**Final Project Report**

Student Name: **SeHyeong-Lee(Korean Name: 이세형)**

Student ID: **2020-312145**

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

텍스트, 친필, 종이이(가) 표시된 사진

자동 생성된 설명This program was designed to significantly increase customer satisfaction with this movie platform service. The program provides a comprehensive service including movie ticketing, movie rating, and movie score prediction through the 'Slope-One' algorithm, all in one place. This allows users to select and book movies that are closer to their tastes. Additionally, the accumulation of coupons and points with every movie booking is sure to further enhance user satisfaction. For those unfamiliar with special theaters, this program provides organized information for each theater type.

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

First picture is a logic flow of the program and second picture represents the relationships between objects in the program, depicted using the MVC pattern.

텍스트, 친필, 화이트보드, 문서이(가) 표시된 사진

자동 생성된 설명

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

1. EECinema window:

텍스트, 스크린샷, 소프트웨어, 웹사이트이(가) 표시된 사진

자동 생성된 설명

2. Log-In window(multi window):

텍스트, 스크린샷, 소프트웨어, 웹사이트이(가) 표시된 사진

자동 생성된 설명

When you click log-in button, Log-In window appears. After logging in, program update screen and you can use services.

(Before logging in, you can not use other services. There will be an pop-up window noticing about it.)

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

3. Book-Ticket window: 텍스트, 스크린샷, 소프트웨어, 웹사이트이(가) 표시된 사진

자동 생성된 설명

If you click Book-Ticket button, Book-Ticket window appears. You can select several options and then book a ticket. Tciket will be printed in text file with randomly generated reservation number.

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

4. Theater Information window:

텍스트, 스크린샷, 직사각형, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

If you click theater-information button, you can get information about each theater types(There are four theater types in total.)

5. Rate Movie window:

텍스트, 스크린샷, 인간의 얼굴, 사람이(가) 표시된 사진

자동 생성된 설명

If you click Rate Movie button, Rate Movie window appears. You can rate for each movie and it will be updated in a screen(eg. movie score)

6. Personalized Score Prediction window:

텍스트, 인간의 얼굴, 사람, 스크린샷이(가) 표시된 사진

자동 생성된 설명

If you click predict score button, Personalized Score Prediction window appears. It will predict score for each movie by using 'Slope-One' algorithm. The 'Slope-One' algorithm increases in accuracy as the amount of stored data increases, and decreases in accuracy as the amount of data decreases. Therefore, in the initial state where no rating information is stored, it predicts a score of zero. You can increase the accuracy by accumulating additional rating information.

You can use the services with the multi-window panes, as described earlier, open simultaneously. However, I've ensured that the same type of window doesn't open more than once at the same time for less confussion.

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

The program consists of a total of 15 classes. There are two classes that serve the role of Model in the MVC pattern, six classes that serve the role of View, and six classes that serve the role of Control, along with one class responsible for the Main Thread. The main window is handled by one class each from the Model and Control roles. The remaining five classes handle multi-windows and consist of five Model role classes and five Control classes. The main functions of these multi-windows include 'Log in', 'Book movie tickets', 'Get information by cinema', 'Rate movies', and 'Predict movie scores'.

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

**-Multi-threading:** I have implemented the multi-threading of SwingWorker to enable multiple types of movie platform services to proceed simultaneously.

**-Multi-window:** Given the characteristics of a movie platform, having more than one window open at the same time is more efficient and maximizes the interface, so I implemented a multi-window.

**-File I/O:** I implemented a system where booked movie tickets are output in the form of a text file.

**-Inheritance:** I have inherited from the Exception class and implemented anew to handle various exceptions.

**-List:** I used a list instead of an array for efficiency in storing and organizing object instances.

**-Various pop-up windows:** I implemented pop-up windows for each situation, whether an exception occurs, to provide information, or for logging in.

**-Polymorphism:** I made program caught different types of exception handling in the catch block as an Exception class object using polymorphism. (Reason: This is possible because the Exception class is a superclass.)

**-Exception handling:** I handled all expected exceptions in each situation with try-catch statements for a safer and optimized platform service.