Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Automated Garage System

Software Requirements Specifications (SRS)

Nour Muhammed

Mohamed Samy

Tarek Magdy  
Weam Taie

May & 2022

Contents

[Team 3](#_Toc102943388)

[Document Purpose and Audience 3](#_Toc102943389)

[Introduction 3](#_Toc102943390)

[Software Purpose 3](#_Toc102943391)

[Software Scope 3](#_Toc102943392)

[Definitions, acronyms, and abbreviations 3](#_Toc102943393)

[Requirements 4](#_Toc102943394)

[Functional Requirements 4](#_Toc102943395)

[Non-Functional Requirements 4](#_Toc102943396)

[Pseudo requirements: 5](#_Toc102943397)

[System Models 6](#_Toc102943398)

[Use Case Model 6](#_Toc102943399)

[Use Case Tables 6](#_Toc102943400)

[Ownership Report 11](#_Toc102943401)

# 

# Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** | **Group** |
| 20200605 | Nour Muhammed Anwar | Nourmuahmmed36@gmail.com | 01113356660 | 20 |
| 20200446 | Mohamed Samy Atwa | Hellosamysamy4@gmail.com | 01008635843 | 20 |
| 20201107 | Tarek Magdy Zein | tarekzein800@gmail.com | 01066649319 | 20 |
| 20200627 | Weam Samy Taie | weamtaie2002@gmail.com | 01144873292 | 3 |

# Document Purpose and Audience

* The purpose of this documents is to document the process of the whole project of helping the administrator to operate his garage to maximize the profit.
* describe the changes that are occurring in our application to ease the whole process also save time.
* The audience to read this document are Technicians, especially project managers and Executives.
* Generally, it is useful for everyone who make business, economic, administrative, legal, governmental, political decisions on the stuff that the experts and technicians work with.

# Introduction

## Software Purpose

* This software helps the Administrator to manage his garage by adding features like calculation total income, vehicles with fully editable dimensions to be more profitable.
* Also, this software aims to make parking in the garage easy to the driver by using automated parking application pointing to free parking slot.
* this helps in reducing time of parking also reducing carbon gas that emit from the car.

## Software Scope

* Software doesn’t include the API of Banks.
* Software doesn’t include Cash counting.
* software focus on Administrator management side.

## Definitions, acronyms, and abbreviations

* API: An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software

# Requirements

## Functional Requirements

* Log in: Administrators Log in to the system Using his Unique ID so he could manage the garage
* Fill Slot: administrator could fill/edit number of slots and dimensions in the garage
* Display Free slots: it helps the administrator to display the free slots in the garage
* Saving time in DB: system will save the time automatically in the database and retrieve it again in calculations.
* calculate Fees: system will calculate fees to the driver during parking out his car
* Calculate total income at specific point of time: Administrator will be able to calculate garage total income
* Calculate total vehicles at specific point of time: Administrator will be able to calculate garage total Vehicles
* payment: the system will show the way that the driver can pay with either by cash / Visa
* Counting Cash: User will enter the cash into Machine and the Machine will Count the Cash.
* Set configuration: it allows the administrator to choose which method the system will select to Display the free slots to the Driver
* Park in: this feature allow the driver to input vehicle model name , model year , unique ID, and vehicles dimensions it also make system capture the arrival time
* Park out: it display the fees on the screen also make system capture departure time after the driver input vehicle id.

## Non-Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **usability** | * **This application is very easy to use, there is instructions for all steps.** * **Number of clicks which administrator and driver done must be measurable (5 for driver ,4 for the administrator)** |
| **safety** | **Administrator has his own login interface with password database are encrypted** |
| **robustness** | **The system works well if the user enters a required error, and the user can be helped and directed through errors acknowledgment.** |
| **scalability** | **The system can be used by 4 people at the same time, and more than one operation is performed at the same time.** |
| **Response time** | **Ten seconds at most.** |
| **availability** | **The system can be down not more than 5 minutes each week** |
| **maintainability** | **System can be maintained, and new features can be added to it** |
| **functionality** | **This is park garage application that provide the Driver free slot for his car to park.**  **This application saves users time searching for free slots.** |
| **supportability** | **The system supports a lot of functions that could help the user if s/he finds problem to find a hints for user immediately. “Added later”** |

## Pseudo requirements:

|  |  |
| --- | --- |
|  | **Details** |
| **implementation** | **System implemented by Java.** |
| **Operation** | **Only administrator can modify system. Driver has limited features.** |
| **legal** | **The forbidden similarity between ID of slots.** |

# System Models

## Use Case Model

**Diagram

Description automatically generated**

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Park In | |
| Actors: | Driver | |
| Pre-conditions: | Driver selects park in from screen | |
| Post-conditions: | Slot is reserved for the driver | |
| Flow of events: | **User Action** | **System Action** |
| 1.Driver selects park in from screen menu |  |
|  | 2. Garage is created and create vehicle form 3. Vehicle form is appeared to driver |
| 4.Drivers fill the form by adding Vehicle id and model year and model name and vehicle Dimensions 5. Driver submits the form |  |
|  | 6. Garage creates the form in slot class(entity).  7-slot class. checks for available slot.    8- slotDisplayer sends appropriate slot to the driver.  9- Garage delete this slot from available slots. |
| 10.Driver Park in in the slot selected by slot entity. |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Driver fill form with wrong input | 2-No available slot for this car dimensions.  3- screen<<No available >> |
| Includes: | ParkIn | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Total income | |
| Actors: | Administrator | |
| Pre-conditions: | The administrator wants to calculate the total income of the garage. | |
| Post-conditions: | The administrator calculated the total income | |
| Flow of events: | **User Action** | **System Action** |
| 1-administrator selects Calculate income from screen menu.  2-administrator enter ID. |  |
|  | 3. Garage create the log in function and check for the ID form in the slot.  4- Garage display a menu for operations to screen. |
| 5. administrator choose Calculate income from screen. |  |
|  | 6- Garage create totalIncome() function. To calculate the income.  7- Garage display the total in come to screen. |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- administrator fill form with wrong input | 2- No access for administrator.  3- slot displays <<No available >> |
| Includes: | Log in | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Park out | |
| Actors: | driver | |
| Pre-conditions: | The driver wants to park out and pay the fees. | |
| Post-conditions: | The driver parked out when the system capture the departure time . | |
| Flow of events: | **User Action** | **System Action** |
| 1.Driver selects park out from screen menu |  |
|  | 2. Garage recalls the vehicle form 3. Departure time will be captured here.  4- Garage call payment () function to calculate the fee and display it at the screen. |
| 5-the driver choose to pay fees by cash or visa. |  |
| Exceptions: | **User Action** | **System Action** |
| 1-Driver fill form with wrong input | 2- slot displays <<No available >> |
| Includes: | Parking | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Park in | |
| Actors: | Administrator | |
| Pre-conditions: | The administrator wants to maximize the profit. | |
| Post-conditions: | The administrator chooses one of the two methods | |
| Flow of events: | **User Action** | **System Action** |
| 1.the administrator selects Set configurations from screen menu. |  |
|  | 2. Garage sent a form to ensure that he is administrator. |
| 4. administrator. fill the form by adding email and password.  5. administrator log in the system. |  |
|  | 6. Garage saved data in the Garage DB.  7- parking display a menu for operations to screen. |
| 10. administrator choose Set configurations operation with the way of the both. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- administrator fill form with wrong input | 2-No access for administrator.  3- slot displays <<No available >> |
| Includes: | Log in | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Java code | *Tarek Magdy , Mohamed Samy* |
| Class diagram / Sequence/ Use case / Description | *All team understand it 100%* |