Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering I**

Parking Garage

Software Design

Griffin

June & 2022

Contents

[Instructions [To be removed] 3](#_Toc101814919)

[Team 3](#_Toc101814920)

[Document Purpose and Audience 3](#_Toc101814921)

[System Models 3](#_Toc101814922)

[I. Class diagrams 3](#_Toc101814923)

[Important Algorithm 4](#_Toc101814924)

[II. Sequence diagrams 5](#_Toc101814925)

[Class - Sequence Usage Table 6](#_Toc101814926)

[Ownership Report 6](#_Toc101814927)

[Policy Regarding Plagiarism: 7](#_Toc101814928)

# Team

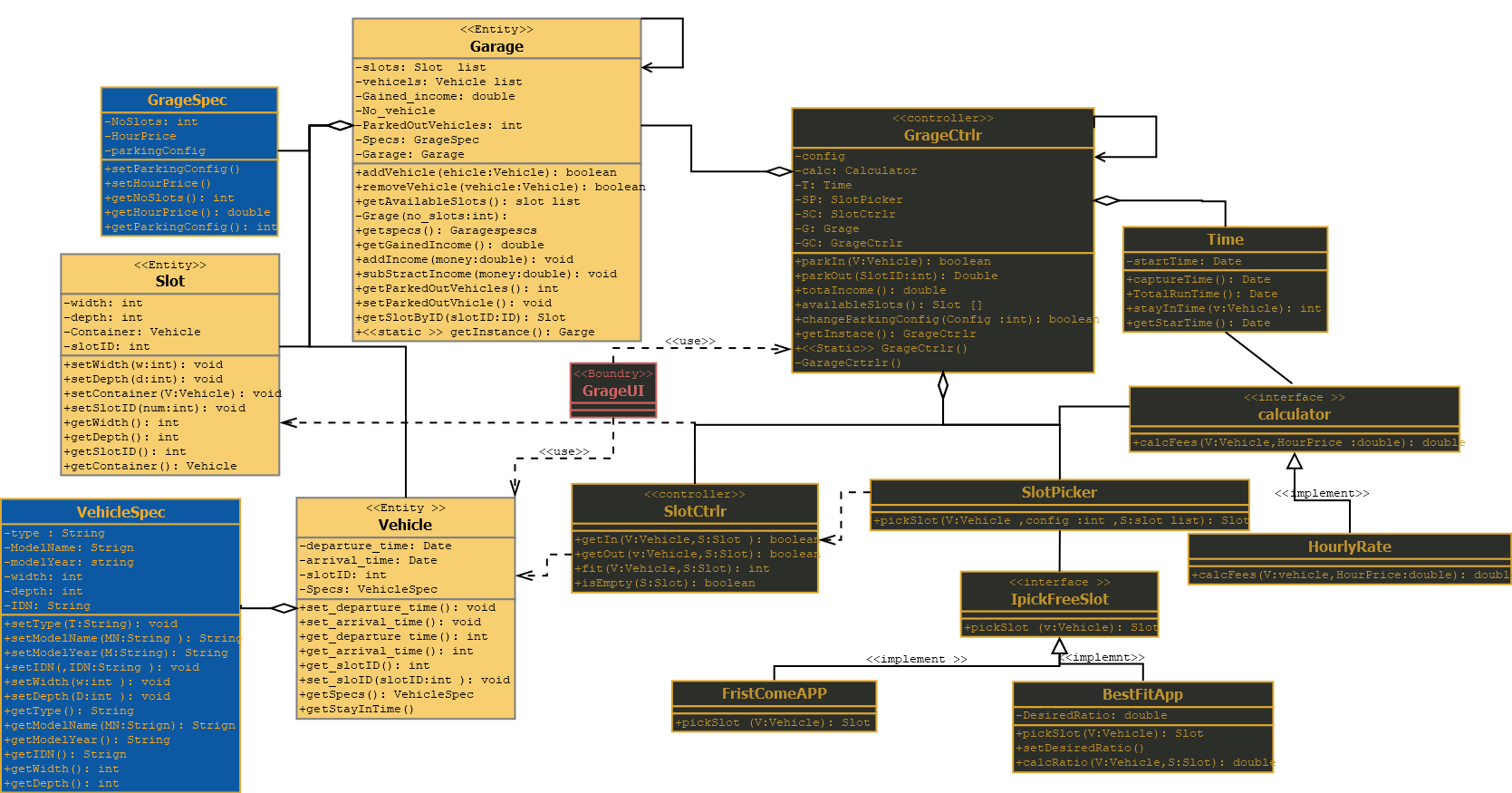
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20200693 | Mohammed Ahmed Saleh | not.m7mdd@gmail.com | 01141835675 |
| 20200149 | Hossam Hassan Farouk | darkknghit40@gmail.com | 01012447770 |
| 20200437 | Mohammed Gamal Ragab | mohamed6688745@gmail.com | 01119899140 |
| 20200086 | Akram Mohammed Hassan | am58627939@gmail.com | 01158627939 |

# Document Purpose and Audience

* This document includes SDS description for Parking Garage application.
* This document includes the class diagram, sequence diagram
* It is meant for all stakeholders, especially the client to understand what features will in the
* system.
* It serves as a guide for the developers to understand what they will develop.

# System Models

## I. Class diagrams



**List down your classes and describe them**

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1 | Slot | This class tack information of the slots like(,width,depth), and give it an id.  It has some setters and getters methods to set and get dimensions of the vehicle |
| 2 | VehicleSpec | This class related to tack vehicle information  Like(name,type,width,length,model) for each vehicle and its id, all methods in this class serve this functionality for all of them. attributes in this class (ModelName, modelYear, width,depth,idn). |
| 3 | Vehicle | This class record the arrival time for each vehicle and departure time , it has number of slot that the vehicle will park in it ,and it has some setter and getters for it attributs |
| 4 | GarageSpec | This class represent the size of the garage  And it capacity ,it contain the number of slots  And from it we can choose one of the two configuration of parking In. |
| 5 | Garage | This class is responsible for managing the entire system and uniting the other classes together. It has vector of slot and vector of vehicle contains all the information of vehicle and all information of slot. we can from it can display all vehicle and slot and system can display Available Parking slots and the system calculate the fees automatically,and  Total income |
| 6 | Time | We make this class to capture the time and to calculate total time that each vehicle stay in the garage . |
| 7 | GarageCtrl | By this class we can park in , park out vehicles, calculate total in come ,show available slots,and change parking configuration |
| 8 | SlotCtrl | This class has some functions like (Fit) to check if this slot suitable for this vehicle ,  Also we can check if the slot is empty or not |
| 9 | Calculator | This class to calculate the fees and total income |
| 10 | SlotPicker | To choose a suitable slot for the vehicle |
| 11 | PickFreeSlot | To pick slot based on two configuration  1-FristCome 2-BestFit |

### Important Algorithm

* Park\_in method should parkin with one of to configurations the owner choose it .

calculateFees Method parking fees during the park-out based on the time-of-stay with an hourly rate of 5 EGP

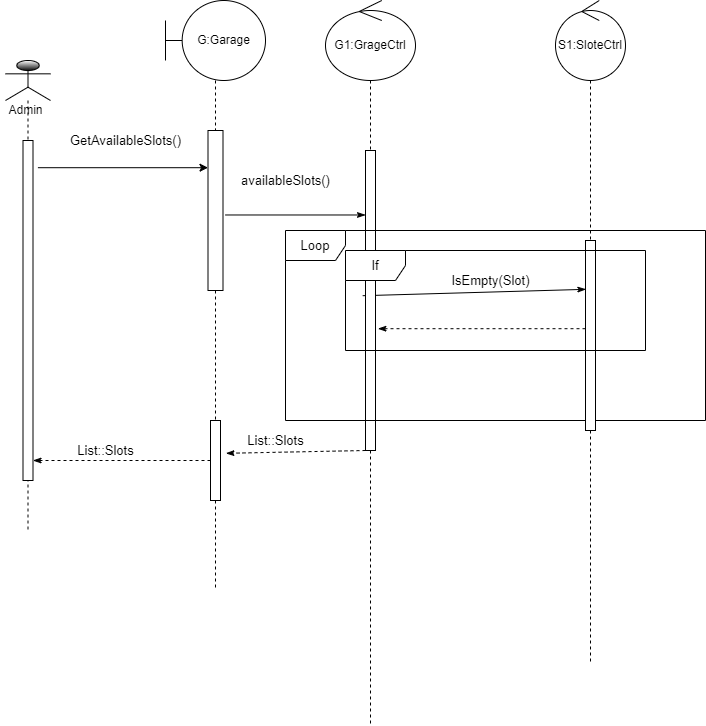
## II. Sequence diagrams

Graphical user interface, application, table

Description automatically generated

Diagram

Description automatically generated with medium confidence



Diagram

Description automatically generated

### Class - Sequence Usage Table

| **Class Name** | **Sequence Diagrams** | **Overall used methods** |
| --- | --- | --- |
| Garage, GarageCtrl, SlotCtrl | Get Available slots | GetAvailableSlots(),availableSlots(),  Isempty() |
| GarageCtrl,Time,Calculator | Calculate total income | Totalincome(),stayinTime(),CalcFees() |
| GarageGUI,GarageCtrl,Slotpicker,SlotCtrl,Slot,Time,Garage | Park in | Parkin(),availableSlot(),getAvailableSlot(),pickSlot(),getIn(),set Constainer(),setSlotID(),CuptureTime(),setArrivalTime(),addVehicle() |
| GarageUI,GarageCtrl,Calculator,SlotCtrl,Slot,Garage,GarageSpec | Park out | parkOut(),getSlotById(),getContainer(),gethourPrice(),calcfees(),stayInTime(),getOut(),removeVehicle() |

# The solid principles are respected in the class diagram because each class has only one responsibility, and the class that does the job in multiple ways is open for expansions but closed for modification. The Liskov principles are achieved because if you substitute the subtypes in ipickFreeslot, the behavior should not change, and no interface has all the methods.

# The higher-level class is unaware of the low-level class's implementation and instead relies on abstraction.

**Design pattern**

Creational Design Patterns:

singleton: Garege , GargeCtrlr

Behavioral Design Patterns:

strategy design pattern: IpickFreeSlot , Calculator