**Mobile Application Programming – 1**

**Term Paper, fall 2021**

Krutarth Patel

William KamgnaTango

Prof. Haim Levkowitz

Android is the widely used operating system with about 88% of mobile phones running android. Android studio is a platform on which one can develop and test android mobile phone applications. It gives us option to use different programming languages to develop applications. This fall, we came up with an idea to create a social app through which restaurants can donate leftover foods at the end of the day.

**Abstract**

Our mobile app programming project is focused to solve food waste problem. Majority of food waste is happening in restaurants, farms, and retails stores. We brought up an idea to design an app by which a restaurant can upload pictures with description of fresh leftover food on this app. All users in nearby location could get notify if they are interested in getting it. Here, the restaurant can decide if they want to donate or sell this food item at lower prices. We are using Java programming language on Android studio to develop this application. To store and retrieve user information such as profile images, post images, name, location, we are using Firebase. The newest authentication, database, and storage dependencies are added in Gradle module to connect to our firebase console.

**Implementation**

The first step is to learn and understands Java and learn different tools of Android studio. Android studio documentation is very useful to learn how to get started. Also, while completing the introductory assignments, we learnt more and more about different tools of this software. We started creating main activity.

There are two important types of files. Xml and Java. Layout files are static screens where we can drag and position images, buttons, texts and much more. All these can be manipulated by java file bonded with this layout file. Both together creates an activity. We created a main activity as an introduction page. It displays our app icon and project name for 3 seconds and then intents to login activity where user can login or register. An application can have numerous activities. But at a time, only one screen can be in user’s focus. intent is used to switching from one to other activity and passing information.

In our first week, we focused on developing the login and register page for our app. Using EditText, a user can input email and password. An editText is used to get user input and save them in variables. There are clickable textviews through which user can switch between login and register screen. A textView is used to display a text on screen. Two button, login and register are positioned on the screen. Clicking register button, user’s email and password will be saved in firebase authentication. Then user can click on login button and the email and password, and the Login activity verifies the credentials with the one that are saved in firebase. An onCompleteListener function is used to give tasks to perform after a task is completed. Here, we are verifying a user if it is existing in firebase. If the task is successful, the screen intents to home screen else it gives a toast, displaying the error. We have onStart and onCreate functions implemented to verify the existing user. With that, user will stay logged in every time the app is opened. This is how we login a user to this app.

In the second week, we focused on working with home screen layout and back-end programming. It starts with a tool bar on the top of the screen. A separate tool bar layout is created and then bonded it with this home screen using Toolbar library. It consists of a title, and a drawer button. This button will pull out the drawer when clicked once. It consists of options of profile, post, logout, and more. A separate layout named menu is created that has all these options. Drawer also holds a user profile image and username. A separate layout named header is created that holds these two things and it is bonded with the drawer layout using inflateHeader function. In this activity, there are two variables, firebaseAuth and userRef. FirebaseAuth holds the instance of current user and by using it, we can identify that user existence in firebase database. On start of this home activity, onStart function will verify it the profile image, full name, town name of current user is existing in firebase database. If nothing is found, it will redirect user to profile activity to get user information. If the information is found, then user will be able to stay on home screen and has access to all options in drawer layout. At last, user can also logout. When logout option is selected, using firebaseAuth, we call signOut function and intent user to login page.

In the third week, we started designing the profie activity. The profile activity is using CropImage library. The circularImage is set on cliackable. So, When the CircularImage view is clicked, the screen is intended to gallery of user’s phone from where user can select a profile image. This file type is selected as image. So, user will only be able to select images from file directory. Here, CropImage library allows users to crop an image and set it as profile picture. Using the firebase storage reference, this profile image is saved in a folder with a unique name (all user ids are unique) and the path to this image is saved in current user node of firebase database. And further, the full name, town name is also saved in database after the Save button is pressed. Then the screen intents back to the home screen. The profile image in the drawer layout is displayed using Picasso library. The drawer in home screen allows user to visit again in the profile activity to modify the existing information.

As we progressed in fourth week, we started working on post activity. Our plan was to let user post picture with description and save it in database. So, a post activity is created, and it contains a image view, text input for description and buttons. On clicking search button, a intent to gallery is generated and from where user can select any image to post. On clicking upload button, this image is stored in firebase storage using storage reference in a folder that only contains post images. Here, we have tried to give each image a unique name. So, the code combines current user id with current date and time including seconds to create a unique name of the post image. The path to this image is stored in firebase database in post node, which also holds description, date time, and user information who posted it. User will be notified through a toast that that image is posted and saved.

**Goal**

Our goal was to display all this user post on home activity as a stack like social networking ap. The difference here will be that these posts get deleted automatically after 12 hours or the user who posted it can also delete it. Another difference was to set all posts as clickable so that if user clicks on the post, it will redirect user to map activity where user can easily get directions to the place. However, we are still working on getting dependencies to display all user posts on home activity. The links that professor provided to learn how to develop apps were so useful. Also, the Android studio documentation, YouTube tutorials from different channels, and dependencies created by other developers on GitHub were very useful to us to create this app. During this period, we encountered many errors and spend time solving them. The plus point here was that we learned new skills of programming and acquired knowledge of Java for creating application.

**Resources used to develop this application**

* Android Studio to develop the application.

<https://developer.android.com/studio>

* Android studio design libraries.

<https://developer.android.com/topic/libraries/support-library/packages>

* Circular Image View in this application is possible by using dependency from

<https://github.com/ArthurHUb/Android-Image-Cropper>

* Retrieving the images from firebase database and storage uses dependency from

<https://square.github.io/picasso/>

* YouTube Tutorials to learn basics:

<https://www.youtube.com/watch?v=fis26HvvDII&ab_channel=freeCodeCamp.org>

* Using Firebase services to store and retrieve data for this application.