Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering I**

Project Name

Software Design

Team Names

Month & Year

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** |
| 20200575 | Mirette Shenouda Maher | [miretteshenouda01@gmail.com](mailto:miretteshenouda01@gmail.com) | 01223898053 |
| 20200567 | Monica Saeed Habib | monicasaeed12@gmail.com | 01226489661 |
| 20200686 | Youstina Saadawy Thabet | [youstinasaadawy@gmail.com](mailto:youstinasaadawy@gmail.com) | 01207227565 |
| 20200386 | Catherine Ramy Mikhail | [catherineramy02@gmail.com](mailto:catherineramy02@gmail.com) | 01201700544 |

# Document Purpose and Audience

Audience: software developers

Class diagram: They use class diagrams during the development of a system during analysis, system design, object design and implementation.

sequence diagram: are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

Notes:

This document contains the class diagram for a Parking Garage application

It will help developers during analysis and implementation

And it also contains sequence diagrams to imagine the ways a customer can use the system because the sequence diagram is a type of interaction diagram as it describes how and in what order a group of objects work together

# 

# System Models

## I. Class diagrams

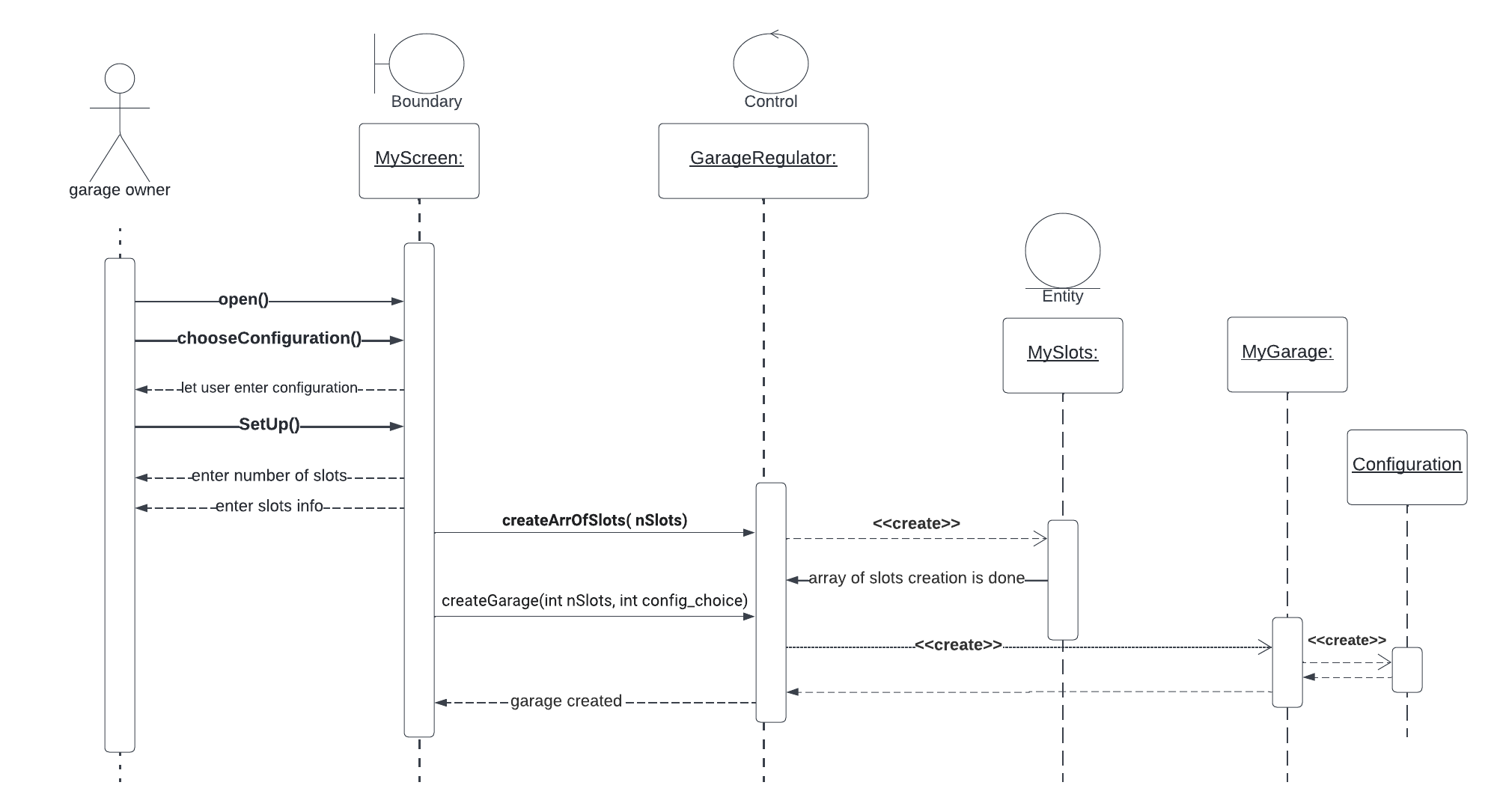
**Diagram, schematic

Description automatically generated**

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1 | MyScreen | This class is mainly use to show the interaction between the system and the users, open function, setup sets the data the garage owner wants, chooseConfigurations() that chooses on of the 2 configurations to display the cars by in the garage , creatGarage() start creating garage by the constraints the garage owner displayed, it also displays available slots, total income of the garage and total number of vehicles |
| 2 | Vehicle | Interface for car class |
| 3 | Car | Type of vehicle, the class contains parametrized constructor, it takes car info, the class also contains data about the slots and calculate the arrival and departure date of the driver |
| 4 | Garage | Interface for class MyGarage |
| 5 | MyGarage | it has parkin in that returns to the user the slot available for him by using choose slot function that use one of the 2 configurations to assign a slot to the user and the configuration used is chosen by the garage owner through displaying his choice in getOwnerChoice() ,and It also has parkout function that display total fees the user will pay.  It has display available slots that shows the garage owner the slots that no one used. |
| 6 | Slot | Interface for Slots class |
| 7 | MySlots | It is the class that takes details about the slot, it has a constructor, checks if the slot is empty or not and assign status to it. and gives the slot an id. |
| 8 | Configuration | Interface for classes FirstCome and BestFit |
| 9 | FirstCome | Type of configuration that is responsible for parking in criteria |
| 10 | BestFit | Type of configuration that is responsible for parking in criteria |
| 11 | GarageCalculations | This class is responsible for calculating total income and total fees. |
| 12 | GarageRegulator | This is the controller of the system that is responsible for making the connection between the screen (as boundary object) and the rest of the system. |

## 

## II. Sequence diagrams

**1)Garage setup**

**Diagram, schematic

Description automatically generated**

**2)Parkin**

**Graphical user interface, diagram

Description automatically generated**

**3)Park out**

Diagram

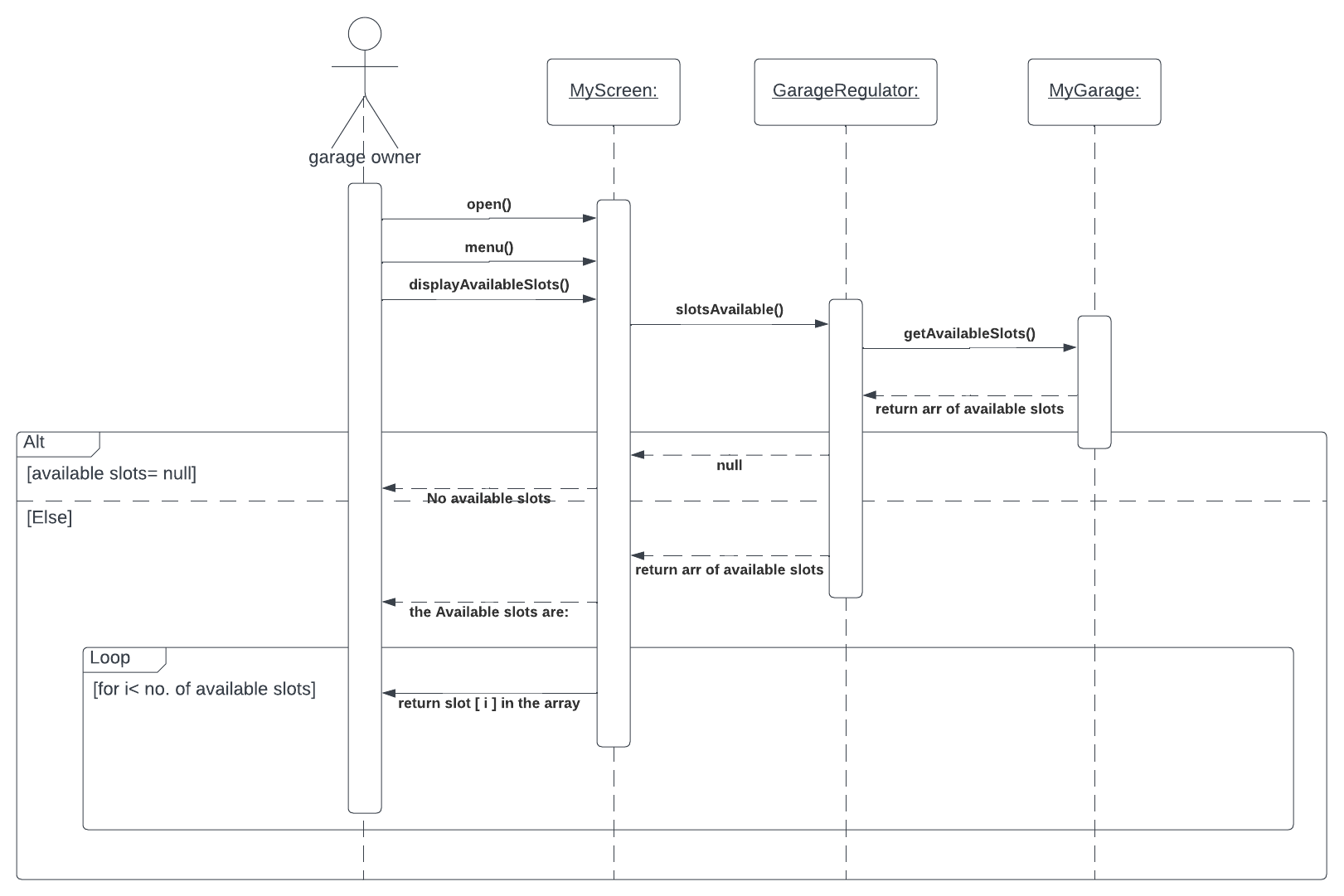
Description automatically generated

**4)Display total number of vehicles**

Diagram

Description automatically generated

**5)Display total income**

 **6)Display available Slots**

### 

### Class - Sequence Usage Table

| **Class Name** | **Sequence Diagrams** | **Overall used methods** |
| --- | --- | --- |
| **<<Boundary>>**  **MyScreen** | **1)Garage setup**  **2)Parkin**  **3)Park out**  **4)Display total number of vehicles**  **5)Display total income**  **6)Display available Slots** | Open()  setup()  setupCar()  choosecontiguration()  displayavalableSlots()  displayParkOut()  displayTotallncome()  displayTotalVehicles()  out()  menu()  userChoice()  getFunChoice() |
| **<<Controller>>**  **GarageRegulator** | **1)Garage setup**  **2)Parkin**  **3)Park out**  **4)Display total number of vehicles**  **5)Display total income**  **6)Display available Slots** | getStatus()  getGarageStatus()  parkIn(vehicle1)  parkOut(vehicleID) getAvailableSlots() getOwnerChoice()  getCapacity():  getSlots(int index)  setSlots(index ,MySlots)  getTotalVehicle()  getTotalIncome() |
| **MyGarage** | **1)Garage setup**  **2)Parkin**  **3)Park out**  **4)Display total number of vehicles**  **5)Display total income**  **6)Display available Slots** | getAvailableSlots() getGarageStatus()  parkIn(vehicle1) parkOut(vehicleID) getOwnerChoice() getTotalVehicle()  getTotalIncome()  getSlots(index)  getStatus()  getCapacity()  setSlots(index,MySlots) |
| **<<Entity>>**  **MySlots** | **1)Garage setup**  **3)Park out** | setData(slotW, slotD,\_slotId)  getwidth()  getdepth()  getId()  setStatus(status)  getStatus() |
| **<<Entity>>**  **Car** | **2)Parkin**  **3)Park out** | setStartDate(arrivalTime)  setEndDate(EndDate)  getStartDate()  getEndDate()  getVehicleDepth()  getVehicleWidth() |
| **GarageCalculations** | **4)Display total number of vehicles**  **5)Display total income** | incrementVehicleCount()  calcFees(vehicle1)  getvehicleCount()  gettotalIncome() |
| **<<Interface>>**  **Configuration** | **1)Garage setup**  **2)Parkin** | incrementVehicleCount()  calcFees(vehicle1)  getvehicleCount()  gettotalIncome() |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Diagram 3,5,6 | *Mirette shenouda maher* |
| Document purpose and audience  Diagram 2, 4 | *Youstina Saadawy Thabet* |
| Class diagram | *Catherine Ramy Mikhail* |
| Sequence usage table  Diagram 1 | *Monica Saeed Habib* |
|  |  |

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.