

Chapter 1

INTRODUCTION

This chapter is a part of our software requirement specification for the project “EventMeetApp”. In this chapter we will focus on the intended audience for this project.

1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of EventMeetApp. It contains the functional, non-functional and the supporting requirements and establishes a requirements’ baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 INTENDED AUDIENCE

This SRS report is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.

- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
- The designers will use this SRS as a basis for creating the system’s design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s needs.
- The developers will use this SRS as a basis for developing the system’s functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer’s documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software

fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 CONCLUSION

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

Chapter 2

INCEPTION

In this chapter, the Inception part of the SRS will be discussed briefly.

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. To establish the groundwork we have worked with the following factors related to the inception phases:

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the First Questions

2.1.1 Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. To identify the stakeholders we consulted with Event organizer group and asked them following questions:

- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed).

Concluding thoughts on Stakeholders, We identified following stakeholders:

1. System Operator: System Operator will directly interact with this software. They can be considered as Super Admin.

2. Organizers: Organizers will arrange an event and maintain all the tasks related to the event. They are the admin for a particular event.
3. Participants: Participants join and follow events arranged by the organizers. They are the largest user group of the system.
4. Resource persons: Resource persons are invited to an event. They take classes, share their experiences, provide important resource materials.
5. Developers: We selected developers as stakeholders because they develop this system and work for further development. If occurs any system interruption, they will find the problem and try to solve it.

2.1.2 Recognizing multiple viewpoints

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below.

1. Organizer/Administrator's viewpoints:
 - a. User friendly interface
 - b. Web based application
 - c. Restrict access to functionality of the system based upon user roles.
 - d. The application can be accessed from any computer that has Internet access.
 - e. Maintain a database for all items of the project.
2. Participant's viewpoints:
 - a. Less complex authentication system.
 - b. Android application is preferable.
 - c. Easy access to all features.
 - d. Always get updated notification.
 - e. Good communication system.
 - f. Allow user to search for items.
3. Resource Person's viewpoints:
 - a. A very user friendly interface.
 - b. Good notification system

2.1.3 Working towards collaboration

Every stakeholder has their own requirements. We followed following steps to merge these requirements:

- Identify the common and conflicting requirements
- Categorize the requirements
- Take priority points for each requirements from stakeholders and on the basis of this
- voting prioritize the requirements
- Make final decision about the requirements.

Common requirements:

- Easy authentication system
- The application can be accessed from any computer that has Internet access
- Allow any user to search for items
- Maintain a database of all items in the application.

Conflicting Requirements:

We found some requirements conflicting each other .We had to trade-off between the requirements.

- Administrator wants web based interface and participants demand for android application
- Allow any user to create an event and allow valid user to create an event.

Final Requirements:

We finalized following requirements for the system by categorizing and prioritizing the requirements.

- Web based interface for administrator and android application for others.
- Error free system (5% error may be considerable)
- Accessible via the Internet.
- Use google oAuth for authentication.
- Super Admin will create an event and assign admin for that event.
- Admin will maintain the whole event tasks.
- Restrict access to functionality of the system based upon user roles
- There will be a discussion forum and conversation system for better communication.
- Participants can see other participants' details for a particular event.
- Maintain a database for the entire application.

2.1.4 Asking the First Questions

We set our first set of context-free questions focuses on the customer and other stakeholders, overall project goals and benefits. The questions are mentioned above. These questions helped us to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped us to gain a better understanding of problem and allows the customer to voice his or her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.2 Conclusion

In this inception phase, a basic understanding of the problem was developed and a preliminary nature of the solution was obtained. The requirements which are identified in this phase, will be used later for further steps of requirement engineering.

Chapter 3

ELICITATION

The purpose of this chapter is to specify the elicitation part.

3.1 Introduction

Elicitation is a task that helps the customer to define what is required. To complete the elicitation step we face many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, we have worked with the Eliciting requirements activity in an organized and systematic manner.

3.2 Eliciting Requirements

Unlike inception where Q&A (Question and Answer) approach is used, elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements .To elicit requirements we completed following four works.

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation work products

3.3 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed following steps to do it.

- The meetings were conducted with the Event organizer who was questioned about the requirements and expectations for the system.
- We have made a questionnaire and gathered requirements from the people who have participated in seminar and workshop.
- At last we selected our final requirements from the meetings.

3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

3.4.1 Normal Requirements

The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction.

These are the normal requirements for the project.

1. Accessible via the Internet.
2. View event list details.
3. View user list and details.
4. Allow users to login and logout
5. Restrict access to functionality of the system based upon user roles
6. User can edit their profile.
7. Admin can ban participants.
8. Super Admin can disable event.
9. Admin can edit event information.
10. System will have good notification system.
11. Good conversation system
12. A particular event will have its own discussion forum.

3.4.2 Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

1. Event location will be shown in Map
2. Error free system.
3. Good maintainable database
4. Admin will get a web based interface.
5. Participants and Resource persons will use android application.
6. System will send notification automatically if there is change in any event.
7. Participants can mention other in forum to ask question.
8. Participant can download files.
9. Only admin can upload files.

3.4.3 Exciting requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying. Some of these requirements are:

1. User will be able to sign in with their google account.
2. Participant can use one tap contact saving feature.
3. There will be a “Things To Do” section. In this section, participants and resource person can have suggestions of important places near event spot.
4. Participant will have “check in” option.

3.4.4 Usage Scenario

Event Meet App is an android application which will give a platform for an Event (Workshop, Seminar etc) connecting both the Organizers, Participants and Resource Persons. This application will perform the following activities.

1. Event Creation

After manual verification of a requested Event, Super Admin will create an event and she/he will make a user Admin of the event. Then the admin will be provided an Event ID. With this Event ID users can enter into this event. To complete event creation, admin has to provide following information:

- i) Event Code
- ii) Event Name
- iii) Description
- iv) Location: Specific location of the event on Google Map
- v) Schedule

All of the administrative works will be done through web service. Only Admins can get access to the web service.

2. User Authentication

Our system can have four kind of users: Super Admin(Authority), Admin (Organizer), Participant and Resource Person. Users will be authenticated through OAuth, that is, users can sign in to this application using their google accounts. When a user sign in to our system for the first time, initially user profile will be created using the information of their google

accounts. Then he/she has to provide following information for completing profile creation:

1. User Name
2. Address
3. Phone No
4. Email
5. Occupation
6. Date of Birth
7. Nationality

After completing profile creation, next time whenever user sign in to the system he/she need not to create profile again. But users can update their profile fields later. Users can sign out from the system.

To join a particular event, users have to provide a particular Event ID and Event Code.

Admin can invite Resource Persons to join the event. He/she can disable any participant for a particular event. Super admin can disable any user from the system.

3. User Information Management

System will store all the information provided by users. Admin can disable a participant for a particular event, but cannot permanently delete his information.

Users can view other users' information from a particular event's Participant list and Resource person list respectively. There will be an option for direct contact saving so that within a particular event a user can save others' contact info to his own phonebook if necessary. Whenever users attend in an event, they can update their status clicking "checked in" and it will be shown in the event participant list. Thus users can see whoever have attended the event.

4. Event Information Management

Every event will have a lifetime. Within that period of time, information of an event will remain visible. If lifetime of an event is over, it will be disabled. However, information will remain in the Database.

Any user can view event information but only Admin has the authority to change any event information. A user can see where the event is going to be held, from the application with the help of Google Map API. Users can see all of their previous and upcoming events in "My Events" section.

5. File Management

Admin can upload different files such as slides, pdfs and photos to the event. Photos will be shown in “Photo Gallery” and other files will be shown in “Resources” section of the application. Every file will have file ID, file name, file type and upload time. Users can download the files of a particular event anytime within the lifetime of the event.

6. Notification

Users will get notices and reminder of the event through notification. Every notification will have Notification Id, Notification type and content. Admin can set notification to send the users of the event. Again system can generate notification if there are any changes in the event, for example, changes in schedule, location of the event. If a user is logged in to a particular event, he/she will be sent push notifications by the system as reminder to attend the event or whenever he/she is mentioned in the forum.

7. Communication

A user can communicate with others in two ways:

i) Forum: There will be a forum for a particular event. Here, a participant or a resource person can post and reply to other's post. Users can ask questions or give feedback through posts. Again users can mention other users of the event.

Forum will be stored throughout the every lifetime. Each forum will have Forum ID and posts. There will be Post ID, subject, post content, reply ID and reply content corresponding to a post.

ii) Conversation: A user can make a personal conversation with another user. The sender id, receiver id and messages will be stored temporarily.

8. Things To Do

Users can get useful suggestions about the nearby places and also view location in google map in Things To Do section. For example, if there are restaurants, resorts or tourist spot nearby the event location, users will come to know about them from this section as well as get direction from the google map.

Chapter 4

SCENARIO-BASED MODEL

4.1 INTRODUCTION

In this model the system is described from the user's point of view. As this is the first model, it serves as input for creation of other modeling elements. Although the success of a computer based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end-users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swimlane diagrams.

4.2 DEFINITION OF USE CASE

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.3 USE CASE SCENARIO

Table : Use Case Scenario

Level 0	Level 1	Level 2	Actor
Event Meet App	1.1 Authentication	1.1.1 Sign In with Google Account	S. Admin, Admin, Participant, Resource Person
		1.1.2 Profile Creation	S. Admin, Admin, Participant, Resource Person, System
		1.1.3 Profile Modification	S. Admin, Admin, Participant, Resource Person
		1.1.4 Sign Out	S. Admin, Admin, Participant, Resource Person
	1.2 Event Management	1.2.1 Event Creation	Super Admin
		1.2.2 Viewing Events	Super Admin, Admin
		1.2.3 Updating Event Info	Admin

		1.2.4 Joining New Event	Participant, Resource Person
		1.2.5 Disable Event	Super Admin, Admin
1.3 User Information Management	1.3.1 Viewing Participant Info	S. Admin, Admin, Participant, RP	
	1.3.2 Viewing RP Info	S. Admin, Admin, Participant, RP	
	1.3.3 Inviting RP	Admin	
	1.3.4 Contact Saving	Participant, RP	
	1.3.5 Check In	Participant, RP	
1.4 File Management	1.3.6 Disable User	S. Admin, Admin	
	1.4.1 Uploading File	Admin	
	1.4.2 Downloading File	Admin, Participant, RP	
	1.4.3 Remove File	Admin	
	1.4.4 View Photo Gallery	Admin, Participant, RP	
1.5 Notification	1.4.5 View Resources	Admin, Participant, RP	
	1.5.1 Send Notification	Admin, System	
	1.5.2 Receive Notification	Participant, RP	
1.6 Forum	1.6.1 Post	Participant, RP	
	1.6.2 Reply	Participant, RP	

		1.6.3 Mention	Participant, RP
1.7 Conversation		1.7.1 Send Message	Participant, RP, System
		1.7.2 Receive Message	Participant, RP
1.8 Things To Do		1.8.1 View Suggestion	Participant, RP
		1.8.2 View Location	Participant, RP, Google Map API

4.4 USE CASE DESCRIPTIONS

There are eight subsystems of EventMeetApp : Authentication, Event Management, User Information Management, File Management, Notification, Forum, Conversation and Things To Do.

4.4.1 Authentication

Authentication is divided into four parts: Sign In with Google Account, Profile Creation, Profile Modification and Sign Out.

4.4.1.1 Sign In With Google Account

Use Case: Sign In With Google Account

Primary Actor: Super Admin, Admin, Participant, Resource Person

Secondary Actor: Google API

Goal in context: to sign in to the system through Google Account

Scenario:

1. To access the system, user need to sign in.
2. Users will use their google account to sign in

Priority: Essential, must be implemented.

When Available: First increment

4.4.1.2 Profile Creation

Use Case: Profile Creation

Primary Actor: Super Admin, Admin, Participant, Resource Person

Goal in context: to create profile when a user sign in to the system for the first time.

Scenario:

1. When a user signs in for the first time, a new profile of his own is created.
2. No profile creation is needed if already there is a profile with the same email.

Priority: Essential, must be implemented.

When Available: First increment

4.4.1.3 Profile Modification

Use Case: Profile Modification

Primary Actor: Super Admin, Admin, Participant, Resource Person

Goal in context: to modify a previously created profile in the system.

Scenario:

1. User updates information of his own

Priority: Essential, must be implemented.

When Available: First increment

4.4.1.4 Sign Out

Use Case: Sign Out

Primary Actor: Super Admin, Admin, Participant, Resource Person

Goal in context: to sign out from the system.

Scenario:

1. Access system
2. Click logout

Priority: Essential, must be implemented.

When Available: First increment

4.4.2 Event Management

Event Management is divided into five parts.

4.4.2.1 Event Creation

Use Case: Event Creation

Primary Actor: Super Admin

Goal in context: to create a new event by the authority after manual verification.

Scenario:

1. Super admin creates a new event.
2. At the time of event creation, super admin assigns an Event Admin. Event Admin is basically one from the event organizers.
3. Admin fills up the additional event information.

Priority: Essential, must be implemented.

When Available: First increment

4.4.2.2 Viewing Event(s)

Use Case: Viewing Event

Primary Actor: Super Admin, Admin, Participant, Resource Person

Secondary Actor: Google Map Api

Goal in context: to view event details

Scenario:

1. Users can get access to an event with Event Id and Event Code.
2. Entering event, users can view every event details.

Priority: Essential, must be implemented.

When Available: First increment

4.4.2.3 Updating Event Information

Use Case: Updating Event Information

Primary Actor: Admin

Goal in context: to update Event info if necessary and store them.

Scenario:

1. Event Admin can update event information throughout the event lifetime.
2. Users can see the updated information.

Priority: Essential, must be implemented.

When Available: First increment

4.4.2.4 Joining New Event

Use Case: Joining New Event

Primary Actor: Participant, Resource Person

Goal in context: to enter into a new event using Event Id.

Scenario:

1.If a user wants to join a new event, he/she must provide the particular event's Event Id and Event Code.

2. If a user has multiple events, they will be shown in My Events section.

Priority: Essential, must be implemented.

When Available: First increment

4.4.2.5 Disable Event

Use Case: Disable Event

Primary Actor: Super Admin, Admin

Goal in context: to disable an Event whenever lifetime is over or whenever the authority want .

Scenario:

1.Super Admin or Admin can disable an Event.

2. All the event activities stop after disabling an event, but information of that event remains in database.

Priority: Essential, must be implemented.

When Available: First increment

4.4.3 User Information Management

User Information Management is divided into six parts.

4.4.3.1 Viewing Participant Information

Use Case: Viewing Participant Information

Primary Actor: Super Admin, Admin, Participant, Resource Person

Goal in context: to store and view participants' information on his/her consent.

Scenario:

1. Information of the participants of an event is stored.

2. Users of an event can view other participant's information of the same event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.3.2 Viewing Resource Person Information

Use Case: Viewing Resource Person Information

Primary Actor: Super Admin, Admin, Participant, Resource Person

Goal in context: to show resource person's information.

Scenario:

1. Information of resource persons is stored.
2. Users of an event can see the resource persons' information of that event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.3.3 Inviting RP

Use Case: Inviting RP

Primary Actor:, Admin

Goal in context: to invite a user as resource person by admin of an Event.

Scenario:

1. Admin can invite resource persons for a particular event by clicking “invite”.
2. Resource Persons can approve invitation

Priority: Essential, must be implemented.

When Available: First increment

4.4.3.4 Contact Saving

Use Case: Contact Saving

Primary Actor: Admin, Participant, Resource Person

Goal in context: to save the contact info of a user of a particular event by a single “Save contact” button.

Scenario:

1. Users of an event can view other's contact information in user list.
2. They can save contact information to their phonebook by clicking “Save contact” button.

Priority: Essential, must be implemented.

When Available: First increment

4.4.3.5 Check In

Use Case: Check In

Primary Actor: Participant.

Goal in context: to change the status of a participant clicking “Check In” in the participant list.

Scenario:

1. After arriving the event, a user can give check in by clicking the “Check in” button beside his/her profile.

2. Users can see in the users list if other users give “check in”.

Priority: Essential, must be implemented.

When Available: First increment

4.4.3.6 Disable User

Use Case: Disable User

Primary Actor: Super Admin, Admin

Goal in context: to disable a User if he/she is found not authenticated.

Scenario:

1. Admin or super admin can disable a user for a particular event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.4 File Management

File management is divided into five parts.

4.4.4.1 Uploading File

Use Case: Uploading File

Primary Actor: Admin.

Goal in context: to upload photos or resource materials in the system.

Scenario:

1. Only Event admin can upload files for a particular event.
2. Files can contains photo or resource materials(pdf, ppt etc).

Priority: Essential, must be implemented.

When Available: First increment

4.4.4.2 Downloading File

Use Case: Downloading File

Primary Actor: Admin, Participant, Resource Person

Goal in context: to download files of a particular event.

Scenario:

1. Users of an event can see and download files of an event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.4.3 Remove File

Use Case: Remove File

Primary Actor: Admin

Goal in context: to remove a unnecessary file from the event's file section.

Scenario:

1. Admin can remove any files from the event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.4.4 View Photo Gallery

Use Case: View Photo Gallery

Primary Actor: Admin, Participant, Resource Person

Goal in context: to view all photos of an event together in this section.

Scenario:

1. Users of an event can view all the photos of the event in “Photo Gallery” section.

Priority: Essential, must be implemented.

When Available: First increment

4.4.4.5 View Resource Files

Use Case: View Resource Files

Primary Actor: Admin, Participant, Resource Person.

Goal in context: to view the important materials of the event together.

Scenario:

1. Users of an event can view all the resources files of an event in “Resource File” section.

Priority: Essential, must be implemented.

When Available: First increment

4.4.5 Notification

Notification is divided into two parts.

4.4.5.1 Send Notification

Use Case: Send Notification

Primary Actor: Admin, System

Goal in context: to set the notification by admin or generate by system automatically and send them to all users of a particular event.

Scenario:

1. System can send notification to users whenever any event information is changed.
2. Admin can set important notices as notification and these are send to all users of an event by system.

Priority: Essential, must be implemented.

When Available: First increment

4.4.5.2 Receive Notification

Use Case: Receive Notification

Primary Actor: Participant, Resource Person

Goal in context: to receive the notification of a particular event by the users of that event.

Scenario:

1. Users can view their notification through “Notification” section.

Priority: Essential, must be implemented.

When Available: First increment

4.4.6 Forum

Forum is divided into three parts.

4.4.6.1 Post

Use Case: Post

Primary Actor: Participant, Resource Person

Goal in context: to post to the public forum of a particular event for asking question and giving feedback.

Scenario:

1. Participant or resource person can give posts to the forum.
2. They can question or give feedback through posts.
3. Forum is public for all users within an event.

Priority: Essential, must be implemented.

When Available: First increment

4.4.6.2 Reply

Use Case: Reply

Primary Actor: Participant, Resource Person

Goal in context: to response to the post of a user.

Scenario:

1. Participants or resource persons can reply to any post.
2. Multiple users can reply to a post.

Priority: Essential, must be implemented.

When Available: First increment

4.4.6.3 Mention

Use Case: Mention

Primary Actor: Participant, Resource Person

Goal in context: to mention a participant or a resource person on a forum asking something.

Scenario:

1. Participants or Resource person can mention other included into the event in the forum.
2. If anyone is mentioned, he/she is notified by system through Notification.

Priority: Essential, must be implemented.

When Available: First increment

4.4.7 Conversation

Conversation is divided into two parts.

4.4.7.1 Send Message

Use Case: Send Message

Primary Actor: Participant, Resource Person

Goal in context: to send any message through plain text to a user by one to one communication.

Scenario:

1. Any user can send text message to other user.
2. This conversation is one to one conversation.

Priority: Essential, must be implemented.

When Available: First increment

4.4.7.2 Receive Message

Use Case: Receive Message

Primary Actor: Participant, Resource Person

Goal in context: to receive messages from another user.

Scenario:

1. Receiver can view their messages in the Message section.
2. Messages are temporarily stored.

Priority: Essential, must be implemented.

When Available: First increment

4.4.8 Things To Do

Things To Do is divided into two parts.

4.4.8.1 View Suggestion

Use Case: View Suggestion

Primary Actor: Participant, Resource Person

Goal in context: to view suggestions about nearby important places.

Scenario:

1. In the Things To Do section, any user can view suggestions of nearby places from the event spot set by the admin.

Priority: Essential, must be implemented.

When Available: First increment

4.4.8.2 View Location

Use Case: View Location

Primary Actor: Participant, Resource Person

Secondary Actor: Google Map

Goal in context: to view location of the suggested places on Google map and get directions.

Scenario:

1. Users can see the location of the suggested places in the Google Map by clicking “View location”.
2. They can also get directions of that location.

Priority: Essential, must be implemented.

When Available: First increment

4.5 USE CASE DIAGRAMS

Use Case diagrams give the non-technical view of overall system.

4.5.1 Level-0: Use Case Diagram - Event Meet App

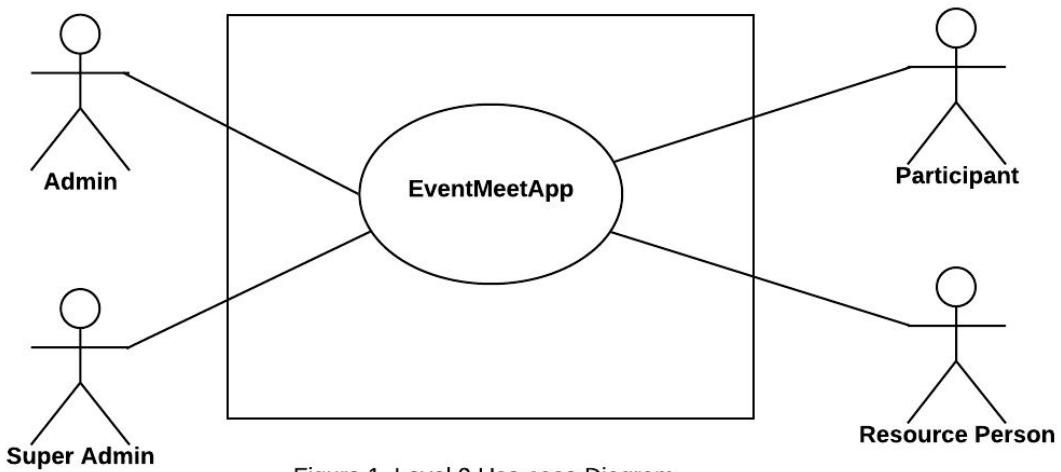


Figure 1. Level 0 Use case Diagram -
EventMeetApp

4.5.2 Level-1: Use Case Diagram- Subsystems

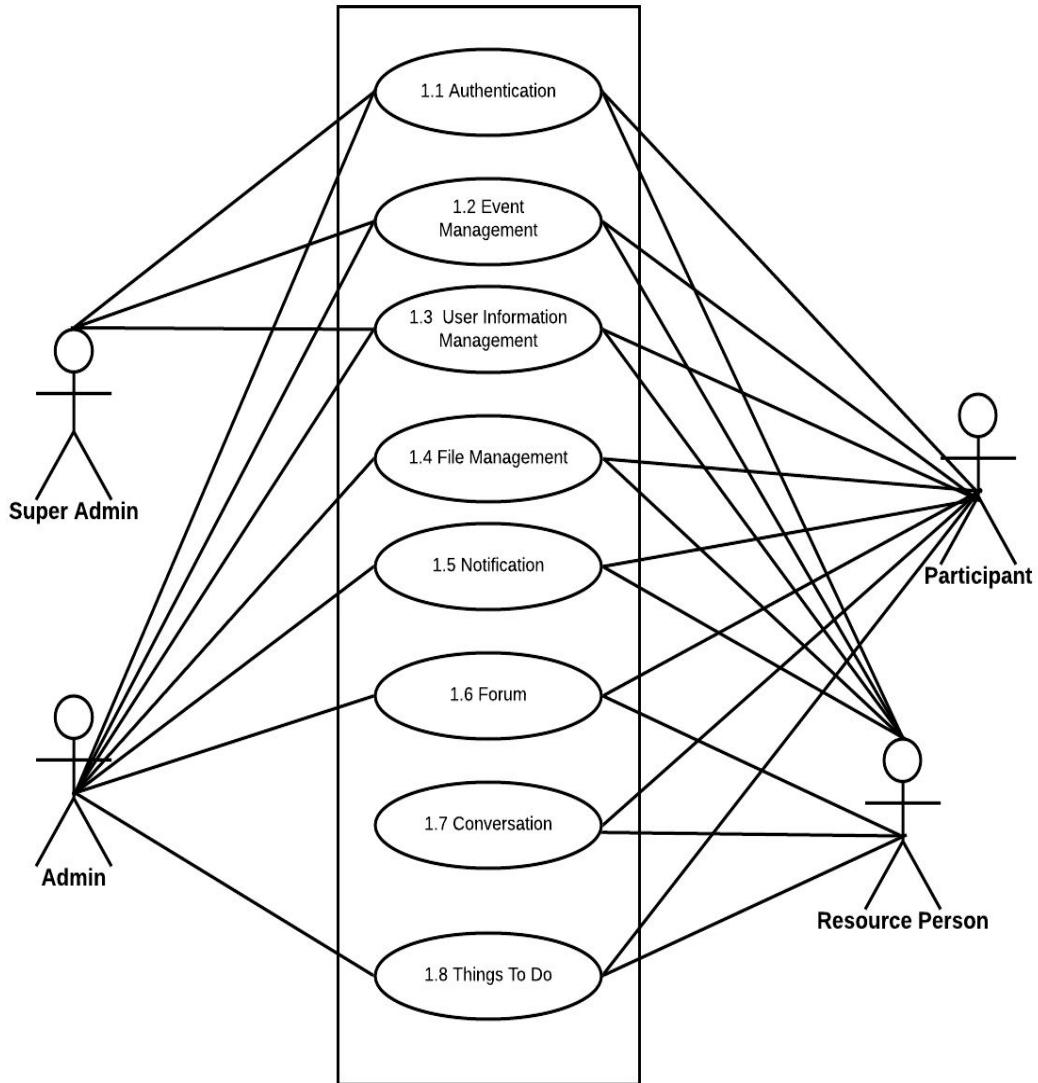


Figure 2. Level-1 Use Case Diagram - Subsystems

4.5.3 Level-2.1 Use Case Diagram - Authentication

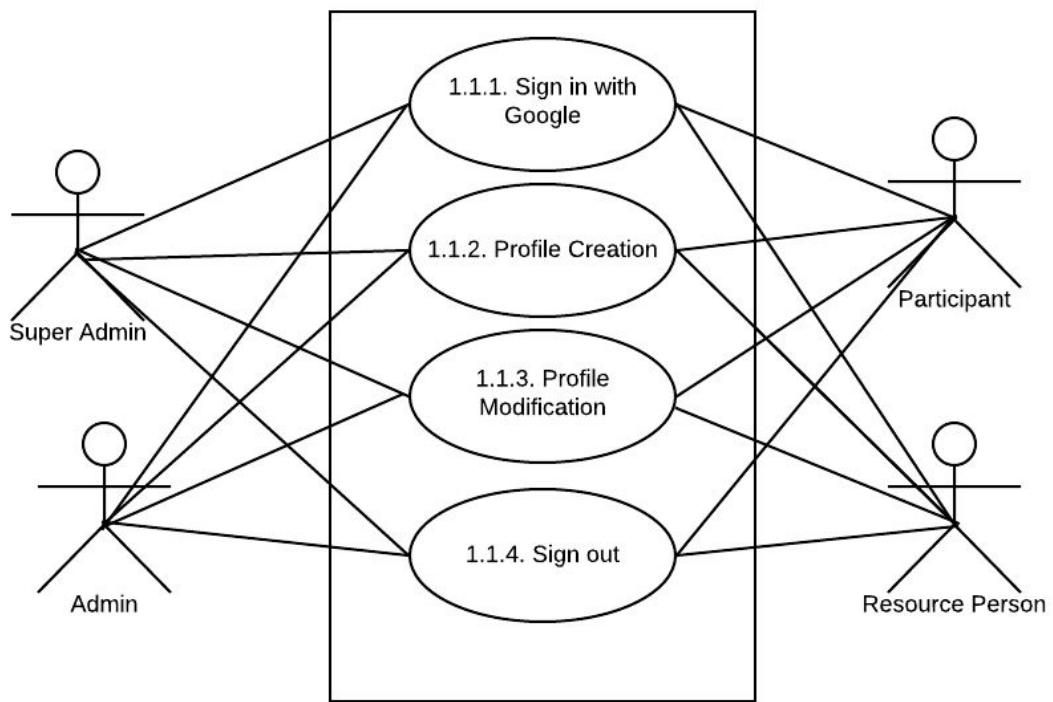


Figure 3. Level-2.1 Use Case Diagram - Authentication

4.5.4 Level-2.2 Use Case Diagram - Event Management

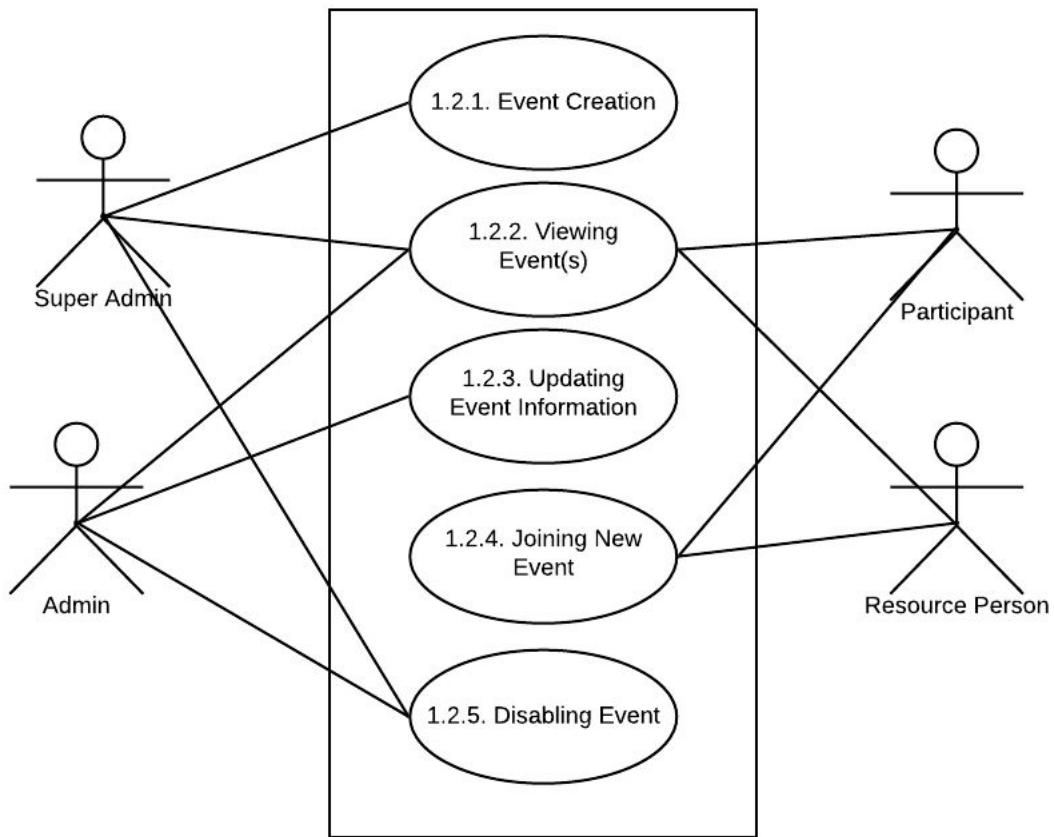


Figure 4. Level-2.2 Use Case Diagram - Event Management

4.5.5 Level-2.3 Use Case Diagram - User Information Management

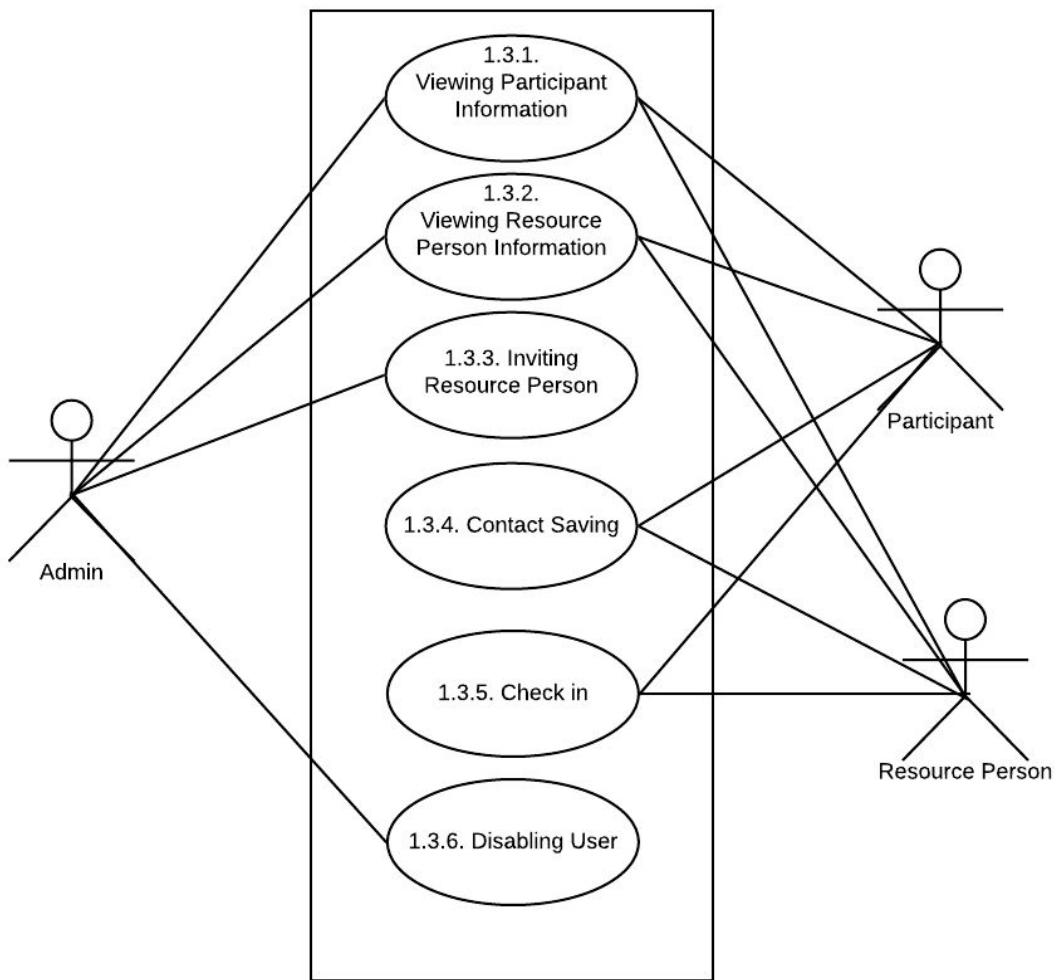


Figure 5. Level-2.3 Use Case Diagram - User Information Management

4.5.6 Level-2.4 Use Case Diagram - File Management

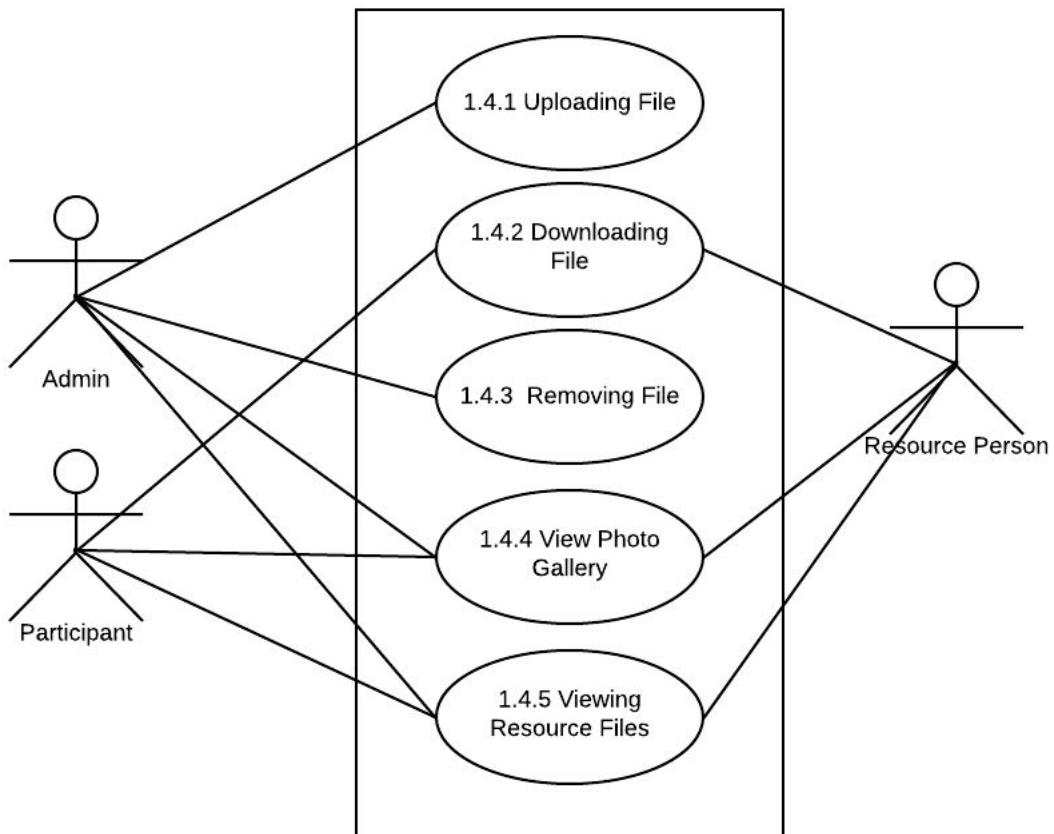


Figure 6. Level-2.4 Use Case Diagram - File Management

4.5.7 Level-2.5 Use Case Diagram - Notification

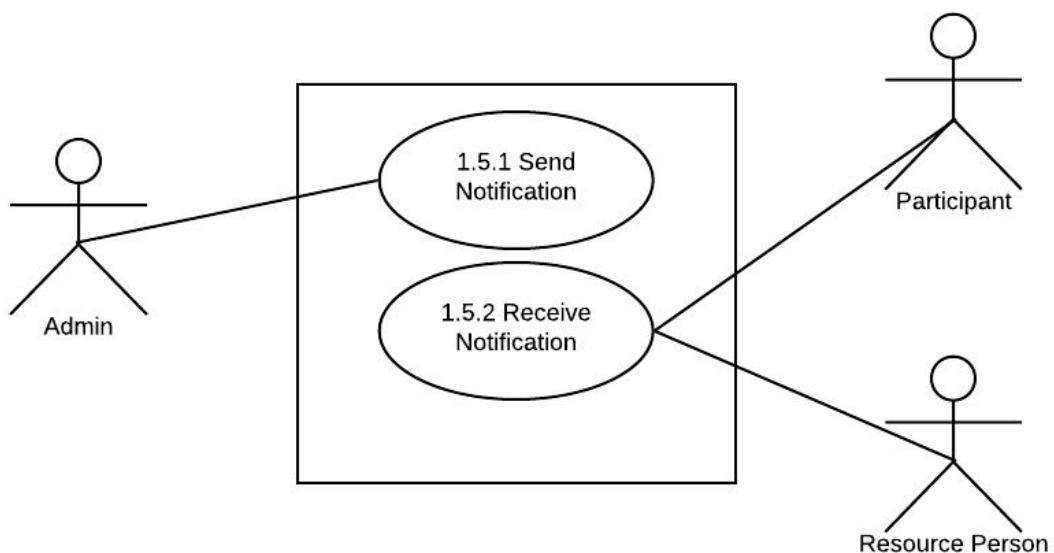


Figure 7. Level-2.5 Use Case Diagram - Notification

4.5.8 Level-2.6 Use Case Diagram - Forum

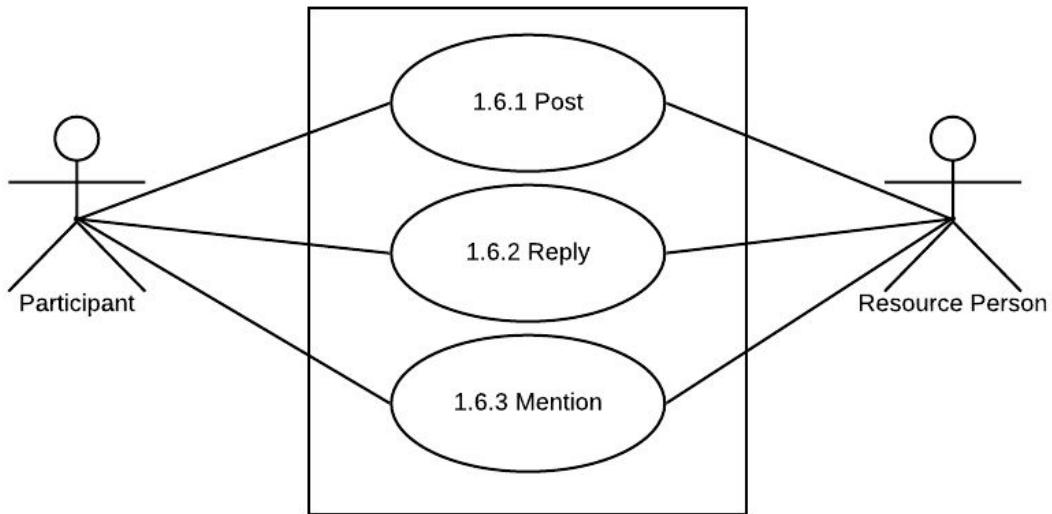


Figure 8. Level-2.6 Use Case Diagram - Forum

4.5.9 Level-2.7 Use Case Diagram - Conversation

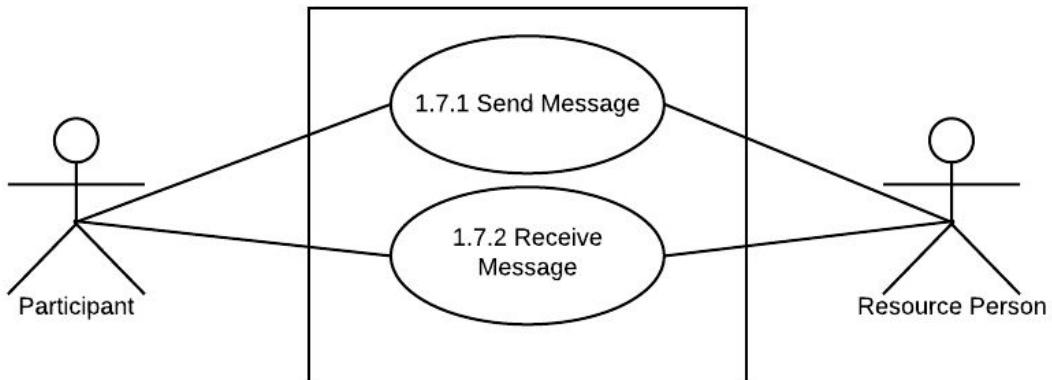


Figure 9. Level-2.7 Use Case Diagram - Conversation

4.5.10 Level-2.8 Use Case Diagram - Things To Do

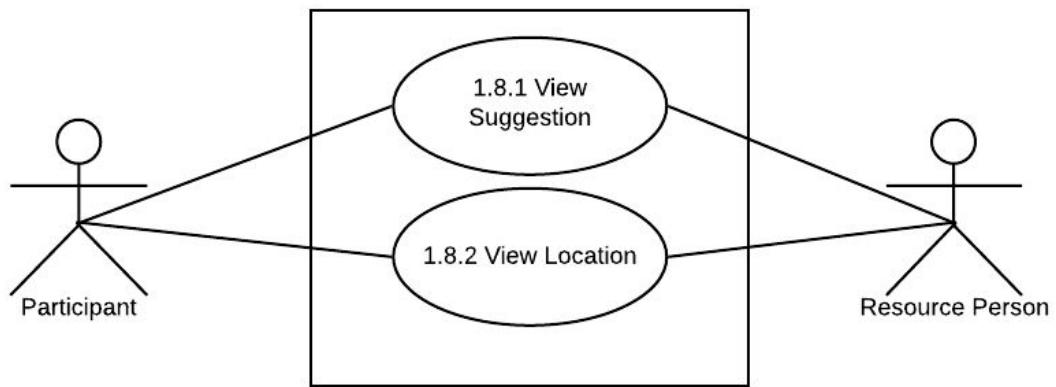


Figure 10. Level-2.8 Use Case Diagram - Things To Do

4.6 Activity Diagrams

4.6.1 Activity: Sign in

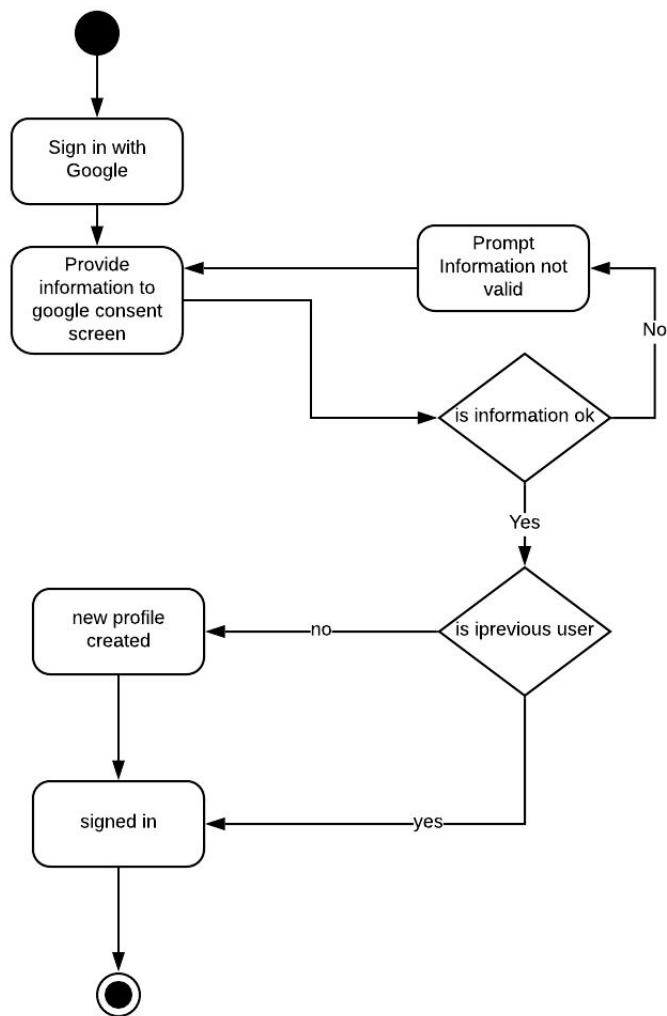


Figure 11. Activity Diagram (Sign In)

4.6.2 Activity: Sign Out

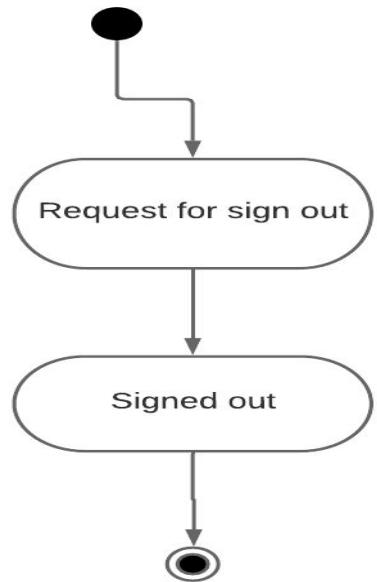


Figure 12. Activity Diagram (Sign Out)

4.6.3 Activity: Modify profile

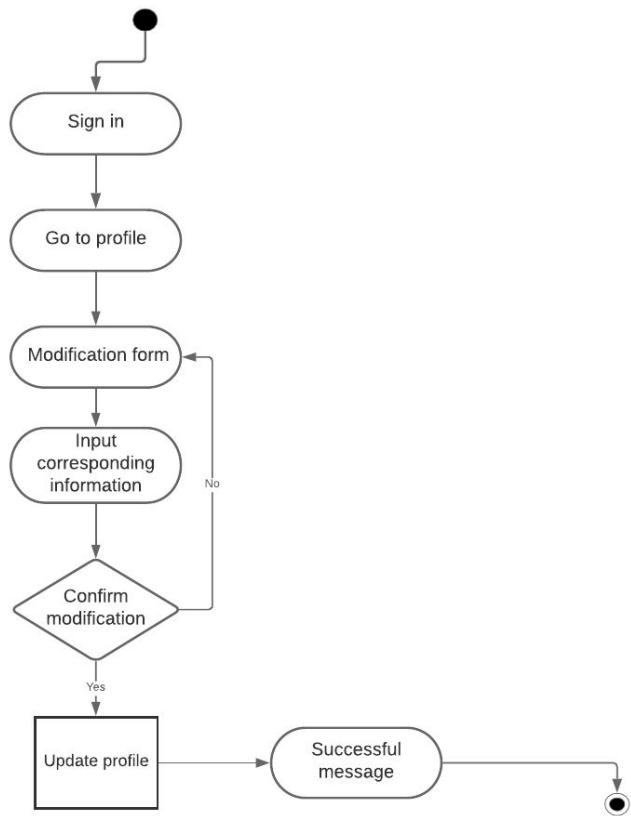


Figure 13. Activity Diagram (Modify profile)

4.6.4 Activity: View Events

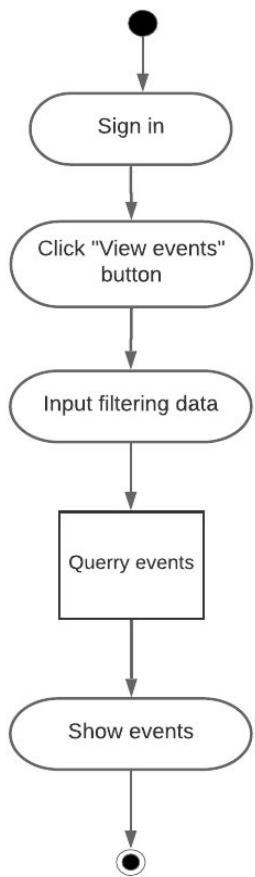


Figure 14. Activity Diagram (View Events)

4.6.5 Activity: Event Creation

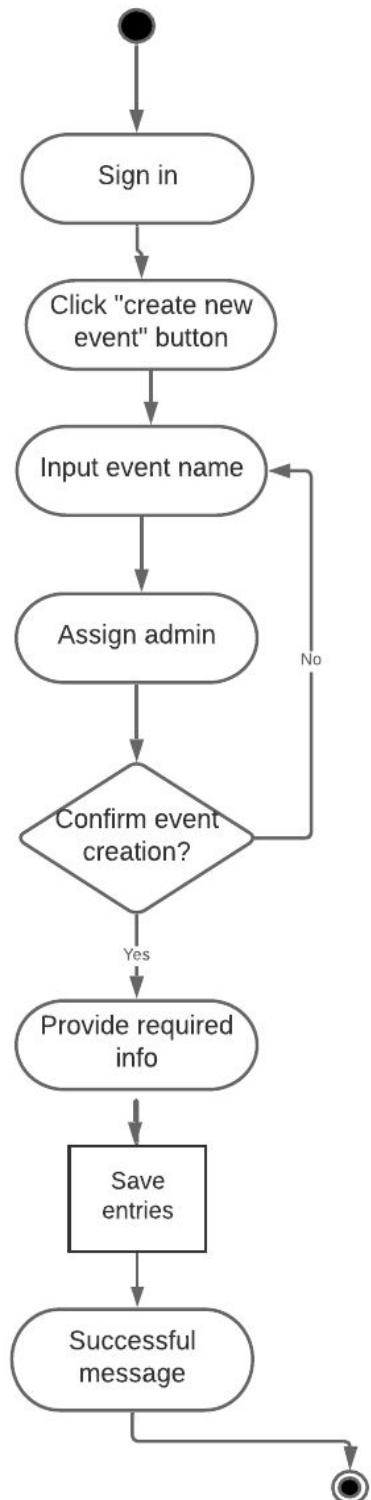


Figure 15. Activity Diagram (Event Creation)

4.6.6 Activity: Joining new Event

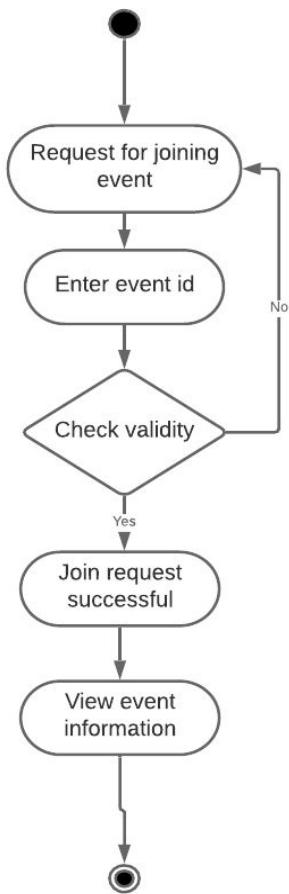


Figure 16. Activity Diagram (Joining New Event)

4.6.7 Activity: Disabling Event

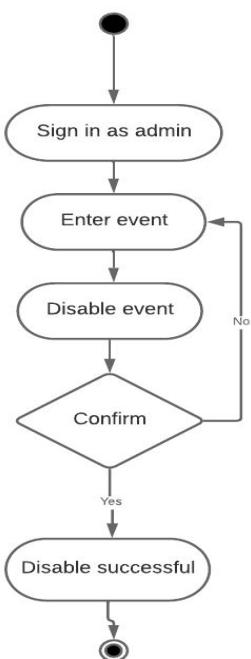


Figure 17. Activity Diagram (Disabling Event)

4.6.8 Activity: View Participant

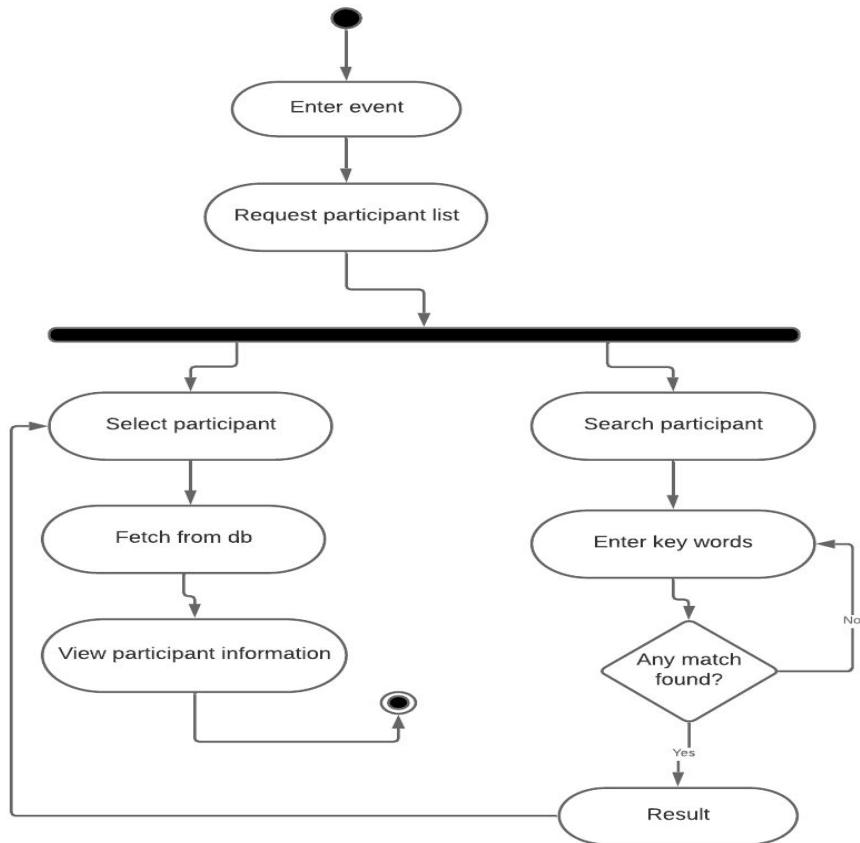


Figure 18. Activity Diagram (View Participant)

4.6.9 Activity: View Resource Person

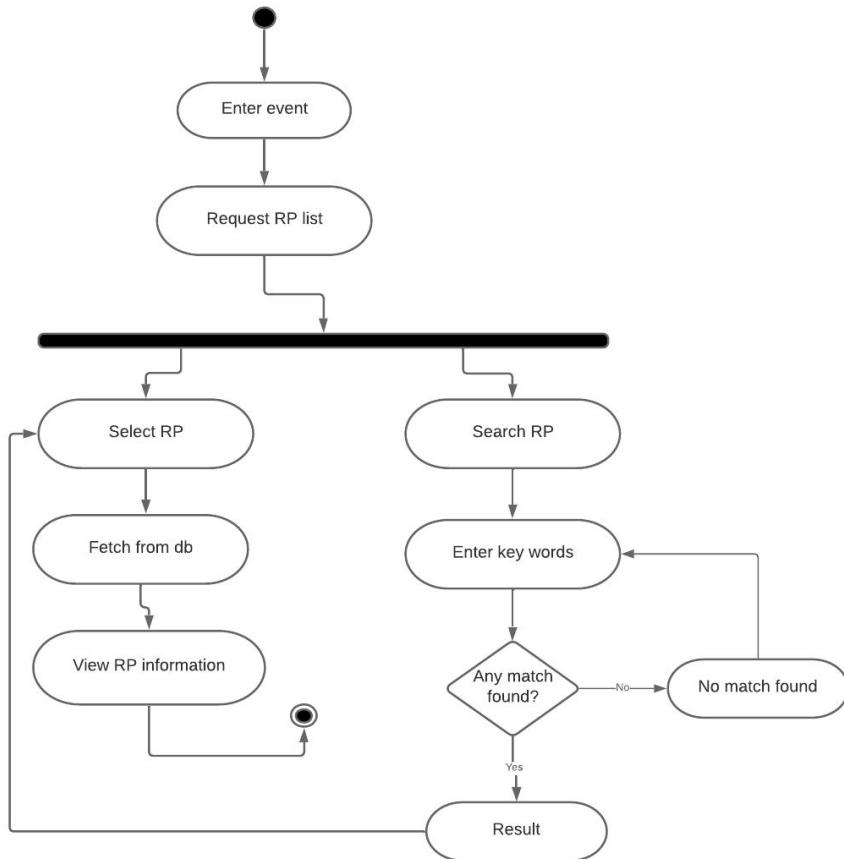


Figure 19. Activity Diagram (View Resource Persons)

4.6.10 Activity: Contact Saving

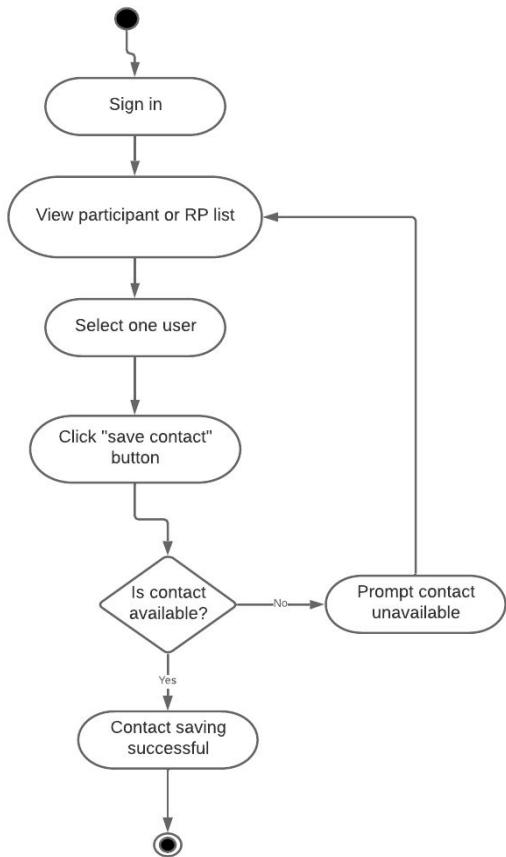


Figure 20. Activity Diagram (Contact Saving)

4.6.11 Activity: Inviting Resource Person

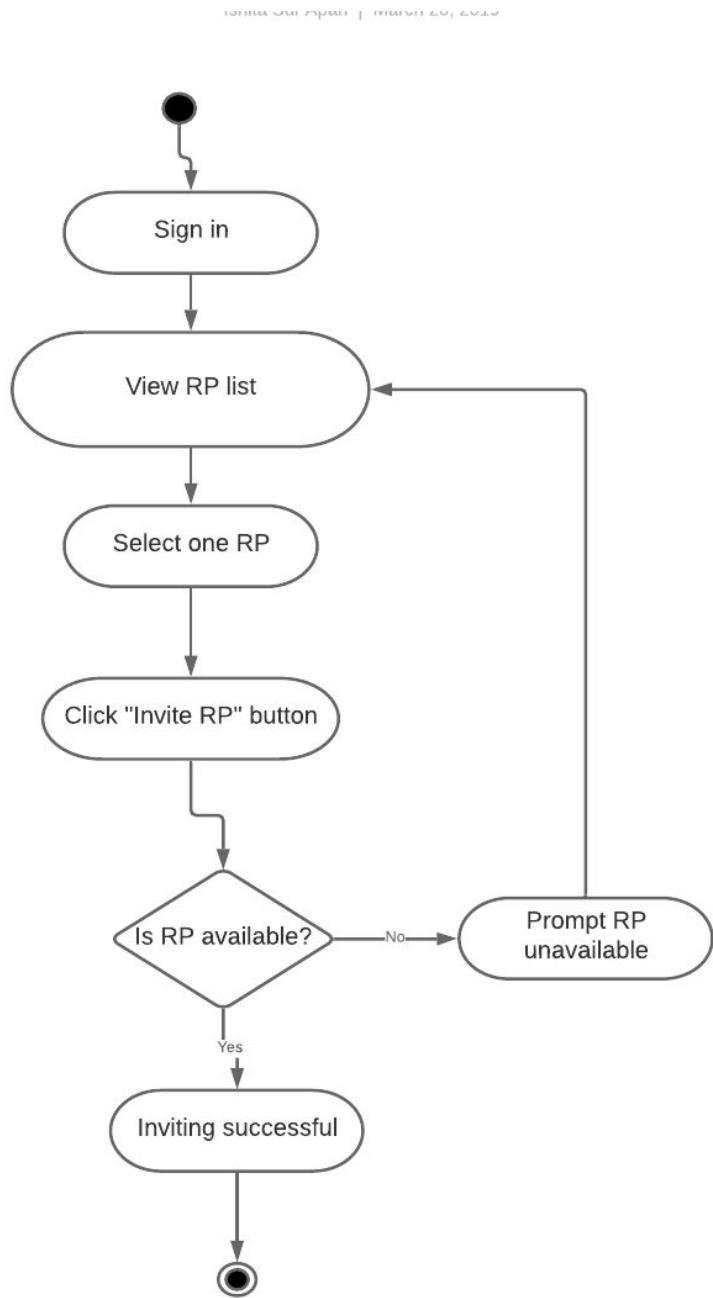


Figure 21. Activity Diagram (Inviting Resource Person)

4.6.12 Activity: Check In

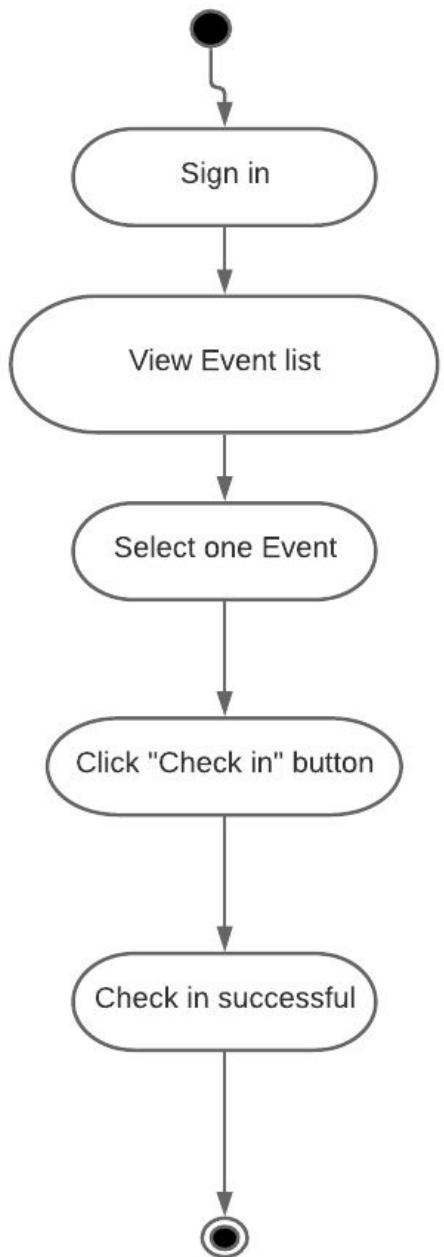


Figure: Activity Diagram (Check In)

4.6.13 Activity: Disable User

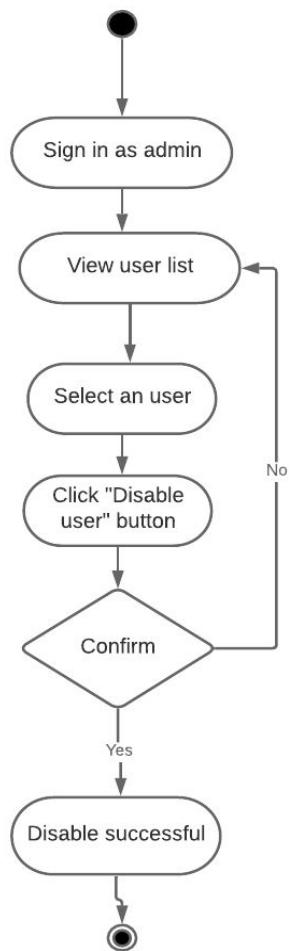


Figure: Activity Diagram (Disable User)

4.6.14 Activity: Uploading File

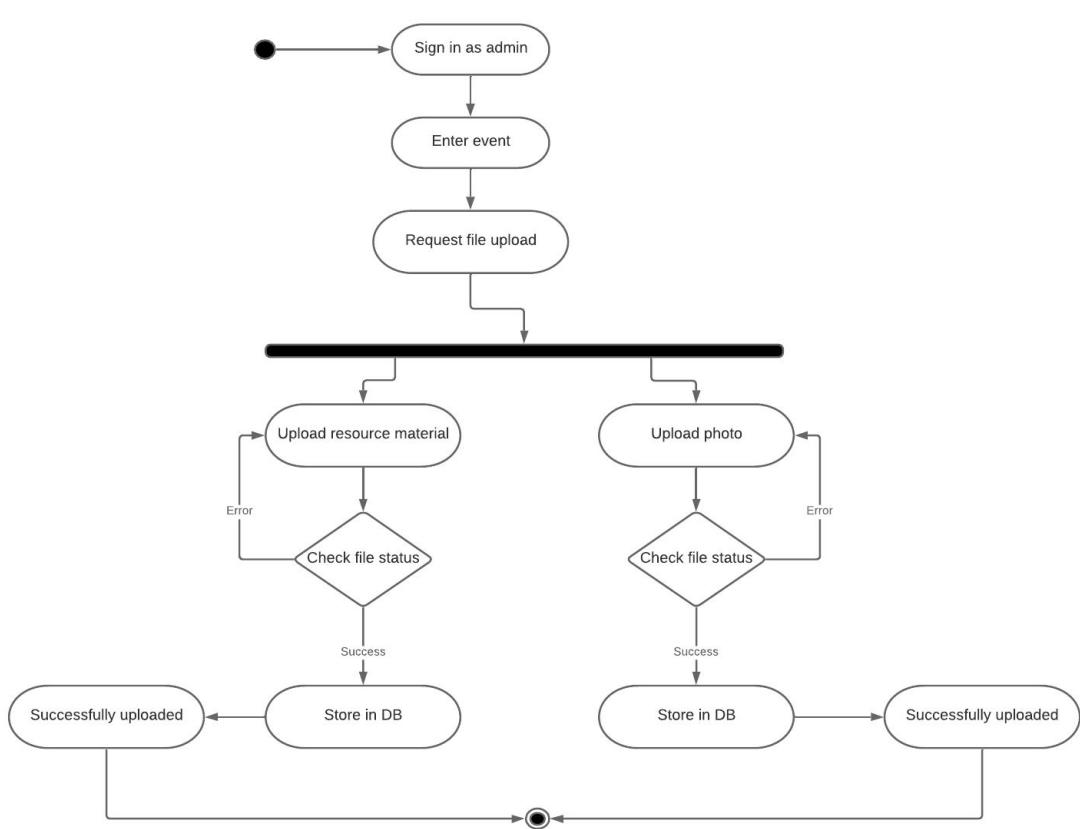


Figure : Activity Diagram (Uploading File)

4.6.15 Activity: Downloading File

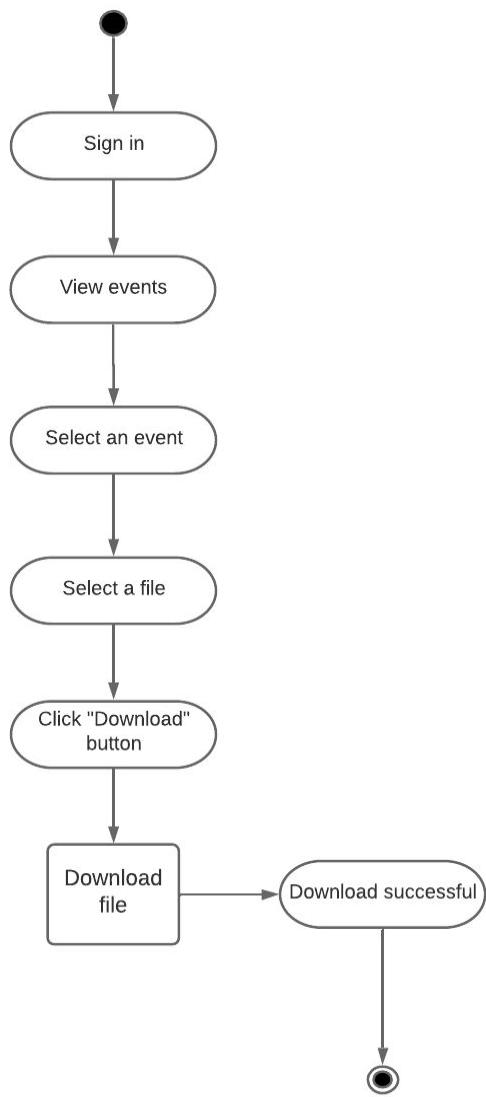


Figure : Activity Diagram (Downloading File)

4.6.16 Activity: Remove File

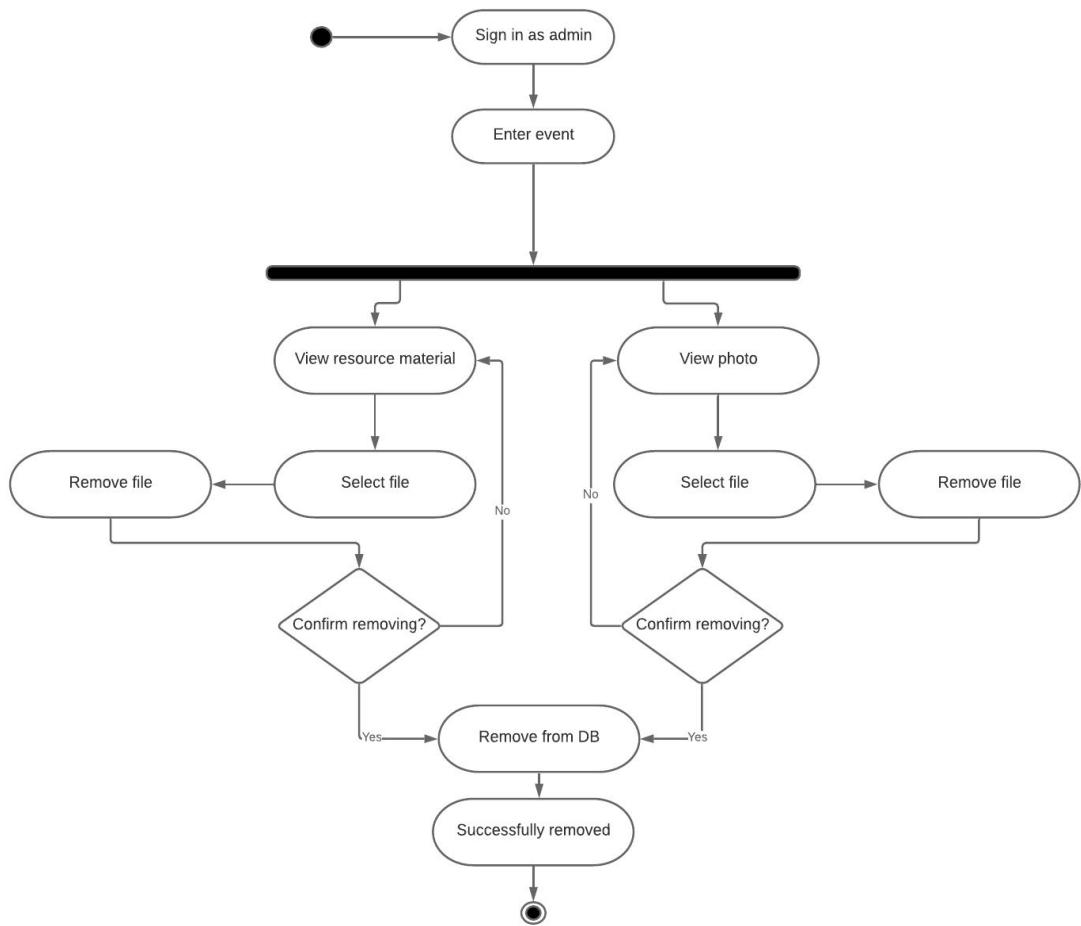


Figure: Activity Diagram (Remove File)

4.6.17 Activity: Reply

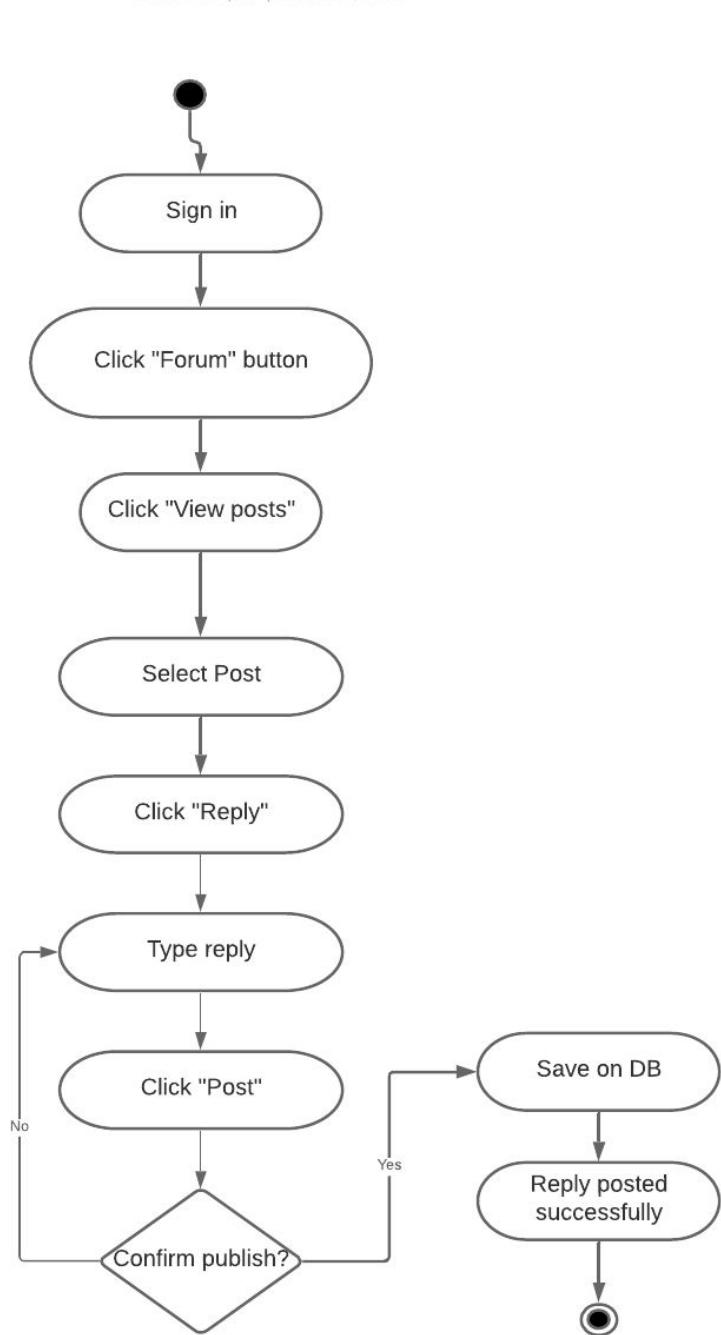


Figure. Activity Diagram (Reply)

4.6.18 Activity: Post

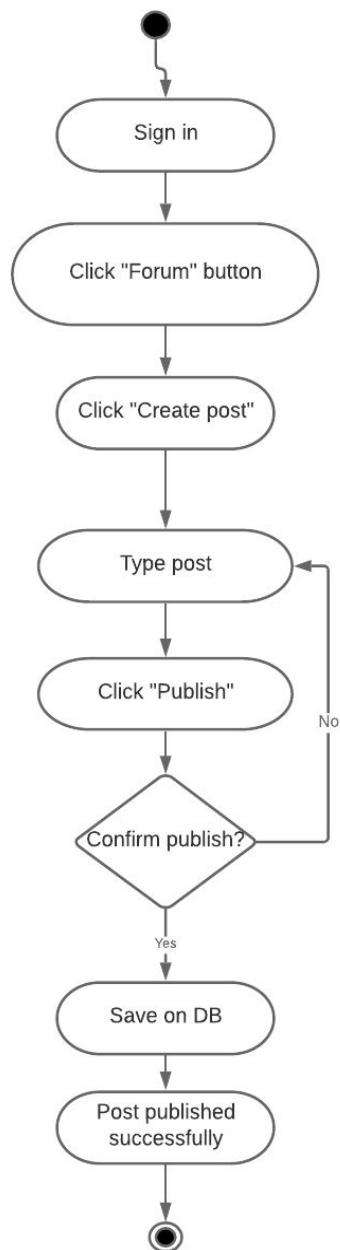


Figure : Activity Diagram (Post)

4.6.19 Activity: Mention

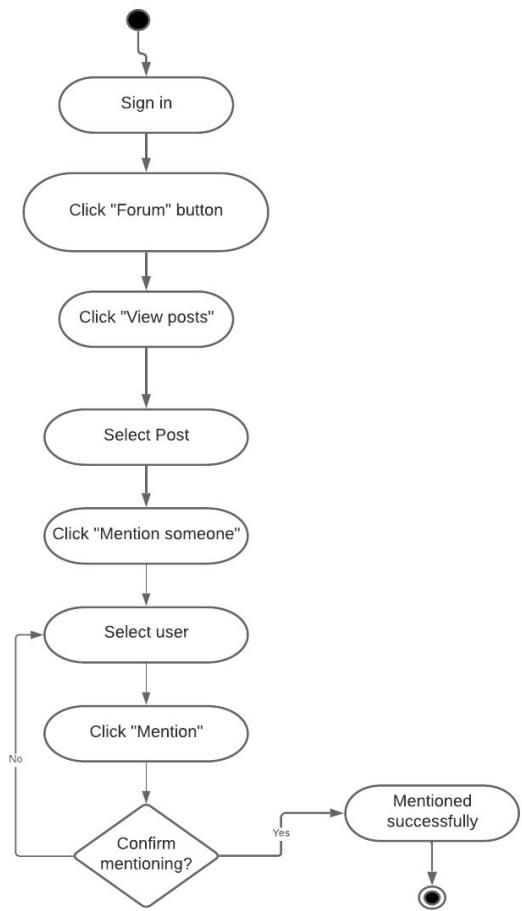


Figure: Activity Diagram (Mention)

4.6.20 Activity: Send Message

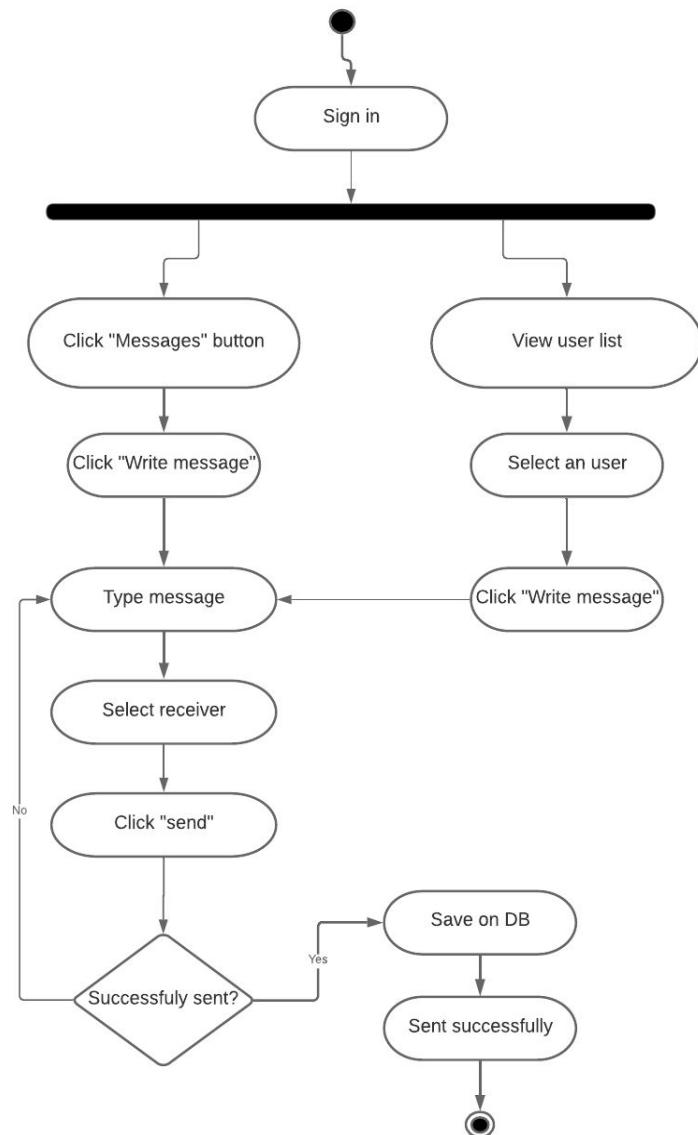


Figure: Activity Diagram (Send Message)

4.6.21 Activity: Receive Message

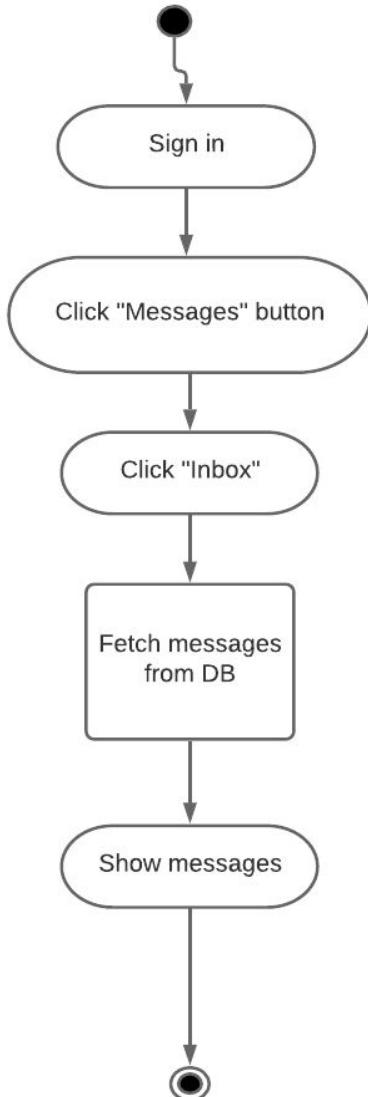


Figure: Activity Diagram (Receive Message)

4.6.22 Activity: Send Notification

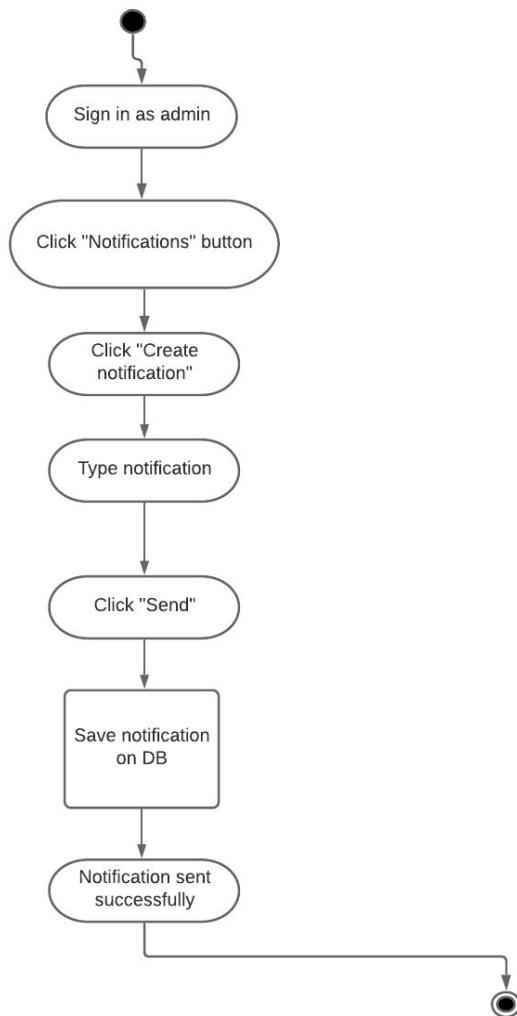


Figure: Activity Diagram (Send Notification)

4.6.23 Activity: Receive Notification

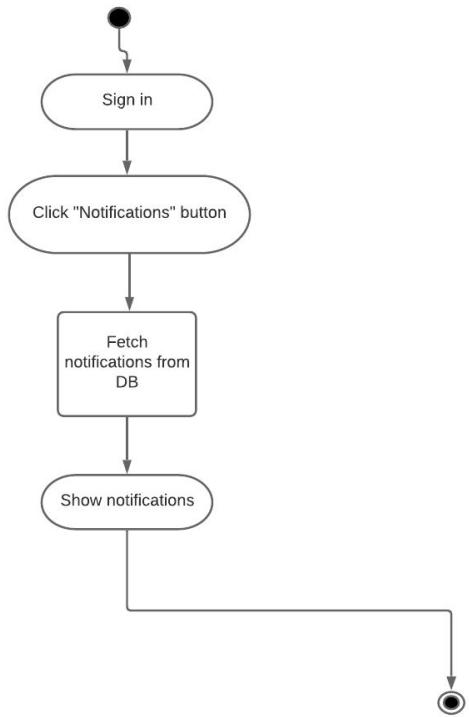


Figure: Activity Diagram (Receive Notification)

4.6.24 Activity: View Photo Gallery

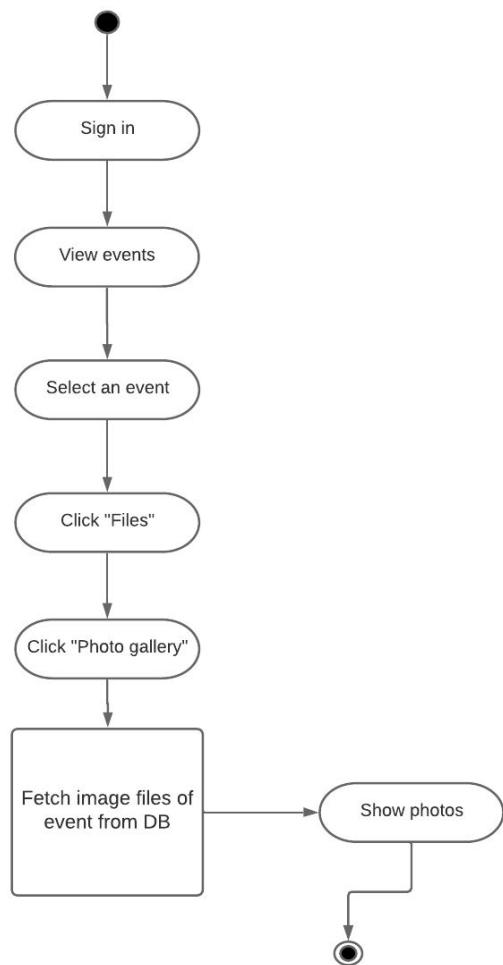


Figure: Activity Diagram (Disabling Event)

4.6.25 Activity: View Resource Files

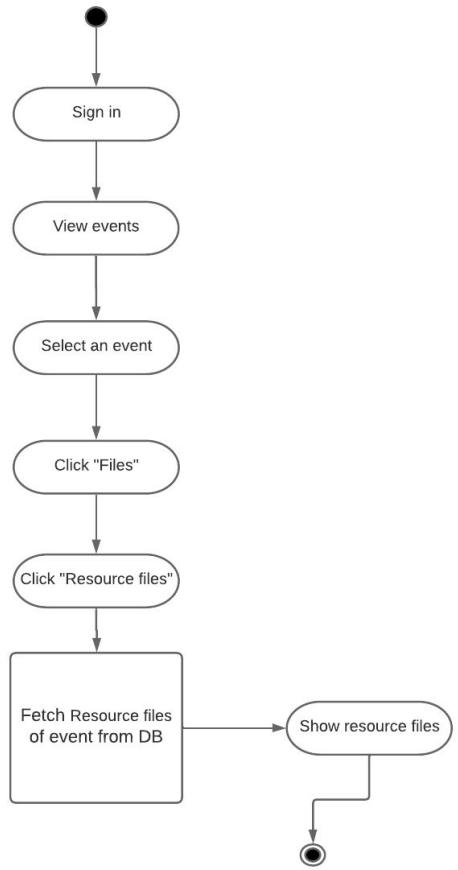


Figure: Activity Diagram (View Resource Files)

4.6.26 Activity: View Location

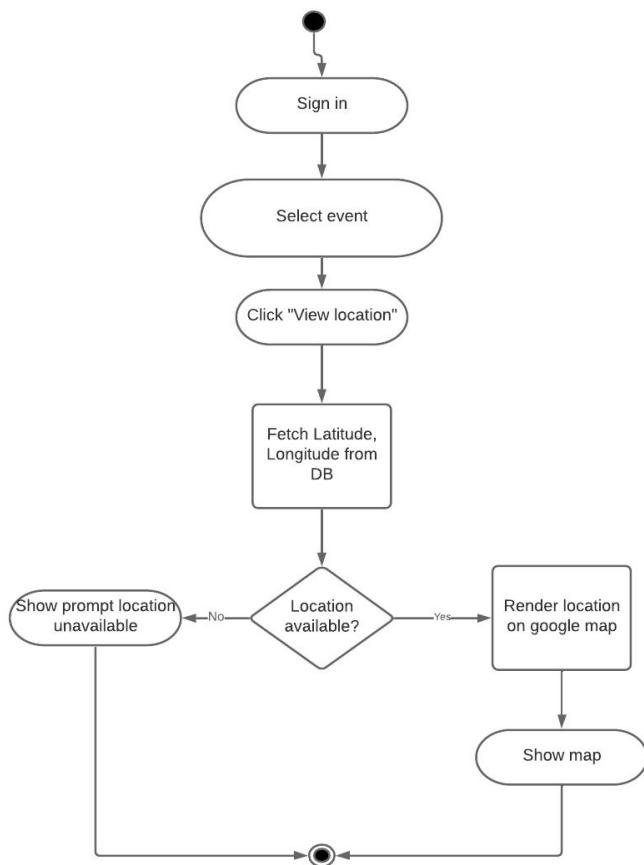


Figure: Activity Diagram (View Location)

4.7 Swimlane Diagrams

4.7.1 Use Case: Sign In

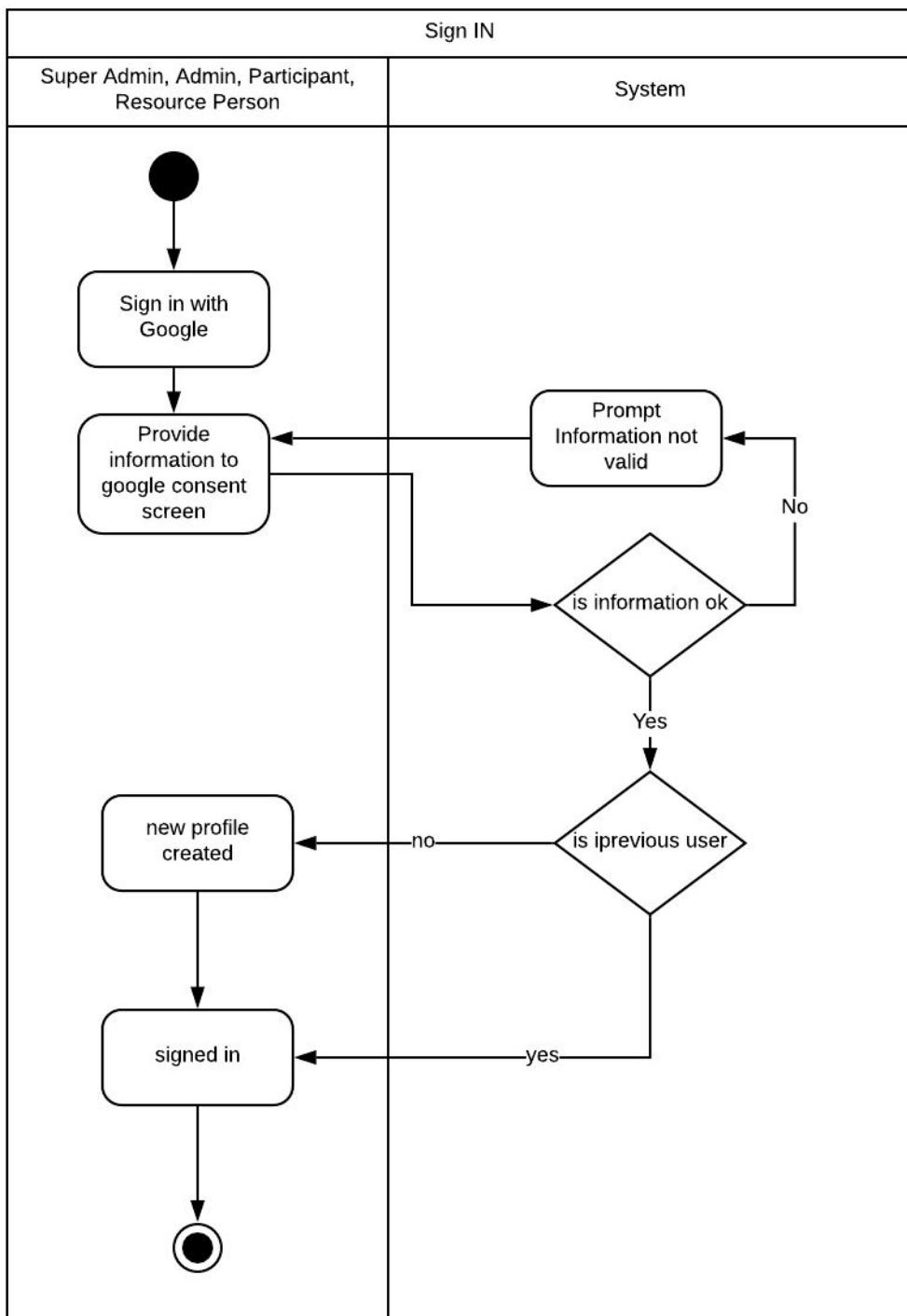


Figure. Swimlane Diagram (Sign In)

4.7.2 Use Case: sign out

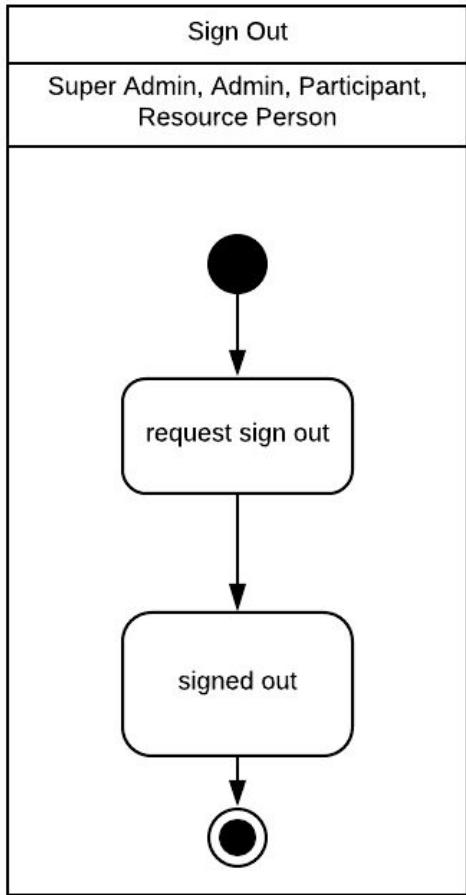


Figure. Swimlane Diagram (Sign Out)

4.7.3 Use Case: Creating Event

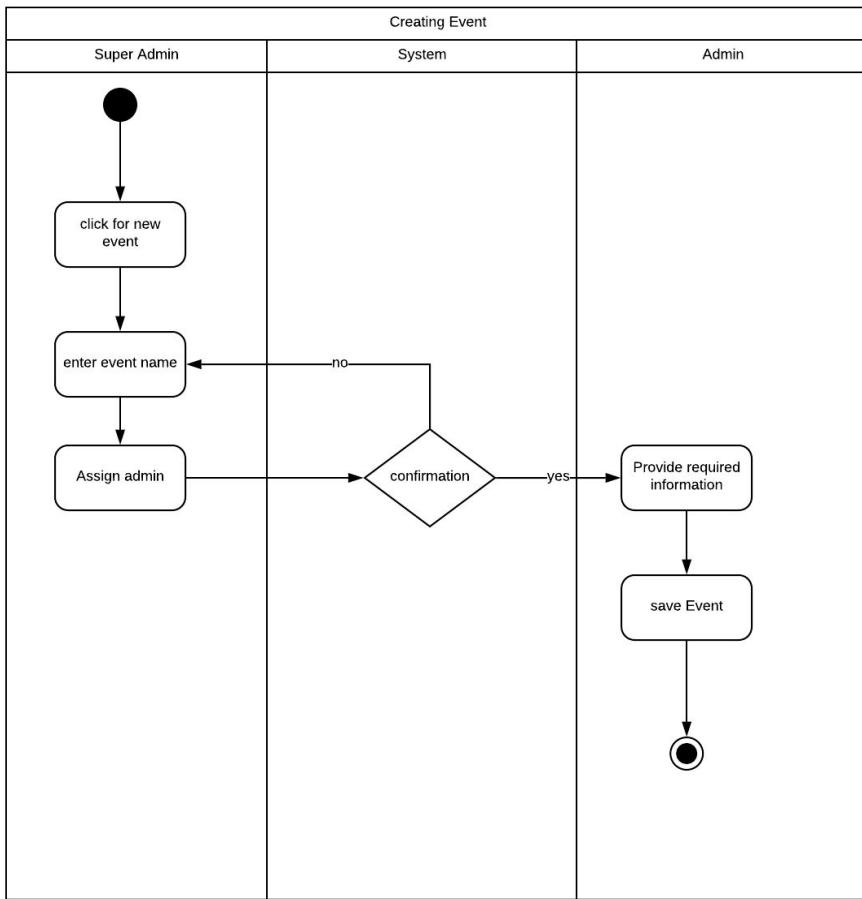


Figure. Swimlane Diagram (Creating Event)

4.7.4 Use Case: Updating Event

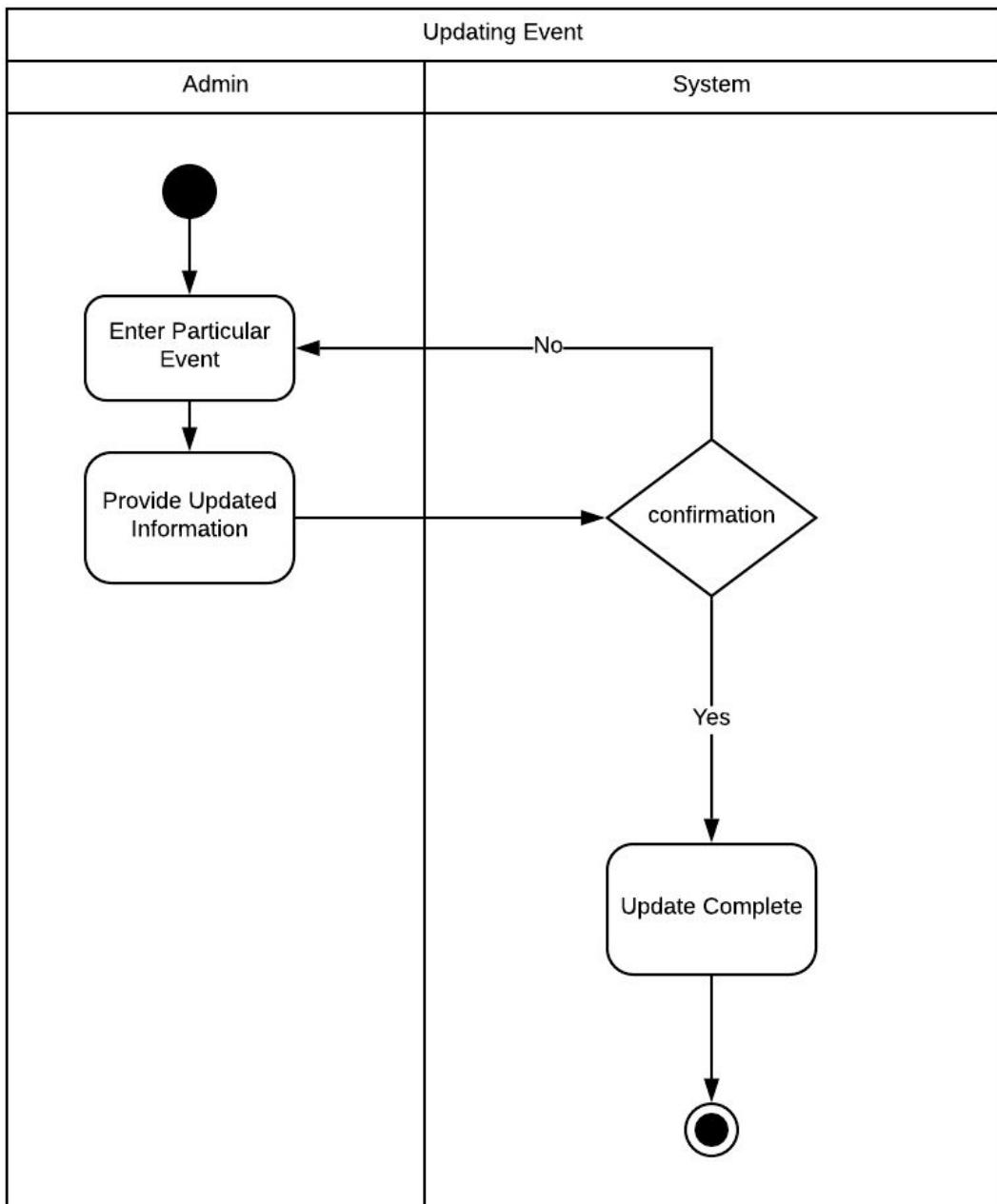


Figure. Swimlane Diagram (Updating Event)

4.7.5 Use Case: Joining New Event

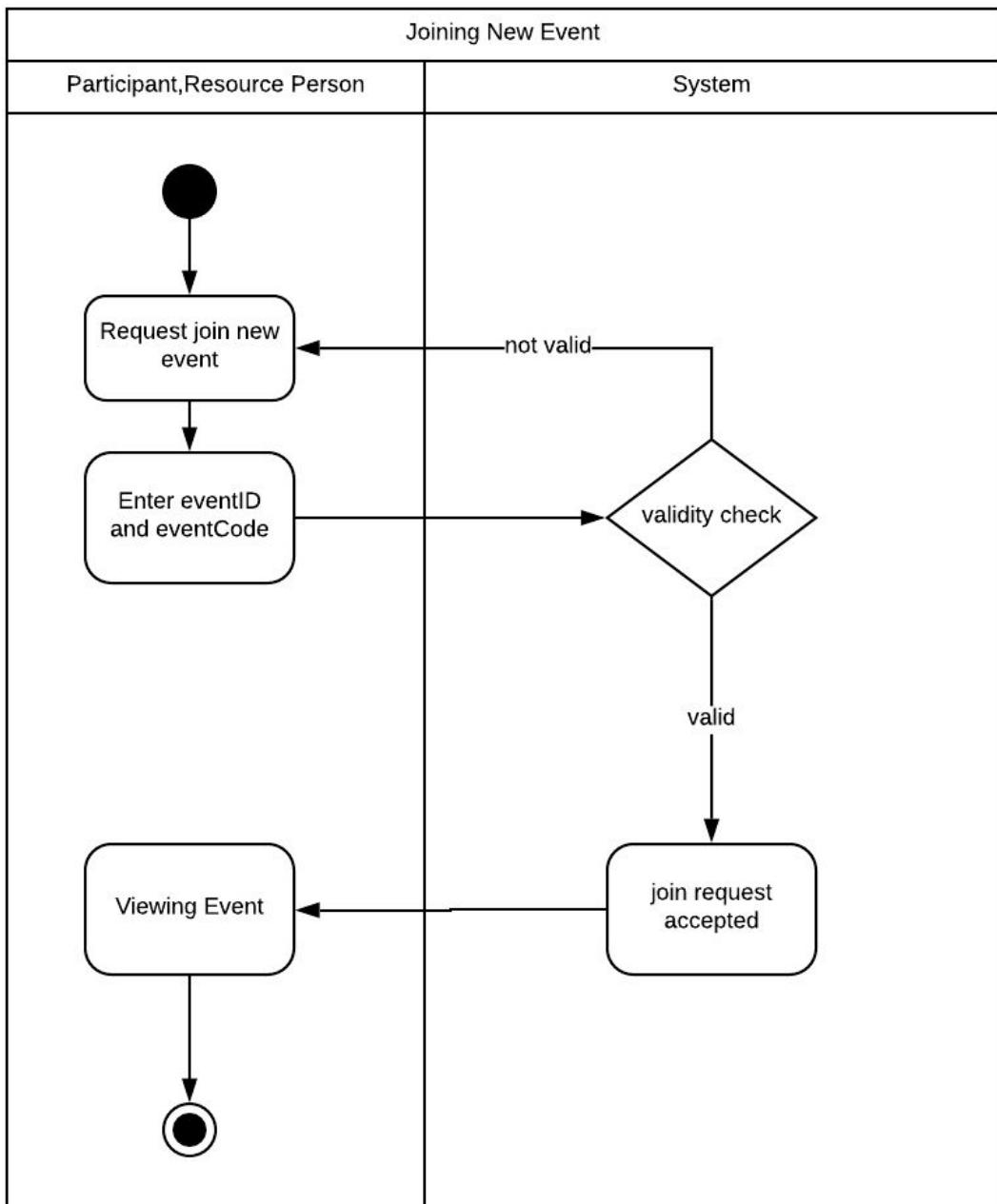


Figure. Swimlane Diagram (Joining New Event)

4.7.6 Use Case: Disabling Event

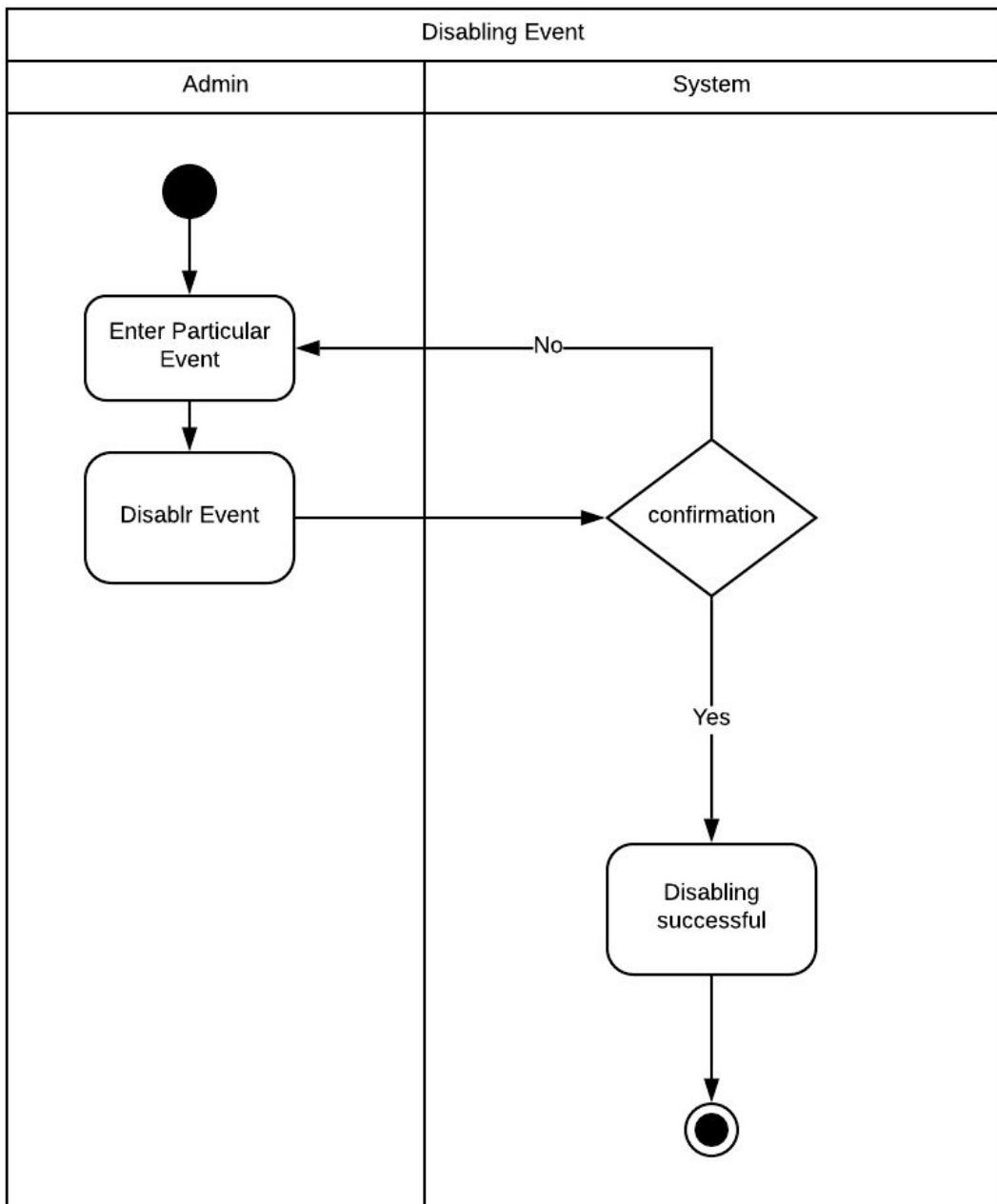


Figure. Swimlane Diagram (Disabling Event)

4.7.7 Use Case: Viewing Participant /Resource Person

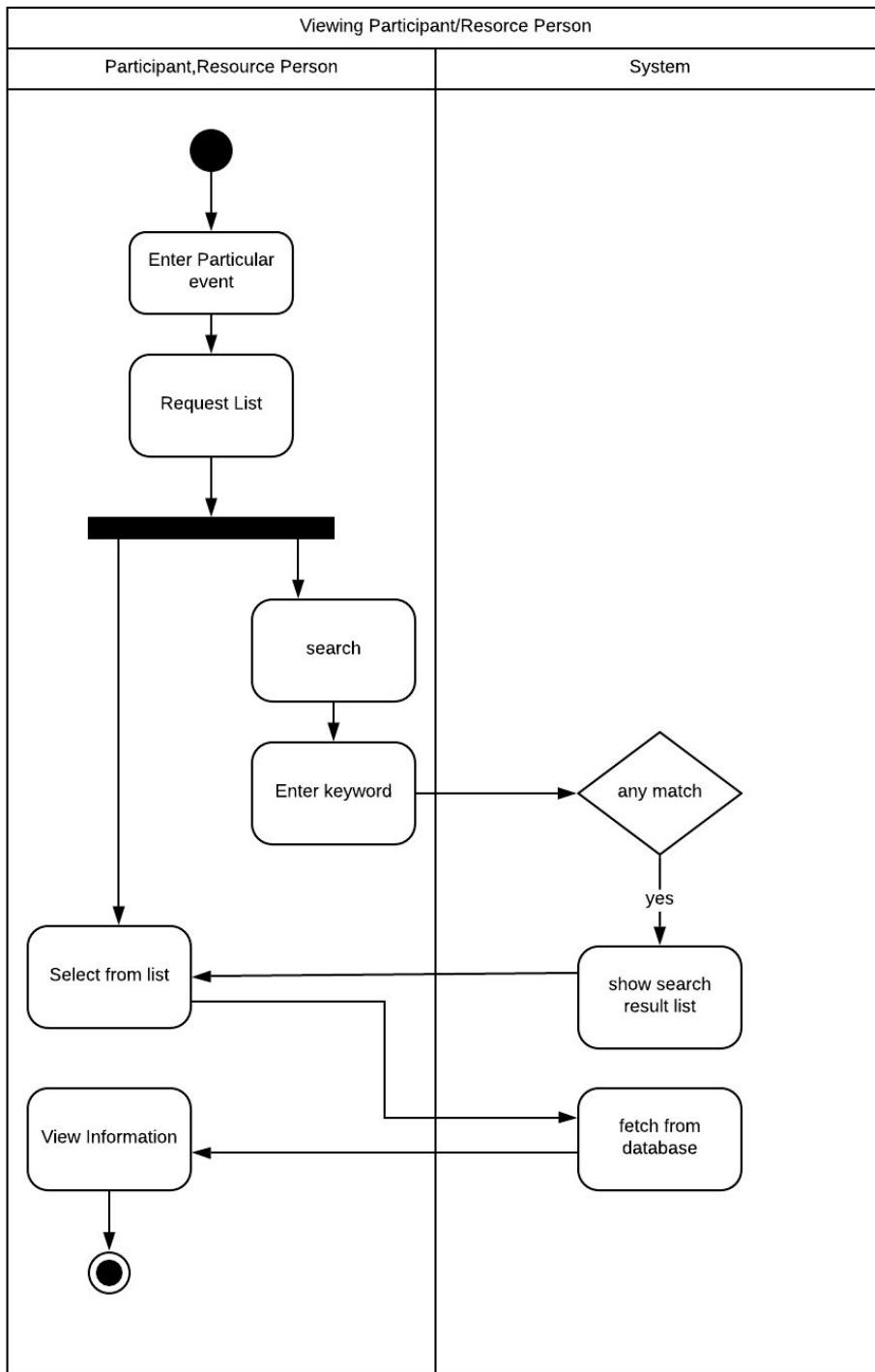


Figure. Swimlane Diagram (Viewing Participant/Resource Person)

4.7.8 Use Case: Modifying User Profile

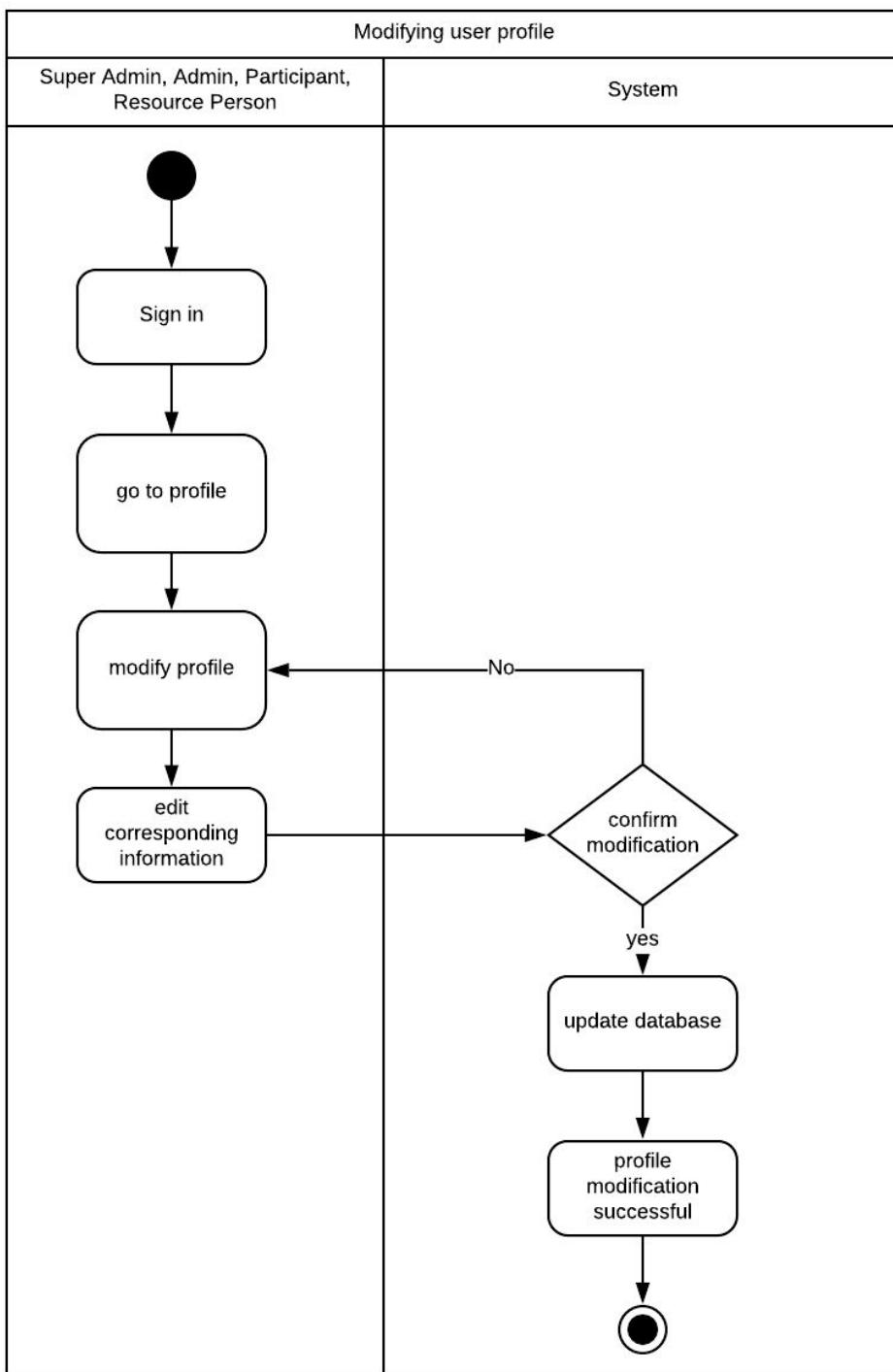


Figure. Swimlane Diagram (Modifying User Profile)

4.7.9 Use Case: Download File

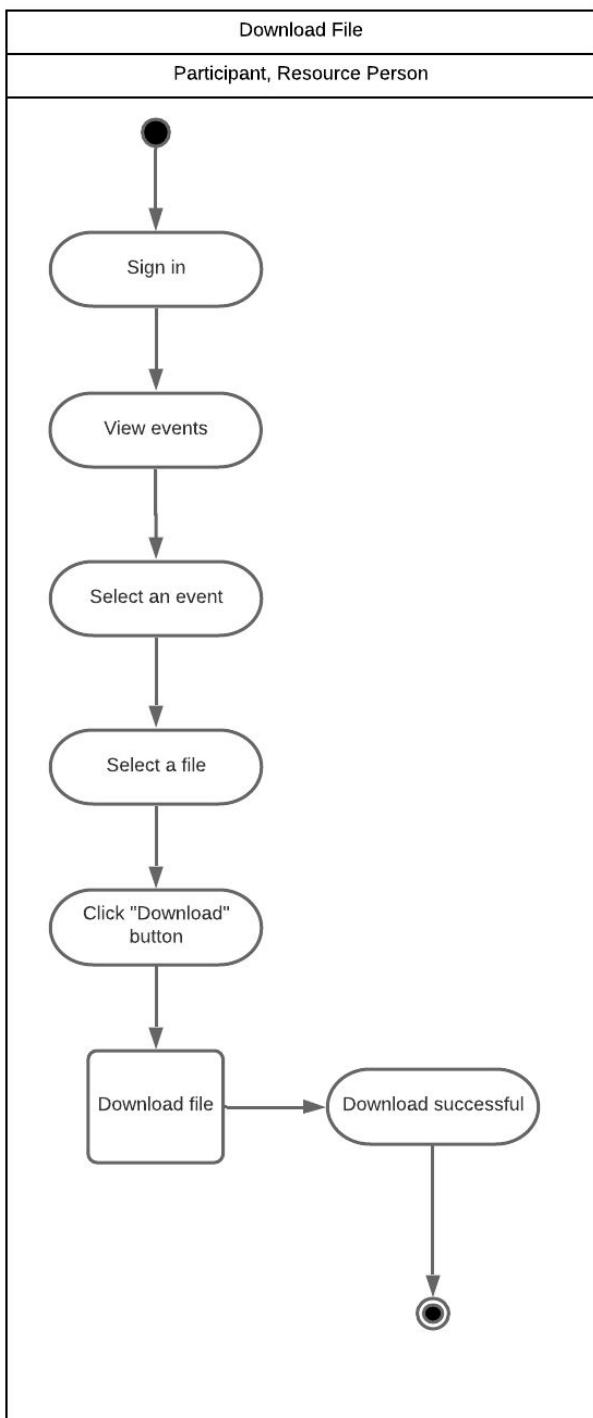


Figure. Swimlane Diagram (Downloading File)

4.7.10 Use Case: Contact Saving

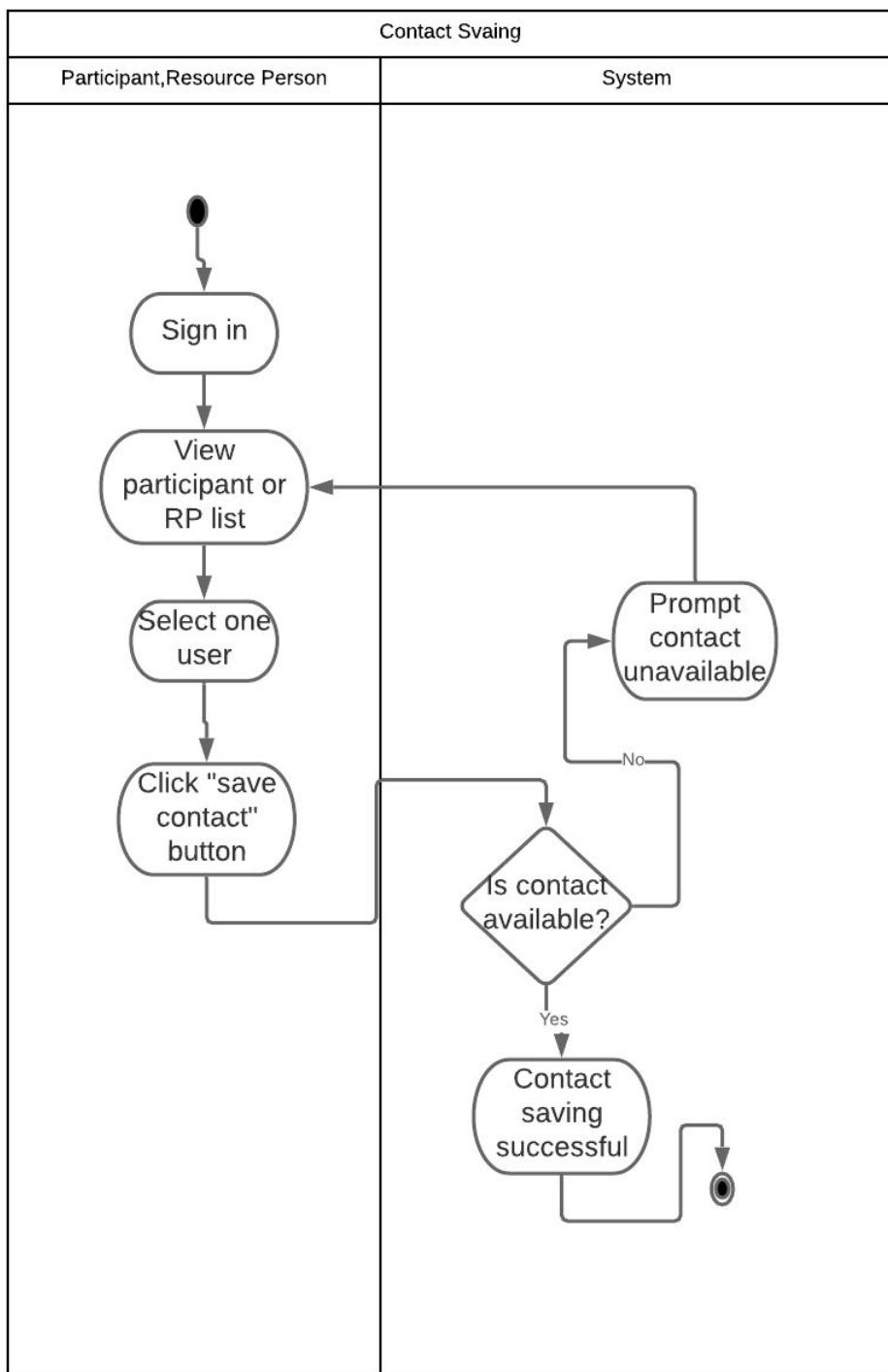


Figure. Swimlane Diagram (Contact Saving)

4.7.11 Use Case: Receive Notification

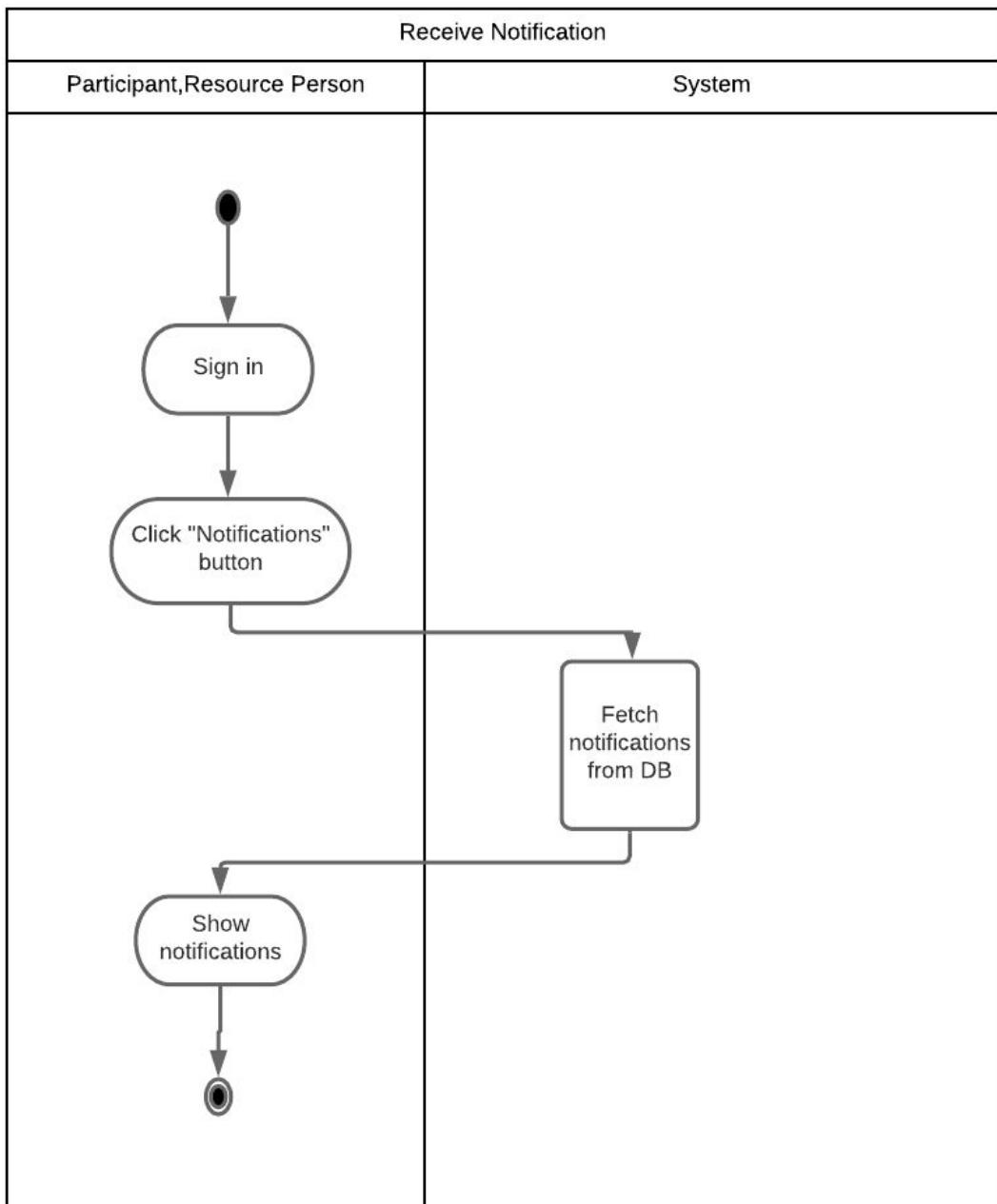


Figure. Swimlane Diagram (Receive Notification)

Chapter 5

DATA MODELING

5.1 Data Modeling Concepts

If software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

5.2 Data Object Selection

A data object is a representation of composite information that must be understood by the software. Here, composite information means an information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

5.2.1 Noun Identification

No.	Noun	Problem/Solution Space	Attribute
1.	Event Meet App	P	
2.	Android application	P	
3.	Platform	P	
4.	Event	S	19,20, 23-26, 47
5.	Workshop	P	
6.	Seminar	P	
7.	User	S	34-42

8.	Organizer	S	37-42
9.	Participant	S	10, 34-42,45
10.	Participant ID	S	
11.	Resource Person	S	12, 34-42
12.	Resource Person ID	S	
13.	Admin	S	14, 34-42
14.	Admin ID	S	
15.	Super Admin	S	16, 34-42
16.	Super Admin ID	S	
17.	Verification	P	
18.	Activity	P	
19.	Event ID	S	
20.	Event Code	S	
21.	Event Creation	P	
22.	Information	P	
23.	Event Name	S	
24.	Description	S	
25.	Location	S	27
26.	Schedule	S	
27.	Google Map API	S	
28.	Web Service	S	
29.	Authentication	P	34
30.	System	S	
31.	OAuth	S	

32.	Information Management	P	
33.	Profile fields	P	
34.	Google Account	S	
35.	User ID	S	
36.	User Name	S	
37.	Address	S	
38.	Contact No	S	
39.	Email	S	
40.	Occupation	S	
41.	Date of Birth	S	
42.	Nationality	S	
43.	Contact Saving	P	
44.	Phonebook	P	
45.	Status	S	
46.	Event Info Management	P	
47.	Lifetime	P	
48.	Time Stamp	S	
49.	Database	S	
50.	Authority	P	
51.	My Event List	S	
52.	File Management	P	
53.	Materials	P	
54.	Slides	S	

55.	Pdf	S	
56.	Photos	S	
57.	Photo Gallery	S	
58.	Files	S	59-63
59.	File ID	S	
60.	File name	S	
61.	File type	S	
62.	File Upload Time	S	
63.	File Content	S	
64.	Resources	S	
65.	Notification	S	68-70
66.	Notices	S	
67.	Reminder	S	
68.	Notification ID	S	
69.	Notification type	S	
70.	Notification Content	S	
71.	Plain text	S	
72.	Communication	P	
73.	Forum	S	74-76
74.	Forum ID	S	
75.	Forum Type	S	
76.	Post	S	77-80
77.	Post ID	S	
78.	Post Subject	S	

79.	Post Content	S	
80.	Date	S	
81.	Question	S	
82.	Answer	S	83
83.	Answer content	S	
84.	Feedback	S	
85.	Storage	P	
86.	Data	S	
87.	Conversation	P	88
88.	Message	S	
89.	Things To Do	P	
90.	Suggestion	S	
91.	Restaurant	P	
92.	Resort	P	
93.	Event Spot	P	
94.	Direction	S	

5.2.2 Potential Data Objects

Event : 19,20, 23-26, 47

User : 34-42

Organizer : 37-42

Participant : 10, 34-42,45

Resource Person : 12, 34-42

Admin : 14, 34-42

Super Admin : 16, 34-42

File : 59-63

Notification : 68-70

Forum : 74-76

Post : 77-80

5.2.3 Analysis for Finalizing Data Objects

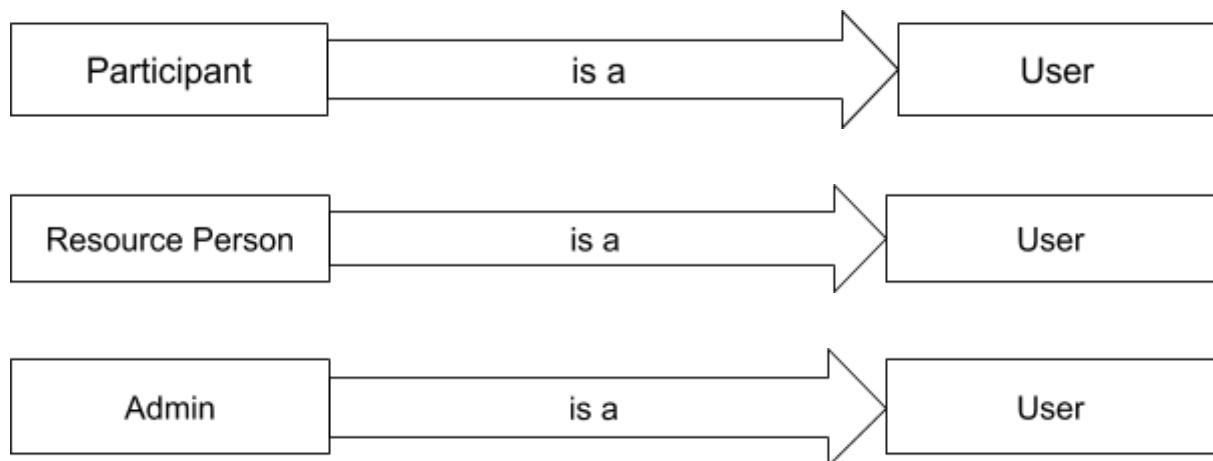
1. Both Super Admin, Admin, Participant and Resource Person have some common attributes. So their common attributes can be stored as **User**.
2. Both the information and activities of Super admin and Admin must be stored in **Admin**.
3. Details of participants of an event should be stored in **Participant** for updating information and tracking their activities.
4. Information about resource persons of each event can be stored in **Resource Person**.
5. An **Event** table stores each and every event details both online and offline.
6. Users receive Notification. Admin can set and system can generate notification. So notification needs to be saved and all information must be stored in **Notification**.
7. **Files** uploaded by Admin must be stored as users can download them throughout event lifetime.
8. Participants and resource person can post in Forum and it can be viewed within the event lifetime. So it must be stored in **Forum** table.
9. In forum, users can give posts and also can answer them. So **Post** should be stored.

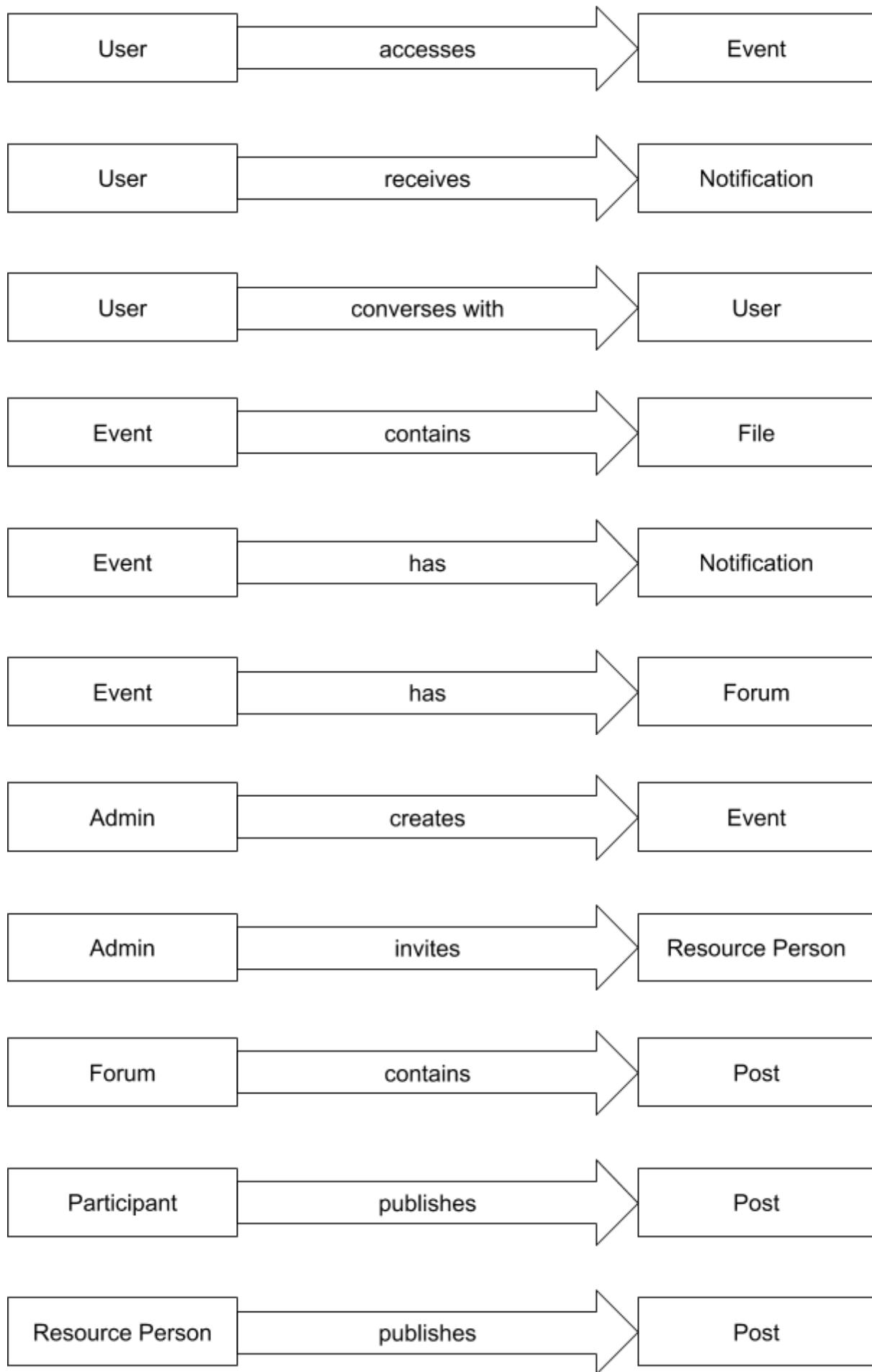
5.2.4 Final Data Objects

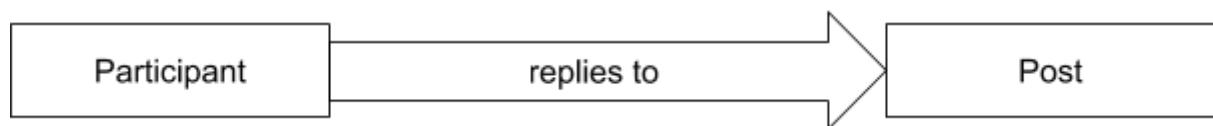
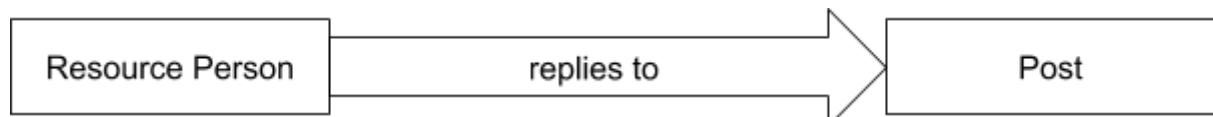
No	Entity	Attributes
1	Event	Event ID, Event Code, Event name, description, location, schedule, lifetime
2	User	User ID, Google Account, user name, address, contact no, email, occupation, date of birth, nationality
3	Admin	Admin ID, User ID, Google Account, user name, address, contact no, email, occupation, date of birth, nationality, status

4	Participants	Participant ID, User ID, Google Account, user name, address, contact no, email, occupation, date of birth, nationality, status
5	Resource Person	Resource Person ID, User ID, Google Account, user name, address, contact no, email, occupation, date of birth, nationality, status
6	Notification	Notification ID, Notification type, Notification content
7	File	File ID, File Name, File type, File upload time, File content
8	Forum	Forum ID, Forum type, Forum post
9	Post	Post ID, Post content, Post subject, Date

5.3 Data Objects Relation







5.4 Entity Relationship Diagram

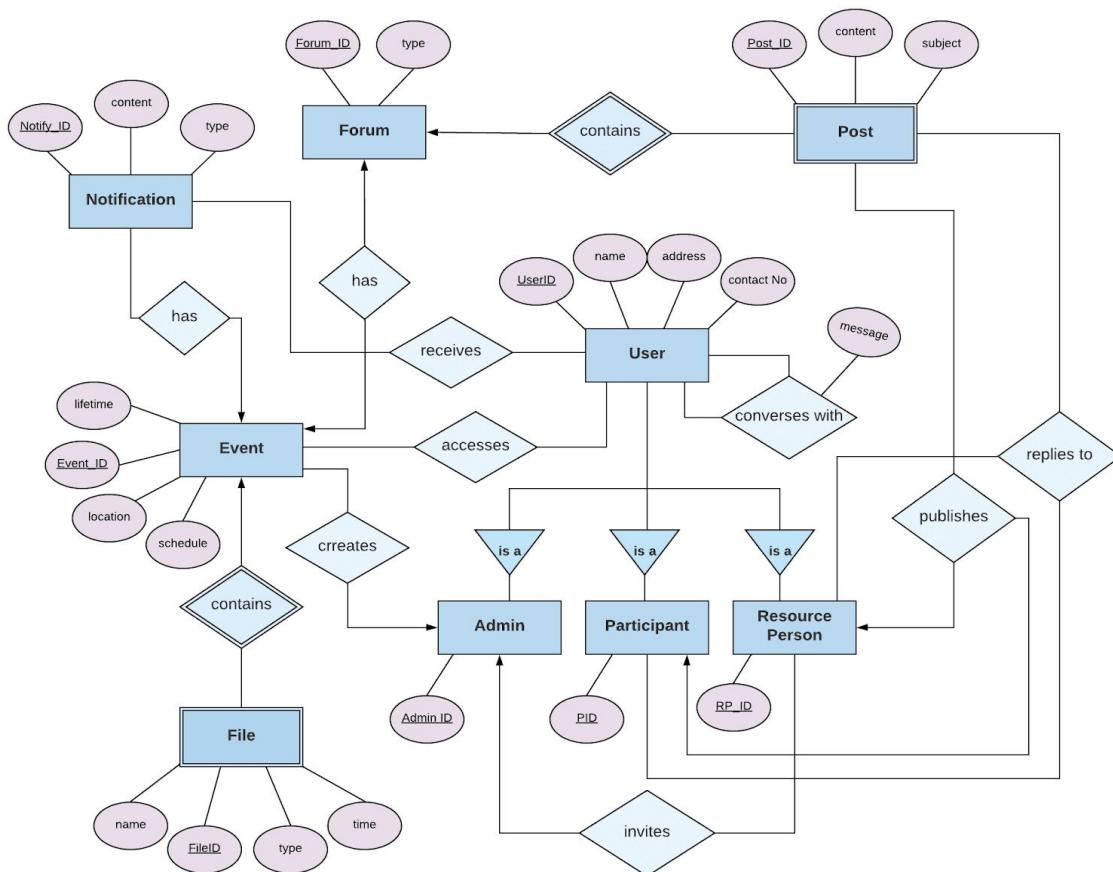


Figure: ER Diagram

5.5 Schema Diagram

Table : Schema for Event

Event		
Attributes	Type	Size
Event ID	Varchar2	20
Event Code	Varchar2	20
Event name	Varchar2	80
Description	Varchar2	200
Location	Varchar2	80
Schedule	Varchar2	20
Lifetime	Date	20

Table : Schema for User

User		
Attributes	Type	Size
User ID	Varchar2	20
User name	Varchar2	80
Address	Varchar2	80
Contact No	Varchar2	80
Email	Varchar2	80
Occupation	Varchar2	80
Date of Birth	Date	20
Nationality	Varchar2	80

Table : Schema for Admin

Admin		
Attributes	Type	Size

Admin ID	Varchar2	20
User ID	Varchar2	20
User name	Varchar2	80
Address	Varchar2	80
Contact No	Varchar2	80
Email	Varchar2	80
Occupation	Varchar2	80
Date of Birth	Date	20
Nationality	Varchar2	80
Status	Varchar2	80

Table : Schema for Participant

Participant		
Attributes	Type	Size
Participant ID	Varchar2	20
User ID	Varchar2	20
User name	Varchar2	80
Address	Varchar2	80
Contact No	Varchar2	80
Email	Varchar2	80
Occupation	Varchar2	80
Date of Birth	Date	20
Nationality	Varchar2	80
Status	Varchar2	80

Table : Schema for Resource Person

Resource Person		
Attributes	Type	Size

Resource Person ID	Varchar2	20
User ID	Varchar2	20
User name	Varchar2	80
Address	Varchar2	80
Contact No	Varchar2	80
Email	Varchar2	80
Occupation	Varchar2	80
Date of Birth	Date	20
Nationality	Varchar2	80
Status	Varchar2	80

Table : Schema for Notification

Notification		
Attributes	Type	Size
Notification ID	Varchar2	20
Notification Type	Varchar2	80
Notification Content	Varchar2	500

Table : Schema for File

File		
Attributes	Type	Size
File ID	Varchar2	20
File Name	Varchar2	80
File Type	Varchar2	80
File Upload Time	Date	20
File Content	Varchar2	300

Table : Schema for Forum

Forum		
--------------	--	--

Attributes	Type	Size
Forum ID	Varchar2	20
Forum Post	Varchar2	500
Forum Type	Varchar2	80
Forum Creation Time	Date	20

Table : Schema for Post

Post		
Attributes	Type	Size
Post ID	Varchar2	20
Post Content	Varchar2	500
Post Subject	Varchar2	80
Post Time	Date	20

Chapter 6

CLASS BASED MODELING

This Chapter is intended to describe class based modeling of “EventMeetApp”.

6.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. These were then characterized in seven general classification. The seven general characteristics are as follows

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the specifications of the nouns according to the general classifications:

Table : General Classification of Noun

No.	Noun	General Classification

1.	Event Meet App	7
2.	Android application	7
3.	Platform	6
4.	Event	2,5,7
5.	Workshop	4
6.	Seminar	4
7.	User	4,5,7
8.	Organizer	4,5,7
9.	Participant	4,5,7
10.	Participant ID	
11.	Resource Person	4,5,7
12.	Resource Person ID	
13.	Admin	4,5,7
14.	Admin ID	
15.	Super Admin	1,4,5,7
16.	Super Admin ID	
17.	Verification	3
18.	Activity	3
19.	Event ID	
20.	Event Code	2
21.	Event Creation	3
22.	Information	2
23.	Event Name	
24.	Description	
25.	Location	6
26.	Schedule	
27.	Google Map API	1,7
28.	Web Server	1,7

29.	Authentication	1,3,5
30.	System	2,4,7
31.	OAuth	1
32.	Information Management	
33.	Profile	2,7
34.	Google Account	
35.	User ID	
36.	User Name	
37.	Address	
38.	Contact No	
39.	Email	
40.	Occupation	
41.	Date of Birth	
42.	Nationality	
43.	Contact Saving	3
44.	Phonebook	2
45.	Status	
46.	Event Info Management	
47.	Lifetime	
48.	Time Stamp	
49.	Database	1,2,4,7
50.	Authority	5
51.	My Event List	
52.	File Management	3
53.	Materials	2
54.	Slides	2
55.	Pdf	2
56.	Photos	2

57.	Photo Gallery	2,7
58.	Files	2,3,7
59.	File ID	
60.	File name	
61.	File type	
62.	File Upload Time	
63.	File Content	
64.	Resources	
65.	Notification	2,3,7
66.	Notices	2
67.	Reminder	3
68.	Notification ID	
69.	Notification type	
70.	Notification Content	
71.	Plain text	
72.	Communication	3
73.	Forum	2,3,7
74.	Forum ID	
75.	Forum Type	
76.	Post	2
77.	Post ID	
78.	Post Subject	
79.	Post Content	
80.	Date	
81.	Question	
82.	Occupation	
83.	Answer content	
84.	Feedback	

85.	Storage	7
86.	Data	2
87.	Conversation	3
88.	Message	2
89.	Things To Do	1,3,7
90.	Suggestion	
91.	Restaurant	6
92.	Resort	6
93.	Event Spot	6
94.	Direction	

6.3 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

No	Class	Selection Criteria
1	Event	1-3,5
2	User	1-5
3	Admin	1-5
4	Participant	1-5
5	Resource Person	1-5
6	Super Admin	1-5
7	Forum	2-6
8	Conversation	2

9	Notification	3,4,5,6
10	System	1
11	Server	1
12	Google Map API	2
13	Authentication	3
14	OAuth	2,3
15	Profile	1
16	Files	1-6
17	Things To Do	2
18	Database	1

6.4 Associate Noun and Verb Identification

The noun and the verbs associated with the potential classes are identified to find out the attributes and methods of every class.

No	Class Name	Noun	Verb
1	Event	eventID, event code, event name, location, start date, end date, type, status	Creating event, updating event information, setting location. Adding user
2	User	userID, name, email, nationality, contact no, age, occupation	Updating profile, sign in, sign out
3	Admin	adminID, name, email	Updating profile, sign in, sign out, updating event info, invite resource person
4	Participant	Participant ID, name, email, nationality, contact no, age, occupation	Updating profile, sign in, sign out, viewing other user details
5	Super Admin	adminID, name, email	Creating event, ban admin, disable event
6	Resource Person	userID, name, email, nationality, contact no, age, occupation, Institution	Updating profile, sign in, sign out, viewing other user details
7	Notification	Notification ID, notification	Generate notification,

		content, date, time	send notification, remove notification
8	Forum	Forum ID, forum Post, date, time, subject	Creating forum, add post, edit post, remove post
9	System	Database, event, notification	Creating database, deleting database, accessing database, disconnecting database, creating temporary files, authentication, showing interface, showing files, locking system
10	Server	Database, port, host	Connect to api, disconnect, upload file
11	Database	User, Event, File, Database name	Insert, update, delete, checking connection, search
12	Conversation	Message , timestamp	Send message, receive message
13	Google Map API	Longitude, latitude, distance, direction	Get direction, show map, get longitude, get latitude
14	Files	File ID, File Name, File type, upload date	Upload photo, download photo, upload resource file, download resource file, remove file
15	Profile	Name, age, occupation	Update information
16	Things To Do	Place, longitude, latitude	Get suggestions, view suggestions
17	Authentication	User, Database	Sign in, sign out
18	OAuth	API, key	Generate key, get profile

6.5 Finalized Classes and Class Cards:

To identify final classes we need to first check that if there can be any hierarchies or merges. These are given below

1. Administrator and Super Admin can be merged and a new attribute can be introduced called “role”. The merged class name is set as “Admin”.

2. Participants and Resource person both class has same functionalities and attributes. So they are merged as ParticipantAndRP.

3. Both the previous two classes are subclass of User class.

4. The functionalities of Profile class can be done while authenticating user.

5. System class can handle server configuration.

Based on the above analysis, the following classes have been finalized for the project.

User	
Attributes	Methods
userID name email	signIn() signOut() getUserID()
Responsibilities	Collaborative Class
Authentication	Authentication

Admin	
Attributes	Methods
userID name email role	editProfile() createEvent() viewEventList() inviteResourcePerson() setNotification() uploadFiles() deleteFiles() banUser() assignNewAdmin() removeAdmin() manageForum() setEventLifeTime() updateEvent() setThingsToDoSuggestion()

Responsibilities	Collaborative Class
<p>Authentication</p> <p>Editing profile</p> <p>Creating event</p> <p>Editing event details</p> <p>Setting suggestions of important places near event location</p> <p>Managing discussion forum</p>	<p>Notification</p> <p>Files</p> <p>Event</p>

ParticipantAndRP	
Attributes	Methods
<p>userID</p> <p>name</p> <p>email</p> <p>address</p> <p>age</p> <p>nationality</p> <p>contactNumber</p> <p>institution</p> <p>userType</p> <p>occupation</p>	<p>joinEvent()</p> <p>viewEventInfo()</p> <p>viewParticipants()</p> <p>viewResourcePerson()</p> <p>saveContact()</p> <p>downloadFile()</p> <p>accessForum()</p> <p>viewForum()</p> <p>viewNotification()</p> <p>viewPhotoGallery()</p> <p>chat()</p> <p>accessThingsToDo()</p> <p>editProfile()</p> <p>checkIn()</p>
Responsibilities	Collaborative Class
<p>Authentication</p> <p>Joining event</p> <p>Viewing participant and resource person list</p> <p>Adding post to forum</p> <p>Viewing photo gallery</p> <p>Chat with others</p>	<p>Notification</p> <p>Files</p> <p>Event</p> <p>Conversation</p>

Authentication

Attributes	Methods
User System	signInWithGoogle() signOut() isPreviousUser() createProfile() modifyProfile()
Responsibilities	Collaborative Class
Creating new profile	User Database System
Checking profile information	

Event	
Attributes	Methods
eventID eventCode eventName locationLongitude locationLatitude startDate endDate eventType status	getEventDetails() updateEventDetails() setLocationInMap() getEventLocation addNewParticipant() addNewRP() getParticipantList() getRPLlist() attachForum() includeNotification()
Responsibilities	Collaborative Class
Setting up event details and location Keeping information of the participants and resource person Maintaining Forum	User Forum Notification System GoogleMapAPI

Notification	
Attributes	Methods
notificationID notificationContent Date Event	generateNotification() sendNotification() removeNotification() saveNotification()
Responsibilities	Collaborative Class
Managing notification system	Admin

Storing notification content	Event System
------------------------------	--------------

Forum	
Attributes	Methods
forumID forumPost publishTime Date	viewForum() mentionUser() addPost() editPost() replyToPost()
Responsibilities	Collaborative Class
Adding post to forum Displaying forum contents Managing forum contents	Event Database

System	
Attributes	Methods
serverIP port Database	connectServer() disconnectServer() configureDatabase() generateNotification() sendNotification() checkEventValidity() checkException() promptConfirmationMessage() renderActivity()
Responsibilities	Collaborative Class
Configuring database and server Rendering interface Checking exception and error Showing prompt message	Database Event Notification

Database

Attributes	Methods
databaseName hostName password sqlQuery	configure() insert() update() delete() search()
Responsibilities	Collaborative Class
Inserting elements Updating data Storing Data Searching Data	User Event File System

Files	
Attributes	Methods
fileID fileName fileType filePath uploadDate	uploadPhoto() downloadPhoto() uploadResourceFile() downloadResourceFile() removeFile()
Responsibilities	Collaborative Class
Managing file management system Removing files	Database User Event

Conversation	
Attributes	Methods
chatID messageID messageContent timestamp	sendMessage() receiveMessage() displayMessage() getMessageTime() setMessageTime() deleteMessage()
Responsibilities	Collaborative Class

Sending message to other user	ParticipantAndRP Database
Receiving message from other user	
Removing message content	

GoogleMapAPI	
Attributes	Methods
longitude latitude distance apiKey	setUpConnection() getDirection() getDistance() getLongitude() setLongitude() getLatitude() setLatitude() showMap()
Responsibilities	Collaborative Class
Configuring google Map API	Event ThinsToDo

ThingsTo Do	
Attributes	Methods
locationList distance Map	showSuggestion() showMap() showDirection()
Responsibilities	Collaborative Class
Suggesting Important places Showing place location on google Map	GoogleMapAPI

6.6 UML Class Diagram

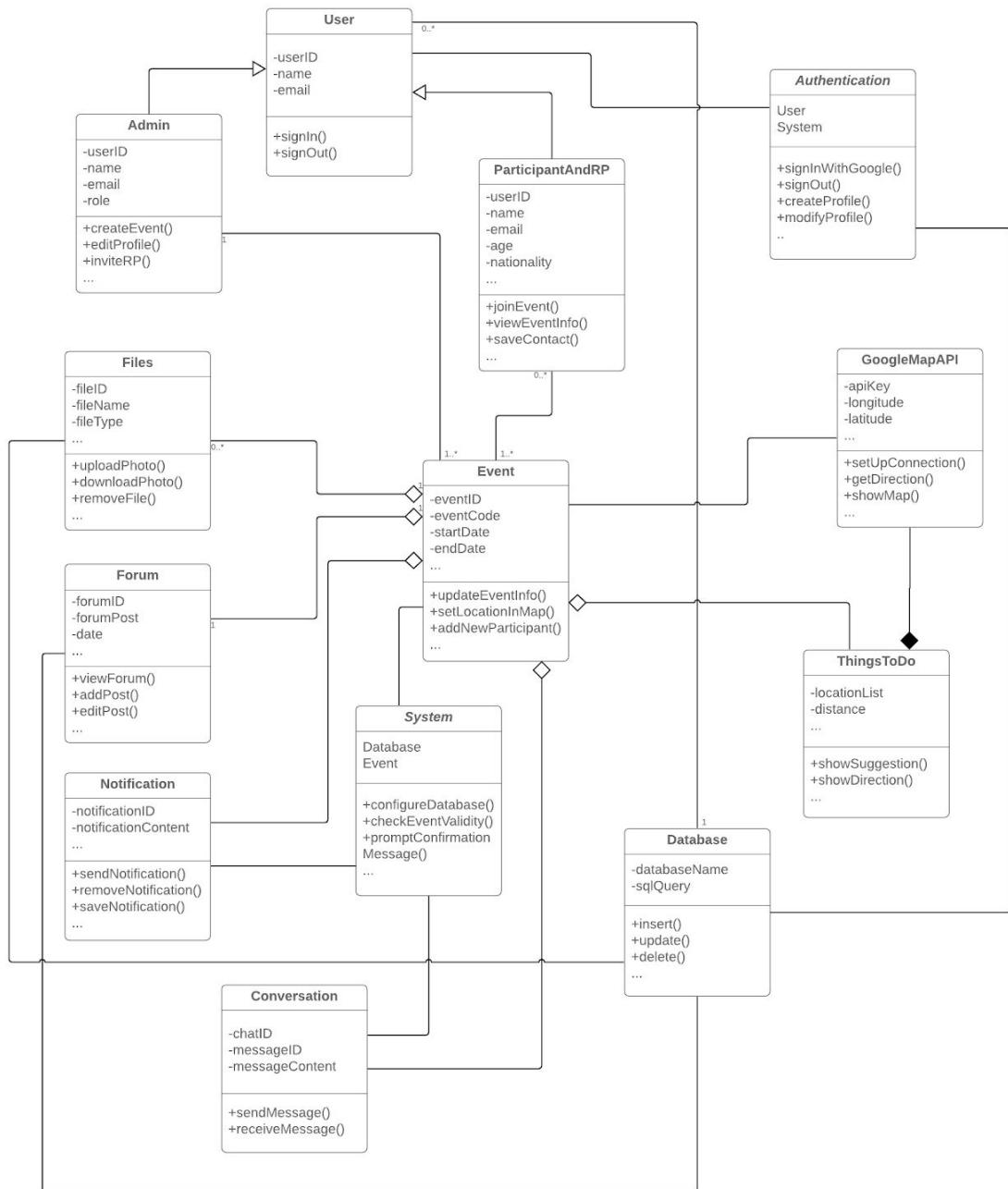


Figure: UML Class Diagram

Chapter 7

FLOW ORIENTED MODELING

7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today. It provides additional insight into system requirements and flow

7.2 Data Flow Diagram (DFD)

The DFD takes an input-process-output view of a system. In the figures, data objects are represented by labeled arrows and transformations are represented by circles.

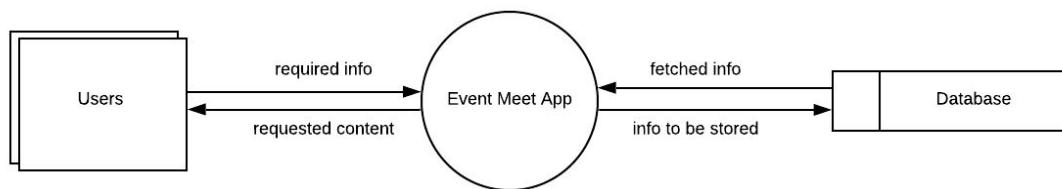


Figure. Level-0 for Event Meet App

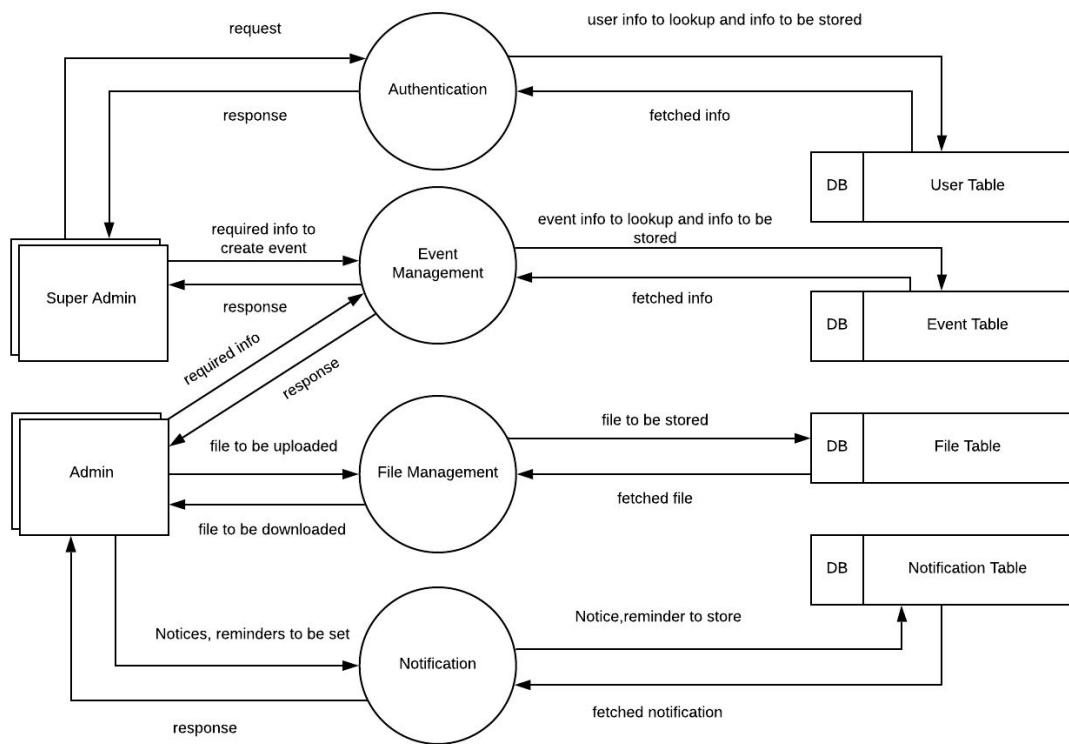


Figure. Level-1.1 for Event Meet App

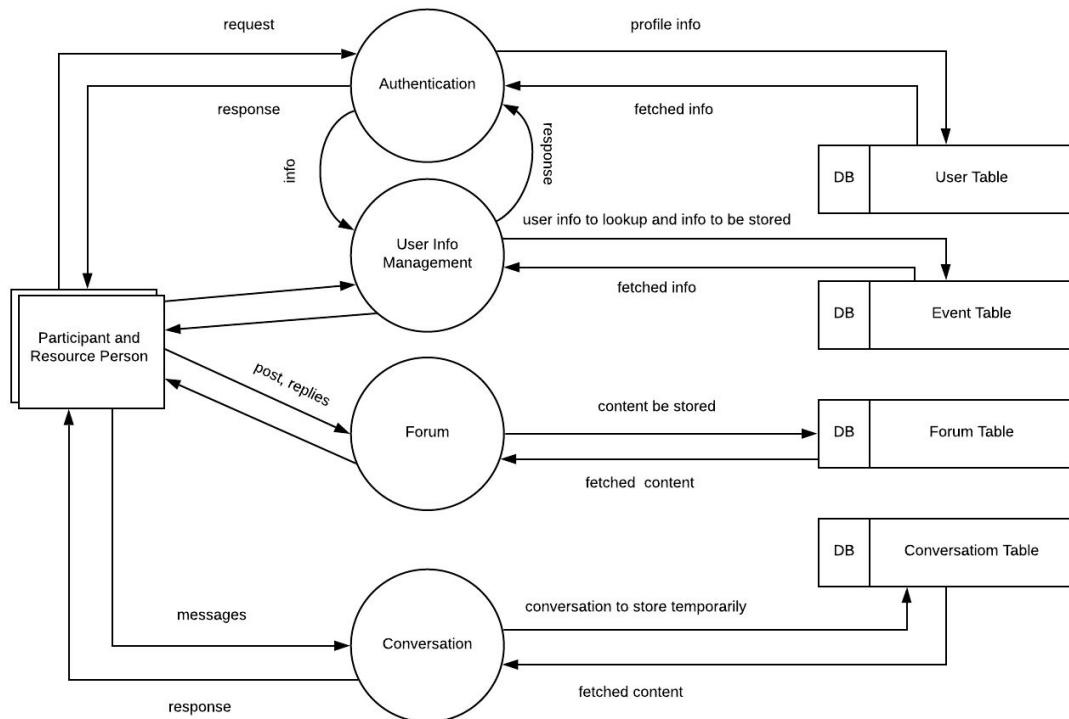


Figure. Level 1.2 for Event Meet App

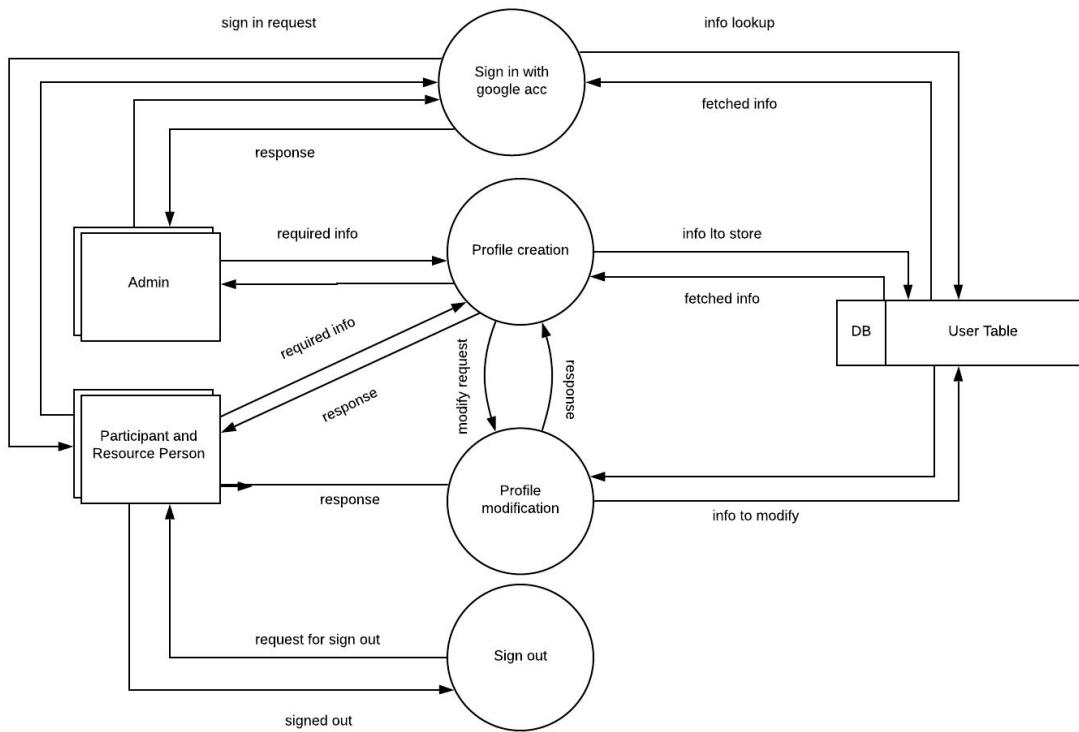


Figure. Level-2.1 for Authentication of Event Meet App

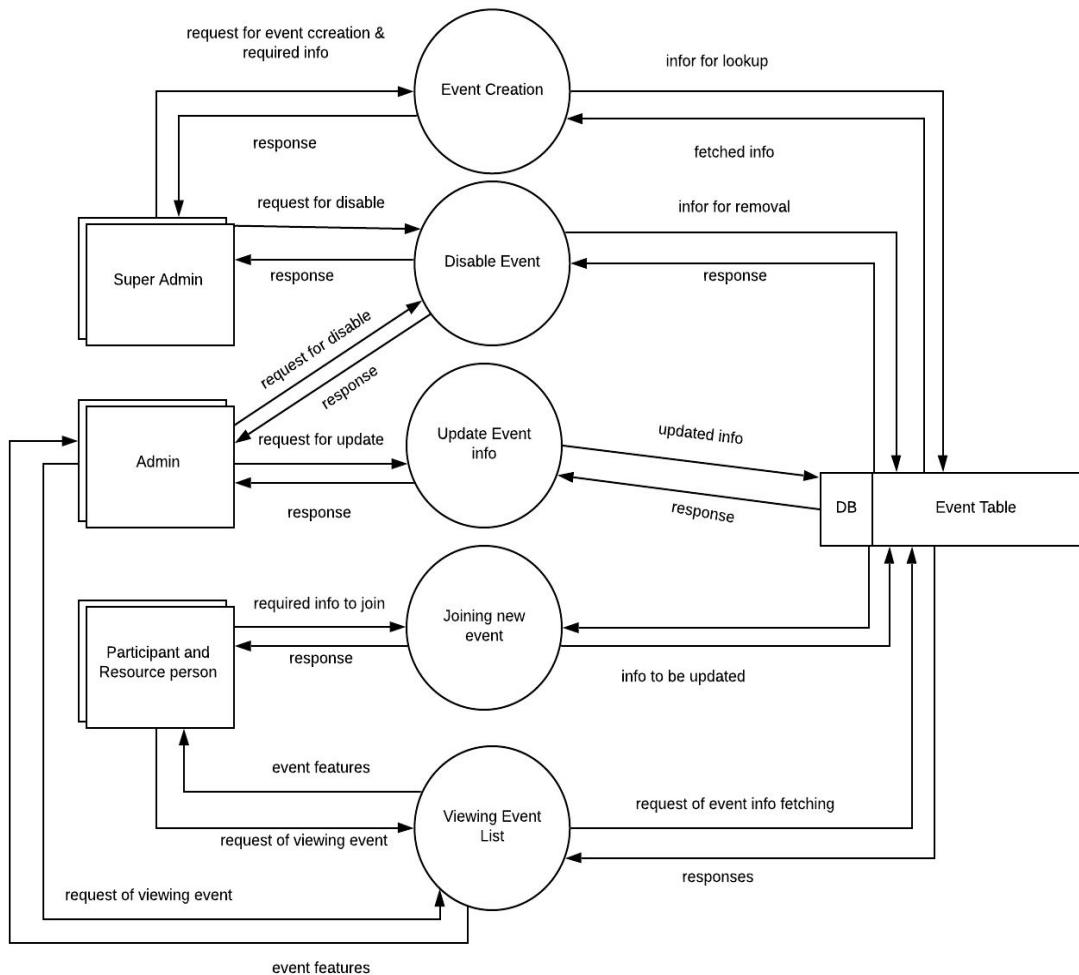


Figure. Level-2.2 for Event Management of Event Meet App

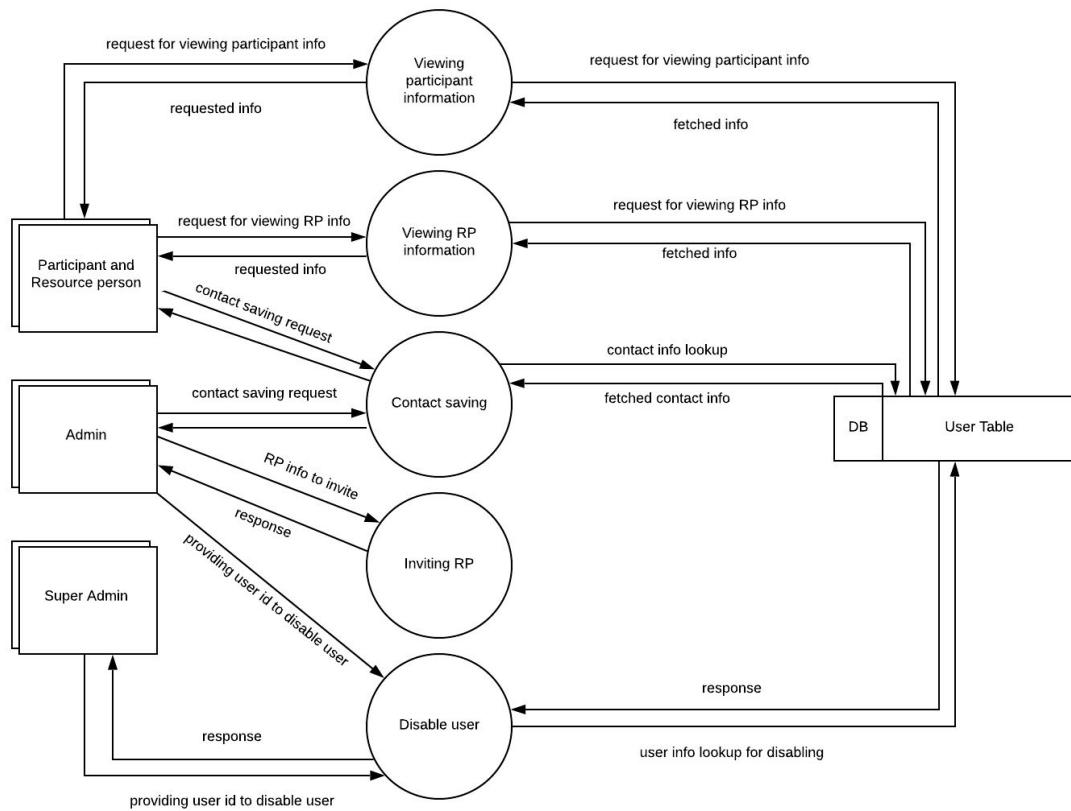


Figure. Level-2.3 for User Information Management of Event Meet App

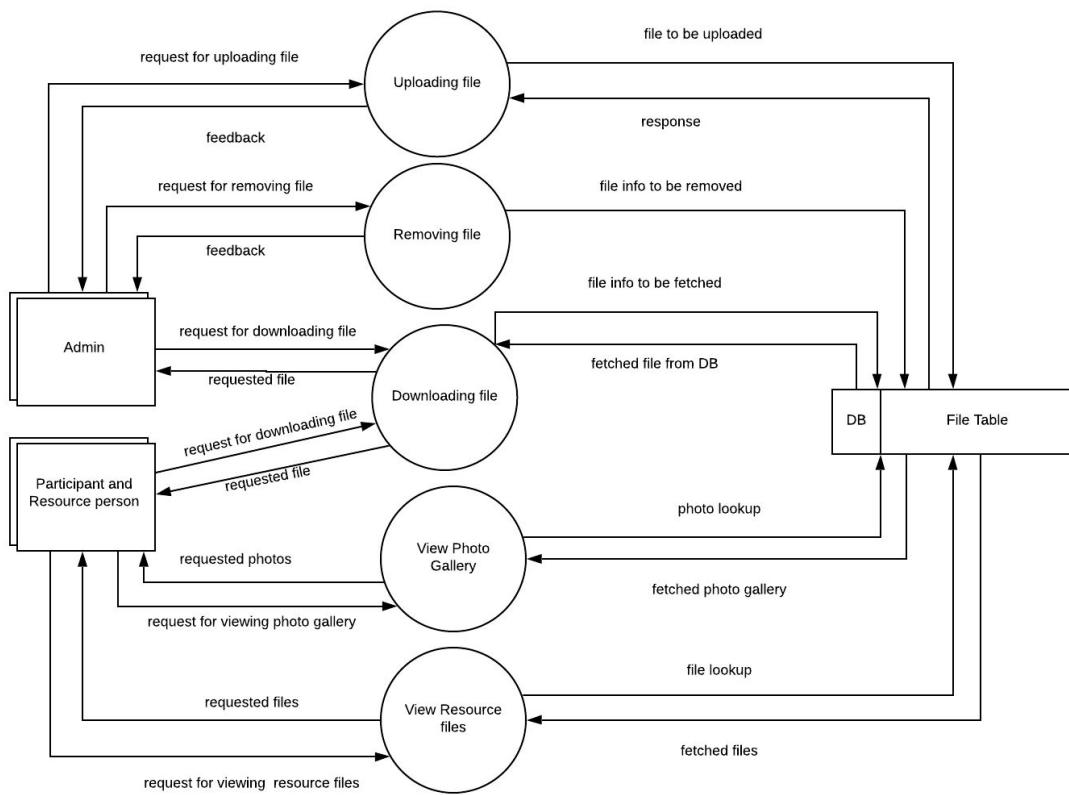


Figure. Level-2.4 for File Management System of Event Meet App

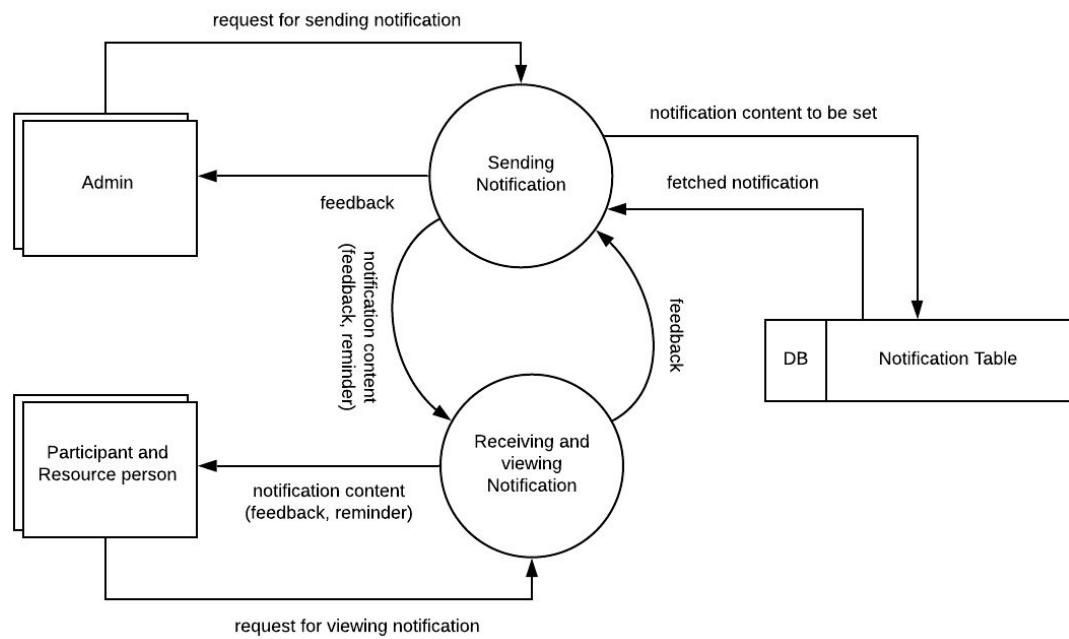


Figure. Level 2.5 for Notification of Event Meet App

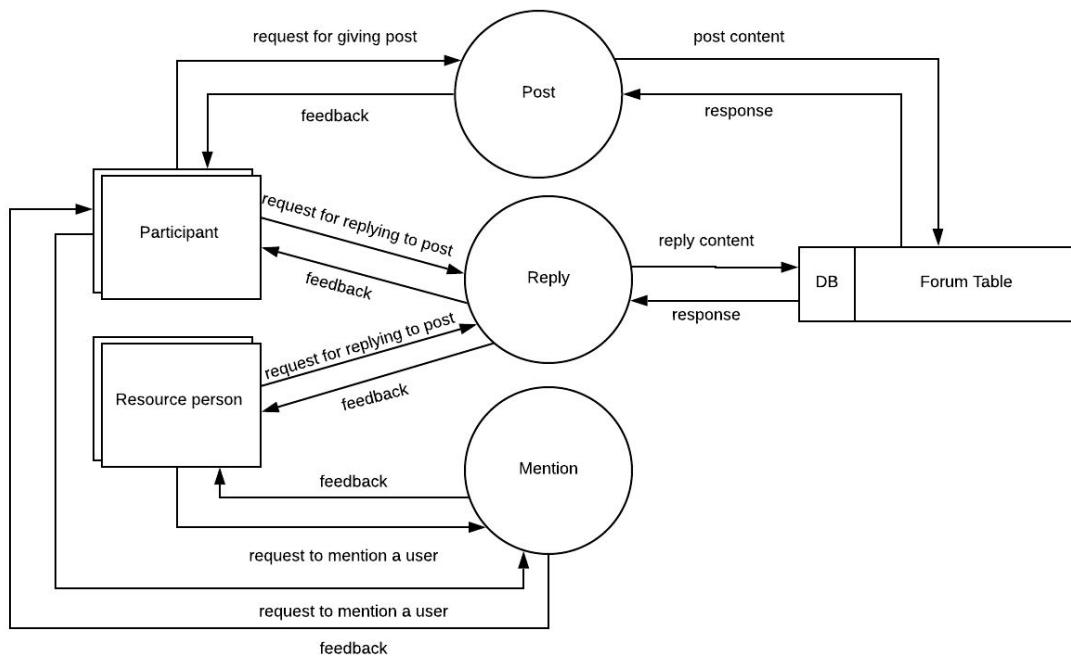


Figure. Level-2.6 for Forum of Event Meet App

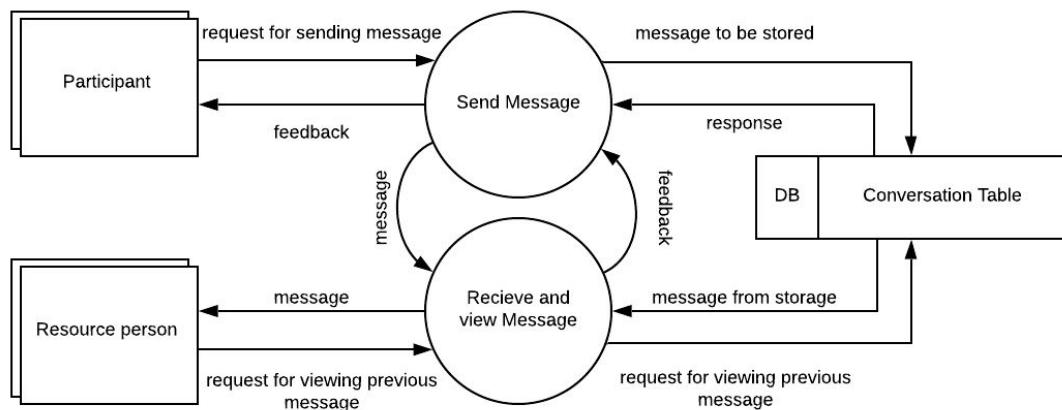


Figure. Level-2.7 for Conversation of Event Meet App

Chapter 8

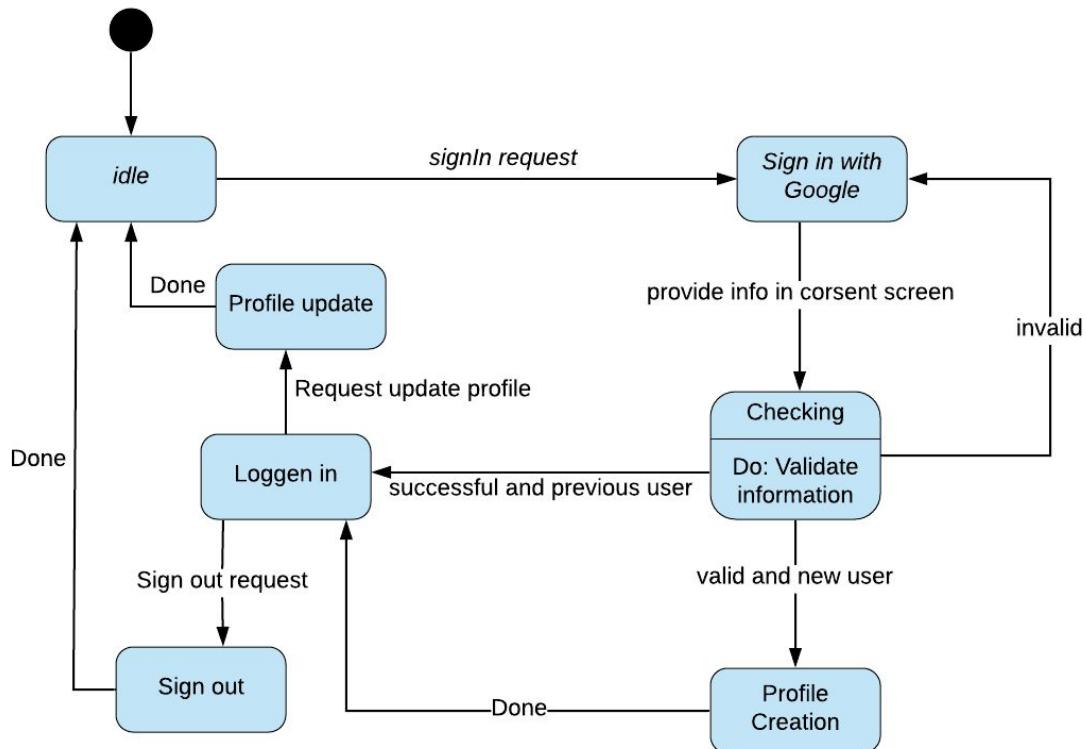
BEHAVIORAL MODELING

The behavioral model indicates how software will respond to external events.

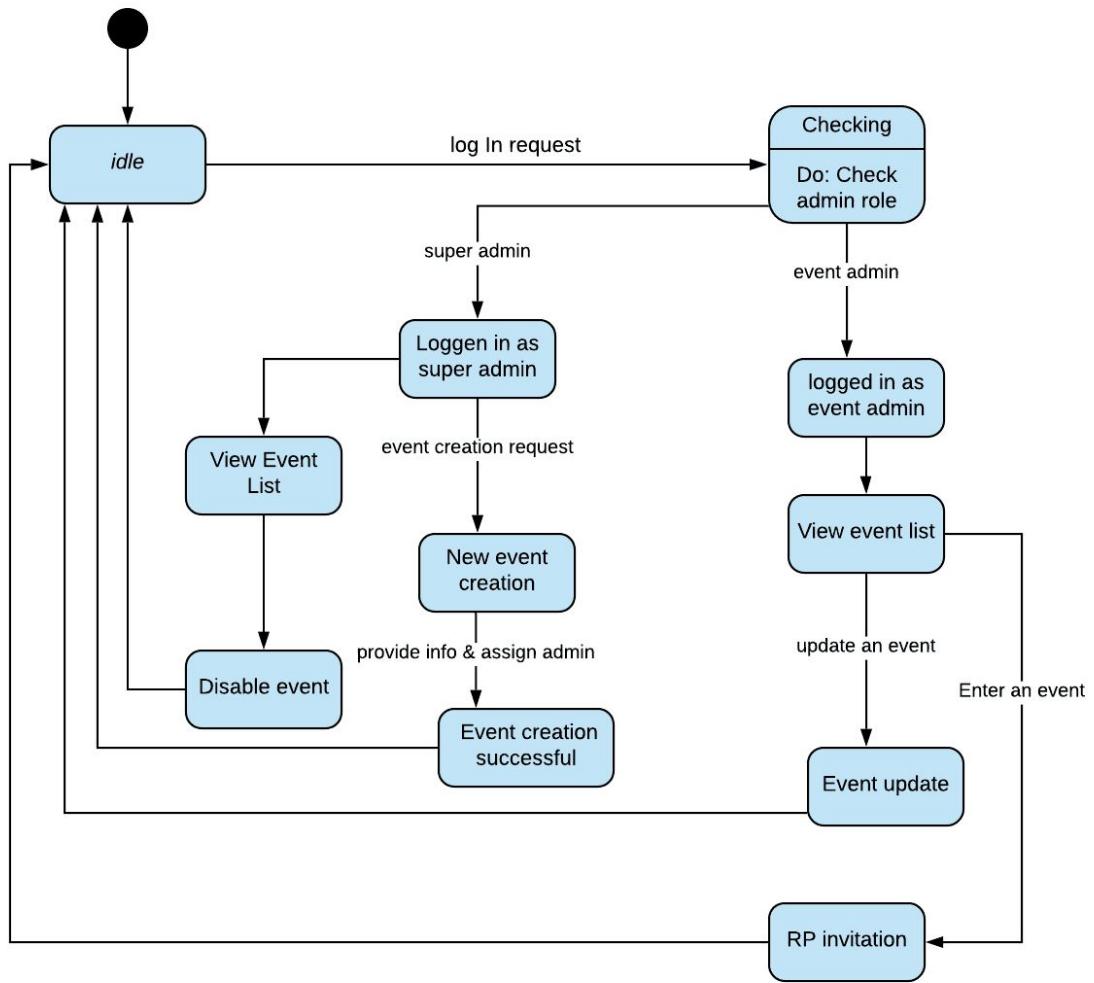
8.1 State Diagram

State diagram represents active states for each class the events (triggers).

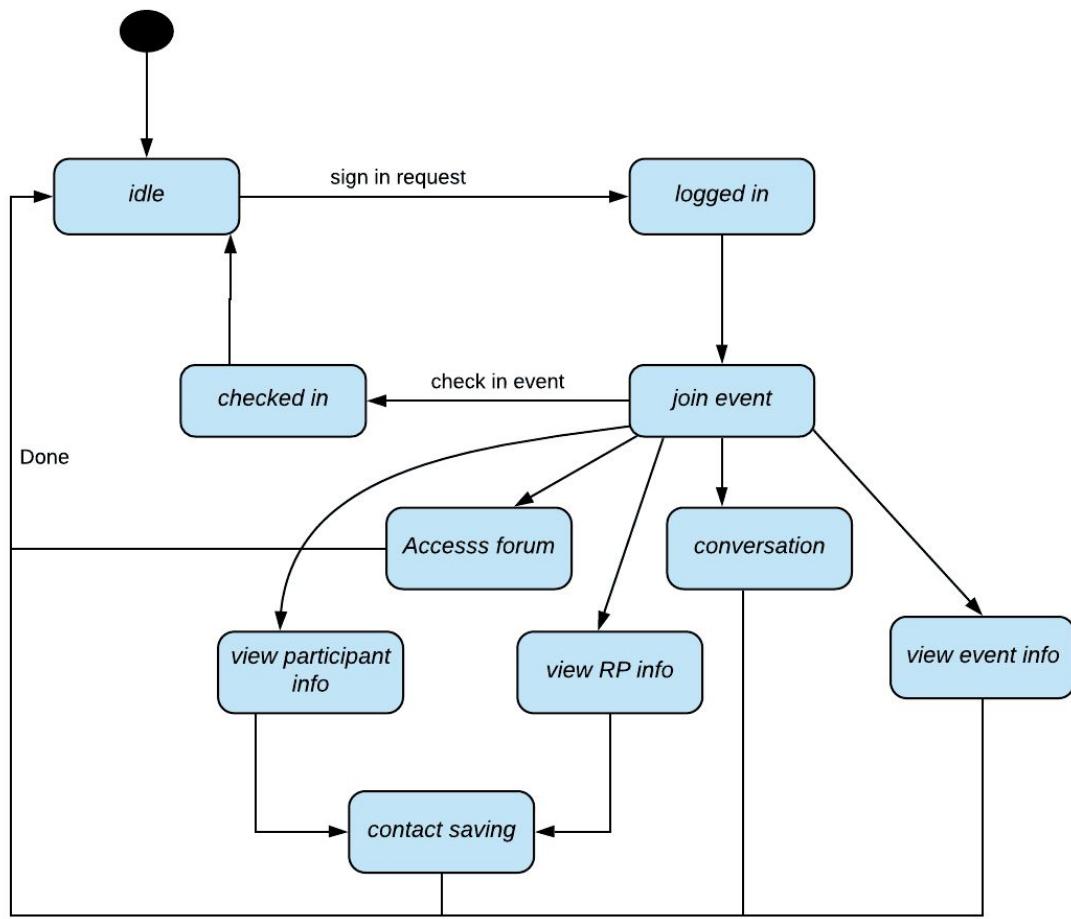
8.1.1 Authentication Class:



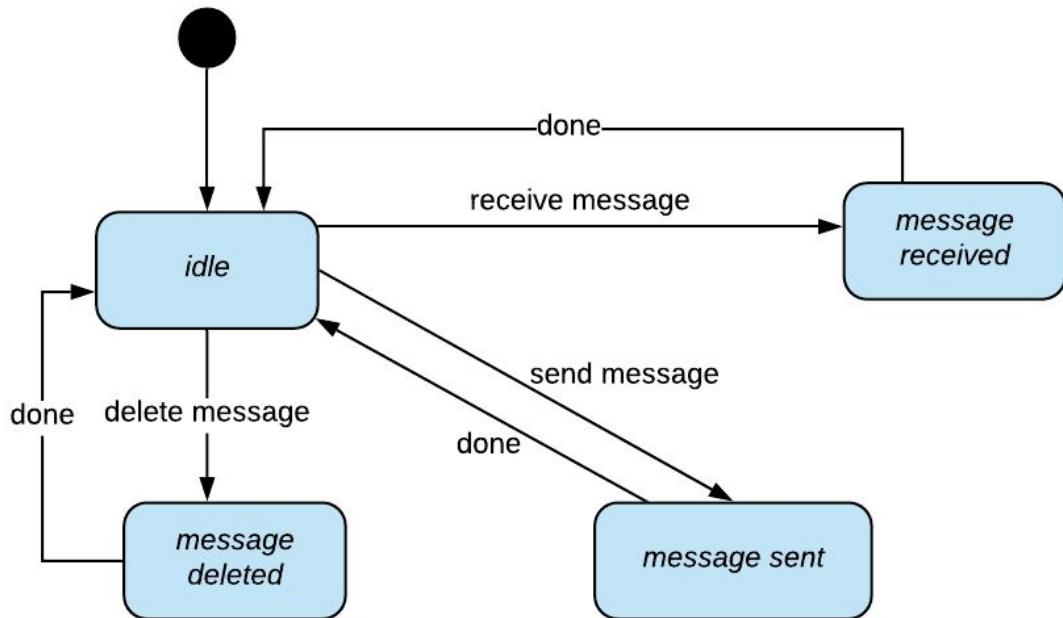
8.1.2 Admin Class:



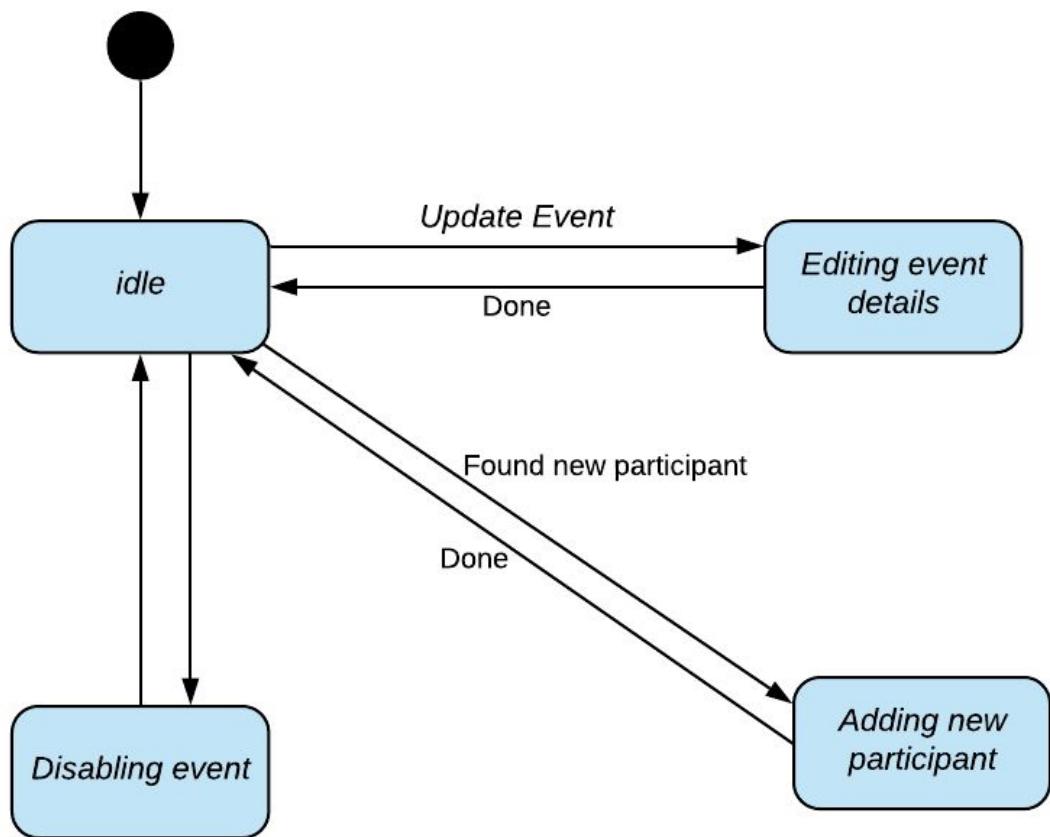
8.1.3 ParticipantAndRP Class:



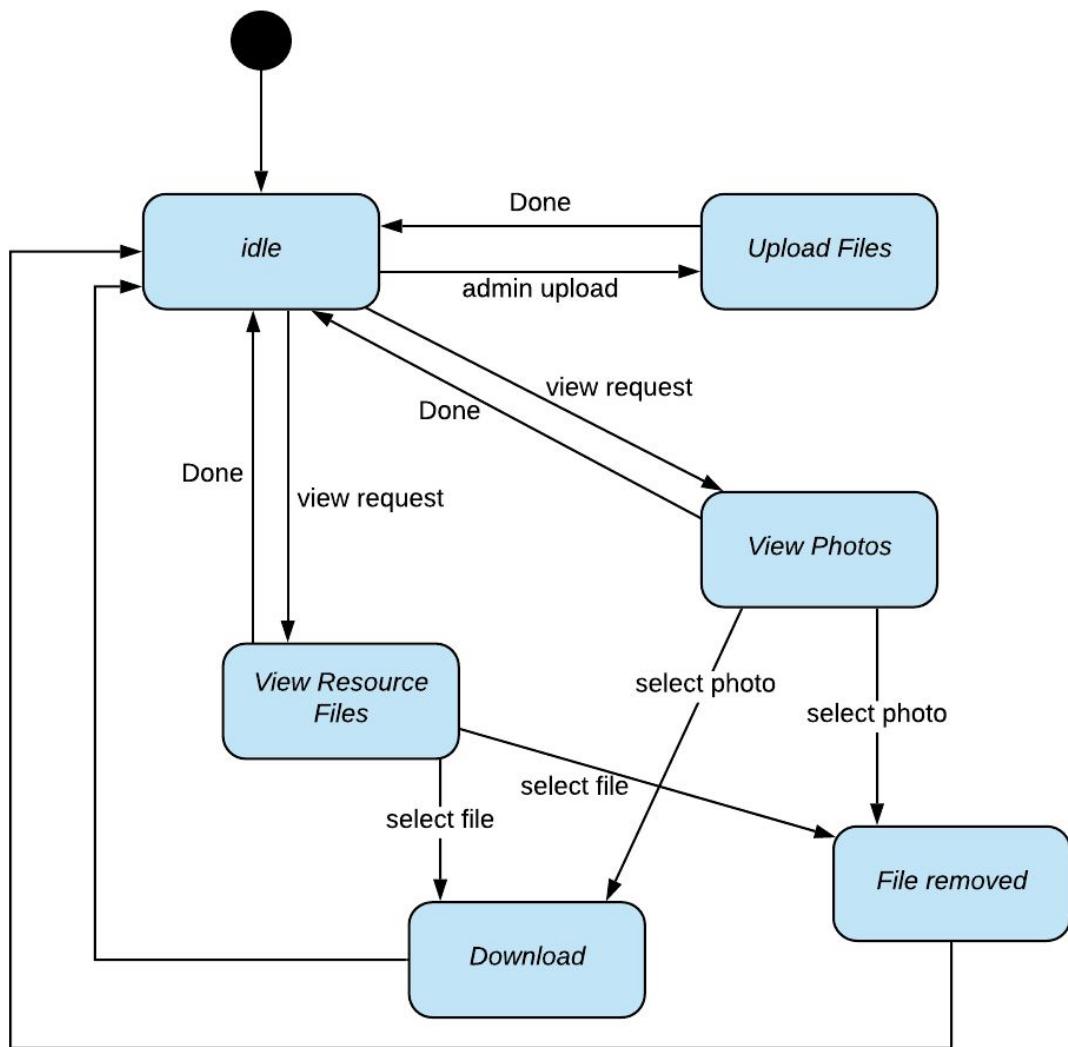
8.1.4 Conversation Class:



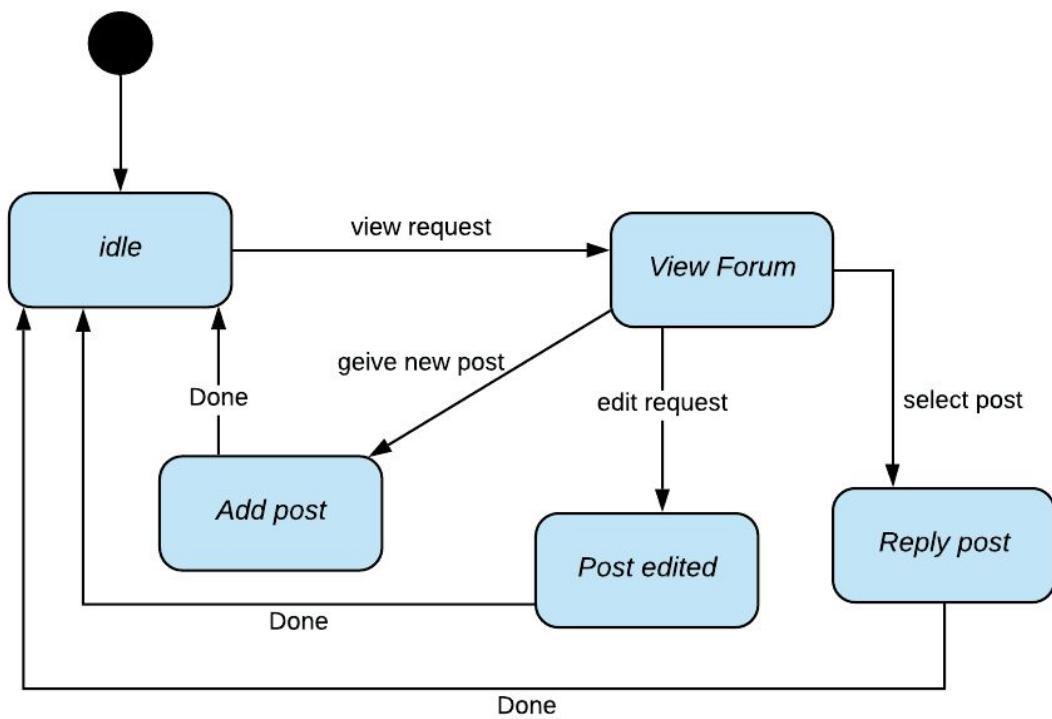
8.1.5 Event Class:



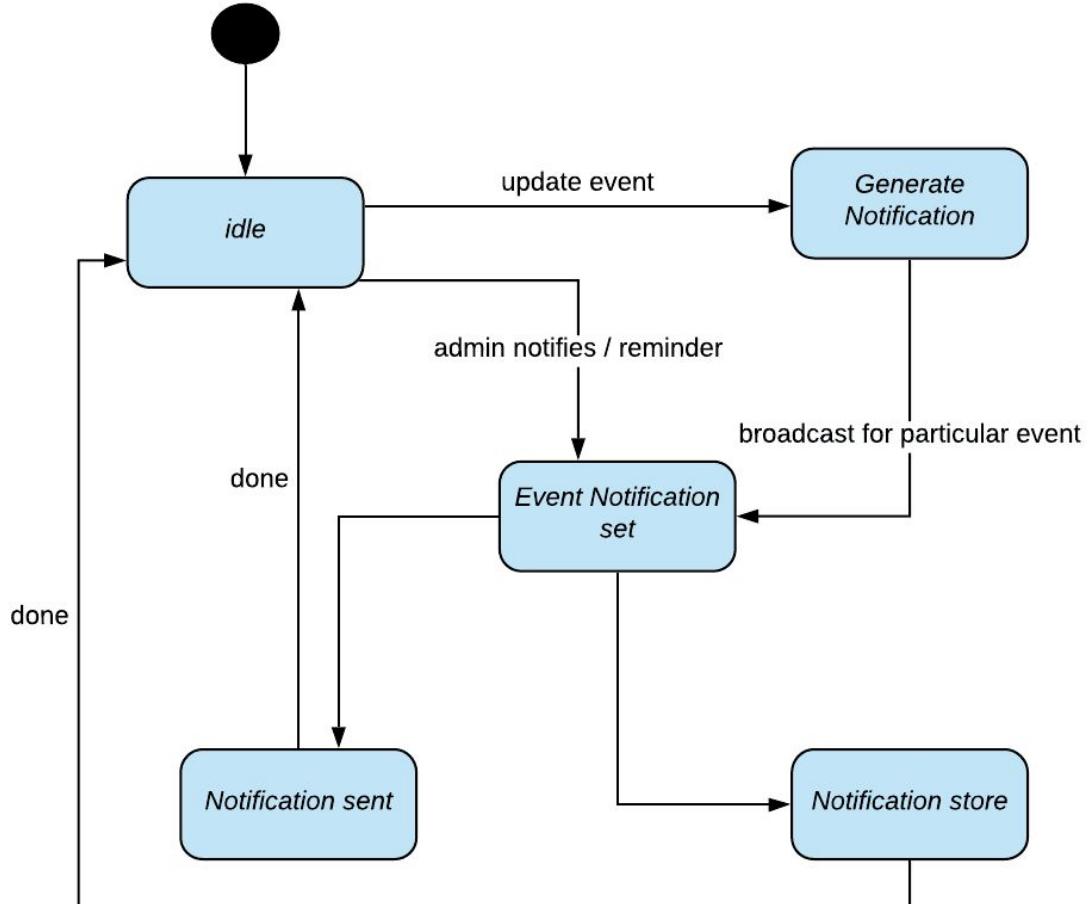
8.1.6 Files Class:



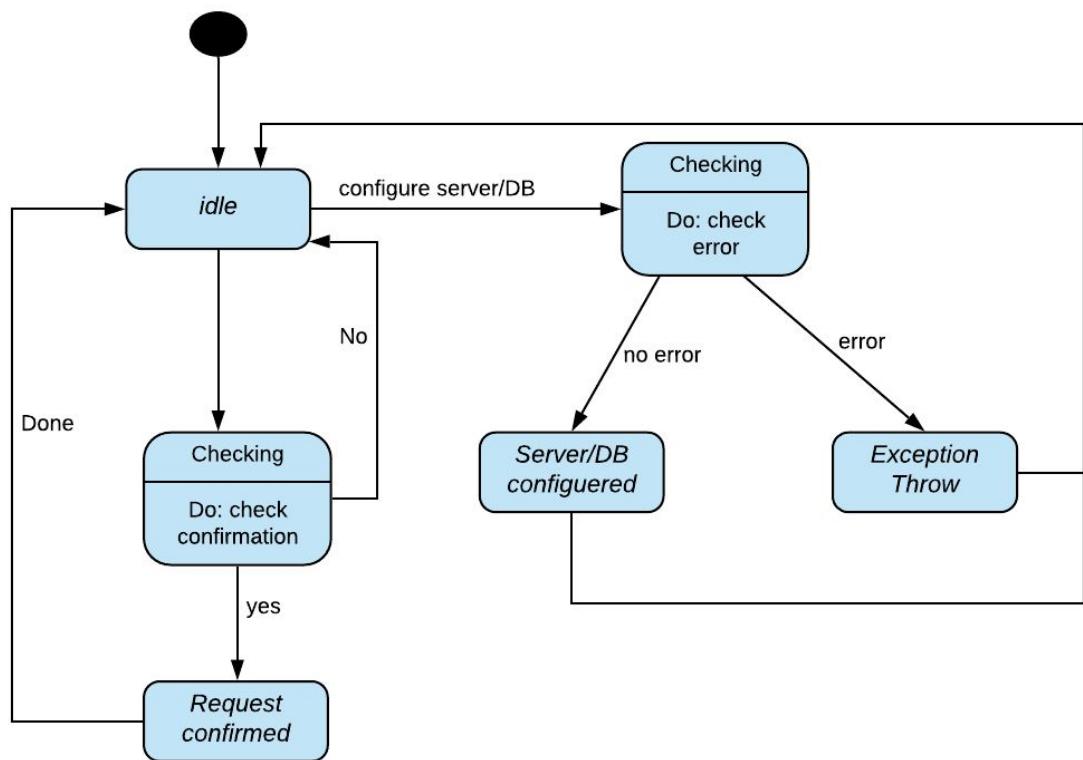
8.1.7 Forum Class:



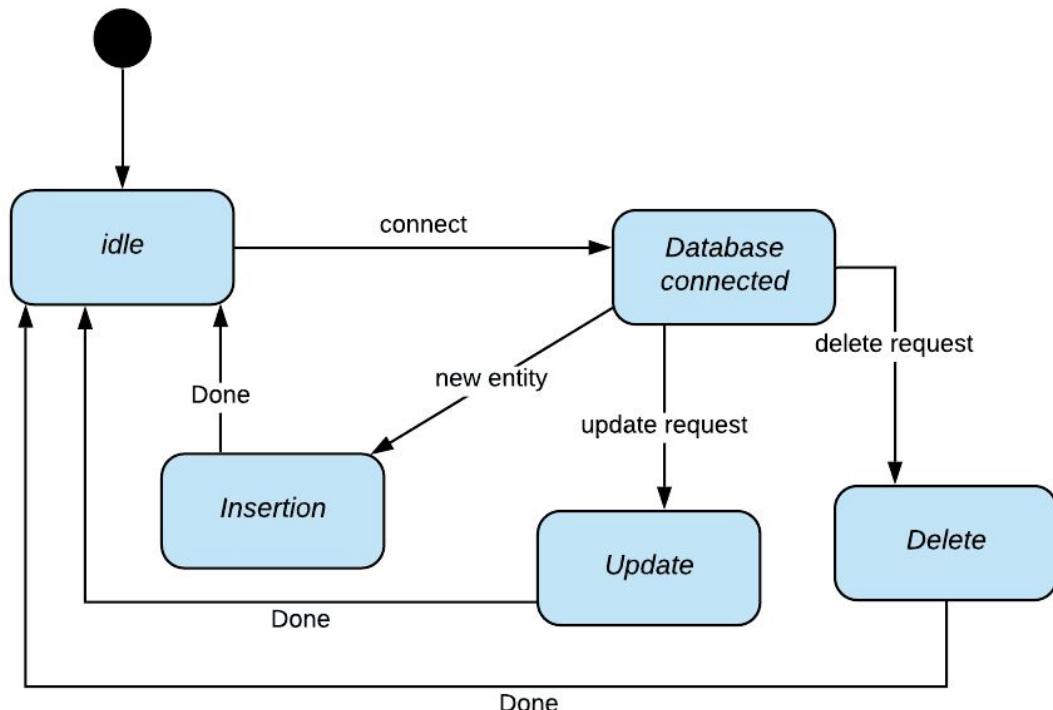
8.1.8 Notification Class:



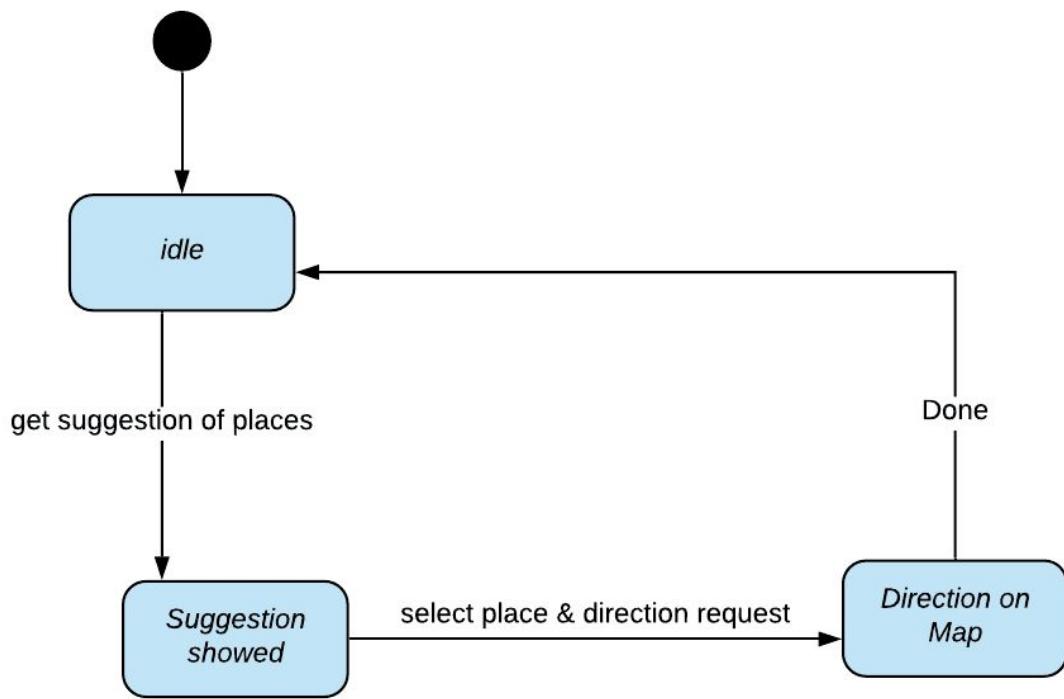
8.1.9 System Class:



8.1.10 Database Class:



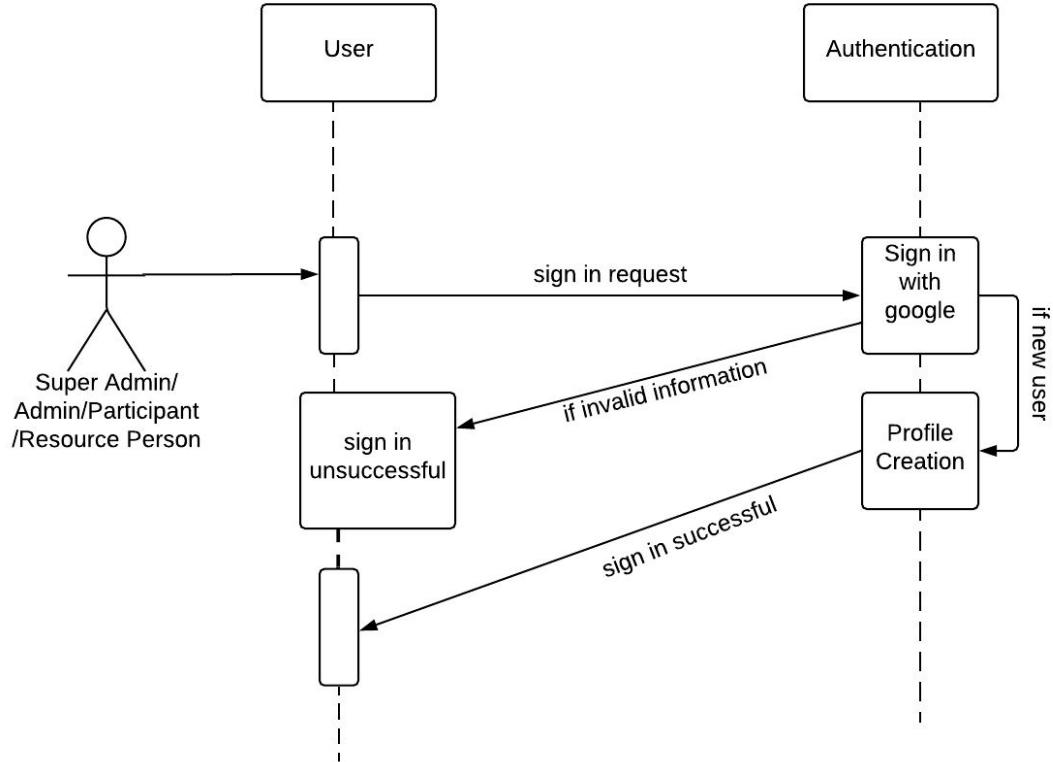
8.1.11 Things To Do Class:



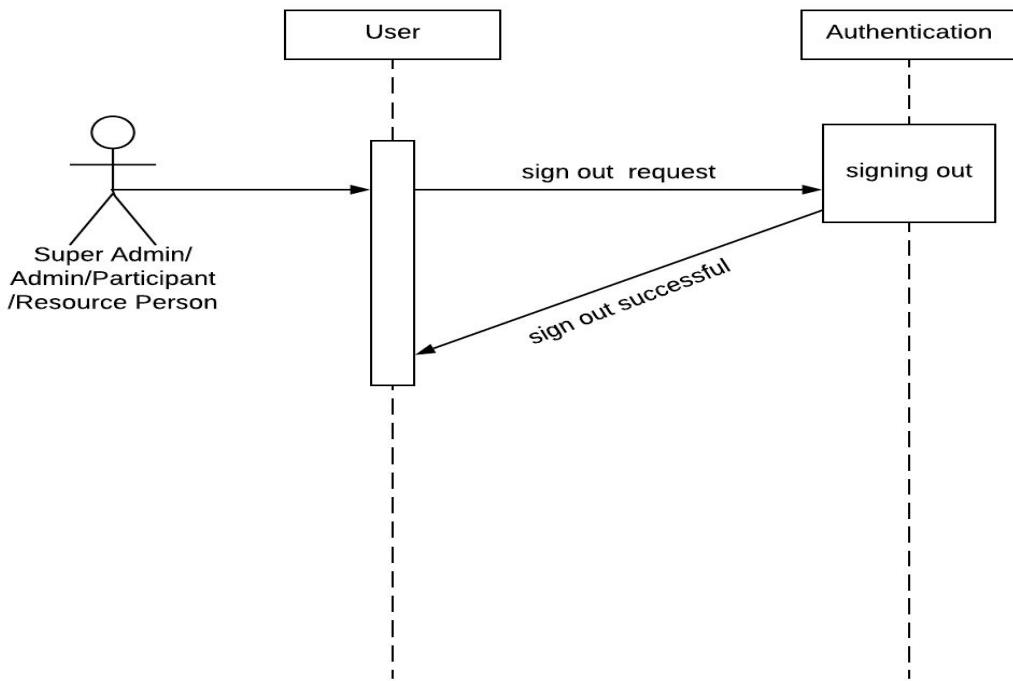
8.2 Sequence Diagram:

Sequence diagram indicates how events cause transitions from object to object.

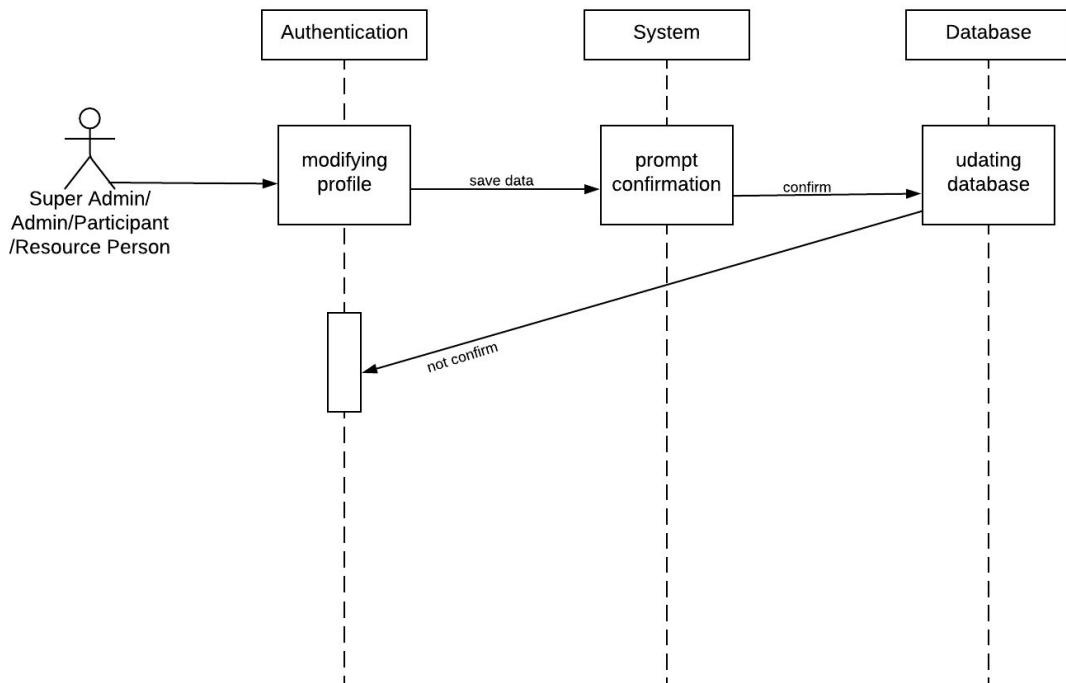
8.2.1 Sign In



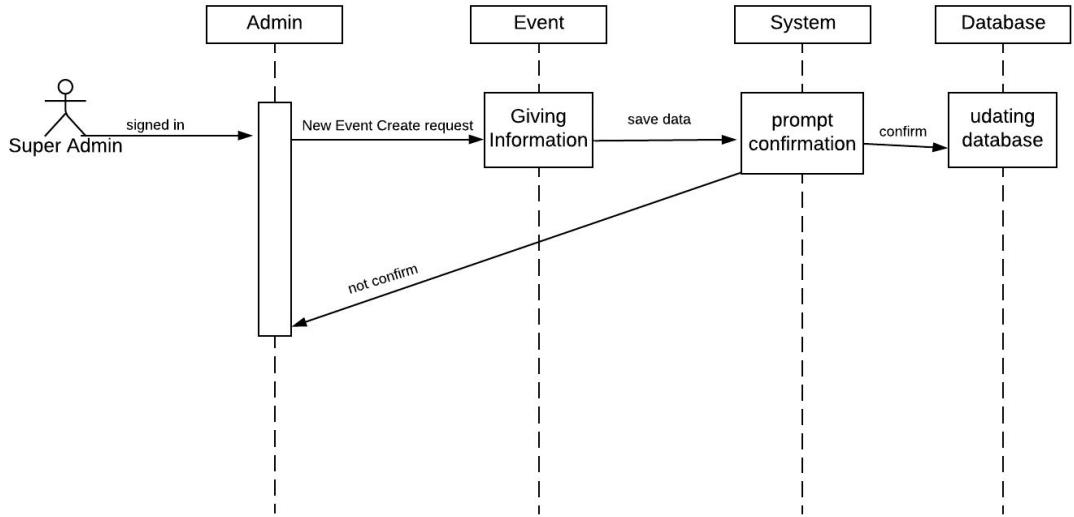
8.2.2 Sign Out



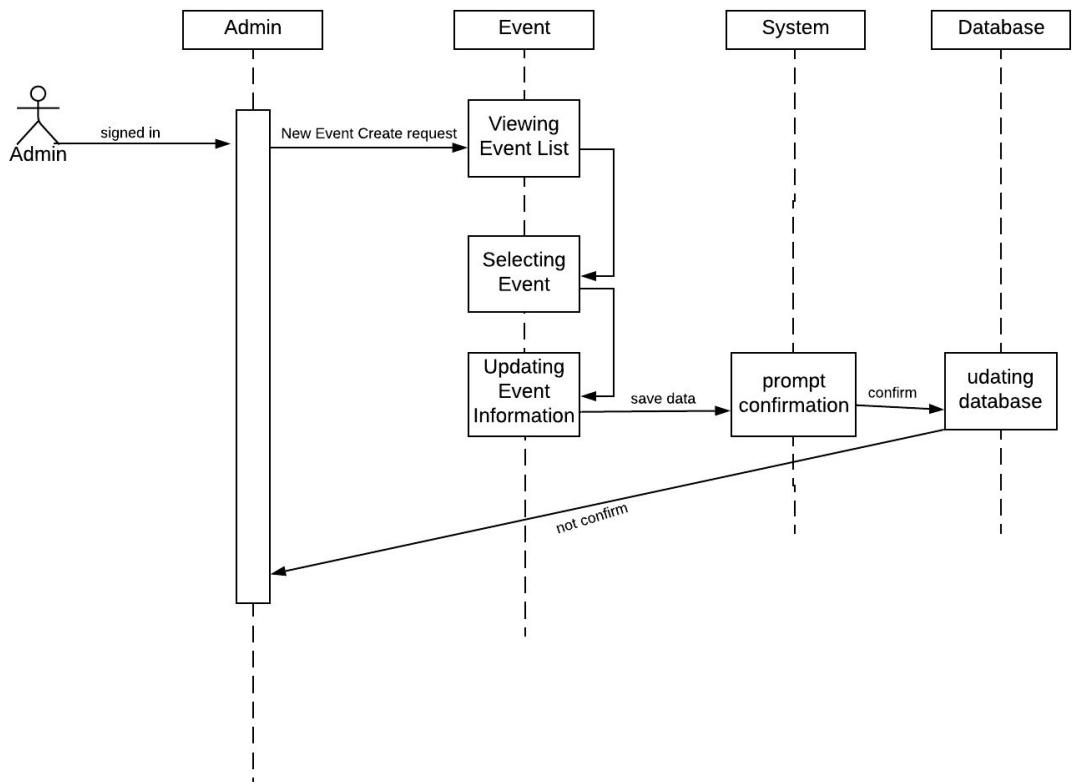
8.2.3 Profile Modification



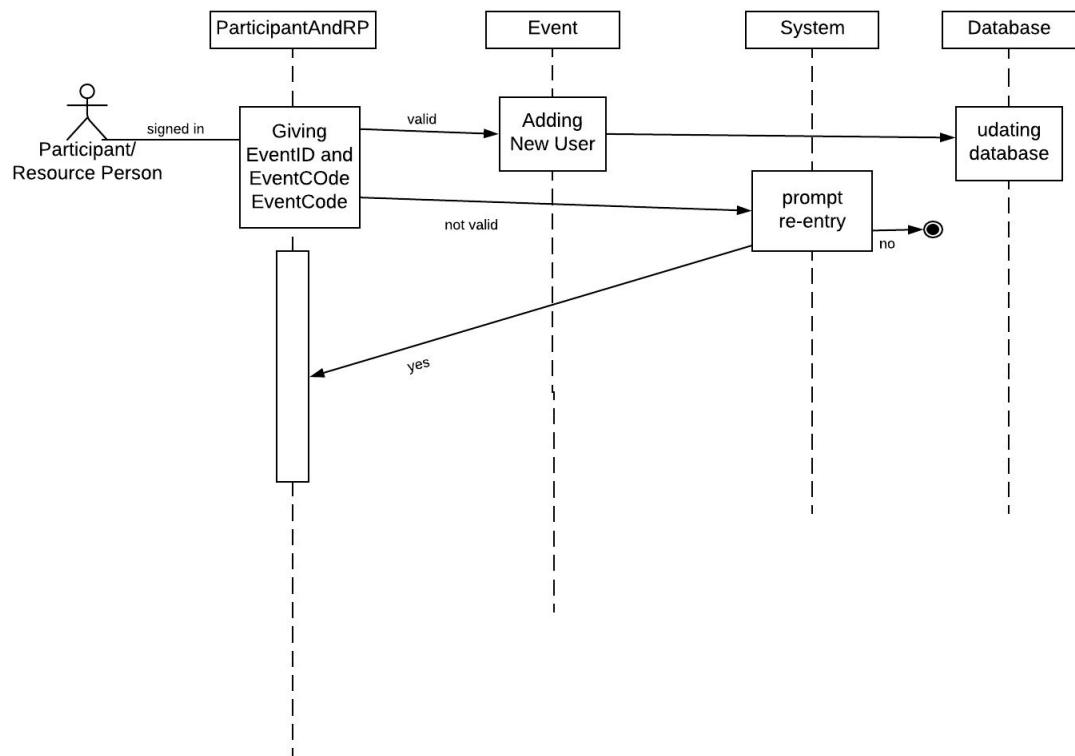
8.2.4 Event Creation



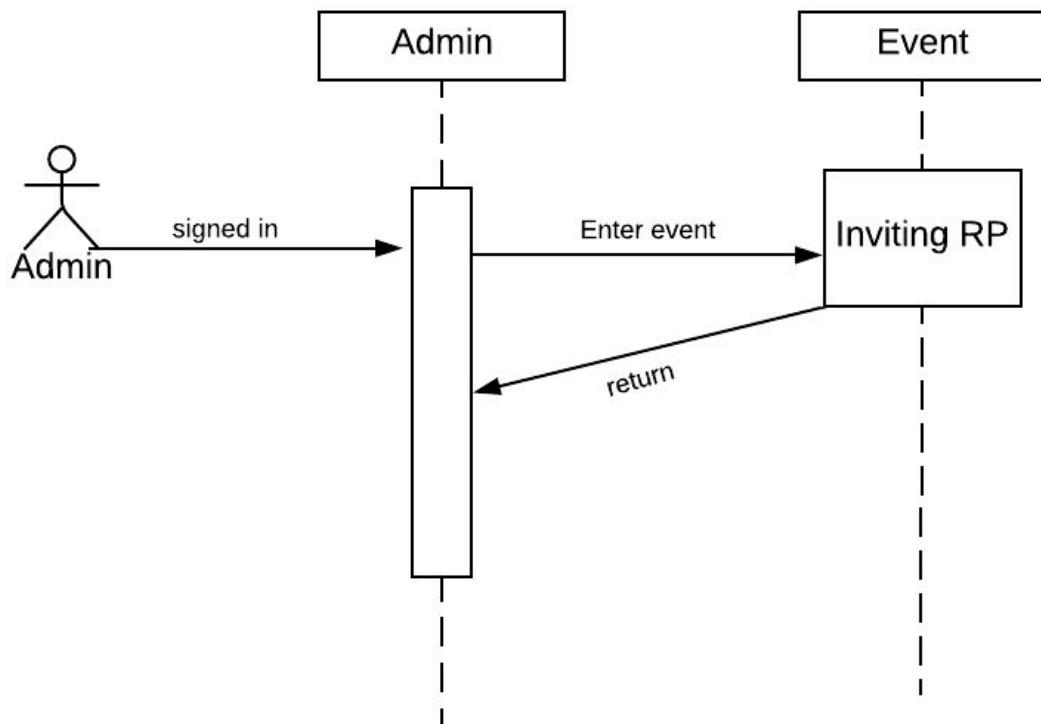
8.2.5 Updating Event Information



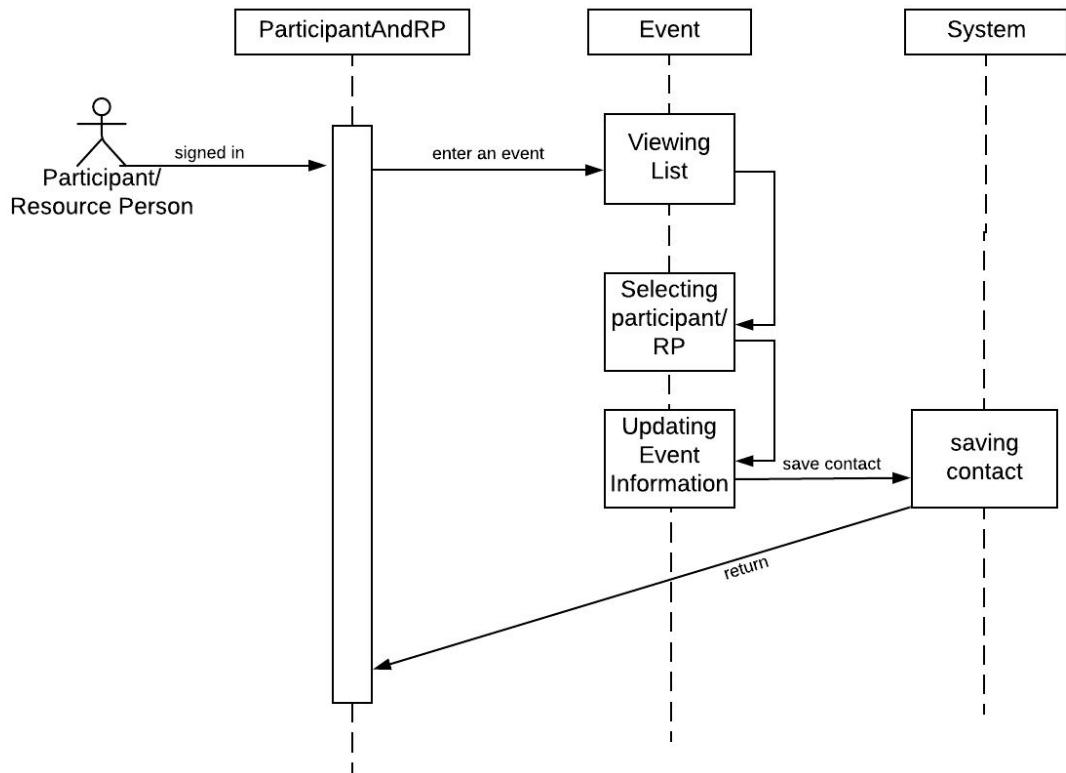
8.2.6 Joining New Event



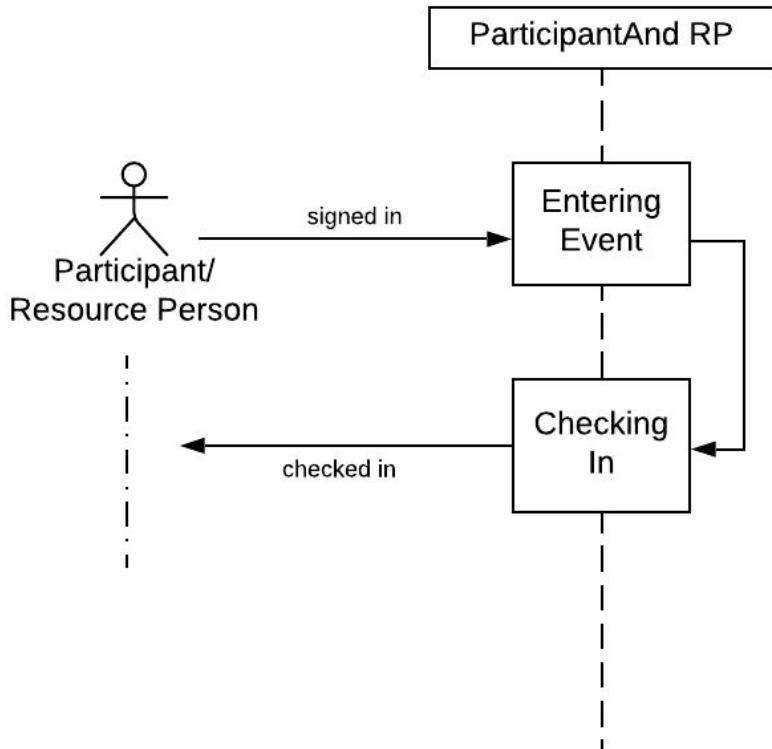
8.2.7 Inviting Resource Person



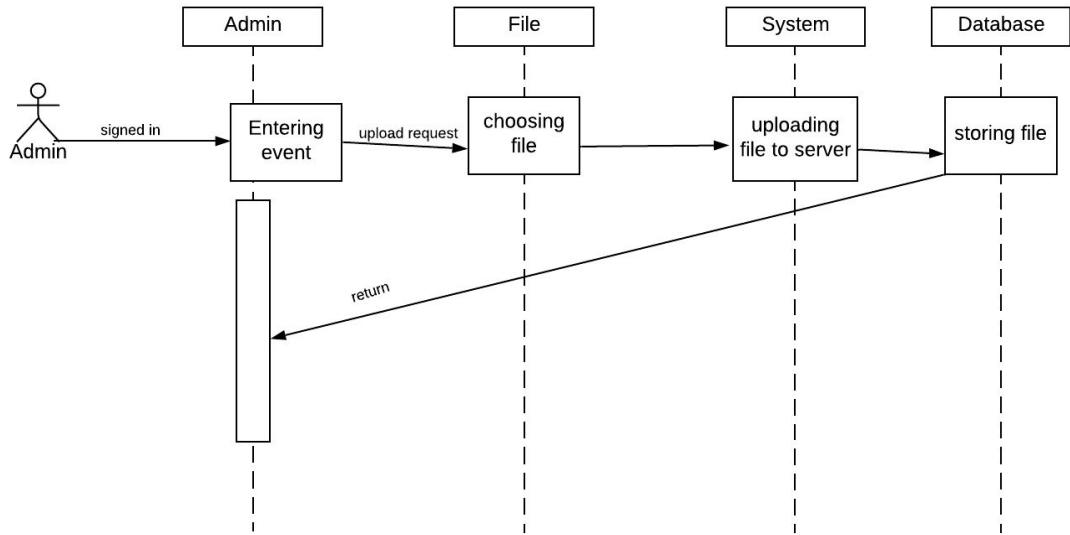
8.2.8 Contact Saving



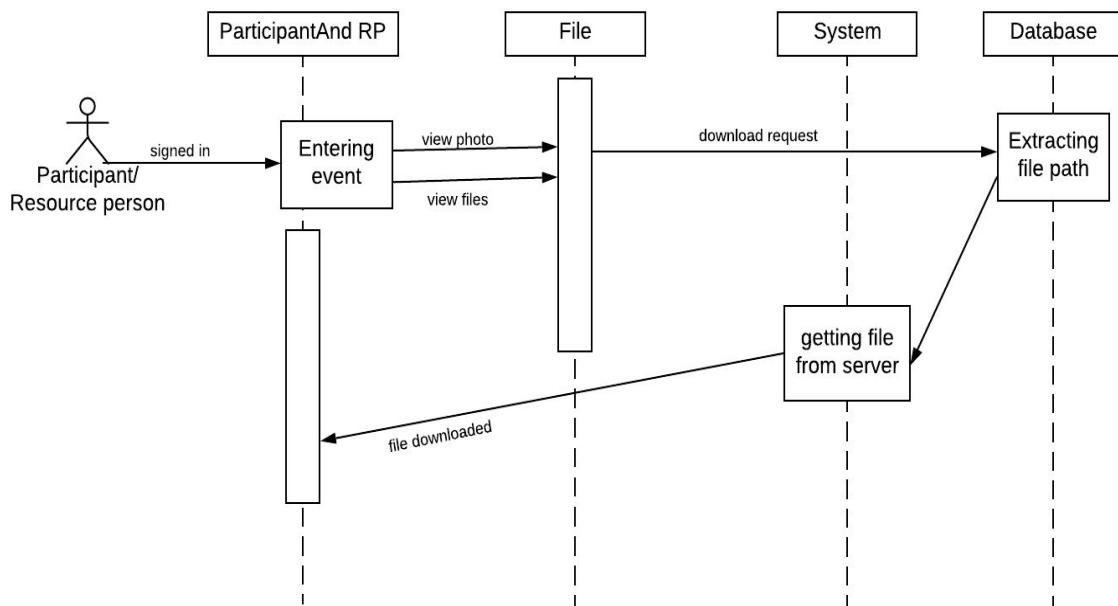
8.2.9 Check in



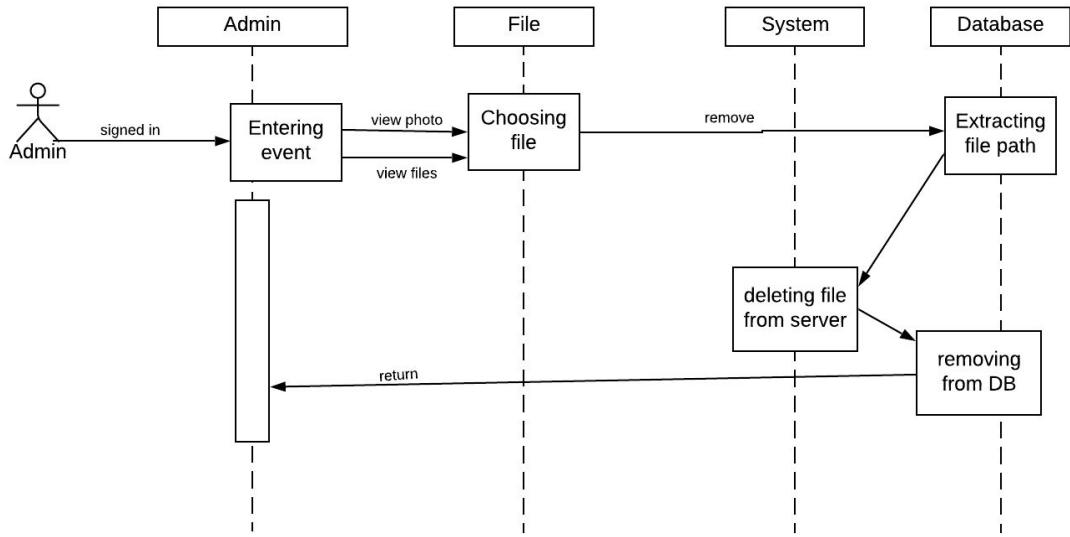
8.2.10 Uploading File



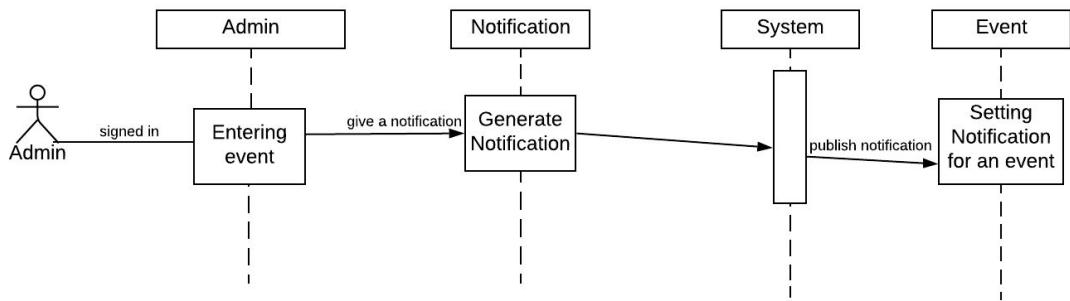
8.2.11 Downloading File



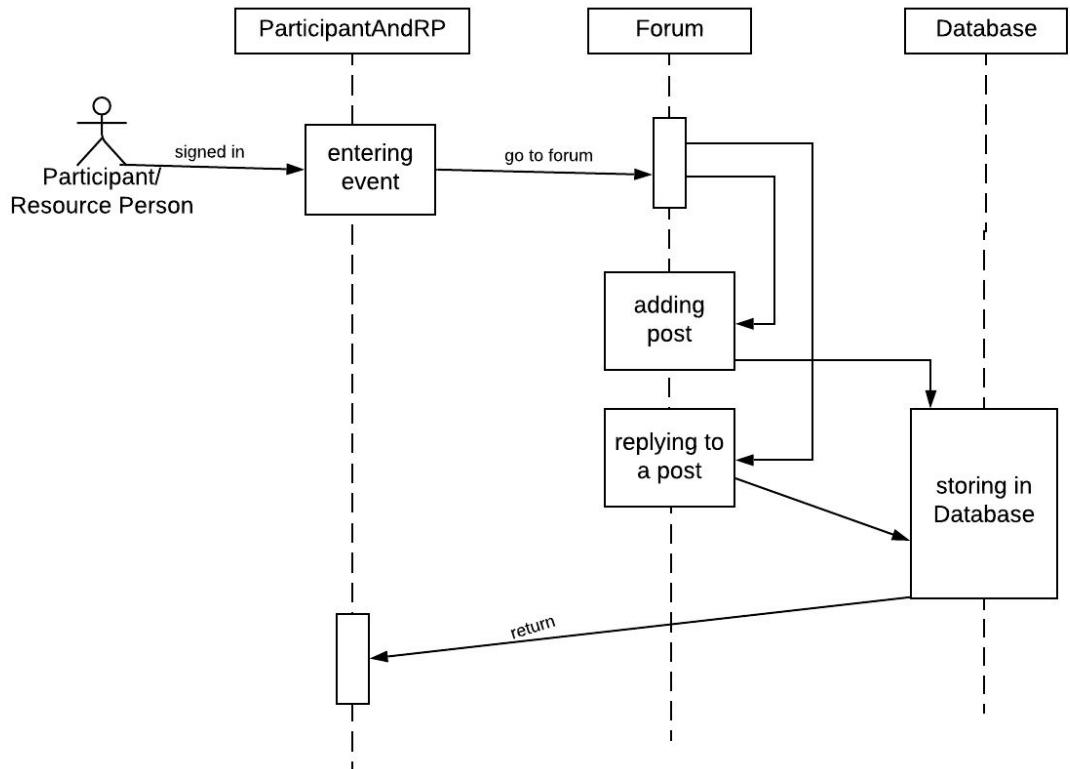
8.2.12 Remove File



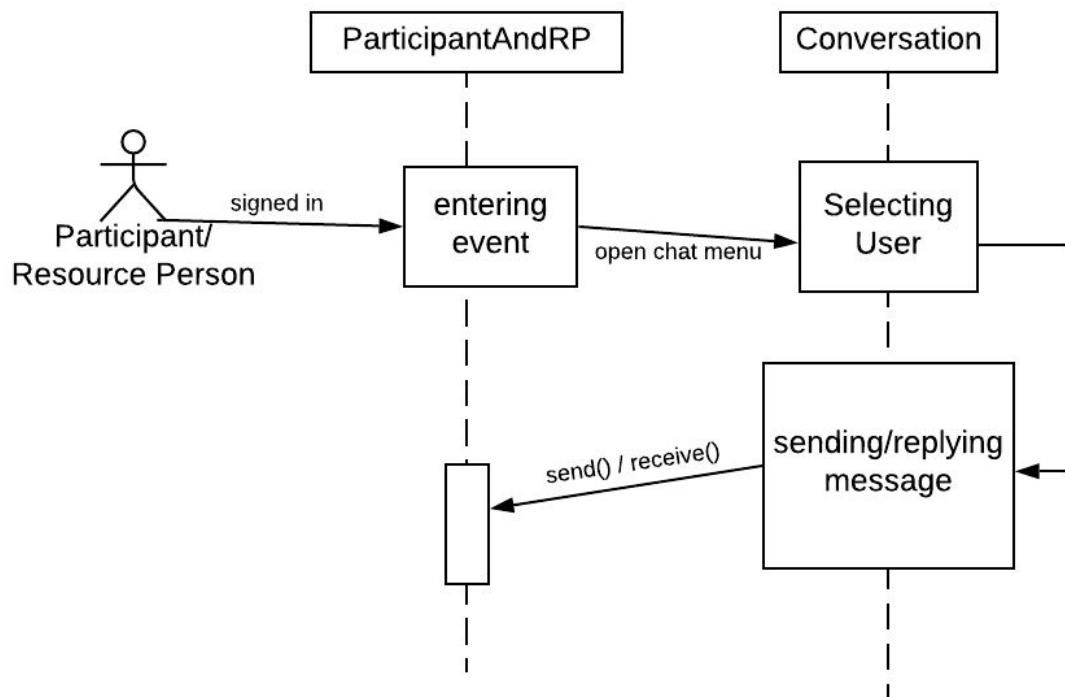
8.2.13 Send Notification



8.2.14 Forum



8.2.15 Conversation



CONCLUSION

From this SRS report on EventMeetApp, the readers will get a clear and easy view of the overall system. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using SRS. We tried our best to remove all dependencies and make an effective and fully designed SRS.

REFERENCES

- [1] Pressman, Roger S. Software Engineering: A practitioner's Approach (7th Edition)