Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Garage Application

Software Requirements Specifications (SRS)

Coders

1 June 2022

Contents

[Team 3](#_Toc104996632)

[Document Purpose and Audience 3](#_Toc104996633)

[Introduction 3](#_Toc104996634)

[Software Purpose 3](#_Toc104996635)

[Software Scope 3](#_Toc104996636)

[Definitions, acronyms, and abbreviations 4](#_Toc104996637)

[Requirements 5](#_Toc104996638)

[Functional Requirements 5](#_Toc104996639)

[Non Functional Requirements 5](#_Toc104996640)

[System Models 6](#_Toc104996641)

[Use Case Model 6](#_Toc104996642)

[Use Case Tables 7](#_Toc104996643)

[Ownership Report 15](#_Toc104996644)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200583 | نبيل شريف نبيل ابراهيم درويش | 11410120200583@stud.cu.edu.eg | 01026276662 |
| 20200196 | زياد اسامه عبد الفضيل محمد | 11410120200196@stud.cu.edu.eg | 01018420940 |
| 20200153 | حسن خالد حسن طه | 11410120200153@stud.cu.edu.eg | 01156341785 |
| 20200091 | الحسن على فاروق | 11410120200091@stud.cu.edu.eg | 01110357406 |

# Document Purpose and Audience

**What is this document?**

This document explains:

* what this software will accomplish.
* How it will function.
* what features this software must have in order to meet the needs of all stakeholders.

# Introduction

## Software Purpose

This software document helps the user (The Garage owner) to easily understand how this software (Garage Application) works, and what is the different important functionalities that it can do

## Software Scope

This application was initiated as the garage owner wants an application to make the parking mechanism easier for him and his customers.

This application’s goal is to do the parking functionality for a garage, as it will handle the park in and park out functions and will calculate all the vehicles’ fees.

This application is now in its first stage as it is expected to do the previous mentioned requirements, but

* The payment method for this application is now cash only via the customer outside the system and still there isn’t a payment method via a credit card, that’s why there is no guarantee that the customer will pay his fees if the garage owner wasn’t there to take the money, or the customer left without him noticing.
* The application is designed to be used by the garage owner only and not by the customers.

This application is on limited budget so there is no form of an AI system to check the vehicles entered and left automatically as everything is done manually via the garage owner and the customers.

## Definitions, acronyms, and abbreviations

* **first come first served slots**: It is the first park-in function that will use the first free slot available from the parking garage slots
* **best-fit approach**: It is the second park-in function that will find the slot with the minimum dimension to hold the vehicle.

# Requirements

## Functional Requirements

* add any number of slots.
* apply one of two configurations (best fit approach and first come first served).
* display the available parking slots.
* park in.
* park out.
* calculate the fees for the vehicles that is parking out.
* calculate and display the total income at any point of time.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * The response time does not take more than 0.5 Second. * The garage owner can add new slots at any point of time. |
| **Scalability** | * The system supports only 1 user for now (The Garage Owner). |
| **Usability** | * The garage owner can use the system without registration. |
| **Robustness** | * The system can deal with wrong inputs. |

# System Models

## Use Case Model

Diagram, schematic

Description automatically generated

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Set the application configuration and calculate the total income with the total number of vehicles | |
| Actors: | The garage owner | |
| Pre-conditions: | The garage owner starts the application | |
| Post-conditions: | The application configuration has been set successfully and calculations has been done | |
| Flow of events: | **User Action** | **System Action** |
| 1- The garage owner selects the number of slots of the garage followed by the width and depth of each slot |  |
|  | 2- The system saves the number of the garage slots with its specification in the Garage Database |
| 3- The garage owner selects then the First Come First Served configuration |  |
|  | 4- The system now works with the First come First Served configuration |
| 5- The Garage owner selects calculate from the main menu |  |
|  |  | 6- The system gets from the Garage Database the total garage’s income as well as the total vehicles stored in the garage |
| Exceptions: | **User Action** | **System Action** |
| 1- The garage owner selects the number of slots of the garage followed by the width and depth of each slot |  |
|  | 2- The number entered cannot be saved (a negative number or a very big number) |
| Includes: |  | |
| Notes and Issues: | The garage owner enters the number of slots and the park in configuration only once in the program. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Park in using First come first served configuration | |
| Actors: | The garage owner | |
| Pre-conditions: | The customer comes into the garage  The garage owner starts the application | |
| Post-conditions: | The customer parks his vehicle successfully or  There are no free slots | |
| Flow of events: | **User Action** | **System Action** |
| 1- The garage owner chooses the Park In option from the main menu |  |
|  | 2- The system displays a form for the garage owner to enter his vehicle specification |
| 3- The garage owner enters the vehicle’s specifications (Model Name, Unique ID, Model Year, Vehicle Dimensions) |  |
|  | 4- The system checks for an available slot in the Garage Database and will assign the vehicle to the first free slot it finds. |
|  | 5- The System then will store the vehicle’s data and the arrival time of the vehicle in the Garage Database |
|  |  | 6- This slot will now be not empty in the Garage Database. |
|  | 7- The customer now can park his vehicle in that slot. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The garage owner enters the vehicle’s specifications (Model Name, Unique ID, Model Year, Vehicle Dimensions) |  |
|  | 2- An error might occur (The garage owner entered a wrong Unique ID or a very big vehicle dimensions) |
|  |  | 3- The system couldn’t find a free slot in the Garage Database |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park in using Best Fit Approach | |
| Actors: | The garage owner | |
| Pre-conditions: | 1. The customer comes into the garage 2. The garage owner starts the application | |
| Post-conditions: | The customer parks his vehicle successfully or  There are no free slots | |
| Flow of events: | **User Action** | **System Action** |
| 1- The garage owner chooses the Park In option from the main menu |  |
|  | 2- The system displays a form for the garage owner to enter his vehicle specification |
| 3- The garage owner enters the vehicle specifications (Model Name, Unique ID, Model Year, Vehicle Dimensions) |  |
|  | 4- The system checks for an available slot with the minimum dimensions in the Garage Database to hold the vehicle and the vehicle will be assigned to that slot |
|  | 5- The System then will store the vehicle’s data and the arrival time of the vehicle in the Garage Database |
|  |  | 6- This slot will now be not empty in the Garage Database. |
|  | 7- The customer now can park his vehicle in that slot. |  |
|  | 8- The customer chooses the display available slots from the main menu |  |
|  |  | 9- The system searches the database for the slots it has and displays them |
| Exceptions: | **User Action** | **System Action** |
| 1- The garage owner enters the vehicle specifications (Model Name, Unique ID, Model Year, Vehicle Dimensions) |  |
|  | 2- An error might occur (The garage owner entered a wrong Unique ID or a very big vehicle dimensions) |
|  |  | 3- The system couldn’t find a free slot in the Garage Database, or the system couldn’t find a free slot with the minimum dimensions entered |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Park out | |
| Actors: | The garage owner | |
| Pre-conditions: | 1. The customer comes into the garage to park out his vehicle 2. The garage owner starts the application | |
| Post-conditions: | The customer parks out his vehicle successfully and the vehicle’s fees has been paid to the garage owner | |
| Flow of events: | **User Action** | **System Action** |
| 1- The garage owner chooses the Park out option from the main menu and chooses the vehicle’s Unique ID he wants it to park out |  |
|  | 2- The system marks out the departure time of the vehicle from the garage |
|  | 3- The system displays the Parking Fees for that vehicle that should be paid to the garage owner and waits for the garage owner to press “OK” that he paid the money |
| 4- The garage owner gets the parking fees from the customer and the customer presses “OK” that he received the money |  |
|  | 5- the system deletes the vehicle specification from the Garage Database, and the slot that was specified for that vehicle is now empty again |
| Exceptions: | **User Action** | **System Action** |
| 1- The garage owner chooses the Park out option from the main menu and chooses the vehicle’s Unique ID he wants it to park out |  |
|  | 2- The system couldn’t find the vehicle’s Unique ID (maybe typed wrong), The system asks to enter the vehicle’s Unique ID again |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| **Document purpose and audience** | نبيل شريف نبيل  حسن خالد حسن طه |
| **Introduction** | نبيل شريف نبيل  حسن خالد حسن طه |
| **Functional and non-functional requirements** | الحسن على فاروق  زياد اسامه عبد الفضيل محمد |
| **Use case model** | نبيل شريف نبيل  حسن خالد حسن طه  الحسن على فاروق  زياد اسامه عبد الفضيل محمد |
| **Use case tables** | نبيل شريف نبيل  زياد اسامه عبد الفضيل محمد |