Winter Term Retrospective

Team members:

Wei Huang

Zhongkai Huang

Zijing Huang

Zisong Zhang

1 Introduction (Zhongkai Huang)

This document mainly reviews the work done by our team during this term. We will introduce the development tools and technical methods we use in detail, such as what tools we use for development and which test method we use to test. After that, I talked about how we carried out project management, and finally, we have some suggestions for project partners and the team to inherit our project.

The project is an Android eCommerce application targeting RPG *(role-playing-game)* in the tabletop genre, offering a curated shopping experience for game merchandise and the games. Besides the eCommerce experience driven by eBay public APIs *(application programming interface)*, some gaming tools such as a dice roller will enhance RPG games' experience. The preliminary model currently includes the app will have a bottom navigation bar and a search bar at the top of the app, social page, favorite page, and tools page. The overall look of this app will be similar to the Google Play Store on Android devices. The application will use Kotlin for the Android development of the project. Java will also be used for the project as needed to meet project requirements.

2 Tools and Technical Approaches (Wei Huang)

These tools are the applications that are good and easy to learn. They are accommodating for our project. It is important that they are very friendly to the developer. The learning curve of these tools is shallow, and the developers can speed up the progress of the project. I would like to choose the same tools for similar projects. These tools are the perfect tools for Android developers and happen to meet our requirements and satisfaction.

2.1 Language

Kotlin

Our project uses Kotlin to build the Android code. Kotlin is perfectly compatible with Java. Kotlin is an object-oriented programming language with functional programming ideas, and it has many features of static programming languages, easier to learn. It will help us to learn the language quickly and easy to code for the project. We enjoy using Kotlin to build Android code.

Java

Java is a supplement language for our project. Java is an object-oriented programming language that absorbs some advantages of the C++ language and discards multiple inheritance and pointer concepts, which are difficult to understand in C++. Hence, the Java language has two characteristics of powerful and easy to use. Besides, Kotlin and Java can call functions from each other. Therefore, we use Java to implement some more accessible code for us than Kotlin in some back-end functions.

2.2 Application tool

Android Studio

This tool is specially built for Android development, based on IntelliJ IDEA. Android Studio provides integrated Android development tools for development and debugging. Our project focuses on the android application, so the android studio is a good development application for our project. An emulator feature in Android studio makes it easy for the developer to know how the feature, which the developer adds, looks like in the mobile phone emulator. Also, we can connect to the android mobile phone to check if the feature, which the developer adds, works well. Android Studio is easy to use and quickly tests for android applications.

GitHub

We built this project together on the GitHub repository. Each of us was responsible for a part of the project, and we needed to merge each other's code without breaking it. Besides, GitHub can commit the code and track its workflow on the website with git or Github Desktop. We can implement individual features on different branches for each person, making sure not to affect other people's code.

Postman

Postman is an API/interface function testing tool that makes it easy to plug data, view responses, set checkpoints/assertions, and automate testing to a certain extent. Our project requires the eBay public API/ interface to contact the eBay website, so Postman is an easy and convenient API/interface function testing tool. We can use Postman to check the success of the interface quickly.

Photoshop

Photoshop is essential for our project UI design. Our project is an application where users can see the table game's information or buy the application's table games. Therefore, there are many pictures in our application, so we need to design the project's image. Photoshop is an excellent application to edit and design the picture, and it is easy to learn. We can create a nice photo for our project.

2.3 Test

We plan to use a modular Testing Framework. After completing each feature, we will test the application to make sure the integrity of the feature. We check if the feature works well by the emulator in Android Studio or connect to an android mobile phone. We can directly use the application to make sure the application is friendly to users. If everything goes well, the developer is going to do the next feature. After we have made the whole application, we need to do the user test for the project. Some users will test the application and give feedback. Then, we can improve the application by the users' feedback.

3 **Project Management** (Zijing Huang)

We divided the work by features *(table 1)*. Each member took one part of the project and implemented the features of that part throughout this term. It is a good approach since everyone's algorithm might be different. Dividing by feature could avoid redundant work.

Wei Huang: Social Page	Zhongkai Huang: Favorite Page	Zijing Huang: Home Page	Zisong Zhang: Game Tools Page
Cardview Image	Product Page with more images and details	Product Card with title, image, and price	The different choice of dices
Click and jump to the website	Add the product to the favorite list (page)	Search Bar with the search feature	Click and jump to the dice shop

table 1

If anyone not available to work or complete the sprints, other members would help him/her out and cover the "loophole" during the sprint. After the sprint, the member who couldn't complete his/her work during the sprint would take more work at the next sprint.

We wasted some time merging different member's branches into the main branch. Some were because the branch was outdated and had conflicts with the main branch. When this happens, each member will take a deep look at the conflict or outdated files and manually merge them into their branches.

Here's our development process. Each week, each member will develop one tiny feature in their branch and then merges it into the master branch during weekends. If any member runs into some issues or errors during compiling, the site and sources for the solution usually will be Stack Overflow or Google's Developer Documentation. If he/she still has trouble running the code, he/she will post the question in the chat, and some of the group, or the engineers from eBay, will answer the questions.

If we could start over, we would contact each other more frequently to make sure each member was on the same page. We had some issues and errors with merging during this term, where the member's branch was outdated before merging into the master branch. We would make sure each member had the latest version of the master branch. Therefore, merging could be a lot easier and save a huge amount of time.

4 **Advice and Recommendations** (Zisong Zhang)

The project partner can explain the project requirements in more detail because our team sometimes had difficulty understanding the project partner's requirements. Although the partner prepared a requirement sheet for our team, some of the requirements were vaguely explained. We may sometimes misunderstand some details of these requirements, which may lead our team to develop in the wrong direction, thus wasting more time. Therefore, our team suggested our project partner give a more straightforward explanation in the requirement list and add more pictures and examples for our reference to improve our team's efficiency.

Our team would have liked more help on blockers because most of our members are beginners in Kotlin(our primary programming language). At the beginning of the program, we had no experience with Kotlin, so our team spent a lot of time learning the Kotlin language. So we hope we can get more learning materials such as documentation and videos from our project partner. Simultaneously, the weekly meetings are not effective in solving our development problems, because the time of each meeting is limited and sometimes we can't solve all development problems. We would like to add a weekly meeting with eBay engineers to solve programming problems.

The advice I would give to the team inheriting our project is the API *(application programming interface)* key confidentiality issue. Our program has to get data from eBay API, so we need to use eBay's API key. It is worth mentioning that according to eBay's confidentiality agreement, the API key needs to be kept confidential, so your team can be careful not to save the API key directly in the file to prevent the API key from being leaked. Your group can keep the API key in the config file and read the API key from the config file every time the API is used.