Software Requirements Specification

AutoMate - Vehicle Assistance System

SCS 2102 – Group Project I

Group CS - 03

Team nodeX;

Table of Contents

Group Details

Group: CS – 03

Name	Index Number	Registration Number	Signature
1) A.Rangoda	16001184	2016/CS/118	……
2) M.T.T.Dilshan	16000382	2016/CS/038	……
3) S.D.S.Karunarathna	16000692	2016/CS/069	……
4) K.D.Priyadarshani	16001151	2016/CS/115	……

Details of Project Supervisor, Mentor

- **Supervisor :-**

  Mr. Malik Silva

- **Mentor :-**

  Ms. Lalithi Perera

Abstract

This software requirement specification report, we have analyzed most common issues that a one can face in an unforeseen roadside breakdown. And also we have analyzed the inefficiency of service appointment handling process and keeping vehicle maintenance details. By this platform we can bridge the gap between vehicle owners and vehicle service/repair centers by building up an interactive communication between them.

1.0 Introduction

1.1 Purpose

This document intends to provide a detailed description of the requirements for the system "AutoMate". It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, system's feasibility, requirements, design and interfaces of the proposed system and interactions with other external applications along with all required diagrams. This document aims to provide concise description of system requirements, system feasibility and architecture of the proposed system.

1.2 Acronyms and Abbreviations

- SRS – System Requirement Specification
- UI – User Interface
- CRUD – Create Retrieve Update Delete

1.3 Domain Description

Vehicle breakdown is one of the most common issues which a one can face in at any time. Most people find it difficult to get a roadside assistance in an emergency situation. There are few helpline portals which helps to contact mechanical support for your roadside breakdown. But low efficiency of communication is a drawback of these helpline systems. Keeping service records and other records are very essential aspects of proper vehicle maintenance. Managing these records manually has become a very hectic process. So vehicle owners tends to find an automated way to fulfill these tasks. By the proposed system we will try to bridge the communication gap between vehicle owners and service providers and create an integrated system to achieve this target.

1.4 Current systems

Currently there is no specifically dedicated system for roadside breakdown in Sri Lanka. Mostly there are helpline systems which the users can request services over the phone. And those systems only provide limited amount of services. And also there is no integrated system which provide multiple vehicle support solutions. Each system provide independent solutions.

1.5 Limitations of current systems

- There is no integrated system which provides multiple solutions for vehicle support.
- Having difficulties when finding the most nearest and available service/repair center at a particular time.
- Difficult collaboration between vehicle owners and service centers.
- There is no proper way of retrieving exact location when there is an emergency.
- There are is no effective way to store vehicle history details, service details, insurance details and other details related to vehicle maintenance.
- No reminder generation for vehicle owners regarding upcoming service repairs and maintenances.

1.6 Proposed Solution

- Introduce an integrated mobile based platform in order to find service providers based on current location in an unforeseen roadside breakdown situation.
- Provide other vehicle support solutions like vehicle service and insurance renewal reminders, service detail reports, updating the vehicle owner with upcoming service repairs and maintenances.

1.7 Goal & Objectives

1.7.1 Goal

Creating an integrated platform for vehicle owners in order to support their vehicle maintenance activities. And also creating a platform for service centers to manage their day-to-day service activities.

1.7.2 Objectives

- Bridge the gap between customer and the service provider and make those services efficient and effective.
- Letting the system to be accessed from anywhere at any time.
- Creating a user friendly platform to the users.
- Enabling the vehicle owner to find the closest service center/mechanic in an unforeseen emergency situation.
- Ensuring better accessibility by providing a mobile application platform.
- Enabling the customers to decrease the time wastage by handling the emergency requests and service appointments in an efficient manner.
- Report generation.

1.8 Product Scope

1.8.1 In-scope

As the solutions for the limitations recognized above, we intend to introduce a mobile based platform to the vehicle owner in order to accommodate following tasks.

- Emergency request management
- Non-emergency service appointment management
- Manage vehicle details
- Manage service details
- Sending necessary alerts

For the vehicle owner, service center and the admin, the system will provide report generation and analysis facility.

1.8.2 Out-scope

Due to various limitations and constraints, the proposed system will not provide functionalities for the performance of the following functions

- Manage payments
- Schedule management (for vehicle services & maintenances)

1.9 Assumptions

- Vehicle owner having smart phone with internet connection with sufficient bandwidth.
- Service center having desktop computer with internet connection or a smart device with internet connection.
- The system will be implemented in an area which is within the GSM range of an ISP.
- Users will not falsely produce any document or enter fake details to the system.
- Users may enter valid details for fields where proper validation is not feasible such as signing up in the system and creating their profile, registering vehicles to the system.
- The system is implemented on a dedicated server.
- The database will be updated in real time.

1.10 Constraints & Limitations

- **Time constraints**

Main task of our system is locate most nearest and available service center in a roadside breakdown situation at any given time. So finding the best searching algorithms to achieving this task within this time period will be a challenge task for us.

- **System does not cover financial management**

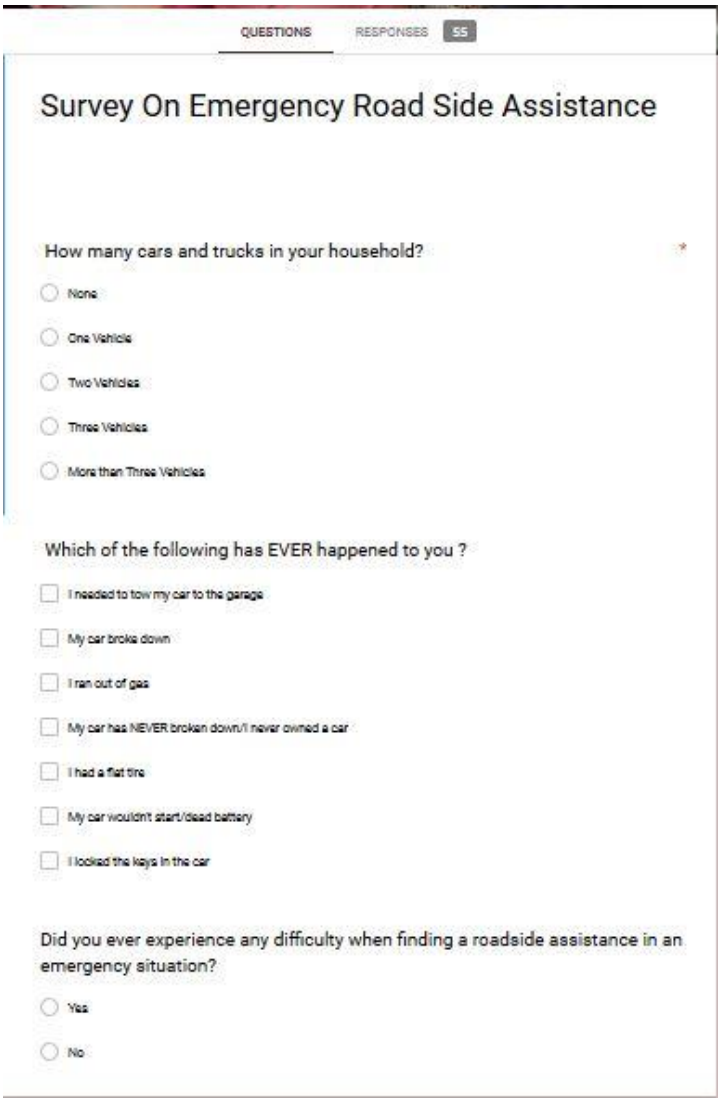
Since service charges fluctuates and toughness to assume a fixed price in these kind of services leads us not to process a mechanism to manage financial aspects.

2.0 Feasibility Study

In this section, we have discussed the feasibility analysis of developing "AutoMate" under few sections, such as operational, technical, economic, legal & ethical and schedule.

We have carried out an online survey regarding emergency roadside breakdown via a Google forum and identified relevant details in order to build our system.

And also we have generated results from the survey to analyze user requirements.



QUESTIONS RESPONSES 55

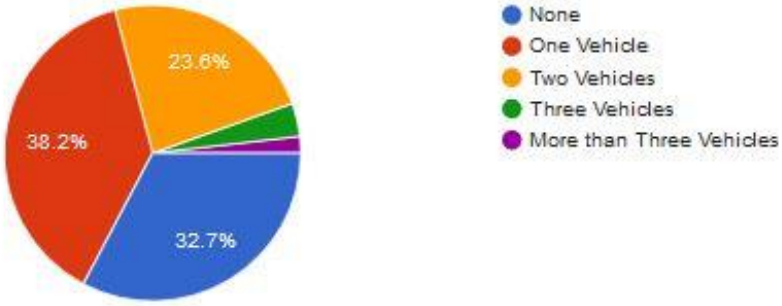
55 responses

SUMMARY INDIVIDUAL

Accepting responses

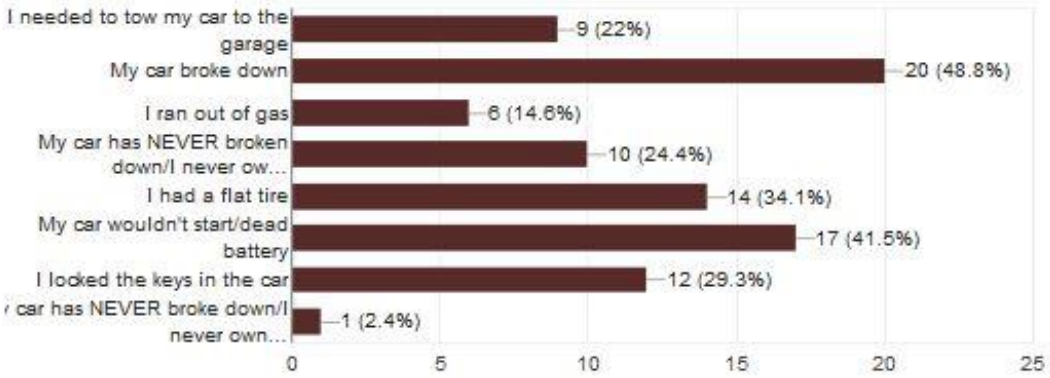
How many cars and trucks in your household?

55 responses



Which of the following has EVER happened to you ?

41 responses



2.1 Operational Feasibility

- This measures how well the solution meets the identified system requirements to solve the problems in the current system of vehicle maintenance.
- The system has 3 types of users.
    - Vehicle Owner
    - Service Center
    - Admin
- Any type of a vehicle owner can handle this system as it requires very least amount of IT knowledge. The system will provide simple and user friendly UIs so the user interaction will become high.
- Some small service centers might not have desktop computer to work with their operations. From this mobile friendly platform the system enables for service centers to work with their mobile device.
- Service centers don't need to monitor the system all the time to receive emergency service requests. System will notify the service centers about emergency services via a text message (SMS) to the service center receptionists' mobile device.
- For a general user, system does not require any other hardware components other than an electronic device with internet connection. Both mobile and web application will provide simple and intuitive interfaces to all the users.
- When we consider about customer and service center, sufficient knowledge will be enough to work with the system. No need to train the user. User can just download the app and work with it easily.
- Admin will monitor the system activities and handle issues.
- For all users the system can be accessed at anytime from anywhere.

2.2 Technical Feasibility

- Automate mainly consists of a Mobile application for vehicle owners, Web application for service providers and a database system.
- We hope to use modern technology, to develop the product. So the end users can easily use our system and get a real experience
- The mobile application will run on cross platform since it will be developed on top of Apache Cordova. The mobile application will be developed with ionic and angular2 and the backend server will be written with NodeJS.
- The web system will be developed with HTML, CSS and Javascript and for backend we use Laravel PHP framework which includes many features and also it has a large community.
- The Database Management System will be MySQL. This can be seamlessly integrated with the rest of the technological stack.

2.3 Economic Feasibility

- The system is to be implemented with the help of free and open source software. So there won't be any **software cost** to develop "AutoMate".

Cost Benefit Analysis

Total estimated cost of the proposed system is discussed under two categories. There are development cost and projected annual operating cost.

Development Cost

Personnel:

System Analyst	Rs. 20 000
System Designer	Rs. 15 000
UI/UX Designer	Rs. 10 000
Mobile App Developer	Rs. 50 000
Web Developer	Rs. 40 000
Database Specialist	Rs. 15 000
Software Tester	Rs. 30 000

Estimated Total Personnel Cost = Rs. 180 000

- According to the cost analysis shown above, the total development cost of AutoMate is Rs. 180 000.
- If we assume that the service centers own a computer and an internet connection, they do not have to purchase those. Although if they do not have a computer, the system is supposed to run on mobile devices as well. So having a computer and an internet connection is not essential.
- Hence there's no New Hardware & Software cost in our system.

Projected Annual Operating Cost

System maintenance	Rs. 10 000
Hosting	Rs. 3 000
Web server	Rs. 15 000
Database server	Rs. 25 000

Estimated Annual Operating Cost = Rs. 53 000

- According to the cost analysis shown above, the estimated annual operating cost of AutoMate is Rs. 53 000.

2.4 Legal & Ethical Feasibility

- Almost all the software and resources used during the development will be free and open sourced. Therefore illegal software issues will be minimum.
- User location tracking will be operate under specific user's agreement policy.
- The system will collect sensitive data like phone number, current location only with user approval.
- Sensitive information will not be published to the outside world by the system.

2.5 Schedule Feasibility

- We are hoping to complete our product with the proposed time schedule.
- Project plan -
    - 4 months for requirement analysis and design.
    - 5 months of Implementation and testing.
- Development team working hours allocation as follows:

    No of team members = 4
    No of working hours for each member:
        Weekdays = 2
        Weekend = 3

    Man hours per week = (2+3)*4 = 20

    For the whole project:
    Estimated no of months = 9
    Estimated no of weeks = 36
    Estimated total man hours = 20*36 = 720

    For the development and testing:
    Estimated no of months = 5
    Estimated no of weeks = 20
    Estimated total man hours = 20*20 = 400

    400 hours to complete the overall project development and there are 8 main components. For each component the development team have 50 hours.

- Therefore this schedule feasibility implies that there is sufficient amount of time to complete the whole project.
- Following is the project plan for the system "AutoMate"

	Task	Task Owner	March				April				May				June				July				August				September				October				November			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Problem Identification	Requirement Gathering	Team																																				
	Requirement Analysis	Team																																				
	Scope Identification	Team																																				
	Feasibility Study	Team																																				
System Analysis	Use Case Modeling	Team																																				
	Activity Modeling	Team																																				
	Data Flow Modeling	Team																																				
	SRS Completion	Team																																				
System Design	User Interface Design	69 & 38																																				
	Database Design	118 & 115																																				
System Development	**Mobile Platform**																																					
	Emergency Service Module	118 & 38																																				
	Location Navigator Module	118 & 38																																				
	Non-emergency Service Module	118 & 115																																				
	Vehicle Management Module	69 & 115																																				
	Notification Module	118 & 38																																				
	Web Platform																																					
	Emergency Service Module	69 & 115																																				
	Non-emergency Service Module	69 & 115																																				
	Report Generation Module	118 & 38																																				
Testing	Unit Testing	Team																																				
	Integrated Testing	Team																																				
	System Testing	Team																																				
	Alpha Testing	Team																																				
Maintenance	Mobile App,Web App,Database	Team																																				

3.0 Deliverables

Our deliverables will be a mobile application for vehicle owners (user) and a web application for service centers which is dedicated to above mentioned criteria.

3.1 Mobile Application

The mobile based application would include all the functional requirements as specified in this document.

3.2 Web Application

The web application would include all the functional requirements that are specified with the service center in this document.

4.0 Requirements

4.1 Stakeholders

The main stakeholders who are going to use the Vehicle Assistance System can be identified as below.

4.1.1 Vehicle Owner

Vehicle owner can send emergency request and book non-emergency appointments online for services. And he/she can maintain their vehicle details, insurance details, view service history of the vehicle, rate/review service center after completing a service.

4.1.2 Service Center

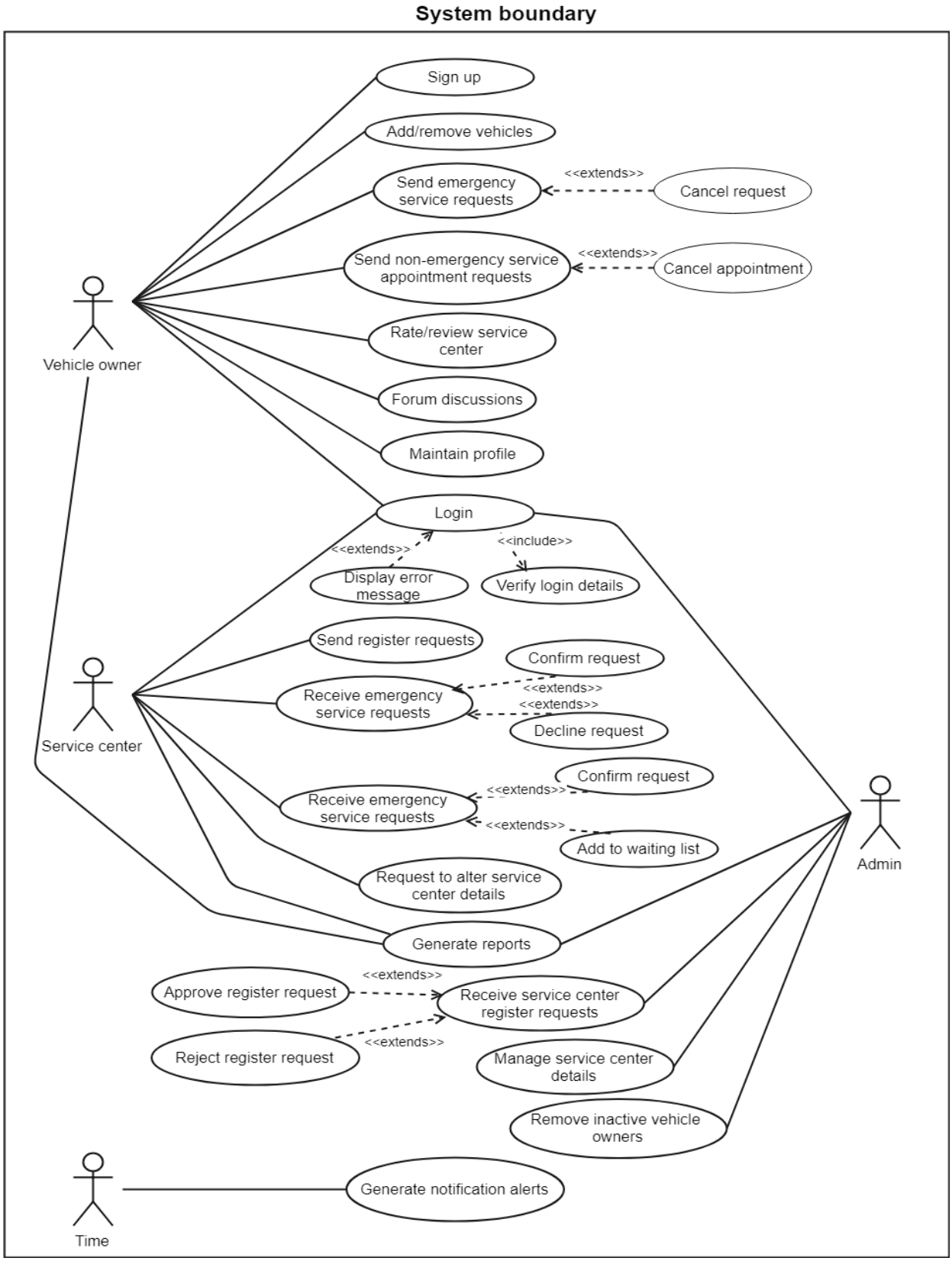
Service center will send notifications to the customer regarding upcoming appointments, offers etc. And it can view emergency requests that are coming from the vehicle owners and the service center will manage those emergency requests.

4.1.3 Admin

Administrator can manage vehicle owners, vehicles and service centers. Also, administrator is the one who's responsible of accepting/rejecting register requests from service centers.

Also he has the ability to avoid unnecessary database overloading by removing inactive vehicle owners and service providers.

4.2 Use Cases and Use Case Diagram



4.3 Functional Requirements

4.3.1 Mobile Application

- Login.
- Add/remove vehicles.
- Send emergency service requests.
- Send non-emergency service requests.
- Rate/Review service center.
- Update user profile.
- Receive service reminders and notification alerts.
- Generate service history detail reports.
- Forum discussions.

4.3.2 Web Application (Service Center)

- Login.
- Send system registration request.
- Receive emergency service requests.
- Receive non-emergency service requests.
- Generate Vehicle owner and service center detail reports.

4.3.3 Web Application (Admin)

- Login.
- Manage service center register requests.
- Manage service center details.
- Remove inactive vehicle owners.
- Generate vehicle, vehicle owner, service center detail reports.

4.3.4 Use case narratives

Mobile Application

Use case name	Login
Priority	Medium
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• Application home interface is available • User has his own profile/account • User is connected to the internet
Main flow	1. User clicks on the login button 2. Login page is displayed 3. User enters his/her username and password 4. System authenticates user 5. System logs in user
Alternative flows	If entered username and password are not matching, displays an error message
Post conditions	User profile is loaded

Use case name	Sign up
Priority	Medium
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• User does not have an account • User is connected to the internet • Sign up page is displayed
Main flow	1. User enters his/her details in relevant fields 2. Clicks on the OK button 3. User gets signed up
Alternative flows	None
Post conditions	Vehicle owner details are recorded. User profile is displayed

Use case name	Add vehicles
Priority	High
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	Administrator
Pre-conditions	• Vehicle owner is connected to the internet • Vehicle owner has logged into the system • Dashboard is available
Main flow	1. Vehicle owner clicks on the "Add" tab 2. Selects "Add vehicle" from the drop down list 3. Enter vehicle information in relevant fields 4. Click on "Save details'
Alternative flows	If any required field is not filled, an alert is displayed before submitting
Post conditions	vehicle details regarding to the relevant vehicle owner are recorded

Use case name	Send emergency service requests
Priority	High
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	Service center
Pre-conditions	1. Vehicle owner is connected to the internet 2. Vehicle owner has logged into the system 3. Dashboard is available
Main flow	1. Vehicle owner clicks on the "Services" tab 2. Selects "Emergency services" from the drop down list 3. Click on the radio button in front of the relevant service type (if the required service is known. Else click on "Other" type) 4. Click on "Search" button 5. List of available and nearest service providers are displayed 6. Select the service provider from the list

	7. Click on “OK” button
Alternative flows	None
Post conditions	Service request with vehicle details and owner’s current location is submitted to the selected service provider

Use case name	Send non-emergency service request
Priority	High
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	Service center
Pre-conditions	• Vehicle owner is connected to the internet • Vehicle owner has logged into the system • Dashboard is available
Main flows	1. Vehicle owner clicks on the “Services” tab 2. Select “non-emergency service” from the drop down list 3. Select the service provider 4. Select an available date and time for the appointment from the calendar 5. Enter service information on relevant fields 6. Click on “Proceed” button
Alternative flows	None
Post conditions	Service request is submitted to the selected service provider

Use case name	**Rate/review service**
Priority	Low
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• Vehicle owner is connected to the internet • Vehicle owner has logged into the system

	• Vehicle owner and service provider has concluded initial communication • Rating/ reviewing window is available
Main flow	1. Vehicle owner selects number of stars to rate the service 2. Writes comments on text area (optional) 3. Click on "Submit" to submit the rating
Alternative flows	None
Post conditions	Rating is recorded

Use case name	Maintain the profile
Priority	Medium
Primary business actor	Vehicle owner
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• Vehicle owner is connected to the internet • Vehicle owner has logged into the system • Dashboard is available
Main flow	1. Click on the "Profile" tab 2. Edit/update details in the profile window 3. Clicks on "Save changes" button
Alternative flows	If any field is out of format, an error message will displayed
Post conditions	Vehicle owner's profile information is updated

Use case name	Generate notification alerts
Priority	High
Primary business actor	Time
Other participating actors	Vehicle owner
Other interesting stake holders	Service center
Pre-conditions	• Vehicle owner is connected to the internet • Vehicle owner has logged into the system

Main flow	1. A reminder will be generated sequentially according to a given time scheme
Alternative flows	None
Post conditions	The generated notification is displayed in vehicle owner's mobile phone when he connected to the internet

Web Application

Use case name	Login
Priority	Medium
Primary business actor	System admin, Service center
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• Application home interface is available • User has his own profile/account • User is connected to the internet
Main flow	1. User clicks on the login button 2. Login page is displayed 3. User enters his/her username and password 4. System authenticates user 5. System logs in user
Alternative flows	If entered username and password are not matching displays and error message
Post conditions	User profile is displayed

Use case name	Send register request
Priority	Medium
Primary business actor	Service center
Other participating actors	None
Other interesting stake holders	System administrator
Pre-conditions	• User is connected to the internet • User does not have an account/profile • Sign up interface is available

Main flow	1. User clicks on the register button 2. Account registration page is displayed 3. User enters valid details about service center in relevant fields 4. User clicks on the "Submit" button to send the register request
Alternative flows	If entered details are not in the correct format, an alert is displayed before submitting
Post conditions	Service center's register request is sent to a system admin for confirmation

Use case name	Receive emergency service request
Priority	High
Primary business actor	Service center
Other participating actors	None
Other interesting stake holders	Vehicle owner
Pre-conditions	• User is connected to the internet • User is logged into the system • An emergency service request is received
Main flow	1. Click on "Go" button on the new emergency request notification 2. Check the details of the request 3. Check availability 4. Click on accept/decline button to accept/reject the request
Alternative flows	• If a suitable mechanic is available accept the request and appoint the mechanic • Else decline the request
Post conditions	Confirmation (accept/decline) message is sent to the vehicle owner via SMS and displays other information in vehicle owner's mobile application

Use case name	Receive non-emergency service request
Priority	High
Primary business actor	Service center
Other participating actors	None

Other interesting stake holders	Vehicle owner
Pre-conditions	• User is connected to the internet • User is logged into the system • Dashboard is available
Main flow	1. Click on “Requests” tab and select “Non-emergency service request” 2. If there any pending requests, click on the request and check the details 3. Check availability 4. Click on accept/decline button to accept/reject the request
Alternative flows	• If the requested time slot is not available add that request to the waiting list • Else accept the request
Post conditions	• Update the calendar • If available acceptance message is sent via SMS. Displays other information on vehicle owner’s mobile phone • Else nearest available time slot is sent via SMS. Displays other information regarding the service on vehicle owner’s mobile phone

Use case name	Receive service center register request
Priority	High
Primary business actor	System admin
Other participating actors	None
Other interesting stake holders	Service center
Pre-conditions	• User is connected to the internet • User is logged into the system • Received a register request
Main flow	1. Click on the “Go” button on new register request notification 2. Check the details of the request 3. Accept or reject the request
Alternative flows	• If the given details are not valid reject the request • If the service centers does not meet the required standards reject the request
Post conditions	Send confirmation(accept/reject) message to service provider via email

Use case name	Generate reports
Priority	Medium
Primary business actor	Service center, System admin
Other participating actors	None
Other interesting stake holders	None
Pre-conditions	• User is connected to the internet • User is logged into the system
Main flow	1. Click on “Reports” tab 2. Select the type of the report 3. Edit the content in the generated report (optional) 4. Click on “Print” button to print the report or click on “Save” button to save
Alternative flows	None
Post conditions	None

Use case name	Remove inactive vehicle
Priority	Medium
Primary business actor	Admin
Other participating actors	None
Other interesting stake holders	Vehicle owner
Pre-conditions	• User is connected to the internet • User is logged into the system • Dashboard is available
Main flow	1. Click on vehicle owner tab 2. Select “Remove vehicle” 3. A list of inactive vehicle are displayed 4. Click on “more” button in front of the vehicle for more details 5. Click on “Remove” button to remove vehicle
Alternative flows	None
Post conditions	Vehicle details are removed from the system

Use case name	Manage service center details
Priority	Medium
Primary business actor	System admin
Other participating actors	None
Other interesting stake holders	Service center
Pre-conditions	• User is connected to the internet • User is logged into the system • Service center details alteration request is received
Main flow	1. Click on “Request” tab 2. Select alteration requests from the drop down list 3. Go to the relevant service provider’s profile 4. Edit the content as requested
Alternative flows	• Informing service center about the modification via an email
Post conditions	Service center details has updated

Use case name	Request to alter service details
Priority	High
Primary business actor	Service center
Other participating actors	None
Other interesting stake holders	System admin
Pre-conditions	• User is connected to the internet • User is logged into the system • Dashboard is displayed
Main flow	1. Click on profile tab 2. Select “Modification” from the drop down list 3. Modification window is appeared 4. Change the details in the profile 5. Click on “Request to modify” button
Alternative flows	None
Post conditions	Service center alteration request is sent to the system admin

5.0 Other Non-Functional Requirements

5.1 Performance Requirements

- Since this software is going to be web-based, it does require a powerful server machine with high band internet access. So, we hope to hire an effective web server.
- From the user's point of view, mobile application should be developed as lightweight web app so that it can work on almost any platform even with slower internet connections. Because in remote areas, it is hard to find a very good internet connection. Users have to stick with the poor and available internet connection.

5.2 Security Requirements

- Since we use client server architecture, all the user data will be kept on the cloud server. Product should be able to guarantee the privacy of user data. Workspace of the user should only be accessed through user own credentials and any other user should not be able to access to the user private data.
- There are authorization and authentication levels in our system and we hope to provide a role based access model, which the privileges which have designed for each and every role. The system will provide user authentication by the email, password and user role
- Access privileges are maintained and also user can report system issues.

5.3 User Friendliness and Responsiveness Requirements

- Simple and clear interfaces will help vehicle owners to interact especially in emergency situation with our system.
- The software should be lightweight so that it can run on a device with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase such as, Javascript and HTML5.
- In order to make location navigation efficient and more responsive, we use Launch Navigator plugin which is available in ionic framework. It requires, ***Cordova plugin: uk.co.workingedge.phonegap.plugin.launchnavigator***.

5.4 Availability Requirements

- Web based system is available 24x7, so vehicle owners can send emergency service requests and non-emergency service appointments at any time.
- Admin and service centers can also access the system at any time.
- System shall extremely mobile friendly, which is responsive to all mobile platforms.

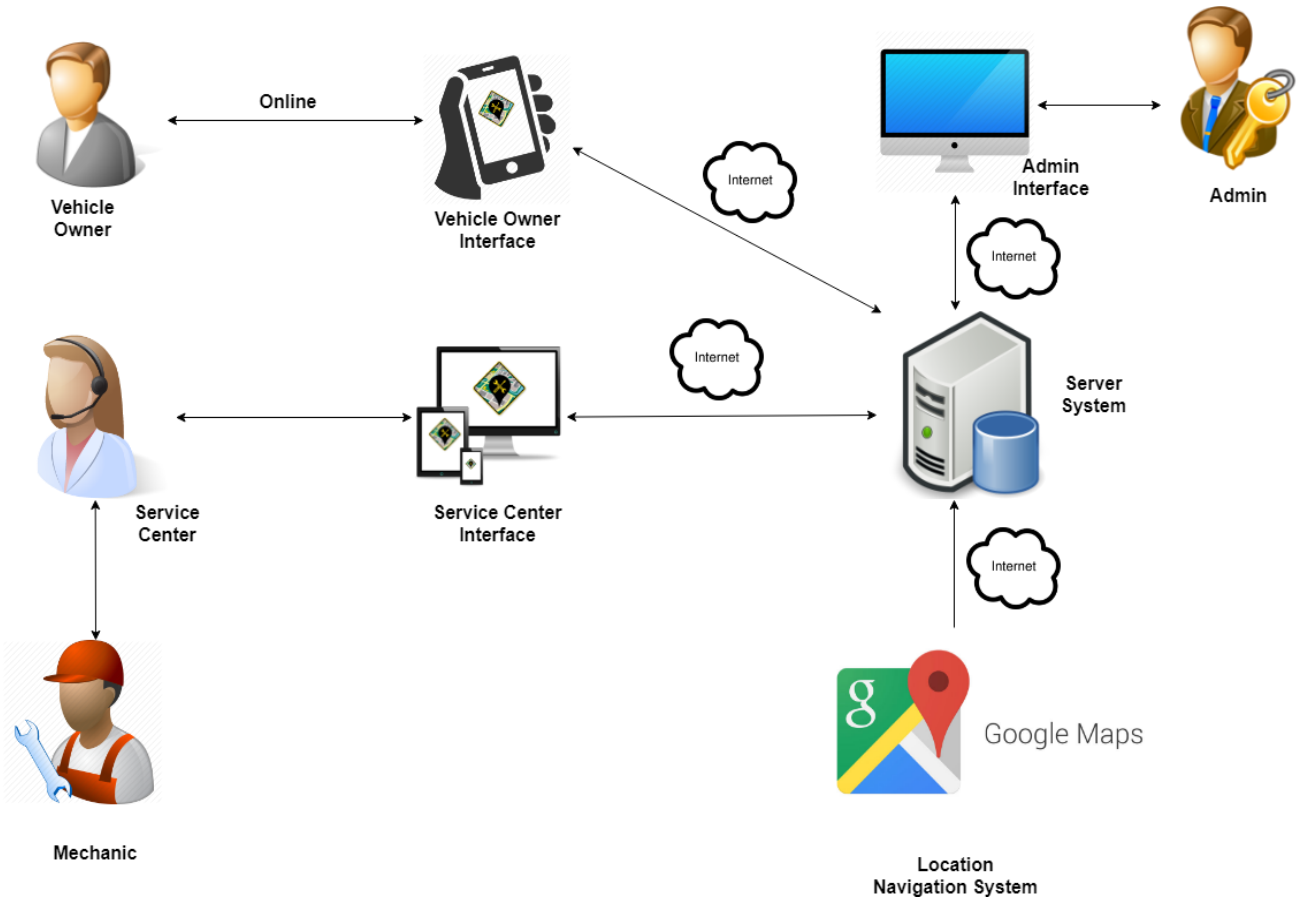
5.5 Reliability Requirements

- The system shall be able to process all work correctly and completely without being aborted.
- Service center registration confirmation message shall be sent via email within 3 minutes after accepting the registration.
- Daily backups of the system shall be maintained automatically.

6.0 Proposed System Architecture

6.1 High level architecture

The system will be following client-server architecture. We will be using an independent web server to host both database and web system. In an emergency situation, we will be using Google Maps API to locate nearest available service center. Other services will also provide through the system by interacting user interfaces with the server. For vehicle owners, the system will be accessible from the mobile app and for service center, the system will be accessible from desktop/notebook browsers and smart mobile phones. Hence the web system will be mobile friendly for the users.



6.2 Components and their responsibilities

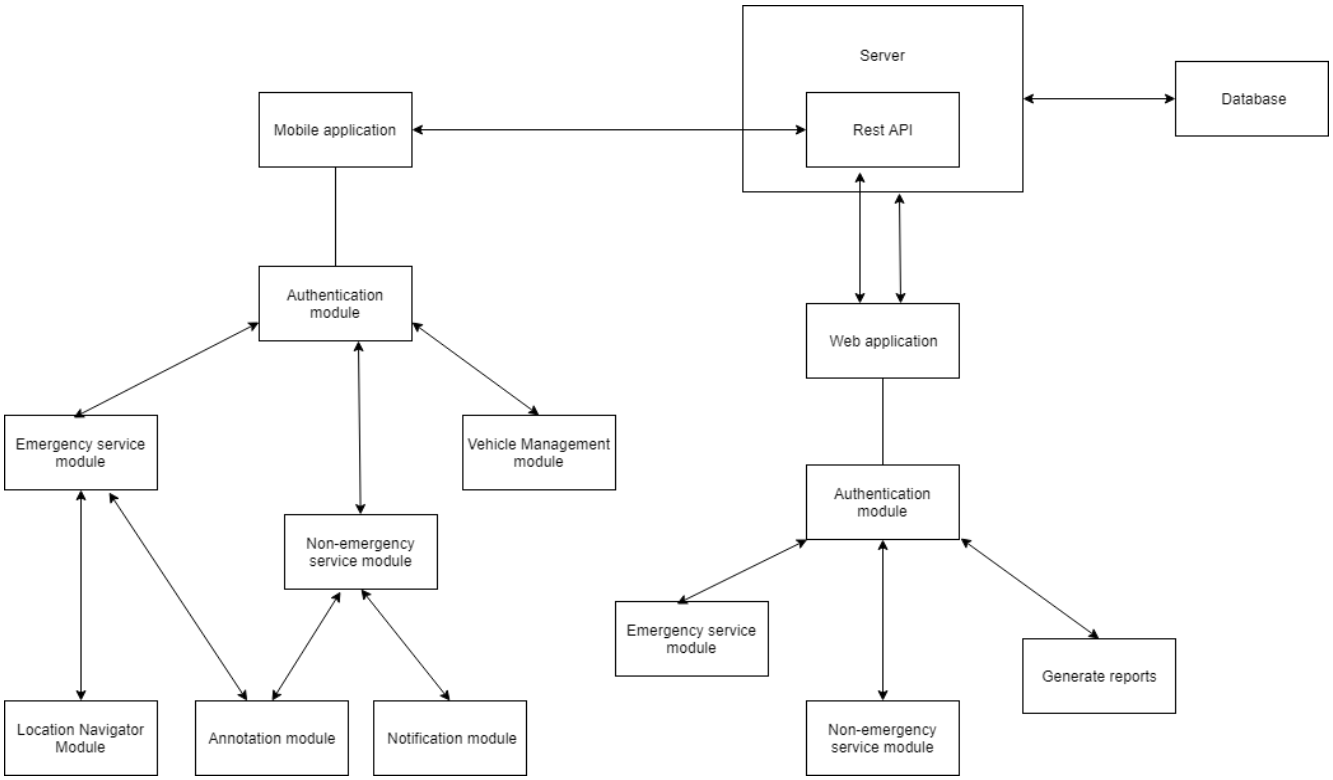
6.2.1 Mobile Application

Component/Module	Responsibility
Authentication module	• Signup/Login to the system • Logout from the system
Emergency service module	• Search for the most nearest & available service center and request for service. • Cancel request • Update request details
Non-emergency service module	• Create new appointments • Cancel appointments • Update appointment details
Vehicle management module	• Insert new vehicles • Remove vehicles • Update vehicle details
Location navigator module	• Locate the most nearest service center available. • Sending location details of the vehicle to the service center.
Notification module	• Receive notifications about upcoming services. • Receive notification about offers • Receive notifications about insurance renewal and other insurance details.
Annotation module	• Rate/review service center after completing the service

6.2.2 Web application

Component/Module	Responsibilities
Authentication module	• Signup/Login to the system • Logout from the system
Emergency service module	• Receive emergency service requests • Receive vehicle location and other details
Non-emergency service module	• Handle appointment schedule • Get new appointments • Cancel appointments
Generate reports	• Calculate daily collection • Produce monthly reports

6.3 Component Interactions



6.3.1 Mobile Application

The mobile application would be developed using ionic and angular 2. There are lot of PHP frameworks provided free of charge to create web based applications. We would be using Slim PHP framework to connect our application with API's. Slim Restful API's will be used in order to get web services for the mobile app.

6.3.2 Web Application

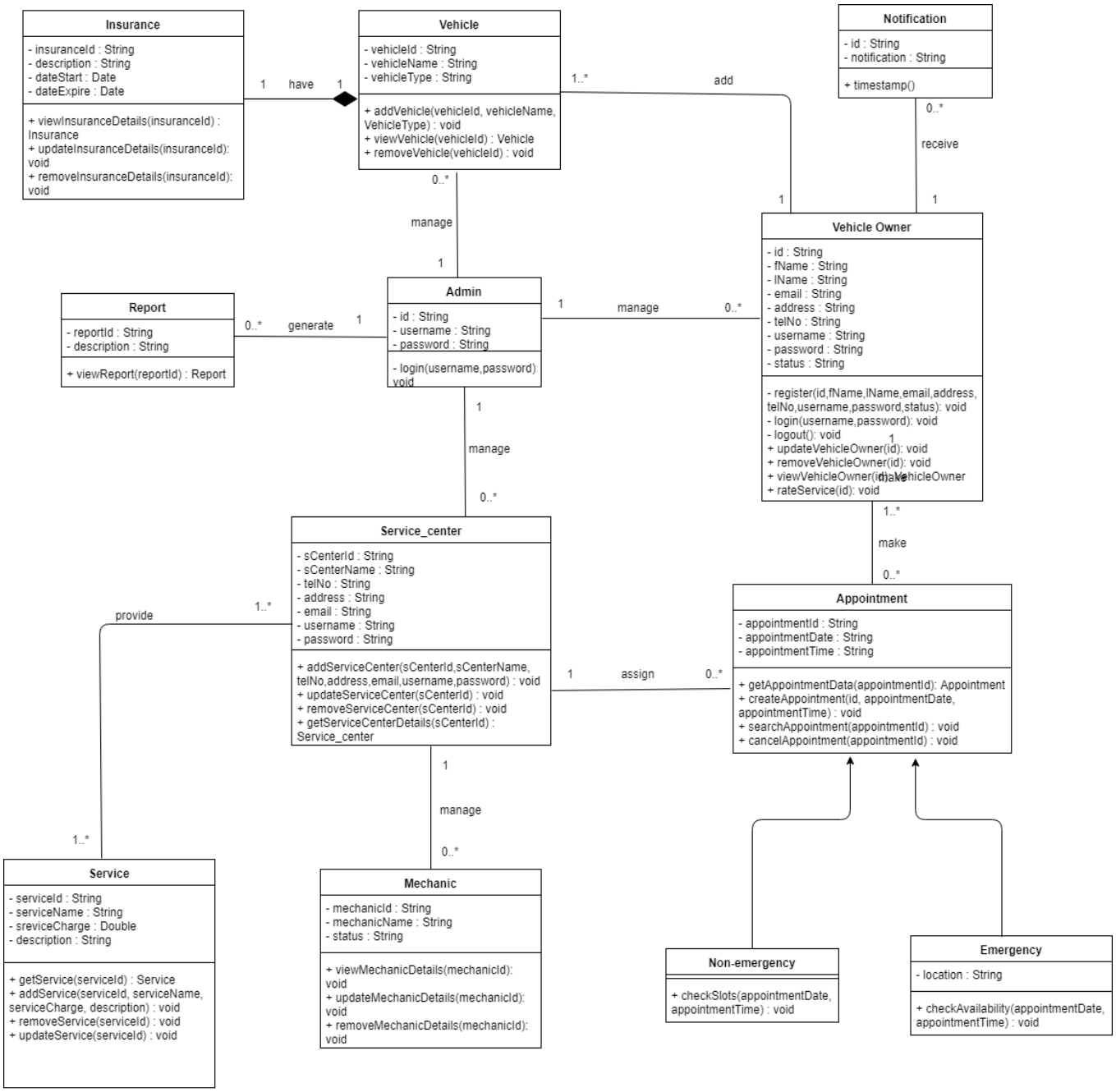
The web application would be built using free and open source software such as: HTML, CSS, Javascript. For the backend we are using Laravel PHP framework.

6.3.3 Database Design

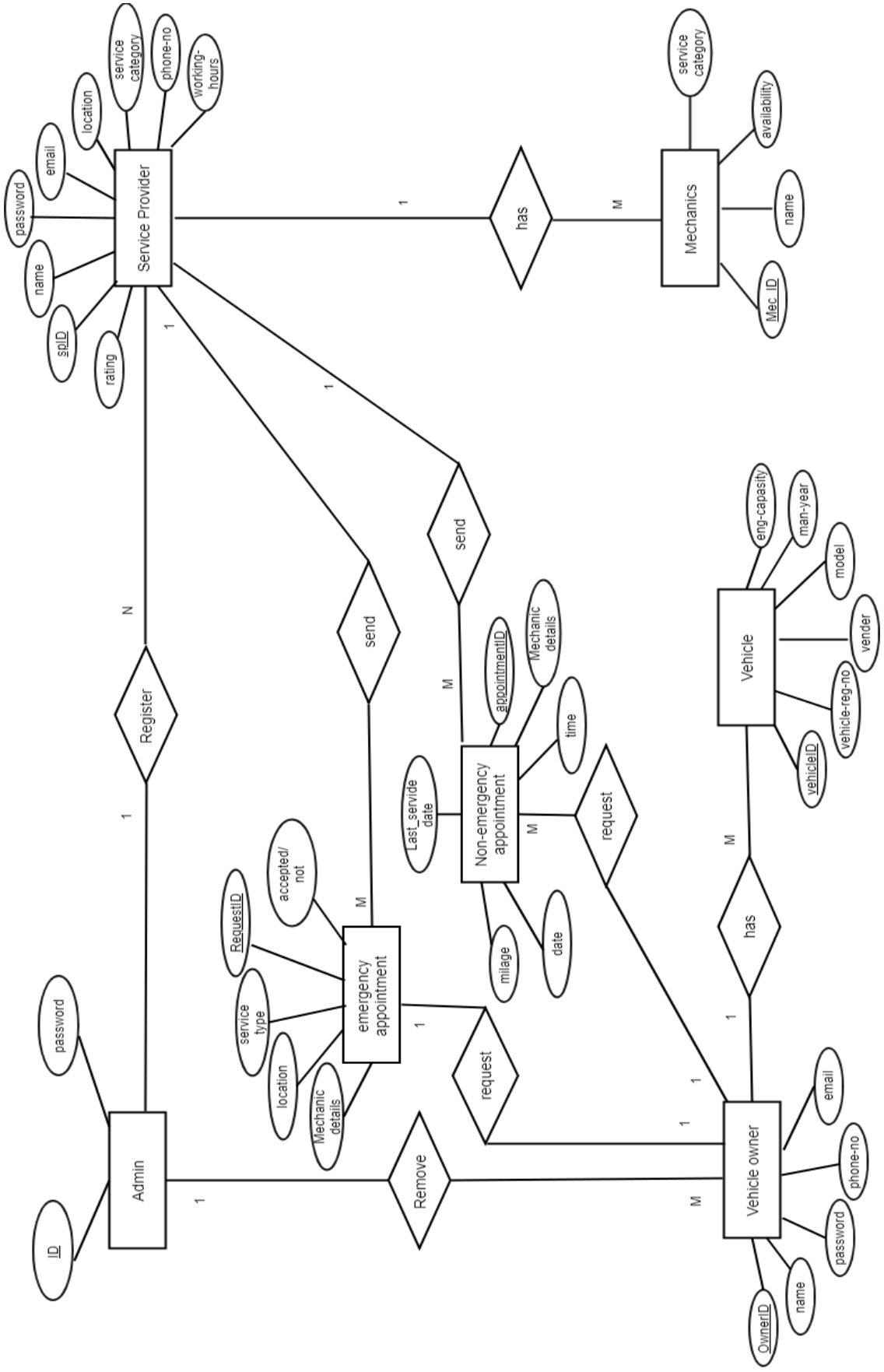
We would be using MySQL to build and serve our database, because it is free, and it has a relatively better performance when performing CRUD operations.

7.0 System Design

7.1 Class Diagram

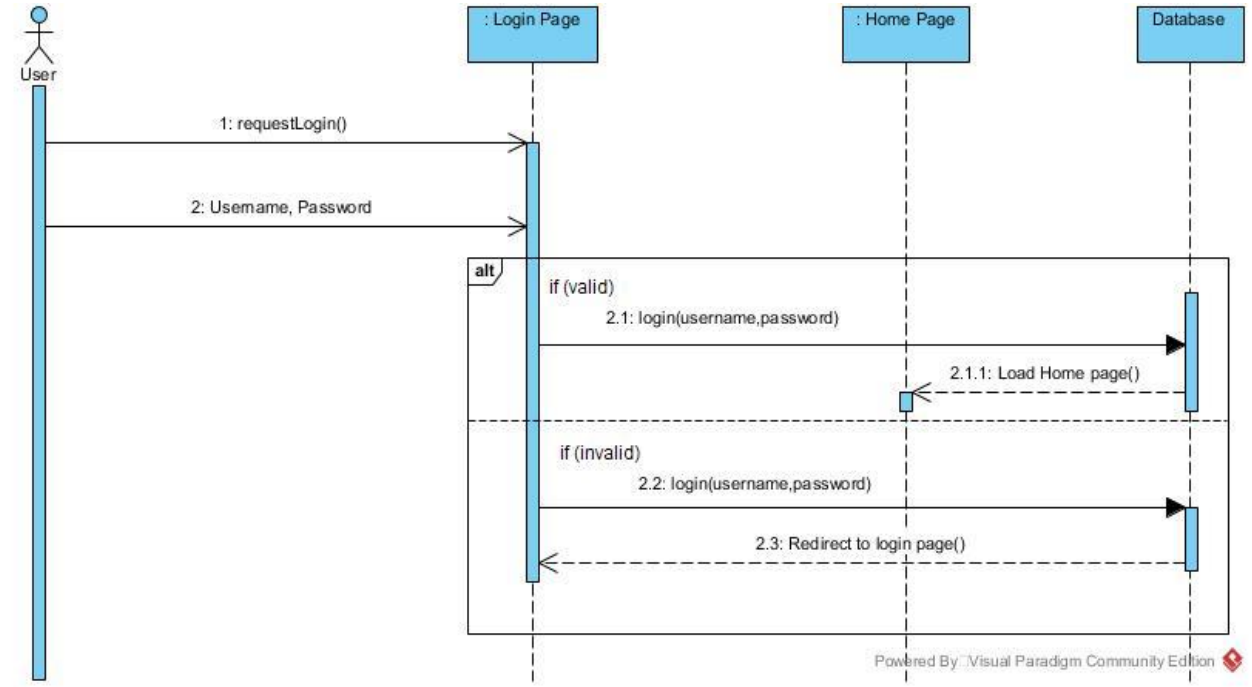


7.2 ER Diagram

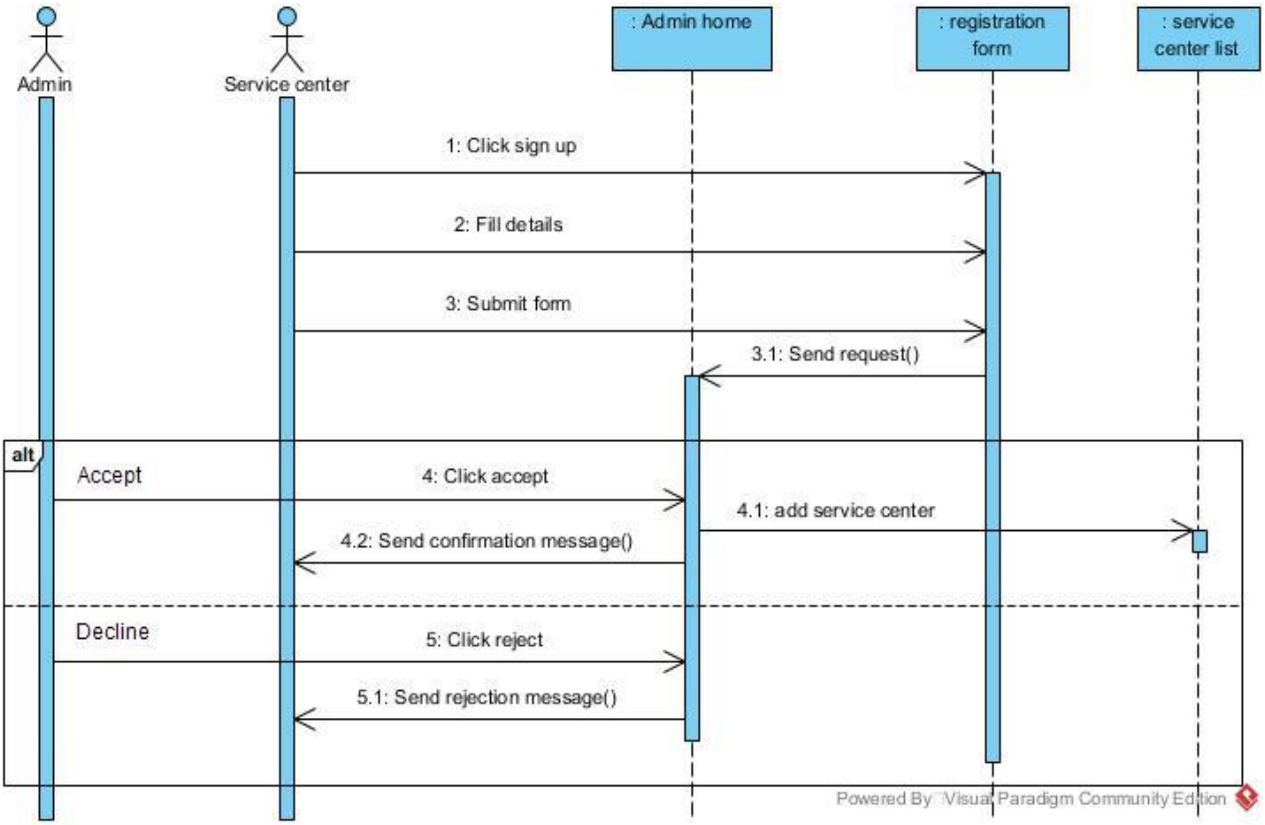


7.3 Sequence Diagrams

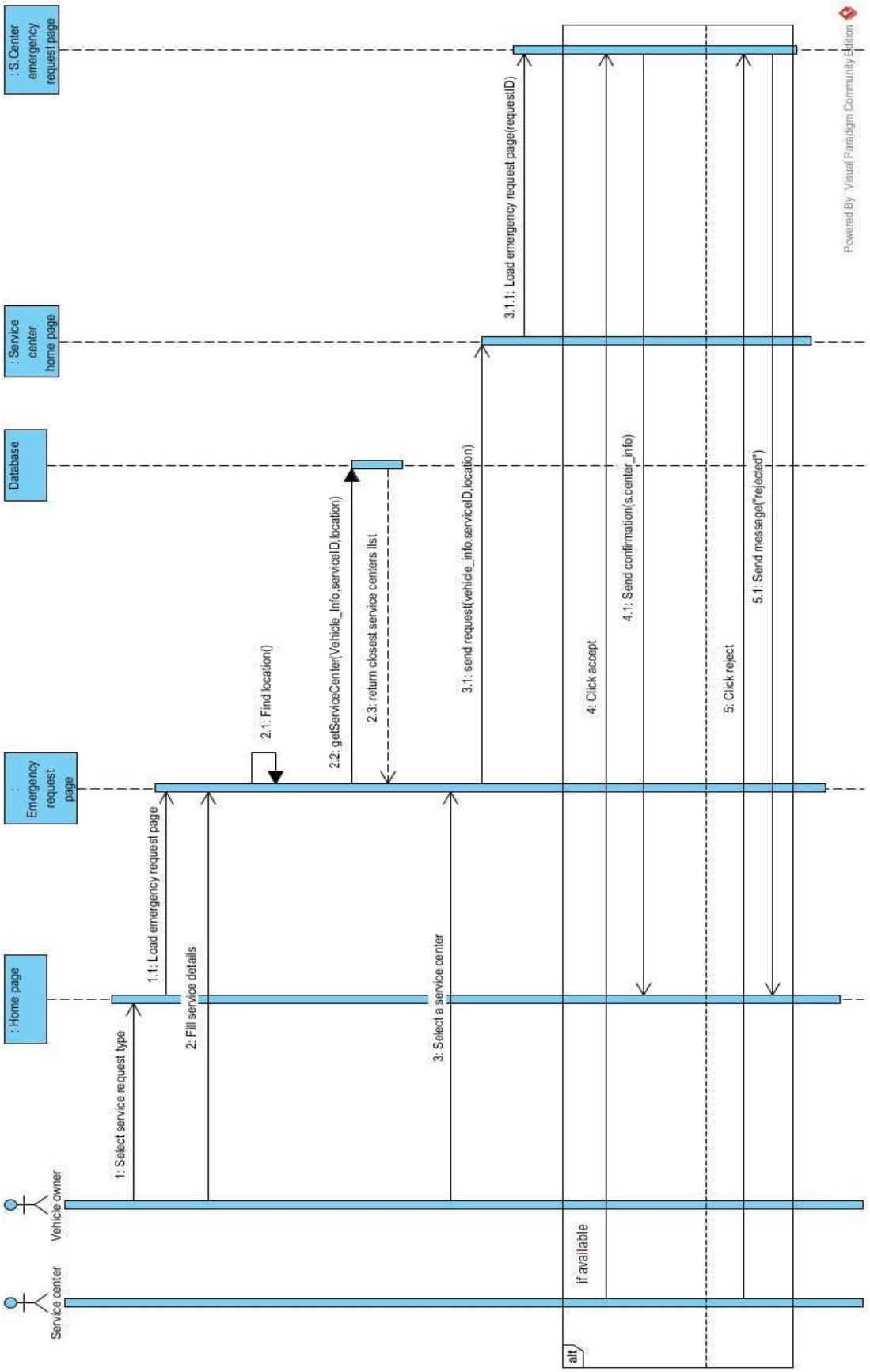
7.3.1 Login



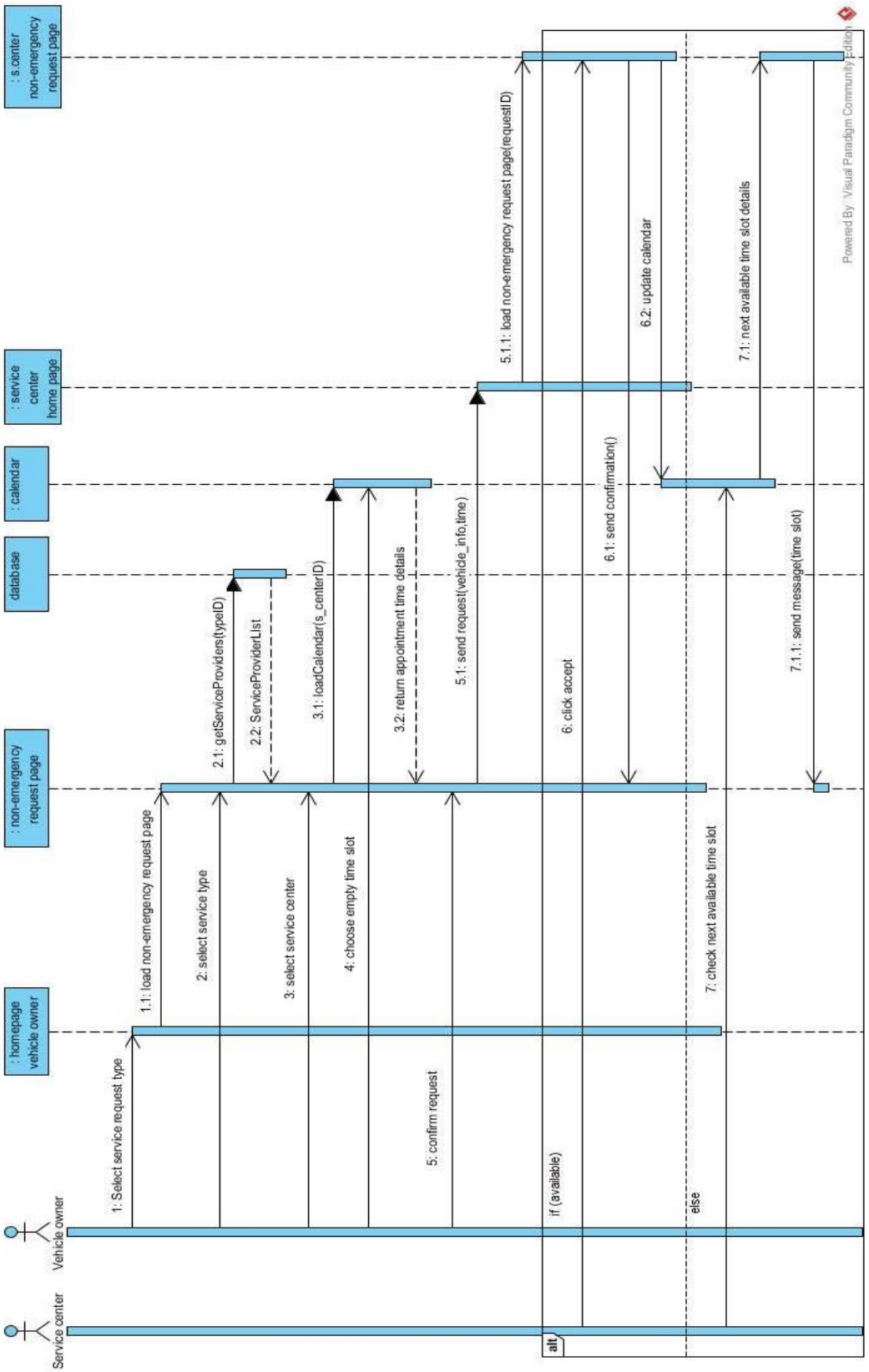
7.3.2 Send register request (by service center)



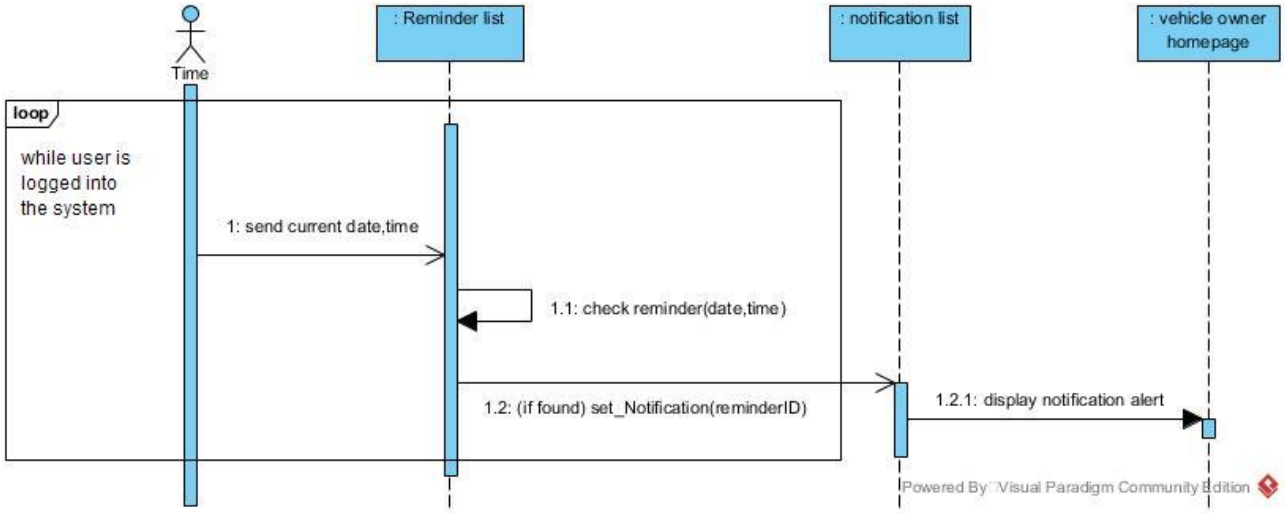
7.3.3 Send emergency service request



7.3.4 Send non-emergency service appointment request

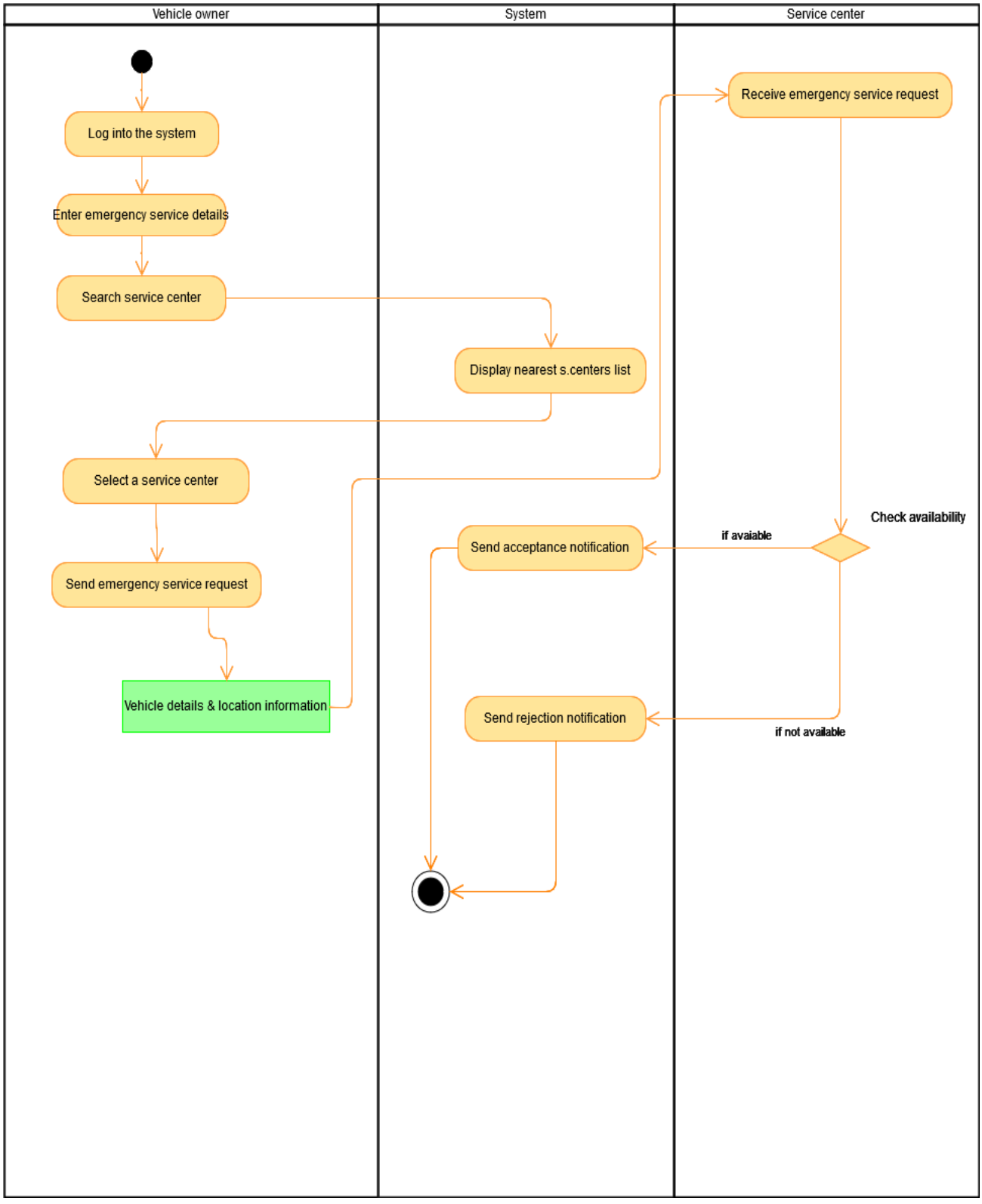


7.3.5 Reminder generation (for vehicle owner)

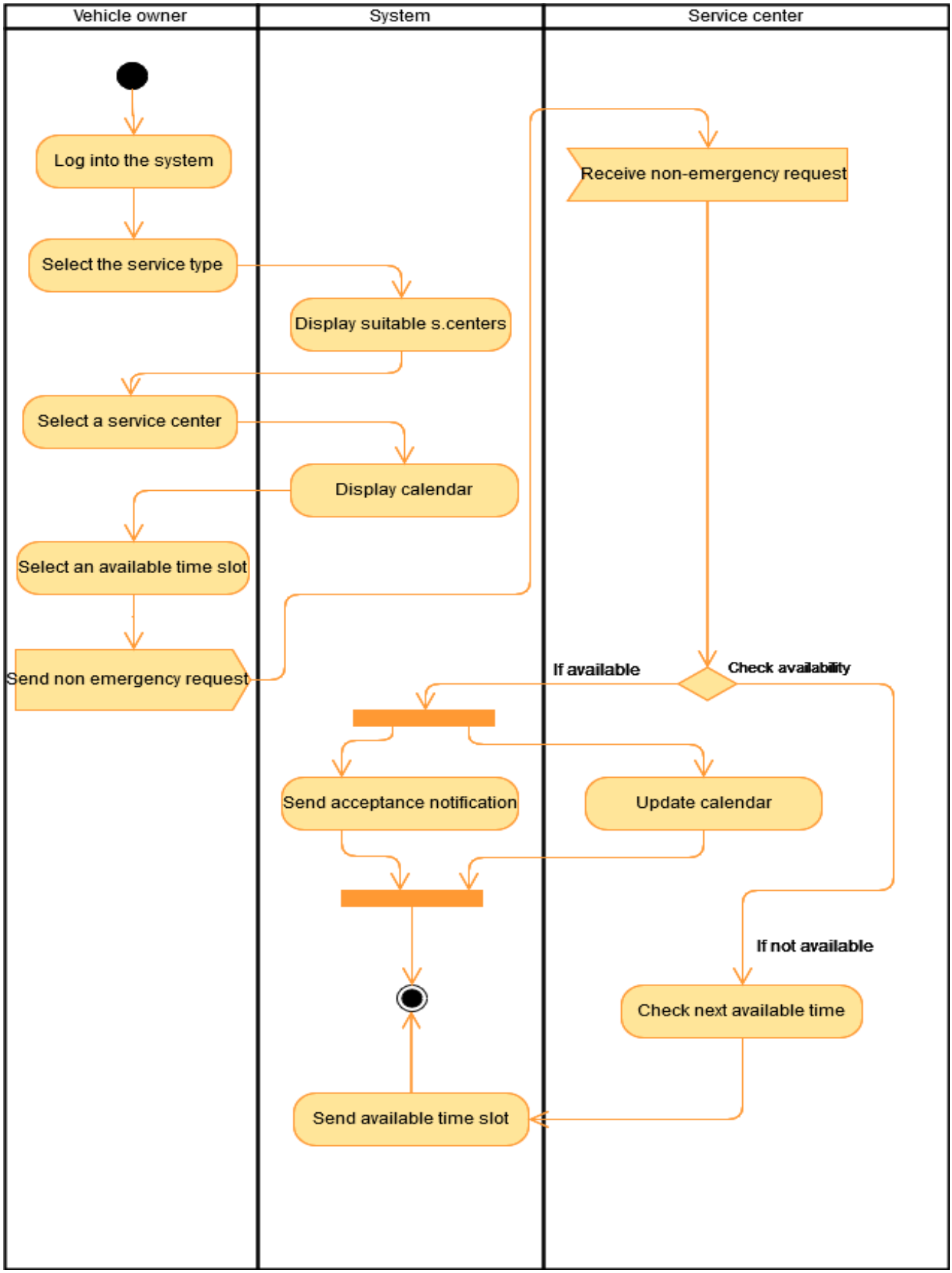


7.4 Activity Diagrams

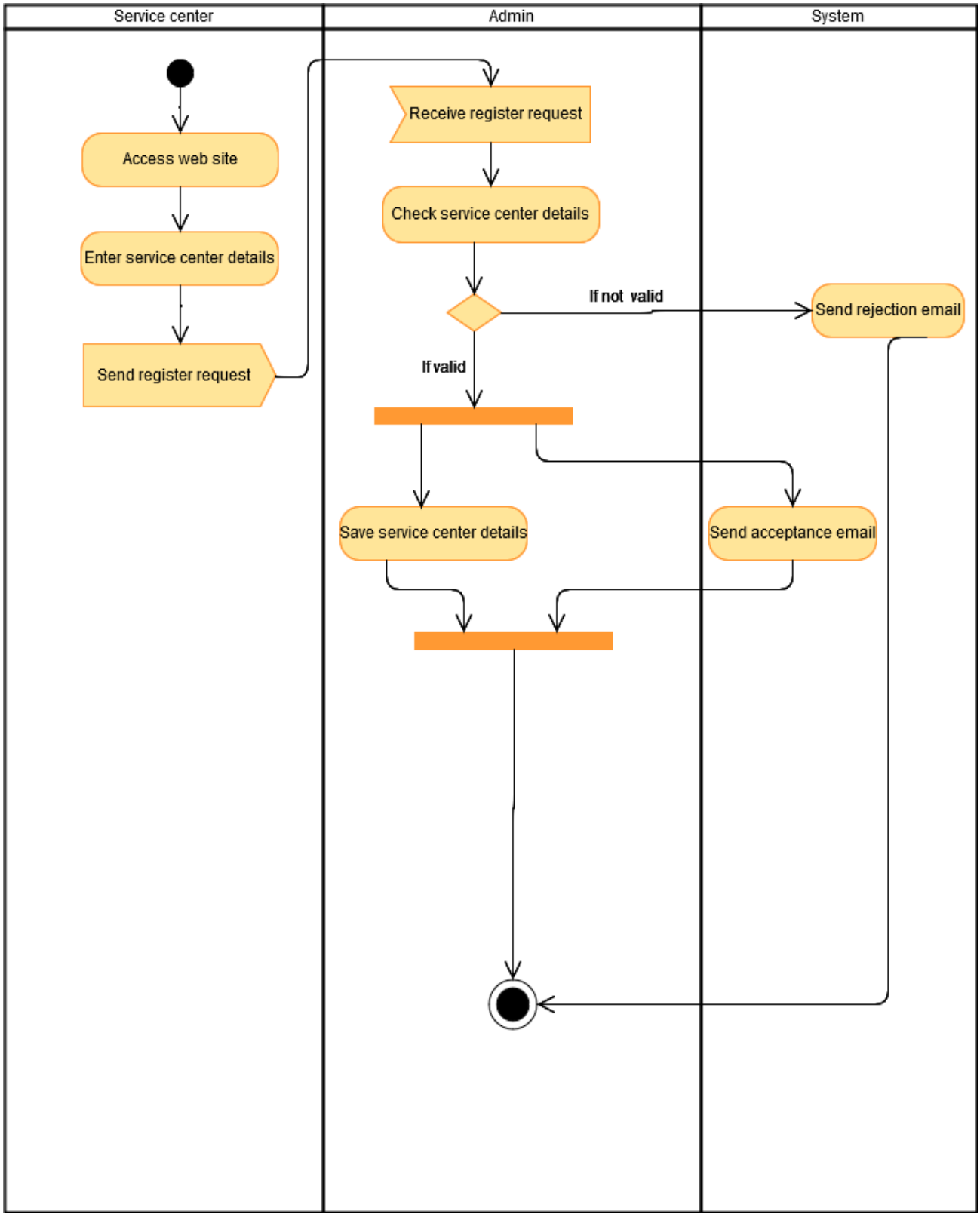
7.4.1 Making an emergency service request



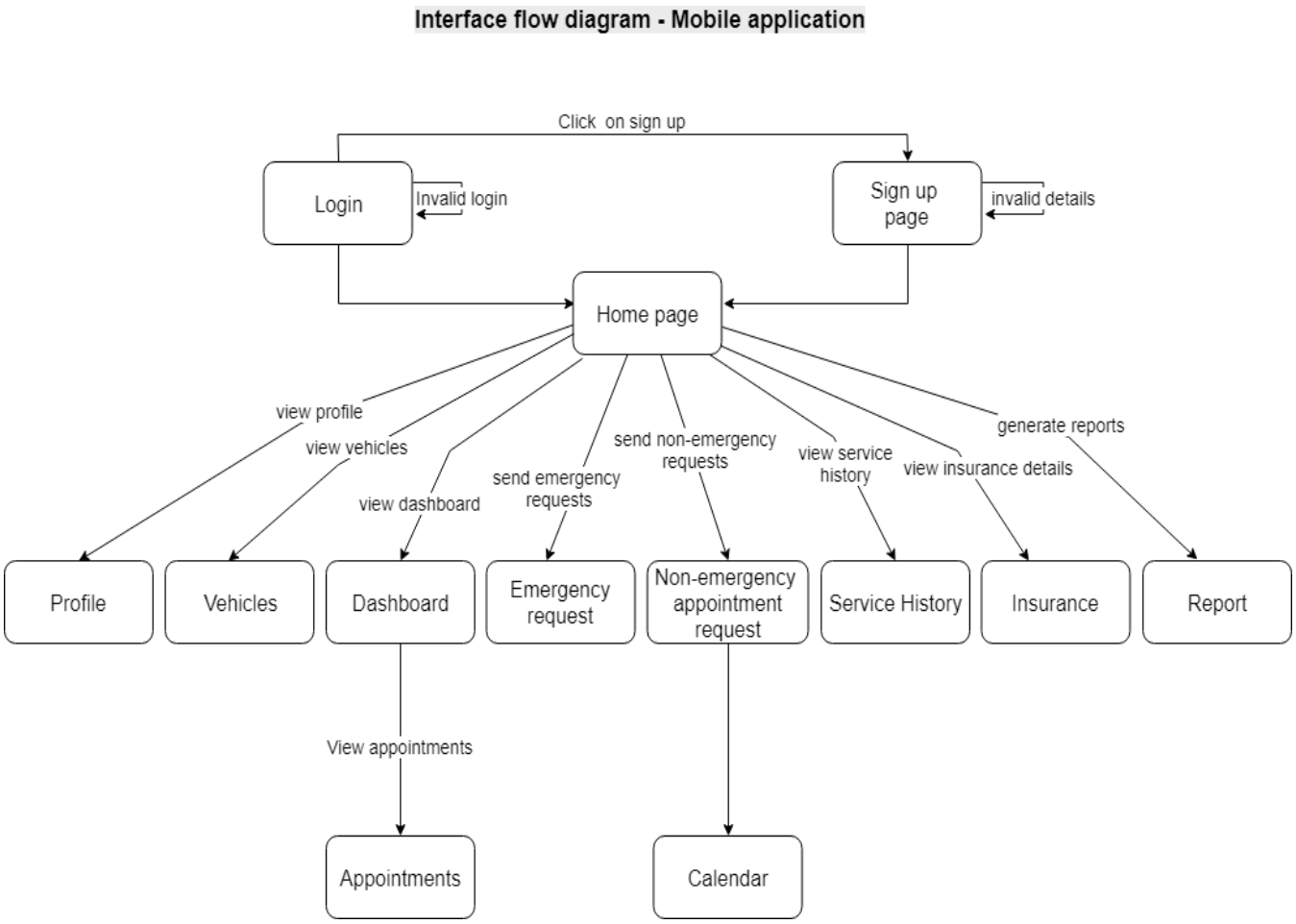
7.4.2 Booking a non-emergency service appointment

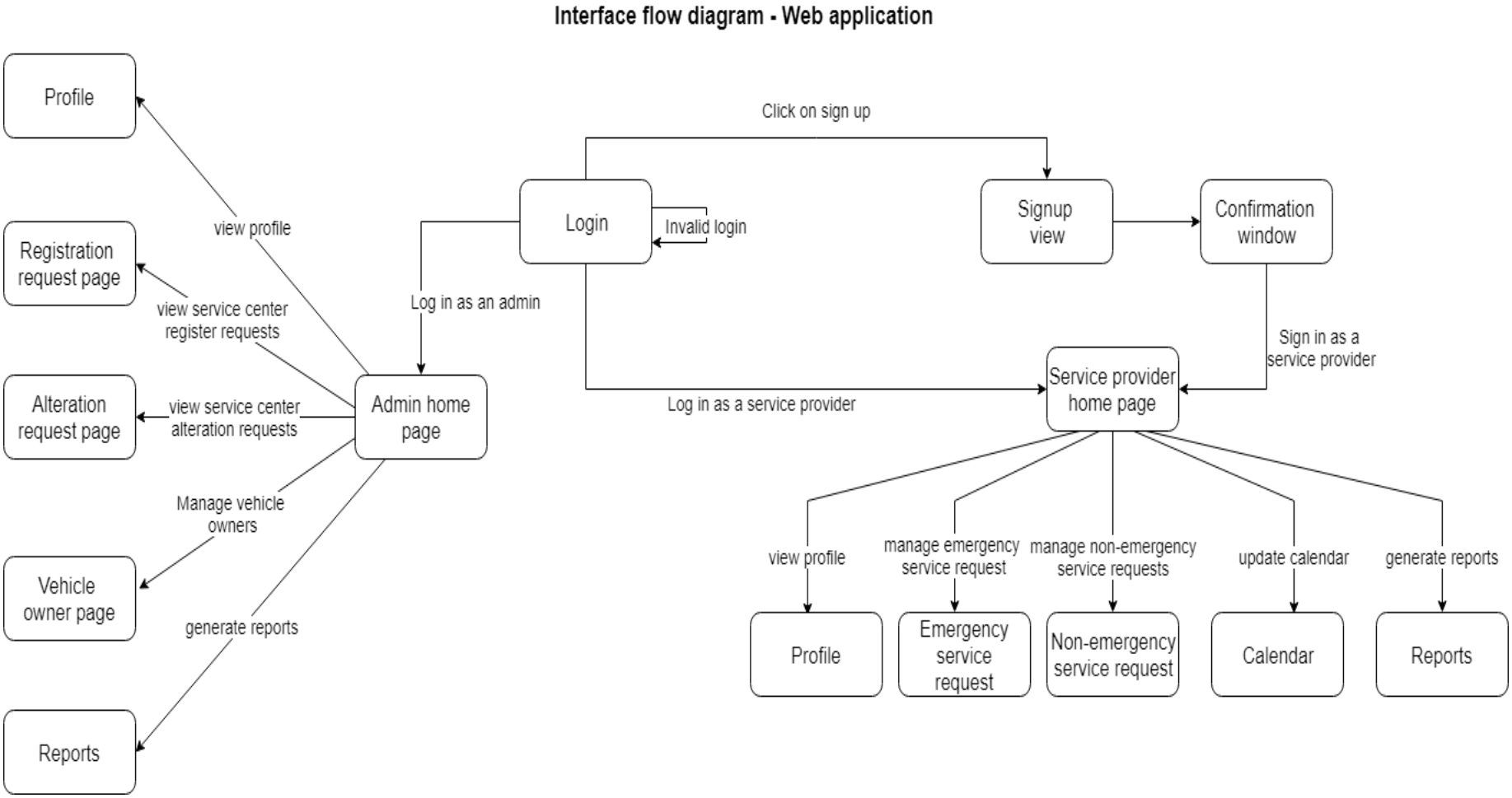


7.4.3 Service center registration



8.0 User Interface Flow Diagram

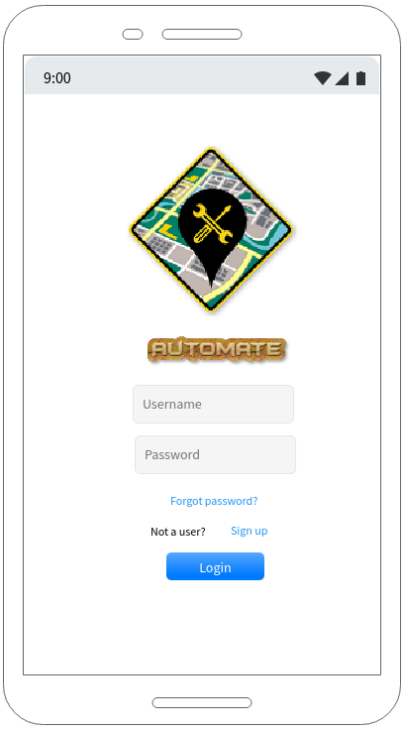



Interface flow diagram - Web application
Profile
Registration request page
Alteration request page
Vehicle owner page
Reports
view profile
view service center register requests
view service center alteration requests
Manage vehicle owners
generate reports
Admin home page
Log in as an admin
Login
Invalid login
Click on sign up
Signup view
Confirmation window
Sign in as a service provider
Log in as a service provider
Service provider home page
view profile
manage emergency service request
manage non-emergency service requests
update calendar
generate reports
Profile
Emergency service request
Non-emergency service request
Calendar
Reports

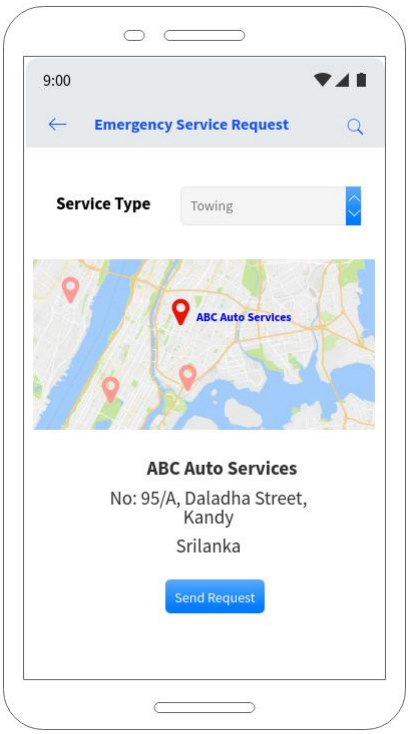
9.0 User Interfaces

These are the sample UI wireframes that we have created during our system design phase.

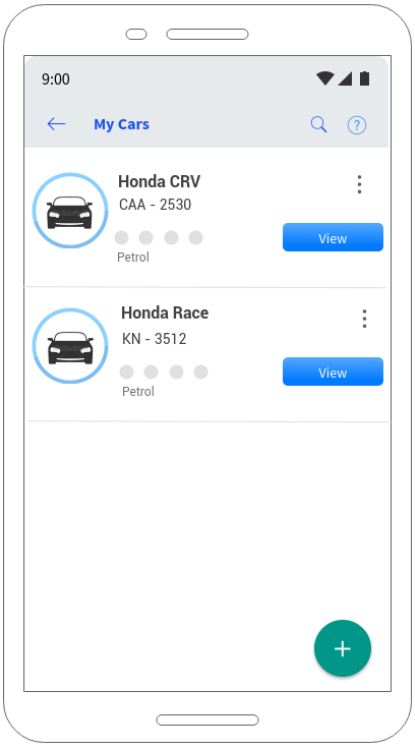
9.1 Mobile Platform (Vehicle owner)



Login Page



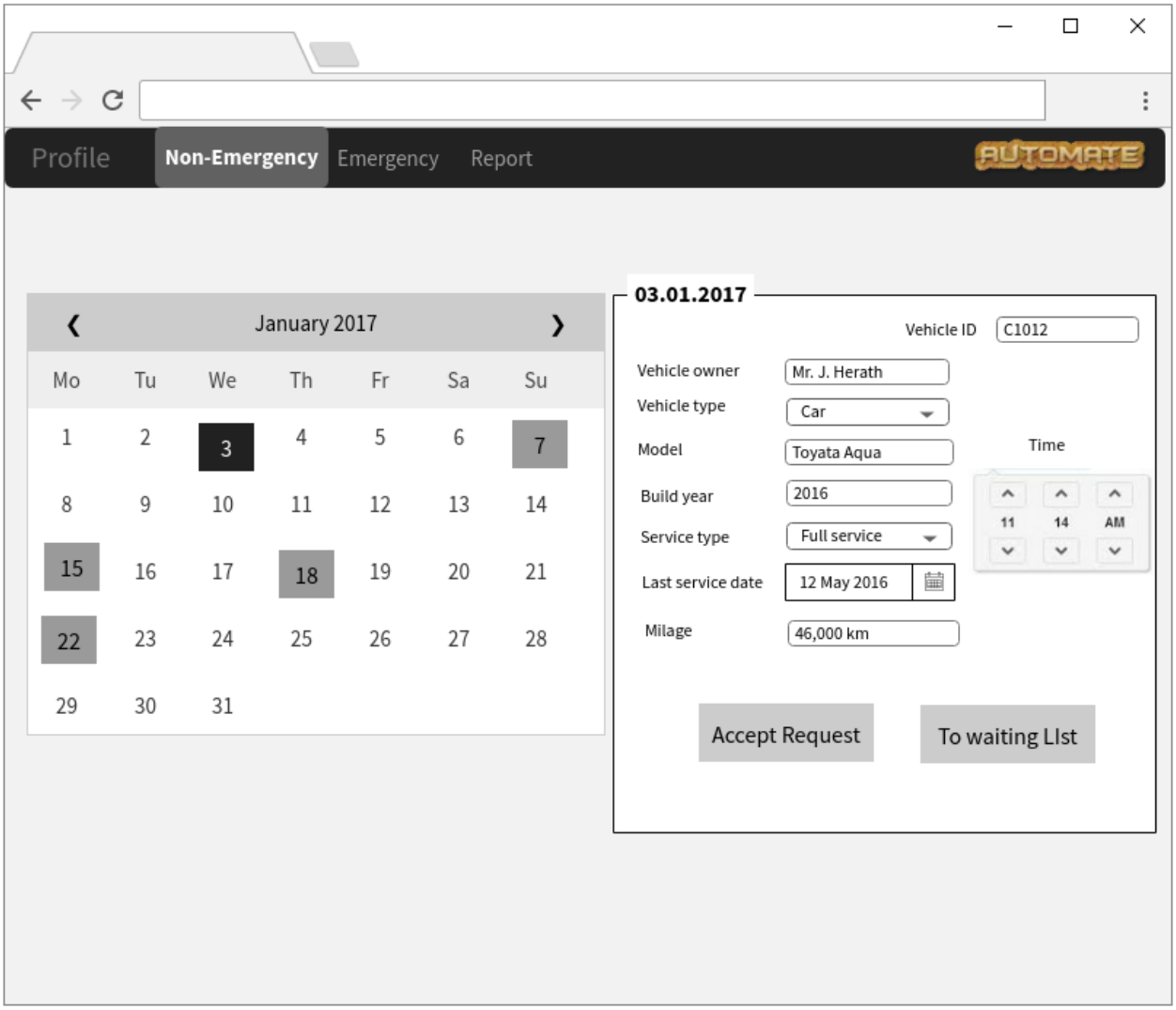
Emergency Request Page



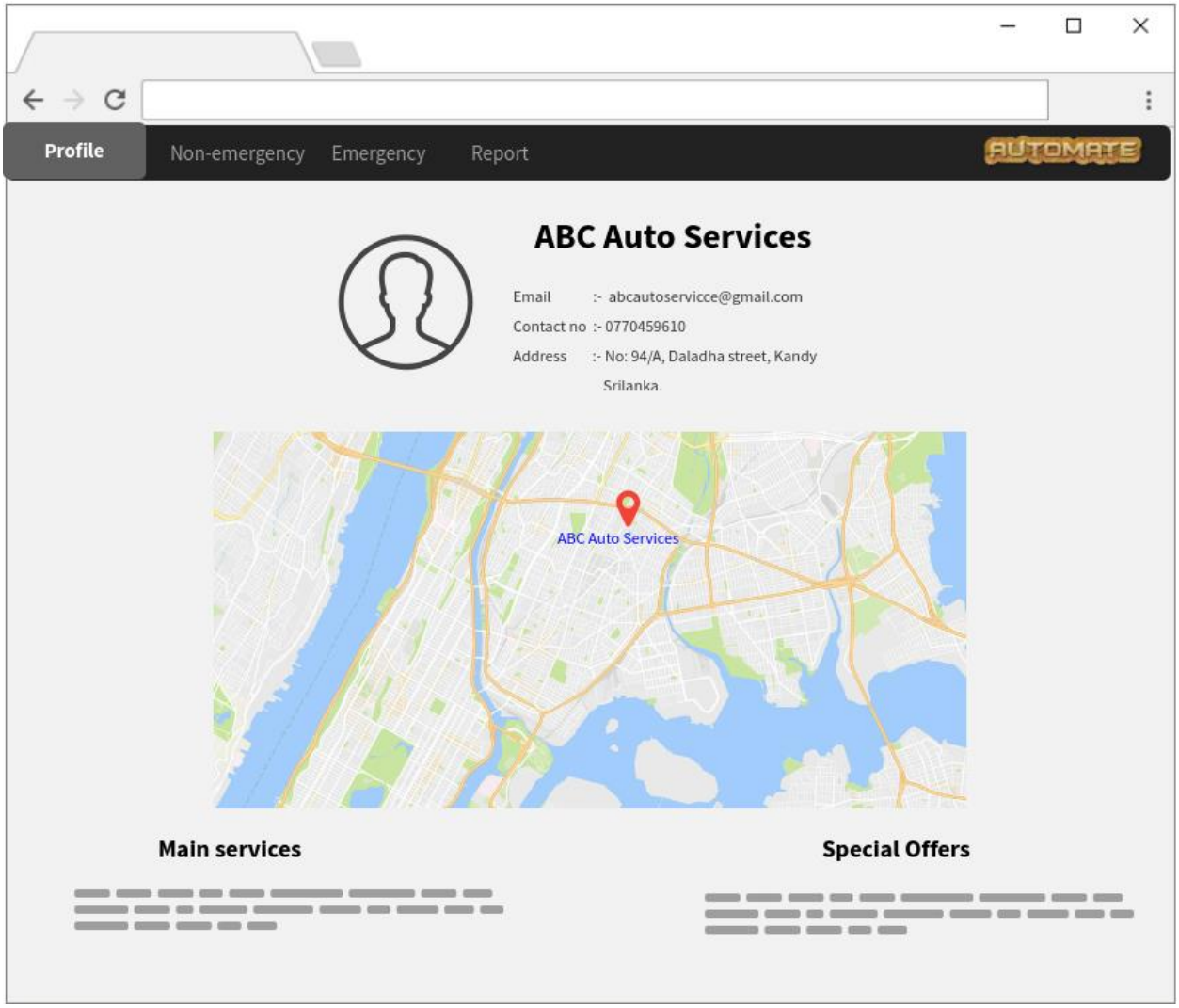
View Vehicles Page

9.2 Web Platform (Service center)

Non-emergency Appointments Page



Profile Page



Report Generation Page