Cairo University  
Faculty of Computers and Artificial Intelligent

Software Requirements Specifications (SRS)

Month & Year

May 2022

Contents

[Team 3](#_Toc101814800)

[Document Purpose and Audience 3](#_Toc101814801)

[Introduction 3](#_Toc101814802)

[Software Purpose 3](#_Toc101814803)

[Software Scope 3](#_Toc101814804)

[Definitions, acronyms, and abbreviations 3](#_Toc101814805)

[Requirements 4](#_Toc101814806)

[Functional Requirements 4](#_Toc101814807)

[Non Functional Requirements 4](#_Toc101814808)

[System Models 4](#_Toc101814809)

[Use Case Model 4](#_Toc101814810)

[Use Case Tables 5](#_Toc101814811)

[Ownership Report 6](#_Toc101814812)

[Policy Regarding Plagiarism: 6](#_Toc101814813)

# Document Purpose and Audience

* **This document represents a software system that is used as a garage parking system.**
* **Audience: Garage owner.**

# Introduction

## Software Purpose

* **This software describes requirement specification for a software used for the parking process in a garage.**

## Software Scope

* **The software has a fee calculation system that indicates when the vehicle enters and when it leaves then calculates the fee of that period.**
* **The software has two concepts for parking slots search.**
* **The software handles exceptions for bad inputs.**

# Requirements

## Functional Requirements

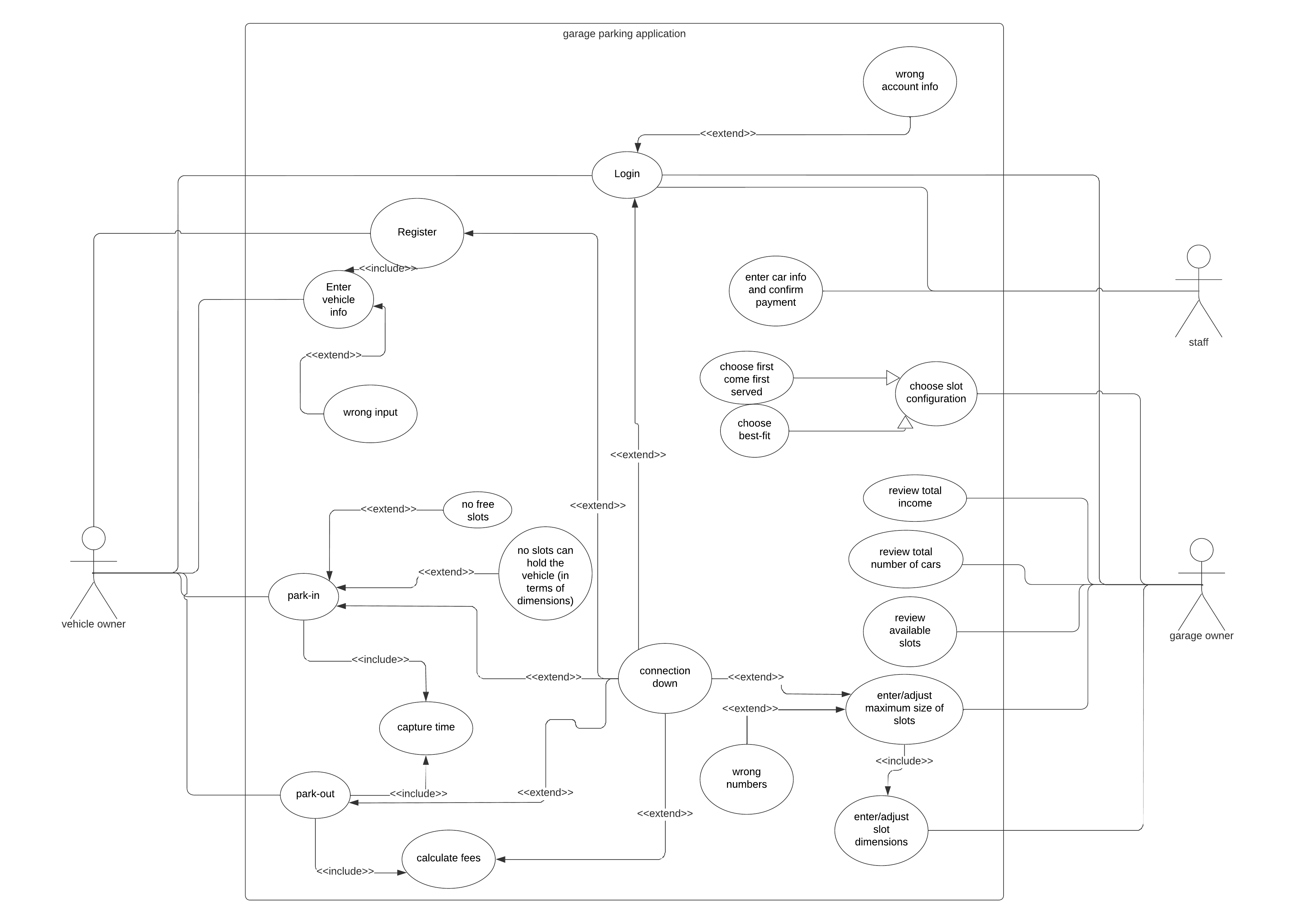
* **The vehicle owner uses the application to park his vehicle, this is done by the usage of the Park-in function which reserves a parking slot, and the Park-out function to leave and pay at the pay center.**
* **The garage owner should be able to use the application to display the total income of the garage, display the number of vehicles in the garage at any given point in time, review available slots and to adjust the maximum number of slots and the dimensions of each slot and choose the configuration used during searching a slot.**
* **The staff who is in the pay center should use the application to review the fees of any slot and the user who has to pay it and confirm payment.**

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **The system will be available 24 /7 without any dropout. (**Availability**)** * **The system will be able to find the right place for the vehicle within 20sec. (**Responsetime**)** |
| **Usability** | * **The system can reserve a slot for a vehicle in (at most) 3 steps.** |
| **Reliability** | * **The system can respond to the maximum number of vehicles the garage can hold at the same time. (**Robustness**)** * **The system should work even when the customer enters wrong input.** * **The data about the vehicles is safely stored and can’t be exposed. (Safety)** |

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Monitoring Garage | |
| Actors: | Garage owner | |
| Pre-conditions: | 1-the Garage owner logged in using an admin account. | |
| Post-conditions: | 1-The Garage owner has set the max size of the garage.  2- The Garage owner entered the dimensions of the slots.  3- The Garage owner reviewed the total number of cars in the garage.  4- The Garage owner reviewed the available slots in the garage.  5- The Garage owner reviewed the total income of the garage. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Garage owner clicks on adjust size. |  |
|  | 2- System asks him to enter the size. |
| 3- Garage owner enters the size and clicks on confirmation button. |  |
|  | 4-System assigns that number of slots and asks him to enter the dimensions of each slot. |
| 5- Garage owner enters the dimensions of the slots and clicks on confirmation button. |  |
|  |  | 6- the system creates the slots. |
|  | 7- Garage owner clicks on review total number of cars. |  |
|  |  | 8- System displays total number of cars. |
|  | 9- Garage owner clicks on review available slots. |  |
|  |  | 11-System displays all the available slots in the garage. |
|  | 12- Garage owner clicks on display Total income. |  |
|  |  | 13-System displays the total income of the garage. |
| Exceptions: | **User Action** | **System Action** |
| 1- Garage owner clicks on any button of the available options while the connection is down. |  |
|  | 2-The system displays an error message. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Park-in | |
| Actors: | Car owner | |
| Pre-conditions: | Already registered before. | |
| Post-conditions: | Capture the time when park in. | |
| Flow of events: | **User Action** | **System Action** |
| 1- user clicks on “login” button. |  |
|  | 2- System returns the login form. |
| 3-user enters name and password. |  |
|  | 4-the system checks name and password.  5- the check result found true then system display function page. |
| 5-the user clicks “book slot”. |  |
|  | 6-the system check car information.  7-the system get all data about car  8-system return “book slot” page. |
| 9-the user clicks “park in”. |  |
|  | 10-the system search for suitable and free slots in the garage according to the configuration.  11-the system then saves the car id and the car slot  12-the system then display the reserved slot for user. |
| 13-The user then park in garage. |  |
|  | 14-capture time when park in. |
|  |  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- the user submits login form. |  |
|  | 2- invalid account information.  3-or the connection is down. |
|  | 4-Enter vehicle information. |  |
|  |  | 5-invalid input from user about the information of the vehicle. |
|  | 6-park in |  |
|  |  | 7-no free slots  8-no slots can hold the vehicle (in terms of dimensions) |
| Includes: | 1-Capture departure time.  2-Calculate fees. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park-out | |
| Actors: | **vehicle** owner | |
| Pre-conditions: | 1-the user parked in using the application.  2-the user log in to the application. | |
| Post-conditions: | 1-the user park out of the garage system successfully.  2-The connection is down to the server. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose park-out function. |  |
|  | 2- System verifies that the user parked-in before then gets the slot number. |
|  | 3-System captures the time of departure. |
|  | 4-System calculates the time in hours and calculate the fees. |
|  | 5-System frees up the slot and increase the total number of vehicles used the garage and assigns fees to the customer. |
|  |  | 6-System displays that the operation is successful and the required fees. |
|  |  | 7-System sends fee amount, username, slot number, and estimated time to the staff account. |
|  | 8-User goes to the pay center. |  |
|  | 9-The staff checks the amount of fees assigned to the user by clicking display fees. |  |
|  |  | 10-System displays all fees to be paid with a button next to each fee. |
|  | 11-Staff confirms the payment assigned to the user. |  |
|  |  | 12-the system adds the fees to the total income and delete fee request. |
| Exceptions: | **User Action** | **System Action** |
| 1- User choose park-out function. |  |
|  | 2-The Application cannot connect to the system. |
|  |  | 3-The system displays an error message and asks the user to go to the staff to complete the operation manually. |
|  | OR |  |
|  |  | 2-The user didn't park in any slot. |
|  |  | 3-The System displays an error message. |
| Includes: | 1-Capture departure time.  2-Calculate fees. | |
| Notes and Issues: |  | |