# Nights'On

## 1. Introduction

Nights'on is an Android-based mobile application that enables users to find the parties nearby. The app has the ability to discover the parties around you as you move or navigate on maps. It gives the user the ability to look at the information of the party like date, time, cover charge, number of people attending etc. It has two sides to it, one where the host updates on the App about his party details and the the other side the user can see the details. The best part of this app is, it is scalable. It can be used to incorporate events, exhibitions etc., in future.

### 1.1 Basic functions

**Client**

### View nearby parties

- Allows users to see all the nearby parties on the map
- Allows users to see their current location on the map
- Allows users to move and reset their location to current location on the map
- All the parties viewable on the map are clickable
- Each party on being clicked will show the details of the event
- The details include date, time and precise location of the event
- The user also has the ability to RSVP the event he clicked to show his interest

### Add parties

- Every user has an ability to host an event- a Party
- Each event has the location field, time, data of the event
- The user can navigate around the map to point the location where the party is being hosted
- A user can also attach a photo of the event

### View/ Edit parties:

- Users can view and edit parties
- Users can search parties by moving the location on the map

## Profile

- Every user has to register in order to do any activities in the map
- Each user will be authenticated and a session will be created for all the API calls
- Each user can update the profile photo
- Each user can change the password

## Server

- Accepts requests from a client to add an event
- Accepts requests from a client to delete an event
- Accepts requests from a client to RSVP
- Accepts requests from a client to remove the RSVP
- Accepts requests from client to query the event from the database
- The server will store data on PostgreSQL database

## 1.2 New Ideas/features

### 1.2.1 Searching for nearby Parties

Users who cannot find the parties around instantly, this app performs the range queries and fetches the parties around you by the radius mentioned before.

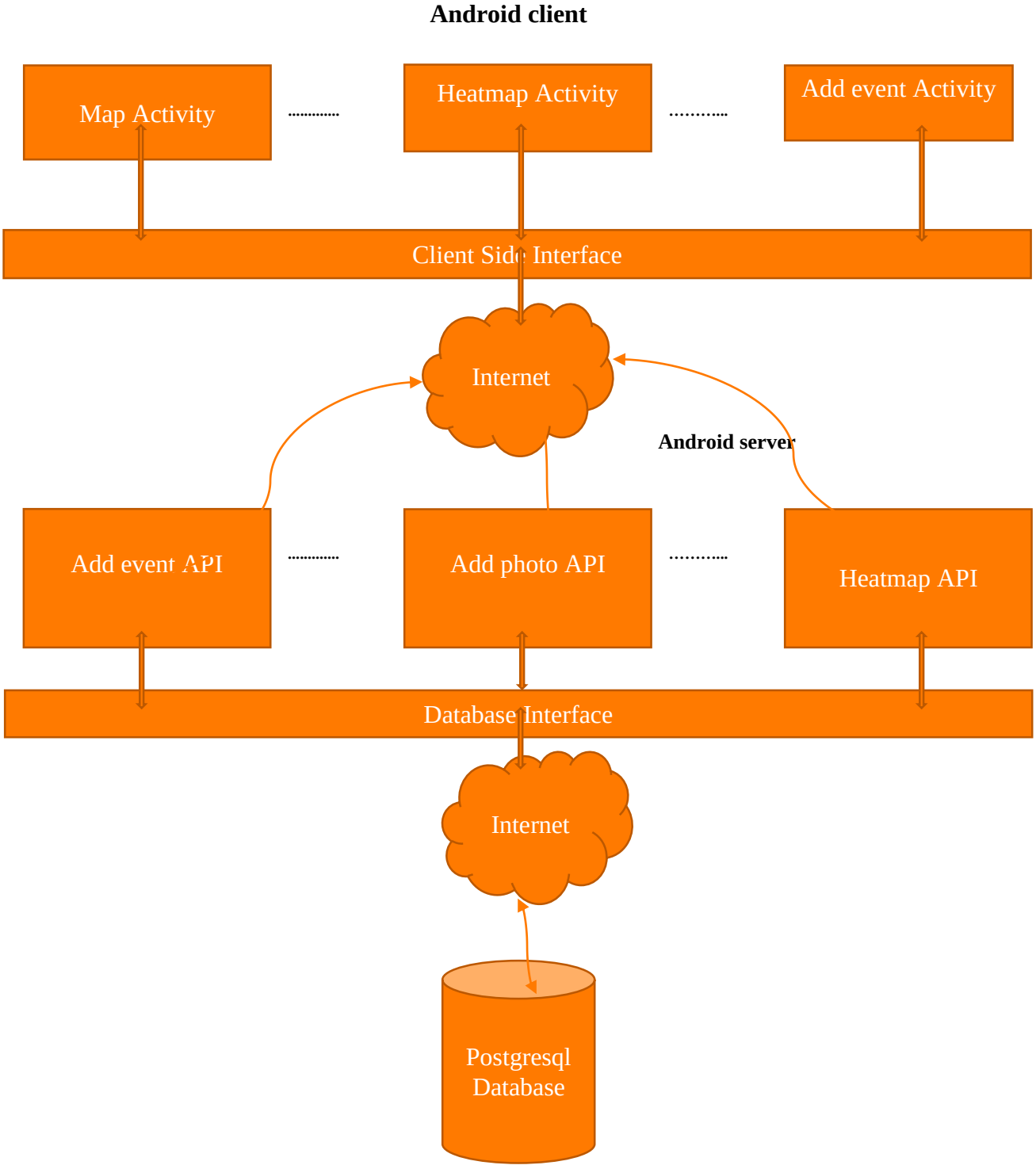
### 1.2.2 RSVP'ing the interested parties

The users also have the ability to RSVP the parties which they are interested in and this way the host can know the number of users/people attending the party. They also definitely have the chance to unRSVP the events they have already RSVP'd to remove themselves from the interested people

### 1.2.3 Heat map Generation

The users as soon as they land on the map view have a new feature integrated which can help them identify the popular events around by showing the heat map spread around the events. The heat map is generated on the basis of the number of people interested in the party i.e., the number of the people RSVP'd to the event

## 2. Application



## 2.1 Server

### 2.1.1 Events API:

- Get Events:
  Method: GET
  URL: /events
  Fields: id
- Create Events:
  Method: PUT
  URL: /events
  Fields: name, latitude, longitude, start_time, end_time, photo_url, description
- Update Events:
  Method: POST
  URL: /events
  Fields: id, name, latitude, longitude, start_time, end_time, photo_url, description
- Delete Event:
  Method: DELETE
  URL: /events
  Fields: id

### 2.1.2 Users API:

- Get Users:
  Method: GET
  URL: /users
  Fields: id
- Update Users:
  Method: POST
  URL: /users
  Fields: id, first_name, last_name, email, photo_url, phone, location, location_recorded_at
- Delete Users:
  Method: DELETE
  URL: /users
  Fields: id

### 2.1.3 Signup API:

Method: PUT
URL: /signup
Fields: first_name, last_name, email, photo_url, password, phone, latitude, longitude, location_recorded_at

### 2.1.4 Login API:

Method: POST
URL: /login
Fields: email, password

### 2.1.5 Logout API:

Method: POST
URL: /logout
Fields: all_session (If set to 'true' all session will be kicked out otherwise, only the current session will be kicked out)

### 2.1.6 RSVP:

- RSVP Event:
  Method: POST
  URL: /subscribe
  Fields: event_id
- UN-RSVP Event:
  Method: PUT
  URL: /subscribe
  Fields: event_id

### 2.1.7 Upload Media API:

Method: POST
URL: /upload
Fields: file_name => file object that needs to be uploaded

### 2.1.8 Get Media API:

Method: GET
URL: /static/<photo.extension>
Fields: None

### 2.1.9 Search API:

Method: GET
URL: /search
Fields: latitude, longitude, radius (in meters)

### 2.1.10 Update Password API:

Method: POST
URL: /updatepassword
Fields: old_password, new_password

### 2.1.11 Authenticate API:

Method: GET
URL: /tokenauthenticate (returns HTTP_ACCEPTED if token is valid otherwise, HTTP_UNAUTHORIZED)
Fields: None

### 2.1.12 Events-Users API:

- Get All Users subscribed to an event
  Method: GET
  URL: /eventusers
  Fields: event_id
- Get All Events Owned by the current user
  Method: GET
  URL: /eventsownedbyuser
  Fields: None
- Get All Users subscribed to an event
  Method: GET
  URL: /userevents
  Fields: None

### 2.1.13 User-Location update API:

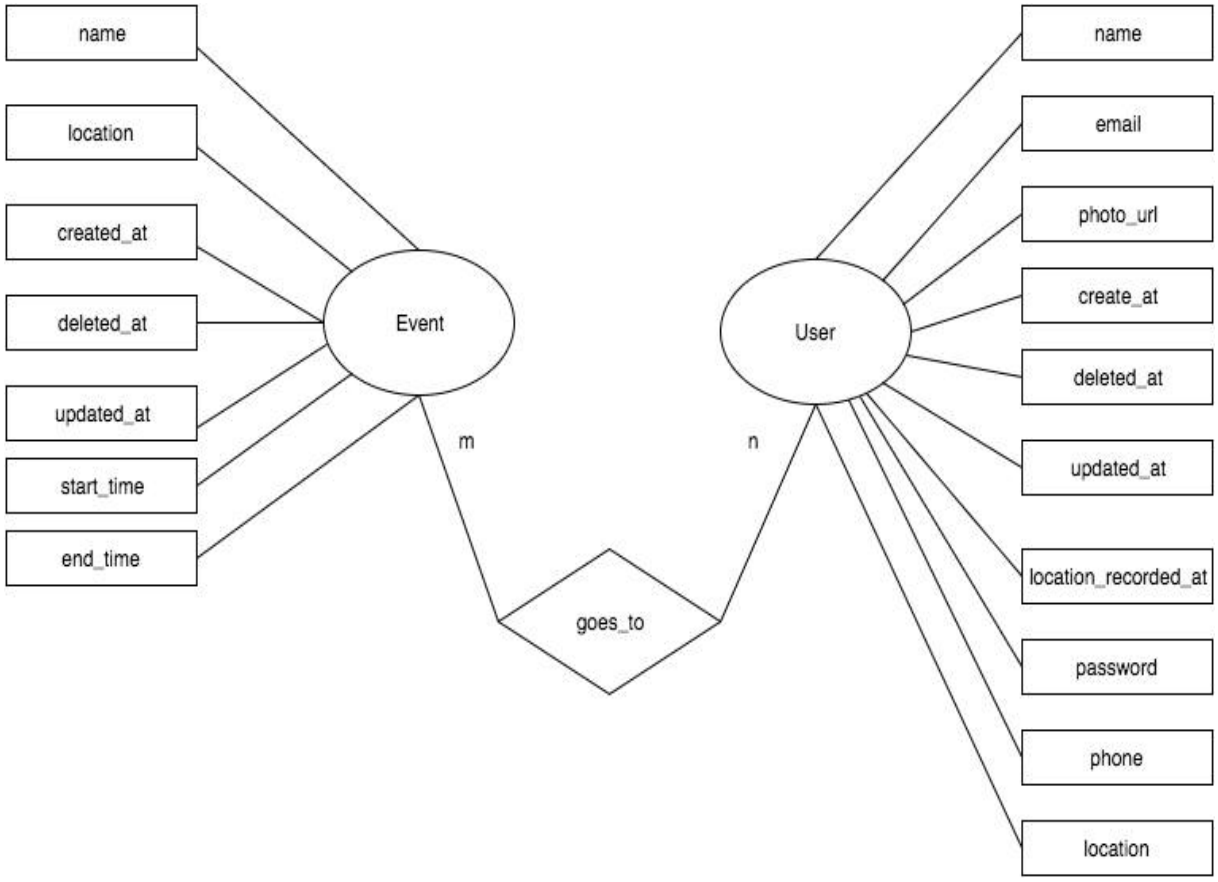
Method: POST
URL: /updatelocation
Fields: latitude, longitude

## 2.2 Database

An entity-relationship diagram for the database component is shown in the figure below. There are Two major entities: Event and User. Event and user are related through a relationship entity goes to Each of the event and the user have certain attributes which have been mentioned in the diagram.

The actual database used is PostgreSQL as it has spatial entity evolved. It has two major tables which are the entities in the ER diagram.

# CSCI587 Project Final Report



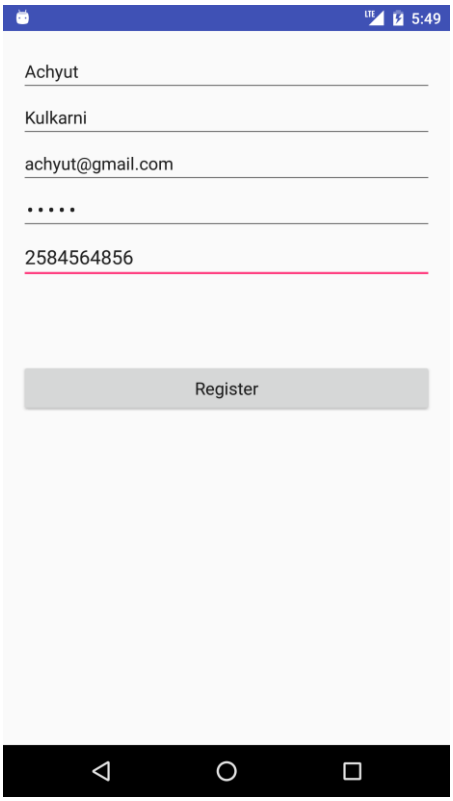
## 2.3 Android client

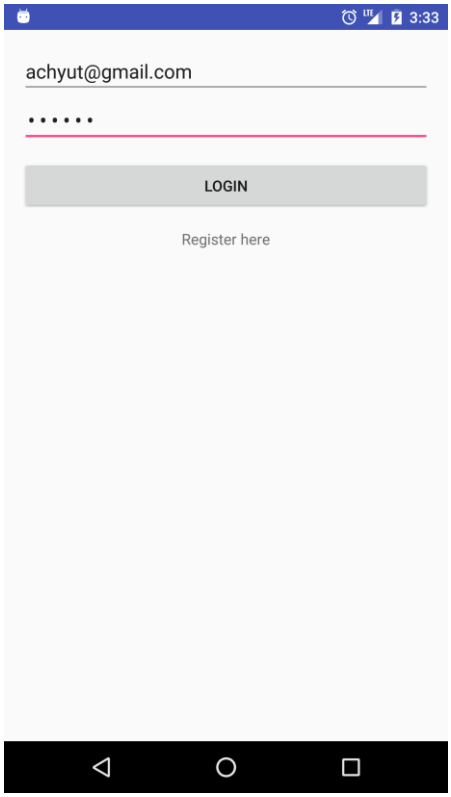
The android client interacts with the user via set of activities. Below is the set of some of the important activities and Fragments in terms of user interaction.

### 2.3.1 Representative Activities and Fragments

#### 2.3.1.1 LoginActivity

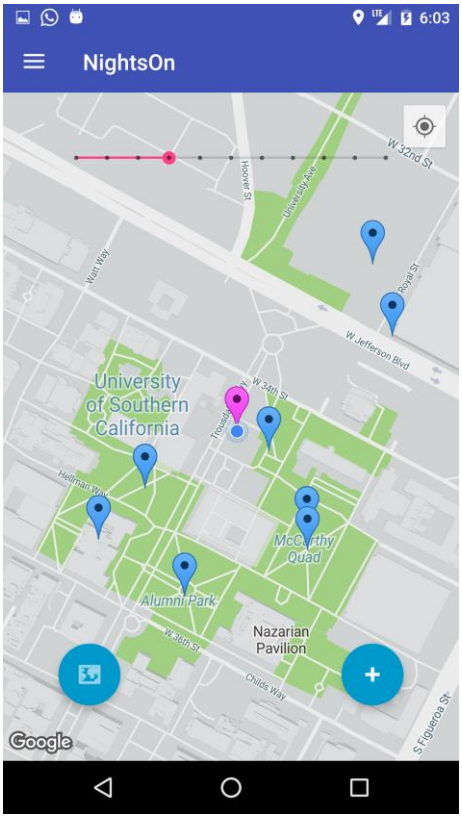
Login activity provides an authenticated login session.




#### 2.3.1.2 MapActivity

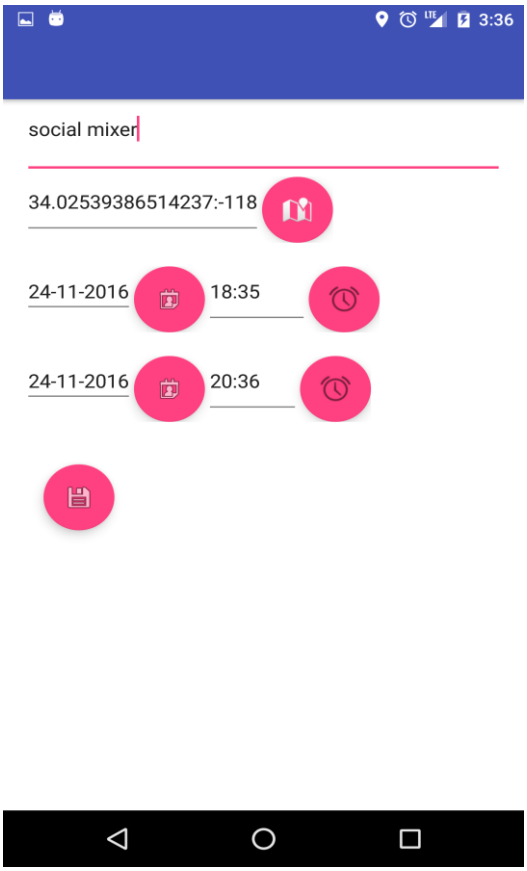
This activity is loaded as the homeview or the homepage of the activity. This activity is used to load all the fragments like Map fragment, Profile Fragment, navigation Fragment etc. It shows all the events that the user can look on the map. It has floating buttons to add an event for the user and to toggle between heat map and normal map mode.

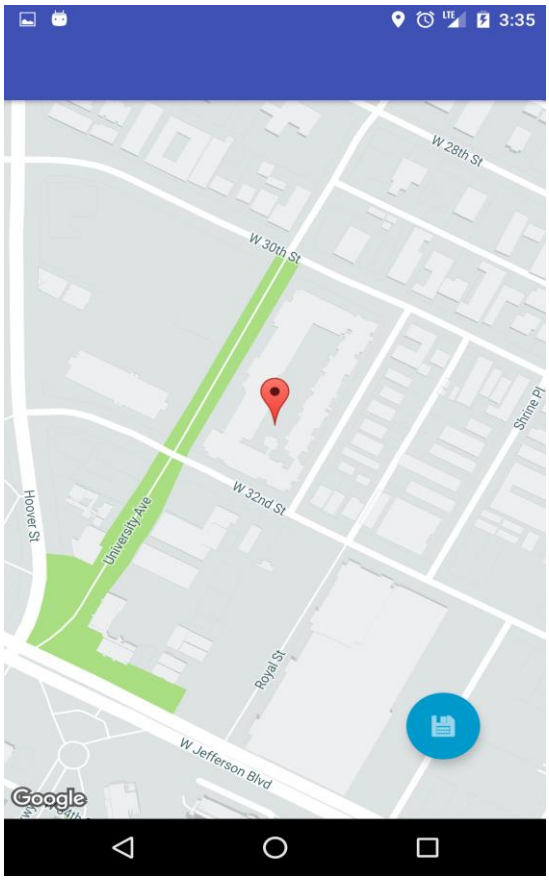


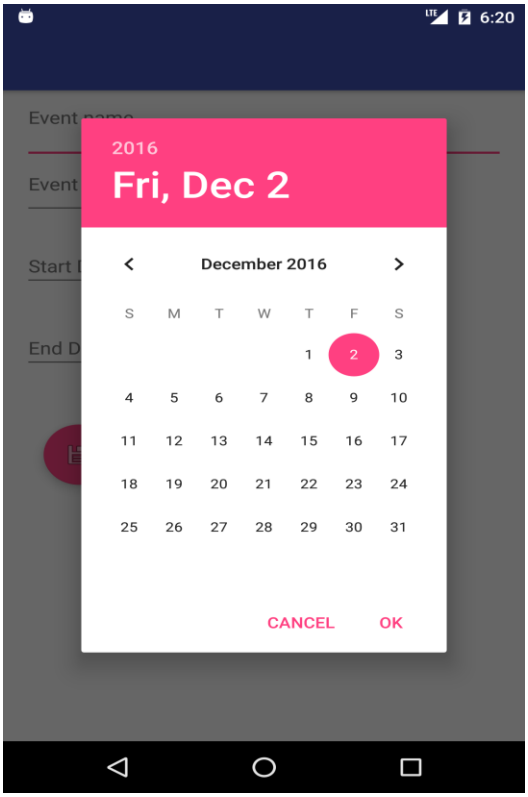

### 2.3.1.3 AddEventActivity

This activity is loaded as soon as the user clicks on the floating button on the Mapfragment of the MapActivity. This activity has many fields on filling of which will create a new event.

# CSCI587 Project Final Report

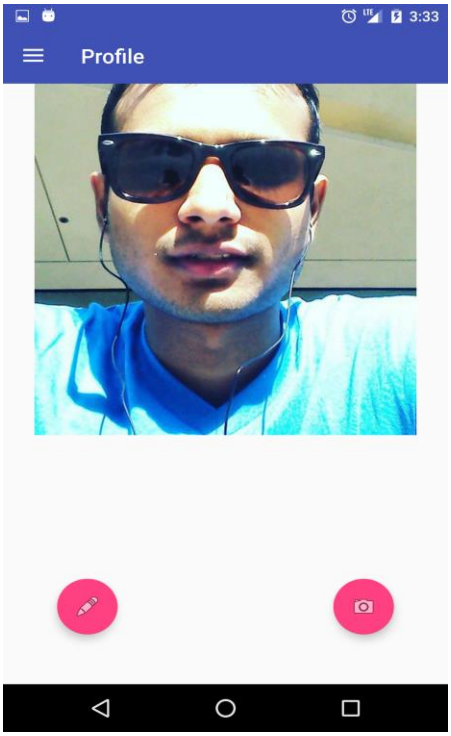


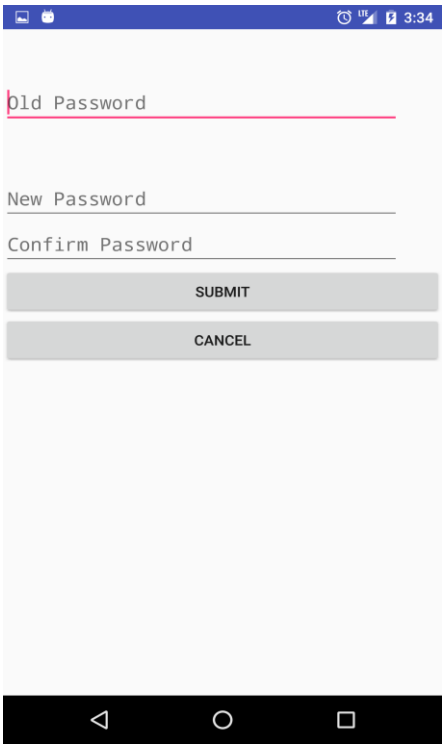






#### 2.3.1.4 ProfileFragment

This Fragment allows the user to view his profile details and update the profile information. It has change password fragment and change photo fragment which update the user's new password and photo at the backend and can use the updated password from then.

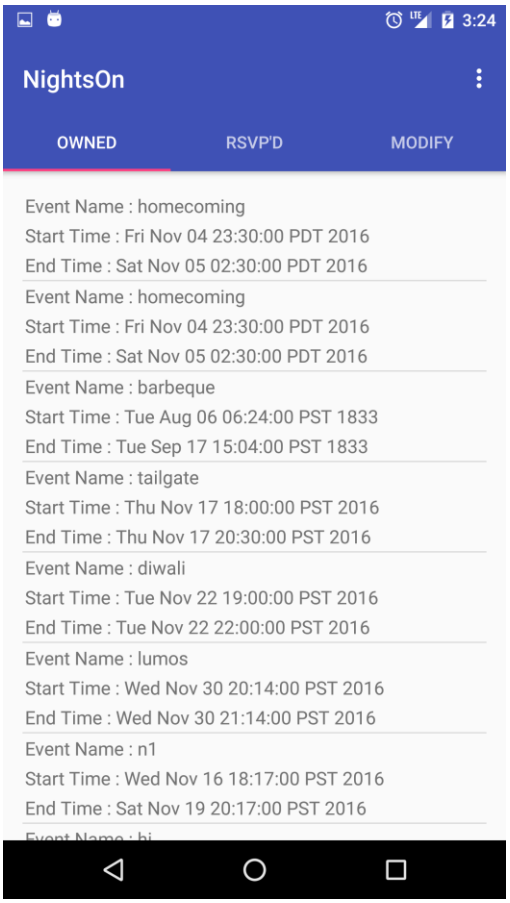


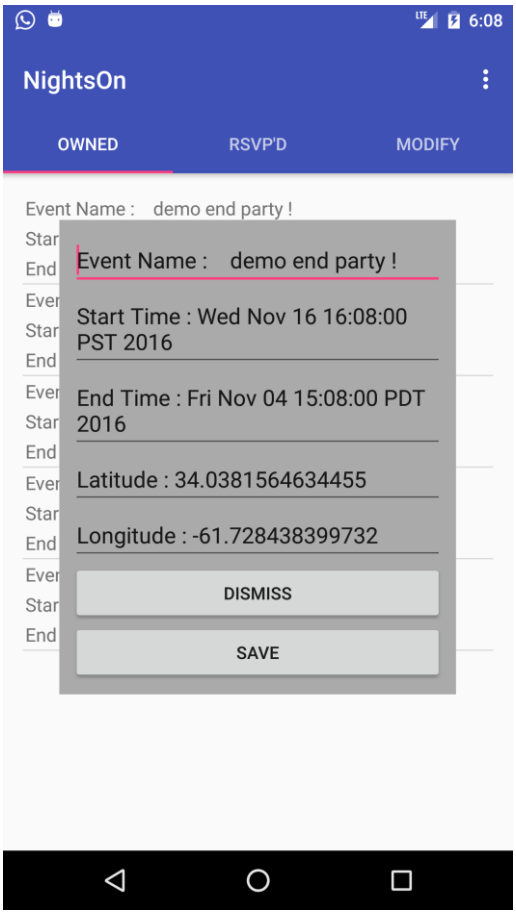



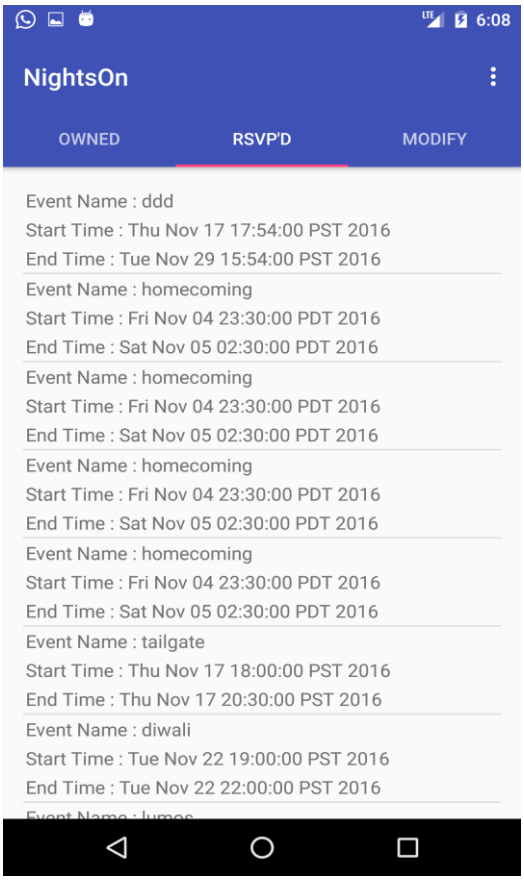

### 2.3.1.5 EventsFragment

This is one of the important fragments which the user has access to . This fragment mainly contains the information about the events which he has RSVP'd to , the events which he has created too. It is this fragment which gives him the ability to control over the parties he created and RSVP'd. From this fragment he can update or modify information of the parties which he owns(hosts) only and cannot modify the events which are created by other people.

# CSCI587 Project Final Report

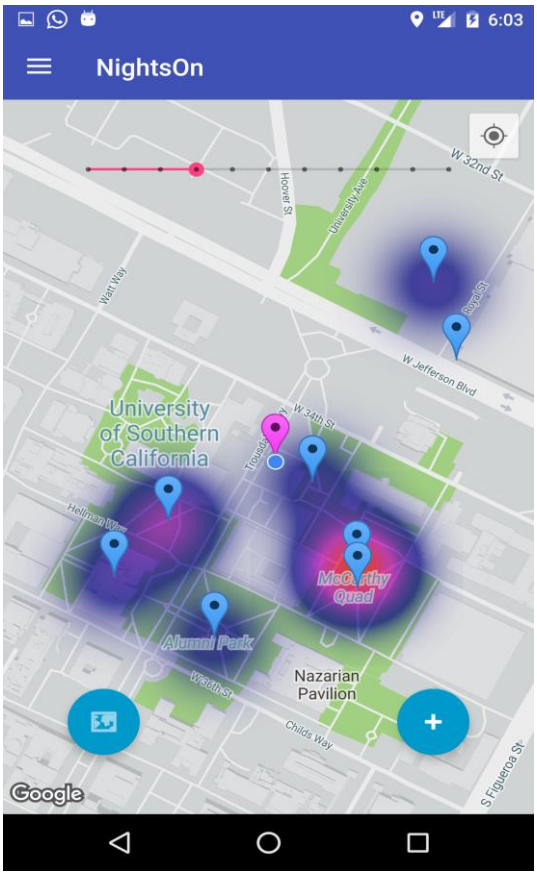




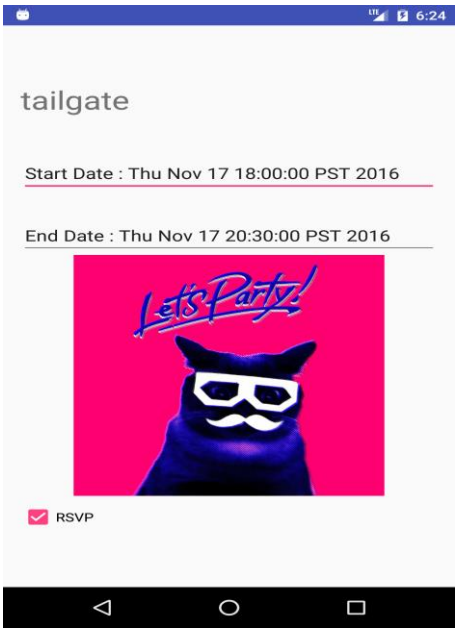

### 2.3.1.6 HeatMapFragment

This is a new feature incorporated for making the app more User friendly. Here on clicking on the toggle button of the MapFragment of the MapActivity, this fragment comes into foray which makes API calls to generate a heatmap overlay on the existing map



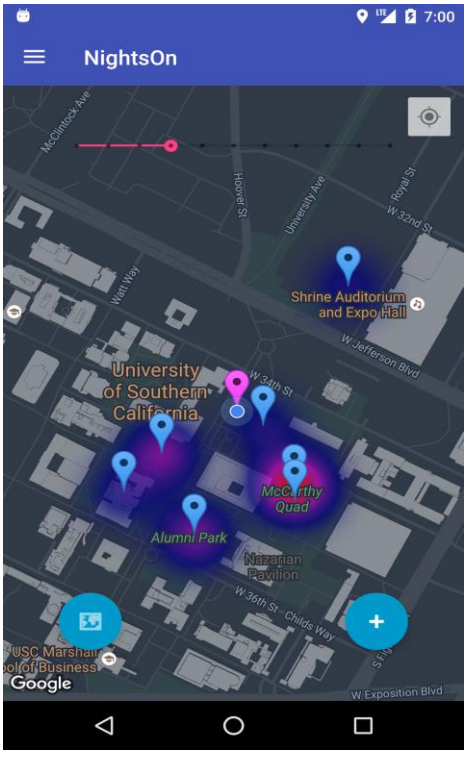

### 2.3.1.7 EventDescriptionActivity

This is the activity which is opened on clicking or tapping the markers on the map inside the MapActivity. On opening this activity , the user will be able to see the description of the event he clicked on. He can see the location, time and date of the event . Also this activity provides the user an ability to RSVP or unRSVP an event.




### 2.3.1.8 Night Mode Map view

This feature toggles between normal map view and night mode map view based on the time of the day. The night mode map gets loaded after 6:00 pm.



# 3. Installation and Configuration

The application consists of 3 main components. Client, Server and Database. Below are instructions on installing, configuring and deploy each component to get application working.

## 3.1 Database Installation

- Install postgres using on MAC (DEV mode)

```
brew install postgres => to install postgres
brew install postgis => to install spatial extension for postgres
```

- Install postgres on Ubuntu (EC2 server – production mode)

```
sudo apt-get update
sudo apt-get install -y postgresql postgresql-contrib
```

- Create a user and Database for our App

```
sudo -u postgres createuser -P vswamy
sudo -u postgres createdb -O nightson vswamy
```

- Installing Postgis is not enough to use spatial features. It needs to be enabled for each database where spatial features are used.

```
-- Enable PostGIS (includes raster)
CREATE EXTENSION postgis;
-- Enable Topology
CREATE EXTENSION postgis_topology;
-- Enable PostGIS Advanced 3D
-- and other geoprocessing algorithms
-- sfcgal not available with all distributions
CREATE EXTENSION postgis_sfcgal;
-- fuzzy matching needed for Tiger
CREATE EXTENSION fuzzystrmatch;
-- rule based standardizer
CREATE EXTENSION address_standardizer;
-- example rule data set
CREATE EXTENSION address_standardizer_data_us;
-- Enable US Tiger Geocoder
CREATE EXTENSION postgis_tiger_geocoder;
```

## 3.2 Server Installation

- Tornado is a python asynchronous web framework and networking library which runs using a single threaded non-blocking IOLoop and can scale to thousands of open connections making it ideal for long polling, web sockets, chat applications and applications which require long live connections to the server.
- Tornado comes with its own webserver. Installing Nginx/Apache is not required.

- Install virtualenv to create a fresh virtual environment for every python project that will be run. The server will have its own copies of dependencies and will not collide with system python libraries.
- Use make program which is available by default on MAC/Ubuntu to build the project and launch.
- Commands to launch the service –

```
// creates a new virtual env
Python virtualenv env

// Install all dependencies
make bootstrap

// Check for coding rules using flake8 program
make lint

// Run tests
make test

// Run server
make server

To launch as a daemon service:
nohup make server . &
```

## 3.3 Client Installation

- In the src/android directory you should find the Geospatial.apk . To install this application on your android phone, goto your android SDK installation directory and use the adb in the tools directory command to install

  ***./adb install <path-to-file>***

- The application is configured and hosted on our private server at **http://vswamy.net:8888/**