Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Garage System

Software Requirements Specifications (SRS)

Hour Ahmed Mohamed

Nada Mostafa Mohamed

Rana Mohamed Ali

Hussam Ahmed Mahmoud

8/5/2022

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** |
| 20201058 | Hour Ahmed Mohamed Fathi | [Hourahmed11@gmail.com](mailto:Hourahmed11@gmail.com) | 01018525409 |
| 20200182 | Rana Mohamed Ali | Ranamkk55@gmail.com | 01148459501 |
| 20200594 | Nada Mostafa Mohammed Mahmoud Hefnawy | Nadaelhefnawy2002@gmail.com | 01141971413 |
| 20200148 | Hussam Ahmed Mahmoud |  | 01141971413 |

# Document Purpose and Audience

Audience:

This project is a prototype for the garage management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the garage management team and as well as for vehicle drivers.

## Document Purpose is to be a reference to all audiences and especially to the owner to be aware of his system and the manager to be certain in the efficient of his team

## Software Purpose

The purpose of the online garage management system is to ease garage management and to create a convenient and easy-to-use application for drivers (customers), trying to find the most suitable slot to park their cars. The system is based on searching for the most suitable slots according to known configurations Above all, we hope to provide a comfortable user experience along with the best pricing available.

## Software Scope

Our software scope has too many components such as allowing the owner to enter all garage details, software can take the dimension of the car to choose the best slot can car parking. In addition, if the garage has a slot available turn the light to green and the gate will open and take the car id. Our software makes the garage more organized and display to the owner all slots available and all cars in the garage and when any car enters the garage, the system counts it. Moreover, make payment easier for the customer and the owner to receive. And when the customer exit if he tried more than 10 times. The system sends a human helper to help him

# Requirements

## Functional Requirements

**1-The garage customer should enter the model name, Model year and vehicle dimensions.**

**2-System should get car data from the user.**

**3-The system should give each vehicle unique identification number.**

**4-The system should have a park-in function that do the following:**

* **Marks the arrival time of a vehicle if there is an available slot.**
* **Pick a free slot based on the active slot configuration.**
* **Park in function should have 2 configurations :**

1. **1-first come first served slots.**
2. **2-best-fit approach.**

**5-System should send approval messages if there are available slots.**

**6-The application shall capture arrival time automatically from the system.**

**7-System should mark the slot as taken when car arrives and empty when car leaves.**

**8-The system should have a park-out function that do the following:**

* **Marks the departure time of a vehicle from the garage.**

**9-The application shall capture departure time automatically from the system.**

**10-System should calculate the parking fees during the park-out based on the time-of-stay with an hourly rate of 5 EGP.**

**11-System should calculate parking time.**

**12-System should calculate the total income as well as the total number of vehicles that used the parking garage at any given point in time.**

**13-System should display the available parking slots.**

**14-System should display all cars in the garage.**

**15-System should allow the customer to pay by cash or by card.**

**16-System should handle exceptions by displaying descriptive error message for exceptional behavior.**

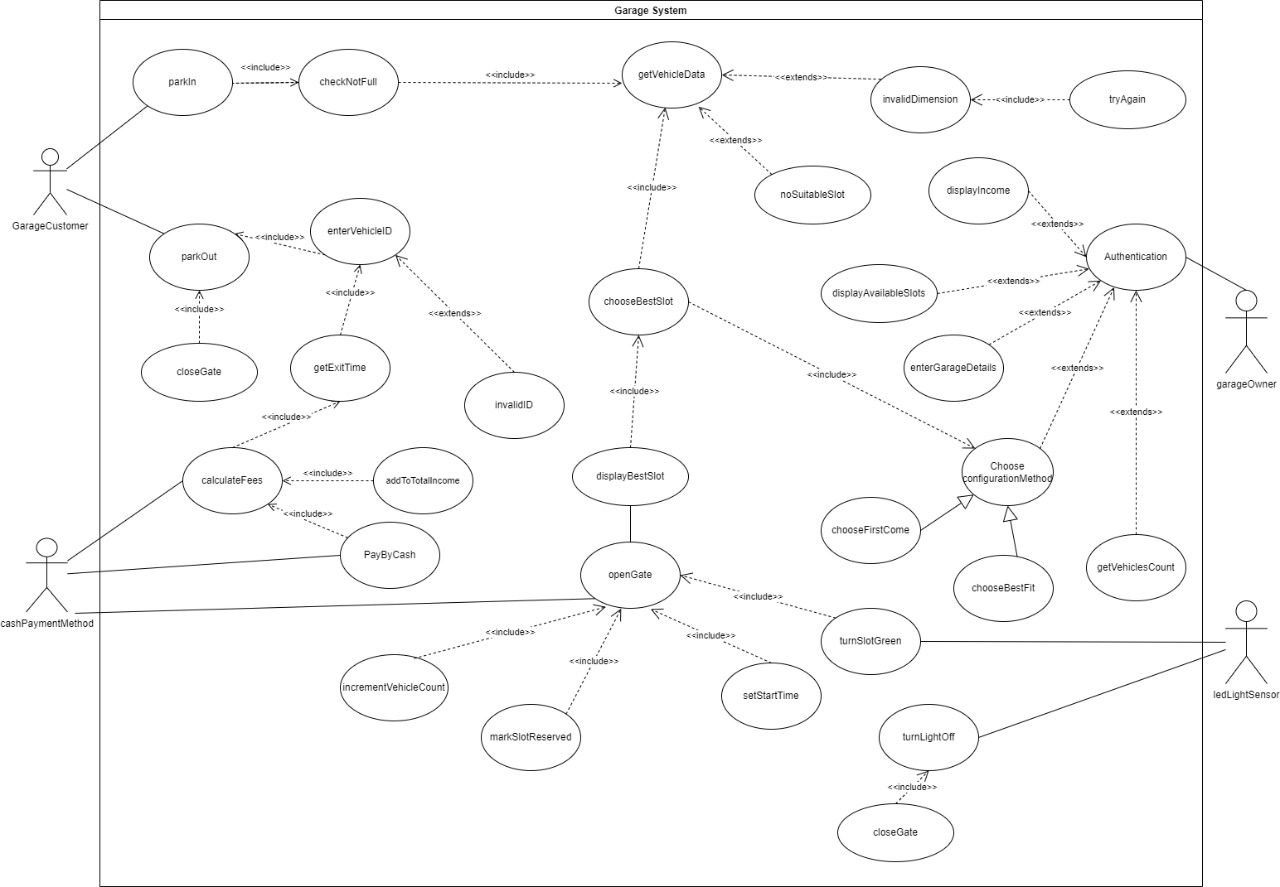
## Non Functional Requirements

1. **If the driver takes more than 1 minute to enter his car information, the process is canceled and program return to start point in case he left. (While entering garage) (availability)**
2. **If the driver enters invalid car dimensions more than 10 times, he is blocked from trying again for one hour. (While entering garage). (safety)**
3. **The system determines the most suitable slot in 20 seconds at most. (Response time)**
4. **The system can be implemented on different type of vehicles including cars, trucks, motorcycles and bikes. (Scalability)**
5. **The system can work and calculate hours in any time of the day. (Availability)**
6. **The display should be clear and readable. (usability)**
   1. **Lines should have a space of at least 3 cm between them.**
   2. **Touch buttons have to be in a bright color, with understandable words on them.**
   3. **A keyboard icon should be available.**
7. **Admin or garage owner can interfere and change garage info if needed. (Adaptability)**
8. **The process of opening the garage door should take 10 clicks at most. (usability)**
9. **Data of vehicles is not shared with anyone except garage owners and police if needed. (safety)**
10. **If the driver enters invalid time more than 10 times when he is leaving, the system send notification for human help. (usability) / (maintainability)**
11. **If an unrecoverable error occurred, the program will restart and display an apology message. (robustness)**
12. **When process of choosing a slot is finished, the system goes back to the starting point in less than 5 seconds. (Response time)**
13. **A car passes from the gate in average 2 minutes. (Response time)**
14. **The system chooses a slot successfully in at least 99% rate. (throughput)**
15. **The system send message to the door to open in less than 5 seconds. (Response time)**
16. **If power is shut from the system, it should recover data from database. (robustness)**

|  |  |
| --- | --- |
| 1. If the driver takes more than 1 minute to enter his car information, the process is canceled and program return to start point in case he left | (availability) |
| 2. If the driver enters invalid car dimensions more than 10 times, he is blocked from trying again for one hour. | (safety) |
| 3. The system determines the most suitable slot in 20 seconds at most. | (Response time) |
| 4. The system can be implemented on different type of vehicles including cars, trucks, motorcycles and bikes. | (Scalability) |
| 5. The system can work and calculate hours in any time of the day | (Availability) |
| 6. The display should be clear and readable. | (usability) |
| 7. Admin or garage owner can interfere and change garage info if needed. | . (Adaptability) |
| 8. The process of opening the garage door should take 10 clicks at most. | (usability) |
| 9. Data of vehicles is not shared with anyone except garage owners and police if needed. | (safety) |
| 10. If the driver enters invalid time more than 10 times when he is leaving, the system send notification for human help. | (usability) / (maintainability) |
| 11. If an unrecoverable error occurred, the program will restart and display an apology message. | (robustness) |
| 12. When process of choosing a slot is finished, the system goes back to the starting point in less than 5 seconds. | (Response time) |
| 13. A car passes from the gate in average 2 minutes. | (Response time) |
| 14. The system chooses a slot successfully in at least 99% rate. | (throughput |
| 15. The system send message to the door to open in less than 5 seconds. | (Response time) |
| 16. If power is shut from the system, it should recover data from database. | (robustness) |

# System Models

## Use Case Model

****

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Setting configuration method | |
| Actors: | Customer | |
| Pre-conditions: | No pre-conditions are required | |
| Post-conditions: | Configuration method setted successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User clicks on admin option through the screen. . |  |
|  | 2- System asks for the admin information. |
| 3- User Enters admin’s info through the screen (username and password). |  |
|  | 4- System sets both admin’s username and password.  5- System checks for Admin’s information validations then displays an approval message to the Admin through the screen. |
| |  | | --- | | 6- User approves the message | |  | | 8-user enter number of slots  9-user set slots count | |  | | 11-user enter Slot Dimension  12- user enter Slot Dimension | |  | | 14-user enter configuration  method  15-set Configuration Method | | |  | | --- | |  | | 7-system asked for slot count | |  | | 10- system asked for SlotDimension | |  | | 13-system asked For Configuration Method | |  | |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid information. |  |
|  | 2- Card is invalid and unreadable.  3- System display main page. |
| Includes: | 1-log in  2-show number of slots  3-show dimension of car  4-show Configuration Method | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Check garage | |
| Actors: | owner | |
| Pre-conditions: | No pre-conditions are required | |
| Post-conditions: | check successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- User clicks on driver option through the screen. . |  |
| 2-ask system to check if the garage is full |  |
|  | 3-system return yes |
|  | 4-system print whether the garage is full or not |
|
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: | 1-checkNotFull | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park in | |
| Actors: | driver | |
| Pre-conditions: | Garage is not full | |
| Post-conditions: | Park in successfully | |
| Flow of eve  nts: | **User Action** | **System Action** |
| 1- User clicks on driver option through the screen. . |  |
| 2-driver asked to Configuration Method |  |
|  | 3-system return “first come” |
| 4-driver asked to get best slot |  |
| |  | | --- | |  | |  | | 8-driver park in | |  | |  | | |  | | --- | | 5-system return slot | | 6-system opens gate  7-system set Vehicles id | |  | | 9-system display Vehicles id | | 10- system set arrival time  11- system increment Vehicles Count  12- system mark slot taken | |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  | . |
| Includes: | 1-park in  2- Configuration Method  3- get best slot  4-display Vehicles id  5- set arrival time  6-increment Vehicles Count | |
| Notes and Issues: | none | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Display cost to driver | |
| Actors: | Driver | |
| Pre-conditions: | Driver enter his car ID | |
| Post-conditions: | Cost is displayed on the screen | |
| Flow of events: | **User Action** | **System Action** |
| 1-User clicks on park-out option through the screen. |  |
|  | 2-System asks for the car ID. |
| 3-Driver enters the car ID. |  |
|  | 4-System checks the car ID.  5-System set departure time.  6-System gets start & End time |
| 7-Driver Parks-out |  |
|  | 8-System marks slot empty.  9-System calculate fees and displays it on the screen. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid car ID |  |
|  | 2- ID invalid.  3- System asks for car ID again. |
| Includes: | 1-askForCarID  2-checkCarID  3-setEndTime  4-getStartTime  5-getEndTime  6-park-out  7-calculateFees  8-printFee  9-setCost  10-displayCost | |
| Notes and Issues: | Driver should enter the same car ID he parked-in with | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Park-out | |
| Actors: | Driver | |
| Pre-conditions: | Driver enter his car ID | |
| Post-conditions: | Driver parks-out | |
| Flow of events: | **User Action** | **System Action** |
| 1-User clicks on park-out option through the screen. |  |
|  | 2-System asks for the car ID. |
| 3-Driver enters the car ID. |  |
|  | 4-System checks the car ID.  5-System set departure time.  6-System gets start & End time |
| 7-Driver Parks-out |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid car ID |  |
|  | 2- ID invalid.  3- System asks for car ID again. |
| Includes: | 1-askForCarID  2-checkCarID  3-setEndTime  4-getStartTime  5-getEndTime  6-park-out | |
| Notes and Issues: | Driver should enter the same car ID he parked-in with | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Display all available slots | |
| Actors: | Garage owner | |
| Pre-conditions: | No pre-conditions are required | |
| Post-conditions: | All available slots is successfully displayed. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User clicks on admin option through the screen. |  |
|  | 2- System asks for the admin information. |
| 3- User Enters admin’s info through the screen (username and password). |  |
|  | 4- System sets both admin’s username and password.  5- System checks for Admin’s information validations then displays an approval message to the Admin through the screen. |
| 6- User approves the message.  7- User chooses to display all available slots |  |
|  |  | 8- System display all available slots to the user through the screen with an approval message. |
|  | 9- User approves message. |  |
|  |  | 10- System askes user for edits. |
|  | 11- User clicks Yes for edit or no for save. |  |
|  |  | 12- if no : system saves the slots as they were .  13- if yes: System askes for new info from the user through the screen. |
|  | 14- User enters new info and clicks save. |  |
|  |  | 15- System updates slots info with new info. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enters admin’s info through the screen (username and password). |  |
|  | 2- System checks for Admin’s information validations then displays a rejection message to the Admin through the screen.  3- System returns to the main page. |
| Includes: | EnterGarageDetails  chooseConfiguration | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Display total income | |
| Actors: | Garage owner | |
| Pre-conditions: | No pre-conditions are required | |
| Post-conditions: | The total income of the garage for a given point of time is successfully displayed. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User clicks on admin option through the screen. |  |
|  | 2- System asks for the admin information. |
| 3- User Enters admin’s info through the screen (username and password). |  |
|  | 4- System sets both admin’s username and password.  5- System checks for Admin’s information validations then displays an approval message to the Admin through the screen. |
| 6- User approves the message. |  |
|  |  | 7- System calculates total income then displays the total income in a message through the screen. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enters admin’s info through the screen (username and password). |  |
|  | 2- System checks for Admin’s information validations then displays an rejection message to the Admin through the screen.  3- System returns to the main page. |
| Includes: | None | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | Payment | |
| Actors: | Driver | |
| Pre-conditions: | Driver enter his car ID | |
| Post-conditions: | Driver paid fees | |
| Flow of events: | **User Action** | **System Action** |
| 1-User clicks on park-out option through the screen. |  |
|  | 2-System asks for the car ID. |
| 3-Driver enters the car ID. |  |
|  | 4-System checks the car ID.  5-System set departure time.  6-System gets start & End time |
| 7-Driver Parks-out |  |
|  | 8-System marks slot empty.  9-System calculate fees and displays it on the screen. |
| 10-Driver pay fees. |  |
|  | 11-System add income to total income. |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter invalid car ID |  |
|  | 2- ID invalid.  3- System asks for car ID again. |
| Includes: | None | |
| Notes and Issues: | Driver should enter the same car ID he parked-in with | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| ParkIn Function | *Hour Ahmed* |
| DisplayAvailableSlot Function | *Nada Mostafa* |
| Use Case Diagram | *Hour Ahmed* |
| Class Diagram | *Nada Mostafa,hossam ahmed* |
| Project details (scope, audience, etc) | *Hossam Ahmed* |
| Functional Requirements | *Rana Mohamed, Nada Mostafa* |
| Non-Functional requirements | *Hour Ahmed* |
| Sequence Diagrams | *Hour Ahmed, Rana Mohamed, Nada Mostafa* |
| Use case description / table | *hossam Ahmed, Nada Mustafa* |
| Code | *Hour Ahmed, Hossam Ahmed, Nada Mostafa, Rana Mohamed* |

# 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.