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** |
| 20200324 | Abdallah emadeldin abdallah | Abdallahelnabarawy2@gmail.com | 01012035979 |
| 20200598 | Nermin mostafa madbouly | nerminmostafa78@gmail.com | 01207253810 |
| 20200589 | Nada alaaeldin | nadaalaa2909@gmail.com | 01142071128 |

# Document Purpose and Audience

# • Any garage owner wants to set a system for his garage is expected to read this document.

# • This document describes garage application software.

# • Project Manager (Garage owner), Customer (vehicle owner).

# Introduction

## Software Purpose

* It is helping the garage owner to build a system to help him manage his garage by choose the configuration way to park in with cars and help him to calculate the fees for each vehicle also help him to know the total income from the garage**.**

## Software Scope

When the garage owner chooses the configuration way the system will use it to park the cars.

## Definitions, acronyms, and abbreviations

* There is a garage owner who will build the way for the system.
* The vehicle owner should use the system and enter information about his car to enter the garage.
* The system will handle the parking for the vehicle.
* The system will calculate the time when the vehicle enters the garage and when it leaves.
* The system will calculate the fees for each vehicle.
* The system will show the total income for the garage owner.

# Requirements

## Functional Requirements

1. Admin (garage owner) must be able to enter the configuration way, create new parking spaces, specify a range of parking slots, the cost of parking per hour and other information.
2. The vehicle owner must be able to use the system and provide vehicle information.
3. Admin must be able to update existing parking area data.
4. All registered parking spots' details must be viewed by the admin.
5. The backend administration system must be capable of accepting parking slot reservations depending on configuration chosen by admin (garage owner).
6. Recording the time of arrival of the vehicle to the garage.
7. The program must determine the slot which the car will enter through the configuration chosen by admin.
8. Recording the time, the vehicle is out of the garage.
9. Calculating the cost of parking based on the time the vehicle is in the garage and the hourly rate is 5 pounds.
10. Calculate the amount of income and the number of vehicles parked in the garage during any time.

## Non Functional Requirements

1. Data should be displayed in real-time in the parking overview.
2. User should be able to see real-time parking occupancy data.
3. User should see unresponsive parking slots as occupied.
4. Choose a place for the vehicle in less than 7 seconds.
5. The cost appears in less than 7 seconds.
6. The system should provide a reliable environment to admin. All orders should be reaching at the admin without any errors.
7. It is required that the system be able to enter any number of users without any error.
8. The system is designed for user friendly environment and ease of use.

# 

# System Models

## Use Case Model

Diagram

Description automatically generated

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Enter the system | |
| Actors: | Garage owner. | |
| Pre-conditions: | Login. | |
| Post-conditions: | Set garage info successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin enter email and password. |  |
|  | 2- System verify admin data. |
| 3- Admin select set garage info. |  |
|  | 4- System ask the admin to set configuration. |
| 5- Admin enter the way of configuration. |  |
|  |  | 6- System ask the admin to enter number of slots. |
|  | 7- Admin enter the number of slots and width and depth for each slot. |  |
|  |  | 8- System display message that garage info set successfully. |
| Exceptions: | **User Action** | **System Action** |
| 1- User enter wrong email or Password. |  |
|  | 2- Email or password is incorrect.  3- System ask the user to enter them again. |
| Includes: |  | |
| Notes and Issues: | Functional requirements point (1). | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Park in the garage | |
| Actors: | Vehicle owner. | |
| Pre-conditions: | Enter the garage. | |
| Post-conditions: | Vehicle owner park successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User open the application and choose to display available parking slots. |  |
|  | 2- System check available slots.  3- System display number and position of available parking slots to the user. |
| 4- User choose park-in. |  |
|  | 5- System ask the user to enter Vehicle info. |
| 6- User enter Vehicle info (model name, unique identification number, Model year and vehicle dimensions) |  |
|  |  | 7- System save car info.  8- System capture arrival time of the car. |
| Exceptions: | **User Action** | **System Action** |
| 1- User choose to display available parking slots. |  |
|  | 2- System check available slots.  3- System doesn’t found any available slots (the garage was full), so the system display to the user message that there are no available slots right now. |
| Includes: | Use case (1) | |
| Notes and Issues: | Functional requirements point (7). | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park out from the garage | |
| Actors: | Vehicle owner. | |
| Pre-conditions: | Vehicle owner wants to leave garage. | |
| Post-conditions: | Checkout. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User open the application and choose park-out and choose to view total parking fees. |  |
|  | 2- System capture time of leaving.  3- System calculate the parking fees during the park-out based on the time of stay with an hourly rate of 5 EGP and display total fees to the user. |
| 4- User choose checkout. |  |
|  | 5- System. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Use case (1) | |
| Notes and Issues: | Functional requirements point (9). | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Show income | |
| Actors: | Garage owner. | |
| Pre-conditions: | Garage owner wants to calculate total income. | |
| Post-conditions: | Close the application. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin open the application and choose to calculate the total income. |  |
|  | 2- System calculate total income and display it. |
| 4- Admin choose to display available slots. |  |
|  | 5- System display available slots. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Use case (3) | |
| Notes and Issues: | Functional requirements point (10). | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Functional requirements, non-Functional requirements + code | *Nada Alaa* |
| Use case, use case tables, sequence diagram + code | *Nermin Mostafa* |
| Class diagram, sequence diagram + code | *Abdallah emad* |

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

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.