Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Project Name

Software Requirements Specifications (SRS)

Team Names

May-2022

Contents

[Team 3](#_Toc102936512)

[Document Purpose and Audience 3](#_Toc102936513)

[Introduction 3](#_Toc102936514)

[Software Purpose 3](#_Toc102936515)

[Software Scope 3](#_Toc102936516)

[Definitions, acronyms, and abbreviations 3](#_Toc102936517)

[Cash payment 4](#_Toc102936538)

[To pay the parking ticket through cash 4](#_Toc102936539)

[Requirements 4](#_Toc102936540)

[Functional Requirements 4](#_Toc102936541)

[Non-Functional Requirements 5](#_Toc102936542)

[System Models 5](#_Toc102936543)

[Use Case Model 5](#_Toc102936544)

[Use Case Tables 6](#_Toc102936545)

[Ownership Report 7](#_Toc102936546)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200501 | Mahmoud Abdelrady Gadelrab | Mahmoudten55@gmail.com | 01117332086 |
| 20190343 | Ali Mohamed Ali | alimohamed8938893@gmail.com | 01147014302 |
| 20190775 | Hayat Nasser Saeed | hayat8naser@gmail.com | 01015981662 |
| 20200165 | Hanen Mohamed Ewiss | hanenmohamed473@gmail.com | 01110629701 |

# Document Purpose and Audience

* **Document purpose: This document is a clarification about everything about the application and provide every step in building the application.**
* **Document audience: Subject’s professor and TA.**

# Introduction

## Software Purpose

* **Is to track and manage occupancy of a parking garage and allow customers to find and reserve available parking places.**

## Software Scope

* **The parking garage currently operates without any computerized system.**
* **The management has concerns about inefficiencies of sub-optimal usage of parking space (lost opportunity/profit).**
* **Congestion inside the garage is often caused by drivers searching for vacant spots.**
* **it is well known that a great deal of traffic congestion in cities generally is caused by drivers looking for a parking space.**
* **Currently, the management monitors the garage occupancy by having employees walk around the decks to inspect the occupancy of individual spots.**
* **Spots sensor system which keeps tally of the vehicle entrance and exit events that occur in the parking structures.**

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| phrase | Definition |
| Admin | Mainly responsible for adding and modifying parking floors, parking spots, entrance, and exit panels, adding/removing parking attendants, etc |
| Customer | All customers can get a parking ticket and pay for it |
| Parking attendant | Parking attendants can do all the activities on the customer’s behalf, and can take cash for ticket payment |
| System | To display messages on different info panels, as well as assigning and removing a vehicle from a parking spot |
| Add/Remove/Edit parking floor | To add, remove or modify a parking floor from the system. Each floor can have its own display board to show free parking spots |
| Add/Remove/Edit parking spot | To add, remove or modify a parking spot on a parking floor |
| Add/Remove a parking attendant | To add or remove a parking attendant from the system |
| Take ticket | To provide customers with a new parking ticket when entering the parking lot |
| Scan ticket | To scan a ticket to find out the total charge |
| Cash payment | To pay the parking ticket through cash |

# Requirements

## Functional Requirements

* The application should capture the time correctly from the system.
* The application should mark the arrival time of the vehicle correctly based on the time it captured from the system.
* The application should allow the garage owner to specify the garage’s specifications.
* The parking fees should be calculated without any wrong calculations that might make the customer pay more than he should.
* The total number of vehicles should be calculated correctly so that the total income won’t cause false belief that the income is below the expected.
* The application should display the parking slots without any errors.
* The application should handle any exceptions that can happen during user interaction and through any other calculations.

## Non-Functional Requirements

* The application should provide up to 100 parking slots.
* The application should pick up free slots for multiple vehicles at the same time.
* The application should capture the time from the system instantly.
* The application should calculate the total income withing 1ms.
* The application should be easy to use providing the least number of steps to use any function.

# System Models

## Use Case Model

Diagram

Description automatically generated

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC1 | |
| Use Case Name: | Booking parking | |
| Actors: | Customer | |
| Pre-conditions: | Customer sign into the system | |
| Post-conditions: | System get available slot for the customer vehicle | |
| Flow of events: | **User Action** | **System Action** |
| 1- User enter email and password. |  |
|  | 2- System authenticate user data.  3-System ask the user for vehicle data |
| 4-User insert vehicle data. |  |
|  | 5-System display a list of available slot that fits the vehicle's data. |
| 6-User reserve the suitable slot. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User enter email and password. |  |
|  | 2- User details is invalid and unreadable. |
|  | 3-User search for available slot. |  |
|  |  | 4-System cannot find available slot. |
| Includes: | None | |
| Notes and Issues: | If there's no slot ,the system returns an empty list. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | UC2 | |
| Use Case Name: | Modify parking slot | |
| Actors: | Admin | |
| Pre-conditions: | Admin view accounts that sign in the application. | |
| Post-conditions: | Admin modify the slots list. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Admin open the system to insert data. 2. Admin insert the number of slots and the dimensions | 3-System display empty list of slots to fill it with data. |
|  | 4-System display selection button to choose which configuration admin want. |
| 4-Admin select the configuration which customers will park their vehicle based on it. |  |
|  | 5-System save the data. |
| 6-Admin view accounts of customers who sign in and insert data of their vehicles.  7-Admin view list of slots that he filled and specify a slot for each customer. |  |
|  |  | 8-System save the data that the admin insert . |
| Exceptions: | **User Action** | **System Action** |
| 1- Admin want to select the configuration that he wants. |  |
|  | 2- System display only one configuration |
|  | 3-Admin specify a slot depending on vehicle data. |  |
|  |  | 4-System cannot find available slot that fits vehicle data. |
| Includes: | None | |
| Notes and Issues: | None | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Document Purpose and Audience -Functional Requirements - Non Functional Requirements | *Mahmoud Abdelrady Gadelrab* |
| Software purpose - Software Scope - Definitions | *Ali Mohamed Ali* |
| Use Case Model | *Hayat Nasser Saeed* |
| Use Case Tables | *Hanen Mohamed Ewiss* |