Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

C251-Project Description-ParkingGarage-v1.0

Software Requirements Specifications (SRS)

Team Names

Month & Year

Contents

[Team 3](#_Toc101814800)

[Document Purpose and Audience 3](#_Toc101814801)

[Introduction 4](#_Toc101814802)

[Software Purpose 4](#_Toc101814803)

[Software Scope 4](#_Toc101814804)

[Definitions, acronyms, and abbreviations 4](#_Toc101814805)

[Requirements 4](#_Toc101814806)

[Functional Requirements 4](#_Toc101814807)

[Non Functional Requirements 5](#_Toc101814808)

[System Models 5](#_Toc101814809)

[Use Case Model 5](#_Toc101814810)

[Use Case Tables 6](#_Toc101814811)

[Ownership Report 12](#_Toc101814812)

[Policy Regarding Plagiarism: 12](#_Toc101814813)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200575 | Mirette Shenouda Maher | [miretteshenouda01@gmail.com](mailto:miretteshenouda01@gmail.com) | 01223898053 |
| 20200567 | Monica Saeed Habib | monicasaeed12@gmail.com | 01226489661 |
| 20200686 | Youstina Saadawy Thabet | [youstinasaadawy@gmail.com](mailto:youstinasaadawy@gmail.com) | 01207227565 |
| 20200386 | Catherine Ramy Mikhail | [catherineramy02@gmail.com](mailto:catherineramy02@gmail.com) | 01201700544 |

# 

# Document Purpose and Audience

This document is used to summarize client (TA) requirements.

In this document, we will show functional requirements, non-functional requirements, use case diagram and use case tables.

1. Functional requirements are the features that the developers must implement to enable the user to accomplish his tasks by fulfilling the client’s needs i.e., things he wants to specifically make in the system.
2. Non- functional requirements are the specifications that describe the system’s operations, capabilities, and constraints and how it works such as speed, usability, reliability …etc
3. A use case is a system analysis methodology for identifying, clarifying, and organizing system needs. (Used during requirements elicitation and analysis to represent external behavior)

Audiences: client (TA), and the engineer who will design class diagrams and sequence diagram

# 

# Introduction

## Software Purpose

This software program wants to set the vehicle in the appropriate slot in the garage and calculate the garage income at any time automatically.

## Software Scope

What the system will do?

1. check available slots.
2. choose the correct slot.
3. calculate payment.
4. calculate total income.
5. calculate total vehicles.

What the system will not do?

1. if the system stopped for any reason, it cannot calculate total income at this time or display for the user the total fees he needs to pay.
2. If the garage is already full or there are no empty slots suitable for the vehicle, the system cannot solve it, the vehicle will leave.

## Definitions, acronyms, and abbreviations

Configurations to pick free slot based on active slot configuration

* best-fit approach where you need to find the slot with the minimum dimension to hold the vehicle.
* first come first served slots i.e., the park-in function will use the first free slot available from the parking garage slots

use case diagram

setup\_garage: the garage owner sets the slot depth, width and id, also enters how many slots he wants in the garage and choose the configuration that we pick free slot based on it.

Park in: check if there is an available slot, displays its id, marks arrival time if the vehicle and also the configurations

Park out: marks departure time, and displays the fees to the user

# Requirements

## Functional Requirements

- The system allows you to enter data about your vehicle such as model year and name, car identification number and vehicle dimensions.

- System displays the available slots.

- System assigns to the customer a suitable slot to park in by two methods best-fit approach and first come first.

- System calculates the total income and total number of vehicles at any given point in time - System calculates how long the vehicle stayed in the garage and how much the customer will pay.

## 

## Non-Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| Usability | * it takes the customer 3 steps to park, first he enters the garage. second, he enters the required data about the vehicle, third he parks in the slot assigned to him by the system. * The system must support only one vehicle driver. |
| Scalability | System should be able to support up to 500 vehicles. |
| Robustness | if the user enters vehicle width or depth with a negative value, the system will refuse it. |
| Availability | The system is down not more than 1 hour per week. |
| Usability | The system must support only one vehicle driver. |

# System Models

## Use Case Model

* **Using UML, write the use case model expressing the system actors & operations**

Diagram

Description automatically generated

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Garage setup | |
| Actors: | Garage owner | |
| Pre-conditions: | The garage owner opens her system | |
| Post-conditions: | The garage owner will set all designs for her | |
| Flow of events: | **User Action** | **System Action** |
| 1- garage owner opens the system. |  |
|  | 2- the system asks the user to set slots with specific dimensions for all slots and choose the way to display the cars |
| 3-user will choose one of the 2 configurations and set the slots dimensions and ids |  |
|  | 4-system will save all information  And create garage having array of slots with the info. |
| Exceptions: | **User Action** | **System Action** |
| 1- the user entered wrong dimensions of any slot. |  |
|  | 2- there is an option in the system to set new dimensions. |
| Includes: |  | |
| Notes and Issues: | user cannot change the way of configurations after using the system | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Park in | |
| Actors: | Vehicle driver | |
| Pre-conditions: | The vehicle with complete info wants to park in the garage. | |
| Post-conditions: | Getting the id of the slot to park in. | |
| Flow of events: | **User Action** | **System Action** |
| 1- the vehicle driver comes to the garage to park his car, the vehicle driver set vehicle information (model name, Model year and vehicle dimensions (vehicle width and depth) ). |  |
|  | 2- check for the suitable slot for the vehicle according to the configuration of the garage.. the system automatically displays to the driver the slot ID that best suits his vehicle. |
| 3-The driver will go to the slot and park there. |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1-system display that there aren't any free slots for the vehicle. |
| 2-the customer will leave the garage. |  |
| Includes: | Pick slot | |
| Notes and Issues: | The vehicle driver may park in a different slot, and this slot was not chosen for him by the system. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park out | |
| Actors: | Vehicle driver | |
| Pre-conditions: | The vehicle driver having his car id wants to park out. | |
| Post-conditions: | Displaying the parking fees which will be paid by the driver. | |
| Flow of events: | **User Action** | **System Action** |
| 1- the vehicle driver decided to leave the garage. |  |
|  | 2- the system display message to take car id from the driver. |
| 3-the driver will enter his car id to park out. |  |
|  | 3- the system automatically verifies vehicle information and checks how long it spent in the garage and displays to the customer how much he is going to pay. |
| 4- the vehicle driver pays the money for garage owner |  |
| Exceptions: | **User Action** | **System Action** |
| 1-The driver enters id which is not in the garage right now. | 2- display error message telling him that this id not found in the garage. |
|  |  |
| Includes: | Calculate fees | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Calculate total number of vehicles | |
| Actors: | Garage owner | |
| Pre-conditions: | Garage owner (which already have a setup garage) want to check how many vehicles are in there at any given point in time. | |
| Post-conditions: | Garage owner will know number of vehicles at this time.  By displaying it on the screen by the system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- garage owner ask the system to show him how many vehicles come in my garage. |  |
|  | 2- the system will go get the number of total vehicles used the system which is already stored in it. |
|  | 3- The system displays the number of vehicles from the beginning of the system until now. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Park in. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Calculate total income | |
| Actors: | Garage owner | |
| Pre-conditions: | Garage owner (which already have a setup garage) want to check total income at any given point in time. | |
| Post-conditions: | Garage owner will know total income in this time.  By displaying it on the screen by the system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- garage owner ask the system to view for him total income. |  |
|  | 2- the system will go get the number of total income used the system which is already stored in it. |
|  | 3- The system displays the total income from the beginning of the system until now. |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | Calculate fees. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Display available slots. | |
| Actors: | Garage owner. | |
| Pre-conditions: | Garage owner (which already have a setup garage) want to view available parking slots. | |
| Post-conditions: | garage owner will know the available parking slots.  By displaying it on the screen by the system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- garage owner ask the system to view for him available parking slots. |  |
|  | 2- the system will go check every slot status in his array of slots and return slots which only have available status. |
|  |  | 1. The system will display this new array of available slots to the garage owner. |
| Exceptions: | **User Action** | **System Action** |
| 1. - garage owner ask the system to view for him available parking slots. |  |
|  | 1. If there are no available slots the system will display an error message |
| Includes: |  | |
| Notes and Issues: |  | |

# 

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| - Document purpose and audience  -use case table 3 | *Mirette Shenouda Maher* |
| -requirements  -use case table 2 | *Catherine Ramy Mikhail* |
| - introduction  -use case model (diagram) | *Monica Saeed Habib* |
| -use case table 1,4,5,6 | Youstina Saadawy Thabet |

# 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.