



Daffodil
International
University

Project Title: Integrated Event Recommender and Daily Scheduling System.

Supervised by

K. M. Imtiaz-Ud-Din

Assistant Professor

Department of Software Engineering

Daffodil International University

Submitted by

1.Mrinal Mollik

ID: 151-35-975

2.Jubair Hossain Mia

ID: 151-35-1005

Department of Software Engineering

Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of

Bachelor of Science in Software Engineering.

© All right Reserved by Daffodil International University

APPROVAL

This Project titled “Integrated Event Recommender and Daily Scheduling System”, submitted by JubairHossain Mia, ID: 151-35-1005 & MrinalMollik, ID: 151-35-975 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. TouhidBhuiyan
Professor and Head

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman

K. M. Intiaz-Ud-Din
Assistant Professor

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Asif Khan Shakir
Lecturer

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Md. Nasim Akhtar
Professor

Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

External Examiner

DECLARATION

We hereby declare that we have taken this thesis under the supervision of **K. M. Imtiaz-Ud-Din, Assistant Professor, Department of Software Engineering, Daffodil International University.**

We also declare that neither this thesis/project nor any part of this has been submitted elsewhere for award of any degree.

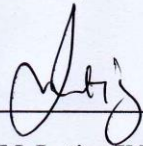
Jubair

Jubair Hossain Mia
ID: 151-35-1005
Batch :16th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Mrinal

Mrinal Mollik
ID: 151-35-975
Batch :16th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:



K. M. Imtiaz-Ud-Din

Assistant Professor

Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Integrated Event Recommender and Daily Scheduling System

ABSTRACT

Event Recommender System are used widely in social network like Facebook in any situation where user get help to find out various kinds of event. User can easily go on an event and can express interest about an event. This document perspective we would have collect and information of every single user. This system process involves providing information about various events and user own task schedule. This system will show to users the user details such as name, email, phone-number, follower and following from user own profile. The database will store that related information.

ACKNOWLEDGEMENT

We would like to express the gratitude and appreciation to all those who gave us the possibility to make our documentation more effective and also completed. A special thanks to my supervisor and our honourable teacher “K.M.Imtiaz-Ud-Din ”, whose help, simulation and encouragement, helped us to coordinate to writing this documentation.

My sincere thanks to my friends who have supported our work on the document: specially, Daffodil International University’s family members, friends and brothers and also K.M.Imtiaz-Ud-Din for his valuable and important new ideas.

Finally, and above all, we would like to thank our family and friend for their support. We wouldn’t have been able to get here without them.

Table of Contents

Contents	Page
Abstract	ii
Acknowledgement	iii
Chapter 1: Introduction	1-3
1.1 Project overview	1
1.2 Project Purpose	2-3
Chapter 2: Software Requirement Specification	4-10
2.1 Functional requirements	4-5
2.2 Non-functional requirements	5
2.3 Performance requirements	5
2.4 Security Requirements	6
2.5 SRS (Details)	6-10
Chapter 3: System Analysis	11-30
3.1 Use case Diagram	11
3.2 Use case Description	12-21
3.3 Activity Diagram	22
3.4 Sequence Diagram	23-30
Chapter 4: System Design Specification	31-33
4.1 Class Diagram	31
4.2 Entity Relationship Diagram	32
4.3 Development Tools & Technology	33
Chapter 5: System Testing	34-41
5.1 Testing Features	34
5.2 Test case result & Decision	35-41
Chapter 6: System Maintenance	42
6.1 Corrective, Adaptive, Preventive Maintenance	42
Chapter 7: User Manual	43-47
7.1 Introduction	43
7.2 Login Page	43
7.3 Registration Page	44

7.4 Create Event Page	45
7.5 Create schedule Page	46
7.6 User Profile	47
Chapter 8: Project Summary	48
8.1 GitHub Link	48
8.2 Limitations	48
8.3 Future Scope	48
8.4 Conclusion	48

List of Figures

Figure:1	Use Case Diagram
Figure:2	Activity Diagram
Figure:3	Create an event Sequence Diagram
Figure:4	Edit event Sequence Diagram
Figure:5	Delete event Sequence Diagram
Figure:6	Search event Sequence Diagram
Figure:7	Add schedule Sequence Diagram
Figure:8	Edit schedule Sequence Diagram
Figure:9	Delete schedule Sequence Diagram
Figure:10	Daily schedule Sequence Diagram
Figure:11	Class diagram of event recommender system
Figure:12	Entity Relationship Diagram
Figure:13	Login Page
Figure:14	Registration Page
Figure:15	Create Event Page
Figure:16	Create Schedule Page
Figure:17	User Profile

Chapter:1

Introduction

1.1 Project Overview:

This report shows the design of an application that we developed. For this project perspective we would have given a name and that's "**Event Recommender System**". This System are easily to use for the user. Where we can save details about the user easily.

In this project we are representing about event, schedule and daily schedule. An event can be described as a public assembly for the purpose of celebration, education, marketing and reunion. On the other side, Scheduling define how to order the individual task in daily life and Daily schedule will merge event and schedule both.

While User is open this System, it wants a user name and password to enter the system. If the user is not authenticated then that user will get registration form to be authenticated by fulfil registration page. Then user can access the system through valid email & password.

In **Event Recommender System** option, we have given some menu like Search, schedule, daily schedule, Registration, log out, user profile etc.

1.2 Project Purpose:

1.2.1 Background:

In this project an authenticate user can create multiple event, then the other authenticate user can join or remove on this event with their suitable teste. After joining an event user can see all the details about that event. User also auto join with similar kinds of events. User can search an event through his friend's name, event place, event type, date.

In our project we have manage every single user profile with his own details such as name, email, phone-number, user can make his profile public or private. If a user makes his profile public then other user can see his interest, going on event on the other hand if a user makes his profile private then other user can't get any information from that user account.

Here, also a scheduling process which helps user to create his own working time schedule. User can fix his work or meeting time on this calendar. This calendar schedule will help user to join event, which one he (User) really wants.

The final one is daily schedule which merge all event and schedule. In daily schedule user can see all event that he (user) wants to going and also, he (user) can see his personal schedule plan. In our system, in daily schedule user will get a notification before every event or schedule task coming on our system.

1.2.2 Benefits & Beneficiaries:

In our project, we are trying to solve some major issue on social media communication. For example, In Facebook network, we are facing some problems in event part like we can only see that kinds of event that are sharing by our friend list friends but we can't see those kinds of event that we really want to go or interested.

In our project we solve that problem that's why an authenticate user must be helpful by using our project. An authenticate user can see all kinds of his (user) interested event. User can express his (user) own opinion on that kinds of event.

Our project helps a user to save his (user) time and make it easy to find his interested event. In our project also has scheduler that helps a user to set a schedule about his (user) own task and has a daily scheduler that helps user to see all kinds of event that he (user) wants to go and his (user) own scheduler task.

1.2.3 Goals & Objectives:

Our main objectives on this project is to solve the problem of a user who face problem on Facebook in event recommender sector. Our project helps that kinds of authenticate user to search event that he (user) liked or based on his (user) taste.

An authenticate user can create various types of event and express his (user) own decision about that event. User also can make his (user) own task scheduler on another portion. User also see various event and task scheduler on a particular newsfeed.

User can add his interest type on his own profile also user can change his interest type when user change his interest type user will get those events on home page which he expresses interest recently on our system.

On our system user can get those kinds of event that is going or interested by his friends, user can search events through his friend's name, place name, event type etc.

Chapter: 2

Software Requirement Specification

2.1 Functional Requirements:

2.1.1 Functional requirements of Events:

2.1.1.1 Event Image: User can add an image when he (user) create a new event on our system.

2.1.1.2 Event Type: User will add a type when he (user) create a new event such as dancing, study, work, movie etc.

2.1.1.3 Event Name: User will add a name of a new event. For example: event 1, event 2 etc.

2.1.1.4 Event Place: User need to declare place where that event will be organized. For example: Dhanmondi, Kolabagan etc.

2.1.1.5 Event Date: User will declare event start date and end date on that event.

2.1.1.6 Event Time: User will declare event start time and end time on that event.

2.1.2 Functional requirements of Scheduler:

2.1.2.1 Scheduler Name: User will add a name of a new schedule task. For example: meeting, food, party etc.

2.1.2.2 Scheduler Date: User will declare task start date and end date on that scheduler.

2.1.2.3 Scheduler Time: User will declare task start time and end time on that scheduler.

2.1.2.4 Scheduler Description: User will give a short description about that schedule task on the scheduler.

2.1.3 Functional requirements of User Interest:

2.1.3.1 User Id: Every user has a unique user id on our system. That user id saves on our database.

2.1.3.2 Interest on: User can express his own interest on an event as going, interested.

2.1.4 Interest on Events:

2.1.4.1 Event Type: User can see every event category on our system. Such as music, dancing, movie etc.

2.1.4.2 Previous Interest: Our system is highly capable to remind user previous interest on an event. Our system will show user previous interest.

2.1.4.3 Previous Going: Our system is also can remind that user previous going details on an event.

2.2 Non-functional Requirements:

2.2.1 Performance requirements: The performance shall depend upon hardware components of the users.

2.2.2 Security: The system's back-end servers shall only be accessible to authenticated administrators.

2.2.3 Maintainability: Repair time shall be within some hours.

2.3 Performance Requirements:

2.3.1 Speed requirements: The response of particular part on our project depends on user using experience. If any user is used to with social network such as Facebook then that user can easily use our system.

2.3.2 Latency requirements: There is no latency so far on our project, it perform well. Every particular field work well in our project and it run well.

2.3.3 Accuracy requirements: On our project there is no requirement that conflict with other requirements. Each requirement is independently assigning on our project.

2.4 Security Requirements:

2.4.1 Access requirements: To access our system as a user need to be authenticated and that user must be registered. If any user is not authenticated then user can't access our system. In our system we have done email and password validation.

2.4.2 Integrity requirements: Our system will notify all the authenticate user if there is any problem on our system in future. Our system will send email on every user email id if our system faces any problem in near future.

2.4.3 Privacy requirements: Our system is highly capable to provide user privacy about their account on the system. User can make his (user) profile public or private. If any user makes his profile private that any other user can't see that user interest or user details.

2.5 Software Requirements Specification (Details):

2.5.1 Software Requirements Specification for Event:

1. Introduction:

1.1 Purpose:

The purpose of this SRS document is to provide a detailed overview of event process. It defines how can a user create an event, can search an event.

1.2 Scope:

The event will help the user through provide event name, type, place, short description about that event.

2. Overview:

Event system is a web-based system that help user to find his suitable event and can express his own opinion. It allows users to visit the website. User can create own event with fulfil all requirements.

3. Specific Requirements:

3.1 Functional requirements:

3.1.1 Event Image: User can add an image when he (user) create a new event on our system.

3.1.2 Event Type: User will add a type when he (user) create a new event such as dancing, study, work, movie etc.

3.1.3 Event Name: User will add a name of a new event. For example: event 1, event 2 etc.

3.1.4 Event Place: User need to declare place where that event will be organized. For example: Dhanmondi, Kolabagan etc.

3.1.5 Event Date: User will declare event start date and end date on that event.

3.1.6 Event Time: User will declare event start time and end time on that event.

3.2 Non-Functional requirements:

3.2.1 Performance requirements: The performance shall depend upon hardware components of the clients/ customers.

3.2.2 Security: The system's back-end servers shall only be accessible to authenticated administrators.

3.2.3 Maintainability: Repair time shall be within some hours.

4. System specifications:

4.1 Software requirements:

4.1.1 Front-end development: JQuery, HTML, CSS, PHP.

4.1.2 Back-end development: PHP, MySQL.

2.5.2 Software Requirements Specification for scheduler:

1. Introduction:

1.1 Purpose:

The purpose of this SRS document is to provide a detailed overview of Schedule process. It defines how can a user create his own schedule task on our system.

1.2 Scope:

The schedule will help the user through provide set his own task and user can see an event if it's happened on that time when that user set an individual task, then user can skip that event and focus on his own task.

2. Overview:

Schedule system is a web-based system that help user to set his own task on our system and can express his own opinion on event. It allows users to visit the website. User can create own schedule with fulfil all requirements.

3. Specific Requirements:

3.1 Functional requirements:

3.1.1 Scheduler Name: User will add a name of a new schedule task. For example: meeting, food, party etc.

3.1.2 Scheduler Date: User will declare task start date and end date on that scheduler.

3.1.3 Scheduler Time: User will declare task start time and end time on that scheduler.

3.1.4 Scheduler Description: User will give a short description about that schedule task on the scheduler.

3.2 Non-Functional requirements:

3.2.1 Performance requirements: The performance shall depend upon hardware components of the clients/ customers.

3.2.2 Security: The system's back-end servers shall only be accessible to authenticated administrators.

3.2.3 Maintainability: Repair time shall be within some hours.

4. System specifications:

4.1 Software requirements:

4.1.1 Front-end development: JQuery, HTML, CSS, PHP.

4.1.2 Back-end development: PHP, MySQL.

2.5.3 Software Requirements Specification for Interest on Event:

1. Introduction:

1.1 Purpose:

The purpose of this SRS document is to provide a detailed overview of Interest on event process. It defines how can a user express his interest on various types of events on our system.

1.2 Scope:

The process will help the user through provide events that the user chooses and wants from our system and user can see an event, express his own opinion on that events then user can skip that event if he doesn't want.

2. Overview:

Interest on event system is a web-based system that help user to find out various kinds of events from our system and can express his own opinion on event. It allows users to visit the website.

3. Specific Requirements:

3.1 Functional requirements:

3.1.1 User ID: Every user has a unique user id that save our system on our database

3.1.2 Event ID: Every event that created by an authenticate user has its own event id.

3.1.3 Interest type: User can set his interest type on his own profile, user also can change his interest type from his profile.

3.1.4 Previous interest: Our system is highly capable to record that user previous interest type and also can remind new interest type on our system.

3.1.5 Previous Going: Our system is highly capable to record that user previous going expression on an event.

3.2 Non-Functional requirements:

3.2.1 Performance requirements: The performance shall depend upon hardware components of the clients/ customers.

3.2.2 Security: The system's back-end servers shall only be accessible to authenticated administrators.

3.2.3 Maintainability: Repair time shall be within some hours.

4. System specifications:

4.1 Software requirements:

4.1.1 Front-end development: JQuery, HTML, CSS, PHP.

4.1.2 Back-end development: PHP, MySQL.

2.5.4 Software Requirements Specification for Interest on User Interest:

1. Introduction:

1.1 Purpose:

The purpose of this SRS document is to provide a detailed overview of user interest process. It defines how can a user express his interest on various types of events on our system.

1.2 Scope:

The process will help the user through provide events and user can express his interest on events from our system.

2. Overview:

User Interest system is a web-based system that help user to express his own opinion on various kinds of events from our system and change his interest from on type to another type.

3. Specific Requirements:

3.1 Functional requirements:

3.1.1 User ID: Every user has a unique user id that save our system on our database.

3.1.2 Interest On: User can change his interest list, update his interest list, delete his interest list from our system.

3.2 Non-Functional requirements:

3.2.1 Performance requirements: The performance shall depend upon hardware components of the clients/ customers.

3.2.2 Security: The system's back-end servers shall only be accessible to authenticated administrators.

3.2.3 Maintainability: Repair time shall be within some hours.

4. System specifications:

4.1 Software requirements:

4.1.1 Front-end development: JQuery, HTML, CSS, PHP.

4.1.2 Back-end development: PHP, MySQL.

Chapter:3 System Analysis

3.1 Use case Diagram:

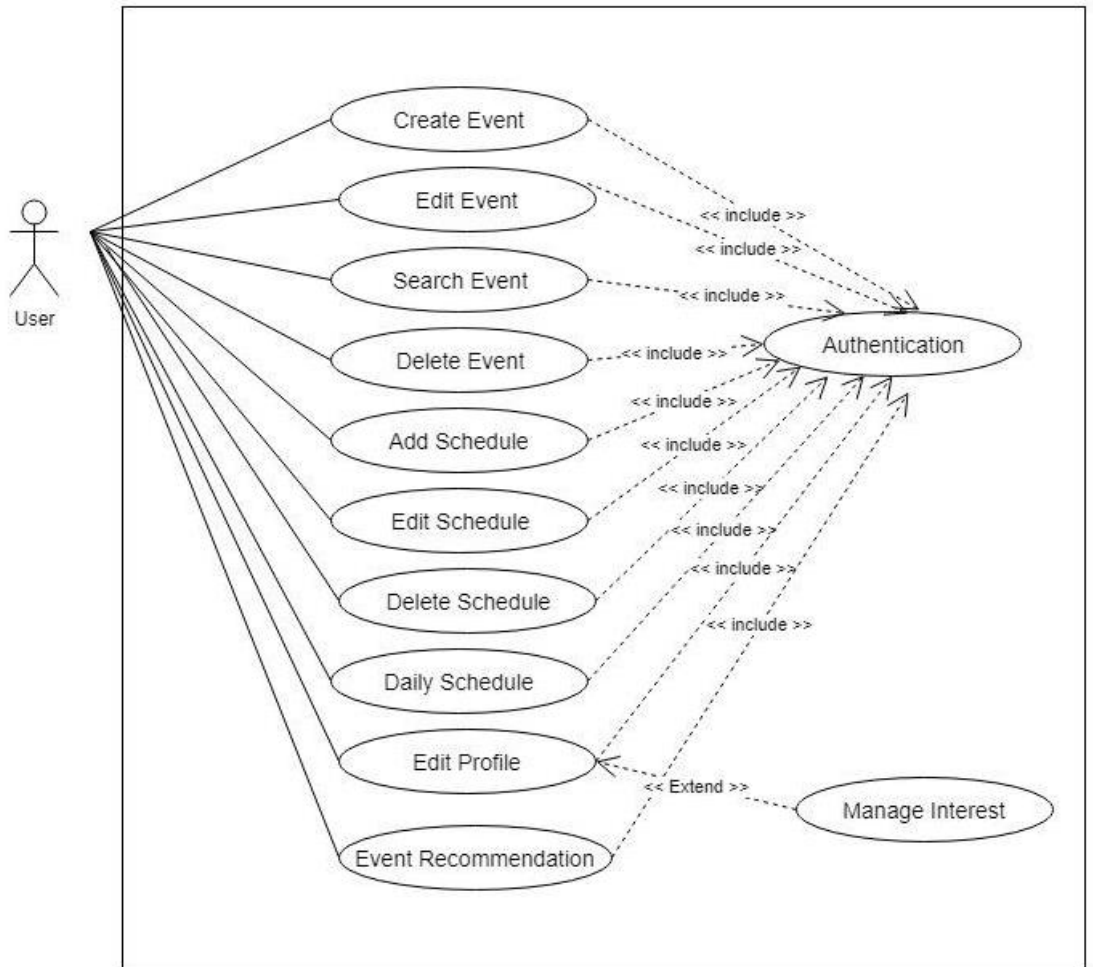


Figure 1: Use case diagram

3.2 Use Case Description (for each use case):

3.2.1 Create Event:

Use case name	Create event	
Scenario	Create a new event on system	
Brief description	An authenticate user can create an event through fulfil all the fields of add event (event name, event place, start date, start time, ending date, end time, event type, event description, image).	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Event name, event place, start date, start time, ending date, ending time, event type must exist for requested new events.	
Postconditions	New event must be created. New event contains all kinds of information. The event must be related with the user.	
Flow of events	Actor	System
	1.Create new event	1.Response for create new event
Failure results	If system not response then show an error message.	

3.2.2 Edit Event:

Use case name	Edit event.	
Scenario	Edit event on system which already created.	
Brief description	An authenticate user can edit on event. user can change event time, date, place, start time, end time etc.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Event name, event place, start date, start time, ending date, ending time, event type must exist for requested to edit an event.	
Postconditions	Event must be edited. Event contains all kinds of information that changed. The event must be related with the user.	
Flow of events	Actor	System
	1.Edit event	1.Response for edit event
Failure results	If system not response then show an error message.	

3.2.3 Search Event:

Use case name	Search event.	
Scenario	Search event on system which already created.	
Brief description	An authenticate user can search an event through event date, event place, event type, friend name etc.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Event name, event place, start date, start time, ending date, ending time, event type must exist for search an event.	
Postconditions	Search event must be showed. Event contains all kinds of information. The event must be related with the user.	
Flow of events	Actor	System
	1.Search event	1.Response for search event
Failure results	If system not response then show an error message.	

1.2.4 Delete Event:

Use case name	Delete event.	
Scenario	Delete an event on our system which already created.	
Brief description	An authenticate user can delete an event even after that event is complete or not through our system.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Event name, event place, start date, start time, ending date, ending time, event type must exist for delete an event.	
Postconditions	Event must be deleted. The event must be related with the user.	
Flow of events	Actor	System
	1.Delete event	1.Response for delete event
Failure results	If system not response then show an error message.	

3.2.5 Add Schedule:

Use case name	Create schedule task	
Scenario	Create a new schedule task on system	
Brief description	An authenticate user can add task on the system. User will helpful by that schedule. User can add various kinds of task such as food, meeting, tour etc.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Schedule name, start date, start time, ending date, ending time, schedule description must exist for requested new schedule.	
Postconditions	New schedule must be created. New schedule contains all kinds of information. The schedule must be related with the user.	
Flow of events	Actor	System
	1.Create new schedule	1.Response for create new schedule
Failure results	If system not response then show an error message.	

3.2.6 Edit Schedule:

Use case name	Edit schedule task	
Scenario	Edit schedule task on system that already created.	
Brief description	An authenticate user can edit on his personal schedule. He can change his schedule time, place etc.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Schedule name, start date, start time, ending date, ending time, schedule description must exist for requested to edit schedule.	
Postconditions	Schedule must be edited. Schedule contains all kinds of information. The schedule must be related with the user.	
Flow of events	Actor	System
	1.Edit schedule	1.Response for edit schedule
Failure results	If system not response then show an error message.	

3.2.7 Delete Schedule:

Use case name	Delete schedule task	
Scenario	Delete schedule task from system that already created.	
Brief description	An authenticate user can delete his schedule from the system.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. Schedule name, start date, start time, ending date, ending time, schedule description must exist for requested to delete schedule.	
Postconditions	Schedule must be deleted. The schedule must be related with the user.	
Flow of events	Actor	System
	1.Delete schedule	1.Response for delete schedule
Failure results	If system not response then show an error message.	

3.2.8 Daily Schedule:

Use case name	Daily schedule task					
Scenario	Daily schedule task show going event and schedule.					
Brief description	An authenticate user can see his all going event and schedule through daily schedule.					
Actor	User					
Related use cases	Includes: check authentication					
Preconditions	<p>User must exist.</p> <p>Schedule name, start date, start time, ending date, ending time, schedule description must exist for requested to schedule.</p> <p>And event name, event place, start date, start time, ending date, ending time, event type must exist for an event.</p>					
Postconditions	<p>Schedule and going event must be showed.</p> <p>The schedule and event must be related with the user.</p>					
Flow of events	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1.Add schedule and express going on event</td> <td>1.Response for schedule and going event.</td> </tr> </tbody> </table>	Actor	System	1.Add schedule and express going on event	1.Response for schedule and going event.	
Actor	System					
1.Add schedule and express going on event	1.Response for schedule and going event.					
Failure results	If system not response then show an error message.					

3.2.9 Edit Profile:

Use case name	Edit profile	
Scenario	User can edit his/her personal profile.	
Brief description	An authenticate user can edit his own profile by changing name, e-mail address etc.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. User can make his profile up to date on our system.	
Postconditions	User new update must be showed.	
Flow of events	Actor	System
	1.Request to change his/her own information	1.Response with user changed.
Failure results	If system not response then show an error message.	

3.2.10 Event Recommender:

Use case name	Event Recommender	
Scenario	User can see and express his opinion on recommended event based on his/her interest type.	
Brief description	An authenticate user can see and express his/her own opinion on a recommended event that is coming from check his interest type from his/her profile.	
Actor	User	
Related use cases	Includes: check authentication	
Preconditions	User must exist. User can express his opinion on an event and view all event that are recommended him based on his interest type.	
Postconditions	Recommended event must be showed.	
Flow of events	Actor	System
	1.Get event on his news feed based on interest type.	1.Response send event to the user based on interest type.
Failure results	If system not response then show an error message.	

3.3 Activity Diagram:

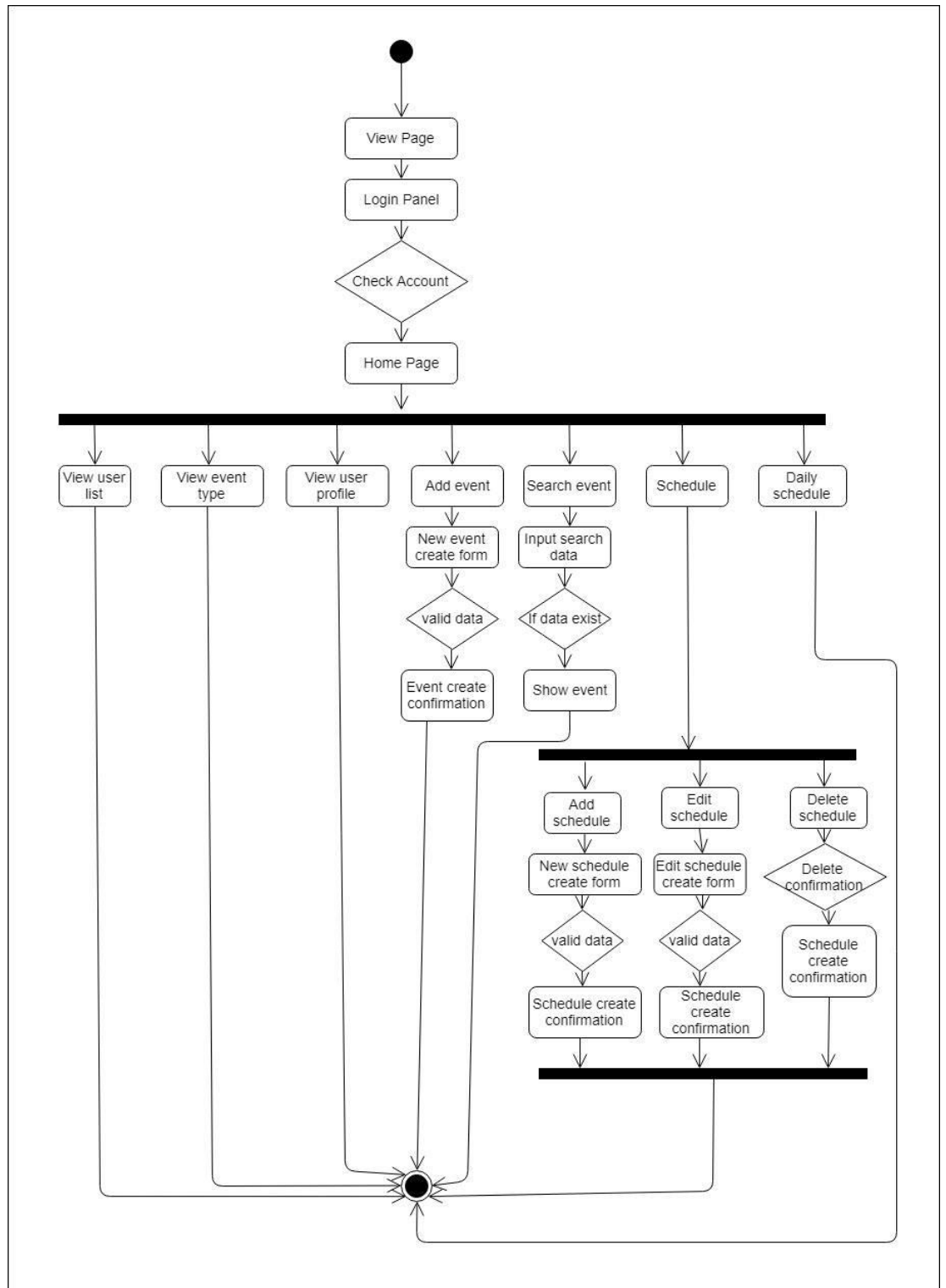


Figure-2: Activity Diagram

3.4 Sequence Diagram (for each use case)

3.3.1 Create an event use case:

The create an event use is initiated when the user clicks the “create event” button in the window. A new window will appear containing field of event name with data time. After clicking on “submit” button fill-up the given input field new event will be create sequence diagram of create an event is given below:

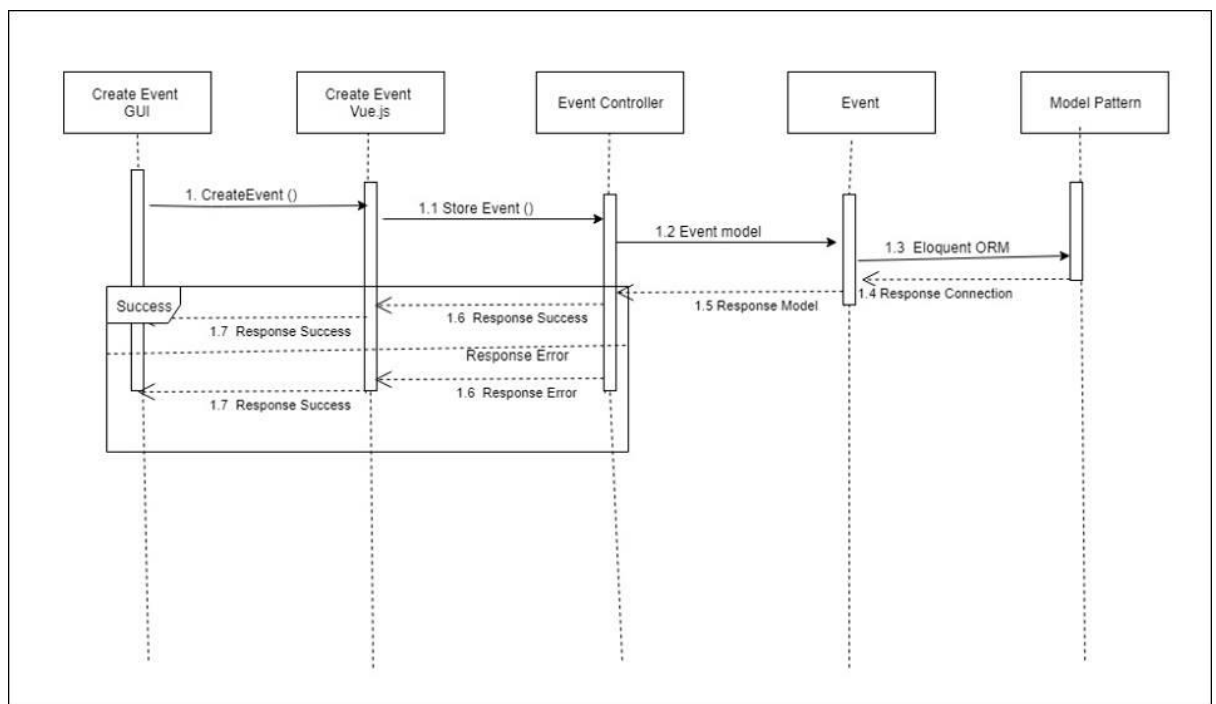


Figure 3: Create an event sequence diagram.

3.3.2 Edit event use case:

The Edit an event case is initiated when the user either highlights and edit in the list of events in the main window and then clicks the "Edit" button. The sequence diagram of an event is given below:

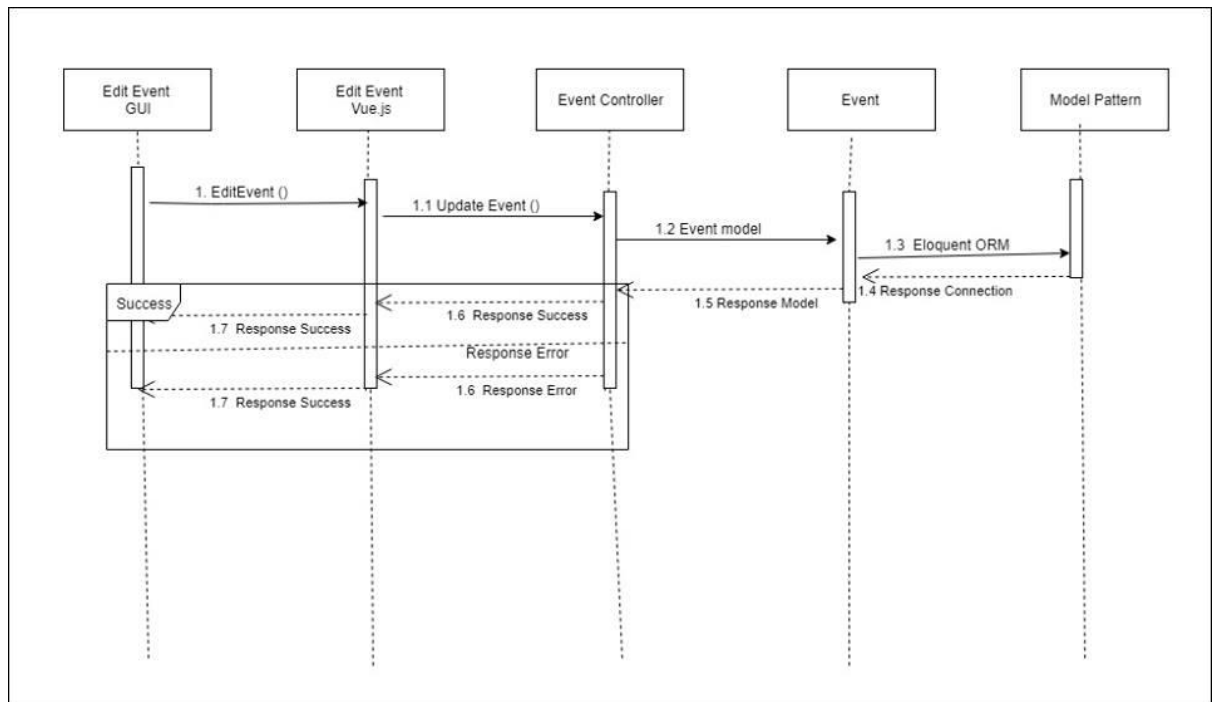


Figure 4: Edit event sequence diagram.

3.3.3 Delete event use case:

The Delete an event case is initiated when the user either highlights and edit in the list of events in the main window and then clicks the "Delete" button. The sequence diagram of an event is given below:

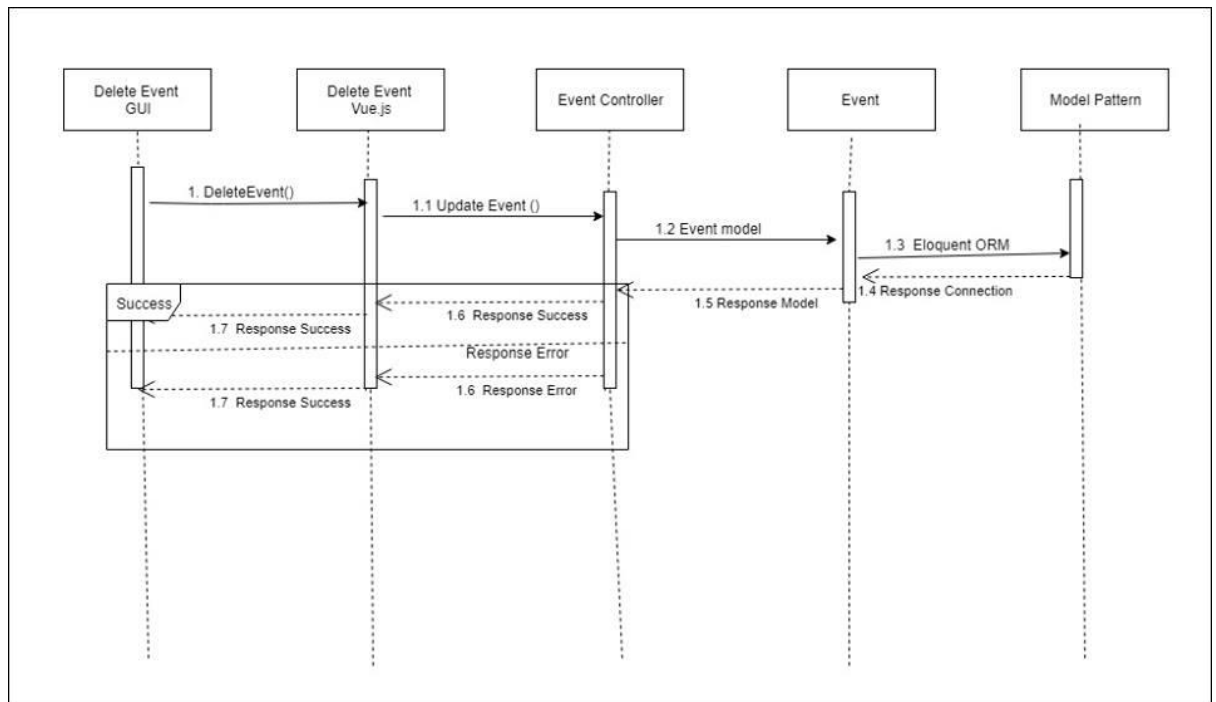


Figure 5: Delete event sequence diagram.

3.3.4 Search event use case:

The search an event case is initiated when the user either highlights and edit in the list of events in the main window and then clicks the "Search" button. The sequence diagram of an event is given below:

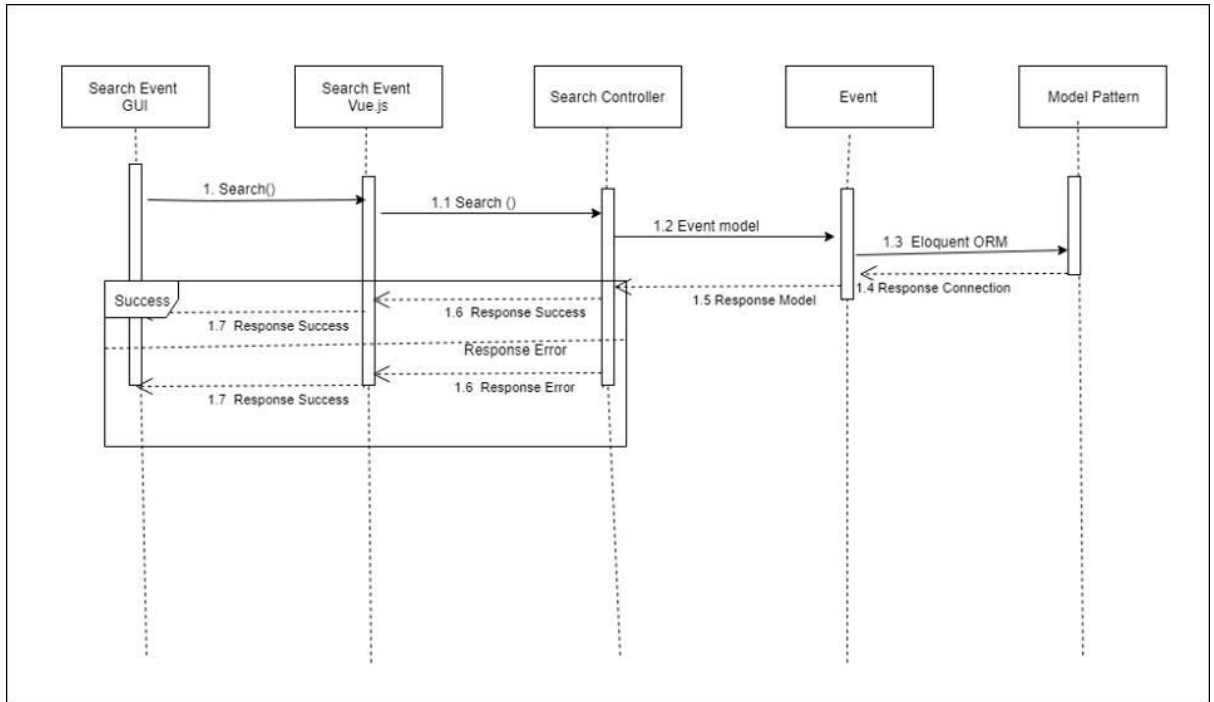


Figure 6: Search event sequence diagram.

3.3.5 Add Schedule use case:

The add schedule case is initiated when the user either highlights and edit in the list of schedules in the main window and then clicks the "Schedule" button. The sequence diagram of a schedule is given below:

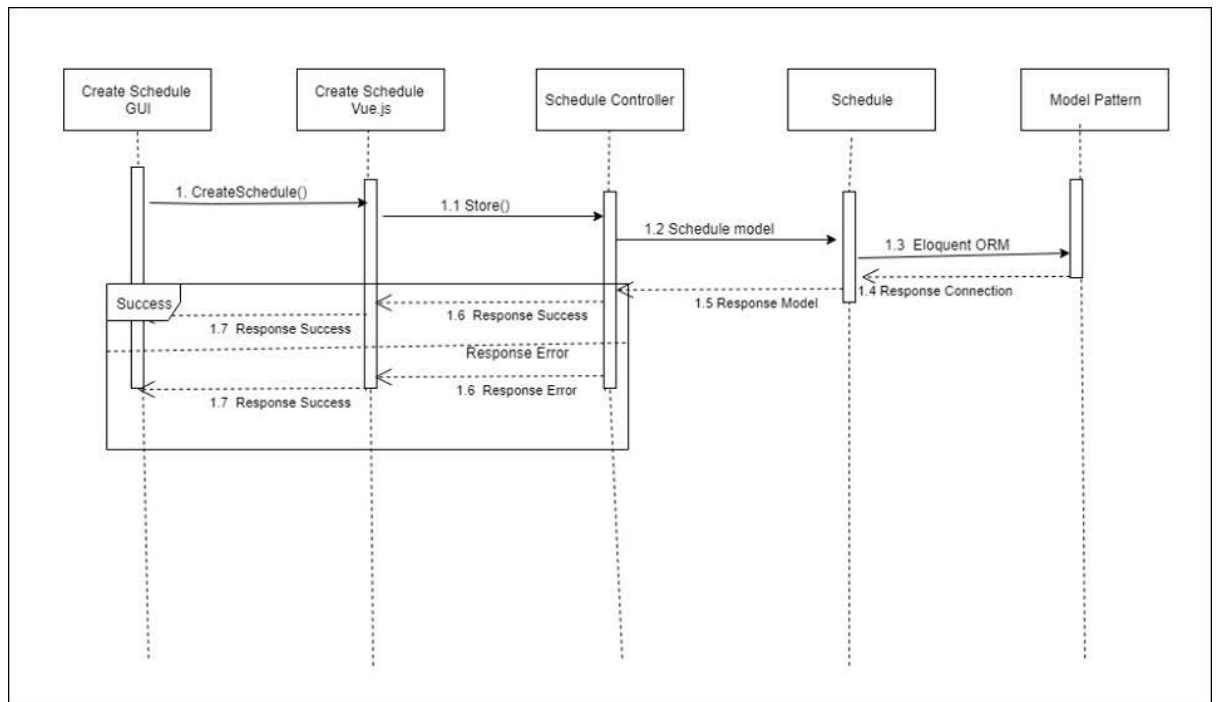


Figure 7: Add schedule sequence diagram.

3.3.6 Edit Schedule use case:

The edit schedule case is initiated when the user either highlights and edit in the list of schedules in the main window and then clicks the "Edit schedule" button. The sequence diagram of a schedule is given below:

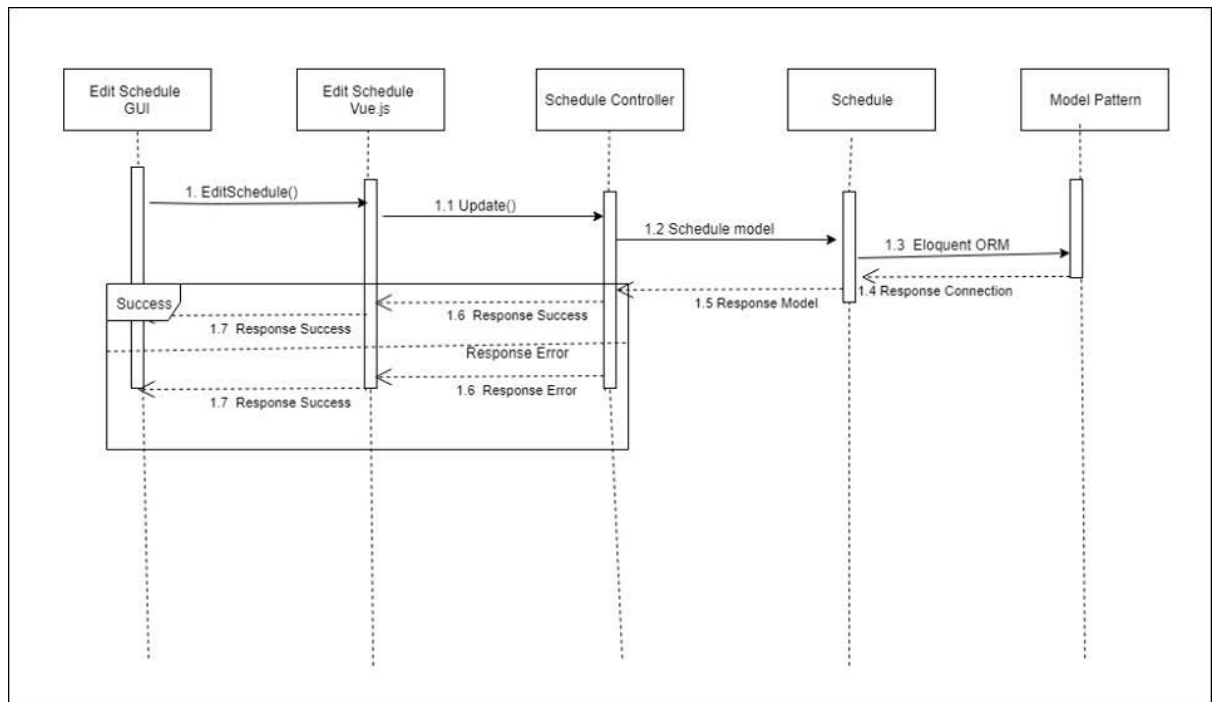


Figure 8: Edit schedule sequence diagram.

3.3.7 Delete Schedule use case:

The delete schedule case is initiated when the user either highlights and edit in the list of schedules in the main window and then clicks the "Delete schedule" button. The sequence diagram of a schedule is given below:

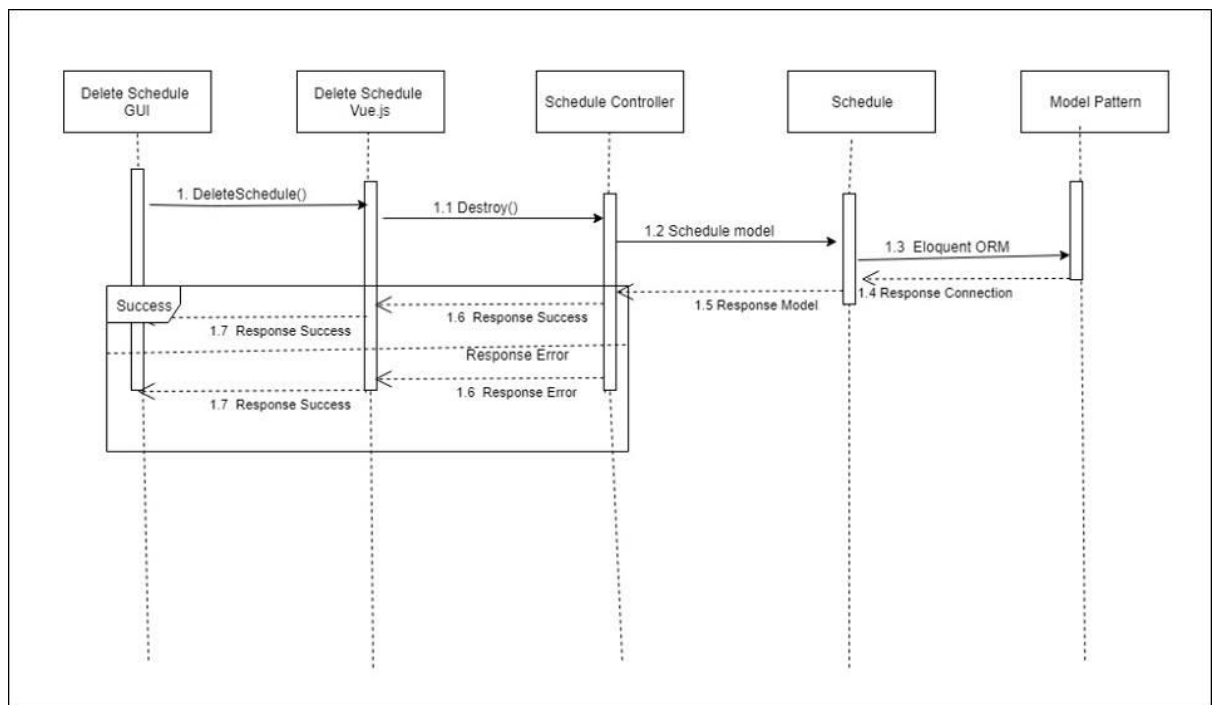


Figure 9: Delete schedule sequence diagram.

3.3.8 Daily Schedule use case:

The daily schedule case is initiated when the user either highlights and edit in the list of schedules and events in the main window and then clicks the "Daily schedule" button. The sequence diagram of a schedule is given below:

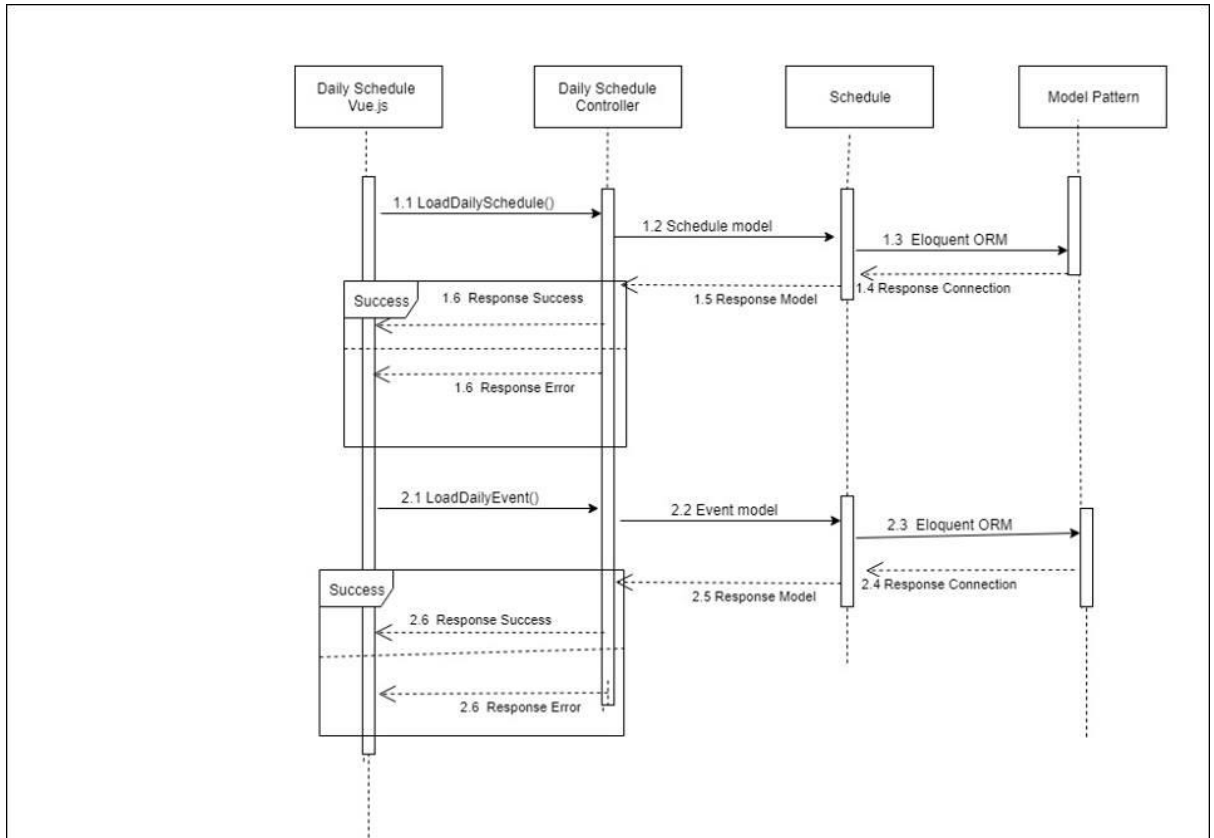


Figure 10: Daily schedule sequence diagram.

Chapter :4

System Design Specification

Introduction:

A system design specification is an information system that provide details information about the system with some major diagram. For example, design specification gives clear knowledge about the system. In this chapter we draw some diagram with its details which give much information about our “Event Recommender system”.

Diagram that we draw for our system is given below:

1. Class Diagram.
2. Entity Relationship Diagram.
3. Activity Diagram.
4. Data Flow Diagram.

4.1 Class Diagram:

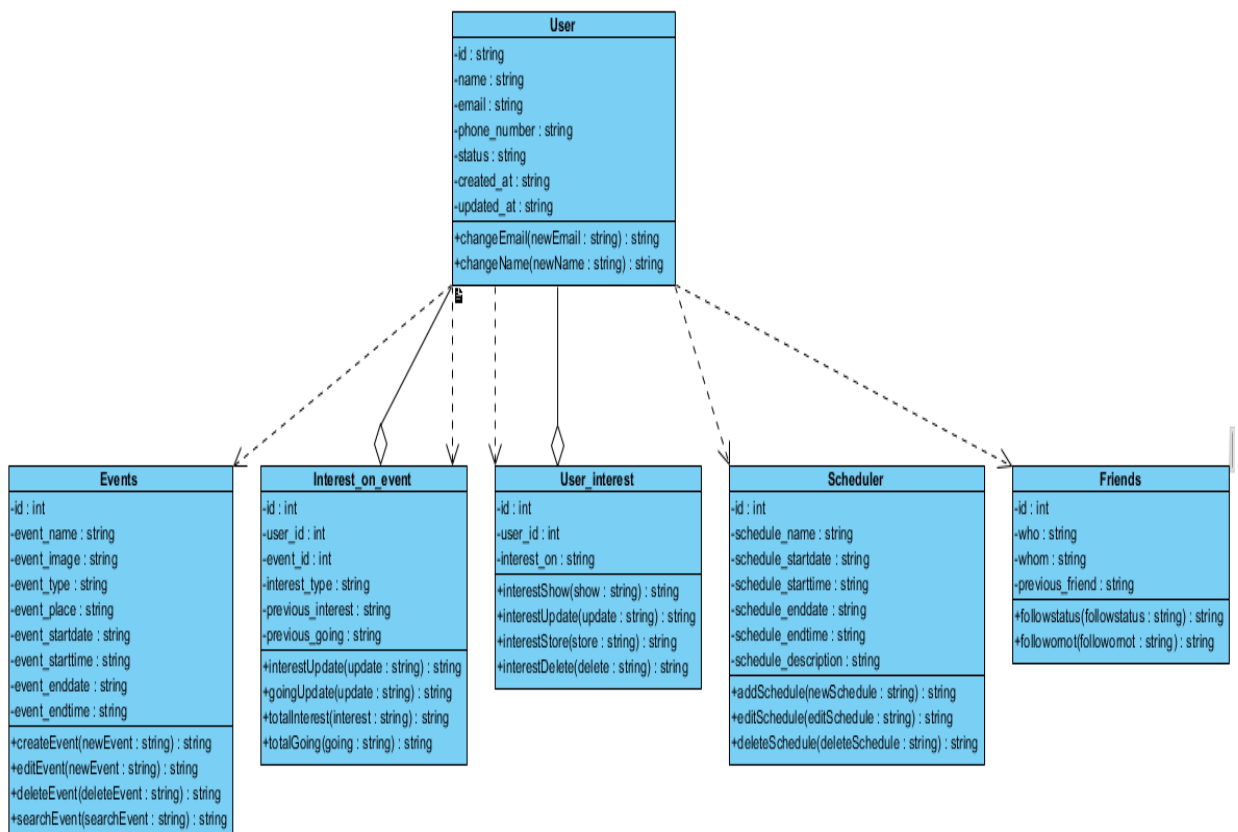


Figure 11: Class diagram of event recommender system.

4.2 Entity relationship diagram:

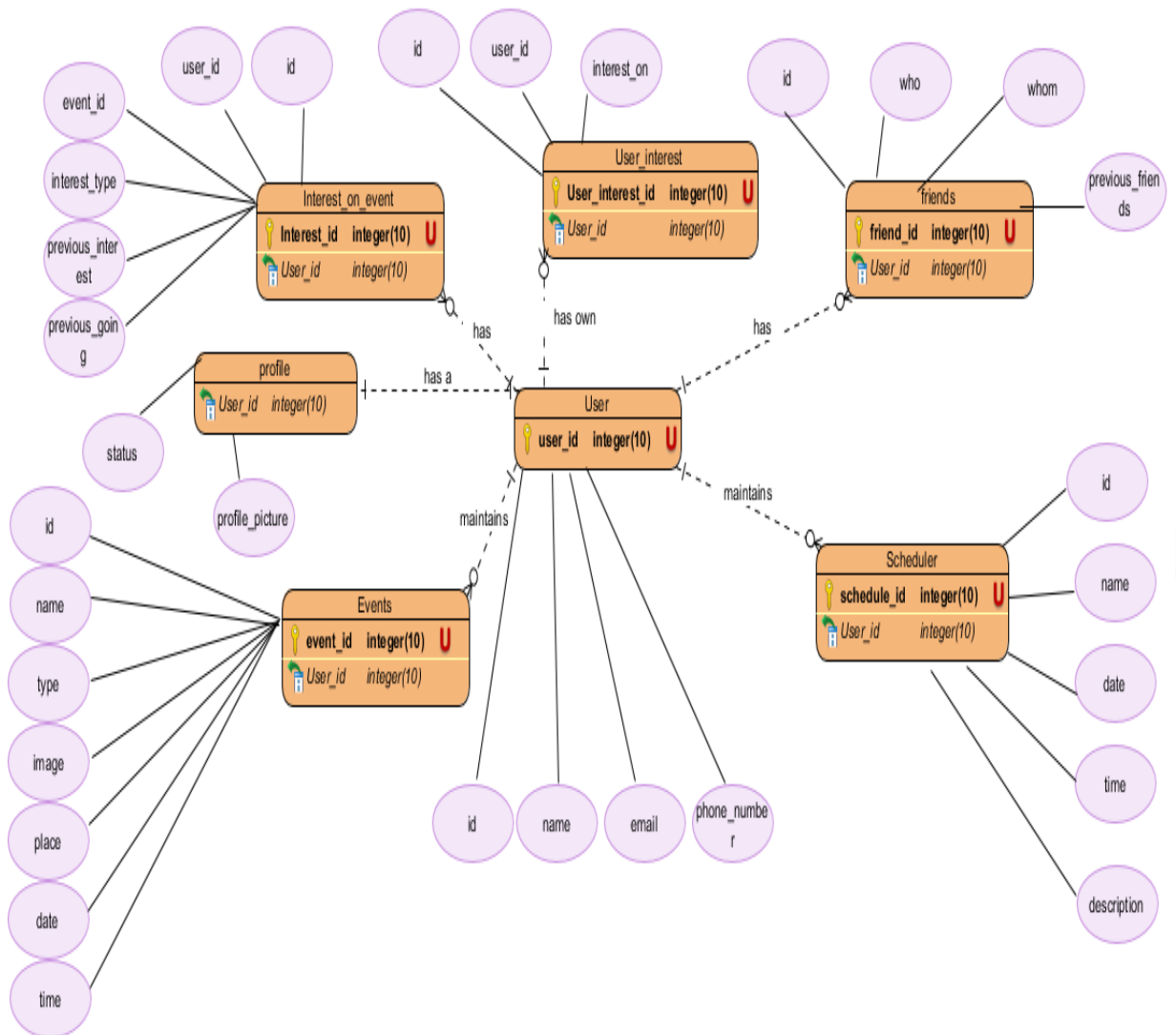


Figure-12: Entity Relationship Diagram.

4.3.1 User Interface Technology:

4.3.1.1 Laravel Framework: In our project we used Laravel framework. It is an open-source php framework. Laravel used in our project such as authentications, routing, sessions and caching.

4.3.1.2 Vue.js: In our project we also used Vue.js. It is an open-source JavaScript framework. It works for building user interface.

4.3.2 Implementation Tools & Platforms:

4.3.2.1 Microsoft visual studio 2015: In our project we used Microsoft visual studio to written code. Microsoft visual studio provides an editor for PHP, HTML and JavaScript with code analysis.

4.3.2.2 Eloquent ORM: In our project we used phpMyAdmin for database design. phpMyAdmin managing of database, table, fields, relations, user, permission.

Chapter: 5

System Testing

5.1 Testing Features:

5.1.1 Features to be tested:

5.1.1.1 Add Event

5.1.1.2 Edit Event

5.1.1.3 Delete Event

5.1.1.4 Search Event

5.1.1.5 Add Schedule

5.1.1.6 Edit Schedule

5.1.1.7 Delete Schedule

5.1.2 Features not to be tested:

5.1.2.1 Daily Schedule

5.2 Test Case Result Table and Decision:

This program is tested many times. Now, we are optimistic that the program will be executed properly. The results of the testing are...

Testing-1: Firstly, Our Application has a username and password system. If user gives wrong input, it gives a message and tells to try again doing. If it is correct then it goes to the next page. This is a Strong point of our program. We are done here validation also. Expected result is come from this function.

Test Scenario:	Login	Test Case ID:	01
Test Case Description:	Login test case	Test Priority:	High
Pre-Requisite:	A valid user account	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Enter correct email and password	Email ID: jubair@gmail.com Password: *****	Login Successfully	Login Successfully	Pass

Testing-2: User who are not authenticate, they (user) can't enter our system. If user try to login our system with invalid email and password.

Test Scenario:	Login	Test Case ID:	02
Test Case Description:	Login test case	Test Priority:	High
Pre-Requisite:	An invalid user account	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Enter incorrect email and password	Email ID: j@.com Password: *****	Login Failed	Login Failed	Fail

Testing-3: After successfully login on our system, User enter on the home page and see add event, schedule, daily schedule, user list and event type.

Test Scenario:	Home page	Test Case ID:	03
Test Case Description:	Enter home page	Test Priority:	High
Pre-Requisite:	A valid user account	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Enter correct email and password	Email ID: jubair@gmail.com Password: *****	Show home page of our system	Show home page	Pass

Testing-4: If user want to do change any information on his profile, they (user) will easily maintain this System. Our system is highly capable to save user information.

Test Scenario:	User profile	Test Case ID:	04
Test Case Description:	Update information from user profile	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Enter user profile	Phone Number: ***** Profile type: public	Update information successfully	Update successfully	Pass

Testing-5: User can search any event on our system though event name, date, event place, friend's name. By this time, they will be found it.

Test Scenario:	Search event	Test Case ID:	05
Test Case Description:	Search event through search bar	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Search event through event type Such as: Music	Event type: Music	Show music type events	Show music type events	Pass

Testing-6: An authenticate user can create a new event by complete event create form. User need to fulfil all the option of event create form such as event name, event image, event place, event description, event start time- end time, event start date- end date.

Test Scenario:	Create event	Test Case ID:	06
Test Case Description:	Create a new event	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Fill up all fields of add event.	Event Id: 03 Event type: Study Event Name: Physics Event Place: Lalbagh Event Start-time: 12:00pm	Create new event successfully	Create event Successfully	Pass

Testing-7: User can create his personal task schedule on our system by fulfil schedule form. Schedule form contains schedule name, schedule description, schedule start date- end date, schedule start time- end time.

Test Scenario:	Create schedule	Test Case ID:	07
Test Case Description:	Create a new schedule	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Fill up all fields of add schedule.	Schedule Id: 03 Schedule Name: Food Schedule Start-time: 12:00pm Schedule Start-date: 22/12/18	Create new schedule successfully	Create schedule Successfully	Pass

Testing-8: User can see a particular date event and schedule by daily schedule on our system. Daily schedule will show all event and task.

Test Scenario:	Daily Schedule	Test Case ID:	08
Test Case Description:	Show daily schedule on news feed	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Set schedule task and express going on an event	Press on daily schedule	Show event and schedule task	Show event and schedule task	Pass

Testing-9: User can see follower and following friends by his own account through our system and user can follow his friend's post about new event.

Test Scenario:	Follow Friends	Test Case ID:	09
Test Case Description:	Follow friend from his own account	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	Visit user profile and press follow button	Press follow option	Show follow successfully	Show follow successfully	Pass

Testing-10: User can express his (user) own opinion on an event by going, interested, following option.

Test Scenario:	Express opinion on event	Test Case ID:	10
Test Case Description:	User can express his opinion on various events	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	User can express going, interest, follow, following	Press going on an event	Show going on that event successfully	Show going on that event successfully	Pass

Testing-11: If user input any unexpected commend the program will error.

Expected result is come from this function. This is the strong point in our application.

Test Scenario:	Unexpected commend	Test Case ID:	11
Test Case Description:	If user press any unexpected commend	Test Priority:	High
Pre-Requisite:	A valid user	Post-Requisite:	NA

Test Execution Steps:

SL. No	Action	Inputs	Expected Output	Actual Output	Test Result
01	If user press any unexpected input on our system in any fields. For example: search event with event type: Move	Event Type: Move	Error	Error	Pass

Chapter: 6

System Maintenance

6.1 Corrective Maintenance:

6.1.1 Bug fixing: In our project we are trying to fixed all kinds of bug. Therefor, we could not find any bug to use our system. Each and every part working well and give better feed back to the user.

6.2 Adaptive Maintenance:

6.2.1 Migrating: In our project sql server database can maintain one server to another server. Every single package work perfectly on our system.

6.3 Preventive Maintenance:

6.3.1 Failure Finding: In our project failure finding should not be confused if any item is not working properly. For example: if a user searches an event with his friend name but not get proper output then it will not be confused, show an error message on the output.

6.3.2 Time Based: On our project if any user searches any event or create his own schedule then our system will take some second and give proper output on the display.

6.3.3 Condition Based: In our project condition-based maintenance helped to decide what maintenance needs to be done. For example: user searches an event through date, our system is well capable to show the output on user based on user searches.

Chapter: 7

User Manual

7.1 Introduction:

This document contains the user manual of “Event Recommender System”. Our system contains information about various events, schedule task and daily schedule.

The system can be accessed by one type of user:

1. Authenticate user: An authenticate user can access our system with his valid email and password.

7.2 Login Page:

User can login using the username and password. Forgot password option is available to retrieve password if user forgets password.

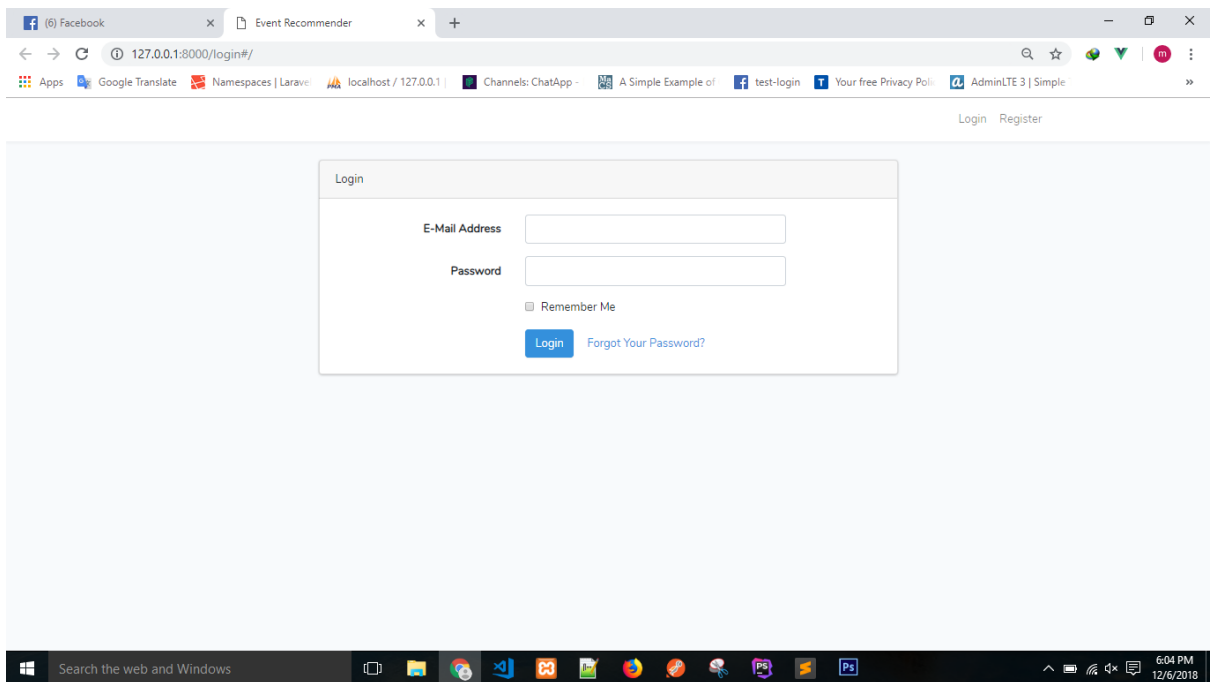


Figure-13: Login Page

7.3 Registration Page:

To become a user of our system needs to be authenticated. As user first fulfil registration page then become an authenticate user of our system.

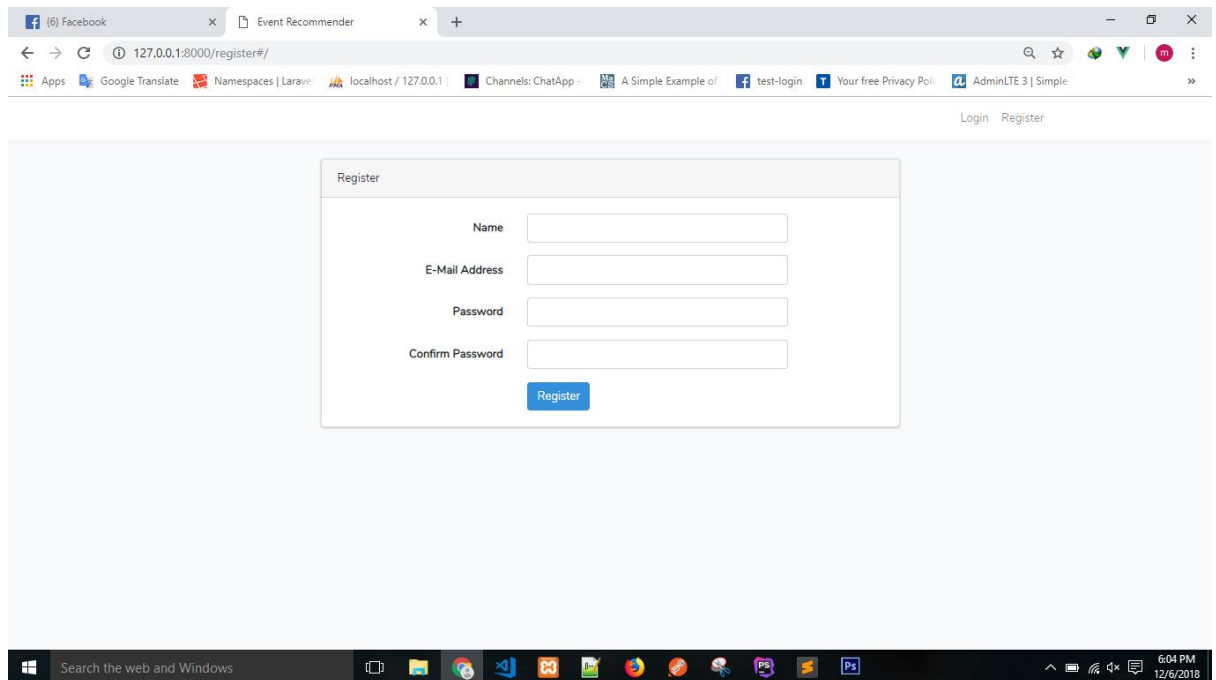


Figure-14: Registration Page

7.4 Create New Event Page:

As a user can create a new event on our system by fulfil all the fields of create event. User must write valid input on every fields. User can update, delete, search event by our system.

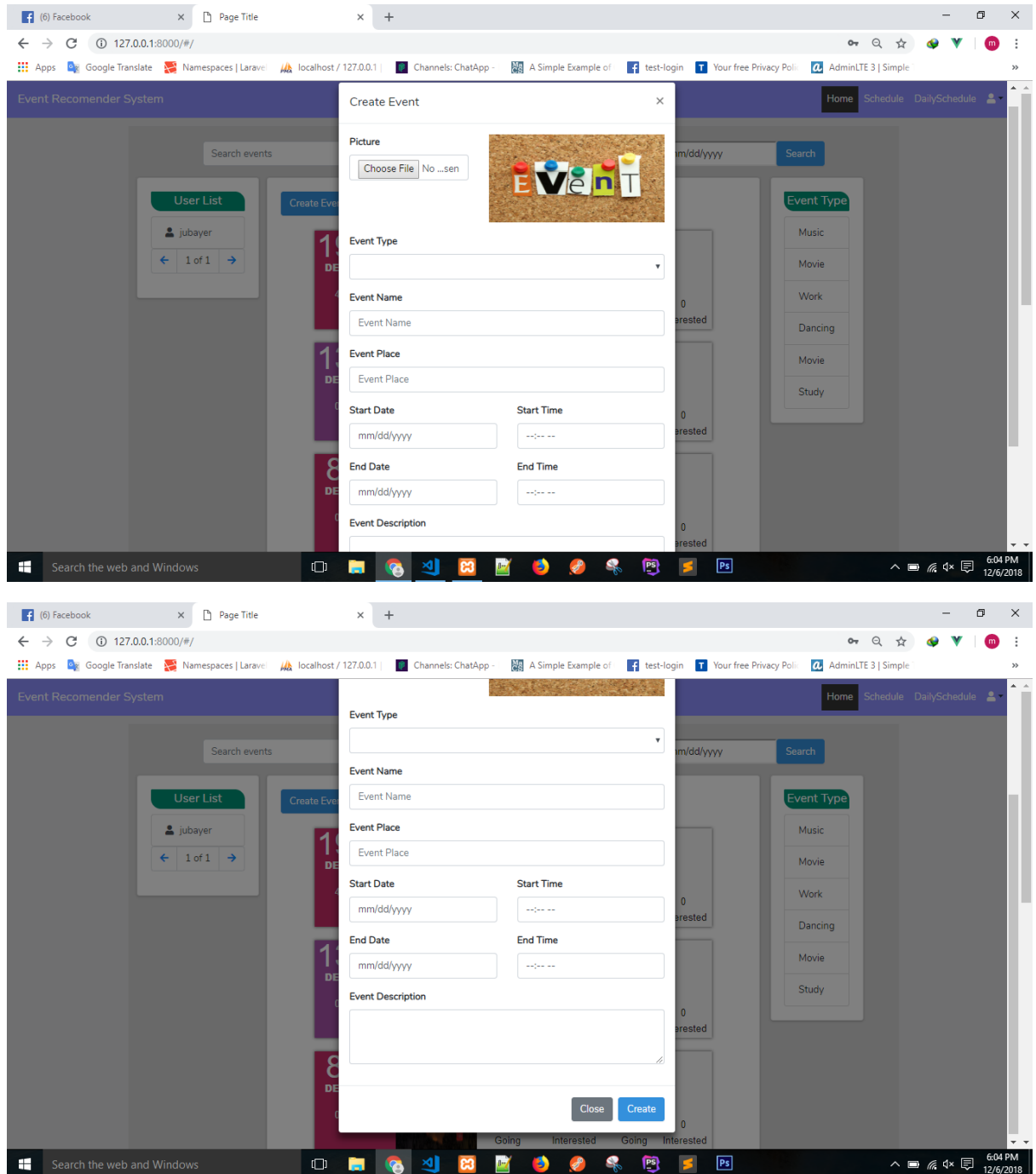


Figure-15: Create Event Page

7.5 Create Schedule Page:

User can create his own task schedule through our system. User can also edit, delete his schedule through our system.

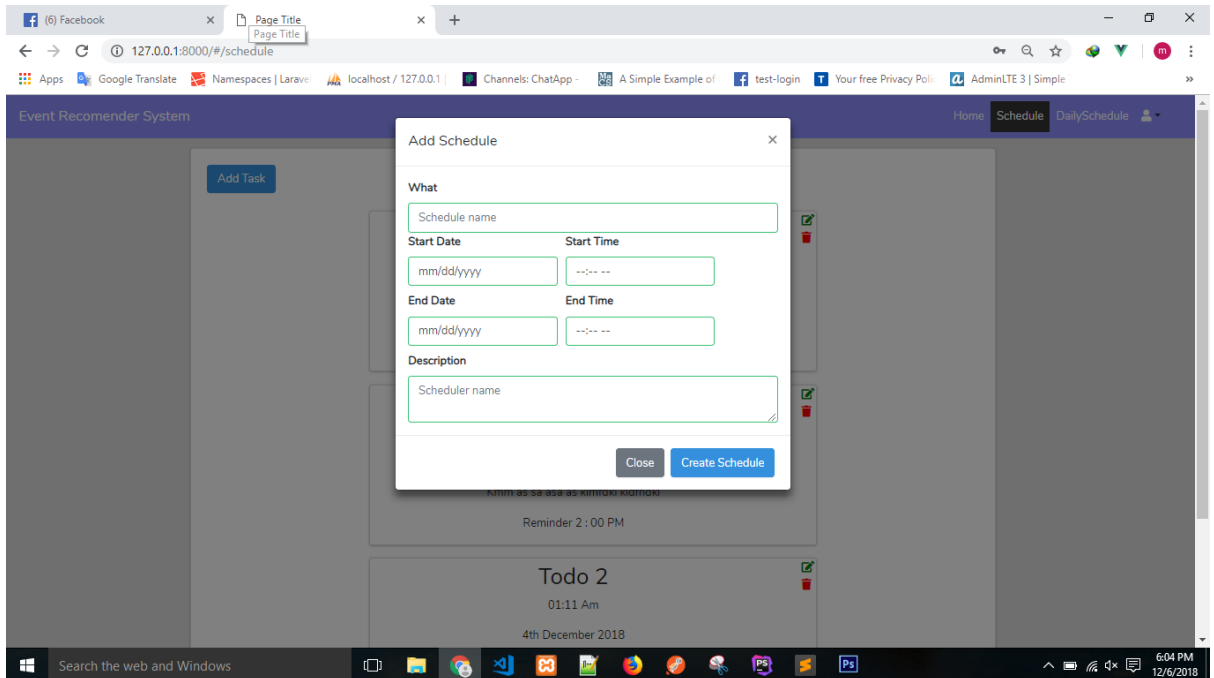
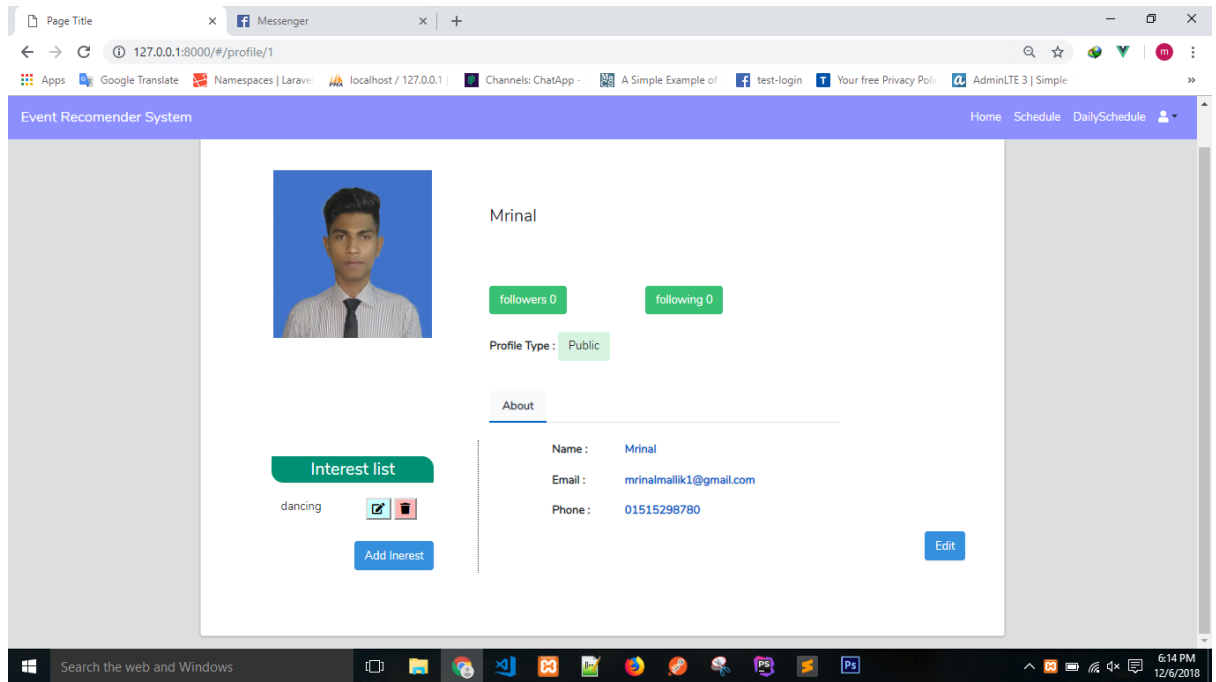


Figure-16: Create Schedule Page

7.6 User Profile:

User has own profile in our system. User can public or private his profile on our system. User can update his personal information through our system.

Profile type: Public



Profile type: Private

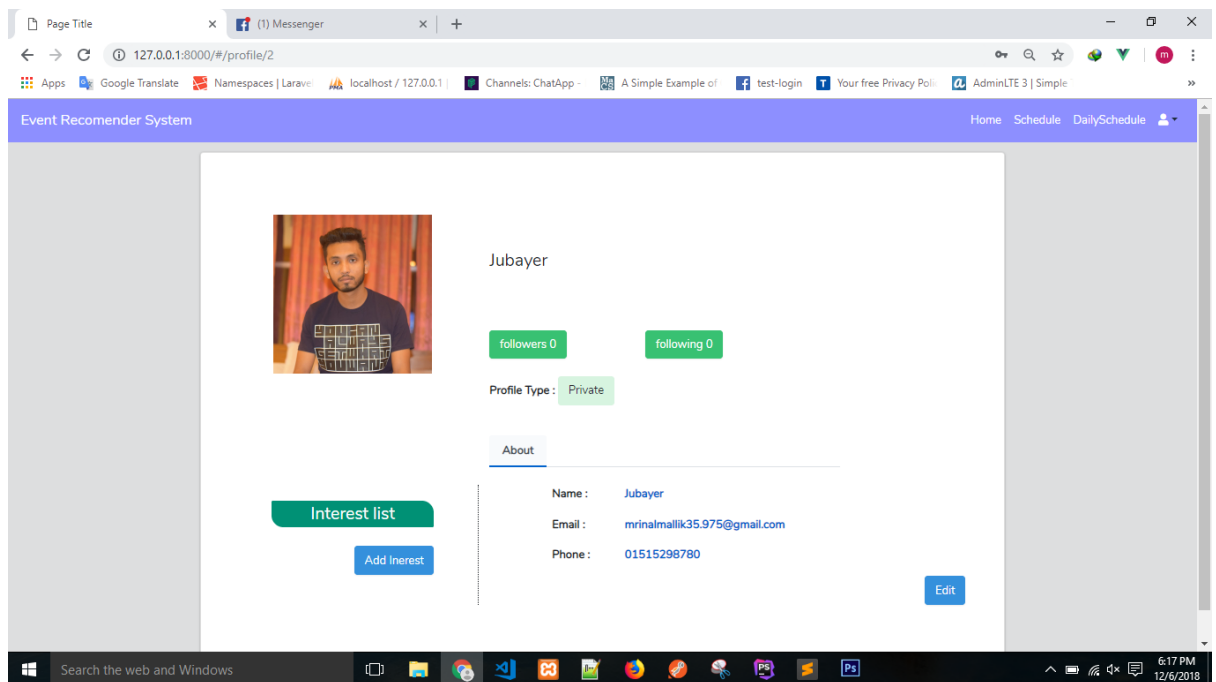


Figure-17: User Profile

Chapter: 8

Project Summary

8.1 GitHub link:

https://github.com/mrinal975/Event_Recommendation

8.2 Limitations:

8.2.1 One email address is valid for one user only in our system.

8.2.2 User can't enter our system without registration.

8.2.3 User must be used a valid user name in his profile on our system.

8.2.4 User password must be containing at least 6 characters or numbers.

8.2.5 Many users do not give the exact picture on event image as their opinion.

8.3 Future Scope:

Our main target is that sort the problem which is we already identify. We also want to add more option like “Post & share”, “video upload”, “priority-based event setup”, “reminder button on daily schedule” option, so that visitor or user can use or access all of parts as their role.

1. Post & share
2. Video upload
3. Priority-based event setup (Going, Interest, Interest-list)
4. Reminder button on daily schedule

8.4 Conclusion:

In this documentation, an information system's development has been presented. And documentation is the note of whole process. So, document should be correct and informative. In this document, we try to add correct requirement. At last We will thank our honourable teacher K.M.Imtiaz-Ud-Din to help us in making this Document.