A location based social network for mobile

Group - 12

Submitted by - Meet Sinojia (201601126)

Group members :

Meet Sinojia(201601126)
Hardik Modi(201601089)
Detroja Smit(20160113)
Mudra Desai(201601098)
Harsha Mahaver(201601080)
Gautam Radadiya(201601115)
Yash Panwar(201601110)

System Requirements Specification

Table of Contents

Version 2.0

Revision History

Name	Date	Reason For Changes	Version
Group 12	25-02-19	First Draft	1.0
Group 12	02-03-19	Added page numbers Modified MoM format Updated non-functional requirements	1.1
Group 12	12-04-19	Updated product scope Updated user interface Updated features Updated user stories Updated MoM	2.0

1. Introduction

1.1 Purpose

The purpose of this document is stated as below :

- To provide a proper description of the requirements for the app "A location based social network for mobile" which allows communication between people in the groups based on their location.
- To connect users individually for personal chats in nearby locations.
- To explain how we derived the requirements from the different source of inputs and how we prioritised them by using different prioritising techniques and how they correlate to fulfil our purpose of making this project.
- To explain user interface for the users and assumptions we made for our project.

1.2 Product Scope

As this is a social networking app based on the location, people with the same interest can join the same groups. This app will be used for the fun activities as well as for the formal activities too. For example any educational institute can create their groups for providing a large variety of learning environments and learning spaces. This app will provide a dynamic product ads for people who are having commerce platforms. Health centers and sports centers can make full use out of it to increase their approachability. This app will be helpful for the users for attending conferences conducting in their cities.

1.3 Overview

Section 2 : General description, users and stakeholders, assumptions and constraints
Section 3 : Requirement Analysis, prioritization and process model
Section 4 : Functional And Non-functional requirements
Section 5 : User Stories
Section 6 : Effort Estimation
Section 7 : Minutes of Meetings

1.4 References

Article for basic design of the app.
https://medium.com/swlh/how-to-develop-a-chat-app-like-whatsapp-e695257320f4

IEEE paper for the handling range related queries
https://ieeexplore.ieee.org/document/5088957

A video describing the basics of any chatting/messaging system
https://www.youtube.com/watch?v=vvhC64hQZMk&feature=youtu.be

2. Overall Description

2.1 General Description

This project is an android APP application which facilitates the creation and joining of local groups via their smartphones. This application is a location based social network for the people of same interest. This app will use GPS of the smartphone to discover and build the groups in their local area. All the groups of some particular fixed range will be shown to the users present in that range.

This has a number of potential use cases, from creating groups at the start of pub crawls to creating interest groups at conferences. The user can find their nearby groups by signing up/login in the app. For signing up in the app, user need to fill up some details like user's full name, user_name, e-mail address, contact number, password and interests etc. All the details are need to be filled by the user. Once they create an account, all the nearby groups will be shown and there will be an option to join the existing group or create a new group. There will be an option of logout from the app. To use the app again after logging out from the app user need to enter his user_name and password every time. In the case user forgot the password, this app will provide an option of forgot password whenever user is on the login page. If user clicks there, he would be asked to enter his email address and through that verification will be done. There will be no constraints in the usability of this application that means anyone who is interested can use the app. User don't need to send any specific instructions to every potential member to join any group of their interest. Individuals can create/join the group according to their interest. There will be an option for individual chat if user wants to do. For both the type of chats (individual chat and group chat) there will be two tabs. If a user has provided his/her contact number, there will be a feature that another user in the same range can contact to that user. When a user clicks on that number, calling tab will be opened and user can call another user.

2.2 Users And Stakeholders

Stakeholders :
The stakeholders are people who are particularly involved in proper functioning and success of the software.

- Administrators
- Educational centers
- Sports centers
- Group admin
- Health centers
- Project Manager

Users :
The users of the software would be any citizens of a particular area. All the stakeholders will be users but not all users will be stakeholders.

- Individuals using the application

2.3 User Interface And Technologies To Be Used :

User Interface :

We have decided to keep user interface as simple and easy to understand as possible. And google's material design to make it attractive. On startup user have signup/login screen on the basis of whether he is signed in or not. If he is already signed in then redirected to home page which has two tabs, one with personal chat and second with group chat. Each tabs has list of persons and groups respectively in the location of user. We will be using drawer for all other routes like profile page, change password, find groups, etc. User will be redirected to that particular page when he clicks on it. Profile page has all information user had entered during signup and can easily edit this information. Find group contains the list of groups that user had not joined but are in the locality of that user. User can view the information of that group by clicking on it. We will also show him the user and group in his locality on the map.

Technologies used :

We are using flutter for mobile app development so that it can easily work on both android devices. Node.js framework for backend development. After healthy discussion we arrived at the conclusion to use NoSQL database - MongoDB to store all chats and information of user. And for API development, we are giving try to GraphQL standards instead of REST, GraphQL subscriptions for chatting which internally is nothing but sockets.

2.4 Assumptions And Constraints :

- Robust internet connectivity in the smartphones.
- GPS should be supported and enabled in the users smartphone.
- Location of the group is fixed.
- Every group should have unique group id.
- Every individual has a unique username.
- The third party server is assumed to be working 24x7

2.5 Open Issues :

- The chats are not recommendation based which can be incorporated in future.
- The feature of deleting messages sent by mistake.
- The feature of sending messages other than text messages is also not incorporated.
- There is no feature to join the group via group links or any such medium.

3. Requirement Elicitation

3.1 Analysis of Existing Systems:

Mobile based Social network apps are an example of social software that are used as connecting billions of end users and communicating with others at anytime and anywhere. We analysed existing mobile based social applications, for example, Facebook, WhatsApp, Telegram which can help us in gathering and prioritizing our requirements.

In order to identify the most suitable social networking application for our project, we performed an initial comparison after defining the key requirements for Mobile based social network. We investigated which of their features are used to support the predefined key requirements. We find out that some of the features that we want in our social networking application are already present in WhatsApp like group chatting and individual chatting. We don't want that that the application we are making is too different from the ones which are present today because people don't accept new things easily.

3.2 Brainstorming

Brainstorming is the requirement elicitation technique we used. It was very informal and at first we had given different ideas about the requirements. Some had gone deep down into the problem and try to evolve some of the requirements.

Some suggested that we should keep individual chats only and other suggested that adding group chats to our application will be a good feature. Some also suggested to implement a feature to send images to individuals or groups and adding calling feature to our application. Some came with the basic requirement to verify user before accessing the main features of the application.

Then we have taken one by one every requirement and tried to debate on various pros and cons of implementing the requirement. Unique features that we come up with after brainstorming are we have decided to allow the user to rate the group he is part of, so that other user can easily check the rating and decide whether group is useful for him or not. We also agreed on showing the nearby individuals and groups on map and using this user can also set the range in which he want to see the groups.

We also discussed deeply about different aspects related to technology we are going to use. We discussed whether the particular requirement can be implemented with the given technology in the given time or not.

The reasons behind using brainstorming as our base requirement elicitation technique are:

1. It requires less amount of time compared to other techniques.
2. It requires less preparation to discuss and gather requirements.
3. As there can't be any complex requirements or there can't be any ambiguity in our requirements, this technique is best suited.

3.3 Process Model

Scrum model (which is based on Agile methodology) is chosen for this project. It overcomes almost all disadvantages of models such as waterfall model, prototyping model and incremental model, while taking positive points from those models. Scrum model is a very realistic approach to software development in case of this project. It allows rapid development and demonstration of functionalities. It welcomes changing requirements. This is especially desirable in chat systems which are constantly evolving these days, and requirements are often expected to change depending on trend. Daily scrum meetings ensure that any issues are addressed early on in development. Also, short sprints (typically of 1 to 3 weeks) help keep track of how the project is going and whether it will complete within deadline, or whether some adjustments need to be made to reduce such risks. It also promotes

quick delivery of partial systems, and taking feedback from client to plan for next release. This ensures active participation of clients in all stages, so it keeps them satisfied. So client won't be shocked by seeing a completely new system. Instead, few new features will be added in each sprint, then in accordance with client suggestions, issues will be resolved in next sprint, along with addition of some more new features. This process continues. Since this approach is people-oriented, it gives more flexibility to both development team as well as client.

3.4 Requirement Prioritization

Requirement prioritization helps :
- To concentrate on the most important users and their requirements.
- To manage projects more efficiently.
- Allocate resources properly.

All the requirements are important but they need to be prioritized.
For our agile scrum model we have decided to use MoSCoW prioritization technique.
MoSCoW is a prioritization technique for helping to understand and manage priorities.
The letter stand for :
- **M**ust Have
- **S**hould Have
- **C**ould Have
- **W**on't Have This Time

Must Have :
Requirements labeled Must Have are critical. For the current sprint delivery, all the Must Have requirements have to be included for it to be a success. If even any one of the Must Have requirements is not included than the current sprint delivery should be considered as a failure.

Should Have :
Requirements labeled Should Have are important but not vital. They are often not as time-critical or there may be another way to satisfy the requirement.

Could Have :
Requirements labeled Could Have wanted or desirable but less important. It could improve user experience or customer satisfaction for little development cost. These will typically be included if time and resources permit.

Won't Have :

Requirements labeled Won't Have are those which the project team has agreed will not be delivered. This also helps to manage expectations that some requirements will simply not make it into the Deployed Solution, at least not this time around.

Requirement Prioritization For Sprint 1 :

Must Have :
- Login Page
- Sign Up Page
- Fetching Location

Should Have :
- Sign Out Feature
- Profile Page

Could Have :
- Verification of e - mail
- Verification of contact number

Won't Have :
- Chatting Feature

Requirement Prioritization For Sprint 2 :

Must Have :
- Individual Chat

Should Have :
- Searching Individual

Could Have :
- Block Individual

Won't Have :
- Chatting In Group
- Delete Sent Messages

Requirement Prioritization For Sprint 3 :

Must Have :
- Create Group
- Search Group

Should Have :
- Join Group

Could Have :
- Chatting In Group

Won't Have :
- Removing members from group
- Leave Group
- Delete Sent Messages

Requirement Prioritization For Sprint 4 :

Must Have :
- Admin Feature
- Removing Member From Group

Should Have :
- Leave Group

Could Have :

- Adding Description To The Group

Won't Have :
- Sharing Images
- Delete Sent Messages

Requirement Prioritization For Sprint 5 :

Must Have :
- Rating Feature for Groups
- Showing the locations on map

Should Have :
- Dynamic Range Feature

Could Have :
- Calling Directly From App

Won't Have :
- Sharing Group Links
- Delete Sent Messages

4. Features:

4.1 Functional Requirements :

- Login / Signup page
  - ➔ The user first sign ups the application using unique username, email-id, contact number, password.
  - ➔ The user can sign in using the registered username/email-id and password.
  - ➔ The user is also provided with an option to recover password using the provided email-id in-case he forgets it.
- Profile page
  - ➔ The user can add profile photo on the profile page as well as view the profile of other users.
  - ➔ A sign-out option is provided to quit the application.
- Location
  - ➔ The dynamic location of the user is taken at the time of signup as well as login.
  - ➔ Also the location of the user is updated in time-intervals so as to keep him updated with the groups and users in the given range of location.
  - ➔ A specific range for the location is specified for each group.
  - ➔ A user who moves out of his location range is disconnected with the users not in his location range as well as removed from the groups.
- Chats
  - ➔ Two chat options are provided to the user.
    - ➢ Individual Chat
    - ➢ Group Chat
  - ➔ Different tabs are provided for chats.
  - ➔ The user can search for individuals or groups to chat on the basis of his dynamic location.
  - ➔ The user can converse in the groups present in the range of his location.
  - ➔ A user can also create a group by adding a group description.
  - ➔ The user is allowed only to send text messages.
  - ➔ The user can leave any group he wishes no longer to be a part of.
  - ➔ An individual is provided with an option to block any other user.
- Admin
  - ➔ The user who creates the group is assigned the admin of the group.
  - ➔ The admin of the group can make other members admin for the group.
  - ➔ The admin of the group can remove members from the group.

- ➔ If the admin leaves the group, the new admin is automatically assigned as the oldest member based on timestamp.
- Rating
  - ➔ A user can rate a group after being a member for some days.
  - ➔ The user who searches for a group is shown group based on the ratings of the group given by members.
- Map
  - ➔ A user interface is provided where a map is shown with the individuals and groups in the user's range of location.
- Calling
  - ➔ The user is provided with calling feature. The user can call the users he interacts with.
  - ➔ The calling feature directs the app to the phone calling.

4.2 Non - Functional Requirements :

Scalability :
The ability for an increasing number of users to easily share the system. Multiple users can access the app at the same time.

Data Integrity :
Maintenance and assurance of the accuracy and consistency of the data. Stop any unintended changes to the data. It will be ensured by using different error detecting algorithms such as RAID.

Extensibility :
The ability to enhance the system by adding new functionality without any major changes. As software should be long lived, extensibility enables developers to expand or to add software capabilities and facilitates systematic reuse.

Performance :
To ensure minimum response time for each queries. Better implementation of the databases and the search functions involved.

Privacy :

The ability of an individual to express themselves selectively. Limited access refers to a person's ability to participate in society without having other individuals and organizations collect information about them.

5. User Stories

- As a user I need to sign-in so that I can send messages.
- As an administrator I need signup page so that I can get user data.
- As a user I need a recover password so that if I forget the password I can get it back.
- As an administrator I a unique username so that each user can be uniquely identified.
- As an administrator I need a profile page so that other users can see username, name, contact info, email-id and profile photo of people they contact to..
- As a user I need a sign-out option so that I can sign-out from the application.
- As a user I need contact information so that I can contact other users.
- As a User, I want to search a user by name or username so that I can find new users whose name or username is known to me, and also find a user to whom I want to send a message or read previous chats with that user, instead of scrolling through long list of chats shown on Individual Chats tab to find that user, which would be time consuming and cumbersome.
- As a User, I want to chat with another user personally so that I can have a private conversation with another user which I don't want others to read, unlike in group chats where a message is public to all group members.
- As a user I need to make the group so that I can send the message simultaneously to many users with same interest.
- As a Group admin I need to add description to the group so that other user in the locality can easily see the purpose of the group.
- As a member of group I need admin to the group so that he can control the activities of the group.
- As a admin I need to remove the user spreading nuisance so that our group is as productive as possible.
- As a admin I need to able to add the member to the group so that more and more people with same interest are on same platform.
- As a user i want a feature to **search the group** according to my interest. Search group feature will help the users to find the particular group of their interest.

- As a user, After finding the groups of my interest, i should be able to join the particular group. To join the group there must be an option of **join group** whenever i click on the group name.
- To join the group, as a user i need not to send any specific join instruction to anyone, i should be able to join any nearby group without any constraint.
- As a user i need to communicate in the groups. There must a feature to send messages in the groups.
- But as a user i want that this chat feature in any group should be provided after i have joined that group, without joining the group i should not be able to send messages in that group.
- As a user I need functionality to rate the group in the range of 1 to 5.
- As a user I need to check the rating of the group so that I am always the part of good group.
- As a User I need functionality to leave the group so that I can leave annoying group.
- As a User I need to change the range or radius so that I can check out more groups around me.
- As a user I need to fix radius of the group while creating group.
- As a user I should not be able to change the range of group.
- As a user i want an application which will help to communication between people of local area, help me to discover and build groups in my nearby area.
- As a user i want chatting feature for groups as well as for individuals.
- As a user i want that all the nearby groups will be shown to me based on my location and according to my interest i should be able to join the group.
- As a admin i want to make sure if any member of the group goes out of the fixed range of their location he/she should be unable to send messages in the group and automatically leave the group.
- As a user I need to see the location of the group on map so that it would help me to realize the reachability of the place where the group is currently located.
- As a user I want the app to support calling feature so that I can directly make calls whilst in the same application.
- As a user I want the feature to change password so that I can ensure continuous security.
- As a user I want the feature to delete the sent messages so that I can rectify erroneous communications if at all any.
- As a group admin I want the feature of creating link of the group so that anyone can directly join the group.
- As a user I want the feature of sending images so that one can have more interactive and more efficient communicating experience.

6. Effort Estimation

6.1 Using Object Point(OP) analysis

Object points are way to estimate the effort required in the software development techniques. It is not necessarily related to objects in object oriented programming. Here, we show the calculation of object points in our project 'Location based social networking app'. Firstly, we will see the object point weighting

Object Type	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3 GL Modules	10	10	10

Following is the object counts.

Object	Count	Weightage	Total
Simple Screen	4	1	4
Medium Screen	3	2	20
Difficult Screen	15	4	60
Medium Report	1	5	5
Difficult Report	1	8	8
3 GL Modules	0	10	0

Total	99

So, total number of object points without accounting for reusability of code is 99. But 4 of our screens can be reused, that is 4 out of 21 screens is reusable. And that comes out to be 20% of total screens. Now we also need to account for it. Amd formula for that is

$$Adjusted\ NOP\ =\ NOP(1-(\%reuse))$$

Putting the value of %reuse that is 20/100 in above formula

$$\Rightarrow Adjusted\ NOP\ =\ 99(1-0.2)$$
$$=\ 99(0.8)$$
$$=\ 80$$

Developer's Experience	Very low	low	Nominal	High	Very High
Productivity	4	7	13	25	50

Our experience can be counted in low.

$$Effort\ in\ p-m\ =\ Adjusted\ NOP/Prod\quad p-m$$
$$=\ 80\ /\ 7$$
$$=\ 12\quad p-m$$

Effort can also be counted without accounting for adjusting and reusability, and that is given by :

$$Effort\ in\ p-m\ =\ NOP/Prod\quad p-m$$
$$=\ 99\ /\ 7$$
$$=\ 15\quad p-m$$

So,

Effort without reusability = 15 p-m.
Effort with NOP reusability = 12 p-m

1. Function Point(FP) calculation -

It is the measure of functionality particular software provide to user.

Measurement Parameters	count	weightage	Total
Simple Inputs	12	3	36
Complex Inputs	5	6	30
Simple Outputs	11	4	44
Complex Outputs	3	7	21
Simple Queries	3	3	9
Complex Queries	5	6	30
Simple Files	41	7	287
Comp. Files	59	15	885
Complex Ext.	6	15	90

Interfaces			

Total	1432

Now, we need to ask the 14 questions to decide complexity factor. Following are the 14 questions and answer to it. 0 means not important or not applicable and 5 means absolutely essential.

- Backup and recovery   4
- Data Communication   4
- Distributed processing functions   1
- Is performance critical   4
- Existing operating environment   5
- Online data entry   5
- Input transactions built over multiple screens   1
- Master files updated online   3
- Complexity of inputs, outputs, files, inquiries   5
- Complexity of processing   3
- Code design for reuse   5
- Are installation included in design   5
- Multiple installation   5
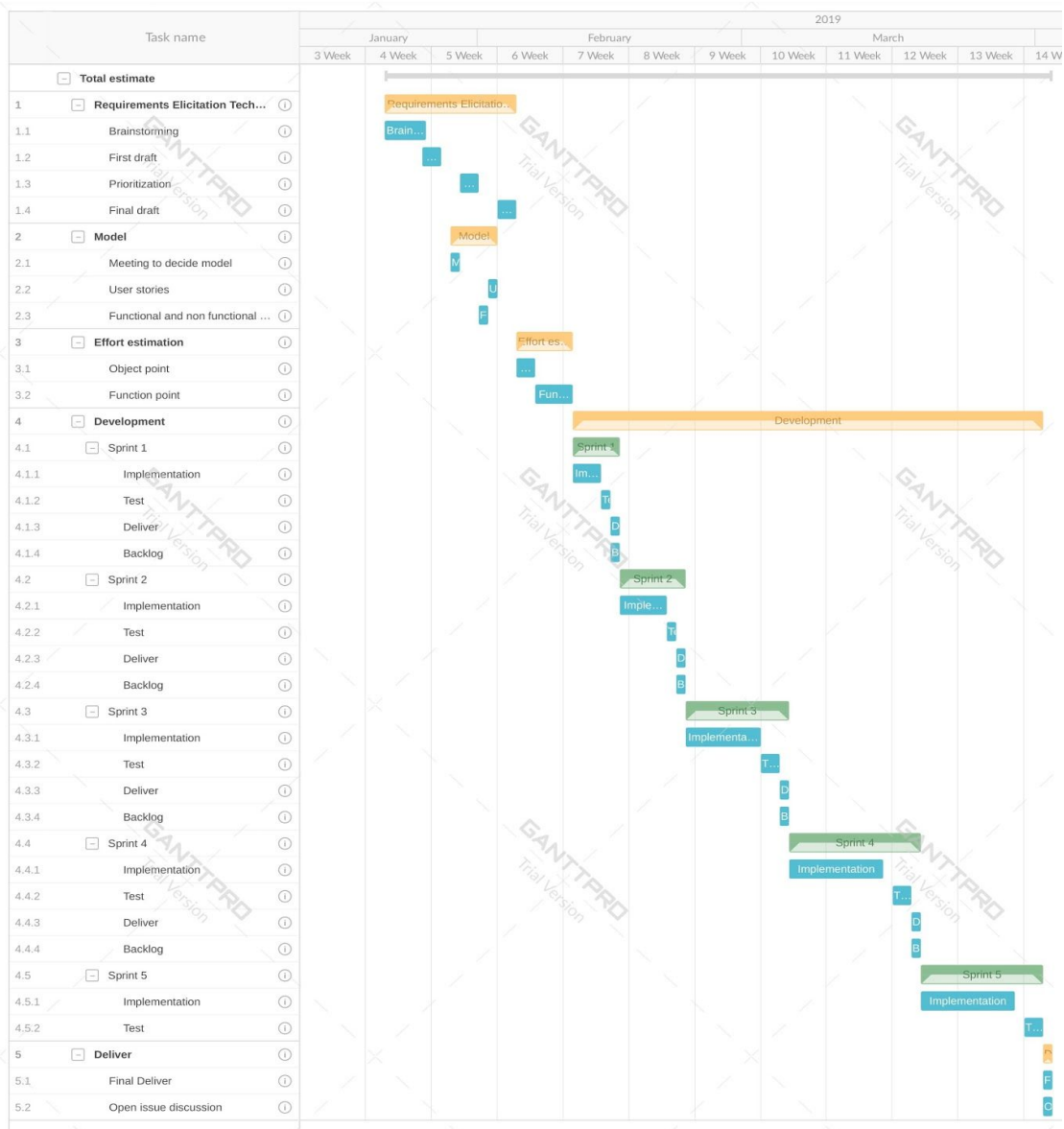- Application designed to facilitate change by the user 4

Total	54

Now, we can easily calculate the adjusted functional points. Formula for the same is

$$AFPC = UFPC * [0.65 + 0.01 * \Sigma Fi]$$
$$= 1432 * [0.65 + 0.01 * 54]$$
$$= 1432 * 0.65 + 0.54$$
$$= 1432 * 1.19$$
$$= 1704.08$$

So, **<u>adjusted function points are 1704.08</u>**

Gantt Chart

7. Minutes of meeting(MoM)

First Meeting

Objective: To learn about skills and interests of each member which would help in deciding what kind of project to take up for the subject.

Date: 16 January, 2019

Duration: 45 minutes

Details:
- Each member of the group turn by turn spoke about their areas of interest and expertise, and skills which they have.
- Web development and android development were among the most common areas of expertise/skills, with almost every member having at least one of these two skills, while 1-2 members also had some experience with Machine Learning.
- Members were asked if they were willing to learn a particular skill (say Android Development), if they did not know about it already, and also whether it was feasible to learn it given short span of time, if the chosen project requires it.

Outcomes: Strengths/skills and areas of interests of all members were known, which would further help in deciding the project.

Members present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Project Decision Meeting

Objective: To decide a set of projects – one to to fill in the form and others as a backup in case clash occurs with some other team.

Date: 18 January, 2019

Duration: 1.5 hours

Details:
- Before the meeting, all members had read the list of project choices given for the subject, and came prepared with their own 3-4 choices.
- Each member spoke about the choices of projects which they came up with, and reasons for why that particular project is good for the team.
- This was followed by voting for each of these projects (weather a member approves of the project or not, given the reasons provided in previous stage), and projects with less than half of members approving it were filtered out.

- The remaining projects were then prioritized after detailed discussions about why a project would be good fit over the other.

Outcomes: A list of 5 projects listed in order of priority.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Lab 1 Meeting

Objective: To understand concepts of requirement elicitation, prioritization, use cases, etc. through example of an insurance company.

Date: 22 January, 2019

Duration: 2.5 hours

Details:
- Brainstorming was done to identify stakeholders and requirements of the system.
- The questions were divided among all members (One question per member). Each member searched online to gather necessary information required to answer the question.
- Once all members were done with the above step, the respective members explained answer of their allotted question to others.
- This was followed by a brief discussion on viewpoints of other members which might have been overlooked or missed by one member who answered that.

Outcomes: We learned about the techniques of Requirement Elicitation, Requirement Prioritization, and concepts of Use Cases and User Stories.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Lab 2 Meeting

Objective: To identify scope of the chosen project (Location based Social Network), users and stakeholders, and various features of the project based on requirements.

Date: 29 January, 2019

Duration: 2.5 hours

Details:
- Brainstorming was done to identify users and stakeholders of the system, and then deciding on possible features based on users/stakeholders' requirements.
- We also looked at various similar existing systems online and what kind of features they provide, and weather we can improve any of those features or implement them in a slightly different and more useful/intuitive way.
- We discussed in detail about identified features, and what kind of user interface to have for each user based on these features. During this phase, ambiguities and inconsistencies were identified in some features and those features were modified accordingly.
- Finally, we discussed about issues with the system, and whether they are within or beyond the scope of our system, and tagged them as Open Issues if they were beyond scope.

Outcomes: A document listing all features in detail, users and stakeholders of the system, UI, and open issues.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Lab 3 Meeting

Objective: To choose a process model for the project and write use cases / user stories based on chosen model.

Date: 5 February, 2019

Duration: 2.5 hours

Details:

- We discussed about various process models taught in class (Waterfall model, Prototyping model, Incremental waterfall model, Spiral model, Scrum, XP, etc.) and whether a particular model will be a good fit for the project.
- After analysing advantages and disadvantages of each model in the context of our project, taking into consideration the complexity and requirements of the project, we chose Agile (Scrum) as the best fit for the project.
- This was followed by writing user stories based on users and features identified in previous lab.

Outcomes: Scrum, a framework based on agile methodology, was chosen as the process model and various User stories were documented.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Play Planning Meeting

Objective: To plan for a play showing process of Agile development in the context of our project.

Date: 7 February, 2019

Duration: 2.5 hours

Details:
- Major roles necessary to depict a typical Agile (Scrum) development environment through a play in the context of our project were decided. Necessary roles were identified as – Product Owner, Scrum Master, Subject Matter Expert, Coder, Developer and Tester.
- We then researched about Scrum development process and discussed about the kind of conversations that can happen between various actors which would help audience understand the essence of Scrum.
- Based on these discussions, a rough script was prepared for the play.

Outcomes: Various roles were identified for the play and a rough script was drafted.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)

6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Play Finalising and Rehearsal Meeting

Objective: To make final draft of the play script, allot identified roles to all members, and to rehearse the play.

Date: 10 February, 2019

Duration: 2 hours

Details:
- Prior to the meeting, all members had read the rough script thoroughly to identify potential ambiguities and inconsistencies in the script.
- The results were then discussed in the meeting and all identified issues were resolved to give final draft of the script.
- Members then gave preferences on what kind of roles they would like to perform in the play, and based on these inputs, we mutually decided allotments for the roles.
- We then rehearsed the play.

Outcomes: Final draft for the script, allotments of roles to team members, and rehearsal of play.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

SRS Meeting

Objective: To make SRS document by merging individual documents made by team members.

Date: 22 February, 2019

Duration: 2 hours

Details:
- Prior to the meeting, work was divided among all members. Hence, everyone had come prepared with a document of the work given to them.
- Initially, we discussed order in which to keep things in the SRS. Then we started merging things in that order from the individual documents.

- Minor edits were made to maintain the flow of the SRS.
- After the initial merging, all members proofread the SRS.
- This was followed by resolving the issues and inconsistencies found during proofreading.

Outcomes: SRS document was prepared.

Members Present:
1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

8. Roles :

Project Manager :
Smit Detroja (201601113)

Meet Sinojia (201601126)   - Requirement Analyst & Tester
Hardik Modi (201601089)   - Backend Developer & Database Designer
Detroja Smit (201601113)   - Backend Developer & Tester
Mudra Desai (201601098)   - Frontend Designer
Harsha Mahaver (201601080)   - Frontend Designer
Gautam Radadiya (201601115)   - Backend Developer & Database Designer
Yash Panwar (201601110)   - Subject Matter Expert

Version 2.0

1. Introduction

1.2 Product Scope

As this is a social networking app based on the location, people with the same interest can join the same groups. This app will be used for the fun as well as for the formal activities. The app also provides a feature to create events and set specific date and time for the same which is visible to users in their locations. For example any educational institute can create their groups for providing a large variety of learning environments and learning spaces. This app will provide a dynamic product ads for people who are having commerce platforms. Health centers and sports centers can make full use out of it to increase their approachability. This app will be helpful for the users for attending conferences conducting in their cities.

2. Overall Description

2.3 User Interface :

We have decided to keep user interface as simple and easy to understand as possible. We have used google's material design to make it attractive. On startup user have signup/login screen on the basis of whether he is signed in or not. If he is already signed in then redirected to home page which has two tabs, one with personal chat and second with group chat. Each tabs has list of persons and groups respectively in the location of user. We will be using drawer for all other routes like profile page, change password, find groups, etc. User will be redirected to that particular page when he clicks on it. Profile page has all information user had entered during signup and can easily edit this information. Find group contains the list of groups that user had not joined but are in the locality of that user. User can view the information of that group by clicking on it. We will also show him the user and group in his locality on the map. The drawer on the top left shows features to change password, create events, view events, settings and logout feature.

2.5 Open Issues :

- The chats are not recommendation based which can be incorporated in future.
- The feature of deleting messages sent by mistake.
- The feature of sending messages other than text messages is also not incorporated.
- There is no feature to join the group via group links or any such medium.
- When a user presses the send button continuously same message is sent till the message is actually sent because of time lag.
- The map for showing location sets only one pin for the user.
- The change password option currently allows on three mails to be sent.

3. Requirement Elicitation

Requirement Prioritization For Sprint 1 :

Must Have :
- Login Page
- Sign Up Page
- Fetching Location

Should Have :
- Sign Out Feature
- Profile Page

Could Have :
- Verification of e - mail
- Verification of contact number

Won't Have :
- Chatting Feature

Requirement Prioritization For Sprint 2 :

Must Have :
- Individual Chat

Should Have :
- Searching Individual

Could Have :
- Block Individual

Won't Have :
- Chatting In Group
- Delete Sent Messages

Requirement Prioritization For Sprint 3 :

Must Have :
- Create Group
- Search Group
- Chatting In Group

Should Have :
- Join Group
- Leave Group

Could Have :
- Removing members from group
- Make admin
- Adding Description To The Group

Won't Have :
- Delete Sent Messages
- Rating of group

Requirement Prioritization For Sprint 4 :

Must Have :
- Rating Feature
- Show location on map
- Calling Feature

Should Have :
- Change Password

Could Have :
- Mailing Feature

Won't Have :
- Sharing Images
- Delete Sent Messages

Requirement Prioritization For Sprint 5 :

Must Have :
- Create Event feature
- Show past and future events

Should Have :
- Like and Dislike Event

Could Have :
- Dynamic Range Feature for groups

Won't Have :
- Sharing Group Links
- Delete Sent Messages

4. Features:

4.1 Functional Requirements :

- Events
    - ➔ The user can create events in his location.
    - ➔ The event has to be provided with a date and time.
    - ➔ Event description had to be provided for the same.
    - ➔ Past event details will be provided to the user if the user had participated in any.
    - ➔ If there are any upcoming events going to happen in the user's nearby location then all those events will be shown to the user as future events.

- Mailing Feature
    - ➔ The feature allows users to mail any other user in the nearby location.
    - ➔ The feature redirects to the gmail application and the sender's id is automatically added.

5. User Stories

- As a user I need to **sign-in** so that I can send messages.
- As an administrator I need **signup page** so that I can get user data.
- As a user I need a recover password so that if I forget the password I can get it back.
- As an administrator I a **unique username** so that each user can be uniquely identified.
- As an administrator I need a **profile page** so that other users can see username, name, contact info, email-id and profile photo of people they contact to..
- As a user I need a **sign-out option** so that I can sign-out from the application.
- As a user I need **contact information** so that I can contact other users.
- As a User, I want to search a user by name so that I can find new users whose name or username is known to me, and also find a user to whom I want to send a message or read previous chats with that user, instead of scrolling
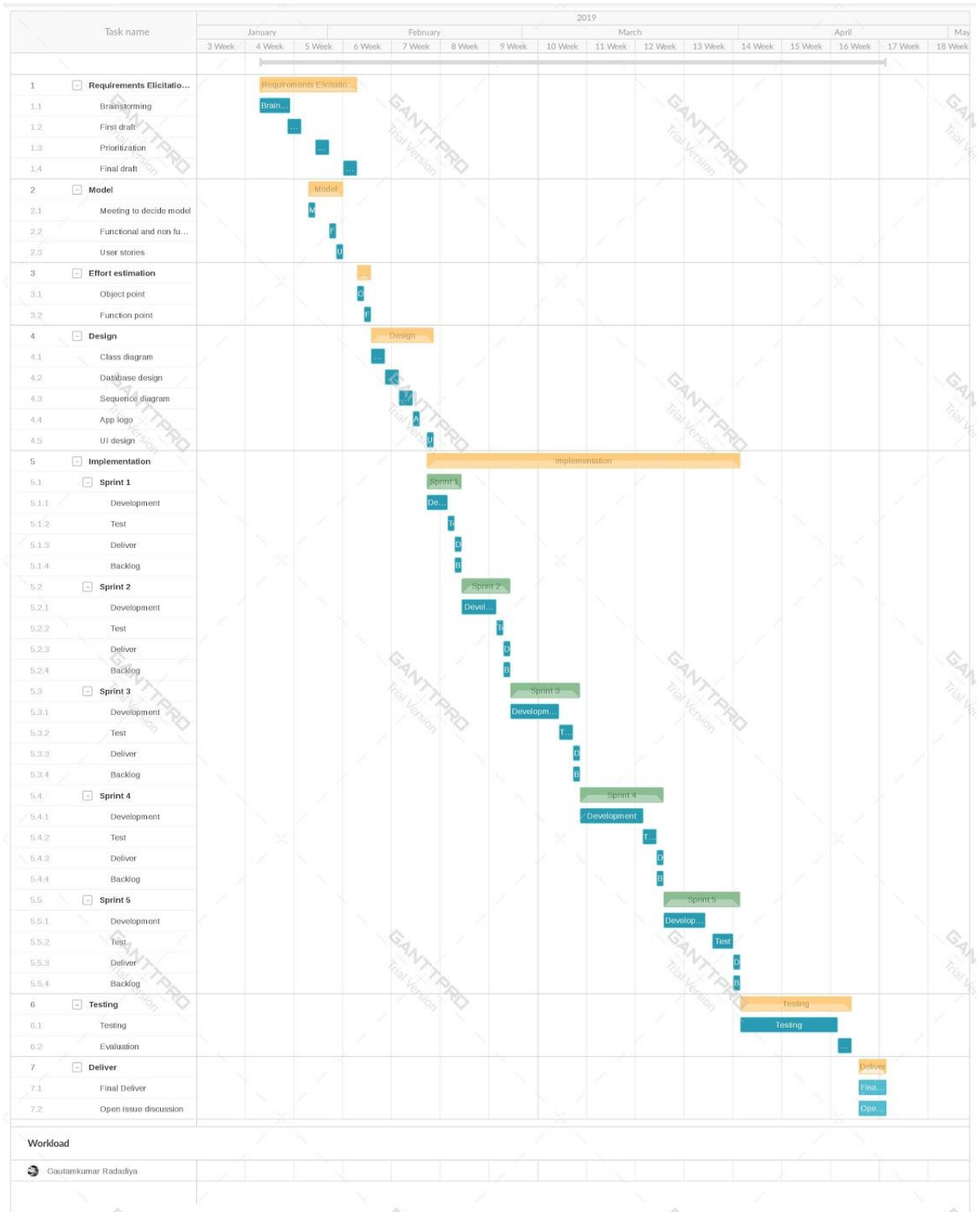
through long list of chats shown on Individual Chats tab to find that user, which would be time consuming and cumbersome.

- As a User, I want to chat with another user personally so that I can have a private conversation with another user which I don't want others to read, unlike in group chats where a message is public to all group members.
- As a user I need to make the group so that I can send the message simultaneously to many users with same interest.
- As a Group admin I need to **add description to the group** so that other user in the locality can easily see the purpose of the group.
- As a member of group I need **admin to the group** so that he can control the activities of the group.
- As a admin I need to **remove the user** spreading nuisance so that our group is as productive as possible.
- As a admin I need to able to **add the member to the group** so that more and more people with same interest are on same platform.
- As a user i want a feature to **search the group** according to my interest. Search group feature will help the users to find the particular group of their interest.
- As a user, After finding the groups of my interest, i should be able to join the particular group. To join the group there must be an option of **join group** whenever i click on the group name.
- To join the group, as a user i need not to send any specific join instruction to anyone, i should be able to join any nearby group without any constraint.
- As a user i need to communicate in the groups. There must a feature to **send messages** in the groups.
- But as a user i want that this **chat feature** in any group should be provided after i have joined that group, without joining the group i should not be able to send messages in that group.
- As a user I need functionality to **rate the group** in the range of 1 to 5.
- As a user I need to check the **rating of the group** so that I am always the part of good group.
- As a User I need functionality to **leave the group** so that I can leave annoying group.
- As a User I need to **change the range or radius** so that I can check out more groups around me.
- As a user I need to fix radius of the group while creating group.
- As a user I should not be able to change the range of group.
- As a user I want an application which will help to communication between people of local area, help me to discover and build groups in my nearby area.
- As a user I want chatting feature for groups as well as for individuals.
- As a user I want that all the nearby groups will be shown to me based on my location and according to my interest i should be able to join the group.
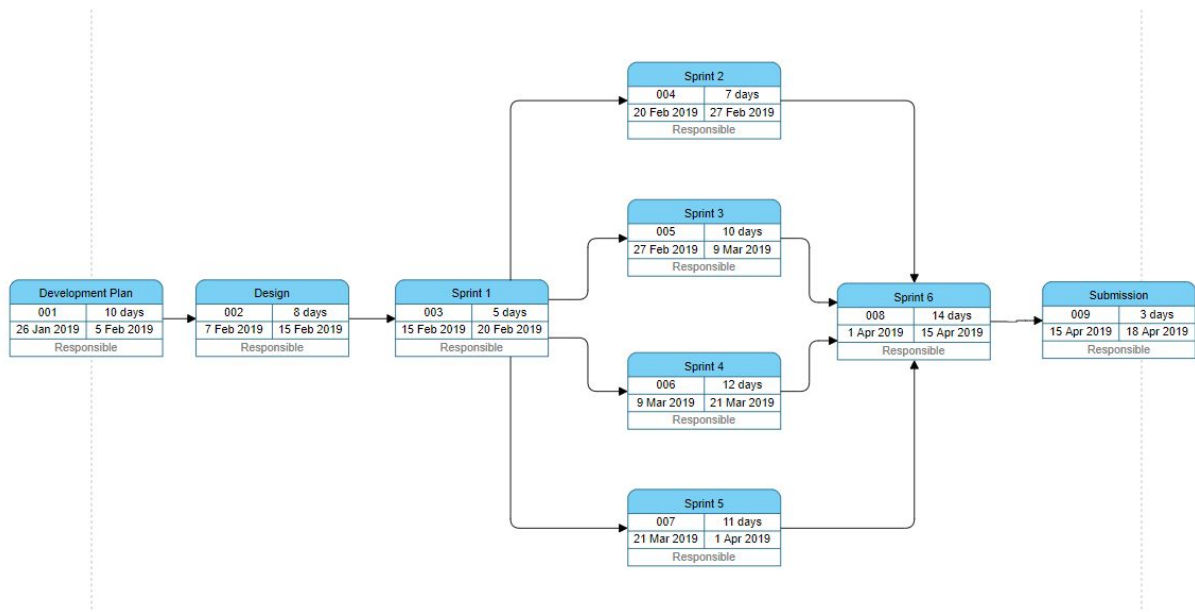
- As a admin I want to make sure if any member of the group goes out of the fixed range of their location he/she should be unable to send messages in the group and automatically leave the group.
- As a user I need to see the **location of the group** on map so that it would help me to realize the reachability of the place where the group is currently located.
- As a user I want the app to support **calling feature** so that I can directly make calls whilst in the same application.
- As a user I want the feature to **change password** so that I can ensure continuous security.
- As a user I want the feature to delete the sent messages so that I can rectify erroneous communications if at all any.
- As a group admin I want the feature of creating link of the group so that anyone can directly join the group.
- As a user I want the feature of sending images so that one can have more interactive and more efficient communicating experience.
- As a user i want the feature to create my own event which is going to happen in my location so that interested people can participate in the event.
- As a user i want a feature which shows me all the past event details if i attended any and also shows the upcoming events conducting in my nearby location.

Gantt Chart

Pert Chart

Development Plan	
001	10 days
26 Jan 2019	5 Feb 2019
Responsible	

Design	
002	8 days
7 Feb 2019	15 Feb 2019
Responsible	

Sprint 1	
003	5 days
15 Feb 2019	20 Feb 2019
Responsible	

Sprint 2	
004	7 days
20 Feb 2019	27 Feb 2019
Responsible	

Sprint 3	
005	10 days
27 Feb 2019	9 Mar 2019
Responsible	

Sprint 4	
006	12 days
9 Mar 2019	21 Mar 2019
Responsible	

Sprint 5	
007	11 days
21 Mar 2019	1 Apr 2019
Responsible	

Sprint 6	
008	14 days
1 Apr 2019	15 Apr 2019
Responsible	

Submission	
009	3 days
15 Apr 2019	18 Apr 2019
Responsible	

7. Minutes of meeting(MoM)

Class Diagram Meeting

Objective: To make class diagram for project.

Date: 7 February, 2019

Duration: 2 hours

Details:
- We first brainstormed about what classes are required to capture all requirements/specifications of the project.
- This was followed by identifying how the classes interact with each other, that is, we identified association, generalization/specialization and dependencies between classes.
- Attributes and methods for each class were then identified.
- The members then did proof-reading of final class diagram, to rectify any mistakes and omissions.

Outcomes: Class diagram for application was prepared.

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

UI Design Meeting

Objective: To design the User Interface of application.

Date: 15 February, 2019

Duration: 2 hours

Details:
- Based on the class diagram made in previous meeting, we discussed what screens would be required in the application, their purpose, and then noted them down.
- For each screen, we then discussed input fields / buttons required on the screen.
- This was followed by designing the layout for each screen. Wireframes were developed for each screen.

Outcomes: Complete User Interface for the application was designed on paper.

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Sprint 1 Planning Meeting

Objective: To plan tasks to be done in Sprint 1 of application development.

Date: 15 February, 2019

Duration: 2 hours

Details:
- Of all features that were specified in SRS, we discussed the functionalities that are essential to implement before implementing anything else, that is, those functionalities on which most of other functionalities depend.
- We did requirements prioritization, and ranked a few functionalities according their importance.
- We then selected first few functionalities which could be implemented and tested in approximately a week.

Outcomes:

Following functionalities were selected to be implemented in first sprint:
- Login / Signup
- Profile page of user
- Fetch location of user

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Sprint 2 Planning & Sprint 1 Evaluation Meeting

Objective: To plan tasks to be done in Sprint 2 of application development, and to evaluate outcomes of Sprint 1.

Date: 20 February, 2019

Duration: 2 hours

Details:
- We evaluated test results of Sprint 1 and discussed if anything needs to be modified in functionalities implemented in the first Sprint, depending on feedback given by all members.
- We then began planning for Sprint 2. After having implemented some essential features in Sprint 1, we tried to answer the question that what are the next few features that could be implemented in about a week, and make the application much more useful than it is now.
- Focus was on implementing the core of the application in this Sprint, which though limited in functionality, can be used by a user.

Outcomes:

Following functionalities were selected to be implemented in second sprint:
- Chat with an individual user
- Search an individual user

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Sprint 3 Planning & Sprint 2 Evaluation Meeting

Objective: To plan tasks to be done in Sprint 3 of application development, and to evaluate outcomes of Sprint 2.

Date: 27 February, 2019

Duration: 2 hours

Details:
- We evaluated the outcomes of Sprint 2, and if anything needs to be modified.
- We then began planning for Sprint 3.
- Each member gave their views on what are the next set of important functionalities, which would make application more usable for users.
- Since a holiday break were to follow, it was known that this Sprint would be slightly longer than others, and so more functionalities can be implemented.
- Based on inputs given by all members, next set of functionalities to be implemented in Sprint 3 were finalized.

Outcomes:

Following functionalities were selected to be implemented in Sprint 3:
- Create group
- Join an existing group
- Search group by name
- Make a user an admin of a group
- Admin can remove a member from group
- Leave group

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Sprint 4 Planning & Sprint 3 Evaluation Meeting

Objective: To plan tasks to be done in Sprint 4 of application development, and to evaluate outcomes of Sprint 3.

Date: 9 March, 2019

Duration: 2 hours

Details:

- We first evaluated outcomes of Sprint 3, and discussed if anything has been left out or any modification is required.
- This was followed by planning of Sprint 4.
- By Sprint 3, core of the application was developed. Focus was now on additional features which would make users' experience even better.
- As usual, members first gave their inputs on what are the next set of features that can be implemented in the next sprint.
- Based on these inputs, features to be implemented in Sprint 4 were finalized.

Outcomes:

Following functionalities were selected implemented in Sprint 4:
- Ability to give groups a rating
- Map on which location of different users is shown
- User can change range within which to perform searches

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Sprint 5 Planning & Sprint 4 Evaluation Meeting

Objective: To plan tasks to be done in Sprint 5 of application development, and to evaluate outcomes of Sprint 4.

Date: 21 March, 2019

Duration: 2 hours

Details:
- We first evaluated outcomes of Sprint 4 and discussed if anything needs to modified, or if anything was missing in the application, that was planned.
- We then began planning for the final Sprint, that is, Sprint 5.
- We went through the list of features specified in SRS, but yet to be implemented.
- After a discussion among all members, features were ranked in order of priority, so that due to time constraints, if some features need to be left out, then those features are the ones which make least impact on usability of the app.

Outcomes:

Following features were finalized to be implemented in Sprint 5:
- Create Event
- Add an event to the list of events to be attended
- Block a user
- Change Password

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Work Division for Final Submission Meeting

Objective: To divide work among all group members to prepare all required documents for final submission of the project.

Date: 11 April, 2019

Duration: 1 hour

Details:
- By this time, application was working and was unit tested. A few last things remained like GUI testing, performance testing, acceptance testing, making video of the application working, and some documents which were to be submitted.
- We first made a list of things which are yet to be done.
- Then, depending on members' skills and interests, work was divided among everyone.
- It was also planned that all members would next meet on 16 April, that is, after 5 days, to combine work and discuss open issues, if any.

Outcomes:

Work to be done was divided among all members.

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

Work Merging & Open Issues Discussion Meeting

Objective: To merge work divided among team members in previous meeting, in order to prepare list of final documents to be submitted, and also to discuss any open issues still remaining.

Date: 16 April, 2019

Duration: 4 hours

Details:

- Individual documents prepared by individual members were checked and verified by all members, and necessary modifications were made.
- These documents were then merged in proper order.
- We looked at the list of documents prepared and checked them off in list of documents to be submitted, to make sure that everything is ready.
- This was followed by a discussion of open issues that could not be resolved in the duration of this project due to different reasons, and made a list of them.

Outcomes:

All documents were merged to give a final documentation of the project, and open issues were listed.

Members Present:

1. Meet Sinojia (201601126)
2. Hardik Modi (201601089)
3. Detroja Smit (201601113)
4. Mudra Desai (201601098)
5. Harsha Mahaver (201601080)
6. Gautam Radadiya (201601115)
7. Yash Panwar (201601110)

8. Roles :

Project Manager :
Smit Detroja (201601113)

Meet Sinojia (201601126)	- Requirement Analyst & Tester
Hardik Modi (201601089)	- Frontend & Backend Developer & Database Designer
Detroja Smit (201601113)	- Frontend & Backend Developer & Tester
Mudra Desai (201601098)	- Tester & Documentation
Harsha Mahaver (201601080)	- Documentation
Gautam Radadiya (201601115)	- Frontend & Backend Developer & Database Designer
Yash Panwar (201601110)	- Subject Matter Expert, Creating Video