Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking garage

Software Requirements Specifications (SRS)

Team Names

May & 2022

Contents

[Instructions [To be removed] 3](#_Toc101814799)

[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)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20201183 | Mayar Mohamed Hamed | mayarmohamedhamed12345@gmail.com | 01024079836 |
| 20200595 | Nadeen Badr Eldin | Nadeenbadr0@gmail.com | 01122062048 |
| 20200095 | Amira Ahmed Ibrahim | am27335102@gmail.com | 01114989003 |
| 20201218 | Walaa Soudy Ibrahim | Walaasoudy36@gmail.com | 01150324662 |

# Document Purpose and Audience

**.** The following section provide an overview the software requirement specifications SRS for the parking garage application

**.** Any driver who wants to park his car or the workers responsible for organizing the operations that concern

# Introduction

## Software Purpose

The purpose of SRS is to determine both function and nonfunctional requirement of parking garage application, also the document provides as overall description about the parking garage application with UML analysis model

## Software Scope

Parking garage application helps people who want to park and register their car data in the system

Parking garage application helps workers to register vehicle data and locate the car's specification that need to be parking

## Definitions, acronyms, and abbreviations

Customer: is the person who is waiting for the system to show him the scene and whether there are places to park or not

Display: Display that shows empty spaces and displays how to pay

Accunting: Accounting is the one who calculates the parking income and parking fees

Sensor: is the one who can see the time of entry and exit of the car and the ID of the car

Owner: Owner sees the slot and does an update when it takes place,

# Requirements

## Functional Requirements

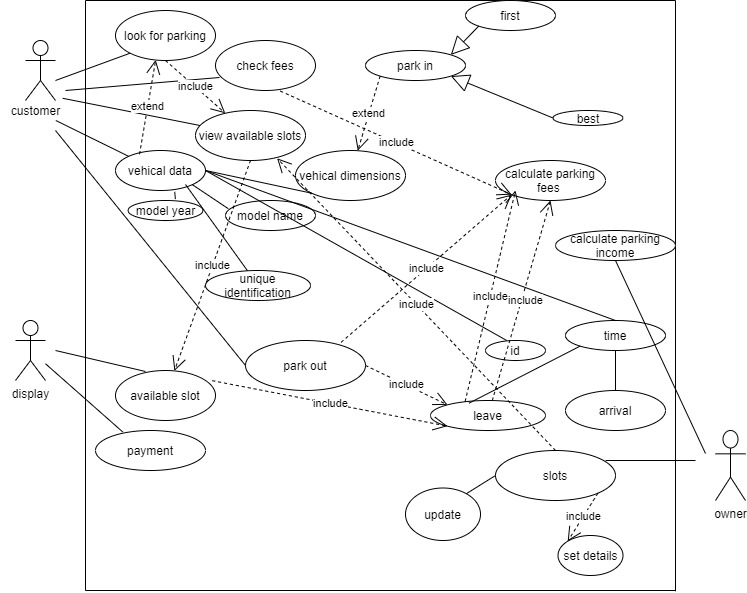
1. Customer can look for parking slot.
2. Customer can view available slots.
3. Customer must enter your cars model name, unique identification number, Model year and vehicle dimensions (vehicle width and depth) and choose from parking function he want first or best slot.
4. Customer can check fees after sensor capture arrive time and leave time and accounting system calculated it.
5. Accounting system calculated parking fees.
6. Accounting system calculated parking income.
7. Database manager update system data after any car come or leave.
8. Sensor take arrive and leave time to all cars come and leave and give them id.
9. Display system display available slot and charges to user.

## Non Functional Requirements

1. Number of steps to do any function must be less than 7 (usability).
2. The system must not be down more than 10 minutes per week (availability).
3. The system can support up to 1000 user at the same time (performance).
4. The respond time of the system must be 7 seconds at most (performance).
5. The server must be available 24 hours a day.

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | park in | |
| Actors: | Customer | |
| Pre-conditions: | Look for slot to park in. | |
| Post-conditions: | Already park his car. | |
| Flow of events: | **User Action** | **System Action** |
| 1-When customer choose park in. |  |
|  | 2- If system use best technique display to him available slots. |
| 3- Customer chooses one from those empty slots. |  |
|  | 4- System asks him to enter data to his car. |
| 5-Customer will enter car data. |  |
|  | 6-System will check width and depth with slot dimension and display result. |
| 7-Customers park his car. |  |
|  | 8- If system uses first technique ask him to enter data of car. |
|  | 9- Customer will enter car data and can park his car. |  |
|  |  | 10-System display to customer cars detail and id and slot. |
| Exceptions: | **User Action** | **System Action** |
| 1-Customer choice one of empty slot but not similar to his car dimension. |  |
|  | 2-System told him that is invalid. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Parking out and paying fees | |
| Actors: | Customer | |
| Pre-conditions: | The customer wants to view the fees and leave the garage | |
| Post-conditions: | Viewing the fees | |
| Flow of events: | **User Action** | **System Action** |
| 1- Customer wants to park out |  |
|  | 2- System ask him to choose his slot |
| 3- customer enter his slot number |  |
|  | 4- System retrieves the total hours from the slot saved data and send it to display system |
|  | 5- display system calculates the fees and view it to the customer |
|  |  | 6-empty the slot |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Income | |
| Actors: | Owner, calculateIncom , slot, website | |
| Pre-conditions: | The customer entered into parking garage application | |
| Post-conditions: | The customer already entered into parking garage application | |
| Flow of events: | **User Action** | **System Action** |
| 1 the owner is asking the website the slot data |  |
| 2-The owner calls from the slot the slot number and the width and the depth and the slots number |  |
| 3. The owner will ask the calculate income to calculate income |  |
|  | 4.calculateincome calculate total car and sends it to the slot |
|  |  |
|  |  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Isues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Details | |
| Actors: | Customer, website ,vechile | |
| Pre-conditions: | the customer enters the car details | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1. The customer entered into parking garage application |  |
|  | 2. asks the customer to enter the details of the car type, size and so on |
|  | 3. parking garage application ensures the customer that the data has been registered |
|  |  |
|  |  |
|  |  |  |
| Exceptions: | **User Action** | **System Action** |
| 1. parking garage application not ensures the customer that the data has been registered |  |
|  |  |
| Includes: |  | |
| Notes and Isues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | View slots | |
| Actors: | Customer and Parking Garage application | |
| Pre-conditions: | The customer wants to see the available slots | |
| Post-conditions: | The customer has already seen the available slots | |
| Flow of events: | User Action | System Action |
| 1- The customer entered into Parking Garage application. |  |
|  | 2- Parking Garage application asks him to enter the vehicle details. |
| 3- He entered the details |  |
|  | 4- display function show the available slots. |
|  |  |
| Exceptions: | User Action | System Action |
| 1- User Enter Card and Password. |  |
|  | 2- Card is invalid and unreadable.  3- System rejects cars. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Garage owner | |
| Actors: | Garage owner | |
| Pre-conditions: | Owner wants to set data and number of slots. | |
| Post-conditions: | Owner has entered data. | |
| Flow of events: | **User Action** | **System Action** |
| 1-Owner open parking garage system. |  |
|  | 2- System (website) asks him to enter number of slots in garage and store in database. |
| 3- Owner entered number of slots. |  |
|  | 4- System (website) asks him to enter width and height for all slots garage and store in database. |
| 5-Owner entered them. |  |
|  | 6- System (website) asks him to choice which technique he wants. |
| 7-Owner choice technique (First or best). |  |
|  | 8-Parkin function works by this technique. |
|  | 9-Owner goes to view available slots. |  |
|  |  | 10-Website display available slots. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Java code, SOLID | *Mayar, Nadeen* |
| Sequence diagrams , class diagram , use case diagram, requirements | *All team* |
| Drawing class diagram | *Amira* |
| Drawing use case diagram | *Walaa* |
| Purpose n scope , introduction, design specifications, Design pattern | *Amira , Walaa* |