Project Name

Software Requirements Specifications (SRS)

Team Names

**Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200334 | Ali Mohamed Abdelaty Mohamed | asd.bobo2020@gmail.com | 01119401143 |
| 20200531 | Mariam Nagy Mansour | Nagymarem93@gmail.com | 01127384795 |
| 20200089 | Alaa Khamis Ibrahim | ha2785766@gmail.com | 01009775853 |
| 20200839 | Mohamed Magdy Yassin | mmagdym36@gmail.com | 01061415817 |

**Document Purpose and Audience**

* The goal of this project is to create a Parking Garage application that enable users to park in easily and staff to do their tasks without errors and save information which they need to save so that they can start their work and complete it. We aim to accomplish this within this term so after finishing the customer can read this document and understand it.
* this document will show what's our system and what's App scope and the functions which the system will do.
* our audience is the customer and another people who will review this project.

**Introduction**

This system will simulate Parking Garage application so we will design application that manage garage and enable drivers and staff to do their tasks within this application.

**Purpose**

This application will design for garage system which will enable users to simulate garage and manage it so staff can make some operations to manage garage like checking free and suitable slot and calculate fees and total income...etc., and driver can park in according to available slot in garage which managed by staff.

**Scope**

in our application we will Parking Garage application so we will implement important functions which enable our system to simulate Parking Garage clearly so our application will implement this function:

* **in our system user can park in Garage according to specific system.**
* **we will save vehicle attributes.**
* **system can Calculate the parking fees in a specific time.**
* **calculate time in and time out for arrival vehicle.**
* **Calculate the total income.**
* **Display the available parking slots.**
* **User can search for free slot according to specific system which will choose.**

**Requirements**

**Functional Requirements**

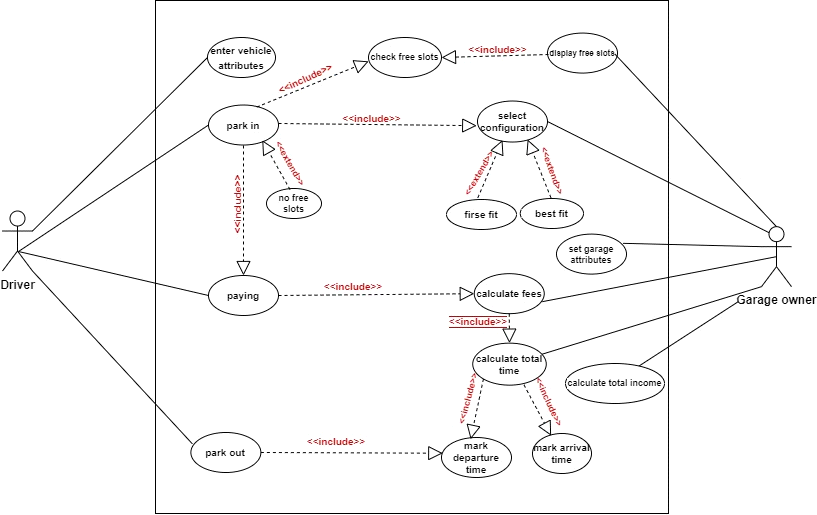
* Garage owner should select configuration (system control) to manage parking in free slot.
* Drivers enter attributes for vehicle.
* Driver can see free slots.
* System can calculate fees based on the time-of-stay with an hourly rate of 5 EGP.
* driver can pay fees.
* System can calculate number of vehicles.
* System can calculate number of free slots.
* system can mark the arrival time of a vehicle if there is an available slot.
* System can mark the leaving time of a vehicle.
* system can calculate the total income as well as the total number of vehicles that used the parking garage at any given point in time.
* System should save information about vehicles and space of slots.
* System should save information about drivers.
* If there is no free slot system will display message to tell user.

**Non-Functional Requirement**

|  |  |
| --- | --- |
|  | Details |
| Reliability | System should be secured for driver information. |
| Performance | System should interact with user quickly and time or any response doesn't exceed 30seconds. |
| Usability | - User can use system easily.  - if there is a problem with usage display number to call for help. |
| Robustness | If there is a problem with payment system should return money for user. |
| Supportability | The developer used the Object-Oriented Programming, so he can easily update the system. |

**System Models**

**Use Case Model**



**Use Case Tables**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | #1 | |
| Use Case Name: | Park in | |
| Actors: | Driver | |
| Pre-conditions: | -Driver go to garage to park his vehicle if there are available slots.  - Driver enter the information of the car name, number, year and its dimension. | |
| Post-conditions: | Driver Park his vehicle successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1- First the driver goes to garage and enter information about vehicle |  |
|  | 2- System check free slots according specific configuration . |
|  | 3- if there free slots system display free slots . |
|  | 4- if there no free slots system tell user and end |
|  | 5- if there free slots system book a specific suitable slot for user |
|  | 6- system marks the arrival time of a vehicle if there is an available slot |
| 7 - user park in |  |
| Exceptions: | **User Action** | **System Action** |
| 1- Driver go to parking. |  |
|  | 2- There is no available slots.  3- System rejects cars.  4- Display an error message that not available slots to park. |
| Includes: | Vehicle information, check available slots, display free slots. | |
| Notes and Issues: | No exist. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | #2 | |
| Use Case Name: | Park out | |
| Actors: | Driver | |
| Pre-conditions: | -Driver take his vehicle to go out from the garage. | |
| Post-conditions: | -Driver Park out successfully.  -Driver pay a park fee. | |
| Flow of events: | **User Action** | **System Action** |
| 1- First driver take his vehicle from the garage. |  |
|  | 2. System record the departure time. |
|  | 3- System calculate the stay time in garage. |
|  | 4- System calculate fees |
| 5-Driver pay fees |  |
| 6 - driver take his/her viecle |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Driver pay with visa. |  |
|  | 2- System display an error message that is not allowed. |
| Includes: | payment, departure time, calculate parking fees. | |
| Notes and Issues: | No exist. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | #3 | |
| Use Case Name: | Calculate total income | |
| Actors: | Garage owner | |
| Pre-conditions: | Calculating the parking fees for each driver. | |
| Post-conditions: | Total income has been calculated successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- garage owner Specifies a specific period for calculating total income and number of vehicles. |  |
|  | 2- System Specifies the number of vehicles in this period. |
|  | 3- System calculate the fees of each vehicle base on the stay time. |
|  | 4- System calculate the total income. |
|  | 5- garage owner take the money successfully. |  |
| Exceptions: | **User Action** | **System Action** |
| 1-garage owner enter a Specifies period (and no vehicle parking in this period). |  |
|  | 2- System display an error message that no money to be calculated. |
| Includes: | Calculate total time, calculate fees. | |
| Notes and Issues: | No exist. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | #4 | |
| Use Case Name: | set configuration | |
| Actors: | owner | |
| Pre-conditions: | -garage isn't prepared | |
| Post-conditions: | - owner set garage configuration that will used to specify role to check free slot | |
| Flow of events: | **User Action** | **System Action** |
| 1- owner set configuration . |  |
|  | 2. System record configuration and save it to use after . |
|  |  |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Driver cant enter configuration |  |
|  | 2- System can't start without configuration |
| Notes and Issues: | No exist. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | #5 | |
| Use Case Name: | paying | |
| Actors: | driver | |
| Pre-conditions: | -driver making park out | |
| Post-conditions: | - driver pay fees | |
| Flow of events: | **User Action** | **System Action** |
| 1- driver make park out . |  |
|  | 2. System capture time and calculate total time . |
|  | 3. System calculate total fees |
|  | 4. System display fees |
| 5 - driver pay fees |  |
|  | 6. System confirm park out |
| Exceptions: | **User Action** | **System Action** |
| 1-Driver can't pay |  |
|  | 2- System can't cant confirm parking out |
| Notes and Issues: | No exist. | |