

**Daffodil International University**

**Department of Software Engineering, FSIT**

**SWE-431 Project / Thesis**

**Project Documentation**

**Traffic Jam**

**Supervised by:**

Kaushik Sarker

Assistant Professor & Associate Head

Department of Software Engineering

Daffodil International University

**Submitted by:**

**Partha Paul**

**152-35-1154**

Department of Software Engineering

Daffodil International University

# **Approval**

# Acknowledgement

First of all I would like to thanks my God. After that I would like to express my special thanks of gratitude to my supervisor teacher Kaushik Sarker who gave me the golden opportunity to do this wonderful project on the topic of “Traffic jam”, which also helped me in doing a lot of Research, and I came to know about so many new things. He always supports me to make this project successfully. I am really thankful to him.

Secondly I would also like to thank my parents and friends who helped me lots to complete this project within the limited time.

I am also thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Software Engineering which helped me to successfully completing my project properly.

# Table of Contents

[Chapter 1: Introduction 1](#_Toc7297422)

[1.1. Project Overview 1](#_Toc7297423)

[1.2. Project Purpose 1](#_Toc7297424)

[1.2.1. Background 1](#_Toc7297425)

[1.2.2. Benefits & Beneficiaries 3](#_Toc7297426)

[1.2.3. Goals 3](#_Toc7297427)

[1.3. Stakeholders 3](#_Toc7297428)

[1.4. Proposed System Model 4](#_Toc7297429)

[1.5. Project Schedule 4](#_Toc7297430)

[1.5.1. Gantt chart 5](#_Toc7297431)

[Figure 1.5.1: Gantt chart 5](#_Toc7297432)

[1.5.2. Release Plan/Milestone 6](#_Toc7297433)

[Chapter 2: Software Requirement Specification 7](#_Toc7297434)

[2.1. Functional Requirements 7](#_Toc7297435)

[2.1.1. Sing up 7](#_Toc7297436)

[2.1.2. Sign in 7](#_Toc7297437)

[2.1.3. Create Post 7](#_Toc7297438)

[2.1.4. Update Post 7](#_Toc7297439)

[2.1.5. View Post 7](#_Toc7297440)

[2.1.6. Comment on Post 8](#_Toc7297441)

[2.1.7. Delete Post 8](#_Toc7297442)

[2.1.8. Manage User 8](#_Toc7297443)

[2.1.9. Edit and Update Profile 8](#_Toc7297444)

[2.1.10. View Site 8](#_Toc7297445)

[2.1.11. Search Post 8](#_Toc7297446)

[2.1.12. Current location 9](#_Toc7297447)

[2.1.13. Traffic Jam map 9](#_Toc7297448)

[2.1.14. Current Weather 9](#_Toc7297449)

[2.2. Data Requirements 9](#_Toc7297450)

[2.3. Performance Requirements 9](#_Toc7297451)

[2.3.1. Speed and Latency Requirements 9](#_Toc7297452)

[2.3.2. Precision or Accuracy Requirements 10](#_Toc7297453)

[2.3.3. Capacity Requirements 10](#_Toc7297454)

[2.4. Dependability Requirements 10](#_Toc7297455)

[2.4.1. Reliability Requirements 11](#_Toc7297456)

[2.4.2. Availability Requirements 11](#_Toc7297457)

[2.4.3. Robustness or Fault-Tolerance Requirements 11](#_Toc7297458)

[2.4.4. Safety-Critical Requirements 11](#_Toc7297459)

[2.5. Maintainability and Supportability Requirements 11](#_Toc7297460)

[2.5.1. Maintenance Requirements 11](#_Toc7297461)

[2.5.2. Supportability Requirements 12](#_Toc7297462)

[2.5.3. Adaptability Requirements 12](#_Toc7297463)

[2.5.4. Scalability or Extensibility Requirements 12](#_Toc7297464)

[2.6. Security Requirements 12](#_Toc7297465)

[2.6.1. Access Requirements 12](#_Toc7297466)

[2.6.2. Integrity Requirements 12](#_Toc7297467)

[2.6.3. Privacy Requirements 12](#_Toc7297468)

[2.7. Usability and Human-Interaction Requirements 13](#_Toc7297469)

[2.7.1. Ease of Use Requirements 13](#_Toc7297470)

[2.7.2. Personalization and Internationalization Requirements 13](#_Toc7297471)

[2.7.3. Understandability and Politeness Requirements 13](#_Toc7297472)

[2.7.4. Accessibility Requirements 13](#_Toc7297473)

[2.7.5. User Documentation Requirements 13](#_Toc7297474)

[2.7.6. Training Requirements 13](#_Toc7297475)

[2.8. Look and Feel Requirements 13](#_Toc7297476)

[2.8.1. Appearance Requirements 13](#_Toc7297477)

[2.8.2. Style Requirements 14](#_Toc7297478)

[2.9. Operational and Environmental Requirements 14](#_Toc7297479)

[2.9.1. Expected Physical Environment 14](#_Toc7297480)

[2.9.2. Requirements for Interfacing with Adjacent Systems 14](#_Toc7297481)

[2.9.3. Projectization Requirements 14](#_Toc7297482)

[2.9.4. Release Requirements 14](#_Toc7297483)

[2.10. Legal Requirements 14](#_Toc7297484)

[2.10.1. Compliance Requirements 14](#_Toc7297485)

[2.10.2. Standards Requirements 14](#_Toc7297486)

[Chapter 3: System Analysis 15](#_Toc7297487)

[3.1. Use Case Diagram 15](#_Toc7297488)

[3.2. Use Case Description 16](#_Toc7297489)

[3.2.1. Sign up 16](#_Toc7297490)

[3.2.2. Sign in 16](#_Toc7297491)

[3.2.3. Create user account 16](#_Toc7297492)

[3.2.4. Delete user account 17](#_Toc7297493)

[3.2.5. Create new post 17](#_Toc7297494)

[3.2.6. View all post 18](#_Toc7297495)

[3.2.7. View site 18](#_Toc7297496)

[3.2.8. Search post 18](#_Toc7297497)

[3.2.9. Get traffic jam map 19](#_Toc7297498)

[3.2.10. Get current location 19](#_Toc7297499)

[3.2.11. Get current weather 20](#_Toc7297500)

[3.2.12. Update profile 20](#_Toc7297501)

[3.2.13. Edit & Update post 20](#_Toc7297502)

[3.2.14. Comment on post 21](#_Toc7297503)

[3.2.15. Edit user profile 21](#_Toc7297504)

[3.2.16. Delete post 22](#_Toc7297505)

[3.2.17. Sign out 22](#_Toc7297506)

[3.3. Activity Diagram 23](#_Toc7297507)

[3.3.1. Sign up 23](#_Toc7297508)

[3.3.2. Sign in 23](#_Toc7297509)

[3.3.3. Create user account 23](#_Toc7297510)

[3.3.4. Delete user account 24](#_Toc7297511)

[3.3.5. Create new post 24](#_Toc7297512)

[3.3.6. View all post 24](#_Toc7297513)

[3.3.7. View site 24](#_Toc7297514)

[3.3.8. Search post 25](#_Toc7297515)

[3.3.9. See traffic jam map 25](#_Toc7297516)

[3.3.10. Get current location 25](#_Toc7297517)

[3.3.11. Get current weather 26](#_Toc7297518)

[3.3.12. Update profile 26](#_Toc7297519)

[3.3.13. Edit post & Update post 26](#_Toc7297520)

[3.3.14. Comment on post 27](#_Toc7297521)

[3.3.15. Edit user’s profile 27](#_Toc7297522)

[3.3.16. Delete published post 27](#_Toc7297523)

[3.3.17. Sign out 27](#_Toc7297524)

[3.4. System Sequence Diagram 28](#_Toc7297525)

[3.4.1. Sign up & Sign in 28](#_Toc7297526)

[3.4.2. Create user account 28](#_Toc7297527)

[3.4.3. Delete user account 28](#_Toc7297528)

[3.4.4. Edit user’s profile 28](#_Toc7297529)

[3.4.5. Update profile 29](#_Toc7297530)

[3.4.6. Create post 29](#_Toc7297531)

[3.4.7. View post 29](#_Toc7297532)

[3.4.8. Edit post 29](#_Toc7297533)

[3.4.9. Comment on post 30](#_Toc7297534)

[3.4.10. Search post 30](#_Toc7297535)

[3.4.11. Delete post 30](#_Toc7297536)

[3.4.12. View all site 30](#_Toc7297537)

[3.4.13. Get current location 31](#_Toc7297538)

[3.4.14. Get current weather 31](#_Toc7297539)

[3.4.15. Get traffic map 31](#_Toc7297540)

[3.4.16. Sign out 31](#_Toc7297541)

[Chapter 4: System Design Specification 32](#_Toc7297542)

[4.1. Class Responsibilities Collaboration (CRC) Cards 32](#_Toc7297543)

[4.1.1. User 32](#_Toc7297544)

[4.1.2. Admin 32](#_Toc7297545)

[4.1.3. Weather 32](#_Toc7297546)

[4.1.4. Post 33](#_Toc7297547)

[4.1.5. Comment 33](#_Toc7297548)

[4.1.6. Contact 33](#_Toc7297549)

[4.2. Class Diagram 34](#_Toc7297550)

[4.3. Database Design Diagram 35](#_Toc7297551)

[4.4. Development Tools & Technology 35](#_Toc7297552)

[4.4.1. User Interface Technology 35](#_Toc7297553)

[4.4.2. Implementation Tools & Platforms 36](#_Toc7297554)

[Chapter 5: System Testing 36](#_Toc7297555)

[5.1. Testing Features 36](#_Toc7297556)

[5.1.1. Features to be tested 36](#_Toc7297557)

[5.1.2. Features not to be tested 37](#_Toc7297558)

[5.2. Testing Strategies 37](#_Toc7297561)

[5.2.1. Test Approach 37](#_Toc7297562)

[5.2.1.1. Black Box Testing 37](#_Toc7297563)

[5.2.1.2. White Box Testing 37](#_Toc7297564)

[5.2.2. Pass/Fail Criteria 37](#_Toc7297565)

[5.2.3. Suspension and Resumption 38](#_Toc7297566)

[5.2.4. Testing Schedule 38](#_Toc7297567)

[5.1.1. Traceability Matrix 39](#_Toc7297568)

[5.2. Testing Environment (hardware/software requirements) 39](#_Toc7297569)

[5.3. Test Cases 40](#_Toc7297570)

[5.3.1. Sign up 40](#_Toc7297571)

[5.3.2. Sign in 40](#_Toc7297572)

[5.3.3. Create user account 41](#_Toc7297573)

[5.3.4. Create post 41](#_Toc7297574)

[5.3.5. Search post 41](#_Toc7297575)

[5.3.6. Comment on post 42](#_Toc7297576)

[5.3.7. Get current weather 42](#_Toc7297577)

[5.3.8. Get Traffic jam map 42](#_Toc7297578)

[5.3.9. Get current location 43](#_Toc7297579)

[Chapter 6: User Manual 43](#_Toc7297580)

[6.1. Sign up 43](#_Toc7297581)

[6.2. Sign in 44](#_Toc7297582)

[6.3. Traffic jam first look 44](#_Toc7297583)

[6.4. Home 44](#_Toc7297584)

[6.5. Profile 45](#_Toc7297585)

[6.6. Edit profile 45](#_Toc7297586)

[6.7. Change password 45](#_Toc7297587)

[6.8. Create post 46](#_Toc7297588)

[6.9. Comment on post 46](#_Toc7297589)

[6.10. View all comments 46](#_Toc7297590)

[6.11. Edit post 47](#_Toc7297591)

[6.12. Delete post 47](#_Toc7297592)

[6.13. Search city weather (Celsius) 47](#_Toc7297593)

[6.14. Search city weather (Fahrenheit) 48](#_Toc7297594)

[6.15. Road map with current location 49](#_Toc7297595)

[6.16. Satellite map with current location 49](#_Toc7297596)

[6.17. Traffic jam map with current location 50](#_Toc7297597)

[6.18. Search post by place name 50](#_Toc7297598)

[6.19. About us 50](#_Toc7297599)

[6.20. Contact us 51](#_Toc7297600)

[6.21. Follow us 51](#_Toc7297601)

[6.22. Sign out 51](#_Toc7297602)

[Chapter 7: Project Summary 52](#_Toc7297603)

[7.1. Github Link 52](#_Toc7297606)

[7.2. Limitations 52](#_Toc7297607)

[7.3. Obstacles & Achievements 52](#_Toc7297608)

[7.4. Future Scope 52](#_Toc7297609)

[7.5. References 53](#_Toc7297610)

Table of Figures

Figure [1.4. Proposed System Model 4](#_Toc6965227)

Figure [1.5.1. Gantt chart 5](#_Toc6965268)

Figure [3.1. Use Case Diagram 15](#_Toc6965476)

Figure [3.3.1. Sign up 23](#_Toc6965496)

Figure [3.3.2. Sign in 23](#_Toc6965497)

Figure [3.3.3. Create user account 23](#_Toc6965498)

Figure [3.3.4. Delete user account 24](#_Toc6965499)

Figure [3.3.5. Create new post 24](#_Toc6965500)

Figure [3.3.6. View all post 24](#_Toc6965501)

Figure [3.3.7. View site 24](#_Toc6965502)

Figure [3.3.8. Search post 25](#_Toc6965503)

Figure [3.3.9. See traffic jam map 25](#_Toc6965504)

Figure [3.3.10. Get current location 25](#_Toc6965505)

Figure [3.3.11. Get current weather 26](#_Toc6965506)

Figure [3.3.12. Update profile 26](#_Toc6965507)

Figure [3.3.13. Edit post & Update post 26](#_Toc6965508)

Figure [3.3.14. Comment on post 27](#_Toc6965509)

Figure [3.3.15. Edit user’s profile 27](#_Toc6965510)

Figure [3.3.16. Delete published post 27](#_Toc6965511)

Figure [3.3.17. Sign out 27](#_Toc6965512)

Figure [3.4.1. Sign up & Sign in 28](#_Toc6965514)

Figure [3.4.2. Create user account 28](#_Toc6965515)

Figure [3.4.3. Delete user account 28](#_Toc6965516)

Figure [3.4.4. Edit user’s profile 28](#_Toc6965517)

Figure [3.4.5. Update profile 29](#_Toc6965518)

Figure [3.4.6. Create post 29](#_Toc6965519)

Figure [3.4.7. View post 29](#_Toc6965520)

Figure [3.4.8. Edit post 29](#_Toc6965521)

Figure [3.4.9. Comment on post 30](#_Toc6965522)

Figure [3.4.10. Search post 30](#_Toc6965523)

Figure [3.4.11. Delete post 30](#_Toc6965524)

Figure [3.4.12. View all site 30](#_Toc6965525)

Figure [3.4.13. Get current location 31](#_Toc6965526)

Figure [3.4.14. Get current weather 31](#_Toc6965527)

Figure [3.4.15. Get traffic map 31](#_Toc6965528)

Figure [3.4.16. Sign out 31](#_Toc6965529)

Figure [4.1.1. User 32](#_Toc6965532)

Figure [4.1.2. Admin 32](#_Toc6965533)

Figure [4.1.3. Weather 32](#_Toc6965534)

Figure [4.1.4. Post 33](#_Toc6965535)

Figure [4.1.5. Comment 33](#_Toc6965536)

Figure [4.1.6. Contact 33](#_Toc6965537)

Figure [4.2. Class Diagram 34](#_Toc6965538)

Figure [4.3. Database Design Diagram 35](#_Toc6965539)

Figure [6.1. Sign up 43](#_Toc6965569)

Figure [6.2. Sign in 44](#_Toc6965570)

Figure [6.3. Traffic jam first look 44](#_Toc6965571)

Figure [6.4. Home 44](#_Toc6965572)

Figure [6.5. Profile 45](#_Toc6965573)

Figure [6.6. Edit profile 45](#_Toc6965574)

Figure [6.7. Change password 45](#_Toc6965575)

Figure [6.8. Create post 46](#_Toc6965576)

Figure [6.9. Comment on post 46](#_Toc6965577)

Figure [6.10. View all comments 46](#_Toc6965578)

Figure [6.11. Edit post 47](#_Toc6965579)

Figure [6.12. Delete post 47](#_Toc6965580)

Figure [6.13. Search city weather (Celsius) 47](#_Toc6965581)

Figure [6.14. Search city weather (Fahrenheit) 48](#_Toc6965582)

Figure [6.15. Road map with current location 49](#_Toc6965583)

Figure [6.16. Satellite map with current location 49](#_Toc6965584)

Figure [6.17. Traffic jam map with current location 50](#_Toc6965585)

Figure [6.18. Search post by place name 50](#_Toc6965586)

Figure [6.19. About us 50](#_Toc6965587)

Figure [6.20. Contact us 51](#_Toc6965588)

Figure [6.21. Follow us 51](#_Toc6965589)

Figure [6.22. Sign out 51](#_Toc6965590)

# Chapter 1: Introduction

## Project Overview

This system name is Traffic jam. In this system, there are two types of actor. They are User and Admin

First of all, users must need to authenticate for use this system. After authentication users can view all site .User can see where traffic jam is. Users will know the reason for traffic jam. Here users can create post of traffic jam with description. Users can comments on post. User can find of their current location. For this post others user easily can know where is jam and why it was created. Users can view all post. Users can search traffic jam post using by place name. Here user will be update their profile and posts. User can delete their posts. Users will be get current weather information from this project. Users get traffic jam map and current location also. Users will know all from our about page. If users need any help they can contact with us using phone number or email. It will be helpful for users.

In other side, Admin must need to login. After that Admin can view all site. Admin can create post of traffic jam. Admin can comments on post. Admin will find current location. Admin can add users and remove users. Admin can update his/her profile. Here admin will be update users profile and posts. Admin can view all posts. Admin will know the reason of traffic jam. Admin can delete their post. Admin also can delete users post. Admin will be get current weather information. Admin get traffic jam map and current location also.

## Project Purpose

This project name is “Traffic jam”. The main purpose of this project finds out the reason of traffic jam. When users will use this system they find post of traffic jam. From that post description users will find the reason of traffic jam. Users also know from that post why and when traffic jam is created.

### Background

Here I have studied some applications and websites like traffic jam. Such are:

**Live Traffic Map:**

Advantage:

* Showing “normal map” of hull world.
* Showing “satellite map” of hull world.
* It has zoomed in and out option

Disadvantage:

* Don’t take user current location.

**Traffic near Me:**

Advantage:

* Many types of maps are there.
* Traffic works perfectly.
* User can use location and traffic together or one by one.

Disadvantage:

* It takes users location for very short time.
* It has a location check box but it’s not work.

**Traffic Report:**

Advantage:

* Showing “normal map” of hull world and start with a fixed country.
* Many types of maps are there.
* Takes user location.

Disadvantage:

* It takes more time to find user location.
* Sometimes user location works sometimes it not.

**Traffic Jam Information:**

Advantage:

* It has “normal map” and “satellite map”.

Disadvantage:

* Takes long time to open.
* Not work properly.

**Traffic info and Traffic map:**

Advantage:

* It takes user location.
* It can hide user’s location.
* It provides traffic news.
* The information is updated every 3 minutes.

Disadvantage:

* It can’t store previous news.

### Benefits & Beneficiaries

Benefits of this system:

* This system helps user to make registration easily
* This system helps user to know about Traffic jam
* Users will know the reason of Traffic jam
* Users find their current location
* It’s helps user to find current weather of city
* It’s helps user to find traffic information’s using Traffic map
* Using this system, users can search post of Traffic jam by place name.
* Using this system, users can ask something about post by comment
* This system is user friendly

I have also mentioned some benefits as well as beneficiaries. So, I think this system is very much helpful for users

### Goals

After studied, I realize we don't know the reason of traffic jam. If we know the reason of traffic jam we can save our valuable time and easily take another way to reach our destination.

The main goal of my project is find out the reason of traffic jam. When users will use this system they find post of traffic jam. From that post description users will find the reason of traffic jam. Users also know from that post why and when traffic jam is created.

## Stakeholders

There are two types of stakeholders in Traffic jam. Such as:

* Users
* Admin

**User:** Users must need to authenticate for use this system. After authentication users can view all site .User can see where traffic jam is. Users will know the reason for traffic jam. Here users can create post of traffic jam with description. Users can comments on post. User can find of their current location. For this post others user easily can know where is jam and why it was created. Users can view all post. Users can search traffic jam post using by place name. Here user will be update their profile and posts. User can delete their posts. Users will be get current weather information from this project. Users get traffic jam map and current location also. Users will know all from our about page. If users need any help they can contact with us using phone number or email.

**Admin:** Admin must need to login. After that Admin can view all site. Admin can create post of traffic jam. Admin can comments on post. Admin will find current location. Admin can add users and remove users. Admin can update his/her profile. Here admin will be update users profile and posts. Admin can view all posts. Admin will know the reason of traffic jam. Admin can delete their post. Admin also can delete users post. Admin will be get current weather information. Admin get traffic jam map and current location also.

## Proposed System Model

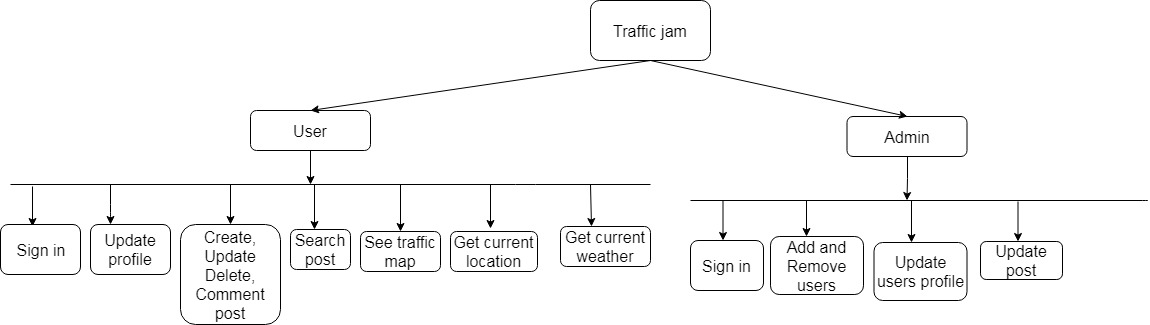


Figure 1.4: Proposed system model

## Project Schedule

Here I need to make a plan of my project schedule. Now I am going to make a chart of my project schedule.

### Gantt chart

Gantt chart is the most important part of a project. It’s containing the time table of complete a project. Now I am showing my project Gantt chart.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** |  | 1 w | 2 w | 3 w | 4 w | 5 w | 6 w | 7 w | 8 w | 9 w | 10 w | 11 w | 12 w | 13 w | 14 w | 15 w | 16 w |
| **Planning** | Ideas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Problem definition |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Problem planning |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirements | Requirement  specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requirement  analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| QA – 1 | Quality assurance |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System design | Design  specification |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Interface design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Database design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Development** | Development system modules |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Integrate system modules |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| QA - 2 | Test cases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing | Unit testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Blackbox testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Resolve Issues** | Resolve Issues  found |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Release** | Software release |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Figure 1.5.1: Gantt chart

### Release Plan/Milestone

|  |  |  |
| --- | --- | --- |
| Activities | Duration in week | Total week |
| **Ideas** | 1 week | 1 |
| Problem identification | 1 week, 2 week | 2 |
| Requirement specification | 2 week | 1 |
| Requirement analysis | 2 week, 3 week | 2 |
| **Quality assurance** | 3 week | 1 |
| Design specification | 4 week | 1 |
| Database design | 5 week | 1 |
| **Development system modules** | 5 week, 6 week, 7 week, 8 week | 4 |
| **Integrate system modules** | 6 week, 7 week, 8 week | 3 |
| Test case | 3 week, 7 week, 8 week, 9 week, 10 week | 5 |
| Unit testing | 11 week, 12 week | 2 |
| Black-box testing | 13 week, 14 week, 15 week | 3 |
| **Resolve Issues**  **found** | 14 week, 15 week | 2 |
| Software release | 16 week | 1 |

# Chapter 2: Software Requirement Specification

## Functional Requirements

Functional requirements are those which are related to the technical functionality of the system. In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

### 2.1.1. Authentication

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 01** | **Authentication** | | |
| **Description** | Users and Admin will Sign in the system using user name, password & system will verify information. If user name, password incorrect user wouldn’t Sign in this system.  If user have not an account for Sign in for that user must need to Sign up than go to Sign in. | | |
| **Stakeholders** | User, Admin. | **Priority** | High |

### 

### 2.1.2. Create Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 02** | **Create Post** | | |
| **Description** | Users can create post in this system given required information. | | |
| **Stakeholders** | User, Admin. | **Priority** | High |

### 2.1.3. Update Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 03** | **Update Post** | | |
| **Description** | Users and Admin both can Update post. Admin can update users post. Before update post Admin and Users must be logged in. | | |
| **Stakeholders** | User, Admin. | **Priority** | High |

### 2.1.4. View Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 04** | **View Post** | | |
| **Description** | Users and Admin will see post & post details where Users or Admin post in this system. Before view post users and admin must Signed in this system. | | |
| **Stakeholder** | User, Admin. | **Priority** | High |

### 

### 2.1.5. Comment on Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 05** | **Comment on Post** | | |
| **Description** | Users can Comment on Post. Before comment on post users must Signed in this system. | | |
| **Stakeholder** | User. | **Priority** | High |

### 2.1.6. Delete Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 06** | **Delete Post** | | |
| **Description** | Users and Admin both can delete there created post. Admin can delete users post. Before delete post Users and Admin must Signed in this system. | | |
| **Stakeholders** | User , Admin | **Priority** | High |

### 2.1.7. Manage User

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 07** | **Manage User** | | |
| **Description** | Admin can manage users in the system. Like Admin can add users, admin can remove users from this system. Before that Admin must Signed in this system. | | |
| **Stakeholders** | Admin | **Priority** | High |

### 

### 2.1.8. Edit and Update Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 08** | **Edit & Update Profile** | | |
| **Description** | Users & Admin both will edit and update their profile. Like name, address, phone number. Admin also edit user’s profile. But email address cannot edit ones they insert in this system. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

### 2.1.9. Search Post

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 9** | **Search Post** | | |
| **Description** | Users can search all post using by place name. | | |
| **Stakeholders** | User | **Priority** | High |

### 2.1.10. Traffic Jam map

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 10** | **Traffic Jam map** | | |
| **Description** | Users can see Traffic jam map with current location. | | |
| **Stakeholders** | User | **Priority** | High |

### 2.1.11. Current Weather

|  |  |  |  |
| --- | --- | --- | --- |
| **FR 11** | **Current Weather** | | |
| **Description** | Users and should be able to get current weather. | | |
| **Stakeholders** | User | **Priority** | High |

## Data Requirements

* Entry data from the system
* Store data requirements
* Capacity of the data requirements
* Resources of the data requirements
* Quantity of data
* Availability of data
* Data source sequence
* Data availability schedules

## Performance Requirements

Incorrectly defined performance specifications can lead to disputes Users. It is very important to maintain performance of any software system. Get better performance, here need to follow some steps. Now, I will explain performance of this project.

### Speed and Latency Requirements

Speed and latency requirements must be ensured while retrieving data from the server.

|  |  |  |  |
| --- | --- | --- | --- |
| **SLR 01** | **Search result must be faster.** | | |
| **Description** | When users search post of traffic jam using place name, then the search result must show within seconds. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

|  |  |  |  |
| --- | --- | --- | --- |
| **SLR 02** | **Map show time must be faster.** | | |
| **Description** | When users what to see traffic map the result must need to show within seconds. | | |
| **Stakeholders** | User, Admin | Priority | Medium |

### Precision or Accuracy Requirements

A result must need to be accurate. If users find out wrong information from this system, they will not use this system any more.

|  |  |  |  |
| --- | --- | --- | --- |
| PAR 01 | **Search result must be accurate** | | |
| **Description** | When users search post of traffic jam, then the search result must be according to the input value given by user. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PAR 02 | | **Current location must be accurate** | | | | | |
| **Description** | | When users want to see their current location, it must need to display accurate location. | | | | | |
| **Stakeholders** | | User, Admin | | **Priority** | | Medium | |
| PAR 03 | **Current weather must be accurate** | | | | | |
| **Description** | When users want to see their current weather of city, it must need to display accurate weather of user’s city. | | | | | |
| **Stakeholders** | User, Admin | | **Priority** | | Medium | |

### Capacity Requirements

This system must be capable to handle user data, provide accurate information, handling database etc.

|  |  |  |  |
| --- | --- | --- | --- |
| CR 01 | Capacity Requirements | | |
| **Description** | The system need to handle data thousands of data every moment. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

## Dependability Requirements

The term dependability is measured based on four dimensions. Such as:

* Reliability
* Availability
* Security
* Safety

### Reliability Requirements

Now, I will mention requirements which are related to reliability.

|  |  |  |  |
| --- | --- | --- | --- |
| RAR 01 | Reliability Requirements | | |
| **Description** | This system must be Reliable.   * The system must give accurate information of the traffic jam. It must provide a password protected login system. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

### Availability Requirements

Now, I will mention requirements which are related to availability.

|  |  |  |  |
| --- | --- | --- | --- |
| RAR 01 | Availability Requirements | | |
| **Description** | The system must be available on 24\*7.   * Our system must be available all day long, every day in a week * The system must be updated regularly | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

### Robustness or Fault-Tolerance Requirements

This system is user friendly. Its must need to ensure 0% error in this system. This system must provide accurate information’s. All users could be access in this system without any system errors. Thousands of users might be access in this system at a time. All their requests must be handled without any fault.

### Safety-Critical Requirements

I am developing my project in a language that has some inbuilt security. Otherwise this system has no specific Safety-Critical Environments.

## Maintainability and Supportability Requirements

Maintenance and Support is very important to this system.

### Maintenance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| MR 01 | **System helps to update user profile** | | |
| **Description** | It is very important to update user profile. | | |
| **Stakeholders** | Admin | **Priority** | Medium |

### Supportability Requirements

Supportability requirements may have related to some extends. Such as:

* Testability
* Install ability
* Adaptability
* Maintainability
* Compatibility
* Configurability
* Serviceability

### Adaptability Requirements

This system has no specific Adaptability Environments.

### Scalability or Extensibility Requirements

Scalability means how easily this system will be able to deal with growing user base / additional data, etc. Extensibility means how easily this system can support for new functionalities, interfaces, devices, input types etc.

## Security Requirements

Security is very important for software. Software security requirements should be its functional requirement. Software security provides security of this system.

### Access Requirements

To access this system, all users must need to be authenticated. Without authentication users can’t visit any site of this system. Here I give an example below:

|  |  |
| --- | --- |
| SSR 01 | **Provides** Security |
| **Description** | Users will access this system after authentications. Only one way users must need to authentication first. |
| **Stakeholders** | User, Admin |

### Integrity Requirements

Integrity requirements give a system a security. This will protect all the data on a system. This ensures that all data on the system is never exposed. For this, we will save user passwords as an encrypted format, which would be impossible to decrypt?

### Privacy Requirements

Every system has some privacy. It is very important to ensure privacy of the system. Privacy requirements ensure users privacy. All data are store in database. To ensure privacy, database should be protected by the anonymous. Users will be able to access the information, but they need to be authenticated.

## Usability and Human-Interaction Requirements

All user wants to use a system very easy way. Now main target of this system is to make the system user friendly and easy to usable for all users.

### Ease of Use Requirements

Ii is easy to use and also easily understandable.

|  |  |  |  |
| --- | --- | --- | --- |
| EUR 01 | Must be usable for the users. | | |
| **Description** | This system is user friendly and very easy to use. | | |
| **Stakeholders** | User, Admin | **Priority** | Medium |

### Personalization and Internationalization Requirements

This system has no Personalization and Internationalization requirements. Now this system is useable only for Bangladesh.

### Understandability and Politeness Requirements

When I will make this system its will be user friendly and very easy to use. Here I am ensure that, this system will be enough understandable.

### Accessibility Requirements

If any user wants to access any part of this system then they must need to be authenticated user. Otherwise there is no specific accessibility of this system.

### User Documentation Requirements

This system is mainly two types. One is internal documentation which is generally written by the application engineers. It is prepared to make development life cycle easier for the system engineers or system analysts.

### Training Requirements

After developed a system Training requirements is very necessary for any user. How we would use this system, it does will be first question from end users. How do users use this system, it does will be first question from developers. To solve this problem Training is the most important. After delivered this system user will get trained from us. After trained, it will be very easier to use this system for users.

## Look and Feel Requirements

It’s refers what will be the system look like? How to do display users the system interface?

### Appearance Requirements

All user must know which input fields are required and which are not required. For that, I will all input fields. Input fields might be TextField, CharField, Checkbox, required etc. Here I ensure that all input fields must have Placeholder to make it easier for the users.

### Style Requirements

It's very important to load style part of this system. This system should be web base system. After successfully use style in this system, users could be use this system in phone browser.

## Operational and Environmental Requirements

It’s refers:

* Performance
* Capabilities
* Measurements
* Measurements of effectiveness
* Measurements of performance
* Process
* Technical performances etc.

### Expected Physical Environment

This system has no specific Physical Environments.

### Requirements for Interfacing with Adjacent Systems

This system has no Requirements for Interfacing with Adjacent Systems.

### Projectization Requirements

This system has no Projectization Requirements.

### Release Requirements

This system has no specific release requirements.

## Legal Requirements

It’s referring the privacy policy of any Organizations. The terms and condition of this system, there is no third party software or person is allowed to engage to use system data for their business purpose.

### Compliance Requirements

This system has no specific Compliance requirements.

### Standards Requirements

This system has no specific Standards requirements.

# 

# Chapter 3: System Analysis

## Use Case Diagram

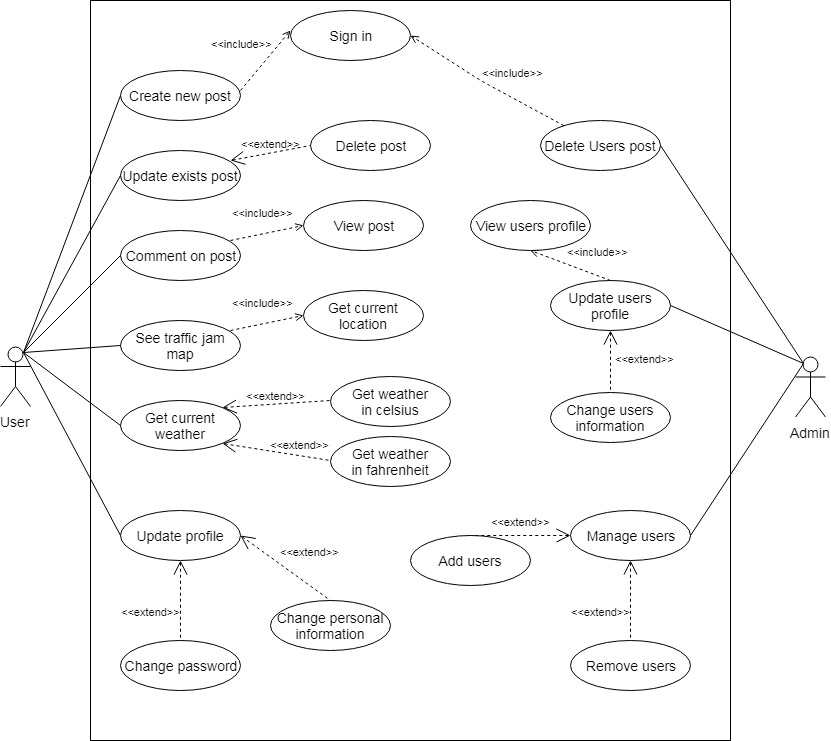


Figure 3.1: Use Case Diagram for the project

## Use Case Description

### 3.2.1. Authentication

|  |  |  |
| --- | --- | --- |
| **Use case** | **Authentication** | |
| **Use case no.** | 01 | |
| **Goal** | It will be the sign in or sign up part for this system. | |
| **Pre-condition** | To access this system users must need to be sign in. | |
| If not have an account to sign in then go for signs up first. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text boxes and buttons | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Give user id and password |
| 2. | Signed in |
| 3. | If users have no account goes to sign up first and fill all required fields. After that go to sign in. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.2. Create new post

|  |  |  |
| --- | --- | --- |
| **Use case** | **Create new post** | |
| **Use case no.** | 02 | |
| **Goal** | Users could be create a new post of this system. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text boxes and buttons. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Go to post |
| 2. | Fill up all required fields |
| 3. | Click submit button to successfully create post. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.3. Update Exists post

|  |  |  |
| --- | --- | --- |
| **Use case** | **Update exists post** | |
| **Use case no.** | 03 | |
| **Goal** | Uses could be update post of this system. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text boxes and buttons. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click edit button. |
| 2. | Fill up all required fields |
| 3. | Click submit button to successfully edited post. |
| **Alternative flows** | No Alternative flows | |
| **Quality requirement** | No Quality requirement | |

### 3.2.4. Comment on post

|  |  |  |
| --- | --- | --- |
| **Use case** | **Comment on post** | |
| **Use case no.** | 04 | |
| **Goal** | Users could be comment on post of this system. For that user must need to check the post. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text boxes and buttons. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click comment button. |
| 2. | Fill up all required fields |
| 3. | Click submit button to successfully comment on post. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.5. See traffic jam map

|  |  |  |
| --- | --- | --- |
| **Use case** | **See traffic jam map** | |
| **Use case no.** | 05 | |
| **Goal** | Users can see traffic jam map with their current location of this system. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be Traffic map buttons of this system. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click Traffic map button. |
| 2. | Successfully show traffic jam map. |
|  |  |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.6. Get current weather

|  |  |  |
| --- | --- | --- |
| **Use case** | **Get current weather** | |
| **Use case no.** | 06 | |
| **Goal** | Users will get current city weather of users in this system. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text box and weather button of this system.  Celsius and Fahrenheit button also. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click weather button. |
| 2. | Type city in box and search. Two way to search city weather  Celsius and Fahrenheit. |
| 3. | Successfully show current city weather of users. |
|  |  |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.7. Update profile

|  |  |  |
| --- | --- | --- |
| **Use case** | **Update profile** | |
| **Use case no.** | 07 | |
| **Goal** | Users update their personal information and change password in this system. | |
| **Pre-condition** | Users must have to authenticate. | |
| **Primary actors**  **Secondary actors** | User  Admin | |
| **Trigger** | This will be text boxes and buttons. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click update profile button. |
| 2. | Perform the changes on box |
| 3. | Click submit button for save. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.8. Delete users post

|  |  |  |
| --- | --- | --- |
| **Use case** | **Delete users post** | |
| **Use case no.** | 08 | |
| **Goal** | Admin could be Delete post of this system. Before that admin must need to be signed in. | |
| **Pre-condition** | Admin must have to authenticate. | |
| **Primary actors**  **Secondary actors** | Admin  None | |
| **Trigger** | This will be delete button. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click delete button. |
| 2. | Successfully delete post. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 

### 3.2.9. Update Users profile

|  |  |  |
| --- | --- | --- |
| **Use case** | **Update profile** | |
| **Use case no.** | 09 | |
| **Goal** | Admin can update user’s information in this system. Before that admin must need to view user’s profile. | |
| **Pre-condition** | Admin must have to authenticate. | |
| **Primary actors**  **Secondary actors** | Admin  None | |
| **Trigger** | This will be text boxes and buttons. | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click update profile button. |
| 2. | Perform the changes on box |
| 3. | Click submit button for save. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

### 3.2.10. Manage user

|  |  |  |
| --- | --- | --- |
| **Use case** | **Manage user** | |
| **Use case no.** | 10 | |
| **Goal** | Admin could be Add and Delete users of this system. | |
| **Pre-condition** | Admin must have to authenticate. | |
| **Primary actors**  **Secondary actors** | Admin  None | |
| **Trigger** | This will be text boxes and buttons | |
| **Description / Main success** | **Step** | **Action** |
| 1. | Click add user button. |
| 2. | Give users information. |
| 3. | Successfully add user. |
| 4. | For delete ,click delete button |
| 5 | Successfully delete user. |
| **Alternative flows** | N/A | |
| **Quality requirement** | N/A | |

## Activity Diagram

I have prepared some activity diagram according to Use Case. These activity diagrams are properly referring the flow of the individual conditions of my project.

### 3.3.1. Authentication

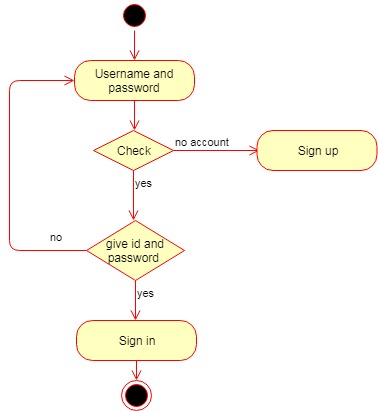
****

Figure 3.3.1: Activity Diagram for Authentication

### 3.3.2. Create new post

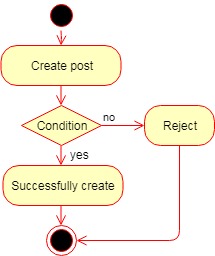
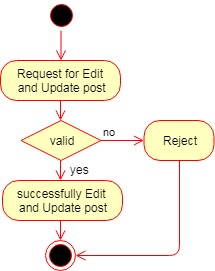
****

Figure 3.3.2: Activity Diagram for Create New Post

### 3.3.3. Update Exists post

****

### Figure 3.3.3: Activity Diagram for Update exists post

### 3.3.4. Comment on post

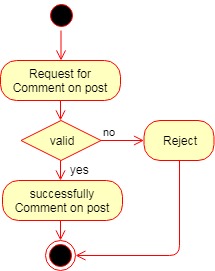
****

Figure 3.3.4: Activity Diagram for Comment on post

### 3.3.5. See traffic jam map

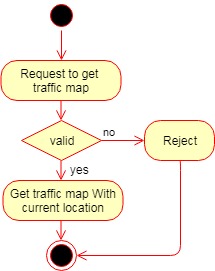
****

Figure 3.3.5: Activity Diagram for See Traffic Jam

### 3.3.6. Get current weather

****

Figure 3.3.6: Activity Diagram for Get current weather

### 3.3.7. Update profile

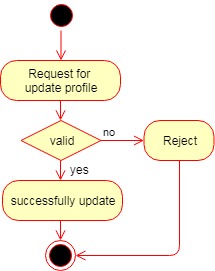
****

Figure 3.3.7: Activity Diagram for Update profile

### 3.3.8. Delete Users post

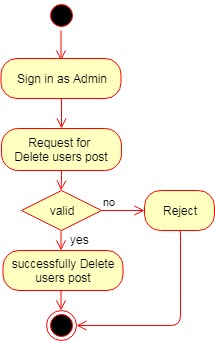


Figure 3.3.8: Activity Diagram for Delete users post

### 3.3.9. Update users profile

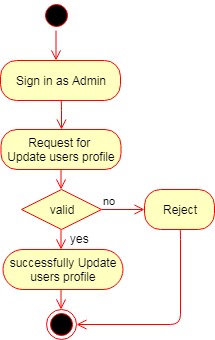
****

Figure 3.3.3: Activity Diagram for Update user’s profile

### 3.3.10. Manage User

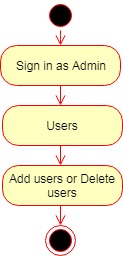
****

Figure 3.3.3: Activity Diagram for Manage User

## System Sequence Diagram

Mainly sequence diagrams understand us how the data will be followed in any application.

Now I am going to show some sequence diagrams.

### 3.4.1. Authentication

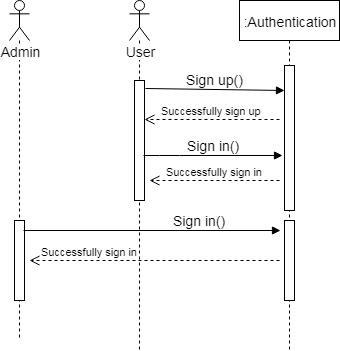
****

Figure 3.4.1: Sequence diagram for Authentication

### 3.4.2. Create new post

****

Figure 3.4.2: Sequence diagram for Create new post

### 3.4.3. Update Exists post

****

Figure 3.4.3: Sequence diagram for Update Exists post

### 3.4.4. Comment on post

****

Figure 3.4.4: Sequence diagram for Comment on post

### 3.4.5. Get traffic map

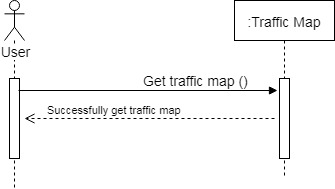
****

Figure 3.4.5: Sequence diagram for Get traffic map

### 3.4.6. Get current weather

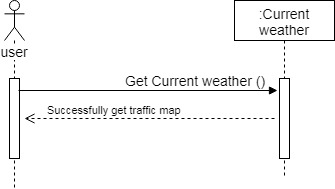
****

Figure 3.4.6: Sequence diagram for Get current weather

### 3.4.7. Update profile

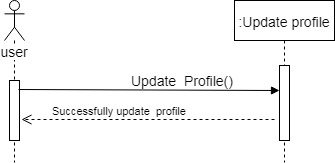
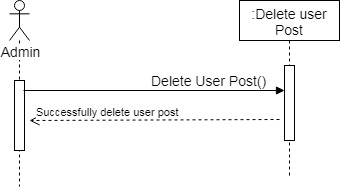
****

Figure 3.4.7: Sequence diagram for Update profile

### 3.4.8. Delete user’s post

****

### Figure 3.4.8: Sequence diagram for Delete user’s post

### 3.4.9. Update user’s profile

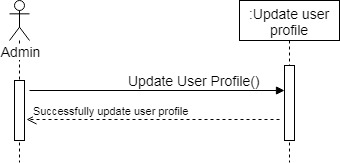
****

Figure 3.4.9: Sequence diagram for Update user’s profile

### 3.4.10. Manage user

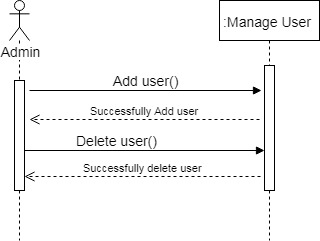
****

Figure 3.4.10: Sequence diagram for Manage user

### 

# Chapter 4: System Design Specification

## Class Responsibilities Collaboration (CRC) Cards

### 4.1.1. User

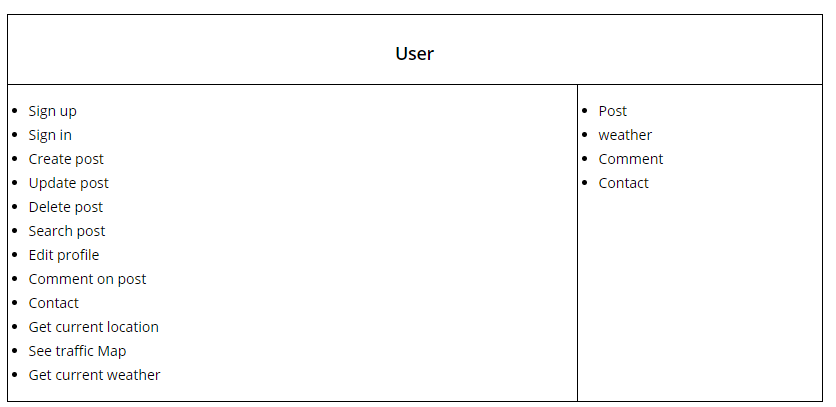


Figure 4.1.1: CRC for User

### 4.1.2. Admin

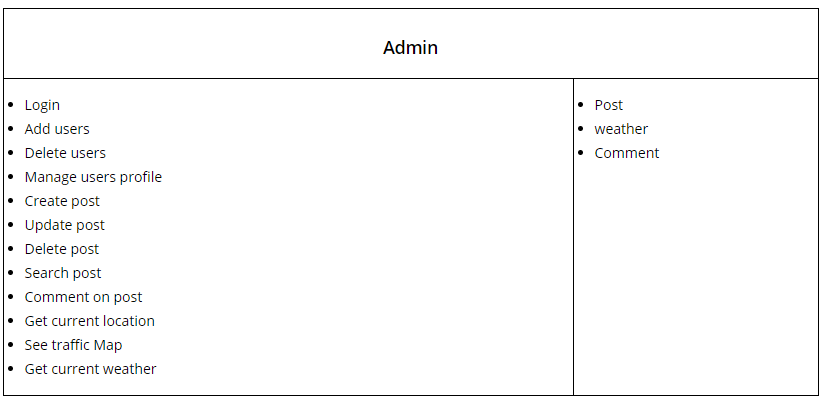


Figure 4.1.2: CRC for Admin

### 4.1.3. Weather

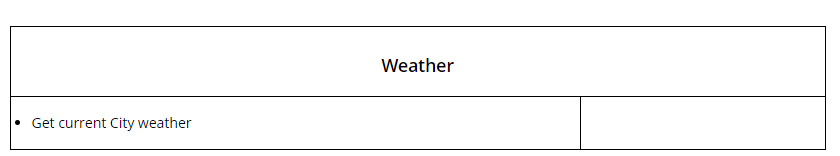


Figure 4.1.3: CRC for Weather

### 4.1.4. Post

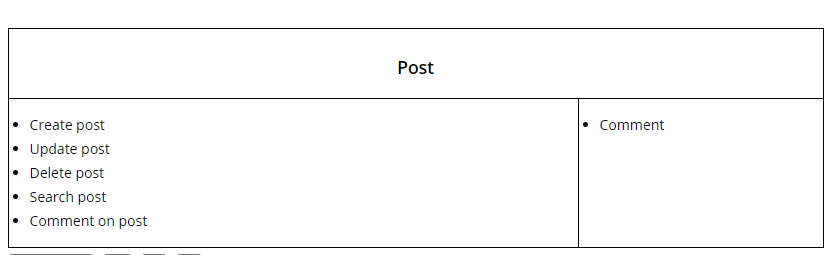


Figure 4.1.4: CRC for Post

### 4.1.5. Comment

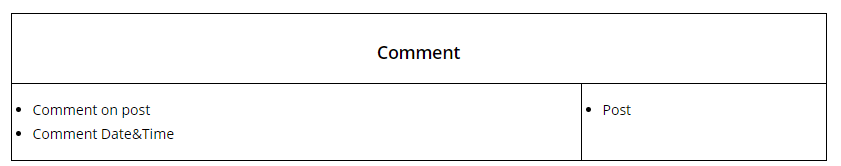


Figure 4.1.5: CRC for Comment

### 4.1.6. Contact

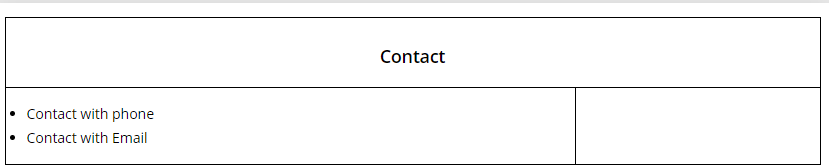


Figure 4.1.6: CRC for Contact

## Class Diagram

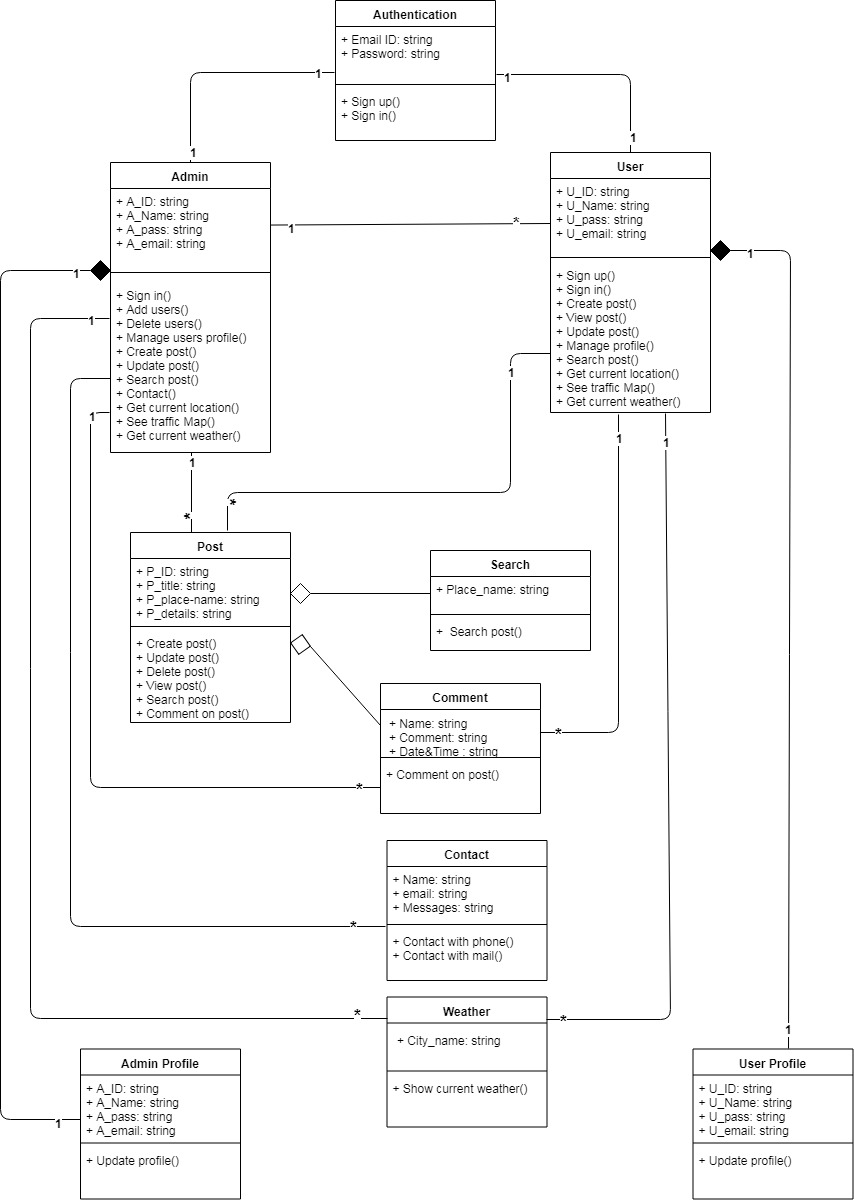


Figure 4.2: Class diagram for this project

## Database Design Diagram

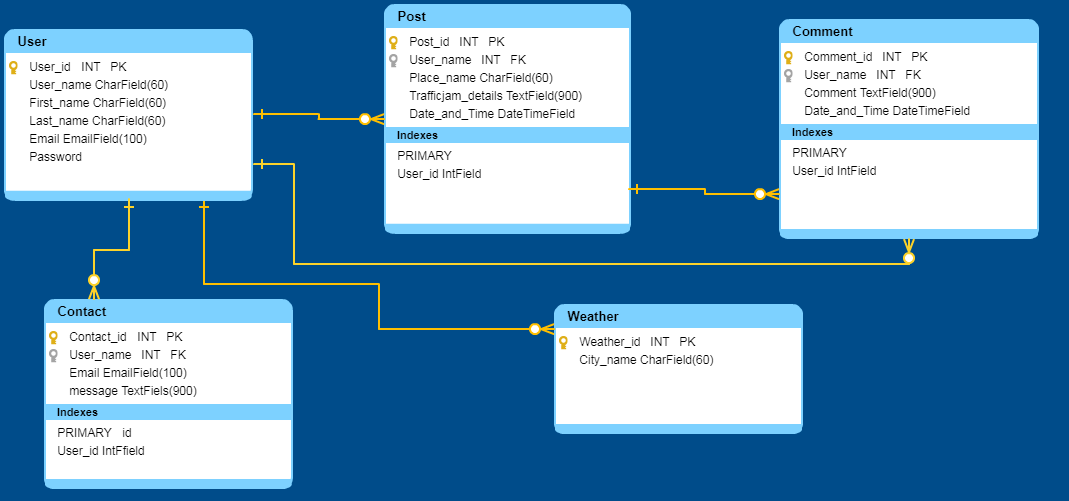


Figure 4.3: Database Design diagram for this project

## Development Tools & Technology

Software development tools are used to develop software. Now I am telling here what kind of tools and technology will use in this system.

### User Interface Technology

First of all, after landing any web site, interface of that software is appeared to the user. So, The importance of user interface is very high. For success of any software application, a good Looking user interface plays a vital role. User interface includes using good image, graphics, Typography, style sheets, scripting etc.

#### 4.5.1.1 Django Framework

For developing any application system minimum one programming language is essential. Django is a web Framework of Python language. In this project I am go to use Django Framework.

#### 4.5.1.2 jQuery Framework

JQuery UI is library name which uses JavaScript as core programming language. It simplifies codes of a programming language named JavaScript. Generally, it interacts with Graphical User Interface (GUI). It also provides visual effects with animation. It is also a subset of JavaScript. By using AJAX no page reloading is required. In this project I use jQuery Framework. It is also compatible with any browser like Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer etc.

#### 4.5.1.3 CSS Framework or Bootstrap

Bootstrap also provides some JavaScript components also. There are some built in components

Like Query UI. By using Bootstrap framework, we get both Cascading Style Sheets (CSS) and JavaScript facilities with a single platform. But before start designing an application interface with Bootstrap, one may have some basic knowledge about this framework. It will increase the efficiency.

### Implementation Tools & Platforms

Now I am telling here what kind of tools and Platforms will use in this system.

#### 4.5.2.1 Integrated Development Environment (IDE)

IDE stands for Integrated Development Environment. Programmers write code on IDE. After that IDE provide the feature to execute the source code. For developing my web site, I have used an IDE. For developed my project here I use “PyCharm” IDE.

#### 4.5.2.2 Database

Database is the main part of a project. All data of this project would store in database. Here I am use sqlite3 database. Django carries this database, you don’t need to add or anything it. It is also very easy to use. It can also ensure the security, scalability, high performance and many things.

# Chapter 5: System Testing

## Testing Features

Feature testing can be considered as making change to add or modify the new functionality to the existing project. To test the features and functionality, a new test set is to be written for testing purpose. Almost every feature and functionality has different characteristics. Those are designed to make the application more useful, intuitive, reliable, secured, scalable, effective and efficient.

### Features to be tested

|  |  |  |
| --- | --- | --- |
| **Features** | **Priority** | **Description** |
| Authentication | High Priority | User must be authenticated by sign in. Session must be destroyed after sign out. |
| Insert post data | Medium Priority | Post data must be inserted properly |
| Current location | High Priority | Ensure that uses get current location properly |
| Traffic map | High Priority | Ensure that uses can see Traffic map properly |
| **Technological Features** | | |
| Database | Low Priority | Database will be used at almost every operation. So this is why, this part must be controlled tightly. |

## 5.1.2. Features not to be tested

Here we must need to see what is 'not' to be tested from both the user's viewpoint of what the system does and a configuration management/version control view.

* Users interface
* Users personal record
* Post publish time



## Testing Strategies

Testing strategy is to be considered as a general purpose of testing process. It is also the indicator of test levels that are to be performed on the whole software development life cycle. Those strategies that are prepared by the quality assurance team should be reviewed by the developers of the system. After that it should be reviewed by the test team leads. Different kinds of testing strategies can be performed according to the type of system that needs to be tested.

### Test Approach

To complete the test process, testers must take some approach. There is mainly two test approach.

* **Automation testing:** Automation testing is a name of testing technique by which test engineers prepare some scripts according to test plan and after that they use suitable tools to perform testing of the software. Nowadays, almost every software company follows the approach of automation testing.
* **Manual testing:** Manual testing is also a name of technique of testing by searching out the bugs or vulnerability in a system. In this process, test engineers manually test and execute the test cases without having any automation tools.

### 5.2.1.1. Black Box Testing

Black box testing is a software testing method by which test engineers test the software without having knowledge of the internal architecture of the system that need to be tested. It is also known as behavioral testing. Black box testing can be both functional and non-functional. It Ignores the internal mechanism of a system.

### 5.2.1.2. White Box Testing

White box testing is also a name of testing approach which is also known as clear box testing,

glass box testing, open box testing, transparent box testing, code based testing or structural

testing. It is opposite to the black box testing.

White box testing can be classified into some levels. Such as:

* Unit Testing
* Integration Testing
* System Testing

The main advantage of white box testing is that testing is more throughout and the testing can

be started from the very beginning stage.

### Pass/Fail Criteria

Pass or fail criteria will be set by the test engineers. They will prepare the pass / fail criteria on

the basis of which input data are worked and which are not works well. Those data that are

worked well will considered as pass criteria. And rest of input data will be considered as fail criteria.

Now I will give the pass / fail criteria below.

* System crash will not be considered as pass case.
* If any criteria pass 100% times, then it will be considered as pass criteria only.
* If data can’t be displayed to the system properly, then it is also to be considered as fail criteria.

### Suspension and Resumption

Suspension and Resumption is very need of testing. Suspending a testing process takes place when there is a need to fix a defect in s system.

### Testing Schedule

|  |  |
| --- | --- |
| **Test Phase Time** | **Time** |
| Testing plan create | 7 days |
| Test specification | 7 days |
| Unit testing | During development time |
| Component test | 7 days |
| Integration testing | 7days |
| Validating use cases | 5days |
| Testing user interfaces | 10days |
| Load testing | 4days |
| Performance testing | 10days |
| Release to production | 5-6 days |

### Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Manager | | | Business Analyst Lead | | |
| QA Lead | | | Target Implementation Date | | |
|  | Category / Functionality / Activity | Requirement  Description | Use Case Reference | Test Case Reference | Comments |
| 1 | Functional | Manage User | Use case 3.2.10 | Test case 5.3.2 |  |
| 2 | Functional | Create new post | Use case 3.2.2 | Test case 5.3.3 |  |
| 3 | Functional | Comment on post | Use case 3.2.4 | Test case 5.3.4 |  |
| 4 | Functional | Get traffic jam map | Use  Case 3.2.5 | Test case 5.3.5 |  |
| 5 | Functional | Get current weather | Use case 3.2.6 | Test case 5.3.6 |  |
| 6 | Non­  functional | Reliability | No | No |  |

## Testing Environment (hardware/software requirements)

Testing environment means to prepare the environment with hardware and software so that test engineers can be able to execute test cases as required. Besides hardware and software usage, network configuration might be needed to execute test plans.

For making the environment for testing, some key area need to setup. Those are:

* Test data
* Database server
* User’s operating system
* Front site running environment
* Browser
* System
* Network
* Hardware with server operating system
* Documentation is also required. Like: user manuals, installation guides, configuration guides, documents etc.

## Test Cases

For testing this system, I have prepared some test cases. Now, I am going to provide them.

### 5.3.1. Authentication

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test case 1 | | | | Test case name: Authentication | | |
| System: Traffic jam | | | | Subsystem: | | |
| Designed by: Partha Paul | | | | Designed date: 10/4/19 | | |
| Executed by: | | | | Executed date: | | |
| Short description: It is the registration part for this system. | | | | | | |
| Pre-conditions:   * If have an account then go to login. * If not have an account then stays here and signs up first. | | | | | | |
|  | | | | | | |
| Step | **Name** | **Email** | Password | Expected  result | Pass/Fail | Comment |
| 1 | p | p@ | 001 | Invalid | Pass |  |
| 2 | Partha | [Parth743@gmail.com](mailto:Parth743@gmail.com) | Parthapaul123 | Successfully signed up and go to sign in | Pass |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### 5.3.2. Manage User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case 2 | | | Test case name: Manage User | | |
| System: Traffic jam | | | Subsystem: | | |
| Designed by: Partha Paul | | | Designed date: 10/4/19 | | |
| Executed by: | | | Executed date: | | |
| Short description: Admin could be Add and Delete users of this system. | | | | | |
| Pre-conditions:   * Admin must have to authenticate. | | | | | |
| Step | Action | Response | | Pass/Fail | Comment |
| 1 | Users all information’s is not provided | System need all required data | | Pass |  |
| 2 | Users all information’s is provided properly | Data saved successfully | | Pass |  |
| 3 | Click delete user button | Successfully delete | | Pass |  |

### 

### 5.3.3. Create new post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case 3 | | | Test case name: Create New post | | |
| System: Traffic jam | | | Subsystem: | | |
| Designed by: Partha Paul | | | Designed date: 10/4/19 | | |
| Executed by: | | | Executed date: | | |
| Short description: Uses could be create a new post of this system. | | | | | |
| Pre-conditions:   * Users must have to authenticate. | | | | | |
| Step | Action | Response | | Pass/Fail | Comment |
| 1 | Go to post and fill not required fields | System need  All required data | | Pass |  |
| 2 | Go to post and fill up all required fields | Data saved successfully | | Pass |  |

### 

### 5.3.4. Comment on post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case 4 | | | Test case name: Comment on post | | |
| System: Traffic jam | | | Subsystem: | | |
| Designed by: Partha Paul | | | Designed date: 10/4/19 | | |
| Executed by: | | | Executed date: | | |
| Short description: Uses could be comment on post of this system | | | | | |
| Pre-conditions:   * Users must have to authenticate. | | | | | |
| Step | Action | Response | | Pass/Fail | Comment |
| 1 | Go to comment and fill not required fields | System need  All required data | | Pass |  |
| 2 | Go to comment and fill all required fields | Comment on post successfully | | Pass |  |

### 5.3.5. Get Traffic jam map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case 5 | | | Test case name: Get Traffic jam map | | |
| System: Traffic jam | | | Subsystem: | | |
| Designed by: Partha Paul | | | Designed date: 10/4/19 | | |
| Executed by: | | | Executed date: | | |
| Short description: Uses will get traffic jam map with current location of this system. | | | | | |
| Pre-conditions:   * Users must have to authenticate. | | | | | |
| Step | Action | Response | | Pass/Fail | Comment |
| 1 | Click map button and show traffic jam map. | Show current traffic jam map successfully | | Pass |  |

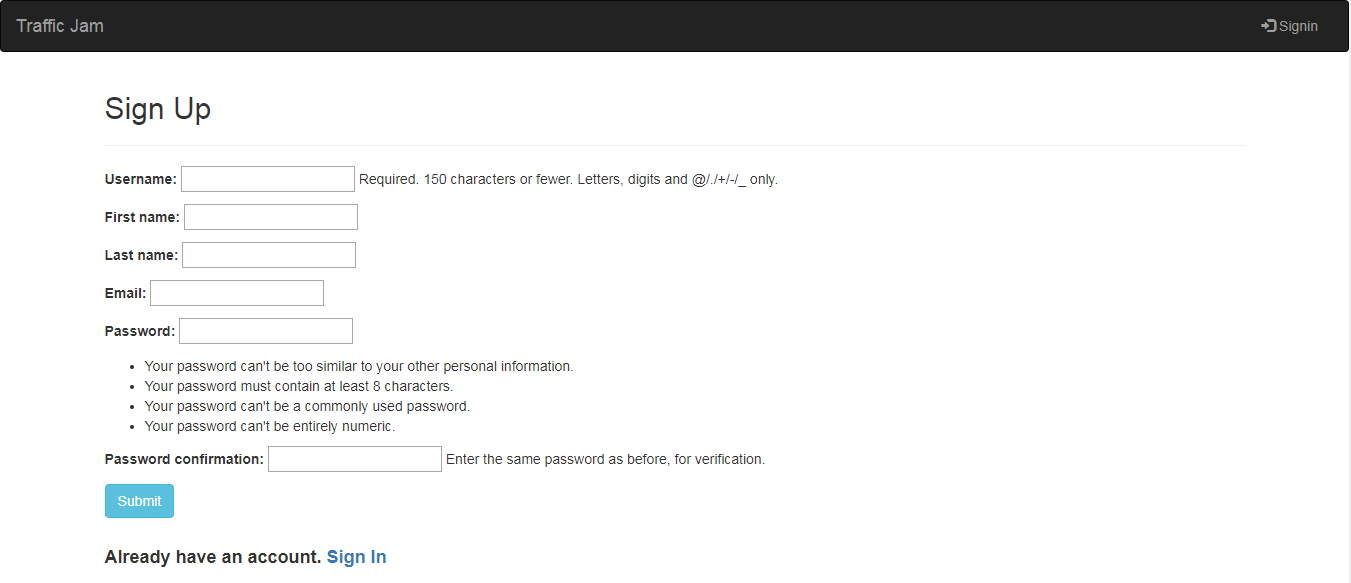
### 5.3.6. Get current weather

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case 6 | | | Test case name: Get current weather | | |
| System: Traffic jam | | | Subsystem: | | |
| Designed by: Partha Paul | | | Designed date: 10/4/19 | | |
| Executed by: | | | Executed date: | | |
| Short description: Uses will get to see current city weather of users in this system. | | | | | |
| Pre-conditions:   * Users and Admin must have to authenticate. | | | | | |
| Step | Action | Response | | Pass/Fail | Comment |
| 1 | Don’t use city name in search box. | System need city name | | Pass |  |
| 2 | Use city name in search box. | Show current weather result successfully | | Pass |  |

# Chapter 6: User Manual

## 6.1. Authentication

To access this system user must need an account. For create account user must need sign up first. After Sign up user need to sign in to access this system. Without sign in they can’t access in this system.



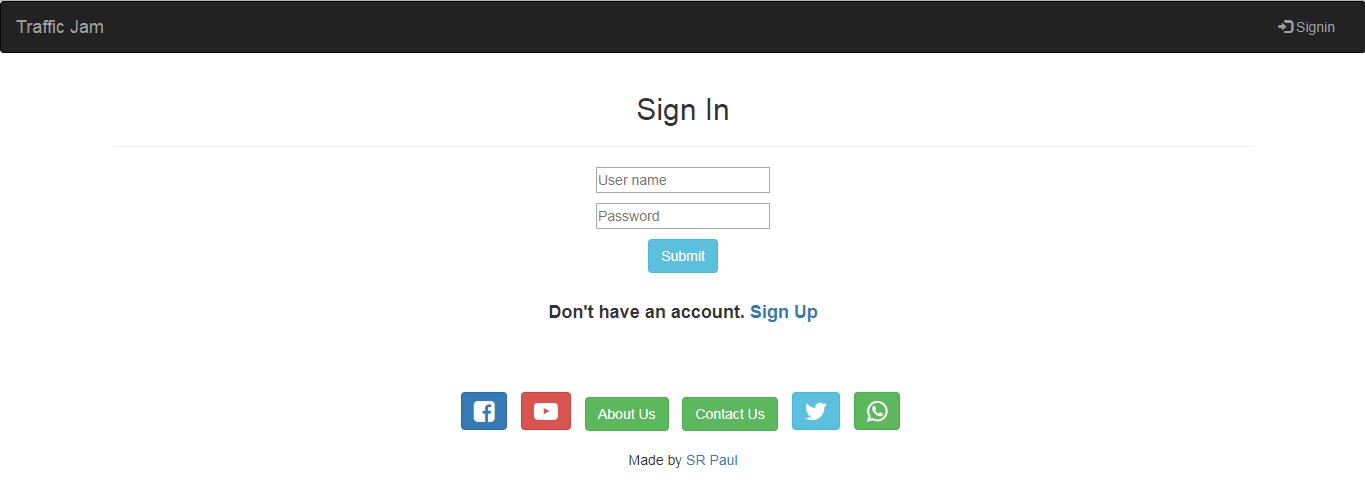


Figure 6.1: User manual for Authentication

## 6.3. Traffic jam first look

This is the first look of Traffic jam project. Here I set a picture of a Traffic jam.

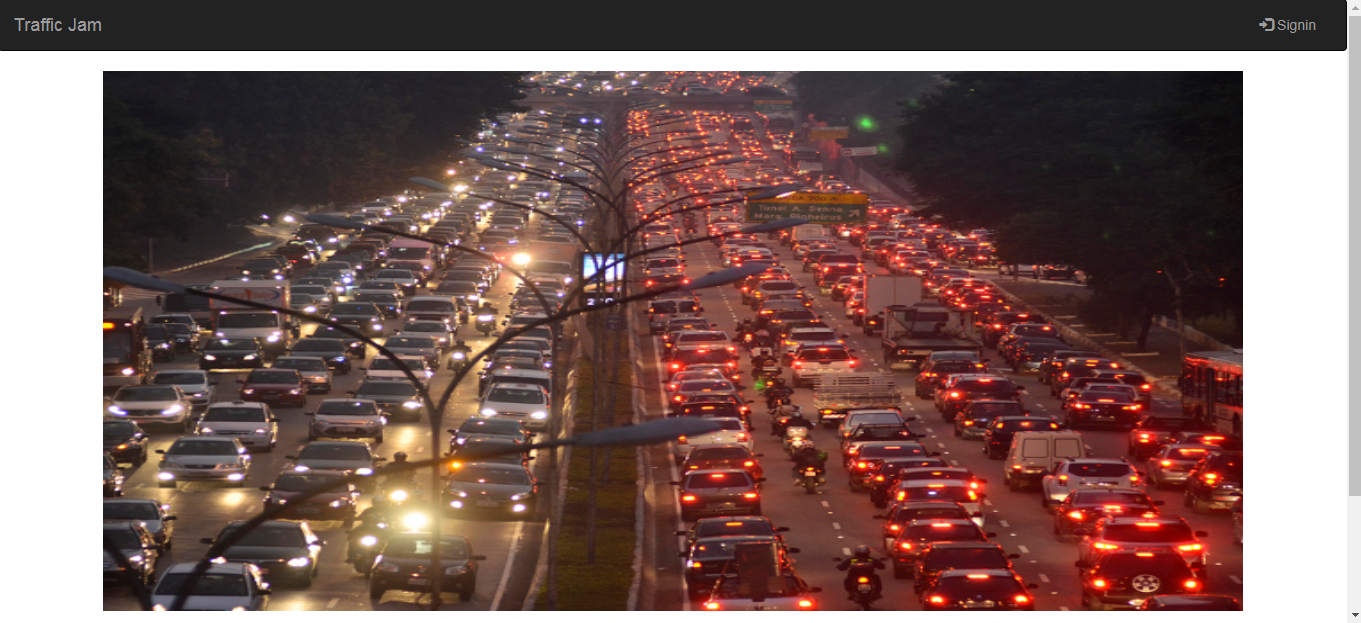


Figure 6.3: User manual for Traffic jam first look

## 6.4. Home

This is the home page of this system. Here users can see the list of post. After that user find comment, edit and delete button here.

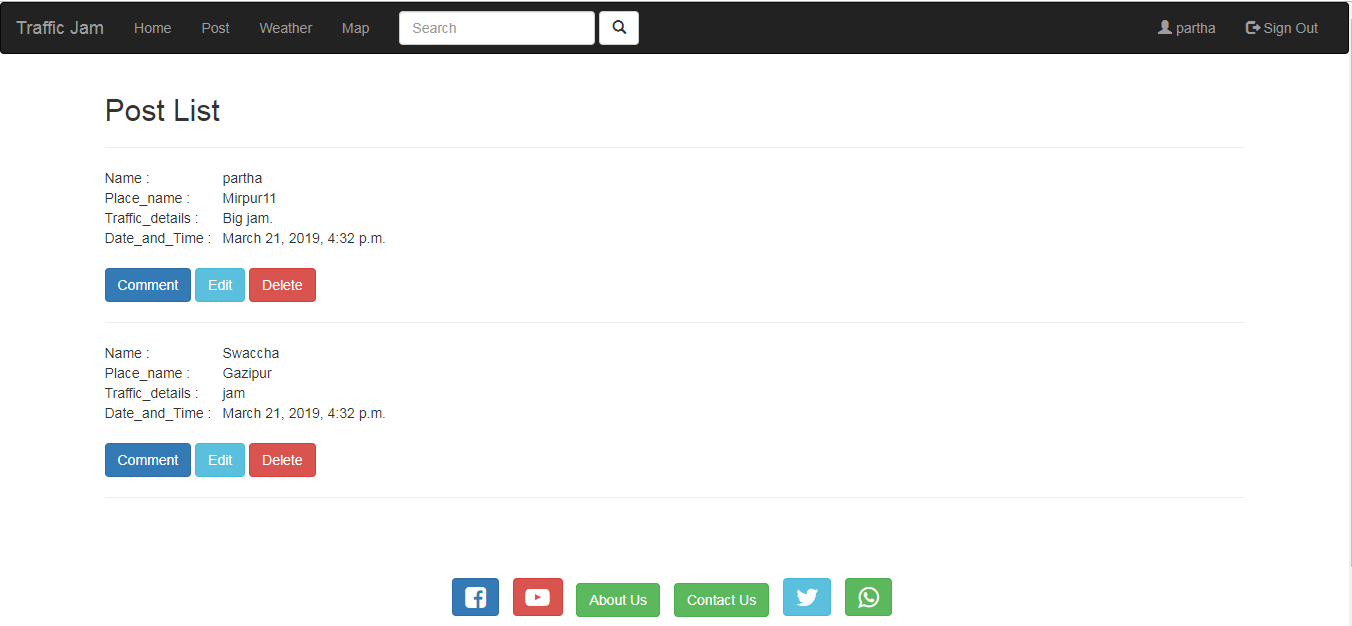


Figure 6.4: User manual for Home

## 6.5. Profile

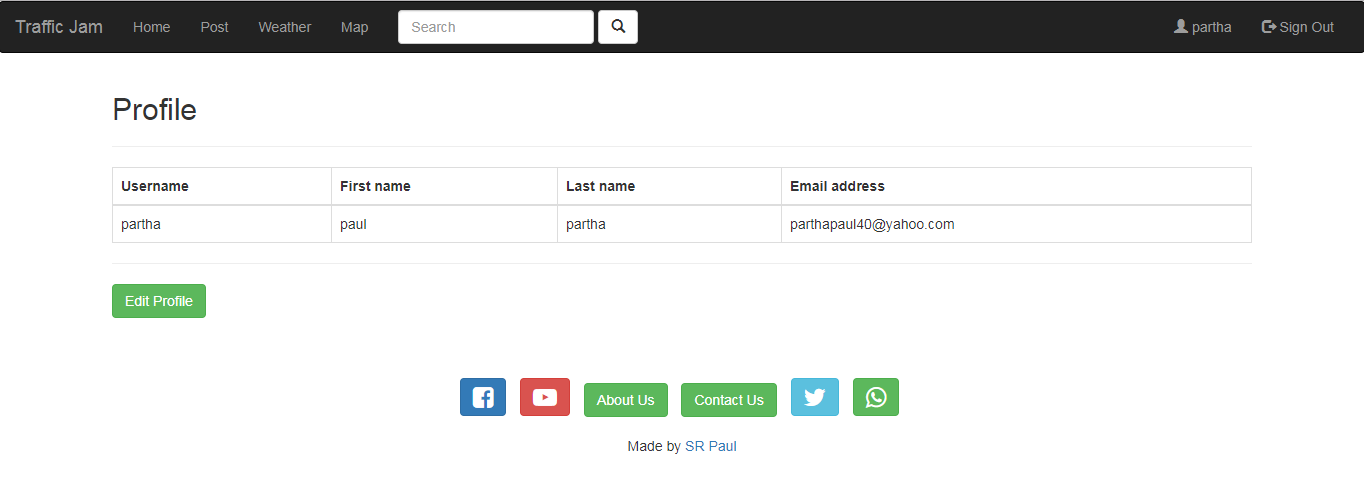
It is user’s profile. It’s containing user’s information’s.  


Figure 6.5: User manual for Profile

## 6.6. Edit profile

User can edit their personal information’s. When they click submit button then its show user profile page.

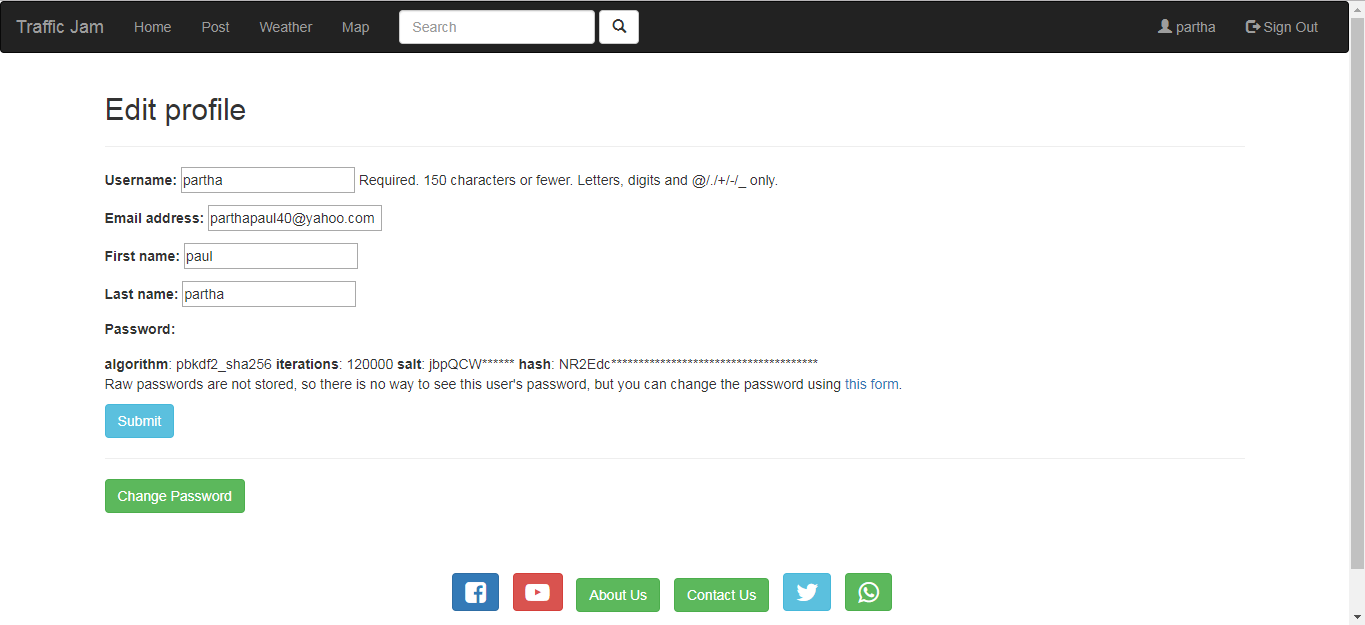


Figure 6.6: User manual for Edit Profile

## 6.7. Change password

If users want they can change their password.

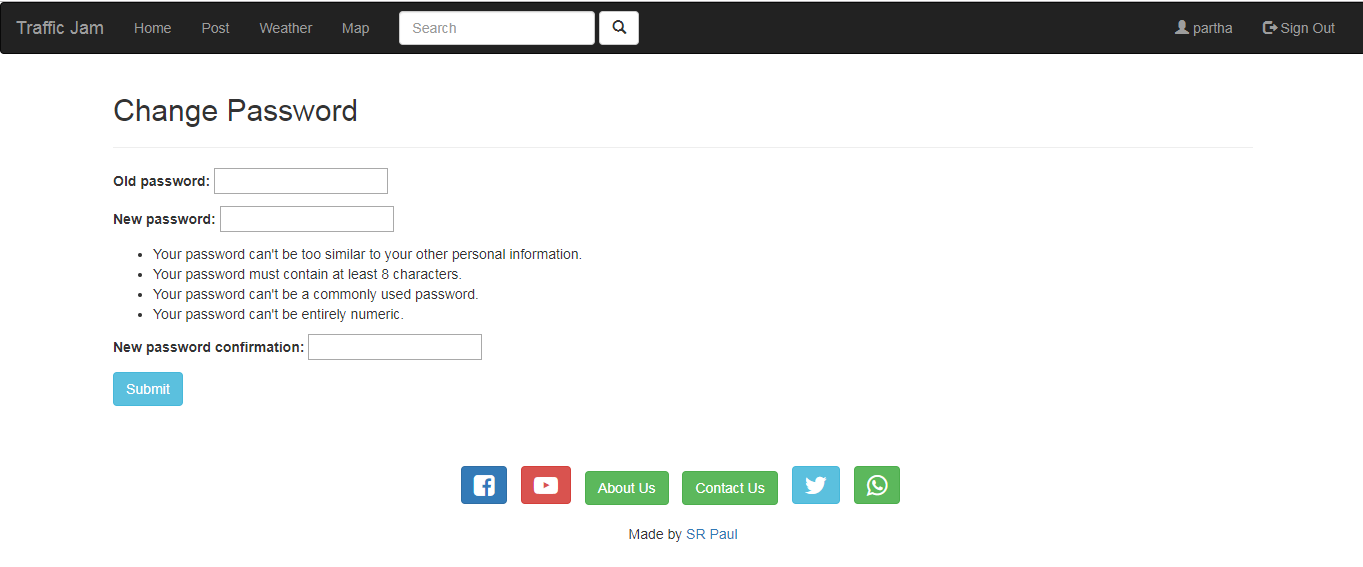


Figure 6.7: User manual for Change Password

## 6.8. Create post

Users can create a new post here about traffic jam with description.

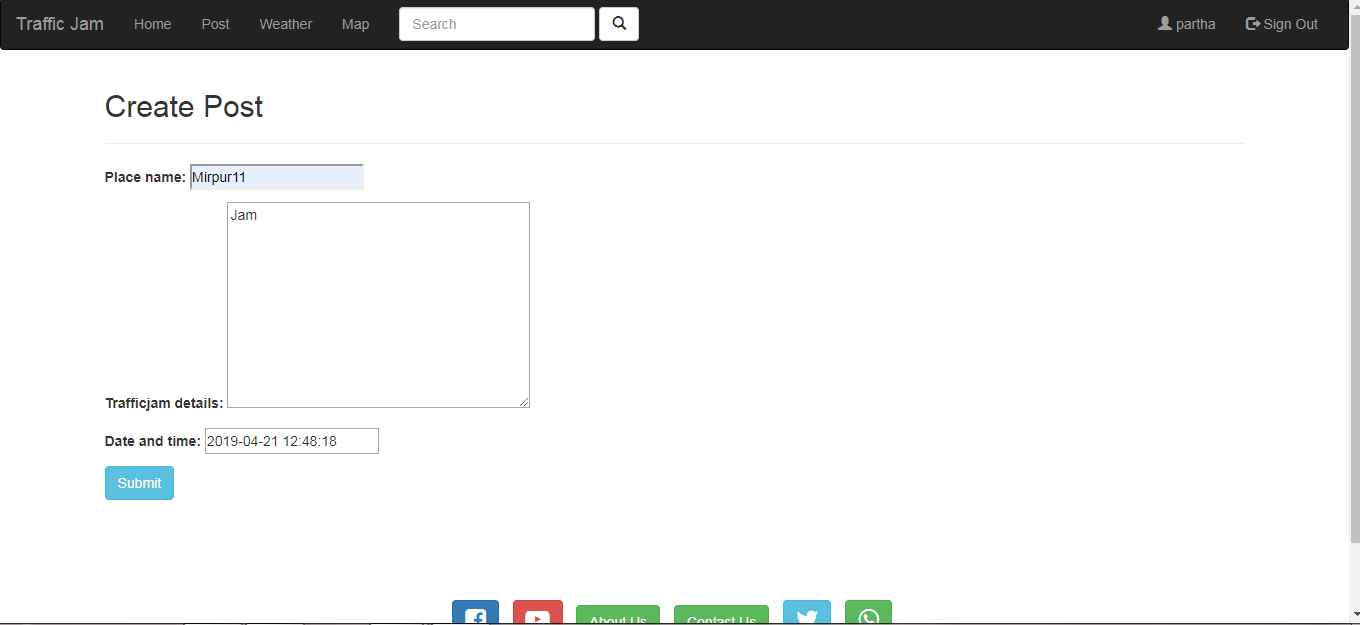


Figure 6.8: User manual for Create post

## 6.9. Comment on post

If users want they can comment on post.

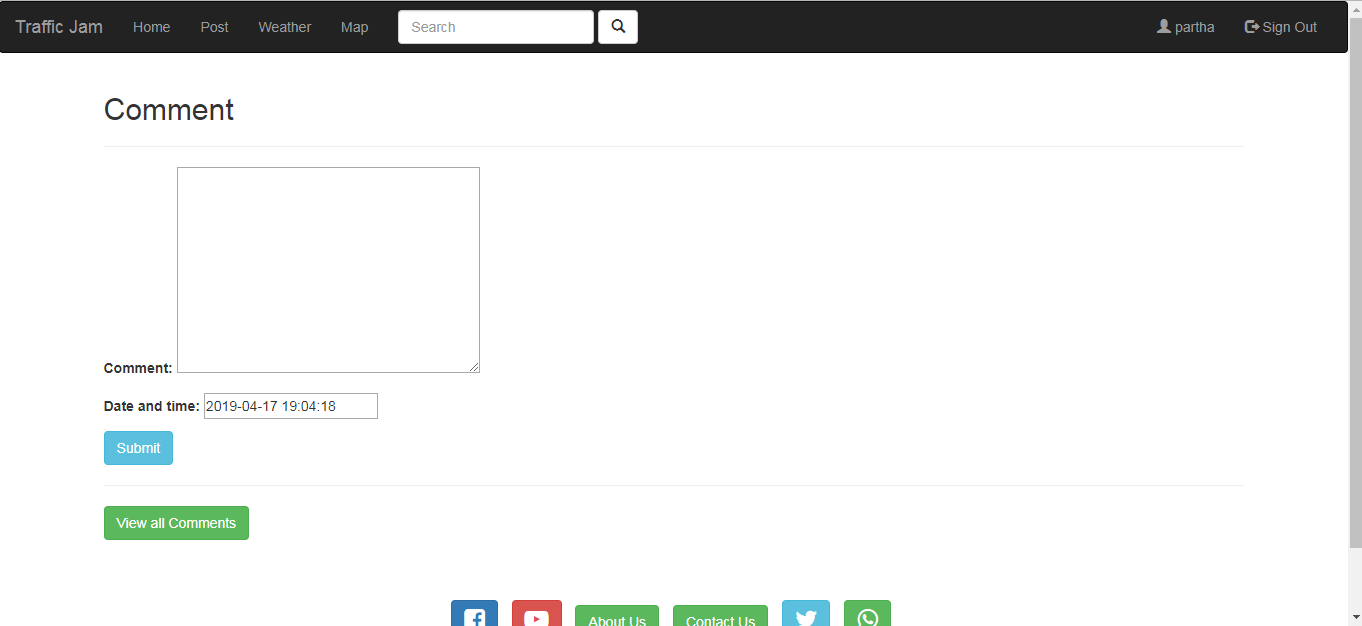


Figure 6.9: User manual for Comment on post

## 6.10. View all comments

After comment user can see all post.

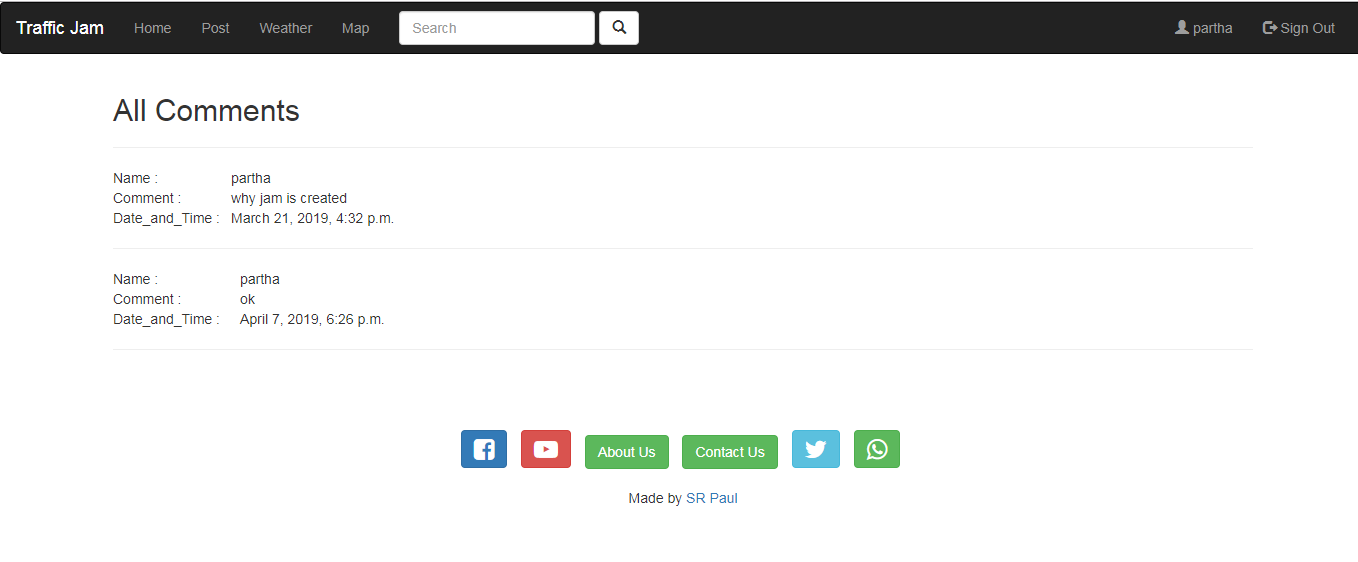


Figure 6.10: User manual for View all comments

## 6.11. Edit post

After publish post if user wants they can edit or update their post.

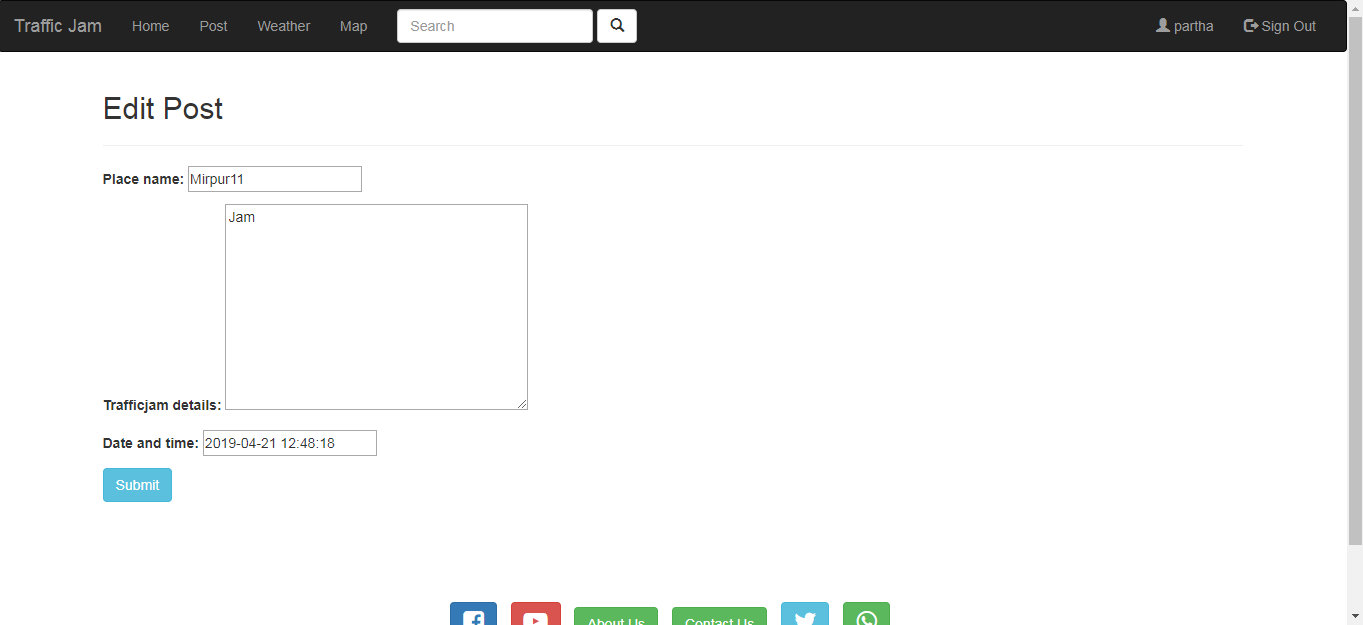
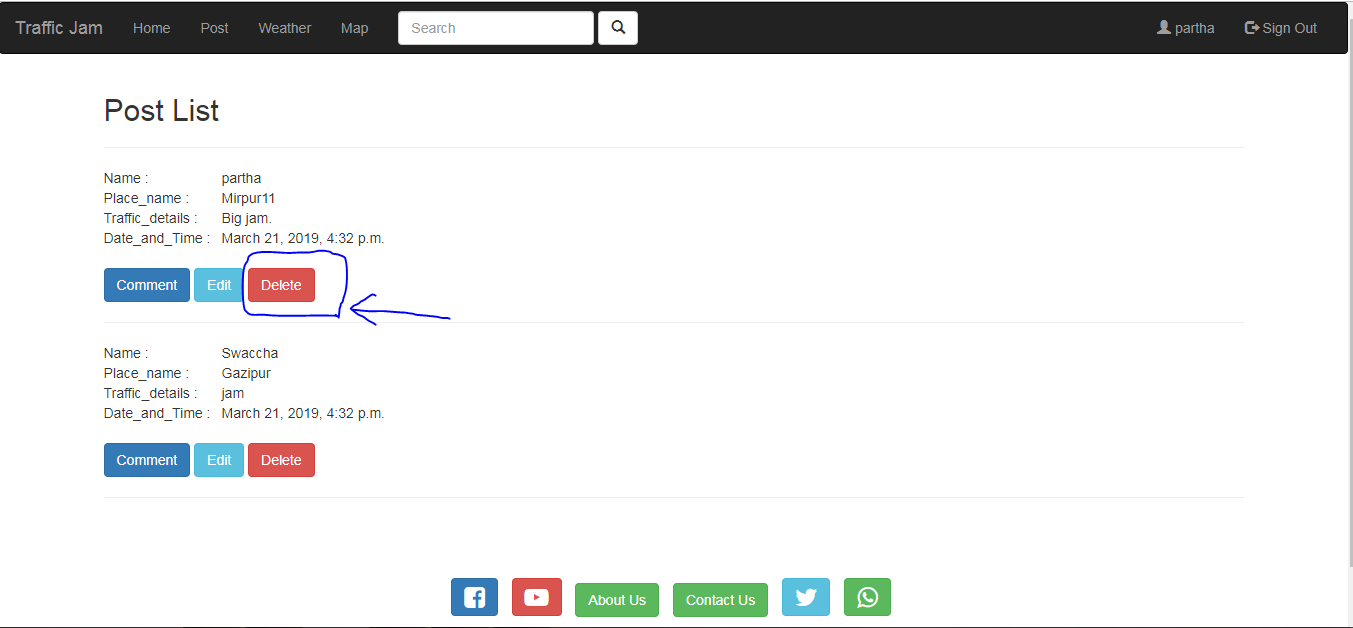


Figure 6.11: User manual for Edit post

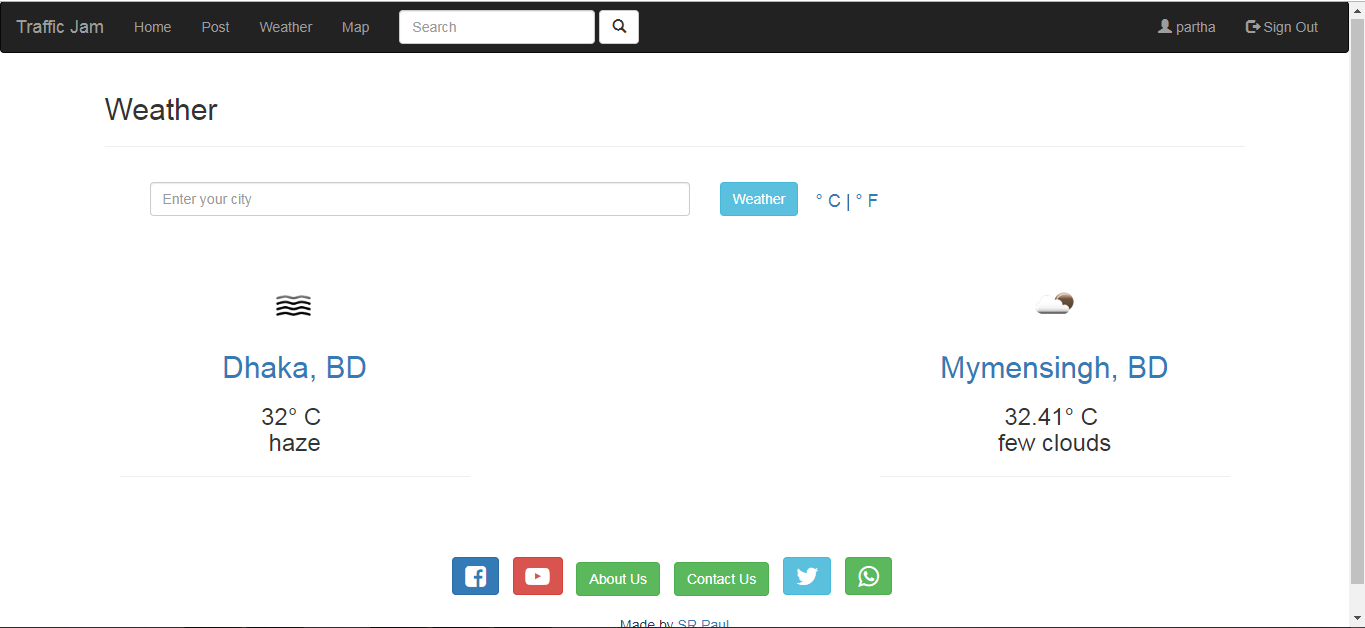
## 6.12. Delete post

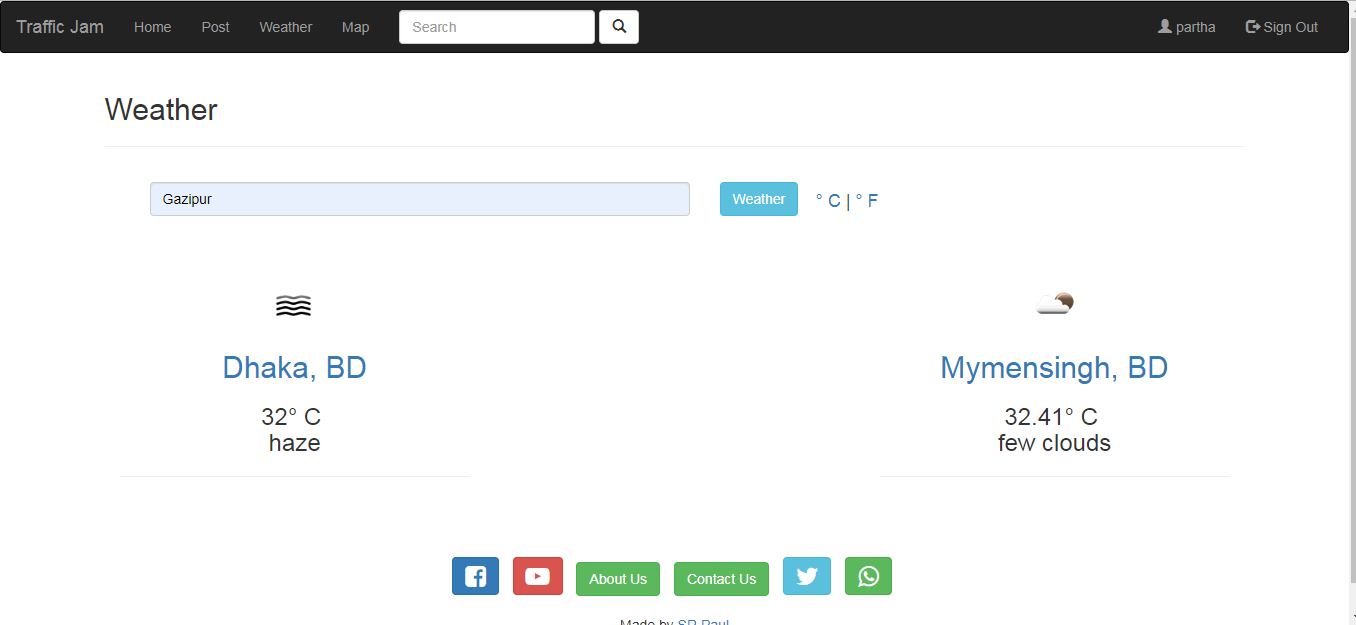
After publish post user can delete their post.

Figure 6.12: User manual for Delete post

## 6.13. Search city weather (Celsius)

Users can see their current city weather. This part would be only use for Celsius.





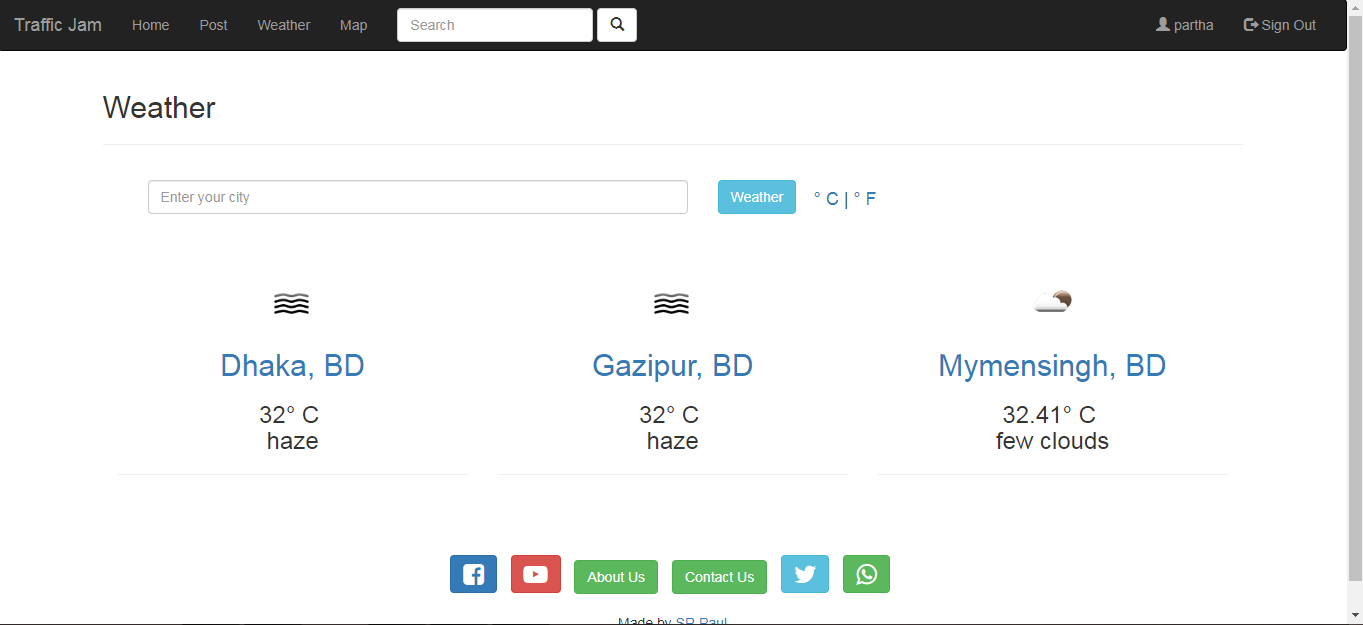
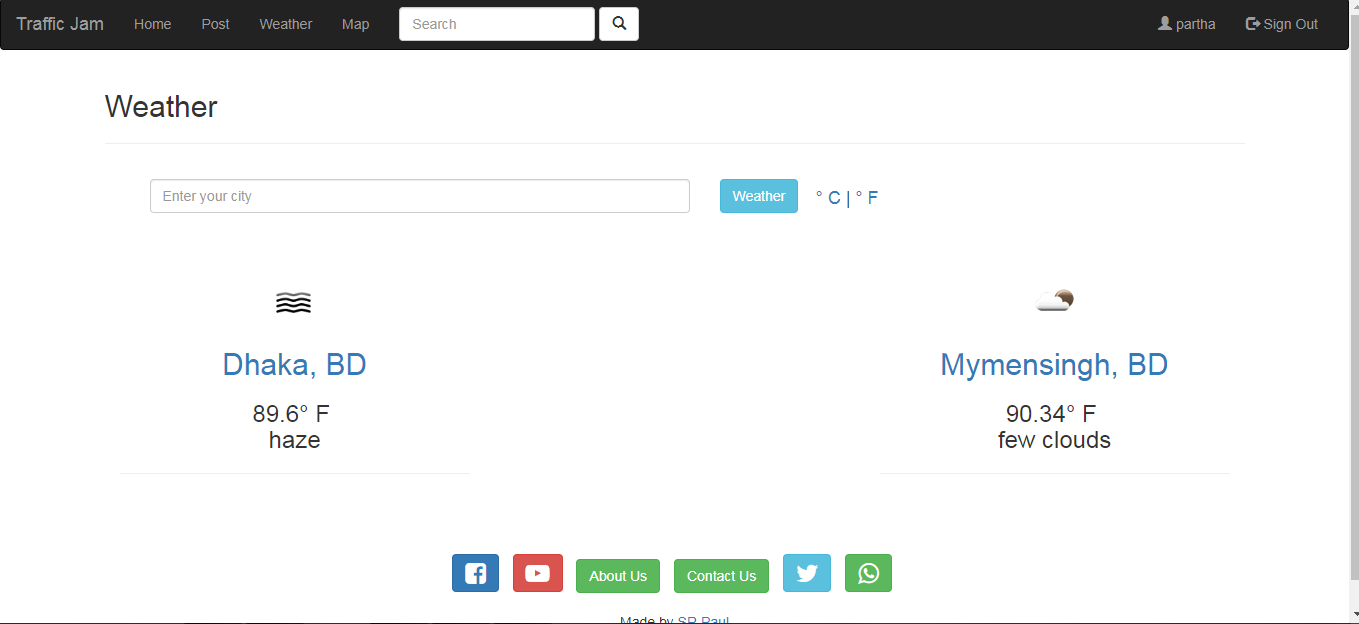
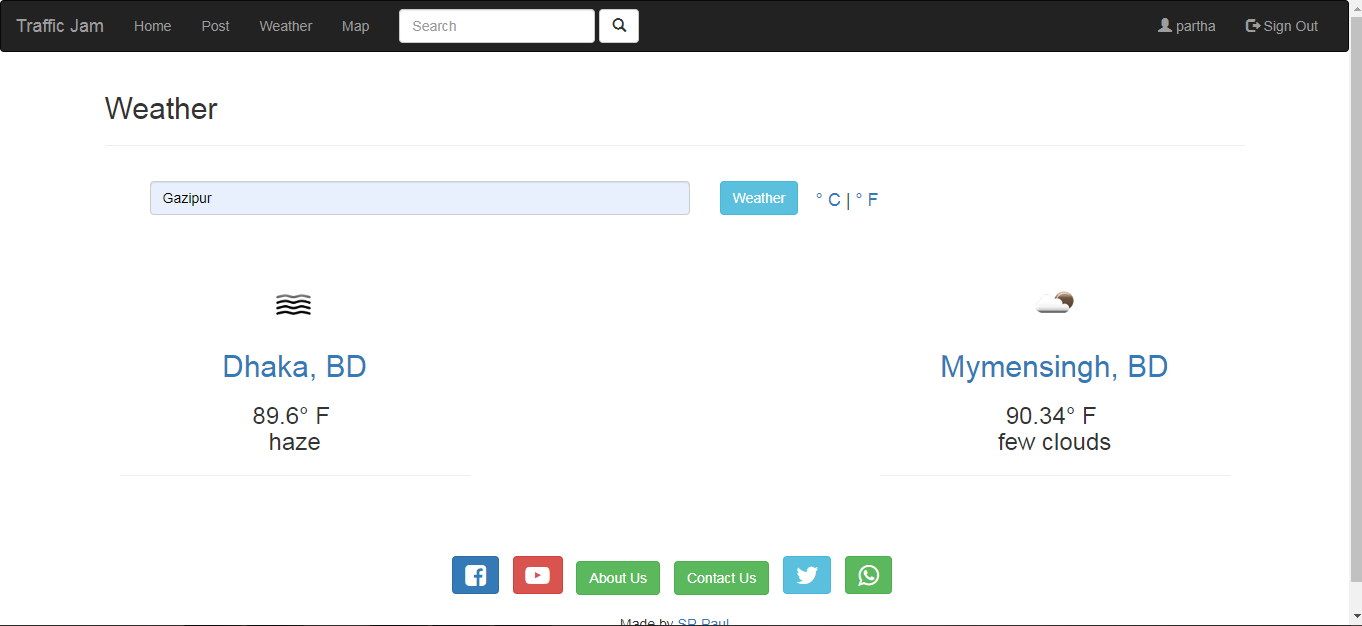


Figure 6.13: User manual for Search city weather (Celsius)

## 6.14. Search city weather (Fahrenheit)

Users can see their current city weather. This part would be only use for Fahrenheit.****



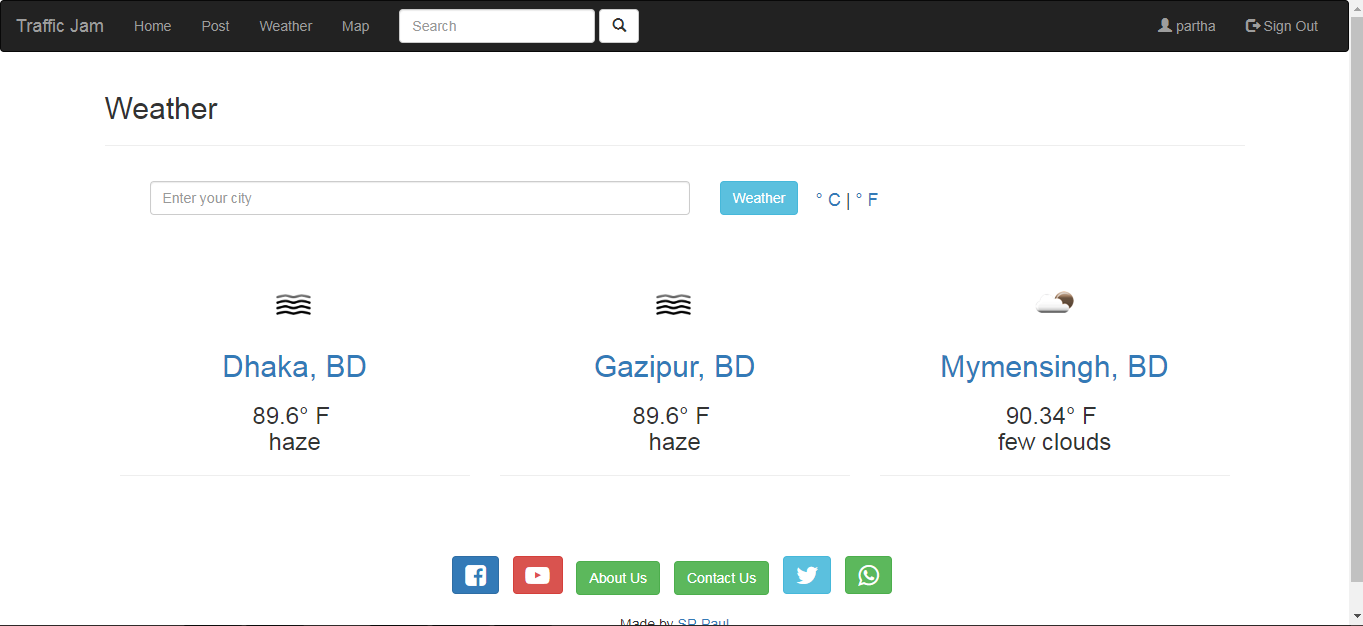


Figure 6.14: User manual for Search city weather (Fahrenheit)

## 6.15. Road map with current location

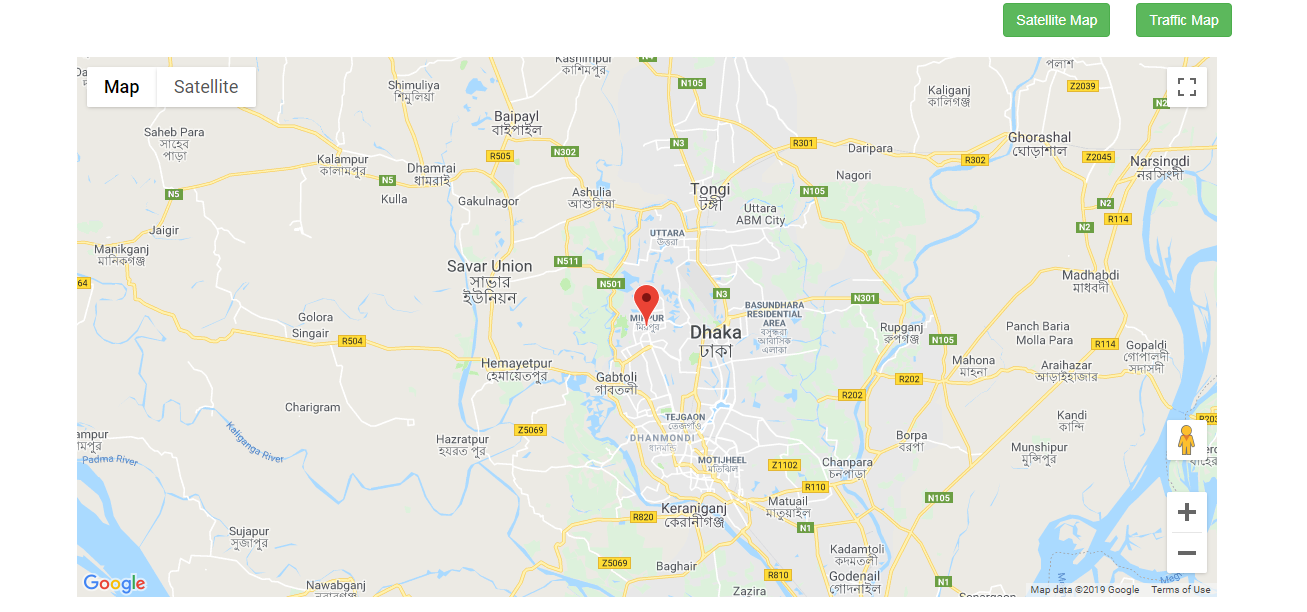
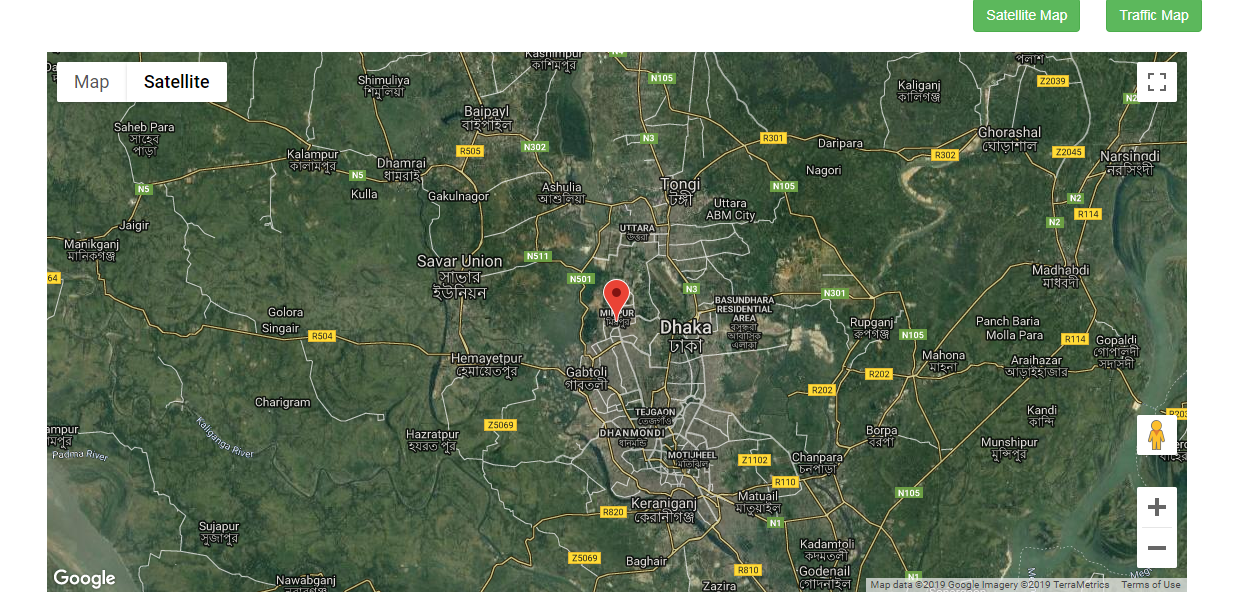
When user click map button then they will get Road map with current location.

Figure 6.15: User manual for Road map with current location

## 6.16. Satellite map with current location

When user click map button then they will get Satellite map with current location.

Figure 6.16: User manual for Satellite map with current location

## 6.17. Traffic jam map with current location

When user click map button then they will get Traffic jam map with current location.

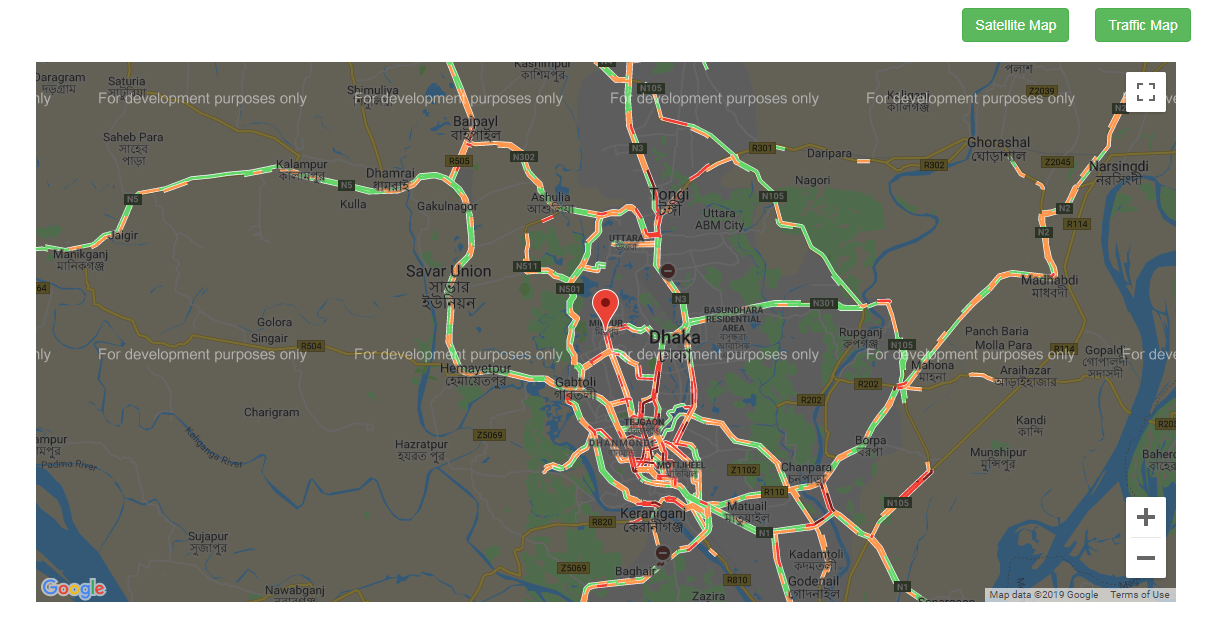
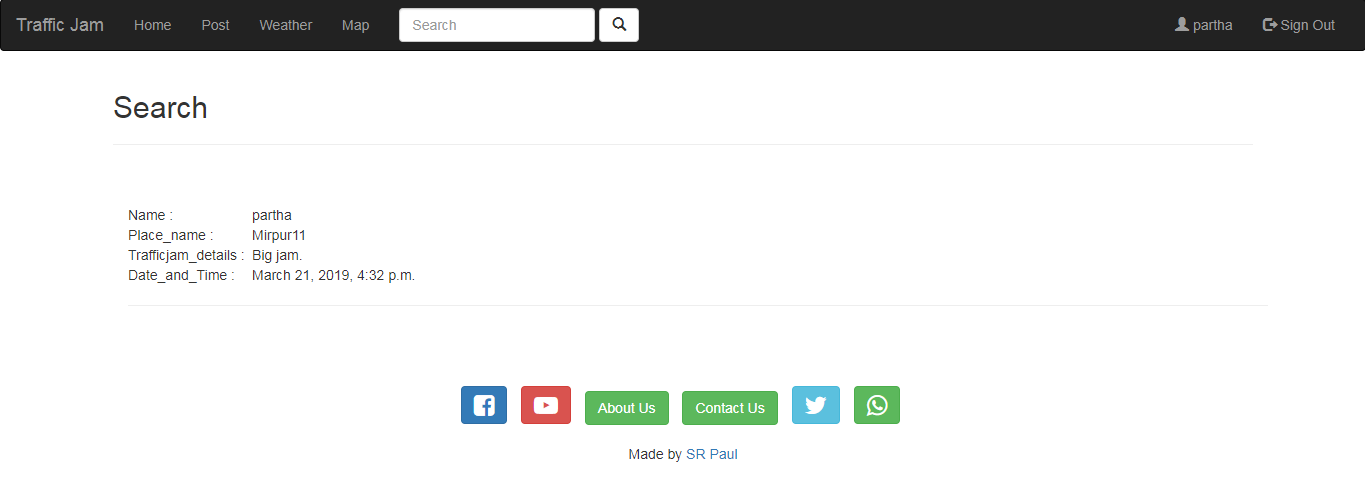


Figure 6.17: User manual for Traffic jam map with current location

## 6.18. Search post by place name

User can search post using by place name or date.

C:\Users\Partha Paul\Desktop\user\search1.PNG

Figure 6.18: User manual for Search post by place name

6.19. About usFrom about page use can know about this system.

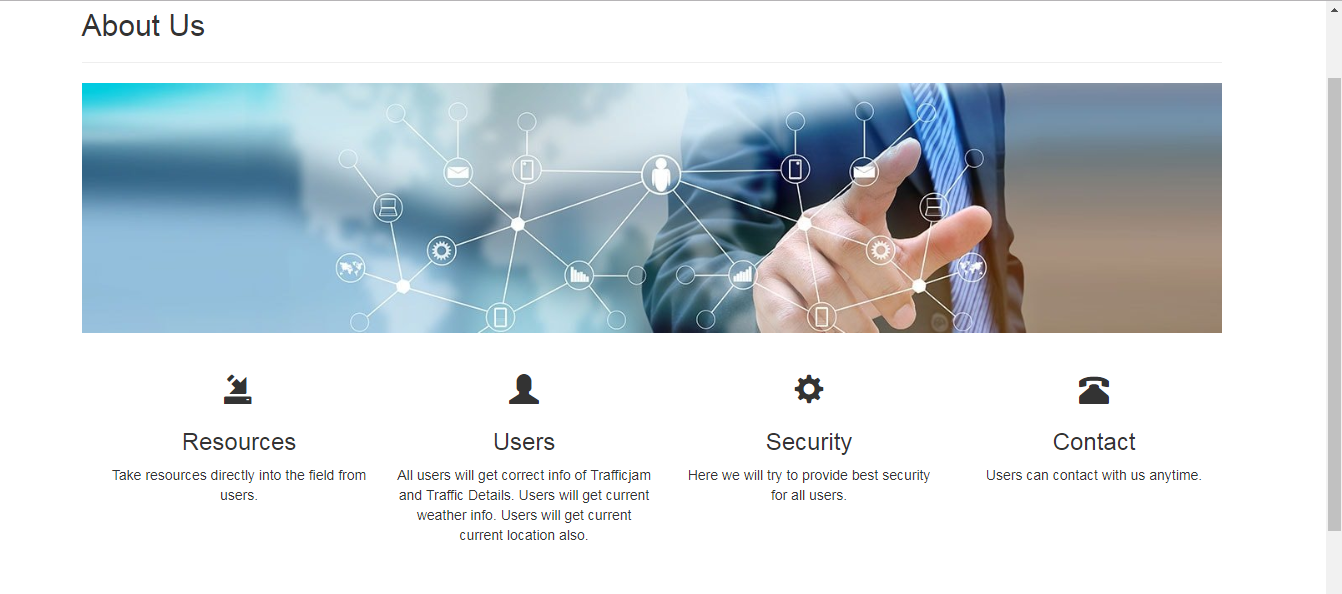
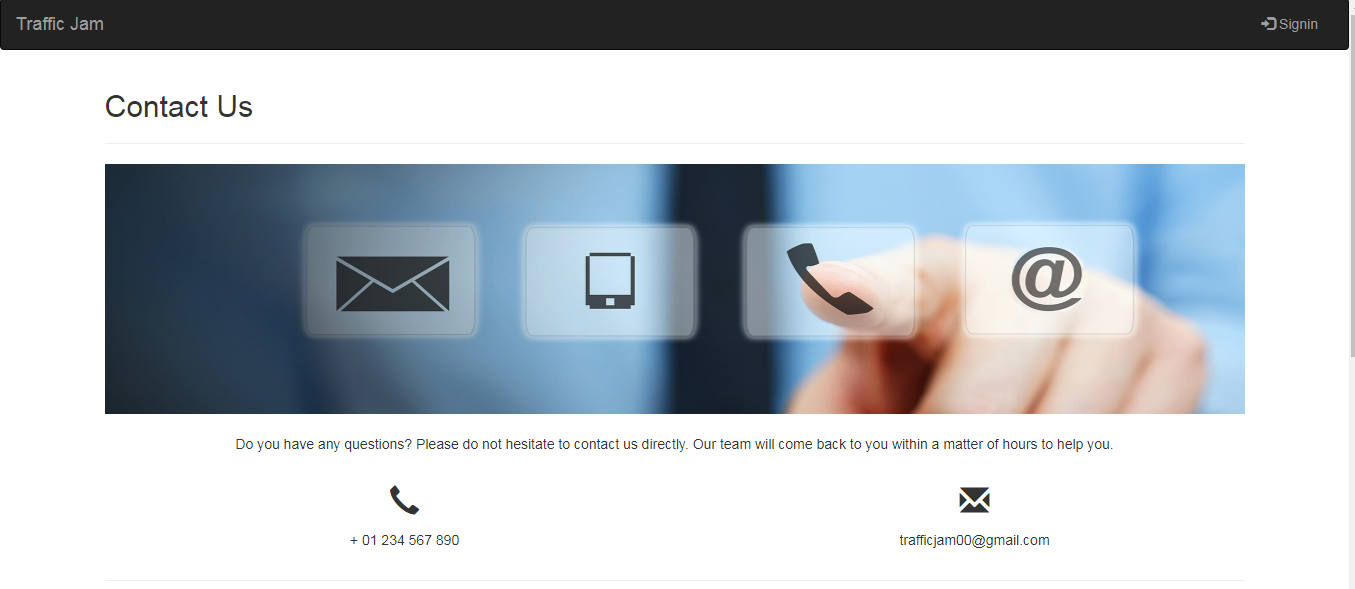


Figure 6.19: User manual for About us

## 6.20. Contact us

If user faces any problem they can contact with us phone number, email or messages.



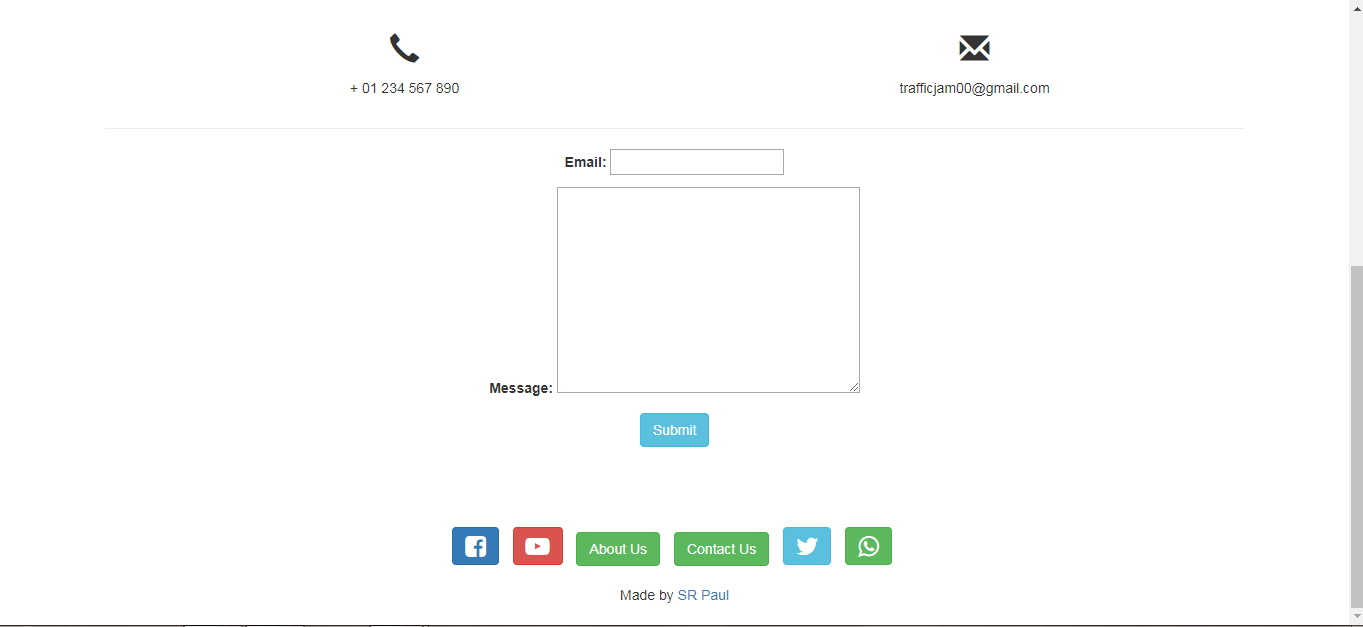


Figure 6.20: User manual for Contact us

## 6.21. Follow us

All users can follow us in Facebook, YouTube, twitter, Whatsapp.

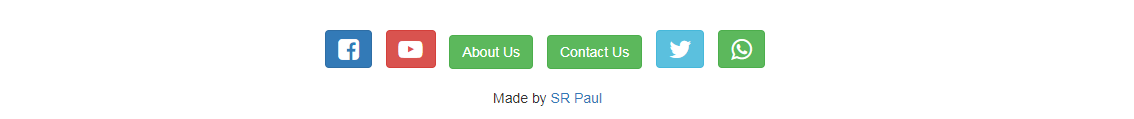


Figure 6.21: User manual for Follow us

## 6.22. Sign out

After all user can sign out from this system.

****Figure 6.22: User manual for Sign out

# Chapter 7: Project Summary

My project name is “Traffic jam”. I had started my project in January. From starting this project, first I collected required data from users. After that Then I make a design my project. After that I have proposed it. Then started to work.

Database is the main part of any system. Its plays a vital role for any system. That’s why; I have designed the database diagram having tables with proper relationship. After that, I have prepared the user interface. The interface of my system is very simple and easy to understand. After that, I have started to write the core functionality of the project.

After that testing plays second vital role. It is also known as quality assurance. Almost at every software company there is a quality assurance team. Their main responsibility is to find the vulnerability of any system. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have assured the quality of this project



## Github Link

https://github.com/parthapaulpartha/Trafficjam.git

## Limitations

For this project, it has a limitation. Now I will describe it.

* **Web site only:** This system only use for website. It’s could be use in phone browser also.

## Obstacles & Achievements

I believe that if there are not any obstacles to develop a project, then there doesn’t have any challenges. Because we know, challenge gives us the opportunity to prove ourselves.

Before starting this project, I didn’t know the actual flow of software development life cycle. By developing this project, I have learnt many things. My supervisor helps me a lot from the very beginning of the development of this project.

There are some other obstacles and achievements also that I will describe below.

* **Lack of User’s Engagement:** In this system all uses has different functionality. And almost they are busy with their day to day activities. So that’s why, I didn’t get all of users in proper time.
* **Scope Change:** Sometimes, some features need to be changed or modified. Then I need

to follow reverse engineering process. And again designed to meet the new requirements. It also made me frustrated sometimes.

## Future Scope

This system is now web base system only. From this project I have learnt lots of think. In the future, I want to publish the Android and iOS versions of this system.

## References

I have gained some knowledge from some platforms. Obviously I will mention those references.

Author (or Company Name): Refsnes data

Publish date: 1998

Website title: W3School

Date of access: 10-01-2019

Website address (URL): e <https://www.w3schools.com/>

Author (or Company Name): John Resig

Publish date: 2006

Website title: jquery

Date of access: 10-01-2019

Website address (URL): e <http://jquery.com/>

Author (or Company Name): Chad Hurley, Steve Chen, Jawed Karim

Publish date: 2005

Website title: YouTube

Date of access: 10-01-2019

Website address (URL): e <https://www.youtube.com/>

Author (or Company Name): [Tom Preston-Werner](https://en.wikipedia.org/wiki/Tom_Preston-Werner) , [Chris Wanstrath](https://en.wikipedia.org/wiki/Chris_Wanstrath) , [P. J. Hyett](https://en.wikipedia.org/wiki/P._J._Hyett) , Scott Chacon

Publish date: 2008

Website title: Github

Date of access: 20-04-2019

Website address (URL): e <https://github.com/>

Author (or Company Name): Mark Otto, Jacob Thornton

Publish date: 2011

Website title: Bootstrap

Date of access: 20-02-2019

Website address (URL): e <https://getbootstrap.com/>

Author (or Company Name): Jet Brains

Publish date: 2010

Website title: PyCharm

Date of access: 01-01-2019

Website address (URL): e <http://www.pycharm.com/>

Author (or Company Name): Google LLC

Publish date: 2005

Website title: Google Maps

Date of access: 20-03-2019

Website address (URL): e <https://www.google.com/maps/>