

**<<car rental management system>>**

**Software Requirement Specification**

– Da Nang, May 2023 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\*  M, EASY | Print charge | Change Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview](#_Toc110459974)  [4](#_Toc110459974)

[1. Introduction](#_Toc110459975)  [4](#_Toc110459975)

[2. System Functions](#_Toc110459976)  [5](#_Toc110459976)

[3. Entity Relationship Diagram](#_Toc110459977)  [6](#_Toc110459977)

[II. Functional Requirements](#_Toc110459978)  [7](#_Toc110459978)

[1. <<Feature Name 1>>](#_Toc110459979)  [7](#_Toc110459979)

[a. <<Function Name 1>>](#_Toc110459980)  [7](#_Toc110459980)

[b. <<Function Name 2>>](#_Toc110459981)  [7](#_Toc110459981)

[2. <<Feature Name 2>>](#_Toc110459982)  [7](#_Toc110459982)

# I . Overview

## 1. Introduction

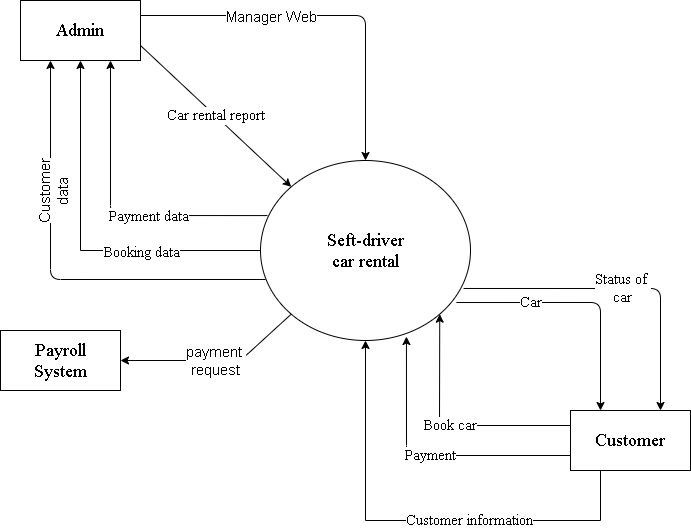
a, Background:

Building a self-driving car rental system based on people's needs and market trends, everyone needs a vehicle that can accommodate many people and sometimes a large family to go out and travel together. schedule in and out of the city, meeting the travel needs of the majority of people, and following the trend of people fighting switching to cars instead of motorbikes.

b, Business opportunities:

In order to solve everyone's need for car travel at the moment, this project parties promises to be very developed, trusted and trusted by everyone, and can compete directly with car rental. other self-driving. Because in addition to the need to travel, people also have requirements for comfort, freedom and above all convenience when using this service.

c, Business goals:

The goal of the project is to master the self-driving car rental industry, where customers trust, trust and receive absolute satisfaction. Through each stage of development, the system will collect comments from customers, who directly use the service to have the most intuitive view of the system, know whether to develop further and fix errors. in what places.

## 2. User requirements:

#### a. Actors:

|  |  |  |
| --- | --- | --- |
| **#** | **Actors** | **description** |
| first | Guest | Customers who have not registered an account, can use the search function, view information about vehicles, when they want to rent a car, they must log in and then be counted as a customer. |
| 2 | Customer | The end user who orders and rents cars from the system. Customers use the system to search, book, view information about vehicles, make deposits, pay and manage car rental transactions. |
| 3 | System Administrator | System administrators work on the administrative side of the system. They have access to management functions, including customer account management, employee information management. |
| 4 | Fleet manager | Managers are responsible for managing and maintaining vehicles in the system. They manage detailed information about vehicles, including location and status information. |
| 5 | Staff | Those who are responsible for assisting customers when they need them, current car rental orders and approve them. |

#### b. Use case diagrams:

A picture containing text, diagram, screenshot, line

Description automatically generated

#### c . Use case specification:

|  |  |
| --- | --- |
| **Name of use case** | **Log in** |
| Main actor | User |
| Level | first |
| Condition | User already has an account to access the system |
| Minimum guarantee | The system displays the login interface again |
| Guaranteed success | Users can access the system |
| Activated | User enters username and password and then clicks “login” button |
| Main sequence of events:  1. The system displays the login interface.  2. User enters username, password and clicks “login” button.  3. The system checks the validity of the username and password.  4. The system displays the corresponding functional interface to the User | |
| Exception:  The system checks that the user information is invalid  a. The system notices that the login information is invalid and asks to re-enter it.  b. User re-enters login information. | |

**1. Log in**

|  |  |
| --- | --- |
| **Name of use case** | **Log out** |
| Main actor | User |
| Level | first |
| Condition | User has logged into the system |
| Minimum guarantee | The system displays the previous interface again |
| Guaranteed success | The user exits his interface.  The system returns to the login interface |
| Activated | User chooses logout function |
| Main sequence of events:  1. The user selects the Logout function on the system's interface.  2. The system saves the user's session data.  3. Close the user interface.  4. The system returns to the login interface. | |
| Exception: | |

**2. Log out**

|  |  |
| --- | --- |
| **Name of use case** | **User management** |
| Main actor | Administrator (Administrator) |
| Level | 2 |
| Condition | The administrator has logged into the system |
| Minimum guarantee | The system displays the previous interface again |
| Guaranteed success | Update user information |
| Activated | Select the user management function |
| Main sequence of events:  1. The system administrator selects the user management function on the interface  system function area.  2. The system displays the user information management interface.  3. The system administrator enters the information the user wants to update and  press function button (add, delete, edit) employee.  4. The system checks the validity of user information and updates notifications  successful user. | |
| Exception:  The system checks that the user information is invalid  a. The system reported an error and asked to re-enter user information  b. System administrator re-enter user information | |

**3. User management**

|  |  |
| --- | --- |
| **Name of use case** | **Vehicle management** |
| Main actor | Fleet manager |
| Level | 2 |
| Condition | The administrator has logged into the system |
| Minimum guarantee | The system displays the previous interface again |
| Guaranteed success | Update vehicle information: add, edit, delete vehicle status |
| Activated | Select vehicle management function |
| Main sequence of events:  1. The management staff selects the vehicle management function to use on the system function interface.  2. The system displays the vehicle information management interface.  3. The management staff enter the vehicle information they want to update and press the function button (add, delete, edit) the vehicle.  4. The system checks the validity of vehicle information and announces successful vehicle update. | |
| Exception  The system checks that the vehicle information is invalid  a. The system reports an error and requires re-entering vehicle information  b. The manager re-enters vehicle information. | |

**4. Vehicle management**

## 3. Entity Relationship Diagram

A screenshot of a computer

Description automatically generated with medium confidence