**Cairo University  
Faculty of Computers and Artificial Intelligent** 

**CS251 - Software Engineering I**

Project Name

Software Requirements Specifications (SRS)

Team Names

Maher mohsen bahig labib

Mario guirguis abdel-messih

Kerolos george fayek nagib

Hazem adel khalel nabwy

May,2022

**Contents**

[Team 3](#_heading=h.gjdgxs)

[Document Purpose and Audience 3](#_heading=h.30j0zll)

[Introduction 4](#_heading=h.1fob9te)

[Software Purpose 4](#_heading=h.3znysh7)

[Software Scope 4](#_heading=h.2et92p0)

[Definitions, acronyms, and abbreviations 4](#_heading=h.tyjcwt)

[Requirements 5](#_heading=h.3dy6vkm)

[Functional Requirements 5](#_heading=h.1t3h5sf)

[Non Functional Requirements 5](#_heading=h.4d34og8)

[System Models 6](#_heading=h.2s8eyo1)

[Use Case Model 6](#_heading=h.17dp8vu)

[Use Case Tables 7](#_heading=h.3rdcrjn)

[Ownership Report](#_heading=h.26in1rg) 13

[Policy Regarding Plagiarism: 1](#_heading=h.lnxbz9)3

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200415 | Maher mohsen bahig labib | [-mezwmohsen@gmail.com](mailto:-mezwmohsen@gmail.com)  -11410120200415@stud.cu.edu.eg | -01287695176  -01001560655 |
| 20200405 | Mario guirguis abdel-messih | [-marioguirguis181@gmail.com](mailto:-marioguirguis181@gmail.com)  -11410120200405@stud.cu.edu.eg | -01550319008 |
| 20200399 | Kerolos george fayek nagib | [-kerolosgeorge290@gmail.com](mailto:-kerolosgeorge290@gmail.com)  -11410120200399@stud.cu.edu.eg | -01224587074 |
| 20200136 | Hazem adel khalel nabwy | [-hazemadelkhalel@gmail.com](mailto:-hazemadelkhalel@gmail.com)  -11410120200136@stud.cu.edu.eg | -01116104321 |

# Document Purpose and Audience

**Document Purpose:**

The purpose of this document is to list all the requirements that were chosen by the Customer and discussed with the team of software engineers. The functional requirements within the software are detailed with how it's achieved in the program with specific features and by discussing with the Customer also non-functional requirements are detailed to how to be achieved so in the next sections we discuss the features of every part of the system without any technical details.

**Document Audience:**

* Project Manager
* developer
* customer
* tester

# 

# Introduction

## Software Purpose

The Purpose of the Parking Garage Software Application is to manage the parking spaces within the garage.

## Software Scope

* Parking Garage Software displays for the user the available slots.
* Parking Garage Software helps the user to determine a fit slot for a vehicle to park in.
* Parking Garage Software calculates the parking fees during the park-out.

## Definitions, acronyms, and abbreviations

**Functional Requirements (FR):**

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs.

**Non-Functional Requirements:**

A non-functional requirement is a specification that describes the system’s operation capabilities and constraints that enhance its functionality. These may be speed, security, reliability, etc.

# Requirements

## Functional Requirements

1. Administrator can enter Parking garage number of slots and their depth and width.
2. Administrator can display available slots.
3. Administrator choose the park-in configuration option and he could change it   
    later.
4. Administrator enters the vehicle’s data (Depth, Width, Model Name, Model Year)
5. The system gives each parking user’s vehicle a unique ID.
6. The system mark in the arrival time for vehicle.
7. The parking user can park-in in slot based on the configuration was chosen by the administrator at the beginning of the program.
8. During parking-out for the parking user, System marks departure time.
9. Based on the difference between arrival and departure time, the system calculates the parking fees (Hourly 5 EGP)
10. System can calculate the total income, as well as the total number of vehicles that used the parking garage.

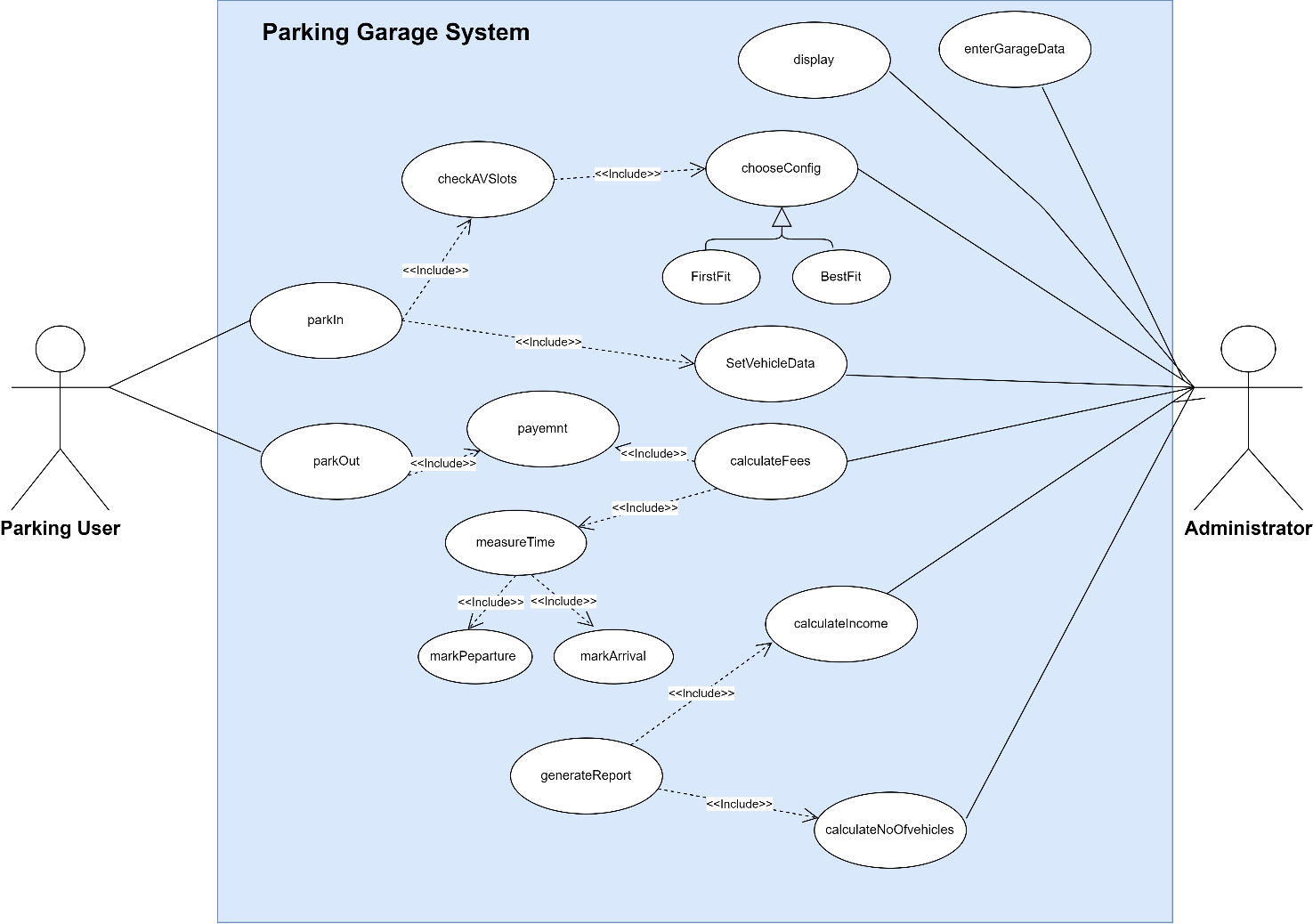
## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **Grantee any search for slot will take <= 1 Sec.** |
| **Usability** | * **Administrator will do finish Parking Operation atmost 5 minutes.** |
| **Compatibility** | * **System works on any hardware.** |

# 

# System Models

## Use Case Model

****

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Enter Garage data | |
| Actors: | Admin | |
| Pre-conditions: | Admin starts the system. | |
| Post-conditions: | 1. Garage data has been set. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Admin starts system. |  |
| 1. Admin clicks set slot button. |  |
|  | 1. System returns garage data form. |
| 1. Administrator enters garage data . |  |
|  | 1. System set garage data. |
|  |  |  |
| Exceptions: | **User Action** | **System Action** |
| No Exceptions |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Display Slots | |
| Actors: | Admin of the system. | |
| Pre-conditions: | When admin try to show all available slots in parking garage. | |
| Post-conditions: | There is a menu of available slots has been displayed. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Adminr clicks displayAVSlots button. |  |
|  | 2- The system responds by loading all available slots and then returns a list of all available slots. |
| Exceptions: | **User Action** | **System Action** |
| Exception 1: Administrator try to show available slots | 1- The system responds with a “**There is no available slots**” message if all slots aren’t empty. |
| Includes: | None | |
| Notes and Issues: | The administrator should receive a list of available slots within 10 seconds | |

=

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | parkIn | |
| Actors: | Parking User, Administrator | |
| Pre-conditions: | Parking user wants to park in. | |
| Post-conditions: | -Marks arrival time  -A new slot reserved | |
| Flow of events: | **User Action** | **System Action** |
| 1- Parking user choose parkIn |  |
| 2-Administrator use setVehicle data. |  |
|  | 3- System check Validate ID |
|  | 4- System check if there are any available slots based on the configuration the Administrator chose. |
|  | 5-System reserve the chosen slot. |
|  | 6- System mark arrival time |
|  | 7-Parking user park in successfully. |  |
| Exceptions: | **User Action** | **System Action** |
| Exception 1 : Parking user choose parkIn. | 1-System does not find any available slots.  2- System prints NO Available Slot message .  3-System rollback. |
| Exception 2 : Parking user choose parkIn. | 1-System finds that car already parked in garage.  2-System prints alert.  3-System rollback. |
|  |  |  |
| Includes: | checkAVSlots  setVehicleData | |
| Notes and Issues: | • Administrator will finish Parking Operation at most 5 minutes  • Grantee any search for slot will take <= 1 Sec. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | parkOut | |
| Actors: | Administrator, Parking User | |
| Pre-conditions: | The parking user that already parked want to park-out. | |
| Post-conditions: | Calculate fees and check the car out and free the slot. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Parking User choose to park out |  |
|  | 2- System validate that car already in garage. |
|  | 3-System marks departure time and calculate the difference between arrival time and departure time. |
| 4- Administrator select calculate fees |  |
|  | 5-System calculates fees. |
|  | 6-Parking user pays fees. |  |
|  |  | System checks out the car and frees the slot. |
| Exceptions: | **User Action** | **System Action** |
| Exception 1: Administrator enters Invalid ID. | 1-System prints alert.  2-System rollback. |
| Includes: | payment , CalculateFees | |
| Notes and Issues: | Administrator will finish Parking Operation at most 5 minutes  • Grantee any search for slot will take <= 1 Sec | |

# 

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | GenerateReport | |
| Actors: | Administrator | |
| Pre-conditions: | Administrator returns report feedback for the manager. | |
| Post-conditions: | Administrator has a report of total income and total number of vehicles. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Administrator click generate report button. |  |
|  | 2- System calculates total income. |
|  | 3-System returns total number of vehicles parked in the garage. |
|  | 4- System prints detailed report of the total income and the total number of vehicles. |
| 5-Administrator picks the report. |  |
| Exceptions: | **User Action** | **System Action** |
| No Exceptions |  |
| Includes: | calculateIncome, calculateNoOfvehicles | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| EnterGarageData Table | *all* |
| DisplaySlots Table | *Hazem adel* |
| ParkIn Table | *kerolos george* |
| ParkOut Table | *Maher mohsen* |
| GenrateReport Table | *Mario guirguis* |

# Policy Regarding Plagiarism:

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.