Software Requirements Documents

# Project Title – Vehicle Doctor

# Revision History

# Team Structure

|  |  |  |  |
| --- | --- | --- | --- |
| Employee Name | Employee ID | Email ID | Mobile |
| Prajwal Agarwal | 8107900 | [prajwalagarwal@virtusa.com](mailto:prajwalagarwal@virtusa.com) | 8960733960 |
| Mohit Kushwah | 8107901 | [mohitkushwah@virtusa.com](mailto:mohitkushwah@virtusa.com) | 8668519500 |
| Shreenath Baral | 8107931 | [shreenathb@virtusa.com](mailto:shreenathb@virtusa.com) | 8979709032 |

# Overview of the Project

Vehicle Doctor allows customers to schedule mechanics at their own convenience for their vehicle while giving garage owners the display of current availability of their staffs, previous orders, invoices and feedbacks. Tasks that are needed to be completed are displayed to the mechanics along with the information like customer address, etc.

# Actors of the System

|  |  |
| --- | --- |
| Actor | Responsibilities |
| Customer | Customer is a human who interacts with the system using the interface provided, and majorly makes the orders in the system that’s going to be the key business driver of the entire system. |
| Garage Owner | Garage Owner is a human who interacts with the system using the interface provided, and majorly manages the mechanics that provide the service to Customers. |
| Mechanic | Mechanic is a human who interacts with the system and provide the service to Customers. |

# Functional Requirements

## Customer Test Cases

|  |  |
| --- | --- |
| Use Case #1 | Customer registers in the system |
| Actor | Customer |
| Successful Flow | 1. The customer lands on the home page of the system 2. The customer views the option either to register in the systems 3. The customer selects to register. 4. The system displays option to enter the first name, last name, email id, mobile and address etc. 5. Upon submission, the system displays a registration success message, along with customer details entered in the previous step. |
| Exceptional Flow | 1. When invalid characters are entered first name, last name, email id, mobile and address line, system throws the relevant error message 2. If the registration process is not successful for any reason, system informs the user with relevant error message. |
| Outcome | Customer account is created in the system |

|  |  |
| --- | --- |
| Use Case #2 | Customer Login into the system |
| Actor | Customer |
| Successful Flow | 1. The customer lands on the home page of the system 2. The customer views the option either to login in the systems 3. The customer selects to login. 4. The customer enters the email id and password. |
| Exceptional Flow | 1. When invalid email or password is entered. |
| Outcome | Customer successfully login into the system. |

|  |  |
| --- | --- |
| Use Case #3 | Customer Books a Slot |
| Actor | Customer |
| Successful Flow | 1. The customer lands on the login home page. 2. The customer selects the option to book a slot. 3. Customer enter the pick-up location and vehicle details like make, model, type and problem. 4. Customer selects the desired date and time for the visit. 5. Customer submit the order form. |
| Exceptional Flow | 1. Mechanic not available at the desired time slot. |
| Outcome | Customer successfully books the slot |

|  |  |
| --- | --- |
| Use Case #4 | Customer Tracks Current Order |
| Actor | Customer |
| Successful Flow | 1. The customer lands on the login home page. 2. The customer selects the option to track the order. 3. If the order is not started customer can select to cancel the order. 4. On order completion invoice is generated. |
| Exceptional Flow | 1. No current orders |
| Outcome | Customer successfully tracks the order. |

|  |  |
| --- | --- |
| Use Case #5 | Customer Views Order History |
| Actor | Customer |
| Successful Flow | 1. The customer lands on the login home page. 2. The customer selects the option to view previous orders. 3. If the order is not started customer can select to cancel the order. |
| Exceptional Flow | 1. No previous orders |
| Outcome | Customer views previous order. |

|  |  |
| --- | --- |
| Use Case #6 | Customer pays for the orders |
| Actor | Customer |
| Successful Flow | 1. Once mechanic submits the completion status 2. Customer pays for the order |
| Exceptional Flow | 1. No previous orders |
| Outcome | Customer pays for the order |

## Garage Owner

|  |  |
| --- | --- |
| Use Case #1 | Garage Owner Logins into the system |
| Actor | Garage Owner |
| Successful Flow | 1. The Garage Owner lands on the home page of the system 2. The Garage Owner views the option either to register or login in the systems 3. The Garage Owner selects to login. 4. The Garage Owner enters the email id and password. |
| Exceptional Flow | 1. When invalid email or password is entered. |
| Outcome | A Garage Owner successfully login into the system. |

|  |  |
| --- | --- |
| Use Case #2 | Garage Owner manages the Employee |
| Actor | Garage Owner |
| Successful Flow | 1. The Garage Owner lands on the dashboard 2. The Garage Owner selects the option to view employees. 3. The Garage Owner selects to add employees and generate credentials. 4. The Garage Owner enters the details of employee like first name, last name, email id, mobile and address etc. |
| Exceptional Flow | 1. When invalid characters are entered first name, last name, email id, mobile and address line, system throws the relevant error message 2. If the registration process is not successful for any reason, system informs the user with relevant error message. |
| Outcome | Garage Owner successfully enters the employee. |

|  |  |
| --- | --- |
| Use Case #3 | Garage Owner views current status of employees |
| Actor | Garage Owner |
| Successful Flow | 1. The Garage Owner lands on the dashboard 2. The Garage Owner selects the option to view employees. 3. The Garage Owner selects the employee with busy status. |
| Exceptional Flow | 1. No employees added. 2. No employees busy. |
| Outcome | Garage Owner successfully views employee status |

|  |  |
| --- | --- |
| Use Case #4 | Garage Owner views all orders. |
| Actor | Garage Owner |
| Successful Flow | 1. The Garage Owner lands on the dashboard 2. The Garage Owner selects the option to view orders and current order status. |
| Exceptional Flow | 1. No orders to display |
| Outcome | Garage Owner successfully views orders. |

|  |  |
| --- | --- |
| Use Case #4 | Garage Owner assigns the mechanic |
| Actor | Garage Owner |
| Successful Flow | 1. Once Customer places the order garage owner assigns the task to Mechanic based upon availability |
| Exceptional Flow | 1. No orders to assign 2. No mechanic available |
| Outcome | Garage Owner successfully assigns task to mechanic |

## Mechanic Cases

|  |  |
| --- | --- |
| Use Case #1 | Mechanic Logins into the system |
| Actor | Mechanic |
| Successful Flow | 1. Mechanic lands on the home page. 2. Mechanic login using the given credential. |
| Exceptional Flow | 1. Mechanic enters the invalid login credential |
| Outcome | Mechanic successfully logins |

|  |  |
| --- | --- |
| Use Case #2 | Mechanic Views Current Order |
| Actor | Mechanic |
| Successful Flow | 1. Mechanic lands on the dashboard. 2. Mechanic selects the current order option. 3. Mechanic views the problem statement and customer details. 4. Mechanic adds the parts changed and work done. |
| Exceptional Flow | 1. No current order. 2. Mechanic selects wrong part. |
| Outcome | Mechanic successfully submits the form |

|  |  |
| --- | --- |
| Use Case #3 | Mechanic submits the task completed status |
| Actor | Mechanic |
| Successful Flow | 1. Mechanic submits the add work form. 2. Mechanic is shown a set of tasks to complete. 3. Mechanic completes all the tasks and submit |
| Exceptional Flow | 1. Mechanic did not complete all the task |
| Outcome | Mechanic successfully submits the task complete status |

|  |  |
| --- | --- |
| Use Case #4 | Mechanic views all the pending orders |
| Actor | Mechanic |
| Successful Flow | 1. Mechanic lands on the dashboard. 2. Mechanic selects the pending order option. |
| Exceptional Flow | 1. No pending order to display |
| Outcome | Mechanic successfully views all the pending orders. |

# Functional Requirement Summary

|  |  |  |
| --- | --- | --- |
| Use Case No | Use Case Title | Member Name |
|  |  |  |
| C - 1 | Customer Registers in the system | Shreenath |
| C - 2 | Customer Login into the system | Shreenath |
| C - 3 | Customer Books a Slot | Prajwal |
| C – 4 | Customer Tracks Current Order | Mohit |
| C – 5 | Customer Views Order History | Mohit |
| C – 6 | Customer pays for the orders | Mohit |
| G – 1 | Garage Owner Logins into the system | Shreenath |
| G – 2 | Garage Owner manages the Employee | Mohit |
| G – 3 | Garage Owner views current status of employees | Mohit |
| G – 4 | Garage Owner views all orders. | Prajwal |
| G – 5 | Garage Owner assigns the mechanic | Prajwal |
| M – 1 | Mechanic Logins into the system | Shreenath |
| M – 2 | Mechanic Views Current Order | Shreenath |
| M – 3 | Mechanic submits the task completed status | Prajwal |
| M – 4 | Mechanic views all the pending orders | Prajwal |