

CSE 311 Final Report

Group Stardust

(Yiheng Dai, Sida Huang, Jiamin Chen, and Zheng Wu)

Department of Computer Science and Software Engineering

Xi'an-Jiaotong Liverpool University

{yiheng.dai16, sida.huang16, jiamin.chen16, zheng.wu16}@student.xjtlu.edu.cn

Source code: <https://github.com/Taku-firi/Cabicat>

Abstract

The presented project called ‘Cabicat’ is about developing a lightweight file storage system with automatic volume control. This project was developed in Java with Android Studio, and Git was used for version control. So far, the team has generally completed the related works after the demand freeze. The entire development cycle meets the necessary stages of software engineering. At each stage, team members were assigned corresponding tasks. During the requirements analysis stage, the team conducted a detailed questionnaire survey to fully analyze the product positioning and user requirements. In the development phase, the design content mainly includes Activities and Fragments design, navigations between different activities, Database design, File Operation and UI design. Continuous support and maintenance will be provided based on user feedback and market changes.

1. Introduction

As mobile devices play an increasingly important role in people's lives, people tend to store various of personal information in them. If private leak happens because of the loss of mobile phone, it will impose a significant adverse effect on user.

Before starting the project, the team conducted a survey, which shows that more than 90% of Android users worried about their information leaks caused by devices lost. Therefore, the team decided to develop a file storage system to solve the problem. This is one of the motivations to developing ‘Cabicat’. The name ‘Cabicat’ is composed of ‘cabinet’ and ‘cut’, which implies storage and delete. These two are both vital parts for this project.

After discussed to start this project, members did related research on Android apps in the market such as the file managers of OPPO, VIVO and Huawei. For OPPO, it was clear that it lacks the significant privacy protection function. Despite its simple classification of file types, it was still difficult to manage important files. By contrast, Huawei had a safe box but with obvious use defects. The entire folder was put into the box, which means it took more time when importing some large files such as videos. Compared to these managers, the Cabicat has four innovation points:

- Ease of use
- Time saving for searching important files.
- Preventing the leakage of privacy in some situations.
- Releasing phone storage by deleting documents intelligently.

This report will describe the application design and system architecture. Evaluations will be conducted on the problems group have encountered and the usability of Cabicat.

2. App Design

As illustrated in presentation, Figure 1 shows the basic system architecture.

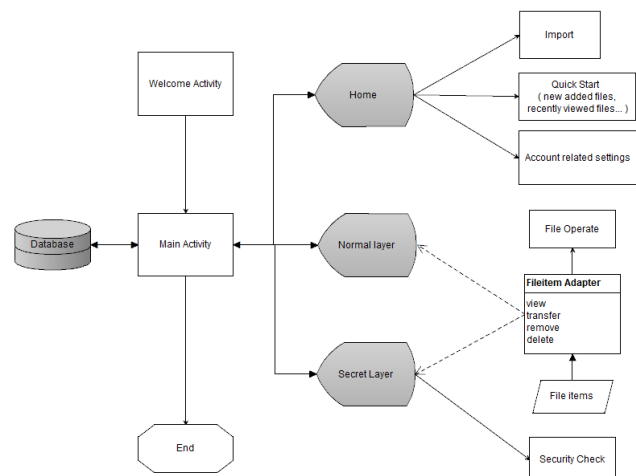


Figure. 1 Cabicat Core Design

Applications are mainly divided into three parts, the interface layer, the execution logic layer and the data operation layer. The interface contains series of xml files and classes about the activity layout and adapters. For the execution logic, the welcome activity is first called to show a boot page and then enter the main activity. The main activity is subdivided into three separate fragments to manage the logic of the home page, normal layer and secret layer with a bottom navigation layout. The home page is utilized to interact with users, functions provided including Import, Recent files, Account-related operations and so on. Once users import a local file into the Cabicat system, data

about this specified file will be record into the system's SQLite database. The account-related operations include Register and Change Password, the password will be with encrypted with md5 to increase the security. The remaining two segments take over the normal and secret layers, including content display, and provide interface for users to directly manipulate the file, such as transfer between different layers, remove from Cabicat or delete the file. Finally, the data operation layer deals with all the file data including database storage and SharedPreferences, while taking charge of real-time update.

The developing process of Cabicat is a gradually Incremental development procedure. At the beginning, the application had only the fundamental framework with the home fragment, the normal layer and the secret layer aside from a navigation bar for navigating between these three basic pages. After the first presentation, the database in the back-end had been built to package the elementary attributes of each files. Then the import method had implemented to get the required file item by utilized the file manager that pre-built inside the Android system. After obtained file items to the database, the welcome activity and a relatively user-friendly user interface (UI) had been structured. At the same time, the encryption and register system had been coded. The next function that had implemented is a help class that used to open and delete different type of files. For the last stage of the currently finished developing, the team added a About activity to show users some related backgrounds and the information for open source libraries and licenses.

3. Evaluation

3.1 Problem Solving

During the process of completing the code, the group had encountered many problems, most of which came from the code part from front-end to back-end. Many problems have been solved via the group discussions and asking help from the professor. When the group were designing the structure for the home page, several problems arose regarding the page display. For example, a vertical scrollbar had been designed in the home page, however, when tried to drag the bar up and down, it seemed that a part of the words cannot be shown in the screen. The team checked the component tree of the design but could not find any problems. Finally, one of the members found that these invisible items were covered by the navigation bar, so the group added an empty TextView and set the margins to solve this problem. This is one of the typical problems the team faced.

However, we still have some problems which have not been solved. For example, there are loopholes in the delete () method. Files cannot be removed successfully in some

devices. In addition, the deletion in time function has not been developed. Furthermore, Security issues still need to be discussed, and our database is currently not secure and garbled password protection is not enough, it seems that md5 has free decoding tools on the Internet. Another problem is that we can only obtain the path and name of the required file, but not able to get the size of the appoint file. Therefore, volume control cannot be achieved in code and the priority algorithm cannot be well test as the size of the item is a vital factor. The import method of Cabicat has implemented by the file manager that built into android system, hence the file manager in each devices of users might different, this characteristic may also cause some problem. The group will continue working on trying to overcome these limitations.

3.2 Usability Analysis

People through a huge number of standards and principles to justify the usability of products. The purpose of this is to identify whether this application allow users to use easily and they will not feel troublesome. If one application has low usability, it will affect the users' experience. There are four criteria to prove that Cabicat has good usability.

Firstly, User Interface is to give customers the most intuitive experience [1], a clear structure of the User Interface can allow the first-time users to understand the structure and functions more convenient and faster. For example, the home page of Cabicat with the username at the top with background. Below buttons have three functions to import files, register or change password. Secondly, there are two card view parts that can allow users to check their top 10 latest visited and added files which can drag left or right.

Then, the standard guidance on the buttons are also a significant criterion. Users can get the correct indication from the words on the Buttons, and these buttons will not mislead users to do something wrong. Moreover, users can get a response when they click the buttons. For example, when customers want to import files, there are two transparent button Normal Layer and Secret Layer to allow them to choose. When users click on these two buttons, they will change color as light pink to response.

The following demonstration shows the usability in the front-end. The improvements of usability in the back-end are also considered in Cabicat. The algorithm of remember password function in the secret layer was implemented. When users want to check the secret layer at the first time, they are required to input the correct password to authorize the identification. However, after that if they want to operate in other pages, they don't need to input password again because Cabicat has already remembered them.

4. Conclusion

In conclusion, each group member has learned a lot from this project. At the beginning, every team member manages to propose innovative ideas about the scheme of the application through brainstorm. Each time the team came up with new ideas, members would discuss the advantages and disadvantages through analyzing the usability of this application, improving the critical thinking. During the developing process, the team encountered a huge number of bugs and difficult problems. However, after heated discussions and required help from the professor, most of the problems has been solved actively. Furthermore, throughout the project, members realize the consciousness of teamwork and get familiar with the developing process of an Android application.

REFERENCES

- [1] N.Babich, (2019, Oct, 7th). The 4 Golden Rules of UI Design, [Online]
Available:
<https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/>