**Final Project Report**

Student Name: Choi Jin Woo

Student ID: 2020315798

**1. Briefly describe the project purpose:**

This project is mainly about Movie Theater Ticketing System for SKKU. Its name is SKKUBOX, which came from MegaBox, Korean famous multiflex. Whole background was from my experience as a partimer in Megabox. There were three parts:Box, Con, Usher. Box handles ticketing services, Con handles all the concession goods such as popcorn, coke, and Usher serves people for entering right room. During the work, I worked as Con, which was physically tough position. I always admired to work as a Box crew. So I proposed to make ticketing system with JAVA in this project.

The project is technically based on MVC model. All GUIs are view and there are some interfaces and implemented classes for Model. Since there is no database, I used .txt with File I/O to make it work as DB.

The user can login as 'user' or 'admin'. In 'user' login type, the user can reserve the movie whenever there is an available seat and no limitation with age. There are two steps: selecting the movie and selecting the seats. In 'admin' login type, the administrator can enroll the movie with some information.

Also, there is SessionManage to maintain user session information.

With this project, I wanted to implement login functionality, DB sharing with both 'user' and 'admin', some advanced GUI approach.

**2. Draw the logic flow of the program (with flowchart):**

MOVIE.txt

USER.txt

RESERVATION.txt

User Implement

Movie Implement

Movie Interface

User Interface

SessionManager

UserBoxReservation

UserBoxFormSeatSelect

UserBoxFormMovieSelect

AdminBoxForm

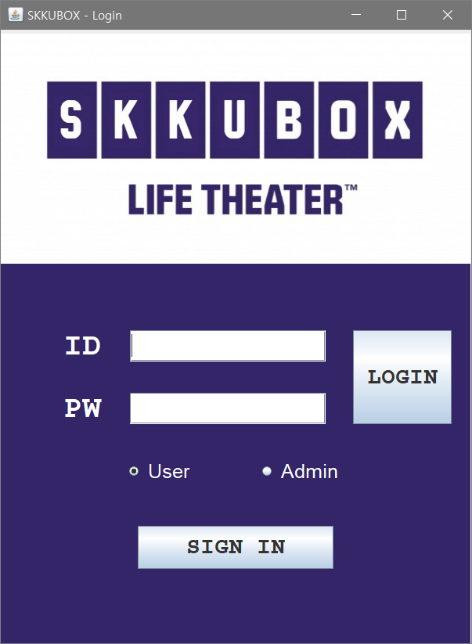
AdminManager

UserManager

UserRegisterForm

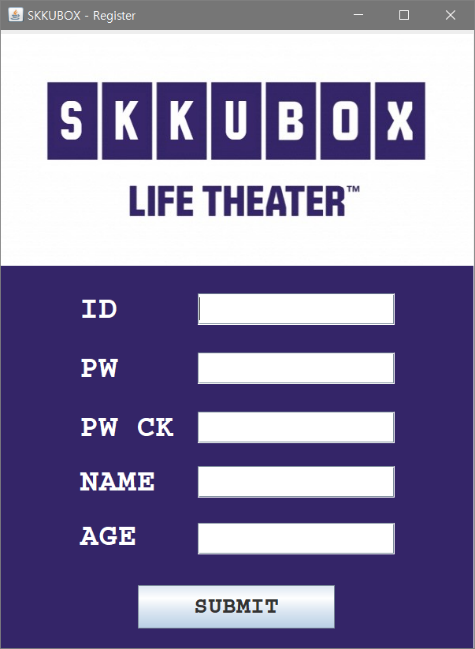
UserLoginForm

**3. Provide screenshots for each screen with brief description:**



1. Login Page

- User can login with either user mode or admin mode. Also, user can sign in with user mode. Admin account is initally enrolled in USER.txt.

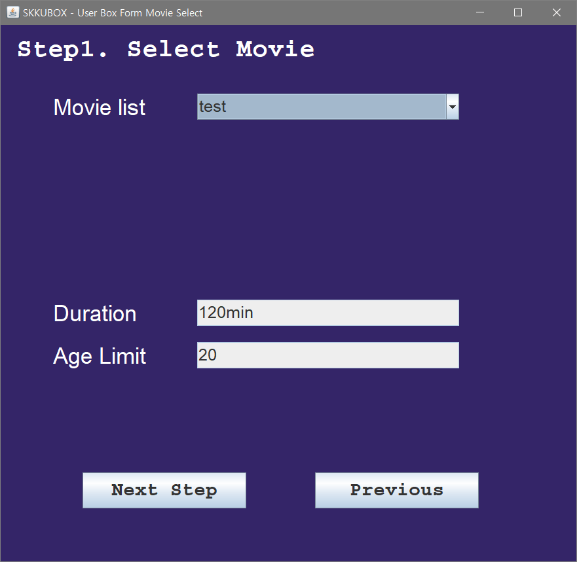
2. Register Page

- User can register account with inserting ID, PW, Name, Age. Name will appear at the result page of reservation and Age will work as limitation of reservation per movie.

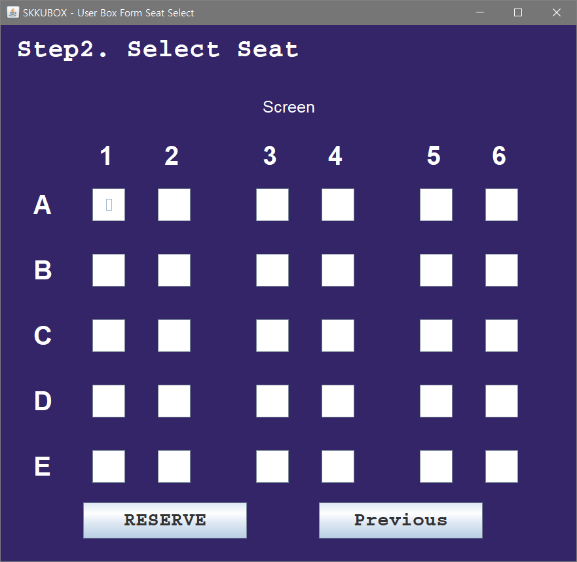
3. Movie Register Page

- Administrator can enroll the movie by inserting Name, Duration, Age



4. Movie select Page

- There are list of enrolled movie. If age limit is not a problem, the user can proceed with Next step button.

5. Seat select Page

- There are 30 available seats initially. User can choose maximum 30 seats, if there's any booked one. If other user reserved some seats, the seats will be disabled to click.

6. Reservation Result Page

- For the result Page, there shows the movie name and reserved seats with its own row/col name, not index name.

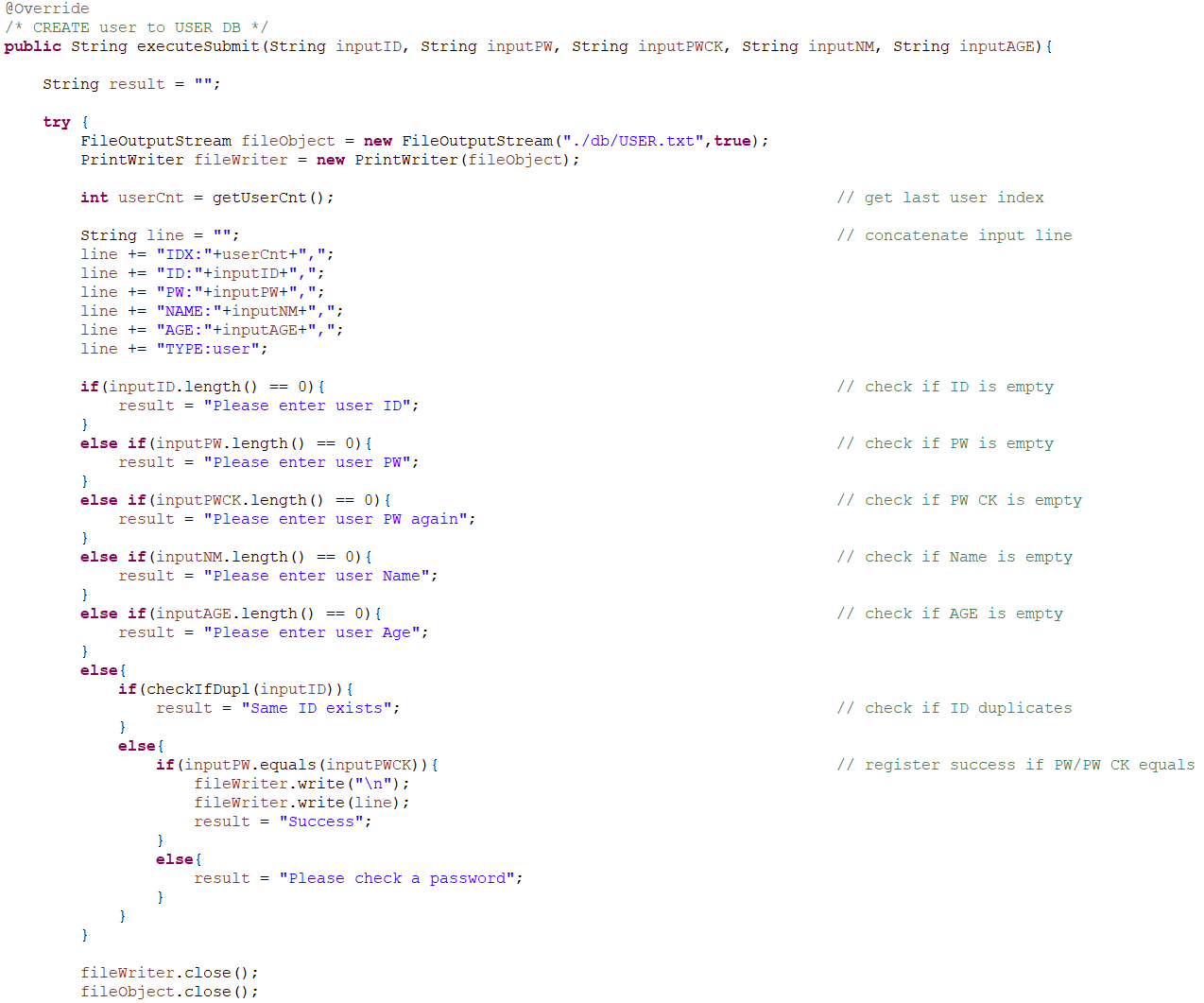
Also, the user name will appear at the bottom by using SessionManager.

**4. Explain the code of the main functionalities**

1. Login logic

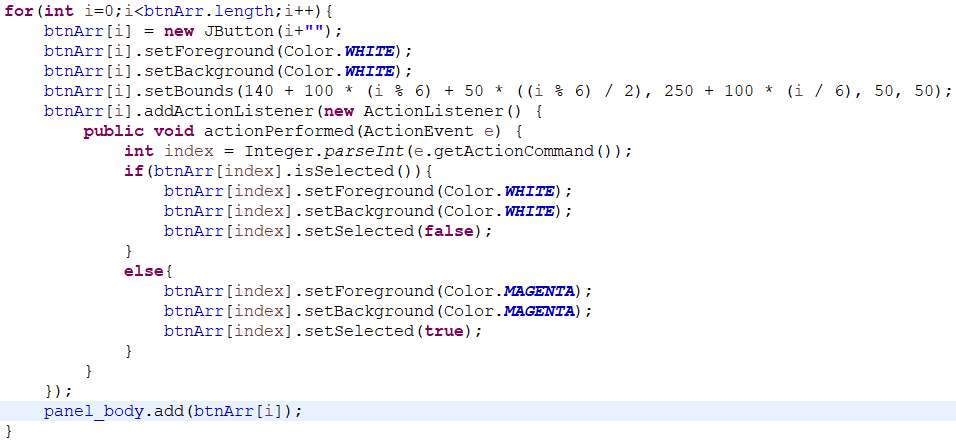
Most functionalities follow the same procedure, so I want to propose brief example with login function. If user clicks login button, it will call userImpl.executeLogin. It will

select user accounts and compare with user input. If login succeed, session will take user info also.

****2. Register logic

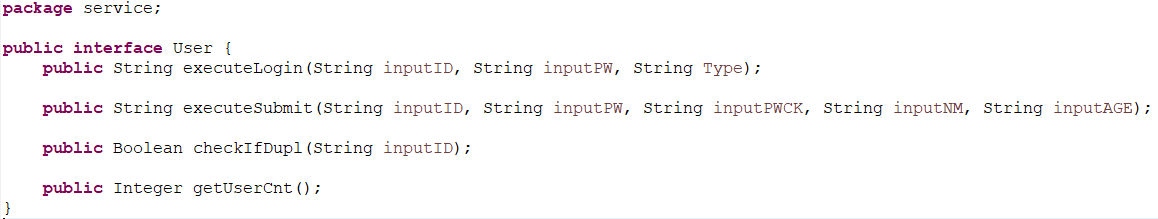
In executeSubmit method, which is account enrollment, it checks if there's any empty input or ID duplicates with current DB. But for any errors, it will insert data to txt file.

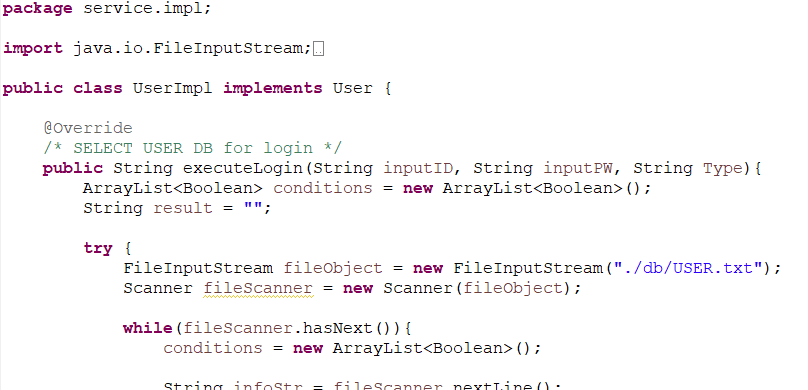
3. Seat array



I implemented seats by array with Jbutton, not with GUI. It was impossible to make each objects so I used array.

4. Dependency

****

****

When I searched about Model, I found there is some importance with dependency when making this kind of project. So I made interface and implement classes.

**5. Explain what is included in your project and why it is used (Polymorphism, Inheritance, File I/O, etc)**

1.FILE I/O

- FILE I/O is used as work as a database. There are three txt that been used as DB table. The text is similar to somewhat JSON type but I used split method to get key, value couple. Unlike a real database, all data types were String, although there is 'age', 'duration' fields.

2.Inheritance

- Inheritance is used for the movie object. The superclass is Product class, which has a property of 'index' and 'name'. Movie class extends Product class with its own property of 'age' and 'duration'. If I could have implemented Concession portion, Product class would take Food class with 'price' and 'amount' property.

Movie class object is used as return object with List<Movie> when getting result from the txt File I/O method.

3.Interface

- Interface and its implementation is used for dependency injection. The code might become longer, but with the injection, we can improve flexibility while designing the project.

4. ETC

- Package system, ArrayList datatype are also used for maintaing the file system of the project.