Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking Garage App

Software Requirements Specifications (SRS)

|  |  |
| --- | --- |
| **ID** | **Name** |
| 20200442 | Mohamed Hamdy Mohamed |
| 20201043 | Basant Mahmoud Mohamed |
| 20201217 | Walaa Hassan Saad |
| 20201120 | Aliaa Mohamed Saad |

1/6/2022

Contents

[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)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200442 | Mohamed Hamdy Mohamed | Mohamedhamdyy55@gmail.com | 01157417179 |
| 20201043 | Basant Mahmoud Mohamed | basant.elshehamy31@gmail.come | 01121841838 |
| 20201217 | Walaa Hassan Saad | walaa.elbishlawy@gmail.com | 01124525839 |
| 20201120 | Aliaa Mohamed Saad | elbashlawyaliaa@gmail.com | 01117891042 |

# Document Purpose and Audience

**Purpose:**

* In this document the reader can identify all the functional Requirements and nonfunctional requirements that describes our system has.
* Purpose of this document is to show all the analysis components for our system like analysis diagrams (use case diagram) and tables (use case descriptions).

**Audience:**

* Customers that want to make a complete parking garage system for their parking lot/s.
* Users, project Management, the system analysts, and the system designers.
* Persons how are interested in knowing how to make a requirements analysis to a software system in general and to a parking garage app specifically

# Introduction

## Software Purpose

-The system should:

1. Allow the parking employee and admin to check all the available, and unavailable slots in the parking lot which he works in.
2. Allow admin to add new employee or remove any employee information from the system.
3. Calculate the time of parking will be captured automatically by the system.
4. Allow the parking employee to end parking for specific vehicle in any time when the driver tells him that he was leave, and system must capture the time of parking-out automatically.
5. Allow the admin of the parking lot to display the income of any parking lot in any time, system should increase the income every time after a payment process for specify slot done.
6. Allow admin and user to see profile information.
7. Allow admin to update his profile information.

## Software Scope

## The scope of this project is a parking garage system that supports assigning a specific slot to specific vehicle when the user press start parking, system should calculate the time of parking and the total parking money that should be pied , calculate the income of specific parking lots and the number of cars that use this parking lot, support modify and delete slots by the admin , support displaying the income and number of cars to user, display user profile to any one use , support update profile by any user of the system, any modify on data must be stored in the system and update the information in the system after update ,

## -our system doesn’t support payment either by cash or by credit card through the system, the payment done directly between parking employee and driver and system only update the income in every park-out process.

## Functional Requirements

* The use of the system can login, enter his information, and using any features of the system.
* Each driver’s vehicle shall be identified by a model name, unique identification number, Model year and vehicle dimensions (vehicle width and depth).
* The parking lot should have one floor with 300 parking slots where customers can park their vehicles.
* The parking lot should have multiple entry and exit points.
* The system should not allow more vehicles than the maximum capacity of the parking lot, if the parking lot is full.
* Each slot should have a specific dimension to fit different dimensions for multiple vehicles.
* The system should support parking for different types of vehicles like car, truck, van, motorcycle. If all the available slots have dimensions less than the dimensions of the vehicle that want to park, the system should show a message that the slots can’t fit this vehicle, and if all the available slots have dimensions larger than the vehicle that want to park, the system should show a message that the slots will fit but the fee increase.
* Park-in function that marks the arrival time of a vehicle if there is an available slot. The application shall capture such time automatically from the system.
* During the park-in function the application shall pick an available slot based on the active slot configuration. There are two configurations. i) first come first served slots i.e., the park-in function will use the first free slot available that fit the driver vehicles from the parking garage slots.
* Park-out function that marks the departure time of a vehicle from the garage. The application shall capture such time automatically from the system.
* Calculate the parking fees during the park-out based on the time-of-stay with an hourly rate of 5 EGP.
* Customers can pay cash
* Calculate the total income as well as the total number of vehicles that used the parking garage at any given point in time.
* The system can display the available parking slots.

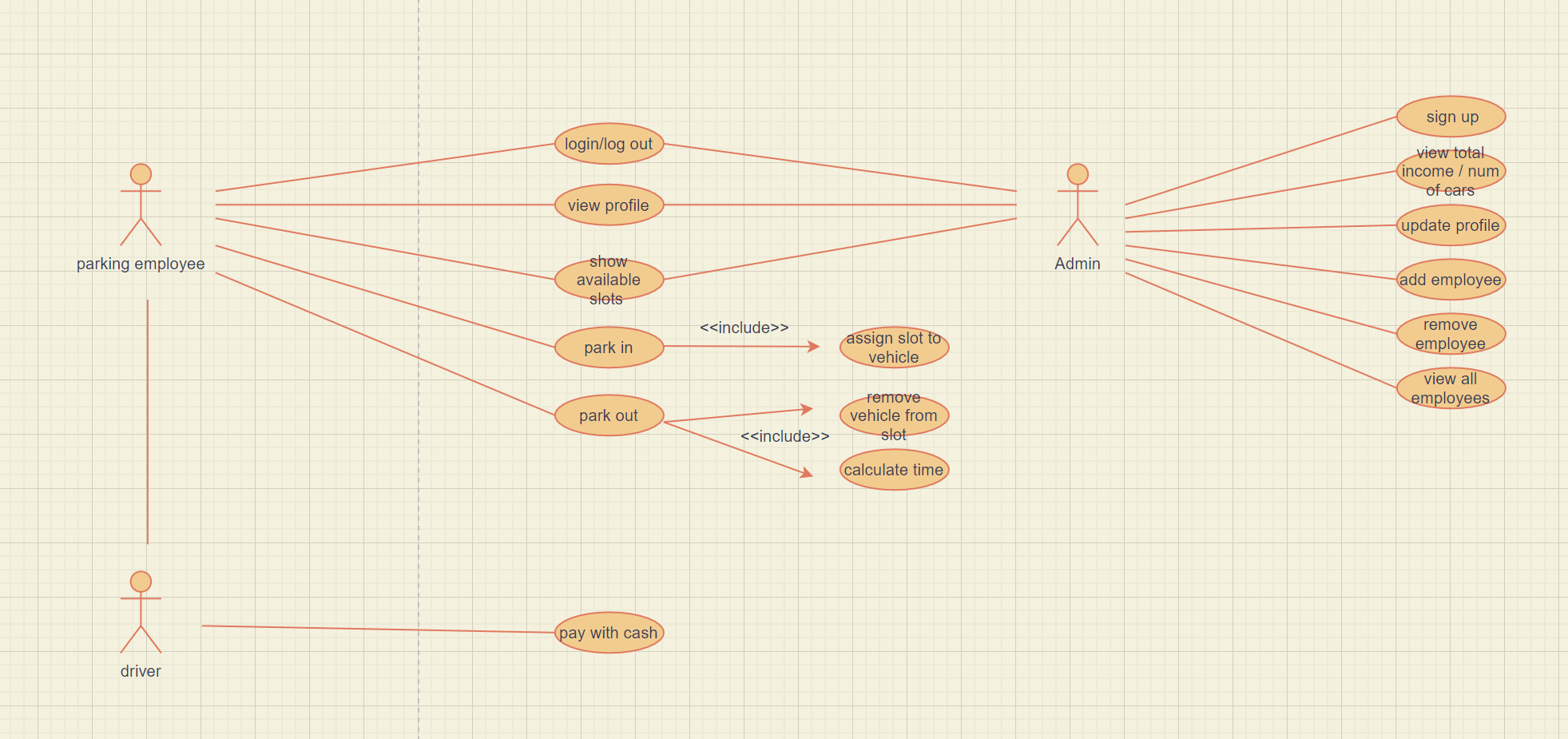
## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **Assign an available slot should take place no more than 7 sec.** |
| **Usability** | * **The employee must reach every function after no more than 7 clicks.** |
| **Reliability** | * **System should not allow assign slots which are occupied.** |
| **Reliability** | * **System should not allow the assign slot of the same parking slot by multiple users at the same time** |
| **Availability** | * **System will be available all weak and it will down 10 minutes in Friday for update.** |

# 

# System Models

## Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “1” | |
| Use Case Name: | Sign up | |
| Actors: | admin | |
| Pre-conditions: | Admin wants to sign up for first time. | |
| Post-conditions: | admin sign up successfully | |
| Flow of events: | **User Action** | **System Action** |
| * admin open system |  |
|  | * The system will display a form to the admin with Name, Phone, Address, User Name, Password fields. |
| * The admin will enter the information and click sign up button |  |
|  | * The system will check the validation of Name, Phone, Address, User Name, Password and display page of admin |
| Exceptions: | **User Action** | **System Action** |
| * If the admin enters invalid user name and password. |  |
|  | * System allow user to enter user name and password again and show forget password. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “2” | |
| Use Case Name: | Login | |
| Actors: | admin, parking employee | |
| Pre-conditions: | The Parking employee wants to login | |
| Post-conditions: | The Parking employee login successfully | |
| Flow of events: | **User Action** | **System Action** |
| * Parking employee open system |  |
|  | * The system will display a form to the Parking employee with user name and password fields |
| * The employee will enter the information and click login button |  |
|  | * The system will check the validation of user name and password and display page of parking employee |
| Exceptions: | **User Action** | **System Action** |
| * If the parking employee enters invalid user name and password. |  |
|  | * System allow user to enter user name and password again and show forget password. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “3” | |
| Use Case Name: | Park in | |
| Actors: | Parking employee | |
| Pre-conditions: | The Parking employee will choose add vehicle , and add vehicle information | |
| Post-conditions: | The vehicle has added successfully, and Parking employee choose start parking  Time park in capture automatically. | |
| Flow of events: | **User Action** | **System Action** |
| * The Parking employee login successfully |  |
| * The Parking employee choose add parking slot |  |
|  | * The system display page includes all fields that he should add vehicle with their information vehicle type * [vehicle id, vehicle model name, vehicle model year] |
| * Parking employee enter information of vehicle |  |
|  | * And system will display do you want to start parking |
| * Parking employee press start parking |  |
|  |  | * System will capture time of parking automatically |
| Exceptions: | **User Action** | **System Action** |
| * Parking employee enters Vehicle id that is already exist |  |
|  | * The system will display message contain (this vehicle is already existing) |
| Includes: | Log in. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “4” | |
| Use Case Name: | Park-out | |
| Actors: | Parking employee | |
| Pre-conditions: | Parking employee clicks on the ‘end parking’ button | |
| Post-conditions: | The system captures the parking out time and go to payment use case | |
| Flow of events: | **User Action** | **System Action** |
| * Parking employee press in remove vehicle |  |
|  | * System will capture time automatically of parking out and calculate time by using park in –park out |
|  | * The parking employee take the money. |  |
|  |  | * The system will remove this vehicle from slot and change the status of the slot from unavailable to available and unassigned the vehicle id from this slot. |
| Exceptions: | **User Action** | **System Action** |
|  | No special requirement |  |
| Include: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “5” | |
| Use Case Name: | Pay with cash | |
| Actors: | Driver and parking employee in the exit point | |
| Pre-conditions: | When parking employee presses end parking button, the driver will pay. | |
| Post-conditions: | Payment done successfully and system will remove vehicle | |
| Flow of events: | **User Action** | **System Action** |
| * The parking employee presses remove vehicle |  |
|  | * system will capture time automatically and display removed successfully |
| * Parking take money from driver |  |
|  | * System adds the money to the total income. |
| Exceptions: | **User Action** | **System Action** |
| * If parking employee clicks on the cancel buttons |  |
|  |  | * The system should reset the park-out time to this slot and return to parking information page. |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “6” | |
| Use Case Name: | View profile. | |
| Actors: | Parking employee, admin. | |
| Pre-conditions: | The parking employee clicks on view profile button. | |
| Post-conditions: | The user viewed his profile and click on exit button and back to home page. | |
| Flow of events: | **User Action** | **System Action** |
| * Admin logs in into system. |  |
|  | * The system takes login information and validate it, login successfully and displays all available options. |
| * Admin clicks on view profile button to see his profile. |  |
|  | * The system displays to the admin its own profile includes an update button in the bottom of the page |
| * The admin sees the information (user name, phone number, name, address, and password.) |  |
| * If the admin clicks on update button. |  |
|  | * The system allows the admin only to update and use case ‘update profile’ will be done |
| * If the admin wants to leave the page he click on exit |  |
|  | * The system makes the admin exit the page and returns to the main page. |
| Exceptions: | **User Action** | **System Action** |
| No exception. |  |
| Includes: | Log in, update profile. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “7” | |
| Use Case Name: | Update profile. | |
| Actors: | admin. | |
| Pre-conditions: | Admin clicks on view profile button and press update profile button. | |
| Post-conditions: | The admin updated his profile and press exit button and return to the home page. | |
| Flow of events: | **User Action** | **System Action** |
| * The admin logs in to system. |  |
|  | * The system displays all available options. |
| * The clicks on view pro admin file button |  |
|  | * The system displays to the admin its own profile includes an update button in the bottom of the page |
| * The admin press ‘update profile’ button |  |
|  | * The system allows the user to update his/her profile information |
| * The admin can change his information (user name, phone number, address, password) |  |
| * Then when the admin has finished changing, he presses the save button. |  |
|  | * The system updates his profile and save changes. |
|  | * The system displays a message says “your update saves successfully” |
| * The admin clicks on exit to leave the page. |  |
|  | * The system makes the admin exit the page and returns to the main page. |
| Exceptions: | **User Action** | **System Action** |
| * The admin wants to update the password and enter password with spaces or weak or invalid. |  |
|  | * The system will display a message contain (“please enter valid password to update.”) and system rejects the update. |
| Includes: | Log in, view profile. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “8” | |
| Use Case Name: | Display income and Number of vehicles | |
| Actors: | Admin | |
| Pre-conditions: | Admin press the ‘display income’ button | |
| Post-conditions: | Admin press the exit button | |
| Flow of events: | **User Action** | **System Action** |
| * Admin will open system and start to log in |  |
|  | * System take log in info and login use case done successfully. |
| * Admin will click total money from spots |  |
|  | * System will display the number of vehicles which use the garage according to each type and the total income. |
| * Admin read the information then press the ‘exit’ button |  |
|  |  | * System closes the page and return to home page. |
| Exceptions: | **User Action** | **System Action** |
| No exception. |  |
| Includes: | Log in. |  |
| Notes and Issues: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “9” | |
| Use Case Name: | Add employee | |
| Actors: | Admin | |
| Pre-conditions: | Admin press the ‘add employee’ button | |
| Post-conditions: | Admin press the exit button | |
| Flow of events: | **User Action** | **System Action** |
| * Admin will open system and start to log in |  |
|  | * System take log in info and login use case done successfully. |
| * Admin will click ‘add employee’ button |  |
|  | * System will form with Name, Phone, address, User Name, Password fields. |
| * Admin enter the information then press ‘save’ button |  |
|  |  | * System will save this information and display message ‘added successfully’. |
|  | * Admin enter ‘exit’ button, |  |
|  |  | * System closes the page and return to home page. |
| Exceptions: | **User Action** | **System Action** |
| * Admin enter information to employee has already exist. |  |
|  |  | * System allow user to enter information again and show’ the employee has already exist’. |
| Includes: | Log in. |  |
| Notes and Issues: |  |  |

# 

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “10” | |
| Use Case Name: | remove employee | |
| Actors: | Admin | |
| Pre-conditions: | Admin press the ‘remove employee’ button | |
| Post-conditions: | Admin press the exit button | |
| Flow of events: | **User Action** | **System Action** |
| * Admin will open system and start to log in |  |
|  | * System take log in info and login use case done successfully. |
| * Admin will click ‘remove employee’ button |  |
|  | * System will form with Name fields. |
| * Admin enter the name then press ‘save’ button |  |
|  |  | * System will delete this employee and display message ‘removed successfully’. |
|  | * Admin enter ‘exit’ button, |  |
|  |  | * System closes the page and return to home page. |
| Exceptions: | **User Action** | **System Action** |
| * Admin enter employee name hasn’t exist. |  |
|  |  | * System allow user to enter name again and show’ the employee hasn’t exist’. |
| Includes: | Log in. |  |
| Notes and Issues: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “11” | |
| Use Case Name: | Display all employees | |
| Actors: | Admin | |
| Pre-conditions: | Admin press the ‘display all employees’ button | |
| Post-conditions: | Admin press the exit button | |
| Flow of events: | **User Action** | **System Action** |
| * Admin will open system and start to log in |  |
|  | * System take log in info and login use case done successfully. |
| * Admin will click ‘display all employees’ button |  |
|  | * System will form with all employees with them information fields. |
|  | * Admin read the information and enter ‘exit’ button, |  |
|  |  | * System closes the page and return to home page. |
| Exceptions: | **User Action** | **System Action** |
| No exceptions |  |
| Includes: | Log in. |  |
| Notes and Issues: |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | “12” | |
| Use Case Name: | Show all spots | |
| Actors: | Admin, parking employee | |
| Pre-conditions: | Admin press the ‘Show all spots’ button | |
| Post-conditions: | Admin press the exit button | |
| Flow of events: | **User Action** | **System Action** |
| * Admin will open system and start to log in |  |
|  | * System take log in info and login use case done successfully. |
| * Admin will click ‘Show all spots’ button |  |
|  | * System will form with all spots. |
|  | * Admin read the information and enter ‘exit’ button, |  |
|  |  | * System closes the page and return to home page. |
| Exceptions: | **User Action** | **System Action** |
| No exceptions |  |
| Includes: | Log in. |  |
| Notes and Issues: |  |  |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| All use case descriptions, functional requirements, nonfunctional requirements, document purpose and audience and introduction | Mohamed Hamdy Mohamed |
| All use case descriptions, functional requirements, nonfunctional requirements, document purpose and audience and introduction | Basant Mahmoud Mohamed |
| All use case descriptions, functional requirements, nonfunctional requirements, document purpose and audience and introduction | Walaa Hassan Saad |
| All use case descriptions, functional requirements, nonfunctional requirements, document purpose and audience and introduction | Aliaa Mohamed Saad |