

Asmeri: A Vehicle Rental Registration Sysytem

Senior Research Report



October , 2022

Brukh Seaye Isaac

In partial fulfillment of the requirement for the award of BS in Computer Science degree.

Advisor: Professor Obinna Kalu

Table of Contents

[**Use Case Diagram** 2](#_Toc75500876)

[**Use case Descriptions** 2](#_Toc75500877)

[**Supplementary Specification** 7](#_Toc75500878)

[**Glossary** 8](#_Toc75500879)

[Introduction 8](#_Toc75500880)

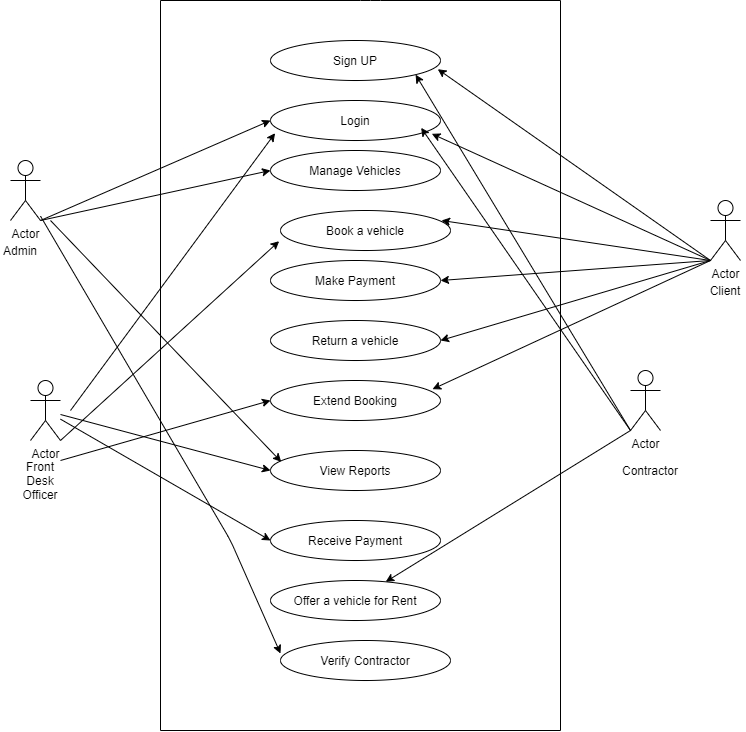
[Definitions 9](#_Toc75500881)

**Conclusion**……………………………………………………………………………………………………………………………………………10

**GitHub Repository Link**

# https://github.com/bisaac3333/SWE.Final.Project.git

# **Use Case Diagram**



# **Use case Descriptions**

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 1** | | |
| **Name      Sign-Up** | | |
| **Brief description**This use case allows to create a profile | | |
| **ActorsAdmin/client/** | | |
|  | | |
| **Preconditions** | | |
| No precondition | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.0 Sign up** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The user calls the sign-up command | The system displays a form with the fields for first name, last name, email, password. |
| 2 | The user fills out the form and requests the system to save the details | The system verify that there is no other profile in the database with the same email address, saves the user and returns the success message. If Another profile exists, it returns a message saying to enter new values. |
| **Postconditions** | | |
| The user is persisted in the system | | |
| **Business Rule** | | |
| No duplicate user profiles. | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 2** | | |
| **Name      Registering an owner and vehicle** | | |
| **Brief description**This use case provides a functionality for registering owners and their Vehicles for an Uber type rides. | | |
| **ActorsAdmin/Front Desk - receptionist** | | |
|  | | |
| **Preconditions** | | |
| The user (**Front Desk - receptionist** ) must logged in to the system | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.1 Adding a Vehicle** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The  user admin calls the add vehicle command | The system displays the vehicle form with model, year, and type, plate number, registration number |
| 2 | The admin fills out the form and requests the system to save the details | The concept is that the system would enter a name, license plate of a person and if a duplicate exists, the a notification would be shown that says it is a duplicate. The program will register users and create a personal profile to upload information so they could participate in the making of money for themselves as contractors. |
| **Postconditions** | | |
| The vehicle is persisted in the system | | |
| **Business Rule** | | |
| No duplicate vehicle entry. | | |

|  |  |  |
| --- | --- | --- |
| **1.3** **Update a Vehicle** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The  user/admin requests for a list of vehicles | The system returns a list of vehicles. |
| 2 | The admin selects to view a vehicle they want to update. | The concept in works of the system displays an editable form with the vehicle details |
| 3 | The admin changes the parts they want to update and requests system to save the new details | The system updates the vehicle and returns a success message or a fail message on exception. |
| **Postconditions** | | |
| The vehicle details will be updated | | |
| **Business Rule** | | |
| If a vehicle is no more available for rent | | |
|  | | |
| **1.4  Delete** **Vehicle** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The  user/admin requests for a list of vehicles | The system returns a list of all vehicles. |
| 2 | The admin selects to delete a vehicle from the list. | The system displays an editable form with the vehicle details |
| 3 | The admin changes the parts they want to update and requests system to save the new details | The system updates the vehicle and returns a success message or a fail message on exception. |

|  |  |  |
| --- | --- | --- |
| **Use case Number: 3** | | |
| Name:    User verification | | |
| Brief description: This use case allows the company to verify it’s clients as well as it’s contractors | | |
| Actors: Admin, Client, Contractor | | |
|  | | |
| Preconditions: | | |
| The Admin, Client and Contractor must be logged into the system | | |
| Flow of Events: | | |
| 1. Basic Flows | | |
|  | | |
| 1.1 ID verification | | |
| 1.1.1 Upload Id Document | | |
| Step | User Actions | System Actions |
| 1 | Client/Contractor selects upload ID option | System redirects to upload page |
| 2 | Client/Contractor selects choose file option and picks from device or from URL option and then upload | System saves the Id to the server |
| Postconditions | | |
| System updates verification status to Id document uploaded | | |
|  | | |
| 1.1.2 Verify Id Document | | |
| Step | User Actions | System Actions |
| 1 | Admin checks uploaded Id document and clicks verified | System updates verification status to verified |
| Postconditions | | |
| System updates verification status to verified | | |
| Business Rule | | |
| Client/contractor cannot use system without being verified | | |
|  | | |
| 1.2 Email Verification | | |
| 1.2.1 Verify Email | | |
| Step | User Actions | System Actions |
| 1 | Client/Contractor selects verify email | System generates a random otp, saves it to database, sends an email with otp to email address and redirects to enter opt page |
| 2 | Client/Contractor enters opt on enter opt page | System checks if opt entered matches with opt saved in database, and update verification status to verified it true |
| Postconditions | | |
| System updates verification status to verified | | |
| Business Rule | | |
| Client/contractor cannot use system without being verified | | |
|  | | |
| Alternate flows | | |
| Client/Contractor Verification fails:  User profile remains deactivated, and user cannot access system services. | | |
|  | | |

# 

# 

# **Supplementary Specification**

Objectives The purpose of this section is to define non-functional requirements of the Asmeri Car Registration System. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The Supplementary Specifications and the use-case model together capture a complete set of requirements on the system.

Scope

This Supplementary Specification applies to the Asmeri Car Registration System System. This specification defines the non-functional requirements of the system; such as reliability, usability, performance, and supportability as well as functional requirements that are common across a number of use cases.

Reliability

The main system must be running 95% of the time.

Performance

The system shall support up to 100 simultaneous users against the central database at any given time.

Security

The system should not allow unauthenticated users to access booking services.

# 

# **Glossary**

## Introduction

This section is used to define terminology specific to the problem domain, explaining terms, which may be unfamiliar to the reader of the use-case descriptions or other project documents. Often, this document can be used as an informal data dictionary, capturing data definitions so that use-case descriptions and other project documents can focus on what the system must do with the information.

## Definitions

The glossary contains the working definitions for the key concepts in the Asmeri Car Registration System l System.

1. Front Desk Receptionist

A person who meets a customer who wants to rent or book a car in person.

1. Administrator

A manager of the front desk receptionist.

1. Client

A person who books a car or requests services from the car-rental company

1. Contractor

A person who registers to rent out their car through the car rental company

1. Vehicle

          Can be a bike, a car, a motorcycle, a truck, a bus or a lorry

1. Reservation

         A vehicle is booked by a client and a payment is made for some interval of time.

1. Overdue Fee

            Additional payment   for late return of a vehicle by the client.

1. Pick up date

          The date the vehicle is ready for the Customer to take it.

1. Pick up location

                The garage where the car is located.

1. Drop off date

               Due date for returning the Car to the Drop Off Location.

1. Drop of location

      The location/garage where the car is supposed to be returned to.

1. Underage

      Age restriction eligibility.

1. Payment Methods

       Accepted payment methods; cash, checks etc.

1. Car Model

       Car Brands.

1. Driving License

      A document granted for operating a vehicle.

# **Conclusion**

All in all, the project is the beginning of an online car rental systems that has the potential to compete with applications such as Turo in the real world. A user is able to create an account and sign with, with great security in place. Once a user is signed in, he has the ability to upload personal and vehicular information in order to establish himself/herself as a contractor and start earning money. This is the beginning of it.