

Module Code: CS2PJ20

Assignment report Title: Programming in Java

Date (when the work completed): 26th March 2024

Actual hrs spent for the assignment: 20 days.

Overview:

The University of Dreamland application aims to provide a comprehensive and user-friendly platform that provides essential university services and communication facilities within the student community. The app is designed to enhance the student experience by offering a range of features tailored to their academic journey and campus life. Key features of the app include user authentication, personalised timetables, university events, worldwide educational news, and communication details.

Purpose:

The purpose of the app is to streamline various aspects of student life, from academic planning to staying informed about campus events and staying UpToDate with worldwide news. Whether it's checking students class schedule, reading campus events, viewing their personal profile, or accessing university contact information, they can do seamlessly through this app. Overall, this app's purpose is to serve as a valuable tool for students, offering convenience, accessibility, and functionality to support them throughout their academic journey at the university.

Target Audience:

The app caters primarily to students enrolled at the University of Dreamland. It aims to meet the diverse needs of undergraduate and postgraduate students across various disciplines and academic levels. Additionally, prospective students considering enrolment at the university can also benefit from the app's features. It provides insights into campus life, academic programs, and essential university services, helping prospective students make informed decisions about their education.

Key Features:

- **User Authentication:** Users can securely log in to the app using their university credentials, ensuring access control and data privacy.
- **Personalized Timetable:** The app offers personalized timetables for students, displaying their class schedules, lectures, tutorials, and other academic events.
- **Campus events:** Stay informed with the latest campus events, announcements, and events through the app's dashboard. Provide details about events, dates, and descriptions.
- **Personal Profile:** Users can view their personal profiles within the app, including basic information such as their name, student number, and email, also their academic details such as their department location, year of study and course.
- **Educational News:** Offer a news feed feature that delivers educational news and updates from various sources worldwide. Keep users abreast of developments in the education sector.
- **Bottom Navigation:** The app features a convenient bottom navigation bar, providing easy access to different sections and features of the app, including the timetable, dashboard, news feed, personal profile, and settings.

Objectives:

- **Enhanced Student Experience:** The primary objective is to create an intuitive and user-friendly interface that allows students to navigate the app effortlessly. This includes implementing responsive design principles and ensuring smooth interaction with features like personalized timetables, campus events, and news feeds.
- **Convenient Navigation:** The implementation of a bottom navigation bar aims to enhance user experience by providing easy access to different sections and features of the app. This ensures that users can quickly find the information they need without navigating through complex menus.
- **Optimize Academic Management:** Assist students in managing their academic schedules efficiently through personalized timetables. Enable students to access their class schedules, lectures, tutorials, and other academic events seamlessly. This feature helps students stay organized and manage their time effectively.

- **Secure User Authentication:** Ensuring data privacy and security is paramount. The app implements robust user authentication mechanisms to protect users' personal information and maintain the integrity of their accounts.
- **Improved Communication:** Facilitating effective communication within the university community is a key objective. The app provides a platform for students to stay informed about campus events, announcements, and news relevant to their academic journey.
- **Personalized Profiles:** Providing users with personalized profiles allows them to view their academic details, including their department, year of study, and course information.
- **Support for Prospective Students:** In addition to serving current students, the app aims to support prospective students by providing insights into campus life, academic programs, and essential university services. This helps prospective students make informed decisions about their education and enrolment at the University of Dreamland.

Technical Specifications:

- **User Authentication:**
Functional Component: Firebase Authentication
Algorithm:
 When a user attempts to log in, their credentials are sent securely to Firebase Authentication. Firebase Authentication verifies the credentials against the registered users in the database. If the credentials are valid, the user is granted access to the app; otherwise, an error message is displayed.
- **Dashboard/Campus event:**
Functional Component: Dashboard display
Algorithm:
 Fetch event data from predefined arrays.
 Dynamically set event data to respective views and display each event as a separate card and allows user to interact with event details.
- **Personalized Timetable:**
Functional Component: Timetable Management System
Algorithm:
 Retrieve the user's schedule from the database based on their logged-in credentials.
 Populate the timetable UI with the retrieved schedule, displaying class timings, subjects, and locations.
 Allow users to customize their timetable by adding, editing, or removing classes, and synchronize these changes with the database.
- **News:**
Functional Component: News Feed Integration
Algorithm:
 Fetch the latest news articles from API.
 Display the news articles in a scrollable feed, sorted by publication date.
 Allow users to read full articles and navigate to external links for more information.
- **Personal Profile:**
Functional Component: User Profile Management
Algorithm:
 Allow users to view their profile details, including name, student ID, email, profile picture, and other academic information.
- **Bottom Navigation:**
Functional Component: Navigation Controller
Algorithm:
 Implement a bottom navigation bar to provide easy access to different sections of the app.

Define navigation destinations for each bottom navigation item, linking them to their respective fragments or activities.

Handle navigation events and UI updates when users switch between different sections of the app.

User Interface (UI) of the Application:

The Login Screen presents users with fields for entering their username and password for authentication purposes. Additionally, it offers options for new users to register providing a seamless experience for users. The design is clean and intuitive, incorporating branding elements of the university to maintain a consistent visual identity and enhance user recognition.

The Registration Screen features fields for users to input their personal information, including their email, and password. It includes validation checks to ensure data accuracy and security, along with error messages to alert users of any invalid input or existing account detection, thus promoting a smooth registration process.

On the Dashboard/Home Screen, a bottom navigation bar with icons representing different app sections enables users to navigate easily between various functionalities. It provides an overview of essential information such as upcoming classes, campus news, and personal notifications, along with quick access buttons or shortcuts for common actions like accessing the timetable or profile, enhancing user convenience and productivity.

The Timetable Screen showcases a calendar layout displaying class schedules with time slots and subject details. Users can view the timetable by day facilitating efficient schedule management and organization.

In the News Feed, users can scroll through a list of news articles featuring titles, publication dates, and brief summaries. Thumbnail images accompanying each news article add visual appeal, while the full article view offers rich text formatting and multimedia content support, providing an immersive reading experience.

The Personal Profile Screen prominently displays the user's profile picture at the top, with sections for viewing and editing profile information such as name, email, and contact details. Users can also upload or change their profile picture using image cropping and resizing functionality, allowing for personalization and customization of their profile.

The Settings Screen offers contact details for student support which enhance communication. Additionally, users can sign out account if needed.

Finally, the UI design adheres to principles such as consistent colour scheme, typography, and iconography for a cohesive visual experience. Responsive layout design ensures compatibility with different screen sizes and orientations, while intuitive navigation flow and clear labels promote ease of use.

Application Implementation:

User Authentication:

The university student app integrates Firebase Authentication to enable user login and registration functionalities. It utilizes Firebase's authentication services to securely handle user authentication processes.

In the Login activity, Firebase's "signInWithEmailAndPassword" method is used to authenticate users with their email and password. Upon successful authentication, a toast message, "Login successful" is displayed and users are redirected to the main activity. In case of authentication failure, error messages "Login failed" is displayed to the user. (*see figure 1 & 2*)

Similarly, in the Register activity, Firebase's "createUserWithEmailAndPassword" method is employed to create new user accounts with email and password. Upon successful registration, a toast message, "Account created"

Firebase Authentication ensures that user credentials are securely managed and validated. It handles tasks such as email verification and password reset functionalities, enhancing the overall security and reliability of the authentication process. (see figure 5)

[illegible][illegible]

Figure 3: Registration fail

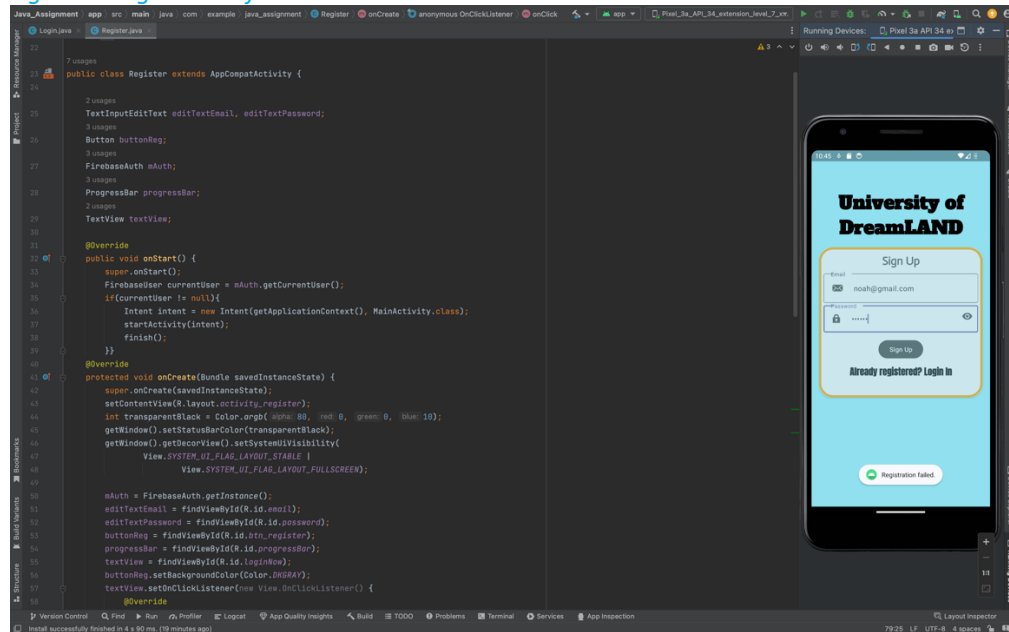


Figure 4: Account created

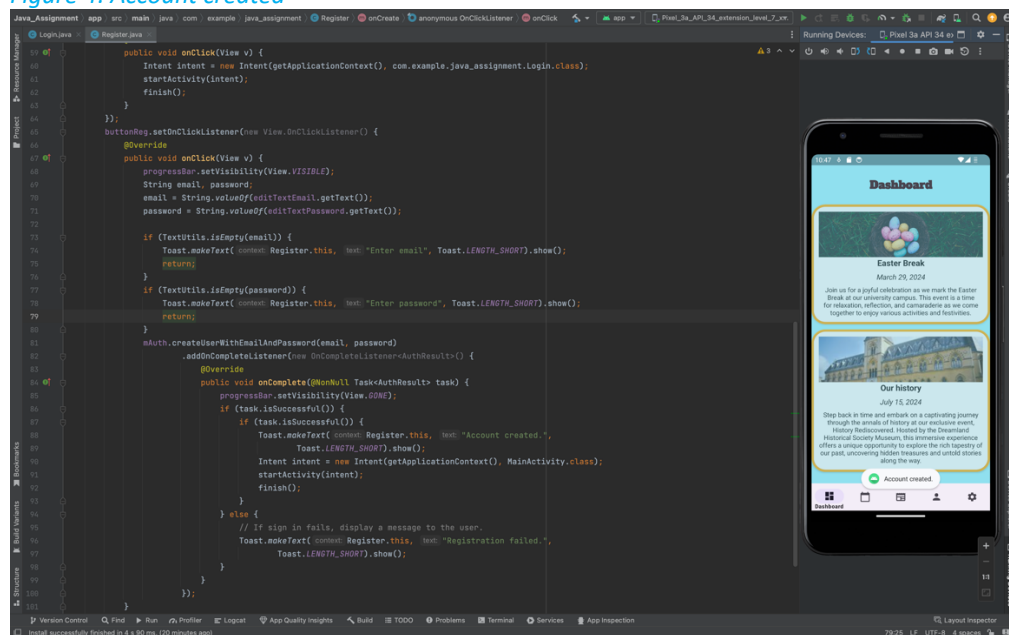
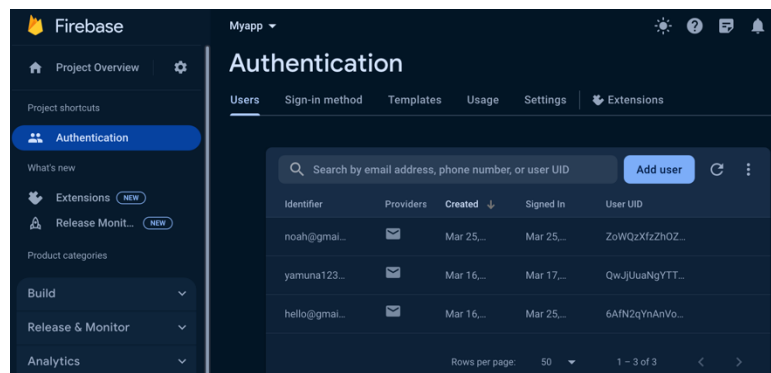
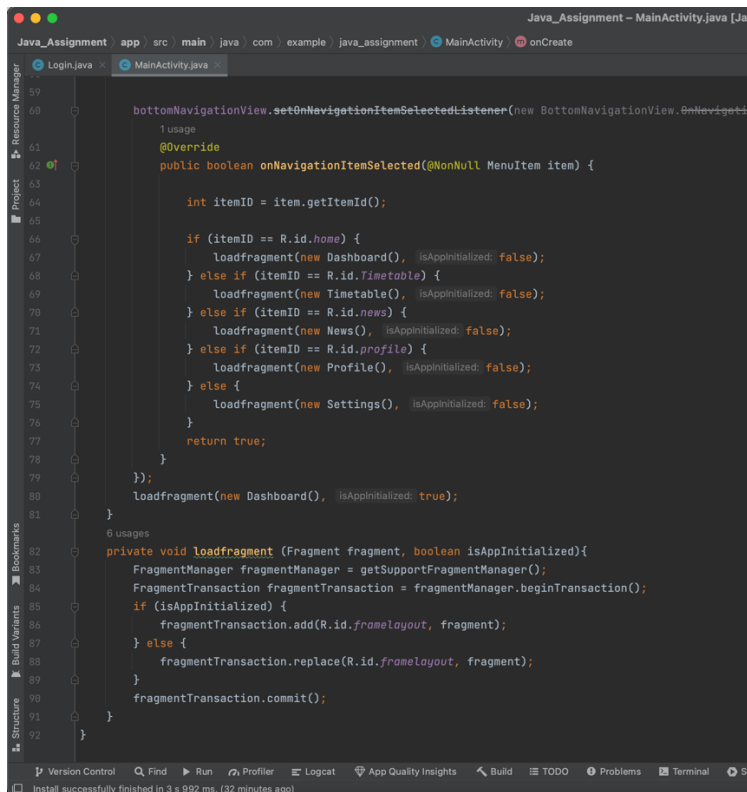


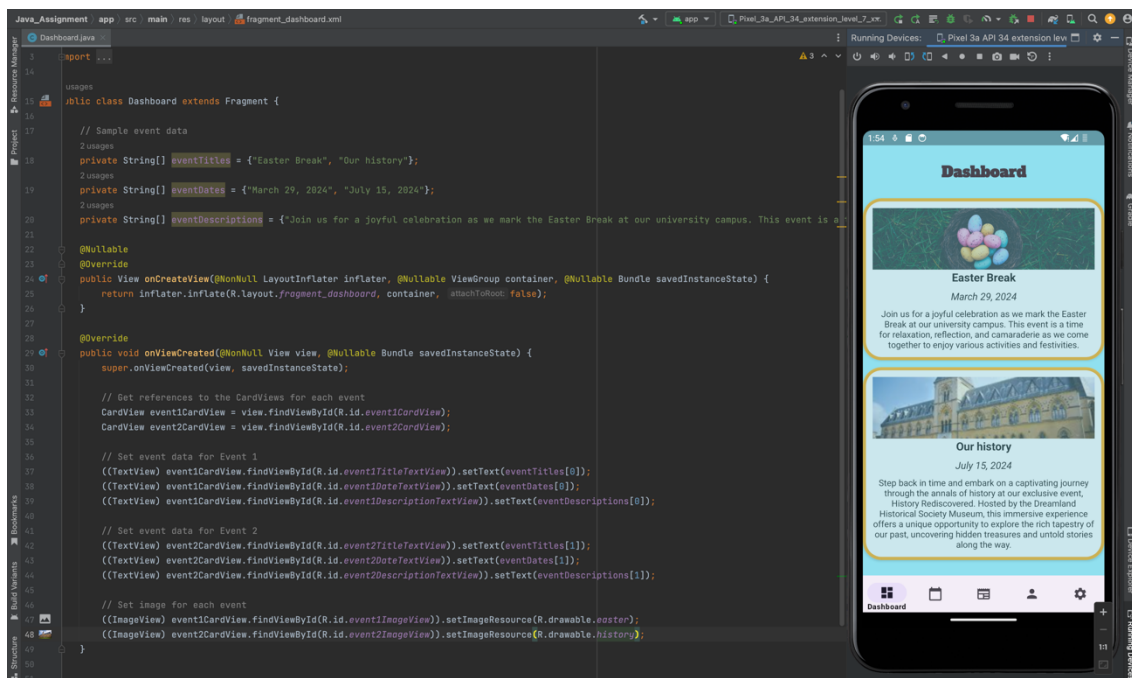
Figure 5: Registered Users





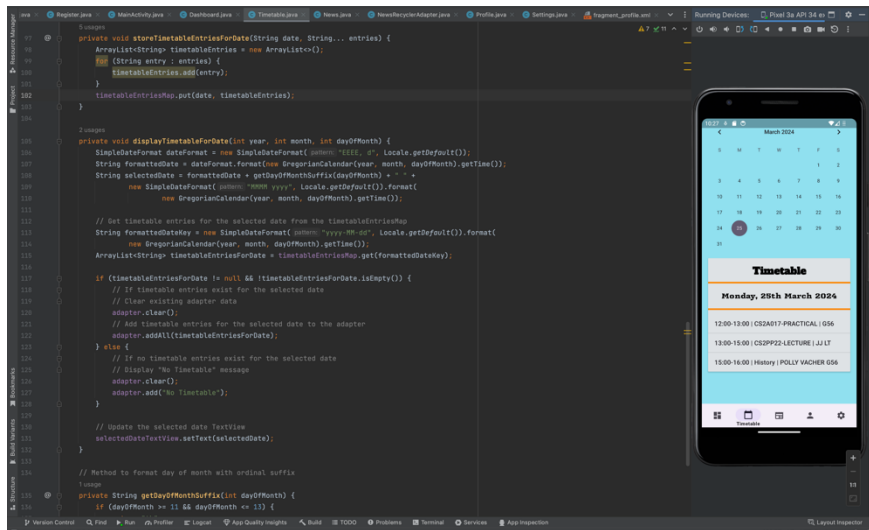
The main activity of mobile application incorporates a bottom navigation view for seamless navigation between different app sections. It initializes Firebase Authentication to check whether a user is logged in, and if not, redirects them to the login activity. Once authenticated, the user's email is displayed in a text view. The bottom navigation view enables switching between fragments representing various app sections such as the dashboard, timetable, news feed, profile, and settings. Fragment transactions are used to load the selected fragment into the frame layout container.

Figure 6: Dashboard



Dashboard responsible for displaying upcoming events within the application. It inflates a layout file containing 'CardViews' to represent each event. In the 'onViewCreated' method, it retrieves references to the 'CardViews' and populates them with event data such as titles, dates, and descriptions. Additionally, it sets images for each event. This implementation allows users to view details about upcoming events in a visually appealing manner, enhancing the user experience and engagement with the application.

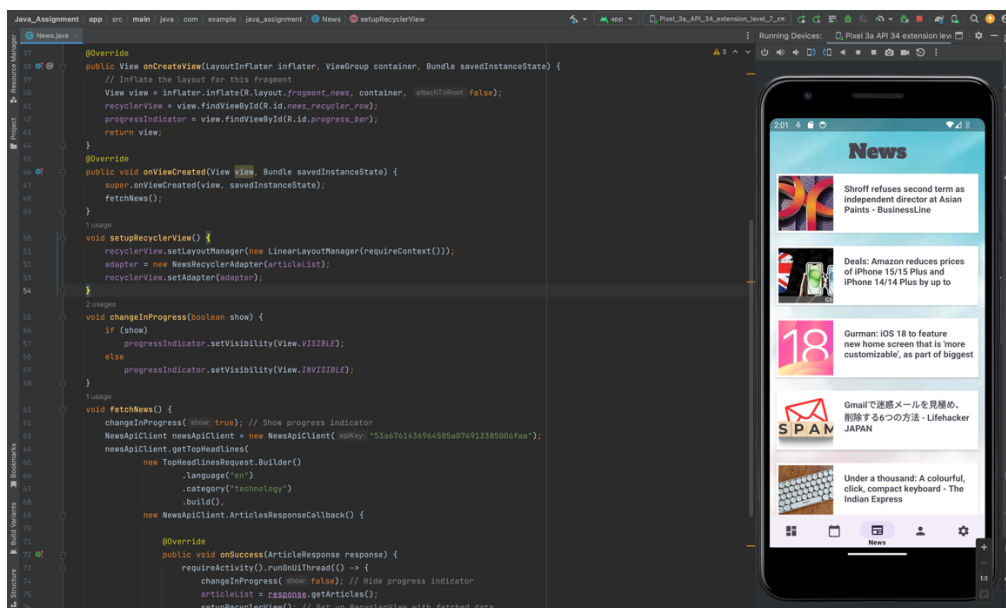
Figure 7: Timetable



Timetable fragment, responsible for displaying a timetable of events based on a selected date using a 'CalendarView'. In 'onCreateView', the fragment initializes its layout and sets up the 'CalendarView' and 'ListView' components. It also populates a HashMap with timetable entries for specific dates. When a date is selected in the 'CalendarView', the 'displayTimetableForDate' method is called to fetch and display the corresponding timetable entries. The method formats the selected date and retrieves timetable entries from the HashMap, updating the 'ListView' accordingly. Additionally, the code includes a utility method to format the day of the month with an ordinal suffix for better readability.

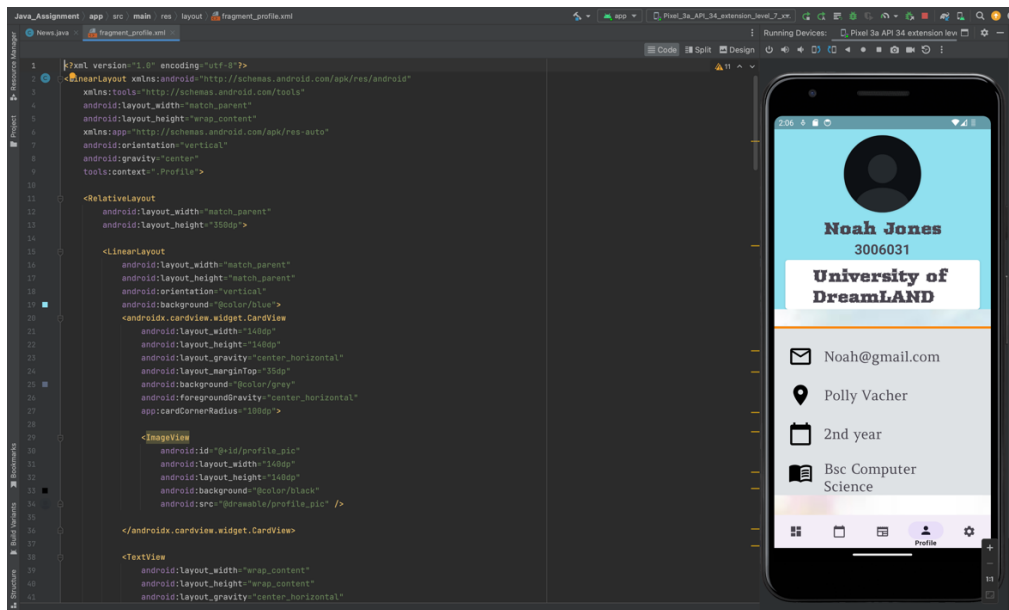
method is called to fetch and display the corresponding timetable entries. The method formats the selected date and retrieves timetable entries from the HashMap, updating the 'ListView' accordingly. Additionally, the code includes a utility method to format the day of the month with an ordinal suffix for better readability.

Figure 8: News



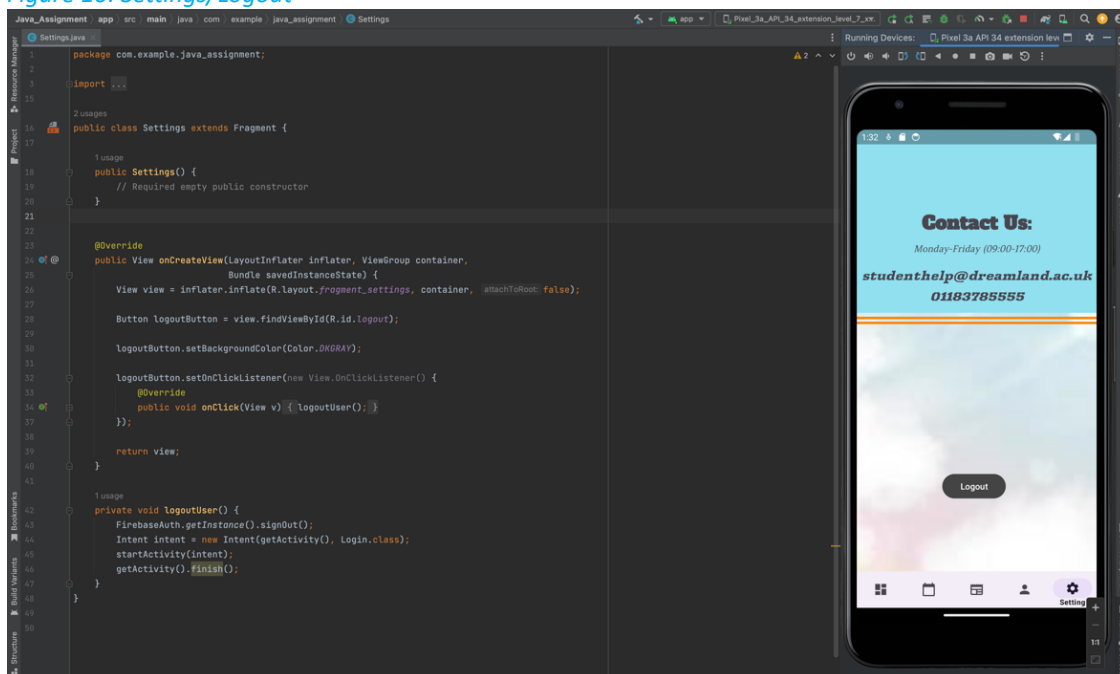
News and NewsRecyclerViewAdapter, responsible for fetching news articles from an API and displaying them in a 'RecyclerView'. In the News fragment, the 'onCreateView' method inflates the layout and initializes the 'RecyclerView' and progress indicator. 'onViewCreated' calls 'fetchNews' to retrieve news articles from the API, while 'setupRecyclerView' configures the RecyclerView's layout manager and adapter. The 'fetchNews' method uses the 'NewsApiClient' library to fetch top headlines in the technology category asynchronously. Upon success, the 'articleList' is updated with the fetched articles, and the 'RecyclerView' is updated accordingly. The NewsRecyclerViewAdapter class defines a 'RecyclerView' adapter that binds news article data to corresponding views in the 'RecyclerView' rows. 'onBindViewHolder' populates each row with data from the 'articleList', while NewsViewHolder holds references to the title TextView and image ImageView for efficient view binding.

Figure 9: Profile



Profile consists of two main sections: the upper section displays the user's profile picture, name, and ID, while the lower section includes contact information and academic details. The upper section is designed with a `RelativeLayout` containing a `CardView` holding the profile picture and `TextView`s for the user's name and ID. The lower section is structured with a `LinearLayout` and contains multiple `LinearLayout`s, each consisting of an `ImageView` and a `TextView` representing different types of information such as email, address, academic year, and course.

Figure 10: Settings/Logout



The Settings fragment handles the user's logout functionality. Upon inflation of the fragment's layout, it initializes a logout button. The button's click listener triggers the `logoutUser` method, which signs out the current user using `FirebaseAuth.getInstance().signOut()`. After signing out, it creates an intent to navigate the

user back to the login screen (Login activity) and finishes the current activity to prevent the user from returning to the previous screen with the back button.

Discussion of additional features:

The additional features showcase a creative and user-centric approach to feature development. In the Dashboard fragment, upcoming events are presented with visually appealing CardViews, demonstrating an understanding of users' need for quick access to event information. The News fragment creatively fetches and displays top headlines in the technology category, with a RecyclerView and progress indicator enhancing the user experience by efficiently handling news articles and providing transparency during data retrieval. The NewsRecyclerViewAdapter creatively binds news articles to RecyclerView items, delivering visually appealing content and efficient image loading. Additionally, profile fragment uses RelativeLayout and LinearLayouts which allows for flexible arrangement of elements, ensuring a visually balanced and organized layout. Together, these components highlight the understanding of users' needs, contributing to an engaging and user-friendly experience within the Dreamland app.

Summary:

The Dreamland mobile application has been successfully completed, designed to provide users with a seamless experience for accessing news articles, viewing upcoming events, and managing their profiles. Built on the Android platform, the app incorporates key features such as user authentication using Firebase Authentication, real-time news updates fetched from external APIs, and an intuitive user interface crafted with RecyclerView and CardView components. The app's functionality is complemented by its responsive layout design, ensuring optimal usability across various devices. With its comprehensive set of features and user-friendly interface, the Dreamland app offers a valuable tool for users to stay informed and engaged with relevant news and events. The project achieved its primary objectives of delivering a user-friendly interface, integrating real-time news updates, and providing a seamless user experience.

Reflection:

The journey of developing the Dreamland app has been both challenging and rewarding. Throughout the development process, several key learnings were gained, including proficiency in Android app development, integration of third-party APIs, and implementation of Firebase authentication. Moreover, the project fostered teamwork, communication, and problem-solving abilities, contributing to personal and professional growth. Achievements include the successful implementation of various features such as the news feed, event dashboard, timetable and user authentication. Additionally, the project provided valuable insights to problem-solving and project management.

Future Work:

- **Advanced Analytics:** Implement analytics tools to track user engagement, content popularity, and app performance metrics for informed decision-making.
- **Social Integration:** Enable users to share news articles and event details directly from the app to their social media accounts, fostering engagement and expanding the app's reach.
- **Community Forums:** Introduce community forums or discussion boards within the app where users can engage in conversations, ask questions, and share insights on various topics related to news and events.
- **Enhanced Personalization:** Introduce features for users to customize their news feed based on preferences and interests.
- **Notification Enhancements:** Enhance the notification system to provide timely alerts for breaking news, upcoming events, or personalized recommendations, keeping users informed and engaged.
- **Accessibility Features:** Ensure the app meets accessibility standards by incorporating features such as screen reader compatibility and adjustable font sizes.