Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking Garage application

Software Requirements Specifications (SRS)

**May,2022**

Contents

|  |  |  |
| --- | --- | --- |
| Sr.no | **Title** | **Page** |
| **1** | **Team** | **3** |
| **2** | **Document Purpose and Audience** | **3** |
| **3** | **Introduction** | **4** |
| **3.1** | **software purpose.** | **4** |
| **3.2** | **software scope.** | **5** |
| **3.3** | **Definitions, acronyms, and abbreviations .** | **5** |
| **4** | **Requirements** | **6-7** |
| **4.1** | **functional requirements .** | **6-7** |
| **4.2** | **nonfunctional requirements .** | **8** |
| **5** | **system models.** | **9-15** |
| **5.1** | **Use case model.** | **9** |
| **5.2** | **Use case tables** | **10-15** |
| **6** | **Owner report** | **16** |

# 

# 

# Document Purpose and Audience

**-What is this document?**

**-This document about The parking system which works with a combination of vehicle devices, software technology, and a mobile app for the best user experience.**

**-What will be included in this document?**

**1) software purpose, software scope, Definitions, acronyms, and abbreviations .**

**2)** **Requirements**

**2.1) functional requirements .**

**2.2) nonfunctional requirements .**

**3) system models.**

**3.1)use case model.**

**3.2)use case tables.**

**-who is excepted to read it?**

**1) people have parking garage**

**2) people have cars**

**Introduction**

**Parking management system for managing the records of the incoming and outgoing vehicles in an parking house**

**It’s an easy for Admin to retrieve the data if the vehicle has been visited through number he can get that data .**

**Now days in many public places such as malls, multiplex system, hospitals, offices, market areas there is a crucial problem of vehicle parking. The vehicle parking area has many lanes/slots for car parking. So to park a vehicle one has to look for all the lanes. Moreover this involves a lot of manual labour and investment. Instead of vehicle caught in towing the vehicle can park on safe and security with low cost.**

**Parking control system has been generated in such a way that it is filled with many secure devices such as, parking control gates, toll gates, time and attendance machine, car counting system etc. These features are hereby very necessary nowadays to secure your car and also to evaluate the fee structure for every vehicles entry and exit**

1. **software purpose.**

**The purpose of this project is to build a Vehicle Parking management system that enables the time management and control of vehicles using number plate recognition. The system that will track the entry and exit of cars, maintain a listing of cars within the parking lot, and determine if the parking lot is full or not. It will determine the cost of per vehicle according to their time consumption.**

1. **software scope.**

**Many people have vehicles. There are many corporate offices and shopping centers etc. There are many recreational places. So, all these places need a parking space where people can park their vehicles safely and easily. Every parking area needs a system that records the detail of vehicles to give the facility.**

**Vehicle parking system is an automatic system which delivers data processing in very high speed in systematic manner. Parking is a growing need of the time. Development of this system is very useful in this area of field. By using our system they can maintain records very easily. Our system covers the every area of parking management. In coming future there will be excessive need of Vehicle parking system.**

1. **Definitions, acronyms, and abbreviations**

**There are no abbreviations in this document.**

**Requirements**

1. **functional requirements.**

* Customer Register:

if the user is a new customer should insert the personal information and the vehicle information, and each customer can have more than one vehicle.

* Customer login:

when user login should enter the user name and password to can use park in operation or park out operation.

* Owner garage Register:

if the owner is a new owner garage should insert the personal information and the garage information, and each owner can have more than one garage.

* Owner garage login:

when owner login should enter the user name and password to can see available slots in garage and calculate total income.

* Vehicle:

the user should identify all information about the vehicle (e.g model name, model year, width, depth and the vehicle's unique number)

* garage:

the owner should identify all information about the garage should enter the number of slots and the dimensions of slots and garage’s address.

* park\_in:

the user should open the application when he park in, the system will check if found empty slots in garage, if founded slots the application will display it, if not founded the application will exist the user.

* Select configuration:

The owner garage will select the configuration during identify the garage to determine available slots from the next two configuration

1. first come first served slots. 2) The best slots are of a suitable size

* Display:

The application will display the available slots in garage.

* Select slot:

The user will select slot from free slots.

* Park\_out:

the user should open the application when he park out, the system will marks the departure time of a vehicle from the garage and calculate the parking fees during the park out based on the time of stay.

* calculate the parking fees:

the application will calculate the parking fees of each vehicle based on the time of stay with an hourly rate of 5 EGP. And the admin will calculate total income for each day.

* Payment: The user will pay cash.
* save all the details of user and vehicle:

the system will save all information about the user and the vehicle after the register, user don’t need to insert information again after register.

* Calculate total income:

The application will calculate the total income.

The garage owner can see total income in each time.

1. **nonfunctional requirements.**

* **Reliability:**

**The system should provide a reliable environment to both user and admin. All orders should be reaching at the admin without any errors.**

* **Usability:**

**The system is designed for user friendly environment and ease of use.**

* **Implementation Requirement:**

**Implementation of the system using, JAVASCRIPT, JQUERY, BOOTSTRAP, HTML, CSS in front end with java as back end and it will be used for database connectivity. And the database part is developed by SQL. Responsive web designing is used for making the website compatible for any type of screen.**

* **Security:**

**Other customers or unauthorized users should not have access to or be able to edit a customer’s account details or reservations**.

* **Fault-tolerance:**

**the application should quickly recover from a malfunction when a customer is inside the slot.**

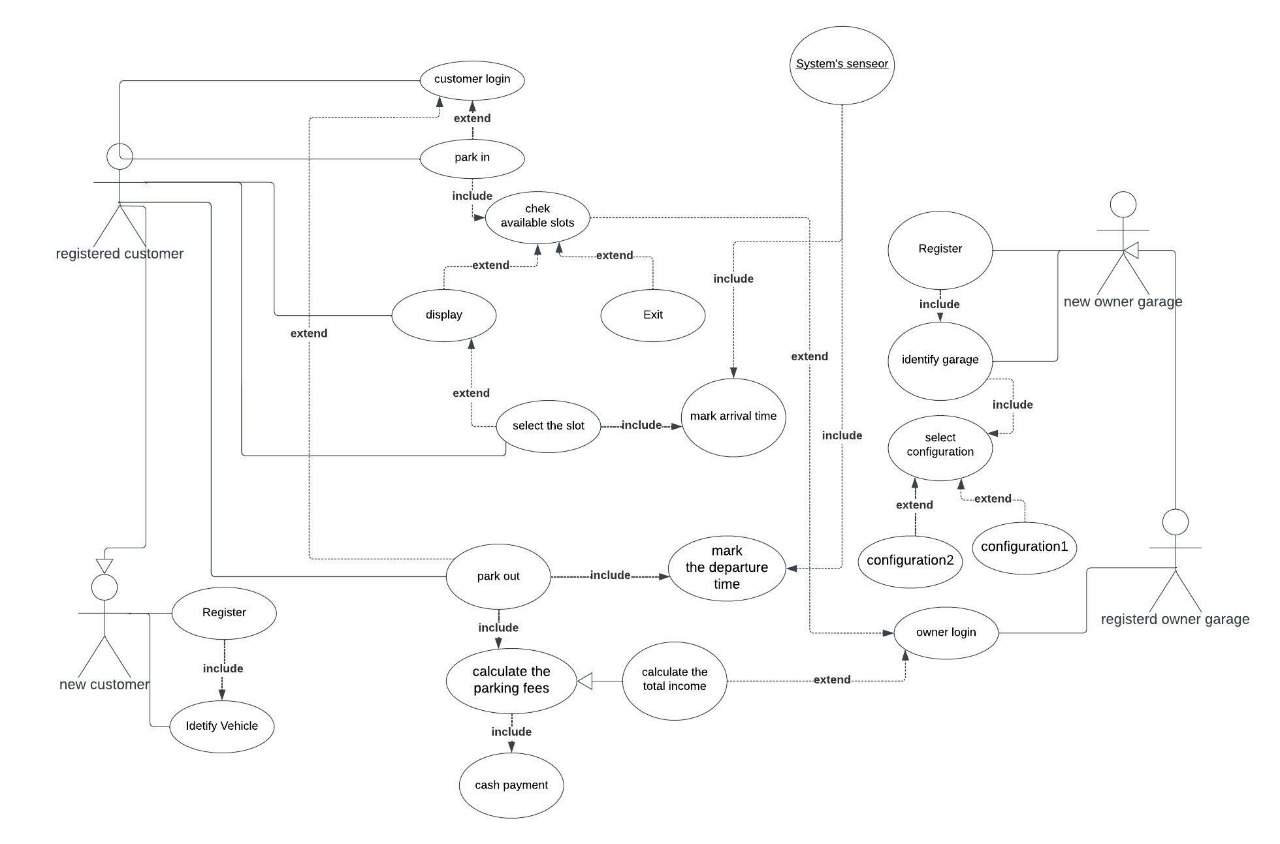
* **Performance:**

**1) The system should minimize connection times to the database and provide aquick and painless experience to the customer.**

**2) Initially, system can support at least 100 customers and 1,000 reservations. Over time should seek to increase these numbers ten-fold or more.**

**System model**

**Use case model:**



**Use case tables:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Customer register | |
| Actors: | New customer | |
| Pre-conditions: | The customer will open the application web page. | |
| Post-conditions: | Insert information about himself and his vehicle. | |
| Flow of events: | **User action** | **System Action** |
| 1-user will insert the first name and last name the phone number |  |
|  | 2-system will save information and send message “please insert all information about your vehicle.” |
| 3-user input vehicle model name, vehicle model year, width, depth and vehicle’s unique number. |  |
|  | 4-the system will save all information about user and vehicle and give unique id to vehicle and Send message “register is successful.”  Your user name:\*\*\*\*\*\*  Your password:\*\*\*\*\*\*\* |
| Exceptions: | **User action** | **System Action** |
| 1-user input vehicle model name, vehicle model year, width, depth and vehicle’s unique number. |  |
|  | The vehicle’s unique number is wrong, please check and try again. |
| Includes: | Identify vehicle | |
| Notes and Issues: | each customer can have more than one vehicle | |
| Use case id | 2 | |
| Use Case Name: | Garage owner register | |
| Actors: | New garage owner | |
| Pre-conditions: | The garage owner will open the application web page. | |
| Post-conditions: | Insert information about himself and his garage. | |
| Flow of events: | **User action** | **System Action** |
| 1-owner will insert the first name and last name the phone number |  |
|  | 2-system will save information and send message “please insert all information about your garage.” |
| 3-owner input garage address, garage name, number of garage slots, dimension slots. |  |
|  | 4-the system will save all information about owner and garage and Send message “please select the configuration to display the slots.” |
| 5-Select configuration. |  |
|  | 6- the system will save information and Send message “register is successful.”  Your user name:\*\*\*\*\*\*  Your password:\*\*\*\*\*\*\* |
| Exceptions: | **User action** | **System Action** |
| 1-owner input wrong address. |  |
|  | This address not found on application map. |
| Includes: | Identify garage | |
| Notes and Issues: | each owner can have more than one garage. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Select configuration | |
| Actors: | Garage owner | |
| Pre-conditions: | register | |
| Post-conditions: | login | |
| Flow of events: | **User Action** | **System Action** |
| 1-owner input garage address, garage name, number of garage slots, dimension slots. |  |
|  | 2- please select the configuration from the next to display the free slots  1) first come first served slots  2) The best slots are of a suitable size |
| 3- User Select configuration 1 |  |
|  | 4- System save information |
| Exceptions: | **User Action** | **System Action** |
| 1-owner input garage address, garage name, number of garage slots, dimension slots. |  |
|  | 2- please select the configuration from the next to display the free slots  1) first come first served slots  2) The best slots are of a suitable size |
| 3-user select configuration 2 |  |
| Includes: |  | |
| Notes and Issues: | The owner can change configuration after register in any time. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Park in | |
| Actors: | Registered customer | |
| Pre-conditions: | The customer must first log-in and if he is a new customer, he or she must register, then identify vehicle and then log-in as well. | |
| Post-conditions: | Select park in operation and after end operation logout. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter user name and Password. |  |
|  | 2- System Verify user data |
| 3- User Select garage address |  |
|  | 4- System check if found free slots and display it. |
| 5-user select slot and go to parking. |  |
|  | 6-system marks arrival time. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Select garage address. |  |
|  | 2- not found slots, system exist the user |
| Includes: | Display and select slot. | |
| Notes and Issues: | The system will display free slots independent on the selected configuration. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Park out | |
| Actors: | Registered customer | |
| Pre-conditions: | The registered customer must first log-in, and he must have done park-in first. | |
| Post-conditions: | After the registered customer park-out, the system mark the departure time, and the garage owner calculate the parking fees. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter user name and Password. |  |
|  | 2- System Verify user data |
| 3-user select park-out operation |  |
|  | 4- System mark departure time  5-calculate number hours of stay  6-calculate fees |
| 5-payment cash |  |
| Exceptions: | **User Action** | **System Action** |
| 1-The registered customer does login.  2-the registered customer select park-out, and he does not make park-in before. |  |
|  | 2- the system cannot mark the departure time. |
| Includes: | Calculate fees. | |
| Notes and Issues: | stay with an hourly rate of 5 EGP. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Calculate total income | |
| Actors: | Registered garage owner | |
| Pre-conditions: | The garage owner must first log-in and if he is a new garage owner, he or she must register, then identify garage and then log-in as well. | |
| Post-conditions: | After the garage owner does log-in, he select calculate the total income. | |
| Flow of events: | **User Action** | **System Action** |
| 1-owner Enter user name and Password. |  |
|  | 2- System Verify user data |
| 3-owner select park-out operation |  |
|  | 4- System display the numbers of vehicles which calculated fees. |
| 5-owner show total income |  |
| Exceptions: | **User Action** | **System Action** |
| 1-owner Enter user name and Password. |  |
|  | 2- System Verify user data |
| 3-owner select park-out operation |  |
|  | 4-There are no vehicles that have been parked |
| Includes: |  | |
| Notes and Issues: | Owner can see total income in any time | |

# 