Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking Garage application

Software Requirements Specifications (SRS)

June 2022

Contents

[Team 3](#_Toc101814800)

[Document Purpose and Audience 3](#_Toc101814801)

[Introduction 3](#_Toc101814802)

[Software Purpose 3](#_Toc101814803)

[Software Scope 4](#_Toc101814804)

[Requirements 5](#_Toc101814806)

[Functional Requirements 5](#_Toc101814807)

[Non Functional Requirements 5](#_Toc101814808)

[System Models 6](#_Toc101814809)

[Use Case Model 6](#_Toc101814810)

[Use Case Tables 7](#_Toc101814811)

[Ownership Report 14](#_Toc101814812)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200323 | Abdallah Emad Abdelhamed | abdallahelal102@gmail.com | 01098486290 |
| 20200381 | Farida Said Abdellatif | faridasaid830@gmail.com | 01127871312 |
| 20200184 | Rawda Atef Muhammad | rawdaattef@gmail.com | 01113318556 |
| 20200544 | Mostafa Muhammad Ali | mostafa.fcai@gmail.com | 01142627589 |

# Document Purpose and Audience

This document gives a perception about how a Parking Garage application works, which helps to manage parking areas to achieve the best benefit from it as a customer and an admin.

As this document shows how the system works by viewing the expected functionalities and the flow of the expected process.

So the expected audience for this document is the project manager and the developers to take an overall view about the system and how it works.

# Introduction

## Software Purpose

A set of instructions, data, or programs used to control computers and perform certain activities is referred to as***software***. It's the polar opposite of hardware, which refers to a computer's physical components. Applications, scripts, and programs that operate on a device are collectively referred to as ***software***. It's the movable component of a computer, whereas the unmovable part is the hardware.

***Software*** can help your organization work more efficiently as well as make your computer hardware do crucial jobs. The appropriate software can potentially inspire new working methods. As a result, it is an important corporate asset, and you should carefully select software that meets your demands.

The software reduces the workload and automation of activities. It also assists in elimination of human errors, as a result, improves effectiveness and consistency.

There are various types of business software commonly used by companies. Software is responsible for accomplishing different tasks.

## Software Scope

This is software that helps organizations to organize their parking areas and making it an easy going system to park in a certain slot and guarantees safety and security. The system captures the arrival time and the departure time automatically and uses them to calculate the time that the vehicle stays then calculates the fees for each vehicle separately.

The system also enables the system admin to access the total income of the garage as well as the total number of vehicles that used the parking garage at any given point in time.

# Requirements

## Functional Requirements

1. **Vehicle owner must be able to define new car, specify its details (model, name, unique identification number أرقام اللوحات المعدنية , model year and dimensions).**
2. Admin must be able to update data of existing parking areas.
3. **System will automatically assign the car into a suitable slot using one of the two** configurations (i) first come first served approach. (ii) best-fit approach**.**
4. **System should be able to save the arrival &** departure **time** automatically **to calculate the fees.**
5. **Admin must be able to access** the total income and the total number of vehicles at any given point in time.
6. Admin must be able to access the information of all reserved parking areas and empty areas.
7. Displaying an error message when there is not a suitable slot.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Security** | **Vehicle owner can’t access information about any other car in the garage.** |
| **Reliability** | **System must be available 24/7 hours per week with failure rate of 2hours per week.** |
| **Performance** | **System response must be no more than 3 seconds.** |
| **Performance** | The system must be able to serve (2\*number of slots) at the same time. |

# System Models

## Use Case Model

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-1 | |
| Use Case Name: | Calculate fees | |
| Actors: | Vehicle owner | |
| Pre-conditions: | The vehicle owner wants to park out | |
| Post-conditions: | The system starts to calculate the duration time of stay | |
| Flow of events: | **User Action** | **System Action** |
| 1- The vehicle owner wants to park out from the garage |  |
|  | 2- The system calculates the time of stay using the arrival time and the departure time |
| 3- The vehicle owner knows the time of stay he spent |  |
|  | 4- The system calculates the fees by using the hourly rate of 5 EGP. |
| The vehicle owner pays the fees |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The vehicle owner pays a wrong amount of money |  |
|  | 2- The system displays an error message  3- The system enables the vehicle owner to re-pay |
| Includes: | Calculate the time of stay | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-2 | |
| Use Case Name: | Define new vehicle | |
| Actors: | Vehicle owner | |
| Post-conditions: | New vehicle has been saved in the database | |
| Flow of events: | **User Action** | **System Action** |
| 1- Vehicle owner chooses to define a new car |  |
|  | 2- System asks for a model name and model year |
| 3- Vehicle owner enters the model name and model year |  |
|  | 4-System verifies the model and model year and asks for the identification number |
| 5- Vehicle owner enters the identification number  6- Vehicle owner enters dimensions of the car |  |
|  | 7-System saves the information and shows a success message |
| Exceptions: | **User Action** | **System Action** |
| 1- Vehicle owner Enters a wrong model name or model year. |  |
|  | 2- Verifying model name or model year if valid or not inputs (invalid year or not a real model).  3- System rejects input.  4-System asks for input again |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-3 | |
| Use Case Name: | Park in | |
| Actors: | Vehicle owner | |
| Pre-conditions: | Vehicle owner is trying to park his vehicle. | |
| Post-conditions: | Vehicle owner has parked his vehicle. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver enters the garage trying to find a slot to park. |  |
|  | 2- System chooses a suitable slot and inform the user |
| 3- Driver parks the vehicle. |  |
|  | 4- System automatically capture the arrival time and saves it  5- system marks the slot as reserved slot and shows success message |
| Exceptions: | **User Action** | **System Action** |
| 1- Driver tries to find a slot. |  |
|  | 2- System cannot find a suitable slot (garage is full).  3- System shows an apologize message to the user. |
|  | 4- Driver doesn’t park the vehicle and leaves the garage. |  |
| Includes: | Define new vehicle | |
| Notes and Issues: | System works as first come first served | |

# 

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-4 | |
| Use Case Name: | Park out | |
| Actors: | Driver | |
| Pre-conditions: | Driver reserved a slot in the parking.  Driver wants to depart. | |
| Post-conditions: | Driver paid the fees. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver selects the parkOut choice and choose the slot he was parking in to leave the parking and checkout. |  |
|  | 2- System marks the departure time |
| 3- Driver Selects the payment method |  |
|  | 4- System calculates time of stay in the parking  5- System calculates the parking fees and displays it |
| 6- Driver pays the fees |  |
| Exceptions: | **User Action** | **System Action** |
| 1- There is a problem with the payment method that the driver chose |  |
|  | 2- System requests the user to choose a different payment method |
| Includes: | Park in , Calculate fees | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** | UC-5 | |
| **Use Case Name:** | Calculate Total income | |
| **Actors**: | System Admin | |
| **Pre-conditions:** | Calculate fees for every vehicle | |
| **Post-conditions:** | The total income changes in the database | |
| **Flow of events:** | **User Action** | **System Action** |
| 1- User selects the menu option “Park out”. |  |
|  | 2- System Verifies user’s data |
|  | 3- System calculates time of stay |
|  | 4- System calculates fees |
|  | 5- System shows the user the fees required |
|  | 6- User pays the fees |  |
|  |  | 7-System updates the total income in database including the new fees entered |
| Exceptions: | **User Action** | **System Action** |
| 1- User doesn’t pay the fees |  |
|  | 1-User’s vehicle isn’t already existed. |
| Includes: | Calculate fees | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-6 | |
| Use Case Name: | Define Garage | |
| Actors: | Garage owner | |
| Pre-conditions: | The garage owner wants to define a new system to his garage | |
| Post-conditions: | The structure of the system of the garage is done | |
| Flow of events: | **User Action** | **System Action** |
| 1- The garage owner chooses to define a new garage system |  |
|  | 2- The system displays the garage configurations to choose from |
| 3- The garage owner chooses the preferable garage configuration to him |  |
| 4-The garage owner enters the number of slots of the garage |  |
|  | 5-The system sets this number of slots to the garage |
| Exceptions: | **User Action** | **System Action** |
| 1- The garage owner enters a configuration that doesn’t exist |  |
|  | 2- The system displays an error message  3- The system enables the garage owner to re-enter his choice |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Source code | *Farida,Abdallah,Mostafa,Rawda* |
| Requirements | *Farida,Abdallah,Mostafa,Rawda* |
| Use Case Model | *Farida,Abdallah,Mostafa,Rawda* |
| Use Case Tables | *Farida,Abdallah,Mostafa,Rawda* |
| Class Diagram | *Farida,Abdallah,Mostafa,Rawda* |
| Sequence Diagrams | *Farida,Abdallah,Mostafa,Rawda* |