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

**CS251 - Software Engineering I**

Parking Garage application

Software Requirements Specifications (SRS)

El-Sayes

Winter 2022

**Contents**

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

[Team 3](#_30j0zll)

[Document Purpose and Audience 3](#_1fob9te)

[Introduction 3](#_3znysh7)

[Software Purpose 3](#_2et92p0)

[Software Scope 3](#_tyjcwt)

[Definitions, acronyms, and abbreviations 3](#_3dy6vkm)

[Requirements 4](#_1t3h5sf)

[Functional Requirements 4](#_4d34og8)

[Non Functional Requirements 4](#_2s8eyo1)

[System Models 4](#_17dp8vu)

[Use Case Model 4](#_3rdcrjn)

[Use Case Tables 5](#_26in1rg)

[Ownership Report 6](#_lnxbz9)

[Policy Regarding Plagiarism: 6](#_35nkun2)

# 

# Team : El Sayes

| **ID** | **Name** |
| --- | --- |
| **20190457** | **Mohamad abd Elnasser shehata** |
| **20190561** | **Mohanad El-Areaf Bellah Mohammed** |

# Document Purpose and Audience

* **This document is a simple and well organized document which holds requirements specification for parking garage application.**
* **The target audience to read this document is { the customer who needs to have an application, senior software engineer, lead software engineer, the manager of a software company} .**

# Introduction

## Software Purpose

* **Develop a Parking Garage Application which manages vehicles.**

## Software Scope

* **Any software could have too many components / Major features .. but we should implement specific things...this is the scope**
* **In simple points, what is the software scope (focus on components / Major features, not tiny things)**
* **This application manages a parking space for a configurable maximum number of vehicles**

## Definitions, acronyms, and abbreviations

* **Slot : The space of the car to be parked in the garage.**

# 

# Requirements

## Functional Requirements

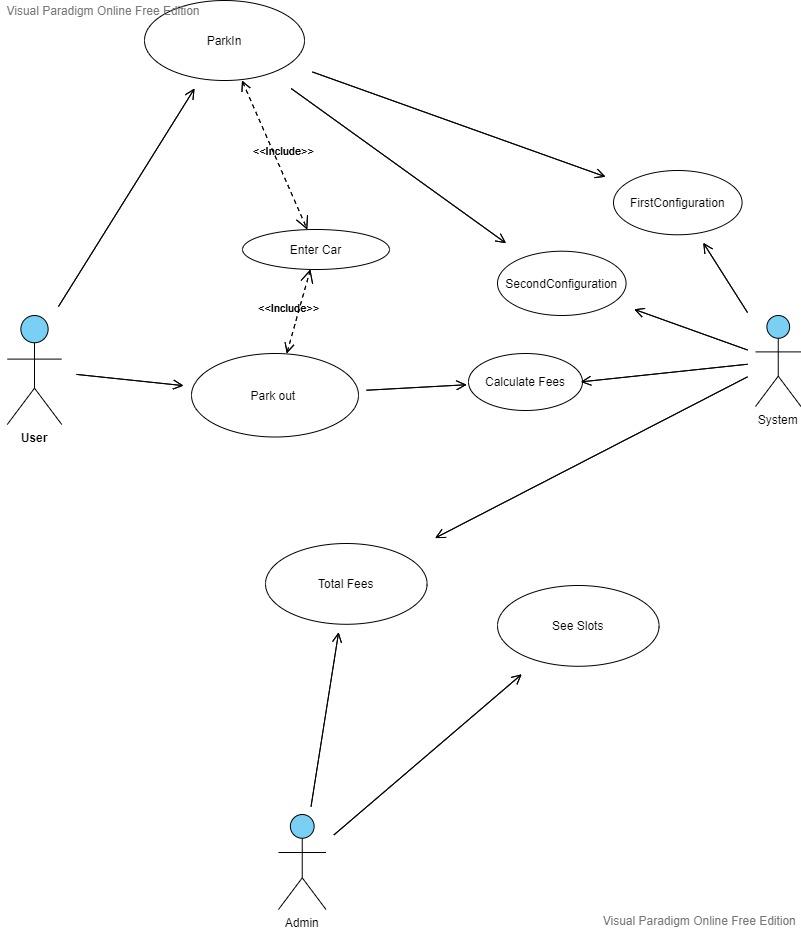
* **The system will be able to park users (normal person not owner) in the garage, by giving the information of the car such as {model name, unique identification number, Model year and vehicle dimensions (vehicle width and depth) }.**
* **The system uses 2-configurations to be able to park in the garage { first come first serve, best fit approach}.**
* **The user can park out and leave if he wants at his specific time, then display his total cost to him one hour of parking in the garage costs 5 USD (5 USD per hour).**
* **The system shows all available slots in the whole system.**
* **The system can get all total income as well as total number of vehicles at any specific time.**

## Non Functional Requirements

* **Time response must be less than 3 seconds.**
* **The app must be easy to use.**
* **While parking out and calculating the fees the user (person who has a car) must have a receipt.**
* **The system must be maintainable, flexible and stable.**

# System Models

## Use Case Model



## 

## Use Case Tables

* **Using below table template, for each requirement write a use case table that shows user/system interaction**
  + **If one requirement is so big, you could divide it to more than table**
  + **If some requirements are not major, you could plugin them in other senario**
    - **E.g. you may not do LogIn Usecase table as it is simple functionality**
* **Flow of events should be very detailed**

| Use Case ID: | # 1 | |
| --- | --- | --- |
| Use Case Name: | Park in | |
| Actors: | User - System | |
| Pre-conditions: | user want to park  User enter vehicle properties{ id, model name, model year, dimensions} | |
| Post-conditions: | User finished parking his vehicle successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User want to park in |  |
|  | 2- System want his data {his vehicle} |
| 3- User enter his data |  |
|  | 4- System receive his data show him an available id slot |
| 5- User press OK button |  |
|  |  | 6- System show a verification message |
| Exceptions: | **User Action** | **System Action** |
| 1- User press OK button without complete his data |  |
|  | 2- Data is invalid and incomplete.  3- System clear incomplete data. |
| Includes: | User’s vehicle properties - select slots based on configurations | |

| Use Case ID: | # 2 | |
| --- | --- | --- |
| Use Case Name: | Park out | |
| Actors: | User - System | |
| Pre-conditions: | The user already has an slot’s id, and his vehicle at this garage | |
| Post-conditions: | The user leaves garage with his vehicle and pay the bill (fees) | |
| Flow of events: | **User Action** | **System Action** |
| 1- The User enters his vehicle properties. |  |
|  | 2-  i.show vehicle’s slot  ii. show the user fees (the bill) to pay |
| 3- User pay the fees |  |
|  | 4- System receives fees and shows him a receipt. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters the wrong vehicle properties . |  |
|  | 2- The vehicle is invalid and unreadable. |
| Includes: | Read data of vehicle | |

| Use Case ID: | # 3 | |
| --- | --- | --- |
| Use Case Name: | Total fees | |
| Actors: | Admin - System | |
| Pre-conditions: | The admin enter the system and choose this property | |
| Post-conditions: | The system shows total fees to admin | |
| Flow of events: | **User Action** | **System Action** |
| 1- The admin enter the system |  |
|  | 2- The system welcomes him and shows him the total fees. |
| Exceptions: | **User Action** | **System Action** |
| 1- Admin close the window before get total fees |  |
|  | 2- The system show him confirmation message |
| Includes: | calculate fees | |

| Use Case ID: | # 4 | |
| --- | --- | --- |
| Use Case Name: | Display all available slots | |
| Actors: | Admin - System | |
| Pre-conditions: | The admin choose this property | |
| Post-conditions: | The system show all available slots to admin | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin close the window before get total fees |  |
|  | 2- The system welcomes him and shows him the total fees. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enters the wrong vehicle properties . |  |
|  | 2- The system show him confirmation message |

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%**
* **Team leader must verify the table with the team members.**

| **Item** | **Owners** |
| --- | --- |
| [Document Purpose and Audience](#_1fob9te), [Introduction](#_3znysh7), [Software Purpose](#_2et92p0), [Software Scope](#_tyjcwt), [Definitions, acronyms, and abbreviations](#_3dy6vkm), [Requirements](#_1t3h5sf) {[Functional Requirements](#_4d34og8)}, use case diagram, use case tables. | *Mohammed Abd El-Nasser shehata* |
| [Requirements](#_1t3h5sf) {[Functional Requirements](#_4d34og8)}, use case diagram, use case tables. | *Mohanad El-Areaf Bellah Talat* |