

# National College of Ireland

Program name: **Software Project**  
Specialisation: **BSHBIS4**  
Academic Year: **2020/2021**  
Student Name: **DERU MA**  
Student Number: **x17466626**  
Student Email: [x17466626@student.ncirl.ie/](mailto:x17466626@student.ncirl.ie/)  
[emilymaderu@icloud.com](mailto:emilymaderu@icloud.com)

**<Android mobile application – On The Go>**

Technical Report

**Supervisor: Catherine Mulwa**

## Contents

Executive Summary .....	2
1.0 Introduction .....	2
1.1. Background .....	2
1.2. Aims.....	4
1.3. Technology .....	4
1.4. Structure .....	5
2.0 System.....	6
2.1. Requirements.....	6
2.1.1. Functional Requirements.....	6
2.1.1.1. Use Case Diagram .....	6
2.1.1.2. Requirement 1 <Name of requirement in a few words>.....	6
2.1.1.3. Description & Priority.....	6
2.1.1.4. Use Case .....	6
2.1.2. Data Requirements .....	30
2.1.3. User Requirements .....	30
2.1.4. Environmental Requirements .....	30
2.1.5. Usability Requirements.....	31
2.2. Design & Architecture .....	31
2.3. Implementation .....	32
2.4. Graphical User Interface (GUI) .....	35
2.5. Testing.....	40
2.6. Evaluation .....	42
3.0 Conclusions .....	42
4.0 Further Development or Research .....	42
5.0 References .....	43
6.0 Appendices.....	43
6.1. Project Plan .....	43
6.2. Reflective Journals .....	47
6.3. Invitation & consent form.....	52

## Executive Summary

This project was aiming to build an android mobile application by using android studio. The overall problem has proposed and being solved is to provide a platform for local customers to order things from multiple stores, as well as to help customers communicate each other, in order to develop a self-help environment, instead of delivery service provided by stores. Users could utilize this application to “be” the delivery man to help other local users. The application includes following functionalities: registration, login, Post lists, view post, comment, like, home page, categories, item lists, view item details, cart page. Add to cart, payment and message functionalities were planned to develop in the future. The main point for this project is to allow users to view the items and be able to communicate. This technical report includes seven major aspects which are introduction, different types of requirements, design and architecture, implementation, testing, evaluation, and conclusion. Each aspect includes a detailed description and interpretation regarding this project at a professional level. Those seven aspects would help the audiences to understand the rationale of this project as well as the essential technical requirements were utilized in the project. The purpose of this report is to demonstrate the project in different aspects and dimensions more profound rather than just displaying the project itself, in order to let the audiences, gain a better insight of this project.

The result of the report is focusing on the reflections, recommendations, and further plans more than the demonstration of the application itself. Therefore, this report not only documented the entire process of the project from different dimensions, but also summarized the vacancies that need to be filled in the further development of this project. For example, one of the major vacancies is the lack of few available functionalities within the application at the current phase, thus, this report would provide some valuable recommendations and contributions to the further development of the project based on these vacancies.

## 1.0 Introduction

### 1.1. Background

According to the data (Max Roser et al., 2015), the internet has been introduced few decades ago, email has been used since the 1960s and the sharing of files has been around since the 1970s, however, the huge revolution of our communication has started since the creation of the world wide web in 1989. Even since the world wide web has been released all over the world, the era of the internet has begun. As the result of that, the use of mobile phone has grown sharply as shown in figure 1.

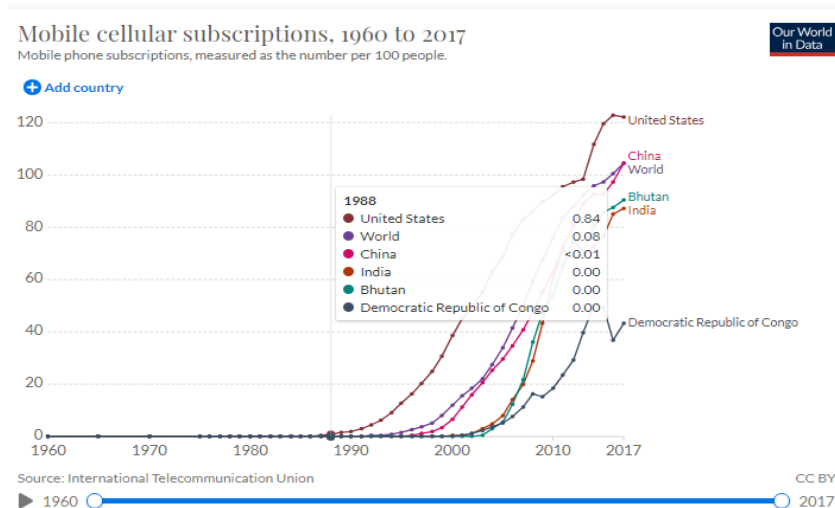


figure 1. (Max Roser et al., 2015)

Many popular social media and mobile applications has risen after the development of the use of mobile phones since 2004 as shown in figure 2.

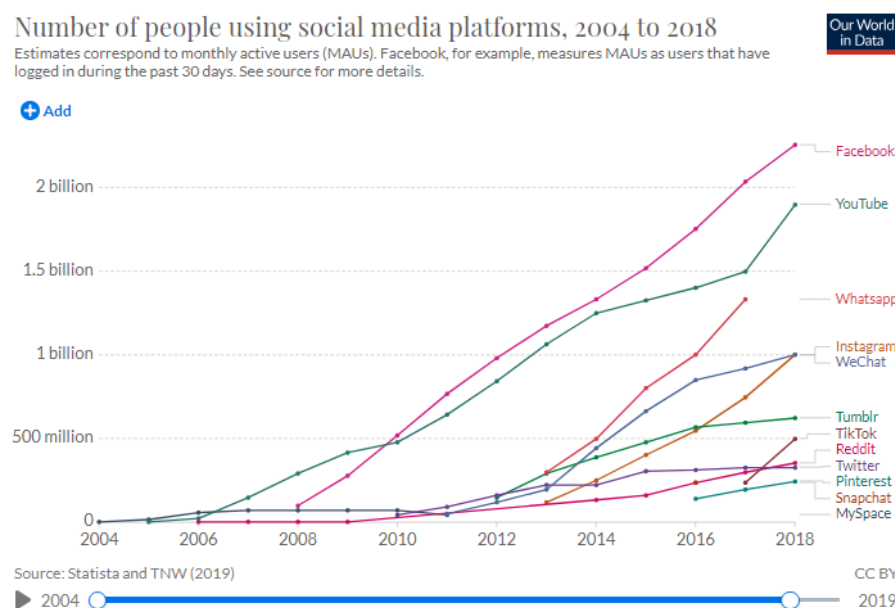


Figure 2 (Max Roser et al., 2015)

According to these graphs, they have indicated that the mobile phones and mobile applications are definitely one of the most popular trends worldwide right now, especially in this particular hard time this year. Most things that used to be off-lines has been transforming into on-lines, for example, a lot of walk-in services in restaurant or shops has been cancelled and changed into online services. Therefore, I have chosen mobile application as the theme of this project. I think mobile application will not be replaced soon in line with this research.

Additionally, in the comparison of web applications and mobile applications, mobile applications are faster, safer, and greater functionality as the users have accessed into the resource system than

web applications, according to (STEVENS, 2018). Thus, i have chosen the mobile application rather than the web application as the project theme. Furthermore, I am familiar with android studio and there are a lot of public resources that can be used online regarding this software, so I have decided to build an android mobile application by using Android Studio as my project initially.

Moreover, according to the research (Hindi, 2020), on-demand, mobile commerce and mobile wallet applications are the trends in 2020. Therefore, I have proposed to build a mobile application that can be used for both sharing and ordering things online together, in order to make users life much easier in this difficult period.

## 1.2. Aims

Overall aims:

1. Provide a friendly and easy mobile application for local users.
2. Develop key features to achieve the usability and integrity.
3. Achieve most implementations for the prototype.
4. Be more unique and gain much absolute advantages in the market.

In the project proposal, I have proposed few objectives at the beginning. They were as follows:

**“Objective 1:** Aim to achieve login, registration, message, comment, post, Cart and payment function before midterm.

**Objective 2:** Aim to modify payment function and try to make it innovative. For example, could I give users certain amount loan every month, so that they could pay afterwards or installment, in order to promote their usability?

**Objective 3:** Be innovation and Do more investigation. Try to make a unique function for my project.”

Those are objectives of the entire project, only objective 1 was for the mid-term. The main content of these objectives has not changed too much at this point of the project. They have remained the same as before, however, the project has not fully achieved objective 1 at this phase. The application only achieved Login, Registration, post and view the detail of the post functions at the midpoint phase.

Therefore, the main and prioritized objective for the further development of the project would be the implementation of those functions mentioned above (Cart, Message and Payment).Furthermore, the usability of those functions within the application for users is the most significant aim after the mid-term. The unique and innovation concept for the application will be the complementary aim in the late phase of this project.

## 1.3. Technology

**Hardware:**

Dell Laptop

**Software:**

Android studio

Firebase

Gmail

YouTube

StackOverflow

Udemy

Udacity

GitHub

Build an android application is cheaper than IOS application, so I have decided to build an android application. The main technology used in this project was Java, the toolkit used was Android Studio. Android studio has its own SDK tools, so the Android SDK were used in this project for the emulators. The reason that I have chosen Java rather than Kotlin as follows:

- I am familiar with Java.
- Java is much easier to understand and learn.
- Android Studio itself is built on Java.
- Java are much more compact and lighter than Kotlin.
- From the perspective of results, Java leads to a faster app experience.

Even though Java is more suitable for this project, it has some disadvantages as well. The major two drawbacks where Java contains more codes so that increase the chance of bugs and errors, as well as Java requires more memory so that tend to be slower. These two disadvantages have affected the project in some respects, it would be mentioned in the evaluation section of this report.

Moreover, I have used YouTube, Udemy and StackOverflow as my technical assistant and support during the project. There are a lot of useful and effective tutorials on these platforms that helped me to build this application. And also helped me to solve many issues raised during the project. There is one other platform called Udacity, it also contains some tutorial regarding android studio. I have learnt few courses on Udacity before, they were quite beneficial, thus I might use it as my technical support as well in the later stage of the project.

#### 1.4. Structure

**Requirements:** Functional Requirements

**Use Case:**

- Use Case Diagram
- Use cases (includes 15 use cases: Splash Screen, Login, Registration, Main page, Home(Post) page, Add Post page, View post detail page, Comment, Like, Cart (Category) page, Item page, View item detail page, Message page, add to cart page and payment page)

**Implementation:** Introduction of the implementation of the application in general.

**Graphical User Interface (GUI):** 13 UI includes the screenshots.

**Testing:** Testing tools and results.

**Evaluation:** General evaluation of the project.

**Conclusions:** Advantages and disadvantages of the project.

**Further Development or Research**

**References**

**Appendices**

- Project Plan
- Reflective Journals

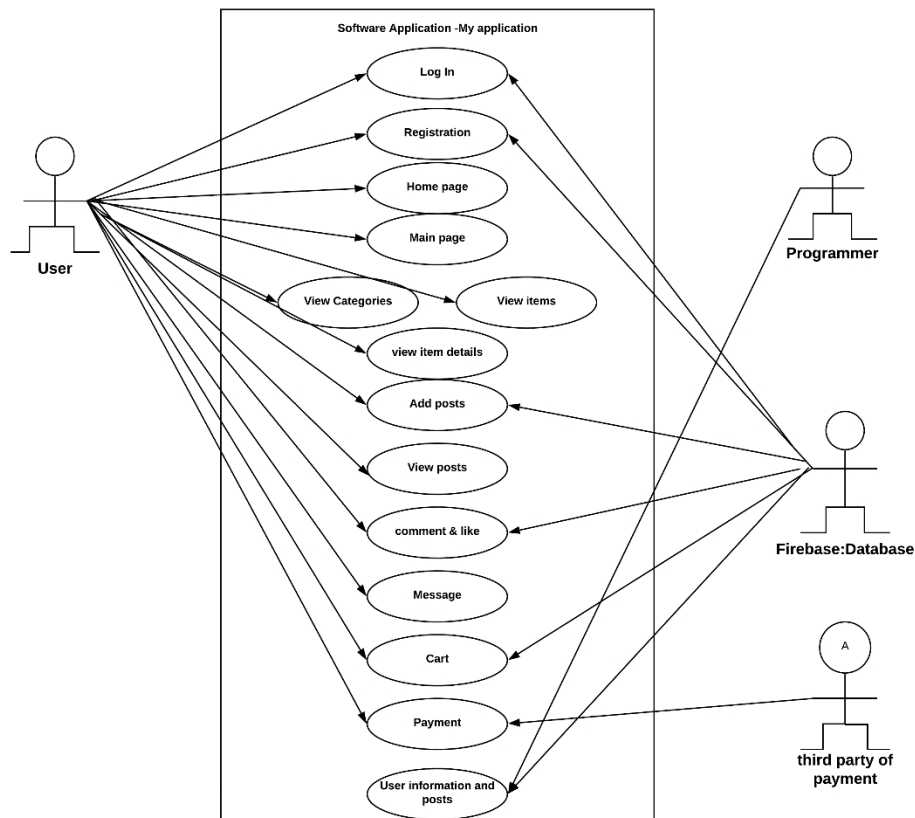
## 2.0 System

### 2.1. Requirements

#### 2.1.1. Functional Requirements

- The application is used for sharing and ordering things.
- Users must have an android mobile phone.
- The application must allow users to login and registration.
- The application must record the data of the registration and login in the firebase (database of this project).
- The application must allow users to post and view the post.
- The application must allow users to like and comment on the existing post.
- The application must allow users to message each other.
- The application must allow users to view the category and view the items and their detailed information.
- The application must allow users add items into their cart.
- The application must allow users make a payment.

### 2.1.1.1. Use Case Diagram



### 2.1.1.2. Requirement 1 <Splash Screen>

The requirement 1 is Splash Screen.

### 2.1.1.3. Description & Priority

This requirement indicated the name of the application before the first UI (Log in and registration). It only lasts 2000 milliseconds.

### 2.1.1.4. Use Case

#### Use Case: Splash Screen

##### Scope

The scope of this use case is to allow users to recognize the logo of the application.

##### Description

This use case describes the screen of the logo of the application.

##### Flow Description

##### Precondition



The user has a mobile phone and have the application downloaded.

The user starts the application.

### **Activation**

This use case starts when an actor (the user) has started this application.

### **Main flow**

1. The system identifies the splash screen of the application.

### **Exceptional flow**

E1 : <System errors cause the failure of the splash screen>

1. The system shows the failure of the splash screen.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 3 of the main flow. (See E2)

E2 : <System errors cause the failure of the splash screen after restarting the system>

1. The system shows the failure of the splash screen.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### **Termination**

The system presents the next Registration page (Start the Use case: Registration).

### **Post condition**

The system goes into a wait state.

## 2.1.1.5. Requirement 2 <User Registration>

The requirement 2 is User Registration.

## 2.1.1.6. Description & Priority

This requirement allows user to register an account by typing in the full name, address, email and password so that users could log in to their account and other functions can be triggered and used afterwards. This requirement is the most essential foundation of this system. Every user should register before they could start using this application.

## 2.1.1.7. Use Case

### **Use Case: Registration**

## Scope

The scope of this use case is to allow users to register an account in the application. So that the firebase could store the related information of this user.

## Description

This use case describes the registration process in the system. Users should register before they start using the application. The Firebase will store all the related information of users once they have registered. So that they could start the next use case: Log In to their account and trigger other functions.

## Flow Description

### Precondition

The user has a mobile phone and have the application downloaded.

The user starts the application.

### Activation

This use case starts when an actor (the user) has started this application.

### Main flow

1. The system identifies the registration of the application.
2. The actor (the user) starts the application.
3. The actor (the user) clicks "Register" button.
4. The system leads the user to the register page.
5. The actor (the user) types in their full name, address, email and password into the edit text box.
6. The actor (the user) clicks the "Sign In" button and waits the system to store his information into the database on Firebase. (See A1) (See A2)
7. The system shows the "registered" once it collects and stores the user information into the database on Firebase. (See E1)
8. The actor (the user) now registered and created an account in the system.

### Alternate flow

A1 : <One of the edit text boxes is empty: full name, address, email and password.>

1. The actor (the user) leaves a blank into one of four edit text boxes: full name, address, email and password.
2. The system shows "full name/ address/ email/ password is required".
3. The actor (the user) needs to make sure all the edit text boxes are not empty.
4. The use case continues at position 4 of the main flow.

A2: <The user has registered before.>

1. The system shows "The user has been registered".
2. The actor (the user) needs to click the Text View "Already have an account? Log in Here."
3. The use case continues at position 1 of the main flow of the use case: Log In.

**Exceptional flow**

E1 : <System errors cause the failure of the registration>

1. The system shows the failure of the registration.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 3 of the main flow. (See E2)

E2 : <System errors cause the failure of the registration after restarting the system>

1. The system shows the failure of the registration.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**

The system presents the next Log In page (Start the Use case: Log in).

**Post condition**

The system goes into a wait state.

#### 2.1.1.8. Requirement 3 <User Log In >

The requirement 3 is User Log In.

#### 2.1.1.9. Description & Priority

This requirement allows user to log in to their account which they have created in the use case: Registration. So that other functions can be triggered and used afterwards. This requirement is the second most essential foundation of this system. Every user should log in before they could start using this application.

#### 2.1.1.10. Use Case

**Use Case: Log In****Scope**

The scope of this use case is to allow users to log in to their account in the application. So that the firebase could match their information and lead users to their Main page which is the use case: Main page.

**Description**

This use case describes the Log in process in the system. Users should log in before they start using the application. The Firebase will find and match all the related information of users once they have logged in. So that they could start the next use case: main page and trigger other functions.

**Flow Description**

**Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

**Activation**

This use case starts when an actor (the user) have started this application and registered already.

**Main flow**

1. The system identifies the Login of the application.
2. The actor (the user) starts the application.
3. The actor (the user) types in email and password into the edit text box.
4. The actor (the user) clicks the "Log In" button and waits the system to match his information into the database on Firebase. (See A1)
5. The system shows the "Logged in successfully" once it finds and matches the user information into the database on Firebase. (See E1)
6. The system leads the user to main page.

**Alternate flow**

A1 : <One of the edit text boxes is empty: email and password.>

1. The actor (the user) leaves a blank into one of two edit text boxes: email and password.
2. The system shows "email/ password is required".
3. The actor (the user) needs to make sure all the edit text boxes are not empty.
4. The use case continues at position 4 of the main flow.

**Exceptional flow**

E1 : <System errors cause the failure of the Log In >

1. The system shows the failure of the Log In.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 3 of the main flow. (See E2)

E2 : <System errors cause the failure of the Log In after restarting the system>

1. The system shows the failure of the Log In.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**

The system presents the next Main page (Start the Use case: Main page).

**Post condition**

The system goes into a wait state.

#### 2.1.1.11. Requirement 4 <Main Page>

The requirement 4 is Main page.

#### 2.1.1.12. Description & Priority

This requirement allows user to choose the next function that they are willing to use between post and cart functions. So that other functions can be triggered and used afterwards. This requirement is one of the most essential requirements in this system. Every user should choose the next functions in this page.

#### 2.1.1.13. Use Case

##### **Use Case: Main page**

##### **Scope**

The scope of this use case is to allow users to choose the next function that they are willing to use between post and cart functions.

##### **Description**

This use case describes the main page in the system. Users should select one of the functions between post and Cart.

##### **Flow Description**

##### **Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

##### **Activation**

This use case starts when an actor (the user) has logged in to their account.

##### **Main flow**

1. The system identifies the main page of the application.
2. The actor (the user) clicks one of the buttons "Get start" and "Ready Go". (See A1) (See A2)
3. The system leads the user to either the home page or the Cart page. (See E1)

##### **Alternate flow**

A1 : <The user click "Get start".>

1. The actor (the user) clicks the “Get start” button.
2. The system leads the user to the home page.

A2: <The user clicks “Ready Go”.>

1. The actor (the user) clicks the “Ready Go” button.
2. The system leads the user to the Cart page.

### Exceptional flow

E1 : <System errors cause the failure of the Main Page>

1. The system shows the failure of the Main Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the Main page after restarting the application>

1. The system shows the failure of the main page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### Termination

The system presents the next page (Start the Use case: Home page or use case: Cart page).

### Post condition

The system goes into a wait state.

#### 2.1.1.14. Requirement 5 <Home page>

The requirement 5 is Home page.

#### 2.1.1.15. Description & Priority

This requirement allows user to choose the next function that they are willing to use between post (home) and Message as well as allows user to add a post. So that other functions can be triggered and used afterwards. This requirement is one of the common requirements in this system. Every user could choose the next functions in this page.

#### 2.1.1.16. Use Case

##### Use Case: Home page

##### Scope

The scope of this use case is to allow user to choose the next function that they are willing to use between post(home) and Message, and also allows user to add a post and view the posts. The user could sign out as well.

### Description

This use case describes the home page in the system. Users could select one of the functions among post(home), message, add a post to the home page and view the posts. And also the user could sign out.

### Flow Description

#### Precondition

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the "Get Start" button.

#### Activation

This use case starts when an actor (the user) have logged in to their account and click the "Get start" button.

#### Main flow

1. The system identifies the home page of the application.
2. The actor (the user) clicks one of the buttons "post" and "message" on the bottom navigation bar. (See A1) (See A2)
3. The system leads the user to either the post(home) page or the message page. (See E1)
4. The actor (the user) clicks "add" icon to add posts.
5. The system leads the user to the post page. (See E1)
6. The actor (the user) clicks on the post to view the detail of the post.
7. The system leads the user to the detail post page.
8. The actor (the user) clicks the "sign out" button.
9. The system leads the user to log in page.

#### Alternate flow

A1 : <The user clicks "Post" on the bottom navigation bar.>

1. The actor (the user) click "Post" on the bottom navigation bar.
2. The system leads the user to the Post (home) page.

A2: <The user clicks "message" on the bottom navigation bar.>

1. The actor (the user) clicks "message" on the bottom navigation bar.
2. The system leads the user to the Cart page.

**Exceptional flow**

E1 : <System errors cause the failure of the home page>

1. The system shows the failure of the home Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the home page after restarting the application>

1. The system shows the failure of the home page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**

The system presents the next page (Start the Use case: Post (home) page, use case: Cart page or use case: add post page).

**Post condition**

The system goes into a wait state.

**2.1.1.17. Requirement 6 <Add Post page>**

The requirement 6 is Add Post page.

**2.1.1.18. Description & Priority**

This requirement allows user to add a post by adding image, title and a content as well as showing it in the home page. This requirement is one of the common requirements in this system. Every user could choose to add a post in this page.

**2.1.1.19. Use Case****Use Case: Add Post page****Scope**

The scope of this use case is to allows user to add a post by adding image, title and a content as well as showing it in the home page.

**Description**

This use case describes the add post page in the system. Users could add a new post by choose an image, title and content and show it in the home page.

**Flow Description****Precondition**



The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Get Start” button and click “add post” icon.

### **Activation**

This use case starts when an actor (the user) has logged in to their account, click the “Get start” button and click the “add post” icon.

### **Main flow**

1. The system identifies the add post page of the application.
2. The actor (the user) clicks the “add post” icon in the home page.
3. The system leads the user to add post page.
4. The system shows the current user and current date in the add post page.
5. The actor (the user) needs to add an image, title and content.
6. The actor (the user) needs to click upload button. (See A1) (See E1)
7. The system leads the user to the home page.

### **Alternate flow**

A1 : <One of the fields is empty: image, title and content.>

1. The actor (the user) leaves a blank into one of three fields: image, title and content.
2. The system shows red icon besides the field that needs to be filled.
3. The actor (the user) needs to make sure all the fields are not empty.
4. The use case continues at position 6 of the main flow.

### **Exceptional flow**

E1 : <System errors cause the failure of the add post page>

1. The system shows the failure of the add post Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the add post page after restarting the application>

1. The system shows the failure of the add post page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### **Termination**

The system presents the next page (Start the Use case: Post (home) page).

#### **Post condition**

The system goes into a wait state.

### 2.1.1.20. Requirement 7 <View Post Detail page>

The requirement 7 is the View Post Detail Page.

#### 2.1.1.21. Description & Priority

This requirement allows users to click the post and view the detail of the posts. This requirement is one of the common requirements in this system. Every user could choose to view any existing posts.

#### 2.1.1.22. Use Case

##### **Use Case: View Post Detail Page**

#### **Scope**

The scope of this use case is to allows users to view any posts in the application.

#### **Description**

This use case describes the view post detail function in the system. Users could view the detail of the posts.

#### **Flow Description**

##### **Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Get Start” button and click “Post” icon.

The user has clicked any posts on the list.

##### **Activation**

This use case starts when an actor (the user) has logged in to their account, click the “Get start” button and click the “Post” icon.

##### **Main flow**

1. The system identifies post page of the application.
2. The actor (the user) clicks the “post” icon in the home page.
3. The system leads the user to post page which is the default page. (See E1)
4. The actor (the user) could choose other users or themselves’ post to view the detail of the posts in the system.

**Exceptional flow**

E1 : <System errors cause the failure of the view post detail page>

1. The system shows the failure of the View post detail Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the view post detail page after restarting the application>

1. The system shows the failure of the view post detail page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**

The system presents the next page (Start the Use case: Post (home) page).

**Post condition**

The system goes into a wait state.

**2.1.1.23. Requirement 8 <Comment>**

The requirement 8 is Comment Functionality.

**2.1.1.24. Description & Priority**

This requirement allows users to comment on the existing posts. This requirement is one of the common requirements in this system. Every user could choose to comment on other posts.

**2.1.1.25. Use Case****Use Case: Comment****Scope**

The scope of this use case is to allows users to comment on the posts.

**Description**

This use case describes the comment function in the system. Users could communicate with each other within the posts.

**Flow Description****Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.


The user has logged into the system.

The user has clicked the “Get Start” button and click “Post” icon.

### Activation

This use case starts when an actor (the user) has logged in to their account, click the “Get start” button and click the “Post” icon.

### Main flow

1. The system identifies post page of the application.
2. The actor (the user) clicks the “post” icon in the home page.
3. The system leads the user to post page which is the default page.
4. The actor (the user) can click on the comment icon  to leave a comment on the existing posts in the system. (See E1)

### Exceptional flow

E1 : <System errors cause the failure of the comment function>

1. The system shows the failure of the comment page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the comment page after restarting the application>

1. The system shows the failure of the comment page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### Termination

The system presents the next page (Start the Use case: Post (home) page).

### Post condition

The system goes into a wait state.

## 2.1.1.26. Requirement 9 <Like Functionality>

The requirement 9 is the like functionality.

### 2.1.1.27. Description & Priority

This requirement allows users to like the posts. This requirement is one of the common requirements in this system. Every user could like any posts.

### 2.1.1.28. Use Case

#### Use Case: Like Functionality

##### Scope

The scope of this use case is to allows users to like the posts.

##### Description

This use case describes the like function in the system.

##### Flow Description

##### Precondition

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.


The user has logged into the system.

The user has clicked the "Get Start" button.

##### Activation

This use case starts when an actor (the user) have logged in to their account, click the "Get start" button.

##### Main flow

1. The system identifies post page of the application.
2. The actor (the user) clicks the like icon  for any existing posts. (See E1)
3. The like icon would turn into grey if the actor (the user) clicks the like icon.

##### Exceptional flow

E1 : <System errors cause the failure of the like functionality>

1. The system shows the failure of the like functionality.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the like functionality after restarting the application>

1. The system shows the failure of the like functionality.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**

The system presents the next page (Start the Use case: Post (home) page).

**Post condition**

The system goes into a wait state.

## 2.1.1.29. Requirement 10 &lt;Message page&gt;

The requirement 10 is Message page.

## 2.1.1.30. Description &amp; Priority

This requirement allows users to message each other. This requirement is one of the common requirements in this system. Every user could choose to have a conversation with each other among users in this page.

## 2.1.1.31. Use Case

**Use Case: Message page****Scope**

The scope of this use case is to allows users to message each other.

**Description**

This use case describes the message function in the system. Users could communicate with each other among users.

**Flow Description****Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Get Start” button and click “Message” icon.

**Activation**

This use case starts when an actor (the user) have logged in to their account, click the “Get start” button and click the “Message” icon.

**Main flow**

1. The system identifies message page of the application.
2. The actor (the user) clicks the “message” icon in the home page.

3. The system leads the user to message page. (See E1)
4. The actor (the user) could choose other users to have a conversation in the system.

### Exceptional flow

E1 : <System errors cause the failure of the message page>

1. The system shows the failure of the message Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the message page after restarting the application>

1. The system shows the failure of the message page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### Termination

The system presents the next page (Start the Use case: Post (home) page).

### Post condition

The system goes into a wait state.

## 2.1.1.32. Requirement 11 <Cart (Category) page>

The requirement 11 is Cart page.

### 2.1.1.33. Description & Priority

This requirement allows user to view the categories of the shopping lists. This requirement is one of the most essential requirements in this system. Every user could view the category.

### 2.1.1.34. Use Case

#### Use Case: Cart page

#### Scope

The scope of this use case is to allows user to view the categories.

#### Description

This use case describes the cart (category) page in the system. Users could view the categories.

#### Flow Description

**Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Ready Go” button.

**Activation**

This use case starts when an actor (the user) have logged in to their account and click the “Ready Go” button.

**Main flow**

1. The system identifies the cart page of the application.
2. The actor (the user) clicks buttons “ready Go” in the home page.
3. The system leads the user to the cart page. (See A1)
4. The actor (the user) could click on each category to view the items.
5. The system leads the user to the item page. (See E1)

**Alternate flow**

A1 : <Users don’t want to View the categories>

1. The actor (the user) clicks the “return” button.
2. The system leads the user to the main page.
3. The use case continues at position 1 of the main flow of main page.

**Exceptional flow**

E1 : <System errors cause the failure of the cart>

1. The system shows the failure of the cart Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the cart page after restarting the application>

1. The system shows the failure of the cart page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

**Termination**



The system presents the next page (Start the Use case: item page).

#### **Post condition**

The system goes into a wait state.

#### 2.1.1.35. Requirement 12 <item page>

The requirement 12 is item page.

#### 2.1.1.36. Description & Priority

This requirement allows user to view the items in each category. This requirement is one of the most essential requirements in this system. Every user could view the items.

#### 2.1.1.37. Use Case

##### **Use Case: Item page**

##### **Scope**

The scope of this use case is to allows user to view the items.

##### **Description**

This use case describes the item page in the system. Users could view the items.

##### **Flow Description**

##### **Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Ready Go” button.

##### **Activation**

This use case starts when an actor (the user) have logged in to their account and click the “Ready Go” button.

##### **Main flow**

1. The system identifies the item page of the application.
2. The actor (the user) clicks buttons “ready Go” in the home page.
3. The system leads the user to the cart page. (See A1)
4. The actor (the user) clicks the “add to cart” button or click on the items to view the detail of the items.

5. The system leads the user to the add to cart / view item detail page. (See E1)

### **Alternate flow**

A1 : <Users don't want to view the items>

1. The actor (the user) clicks the "return" button.
2. The system leads the user to the main page.
3. The use case continues at position 1 of the main flow of main page.

### **Exceptional flow**

E1 : <System errors cause the failure of the item>

1. The system shows the failure of the item Page.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the item page after restarting the application>

4. The system shows the failure of the item page.
5. The actor (the user) needs to report to the end user and waits to the respond from the end user.
6. The use case continues at position 1 of the main flow.

### **Termination**

The system presents the next page (Start the Use case: item detail page).

### **Post condition**

The system goes into a wait state.

#### **2.1.1.38. Requirement 13 view item detail page>**

The requirement 13 is the view item detail page.

#### **2.1.1.39. Description & Priority**

This requirement allows user to view the detail of the items. This requirement is one of the most essential requirements in this system. Every user could view the detail of the items.

#### **2.1.1.40. Use Case**

#### **Use Case: View Item Detail page**

#### **Scope**

The scope of this use case is to allows user to view the detail of the items.

### Description

This use case describes the view item detail page in the system. Users could view the item details.

### Flow Description

#### Precondition

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Ready Go” button.

#### Activation

This use case starts when an actor (the user) have logged in to their account and click the “Ready Go” button.

#### Main flow

1. The system identifies the item page of the application.
2. The actor (the user) could click on each item.
3. The system leads the user to the view item detail page to view more details.

#### Exceptional flow

E1 : <System errors cause the failure of the view item detail>

1. The system shows the failure of the view item detail.
2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the view item detail after restarting the application>

1. The system shows the failure of the view item detail.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

#### Termination

The system presents the next page (Start the Use case: add to cart page).

**Post condition**

The system goes into a wait state.

## 2.1.1.41. Requirement 14 &lt;Add to Cart page&gt;

The requirement 14 is add to cart page.

## 2.1.1.42. Description &amp; Priority

This requirement allows user to add items to their cart, in order to make a payment later.

## 2.1.1.43. Use Case

**Use Case: Add to Cart page****Scope**

The scope of this use case is to allows user to add items to their cart.

**Description**

This use case describes the add to cart page in the system.

**Flow Description****Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Ready Go” button.

**Activation**

This use case starts when an actor (the user) have logged in to their account and click the “Ready Go” button and click category.

**Main flow**

1. The system identifies the add to cart page of the application.
2. The actor (the user) clicks buttons “add to cart” in the item page.
3. The system leads the user to the add to cart page. (See E1)

**Exceptional flow**

E1 : <System errors cause the failure of the add to cart page>

1. The system shows the failure of the add to cart page.

2. The actor (the user) needs to restart the application.
3. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the add to cart page after restarting the application>

1. The system shows the failure of the add to cart page.
2. The actor (the user) needs to report to the end user and waits to the respond from the end user.
3. The use case continues at position 1 of the main flow.

### **Termination**

The system presents the next page (Start the Use case: payment page).

### **Post condition**

The system goes into a wait state.

#### 2.1.1.44. Requirement 15 <Payment page>

The requirement 15 is payment page.

#### 2.1.1.45. Description & Priority

This requirement allows user to make a payment for the items in their cart by using their credit cards. This requirement is one of the most essential requirements in this system. Every user could pay the items in the cart in this page.

#### 2.1.1.46. Use Case

### **Use Case: Payment page**

#### **Scope**

The scope of this use case is to allows user to make a payment for the items in their cart.

#### **Description**

This use case describes the payment page in the system. Users could make a payment by using their credit card for the items in their cart.

#### **Flow Description**

#### **Precondition**

The user has a mobile phone and have the application downloaded.

The user starts the application.

The user has an account in the system.

The user has logged into the system.

The user has clicked the “Ready Go” button.

The user has added some items in the cart.

### Activation

This use case starts when an actor (the user) have logged in to their account and click the “Ready Go” button and click “payment” button.

### Main flow

4. The system identifies the payment page of the application.
5. The actor (the user) clicks buttons “payment” in the cart page.
6. The system leads the user to the payment page. (See E1)
7. The actor (the user) need to type in their credit card information.
8. The actor (the user) clicks the “continue & pay” button.
9. The system leads the user to the payment processing page. (See A1)
10. The system leads the user to the cart page. (See E1)

### Alternate flow

A1 : <Payment doesn't go through successfully>

1. The system shows error when click the “continue & pay” button.
2. The actor (the user) asked to use another payment method.
3. The use case continues at position 5 of the main flow.

### Exceptional flow

E1 : <System errors cause the failure of the payment>

4. The system shows the failure of the payment Page.
5. The actor (the user) needs to restart the application.
6. The use case continues at position 1 of the main flow. (See E2)

E2 : <System errors cause the failure of the payment page after restarting the application>

4. The system shows the failure of the payment page.
5. The actor (the user) needs to report to the end user and waits to the respond from the end user.
6. The use case continues at position 1 of the main flow.

### Termination

The system presents the next page (Start the Use case: cart page).

### Post condition

The system goes into a wait state.

### 2.1.2. Data Requirements

From the perspective of the tester (the user):

- The application requires users' personal information to register, login and pay when using the app.
- The application requires users to provide the full name, address, email, password, and payment information.
- The application requires users to add image, title, and content. The image must from the users' phone gallery or from the legal platform.

From the perspective of the database (Firebase):

- All the information from users is stored in my Firebase, only the programmer can access to them as the author of the project.
- The programmer can change the password of users inside the Firebase.
- The programmer is required to secure all the information that users provide when using the application.
- All the data provided in the application are from public legal recourses such as google searching engine.
- No special data required for this project that need to be ask for permission to use.

### 2.1.3. User Requirements

Based on the input from the testers (users), the following user requirement are needed.

- Users are required to have mobile phones/ or they have emulator installed on their laptop.
- Users are required to have android system in their mobile phones.
- Users are required to provide full name, address, emails and passwords to the application for the registration.
- Users are required to provide the emails and the matching passwords to the application for the log in.
- Users are required to add image, title and content to the application for adding posts.
- Users are required to provide their credit cards information to the application for the payment.

### 2.1.4. Environmental Requirements

From the perspective of the testers (the user):

- The users must have a mobile phone with android system.

- The users are required to run the application within in well internet condition, either under WIFI or using Cellular data.
- The users are required to provide their credit card information for the payment function under a secure environment, in order to avoid the loss of important personal data.

### 2.1.5. Usability Requirements

Usability Test:

- Make enough usability tests before letting users to download the application.
- Make sure that there are at least 10 random participants take the usability tests before it launches.

From the reflective of the testers (the user):

The application must meet four requirements in order to improve the usability: **easy to use, easy to learn, attractiveness and easy to operate. Achieve the integrity of the application.**

- Make sure that the system is easy to use by arranging the layouts of the UI in a common logical and manner.
- Make sure that the customer experience is well by having a customer support page to connect the customer with the programmer, so that improve both the satisfaction of customers and usability of the application.
- Make sure the UI of the application is attractive to the potential customers by utilizing the simple design.
- The application can be downloaded on every mobile phone with an android system.
- Keep the integrity of the application, rather than focus on the complexity of the application.

## 2.2. Design & Architecture

Java was the main language of this project. So, there must have few Java classes for each function and also the corresponding xml files for their layouts. The design and architecture that I have chosen was the very basic and simple one. The first class that the user would see is the Log In page, then the users could follow the texts and buttons on the UI to choose next function. The theme colour of the application is pink, in order to create a relaxed and cute atmosphere of the project.

Reducing the complexity of the architecture of the project, in order to facility the speed of the project, as well as improve the usability of the application. However, simple design would also cause a poor UI to the users, that might cause the satisfaction of the users' experiences at the level of visualisation. The project could not fully implement both the fancy UI and the simple architecture, so that the project would have to choose one of these two as its design.

Considering the unwell condition of the hardware and software on my laptop and also the poor condition of the internet environment, I have chosen to implement the simple architecture as the design of the project, in other words, the UI would be less fancy than similar applications in the market.



### 2.3. Implementation

The database is the main foundation of this project. The Firebase was chosen due to its flexibility and Simplicity. Few dependencies need to be added in the "Gradle script" as shown in the figure 3.

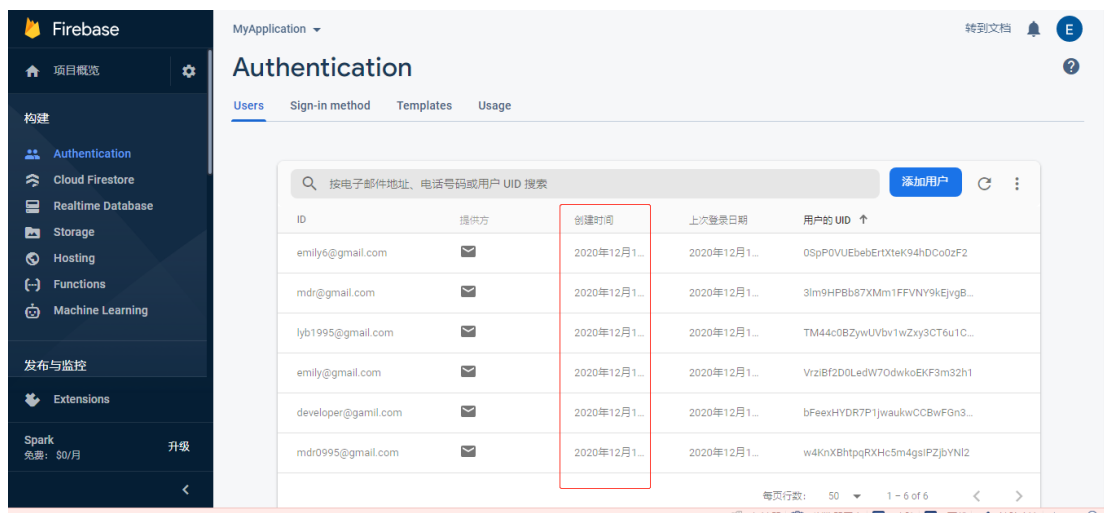
```
implementation 'com.google.firebase:firebase-bom:25.12.0'
implementation 'com.google.firebase:firebase-analytics'
implementation 'com.google.firebase:firebase-auth:16.0.5'
implementation 'com.google.firebase:firebase-database'
implementation 'com.google.firebase:firebase-storage:16.0.3'
```

Figure 3.

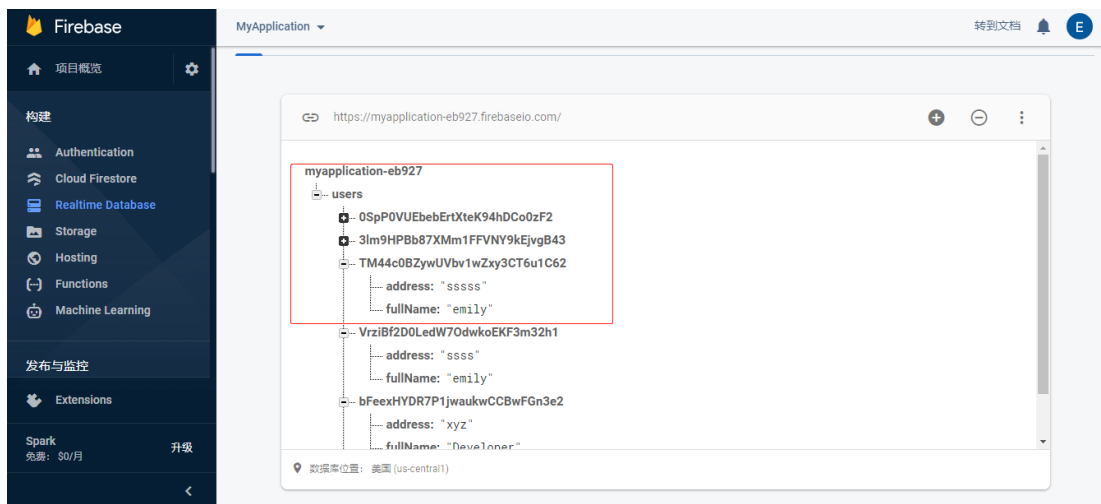
There are a lot of versions of firebase, so it caused many bugs at the beginning of the project. As shown in the diagram below. I have set up the firebase once in the early stage of the project.



However, it kept showing errors. Therefore, I have deleted this project in my Firebase and created a new one in my Firebase as shown below.

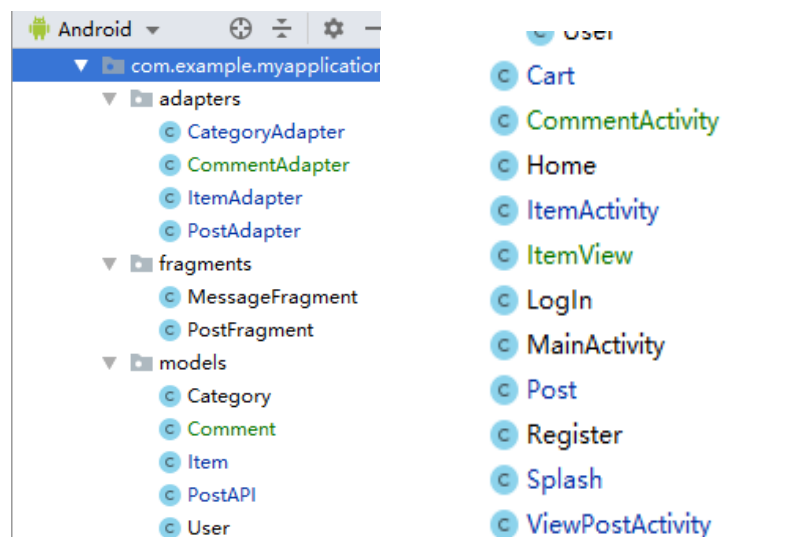


In addition, this time I have linked the storage and the real-time database to the project in android studio. Therefore, in my Firebase, I could also view users' address and full name not only the email and password.



After setting up the database, the project has started.

The main functions are registration, login, message, post, view posts, comment, like, item, view item and payment. There are 22 java classes. They are lists as shown below:



The message fragment and post fragment Java classes are used for the bottom navigation bar. The adapter Java classes were used to implement the recycler View, card View. Adapter is used to link between list of objectives with the list of UIs in general. For this specific project, adapter is used to gather posts, categories, comments, and items. The models are used for making a package, in order to allow the functions in Post, category, item and Registration/ Login Java classes to get the correct information (images, title, content for post page; address and full name for registration page).

```
public class PostAPI {
    String key, date, title, content, imageUrl;
}

public PostAPI() {
}

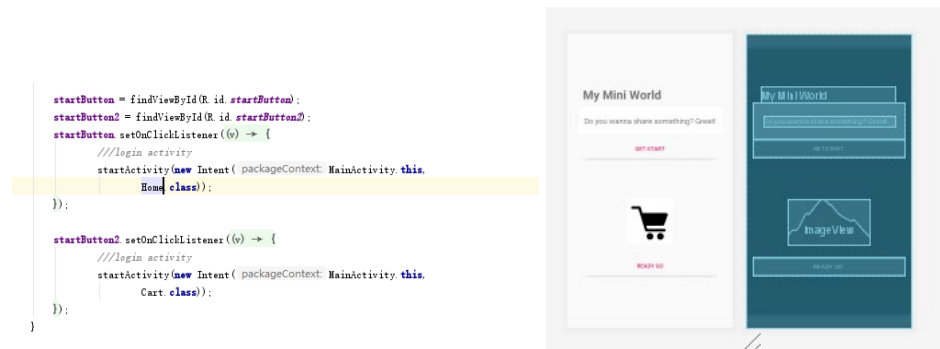
public PostAPI(String date, String title, String content, String imageUrl) {
    this.date = date;
    this.title = title;
    this.content = content;
    this.imageUrl = imageUrl;
}
```

```
public class User {
    String fullName, address;
}

public User() {
}

public User(String fullName, String address) {
    this.fullName = fullName;
    this.address = address;
}
```

Other Java classes are used for their own functions regarding their names. For example, Main activity Java class includes two buttons which links to the home page and cart page. In addition, there are corresponding xml file in the layout package in “res”.



Moreover, I found one function was very interesting – show the current username and today’s date. This one was completely new to this project. Thus it was also a challenge for me as well. This was the user and postAPI java classes that I have mentioned above. These two packages were already stored into the Firebase. So that I could call it by using these codes.

```
database = FirebaseDatabase.getInstance();
myRefUser = database.getReference().child("users").child(fAuth.getId());
myRef = database.getReference().child("posts").child(fAuth.getId());

String imageUrl = uri.toString();
PostAPI post = new PostAPI(date, title, content, imageUrl);

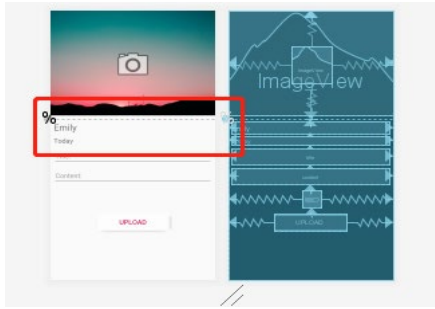
// ... (code for saving post to database) ...

myRefUser.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.exists()) {
            User user = dataSnapshot.getValue(User.class);
            currentUserTextView.setText(user.getFullName());
        }
    }

    @Override
    public void onCancelled(DatabaseError error) {
        Toast.makeText(context, error.toString(), Toast.LENGTH_SHORT).show();
    }
});

@Override
public void onSuccess(Uri uri) {
    String imageUrl = uri.toString();
    PostAPI post = new PostAPI(date, title, content, imageUrl);
    myRef.child(UUID.randomUUID().toString()).setValue(post);
}
});
```

Then it would show the current username and today’s date in the UI.



The biggest challenge that I have met during the implementation phase was the unreasoned error. Those errors were not shown when you build the project. But it would stop the application runs smoothly in the emulator. I have found that these errors happened due to the redundancy of the codes as well as the versions of different settings, configures in the “build gradle”. For example, this bug caused the app kept stopping, but the codes were looks fine.

```

//android:runtime: fatal exception: main
Process: com.example.myapplication, PID: 4825
android.content.ActivityNotFoundException: Unable to find explicit activity class {com.example.myapplication/com.example.myapplication.ItemView}: have you declared this activity in your AndroidManifest
  at
  
```

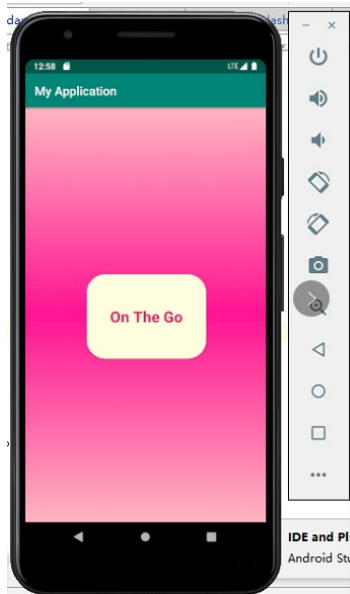
To solve these issues, I have figured out several methods. The first method is to follow the prompt from the android studio. Mostly, it was helpful. Secondly is trying to write codes slowly. Run them often, so that I would know which part of the codes caused problem. This method was quite useful for me. The third method is that when I have written too many codes without running the project, I would de-functioned (put some codes as comments) each complicated parts of codes that might cause the problem. This method would help me to find which function or which part of codes caused the problem. This method is useful as well, but it took longer time than the first method.

Moreover, for the different versions of settings, the only method is to try different versions, in order to find the one that works with my android studio and laptop. I would always make a record of the last version in case I need to go back to that version. This would help me to find the most coordinated versions of setting without loss of the essential ones.

## 2.4. Graphical User Interface (GUI)

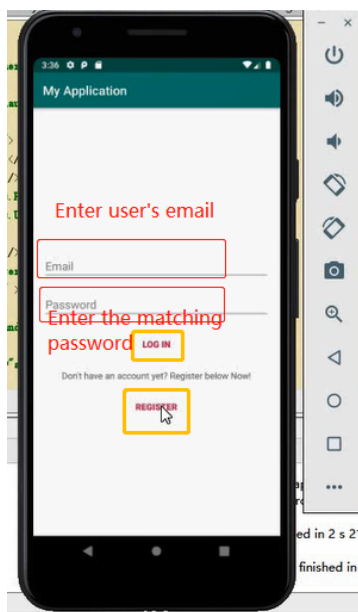
### Splash Screen:

- Show the logo of the application when the user starts the application.



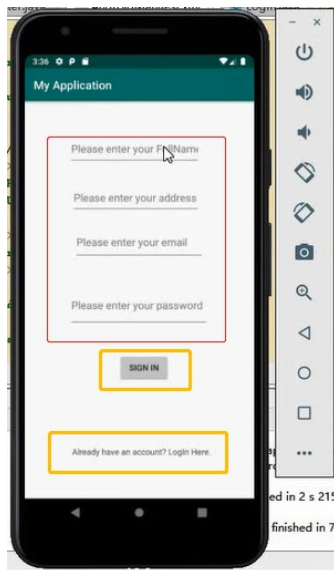
### Log In page:

- Users could type in the email and the matching password, then click the “Log In” button to log into the account.
- User could also click the “register” button if this user doesn’t have an account yet.
- User will see a progress bar after clicking the button.



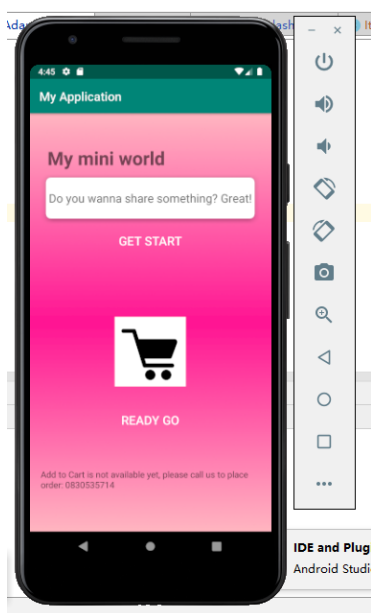
### Registration page

- Users could type in their full name, address, email and password to create an account.
- Users could click sign in button to create an account.
- User will see a progress bar after clicking the button.
- User could also go back to the login page by clicking the text: “Already have an account? Click here below.”



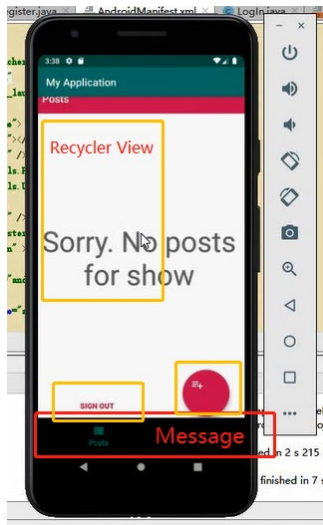
### Main page

- Users could see a card view says: “DO you wanna share something? Great”.
- Users could also see a cart image.
- Users could click either “get start” or “Ready Go” button.



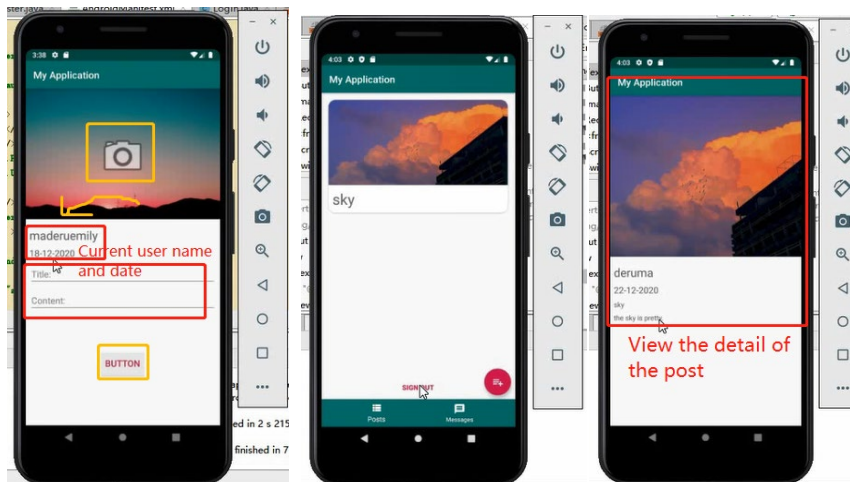
### Home page

- Users could see the post and message option on the bottom navigation bar.
- Users could see “Sorry no post for show”.
- Users could sign out by clicking the “Sign out” button.
- Users could click add post icon to go to post page.



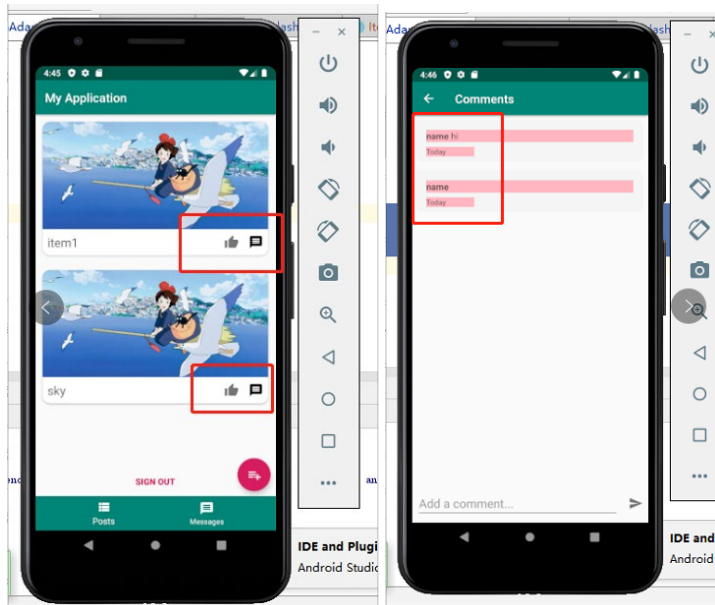
### Post page

- Users could see the current username and today's date.
- Users could click the photo icon to add image.
- Users could fill in the title and content.
- Users could upload by clicking the “upload” button.



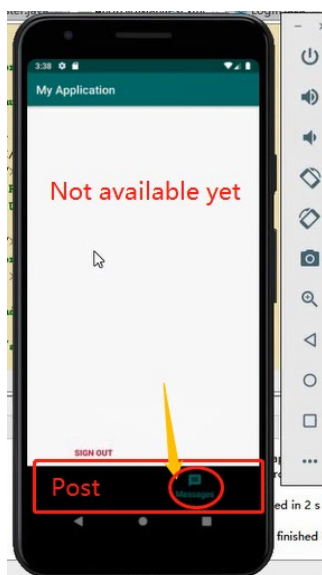
### Comment & like:

- User could comment and like any existing posts.



### Message page

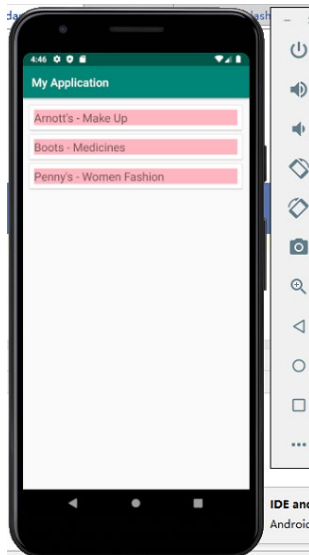
- Users could message with each other.



### Cart page/category page

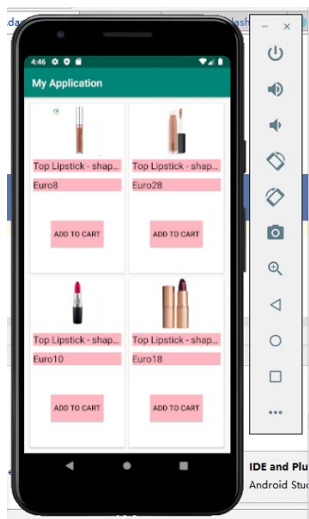
- Users could view the categories.





### Item page & Item detail page

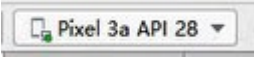
- Users could view the items and view the item details.



### Payment page and message page (Not available).

## 2.5. Testing

The only testing tool is the android emulator within the Android Studio. However, there are a lot of versions of SDK and emulators, so they were causing many errors at the beginning. The most stable

one that I am using currently is Pixel 3a API 28. . Even though this one is the most stable one for my android studio and my laptop, it quite slows and shut down sometimes without reasons.

From the user's respect:

- Due to the lockdown period of Ireland, the participants were hard to find and gather.
- Thus, there were only three participants have been chosen near me for testing.
- There was a short text message and a consent form have been sent to them previously. These are attached in the appendix.
- After the confirmation of their consent, they utilized the emulator on either their laptop or mine to test the usability of the application.
- All three participants have proposed that the usability of available functionalities currently is work perfectly. However, the UI is not fancy enough to them.

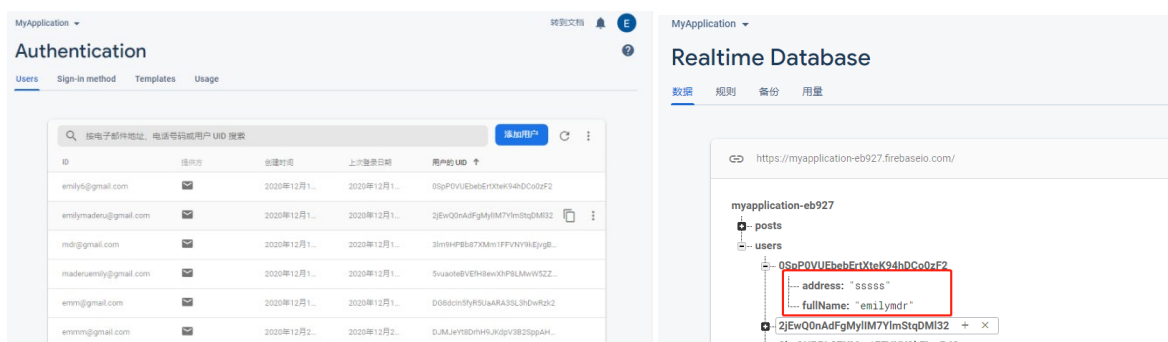
Testing results detail:

- All three participants were able to see the splash screen.
- All three participants were able to register accounts.
- All three participants were able to log in to their accounts.
- All three participants were able to view the main and home page.
- All three participants were able to see the post lists.
- All three participants were able to add posts.
- All three participants were able to like and comment on the posts.
- All three participants were able to click two buttons in main page. (get started & ready go).
- All three participants were able to view the categories, item lists and item details.

Even though, three participants are not enough to provide a high possibility for getting the accurate results, the testing results were still considered as success, regarding the feedback from all participants.

## 2.6. Evaluation

The project was evaluated by running the project through the emulator. And all log in and registration information of users would be stored into my Firebase. And all the tests so far were successful. They were all showed in my Firebase.



Moreover, the testing results of all three participants indicated positively. All available functionalities were implemented successfully. Thus, the prototype has achieved basic goals at the very beginning of the project. However, there are still few vacancies left for the further development and research.

The main advantage of this project was that the available functions are all fully functioned. Because for each single function, it caused me so much time to debug errors and fix issues of the software itself. Thus, these three available functions in this application have built a solid foundation for the further. Even though it has not met the objective 1 in the proposal, it still made a good start of the project. I have learnt so much and solved many bugs by myself, thus this project not only just building some functions in the application, but also gives me the ability to do better in the late stage of this project. I have the confident to do much better in the further development of this project.

The main disadvantage that I have mentioned in the “technology” section above was caused by the drawbacks of Java. It contains more codes and slows down my laptop. My android studio and the emulator always shut down or stuck by itself without reasons. Meanwhile, there are always some errors without showing, thus, I have to figure them out by myself. Most of them were caused by the collision between some codes because I learned these codes from many different sources. Therefore, some of them are the same but written in different ways.

### 3.0 Conclusions

This mobile application provides a platform for users to communicate others and order things from the internet. This application is not only a typical online shopping product, but also allow user to help each other. Competitors in the market have provides the products and the delivery is a plus service based on the products, however, this application makes the delivery service as the product, so that the local users could get things from multiple stores as well as “be” the delivery guy for other users. Therefore, especially in this hard time and social distancing environment, this application could make the local users’ life happier and much easier.

### 4.0 Further Development or Research

Overall, the most important objective for further development is to implement the message and the payment functions, in order to make the application fully works so that it would be ready to launch to the market.

Moreover, develop the UI, in order to make the application looks nicer for users, so that the application could attract more potential users by the fancy front end. As well as developing more unique functions.

From my side, I can improve more in the project plan and implementation phases in the future. I am not able to achieve all three initial objectives proposed in the beginning, due to the imperfect project plans. For the implementation phase of the project, I have met a lot of bugs and confusion during that phase, thus, in the further development, I could be more professional and more effective.

In conclusion, for the business side, this application is not the very unique one, but it does provide some unique reflections compared with competitors such as Just eat (businessofapps, 2020) in the market. It provides an idea of the combination between delivery services and social media platforms. Therefore, the application can be developed further and deeper at both vertical and horizontal level, in order to occupied more places in the market.

## 5.0 References

businessofapps, 2020. *www.businessofapps.com*. [Online]  
Available at: <https://www.businessofapps.com/data/just-eat-statistics/>  
[Accessed 13 April 2020].

Hindi, D., 2020. *mobile-app-development-trends*. [Online]  
Available at: <https://buildfire.com/mobile-app-development-trends/>  
[Accessed 13 11 2020].

Max Roser et al., H. R. a. E. O.-O., 2015. *Internet*. [Online]  
Available at: <https://ourworldindata.org/internet>  
[Accessed 21 12 2020].

STEVENS, E., 2018. *what-is-the-difference-between-a-mobile-app-and-a-web-app*. [Online]  
Available at: <https://careerfoundry.com/en/blog/web-development/what-is-the-difference-between-a-mobile-app-and-a-web-app/>  
[Accessed 21 12 2020].

Link to the **midpoint** video presentation: [https://youtu.be/WA\\_-jPi1hw](https://youtu.be/WA_-jPi1hw)

**Link to the final code repository:**

[https://github.com/MADERUEmily/My\\_Application\\_On\\_The\\_Go](https://github.com/MADERUEmily/My_Application_On_The_Go)

## 6.0 Appendices

This section should contain information that is supplementary to the main body of the report.

### 6.1. Project Plan



# National College of Ireland

## Project Proposal

< Mobile Application – On The GO>

08/11/2020

Business Information System

2020/2021

Student Name: DERU MA

Student Number: x17466626

Student Email: emilymaderu@icloud.com

### Contents

<a href="#">1.0</a>	<a href="#">Objectives</a>	<a href="#">1</a>
<a href="#">2.0</a>	<a href="#">Background</a>	<a href="#">1</a>
<a href="#">3.0</a>	<a href="#">Technical Approach</a>	<a href="#">1</a>
<a href="#">4.0</a>	<a href="#">Special Resources Required</a>	<a href="#">1</a>
<a href="#">5.0</a>	<a href="#">Project Plan</a>	<a href="#">1</a>
<a href="#">6.0</a>	<a href="#">Technical Details</a>	<a href="#">1</a>
<a href="#">7.0</a>	<a href="#">Evaluation</a>	<a href="#">1</a>

## 1.0 Objectives

The proposed project will allow users to order different goods in different stores at the same time. Users could add items on the application to their carts and be able to select a time slot as well as

make a payment through the application. It will not be like typical online shopping applications in the market, it will be more like users could “hire” someone who will buy and deliver the items to them incidentally. The application also will allow users to post, chat and trade with each other. Therefore, they could help each other through posts and messages. The application provides a platform for users to get items much quicker and easier.

**Objective 1:** Aim to achieve login, registration, message, comment, post, Cart, and payment function before midterm.

**Objective 2:** Aim to modify payment function and try to make it innovative. For example, could I give users certain amount loan every month, so that they could pay afterwards or pay through installment, in order to promote their usability?

**Objective 3:** Be innovation and Do more investigation. Try to make a unique function for my project.

## 2.0 Background

The world is going fast and crazy, time is really precious and valuable to everyone. Therefore, on-demand and mobile wallet are two popular trends in today's world, especially in this difficult time that everyone suffered under. So, my rationale is trying to make users' life easier, and try to connect them together, provide a platform for them to solve their problems regarding goods. There are a lot of competitors in the market, therefore, I need to propose more innovative and unique functions in this area during the project. <https://buildfire.com/mobile-app-development-trends/>

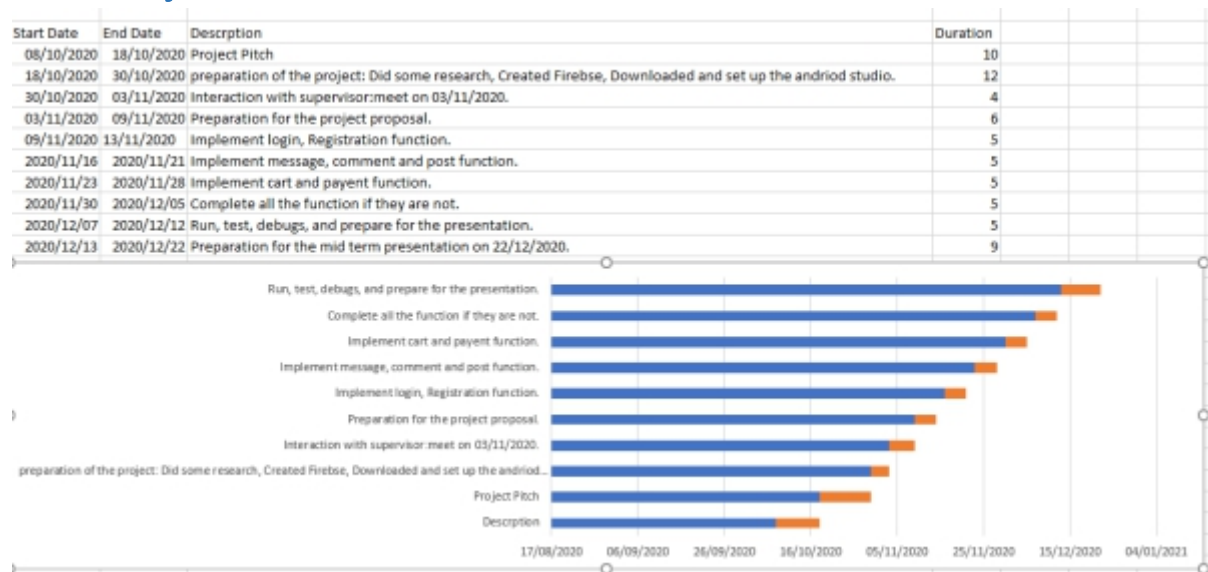
## 3.0 Technical Approach

I will be developing this project through Android Studio and built using Java. I will use online resources for study purpose. For example, Udemy, YouTube, Udacity.

## 4.0 Special Resources Required

No special resources.

## 5.0 Project Plan



## 6.0 Technical Details

Build an android application is cheaper than IOS application, so I have decided to build an android application. The main technology used in this project was Java, the toolkit used was Android Studio. Android studio has its own SDK tools, so the Android SDK were used in this project for the emulators. The reason that I have chosen Java rather than Kotlin as follows:

- I am familiar with Java.
- Java is much easier to understand and learn.
- Android Studio itself is built on Java.
- Java are much more compact and lighter than Kotlin.
- From the perspective of results, Java leads to a faster app experience.

Even though Java is more suitable for this project, it has some disadvantages as well. The major two drawbacks where Java contains more codes so that increase the chance of bugs and errors, as well as Java requires more memory so that tend to be slower. These two disadvantages have affected the project in some aspects, it would be mentioned in the evaluation section of this report.

Moreover, I have used YouTube, Udemy and StackOverflow as my technical assistant and support during the project. There are a lot of useful and effective tutorials on these platforms that helped me to build this application. And also helped me to solve many issues raised during the project. There is one other platform called Udacity, it also contains some tutorial regarding android studio. I have learnt few courses on Udacity before, they were quite beneficial, thus I might use it as my technical support as well in the later stage of the

project. will be using Java to develop the application and built on android studio. I will be using Firebase for the database that store login and registration information.

## 7.0 Evaluation

I will invite 5 - 10 participants to do the usability test before the midterm, and i will be send them an email within the consent form, so that there will be no ethic risks during this project. In addition, i will only use data that published on public resources which are already be consented to use in the public.

For the show case and midterm presentation, i will be using the virtual android phone emulator to show the application through android studio.

DERU MA

---

Signature

## 6.1. Reflective Report

### Software Project Reflective Report

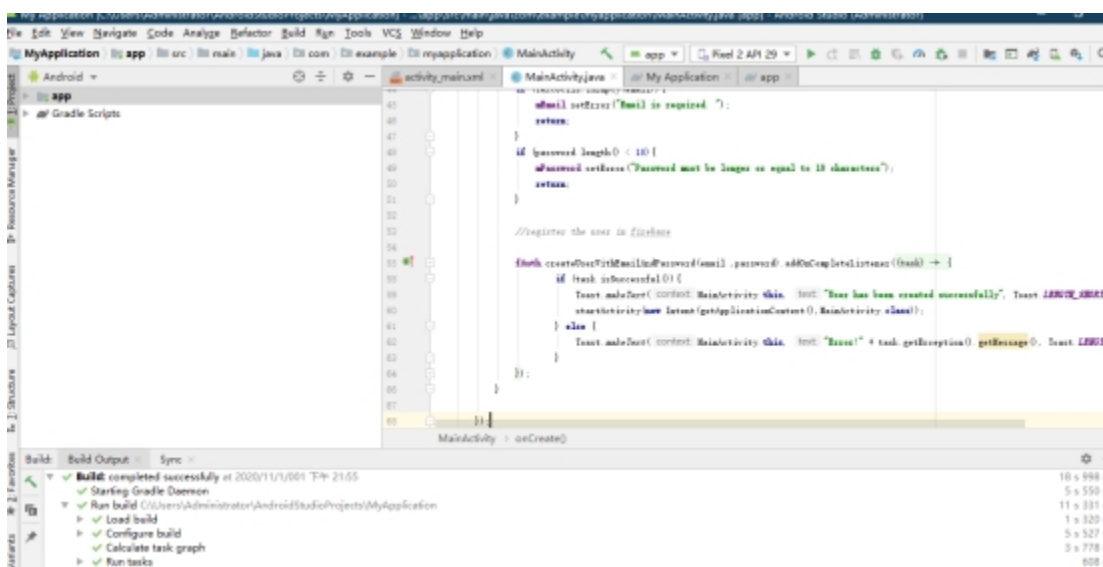
#### Month 1 (October):

- In the first month, first, i have submitted the project pitch for this project. I am going to build a mobile application through Android Studio.
- Before the project pitch, i have did some research online, i have checked some resources and classes online to give myself an idea how far could i reach or how much effort should i put if i am going to complete this project.
- After the general research, i have made the video and submitted on time.
- While waiting the allocation of the supervisor, i have connected my application on Android Studio with Firebase. Because no matter what application i am going to make, database is always an essential element to the project.





- And also I have simply created an Registration Activity, on one hand, I wanted to know my ability on Android Studio. On the other hand, I needed to check the Firebase connection.



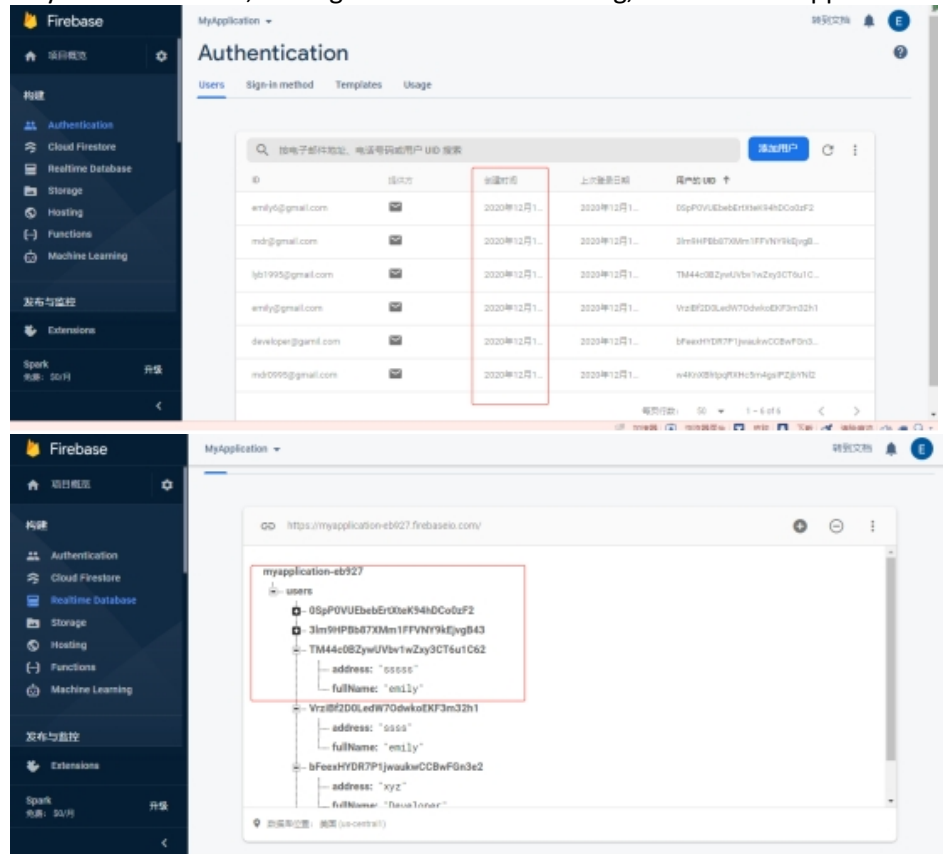
- After I got the supervisor list, I have contacted with my supervisor. She has confirmed that my project pitch passed and arranged a meeting next week on 03/11/2020.

**Month 2 + Month 3(December + November):**

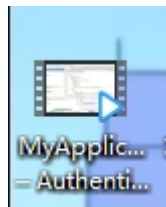
- My internet and laptop condition were really bad during the lockdown period, and I have been working all the way through the entire lockdown period. These weren't excuses for delivering the monthly report at this stage, but they were the reason for it. However, even though it was super difficult for me, I still tried my best to deliver this project, the following would show what have I done so far till today.
- I used to watch tutorials on YouTube for my login and register part. And I have done it in the first month, however, it was simple and not suitable for the project, so I have redone these parts by following few tutorials on Udemy (I bought few course).
- The first challenge that I have met was that the tutorial used "Fire store" on Firebase. However, after I have set up all the firebase, authentication and stores in android studio, I realized that it changed from fire store to the real-time database. They were using different

codes, but there were a lot of examples and tutorials online, so I have overcome this challenge.

- The next challenge was regarding android studio emulator. Due to my poor internet condition and poor laptop hardware, the software always shutdown by itself, so does the emulator. I have tried many methods to solve this problem, I have tried to redo the whole project and reset up the database and tried different emulators. They were all failed.
- As you can see here, the login function was working, but then it stopped working since then.



- The above image showed my second firebase.
- So, at this point, I have successfully build the login, register, and post function, however, the emulator doesn't work, so I need to focus on it before the mid presentation. And also focus on the technique report as well. I have made a short video on demonstrating the emulator when it worked, in case, there were something happened on the mid presentation.

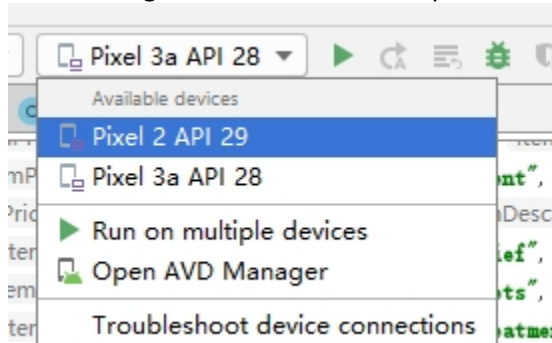


#### Month 4(February):

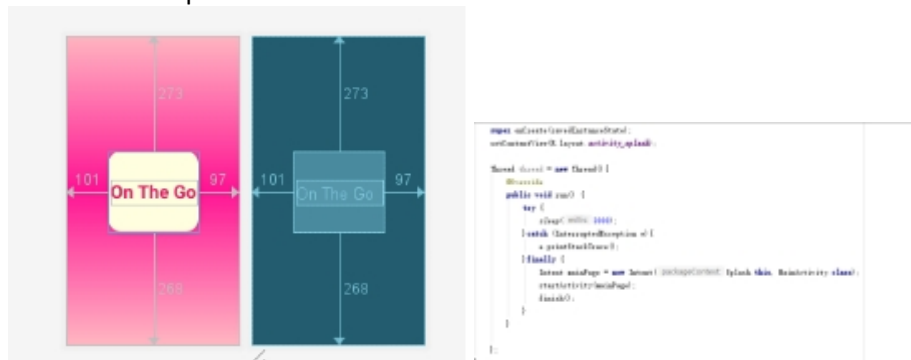
- In this month, I have focused on the project showcase and profile. Because I am partly working in Transpoco which is the company that I did my Internship. I have been working in

Transpoco after the Internship until now. Thus, I have to manage the balance between my project and work.

- So, this month, I was fixing the remaining problem from the mid-point. The first one is the emulator. I had the problem with the emulator before the mid-point and I got lucky that it was working when I filmed the mid presentation.



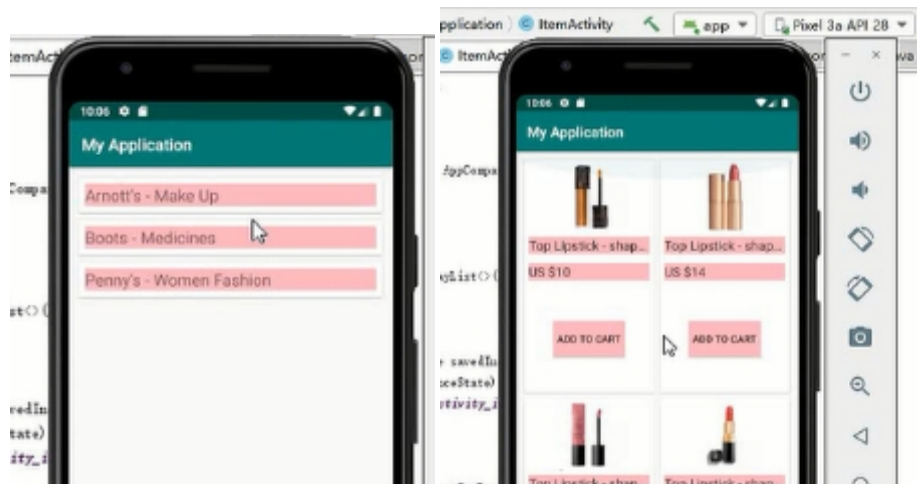
- I have tried to download different emulator and tried to wipe the data from it. They were not working at all. I have tried to use an actual android phone. It was working but the device was not permanently available. Thus, i had to fix the emulator issue.
- Finally, it worked with "Pixel 3a API 28." I think the problem was the conflict between my laptop and the newest emulator. My laptop cannot hold the newest version to run. Thus, one of the old versions worked for me.
- I have done a splash function.



- Unfortunately, it wasn't working. I am currently working on this. As well as doing some research on payment function. The goal of my app was to add items in the cart and make the payment. As well as chatting function. However, these three functions were quite hard to implement.
- I have tried to find some source to learn but they are too complicated (Like some of them created based on an existing project). Thus, I want to finish some easier functions that could work on time.

#### Month 5 + 6(March + April):

- I have started coding the cart page - using the Recycler View and Card View to show the categories and items. The process was taking nearly a whole month. I learned the code from different sources. So, it was difficult to put them together. The blow prototype was the result.



- 
- I have also tried to fix the splash page and it was working. I think the error was due to here.

```

<activity android:name=".Post" />
<activity android:name=".Register" />
<activity android:name=".LogIn">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".MainActivity" />

<meta-data

```

- The system starts with the main page instead of the splash class.
- Then I have tried to do the message function. However, it was a huge implementation as well. It was hard to start, so i was start doing "Comment" function. The main point is to communicate for the application. It is still in process.
- I think I would leave the payment and message function out of the project and finishing the comment function and add items to Cart in the following days.
- I have also booked the appointment for the showcase, meanwhile i got an offer already from Transpoco.
- And also, during these months, i have kept working on the technique report as well.
- I have contacted with my supervisor twice after the midpoint, due to my condition, I wasn't able to make frequent contact with her, but she has provided a lot of useful information for me to improve. I have taken the advice from her during these months.
- I still have a lot of work to do in the following week. I need to finish my functions and make sure it was uploaded to GitHub for the submission. And prepare for the presentation and the report as well.
- Thank you for all supports from all my lectures.

## 6.2 Invitation and consent form

Invitation message:

“Hi xxx,

I am DERU MA, and I want to invite you to participant to test my project which is a mobile application. If you are agreed, please sign the consent form. Thank you!

Regards,

DERU”

# National College of Ireland

## Consent Form

– testing the mobile application

< Mobile Application – On The Go>

- I agree that I have consent to participant in the testing phase of the application.
- I agree that I can withdraw anytime during the testing.
- I agree that it is a very short and flexible testing process.
- I agree that there is no commission for participating.
- I agree that all data will only be used for the college project and accessed by DERU MA.

Signature: