StudyBuddy

1. Introduction

StudyBuddy is an Android-based application that provides a platform for students to collaborate, create, search and discover groups based on their interests and location. It caters to all the specifications and needs of the students when searching for groups, filtering groups by subject, time and maximum group capacity. Groups can also be filtered based on spatial queries such as range and k nearest groups.It also provides an option to view these groups in different formats (Map View & List View) and add them to favorites. Last but not the least, users can create their own groups and request to join other groups.

1.1 Basic functions

Create a Group

1. Users can create a group with a group name.
2. Users can specify subject to be studied, topic, maximum capacity of group and start and end time.
3. This newly created group is plotted on the map for all active users.

View/Join/Delete groups

1. Users can view all active groups plotted on the map or view them in a list form.
2. Each group marker has an info window giving detailed information about the group.
3. Users can join group using the Join option.
4. Users can delete the groups created by them.

User profile & preferences

1. Each user has a profile and signs up for the application using username, email and password.
2. Users can add Study groups to favorites section.
3. Users can view all groups created by them in a listview.
4. Users can view all joined groups in a listview.

Search & Filter Groups

1. Users can search for groups using 2 Geospatial and 3 Non GeoSpatial filters.
2. GeoSpatial features include searching K nearest groups and searching groups in a user defined range.
3. Non GeoSpatial features include filtering groups based on time, maximum capacity and subjects.
4. User can apply multiple filters and gets the intersection of all filter queries as the result.

1.2 New Features

Dynamically Create Groups Based on Max Capacity

Users can create Study Groups on the go and specify the maximum no. of group members they wish to have in the group. When the group reaches the maximum capacity, the group is not plotted on the map anymore and is thus not visible to the active users.

Search K nearest groups to the user

Users can filter K nearest groups to them based on their current location. The nearest groups are fetched based on the Euclidean distance from the user.

Search groups In a Circular range around the user

Users can filter groups based on a circular range by setting the radius using the seekbar. The circle drawn on the map is around the user location as center.

Favourites & Preference Section For Each user

Users can add groups to their favorites and revisit them anytime. Users can also set their preference and view groups related to their profile.

Apply Multiple Filters Simultaneously

Users can filter K nearest groups to them based on their current location. The nearest groups are fetched based on the Euclidean distance from the user.

Join groups

Users can join any group that he is interested in by clicking on the info icon on the marker info popup and then clicking on the “Join” button.

2. Application

Our application is designed using the client-server architecture model.The client is the Android application.Server is RESTful web service developed using JAVA Spring MVC framework. For storing the group and other information we have used Postgre DB with its PostGIS spatial extension for the spatial related features. We are using an AWS RDS instance to store the data on the cloud.

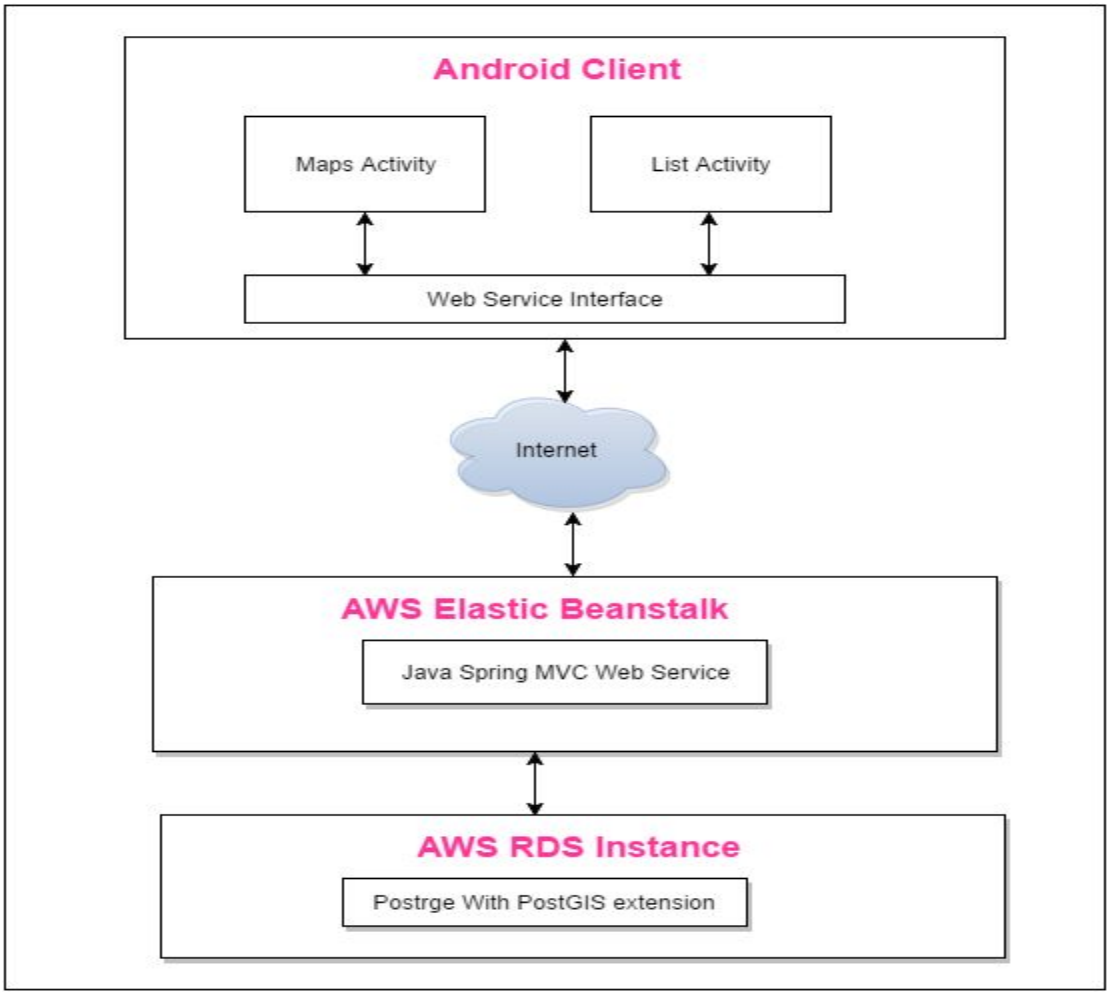

Fig 1 . Architecture Diagram

2.1 Web Service/Application Server

We have strictly followed a Client-Server model to allow decoupling the application and making the development fast and scalable.The RESTful Web Api's are developed using Java Spring MVC framework.

Below is the contract for all the api's we developed for the features needed in our application.Url stands for the the cloud url(AWS) where the api's have been deployed.We have grouped the api's into high level categories below.We are providing the HTTP method for the api along with the Url. The api names are intuitive so not providing the explanation for all of them.

2.1.1 User Management/Login/SignUp

GET http://{url}/getAllUsers

POST http://{url}/signUp

{ "userId": {user_name},"name": {name},"password" : {password}}

GET http://{url}/getUser/{user_id}

DELETE http://{url}/deleteUser/{user_id}

2.1.2 Get Groups,Create/Delete/Join Group

GET http://{url}/getAllGroups

POST http://{url}/createGroup (Not providing the create json as its pretty big).

DELETE http://{url}/deleteGroup/5

GET http://{url}/joinGroup?groupId=2&userId={}

GET http://{url}/getJoinedGroups/{user_id}

2.1.2 Search Groups based on both spatial and non-spatial features

GET http://{url}/searchGroups?maxCapacity=10&subjectId=2

GET http://{url}/searchGroups?startTimestamp={}&endTimestamp={}

GET http://{url}/searchGroups?latitude={}&longitude={}&k={NN_value} - **KNN**

GET http://{url}/searchGroups?latitude={}&longitude={}&range={range} - **RANGE**

2.2 Database

As mentioned above we are using an Amazon RDS DB instance to deploy the PostGRE database.For spatial queries we are using the PostGIS extension.

Below is the Entity-Relation diagram for our database schema.The primary tables are User, Group, Group_Members and Subject.

User: This contains all the user details such as their Name,Username(userId).The user password is stored as one way hash to maintain the privacy of the user credentials.

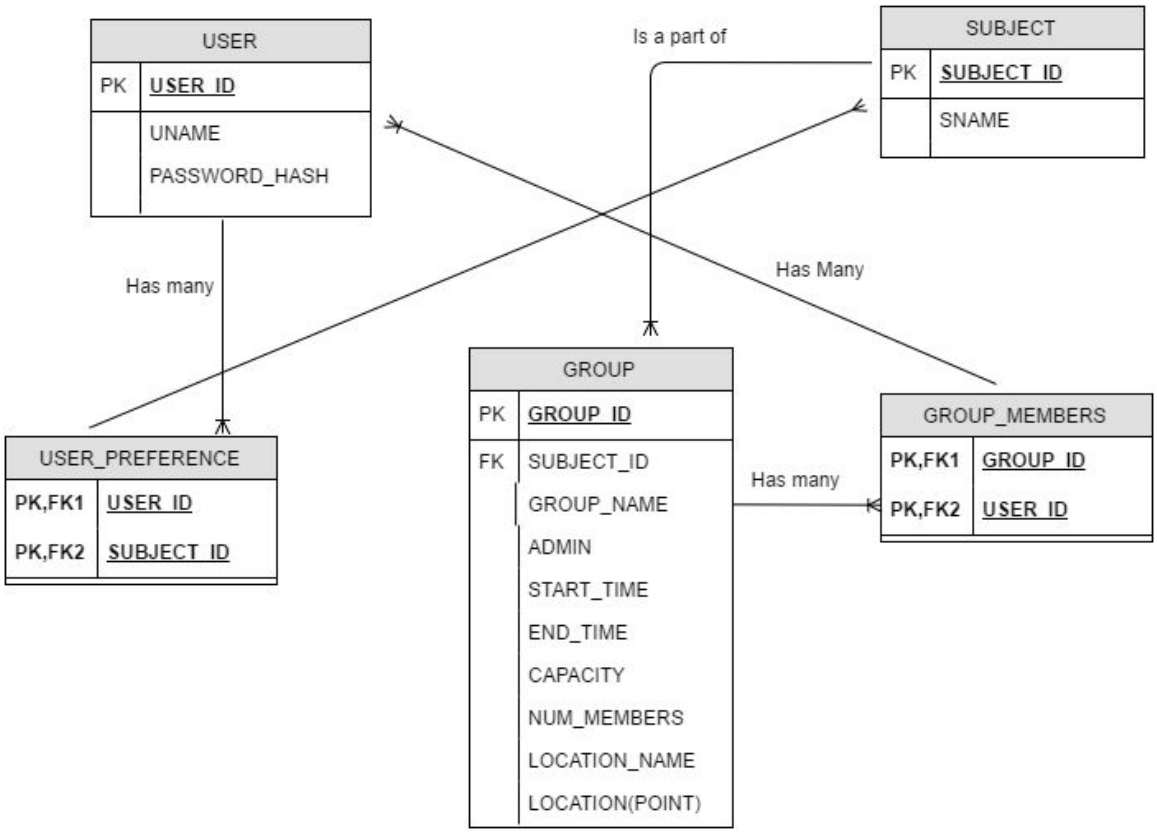

Fig 2 . Database Schema

Group: This table is the main table which contains all the information related to all the existing study groups like group name, its latitude and longitude, group admin, start time and end time, max capacity and current number of members.

Group Members: This table has many to many mapping with the User table and the Group table.Basically it contains the information about which user is a part of which group.

Subject: This table consists the list of all the subjects for which the groups can be created. It has a subject Id, subject name and description.

2.3 Android Client

The Android client consists of list of the activities which are responsible for the interaction with the user. Below are the main functionalities of the client with screenshots

Start Screen

The start screen of the application has two options - Login and Sign Up

Fig 3 . Start screen

SignUp/Login Screen

The user needs to enter his or her details like username,email id and password to sign up for the application.SignUp activity performs basic checks on the entered details and sends the information to the server.

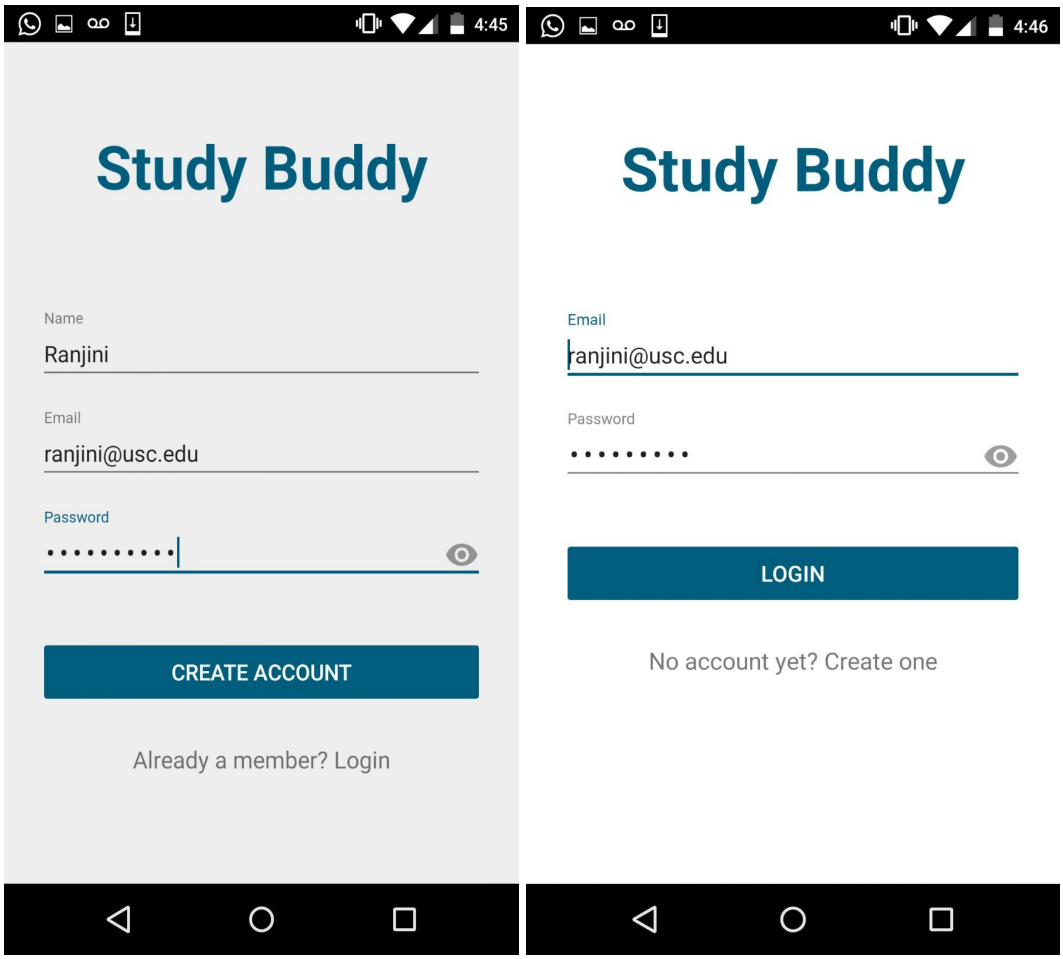

Fig 4 . Sign up and Login screen

The user will then be able to login into the application with the credentials provided during sign up.

Landing Page

The Landing page of the applications looks like the below screenshot

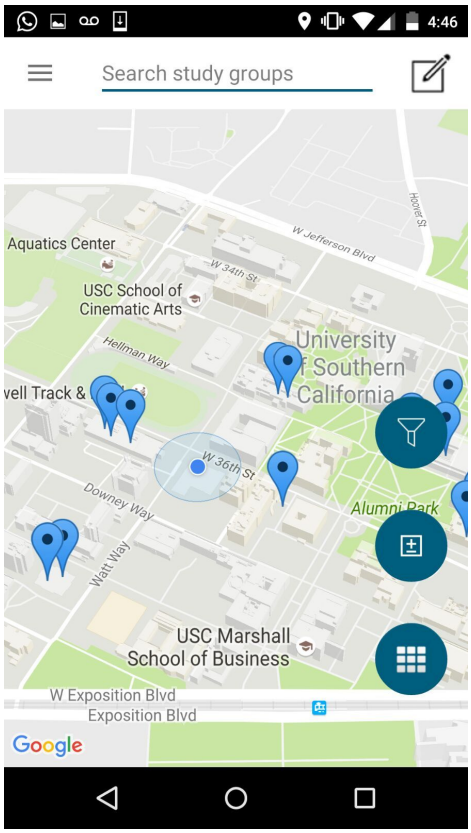

Fig 5 . Landing screen

The landing page has the all the study groups around three miles of vicinity and within two weeks of time .The search bar is used to filter groups based on the subject.The button to the right of search bar is the create group icon used to create study group. The icon to the left of the search bar opens the navigation drawer for user's profile.The three floating buttons on the right bottom of the screen are used to perform filter based on range, filter based on KNN,Time and Capacity and provide a list view of the groups.

List View

The list view provides detailed description of all the groups available on the Map

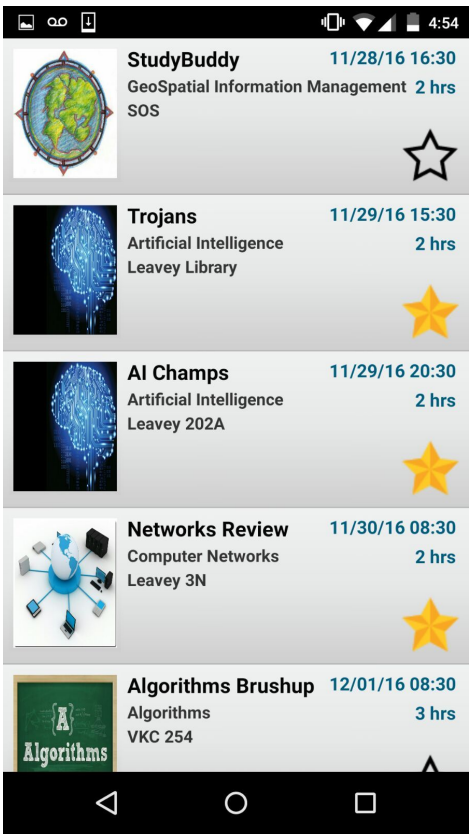

Fig 6 . List View

The Study group information like group name,subject,start time of study, duration and location is provided.The list view is sorted based on the start time.The users can mark the groups of their interest as favorites which can be viewed in the favorites tab of user's profile.

Search

The app provides a very rich search functionality.The user can search for the interested groups by filtering them based on subject,range,nearest group option, time and capacity.

RANGE FILTER

The user can search for the groups within a particular range by clicking on the filter icon in the bottom right corner of the landing page and by moving the slider. The Map gets updated with the filtered study groups.

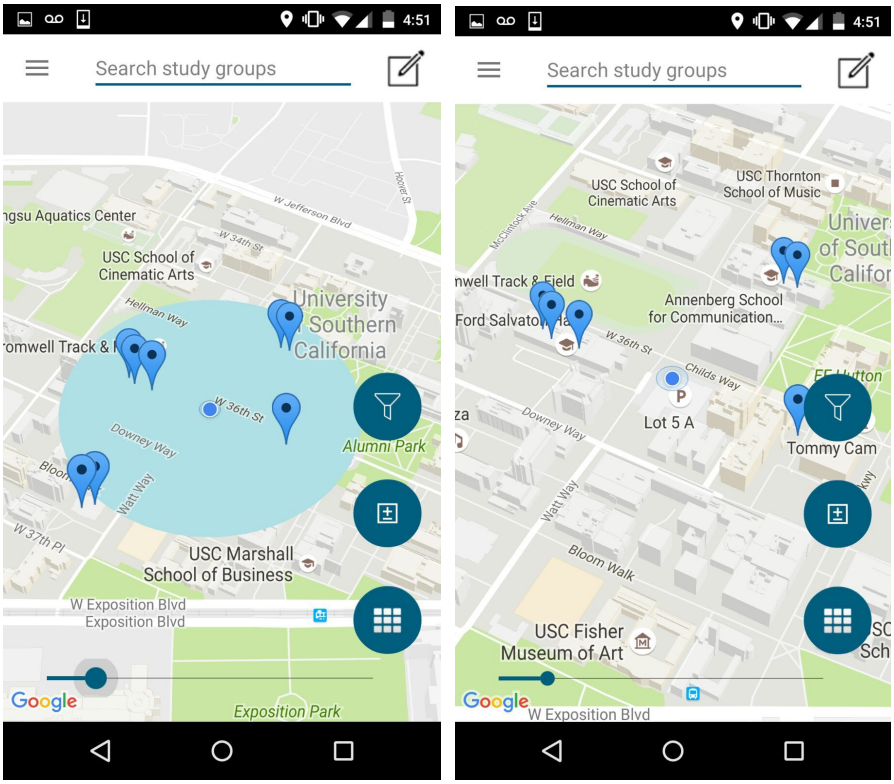

Fig 7 . Range Filter

SUBJECT SEARCH

The search bar on the top of the landing page is used to search the groups based on the subject.

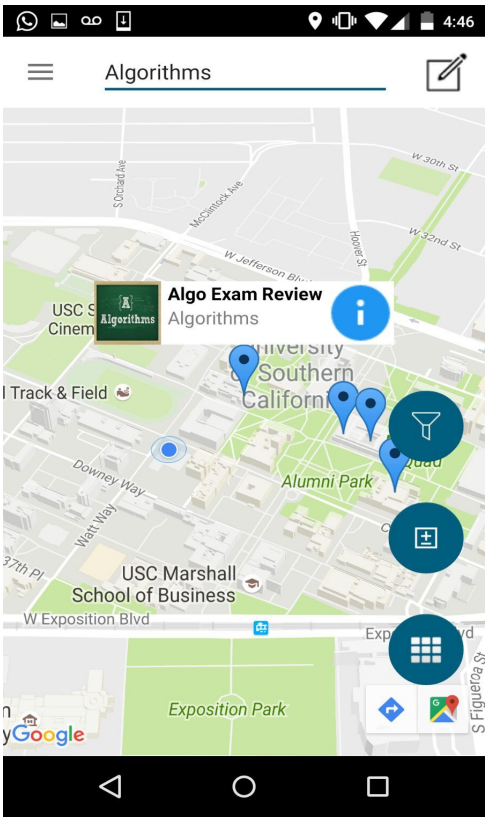

Fig 8 . Subject based search

OTHER FILTERS(KNN,TIME,CAPACITY)

The second floating button on the bottom of the landing page provides option to filter based on start time and end time, maximum group capacity and K-Nearest Neighbours.

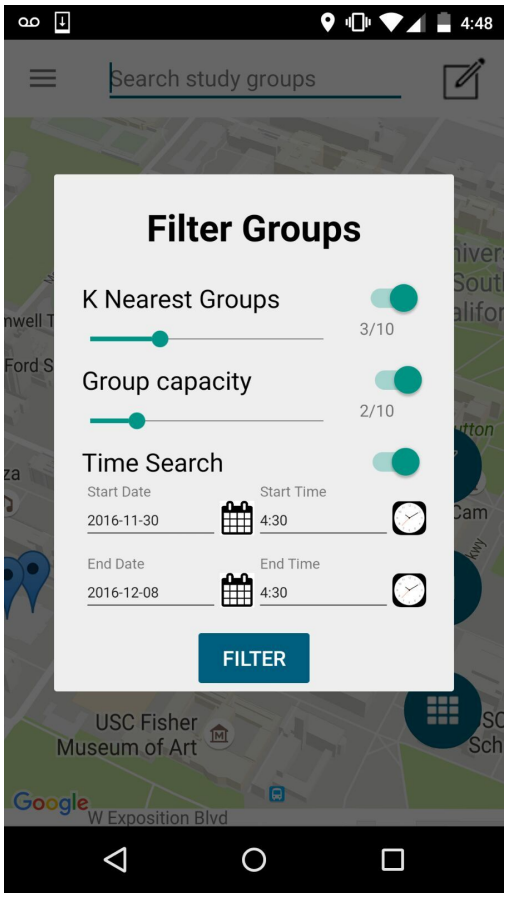

Fig 9 . KNN,Time and Capacity filter

Create Group

Create Group screen helps the user to create the study group.The user provides information like group name,subject,topic details,location details,capacity and start and end time while creating the group.Once the create group button is clicked the information is sent to the server and map view of the app is updated with the newly created group.

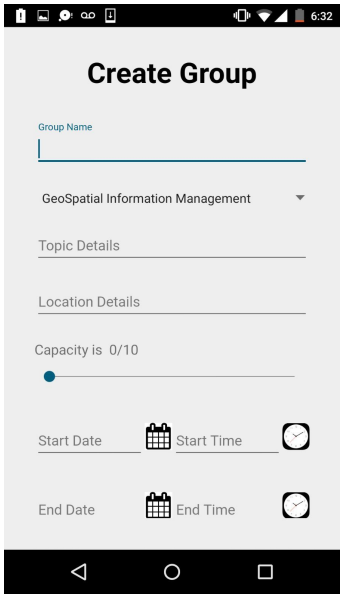

Fig 10 . Create Group screen

Profile information

The side navigation drawer has profile information about the user. The user will be able to view the groups he/she has joined by clicking on the joined groups option.User can also view the favourite groups by clicking on the favorites option.The other most important option is the groups owned by the user.By selecting this option user will be able to view the groups created by him/her and delete the groups which have expired.

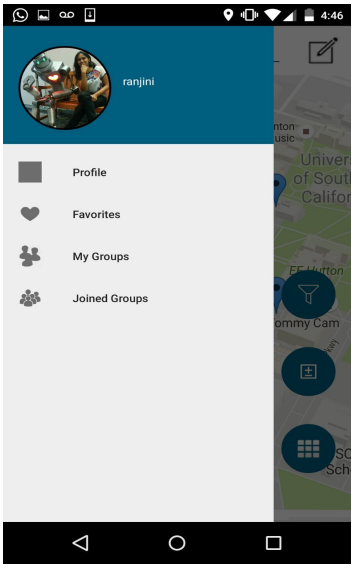

Fig 11 . Profile view

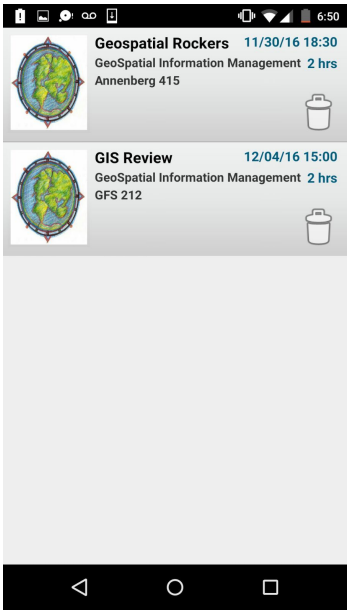

Fig 12 . List view of user owned groups

Join Group

When the user clicks on the marker they would be able to view the short description of the group . when the user can click on the info icon to view detailed description. In the detailed description info window he/she will be able to join the group to become part of it.

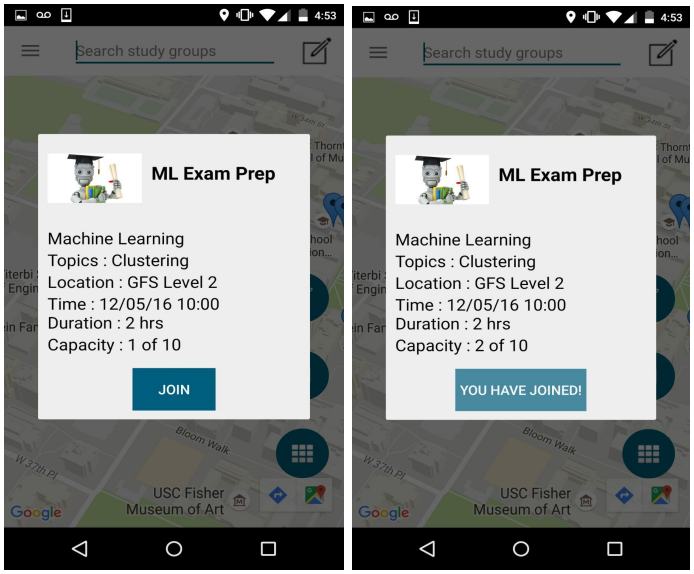

Fig 13 . Join Group Screens

3. Installation and configuration

We have three main components of our application - Android application(Client), Web Service(Server) and Database.

The source code and related scripts are placed inside the src directory of the submitted project materials.We are also providing the Github links so that the projects can be directly cloned as well.

Below are the instructions for deploying each component to get the application up and running.

3.1 Android Application(Client)

https://github.com/nishant4498/StudyBuddyApp

1. The apk file for the app can be found in the below location with the name Study-BuddyApp.apk

   src\StudyBuddyApp\StudyBuddyApp\MyApplication\builds.

2. The apk file can be installed using adb tools with the command

   ./adb install <apk file path>

3. To build your own apk from the source code you need to get the google api key using https://developers.google.com/maps/documentation/android-api/signup so that the map shows up after building the application.
4. Please make sure that the location and wifi settings are enabled for the app to work properly.

3.2 Database installation

1. The database and table creation scripts are located in the directory src/Scripts.Run these scripts on PostGRE server. We also need to install the extension PostGIS for the spatial queries.
2. Note the database instance name(localhost if installed locally), portnumber, username and password.These values will be needed in 3.3 Step 2 to connect the web service to the database.

3.3 Web Service(Server)

https://github.com/nishant4498/StudyBuddyService

1. Set up Java(7 or higher) with Spring MVC and maven on the machine.Also install Tomcat server(Tomcat 7 or higher) for deploying the application.
2. In the file below update the hostname of the database instance, port, username and password.

src\StudyBuddyService\StudyBuddyWebService\src\main\webapp\WEB-INF\springrest-servlet.xml

3. Copy the entire folder src\StudyBuddyService\StudyBuddyWebService inside the tomcat home directory and start the tomcat server.
4. If the application is successfully installed the below page(home page) should be accessible:

   http://{host_name}:{port_number}/StudyBuddyWebService/