Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Term Project: Parking Garage application

Software Requirements Specifications (SRS)

Avengers

May 2022

Contents

[Author 3](#_Toc102937930)

[Document Purpose and Audience 3](#_Toc102937931)

[Introduction 3](#_Toc102937932)

[Software Purpose 3](#_Toc102937933)

[Software Scope 3](#_Toc102937934)

[Definitions, acronyms, and abbreviations 4](#_Toc102937935)

[Requirements 4](#_Toc102937936)

[Functional Requirements 4](#_Toc102937937)

[Non Functional Requirements 5](#_Toc102937938)

[System Models 6](#_Toc102937939)

[Use Case Model 6](#_Toc102937940)

[Use Case Tables 6](#_Toc102937941)

# Author

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20201075 | Reem Ahmed Khalil Mohamed Taha | reem1892005@gmail.com | 01033493140 |

# Document Purpose and Audience

* **This is a requirement specification document for the Parking Garage Application, It contains the functional and non-functional requirements besides the use case diagram and use case description table.**
* **The Targeted audience is the customer, the garage owner and the vehicle owner.**

# Introduction

## Software Purpose

* **A software that manages parking space for a configurable maximum number of vehicles, it takes the vehicle identifications and picks a free slot for the vehicle to park in it, in addition to calculating parking fees.**

## Software Scope

* **This software manages parking spaces for a number of vehicles, each space has dimensions, depth and width**
* **This software takes vehicle identification and information, and vehicle dimensions such depth and width**
* **The software has a park-in function that marks the arrival time of a vehicle and it picks a free slot for the vehicle to park in based on active slot configurations which are : 1)** **first come first served slots which means that the park-in function will use the first free slot available from the parking garage slots and make the vehicle park in it if the dimensions fit. 2)** **best-fit approach where you need to find the slot with the minimum dimension to hold the vehicle, that suit the dimensions of the vehicle perfectly. This software has a park-out function that marks the departure time of the vehicle.**
* **The software has a function to calculate the parking fees in the garage based on the duration (the time of stay) in the garage during the park out function. And a function to calculate the total income based on the number of vehicles in the garage at any point in time.**
* **The software has a function to display the available parking slots.**
* **The system handles wrong input data and displays an error message to the user.**

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Slot** | **A parking space that has width and depth as dimensions and a part of the garage.** |
| **first come first served configuration** | **the park-in function will use the first free slot available from the parking garage slots** |
| **best-fit approach configuration** | **It finds the slot with the minimum dimension to hold the vehicle.** |
| **UC** | **Short for use case, in the IDs of use cases** |

# Requirements

## Functional Requirements

* **The system manages a parking space in a garage for a number of vehicles.**
* **The system allows the garage owner to set up the garage with number of slots and set their dimensions (width and depth).**
* **The system allows the garage owner to determine which active slot configuration he needs to select (first in first served or best fit approach).**
* **The system allows the vehicle owner to identify their vehicle (enter the vehicle information).**
* **The system marks the arrival time of the vehicle in the garage during park in function and select an available free slot based of the active slot configuration the garage owner has selected and display the available slot.**
* **The system marks the departure time of the vehicle from the garage during park out function after looking for the vehicle using its identification number.**
* **The system calculates the parking fees of the customer based on the time of stay in the garage during park out with hourly fee of 5 EGP.**
* **The system allows the vehicle owner to pay parking fees with cash.**
* **The system calculates the total income of the garage after calculating the number of vehicles that left and payed their parking fees.**
* **The system handles wrong input data and displays an error message to the user.**
* **The system displays a message to inform the vehicle owner that the system couldn’t find the vehicle identification number in case the vehicle owner couldn’t identify their vehicle.**

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Reliability** | **System crash should not result in data loss, all the data is secured. Error message will be displayed in case of invalid input.** |
| **Usability** | **The system is designed with a user-friendly environment and offers ease of use for the user. Owner can set up the garage system with minimum number of 3 steps.** |
| **Performance** | **The system should not allow the booking of the same parking slot by multiple users at the same time.**  **The vehicle owner should have an available slot to park in with-in 30 seconds.** |
| **Performance** | **All user input data must be acknowledged by the system in less than 2 seconds.** |
| **Performance** | **All error messages must be displayed to the user in less than 2 seconds.** |
| **Availability** | **User can use the system throughout the week at any time during the day. In the case of unplanned system downtime, all features will be available again after one working day.** |
| **Security** | **User can’t access another’s user’s vehicle info.** |
| **Robustness** | **The system maintains available even when the user enters invalid input.** |

# System Models

## Use Case Model

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-01 | |
| Use Case Name: | ParkIn | |
| Actors: | 1. Vehicle Owner. | |
| Pre-conditions: | 1. There must be available free slots to park the vehicle. 2. The garage owner must set up the garage with the slots’ dimensions and the number of slots. 3. The garage owner while starting the application selects the active slot configuration he needs to use (first in first served or best-fit approach). | |
| Post-conditions: | The vehicle is parked in a slot successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Vehicle Owner enters the garage with their vehicle |  |
|  | 2- System checks the selected active slot configuration method (i.e. the first in first served or the best-fit approach) that was chosen by the garage owner. |
| 3- Owner enter their vehicle information. |  |
|  | 4- System looks for the available slot and adds the vehicle to this slot using the chosen configuration  5- System marks the arrival time of the vehicle in the garage.  6- System increase the number of vehicles by one  7- System display a success message |
| Exceptions: | **User Action** | **System Action** |
| 1- Vehicle Owner enters the garage with their vehicle |  |
|  | 2- System checks the selected active slot configuration method (i.e. the first in first served or the best-fit approach) that was chosen by the garage owner. |
|  | 3- Owner enter their vehicle information. |  |
|  |  | 4- System looks for the available slot and adds the vehicle to this slot using the chosen configuration.  5- System displays that they did not find an available slot. |
| Includes: | 1. MarkTimeOfArrival. 2. IdentifyVehicle | |
| Notes and Issues: | None. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-02 | |
| Use Case Name: | ParkOut | |
| Actors: | 1. Vehicle Owner. | |
| Pre-conditions: | 1. Vehicle owner has a parked vehicle in a slot in the garage. 2. Vehicle owner identifies their vehicle. | |
| Post-conditions: | Vehicle parks out the slot successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Vehicle owner parks out the slot  2- Vehicle owner identify their vehicle. |  |
|  | 3- System looks for the vehicle using its identification number  4- System marks the departure time of the vehicle it found.  5- System calculates parking fees based on the time-of-stay with an hourly rate of 5 EGP and displays it. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- Vehicle owner parks out the slot  2- Vehicle owner identify their vehicle. |  |
|  |  | 3- System looks for the vehicle using its identification number.  4- System does not find the vehicle because the identification number is not available.  5- System displays an error. |
| Includes: | MarkTimeOfDeparture. | |
| Notes and Issues: | None. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-03 | |
| Use Case Name: | CalculateTotalIncome | |
| Actors: | 1. Garage Owner | |
| Pre-conditions: | The parked vehicles have left the garage and their vehicle owners payed the parking fees. | |
| Post-conditions: | Calculates the number of total cars that parked and left the garage as well as the total income at any given point in time. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Vehicle owner parks out the slot  2- Vehicle owner identify their vehicle. |  |
|  | 3- System looks for vehicle using identification number.  4- System marks the time of departure.  5- System calculates parking fees during the park-out based on the time-of-stay and display it. |
|  | 6- System adds those fees to the total income |
| Exceptions: | **User Action** | **System Action** |
| 1- Vehicle owner parks out the slot  2- Vehicle owner identify their vehicle. |  |
|  |  | 3- System looks for vehicle using identification number.  4- System can not find the vehicle with that number.  5- System displays error.  6- The total income will not be affected |
| Includes: | CalculateNumberOfVehicles | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC-04 | |
| Use Case Name: | SetUpTheGarage | |
| Actors: | 1. Garage Owner. | |
| Pre-conditions: | Garage owner has a garage with a space to hold a number of vehicles. | |
| Post-conditions: | Garage owner sets the garage successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Garage owner sets up the number of slots in the garage.  2- Garage owner sets the depth and width of each slot.  3- Garage owner selects the active slot configuration method (i.e. first in first served or best-fit approach). |  |
|  | 4- System displays available slots  5- System is set. |
| Exceptions: | **User Action** | **System Action** |
|  | 1- Garage owner sets up the number of slots in the garage.  2- Garage owner sets the depth and width of each slot.  3- Garage owner selects the active slot configuration method (i.e. first in first served or best-fit approach). |  |
|  |  | 4-System displays an error that the configuration chosen is not available |
| Includes: | 1. SetSlotWidth. 2. SetSlotDepth. 3. SetMaxNumOfSlots | |
| Notes and Issues: | The configuration part is related to the park in function. | |