School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT COVERSHEET
GROUP ASSIGNMENT

Unit of Study: Mobile Computing

Assignment name: Build a Mobile App – Proposal

Assignment group members:

YUMING JIANG   460444981

YUHAO CHENG   450580673

Abstract

The purpose of this project is to build a notes-taking app. This project proposal aims to clarify the idea and set the schedule. It consists of five sections: Introduction, Objective, Related Work, App Storyboard and Schedule. In the Introduction and Objective, the motivation of this project and the purpose of the final app will be discussed. Pros and cons of five famous notes-taking app will be covered in Related Work. Also, the storyboard of the final app will be explained in detail in App Storyboard. In the end, the schedule of this project will be displayed.

Contents

Background & Significance

Nowadays, taking notes on the smart phone is essential in people's daily life. There are also many useful notes-taking apps such as Google Keep, Evernote, Any.DO and ToDoist, developed for Android to help people memory things easily. However, all of these apps have a significant problem that when the number of notes grow to hundreds or thousands, it is really difficult to find the note that the user exactly wants. Furthermore, these apps are always focus on just one or two functions, which is not enough to match people's needs. For example, Evernote is a kind of popular app that can let people take notes very easily but it can not alarm people to do something when it is necessary. To solve these problems, this project will try to build one new Android app to help people easily take notes, find notes and help to remind people what they will do next.

Objective

This project aims to help users finding necessary notes with both passive and active ways. For passive way, users can find a specific note with a sequence of distance, date and so on. The final app will also allow users to search notes with keywords. For active ways, the final app will also remind users the notes automatically according to the user-setting position, date and so on.

Comparing with related Work

To determine whether this project is valuable, four notes-taking app developed for Android is compared with the objective of the final app. Those four apps are Google Keep, Evernote, Any.DO and ToDoist.

①Evernote

Evernote is a kind of notebook application which has multiple functions. It was founded in America on May 2012. [1]
Nowadays, this application already has a large amount users and be highly acclaimed. But it still has some problems when comparing with our product.
The format of every file is very complicated.
Users may be crazy when they are going to finish the articles format, it is even more complicated than WORD! But our product is clear, easy, understandable and user-friendly note application.
It is just a note collector.

The easiest way for users to find a note is searching the name or keyword of the note. Honestly, it is a good function but it can not be the only function. So our product provides the sorting function according to the distance and time.

The functions in free edition is limited

Users cannot take notes when it is offline and can not set passwords for notes if they do not pay the money. The good news is our product is totally free!

Further more, it still has some drawbacks such as non-supporting adjusting images size. Anyway, it is a good note application but not a good editing application.

②**Google Keep**

Not like IOS, Android system does not have the notebook on their own. Every cellphone company has written a notebook into their phone system when the phones released, but it is just a simple application which provide simple recording functions only. According to this situation, Google decided to develop an application which can support all the Android system to let user save their notes and files online. [2] Further more, users can share these files to their friends by using this app. Comparing with our product, Google Keep is more similar with ours than Evernote, but there are still some differences.

It is like an alarming application

According to my research, people are more willing to use this app to remind them to do something instead of recording everything them need.

The number of words in this app is limited

We can not use it to record an article or a website when we need due to this reason. Users can only use it to write down some short notes. But our product has no words limitation.

Location limited due to relying on google.

We all know that China and some other countries can not use full google functions, so it will lose a huge amount of customers. But our product can will not rely on anything.

③**Any.DO**

Any.DO is one of the most famous notes-taking app which was launched on Android in November 2011. [3] After getting into the app, the list of notes will be displayed. However, only the excerpt of each note is displayed on the screen. The final app of this project will also show the location and reminding time. After clicking on the notes, the reminding time will be displayed. There is no search box in the main interface. The users can only sequence the notes with date and priority. Users should pay for the location reminder service.

④**Todoist**

Todoist is also very famous among notes-taking app users. [4] The date and excerpt of

each note is displayed in the list in the main interface. There is a search button in the top right of the main interface. Users can sort the notes by date, priority and name. However, this app can not link the note with the location. All of reminders are not free. The final app of this project will actively remind users things based on positions for free.

App Storyboards

The final app will contain 4 layouts. In this section, all of the four layouts will be explained separately in detail.

①Main Interface – layout 1

Figure.511

The Main Interface is the first layout that users will see after opening the app. As shown in Figure.511, in this Main Interface layout, there will be one list and three buttons initially. The three buttons are denoted as 'New' button, 'Edit' button and 'Sequence' button. After users clicking on 'New' button, the final app will jump to Editing Interface (which will be discussed in layout 3). After clicking on 'Edit' button, the final app will jump to Deleting Interface (which will be discussed in layout 2). This project will use widget 'Button' with attribute 'android: onClick' in Android API to realize this function. [5]
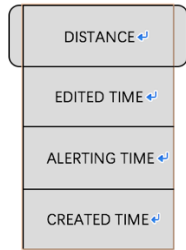
Figure.512

As shown in Figure.512, after clicking on 'SORT' button, users can choose the way of sequencing the notes. The sequence of all of the notes shown in the list will be changed based on the users' choices. This project will use the method 'sort' in 'jave.util.Collections' to achieve this goal. [6]

Users' notes are shown in the list. The list will be displayed with the widget 'ListView' in Android API. [7] In this layout, only the abstracts of the notes are shown in the list. The abstract of each note typically contains the user-setting position, editing time and the excerpt of that note. After short clicking on the abstracts of the notes, the final app will jump to the Editing Interface.
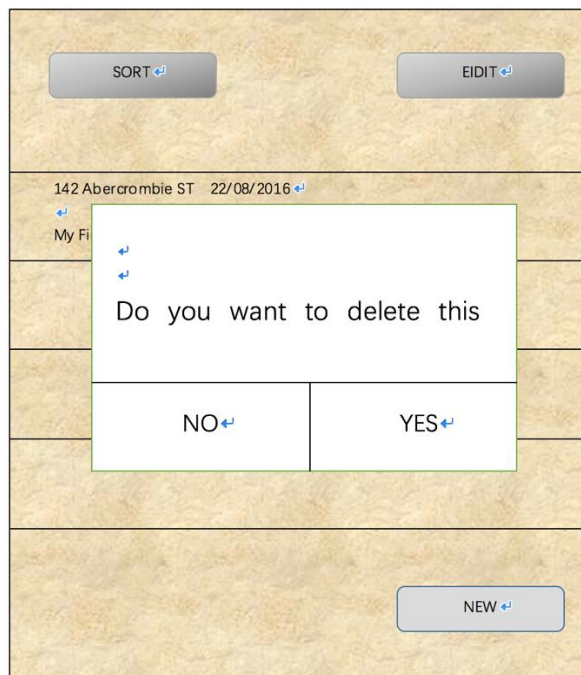

Figure.513

As shown in Figure.513, after long clicking on the abstracts of the notes, the final app will pop up a dialog to ask that if users want to delete this note. The dialog will be built with class 'android.app.AlertDialog' in Android API. [8]
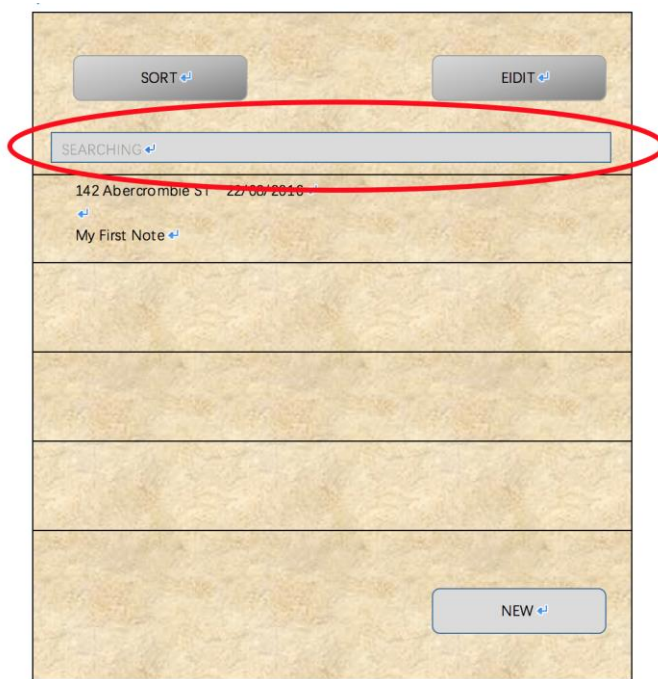
Figure.514

As shown in Figure.514, after users sliding fingers down, the searching box will be shown upon the list. The searching box will be based on the widget 'SearchView' in Android API. [9]

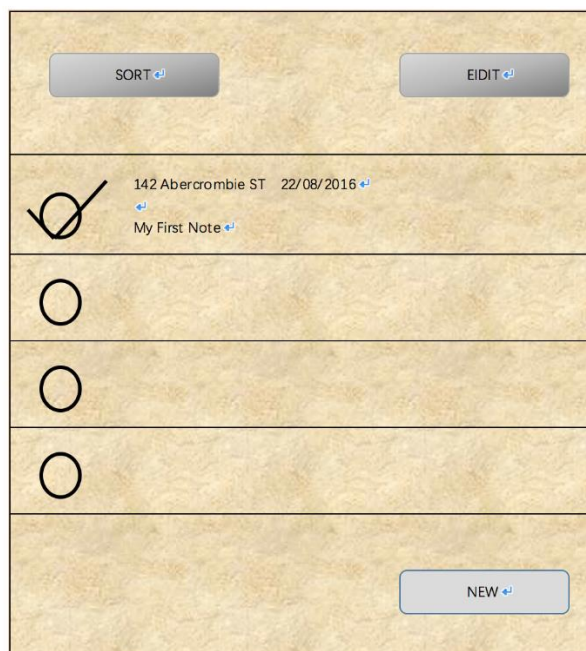②**Deleting Interface – layout 2**

Figure.521

After clicking on 'Edit' button in the Main Interface, users will see this Deleting Interface. As shown in Figure.521, in this layout, there will be a list and four buttons. The four buttons are denoted as 'Delete All' button, 'Delete Selection' button, 'Cancel' button and 'Sequence' button. The function of 'Sequence' button is the same as the

function of 'Sequence' button in the Main Interface. After users clicking on 'Cancel' button, the final app will jump back to the Main Interface without changing anything. After clicking on 'Delete All' button, all of the notes displayed on the screen will be deleted. After clicking on ''Delete Selection' button, all of the selected notes will be deleted. There is also CheckBox before each note displayed in the list. The CheckBox is used to check that if users want to select that note. The abstracts of the notes are also shown in the list. This project will use the widget 'CheckBox' in Android API to implement this target. [10]

③Editing Interface - layout 3

Figure.531

This is editing interface layout. After clicking the notes on main interface, the app will jump to this layout where users can edit their notes. In this layout, we use a plain txt area to let people type their notes and no character limitation. What is more, users can add pictures in photo gallery or by taking photos using camera into their notes. The layout will also show the creating time and last editing time on the right corner above the editing area. Furthermore, users can share, delete and renew their notes by clicking different buttons (using onclick functions in Android studio). Lastly, users have to type the address in a right format so that app can recognize it and choose time by using the roller.
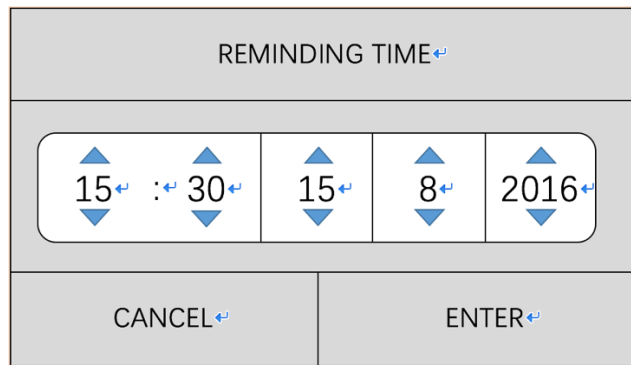
Figure.532

When clicking 'SET TIME' button, this roller will jump out and let users choose the remind time. This function will use the Timepicker and Datepicker in Android studio.
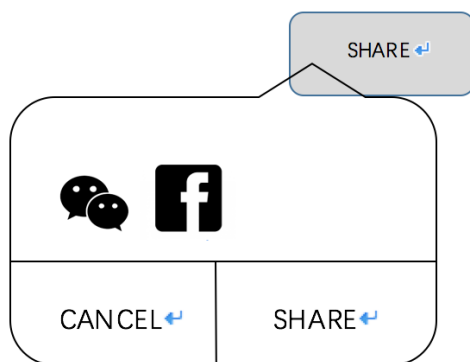

Figure.5.33

When clicking 'SHARE' button, this window will jump out to let users choose the note to 'Wechat' or 'Facebook' which is using the 'dialog' function in Android studio.
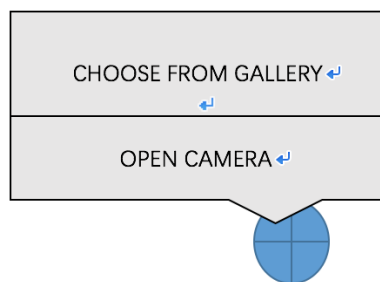

Figure.5.34

When clicking this button, the dialog will jump out to let users add the photos into their texts. It can be added by choosing from gallery or taking by camera. This function will be realized by using 'dialog' and 'picture' in Android studio.
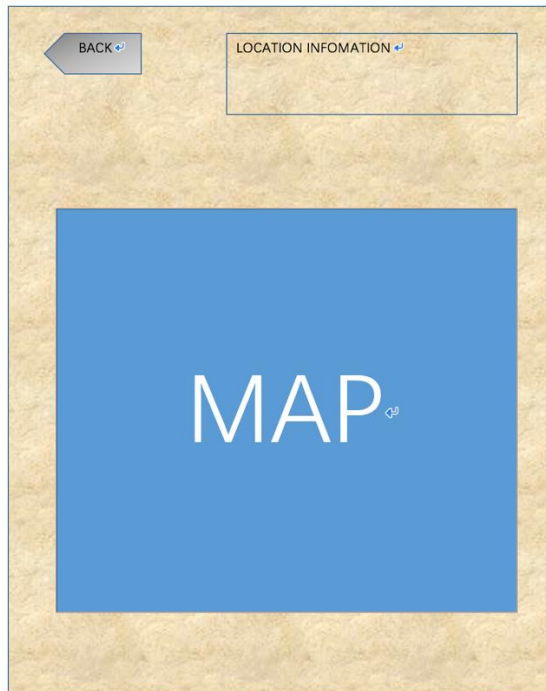
④Map Interface – layout 4

Figure.5.41

This is map interface which will be shown after users enter the location and click it. In this interface users will see the information of the location, not only the address but also some simple information about this place. Also, the map area will show users the exact address where users entered before. In this interface, the map function in android studio will be used.

Conclusion

As a conclusion, our product was designed to make a combined application of navigation, memo recording and notepad. Nowadays, products which have these features are really few in the market, some applications have achieved this goal but not perfect, where others are only focused on one of these functions. Compared with other applications, the advantages of our product can be concluded as some aspects.

Firstly, the notes can be sorted according to time or place so that the user will clearly find memo matters.

Secondly, if the reminder time has been late, we can also be triggered by location which is really goo function for users who have poor memory.

Thirdly, our application is no characters limited, users can add web pages, images and links that enable users to obtain a better recording experience.

For now, the main functions of our product are focused on offline, because in the final analysis this is a notepad app and the only online function is sharing the cloud or sharing to a friend. In fact, it is already fully enough for a notebook application.

Week Plan

Week 1: Making a group, getting familiar with Android Studio.
Week 2: Discussing the Project-Assignment with group member to realize the goal of this assignment.
Week 3: Deciding the application we will make.
Week 4: Starting writing project proposal.
Week 5: Finishing the project proposal.
Week 6: Making some changes on proposal according to the group discussion.
Week 7: Starting programing.
Week 8: Finishing basic layouts.
Week 9: Finishing the jump-out windows and super links. And debug the programing.
Week 10: Finishing the demo and try to run it on phones.
Week 11: Designing and finishing UI for the application.
Week 12: Packaging the application and make out the final application.
Week 13: Submitting the application and doing a presentation.

References

[1] https://evernote.com/?var=c&noredirect
[2] https://www.google.com/keep/
[3] https://www.any.do/
[4] https://todoist.com/
[5] https://developer.android.com/reference/android/widget/Button.html#
[6] https://developer.android.com/reference/java/util/Collections.html#
[7] https://developer.android.com/reference/android/widget/ListView.html
[8] https://developer.android.com/reference/android/app/AlertDialog.html
[9] https://developer.android.com/reference/android/widget/SearchView.html
[10] https://developer.android.com/reference/android/widget/CheckBox.html