Megan Cash

C19317723

Submitted in partial fulfilment of the requirements for the degree of

BSc in Business Computing

Technological University Dublin (City Campus)

Year 4

Supervisor: Neil O’ Connor

April 2023

Declaration

This is an original work. All references and assistance are acknowledged.

Signed: *Megan Cash*

Date: 28/04/2023

Table of Contents

Acknowledgements

I would like to give a special thanks to my family, friends, classmates, and lecturers for their continuous unconditional support, without whom I would not have been able to develop this application, and without whom I would not have made it through my bachelor's degree in business computing!

Firstly, I would like to thank my family and friends who have been so supportive and who have helped keep me motivated throughout all my education. They have been with me through every step of the way, especially by keeping me focused on my studies throughout the pandemic. I would not have achieved all that I have today without them.

Secondly, I would like to thank all my lecturers that I have had over the past four years. I have learnt so much valuable knowledge in every module that I have had, all of this helped the development of my Project. I firstly would like to thank Neil O' Connor for his continuous support throughout the duration of this project. I have really appreciated all of his valuable guidance and positive encouragement. I consider myself very lucky to have had Neil as my supervisor. Secondly, I would like to thank my second reader Hung Ngo for his valuable suggestions and input. I want to thank Catherine Higgins for being so supportive and attentive in my first two years of college as I did quite struggle with coding, she was a major reason as to why I became so determined to learn and improve my coding skills. Lastly, a major thanks to Thoa Pham and Jenny Munnelly. I learned a lot about mobile application development from Thoa, which really contributed to my final year project. I also learned a lot from Jenny through her challenging modules.

I also want to express my gratitude to Alan Fahey for giving me the amazing opportunity of being one of twenty VT STAR students in SAP. He has provided me with continuous guidance and support throughout my last two years of college. My experience in this program really helped me, by giving me a good understanding of the technologies and tools that are used in SAP to help support the business, along with dealing with people internally and with customer issues. All of the skills I learned during my internship really helped me with the development of my application.

Finally, I would like to thank all my classmates who have really had a positive impact on me and helped me have the best college experience. In particular, I would like to thank Emma McGuiness, Erika Reilly, Katie Flynn, Yasmin McCann and Nadia Mazgaj.

Abstract

First Year Companion is a mobile application aimed to help with the onboarding of new students beginning college. This application will work as a type of workflow that will lead the student through the things they need to do to integrate and to be onboarded into the university, with a heavy importance on the social aspect of beginning university.

In this report, I will display the applications main features and how these features were designed, and also the technologies involved. I will also discuss any challenges that I have encountered throughout and how I developed each feature.

Chapter 1 - Introduction

Introduction to Project

Before coming into my final year of college, I was constantly thinking of what I could do for my application for my final year project. My brother started college when I was in third year and my sister started college at the start of this year, this brought me back to when I was starting college and all the feelings I had about starting too. I remember feeling really unprepared and didn’t know what to expect. I was particularly anxious about how I would make friends as I did not know anyone else that was going to the Aungier Street campus. This gave me the idea to create First Year Companion.

I began by researching what is available on the current market. From my research I identified multiple mobile applications offered by different colleges within Ireland that aid students in university. However, many of these applications were missing the social aspect and did not focus on how vital the onboarding process is for students.

My main aim of this project idea is that it will help the onboarding of new students to the college. The application will work as a type of workflow that will lead the student through the things they need to do to integrate and to be onboarded into university. The application will also focus on the community aspect. For any student that registered, the students would be able to view everyone on the application and can interact with one another and arrange meet ups, as socialising and making friends is one of the main concerns that new students have. By having this as another way for students to socially interact it can help boost the student’s self-esteem and academic development.

The user can register and login, once logged in to their account a home dashboard will be displayed where the user will have multiple menu tile options. These tile options include the social functionalities such as ‘Friends’ where a list of all users on the system is displayed, they can chat one on one with any of these users. Another tile is the group chat functionality, where they can create a group whether its for a class group, project group or a society group etc. Within these groups they can chat with one another, and they can also arrange meet up events that the participants can accept or decline, if the user accepts this event will then be added to their calendar. The student can access these social functionalities on the navigation panel on the bottom of the screen, through this they can also access the profile fragment and home dashboard fragment.

Other functionalities that aid the students onboarding experience on the home dashboard include timetable, to-do list, calendar, transport options, accommodation, and additional links. This allows students to have all necessary information and functionality in one place. From my own personal experience, these options would have really helped me when I started college. The timetable functionality allows the student to upload their timetable to the application in order to allow them to keep track of what modules they have for the week which will allow them to manage their time more effectively. The to-do list functionality also is used to help the student with their organisation skills, by having a to do list for their college tasks it will give the student a sense of accomplishment and satisfaction once a task is marked off as completed.

The calendar option includes the academic calendar for TU Dublin and all of the key dates marked off in the student’s personal calendar. The transport option allows the student to select which campus they are attending and will inform the student about their campus and the quickest route to get there. Considering TU Dublin does not have student accommodation on campus, the accommodation option provides the students with several links in order to get accommodation for when they attend college. For students that need accommodation, getting this sorted for when they start college is vital as it is important to feel settled in their living environment so that they can alleviate stress and increase their confidence in order to excel in their studies. The additional links option contains other important links that the student may need during their time in college such as their learning platforms and links to where they can access their timetable, exam results etc.

All these functionalities are valuable tools to help students stay organised, set goals, reduce stress, and manage their time effectively. Ultimately, allowing the students to have a balanced student life and an improved academic performance.

Objectives of Project

My objective when developing this project was to create an application that aids new students when onboarding into college. All students have expectations about college life before they have even started their first week. Some students are enthusiastic and eager for their new independence and educational journey. However, other students can be quite anxious beginning college. All new students, regardless of their initial expectations, can encounter obstacles and challenges that they did not expect during their transition into college.

My overall objective was to have all of the relevant needs of the student in one place, with the application being easily navigable.

Users should be able to:

- Register and login to their account
- Reset their password if they have forgotten it
- Logout of their account
- View all of the features on the home dashboard
- Edit their own profile information
- View and search through users on the application
- Individually chat with another user
- Create groupchats
- Search through groupchats list
- Chat within groupchats
- Edit group chat information
- Create an event within a group chat
- Access the transport tile to find the quickest route to their campus
- Access the accommodation tile to access the accommodation links
- Access the calendar tile to add dates, view key dates and academic calendar
- Access the timetable tile to create and store their college timetable
- Access to-do list tile to create to-do list tasks and mark them off when they are completed
- Access additional links tile to view important resources such as Brightspace, Moodle, Publish, TU Dublin library etc.

The system should provide the following:

- Authenticate user for register and login
- A way to access core functionality

Business Case of Project

I feel that this will really benefit new students to the college. Every year universities and colleges around the world welcome new students into their institutions. Starting college can be a very challenging and stressful time for new students to orientate themselves. From viewing multiple college websites many students can face a variety of challenges trying to find all the correct information. Some college websites have an overwhelming amount of information, thus making it difficult for students to know where to start. This can all lead to the students feeling quite overwhelmed and frustrated. Another challenge can be the difficulty navigating their college website, some of the websites can be quite complex and difficult to navigate, therefore making it challenging for students to find the correct information that they need. I feel like I would have benefitted from an application like this one when I started college as this would have eased the overall process of onboarding for me, which I hope will be the same for if other students were to use this application.

The transition to higher level education can be quite stressful. A successful onboarding process is vital for universities in order "to protect and hold their new students" (Schilling et al., 2022). Onboarding is the "process of introducing and orienting new students to their college or university", it is important for colleges to create a "welcoming atmosphere that encourages engagement in educational opportunities and involvement in extracurricular activities, colleges can ensure successful student onboarding experiences" (Bennett, 2023). Ensuring an engaging onboarding experience for the student can provide them with all of the necessary information for them to be successful in their studies and it will also provide them with a positive experience when beginning college. Therefore, this application will be very useful for students to stay organised, create connections with other students and overall ease their onboarding experience.

Business Rules

- User's must register to use the application for the first time
- User's must be logged in to use the application
- Each user must give permission for the application to use their camera or access their photo gallery
- Each use can only have one account associated to their email

Overview of Technologies used in Project

Firebase

Firebase

The database I decided to use for my application was Firebase. I chose to use this as it is in the cloud and a Realtime database, so this suited my application. I also used authentication to provide a scalable and secure solution for implementing user authentication and authorization. Using android and firebase can provide robust backend functionality that can greatly accelerate the development process, simplify the overall integration of cloud services into Android applications and enhance overall application performance.

Firebase Cloud Storage

Cloud Storage
for Firebase

I used Firebase Storage to store any of the images uploaded by the user. Each time a user added or updated their profile/group image this was uploaded to Firebase storage and an image URL is then saved to the Realtime database. I used cloud storage along with the Realtime database and Authentication as it is a flexible cloud-based storage that efficiently stores and manages user-generated content.

Java

Java™

I chose to use Java as it was the programming language that I was most confident using. It was the first language that I had learned in college, and I had a lot of experience writing in it, therefore I felt that I could implement more complex features in my application a lot more easily than if I chose to develop the application with a different programming language.

Android Studio

I chose to do my project as an android application rather than a web application since students would mainly use their mobile phones. I also thoroughly enjoyed Thoa's mobile application development module where we got to learn how to use Android studio. It is the official IDE for Android and is specifically designed for android development. Therefore, these are the reasons I chose to use Android Studio as my integrated development environment for developing my application.

XML

When developing the android application, I used XML for a wide range of things. For example, it is used within my Android Manifest file. It is also used in my application to define the UI layout. Within my application, the layout files were used to arrange the layout of my activities.

GitHub

I used GitHub as my version control system to keep my project up to date in the cloud and also for piece of mind to have it backed up just in case something happened to my laptop.

Chapter 2 – Requirements Capture and Analysis

Business Requirements

Actors

In my application I only have one type of actor, the User (which is the student).

User
Once the student has registered on the application, the student is then stored on the database. When registering, the student must provide their email and password. After registering, the student can login with their email and password that has been authenticated by the system.

When the user is logged in, they will be directly brought to the home dashboard where they will be able to view all eight menu tile options. Two of these tiles include the social aspect of the application, these can also be accessed in the navigation menu at the end of the screen, along with the home dashboard fragment itself and the profile fragment. The six other menu tile options include all of the necessary features that a student will need to help with their onboarding experience.

Transport Option:

The user is presented with buttons that represent all of the TU Dublin campuses. When the user selects a campus, a gallery slide show is shown to give the student an insight to what the campus looks like. Details about the campus is also displayed. Users can input their current location of the campus they want to go to, the are then redirected to maps to which they are shown the quickest route to their destination

Accommodation Option:

The user is presented with multiple card items in a card view layout that links and redirects them to certain accommodation options such as private accommodation (myhome.ie, homestay.com, daft.ie), TU Dublin StudentPad and International student accommodation (Yugo.com and Mezzino.com).

Timetable Option:

The user can create and store their college timetable on the application.

Calendar Option:

The user can view the 2022/2023 academic calendar and add and store key events to their calendar.

To-do list Option:

The user can create to-do list tasks and check them off once they are completed.

Additional links Option:

The user can view all additional link options, this contains multiple card items within a card view with links to important student resources such as Brightspace, Moodle, Publish, TU Dublin library etc.

Friends Option:

Users can view all users on the system and individually chat with one another. They can also search through the user list.

Group chat Option:

Users can search through groupchats, chat within groupchats, create new groupchats to which they can add participants and edit group chat information. Within the group chat they can create group events where invitees can accept or deny the dynamic link and then save this event to their calendar.

Non-Functional requirements

Availability

- The student's device must be connected to the internet to access the user data.

Recoverability

- In terms of recoverability, the user's data is all stored remotely online. This data is also persisted constantly. If the student experiences any issue on their mobile phone, all of their data is readily available the next time the application is launched.

Security

- In order for the student to access the applications functionality, the user needs to be authenticated when logging in to their account. This is to ensure the students date is protected and identifiable.

Usability

- Usability is important in the application to allow the student to navigate through it effectively. The user interface should be easily understandable to improve the user's experience. Also, the time in which it takes for the student to open the application should take only a few seconds and the ability to use the features should be time efficient too. The student also should have maps installed in order to allow the application to get the quickest route to their selected campus. If they do not have this application downloaded, they will be redirected to their play store to download it.

Theme

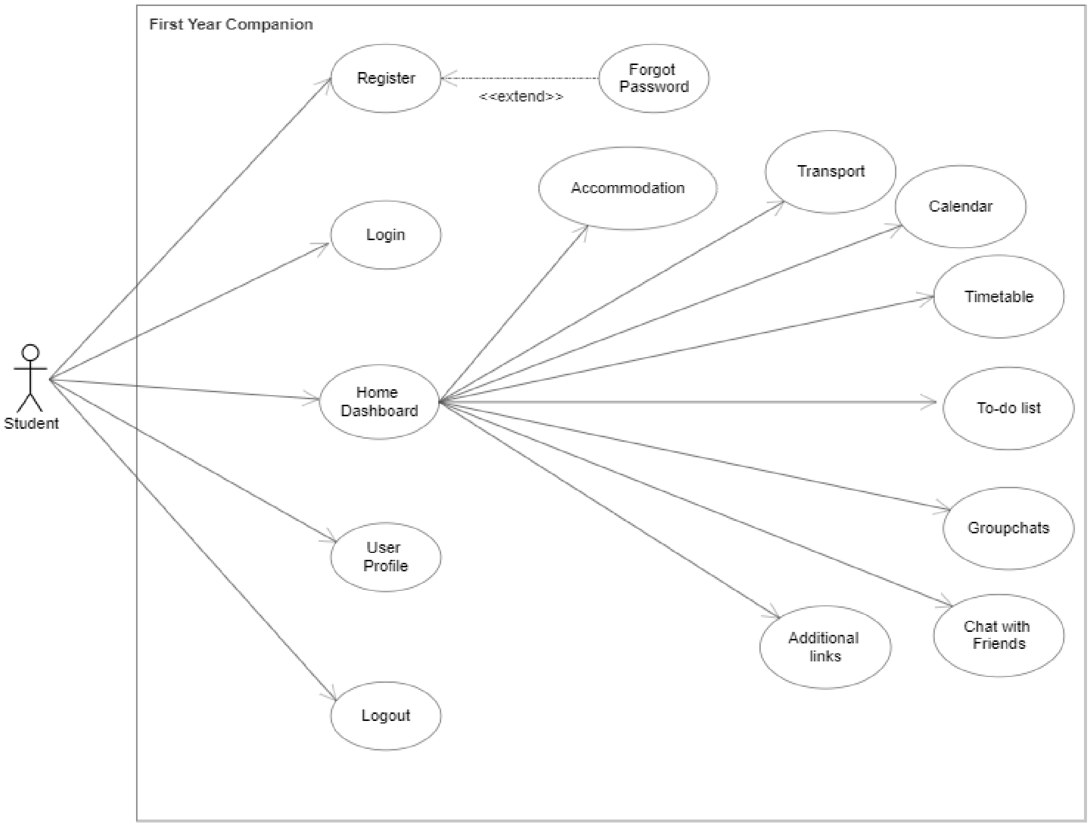
- The user should be presented with the same theme throughout the application when accessing the multiple different screens. The application should have the same common colours, design, and text style throughout. This adds to the consistency of the application and is essential to the overall user experience.

Requirements Analysis

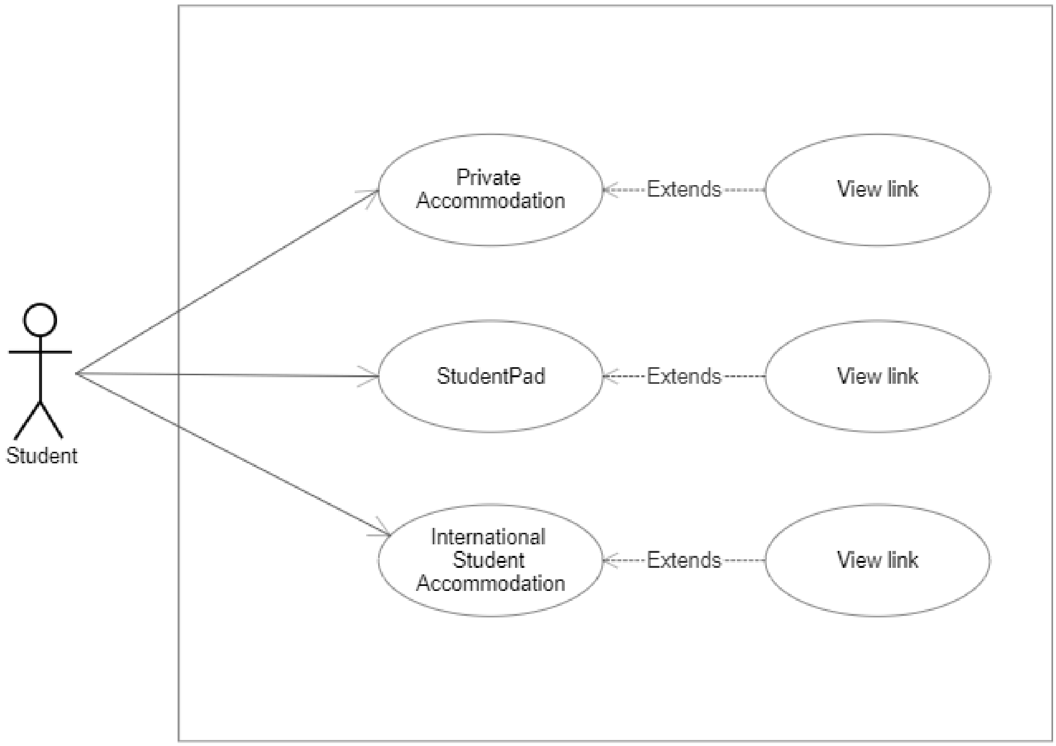
I reached out to all my friends and family to get their perspectives of their onboarding experience when they were transitioning from secondary school to higher level education. I asked what features of an application would have helped them during this time. Taking into consideration that most of their onboarding process was during the pandemic, they all agreed that the social functionality would prove useful in order to connect with other students in the college, they also said that they would like added features to aid in organisation. Essentially, they all agreed that having an application where all of these aids for onboarding are in one place, would be really beneficial.

Use Cases

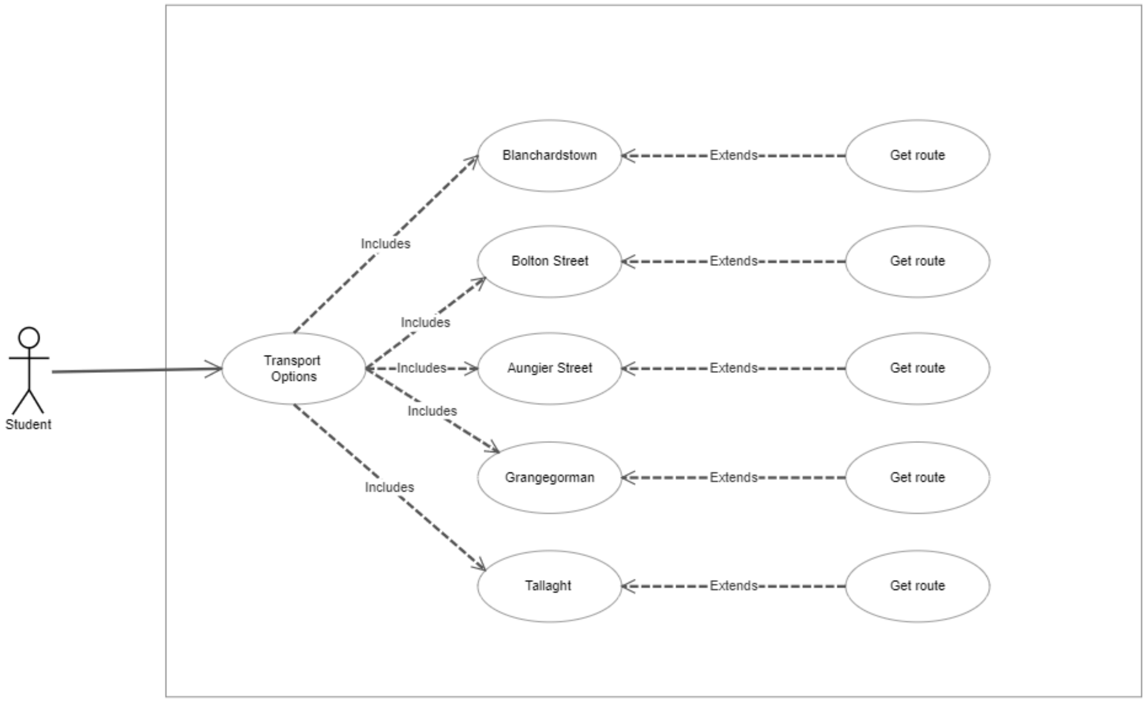
High level analysis



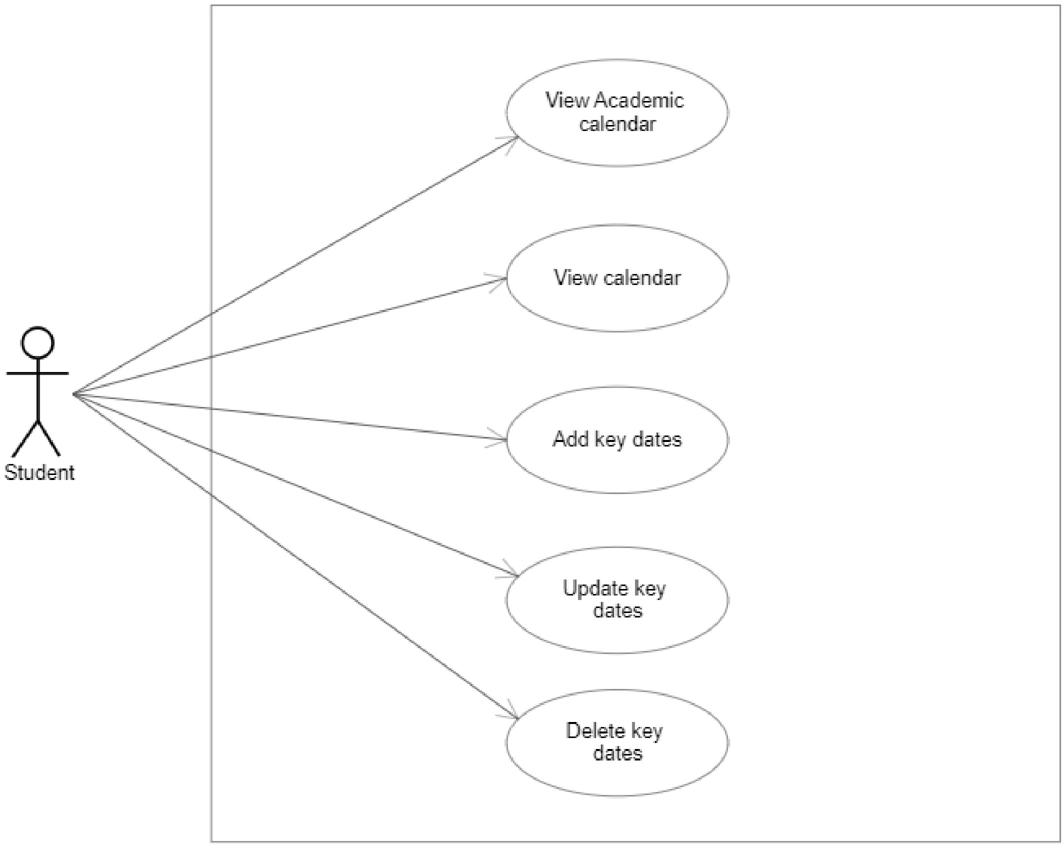
Accommodation



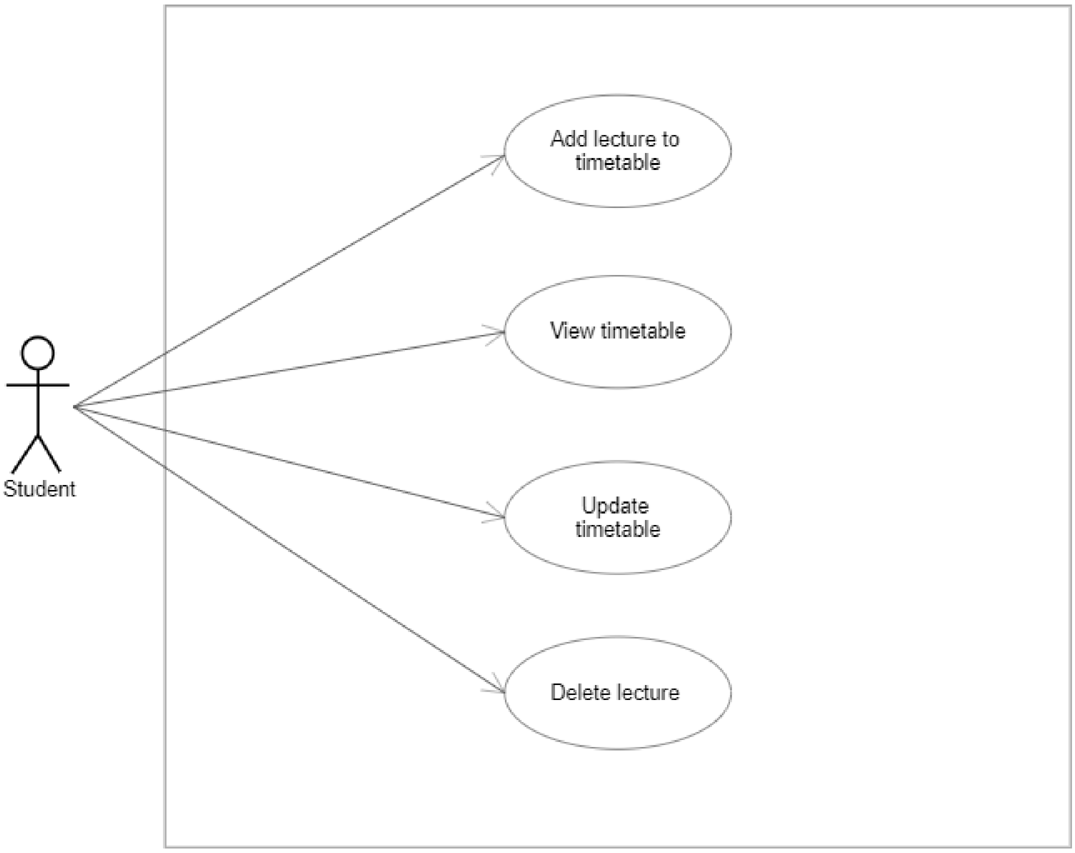
Transport



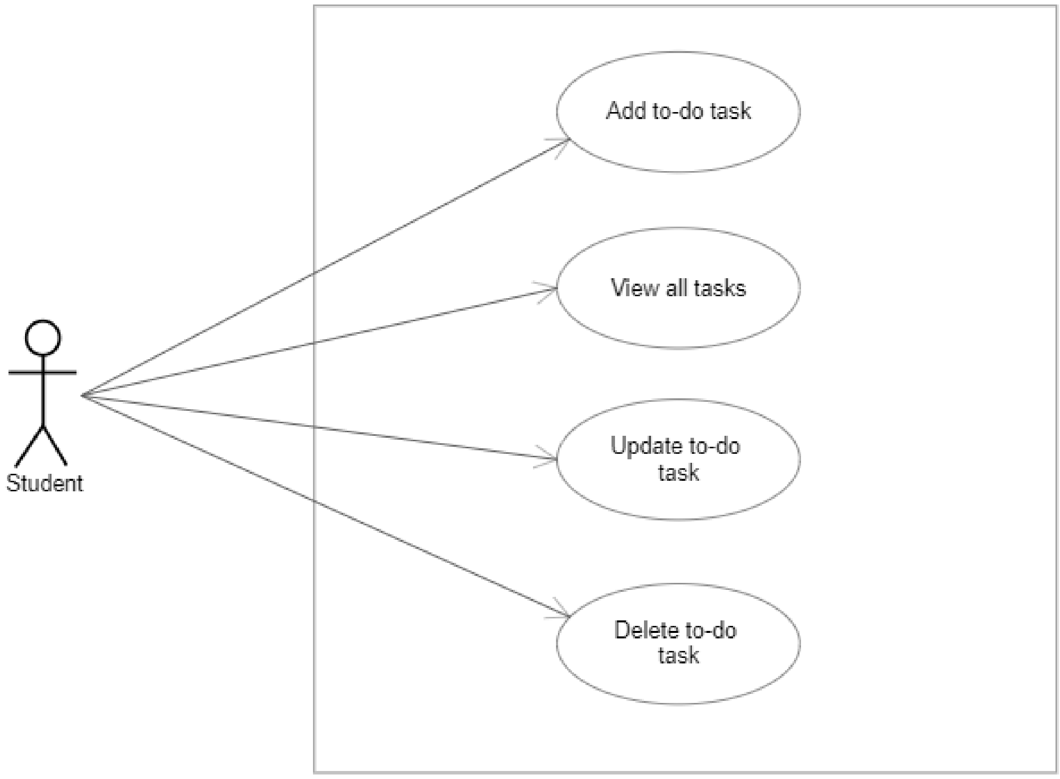
Calendar



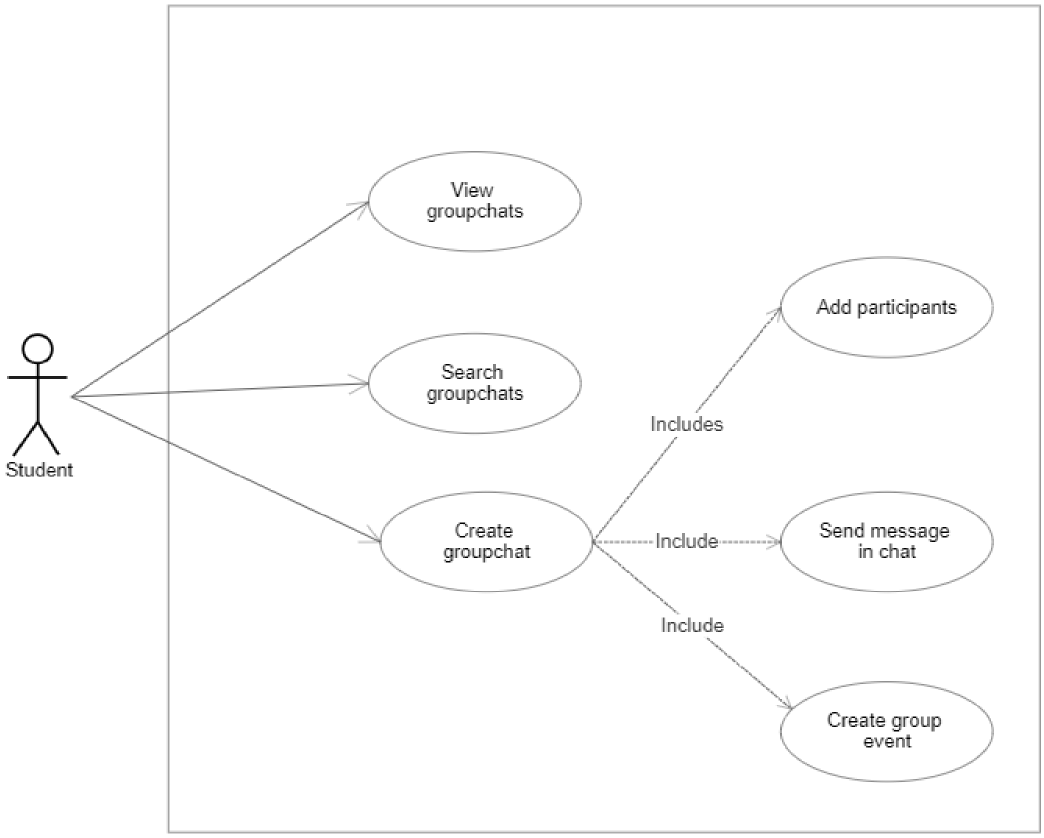
Timetable



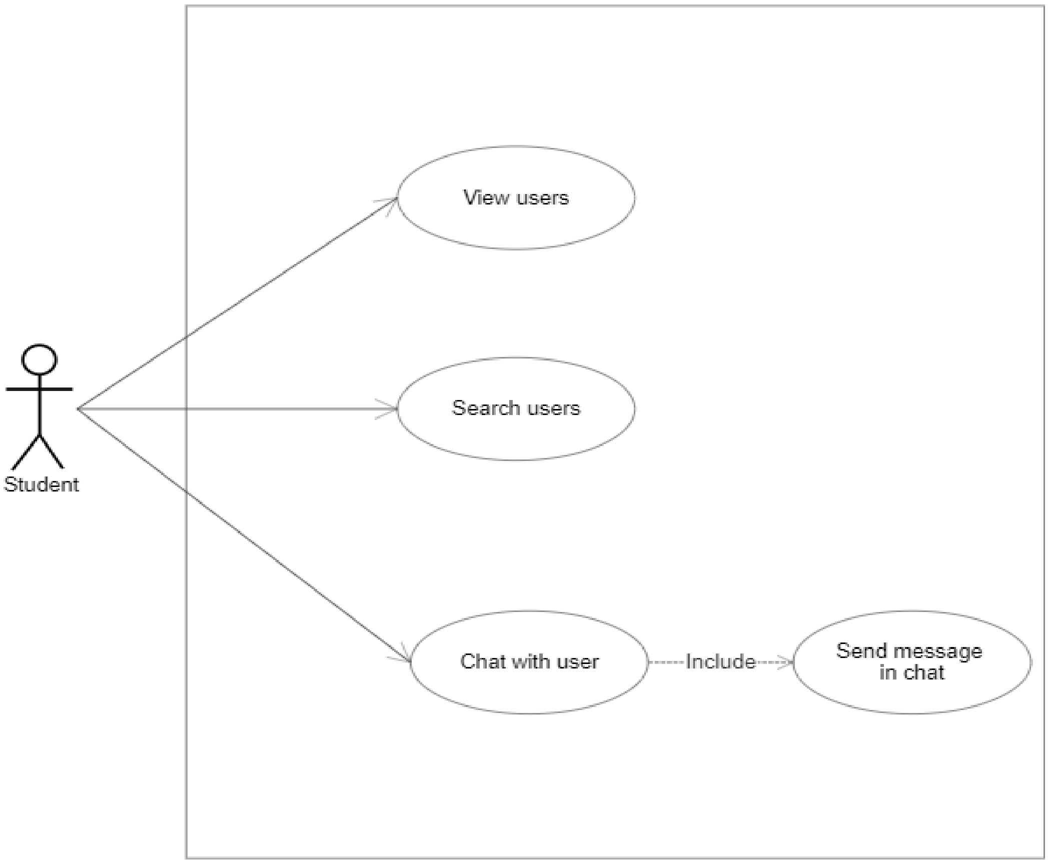
To-do List



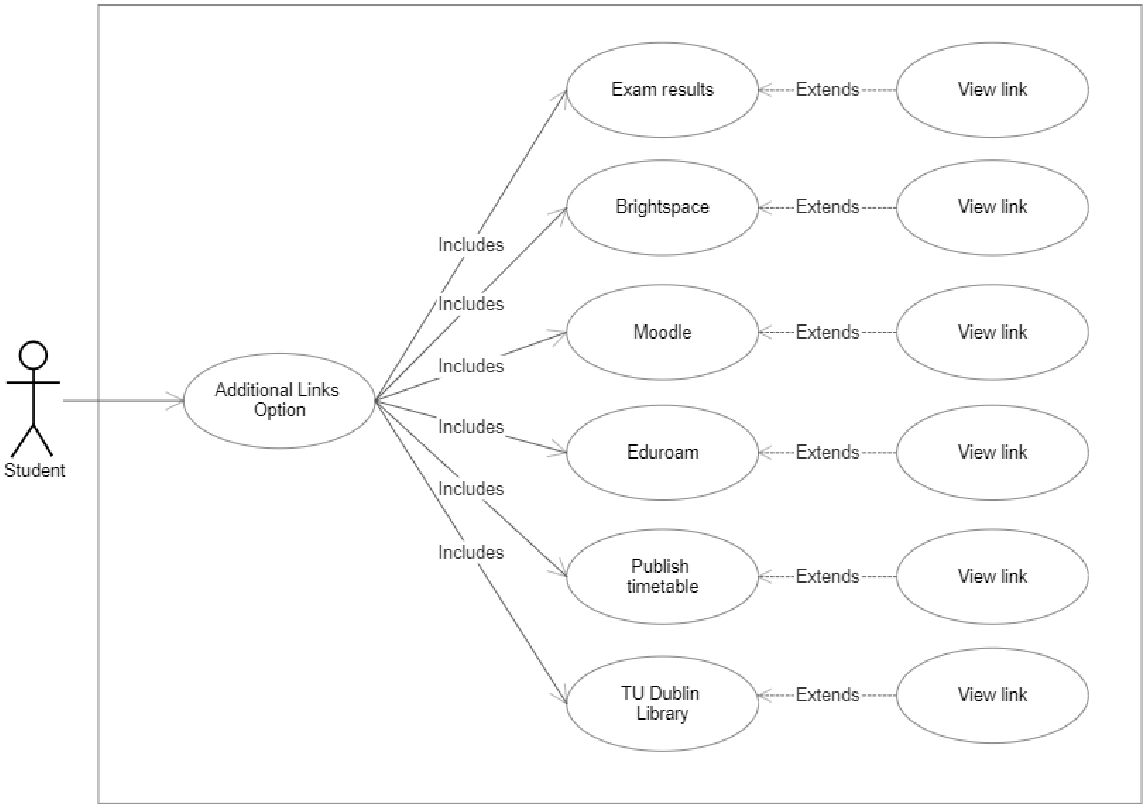
Groupchats



Chat with Friends

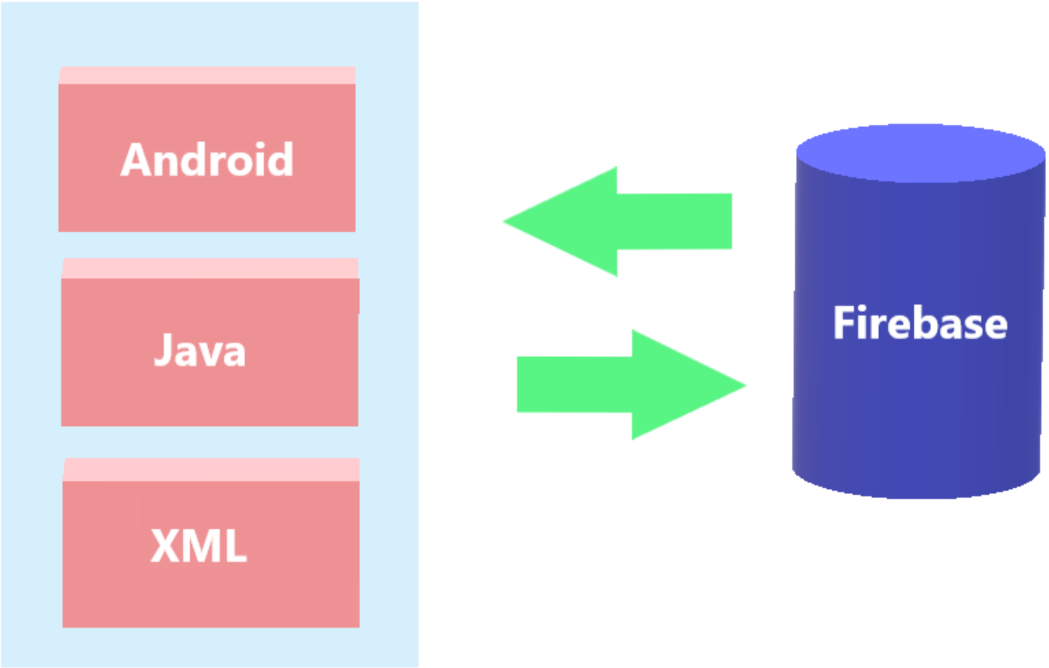


Additional Links



Chapter 3 - Design

System Architecture



First Year Companion is an Android Firebase application, the system architecture includes:

- **Presentation Layer:**

  This includes activities, views, and fragments. By using the Firebase SDK's these components interact with the Firebase backend. This layer is for handling the UI of the application. This layer is the layer that can be viewed by the student.

- **Application Layer:**

  This layer is needed to handle all of the business logic of the application. This layer interacts with the Firebase backend and performs user authentication and firebase data storage.

- **Data Layer:**

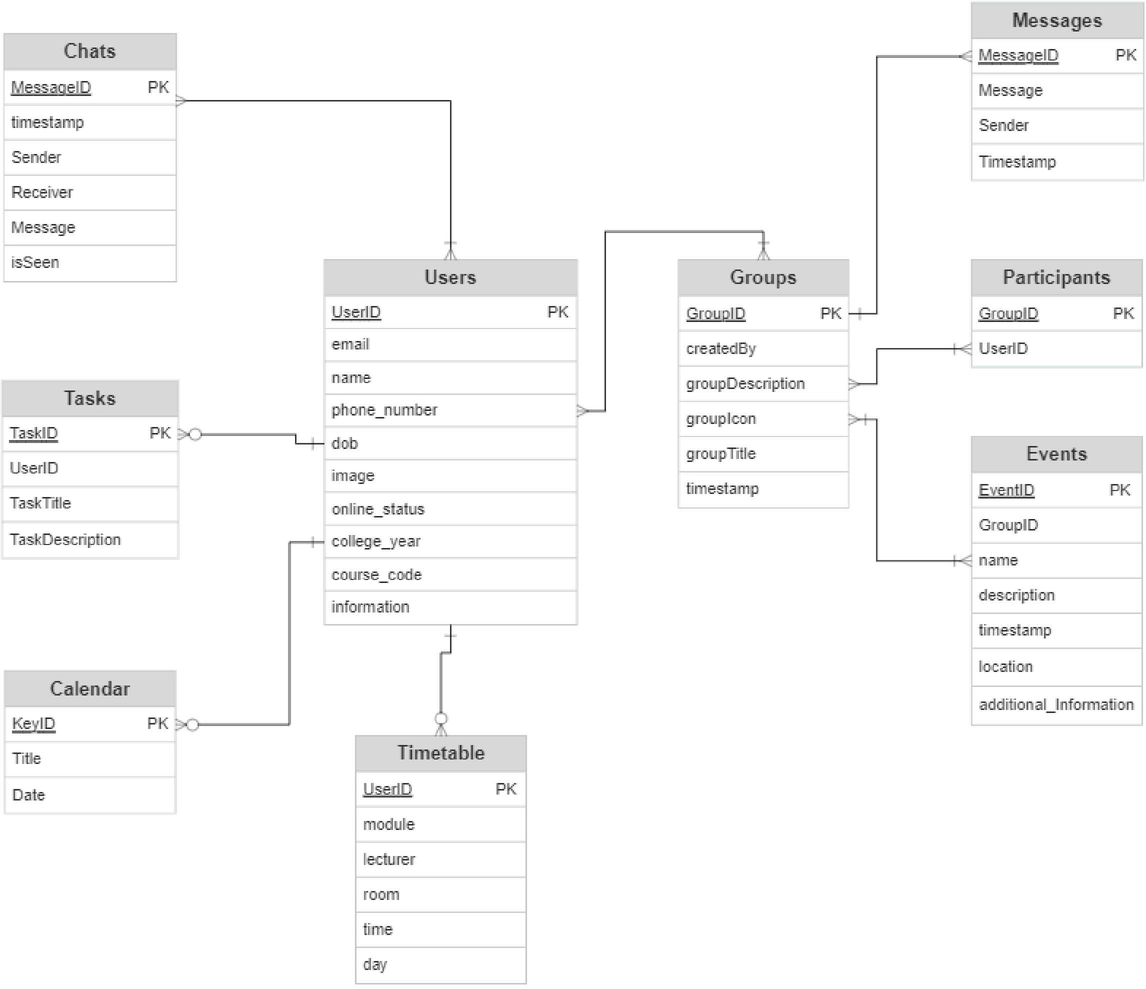
  This layer includes Firebase cloud storage and the Realtime database, this layer is needed to handle the data persistence of the application. It provides all of the necessary data to the Application layer above.

- **Firebase Backend:**

  The Firebase backend is a cloud-based infrastructure. This includes the Firebase Realtime database, Firebase Authentication and Firebase Cloud Storage for my application. The backend provides all of the relevant services and APIs for the Android application to communicate with it.

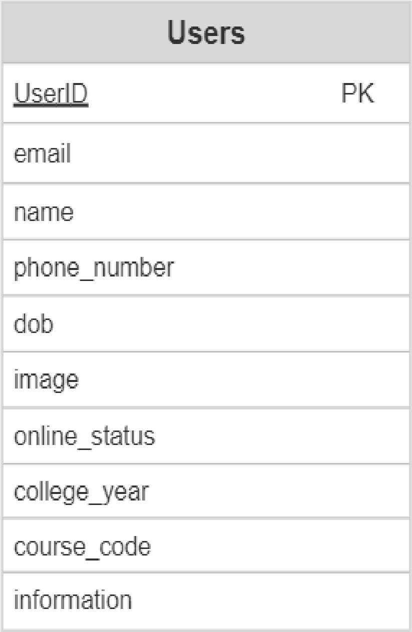
Data Model Design

I used Firebase as my database layer. Firebase is a mobile and web application development platform that provides users with a range of tools and services in order for developers to develop high quality applications easily and efficiently. For my project I used the Firebase Realtime database, Firebase Authentication and Firebase Cloud Storage, which are three key features provided by Firebase.



Database Object Summary

Users



Users	
UserID	PK
email	
name	
phone_number	
dob	
image	
online_status	
college_year	
course_code	
information	

The Users table contains all the student's details including the unique ID (Primary Key) which is the UserID, this is generated through the Firebase Authentication when a student registers for an account.

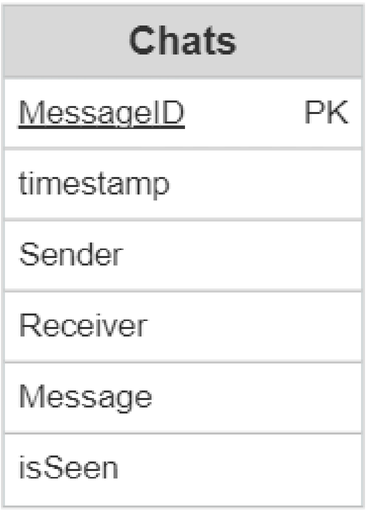
The student's email is entered when the user registers.

The user table also contains details such as:

- Student's full name
- Phone number
- Date of birth
- Online status
- College year
- Course code
- About me information

These are set as default when the user first registers, these can be customised in the student profile fragment.

Chats



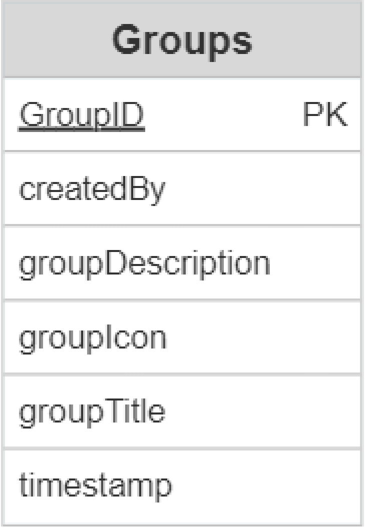
Chats	
MessageID	PK
timestamp	
Sender	
Receiver	
Message	
isSeen	

The Chats table holds all the details about the students' individual chats. The primary key is MessageID.

The chat details include:

- Sender- The ID of the sender who sent the message.
- Receiver – The ID of the receiver who received the message.
- Timestamp – The date and time the message was sent.
- Message – The context of the message that was sent.
- isSeen – is a boolean data type for if the other student has viewed the current student's message.

Groups



Groups	
GroupID	PK
createdBy	
groupDescription	
groupIcon	
groupTitle	
timestamp	

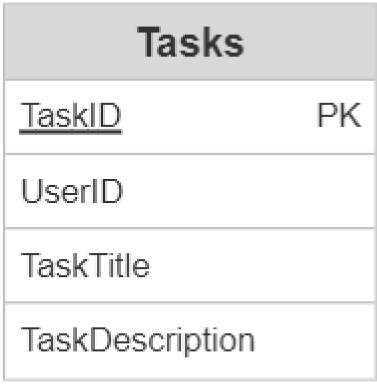
The Groups table contains all the group chat information.

The primary key is GroupID.

When a group is created by a user these following details are created and stored:

- GroupID
- createdBy – this indicates which user created the group chat.
- groupDescription
- groupIcon – the image URL for the group profile icon.
- groupTitle
- Timestamp

Tasks



Tasks	
TaskID	PK
UserID	
TaskTitle	
TaskDescription	

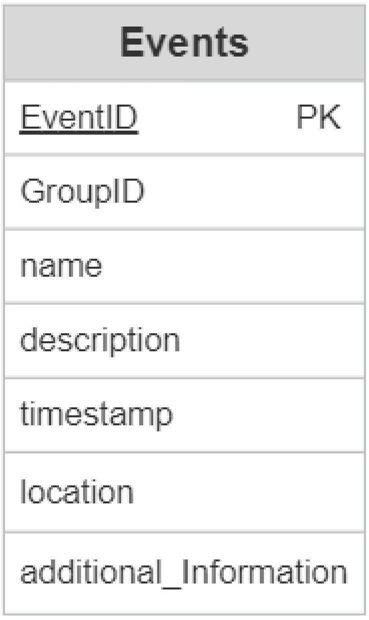
The tasks table contain all the to-do list task information.

The primary key is TaskID.

The following details are created and stored in the database when a user creates a to-do tasks:

- TaskID
- UserID
- TaskTitle
- TaskDescription

Events



Events	
EventID	PK
GroupID	
name	
description	
timestamp	
location	
additional_Information	

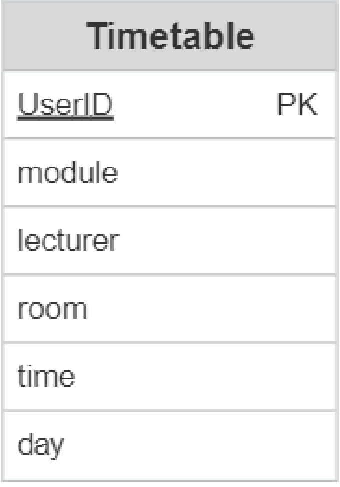
The events table holds all the details about the meetup event created within a group chat.

The primary key is EventID

The Events details include:

- EventID
- GroupID – the group in which the event was created.
- Name – The name of the event.
- Description – description of the event.
- Timestamp – when the event is taking place.
- Location – where the event is taking place.
- Additional information

Timetable



Timetable	
UserID	PK
module	
lecturer	
room	
time	
day	

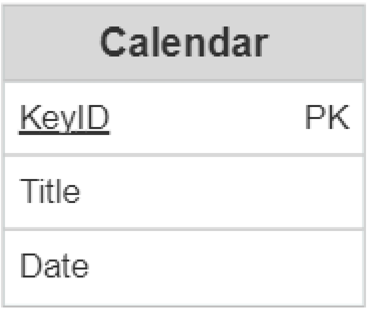
The timetable table holds all details about the student's timetable.

The primary key is UserID

The timetable table details include:

- UserID
- Module
- Lecturer
- Room
- Time
- day

Calendar



Calendar	
KeyID	PK
Title	
Date	

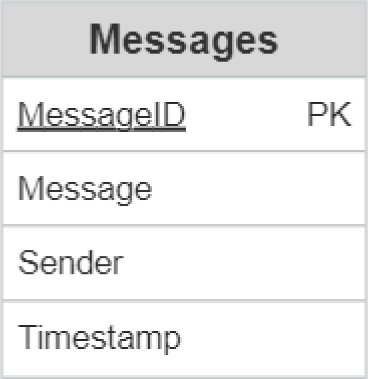
The calendar table holds all the calendar feature details.

The primary key is KeyID.

The table details include:

- KeyID
- Title
- Date

Messages



Messages	
MessageID	PK
Message	
Sender	
Timestamp	

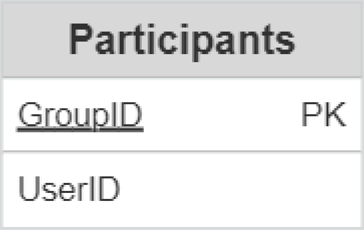
The messages table holds all the group chat message details.

The primary key is MessageID.

The tables details include:

- MessageID
- Message – the message inputted by the user in the group chat
- Sender – The sender who sent the message into the group chat.
- Timestamp

Participants



Participants	
GroupID	PK
UserID	

The participants table holds all the participant details that have been added to the group chat.

The primary key is GroupID

The table details also include the UserID of the user that has been added to the selected group.

Software Design

The software design of the First Year Companion application follows the pattern of Model View Controller (MVC) architecture, with Firebase SDK's as the application's backend.

The high-level overview of the applications software design:

- **Model classes:** I used these classes to define data objects in the application such as User, Chat, GroupChat, GroupChatList, ToDo, Event etc. They define the behaviour and structure of the data objects. These model classes represent the data that is retrieved from Firebase.

- **Adapter classes:** The adapter classes were used to bind the data from the model classes into the views of the application. They were used to create views for model item lists and populate these with data from the corresponding model class.

- **Fragments:** My application contains four fragments which are home fragment, chat fragment, group chat fragment and profile fragment. Fragments are reusable components that are used in an application to represent a segment of the user interface.

- **Activities:** My application contains many activities; these activities represent the different screens in the android application. These are used to handle the student's inputs, manage the UI, and communicate with all the other components within the application.

By using the MVC architecture this helped me manage and maintain the code easily as the system design provides clear separation of the model, view, and controller components.

User Interface Design

When designing the user interface, I prioritised certain requirements such as consistency and ease of use to enhance the student's user experience, while also integrating Firebase features.

User Interface Design:

- **Application Layout:** I wanted to ensure that the application was organised and clear for the user. I focused on consistency across all of the screens, ensuring that the same font and colours are used throughout.

- **Navigation:** I tried my best to ensure a consistent and clear navigation system for accessing all of the features of the application. I wanted to make it as easy to use as possible. The application included a bottom navigation bar for accessing the social features, home dashboard fragment and profile fragment. I also have menu options that are different on each screen depending on the screen its displayed on, for example the GroupchatActivity contains add participants menu option, group information menu option, and create meet up event menu option. The home dashboard contains eight card tiles within card views that represent all of the applications features that aid in onboarding.

- **Use of Libraries:** I used various libraries within the application to enhance the student's user experience. The circle image view library is for the student's profile image and group image, the Picasso and retrofit libraries are also used to aid with this. ImageSlideshow library is used within the transport option feature to give the student an insight into the various campuses within TU Dublin.

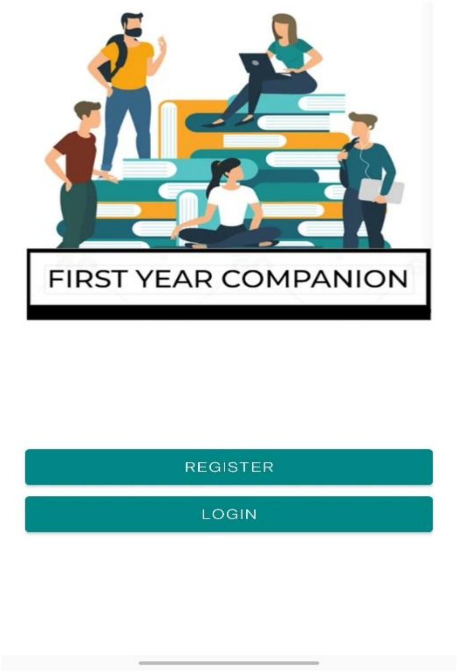
- **Android Input Controls:** I used multiple buttons and text fields within the application, all these input controls are clear and labelled in an understandable way for the student.

Chapter 4 - Implementation

Overview of Implementation

Front end

- **Main Activity Screen**

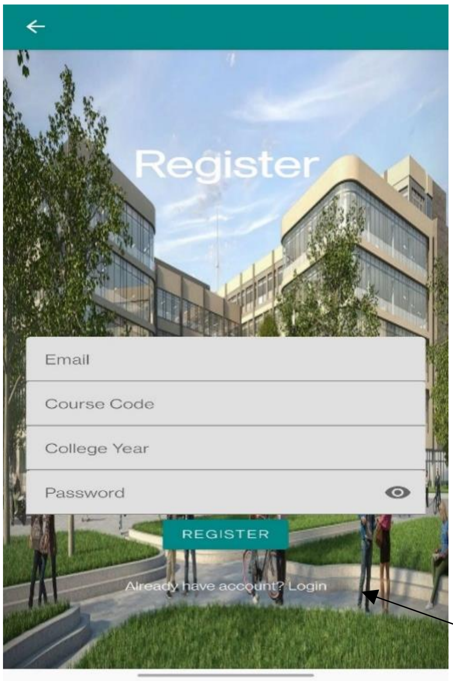


The user can select whether they want to register or login to their account by selecting the register or login button.

Component:

com.example.fypapplication1.MainActivity

- **User Registration Screen**



AlreadyHaveAnAccount redirects to Login screen

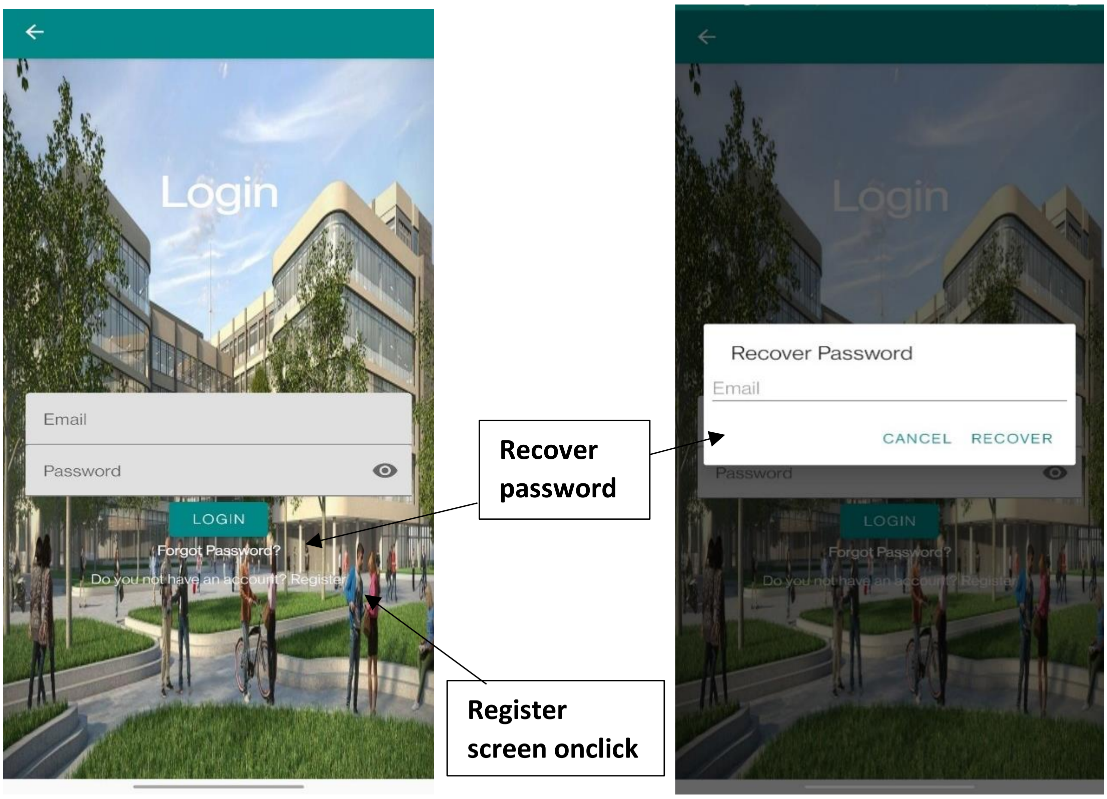
The student can register by entering their email, course code, college year and password.

Upon clicking register, the student will be automatically logged in and presented with the Home fragment that contains the home dashboard.

Component:

com.example.fypapplication1.RegisterActivity

- **Login Activity Screen**

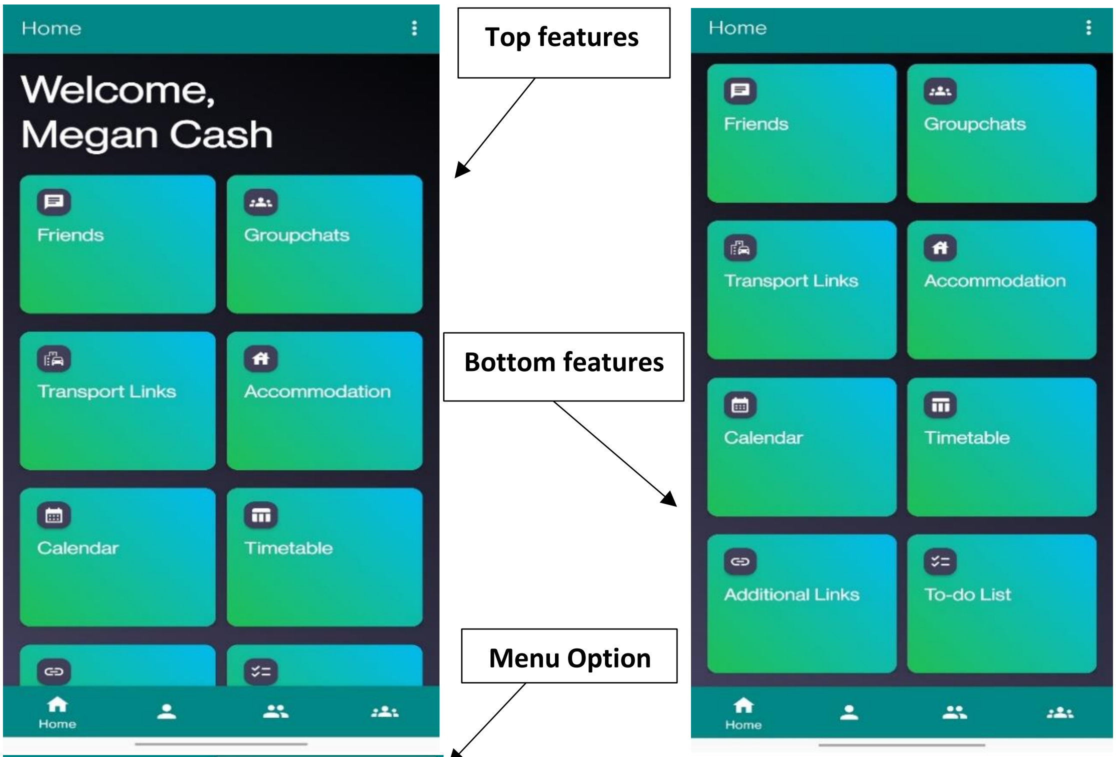


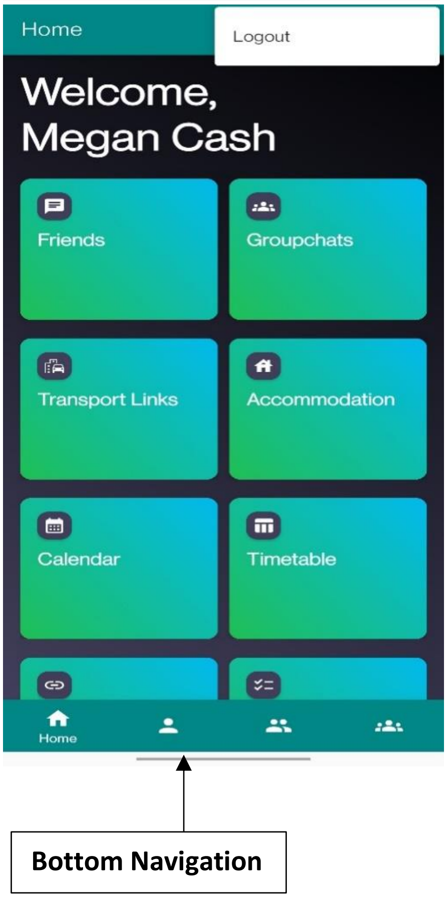
Once the student has registered, they can log in to their account with their email and password once they have been authenticated by Firebase Authentication. If a student has forgotten their password they can recover their password by entering their email to which an email will be sent to their inputted email address where they can reset their password.

Component:

com.example.fypapplication1.LoginActivity

- **Home Fragment**





The Home Fragment screen is displayed after a student has registered for the first time, or logged in. From the Home fragment the user can access the bottom navigation bar at the bottom of the screen to access fragments such as profile, friends, and group chats. The toolbar contains the option for the user to log out of their account. The home dashboard contains card items which include social functionalities such as Friends and Groupchats. Other features include Transport links, Accommodation, Calendar, Timetable, To-do List and Additional Links.

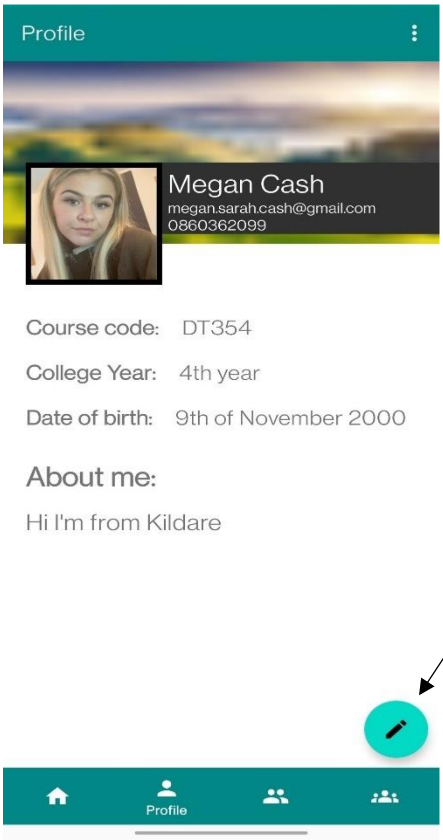
Component:

com.example.fypapplication1.Fragments.HomeFragment

Toolbar menu: home_menu.xml

Bottom navigation: navigation_menu.xml

- **Profile Fragment**



Megan Cash
megan.sarah.cash@gmail.com
0860362099

Course code: DT354

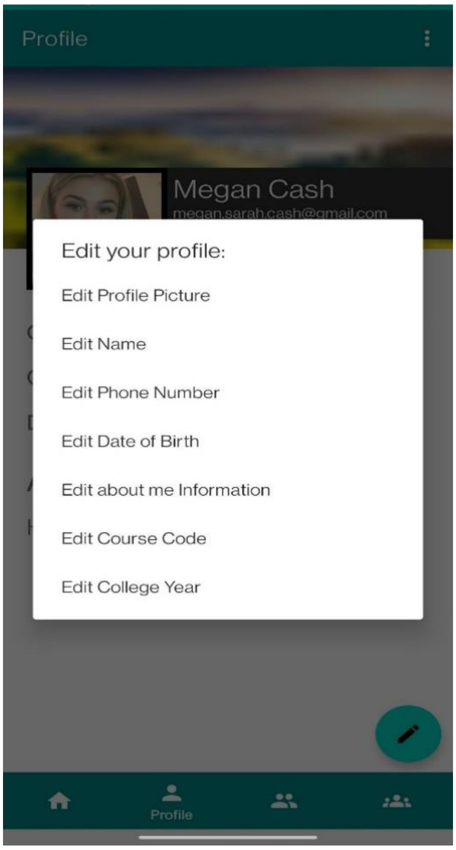
College Year: 4th year

Date of birth: 9th of November 2000

About me:

Hi I'm from Kildare

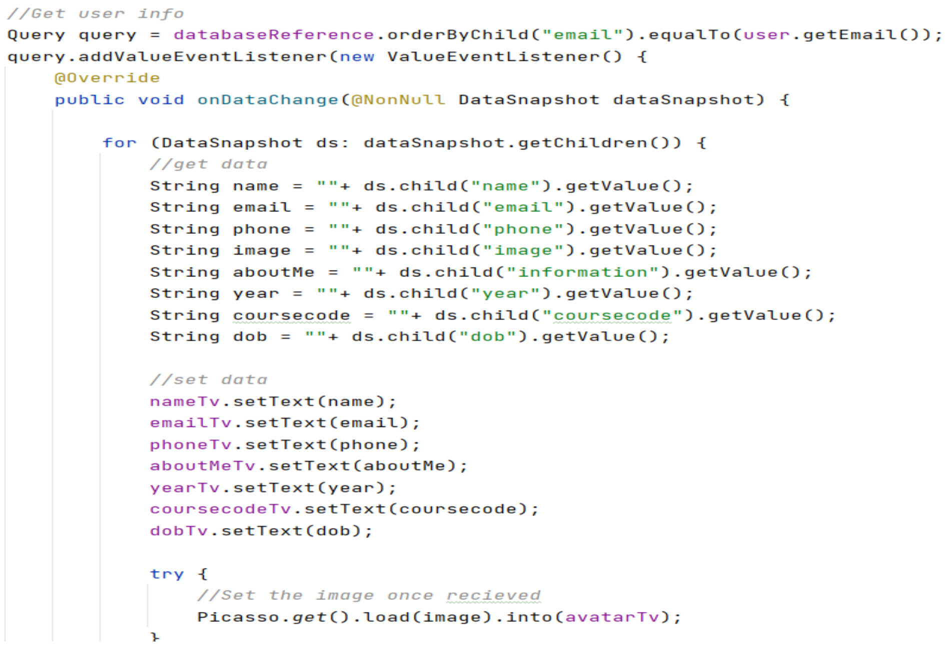
Floating action button to edit student Profile



The profile fragment screen is displayed when the user selected the second bottom navigation menu option. The floating action button on the bottom right-hand side is where the user can edit their profile details.

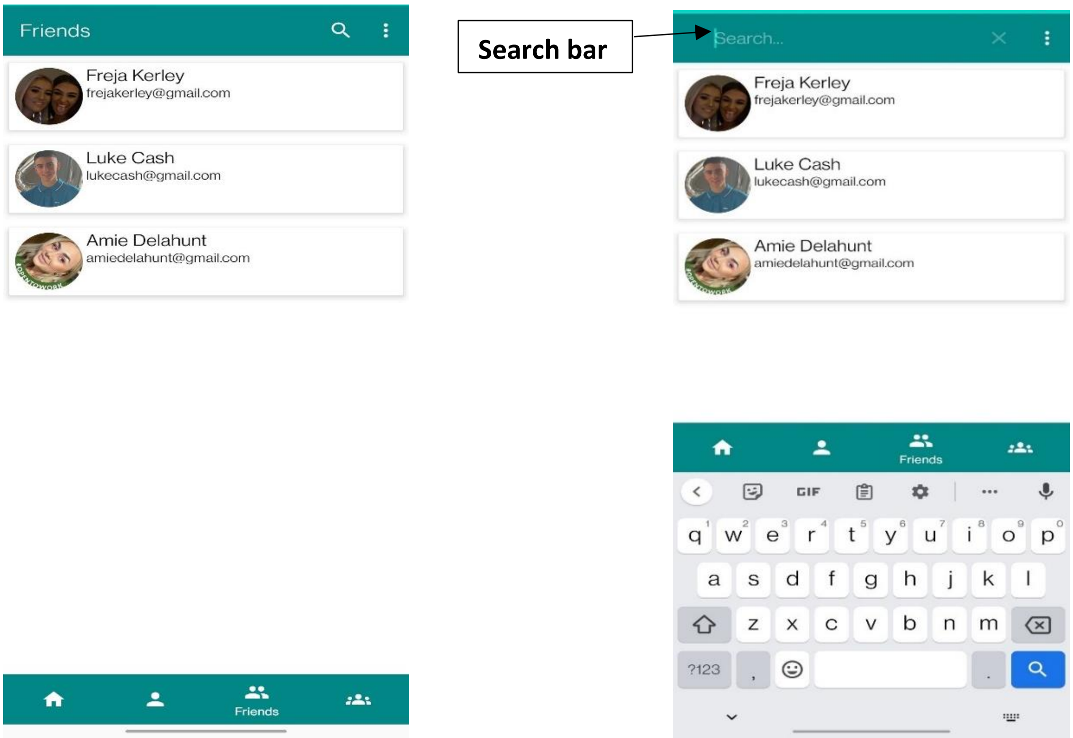
Component:

com.example.fypapplication1.Fragments.ProfileFragment



```java
//Get user info
Query query = databaseReference.orderByChild("email").equalTo(user.getEmail());
query.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

        for (DataSnapshot ds: dataSnapshot.getChildren()) {
            //get data
            String name = ""+ ds.child("name").getValue();
            String email = ""+ ds.child("email").getValue();
            String phone = ""+ ds.child("phone").getValue();
            String image = ""+ ds.child("image").getValue();
            String aboutMe = ""+ ds.child("information").getValue();
            String year = ""+ ds.child("year").getValue();
            String coursecode = ""+ ds.child("coursecode").getValue();
            String dob = ""+ ds.child("dob").getValue();

            //set data
            nameTv.setText(name);
            emailTv.setText(email);
            phoneTv.setText(phone);
            aboutMeTv.setText(aboutMe);
            yearTv.setText(year);
            coursecodeTv.setText(coursecode);
            dobTv.setText(dob);

            try {
                //Set the image once recieved
                Picasso.get().load(image).into(avatarTv);
            }
```

- **Friends Fragment**



Friends

Freja Kerley
frejakerley@gmail.com

Luke Cash
lukecash@gmail.com

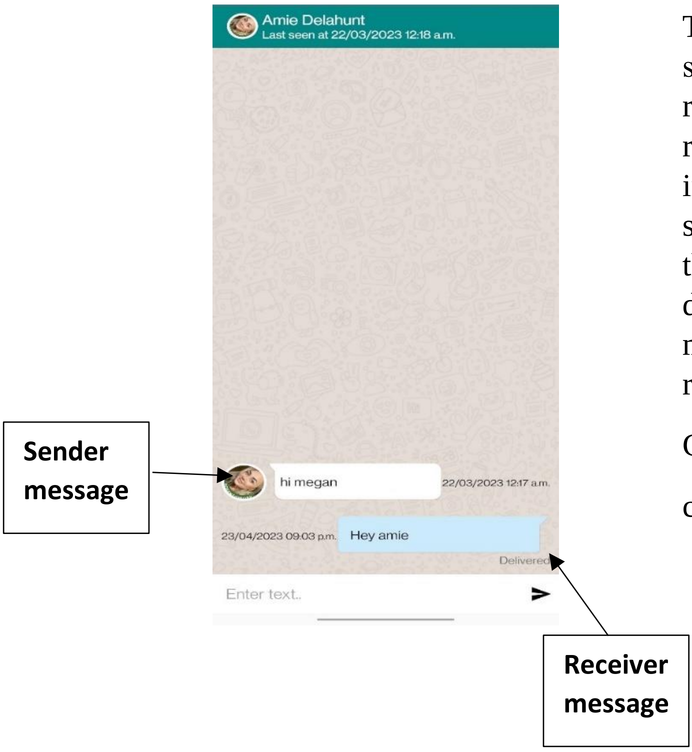
Amie Delahunt
amiedelahunt@gmail.com

The friends fragment displays all of the users on the system. The student is able to select any of the users to begin individually chatting with them. They can also use the search bar to find the specific user that they want to chat to.

Component:

com.example.fypapplication1.Fragments.FriendsFragment

- **Chat Activity**



Amie Delahunt
Last seen at 22/03/2023 12:18 a.m.

hi megan 22/03/2023 12:17 a.m.

23/04/2023 09:03 p.m. Hey amie
Delivered

The chat activity screen is displayed when the current user selects on one of the users in the friend's fragment recyclerview. This screen displays all messages sent and received. It also displays the other users name, profile image and when they were last online in the toolbar of the screen. The messages contain a timestamp that is within the Chats table in the realtime database, this timestamp displays the time in which the message was sent. The message layout files include row_chat_receiver.xml and row_chat_sender.xml

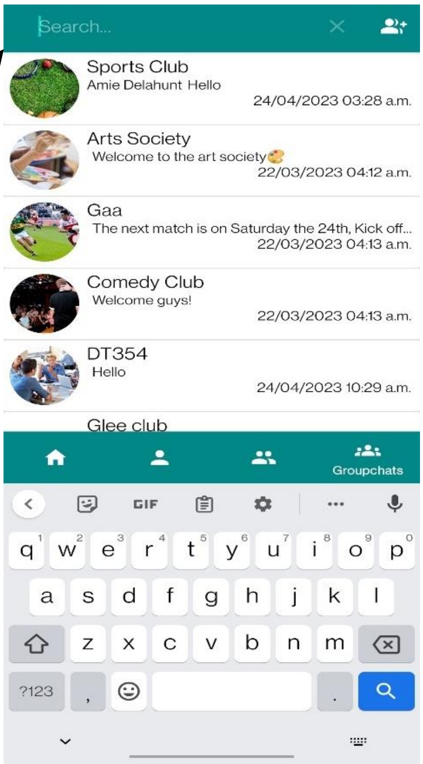
Component:

com.example.fypapplication1.ChatActivity

- **Groupchats Fragment**

Create group menu option

Search bar



Search...

Sports Club
Amie Delahunt Hello
24/04/2023 03:28 a.m.

Arts Society
Welcome to the art society🎨
22/03/2023 04:12 a.m.

Gaa
The next match is on Saturday the 24th, Kick off...
22/03/2023 04:13 a.m.

Comedy Club
Welcome guys!
22/03/2023 04:13 a.m.

DT354
Hello
24/04/2023 10:29 a.m.

Glee club

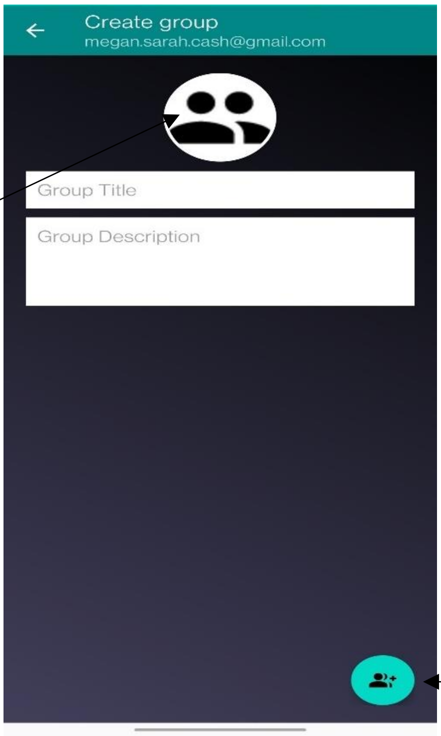
The groupchat fragment contains all of the groupchats that the current user is a participant in. A recyclerview of the groupchats list is displayed to the current user, the student is able to search through these if needed.

Component:

com.example.fypapplication1.Fragments.GroupchatsFragment

- **Create Group Chat Activity**

Group profile image upload on click



Create group
megan.sarah.cash@gmail.com

Group Title

Group Description

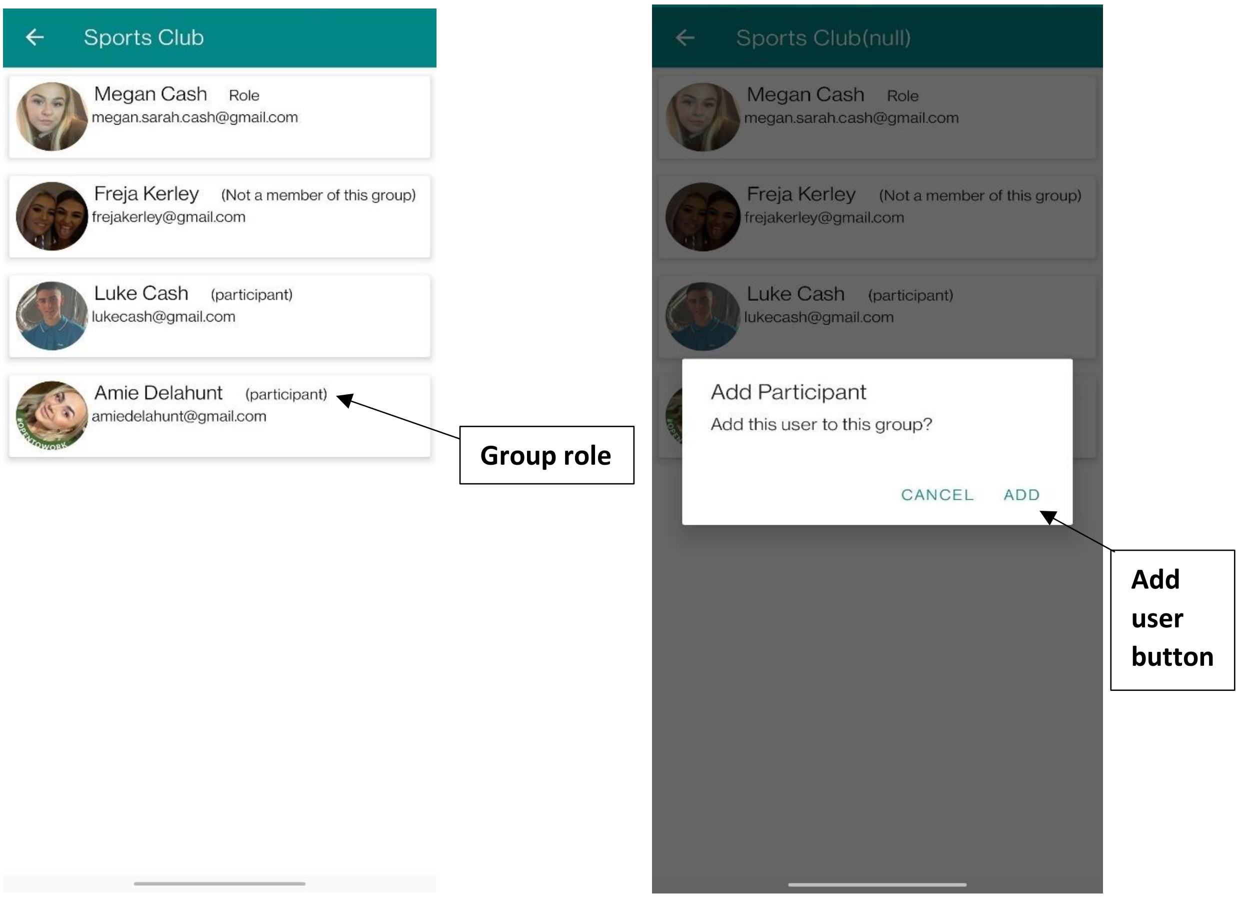
This is the creategroupchatActivity screen. This allows the student to create a group, the student must add a group title, group profile image and group description. Once they click the floating action button, the group with be added to Groups in the realtime database. When the group is created they are then presented with a menu option in the toolbar to then add participants to the group.

Component:

com.example.fypapplication1.CreateGroupchatActivity

Create group floating action button

- **Add group participants Activity Screen**



Sports Club

Megan Cash Role
megan.sarah.cash@gmail.com

Freja Kerley (Not a member of this group)
frejakerley@gmail.com

Luke Cash (participant)
lukecash@gmail.com

Amie Delahunt (participant)
amiedelahunt@gmail.com

Sports Club(null)

Add Participant
Add this user to this group?

This is the addParticipantsActivity screen where if the textView beside the student's name says '(participant)' this indicates that they are already a participant of this group. If the role says '(not a member of this group)', the admin of the group can select on this student, and they will be prompted with the pop up dialog box asking them whether they would like to add the user as a participant to the group.

Component:

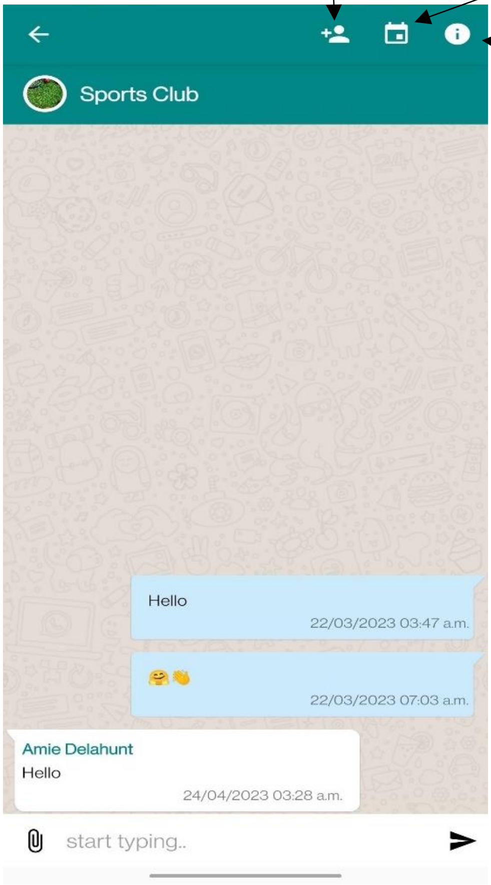
com.example.fypapplication1.AddParticipantsActivity

- **Group chat Activity**

Add participants

Create event

Group information


Sports Club

Hello
22/03/2023 03:47 a.m.

🤗👏
22/03/2023 07:03 a.m.

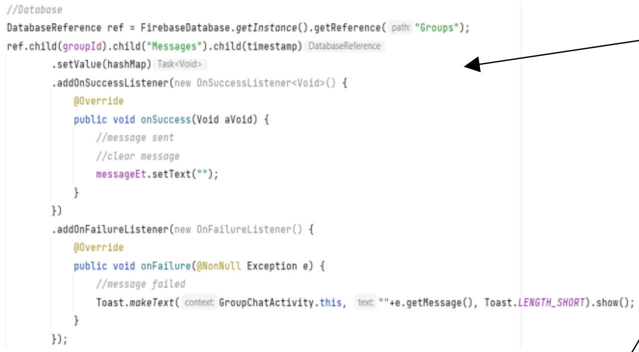
Amie Delahunt
Hello
24/04/2023 03:28 a.m.

The groupchat activity screen is displayed when the current user selects on one of the groupchats that they are a participant of within the groupchat fragment recyclerview. This screen displays all messages sent and received in the groupchat. It also displays the other students' names. The messages contain a timestamp that is within the Groups table in the realtime database, this timestamp displays the time in which the message was sent. Within the action bar the name and image of the group is displayed. Menu options such as create event, add participants, group information options are at the top right-hand side of the screen in the toolbar.

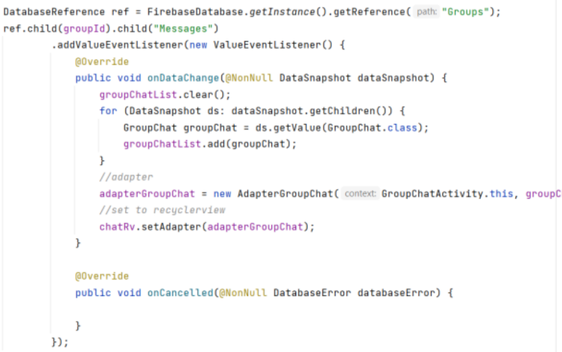
Component:

com.example.fypapplication1.GroupChatActivity

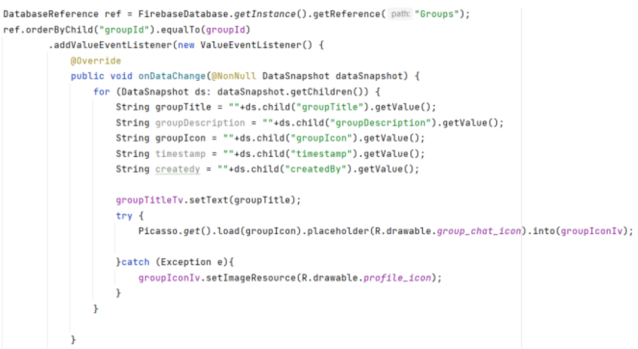
To send a message in the groupchat


```
//Database
DatabaseReference ref = FirebaseDatabase.getInstance().getReference( path: "Groups");
ref.child(groupId).child("Messages").child(timestamp) DatabaseReference
        .setValue(hashMap) Task<Void>
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                //message sent
                //clear message
                messageEt.setText("");
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                //message failed
                Toast.makeText( context: GroupChatActivity.this,  text: ""+e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
```

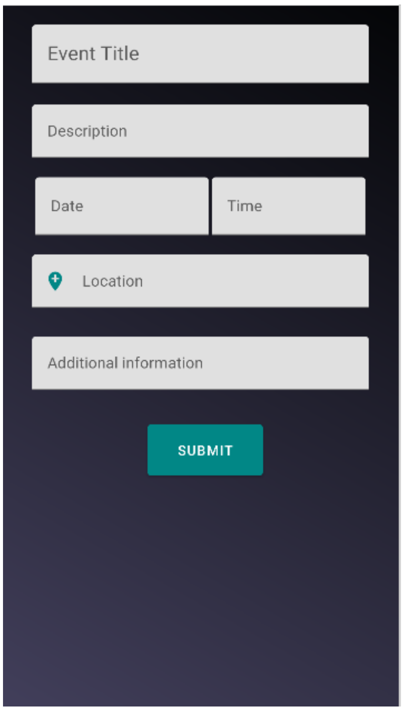
To retrieve all the group messages


```
DatabaseReference ref = FirebaseDatabase.getInstance().getReference( path: "Groups");
ref.child(groupId).child("Messages")
        .addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                groupChatList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    GroupChat groupChat = ds.getValue(GroupChat.class);
                    groupChatList.add(groupChat);
                }
                //adapter
                adapterGroupChat = new AdapterGroupChat( context: GroupChatActivity.this, groupChatList);
                //set to recyclerview
                chatRv.setAdapter(adapterGroupChat);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
```

To retrieve group information


```
DatabaseReference ref = FirebaseDatabase.getInstance().getReference( path: "Groups");
ref.orderByChild("groupId").equalTo(groupId)
        .addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot ds: dataSnapshot.getChildren()) {
                    String groupTitle = ""+ds.child("groupTitle").getValue();
                    String groupDescription = ""+ds.child("groupDescription").getValue();
                    String groupIcon = ""+ds.child("groupIcon").getValue();
                    String timestamp = ""+ds.child("timestamp").getValue();
                    String createdy = ""+ds.child("createdBy").getValue();

                    groupTitleTv.setText(groupTitle);
                    try {
                        Picasso.get().load(groupIcon).placeholder(R.drawable.group_chat_icon).into(groupIconIv);

                    }catch (Exception e){
                        groupIconIv.setImageResource(R.drawable.profile_icon);
                    }
                }

            }
```

- **Event Activity Screens**

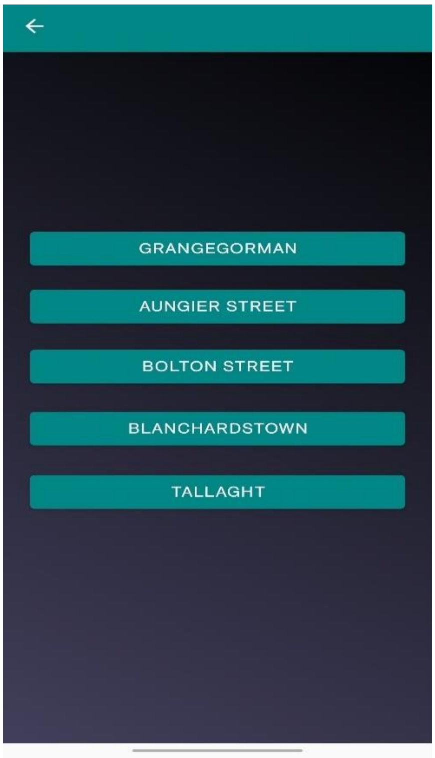


When the user selects the create event menu option within the groupchat activity, they are brought to the createEventActivity where they are able to input vital details about the meet up event such as the event title, event description, the date and time the event is taking place, the location where it will be held and then there is also an extra input for any additional information that the student needs to provide. Once the submit button is selected by the user, the event is created and sent out to the participants of the group to which they can accept or deny if they are attending. The student can then be brought to the EventActivity screen.

Component:

com.example.fypapplication1.GroupEvent.CreateEventActivity

- **Transport Option Activity**

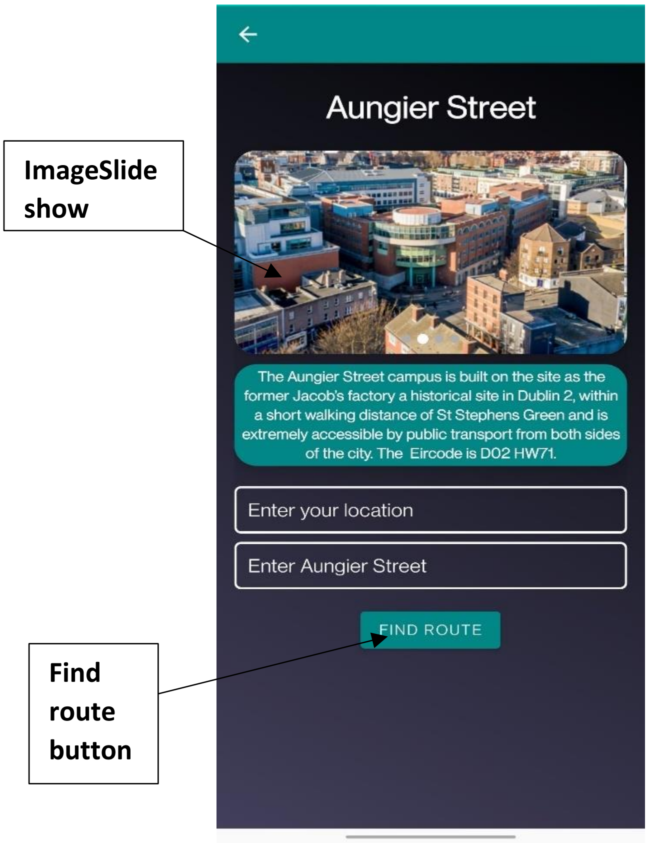


When the user selects the transport option on the home dashboard, they will see the Transport Option activity screen that contains 5 buttons that represent the different TU Dublin campuses. Each button contains a imageslide show of each campus, as well as details about the campus.

Component:

com.example.fypapplication1.CardOptions.TransportActivity

- **Aungier Street Activity**



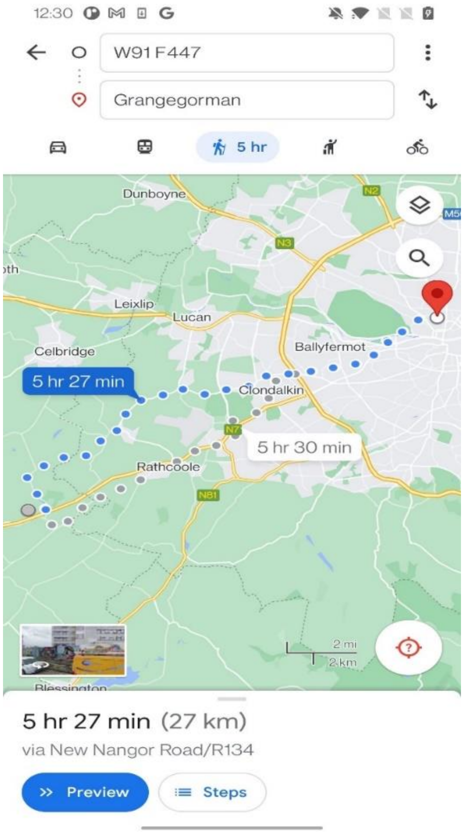
Aungier Street

The Aungier Street campus is built on the site as the former Jacob's factory a historical site in Dublin 2, within a short walking distance of St Stephens Green and is extremely accessible by public transport from both sides of the city. The Eircode is D02 HW71.

Enter your location

Enter Aungier Street

FIND ROUTE



5 hr 27 min (27 km)

via New Nangor Road/R134

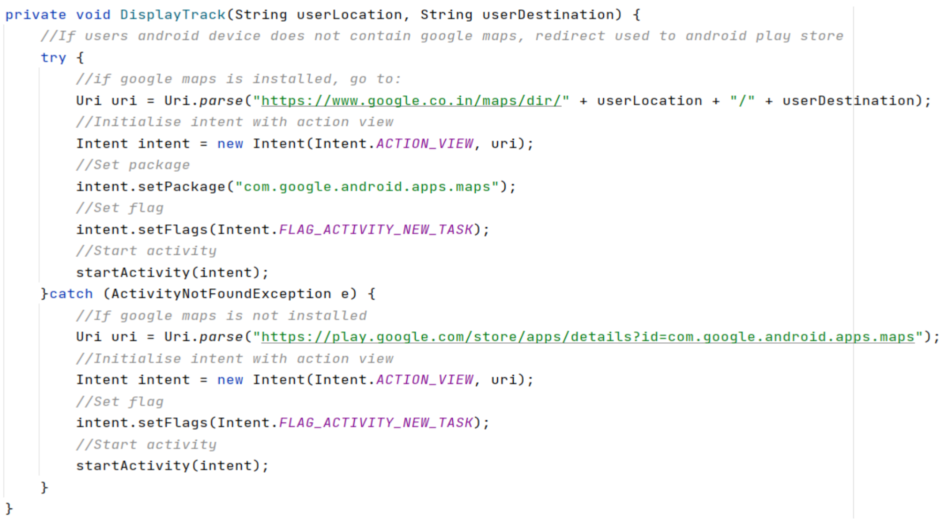
Preview

Steps

All of the button options from the Transport option activity follow the same concept. These screens contain two edit text fields, one which is a prompt for the student to enter their current location, the other is to enter the name of the campus. The student is then redirected to google maps where the quickest route is displayed. If the user does not have google maps, they are redirected to the play store to download google maps.

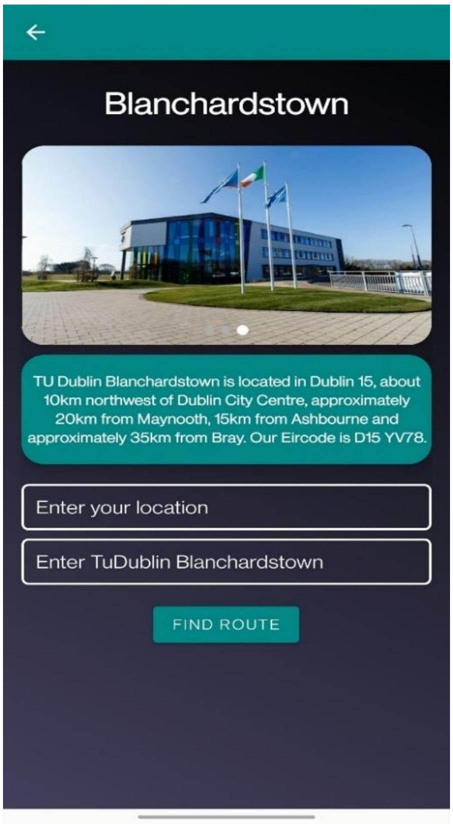
Component:

com.example.fypapplication1.CardOptions.TransportOptions.AungierStreetActivity



```
private void DisplayTrack(String userLocation, String userDestination) {
    //If users android device does not contain google maps, redirect used to android play store
    try {
        //if google maps is installed, go to:
        Uri uri = Uri.parse("https://www.google.co.in/maps/dir/" + userLocation + "/" + userDestination);
        //Initialise intent with action view
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        //Set package
        intent.setPackage("com.google.android.apps.maps");
        //Set flag
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        //Start activity
        startActivity(intent);
    }catch (ActivityNotFoundException e) {
        //If google maps is not installed
        Uri uri = Uri.parse("https://play.google.com/store/apps/details?id=com.google.android.apps.maps");
        //Initialise intent with action view
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        //Set flag
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        //Start activity
        startActivity(intent);
    }
}
```

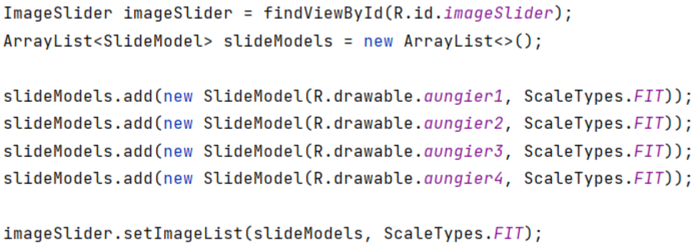
- **Blanchardstown Activity**



TU Dublin Blanchardstown is located in Dublin 15, about 10km northwest of Dublin City Centre, approximately 20km from Maynooth, 15km from Ashbourne and approximately 35km from Bray. Our Eircode is D15 YV78.

Component:

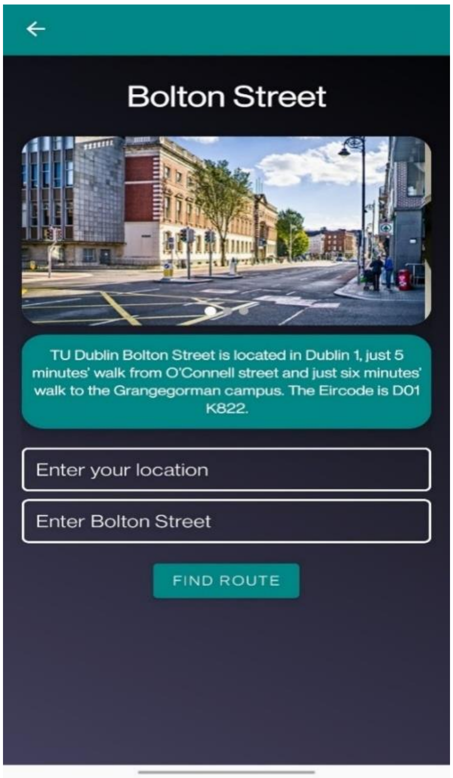
com.example.fypapplication1.CardOptions.TransportOptions.BlanchardstownActivity



```
ImageSlider imageSlider = findViewById(R.id.imageSlider);
ArrayList<SlideModel> slideModels = new ArrayList<>();

slideModels.add(new SlideModel(R.drawable.aungier1, ScaleTypes.FIT));
slideModels.add(new SlideModel(R.drawable.aungier2, ScaleTypes.FIT));
slideModels.add(new SlideModel(R.drawable.aungier3, ScaleTypes.FIT));
slideModels.add(new SlideModel(R.drawable.aungier4, ScaleTypes.FIT));

imageSlider.setImageList(slideModels, ScaleTypes.FIT);
```

Slideshow code

- **Bolton Street Activity**

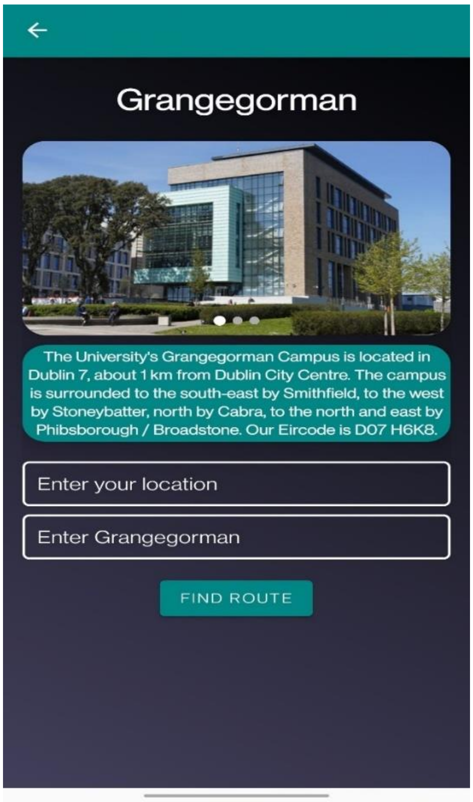


TU Dublin Bolton Street is located in Dublin 1, just 5 minutes' walk from O'Connell street and just six minutes' walk to the Grangegorman campus. The Eircode is D01 K822.

Component:

com.example.fypapplication1.CardOptions.TransportOptions.BoltonStreetActivity

- **Grangegorman Activity**



Grangegorman

The University's Grangegorman Campus is located in Dublin 7, about 1 km from Dublin City Centre. The campus is surrounded to the south-east by Smithfield, to the west by Stoneybatter, north by Cabra, to the north and east by Phibsborough / Broadstone. Our Eircode is D07 H6K8.

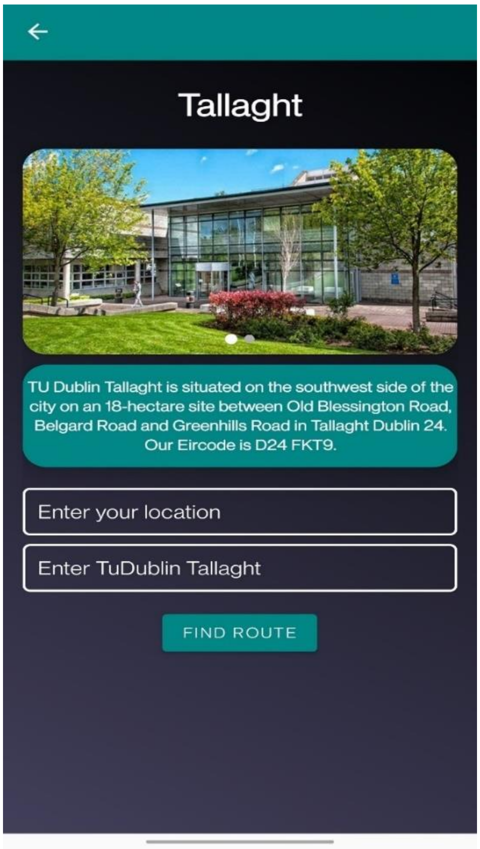
Enter your location

Enter Grangegorman

Component:

com.example.fypapplication1.CardOptions.TransportOptions.GrangegormanActivity

- **Tallaght Activity**



Tallaght

TU Dublin Tallaght is situated on the southwest side of the city on an 18-hectare site between Old Blessington Road, Belgard Road and Greenhills Road in Tallaght Dublin 24. Our Eircode is D24 FKT9.

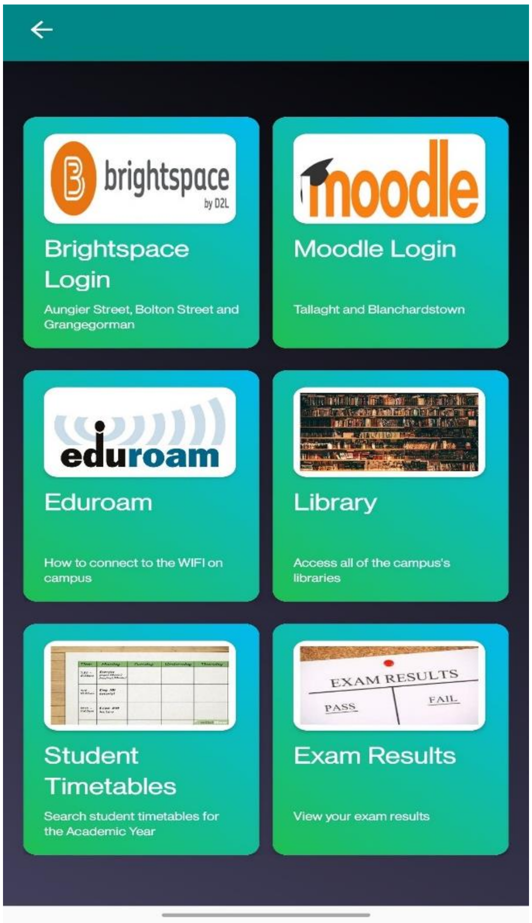
Enter your location

Enter TuDublin Tallaght

Component:

com.example.fypapplication1.CardOptions.TransportOptions.TallaghtActivity

- **Additional Links Activity**


Brightspace Login

Aungier Street, Bolton Street and Grangegorman

Moodle Login

Tallaght and Blanchardstown

Eduroam

How to connect to the WIFI on campus

Library

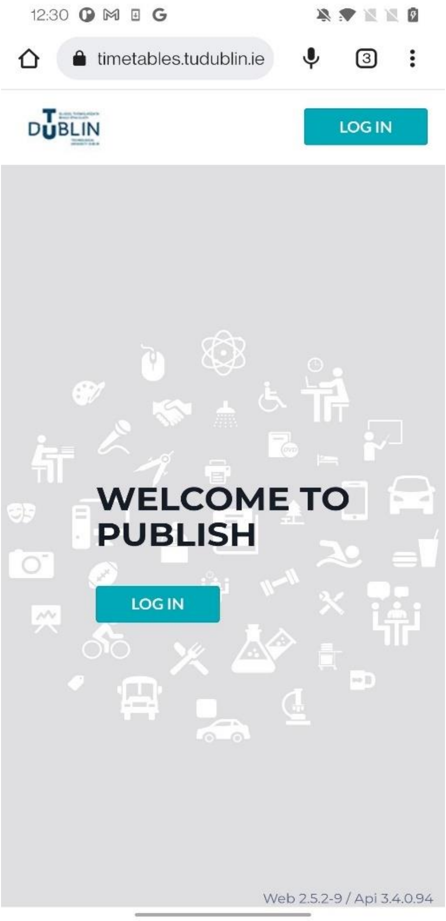
Access all of the campus's libraries

Student Timetables

Search student timetables for the Academic Year

Exam Results

View your exam results


12:30

timetables.tudublin.ie

LOG IN

WELCOME TO PUBLISH

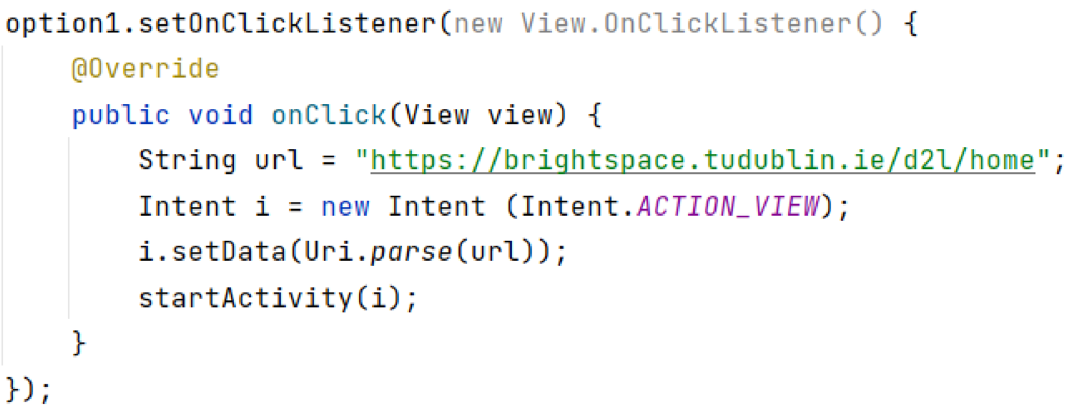
LOG IN

Web 2.5.2-9 / Api 3.4.0.94

When the student selects the additional links option on the home dashboard, they will see the Accommodation Option activity screen that contains 6 card view items within a card view layout, the card items represent different types of accommodation the student can access and be redirected to the links of. For example, if the fifth card item ‘Student Timetables’ is selected, the user is redirected the TU Dublin publish link.

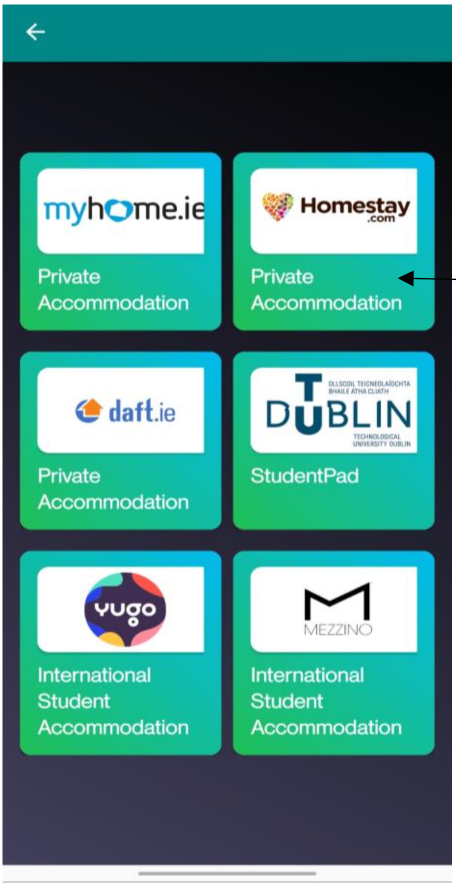
Component:

com.example.fypapplication1.CardOptions.LinksActivity

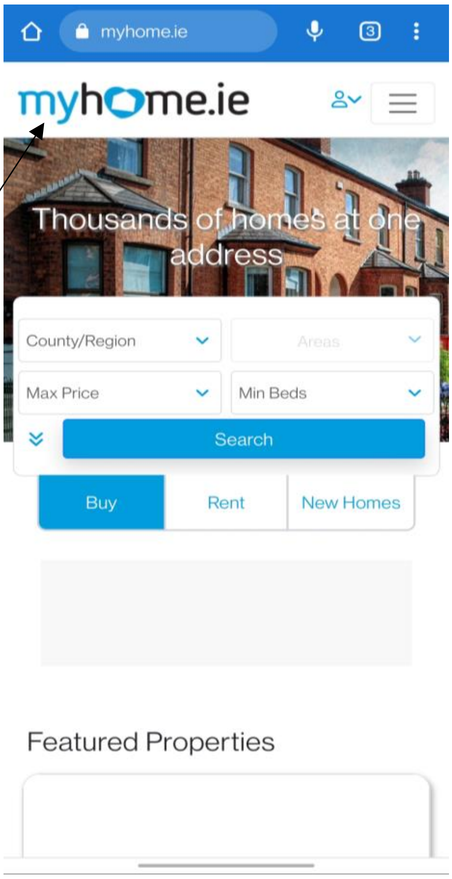

```
option1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String url = "https://brightspace.tudublin.ie/d2l/home";
        Intent i = new Intent (Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
});
```

- **Accommodation Activity**

Back press to the home dashboard



When any card view item is selected they are redirected to the corresponding link



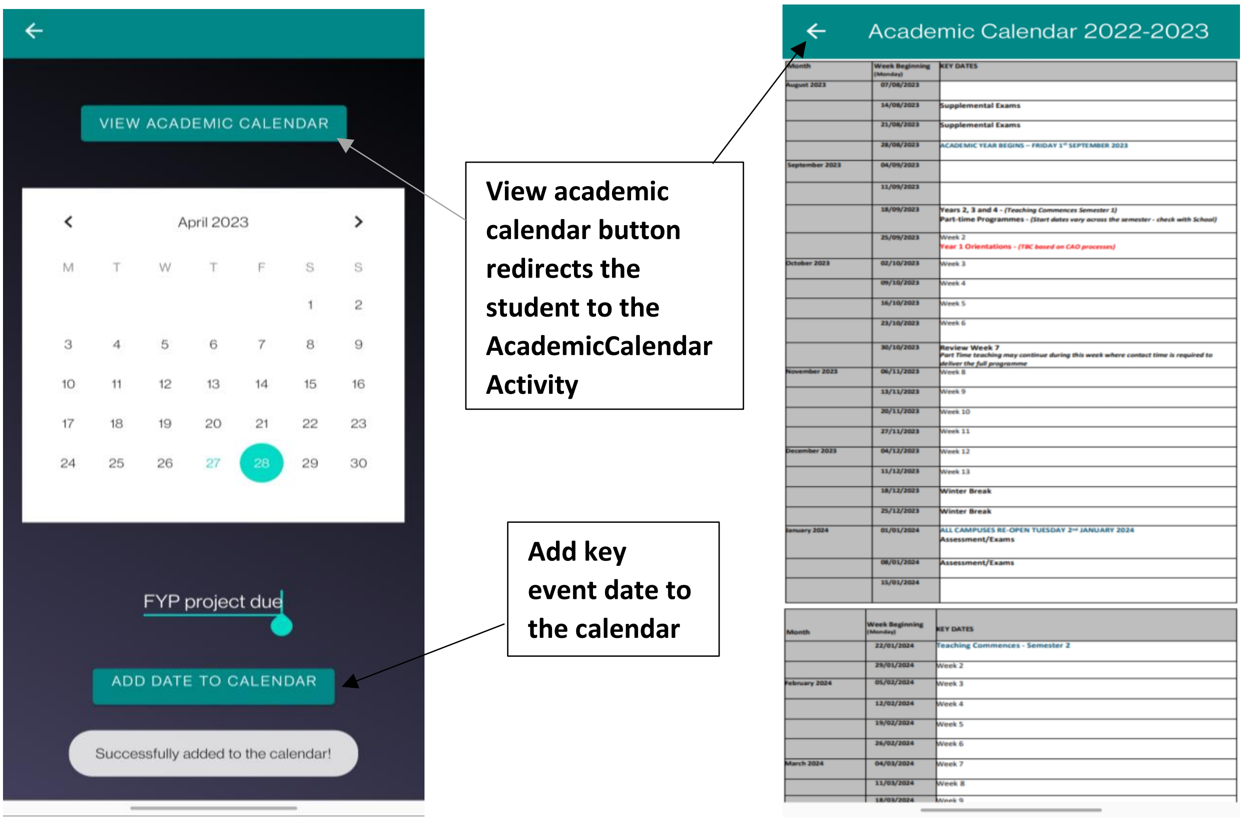
When the student selects the accommodation option on the home dashboard, they will see the Accommodation Option activity screen that contains 6 card view items within a card view layout, the card items represent different types of accommodation the student can access and be redirected to the links of. For example, if the first card item ‘private accommodation myHome.ie’ is selected, the user is redirected the myHome.ie link.

Component:

com.example.fypapplication1.CardOptions.AccomodationActivity

```
//MyHome.ie
accomodationOption1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String url = "https://www.myhome.ie/";
        Intent i = new Intent (Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
});
```

- **Calendar Activity**



M	T	W	T	F	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

FYP project due

Successfully added to the calendar!

View academic calendar button redirects the student to the AcademicCalendar Activity

Add key event date to the calendar

Academic Calendar 2022-2023

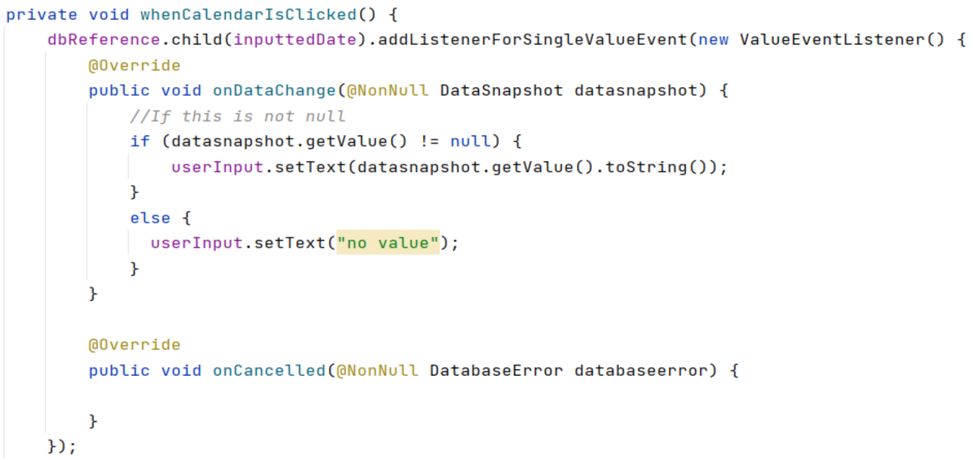
Month	Week Beginning (Monday)	KEY DATES
August 2023	07/08/2023	
	14/08/2023	Supplemental Exams
	21/08/2023	Supplemental Exams
	28/08/2023	ACADEMIC YEAR BEGINS – FRIDAY 1st SEPTEMBER 2023
September 2023	04/09/2023	
	11/09/2023	
	18/09/2023	Years 2, 3 and 4 - *(Teaching Commences Semester 1)* Part-time Programmes - *(Start dates vary across the semester - check with School)*
	25/09/2023	Week 2 Year 1 Orientations - *(TBC based on CAO processes)*
October 2023	02/10/2023	Week 3
	09/10/2023	Week 4
	16/10/2023	Week 5
	23/10/2023	Week 6
	30/10/2023	Review Week 7 *Part Time teaching may continue during this week where contact time is required to deliver the full programme*
November 2023	06/11/2023	Week 8
	13/11/2023	Week 9
	20/11/2023	Week 10
	27/11/2023	Week 11
December 2023	04/12/2023	Week 12
	11/12/2023	Week 13
	18/12/2023	Winter Break
	25/12/2023	Winter Break
January 2024	01/01/2024	ALL CAMPUSES RE-OPEN TUESDAY 2nd JANUARY 2024 Assessment/Exams
	08/01/2024	Assessment/Exams
	15/01/2024	

Month	Week Beginning (Monday)	KEY DATES
	22/01/2024	Teaching Commences - Semester 2
	29/01/2024	Week 2
February 2024	05/02/2024	Week 3
	12/02/2024	Week 4
	19/02/2024	Week 5
	26/02/2024	Week 6
March 2024	04/03/2024	Week 7
	11/03/2024	Week 8
	18/03/2024	Week 9

When the student selects the calendar option on the home dashboard, they will see the calendar Option activity screen that contains a calendar view and two buttons. The first button option is to allow the user to view the academic calendar for 2022/2023, this is within a ScrollView layout constraint so the user can view all of the key dates. The student can select a date on a calendar and input an important key date to allow them to keep track and remember. A toast message is displayed when this key date is added successfully to the realtime database.

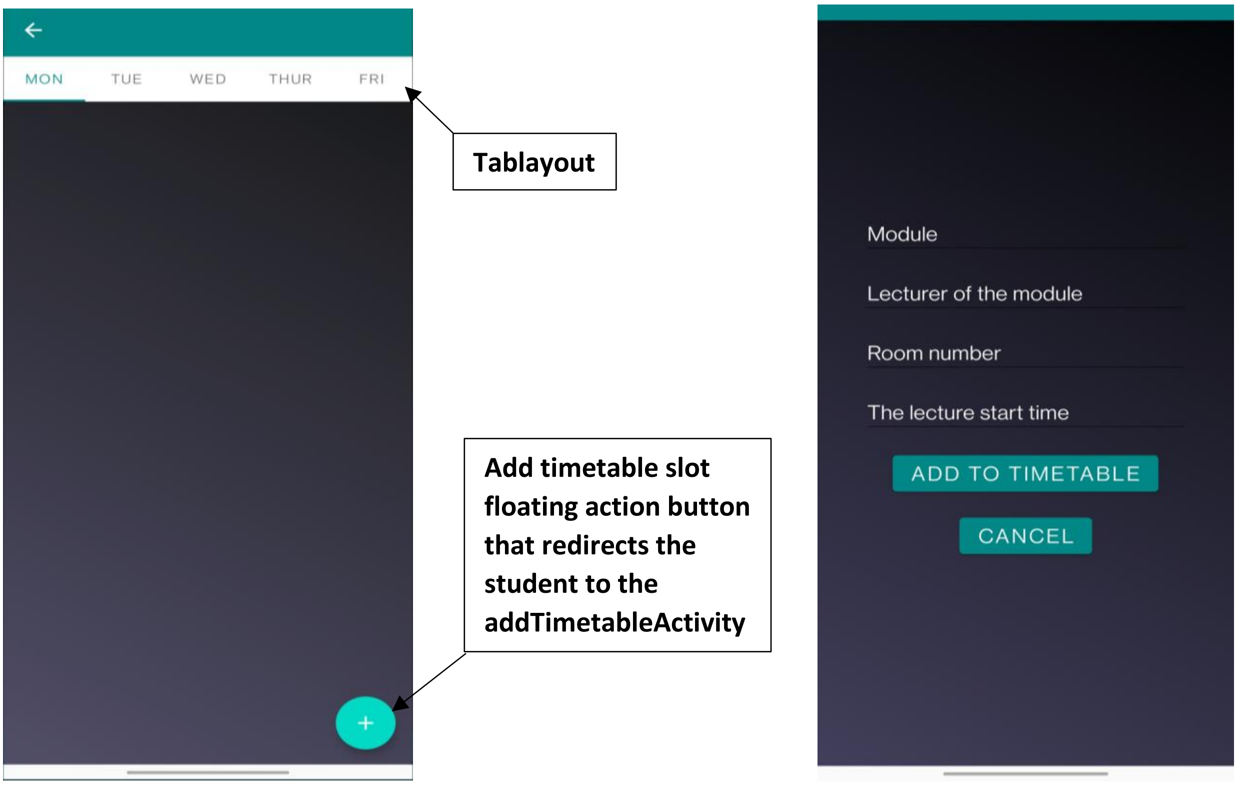
Component:

com.example.fypapplication1.CardOptions.CalendarActivity



```java
private void whenCalendarIsClicked() {
    dbReference.child(inputtedDate).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot datasnapshot) {
            //If this is not null
            if (datasnapshot.getValue() != null) {
                userInput.setText(datasnapshot.getValue().toString());
            }
            else {
              userInput.setText("no value");
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseerror) {

        }
    });
```

- **Timetable Activity**



Tablayout

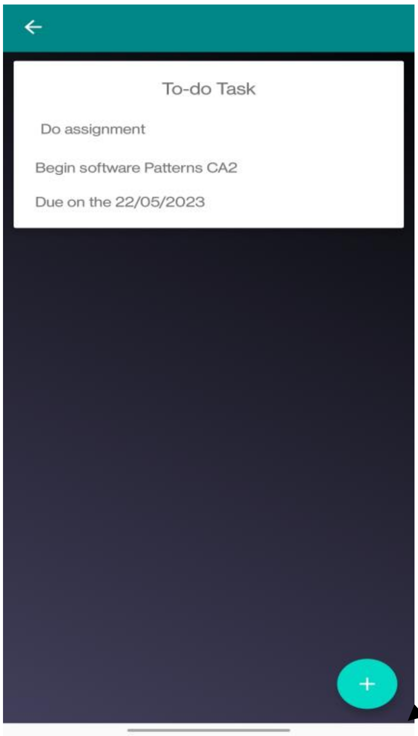
Add timetable slot floating action button that redirects the student to the addTimetableActivity

When the student selects the timetable option on the home dashboard, they will see the timetable Option activity screen that contains 5 fragments within a tab layout the represents the days of the week. The floating action button on the bottom right-hand side of each fragment screen is used to add a slot to their timetable, this contains details such as the module the student is attending, the name of the lecturer, the room number that the lecture is taking place in within the college and the time that the lecture begins. On the right hand side you can see the addTimetableActivity where the student can add their slot to the recyclerview of the timetable activity for each fragment. The student can also delete or update the timetable details.

Component:

com.example.fypapplication1.CardOptions.TimetableActivity

- **To-do list Activity**



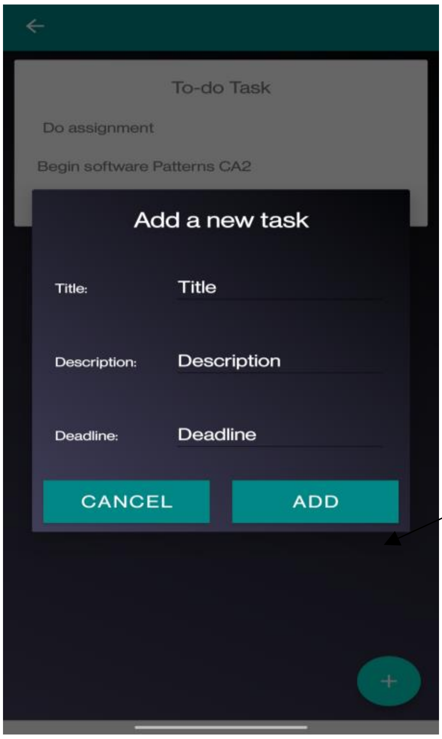
To-do Task

Do assignment

Begin software Patterns CA2

Due on the 22/05/2023

Add to-do task



To-do Task

Do assignment

Begin software Patterns CA2

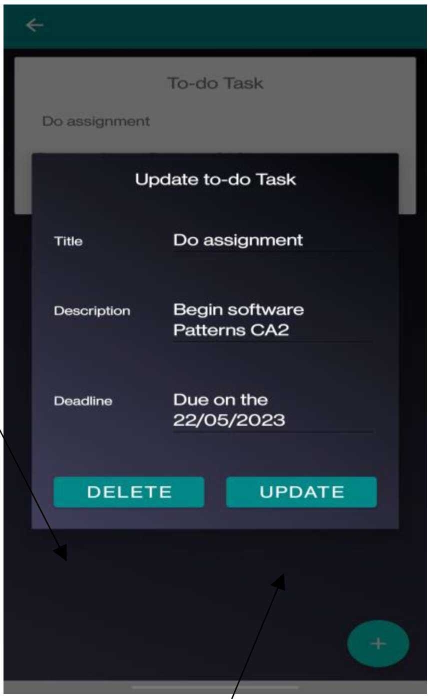
Add a new task

Title: Title

Description: Description

Deadline: Deadline

Add to-do task



To-do Task

Do assignment

Update to-do Task

Title Do assignment

Description Begin software Patterns CA2

Deadline Due on the 22/05/2023

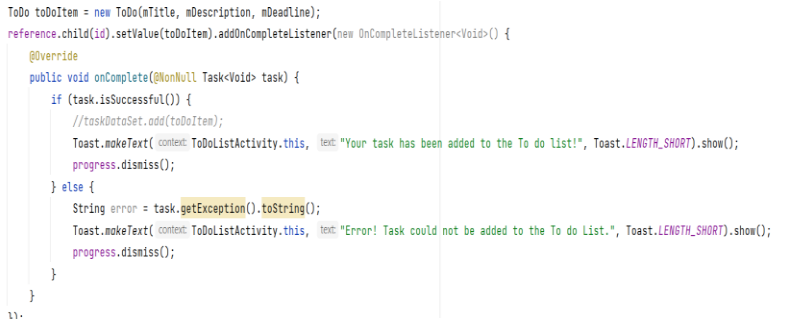
Delete to-do task

Update to-do task

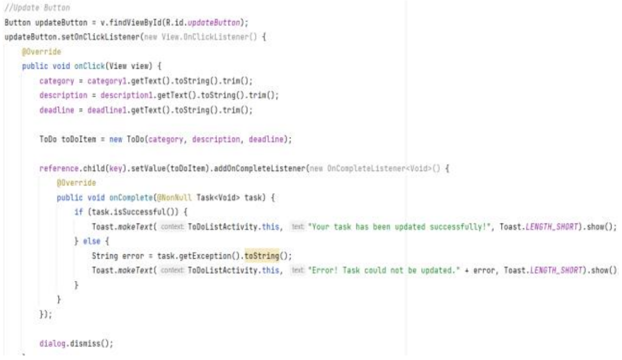
When the student selects the to-do list option on the home dashboard, they will see the to-do list activity screen which is where they can view all of their to-do tasks in the recyclerView array list. The student can add new tasks by selecting the floating action button on the bottom right-hand side. By selecting the to-do task, they can either delete the task or update the task details.

Component:

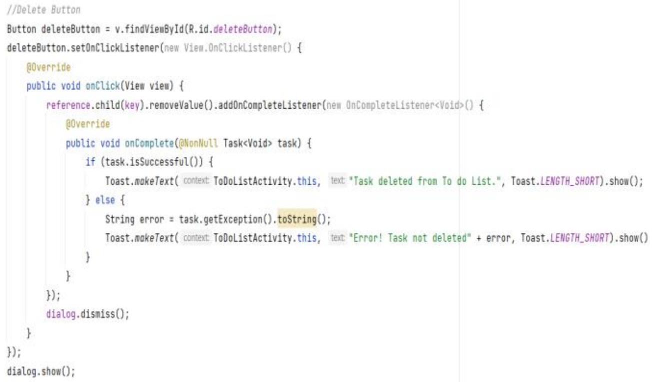
com.example.fypapplication1.CardOptions.TodoListActivity



```
ToDo toDoItem = new ToDo(mTitle, mDescription, mDeadline);
reference.child(id).setValue(toDoItem).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if (task.isSuccessful()) {
            //taskDataSet.add(toDoItem);
            Toast.makeText( context: ToDoListActivity.this,  text: "Your task has been added to the To do list!", Toast.LENGTH_SHORT).show();
            progress.dismiss();
        } else {
            String error = task.getException().toString();
            Toast.makeText( context: ToDoListActivity.this,  text: "Error! Task could not be added to the To do List.", Toast.LENGTH_SHORT).show();
            progress.dismiss();
        }
    }
});
```



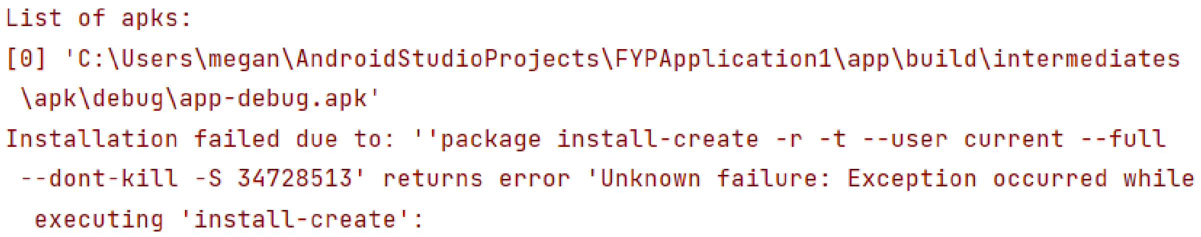
```
//Update Button
Button updateButton = v.findViewById(R.id.updateButton);
updateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        category = category1.getText().toString().trim();
        description = description1.getText().toString().trim();
        deadline = deadline1.getText().toString().trim();

        ToDo toDoItem = new ToDo(category, description, deadline);

        reference.child(key).setValue(toDoItem).addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText( context: ToDoListActivity.this,  text: "Your task has been updated successfully!", Toast.LENGTH_SHORT).show();
                } else {
                    String error = task.getException().toString();
                    Toast.makeText( context: ToDoListActivity.this,  text: "Error! Task could not be updated." + error, Toast.LENGTH_SHORT).show()
                }
            }
        });

        dialog.dismiss();
```



```
//Delete Button
Button deleteButton = v.findViewById(R.id.deleteButton);
deleteButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        reference.child(key).removeValue().addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText( context: ToDoListActivity.this,  text: "Task deleted from To do List.", Toast.LENGTH_SHORT).show();
                } else {
                    String error = task.getException().toString();
                    Toast.makeText( context: ToDoListActivity.this,  text: "Error! Task not deleted" + error, Toast.LENGTH_SHORT).show()
                }
            }
        });
        dialog.dismiss();
    }
});
dialog.show();
```

Issues and Resolutions with Implementation

- The first issue I ran into was when I was trying to run my application on the Android emulator.



```
List of apks:
[0] 'C:\Users\megan\AndroidStudioProjects\FYPApplication1\app\build\intermediates
 \apk\debug\app-debug.apk'
Installation failed due to: ''package install-create -r -t --user current --full
 --dont-kill -S 34728513' returns error 'Unknown failure: Exception occurred while
  executing 'install-create':
```

  I realised that this was due to insufficient storage space, which was causing the installation to fail, as this was such a broad application, the resolution I came to was to use my own android mobile to run the application.

- I had a few issues when implementing Firebase into my application, specifically the Firebase cloud storage. Implementing and deploying these can be quite complex. I would either get the information only displaying in the database or not displaying at all in the database or the application. In particular it took me a while to figure out how to upload a user’s image to the fire store database. I resolved this by paying close attention and following the firebase guides correctly.
- At one stage it took me a while to figure out that versions of SDK’s can change which then would cause errors in my code or it would no longer work as expected. In order to resolve this, I learned the importance of debugging and learned that it is essential to keep up to date with the newest SDK version to ensure that my code was compatible with the other SDK’s that I had.
- At the beginning of the project, I had made the mistake of having a lot of incorrect naming conventions throughout my code. This made it hard for me to understand where I was getting the issues. After this experience I realised how vital it was to have a purpose for each naming convention which should be clear, descriptive and consistent throughout the code in order to prevent future errors and to ensure I could understand all of my code.
- I struggled with using the firebase cloud storage to upload the students profile image url and the groupchat image url. Through use of various guides and tutorials I figured out how to upload the images to Firebase and then retrieve these image URLs to then be displayed when the student ran the application.
- The main piece of functionalities that I struggled with was the social features as these involved a lot of components, so I struggled trying to connect each of these components in my head. There were many times I would have to jot down on a page to fully understand how I would go about implementing these features.

Chapter 5 – Test Plan

Administrator Tests

Name of Test	Test Scenario	Expected Result	Actual Result	Test Result (Pass/Fail)
Introduction Page	Show start page where a user can select to login or register	The first launch activity is displayed	As expected	Pass
Register	The ability of the user to register for an account	User registered and added to the Firebase database	As expected	Pass
Login	The ability of the user to login to their registered account	The user can log in to their account	As expected	Pass
Logout	The ability of the user to log out of their account	The user can log out of their account	As expected	Pass
Retrieve password	The ability of the user to reset their password if they have forgotten it	The user can reset their password through their email if they have forgotten it	As expected	Pass
User profile	The ability of the user to view their user profile	The user should be able to view their profile page	As expected	Pass
Edit profile	The ability of the user to edit their profile details	The user can edit their profile details	As expected	Pass
Home dashboard	Ability to see home dashboard options	The user can see all of the applications features on the home dashboard	As expected	Pass
Transport option	Ability of the user to select this option and select their chosen campus and view the campuses details	The user can chose their campus and view all of their chosen campuses' details	As expected	Pass

Name of Test	Test Scenario	Expected Result	Actual Result	Test Result (Pass/Fail)
Transport route	The ability of the user to input their current location and their desired location to find the quickest route to their chosen campus	The user will be able to find the quickest route to their chosen campus	As expected	Pass
Accommodation Option	The ability of the user to select this option and, select and access their chosen accommodation links	The user will be able to access their chosen accommodation option link	As expected	Pass
Additional Links Option	The ability of the user to select this option and, select and access their chosen additional links	The user will be able to access their chosen option link	As expected	Pass
View to-do list	The ability of the user to view their to-do list items	The user will be able to view their to-do list items	As expected	Pass
Add to to-do list	The ability of the user to add a to-do list item to their to-do list	The user will be able to add a to-do list item to their to-do list	As expected	Pass
Delete to-do list	The ability of the user to delete a to-do list item from their to-do list	The user will be able to delete a to-do list item from their to-do list	As expected	Pass
Update to-do list	The ability of the user to update a to-do list item in their to-do list	The user can update a to-do list item in their to-do list	As expected	Pass

Name of Test	Test Scenario	Expected Result	Actual Result	Test Result (Pass/Fail)
Add key dates to calendar option	The ability of the user to add key dates to their personal calendar	The user can add key dates to their personal calendar	As expected	Pass
View key dates in the calendar option	The ability of the user to view key dates in the calendar	The user can view key dates in their calendar	As expected	Pass
View the academic calendar	The ability of the user to view the academic calendar	The user can view the academic calendar for this year	As expected	Pass
Add to timetable	The ability of the user to add to their timetable	The user can add their lectures to their timetable	As expected	Pass
View the timetable	The ability of the user to view their timetable once the data has been added	The user can view their lectures for the week in the timetable	As expected	Pass
Delete from the timetable	The ability of the user to delete timetable slots	The user can delete a timetable slot from their timetable	As expected	Pass
Update the timetable	The ability of the user to update their timetable slots	The user can update their timetable details	As expected	Pass
View friends	The ability of the user to view all their friends	The user can view their friends	As expected	Pass
Access chat screen	The ability of the user to select a friend to chat to and they will then be brought to the chat screen	The user can access the chat screen with the selected friend	As expected	Pass
Send a message to a friend	The ability of user to send a message to a friend	The user can send a chat message to their friend	As expected	Pass

Name of Test	Test Scenario	Expected Result	Actual Result	Test Result (Pass/Fail)
Receive a message from a friend	The ability of the user to receive messages from their friends	The user can receive messages	As expected	Pass
Search friends	The ability of the user to search through their friends list	The user can search through their friends list	As expected	Pass
Load all previous chat messages	The ability of the user to view all chat messages of previous conversations with a friend	The user can view all of their previous messages	As expected	Pass
View groupchats	The ability of the user to view all the groupchats that they are a participant in	The user can view all the groupchats that they are a participant in	As expected	Pass
Create groupchats	The ability of the user to create groupchats	The user can create a groupchat	As expected	Pass
Send a message in a groupchat	The ability of the user to send a message within a groupchat	The user can send a message in a groupchat	As expected	Pass
Receive messages in a groupchat	The ability of a user to receive messages in a groupchat	The user can receive groupchat messages	As expected	Pass
Search groupchats	The ability of the user to search through the groupchats list	The user can search through the groupchats	As expected	Pass
Load all previous groupchat messages	The ability of the user to view all previous groupchat messages	The user can view all previous groupchat messages	As expected	Pass
View group details	The ability of the user to view the group information	The user can view group information	As expected	Pass

Name of Test	Test Scenario	Expected Result	Actual Result	Test Result (Pass/Fail)
Edit group details	The ability of the user to edit a groupchats details	The user can edit groupchat details	As expected	Pass
Create group event	The ability of the user to create a group meet up event	The user can create a group meet up event	As expected	Pass
View group event details	The ability of the user to view a group event details	The user can view group event details	As expected	Pass

Chapter 6 - Conclusions

Final Conclusions

Overall, the experience of building my very own application was a mix of emotions ranging from excitement and satisfaction to frustration. I am thankful for all of the great opportunities that came with developing this application. By the time of submission, it was a really rewarding valuable experience. I had many challenges throughout, but I continued to stay persistent.

Throughout the development of the application, I faced many challenges. I would not be the most technically gifted developer, therefore developing a full stack application proved to display some difficulties. There were numerous times where my code would not run due to small errors that I had overlooked, from this experience I learned how vital debugging was for programming.

If I could have changed anything about this project, to begin I would have ensured that I knew how to correctly debug as this would have allowed me to use my time a lot more effectively and this would have also alleviated stress as there were times that I was stuck on particular errors for way too long. I would have also spent more time planning and creating wireframes and prototypes, instead of just adding features as I was developing the project. It would have been much easier if I knew how to correctly debug and had all my features in mind before I began the project.

In conclusion, I have learnt a lot from this experience. I believe my coding abilities have significantly improved. When I first started this project, I had quite a hesitant and negative attitude towards coding, but now my outlook has changed as I had quite an enjoyable experience seeing my idea develop and come to light. It has also further enhanced skills such as problem-solving, time management, creativity and debugging. Enhancing these skills will be really valuable for the future for when I am working in a professional setting. Overall, I am really happy with my project, there is still a lot of potential features that could be added, to which I am hoping to implement in the future.

References

- Schilling, H., Wittner, B. and Kauffeld, S. (2022) *Current interventions for the digital onboarding of first-year students in Higher Education Institutions: A scoping review*, *MDPI*. Multidisciplinary Digital Publishing Institute. Available at: https://www.mdpi.com/2227-7102/12/8/551 (Accessed: April 26, 2023).
- Bennett, B. (2023) *Maximizing student onboarding*, *OptimumHQ*. OptimumHQ. Available at: https://www.optimumhq.com/blog/solution-modules-maximizing-student-onboarding#:~:text=Creating%20an%20effective%20student%20onboarding,be%20successful%20in%20their%20studies. (Accessed: April 26, 2023).

Websites

- Android Studio - https://developer.android.com/studio
- Firebase - https://firebase.google.com/
- Retrofit - https://github.com/square/retrofit
- Picasso - https://github.com/square/picasso
- CircularImageView - https://github.com/AtifSayings/CircularImageView
- ImageSlideshow - https://github.com/denzcoskun/ImageSlideshow
- Google maps - https://www.google.co.in/maps/dir/
- Brightspace for Additional Links - https://brightspace.tudublin.ie/d2l/home
- Moodle for Additional Links - https://elearning-ta.tudublin.ie/login/index.php
- Publish for Additional Links - https://timetables.tudublin.ie/
- TU Dublin website for Additional Links - https://www.tudublin.ie/connect/technology-services/wi-fi/
- Exam results login for Additional Links - https://studentapps.dit.ie/BAN8L1/twbkwbis.P_WWWLogin
- Private accommodation for accommodation option – https://www.myhome.ie/
- Private accommodation for accommodation option - https://www.daft.ie/
- Private accommodation for accommodation option – https://www.homestay.com/
- StudentPad for accommodation option - https://www.tudublinstudentpad.ie/Accommodation
- International student accommodation for accommodation option – https://yugo.com/en-gb
- International student accommodation for accommodation option – https://www.mezzino.com/