Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251**

**Software Engineering I**

Term Project: Parking Garage application

Software Design

Avengers

May 2022

Contents

[Author 3](#_Toc104956418)

[Document Purpose and Audience 3](#_Toc104956419)

[System Models 3](#_Toc104956420)

[I. Class diagrams 3](#_Toc104956421)

[Important Algorithm 4](#_Toc104956422)

[II. Sequence diagrams 5](#_Toc104956423)

[Class - Sequence Usage Table 7](#_Toc104956424)

# Author

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20201075 | Reem Ahmed Khalil Mohamed Taha | reem1892005@gmail.com | 01033493140 |

# Document Purpose and Audience

* **This is a design specifications document to show the system components and the sequence**
* **Target audience is the customer (the garage owner) and for the vehicle owner.**

# System Models

## 

## D:\reem\SoftwareProject\SequenceDiagrams\classDiagram.pngI. Class diagrams

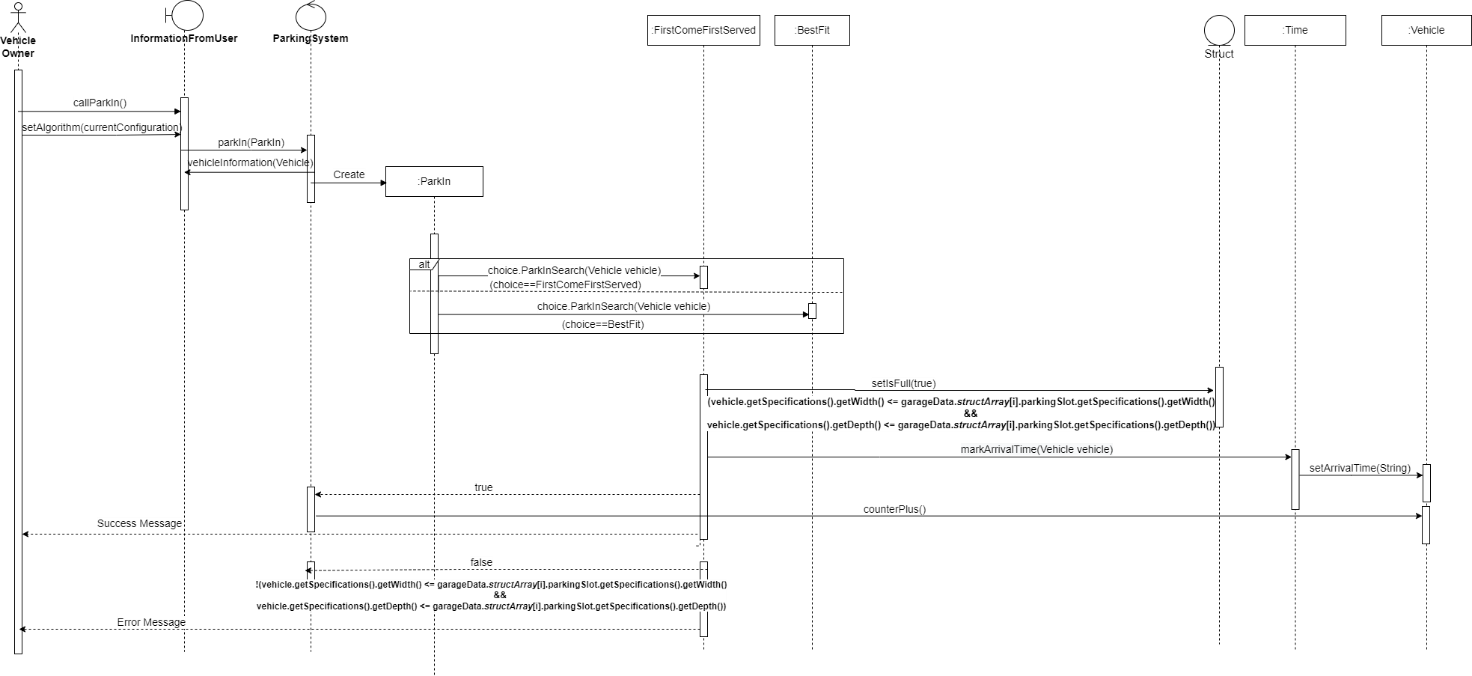
| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| C-01 | InfrormationFromUser | It’s a boundary class that takes in the necessary information/input from user and display information to the user. |
| C-02 | ParkingSystem | It’s a control class that implements the fundamental functions in the system such as park in/ park out, calculate parking fees and so on from the basic functionalities of the system. |
| C-03 | GrageData | It’s an entity class, that stores the number of slots |
| C-04 | Vehicle | A class to hold all information about the vehicle, its model name, model year, id .. etc with setters and getters and functions to mark arrival time of the vehicle in the slot and a function to mark the departure time of the vehicle from the slot. |
| C-05 | ParkingSlot | A class to hold all information about the parking slot, its depth, width, id and status full or not .. etc it has setters and getters for every attribute. |
| C-06 | ParkIn | It’s an interface class, which means it’s completely abstract as it has the function park in with no implementation, to allow using the same function with different implementation in other classes (best fit and first come first served). |
| C-07 | FirstComeFirstServed | A class that inherits from the interface ParkIn class and uses the park in function with the implementation that the park-in function will use the first free slot available from the parking garage slots. |
| C-08 | BestFit | A class that inherits from the interface ParkIn class and uses the park in function with the implementation that the park-in function finds the slot with the minimum dimension to hold the vehicle. |
| C-09 | ParkOut | A class to perform the park-out function. |
| C-10 | ParkingSlotSpecifications | A class to set and get the parking slot width and depth |
| C-11 | VehicleSpecifications | A class to set and get the vehicle width and depth |
| C-12 | Time | A class with two functions to mark the arrival time and the departure time of the vehicle |
| C-13 | CurrentConfiguration | A class to set the chosen configuration |
| C-14 | Calculations | A class that calculates the parking fees of a vehicle and the total income |
| C-15 | Struct | A class that works as a struct of 2 class objects, Vehicle and ParkingSlot |

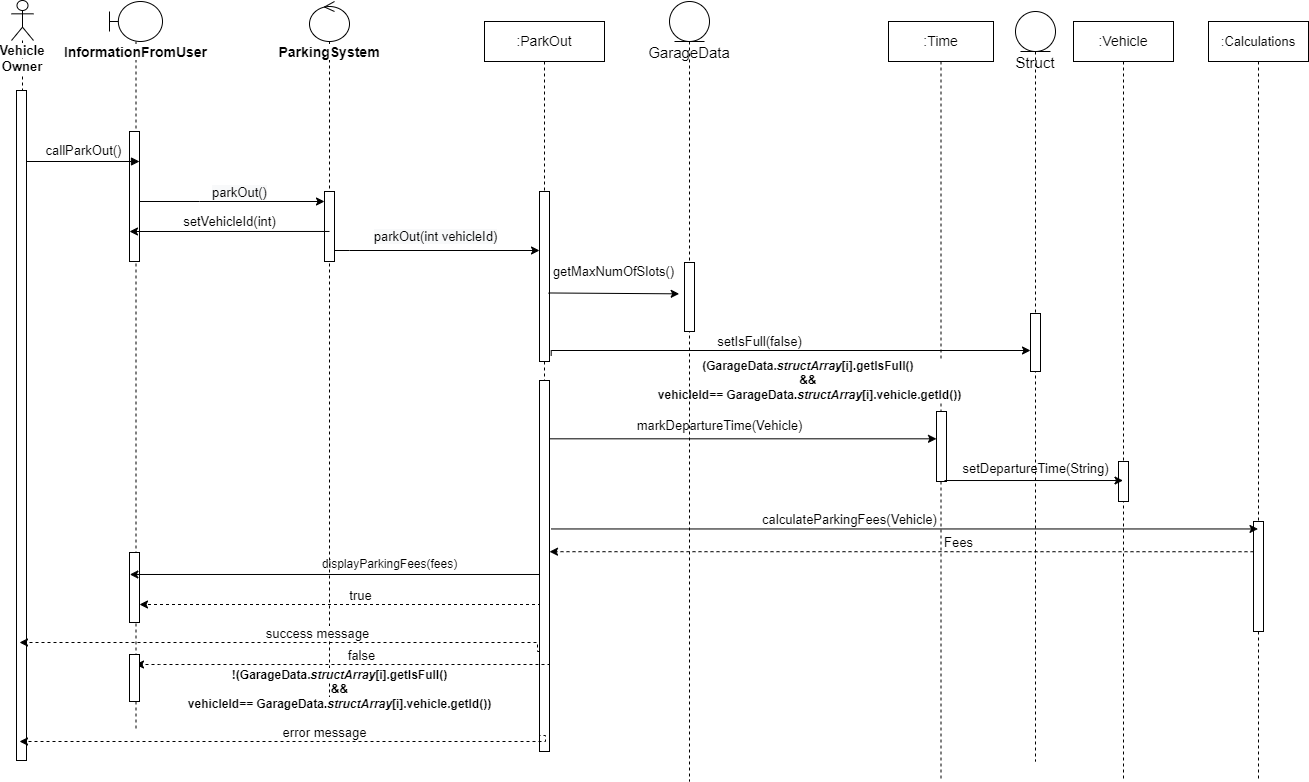
### Important Algorithm

* **Method callParkIn() in the boundary calls the parkIn() function in the system.**
* **Method callParkOut() in the boundary calls the parkOut() function in the system.**

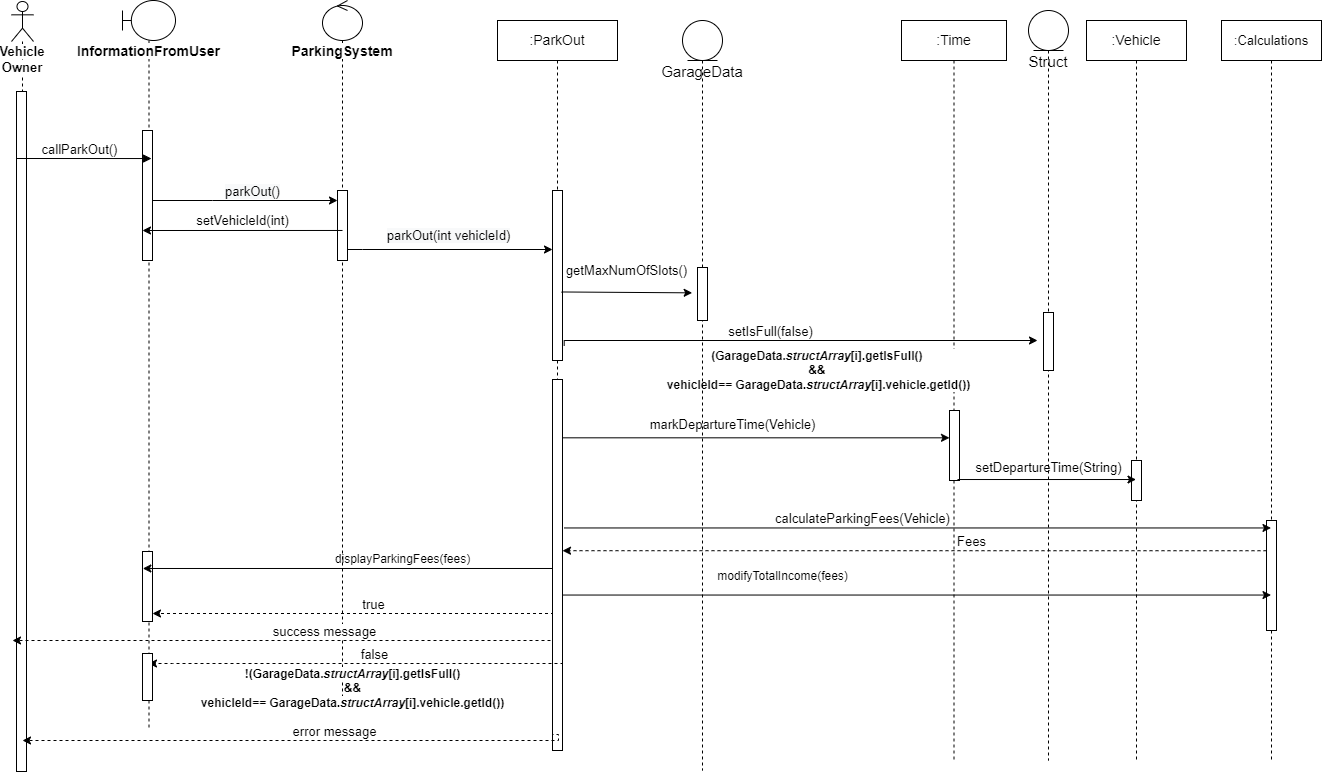
## 

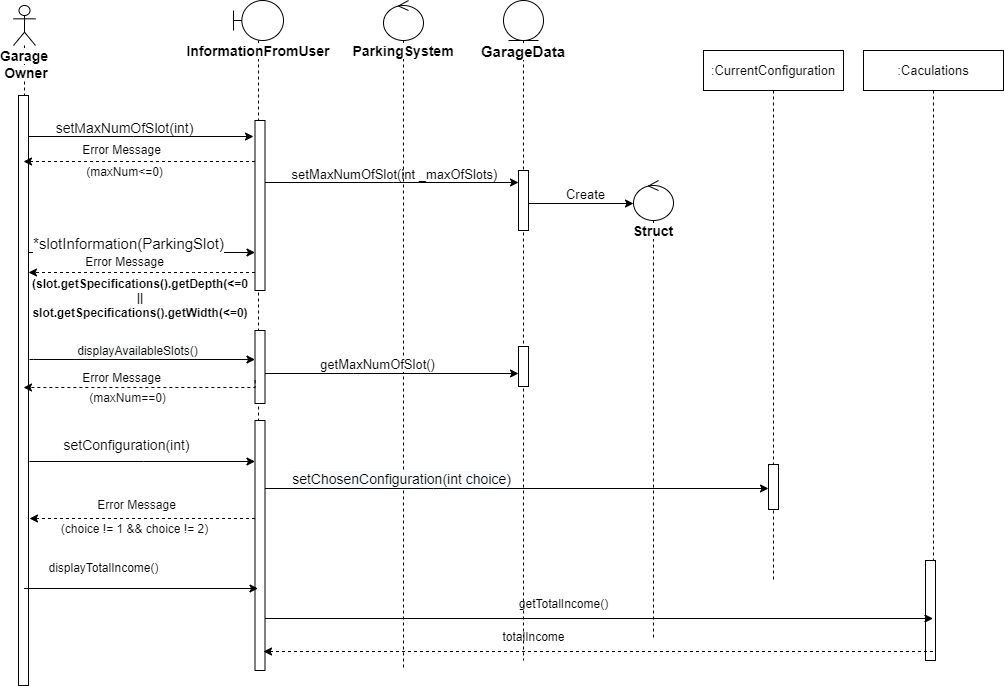
## II. Sequence diagrams

**SQ-01 : Park In Sequence Diagram (Note: chosen configuration in the sequence diagram is First Come First Served Approach)**

**SQ-02 : Park Out Sequence Diagram**

**SQ-03 : Calculate Total Income Sequence Diagram**



**SQ-04 : Set The Garage Sequence Diagram**

### Class - Sequence Usage Table

| **Class Name** | **Sequence Diagrams** | **Overall used methods** |
| --- | --- | --- |
| InformationFromUser | SQ-01, SQ-02, SQ-03, SQ-04 | vehicleInformation(Vehicle),  callParkIn(),  callParkOut(),  setMaxNumOfSlots(int),  displayParkingFees(double, int),  displayNumberOfVehicles(),  displayTotalIncome(),  slotInformation(ParkingSlot),  displayAvailableSlots(),  setConfiguration(Iint) |
| ParkingSystem | SQ-01, SQ-02, SQ-03, SQ-04 | parkOut(),  parkIn(ParkIn),  setAlgorithm() |
| ParkOut | SQ-02 | parkOut(int) |
| Struct | SQ-01, SQ-02, SQ-03, SQ-04 | setIsFull() |
| Time | SQ-01, SQ-02, SQ-03 | markDepartureTime(Vehicle),  markArrivalTime(Vehicle) |
| Vehicle | SQ-01, SQ-02, SQ-03 | setDepartureTime(string),  setArrivalTime(string),  counterPlus() |
| Calculations | SQ-02, SQ-03 | calculateParkingFees(Vehicle),  modifyTotalIncome(double), |
| ParkIn | SQ-01 | <<interface>> |
| CurrentConfiguration | SQ-01, SQ-04 | currentConfigruation()  setChosenConfiguration(int) |
| FirstComeFirstServed | SQ-01 | parkInSearch(Vehicle) |
| BestFit | SQ-01 | parkInSearch(Vehicle) |
| GarageData | SQ-04 | getMaxNumOfSlot()  setMaxNumOfSlot(int) |