GitHub Username: http://github.com/Alluajay

Track Your Pal

Description

There are many applications to chat or share info with your friends. Our Application opens a new and exciting way to keep track on your friends journey. Our application provides an awesome environment to share his/her travel experience. This application is specially designed

for Travellers who rise their adrenaline by exploring various parts of the world. They can share their experience at realtime.

The application can also help journalists as they can provide on spot news to the masses.The application offers an exciting user interface through which one can picturise their onspot views to the world.

Intended User

The application covers a wide range of users that include the following :

- Travellers
- Journalist
- Bloggers
- Socialist
- General audience (Friend zone)
- And others

Features

The main features of the application are:

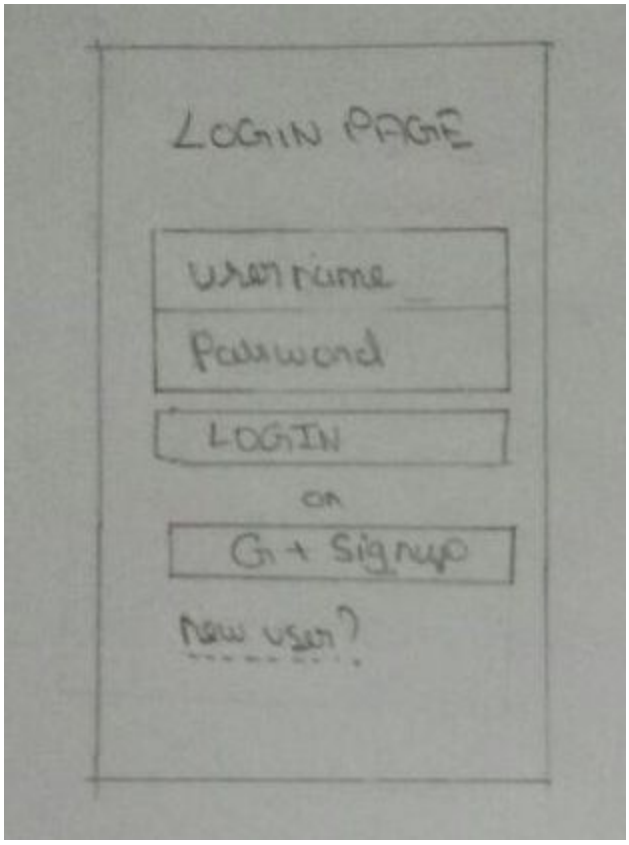
- Share current location with others.
- Share your views about the place you are in.
- Open a group chat box for your journey.
- Share images from your location to your friends.
- Personalise your chat head.

User Interface Mocks

The application provides various activities for users. They are listed below

Screen 1

Login



This page is designed to carry out user authentication through regular email/password and G+ sign up.It also provides link to new page where the new users can sign up. All authentication process is carried out by firebase authentication system.

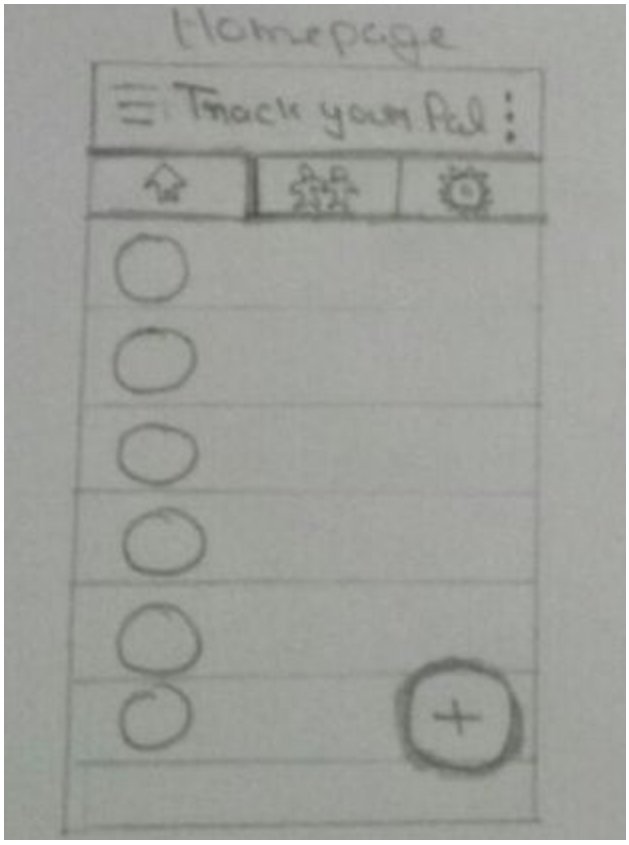
Screen 2

Sign up

This activity is intended to create new user with default email id and password.After creating new user the user will be redirected to login page

Screen 3

Homepage

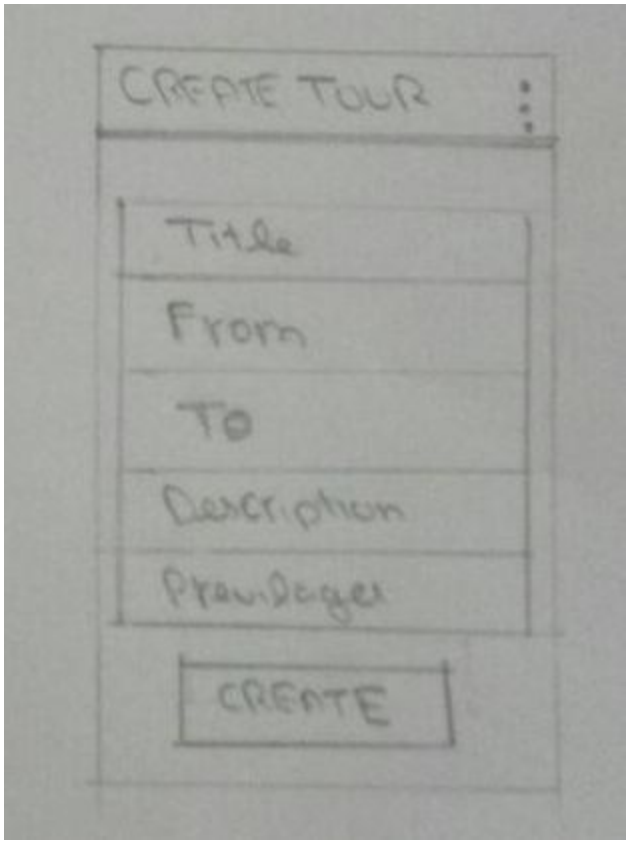


This is a tabbed activity. In this activity there will be three tabs:

1. The main tab has a FAB and a list view.The fab creates a new tour for a user. And the listview contains the tour list of the user's friend.
2. The second tab is for finding the friends. This Tab has a search textview on the toolbar through which the user can find the users based on their emailid. The TAB also provides a list view of friends.
3. The third tab is the settings tab which is used to set the app configurations.

Screen 4

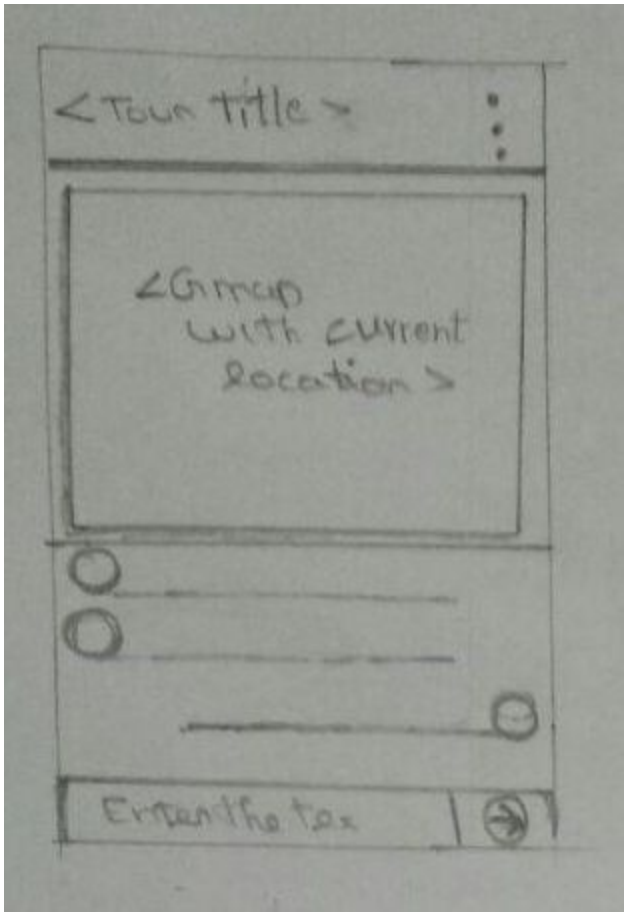
Create new tour



This activity is designed to initiate a tour for a user. The UI has various fields through which the user can provide details about his/her tour.

Screen 5

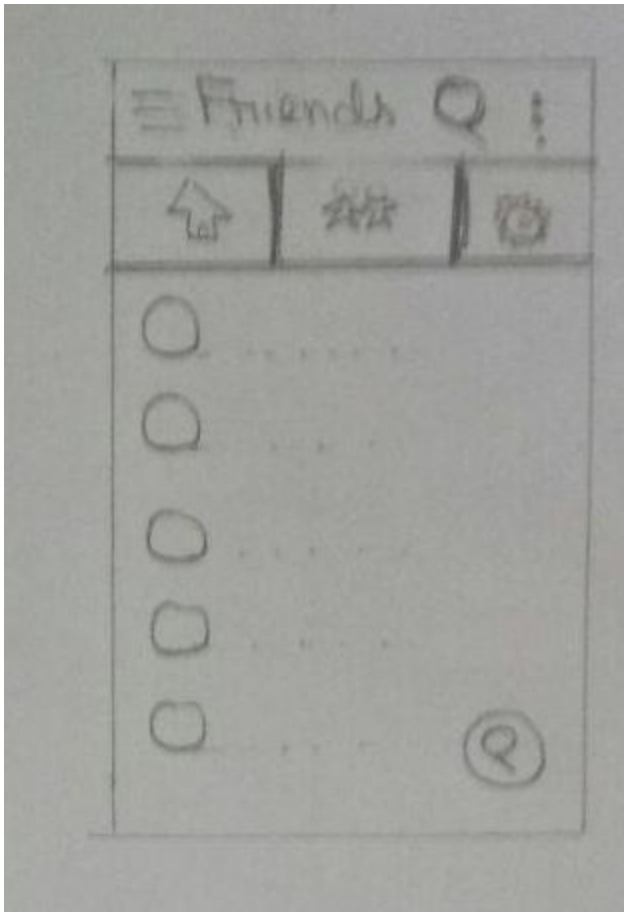
Tour Activity



This activity is one of the focal point of this application.This activity is created when a new Tour is created by the user or the user views his/her friends tour. The screen is divided into 2 parts. One consist of a map layout which displays the current location of the user or his/her friend. The second part of the screen has the conversations in which all the user's friends can have a chat.

Screen 6

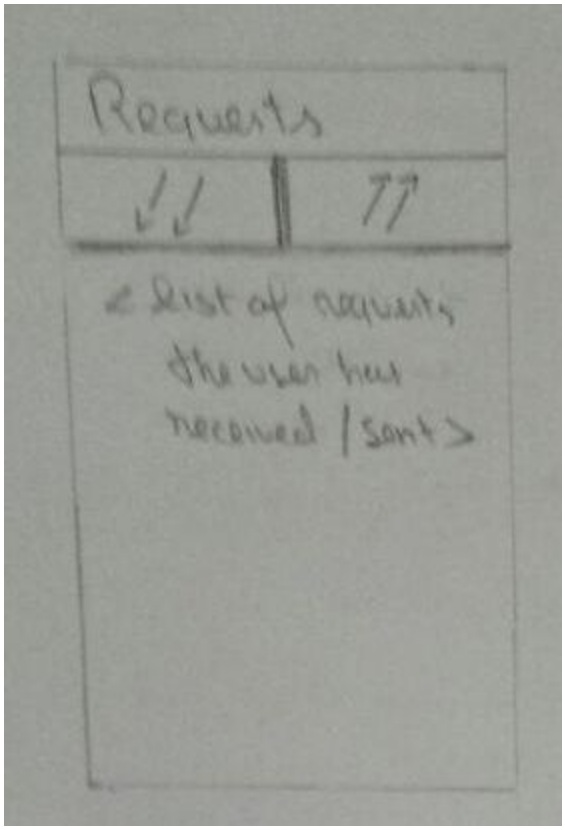
Friends



This activity contains a list of user's friends. On clicking the list item the user can view his/her friends profile.The fab button is used to find new friends. On pressing the search button a textview is focused on the toolbar by which the user can search other users by using their email id.The listview will be updated based on the result on the search. On clicking the search result list view the user's profile is shown and the user can pass a friend request using the button given.

Screen 7

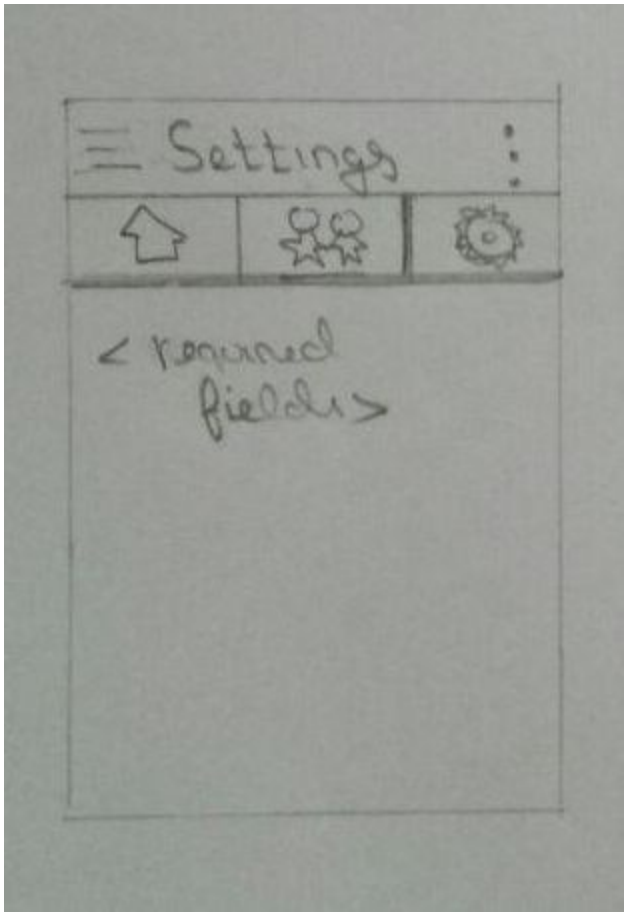
Requests Activity



This activity contains the list of user id’s the user has given request to and the list of id’s the user has got the request from.

Screen 8

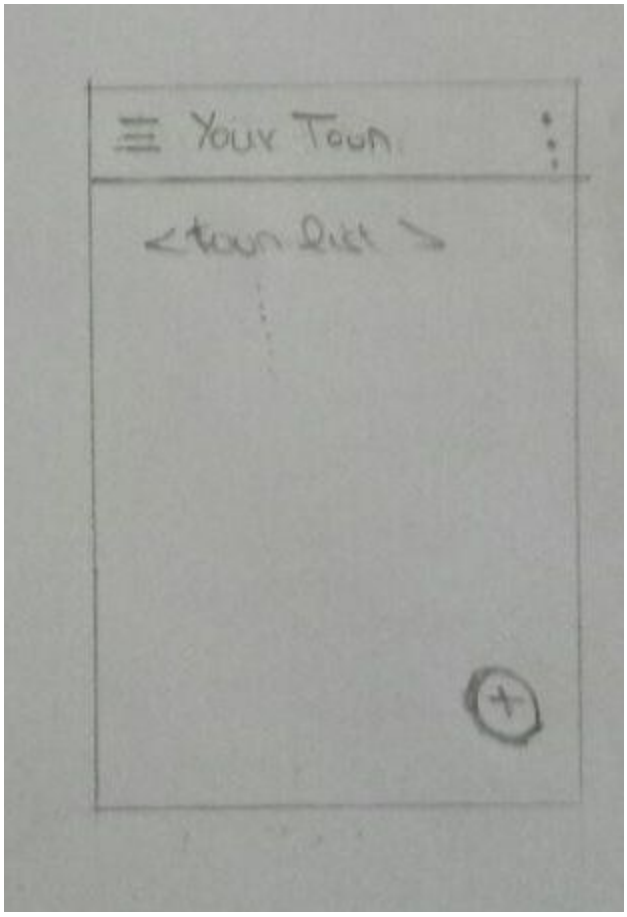
Settings



The user can set the application’s default parameters and his/her preferences through this activity.User’s profile information can also be modified through this activity.

Screen 9

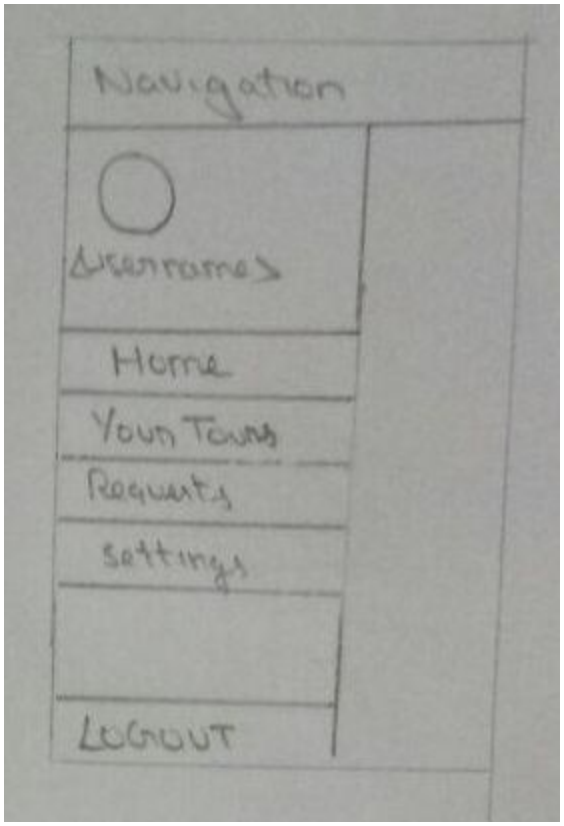
Your tour



This activity has a list of all tours the user have created.on clicking the list item the user can view his/her tour in the Tour activity with an additional status(live/completed).

Screen 10

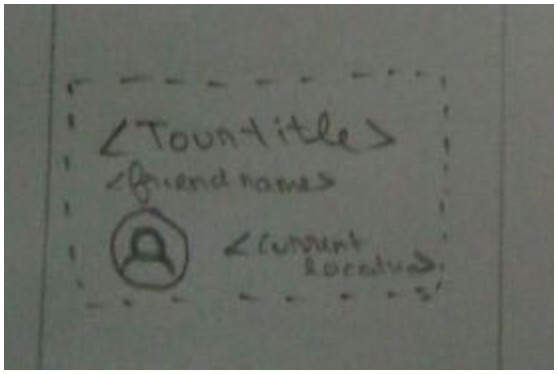
Navigation



This screen is of the navigation drawer which links all the pages/activities together.

Screen 11

Widget



Widget is very useful to convey all the information to the user effectively. The widget provides the description of latest tour by the friends of the user.

Key Considerations

How will your app handle data persistence?

All the data of this application will be stored using shared preferences or SQLlite databases.
For managing the friends list the Content provider for the contacts will be used.
Most part of the data will be synced with the Firebase real time database.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

Describe any libraries you'll be using and share your reasoning for including them.

- **com.github.ViksaaSkool:AwesomeSplash:v1.0.0**
  - **To implement a animated splash screen**
- **Firebase libraries**
  - **To handle firebase components.**
- **Libraries to handle google maps.**
- **Library to handle Firebase job dispathcer**
- Material Design libraries like **'com.android.support:design:25.1.0'**
  - **To handle material design concepts.**

Describe how you will implement Google Play Services.

The following is list of google play services used in the application

- Firebase real time database
- Firebase user authentication
- Location service

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

The first and foremost step will be to set-up all the components required to build the application. In this phase all the library file will be included in the gradle and synced.
Along with libraries all the resources like images, icons logo will be decided.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Splash screen with the help of library.
- Build UI for for Login page and Signup page. This activity will be connected to firebase authentication system through which user authentication will be carried out.
- Build the UI for HomeActivity.
    - This Activity will be a tabbed activity which has 3 tabs.They are
        - Home Tab
        - Friends Tab
        - Settings Tab
- Build the UI for New Tour activity. This activity will have all the fields to define a new Tour.
- Build UI for Tour Activity. This activity will include a map and a chat fragments which will be connected to firebase real-time database.
- Build the UI for Your Tour activity which will have the list of user's tour.
- Build UI for Friend request activity. This activity will handle all new Friend requests and the request the user have given to others with two fragments.
- Build UI for Settings Tab which may include all the user preferences.
- Build a navigation drawer common to all the pages.
- Build the widget.

Task 3: Implement Data handling components

As the application handles vast amount of data from internet major part of the will be stored using SQLlite and content providers. This phase will implement all the data storage techniques.

Task 4: Implementation of Job dispatcher

As the application uses more data transaction from firebase. A firebase job dispatcher is created in order to load all data in background.

Task 5: Implement Notification

This phase will be used to implement notification managers in order to push notification to users based on the data changes in the firebase.

Task 6: Error handling

Implement error handling mechanisms to avoid or catch any exceptions which may in the application in order to provide a smooth experience to the user.

Task 7: Implementation of common classes

All the redundant code will be summed up and a utility class will be created to make the code more reusable.

Task 8: Check all the code

Verify all the classes and implement test cases.

Task 9: Build the release version

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"