The CMS does not support tag or resource deletion. It is because the tag is supposed to be overwritten for different purpose. Also, it can guarantee the Tag ID is in ascending order so the app can search all the tag simply looping the Tag ID. For the resource, we concern about the problem if the resource is deleted but some of the layout xml code still using the resource without updated. So the CMS does not provide the tag or resource delete function with these reasons.



The CMS is divided into two parts, frontend and backend. In frontend, it provides a simple user interface for administrators to get the tag information, layout information, resource, or version to database while backend will connect to the database and transfer or update that information.



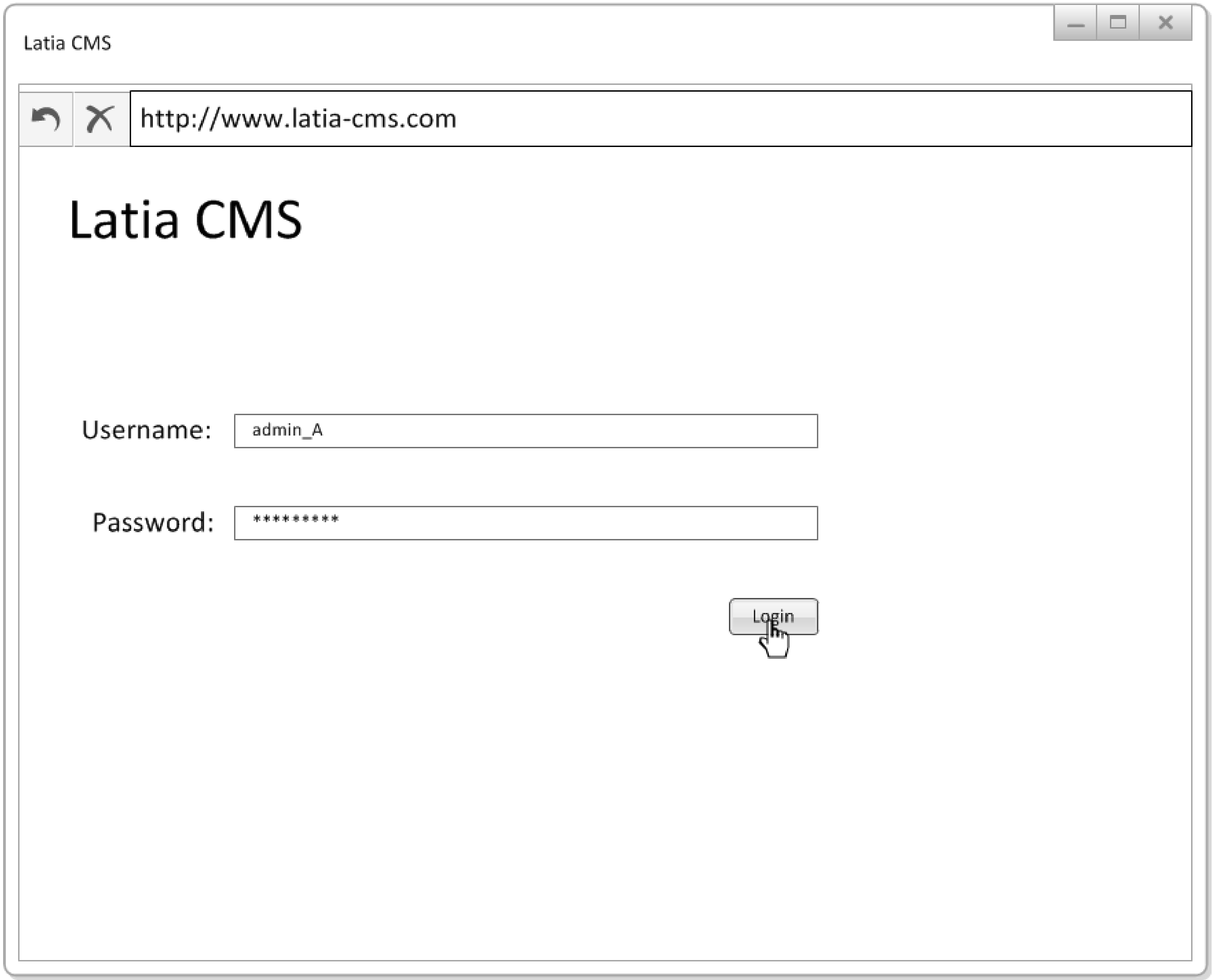
6.2.1 Frontend

The frontend will provide a simple CMS user interface to the administrator for modification with database. There will be seven main pages in CMS: Login, add tag, add layout, modify tag, modify layout, upload resource and update version.

User Interface

Login

Before letting administrators use CMS modify the information, there will be a login webpage to request administrator enter their username and password for authentication. The login page contains two textbox for getting the username and password. After administrators entered and press the 'Login' button, the information will be validated by backend process.



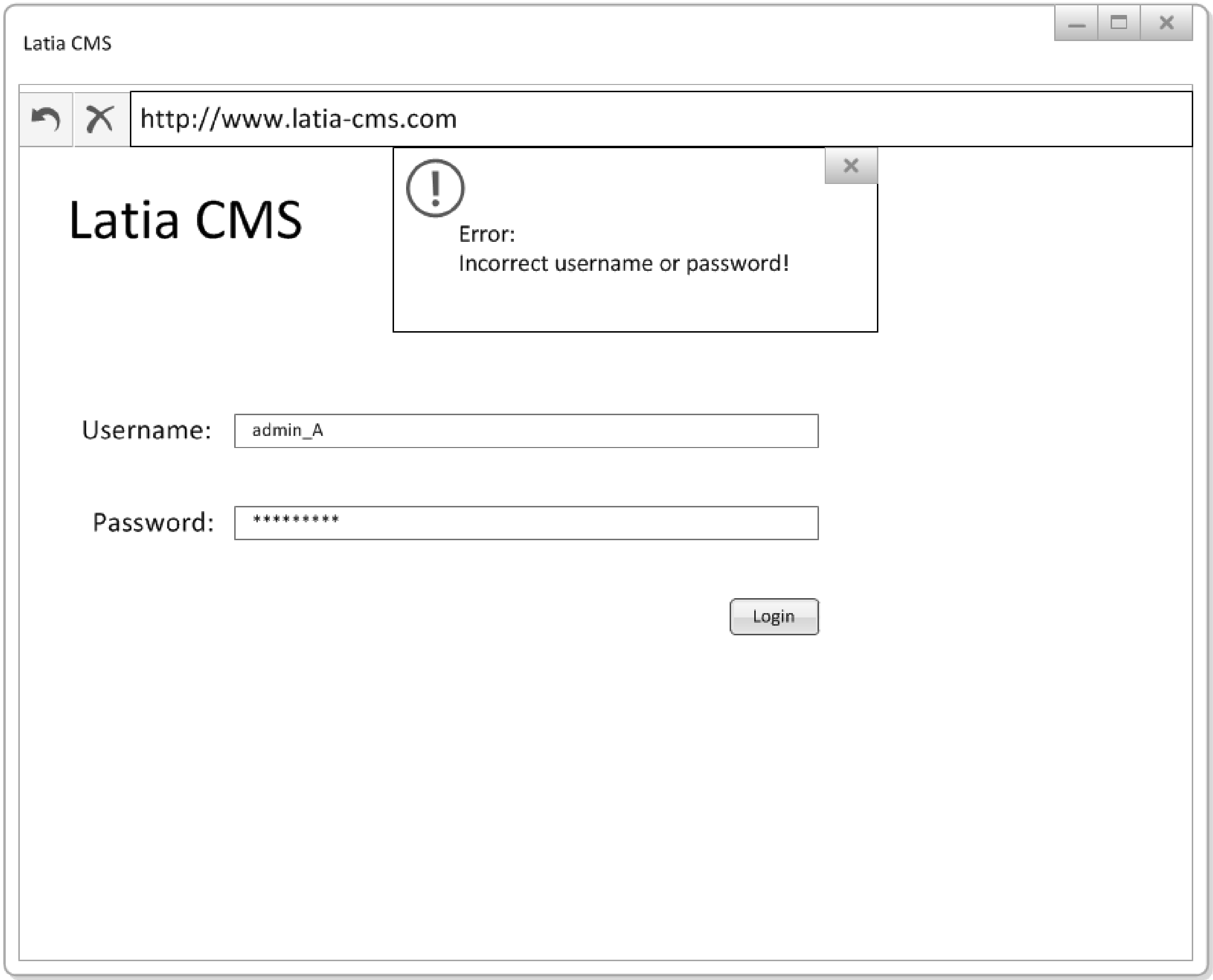
http://www.latia-cms.com

Latia CMS

Username: admin_A

Password: *********

If the administrators input the wrong username or password, CMS will pop up an error message to inform the administrators. The administrator can close the error message and enter the correct one.



Latia CMS

http://www.latia-cms.com

Latia CMS

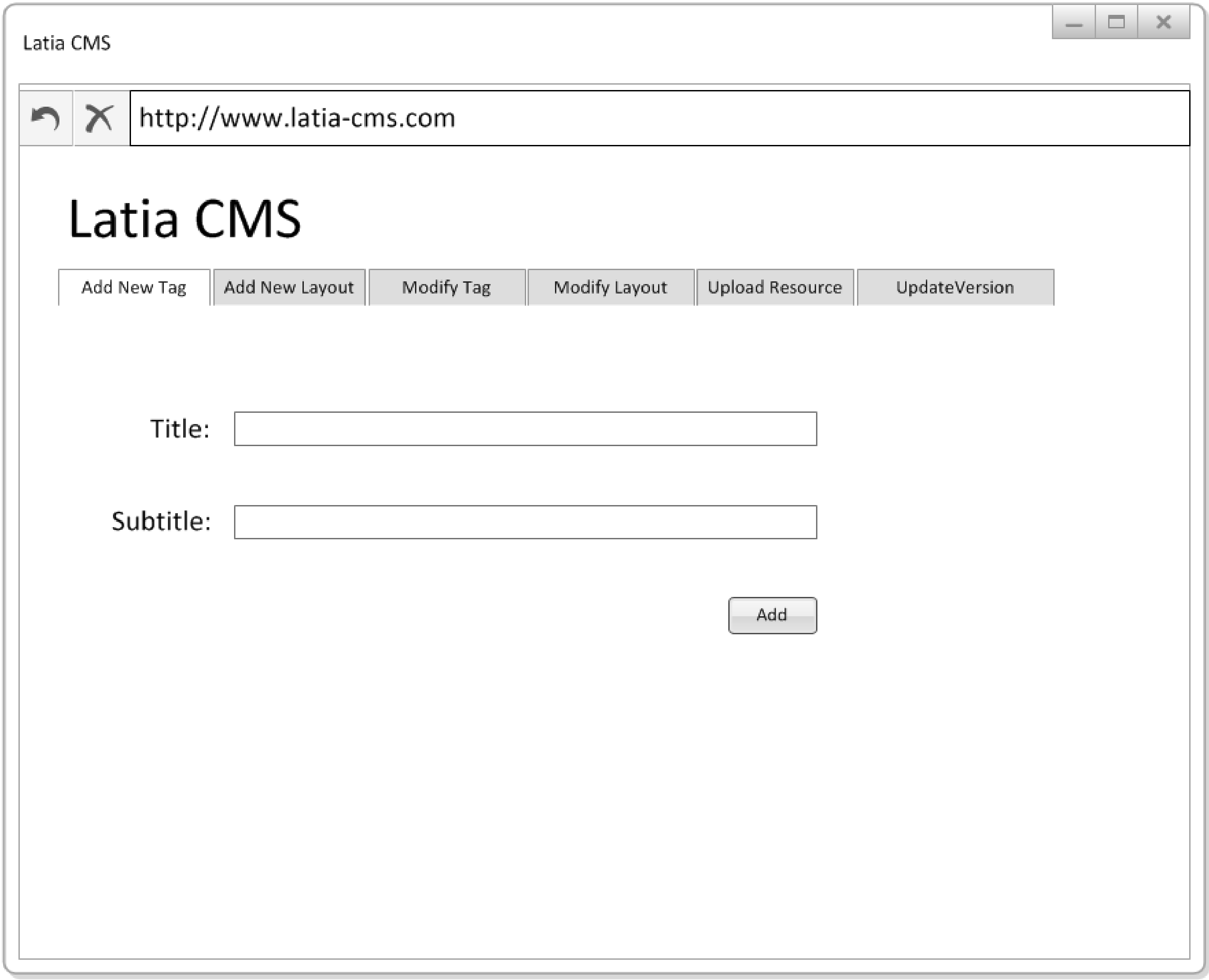
Error:
Incorrect username or password!

Username: admin_A

Password: *********

Main

After the administrators login successfully, the webpage will redirect to the main page. The main page will consist with six tab pages: add new tag, add new layout, modify tag, modify layout, upload resource and update version. The tab page design display all the function, so the administrators can easy select what function they want to perform.



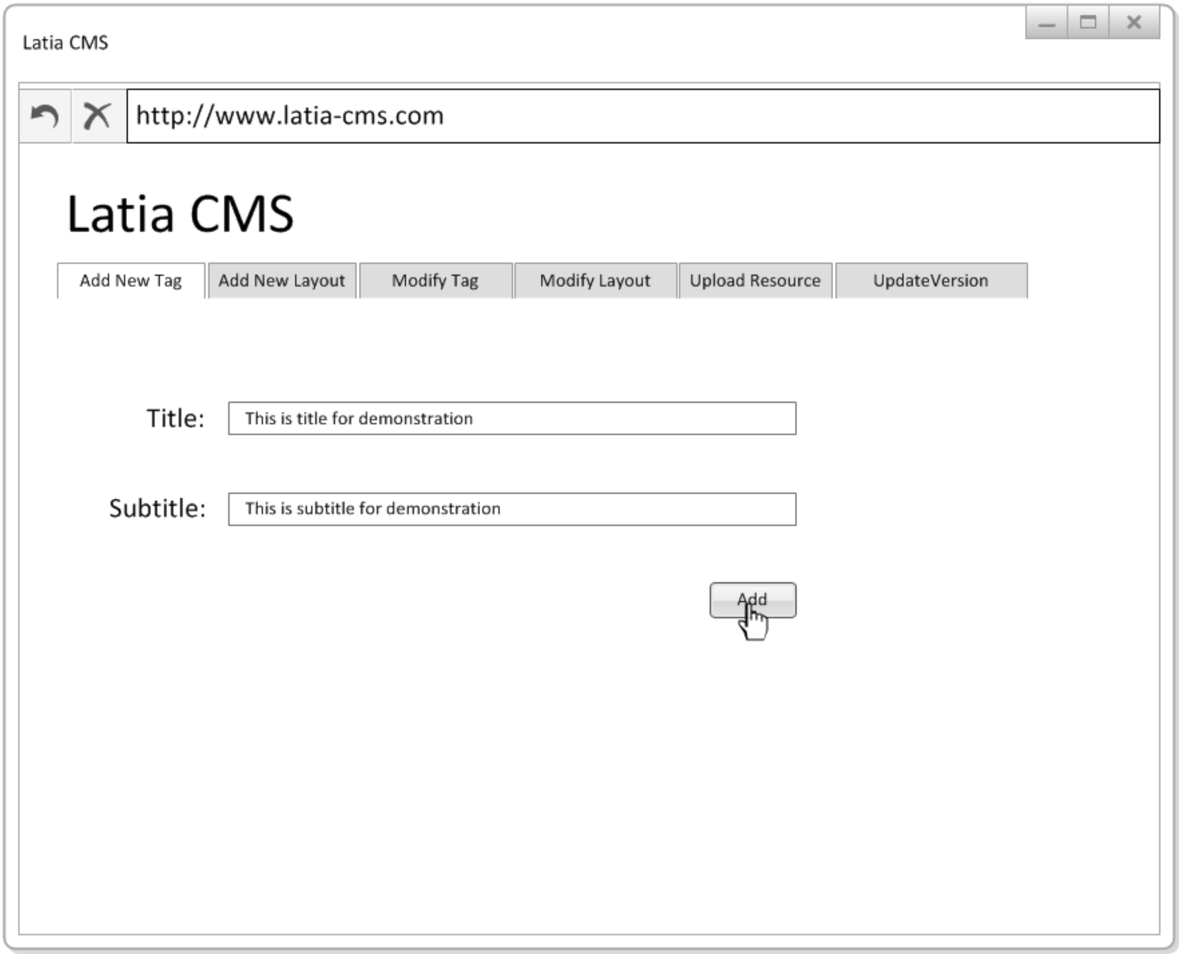
Latia CMS

http://www.latia-cms.com

Latia CMS

Add New Tag

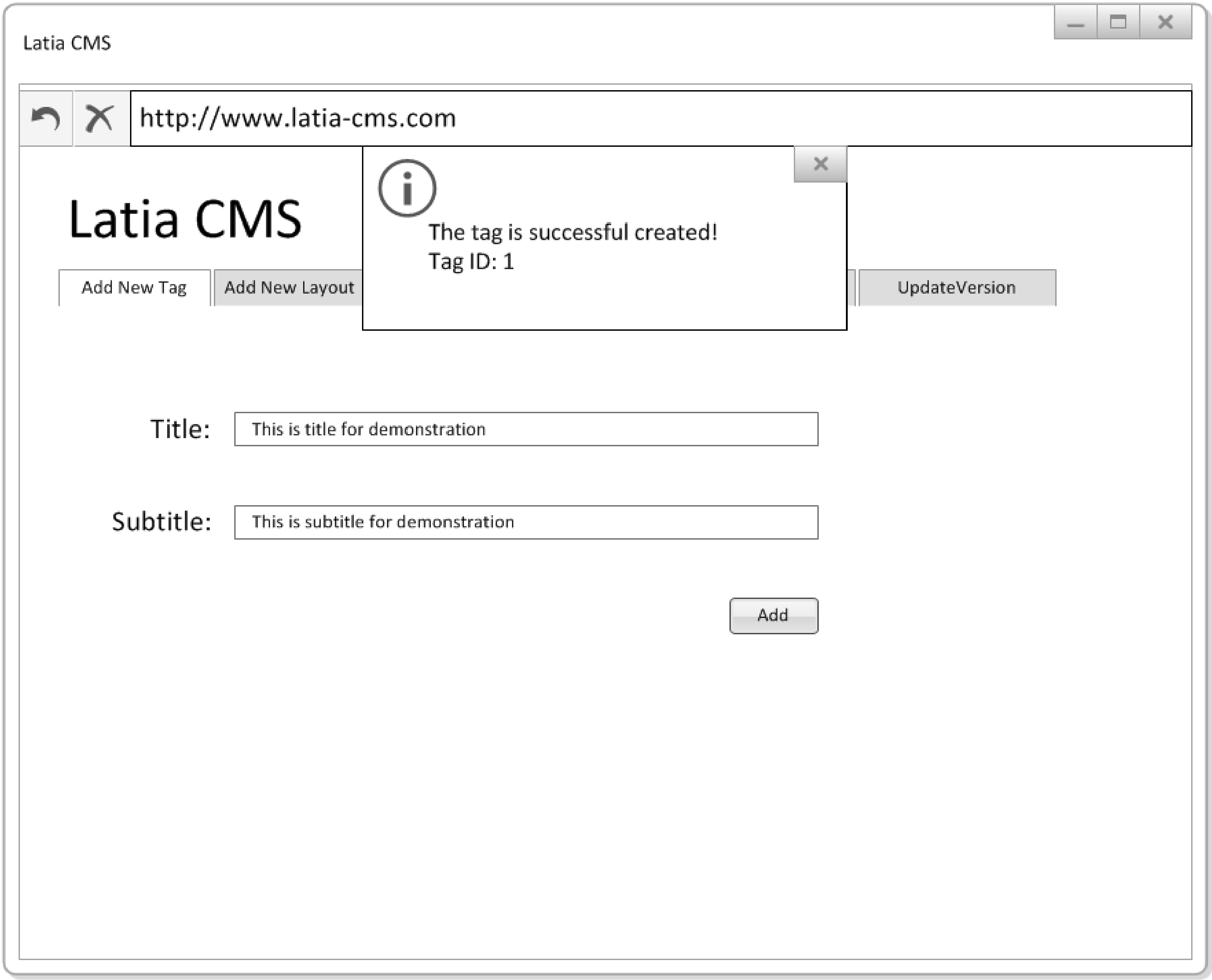
There are two textbox, title and subtitle in the Add New Tag tab page. After the administrator enter the information and press ‘Add’ button, the backend will process the insert function.



Title: This is title for demonstration

Subtitle: This is subtitle for demonstration

If the tag is successful created, CMS will pop up a message to inform the administrators. It will display the Tag ID for this new tag, so that administrator can use it for further modify the tag information in Modify tag tab page.



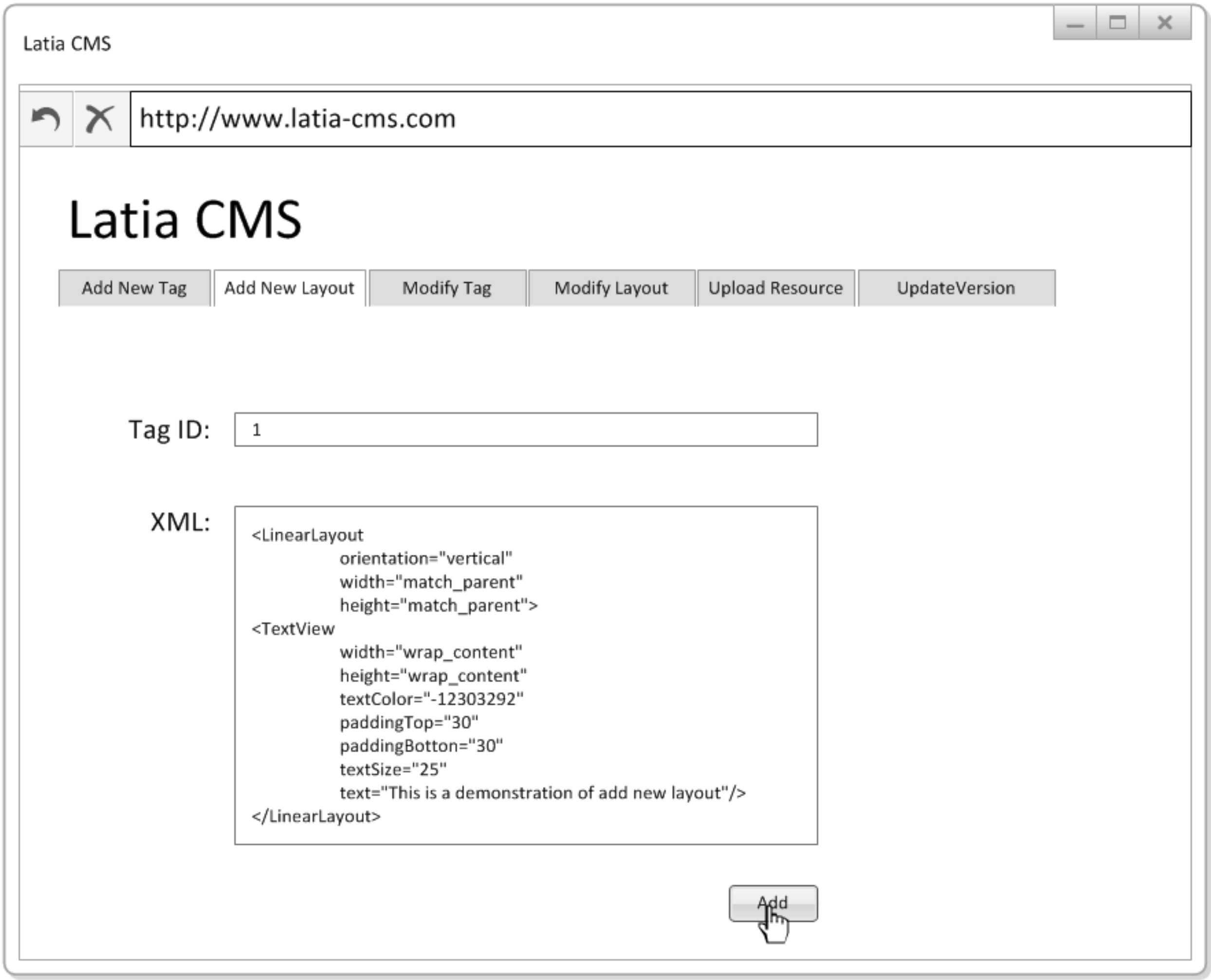
The tag is successful created!
Tag ID: 1

Title: This is title for demonstration

Subtitle: This is subtitle for demonstration

Add New Layout

In add new layout tab page, there will be one textbox for administrator entering the particular Tag ID which they want to add a new page on its tag. Also, there will be a text area for administrator entering the layout xml code. After the administrators finished entering the Tag ID and layout information, they can press the ‘Add’ button to add the new layout. The backend will get the information and insert the new layout to that tag.

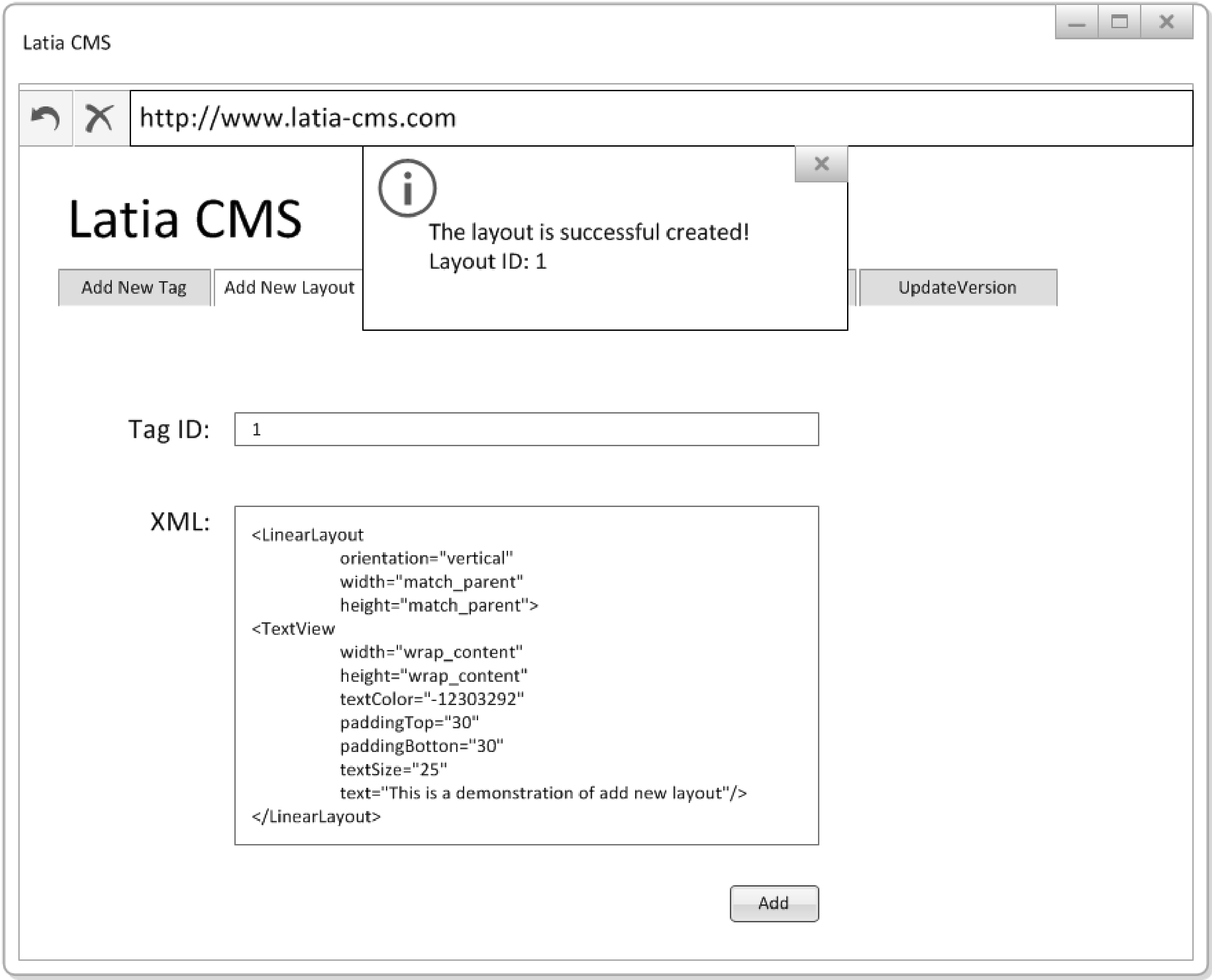


Tag ID: 1

XML:

```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">
<TextView
        width="wrap_content"
        height="wrap_content"
        textColor="-12303292"
        paddingTop="30"
        paddingBotton="30"
        textSize="25"
        text="This is a demonstration of add new layout"/>
</LinearLayout>
```

If it is successful insert, CMS will pop up a message to inform the administrators. It will display the Layout ID for that tag, so that administrator can use it for further modify the layout information in Modify layout tab page.



The layout is successful created!
Layout ID: 1

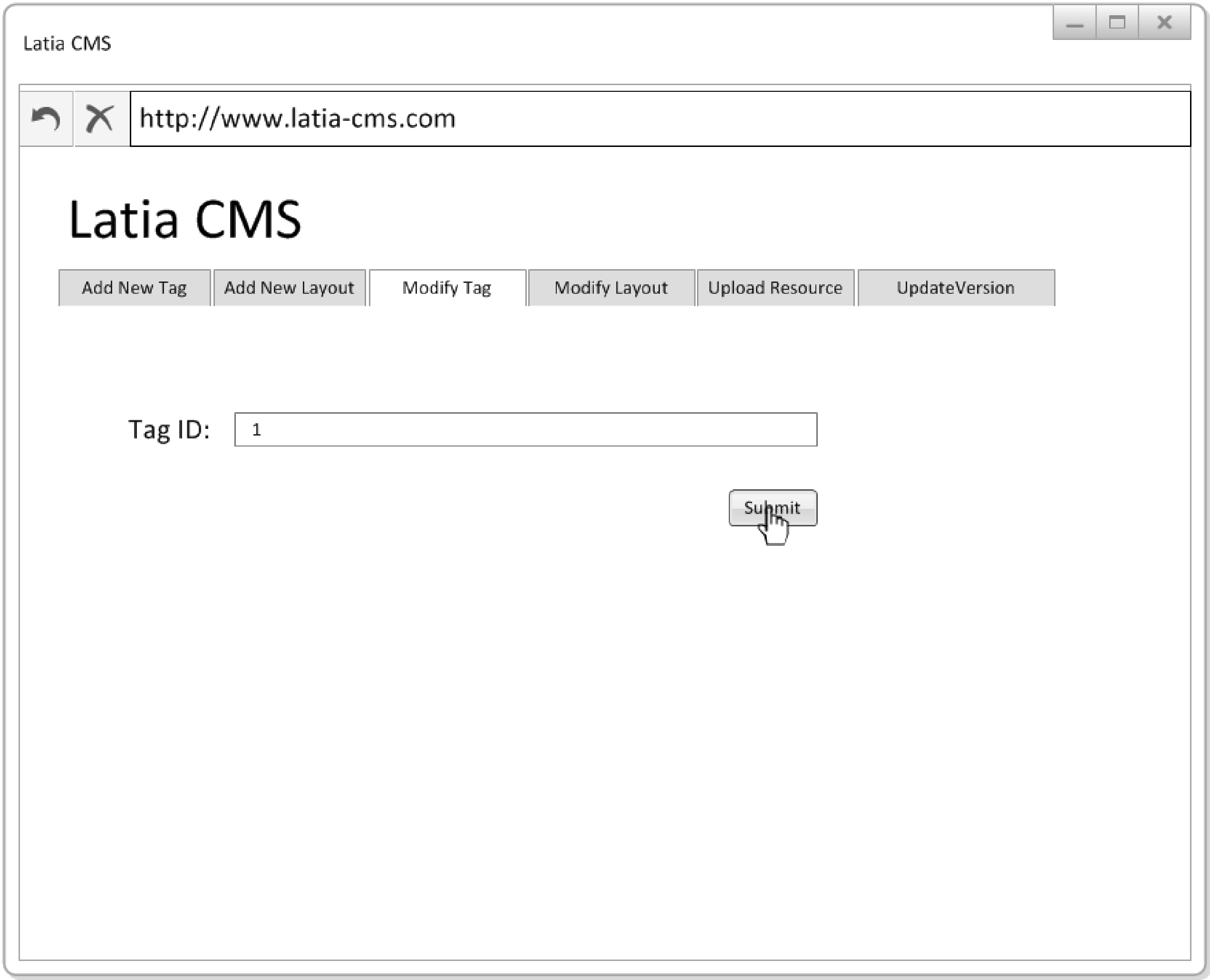
Tag ID: 1

XML:

```
<LinearLayout
            orientation="vertical"
            width="match_parent"
            height="match_parent">
<TextView
            width="wrap_content"
            height="wrap_content"
            textColor="-12303292"
            paddingTop="30"
            paddingBotton="30"
            textSize="25"
            text="This is a demonstration of add new layout"/>
</LinearLayout>
```

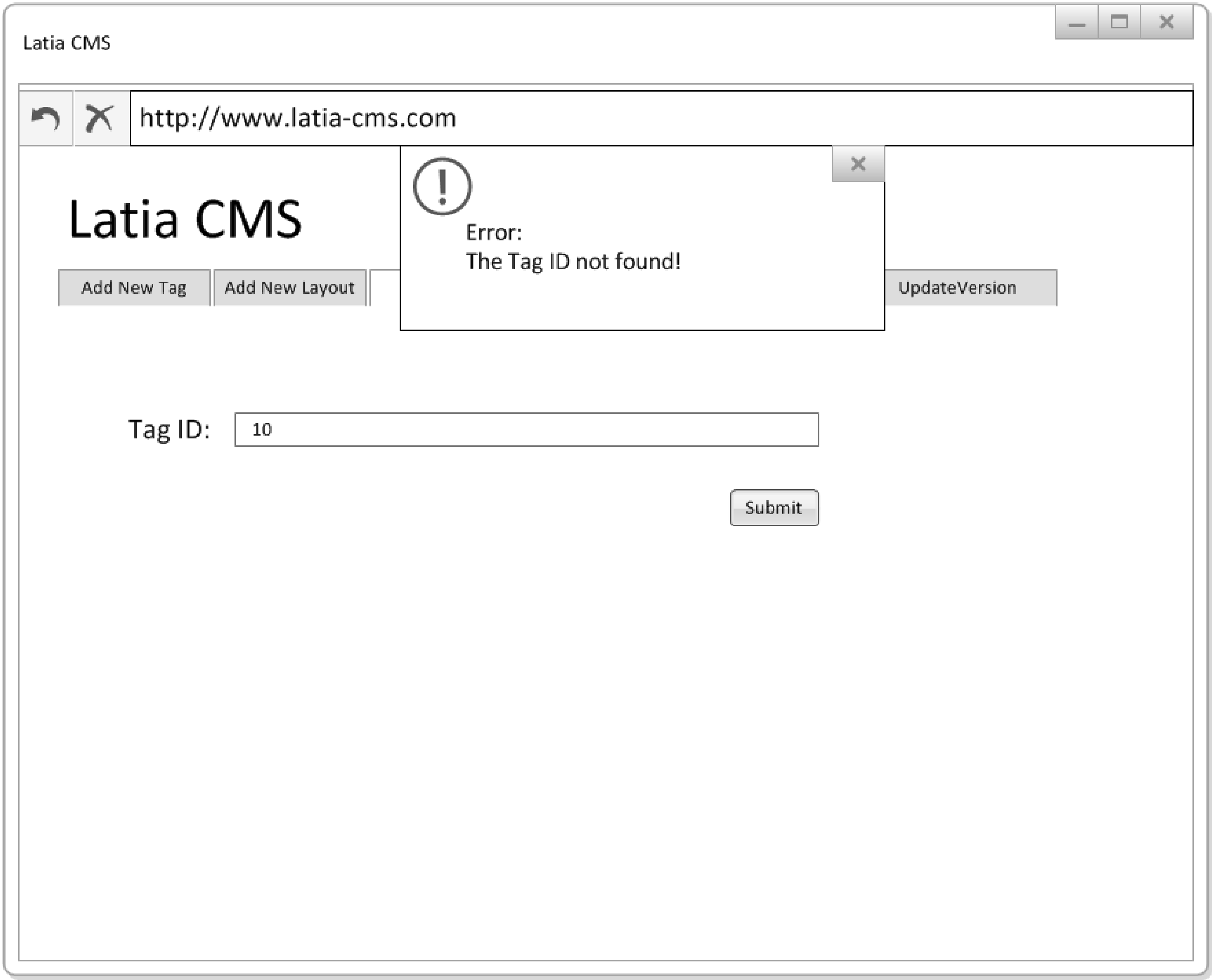
Modify tag information

In the modify tag information, the administrators need to enter the Tag ID that they want to modify. After entering it and press 'Submit', the backend will retrieve the tag information to CMS.



Tag ID: 1

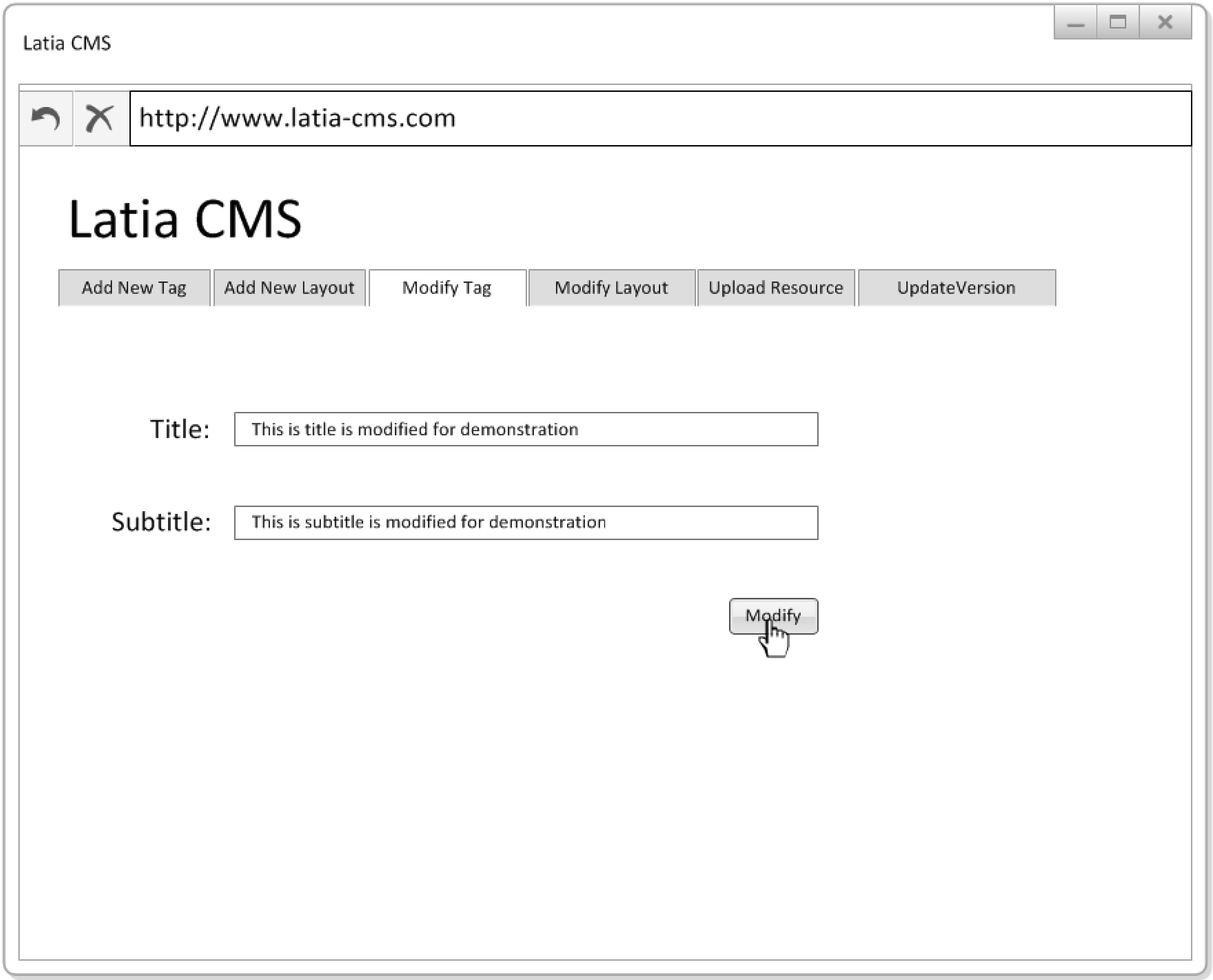
If the Tag ID is not found, the CMS will pop-up error message to inform the administrators. So, the administrators can re-enter the correct Tag ID to continue the action.



Error:
The Tag ID not found!

Tag ID: 10

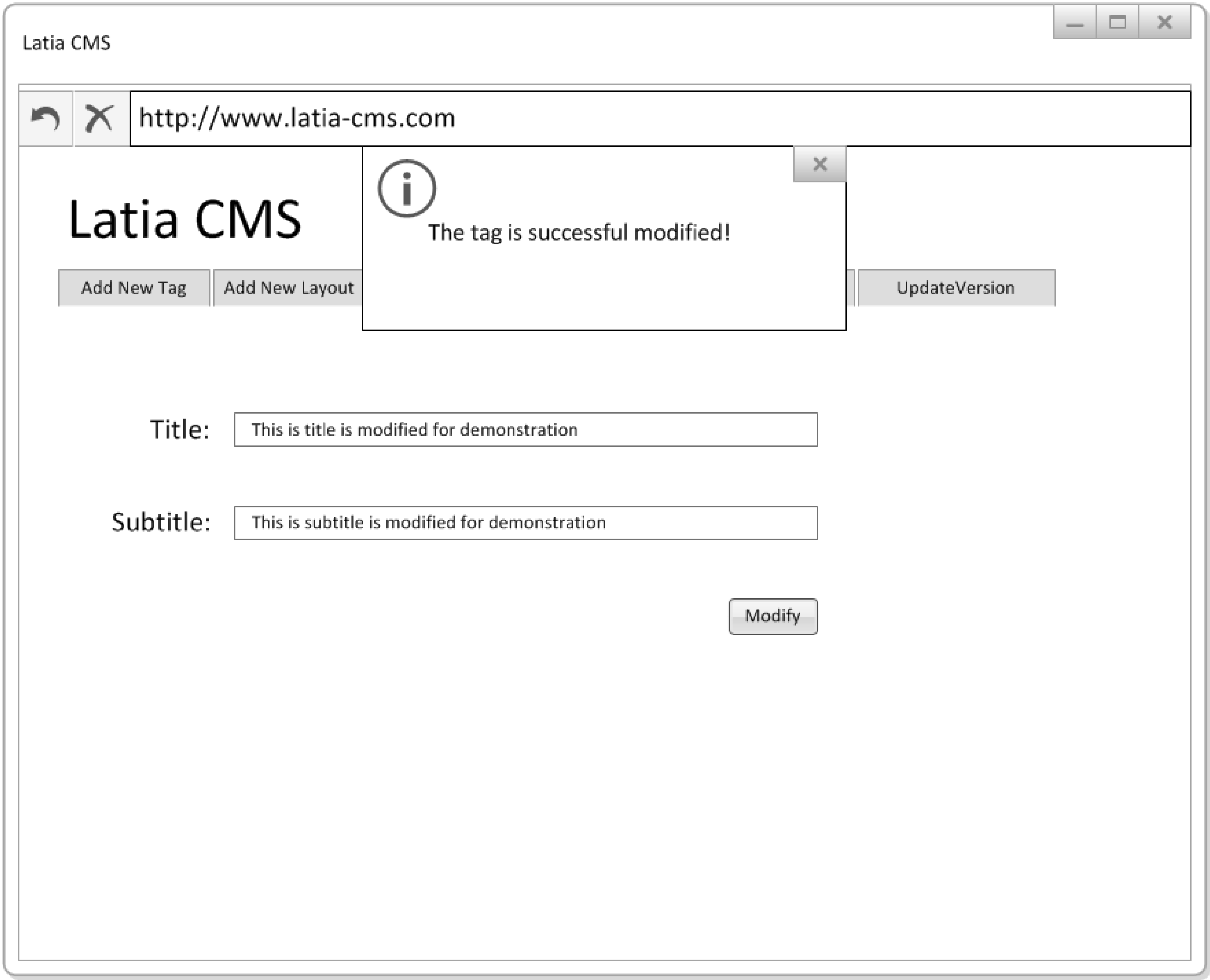
After the backend return the tag information of that tag, CMS will display the information in the textbox. So, the administrators can modify the tag information and press ‘Modify’ button. The backend will get the modified tag information and process update.



Title: This is title is modified for demonstration

Subtitle: This is subtitle is modified for demonstration

CMS will pop up a successful modified message to inform the administrators if backend process success.



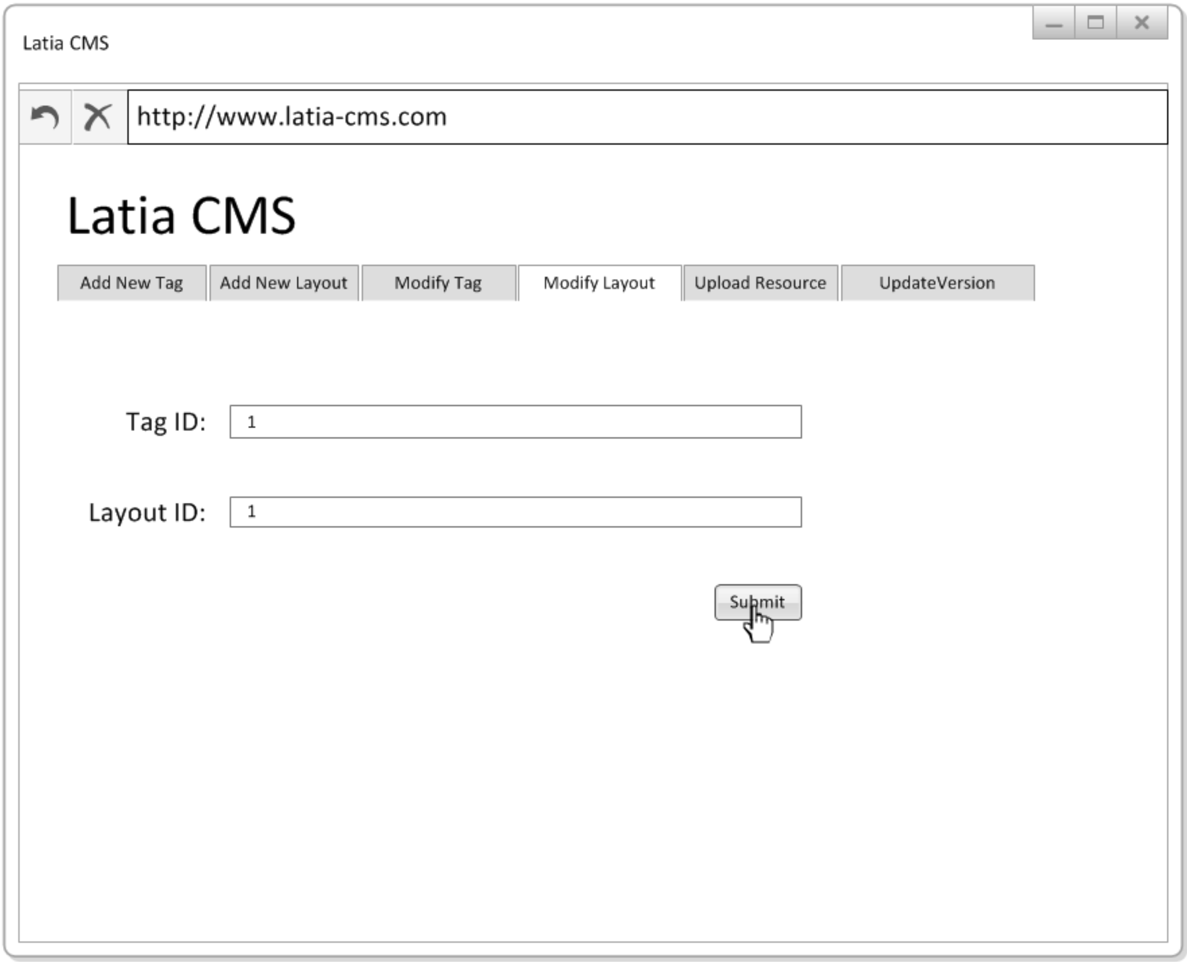
The tag is successful modified!

Title: This is title is modified for demonstration

Subtitle: This is subtitle is modified for demonstration

Modify Layout

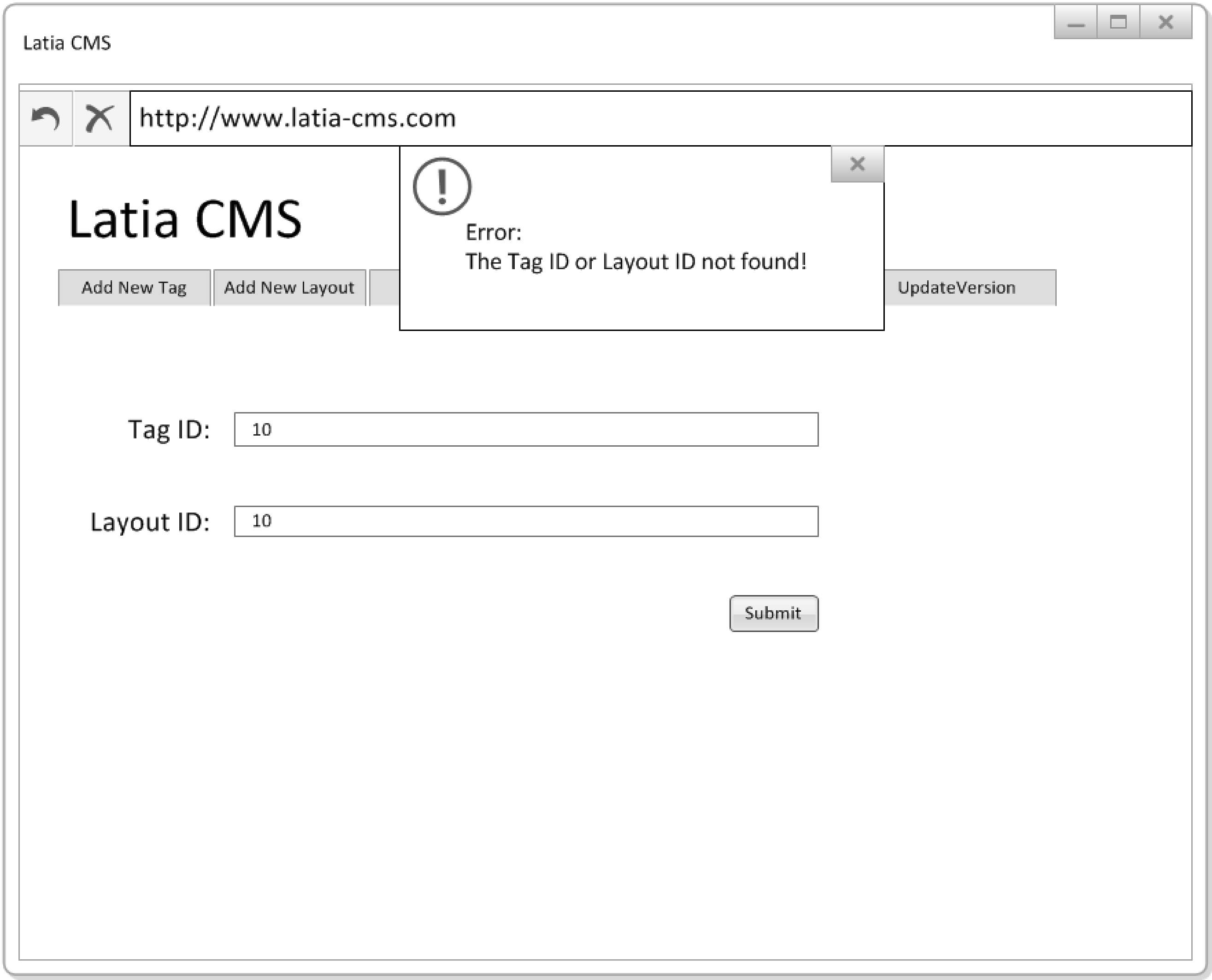
The modify layout tab page includes two textbox, Tag ID and Layout ID. The administrators can enter the Tag ID and Layout ID of the layout information that they want to modify. After pressing the ‘Submit’ button, the backend will get information and query the database.



Tag ID: 1

Layout ID: 1

If the administrators enter incorrect Tag ID or Layout ID, the CMS will get the fail message from backend. It will alert an error message to notice administrators the Tag ID or Layout ID they entered are not found. So, the administrators can enter the correct one again and continue the process.

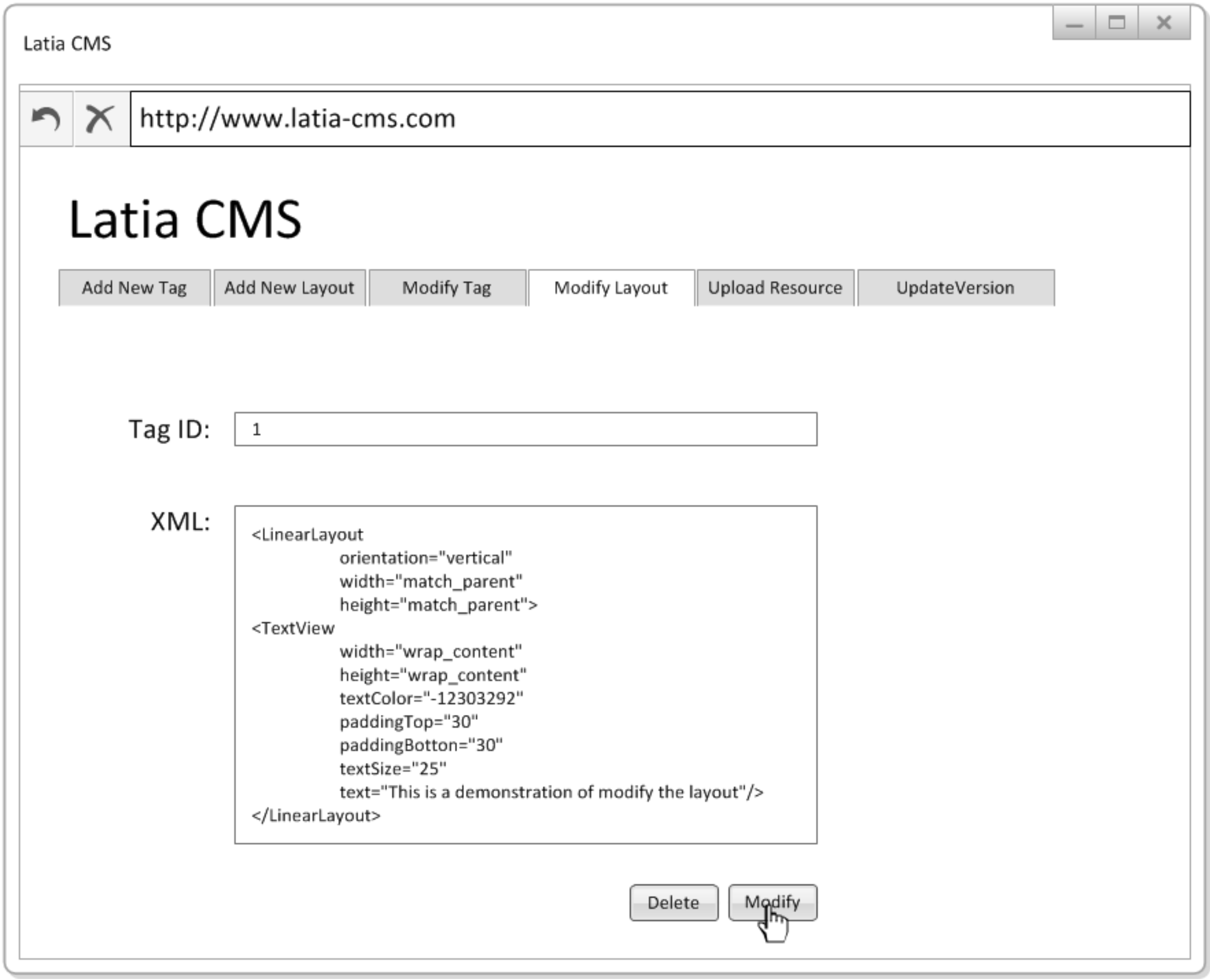


Error:
The Tag ID or Layout ID not found!

Tag ID: 10

Layout ID: 10

After the administrator entered the correct Tag ID and Layout ID, backend will return the layout xml code of that page to CMS. CMS will display the information in the text area. So, the administrators can modify the layout information and press ‘Modify’ button. The backend will get the modified layout information and process update.

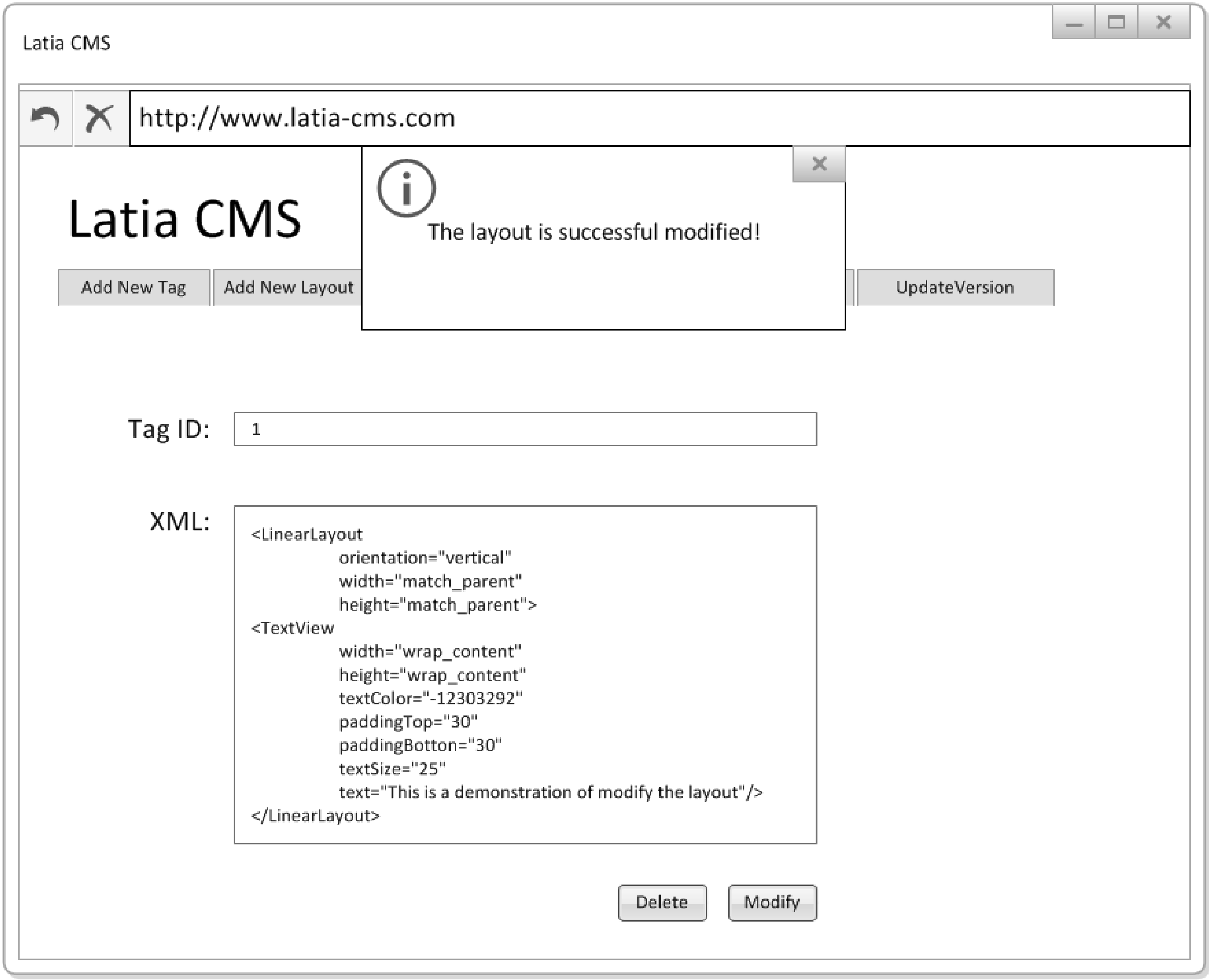


Tag ID: 1

XML:

```
<LinearLayout
            orientation="vertical"
            width="match_parent"
            height="match_parent">
<TextView
            width="wrap_content"
            height="wrap_content"
            textColor="-12303292"
            paddingTop="30"
            paddingBotton="30"
            textSize="25"
            text="This is a demonstration of modify the layout"/>
</LinearLayout>
```

If the update is success, CMS will pop-up a success message to inform administrator the update process is completed.



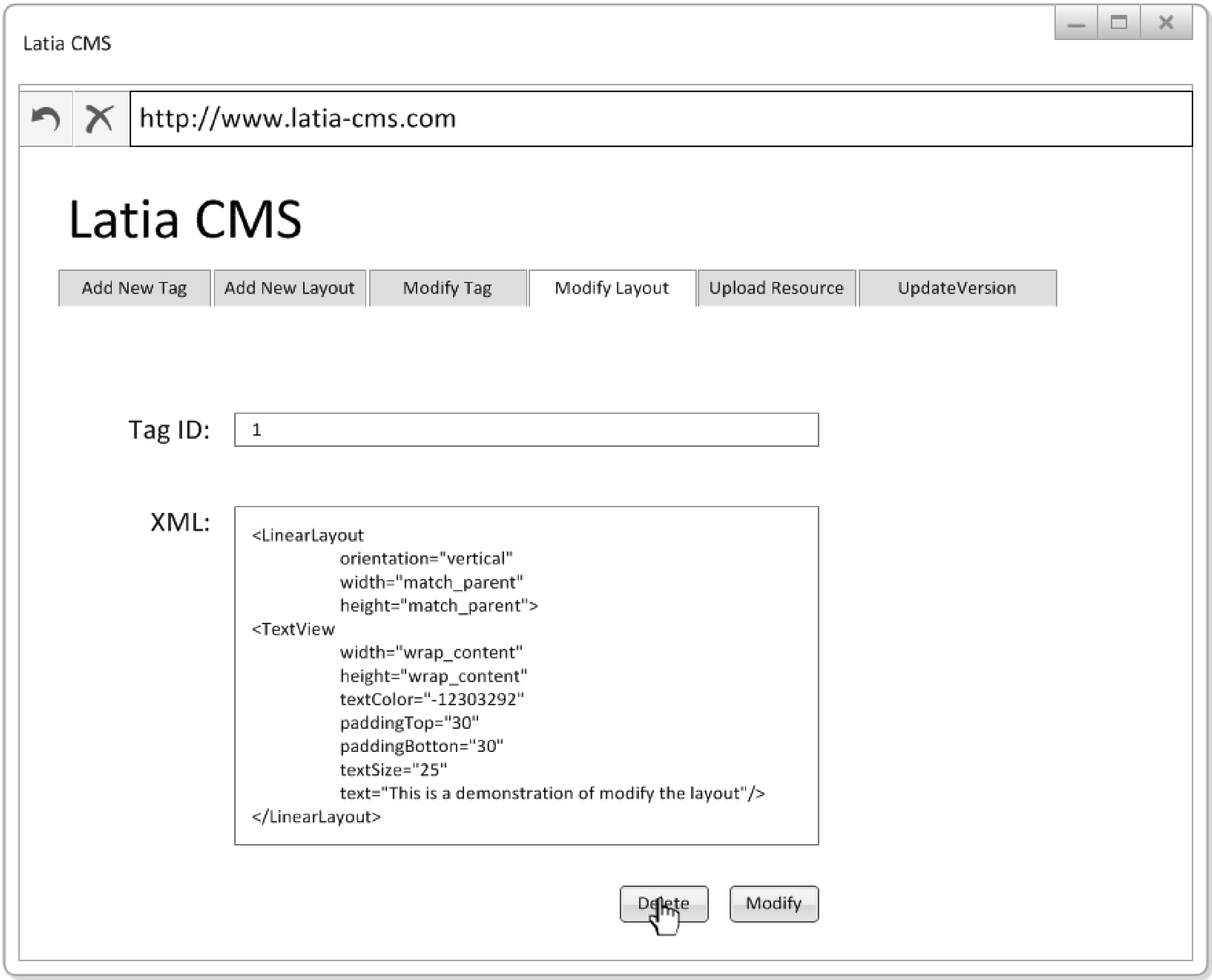
The layout is successful modified!

Tag ID: 1

XML:

```
<LinearLayout
            orientation="vertical"
            width="match_parent"
            height="match_parent">
<TextView
            width="wrap_content"
            height="wrap_content"
            textColor="-12303292"
            paddingTop="30"
            paddingBotton="30"
            textSize="25"
            text="This is a demonstration of modify the layout"/>
</LinearLayout>
```

If the administrators want to the delete this layout, they can press ‘Delete’ button.

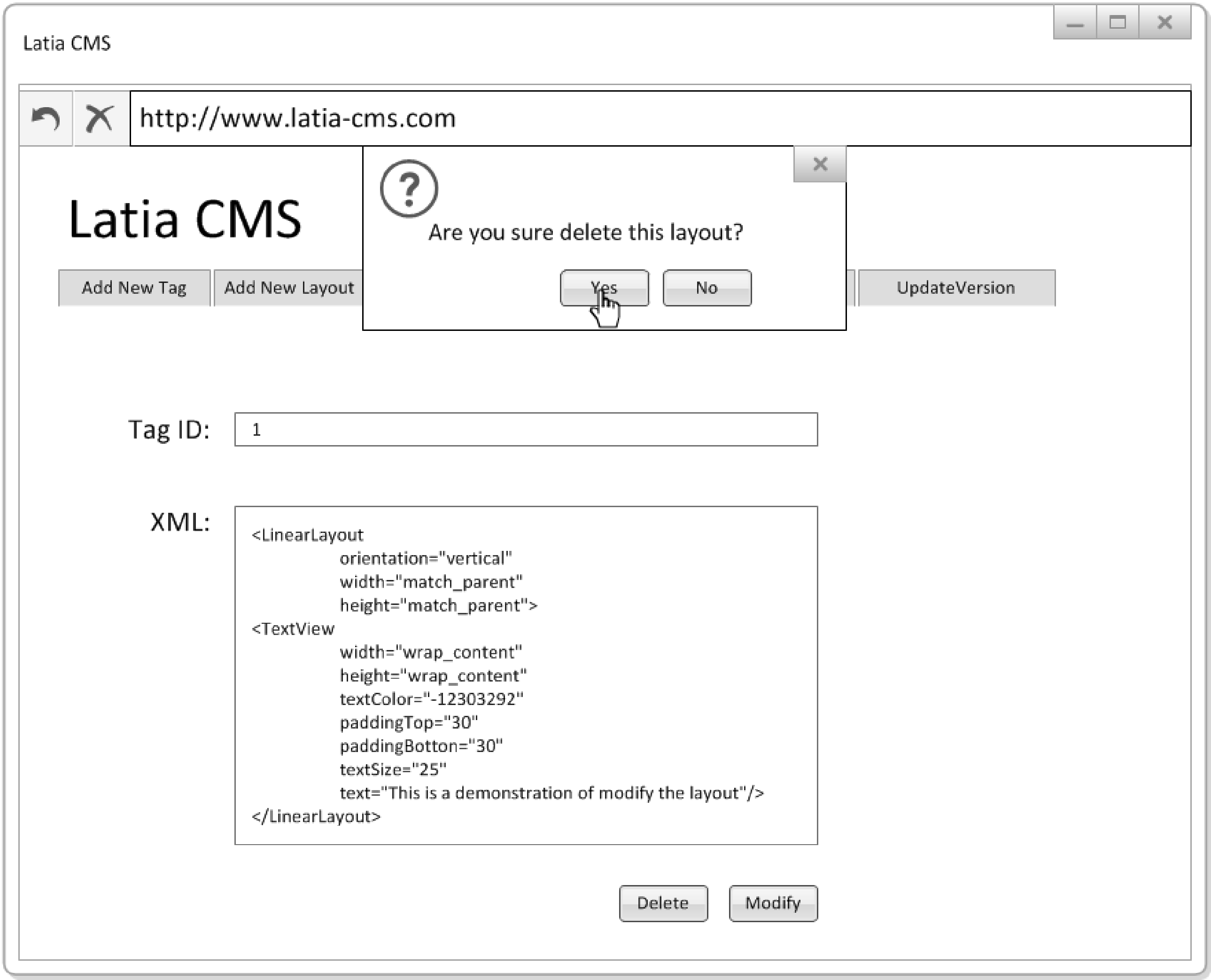


Tag ID: 1

XML:

```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">
<TextView
        width="wrap_content"
        height="wrap_content"
        textColor="-12303292"
        paddingTop="30"
        paddingBotton="30"
        textSize="25"
        text="This is a demonstration of modify the layout"/>
</LinearLayout>
```

After the ‘Delete’ button is pressed, an alert message will be popped-up and ask for confirm. If the administrators confirm deletion, they can press ‘Yes’ button. The backend will process the delete with database using the Tag ID.



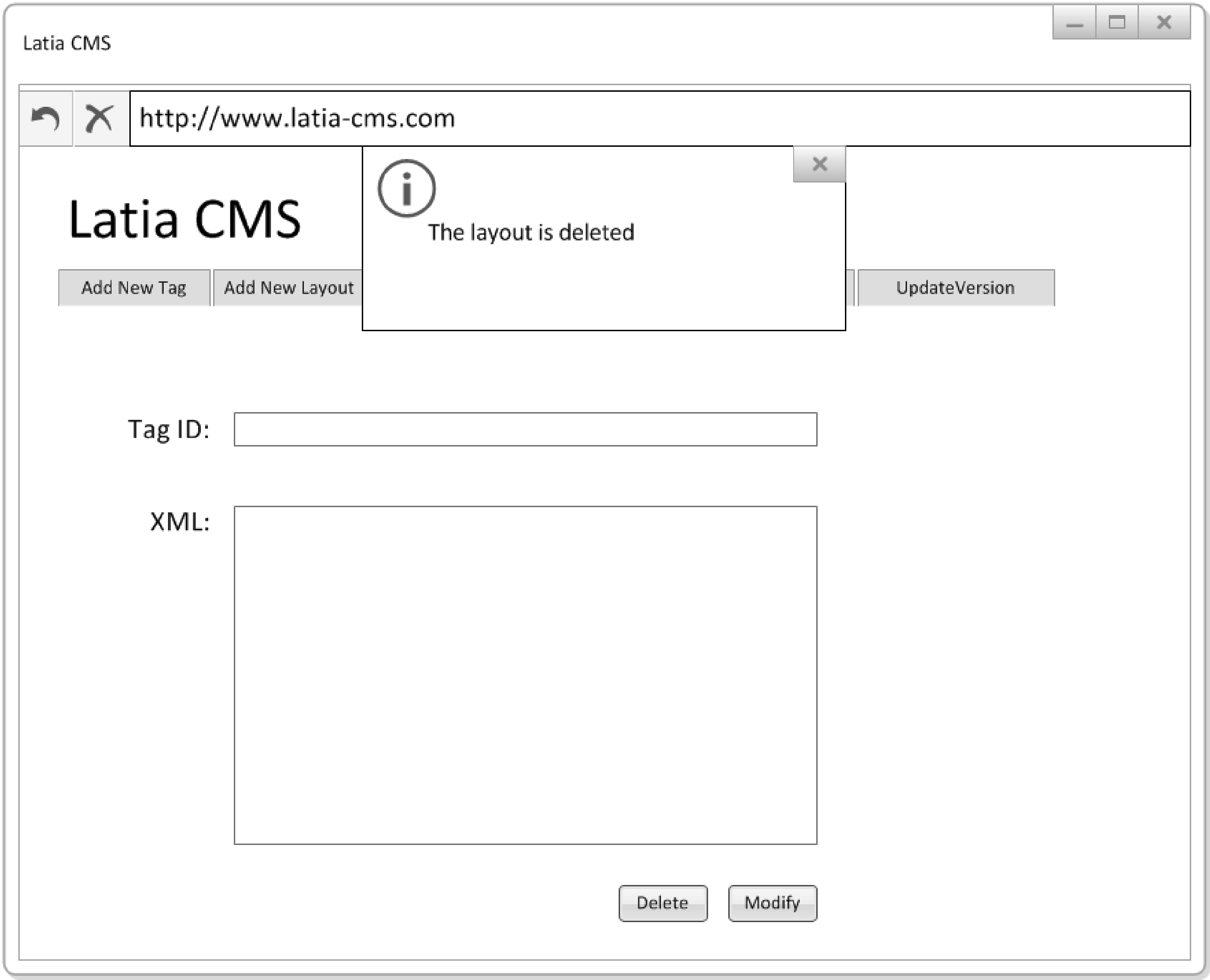
Are you sure delete this layout?

Tag ID: 1

XML:

```
<LinearLayout
            orientation="vertical"
            width="match_parent"
            height="match_parent">
<TextView
            width="wrap_content"
            height="wrap_content"
            textColor="-12303292"
            paddingTop="30"
            paddingBotton="30"
            textSize="25"
            text="This is a demonstration of modify the layout"/>
</LinearLayout>
```

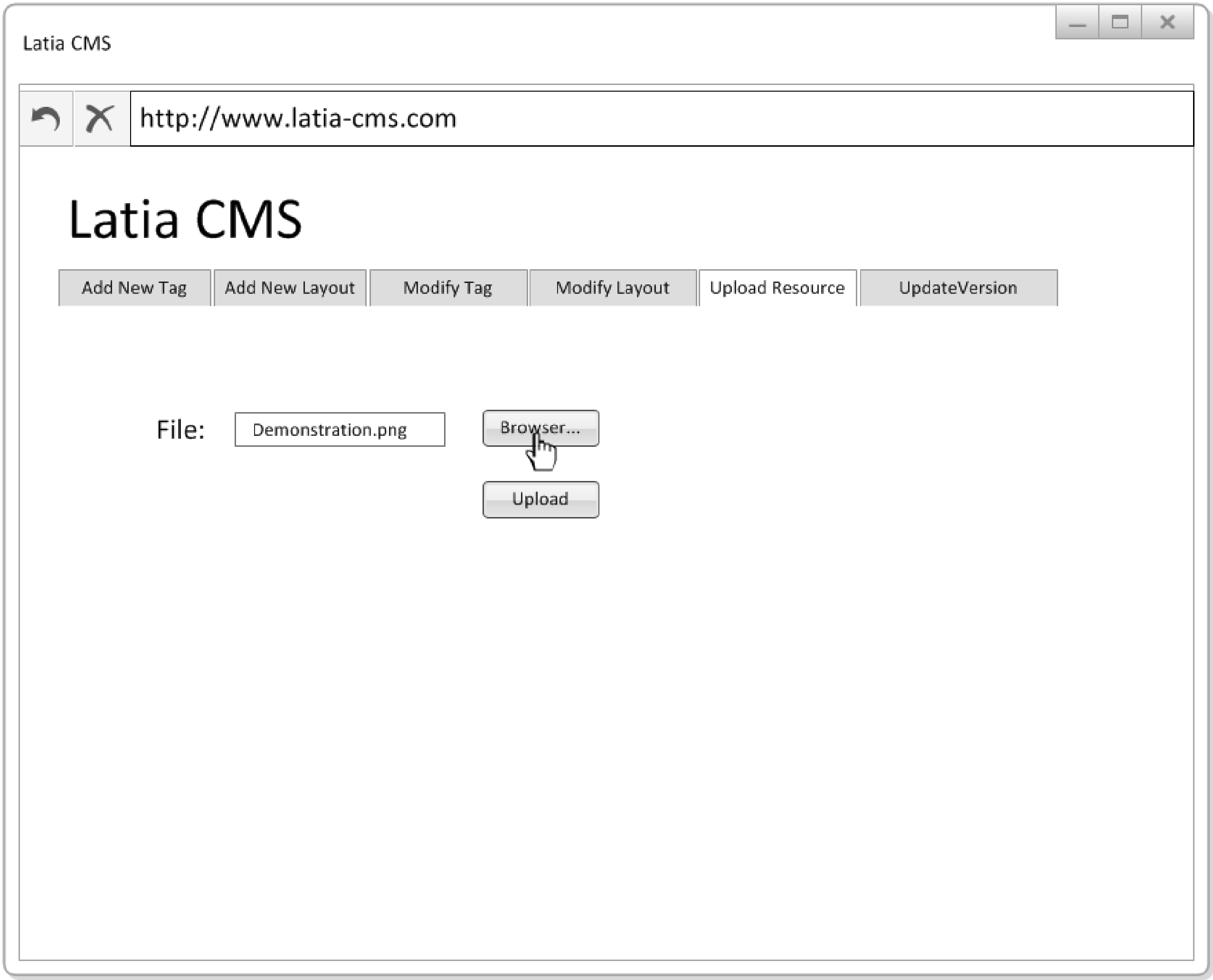
If the delete is successful, the CMS will pop-up a success message to inform administrator the delete process is completed.



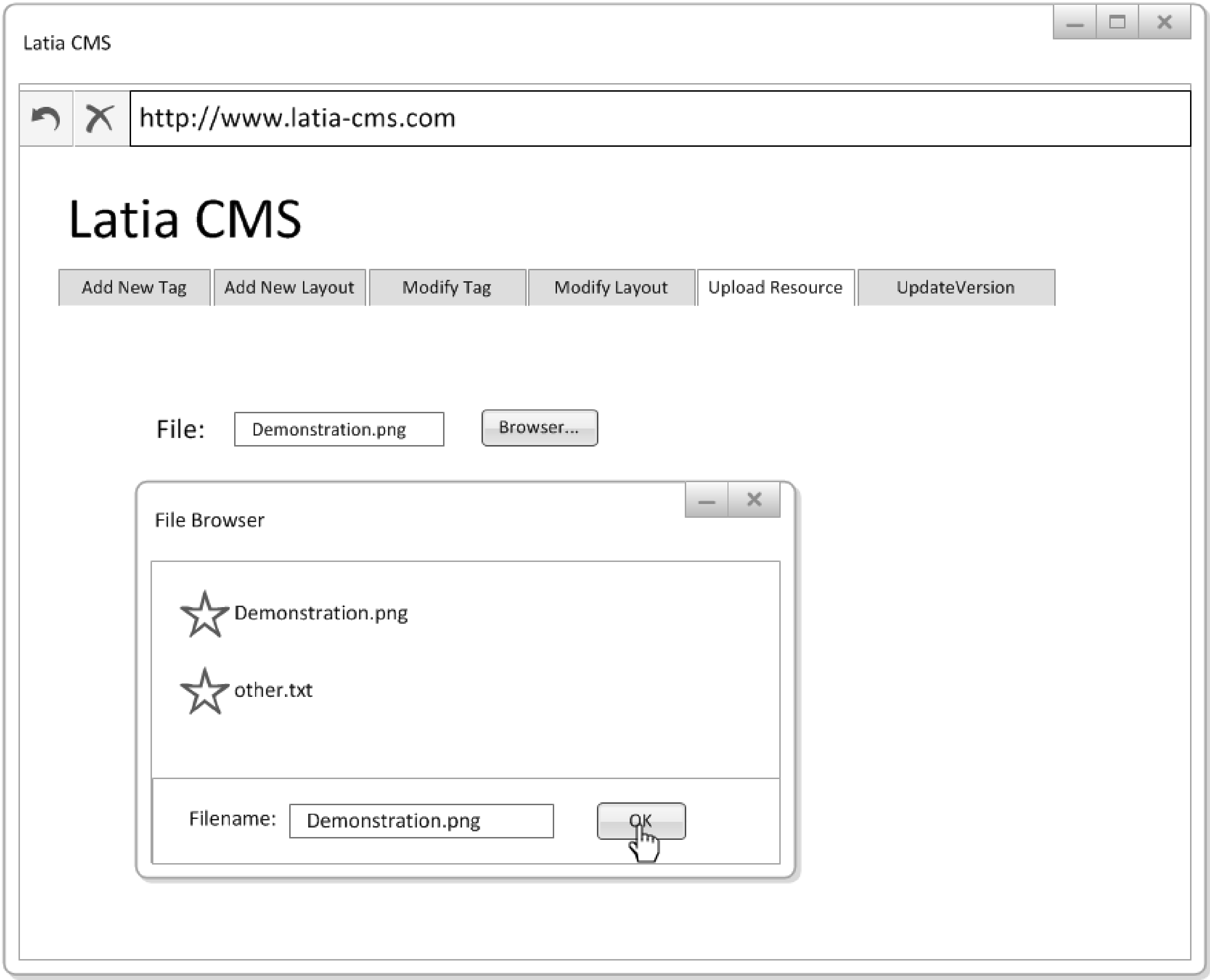
The layout is deleted

Upload Resource

The upload resource tab page provides a ‘Browser’ button and ‘Upload’ button. If the administrators want to upload a resource, they can press the ‘Browser’ button.



After the administrators press the ‘Browser’ button, a file browser window will be popped-up. Administrators can select a resource from the local directory for upload. Then, click ‘OK’ to confirm the resource.



Latia CMS

http://www.latia-cms.com

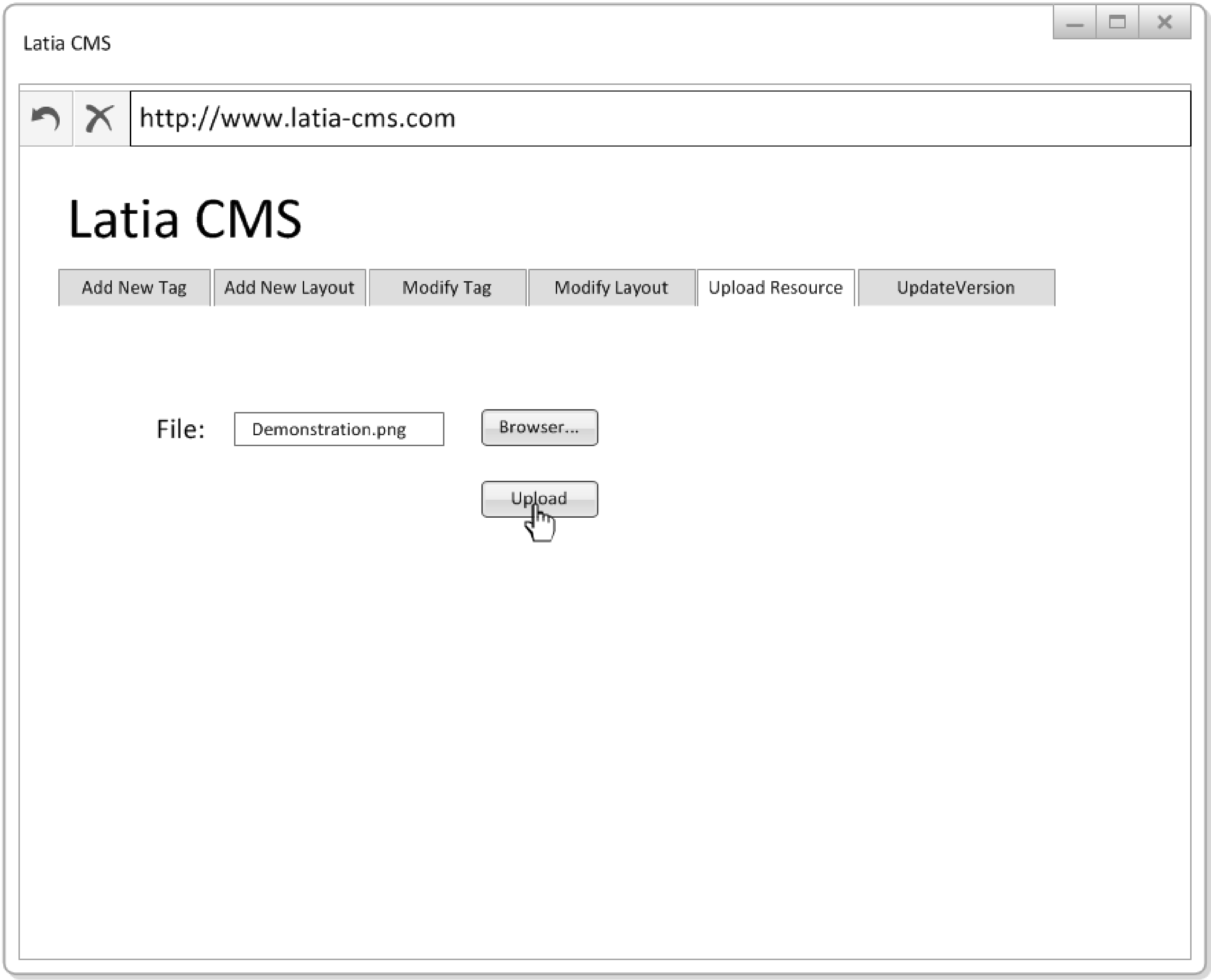
Latia CMS

File: Demonstration.png

File Browser

Filename: Demonstration.png

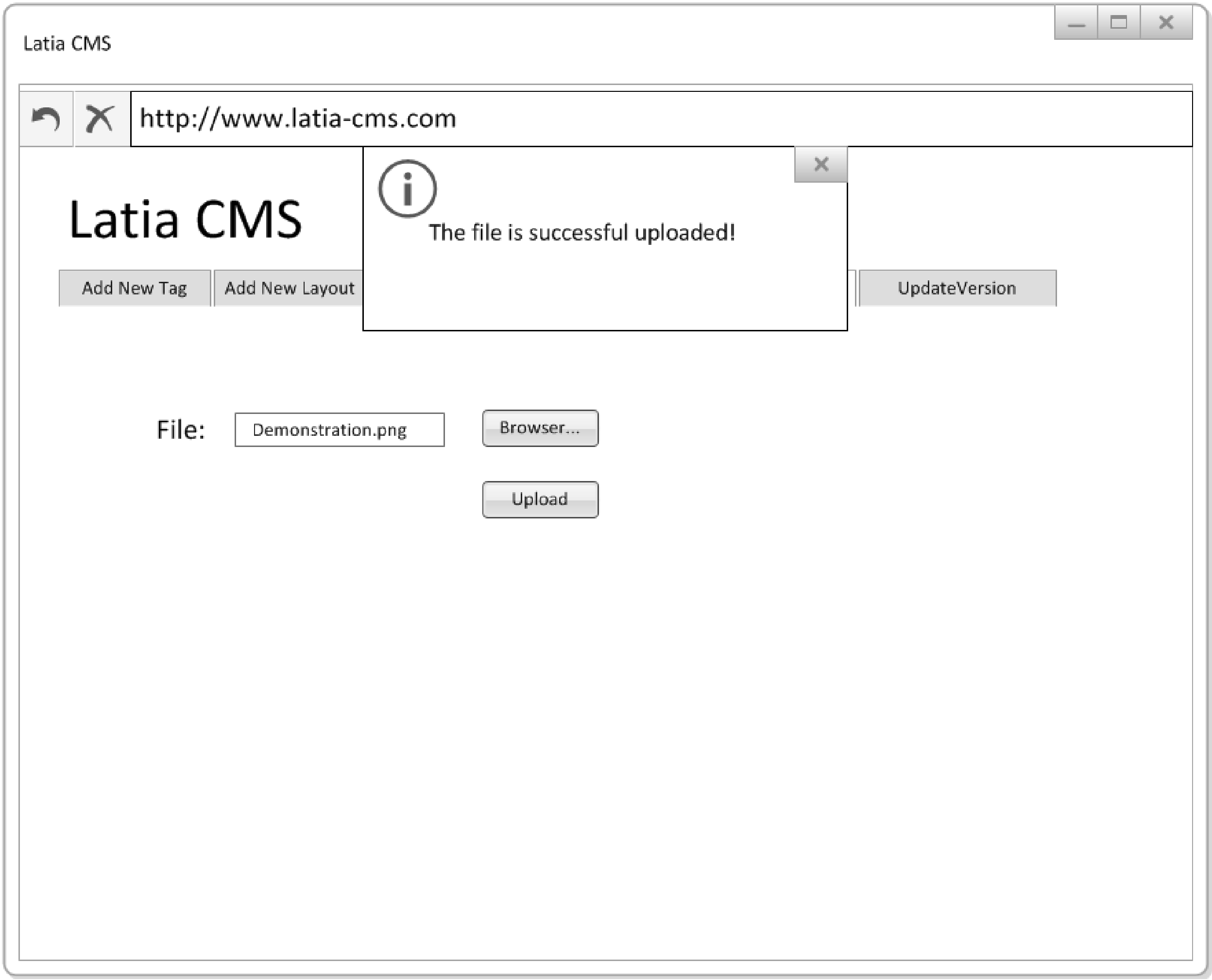
After the administrators selected the resource and press ‘Upload’ button, the backend will get the file and upload the resource to the database.



http://www.latia-cms.com

File: Demonstration.png

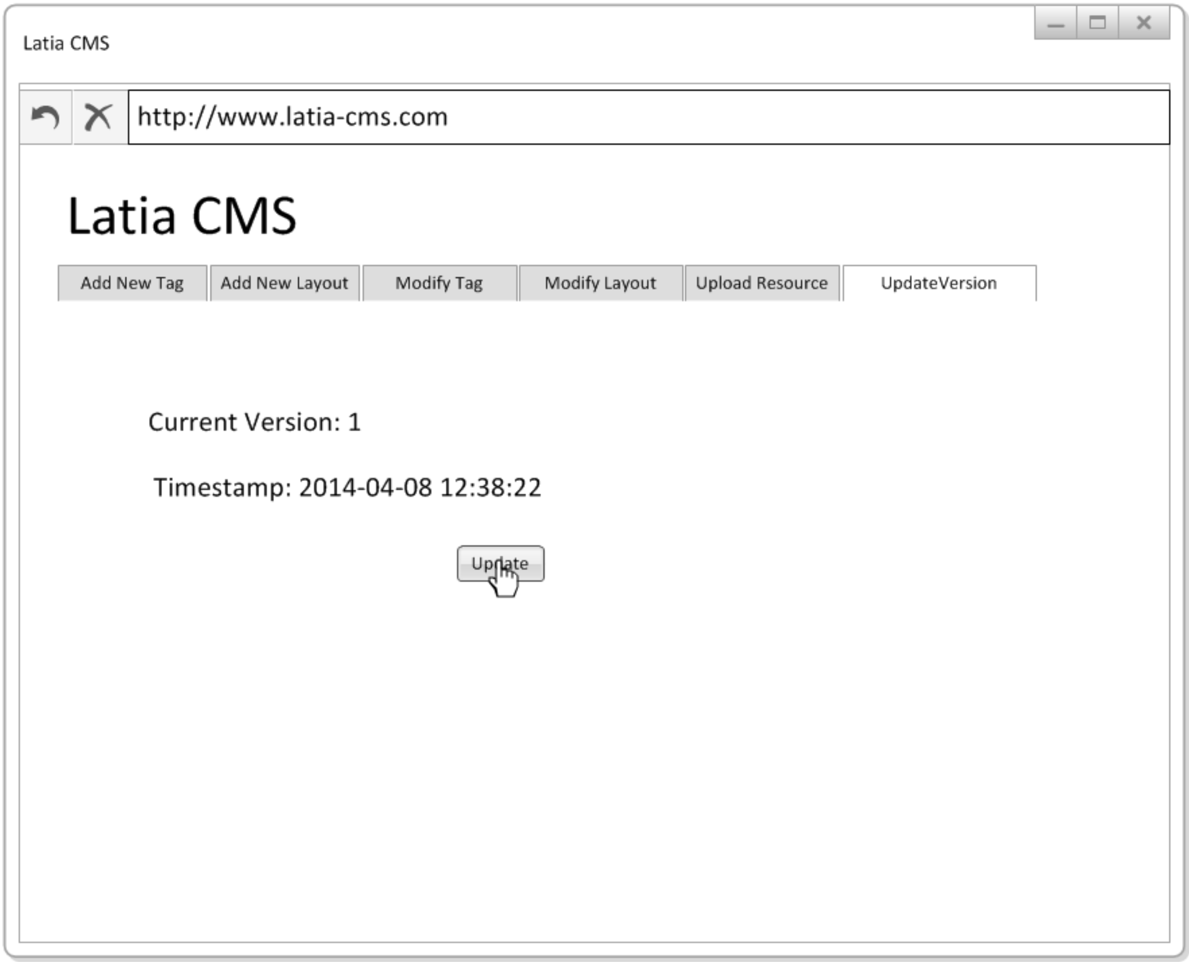
If the resource is uploaded successfully in backend, the CMS will pop-up a message to inform administrator the upload process completed.



The file is successful uploaded!

Update Version

In the update version, it contains two labels, current version and timestamp. Also, there will be a 'Update' button. If the administrators want to release a new version, they can press the 'Update' button. The backend process will add a new version in database.


Latia CMS

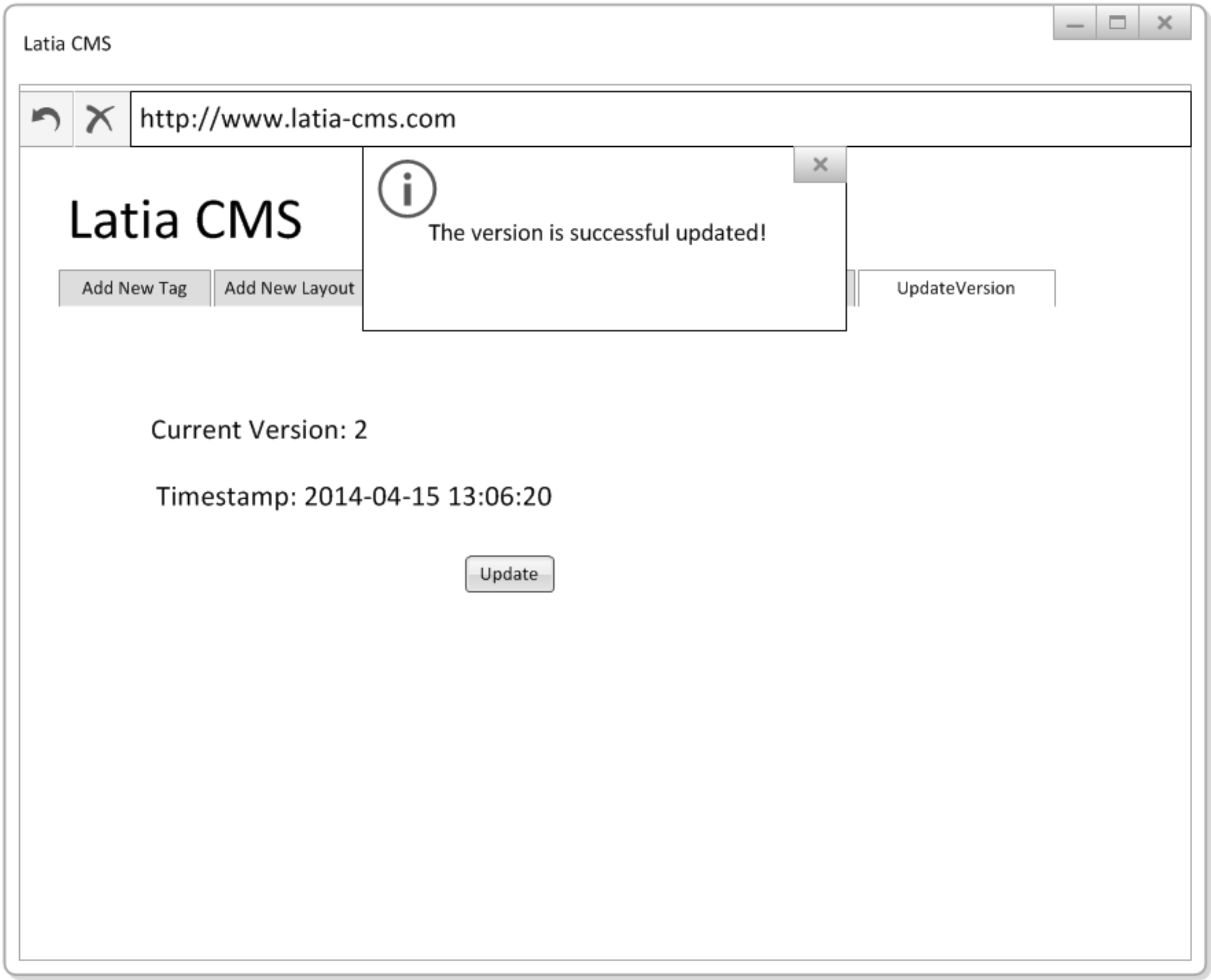
http://www.latia-cms.com

Latia CMS

Current Version: 1

Timestamp: 2014-04-08 12:38:22

If the update is successful, the CMS will pop-up a message to inform administrator the update version process is completed. And the current version and timestamp will be reloaded in the page.



The version is successful updated!

Current Version: 2

Timestamp: 2014-04-15 13:06:20

6.2.2 Backend

The interface for CMS used to query or modify the database with the administrators input from the CMS and returns the requested data from database to CMS and display to administrators. The interface involves eight activities: get login information, add tag information, modify tag information, add layout information, modify layout information, delete layout, upload resource, and update version.

Get Login Information

Add Tag Information

Modify Tag Information

Add Layout Information

Modify Layout Information

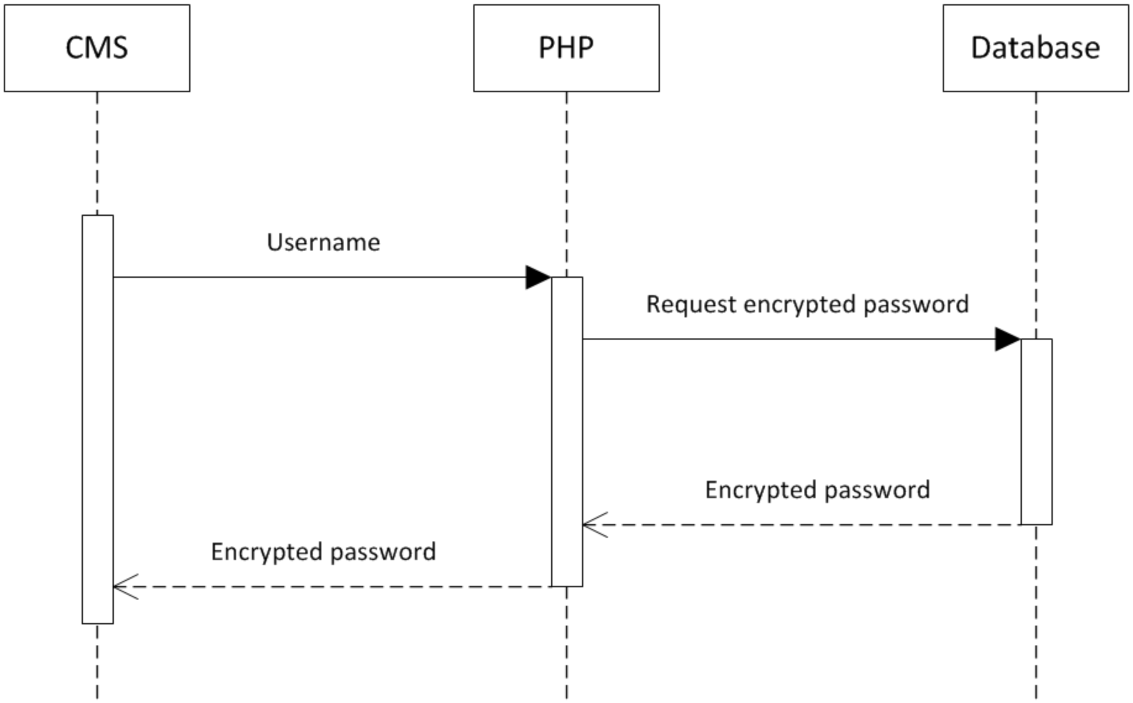
Delete Layout

Upload Resource

Update Version

Get Login Information

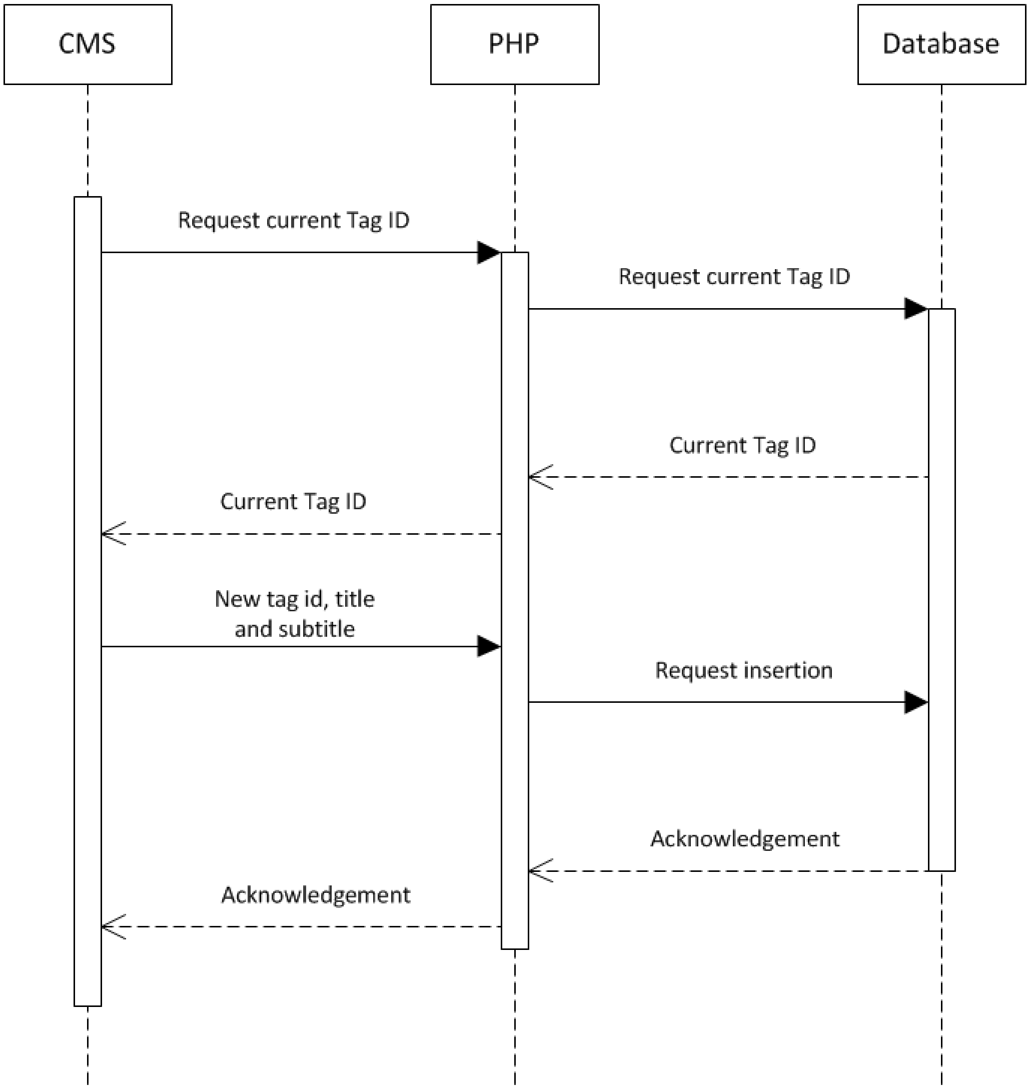
After the administrators enter the username and password, CMS will send the username to PHP to request on getting the encrypted password. PHP will use the username to query the database for retrieving the encrypted password. If it is success, PHP will return the resource data and CMS can compare password. If the username is not found, PHP will return error to the CMS.



Add Tag Information

CMS allow administrator add a tag by simply entering the title and subtitle. CMS will send a request to get the current Tag ID. PHP will query the database and return the current Tag ID to CMS.

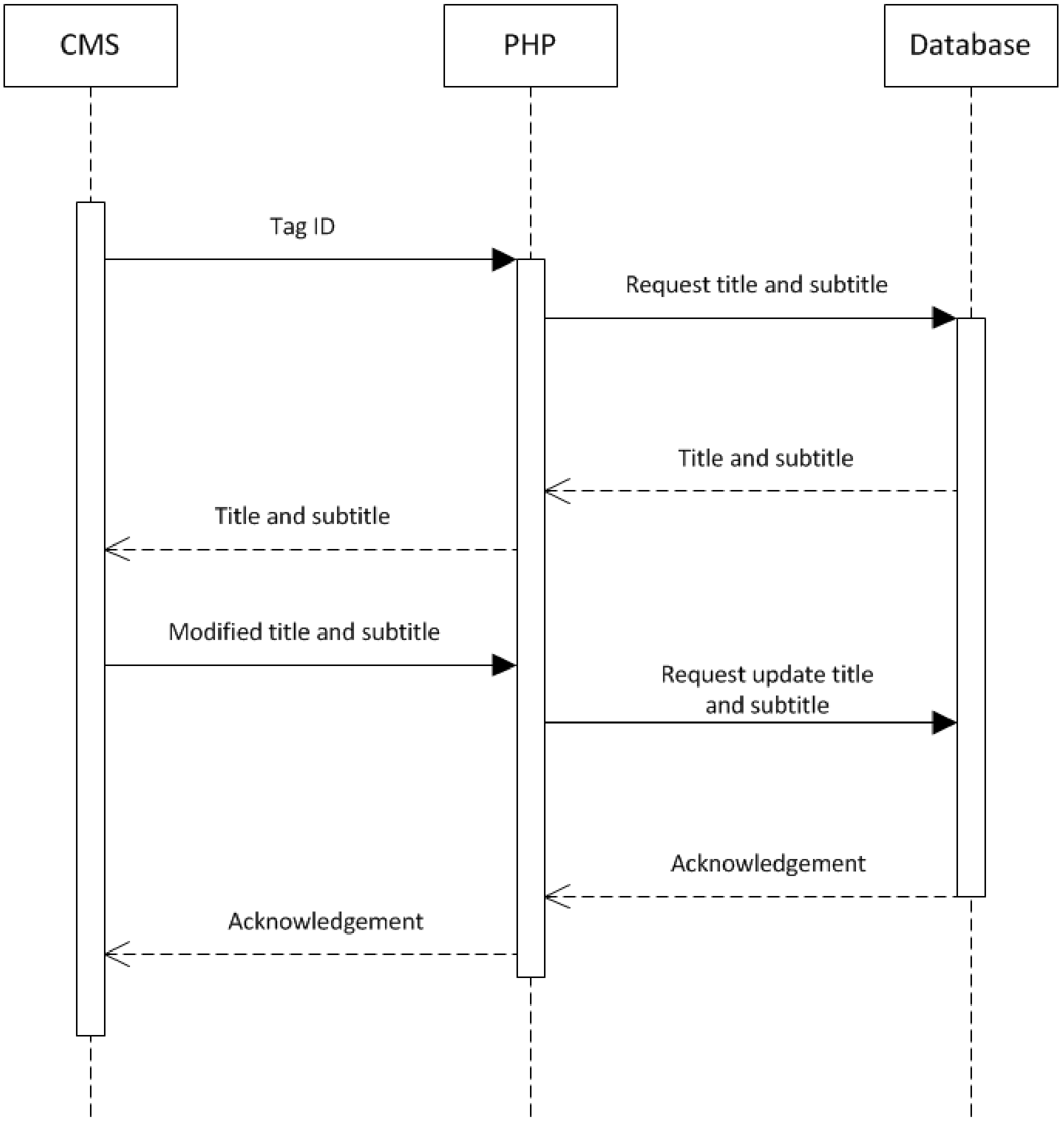
After the CMS get the current Tag ID, it can generate a new Tag ID and send title and subtitle with it to PHP. The PHP will send the tag information to database for insertion. If it is success, PHP will send acknowledgement to the CMS.


CMS
PHP
Database
Request current Tag ID
Request current Tag ID
Current Tag ID
Current Tag ID
New tag id, title and subtitle
Request insertion
Acknowledgement
Acknowledgement

Modify Tag Information

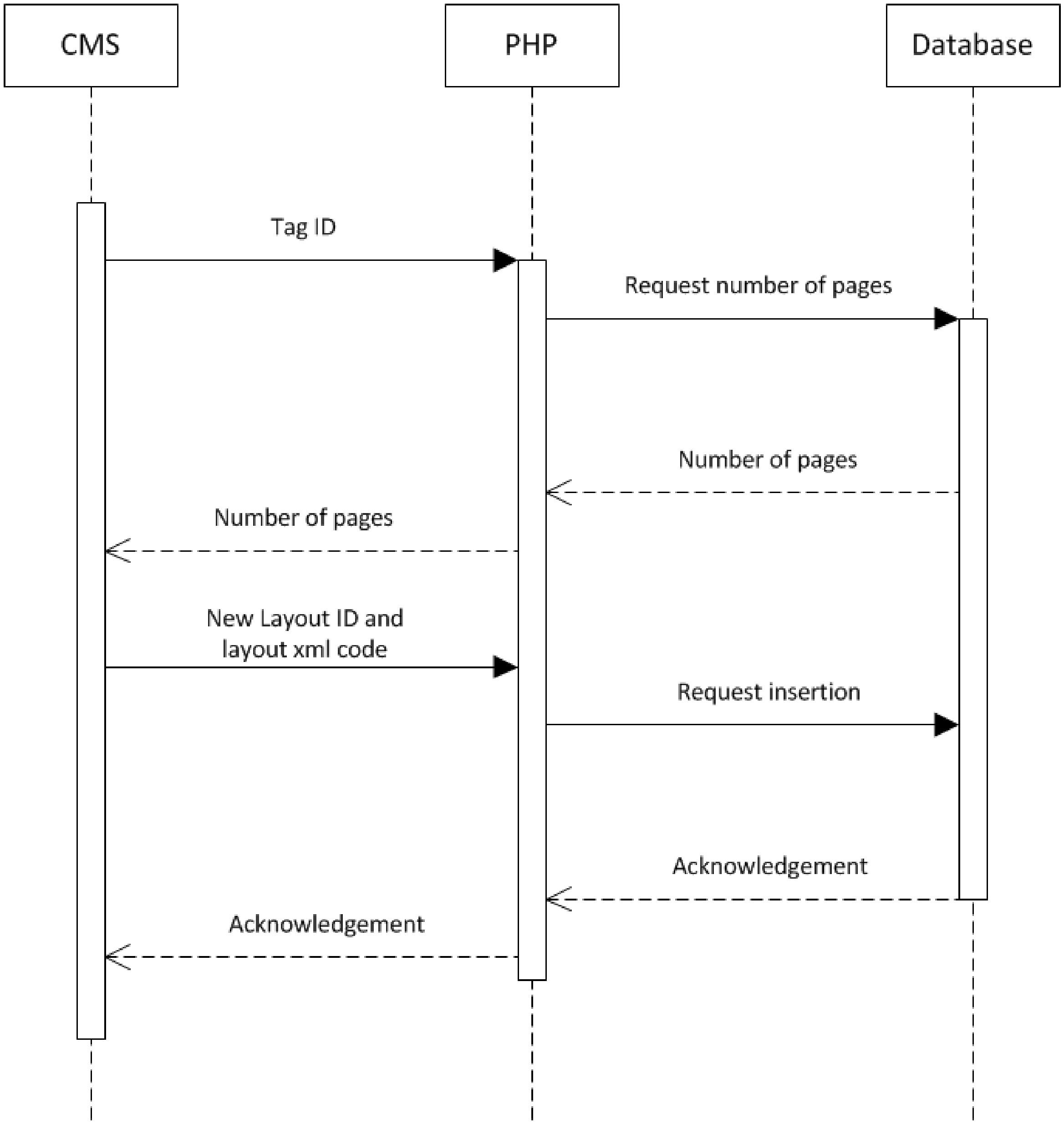
After the administrator selected the tag, CMS will send the selected Tag ID to PHP request getting the tag information. The PHP will query the database and return the result to CMS.

If the Tag ID is found, CMS will receive the title and subtitle from PHP. So, CMS can display to administrators and let them modify it. Then, CMS will send the modified tag information and Tag ID to PHP. And the PHP will request the database to update the particular tag information with the modified one. If it is success, PHP will send acknowledgement to the CMS.


CMS
PHP
Database
Tag ID
Request title and subtitle
Title and subtitle
Title and subtitle
Modified title and subtitle
Request update title and subtitle
Acknowledgement
Acknowledgement

Add Layout Information

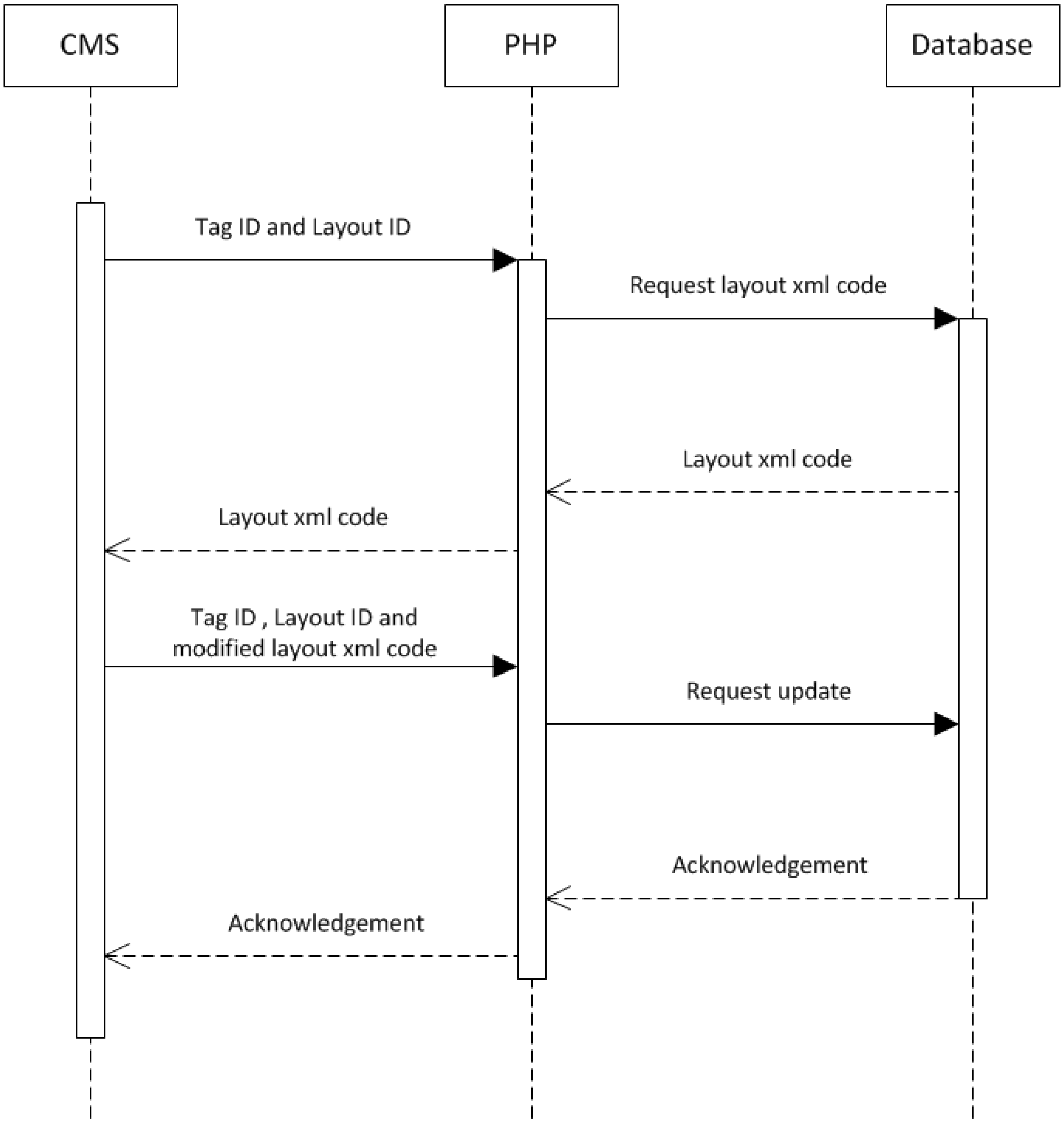
The administrators can add a layout xml to a specified Tag ID. The CMS will first send the Tag ID to PHP to request on getting its number of pages. The PHP will return the result after querying the database. Then, CMS can generate the new Layout ID and send it with the layout xml code entered by the administrators to PHP. After that, PHP will send the layout information to the database for insertion. If it is success, PHP will send acknowledgement to the CMS.


CMS
PHP
Database
Tag ID
Request number of pages
Number of pages
Number of pages
New Layout ID and
layout xml code
Request insertion
Acknowledgement
Acknowledgement

Modify Layout Information

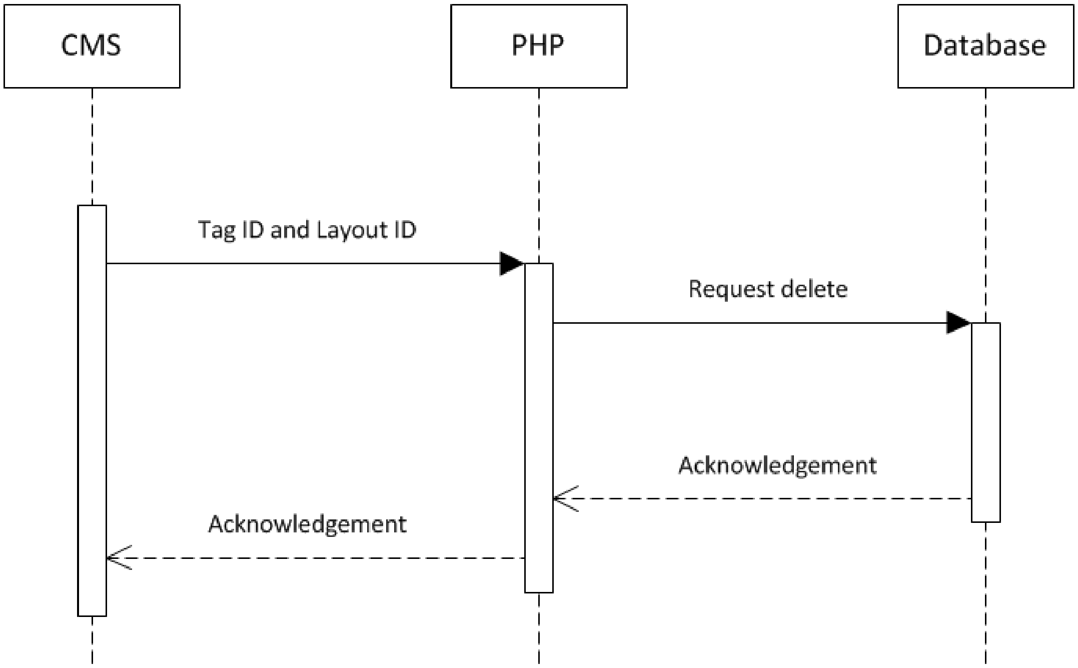
CMS get the Tag ID and Layout ID that selected by the administrators. Then, CMS will send the information to PHP request getting the layout information for that Tag ID. The PHP will query the database and return the result to CMS.

If the Tag ID and Layout ID are found, CMS will receive the layout xml code from PHP and display it to administrators so that they can modify it. Then, CMS will send the modified layout information and Tag ID to PHP. And the PHP will request the database to update the particular layout information with the modified one. If it is success, PHP will send acknowledgement to the CMS.


CMS
PHP
Database
Tag ID and Layout ID
Request layout xml code
Layout xml code
Layout xml code
Tag ID , Layout ID and
modified layout xml code
Request update
Acknowledgement
Acknowledgement

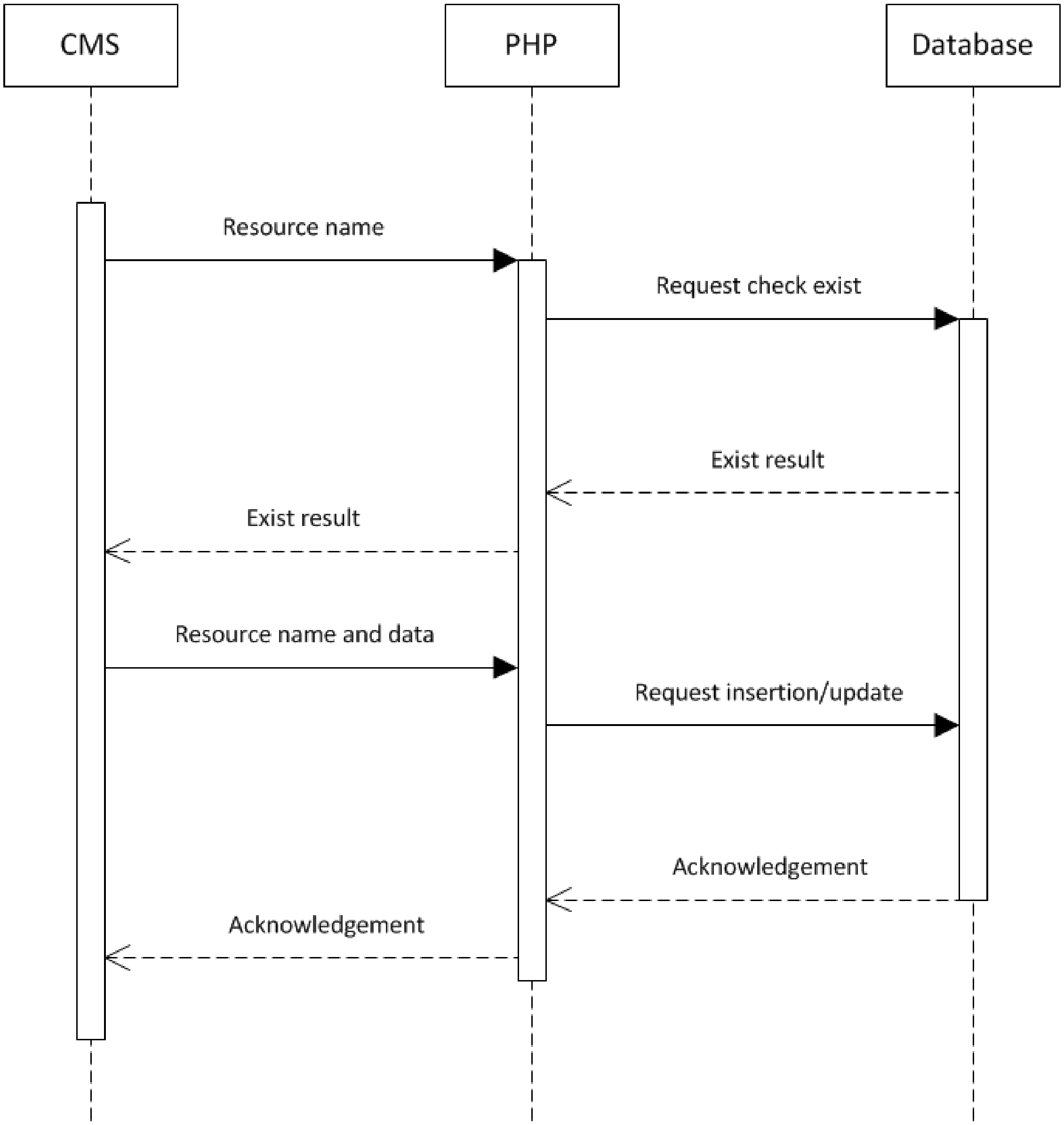
Delete Layout

CMS get the selected Tag ID and Layout ID from administrators. It will send the information to PHP. After PHP received, it will request the database to delete the entry with the same Tag ID and Layout ID. If the delete is success, PHP will send acknowledgement to the CMS.



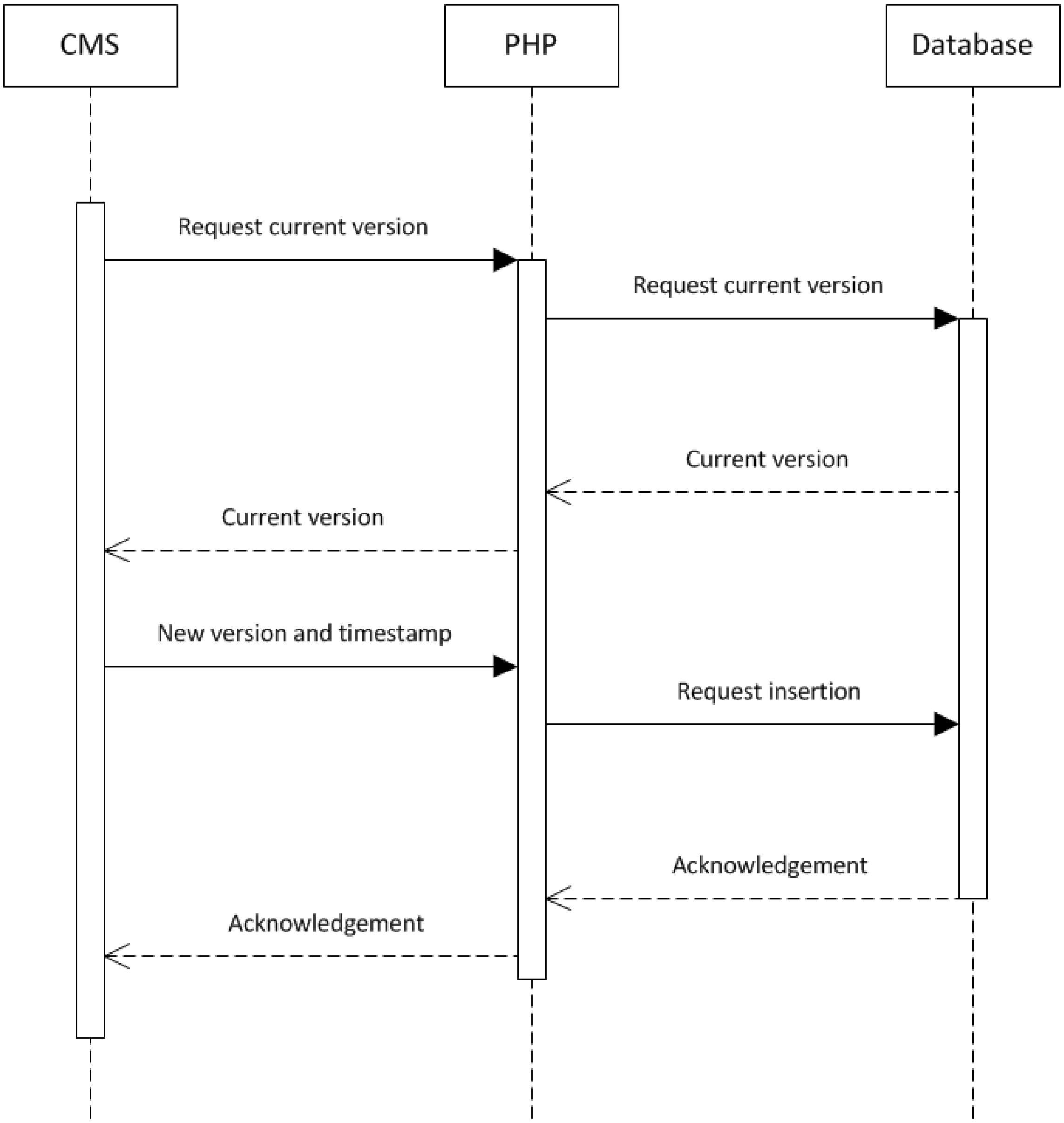
Upload Resource

The administrator can browser a resource in CMS for uploading. The CMS will send the selected resource name to PHP. Then, PHP will query the database to see if the resource is existed. If the resource not exists, the CMS will send a request to PHP for insertion or it will send a request to PHP for update. After that, PHP will pass the request to database. If it is success, PHP will send acknowledgement to the CMS.


CMS
PHP
Database
Resource name
Request check exist
Exist result
Exist result
Resource name and data
Request insertion/update
Acknowledgement
Acknowledgement

Update Versions

CMS will send a request to PHP for getting the current version number. The PHP will return the result after querying the database. So, CMS can generate a new version number and send it to PHP. Then, PHP will request the database update the version number with it and update the timestamp to current time.

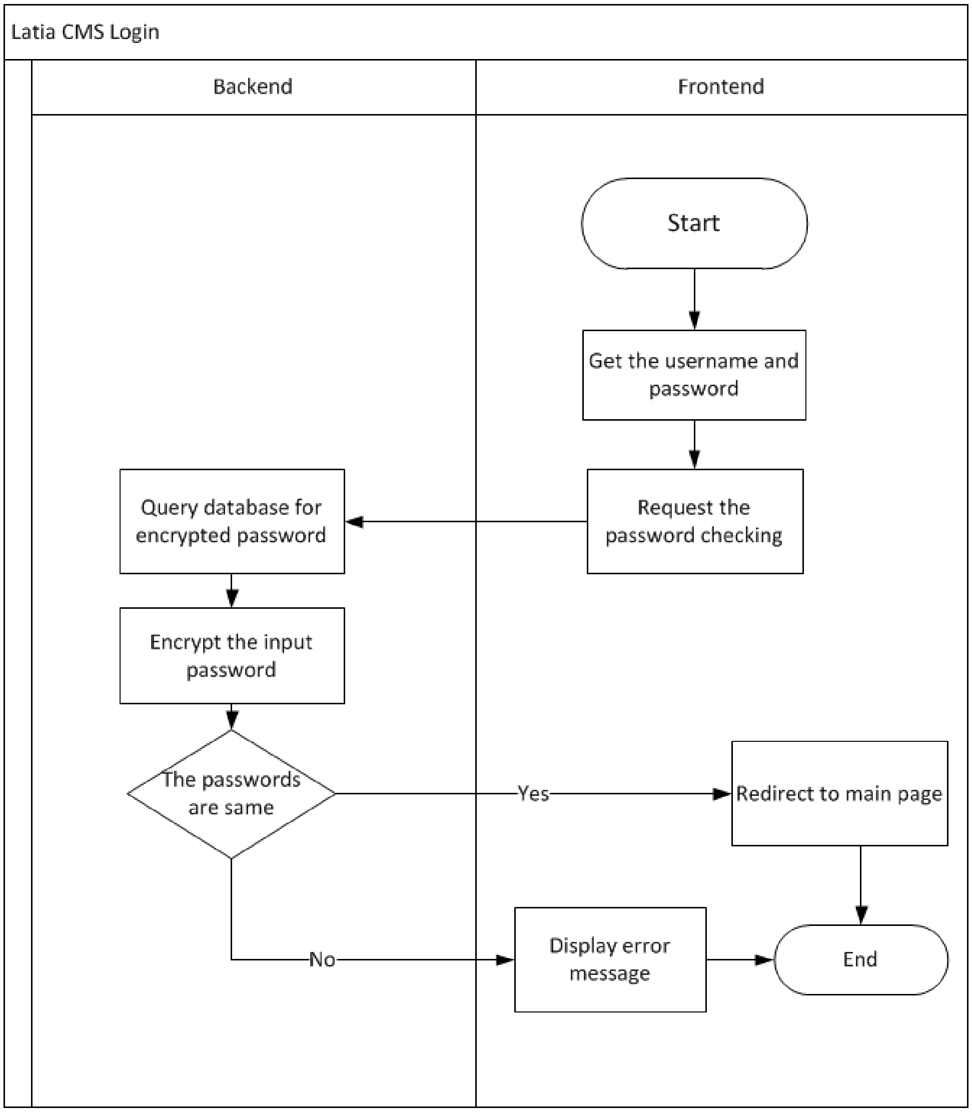

CMS
PHP
Database
Request current version
Request current version
Current version
Current version
New version and timestamp
Request insertion
Acknowledgement
Acknowledgement

6.2.3 Activity flow diagram

The CMS supports the seven functions: add new tag information, add layout information, modify tag information, modify layout information, delete layout information, insert or update resource and update version functions. These functions require backend and frontend communicate with each other in order to let administrators add or modify the information in database. The following Activity flow diagram will show the process flow between them in each function.

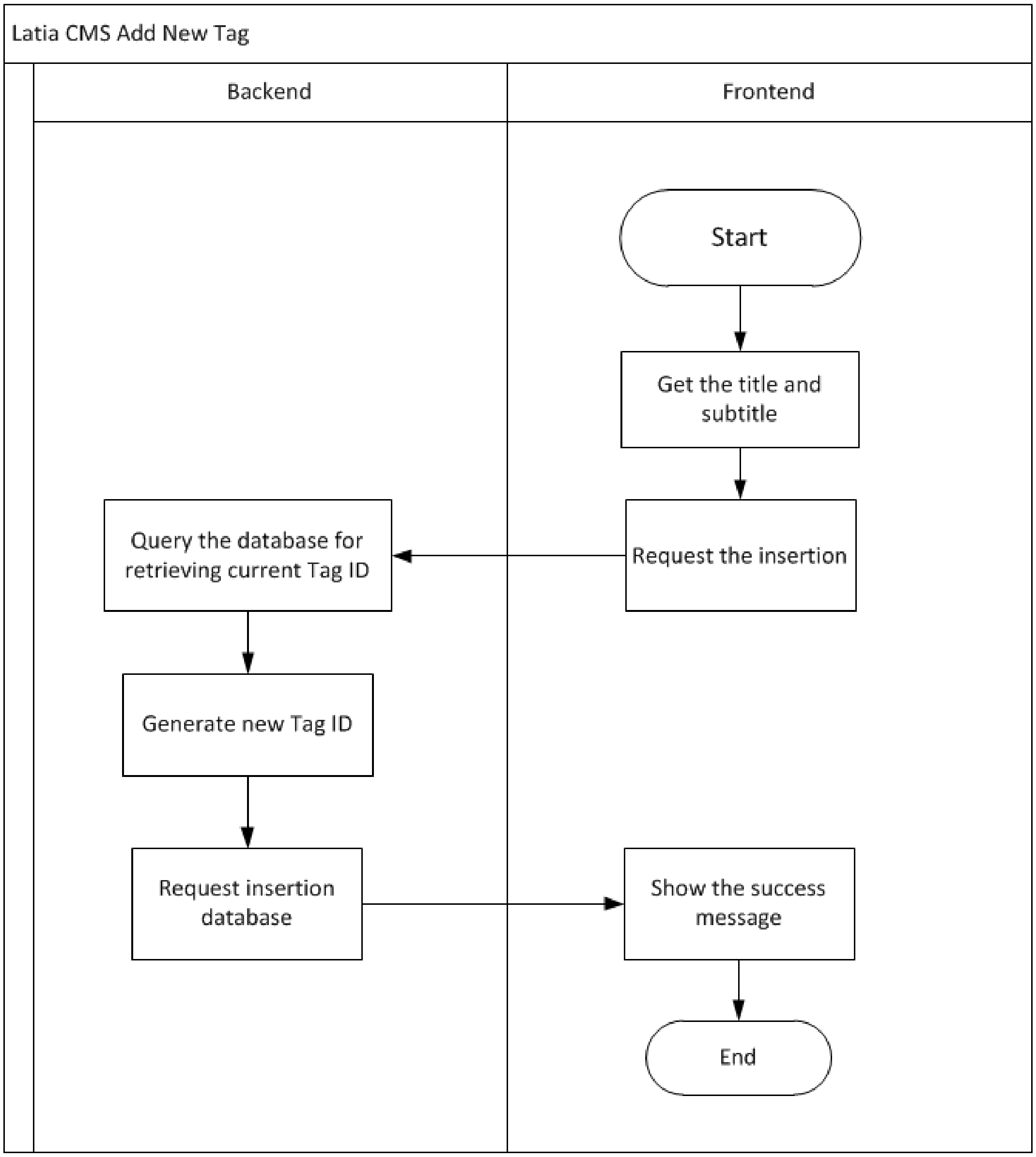
Login

The frontend will get the username and password from administrators. It will pass the information to backend for password validation. After backend received, it will query the database to get the encrypted password using the username. Then, backend will encrypt the password received from frontend. It will compare the two encrypted passwords to check if they are equal. It will send the result to frontend. If they are equal, it will redirect to the CMS main page. If not, it will show the error message and reload the login page.


Latia CMS Login
Backend
Frontend
Start
Get the username and password
Request the password checking
Query database for encrypted password
Encrypt the input password
The passwords are same
Yes
Redirect to main page
No
Display error message
End

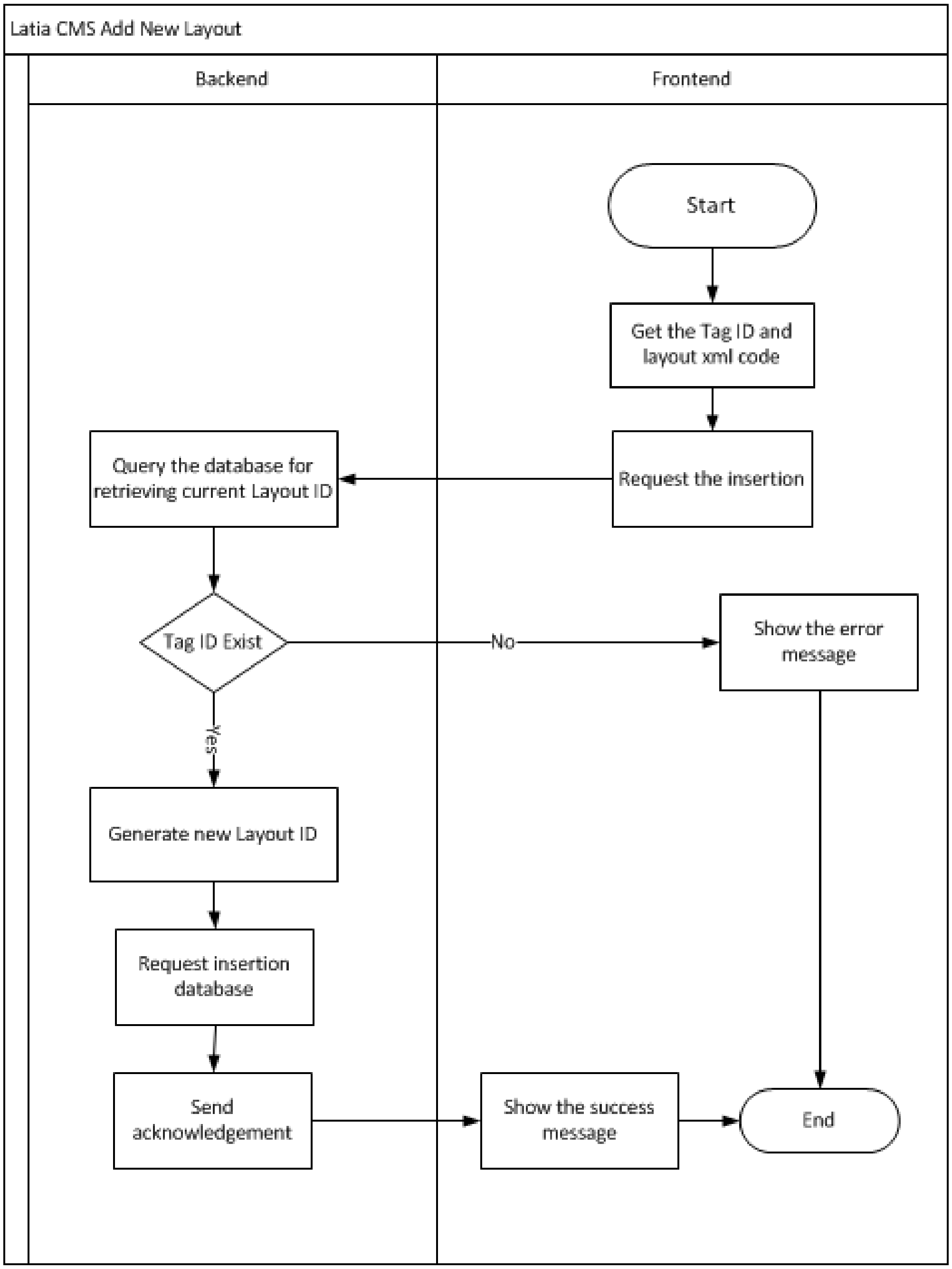
Add New Tag

The administrator will pass the new tag information, title and subtitle, to frontend. It will pass the information to the backend and request insertion. In order to insert a new tag, new Tag ID is needed. So, the backend will first query the database for retrieving the current Tag ID. It will generate the new Tag ID. Then, it will construct a query to request insertion the new tag to the database. After that, it will acknowledge frontend. Finally, frontend will show the success message to administrators.


Latia CMS Add New Tag
Backend
Frontend
Start
Get the title and subtitle
Request the insertion
Query the database for retrieving current Tag ID
Generate new Tag ID
Request insertion database
Show the success message
End

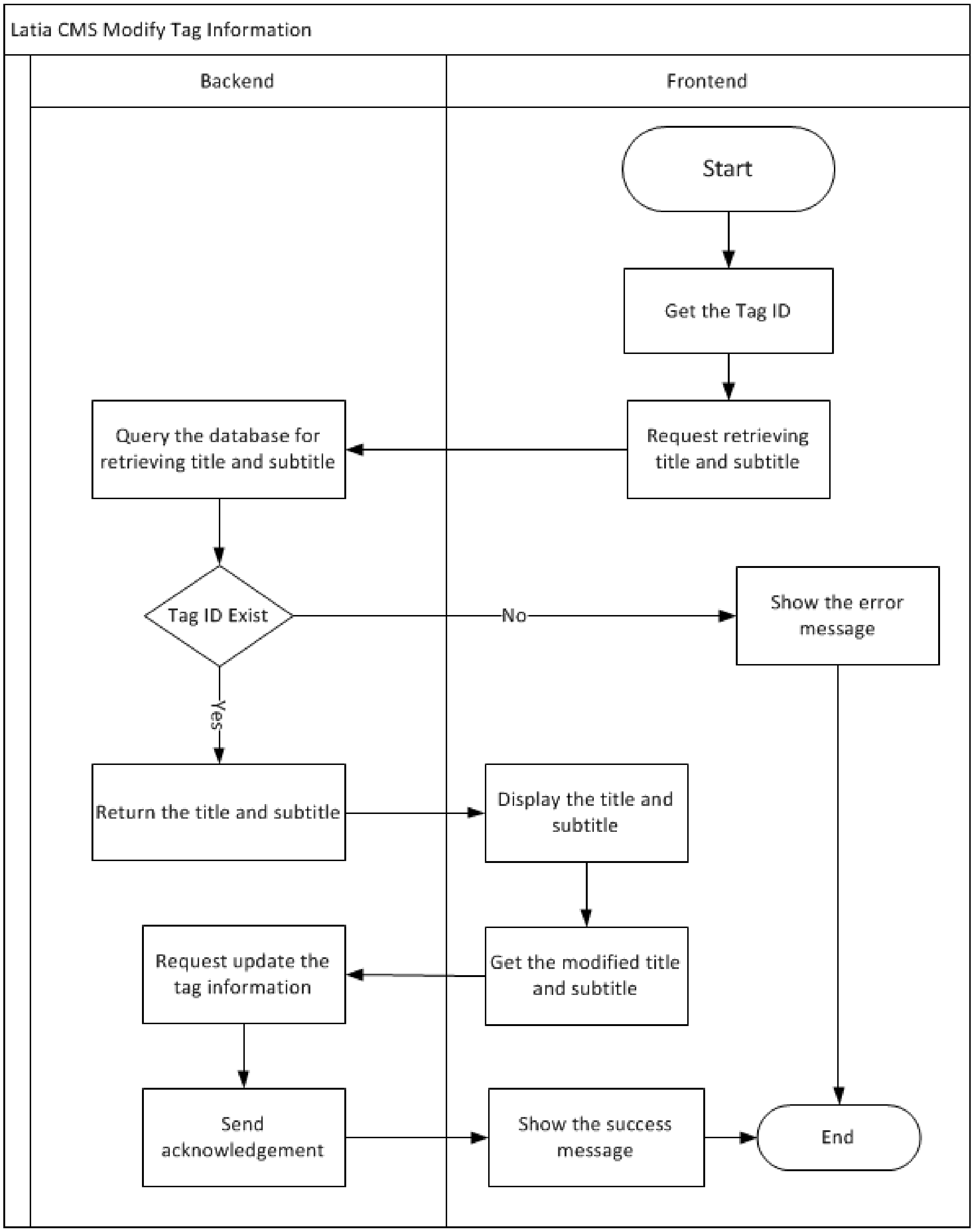
Add New Layout

The frontend will receive the Tag ID that administrators want to add a page on it and the layout xml code. It will pass the information to backend and request for inserting the new layout. The Backend will query the database for retrieving current Layout ID for that Tag. If the Tag ID input by the administrators does not exist, the backend will return a fail message to frontend. Then, error message will be shown by frontend. If the Tag ID exists, backend will generate the new Layout ID and query the database to request insertion of the new Layout.


Latia CMS Add New Layout
Backend
Frontend
Start
Get the Tag ID and layout xml code
Request the insertion
Query the database for retrieving current Layout ID
Tag ID Exist
No
Show the error message
Yes
Generate new Layout ID
Request insertion database
Send acknowledgement
Show the success message
End

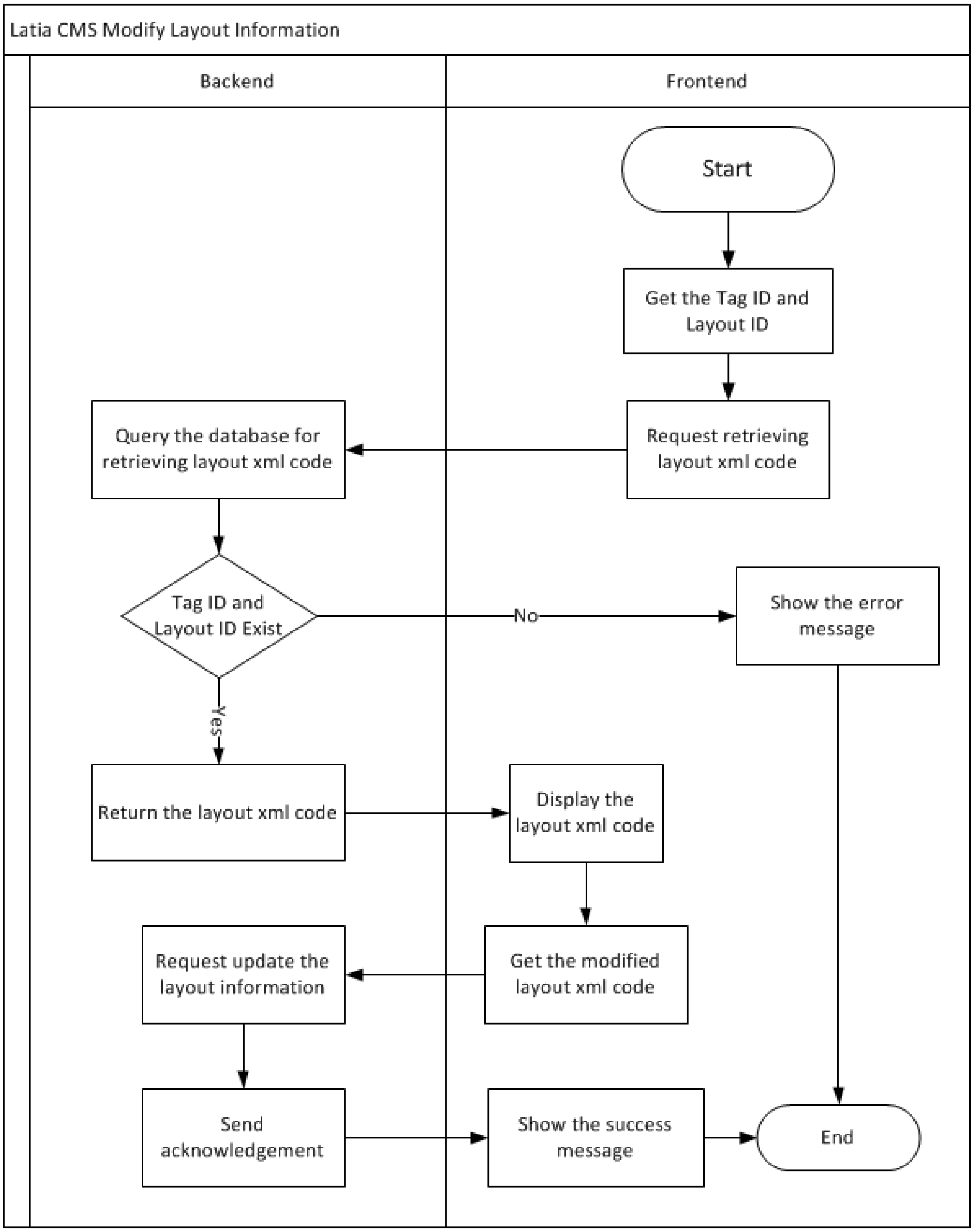
Modify Tag Information

The administrators will send the Tag ID that need to be modified to the frontend. Then, frontend will request backend give the title and subtitle of that tag to it. The backend will query the database to get this information. If the Tag ID entered by the administrator does not exist, it will return fail message to frontend so it can show the error message to administrators. If the Tag ID exists, backend will return the title and subtitle to the frontend. Then, frontend displays the title and subtitle to administrator for modification. It will send the modified tag information to backend again to request insertion of the new tag. After that, backend will send an acknowledgement to frontend and it will show the success message to the administrators.


Latia CMS Modify Tag Information
Backend
Frontend
Start
Get the Tag ID
Request retrieving title and subtitle
Query the database for retrieving title and subtitle
Tag ID Exist
No
Show the error message
Yes
Return the title and subtitle
Display the title and subtitle
Get the modified title and subtitle
Request update the tag information
Send acknowledgement
Show the success message
End

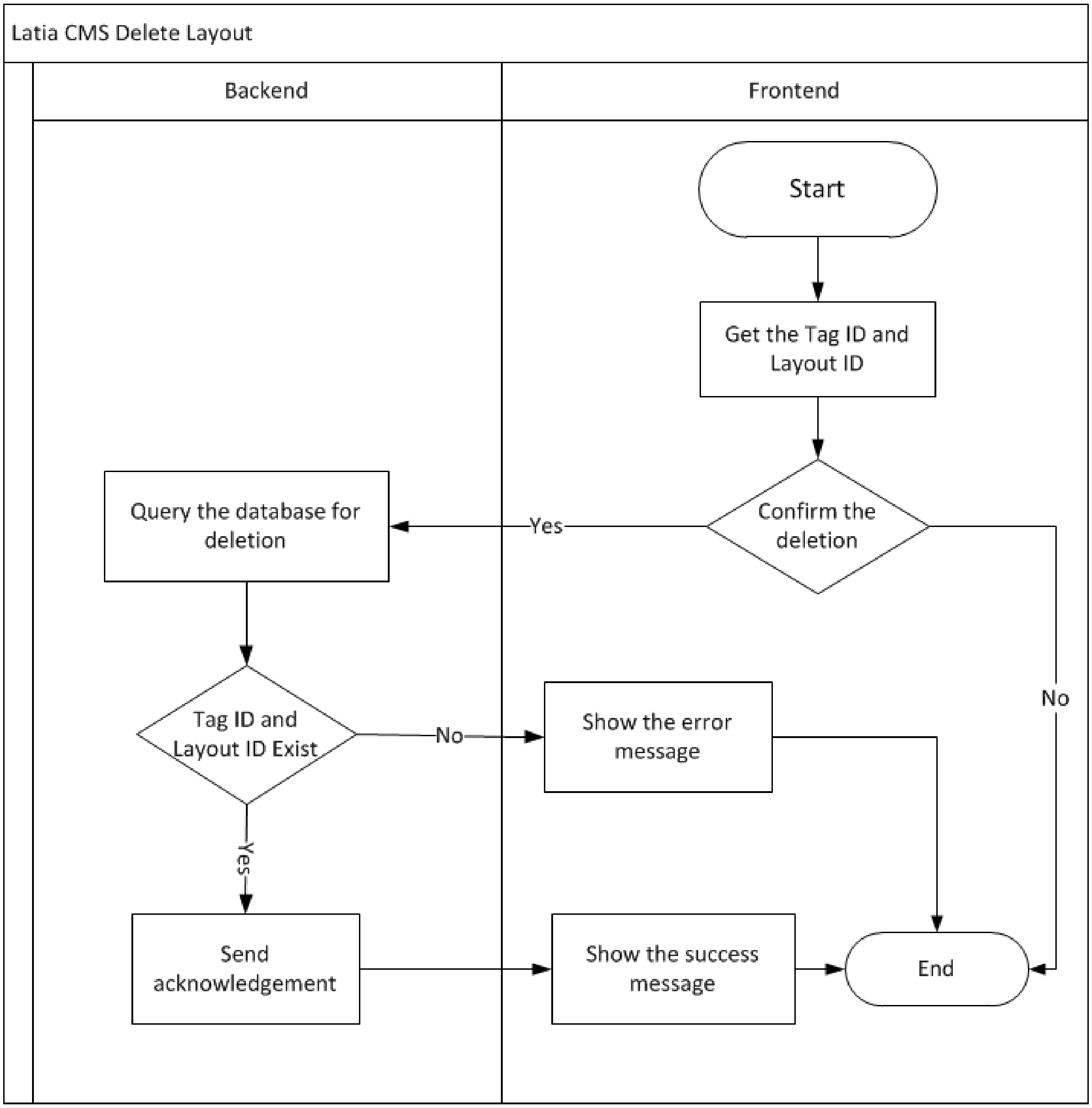
Modify Layout Information

The frontend will receive the Tag ID and Layout ID of the layout xml code that need to be modified from the administrators. It will request backend to retrieve the layout xml code. Then, the backend will query the database for getting the layout xml code. If the Tag ID or Layout ID does not exist, it will return fail message to the frontend and display error message. If exists, the backend will get the layout xml code from the database and pass to frontend. Frontend will show it to the administrators. After the administrators modified, frontend will pass it to backend and request update the layout information. When the update process complete, backend will acknowledge frontend and show the success message to inform administrator.


Latia CMS Modify Layout Information
Backend
Frontend
Start
Get the Tag ID and Layout ID
Request retrieving layout xml code
Query the database for retrieving layout xml code
Tag ID and Layout ID Exist
No
Show the error message
Yes
Return the layout xml code
Display the layout xml code
Get the modified layout xml code
Request update the layout information
Send acknowledgement
Show the success message
End

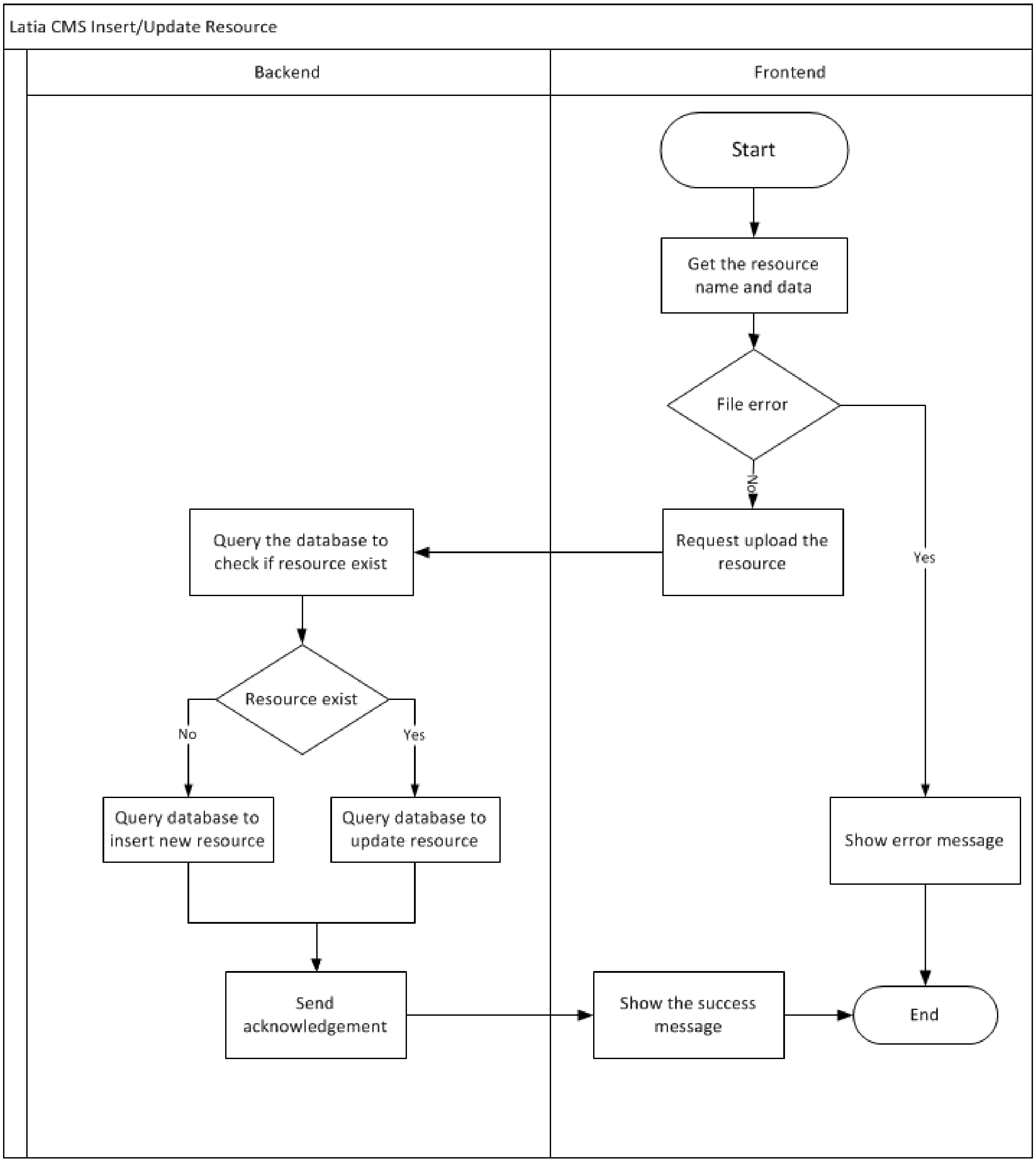
Delete Layout

The frontend will get the Tag ID and Layout ID of layout information that need to be deleted from the administrator. The frontend will ask the administrator for confirm before the deletion. If it is confirmed, the frontend will send the Tag ID and Layout to the backend. If the Tag ID or Layout ID does not exist, backend will send a fail message to frontend. An error message will be shown by frontend. If exists, the backend will query the database for deletion. After that, backend will send an acknowledgement to frontend. A success message will be displayed by the frontend to notice administrator the deletion is completed.


Latia CMS Delete Layout
Backend
Frontend
Start
Get the Tag ID and Layout ID
Confirm the deletion
Yes
No
Query the database for deletion
Tag ID and Layout ID Exist
No
Show the error message
Yes
Send acknowledgement
Show the success message
End

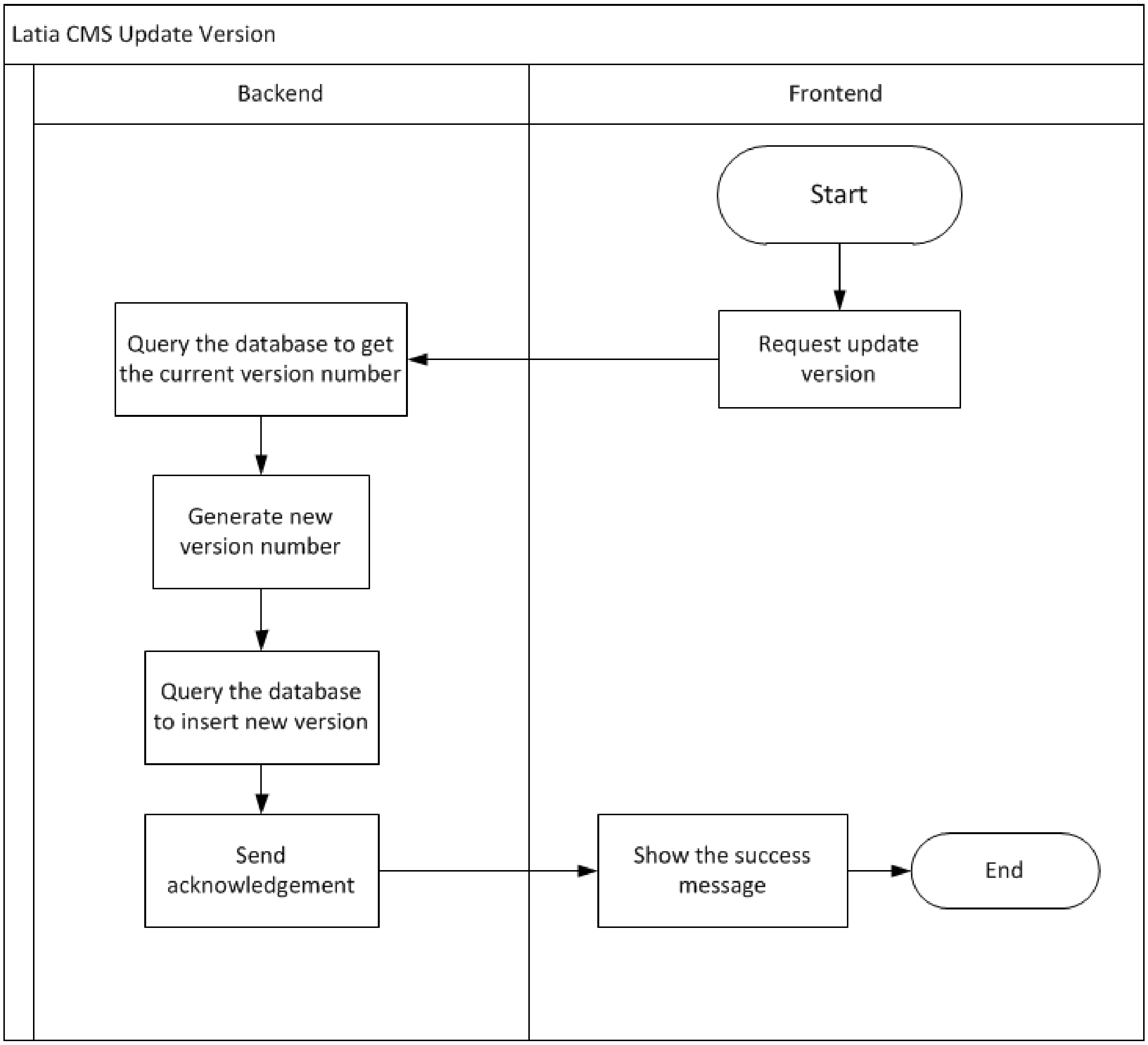
Insert/Update Resource

The administrator will pass the resource name and data to frontend. If the resource does not consist of any file error, frontend will send the resource name and data to the backend and request upload. If not, the frontend will show the error message to the administrator. After backend receive the resource name and information, it needs to know that if the request is insertion or update. So, backend will query the database to check if the resource exists. If it exists, backend will request the database to update the resource with the new data. If not, it will request the database to insert the new resource. After that, the backend will send the acknowledgement to frontend and it will show the success message to administrators.


Latia CMS Insert/Update Resource
Backend
Frontend
Start
Get the resource name and data
File error
No
Yes
Request upload the resource
Query the database to check if resource exist
Resource exist
No
Yes
Query database to insert new resource
Query database to update resource
Send acknowledgement
Show error message
Show the success message
End

Update Version

The administrators will request to update version in frontend when they want to release a new version. Frontend will request backup to update the current version. After that, backend will first query the database to get the most current version and generate a new version. Then, it will query the database to insert the new version with the current time. Finally, it sends an acknowledgement to frontend so that a success message will be displayed to inform the administrator.


Latia CMS Update Version
Backend
Frontend
Start
Request update version
Query the database to get the current version number
Generate new version number
Query the database to insert new version
Send acknowledgement
Show the success message
End

6.3 Android App

Our android application provides an interface between user and physical object. It divided into two parts: main application and NFC intent activity. Main application is required because the application needs to update itself regularly without reinstallation from Google Play.

Every time the main application is launched, it automatically downloads all the layouts and tag information from the internet to the in-app database. After that, it compares its version with the newest version and downloads the necessary resources. Our update system is designed to suit progressive download; therefore it won’t download unnecessary information and used up the bandwidth.

When a Latia NFC tag is attached, it will search the information of the corresponding tag from the in-app database. Several layouts will be displayed on the screen and allowing user to swipe between layouts. The title, subtitle of the tag is also displayed is also displayed.

For the ICT Expo, we have added a new module for deployment. This module allows user to revisit the tag which he has touched.

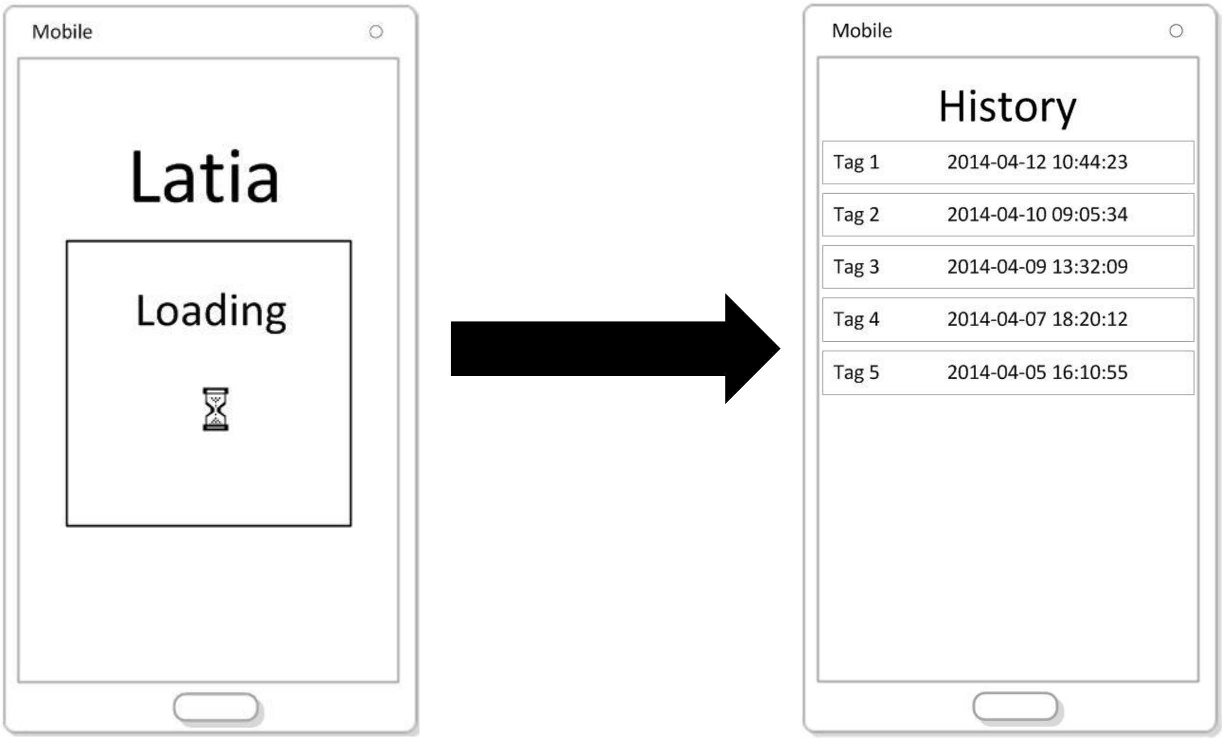
6.3.1 User Interface

The user interface is one of the most concerns in our project. A good user interface can attract the users to make use of the application. We think that a good interface not only about the fancy layout, but also the user experience. Thus, our interface mainly focuses on user experience. The following is the wireframes that show the basic user interface design of Latia.

Latia Android application can be launched in two ways: through main menu and through NFC tag. If user clicked the application in the main menu, it will be launched to an updating screen. However, Latia can also be launched by touching a Latia NFC tag. The NFC tag sends the NDEF message to the kernel then it will trigger the Latia application as well.

Main Application

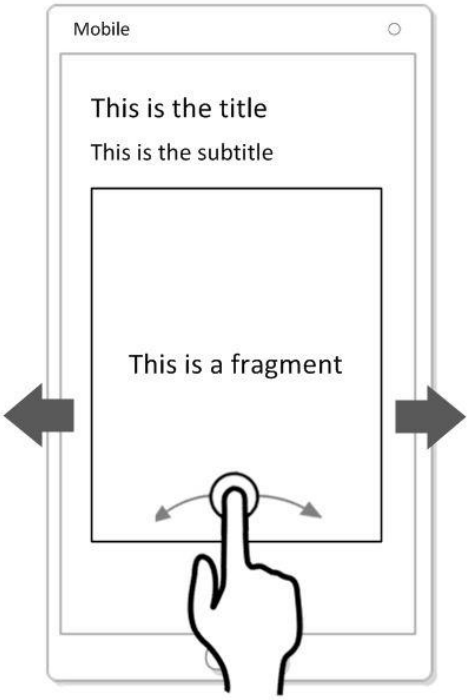
After user clicked the Latia logo from main menu, an updating page will show up. A loading logo will keep animating and a progress bar indicates the update progress. After the whole updating process is finished, it will turns into a history page showing all the tags in the past. User can click any of them to go back to the corresponding NFC page.



Tag 1	2014-04-12 10:44:23
Tag 2	2014-04-10 09:05:34
Tag 3	2014-04-09 13:32:09
Tag 4	2014-04-07 18:20:12
Tag 5	2014-04-05 16:10:55

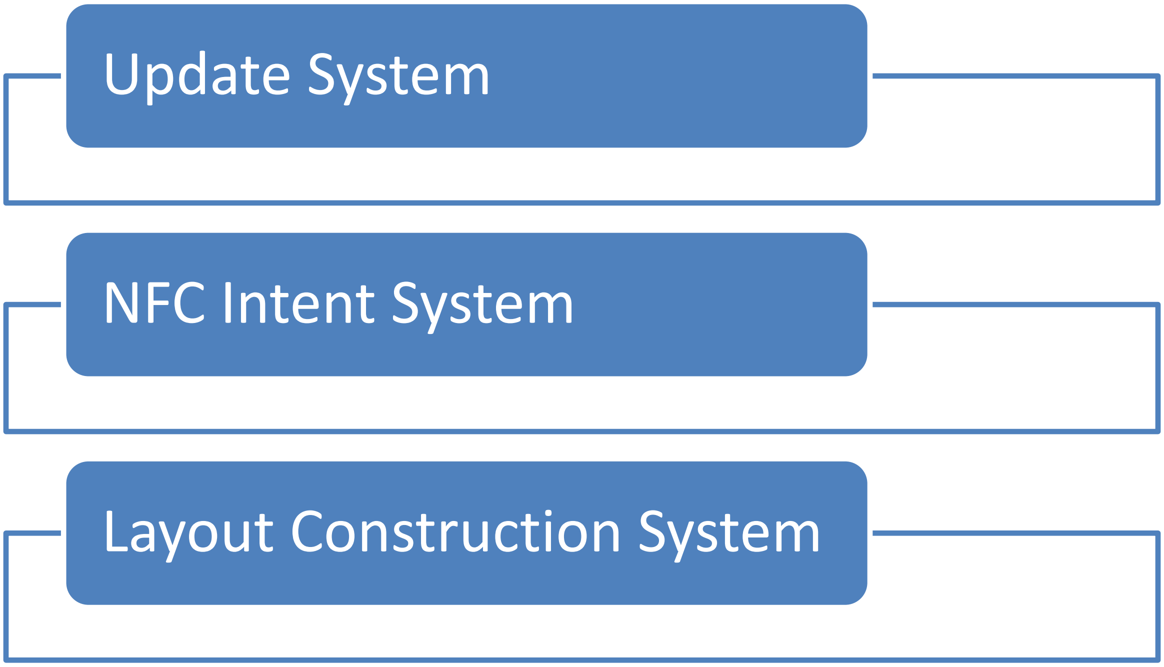
NFC Activity

After the device attached to a Latia NFC tag, the NFC activity will immediately popup. User can see the title and subtitle in this activity. A set of pages of different layout are also appeared in the middle of the screen. User can swipe left or right to switch pages and swipe up or down to scroll. Depends on the tag, different tag provides different set of pages.



6.3.2 Modules

Latia Android Application divided into 3 modules: update system, NFC intent system and layout construction system.



Update System

NFC Intent System

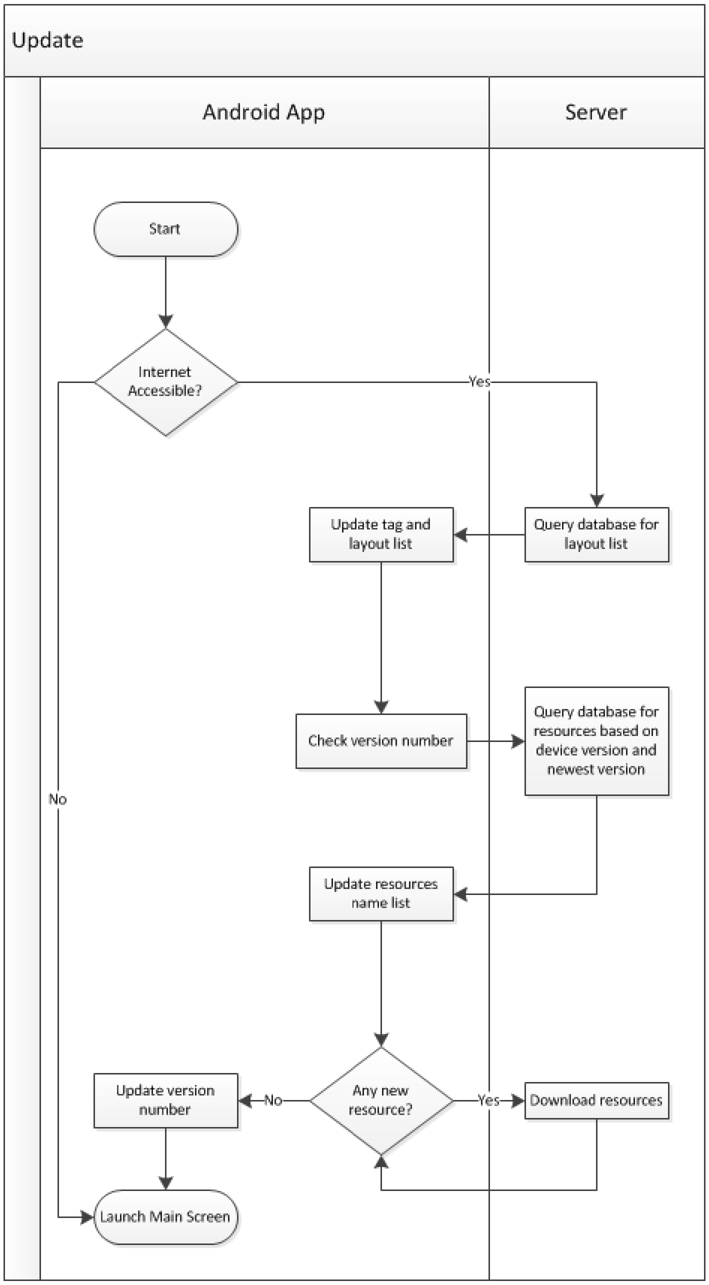
Layout Construction System

Update System

In order to provide the most recent information without frequent reinstallation, we have added an in-app update system. The update system will be triggered whenever the application is launched through clicking the icon and the internet is accessible.

Workflow

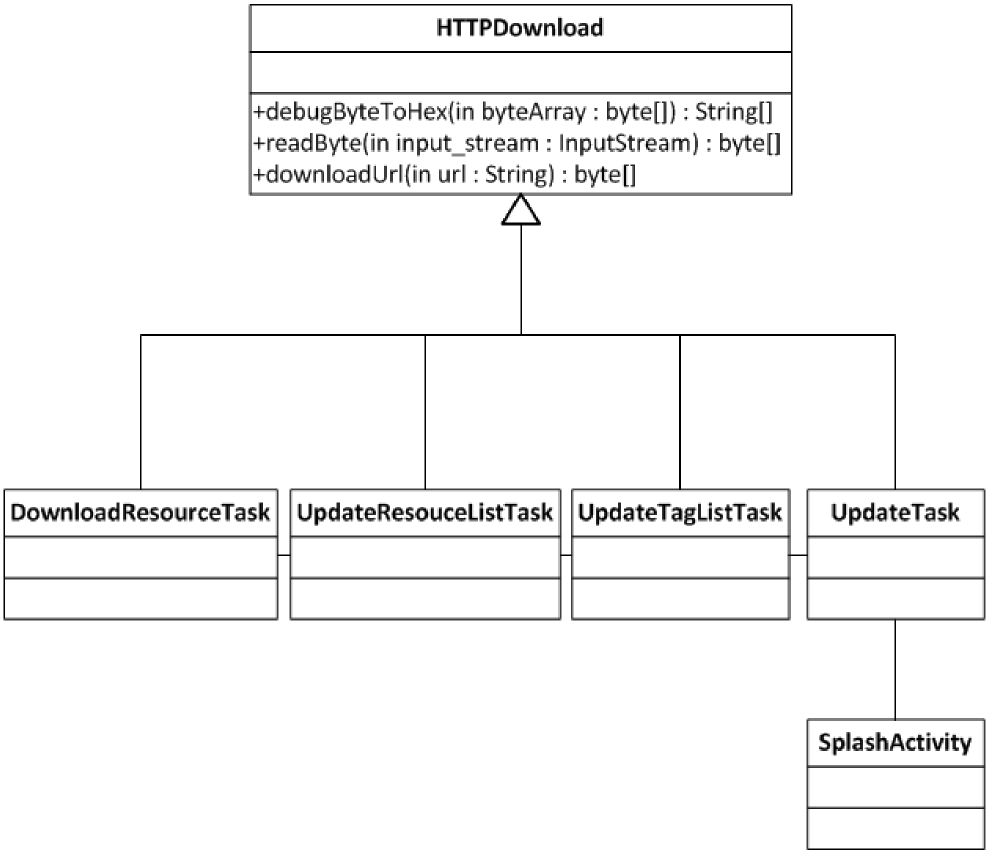
In the beginning, it downloads all the tag titles and subtitles as well as the tag count. Then it base on the tag count to download the layouts one by one. This process will run every time to keep layout update. Since, some layouts contain images, these images are also needs to be updated. However, some images are already in the application database, it is unnecessary to download them again especially when the file size is large.


Update
Android App
Server
Start
Internet Accessible?
Yes
No
Query database for layout list
Update tag and layout list
Check version number
Query database for resources based on device version and newest version
Update resources name list
Any new resource?
Yes
Download resources
No
Update version number
Launch Main Screen

We have used a version code to control the update content. By sending current version to the server, the server decides which image or resource is required to send to the Android application. It also deletes unnecessary resources from the application.

Methods

The classes in this module are shown below.


HTTPDownload
+debugByteToHex(in byteArray : byte[]) : String[]
+readByte(in input_stream : InputStream) : byte[]
+downloadUrl(in url : String) : byte[]
DownloadResourceTask
UpdateResouceListTask
UpdateTagListTask
UpdateTask
SplashActivity

HTTPDownload

This is the background asynchronized task that does HTTP communication by creating a thread. This is the core class, since all the update procedures are based on HTTP. Classes can do HTTP download by inherit this class and use *downloadUrl()* method.

downloadUrl()

This method inputs the URL of the target website and fetches data through HTTP, then return the result as byte array. The application layer protocol is handled in this method. It also checks the status of return value and throws IOExeption.

UpdateTask

This task checks the internet accessibility and initiates the sequence of update tasks. It also downloads the metadata of tag and layout list such as number of tags and number of layouts. After the task is finished, it triggers the UpdateTagListTask.

UpdateTagListTask

This task updates all the layouts. After update it triggers the UpdateResourceListTask.

UpdateResourceListTask

This task downloads the name list of resources by providing the current version code to server. Then for each resource in the name list, it calls DownloadResourceTask to update every new resource.

UpdateResourceTask

This task simply downloads a file from the internet.

SplashActivity

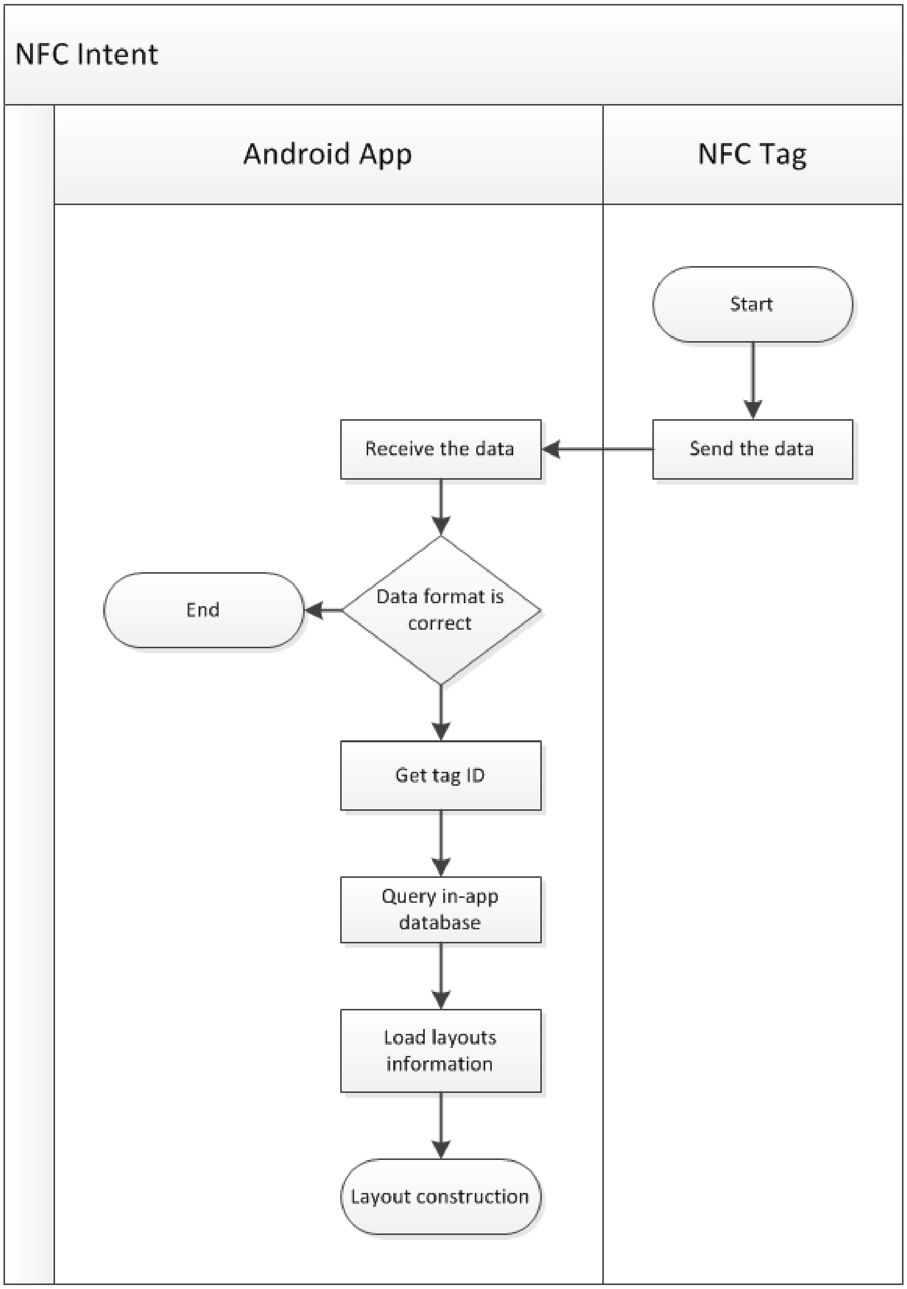
This is the update screen activity. While this activity is active, it will trigger UpdateTask to update the tag information. During the updating process, it keeps track to the update progress and show to the user. After the update is complete or no update is needed, it will intent the main activity to show history information.

NFC Intent System

NFC tag intent dispatch system and foreground dispatch system. Latia has registered to a self creates external type. The registration can be found in the manifest file. The intent dispatch system allows Latia to pop up when user scans the NFC Tag.

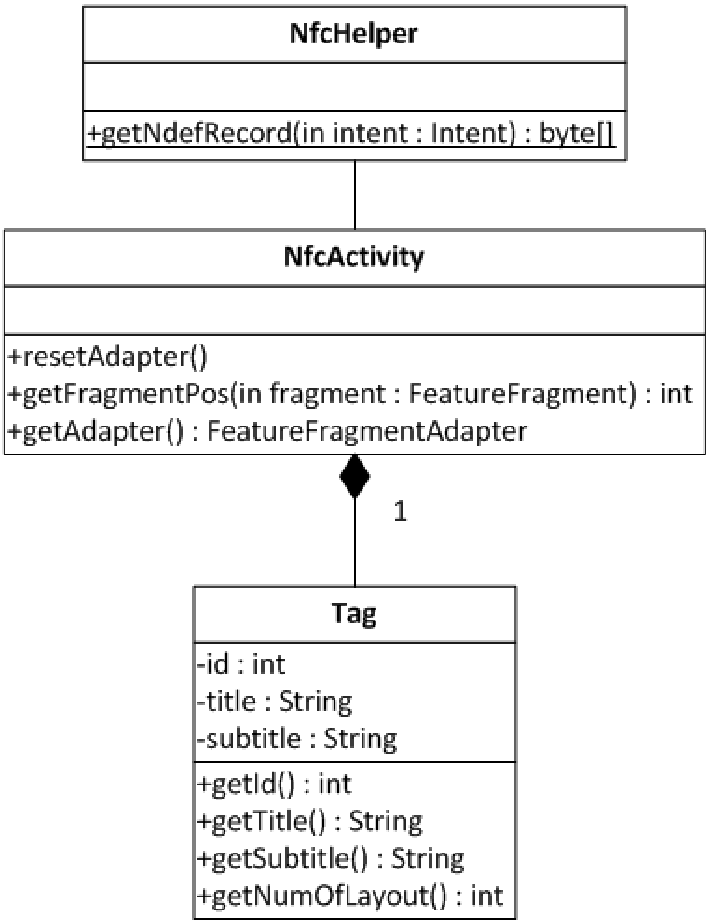
Workflow

After receiving the NFC data structure, the kernel looks for Latia and sends the NFC message to the application. Our application will validate the data format and get the NFC tag ID from the message. Then it searches the corresponding layouts from in-app database. After that, it delivers the layouts to the layout construction system which is a parser.


NFC Intent
Android App
NFC Tag
Start
Send the data
Receive the data
Data format is correct
End
Get tag ID
Query in-app database
Load layouts information
Layout construction

Method

NfcActivity is the activity called by NFC intent. It uses NfcHelper to decode the NDEF message and creates a Tag.



NfcActivity

This activity shows the information of the tag such as title and subtitle. It also contains all the layouts of the tag depending on the tag ID from Latia NFC tag.

NfcHelper

This is a static class that helps NfcActivity to analyze the NDEF data structure.

getNdefRecord()

This method returns a stream of bytes from the intent. It gets the NDEF raw message and extract the Latia record from the message. If it is success, the whole payload of the record will be returned.

Tag

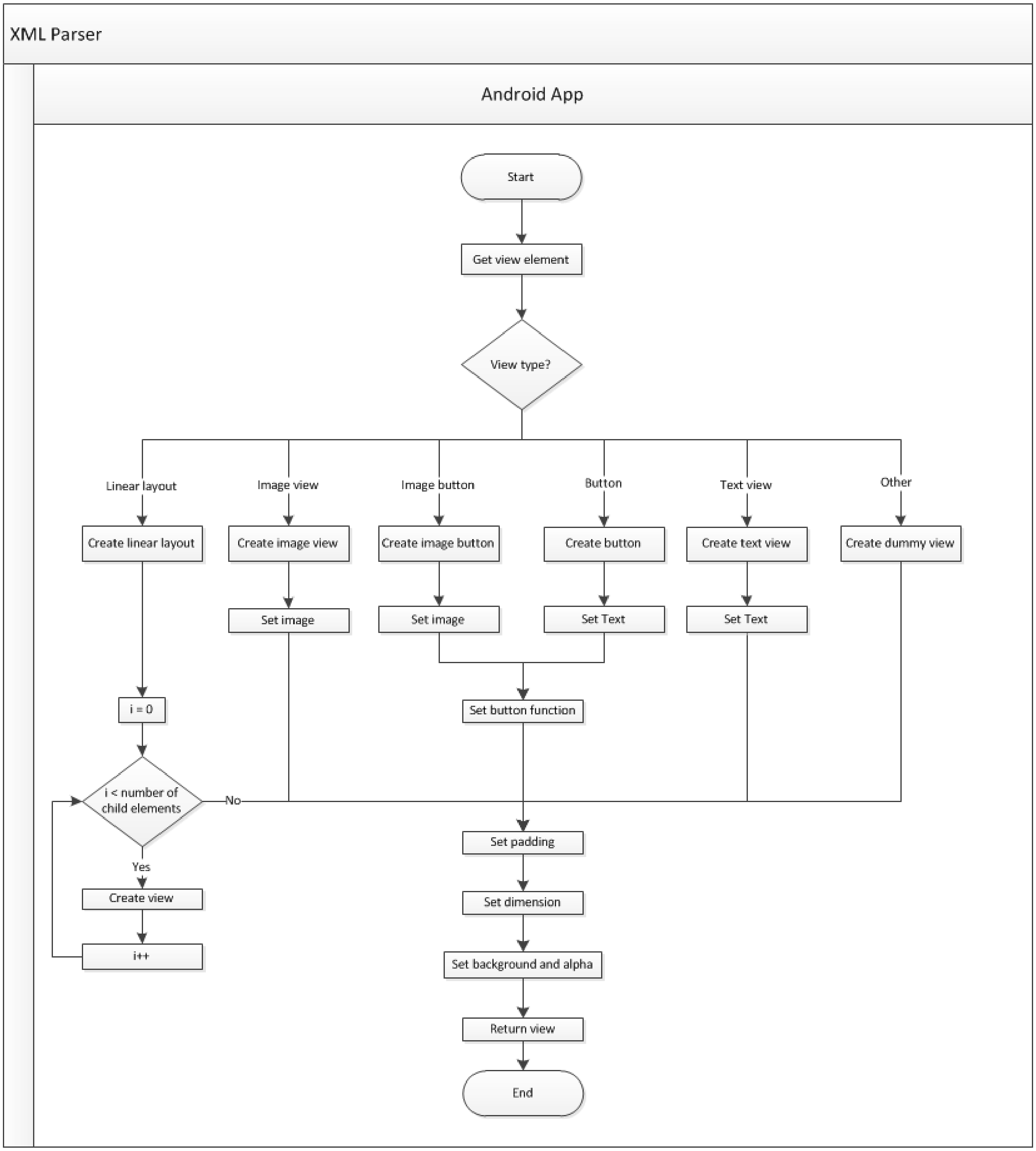
This class represents a logical NFC tag. It contains all the information related to the tag such as id, title and subtitle.

Layout Construction System

Allowing administrator to modify the layout dynamically is one of our goals in our project. However, the Android built-in XML layout parser is not allowed to work in runtime, due to the complexity and efficiency issue. Hence, we have to build our own XML parser for layout design.

Workflow

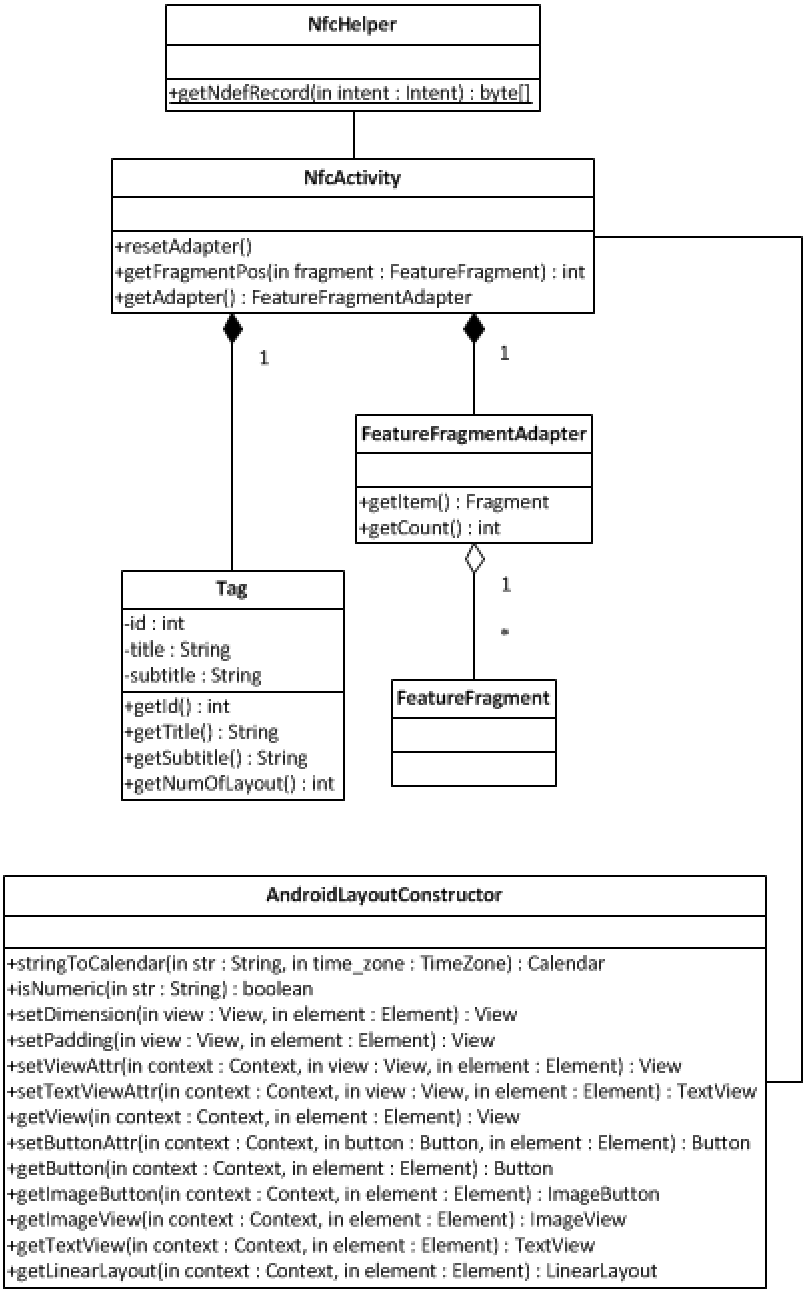
In the beginning, we get the layout information from the NFC intent system. For each layout, it reads the root element from the XML file. Base on the type of element, it creates the view for the element. If the element is a linear layout, it will look at the child elements of the linear layout and continue building views. After view is created, it adds attributes to the view and returns the view. In this manner, the view will be created recursively.


XML Parser
Android App
Start
Get view element
View type?
Linear layout
Image view
Image button
Button
Text view
Other
Create linear layout
Create image view
Create image button
Create button
Create text view
Create dummy view
Set image
Set image
Set Text
Set Text
Set button function
i = 0
i < number of child elements
No
Yes
Create view
i++
Set padding
Set dimension
Set background and alpha
Return view
End

Method

The major class in this system is AndroidLayoutConstructor which act as an XML parser and view builder. FeatureFragmentAdapter helps NfcActivity to handle all the paging issues. FeatureFragment is the container where all the layout views should go.

This structure allows developer to separate concerns between paging and layout, it also enhance the robustness of the whole process.


NfcHelper
+getNdefRecord(in intent : Intent) : byte[]
NfcActivity
+resetAdapter()
+getFragmentPos(in fragment : FeatureFragment) : int
+getAdapter() : FeatureFragmentAdapter
1
1
FeatureFragmentAdapter
+getItem() : Fragment
+getCount() : int
1
*
Tag
-id : int
-title : String
-subtitle : String
+getId() : int
+getTitle() : String
+getSubtitle() : String
+getNumOfLayout() : int
FeatureFragment
AndroidLayoutConstructor
+stringToCalendar(in str : String, in time_zone : TimeZone) : Calendar
+isNumeric(in str : String) : boolean
+setDimension(in view : View, in element : Element) : View
+setPadding(in view : View, in element : Element) : View
+setViewAttr(in context : Context, in view : View, in element : Element) : View
+setTextViewAttr(in context : Context, in view : View, in element : Element) : TextView
+getView(in context : Context, in element : Element) : View
+setButtonAttr(in context : Context, in button : Button, in element : Element) : Button
+getButton(in context : Context, in element : Element) : Button
+getImageButton(in context : Context, in element : Element) : ImageButton
+getImageView(in context : Context, in element : Element) : ImageView
+getTextView(in context : Context, in element : Element) : TextView
+getLinearLayout(in context : Context, in element : Element) : LinearLayout

FeatureFragment

This is the fragment class for the NfcActivity, each layout has its own fragment to display. After creating the view of a layout, the view has to map to one feature fragment and let the adapter to do its work.

FeatureFragmentAdapter

This class contains an array of FeatureFragment allowing them to display in the activity. The adapter handles all controls of fragments, for example listening to user’s finger to flip page and defining the basic layout of fragments.

AndroidLayoutConstructor

This is a static class contains all the XML parsing methods. The method received an XML element and returns a view.

setDimension

Set the dimension of the view such as width, height and weight.

setPadding

Set the padding of the view in the surrounding: top, bottom, left, right.

setViewAttr

Set general attributes for all views, such as background color, background image and alpha value *setButtonAttr*

Set button attributes to a Button or imageButton object, it sets the function of the button. For now, we allows designer to set button for like, dislike, add contact, redirect URL or add event to calendar.

setTextViewAttr

Set text attributes to TextView object including button. Text attributes includes setting text, text size, text color.

getTextView

Returns a TextView object with all attribute set.

getImageView

Returns an ImageView object with all attribute set.

getButton

Returns a Button object with all attribute set.

getImageButton

Returns an ImageButton object with all attribute set.

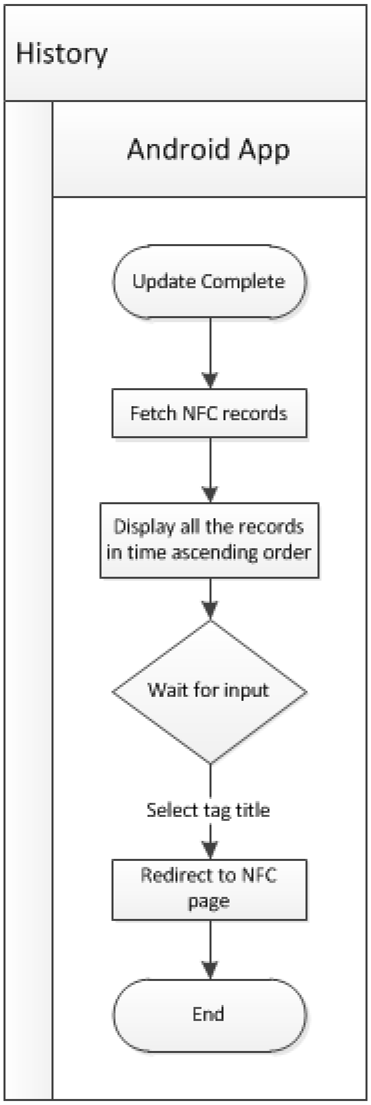
getLinearLayout

Return a LinearLayout which already contains all its child views and all attribute are set.

History

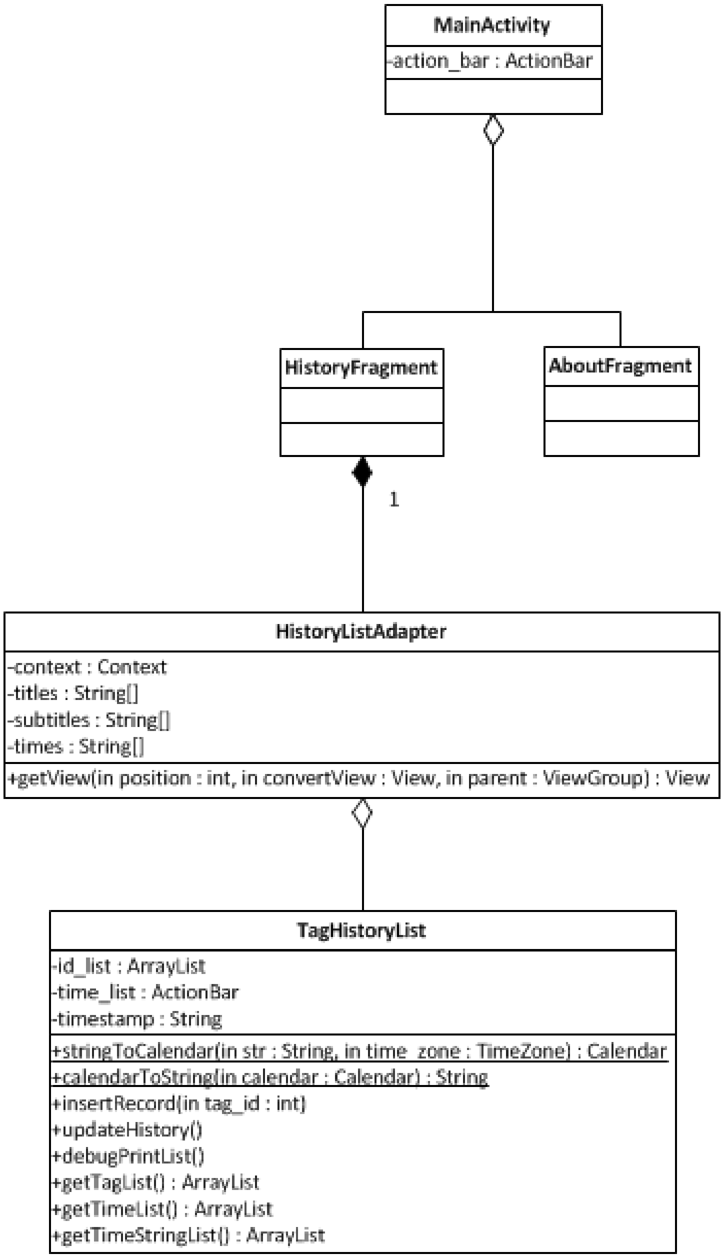
This module is added for Expo. Because after people used Latia, they usually needs to review what they have seen. A history page becomes handy for small scale deployment. This page is run after the main application is launched (After update completion). It shows all the record in the list allowing user to click it and immediately returns to the corresponding NFC page.

Workflow


History
Android App
Update Complete
Fetch NFC records
Display all the records in time ascending order
Wait for input
Select tag title
Redirect to NFC page
End

Method

In the main activity, it contains two fragments: history fragment and about fragment. In the history fragment, it has an adapter to handle all the controls on each history entry.


MainActivity
-action_bar : ActionBar
HistoryFragment
AboutFragment
1
HistoryListAdapter
-context : Context
-titles : String[]
-subtitles : String[]
-times : String[]
+getView(in position : int, in convertView : View, in parent : ViewGroup) : View
TagHistoryList
-id_list : ArrayList
-time_list : ActionBar
-timestamp : String
+stringToCalendar(in str : String, in time_zone : TimeZone) : Calendar
+calendarToString(in calendar : Calendar) : String
+insertRecord(in tag_id : int)
+updateHistory()
+debugPrintList()
+getTagList() : ArrayList
+getTimeList() : ArrayList
+getTimeStringList() : ArrayList

TagHistoryList

This is a list contains all the tags that have been read, it maintains a time ascending order.

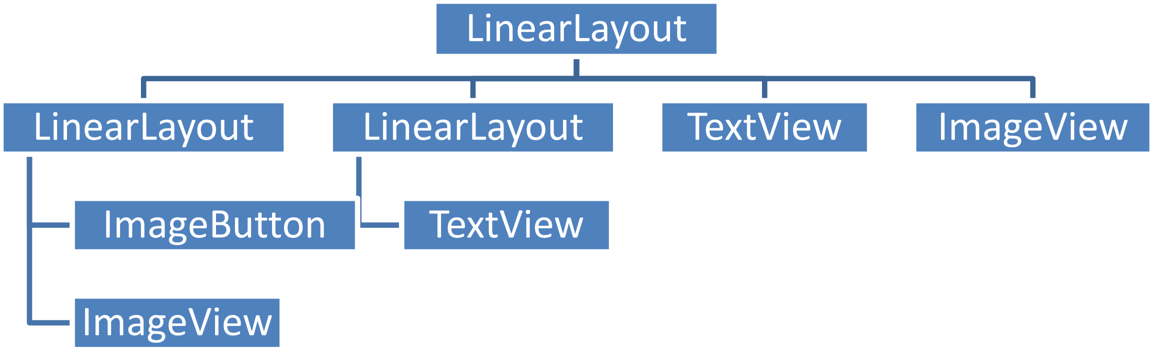
HistoryListAdapter

Contains a TagHistoryList and displays on the HistoryFragment. When user clicks one of the entries, It redirects user to the corresponding NFC page.

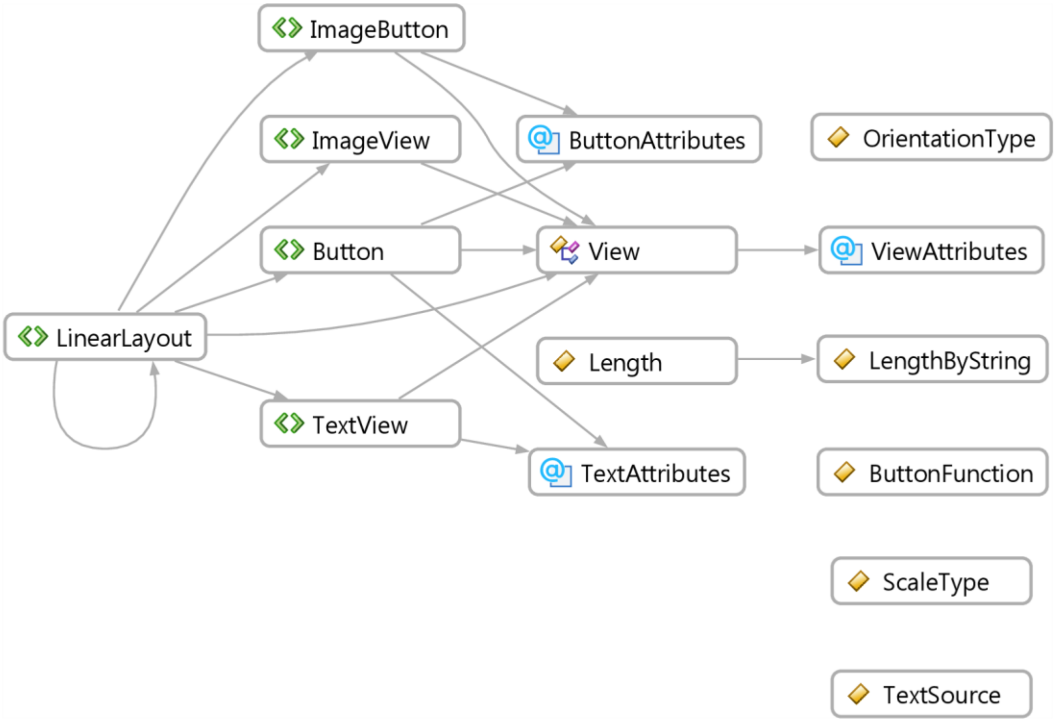
6.4 XML

The layout defines the visual structure for a user interface. To obtain a flexible structure for designer, complete layout syntax has to be developed. We take reference to several XML rules and developed our own XML schema. Basically, it is designed base on the Android XML layout format.

Here is an example of an XML layout format.



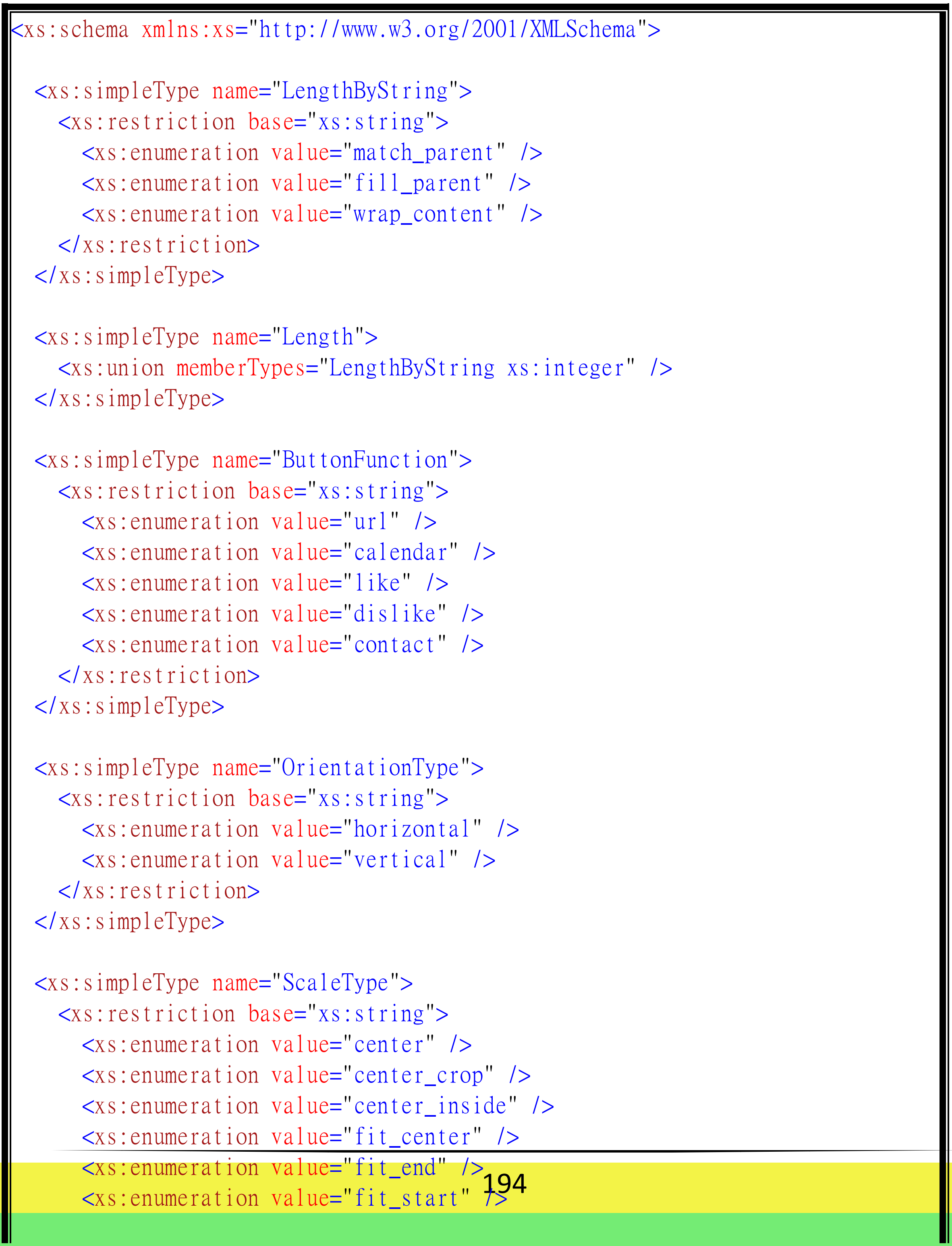
We adopt the vocabulary used in android XML layout. Hence it is easy for android designer to use. We have add some restriction on the original rules and add some preset functionalities to the button.



6.4.1 Fundamental elements

We have included five elements: LinearLayout, TextView, ImageView, ImageButton, Button.

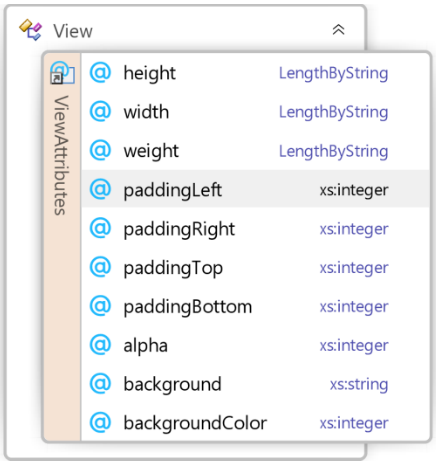
The schema is described as follows.



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="LengthByString">
    <xs:restriction base="xs:string">
      <xs:enumeration value="match_parent" />
      <xs:enumeration value="fill_parent" />
      <xs:enumeration value="wrap_content" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="Length">
    <xs:union memberTypes="LengthByString xs:integer" />
  </xs:simpleType>

  <xs:simpleType name="ButtonFunction">
    <xs:restriction base="xs:string">
      <xs:enumeration value="url" />
      <xs:enumeration value="calendar" />
      <xs:enumeration value="like" />
      <xs:enumeration value="dislike" />
      <xs:enumeration value="contact" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="OrientationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="horizontal" />
      <xs:enumeration value="vertical" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ScaleType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="center" />
      <xs:enumeration value="center_crop" />
      <xs:enumeration value="center_inside" />
      <xs:enumeration value="fit_center" />
      <xs:enumeration value="fit_end" />
      <xs:enumeration value="fit_start" />
    <xs:enumeration value="fit_xy" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="TextSource">
  <xs:restriction base="xs:string">
    <xs:enumeration value="like_count" />
    <xs:enumeration value="dislike_count" />
  </xs:restriction>
</xs:simpleType>

<xs:attributeGroup name="ViewAttributes">
  <xs:attribute name="height" type="LengthByString" use="required"/>
  <xs:attribute name="width" type="LengthByString" use="required"/>
  <xs:attribute name="weight" type="LengthByString"/>
  <xs:attribute name="paddingLeft" type="xs:integer"/>
  <xs:attribute name="paddingRight" type="xs:integer"/>
  <xs:attribute name="paddingTop" type="xs:integer"/>
  <xs:attribute name="paddingBottom" type="xs:integer"/>
  <xs:attribute name="alpha" type="xs:integer"/>
  <xs:attribute name="background" type="xs:string"/>
  <xs:attribute name="backgroundColor" type="xs:integer"/>
</xs:attributeGroup>

<xs:attributeGroup name="ButtonAttributes">
  <xs:attribute name="onClick" type="ButtonFunction" use="required"/>
</xs:attributeGroup>

<xs:attributeGroup name="TextAttributes">
  <xs:attribute name="text" type="xs:string"/>
  <xs:attribute name="textSize" type="xs:integer"/>
  <xs:attribute name="textColor" type="xs:integer"/>
  <xs:attribute name="textSrc" type="TextSource"/>
</xs:attributeGroup>

<xs:complexType name="View">
  <xs:attributeGroup ref="ViewAttributes"/>
</xs:complexType>
```

```
<xs:element name="LinearLayout">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="View">
        <xs:sequence>
          <xs:element ref="LinearLayout" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Button" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ImageButton" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="TextView" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="ImageView" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="orientation" type="OrientationType"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="Button">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="View">
        <xs:attributeGroup ref="ButtonAttributes"/>
        <xs:attributeGroup ref="TextAttributes"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="TextView">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="View">
        <xs:attributeGroup ref="TextAttributes"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="ImageView">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="View">
          <xs:attribute name="src" type="xs:string"/>
          <xs:attribute name="scaleType" type="ScaleType"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="ImageButton">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="View">
          <xs:attributeGroup ref="ButtonAttributes"/>
          <xs:attribute name="imageUp" type="xs:string"/>
          <xs:attribute name="imageDown" type="xs:string"/>
          <xs:attribute name="scaleType" type="ScaleType"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

View



View

ViewAttributes

height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer

This is the basic component of visual element. All other elements are inherited from this element, which also inherited the attributes as well. In this view, it contains dimension information: width and height. There are three values in this field. wrap_content tells the view to size itself to the dimensions required by its content. match_parent tells the view to become as big as its parent view group will allow. If number is entered, the view will set exactly this length in unit dip.

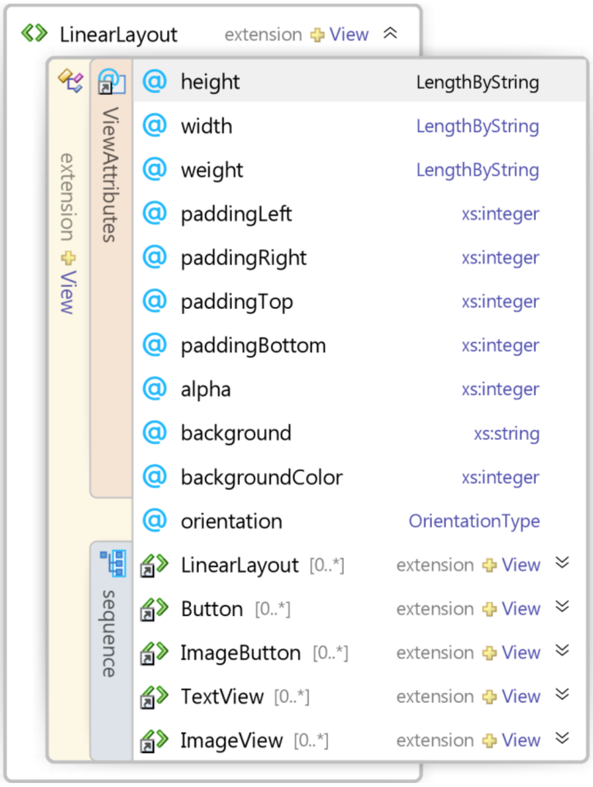
Sometimes, weight is useful to set dimension instead of using width and height. By setting a fraction to this attribute, it will look for all the views in the same LinearLayout and compare the weights. It affects the scale of the view compare with other views in the same parent node.

Padding values defines the margin between the view and the surrounding. By entering an integer to the attribute, it will set the border thickness as the value in unit dip.

There are two background types: image and color. Setting a filename to background gives an image background from the specified filename. Setting a color value in integer to backgroundColor gives a pure color for background.

We also allow transparency in the layout. Setting the alpha value gives a transparent effect.

LinearLayout



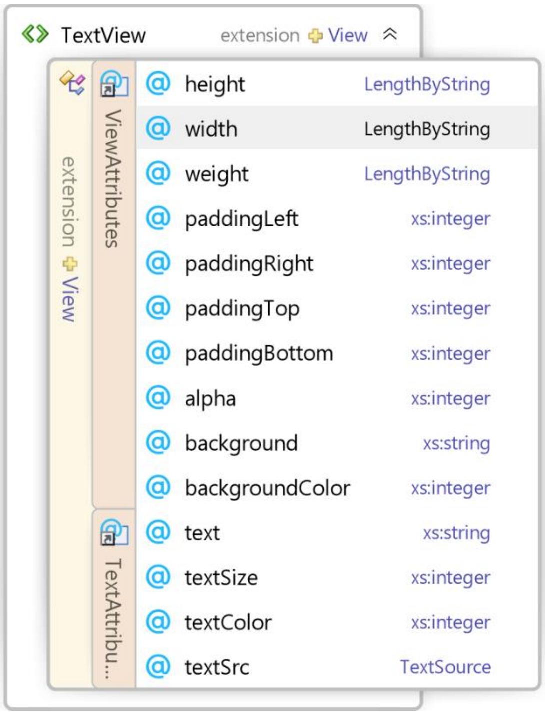
height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer
orientation	OrientationType
LinearLayout [0..*]	extension View
Button [0..*]	extension View
ImageButton [0..*]	extension View
TextView [0..*]	extension View
ImageView [0..*]	extension View

It is the only view that allows child views. It can contain any number of views and displays all the views using two alignments: vertical and horizontal.

If orientation is set to horizontal, all the child views will be aligned in a row and weight controls the width.

If orientation is set to vertical, all the child views will be aligned in a column and weight controls the height.

TextView



TextView extension View

extension View

ViewAttributes

height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer

TextAttribu...

text	xs:string
textSize	xs:integer
textColor	xs:integer
textSrc	TextSource

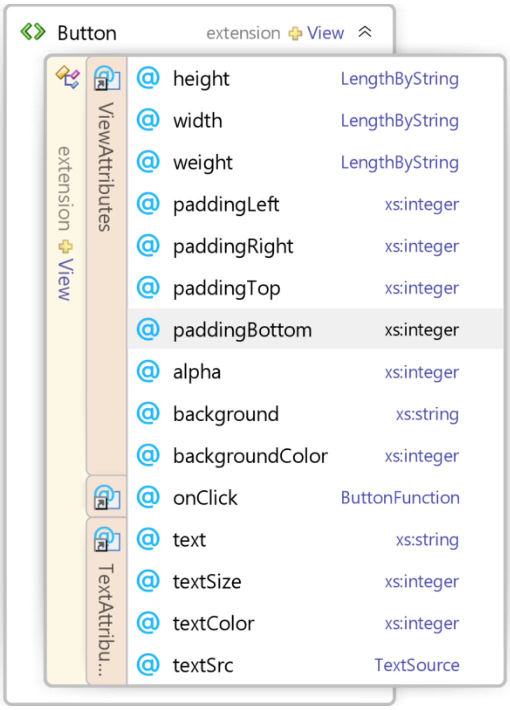
This view is used to show text only.

textSize is in unit of sp, it will scale for different size of screen.

textColor can be set using an integer color value, similar to background color.

textSrc can be set as an URL, which will automatically updates through internet, but this requires user to have internet access.

Button



Button extension View

extension View

ViewAttributes

height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer
onClick	ButtonFunction
text	xs:string
textSize	xs:integer
textColor	xs:integer
textSrc	TextSource

TextAttribu...

Button is has all the attributes from TextView. The only difference is the onClick attribute.

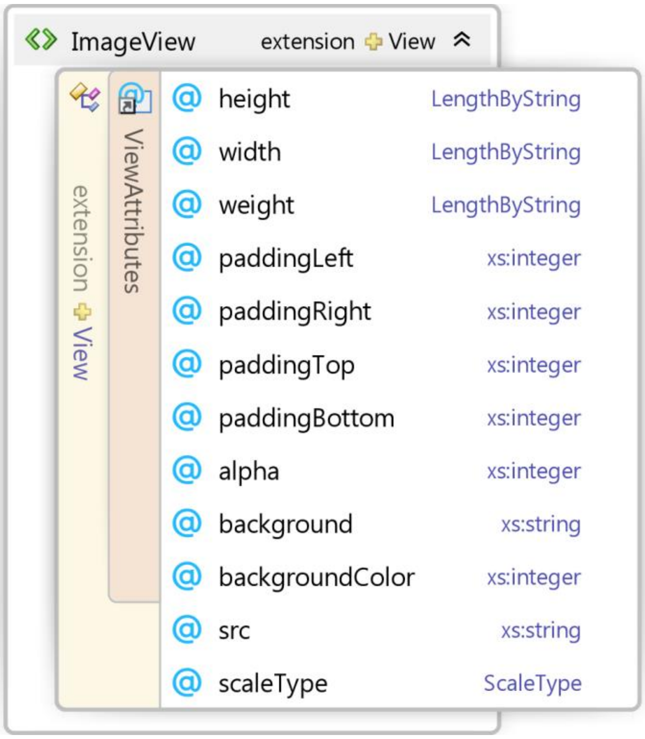
There are 5 functionalities for button: like, dislike, url, calendar and contact.

In our database, on default we assign a like count and dislike count for each NFC tag. Designer only needs to specify which button is like and which is dislike, then the like/dislike function is automatically added.

Designer can also set a URL for the button. When user clicks the button, it will automatically redirects to that URL.

Adding event to calendar and adding contact is also allowed in the button functionality.

ImageView



ImageView extension View

extension View

ViewAttributes

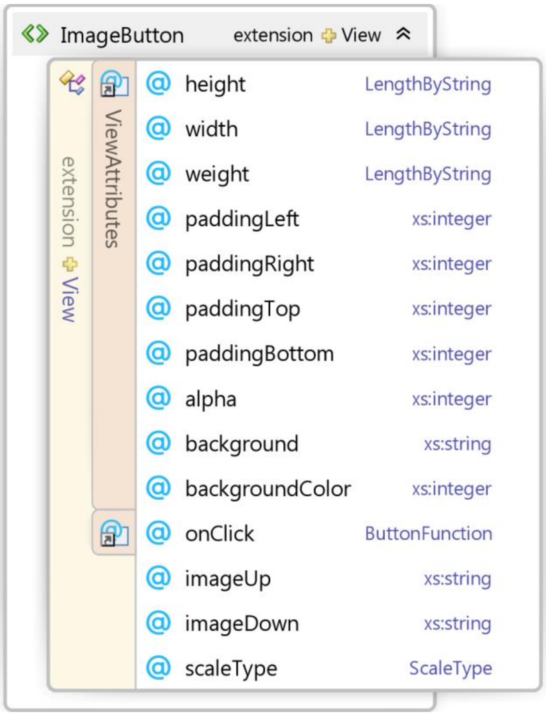
height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer
src	xs:string
scaleType	ScaleType

Image view allows designer to put image in the layout. There are two attributes are specified to do the job: src and scaleType. Setting a file name to src tells the view to show an image from that file. However, each image has its own resolution and density, if we use it with height and width it may cause ambiguous and difficult to compatible with different devices. Hence, we have added a scaleType to clarify different situation.

Different scaleType is listed in the following table

Values	
center	Center the image in the view, but perform no scaling.
center_crop	Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding).
center_inside	Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding).
fit_center	Compute a scale that will maintain the original src aspect ratio, but will also ensure that src fits entirely inside dst.
fit_end	Compute a scale that will maintain the original src aspect ratio, but will also ensure that src fits entirely inside dst. At least one axis (X or Y) will fit exactly. END aligns the result to the right and bottom edges of dst.
fit_start	Compute a scale that will maintain the original src aspect ratio, but will also ensure that src fits entirely inside dst. At least one axis (X or Y) will fit exactly. START aligns the result to the left and top edges of dst
fit_xy	Scale in X and Y independently, so that src matches dst exactly

ImageButton



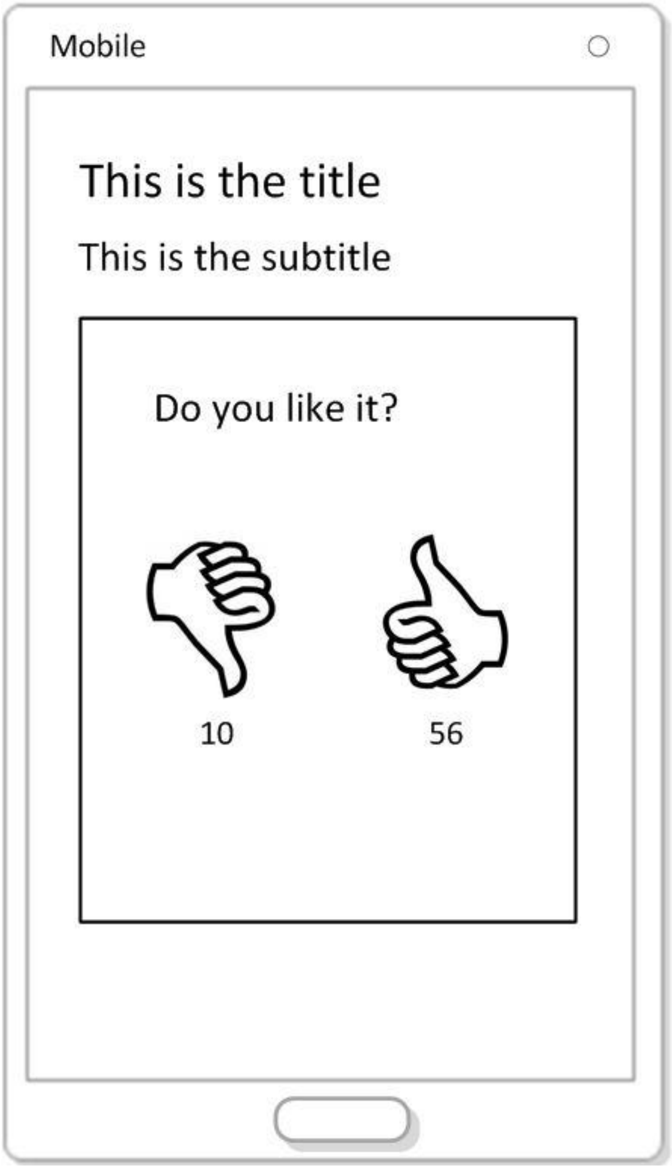
height	LengthByString
width	LengthByString
weight	LengthByString
paddingLeft	xs:integer
paddingRight	xs:integer
paddingTop	xs:integer
paddingBottom	xs:integer
alpha	xs:integer
background	xs:string
backgroundColor	xs:integer
onClick	ButtonFunction
imageUp	xs:string
imageDown	xs:string
scaleType	ScaleType

ImageButton is the combination of ImageView and Button. It allows designer to set two images. imageUp specifies the image in normal state and imageDown specifies the image when the button has been pressed.

6.4.2 Layout Design

We have designed several layouts which have potential usage for many applications.

Voting



Mobile

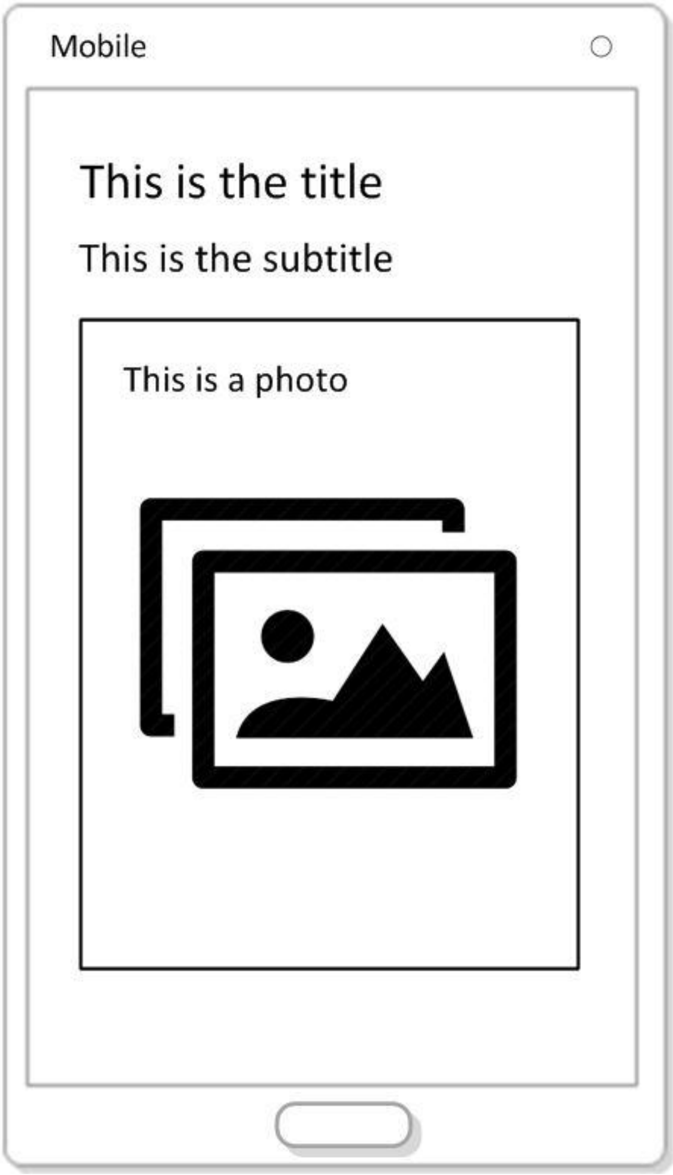
This is the title

This is the subtitle

Do you like it?

This layout allows user to vote up and vote down to a specific place.

Photos/Image



Mobile

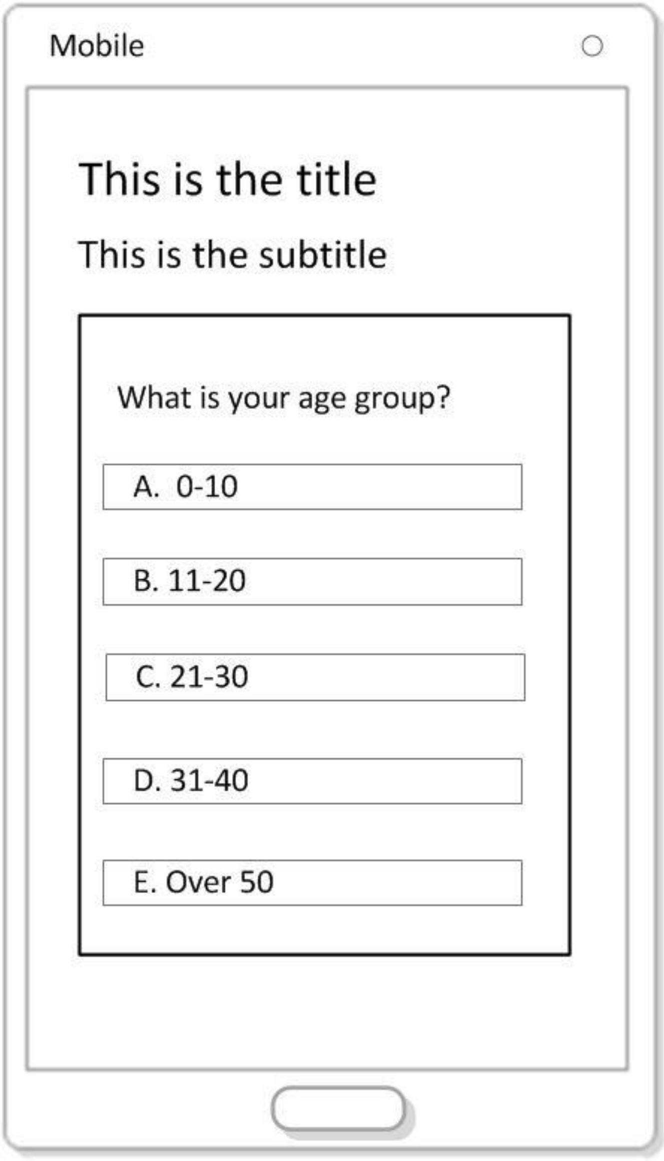
This is the title

This is the subtitle

This is a photo

Depends on the purpose of the tag, it shows the image for information.

Questionnaire



This is the title

This is the subtitle

What is your age group?

A. 0-10

B. 11-20

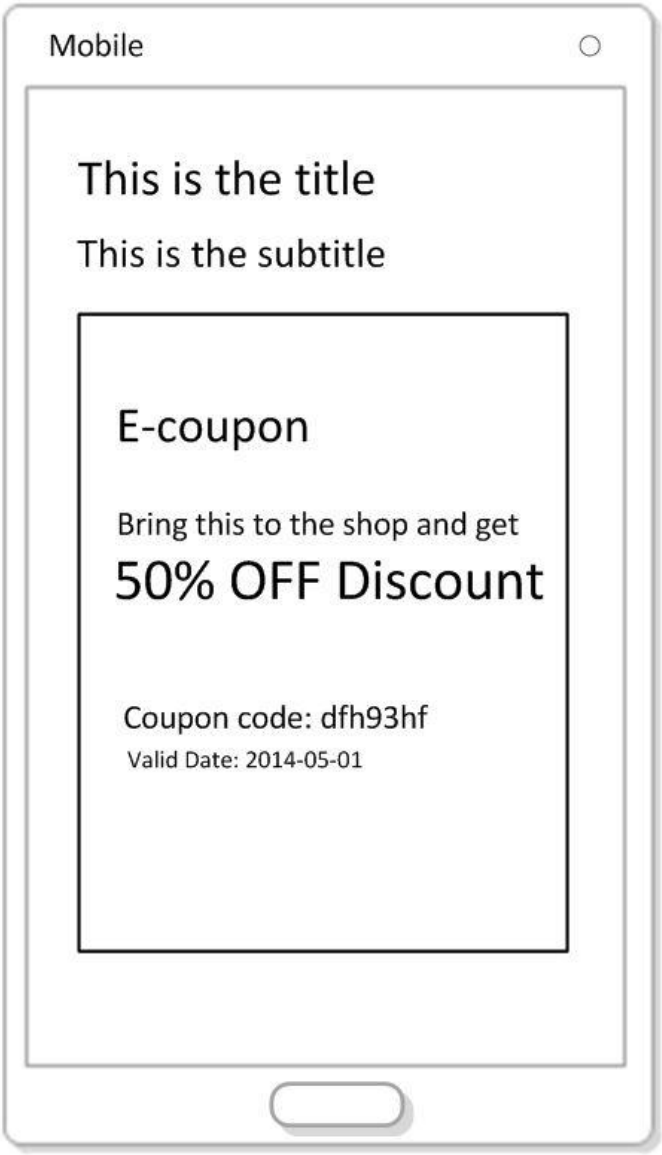
C. 21-30

D. 31-40

E. Over 50

Designer can set different kinds of questionnaire to gather statistics and comments.

Special Offer/ Electronic Coupons



This is the title

This is the subtitle

E-coupon

Bring this to the shop and get

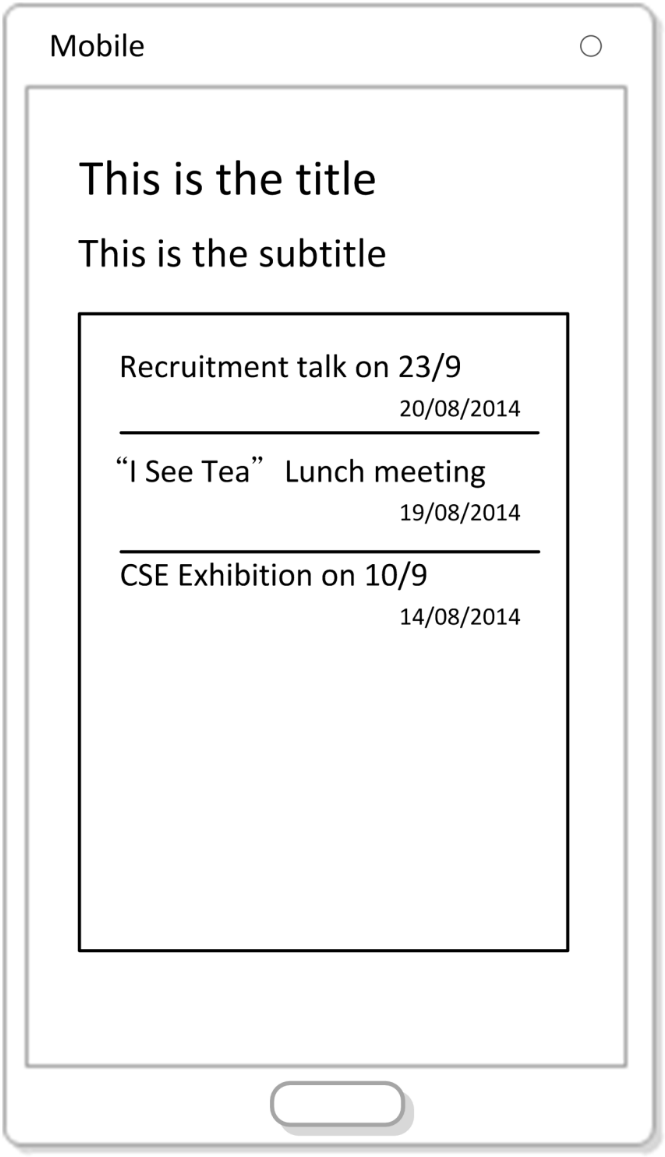
50% OFF Discount

Coupon code: dfh93hf

Valid Date: 2014-05-01

Shops and restaurants can use a layout to make electronics coupon for advertising.

News



Mobile ○

This is the title

This is the subtitle

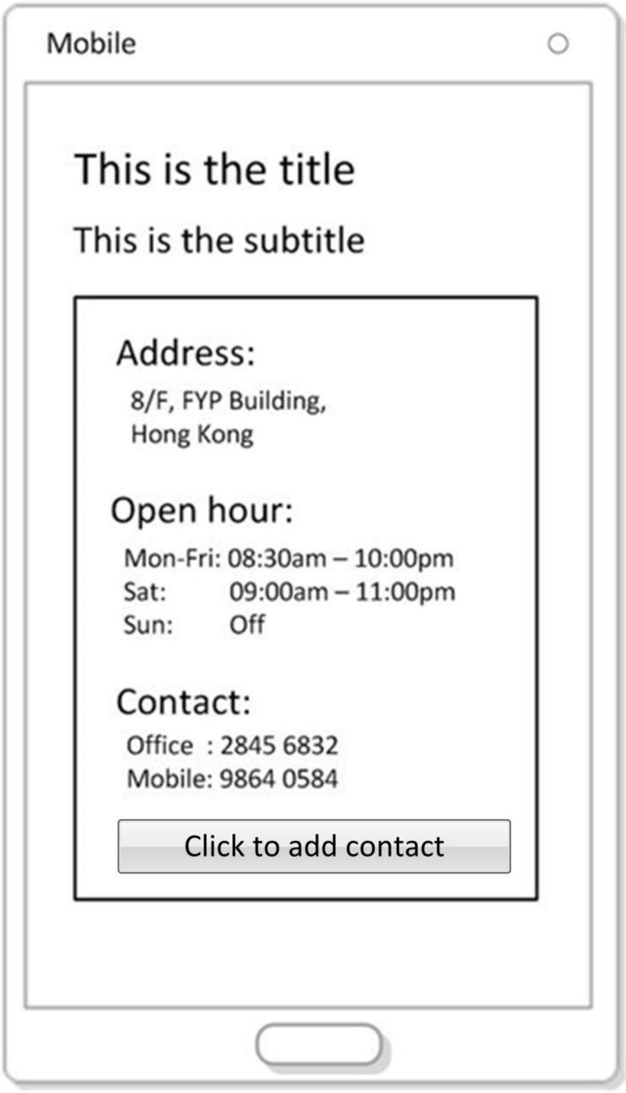
Recruitment talk on 23/9
20/08/2014

“I See Tea” Lunch meeting
19/08/2014

CSE Exhibition on 10/9
14/08/2014

This layout shows all the latest events and will be updated frequently.

General Information



This is the title

This is the subtitle

Address:

8/F, FYP Building,
Hong Kong

Open hour:

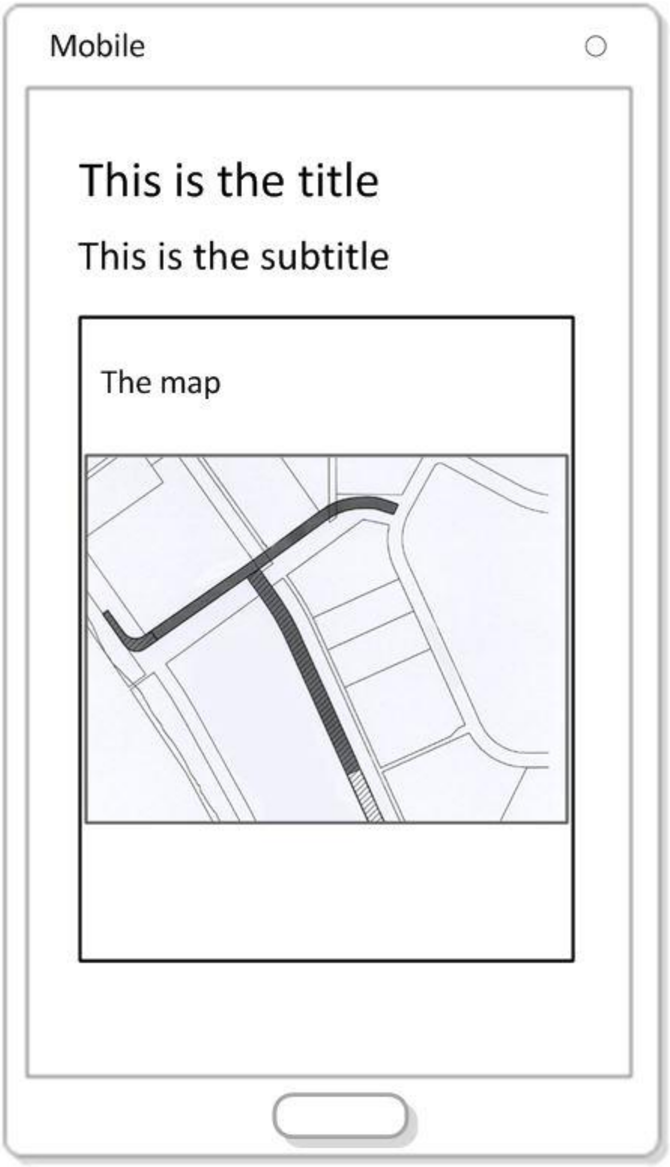
Mon-Fri: 08:30am – 10:00pm
Sat: 09:00am – 11:00pm
Sun: Off

Contact:

Office : 2845 6832
Mobile: 9864 0584

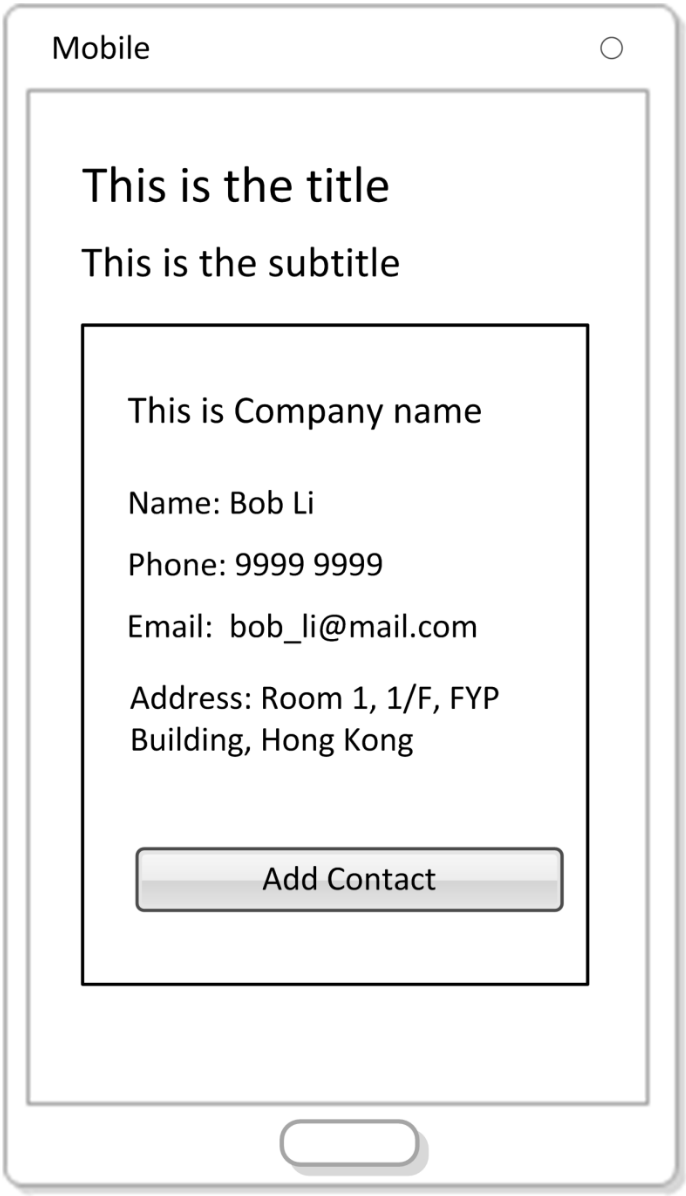
This layout provides general information such as the opening hour of public place.

Map



An image provides route or location information.

Add contact



Mobile

This is the title

This is the subtitle

This is Company name

Name: Bob Li

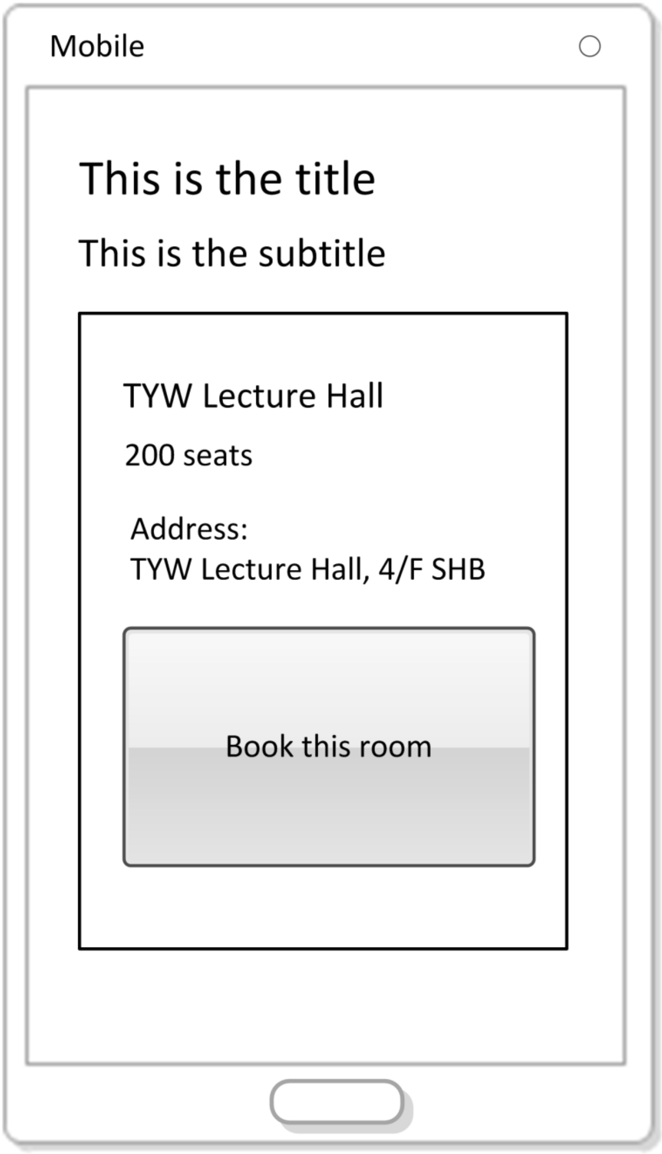
Phone: 9999 9999

Email: bob_li@mail.com

Address: Room 1, 1/F, FYP
Building, Hong Kong

This layout acts as a business card and provides contact information.

Room booking



This is the title

This is the subtitle

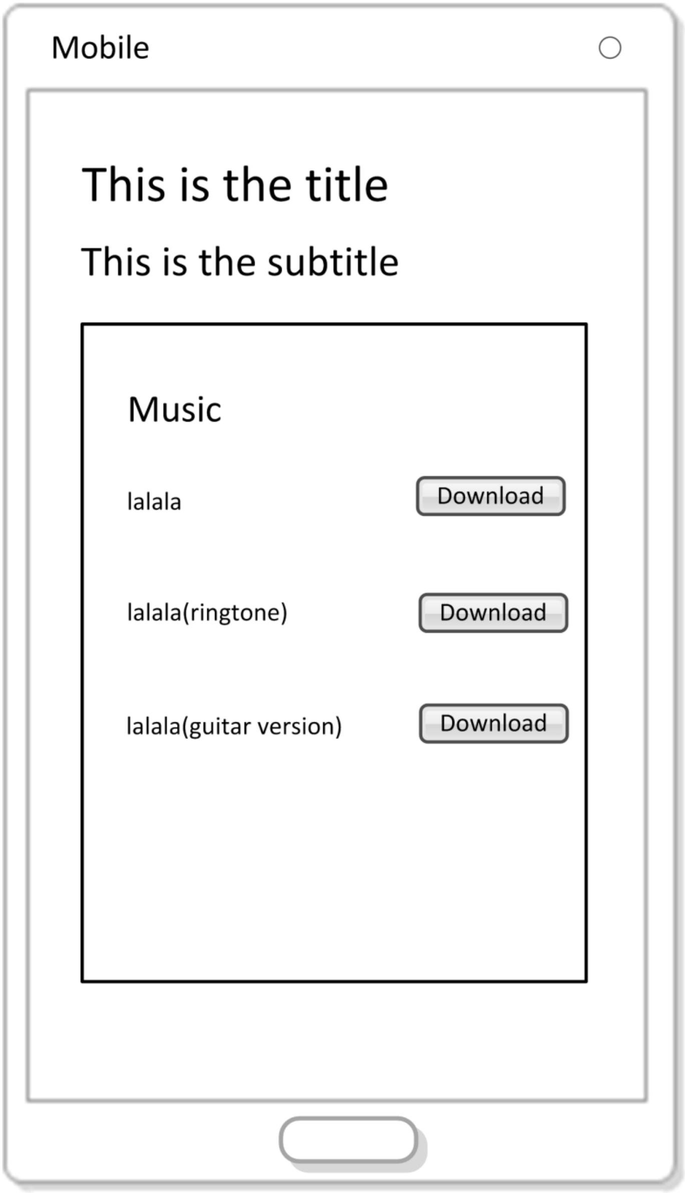
TYW Lecture Hall

200 seats

Address:
TYW Lecture Hall, 4/F SHB

User can book a room through this layout. It can also be changed to a job application form redirection.

File download



Mobile ○

This is the title

This is the subtitle

Music

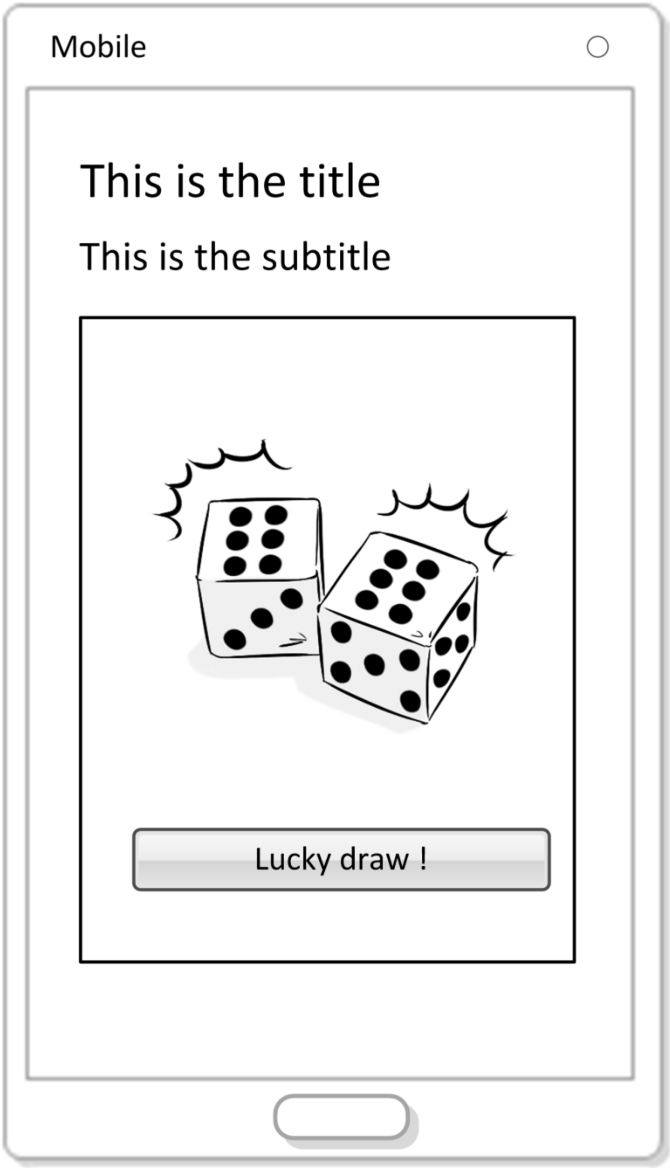
lalala [Download]

lalala(ringtone) [Download]

lalala(guitar version) [Download]

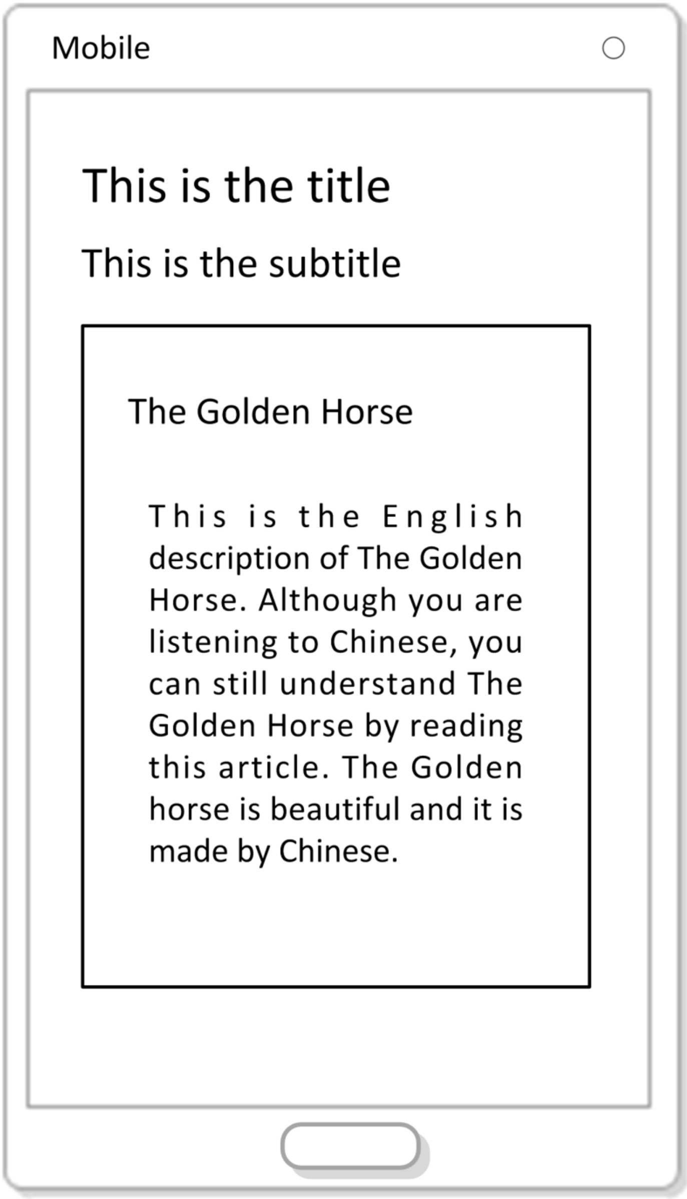
Distributing files become easy when there are many files. Users are free to choose download which file.

Lucky draw



Shops can create a lucky draw event to attract attention.

Translator



This is the title

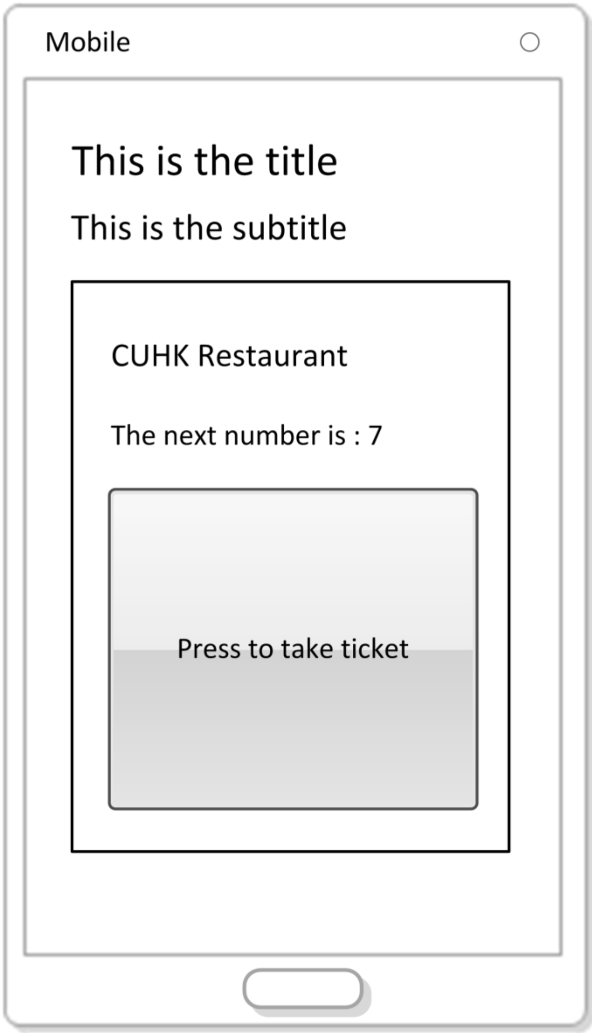
This is the subtitle

The Golden Horse

This is the English description of The Golden Horse. Although you are listening to Chinese, you can still understand The Golden Horse by reading this article. The Golden horse is beautiful and it is made by Chinese.

Sometimes in the museum, when visitors are looking at exhibits, they may not understand the language of description. But it is impossible to put all the languages on the board. Hence this layout provides an interface for museum to show several language on the same exhibit.

Queuing



This is the title

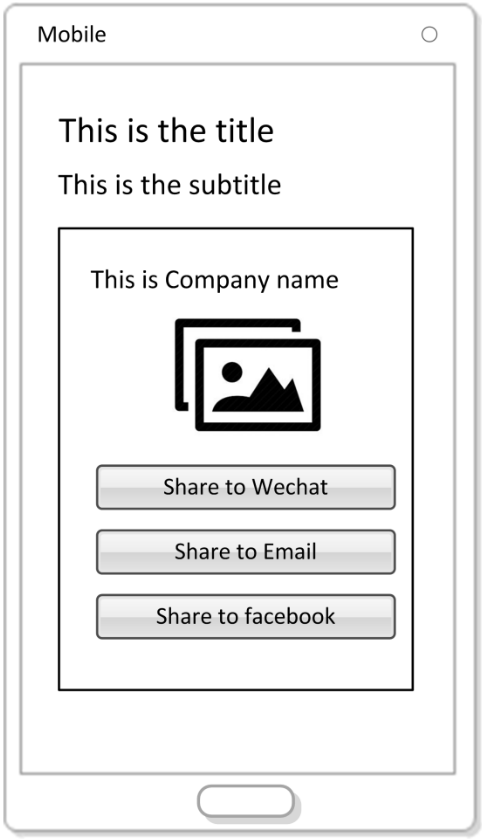
This is the subtitle

CUHK Restaurant

The next number is : 7

This layout allows user to take ticket from the restaurant and wait without physically queue up.

Sharing



This layout helps user to share with friends easily.

6.4.3 NFC Tag Applications

2. Product Promotion	**I. Information** **II. News** **III. Photos / Image** **IV. Special Offer / Coupons** **V. Sharing**
3. Event Promotion	I. Information II. News III. Photos / Image IV. Special Offer / Coupons / Lucky Draw V. Application / Booking VI. Map VII. Sharing
4. Shop / Place Promotion	I. Special Offers / Coupons / Lucky Draw II. News III. Information IV. Photos / Images V. Map VI. Booking VII. Add Contact VIII. Sharing
5. Restaurant	I. Information II. Special Offers / Coupons III. Booking / Queuing IV. Menu

	V. Voting VI. Add Contact VII. Sharing
6. Information Center	I. Information II. Map III. Add Contacts IV. Photos / Images V. Downloading VI. Sharing
7. Transportation Schedule	I. Information II. Schedule / Event Management III. Map

6.5 NFC Tag

Latia NFC tag represents the location where it sticks on. In the user perspective, it contains all the details and interactions of the location. However in practice, it is impossible to store these large data (especially photo) into a single NFC tag. Not only because the capacity of NFC tag, but also the transferring speed is limited to 424 Kbps. To achieve this, we have saved all the fixed data into the mobile app such as photo, name and description. In the NFC Tag, an ID is needed to store.

7 Latia Implementation

7.1 Android application

In the Latia application, we have chosen to use Java with Eclipse (provided by official Android SDK) in Windows. Since we are developing NFC application which is relative new in the Android market, using official Android SDK can get better support. We also limits the version of Android, from version 4.0 (Honeycomb) to version 4.2 (Jelly Bean), in order to support NFC foreground dispatch system.

7.2 Server

We have considered two database management systems: SQLite and MySQL. The following are the advantages of MySQL and SQLite.

MySQL	SQLite
Perfect for concurrency transactions on the data	Easy to set up
Well suited for multi-user environment	Great for databases need on a temporary basis or for test purpose
Perfect in a Client Server Architecture setup	Suitable for using in embedded applications
Suited for managing users and permissions	Suitable for using on small to medium website

According to these advantages, we have chosen MySQL as our database management system. The reason to choose MySQL is that our project is a multi-user environment, users can concurrent submit their comments to server using MySQL. It is perfect in Client and Server architecture setup. Moreover, it is suitable to manage users and their permissions. We can take benefit on it for further development to add user account.

The version of MySQL we are using is 5.0.51a-24+lenny5 - (Debian). Also, we use the PHP as the interface.

7.2.1 MySQL

We have created five tables, tag, layout, resource, version and admin.

The tag table

```
CREATE TABLE Tag

(

  tid            INT(11),

  title          TINYTEXT,

  subtitle       TINYTEXT,

  like           INT(11),

  dislike        INT(11),

  PRIMARY KEY(tid ASC)

);
```

This SQL command creates the tag table with tid(Tag ID), title, subtitle, like, and dislike. The tid is set as the primary key. The data are stored in ascending order of tid.

The layout table

```
CREATE TABLE layout

(

  tid             INT(11),

  lid             INT(11),

  xml             TEXT,

  PRIMARY KEY(tid ASC,lid ASC)

);
```

This SQL command creates the layout table with tid(Tag ID), lid(Layout ID), and xml. The tid and lid are set as to be the primary key. The data are stored in ascending order of tid and lid.

The resource table

```
CREATE TABLE resource

(

  rname          CHAR(255),

  utime          TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

  data           MEDIUMBLOB,

  PRIMARY KEY(rname)

);
```

This SQL command creates the resource table with rname(resource name), utime(update time), and data. The rname is set to be the primary key. The default value of utime is the current time. Also, it will be updated to be current time automatically when the record is updated.

The version table

```
CREATE TABLE version

(

  vid              INT(11) NOT NULL AUTO_INCREMENT,

  rtime            DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

  PRIMARY KEY(vid)

);
```

This SQL command creates the version table with vid (version number) and rtime (release time). The vid is set to be the primary key. The default value of rtime is the current time. Also, it will be updated to be current time automatically when the record is updated.

The admin

```
CREATE TABLE admin

(

  uname          VARCHAR(10),

  pwd            TINYTEXT,

  PRIMARY KEY(uname)

)
```

This SQL command creates the admin table with uname (username) and pwd (encrypted password). The uname is set to be the primary key.

7.2.2 PHP

The format of parameter passing between Android app and PHP is using JSON. The reason we are using JSON but not XML is that Java and PHP have libraries on JSON. With this support, it is easy for us to convert raw data to JSON and perform the data transmit.

Our PHP involves six modules: database_connect (), tag_information(), layout_information(), resource_list(), resource_data(), and vote(). All of them need to get some parameters from the app. To prevent injection attack, we have use the PHP function bind_param to bind the administrator input to the sql query for preventing this attack.

database_connect ()

It is a class used to connect with the database. The connect() is the function to connect the database using the login parameters. The getDatabase() will return the database object.

```
<?
        class DATABASE_CONNECT {
                private $db;
                // constructor
                function __construct() {
                        // connecting to database
                        $this->connect();
                }
                // destructor
                function __destruct() {
                        // closing db connection
                        $this->close();
                }
                /**
                 * Function to connect with database
                 */
                function connect() {
                        // Connecting to mysql database
                        $this->db = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD,
                         DB_DATABASE)
                        or die(mysqli_error());
                        // returing connection cursor
                        return $this->db;
                }
?>
```

tag_information()

The tag_information() gives the overview of the tag information or tag information for a particular Tag ID to app.

First, it uses database_connect() to connect to the database. Then, it will retrieve the Tag ID by using POST method.

If the app passes the Tag ID, it will return the tag information to the app. It will query the layout table in database to count the number of layout of that tag. Also, it will retrieve the title and subtitle in tag table for that tag.

If the app does not pass the Tag ID, the module will give the overview information to the app. It will query the tag table in database to count the number of tag. Then, it will query the layout table to count the total number of layout. Also, it will retrieve the most current version number in the version table.

After these two flow, it will store the value to $result and echo back to the app using JSON format. If not found, it will echo NULL to the app.

```
<?php
        // Connect to database
        $db = new DATABASE_CONNECT();
         $db = $db->getDatabase();
        // Get tag ID
        $tag_id = $_GET['tid'];
        // If the apps pass the tag id, give the tag information
        if(isset($tag_id))
        {
                //Count the numbers of layout of that tag id
                $sql = $db->prepare("SELECT COUNT(DISTINCT lid) AS num_of_layout FROM lay
                out_table WHERE tid =?");
                $sql->bind_param('i', $tag_id);
                $sql-> bind_ result($query);
                if(($obj = $query->fetch_object())!=NULL)
                        $result['num_of_layouts'] = $obj->num_of_layout;
                else
                        $result['num_of_layouts'] = NULL;
```

```
    //Get the title and subtitle of that tag id
    $sql = $db->prepare("SELECT title, subtitle FROM tag_table WHERE tid =?");
    $sql->bind_param('i', $tag_id);
    $sql-> bind_ result($query);

    if(($obj = $query->fetch_object())!=NULL)
    {
        $result['title'] = $obj->title;
        $result['subtitle'] = $obj->subtitle;
    }
    // No such tag
    else
        $result = NULL;
}
//If not set tag id, give the overview
else
{
    //Count the number of tags
    $sql  = "SELECT COUNT(DISTINCT tid) AS num_of_tags FROM tag_table";

    $query = $db->query($sql);
    if(($obj = $query->fetch_object())!=NULL)
        $result['num_of_tags'] = $obj->num_of_tags;
    else
        $result['num_of_tags'] = NULL;
    //Count the total number of layouts
    $sql  = "SELECT COUNT(*) AS num_of_layouts FROM layout_table";
      $query = $db->query($sql);
      if(($obj = $query->fetch_object())!=NULL)
         $result['num_of_layouts'] = $obj->num_of_layouts;
    else
```

```
        //Get the most current version number
        $sql = "SELECT MAX(vid) AS version FROM version_table";
        $query = $db->query($sql);
        if(($obj = $query->fetch_object())!=NULL)
            $result['version'] = $obj->vid;
        else
            $result['version'] = 0;
    }
    echo json_encode($result);
?>
```

layout_information()

The layout_information() gives the layout information of a particular Tag ID and Layout ID to the app.

First, it uses database_connect() to connect to the database. Then, it will retrieve the Tag ID and Layout ID by using POST method. After that, it will query the database to retrieve the layout xml code of that Tag ID and Layout ID in the layout table and echo to the app. If it is not found, it will echo No XML error.

```
<?php
        // Connect to database
         $db = new DATABASE_CONNECT();
        $db = $db->getDatabase();

        // Get tag_id and layout_id
        $tag_id = $_GET['tid'];
        $layout_id = $_GET['lid'];
        if(isset($tag_id) && isset($layout_id))
        {
                //Get the layout xml code
                $sql = $db->prepare("SELECT xml FROM layout_table  WHERE tid =?
                AND lid =? ");
                $sql->bind_param('ii', $tag_id, $layout_id);
                $sql-> bind_ result($query);
                if(($obj = $query->fetch_object())!=NULL)
                        echo $obj->xml;
                //If not found
                else
                        echo "No XML error";
                exit;
        }
        echo "null";
?>
```

resource_list()

The resource_list() gives the name list of resource that needs to be updated given the app version number.

First, it uses database_connect() to connect to the database. Then, it will retrieve the current version number by using POST method.

To check if a resource is outdated, we need to know the app last version time and the most current version time. So, it will query the database to get the both app version and most current version time in version table. After that, it will query the database to retrieve the name list of resource that in between these two version times. If there are outdated resources, it will echo the name list using JSON format to the app and echo empty list vice versa.

```
<?php

        // Connect to database
         $db = new DATABASE_CONNECT();

         $db = $db->getDatabase();

        // Get tag_id and layout_id

        $current_version = $_GET['current_version'];

        if(isset($current_version))

        {

                // Get the current version time

                $sql = $db->prepare("SELECT rtime FROM version_table WHERE version =?");

                $sql->bind_param('s', $current_version);

                $sql-> bind_ result($query);

                if(($obj = $query->fetch_object())!=NULL)

                $current_version_time = $obj->rtime;
                // Get the newest version time

                $sql = "SELECT MAX(rtime) as new_version_time FROM version_table";

                $query = $db->query($sql);

                if(($obj = $query->fetch_object())!=NULL)

                    $new_version_time = $obj->new_version_time;

                // Find the res that need to update
                if(isset($current_version_time) && isset($new_version_time))
                {
                    $sql = $db->prepare("SELECT rname FROM resource_table WHERE
                    utime > ? AND utime <= ?");
                    $sql->bind_param('ss', $current_version_time , $new_version_time);

                    $sql-> bind_ result($query);

                    $res_name = Array();
                    while(($obj = $query->fetch_object())!=NULL)
                        $res_name[] =  $obj->name;
                    echo json_encode($res_name);
                    exit;
                }
        }
        echo "[]";
?>
```

resource_data()

The resource_data() gives the resource data given a particular resource name.

First, it uses database_connect() to connect to the database. Then, it will retrieve the resource name by using POST method. After that, it will query the resource table to get the resource data using the resource name. Finally, it will echo the resource data to the app. If the resource name is not found, it will echo null to the app.

```
<?php
        // Connect to database
         $db = new DATABASE_CONNECT();
        $db = $db->getDatabase();
        $filename = $_GET['resource_name'];
        if(isset($filename))
        {
                //Get the resource data
                $sql = $db->prepare("SELECT data FROM resource_table WHERE rname =?");
                $sql->bind_param('s', $filename);
                $sql-> bind_ result($query);
                if(($obj = $query->fetch_object())!=NULL)
                {
                        echo $obj->data;
                        exit;
                }
        }
        echo 'null';
?>
```

vote()

The vote() either gives the likes/dislike statistic to the app or receive request on increment the likes/dislikes statistic.

First, it uses database_connect() to connect to the database. Then, it will retrieve the Tag ID and message by using POST method.

If the message is set to be ‘likes’ or ‘dislikes’, it means to increment the likes or dislikes of that Tag ID. So, the module will query the database to retrieve the current likes or dislikes statistic. After that, it will update the statistic by adding one.

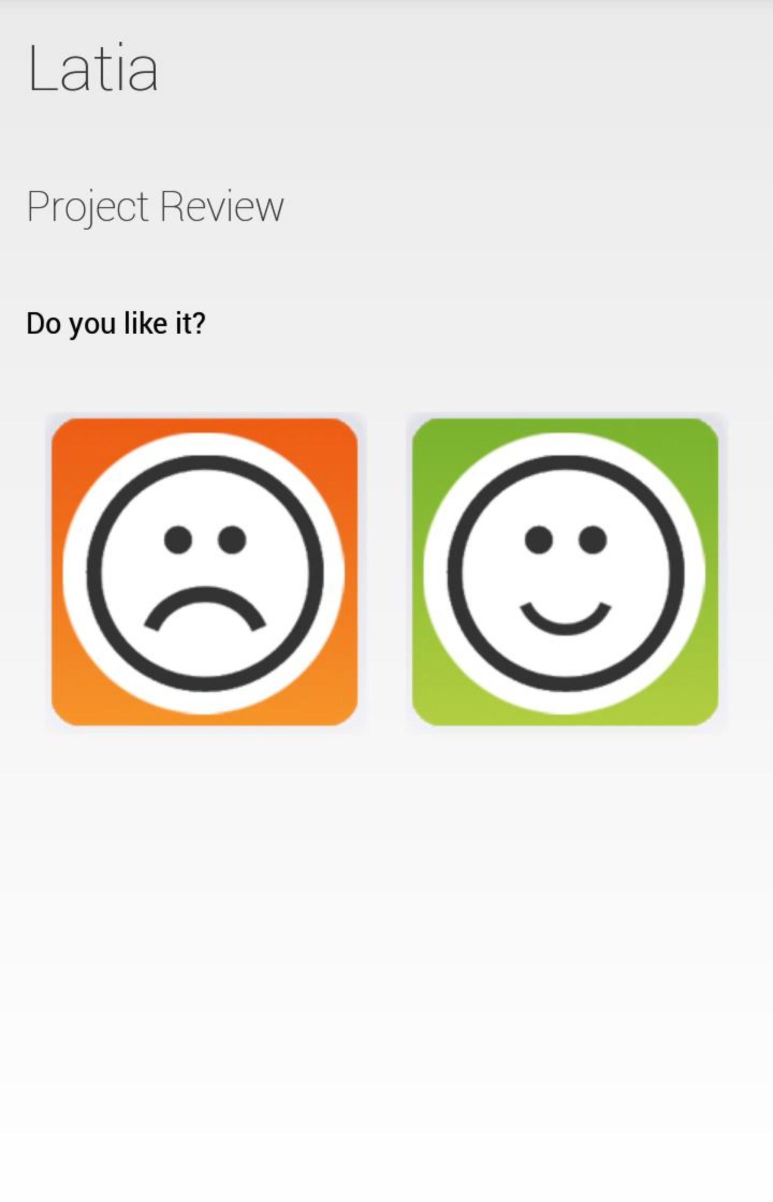
If the message is set to be ‘likes_count’ or ‘dislikes_count’, it means to get the likes or dislikes statistic of that Tag ID. So, the module will query the database to retrieve the likes or dislikes statistic in the tag_table.

```
<?php
        // Connect to database
        $db = new DATABASE_CONNECT();
        $db = $db->getDatabase();

        $tag_id = $_GET['tid'];
        $message = $_GET['msg'];
        if(isset($tag_id) && isset($message))
        {
                // Increment likes statistic
                if(strcmp($message, "likes")==0)
                {
                        $sql = $db->prepare("SELECT like FROM tag_table WHERE tid = ?");
                        $sql->bind_param('s', $tag_id);
                        $sql->execute();
                        $sql->bind_result($like);
                        $sql->fetch();

                        // increment
                                $sql = $db->prepare("UPDATE tag_table SET likes= ?
                                 WHERE tid = ?");
                                $sql->bind_param('is', $like, $tag_id);
                                $sql->execute();
                                mysqli_query($db, $sql) or die(mysqli_error($db));
                }
```

7.3 Android app

7.3.1 Voting

This is a layout for the voting on our project. There is a vertical linear layout to contain the content. Beside the title and subtitle, it includes a text view for displaying the question about voting and also a horizontal linear layout that contains two image buttons like and dislike. The onclick functions are set to be like and dislike.



Latia

Project Review

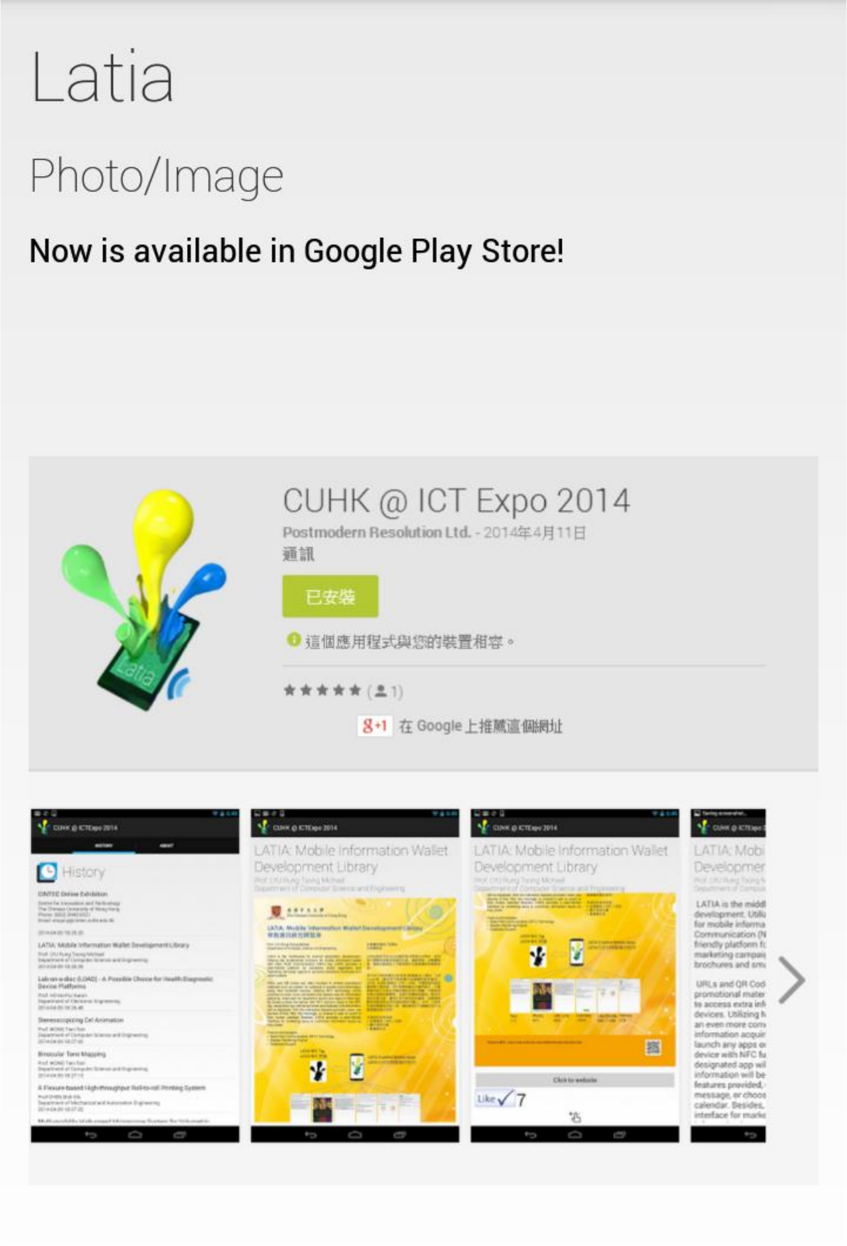
Do you like it?

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

           <TextView
                    width="wrap_content"
               height="wrap_content"
               textSize="40"
               text="Latia" />

           <TextView
               width="wrap_content"
               height="wrap_content"
               textSize="25"
               text="Project Review" />

           <TextView
               width="wrap_content"
               height="wrap_content"
               text="Do you like it?" />

           <LinearLayout
                   width="match_parent"
                    height="wrap_content"
                    orientation="horizontal" >
```

```
            <ImageButton
                  width="0"
                  height="wrap_content"
                  weight="1"
                  onClick="like"
                  backgroundColor="0"
                  imageUp="LikeButton.jpg"
                  imageDown="LikeButton.jpg"/>

              <ImageButton
                  width="0"
                  height="wrap_content"
                  weight="1"
                  onClick="dislike"
                  backgroundColor="0"
                  imageUp="DislikeButton.jpg"
                  imageDown="DislikeButton.jpg"/>

        </LinearLayout>

</LinearLayout>
```

7.3.2 Photos/Image

This is a photo/image for our project. There is a vertical linear layout to contain the title, subtitle and content. The content includes a text view for the photo description and an image view.



Latia

Photo/Image

Now is available in Google Play Store!

CUHK @ ICT Expo 2014

Postmodern Resolution Ltd. - 2014年4月11日

通訊

這個應用程式與您的裝置相容。

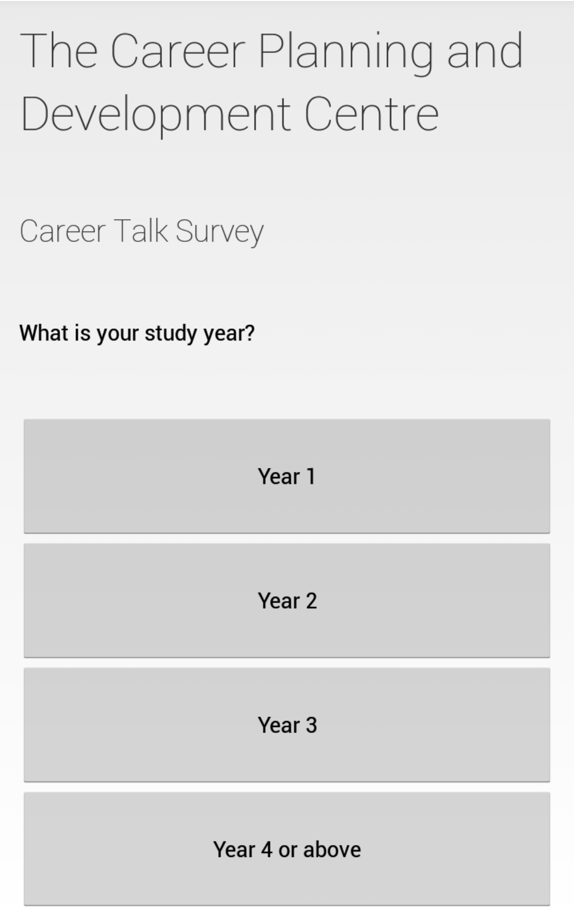
★★★★★ (1)

在 Google 上推薦這個網址

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                 width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="Latia" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Photo/Image" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              text="Now is available in Google Play Store!" />

            <ImageView
                  width="fill_parent"
                  height="wrap_content"
                  src="play_store.png"/>

</LinearLayout>
```

7.3.3 Questionnaire

This is a layout for the career talk questionnaire of Career Planning and Development Centre. There is a vertical linear layout to contain the title, subtitle and content. It includes a text view for the question and numbers of button for the choices. The onclick function will be set to url that handles the statistic collection.



The Career Planning and Development Centre

Career Talk Survey

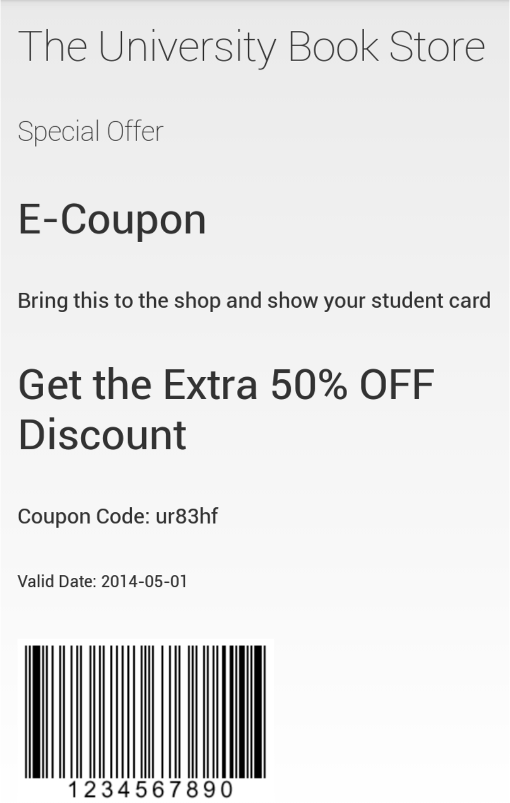
What is your study year?

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                  width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The Career Planning and Development Centre" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Career Talk Survey" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              text="What is your study year?" />

            <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Year 1"
                    onClick="url"
                    url="http://www.career_planning.com/survey.php"/>
            <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Year 2"
                    onClick="url"
                    url="http://www.career_planning.com/survey.php"/>
```

```
        <Button
                width="match_parent"
                height="wrap_content"
                text="Year 3"
                onClick="url"
                url="http://www.career_planning.com/survey.php"/>
        <Button
                width="match_parent"
                height="wrap_content"
                text="Year 4 or above"
                onClick="url"
                url="http://www.career_planning.com/survey.php"/>

</LinearLayout>
```

7.3.4 Special Offer/ Electronic Coupons

This is a layout for e-coupon in University Book store. There is a vertical linear layout to contain the title, subtitle and content. It includes some text views for the usage E-coupons, coupon code and the valid date. At the bottom, there is an image view to show a barcode for reference.



The University Book Store

Special Offer

E-Coupon

Bring this to the shop and show your student card

Get the Extra 50% OFF Discount

Coupon Code: ur83hf

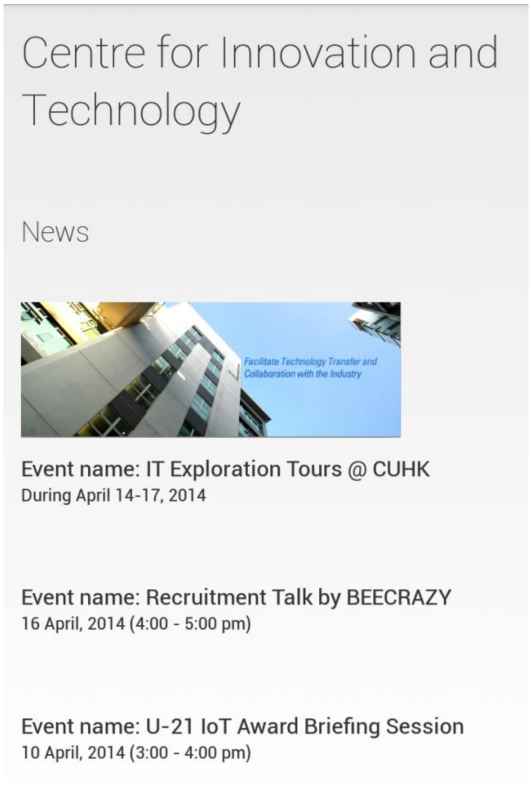
Valid Date: 2014-05-01

1234567890

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">
          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The University Book Store" />
          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Special Offer" />
          <TextView
              width="wrap_content"
              height="wrap_content"
              text="Bring this to the shop and show your student card" />
            <TextView
              width="wrap_content"
              height="wrap_content"
                   textSize="40"
              text="Get the Extra 50% OFF Discount" />
            <TextView
              width="wrap_content"
              height="wrap_content"
                   textSize="20"
              text="Coupon Code: ur83hf" />
            <TextView
              width="wrap_content"
              height="wrap_content"
                   textSize="15"
              text="Valid Date: 2014-05-01" />
            <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="coupon.png"/>
```

7.3.5 News

This is the layout for the news in the Centre of Innovation and Technology. It is a layout is a vertical linear layout that contains the image view and some text views. The image views display the banner of the department. The text views announce the news and upcoming events.



Centre for Innovation and Technology

News

Facilitate Technology Transfer and Collaboration with the Industry

Event name: IT Exploration Tours @ CUHK
During April 14-17, 2014

Event name: Recruitment Talk by BEECRAZY
16 April, 2014 (4:00 - 5:00 pm)

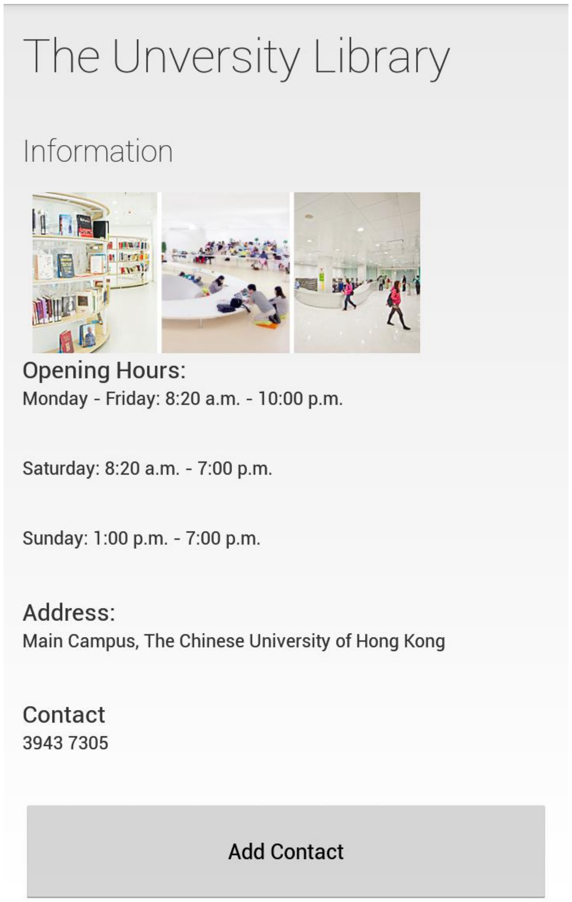
Event name: U-21 IoT Award Briefing Session
10 April, 2014 (3:00 - 4:00 pm)

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="Centre for Innovation and Technology" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="News" />

            <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="news_banner.png"/>

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Event name: IT Exploration Tours @ CUHK" />
```

```
         <TextView
          width="wrap_content"
          height="wrap_content"
              textSize="20"
          text="During April 14-17, 2014" />

         <TextView
          width="wrap_content"
          height="wrap_content"
          text="Event name: Recruitment Talk by BEECRAZY" />

         <TextView
          width="wrap_content"
          height="wrap_content"
              textSize="20"
          text="16 April, 2014 (4:00 - 5:00 pm)" />

         <TextView
          width="wrap_content"
          height="wrap_content"
          text="Event name: U-21 IoT Award Briefing Session" />

         <TextView
          width="wrap_content"
          height="wrap_content"
              textSize="20"
          text="10 April, 2014 (3:00 - 4:00 pm)" />

</LinearLayout>
```

7.3.6 General Information

This is the layout of general information for the University Library. The vertical layout includes the image view, some text views, and a button. The image view displays the banner of Library. The text views show the opening hours, address and contact. There is a button set to be contact. All the contact details are set in the button element.



The Unversity Library

Information

Opening Hours:
Monday - Friday: 8:20 a.m. - 10:00 p.m.

Saturday: 8:20 a.m. - 7:00 p.m.

Sunday: 1:00 p.m. - 7:00 p.m.

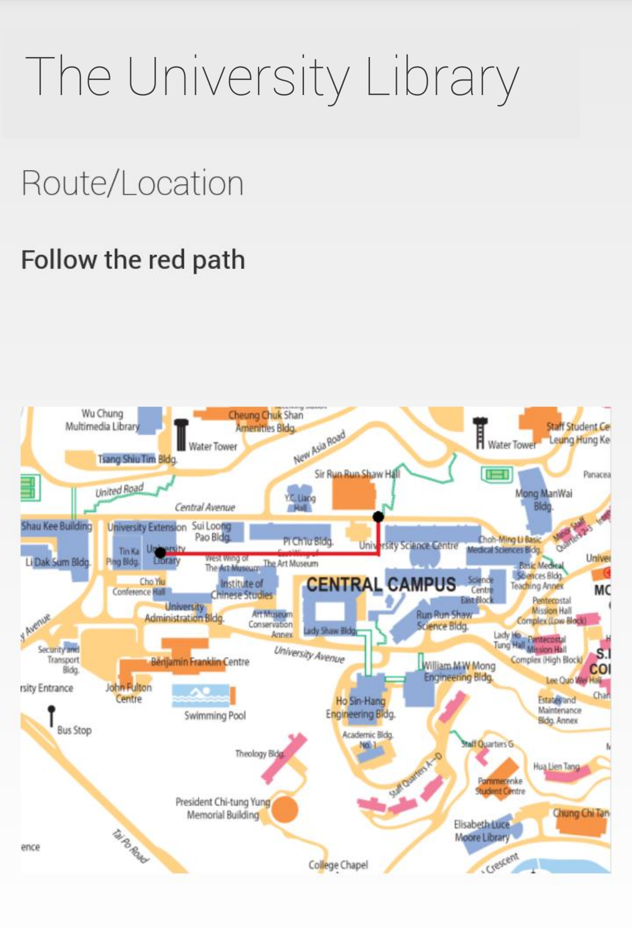
Address:
Main Campus, The Chinese University of Hong Kong

Contact
3943 7305

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The Unversity Library" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Information" />

            <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="library_banner.png"/>

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Opening Hours:" />

            <TextView
              width="wrap_content"
              height="wrap_content"
                   textSize="20"
              text="Monday - Friday: 8:20 a.m. - 10:00 p.m." />

            <TextView
              width="wrap_content"
              height="wrap_content"
                   textSize="20"
              text="Saturday: 8:20 a.m. - 7:00 p.m." />
```

7.3.7 Map

This is a layout for locating the path of the University Library. It is a vertical linear layout that contains the text view for the instruction and an image view to display the map.



The University Library

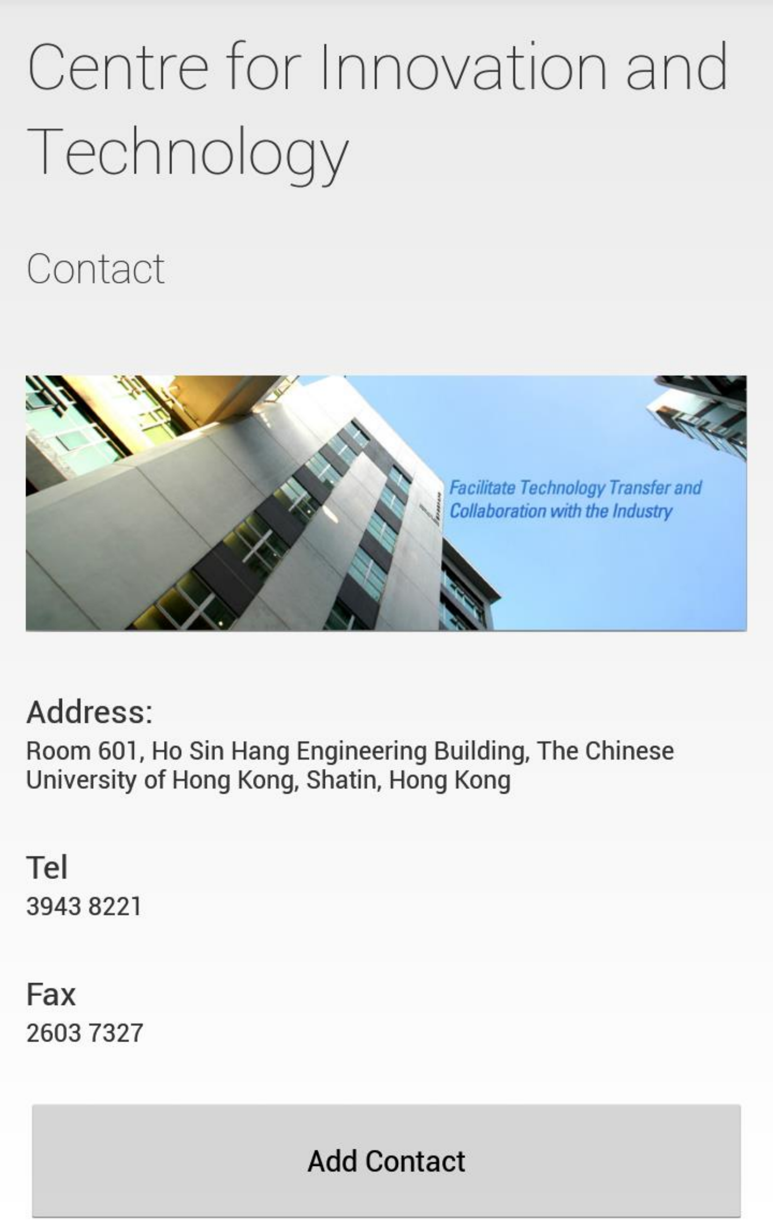
Route/Location

Follow the red path

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The Unversity Library" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Route/Location" />

             <TextView
              width="wrap_content"
              height="wrap_content"
              text="Follow the red path" />

             <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="map.png"/>

</LinearLayout>
```

7.3.8 Add contact

This is a layout for add contact of Centre for Innovation and Technology. It includes an image view at the top to display the banner of the Centre. Below are the text views that display the address, telephone number, fax number, and email. At the bottom, there is the button that set to be contact and have set the contact details.



Centre for Innovation and Technology

Contact

Facilitate Technology Transfer and Collaboration with the Industry

Address:
Room 601, Ho Sin Hang Engineering Building, The Chinese University of Hong Kong, Shatin, Hong Kong

Tel
3943 8221

Fax
2603 7327

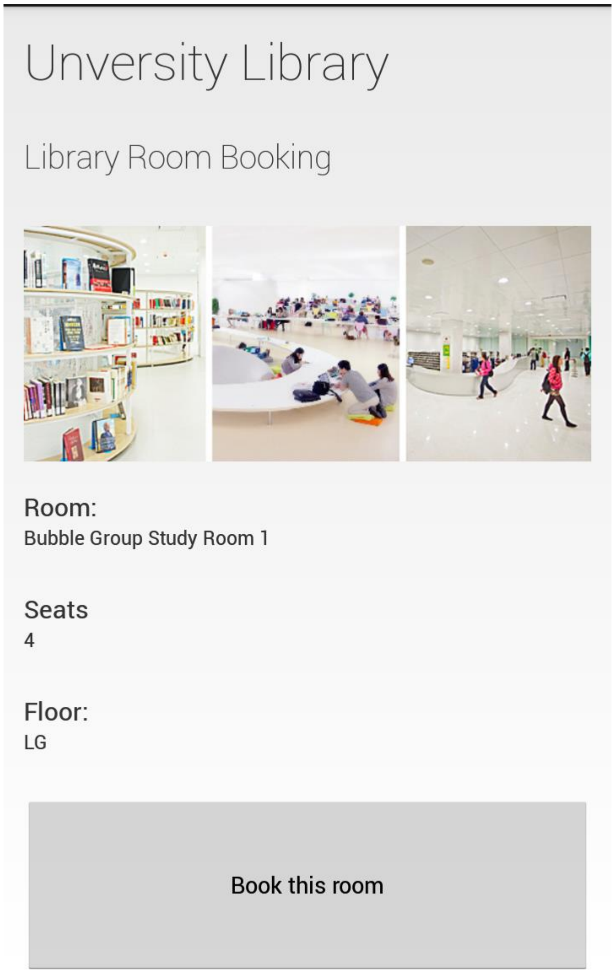
```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="Centre for Innovation and Technology" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Contact" />

            <ImageView
                   width="fill_parent"
                   height="wrap_content"
                   src="cintec_banner.png"/>

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Address:" />

            <TextView
              width="wrap_content"
              height="wrap_content"
             textSize="20"
             text="Room 601, Ho Sin Hang Engineering Building, The Chinese
             University of Hong Kong, Shatin, Hong Kong" />
```

```
<TextView
  width="wrap_content"
  height="wrap_content"
  text="Tel:" />

<TextView
  width="wrap_content"
  height="wrap_content"
        textSize="20"
  text="3943 8221" />

<TextView
  width="wrap_content"
  height="wrap_content"
  text="Fax:" />

<TextView
  width="wrap_content"
  height="wrap_content"
        textSize="20"
  text="2603 7327" />

<TextView
  width="wrap_content"
  height="wrap_content"
  text="Email:" />

<TextView
  width="wrap_content"
  height="wrap_content"
        textSize="20"
  text="enquiry@cintec.cuhk.edu.hk" />
```

```
                <Button
                          width="match_parent"
                          height="wrap_content"
                          text="Add Contact"
                          onClick="contact"
                          contactName="Centre for Innovation and Technology"
                          contactPhone="39438221"
                          contactEmail="enquiry@cintec.cuhk.edu.hk" />

</LinearLayout>
```

7.3.9 Room booking

This is a layout for booking the study room in University library. It is a vertical linear layout that contains the image view, text view and button. The image view displays the library banner. The text view show the room details. The bottom button is set to be URL that handle the room booking step.



Unversity Library

Library Room Booking

Room:
Bubble Group Study Room 1

Seats
4

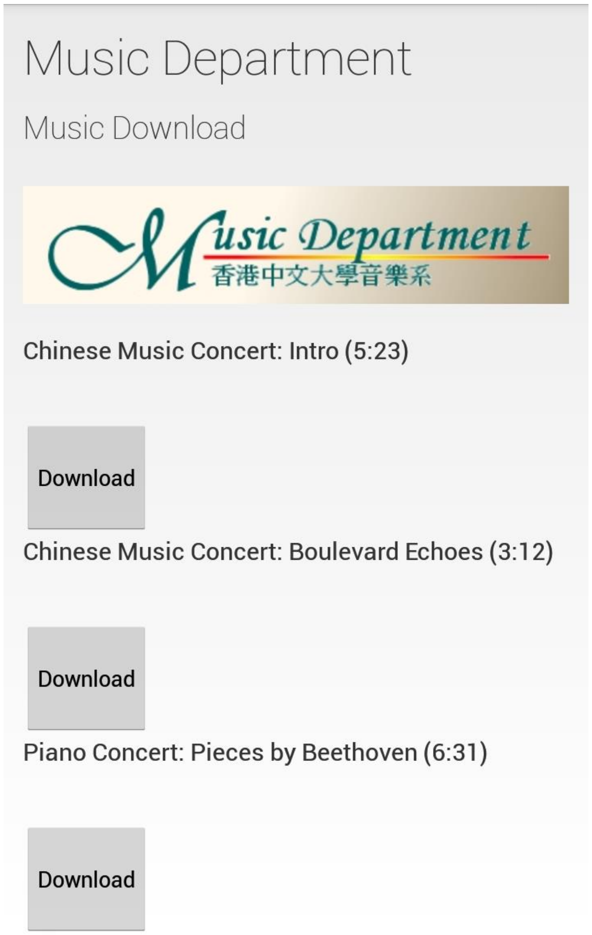
Floor:
LG

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The Unversity Library" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Library Room Booking" />

            <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="library_banner.png"/>

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Room:" />

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Bubble Group Study Room 1" />

            <TextView
              width="wrap_content"
              height="wrap_content"
              text="Seats:" />
```

```
             <TextView
              width="wrap_content"
              height="wrap_content"
              text="4" />
             <TextView
              width="wrap_content"
              height="wrap_content"
              text="Floor:" />
             <TextView
              width="wrap_content"
              height="wrap_content"
              text="LG" />
             <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Book this room"
                    onClick="url"
                    url="https://rbs.lib.cuhk.edu.hk/Booking/Secure/FacilityStatus.p
                    hp" />
</LinearLayout>
```

7.3.10 File download

This is a layout for Music Department for music download. It is a vertical linear layout contains numbers of text views and buttons. The text views show the song details. The button is set to be url that handle the song download.



Music Department

Music Download

Chinese Music Concert: Intro (5:23)

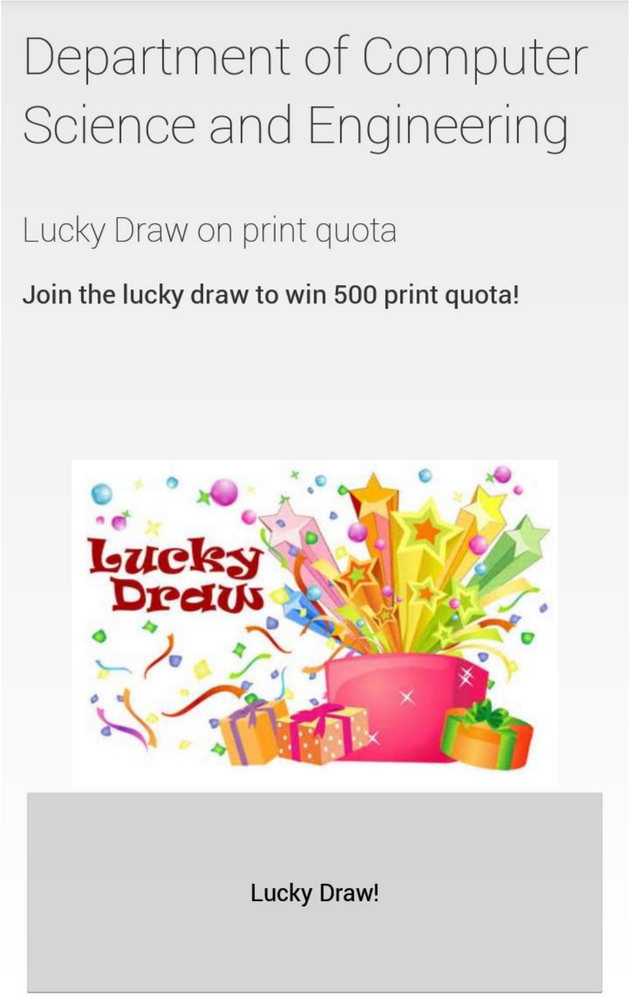
Chinese Music Concert: Boulevard Echoes (3:12)

Piano Concert: Pieces by Beethoven (6:31)

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="Music Department" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Music Download" />

             <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="music_depeartment.png"/>

             <TextView
              width="wrap_content"
              height="wrap_content"
              text="Chinese Music Concert: Intro (5:23)" />
             <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Download"
                    onClick="url"
                    url="http://www.cuhk.edu.hk/mus/m1.mp3" />
             <TextView
              width="wrap_content"
              height="wrap_content"
              text="Chinese Music Concert: Boulevard Echoes (3:12)" />
             <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Download"
                    onClick="url"
                    url="http://www.cuhk.edu.hk/mus/m2.mp3" />
             <TextView
              width="wrap_content"
              height="wrap_content"
              text="Piano Concert: Pieces by Beethoven (6:31)" />
             <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Download"
                    onClick="url"
                    url="http://www.cuhk.edu.hk/mus/m3.mp3" />

</LinearLayout>
```

7.3.11 Lucky draw

This is a lucky draw layout for the print quota in CSE department. It contains the image view to indicate the lucky draw event, the text view that display the instruction and a button which is set to be URL that start the lucky draw.



Department of Computer Science and Engineering

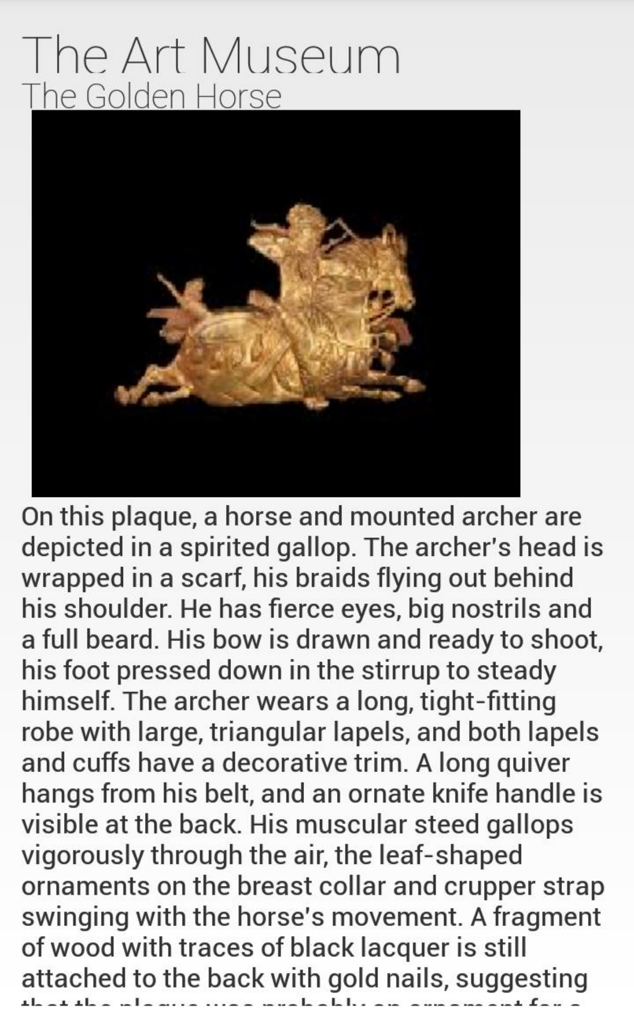
Lucky Draw on print quota

Join the lucky draw to win 500 print quota!

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

           <TextView
                    width="wrap_content"
               height="wrap_content"
               textSize="40"
               text="Department of Computer Science and Engineering" />

           <TextView
               width="wrap_content"
               height="wrap_content"
               textSize="25"
               text="Lucky Draw on print quota" />

             <ImageView
                     width="fill_parent"
                     height="wrap_content"
                     src="lucky_draw.png"/>
             <Button
                     width="match_parent"
                     height="wrap_content"
                     text="Lucky Draw!"
                     onClick="url"
                     url="http://www.cse.cuhk.edu.hk/lucky_draw.php" />


</LinearLayout>
```

7.3.12 Translator

This is a translator layout for the Art Museum. It includes an image view to display the exhibit and also a text view for displaying the translated description.



The Art Museum

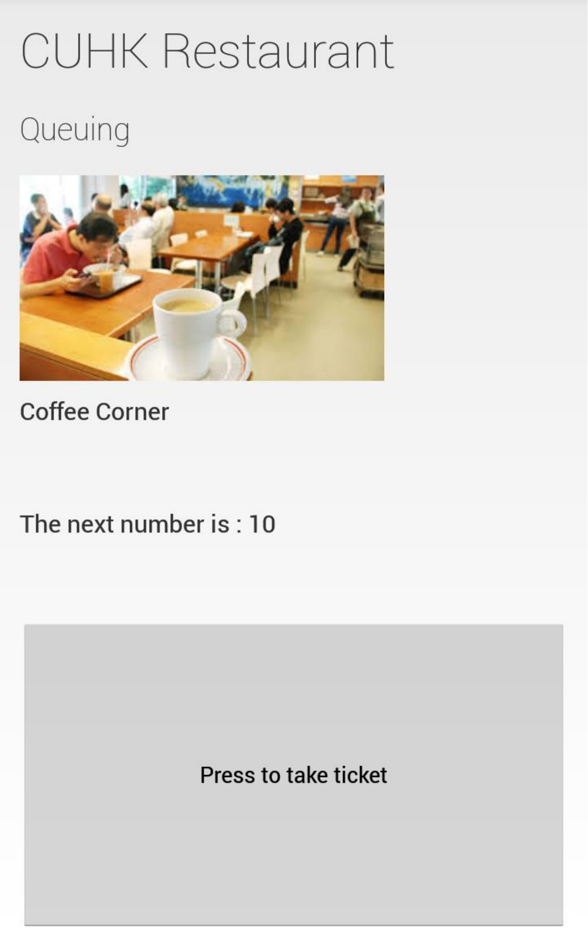
The Golden Horse

On this plaque, a horse and mounted archer are depicted in a spirited gallop. The archer's head is wrapped in a scarf, his braids flying out behind his shoulder. He has fierce eyes, big nostrils and a full beard. His bow is drawn and ready to shoot, his foot pressed down in the stirrup to steady himself. The archer wears a long, tight-fitting robe with large, triangular lapels, and both lapels and cuffs have a decorative trim. A long quiver hangs from his belt, and an ornate knife handle is visible at the back. His muscular steed gallops vigorously through the air, the leaf-shaped ornaments on the breast collar and crupper strap swinging with the horse's movement. A fragment of wood with traces of black lacquer is still attached to the back with gold nails, suggesting that the plaque was probably an ornament for a

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="The Art Museum" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Queuing" />

            <ImageView
                   width="fill_parent"
                   height="wrap_content"
                   src="golden_horse.png"/>
            <TextView
              width="wrap_content"
              height="wrap_content"
             textSize="25"
            text="On this plaque, a horse and mounted archer are depicted in a
            spirited gallop. The archer's head is wrapped in a scarf, his braids
            flying out behind his shoulder. He has fierce eyes, big nostrils and a
            full beard. His bow is drawn and ready to shoot, his foot pressed down
            in the stirrup to steady himself. The archer wears a long, tight-fitting
          robe with large, triangular lapels, and both lapels and cuffs have a
          decorative trim. A long quiver hangs from his belt, and an ornate knife handle
          is visible at the back. His muscular steed gallops vigorously through the air,
          the leaf-shaped ornaments on the breast collar and crupper strap swinging with
          the horse's movement. A fragment of wood with traces of black lacquer is still
          attached to the back with gold nails, suggesting that the plaque was probably
          an ornament for a wooden object." />

</LinearLayout>
```

7.3.13 Queuing

This is a layout for restaurant queuing. It is a vertical layout that contains the image view that displays the restaurant banner and the text view that show the next number of the table. The bottom button is set to be url that handle the restaurant queuing.



CUHK Restaurant

Queuing

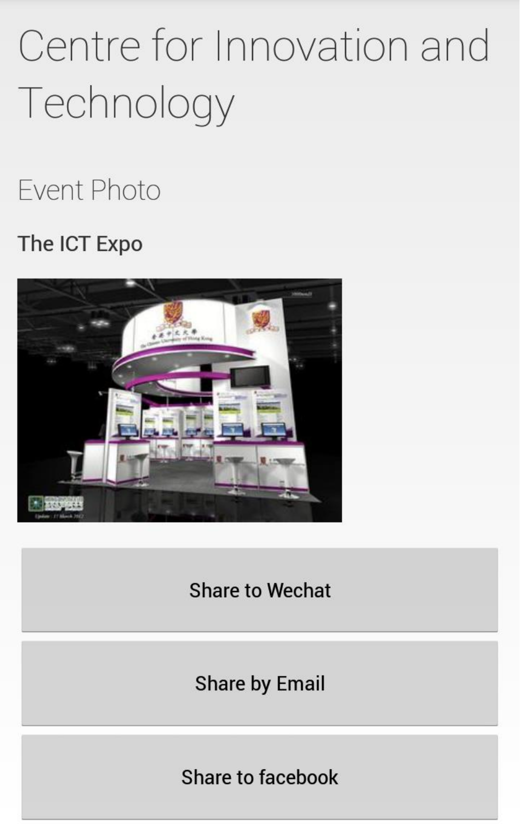
Coffee Corner

The next number is : 10

```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">
          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="CUHK Restaurant" />
          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Queuing" />
            <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="restaurant.png"/>
            <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Coffee Corner" />
            <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="The next number is : 10" />
            <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Press to take ticket"
                    onClick="url"
                    url="http://www.cuhk_restaurant.com/queuing.php" />
</LinearLayout>
```

7.3.14 Sharing

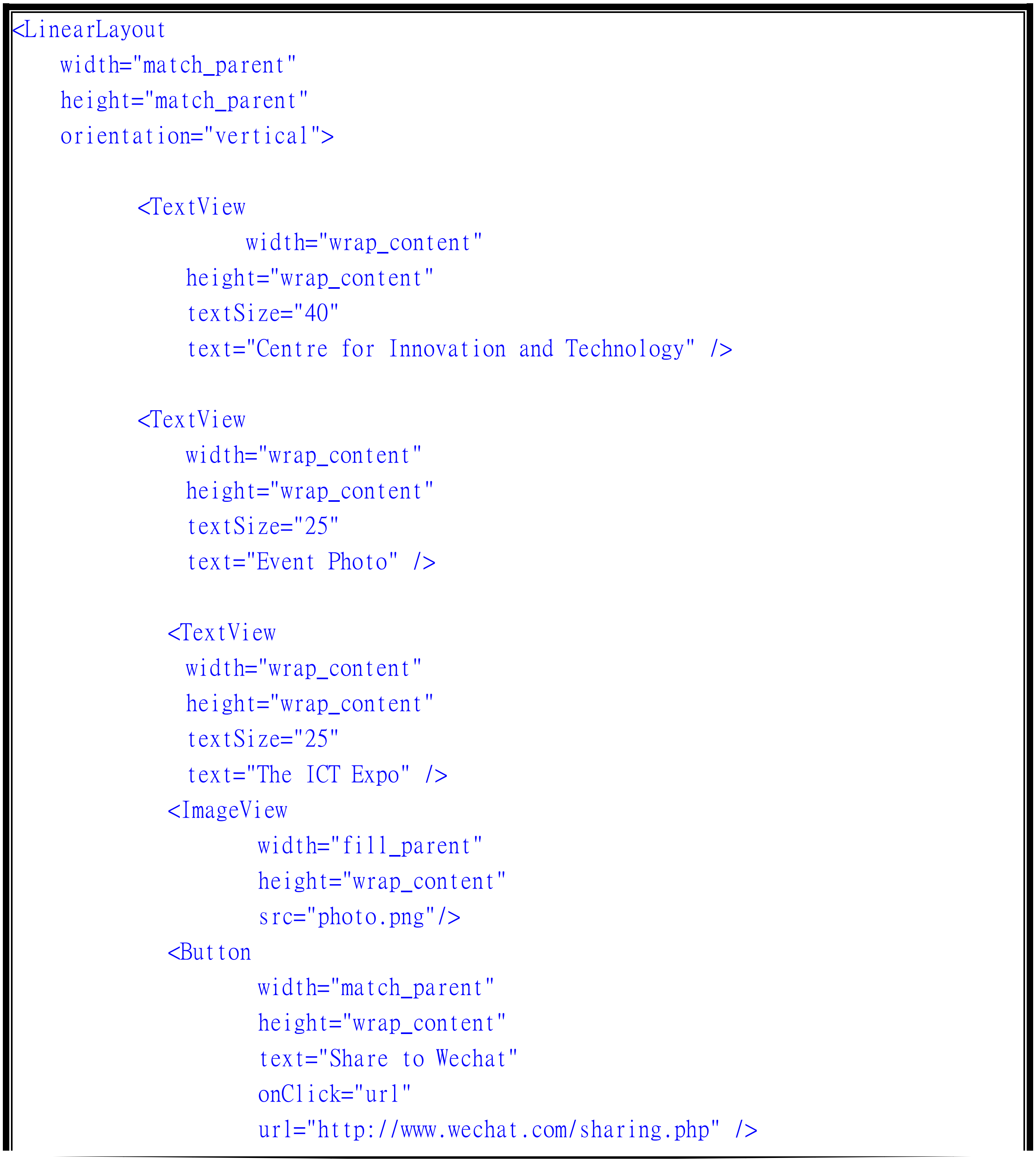
This is a layout for sharing the ICT Expo highlight. It contains the text view for photo description and an image view displaying the photo. Below is the buttons for sharing by Wechat, facebook or email. The buttons are set to be url that handle the sharing.



Centre for Innovation and Technology

Event Photo

The ICT Expo



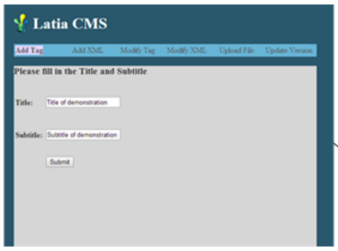
```
<LinearLayout
    width="match_parent"
    height="match_parent"
    orientation="vertical">

          <TextView
                   width="wrap_content"
              height="wrap_content"
              textSize="40"
              text="Centre for Innovation and Technology" />

          <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="Event Photo" />

             <TextView
              width="wrap_content"
              height="wrap_content"
              textSize="25"
              text="The ICT Expo" />
             <ImageView
                    width="fill_parent"
                    height="wrap_content"
                    src="photo.png"/>
             <Button
                    width="match_parent"
                    height="wrap_content"
                    text="Share to Wechat"
                    onClick="url"
                    url="http://www.wechat.com/sharing.php" />
             <Button
                     width="match_parent"
                     height="wrap_content"
                     text="Share by Email"
                     onClick="url"
                     url="http://www.email.com/sharing.php" />
             <Button
                     width="match_parent"
                     height="wrap_content"
                     text="Share to facebook"
                     onClick="url"
                     url="http://www.facebook.com/sharing.php" />

</LinearLayout>
```

7.4 CMS

We need to build a CMS that provide a reliable way for administrator to modify the information in the database. As we want to let the administrators modify it without installing any system, we decided to build the CMS using HTML5 and PHP. So, the administrators can do the modification in everywhere through the website. We use HTML5 and CSS to build the user interface for frontend and uses PHP to connect the database for backend.

7.4.1 HTML5

There will be seven webpages in CMS: Login, add tag, add layout, modify tag, modify layout, upload resource and update version. After the administrators login to CMS, it will redirect to the main page. On the top of the main page, there will be a tab bar to show the link of other page.

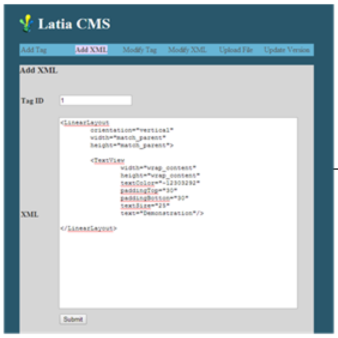


Please fill in the Title and Subtitle

Title: Title of demonstration

Subtitle: Subtitle of demonstration

Add Tag webpage



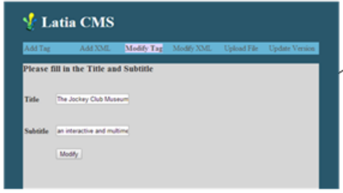
Add XML

Tag ID 1

XML

```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">

        <TextView
                width="wrap_content"
                height="wrap_content"
                textColor="-12303292"
                paddingTop="30"
                paddingBottom="30"
                textSize="25"
                text="Demonstration"/>

</LinearLayout>
```

Add XML webpage

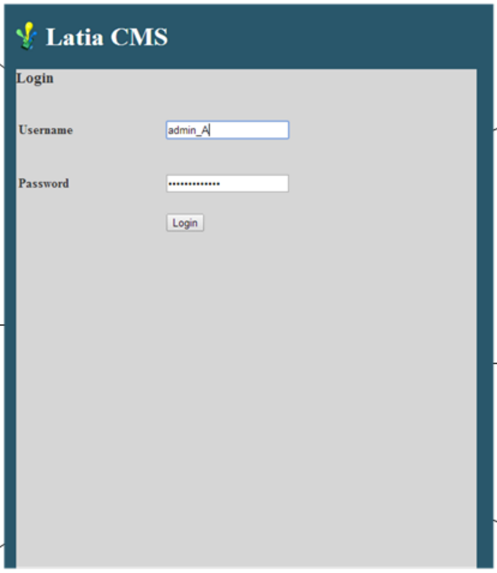


Please fill in the Title and Subtitle

Title The Jockey Club Museum

Subtitle an interactive and multime

Modify Tag webpage

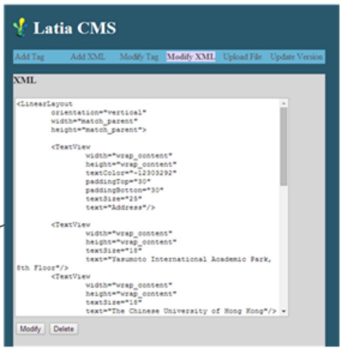


Login

Username admin_A

Password ••••••••••••••

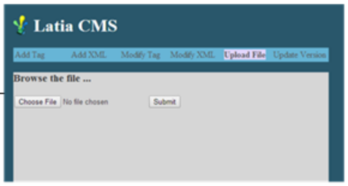
Login webpage



XML

```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">

        <TextView
                width="wrap_content"
                height="wrap_content"
                textColor="-12303292"
                paddingTop="30"
                paddingBottom="30"
                textSize="25"
                text="Address"/>

        <TextView
                width="wrap_content"
                height="wrap_content"
                textSize="18"
                text="Yasumoto International Academic Park,
8th Floor"/>
        <TextView
                width="wrap_content"
                height="wrap_content"
                textSize="18"
                text="The Chinese University of Hong Kong"/>
```

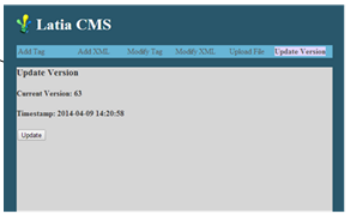
Modify XML webpage



Browse the file ...

No file chosen

Upload File webpage



Update Version

Current Version: 63

Timestamp: 2014-04-09 14:20:58

Update Version

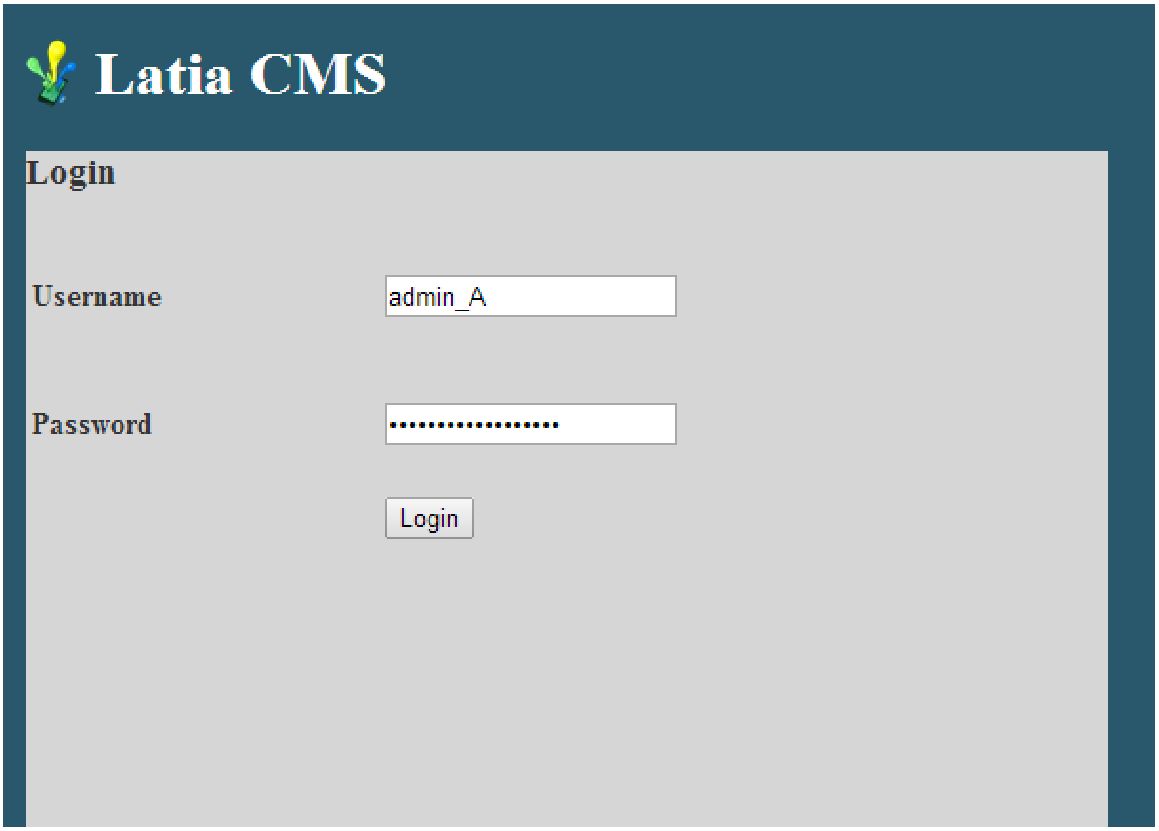
We have added some div to separate the layout of the CMS. Each of the div has its own id. So, we can use CSS later to modify the style of the layout.

Each page will contain a logo div on the top. Under the logo div, we added the tabs div for creating the tabs bar. At the bottom, there is a content div. In the logo div, it contains a logo image and the name of the CMS. In the tabs div, it will include the six webpages hyperlink. In the content div, it leave blink for different module add the contents.

```
<!DOCTYPE html>
<html>
 <head>
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <meta charset=utf-8 />
  <title>Latia CMS</title>
  <script src="js/jquery-1.11.0.min.js"></script>
  <script src="js/db_function.js"></script>
 </head>
 <body>
  <div class="Logo">
   <h1><img src="res/logo.png" width=5% height=11% style="vertical-align:middle;">
   <span style="vertical-align: middle;">Latia CMS</span></h1>
  </div>
  <div class="tabs">
   <table>
    <tr>
     <td width="100"><a href="add_tag_info.php">Add Tag</a></td>
     <td width="100" align="center"><a href="add_xml.php">Add XML</a></td>
     <td width="100" align="center"><b class="current">Modify Tag</b></td>
     <td width="100" align="center"><a href="modify_xml.php">Modify XML</a></td>
     <td width="100" align="center"><a href="upload_file.php">Upload File</a></td>
     <td width="110" align="center"><a href="update_version.php">Update Ver
      sion</a></td>
     </tr>
   </table>
  </div>
  <div class="content">
  </div>
 </body>
</html>
```

Login page

In the login page, it consists of logo div and content div only as the tabs div will only be shown after login. The right of the content div is some labels to indicate the administrators enter the username and password. The login button is located at the bottom of the page.



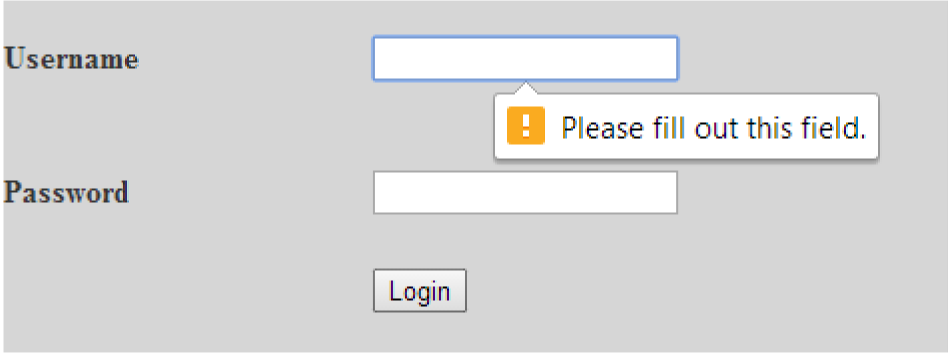
Latia CMS

Login

Username admin_A

Password ••••••••••••••••••••

The username textbox is set to be focus when the page loading completed. The textbox of username and password is set to be required so that the administrator must fill in this two textbox. Each of the textbox has set its id so that it can pass to PHP by POST method.

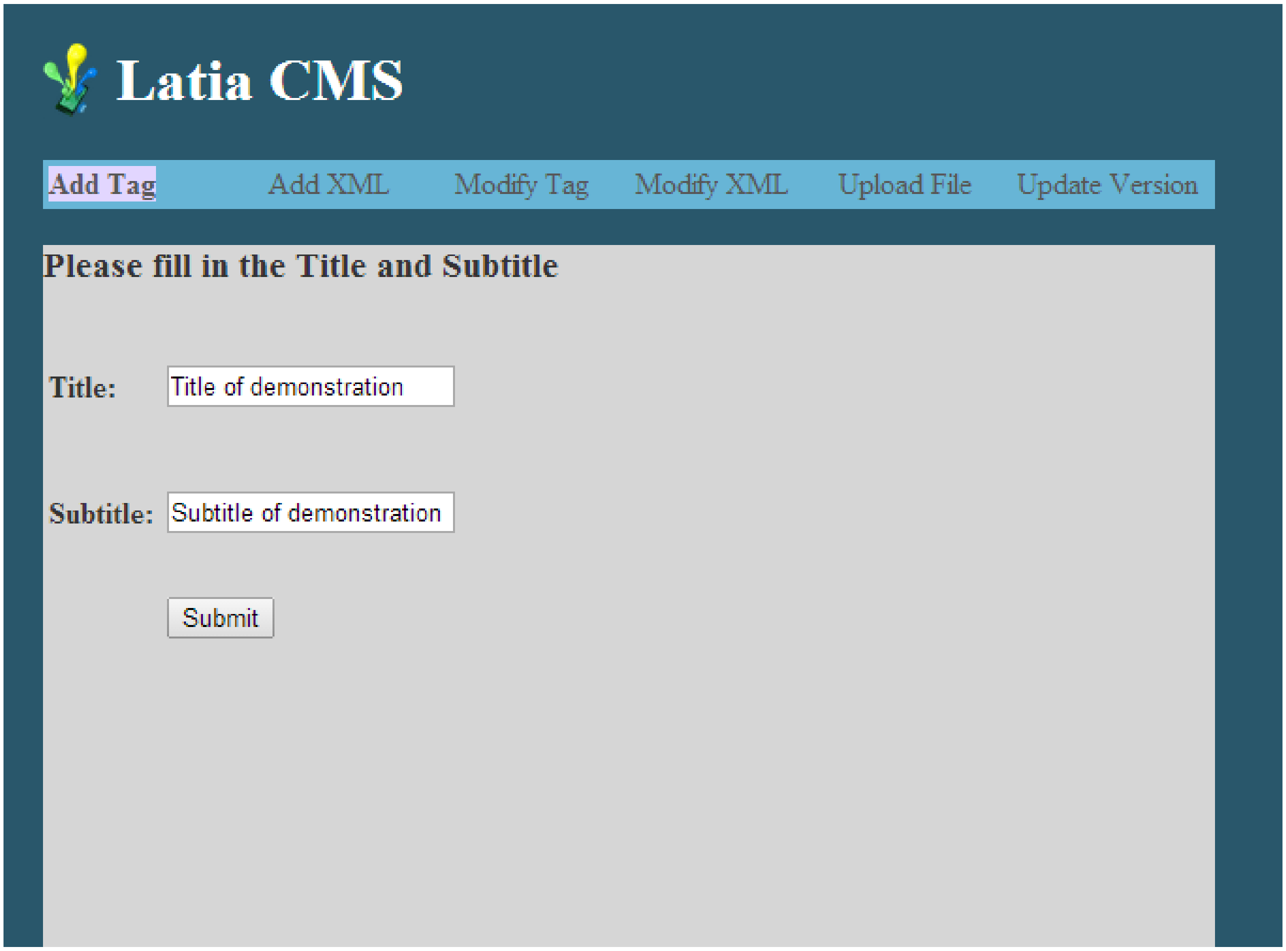


Please fill out this field.

```
<div class="content">
  <h3>Login</h3>
  <form action="" method="POST" id="login">
                      <table width="600" border="0">
                      <tr>
                      <td><h4>Username</h4></td>
                      <td><input type="text" id="uname" class="input_text_field" autofocus re
                      quired/></td>
                      </tr>
                      <tr>
                      <td><h4>Password</h4></td>
                      <td><input type="password" id=" pwd" class="input_text_field"
                       required/></td>
                      </tr>
                      <tr>
                      <td></td>
                      <td><input type="submit" name="Submit" value="submit"></td>
                      </tr>
                      </table>
            </form>
</div>
```

Add Tag page

After the login success, it will first redirect to the add tag webpage first. In the add tag page, the content div contains two textbox, title and subtitle. And at the bottom, there is a submit button to pass the parameters to the PHP using POST method.



Please fill in the Title and Subtitle

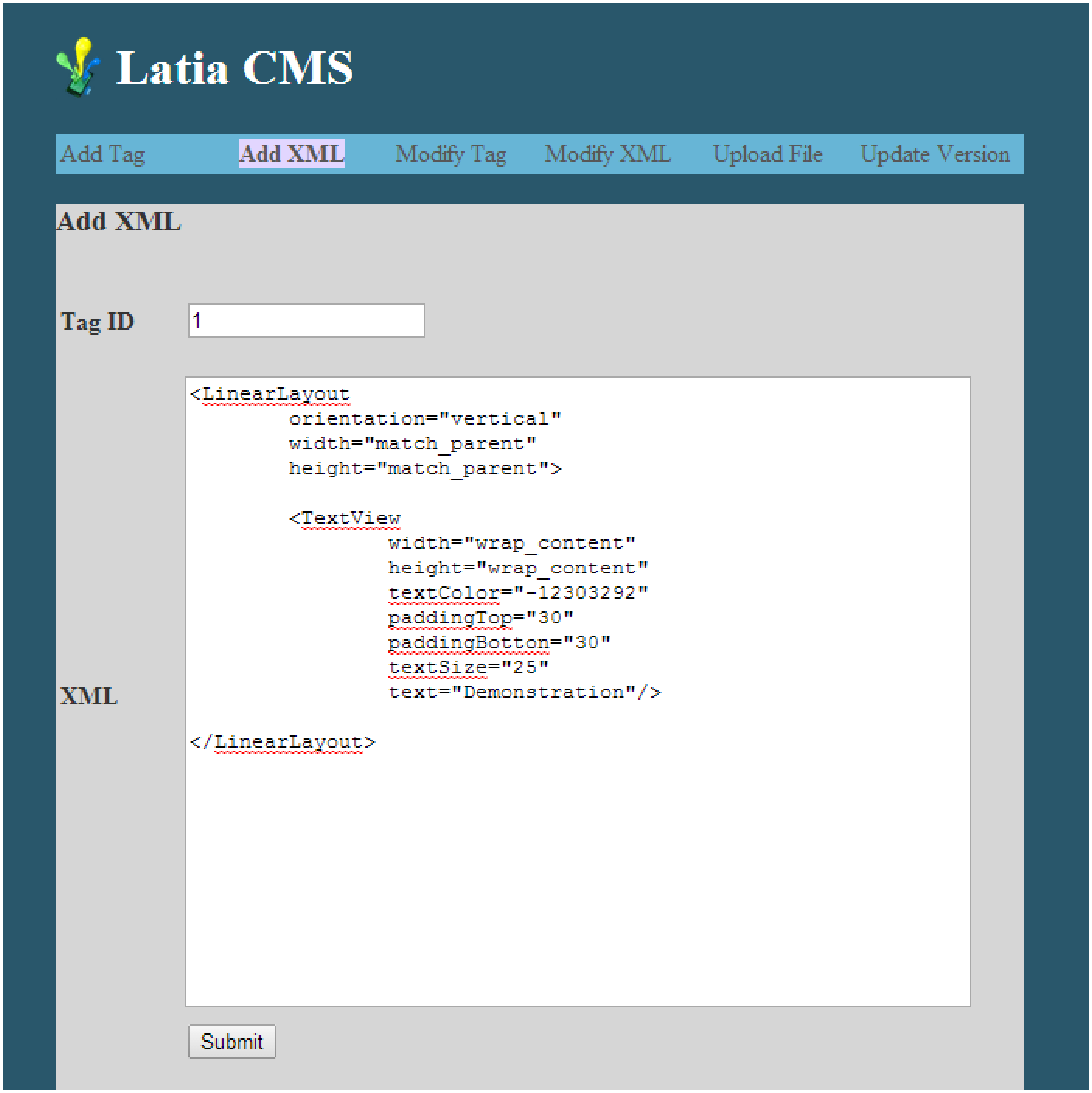
Title: Title of demonstration

Subtitle: Subtitle of demonstration

```
<div class="content">
        <h3>Please fill in the Title and Subtitle</h3>
        <form id="add_tag " name="add_tag" method="post" action="add_tag.php">
        <table width="254" border="0">
                <tr>
                        <td width="111"><h4>Title:</h4></td>
                        <td width="982"><input type="text" id="title" required/></td>
                </tr>
                <tr>
                        <td><h4>Subtitle:</h4></td>
                        <td><input type="text" id="subtitle" required/></td>
                </tr>
                <tr>
                        <td height="41"></td>
                        <td><input type="submit" name="Submit" value="Submit" /></td>
                </tr>
         </table>
        </form>
</div>
```

Add XML page

In the content div of add xml page, it consists a textbox for entering the Tag ID and a text area for entering the layout xml code. The submit button is located at the bottom of the page.



Add XML

Tag ID: 1

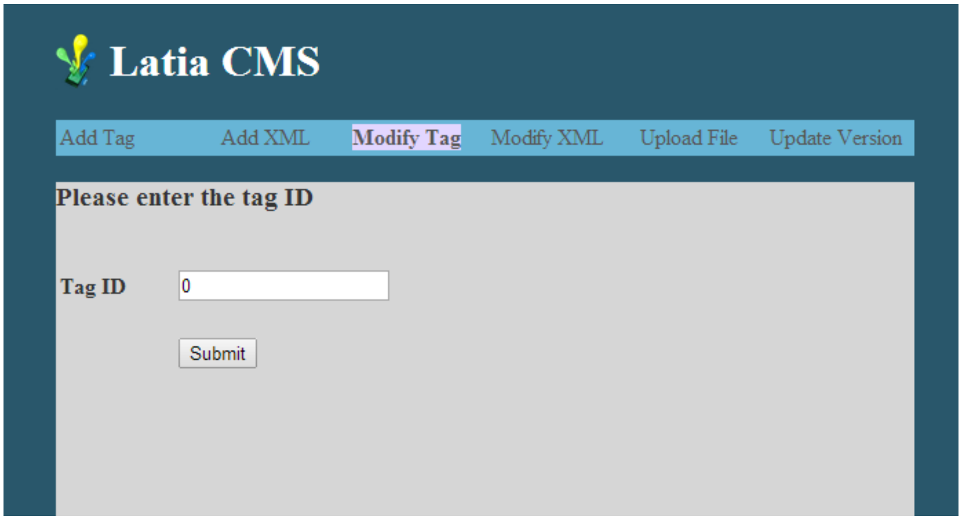
XML:

```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">

        <TextView
                width="wrap_content"
                height="wrap_content"
                textColor="-12303292"
                paddingTop="30"
                paddingBotton="30"
                textSize="25"
                text="Demonstration"/>

</LinearLayout>
```

```
<div class="content">
        <h3>Add XML</h3>
        <form id="add_xml" name="add_xml" method="post" action="add_xml.php">
        <table border="0">
                <tr>
                        <td width="110"><h4>Tag ID</h4></td>
                        <td width="563"><input type="text" id="tid" required/></td>
                </tr>
                 <tr>
                        <td><h4>XML</h4></td>
                        <td><textarea id="xml" style=”overflow:auto;resize:none;"
                        required></textarea></td>
                 </tr>
                <tr>
                        <td></td>
                        <td><input type="submit" name="Submit" value="Submit" /></td>
                </tr>
        </table>
        </form>
</div>
```

Modify Tag page

The modify tag page have two subpages. The first subpage contains a textbox for administrator entering the Tag ID that they want to modify its information. A submit button is located at the bottom of the page. After passing the submit button, it will send the parameter to PHP.

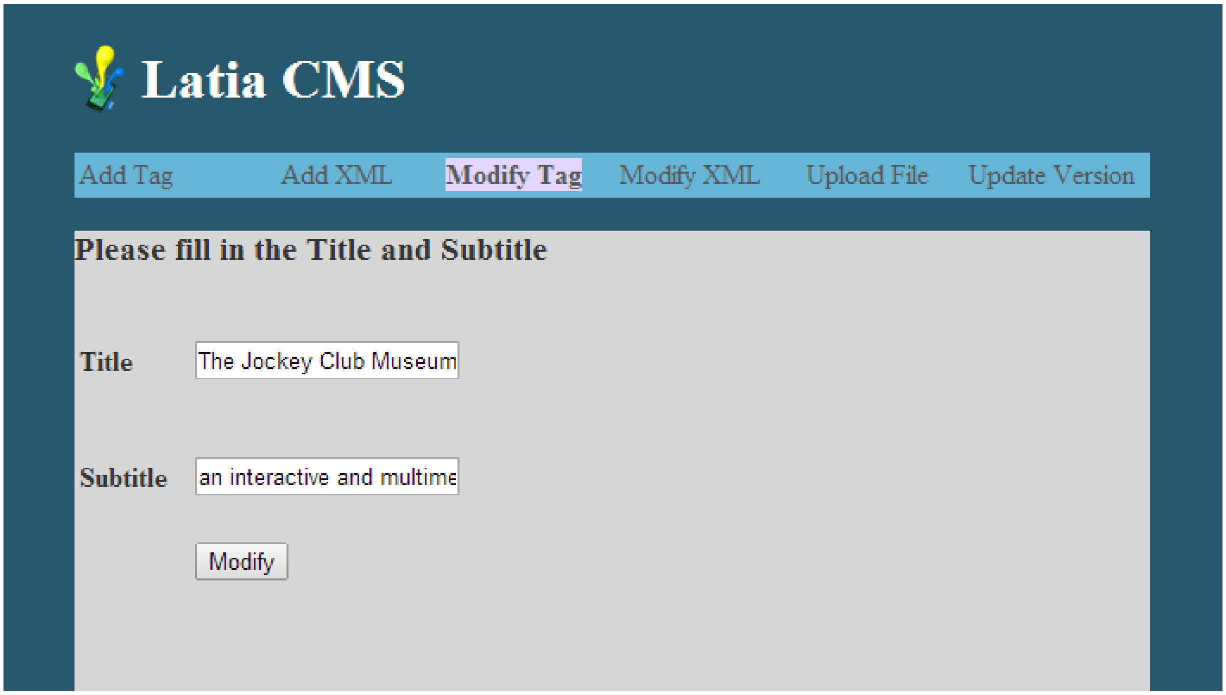


Please enter the tag ID

Tag ID 0

```
<div class="content">
   <h3>Please enter the tag ID</h3>
   <form id="modify_tag " name="modify_tag " method="post" action=" modify_tag.php">
        <table width="388" border="0" >
                <tr>
                <td width="80"><h4>Tag ID</h4></td>
                <td width="298"><input type="text" id="tid" required/></td>
                </tr>
                <tr>
                <td></td>
                <td><input type="submit" name="Submit" value="Submit" /></td>
              </tr>
      </table>
   </form>
</div>
```

If the Tag ID exists, the PHP will send the tag information back to this page. The tag information will show in the textbox so that administrators can modify it. There will be a modify button at the bottom.



Latia CMS

Please fill in the Title and Subtitle

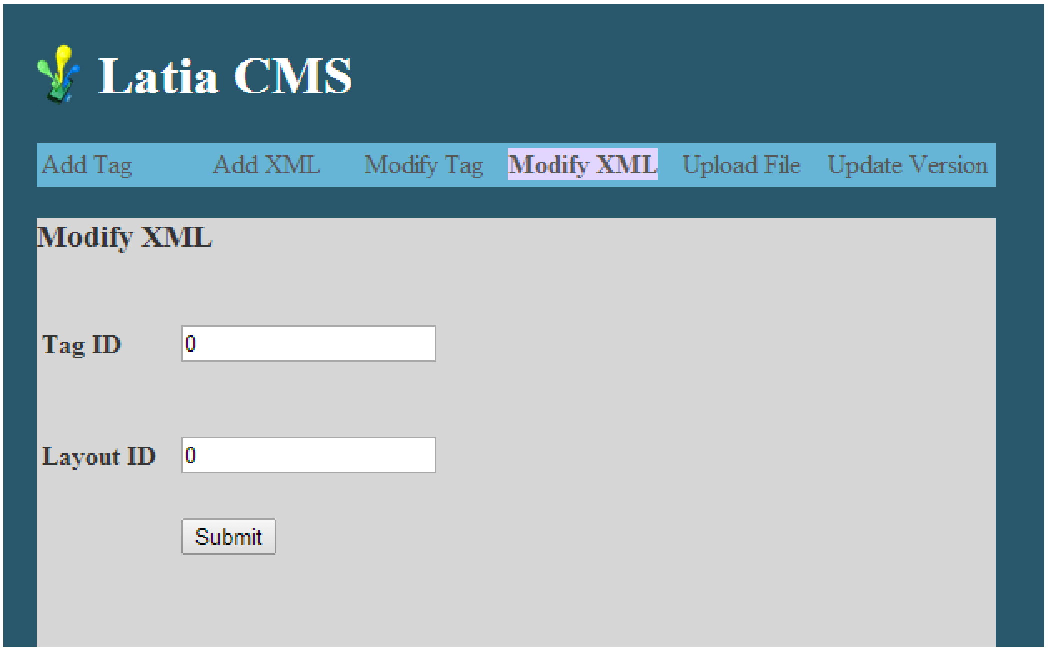
Title: The Jockey Club Museum

Subtitle: an interactive and multime

```
<div class="content">
        <h3>Please fill in the Title and Subtitle</h3>
         <form id="modify_tag " name="modify_tag " method="post" >
                <table width="281" height="80" border="0">
                <tr>
                <td width="60"><h4>Title</h4></td>
                <td width="205"><input type="text" name="title" value= "<?php if(isset($title))
                echo $title; ?>" required/></td>
                </tr>
                <tr>
                <td><h4>Subtitle</h4></td>
                <td><input type="text" name="subtitle" value= "<?php if(isset($subtitle)) echo
                 $subtitle; ?>" required/></td>
                </tr>
                <tr>
                <td></td>
                <td><input type="submit" value="Modify" /></td>
                </tr>
                </table>
                <input type = "hidden" name = "tid" value="<?php if(isset($tag_id)) echo $tag_id;?>"/>
         </form>
</div>
```

Modify XML page

Similar to modify tag page, it separated into two subpages. The first subpage contains two textbox, Tag ID and Layout ID. Under them is a submit button. After passing the submit button, it will send the parameter to PHP.



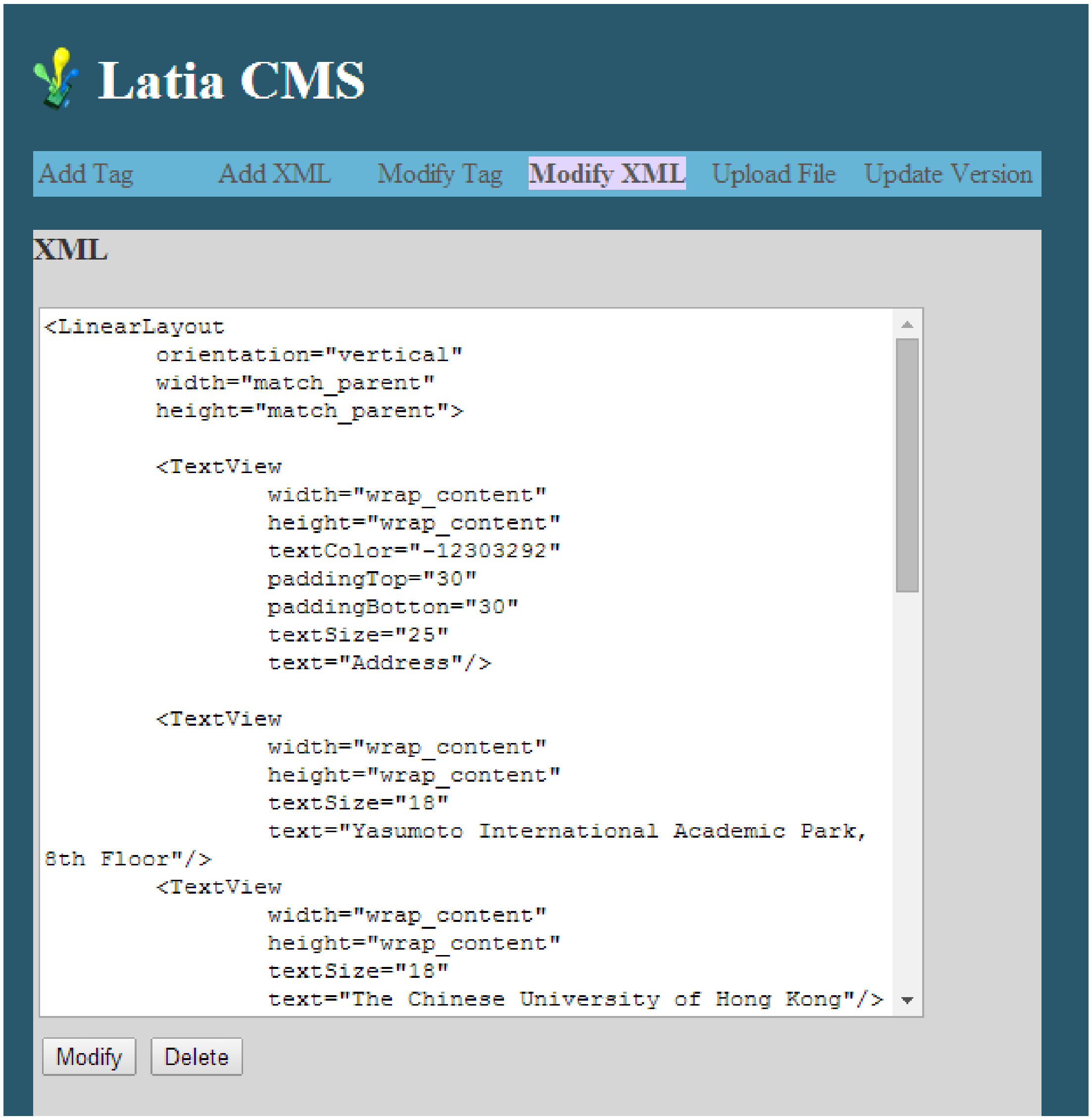
Modify XML

Tag ID 0

Layout ID 0

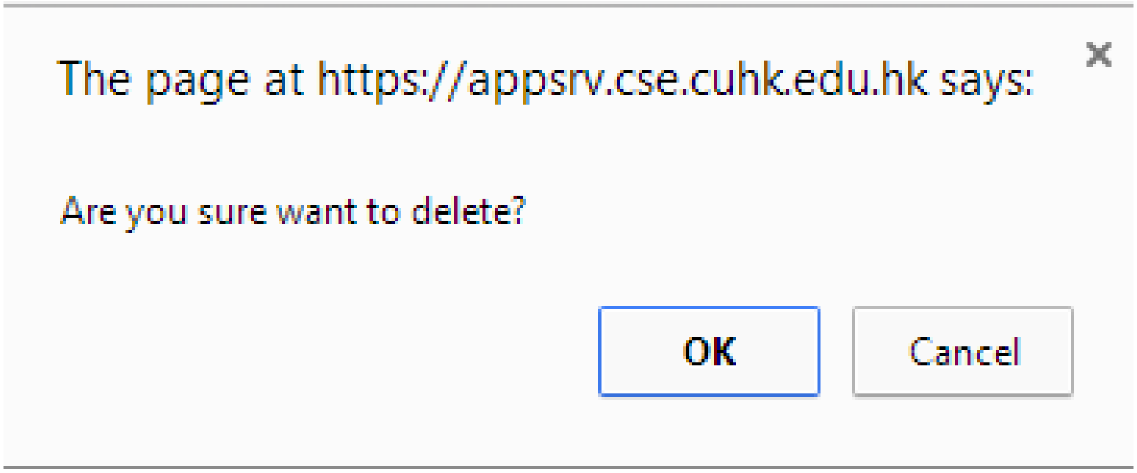
```
<form id="modify_xml " name="modify_xml " method="post" action="modify_xml.php">
        <table width="265" border="0">
                <tr>
                 <td width="77"><h4>Tag ID</h4></td>
                <td width="172"><input type="text" name="tid" required/></td>
                </tr>
                <tr>
                <td><h4>Layout ID</h4></td>
                <td><input type="text" name="lid" required/></td>
                </tr>
                <tr>
                <td></td>
                <td><input type="submit" name="Submit" value="Submit" /></td>
                </tr>
        </table>
</form>
```

If the Tag ID and Layout ID exist, the PHP will send the layout information back to this page. The layout xml code will show in the text area so that administrators can modify it. There will be a modify button and a delete button at the bottom.



```
<LinearLayout
        orientation="vertical"
        width="match_parent"
        height="match_parent">

        <TextView
                width="wrap_content"
                height="wrap_content"
                textColor="-12303292"
                paddingTop="30"
                paddingBotton="30"
                textSize="25"
                text="Address"/>

        <TextView
                width="wrap_content"
                height="wrap_content"
                textSize="18"
                text="Yasumoto International Academic Park,
8th Floor"/>
        <TextView
                width="wrap_content"
                height="wrap_content"
                textSize="18"
                text="The Chinese University of Hong Kong"/>
```

When press the delete button, it will pop-up a message to ask for confirm so that it can prevent the administrator press it wrongly.



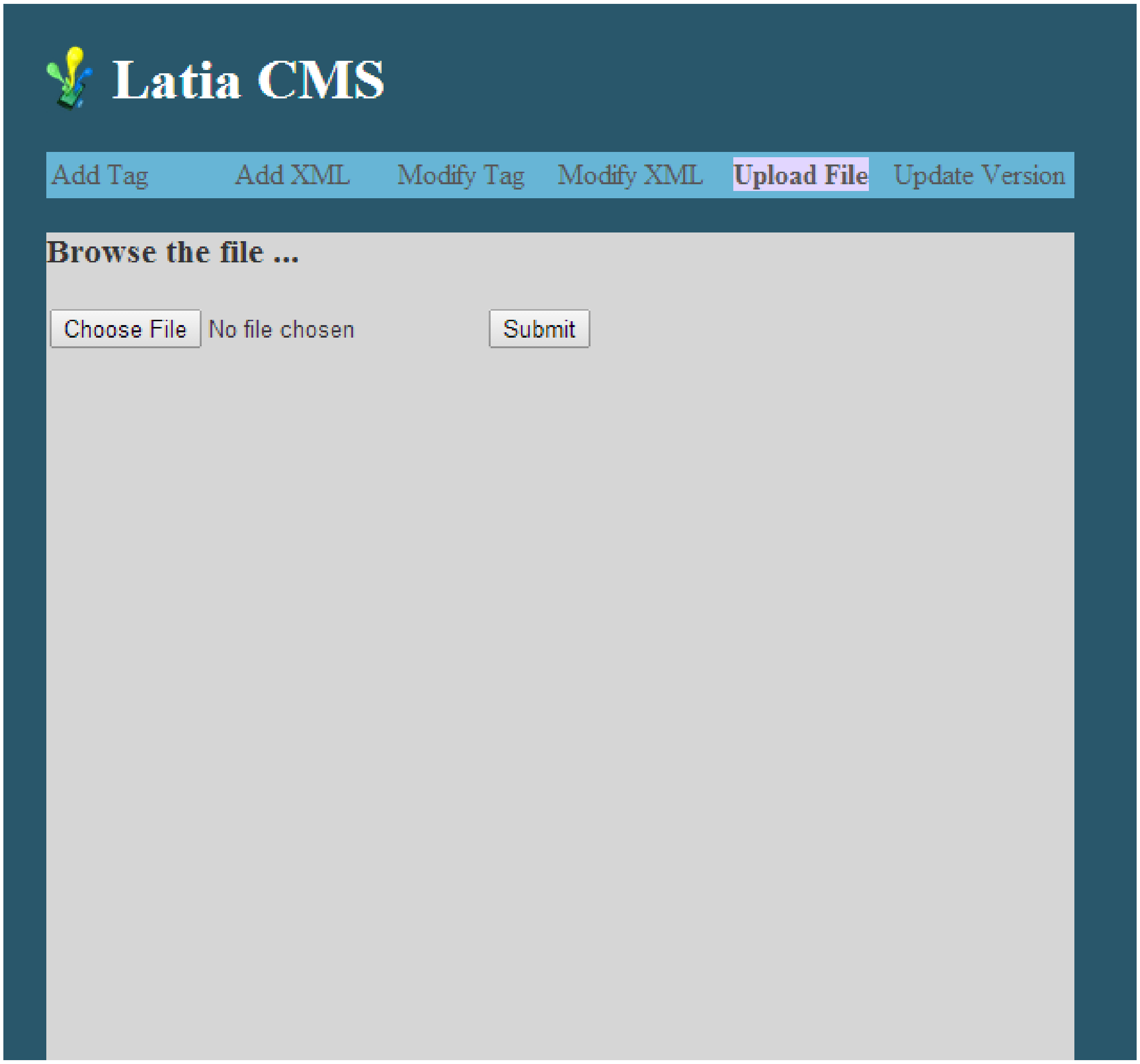
The page at https://appsrv.cse.cuhk.edu.hk says:

Are you sure want to delete?

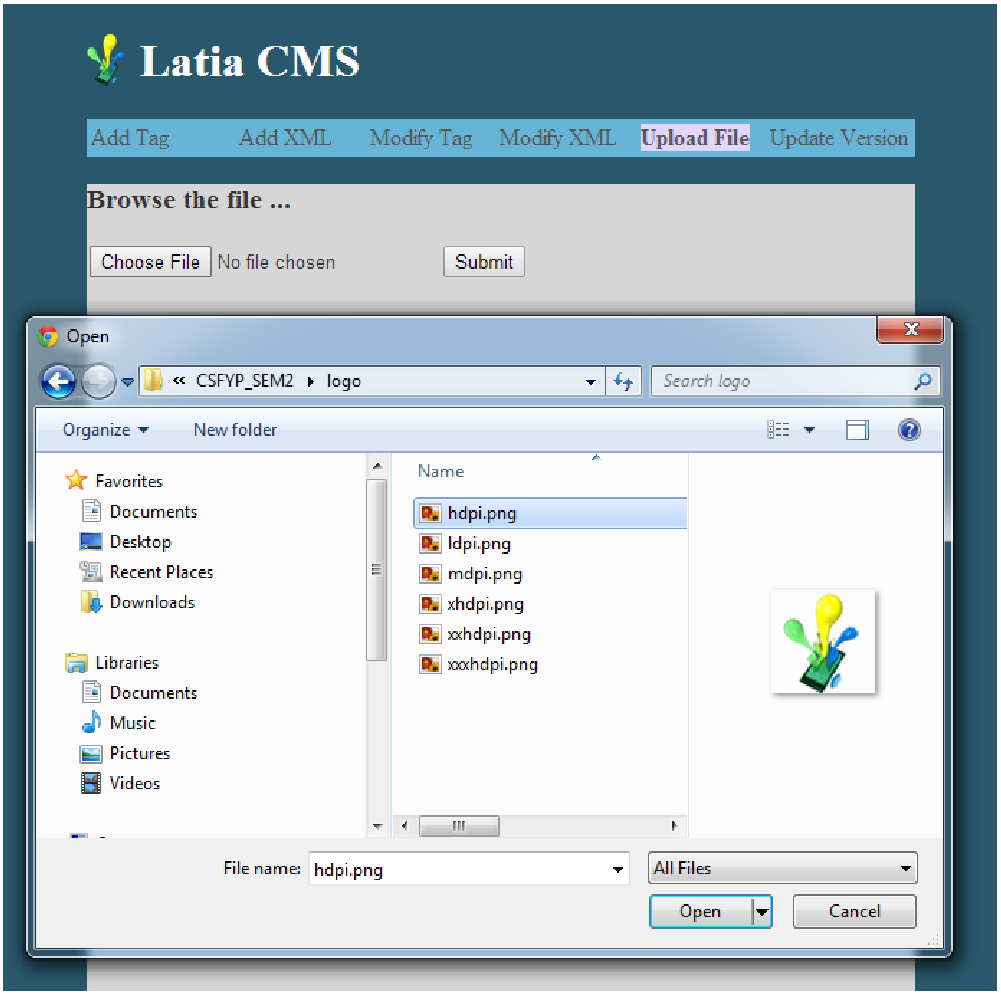
```
<div class="content">
        <h3>XML</h3>
        <form id="modify_xml" name="modify_xml" method="post" action="">
                <table width="319" height="81" border="0">
                        <tr>
                        <td width="246"><textarea name="xml" style="overflow:
                        auto;resize:none;" required><?php if(isset($xml))
                        echo trim($xml);?></textarea></td>
                        </tr>
                         <tr>
                        <td><input type="submit" name="submit" value="Modify" />
                        <input type="submit" name="delete" value="Delete" onClick="return
                        confirm('Are you sure want to delete?');" />
                        </td>
                        </tr>
                 </table>
                <input type="hidden" name="tid" value="<?php if(isset($tag_id)) echo
                $tag_id; ?>" />
                <input type="hidden" name="lid" value="<?php if(isset($layout_id)) echo
                $layout_id; ?>" />
        </form>
</div>
```

Upload File page

In upload file page, there is a file browser button for administrators to browse the resource that in local directory. The submit button is next to it



When the administrators pass the file browser button, the file browser will be popup and administrators can choose the resource that they need to upload.



Browse the file ...

Choose File No file chosen

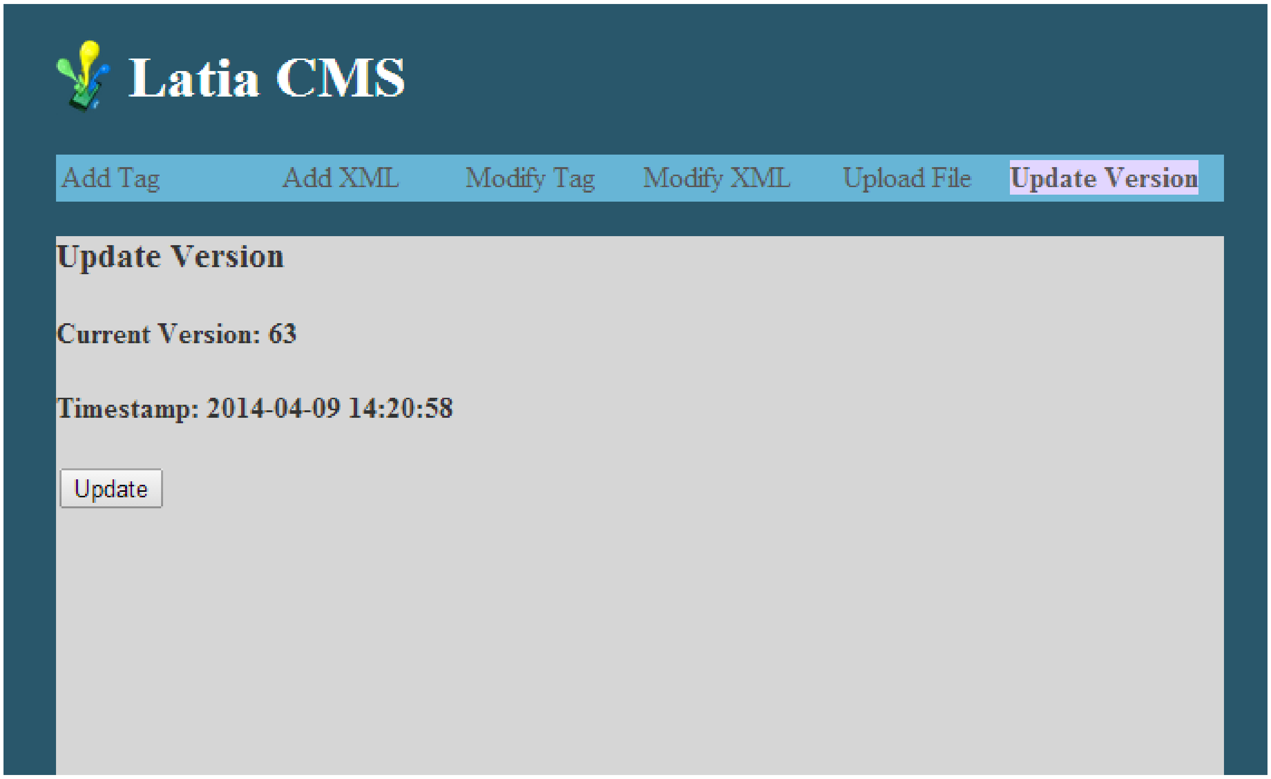
hdpi.png
ldpi.png
mdpi.png
xhdpi.png
xxhdpi.png
xxxhdpi.png

File name: hdpi.png

```
<div class="content">
        <h3>Browse the file ...</h3>
        <form action="upload_resource.php" method="post" enctype="multipart/
        form-data">
                <input type="file" id="file" required/>
                <input type="submit" name="submit" value="Submit">
        </form>
</div>
```

Upload Version page

The update version page consist two labels, one is to show the current version and other is to show the released time. At the bottom, there is an update button. After the administrator pass it, the PHP will send the new version number and the upload version page will be reloaded.



Update Version

Current Version: 63

Timestamp: 2014-04-09 14:20:58

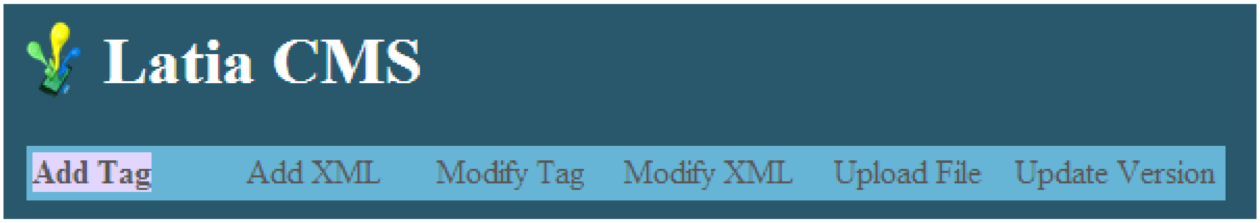
```
<div class="content">
        <h3>Update Version</h3>
        <form id="update_version"  method="post" action="">
                <h4>Current Version: <?php echo $max; ?></h4>
                <h4>Timestamp: <?php echo $rtime; ?></h4>
                <input type="submit" name="Submit" value="Update" />
        </form>
</div>
```

7.4.2 CSS

We have defined the webpage layout style in the extended CSS file. This approach can unify all the webpage style. Also, it is easier for us to change the all webpages style in future by simply modify the CSS.

```
b.current{
background-color:#E2D6FF;
color:#575757;}
a:link{
color:#575757;
text-decoration:none;}
a:visited{
color:#575757;
text-decoration:none;}
a:hover{
background-color:#E2D6FF;
color:#555555;
text-decoration:none;}
a:active{
background-color#67B5D6;
color:#575757;
text-decoration:none;}
body{
margin-left: auto;
margin-right: auto;
width:70%;
background-color:#29576B;
color:#FFFFFF;}
div.tabs{
background-color:#67B5D6;
color:#575757;
width:100%;
text-align:left;}
div.content{
background-color:#D6D6D6;
color:#333333;
width:100%;
height:700px;
text-align:left;}
```

In order to indicate which hyperlink of page in tab bar the mouse is over on it, we have defined the background color parameter in a:hover so that the background color change whenever the mouse is over the hyperlink. Also, we have define the background color of current tab which set to be different to indicate the current page



7.4.3 PHP

After the administrator entering the parameter in the frontend created by HTML5 and CSS, it will be passed to the PHP. So the frontend can modify the database thought PHP in backend. Each of the webpages needs to connect the database and the following is the PHP function for each webpage respectively.

Login function

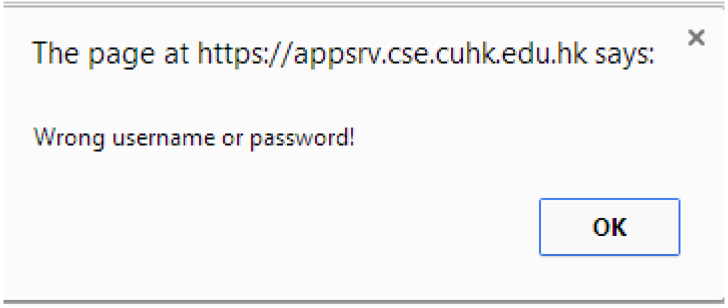
When the login function is called, it will first check whether it there exists the login time session and if the time is expired. If the administrators are already login and not expired yet, it will redirect the administrators to main page of CMS.

If the administrators are not login yet, the function will query the database to get the password from admin table. Then, it will compare the password fetch from database with the encrypted password received from HTML. If the password entered is correct, it will create a login time session to store the time and redirect to the main page of CMS

```
<?php
	// Redirect to the main page if users are already login and not expired
	session_start();
	if(isset($_SESSION['login_time']) && ((time() - $_SESSION['login_time']) <= 1800))
	 {
			header("refresh:0;url=main" );
		exit;
	}
	if(isset($_POST['uname']) && isset($_POST['pwd']))
	{
		//Connect to DB
		$db = new DATABASE_CONNECT();
		$db = $db->getDatabase();
		//Get password from database
		$sql = $db->prepare("SELECT pwd FROM admin_table WHERE uname = ?");
		$sql->bind_param('s', $_POST['uname']);
		$sql->execute();
		$sql->bind_result($password);
```

```
        $sql->fetch();
        $sql->close();
        //Validate the input password
        if(encrypt($_POST['pwd'])!=$password)
            //incorrect password
            echo "<script type='text/javascript'>alert('Wrong username or
            password!'); </script>";
        else
        {
            // create the session to store the current time
            session_start();
            $_SESSION[' login_time''] = time();
            //redirect to main page
            header("refresh:0;url=main" );
        }
    }
?>
```

If the password is incorrect, PHP will alert the administrators and popup an error message. Then, it will return back to HTML.



The page at https://appsrv.cse.cuhk.edu.hk says:

Wrong username or password!

In the following function will be called for all the webpage except login. It will check if the administrators are already login by checking the login time session. If the login duration is more than 30 minutes, it will redirect back to login page and require administrator login again.

```
<?php
   //Redirect to login page if the users not login yet
   session_start();
   if(!isset($_SESSION['login_time']) || ((time() - $_SESSION['login_time']) > 1800))
   {
         header("refresh:0;url=login" );
         exit;
   }
?>
```

Add Tag function

The add tag function will be called when the submit button in add tag page is pressed. The function will receive the title and subtitle. Then, it will query the database to get the Tag ID for generating a new Tag ID. It will insert the new tag information to the tag table and popup the Tag ID to the administrators. Finally, it will return back to the add tag page.

```
<?php
	 //Get the title and subtitle
	$title = $_POST['title'];
	$subtitle = $_POST['subtitle'];

	if(isset($title) && isset($subtitle))
	{
			//Connect to database
			$db = new DATABASE_CONNECT();
			$db = $db->getDatabase();
```

```
            //Get the current tag id
            $sql = "SELECT COUNT(tid) AS current_tag_id FROM tag_table";
            $query = $db->query($sql);

            if(($obj = $query->fetch_object())!=NULL)
                    $current_tag_id = $obj->current_tag_id;
            if(isset($current_tag_id))
            {
                //Insert the new tag information
                $sql = $db->prepare("INSERT INTO tag_table(tid, title, subtitle)
                VALUES(?, ?, ?)");
                $sql->bind_param('iss',$current_tag_id, $title, $subtitle);
                $sql->execute();
                //Popup the tag id
                echo "<script> alert('The record is added with
                tag_id: ".$current_tag_id."'); </script>";
                //Redirect to HTML
                header("refresh:0;url=add_tag" );
                exit;
            }
    }
?>
```

Add Layout function

The add layout function is called when the administrator pressed the submit button in the add xml page. The function will receive the Tag ID and the layout xml code. If the Tag ID exists, it will generate the new Layout ID by counting number of layout. Then, it will query the database for inserting the new layout information. Finally it will pop up the new Layout ID to administrators.

```
<?php
        function add_xml($tag_id, $xml)
        {
                //Connect to database
                $db = new DATEBASE_CONNECT();
                $db = $db->getDatabase();
                //Check if the tag id exists
                $sql = $db->prepare("SELECT count(*) FROM tag_table WHERE tid = ?");
                $sql->bind_param('i', $tag_id);
                $sql->execute();
                $sql->bind_result($count);
                $sql->fetch();
                $sql->close();
                if($count==0)
                {
                        echo "<script> alert('The tag id is not found!'); </script>";
                        header("refresh:0;url=add_xml" );
                        exit;
                }
                //Count the layout id
                $current_layout_id = 0;
                $sql = $db->prepare("SELECT COUNT(lid) AS current_layout_id
                FROM layout_table WHERE tid = ?");
                $sql->bind_param('i', $tid);
                $sql->execute();
                $sql->bind_result($current_layout_id);
                $sql->fetch();
                $sql->close();
                //Insert information
                $current_layout_id = $current_layout_id;
                $sql = $db->prepare("INSERT INTO layout_table(tid, lid, xml) VALUES(?,?,?)");
                $sql->bind_param('iis', $tag_id, $current_layout_id, $xml);
                $sql->execute();
                $sql->close();
                //Return the layout id
                echo "<script> alert('The record is added with layout_id:
                ".$current_layout_id."'); </script>";
                header("refresh:0;url=add_xml" );
                exit;
        }
        //Get the tag id and layout xml code
        $tag_id = $_POST['tag_id'];
        $xml = $_POST['xml'];
        if(isset($tag_id) && isset($xml))
                add_xml($tag_id, $xml);
        //Redirect to add xml page
        else
                header("refresh:0;url=add_xml" );
?>
```

Modify Tag function

The modify function is called when the administrators pressed the modify button in the modify tag page. The function will either receive the Tag ID only or receive Tag ID, title and subtitle.

If the function receives Tag ID only, the function will query the database to check if the Tag ID exists. If it exists, the function will get the title and subtitle in tag table. If not, the function return error message.

If the function receives Tag ID, title and subtitle, it will query the database for update the tag information.

Finally, it will pop up a success message and return to the modify tag page.

```
<?php
        function get_tag_info($tag_id, &$title, &$subtitle)
        {
                //Connect to database
                $db = new DATABASE_CONNECT();
                $db = $db->getDatabase();
                //Check if the tag id exist
                $sql = $db->prepare("SELECT count(*) FROM tag_table WHERE tid = ?");
                 $sql->bind_param('i', $tid);
                $sql->execute();
                $sql->bind_result($count);
                $sql->fetch();
                $sql->close();
                //Alert error message and redirect to the
                //modify tag page if the tag is not found
                if($count==0)
                {
                        echo "<script> alert('The tag id is not found!'); </script>";
                        header("refresh:0;url=modify_tag" );
                        exit;
                }
                //Get the title and subtitle of that tag
                $sql = $db->prepare("SELECT title, subtitle FROM tag_table WHERE tid = ?");
                $sql->bind_param('i', $tag_id);
                $sql->execute();
                $query = $db->query($sql);
                $sql->bind_result($title, $subtitle);
        $sql->fetch();
        $query = $db->query($sql);
        $sql->bind_result($title, $subtitle);
        $sql->fetch();
        $sql->close();

    }
    function modify_tag_info($tag_id, $title, $subtitle)
    {
        //Connect to database
        $db = new DATABASE_CONNECT();
        $db = $db->getDatabase();
        //update the title and subtitle
        $sql = $db->prepare("UPDATE tag_table SET title = ?, subtitle = ?
        WHERE tid = ?");
        $sql->bind_param('ssi',$title, $subtitle, $tag_id);
        $sql->execute();
        $sql->close();
        echo "<script> alert('The row is modified'); </script>";
        //Redirect to the modify tag page
        header("refresh:0;url=modify_tag" );
        exit;
    }
    //Get the parameters
    $tag_id = $_POST['tag_id'];
    $title = $_POST['title'];
    $subtitle = $_POST['subtitle'];
    //Update with the modified tag information
    if(isset($title) && isset($subtitle) && isset($tag_id))
        modify_tag_info($tag_id,$title,$subtitle);
    //Get the tag information
    else if(isset($tag_id))
        get_tag_info($tag_id, $title, $subtitle);
    else
        header("refresh:0;url=modify_tag” );
?>
```

Modify Layout function

The modify layout function is called when the administrator enter the Tag ID, Layout ID and pressed submit in first subpage of modify xml page, or pressed the modify button or delete button in the other subpage.

If the submit button is pressed, the function will retrieve the Tag ID and Layout ID. It will use these two parameters queries the database to retrieve the layout xml code if the Tag ID and Layout ID exist. If not, it will show error message and return. After that, it will return the layout xml code back to modify xml.

If the modify button is pressed, the function will receive the modified layout xml code. Then, it will query layout table to update the layout information and return back to modify xml page.

If the delete button is pressed, the function will receive delete message. Then, it will request deletion on that layout information in layout table. Finally, it will return back to the modify xml page.

```
<?php
	function get_xml($tag_id,$layout_id,&$xml)
	{
		//Connect to database
		$db = new DATABASE_CONNECT();
		$db = $db->getDatabase();
		//Check if it is Exist
		$sql =  $db->prepare("SELECT COUNT(*) FROM layout_table WHERE tid = ?
		AND lid = ?");
		$sql->bind_param('ii', $tag_id, $layout_id);
		$sql->execute();
		$sql->bind_result($count);
		$sql->fetch();
		$sql->close();
		//Alert error message and redirect back if not exist
		if($count==0)
		{
				echo "<script> alert('The tag or layout is not found!');</script>";
				header("refresh:0;url=modify_xml" );
				exit;
		}
```

```
        //Get the layout xml code
        $sql =  $db->prepare("SELECT xml FROM layout_table WHERE tid = ?
        AND lid = ?");
        $sql->bind_param('ii', $tag_id, $layout_id);
        $sql->execute();
        $sql->bind_result($xml);
        $sql->fetch();
        $sql->close();
}
function modify_xml($tag_id,$layout_id,$xml)
{
        //Connect to database
        $db = new DATABASE_CONNECT();
        $db = $db->getatabase();
        //Update the layout information
        $sql = $db->prepare("UPDATE layout_table SET xml = ? WHERE tid = ?
        AND lid = ?");
        $sql->bind_param('sii', $xml, $tag_id, $layout_id);
        $sql->execute();
        $sql->close();
        echo "<script> alert('The XML is modified');</script>";
        header("refresh:0;url=modify_xml " );
        exit;
}
function delete_xml($tag_id,$layout_id)
{
        //Connect to database
        $db = new DATABASE_CONNECT();
        $db = $db->getDatabase();

        //Check if it is Exist
        $sql =  $db->prepare("SELECT COUNT(*) FROM layout_table WHERE tid = ?
         AND lid = ?");
        $sql->bind_param('ii', $tag_id, $layout_id);
        $sql->execute();
        $sql->bind_result($count);
        $sql->fetch();
        $sql->close();
        // Alert error message and redirect back if not exist
        if($count==0)
        {
            echo "<script> alert('The tag or layout is not found!');</script>";
            header("refresh:0;url=modify_xml" );
            exit;
        }
        //Sort the database
        $sql = $db->prepare("SELECT MAX(lid)FROM layout_table WHERE tid = ?");
        $sql->bind_param('i', $tag_id);
        $sql->execute();
        $sql->bind_result($max);
        $sql->fetch();
        $sql->close();
        //Delete
        $sql =  $db->prepare("DELETE FROM layout_table WHERE tid = ? AND lid = ?");
        $sql->bind_param('ii', $tag_id, $layout_id);
        $sql->execute();
        $sql->close();
?>
```

Upload Resource function

The upload resource function is called when the administrator click the upload button in upload file page. The function will receive the resource file. If file error occurred, the function show error message and return back to upload file page. It not, it will check if the resource exists in resource table. If yes, the function will request database update with the new resource data. If not, it will request for new insertion. The function will return back to upload file page if the process completed.

```
<?php
        if(isset($_FILES["file"]))
        {
                //Return to the upload file page if file error occurred
                if($_FILES["file"]["error"] > 0)
                {
                        echo "<script> alert('File error');</script>";
                        header("refresh:0;url=upload_file " );
                        exit;
                }
                //Connect to database
                $db = new DATABASE_CONNECT();
                $db = $db->getDatabase();
                //Check if the resource exists
                $sql = $db->prepare("SELECT count(*) FROM resource_table
                 WHERE rname = ?");
                $sql->bind_param('s', $_FILES["file"]["name"]);
                $sql->execute();
                $sql->bind_result($count);
                $sql->fetch();
                $sql->close();
                //Insert the new resource
                if($count==0)
                {
                        $sql = $db->prepare("INSERT INTO resource_table(rname, utime, data)
                        VALUES(?,CURRENT_TIMESTAMP,?)");
                        $sql->bind_param('ss', $_FILES["file"]["name"],
            file_get_contents($_FILES["file"]["tmp_name"]));
            $sql->execute();
            $sql->close();
            echo "<script> alert('The file is uploaded'); </script>";
            header("refresh:0;url=upload_file");
            exit;
        }
        //Update the resource
        else
        {
            $sql = $db->prepare("UPDATE resource_table SET data = ?,
            update_time = CURRENT_TIMESTAMP WHERE name = ?");
            $sql->bind_param('ss', file_get_contents($_FILES["file"]["tmp_name"]),
            $_FILES["file"]["name"]);
            $sql->execute();
            $sql->close();
            echo "<script> alert('The file is updated'); </script>";
            header("refresh:0;url=upload_file ");
        }
    }
?>
```

Update Version function

The update version function is called when the update version is loading or administrator pressed the update button. When the page is loading, the function will query the database to get the most current version number in version table. Then, it will request the release time of that version and return these two parameters back to the page.

If the administrator pressed the update button, the function will get the current version number and generate the new version number. Then, it will request inserting the new version into version table with current time. It will return to the update version page after the process is completed.

```
<?php
        function get_version(&$max,&$timestamp)
        {
                //Connect to database
                $db = new DATABASE_CONNECT();
                $db = $db->getDatabase();
                //Get the most current version number
                $sql = $db->prepare("SELECT MAX(vid) FROM version_table");
                $sql->execute();
                $sql->bind_result($max);
                $sql->fetch();
                $sql->close();
                //Get the release time of the current version number
                $sql = $db->prepare("SELECT rtime FROM version_table where vid = ".$max);
                $sql->execute();
                $sql->bind_result($timestamp);
                $sql->fetch();
                $sql->close();
        }
        function update_version($max)
        {
                //Connect to database
                $db = new DATABASE_CONNECT();
                $db = $db->getDatabase();
                //Generate the version number
                $max = $max +1;
                //Insert the new version information
                $sql = $db->prepare("INSERT INTO version_table (vid, rtime) VALUES
                (?,CURRENT_TIMESTAMP)");
                $sql->bind_param('i', $max);
                $sql->execute();
                $sql->fetch();
                $sql->close();
                echo "<script> alert('The version is updated'); </script>";
                header("refresh:0;url=update_version");
                exit;
        }
        get_version($max,$timestamp);
        if($_POST['Submit']=="Update")
                update_version($max);
?>
```

Update System

The update system uses the AsyncTask provided by java to implement. Since networking access requires time especially mobile device has unstable network connection, hence it is better to use a background thread to do downloading.

To simplify our work, we have implemented an abstract class which can only download content through HTTP.

```
// Given a URL, establishes an HttpUrlConnection and retrieves
// the web page content as a InputStream, which it returns as
// a string.
protected byte[] downloadUrl(String myurl) throws IOException {

    // Only display the first 500 characters of the retrieved
    // web page content.

  URL url = new URL(myurl);
  HttpURLConnection conn = (HttpURLConnection) url.openConnection();
  conn.setReadTimeout(10000 /* milliseconds */);
  conn.setConnectTimeout(15000 /* milliseconds */);
  conn.setRequestMethod("GET");
  conn.setDoInput(true);

  // Starts the query
  conn.connect();
  int response = conn.getResponseCode();
  Log.d(this.getClass().getSimpleName(), "Download from: " + myurl);
  Log.d(this.getClass().getSimpleName(), "The response is: " + response);
  if(response != 200)
       throw new IOException();

  return readBytes(conn.getInputStream());
}
```

Callback method is needed to wait the downloading process.

```
@Override
protected void onPostExecute(byte[] result) {
  // Convert the InputStream into a string
  String message = new String(result);
        Log.d(getClass().getSimpleName(), "post execute\n" + message);
}
```

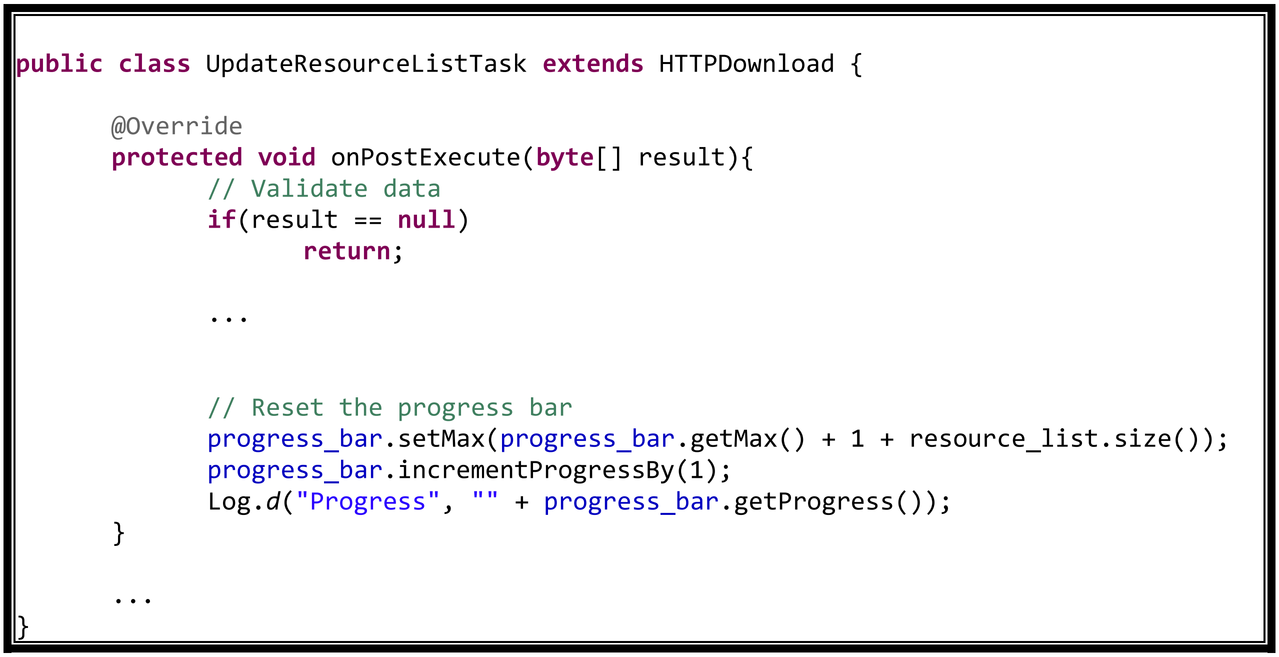
Hence, by overriding `onPostExecute()`, we can handle the result with ease.

The in-app database is very closely connected to the update system. Because all the updated metadata will be stored in the database, the efficiency is important. Android provides several permanent storage options. Among those, we have chosen to use shared preference. Since shared preference is very simple, it consists of only key-valued pairs, there are no complex SQL queries to execute, and hence the overhead on IO is quite small. However, the biggest issue is the networking problem. Since, updating time affects user experience a lot, we have to minimize the connection time. One of our improvements is the increment updating.

```
    @Override
    protected void onPostExecute(byte[] result){
        // Validate data
        if(result == null)
            return;

        // List of resource name
        ArrayList<String> resource_list = getResourceList(result);

        . . .

    }
```

Progress bar

Allowing user to know the updating progress can gain some user-experience as well, otherwise user does not know and cannot prepare when does the update finished. By updating the progress bar in every download task, we can create a trustful progress bar.



```
public class UpdateResourceListTask extends HTTPDownload {

      @Override
      protected void onPostExecute(byte[] result){
            // Validate data
            if(result == null)
                  return;

            ...


            // Reset the progress bar
            progress_bar.setMax(progress_bar.getMax() + 1 + resource_list.size());
            progress_bar.incrementProgressBy(1);
            Log.d("Progress", "" + progress_bar.getProgress());
      }

      ...
}
```

```
public class DownloadResourceTask extends HTTPDownload {
        @Override
        protected void onPostExecute(byte[] result){

                ...

                // Update progress bar
                progress_bar.incrementProgressBy(1);
                Log.d("Progress", "" + progress_bar.getProgress());

                // Check update completion
                if(progress_bar.getProgress() == progress_bar.getMax())
                {
                        Log.d(getClass().getSimpleName(), "Update Completed NEW VERSION
IS " + new_version);

                        // Intent to next activity
                        ((SplashActivity)activity).onUpdateComplete(new_version);
                }

                ...

        }

        ...

}
```

When the progress bar meets the maximum, it automatically update its version and move on to the MainActivity (History page).

```
/**
 * Intent to main activity (After update)
 */
public void onUpdateComplete(int new_version)
{
        // Get Current version
        SharedPreferences sharedPref =
        getSharedPreferences(getString(R.string.spkey), Activity.MODE_PRIVATE);
        sharedPref.edit().putInt(getString(R.string.spkey_version),
        new_version).commit();

        // Intent new activity with some delay
        new Handler().postDelayed(new Runnable() {
         @Override
         public void run() {
               final Intent intent = new
               Intent(SplashActivity.this,MainActivity.class);
              SplashActivity.this.startActivity(intent);
              SplashActivity.this.finish();
         }
    }, 1000);
}
```

NFC Intent System

In order to use the NFC service of the mobile device, NFC permission is needed to be enabled.

```
<uses-permission android:name="android.permission.NFC" />
```

In the manifest file, we have registered the intent filter of our external type.

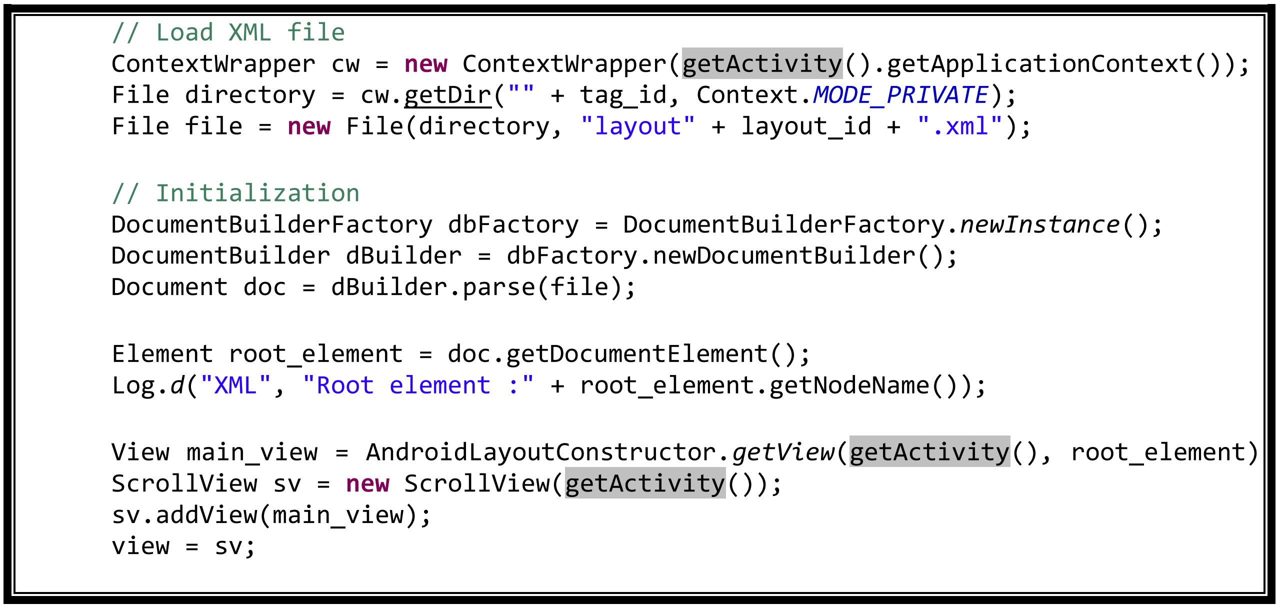
```
      <intent-filter>
                <action android:name="android.nfc.action.NDEF_DISCOVERED" />
                <category android:name="android.intent.category.DEFAULT" />
                <data
                    android:host="ext"
                    android:pathPrefix="/_____:___"
                    android:scheme="vnd.android.nfc" />
            </intent-filter>
```

Hence, whenever an Latia NFC Tag is scanned, the NFC activity in Latia will be launched. The intent will attach an NFC Tag object. We have to construct the Tag using this object.

```
//Confirm intent type
if(NfcAdapter.ACTION_NDEF_DISCOVERED.equals(getIntent().getAction())){
	Log.d(NFC_DEBUG_TAG, "NFC Intent detected");

	//Get NDEF message
	byte[] ndef_payload = NfcHelper.getNdefRecord(getIntent());

	//Construct a CTag object
	tag = new Tag(payload);
}
```

Layout Construction System

This is the key module of Latia, it responds to parse the XML file and construct a view from the file. In the implementation, we uses the DocumentBuilderFactory provided by Java to load XML file.



```
        // Load XML file
        ContextWrapper cw = new ContextWrapper(getActivity().getApplicationContext());
        File directory = cw.getDir("" + tag_id, Context.MODE_PRIVATE);
        File file = new File(directory, "layout" + layout_id + ".xml");

        // Initialization
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
        Document doc = dBuilder.parse(file);

        Element root_element = doc.getDocumentElement();
        Log.d("XML", "Root element :" + root_element.getNodeName());

        View main_view = AndroidLayoutConstructor.getView(getActivity(), root_element)
        ScrollView sv = new ScrollView(getActivity());
        sv.addView(main_view);
        view = sv;
```

After we have loaded the XML file, we will pass the root element to our LayoutConstructor.

```
        static View getView(Context context, Element element)
        {
                String node_name = element.getNodeName();
                View view;

                if(node_name.equals("LinearLayout"))
                        view = getLinearLayout(context, element);
                else if(node_name.equals("TextView"))
                        view = getTextView(context, element);
                else if(node_name.equals("Button"))
                        view = getButton(context, element);
                else if(node_name.equals("ImageButton"))
                        view = getImageButton(context, element);
                else if(node_name.equals("ImageView"))
                        view = getImageView(context, element);
                else
                        view = new View(context);

                setViewAttr(context, view, element);

                return view;
        }
```

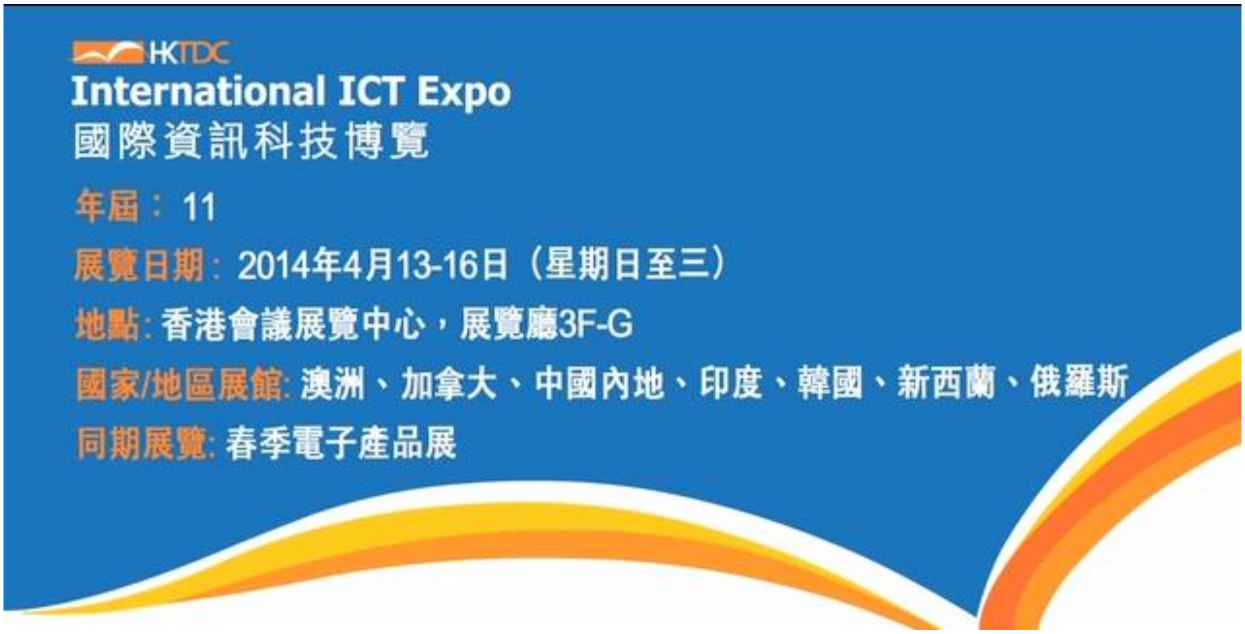
Depends on the name of the XML node, the method will create the corresponding view from context. For example, if the name is “TextView”, getTextView() will be invoked. This method will look at the attributes of “TextView” and set the corresponding attributes to the TextView.

```
    static TextView getTextView(Context context, Element text_view_element)
    {
            TextView tv = new TextView(context);

            // Handle attributes
            setTextViewAttr(context, tv, text_view_element);

            return tv;
    }
```

Some view includes child views, for example LinearLayout can contain other views including LinearLayout itself. Hence the invoking procedure should be careful. Otherwise it will be easy to stick in an infinite loop.

```
    static LinearLayout getLinearLayout(Context context, Element linear_layout_element)
    {
            LinearLayout linear_layout = new LinearLayout(context);

            // Handle attributes
            if(linear_layout_element.getAttribute("orientation").equals("vertical"))
                    linear_layout.setOrientation(LinearLayout.VERTICAL);
            else
                    linear_layout.setOrientation(LinearLayout.HORIZONTAL);


            // Add children into the linear layout
            NodeList child_elements = linear_layout_element.getChildNodes();
            Log.d("XML", "Number of children :" + child_elements.getLength());
            for(int i = 0; i < child_elements.getLength(); i++)
                    if(child_elements.item(i).getNodeType() == Node.ELEMENT_NODE)
                    linear_layout.addView(getView(context, (Element)child_elements.item(i)));
            return linear_layout;
    }
```

8 Deployment Issue



HKTDC
International ICT Expo
國際資訊科技博覽
年屆：11
展覽日期：2014年4月13-16日（星期日至三）
地點：香港會議展覽中心，展覽廳3F-G
國家/地區展館：澳洲、加拿大、中國內地、印度、韓國、新西蘭、俄羅斯
同期展覽：春季電子產品展

Every year, The Centre for Innovation and Technology (CINTEC) of The Chinese University of Hong Kong participate in the International ICT Expo organized by Hong Kong Trade Development Council. They display the R&D projects investigated by the university research teams to the public through this expo.

In previous years, the layout of their booth mainly contains numbers of poster and the LCD display. The posters are located in the bulletin board to exhibit the project description and feature technologies. Each of the projects has its own hyperlink in CINTEC Online Expo to illustrate more details. The link is also included in the poster. On the other hand, some of the projects may have a video to demonstrate how their project works. These video are shown continuously on the LCD display.

8.1 Problems in previous Expo

香港中文大學
The Chinese University of Hong Kong
THE CHINESE UNIVERSITY OF HONG KONG
THE CHINESE UNIVERSITY OF HONG KONG

We found out some problem related with this exhibiting method and needs to be improved.

1. It is difficult for the visitor to read the project description. There are some distance between the posters and visitors. The visitors need to move closer to look up the tiny word in the bulletin board. So, it will block other people from reading it.

2. Keeping the leaflets is troublesome and easy to lose it. Each of the projects has its own leaflet. When the visitors walk around, they will receive numerous of leaflet. Few of them will keep the leaflets after the Expo and they may lose the project description that they feel interesting. It leads the visitors cannot contact with the research team and ask for cooperation.

3. The exhibitors distribute their name card but the numbers of name card is limited. Besides, the visitor may not store the contact immediately and it may have a chance to lose it. So, the research team may lose some investors.

4. The visitor cannot watch their preferred video immediately. The LCD display plays all video continuously. The visitor may not wait for it and miss the video. Thus, it may lose a chance to explain the project works to visitors.

5. If CINTEC plan to distribute the souvenir, the helper in CINTEC may not remember all visitors who get the souvenir already. Therefore, the visitors may get souvenir again and again.

8.2 Suggestions

To improve these issues, CINTEC can adopt Latia to construct a serial of tags for displaying the exhibits more interactive. Each of the posters can be assigned a Tag. Then, administrators can write the information for each poster such as project name through CMS. Moreover, they can design the tag layout to fulfill their requirement. For this situation in ICT Expo, we have come up with the following suggestions:

1. Add a page for each tag to show the project description and the leaflet. The app users can read the project description from their smartphone clearly without blocking others. So, it can solve the tiny word problem in the smart poster.

2. Create a 'History' page to record the poster they have scanned. When the app users tap the record, it will show the information of that tag again. Thus, the app users can review the leaflet by referring the records in the history.

3. Add an 'Add contact' button near the project description in the layout. When the app users want to contact the investigator of that project, they can tap this button and add the contact. Hence, the contact will not be lost and also it can improve the insufficient name card problem.

4. Add a video link near the project description. When the app user is interested in the project and scan the tag, they can tap on the button and

watch the video in their smartphone rather than the LCD display. So, the app users can watch the video again even the Expo is completed.

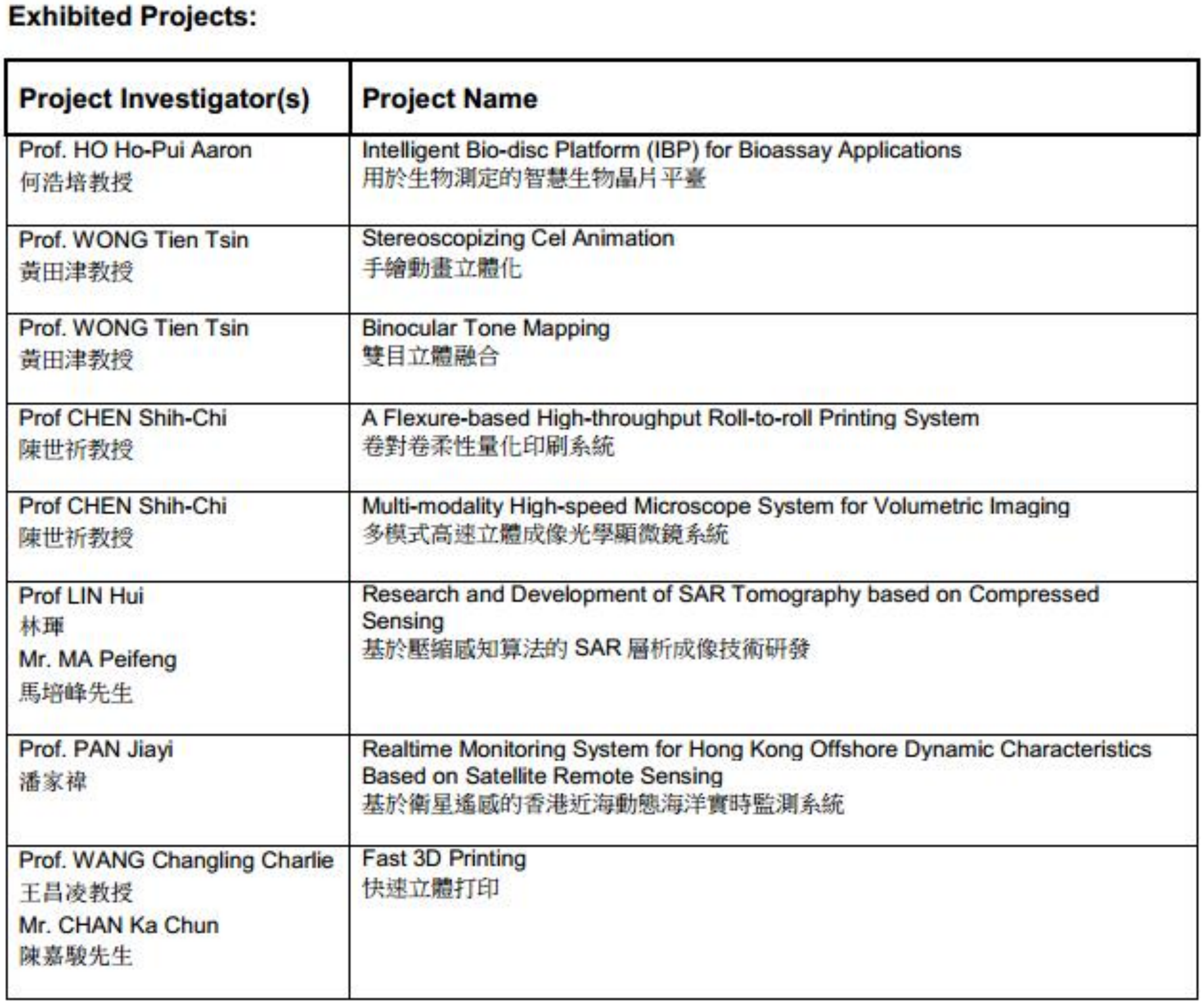
5. Assign a tag to represent receiving the souvenir. The app users need to scan this tag in order to obtain the souvenir. The ‘History’ page will record this tag. So, before the app users get the souvenir, the helper can check the ‘History’ page and prevent visitors getting the souvenir repeatedly.

We have a conversion with CINTEC to discuss about these suggestions. They accepted and approve the deployment of our apps for coordinating the R&D projects display. During this Expo, we are helping CINTEC to construct the tag and layout.

8.3 Design

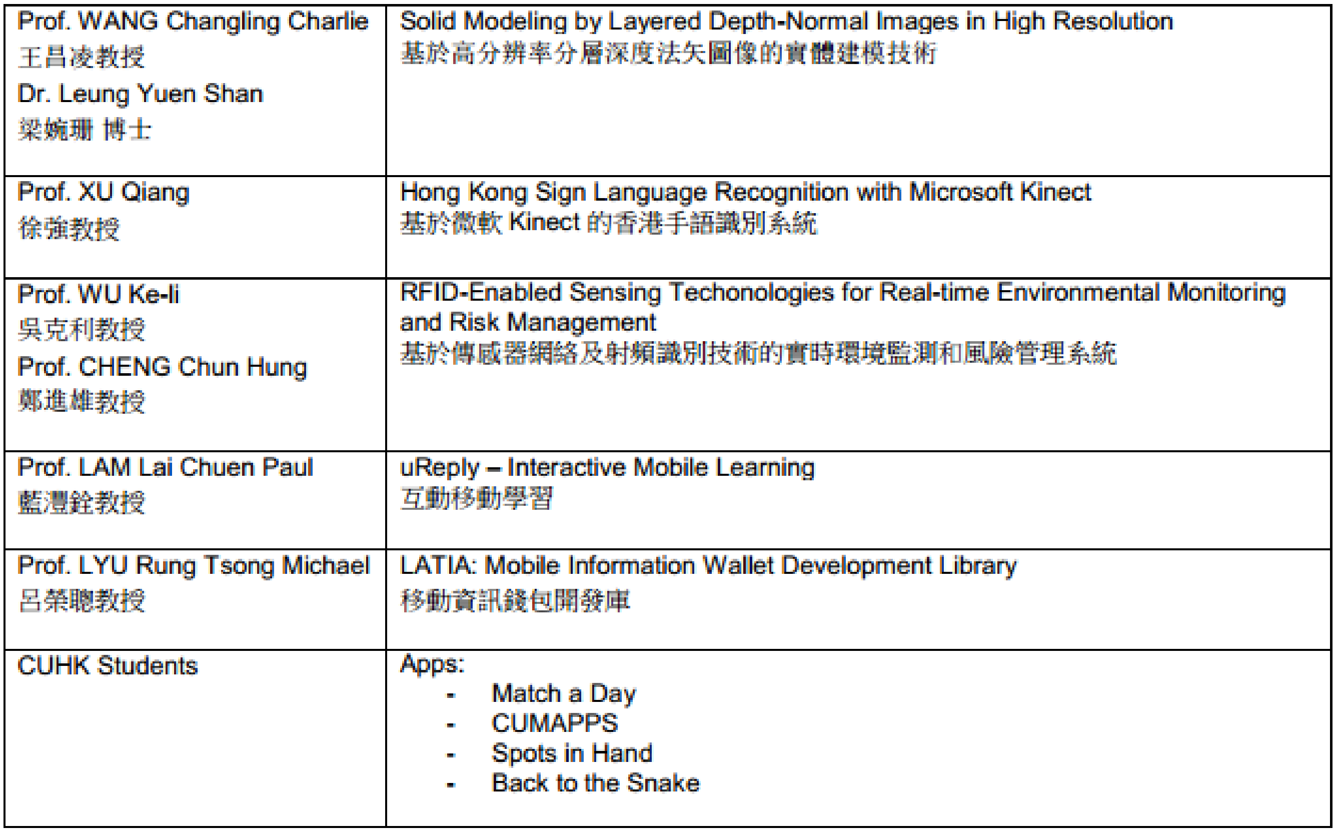
First, we need to choose the tags for the Expo. As we decided to stick the tag on the posters, the tags need to be sticky enough to stick with the poster during the Expo. Also, the memory of the tag should be big enough for writing around 100 bytes information on it. There are 17 projects to be displayed and we need to ready around 20 tags for them.

LYU1301
Mobile Application Using NFC



Exhibited Projects:

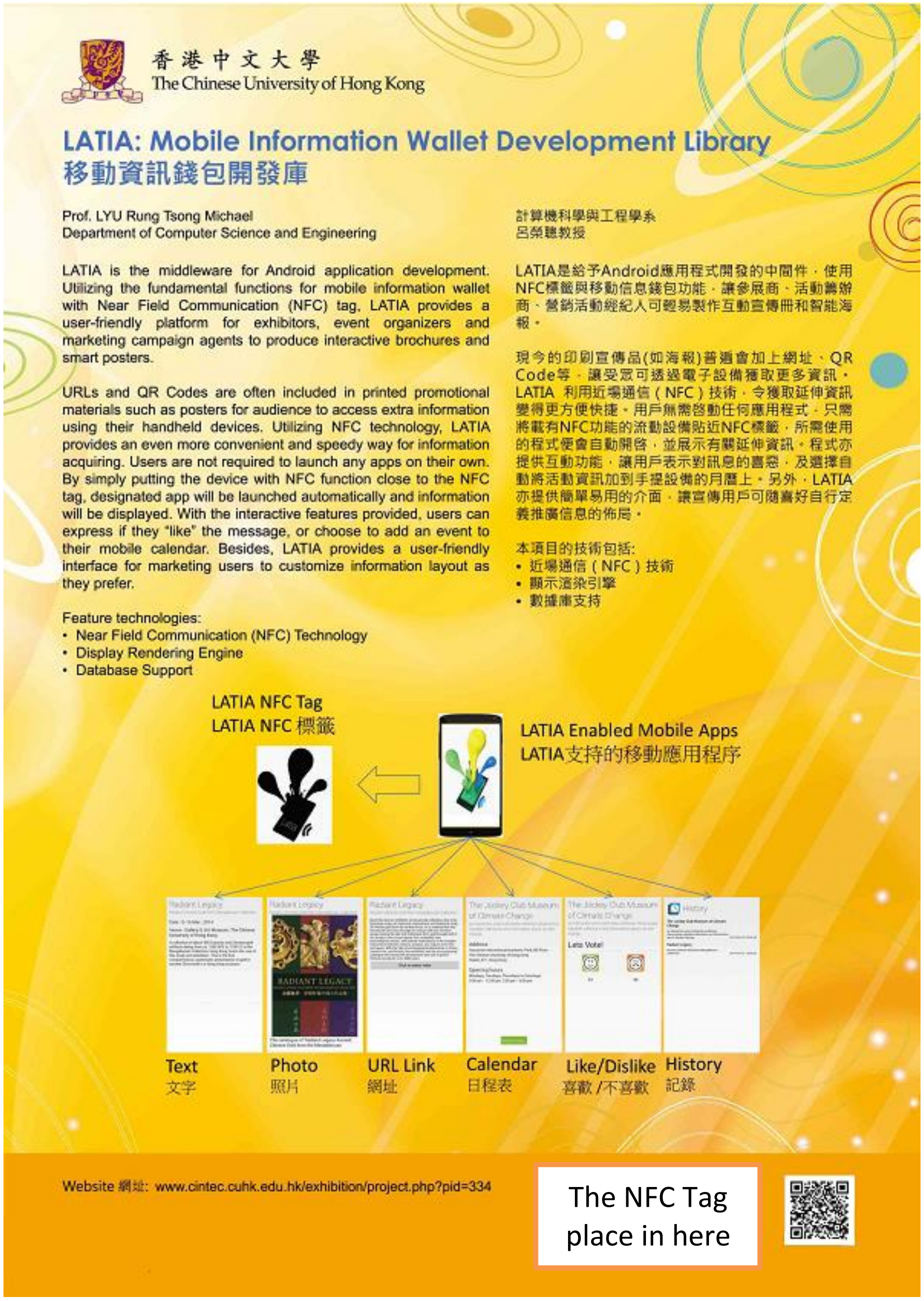
Project Investigator(s)	**Project Name**
Prof. HO Ho-Pui Aaron 何浩培教授	Intelligent Bio-disc Platform (IBP) for Bioassay Applications 用於生物測定的智慧生物晶片平臺
Prof. WONG Tien Tsin 黃田津教授	Stereoscopizing Cel Animation 手繪動畫立體化
Prof. WONG Tien Tsin 黃田津教授	Binocular Tone Mapping 雙目立體融合
Prof CHEN Shih-Chi 陳世祈教授	A Flexure-based High-throughput Roll-to-roll Printing System 卷對卷柔性量化印刷系統
Prof CHEN Shih-Chi 陳世祈教授	Multi-modality High-speed Microscope System for Volumetric Imaging 多模式高速立體成像光學顯微鏡系統
Prof LIN Hui 林琿 Mr. MA Peifeng 馬培峰先生	Research and Development of SAR Tomography based on Compressed Sensing 基於壓縮感知算法的 SAR 層析成像技術研發
Prof. PAN Jiayi 潘家褘	Realtime Monitoring System for Hong Kong Offshore Dynamic Characteristics Based on Satellite Remote Sensing 基於衛星遙感的香港近海動態海洋實時監測系統
Prof. WANG Changling Charlie 王昌凌教授 Mr. CHAN Ka Chun 陳嘉駿先生	Fast 3D Printing 快速立體打印



Prof. WANG Changling Charlie 王昌凌教授 Dr. Leung Yuen Shan 梁婉珊 博士	Solid Modeling by Layered Depth-Normal Images in High Resolution 基於高分辨率分層深度法矢圖像的實體建模技術
Prof. XU Qiang 徐強教授	Hong Kong Sign Language Recognition with Microsoft Kinect 基於微軟 Kinect 的香港手語識別系統
Prof. WU Ke-li 吳克利教授 Prof. CHENG Chun Hung 鄭進雄教授	RFID-Enabled Sensing Techonologies for Real-time Environmental Monitoring and Risk Management 基於傳感器網絡及射頻識別技術的實時環境監測和風險管理系統
Prof. LAM Lai Chuen Paul 藍澧銓教授	uReply – Interactive Mobile Learning 互動移動學習
Prof. LYU Rung Tsong Michael 呂榮聰教授	LATIA: Mobile Information Wallet Development Library 移動資訊錢包開發庫
CUHK Students	Apps: - Match a Day - CUMAPPS - Spots in Hand - Back to the Snake

For location of the tag, we decide to stick the tag near the QR code. So, the visitors can observe it easier. On top of the tag, it will have our app icon to indicate the visitor scan the tags.

LYU1301
Mobile Application Using NFC



香港中文大學
The Chinese University of Hong Kong

LATIA: Mobile Information Wallet Development Library
移動資訊錢包開發庫

Prof. LYU Rung Tsong Michael
Department of Computer Science and Engineering

LATIA is the middleware for Android application development. Utilizing the fundamental functions for mobile information wallet with Near Field Communication (NFC) tag, LATIA provides a user-friendly platform for exhibitors, event organizers and marketing campaign agents to produce interactive brochures and smart posters.

URLs and QR Codes are often included in printed promotional materials such as posters for audience to access extra information using their handheld devices. Utilizing NFC technology, LATIA provides an even more convenient and speedy way for information acquiring. Users are not required to launch any apps on their own. By simply putting the device with NFC function close to the NFC tag, designated app will be launched automatically and information will be displayed. With the interactive features provided, users can express if they "like" the message, or choose to add an event to their mobile calendar. Besides, LATIA provides a user-friendly interface for marketing users to customize information layout as they prefer.

Feature technologies:
- Near Field Communication (NFC) Technology
- Display Rendering Engine
- Database Support

計算機科學與工程學系
呂榮聰教授

LATIA是給予Android應用程式開發的中間件，使用NFC標籤與移動信息錢包功能，讓參展商、活動籌辦商、營銷活動經紀人可輕易製作互動宣傳冊和智能海報。

現今的印刷宣傳品(如海報)普遍會加上網址、QR Code等，讓受眾可透過電子設備獲取更多資訊。LATIA 利用近場通信（NFC）技術，令獲取延伸資訊變得更方便快捷。用戶無需啓動任何應用程式，只需將載有NFC功能的流動設備貼近NFC標籤，所需使用的程式便會自動開啓，並展示有關延伸資訊。程式亦提供互動功能，讓用戶表示對訊息的喜惡，及選擇自動將活動資訊加到手提設備的月曆上。另外，LATIA亦提供簡單易用的介面，讓宣傳用戶可隨喜好自行定義推廣信息的佈局。

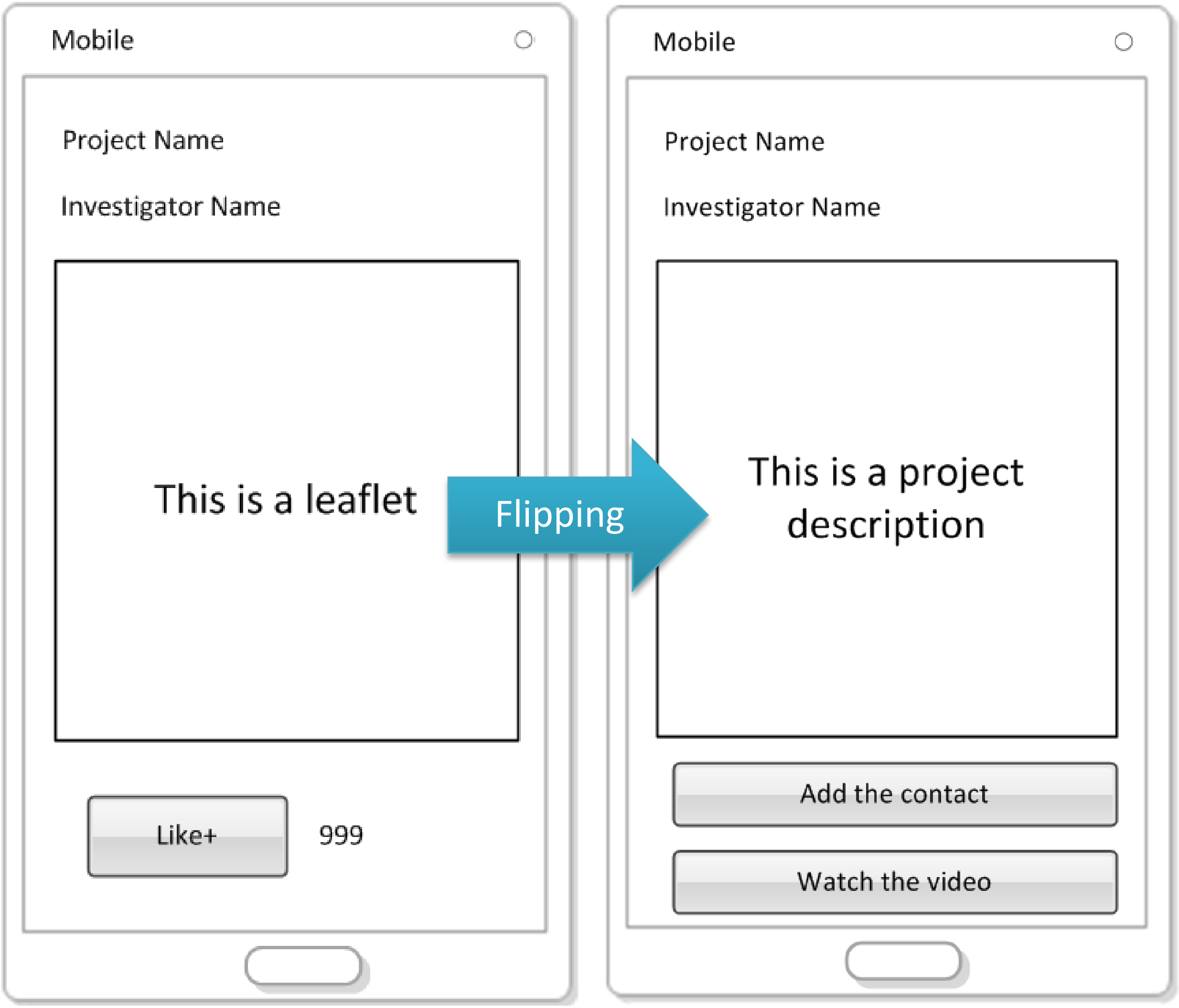
本項目的技術包括:
- 近場通信（NFC）技術
- 顯示渲染引擎
- 數據庫支持

Website 網址: www.cintec.cuhk.edu.hk/exhibition/project.php?pid=334

The NFC Tag place in here

8.3.1 Layout

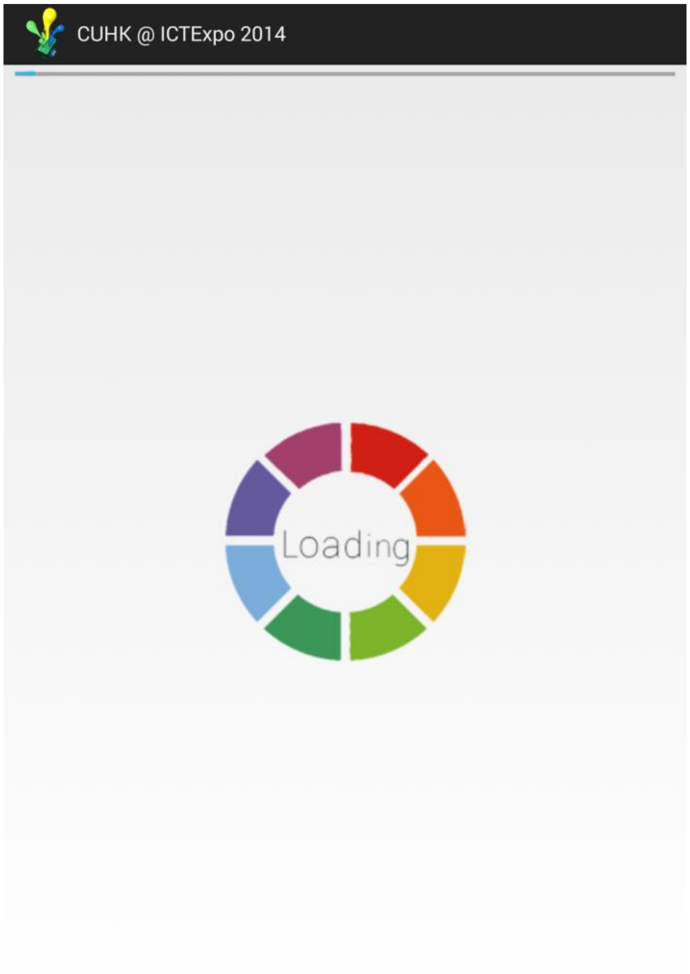
After we assigned the Tag ID to the tag, we need to create the layout for them. For the basic layout, the top are the project name and investigator name. At the bottom, there are at least two pages for each tag. The app users can flip to swipe the pages. The first page will show the leaflet about the project and a ‘likes’ button for users to express their interest. The second page will show the detail project description of the project. There will be an ‘Add the contact’ button to add the investigator contact to the contact list in the phone. Also, if there are any video to demonstrate the project, there will be a ‘Watch the video’ button for playing the video. These are only the basic layout of most of the tags. Some of the tag layout may be added some other element for a better demonstration.


Mobile
Project Name
Investigator Name
This is a leaflet
Like+
999
Flipping
Mobile
Project Name
Investigator Name
This is a project description
Add the contact
Watch the video

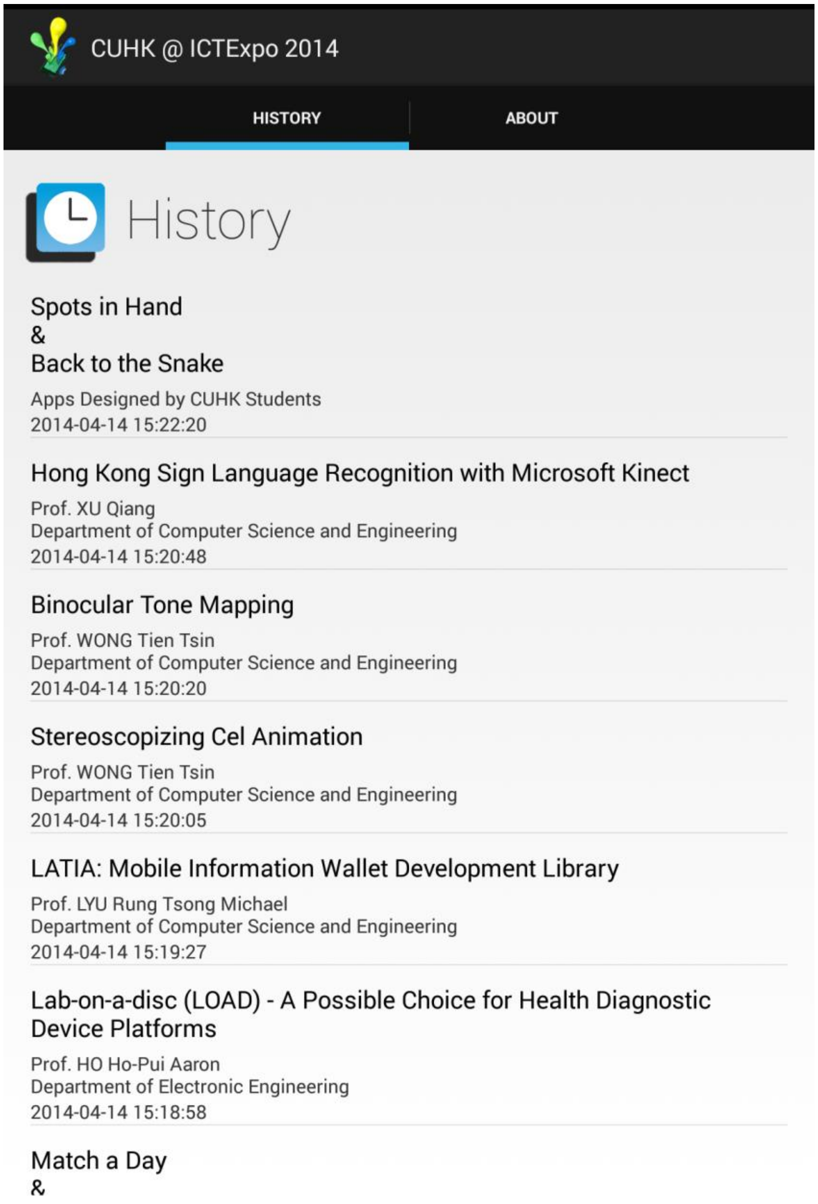
8.3.2 Implement

The following is the Loading and History page that already included in Latia. Please refer to the implementation of Android App.

Loading



History



Spots in Hand
&
Back to the Snake

Apps Designed by CUHK Students
2014-04-14 15:22:20

Hong Kong Sign Language Recognition with Microsoft Kinect

Prof. XU Qiang
Department of Computer Science and Engineering
2014-04-14 15:20:48

Binocular Tone Mapping

Prof. WONG Tien Tsin
Department of Computer Science and Engineering
2014-04-14 15:20:20

Stereoscopizing Cel Animation

Prof. WONG Tien Tsin
Department of Computer Science and Engineering
2014-04-14 15:20:05

LATIA: Mobile Information Wallet Development Library

Prof. LYU Rung Tsong Michael
Department of Computer Science and Engineering
2014-04-14 15:19:27

Lab-on-a-disc (LOAD) - A Possible Choice for Health Diagnostic Device Platforms

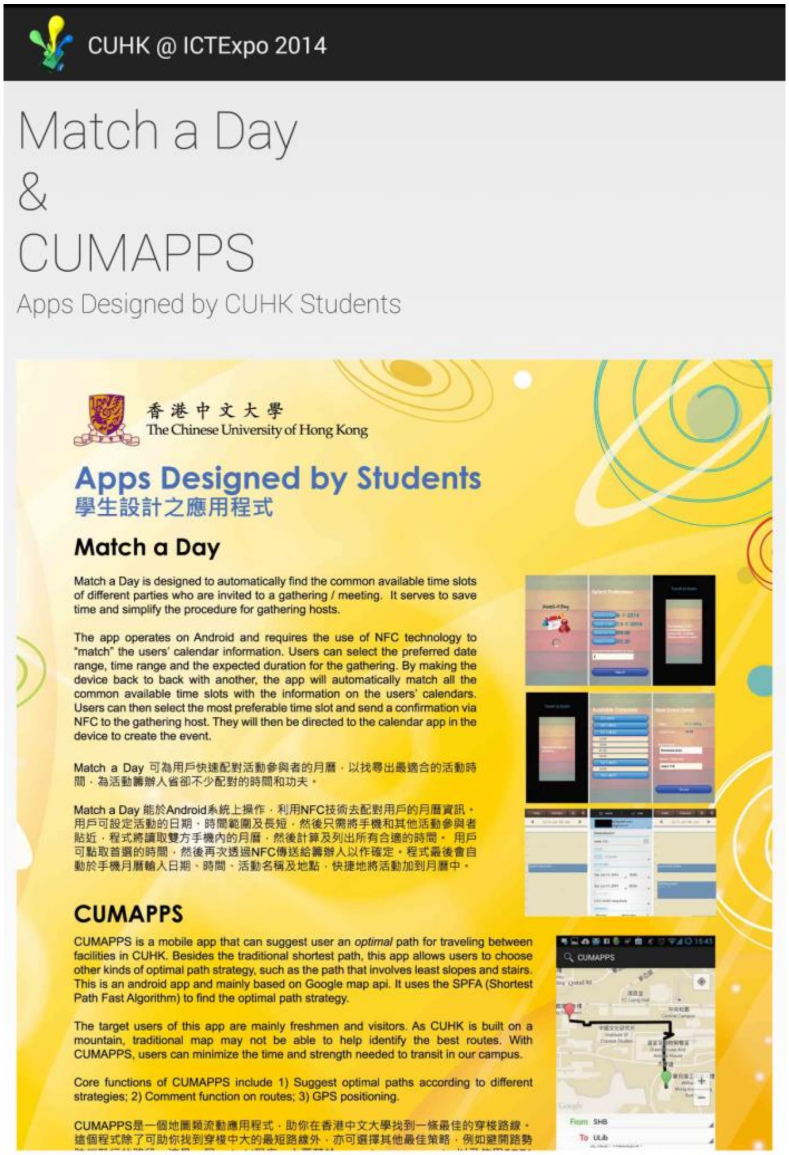
Prof. HO Ho-Pui Aaron
Department of Electronic Engineering
2014-04-14 15:18:58

Match a Day
&

Project Layout

Here is one of the project layout examples. This tag is about the project Match a Day and CUMAPPS. Besides the project name and investigator name, there are two subpages at the bottom.

This is the first subpages. At the top of it is an image view that refers to the leaflet which stored in the database. At the bottom, there is an image button to perform the like function. Near the like button is a text view that show the result of like counting. We have located a swipe image to indicate user flip to the left for next page.



Match a Day
&
CUMAPPS

Apps Designed by CUHK Students

香港中文大學
The Chinese University of Hong Kong

Apps Designed by Students
學生設計之應用程式

Match a Day

Match a Day is designed to automatically find the common available time slots of different parties who are invited to a gathering / meeting. It serves to save time and simplify the procedure for gathering hosts.

The app operates on Android and requires the use of NFC technology to "match" the users' calendar information. Users can select the preferred date range, time range and the expected duration for the gathering. By making the device back to back with another, the app will automatically match all the common available time slots with the information on the users' calendars. Users can then select the most preferable time slot and send a confirmation via NFC to the gathering host. They will then be directed to the calendar app in the device to create the event.

Match a Day 可為用戶快速配對活動參與者的月曆，以找尋出最適合的活動時間，為活動籌辦人省卻不少配對的時間和功夫。

Match a Day 能於Android系統上操作，利用NFC技術去配對用戶的月曆資訊。用戶可設定活動的日期、時間範圍及長短，然後只需將手機和其他活動參與者貼近，程式將讀取雙方手機內的月曆，然後計算及列出所有合適的時間。 用戶可點取首選的時間，然後再次透過NFC傳送給籌辦人以作確定。程式最後會自動於手機月曆輸入日期、時間、活動名稱及地點，快捷地將活動加到月曆中。

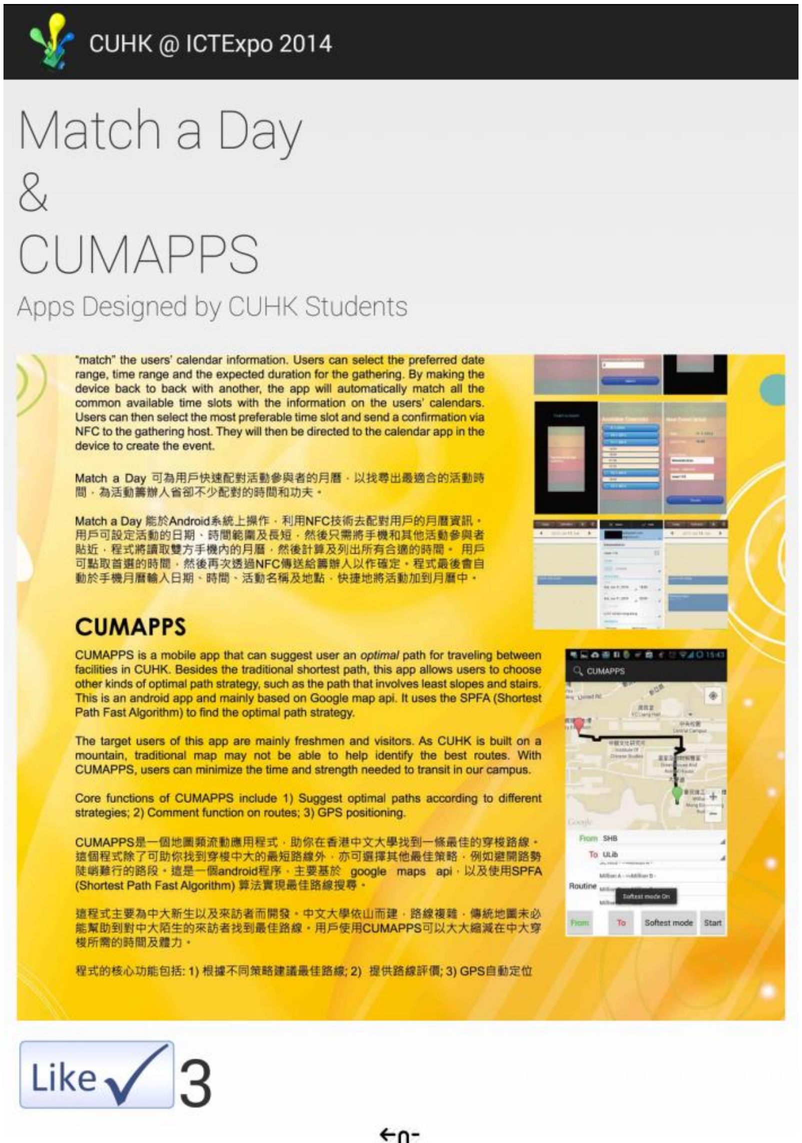
CUMAPPS

CUMAPPS is a mobile app that can suggest user an *optimal* path for traveling between facilities in CUHK. Besides the traditional shortest path, this app allows users to choose other kinds of optimal path strategy, such as the path that involves least slopes and stairs. This is an android app and mainly based on Google map api. It uses the SPFA (Shortest Path Fast Algorithm) to find the optimal path strategy.

The target users of this app are mainly freshmen and visitors. As CUHK is built on a mountain, traditional map may not be able to help identify the best routes. With CUMAPPS, users can minimize the time and strength needed to transit in our campus.

Core functions of CUMAPPS include 1) Suggest optimal paths according to different strategies; 2) Comment function on routes; 3) GPS positioning.

CUMAPPS是一個地圖類流動應用程式，助你在香港中文大學找到一條最佳的穿梭路線。這個程式除了可助你找到穿梭中大的最短路線外，亦可選擇其他最佳策略，例如避開路勢



Match a Day & CUMAPPS

Apps Designed by CUHK Students

"match" the users' calendar information. Users can select the preferred date range, time range and the expected duration for the gathering. By making the device back to back with another, the app will automatically match all the common available time slots with the information on the users' calendars. Users can then select the most preferable time slot and send a confirmation via NFC to the gathering host. They will then be directed to the calendar app in the device to create the event.

Match a Day 可為用戶快速配對活動參與者的月曆，以找尋出最適合的活動時間，為活動籌辦人省卻不少配對的時間和功夫。

Match a Day 能於Android系統上操作，利用NFC技術去配對用戶的月曆資訊。用戶可設定活動的日期、時間範圍及長短，然後只需將手機和其他活動參與者貼近，程式將讀取雙方手機內的月曆，然後計算及列出所有合適的時間。 用戶可點取首選的時間，然後再次透過NFC傳送給籌辦人以作確定。程式最後會自動於手機月曆輸入日期、時間、活動名稱及地點，快捷地將活動加到月曆中。

CUMAPPS

CUMAPPS is a mobile app that can suggest user an *optimal* path for traveling between facilities in CUHK. Besides the traditional shortest path, this app allows users to choose other kinds of optimal path strategy, such as the path that involves least slopes and stairs. This is an android app and mainly based on Google map api. It uses the SPFA (Shortest Path Fast Algorithm) to find the optimal path strategy.

The target users of this app are mainly freshmen and visitors. As CUHK is built on a mountain, traditional map may not be able to help identify the best routes. With CUMAPPS, users can minimize the time and strength needed to transit in our campus.

Core functions of CUMAPPS include 1) Suggest optimal paths according to different strategies; 2) Comment function on routes; 3) GPS positioning.

CUMAPPS是一個地圖類流動應用程式，助你在香港中文大學找到一條最佳的穿梭路線。這個程式除了可助你找到穿梭中大的最短路線外，亦可選擇其他最佳策略，例如避開路勢陡峭難行的路段。這是一個android程序，主要基於 google maps api，以及使用SPFA (Shortest Path Fast Algorithm) 算法實現最佳路線搜尋。

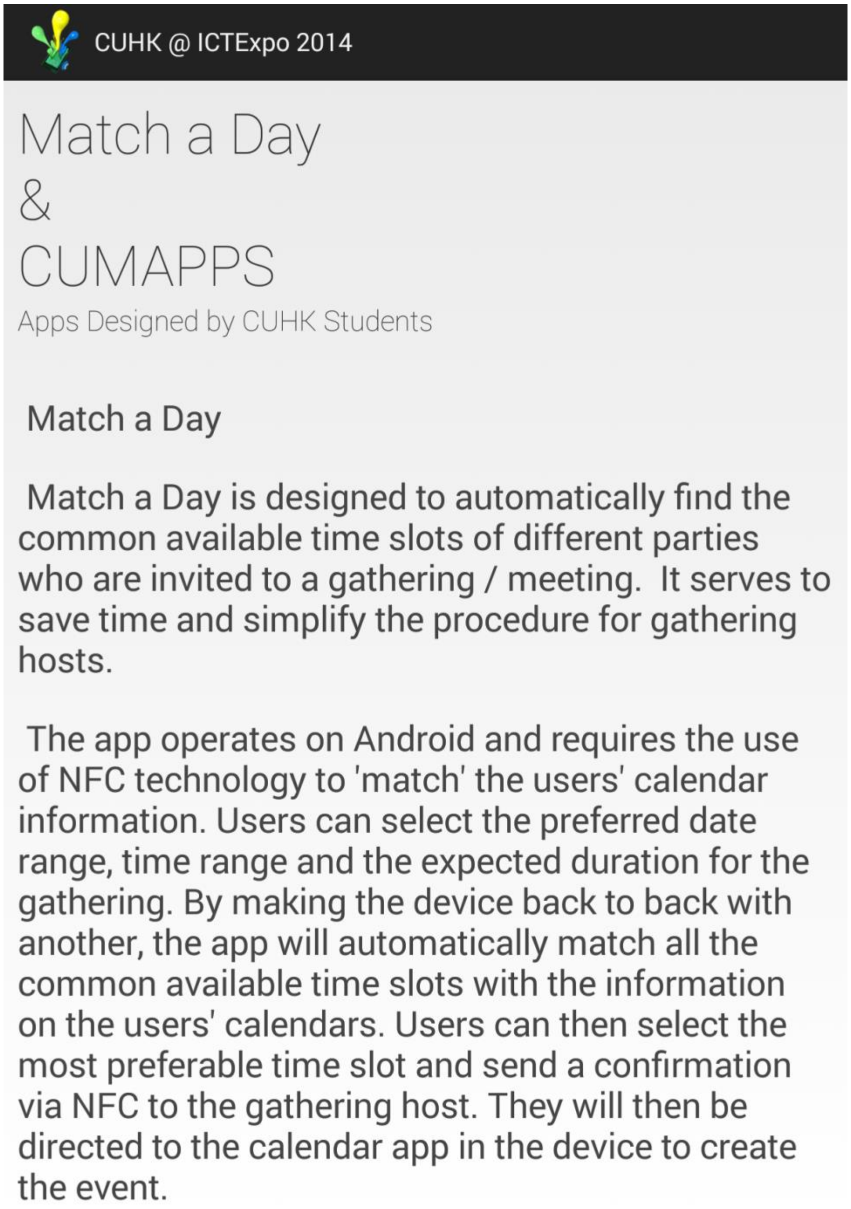
這程式主要為中大新生以及來訪者而開發。中文大學依山而建，路線複雜，傳統地圖未必能幫助到對中大陌生的來訪者找到最佳路線。用戶使用CUMAPPS可以大大縮減在中大穿梭所需的時間及體力。

程式的核心功能包括: 1) 根據不同策略建議最佳路線; 2) 提供路線評價; 3) GPS自動定位

```
<LinearLayout
  orientation="vertical"
  width="match_parent"
  height="match_parent"
>

  <ImageView
    width="wrap_content"
    height="0"
    weight="10"
    src="App1.jpg"/>

  <LinearLayout
    orientation="horizontal"
    width="wrap_content"
    height="wrap_content">

    <ImageButton
      width="0"
      height="wrap_content"
        weight="1"
        onClick="like"
      backgroundColor="0"
      imageUp="LikeButton.jpg"
      imageDown="LikeButton.jpg"/>
    <TextView
      width="fill_parent"
      height="wrap_content"
      weight="1"
      text=""
        textSize="30"
        textSrc="like_count"/>

  </LinearLayout>

    <ImageView
      width="fill_parent"
      height="wrap_content"
      src="SwipeLeft.jpg"/>

</LinearLayout>
```

The second subpage is contains some text views for displaying the paragraphs that about the project description. The users can read the details of the project. Also, the layout contain a button that link to our project video, so the users can discover more through this video.

LYU1301 Mobile Application Using NFC



CUHK @ ICTExpo 2014

Match a Day & CUMAPPS

Apps Designed by CUHK Students

Match a Day

Match a Day is designed to automatically find the common available time slots of different parties who are invited to a gathering / meeting. It serves to save time and simplify the procedure for gathering hosts.

The app operates on Android and requires the use of NFC technology to 'match' the users' calendar information. Users can select the preferred date range, time range and the expected duration for the gathering. By making the device back to back with another, the app will automatically match all the common available time slots with the information on the users' calendars. Users can then select the most preferable time slot and send a confirmation via NFC to the gathering host. They will then be directed to the calendar app in the device to create the event.

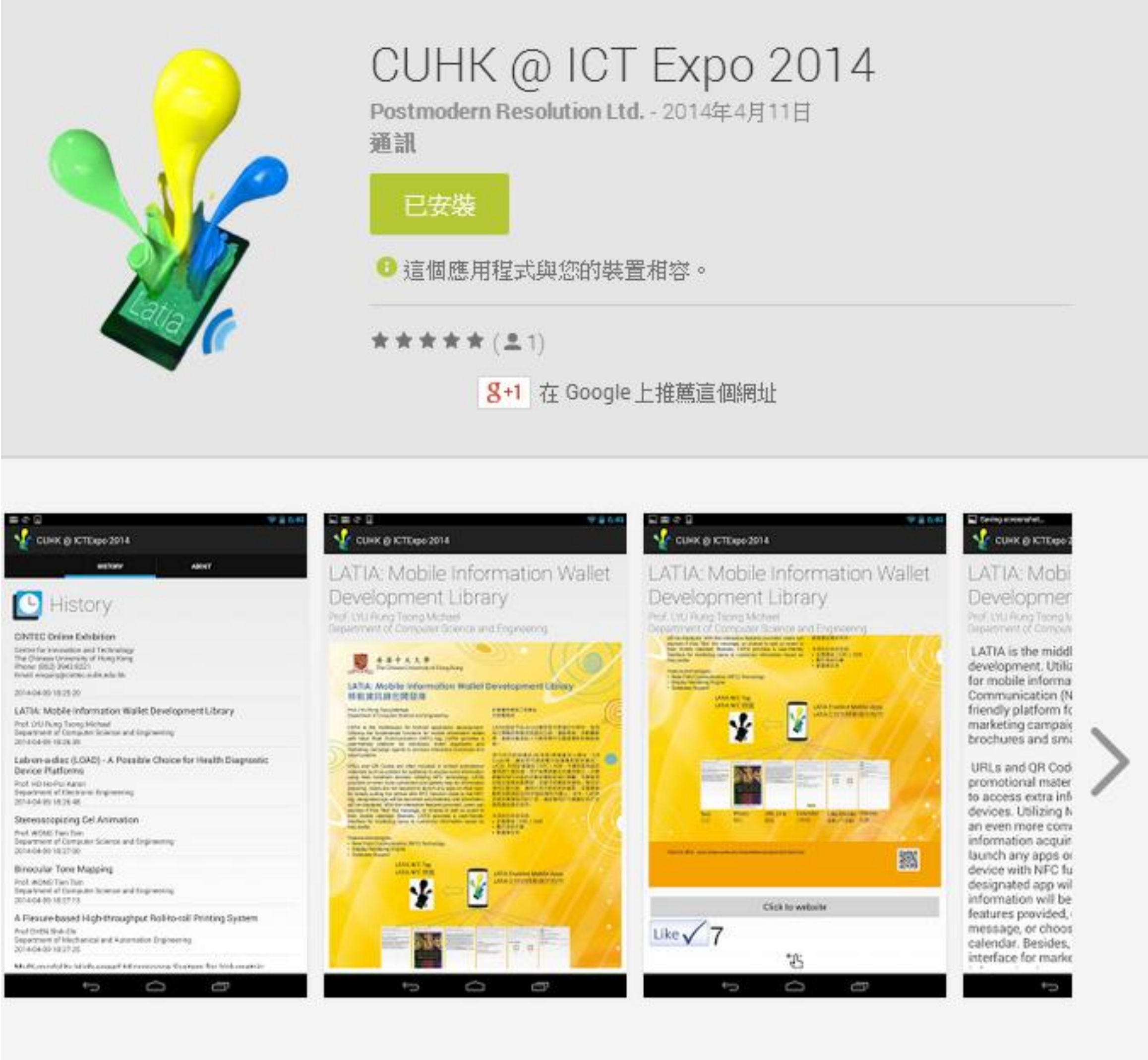
```
<LinearLayout
  orientation="vertical"
  width="match_parent"
  height="match_parent"
>

  <TextView
    width="match_parent"
    height="wrap_content"
    textColor="-12303292"
    paddingTop="30"
    paddingBottom="15"
    textSize="15"
    text="
Match a Day"/>

  <TextView
    width="wrap_content"
    height="wrap_content"
    textColor="-12303292"
    paddingTop="30"
    paddingBottom="15"
    textSize="15"
    text="
Match a Day is designed to automatically find the common available time slots of different
parties who are invited to a gathering / meeting.  It serves to save time and simplify the
procedure for gathering hosts."/>

  <TextView
    width="wrap_content"
    height="wrap_content"
    textColor="-12303292"
    paddingTop="30"
    paddingBottom="15"
```

```
            app will automatically match all the common available time slots with the
            information on the users' calendars. Users can then select the most preferable time
            slot and send a confirmation via NFC to the gathering host. They will then be directed
            to the calendar app in the device to create the event."/>
<TextView
    width="match_parent"
    height="wrap_content"
    textColor="-12303292"
    paddingTop="30"
    paddingBottom="15"
    textSize="15"
    text="CUMAPPS"/>

<TextView
    width="wrap_content"
    height="wrap_content"
    textColor="-12303292"
    paddingTop="30"
    paddingBottom="15"
    textSize="15"
    text="
            CUMAPPS is a mobile app that can suggest user an optimal path for traveling
            between facilities in CUHK. Besides the traditional shortest path, this app allows users
            to choose other kinds of optimal path strategy, such as the path that involves least
            slopes and stairs. This is an android app and mainly based on Google map api. It uses
            the SPFA (Shortest Path Fast Algorithm) to find the optimal path strategy."/>
```

```
 <TextView
   width="wrap_content"
   height="wrap_content"
   textColor="-12303292"
   paddingTop="30"
   paddingBottom="15"
   textSize="15"
   text="
       The target users of this app are mainly freshmen and visitors. As CUHK is built on a
       mountain, traditional map may not be able to help identify the best routes. With
       CUMAPPS, users can minimize the time and strength needed to transit in our
       campus."/>
 <TextView
   width="wrap_content"
   height="wrap_content"
   textColor="-12303292"
   paddingTop="30"
   paddingBottom="15"
   textSize="15"
   text="
       Core functions of CUMAPPS include 1) Suggest optimal paths according to different
       strategies; 2) Comment function on routes; 3) GPS positioning."/>
  <Button
        width="match_parent"
        height="wrap_content"
       text="Download"
       onClick="url"
        url="http://www.youtube.com/watch?v=sd35r1ey54>
</LinearLayout>
```

8.3.3 Deployment

After we implemented it, we have put the apps to the Google play store for visitor to download.



CUHK @ ICT Expo 2014

Postmodern Resolution Ltd. - 2014年4月11日

通訊

已安裝

這個應用程式與您的裝置相容。

★★★★★ (1)

g+1 在 Google 上推薦這個網址

CUHK @ ICTExpo 2014

History

CINTEC Online Exhibition

LATIA: Mobile Information Wallet Development Library

Lab-on-a-disc (LOAD) - A Possible Choice for Health Diagnostic Device Platforms

Stereoscopizing Cel Animation

Binocular Tone Mapping

A Flexure-based High-throughput Roll-to-roll Printing System

LATIA: Mobile Information Wallet Development Library

Prof. LYU Rung Tsong Michael

Department of Computer Science and Engineering

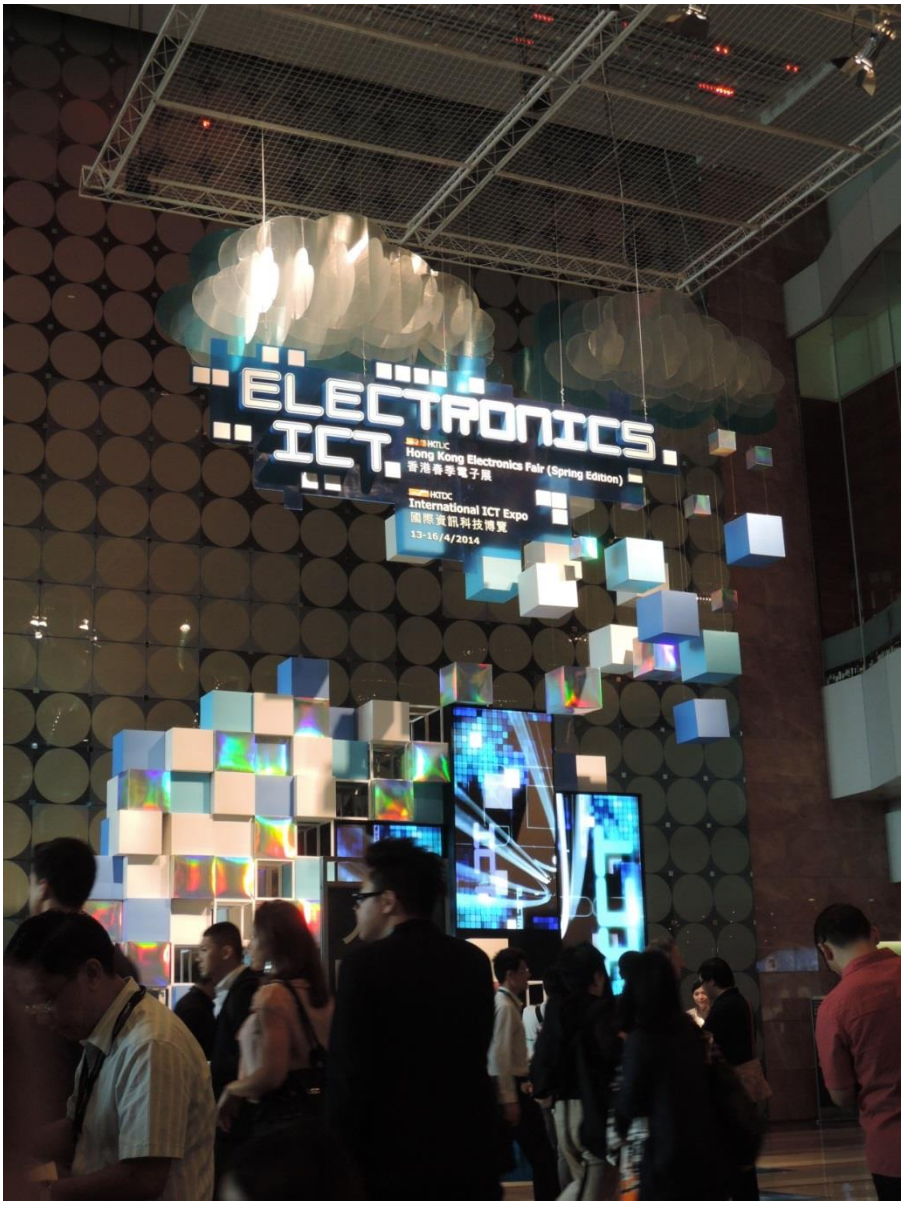
Click to website

Like 7

The following are some highlights in the ICT Expo 2014

The Centre of Innovation and Technology participated in the International ICT Expo 2014 during 13 to 16 April.

LYU1301
Mobile Application Using NFC



The Booth of The Chinese University of Hong Kong

Some visitors are interested in some project investigated by the university research team.

香港中文大學
The Chinese University
Home-grown Innovations
Lab-on-a-disc (LOAD) - A Possible Choice for Health Diagnostic Device Platforms
3G-B04, B06
G KONG ICT AW

We have placed some tag near the each of the posters.

香港中文大學
The Chinese University of Hong Kong
Apps Designed by Students
學生設計之應用程式
Match a Day
CUMAPPS
LATIA: Mobile Information Wallet Development Library
Research and Development of SAR Tomography based on Compressed Sensing

facilities in CUHK. Besides the traditional s
other kinds of optimal path strategy, such as
This is an android app and mainly based on
Path Fast Algorithm) to find the optimal path
The target users of this app are mainly fre
mountain, traditional map may not be ab
CUMAPPS, users can minimize the time and
Core functions of CUMAPPS include 1) Su
strategies; 2) Comment function on routes; 3
CUMAPPS是一個地圖類流動應用程式，助你
這個程式除了可助你找到穿梭中大的最短路線
陡峭難行的路段。這是一個android程序，主要
(Shortest Path Fast Algorithm) 算法實現最佳
這程式主要為中大新生以及來訪者而開發。中
能幫助到對中大陌生的來訪者找到最佳路線。
梭所需的時間及體力。
程式的核心功能包括: 1) 根據不同策略建議最佳
Cloud Hosting
HGC
3G-C08

The visitor can read the project description in their phone after downloaded our app and scan the tag.

Our app is also been exhibited in the International ICT Expo.

香港中文大學
The Chinese University of Hong Kong

LATIA: Mobile Information Wallet Development Library
移動資訊錢包開發庫

Prof. LYU Rung Tsong Michael
Department of Computer Science and Engineering

LATIA is the middleware for Android application development. Utilizing the fundamental functions for mobile information wallet with Near Field Communication (NFC) tag, LATIA provides a user-friendly platform for exhibitors, event organizers and marketing campaign agents to produce interactive brochures and smart posters.

URLs and QR Codes are often included in printed promotional materials such as posters for audience to access extra information using their handheld devices. Utilizing NFC technology, LATIA provides an even more convenient and speedy way for information acquiring. Users are not required to launch any apps on their own. By simply putting the device with NFC function close to the NFC tag, designated app will be launched automatically and information will be displayed. With the interactive features provided, users can express if they "like" the message, or choose to add an event to their mobile calendar. Besides, LATIA provides a user-friendly interface for marketing users to customize information layout as they prefer.

Feature technologies:
- Near Field Communication (NFC) Technology
- Display Rendering Engine
- Database Support

計算機科學與工程學系
呂榮聰教授

LATIA是給予Android應用程式開發的中間件，使用NFC標籤與移動信息錢包功能，讓參展商、活動籌辦商、營銷活動經紀人可輕易製作互動宣傳冊和智能海報。

現今的印刷宣傳品(如海報)普遍會加上網址、QR Code等，讓受眾可透過電子設備獲取更多資訊。LATIA 利用近場通信（NFC）技術，令獲取延伸資訊變得更方便快捷。用戶無需啓動任何應用程式，只需將載有NFC功能的流動設備貼近NFC標籤，所需使用的程式便會自動開啓，並展示有關延伸資訊。程式亦提供互動功能，讓用戶表示對訊息的喜惡，及選擇自動將活動資訊加到手提設備的月曆上。另外，LATIA亦提供簡單易用的介面，讓宣傳用戶可隨喜好自行定義推廣信息的佈局。

本項目的技術包括:
- 近場通信（NFC）技術
- 顯示渲染引擎
- 數據庫支持

LATIA NFC Tag
LATIA NFC 標籤

LATIA Enabled Mobile Apps
LATIA支持的移動應用程序

Text
文字

Photo
照片

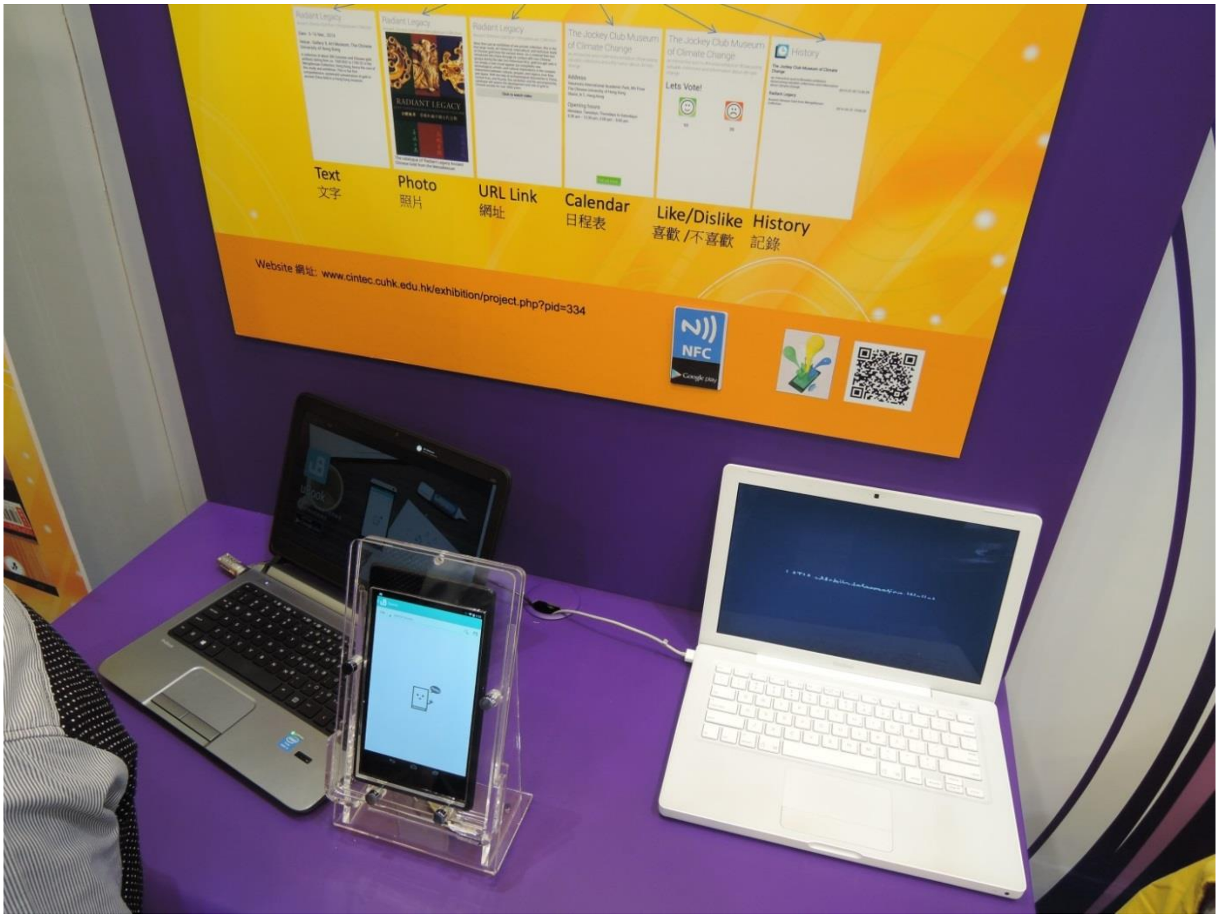
URL Link
網址

Calendar
日程表

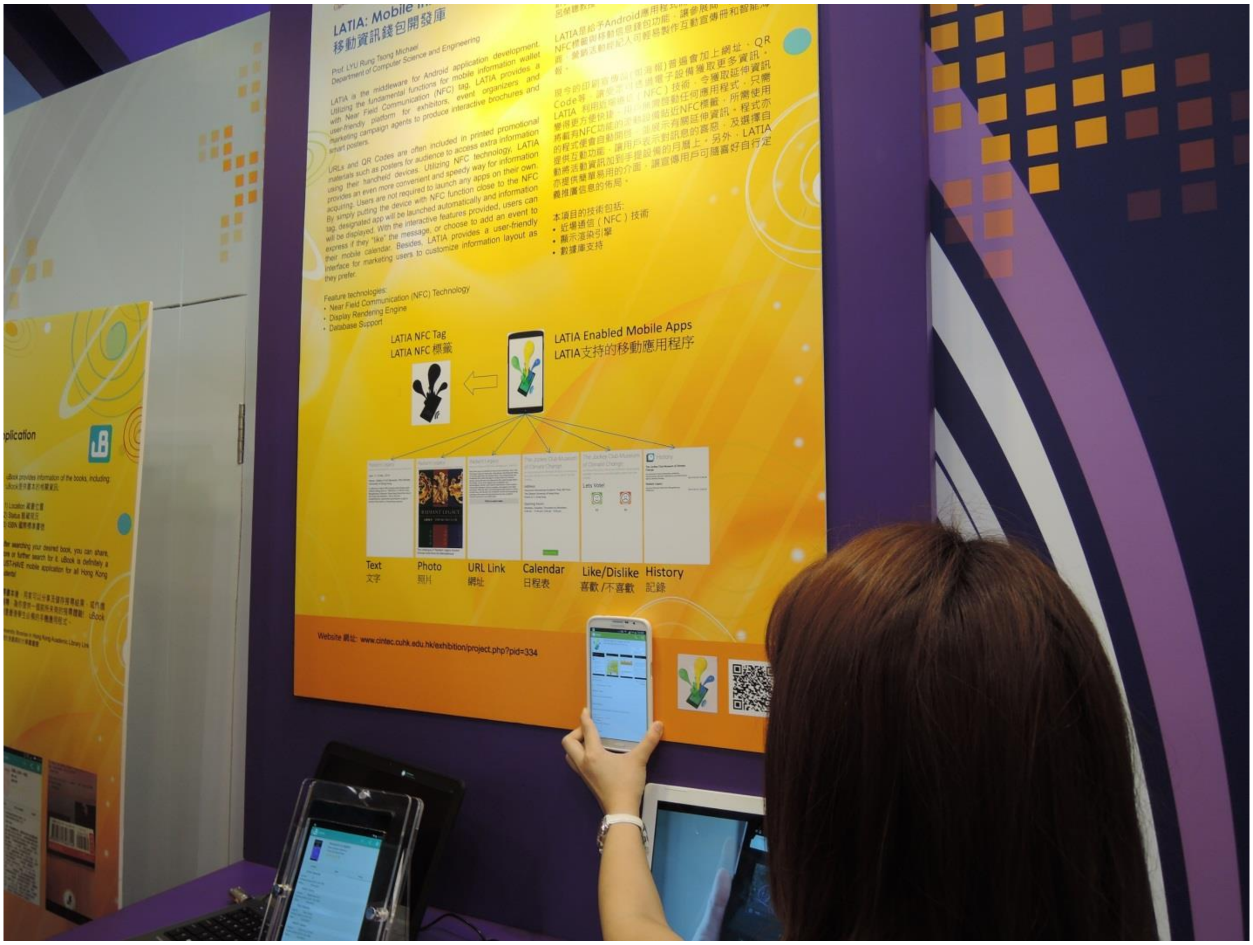
Like/Dislike
喜歡 /不喜歡

History
記錄

Website 網址: www.cintec.cuhk.edu.hk/exhibition/project.php?pid=334


Text
文字
Photo
照片
URL Link
網址
Calendar
日程表
Like/Dislike
喜歡/不喜歡
History
記錄
Website 網址: www.cintec.cuhk.edu.hk/exhibition/project.php?pid=334
NFC

Some visitors have tried our app.



The helper introduces more details to the visitors that interested in our app.

9 Limitations and Difficulties

9.1 Scalability

In order to keep friendly user experience, Latia stores all the layout XML files in the mobile device. However, the total file size should maintain within 200 MB. Total file size exceeds this number may considered unacceptable to most devices.

The overhead of Latia file size is the images. We can roughly calculate the scale of Latia by assuming all the storage space is used by images. If all images have dimension 640 x 480 and file size ~100kB, then 200 MB contains around 2000 images. If one tag has one or two images, there can only exist 1000 to 2000 Latia NFC tags. Although we can resolve this by not saving the image to the local storage, it sacrifices user friendliness.

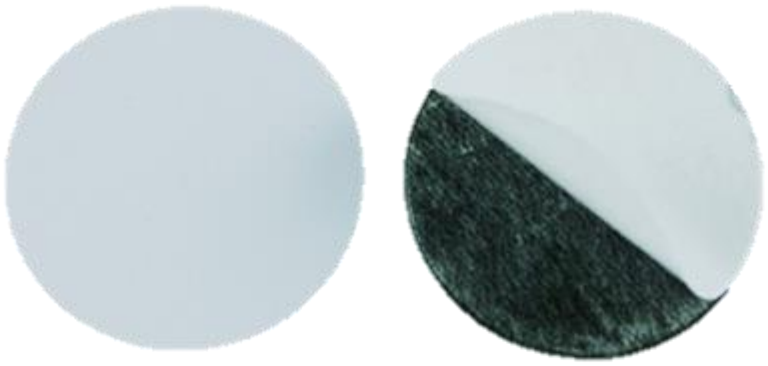
9.2 Compatibility

Latia requires NFC to trigger the layout rendering engine, hence it cannot be tested using virtual machine emulator. Fortunately, we have two devices with different screen sizes, we can test our engine on these environment. However, there are still uncertainties on other sizes.

9.3 Difficult to detect

We wish user can detect the NFC tag and attach his phone easily. Due to the generality of Latia, the NFC tag can be slicked on varies kinds of objects. It is difficult to stick on the object obviously without affecting the object itself, especially when the shape of object is complex.

The material of object and NFC tag is also a concern. Some material like metal has to stick a special NFC tag. The tag is rubber based and rigid.



Therefore, it is not possible to stick an NFC tag on a curved, metallic material.

10 Future Works

10.1 Allocation of NFC ID

Each Latia NFC tag has its own Latia ID to identify the content. Latia welcome different companies and group to make their own NFC tag, therefore the ID of the tag has to organize. We can allocate each company an identifier as the prefix. With the same prefix, they can have at most some number of tags.

The allocation method is similar to that of IP address. This keeps a good organization and easy to make changes. If possible, we can fuse the prefix to the tag with encryption, so companies can only modify their own set of NFC tags. It also improves the security between companies.

10.2 Enrich XML library

Our current XML layout does not allow user to input text and do interaction. Hence, we can add an editable text field in the XML library allowing designer to build more function. For example, commenting, answering questions and filling in form.

A button can only do five operations: like, dislike, URL redirect, add calendar and add contact. Many other useful operations can be added as well, for example background HTTP communication.

10.3 Incremental update of XML files

Each update of Latia, only necessary image is downloaded. We can improve Latia by applying it to XML file as well. So if there are so many layouts and tags, it will download redundant data.

10.4 Supporting different languages

Currently Latia only support Chinese and English. In the future, we can change the character set to Unicode, so all languages are supported. This is useful for museum to display exhibits detail in multi-linguals.

11 Conclusion

NFC is a new technology that will gradually grow to the market. It is our honored to have the chance to develop application using NFC. In this project, we have done intensive research regarding NFC. During the research, we not only learned the usage of NFC but also the fundamentals of electronics.

This is our first time to develop a mobile app in Android and we learned a lot through the process of development in our FYP. We have self-studied and put a lot of effort on studying the NFC and android SDK. From brainstorming and designing to implementation on the idea, we have experienced the convenience brought by NFC Technology. Although we encountered several problems during the development, we have found the solution for it.

In future, we will add more interesting features on Latia to make the app more attractive. We hope that finally Latia can be deployed in the Chinese University of Hong Kong and companies.

12 CONTRIBUTION OF WORK

Summer

After the FYP topic confirmed, we have a regular meeting together to find out some society phenomena and brainstorm some solution that related with Android App and NFC. Henry found that current people are too addicted with the smartphone and neglect the things around us in the real world. So, we start to design a social app with NFC that make people look around the real world.

Semester 1

As we don’t have any experience on Android App development and NFC, we faced a lot of problem in programing on Android and transmission on NFC message. We spend lots of time on studying it. Henry focuses on the NFC passing and programming Android App. Kristie focuses on the layout and the server interface using PHP. Finally, we implemented the social app for The Chinese University of Hong Kong called Cucom.

Semester 2

After the works on previous semester, we start consider the deployment issue. But Henry thinks that the concept of Cucom is limited on the Tourism purpose and we need to improve its flexibility before deployment so that the app can apply on different situation. So, we reconstruct the app structure and rename the app to Latia. Besides, we improve the customization by implement real time layout. Henry reconstructs the app structure and implements a XML parser in order to do so. Kristie reconstructs the database and implements the CMS. After we enhance the features, we found that the Centre of Innovation and Technology is going to participate the International ICT Expo this year. According to their requirement, we have design a serial of Tag to coordinate the exhibits on research project. Also, we have put the app to the Google Play Store.

Item	Wu Han Ning, Henry	Chan Sin Yee, Kristie
Background Research		
Studies on RFID	✓	
Studies on NFC	✓	
Studies on Android	✓	✓
User Experience	✓	✓
Design		
Problem Studies	✓	✓
Solution	✓	
Project Design	✓	✓
Visual Design	✓	✓

Implementation		
NFC Tag	✓	
Android	✓	
CMS		✓
Database		✓
Report Editing		
Introduction	✓	
Studies On NFC	✓	
Studies On Android NFC Package	✓	
User Experience	✓	✓
Overview	✓	✓
Design and Implement on Android App	✓	✓
Design and Implement on NFC Tag	✓	
Design and Implement on XML	✓	
Design and Implement on Server		✓

Design and Implement on CMS		✓
Deployment	✓	✓
Conclusion	✓	✓

13 Reference

[1] L. A. Perlow, Sleeping with Your Smartphone: How to Break the 24/7 Habit and Change the Way You Work, Harvard Business Review Press, 2012.

[2] RapidNFC Ltd., "NFC Tags vs QR Codes - How to Make the Right Choice," 3 May 2013. [Online]. Available: http://rapidnfc.com/blog/73/nfc_tags_vs_qr_codes_how_to_make_right_choice.

[3] Diego López-de-Ipiña,Juan Ignacio Vazquez,Iker Jamardo, "Touch Computing: Simplifying Human to Environment Interaction through NFC Technology," 2007.

[4] M. Egan, "What is NFC? How does NFC work? For what might you use NFC? - a quick guide to NFC," 9 October 2013. [Online]. Available: http://www.pcadvisor.co.uk/how-to/mobile-phone/3472879/what-is-nfc-how-nfc-works-what-it-does/.

[5] Vedat Coskun, Kerem Ok, Busra Ozdenizci, Professional NFC Application Development for Android, Wrox, 2013.

[6] "Near field communication," 18 11 2013. [Online]. Available: http://en.wikipedia.org/wiki/Near_field_communication.

[7] J. Kim, "User-Generated Content (UGC) revolution?: Critique of the promise of YouTube," 2010.

[8] R. Want, "Near Field Communication," *Pervasive Computing, IEEE ,* pp. 4-7, 2011.

[9] M. P. Marie Reveilhac, "Promising Secure Element Alternatives for NFC," 2009.

[10] V. Coskun, K. Ok and B. Ozdenizci, "A Survey on Near Field Communication (NFC)," *Wireless Pers Commun,* pp. 2259-2294, 2012.

[11] G. Madlmayr, J. Langer, C. Kantner and J. Scharinger, "NFC Devices: Security and Privacy," *The Third International Conference on Availability, Reliability and Security,* pp. 642-647, 2008.

[12] ECMA International, "Near Field Communiction Wired Interface (NFC-WI)," *ECMA International,* pp. ECMA-373, 2012.

[13] European Telecommunications Standards Institute, "Host Controller Interface (HCI)," *ETSI TS 102 622 Technical Specification,* 2008.

[14] T. Frederick O. R. Miesterfeld, N. John M. McCambridge, R. Ronald E. Fassnacht and W. a. o. M. Jerr M. Nasiadka.United State of America Patent 4739323, 1986.

[15] "Serial Peripheral Interface Bus," 9 9 2013. [Online]. Available: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus.

[16] NFC Forum, "Smart Posters," 2011.

[17] M. C. PhD and A. Nagy, "New Applications for NFC Devices," *Mobile and Wireless Communications Summit, 2007. 16th IST,* pp. 1-5, 2007.

[18] K. OK, V. COSKUN, M. N. AYDIN and B. OZDENIZCI, "Current Benefits and Future Directions of NFC Services," *International Conference on Education and Management Technology（ICEMT 2010）,* pp.

334-338, 2010.

[19] M. Roland, U. A. U. o. A. S. A. NFC Res. Lab. Hagenberg and J. Langer, "Digital Signature Records for the NFC Data Exchange Format," *Near Field Communication (NFC), 2010 Second International Workshop on,* pp. 71-76, 2010.

[20] "User experience," 8 November 2013. [Online]. Available: http://en.wikipedia.org/wiki/User_experience.

[21] "ISO_9241," 15 October 2013. [Online]. Available: http://en.wikipedia.org/wiki/ISO_9241#ISO_9241-210.

[22] S. Borsci, M. Kurosu, S. Federici and M. L. Mele, "Steps of UX: From The Expectations Of The Users Before Product Purchase To The Final Impression Of The Product," in *Computer Systems Experiences of Users with and without Disabilities*, 2013, p. 52.

[23] S. Borsci, M. Kurosu, S. Federici and M. L. Mele, "User Experience," in *Computer Systems Experiences of Users with and without Disabilities*, pp. 49-52.

[24] A. I. Wasserman, "Software Engineering Issues for Mobile Application Development," [Online]. Available: http://works.bepress.com/cgi/viewcontent.cgi?article=1003&context=tony_wasserman.

[25] M. Negulescu, J. Ruiz, Y. Li and E. Lank, "Tap, Swipe, or Move: Attentional Demands for Distracted Smartphone Input," *AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces,* pp. 173-180, 2012.

[26] "Classification," 14 November 2013. [Online]. Available: http://en.wikipedia.org/wiki/Classification.

[27] "Categorization," 2 November 2013. [Online]. Available: http://en.wikipedia.org/wiki/Categorization.

[28] J. Melzer, "Classification, Tagging & Search," 28 August 2007. [Online]. Available: http://www.slideshare.net/jamesmelzer/search-taxonomy-tagging.

[29] "Taxonomy (general)," 25 November 2013. [Online]. Available: http://en.wikipedia.org/wiki/Taxonomy_(general).

[30] S. Dumais and H. Chert, "Hierarchical classification of Web content," pp. 256-263, 2000.

[31] A. D. GORDON, "A Review of Hierarchical Classification," pp. 119-137, 1987.

[32] "A Survey of Hierarchical Classification," pp. 1-41, 2010.

[33] "Hierarchical classification," [Online]. Available: http://www.quantum3.co.za/CI%20Glossary.htm.

[34] K. JONES, "Tags vs. Hierarchy," 16 March 2009. [Online]. Available: http://engagedlearning.net/post/tags-vs-hierarchy/.

[35] E. Haselsteiner and K. Breitfuß, "Security in Near Field Communication (NFC)," *Workshop on RFID Security RFIDSec. 2006,* 2006.

[36] Atmel Corporation, "Atmel ATMEGA324A-PU Datasheet," Atmel Corporation, 2010.

[37] Olimex Ltd, "AVR-ISP-MK2 Programmer User Manual," Olimex Ltd, 2013.

[38] NXP Semiconductors, "MF1S503x Product data sheet," NXP Semiconductors, 2011.

[39] SHENZHEN TECHSTAR ELECTRONICS CO.,LTD, "TS1620-1 LCD Module Specification Ver1.0," SHENZHEN TECHSTAR ELECTRONICS CO.,LTD.

[40] International Organization for Standardization, "ISO/IEC JTC1/SC17 Cards and personal identification," International Organization for Standardization, 2007.

[41] NXP Semiconductors, "PN532 C106 Application Note," NXP Semiconductors, 2010.

[42] NXP Semiconductors, "PN532 User Manual," NXP Semiconductors, 2007.

[43] NXP Semiconductors, "PN532/C1 Near Field Communication (NFC) controller," NXP Semiconductors, 2012.

[44] NXP Semiconductors, "PN532/C1 Near Field Communication (NFC) Controller Product Data Sheet," NXP Semiconductors, 2007.

[45] EBVElektronik, "RFID Selection Guide," EBVElektronik, 2010.

[46] Motorola, Inc., "SPI Block Guide V03.06," Motorola, Inc., 2003.

[47] J. Lin and H. Shahnasser, "NFC Application Interface for Smart Phones and Appliances," *The Third International Conference on Digital Information Processing and Communications* , pp. 337-340, 2013.

[48] NFC Forum, "NFC Activity Specification," NFC Forum, 2010.

[49] NXP Semiconductors, "MIFARE Ultralight contactless single-ticket IC Product data sheet," NXP Semiconductors, 2010.

[50] NXP Semiconductors, "MF0ICU2 MIFARE Ultralight C Product short data sheet," NXP Semiconductors, 2009.

[51] NFC Forum, "Smart Poster Record Type Definition Technical Specification," NFC Forum, 2006.

[52] NFC Forum, "NFC Analog Specification," NFC Forum, 2012.

[53] NFC Forum, "Logical Link Control Protocol Technical Specification," NFC Forum, 2011.

[54] NFC Forum, "NFC Controller Interface(NCI) Specification," NFC Forum, 2012.

[55] NFC Forum, "NFC Data Exchange Format (NDEF) Technical Specification," NFC Forum, 2006.

[56] NFC Forum, "NFC Record Type Definition (RTD) Technical Specification," NFC Forum, 2006.

[57] NFC Forum, "Text Record Type Definition Technical Specification," NFC Forum, 2006.

[58] NFC Forum, "URI Record Type Definition Technical Specification," NFC Forum, 2006.

[59] NFC Forum, "Signature Record Type Definition Technical Specification," NFC Forum, 2010.

[60] NFC Forum, "Simple NDEF Exchange Protocol Technical Specification," NFC Forum, 2013.

[61] NFC Forum, "Type 1 Tag Operation Specification Technical Specification," NFC Forum, 2011.

[62] NFC Forum, "Type 2 Tag Operation Specification Technical Specification," NFC Forum, 2011.

[63] C. Nabavi, "Product Reliability," © 2006 PCE Systems Ltd, 2006.

[64] AndroidHive, "How to connect Android with PHP, MYSQL," 2 May 2012. [Online]. Available: http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/.

[65] "Draw 9-patch," [Online]. Available: http://developer.android.com/tools/help/draw9patch.html.

[66] "How to Decide Between SQLite and MySQL," [Online]. Available: http://www.sobbayi.com/blog/software-development/decide-sqlite-mysql/.

[67] "Layouts | Android Developers," [Online]. Available: http://developer.android.com/guide/topics/ui/declaring-layout.html.

[68] "ImageView.ScaleType | Android Developers," 12 April 2014. [Online]. Available: http://developer.android.com/reference/android/widget/ImageView.ScaleType.html.

[69] "CINTEC Online Exhibition - Innovation for Better Life," [Online]. Available: http://appsvr.cintec.cuhk.edu.hk/exhibition/project.php?pid=334.

14 Acknowledgement

We would like to express the deepest appreciation to our supervisor, Professor Lyu Rung Tsong Michael, for his guidance and help. He persistently gives gorgeous advice to our ideas. Without his continual help Latia would not have been possible.

Moreover, we sincerely are thankful to Professor Xu Lei and we wish to express our appreciation for his feedback on Cucom. Without his comments we will not consider deployment of our application.

In addition, we would like to thank Mr. Edward Yau and Mr. Un Tsz Lung for their help. They provide us with reliable facilities, devices and lots of ideas. Components of Latia are based on these facilities. Besides, We would like to thank CINTEC providing us a valuable chance to deploy Latia on their booth.