A picture containing text

Description automatically generatedCairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering, I**

Parking Garage application

Software Design

* Marwa Ahmed Mohamed Mubarak
* Mariam Tarek Mohamed Amin
* Reham Rashad Ahmed Mohamed
* Mayar Ahmed Ibrahim Farag

6/2022

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** |
| 20200514 | Marwa Ahmed Mohamed Mubarak | marwamubarak6@gmail.com | 01112274338 |
| 20200524 | Mariam Tarek Mohamed Amin | mariam0155357@gmail.com | 01553573228 |
| 20200190 | Rehem Rashad Ahmed Mohamed | rehamrashad2002@gmail.com | 01032390170 |
| 20200572 | Mayer Ahmed Ibrahim Farag | mayarahmed11211@gmail.com | 01270318645 |

# Document Purpose and Audience

What this document is?

-This document talks about the software requirements (functional and nonfunctional) and the system models for garage parking system and explain the flow of events at the whole system.

-The target audience to read this document:

1. Garage owner

2.The car drivers

# Introduction

## Software Purpose

This system to make the process of the parking very easy for the driver and to the owner of the garage.

## Software Scope

The system support fast park in and park out and easy system to use and can support suitable slot for every car and easy to display the available slots

## Definitions, acronyms, and abbreviations

First come slot: the park-in function will use the first free slot available from the parking garage slots.

Best fit slot: where you need to find the slot with the minimum dimension to hold the vehicle.

# Requirements

## Functional Requirements

**Garage:**

**Garage getInstance ():** function make the class garage can’t have more than one instance.

**getSlot ():** functionthat returnthe dataof specific slot in the garage.

**setSize ():** function that set the number of slots in the garage.

**get Slots ():** function that return the slots of the garage.

**setAllSlots ():** function that set the data of all sots.

**Display:**

**display ():** function that display the availability of specific slot.

**Driver:**

**Driver ():** default constructor.

**setParkingID ():** function set the value of parking id.

**getParkingID ():** function return the value of parking id.

**GarageOwner:**

**setName ():** function set the owner’s name.

**getName ():** function get the owner’s name.

**MarkTime:**

**MarkTime ():** default constructor.

**getTime ():** function return the time.

**ParkIn:**

**parkInSlot ():** function used when driver want to park in get in parameters the configuration used in the garage, width, depth, and array of slots we check with the configuration type if there is available slot or not.

**ParkOut:**

**parkOut ():** function get in its parameters array of slots and parking idand search in slots for this parking id if it is available set end time for the vehicle and make slot available, if the id is not available return that is invalid id.

**Slot:**

**getWidth ():** function that get the width of the slot.

**getDepth ():** function that get the depth of the slot.

**setAvailable ():** function that make the slot available.

**getAvailable ():** function check if the slot is available or not.

**setStartTime ():** function that set the start time which the car has been used the slot.

**getStartTime ():** function return the time which the car has been used the slot.

**setEndTime ():** function that set the end time which the car has been leaved the slot.

**getEndTime ():** function that return the end time which the car has been leaved the slot.

**setWidth ():** function that set the width of the slot.

**setDepth ():** function that set the depth of the slot.

**TouchScreen:**

**Touch ():** function touch button screen.

And there is an implementation for the interface of touchscreen.

**parkingIn: touch ():** function touch on parking in button.

**PrkingOut: touch ():** function touch on parking out button.

**Vehicle:**

**setVehicleInfo():**function let the customer adding info of new vehicle to the system by a model name, unique identification number, Model year and vehicle dimensions (vehicle width and depth).

**getWidth():**function return the width of the car.

**getDepth():**function return the depth of the car.

**getModelName():**function return the model name of the car.

**getUniqueIdentification():**function return the unique id of the car.

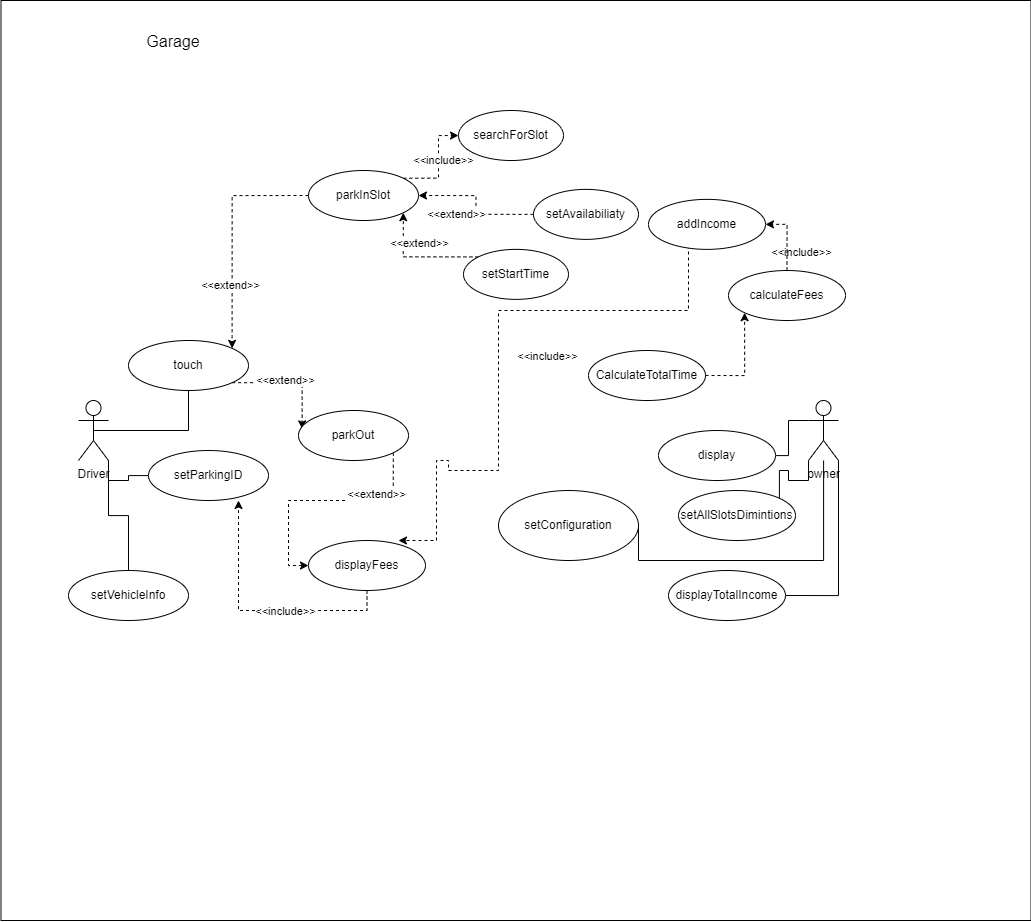
**getModelYear ():** function return the Model Year of the car.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **Withdraw operation will be done within 20 second.** |
| **Scalability** | * **System should be able to support up to 1000 simultaneous drivers.** |
| **Efficiency** | * **system will complete the user process request within the five seconds.** |
| **Availability** | * **the system will work for almost 24/7.** |
| **Data Integrity** | * **this system will maintain consistence and accuracy of the collected and stored data.** |
| **Usability** | * **system will be of higher quality. It is acceptable that the customers** **perform tasks through the user-friendly manual guide.** |

# System Models

## Use Case Model

****

## Use Case Tables

**Table 1:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Parking in the garage successfully | |
| Actors: | 1-Owner  2-Driver | |
| Pre-conditions: | 1-Owner of the garage should insert the numbers of slots in garage  2- Owner of the garage should insert all the dimensions to every slot  3- Owner of the garage shall choose one of these configurations to the whole Garage system.  (i) first come first served slots i.e., the park-in  function will use the first free slot available from the parking garage slots. (ii) best-fit approach where you need to find the slot with the minimum dimension to hold the vehicle  4-The user” driver” should inter a correct data (width, depth, model name, model year) | |
| Post-conditions: | The user parked his vehicle successfully | |
| Flow of events: | **User Action** | **System Action** |
| 1-The owner enter to the system and insert the numbers of slots, the dimensions to every slot and choose one of these configurations to the whole Garage system. |  |
|  | 2-The system accept the garage and slots information |
|
| 3- the driver touches the screen asking to park in and set the vehicle info. |  |
|  | 4- system will call parkinslot function in parkin class to search about suitable slot according to the configuration then set the availability with false and set startTime for the vehicle. |
| 5- the driver will set parkingID. |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User(owner-Driver) Enter an id slot with negative number or out of range. |  |
|  | 2- id is not valid.  3- System rejects the id slot and ask to inter valid one. |
|  | 4-driver put a negative number in (width, depth, model year) |  |
|  |  | 5- (width, depth, model year) is not valid.  6- System rejects the id slot and ask to inter valid one. |
|  | 7-input the slots width or depth or both with zeros or negative integers |  |
|  |  | -System display an error message to enter again |
| Includes: | Usecase1 include usecase2 | |
| Notes and Issues: | Efficiency: system will complete the user process request within the 5 seconds | |

**Table 2:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | The Garage is full | |
| Actors: | 1-Owner  2-Driver | |
| Pre-conditions: | 1-user ask to park in  (i) first come first served slots i.e., the park-in  function will use the first free slot available from the parking garage slots. (ii) best-fit approach where you need to find the slot with the minimum dimension to hold the vehicle.  2-The user” driver” should inter a correct data (width, depth, model name, model year) | |
| Post-conditions: | The message delivered successfully to the driver | |
| Flow of events: | **User Action** | **System Action** |
| 1- The user asks to park in the garage |  |
|  | 2- System ask the information to user’s vehicle (width, depth, model name, model year) |
| 3- User enter this information |  |
|  | 4-System calculate the available slots  5- System don’t find any slot to park  6- System send a message that is no available slots to park |
| 7- The user gets the message |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User(owner-Driver) Enter an id slot with negative number or out of range |  |
|  | 2- id is not valid.  3- System rejects the id slot and ask to inter valid one. |
|  | 4-driver put a negative integer in (width, depth, model year) |  |
|  |  | 5- (width, depth, model year) is not valid.  6- System rejects the id slot and ask to inter valid one. |
|  | 7-input the slots width or depth or both with zeros or negative integers |  |
| Includes: | Usecase2 is part of usecase1 | |
| Notes and Issues: | Performance: the operation will be done within 20 second | |

**Table 3:**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Park out from the garage | |
| Actors: | 1-Owner  2-Driver | |
| Pre-conditions: | 1. the driver touch on button park out using touch function in parkingOut class. 2. The system call function parkOut. | |
| Post-conditions: | The driver park out successfully. | |
| Flow of events: | **User Action** | **System Action** |
| 1-The driver ask to park out the garage by touching on the button in the screen parkout. |  |
|  | 1. the system calls the function parkout in class parkout. 2. The system set the availability ,set end time , calculate the total time and calculate total fees. 3. The system show fees for the driver |
|
| Exceptions: | **User Action** | **System Action** |
| 1- the driver enters an invalid id |  |
|  | 2-System will return an error message |
| Includes: | Didn’t include from any usecase | |
| Notes and Issues: | Data integrity: this system will maintain consistence and accuracy of the collected and stored data | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Use case tables | *All* |
| Use case model | *Marwa Ahmed* |
| requirements | *Reham Rashad* |
| Document Purpose and Audience | *All* |