Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering I**

Parking Garage application

Software Design

*Represented by:*

|  |  |  |
| --- | --- | --- |
| *Name* | *ID* | *Section Number* |
| Mariam Ashraf Amin Mohamed | 20200520 | S15 |
| Hoda Shafek Ahmed said | 20200619 | S15 |
| Reem Ayman Abdel-Fattah Ibrahim | 20200186 | S15 |
| Mariam Saeid Shawky | 20200521 | S15 |

*June,2022*

Contents

[Team 3](#_Toc101814920)

[Document Purpose and Audience 3](#_Toc101814921)

[System Models 4](#_Toc101814922)

[I. Class diagrams 4](#_Toc101814923)

[Important Algorithm 8](#_Toc101814924)

[II. Sequence diagrams 9](#_Toc101814925)

[Class - Sequence Usage Table 16](#_Toc101814926)

[Ownership Report 18](#_Toc101814927)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200520 | Mariam Ashraf Amin Mohamed | mariamashrafamin@gmail.com | 01144456820 |
| 20200619 | Hoda Shafek Ahmed said | hodashafek28@gmail.com | 01142974462 |
| 20200186 | Reem Ayman Abdel-Fattah Ibrahim | reem.ayman52002@gmail.com | 01098834544 |
| 20200521 | Mariam Saeid Shawky | mariamsaeid142@gmail.com | 01110192619 |

# 

# Document Purpose and Audience

This document is a hard copy that describes the functions and the design part of the application, and its purpose is to design all classes and the interaction between them in a sequence way to clear every detail before implementing code and running the actual application. This design manages a parking space for a configurable maximum number of vehicles as possible. This document is the agreement between the client and the software developer team.

* **Audience who is expected to read this document:**

1. Client (Customer): the client who asks (the team) to design a Parking Garage application to organize parking in his/her garage with his/her own requirements and who pays for this design.
2. All Team Members: the team that is assigned to design this document must read this carefully and can check the document at any time.
3. Software Developers: developer benefits from this document, it helps them in code implementation as it organizes ideas, specifies requirements, and expects output.

# System Models

## I. Class diagrams

**Due to bad quality caused by word,**

**We put the picture of the class diagram in the same folder for better quality and also as a pdf.**

**Diagram, schematic

Description automatically generated**

**Description of the Classes**

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1 | InterParkingGarage | It is an interface class which class ParkingGarage inherits from, and it has the main function but without implantation.  It has 6 functions. |
| 2 | *ParkingGarage* | It is a class that stores data of parking garage, it is like system classes. The application will start to work when an object is created from it. It has 6 attributes.  It has 7 functions which one of them is default constructor and 6 of them are the information of garage required after creating an object:  1) getter: it returns attribute value stored in the class (getTotal, getVehiclecalculations, getvehicleHanding, getSlotHanding).  The rest of its functions are explained in the important algorithm part. |
| 3 | InterVehicle | It is an interface class which class *Vehicle* inherits from, and it has Setters and Getters but without implantation.  It has 12 functions. |
| 4 | *Vehicle* | It is a class that stores data of any new vehicle to the system that uses it to run the application. It has 6 attributes.  It has 14 functions which 2 of them are constructors (default and copy) and the other 10 are:  1) setter: it takes value from user and assign it to the class attributes. (setModelName, setModelYear, setIdentificationNumber, setDimensions, setWidth, setDepth).  2) getter: it returns attribute value stored in the class (getModelName, getModelYear, getIdentificationNumber, getWidth, getDepth, getCard). |
| 5 | InterCard | It is an interface class which class *Card* inherits from, and it has the Setters and Getters.  It has 6 functions. |
| 6 | *Card* | Every vehicle that parks in the garage has its card. The class store any information related to the period in which vehicles stay in the garage like time and fees. It has 5 attributes.  It has 8 functions, one of them is Constructor and the other 7 are:  1) setter: it assigns values to the class attributes. (setId, setSlot).  2) getter: it returns attribute value stored in the class (getId, getSlot, getTime, getTimeAdepter, getTimeCalculation). |
| 7 | InterSlot | It is an interface class which class Slot inherits from, and it has Setters and Getters.  It has 8 functions. |
| 8 | Slot | It stores the data of each slot. It has 4 attributes.  It has 11 functions which 3 of them are constructor (one default and two parameterized) the other 8 are:  1) setter: it assigns values to the class attributes. (setWidth, setDepth, setAvailability, setId).  2) getter: it returns attribute value stored in the class (isAvailable, getWidth, getDepth,getid ). |
| 10 | IException | It is an interface class which class NoFreeSlotException, NoFitSlotException, NoEnoughMoneyException, and idException and DimentionException implement it. |
| 10 | *NoFreeSlotException* | It is a class that handles no free slot exception, it has 2 functions :  1) setMessage : it is a function that set the text message that will display to the user when the exception happens.  2) getMessage : it is a function that returns the message to the user due to the exception happening when system didn’t find any available slots. |
| 11 | *NoFitSlotException* | It is a class that handles no fit slot exception, it has 2 functions :  1) setMessage : it is a function that set the text message that will display to the user when the exception happens.  2) getMessage : it is a function that returns the message to the user due to the exception happening when system didn’t find any suitable slots. |
| 12 | *idException* | It is a class that handles id exception, it has 2 functions :  1) setMessage : it is a function that set the text message that will display to the user when the exception happens.  2) getMessage : it is a function that returns the message to the user due to the exception happening when any exception happened in stored data |
| 13 | *DimentionException* | It is a class that handles Dimention exception, it has 2 functions :  1) setMessage : it is a function that set the text message that will display to the user when the exception happens.  2) getMessage : it is a function that returns the message to the user due to the exception happening when any exception happened in stored data |
| 14 | SlotSelecter | It is an interface class which class firstfitSlot and BestfitSlot implement it. |
| 15 | FirstfitSelector | It Select the first suitable slot.  It has 1 functions is selectslot that return slot. |
| 16 | *BestfitSelector* | It Select the best slot.  It has 1 functions is selectslot that return slot. |
| 17 | *Iconfiguration* | It is an interface class which class Configuration implement it, it has 3 function without implementation. |
| 18 | *Configuration* | It saves the configuration (First fit or Best fit) that system will depend on it.  It has one attribute and 4 functions one of them is constructor and others are SetConfiguration , SetSelector and getConfiguration. |
| 19 | *ITotalincome* | It is an interface class which class Configuration implement it, it has 2 function without implementation. |
| 20 | *Totalincome* | It Calculate total income.  It has two attributes and 4 functions one of them is constructor and others are getinstance, increasetotalincome and calculateTotalincome . |
| 21 | *IvehicleHandling* | It is an interface class which class VehicleHandling implement it, it has 4 function without implementation. |
| 22 | *vehicleHandling* | It Calculate handle the vehicle to park in garage.  It has 3 attributes and 6 functions one of them is constructor and others are getinstance, getNumberOfVehicle, getVehcle, addVehicle and increaseNumberOfVehicle. |
| 23 | *IvehicleCalculation* | It is an interface class which class VehicleCalculation implement it, it has 1 function without implementation. |
| 24 | *vehicleCalculation* | It Calculate the number of vehicles in garage.  It has 1 function is calculateNumberOfvehicle and use class SlotHandling. |
| 25 | *IslotHandling* | It is an interface class which class SlotHandling implement it, it has 7 functions without implementation. |
| 26 | *SlotHandling* | It choose slot depend on configuration .  It has 5 attributes and 11 functions one of them is constructor and others are getInstance, RemoveAvailableSlot, addAvailableSlot, setSlot, setNumbersOfSlots, displayAvailableSlots, getAvailableSlots,getAllSlots, getNumbersOfAvailableSlots and getNumbersOfSlots . |
| 27 | *ITimeSetter* | It is an interface class which class TimeSetter implement it, it has 4 functions without implementation. |
| 28 | *TimeSetter* | It set time that vehicle park in and get out garage.  It has 2 attributes and 5 functions are setTimeIn, setTimeOut, setTimeOfStay, getTimeIn and getTimeOut. |
| 29 | *TimeAdpter* | It calculate time of staying vehicle in garage.  It has 1 attribute and 2 functions are CalculateTimeOfStay and GetTimeOfStay . |
| 30 | *Icalculation* | It is an interface class which class calculate implement it, it has 2 functions without implementation. |
| 31 | *calculation* | It calculate parking fees of garage for vehicle.  It has 1 attribute and 2 functions are CalculateParkingFees and GetParkingFees. |
| 32 | *IParkin* | It is an interface class which class Parkin implement it, it has 2 functions without implementation. |
| 33 | *Parkin* | It selects slot and handle vehicle.  It has 4 attributes and 3 functions are Parkin, Park\_in, getConfiguration . |
| 34 | *IParkOut* | It is an interface class which class ParkOut implement it, it has 1 functions without implementation. |
| 35 | *ParkOut* | It handle slot after vehicle get out from garage.  It has 3 attributes and 2 functions are park\_out and ParkOut. |

### Important Algorithm

## parkIn function: it checks for available slots if there is available slot.

## It asks the user for the vehicle's data (model name, identification number, model year, depth, and width) and set them after that it display available slots and pick slot if there is a fit slot

## It set the time that the user entered the garage in and add the time to the database and then it returns this slot and show the user the slot where he should park then it removes available slot from the database, create new vehicle and add it to the database and increase the number of the vehicles then send that new number to the database)

## on the other hand, if there is no fit slot it returns a message to the user that there is no fit slot)

## if there is no available slot (it returns a message to the user that there is no free slot)

## parkOut function: this function determines the time which the user left the garage in and the how long did he stay in the garage and calculate the parking fees that the user should pay then send them all to database and after the user pay it calculate the rest of the money and once the user leaves it add an available slot to the database

* **selectSlot:** it selects the first available slot that there is no car on it or it select the slot the fit the car the best based on the dimensions (width, depth) of the car, It depends on the class that overrides this function

## II. Sequence diagrams

**Diagram, schematic

Description automatically generatedSequence ID: 1** (park-in)

Diagram, schematic

Description automatically generated**Sequence ID: 2** (pick slot)

**Diagram

Description automatically generatedSequence ID: 3** (calculate and pay fees)

**Diagram, schematic

Description automatically generatedSequence ID: 4** (park out)

**Diagram, schematic

Description automatically generatedSequence ID: 5** (Enter garage’s data)

**Sequence ID: 6** (Calculate total income)

### Diagram Description automatically generated

**Sequence ID: 7** (Calculate number of vehicles)

### Diagram Description automatically generated

### Class - Sequence Usage Table

| **Class Name** | **Sequence Diagrams** | **Overall used methods** |
| --- | --- | --- |
| ParkingGarage | 1,3,4,5,6 |  |
| Parkin | 1,2,5 | -park\_in  -numberOfConfiguration |
| Vehicle | 1 | -setModelName  -setIdentificationNumber  -setModelYear  -setDimensions |
| SlotSelecter | 1 | -selectSlot |
| SlotHandling | 1,4,5 | -RemoveAvailableSlots  -addAvailableSlot  -setNumberOfSlot  -setSlot |
| VehicleHandling | 1 | -addvehicle  -increaseNumberOfVehicles  -getNumberOfVehicle |
| Card | 1,3,4 | -getCard |
| Timesetter | 1,4 | -SetTimein  -setTimeOut  -getTimeOut |
| NoFreeSlotException | 1 | -setMessage  -getMessage |
| FirstFitSelector | 2 | -SelectSlot |
| NoFitSlotException | 2 | -setMessage  -getMessage |
| BestFitSlot | 2 | -selectSlot |
| ParkOut | 3,4 | -park\_out |
| TimeAdapter | 3,4 | -calculateTimeOfStay  -getTimeOfStay |
| Calculation | 3,4 | -calculateParkingFees  -getParkingFees |
| Totalincome | 4,6 | -increaseTotalincome  - calculateTotalincome |
| IdException | 4 | -setMessage  -getMessage |
| Configuration | 5 | -setSelector |
| Slot | 5 | -setWidth  -setDepth |
| DimensionException | 5 | -setMessage  -getMessage |
| VehicleCalculation | 7 | calculateNumberOfVehicles |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| All Project. | Mariam Ashraf Amin |
| All Project. | Hoda Shafek Ahmed |
| All Project. | Reem Ayman Abdel-Fattah |
| All Project. | Mariam Saeid Shawky |