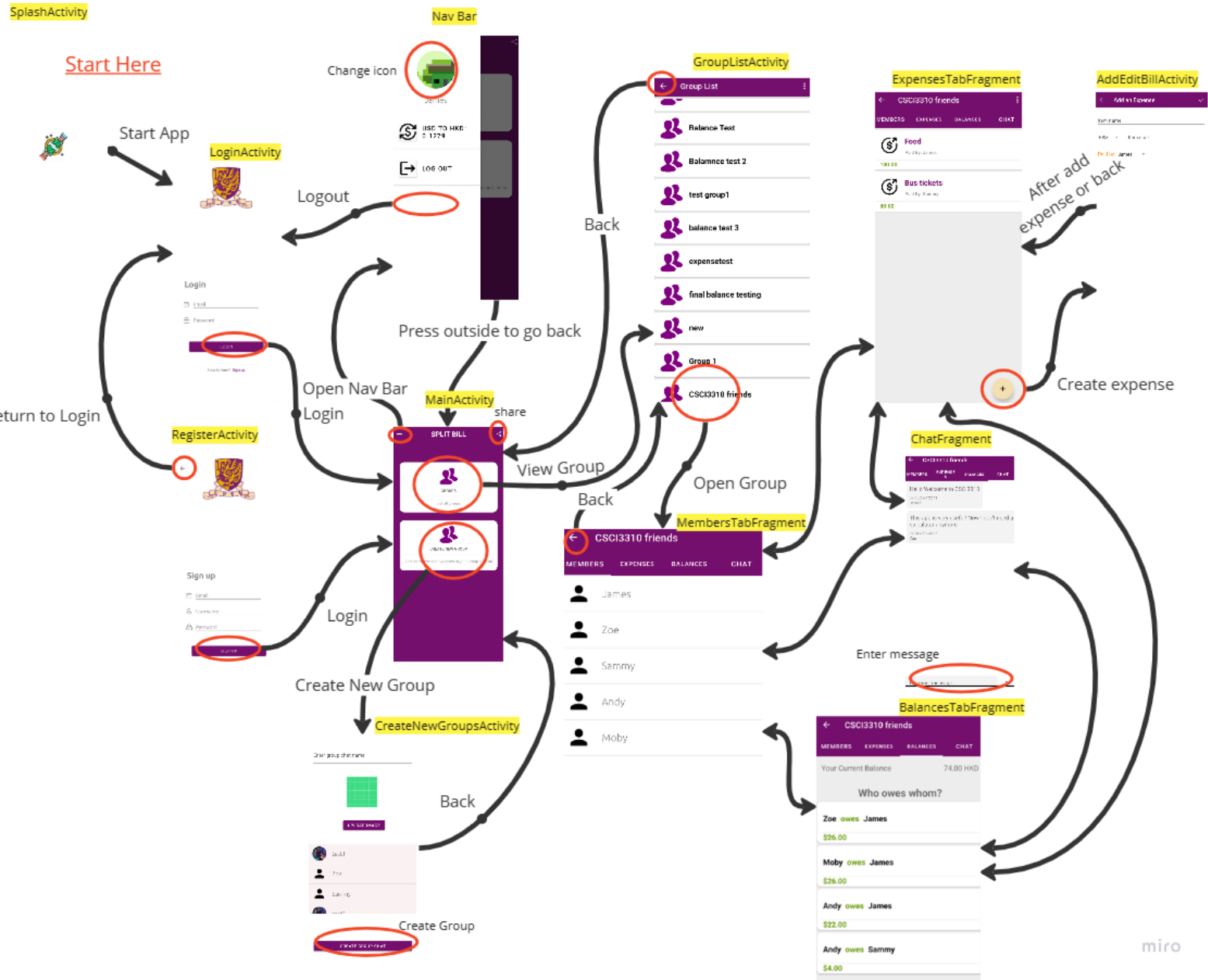


Report: Phase 2 – CUSplit

1. Brief app layout flow



2. User Guide & Build Instructions

2.1. Developing platform

The app is developed on both MacOS and Windows Android Studio(2023.1.1 Hedgehog), supporting Android SDK version 34, using AVD as testing machine with the following configurations.

Device	Pixel 3a
OS	Android 14.0 x86_64
Startup orientation	Portrait

2.2. Permissions/Hardware requirements

Permissions	Set to
Internet	Connected
Location (Fine Location)	While Using the app
Media/Image	Allow All

Please ensure that Internet is connected, and you should set location permission to “While using” when you first log in the app.

2.3. Special dependencies (Not all)

com.github.bumptech.glide:glide:4.15.1
pl.droidsonroids.gif:android-gif-drawable:1.2.19
com.google.firebase:firebase-bom:33.0.0
com.google.code.gson:gson:2.10.1
com.google.android.gms:play-services-location:21.2.0

2.4. Build instructions

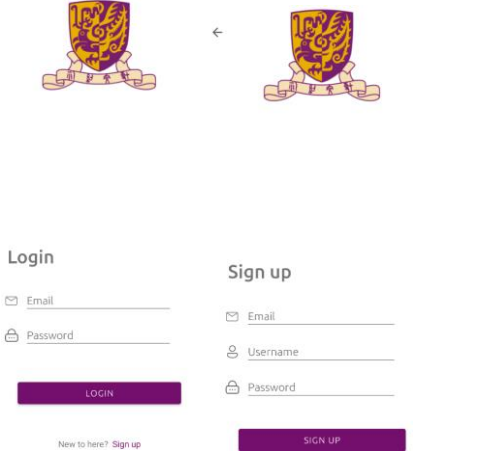
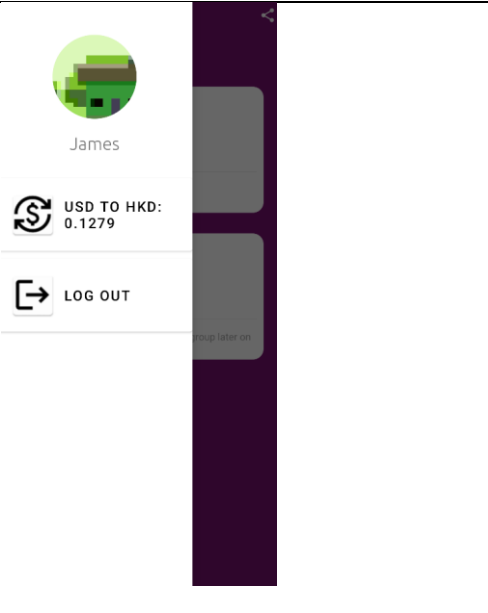
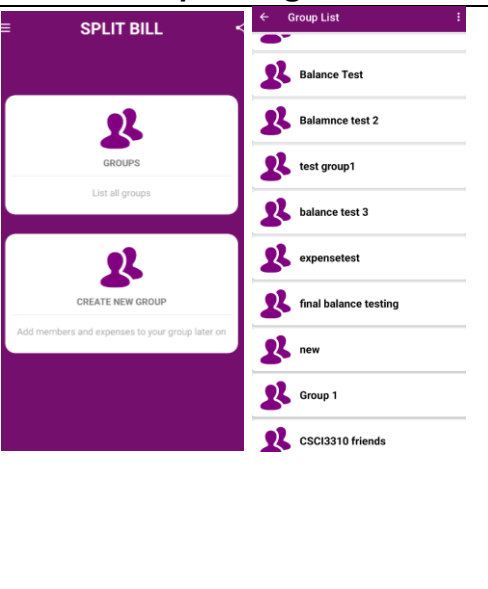
1. Download the project and open with Android Studios(2023.1.1)
2. With Internet connection, install all dependencies
3. Setup an Android device with the specifications mentioned above and start the emulator.
4. Run the project by clicking the “Run” button.

3. Functionalities & Features

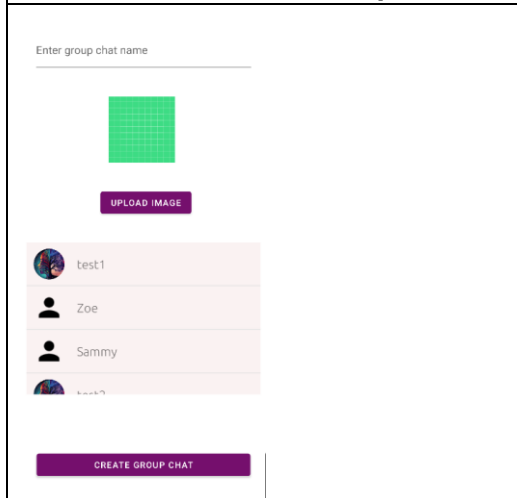
Text in Bold refers to important components in our project.

<Text bracketed red> refers to Jumps to other functionalities & features.

Text Underlined refers to major functionalities of the Activity/Component.

<h3>3.1. User Authorization</h3>  <p>The screenshot shows a mobile app interface for user authorization. At the top, there are two identical crest logos. Below them, the screen is divided into two sections: 'Login' and 'Sign up'. The 'Login' section has fields for 'Email' and 'Password' with a 'LOGIN' button below. The 'Sign up' section has fields for 'Email', 'Username', and 'Password' with a 'SIGN UP' button below. At the bottom of the 'Login' section, there is a link: 'New to here? Sign up'.</p>	<h3>User Registration</h3> <p>Users can first register with their email address, username, and password. Email and Username will be checked unique by comparing with Firebase.</p> <p><u>createUserWithEmailAndPassword()</u> was provided by Firebase, so we directly used it to do signup function.</p> <h3>User Login</h3> <p>Users can login with the email and password. Similarly, <u>signInWithEmailAndPassword()</u> was provided by Firebase and directly used.</p> <h3>User logout</h3> <See 3.2 Sidebar>
<h3>3.2. Side Bar</h3>  <p>The screenshot shows a mobile app sidebar. At the top, there is a circular profile picture of a person named 'James'. Below the name, there is a currency conversion section showing 'USD TO HKD: 0.1279'. At the bottom, there is a 'LOG OUT' button with a right-pointing arrow icon.</p>	<h3>Profile Image</h3> <p>Users can choose a photo from their gallery as their profile picture. The change is immediately reflected on the app and is stored in Firebase and will be reflected at next login. A self-declared utility class ImageUtil.java uploads the image to Firebase Storage (<u>uploadImageToFirebase()</u>). At login, the Firebase storage is fetched and Glide class adds the profile image to the View.</p> <h3>Username</h3> <p>Uses Firebase database to retrieve the current user's username</p> <h3>Exchange rate</h3> Shows the exchange rate in current country to HKD by fetching from Firebase. <See 3.6 Currency Conversion> <h3>Logout</h3> Logs out the user and returns to Login Page
<h3>3.3. Group Management</h3>  <p>The screenshot shows a mobile app interface for group management. The top bar is purple with the text 'SPLIT BILL'. Below it, there is a 'Group List' section with a list of groups: 'Balance Test', 'Balance test 2', 'test group1', 'balance test 3', 'expensetest', 'final balance testing', 'new', 'Group 1', and 'CSCI3310 friends'. Each group entry has a person icon and a text label.</p>	<h3>Groups</h3> <p>After clicking “Groups” button, user will navigate to see all groups they currently belong, from a RecyclerView using an adapter. The View only shows groups the user is currently in (not the groups that they do not have access to). <See 3.5 Group View></p> <h3>Create New Group</h3> <See 3.4 Create Group>

3.4. Create New Group



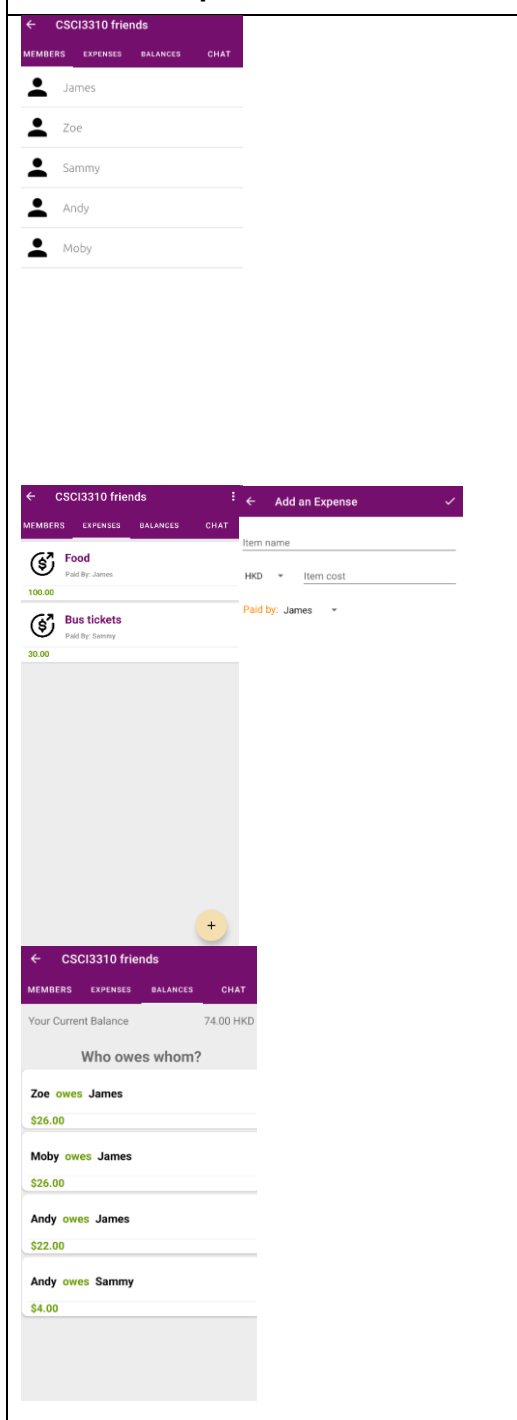
Enter Name: TextView to input group name

Group Image: (Not yet implemented)

Select Members

RecyclerView is used to show all other people using the app with Adapter. User can click on one or more people, then click the “Create Group Chat” button to create a new group. The group information is stored in Firebase.

3.5. Group View



Members

Members in the group (excluding the user themselves) are displayed in a RecyclerView. User can scroll down if there are many items.

Expenses

User can add expense transactions with the + button on the bottom right. User can choose the Item name, currency(will be converted to HKD immediately) <See 3.6 Currency Conversion>, amount, and paid by which person in the group. Changes will be updated on Firebase as well. Each transaction is displayed on the RecyclerView using an Adapter, containing the description, paid by whom, and amount of the expense.

Balance (Statistics Visualization)

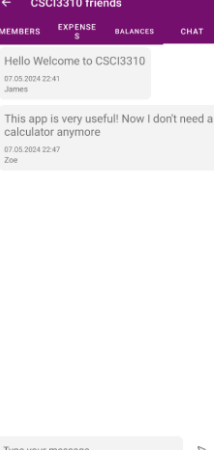
The balance of the current user is shown on top. Then, the net amount due for each other user is displayed, in the format of “**[Person A] owes [Person B] \$X**”. All amounts are converted to HKD, using and the expenses are evenly distributed among all other members with an algorithm as described below.

Pick the largest element from debtors and the largest from creditors. Ex: If debtors = {4,3} and creditors={2,7}, pick 4 as the largest debtor and 7 as the largest creditor.

Now, do a transaction between them. The debtor with a balance of 4 receives \$4 from the creditor with a balance of 7 and hence, the debtor is eliminated from further transactions. Repeat the same thing until and unless there are no creditors and debtors.

The priority queues help us find the largest creditor and debtor in **constant time**. However, adding/removing a member takes **O(log n) time** to perform it.

Optimization: This algorithm produces correct results but the no of transactions is not minimum. To minimize it, we could use the subset sum algorithm which is a NP problem.

	<p><u>Chat</u></p> <p>The message and time sent are fetched from Firebase and displayed. Messages sent will also contain the time sent and the sender of the message(username).</p>
<p>3.6. Global location-aware system</p>	<p>Self-declared class LocationUtil.java is implemented. Upon creation of the Main Activity, it would call the LocationUtil to get the location.</p>
	<p>LocationUtil.java first requires user to provide location permission. Then the FusedLocationProviderClient .getFusedLocationProviderClient() will provide the latitude and longitude to Main Activity.</p>
<p>3.7. Currency Conversion</p>	<p>Self-declared class CurrencyUtil.java is implemented. Upon getting the location data (during Main Activity creation), it would call CurrencyUtil to get the currency and exchange rates real time using the ExchangeRate-API. https://v6.exchangerate-api.com</p>
	<p>CurrencyUtil.java uses the current location and the Geocoder class to find the country code. Then use the country code and Currency class to get the currency code. Finally, use the currency Code and ExchangeRate-API to get the real time exchange rates and upload to Firebase Database. The rates are used in the <3.2 Side Bar> and <3.5 Group View></p>

4. Reflection

A. Incomplete features & Difficulties of development

i. Update and Delete groups, group members, expense transactions

Currently, the app only supports creation of a group, with fixed initial members. Expenses also cannot be updated or deleted. This is due to time limitations, and the algorithm of calculating final balance would change if we allow changing members. There is also issue with members who owe money to/from others suddenly leaving the group.

ii. Image attachment for some parts

As shown in [<3.2 Nav Bar>](#), the profile icon can be created and updated. We have not implemented Group icon customization and upload bill receipt to transactions yet due to development timeline.

iii. Privacy Concerns - Currently new users can see the group and expenses even if they're not part of the group. Allowing unauthorized users to view group information and expenses may compromise the privacy of group members. Users may share sensitive financial information within the group, assuming it's only visible to members. Unauthorized access could lead to privacy breaches and expose personal or financial data to individuals who shouldn't have access to it. To address these issues, we need to implement proper access controls and user authentication mechanisms. Only authenticated users who are members of a group should have access to group information and expenses.

B. Possible enhancements in the future

i. Currency conversion

This app assumes the base currency of all transactions to be HKD, which means the main target audience of the app is people residing in Hong Kong. The app could be further extended to allow users to choose the base currency.

Additionally, the currency conversion API only fetches from a free API that provides daily updates on exchange rates. If a paid plan is used, exchange rates can be updated in real time.

ii. User Authentication actions

Currently the app does not support “Forget Password” or “Change Password” features. These can be implemented to enhance user experience.

iii. Pretty UI

Designing an intuitive navigation structure with clear hierarchy and logical flow ensures that users can easily navigate through different sections of the app. Strategic placement of navigation menus, buttons, and interactive elements optimizes usability and enhances the overall user experience.