Eventbrite For Android

Documentation for Milestone 1

Kumaran Manangatti Dharman Lekha
Venkata Saketh Ram Vemuru
Jason Lescano
Megha Vashisht
Prithvi Mallepula

GitHub repo: https://github.com/mdlkumaran/Event-Brite-Clone

Contents

Introduction

Eventbrite is an android application that gives the information of the events taking place around a certain location. The user is given the option to choose from a number of locations where he/she wishes to see the events for. Each user must have an account which can be created by email or by other social media accounts i.e. Facebook and Google+. All the events and user data is stored in Firebase for Android.

System Requirements Hardware: Android 4.4 or older

Software Systems Used: Android Studio, Google Firebase

Glossary

Firebase Backend for Android:

FireBase is a scalable backend database service provided by Google which enables users to link it to their Android application. This provides an easy way to read, store and curate data on demand and in real time.

Activity

Activity is a component of the application. It is one of the basic building block of the android application. These are where all the action happens, because they are the screens that allow the user to interact with your app.

Fragment

Fragment is a piece of an activity which enable more modular activity design. It will not be wrong if we say, a fragment is a kind of **sub-activity**

REST

Representational State Transfer (REST) is a platform independent, language independent software architecture style used for web services. It relies on a stateless, client-server, cacheable communications protocol (HTTP/ HTTPS).

JSON

Java Script Object Notation (JSON) is a data interchange format used by REST services. Data fetched from a DB is serialized in an order of key value pairs and used by client.

Dashboard

It is a graphical layout of the android application that provides a view of the finished application and also provides an overall view of the interaction flow of the app.

View controllers

Every Dashboard has one/many view controllers. It acts as a middle man between the user interface and the data. It used to write a logical code which makes the app perform a particular task.

Data Adapters

Each listed component is populated with the data from the DB using Data Adapters

GIT

Tool used for source and version control within the team.

Requirement Specification:

Register/Login: The user is required to enter his credentials such as username and password in order to get a valid account authorized in order to use the app. Once created, he can login/logout any number of time in order to use the features provided by the EventBrite app.

User Personalization Page: After logging in, the user will be given the option to choose different event categories so that the event search results will be filtered according to those selections. Examples of categories include “Science & Technology”, “Sports”, “Music” and “Food”.

Events List: After the user has logged in and selected some categories, a list containing all events in the location given and under the criteria will be shown

User Profile Page: This page contains information about the user, including their name, location, interests, past events attended as well as events they are currently registered for.

Custom Events Search/Filter Page: The user can search for free and paid events
.
Upload Picture Event: The users with proper hardware in the device can use the camera of the device and capture images of the event they are attending. On capturing the image and accepting the image to be uploaded the user clicks on a accept in this case a tick mark which then automatically uploads the images captured into the Google Firebase Database and stores it locally in the phone memory as well.
The camera being used takes into consideration the phone's hardware and uses all the capabilities designed for the camera of the device.

Event Registration: This is where the user can register for an event of their choosing. The user will be required to enter some information in order for this process to be completed.

Rating & Reviewing Events system: The users can review and rate the events they have previously attended.

Add Friends/ Share Page: Users will now be able to have in application friends, and then also share events they are going to with them.

Heat Map Page: The "hot" zones will depict popular events with many attendees while the "cold" zones will represent smaller events with fewer participants.

User Requirements

1. The user must have an android device with Android 4.4 or older.
2. The user must have an internet connection.
3. The user must have a valid email address to create the account.
4. The user must allow the application to use the Phones camera and internal Storage in order to store the images of the event.
5. The user must allow the application to use the Phones GPS Location inorder to use the maps module.

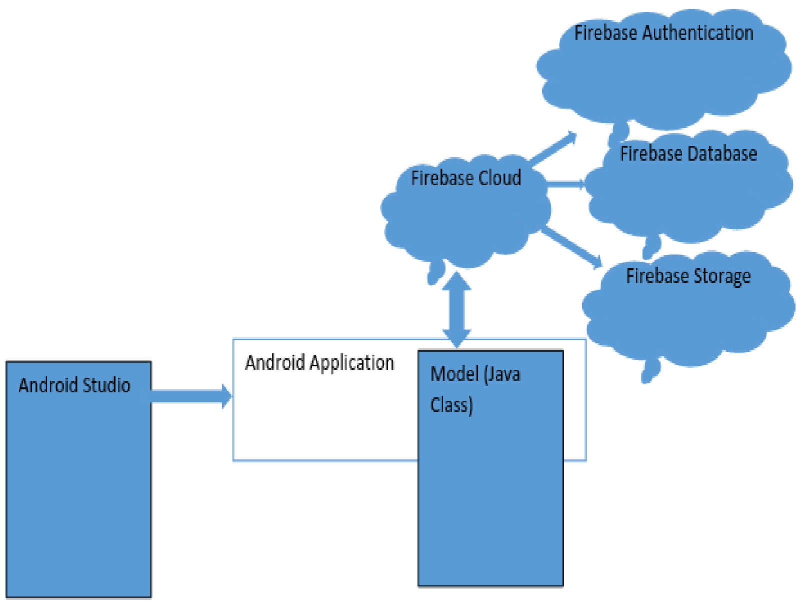
System Architecture

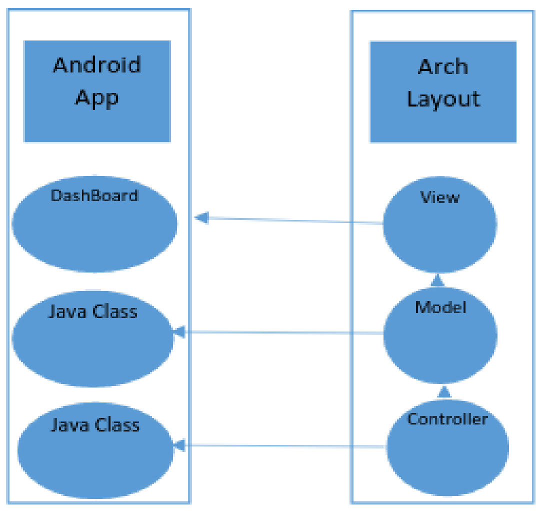
The app runs in various layers. It uses android framework for frontend and firebase is used for the backend.

The system employs a Model-View-Controller (MVC) architecture. The controller and view blocks do not directly interact with each other, they do it with the help of Model. The user requests are sent to the controller. The model class contains the modules needed to fetch data from an external storage and parse it in the format, the view can display. The data is then passed to the view block and this is rendered to the user. As depicted in figure, the view equivalent in Android is the layout.xml files, model and the controller are written in separate Java files. In this system's case, the model java file comprises of a function fetches data from the DB instance hosted in the Firebase cloud and serializes the data, which can be manipulated and used in the view.

The model block of every module (that requires data from DB) in the app consists of a function comprises of Java file which fetches data from DB in the cloud and returns it to the model block as serialized data in String format. This String response is then parsed and manipulated by the model block to fulfill the functional needs and then passed on to the view block.

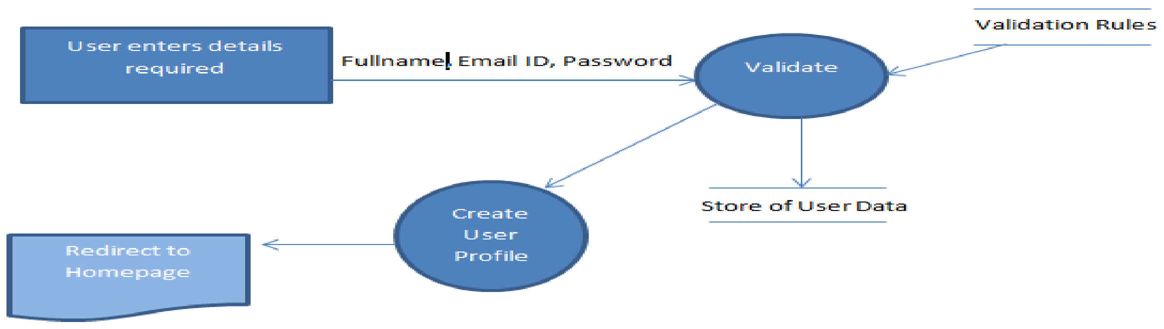
Architecture Diagram:



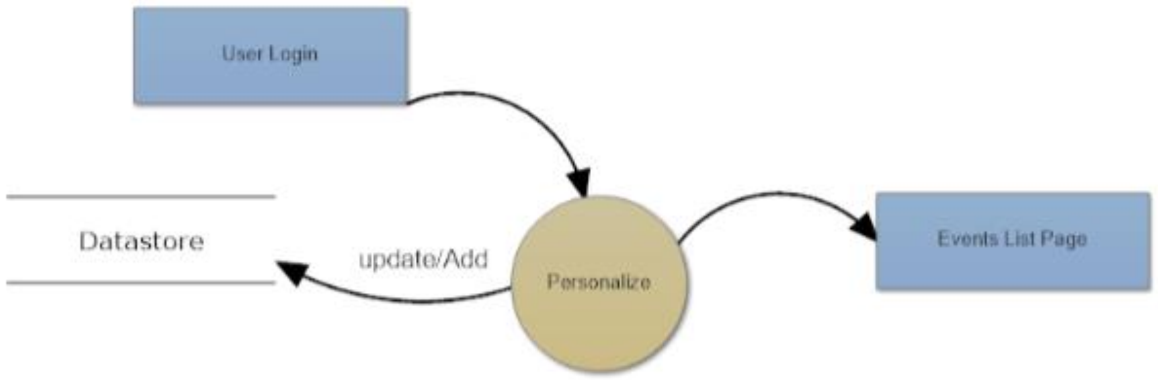


Data Flow Diagram:

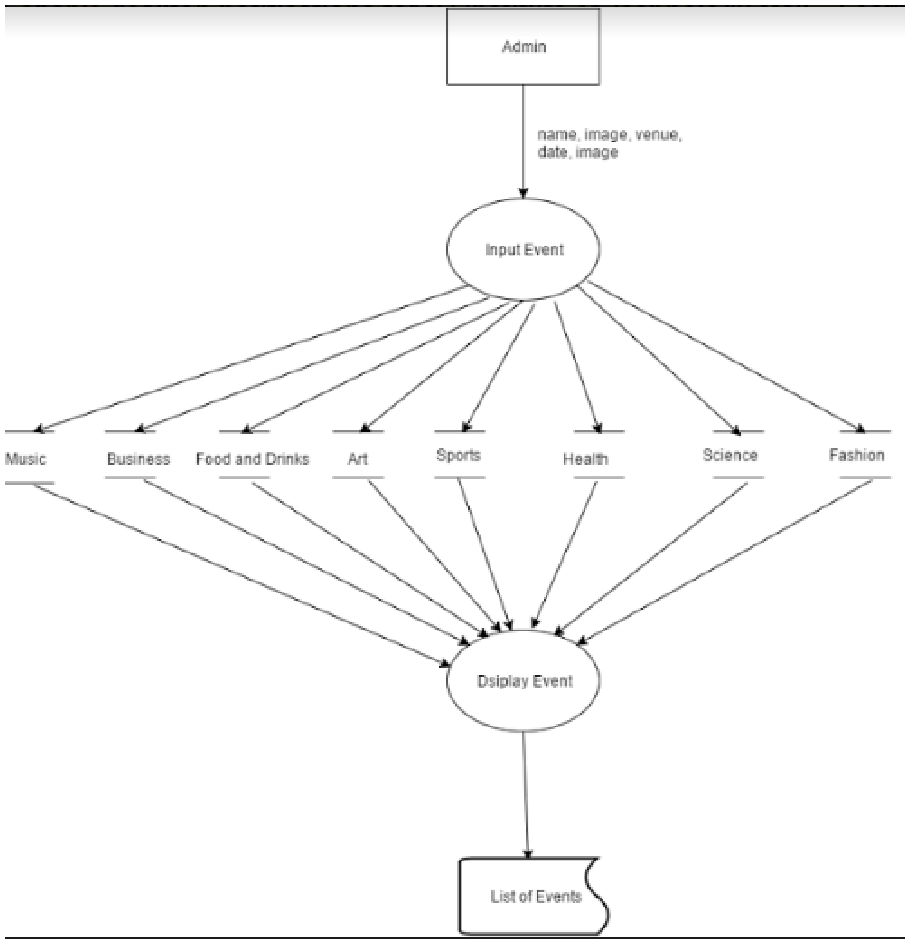
1. Login Page Flow Diagram:



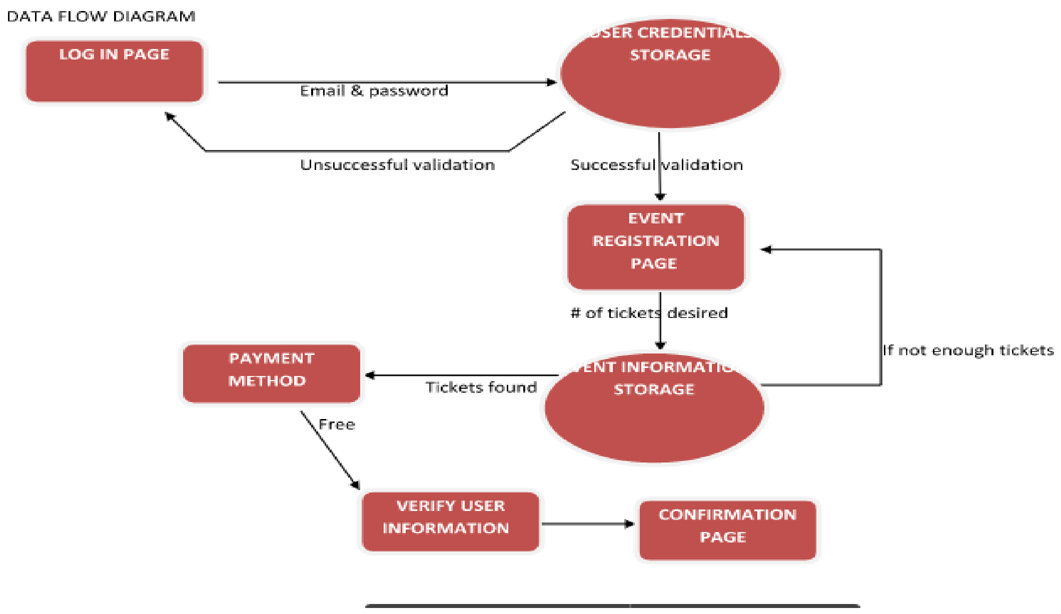
2. User Personalization Page:



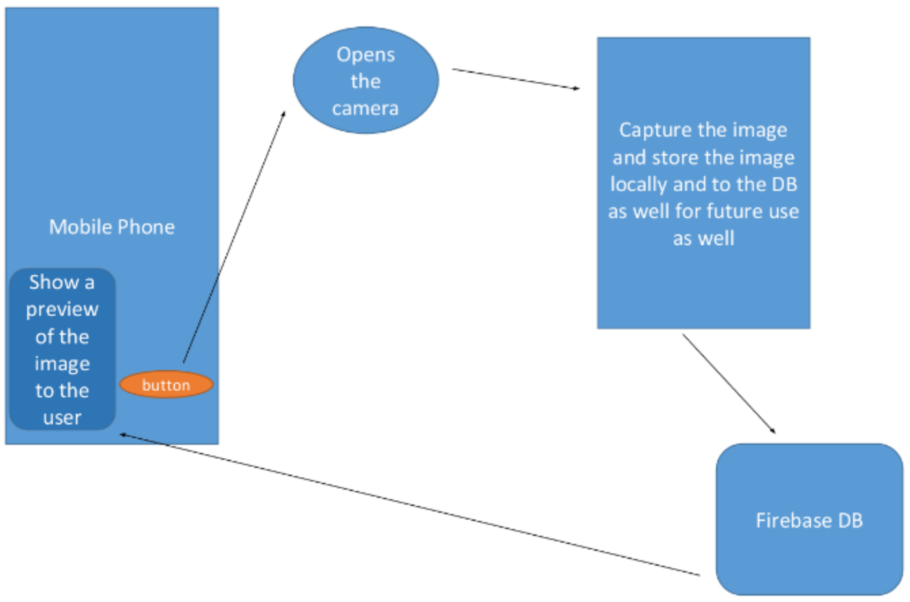
3. User Events Page Flow Diagram:



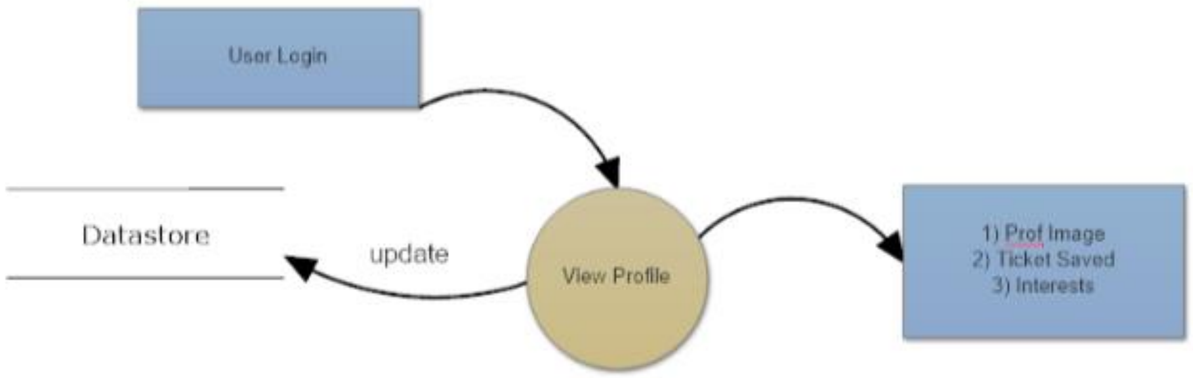
4. User Registration



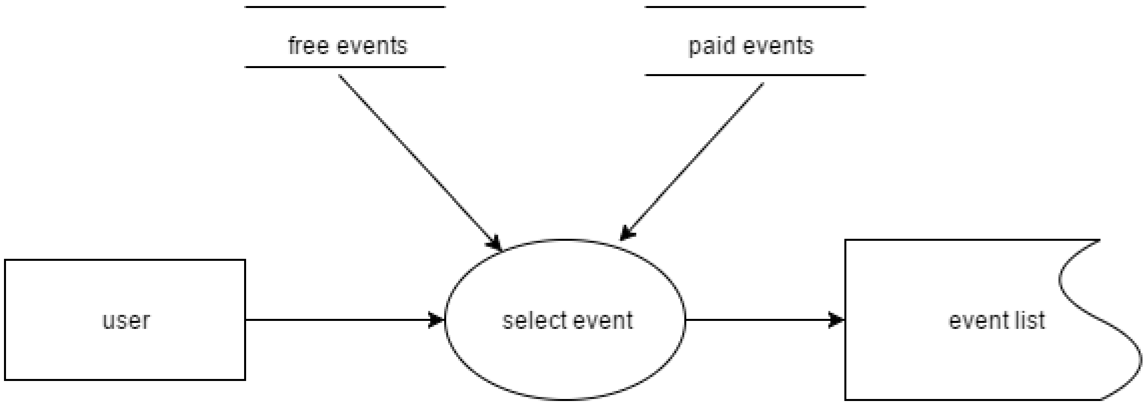
5. Camera Flow Diagram:


Mobile Phone
Show a preview of the image to the user
button
Opens the camera
Capture the image and store the image locally and to the DB as well for future use as well
Firebase DB

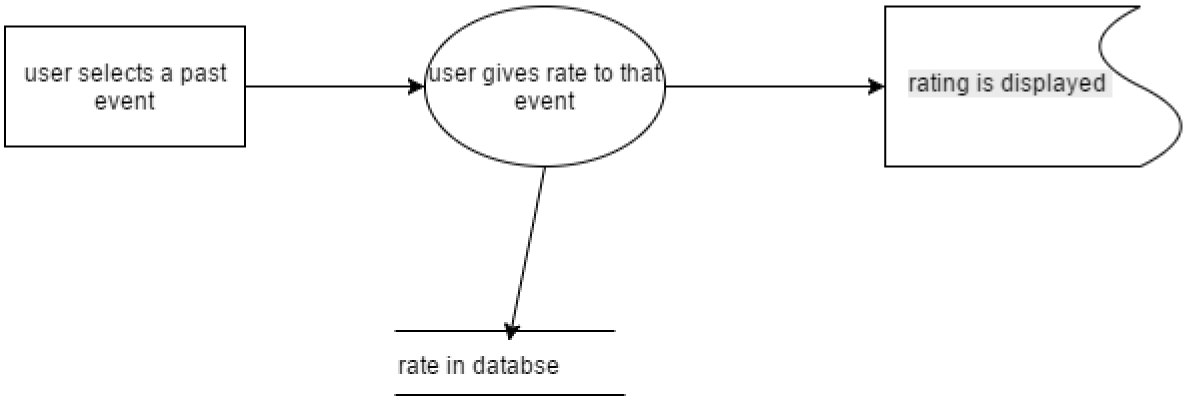
6. User Profile Page Flow Diagram:

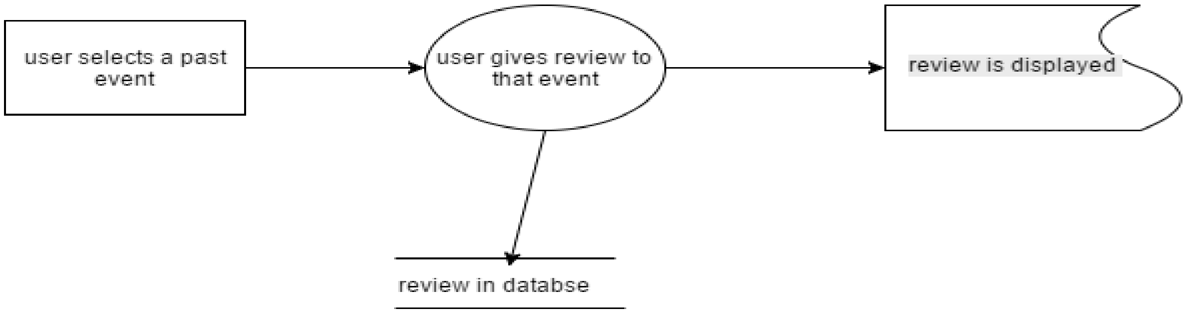


7. Custom Search Page

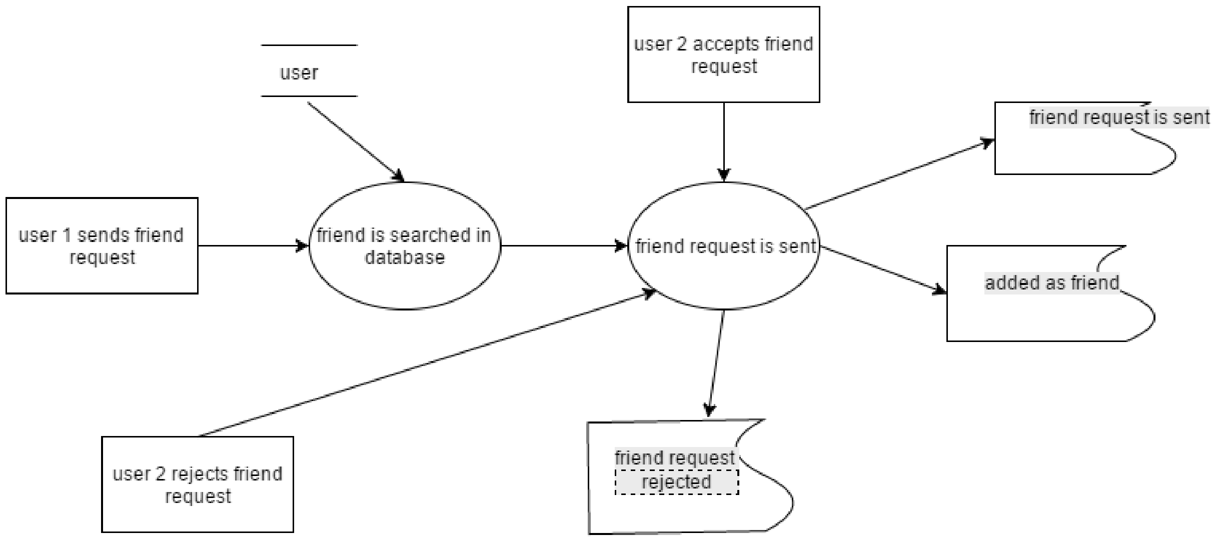


8. Rating and Review

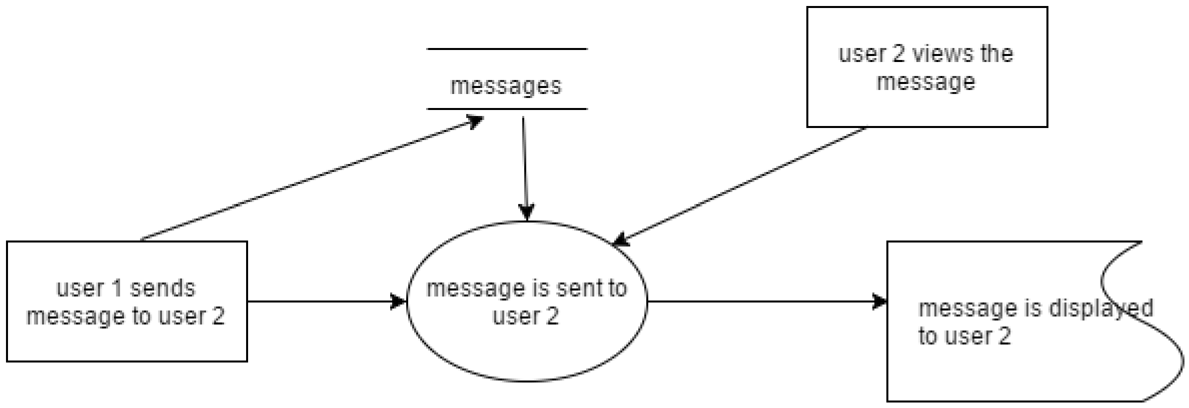




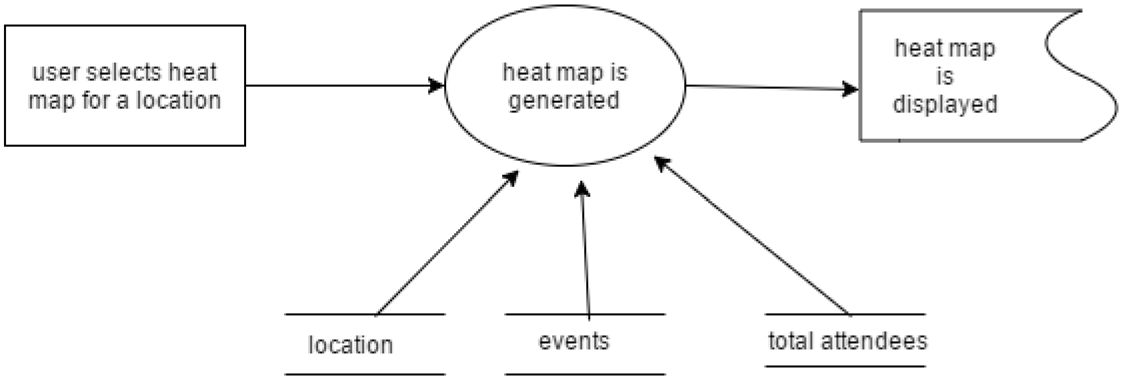
9. Add friends


user
user 2 accepts friend request
friend request is sent
user 1 sends friend request
friend is searched in database
friend request is sent
added as friend
user 2 rejects friend request
friend request rejected

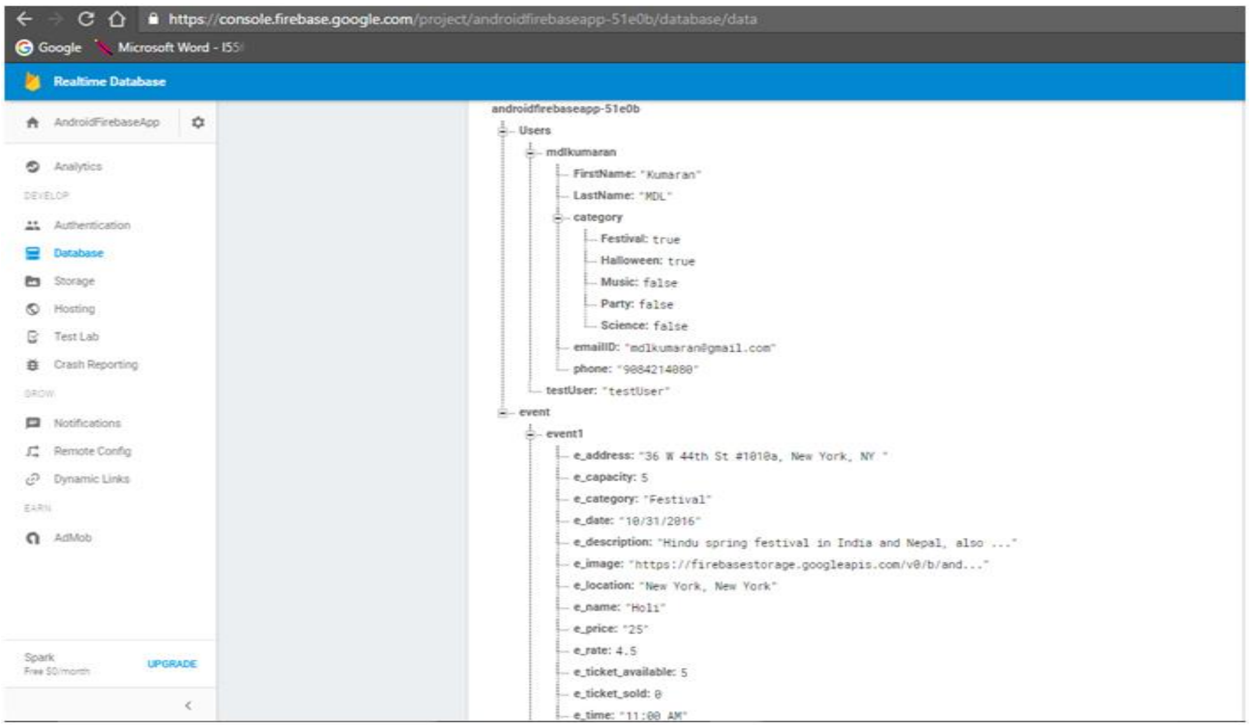
10. Send and View Message


messages
user 2 views the message
user 1 sends message to user 2
message is sent to user 2
message is displayed to user 2

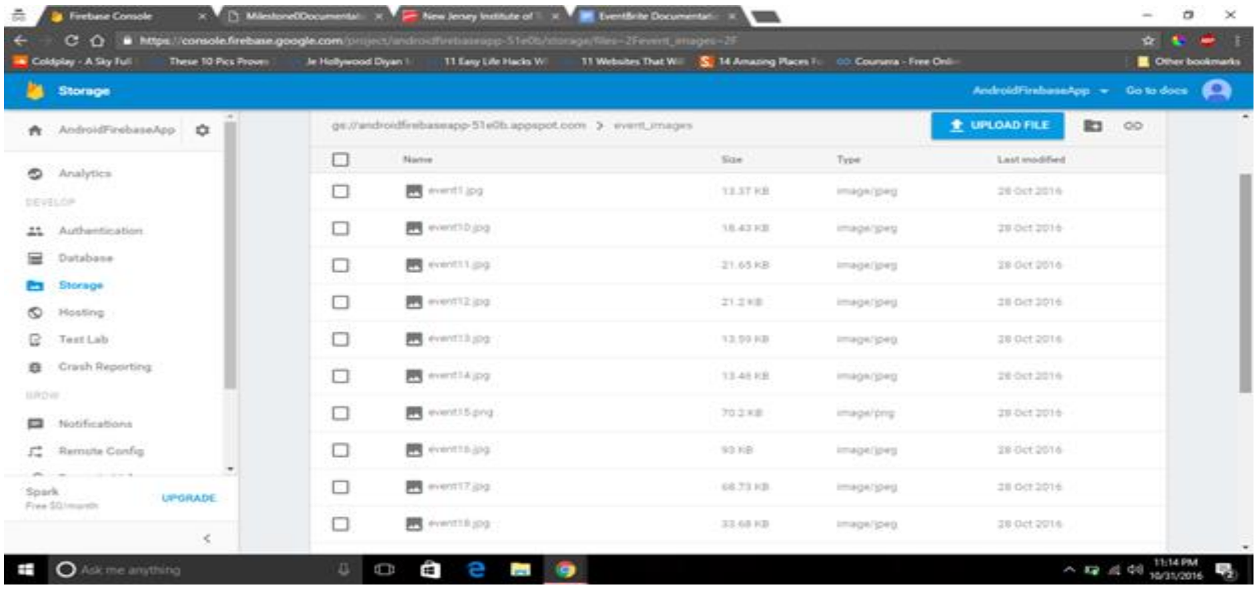
11. Heat Map



Database Model



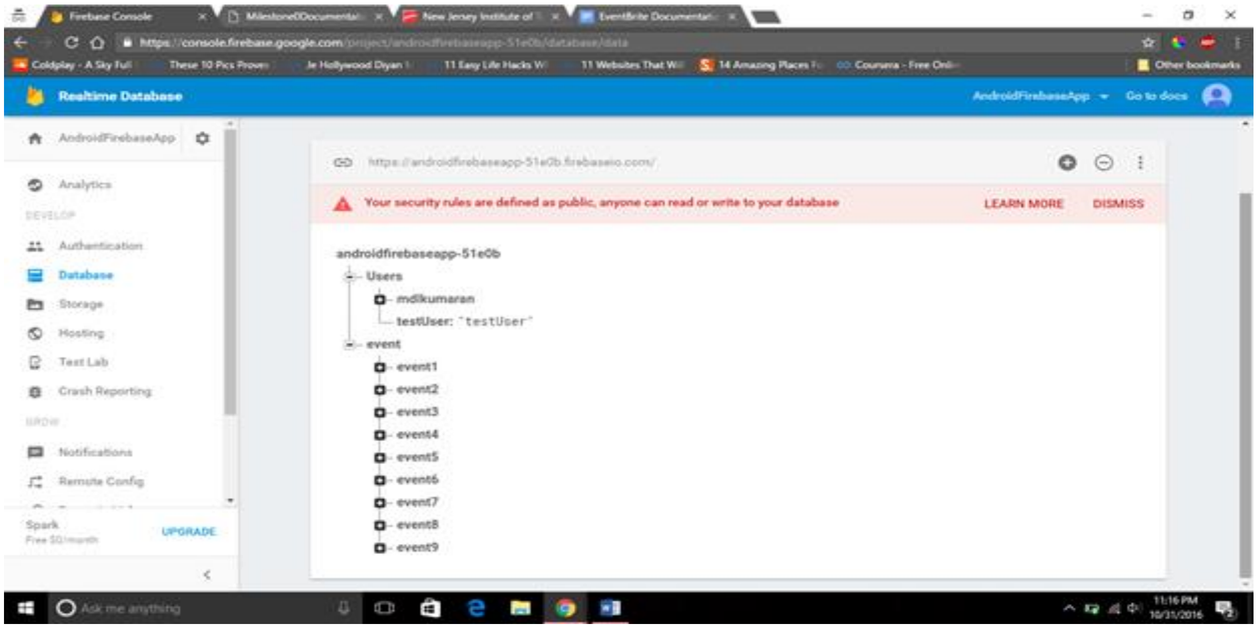
```
androidfirebaseapp-51e0b
  Users
    mdlkumaran
      FirstName: "Kumaran"
      LastName: "MDL"
      category
        Festival: true
        Halloween: true
        Music: false
        Party: false
        Science: false
      emailID: "mdlkumaran@gmail.com"
      phone: "9084214080"
    testUser: "testUser"
  event
    event1
      e_address: "36 W 44th St #1010a, New York, NY "
      e_capacity: 5
      e_category: "Festival"
      e_date: "10/31/2016"
      e_description: "Hindu spring festival in India and Nepal, also ..."
      e_image: "https://firebasestorage.googleapis.com/v0/b/and..."
      e_location: "New York, New York"
      e_name: "Holi"
      e_price: "25"
      e_rate: 4.5
      e_ticket_available: 5
      e_ticket_sold: 0
      e_time: "11:00 AM"
```



gs://androidfirebaseapp-51e0b.appspot.com > event_images

Name	Size	Type	Last modified
event1.jpg	13.37 KB	image/jpeg	28 Oct 2016
event10.jpg	18.43 KB	image/jpeg	28 Oct 2016
event11.jpg	21.65 KB	image/jpeg	28 Oct 2016
event12.jpg	21.2 KB	image/jpeg	28 Oct 2016
event13.jpg	13.59 KB	image/jpeg	28 Oct 2016
event14.jpg	13.46 KB	image/jpeg	28 Oct 2016
event15.png	70.2 KB	image/png	28 Oct 2016
event16.jpg	93 KB	image/jpeg	28 Oct 2016
event17.jpg	68.73 KB	image/jpeg	28 Oct 2016
event18.jpg	33.68 KB	image/jpeg	28 Oct 2016

All the images are stored in the database



https://androidfirebaseapp-51e0b.firebaseio.com/

Your security rules are defined as public, anyone can read or write to your database

```
androidfirebaseapp-51e0b
  Users
    mdlkumaran
    testUser: "testUser"
  event
    event1
    event2
    event3
    event4
    event5
    event6
    event7
    event8
    event9
```

All the events are listed in the database

User Interaction Flow / User Manual:

Registration Page

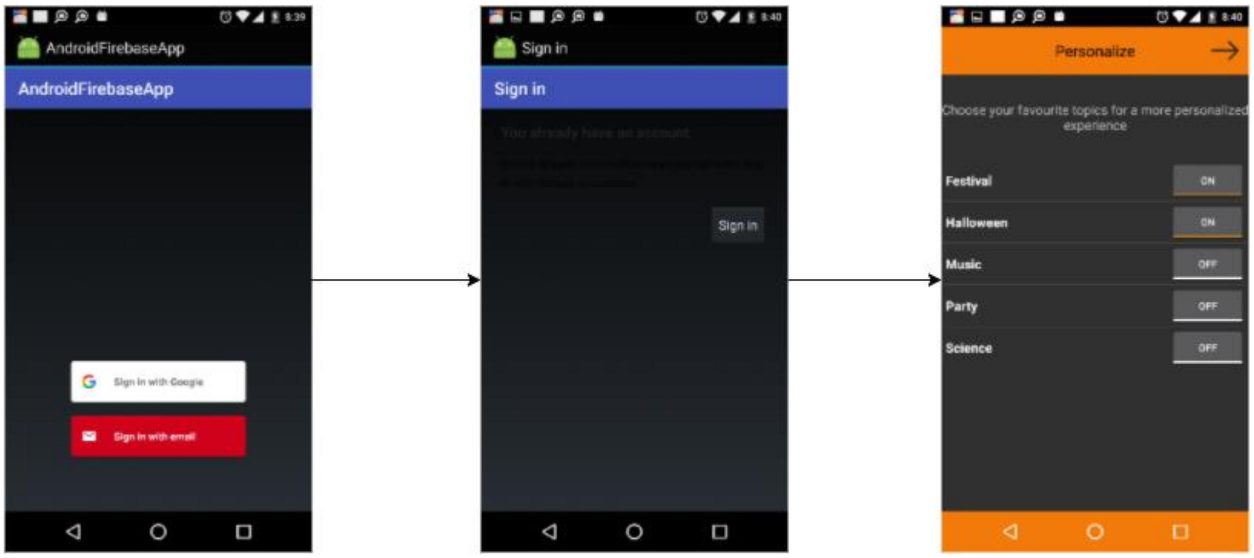
As the user opens the app, the registration page is displayed. The user needs to input a valid email address and password in order to create an account. The user can also use Facebook or Google+ account to login.

Login Page

If the user already has an account, then he needs to enter the email and correct password to login and verify his account. The user can simply login by Facebook as well as Google+.

Category Selection

Once the user is logged in, he can choose from a list of categories for the events he wants to view. All the events from the selected category will be displayed.

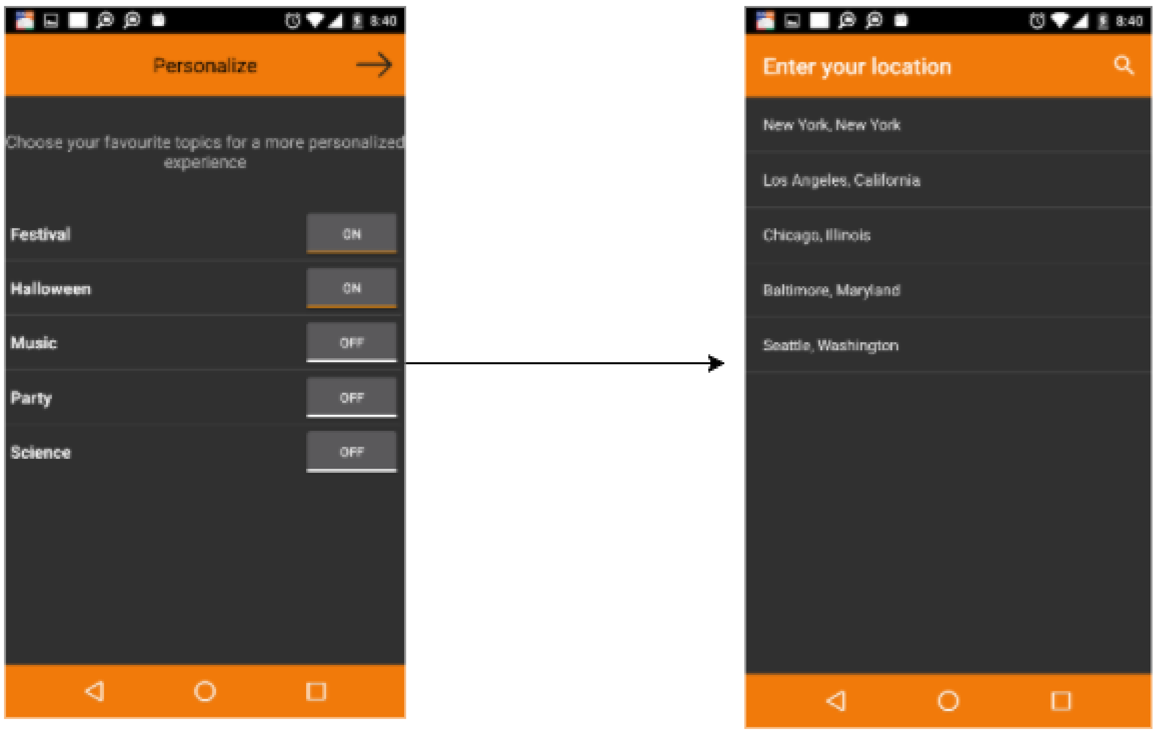


Choose your favourite topics for a more personalized experience

Festival ON
Halloween ON
Music OFF
Party OFF
Science OFF

Location Selection

Next the user has to select the location for which he wants to view the events. He has to select from a list of location provided in the list.



Personalize

Choose your favourite topics for a more personalized experience

Festival	ON
Halloween	ON
Music	OFF
Party	OFF
Science	OFF

Enter your location

New York, New York

Los Angeles, California

Chicago, Illinois

Baltimore, Maryland

Seattle, Washington

Event List

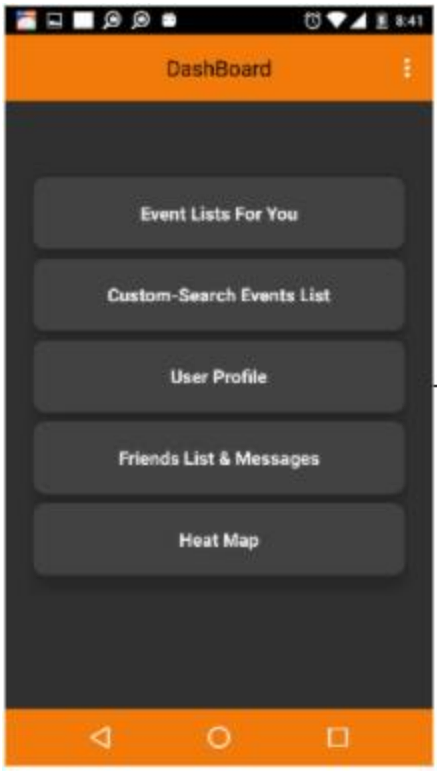
Once the user has selected the category and the location all the events corresponding to that category and event will be displayed in form of a list.

Event Description

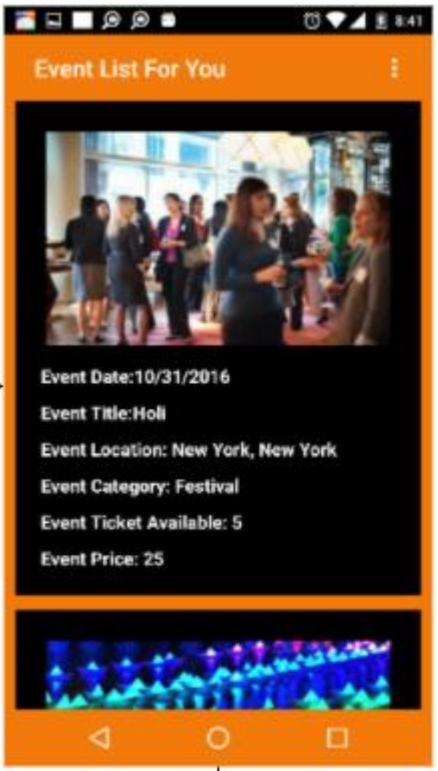
Once the user has selected an event, the information corresponding to it will be shown in the next page. This includes, date, time, place and a brief description.

Ticket Details

This page contains the total number of tickets given to the user for an specific event and if the tickets are free or have to be paid for.



DashBoard



Event List For You

Event Date:10/31/2016

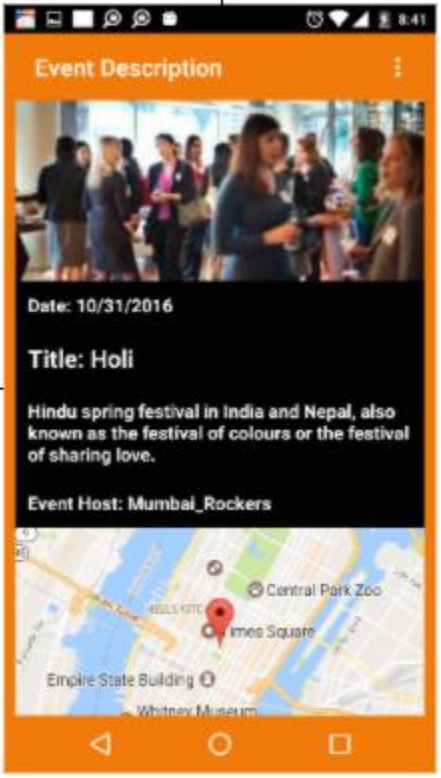
Event Title:Holi

Event Location: New York, New York

Event Category: Festival

Event Ticket Available: 5

Event Price: 25



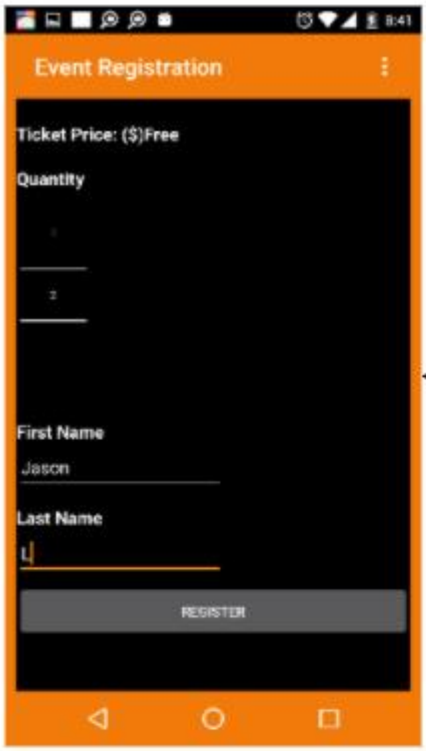
Event Description

Date: 10/31/2016

Title: Holi

Hindu spring festival in India and Nepal, also known as the festival of colours or the festival of sharing love.

Event Host: Mumbai_Rockers



Event Registration

Ticket Price: ($)Free

Quantity

1

2

First Name

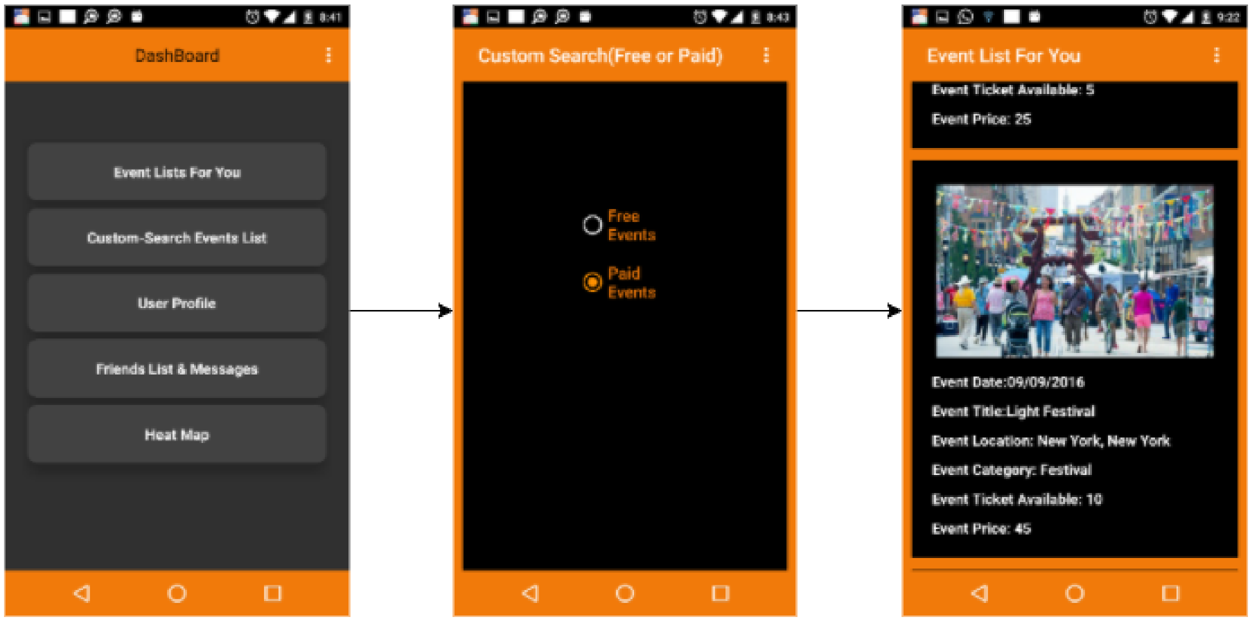
Jason

Last Name

L

<u>Custom Search:</u>

The user will be able to search events by price, either free or paid and will get the results according to his selection.

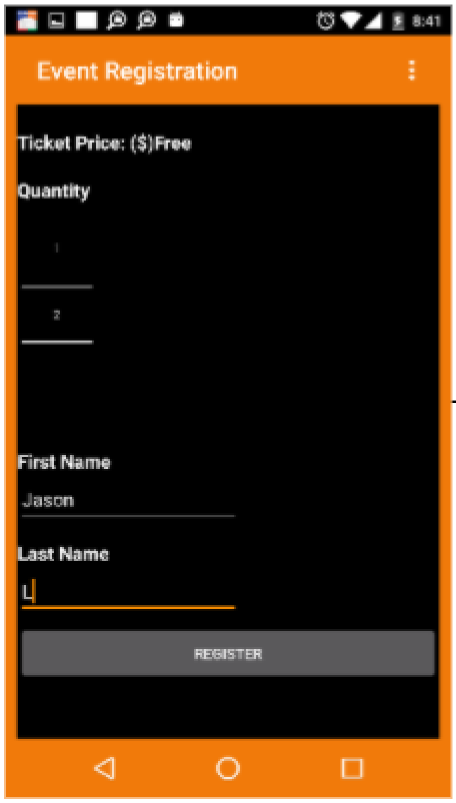


Event Ticket Available: 5
Event Price: 25

Event Date:09/09/2016
Event Title:Light Festival
Event Location: New York, New York
Event Category: Festival
Event Ticket Available: 10
Event Price: 45

Event Registration

The user can register for the event by buying the ticket through the app.



Event Registration

Ticket Price: ($)Free

Quantity

1

2

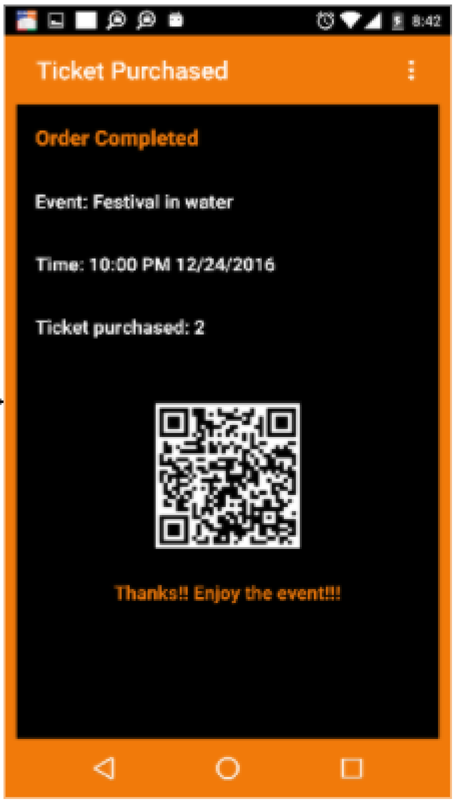
First Name

Jason

Last Name

L

REGISTER



Ticket Purchased

Order Completed

Event: Festival in water

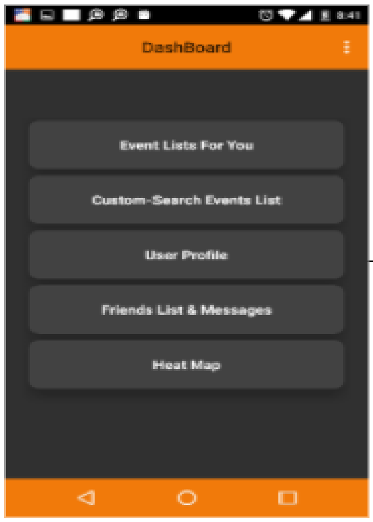
Time: 10:00 PM 12/24/2016

Ticket purchased: 2

Thanks!! Enjoy the event!!!

User Profile

This will show all the tickets ordered by the user.



DashBoard

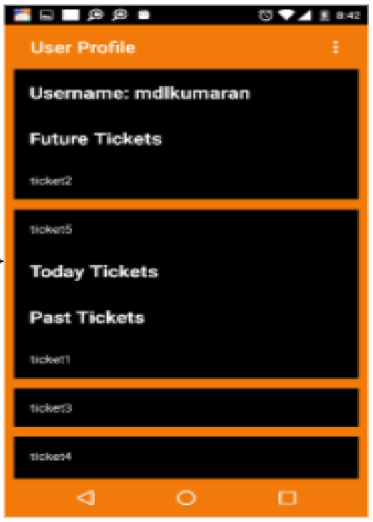
Event Lists For You

Custom-Search Events List

User Profile

Friends List & Messages

Heat Map



User Profile

Username: mdlkumaran

Future Tickets

ticket2

ticket5

Today Tickets

Past Tickets

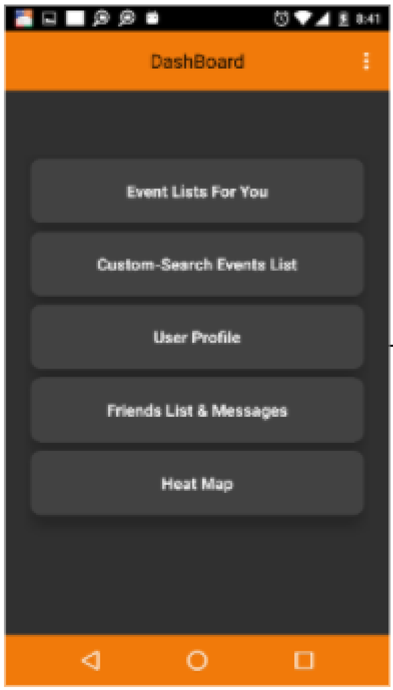
ticket1

ticket3

ticket4

Rating and Review

The user who has attended an event can add reviews and rate the event for other users.



DashBoard

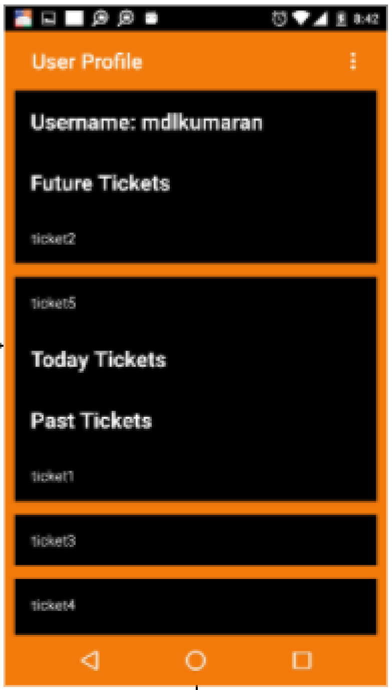
Event Lists For You

Custom-Search Events List

User Profile

Friends List & Messages

Heat Map



User Profile

Username: mdlkumaran

Future Tickets

ticket2

ticket5

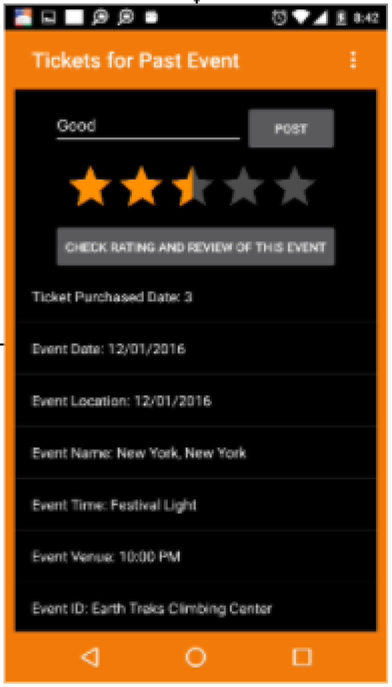
Today Tickets

Past Tickets

ticket1

ticket3

ticket4



Tickets for Past Event

Good

POST

CHECK RATING AND REVIEW OF THIS EVENT

Ticket Purchased Date: 3

Event Date: 12/01/2016

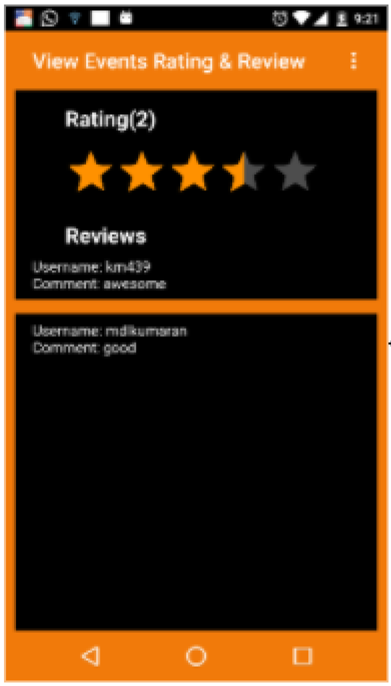
Event Location: 12/01/2016

Event Name: New York, New York

Event Time: Festival Light

Event Venue: 10:00 PM

Event ID: Earth Treks Climbing Center



View Events Rating & Review

Rating(2)

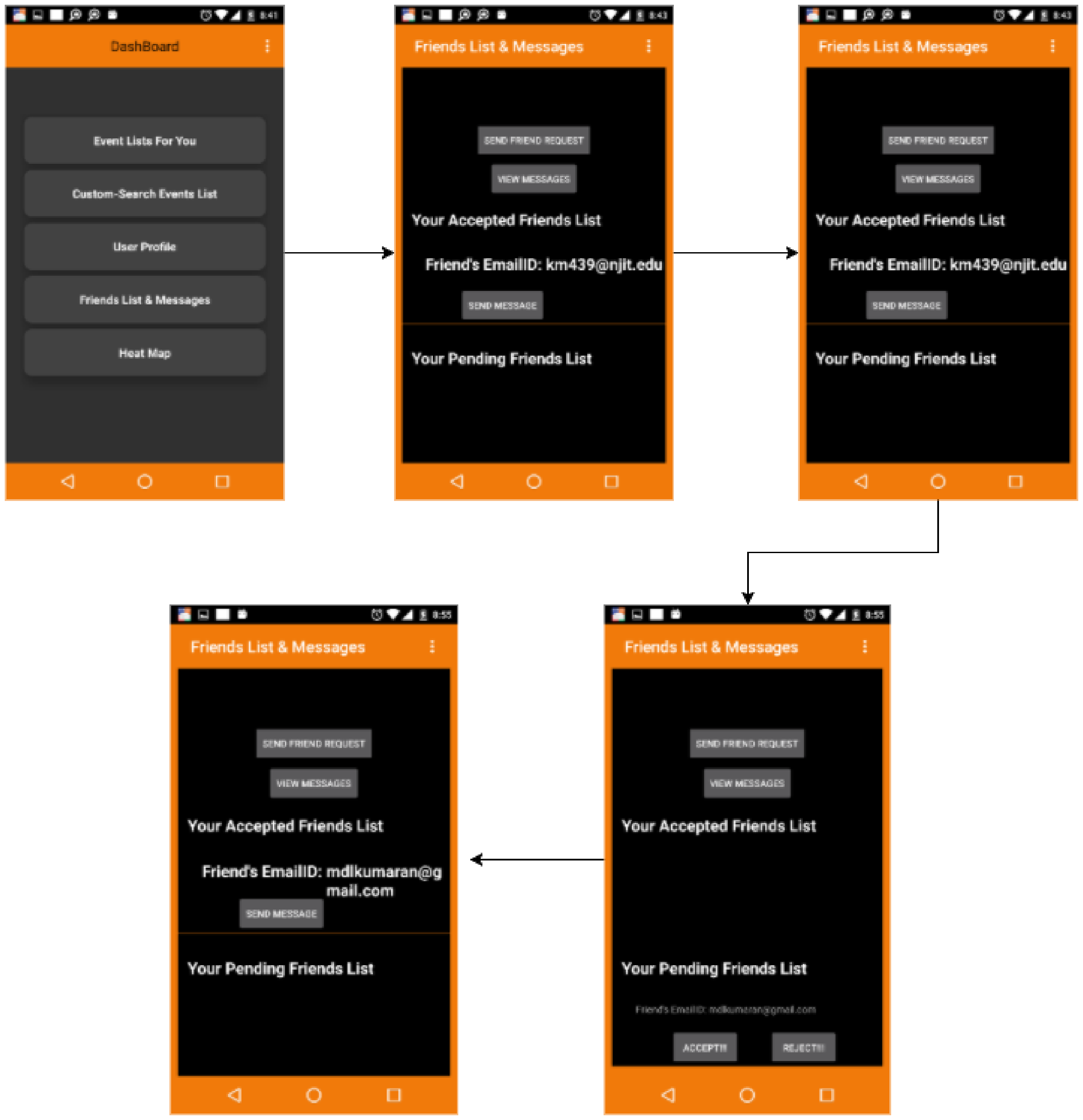
Reviews

Username: km439
Comment: awesome

Username: mdlkumaran
Comment: good

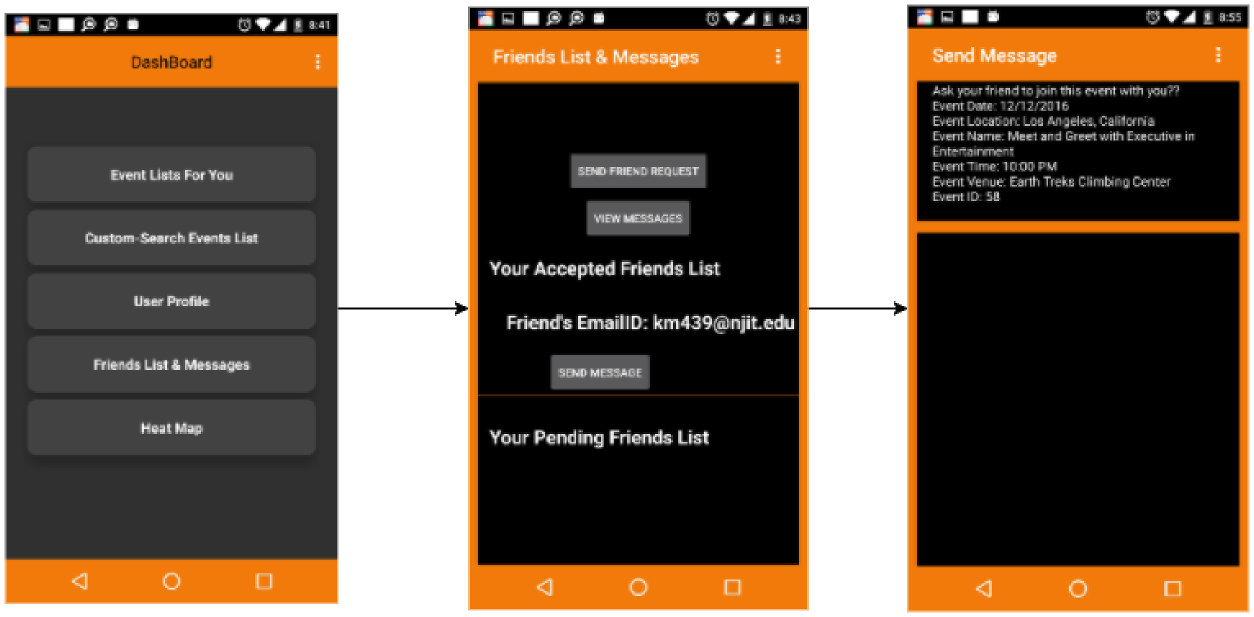
<u>Add Friend</u>

The user can invite the other user of the app to view the events he is going to. The second user can accept or reject the request to add as friend.


DashBoard
Event Lists For You
Custom-Search Events List
User Profile
Friends List & Messages
Heat Map
Friends List & Messages
SEND FRIEND REQUEST
VIEW MESSAGES
Your Accepted Friends List
Friend's EmailID: km439@njit.edu
SEND MESSAGE
Your Pending Friends List
Friends List & Messages
SEND FRIEND REQUEST
VIEW MESSAGES
Your Accepted Friends List
Friend's EmailID: km439@njit.edu
SEND MESSAGE
Your Pending Friends List
Friends List & Messages
SEND FRIEND REQUEST
VIEW MESSAGES
Your Accepted Friends List
Your Pending Friends List
Friend's EmailID: mdlkumaran@gmail.com
ACCEPT!!
REJECT!!
Friends List & Messages
SEND FRIEND REQUEST
VIEW MESSAGES
Your Accepted Friends List
Friend's EmailID: mdlkumaran@gmail.com
SEND MESSAGE
Your Pending Friends List

Send Message

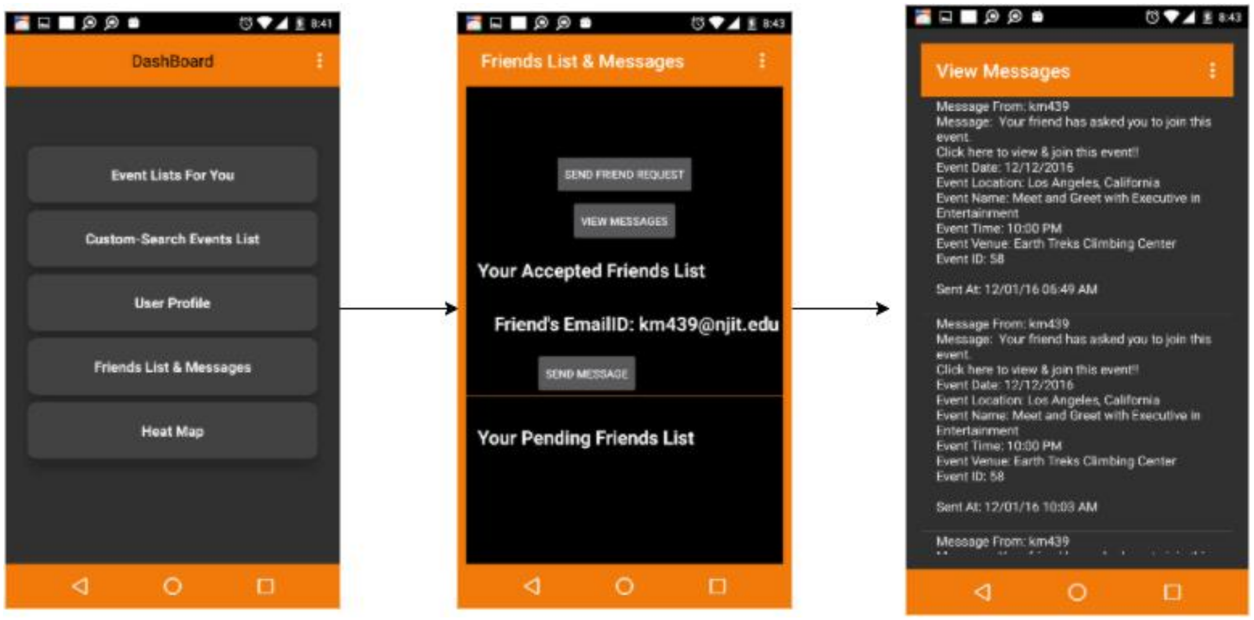
The user can send message to other user who is his/her friend on the app to invite to the friend to event he/she is going to.



Ask your friend to join this event with you??
Event Date: 12/12/2016
Event Location: Los Angeles, California
Event Name: Meet and Greet with Executive in Entertainment
Event Time: 10:00 PM
Event Venue: Earth Treks Climbing Center
Event ID: 58

<u>View Message</u>

The other user can view the message from his friend to get the details of the event he is invited to. The detail of when the message is sent is also displayed after every message.



Friend's EmailID: km439@njit.edu

Message From: km439
Message: Your friend has asked you to join this event.
Click here to view & join this event!!
Event Date: 12/12/2016
Event Location: Los Angeles, California
Event Name: Meet and Greet with Executive in Entertainment
Event Time: 10:00 PM
Event Venue: Earth Treks Climbing Center
Event ID: 58

Sent At: 12/01/16 06:49 AM

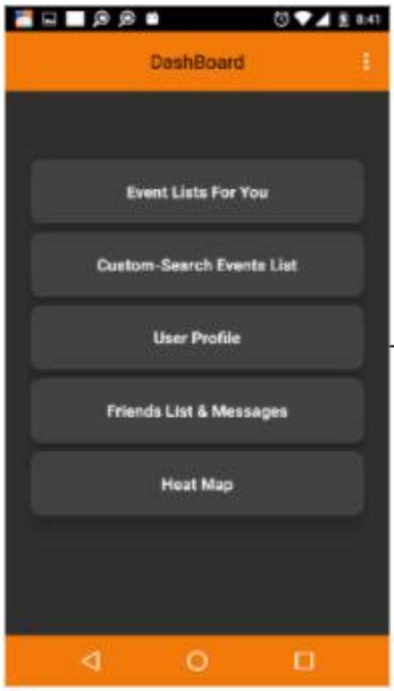
Message From: km439
Message: Your friend has asked you to join this event.
Click here to view & join this event!!
Event Date: 12/12/2016
Event Location: Los Angeles, California
Event Name: Meet and Greet with Executive in Entertainment
Event Time: 10:00 PM
Event Venue: Earth Treks Climbing Center
Event ID: 58

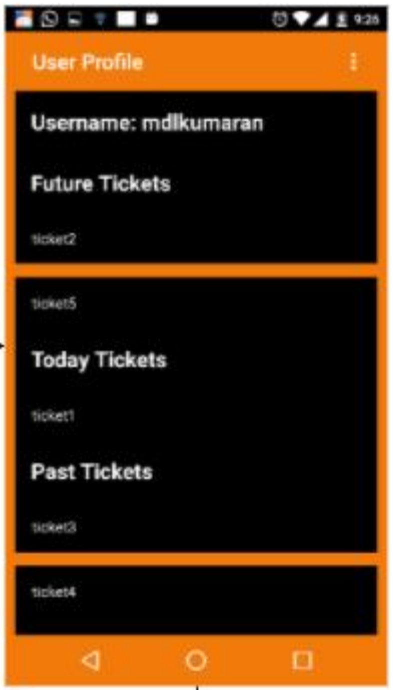
Sent At: 12/01/16 10:03 AM

Message From: km439

<u>Add Image</u>

The user can also upload the image of the event he has visited. The user needs to go to his profile page and look for the event he has visited in the past and upload the picture for that event.





Username: mdlkumaran

Future Tickets

ticket2

ticket5

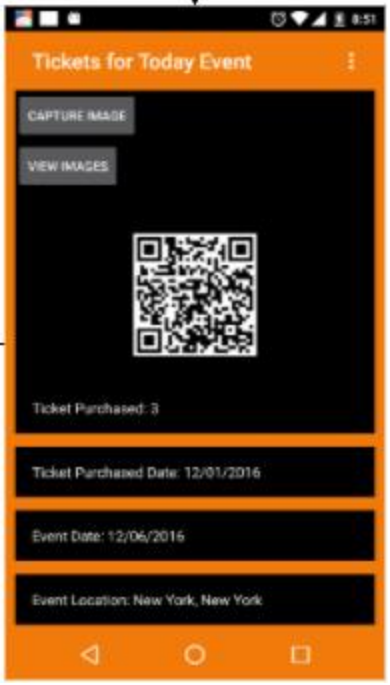
Today Tickets

ticket1

Past Tickets

ticket3

ticket4

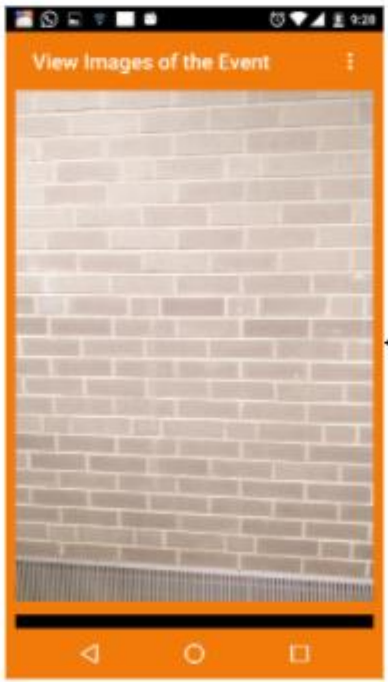


Ticket Purchased: 3

Ticket Purchased Date: 12/01/2016

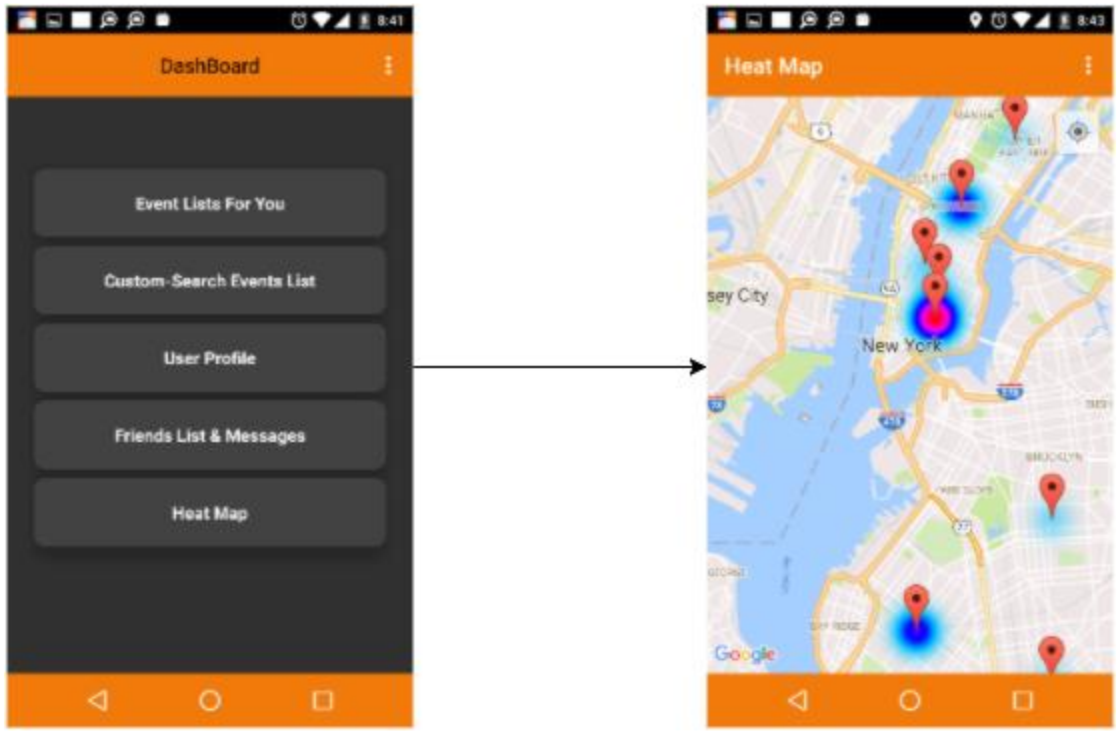
Event Date: 12/06/2016

Event Location: New York, New York



Heat Map

The user can view the heat map of the location through the dashboard. The heat map shows the intensity of the attendees of the events in a particular location in the map by the intensity of color.



Software Validation

The software has been validated by running each activity individually. Unit testing has been performed on the Android devices to check the performance of each activity.

System Testing

LOG IN

Test Case: Login using various methods (Credentials, Facebook, Google+)

Description: This test ensures that the user can successfully log into the application and get his/her credentials/social media account validated so that entry into the application (and use of the features therein) is granted.

Pre-req Test:

(None. As it is the initial page)

Starting State: Application goes to the "Login-User Page" upon launch of application**.**

I. Action: Either enter credentials and click the "login" button. Else click desired social media buttons.

Result: If successfully validated, entry to the next page is granted: "Personalize categories" page.

Pass/Fail Criteria: If user cannot login to app even after entering correct credentials, the test case is a fail

Pass/Fail: Pass

CAMERA APPLICATION/UPLOAD EVENT IMAGES

Users have the ability to use the device's camera application to take pictures when they are attending an event. The camera application is enabled only when the users device is camera hardware compatible.

Test Case: Use the device camera to capture images and store them on the database and in the phone memory.

Description: This will ensure that users with only camera enabled in the phone can capture the images.

Action: On clicking the camera the images are stored in the database (Firebase) in this case and locally in the phone memory

Result: The images are captured with proper clarity and stored in the firebase database and in the memory card of the phone as well.

Pass/Fail Criteria: If the captured images are not stored into the database or the camera does not launch, test is a fail.

Pass/Fail: Pass

<u>TICKET REQUEST:</u>

Test Case: Request more tickets than the amount available for an event.

Description: This will ensure that the amount of event attendees does not exceed its intended capacity.

Pre-req Test: Once an event is selected from the Events List page, the user will be redirected to the Events Registration page. The user will also be required to be logged in with a valid account in order to register to attend events.

Starting State: The user will enter the amount of tickets desired or just be given one, depending on the event.

I. Action: Click the Submit button to send the information to the database.

Result: If enough ticket(s) available, then the user will proceed to the Confirmation page to verify the information entered. If there are not enough tickets, the user will be notified and will have the option to either lower the amount of tickets desired or go back to the events list page.

Pass/Fail Criteria: If after clicking the submit button, the ticket request is not generated, test case is a fail.

Pass/Fail: Pass

<u>USER REGISTRATION</u>

Test Case: Attempt to register to an event without submitting any information

Description: To make sure that the event attendees' list contains accurate contact information from the users.

Pre-req Test: The user will be required to be logged in. He/she will have to submit some information after selecting an event and requesting a certain number of tickets.
Starting State: The user will have to submit some personal information in order to successfully register for an event.
I. Action: Click the Submit button to send the user registration information to the database and proceed to the ticket page.
Result: If the information entered is not valid, an error will be shown to the user, so they can promptly fix the information. Only then, he/she will be able to move to the next page.
Pass/Fail Criteria: If tickets are not correctly generated after details are entered, test case is a fail.
Pass/Fail: Pass

<u>Add Friends</u>
Test Case: Attempt to send request to another user to be added as a friend by providing email ID.
Description: To provide a social aspect to this app, users are allowed to send requests to other users which they in turn can accept and be added as a friend.
Pre-req Test: The user will be prompted to provide the email ID of the friend they want to add.
Starting State: The user will have to submit the email ID of the user he wishes to add as a friend.
I. Action: After providing the email ID and clicking the send request button, a request should be sent to the friend for whom the email ID was provided.
Result: If the email ID entered is valid, a request will be sent to that friend which he/she can choose to accept/ignore. Only after accepting, he/she will be added as a friend. After accepting the request, that friend should be visible in the “Your accepted friends list”.
Pass/Fail Criteria: After sending a request, if the friend does not receive the request or if after accepting a request, friends are unable to be seen, test is a fail.

Pass/Fail: Pass

<u>Personalize Page:</u>

Test Case: Allow the user to select/unselect his/her preferences for the category of events as well as their preferred location.

Description: Personalizing the users page with respect to his event category preference and location allows them to find events which fit that description or within a particular location.

Pre-req Test: After logging in, the user is prompted to select or deselect the event category or location he wants.

Starting State: All categories will be unchecked. After checking the categories, he wants, the user will get redirected to the location selector page.

I. Action: After selecting the categories and locations he wishes to find events in, he clicks on the next icon on the upper right corner.

Result: After personalizing the page based on the user's category preferences as well as location preferences, the app will show only those events which collate with the category and location provided by the user.

Pass/Fail Criteria: If the events are not generated after providing category and location preference, test case is a fail.

Pass/Fail: Pass

<u>Events List:</u>

Test Case: The list of events based on the user's preference and location is shown on the page. By scrolling down, the user can see all the events available near him.

Description: A list of events with details such as date, location, attendees, ticket pricing etc are displayed so the user can select that which appeals to him/her.

Pre-req Test: All the categories which the user selects in the personalize page as well as the location he selects should display the respective events based on these inputs.

Starting State: A list of events is generated on which the user can click if he/she wishes to attend it, upon which the user can go to ticket options.

I. Action: Upon clicking on the event the user wishes to explore, they are redirected to the ticket registration and event registration page.

Result: Once the required event is selected, the user may book tickets for that event on the event registration page.

Pass/Fail Criteria: If the user is unable to get details of the events selected, the test case is a fail.

Pass/Fail: Pass

EVENT REGISTRATION

Test Case: Attempt to register to an event by submitting required information regarding tickets.

Description: To make sure that the user registers for an event, details such as ticket quantity, first and last name are collected and then the ticket may be generated.

Pre-req Test: The user will be to select the event which he wants to attend on the event list page. Once something interests the user, they may choose to register for that event.

Starting State: A blank field prompting the user to enter the number of tickets as well as his name is given.

I. Action: After entering the number of tickets required and the name if the attendee, the ticket should be generated.

Result: If the information entered is valid, a ticket is generated and details of that ticket such as event details and even a QR code is provided.

Pass/Fail Criteria: If the user is unable to register for an event even after valid details are provided, test case is a fail.

Pass/Fail: Pass

Public Functions:

- **Camera_event.Class:** This is what allows the user to access the phone's camera and uploads pictures while attending the events. The pictures are then saved in the Firebase Database.
- **Custom_SearchEventList:** This allows the user to search events by price, whether they are free or have a cost. The next page in the application will show the filtered results.
- **Event_Registration.Class**: Once an event has been selected, the user will have to input some personal information, like name, email and phone number in order to be considered as an attendee for the event.
- **Event_List.Class:** After selecting a location and preferences, a list containing all events matching that criteria will be displayed to the user. Then, he/she can proceed to register for them.
- **Tickets.onCreate:** Generates the tickets for the event on which the user selects.The ticket contains the date place and time for the event. This is saved into the Database.
- **MainActivity.onCreate:** This function enables the user to sign into the app using his credentials and proceed to use the applications features. He may alternatively choose to sign on either using Google+ or Facebook.
- **Logout.onCreate:** Signs out from the app
- **Personalize_Interest.Class:** Once the user is logged in, he/she will have to choose some preferences for the events. These include, festival, music, arts, among others.
- **Location_selector.Class:** Allows the user to select the location in which to find and browse events.
- **User_Profile.Class:** This is where the user can see their past and upcoming events that they have registered for, as well as the corresponding tickets for them.

References:

- https://www.raywenderlich.com/139322/firebase-tutorial-getting-started-2
- https://firebase.google.com/docs/
- https://www.sitepoint.com/creating-a-cloud-backend-for-your-android-app-using-firebase/
- http://www.androidhive.info/2016/06/android-getting-started-firebase-simple-login-registration-auth/

Individual Contribution:

Prithvi Mallepula:

- **Modules worked on:** Login page
- **Tools Used:** User interface was done using Android studio hereas the database used was Google Firebase for Android.
- **LoginPage functionality:** Login the the application upon entering of valid email ID and password. Subsequently, user can login to the application using the Facebook or Google+ login buttons.
- **Work Done:** Worked on login page which involves entering credentials in order to access the features of the app. Alternatively, user can login with Facebook and Google + profiles. Apart from this, worked on project documentation which involves providing the description of the app in terms of software and hardware details. Also worked on curating and compiling the various DFD's in a logically fluid manner.

Venkata Saketh Ram Vemuru:

- **Modules Worked:** User profile, Camera and Images storage, sending messages to added friends.
- **Tools Used:** Android Studio, Java, Firebase database, GIT-version Control
- **Functionalities:**
  Use camera to capture and store images

Design the use profile page to change the profile picture and the details of the user

Send messages to the user and receive the messages from the user after he/she has been added

- **WorkDone:** Provide sample JSON data being used in the application to populate the same into the database.

  Worked to implement the camera functionality, so that users can capture and store images into their phone memory as well as the database Firebase in this case to store the images. Also the photo captured is displayed to the user, completed it successfully. Worked on creating the firebase database to create events and the data to be loaded in the application. The images captured being shown and the events list have been crated and updated by me. Worked on creating the entire user personalization page, where the user can click on an image view to open the camera, and store the images in the firebase and phone memory. Also completed the creation of Username and Email and updated the details to the firebase backend. Worked on the custom search to retrieve the specific data from the database by writing the appropriate the queries. In the next module worked on sending messages to friends who have been accepted and share with them the ticket details and show them only the tickets after the given specific date. Also worked to generate the sample JSON data and images to populate the database for various cities used in the application.

Kumaran Manangatti Dharman Lekha:

- **Modules Worked:** User profile, User Personalization, Heat Map
- **Tools Used:** Android Studio, Java, Firebase database, GIT-version Control
- **Functionalities:**

  Heat map to find the popular events.

  Design the use profile page to view future and past events

  User Personalization for the user to choose category

- **WorkDone:**
  - Setup the firebase database and added the events data.
  - For user category, designed the switch button to make the user on/off the personalization category
  - Designed the user profile picture for viewing past, future and today events
  - Designed the heat map page to make the user view the popular events in a particular location
  - Designed the UI for all the pages.
  - Did unit and integration test.
  - Co-ordinated with the team members to integrate the modules.
  - Styled the app with custom themes.

Jason Lescano

Modules Worked on:

Event Information Page

User Registration

Add Friends

Tools Used:

Android Studio 2.2

Cloud Sources:

Google Firebase Database

Event Information Functionalities:

Display event pictures

Show information about the event selected

A connection to the database was set up through the java file created. This made it possible to retrieve the information we had gathered and saved there. As for the pictures, they were uploaded to the database and were referenced by their specific destination. For the Google Maps feature, Android Studio provides a built in class, which allows to set up a custom marker by taking the address that is saved in the database.

User Registration Functionalities

Sign up form

Select # of tickets

This page allows the user to enter his first/last name and select a number of tickets for the event. Once the register button is clicker, the database will update the number of attendees as well as the number of tickets remaining.

Add Friends:

This module allows the user to add friends by first sending them a request to connect. It is required to know the email of the recipient. Once this is done, the person receiving the request must login and accept/deny it. If it is accepted now both individuals will be able to share messages with each other about information of events they plan on attending to.

Megha Vashisht:

- **Modules Worked:** Event Listing, Rating the event**.**
- **Tools Used:** Android Studio, Java, Firebase database
- **Functionalities:** Make the events stored in the database display in the form of list to the user after he logs into the app.
  For the user who has attended an event can write his reviews for that event.

- **Work Done:** Created events in the firebase with all the details. Events for various categories taking place at different locations were created. These events were then displayed in the app in form of a list for the user. Whenever user logs into the app then user can see the list of events.
  Created a page where user can write the review for an event that he has visited. This review gets stored in the database. The user can change the review as well as view it
  Worked for requirement specifications and created the DFD. Also worked on documentation and presentation of the App.

Individual Module Split-up:

Prithvi Mallepula: ***20%***

- **Modules worked on:** Login page
- Worked on login page which involves entering credentials in order to access the features of the app. Alternatively, user can login with Facebook and Google + profiles. Apart from this, worked on project documentation which involves providing the description of the app in terms of software and hardware details. Also worked on curating and compiling the various DFD's in a logically fluid manner.

Venkata Saketh Ram Vemuru: ***20%***

- **Modules Worked:** User profile, Camera and Images storage, sending messages to added friends.
- **WorkDone:** Provide sample JSON data being used in the application to populate the same into the database.
  Worked to implement the camera functionality, so that users can capture and store images into their phone memory as well as the database Firebase in this

case to store the images. Also the photo captured is displayed to the user, completed it successfully. Worked on creating the firebase database to create events and the data to be loaded in the application. The images captured being shown and the events list have been crated and updated by me. Worked on creating the entire user personalization page, where the user can click on an image view to open the camera, and store the images in the firebase and phone memory. Also completed the creation of Username and Email and updated the details to the firebase backend. Worked on the custom search to retrieve the specific data from the database by writing the appropriate the queries. In the next module worked on sending messages to friends who have been accepted and share with them the ticket details and show them only the tickets after the given specific date. Also worked to generate the sample JSON data and images to populate the database for various cities used in the application.

Kumaran Manangatti Dharman Lekha: ***20%***

- **Modules Worked:** User profile, User Personalization, Heat Map
- **WorkDone:**
  - Setup the firebase database and added the events data.
  - For user category, designed the switch button to make the user on/off the personalization category
  - Designed the user profile picture for viewing past, future and today events
  - Designed the heat map page to make the user view the popular events in a particular location
  - Designed the UI for all the pages.
  - Did unit and integration test.
  - Co-ordinated with the team members to integrate the modules.
  - Styled the app with custom themes.

Jason Lescano ***20%***

Modules Worked on:

Event Information Page

User Registration

Add Friends

Event Information Functionalities:

Display event pictures

Show information about the event selected

A connection to the database was set up through the java file created. This made it possible to retrieve the information we had gathered and saved there. As for the pictures, they were uploaded to the database and were referenced by their specific destination. For the Google Maps feature, Android Studio provides a built in class, which allows to set up a custom marker by taking the address that is saved in the database.

User Registration Functionalities

Sign up form

Select # of tickets

This page allows the user to enter his first/last name and select a number of tickets for the event. Once the register button is clicker, the database will update the number of attendees as well as the number of tickets remaining.

Add Friends:

This module allows the user to add friends by first sending them a request to connect. It is required to know the email of the recipient. Once this is done, the person receiving the request must login and accept/deny it. If it is accepted now both individuals will be able to share messages with each other about information of events they plan on attending to.

<u>Megha Vashisht</u>: ***20%***

- **Modules Worked:** Event Listing, Rating the event**.**
- **Work Done:** Created events in the firebase with all the details. Events for various categories taking place at different locations were created. These events were then displayed in the app in form of a list for the user. Whenever user logs into the app then user can see the list of events.
  Created a page where user can write the review for an event that he has visited. This review gets stored in the database. The user can change the review as well as view it
  Worked for requirement specifications and created the DFD. Also worked on documentation and presentation of the App.