National College of Ireland

Business Information Systems

Academic Year 2020/2021

Cristian Paltinean

X17138736

X17138736@student.ncirl.ie

Online Library

Technical Report

Contents

Executive Summary

For this final year project, I have developed an application for android devices that provides access for students to online academic books. The platform I have used is Android Studio, coding part was done using Java language and the design executed in the xml files.

Online libraries have proved to be one of the most useful and popular apps for the general public, and especially students. Compared to traditional libraries, online libraries have many benefits and advantages starting with convenience. Having the possibility to download such apps on one's phone means 24/7 access, given that there is internet connection. Additionally, there is a bigger selection and storage, organized and easy navigation and approach to desired sources. All these details are notably relevant during current quarantine and pandemic rules, with physical libraries closed. It is particularly helpful for students, who might be under pressure with time consuming papers and deadlines.

The application will allow the users, mainly students, to access academic books, specifically with an interest in computer science. The app requires the user to register and provide details such as name, email and create a password. All the user details are safely stored in the database (Firebase). Another function of the app is the Login, that requires the provision of an email and a chosen password. If the user enters the wrong password 3 times the Login button will be inaccessible and change of password will be required. Also, to prevent this, I have created a function that warns the user of the remaining login attempts. This is displayed as a message on the main screen during login phase. Additionally, I have created a multilingual function that facilitates navigation for non-English speaking users. This function can be accessed from the main page, as well as from the drop down menu available on different pages. The drop down menu contains various options such as: About Us page, Contact Us, Review page and Logout. The app also incorporates add to cart and payment function.

1. Introduction

1.1. Background

The idea behind this application was inspired by the student`s need to access relevant text-books and academic writings in a time effective manner to reduce the pressure around papers and deadlines. Moreover, I was thinking in terms of the current pandemic social distancing rules and restricted access to physical libraries, which usually is an essential network of support for most students. I started thinking of a way for students to access academic writings via mobile phone /tablet. The difficult part here is that I have set high standards regarding the execution of the app and its functionalities as my specialization is not a developer.

The product I have created is an android based application where the users can register, login and search for relevant books.

1.2. Aims

The aim of the project was to develop an android application which can be used on mobiles but also on tablets, to be fully responsive. The objective of this application is to facilitate student's access to online academic books for increase convenience, especially in the context of the current pandemic and its implications.

The intended customers are the students, more specifically students with an interest in computer science. Students can find academic books organised by categories relevant to computing science, add the books in the cart and buy them for a reasonable and affordable price.

1.3. Technology

Below are the technologies used in order to complete this project:

Android Studio

Is the platform where I developed my app. The version I used is 4.1.3 as it was the latest version at that moment.

Java

The language I have used to program the app, where all the coding part was done, in other words it handles the application`s logic.

xml

The design part was done in the xml files meaning, the xml files control the UI of the application. When created a new page or activity, in Android Studio it automatically generates an xml file where the design part takes place.

Firebase

Firebase was used to store all the user's information as one of the requirements of the application for the users is to be registered. Cloud Firestore was used to store all the books added to cart. Ones the payment is completed the cart will be clear and the books will be removed from the cloud as well.

GitHub

Used for cloud based, added, and updated the project. Used mainly to back up my work as an extra security measure and also to provide access to my supervisor and examiners

Dropbox

Dropbox was used to store all the documentation of this project just as an extra security measurement.

LucidChart

Used to create the use case diagram and the class diagram

Balsamique

Used to create the wireframes, the mock-up for every activity

1.4. Structure

This paper is composed of 7 different sections and they are as follows:

- The first section of this report is the introduction, which includes the background, aims, technology used and the layout of the paper.
- Section 2 presents the literature review and the market research conducted which includes the advantages of accessing the online libraries.
- Section 3 outlines requirements such as: Functional requirements, Data requirements, User requirements, Environmental requirements, Usability requirements and use case diagram. It also shows the class diagram summarizing the structure of the system, describes how the application was implemented, types of testing used, how the app was evaluated and holds screenshots of the activities with explanatory description.
- Section 4 is the conclusion, where the main points of the app including advantages, strengths and limitations are summarized
- Section 5 contains an overview of the potential improvements and developments including extra functions that could be implemented for a more sophisticated app.
- Section 6 holds all the references for the relevant sources and material accessed
- The last section of the paper looks at the Project plan and the reflective journals

2. Market Research/Literature Review

A digital or online library is an online database that contains digital objects ranging from still images, texts, audios, videos, digital documents, and a library that can be accessed through the internet. Google Books for instance, is one of the best online libraries whereby individuals can access more than one hundred thousand books for consultation, online purchase, and downloading. Individuals can access the library by searching on the internet or download the app which is ever available for Android and IOS devices. There are many advantages that people will gain from accessing online library apps when compared to the traditional physical libraries.

Easy to Access

The Online library app is more accessible than conventional libraries because we can install it on our phones which we carry along all the time. Additionally, it is easier for the user to navigate as there is a search section that will help them find the books or content they want. The user can also go through different books divided according to specific categories they might be interested in. According to Perdana, (2020) : "*Users can access the collection by using their device everywhere and every time if the digital library is online based"*.

The library does not limit individuals as they can access it wherever they are as long as they are connected to the internet.

Bigger storage

The library contains massive storage that makes it easier to keep many collections which would be difficult for a physical one as it may require a vast room and many bookshelves. According to Kaiser, (2012), it is easier to store more information because it is in digital formats than physical books. The online library uses cloud storage which enables the keeping of vast amounts of information at a time.

Preservation of any collection

Kousha, (2011) noted that the digital library ensures that users can always access the information no matter how long it stays, unlike in conventional ones where it might get lost or destroyed by human or natural elements. Likewise, the digital collections are of better quality than printed ones, making it easier to preserve them. For instance, many people can purchase

to view specific information or books, but it will maintain the same quality for all individuals that want to view it.

Structured Approach

The digital library offers its users access to detailed content in a structured manner. For instance, a user can move from the collection to a specific book and then to a precise chapter, and so on. According to Sun, (2012), this strategy is not possible in a conventional library, but the app provides links that help an individual navigate through the book in an orderly manner.

Networking

The library app is great because it enhances learning through networking, whereby it provides links to any other sources that might be related to the one you are viewing. Additionally, it can connect a user to resources in other digital libraries. The app enables creating an integrated resource sharing platform that is essential for learners and researchers to get all the documents they want without the limitation of physical libraries. As Jones (2011) noted: "Similarly, Google Books may serendipitously discover works that are significant at the item level." Online books enable the reader to find similar resources from different libraries.

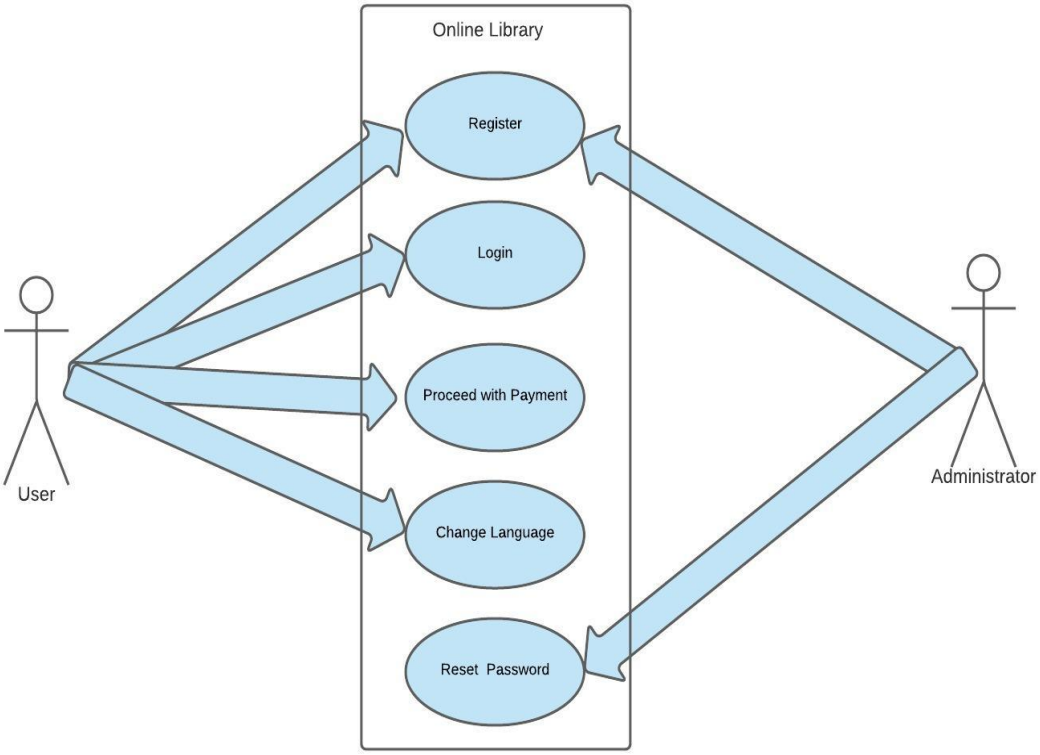
Conclusion

Online library apps are shown to be the best online libraries, as reported by "The Google Play Book" users, who can give a detailed account of the benefits they derive from the resource, especially compared to conventional libraries. The users can easily access it and navigate themselves through it since it is pretty easy. Additionally, it has bigger storage, making it easier to access all the required materials and maintain the documents for a long time without any distortions taking place. Lastly, the library uses a structured approach which makes it easier to navigate through a resource and enables networking through connecting the user to related resources.

3. System Requirements

3.1 Functional Requirements

3.1.1 Use Case Diagram



3.1.2. Requirement 1 < User registration >

3.1.2.1. Description & Priority

In order for the users to access the application I am developing, firstly they need to be registered into the system.

3.1.2.2. Use Case Description Register

Purpose

To allow a user to register.

Scope

The scope is to allow the user to register to use the application.

Description

This user case explains the registering of a new user.

Flow Description

1. **Pre-condition**

   User is not logged in or registered.

   User is at the main screen.

2. **Activation**

   This use case starts when the user decides and choose to register.

3. **Main Flow**

   1. The user clicks on the sign-up button.
   2. The system brings the user to the sign-up page.
   3. System asks the user to fill out the name, email, and password fields.
   4. User completes all the requests and presses button to register. (A1 Already registered)
   5. The system issues a confirmation email to the user.
   6. User accesses the link received in the email and proceeds to email authentication.
   7. The system confirms the user has registered.

4. **Alternative Flow**

   (A1 already registered)

   1. The system tells the user the email they are trying to use has already been used.
   2. The user can choose a new email to register or go back to the login page.
   3. Continues from main flow point 5.

5. **Termination**

   The user is registered and authenticated.

6. **Post-Condition**

   The system goes to the login screen.

3.1.3. Requirement 2 < Login Function >

3.1.3.1 Description & Priority

It is essential for the users to be logged In to the system and it is the only way that they can access the features of the application.

3.1.3.2 Use Case Description Login

Purpose

To allow a user to Login.

Scope

The scope is to allow the user to login to the application.

Description

This use case describes the user steps when logging into the application.

Flow Description

1. **Pre-condition**

   User must not be logged in.

   User is at the home screen.

   User is presented with blank email and password fields.

2. **Activation**

   This use case starts when the user decides to login.

3. **Main Flow**

   1. The system asks the user to enter the email and password.
   2. The user fills out the email and password fields.
   3. The user presses the login button. (A1 username or password are incorrect)
   4. The system authenticates the user and proceeds to next screen.

4. **Alternative Flow**

   (A1 username or password are incorrect)

   1. The system notifies the user that their username and or password are incorrect.
   2. The user tries one or both again.
   3. System shows still incorrect.
   4. User presses on reset password button.

5. **Termination**

   The user is logged in.

6. **Post-Condition**

   The system goes to the next screen.

3.1.4. Requirement 3 <Proceed with Payment>

3.1.4.1. Description & Priority

In order for the users to complete the transaction they need to make a payment

3.1.4.2. Use Case Description Payment

Purpose

To allow a user to make a payment

Scope

The scope is to allow the user to make a payment a finalize the transaction.

Description

This use case describes how the user proceeds with the payment

Flow Description

1.Pre-condition

User must be logged in.

User is at the checkout screen.

User is presented with blank name, type of card and card number fields.

2.Activation

This use case starts when the user decides to check out and make a payment.

3.Main Flow

1.The system asks the user to enter the full name, card type, card number and CVV number.

2.The user fills out the required fields.

3.The user presses the Pay Now button. (A1 One of the required fields entered by the user are incorrect)

4.The system confirms the payment.

4.Alternative Flow

(A1 Full Name, Card Type, Card Number or CVV are incorrect)

1.The system notifies the user that the entered fields are incorrect.

2.The user enters the name and card details again.

3.System shows still incorrect.

5.Termination

The user completed the payment.

6.Post-Condition

The system goes to the confirmation screen.

3.1.5. Requirement 4 <Change Language>

3.1.5.1. Description & Priority

This requirement is very helpful for non-native English speakers, as they can navigate through the app using their own language.

3.1.5.2. Use Case Description Change Language

Purpose

To allow the user to change language.

Scope

The scope is to allow the user to select the preferred language.

Description

This use case describes the user changing the language.

Flow Description

1. **Pre-condition**

User must not be logged in.

User is at the main screen.

2. **Activation**

This use case starts when the user decides to change the language.

3. **Main Flow**

1. The system presents the user with change language button to press.
2. The user will press the change language button.
3. The system will ask the user to select a language.
4. The user selects the language.
5. The system automatically changes and displays the app in selected language

4.Alternative Flow.

5. **Termination**

The user has changed the language.

6. **Post-Condition**

The system goes to the initial state.

3.1.6. Requirement 5 < Reset Password >

3.1.6.1. Description & Priority

From my point of view, the ability to reset the password is an essential feature that every app should have. Giving the users the option of changing the password at any time with confirmation email as well as storing all their details in a database will give the users a

feeling of reassurance and confidence. This will also increase the security and may keep the hackers away from their accounts.

3.1.6.2. Use Case Description Reset Password

Purpose

To allow a user to reset the password.

Scope

The scope is to allow the user to reset the password for the account already created.

Description

This use case describes the user resetting the password.

Flow Description

2. **Pre-condition**

User must not be logged in.

User is at the main screen.

3. **Activation**

This use case starts when the user decides to reset the password.

4. **Main Flow**

1.The system presents the user with a forgot password button to press.

2.The user will press the forgot password button.

3.The system will ask the user to enter email used for account.

4.The user enters their email address and presses enter. (A1 Email not recognised)

5.The system sends the user a confirmation email.

6.The user clicks email link.

7.The system asks the user to enter new password.

8.The user enters new password and clicks enter.

5. **Alternative Flow.**

(A1 Email not recognised)

1. The system tells the user the email is not recognised or not registered.
2. Continues from main flow point 4.

6. **Termination**

The user has reset their password.

7. **Post-Condition**

The system goes to login screen.

3.1.7. Data Requirements

Database Firebase is used to store and retrieve all the user information and Cloud Firestore used to store all the books added to cart by the users. While the database is used as a service as a service (SaaS), the maintenance is performed by the third party body.

3.1.8. User Requirements

In this part of the report, I will describe what are the user requirements; in order for the users to download and run this app. they must assemble the following requirements:

- **Android Device** – The users must have an android phone or any android device as the application was developed only and specially for this type of devices.
- **Email Address** - The users should have a valid email address as the application requires a registration which can be validated and completed by confirming and accessing the link provided and sent by the system.
- **Internet Access** – For the users to use the app it is essential to have internet access as all the user details are stored in the database Firebase and all the books added to cart will be also stored in the cloud Firestore.

3.1.9. Environmental Requirements

The environmental requirements that were needed when created the Online Library app are as follows:

- Windows 10 operating system machine was needed to install the below software
- Android Studio – is the platform the app was built using Java language and xml.files. Utilizing this software, I was able to set up the development environment and to implement the app as has incorporated the following components:
- Android Virtual Device
- Android SDK
- Java Development Kit (JDK)

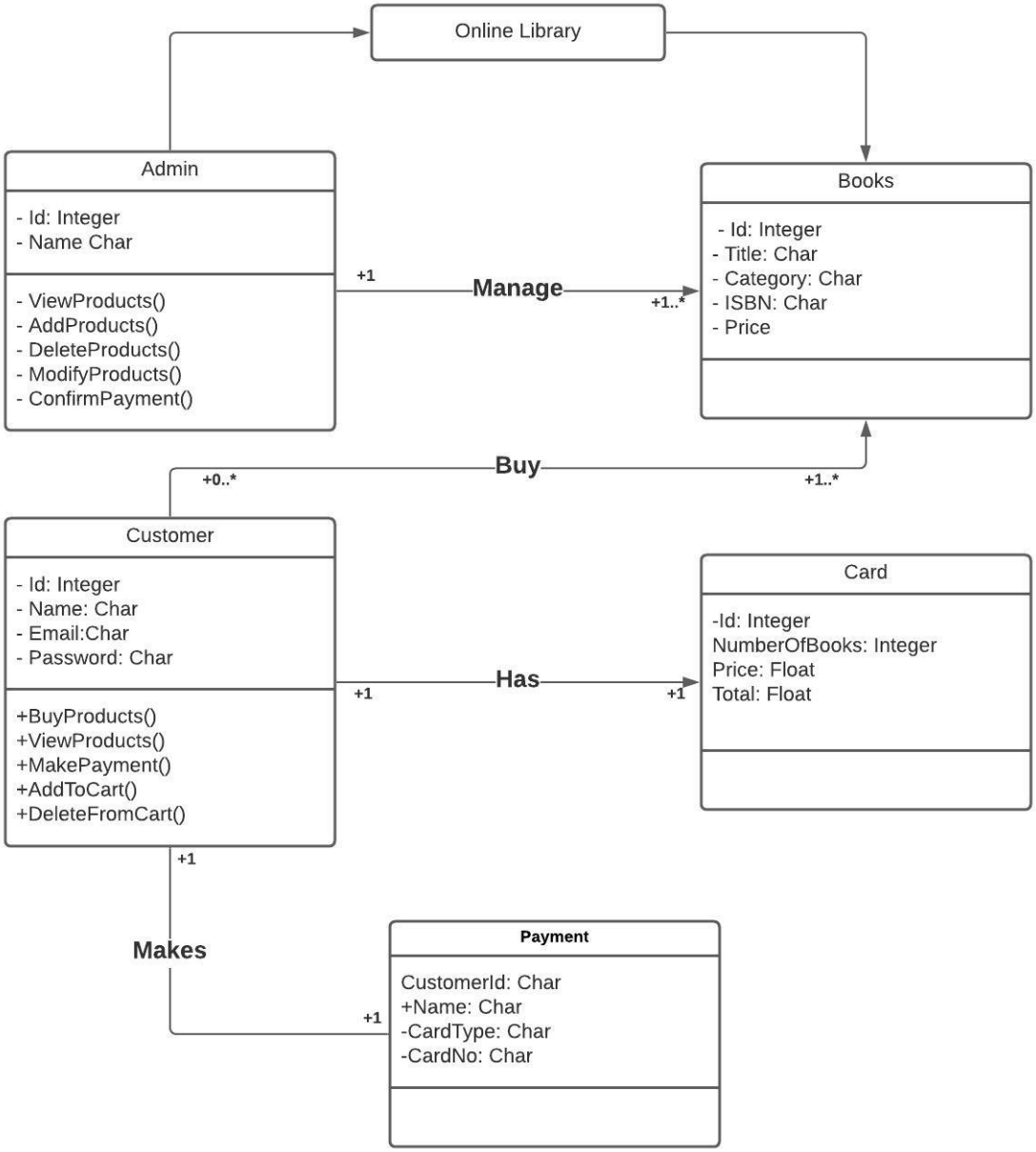
3.1.10. Usability Requirements

In order for the users to access Online Library app they need to sign up and create an account. Assuming that for completion of the mentioned requirement the users are able to fill all the fields as name, email and extremely important to choose a strong password.

3.2. Design & Architecture

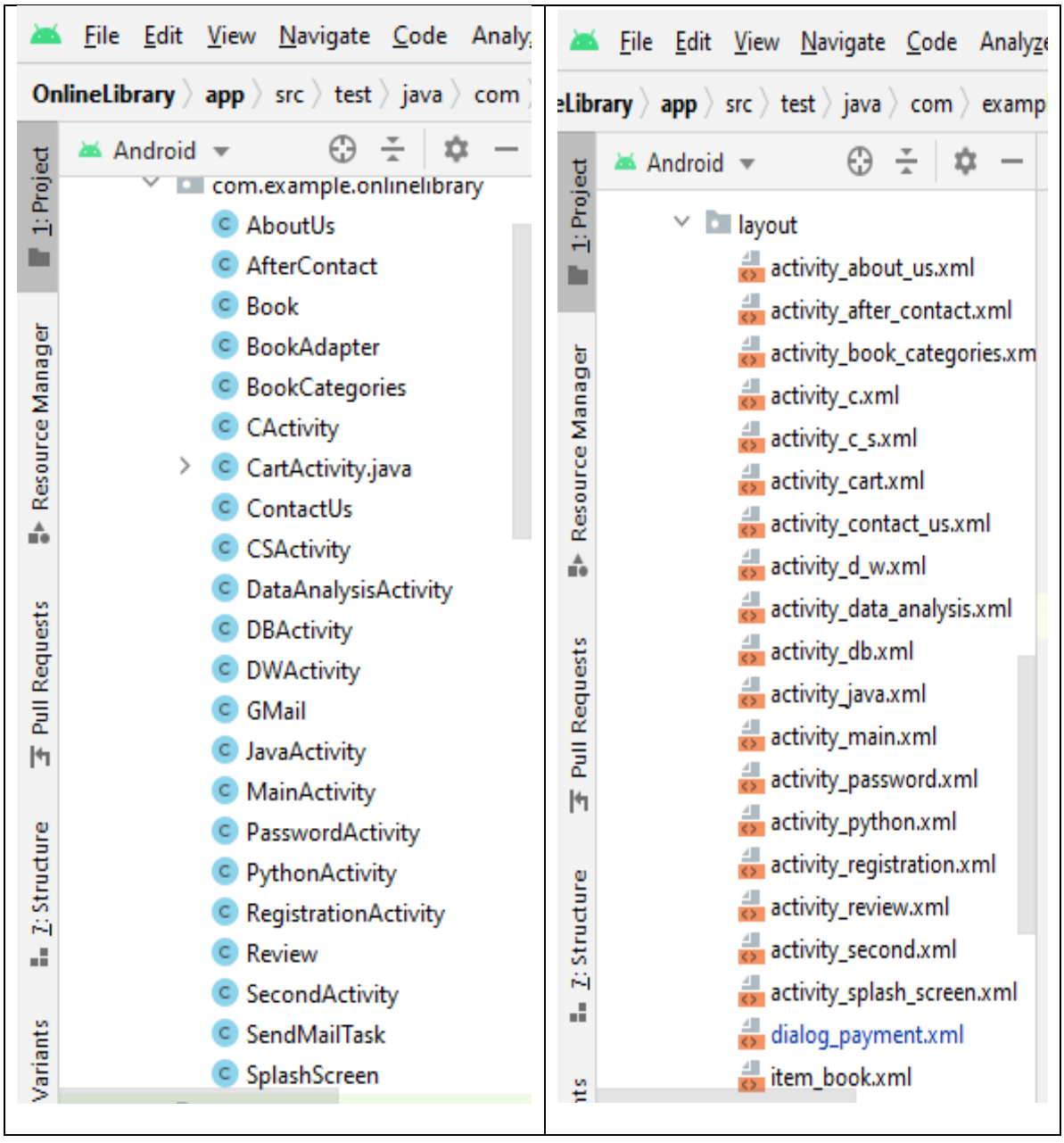
System Architecture

The below class diagram outlines the structure of the system for the android app I am working on for this project. The reason I have chosen this architecture for this application is because I wanted to be easy to operate by all the users.



3.3. Implementation

In this part of the paper, I will describe how I have implemented the application I developed. The below table illustrates the Java files and their appropriate xml files created for every activity of the app.



For setting book data to the respective book category, a recycler view is used and a BookAdapter class is used to populate the recycler view. The recycler view has three main methods which are responsible for handling the data and displaying in the activity view. However, for them to work, a view holder and book list have to be passed. The view holder describes the layout items in which the data will be displayed. These methods are shown in the image below. The method getItemCount indicates how many items will be displayed in the recycler view. The onCreateViewHolder method is responsible for attaching the layout resource of each book item. The onBindViewHolder method can be used to describe each book item by using the position parameter. Here, we set the values of the layout widgets and even handle the add to cart button functionality. The image is added in firestore as a cart item.

```
@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    context = parent.getContext();
    View view = LayoutInflater.from(context).inflate(R.layout.item_book, parent, attachToRoot: false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    holder.getImageView().setImageResource(books.get(position).getImage());
    holder.getTextView().setText("{books.get(position).getPrice()} €");
    holder.getButton().setOnClickListener(listener -> {
        db.collection(Objects.requireNonNull(auth.getUid())).add(books.get(position));
        Toast.makeText(context, text: "Added to Cart", Toast.LENGTH_SHORT).show();
    });
}

@Override
public int getItemCount() { return books.size(); }
```

The next important class is the Gmail class which is responsible for sending emails to a user's account. In the createEmailMessage method, we try to create a Mime Message object from the data passed in the Gmail constructor. A mail session is first created and passed to MimeMessage object which is used to set the from property (sender of email), recipients, subject and the message itself. This method returns a MimeMessage which is later used in the sendEmail method. The sendEmail method describes a transport protocol, connects to it, sends the message, and when done closes the transport object, as it is bad practice to leave it open.

```
public MimeMessage createEmailMessage() throws AddressException,
        MessagingException, UnsupportedEncodingException {

    mailSession = Session.getDefaultInstance(emailProperties,  authenticator: null);
    emailMessage = new MimeMessage(mailSession);

    emailMessage.setFrom(new InternetAddress(fEmail, fEmail));//setting up address
    for(String tEmail : tEmailList){
        emailMessage.addRecipient(Message.RecipientType.TO,
                new InternetAddress(tEmail));
    }

    emailMessage.setSubject(mailSub);
    emailMessage.setContent(mailBody,  type: "text/html");
    return emailMessage;

}

public void sendEmail() throws AddressException, MessagingException
{
    Transport transport = mailSession.getTransport( protocol: "smtp");
    transport.connect(emailHost, fEmail, fPassword);
```

In CartActivity, when the Make Payment button is clicked, a dialog is displayed and prompts the user to enter their card details before completing payment. After they are done entering the details and click the Complete button in the dialog, a payment is simulated returns as successful after which the cart is cleared of all items. This is done in firestore so that the cart data does not persist across different login sessions. Since firestore does not allow deletion of multiple objects at once, the below image shows clearing the cart using a batch method provided by the firestore SDK.

```
builder.setPositiveButton( text: "Complete", (dialog, which) -> {
    WriteBatch writeBatch = db.batch();

    for (int i = 0; i < ids.size(); i++){
        DocumentReference documentReference = db.collection(user).document(ids.get(i));
        writeBatch.delete(documentReference);
    }

    writeBatch.commit().addOnSuccessListener(aVoid -> {
        Toast.makeText(context, text: "Payment Success", Toast.LENGTH_SHORT).show();
        dialog.cancel();
        listener.onPaymentSuccess();
    });
});
```

3.4. Graphical User Interface (GUI)

Below I have attached the screenshots of key activities of the application I have developed. The application will be opening with a splash screen displaying an image and a text for a few second (see figure 1).

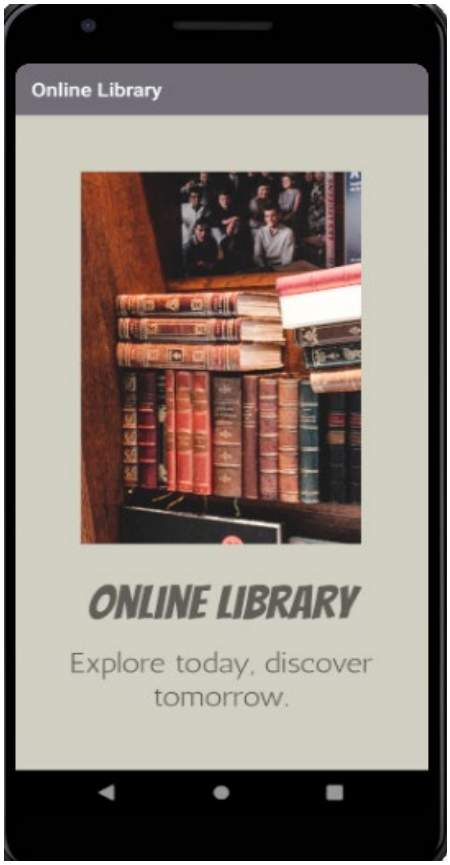


Figure 1

Main Activity

Next page that will be opening, also named the main activity, allows the users to choose the preferred language and to login. Also, from the same page the users can proceed with the registration. In case the users enter the wrong password 3 times, the Login button will be disabled, and the users can reset the password by accessing the forgot password button.

Login

Once the users are registered they can login at their convenience by entering their email and password and clicking on the login button. Then the users will be redirected to the second activity.

Figure 2

Change Language Function

The below displayed screenshot shows the selection of languages available for the users. This function was specially created for the non-native English speakers, enabling them to select the preferred language in order to navigate through the app.

Figure 3

Forgot Password

By clicking on the forgot password button from the main activity, the users will be redirected to the password activity where the users will be asked to enter the email and then to click on the reset password button (see figure 4). The system will automatically send an email asking to follow the link in order for the password to be reset (see figure 5). Accessing the link, a pop up message will appear asking to enter the new password (see figure 6). Completing this step another message will be displayed mentioning that the password was changed and that you can sign in with your new password (see figure 7).

Figure 4

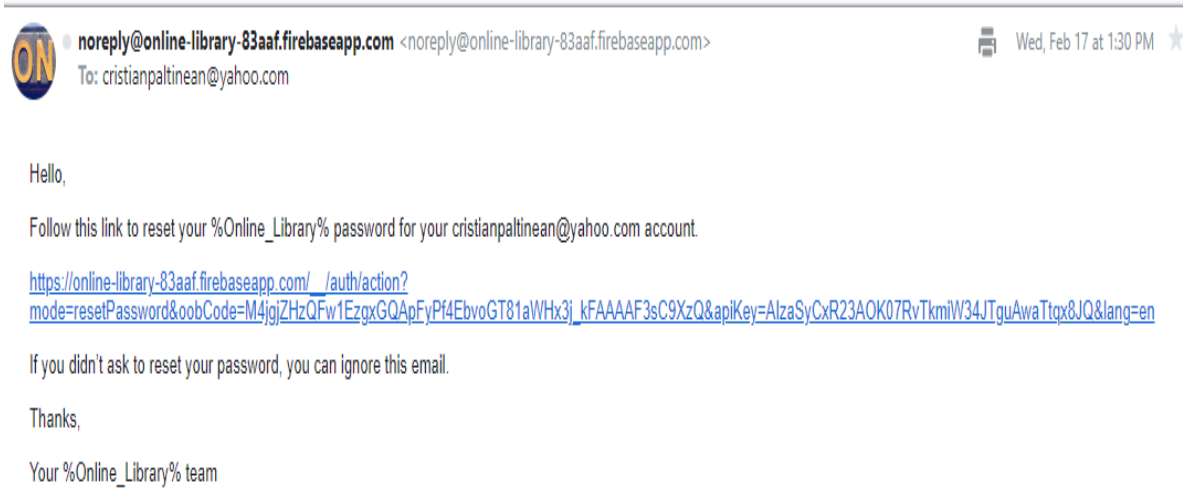


Figure 5

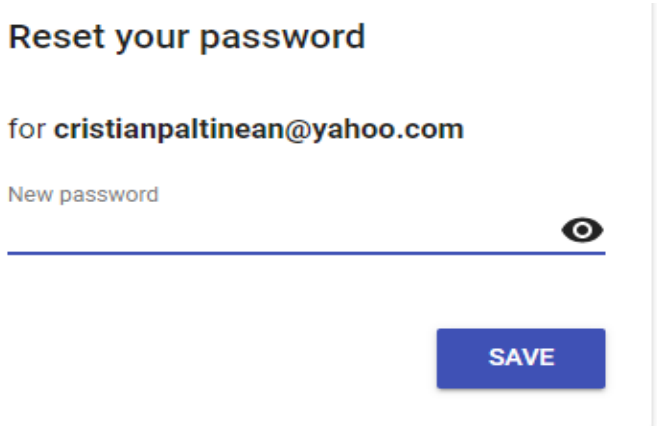


Figure 6

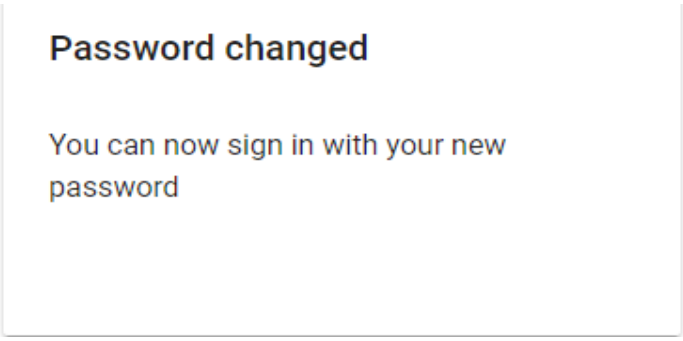


Figure 7

User Registration

In order for the users to have access on the app they primarily need to be registered. When accessing the Sign-Up button from the Main Activity, the user will be redirected to the Registration page where it will be required to enter name, email, and password and then to click on the register button (see figure 8). After that, the system will automatically send a confirmation email to the user. Once the link received is accessed, the registration will be completed, and the user can login to the app at any time (see below 9). If the users accessed the Sign-up button by mistake they can go back to the main activity clicking on the already signed in button.

Figure 8

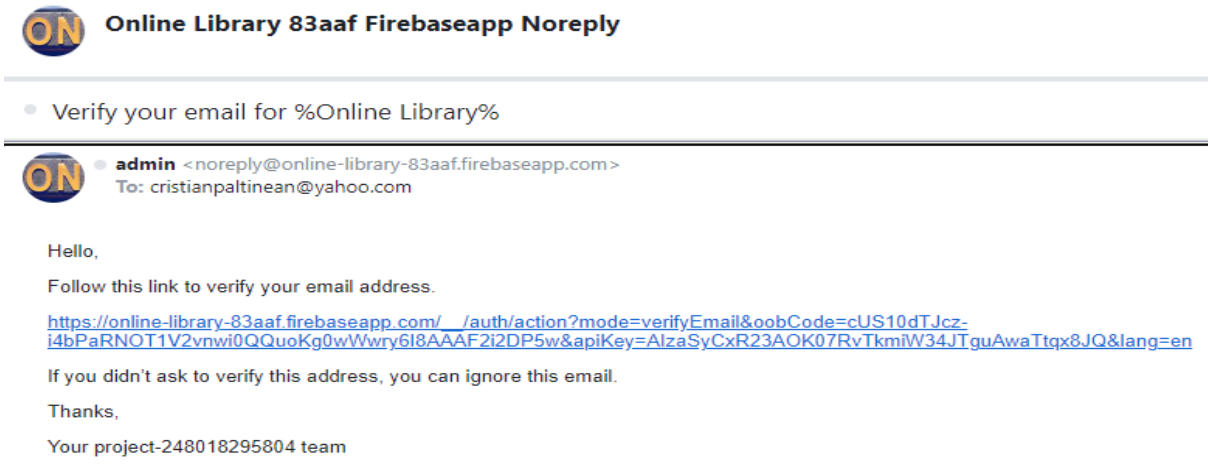


Figure 9

Second Activity

The below screenshots show how this page actually looks. Once the user is logged in, this page will be displayed. Here the users can access the drop down menu and also can choose to search for different classifications of books by clicking on book categories button .

Figure 10

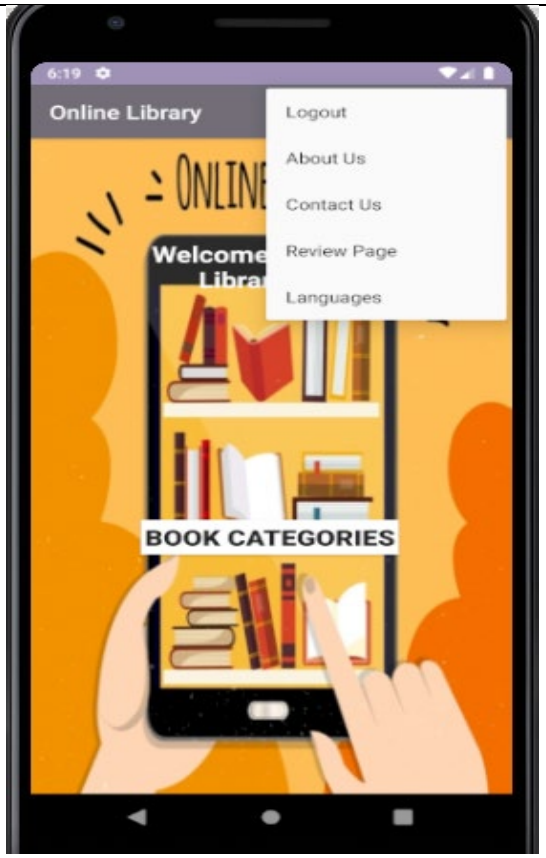


Figure 11

Drop-down Menu

On the second activity or on the first page that opens after the users logged in, by opening the drop down menu, the users can access the Contact Us page, Review page, About Us, changing the language and also they can log out from the app (see figure 11).

About Us

Accessing this page, the users can see information about the app but also information about myself (see figure 12). On this page there is no direct interaction with the users as it contains only text fields.

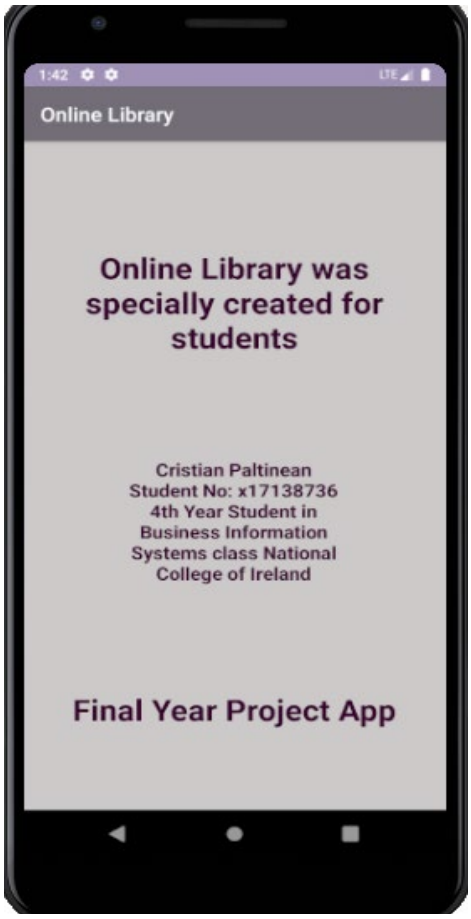


Figure 12

Contact Us Page

On this page the users can contact me by entering their names, emails and typing their messages or query. In order for the message to be sent they need to complete all the required fields in the form (see figure 13). If failed to do so, a pop up message will be displayed saying that all the fields need to be filled. When the message or query is successfully sent another page will be prompted thanking the user for the message (see figure 14). The user's message will be received by administrator via email.

Figure 13

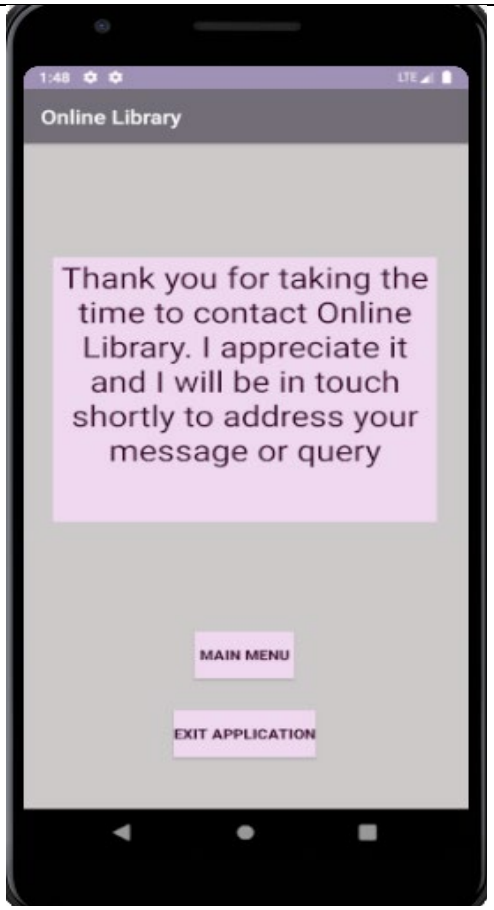


Figure 14

Review Page

On the review page the users have the opportunity to rate the application by selecting the number of stars, according to their satisfaction. This will prompt an automated text relevant for their feedback. The users can also type in their thoughts about the app on the same form and submit it (see figure 15).

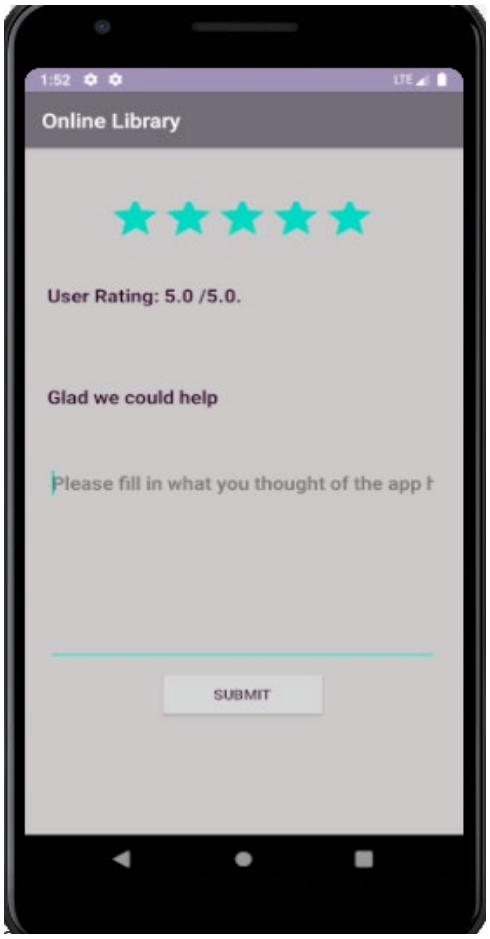


Figure 15

Book Categories

Below, the illustrations display the book categories page, where the users have the option to choose different type of books depending on their interest (see figure 16). Selecting and clicking on any of the buttons from this page will open and display books specific to that selection. Scrolling down this page the users can also access the cart and view the selected books (see figure 17).

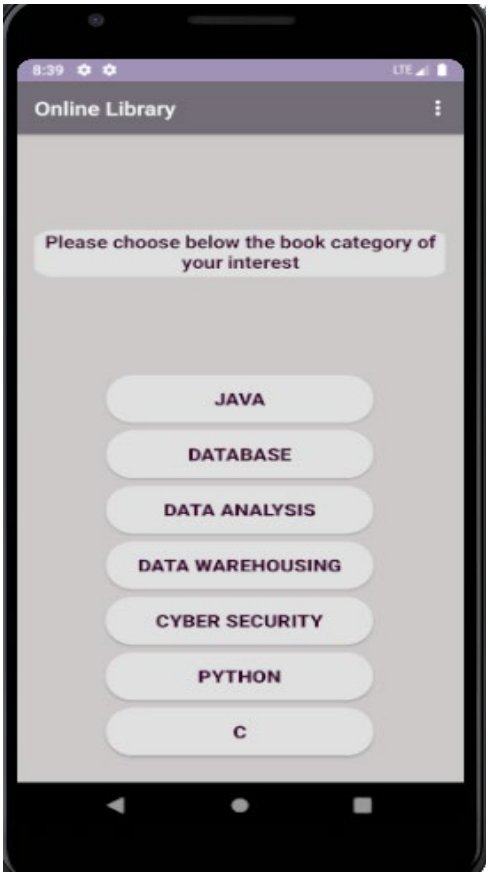
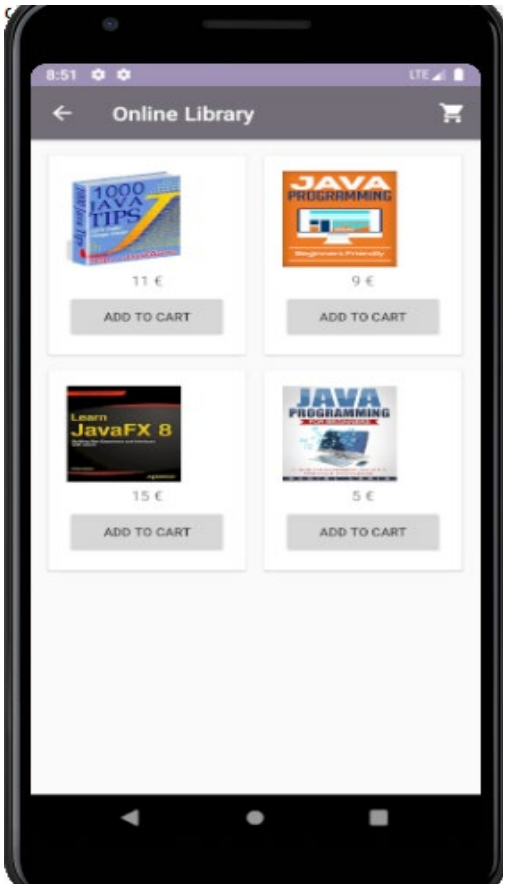
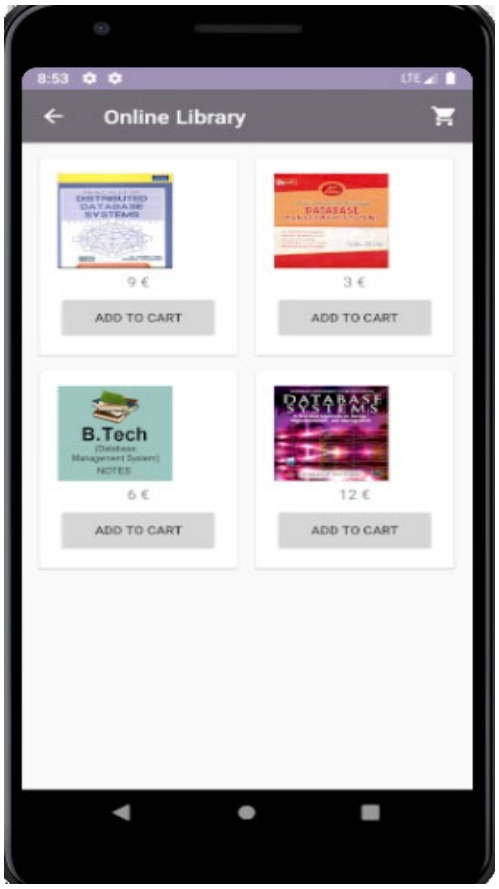


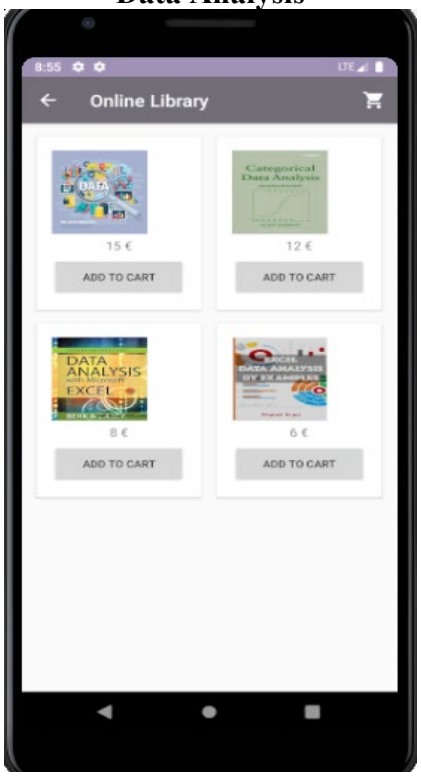
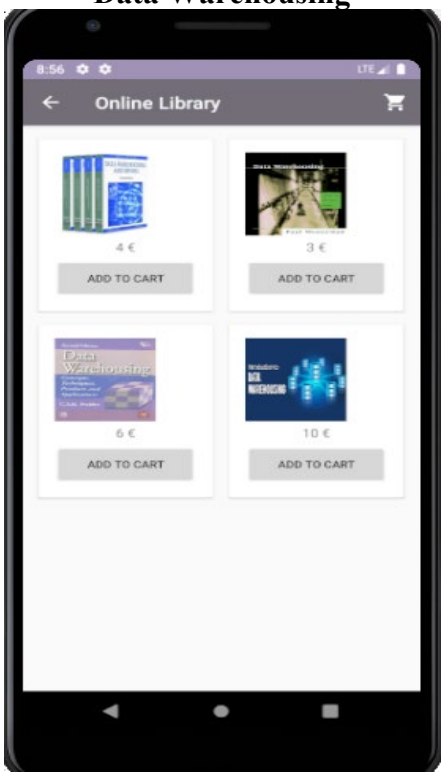
Figure 16

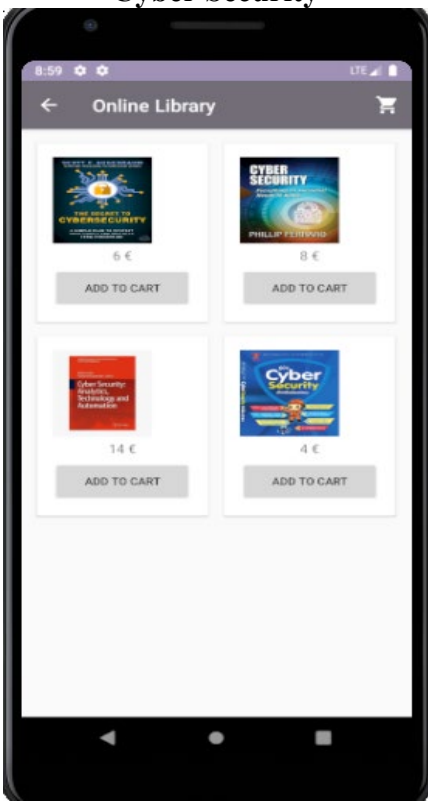
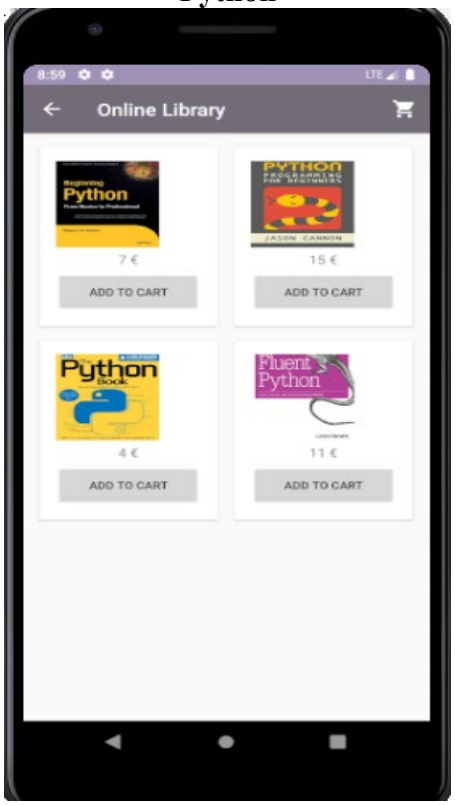
Figure 17

A book model was created with the fields id, and image. A book adapter was also created. This adapter is used with recycler views to display a list of books and an add to cart button. In the corresponding book activities, a book adapter is created and an ArrayList of model Book is passed as the constructor. When the books are displayed, we can click on Add to Cart button which adds the book details in the cart in Firebase Firestore service. Each activity has a cart button on the ActionBar menu that can be clicked and directed to the CartActivity. From there, we can see which books have been added in the cart. We can remove

books from the cart by clicking on the remove button. Doing so removes the book from Firestore. If everything checks out, payment can now be made by clicking on the Make Payment button at the bottom of the screen in CartActivity. This brings up a dialog to confirm the payment or cancel. Cancelling lets you go back and edit the cart, either removing an item or adding a new one. Confirming Payment is a mockery of payment process. It always returns true, meaning a payment is complete and therefore the cart in Firestore is emptied. This is done by deleting the entries in Firestore using a WriteBatch request.

Java	Database
	

Data Analysis	Data Warehousing
	

Cyber Security	Python
	

C++ Activity

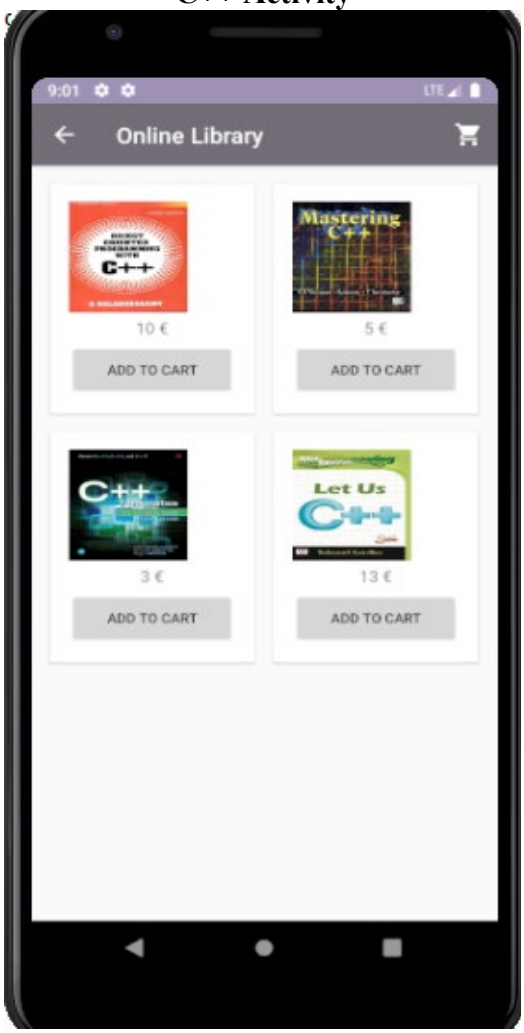


Figure 18

Cart Activity

Accessing the Cart, the users can see the added books, the price of each book and also they can easily remove the books by clicking the remove button located under each book (see figure 19). When a user added a book to the cart, a toast message will be shown on the screen confirming that the book was added to the cart. The Cart can be also opened from every Book Category by clicking the Cart icon located on the action bar in the top right hand corner (see figure 18). For a better navigation on this page, I have implemented the scroll function, as the users might have lots of books in the cart. Once the users decided what books will keep, they may proceed with the payment by clicking on the Make Payment button.

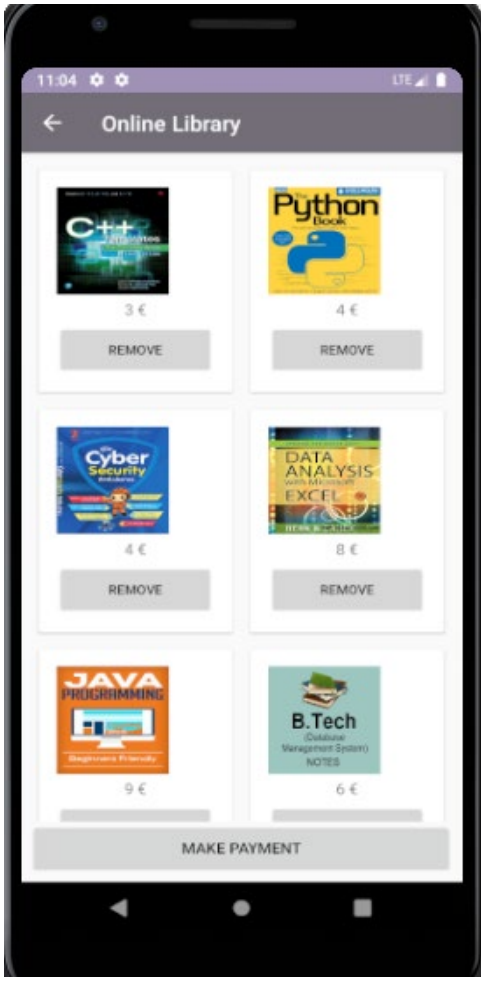

11:04
Online Library
C++
3 €
REMOVE
Python
4 €
REMOVE
Cyber Security
4 €
REMOVE
DATA ANALYSIS
EXCEL
8 €
REMOVE
JAVA
PROGRAMMING
9 €
B.Tech
6 €
MAKE PAYMENT

Figure 19

Proceed with the payment

Clicking on Make payment button, the user will be prompted with a new page where the total amount will be displayed. The user can continue with the payment or can cancel the payment and go back removing or adding a book. In case that the user wants to complete the payment it will be asked to enter the card name, card number, expire day, month, year and also the CCV number (see figure 20). Once all the mentioned details are filled in, the user will click on Complete Payment button. If the payment is completed successfully a toast message will be displayed.

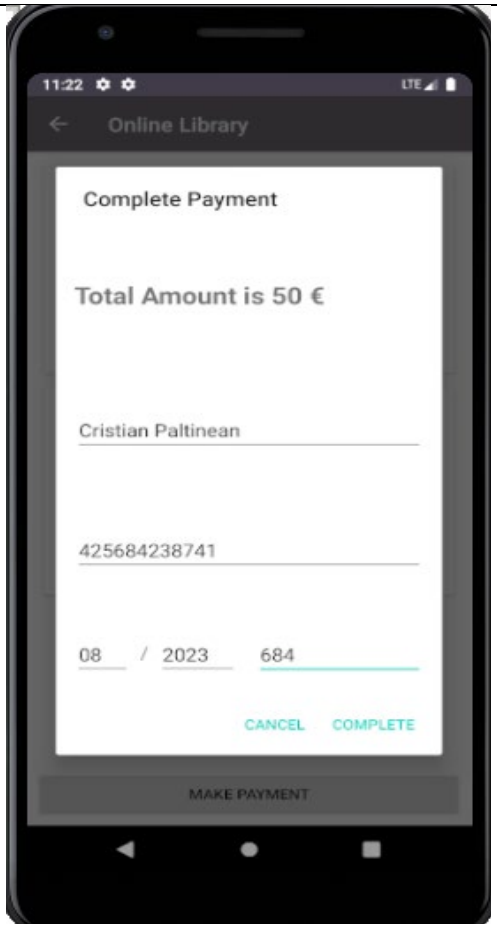


Figure 20

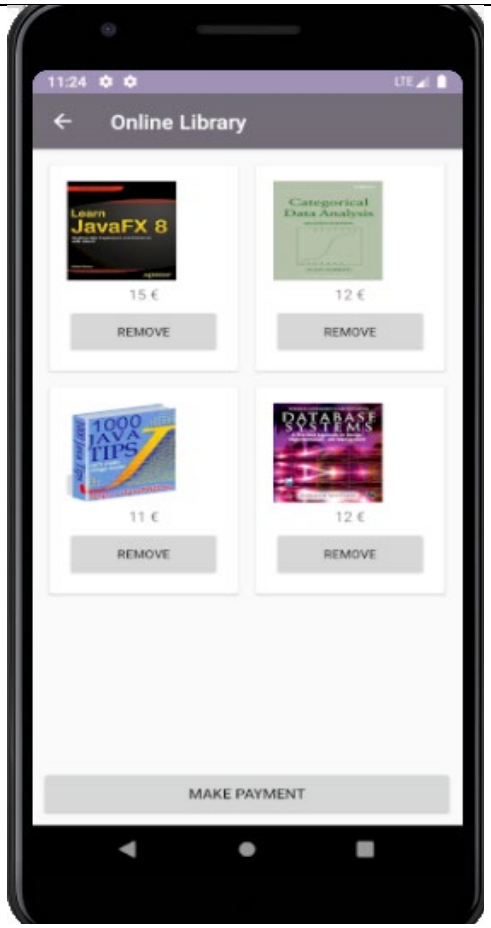


Figure 21

3.5. Testing

For testing, junit library was used. This is used for unit testing, the method of testing implemented in this application. There are two tests created (see Figure 22). The tests are named BookTest and GmailTest.

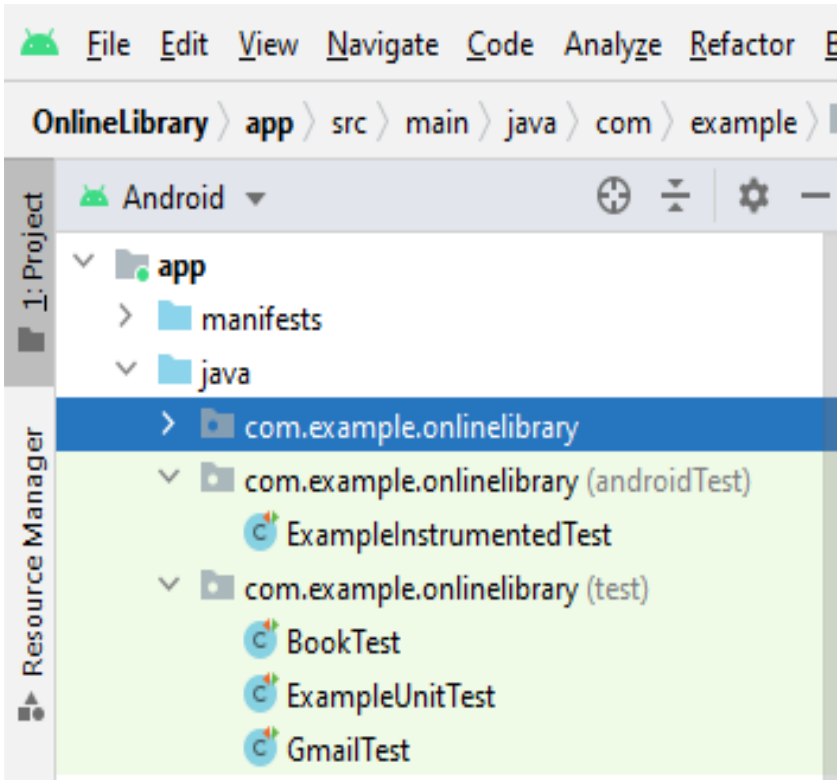


Figure 22

The BookTest is for testing the book model in order to find out whether all fields are correctly assigned. There are different ways to set fields in a model. One is through the model's constructor where all fields are set at once. The other one is through setter methods, which in turn set the value to particular fields in the model. The below image shows the testing implementation results after running it (see figure 23).

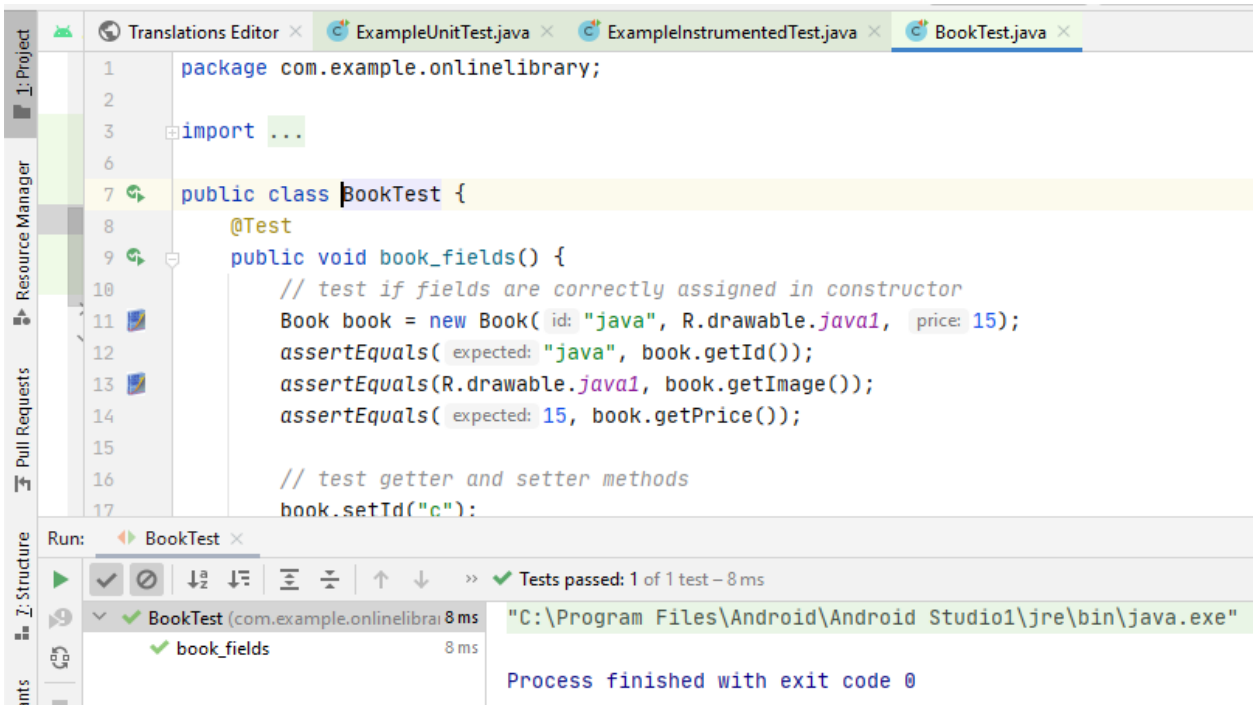


Figure 23

In the GmailTest test, I have checked to see if the email message object was passed with correct attributes and whether they match the default ones set. These tests are specific to the email subject, the email content/message, and the content type. The image below shows the test results of this specific test (see figure 24).

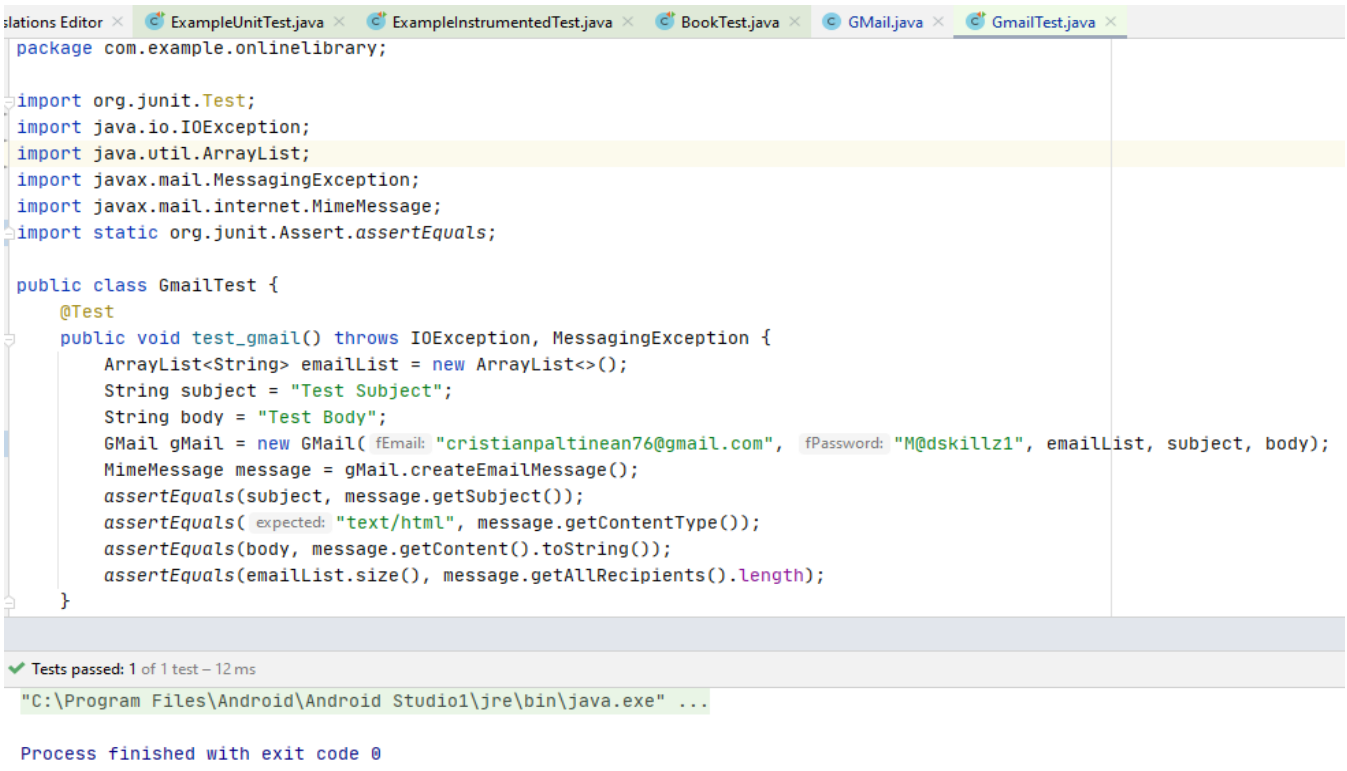
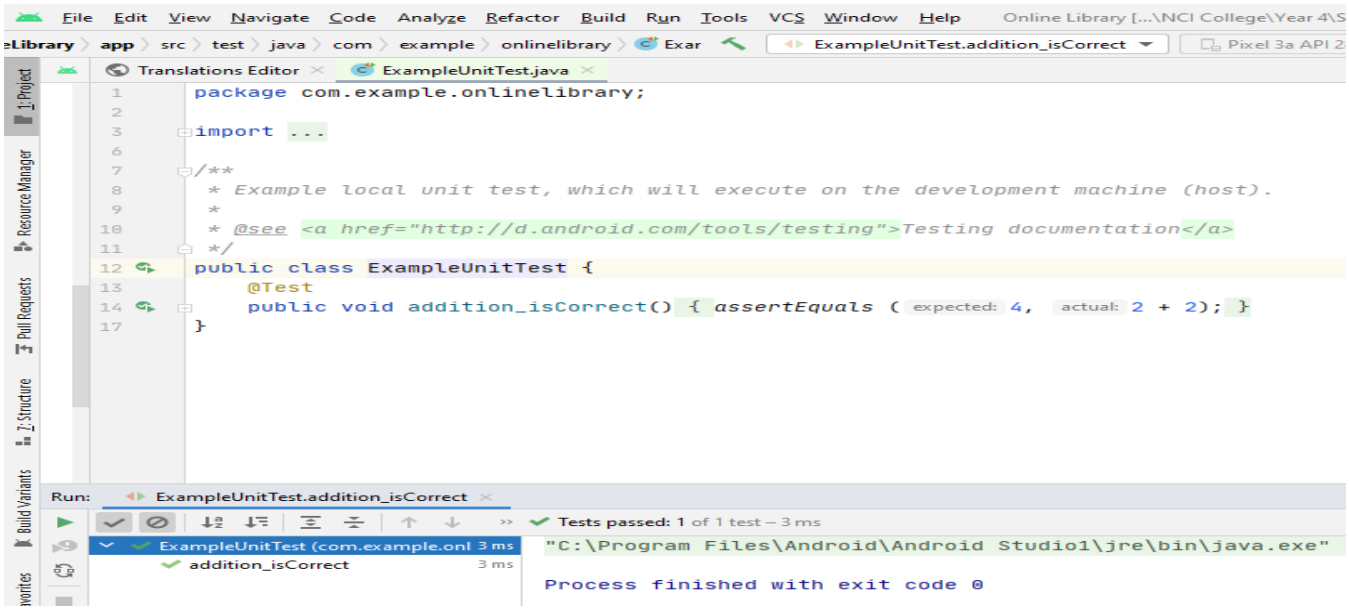
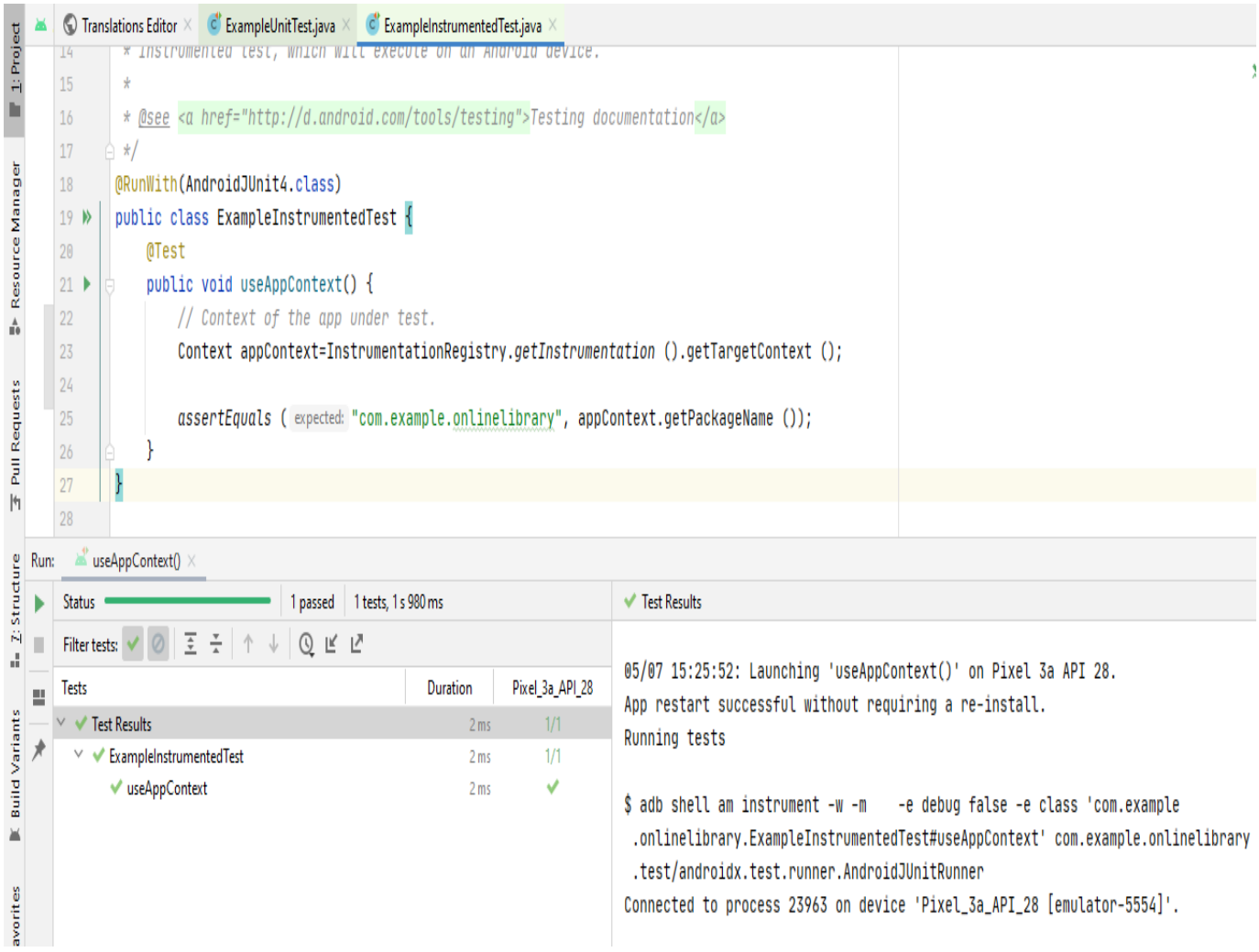


Figure 24

Below are example of unit test and instrumented test successfully passed





3.6. Evaluation

.

For evaluation of the system, since it is not yet available on Google Play Store, I opted to install the application on a couple of devices belonging to family and friends. The application appears to be able to be installed in all devices without any complications. The animations work smoothly in all the devices tested. None of the devices however, had an android API lower than 25. Checking on the usage data, the application did not consume so much data even when calling firebase SDK. This is probably because of how firebase is optimised to use less data and still work in low latency networks. Having offline storage is also a plus as the data is cached locally by firebase automatically.

For the cart items, no user (on the different devices) was able to see another user's cart. This is because of the permissions set. A user can only see what is in their cart and not on any other person's cart. Calls to firebase are also fast and one of the advantages of using the services is that it offers scalability. Therefore, the application has the ability to scale.

4. Conclusions

In conclusion what I have set up to do in the very beginning of the project I managed to complete.

If I have to do the project again I would probably start to work on it sooner to do more work in the first semester, as now I know that the plan I made before starting to work on the app might be delayed by different issues. During the whole academic year, I encountered problems with the coding part where I got different errors. In order to fix them I had to do a lot of research and watch tutorials, and that was very time consuming. Looking at the timeline on my project plan at different points, there was a delay in completion of tasks.

One of the limitations of the app is that the books available are for students and those interested in computing science only, thus it limits the target population. One of the strengths, a very important one is the safe storage of user details in the cloud, and its main benefits are no physical or storage boundary and unlimited, 24/7 access.

5. Further Development or Research

Have I had more time; I could have implemented some other functions on the app. One function would be live chat where the user could get instant, real time interaction and solution provision from the admin. Another function that I would add will be the audio books which are very helpful for blind individuals and not only. Audio books are also useful for the busy people as they can be listening while on the bus or even while driving. The application currently is available only for android devices but with additional time it can be modified/adapted for IOS devices. As visual of the app is vitally important I would also try to improve its design. During the survey I carried out, to collect data regarding user's preferences, a large number of android users emphasized the significance of the icon design when it comes to downloading the app. Having a good-looking app design, makes it more likely to increase the number of users who will download the app.

6. References

Jones, E., 2011. Google Books as a general research collection. *Library Resources & Technical Services*, *54*(2), pp.77-89.

Kaiser, M., 2012. Putting 600,000 books online: the large-scale digitization partnership between the Austrian National Library and Google. Liber Quarterly, 21(2).

Kousha, K., Thelwall, M. and Rezaie, S., 2011. Assessing the citation impact of books: The role of Google Books, Google Scholar, and Scopus. *Journal of the American Society for information science and technology*, *62*(11), pp.2147-2164.

Online Library Images [online] Available at:<https://unsplash.com/s/photos/library https://www.info.com/serp?q=register+here+images&page=28&sc=lc0oSPKjTSFl20 >[Accessed 21 December 2020].

Perdana, I.A. and Prasojo, L.D., 2020, February. Digital Library Practice in University: Advantages, Challenges, and Its Position. In International Conference on Educational Research and Innovation (ICERI 2019) (pp. 44-48). Atlantis Press.

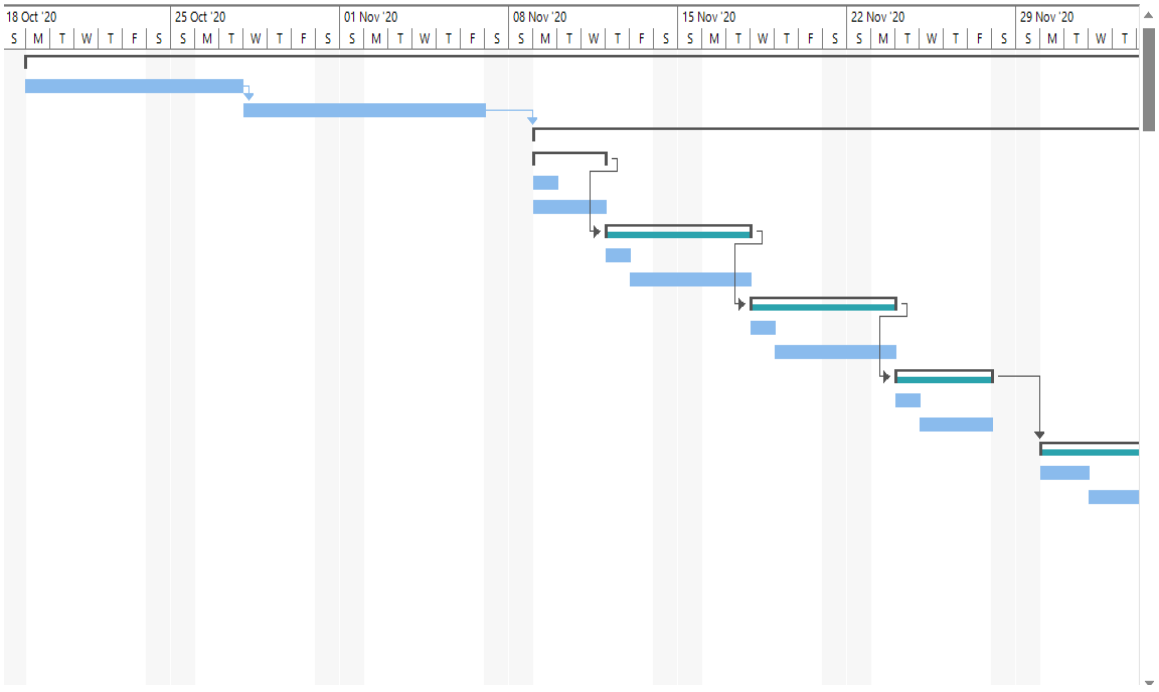
Sun, J., and Yuan, B.Z., 2012. Development and Characteristic of Digital Library as a Library Branch. IERI Procedia, 2: 12–17.

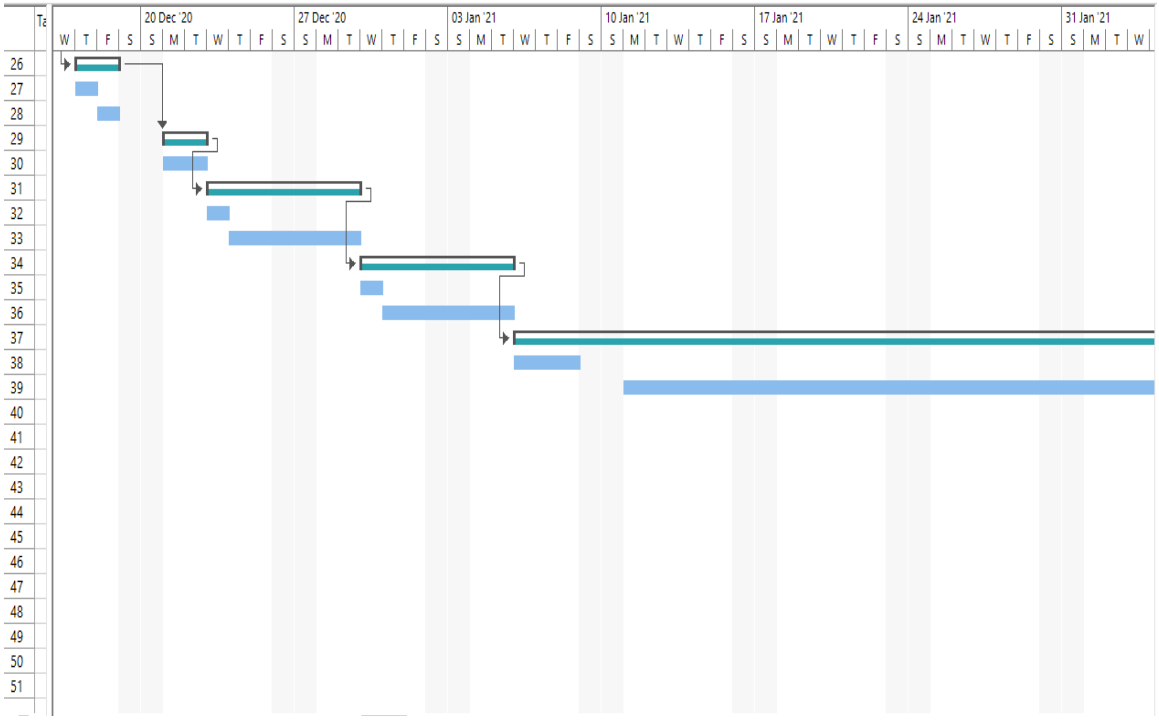
7. Appendices.

7.1. Project Plan

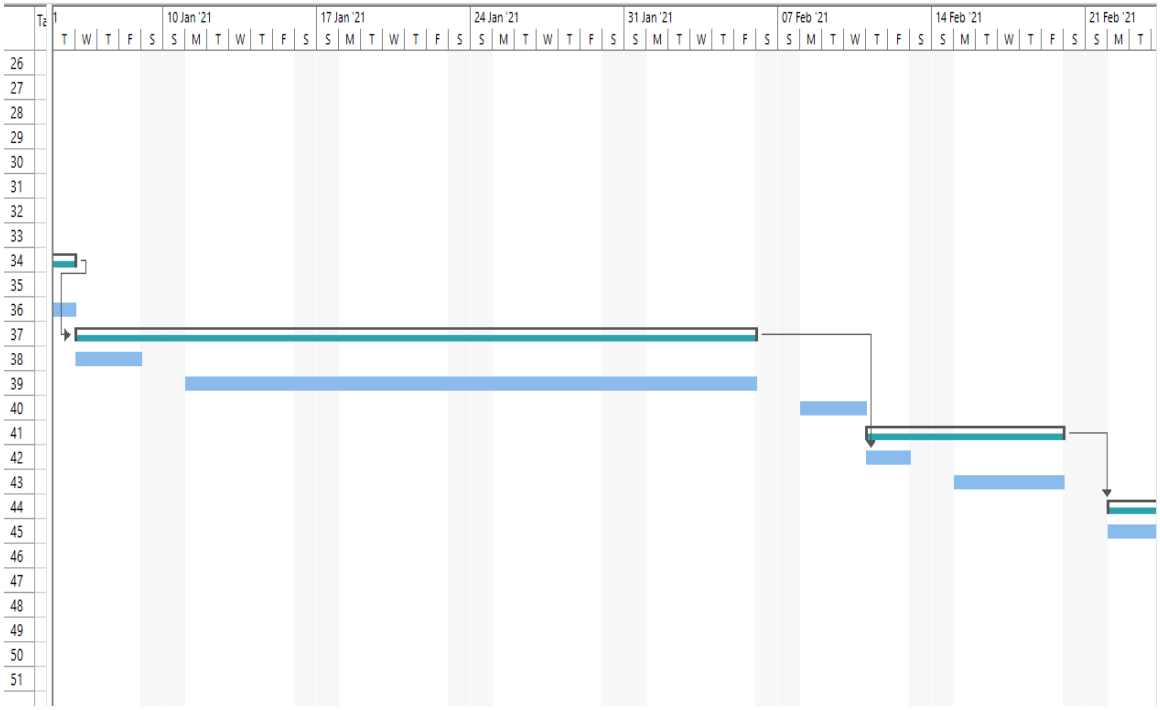
The below Gantt chart illustrates how my project was implemented and the timeline

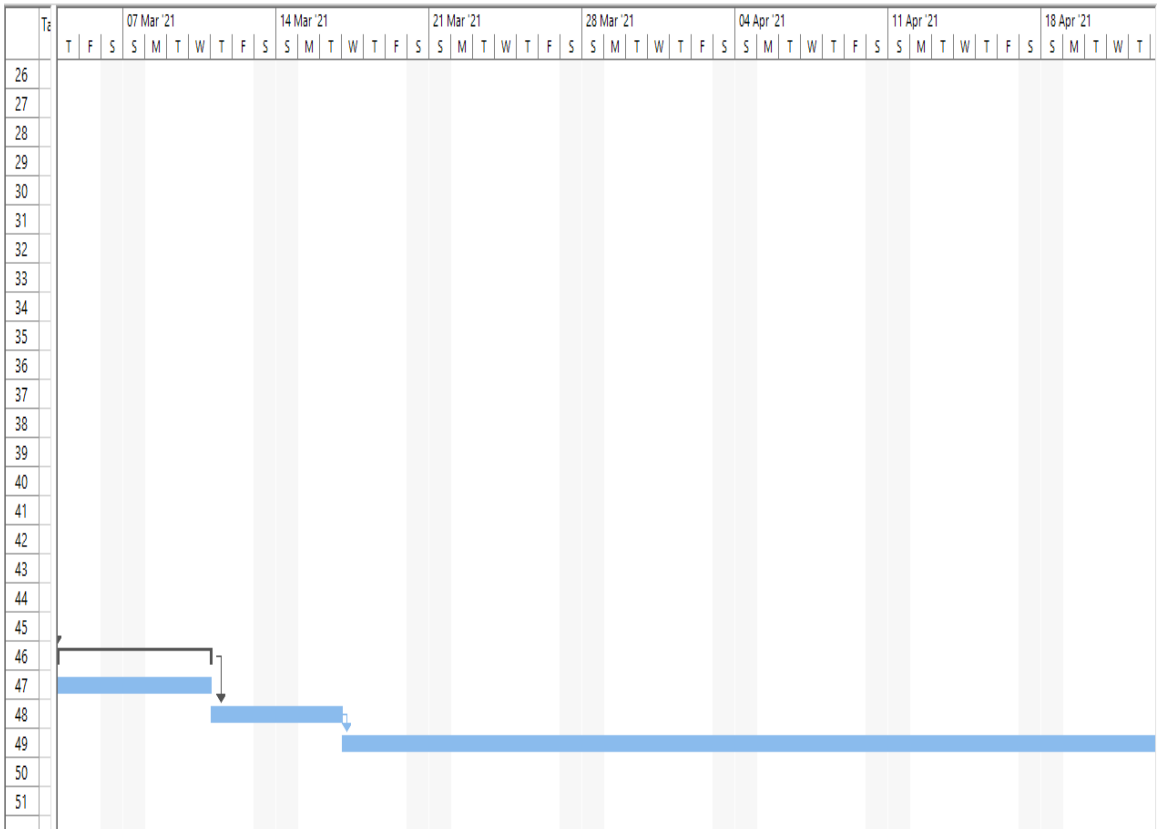
	Task Name	Duration	Start	Finish	Predecessors
1	**1. Online Library**	**145 days**	**Mon 19/10/20 8:00 AM**	**Fri 07/05/21 5:00 PM**	
2	2. Project Pitch	7 days	Mon 19/10/20 8:00 AM	Tue 27/10/20 5:00 PM	
3	3. Proposal	8 days	Wed 28/10/20 8:00 AM	Fri 06/11/20 5:00 PM	2
4	**4. Main Activity**	**20 days**	**Mon 09/11/20 8:00 AM**	**Fri 04/12/20 5:00 PM**	3
5	**4.1 Login**	**3 days**	**Mon 09/11/20 8:00 AM**	**Wed 11/11/20 5:00 PM**	
6	4.1.1 Create and Design Login Button	1 day	Mon 09/11/20 8:00 AM	Mon 09/11/20 5:00 PM	
7	4.1.2 Implement Login Function	3 days	Mon 09/11/20 8:00 AM	Wed 11/11/20 5:00 PM	
8	**4.2 Create Account**	**4 days**	**Thu 12/11/20 8:00 AM**	**Tue 17/11/20 5:00 PM**	5
9	4.2.1 Create and Design Sign Up Button	1 day	Thu 12/11/20 8:00 AM	Thu 12/11/20 5:00 PM	
10	4.2.2 Implement Registration Function	3 days	Fri 13/11/20 8:00 AM	Tue 17/11/20 5:00 PM	
11	**4.3 Forgot Password**	**4 days**	**Wed 18/11/20 8:00 AM**	**Mon 23/11/20 5:00 PM**	8
12	4.3.1 Create and Design Button	1 day?	Wed 18/11/20 8:00 AM	Wed 18/11/20 5:00 PM	
13	4.3.2 Implement the Function	3 days	Thu 19/11/20 8:00 AM	Mon 23/11/20 5:00 PM	
14	**4.4 LogOut**	**3 days**	**Tue 24/11/20 8:00 AM**	**Fri 27/11/20 5:00 PM**	11
15	4.4.1 Create and Design LogOut Button	1 day	Tue 24/11/20 8:00 AM	Tue 24/11/20 5:00 PM	
16	4.4.2 Implement LogOut Function	3 days	Wed 25/11/20 8:00 AM	Fri 27/11/20 5:00 PM	
17	**4.5 Database Integration**	**5 days**	**Mon 30/11/20 8:00 AM**	**Fri 04/12/20 5:00 PM**	14
18	4.5.1 Connect the Project to Firebase	2 days	Mon 30/11/20 8:00 AM	Tue 01/12/20 5:00 PM	
19	4.5.2 Authentication & confirmation	3 days	Wed 02/12/20 8:00 AM	Fri 04/12/20 5:00 PM	
20	**5. Second Activity**	**7 days**	**Mon 07/12/20 8:00 AM**	**Wed 16/12/20 5:00 PM**	17
21	5.1 Create Contact Us Button	1 day	Mon 07/12/20 8:00 AM	Mon 07/12/20 5:00 PM	
22	5.2 Create Review Button	1 day	Tue 08/12/20 8:00 AM	Tue 08/12/20 5:00 PM	
23	5.3 Create Categories Button	2 days	Wed 09/12/20 8:00 AM	Thu 10/12/20 5:00 PM	
24	5.4 Design all Buttons	2 days	Fri 11/12/20 8:00 AM	Mon 14/12/20 5:00 PM	
25	5.4 Link all Buttons to their Activities	2 days	Tue 15/12/20 8:00 AM	Wed 16/12/20 5:00 PM	
26	**6. Dropdown Menu**	**2 days**	**Thu 17/12/20 8:00 AM**	**Fri 18/12/20 5:00 PM**	20




20 Dec '20
27 Dec '20
03 Jan '21
10 Jan '21
17 Jan '21
24 Jan '21
31 Jan '21
W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

	Task Name	Duration	Start	Finish	Predecessors
26	**6. Dropdown Menu**	**2 days**	**Thu 17/12/20 8:00 AM**	**Fri 18/12/20 5:00 PM**	**20**
27	6.1 Home	1 day	Thu 17/12/20 8:00 AM	Thu 17/12/20 5:00 PM	
28	6.2 LogOut	1 day	Fri 18/12/20 8:00 AM	Fri 18/12/20 5:00 PM	
29	**7. Mid Point Submission**	**2 days**	**Mon 21/12/20 8:00 AM**	**Tue 22/12/20 5:00 PM**	**26**
30	7.1 Implementation, Documentation & Presentation	2 days	Mon 21/12/20 8:00 AM	Tue 22/12/20 5:00 PM	
31	**8. Contact Us Activity**	**5 days**	**Wed 23/12/20 8:00 AM**	**Tue 29/12/20 5:00 PM**	**29**
32	8.1 Create Activity	1 day	Wed 23/12/20 8:00 AM	Wed 23/12/20 5:00 PM	
33	8.2 Implement the Activity	4 days	Thu 24/12/20 8:00 AM	Tue 29/12/20 5:00 PM	
34	**9. Review Activity**	**4 days**	**Wed 30/12/20 8:00 AM**	**Tue 05/01/21 5:00 PM**	**31**
35	9.1 Create Activity	1 day	Wed 30/12/20 8:00 AM	Wed 30/12/20 5:00 PM	
36	9.2 Implement the Activity	4 days	Thu 31/12/20 8:00 AM	Tue 05/01/21 5:00 PM	
37	**10. Book Activities For Different Subjects**	**23 days**	**Wed 06/01/21 8:00 AM**	**Fri 05/02/21 5:00 PM**	**34**
38	10.1 Create the Activities	3 days	Wed 06/01/21 8:00 AM	Fri 08/01/21 5:00 PM	
39	10.2 Implement the functions	20 days	Mon 11/01/21 8:00 AM	Fri 05/02/21 5:00 PM	
40	11.Splash Screen	3 days	Mon 08/02/21 8:00 AM	Wed 10/02/21 5:00 PM	
41	**12. Add to Cart**	**7 days**	**Thu 11/02/21 8:00 AM**	**Fri 19/02/21 5:00 PM**	
42	12.1 Create the Button and the Activity	2 days	Thu 11/02/21 8:00 AM	Fri 12/02/21 5:00 PM	37
43	12.2 Implement the Function	5 days	Mon 15/02/21 8:00 AM	Fri 19/02/21 5:00 PM	
44	**13. Proceed with Payment**	**13 days**	**Mon 22/02/21 8:00 AM**	**Wed 24/02/21 5:00 PM**	**41**
45	13.1 Create the Button and the Activity	3 days	Mon 22/02/21 8:00 AM	Wed 24/02/21 5:00 PM	
46	**13.2 Implement the Function**	**5 days**	**Thu 04/03/21 8:00 AM**	**Wed 10/03/21 5:00 PM**	**44**
47	13.3 Additional tasks	5 days	Thu 04/03/21 8:00 AM	Wed 10/03/21 5:00 PM	
48	14. Testing the App	4 days	Thu 11/03/21 8:00 AM	Tue 16/03/21 5:00 PM	46
49	15. Technical Report	30 days	Wed 17/03/21 8:00 AM	Tue 27/04/21 5:00 PM	48
50	16. Preparing Video Presentation	5 days	Wed 28/04/21 8:00 AM	Tue 04/05/21 5:00 PM	49
51	17. Upload Final Deliverable	1 day	Fri 07/05/21 8:00 AM	Fri 07/05/21 5:00 PM	


1
10 Jan '21
17 Jan '21
24 Jan '21
31 Jan '21
07 Feb '21
14 Feb '21
21 Feb '21


07 Mar '21
14 Mar '21
21 Mar '21
28 Mar '21
04 Apr '21
11 Apr '21
18 Apr '21

7.2. Reflective Journals

Reflective Journal for October

Not a lot has been completed so far in terms of the project, other than submitting the pitch video. I was not sure what to pick for this project, so I got inspired from previous projects where lots of books and journals were required to use as references. That' s how I decided to do a project on an online library exclusive for students. However, I am still waiting for it to be approved, so I am in "stand-by" mode.

As this is an individual and long term project, meaning the whole academic year long, I am quite nervous and somewhat stressed as I don t have much of an idea of what is expected in terms of effort. Also attending online only classes increases the pressure as I do not feel as connected or in touch with the others as I used to.

Now I have been appointed a supervisor my plan is to get in touch with her. In case my project will not be approved, or it requires changes I will ask my supervisor for guidance. Hopefully, I will get an answer in the following days as I have to write the project proposal which is due next Sunday.

Reflective Journal for November

Once my project was approved I started to sketch and brainstorming the application by putting on paper the steps, procedures, and timeline in a reasonable and manageable way in order to be able to finish the app on time.

Initially, I have created the first activity on my application where I set as a background a lovely picture which was downloaded from a free source as required. On the first page I created the following buttons:

- Registration button
- Login button
- Forgot Password button

After I designed the buttons I created activities and linked each button to their appropriate pages. Next step I did was to implement the function for each activities and to design the pages by setting the font type, size and colour for the text and also set images.as background. Create Account – created and designed the Sign up button, implemented registration function

Login - Created and designed the Login Button, Implemented the login function,
Forgot Password – Created and designed the button, Implemented the function
Logout - Created and designed the button, Implemented the function
Database Integration – Connected the project to Firebase, where all the user information will be stored.

Reflective Journal for December

Having connected the application with Firebase, the next step was to set an email confirmation which will be send automatically by the system after every registration. Then I created the second activity where I designed and implemented a dropdown menu which includes Logout, About Us page, Review page and Contact Us page. When click on any of them the user will be redirected to that specific activity which was also created, and the function implemented. On the same page I set a background picture with an inspiring text on it. I also added a button and created the activity named Book Categories, which opens another page allowing the users to select different types of books. In the meantime, I worked on the requirements specifications doc and also prepared the mid-point presentation. Finally, I managed to record my presentation, to finalize the report and submit it on time.

Reflective Journal for January

As the Book Categories page was created in the previous month I started to design the page and created activities for the following buttons:

- Java
- Database
- Data Analysis
- Data Warehousing
- Cyber Security
- Python
- C++

Next thing I did was to link all the above mentioned buttons to their relevant pages.
On this page I have also added a drop down menu containing the Logout function, About Us, Review and Contact Us, so, the users will be able to logout, to give feedback or to contact us

from this page as well. During this month, a lot of research was done regarding the application I am working on.

Reflective Journal for February

Designed and implemented the splash screen. As planned from beginning of the project and also mentioned in the mid-point presentation I created a splash screen which contains a picture and text. This splash screen will open the app and will remain on the screen for 5 seconds and then the users will be directed to the main activity where they can either login or register to the app. At this time, I managed to have the applications working with no errors. I have also created the change language button and working on its functionality however, this function can be completed only when the app is completed.

I started doing research about literature review which I am planning to add in my final paper. While necessary and helpful I find it very time-consuming, which adds on to the pressure of completing tasks in time.

Reflective Journal for March

Continued to work and implementing the Change Language functionality. This function requires lots of translations from English to different languages which is very time consuming. I have also worked on the add to cart function and proceed with the payment function. At this stage I have tried to integrate Kotlin language into my app as it is easier to use and requires 20 % less code comparing with Java but unfortunately I encountered difficulties and got errors.

During this month I have also started to work on my final paper, looking on the template provided by the college and having an idea what are the points that I need to focus on. Another thing was to look up for literature review, so I continued the marketing research regarding the topic my application is about. I have also added staff and polished my CV which was a college requirement and worked on my Showcase Profile as they had to be done by the end of the month. In order to complete the showcase profile, I had to create a LinkedIn account and to populate the fields such education, work placements and so on.

7.3. Other materials used

The type of survey conducted was a small questionnaire. Due to lack of time and ability to interact during quarantine restrictions. This was sent or presented via phone text, calls with friends, acquaintances or via Teams with my college mates. Questions were created according to relevancy for users, as that would enable me to learn what their preferred priorities are, and thus create or improve the quality of the app, based on that. The answers reflected clear preferences and standards of users when deciding on app use/download. I concluded that the results categorized reported preferences according to user friendliness, this being a top-priority, safety storage of user details and attractive design. Also, design would ideally be well structured to avoid time spending and hassle while navigation.