**Vehicle Reservation System**

**Use Case Document**



|  |  |  |  |
| --- | --- | --- | --- |
|  | Prepared By / Last Updated By | Reviewed by | Approved By |
| Name |  |  |  |
| Role | Team Member | Team Member | Scrum Master |
| Signature |  |  |  |
| Date |  |  |  |

Table of Contents

[1.0 Introduction 3](#_Toc5286011)

[1.1 Purpose & Scope of the document 3](#_Toc5286012)

[1.2 Intended Audience 3](#_Toc5286013)

[1.3 Use case ‘Login Homepage’ 3](#_Toc5286014)

[1.3.1 Use case attributes 3](#_Toc5286015)

[1.3.2 Business Rules 4](#_Toc5286016)

[1.3.3 UI Requirements 4](#_Toc5286017)

[1.4 Use case ‘User Credential Authentication’ 3](#_Toc5286014)

[1.4.1 Use case attributes 3](#_Toc5286015)

[1.4.2 Business Rules 4](#_Toc5286016)

[1.4.3 UI Requirements 4](#_Toc5286017)

[1.5 Use case ‘User Registration’ 5](#_Toc5286019)

[1.5.1 Use case attributes 5](#_Toc5286020)

[1.5.2 Business Rules 6](#_Toc5286021)

[1.5.3 UI Requirements 6](#_Toc5286022)

[1.5.3 UI Field Validations 6](#_Toc5286022)

[1.6 Use case ‘User Homepage’ 7](#_Toc5286023)

[1.6.1 Use case attributes 7](#_Toc5286024)

[1.6.2 Business Rules 8](#_Toc5286025)

[1.6.3 UI Requirements 8](#_Toc5286026)

[1.6.4 UI Field Validations 9](#_Toc5286027)

[1.7 Use case ‘Search Vehicle’ 7](#_Toc5286023)

[1.7.1 Use case attributes 7](#_Toc5286024)

[1.7.2 Business Rules 8](#_Toc5286025)

[1.7.3 UI Requirements 8](#_Toc5286026)

[1.7.4 UI Field Validations 9](#_Toc5286027)

[1.8 Use case ‘Vehicle Updation’ 9](#_Toc5286028)

[1.8.1 Use case attributes 9](#_Toc5286029)

[1.8.2 Business Rules 10](#_Toc5286030)

[1.8.3 UI Requirements 10](#_Toc5286031)

[1.8.4 UI Field Validations 12](#_Toc5286032)

[1.9 Use case ‘Vehicle Insertion’ 12](#_Toc5286033)

[1.9.1 Use case attributes 12](#_Toc5286034)

[1.9.2 Business Rules 13](#_Toc5286035)

[1.9.3 UI Requirements 13](#_Toc5286036)

[1.9.4 UI Field Validations 15](#_Toc5286037)

[1.10 Use case ‘Notify User’ 15](#_Toc5286038)

[1.10.1 Use case attributes 15](#_Toc5286039)

[1.10.2 Business Rules 16](#_Toc5286040)

[1.10.3 UI Requirements 16](#_Toc5286041)

[1.10.4 UI Field Validations 17](#_Toc5286042)

[2.0 Database Design 18](#_Toc5286043)

[2.1 Tables Structure 18](#_Toc5286044)

[3.0 Data Flow Diagram 19](#_Toc5286043)

# Introduction

## Purpose & Scope of the document

The purpose of this Use case document is to systematically capture requirements for the project and the system to be developed in terms of use cases. Functional use cases are captured in this document. It also serves as the input for the project scoping.

The scope of this document is limited to addressing the use cases from a user, quality, and non-functional perspective.

## Intended Audience

Each member of the project team

## Use case ‘Login Homepage’

### Use case attributes

**Use Case Description:**

This use case deals with the Login Homepage based on user credentials. The ‘user’ here shall be the operator of the system and will be keying in the user information into the system.

**Scope:**

* Login Homepage

**Actors:**

* User – the operator

**Trigger:**

Click ‘Login’ link, after keying in ‘Username’ & ‘Password’ field.

**Pre-Condition:**

User being able to access the User Homepage.

**Post Condition:**

User is in the User Homepage from where it can choose all others option present in Homepage.

**Flow of Events:**

User at Login Page 🡪 Key in ‘Username’ & ‘Password’ field 🡪 User credentials are validated 🡪 User Home page is displayed.

**Primary Scenario:**

User – is able click ‘Login’ link, after keying in ‘Username’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

**Alternative Scenario:**

User – is able click ‘Login’ link, after keying in ‘Username’ & ‘Password’ field and unable to get his credentials authenticated. The user is presented with relevant error messages: Invalid username or password.

### Business Rules

Business rules should be defined using the following attributes:-

* User – is able click ‘Login’ link, after keying in ‘Username’’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

### UI Requirements

Here is a prototype on how the Login Homepage should look like.

## 1.4 Use case ‘User Credential Authentication’

### 1.4.1 Use case attributes

**Use Case Description:**

This use case deals with the capture of user details. The ‘user’ here shall be the operator of the system and will be keying in the user details.

**Scope:**

* User credential authentication

**Actors:**

* User – the operator

**Trigger:**

Click ‘Login’ link, after keying in ‘Username’ & ‘Password’ field.

**Pre-Condition:**

User being able to access the homepage & get redirected to the ‘User Registration’ page upon click of ‘New User’ link on the login page.

**Post Condition:**

User is in the login page and can access its options.

**Flow of Events:**

User at Login page 🡪 Click ‘New User’ link 🡪 User is in the registration page 🡪 User details are submitted and added onto the database.

### 1.4.2 Business Rules

Business rules should be defined using the following attributes:-

* A user – is able click ‘Login’ link, after keying in ‘Username’’ & ‘Password’ field and get his credentials authenticated with the existing database entry.

### 1.4.3 UI Requirements

Here is a prototype on how the Login Homepage should look like.

## 1.5 Use case ‘User Registration’

### 1.5.1 Use case attributes

**Use Case Description:**

This use case deals with the capture of user details. The ‘user’ here shall be the operator of the system and will be keying in the user details.

**Scope:**

* User registration

**Actors:**

* User – the operator

**Trigger:**

Click ‘New User’ button in the ‘User Registration’ page

**Pre-Condition:**

User being able to access the homepage & get redirected to the ‘User Registration’ page upon click of ‘New User’ link on the login page.

**Post Condition:**

User is in the registration page & submit details

**Flow of Events:**

User at Login page 🡪 Click ‘New User’ link 🡪 User is in the registration page 🡪 User details are submitted and added onto the database.

**Primary Scenario:**

User – is able to access login page facilities. after keying in ‘Username’’ & ‘Password’ field and get his credentials authenticated with the existing database entry

### 1.5.2 Business Rules

Business rules should be defined using the following attributes: -

* When the user clicks on the register link, it should re-direct to registration form.
  + - User needs to fill some of the basic attributes/fields as mentioned below in requirement: Employee Id, First Name, Last Name, Gender, Contract number, Email Id, Password and Branch.
* Clicking ‘New User’ should validate the datatype constraints for each field
* Post-successful field level validation, save the information in the database
* Upon saving the information in the database, display the message ‘Your have successfully registered’.

### 1.5.3 UI Requirements

Here is a prototype on how the Registration Page should look like.

## 1.6 Use case ‘User Homepage’

### 1.6.1 Use case attributes

**Use Case Description:**

This use case deals with the homepage used by User. The ‘User’ here shall be the operator of the system and will be keying in the homepage details.

**Scope:**

* User Homepage

**Actors:**

* User – the operator

**Trigger:**

User should get redirected to the ‘User Homepage’ page upon clicking on the ‘Login’ link.

**Pre-Condition:**

User should be able to get redirected to the ‘User Homepage’ page upon clicking on the ‘Login’ link present on the homepage.

**Post Condition:**

User is in the User Homepage page & can access the user facilities.

**Flow of Events:**

User at homepage 🡪 User login as ‘User’ 🡪 User in the login page

**Primary Scenario:**

User – is able to click ‘Search Vehicle’ link and able to provide module details and get module registered into the system.

### 1.6.2 Business Rules

Business rules should be defined using the following attributes: -

* User on logging in should have Update vehicle details Button.
* User on logging in should have the search functionality.

### 1.6.3 UI Requirements

Here is a prototype on how the User Homepage should look like.

### 1.6.4 UI Field Validations

Please refer to the below requirements for field level validations:

* User on logging in should have Update vehicle details Button.
* User on logging in should have the search functionality.

## 1.7 Use case ‘Search Vehicle’

### 1.7.1 Use case attributes

**Use Case Description:**

This use case deals with the search vehicle based on certain attributes used by User. The ‘User’ here shall be the operator of the system and will be keying in the homepage details.

**Scope:**

* User Homepage

**Actors:**

* User – the operator

**Trigger:**

User should get redirected to the ‘User Homepage’ page upon clicking on the ‘Login’ link.

**Pre-Condition:**

User should be able to get redirected to the ‘User Homepage’ page upon clicking on the ‘Login’ link present on the homepage.

**Post Condition:**

User is in the User Homepage page & can access the user facility for searching the vehicle.

**Flow of Events:**

User at homepage 🡪 User login as ‘User’ 🡪 User in the login page 🡪 Search Vehicle

**Primary Scenario:**

User – is able to click ‘Search Vehicle’ link and able to provide module details and get module registered into the system.

### 1.7.2 Business Rules

Business rules should be defined using the following attributes: -

* When the user clicks on the search textbox, and inserts the search by(vehicle number, branch, insurance expiry, service due, last service, vehicle type) option, the corresponding list appears under vehicle number, branch, insurance expiry, service due, last service, vehicle type.
* Double Clicking the vehicle entry in the list should direct to the update page.
* If Expiry insurance or service due falls in next 15 days from the Vehicle details listed, should be highlighted in red.

### 1.7.3 UI Requirements

Here is a prototype on how the Search Vehicle page should look like.

### 1.7.4 UI Field Validations

Please refer to the below requirements for field level validations:

* Any field could be the 'search by' attribute.
* Insurance expiry, service due, last service must be of be of date type.
* Vehicle number, branch and vehicle type must be of alphanumeric type.

## 1.8 Use case ‘Vehicle Updation’

### 1.8.1 Use case attributes

**Use Case Description:**

This use case deals with the updation of vehicle details by User. The ‘user’ here shall be the operator of the system and will be responsible for the updation of vehicle details.

**Scope:**

* Vehicle Updation

**Actors:**

* User – the operator

**Trigger:**

User should get redirected to the ‘Vehicle Updation’ page upon clicking on the ‘Update’ link.

**Pre-Condition:**

User should be able to get redirected to the ‘Vehicle Updation’ page upon clicking the ‘Update’ link on the admin page.

**Post Condition:**

User is able to update the vehicle details and submit.

**Flow of Events:**

User at homepage 🡪 Login as ‘user’ 🡪 User in the user homepage 🡪 Click ‘update' 🡪 Updated details are submitted and added into the database.

**Primary Scenario:**

User – is able to click ‘Vehicle Updation’ link and able to update vehicle details and get registered into the system.

### 1.8.2 Business Rules

Business rules should be defined using the following attributes: -

* When the user clicks on the 'Update' link on homepage, it should re-direct to vehicle page.
* Update Details page should display all the fields- Vehicle No, vehicle type, Branch, insurance type, insurance expiry date, last Serviced Date, Service due date.
* Admin should have access to update details of existing vehicle details. Admin on entering existing Vehicle Number, the rest of the details should be loaded in the respective labels from database.
* On creating/updating the vehicle details, each fields should be validated and appropriate error messages should be thrown.
* Admin should have access to update details of existing vehicle details.
* On creating/updating the vehicle details, each fields should be validated and appropriate error messages should be thrown.

### 1.8.3 UI Requirements

Here is a prototype on how the Vehicle Updation page should look like.

## 1.9 Use case ‘Vehicle Insertion’

### 1.9.1 Use case attributes

**Use Case Description:**

This use case deals with the addition of new vehicle entry. The ‘user’ here shall be the operator of the system and will be responsible for the insertion of the new vehicle.

**Scope:**

* Vehicle Insertion

**Actors:**

* User – the operator

**Trigger:**

User should get redirected to the Vehicle Insertion’ page upon clicking on the ‘Add Vehicle’ link.

**Pre-Condition:**

User should be able to get redirected to the ‘Add Vehicle’ page upon clicking the ‘Add Vehicle’ link on the admin page.

**Post Condition:**

User is able to add new vehicle details and submit.

**Flow of Events:**

User at homepage 🡪 User login as ‘User’ 🡪User in the User Homepage 🡪 Click 'Update’ link 🡪 Click 'Add Product' link. New entry of the vehicle is inserted and added into the database on clicking 'submit'.

**Primary Scenario:**

User – is able to click ‘Add Vehicle’ link and able to new vehicle details and get registered into the system.

### 1.9.2 Business Rules

Business rules should be defined using the following attributes: -

* When the user clicks on the Add Vehicle link, it should re-direct to the page for new vehicle insertion.
* Admin needs to fill some of the basic attributes/fields as mentioned below in requirement: Vehicle No, vehicle type, Branch, insurance type, insurance expiry date, last Serviced Date, Service due date.
* Clicking ‘Submit’ should validate the datatype constraints for each field.
* Post-successful field level validation, save the information in the database.
* Upon saving the information in the database, display the message ’You have been successfully registered.'

### 1.9.3 UI Requirements

Here is a prototype on how the Add Vehicle page should look like.

### 1.9.4 UI Field Validations

Please refer to the below requirements for field level validations:

* All fields are mandatory.
* Vehicle Number must be of 15 digits’ length and have Alphanumeric digits.
* Branch must be of 15 digits’ length and have Alphanumeric digits.
* Vehicle Type must be of 8 digits’ length and have only Alphabetic digits.
* Insurance type must be of 8 digits’ length and have only Alphabetic digits.
* Insurance expiry, last Serviced Date, Service due date must be of date type.

## 1.10 Use case ‘Notify User’

### 1.10.1 Use case attributes

**Use Case Description:**

This use case deals with the ability to generates the mail that should be run fortnightly to trigger Email notification to admin to notify – Expiry of Insurance in next 15 days or Service due in next 15 months.

This use case carefully designs the notification optimizing the details for each vehicle.

**Scope:**

* Email Notification

**Actors:**

* Operator:: Automatic

**Trigger:**

Run fortnightly to trigger Email notification to admin to notify.

**Pre-Condition:**

Email to be sent automatically as per the given condition i.e.it be run fortnightly to trigger

Email notification to admin to notify – Expiry of Insurance in next 15 days or Service due in next 15 months.

**Post Condition:**

User is sent an email notification.

**Flow of Events:**

User at homepage 🡪 User login as ‘User’ 🡪 User in the user home page 🡪 Click ‘Search' link 🡪 Automatic sending of email to users that satisfy the condition.

**Primary Scenario:**

Vehicle selected: Automatic email generated.

### 1.10.2 Business Rules

Business rules should be defined using the following attributes: -

* + - Batch job should be run fortnightly to trigger Email notification to admin to notify – Expiry of Insurance in next 15 days or Service due in next 15 months

### 1.10.3 UI Requirements

Here is a prototype on how the 'E-mail notification’ page should look like.

# 2.0 Database Design

## 2.1 Tables Structure

**User Details:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| EmpId | varchar | 50 | N |
| firstName | varchar | 50 | N |
| lastName | varchar | 15 | N |
| age | int | 2 | N |
| gender | drop down | na | N |
| contactNumber | varchar | 15 | N |
| emailid | varchar | 10 | N |
| password | varchar | 15 | N |
| branch | varchar | 30 | N |

**Vehicle Details:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| vehicleNo | varchar | 50 | N |
| branch | varchar | 50 | N |
| vehicleType | varchar | 15 | N |
| Insurance Expiry date | date | 10 | N |
| Last Serviced Date | date | 10 | N |
| Service Due date | date | 10 | N |

# 3.0 Data Flow Diagram

