A picture containing text

Description automatically generatedCairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering I**

Parking Garage application

Software Requirements Specifications (SRS)

*Represented by:*

|  |  |  |
| --- | --- | --- |
| *Name* | *ID* | *Section Number* |
| Mariam Ashraf Amin Mohamed | 20200520 | S15 |
| Hoda Shafek Ahmed said | 20200619 | S15 |
| Reem Ayman Abdel-Fattah Ibrahim | 20200186 | S15 |
| Mariam Saeid Shawky | 20200521 | S15 |

*June,2022*

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 3](#_Toc101814805)

[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 17](#_Toc101814812)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200520 | Mariam Ashraf Amin Mohamed | mariamashrafamin@gmail.com | 01144456820 |
| 20200619 | Hoda Shafek Ahmed said | hodashafek28@gmail.com | 01142974462 |
| 20200186 | Reem Ayman Abdel-Fattah Ibrahim | reem.ayman52002@gmail.com | 01098834544 |
| 20200521 | Mariam Saeid Shawky | mariamsaeid142@gmail.com | 01110192619 |

# Document Purpose and Audience

This is a garage system that tells us how we can park vehicles, as it is explained the technique that is followed. System organizes and facilitates parking, which helps the garage’s owner and the vehicle’s owner. The first user should enter the vehicle’s data then the system will find a slot to park in.

* **Audience who is expected to use this system are:**

Clients: client who planning to organize parking in his/her garage to save time and increase profit.

Software developers: developer benefits from this document, it helps them in code implementation as it organizes ideas, specifies requirements, and expects output.

# Introduction

## Software Purpose

## Software purpose is to enable garage owner to benefit from his/her garage correctly, this will help him/her in organizing and saving time, through entering his/her garage’s data and choosing the way that he/she wants vehicle’s user park in garage based on availability of slots. Enabling garage owner to know all total income, also enable vehicle’s user to park in and get out from garage by himself.

## Software Scope

## Software helps vehicle’s user to park in garage and get out from it easily.

## Software helps garage owner to manage his garage.

## Software should have database to save vehicles’ data.

## Software has GUI to make it easy for any user to use it.

* Software should support mathematical operations and record current time.
* Software should support more than one configuration for picking slot.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Word** | **Definition** |
| Enter garage con | It is process of entering one configuration from two (best fit or first fit). |
| Park-out | It is process of entering id and check if it is invalid or no. |
| Add vehicle | It is process of adding vehicle’s data to data base. |
| Pick slot | It is process that return suitable slot to user to that vehicle’s user park in either if garage owner chooses best fit or first fit. |

# Requirements

## Functional Requirements

* Vehicle’s user can park his car in garage by entering his vehicle’s data then the system will show suitable slot to park in.
* Vehicle’s user can get his car out of the garage by entering his vehicle id and paying the cost of parking using cash payment.
* Garage owner enters garage data and choose between two configurations of picking slot (best fit or first fit).
* Garage stuff can update garage data.
* Garage stuff can know the total number of vehicles in garage, total income, available slots in garage.

## Non Functional Requirements

|  |  |
| --- | --- |
| Security | All data in the system should be secure from stealing. |
| Response Time | All operations in the system should be completed in less than 5 seconds. |
| Scalability | 200 users can use the system at the same time without breaking down. |
| Maintainability | The system should be easy to be modified and be readable. |
| Usability | 5 steps are the maximum number of steps to do a specific function. |
| Reliability | System should take 5 min to up if it is down. |

# System Models

## Use Case Model

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC1 | |
| Use Case Name: | Park-in | |
| Actors: | Vehicle’s user. | |
| Pre-conditions: | When user arrive to park his/her vehicle. | |
| Post-conditions: | When the system displays available slot. | |
| Flow of events: | **User Action** | **System Action** |
| 1-user choose park in garage. |  |
|  | 2- System will check if there are available slots in garage. |
|  | 1. If there are available slots the system will ask to enter vehicle’s data. |
| 1. User enters vehicle’s data like model name, unique identification number, Model year and vehicle dimensions. |  |
|  | 1. System will pick slot. |
|  | 1. If there is suitable slot the system will display it and add vehicle’s data to database. |
| Exceptions: | **User Action** | **System Action** |
| 1- user choose park in garage. |  |
|  | 2- System will check if there are available slots in garage. |
|  | 3-if there aren’t available slots in the system will display “no available slots”. |
| **User Action** | **System Action** |
| 1- user choose park in garage. |  |
|  | 2- System will check if there are available slots in garage. |
|  | 3- If there are available slots the system will ask to enter vehicle’s data. |
| 4- User enters vehicle’s data like model name, unique identification number, Model year and vehicle dimensions. |  |
|  | 5- System will pick slot. |
|  |  | 6- If return exception, the system will display “No suitable slots”. |
| Includes: | There’s any use case include. | |
| Notes and Issues: | 200 users park in at the same time without breaking down. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC2 | |
| Use Case Name: | Pick slot. | |
| Actors: | Vehicle’s user. | |
| Pre-conditions: | When there are free slots in garage. | |
| Post-conditions: | The system displays slot for vehicle’s user. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-The system will choose slot based on owner garage’s choice (first slot or best slot). |
|  | 2-if owner garage’s choice is first fit the system will check if slot is suitable or not. |
|  | 3- if the slot is suitable the system will choose it if not the system will check the next slot until finds suitable slot. |
|  | 4- but if owner garage’s choice is best fit the system will choose slot with the most dimensions nearly to vehicle’s dimensions. |
| Exceptions: | **User Action** | **System Action** |
|  | 1-The system will choose slot based on owner garage’s choice (first slot or best slot). |
|  | 2-if owner garage’s choice is first fit the system will check if slot is suitable or not. |
|  | 3- if the slot isn’t suitable the system will check the next slot until finds suitable slot. |
|  | 4- if the system doesn’t find any suitable slot the system will return exception. |
| **User Action** | **System Action** |
|  | 1-The system will choose slot based on owner garage’s choice (first slot or best slot). |
|  | 2-if owner garage’s choice is best fit the system will search about slot with the most dimensions nearly to vehicle’s dimensions . |
|  |  | 3- if the system doesn’t find any suitable slot the system will return exception. |
| Includes: | Enter vehicle’s data | |
| Notes and Issues: | Picked slot should be easy to be modified and be readable. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC3 | |
| Use Case Name: | Calculate and pay fees. | |
| Actors: | Vehicle’s user. | |
| Pre-conditions: | When Vehicle’s user decides to park-out from garage. | |
| Post-conditions: | After paying all parking fees. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1- System calculates the parking fees by knowing the park-in and park-out time, first calculating time of stay in hours and multiplying by hourly rate (5). |
|  | 2- System displays the calculated fees to user to pay. |
| 3- User will pay fees in cash. |  |
|  | 4- System adds the paid fees in total income. |
|  | 5-System display successful payment message. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- Time that vehicle is parked was calculated wrong.  2- Park-in and park-out time was deleted from data. |
|  | 1- User pays money. | 1-System returns the rest.  2-System returns no enough money message. |
| Includes: | Park-out | |
| Notes and Issues: | This process should take no more than 10 seconds. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC4 | |
| Use Case Name: | Park out. | |
| Actors: | Vehicle’s user and administrator. | |
| Pre-conditions: | Vehicle’s user park in garage. | |
| Post-conditions: | Vehicle’s user Pays fees. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Vehicle’s user chooses park-out. |  |
| 1. Vehicle’s user Enter vehicle ID. |  |
|  | 1. The system will check if id is invalid or not. |
|  | 1. Then calculate and pay fees. |
|  | 5-The system adds the cost to vehicle’s data. |
|  | 6-The system sets left time. |
|  |  | 7-The system adds available . |
| Exceptions: | **User Action** | **System Action** |
| 1. User chooses park out. |  |
| 1. User enters vehicle id. |  |
|  |  | 1. The system doesn’t find vehicle with this Id in the garage. |
|  |  | 1. The system will display “The entered ID not found in garage” |
| Includes: | There’s any use case include. | |
| Notes and Issues: | No more than five steps needed to park out from garage. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC5 | |
| Use Case Name: | Enter garage’s data. | |
| Actors: | Garage’s owner. | |
| Pre-conditions: | When user enters garage’s data. | |
| Post-conditions: | After choosing valid configuration. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enter garage’s data like (name, address, phone number, number of slots, dimensions of each slot). |  |
|  | 2- System displays configurations: first-fit and best-fit. |
|  | 3- User chooses one of these configurations. |  |
|  |  | 4- System adds chosen configuration to data. |
| Exceptions: | **User Action** | **System Action** |
| 1- User chooses configuration. |  |
|  |  | 2- System can’t find this configuration (invalid input). |
| Includes: | There’s any use case include. | |
| Notes and Issues: | Entered data should be stored in secure manner. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC6 | |
| Use Case Name: | Calculate total income. | |
| Actors: | Garage’s owner. | |
| Pre-conditions: | When garage is closed. | |
| Post-conditions: | After calculating total income. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User wants to check the total income. |  |
|  | 2- System calculates total income through adding all paid fees. |
|  |  | 3- System displays total income to user. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- System can’t display total income. |
| Includes: | There’s not any use case include. | |
| Notes and Issues: | Calculating should be done at most 5 second. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC7 | |
| Use Case Name: | Calculate total number of vehicles. | |
| Actors: | Garage’s owner. | |
| Pre-conditions: | When garage is closed. | |
| Post-conditions: | After calculating total number of vehicles. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User wants to check the total number of vehicles. |  |
|  | 2- System calculates total number of vehicles through adding all park-in vehicles. |
|  |  | 3- System displays total numbers of vehicles to user. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- System can’t display total number of vehicles. |
| Includes: | There’s not any use case include. | |
| Notes and Issues: | Calculating should be done at most 5 second. | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| All Project. | Mariam Ashraf Amin |
| All Project. | Hoda Shafek Ahmed |
| All Project. | Reem Ayman Abdel-Fattah |
| All Project. | Mariam Saeid Shawky |