Final Report

Programming An Intelligent Watch

COMP4801 Final Year Project

Department of Computer Science

University of Hong Kong

Name: Zhao Haozhi Eric

UID: 3035028556

Supervisor: Prof. Francis Lau

10 Apr 2016

Summary

The project is an individual project with the topic “Programming an Intelligent Watch”. The original system of the intelligent watch had some defects in user interface design and functionality. Therefore, the objective of the project is to modify the current system, and the new modified system needs to be more user friendly, intelligent, and have more functions. The final deliverable comprises two parts. The first part is a modified system which 1) can fit well to the rounded screen, 2) has a set of new applications, including three input methods, Notepad, Airline, File Browser, Internet Browser, etc., 3) owns some intelligent functions, for example, an intelligent launcher to sort the apps on the desktop according to the usage frequency. The second part is a java program to help users design their own watch face on the computer and apply the watch face to this smart watch. The major technology used in this project is android and java programming.

Acknowledgment

I would like to express my sincere thanks to my supervisor, Prof. Francis Lau. His guidance helps me to come up with some ideas in the project, and he offers me the development kit required by the project.

Table of Contents

List of Figures:

List of Tables:

Abbreviations:

IWOP: Ingenic Watch Open Platform

UI: User Interface

MVC: Model, View and Controller

JDK: Java Development Kit

SDK: Software Development Kit

NDK: Native Development Kit

1. Background

A new android watch based on the Ingenic Watch Open Platform (IWOP) has been designed recently. The watch has excellent processor, memory and some other good hardware components. (Table 1) What is noteworthy is that the IWOP watch has a Wi-Fi component which is not found in other android wear devices except those newest published android watches. So in software aspects, the watch can achieve more functions, for example, it can be equipped with a browser and other Internet required applications.

Processor	Ingenic dual-core M200 SoC, one core up to 1.2 GHz, the other core up to 300MHz. - GPU: 3D with OpenGL ES 2.0/1.1 and OpenVG 1.1. - VPU: H.264 720P@30fpsencoding and decoding. - ISP for image pre-processing.
Memory	eMCP (4GB eMMC + 4Gb LPDDR2).
PMIC	power management IC.
Sensor	9-axis gyroscope + accelerometer + magnetometer.
USB 2.0	Micro USB device.
UART	Serial debug port.
Wi-Fi	single-band 2.4GHz IEEE 802.11b/g/n.
Bluetooth	Bluetooth 4.1 (Bluetooth Low Energy), Bluetooth 3.0, Bluetooth 2.1 + EDR.
Clocks	24MHz, 32.768kHz; 26MHz (Wi-Fi/BT).

Table 1: *The hardware components of the watch. [1]*

A hardware comparison between IWOP watch and Samsung Gear S is made to show the excellence of the hardware components of IWOP watch. Samsung Gear S is one of the newest android smart watches. And the hardware of IWOP watch is as good as that of Samsung Gear S. The memory size and the bluetooth version are the same. And for the processor and the battery, the IWOP watch is even slightly better than Gear S. (Table 2) Since it has good hardware support, IWOP watch can realize more functions compared with a vast majority of other android wear devices.

Components	Ingenic (IWOP)	Samsung Gear S
Processor	Dual-core 1.2 GHz	Dual-core 1.0 GHz
Memory	512MB RAM, 4GB ROM	512MB RAM, 4GB ROM
Bluetooth	Bluetooth Low Energy	Bluetooth Low Energy
Battery	320mAh	300mAh

Table 2: *Hardware comparison between IWOP watch and Samsung Gear S.*

As for the system, IWOP watch is currently running the IWOP android wear system. The system is a modified version of android 4.4 system. The current system has several applications, including calculator, recorder, alarm, calendar, stop watch, music player, weather forecast, and album. However, almost all those applications have poor UI designs. The shape of the watch in this project is rounded, but most of those applications were designed to fit rectangular watch face. Thus, they cannot fit well to the rounded screen, and some parts of the applications' layout cannot be shown on the rounded watch face. For instance, the keyboard of the input method in the system cannot be entirely displayed on the rounded screen, which seriously affects the normal use of the input method. Besides, the styles of the default installed applications are inconsistent. The button, font and color are not in the same style. Moreover, the system lacks some basic functions, such as some system settings and a power indicator. In a word, the original system is not so user friendly or powerful. Therefore, the purpose of this project is to modify the current IWOP android wear system. The new system will have a more user friendly design, more intelligence and more functions.

2. Objectives

The original project plan had two objectives. The first was to develop a new set of applications and a new watch face. The second was to add some drivers to the system. However, when the development kit of the watch was received, it was found that it was unnecessary to add any drivers. Therefore, the second objective was changed to modify system layout and to add system functions. In addition, a new idea was added into the project: design a program to help users develop their own watch face. Thus, the objectives of this project can be divided into three parts: application development, system customization and watch face design program. The application part covers new application development and old application redesign and redevelopment. The old applications contain all default installed applications of the IWOP system, and the new applications contain three typing input methods, a browser, a notepad, a translator, a weather application, a file browser, a wifi connections helper, an app finder, and a flight search application. The system part involves modifying system layout and adding system functions. Some parts of the system layout are modified to be more user friendly. The desktop of the system is altered to be capable of sorting the applications according to the user's usage frequency. Specifically speaking, the applications with higher usage frequency will be placed in a upper position. The third part is a java program to help users develop watch face. Users can use this program to design their own watch face on the computer and apply the watch face to the smart watch.

As for the contributions, this project will modify some parts of the system layout, contribute more applications and functions to the IWOP system to make it more powerful and easier for user to use. What is noteworthy is that this project supplies three typing input methods and a watch face design program to users, which enables users to find more fun in using this system.

3. Literature Review

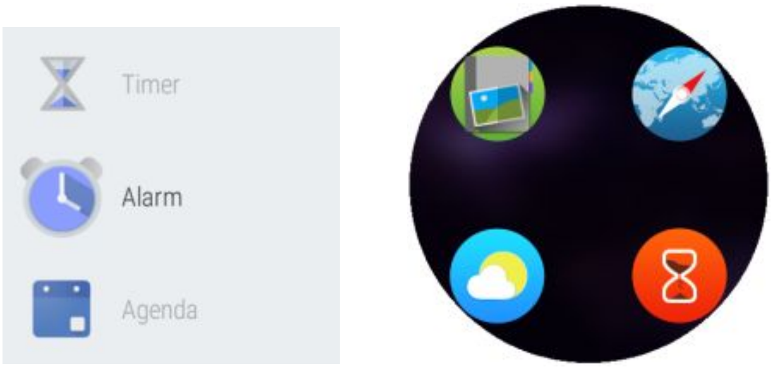
Except the IWOP android wear system, there are also some other published android wear systems. ASUS Zenwatch system is one of the most famous android wear systems, significantly different from the IWOP system in the UI design and functionality. This part will briefly introduce the Zenwatch system and the IWOP system, and make a comparison between the two systems.

The Zenwatch has a rectangular watch face. The default watch face shows current time on a colorful background. (Figure 1(a)) The background color changes as time flows. For example, the background color will change to midnight blue after 23:00 pm. The watch face of the IWOP system shows not only current time but also current date and current date in traditional Chinese calendar. (Figure 1(b)) However, the background of the IWOP watch face is just black, which seems dreary and lacks attraction. Hence, in this project, a supporting program will be developed to help users design their own watch face on the computer and apply it to the watch. To exhibit the feasibility of the program, two new watch faces will be designed in the project as a demonstration of the watch face design program.

(a) (b)

Figure 1: *The UI design of Zenwatch and IWOP. (a) The watch face of Zenwatch. (b) The watch face of IWOP watch.*

Furthermore, the two systems have quite different ways to display applications. The Zenwatch system uses a list to display all the application, as well as the names of the applications. (Figure 2 (a)) User can scroll the list up and down to view all the applications. And the icons of the applications are in the same style with gray and blue color. The IWOP system employs panel pages to display the application icons (Figure 2 (b)), but the names of the applications are not shown on the pages. There are four icons on one page. User can scroll up and down to switch pages and view applications. The list of the Zenwatch system can only display three applications in one page, while the page in the IWOP system can display four applications. But the Zenwatch system also displays application name, while the IWOP system only displays application icon, which makes the IWOP system users hard to find the applications they want. So the new system developed in this project will still use pages like the current IWOP system to display applications but with names of the applications like the Zenwatch system on the pages.



(a) (b)

Figure 2: *Ways to display applications. (a) The application list of Zenwatch. (b) The application list of IWOP watch*

As for the advantages of the Zenwatch system, it has more functions than the IWOP system, and it supplies two more settings than the IWOP system. The Zenwatch has an Agenda application, a Flashlight application and a Translator application, all of which are not found in the IWOP system. The Zenwatch allows users to change the watch face and font size. The Zenwatch system also permits users to choose a phone to connect to the watch through bluetooth setting.

As for the disadvantages, the Zenwatch system is based on the voice recognition input technology and does not have an typing input method. The accuracy of the voice input method is not high enough, and in some cases, for instance, in the noisy environment, the voice input is not the most suitable input method to input words. Current IWOP system has one typing input method, but it cannot fit well to the rounded screen. Some parts of the keyboard cannot be entirely displayed. So the system in this project will introduce three new typing input methods to help users input words.

As for the IWOP system, it does not fit well to the rounded watch face. The layouts of most default installed applications cannot be completely shown on the rounded watch face. (Figure 3) Some parts of the applications are missing from the screen and users cannot see the whole layout of the application, which may seriously affect the normal use of the application. Thus, one of the purpose of this project is to redesign those applications in order to make them better fit the rounded screen of the watch.

Figure 3: *The problem of the UI design of the IWOP watch. The Music Player application on IWOP watch.*

In a word, although the IWOP system has some basic functions, the functions of the system and the layouts of the applications in the system are not good enough to make the watch user friendly. Hence, the project is mainly aimed at the modification of the IWOP system. It will not only improve the UI design, but also add more functions and intelligence to the system to make it more powerful than the original IWOP system. To achieve the aim, the new system will assimilate some good points from other android wear systems like the Zenwatch system.

4. Approaches

Since the objective of the project has three parts, accordingly, the approaches utilized in this project can be divided into three parts. The first part is application design and development; the second part is system development and system image generation; and the third part is other approaches used in this project.

4.1 Application design and development

The first part of the approaches is application design and development. The current system already has some basic applications. However, the system and the watch are too simple and lack attraction for users. Thus, the aim of the development is to program applications with more powerful functions and more user-friendly user interface. Since users' first impression about the watch is, to a great extent, formed by the user interface, the first step to develop an android application is to design the user interface. After the step of design, the android programming technology is used to program and develop the application. The programming is operated on the Android Studio platform; and the test is performed on the development kit of the watch during the development.

The android programming is based on the MVC (model, view, controller) architecture (Figure 4), which means that the programming can be separated into three parts: model part, view part and controller part. The model part contains java classes and content provider classes, both of which are used to deal with the data computation and storage. The view part contains XML files that control the layout of the application. The controller part contains activity classes, services classes, and broadcast receiver classes, controlling the process of the application. The MVC structure works as follows: the controller part receives orders from users and passes them to the model part; the model part updates the data according to the information got from the controller part; and lastly, based on the updated data, the view part is changed to respond to the users.

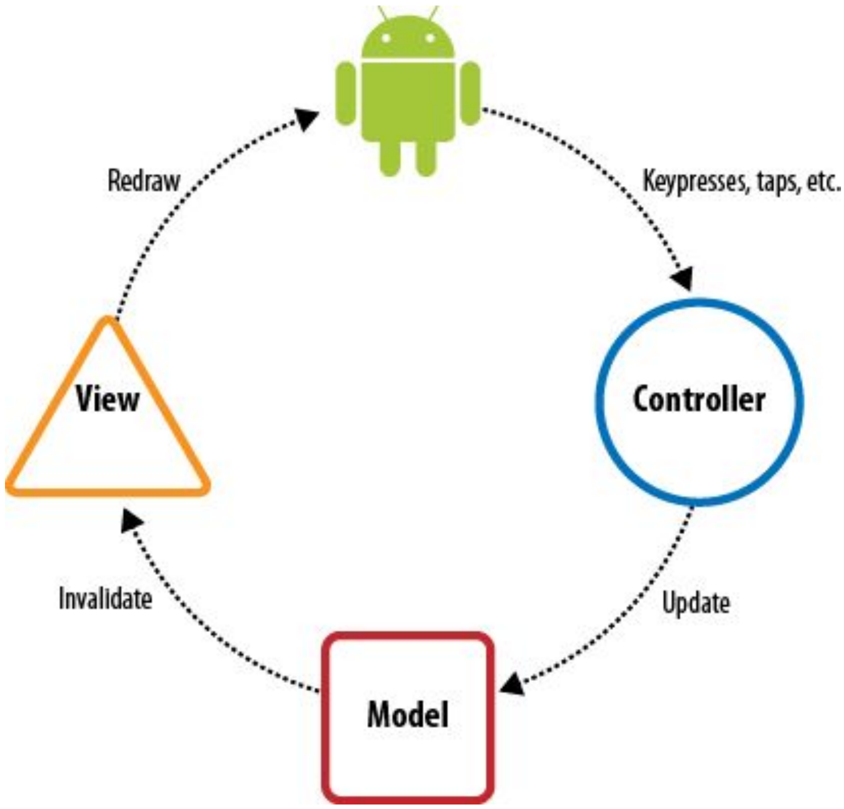


Figure 4: *The MVC structure of the Android programming.* [2]

The android programming has four main components: Activity, Service, Broadcast Receiver and Content Provider. Activity is responsible for displaying a user interface on the screen. Service runs in the background without a user interface, dealing with some background running processes, for instance, playing music. Broadcast Receiver is a component that receives broadcast announcements from the system, for example, the internet connection status, the battery low message. Content provider deals with data storage work. Applications can access the stored data through the content provider. And adding these four components needs to declare a new component in the Manifest.xml file.

Ordinary android application usually has several different activities, and starts from a MainActivity. An activity can be initiated by using Intent class. A new Intent class needs two parameters: current activity Context and new activity Class. After instantiation of a new Intent class, "startActivity" can be used to start a new activity. The activities obtain views based on the XML files, and control the foreground of the applications. Different layouts, including RelativeLayout, LinearLayout, etc., defined by the XML files, structure the view of the android application. And all controls of the view situate in different layouts. Developers can design the position of controls in the layouts by modifying the XML files. And activities can find the controls by using "findViewById".

Some applications need data storage. There are three common methods to store the data. The first approach is to write the data into files and store the data in the file system. The second approach is to store the data in a local database. Android system supplies a SQLite database for developers to store the data and as well as some useful API in the SQLite Helper classes. The third approach is to use a key-value set to store the data. Android system supplies a "SharedPreferences" API which can help store the data in the form of key and value. But one of the problems to use the third approach is that application errors will cause data lost.

Besides the ordinary android programming, this project also covers input methods development. Three input methods are developed in this project. The first one modifies the keyboard of the original input method and enables the whole keyboard to be displayed on the rounded screen. To achieve the goal, the XML files of the original keyboard are modified. The second and third one are new input methods developed and added to the system, employing the android input method service.

To develop a new input method, the first step is to declare a service with the permission of "BIND_INPUT_METHOD". Then, implement an InputView and build a new class that extends KeyboardView. The KeyboardView helps to construct the keyboard view based on XML files and render the keyboard layout. And the listener of action to click and touch needs to be implemented in the InputView.

The second input method in this project uses long press and short press to distinguish two characters on one button. The InputView records the system time when the user presses the button, and records the system time again when the user releases the button. According to different time length, the input method can distinguish between long press and short press, and further distinguish the two characters on the same button.

The third input method in this project uses slide direction to distinguish characters on one button. In this input method, the InputView implements the onTouchListerner. The onTounchListener can get the coordinate of a user's press action. So this input method will record the first coordinate and the last coordinate of the press. According to the two coordinates, the slide direction can be calculated and point to the character the user wants to input.

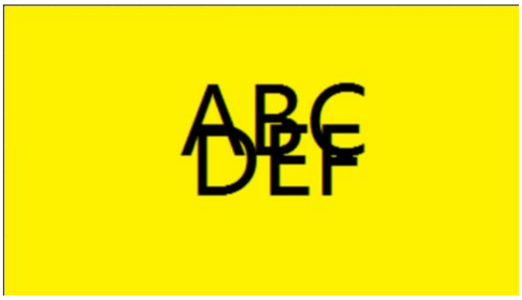
4.2 System development and system image generation

The second part is system development and system image generation. The new system is developed based on the original IWOP system, and the development of the system is performed on the Linux environment. The first step is to get the source code of the system, starting with downloading the *repo* folder from the IWOP website. The *repo* folder only provides the way to download the system code but does not contain the source code. Then the Repo and Git commands are utilized to download the whole source code of the project from the Git repository. After the downloading, the Java environment is required to compile the system because the IWOP system is based on the Android kernel. Therefore, the next step is to set up the Java environment, including the JDK, SDK and NDK. Since the lower architecture of the Android kernel is the Linux Kernel and there are no hardware components to be added to the watch, it has no need to modify the Linux Kernel. Hence, most of the work on the system are performed on the Android part. The work on the system generally has two parts. The first part is to add the developed applications to the system, and the second part is to modify the system appearance and system settings. To add the developed applications, the source codes of the applications are required to be moved into the *elf* folder in the source code file folder. The android source code in the */system/apps* folder will be compiled into apk files when the system is compiled. And those applications will be installed in the system as default applications. To modify the system appearance, the *frameworks* folder under the source code folder has to be modified, inasmuch as the source code in the *frameworks* folder controls the font, background, layout, and the startup screen of the system. To change the system settings, the source code in the */system/app/AmazingSettings* folder will be modified as well.

4.3 Watch Face Design Program Approaches

The project also contains a java program with the purpose of helping users design their own watch face. This program mainly uses the java programming technology. In fact, this program, at the beginning, was developed by C#. However, the controls in C#, such as label and button, cannot display real transparent background. When the controls in C# are set to display transparent background, the controls will display the color or background of their parent controls. (Figure 5 (a)) As a result, the preview of a watch face displayed on the computer is not like that of a real watch. Java swing does not have this problem. Controls in the Java swing can display a real transparent background (Figure 5 (b)), so Java swing can create a more real watch face for users to preview on the computer. Thus, this program utilizes java swing programming instead of C#.

(a)



(b)

Figure 5: *Comparison of transparent background appearance between (a) C# (b) Java.*

The watch face folder contains several files. The XML file, required for the implementation of the watch face, defines the layout and attributes of the watch face. Other files include background pictures, font files and clock hand picture. When the program is started, it first displays a user interface containing several text fields for the user to input his or her requirements for the watch face. After clicking the "Preview" button, this program will display a watch face preview to the user. After clicking the "Generate" button, this program will generate a new watch face folder, and build a XML file according to the information provided by the user as well as the watch face rules of the IWOP system. After the user puts the watch face folder into the "clocks" folder of the watch, system launcher will look into the folder and get the XML file, and then the watch will display the watch face according to the attributes and elements included in the XML file.

5. Results

The results of this project can be divided into three parts. The first part is a new set of default installed applications for the system. The second part is some modified or new system applications. The third part is a java program to help users design a watch face.

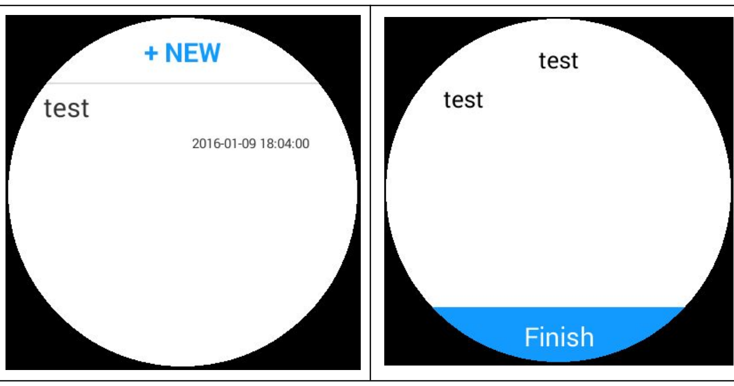
5.1 Default Installed Apps

The first part is a new set of default installed applications. Some new applications are developed and added to the system as default installed applications, and some original default installed applications are successfully redeveloped. Compared with those old applications, the current applications can better fit the rounded watch face.

New applications include Airline Search, Notepad, File Browser, Web Browser, Translator, and App Finder. Airline Search assists users to find available flights. Notepad helps users to take notes and store information. File Browser aids users to browse the files in the watch. Web browser enables users to surf the internet. Translator can translate words between Chinese and English. App Finder is an application to help users find the applications in the watch. Since the watch face is so small that only four applications can be displayed in one page on the desktop, sometimes it will be inconvenient for users to find the application they need. So the App Finder application is developed for users to find applications. Users can open App Finder and input the application name either by keyboard or by voice. Baidu voice recognition interface is used for the voice recognition. After users input the application name and press the “Go” button, App Finder will directly open the application they need.

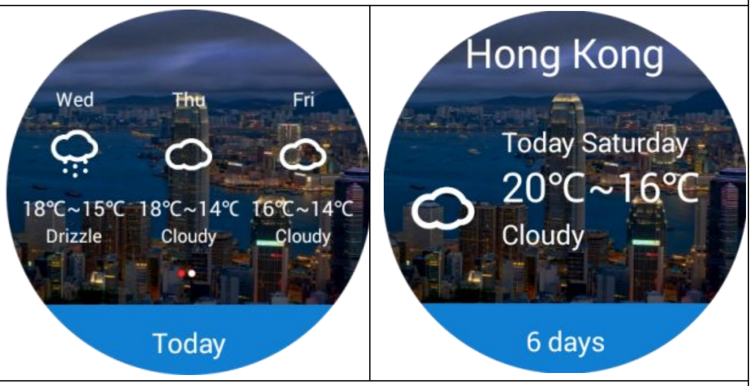
The redeveloped default installed applications include Calculator, Music Player, Timer, Phone, CountDown, Calendar, Alarm and Weather applications. Those applications had poor UI designs, and some parts of the applications cannot be shown on the rounded screen of the watch. This defect brings inconvenience and discomfort to users when they use the applications and the watch. Therefore, the layouts of those applications are changed to better fit the round screen. And the styles of those applications are also changed to be consistent. The redeveloped default installed applications, together with the new default installed applications, supply more convenience and functions for the watch users. (Table 3)

(a) Web Browser

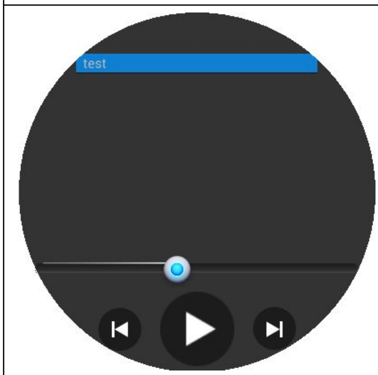


(b) Notepad

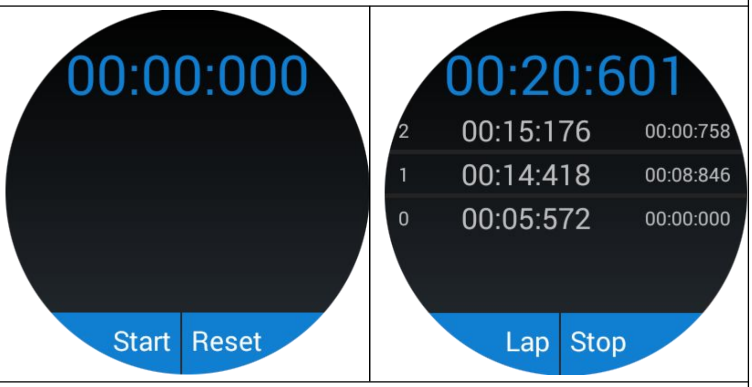
(c) Translator



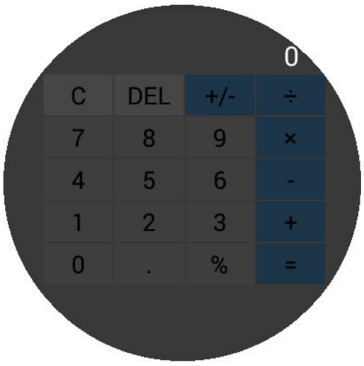
(d) Weather



(e) Music Player



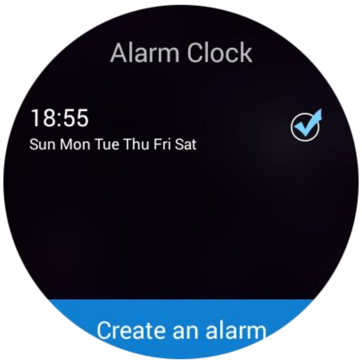
(f) Timer

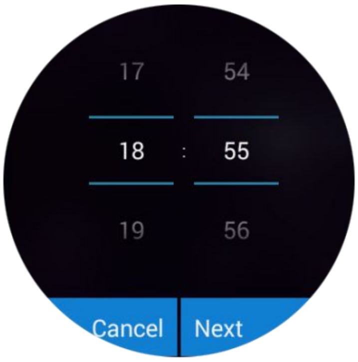


(g) Calculator

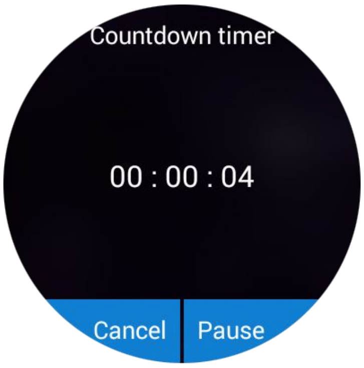
(h) Phone

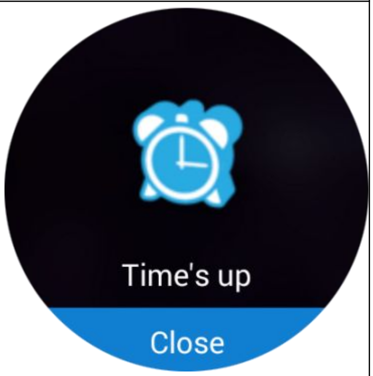
(i) Flight Search





(j) Alarm





(k) Countdown

(l) Calender

(m) App Finder

Table 3: *Default Installed Applications*

5.2 System Applications

The second part is some modified and new system applications, mainly including Launcher, Input methods, Wifi Connections and Input Method Picker.

The original Launcher generates a desktop displaying four application icons without names on one page. However, the absence of the application names makes users feel quite inconvenient to find the applications they need. So the Launcher is changed to display not only the icon of the application but also the name of the application. Another new feature of the Launcher is the sorting function. The Launcher can sort the applications on the desktop according to the usage frequency. The applications with higher usage frequency will be sorted to a prior position on the desktop. The more times the user uses an application, the more preceding place the application will stay. So users can find the applications they may need most in a short time. Nonetheless, the application of App Finder is an exception. It is permanently put at the first place on the desktop, because users may need to use it to find other applications. In addition, a power indicator and a wifi state icon are added into the system layout. Users can pull down the watch face to see the power indicator and watch status, such as wifi status and bluetooth status.

In the original objectives, three settings would be added to the system. They are wifi settng, bluetooth setting and input method setting. Wifi setting enables the watch to connect to the internet. Bluetooth setting makes the watch show the bluetooth devices near the watch, and users can choose a device to connect by bluetooth. Another setting is input method setting. If there are more than one input method in the watch, users may need to choose one to be the default input method. However, due to the limited time for the development, only two settings are added to the system: Wifi Connection and Input Method Picker. The watch has wifi hardware component but the old system does not support wifi connection. So a Wifi Connection application is developed and added to help the system connect to the internet. (Figure 6) And the original system only has one input method but the new system contains three input methods. So users may need to set the default input method. The Input Method Picker application is used to help users set the default input method. (Figure 7)

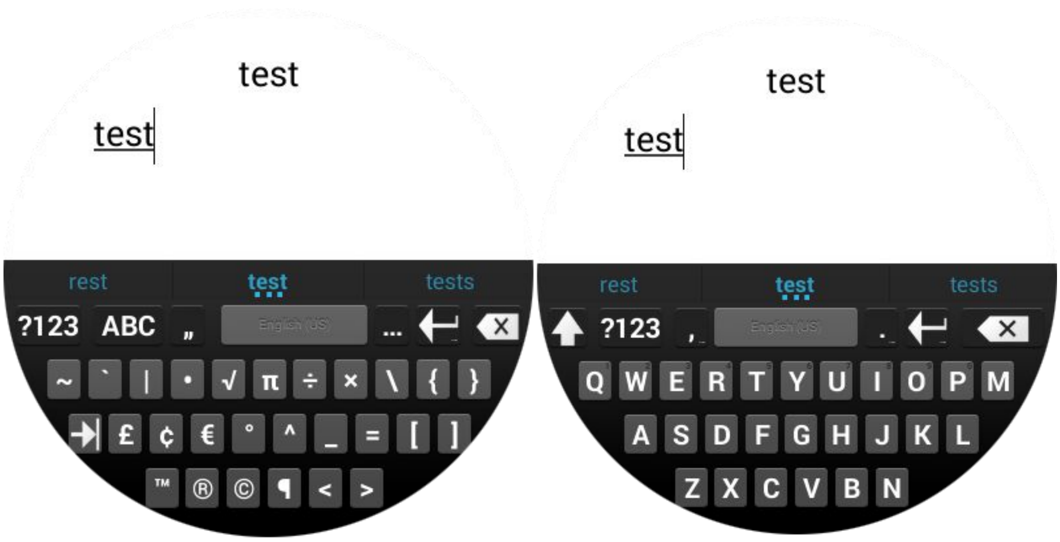
Figure 6: *Wifi Connection*

Figure 7: *Input Method Picker*

Three input methods are developed for the watch system. The old system has an input method of the android phone system. The problem of that input method lies in the size of the keyboard. The keyboard of the old input method nearly covers the whole screen of the watch, and some parts of the keyboard cannot be shown on the screen. (Figure 8) Compared with the old input method, the new input methods adjust the keyboard and have a complete layout.

Figure 8: *Old input method*

The first input method modifies the keyboard layout of the old input method, so that it can display the whole keyboard on the screen. (Figure 9) But the problem is that those buttons on the keyboard are very small, which makes the input method hard to use. Although the keyboard can be entirely displayed, it is still uneasy for users to press the right button. Therefore, another two new input methods are developed to solve this problem.



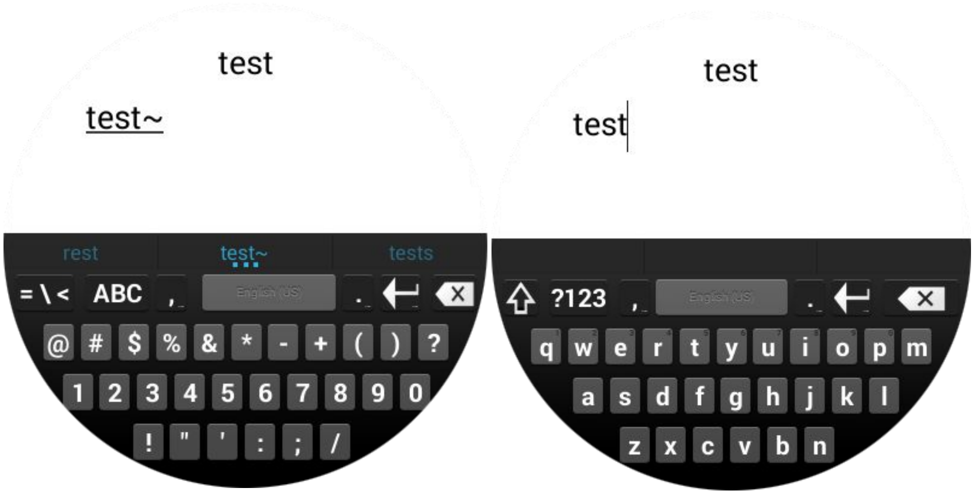


Figure 9: *First new input method*

The second input method combines every two adjacent keys into one key, and short press outputs the left character on the key while long press outputs the right character on the key. (Figure 10) For example, on the original keyboard, ‘Q’ and ‘W’ are two keys, but in the new input method, there is only one key for ‘Q’ and ‘W’. When users put a short press on the key “QW”, ‘Q’ will be output; when users put a long press on the key “QW”, ‘W’ will be output. The original keyboard contains nearly ten buttons in one row, while the new input method only has about five buttons in one row. So the size of the buttons in this input method is larger than the buttons on the original keyboard, which makes this new input method much easier for users to use.

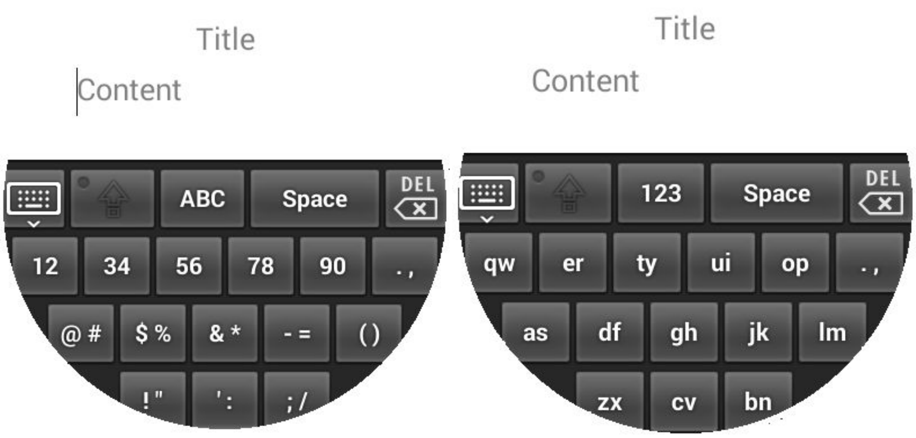
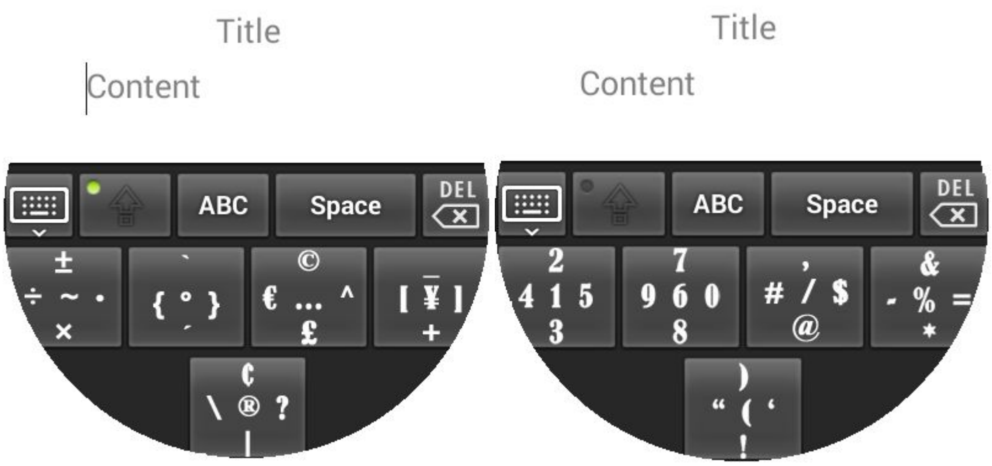


Figure 10: *Second new input method*

The third input method places five characters on one button. (Figure 11) This input method utilizes both press action and slide action from users. When users press the button, the character in the middle of the button will be output. And when users slide the button to the top direction, the character on the top will be output; when users slide the button to the bottom direction, the character on the bottom will be output; when users slide the button to the left direction, the character on the left will be output; when users slide the button to the right direction, the character on the right will be output. This kind of method shrinks the the major part of the keyboard to only five or six large buttons, which to great extent reduces the occurrence of typo and makes users feel more comfortable when they are using the input method.



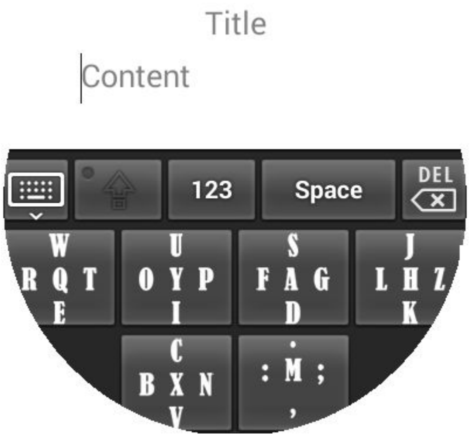


Figure 11: *Third new input method*

To sum up, all the three new input methods correct the keyboard problem of the original input method, and have a complete keyboard layout. And the keyboards of the second and the third input method have larger buttons, thus the last two input methods can effectively reduce the occurrence of typo and make users feel more comfortable when they are inputting words on the watch.

5.3 Watch Face Design Program

In this project, a watch face design program is developed to help users design their own watch face. The system has several default watch faces, but for users those watch faces may look boring and lack attraction. And in some cases designers of the system may also need to make some new watch faces and add them to the system. For instance, some companies or groups may place large orders and require for customization, so system designers have to make some new watch faces with their logos or other features according to their specific requirements. But generating a watch face requires sufficient knowledge about the XML files and rules of the watch face. Now with this watch face design program, users and designers who know nothing about the XML files or watch face rules can also generate their preferred watch faces easily and quickly only with some pictures.

This program utilizes java swing programming instead of C# programming, because java swing performs better on the preview function of this program as stated before. And the following parts are the description about how to use the program.

When the program is opened, a dialog will pop out and let users input the name of the new watch face and choose the type of the watch face. There are two types of watch face for users to choose: “analog” and “digital”. The analog type will help users make a watch face with hour hand, minute hand and second hand, just like a real watch, while the digital type will help users design a digital watch face, using digits to show the time. (Figure 12)

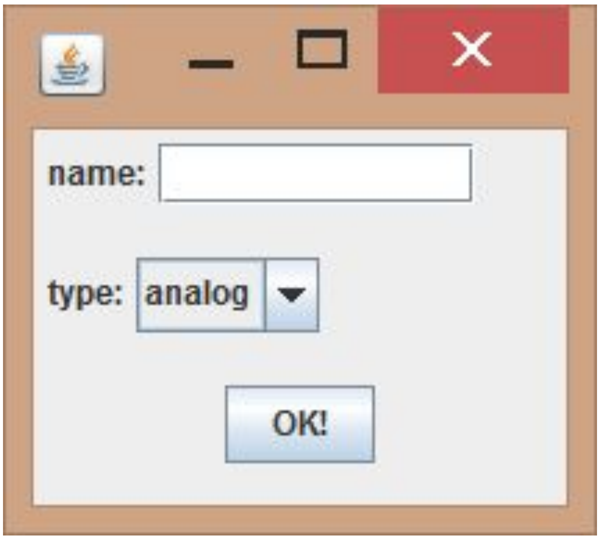
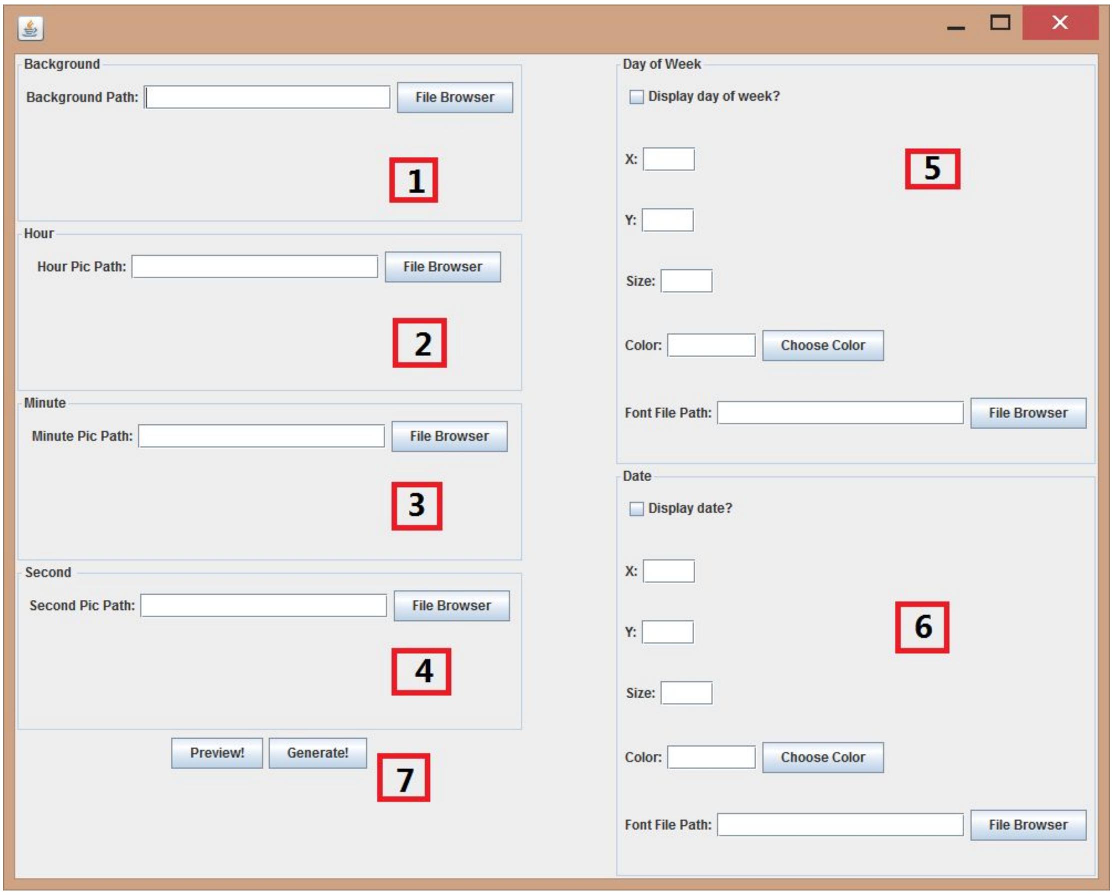


Figure 12: *First dialog of Watch Face Design Program*

The design page of the analog type contains seven parts: Background, Hour, Minute, Second, Day of Week, Date, and the buttons of Preview! and Generate!. (Figure 13) Users must first provide picture paths of the background, hour, minute and second. For clock hand pictures, it is required that the bottom of the clock hand must lie in the center of the picture. (Figure 14) After that, users can choose whether the day of week and current date are shown on the watch. If users want to display the date, tick the checkbox of “Display date?” Then users need to input values of the coordinate to define the position where the date will be displayed on the watch, choose size and color of the texts, and provide font file for the texts. Same steps are required for setting day of week. Finally, users can click Preview! button to preview the analog watch face, and click Generate! button to generate the watch face.



(a)

(b)


Hour
Hour Pic Path:
File Browser

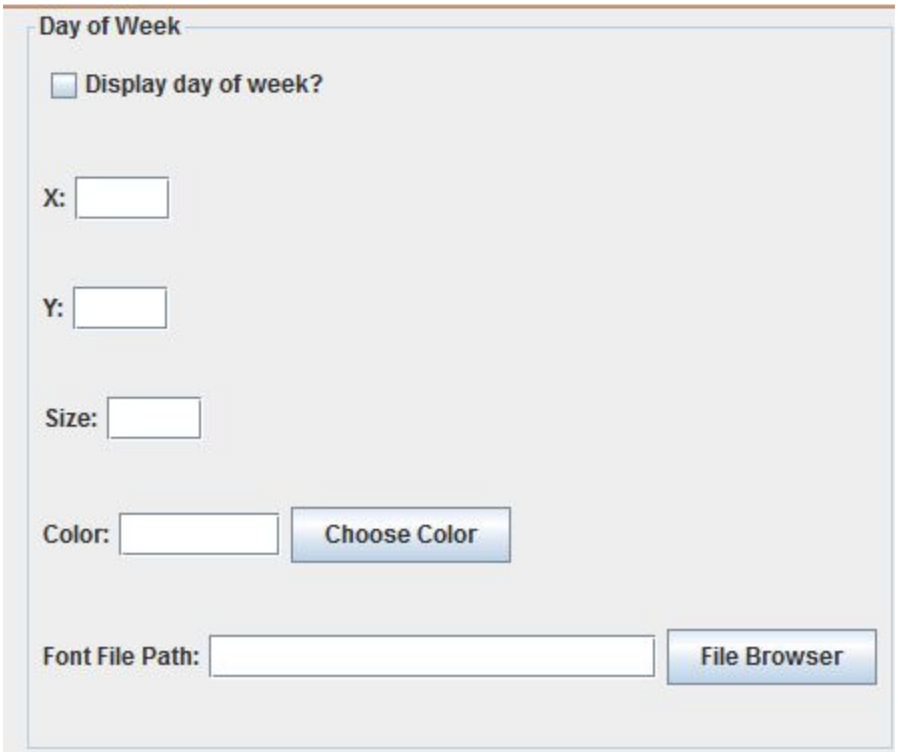
(c)


Minute
Minute Pic Path:
File Browser

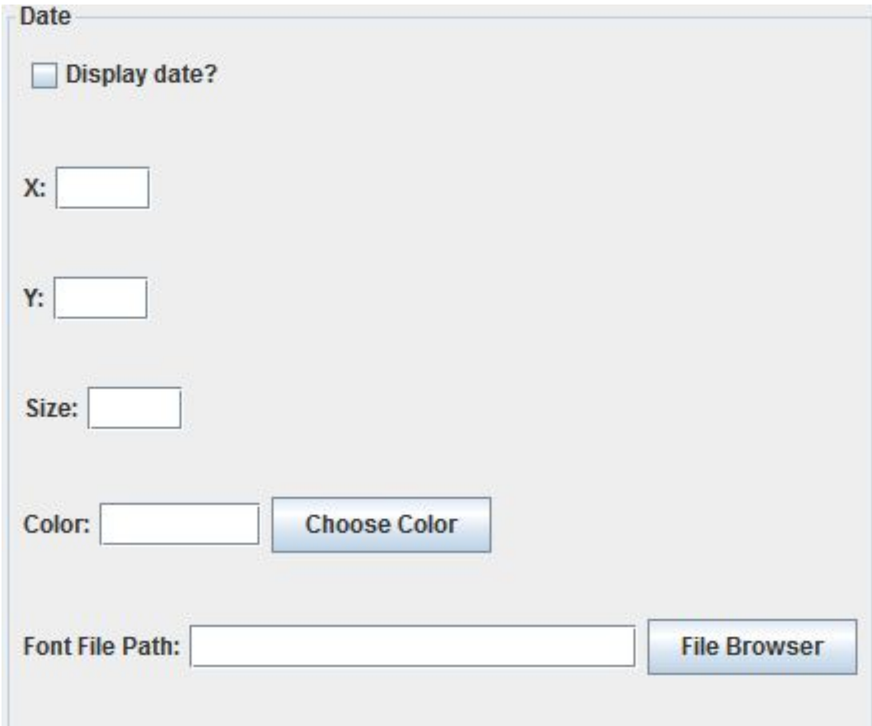
(d)


Second
Second Pic Path:
File Browser

(e)


Day of Week
Display day of week?
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser

(f)


Date
Display date?
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser

(g)

(h)

Figure 13: *Analog watch face design panel (a) Whole panel (b) Part 1 (c) Part 2 (d) Part 3 (e) Part 4 (f) Part 5 (g) Part 6 (h) Part 7*

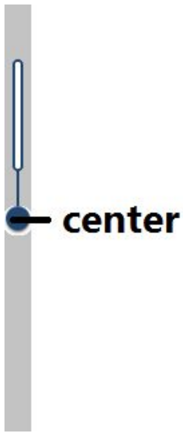
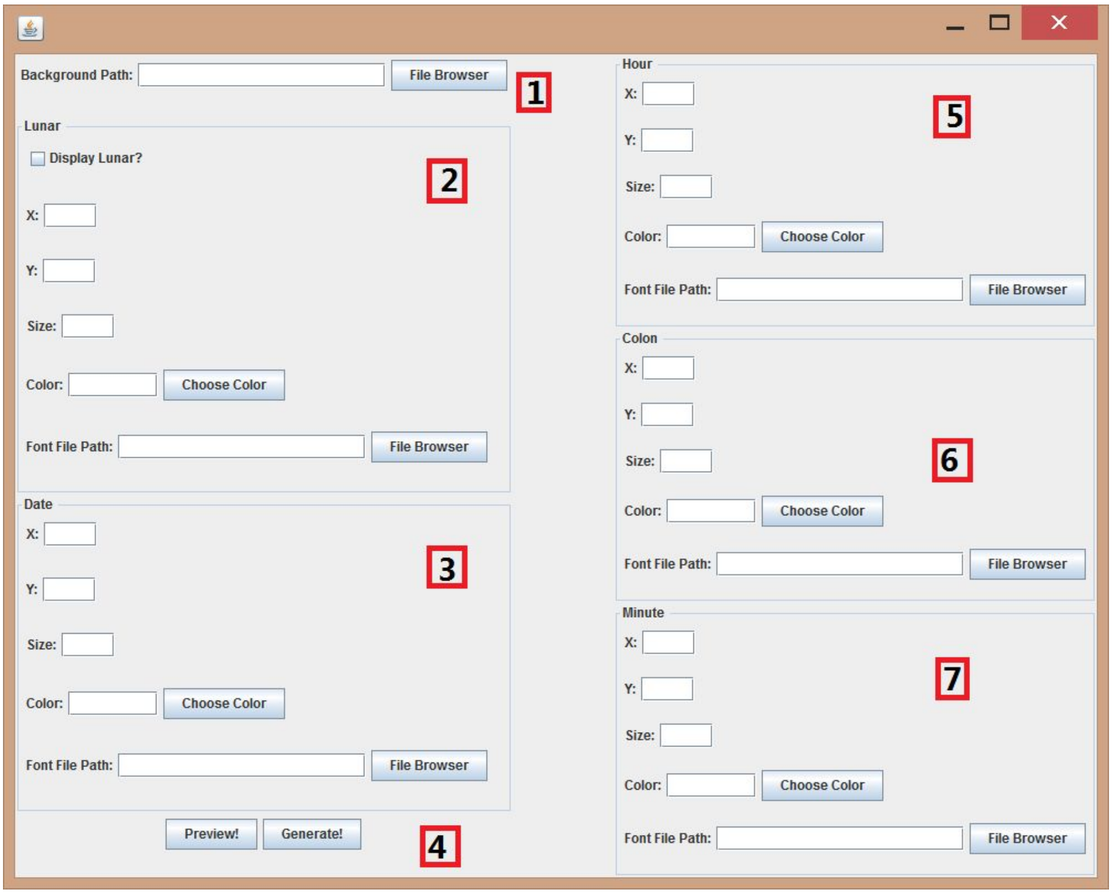


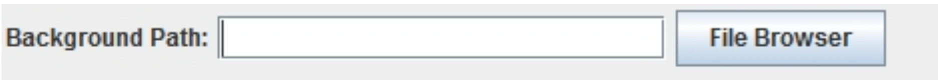
Figure 14: *Example of clock hand picture (The background needs to be transparent in the real case.)*

The digital watch face design panel also contains seven parts: Background, Lunar, Date, Hour, Colon, Minute, and the buttons of Preview! and Generate!. (Figure 15) First of all, users must provide the background picture path. Then users need to set the parts of hour, colon, minute and date by inputting values of the coordinate to define the positions on the watch, choosing size and color of the texts, and providing font file for the texts. After that, users can choose whether the watch face shows the lunar date. If users want the watch to display the lunar date, then tick the checkbox of “Display lunar date?” and set this part with the same steps as setting the parts of hour, colon, minute and date. Finally, users can click Preview! button to preview their work, and click Generate! button to generate their work.



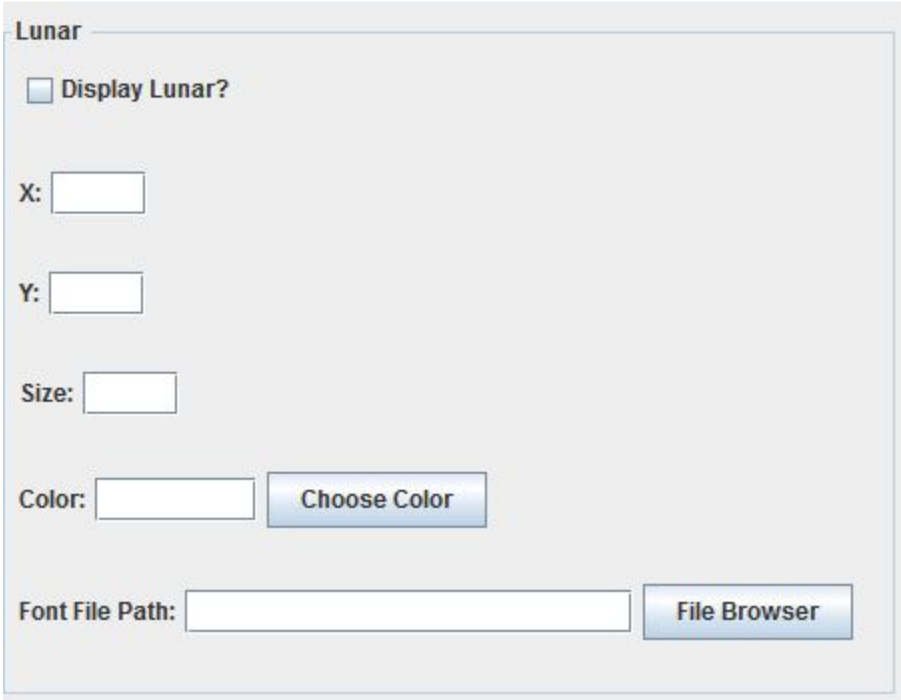
Background Path:
File Browser
1
Lunar
Display Lunar?
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser
2
Date
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser
3
Preview!
Generate!
4
Hour
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser
5
Colon
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser
6
Minute
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser
7

(a)

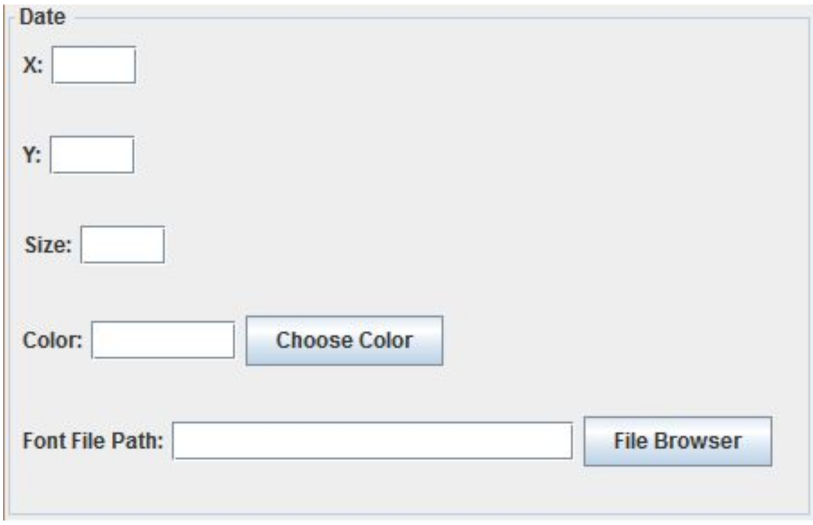


Background Path:
File Browser

(b)


Lunar
Display Lunar?
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser

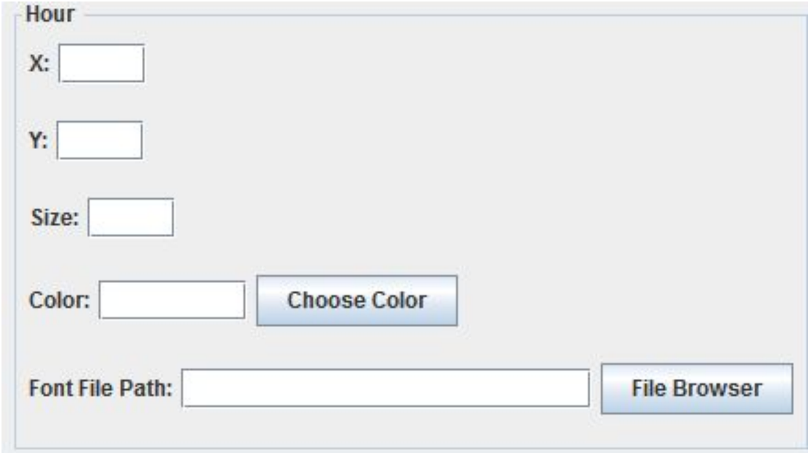
(c)


Date
X:
Y:
Size:
Color:
Choose Color
Font File Path:
File Browser

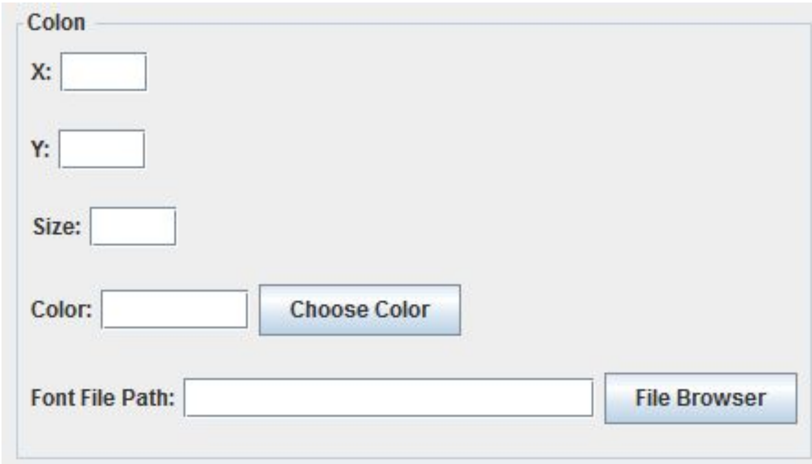
(d)


Preview!
Generate!

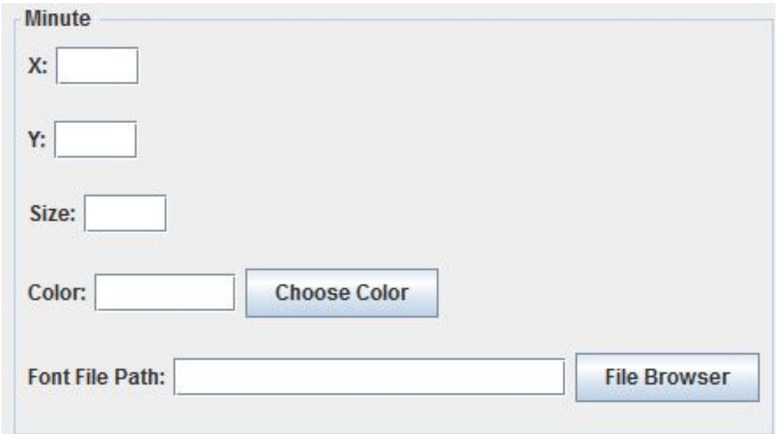
(e)



(f)



(g)



(h)

Figure 15: *Digital watch face design panel.*

After the Generate! button is pressed, the watch face files will be created in a folder with the name of the watch face the user input in the first dialogue, and the folder will be shown in the folder “clockoutput”. To install the new watch face, users must connect the watch to the computer. If it is the first time to use this program, users should build a new folder with the name “clocks” in the SD card directory of the watch. And users need to move the watch face folder into the “clocks” folder. After finishing all the steps, users can reboot the watch to view the new watch face.

To exhibit the efficiency of this program, two watch faces are developed as the demonstration of the program. One is an analog watch face, and the other is a digital watch face.

The first one is an analog watch face, based on the theme of zodiac. The background of the watch is a star map of Scorpio. The watch panel uses twelve signs of zodiac to represent twelve hours. And the clock hands used are in a classic style. (Figure 16)

Figure 16: *Watch face design example 1*

The second watch face is a digital watch face, based on the theme of star wars. The background is one of the most typical signs of star wars. The watch face shows the date at the bottom in green color, a color represents the light side in the start wars movies. Correspondingly, the watch face shows the current time on the top in red color which represents the dark side in the movies. And a traditional digital font is used to show the time to make it more like a digital watch. (Figure 17)

Figure 17: *Watch face design example 2*

In a word, this watch face design program can help a lot for users and designers to generate their own watch faces with more attraction and creativity. But this program still has space to improve. For example, it would be more convenient if the program could allow users to drag the elements to the position they like instead of inputting the exact position coordinate. And this program cannot reload the generated watch faces, which is inconvenient for users to edit the watch faces they made before.

6. Difficulties

Some difficulties and problems occurred during the development. The first problem is the change of the watch system. The system in plan was an android wear system while the system actually being used is android 4.4 system, a system running on the phone or pad. The IWOP system is a modified version of the android 4.4 system, which can run on the IWOP watch. The change of the system leads to the change of the methodology and approaches. The original methodology in the project plan was modifying the android wear system. However, it was changed to modify the IWOP system when it was found that the development kit is running the IWOP android system. Furthermore, the development methods had to be changed accordingly. The method of application development is changed from watch application development to ordinary android development, and the method of watch face development is changed from watch face development to android system framework development.

There are also some problems in the compilation of the IWOP system. The compilation needs to be performed on the 64-bit Linux system, so Ubuntu 14.03 was chosen to build the IWOP system. However, some tool files in the source code of the IWOP system are built in 32-bit system. Thus, the Ubuntu requires several 32-bit libraries to execute those files on the 64-bit computer.

Another difficulty is the layout design. The screen is rounded and very small, so there are some difficulties in the layout design. First of all, the system layout needs to show as many applications as possible in one page, and their names should be shown as well. Moreover, the applications' layouts need to fit well to the rounded screen, and at the same time, be convenient for users to use on the small watch screen. For instance, the layout of an input method must display a whole keyboard, and the buttons on the keyboard need to be large enough for users to press.

The last difficulty is the limited development time. Some applications are not improved owing to the restriction of the development time. For example, the Wifi Connection application and Input Method Picker could be merged into the System Settings, but the time is not enough to add the two applications to the Settings. Besides, the watch face design program can also be improved by adding the function of allowing users to drag elements to the position they like on the watch, as well as the function of reloading previous works as mention before. However, considering the limited development time, the above functions are not included in the project.

7. Deliverables

Date	Content	Deliverable	Status
4 Oct 2015	Study the IWOP android wear system; Decide the apps that will be added to the system; Make detailed project plan. Make a project web page.	Detailed Project Plan. Project Web Page; List of applications that will be developed.	Completed
Oct-Nov 2015	Develop the set of apps.	Apk files of the applications.	Completed
Dec 2015	Develop the set of apps; Prepare for the first presentation and interim report.	The new system with the applications.	Completed
Jan 2016	First presentation Preliminary implementation; Detailed interim report.	First presentation; Preliminary implementation; Detailed interim report.	Completed

Feb-Apr 2016	Finalized tested implementation Final report.	Finalized tested implementation. Final report.	Completed

8. Future Work

Although the modified system in this project can make users feel better when they use the watch, there are still some parts that can be improved in the future.

For the application part, the future improvement can be divided into three parts. First of all, some applications, like Wifi Connection and Input Method Picker, can be merged into System Settings as mentioned before. Secondly some internet-dependent applications can be modified not to rely on the internet. For instance, the voice recognition function of App Finder employs Baidu online voice recognition, which restricts the use of this app when the watch is offline. So in the future, this application can use some offline voice recognition libraries to support offline use. Besides, the Weather application and Web Browser application also need internet connection, which costs more power of the watch. So in the future, these two applications can be modified to use bluetooth to connect to the phone and then access to the internet. Thirdly, some applications excluded from the final result can be developed and added to the system in the future. For example, Wifi Automatic Switch is not included in the final results, because the Android system has a wifi lock service in the background, and through the service, the system can turn the wifi into the sleep mode when users turn off the screen, which lowers the significance of the Wifi Automatic Switch. Along with the limited development time, Wifi Automatic Switch is eventually excluded from the final results. But in the future, this application can still be developed and added to the system as a supplement to the wifi lock service.

For the system part, the system layout and system function can be both improved. The application page can be modified to be more beautiful and user friendly. The Launcher for sorting the applications can be modified to be more intelligent. For example, the Launcher, currently, can sort the applications according to the usage frequency, but in the future, it can learn the user's usage habit deeper and sort the applications on the basis of the user's habit, e.g. if the user usually first opens Weather application in the morning and uses Clock application at last in the evening, then the Launcher will place Weather application at the first place on the desktop in the morning and place Clock application at the top of the desktop in the evening.

For the watch face design program, more functions can be added to the program. Currently there are only two types of watch face displaying day of week and date. In the future, more types of watch face can be added, and the program can support to show the time in different time zones. Furthermore, the program now uses coordinate to locate the position of elements, which is a little inconvenient for user. In the future, as mentioned, it can be improved to allow users to drag the elements to the position they like on the watch face. Finally, this program for now cannot reload previous generated files as mentioned, so that it is very inconvenient for users to edit their previous work. So the future work should also include adding the function of reloading into the watch face design program.

9. Conclusion

In conclusion, this project aims to modify the current IWOP system because the system has poor layout and too few functions. Therefore, a new set of applications are added into the system, system layout and system function are modified, and a watch face design program is developed as a supplementary application for users and designers to generate their own watch faces. The new set of applications contains new developed applications and modified applications of the original system. The new applications include Airline Search, App Finder, File Browser, Notepad, Translator, Web Browser, Input Method Picker and Wifi Connection. Some old default installed applications are also modified to fit well to the rounded screen. Besides, the system layout is improved to be more user friendly. Power indicator and Wifi status icon are added to the system layout, and application pages now show not only the applications but also the application names. Furthermore, the system functions are also modified to be more powerful. Three input methods are introduced to the system, and the system launcher now has a new function, i.e. sort the applications according to the usage frequency of the applications. In addition, a watch face design program is developed to helps users and designers make their own watch faces with more attraction and creativity. All in all, the project has achieved all the objectives, while it still has some space for future improvement.

References:

1. Ingenic [Internet]. China: Ingenic Watch Open Platform; [updated 2015 Oct 17; cited 2015 Oct 17] . Available from: http://iwop.ingenic.com/zh-cn/develop/build-platform/burn-aw808/

2. CCSU [Internet]. USA: Central Connecticut State University; [updated 2014 Jan 14; cited 2015 Oct 17]. Available from: http://www.cs.ccsu.edu/~stan/classes/CS355/notes/03-AndroidUI.html