Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Parking Garage Application

Software Requirements Specifications (SRS)

Ziad Ibrahim, Hamza Abdul-Hameed, Mahmoud Yassin, Omar Abdel-Aziz

5 / 2022

Contents

[Team 3](#_Toc102942951)

[Document Purpose and Audience 3](#_Toc102942952)

[Purpose 3](#_Toc102942953)

[Audience 3](#_Toc102942954)

[Introduction 4](#_Toc102942955)

[Software Purpose 4](#_Toc102942956)

[Software Scope 4](#_Toc102942957)

[Definitions, acronyms, and abbreviations 4](#_Toc102942958)

[Requirements 4](#_Toc102942959)

[Functional Requirements 4](#_Toc102942960)

[Non Functional Requirements 5](#_Toc102942961)

[System Models 6](#_Toc102942962)

[Use Case Model 6](#_Toc102942963)

[Use Case Tables 7](#_Toc102942964)

[Ownership Report 14](#_Toc102942965)

[Policy Regarding Plagiarism: 14](#_Toc102942966)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200193 | Ziad Ibrahim Galal | Ziad.ebrahim00@gmail.com | 01127381814 |
| 20200162 | Hamza Abdul-Hameed Ali | hamzaemad598@gmail.com | 01122090518 |
| 20200506 | Mahmoud Yassin Mahmoud | yassinmahmoudyassin@gmail.com | 01145826595 |
| 20200831 | Omar Abdel-Aziz El-Sayed | [Omarcyclist7@gmail.com](mailto:Omarcyclist7@gmail.com) | 01099611380 |

# Document Purpose and Audience

## Purpose

The purpose of this Software Requirements Specification document is to collect and analyze all requirements for the parking garage project such as function and non-functional requirements and system models (use case diagram and use case description). To enhance our understanding of the Software Design Specification document

## Audience

The intended audience for this document includes the developers of the project and vehicle drivers. Developers should use this document in the development phase to structure the design of each part.

# Introduction

## Software Purpose

The purpose of this software is to implement a parking garage system that can accommodate a certain amount of vehicles for parking.

## Software Scope

The scope of this software is designing a parking garage system with:

* Configurable slots
* 2 approaches for searching for an appropriate slot that suits the vehicle
  + Best-fit approach
  + First come first serve approach
* A way to show the customer (driver) his whole time-of-stay and total fees upon parking out.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| **ASCA** | * **Active slot configuration algorithm:**   **The algorithm used to search for an appropriate slot for the vehicle** |
| **WHS** | * **Whole time-of-stay** |
| **BFA** | * **Best-first approach:**   **Approach in which the garage will select the slot with the best dimensions that match the vehicle** |
| **FCFSA** | * **First come first serve approach:**   **Approach in which the garage will select the first slot with big enough dimensions to accommodate the vehicle** |

# Requirements

## Functional Requirements

* The ability to add new parking slots and update parking slots data.
* Keeping track of the number of cars currently at the garage.
* System’s ability to differentiate between vehicles based on their dimensions, ID and model.
* Selecting the appropriate parking slot for each vehicle.
* The ability to park vehicle in the selected parking slot.
* System shouldn't allow the booking of same parking slot by multiple users.
* The system should mark parked slots as taken then keep track of the number of remaining empty slots in the garage.
* The ability to park out vehicle and free up the parking slot.
* Calculate parking time from parking in to parking out.
* Calculate parking fees according to The whole time-of-stay at the garage with an hourly rate of 5 EGP per hour.
* Billing the user the calculated amount on parking out.
* The system should have a reliable payment method.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **The parking system shouldn't take more than 5 seconds to handle parking requests.** |
| **Scalability** | * **The system can handle up to 250 cars at the same time.** |
| **Availability** | * **The parking system should be able to work 24/7.** |
| **Robustness** | * **The parking system should be able to handle exceptions and display an error message.** |
| **Disaster Recovery** | * **Ability of parking system to recover lost data in case of an error.** |
| **Adaptability** | * **The ability of the admin to view all parking slots.** |
| **Usability** | * **User should be notified in case of successful parking or unsuccessful parking.** |

# System Models

## Use Case Model

## Diagram Description automatically generated

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Park in | |
| Actors: | Driver | |
| Pre-conditions: | The Driver wants to park his vehicle in. | |
| Post-conditions: | The driver receives the slot number which he should park in from the system. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver opens the application. |  |
|  | 2- System displays two options, park in and park out. |
| 3- Driver chooses to park in. |  |
|  | 4- system displays a form asking to fill in the vehicle data which is: vehicle type, model name, model year and dimensions. |
| 5- Driver enters his vehicle’s data. |  |
|  | 6- The system assigns a suitable slot to his vehicle and the displays its number to the driver. |
| Exception 1: | **User Action** | **System Action** |
| 1- Driver chooses to park in. |  |
|  | 2- System displays a message that the garage is full. |
| Exception 2: | **User Action** | **System Action** |
|  | 1- Driver enters his vehicle’s data. |  |
|  |  | 2- The system finds that the vehicle’s size is bigger than any available slot and notifies the driver. |
| Includes: | Fill Vehicle Info use case, Assign Slot use case | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Park out | |
| Actors: | Driver | |
| Pre-conditions: | The driver is already registered | |
| Post-conditions: | The driver successfully parks out and paid his fees | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver opens the application. |  |
|  | 2- System shows park in and park out. |
| 3- Driver selects park out from the list. |  |
|  | 4- system asks the user to enter his vehicle’s slot ID. |
| 5- Driver enters the requested slot ID. |  |
|  | 4- system displays whole time-of-stay and total amount of parking fees. |
| 5- Driver chooses checkout and pays his fees. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- Driver enters the requested slot ID. |  |
|  | 2- System displays an error message saying that he isn’t parked in. |
| Includes: | Includes checkout use case | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Setup garage | |
| Actors: | Admin | |
| Pre-conditions: | Garage is not already set up. | |
| Post-conditions: | The admin sets the garage up including the initial slots’ dimensions and the active slot configuration. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin opens the application. |  |
|  | 2- System displays four options, setup and display, add new parking slot and calculate total income. |
| 3- Admin chooses to setup. |  |
|  | 4- System displays a form that demands for an initial list of slots as well as the active slot configuration algorithm to be chosen. |
| 5- Admin fills out the needed information then clicks setup. |  |
|  | 6- The system stores the initial slots list and the active slot configuration algorithm then notifies the admin. |
| Exceptions: | **User Action** | **System Action** |
| 1- Admin chooses to setup. |  |
|  | 2- System displays an error message saying that garage is already set up. |
| Includes: | Includes Fill garage info use case | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Viewing slots and parked vehicles | |
| Actors: | Admin | |
| Pre-conditions: | Garage is already set up | |
| Post-conditions: | Admin gets a list of requested item | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin opens the application |  |
|  |  | 2- System displays setup garage, display, calculate income, and add parking slots options. |
|  | 3- Admin chooses view option |  |
|  |  | 4- System displays free slots, used slots and parked vehicles option. |
|  | 5-Admin chooses used slots option |  |
|  |  | 6- System displays used slots in a list. |
| Alternative flow 1 | **User Action** | **System Action** |
|  | 5- Admin chooses free slots option |  |
|  |  | 6- System displays free slots in a list |
| Alternative flow 2 | **User Action** | **System Action** |
|  | 5- Admin chooses free slots option |  |
|  |  | 6- System displays free slots in a list |
| Alternative flow 3 | **User Action** | **System Action** |
|  | 5- Admin chooses free slots option |  |
|  |  | 6- System displays free slots in a list |
| Exceptions: | **User Action** | **System Action** |
| 1- Admin chooses display option |  |
|  | 2- System displays a message saying that garager is not set up |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | Calculate the total income | |
| Actors: | admin | |
| Pre-conditions: | Garage is already set up. | |
| Post-conditions: | The admin successful calculate total income | |
| Flow of events: | **User Action** | **System Action** |
| 1- admin opens the garage application |  |
|  | 2- system shows display, Setup garage, add new slot, and calculate total income. |
| 3- admin selects total income |  |
|  | 4- System display the total income and total number of cars. |
| Exceptions: | **User Action** | **System Action** |
| 1- admin selects total income |  |
|  | 2- System displays an error message saying that no cars parked in since setup |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Add a parking slot | |
| Actors: | Admin | |
| Pre-conditions: | The admin already set up the garage | |
| Post-conditions: | The admin successfully adds a parking slot | |
| Flow of events: | **User Action** | **System Action** |
| 1- Admin opens the application |  |
|  | 2- System displays four options: Setup, display, add a parking slot and calculate total income. |
| 3- admin chooses to add a parking slot. |  |
|  | 4- system display a form asking to fill the slot dimensions |
| 5- Admin enter width and length for slot he want to add. |  |
|  |  | 6- System will give new id to new slot and notifies the admin. |
| Exceptions: | **User Action** | **System Action** |
| 1- Admin chooses to add a parking slot |  |
|  | 2- System displays a message saying that there is no space for new parking slots. |
| Includes: | Includes Fill slot info use case | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Introduction | *Ziad Ibrahim* |
| Functional and non-Functional requirements | *Ziad Ibrahim, Hamza Abdul-Hameed* |
| Use case Diagram | *Ziad Ibrahim, Hamza Abdul Hameed, Mahmoud Yassin, Omar Abdel-Aziz* |
| Use Case Table 1 | *Hamza Abdul-Hameed* |
| Use Case Table 2 | *Ziad Ibrahim* |
| Use Case Table 3 | *Hamza Abdul-Hameed* |
| Use Case Table 4 | *Ziad Ibrahim* |
| Use Case Table 5 | *Mahmoud Yassin* |
| Use Case Table 6 | *Mahmoud Yassin* |