Zara El Alaoui

CS 443

Professor Bo Sheng

December 21st, 2021

# Final Project Report

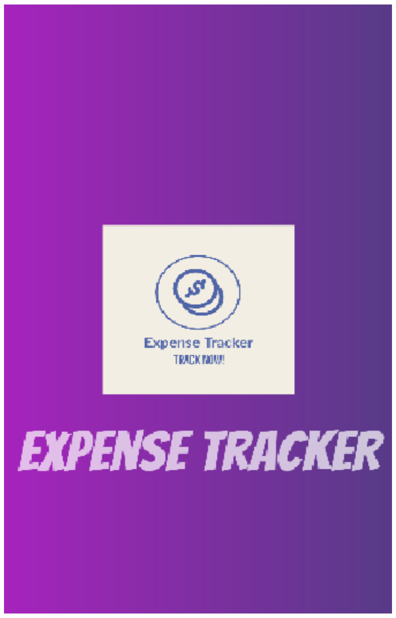
# Group Expense Tracker Application

## I. Project Statement:

When friends, coworkers, or simply classmates plan a trip or a day out together, it is always possible that someone will pay for an uber ride, while another group member pays for drinks or dinner or even the hotel costs. The group might need to track these expenses and split the costs at the end of the trip among the members. So this application will serve as a group expense tracker that will add up the expenses paid by each member with built-in functions that will make it easier for the user to track the costs without ending up with a mess.

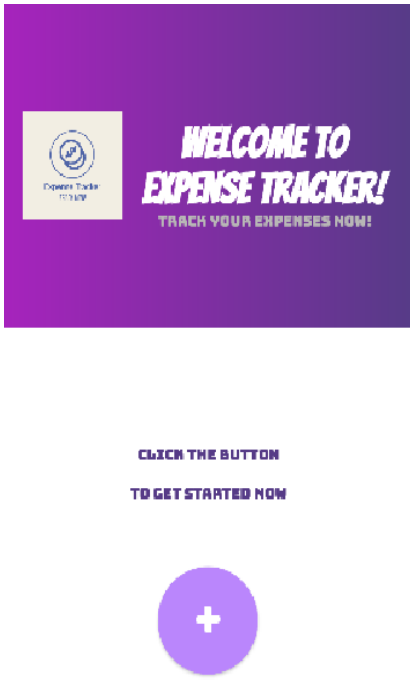
## II. Application Design:

### A. Splash Screen:



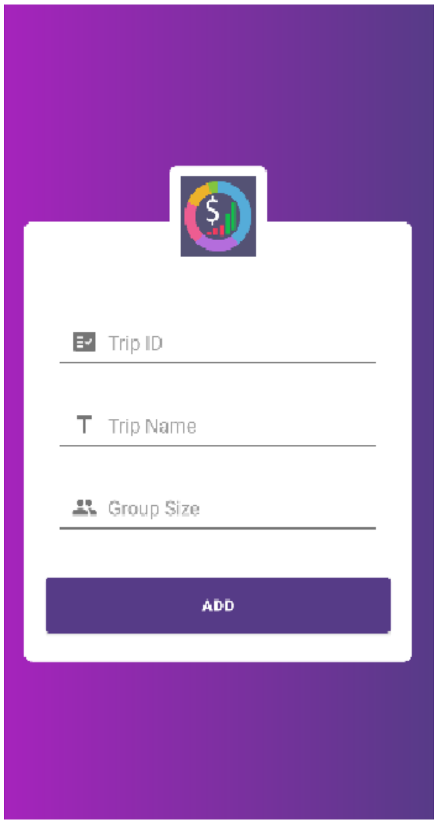
Following a pattern from going from general to specific, my application starts off by displaying a splash screen. In this splash screen, an application logo and its name are animated for 3 seconds. I designed the logo on a free website online and I made use of google fonts to have an app signature in the splash screen. I also made use of animation to animate the components of the screen.

### B. Home screen:



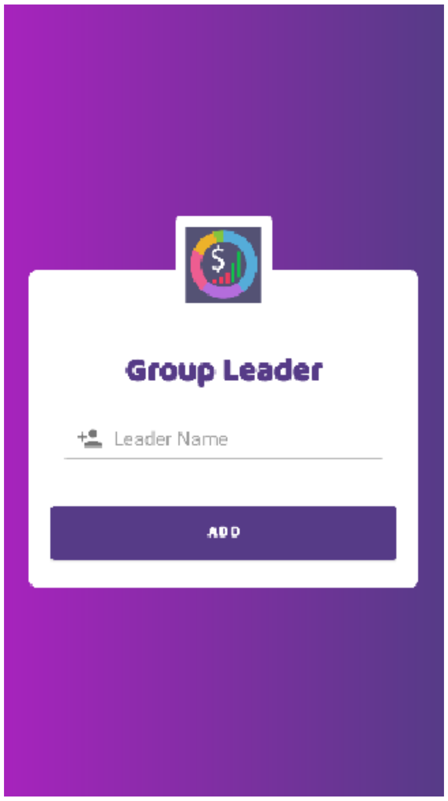
After showing the splash screen for 3 seconds, the user will be able to see the main screen. The main screen uses a modern design and gradient background color. On the home screen, the user can see the logo and a welcome message that is also animated. At the bottom of the screen, the user can see a circular button with a plus sign to get started.

### C. Add a Trip:



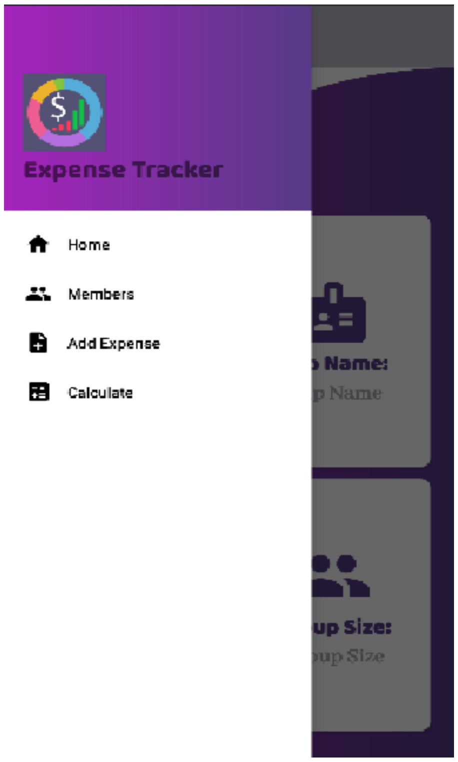
In this screen, the user is able to enter information about a trip to have the expenses tracked. In this screen, I made use of a modern design. The layout displays an elevated card view. In this view, the user is prompted to enter a trip ID, trip name, and group size. I made use of vector assets to add icons to the left of each text view to make the UI look modern. I also made use of a gradient background color. Once the info is entered, the user can click the add button to move to the next screen. Once the add button is clicked, all the info gets stored in shared preferences, so it can be retrieved later on.

## D. Add Group Leader:



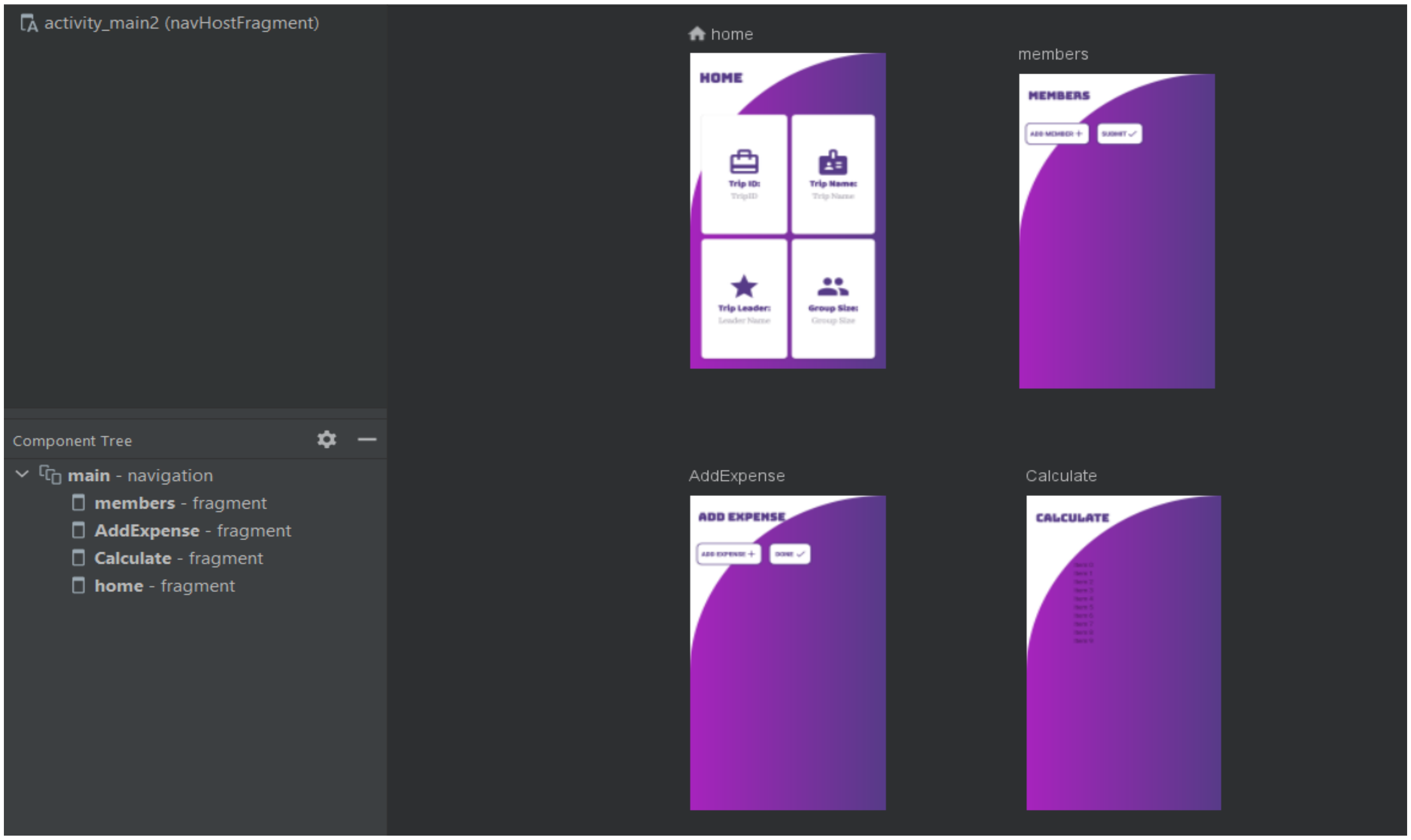
This screen is designed the same way screen C is designed. The user sees a card view in which gets prompted to enter a group leader’s name. The name gets stored in shared preferences to be retrieved later. This was a good practice of storing data at shared preferences and retrieving them.

## E. Navigation Drawer:

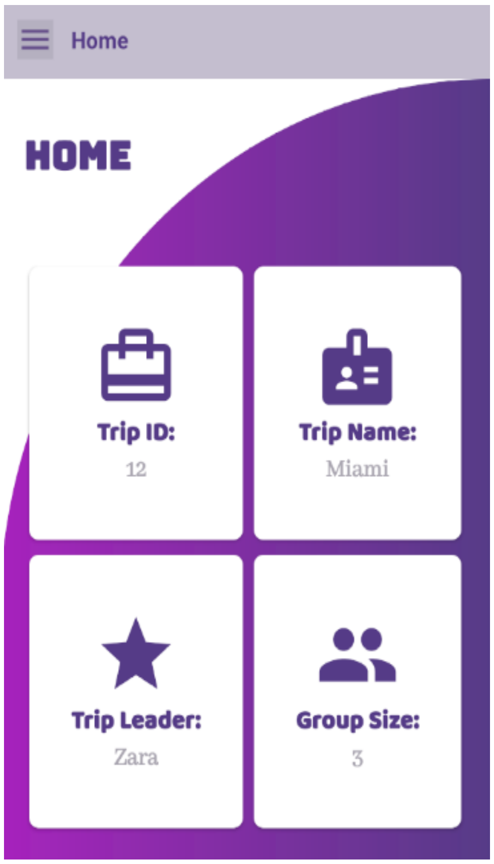


Once the user clicks add in screen D, the user will be able to see a navigation drawer. This navigation drawer has a modern design with a menu, header, and navigation graph. All the components make use of gradient background, vector assets icons, and modern text views and edit texts. The menu has 4 main components: Home, Add Members, Add Expenses and Calculate.

All these components are connected with a navigation graph to let the user switch between the menu items. Each menu item click displays a fragment. This was a great practice of fragments. The drawer starts with a fragment navigation host that hosts all the other fragments.



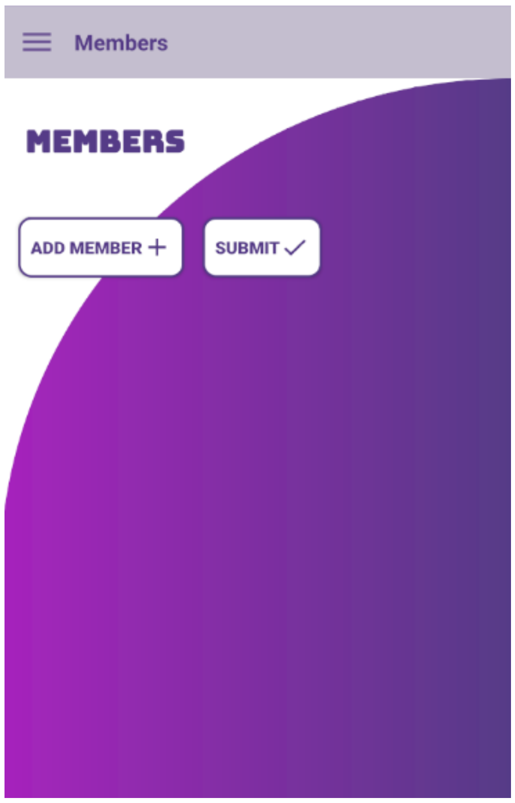
1. **Home:**



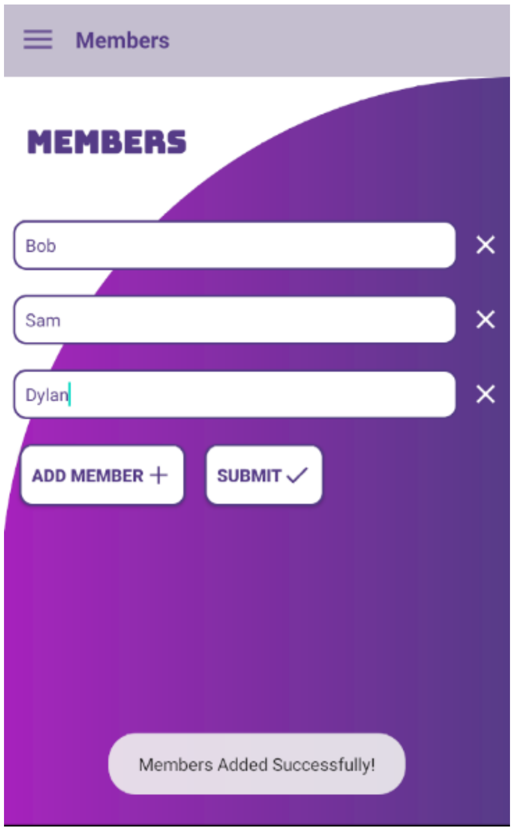
The home screen is basically one of the four fragments in the application. It uses a modern background design using a shape component and playing with the corner radiuses. On the home screen, I designed a modern dashboard with four card views. These four-card views display the data about the trip: trip ID, trip name, group size, and leader name.

In this screen, I made use of shared preferences to retrieve the data entered by the user on-screen C. This was a great practice of storing small data sets and retrieving them. The dashboard card views are designed in a modern way having vector assets that show icons for each data representation.

2. **Add Members:**



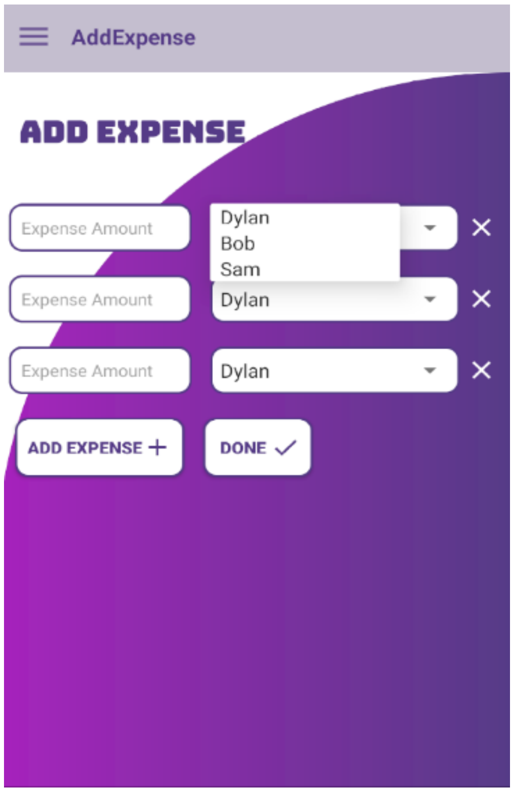
This screen is the second fragment from the navigation graph. In this screen, the user is able to enter members of the group. Since we don't know how many members will be there in a group, I had to use dynamic views. I made a layout for a single member entry and I used layout inflater to inflate it to the fragment once the user clicks a button "Add a Member". To make it more interactive, I also added a close icon to the right to each TextEdit in case the user wants to delete an entry.



I also created a “submit” button. As the user clicks it, the data gets validated and shows a toast message if not all the entries are filled. If all entries are filled, it shows a toast message that the members have been added successfully. Once the user clicks that button, the data gets stored in shared preferences in the array list.

Since ArrayList cannot be stored in shared preferences, I had to convert it to a set and store it as a set and convert it back to an ArrayList when I want to retrieve it.

3. **Add Expense:**

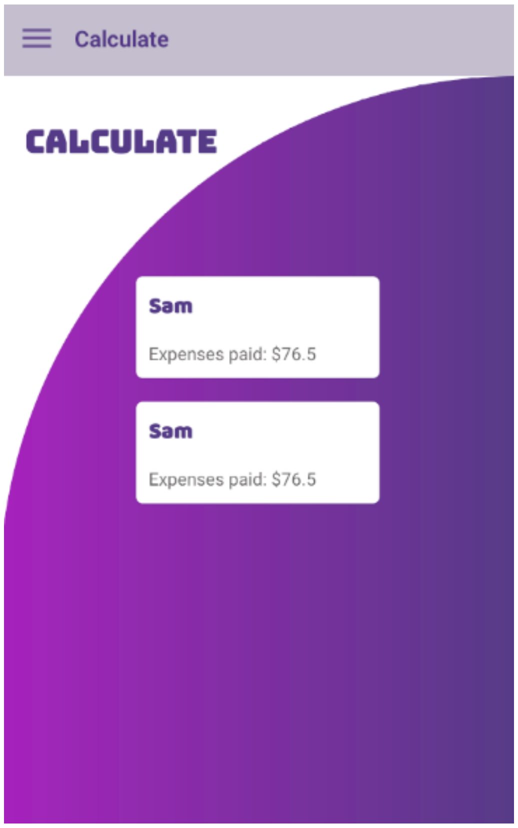


This screen is the third fragment of the navigation graph. In this screen, the user is able to enter expenses paid by each member of the group. In this screen, I also made use of dynamic views for the user to be able to inflate a dynamic view for each expense he wants to add. I retrieved the set of members’ names from shared preferences that were stored when the user clicked submit on the previous screen. I converted the set back to an

ArrayList and fetched it to a spinner so the user can be able to select a member from the spinner without having to write the member name again and add an expense amount. The design is very similar to the previous screen having two buttons “Add Expense” and “Done”.

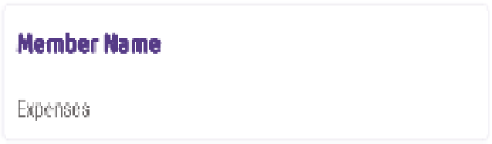
Once the user clicks “Done”, the data gets stored in a hashmap having string keys for member names associated with double values for the expense amount.

**4. Calculate:**



This screen is the last and fourth fragment from the navigation graph. It gets displayed when the user clicks the menu item “Calculate”. This screen makes use of the recycler view of dynamic card views that get added based on the expenses added and the number of members.

In this section, I had to use an Arrayadapter that gets the data from the previous screen and add it to an ArrayList of Member class. This class represents a member by the member’s name and the expense he paid. The getters and setters of the class allow the array adapter to get the member name and get the expense and display it in a card view that gets added dynamically in the recycler view.



This section is supposed to also do calculations of how much each member has to pay to the other member to split the costs, however, since my partner left the group in the last two days before the deadline, I had to do all the work by myself and I was not able to complete this part.

## III. Application Implementation and Evaluation:

One main thing that was used throughout the app is storing data in shared Preferences. I made use of shared preferences to store small data that is used in the app. I encountered a problem when I wanted to store a hashmap, I had to add a dependency of GSON in order to store the hashmap data structure into a string. I used the dependency from the documentation and was able to successfully store it and retrieve the date from it when needed.

I made use of animation to animate the splash screen. I also made use of vector assets and image assets to introduce all the icons throughout the app. I made use of ArrayAdapter, recyclerView, CardView, Navigation Drawer, and its components: Menu, Navigation Host Fragment, Navigation Graph, and Header.

## IV. References:

As for references, one of the main documents that I have been using throughout the process of developing this application is the Android Developer Documentation, as well as Youtube videos, and StackOverFlow that helped me resolve many errors that I encountered while developing this app:

- **Basic Syntax:** https://developer.android.com/guide/components/activities/intro-activities
- **Resolving Errors:** https://stackoverflow.com/
- **Designing UI:** https://www.youtube.com/watch?v=U1dSfjNnd1Q

## V. Experiences and Thoughts:

I enjoyed working on this application. I was able to gain real practice in using features that we discussed in class and even go beyond and do research about either how to develop new features or develop on the ones we learned in class.

Since I have never done android programming before, I was able to learn a new experience working with Android Studio, Java, and XML. Android Studio seemed very confusing at first, however, I have done some training and watched videos online and I was able to become better at it.

As for my experience with XML, I tried to develop a modern user interface design to practice XML skills and the architecture of the layouts of my activities. I gained experience using different types of layouts such as Relative, Linear, and Constraint Layouts. I also practiced the use of animations and fragments.

Overall, this project was a great experience, however, the only issue that I confronted is time management. This was a little bit stressful to manage with taking 5 other courses and having to balance between all of the finals of other courses and working on the project at the same time.

One other issue that arises is having project management issues, as my teammate decided to leave the group and not help at the very last two days. This has put a lot of stress on me trying to finish all the projects by myself in the last two days. Group management was tough and affected the outcome of the project as it was not fully finished.