Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

**Parking Garage application**

Software Requirements Specifications (SRS)

|  |  |
| --- | --- |
| **Name** | **ID** |
| Abdelrahman Mohamed Ramadan | 20200293 |
| Ziyad Ashraf Ali (s24) | 20200197 |
| Nouran Ahmed Abdelaziz | 20200609 |

May / 2022

Contents

[Instructions [To be removed] 3](#_Toc101814799)

[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)

[Policy Regarding Plagiarism: 6](#_Toc101814813)

# 

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200609 | Nouran Ahmed Abdelaziz | noranahmed5617@gmail.com | 01115743944 |
| 20200293 | Abdelrahman Mohamed Ramadan | abdra1396@gmail.com | 01145062539 |
| 20200197 | Ziyad Ashraf Ali (s24) | ziyadazab36@gmail.com | 01026603488 |

# Document Purpose and Audience

**Document Purpose:** This document is to specify the requirements of the parking garage application, by clarifying the functional and nonfunctional requirements and the usecase model.

And to specify the purpose of the parking garage application

**Document Audience:** The Customer.

# Introduction

## Software Purpose

The parking garage application is a software system that will make it easy on the drivers, it will offer a more enjoyable and time-saving experience.

It will save the drivers time, instead of searching around the city for a parking slot, the application will save this time offering us a smart parking, thanks to the internet of Things and sensor technology.

This application will make it easy on the garage attendant as well, as it calculates the fees of each driver parking out.

## Software Scope

* The driver pays cash, we don't support credit card.
* The application has two configurations, first come first served and best-fit approach, it doesn’t support any other configuration.
* The application system marks the arrival time and the departure time.
* The application can calculate fees using the arrival time and the departure time.
* The application can calculate the total income to the garage.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Best-fit approach** | it specifies the ideal slot to park in that has a dimension suits the car dimensions |
| **Single Responsibility** | Each class in the code performs one functionality no more |
| **Open-closed** | parkIn class, we use it as an abstract class to facilitate further extensions in the code. |

# Requirements

## Functional Requirements

A parking garage application manages a parking space for a configurable maximum number of vehicles and displays the available parking slots for the driver.

Parking garage application marks arrival time of a vehicle if there is an available slot and marks the departure time of a vehicle from the garage.

Parking garage application must ask every vehicle driver about his vehicle’s model name and model year, generate and give him an identification number and calculate the vehicle width and depth.

The parking garage application has two configurations the driver choose one of them, which are: 1) first come first served 2) best-fit approach.

Each slot has dimension (width and depth) so the application matches a parking slot for each vehicle by the model name, unique id number, model year and vehicle dimensions.

The parking garage application calculates the parking fees for each vehicle, this application calculates the total income as well by calculating the total number of vehicles that used the garage at a particular time

When the **driver** is parking out, the **Garage Attendant** confirms the payment method

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Usability** | * Any driver, regardless of his age, should be able to deal with the system and be able to enter and exit easily from the garage |
| **Reliability** | * As Parking Garage application manages a parking space for a configurable maximum number of vehicles and each parking space defined with a dimension, so should not be software errors that lead to parking in places other than its place or calculating the fees incorrectly |
| **Performance** | * Response time for a driver action whether he is entering data and selecting configurations must be in less than 1 minute. |
| **Availability** | * The system shouldn't go down for more than 5 minutes a week |
| **Scalability** | * System should be able to support up to 20 driver at the same moment |
| **Supportability** | * The system must be able to add new slots without modifications to existing system and the code should be flexible to changes such as adding or removing configurations |

# System Models

## Use Case Model

Diagram, schematic

Description automatically generated

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Parking in | |
| Actors: | Driver | |
| Pre-conditions: | Driver enters vehicle data | |
| Post-conditions: | System capture arrival time  Or no slots available | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver enter vehicle data |  |
|  | 2- System increment index of slots  3-System check available slots  4-park in If there are available slots. |
| 5- Driver parks in the slot that System choose. |  |
|  | 6-System capture arrival time. |
|
| Exceptions: | **User Action** | **System Action** |
| 1- Driver enter vehicle data. |  |
|  | 2-System check available slots.  3- System display not available  if there are not available slots. |
| Includes: | Enter vehicle data  Capture arrival time | |
| Quality requirements: | The system should give response less than 2 sec. | |

# 

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Parking out | |
| Actors: | Driver | |
| Pre-conditions: | Driver leaves parking slot | |
| Post-conditions: | Driver pays parking fees | |
| Flow of events: | **User Action** | **System Action** |
| 1- Driver parks out |  |
|  | 2- System decreases number of slots  3- System capture departure time  4-System calculate fees. |
| 4- Driver pays his fees |  |
| Includes: | Calculate parking fees.  Capture departure time.  Payment. | |
| Quality requirements: | Correct fees calculation. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Set Up | |
| Actors: | Administrator | |
| Pre-conditions: | Admin opens System. | |
| Post-conditions: | System Stores given data | |
| Flow of events: | **User Action** | **System Action** |
| 1.admin sets maximum number of slots |  |
| 2.admin selects configuration |  |
| 3.admin enters each slot dimensions |  |
| Includes: | Enter max number of slots.  Select configurations. | |
| Quality requirements: | System Stores given data successfully. | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |