**Object Oriented Methodology (OOM)**

**Third Semester July - Dec 2020**

**Mini-Project**

**Smart Parking System**

**Group ID: 3**

**Group Members**

|  |  |
| --- | --- |
| **Name** | **Enrollment No** |
| Animesh Choudhury | IIT2019143 |
| Janit Lodha | IIT2019134 |
| Prince Kumar | IIT2019130 |
| Tamoghno Bhattacharya | IIT2019103 |
| Souvik Patra | IIT2019136 |

**Project Report**

**INTRODUCTION**

In this project we have created a GUI enabled for a Smart Parking System. The system is totally efficient to handle the registration and withdrawal of any number of vehicles in the parking slots. The administrator only needs to configure the parking system. Also, we have implemented an email based alert system to notify the users in case of expiration of maximum time of parking. The user will be able to either update his parking duration or withdraw on expiration of his parking time.

**WORKING**



Fig: Home Frame

This screen is the starting screen or home screen for our project. After this screen we will be redirected to the next screen i.e. ConfigureNew screen.

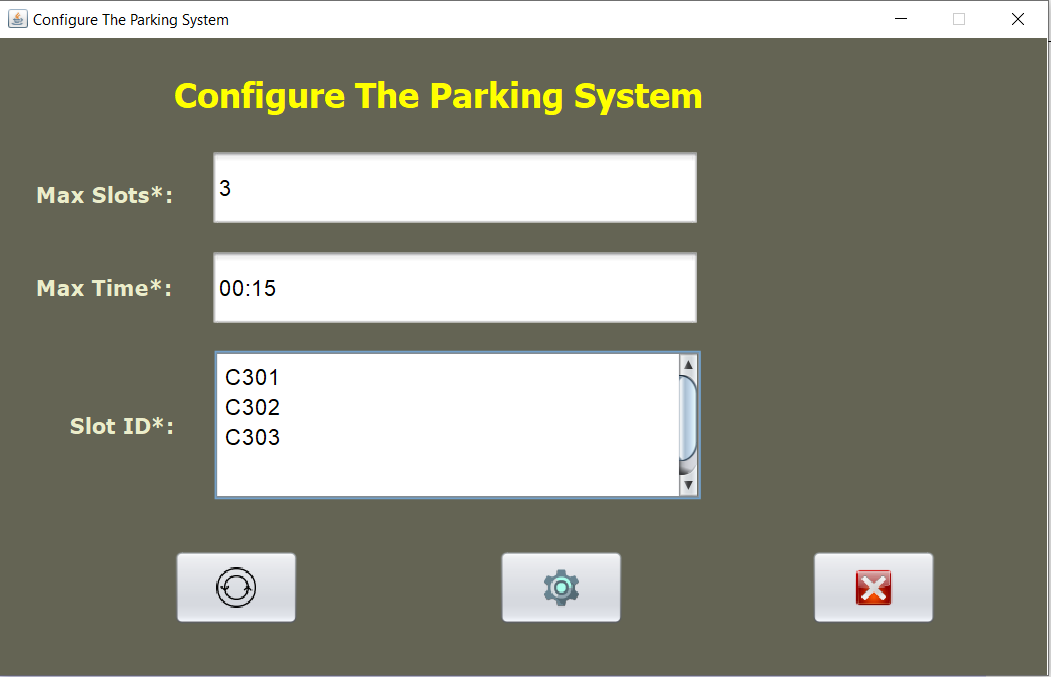


Fig: ConfigureNew Frame

In this screen the administrator needs to configure the parking system i.e. set up max slots, maximum time of parking and slot ids. On configuring the parking system, the project will redirect to the next screen or the Display Frame.

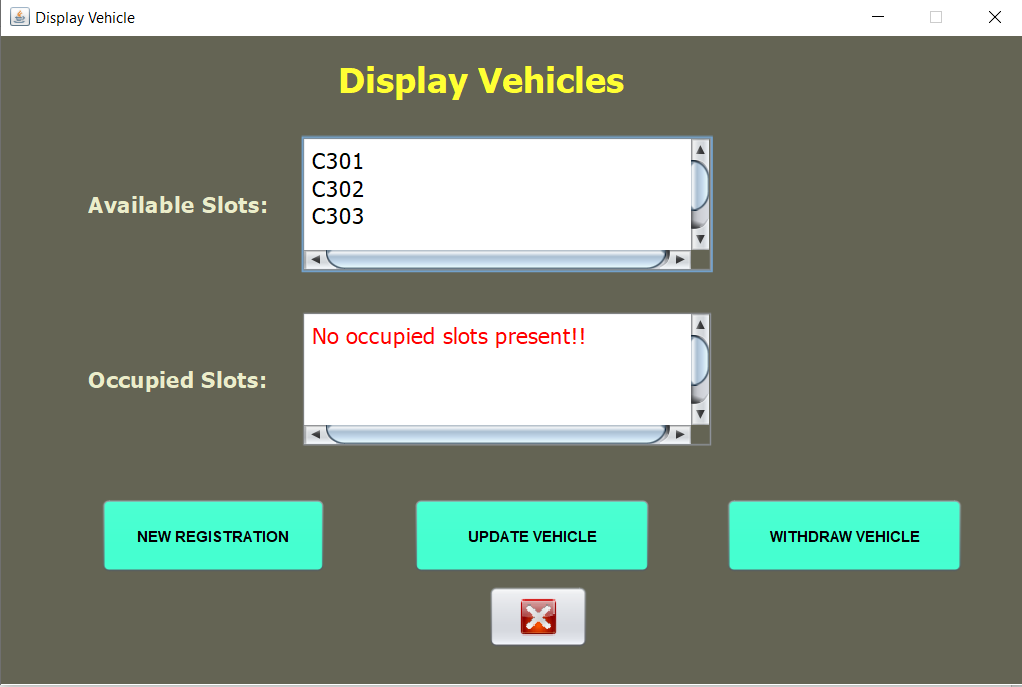


Fig: Display Frame

This is the screen that will be visible to the user. The user has 3 choices: New Registration, Update Vehicle or Withdraw Vehicle. On choosing the New Registration he will be redirected to the next screen i.e. Display screen.



Fig: Display Frame

In the Register Frame the user needs to enter all his details i.e. Vehicle ID, Name, Contact Number, Email Id. He also needs to choose a parking slot from the list of available parking slots. On submitting the details his vehicle will be successfully registered and he will receive an email of successful registration.

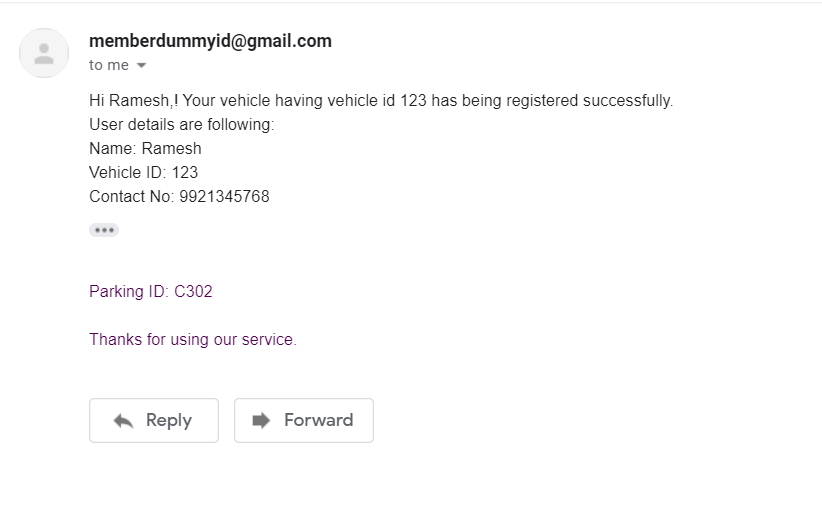


Fig: Successful registration Email

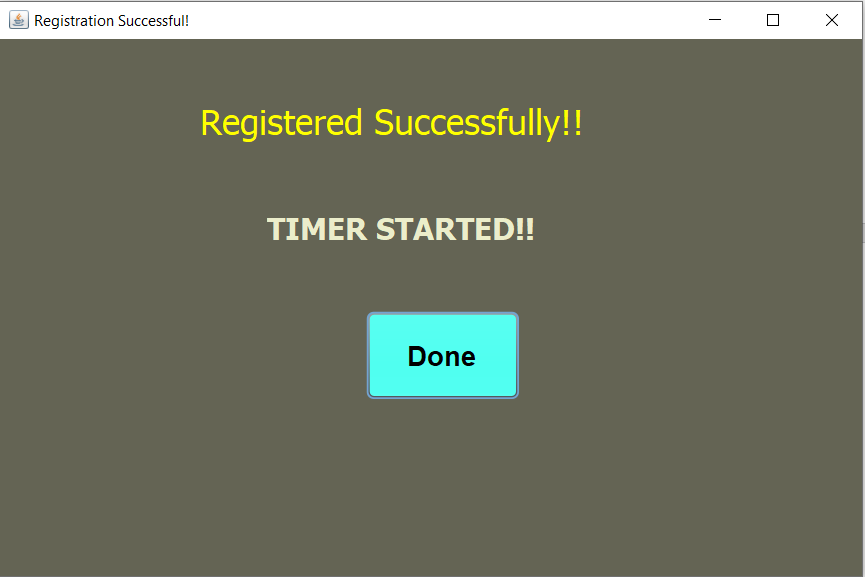


Fig: EmailSystem Frame

The user will then be redirected to the EmailSystem Frame which will start a Countdown Timer until his expiration of max time. After the expiration of his maximum time he will receive a Parking Time Expired Email.

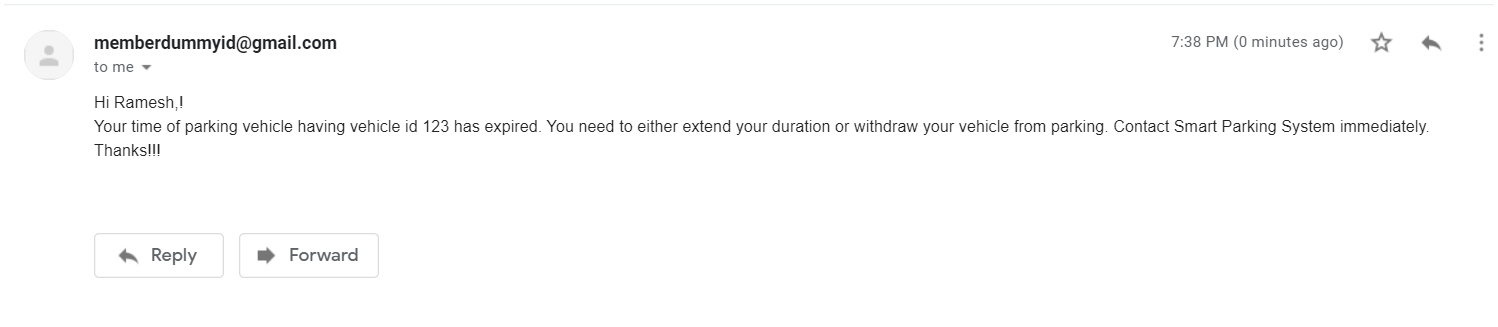


Fig: Parking Time Expired Email

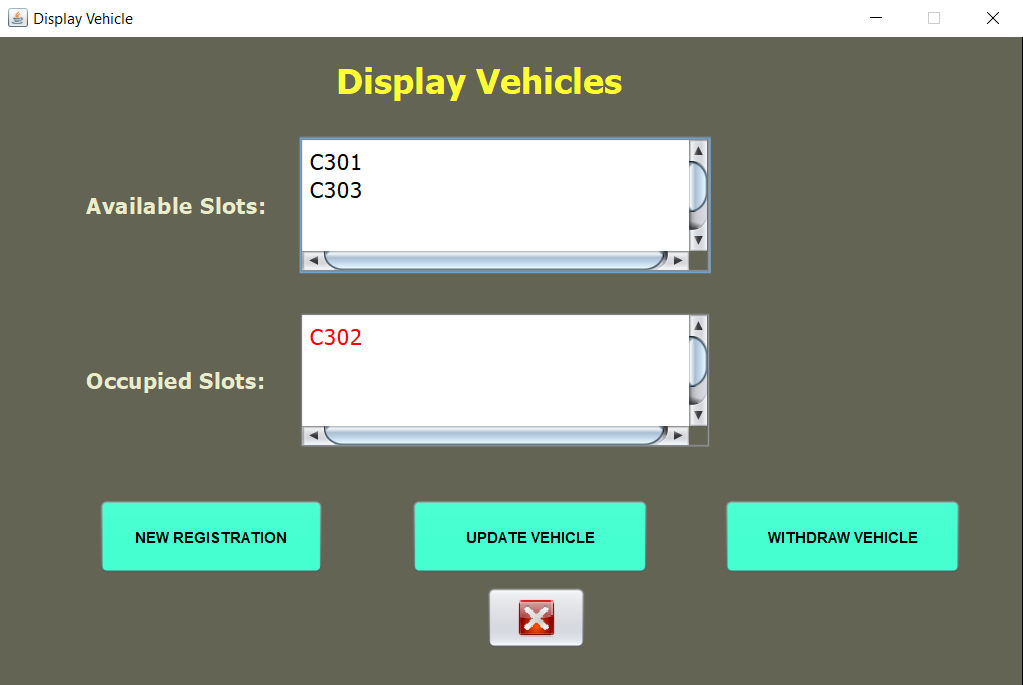


Fig: Successful Registration of the Vehicle

The above picture shows that the vehicle has been successfully registered a now that parking slot is moved to occupied slot.

Now if the user wants to update his parking time, he can click on the Update Vehicle button and he will be redirected to the Update Screen.



Fig: Update Frame

In the Update Screen the user needs to enter his Vehicle Id and the Parking Id to stop unauthorized persons from accessing the System. On entering valid input, the user’s parking slot will be updated and his slot will be again valid for the maximum time slot. He will again receive email at the expiration of his time slot and this cycle continues for the number of times he updates his parking slot.

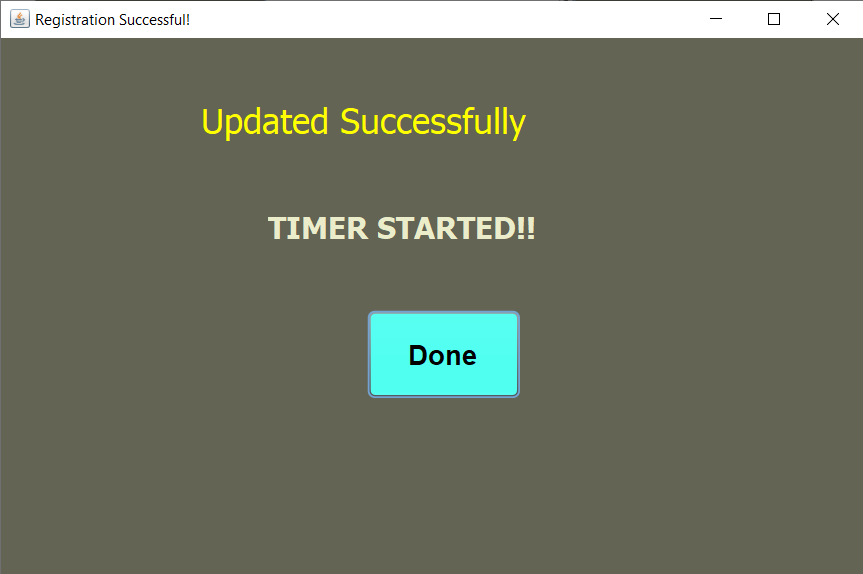


Fig: Successful Update

Now if the User wants to withdraw his vehicle, he needs to click on the Withdraw Vehicle button in the Display Frame. He will then be redirected to the Withdraw screen.

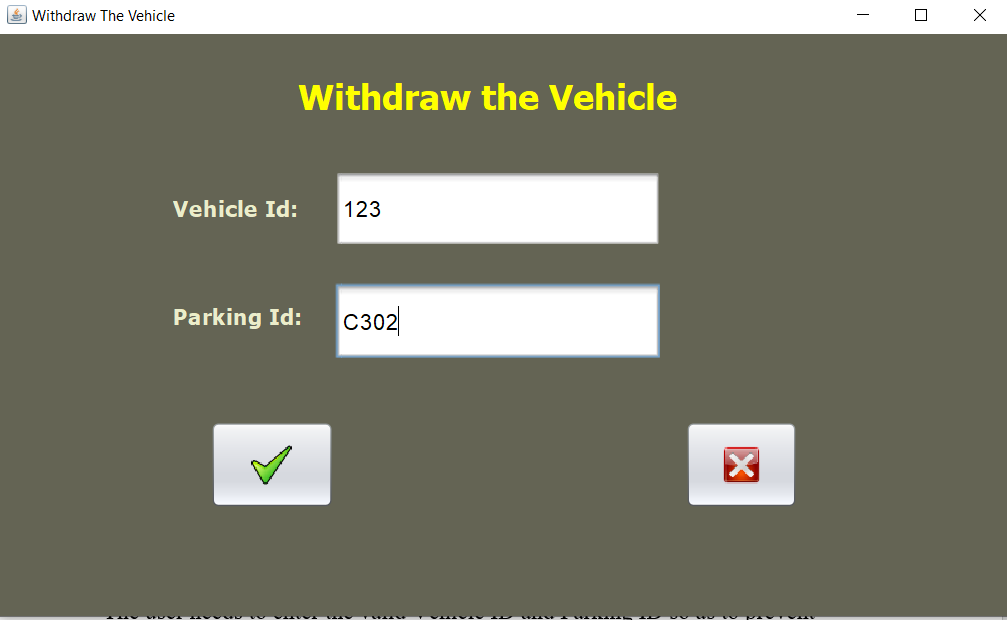


Fig: Withdraw Frame

The user needs to enter the valid Vehicle ID and Parking ID so as to prevent unauthorized access to the system. If the input is valid his vehicle will be withdrawn and he will get the following message

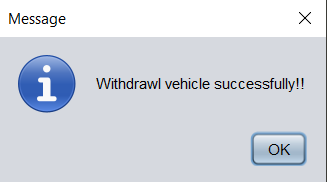


Fig: Successful Withdraw

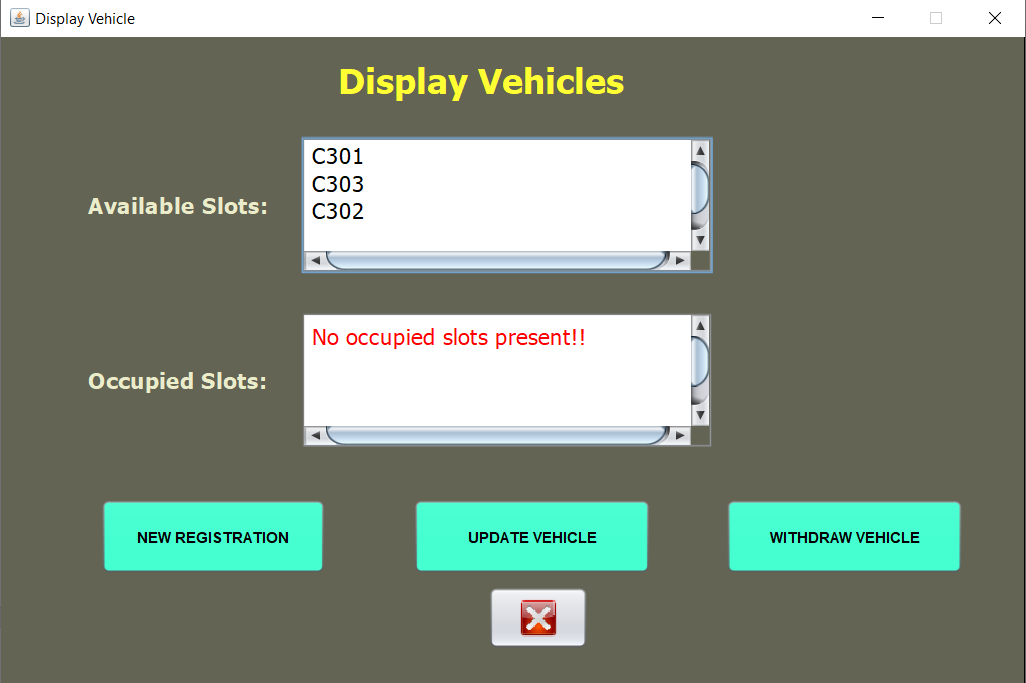
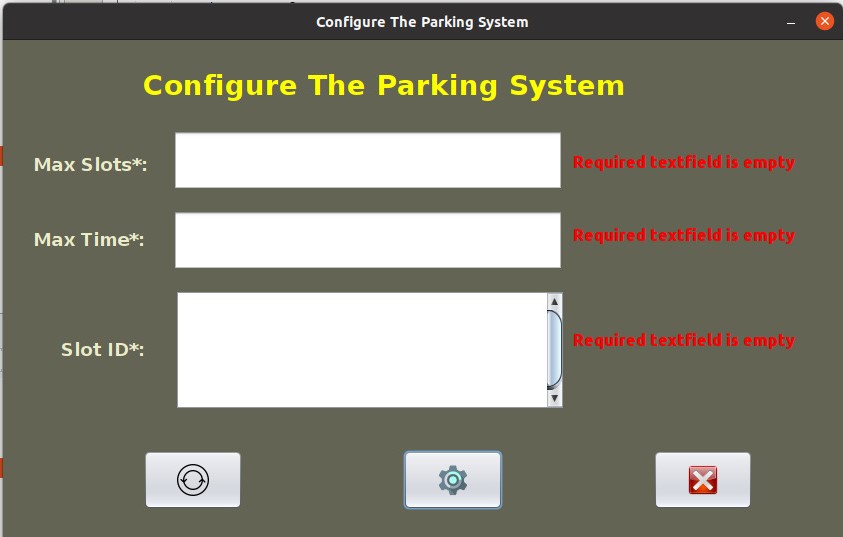


Fig: Display Frame after Successful Withdraw

We see in the above picture that the vehicle has been successfully withdrawn and the slot has been moved to available slot.

**Validations:**For the proper functioning of the project and to ensure that USER or ADMIN can’t enter any invalid data, we have put various validations.  
  
**ConfigureNew Frame:**

* All the three fields are mandatory.
* Maximum number of slots must be an integer.
* Max time should be in the format MM:SS (minutes:second).
* Number of slot ids entered must not exceed the “**max slot”** id and all of them should be pairwise distinct.

  
Fig: ConfigureNew frame- Error showing required fields are empty

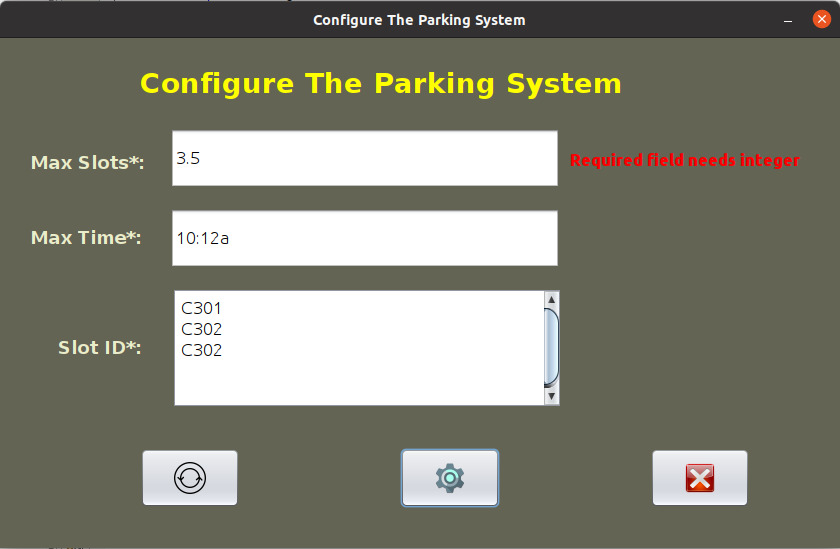


Fig: ConfigureNew frame- Error showing Max Slots must be an integer.

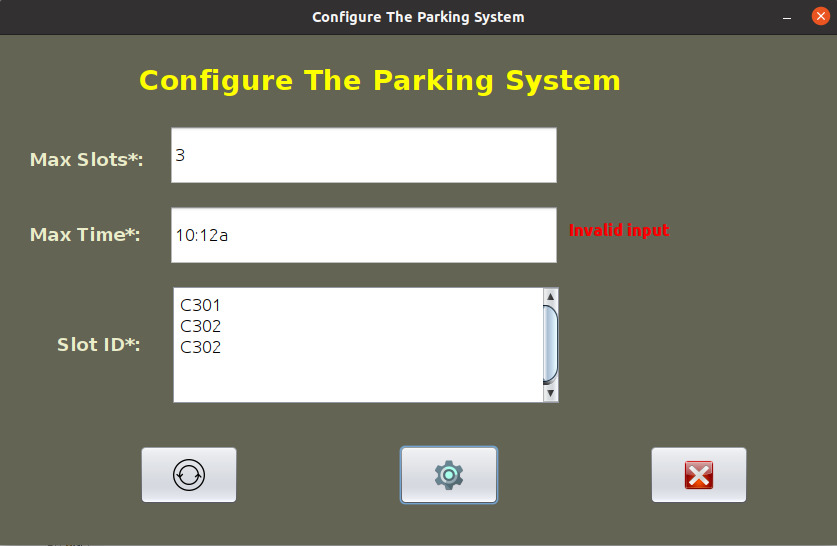


Fig: ConfigureNew frame- Error showing INVALID INPUT,

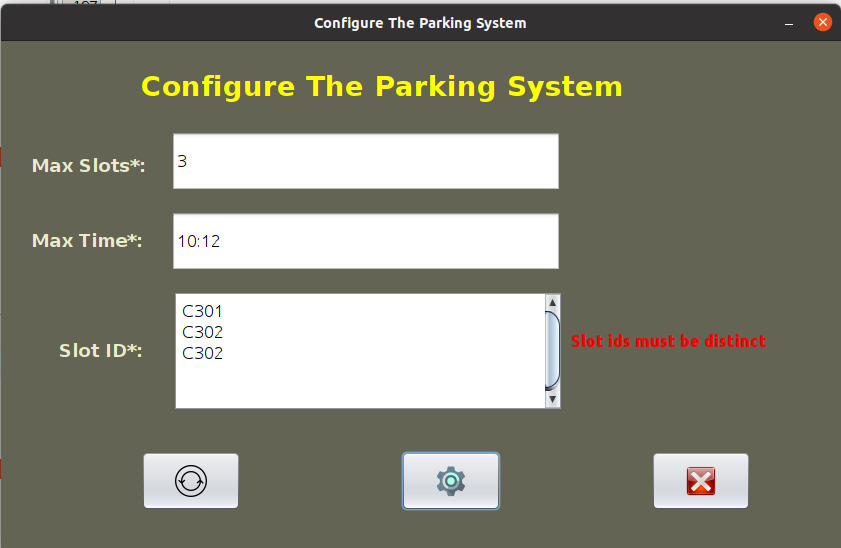


Fig: ConfigureNew frame- Error showing “Slot ids must be distinct”.



Fig: ConfigureNew Frame- Error message when number of slot ids exceed max available slots.

**Registration Frame:**

* All the five fields are mandatory.
* User’s contact number must be of 10 digits containing digits (0-9) only. We assumed that this would be a normal Indian phone user and hence discarded country code.



Fig: Register Frame- Error showing required fields are empty.



Fig: Register Frame- Error showing invalid phone number.

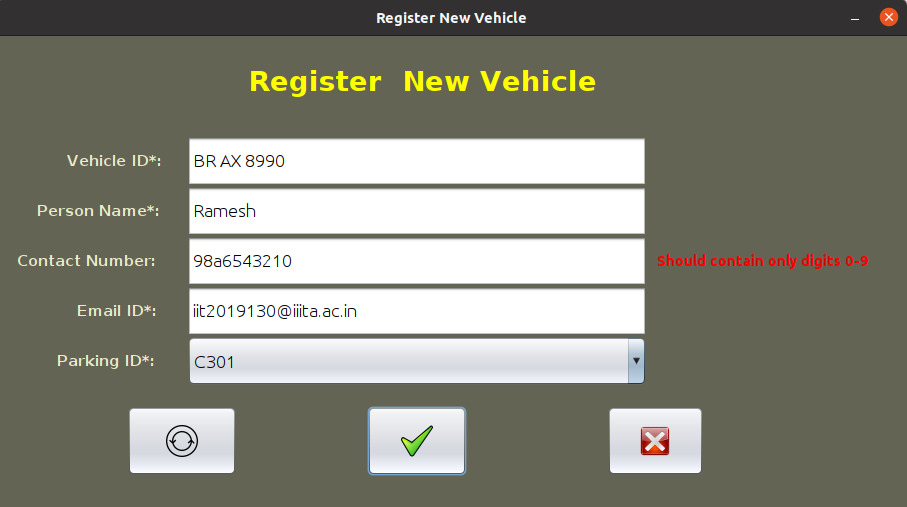


Fig: Register Frame- Error showing invalid phone number.

**Update Frame:**

* Both the fields Vehicle ID and Parking ID are mandatory.
* For authorized access, Vehicle ID and Parking ID should match with the earlier Vehicle ID given by USER and Parking ID allotted to USER respectively.



Fig: Update Frame- Error showing required fields are empty.



Fig: Update frame- Error showing Unauthorized access on invalid data input.

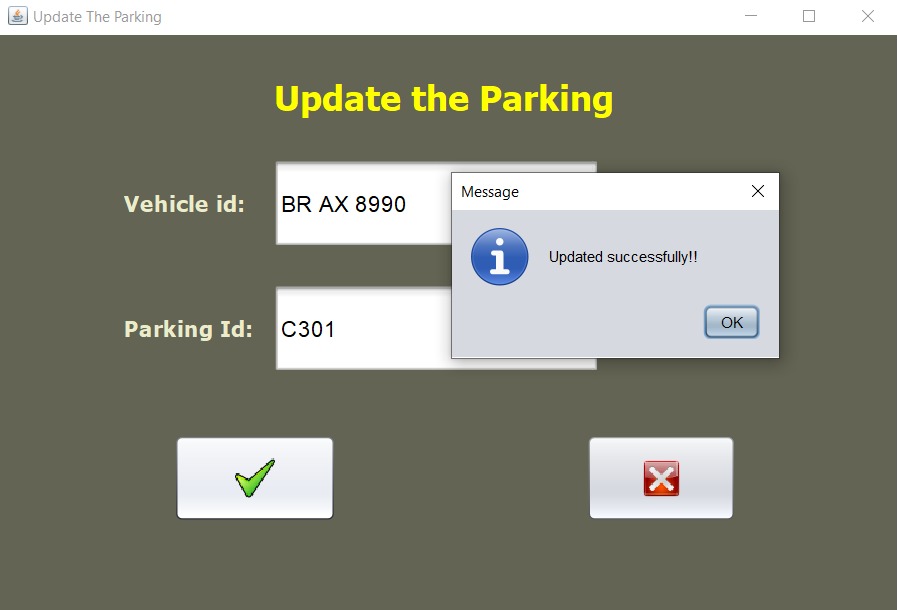


Fig: Update frame- Updated Successfully on valid data input.

**Withdraw Frame:**

* Both the fields Vehicle ID and Parking ID are mandatory.
* For authorized access, Vehicle ID and Parking ID should match with the earlier Vehicle ID given by USER and Parking ID allotted to USER respectively.



Fig: Withdraw Frame- Error showing required fields are empty.

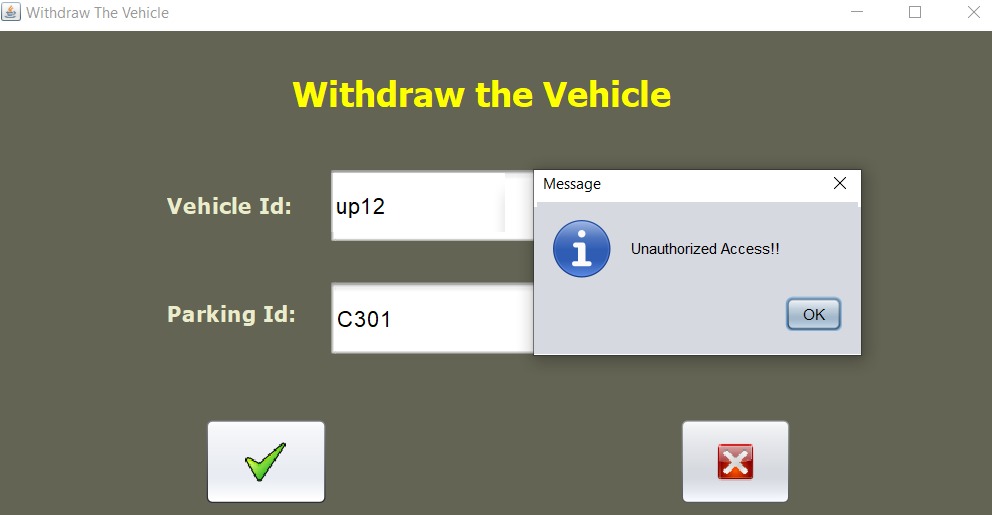


Fig: Withdraw Frame- Error showing Unauthorized access on invalid data input.

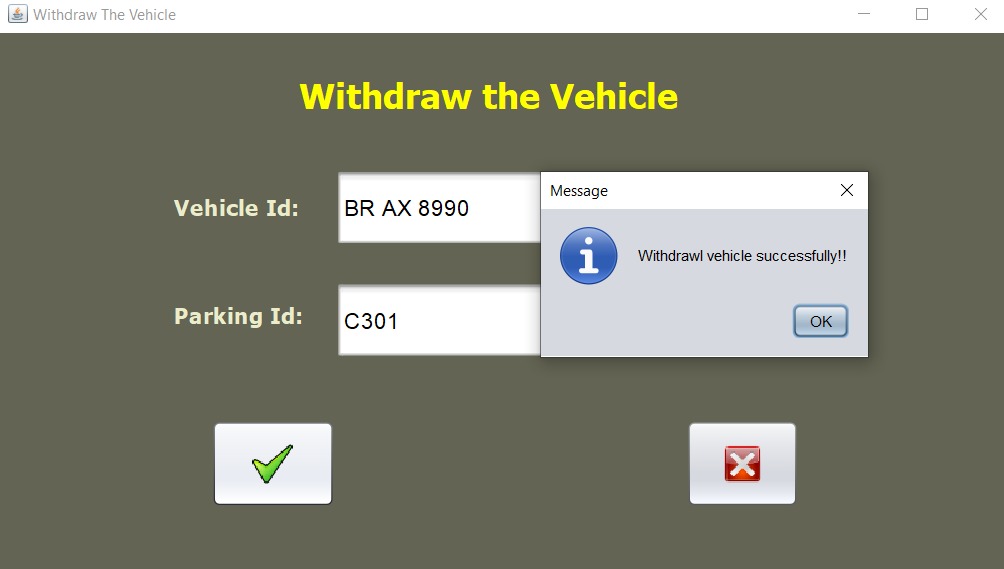
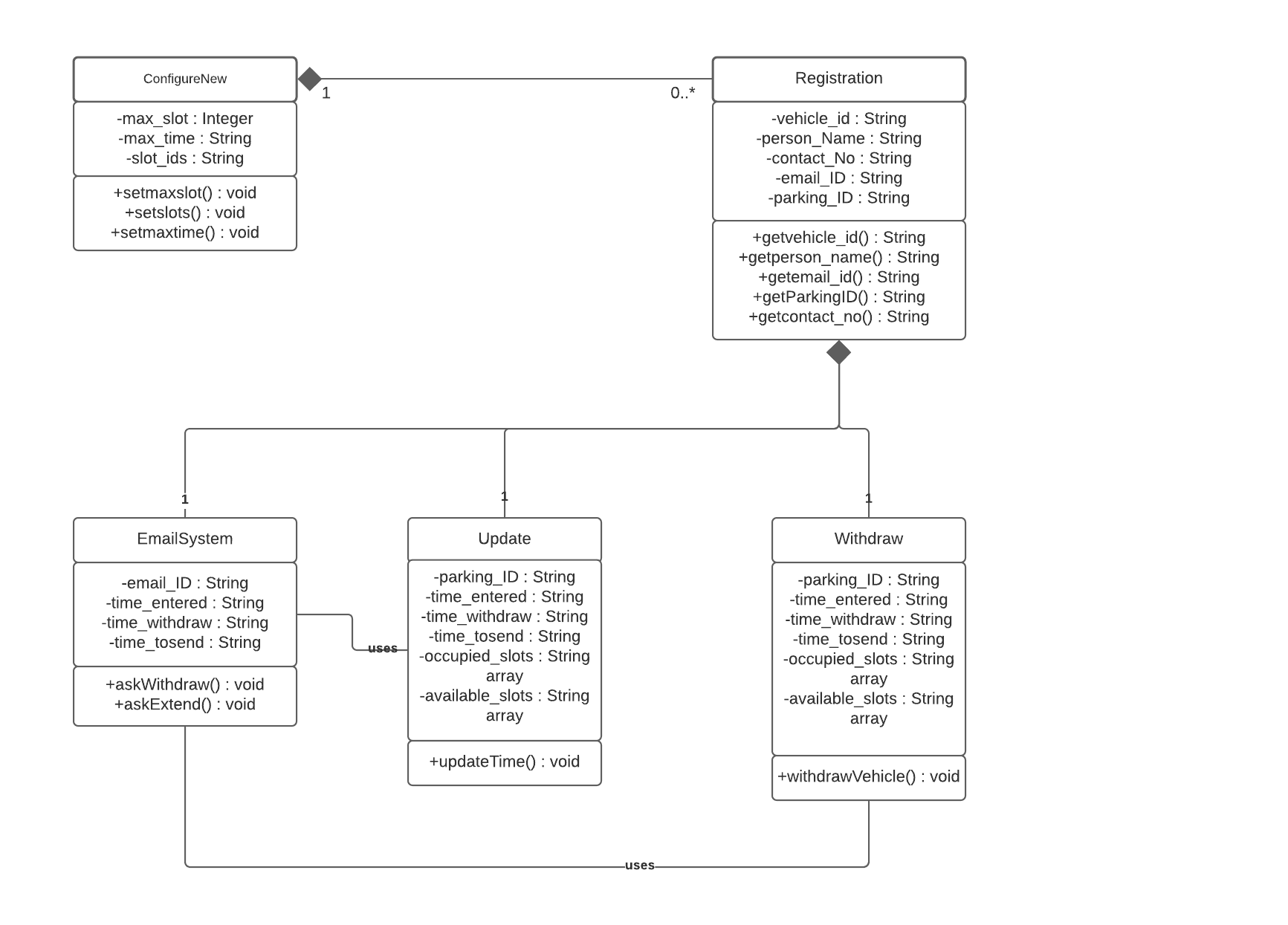
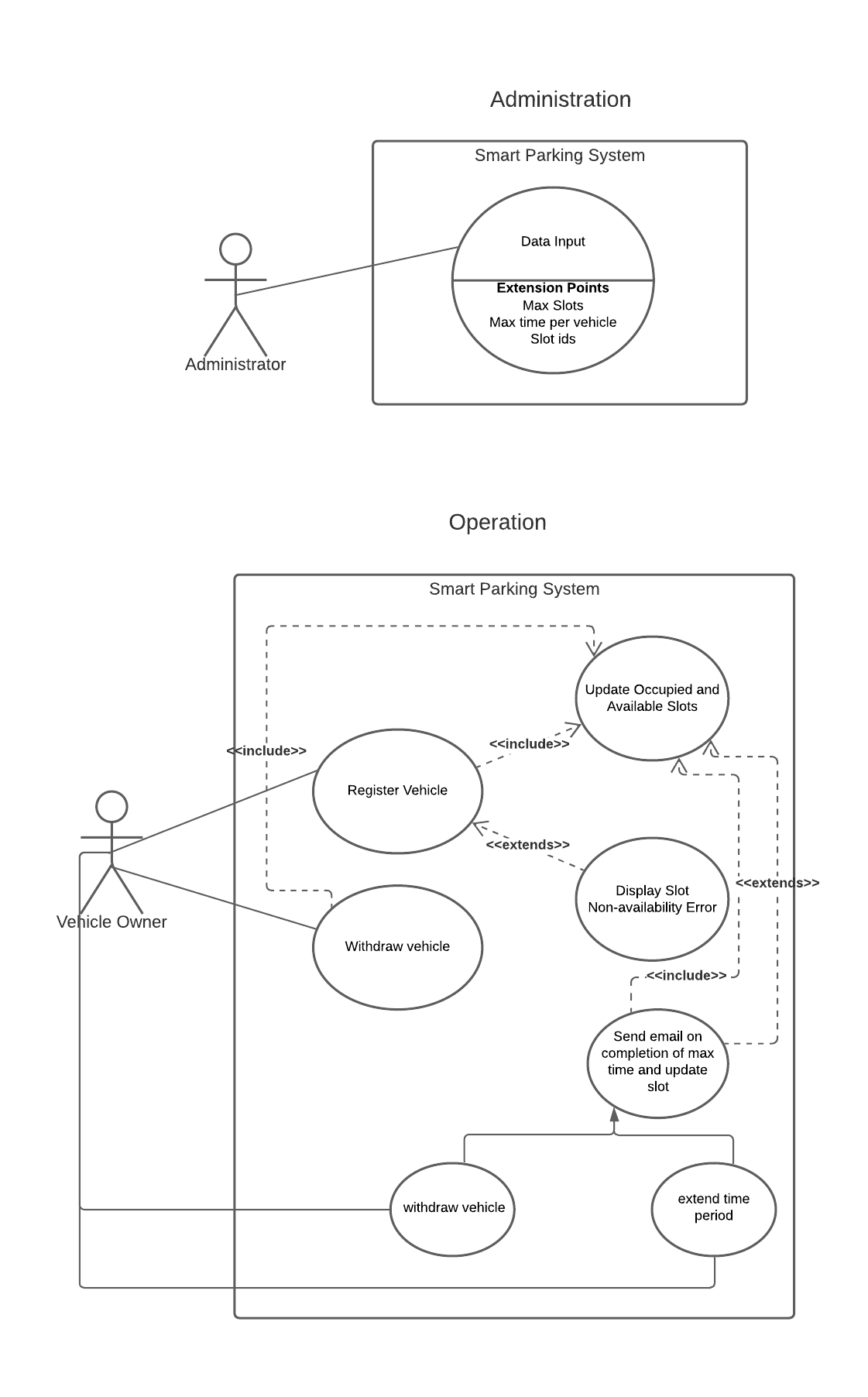


Fig: Update frame- Withdrawn Successfully on valid data on valid data input.

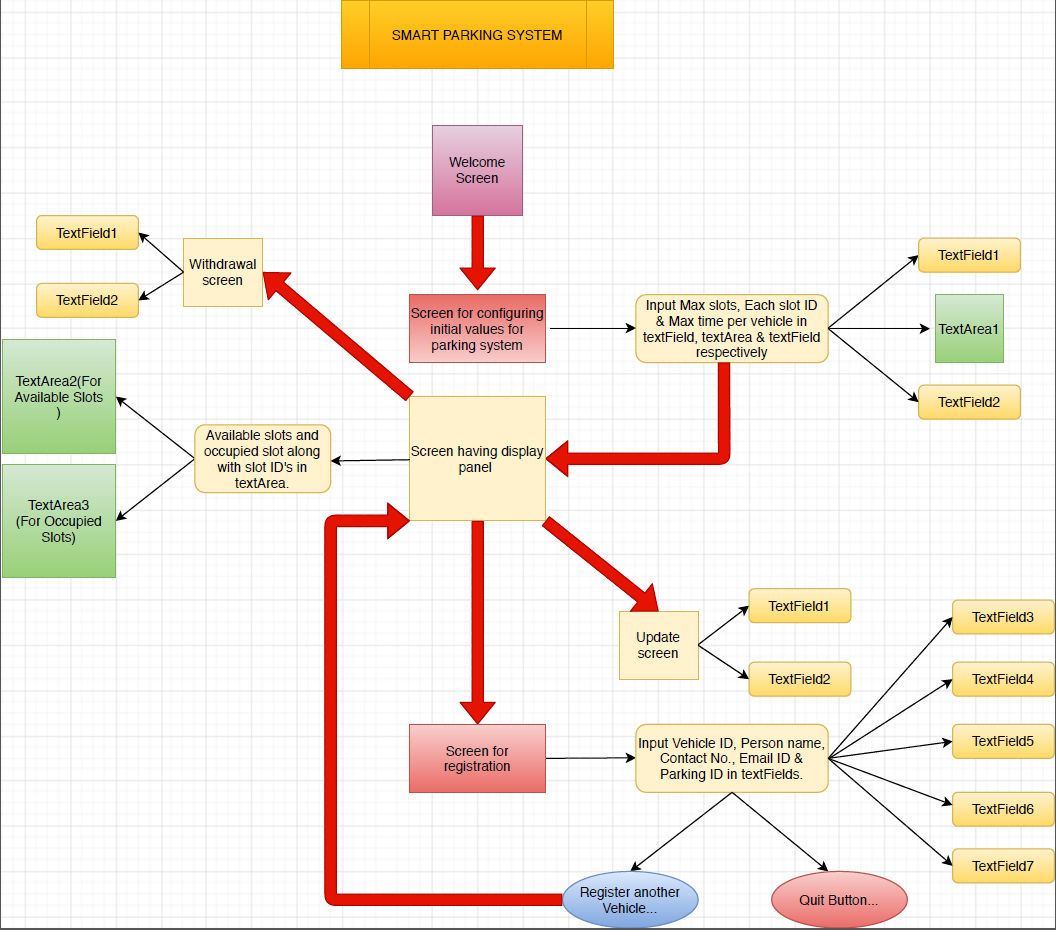
**UML CLASS DIAGRAM**

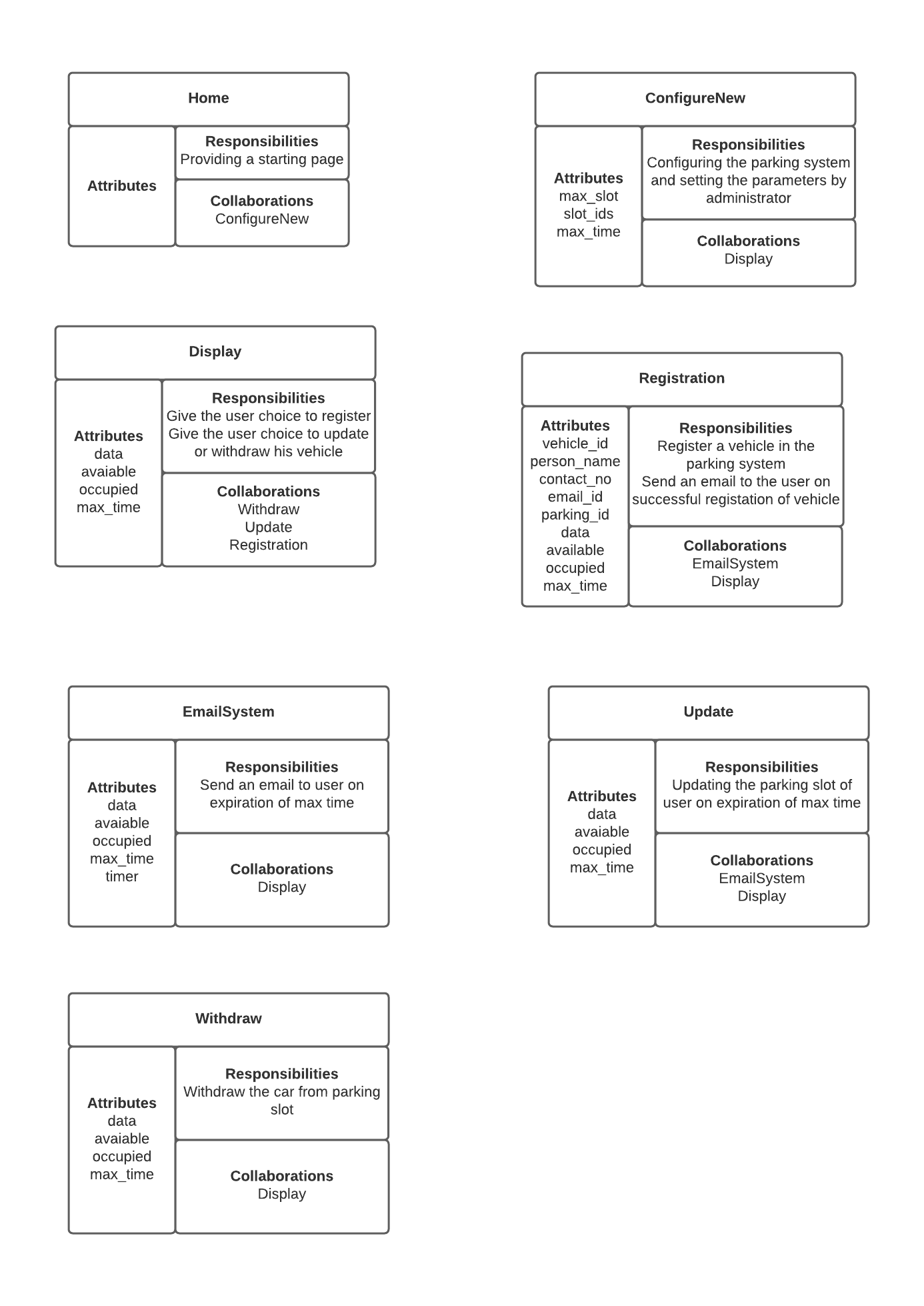


**UML USE CASE DIAGRAM**



**UML BLOCK DIAGRAM**

****

**CRC MODELLING DIAGRAM**

**FEATURES**

* Automated Mailing System so that the user gets automatically notified after his parking time has expired.
* Use of Object-Oriented Modelling properties like encapsulation so that as to prevent unauthorized access into the system.
* User friendly interface so as to provide ease of access to the user.
* Efficient withdraw and update facilities and management of data so as to avoid clash between multiple users.

**FUTURE SCOPE**

In this world of ever-increasing population, it is mandatory to have technology sufficient to meet the needs of the people. One such technology is a smart parking management system. Nowadays parking control is very important for all malls, multiplexes, complexes, hotels as well as big markets. It is necessary to have designated parking spot for every single vehicle. To establish control, many secure devices such as barricades, swing gates, slide gates, control gates, toll gates, time and attendance machine, counting system etc. have been in use to track cars and the fees that are due to be paid by each owner. However, operational overheads have been increasing steadily with the rise of population as vehicles owned by people increase. Speed is another problem since people have to wait with their vehicles in line in many parts of the world, for getting their parking slips. To decrease expenses and waiting times, automation is the way forward in various sophisticated parking lots around the world, with the help of IoT (Internet of Things) technology, which enables us to monitor bigger systems without the use of a lot of expensive hardware as well as manual labor. However, data management is a primary issue here as a lot of complex information is needed to be stored and managed properly to implement such systems. Our smart parking-enabled parking system software aims to handle such complexity issues using Object Oriented Modelling, which can scale to big systems with ease. Speed and efficiency are the two biggest driving forces behind smart technology, and we believe our model fits the software requirements to be deployed with smart IoT-enabled systems with ease, enhancing and speeding up the process of parking, as well as making the entire process much more accessible and user-friendly.