**Movie Data Analysis System**

**System Design Specification (SDS)**

**<Version 3.0>**

**<Date 5.20>**

<Project team members>

**2130026223 Zou Bojun**

**2130026164 Xie Qiwen**

**2130026190 Yuan Yichen**

**2130026040 He Jiacheng**

**2130026100 Lyu Jun hao**

**Computer Science and Technology Program**

**United International College**

1. Document Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| ***Change Date*** | ***Changed By*** | ***Version*** | ***Change Description*** |
| *22/04/2023* | *Yichen Yuan* | *1.0* | *Complete the design of the entire report.* |
| *17/05/2023* | *Yichen Yuan* | *2.0* | *Modify the report according to the functionality we added recently and complete the design of the database* |
| *20/05/2023* | *Yichen Yuan* | *3.0* | *Modify the report based on the finalized functionality* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | *<List the design elements that were changed>* |

1. Table of Contents

[1. Document Change Log 2](#_Toc135520838)

[2. Table of Contents 3](#_Toc135520839)

[3. Design Overview 4](#_Toc135520840)

[4. Tools and Standards 5](#_Toc135520841)

[4.1. Tools 5](#_Toc135520842)

[4.2. Standards 5](#_Toc135520843)

[5. User Interface Design 6](#_Toc135520844)

[5.1. Login Page 6](#_Toc135520845)

[5.2. Registration page 7](#_Toc135520846)

[5.3. Home page 8](#_Toc135520847)

[5.4. Film page 8](#_Toc135520848)

[5.5. Film detail page 10](#_Toc135520849)

[5.5.1. Film detail page (not log in) 10](#_Toc135520850)

[5.5.2. Film detail(logged in) 12](#_Toc135520851)

[5.6. Hot showing film detail page 19](#_Toc135520852)

[5.7. Daily Recommendations page 19](#_Toc135520853)

[5.8. Cinema page 21](#_Toc135520854)

[5.9. Cinemas detail page 21](#_Toc135520855)

[5.10. Admin films page 23](#_Toc135520856)

[5.11. Admin films detail page 24](#_Toc135520857)

[5.12. Admin cinemas. page 24](#_Toc135520858)

[5.13. Admin cinemas detail page 25](#_Toc135520859)

[6. Database Design (If Applicable) 26](#_Toc135520860)

[7. Diagrams 33](#_Toc135520861)

1. Design Overview

**Overview:**

In this project, we are going to build a one-stop film Internet platform integrating film information, online ticket purchasing, user interaction services, and information management. When people are trying to find some basic information of a film such as the title, directors, actors, synopsis, comments etc., or they want to purchase the film ticket for hot showing films, they can use our website. In our website, users can browse the information of different films. If the user has logged in, he can comment and rate the film, or make the purchase if he wanted. Some basic information of cinemas is also provided. When the user is making a purchase, he should choose the cinema, hall, and session of the specific film to complete the purchase.

Data analysis part：

On the basis of realizing all the above functions, we added the real-time box-office prediction of the hot movie and personalized recommend movie (based on the type of movie users have seen and the region where the movie was shot).

**Aim:**

1.Provide users with a website so they can get all the movie information and buy the movie ticket they like.

2.Help client find the right movie for them(recommend)

3.Provide more detailed movie data for users to choose which movie to watch(Forecast box office)

1. Tools and Standards

## Tools

* HTML
* CSS
* JavaScript
* Python 3.10
* Request (python library)
* Matplotlib (python library)
* Django
* MySQL
* SQL

## Standards

* Testing on latest stable version of Google Chrome browser(Version: 112.0.5615.137 (64-bit) or later)
* Using HTML5 Standard.
* Using UTF-8 encoding Standard.

1. User Interface Design

## Login Page

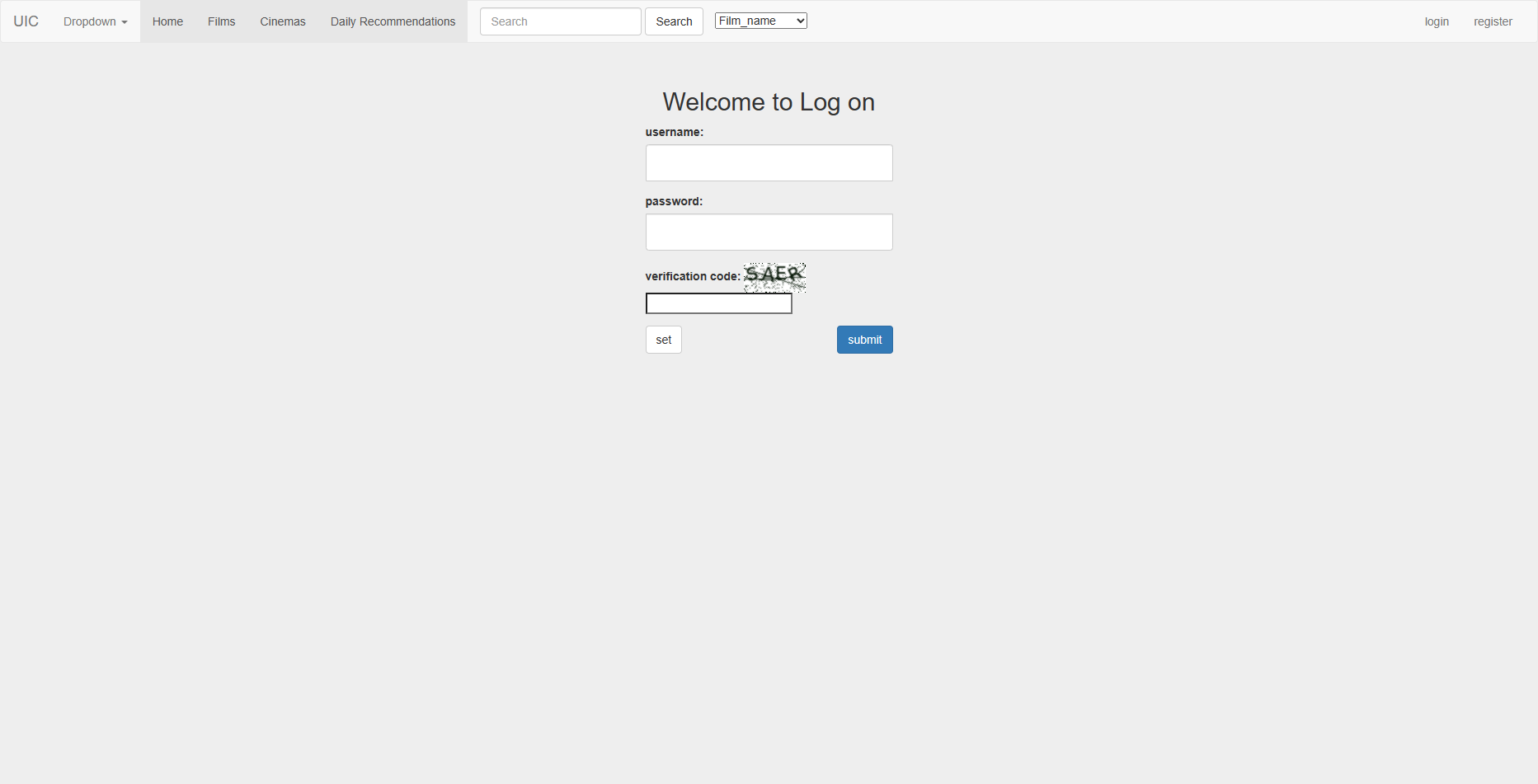
 Figure 1

Figure 1 is a login page. A form displays on the page with a username entry box, a password entry box, a verification code, and a submit button. Users can log in here, if the user has not registered, you can click the registration button to jump to the registration interface. If the user makes a typo, the "set" button can also be used to clear the content

## Registration page

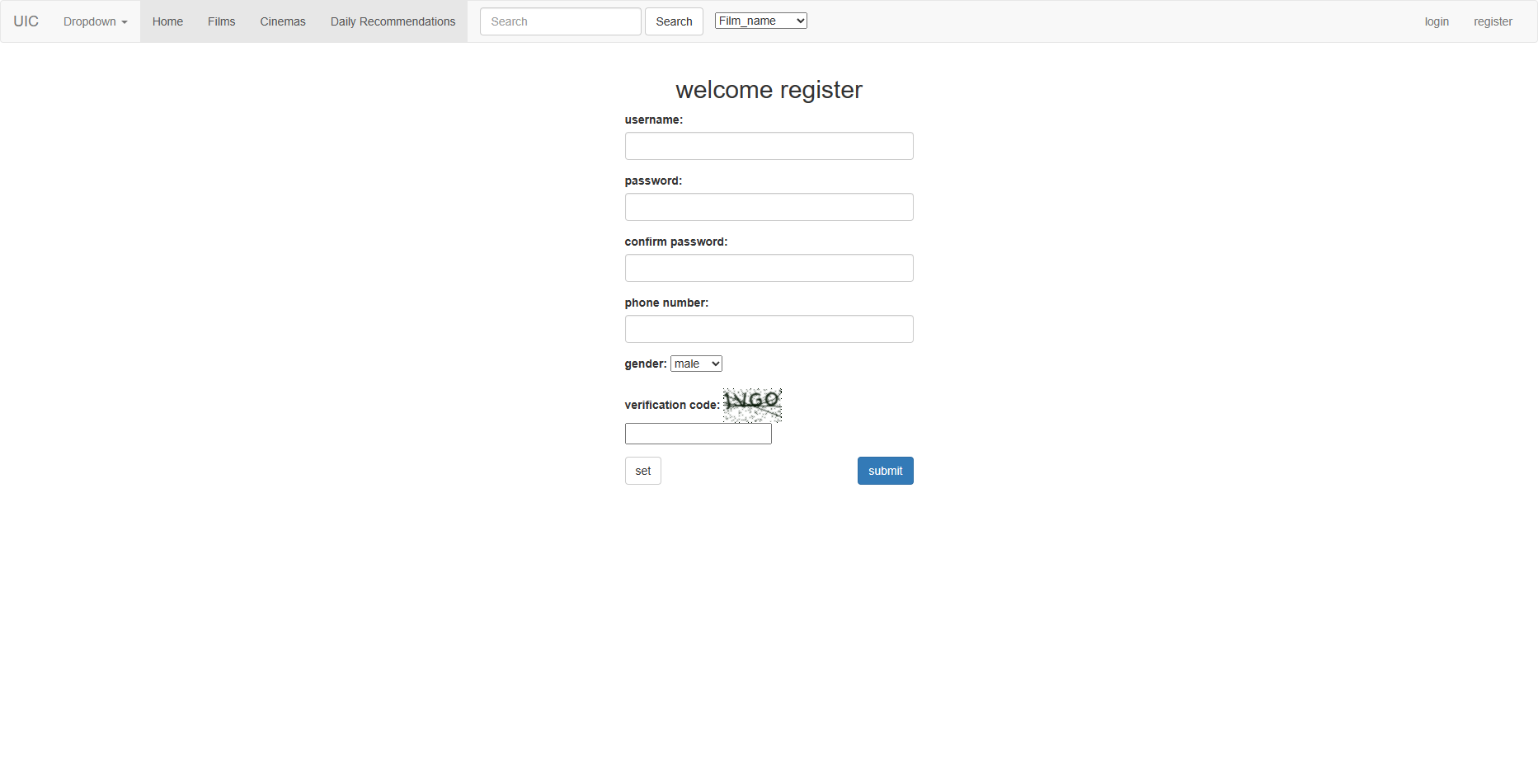
**

Figure 2

If the user is not registered, you can fill in the account and password here, and confirm the password to complete the registration. If the user makes a typo, the "set" button can also be used to clear the content

## Home page

图形用户界面

描述已自动生成 Figure 3

This page shows the movies that are currently playing in theaters, as well as the top 10 movies with the highest ratings. Click on the picture or text page to access the movie details page. In the index above the user can choose to jump to a different main page. Users can also search for movies and cinemas based on different criteria in the search box. If the user wants to log out, click the "Logout" button in the upper right corner.

## Film page

图形用户界面, 网站

描述已自动生成 Figure 4

This page shows all the movies in the website and shows its score under the corresponding movie. All the movies here are crawled from <https://movie.douban.com/> website by us, and the information of the movie is real and reliable. User can filter the movie types according to the index to find the movie you want to watch. Click the picture link to enter the corresponding movie details page.

图形用户界面, 网站

描述已自动生成

*Figure 5*

In figure 5, we can see that at the bottom of this page there is a pagination function, and our database has 260 real movies imported by the crawler. We implement and use pagination to make the page more concise.

## Film detail page

### Film detail page (not log in)

Whether the user is logged in when entering the movie details page, the function of the page and the form of the page are different.

图形用户界面

描述已自动生成

Figure 6

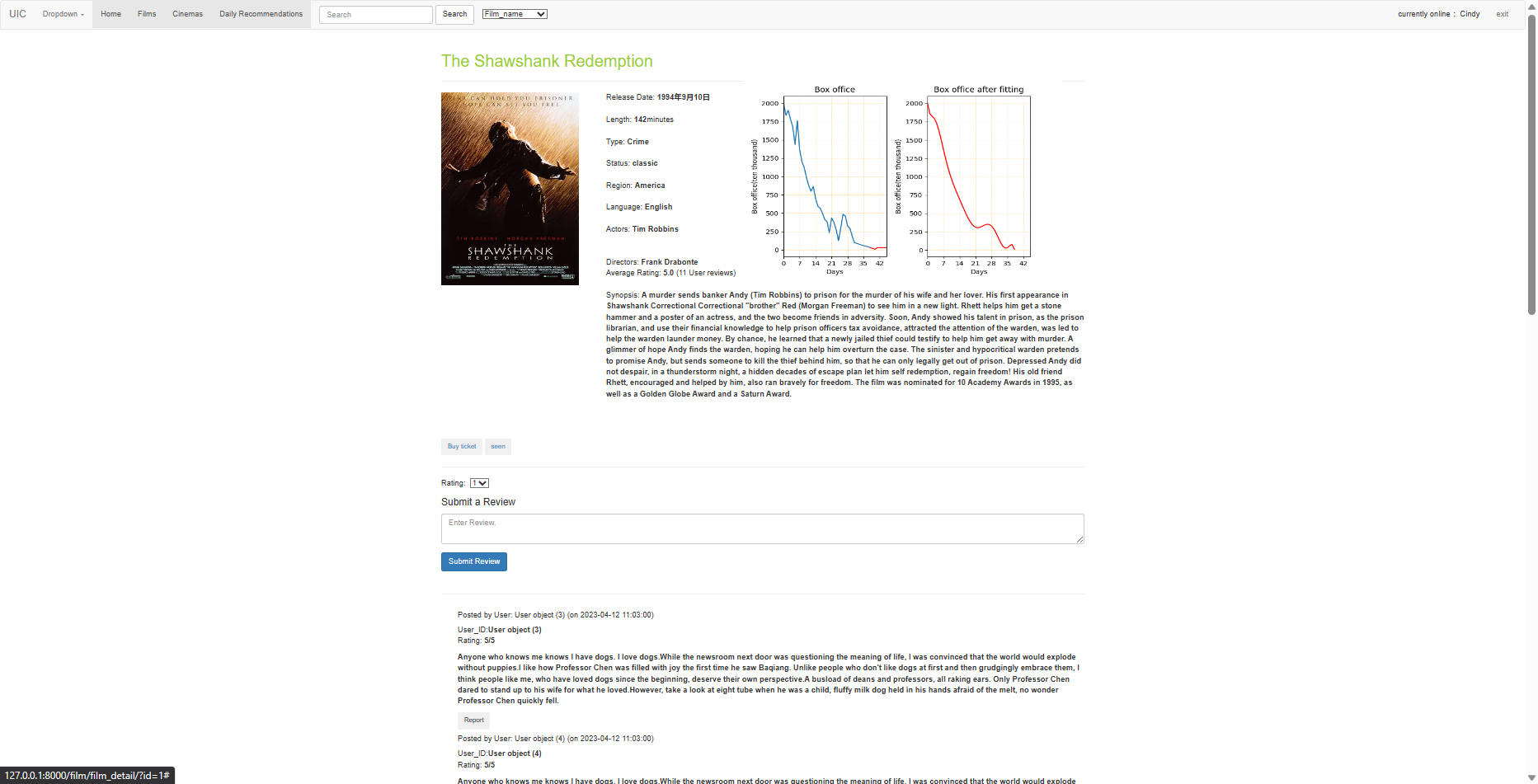
1. Information display：In the upper left corner, you can directly select registration or login and go to the corresponding interface.

2. Information display：Display the film information (rate refresh in real time) and the corresponding picture.

3. Information display：You can see other users' ratings and reviews of the film. You can see the user name and review time of the reviewer, but cannot see the user ID of the reviewer.

4. If you post a comment, you will be prompted to log in first.

### Film detail(logged in)

 Figure 7

1. Information display: Display the user name and welcome in the upper right corner

2. Information display: Display the film information (rate refresh in real time) and the corresponding picture.

3. Ticket-buying function: After clicking the “Buy ticket” button in the film details interface, the user will directly skip to the cinema interface and select the cinema that can play the film.

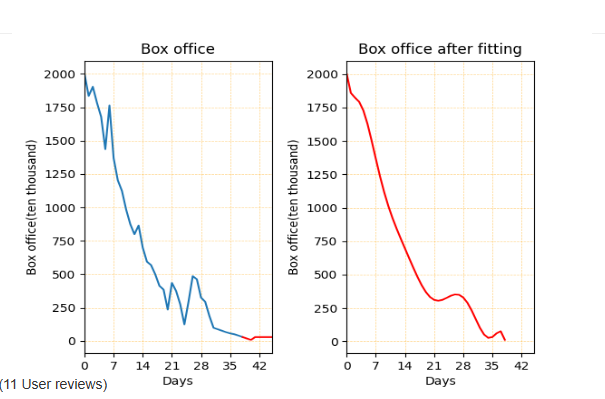


Figure 8

Data Analysis functionality:

4. Information display: For a movie that has been removed from cinema we do a box office visualization, we will show the box office of the movie in the first month since its release, and the box office trends in the latter.We use linear regression to introduce a straight line of shape y = ax + b, according to which we show users the trend of the movie box office a month later. (See the short red line after blue line in figure 8). We also used polynomial fitting to represent fitted box office data. Make the box office data more intuitive, as shown in the image in Figure 8(See the right graph in figure 8).

图形用户界面, 文本, 应用程序

描述已自动生成

Figure 9

5. Saving film have\_watch history(see figure 9): Click the "seen" button to save the movie to the user's history, and then you can see the history of the last 20 movies watched by the user in the "daily recommendation page"

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

Figure 10

After clicking the "seen" button will refresh the page and tell the user "You have watched the movie!" in the top left corner.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Figure 11

6. Comment function :Score and comment on the film (if the comment is empty, the user will be prompted to write a comment). The user's score and comment on the film are bound together. In order to prevent the user from flooding the film rate and comments, each user can only review and rate each film once and only once. After the user scores, the page is refreshed, and the overall score of the movie is recalculated and refreshed.

After submit review :图形用户界面, 文本, 应用程序

描述已自动生成

Figure 12

图形用户界面, 文本, 应用程序

描述已自动生成

Figure 13

7. Edit comment： After users make comments, they can edit or delete their comment and rate directly. If the delete button is selected, a popup window will be displayed on the page to ask users to confirm twice. If the user chooses to confirm, that comment and rate will be deleted. If the Edit button is selected, a popup window will be displayed on the page to allow users to re-score and comment. After the user confirms, the original comment and rate will be deleted and the new comment and rate will be published.

After Editing:

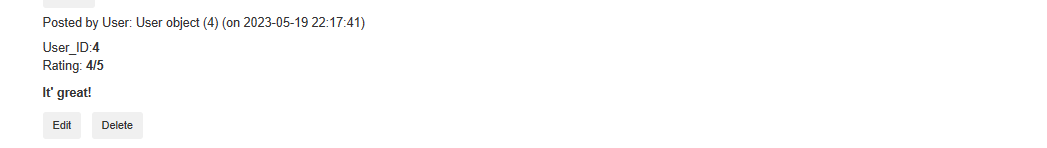


Figure 14

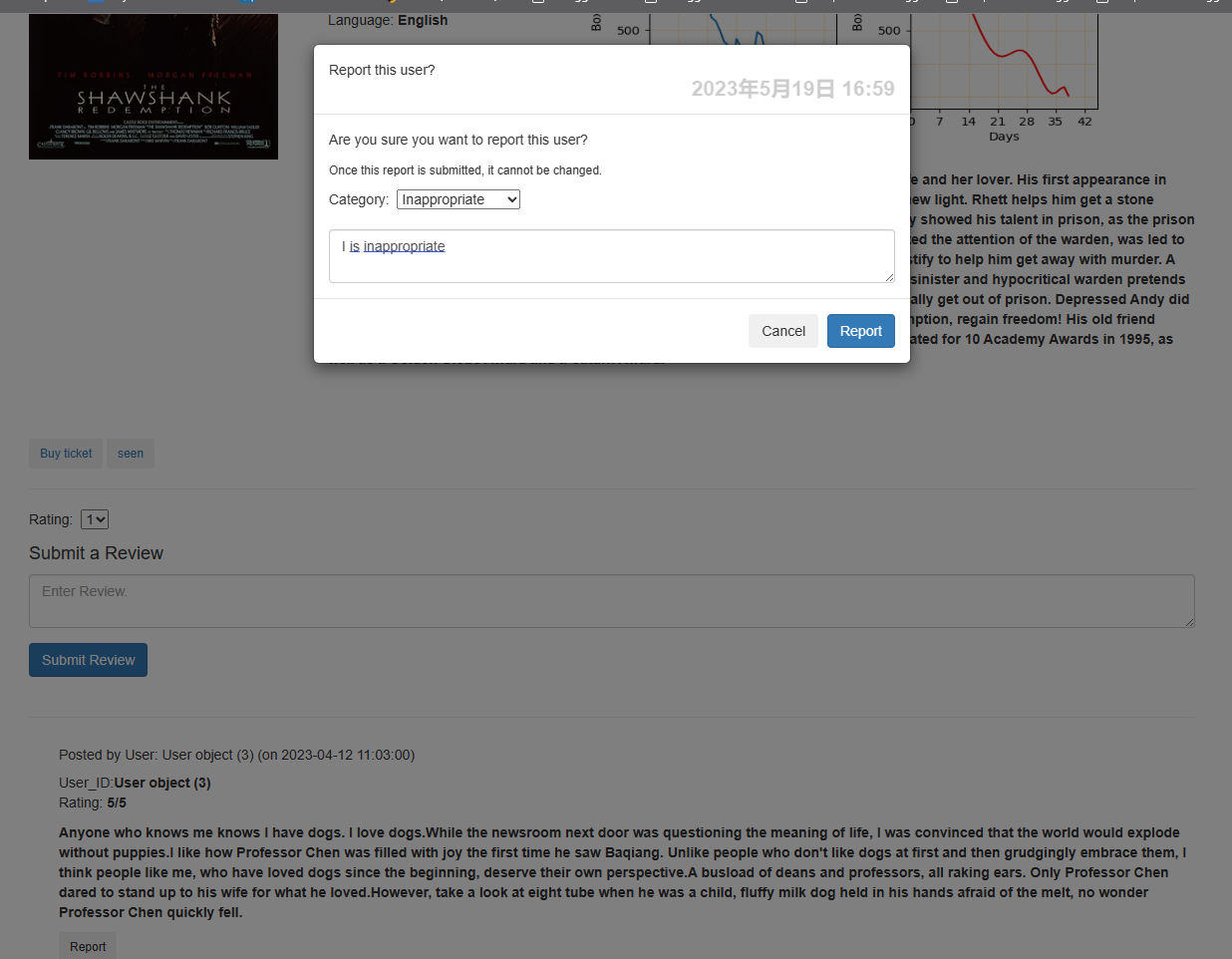


Figure 15

8. Information display：You can see other users' ratings and reviews of the film. You can see the username of the reviewer, the time of the review, and the user ID of the reviewer.

9. Report function：Users can report comments that they think have adverse effects. After clicking the "report" button, the interface pops up and asks whether to report the comment again. The informant can choose the type of report and optionally fill in the explanation and reason for the report .

图形用户界面, 文本, 应用程序

描述已自动生成

Figure 16

10. Delete function: Users can click the "delete" button to delete the comment directly

Data Analysis functionality:

## Hot showing film detail page

图形用户界面, 应用程序

描述已自动生成

Figure 17

The box office graph in figure17 predict the box office of the movie being released in the cinema now. We use the existing data of the film for polynomial fitting and use the resulting polynomial to infer the box office of future movies.

Data Analysis functionality:

## Daily Recommendations page

图形用户界面, 应用程序

描述已自动生成

Figure 18

On the daily recommendation page, we will make recommendations using Collaborative Filtering algorithm.

We select candidates for films based on multiple dimensions (directors, types and regions), We find the user’s favorite directors, types and regions based on the movies that user has seen and then select a certain number of movies (user unwatched) to recommend.

Meanwhile, 20 movies that user have seen recently will be recorded and placed below. On the right side, we query the regional distribution of movies and types of all the movies that users have watched and draw the pie charts.

## Cinema page

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Figure 19

After clicking the “Buy ticket” button in the film details interface or click cinema at the top index, the user will directly skip to the cinema interface. On this page, customers can select the type of movie theater in the "Type" column, select the theater they want, and click "Buy ticket" to purchase tickets

## Cinemas detail page

图形用户界面, 网站

描述已自动生成

Figure 20

On this page, customers can choose to purchase tickets in different theaters for different performances. When the ticket is purchased, a prompt will pop up and you will return to the cinema screen after clicking "OK"

图形用户界面, 文本

描述已自动生成

Figure 21

## Admin films page

图片包含 图形用户界面

描述已自动生成

Figure 22

This page is for staff only. On this page, click "Show Detail" to go to the information page of the corresponding movie.

## Admin films detail page

图形用户界面

描述已自动生成

Figure 23

This page only allows staffs to browse after login. From there, you can view the movie details.

## Admin cinemas. page

图形用户界面, 应用程序

中度可信度描述已自动生成 Figure 24

This page only allows staffs to browse after login. This page only allows staffs to browse after login. This page only allows staffs to browse after login. The administrator can use “Show Detail” to go to the show\_cinemas page, and “Delete” to delete cinema.

## Admin cinemas detail page

图形用户界面, 文本, 应用程序

描述已自动生成 Figure 25

This page only allows employees to browse after login. In this page, the administrator can view the information of the cinema.

1. Database Design (If Applicable)

图示, 示意图

描述已自动生成 Figure 26

**Note: Note: 3% of the tables may not actually be used at the end because the functionality of some members is not implemented. But I still think this is a well design of a complete database**

**Functional Dependencies & Schemas**

* Functional dependency:
* Primary key: . Since the information of a film unpredictable and various, it is necessary to use a unique id to identify the film.
* Critical point: the image here is stored by BLOB.

In order to better manage the classification of the type, language, status, and region of a film, we separate them into single tables. It is more convenient and clearer for the administrators to manage the information like this.

* Functional dependency:
* Primary key:
* Functional dependency:
* Primary key:
* Functional dependency:
* Primary key:
* Functional dependency:
* Primary key:
* Functional dependency:
* Primary key: . Since a user can only post one review per film, the user ID and film ID determine R.

* Functional dependency: .
* Primary key: . Since a user can report other users’ review multiple times for various types of reasons, and each user can re0port the same review, only reportID can determine the other tuples in R.
* Functional dependency:
* Primary key: . Since we only need an actor\_id to help us find a certain actor, we set actor\_id as the primary key.
* Functional dependency: director\_id -> first\_name,last\_name
* Primary key: . Since we only need a director\_id to help us find a certain director, we set director\_id as the primary key.
* Functional dependency:
* Primary key: . In this table, we use unique combination (actor\_id, film\_id) to show the relationships of actors and films
* Critical points:

ALTER TABLE `act`

ADD CONSTRAINT `act\_ibfk\_1` FOREIGN KEY (`film\_id`) REFERENCES `film` (`film\_id`),

ADD CONSTRAINT `act\_ibfk\_2` FOREIGN KEY (`actor\_id`) REFERENCES `actor` (`actor\_id`);

In the table `act`, we want to make sure that every film and actor exist in table `film` and `actor`

* Functional dependency:
* Primary key: . In this table, we use unique combination (director\_id, film\_id) to show the relationships of directors and films
* Critical points:

ALTER TABLE `direct`

ADD CONSTRAINT `direct\_ibfk\_1` FOREIGN KEY (`film\_id`) REFERENCES `film` (`film\_id`),

ADD CONSTRAINT `direct\_ibfk\_2` FOREIGN KEY (`director\_id`) REFERENCES `director` (`director\_id`);

In the table `direct`, we want to make sure that every film and director exist in table `film` and `director`

* Functional dependency:
* Primary key: . Because the cinema may have the same name, we need a unique cinema\_id to distinguish, so we set cinema\_id as the primary key.

* Functional dependency:
* Primary key:. Because other attributes are likely to be duplicated，We need a unique hall\_id for the primary key
* Critical points:

ALTER TABLE hall

ADD FOREIGN KEY (cinema\_id) REFERENCES cinema(cinema\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

When we want to delete the cinema, we need to delete all the hall it contains. So we use the Cascading Actions.

* Functional dependency:
* Primary key:. Because other attributes are likely to be duplicated，We need a unique session\_id for the primary key
* Critical points:

ALTER TABLE session

ADD FOREIGN KEY (film\_id) REFERENCES film(film\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE session

ADD FOREIGN KEY (hall\_id) REFERENCES hall(hall\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

When we want to delete the film or hall, we need to delete all the session it contains. So, we use the Cascading Actions.

DELIMITER |

CREATE TRIGGER film\_session

AFTER INSERT ON session

FOR EACH ROW

BEGIN

IF new.film\_id NOT IN (

SELECT film.film\_id FROM film WHERE status\_id = 1

)THEN

DELETE FROM session WHERE session.film\_id = new.film\_id;

END IF;

END;|

DELIMITER ;

According to reality, the film in the session needs to be hot showing, so we add this Trigger to limit that the film which have sessions should be hot showing.

* Functional dependency:
* Primary key: ticket\_id. Because other attributes are likely to be duplicated，We need a unique ticket\_id for the primary key
* Critical points:

ALTER TABLE ticket

ADD FOREIGN KEY (user\_id) REFERENCES user(id)

ON DELETE CASCADE

ON UPDATE CASCADE

ALTER TABLE ticket

ADD FOREIGN KEY (session\_id) REFERENCES session(session\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

When we want to delete the user or session, we need to delete all the ticket it related. So, we use the Cascading Actions.

DELIMITER |

CREATE TRIGGER seatNum\_capacity

AFTER INSERT ON ticket

FOR EACH ROW

BEGIN

IF new.seat\_num > (SELECT capacity FROM ticket JOIN session USING(session\_id) JOIN hall USING(hall\_id)) WHERE ticket\_id = new.ticket\_id.

THEN

DELETE FROM ticket WHERE ticket\_id = new.ticket\_id;

END IF;

END;|

DELIMITER ;

According to reality, the seat number of ticket should smaller than capacity of its hall, so we add this Trigger to limit that ticket. seat\_num < cinema.capacity.

* Functional dependency:
* Primary key: . The administrators may have same account name or password, so admin\_id should be used to distinguish them.
* Functional dependency:
* Primary key: . The users may have same personal information, so user\_id should be used to distinguish them.
* Critical points:

DELIMITER |

CREATE TRIGGER add\_reward\_point

AFTER INSERT ON ticket

FOR EACH ROW

BEGIN

UPDATE user

SET reward\_points = (reward\_points + 5)

WHERE user.user\_id = new.user\_id

END;|

DELIMITER ;

After buying a ticket, the user will gain 5 reward points, so we add this trigger.

* Functional dependency:
* Primary key: .

(Phone is a multivalued attribute, primary key is schema itself. )

* Critical points:

ALTER TABLE user\_phone

ADD CONSTRAINT user\_cancellation

ADD FOREIGN KEY () REFERENCES user()

ON DELETE CASCADE

After user account cancellation, delete the user's phone using cascading action.

1. Diagrams

图示

描述已自动生成

图示

描述已自动生成

*Figure 27*

图示

描述已自动生成

*Figure 28*

图示

描述已自动生成

*Figure 29*