Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Project Name

Software Requirements Specifications (SRS)

Team Names

Month & Year

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** |
| 20200088 | Alaa Ayman Hashem | alaasalman211@gmail.com | 01003837227 |
| 20200552 | Moaaz Mohsen Attia | Moaazmohsen72@gmail.com | 01125741365 |
| 20201041 | Aya Hasanen said | Ayaayad162002@gmail.com | 01091487695 |
| 20200379 | Farah Mohamed Mohamed | farahemam70@gmail.com | 01030805323 |

# Document Purpose and Audience

This document is an explanation for an application that provide a new and easy way for users to park their vehicles in a garage by providing some functions that take the vehicle information(number , width and length) and find suitable place(slot)for the car to be parked in.

Presented to the project manager and customers.

# Introduction

## Software Scope

This software is to provide to the drivers an easy way to park their vehicle by providing some functions that store the vehicle data, find a suitable slot for it to park in, and make it easy to the driver to find his car to park out. In addition, provide some functions to the garage owner to be able to know number of busy and free slots at his garage and know the total income at any time

# Requirements

## Functional Requirements

1. Park-in: Application marks the arrival time of the vehicle if there is free slot (it captures time automatically).
2. Check slots: Application checks if there is available (free) slots or not.
3. Pick-up slots: Application has two options: a-first come first served slots, b-best fit approach where driver need to find a slot with the minimum dimensions.
4. Park-out: Application marks departure time of the vehicle from the garage.
5. Calculate parking fees: Application calculates the fees as 5 L.E per hour based on the time of stay.
6. Calculate total income: Application calculates the total income based on the total number of vehicles that used the parking garage.
7. Payment: Driver pays the fees of parking during parking out.
8. Display available slots: Application displays all the available slots to the driver.
9. Display total income.
10. display totalfees

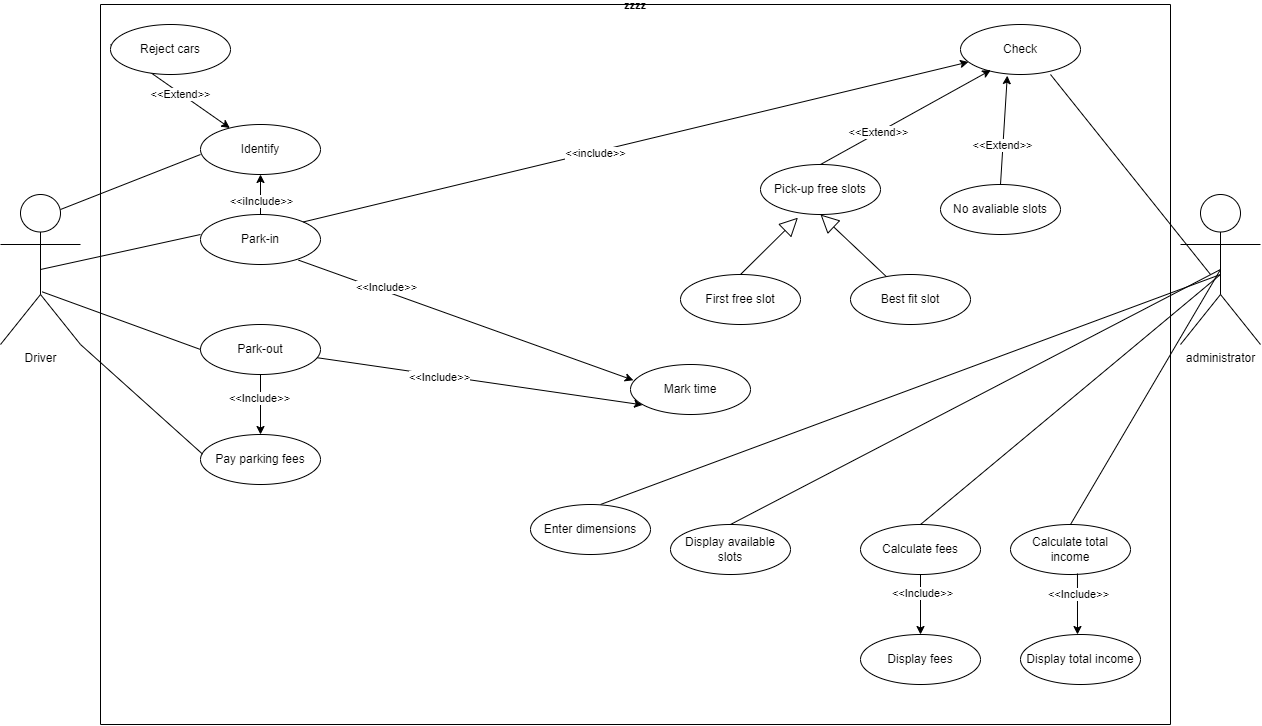
## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Robustness** | * system can cope with errors during execution , and cope with erroneous input |
| **usability** | * Easy to use and provide a condition for its users to perform the tasks safely, effectively, and efficiently |
| **maintainability** | * can be maintained easily |
| **Safety** | * protected against external harm event |

# 

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Park-in | |
| Actors: | Driver, administrator | |
| Pre-conditions: | Driver enters the garage | |
| Post-conditions: | The driver park in the garage | |
| Flow of events: | **User Action** | **System Action** |
| 1- the driver chooses to park-in |  |
|  | 2- system calls park-in function |
| 3-adminstrator enters the dimensions of the slots |  |
| 4- the driver choose the best configuration to park-in the car  (i)first come first served slots garage (ii) best-fit approach |  |
|  | 5- System checks for available slots |
| 6-the driver will enter the car information |  |
|  | 7-the system will choose slot for the car if there is available slots |
| 8-the driver will go and park the car in the chosen slot |  |
|  | 9-system will mark the time |
| Exceptions: | **User Action** | **System Action** |
|  | 1-there is no available slots |
| 2- the driver leaves the garage |  |
| 3- the driver forgot to enter any of the information about the car |  |
|  | 4- System reject the car |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID: | | 2 | | |
| Use Case Name: | | Park-out | | |
| Actors: | | administrator, Driver | | |
| Pre-conditions: | | The driver attends to leave | | |
| Post-conditions: | | The driver left the garage | | |
| Flow of events: | | **User Action** | **System Action** | |
| 1- Driver chooses to park-out |  | |
|  | 2- System calls park-out function | |
| 3-the administrator enters the slot number that the car park in |  | |
|  | 4-system show message that the slot is empty and decrease number of cars | |
|  | 5-system will mark the departure time | |
| 6-adminstrator chooses calculate fees |  | |
|  | 7-system will calculate the total time of staying and calculate the fees and display it to the administrator | |
| 8-the driver pay the fees then leave the garage |  | |
| Exceptions: | | **User Action** | **System Action** | |
|  | 1-System calculate fees in wrong way | |
| Notes and Issues: | | Driver must pay cash to the owner. | | |
| Use Case ID: | | 7 | | |
| Use Case Name: | | Calculate total income | | |
| Actors: | | administrator | | |
| Pre-conditions: | | Administrator chooses the option of calculating total income. | | |
| Post-conditions: | | Application calculates total income. | | |
| Flow of events: | **User Action** | | | **System Action** |
| 1-administrator chooses calculate total income | | |  |
|  | | | 2-system will call calculate total income function |
|  | | | 3-system calculate the total income and display it to the administrator |
|  | | |  |
| Exceptions: | **User Action** | | | **System Action** |
|  | | | 1-Application counted the vehicles in a wrong way. |
| Notes and Issues: | Application must make sure of saving all the number of vehicles that used the parking garage. | | | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Display available slots | |
| Actors: | administrator | |
| Pre-conditions: | administrator chooses the option of displaying all the available slots. | |
| Post-conditions: | Garage owner knows all the available slots in his garage. | |
| Flow of events: | **User Action** | **System Action** |
| 1-admistrator choose display available slots |  |
|  | 2-system will call the display function |
|  | 3- system will check the number of available slots |
|  | 4- if there is available slots system will display it to the user |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1-Application will not display any slots if there is no available ones. |
|  | 2-Application has delay and displays unavailable slots. |
| Notes and Issues: | Garage owner only has the permission to use this option. | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document purpose and audience , nonfunctional requirements and display available slots use case description | All team members |
| Use case diagram and Calculate-total income use case description | All team members |
| Park-in use case description | All team members |
| Park-out use case description | All team members |