Andy Luo
#6223602

CONCERT SYSTEM

Project presented to
Yi Wang

420-SF2-RE DATA STRUCTURES AND OBJECT ORIENTED PROGRAMMING

TABLE OF CONTENTS:

Project description :

SCENARIO

This concert system allows for users to search for concerts, book tickets, and manage their reservations. Admins can manage concerts by adding them or removing them, they can also view bookings and handle cancellations. This system will support different concert types and save data by text tiles.

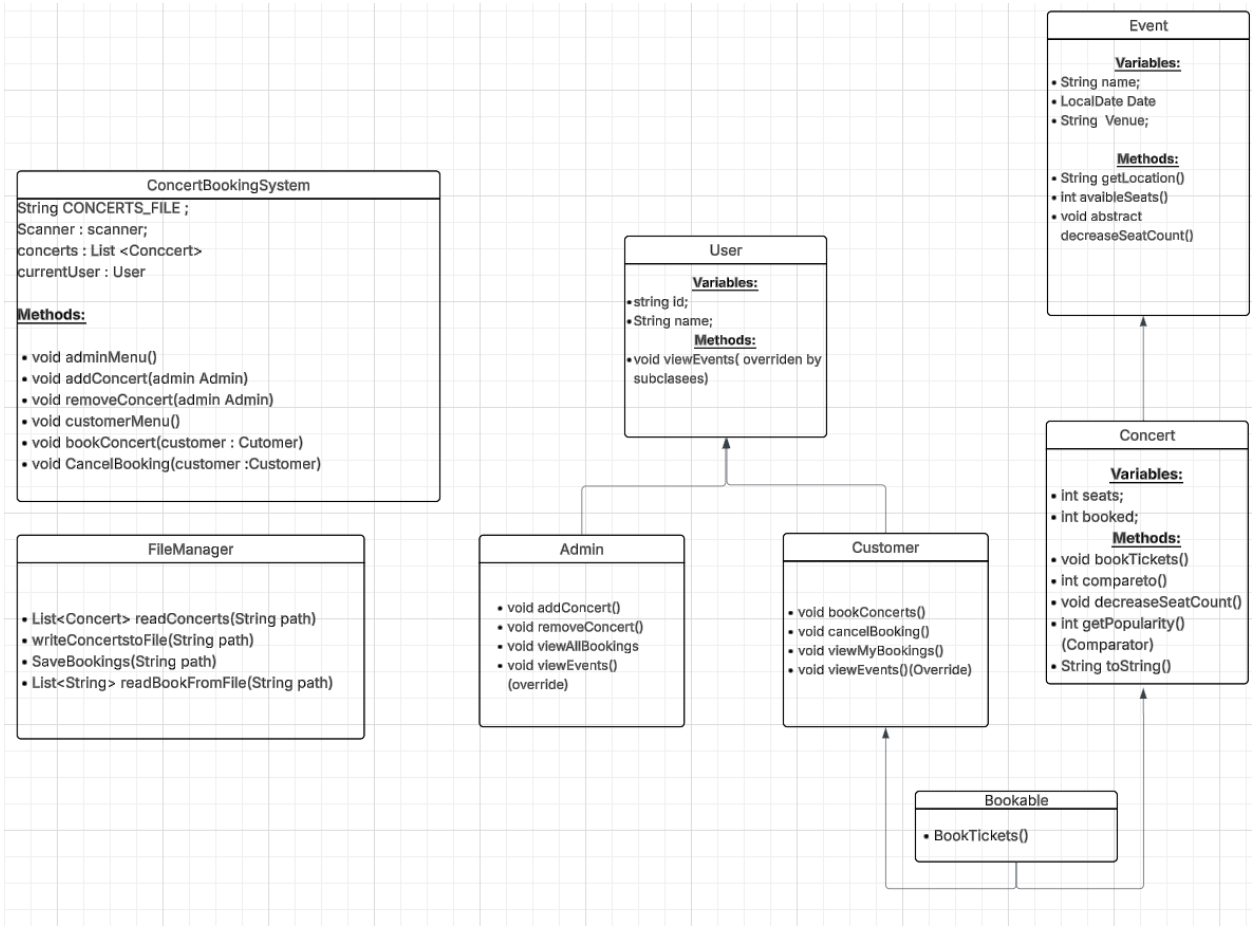
DESIGN PARADIGM

- Admin can add new concerts or remove them. They can also view bookings.
- Users can view available concerts, book tickets, and cancel their booking
- Uses file I/O, exception handling, Streams.

CLASS HIERARCHIES

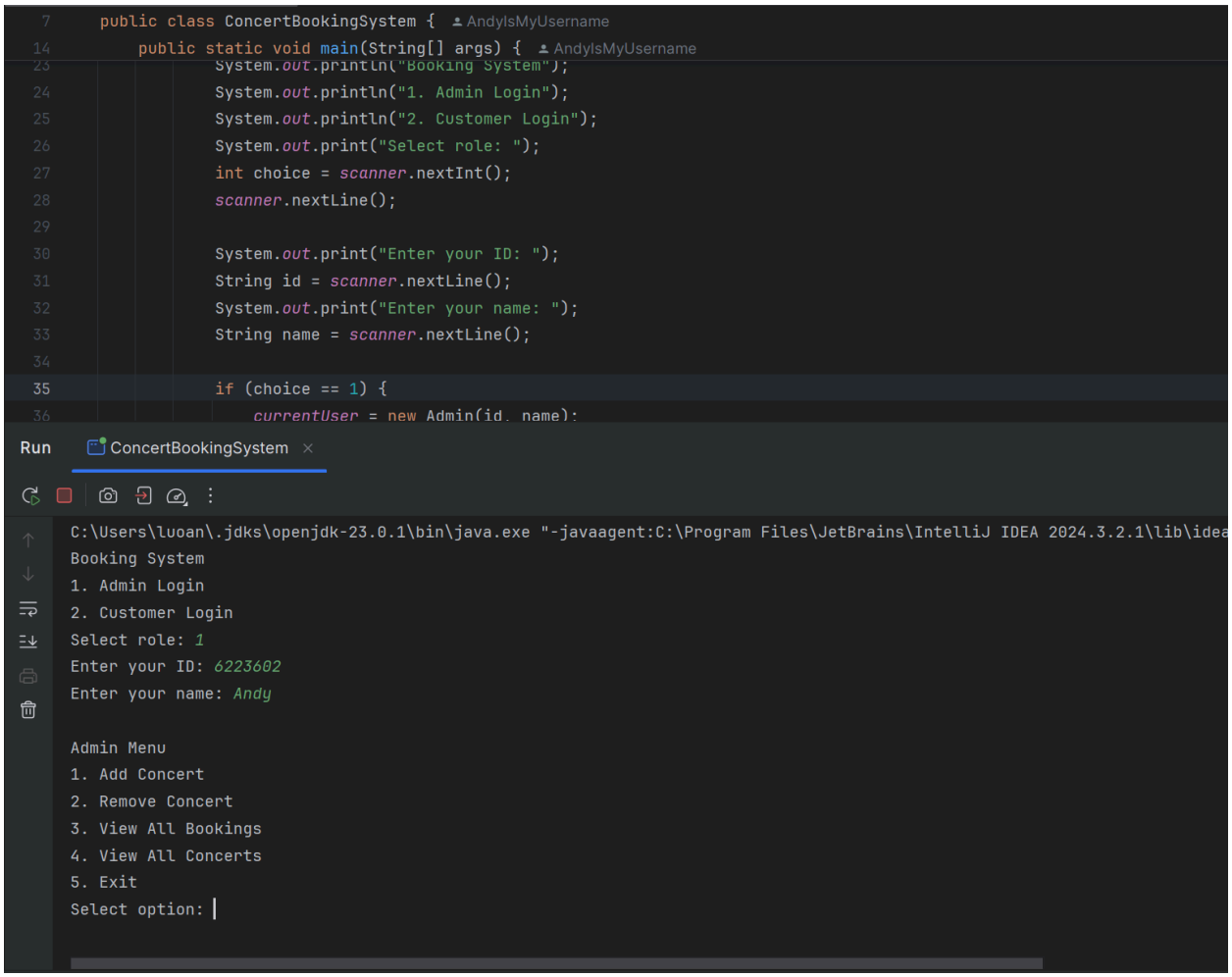
1. User (superclass) -> Admin, Customer
2. Event(Superclass) -> Concert

UML DIAGRAM (new version)



Program features & Screenshots :

Admin Menu : asks user to select between admin or customer then it asks for id and name.



ADD A CONCERT

In the admin menu, add a concert which asks for name, date, venue and number of seats.It then breaks, letting the user go back to the admin menu.

```
Admin Menu
1. Add Concert
2. Remove Concert
3. View All Bookings
4. View All Concerts
5. Exit
Select option: 1
Enter concert name :
Concert`
Enter date (yyyy-mm-dd) :
2006-07-08
Enter venue :
MontrealStadium
Number of seats :
100
Concert Added.

Admin Menu
1. Add Concert
2. Remove Concert
3. View All Bookings
4. View All Concerts
5. Exit
```

It writes the concert in the file.

REMOVE A CONCERT: Let’s user remove a concert from system

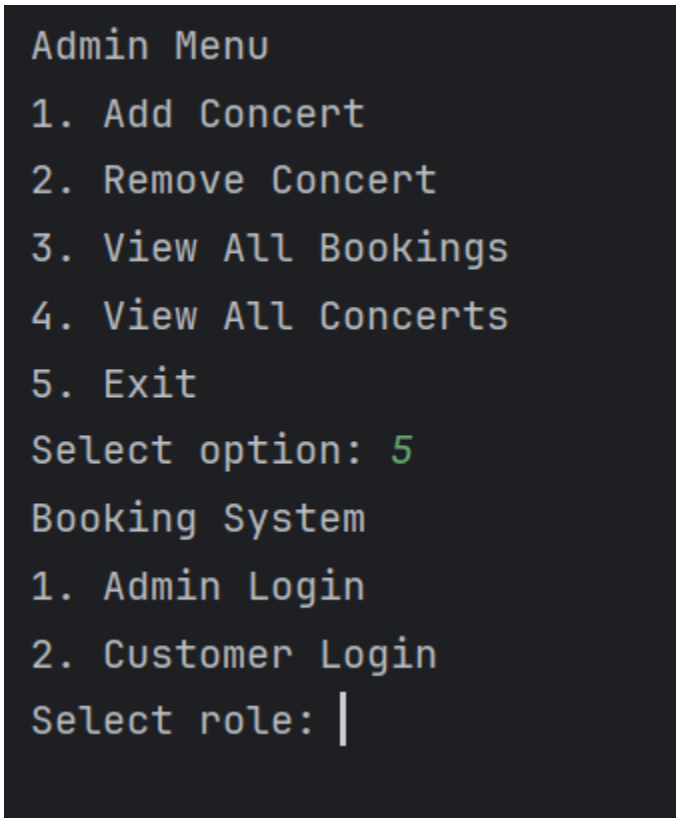
```
1. Add Concert
2. Remove Concert
3. View All Bookings
4. View All Concerts
5. Exit
Select option: 2
Available Concerts :
1. Concert'
Select one that you want to remove :
1
Concert removed successfully.
```

View ALL CONCERTS : shows the available concerts and how many people booked it.

```
Admin Menu
1. Add Concert
2. Remove Concert
3. View All Bookings
4. View All Concerts
5. Exit
Select option: 4

All Concerts (3):
- Concert' | Date: 2006-07-08 | Venue: MontrealStadium | Booked: 0/100
- A | Date: 2006-07-07 | Venue: MontrealStadiu, | Booked: 0/200
- B | Date: 2006-07-06 | Venue: TorontoSatdium | Booked: 0/102
```

RETURNING TO MAIN MENU:



CUSTOMER MENU :

```
Booking System
1. Admin Login
2. Customer Login
Select role: 2
Enter your ID: 6223602
Enter your name: Andy
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option:
```

View all available Concerts :

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 1
Available Concerts (4):
- Montreal | Date: 2006-07-07 | Venue: MontrealStadium | Available Seats: 103
- Montreal2 | Date: 2006-07-08 | Venue: MontrealStadium | Available Seats: 112
- Montreal3 | Date: 2006-07-09 | Venue: MontrealStadium | Available Seats: 194
- Montreal4 | Date: 2006-07-05 | Venue: MontrealStadium | Available Seats: 102
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option:
```

BOOK A CONCERT:

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 2
Available concerts:
1. Montreal (103 seats available)
2. Montreal2 (112 seats available)
3. Montreal3 (194 seats available)
4. Montreal4 (102 seats available)
Select concert to book :
1
Successfully booked Montreal
```

After viewing the concert, it shows available seats decreased by one. (103->102)

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 1
Available Concerts (4):
- Montreal | Date: 2006-07-07 | Venue: MontrealStadium | Available Seats: 102
- Montreal2 | Date: 2006-07-08 | Venue: MontrealStadium | Available Seats: 112
- Montreal3 | Date: 2006-07-09 | Venue: MontrealStadium | Available Seats: 194
- Montreal4 | Date: 2006-07-05 | Venue: MontrealStadium | Available Seats: 102
```

View my bookings: (as customer)

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 4
Your bookings:
Concert{name='Montreal', date=2006-07-07, venue='MontrealStadium',102seats left}
```

Cancel Booking:

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 3
Your bookings :
1. Montreal
Select booking to cancel
1
Booking canceled for Montreal
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option:
```

After cancelling, it restores the seats and removes concert from the user.

```
Booking canceled for Montreal
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 1
Available Concerts (4):
- Montreal | Date: 2006-07-07 | Venue: MontrealStadium | Available Seats: 103
- Montreal2 | Date: 2006-07-08 | Venue: MontrealStadium | Available Seats: 112
- Montreal3 | Date: 2006-07-09 | Venue: MontrealStadium | Available Seats: 194
- Montreal4 | Date: 2006-07-05 | Venue: MontrealStadium | Available Seats: 102
```

Quitting customer menu:

```
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 5
Booking System
1. Admin Login
2. Customer Login
Select role:
```

What if customer fully booked concert: (The concert won't be seen)

```
Select option: 2
Available concerts:
1. Montreal (103 seats available)
2. Montreal2 (112 seats available)
3. Montreal3 (194 seats available)
4. Montreal4 (102 seats available)
5. af (1 seats available)
Select concert to book :
5
Successfully booked af
Customer Menu
1. View Available Concerts
2. Book Concert
3. Cancel Booking
4. View My Bookings
5. Exit
Select option: 2
Available concerts:
1. Montreal (103 seats available)
2. Montreal2 (112 seats available)
3. Montreal3 (194 seats available)
4. Montreal4 (102 seats available)
Select concert to book :
```

From the admin view of all bookings, it will show that 2/2 is booked in the concert system.

```
Admin Menu
1. Add Concert
2. Remove Concert
3. View All Bookings
4. View All Concerts
5. Exit
Select option: 3
Montreal - Booked 0/103
Montreal2 - Booked 0/112
Montreal3 - Booked 0/194
Montreal4 - Booked 1/102
af - Booked 2/2
```

CHALLENGES FACED DURING PROCESS:

In this project, I have faced several challenges. First was with the .txt file, where I had trouble writing the concerts into the file, as everything had to be formatted properly. I made sure that everything was parsed correctly with try-catch blocks to handle the IOException. Another challenge faced was making sure that when booking a ticket, the seats available decreased and the booked seats increased. To prevent this, I made sure the seats and booked were updated in a single transaction (synchronized) and that negative values were prevented.The use of downcasting was also a challenge as with the subclass being converted to a superclass reference . Another challenge was the first time using GitHub by committing and pushing.

LEARNING OUTCOMES :

In this project, I have understood more about abstract methods/class, inheritance(User -> admin/ customer) and interface (bookable). I have also understood more about file input and output in order to read or write structured data in a .txt file. I have bettered my understanding of error handling as it improved my resilience with exceptions. I also learned more about method overriding such as viewEvents() in Admin and Customer. Finally, I have learned more about unit testing where it helped me check for errors in certain parts of my project. I have learned a new annotation such as @BeforeEach where it prepares my object for every test case. In a more general approach, I have learned how to use GitHub, which will help me for the future programming projects that I will do