Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking Garage Application

Software Requirements Specifications (SRS)

Team Names

Somaya Mohamed Youssef

Dina Ahmed Abdelrady

Mariam Alaa El-Deen Ahmed

Eman Ibrahim Shaban Gad

May 2022

Contents

[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 5](#_Toc101814809)

[Use Case Model 5](#_Toc101814810)

[Use Case Tables 6](#_Toc101814811)

[Ownership Report 15](#_Toc101814812)

[Policy Regarding Plagiarism: 15](#_Toc101814813)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200234 | سميه محمد يوسف محمد | sooma3mohammed@gmail.com | 01122515843 |
| 20201061 | دينا احمد عبدالراضي | da449393@gmail.com | 01102693753 |
| 20201038 | ايمان ابراهيم شعبان جاد | emanelkaser@gmail.com | 01012395893 |
| 20200525 | مريم علاء الدين احمد الامين | mariemalaa2002@gmail.com | 01155086103 |

# Document Purpose and Audience

* **Document Purpose:** the purpose of this document is to descript a Parking Garage application as it will represent features that will help users to use this application and get what they want in ease and secured way.
* **Audience:** Clients, Project Managers, Customers, Designer and Developer

# Introduction

## Software Purpose

* The purpose of this softwareis to make users able to park their vehicle in appropriate slot in the garage and make garage managers able to get all information about his garage at any time easily.

## Software Scope

* This software will be designed to help users who want to park their vehicle in the garage and garage managers who want keep track information about his garage and parking slots at any time, that will be done by providing appropriate interface that will be easy to use and organized.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | organized collection of structured information, or data that stored and monitored by this system. |
| Vehicle’s menu | Menu that you have to choose one of the following: park-in, park-out and payment. |

# Requirements

## Functional Requirements

**What does this system introduce for the users?**

* Customer can park in the garage at any time he wants if there is any available slot that appropriate with his vehicle’s dimensions.
* Customer can park out of the garage at any time he wants after finishing payment process.
* Customer can pay for the duration he stayed in the garage using his credit card.
* Owner can setup his garage information, add new slot and can get information about total income, total number of vehicle and available parking slots.

**What functionality should the system do during the process of parking?**

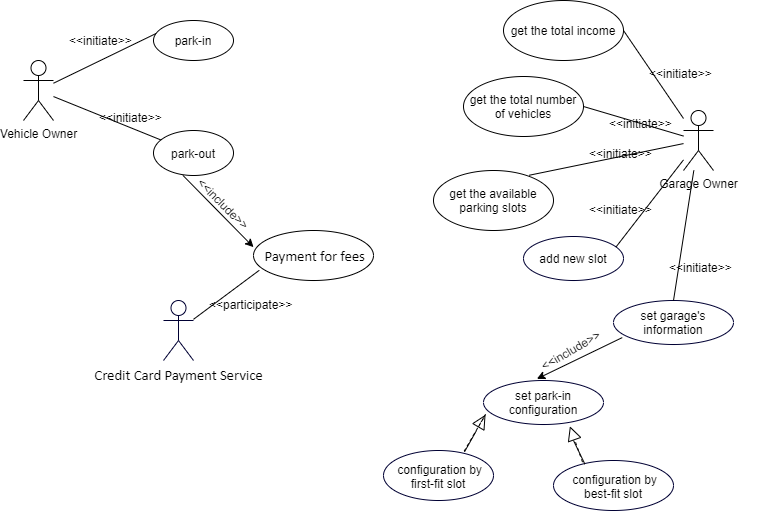
* the system identifies each vehicle by a model name, unique identification number, model year, and vehicle dimensions (vehicle width and depth).
* During park in the system should catch arrival time of viable and show if there are available slots (catch time should be automatic).
* During park in System catches a free slot for a vehicle and the system should choose a suitable slot (based on the dimensional of the vehicle and slot) for the vehicle, not a random place.
* During park out system should capture departure time automatically.
* System should calculate costs of parking during park out and calculate it based on the duration the vehicle stays with the rate of 5 pounds per hour.
* System also should calculate the total income of this project and calculate the number of vehicles at any point in time.
* System also should support to display of free slots available in the garage.
* System should handle exceptions that may happen such as no available free slots in our garage.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| Usability | * The system should be easy to use such a user reaches his requirements after 4 steps. |
| Reliability (security) | * The system should be secured against any attack or hack, no one can access to data except the administrator, data of customers should be stored in safely way. |
| Reliability (robustness) | * A system able to adapt to some changes in the environment such as degrees up to 90c, the system should handle any wrong inputs. |
| Performance (response time) | * Response time for any process should be within 5 seconds. |
| Performance (availability) | * The system is down not more than 7 minutes in the week. |
| Performance (scalability) | * The system should accept 100 users while the process of barking in the garage. |
| Supportability (maintainability) | * The system should allow any modification at any time. |
| Data integrity | * The system will maintain the consistency and accuracy of the collected and stored data. Different authentic validation and checking methods will ensure the degree of integrity and validity of the data and information. |

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-1 | |
| Use Case Name: | add new slot | |
| Actors: | Garage Owner | |
| Pre-conditions: | Garage owner select that he wants to add new slot to his system | |
| Post-conditions: | Garage owner added new slot successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- Garage owner enter slot’s id, width and depth. |  |
|  | 2- System check if slot’s id number is not existing.  3- System add new slot information to the database of garage. |
| Exceptions: | **User Action** | **System Action** |
| 1- Garage owner enter slot’s id, width and depth. |  |
|  | 2- System find that garage owner added a pre-existing id.  3- System rejects addition.  4- System display message explaining the reason for rejection to the garage owner. |
| Includes: | None. | |
| Notes and Issues: | None. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-2 | |
| Use Case Name: | set garage’s information | |
| Actors: | Garage Owner | |
| Pre-conditions: | Garage owner open the application for first time and enter total number of slots in his garage. | |
| Post-conditions: | Garage owner set information for his garage successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Garage owner enter slot’s id, depth and width. |  |
|  | 2- System check id, width and depth validation.  3- System add new slot to the database of garage.  4- System ask the garage owner if he want to add new slot or stop.  5- System ask garage owner to set park-in configuration (best-fit or first fit) |
| 6- Garage owner select he want to follow first-fit configuration. |  |
|  | 7- System save the configuration setting. |
| Exceptions: | **User Action** | **System Action** |
| 1- Garage owner enter slot’s id, depth and width. |  |
|  | 2- System find that garage owner added a pre-existing id.  3- System rejects addition.  4- System display message explaining the reason for rejection to the garage owner. |
| Includes: | Set park-in configuration. | |
| Notes and Issues: | None. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-3 | |
| Use Case Name: | park-in | |
| Actors: | Vehicle Owner | |
| Pre-conditions: | Vehicle owner selects park-in process from vehicle’s menu (park-in, park-out). | |
| Post-conditions: | Vehicle owner parks his vehicle in the right slot | |
| Flow of events: | **User Action** | **System Action** |
| 1- Vehicle owner enter vehicle’s width, depth, model name, unique identification number and model year. |  |
|  | 2- System check if the information is valid.  3- System check if there any available slot fits with vehicle’s dimension.  4- System stored vehicle’s information and entry time that the vehicle owner entered to garage into the database of the garage.  5- System make slot number not available.  6- System update total number of vehicles that entered to the garage.  7- System create ticket with slot number that the vehicle can park-in then display it to the vehicle owner. |
| Exceptions: | **User Action** | **System Action** |
| 1- Vehicle owner enter vehicle’s width, depth, model name, unique identification number and model year. |  |
|  | 2- System find that there is not available slot at this time  3- System asks the vehicle owner to wait until there is an available slot. |
| 1- Vehicle owner enter vehicle’s width, depth, model name, unique identification number and model year. |  |
|  | 2- System find that there is Invalid Information.  3- System asks the vehicle owner to enter vehicle’s information again. |
| Includes: | None | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-4 | |
| Use Case Name: | park-out | |
| Actors: | Vehicle Owner | |
| Pre-conditions: | The vehicle owner selects park-out process from vehicle’s menu (park-in, park-out). | |
| Post-conditions: | The park-out process has been completed successfully and the vehicle got out of the garage. | |
| Flow of events: | **User Action** | **System Action** |
| 1- The vehicle owner enters the number of slot and the unique identification number of his vehicle. |  |
|  | 2- The system verifies the validity of the two numbers.  3- system set exit time for the vehicle.  4-system calculate the fees of the parking.  5- system print an invoice for the required fees.  6- system ask the vehicle owner to complete the payment process in order to get out of the garage.  7- system ask the vehicle owner to enter the number of his credit card. |
| 8- vehicle owner enter the number of his credit card. |  |
|  | 9- system display t to the vehicle owner a massage which tell him that the payment process completed successfully. |
| Exceptions: | **User Action** | **System Action** |
| 1- The vehicle owner enters the number of slot and the unique identification number of his vehicle. |  |
|  | 2- The system verifies the validity of the two numbers.  3- The system find that the number is invalid and there is no vehicle with this number.  4- the system rejects the process and announce the vehicle owner. |
| Includes: | Payment for fees | |
| Notes and Issues: | none | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document Purpose and Audience & Introduction | Somaya |
| Requirements | Eman |
| Use Case Model | All |
| Use Case Tables | All |

# 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.