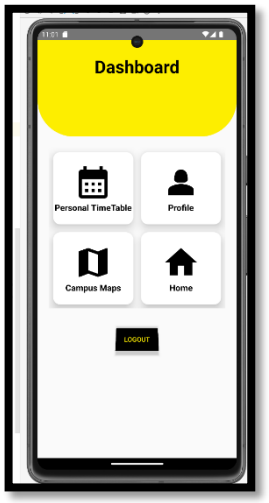
Module Code: CS2PJ20
Assignment report Title: Android Java
Date (when the work completed): 19/03/2024.
Actual hrs spent for the assignment: 22 hours.
StudentID - 31016067

Contents

Overview

For my project, I decided to create a mobile app for university students. The purpose of this was to provide useful features to students that might help them at university. The objective of this was to create an easy-to-use app which would allow all students to stay organised and informed. In order to achieve this, the first basic functionality I implemented was user authentication through the use of firebase which allows the students to login to their account or register for an account. The second basic functionality I implemented was personalised timetable and to-do-list which would allow the students to remain organized as they can see their timetable and set tasks from them to accomplish. The first additional feature I implemented was campus maps which has images of all three campuses and google map links for directions which should allow students to navigate the university better. The second additional feature implemented a personalised profile for the student which has their personal details and a button to view their module with extra information about the module. In this report I will demonstrate screenshots of the features and code along with how I implemented them.

Application Specification



When creating my application, I used different technical specifications such as functional components, which involved User Interface Components like Activities, fragments, views such as button, Text views, Edit views, List views, Recycler view and Card view, also different layouts like linear layouts, relative layout, constraint layout and frame layout which are important part of the apps development. I used basic algorithms functions for user authentication this was done using firebase authentication. This involves integrating Firebase SDK, enabling authentication methods which in my case was for email/password and using Firebase AUTH to manage user sessions and access. For my user interface for the application, I used card view to make it easier to navigate. All the functions of the app are displayed when you first login and are easy to access. I connected different card views by giving them Id and using intent to connect them together (as seen in figure 3).

Application Implementation

Basic functionality 1 - Login and registration using Firebase.

The login and registration feature are performed using Firebase Authentication, which provides a secure and simple way for students to access your application. This feature is critical for providing a personalized and secure user experience since it ensures that each student's data and user session remain private and safe.
For students, the ability to login and register securely is essential. It not only secures users personal information, but it also improves the app's usefulness by allowing access to personalized features such as course schedules, grades, and any content that requires user identification. It ensures that students can access their information on any device, at any time, without fear of unwanted access to their accounts.
Registration: When a user completes the registration form and hits the register button, the app captures their email address and password. It then uses Firebase's createUserWithEmailAndPassword method. If the registration is successful, Firebase securely saves the user's credentials and may provide a user object. The user is then taken to the dashboard.
Login: For logging in, the process is similar. The user enters their email and password, and the app calls Firebase's signInWithEmailAndPassword function. Firebase verifies the credentials and, if correct, allows the user access. The app then sends the user to the dashboard, delivering a more tailored experience depending on their information. Firebase handles session management automatically. When a user logs in, Firebase Authentication saves their session, allowing them to remain logged in even if the app is stopped or the device is restarted.

TextInputEditText boxes within TextInputLayout was used in registration and login for user interfaces to allow users to input their email address and password. (Firebase screen shots – figure 8 and 9)

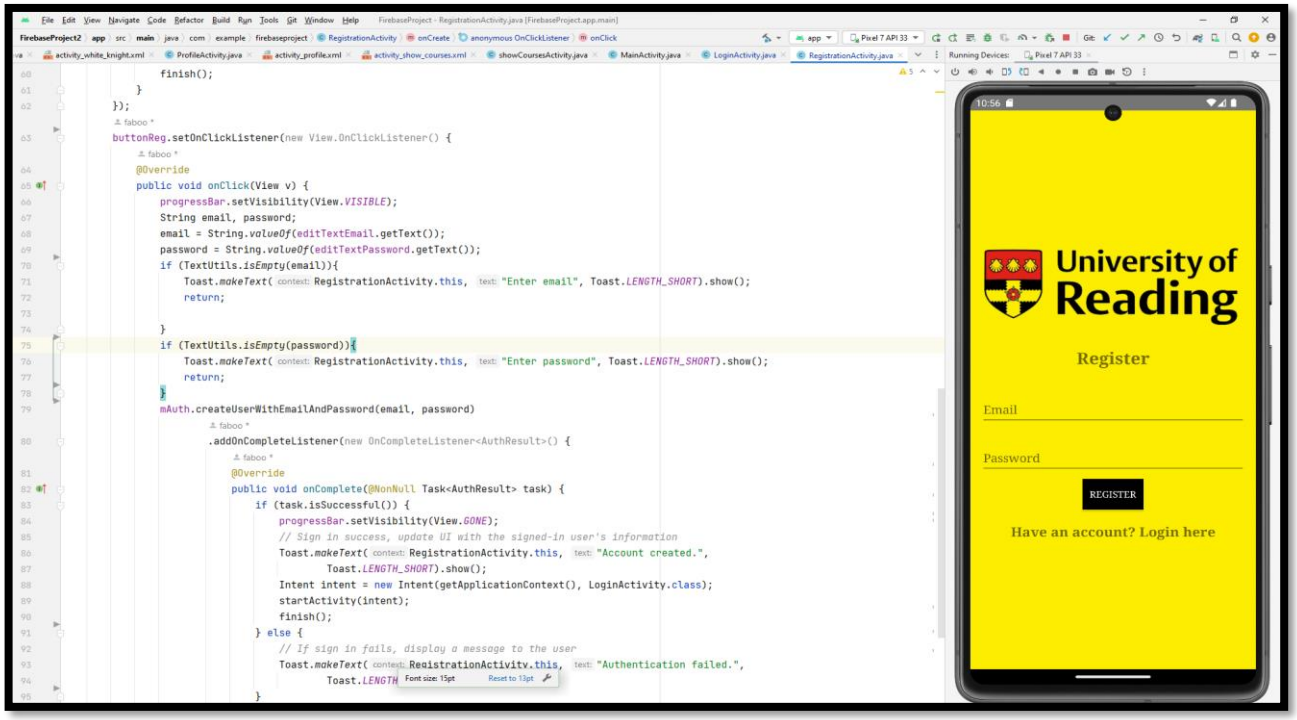
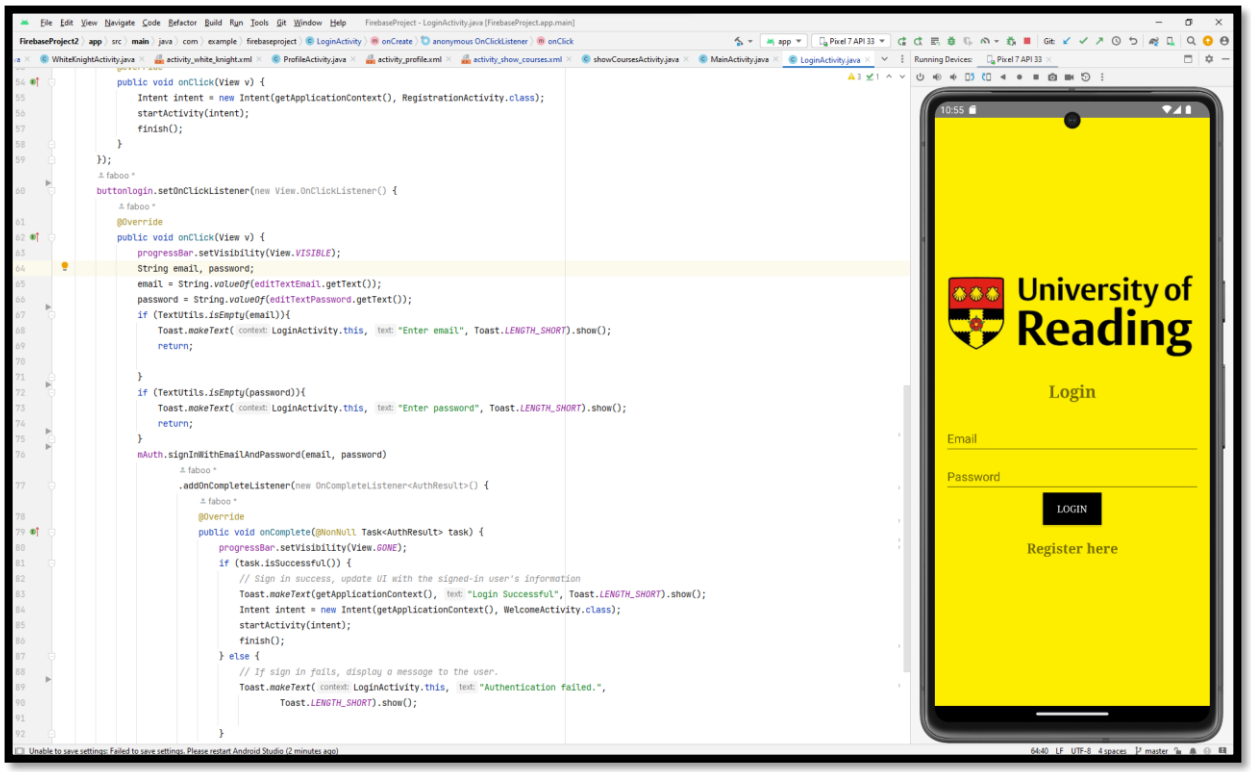
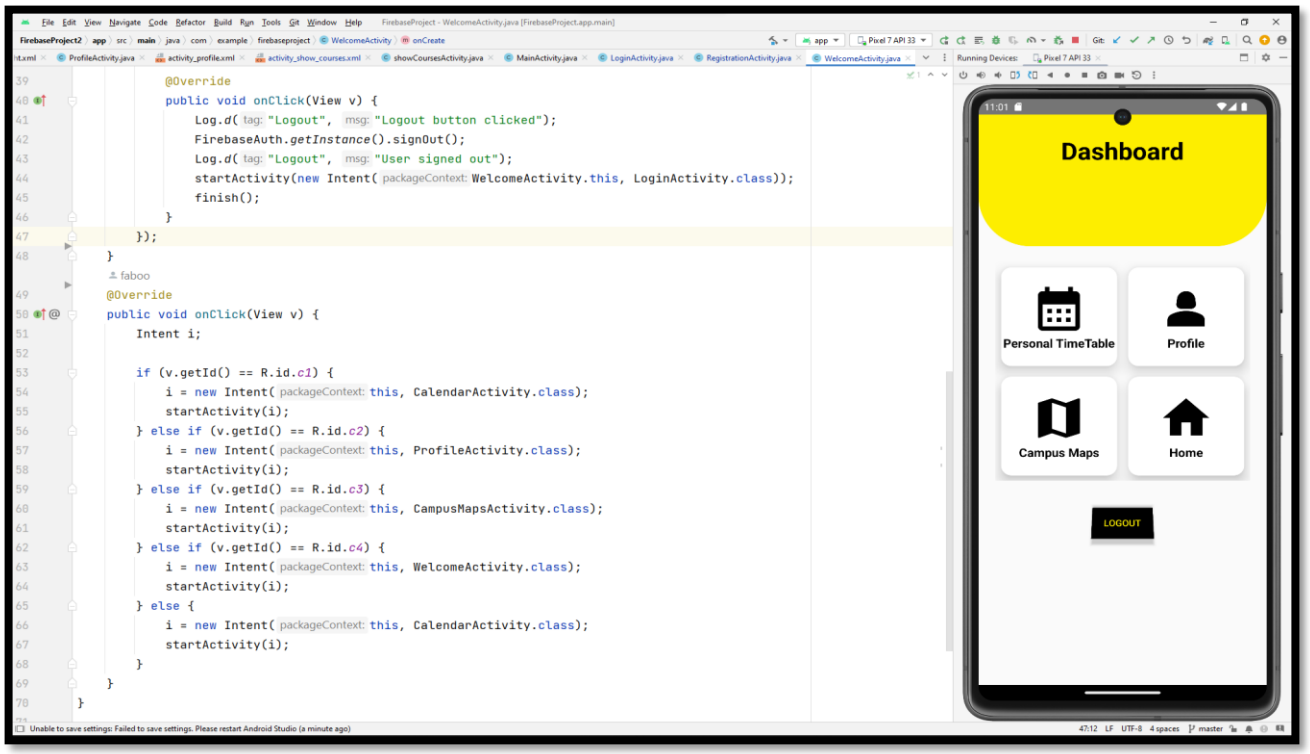


Figure 1



Register – Figure 2 and 3 (home page)

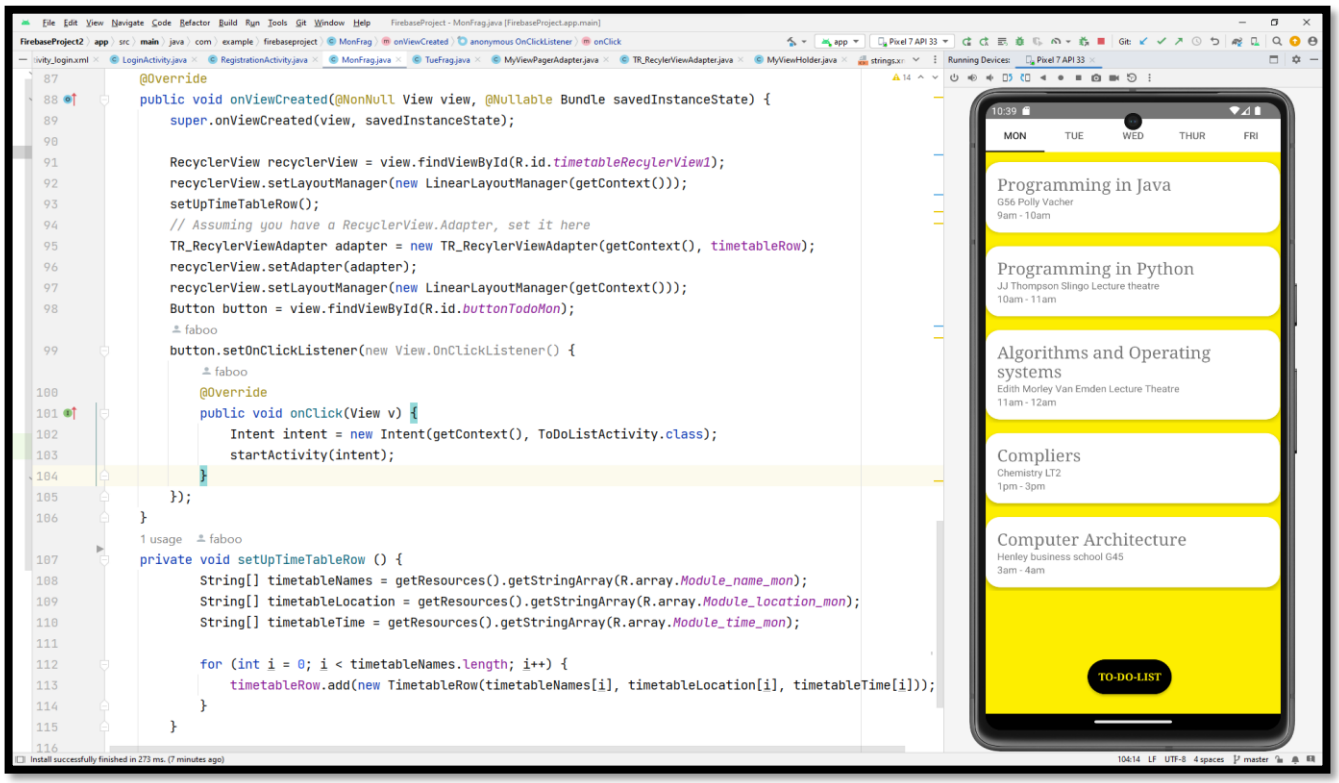


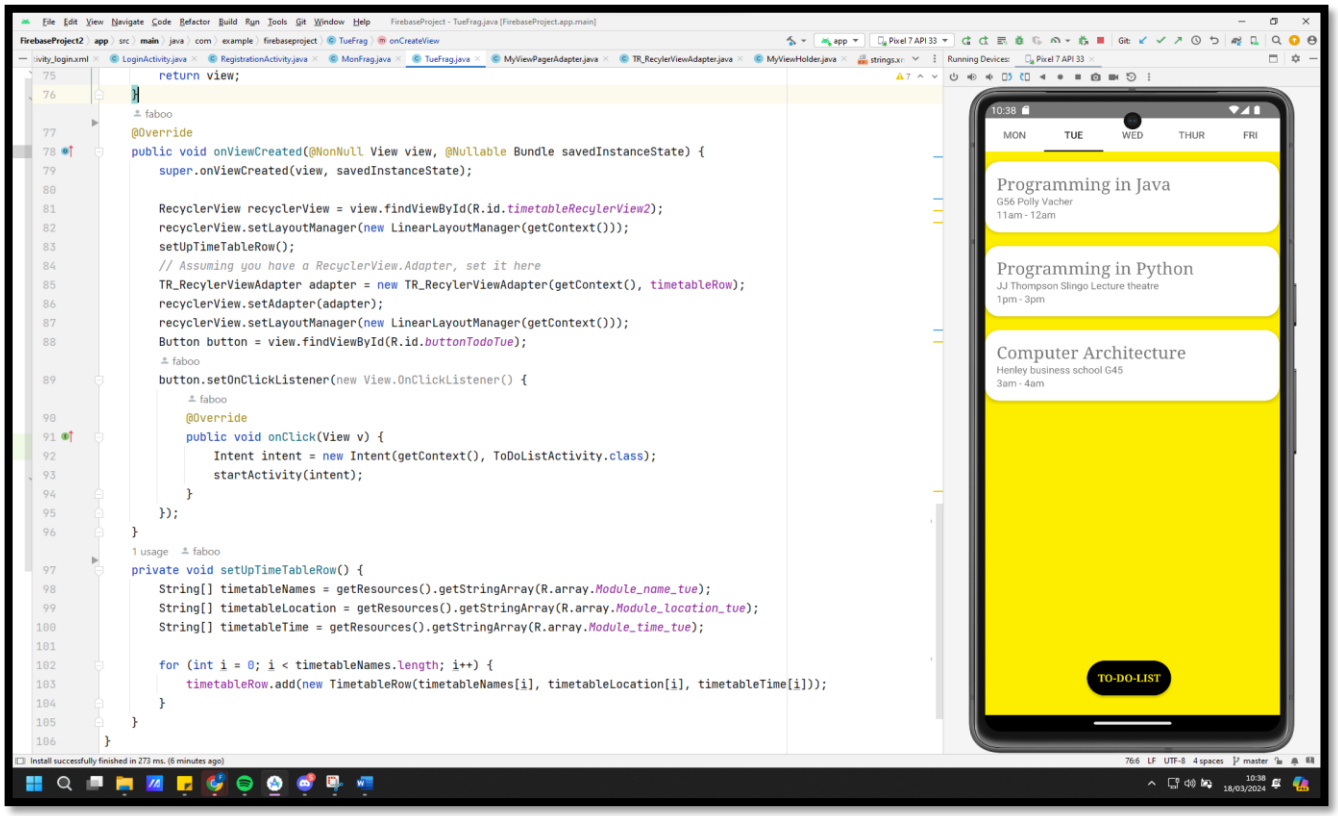
Basic functionality 2 - Personalised Timetable with To-do-list.

The feature allows for users to have personalised timetables which contains the module name, time and location for each day. And allows users to add to a to-do list and remove when they are done with the task. These features are useful as they allow users to have information about the lectures and practical's they must attend. And the to-do-list allows the students to stay organized by setting tasks and removing tasks when done.

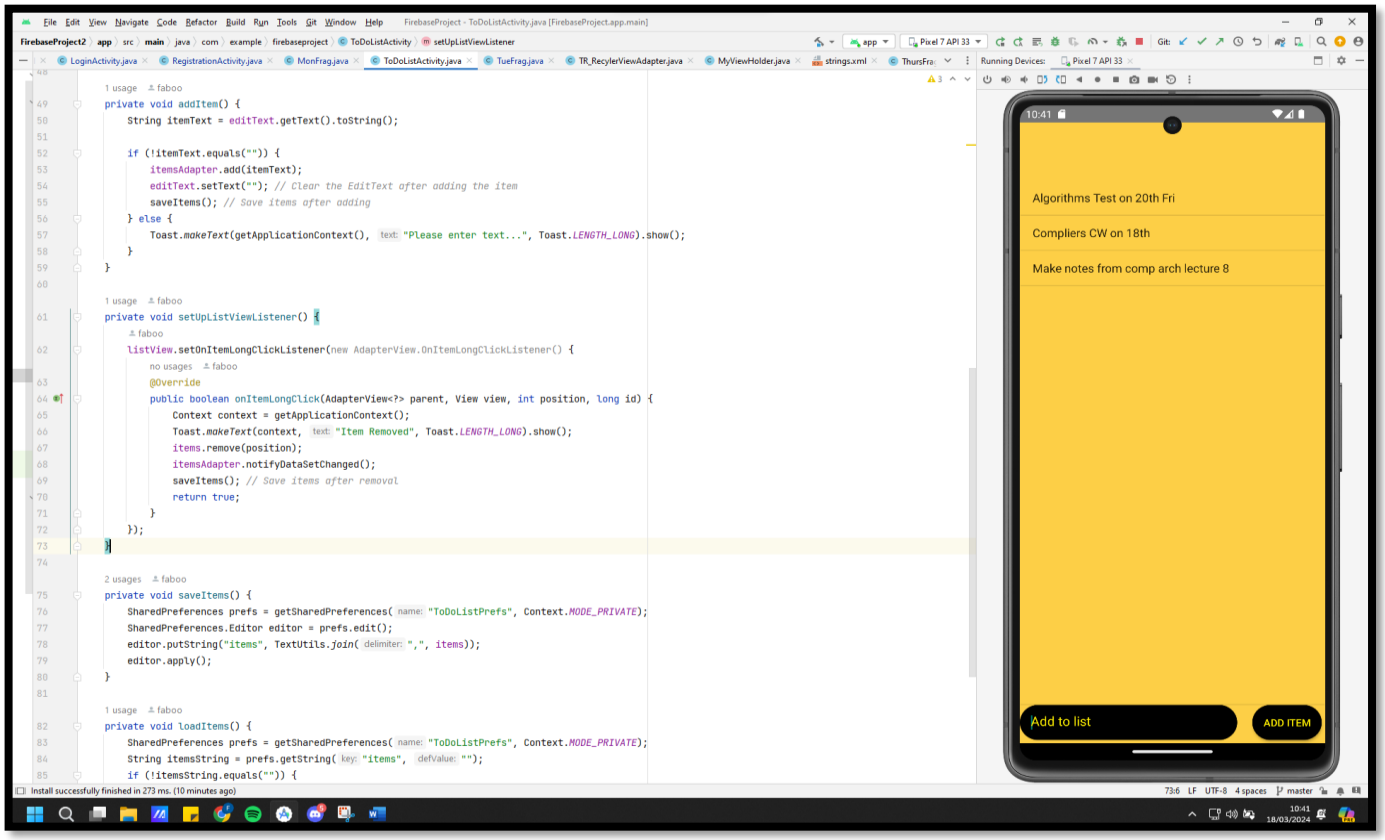
For the timetable I used fragments, card view, recycler view, etc to implement. The ViewPager2 and TabLayout, enable the display and navigation of the various days of the week, Monday through Friday. Users can easily switch between the days since every day is shown as a fragment (MonFrag, TueFrag, WedFrag, ThursFrag, FriFrag) within the ViewPager2. In addition, the MyViewPagerAdapter class which is an extension of FragmentStateAdapter loads the correct fragment dynamically according to the day that is swiped to or chosen. Within the CalendarActivity, where the ViewPager2 and TabLayout are also set up to be in sync with one another, this adapter is initialized and set. To ensure a smooth user experience, the ViewPager2 adjusts to show the relevant day's fragment when a tab is chosen, and vice versa. A list of timetable items or events for each day is displayed using a RecyclerView within each fragment, like MonFrag. A custom adapter called TR_RecylerViewAdapter is used to populate the RecyclerView with the timetable elements, which are represented by the TimetableRow class. Not only does this arrangement make the daily schedule visually appealing, but it also makes it simple to update and adjust the program. Additionally, each fragment may be able to manage its own distinct features, such adding, deleting, or amending events, which would further improve the calendar's customizable feature. The fragments use arrays provided by resources(values->strings.xml) to load various timetable information (e.g., module names, locations, and times) to help customize each day's schedule. This way, each day may display a different set of entries based on the timetable. Additionally, the application features a button within each day's fragments to navigate to a to-do list, to improve the app's focus on planning and organization. The ToDoListActivity allows users to maintain a basic to-do list. When the activity is launched by pressing the button in fragments, previously saved to-do items are loaded from SharedPreferences and shown in a ListView. By inputting text in an EditText field and selecting the "Add Item" button, users can add new entries to the list. Once added, every item is stored to SharedPreferences so that it will remain there even when the program restarts. Using the onItemLongClick function, which initiates a removal confirmation, items can also be removed from the list by long-pressing them. This offers a user-friendly way to track tasks needing completion.

Figure 4(MON) and Figure 5(TUE)

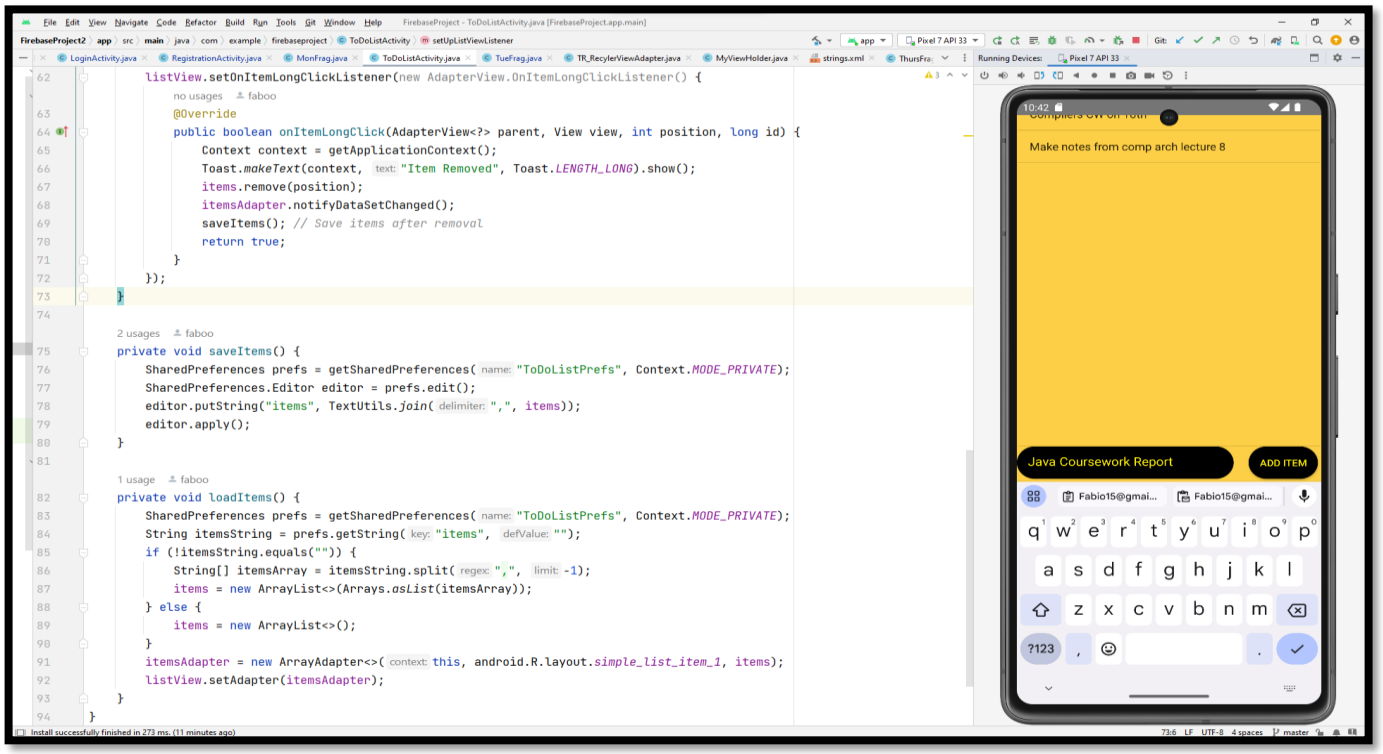




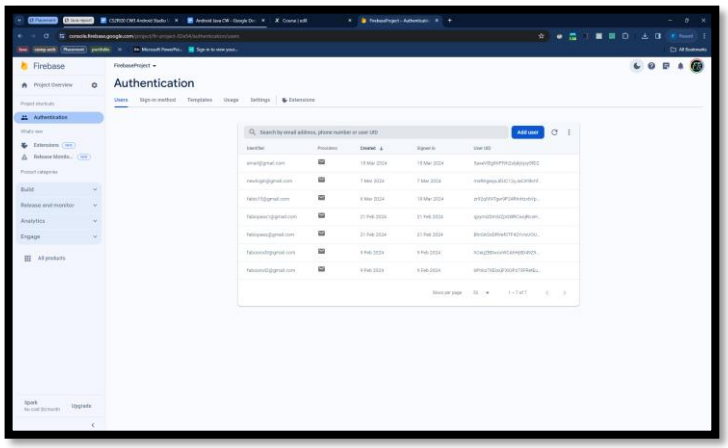
To-do-list – Figure 6

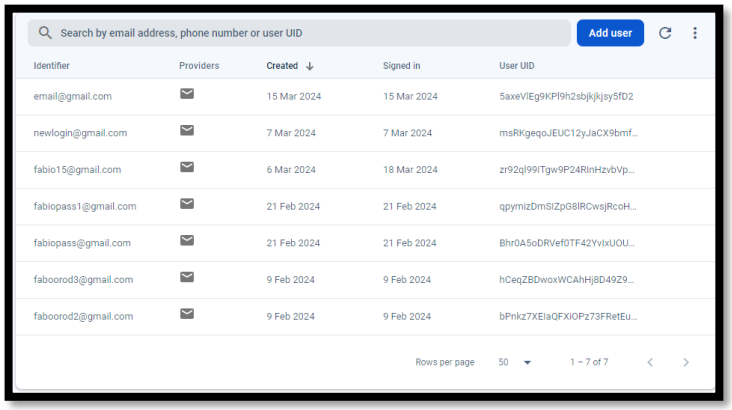


Add Item – Figure 7



Authentication Figure 8 and Figure 9 on firebase

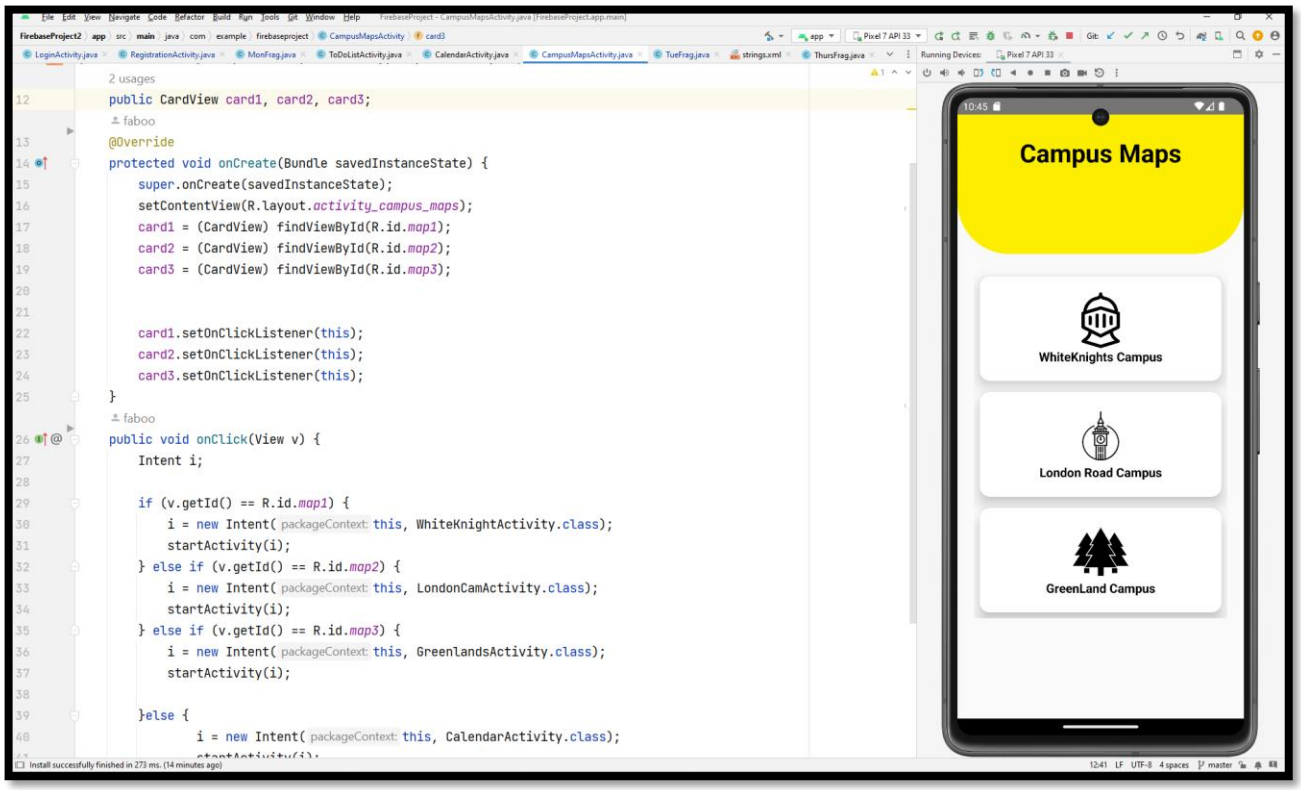




Additional features 1 - Campus maps with Google maps links

The campus maps feature is intended to make it simple for students to move between Reading University's many campuses. This feature makes it easier for new students to navigate the campus and cuts down on the time they spend looking for buildings or facilities. It is especially helpful for new students who are not familiar with the layout of the university. The implementation includes a CampusMapsActivity that shows a main screen with clickable CardView elements representing various campus locations (e.g., Whiteknights Campus, London Road Campus, Greenlands Campus). These cards are placed in a GridLayout to provide a visually organized presentation. When you tap a card, the app launches a new activity tailored to the campus you've selected. For example, picking the Greenlands Campus card launches the GreenlandsActivity, which displays a map of the campus in a HorizontalScrollView, making it simpler to view. This activity also contains a feature that allows you to view the campus on Google Maps, which is made possible by a button that opens a web link to the campus's location via an Intent action with a URI referring to Google Maps. This provides students with detailed geographic visualizations, directions, and other relevant data regarding campus geography, allowing them to better navigate and move around the university. This campus maps section, provides an interactive guide to the university's layout and also integrates external mapping services for comprehensive navigation support, making it an essential tool for the student.

Figure 10 and Figure 11(Scrollable view)



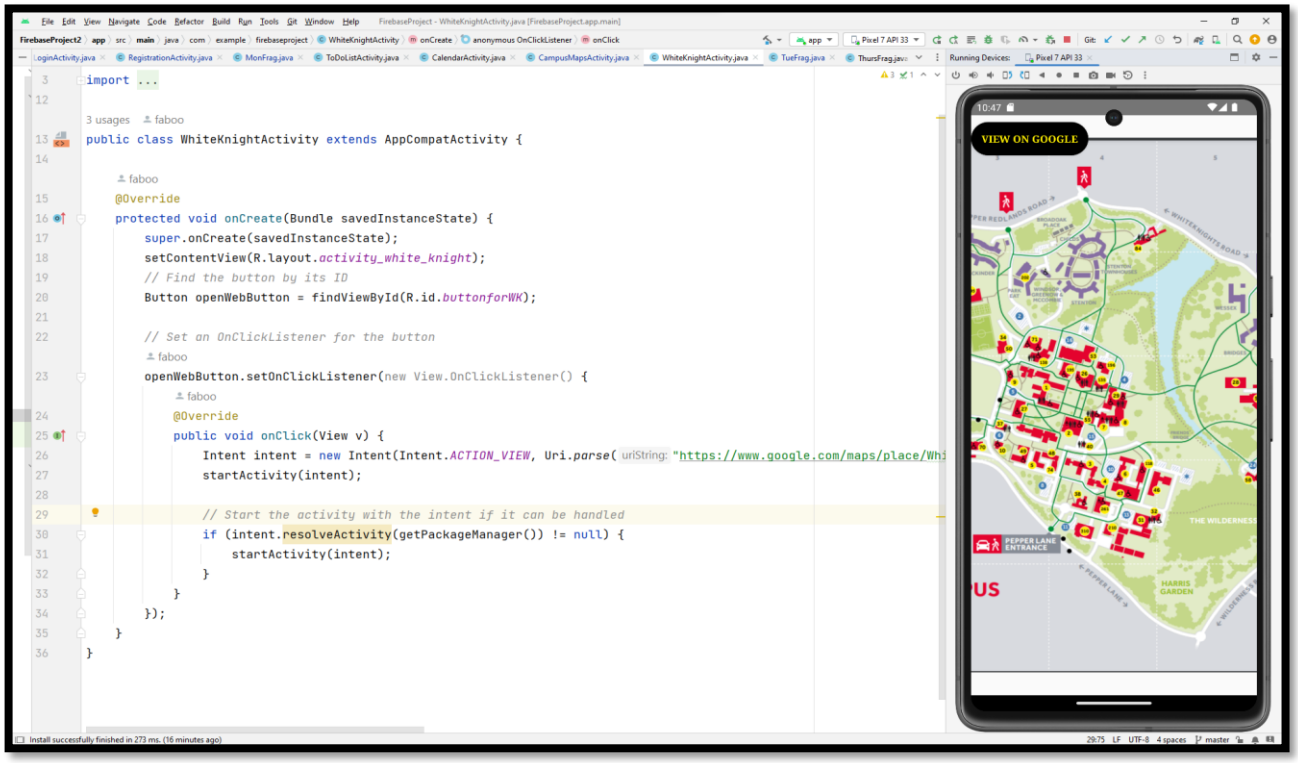
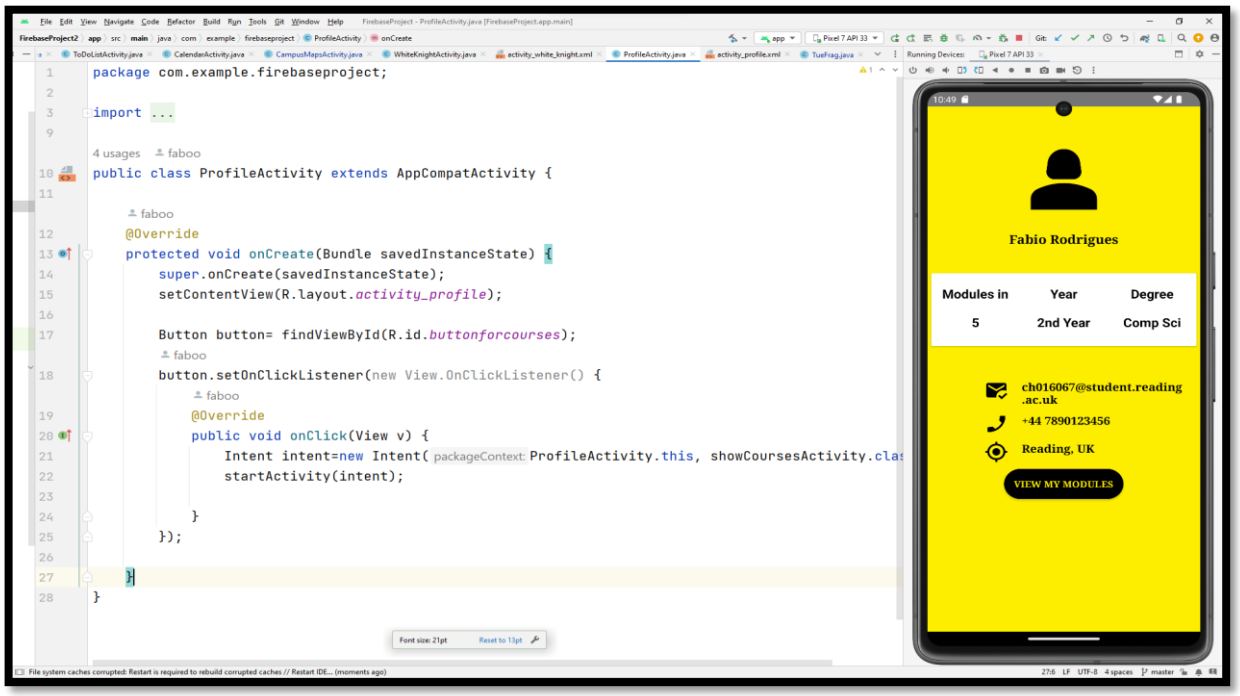


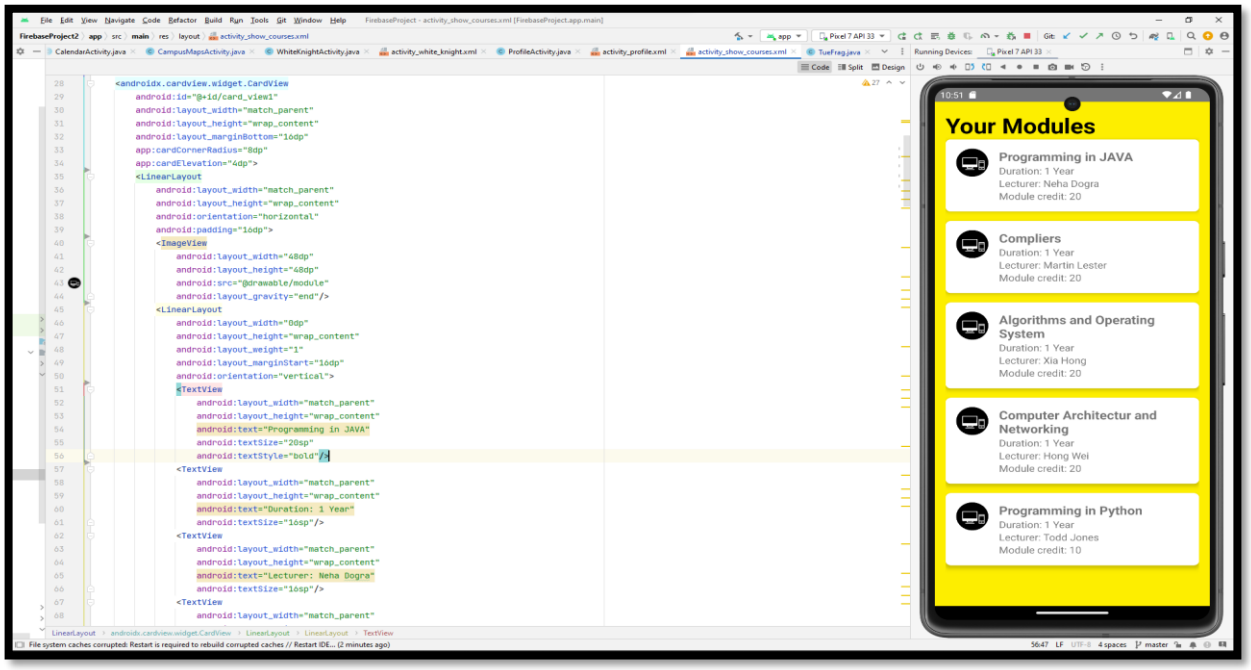
Figure 12 (Button links to google map)

Additional features 2

The apps personal profile feature provides a single place for students to easily access and manage their academic information. It's a valuable tool that improves the student experience by offering a summary of their academic status, such as course enrolments, year of study, degree and contact information.
When students access the ProfileActivity, they are met with a visually appealing layout that contains a profile image, their name, academic facts such as modules enrolled, year of study, and degree program, as well as contact information such as email, phone number, and location. This combination of personal and academic information allows students to keep track of their academic achievement and personal data in one location, making it easier to manage their academic lives.
The functionality is implemented through the ProfileActivity and its XML file, which provide the profile layout and feature a button that, when pressed, takes the user to the showCoursesActivity. This activity presents specific information about the student's enrolled courses, such as course names, durations, lecturers, and credits, all presented in individual CardView widgets for a clean and orderly design. This functionality not only helps with academic planning and management, but it also creates personalization, making it a useful resource for students navigating university life.

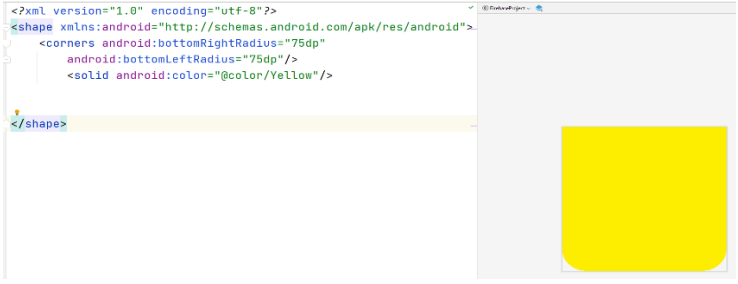
Personal profile – Figure 13 and Figure 14(View my module button)





Design Elements

1)I created this drawable for the view you see in dashboard and campus maps.



2) I created this drawable for my buttons so I can reuse it for different buttons.

3) For my colours I used the following code so I could reuse them.
4) For my font, I mostly used serif.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="Red">#DC143C</color>
    <color name="Yellow">#FDEE00</color>
</resources>
```

Conclusions and Future Work

This mobile application for university students was successfully developed, resulting in a tool that improves its users' academic and personal lives. The addition of core functionalities such as secure login and registration via Firebase, personalized timetables, to-do lists, campus maps with Google Maps integration, and personal profiles has resulted in a comprehensive platform that serves as an essential companion for students navigating university life. The app delivers a seamless, safe, and user-friendly experience that matches the project's goals. Reflecting on this project, I gained a better understanding of numerous key principles in Android app development and Firebase integration. I learnt how to establish secure user authentication, which is essential for securing user data. Another major lesson was the ability to design dynamic, user-friendly interfaces using RecyclerViews and Fragments, which allowed for customizable timetables and to-do lists. Integrating Google Maps for campus navigation shows the potential of external APIs to improve app functionality. Overall, this project not only helped me improve my technical skills, but it also demonstrated the value of user-centric design and the practical usage of cloud-based services in the development of excellent mobile applications.

Moving forward, there are many possibilities for continued development and improvement of the application. One feature that would improve the app would be real-time updates and reminders for schedule changes, assignment deadlines, and exam schedules. And adding a feature that allows instructors to communicate directly with students through the app could increase its utility. Another feature I would have like to add would be Augmented Reality Campus Tours to create an augmented reality feature that allows new students to take interactive campus tours, familiarizing them with university.