Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

**(Parking Garage application)**

Software Requirements Specifications (SRS)

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| **20200428** | **Mohamed Ashraf Ahmed** | **muhamedashref2002@gmail.com** | **01155043953** |
| **20200156** | **Hassan Mohamed Nafed** | **hassan20nafed@gmail.com** | **01126287900** |
| **20200655** | **Yousef Sherif Mohamed** | **yousefsherif979@gmail.com** | **01122661587** |
| **20200540** | **Mostafa Shaban Abd El Rasoul** | **mostfashaban.soudy@gmail.com** | **01097142871** |

**May 2022**

**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)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| **20200428** | **Mohamed Ashraf Ahmed** | **muhamedashref2002@gmail.com** | **01155043953** |
| **20200156** | **Hassan Mohamed Nafed** | **hassan20nafed@gmail.com** | **01126287900** |
| **20200655** | **Yousef Sherif Mohamed** | **yousefsherif979@gmail.com** | **01122661587** |
| **20200540** | **Mostafa Shaban Abd El Rasoul** | **mostfashaban.soudy@gmail.com** | **01097142871** |

# Document Purpose and Audience

1. This document includes Software Requirements Specifications for the Parking Garage application.
2. This document includes Parking Garage application has a Software that helps the driver to Park his vehicle in suitable slot in the garage.
3. It is specially meant for developers that will work on the project in order to be organized.

* This is a Parking application system.
* The main target of audience we focus on are the drivers or vehicle owner and garage owner.

# Introduction

## Software Purpose

* This software is a system for parking application garages.

## Software Scope

* The software is a Parking application garages that provides a suitable way to the driver to park his car in the garage.
* The software has some actors for example driver, system as external providers and garage owner who must choose only way to park by.
* Garage must has a lot of slots with different dimensions.
* The software must support online services and Fawry's services for payment process.
* Software components: driver, vehicle, garage, slots, garage owner, booking a slot way, software system, way for payment.
* Software Features: book slot by pay for it cash or using a eWallet or any online services.

## Definitions, acronyms, and abbreviations

* Update price: This operation exist as the garage owner may change the cost per hour.
* The system must check the empty slots and check slot dimension that suit vehicle dimensions
* Driver may be the vehicle owner or not.
* Booking a slot may not be reserved as there is no empty slots
* Payment operation may be cash or online or by any other online services and online way may be include Fawry's services, credit card or with PayPal.
* Total income happened depend on Operation calculate time.

# Requirements

## Functional Requirements

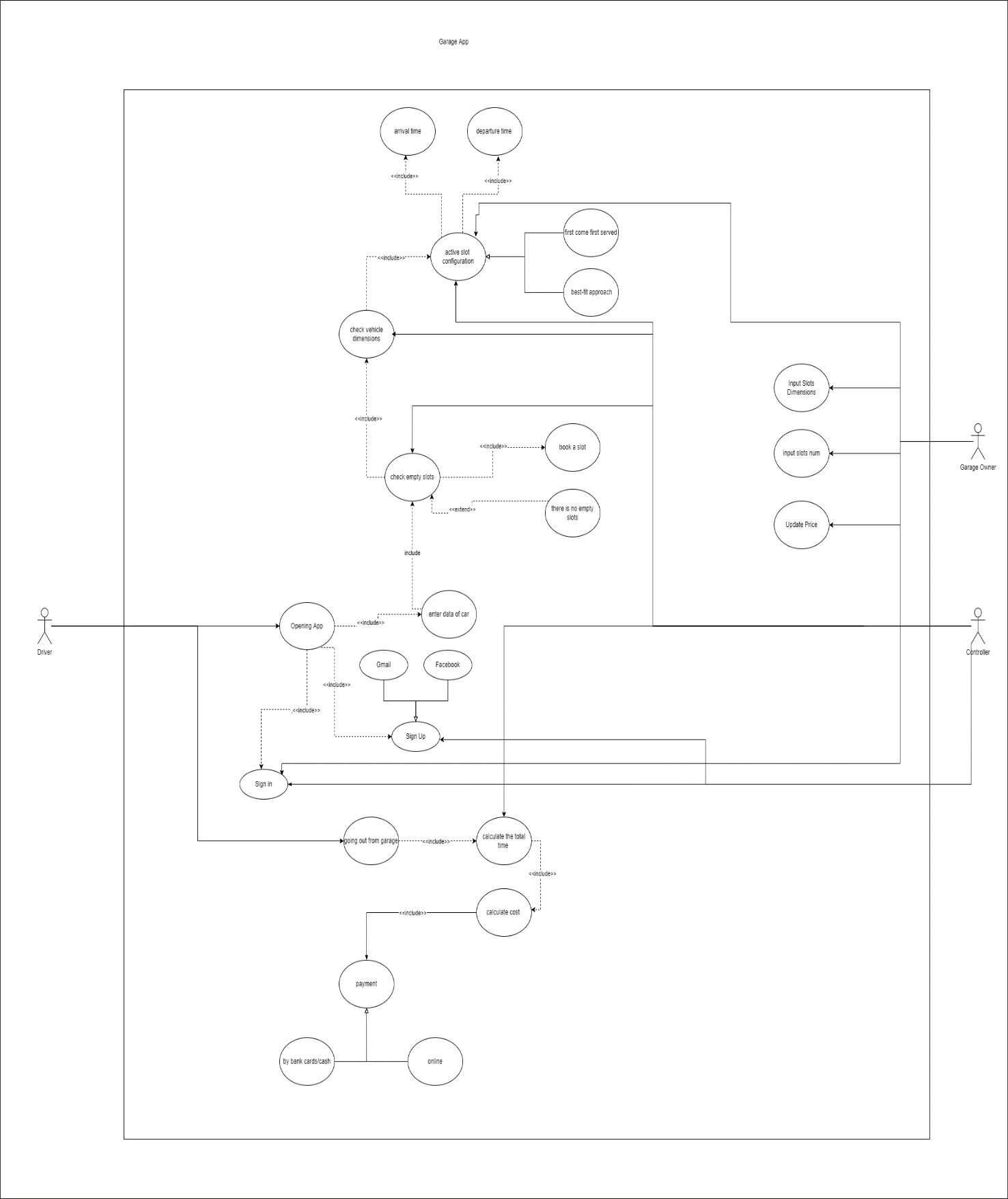
|  |  |  |
| --- | --- | --- |
| **Actor** | **Requirements** | **Descriptions** |
| Driver | Open app and  register | First step is opening app and register if it needed by driver and if he wants to book a slot. |
| Driver | Enter data | Entering the data of vehicle by entering the model’s name, unique identification number, Model year and vehicle dimensions. |
| Driver | Book a slot | In case the driver wants to bark his vehicle, he must book a slot. |
| Driver | Park in | Driver can park in by app that marks the arrival time of a vehicle if there is an available slot. Driver can book a slot. |
| Driver | Cancel booking | The system allows the customer to cancel his slots. |
| Driver | Customer payment | The system allows the customer payment by many ways. |
| Garage  Owner | Update price | To update slot price for regular hour if needed. |
| Garage  Owner | Choose one way to park | Garage owner has two ways to bark, and he must choose one of them. |
| Garage owner | Input slots number | The garage owner must set slots number. |
| Garage owner | Input slots dimensions | The garage owner must set each slot dimensions (width and depth). |
| System | Check empty slot | The system checks empty slots or no empty. |
| System | Check vehicle dimension | The system check vehicle dimensions. |
| System | Pick a slot | The system picks a free slot based on the active slot configuration. |
| System | Arrival time | The system marks the arrival time of a vehicle. |
| System | Total time | Based on the arrival time and Departure time system calculate the total time. |
| System | Parking cost | based on total time, system calculate the cost |
| System | Sending all info | The system sent invoice include all info and method to contact us in case emergency |

## Non Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Non-**  **functional type** | **Requirements** | **Descriptions** |
| Performance | App reacts | Each request should be processed within 10 seconds |
| Performance | App reacts | The app should load in 3 seconds |
| Security | Create account | System requires the user to create account to display his information |
| Security | Password generation | System does not allow the user to access until he enters the password |
| Reliability | Restore data | If the device if shot off during any process so the data be saved and not losing |
| Usability | Time of processing | You take 6 or less clicks to finish the process |
| Performance | Scalability | The system can take more than 10 customers to booking |
| Performance | Availability | User should see unresponsive parking slots as occupied. |
| Usability | Usability | User should see the parking occupancy data on the map in real time |
| Reliability | Protect information | User should not be allowed to see any data until logged in. |
| Performance | Availability | The system will work for almost 7 days in 24 hours. The target customers can gain the required customer service during any time in a day. |
| Reliability | Robustness | System should not allow the booking of the same parking slot by multiple users at the same time. |
| Reliability | Robustness | System should not allow the booking of parking slots which are occupied (not booked). |
| Performance | Response time | Parking overview should have data displayed in real time.  Booking should take place no more than 7 seconds. |

# System Models

## Use Case Model



## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Sign up | |
| Actors: | Driver and system | |
| Pre-conditions: | Driver opening app and enter info data to sign up | |
| Post-conditions: | Driver or user can sign up in app | |
| Flow of events: | **Driver Action** | **System Action** |
| 1- Driver opening app and system show category. |  |
|  | 2- System ask driver to sign in if the first sign in you must sign up. |
| 3- Driver choice to sign up. |  |
|  | 4- System ask driver to choice sign up by Facebook or Gmail or by form. |
|  | 5-driver choice to sign up by form. |  |
|  |  | 6- System check entered data.  7- System have checked data and save it in data base system. |
|  | 8-driver sign up to the app. |  |
| Exceptions: | **Driver Action** | **System Action** |
| 1- User Enter wrong data. |  |
|  | 2- system return invalid data.  3- System rejects booking. |
| Includes: | None. | |
| Notes and Issues: | None. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID: | 2 | | | |
| Use case name: | Check empty slots | | | |
| Actors: | Driver and system | | | |
| Pre-conditions: | Driver enter dimensions of cars and data about this cars. | | | |
| Post-conditions: | Driver find empty slots and book a slot to park in. | | | |
| Flow of events: |  | Driver Actions | Parking garage app Action |  |
| 1-driver want to book a slot to park a car |  |
|  | 2- System ask driver to enter data about car. |
| 3- Driver enter dimensions and data of car. |  |
|  | 4- System check empty slots to park in and tell the user that in garage empty slots. |
|  | 5-system ask user that in garage empty slots. |
| 6-driver book a slot. |  |
| Exceptions: |  | Driver Actions | Parking garage app Action |  |
| 1- Driver enter dimension and data of car. |  |
|  | 2-this dimensions not exist in the slots. |
|  | 3-no empty slots in garage |

|  |  |
| --- | --- |
| Includes: | Enter vehicle data, book a slot. |
| Notes and Issues: | None. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Arrival time | |
| Actors: | Driver and system | |
| Pre-conditions: | Driver must have a book slots. | |
| Post-conditions: | System mark arrival time and parking space status is updated. | |
| Flow of events: | **Driver Action** | **System Action** |
| 1-car approach gate. |  |
|  | 2- System detects the car tag and checks with may data for authorization |
|  | 3-the car is authorized and the system responds by opening the gate. |
| 4-the car drives through the gate. |  |
|  |  | 5-the system closes the gate once the car is through. |
| Exceptions: | Driver Action | System Action |
| 1- The car is authorized and the system responds by opening the gate. |  |
|  | 2-if the car access in not is authorized, an error message is displayed to inform the outcome that the access has been denied. |
| Includes: | None. | |
| Notes and Issues: | None. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Booking slot | |
| Actors: | Customer / system | |
| Pre-conditions: | Customer entering his data, choose his slot. | |
| Post-conditions: | Customer booked his slot. | |
| Flow of events: | **Driver Action** | **System Action** |
| 1- User opening the application. |  |
|  | 2- System show to him category items:  1- sign in  2- sign up  3- check slots |
| 3- User choose to sign in and enter his data |  |
|  | 4- system verify the data entered  5- system view empty slots |
|  | 6- customer enter data of his vehicle |  |
|  |  | 7- System verify the booking |
|  | 8- User select the payment method by cash |  |
|  |  | 9- System verify payment. |
| Exceptions: | **Driver Action** | **System Action** |
| 1- User Enter data to book a slot. |  |
|  | 2- Garage has no empty slots  3- System rejects booking. |
| Includes: | Sign up, sign in, and enter vehicle data and payment. | |
| Notes and Issues: | None. | |

|  |  |
| --- | --- |
| Use Case ID: | 5 |
| Use Case Name: | Input dimensions |
| Actors: | Garage owner and garage parking app |
| Pre-conditions: | The owner of garage entered the app to enter the app information about his garage |
| Post-conditions: | The owner of garage managed to enter the configuration his garage |
| Flow of events: | |  |  |  | | --- | --- | --- | | Garage owner Actions | |  | | --- | | System Action | | | 1-garage owner sign in app by username and password |  | |  | 2-the app accept data | | 3-garage owner input number slots and dimensions to each slot |  | |  | 4-the app saves the entries in its data base | | 5-garage owner input price to each slot |  | |  | 6-the app saves the price but it can update later | |  | 7-the app asked to garage owner about configuration method | | 8-garage owner choose first to come first served or best-fit approach |  | |  | 9-the app saves that in data base | |
| Exceptions: | |  |  | | --- | --- | | Garage owner Actions | Parking garage app Action | | 1- garage owner input number slots and dimensions to each slot |  | |  | 2-number slots and dimensions more space than the garage | |  | 3-the app asked to owner re-entry number slot again | |
| Includes: | Sign in, input slots number, input slots dimensions , first served and best-fit approach |
| Notes and Issues: |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case ID: | 6 | | | |
| Use case name: | Going out from garage | | | |
| Actors: | Driver | | | |
| Pre-conditions: | The driver finished his interest and wanted to get out of the garage | | | |
| Post-conditions: | The driver managed to get out of the garage successfully. | | | |
| Flow of events: |  | Driver Actions | Parking garage app Action |  |
| 1- The driver asked to leave the garage. |  |
|  | 2-the app calculate the total time to stay in the garage. |
|  | 3-the app calculate the total cost to parking process |
| 4-driver asked to the app to pay the cost |  |
|  | 5-the app asked to driver how to payment Online or cash |
| 6-driver choose payment by online |  |
|  | 7-the app asked to driver to enter card number |
| 8-driver enter card number |  |
|  | 9-the app checks the input and accepted payment |
| 10-the driver leave to garage |  |
| Exceptions: |  | Driver Actions | Parking garage app Action |  |
| 1-the card number is invalid | 2-the app return invalid card number please try again |
| 3-the card doesn’t contain enough money to payment |  |
|  | 4-the app refuse payment |

|  |  |
| --- | --- |
| Includes: | Calculate total time and payment. |
| Notes and Issues: |  |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| The team member has worked on the whole project | *Mohamed Ashraf Ahmed* |
| The team member has worked on the whole project | *Hassan Mohamed Nafed* |
| The team member has worked on the whole project | *Yousef Sherif Mohamed* |
| The team member has worked on the whole project | *Mostafa Shaban Abd El Rasoul* |