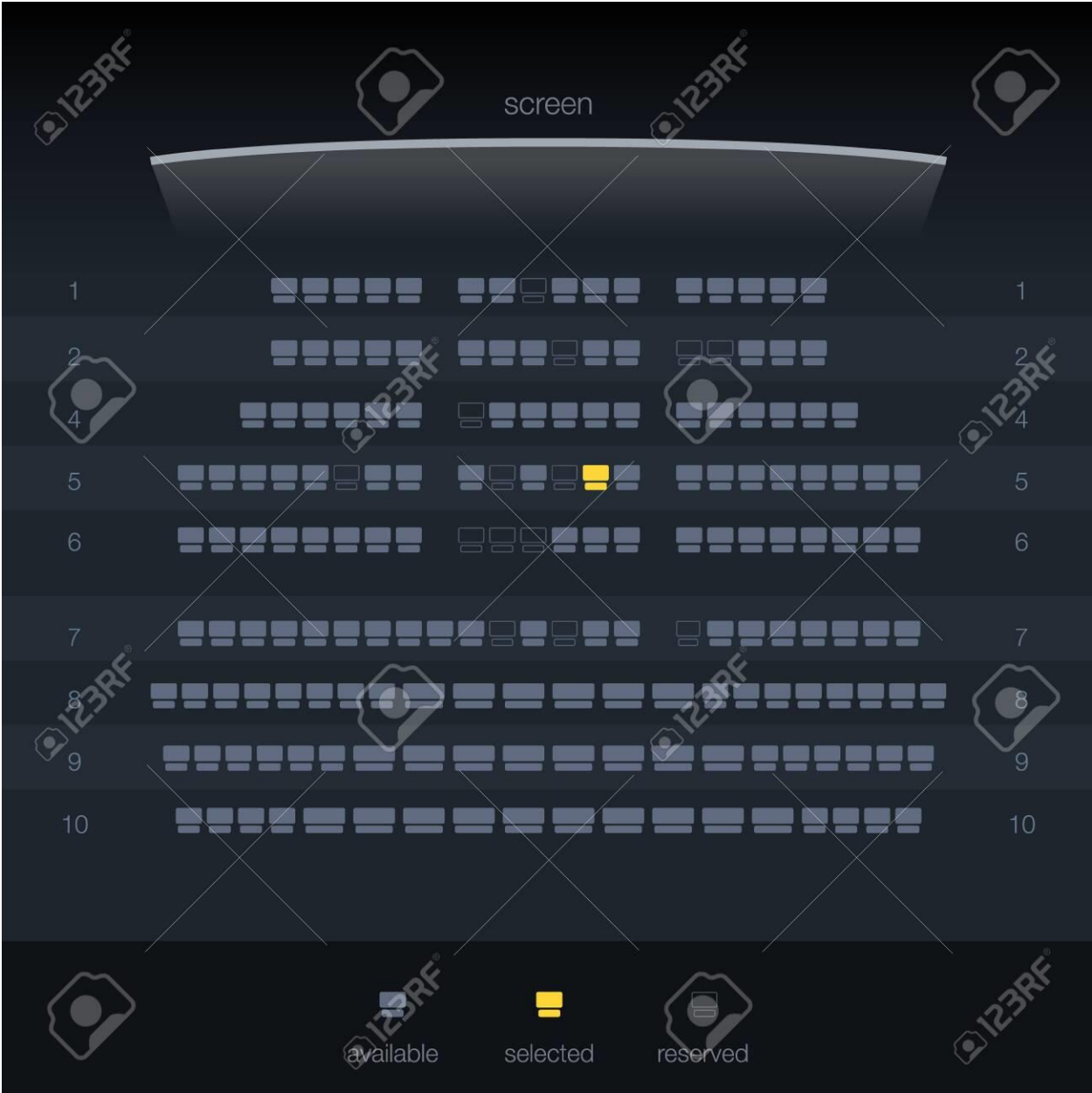


# CINEMA RESERVATION SYSTEM



## Table of Contents

CINEMA RESERVATION SYSTEM .....	1
<i>The First Part Of Our Project</i> .....	3
<b>Statement Of Work</b> .....	3
<b>Requirements List</b> .....	3
<b>Use Case Diagram</b> .....	4
<b>USE Cases</b> .....	5
<b>Textual Analysis</b> .....	7
<b>UML Diagram</b> .....	8
<i>The Second Part Of Our Project</i> .....	8
<b>Classes</b> .....	8
<b>About The Project</b> .....	9
<b>Command-List:</b> .....	9

## *The First Part Of Our Project*

### **Statement Of Work**

In recent years, cinema has taken the place of theater. People now prefer to watch ready-made scenes instead of living stage performances. One of the reasons for this is that movies are given more coverage on social media and appeal to more people than theater performances. After that, people started watching movies in theaters. However, people want to go in a pre-planned way due to their daily intensity. This is where our application comes into play.

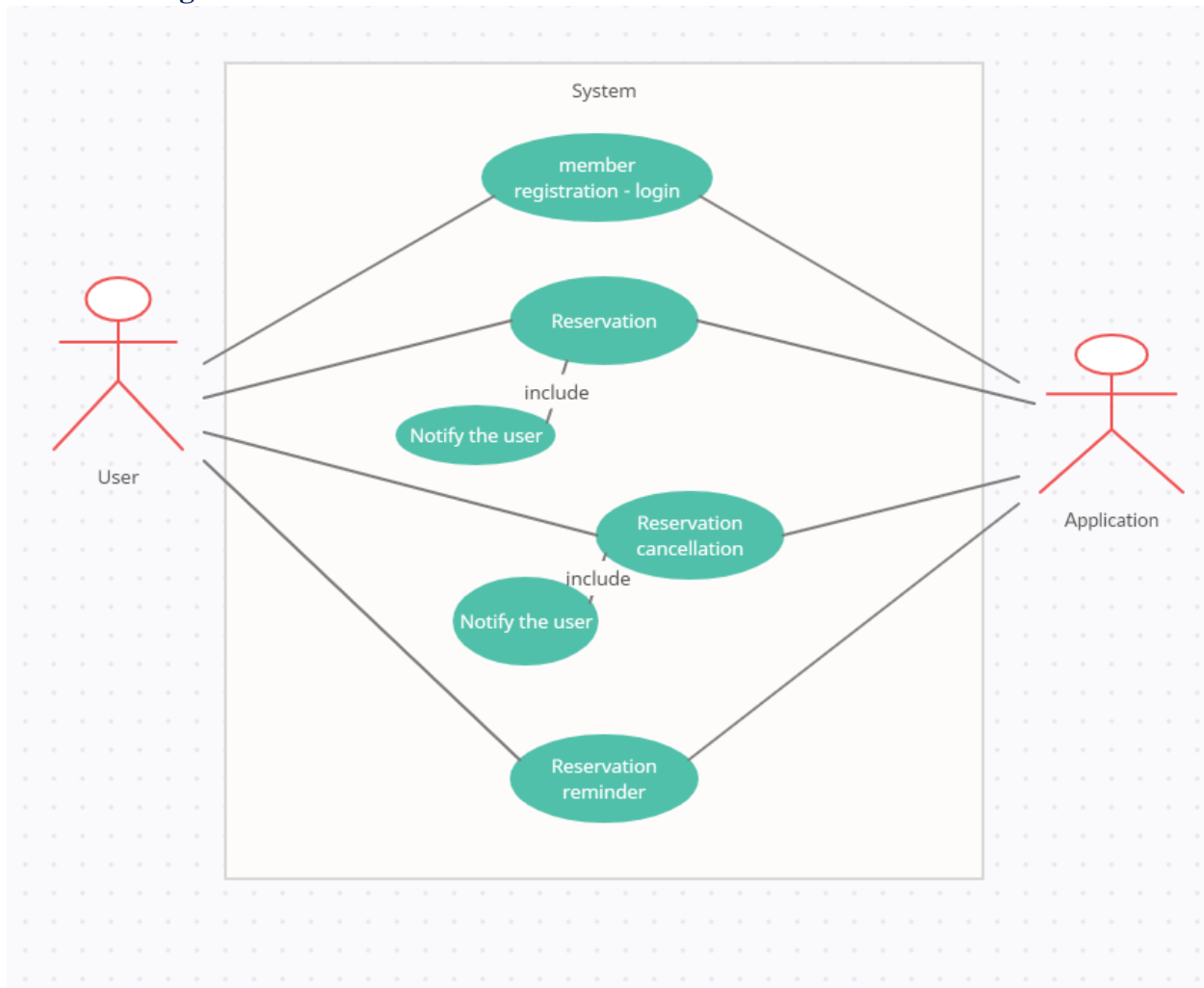
The main reason for making this reservation is that they resort to this method to use their hours more efficiently, as they cannot spare time from the busyness of daily life. The functions of our application allow them to watch the movies they want, on the date and the seat, wherever they want, without waiting in any queue. For this reason, we present the working hours we receive from the workplaces we work within our application at appropriate time intervals that people can choose. We also basically offer them booking and payment methods after registering in our app.

We aim to help people use their time more efficiently in this busy schedule. While time is so valuable, we do not want it to be wasted without our application.

### **Requirements List**

1. The purpose of our application is to make reservations.
2. They must be registered or logged in to be able to make a reservation
3. They can choose the movie they want to make a reservation with.
4. They can choose reservation dates, hours, movie theaters, and seats.
5. In case of an Eligible Reservation, they will receive a confirmation from our application.
6. They can make reservation cancellations.
7. They can see all reservations.
8. They can use our application while making payment transactions.
9. Reservation information is sent to the user.
10. Our application uses reminders so that they do not forget their reservations.
11. Our different payment method is gift coupons given to the user.

## Use Case Diagram



## USE Cases

### *Use case 1*

1. The user enters the application	N/A
2. If the user is a member, login or register	2
3. Select movie for the reservation	3
4. Select day, time, and seat for the reservation	4
5. The application gives information that the transactions have taken place.	5
6. Select payment method	8
7. The user has information about the reservation	9
8. The application turns off	N/A

External initiator: User

Start condition: 2

Stop condition: 8

Clear value: User selects a time, date, seat, and movie from the application

### *Use case 2*

1. The user enters the application	N/A
2. If the user is a member, login or register	2
3. See all reservations	7
4. Select movie for the reservation	3
5. Select day, time, and seat for the reservation	4
6. The application gives information that the transactions have taken place.	5
7. Select payment method	8
7.1 The previous payment made is used	N/A
7.1.1 Users can pay the coupon	11
8. The user has information about the reservation	9
9. The application turns off	N/A

External initiator: User

Start condition: 2

Stop condition: 9

Clear value: The user chooses the payment method.

### Use case 3

- |  |     |
|--|-----|
| 1. The user enters the application   | N/A |
| 2. If the user is a member, login or register                                | 2   |
| 3. Select all reservations and cancel one of them                            | 7   |
| 4. The application gives information that the transactions have taken place. | 5   |
| 5. Select payment method   | 8   |
| 5.1 The payment made is returned to the user                                 | N/A |
| 5.1.1 Users requests coupon refund   | 11  |
| 6. The application turns off   | N/A |

External initiator: User

Start condition: 2

Stop condition: 6

Clear value: User cancels the reservation from the application

### Use case 4

- |   |     |
|---|-----|
| 1. The user enters the application                  | N/A |
| 2. If the user is a member, login or register       | 2   |
| 3. Uses reservation reminder                        | 9   |
| 4. Sending a text to user to know about reservation | N/A |
| 5. The application turns off                        | N/A |

External initiator: User

Start condition: 2

Stop condition: 5

Clear value: User uses reservation reminder from the application

## Textual Analysis

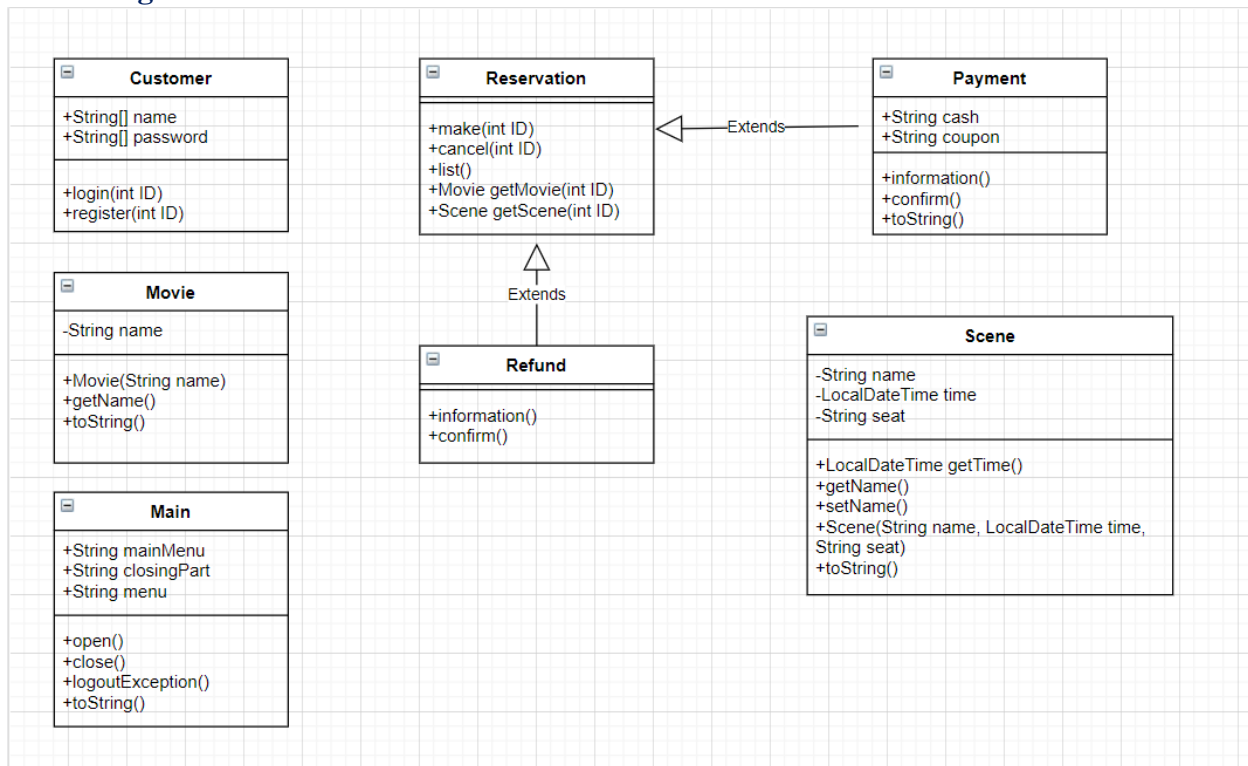
### Noun Analysis

- \*reservation
- \*Information
- \*User
- \*time
- \*date
- \*movie
- \*payment
- \*application
- \*reminder
- \*member
- \*transaction
- \*seat
- \*application
- \*e-mail
- \*coupon

### Verb Analysis

- \*change
- \*cancel
- \*login
- \*register
- \*select
- \*give
- \*turn
- \*enter
- \*has
- \*take
- \*forget
- \*return
- \*send

## UML Diagram



## The Second Part Of Our Project

### Classes

**Customer Class:** In the customer class login method, we put the names and passwords in an array list, then we print them, and if a new name and password are required, we print them in the register class.

**Movie Class:** Defines the names of movies in the movie class.

**Scene Class:** Stage class stage name defines seat and time on stage.

**Reservation Class:** In the reservation class, we first list our models' movies and scenes in a series list, and then we use them for all reservations, then the user can choose one of the movies and scenes if they want to the reservation. You can also choose from the same series list in the reservation cancellation method.



*Payment Class:* Cash payment or coupon payment is defined to the user in the payment class

*Refund Class:* Outputs a refund to the user in the refund class.

*Main Class:* In the main class, we allow the user to make changes to the choices they want to make. In this class, we show the output of the menu, the output of the opening method, the output of the closing method.

## **About The Project**

### *Static changes that can be made to the project*

Comment lines are specified in parts of the code that can be changed by the user in the Main class.

### *Test Class*

The test class made was done according to the use cases we wrote in the project.

### *Package*

Package names are written to facilitate changes in the project.

### *Library*

We used the java.util.ArrayList library in the project Also used java.time.LocalDateTime library.

## **Command-List:**

```
mvn clean package
```

```
java -jar target\bim209hw.jar
```