

Project Title: Let's Meet
(By SEMESTER – VI of III Year M.Sc. (2017-18))

Group Id: 05

Student Name: Roll No.

1. Jaishil Bhavsar: 2153005
2. Zeel Modi: 2153036
3. Bhargavi Jansari: 2153186

Name of Company: Dhruv Corporation

Date of Submission: 28th April, 2018

Submitted to:



K. S. School of Business Management

M.Sc. – Computer Applications and Information Technology

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all for our project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

We respect and thankful to K.S.S.B.M., for giving us an opportunity to do the project work and providing us all support and guidance which made us completes the project on time. We are extremely grateful to project guide for providing such a nice support and guidance though she had busy schedule.

We would like to express our gratitude towards Course coordinator Department and our project mentor for their kind co-operation and encouragement, which helped us in this project.

Group Id: - 5

Jaishil Bhavsar (3005)

Zeel Modi (3036)

Bhargavi Jansari (3186)

Index

Chapter 1: <u>Introduction</u>	5
1.1 <u>Organization Profile</u>	6
1.2 <u>System Details</u>	7
1.2.1 <u>Existing System</u>	8
1.2.2 <u>Proposed System</u>	8
1.3 <u>Scope of System</u>	9
1.4 <u>Objectives</u>	10
Chapter 2: <u>Proposed System Requirement Gathering</u>	11
2.1 <u>Stakeholder of System</u>	11
2.2 <u>Requirements Gathering Technique Used</u>	12
2.3 <u>Consolidated List of Requirements</u>	19
2.4 <u>Project Definition</u>	20
Chapter 3: <u>System Management and Planning</u>	21
3.1 <u>Feasibility Study</u>	21
3.1.1 <u>Technical</u>	21
3.1.2 <u>Economical</u>	22
3.1.3 <u>Operational</u>	22
3.2 <u>Hardware – Software Requirements</u>	23
3.3 <u>System Planning</u>	24
3.3.1 <u>Work Breakdown Structure</u>	24
3.3.2 <u>Gantt Chart</u>	25
3.4 <u>Process Model</u>	26
Chapter 4: <u>System Analysis and Design</u>	28
4.1 <u>UML [Unified Modeling Language]</u>	28
4.2 <u>System Flow Diagram</u>	35
4.3 <u>Data Dictionary</u>	36
4.4 <u>User Interface</u>	39
4.5 <u>System Navigation</u>	50
Chapter 5: <u>Input / Output Design</u>	51
Chapter 6: <u>Testing</u>	125
Chapter 7: <u>Summary</u>	135
7.1 <u>Limitations</u>	135

7.2 Future Scope.....	135
Bibliography.....	136

Chapter 1: Introduction

Let's Meet helps people do more of what they love by finding and creating communities based around the ideas and activities that matter to them. Meetups are formed around a common interest, goal, or cause, and they are made up of regular, face-to-face gatherings.

Meetups help people:

- Do what they love
- Find others and make friends
- Get involved in local communities
- Learn, teach, and share
- Rise up, stand up, unite, and make a difference
- Be part of something bigger – both locally and globally

[1.1] Organization Profile

■ Organization Name:

Dhruv Corporation.

■ Address:

23, RadhaKrishna Avenue, Opp. Golden Triangle, Stadium Road, Navrangpura, Ahmedabad-18, Gujarat, India.

■ Contact No.: 9428106009

■ Name and the Designation of Chief Functionary:

- Name: Mr. Arvind Modi
- Designation: Owner

■ About Organisation:

Dhruv Corporation is a Fabric Trading Company which provides different types of fabric. The Company's owner is Mr. Arvind Modi. He has been running from last 3 years. It has started from 2014. It has started from a little cabin to a big castle. First shop of Dhruv Corporation was in Ahmedabad where the company got all customers.

The company is having a block printers, embroidering artist and even a Kalamkari artist which provides all types of varieties on Fabrics. Company is getting its growth. They are taking customer orders and they also organise Fashion shows every year to advertise and market their different products.

They are serving their Fabric to most of all boutique in Ahmedabad and even out of town. Company is organising a little concept about gathering people and connect them. Company has planned to create events which are on one platform. It is organised around one simple idea "When we get together and do the things that matter to us we are at our best!" Their motto is the connecting bridge between user who are looking for events and event promoters who are in search of their audience.

[1.2] System Details

Project Title	: -	Let's Meet
Team Size	: -	3
Developers	: -	Jaishil Bhavsar Zeel Modi Bhargavi Jansari
Time Duration	: -	6 Months
Basic Requirements	: -	Android/iOS Device Desktop/ Laptop Internet Connection

[1.2.1] Existing System

There is no Current System. The system is working manually. The current system is time consuming because it involves a lot of paperwork and efforts.

In manual system they are calling people and informing them about the event that is about to happen. Sometimes while calling everyone is not reachable at the moment.

It will take more time and this whole procedure is very tedious and takes a lot of time.

The existing system is not as much user friendly compared to our proposed system. So people demand such a system that reduces their time and attend events easily.

[1.2.2] Proposed System

Our System provides a common platform to people to do more of what they want to do in life. It is organized around one simple idea: “when we get together and do the things that matter to us, we’re at our best.” And that’s what our system does. It brings people together to do, explore, teach and learn the things that help them come alive.

[1.3] Scope of System

- Event *recommendations* based on user's interests
- Users can get the information by following *Communities* as well as more knowledge about their field.
- Speakers and attendees can *exchange* their ideas.
- Users can get to know about seminars, webinars and *events* to be organised *nearby*.
- User can follow the *community*(groups e.g. Java, management, book clubs, adventures)
- User will be *notified* about the events via Notification and E-mails.
- *Discover* popular and recommended events near you
- Find *upcoming* events and things you want to do nearby today, this week, this weekend, or any time
- See which events your *friends* are *attending*
- Easily *share* events with your friends via WhatsApp, Hike , E-mail, Twitter, Facebook
- Can *post* in the community
- Attendees can also give their feedback and reviews about event
- It is a Progressive Web App so that user requires Internet connection only at the time of loading webpage after that user can surf website without internet.

[1.4] Objectives

- Openness to know about people in the same field
- Broader Network Opportunities
- People can learn new things
- A space where everyone can know what is happening around.
- We are the connecting bridge between users who are looking for events and event promoters who are in search of their audience.

Chapter 2: Proposed System Requirement Gathering

Let's Meet helps people do more of what they love by finding and creating communities based around the ideas and activities that matter to them. Meetups are formed around a common interest, goal, or cause, and they are made up of regular, face-to-face gatherings.

Meetups help people:

- Do what they love
- Find others and make friends
- Get involved in local communities
- Learn, teach, and share
- Rise up, stand up, unite, and make a difference
- Be part of something bigger – both locally and globally

[2.1] Stakeholder of System

Stakeholders can be defined as “anyone who has an interest in system to be built”. Stakeholders may include individuals and groups of people who have some direct interest in the event’s planning, production and implementation or in participating in. Following are the Stakeholders of our System:

- Admin
- Event Organiser
- Participants
- Community
- User

1. Admin:

- Admin is a person who is responsible for verifying and validating the events that are going to happen.
- He or she manages meet the needs of the user.

2. Event Organiser:

- He is the person who organises the event to provide people of his interest to know more about their field.

3. Participants:

- Participants can be speakers, attendees, fans, visitors who attend event to explore something.
- Without the present of participant an event is impossible so there must be participants to attend an event.

4. Community :

- Local community includes indigenous people, NGOs, network, environmental groups.
- Here user can discuss also about problems.

5. User :

- The main objective of the system is to provide information of nearby events to the user so that all the events are placed on a single platform and doesn’t have to waste time in search of events on another platforms.
- A user is a person who makes bond with the system and take benefits provided by the system.
- User can give feedback for the system.

[2.2] Requirement Gathering Technique Used

Requirements elicitation is the process to find out the requirements for a proposed system by communicating with end-users, client, system-users and other who have a stake in the system development.

There are various ways to discover requirements:

1. Interviews :

- Interviews are strong medium to collect requirements. Organizations may conduct several types of interviews such as:
 - Structured(closed) Interviews
 - Non-structured(open) Interviews
 - Oral Interviews
 - Written Interviews
 - One-to-one Interviews
 - Group Interviews

2. Surveys :

- Organizations may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.

3. Questionnaires :

- A document with pre-defined set of objectives questions and respective options is handed over to all stakeholder to answer, which are collected and then compiled.
- A shortcoming of this technique is, if an option for some issue is not mentioned in the questionnaire, the issue might be left unattended.

4. Task Analysis :

- Team of engineers and developers may analyse the operation for which the new system is required. If the client already has some software to perform certain operation, it is studied and requirements of proposed system are collected.

5. Domain Analysis :

- Every software falls into some domain category. The expert people in the domain can be a great help to analyse general and specific requirements.

6. Brainstorming :

- An informal debate is held among various stakeholders and all their inputs are recorded for further requirements analysis.

7. Prototyping :

- Prototyping is building user interface without adding detail functionality for user to interpret the features of intended software product. It helps giving better idea of requirements.
- The prototype is shown to the client and the feedback is noted. The client feedback serves as an input for requirement gathering.

8. Observation :

- Team of experts visit the client's organization or workplace. They observe the actual working of the existing installed systems. They observe the workflow at client's end and how execution problems are dealt.
- The team itself draws some conclusions which aid to form requirements expected from the software.

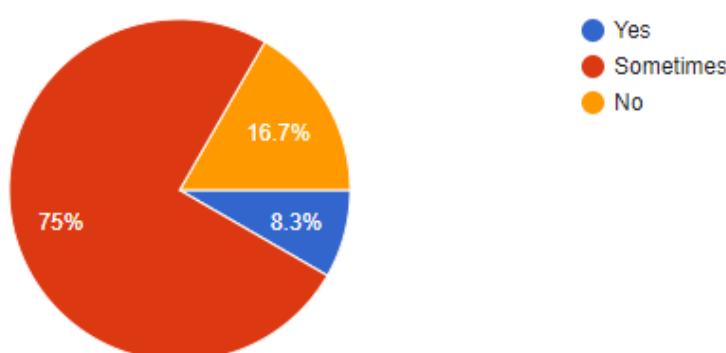
For our system we have used Questionnaire so that we can have wide range of idea of the needs of our stakeholders.

We have given different Questionnaires to different stakeholders. We want to know what they need in proposed system.

For the Users the following Questionnaire form is prepared:

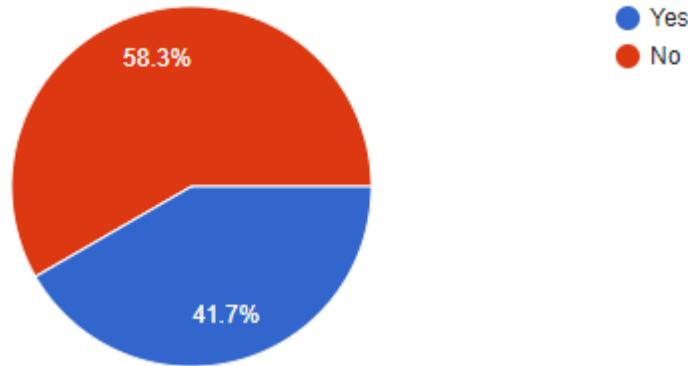
1) Do you regularly like attending Webinars, Seminars, Work-shops, Book Launches, etc.?

- Yes
- Sometimes
- No

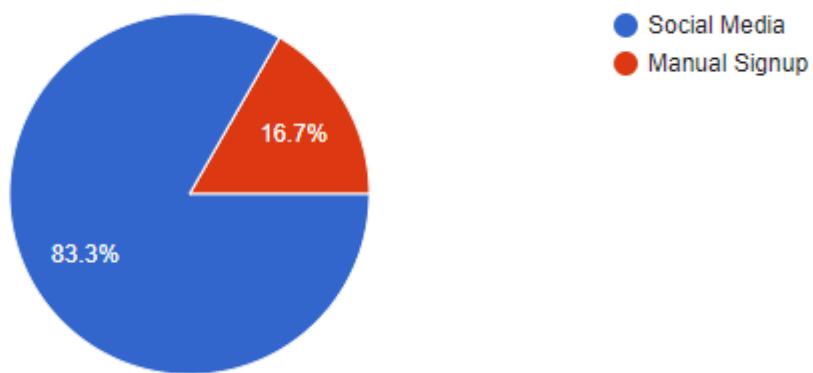


2) Are you using any system to get notified about events? If yes, then which is it?

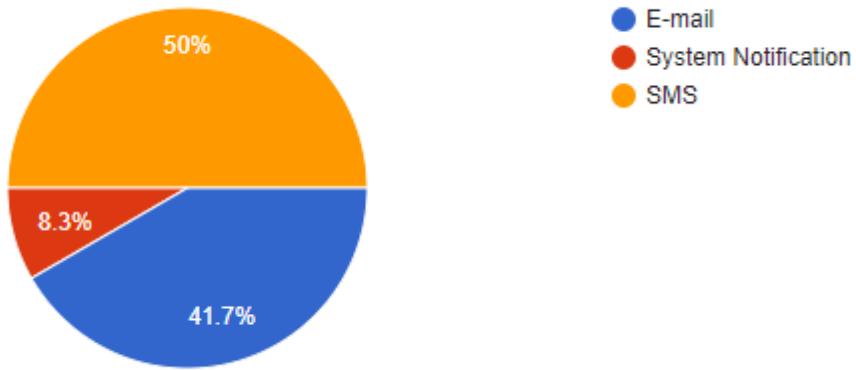
- Yes. Please specify _____
- No



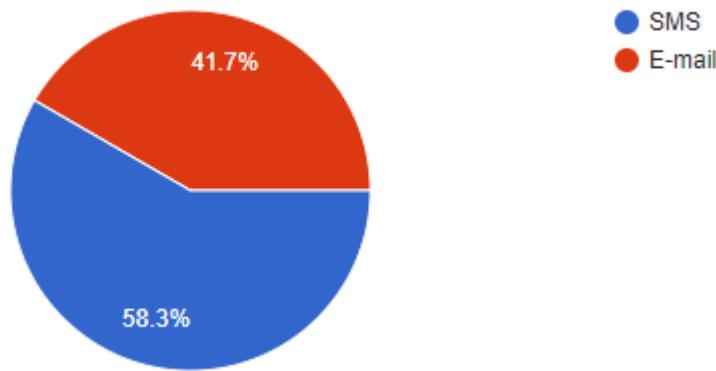
- 3) How would you like to get registered to our website?
- Social Media
 - Manual Signup



- 4) How would you like to be notified about events?
- E-mail
 - System Notification
 - SMS

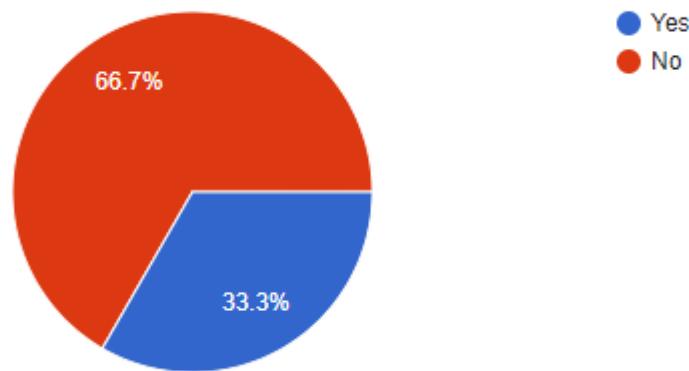


- 5) Which configuration method you prefer?
- SMS
 - E-mail



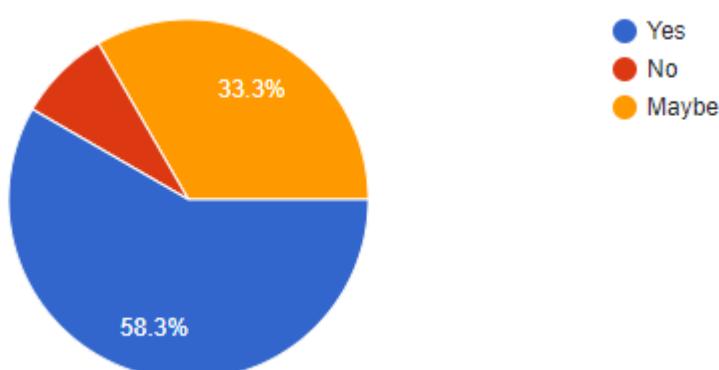
6) Do you know any event agencies? If yes then please specify.

- Yes. Please specify_____
- No



7) Would you like to use our services in future regarding events?

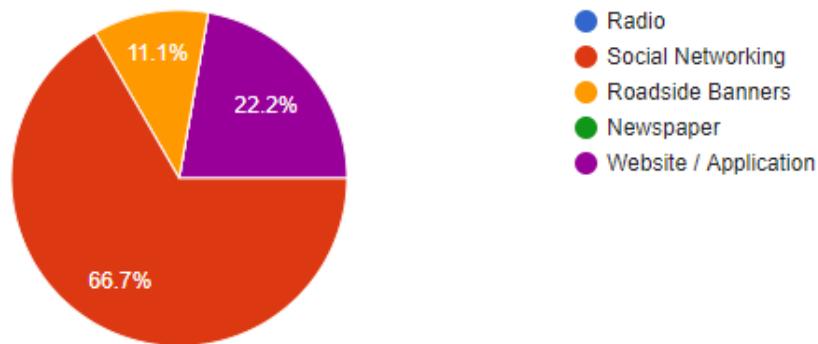
- Yes definitely
- Not at all. Please specify_____



For the Event Organisers the following Questionnaire form is prepared:

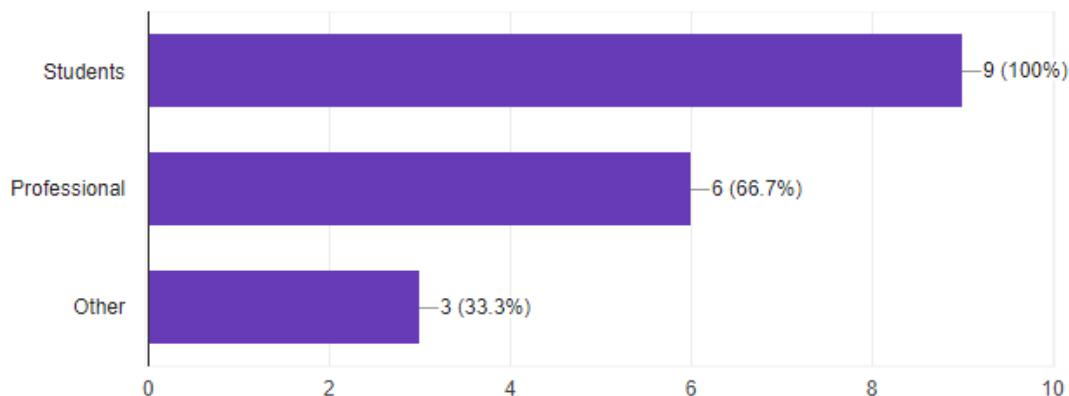
1) Normally how you promotes your events?

- Radio
- Social Networking
- Roadside Banners
- Newspapers
- Application



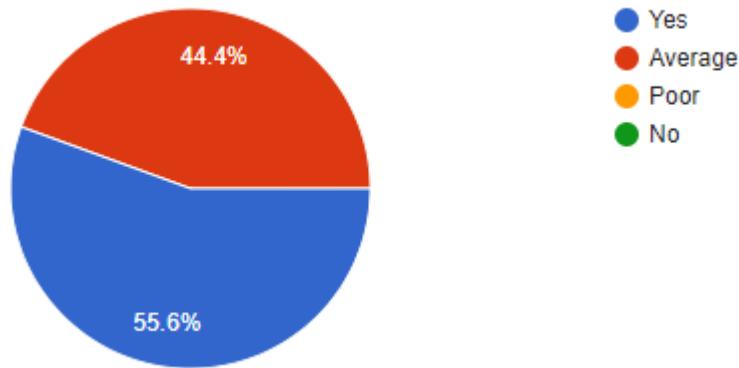
2) What kind of audience are you expecting?

- Professional
- Student
- Others. Please Specify_____



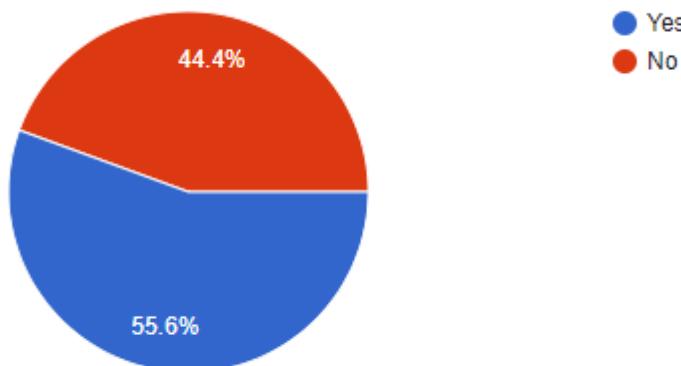
3) Do you get a good amount of audience for your event?

- Yes
- Average
- Poor
- No



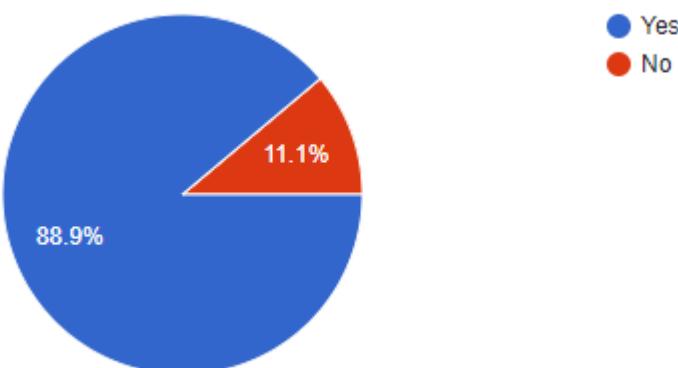
4) Will there be tickets for events?

- Yes
- No



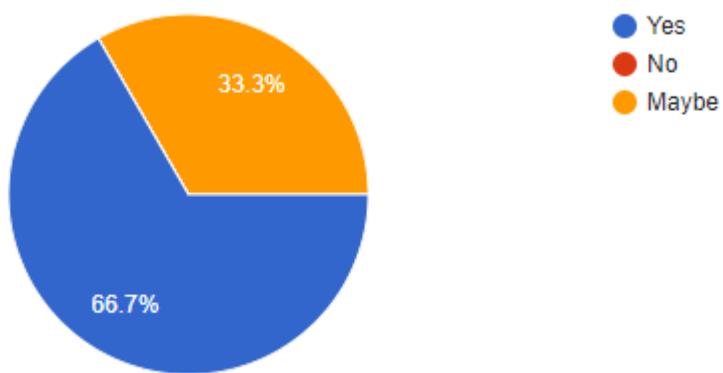
5) Do you take feedback for your events?

- Yes
- No



6) Would you like to promote your events by our system?

- Yes
- Maybe
- No. Please specify reason _____



[2.3] Consolidated List of Requirement

By taking Questionnaire we have decided to make following change in our system:

Users:

- Users need to login from Social Media rather than manual Signup, which saves their time and effort for remaining passwords.
- They wants to be notified by application notifications,
- They prefer SMS authentication over E-mails.
- They need all the types of event happening in city on a single platform.

Event Organisers:

- Organisers need a platform where they can attract a good amount of audience for their events.
- They need a feedback about their event so that they can improve it in future.

[2.4] Project Definition

Hundreds of events happens in our city, but it's difficult to find them out. 'Let's Meet' helps you do that..!!Our System provides a common platform to people to do more of what they want to do in life. It is organized around one simple idea: "when we get together and do the things that matter to us, we're at our best." And that's what our system does. It brings people together to do, explore, teach and learn the things that help them come alive.

Chapter 3: System Management and Planning

Let's Meet helps people do more of what they love by finding and creating communities based around the ideas and activities that matter to them. Meetups are formed around a common interest, goal, or cause, and they are made up of regular, face-to-face gatherings.

Meetups help people:

- Do what they love
- Find others and make friends
- Get involved in local communities
- Learn, teach, and share
- Rise up, stand up, unite, and make a difference
- Be part of something bigger – both locally and globally

[3.1] Feasibility Study

A feasibility study is carried out to select the best system that meets performance requirements.

The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behaviour of the system.

[3.1.1] Technical Feasibility:

This is concerned with specifying the equipments and the software to satisfy the user requirements. The technical needs of the system vary considerably but might include:

The facility to produce outputs in a given time.

Response time under certain conditions.

Ability to process a certain volume of transactions at a specified speed.

Facility to communicate data to a distant location.

Technical feasibility centres on the existing computer system, hardware, software etcetera and to what extent it can support the system. In examining the technical feasibility, the configuration of the system is given more importance than the actual hardware.

Our system can be run on any mobile platform including Android, iOS and Windows. So that we can easily say that our system is technically feasible.

[3.1.2] Economic Feasibility:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. The system is economical feasible because:

There is no extra economical cost because system is developed with an open source technology.

Our system is not much costly to develop.

Organisation is ready to invest in proposed system because it is being developed in latest technology.

[3.1.3] Operational Feasibility:

Operational feasibility study tests the operational scope of the software to be developed. The proposed software must have high operational feasibility. The usability will be high. Operation of the proposed system depend on its various users. Various user-types of Users are mentioned below:

i. **Admin:**

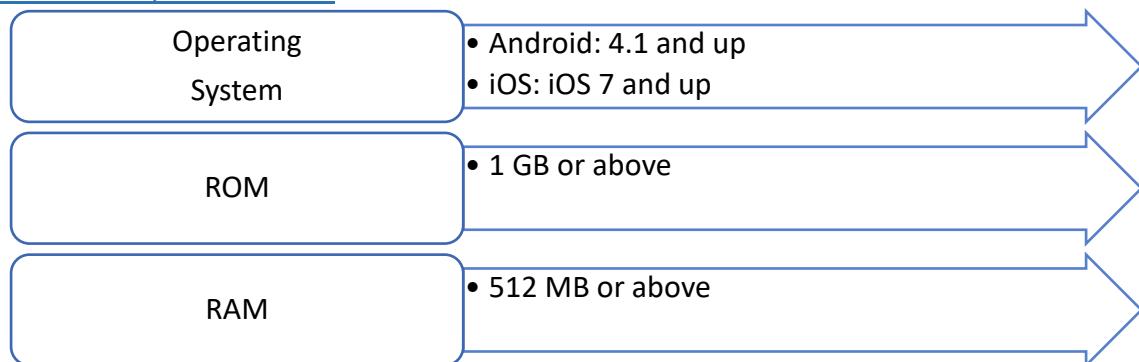
Admin will authenticate events and users. If any of these are invalid admin have rights to remove them from using system. Admin can make any change at a time.

ii. **Normal User:**

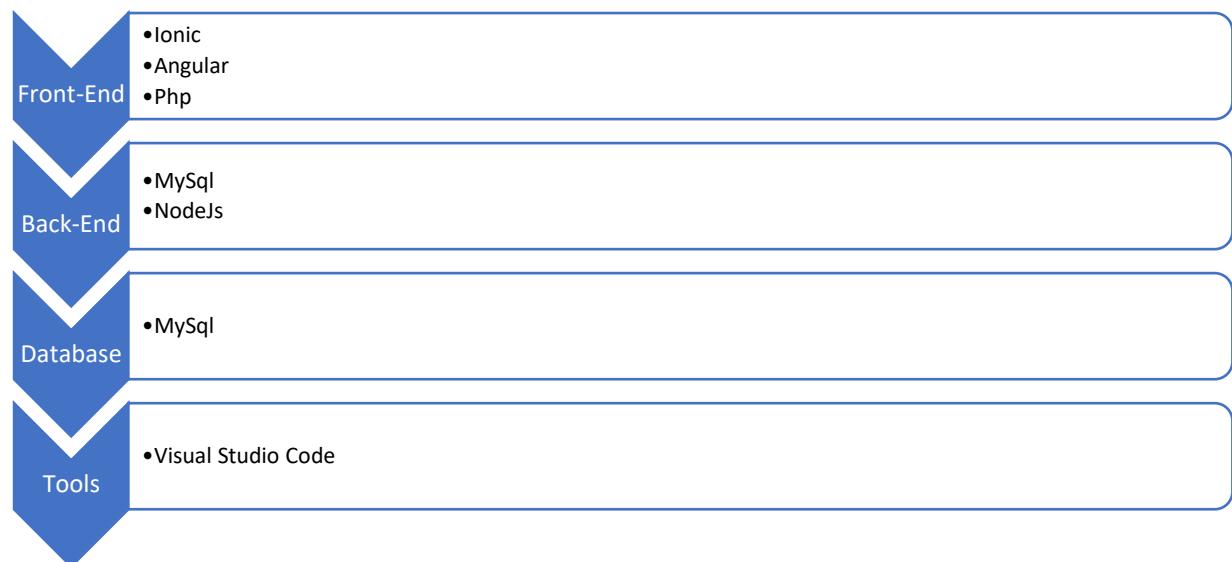
Users will be able to do sign-up, login, feedback any event. Interface of the system is so easy that users don't have to go anywhere for training.

[3.2] Hardware – Software Requirement

Hardware Requirements:

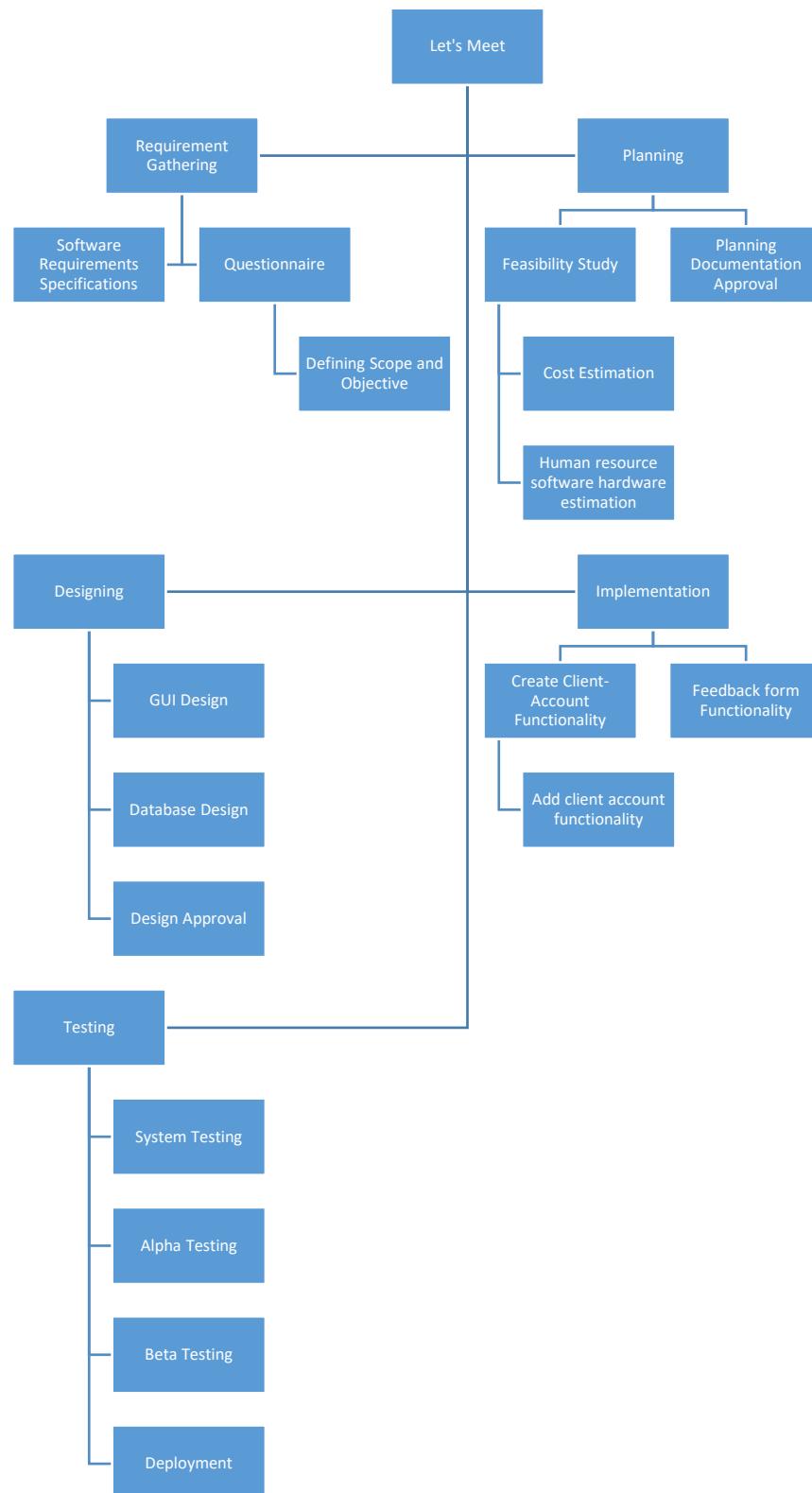


Software Requirements:



[3.3] System Planning

[3.3.1] Work Breakdown Structure



[3.3.2] Gantt Chart



[3.4] Process Model

A software process model is a standardised format for

- planning
- organising, and
- running

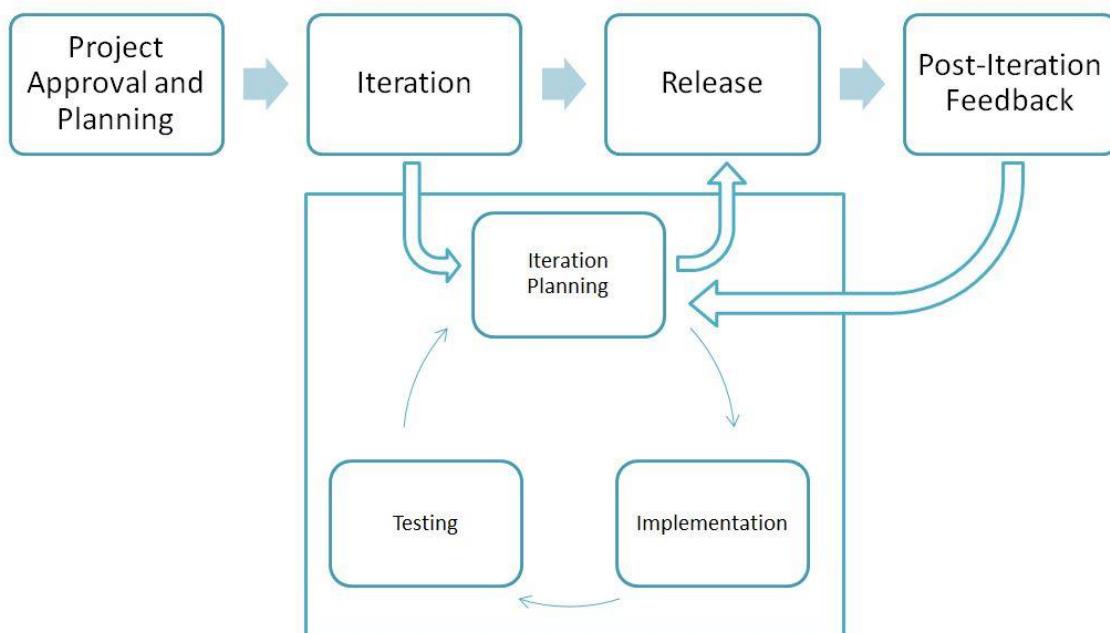
a development project. A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed.

What is Agile Model?

The Agile methodology (it's actually a movement, not a methodology) is essentially a list of principles that advocates self-organizing teams, adaptive planning, early delivery, and continuous improvement.

Agile methodologies focus on iterations in which planning, design, implementation and testing occur in short periods of time. Agile methodology allows planning to occur throughout the project lifecycle, thus allowing decisions to be reactive. In software development, bugs can be caught early and remediated before they grow to become bigger problems.

Here's the agile process for a generic project:



The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Agile method proposes incremental and iterative approach to software design.

Why to Use Agile Model?

- It is flexible and less costly to change scope and requirements.
- Generates working software quickly and early during the software life cycle.
- We can get our customer responded and can change according to their requirements. Rather than other model here customers have exact idea about their proposed system. In other models customers get their system at last so we can't know whether our customer is satisfied or not. Here we're constantly in touch with customer. This model provides higher customer satisfaction.
- Error can be fixed in the middle of the project.

Chapter 4: System Analysis and Design

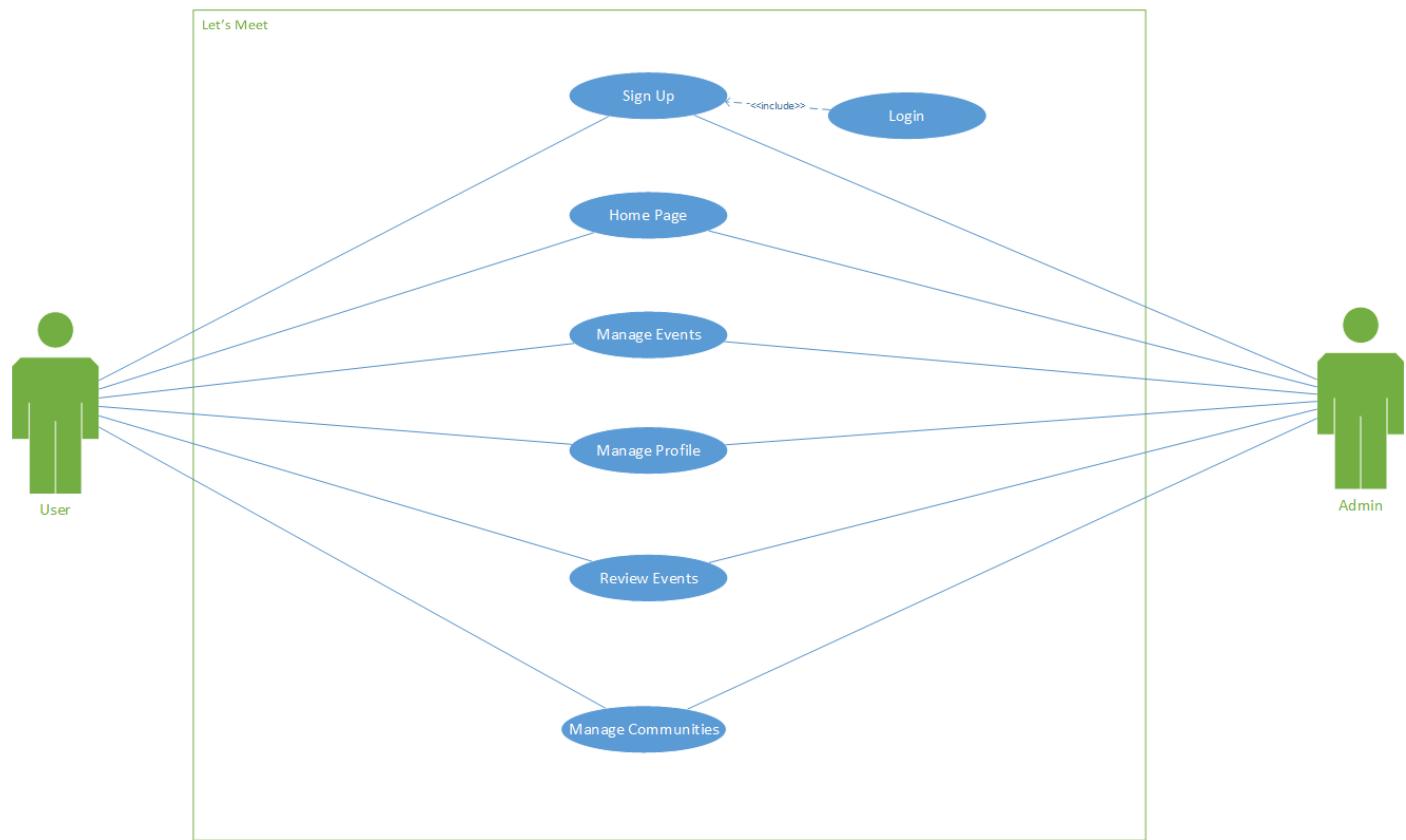
Let's Meet helps people do more of what they love by finding and creating communities based around the ideas and activities that matter to them. Meetups are formed around a common interest, goal, or cause, and they are made up of regular, face-to-face gatherings.

Meetups help people:

- Do what they love
- Find others and make friends
- Get involved in local communities
- Learn, teach, and share
- Rise up, stand up, unite, and make a difference
- Be part of something bigger – both locally and globally

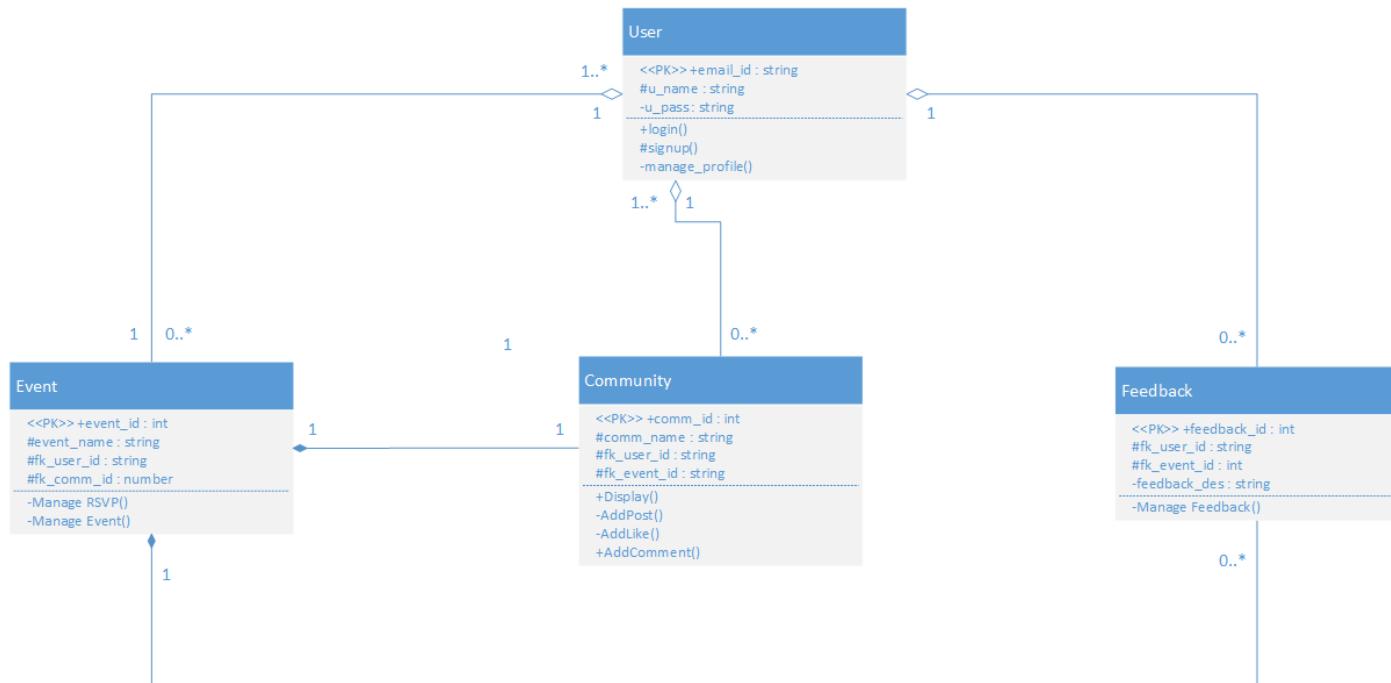
[4.1] UML (Unified Modeling Language)

[4.4.1] Use Case



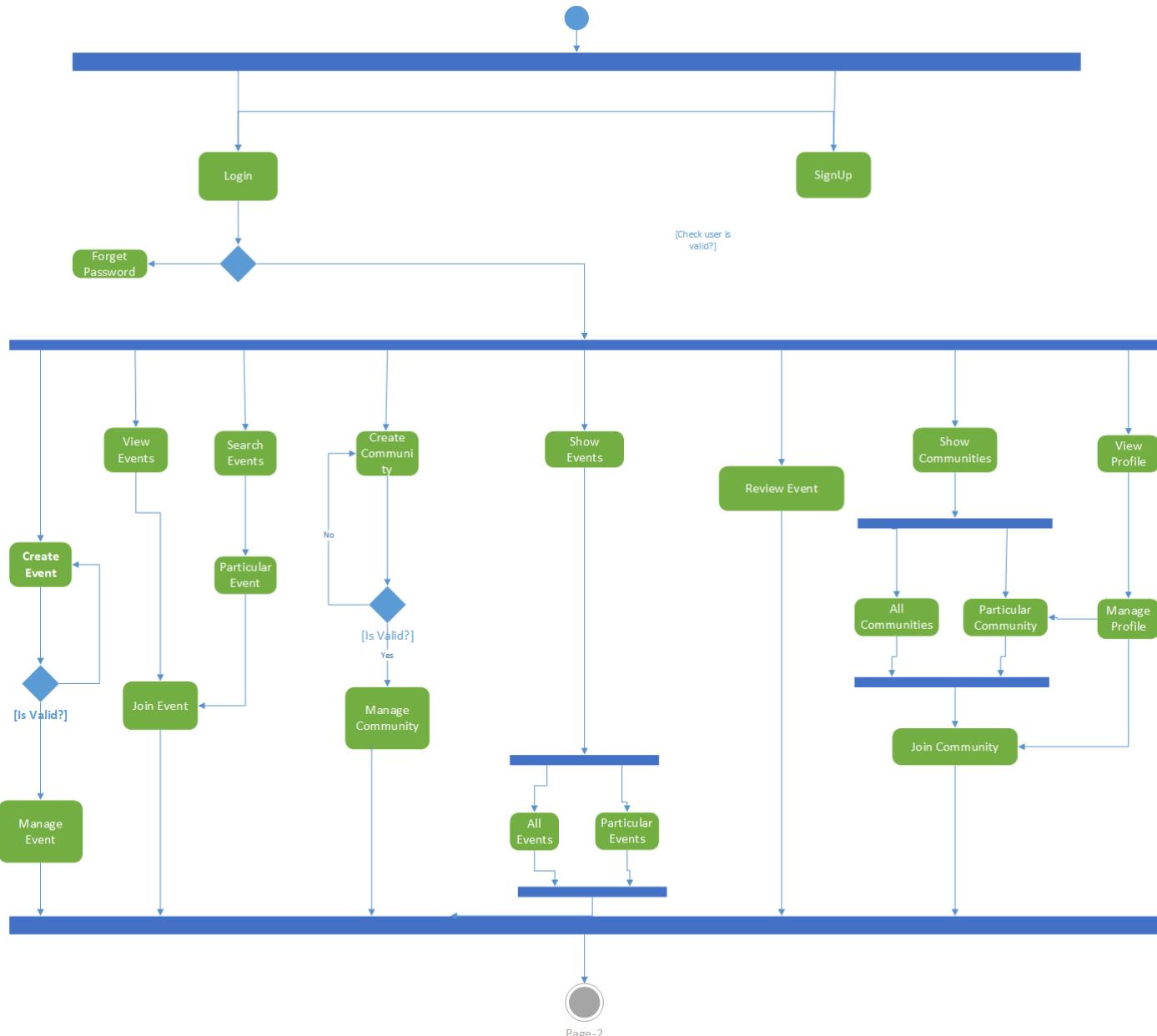
(fig 4.1 – Use Case of System)

[4.4.2] Class Diagram

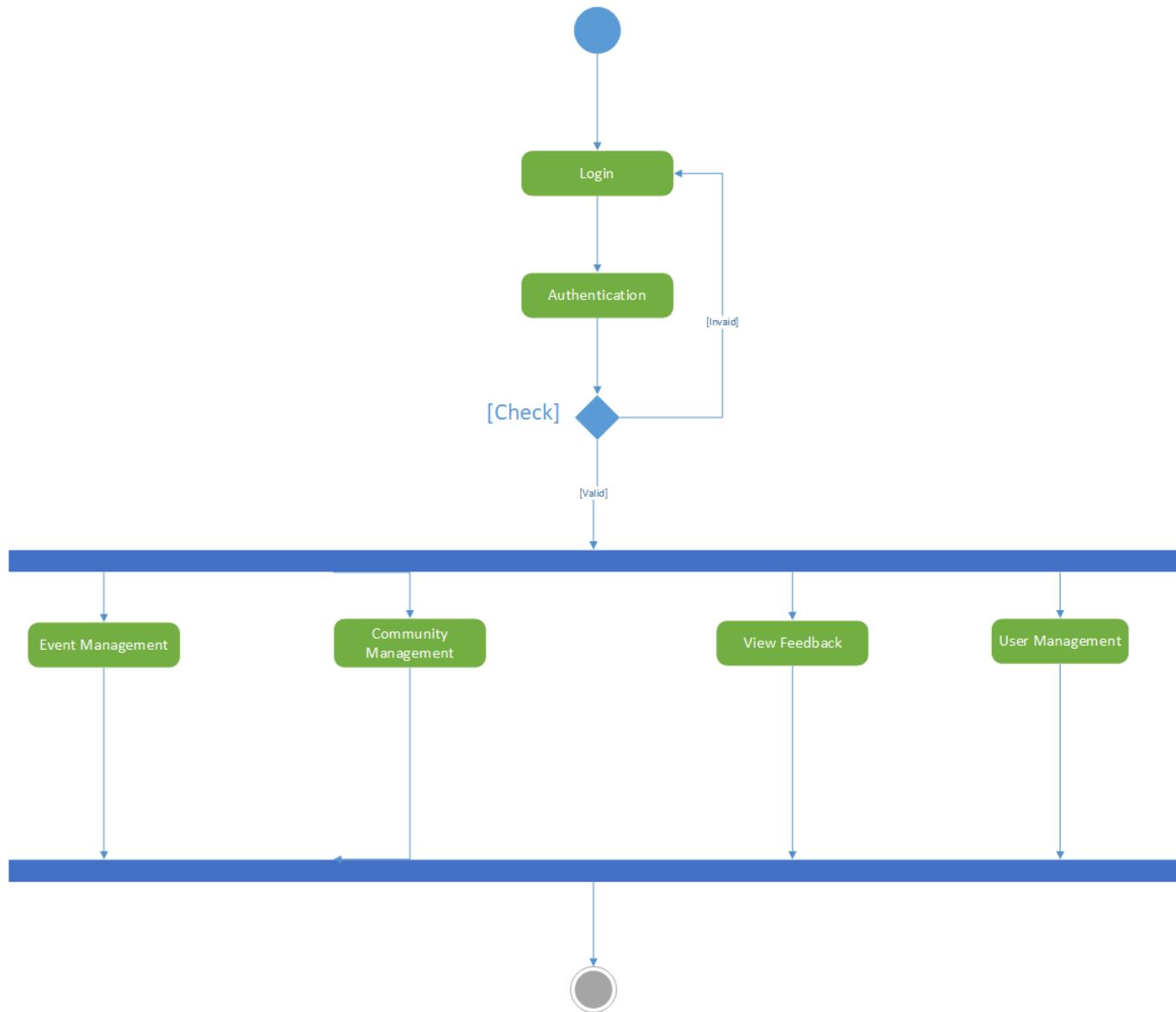


(fig 4.2 – Class Diagram)

[4.4.3] Activity Diagram

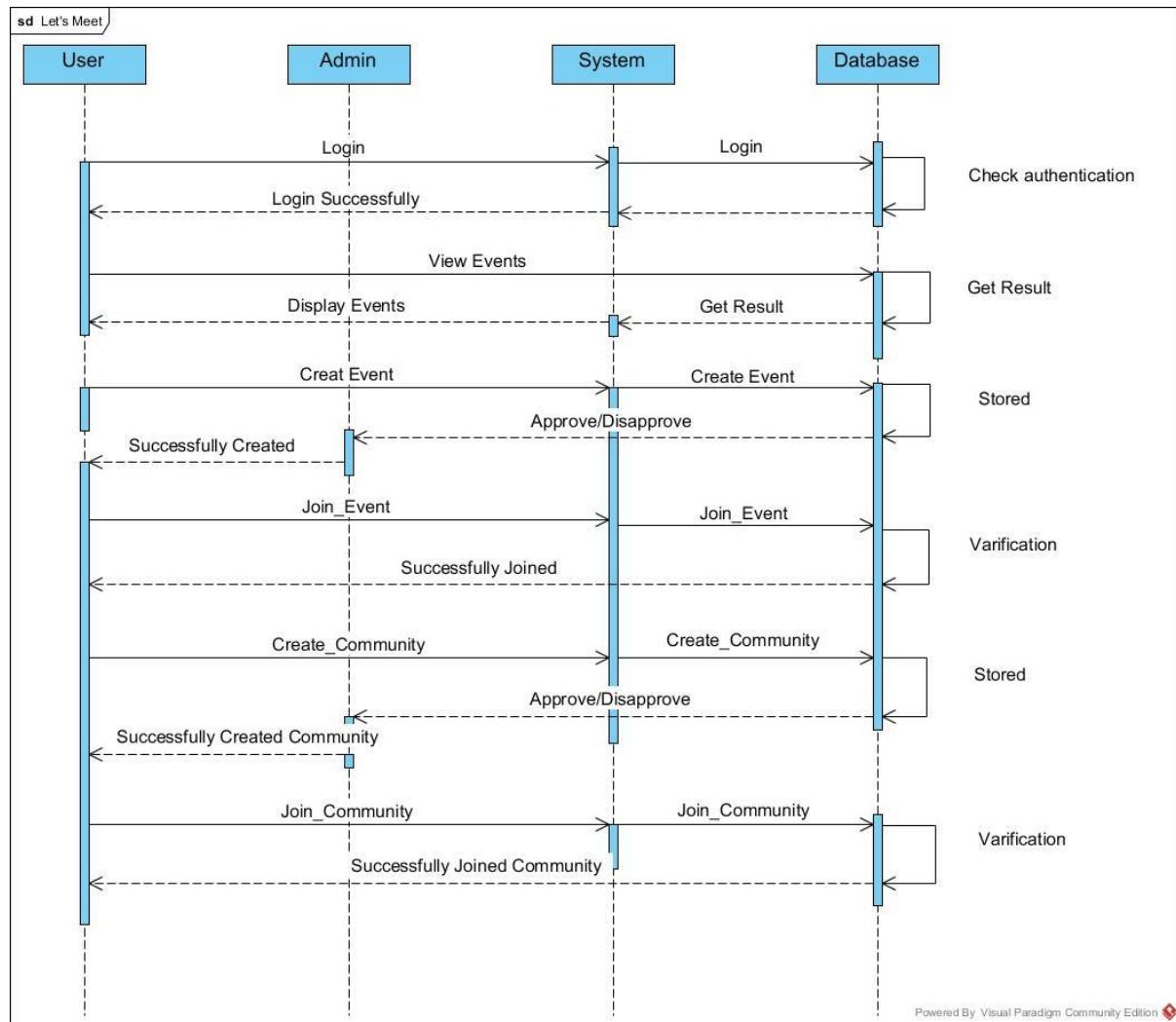


(fig 4.3 – Activity Diagram for User)

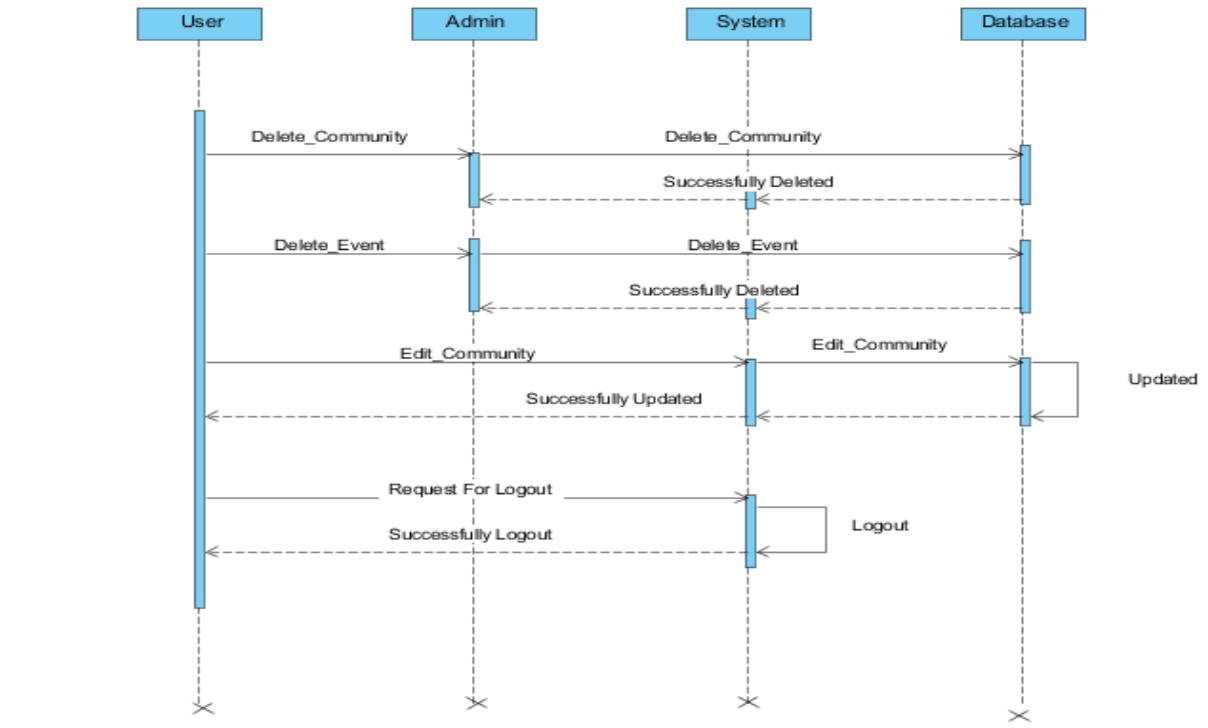


(Fig 4.4 – Activity Diagram for Admin)

[4.4.4] Sequence Diagram

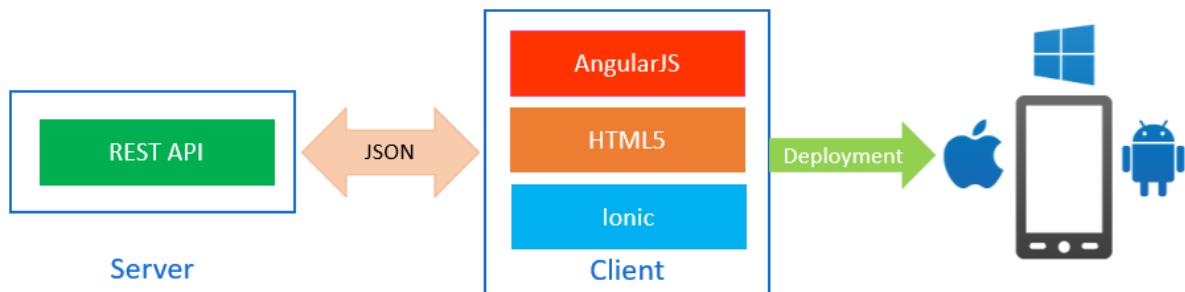


(Fig 4.5 – Sequence Diagram)



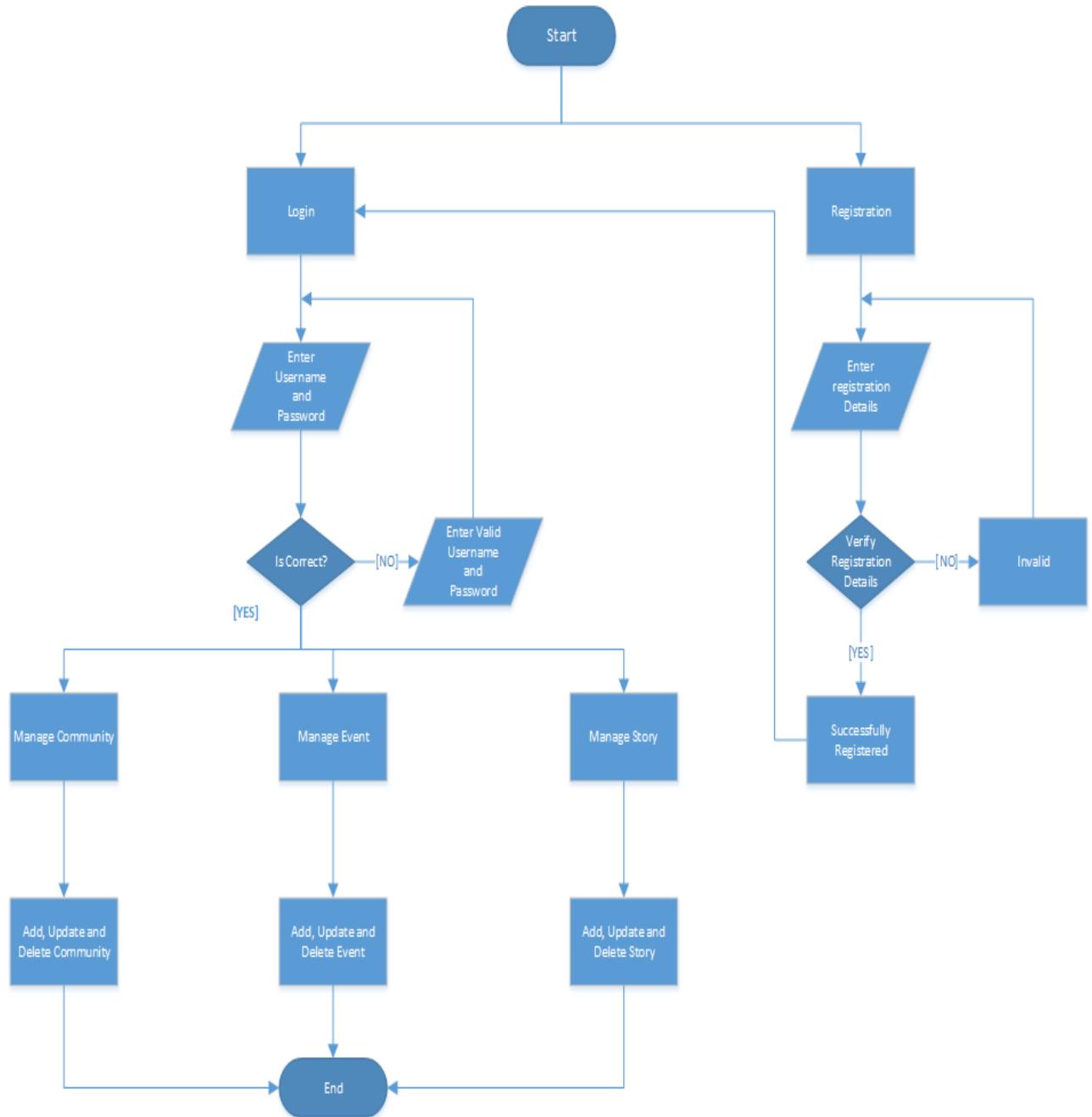
(Fig 4.6 – Sequence Diagram)

[4.4.5] Deployment Diagram



(Fig 4.7 – Deployment Diagram)

[4.2] System Flow Diagram



(Fig 4.8 – System Flow Diagram)

[4.3] Data Dictionary

Users

Field	Data-Type	Constrains	Remarks	Description
User_id	Varchar(30)	Primary Key	-	User's email id.
User_name	Varchar(20)	Not Null	-	Display User's Name
User_photo	Varchar(500)	-	-	Display Picture
Gender	Boolean	-	-	User's Gender
Password	Varchar(10)	Not Null	-	Password for Login
Mobile_no	Varchar(15)	-	-	User's Mobile No
Birth_date	Date	-	-	User's Birth Date

Communities

Field	Data-Type	Constrains	Remarks	Description
Comm_id	Integer	Primary Key	Auto-Increment	Community id
Comm_name	Varchar(20)	Not Null	-	Community Name
Comm_des	Varchar(300)	Not Null	-	Describe the community
Comm_photo	Varchar(500)	-	-	Community Picture
Date	Date	Not Null	-	Date created
Created_by	Varchar(50)	Foreign Key	-	Users created Community
Comm_rating	Integer		-	Rating of Community

Events

Field	Data-Type	Constrains	Remarks	Description
Event_id	Integer	Primary Key	Auto-Increment	Event's Id
Event_name	Varchar(100)	Not Null	-	Event's Name
Event_pic	Varchar(500)	-	-	Event Photo
Event_des	Varchar(500)	-	-	Event Description
Event_time	Varchar(20)	-	-	Event Time
Event_date	Date	Not Null	-	Event Date
Event_loc	Varchar(100)		-	Event Venue
Fk_user_id	Varchar(30)	Foreign Key	-	Who Created?

Fk_comm_id	Integer	Foreign Key	-	Under Which Community?
-------------------	---------	-------------	---	------------------------

Community Members

Field	Data-Type	Constrains	Remarks	Description
Join_id	Integer	Primary Key	Auto-Increment	Community join Id
Fk_user_id	Varchar(30)	Foreign Key	-	User who joined community
Fk_comm_id	Integer	Foreign Key	-	Which community did user joined

Feedbacks

Field	Data-Type	Constrains	Remarks	Description
Feed_id	Integer	Primary Key	Auto-Increment	Feedback id
Fk_event_id	Integer	Foreign Key	-	Under which Event
Fk_user_id	Varchar(30)	Foreign Key	-	Who gave feedback?
Feed_desc	Varchar(200)	Not Null	-	Feedback Description.

Posts

Field	Data-Type	Constrains	Remarks	Description
Post_id	Integer	Primary Key	Auto-Increment	Post Id
Post_title	Varchar(30)	Not Null	-	Caption
Post_des	Varchar(200)	-	-	Describe the Post
Post_pic	Varchar(500)	-	-	Photo in Post
Fk_user_id	Varchar(30)	Foreign Key	-	User who posted this Post
Fk_comm_id	Integer	Foreign Key	-	Under which Community
Date	Date	Not Null	-	Date Created Post.

Likes

Field	Data-Type	Constrains	Remarks	Description
Like_id	Integer	Primary Key	Auto-Increment	Like id
Fk_post_id	Integer	Foreign Key	-	Under which Post
Fk_user_id	Varchar(30)	Foreign Key	-	Who liked?

Comments

Field	Data-Type	Constrains	Remarks	Description
Com_Id	Integer	Primary Key	Auto-Increment	Comment id
Fk_post_id	Integer	Foreign Key	-	Under which post
Fk_user_id	Varchar(30)	Foreign Key	-	Who commented?
Date	Date	Not Null	-	Date Created
Com_des	Varchar(300)	Not Null	-	Content of Comment

RSVP

Field	Data-Type	Constrains	Remarks	Description
Rsvp_id	Integer	Primary Key	Auto-Increment	RSVP id
Fk_user_id	Varchar(30)	Foreign Key	-	Users attending event
Fk_event_id	Varchar(50)	Foreign Key	-	Which event user is going to attend?

Follower

Field	Data-Type	Constrains	Remarks	Description
Follo_id	Integer	Primary Key	Auto-Increment	Follower Id
Fk_user_id	Varchar(30)	Foreign Key	-	User being followed
Fk_us_id	Varchar(30)	Foreign Key	-	Who Followed?

Categories

Field	Data-Type	Constrains	Remarks	Description
Cat_id	Integer	Primary Key	Auto-Increment	Category Id
Cat_name	Varchar(20)	Not Null	-	Category Name

[4.4] User Interface

Tutorial

The image shows a tutorial screen with two panels. The left panel, titled "Discover Events Happening in Your City", features the "Let's Meet" logo at the top, followed by five small circular progress indicators at the bottom. The right panel, titled "Ready to Play?", features a large blue star wand with a yellow star at its tip, surrounded by sparkles, and five small circular progress indicators at the bottom.

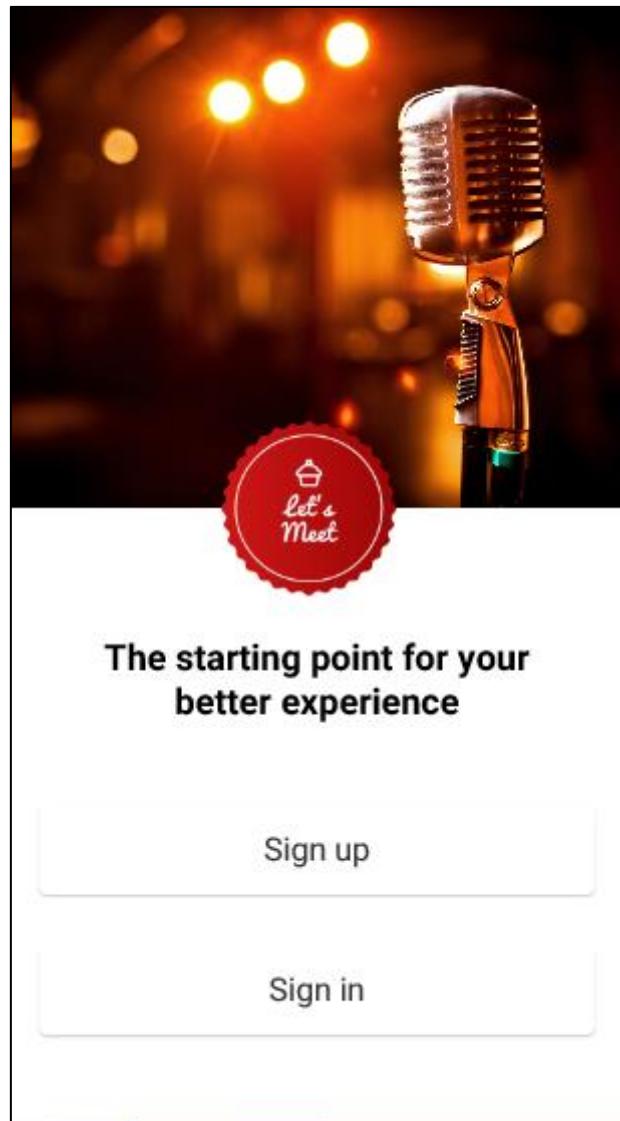
Skip

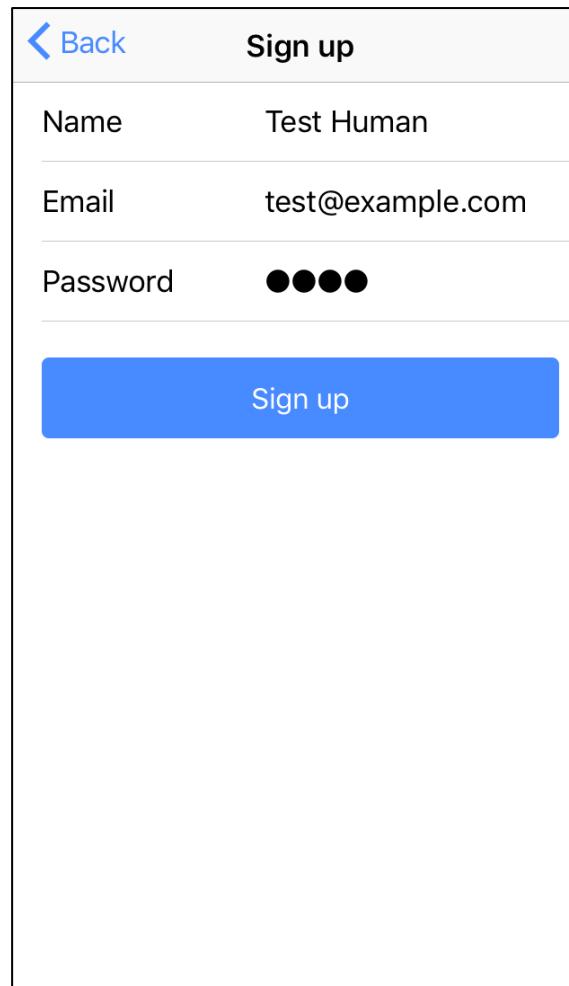
Let's Meet

Discover Events Happening in Your City

Ready to Play?

Continue >

Start-up

Sign Up Page

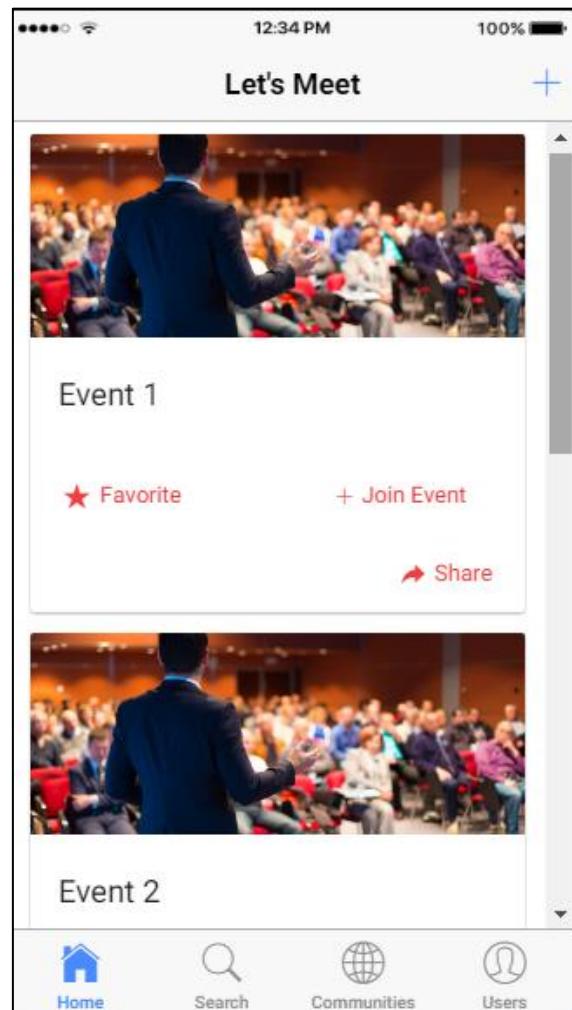
The image shows a mobile application's sign-up screen. At the top left is a blue 'Back' button with a white arrow pointing left. To its right is the title 'Sign up' in bold black font. Below the title are three input fields: 'Name' with the value 'Test Human', 'Email' with the value 'test@example.com', and 'Password' represented by four black dots. At the bottom is a large blue rectangular button with the text 'Sign up' in white.

Name	Test Human
Email	test@example.com
Password	●●●●

Sign up

Sign In Page

 Back	Sign in
Email	test@example.com
Password	●●●●
Sign in	

View Events Tab

Create Event Page

Cancel **Create Event** Done

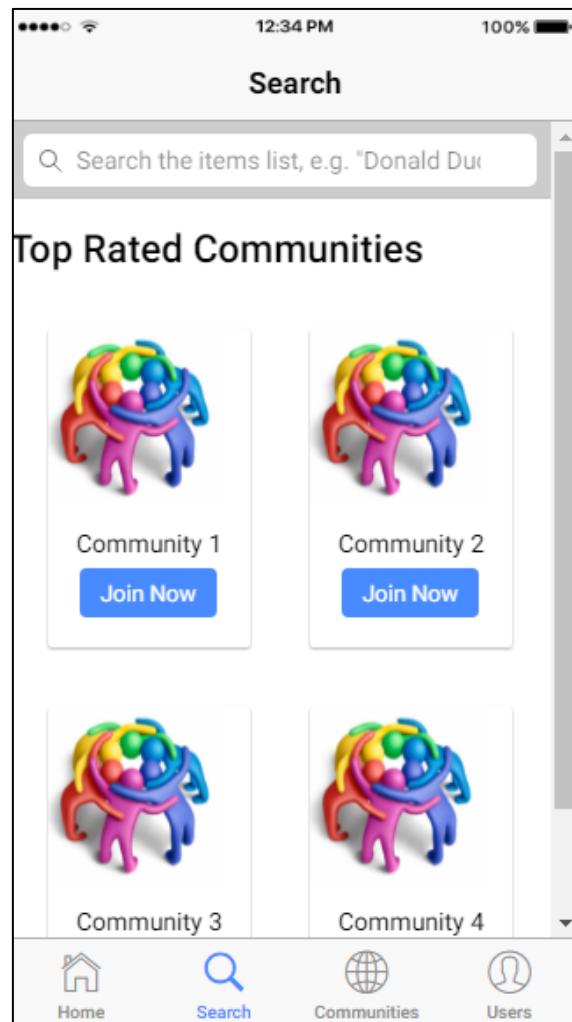
 Add Image

Event Name

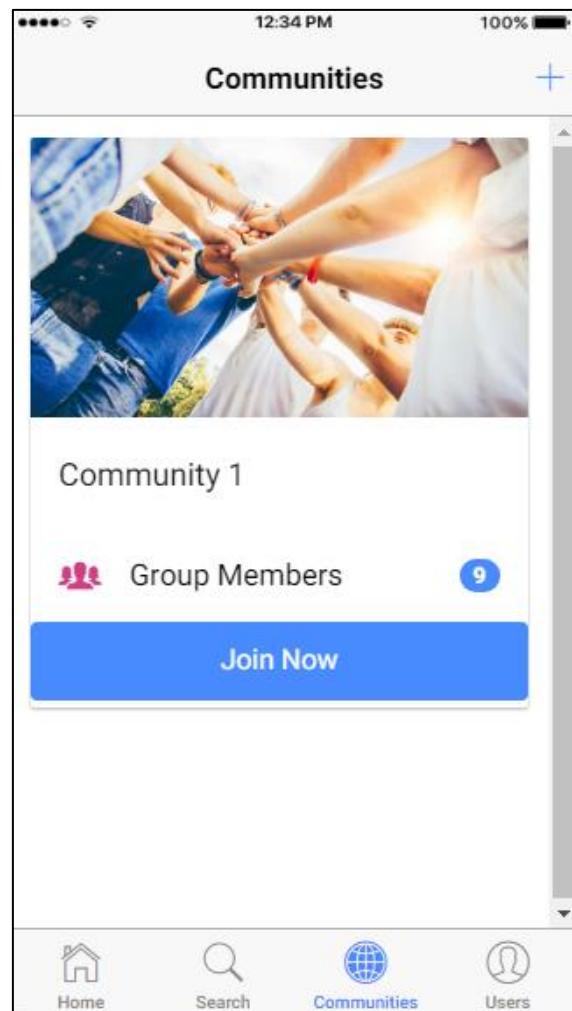
Event Description

Event Category : ▾

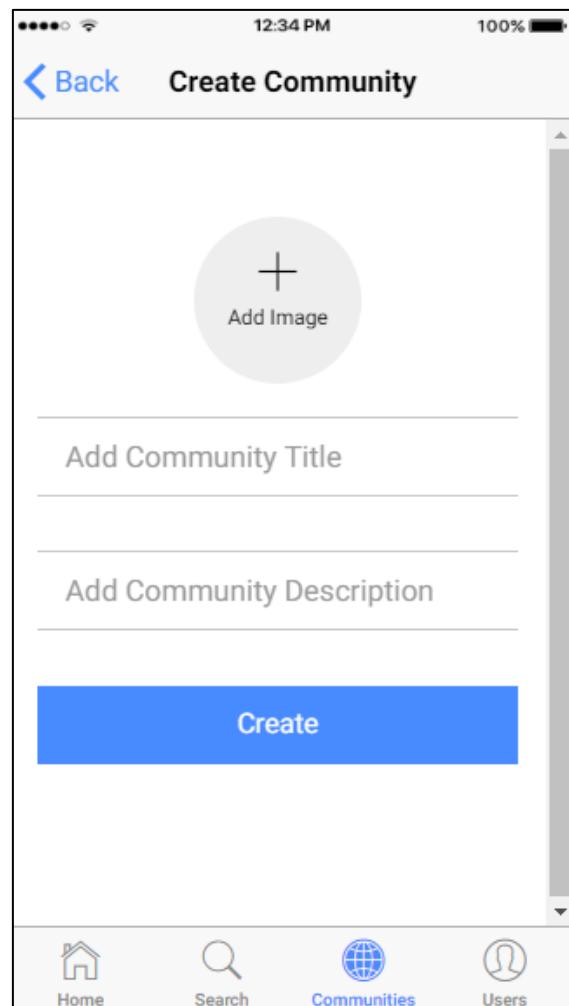
Search Tab

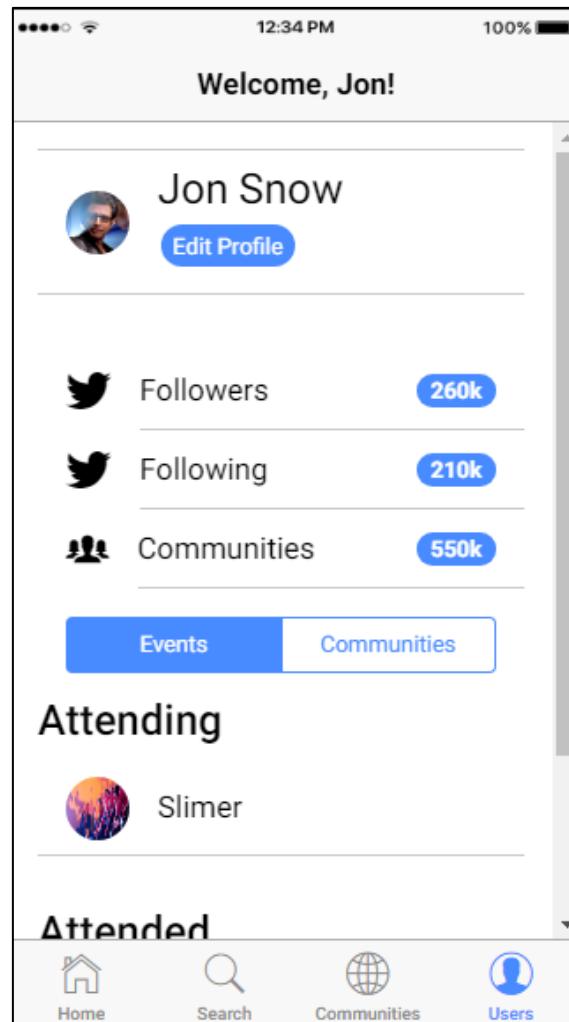


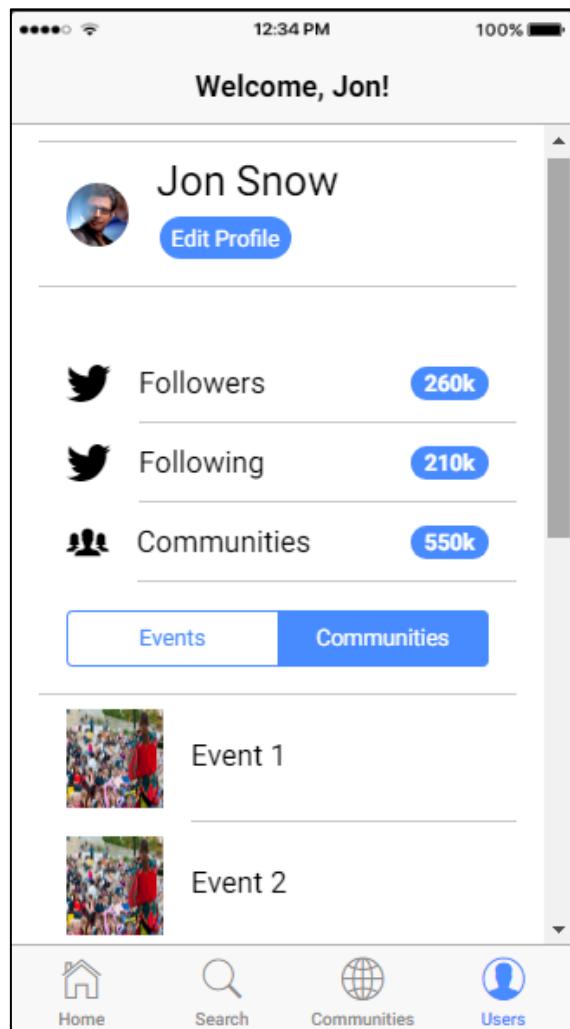
Communities Tab



Create Community Page



User Profile Tab (Events)

User Profile Tab (Communities)

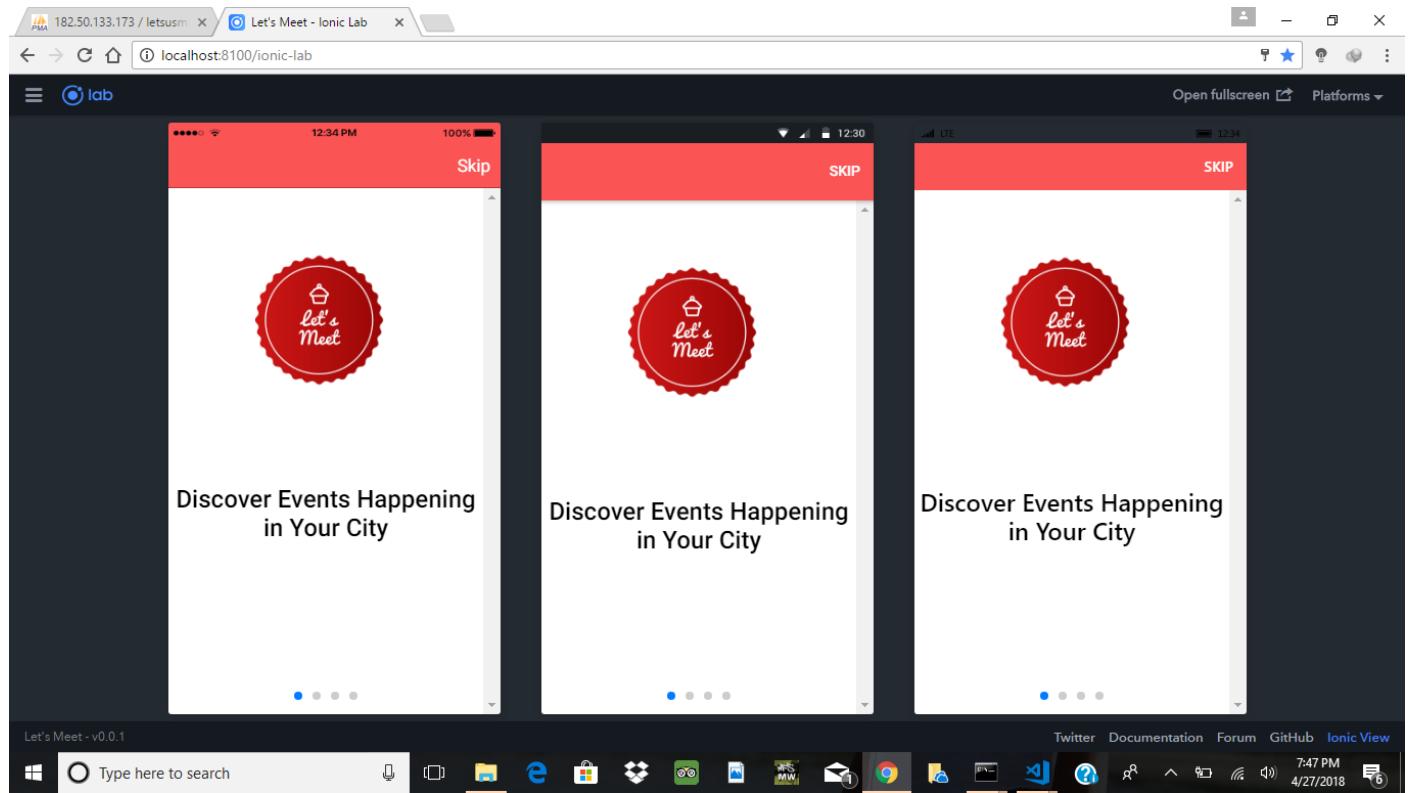
[4.5] System Navigation



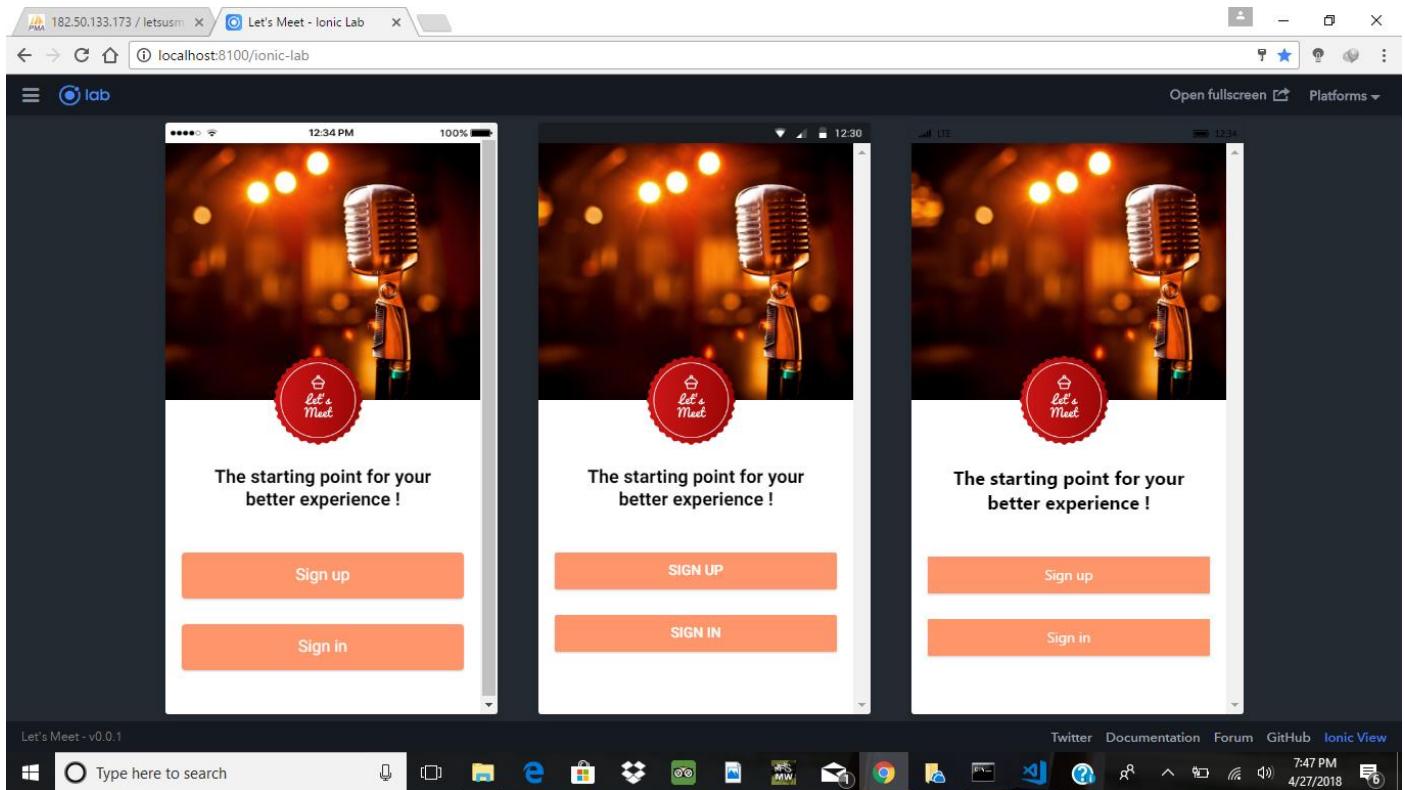
Chapter 5: Input / Output Design

User

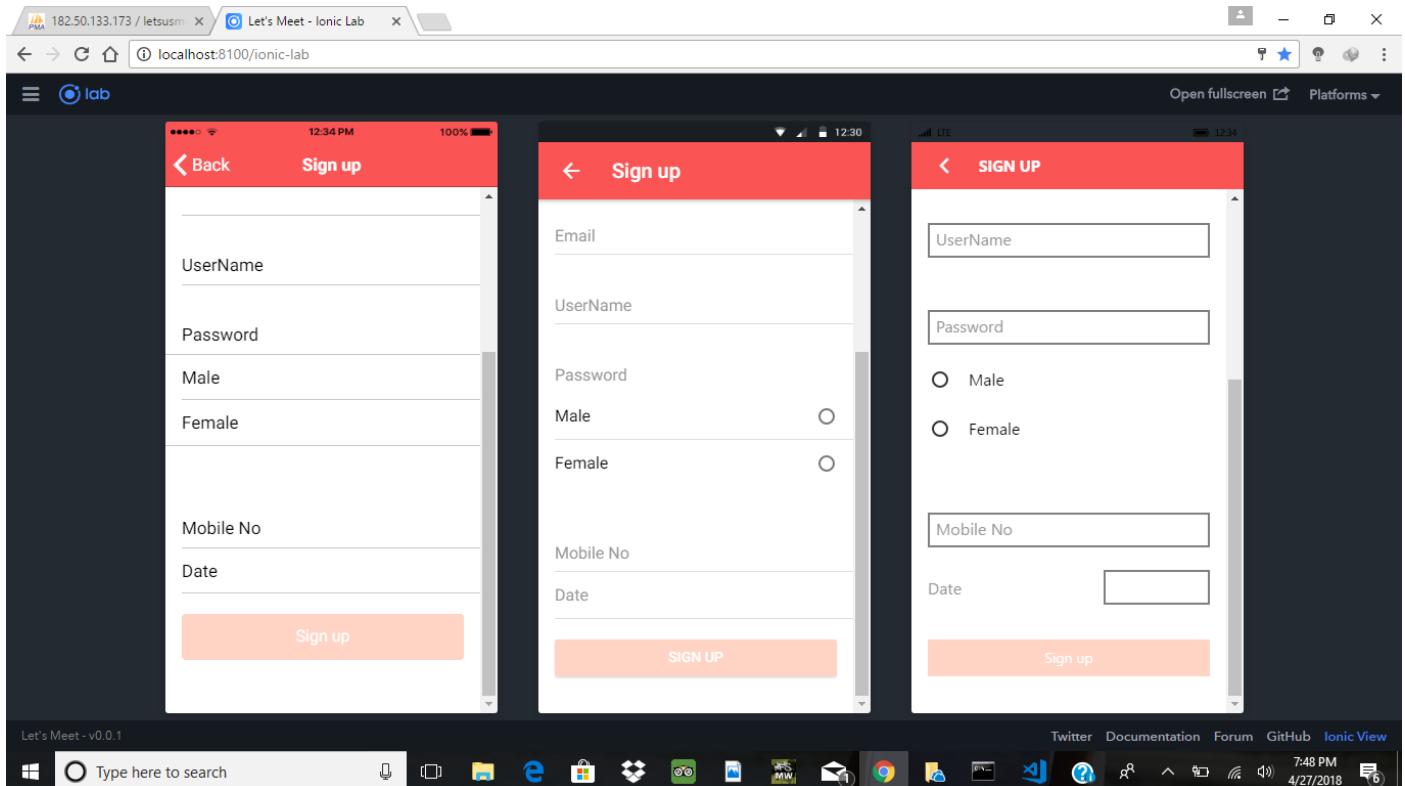
Tutorial



Start-up



Sign Up



Service of Sign Up

```
var db = require('../dbconnection');
```

```

var User = {

    getAllUsers: function (callback) {
        return db.query("select * from user_tbl", callback);
    },
    getUserId: function (id, callback) {
        return db.query("select * from user_tbl where user_id=?", [id], callback);
    },
    addUser: function (u, filename, callback) {
        console.log(u.user_id);
        console.log(u.user_pass);
        console.log(u.gender);
        console.log(u.user_name);
        console.log(u.user_mob_no);
        //console.log(u.user_pic);
        console.log(u.user_bdate);
        console.log(u.token);
        return db.query("insert into
user_tbl(user_id,user_name,user_pass,user_pic,gender,user_mob_no,user_bdate,token)
values(?,?,?,?,?,?)", [u.user_id, u.user_name, u.user_pass, filename, u.gender,
u.user_mob_no, u.user_bdate,u.token], callback);
    },
    deleteUser: function (id, callback) {
        return db.query("delete from user_tbl where user_id=?", [id], callback);
    },
    updateUser: function (u, callback) {
        console.log(u.user_id);
        console.log(u.user_name);
        return db.query("update user_tbl set
user_name=?,user_pic=?,gender=?,user_mob_no=?,user_bdate=?,token=? where user_id=?",
[u.user_name, u.user_pic, u.gender, u.user_mob_no, u.user_bdate,u.token, u.user_id],
callback);
    }
};

module.exports = User;

```

```

public translateService: TranslateService) {
  this.form = formBuilder.group({
    eid: ['', Validators.required],
    uname: ['', Validators.compose([Validators.required, Validators.pattern('[a-zA-Z]*'))]),
    pass: ['', Validators.compose([Validators.minLength(6), Validators.maxLength(10),
      Validators.required])],
    gender: ['', Validators.required],
    mobile: ['', Validators.required],
    myDate: ['', Validators.required],
    image: ['', Validators.required]
  });

  this.form.valueChanges.subscribe((v) => {
    this.isReadyToSave = this.form.valid;
  });

  this.translateService.get('SIGNUP_ERROR').subscribe((value) => {
    this.signupErrorString = value;
  })
}

eid: string = "";
uname: string = "";
pass: string = "";
gender: string = "";
mobile: string = "";
myDate: any;
image: any;
token: string = "user";

getPicture() {
  if (Camera['installed']()) {
    this.camera.getPicture({
      destinationType: this.camera.DestinationType.DATA_URL,
      targetWidth: 96,
      targetHeight: 96
    }).then((data) => {
      this.form.patchValue({ 'image': 'data:image/jpg;base64,' + data });
    }, (err) => {
      alert('Unable to take photo');
    })
  } else {
    this.fileInput.nativeElement.click();
  }
}

processWebImage(event) {
  let reader = new FileReader();
  reader.onload = (readerEvent) => {

    let imageData = (readerEvent.target as any).result;
    this.form.patchValue({ 'image': imageData });
  }
}

```

```

};

reader.readAsDataURL(event.target.files[0]);
this.selectedFile = <File>event.target.files[0];
console.log(this.selectedFile);
//alert(this.selectedFile.type);
if (this.selectedFile.type != 'image/png' && this.selectedFile.type != 'image/jpeg') {
  this.selectedFile = null;
  this.isReadyToSave = this.form.invalid;
  const toast = this.toastCtrl.create({
    message: 'Only Image formats are accepted!',
    showCloseButton: true,
    closeButtonText: 'Ok'
  });
  toast.present();
}
}

getProfileImageStyle() {
  return 'url(' + this.form.controls['image'].value + ')'
}

doSignup() {
  // Attempt to login in through our User service
  this.user.signup(this.account).subscribe((resp) => {
    this.navCtrl.push(MainPage);
  }, (err) => {
    this.navCtrl.push(MainPage);

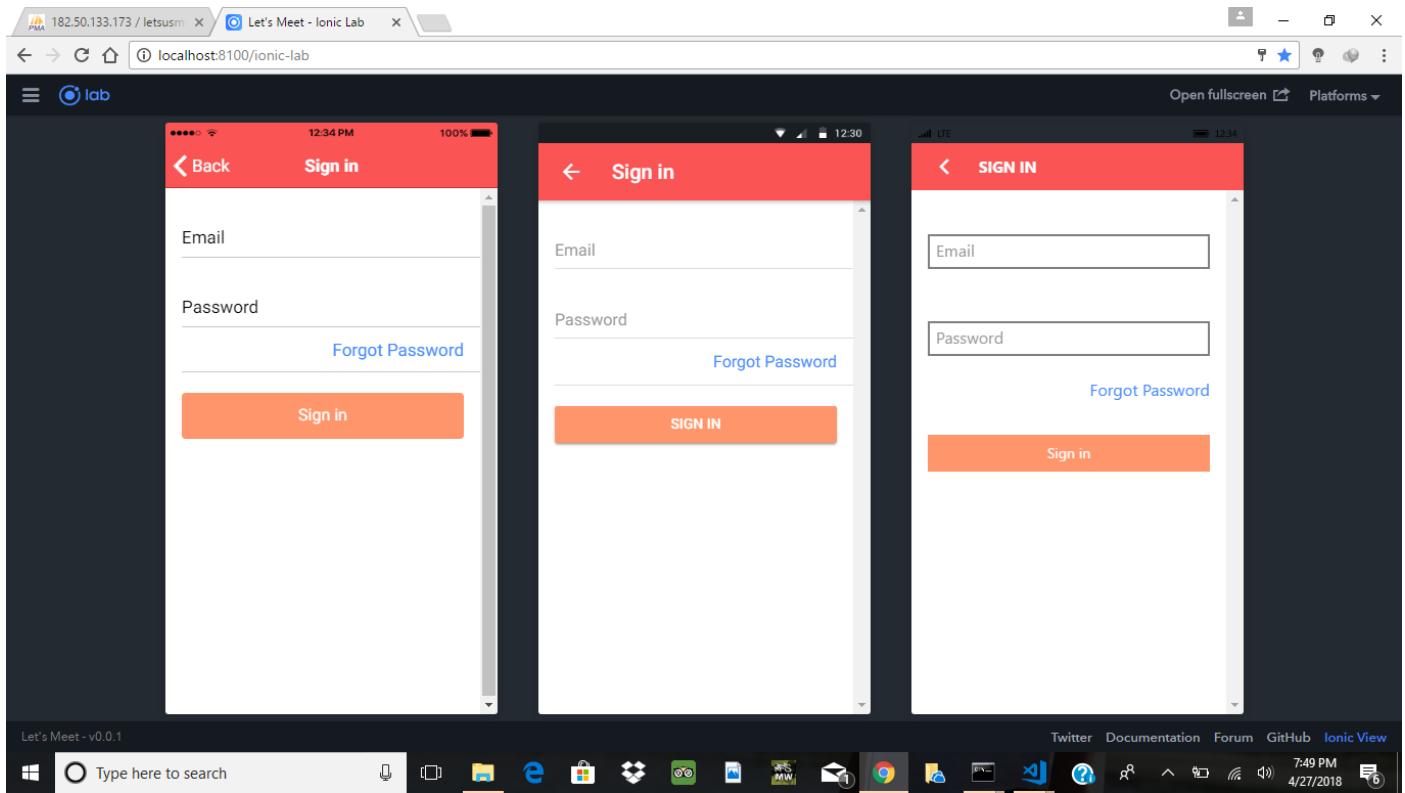
    // Unable to sign up
    let toast = this.toastCtrl.create({
      message: this.signupErrorString,
      duration: 3000,
      position: 'top'
    });
    toast.present();
  });
}

onClick() {
  if (!this.form.valid) { return; }
  const fd = new FormData();
  fd.append("user_id", this.eid);
  fd.append("user_name", this.uname);
  fd.append("user_pass", this.pass);
  fd.append("image", this.selectedFile, this.selectedFile.name);
  fd.append("gender", this.gender);
  fd.append("user_mob_no", this.mobile);
  fd.append("user_bdate", this.myDate);
  fd.append("token", "user");
  alert(this.eid);
  console.log(fd);
  console.log(this.token);
}

```

```
this.data.addUser(fd).subscribe(  
  (data: any) => {  
    alert("done");  
    console.log(data);  
  },  
  function (err) {  
    alert(err);  
  },  
  function () {  
  }  
);  
}  
onfile() {  
  this.fileChooser.open().then(uri => console.log(uri))  
  .catch(e => console.log(e));  
}  
onbirth() {  
  this.datePicker.show({  
    date: new Date(),  
    mode: 'date',  
    androidTheme: this.datePicker.ANDROID_THEMES.THEME_HOLO_DARK  
}).then(  
  date => console.log('Got date: ', date),  
  err => console.log('Error occurred while getting date: ', err)  
);  
}  
}
```

Login



Service of Login

```
var db = require('../dbconnection');
var u = {
  getUser: function (us, callback) {
    console.log(us.user_id);
    console.log(us.user_pass);
    return db.query("select * from user_tbl where user_id=? and user_pass=? and token=?", [us.user_id, us.user_pass, us.token], callback);
  }
};
module.exports = u;
```

TypeScript

```
onClick() {
  this.data.doLogin(this.eid, this.pass, "user").subscribe(
    (dt) => {
      if (dt == "") {
        let toast = this.toastCtrl.create({
          message: "Invalid UserName Or Password , Try Again",
          duration: 4000,
          position: 'top'
        });
        toast.present();
      }
    }
  );
}
```

```

    }

    else {
        this.storage.set('uid', this.eid);
        this.navCtrl.push(MainPage);

    }
},

(err) => alert("invalid"),
);

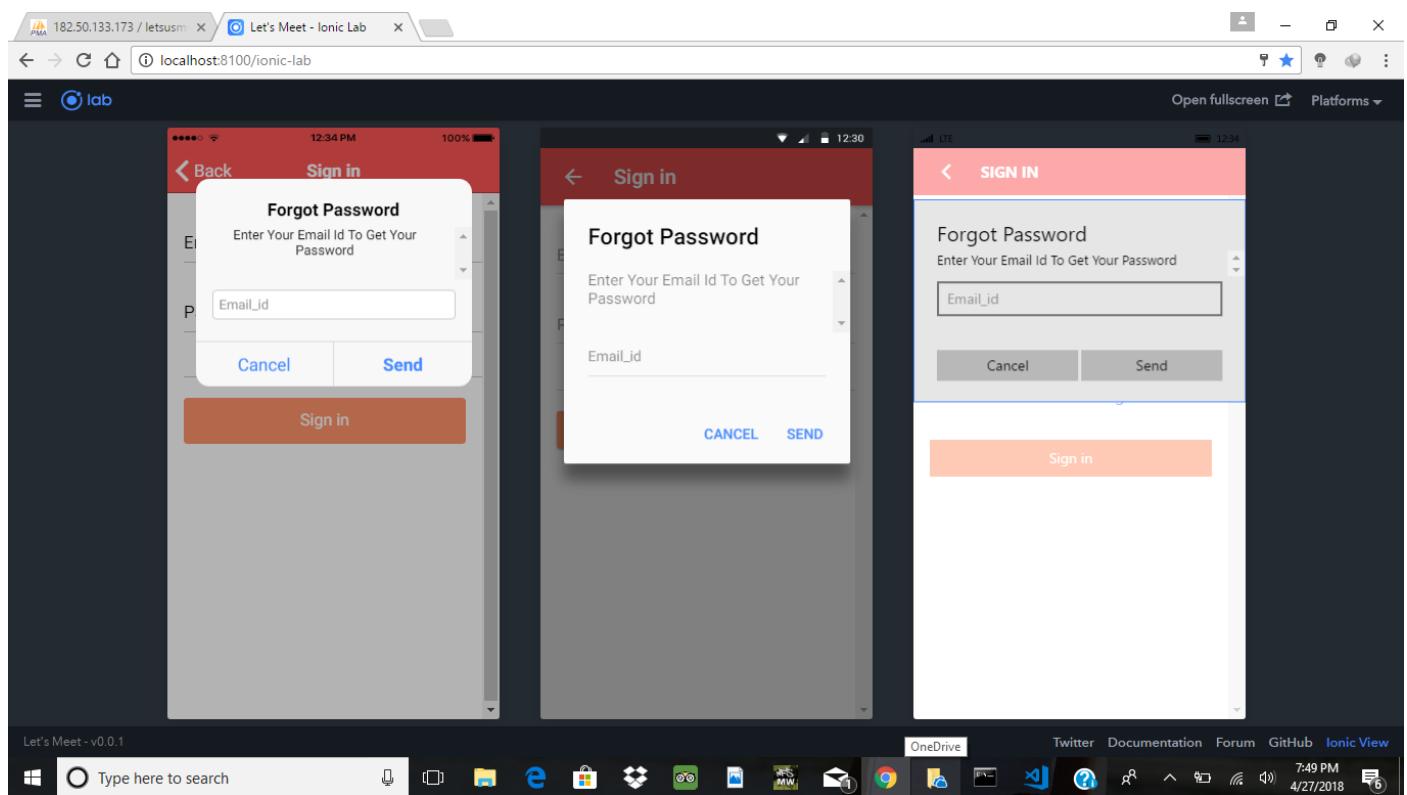
}

forid: string = '';
u: user_class[] = [];
validate() {
    let l1 = this.load.create({
        content: "Loading ..."
    });
    l1.present();
    this.data.getUser(this.forid)
        .subscribe(
            (dt: any) => {
                if (dt != "") {
                    this.u = dt;
                    this.eid = this.u[0].user_id;
                    var message = "Hello " + this.forid + ". You have requested to reset the password. your password is '" + this.u[0].user_pass + "'. Password is one of the confidential thing, Don't share it with anyone.";
                    this.data.sendMail(new email_class(message, this.forid, "Resetting the password of Lets_Meet."))
                        .subscribe(
                            (data1: any) => {
                                console.log("mail sent");
                                alert("The Password has been sent to " + this.forid);
                            },
                            function (e) {
                                alert(e);
                            },
                            function () {

```

```
    l1.dismiss();  
  }  
);  
}  
  
else {  
  alert("enter valid mail");  
}  
}  
}  
  
,  
function (e) {  
  alert(e);  
},  
function () {  
  l1.dismiss();  
}  
);  
}  
}
```

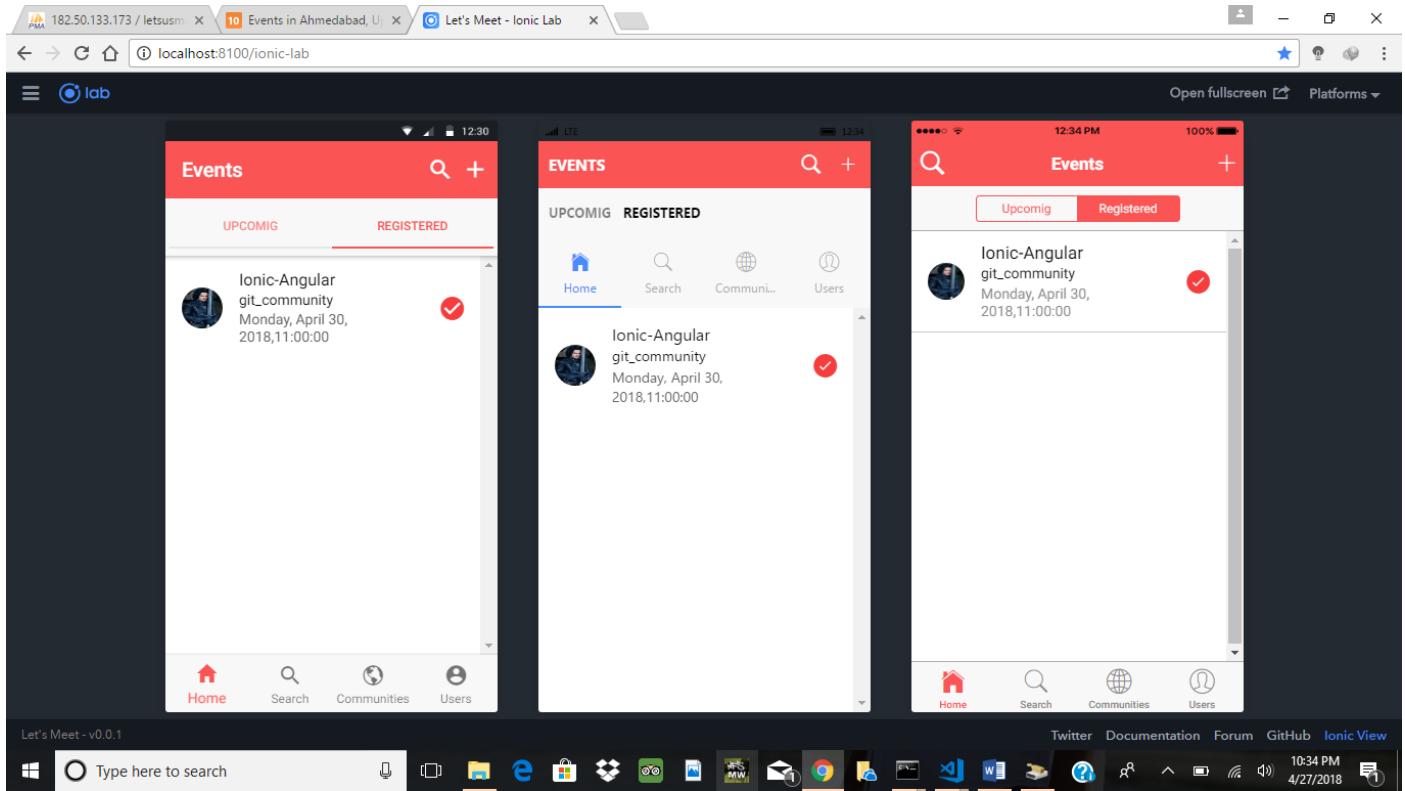
Forgot Password



TypeScript

```
showPrompt() {  
  let prompt = this.alert.create({  
    title: 'Forgot Password',  
    message: "Enter Your Email Id To Get Your Password",  
    inputs: [  
      {  
        name: 'name',  
        placeholder: 'Email_id'  
      },  
    ],  
    buttons: [  
      {  
        text: 'Cancel',  
        handler: data => {  
          console.log('Cancel clicked');  
        }  
      },  
      {  
        text: 'Send',  
        handler: data => {  
          this.forid = data.name;  
          this.validate();  
        }  
      }  
    ]  
  });  
  prompt.present();  
}
```

Events



Service Of Event

```

var db = require('../dbconnection');
var fs = require('fs');
var Event = {

    getAllEvent: function (callback) {
        return db.query("SELECT e.*,c.*,u.* from event_tbl e,communities_tbl c,user_tbl u WHERE e.fk_user_id=u.user_id and e.event_verify='true' and c.comm_id=e.fk_comm_id", callback);
    },
    getUpcomingEvents: function (callback) {
        return db.query("select * from event_tbl where event_date > CURDATE() and e.event_verify='true'", callback);
    },
    getEventById: function (id, callback) {

        return db.query("select e.*,u.* from event_tbl e,user_tbl u where e.event_id=? and e.fk_user_id=u.user_id", [id], callback);
    },
    addEvent: function (e, filename, callback) {
        return db.query("insert into event_tbl values(?,?,?,?,?,?,?,?,?,?)", [null, e.event_name, e.event_des, filename, e.event_s_time, e.event_e_time, e.event_date, e.event_loc, e.fk_user_id, e.fk_comm_id, "true"], callback);
    },
    deleteEvent: function (id, callback) {
        var evn = db.query("select * from event_tbl where event_id=?", [id]);
        evn.on('result', function (row) {
            console.log(row.event_pic);
            if (row.event_pic != '') {

```

```

        var path = 'public/images/events/' + row.event_pic;
        console.log(path);
        fs.unlink(path, function (err) {
            if (err) {
                console.log(err);
            }
            console.log('Deleted successfully');
        });
    });
});

return db.query("delete from event_tbl where event_id=?", [id], callback);
},
updateEvent: function (e, filename, callback) {
    var evn = db.query("select * from event_tbl where event_id=?", [e.event_id]);
    evn.on('result', function (row) {
        console.log(row.event_pic);
        if (row.event_pic != '') {
            var path = 'public/images/events/' + row.event_pic;
            console.log(path);
            fs.unlink(path, function (err) {
                if (err) {
                    console.log(err);
                }
                console.log('Deleted successfully');
            });
        }
    });
}

return db.query("update event_tbl set
event_name=?,event_des=?,event_pic=?,event_s_time=?,event_e_time=?,event_date=?,event_loc=?
,fk_comm_id=? where event_id=?", [e.event_name, e.event_des, filename, e.event_s_time,
e.event_e_time, e.event_date, e.event_loc, e.fk_comm_id, e.event_idid], callback);
},
updateEventOnly: function (e, callback) {
    return db.query("update event_tbl set
event_name=?,event_des=?,event_s_time=?,event_e_time=?,event_date=?,event_loc=?,fk_comm_id=?
where event_id=?", [e.event_name, e.event_des, e.event_s_time, e.event_e_time,
e.event_date, e.event_loc, e.fk_comm_id, e.event_id], callback);
},
eventRegisteredRsvp: function (user_id, callback) {
    return db.query("select e.*,r.*,c.* from event_tbl e,rsvp_tbl r,communities_tbl c
where r.fk_event_id=e.event_id and c.comm_id=e.fk_comm_id and r.rsvp_fk_user_id=? and
e.event_date > CURDATE() and e.event_verify='true'", [user_id], callback);
},
getAllUpcNotReg: function (user_id, callback) {
    return db.query("SELECT e.*,r.*,c.* from event_tbl e,rsvp_tbl r,communities_tbl c
where e.event_id=r.fk_event_id AND e.fk_comm_id=c.comm_id AND r.rsvp_fk_user_id!=? and
e.event_date > CURDATE() and e.event_verify='true'", [user_id], callback);
},
getPastEventUser: function (user_id, callback) {
    return db.query("select e.*,r.*,c.* from event_tbl e,rsvp_tbl r,communities_tbl c
where r.fk_event_id=e.event_id and c.comm_id=e.fk_comm_id and r.rsvp_fk_user_id=? and
e.event_date < CURDATE() and e.event_verify='true'", [user_id], callback);
},
}

```

```

getUpcEventUser: function (user_id, callback) {
    return db.query("select e.*,r.*,c.* from event_tbl e,rsvp_tbl r,communities_tbl c
where r.fk_event_id=e.event_id and c.comm_id=e.fk_comm_id and r.rsvp_fk_user_id=? and
e.event_date > CURDATE() and e.event_verify='true'", [user_id], callback);
},
getPastEventsByCommunity: function (comm_id, callback) {
    return db.query("SELECT e.*,u.*,c.* from event_tbl e,user_tbl u,communities_tbl c
WHERE e.fk_user_id=u.user_id and e.fk_comm_id=c.comm_id and c.comm_id=? and e.event_date <
CURDATE() and e.event_verify='true'", [comm_id], callback);
},
getUpcEventsByCommunity: function (comm_id, callback) {
    return db.query("SELECT e.*,u.*,c.* from event_tbl e,user_tbl u,communities_tbl c
WHERE e.fk_user_id=u.user_id and e.fk_comm_id=c.comm_id and c.comm_id=? and e.event_date >
CURDATE() and e.event_verify='true'", [comm_id], callback);
},
getUnApprovedEvent: function (callback) {
    return db.query("SELECT e.*,u.* from event_tbl e,user_tbl u WHERE
e.fk_user_id=u.user_id and e.event_verify='false' and event_date > CURDATE()", callback);
},
verify_event: function (event, callback) {
    return db.query("update event_tbl set event_verify='true' where event_id=? and
event_verify='false'", [event.event_id], callback);
},
deleteAllEvents: function (event, callback) {
    var delarr = [];
    for (i = 0; i < event.length; i++) {
        delarr[i] = event[i].event_id;
        var evn = db.query("select * from event_tbl where event_id=?", delarr[i]);
        evn.on('result', function (row) {
            console.log(row.event_pic);
            if (row.event_pic != '') {
                var path = 'public/images/events/' + row.event_pic;
                console.log(path);
                fs.unlink(path, function (err) {
                    if (err) {
                        console.log(err);
                    }
                    console.log('Deleted successfully');
                });
            }
        });
    });
    return db.query("delete from event_tbl where event_id in (?)", [delarr],
callback);
}
};

module.exports = Event;

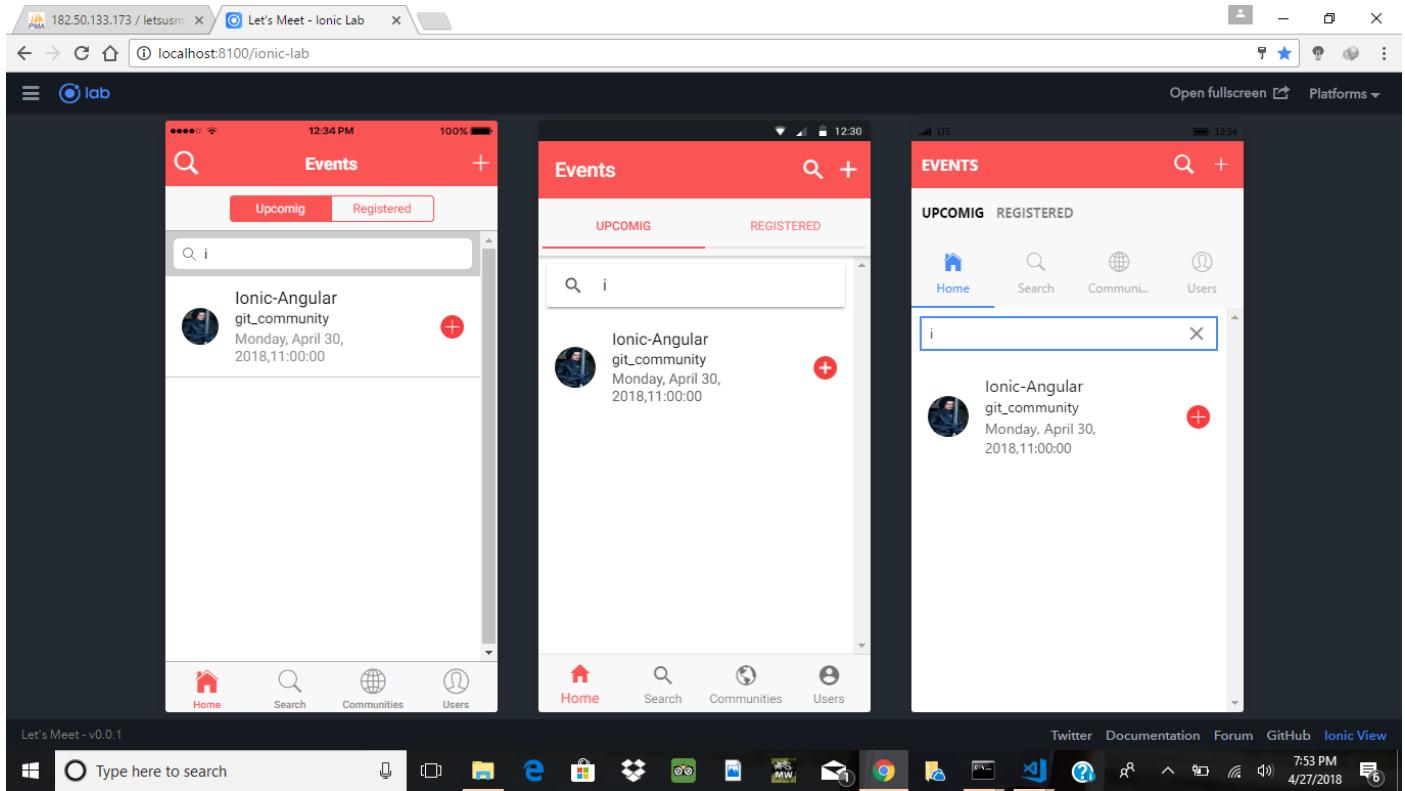
```

TypeScript

```
let l1 = this.load.create({
```

```
content: 'Loading Events'
});
l1.present();
this.storage.get('uid').then((val) => {
  this.u_id = val;
 //this._data1.getAllEventsAndCommunities().subscribe(
 this._data.getNotRegisteredEventsofUser(this.u_id).subscribe(
  (d: Event_Community_Class[]) => {
    this.arr = d;
    this.arr1 = d;
    console.log(this.arr);
    console.log(this.arr[0].event_pic);
  },
  function (e) {
    alert(e);
  },
  function () {
    l1.dismiss();
  }
);
});
let l2 = this.load.create({
  content: 'Registered Events'
});
l2.present();
this.storage.get('uid').then((val) => {
  this.u_id = val;
  this._data.getRegisteredEventsofUser(this.u_id).subscribe(
    (data: Event_Comm_Rsvp[]) => {
      this.reg = data;
      console.log(this.reg);
    },
    function (err) {
      alert(err);
    },
    function () {
      l2.dismiss();
    }
);
});
```

Search Event



TypeScript

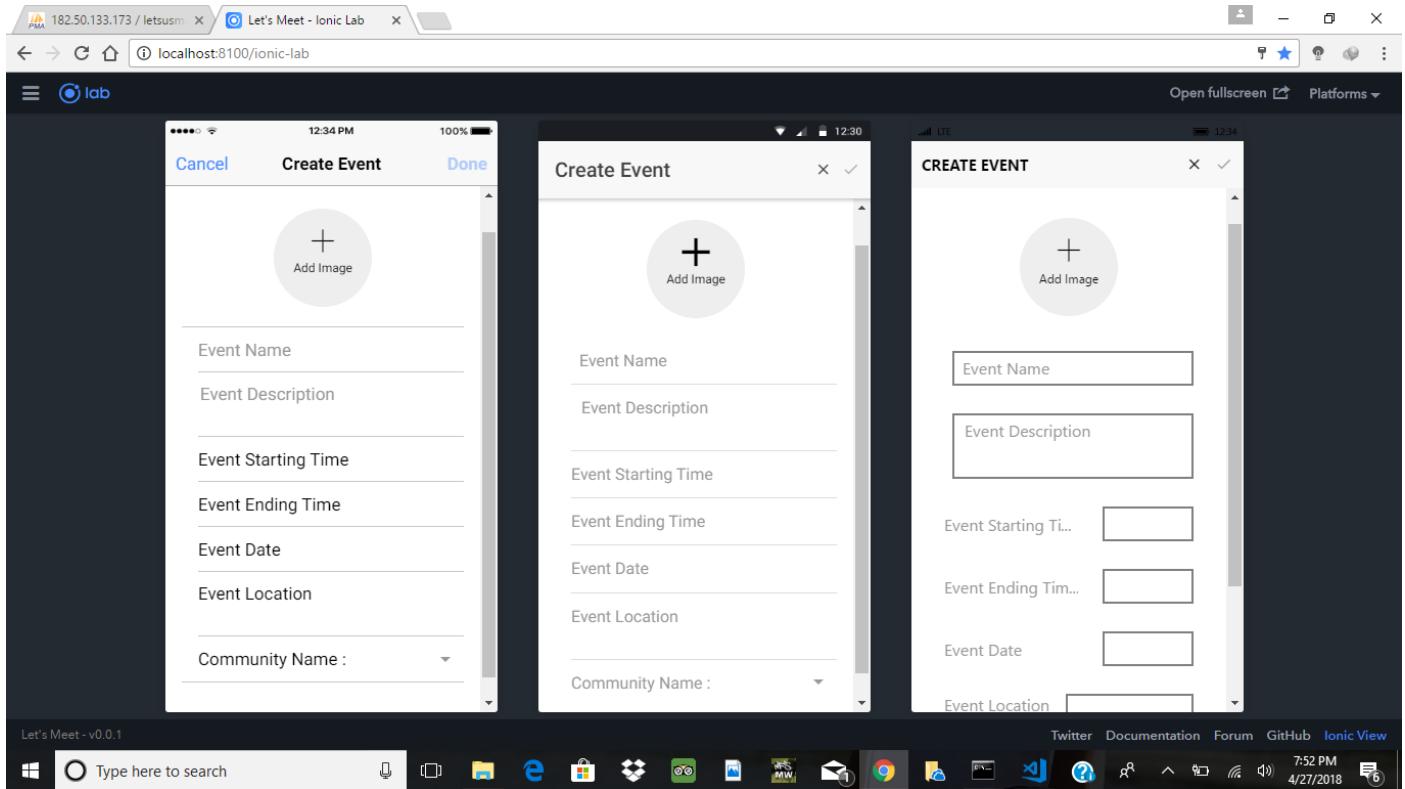
```

onSearchIcon() {
  if (this.flag == true) {
    this.flag = false;
  }
  else {
    this.flag = true;
  }
}

getItems(ev) {
  this.arr = this.arr1;
  var val = ev.target.value;
  if (val && val.trim() != '') {
    this.arr = this.arr.filter((x) =>
      x.event_name.toLocaleLowerCase().indexOf(val.toLocaleLowerCase()) > -1);
  }
}

```

Create Event



TypeScript

```

this.form = formBuilder.group({
  profilePic: ['', Validators.required],
  event_name: ['', Validators.required],
  event_des: ['', Validators.compose([Validators.minLength(15),
Validators.required])],
  event_s_time: ['', Validators.required],
  event_e_time: ['', Validators.required],
  event_date: ['', Validators.required],
  event_loc: ['', Validators.compose([Validators.minLength(10),
Validators.required])],
  community_id: ['', Validators.required]
});

// Watch the form for changes, and
this.form.valueChanges.subscribe((v) => {
  this.isReadyToSave = this.form.valid;
});

// this.today = moment().startOf('day').format('DD/MMMM/YYYY');
this.today = (new Date(Date.now() - (new Date()).getTimezoneOffset() *
60000)).toISOString();
}

ionViewDidLoad() {

  this._data1.getAllCommunities().subscribe(
    (data: Community_Class[]) => {

```

```

        this.arr = data;
    },
    function (e) {
        alert(e);
    },
    function () {
    }
)
}

getPicture() {
    if (Camera['installed']()) {
        this.camera.getPicture({
            destinationType: this.camera.DestinationType.DATA_URL,
            targetWidth: 96,
            targetHeight: 96
        }).then((data) => {
            this.form.patchValue({ 'profilePic': 'data:image/jpg;base64,' + data });
        }, (err) => {
            alert('Unable to take photo');
        })
    } else {
        this.fileInput.nativeElement.click();
    }
}

processWebImage(event) {

    this.selectedFile = <File>event.target.files[0];
    console.log(this.selectedFile);
    if (this.selectedFile.type != 'image/png' && this.selectedFile.type != 'image/jpeg') {
        this.selectedFile = null;
        this.isReadyToSave = this.form.invalid;
        const toast = this.tos.create({
            message: 'Only Image formats are accepted!',
            showCloseButton: true,
            closeButtonText: 'Ok'
        });
        toast.present();
    }
    else {
        let reader = new FileReader();
        reader.onload = (readerEvent) => {

            let imageData = (readerEvent.target as any).result;
            this.form.patchValue({ 'profilePic': imageData });
        };

        reader.readAsDataURL(event.target.files[0]);
    }
}

```

```

}

getProfileImageStyle() {
  return 'url(' + this.form.controls['profilePic'].value + ')'
}

/**
 * The user cancelled, so we dismiss without sending data back.
 */
cancel() {
  this.viewCtrl.dismiss();
}

/**
 * The user is done and wants to create the item, so return it
 * back to the presenter.
 */
done() {
  if (!this.form.valid) { return; }
  this.storage.get('uid').then((val) => {
    this.created_by = val;
    let l1 = this.load.create({
      content: 'Creating ...'
    });
    l1.present();
    let t1 = this.tos.create({
      duration: 3000,
      message: "Added ..."
    })

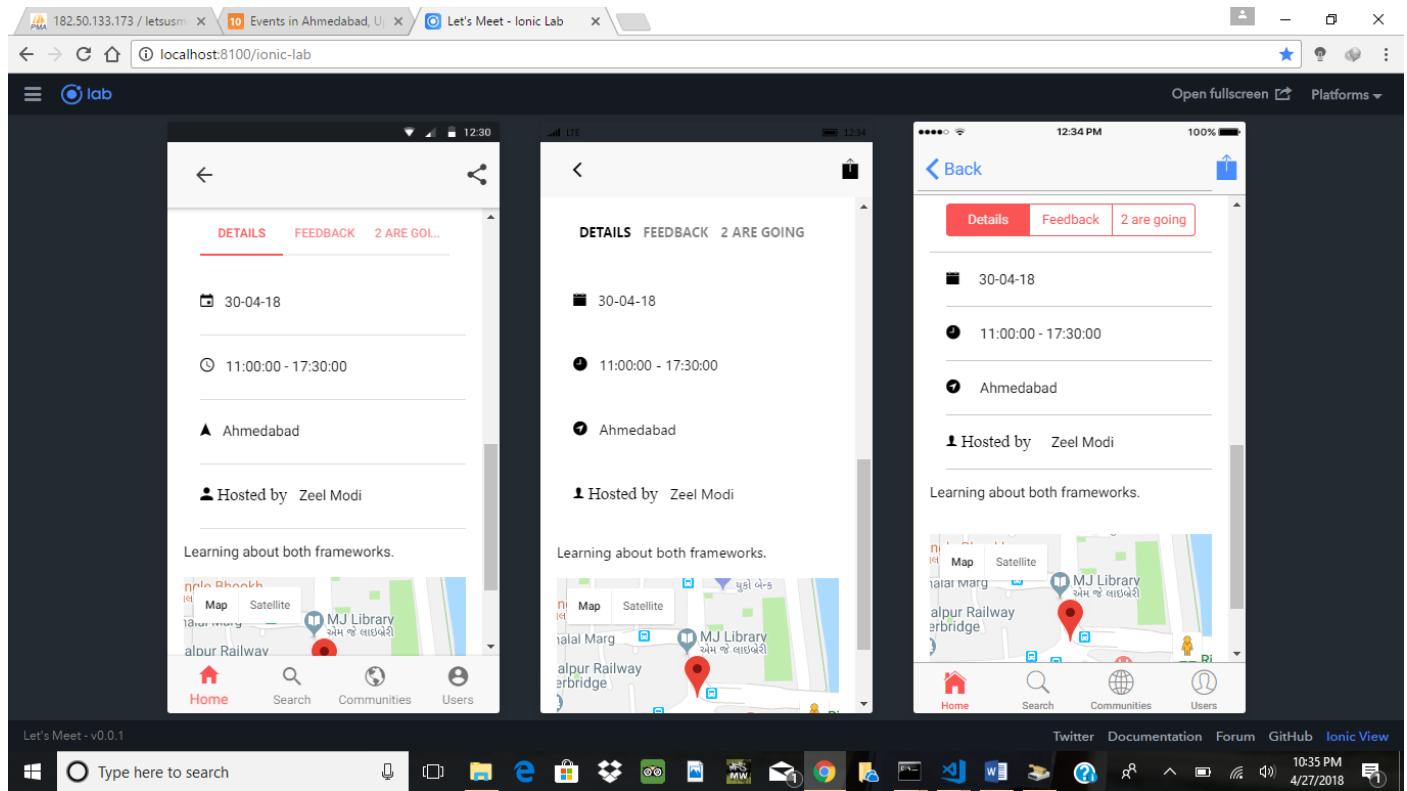
    const fd = new FormData();
    fd.append("event_id", this.event_id);
    fd.append("event_name", this.event_name);
    fd.append("event_des", this.event_des);
    fd.append("image", this.selectedFile, this.selectedFile.name);
    fd.append("event_s_time", this.event_s_time);
    fd.append("event_e_time", this.event_e_time);
    fd.append("event_date", this.event_date);
    fd.append("event_loc", this.event_loc);
    fd.append("fk_user_id", this.created_by);
    fd.append("fk_comm_id", this.community_id);
    fd.append("event_verify", "true");

    this._data.addEvent(fd).subscribe(
      (data: any) => {
        console.log(data);
        this.viewCtrl.dismiss();
        t1.present();
      },
      function (e) {
        alert(e);
      },
      function () {
    )
}

```

```
        11.dismiss());  
    }  
}  
});  
};  
}
```

View Event



TypeScript

```
ionViewDidLoad() {
  console.log('ionViewDidLoad ViewEventPage');
  this.e_id = this.navParams.get('e_id');
  this._dataEvent.getEventById(this.e_id).subscribe(
    (d: Events_User_Class[]) => {
      this.arr = d;
      this.event_name = this.arr[0].event_name;
      this.event_des = this.arr[0].event_des;
      this.event_pic = this.arr[0].event_pic;
      this.event_s_time = this.arr[0].event_s_time;
      this.event_e_time = this.arr[0].event_e_time;
      this.event_date = this.arr[0].event_date;
      this.event_loc = this.arr[0].event_loc;
      this.created_by = this.arr[0].user_name;
      this.u_id = this.arr[0].user_id;
      this.e_date = this.event_date;
      this.e_Str = new String(this.e_date).toString();
      //alert(this.e_Str.substr(0, this.e_Str.indexOf('T')));
      this.e_Str = this.e_Str.substr(0, this.e_Str.indexOf('T'));
      this.to_date = new Date(this.e_Str);
    }
  );
}
```

```

//this.geo_code(this.event_loc);
this.geo_code(this.event_loc);

},
function (e) {
  alert(e);
},
function () {

}

);

this._dataEventComm.getCommunityByEventId(this.e_id).subscribe(
  (data: Event_Community_Class[]) => {
    this.event_community = data;
    this.comm_id = this.event_community[0].comm_id;
    this.comm_name = this.event_community[0].comm_name;
    this.comm_pic = this.event_community[0].comm_pic;
  },
  function (e) {
    alert(e)
  },
  function () {

}

)
}

this.storage.get('uid').then((val) => {
  this.user_id = val;
  this._dataRSVP.checkRSVPofEvent(this.user_id, this.e_id).subscribe(
    (data) => {
      if (data == "") {
        this.join_button = true;
        this.going_button = false;
      }
      else {
        this.join_button = false;
        this.going_button = true;
      }
    },
    function (e) {
      alert(e);
    },
    function () {
    }
  );
});

this._dataRSVP.countRSVP(this.e_id).subscribe(
  (data: Event_RSVP_User_class[]) => {
    //this.cnt_rsvp = data[0].count;
    this.event_rsvp_user = data;
    this.cnt_rsvp = this.event_rsvp_user.length;
  }
);
}

```

```

},
function (err) {
  alert(err);
},
function () {

}

);

//alert(this.e_id);
this._dataFeedback.getFeedbacksByEvent(this.e_id).subscribe(
  (data: Feedback_Event_User_Class[]) => {
    this.feedback_event_user = data;
    //console.log(this.feedback_event_user);
  },
  function (err) {
    alert(err);
  },
  function () {

}

);

this.storage.get('uid').then(
  (val) => {
    this.user_id = val;
    this._dataUser.getUser(this.user_id).subscribe(
      (data: user_class[]) => {
        this.arrUser = data;
        this.user_pic = this.arrUser[0].user_pic;
        this.user_name = this.arrUser[0].user_name;
      },
      function (err) {
        alert(err);
      },
      function () {

}

);

}

);

this.calendar.createCalendar('MyCalendar').then(
  (msg) => { console.log(msg); },
  (err) => { console.log(err); }
);

}

geo_code(address: string) {
  console.log("inside geo_code::");
  this._geolocation.getCurrentPosition(address).subscribe(
    (data: any) => {

```

```

        this.loadMap(data);
        console.log("add:" + data);
    }
);

}

loadMap(data: any) {
    let lat = data["results"][0].geometry.location.lat;
    let lng = data["results"][0].geometry.location.lng;
    let latLng = new google.maps.LatLng(lat, lng);
    console.log("lat:" + lat);
    let mapOptions = {
        center: latLng,
        zoom: 16,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    this.map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);
    var marker = new google.maps.Marker({
        position: latLng,
        title: "Manekchowk,Ahmedabad,Gujarat,India",
        map: this.map,
        draggable: true
    });
}

changeFlag() {
    if (this.flag1 == true) {
        this.flag1 = false;
    }
    else {
        this.flag1 = true;
    }
}
onRSVP() {
    console.log(this.user_id);
    console.log(this.e_id);
    this.storage.get('uid').then((val) => {
        this.user_id = val;
        let l1 = this.load.create({
            content: 'Joining ...'
        });
        l1.present();
        let t1 = this.tos.create({
            duration: 3000,
            message: "Joined ..."
        })
        this._dataRSVP.addRSVP(new RSVP_Class(null, this.user_id, this.e_id)).subscribe(
            (data: RSVP_Class) => {
                t1.present();
                this.join_button = false;
                this.going_button = true;
            }
        )
    })
}

```

```

        this.calendar.createEvent(this.event_name, this.event_loc, this.event_des,
this.to_date, this.to_date).then(
    (msg) => { console.log(msg); },
    (err) => { console.log(err); }
);
},
function (e) {
    alert(e);
},
function () {
    l1.dismiss();
}
);
});
}

onClickRSVP() {
this.storage.get('uid').then((val) => {
    this.user_id = val;
    let l1 = this.load.create({
        content: 'Joining ...'
});
    l1.present();
    this._dataRSVP.checkRSVPofEvent(this.user_id, this.e_id).subscribe(
        (data: any) => {
            if (data == "") {
                this.onRSVP();
            }
            else {
                let t1 = this.tos.create({
                    duration: 3000,
                    message: "You're Going ..."
});
                t1.present();
                this.arrRsvp = data;
                // this.rsvp_id = this.arrRsvp.RSVP_id;
                //alert(this.rsvp_id);
            }
        },
        function (e) {
            alert(e);
        },
        function () {
            l1.dismiss();
        }
);
});
}

onRemoveRSVP() {

this.storage.get('uid').then((val) => {
    this.user_id = val;
}

```

```

let l1 = this.load.create({
  content: 'Removing ...'
});
l1.present();
let t1 = this.tos.create({
  duration: 3000,
  message: "Removed ..."
});
this._dataRSVP.checkRSVPofEvent(this.user_id, this.e_id).subscribe(
  (data: any) => {
    if (data == "") {
      alert("nathi");
    }
    else {
      this.arrRsvp = data;
      //console.log(data);
      //console.log(this.arrRsvp[0].RSVP_id);
      this._dataRSVP.deleteRSVP(this.arrRsvp[0].RSVP_id).subscribe(
        (data: RSVP_Class) => {
          this.calendar.deleteEvent(this.event_name, this.event_loc, this.event_des,
this.to_date, this.to_date).then(
            (msg) => { console.log(msg); },
            (err) => { console.log(err); }
          );
          //alert("thyu");
          this.join_button = true;
          this.going_button = false;
        },
        function (err) {
          alert(err);
        },
        function () {
          }
        );
      }
    },
    function (e) {
      alert(e);
    },
    function () {
      l1.dismiss();
      t1.present();
    }
  );
});
}

doConfirm() {
  let confirm = this.alerCtrl.create({
    title: 'Not going?',
    message: "Are you sure you don't want to attend this event?",
    buttons: [

```

```

        {
          text: 'Disagree',
          handler: () => {
            console.log('Disagree clicked');
          }
        },
        {
          text: 'Agree',
          handler: () => {
            this.onRemoveRSVP();
          }
        }
      ]
    });
  confirm.present()
}

onView() {
  this.navCtrl.push(ViewCommunityPage, { c_id: this.comm_id });
}

newFeedback() {
  if (this.feed_des.length > 0) {
    let l1 = this.load.create({
      content: 'Adding Feedback ...'
    });
    l1.present();
    this.storage.get('uid').then((val) => {
      this.user_id = val;
      this._dataFeedback.addFeedback(new Feedback_Class(null, this.feed_des, this.e_id,
this.user_id, null)).subscribe(
        (d: Feedback_Class) => {
          this.flag1 = true;
          this.feed_des = "";
          this ionViewDidLoad();
        },
        function (err) {
          alert(err);
        },
        function () {
          l1.dismiss();
        }
      );
    });
  }
  else {
    this.flag1 = true;
  }
}

deleteFeedback(id) {
  let t1 = this.tos.create({

```

```
duration: 3000,
message: "Deleted ..."

});

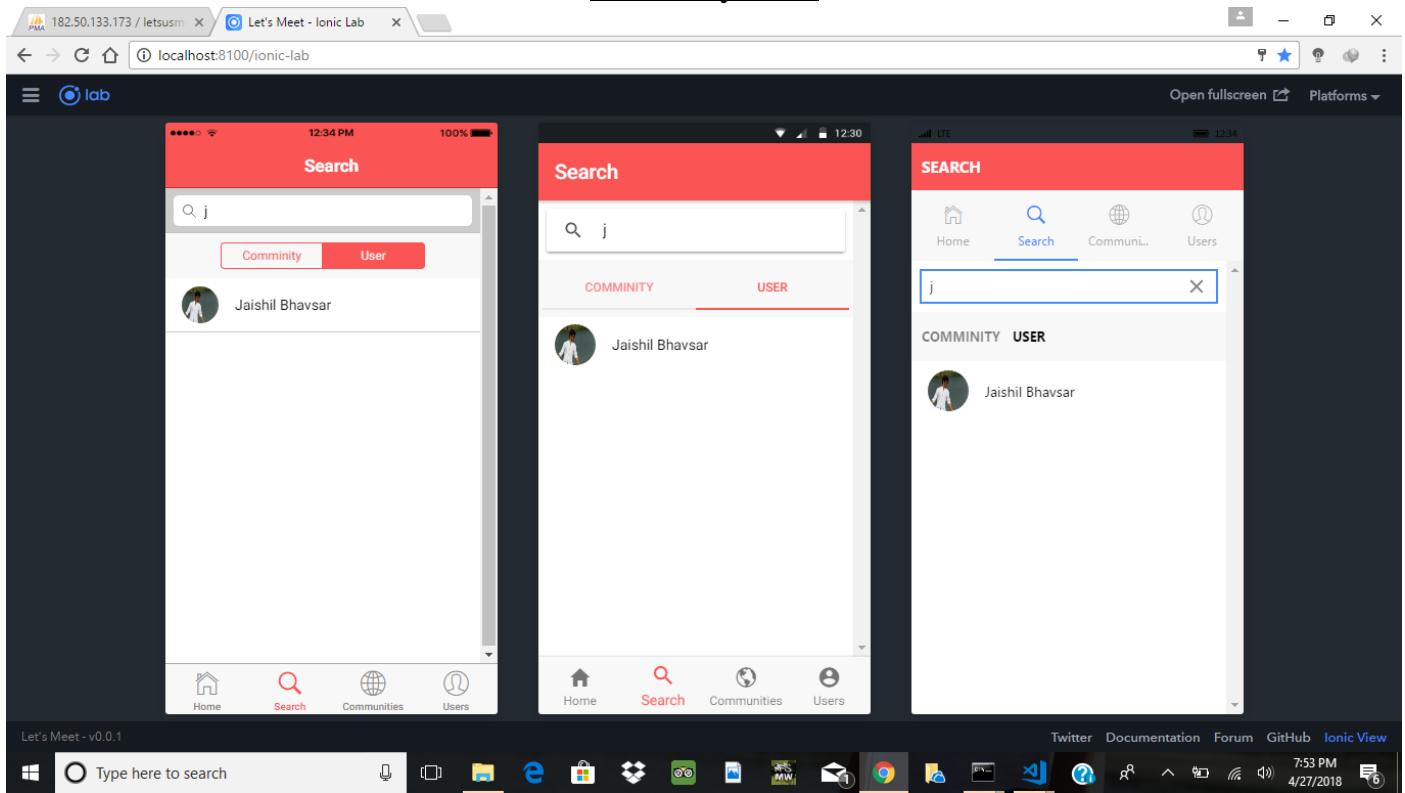
this._dataFeedback.deleteFeedback(id).subscribe(
  (d: any) => {
    this.ionViewDidLoad();
    t1.present();
  },
  function (err) {
    alert(err);
  },
  function () {

  }
);

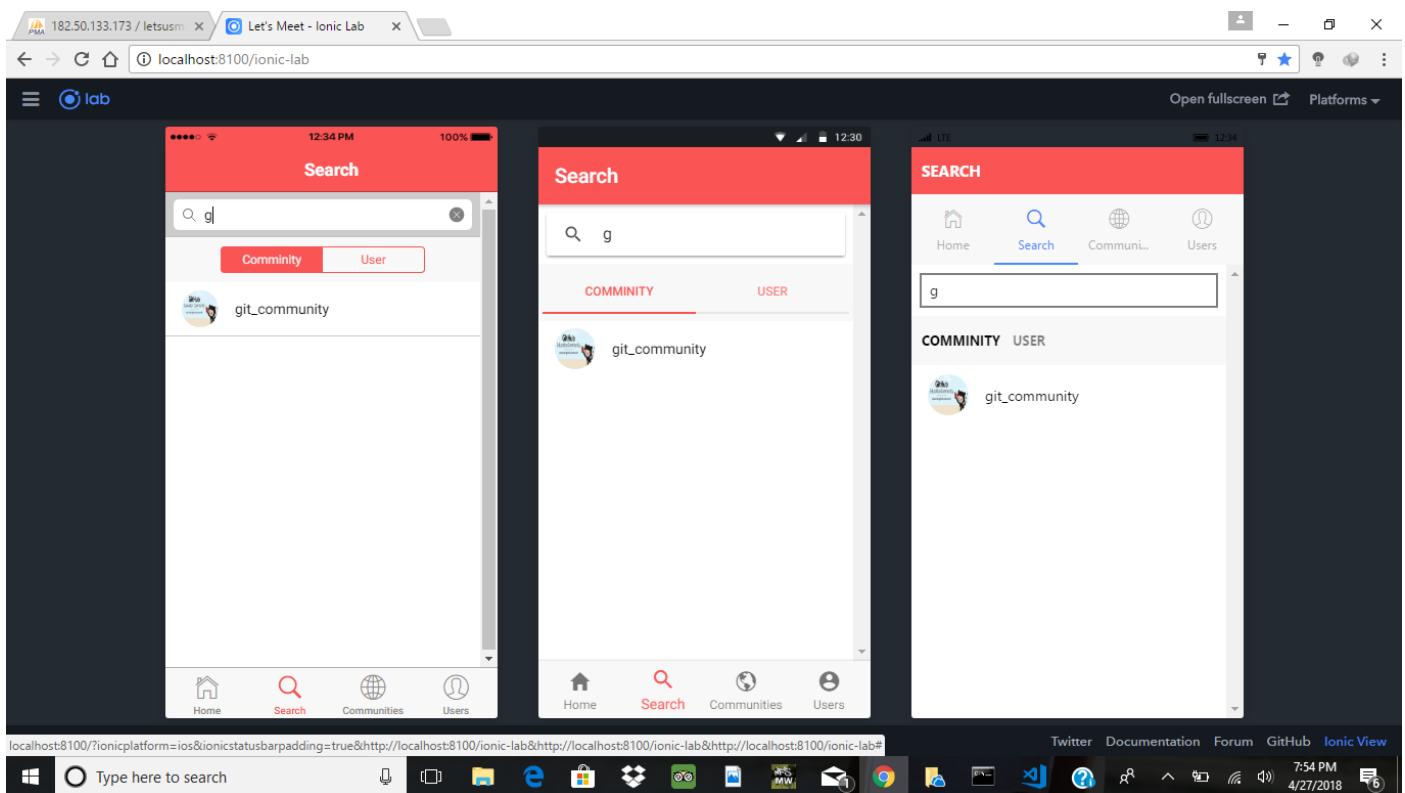
onSharing() {
  this.e_pic = "https://letsmeetbackend.herokuapp.com/public/images/events/" +
this.event_pic;
  this.socialSharing.share(this.event_des, this.event_name).
  catch((err) => alert(err));
}

showuser(id) {
  this.storage.set('viewid', id);
  this.navCtrl.push(ViewuserPage);
}
}
```

Search by User



Search by Community



TypeScript

```

ionViewDidLoad() {

  this._data.getAllUser().subscribe(
    (data: any) => {
      this.arr = data;
    },
    function (e) {
      alert(e);
    }
  );

  this._data1.getAllCommunities().subscribe(
    (data: any) => {
      this.comm_arr1 = data;
      //this.arr1 = data;

    },
    function (err) {
      alert(err);
    },
    function () {
      //l1.dismiss();
    }
  );
}

let l2 = this.load.create({
  content: "Loading..."
});
l2.present();

this._data1.gettopcommunity().subscribe(
  (data: any) => {
    this.topComm = data;
  },
  function (e) {
    alert(e);

  },
  function () {
    l2.dismiss();
  }
);
}

```

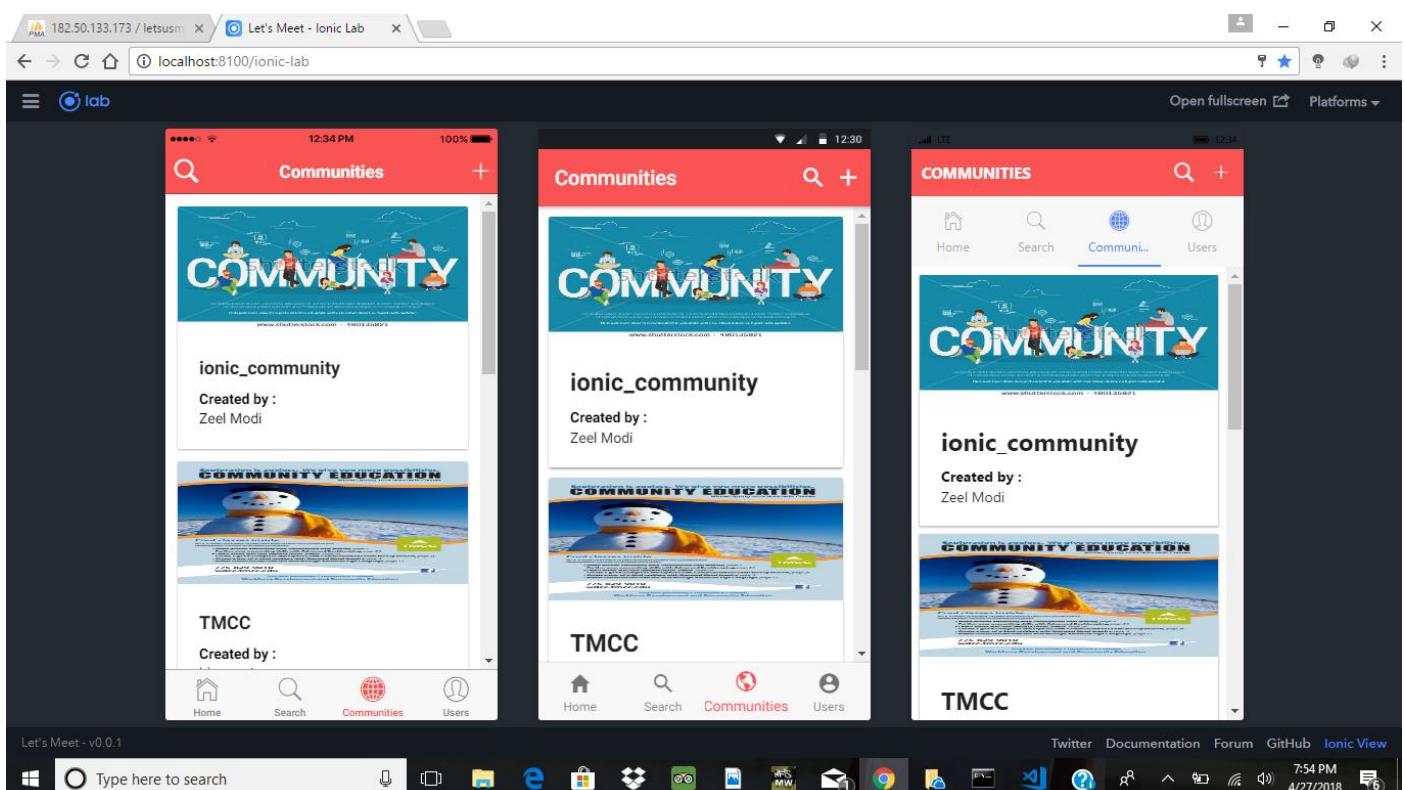
```

onSearch() {
  if (this.txtsearch != '') {
    this.arr1 = this.arr.filter((x) =>
      x.user_name.toLocaleLowerCase().indexOf(this.txtsearch.toLocaleLowerCase()) > -1);
    // this.arr1 = this.arr.filter((x) => x.startsWith(this.txtsearch))
    this.comm_arr = this.comm_arr1.filter((x) =>
      x.comm_name.startsWith(this.txtsearch));
  }
  else {
    this.arr1 = null;
    this.comm_arr = null;
  }
}
onClick(comm_id) {
  this.navCtrl.push(ViewCommunityPage, { c_id: comm_id });
}

onUser(user_id) {
  this.storage.set('viewid', user_id);
  this.navCtrl.push(ViewuserPage);
}
}

```

Communities



Service

```

var db = require('../dbconnection');

var comment = {

    getAllComments: function (callback) {
        return db.query("select * from comment_tbl", callback);
    },
    getCommentById: function (id, callback) {
        return db.query("select * from comment_tbl where com_id=?", [id], callback);
    },
    addComment: function (cm, callback) {

        console.log(cm);
        return db.query("INSERT INTO comment_tbl(com_id, com_date, com_des, fk_post_id,
com_fk_user_id) VALUES (?,?,?,?,?)", [null, cm.com_des, cm.fk_post_id,
cm.com_fk_user_id], callback)
    },
    updateComment: function (id, cm, callback) {
        return db.query("update comment_tbl set com_date=?,com_des=? where com_id=?",
[cm.com_date, cm.com_des, id], callback);
    },
    deleteComment: function (id, callback) {
        return db.query("delete from comment_tbl where com_id=?", [id], callback);
    },
    getCommentsByPost(id, callback) {
        return db.query("SELECT p.*,c.*,u.* FROM post_tbl p,comment_tbl c,user_tbl u WHERE
p.post_id=c.fk_post_id and c.com_fk_user_id=u.user_id and p.post_id=?", [id], callback);
    }
};

module.exports = comment;

```

TypeScript

```

this.form = this.formBuilder.group({});

let l1 = this.load.create({
    content: "Loading..."
});
l1.present();

this._data.getAllCommunities().subscribe(
    (data: any) => {
        this.arr = data;
    }
);

```

```

    this.arr1 = data;
}

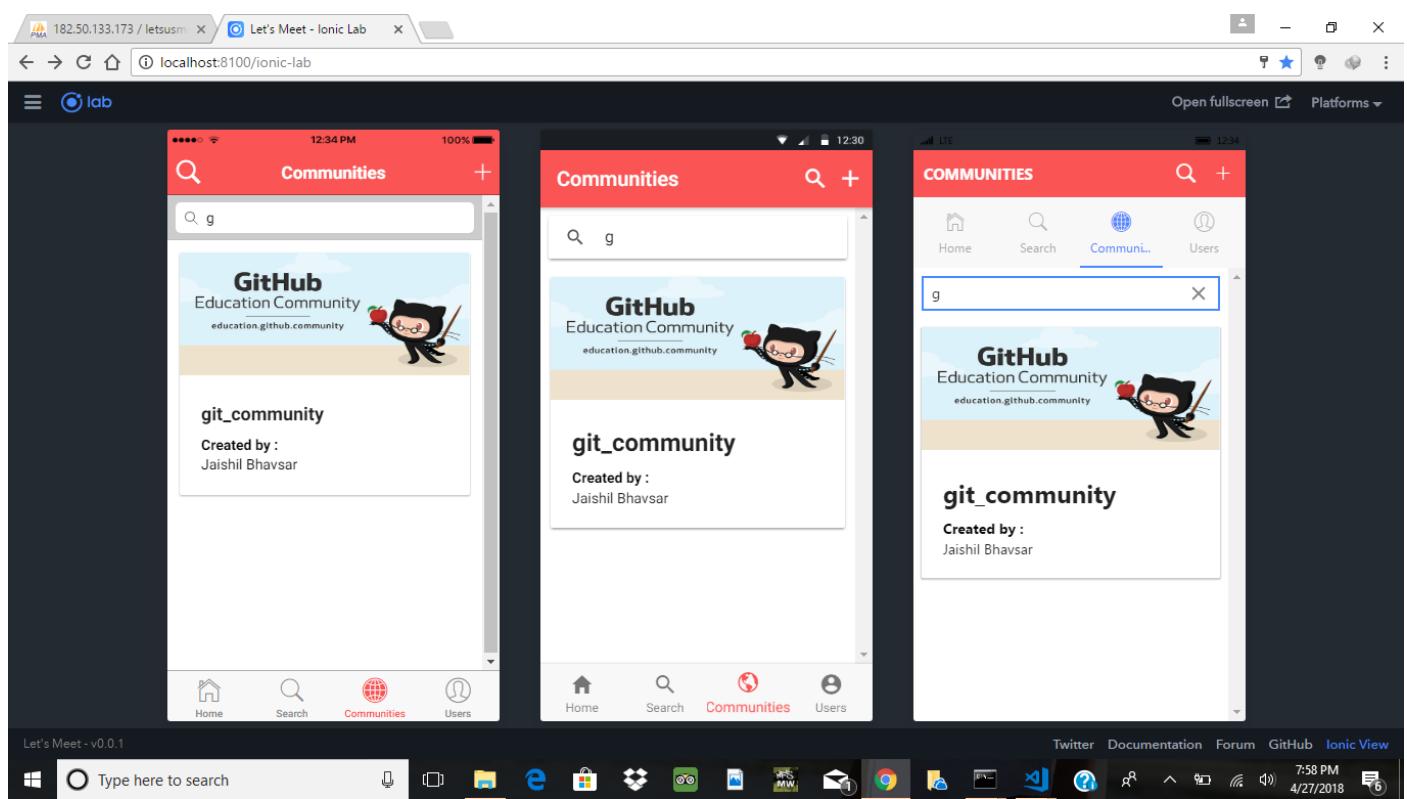
},
function (err) {
  alert(err);
},
function () {

  l1.dismiss();
}

);

```

Search Community



TypeScript

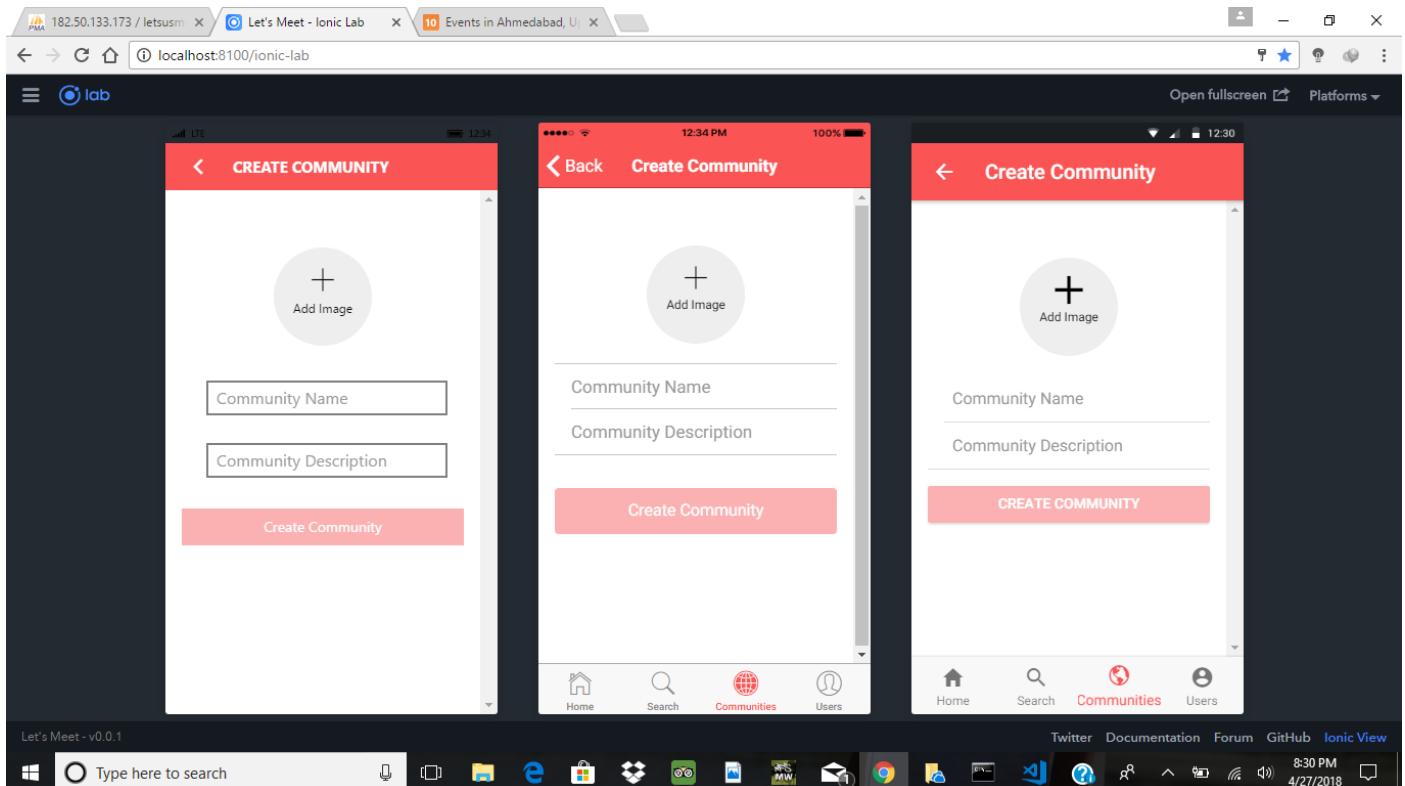
```
onSearch() {
```

```

if (this.txtsearch != '') {
  this.arr = this.arr.filter((x) => x.comm_name.startsWith(this.txtsearch))
}
else {
  this.arr = this.arr1;
}

```

Create Community



TypeScript

```

onCreate() {
  if (!this.form.valid) { return; }
  this.st.get('uid').then((val) => {
    this.created_by = val;

    let t1 = this.tos.create({
      message: "Created",
      duration: 3000
    });

    let l1 = this.load.create({
      content: "Loading..."
    });
    l1.present();

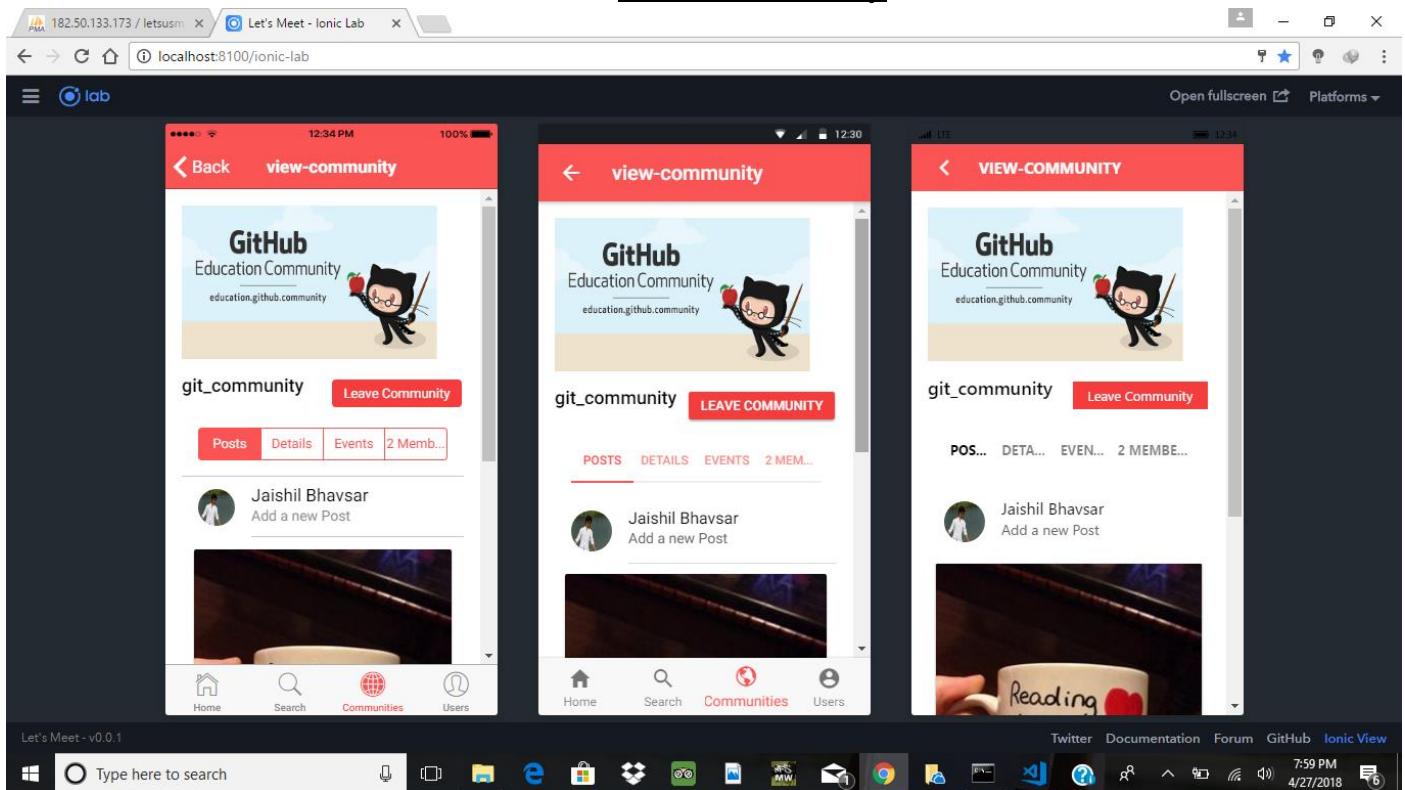
    const fd = new FormData();
    alert(this.created_by);
    fd.append("comm_id", this.comm_id);
    fd.append("comm_name", this.comm_name);
  })
}

```

```
        fd.append("comm_des", this.comm_des);
        fd.append("image", this.selectedFile, this.selectedFile.name);
        fd.append("comm_date", this.comm_date);
        fd.append("comm_rating", this.comm_rating);
        fd.append("created_by", this.created_by);
        fd.append("comm_fk_cat_id", this.comm_fk_cat_id);

    this._data.addCommunity(fd).subscribe(
        (data: any) => {
            console.log(data);
            t1.present();
            this.navCtrl.pop();
        },
        function (err) {
            alert(err);
        },
        function () {
            l1.dismiss();
        }
    );
});
```

View Community



TypeScript

```
ionViewDidLoad() {  
  console.log('ionViewDidLoad ViewCommunityPage');
```

```

this.comm_id = this.navParams.get('c_id');
this.storage.get('uid').then((val) => {
  this.user_id = val;

  let l1 = this.load.create({
    content: "Loading..."
  });
  l1.present();
  this._data.getCommunityById(this.comm_id).subscribe(
    (data: any) => {
      this.arr = data;
      this.comm_name = this.arr[0].comm_name;
      this.comm_des = this.arr[0].comm_des;
      this.comm_pic = this.arr[0].comm_pic;
      this.comm_date = this.arr[0].comm_date;
      if (this.user_id == this.arr[0].created_by) {
        this.comm_rating = this.arr[0].comm_rating;
      }
      this.created_by = this.arr[0].user_name;
    },
    function (err) {
      alert(err);
    },
    function () {
      l1.dismiss();
    }
  );
}

this._rate.getRateCount(1, this.comm_id).subscribe(
  (data: any) => {
    this.s1 = data[0]['COUNT(rate_value)'];
    this.sum += this.s1;
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);

this._rate.getAvgRating(this.comm_id).subscribe(
  (data: any) => {
    this.avg = data[0]['AVG(rate_value)'];
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);

```

```
function () {

};

this._rate.getCountRating(this.comm_id).subscribe(
  (data: any) => {
    this.rate = data[0]['COUNT(rate_value)'];
  },
  function (err) {
    alert(err);
  },
  function () {

});
}

this._rate.getRateCount(2, this.comm_id).subscribe(
  (data: any) => {

    this.s2 = data[0]['COUNT(rate_value)'];
    this.sum += this.s2;
  },
  function (err) {
    alert(err);
  },
  function () {

});
}

this._rate.getRateCount(3, this.comm_id).subscribe(
  (data: any) => {

    this.s3 = data[0]['COUNT(rate_value)'];
    this.sum += this.s3;
  },
  function (err) {
    alert(err);
  },
  function () {

});
}

this._rate.getRateCount(4, this.comm_id).subscribe(
```

```

(data: any) => {

    this.s4 = data[0]['COUNT(rate_value)'];
    this.sum += this.s4;
},
function (err) {
    alert(err);
},
function () {

}

);

this._rate.getRateCount(5, this.comm_id).subscribe(
(data: any) => {

    this.s5 = data[0]['COUNT(rate_value)'];
    this.sum += this.s5;
    this.w = (this.s1 / this.sum);
    this.w2 = (this.s2 / this.sum);
    this.w3 = (this.s3 / this.sum);
    this.w4 = (this.s4 / this.sum);
    this.w5 = (this.s5 / this.sum);
},
function (err) {
    alert(err);
},
function () {

}

);

this._commEvent.getCommunityByPastEvent(this.comm_id).subscribe(
(data:any)=>{
    this.comm_past_event=data;
},
function(e){
    alert(e);
},
function(){

}

);

this._commEvent.getCommunityByUpcomingEvent(this.comm_id).subscribe(
(data:any)=>{
    this.comm_upcoming_event=data;
}
)

```

```

    },
    function(e){
      alert(e);
    }
  );

let l2 = this.load.create({
  content: "Loading..."
});
l2.present();
this._data.getPostByCommunityId(this.comm_id).subscribe(
  (data: Community_Post_User_Class[]) => {
    this.comm_post_user = data;
  },
  function (e) {
    alert(e);
  },
  function () {
    l2.dismiss();
  }
);

this._comm_mem_data.getAllMembersByCommunityId(this.comm_id).subscribe(
  (data: Comm_member_class[]) => {
    this.comm_mem = data;
    this.comm_member_count = this.comm_mem.length;
  }
);

this._comm_mem_data.getAllMembers(this.comm_id).subscribe(
  (data: any) => {
    this.comm_comm_member = data;
    //this.mem_length=this.comm_comm_member.length;

  },
  function (e) {
    alert(e);
  },
  function () {
  }
);

this._data.checkCommMember(this.user_id, this.comm_id).subscribe(
  (data: any) => {
    if (data == "") {
      this.join_button = true;
      this.leave_button = false;
    }
    else {
      this.join_button = false;
      this.leave_button = true;
    }
  }
);

```

```

        this.arrMember = data;
        console.log(this.arrMember);
    }
}
);
});
});

this.commu_member.memberCount(this.comm_id).subscribe(
  (data) => {
    this.cnt_member = data[0].count;
  },
  function (err) {
    alert(err);
  },
  function () {

}

);

this.storage.get('uid').then(
  (val) => {
    this.user_id = val;
    this._dataUser.getUser(this.user_id).subscribe(
      (data: user_class[]) => {
        this.arrUser = data;
        this.user_pic = this.arrUser[0].user_pic;
        this.user_name = this.arrUser[0].user_name;
      },
      function (err) {
        alert(err);
      },
      function () {

      }
    );
  }
);
}

onJoin() {

this.storage.get('uid').then((val) => {
  this.user_id = val;
  //alert(this.user_id);
  let l1 = this.load.create({
    content: 'Joining ...'
  });
  l1.present();
  let t1 = this.toast.create({

```

```

        duration: 3000,
        message: "Joined ..."
    })
    this.commu_member.addCommunityMember(new Comm_member_class(null, this.user_id,
this.comm_id)).subscribe(
    (data: any) => {
        t1.present();
        this.join_button = false;
        this.leave_button = true;
        this.viewDidLoad();

        this._data.checkCommMember(this.user_id, this.comm_id).subscribe(
            (data: any) => {
                if (data == "") {
                    this.join_button = true;
                    this.leave_button = false;
                }
                else {
                    this.join_button = false;
                    this.leave_button = true;
                    this.arrMember = data;
                    console.log(this.arrMember);
                }
            }
        );
    },
    function (e) {
        alert(e);
    },
    function () {
        l1.dismiss();
    }
);
});

onRemove() {

let l1 = this.load.create({
    content: "Loading..."
});
l1.present();
console.log(this.arrMember[0].join_id);
this.commu_member.deleteMember(this.arrMember[0].join_id).subscribe(
    (data: any) => {
        //alert("deleted");
        this.join_button = true;
        this.leave_button = false;
        this.viewDidLoad();
    },
    function (err) {
}
}
)
}

```

```

    alert(err);
},
function () {
  l1.dismiss();
}

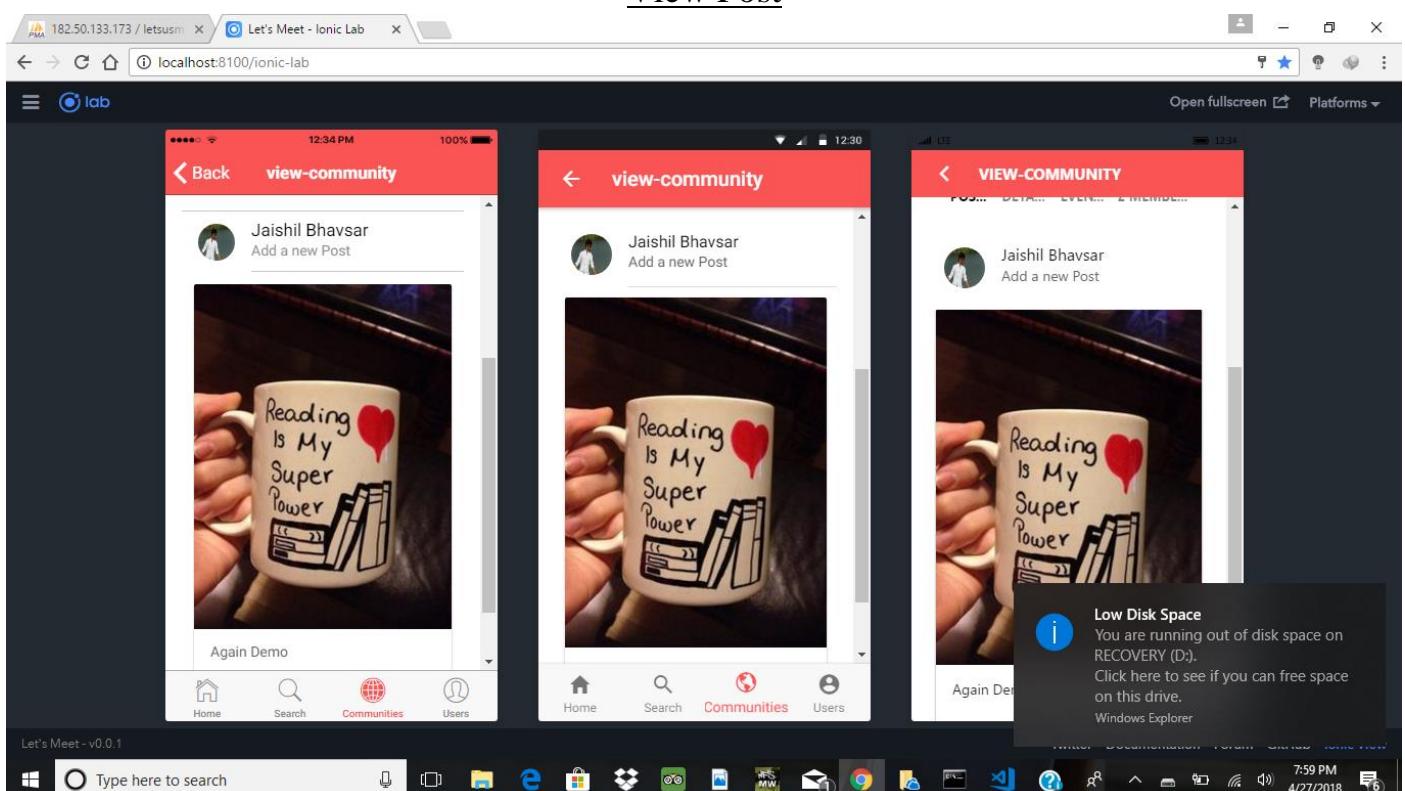
);

}

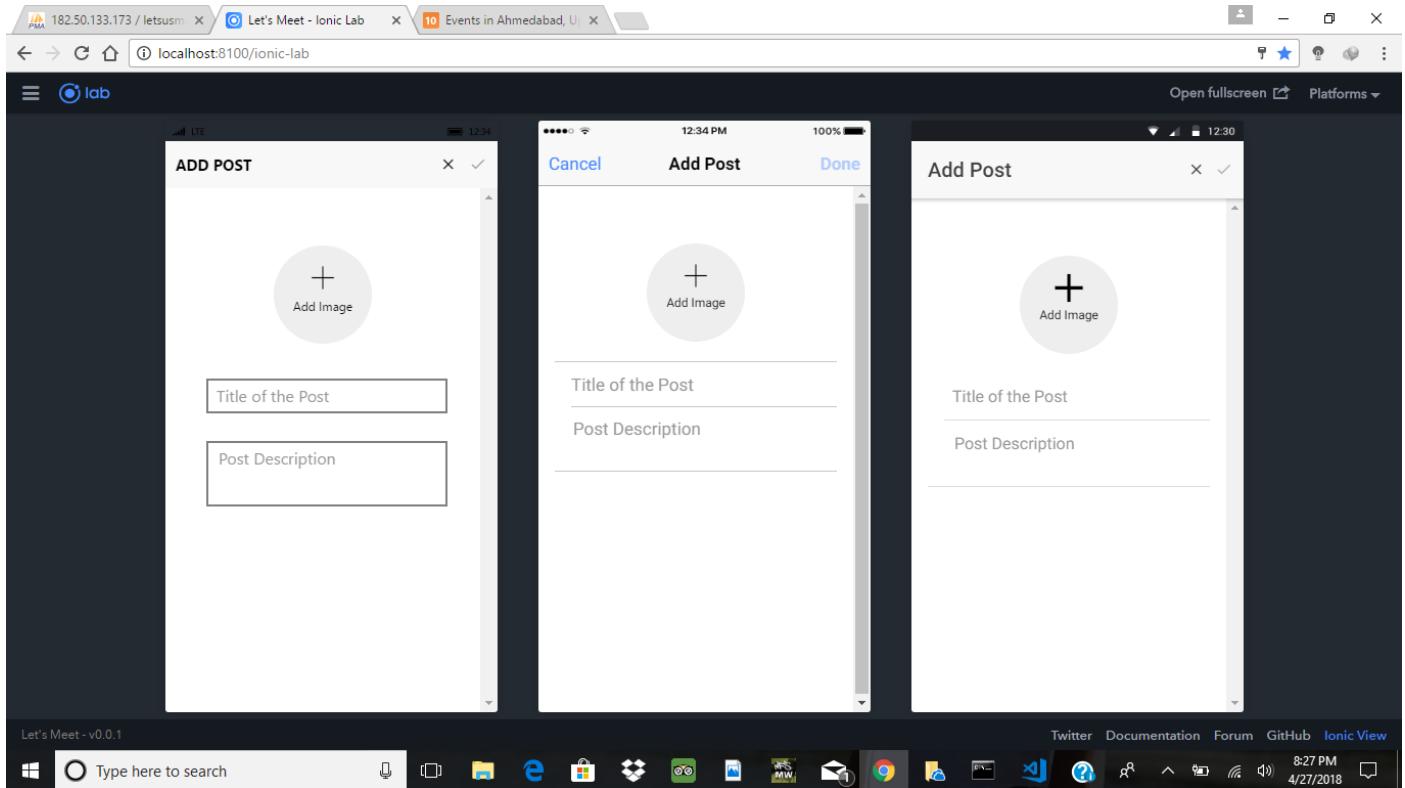
doConfirm() {
  let confirm = this.alertCtrl.create({
    title: 'Leave Community?',
    message: "Are you sure you don't want to be a part of this Community?",
    buttons: [
      {
        text: 'Disagree',
        handler: () => {
          console.log('Disagree clicked');
        }
      },
      {
        text: 'Agree',
        handler: () => {
          this.onRemove();
        }
      }
    ]
  });
  confirm.present()
}
}

```

View Post



Add Post



TypeScript

```
done() {
  if (!this.form.valid) { return; }

  this.storage.get('uid').then((val) => {
    this.user_id = val;
    console.log(this.user_id);
    const fd = new FormData();
    fd.append("post_id", null);
    fd.append("post_title", this.post_title);
    fd.append("post_des", this.post_des);
    fd.append("image", this.selectedFile, this.selectedFile.name);
    fd.append("post_date", null);
    fd.append("post_fk_user_id", this.user_id);
    fd.append("fk_comm_id", this.comm_id);
    this._dataPost.addPost(fd).subscribe(
      (data: Post_Class) => {
        this.viewCtrl.dismiss();
        let t1 = this.tos.create({
          duration: 3000,
          message: "Added ..."
        });
    });
  });
}
```

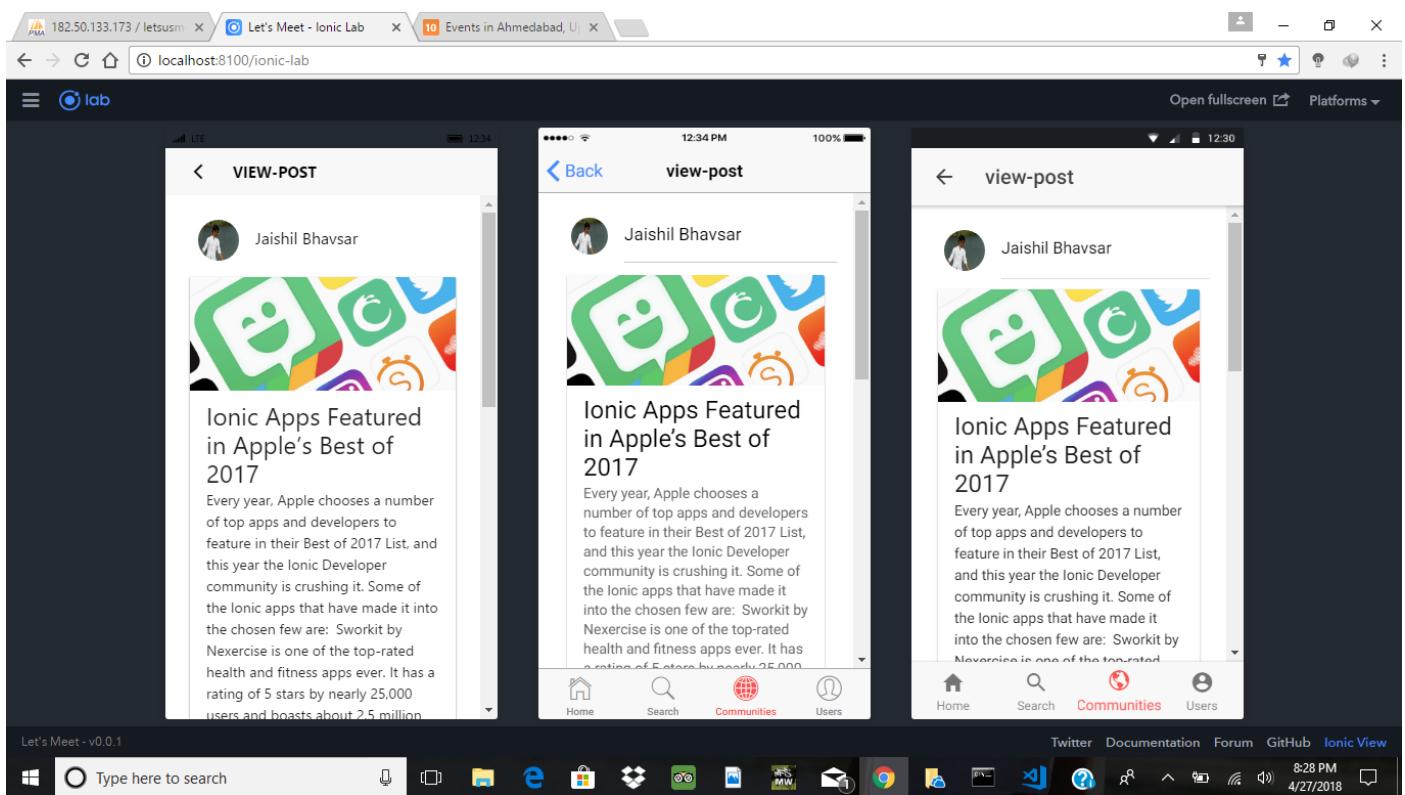
```

    },
    function (err) {
      alert(err);
    },
    function () {
      }
    )
});

}

```

Display Post



TypeScript

```

ionViewDidLoad() {

  console.log('ionViewDidLoad ViewPostPage');

  this.post_id = this.navParams.get('post_id');

  this.storage.get('uid').then((val) => {
    this.user_id = val;
    this._dataUser.getUser(this.user_id).subscribe(
      (data: user_class[]) => {
        this.arrUser = data;
      }
    );
  });
}

```

```

        this.user_name = this.arrUser[0].user_name;
        this.user_pic = this.arrUser[0].user_pic;
        this.u_id = this.arrUser[0].user_id;
    }
);

});

let l1 = this.load.create({
    content: 'Loading...'
});

l1.present();

this._dataPost.getPostById(this.post_id).subscribe(
    (data: Post_Class[]) => {
        this.arrPost = data;
    },
    function (e) {
        alert(e);
    },
    function () {
        l1.dismiss();
    }
);

this._dataComment.getAllCommentsByPostId(this.post_id).subscribe(
    (data: Comment_User_Post[]) => {
        this.arrCommUserPost = data;
        this.comment_count = this.arrCommUserPost.length;
    },
    function (err) {
        alert(err);
    },
    function () {
    }
);

this.storage.get('uid').then((val) => {

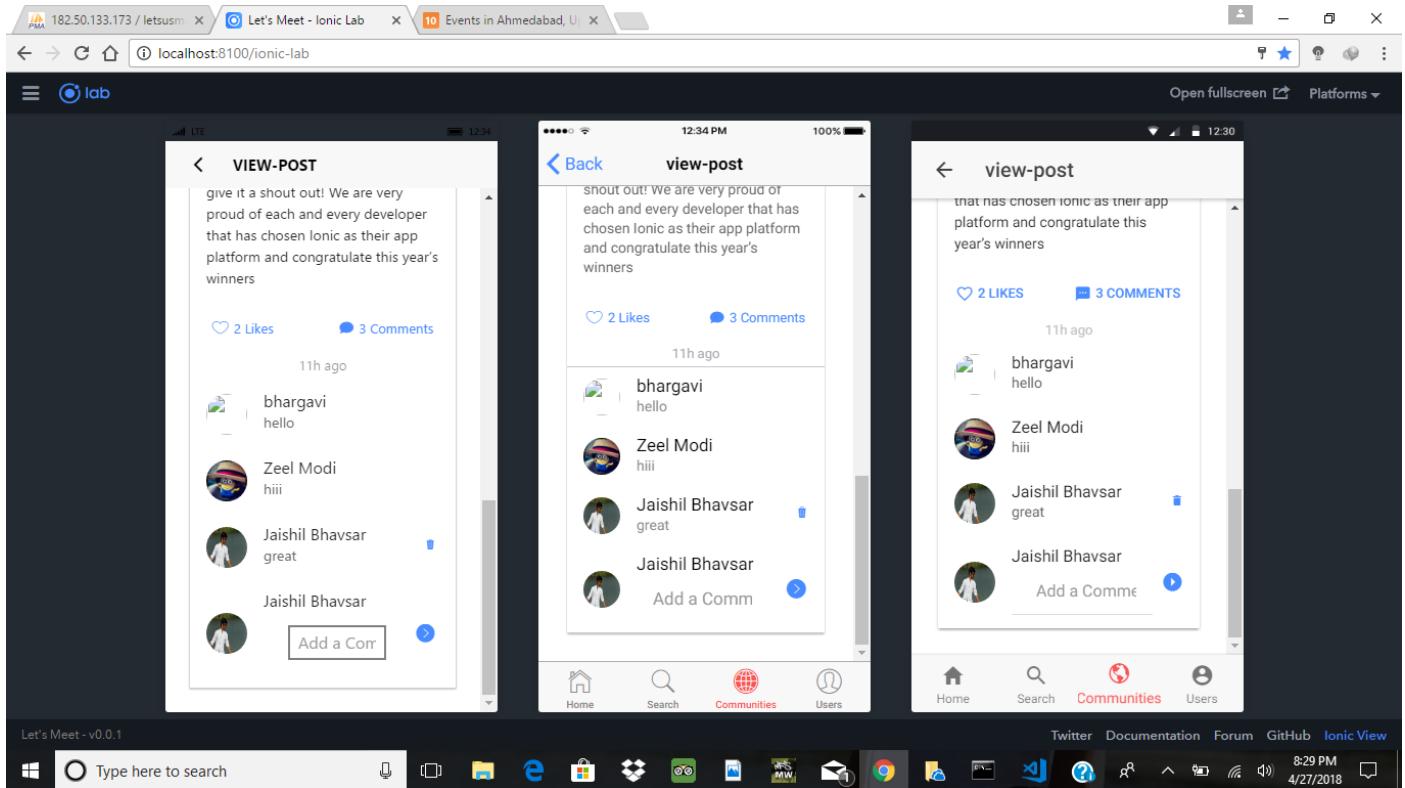
```

```
this.user_id = val;

this._dataLike.getLikeIdByPostAndUser(this.post_id, this.user_id).subscribe(
  (data: any) => {
    if (data == "") {
      this.like = true;
      this.dislike = false;
    }
    else {
      this.like = false;
      this.dislike = true;
      this.arrLike = data;
      console.log(this.arrLike);
    }
  }
);

this._dataLike.getlikesByPosts(this.post_id).subscribe(
  (data: Like_Post_User_Class[]) => {
    this.arrLikePostUser = data;
    this.like_cnt = this.arrLikePostUser.length;
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);
}
```

Add Comment Comment On Post



TypeScript

```

newComment() {
  this.storage.get('uid').then((val) => {
    this.user_id = val;
    console.log(this.comment_date);

    this._dataComment.addComment(new Comment_Class(null, null, this.new_comment,
this.post_id, this.user_id)).subscribe(
      (data: Comment_Class) => {
        console.log(data);
        this.ionViewDidLoad();
        this.new_comment = "";
      },
      function (e) {
        alert(e);
      },
      function () {
        alert("added");
      }
    );
  });
}

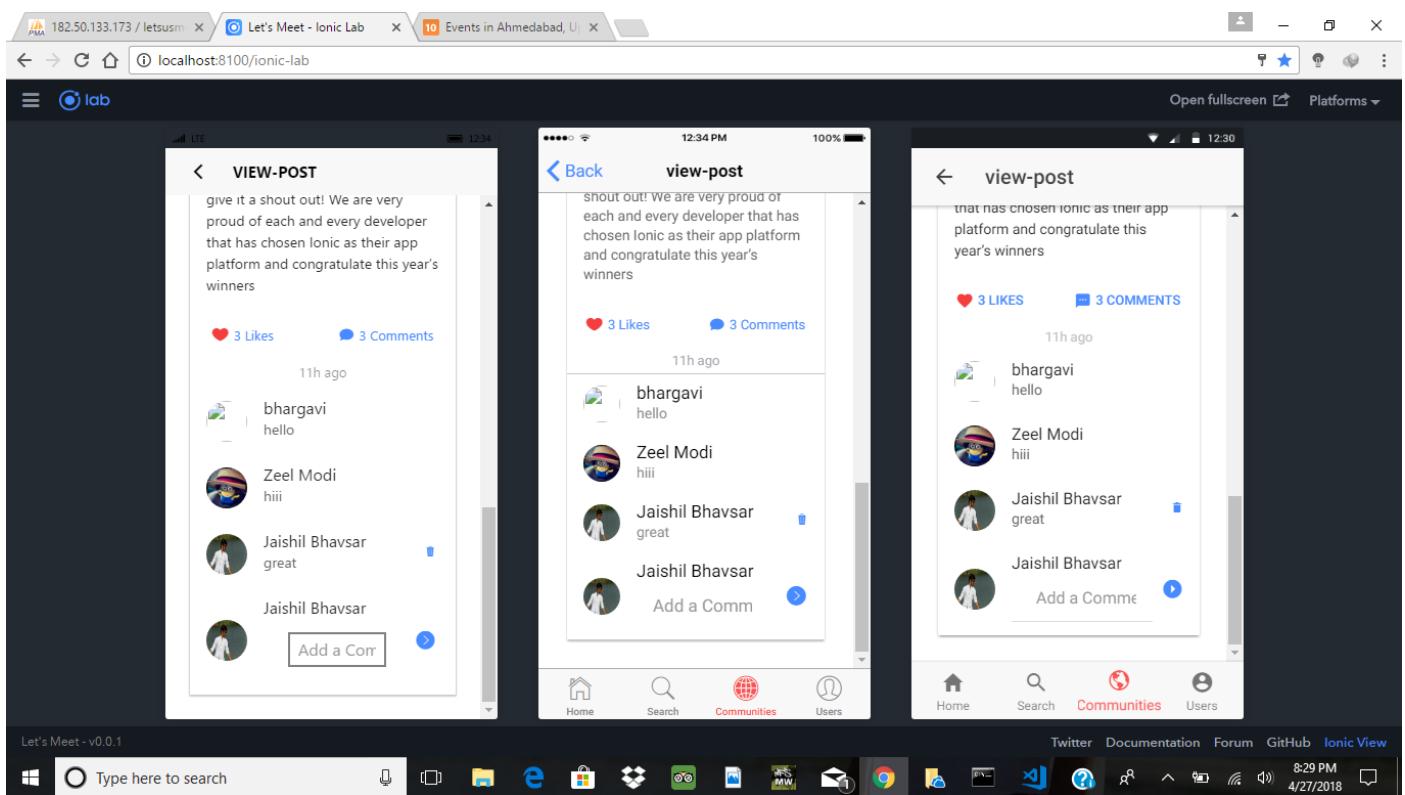
```

```

onDeleteComment(id) {
  this._dataComment.deleteComment(id).subscribe(
    (data) => {
      this ionViewDidLoad();
    },
    function (e) {
      alert(e);
    },
    function () {
      alert("deleted");
    }
  );
}

```

Add Like On Post



TypeScript

```

onClickLike() {
  this.storage.get('uid').then((val) => {
    this.user_id = val;
    let l1 = this.load.create({
      content: 'liking ...'
    });
    l1.present();
    this._dataLike.getLikeIdByPostAndUser(this.post_id, this.user_id).subscribe(

```

```

(data: any) => {
  if (data == "") {
    this.addLike();
  }
  else {
    this.arrLike = data;
  }
},
function (e) {
  alert(e);
},
function () {
  l1.dismiss();
}
);
});
}
}

addLike() {
  let t1 = this.tos.create({
    duration: 3000,
    message: "Liked ..."
  });
  this.storage.get('uid').then(
    (val) => {
      this.user_id = val;
      this._dataLike.addLikes(new Like_Class(null, this.post_id,
this.user_id)).subscribe(
        (data: Like_Class[]) => {
          t1.present();
          this.like = false;
          this.dislike = true;
          this.like_cnt = this.like_cnt + 1;
        },
        function (err) {
          alert(err);
        },
        function () {
          }
        );
    }
  );
}

onLikeRemove() {
  console.log(this.arrLike);
  this.storage.get('uid').then(
    (val) => {
      this.user_id = val;
      this._dataLike.getLikeIdByPostAndUser(this.post_id, this.user_id).subscribe(
        (data: any) => {
          if (data == "") {

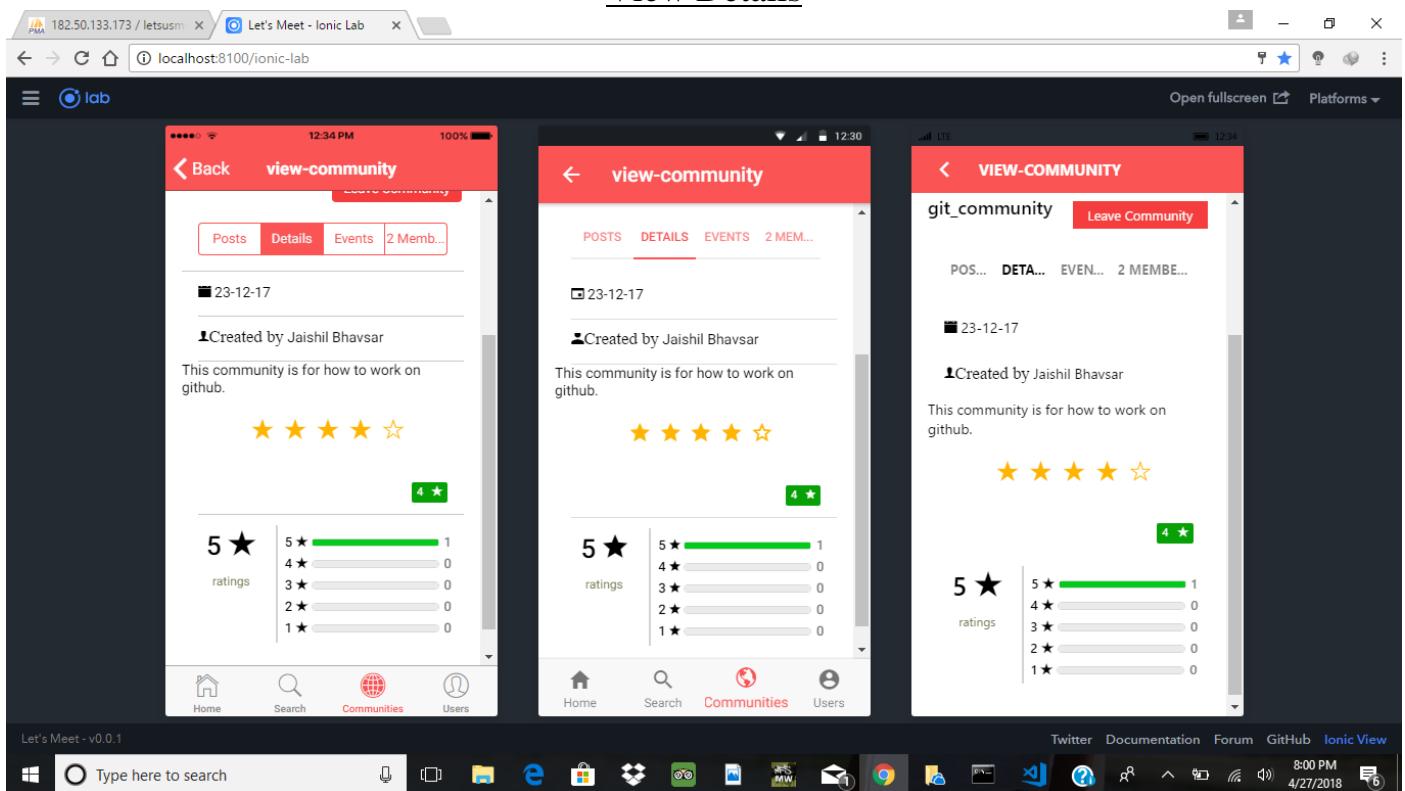
```

```

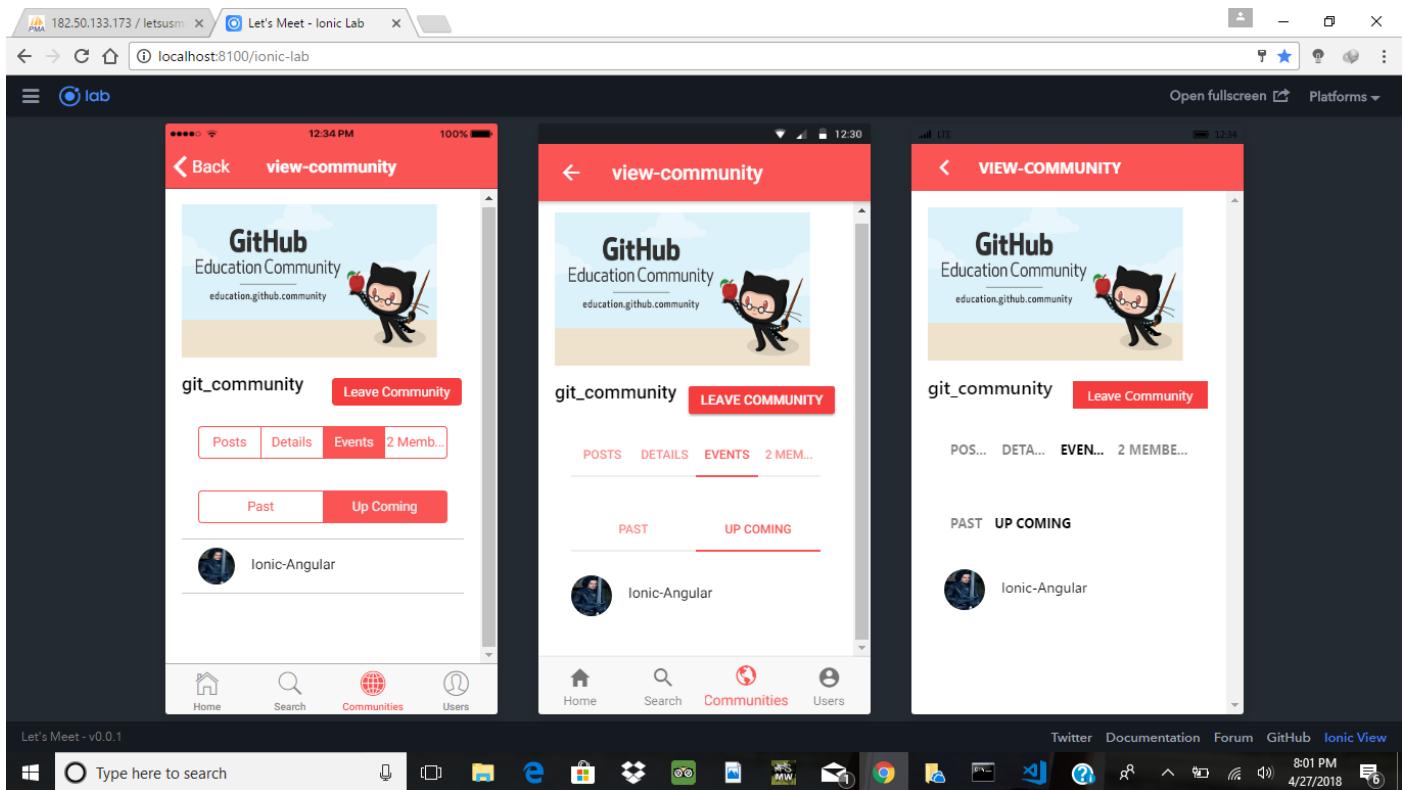
        alert("nathi");
    }
    else {
        this.arrLike = data;
        console.log(this.arrLike);
        this._dataLike.deleteLike(this.arrLike[0].like_id).subscribe(
            (data: Like_Class) => {
                //alert("thyu");
                this.like = true;
                this.dislike = false;
                this.like_cnt = this.like_cnt - 1;
            },
            function (err) {
                alert(err);
            },
            function () {
            }
        );
    }
},
function (e) {
    alert(e);
},
function () {
    alert("disliked");
}
);
});
}
}

```

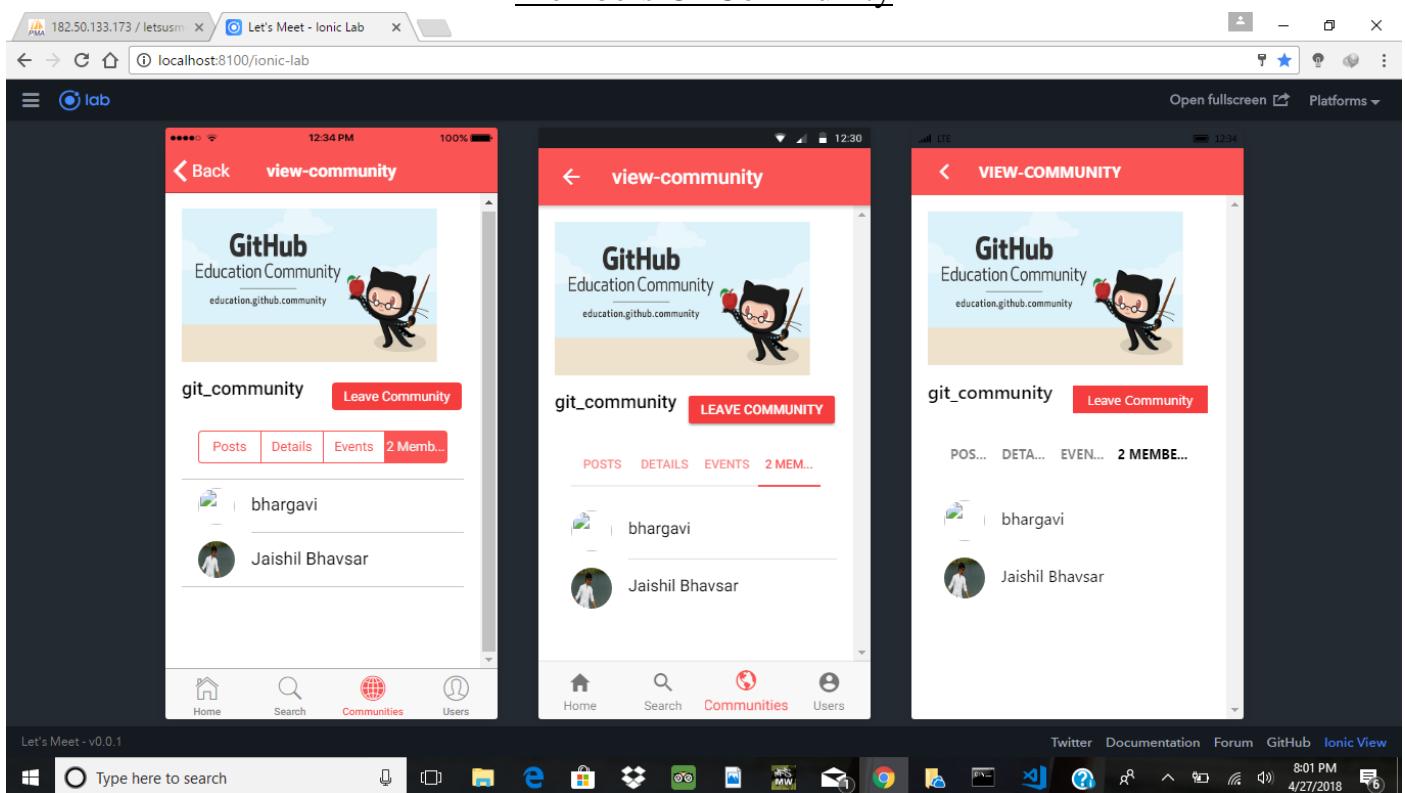
View Details



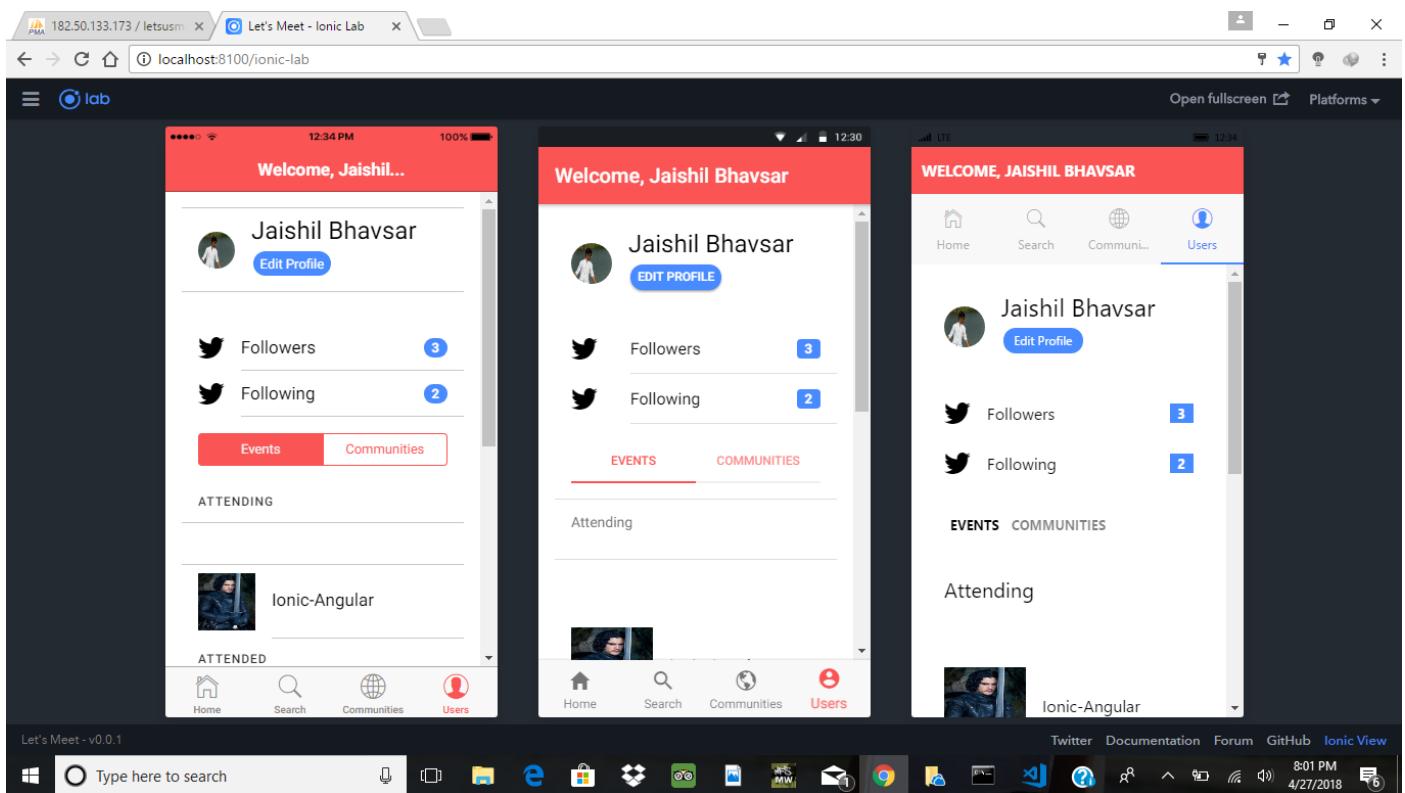
View Event By Community



Members Of Community



User Display And Events By User



TypeScript

```
ionViewDidLoad() {

  this.storage.get('uid').then((val) => {
    this.uid = val;
    let l1 = this.load.create({
      content: "Loading ..."
    });
    l1.present();
    this.data.getUser(this.uid).subscribe(
      (dt: user_class[]) => {
        this.u = dt;
        this.eid = this.u[0].user_name;
        this.img = this.u[0].user_pic;
      },
      function (e) {
        alert(e);
      },
      function () {
        l1.dismiss();
      }
    );
    this.data.getFollowers(this.uid).subscribe(
      (ft: any) => {
        if (ft !== "") {
          this.followers = ft;
        }
      }
    );
  });
}
```

```

        this.followercount = this.followers.length;
    }
},
);
this.data.getFollowing(this.uid).subscribe(
(fl: any) => {
    if (fl != "") {
        this.followings = fl;
        this.followingcount = this.followings.length;
    }
}
);
}

this.storage.get('uid').then((val) => {
    this.uid = val;
    this._dataEvent.getUpcEventRegUser(this.uid).subscribe(
(data: Event_Comm_Rsvp[]) => {
    this.arrUpc = data;
},
function (err) {
    alert(err);
},
function () {

}
);
});

this.storage.get('uid').then((val) => {
    this.uid = val;
    this._dataEvent.getPastEventReg(this.uid).subscribe(
(data: Event_Comm_Rsvp[]) => {
    this.arrPast = data;
},
function (err) {
    alert(err);
},
function () {

}
);
});

this.storage.get('uid').then((val) => {
    this.uid = val;
    this._DataCommu.getcommunitiesofuser(this.uid).subscribe(
(data: Community_Class[]) => {
    this.arrCommu = data;
},
function (err) {
    alert(err);
},

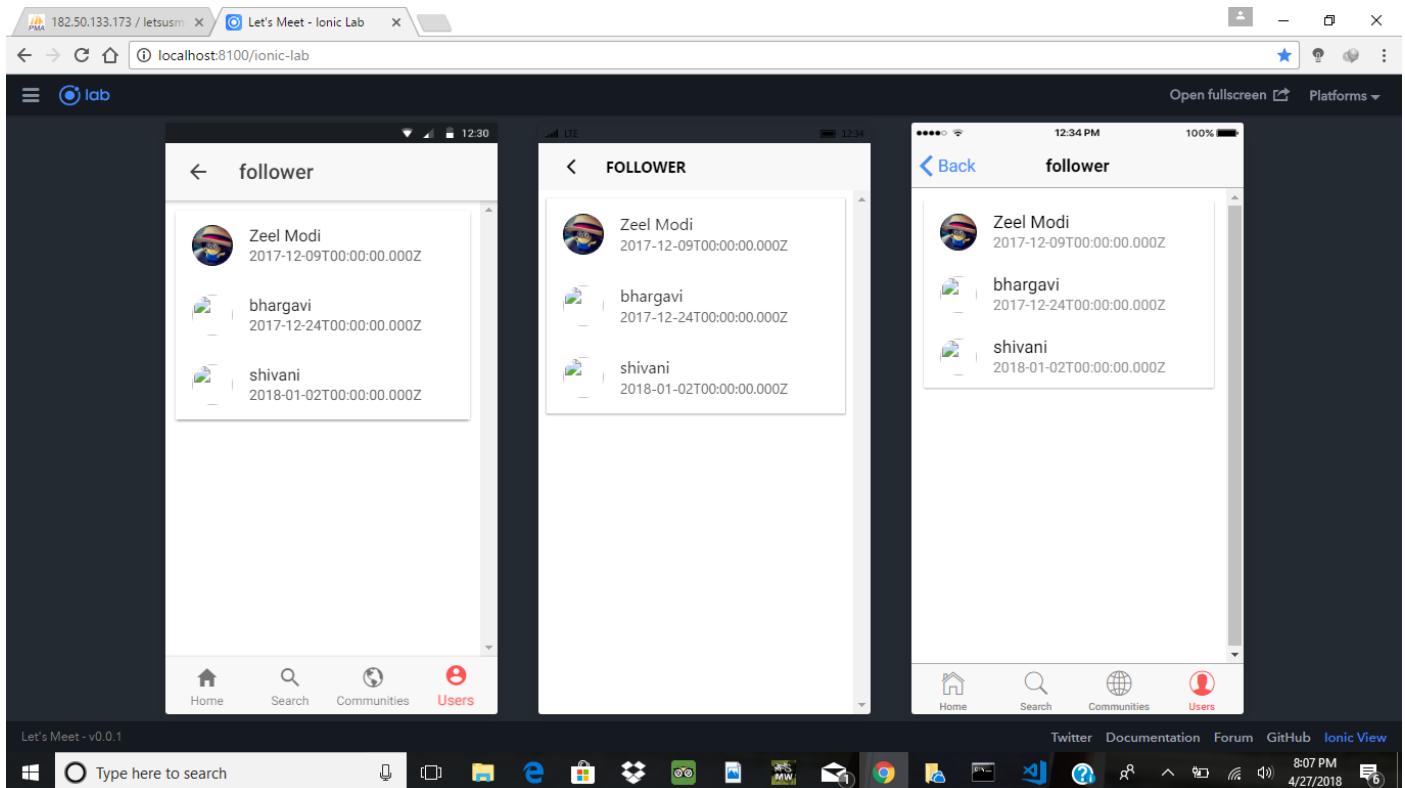
```

```

function () {
    }
);
})
}

```

Followers



TypeScript

```

ionViewDidLoad() {
    console.log('ionViewDidLoad FollowerPage');

    this.uid = this.navParams.get('uid');

    let l1 = this.load.create({
        content: "Loading ..."
    });

    l1.present();

    this.data.getFollowing(this.uid).subscribe(
        (fl: follower_class[]) => {
            if (fl != null) {
                this.follow = fl;
                console.log(this.follow);
                console.log(this.follow.length);
            }
        }
    );
}

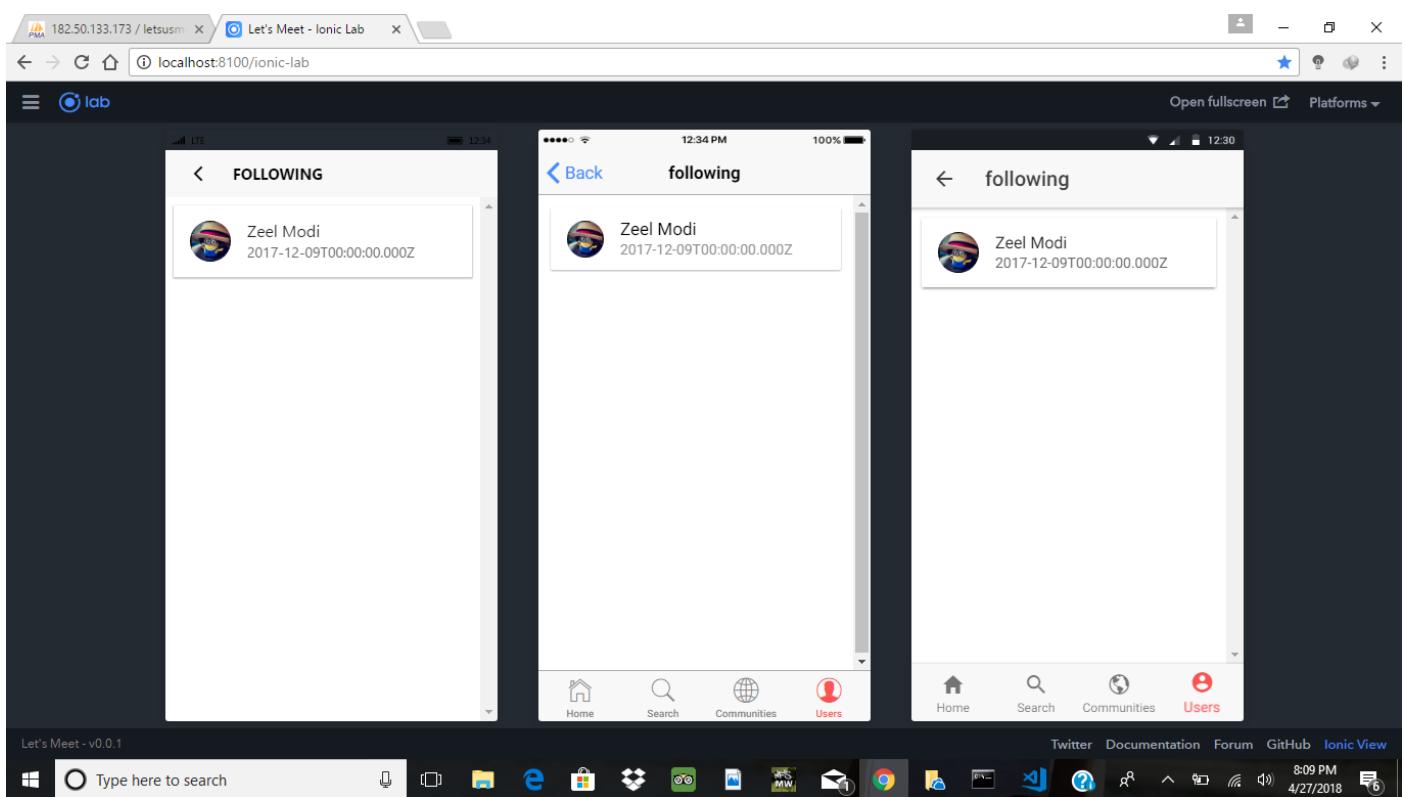
```

```

    this.data.getFollowingUser(this.uid).subscribe(
      (data:any)=>{this.arr=data;},
    );
  },
  function (e) {
    alert(e);
  },
  function () {
    l1.dismiss();
  }
);
}

```

Following



TypeScript

```

ionViewDidLoad()
{
  console.log('ionViewDidLoad FollowingPage');
  this.storage.get('uid').then((val)=>{
    this.uid=val;
    this.uid=this.navParams.get('uid');
    let l1 = this.load.create({
      content: "Loading ..."
    });
  });
}

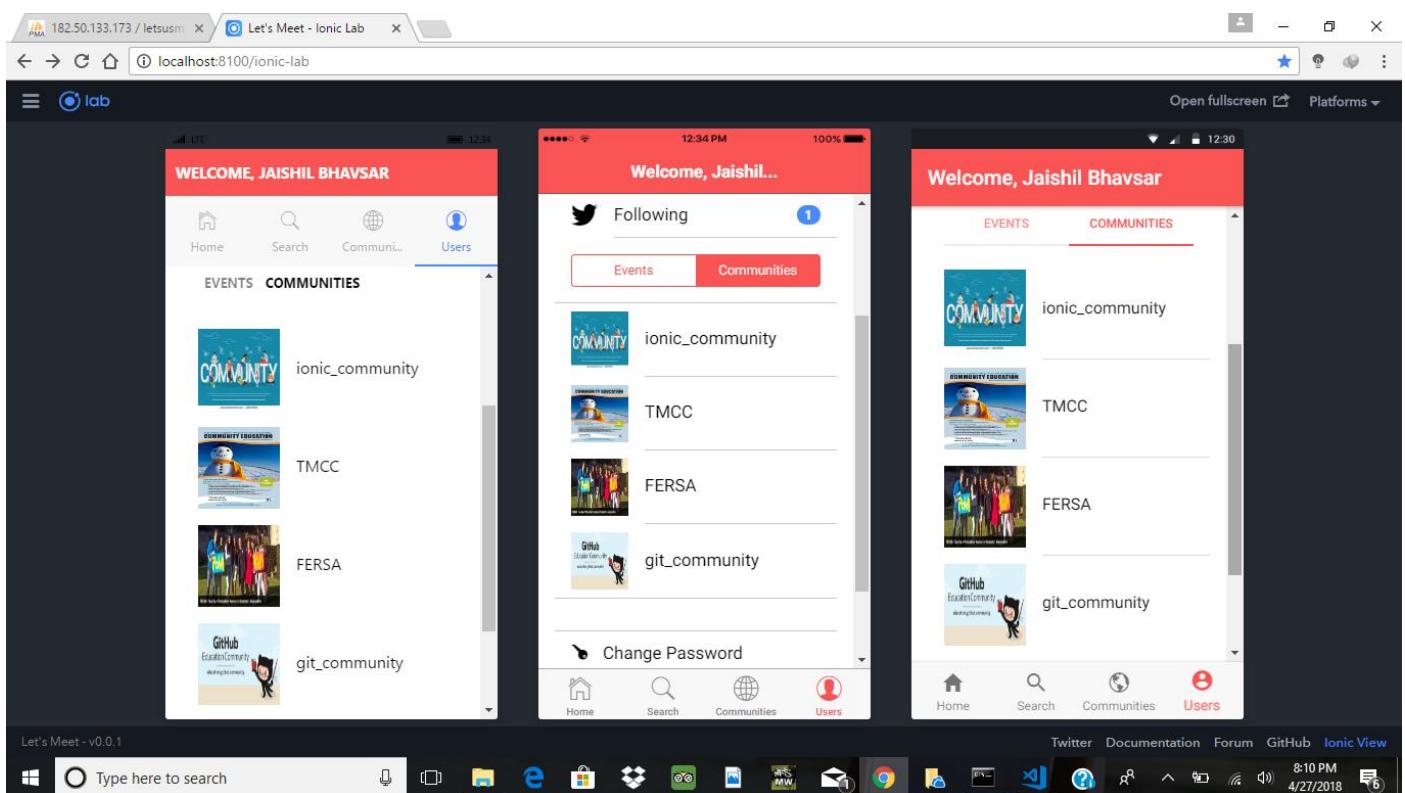
```

```

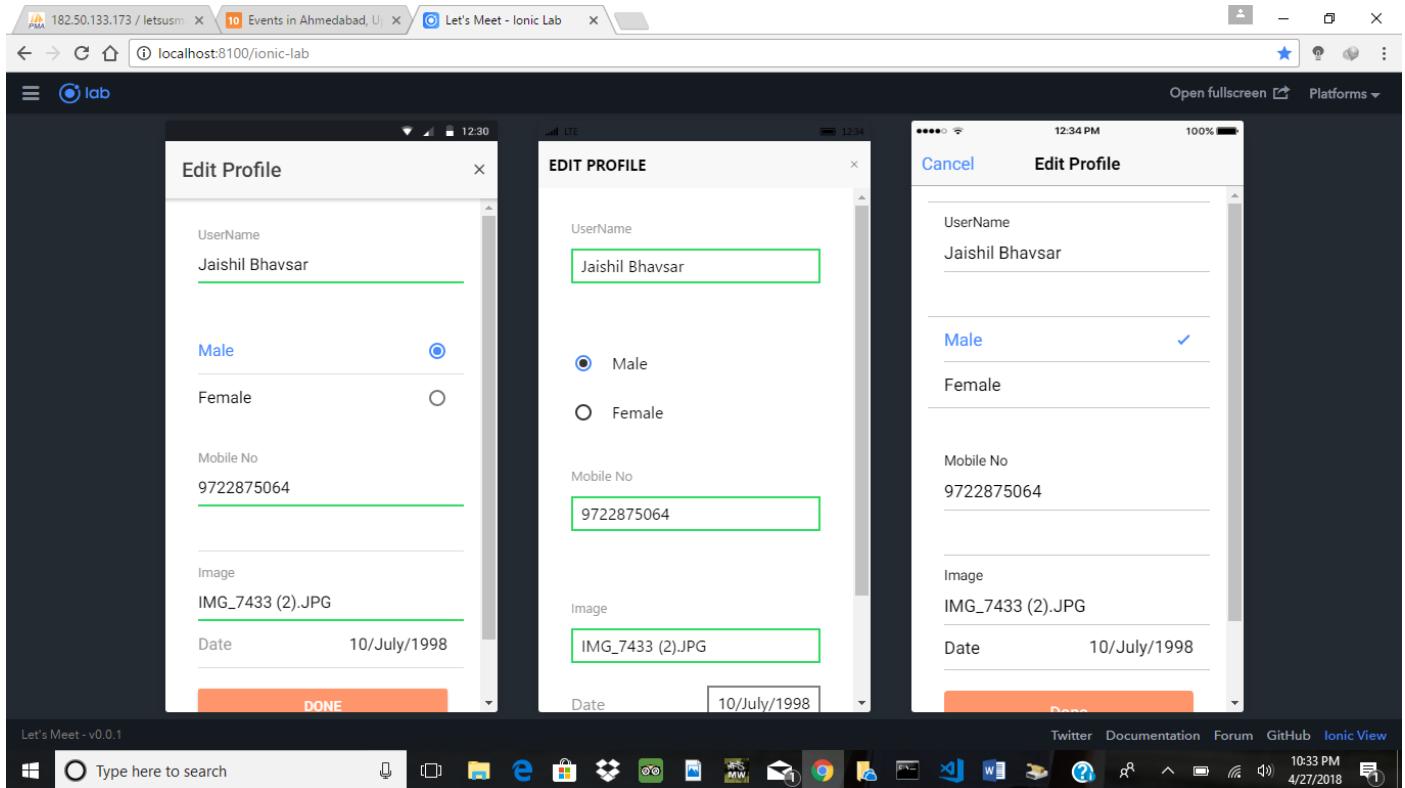
    });
    l1.present();
    this.data.getFollowingwhom(this.uid).subscribe(
      (data:any)=>{this.arr=data;},
      function (e) {
        alert(e);
      },
      function () {
        l1.dismiss();
      }
    )
  );
}
}

```

Communities By User



Edit Profile



TypeScript

```
ionViewDidLoad() {
  console.log('ionViewDidLoad EditprofilePage');
  this.storage.get('uid').then((val) => {
    this.eid = val;
    let l1 = this.load.create({
      content: "Loading ..."
    });
    l1.present();
    this.data.getUser(this.eid).subscribe(
      (dt: user_class[]) => {
        this.u = dt;
        this.id = this.u[0].user_id;
        this.uname = this.u[0].user_name;
        this.image = this.u[0].user_pic;
        this.mydate = this.u[0].user_bdate;
        this.gender = this.u[0].gender;
        this.mobile = this.u[0].user_mob_no;
        console.log(this.mobile);
      },
      function (e) {
```

```
    alert(e);
  },
  function () {
    l1.dismiss();
  }
);

});

}

dismiss() {
  this.navCtrl.dismiss();
}

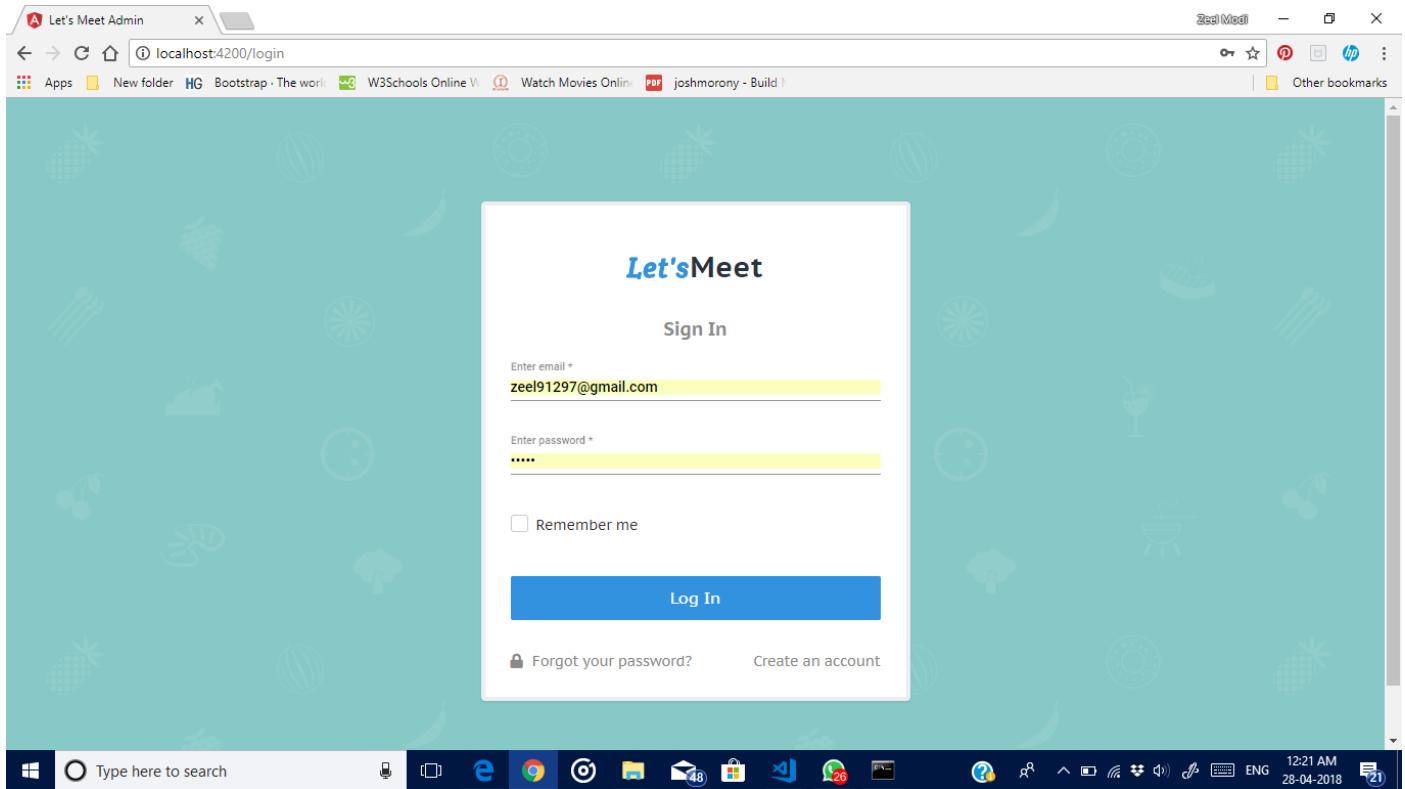
onClick() {
  console.log(this.id);
  this.data.updateUser(this.id, this.uname, this.image, this.gender, this.mobile,
this.mydate).subscribe(
  (dt: any) => {
    if (dt.affectedRows == 1) {
      let mes = this.tos.create({
        message: 'Updated...',
        duration: 4000,
        position: 'top'
      });
      mes.present();
      this.navCtrl.pop();
    }
    else {
      let mes = this.tos.create({
        message: 'Failed...',
        duration: 4000,
        position: 'top'
      });
      mes.present();
    }
  }
}
```

);

}

Admin

Login



TypeScript

```
onLogin(login) {

  this.email_id = login.value.email_id;
  this.pass = login.value.pass;
  console.log(this.email_id);
  console.log(this.pass);
  this._dataUser.doLogin(this.email_id, this.pass, 'admin').subscribe(
    (data1: any) => {
      console.log(data1);
      if (data1.length === 1) {
        localStorage.setItem('u_id', this.email_id);
        this._router.navigate(['/dashboard']);
      } else {
        if (this.email_id.length === 1) {
          if (this.pass.length !== 1) {
            alert('password is wrong');
          }
        } else {
          alert('Incorrect Email and Password');
        }
      }
    },
  );
},
```

```

function (e) {
  alert(e);
}
);
}
}

```

Events List

Check	Event Name	Event Picture	Event Location	Community Name	Event Created By	Action
<input type="checkbox"/>	workshopfor c and ++		Navrangpura	ionic_community	Zeel Modi	Delete Edit View
<input type="checkbox"/>	Genesis 17		Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu	ionic_community	Zeel Modi	Delete Edit View
<input type="checkbox"/>	International Workshop on Android Application Development		Coimbatore Institute of Technology, Auditorium Civil			

Items per page: 1 - 10 of 12 | < < > > |

Delete All

TypeScript

```

ngOnInit() {
  this._dataEvent.getAllEvents().subscribe(
    (data: Event_Community_User_Class[]) => {
      this.arrEvents = data;
      console.log(data);
      this.dataSource = new MatTableDataSource(this.arrEvents);
      this.dataSource.paginator = this.paginator;
      this.dataSource.sort = this.sort;
    },
    function (err) {
      alert(err);
    },
    function () {
    }
  );
}

```

```

applyFilter(filterValue: string) {
  filterValue = filterValue.trim(); // Remove whitespace
  filterValue = filterValue.toLowerCase(); // MatTableDataSource defaults to lowercase
  matches
    this.dataSource.filter = filterValue;
}

deleteEvent(item) {
  this._dataEvent.deleteEvent(item).subscribe(
    (data: any) => {
      this.ngOnInit();
    },
    function (err) {
      alert(err);
    },
    function () {
    }
  );
}

updateEvent(item) {
  this.router.navigate(['/updateEvent', item.event_id]);
}

checkChange(item: Event_Community_User_Class) {

  if (this.delarr.find(x => x === item)) {
    this.delarr.splice(this.delarr.indexOf(item), 1);
  } else {
    this.delarr.push(item);
  }
  console.log(this.delarr);
}

deleteAll() {
  if (confirm('Are you sure you want to delete')) {

    this._dataEvent.deleteAllEvents(this.delarr).subscribe(
      (data: any) => {
        for (this.i = 0; this.i < this.delarr.length; this.i++) {
          this.arrEvents.splice(this.arrEvents.indexOf(this.delarr[this.i]), 1);
          this.ngOnInit();
          console.log('DONE');
        }
        // this.hotel1 = [];
      },
      function (err) {
        console.log(err);
      },
      function () {
      }
    );
  }
}

```

```

    });
}
}

```

Edit Event

The screenshot shows the 'Edit Event' page with the following fields filled:

- Event Name ***: workshopfor c and ++
- Event Description ***: this event shows importance of c and c++
- Event Picture:** A placeholder image labeled 'C/C++'.
- Event Starting Time**: 07:30 AM
- Event Ending Time**: 11:30 AM
- Choose a date**: 4/17/2018
- Event Location ***: Navrangpura

The screenshot shows the 'Edit Event' page with the following additional field:

- Select Community**: ionic_community

At the bottom, there is a red 'Update Event' button.

TypeScript

```

ngOnInit() {

  this._subscription = this.activatedRoute.params.subscribe(
    (para: any) => {
      this.event_id = para['id'];
      console.log(this.event_id);
    }
  );

  this._dataEvent.getEventById(this.event_id).subscribe(
    (data: Events_Class[]) => {
      console.log(data);
      this.arrEvent = data;
      this.event_name = this.arrEvent[0].event_name;
      this.event_des = this.arrEvent[0].event_des;
      this.event_pic = this.arrEvent[0].event_pic;
      this.event_s_time = this.arrEvent[0].event_s_time;
      this.event_e_time = this.arrEvent[0].event_e_time;
      this.event_date = this.arrEvent[0].event_date;
      this.event_loc = this.arrEvent[0].event_loc;
      this.community_id = this.arrEvent[0].fk_comm_id;
    },
    function (err) {
      alert(err);
    },
    function () {
      }
    );
  );

  this._dataCommu.getAllCommunities().subscribe(
    (data: Community_Class[]) => {
      this.arrCommu = data;
    },
    function (e) {
      alert(e);
    },
    function () {
      }
    );
  );
}

getPicture() {
  this.fileInput.nativeElement.click();
}

onFileSelected(value) {
  this.selectedFile = <File>value.target.files[0];
  this.fileFlag = true;
  console.log(value);
}

```

```
onEventUpdate(eventform) {
  if (this.selectedFile === null) {

    // tslint:disable-next-line:max-line-length
    this._dataEvent.updateEventOnly(new Event_update_class(this.event_id,
eventform.value.event_name, eventform.value.event_des, eventform.value.event_s_time,
eventform.value.event_e_time, this.event_date, eventform.value.event_loc,
this.community_id)).subscribe(
      (data: any) => {
        console.log(data);
        this.router.navigate(['/events']);
      },
      function (err) {
        alert(err);
      },
      function () {

      }
    );
  } else {
    const fd = new FormData();
    fd.append('event_id', this.event_id);
    fd.append('event_name', eventform.value.event_name);
    fd.append('event_des', eventform.value.event_des);
    fd.append('image', this.selectedFile, this.selectedFile.name);
    fd.append('event_s_time', eventform.value.event_s_time);
    fd.append('event_e_time', eventform.value.event_e_time);
    fd.append('event_date', this.event_date);
    fd.append('event_loc', eventform.value.event_loc);
    fd.append('fk_comm_id', this.community_id);

    this._dataEvent.editEvent(fd).subscribe(
      (data: any) => {
        console.log(data);
        this.router.navigate(['/events']);
      },
      function (err) {
        alert(err);
      },
      function () {

      }
    );
  }
}
```

Unapproved Event

Unapproved Events

Check	Event Name	Event Picture	Event Location	Community Name	Event Created By	Action
<input type="checkbox"/>	Farm To Fashion		Gujarat University Exhibition Hall, Ahmedabad	Jaishil Bhavsar		Delete ✓ Eye

Items per page: 10 | 1 - 1 of 1 | < < > >|

Delete All

© 2016 WebAdmin - All Rights Reserved.

TypeScript

```

this._dataEvent.getUnapprovedEvents().subscribe(
  (data: Event_Community_User_Class[]) => {
    this.arrEvents = data;
    console.log(data);
    this.dataSource = new MatTableDataSource(this.arrEvents);
    this.dataSource.paginator = this.paginator;
    this.dataSource.sort = this.sort;
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);
  
```

Create Event

The screenshot shows a web application interface for creating an event. On the left, there is a sidebar with navigation links: Dashboard, Events (selected), Communities, Users, and Posts. The main content area has a title 'Create Event'. It contains several input fields:

- Event Name ***: A text input field.
- Event Description ***: A text input field.
- Event Picture:** A file input field with the placeholder 'Choose File' and 'No file chosen'.
- Event Starting Time**: A text input field with placeholder '--:-- --'.
- Event Ending Time**: A text input field with placeholder '--:-- --'.
- Event Date:** A text input field with placeholder 'dd-mm-yyyy'.
- Event Location ***: A text input field.

This screenshot shows the same 'Create Event' page as above, but with additional fields. Below the location field, there is a dropdown menu labeled 'Select Community *'. At the bottom of the form is a large grey button with a white checkmark icon and the text 'Create Event'.

TypeScript

```
ngOnInit() {  
  
    this._dataCommu.getAllCommunities().subscribe(  
        (data: Community_Class[]) => {  
            this.arrCommu = data;  
        },  
        function (e) {  
            console.log(e);  
        }  
    );  
}
```

```

        alert(e);
    },
    function () {
        }
    );
}
}

onFileSelected(value) {
    this.selectedFile = <File>value.target.files[0];
    console.log(value);
}

onAdd(eventform) {
    this.event_name = eventform.value.event_name;
    this.event_des = eventform.value.event_des;
    this.event_s_time = eventform.value.event_s_time;
    this.event_e_time = eventform.value.event_e_time;
    this.event_date = eventform.value.event_date;
    this.event_loc = eventform.value.event_loc;
    this.community_id = eventform.value.community_id;
    const fd = new FormData();
    fd.append('event_id', this.event_id);
    fd.append('event_name', this.event_name);
    fd.append('event_des', this.event_des);
    fd.append('image', this.selectedFile, this.selectedFile.name);
    fd.append('event_s_time', this.event_s_time);
    fd.append('event_e_time', this.event_e_time);
    fd.append('event_date', this.event_date);
    fd.append('event_loc', this.event_loc);
    fd.append('fk_user_id', this.created_by);
    fd.append('fk_comm_id', this.community_id);
    fd.append('event_verify', 'true');

    this._dataEvent.addEvent(fd).subscribe(
        (data: any) => {
            console.log(data);
            this.router.navigate(['/events']);
        },
        function (e) {
            alert(e);
        },
        function () {
        }
    );
}
}

```

Communities List

Community Name	Community Image	Community Created Date	Community Created By	Action
ionic_community		Dec 16, 2017	zeel91297@gmail.com	Delete Edit
TMCC		Dec 7, 2018	jansaribhargavi1@gmail.com	Delete Edit

TypeScript

```
this._dataCommu.getAllCommunities().subscribe(
  (data: Community_Class[]) => {
    this.arrCommu = data;
    console.log(data);
    this.dataSource = new MatTableDataSource(this.arrCommu);
    this.selection = new SelectionModel(true, []);
    this.dataSource.sort = this.sort;
    this.dataSource.paginator = this.paginator;
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);
```

Create Community

The screenshot shows a web application interface for creating a community. The main title is "Create Community". On the left, there is a sidebar with navigation links: Dashboard, Events, Communities, Users, and Posts. The "Communities" link has a dropdown arrow indicating it has sub-options. The main content area contains a form with fields: "Community Name *", "Event Description *", and a file input field for "Community Picture" with the placeholder "Choose File No file chosen". Below the form is a red button labeled "Create Community" with a checkmark icon. At the bottom of the page, there is a copyright notice: "© 2016 WebAdmin - All Rights Reserved." The top of the screen shows browser tabs and a toolbar with various icons.

TypeScript

```
onAddCommunity(commuForm) {
    this.comm_name = commuForm.value.comm_name;
    this.comm_des = commuForm.value.comm_des;

    const fd = new FormData();
    alert(this.created_by);
    fd.append('comm_id', this.comm_id);
    fd.append('comm_name', this.comm_name);
    fd.append('comm_des', this.comm_des);
    fd.append('image', this.selectedFile, this.selectedFile.name);
    fd.append('comm_date', this.comm_date);
    fd.append('comm_rating', this.comm_rating);
    fd.append('created_by', this.created_by);
    fd.append('comm_fk_cat_id', this.comm_fk_cat_id);

    this._dataCommu.addCommunity(fd).subscribe(
        (data: any) => {
            alert('Done');
            this.router.navigate(['/communities']);
        },
        function (err) {
            alert(err);
        },
        function () {
    
```

```

        }
    );
}

```

Users List

<input type="checkbox"/>	bhargaviraval@gmail.com	bhargavi raval	 Female  
<input type="checkbox"/>	jaishil@yahoo.in	jaishil	 Male  

Items per page: 10 | 1 - 10 of 14 | < > >>

© 2016 WebAdmin - All Rights Reserved.

TypeScript

```

this._dataUser.getAllUser().subscribe(
  (data: User_class[]) => {
    this.arrUser = data;
    console.log(data);
    this.dataSource = new MatTableDataSource(this.arrUser);
    this.dataSource.paginator = this.paginator;
    this.dataSource.sort = this.sort;
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);

```

Create User

The screenshot shows a web application interface for creating a new user. The main title is "Create User". The form includes fields for "Email ID *", "User Name *", "Enter your password *", "User Gender" (with radio buttons for Male and Female), "Mobile Number *", "User Picture" (with a "Choose File" button and "No file chosen" message), and "Select User Type *". A red "Create User" button with a checkmark icon is at the bottom. On the left, there's a sidebar with links for "Dashboard", "Events", "Communities", "Users", and "Posts". On the right, there's a user profile for "Zeel Modi Developer". The browser address bar shows the URL "localhost:4200/addUser".

TypeScript

```

this.user_id = userForm.value.user_email;
this.user_name = userForm.value.user_name;
this.user_pass = userForm.value.user_pass;
this.user_gender = userForm.value.user_gender;
this.user_mob_no = userForm.value.user_mob_no;
this.user_bdate = userForm.value.user_bdate;

console.log(this.user_id + '' + this.user_gender + '' + this.user_name + '' +
this.user_pass + '' + this.user_mob_no);

const fd = new FormData();
fd.append('user_id', this.user_id);
fd.append('user_name', this.user_name);
fd.append('user_pass', this.user_pass);
fd.append('image', this.selectedFile, this.selectedFile.name);
fd.append('gender', this.user_gender);
fd.append('user_mob_no', this.user_mob_no);
fd.append('user_bdate', this.user_bdate);
console.log(fd);
this._dataUser.addUser(fd).subscribe(
  (data: User_class) => {
    // alert("added");
    console.log(data);
    this.router.navigate(['/users']);
  },
  function (err) {

```

```

    alert(err);
},
function () {
}
);

```

Posts List

Check	Post Title	Post Date	Posted By	Community	Action
<input type="checkbox"/>	hello1	Dec 1, 2017	jansaribhargavi1@gmail.com	2	Delete Edit
<input type="checkbox"/>	Ionic Apps Featured in Apple's Best of 2017	Jan 1, 2018	zeel91297@gmail.com	1	Delete Edit
<input type="checkbox"/>	Announcing Ionic DevApp	Dec 9, 2017	jaishilbhavsar@yahoo.in	1	Delete Edit
<input type="checkbox"/>	hey	Nov 29, 2017	zeel91297@gmail.com	2	Delete Edit
<input type="checkbox"/>	Game of Thrones	Mar 9, 2018	jaishilbhavsar@yahoo.in	3	Delete Edit
<input type="checkbox"/>	Welcome to Angular World!	Mar 22, 2018	zeel91297@gmail.com	5	Delete Edit
<input type="checkbox"/>	Again Demo	Mar 22, 2018	zeel91297@gmail.com	4	Delete Edit

TypeScript

```

this._dataPost.getAllPosts().subscribe(
  (data: any[]) => {
    this.arrPost = data;
    console.log(data);
    this.dataSource = new MatTableDataSource(this.arrPost);
    this.dataSource.paginator = this.paginator;
    this.dataSource.sort = this.sort;
  },
  function (err) {
    alert(err);
  },
  function () {
}
);

```

Edit Post

The screenshot shows a web application interface for managing posts. On the left, there's a sidebar with navigation links: Dashboard, Events, Communities, Users, and Posts. The main content area is titled "Edit Post" and contains fields for "Post Title" (Ionic Apps Featured in Apple's Best of 2017), "Post Description" (Every year, Apple chooses a number of top apps and developers to feature in their Best of 2017 List, and this year the Ionic Developer community is crushing it. Some of the Ionic apps that have made it into the chosen few are:), and "Event Picture" (a small image of colorful geometric shapes). Below these fields are "Choose File" and "Update Post" buttons. At the bottom of the page, it says "© 2016 WebAdmin - All Rights Reserved." The status bar at the very bottom shows various system icons and the date/time as 10:09 PM 27-04-2018.

TypeScript

```

this._dataPost.getPostById(this.post_id).subscribe(
  (data: Post_Class[]) => {
    this.arrPost = data;
    this.post_title = this.arrPost[0].post_title;
    this.post_des = this.arrPost[0].post_des;
    this.post_pic = this.arrPost[0].post_pic;
    console.log(this.post_pic);
  },
  function (err) {
    alert(err);
  },
  function () {
  }
);

onUpdatePost(postForm) {
  this.post_title = postForm.value.post_title;
  this.post_des = postForm.value.post_des;

  if (this.selectedFile === null) {

    this._dataPost.editPostOnly(new Post_Update_Class(this.post_id, this.post_title,
this.post_des)).subscribe(
      (data: any) => {
        console.log(data);
      }
    );
  }
}

```

```
        this.router.navigate(['/posts']));
    },
    function (err) {
        alert(err);
    },
    function () {

    }
);

} else {
    const fd = new FormData();
    fd.append('post_id', this.post_id);
    fd.append('post_title', this.post_title);
    fd.append('post_des', this.post_des);
    fd.append('image', this.selectedFile, this.selectedFile.name);

    this._dataPost.editPost(fd).subscribe(
        (data: any) => {
            console.log(data);
            this.router.navigate(['/posts']);
        },
        function (err) {
            alert(err);
        },
        function () {

        }
    );
}
}
```

TypeScript

```

onAddPost(postForm) {

    this.post_title = postForm.value.post_title;
    this.post_des = postForm.value.post_des;

    const fd = new FormData();
    fd.append('post_id', null);
    fd.append('post_title', this.post_title);
    fd.append('post_des', this.post_des);
    fd.append('image', this.selectedFile, this.selectedFile.name);
    fd.append('post_date', null);
    fd.append('post_fk_user_id', this.user_id);
    fd.append('fk_comm_id', this.community_id);
    this._dataPost.addPost(fd).subscribe(
        (data: Post_Class) => {
            // alert("added");
            console.log(data);
            this.router.navigate(['/posts']);
        },
        function (err) {
            alert(err);
        },
        function () {

        }
    );
}

}

```

Search Event

The screenshot shows the 'Events' section of the Let's Meet Admin application. The interface includes a sidebar with links for Dashboard, Events, Communities, Users, and Posts. The main content area features a table with columns for Check, Event Name, Event Picture, Event Location, Community Name, Event Created By, and Action. A filter bar at the top allows searching by event name, currently set to 'C++'. One event is listed in the table:

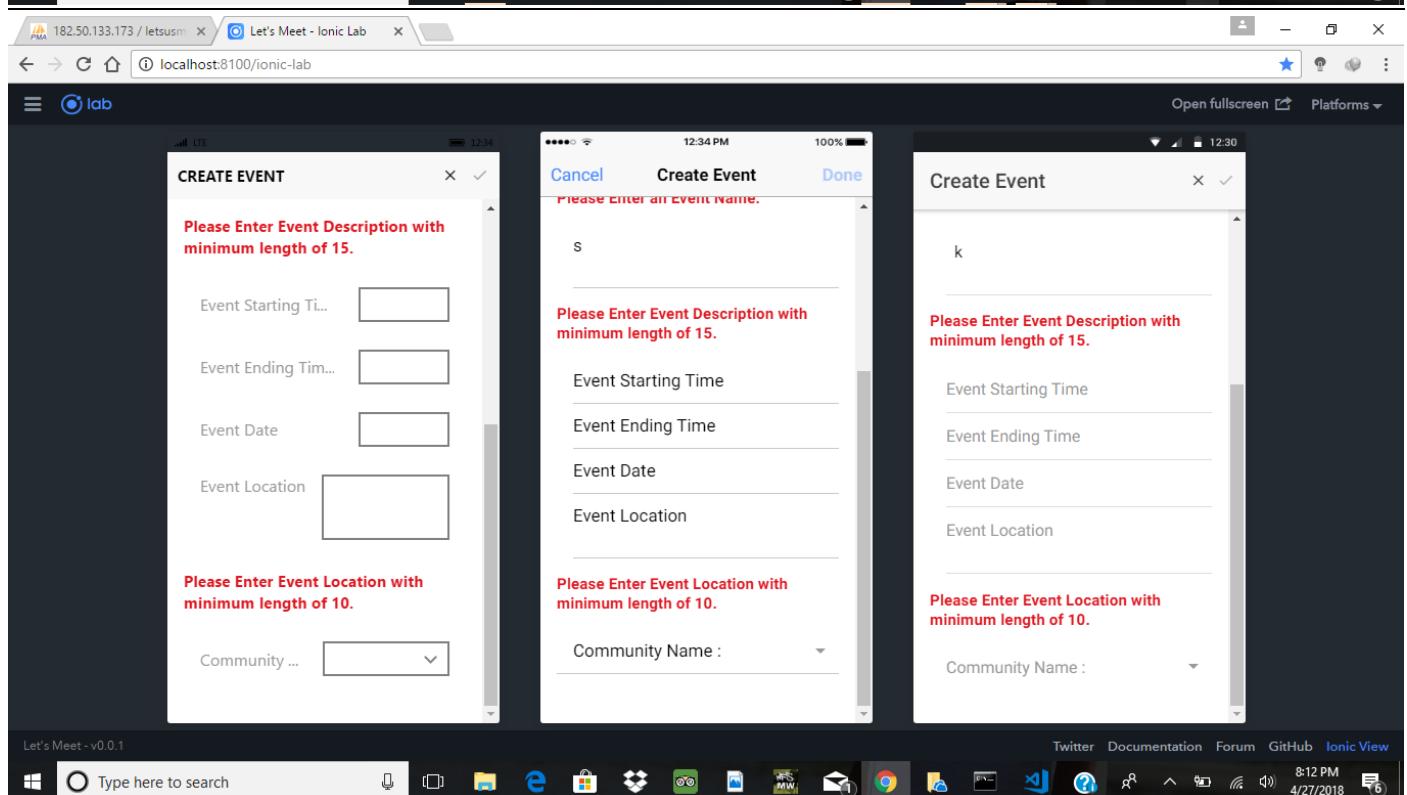
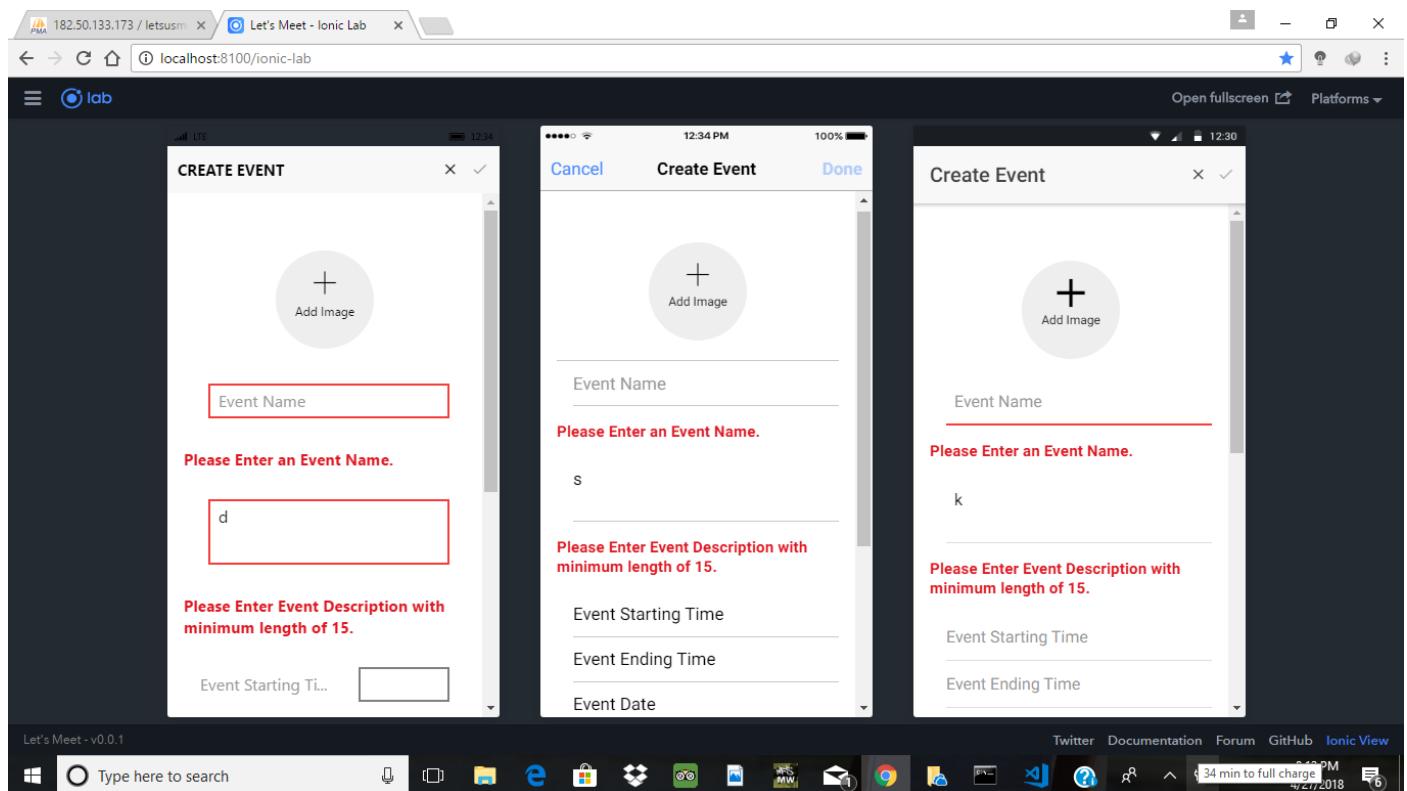
Check	Event Name	Event Picture	Event Location	Community Name	Event Created By	Action
<input type="checkbox"/>	workshopfor c and ++		Navrangpura	ionic_community	Zeel Modi	Delete Edit View

At the bottom of the page, there is a 'Delete All' button and a footer note: '© 2016 WebAdmin - All Rights Reserved.'

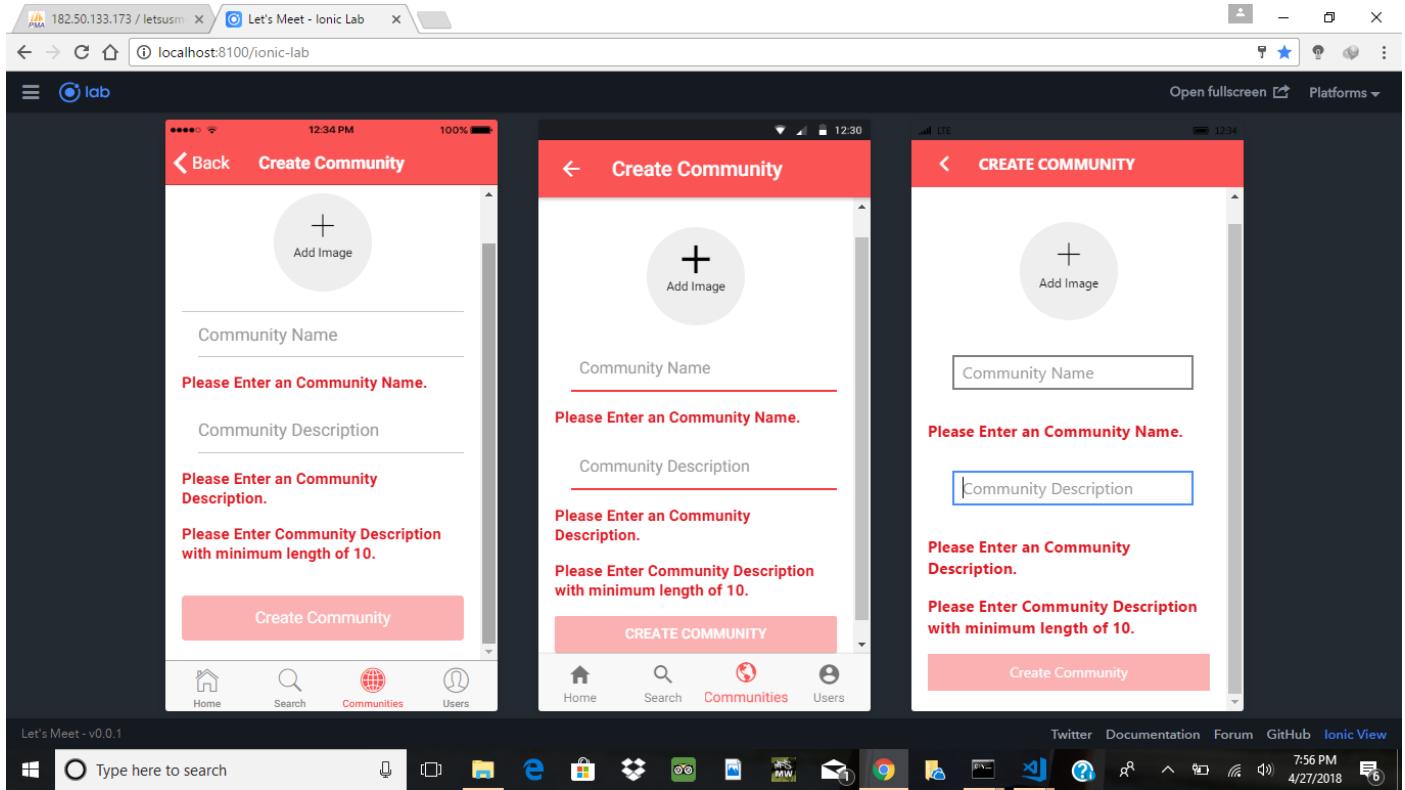
Chapter 6: Testing

User

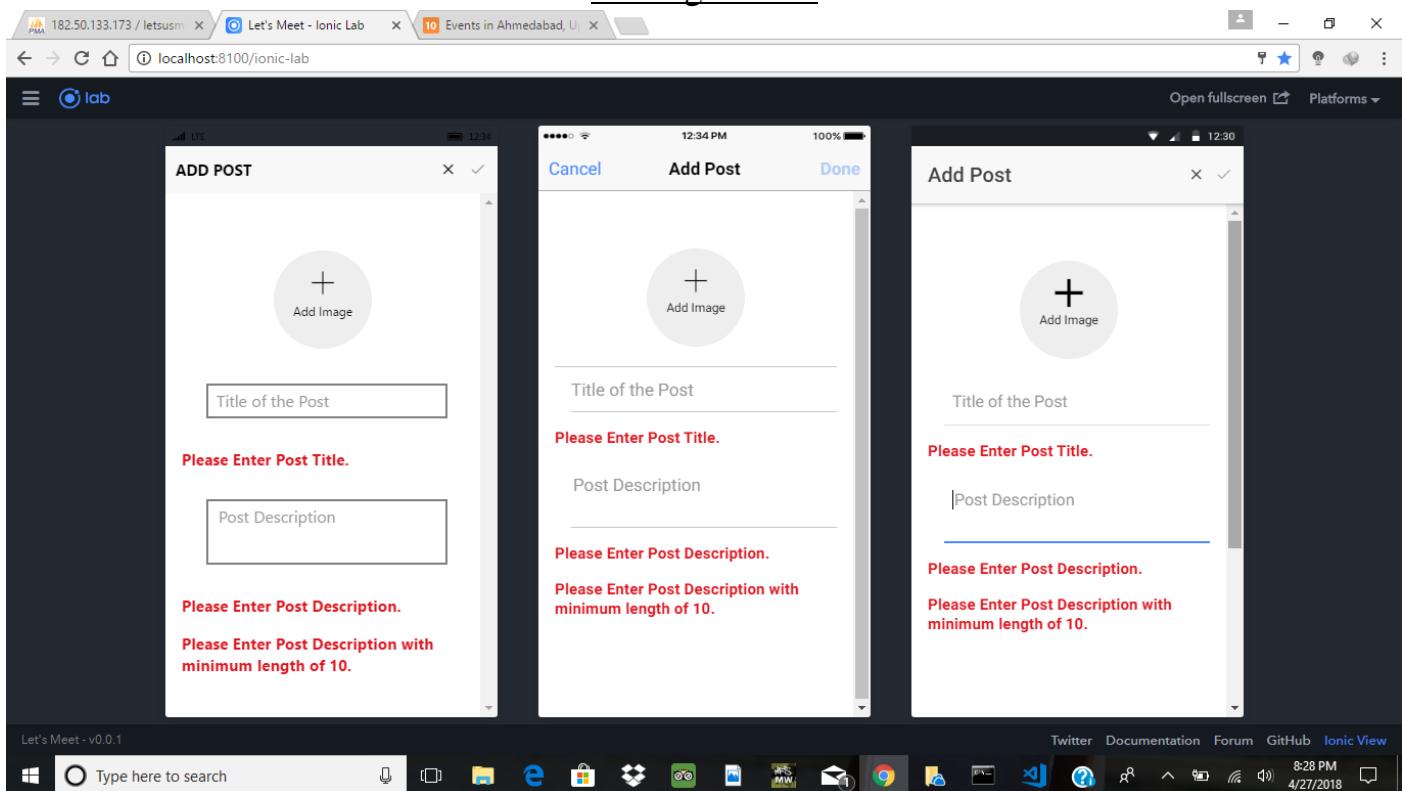
Testing Event



Testing of Community

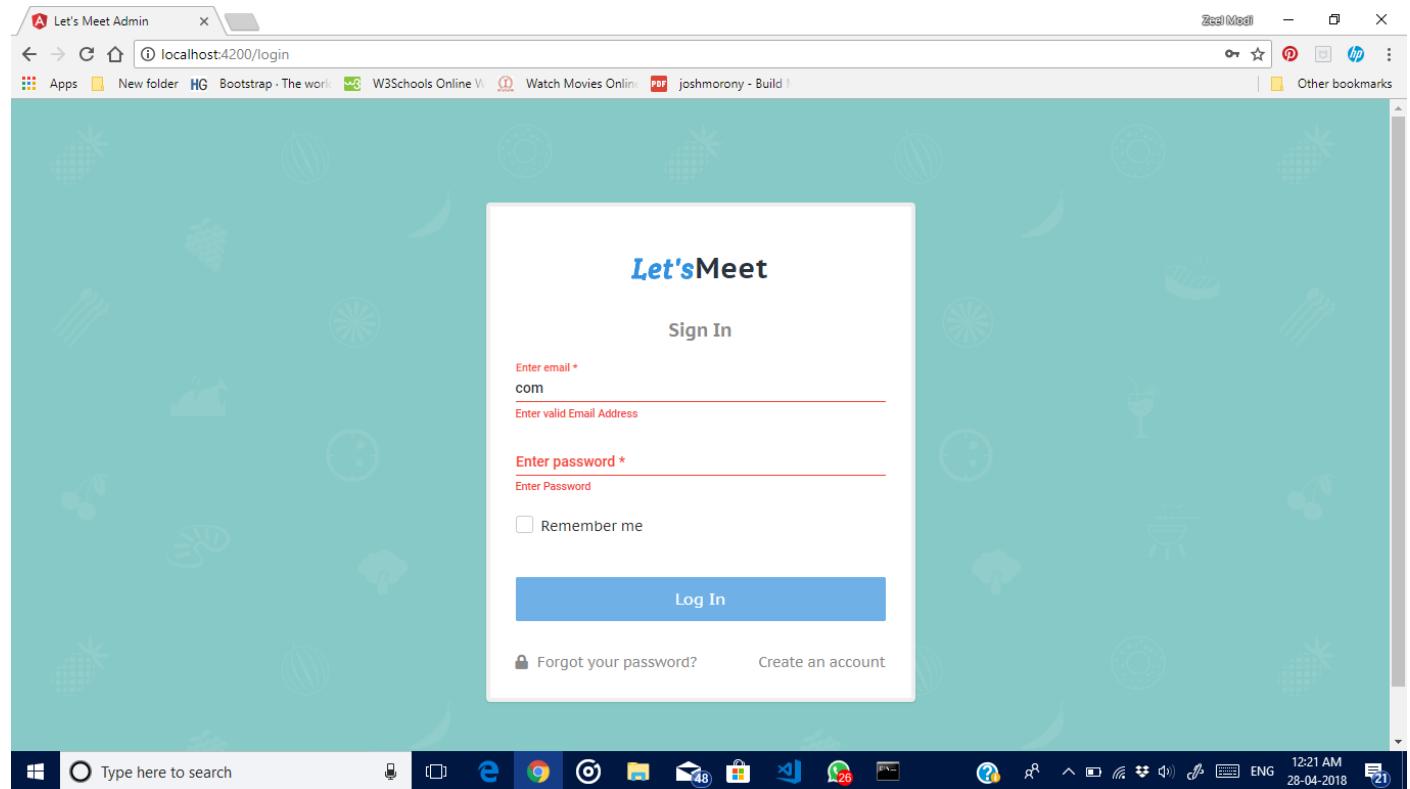


Testing of Post



Admin

Testing of Login



Testing of Event

Create Event

Event Name *
Please Enter Event Name

Event Description *
Please Enter Event Description

Event Picture:
Choose File No file chosen

Event Starting Time
--:-- --
Please Enter Starting Time

Event Ending Time
--:-- --

Event Date:
dd-mm-yyyy
Please Enter Event Date

Event Location *
Please Enter Event Location

Please Enter Event Description

Event Picture:
Choose File No file chosen

Event Starting Time
--:-- --
Please Enter Starting Time

Event Ending Time
--:-- --

Event Date:
dd-mm-yyyy
Please Enter Event Date

Event Location *
Please Enter Event Location

Select Community *
Please Select a Community

Create Event ✓

© 2016 WebAdmin - All Rights Reserved.

Testing of Community

Community Name *

Please Enter Community Name

Event Description *

Please Enter Event Description

Community Picture:

Choose File No file chosen

Create Community ✓

© 2016 WebAdmin - All Rights Reserved.

Testing of Post

Post Title *

Please Enter Post Title

Post Description *

Please Enter Post Description

Event Picture:

Choose File No file chosen

pimg is required

Select Community *

Please Select a Community

Create Post ✓

© 2016 WebAdmin - All Rights Reserved.

Chapter 7: Summary

[7.1] Limitations

- User must have to sign-up or login to access our system.
- User has to login using email id only not by registered mobile number.
- If user upload/share his very important document then there is no provision for security.
- If the user might not able to deal with English language then user might not able to use the system efficiently.
- User must be a member of any community to create event.

[7.2] Future Scope

- There are chances that system can cooperate with payment gateways to generate revenue.
- There are chances that user can register in event directly through our system.
- There are chances that users can directly communicate with each other and as well as with speakers through our system.
- There are chances that user can login in our system using fingerprint.

Bibliography

- SOFTWARE ENGINEERING a practitioner's Approach Seventh Edition by Roger S. Pressman
- <https://angular.io/>
- <https://ionicframework.com/>