|  |  |
| --- | --- |
|  | **FPT ACADEMY INTERNATIONAL**  **FPT – APTECH COMPUTER EDUCATION** |

**Centre Name: ACE-HCMC-2-FPT.**

**Address: 590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam.**

**X-STAR CINEPLEX**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | Mr. Cat Nguyen Hieu Tuan | |
| **Semester:** | II | |
| **Batch No:** | T1.2008.E1 | |
| **Group No:** | 1 | |
| **Order:** | **Full name** | **Roll No.** |
| 1. | Le Tuan Xuyen | Student1275055 |
| 2. | Tran Pham Thanh Hoang | Student1275235 |
| 3. | Nguyen Nhat Nam | Student1275240 |
| 4. | Phung Van Phuc | Student1277217 |

Table of Contents

[REVIEW 1 3](#_Toc77736284)

[1. INTRODUCTION 4](#_Toc77736285)

[2. EXISTING SCENARIO 4](#_Toc77736286)

[3. CUSTOMER REQUIREMENT SPECIFICATION 4](#_Toc77736287)

[4. FUNCTIONAL REQUIREMENT SPECIFICATION 5](#_Toc77736288)

[5. SYSTEM REQUIREMENTS 10](#_Toc77736289)

[REVIEW 1 – TASK SHEET 12](#_Toc77736290)

[REVIEW 2 13](#_Toc77736291)

[1. Use Case and Sequence Diagram 14](#_Toc77736292)

[2. Table Design 36](#_Toc77736293)

[3. Entity Properties 44](#_Toc77736294)

[4. Entity Relationship Diagram (ERD) 53](#_Toc77736295)

[REVIEW 2 – TASK SHEET 54](#_Toc77736296)

[REVIEW 3 55](#_Toc77736297)

[1. Site Map 56](#_Toc77736298)

[2. GUI Design 58](#_Toc77736299)

[REVIEW 3 – TASK SHEET 86](#_Toc77736300)

REVIEW 1

1. INTRODUCTION

X-star is a newly established movie theater. Their mission is to become a role model company for continuous development of the film industry in Vietnam. They are also determinedly providing movie-going experiences for audiences who currently have limited access to quality cinema environment through CSV programs such as Smile Moon and Movie for Everyone.

However, to expand the system, they need to build a website for guests to access tickets and experience the theater's online service system.

1. EXISTING SCENARIO

X-star is a newly established movie theater, but the number of customers is increasing

because of the quality service and reasonable prices. Therefore, they are facing the problem of serving customers at the theater because the number is too large. Besides, they also want to improve the service quality as well as promote the brand of the cinema.

Realizing that the cinema management in a traditional way is not effective, X-star decided to apply technology and build a website to optimize the operation of the theater's system and also allow customers to book tickets online. Building a website helps X-star save costs, manage theaters effectively and serve customers better. Thereby raising the quality and brand of the theater to customers

1. CUSTOMER REQUIREMENT SPECIFICATION

Allows users to visit the website to watch movie trailers and search for their favorite movie titles.

If the customer wishes to book tickets, they must register an account. A customer with an existing account can login. They do not have the permission to change any information apart from their own.

Customers can pre-select their favorite seats, choose the ticket theater location, ticket price as well as show time.

Customers can register as a member to book tickets to accumulate points and receive incentives.

For customers who already have an account, they can log in to book tickets, update and manage membership card information.

If an admin wants to change sensitive information (e.g. delete, ban customer's account, etc.). Aside from managing movies, categories, orders, etc. an admin can also view reports on movies, orders and customers.

1. FUNCTIONAL REQUIREMENT SPECIFICATION
   1. User Can Register
      * 1. Description

Customers must be able to register a new account. Once they have their account, they can login to buy tickets and access more features. A guest, who is not logged in, cannot buy ticket movies.

If customer do not have an account ,they can view movie and trailer information

* + - 1. Functional Requirement

User must provide the information below to create an account:

- Full Name (cannot be blank)

- Account(cannot be blank)

- Password (cannot be blank)

- Email (cannot be blank)

- Address (cannot be blank)

- Phone Number (cannot be blank)

- Birth Day(cannot be blank)

- Sex (male or female)

* 1. User Can Login
     + 1. Description

After registering an account, user can login. Once they have logged in, they can modify their personal information. They can become members and enjoy preferential programs for members

* + - 1. Functional Requirement

User must already have an existing account and provide Account and Password to login.

* 1. User Can Reserve Ticket
     + 1. Description

User can find products they want by searching or filtering results. User can buy products by adding chosen products to cart and proceeding to check out. Users can reserve their favorite seating position, and set the accompanying products as well as the time for the show.They will be asked to provide billing and payment information.

* + - 1. Functional Requirement

User must log in to use this function. User can only buy products that are still in stock. User needs to provide the information below to complete their order placement:

- Name of the receiver

- Required date

- Room number

- Show time

- Payment method

- Valid payment information (card owner, card number, security code)

* 1. User Can Write Reviews on Products
     + 1. Description

User can leave their review and rating on products. Each user can only post one review on each product. Rating can be done by choosing on a scale from 1 to 5, in which 5 is the best score. This score will be calculated and serve as the overall rating of the products.

* + - 1. Functional Requirement

User must login to use the review function. They must provide:

- Review title

- Review content

- A score on a scale from 1 to 5

* 1. User can manage their information in the user’s profile
     + 1. Description

Users can view their informations and change the password if they want. Users can also search and see history of movies they've purchased through the transactions history.

* + - 1. Functional Requirement

User can update their information password. Users can filter and search for their ticket in the transactions history.

* 1. User can manage booked tickets
     + 1. Description

The user can see the points and membership level in their deal. Users can know how many bonus points they have accumulated. Users also can view ticket details as well as cancel tickets.

* + - 1. Functional Requirement

- User must already have an existing account.

- User will see the points which they have accumulated..

* 1. Admin Can Manage Account
     + 1. Description

The site must allow admins to interact with Accounts. When a customer has registered an account, they will show up in customer list. An admin can view a list of existing registered customers

* + - 1. Function Requirement

- Must login as an admin to ban User

- Amin can be manager

- Admin can search customers by account and email

- Admin can ban User

* 1. Admin can view points and membership level
     + 1. Description

The site must allow admins to interact with membership level and points reward. An admin can view a list of existing membership level. Admin also can know the total number of user reward points.

* + - 1. Function Requirement

To access function, admin must provide:

-Must login as admin

- Admin can see membership level of the customer

- Admin can see the total point reward of the customer

* 1. Admin can manage ticket prices
     + 1. Description

The site must allow admins to interact with theaters and price tickets. An admin can view a list of existing theaters and ticket price in each theater. Admin modify a Theater’s ID, Theater’s Address, Theater’s Phone, Pictures about movie theater. Admin also can add movie will be show by theater and date. Admin can calculate the revenue of each theater.

* + - 1. Function Requirement

To add or modify Room and ticket prices, admin must provide:

- ID

- Room Address

- Film\_ ID

- User\_ID

- Status

- Price ticket

Other requirements:

- Must login as an admin to add or modify Room and ticket prices.

- Film\_ID is generated from a Film table

- User\_ ID is generated from the User table

- Admin can search room by ID

* 1. Admin Can Manage Reports
     + 1. Description.

The report displays a list of movies that have been booked, showing the number of customers who have booked tickets for the week and month. At the same time, it displays the revenue of each movie, the movies with the highest rating and the highest revenue of the week and month. Admin can export reports as a Excel file. Admin can also view the general report.

* + - 1. Function Requirement

- Must login as an admin to use report function

- Statistics of ticket sales.

- Food sales statistics.

- Movie revenue statistics.

- Total revenue statistics.

- See revenue chart.

* 1. Admin Can Manage Add Film.
     + 1. Description.

Admin page allows admin to add movies to the cinema, in addition, admin can edit, add or delete movies.

* + - 1. Function Requirement.

To add Film , admin must provide:

- Film\_ID

- Name\_film

- Image

- MetaTitle

- Director

- Actor

- Time

- ReleaseDate

- Country

- Vote

- AgeRestriction

- Description

- Trailer

- Status

1. SYSTEM REQUIREMENTS
   1. Hardware

|  |  |  |
| --- | --- | --- |
|  | SERVER | |
| Minimum | Recommended |
| Processor | 800 MHz Intel Pentium III | 2.6 GHz Intel Pentium IV |
| Memory | 2 GB | 4 GB |
| Disk space | 8 GB | 16 GB |
| Monitor | Super-VGA (800x600) or higher resolution monitor | |
| Internet | Internet access is required | |
|  | CLIENT | |
| Processor | 800 MHz Intel Pentium III | |
| Memory | 512 MB | |
| Monitor | Super-VGA (800x600) or higher resolution monitor | |
| Internet | Internet access is required | |

* 1. Software

|  |  |
| --- | --- |
|  | SERVER |
| Operating System | Windows 7 or later |
| Browser | Google Chrome 65.0 |
| Database | MySQL |
| Software |  |
|  |
|  | CLIENT |
| Operating System | Windows 7 or later |
| Browser | Google Chrome 65.0 |

* 1. DEVELOPMENT SOFTWARE

● Microsoft Windows 10

● MySQL

● Google Chrome 65.0

* 1. TECHNOLOGY

● HTML/CSS

● JavaScript

● AJAX

● Jquery

● PHP

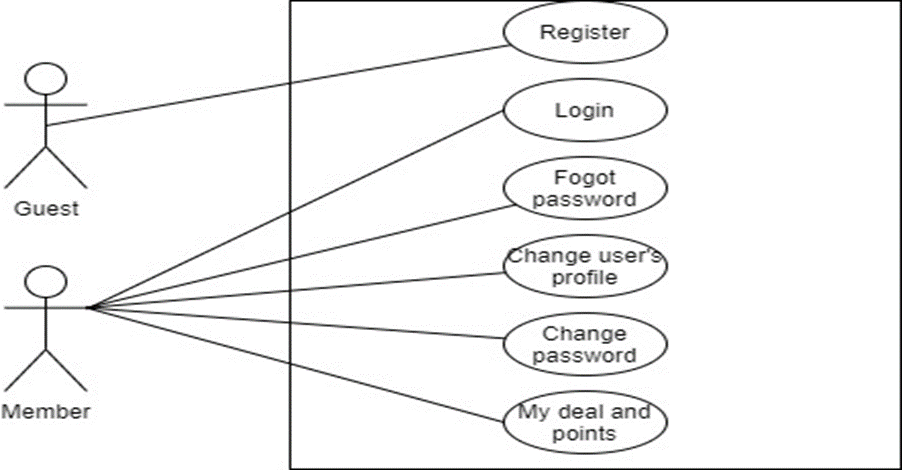
REVIEW 1 – TASK SHEET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Contents | Member | Activity Plan | Preparation | |
| Start Date | Actual Days | Status |
| 1 | Introduction | Le Tuan Xuyen  Tran Pham Thanh Hoang  Phung Van Phuc  Nguyen Nhat Nam | 06/12/2021 | 3 | Completed |
| 2 | Existing Scenario |
| 3 | Customer Requirement Specification |
| 4 | Functional Requirement Specification |
| 5 | System Requirements |
| 6 | Development Software |
| 7 | Technology |

Plan prepared by: Group 1 Tuesday June, 17th, 2021  
 **Group Leader Faculty**

**Le Tuan Xuyen Mr. Cat Nguyen Hieu Tuan**

REVIEW 2

1. Use Case and Sequence Diagram
   1. User
      1. Register as Member

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use Case Name** | Register as Member | |
| **Actors** | Guest | |
| **Description** | Guest registers an account on the website and becomes a member (member) | |
| **Requirements** | Guest provides their register information | |
| **Pre-conditions** | N/A | |
| **Post-conditions** | Success: Guest creates a Member account in this website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks ‘Register’ link on Home Page.            3. Actor inputs register’s information and click ‘Register’ button. | 2. System redirects to Login/Register page. User will click Register button and displays Registration form with the following controls: - Full Name text field - Account text field - Password text field  - Email text field - Date of Birth date field - Sex radio field - Register button  4. System validates the information. 5. System inserts the account into database  6. System show success message. |
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| - Actor left Full Name text field blank.  - Actor left Account text field blank. - Actor left ‘Phone number’ text field blank. - Input phone is not numeric format or phone length is not between and 12 characters. - Actor left Email text field blank.. - Actor left Password field blank.  - Actor left ‘Date of Birth’ text field blank.  - Actor left Sex text field blank. | System displays message to actor: - Full Name cannot be blank.  - Accountcannot be blank - Phone cannot be blank. - Email cannot be blank.  - Account is invalid or registered. - Password cannot be blank. - Date of Birth cannot be blank. - Sex cannot be blank - The phone number is invalid.  - Email is invalid |

* + 1. Login as Member

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use Case Name** | Login | |
| **Actors** | Member | |
| **Description** | Member who has registered an account can login | |
| **Requirements** | Actor provides account and password | |
| **Pre-conditions** | Actor must register an account | |
| **Post-conditions** | Success: User is logged in to website | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks on ‘Login’ button.     3. Actor inputs Account and Password, then click the ‘Login’ button. | 2. System redirects to Login/Register Page with the following controls - Account text field - Password text field - Login button 4. System checks the information. 5. System redirects to User’s dashboard. |
|
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid Account and password. | System redirects to Login page with the following controls: - Account text field  - Account invalid. System show message: This account does not exist. - Password text field.  - Password invalid. System show message: Incorrect Password. - Login button. |

* + 1. Forgot Password

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use Case Name** | Forgot Password | |
| **Actors** | Member | |
| **Description** | Member who has forgotten password can make a request to change password | |
| **Requirements** | Actor provides email address | |
| **Pre-conditions** | Actor must be logged on | |
| **Post-conditions** | Success: Actor can change their password | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks on ‘Forgot Password’ button.   3. Actor inputs Email and clicks ‘Submit’.  5. Actor check Email and login with the new password in Email .  7. Actor login with new password. | 2. System redirects to Password Page with the following controls: - 'Email' text field - 'Send' button 4. System checks the information and sends new password to user. 6. System redirects to login page:  - ‘Account’ text field  - ‘Password’ text field  - ‘Login’ button  8. System validates password then user login success. |
|
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs an email which does not exist in database. | System shows messages:  - ‘Invalid email’. |

* + 1. Change user’s profile

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use Case Name** | Change user’s profile | |
| **Actors** | Member | |
| **Description** | Member wants to change their profile’s information. | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Profile’s information of this user has been updated. | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks ‘Change user’s profile’ in the navbar              3. Actor updates information and click ‘Update’ button. | 2. System redirects to Update Profile page and displays Update Profile form with the following controls : - Full Name text field - Address text field  - Phone number text field - Birthday text field - Update button 4. System checks the information. 5. System updates the account in database. |
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Input phone is not in numeric or phone length is not in between 10 and 12 characters. | System displays message to actor :’  - ‘The phone number should be numeric only and must be between 10 and 12 digits’. |

* + 1. Change password

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | | Phung Van Phuc | |
| **Use Case Name** | | Change password | |
| **Actors** | | Member | |
| **Description** | | Member wants to change their password | |
| **Requirements** | | Actor has a account | |
| **Pre-conditions** | | Actor has logged in. | |
| **Post-conditions** | | Success: Update successful new password | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | | **System’s responses:** |
| 1. Actor clicks “Change password” in the navbar.  3. Actor change password and click ‘Change password’ button. | | 2. System redirects to Change password page with the following information:  - Old password field.  - New password field.  - Re-enter new password field.  4. System checks the information.  5.System updates the account in database. |
| **Basic flow** | | **Actor’s actions:** | **System’s responses:** |
| 1. Input old password isnot correct.  2. Re-enter new password isnot correct.  3. Actor left re-enter password field. | System displays message to actor :’  - ‘Your password isnot correct. Please enter again’.  - “Your re-enter password isnot correct.” |
|

* + 1. My deal and points

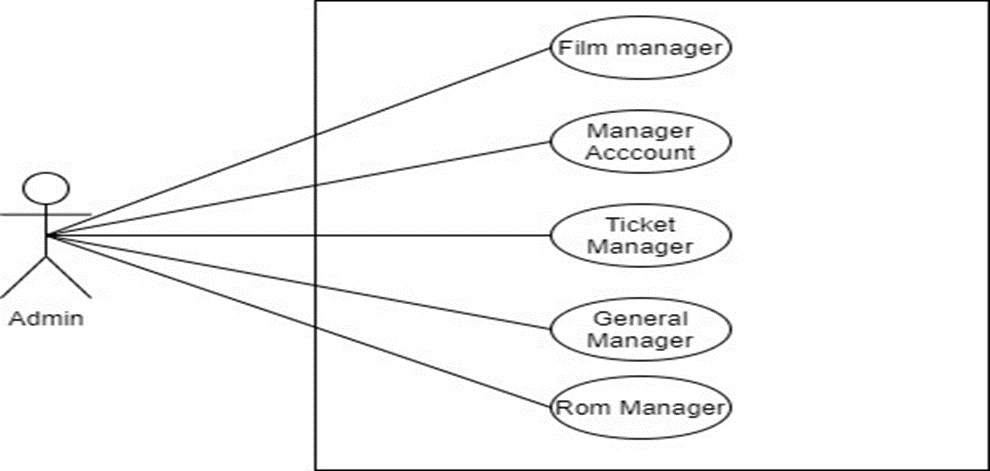
|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use Case Name** | My deal and points | |
| **Actors** | Member | |
| **Description** | Member wants to know about their deal and points | |
| **Requirements** | Actor has account | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: My deal and points will be display | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks ‘My deal and points’ in the navbar.          3. Actor can search their history deals and cancel their ticket. | 2. System redirects to My deal page and display to user with information:  - Movie ticket  - Number ticket  - Duration time  - Date booking  - Conditon  - Points  - Membership level  - Detail ticket 4. System checks the information. 5. System will show the information. |
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor enters values ​​that do not exist. | System displays message to actor : - No information to display in the table |

* + 1. Checkout and Reserve

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Checkout and Reserve | |
| **Actors** | Guest/Customer | |
| **Description** | Guest/Customer wants to order seats position and combo food | |
| **Requirements** | Actor must choose at least one product to add to cart | |
| **Pre-conditions** | Cart has at least one product | |
| **Post-conditions** | Success: Actor order successfully | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses a film and click ‘Book Ticket’ button.      6. Actor inputs Card Owner’s Name, Card Number and Security Code.   8. Actor views order’s status in Order Page. | 2.System adds Film by ID  3.System shows Ticket detail in time and date.  4.System information page and show the following controls:  - Billing Name text field.  - Row and Column’ text field.  - Required date and time.  - Food and Drink.  - Payment page.  5. Systems show Payment Page if user chooses Pay with Card method.  7. System validates information. |
|
|
|
|
|
|
|
|
| **Alternative flow** | 1. Actor clicks ‘X’ button on each product. | 2. System redirects to product page.  3. System removes product from cart. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor inputs invalid voucher code.  2. Actor inputs more products than there are in stock.  3. Actor left ‘Billing Name’ text field blank.  4. Actor left ‘Address’ text field blank.  5. Actor left ‘Required date’ blank.  6. Actor left ‘Payment’ checkboxes blank.  7. Actor inputs invalid card information. | System shows messages:  - ‘Invalid voucher’  - ‘Product out of stock’  - ‘Billing name must not be blank’  - ‘Address must not be blank’  - ‘Must choose a required date’  - ‘Must choose a payment method’  - ‘You are using an invalid card’ |

* + 1. Review film

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use Case Name** | Review Film | |
| **Actors** | User | |
| **Description** | User wants to leave review and rating on products | |
| **Requirements** | User provides their rating and review | |
| **Pre-conditions** | Actor is logged in and has bought product | |
| **Post-conditions** | Success: show actor's review on that product | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor clicks on a film to see an information.  3. Actor chooses rating score out of 5 stars and writes review content. | 2. System redirects to Film’s detail page.  4. System adds review to database and shows actor’s review in film’s detail page. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor does not rate product.  2. Actor does not write title.  3. Actor does not write content. | System displays messages:  - ‘You must rate before submit’  - ‘Review must have a title’  - ‘Content cannot be blank’ |

* 1. Admin
     1. Manager Account
        1. Update information Admin

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use case name** | Update information Admin | |
| **Actors** | Administrator | |
| **Description** | When admin wants to update information admin in the database | |
| **Requirements** | Actor provides admin information | |
| **Pre-conditions** | Actor is logged in as an Admin | |
| **Post-conditions** | Success: Admin information will be updated successfully | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses “Update information” in navbar  3. Actor inputs information to change | 2. System redirects to 'Update Amin' page and display the information  4. System displays admin form with the following fields. - Full Name text field - Address text field - Date of Birth date field - Phone number field  5. System validates inputs.  6. System updates admin or member in database.  7. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Input phone is not in numeric or phone length is not in between 10 and 12 characters. | System displays message to actor :.  - ‘The phone number should be numeric only and must be between 10 and 12 digits’. |
|
|

* + - 1. Delete User

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use case name** | Delete User | |
| **Actors** | Admin | |
| **Description** | When an admin wants to delete a account from the database | |
| **Requirements** | Actor provides account’s information | |
| **Pre-conditions** | Actor is logged in as an admin | |
| **Post-conditions** | Success:Account is deleted from the database | |
| Fail: Account is not deleted from the database | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Manager Account Tab on menu.  3. Actor clicks on ‘Delete’ button and confirm. | 2. System redirects to 'Manager Account' page and display a list of existing accounts.  4. System delete account from database.  5. System displays confirmation message or error message. |

* + - 1. Ban User

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use case name** | Ban an account | |
| **Actors** | Administrator | |
| **Description** | When admin wants to ban an account | |
| **Requirements** | Actor provides account’s information | |
| **Pre-conditions** | Actor is logged in as admin | |
| **Post-conditions** | Success: set account’s status into “Locked” and updated in database | |
| Fail: No changes are made | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Account row on Customer list and click “Uncloked” button to locked account. | 2. System updates customer’s status in database.  3. System displays confirmation message or error message. |

* + - 1. Add admin account

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use case name** | Add admin’s accounts | |
| **Actors** | Administrator | |
| **Description** | When admin wants to add a new admin’s account in the database | |
| **Requirements** | Admin provides admin information | |
| **Pre-conditions** | Actor is logged in as an Admin | |
| **Post-conditions** | Success: a new admin’s account is created in the database | |
| Fail: Refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Manager Account Tab on menu.  3. Actor chooses ‘Add’ button.  5. Actor inputs information to create admin. | 2. System redirects to 'Manager Account' page and display a list of existing admin account.  4. System displays admin form with the following fields. - Full Name text field - Account text field - Password text field  - Email text field  - Address text field  - Phone number - Date of Birth date field - Sex radio field  6. System validates inputs.  7. System adds new admin to database.  8. System displays confirmation message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| - Actor left ‘Name’ text field blank.  - Actor left ‘Phone number’ text field blank.  - Input phone is not numeric format or phone length is not between 10 and 12 characters.  - Actor left ‘Email’ text field blank.  - Email was registered or input an invalid email address.  - Actor left ‘Password’ field blank..  - Actor left ‘Date of Birth’ text field blank.  - Actor left ‘Gender’ text field blank. | System displays messages: - ‘Full Name cannot be blank’.  - ‘Phone cannot be blank’.  - ‘Email cannot be blank’.  - ‘Account cannot blank’  - ‘Password cannot be blank’.  - ‘Date of Birth cannot be blank’’.  - ‘Sex cannot be blank’’  - ‘The phone number should be numeric only and must be between 10 and 12 digits’. |
|
|

* + 1. Membership level and Points
       1. View membership level and Points

|  |  |  |
| --- | --- | --- |
| **Author** | Phung Van Phuc | |
| **Use case name** | View membership level and Points | |
| **Actors** | Administrator | |
| **Description** | When admin wants to view point with member card | |
| **Requirements** | N/A | |
| **Pre-conditions** | Actor is logged in as admin | |
| **Post-conditions** | Success: Display membership level and Points | |
| **Basic flow** | **Actor’s actions:** | **System’s responses** |
| 1. Actor chooses ‘Manager User’s account’ tab | 2. System redirects to manager user account page.Admin wil see total point, membership level in each row |

* + 1. Room
       1. Add new Room

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Nhat Nam | |
| **Use case name** | Add new Room | |
| **Actors** | Administrator | |
| **Description** | When admin wants to add new Room | |
| **Requirements** | Actor provides Room’s information | |
| **Pre-conditions** | Actor is logged in as admin | |
| **Post-conditions** | Success: a new cinema added in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Manager Room Tab on menu.  4. Actor inputs information to create new cinema and click “Add” button. | 2. System redirects to “Manager Room '' page.  3. System displays a form with the following fields:  - Room\_ID  - Name’s Room  - Level Room  - Column Room  - Seat’s Room  - Row Room  5. System validates inputs.  6. System adds card into the database.  7. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left Name’s Room field blank.  2. Actor left Level Room field blank.  3. Actor left ‘Seat’ field blank. Seat is numeric | System displays messages: - Name’s Room cannot be blank.  - Level Room cannot be blank  - Seat’s cannot be blank’. Seat must be numeric |
|
|

* + - 1. Update the Room

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Nhat Nam | |
| **Use case name** | Update the Room | |
| **Actors** | Administrator | |
| **Description** | When admin wants to update the Room | |
| **Requirements** | Actor provides collection’s information | |
| **Pre-conditions** | Actor is logged in as admin | |
| **Post-conditions** | Success: Room is updated in database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Manager Room row on Room list and click “Update” button.  4. Actor inputs information to update card and click “Save” button. | 2. System redirects to 'Update ' page.  3. System displays a form with the following fields. - Room\_ID (read only)  - Name’s Room  - Level Room  - Column Room  - Seat’s Room  - Row Room  6. System validates inputs.  7. System updates collection in database.  8. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor input not numeric in seat field | System displays messages: - ‘Seats must be numeric’ |
|
|

* + - 1. Delete the Room

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Nhat Nam | |
| **Use case name** | Delete the Room | |
| **Actors** | Admin | |
| **Description** | When admin wants to delete the Room from the database | |
| **Requirements** | Actor provides Room’s information | |
| **Pre-conditions** | Actor is logged in as an admin | |
| **Post-conditions** | Success: Room is deleted from the database | |
| Fail: card is not deleted from the database | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Manage Room row on Room list and click “Delete” button. | 2. System deletes Room from database.  3. System displays confirmation message. |

* + 1. Pages films
       1. Add films

|  |  |  |
| --- | --- | --- |
| **Author** | Tran Pham Thanh Hoang | |
| **Use case name** | Add Films | |
| **Actors** | Administrator | |
| **Description** | When admin wants to add a new films in the database | |
| **Requirements** | Provide information of the movie | |
| **Pre-conditions** | You must first log in as an admin | |
| **Post-conditions** | Success: Films is added in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Pages tab on menu.  3. Actor clicks ‘Add Films’.  5. Enter information. | 2. System redirects to 'Add films' page and display a list of existing Pages.  4. System displays a form with the following fields. - Film name  - Meta\_Title - Image  - Contents  - Director  - Actor  - Time  - ReleaseDate  - Country  - Vote  - AgeRestriction  - Trailer  - Status    6. System validates inputs.  7. System add films into database.  8. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Film’s Name’ field blank.  2. Film ’s name existed in database. | System displays messages: - Film’s Name cannot be blank. - Film’s Name must be unique.  - Image cannot blank  - Content cannot blank  - Trailer cannot blank |

* + - 1. List of Films

|  |  |  |
| --- | --- | --- |
| **Author** | Tran Pham Thanh Hoang | |
| **Use case name** | List of Films | |
| **Actors** | Administrator | |
| **Description** | When the administrator wants to see the list of films saved in the database | |
| **Requirements** | Provide information of the movie | |
| **Pre-conditions** | You must first log in as an admin | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Pages tab on menu.  3. Actor clicks ‘Add Films’.  5. Enter information. | 2. System redirects to 'Show List Films' page and display a list of existing Pages.  4. System displays a form with the following fields. - Film name - Show date  - Image Film  - Trailer  - Action  6. System validates inputs.  7. System add films into database.  8. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Film’s Name’ field blank.  2. Film ’s name existed in database. | System displays messages: - Film’s Name cannot be blank. - Film’s Name must be unique.  -Image cannot blank  -Content cannot blank |

* + - 1. Update Film

|  |  |  |
| --- | --- | --- |
| **Author** | Tran Pham Thanh Hoang | |
| **Use case name** | Update Films | |
| **Actors** | Administrator | |
| **Description** | When admin wants to update a category | |
| **Requirements** | Actor provides Films information | |
| **Pre-conditions** | Actors must first log in as an admin | |
| **Post-conditions** | Success: Films is updated in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Pages Tab on menu.  3. Actor clicks “Update” button .  5. Enter information. | 2. System redirects to 'List of Films’page and display a list of existing Pages.  4. System displays a form with the following fields. - Film name  - Meta\_Title - Image  - Contents  - Director  - Actor  - Time  - ReleaseDate  - Country  - Vote  - AgeRestriction  - Trailer  - Status  6. System validates inputs.  7. System updates films in database  8. System displays confirmation message or error message. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Film’s Name’ field blank.  2. Film ’s name existed in database. | System displays messages: - Film’s Name cannot be blank. - Film’s Name must be unique.  -Image cannot blank  -Content cannot blank  -Trailer cannot blank |
|
|

* + - 1. Delete Film

|  |  |  |
| --- | --- | --- |
| **Author** | Tran Pham Thanh Hoang | |
| **Use case name** | Delete Film | |
| **Actors** | Administrator | |
| **Description** | When a super admin wants to delete a film from the database | |
| **Requirements** | Actor provides Films information | |
| **Pre-conditions** | Actors must first log in as an admin | |
| **Post-conditions** | Display the message, do you want to delete this movie?  -Success: Film is delete form the database  -Fail: Not Delete film | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Pages Tab on menu.  3. Actor clicks on ‘Delete’ button and confirm. | 2. System redirects to 'Show List films' page and display a list of existing Pages.  4. System delete Films from database.  5. System displays confirmation message or error message. |

* + 1. Combo Food
       1. Create new Product (with SKUs and Images)

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use case name** | Create New **Combo Food** | |
| **Actors** | Administrator | |
| **Description** | When admin wants to create a new product | |
| **Requirements** | Actor provides product’s information | |
| **Pre-conditions** | Actor is logged in as admin or super admin | |
| **Post-conditions** | Success: new product with SKUs and images is added into the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks ‘Add New Product’ button.  5. Actor inputs information to create product. | 2. System redirects to ‘Product’ page and display a list of existing products.  4. System displays a form with the following fields. - Product’s Name  - Product’s Description  - Collection  - Category  - Discount Value  - Discount Valid dates  - List of attributes  6. System validates inputs.  7. System adds product into database.  8. System displays a form for adding SKUs and Images for this product.  9. System displays confirmation message or error message. |
|
|
|
|
|
|
|
|
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Product Name’ blank.  2. Actor left ‘Product Description’ blank.  3. Actor does not choose a Category.  4. Actor choose invalid Discount Dates  5. Actor does not add any Sku  6. Actor does not add any Image  7. Sku ID existed in database | System displays messages: - ‘Product Name cannot be blank’  - ‘Description cannot be blank’  - ‘Must choose a Category’  - ‘Must choose valid Discount Date’  - ‘Product must have Sku and Image’  - ‘Sku ID must be unique’ |
|
|

* + - 1. Update Combo Food

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use case name** | Update product’s details | |
| **Actors** | Administrator | |
| **Description** | When admin wants to update product’s details | |
| **Requirements** | Actor provides product’s information | |
| **Pre-conditions** | Actor is logged in as admin or super admin | |
| **Post-conditions** | Success: product is updated in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks on ‘View Details’ on a product.  5. Actor chooses the criteria they want to update and input information. | 2. System redirects to ‘Product Details’ page and displays a list of existing products.  4. System displays product’s detail where actor can update the following information: - Product’s Name  - Product’s Description  - Collection  - Category  - Discount Value  - Discount Valid dates  - List of attributes  - Product Status  - Add/Delete Sku  - Add/Delete Images  6. System validates inputs.  7. System updates product in the database.  8. System displays confirmation message or error message.  9. System redirects to product’s detail page. |
|
|
|
|
|
|
|
|
|
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Product Name’ blank.  2. Actor left ‘Product Description’ blank.  3. Actor does not choose a Category.  4. Actor choose invalid Discount Dates  5. Sku ID existed in database | System displays messages: - ‘Product Name cannot be blank’  - ‘Description cannot be blank’  - ‘Must choose a Category’  - ‘Must choose valid Discount Date’  - ‘Product must have Sku and Image’  - ‘Sku ID must be unique’ |
|
|

* + - 1. Delete Combo Food

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use case name** | Update product’s details | |
| **Actors** | Administrator | |
| **Description** | When Administrator wants to delete a Sku from the database | |
| **Requirements** | Actor provides product’s information | |
| **Pre-conditions** | Actor is logged in as admin or super admin | |
| **Post-conditions** | Success: product is deleted in the database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses Product Tab on menu.  3. Actor clicks ‘Product Detail’ button.  5. Actor clicks ‘Delete’ in Sku List and clicks confirm. | 2. System redirects to 'Product List' page and displays a list of existing products.  4. System displays Product’s detail page.  8. System validates inputs.  9. System deletes sku in the database.  10. System displays confirmation or error. |

* + 1. Comment
       1. Response to Comment

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use case name** | Response comment of customer | |
| **Actors** | Administrator | |
| **Description** | When admin wants to response review of customer | |
| **Requirements** | Actor provides customer’s review information | |
| **Pre-conditions** | Actor is logged in as admin or super admin | |
| **Post-conditions** | Success: response content is updated in database | |
| Fail: refill information | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses comment row on comment list and click “Response” button.  4. Actor input content into “admin response” text field and click “Save” button. | 2. System redirects to 'Response comment page and displays review of customer.  3. System displays a form with the following fields. - Customer ID (read only)  - Product ID (read only)  - Title (read only)  - Content (read only)  - Rating (read only)  - Admin response  5. System validates inputs.  6. System updates comment in database.  7. System redirects to 'Review list' page. |
| **Exceptions** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor left ‘Admin response’ field blank. | System displays messages: - ‘Response cannot be blank.’ |
|

* + - 1. Delete Comment

|  |  |  |
| --- | --- | --- |
| **Author** | Le Tuan Xuyen | |
| **Use case name** | Delete **comment** | |
| **Actors** | Super admin | |
| **Description** | When super admin wants to delete a comment of customer from the database | |
| **Requirements** | Actor provides customer’s review information | |
| **Pre-conditions** | Actor is logged in as super admin | |
| **Post-conditions** | Success: comment is deleted from the database | |
| Fail: comment is not deleted from the database | |
| **Basic flow** | **Actor’s actions:** | **System’s responses:** |
| 1. Actor chooses review row on review list and click “Delete” button. | 2. System deletes comment from database.  3. System displays confirmation message. |

1. Table Design
   1. User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: User** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT  IDENTITY(11) | NOT NULL PRIMARY KEY | Stores the ID of account |
| 2 | Account | VARCHAR(20) | NOT NULL | Stores the Name of account and member |
| 3 | Password | VARCHAR(20) | NOT NULL | Stores the Password of member and admin |
| 4 | Email | VARCHAR(255) | NOT NULL | Stores the Email of member and admin |
| 5 | FullName | VARCHAR(255) | NOTNULL | Stores the FullName of member and admin |
| 5 | Phone | VARCHAR(255) | NOT NULL | Stores the Phone of member and admin |
| 6 | Birth Day | DATETIME | NOT NULL | Stores the Birthday of member and admin |
| 7 | Gender | VARCHAR(20) | NOT NULL | Stores the gender of member and admin |
| 8 | Address | VARCHAR(255) | NOT NULL | Stores the Address of member and admin |
| 9 | Type | INT(11) | NOT NULL | Stores the Type of member and admin |
| 10 | Status | TINYINT(1) | NOT NULL | Stores the Status of member and admin |

* 1. Membership

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Membership** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of member card |
| 2 | User\_ID | INT(11) | NOT NULL | Stores ID code User |
| 3 | Name | VARCHAR(255) | NOT NULL | Stores the register Name of card |
| 4 | Point | INT(11) | NOT NULL | Stores the level Point of card |

* 1. Reply

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Reply** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of Reply |
| 2 | Content | Text | NOT NULL | Stores Content of reply |
| 3 | CreateDate | DATETIME | NOT NULL | Stores the Create Date |
| 4 | User\_ID | INT(11) | NOT NULL | Stores ID code User |
| 5 | Comment\_ID | INT(11) | NOT NULL | Stores ID code Comment |

* 1. Food and Drink

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Food and Drink** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Name | VARCHAR(255) | NOT NULL | Stores the Name of product |
| 3 | Image | VARCHAR(255) | NOT NULL | Stores the Image of product |
| 4 | Price | DECIMAL(65,0) | NOT NULL | Stores the Price of product |
| 5 | Descirption | TEXT | NOT NULL | Stores the Description product |

* 1. Add Films

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Add Films** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Name | VARCHAR(255) | NOT NULL | Stores the Films Name |
| 3 | Metatitle | VARCHAR(255) | NOT NULL | Stores the Meta Title |
| 4 | Image | VARCHAR(255) | NOT NULL | Stores the Image of Film |
| 5 | Director | VARCHAR(255) | NOT NULL | Stores the Director of Film |
| 6 | Actor | VARCHAR(255) | NOT NULL | Stores the Actor of Film |
| 7 | Time | VARCHAR(255) | NOT NULL | Stores the Time of Film |
| 8 | ReleaseDate | DATETIME | NOT NULL | Stores the Date of Film |
| 9 | Country | VARCHAR(255) | NOT NULL | Stores the Country of Film |
| 10 | Vote | FLOAT | NOT NULL | Stores the Vote of Film |
| 11 | AgeRestriction | INT(11) | NOT NULL | Stores the Age Restriction of Film |
| 12 | Description | TEXT | NOT NULL | Stores the Description of Film |
| 13 | Trailer | VARCHAR(255) | NOT NULL | Stores the Trailer of Film |
| 14 | Status | TINYINT(1) | NOT NULL | Stores the Status of Film |

* 1. Book Ticket

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Book Ticket** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Date | VARCHAR(255) | NOT NULL | Stores movie ticket booking Date |
| 3 | Time | VARCHAR(255) | NOT NULL | Stores movie ticket booking Time |
| 4 | CreatedDate | DATETIME | NOT NULL | Stores the Creat Date of film |
| 5 | Seat | VARCHAR(255) | NOT NULL | Stores movie ticket booking Seat |
| 6 | Count\_Ticket | INT(11) | NOT NULL | Store discount codes if avaiable |
| 7 | Total\_Price | DECIMAL(8,2) | NOT NULL | Stores movie ticket booking Total Price |
| 8 | User\_ID | INT(11) | NOT NULL | Stores ID code User |
| 9 | Film\_ID | INT(11) | NOT NULL | Stores ID code Film |

* 1. Book Seat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Book Seat** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Seat | VARCHAR(255) | NOT NULL | Stores the Seat when booking ticket |
| 3 | Type | INT(11) | NOT NULL | Stores the Type |
| 4 | Count | INT(11) | NOT NULL | Store the Count of seats |
| 5 | Price | DECIMAL(65,0) | NOT NULL | Store price |
| 6 | Total\_Money | DECIMAL(65,0) | NOT NULL | Store Total of Money |
| 7 | BookTicket\_ID | INT(11) | NOT NULL | Stores ID code BookTicket |
| 8 | RoomDetail\_ID | INT(11) | NOT NULL | Stores ID code RoomDetail |

* 1. Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Category** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Name | VARCHAR(255) | NOT NULL | Stores the movies type |
| 3 | Link | VARCHAR(255) | NOT NULL | Stores the movies link |
| 4 | DisPlay\_Ỏrder | VARCHAR(255) | NOT NULL | Stores the movies Display Order |
| 5 | Status | TINYINT(1) | NOT NULL | Stores the movies Status |

* 1. Category Film

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Category Film** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Film\_ID | INT(11) | NOT NULL | Stores ID code Film |
| 3 | Category\_ID | INT(11) | NOT NULL | Stores ID code Category |

* 1. Comment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Comment** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Content | TEXT | NOT NULL | Archive user comments about the movie |
| 3 | Rate | INT(11) | NOT NULL | Archive user Rating about the movie |
| 4 | User\_ID | INT(11) | NOT NULL | Stores ID code User |
| 5 | Film\_ID | INT(11) | NOT NULL | Stores ID code Film |

* 1. Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Room** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Name | VARCHAR(255) | NOT NULL | Stores the movie room name |
| 3 | Status | TINYINT(1) | NOT NULL | Stores the Room status |

* 1. Room Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Room Detail** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Level | INT(11) | NOT NULL | Stores the movie room name |
| 3 | Row | INT(11) | NOT NULL | Stores Row in room |
| 4 | Column | INT(11) | NOT NULL | Stores Column in room |
| 5 | Price | DECIMAL(8,2) | NOT NULL | Stores the price of each seat |
| 6 | Room\_ID | INT(11) | NOT NULL | Stores ID code Room |

* 1. Book Food and Drink

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Book\_FD** | | | | |
| **No.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| 1 | ID | INT(11) | NOT NULL PRIMARY KEY | Stores the ID of AUTO\_INCREMENT |
| 2 | Quantity | INT(11) | NOT NULL | Stores Quantity |
| 3 | Total\_Price | DECIMAL(8,2) | NOT NULL | Stores Total Prices |
| 4 | BookTicket\_ID | INT(11) | NOT NULL | Stores ID code BookTicket |
| 5 | FoodDrink\_ID | INT(11) | NOT NULL | Stores ID code |

1. Entity Properties
   1. User

Diagram

Description automatically generated

* 1. Membership

Diagram

Description automatically generated

* 1. Reply

Diagram

Description automatically generated

* 1. Food and Drink

**Diagram

Description automatically generated**

* 1. Add Films

Diagram

Description automatically generated

* 1. Book Ticket

Diagram

Description automatically generated

* 1. Book Seat

Diagram

Description automatically generated

* 1. Category

Diagram

Description automatically generated

* 1. Category Film

Diagram

Description automatically generated

* 1. Comment

Diagram

Description automatically generated

* 1. Room

Diagram

Description automatically generated

* 1. Room Detail

Diagram

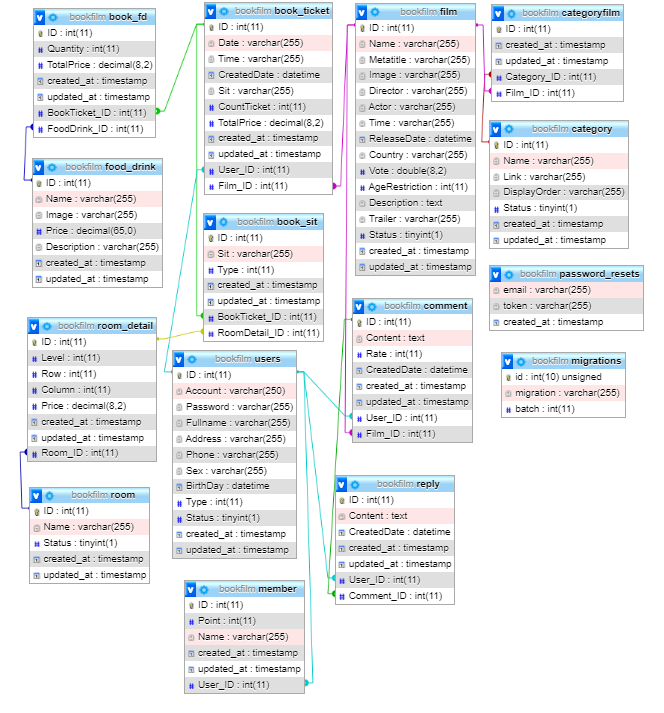
Description automatically generated

* 1. Book Food and Drink

Diagram

Description automatically generated

1. Entity Relationship Diagram (ERD)
   1. Physical ERD



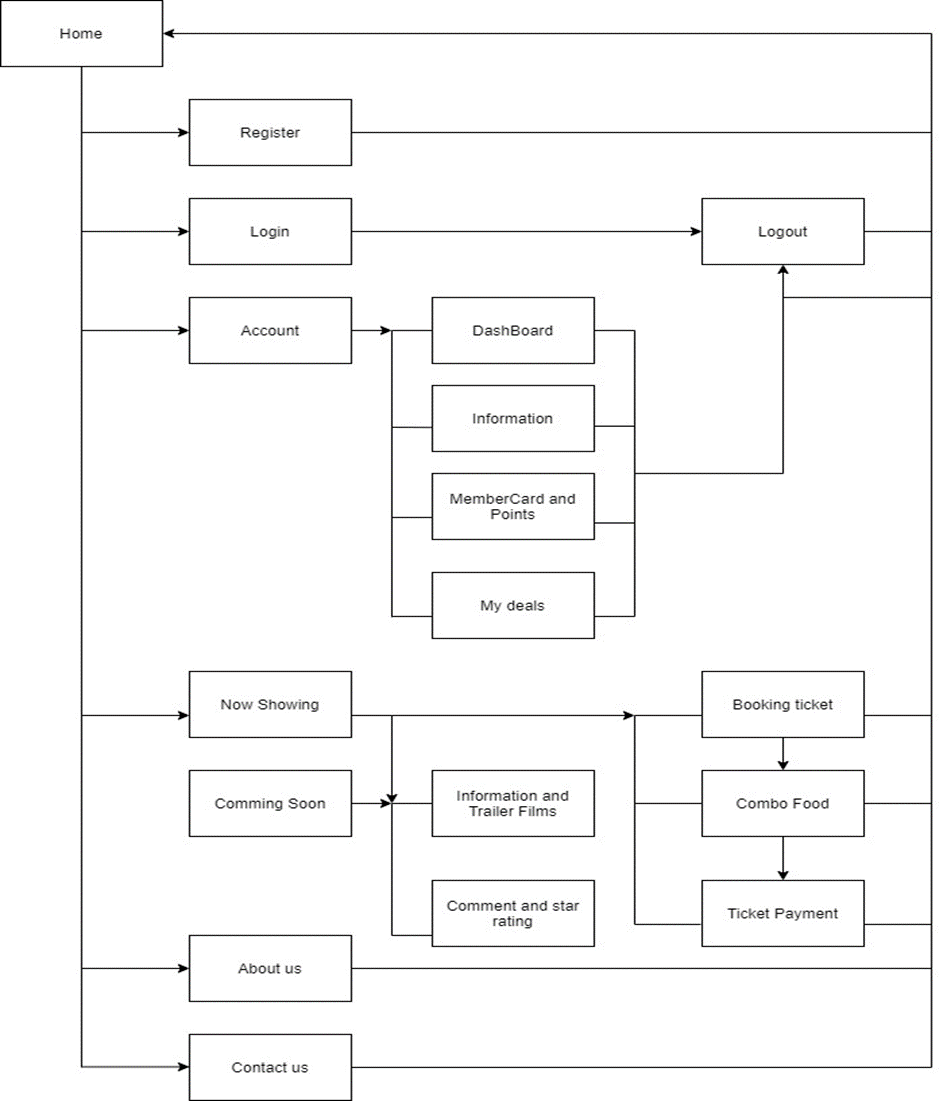
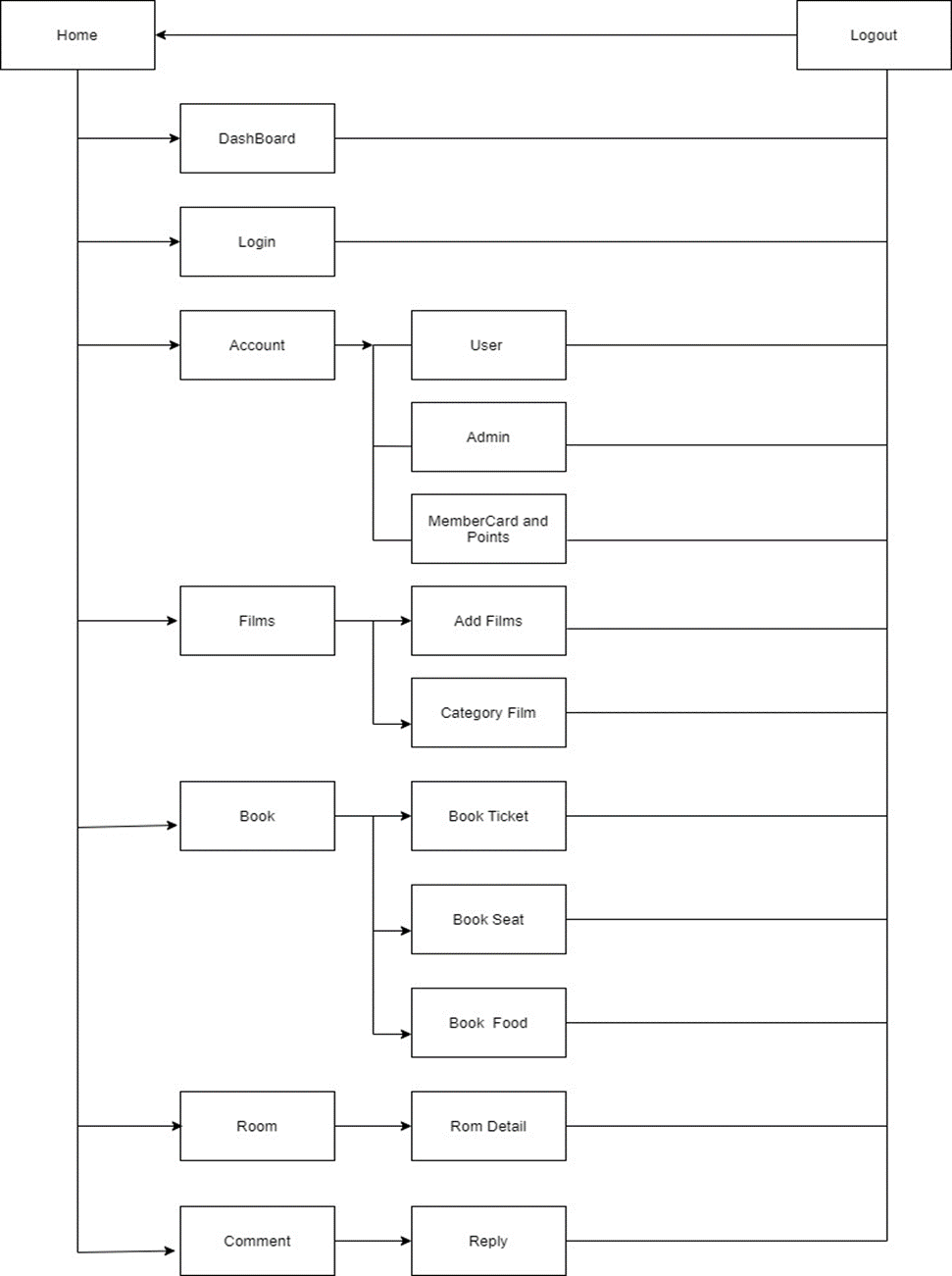
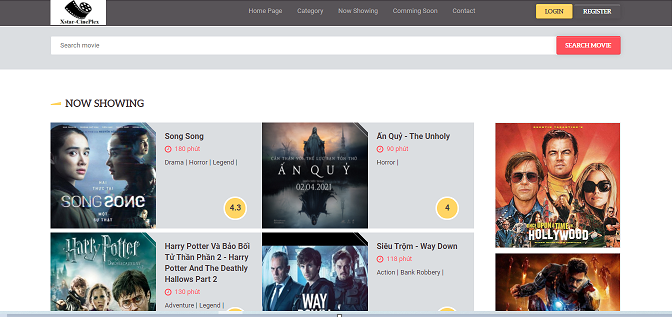
REVIEW 2 – TASK SHEET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Contents | Member | Activity Plan | Preparation | |
| Start Date | Actual Days | Status |
| 1 | Use Case and Sequence Diagram | Le Tuan Xuyen  Tran Pham Thanh Hoang  Nguyen Nhat Nam  Phung Van Phuc | 15/06/2018 | 5 | Completed |
| 2 | Entity Relationship Diagram (ERD) |
| 3 | Entity Properties |
| 4 | Table Design |

Plan prepared by: Group 1 June 22st, 2021  
 **Group Leader Faculty**

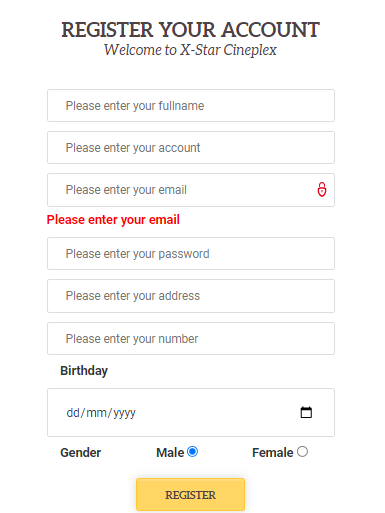
**Le Tuan Xuyen Mr. Cat Nguyen Hieu Tuan**

REVIEW 3

1. Site Map
   1. Client’s Site Map
   2. Admin’s Site Map
2. GUI Design
   1. Client’s Side
      1. Home

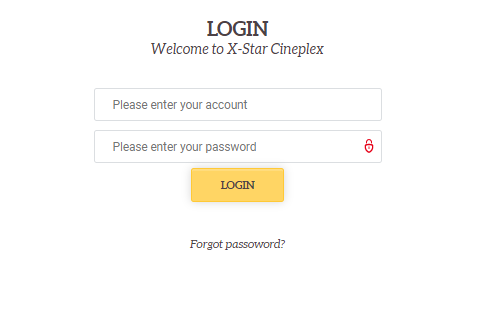
|  |  |
| --- | --- |
| Description | Home page |
| Connect From | N/A |
| Connect To | Login page, Product page, Profile, Cart, Checkout, Search |
| Validation | N/A |
| Initial Values | N/A |
| Events | - Navigate the website by clicking on menu items or products. |

* + 1. Register



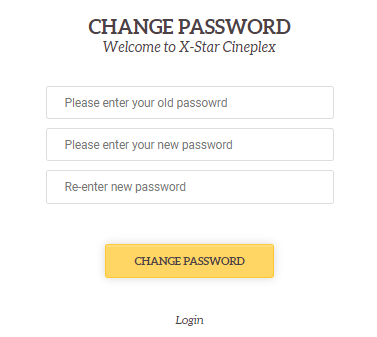
|  |  |
| --- | --- |
| Description | Guest signs up for new account |
| Connect From | Home |
| Connect To | Login page, Home |
| Validation | - All fields: not blank  Fullname: "Please enter your first and last name",  Account: "Please enter your account",  Email: "Please enter your email"  Password: "Please enter a password",  Address: "Please enter your address",  BirthDay: "Please enter your date of birth",  Password: Please enter the phone number  - Email must be in the correct format  - Phone number must be in correct format |
| Initial Values | N/A |
| Events | - When click button "Register", check all fields. If success, go to login page |

* + 1. Login

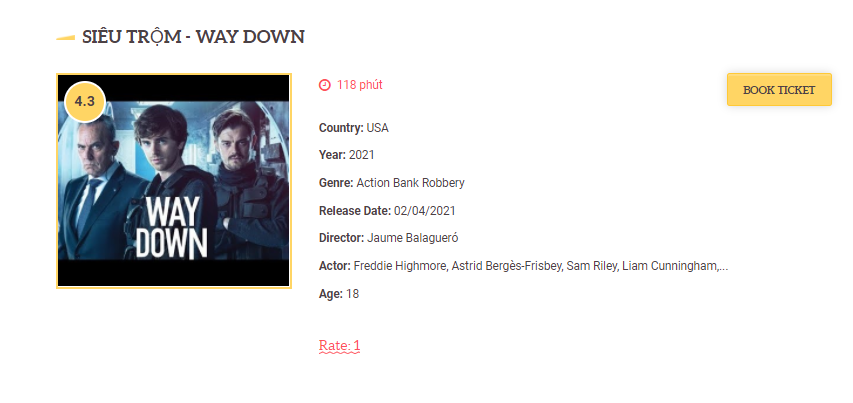


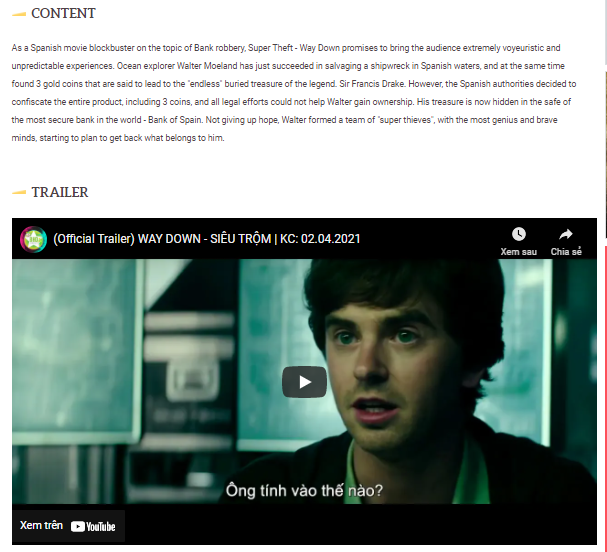
|  |  |
| --- | --- |
| Description | Login with account and password |
| Connect From | Home |
| Connect To | User’s profile, Forget password, Admin pages |
| Validation | All fields: not blank -Name: Please enter your account, - Password: Please enter your password,  - Account & Password must be the same with database |
| Initial Values | N/A |
| Events | - When click button "Login", check account and password, then login. Login success, go to profile page. If account is admin, website will go to management page. - When click "Forgot Your Password?" link, go to Forgot password page |

* + 1. Change Password



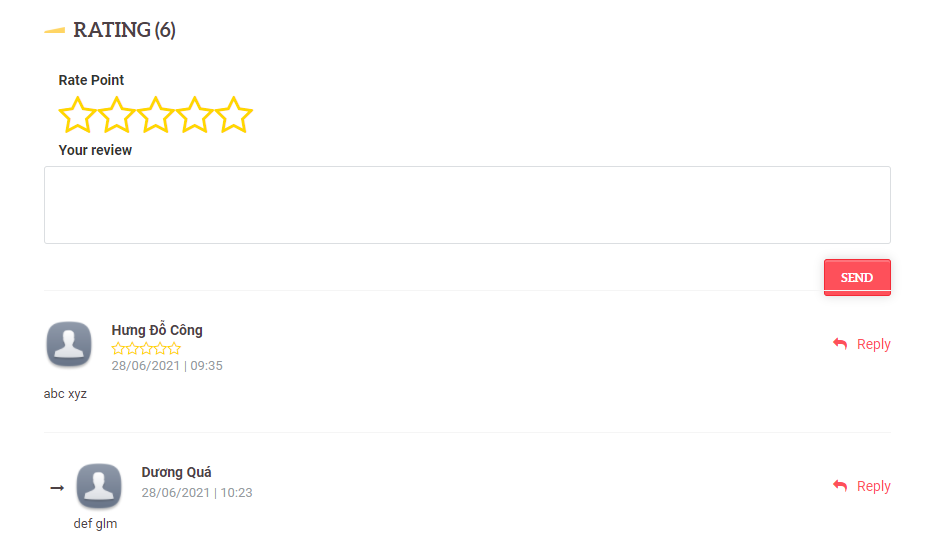
|  |  |
| --- | --- |
| Description | Change password |
| Connect From | Home |
| Connect To | User’s profile, Change password |
| Validation | All fields: not blank -Old Password: Please enter your old password, - New Password: Please enter your new password,  -Re-enter Password: Re-enter new password, |
| Initial Values | N/A |
| Events | - When click button "Change PassWord", check old password and new password, then update succesful. |

* + 1. Movie description and trailer



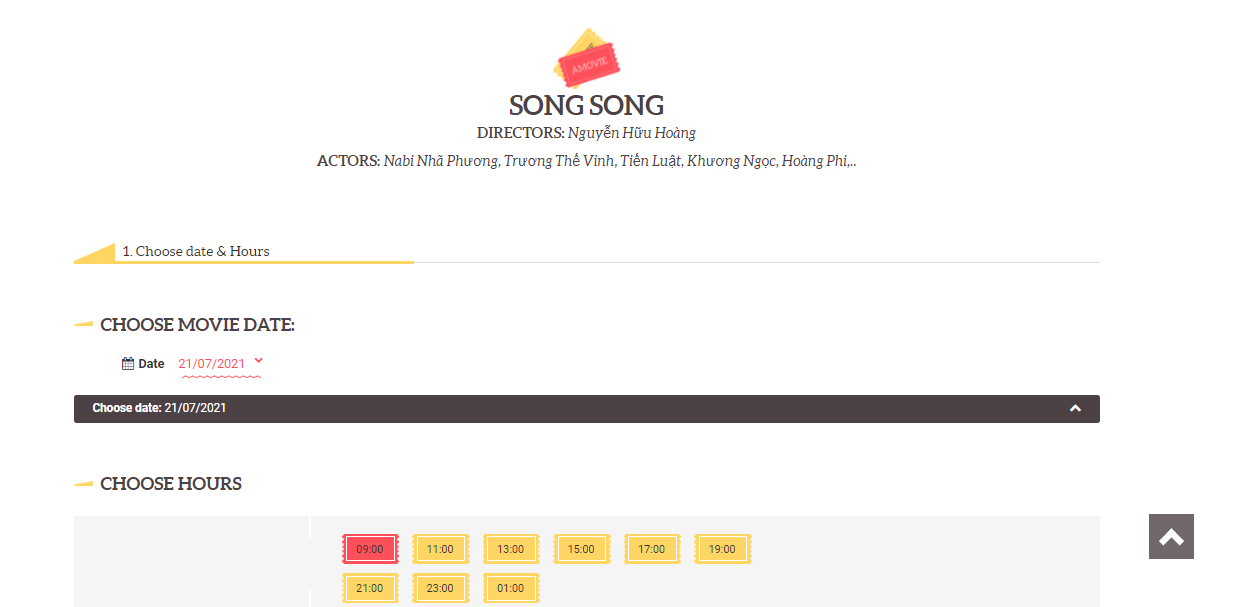
|  |  |
| --- | --- |
| Description | Show detail and description’s movie |
| Connect From | Home |
| Connect To | Cart, share wish list |
| Validation | Customer is logged |
| Initial Values | List of bookmarked movie |
| Events | - When click button cart it show detail and trailer’s movie |

* + 1. Raiting and comment



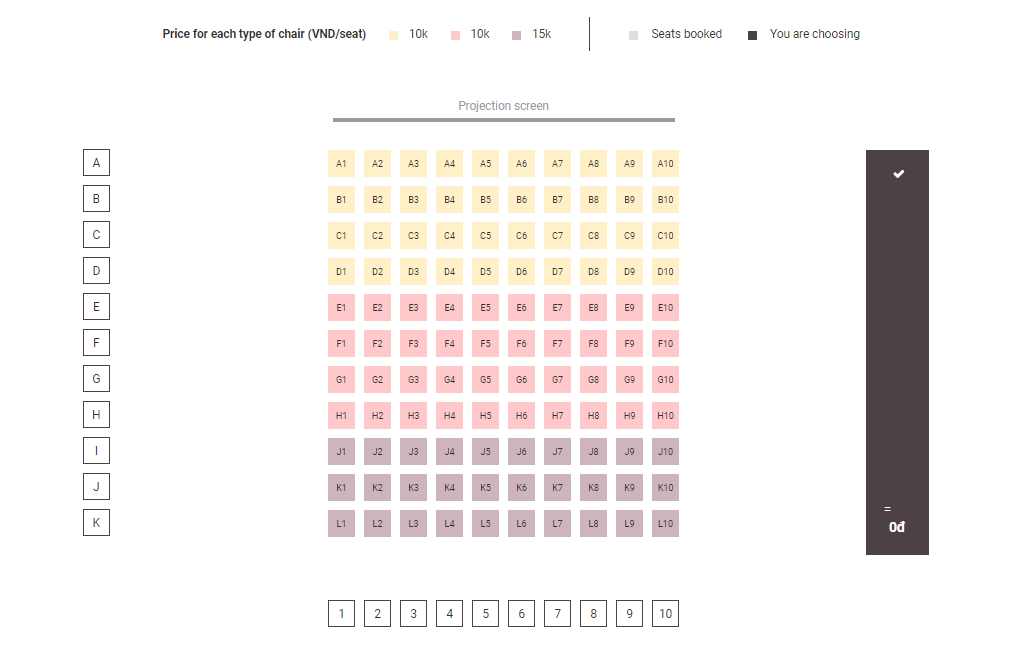
|  |  |
| --- | --- |
| Description | Show comment form and rating with 5 star |
| Connect From | Home |
| Connect To | Reserve Seat |
| Validation | Customer is logged |
| Initial Values | List of comment form |
| Events | - When user enter comment and click button comment and click star symbol to rating |

* + 1. Choose time



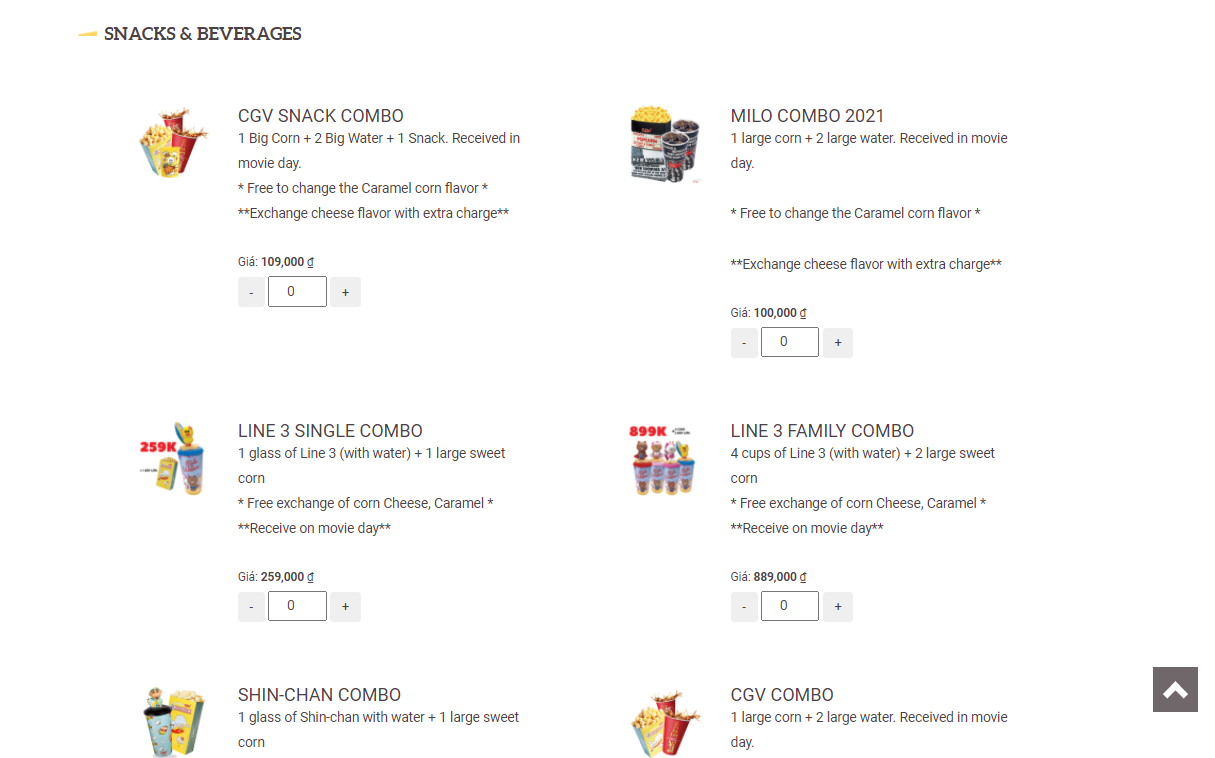
|  |  |
| --- | --- |
| Description | Show time information of the movie |
| Connect From | Home |
| Connect To | Reserve Seat |
| Validation | Customer is logged |
| Initial Values | List of time |
| Events | - When user click button time |

* + 1. Choose seat position



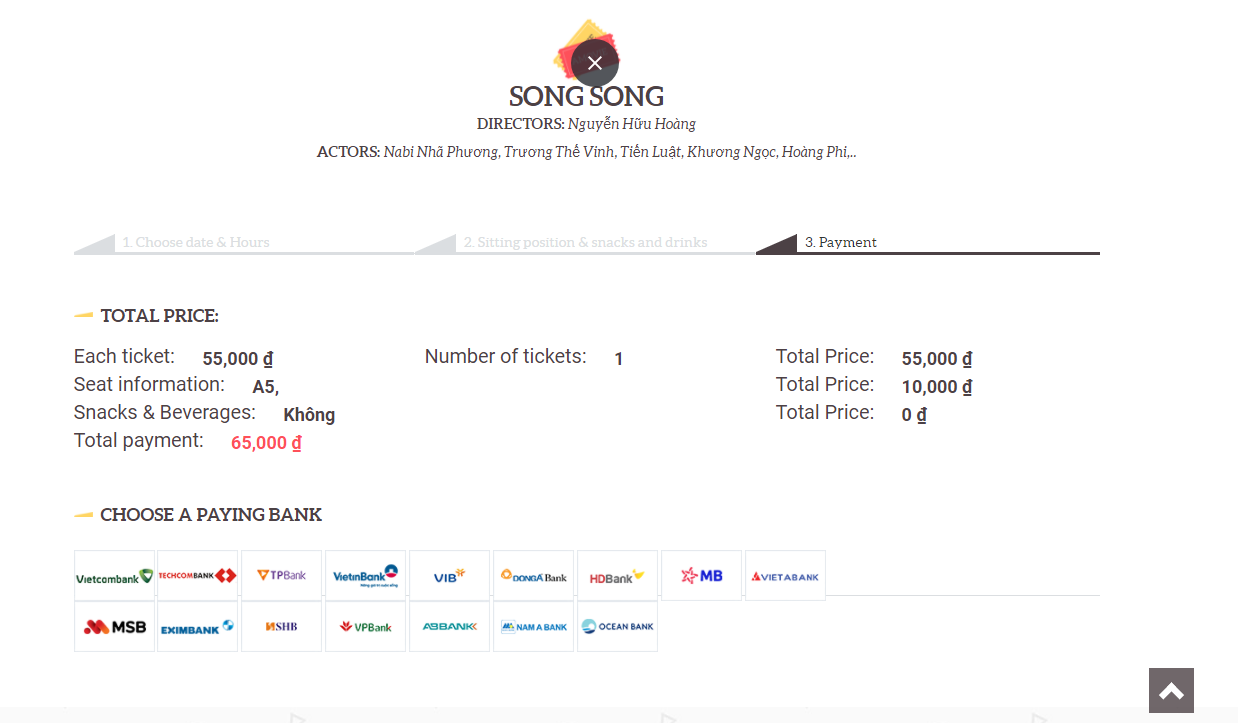
|  |  |
| --- | --- |
| Description | Show organizational chart |
| Connect From | Home |
| Connect To | Payment pay |
| Validation | Customer is logged |
| Initial Values | List of time |
| Events | - When user click seat on organizational chart  - User can select multiple seats at the same time |

* + 1. Combo food and drink



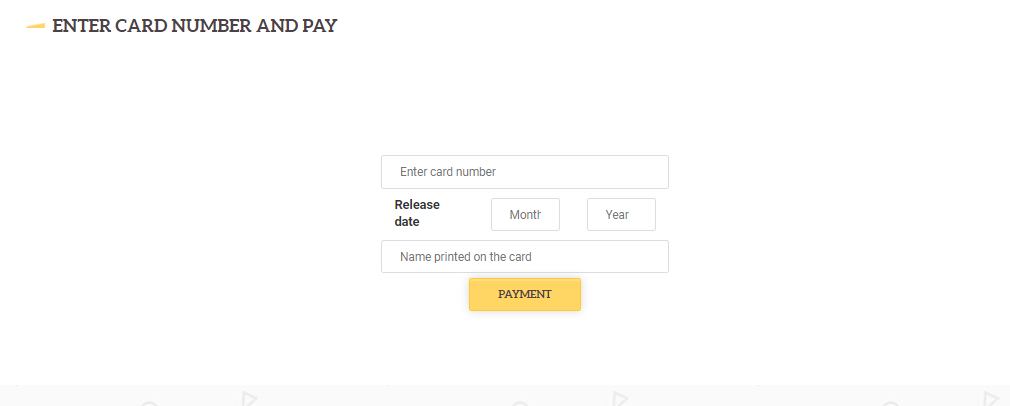
|  |  |
| --- | --- |
| Description | Show list of combo food and drink |
| Connect From | Home |
| Connect To | Payment pay |
| Validation | Customer is logged |
| Initial Values | List of combo food and drink |
| Events | - When user click button plus and minus value to choose combo quantity  - User can select multiple combo at the same time or not. |

* + 1. Total cost of selected products

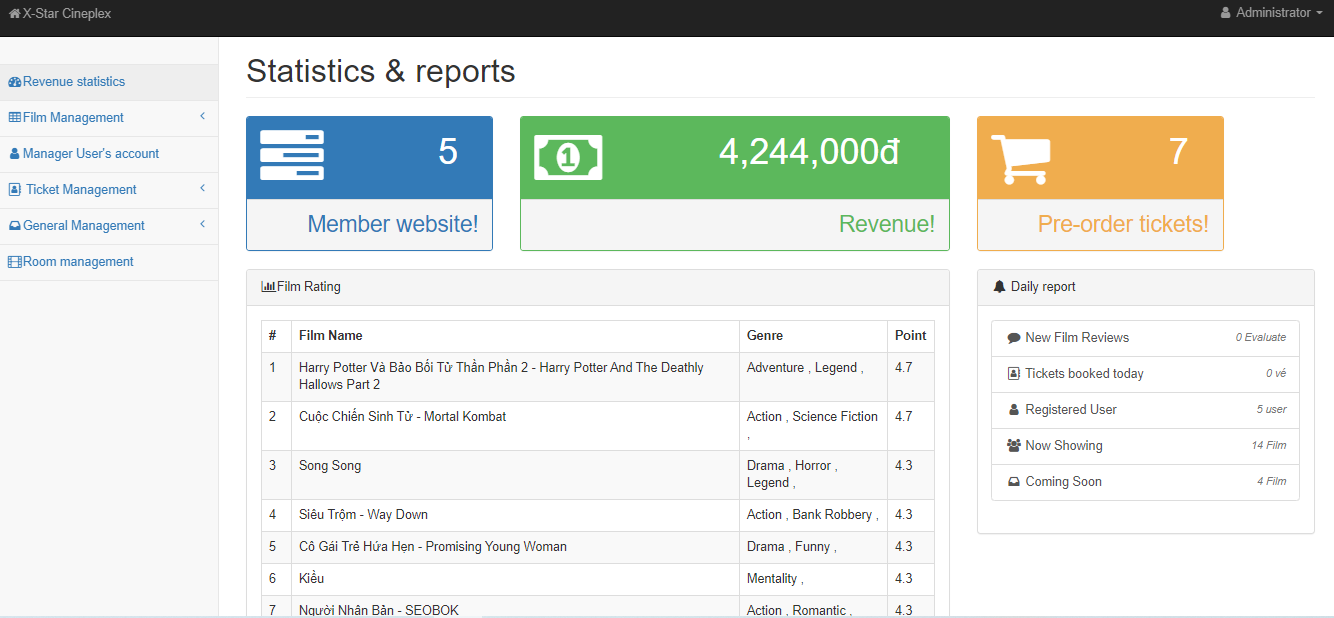


|  |  |
| --- | --- |
| Description | Show list of products and total cost |
| Connect From | Home |
| Connect To | Payment page |
| Validation | Customer is logged |
| Initial Values | List of combo food and drink |
| Events | - When user click button continue it will go to the total amount page  - User can choose below bank for payment |

* + 1. Total cost of selected products

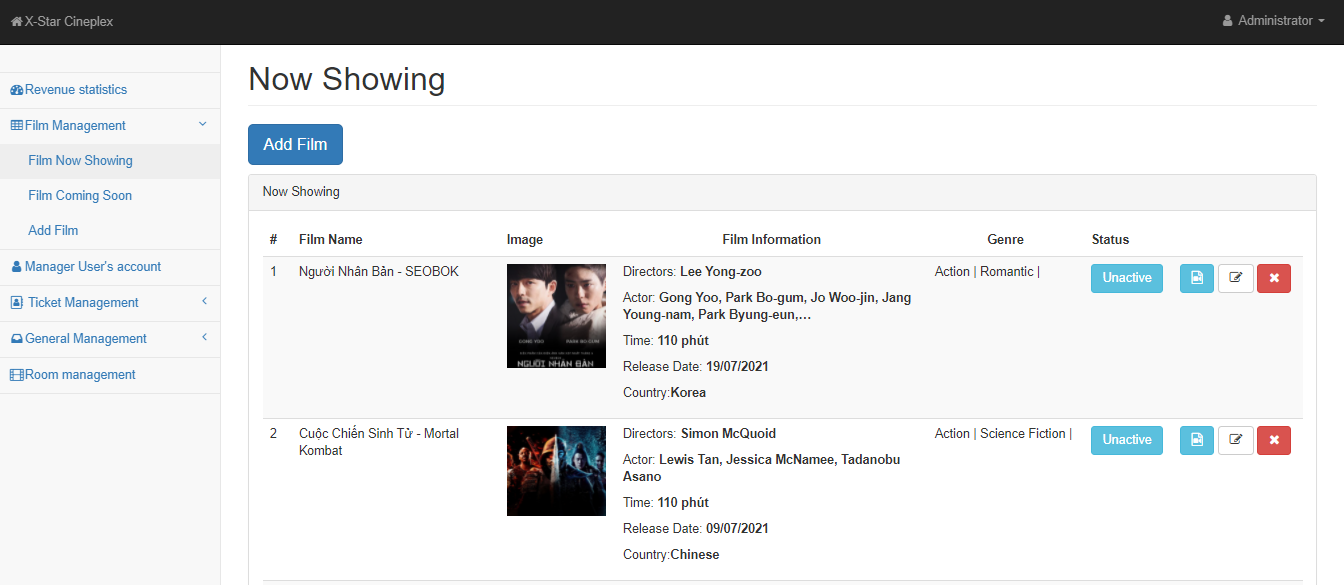


|  |  |
| --- | --- |
| Description | Show payment form |
| Connect From | Home |
| Connect To | Payment page |
| Validation | - All fields: not blank  -Invalid card number  Invalid name printed on card |
| Initial Values | N/A |
| Events | - When click button "Submit", check card number and name. If success, go to payment completion page |

* 1. Admin’s Side
     1. ADMIN’S DASHBOARD

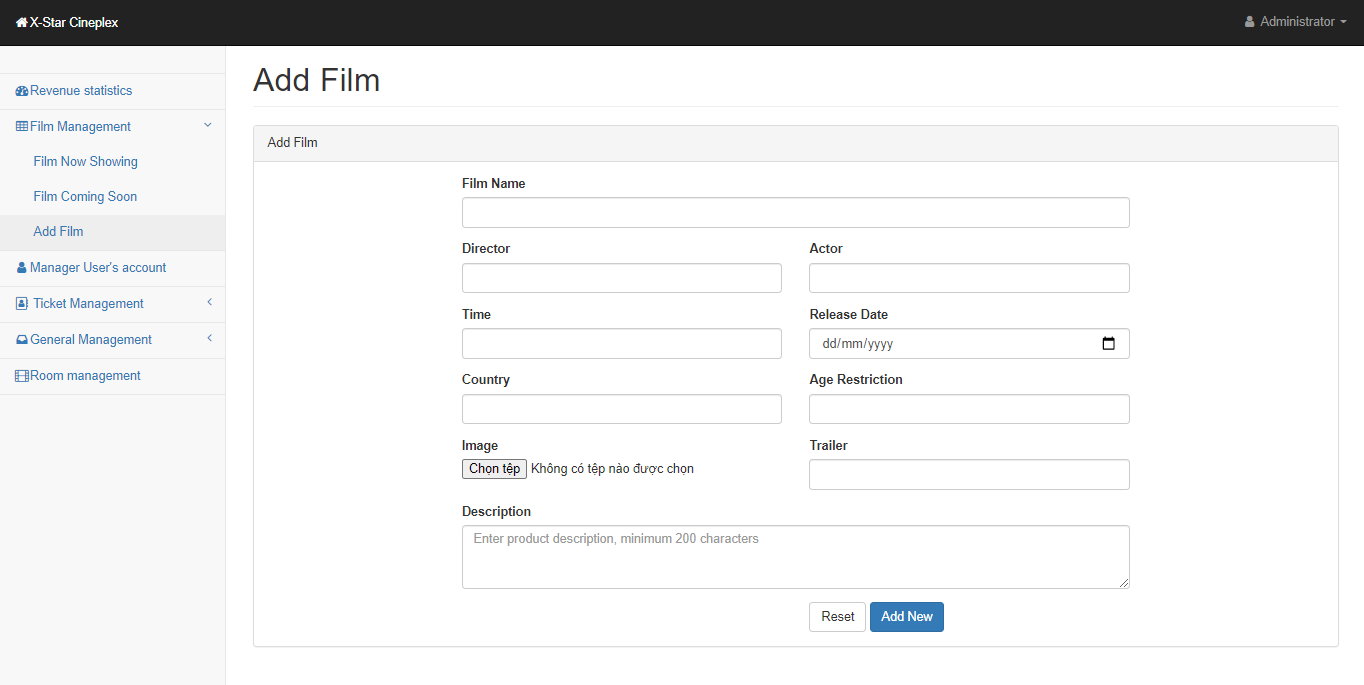
|  |  |
| --- | --- |
| Description | Admin Dashboard |
| Connect From | Login page |
| Connect To | All |
| Validation | N/A |
| Initial Values | N/A |
| Events | - When click on menu tabs, redirect to respective pages |

* + 1. Movie List



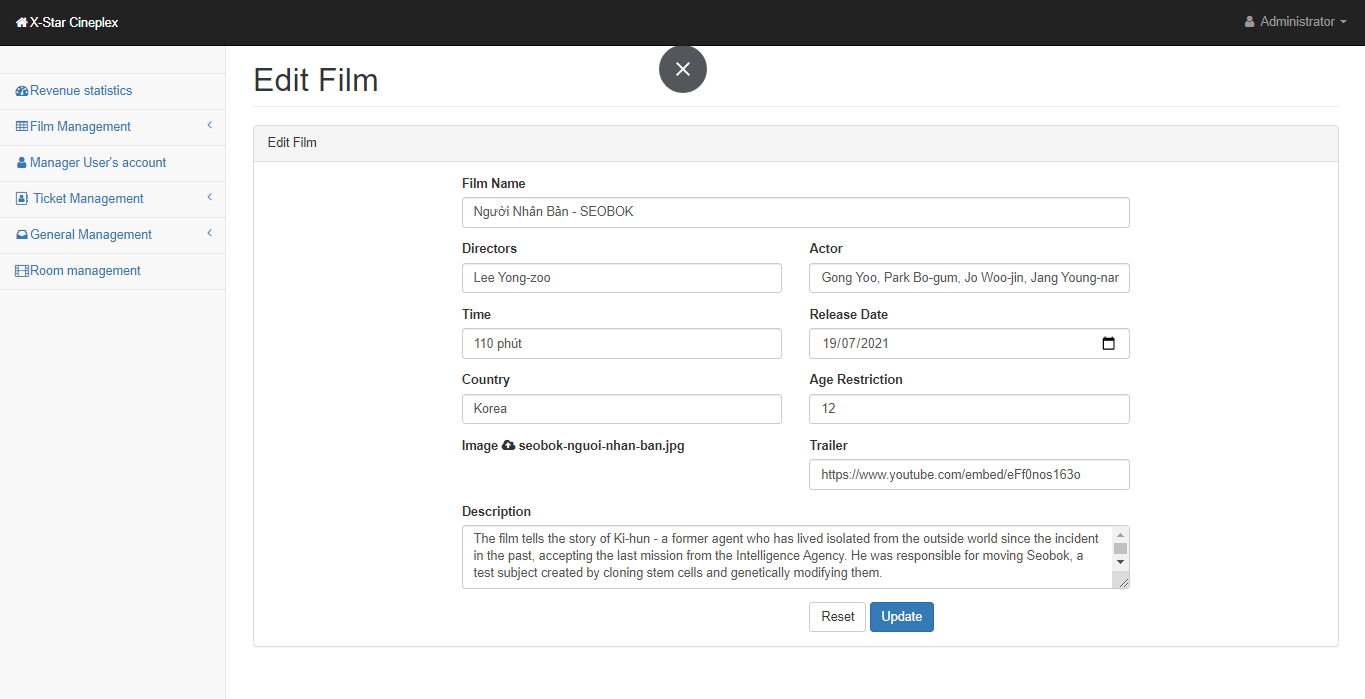
|  |  |
| --- | --- |
| Description | movie overview |
| Connect From | Dashboard |
| Connect To | Category: Edit, delete, Active,Update Genre |
| Validation | N/A |
| Initial Values | Now Showing |
| Events | - when click "Un-active" button, film won't show up on home page - when click "Active" button, film will show up on home page - when click "Update Genre" button, show dialogue to edit movie genre  - when click "Edit" button, go to edit film page - when click "Delete" button, check and delete category in database |

* + 1. Add film



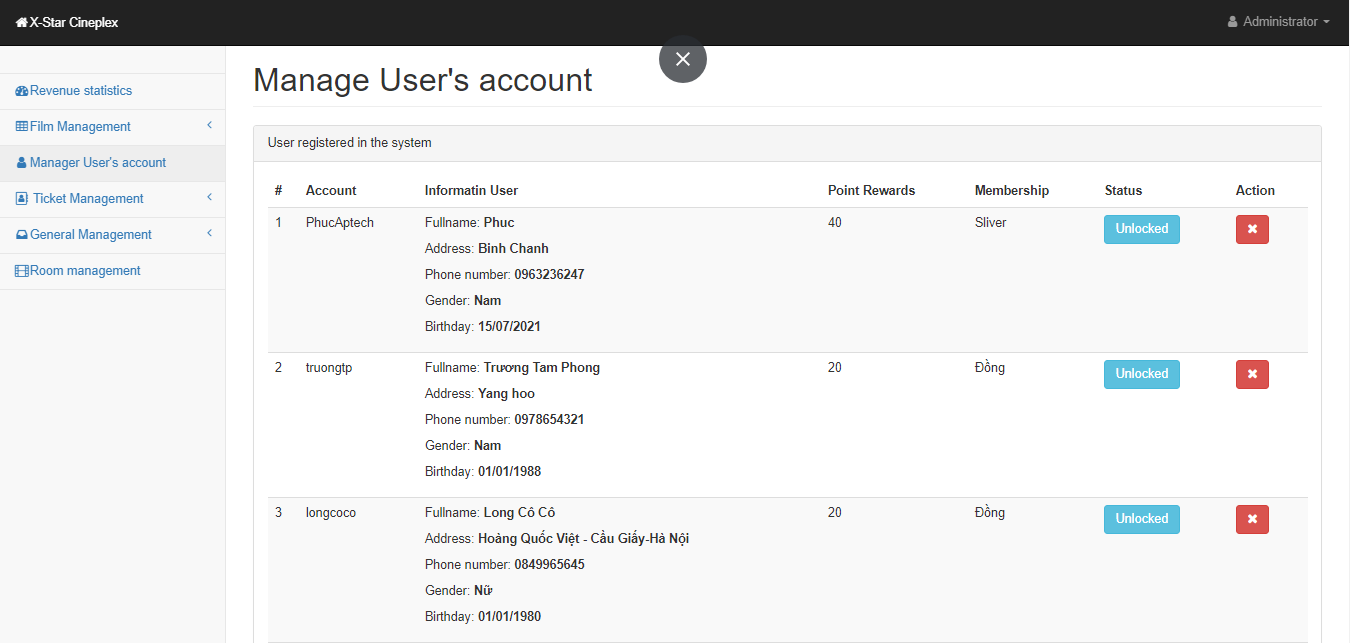
|  |  |
| --- | --- |
| Description | Insert new movie |
| Connect From | movie-overview page |
| Connect To | movie-overview page |
| Validation | - All fields: not blank |
| Initial Values | Movie parent list |
| Events | - when click "Add New" button, save new movie to database and go to category-overview page - when click "Reset" button, reset all fields to blank |

* + 1. Update movie

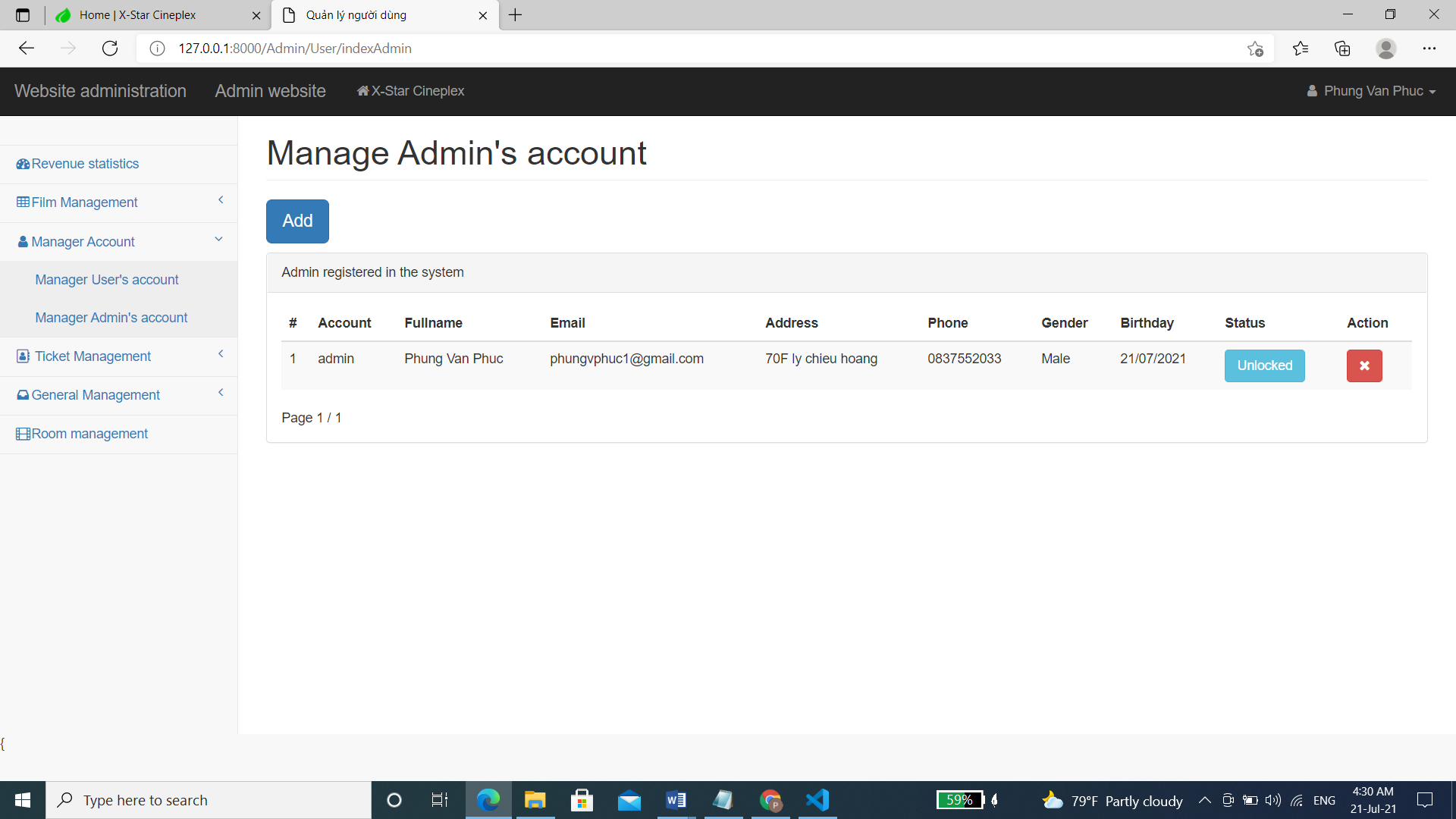


|  |  |
| --- | --- |
| Description | Edit Film |
| Connect From | movie-overview page |
| Connect To | movie-overview page |
| Validation | - All fields: not blank |
| Initial Values | Category parent list |
| Events | - when click "Update" button, update movie to database and go to movie-overview page - when click "Reset" button, reset all fields to blank |

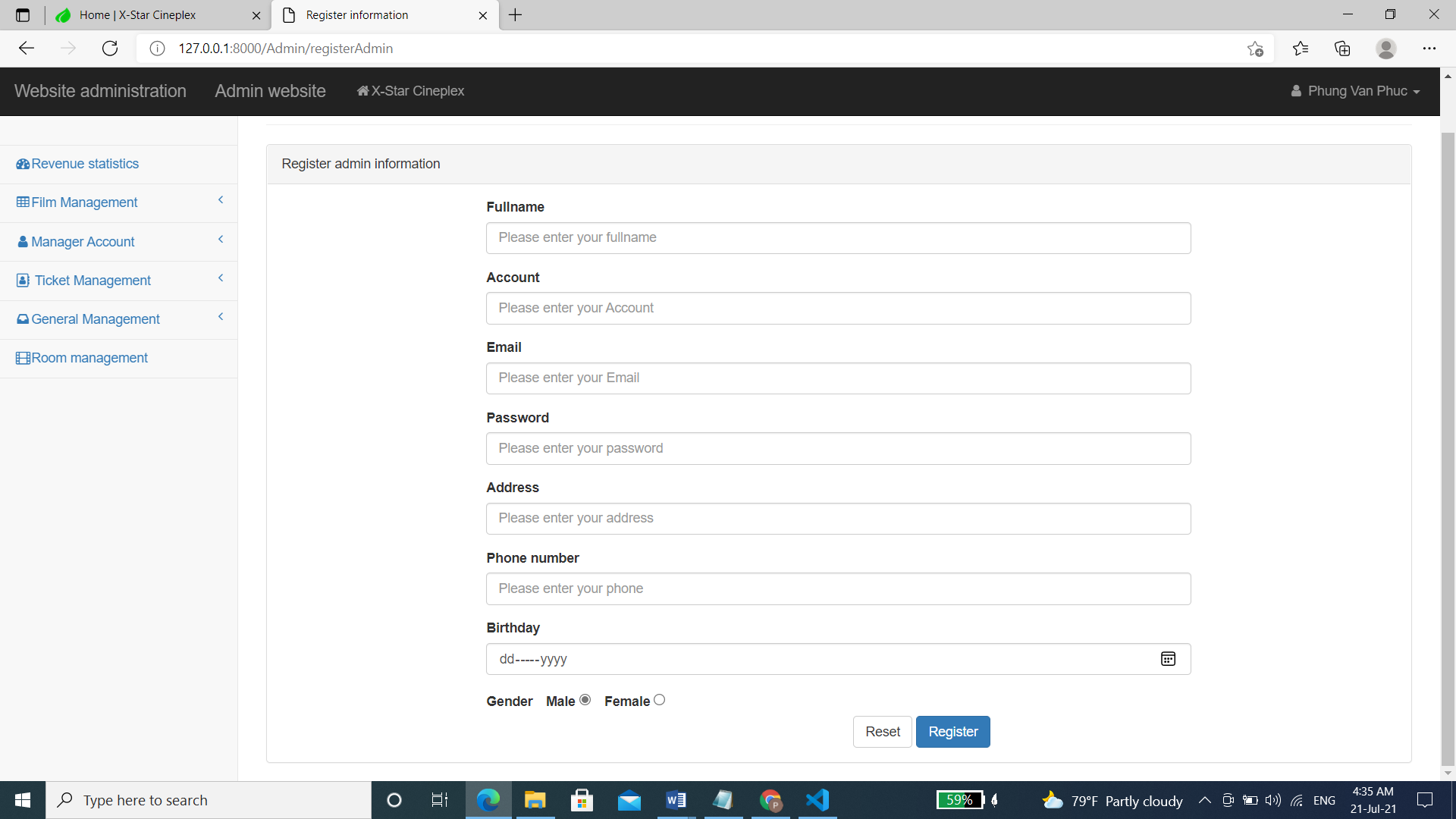
* + 1. Manage user account



|  |  |
| --- | --- |
| Description | Show User List |
| Connect From | Dashboard |
| Connect To | Ban User, Delete user, membership and points |
| Validation | N/A |
| Initial Values | User List |
| Events | - when click "Unlocked" button, User’s account is locked and vice again.  - when click "Delete" button, check and delete user account if the account is not related to ticketlist. |

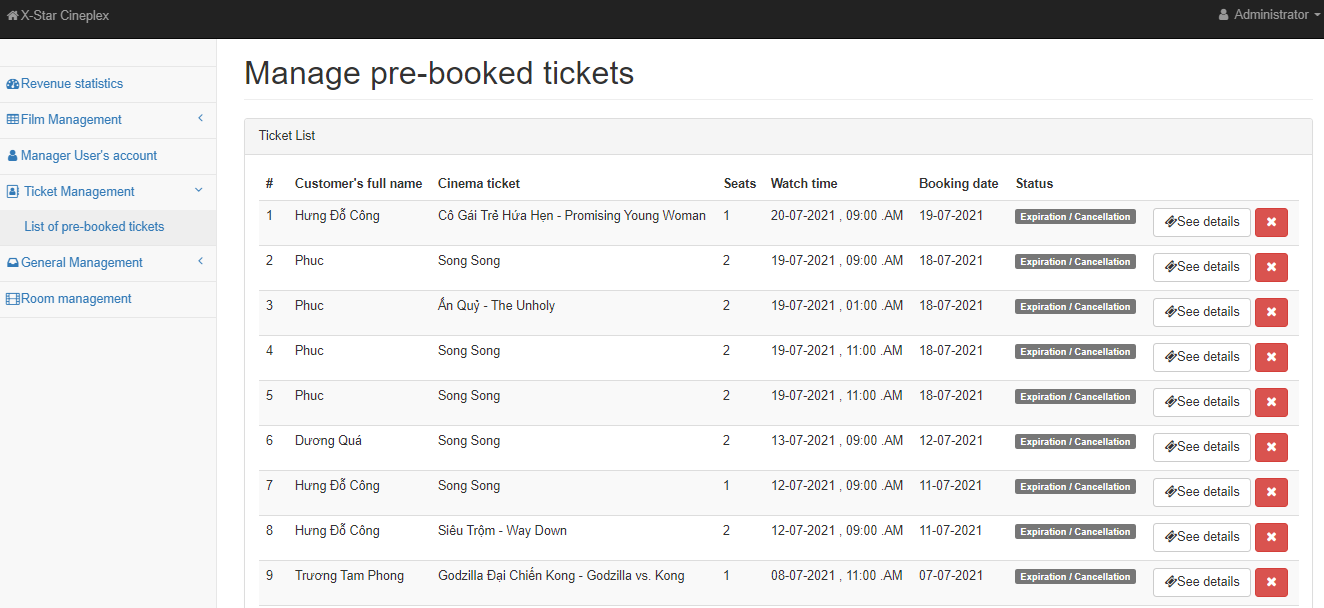
* + 1. Manage admin account

|  |  |
| --- | --- |
| Description | Show Admin List |
| Connect From | Dashboard |
| Connect To | Ban User, Delete admin account |
| Validation | N/A |
| Initial Values | Admin account list |
| Events | - when click "Unlocked" button, Admin’s account is locked and vice versa.  - when click "Delete" button, check and delete admin account. |

* + 1. Add admin account

|  |  |
| --- | --- |
| Description | Admin signs up for new account |
| Connect From | Home Admin Pages |
| Connect To | Login page, Home |
| Validation | - All fields: not blank  Fullname: "Please enter your first and last name",  Account: "Please enter your account",  Email: "Please enter your email"  Password: "Please enter a password",  Address: "Please enter your address",  BirthDay: "Please enter your date of birth",  Password: Please enter the phone number  - Email must be in the correct format  - Phone number must be in correct format |
| Initial Values | N/A |
| Events | - When click button "Register", check all fields. If success, go to login page |

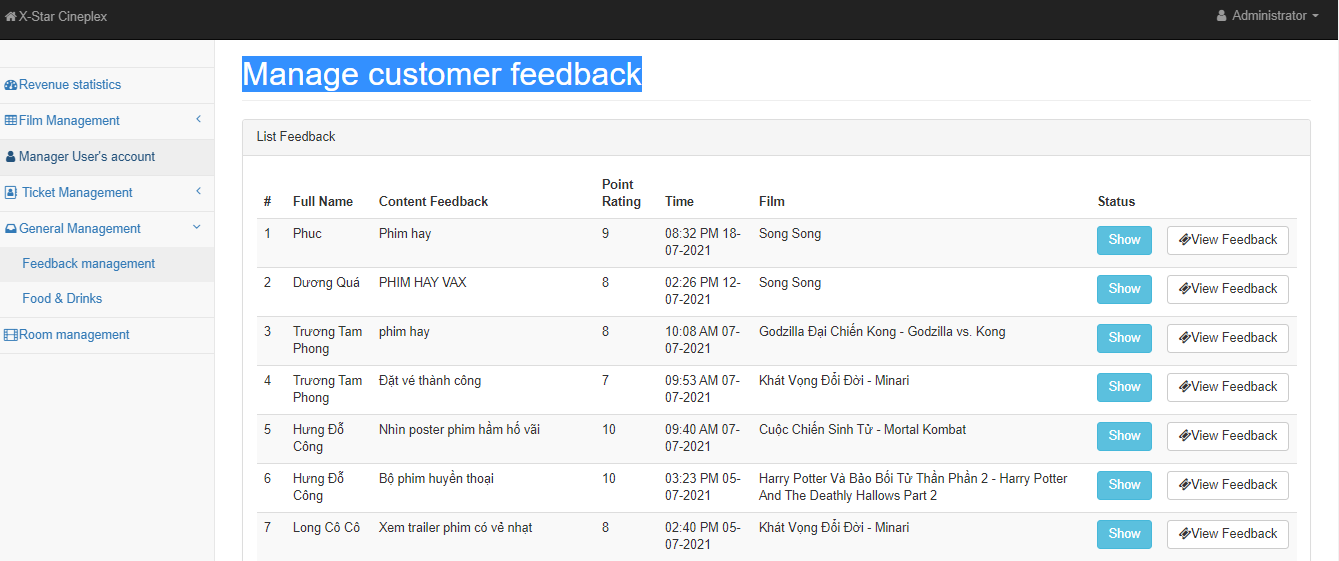
* + 1. Ticket management

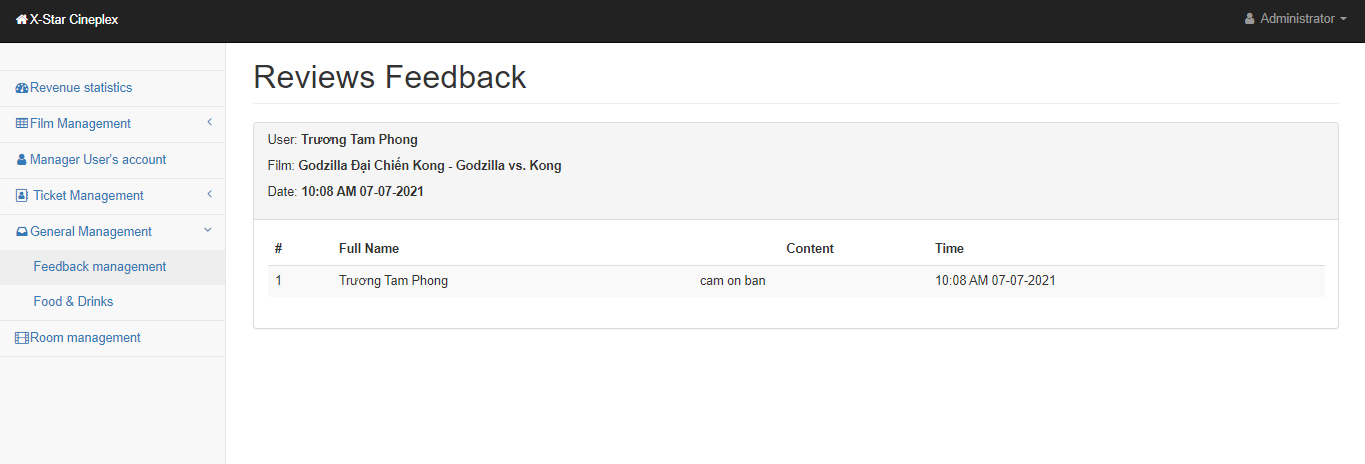


|  |  |
| --- | --- |
| Description | Show User List booked |
| Connect From | Dashboard |
| Connect To | Account User |
| Validation | N/A |
| Initial Values | User List book ticket |
| Events | - when click "See Detail" button, Show the results of the customer who booked the ticket  - when click "Delete" button, Check and clear data of users who have booked tickets |

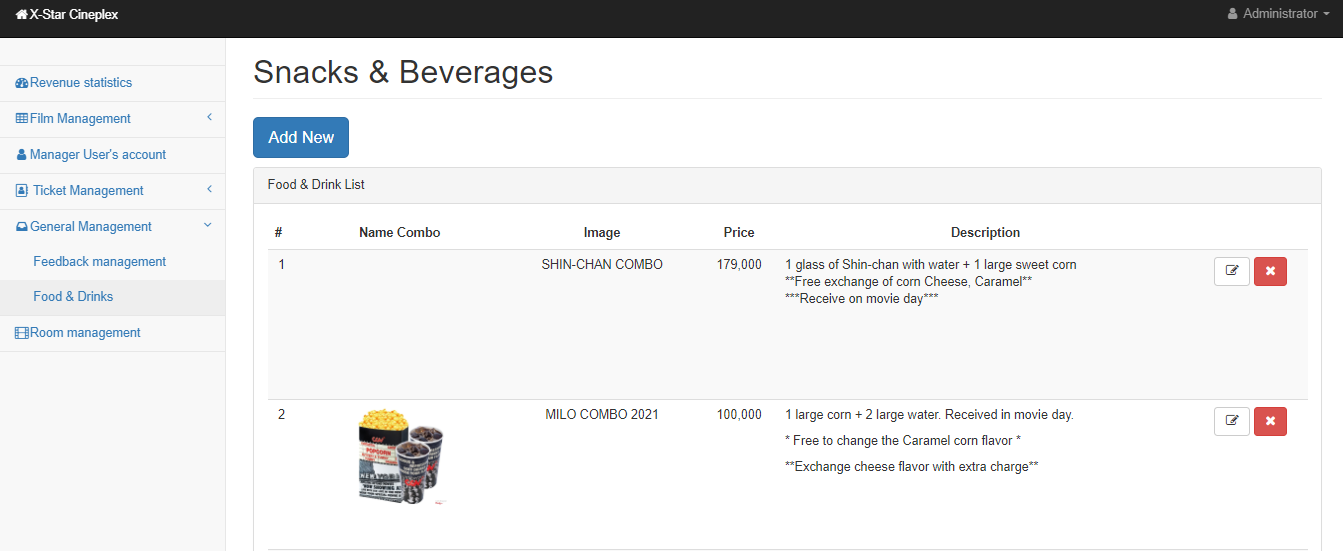
### 

* + 1. Magnage customer feedback



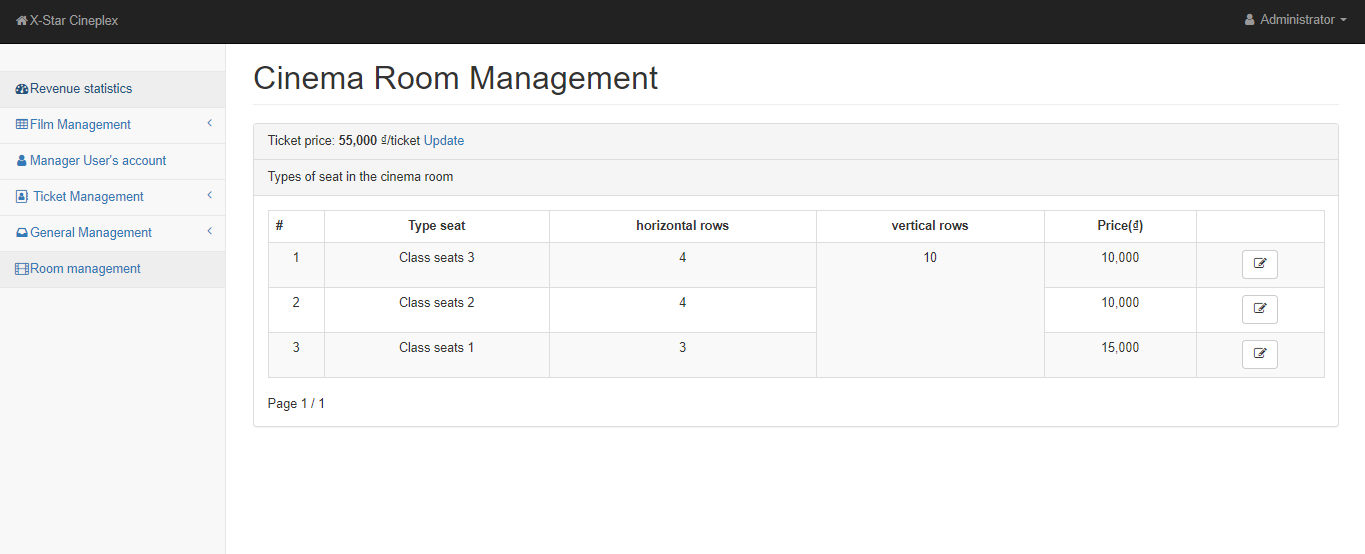


|  |  |
| --- | --- |
| Description | Manage feedback from customers |
| Connect From | Home page |
| Connect To | Dashboard |
| Validation | N/A |
| Events | - when click "Show" button, Comments will be displayed on the homepage  - when click "View Feedback" button, I'll see who answered that |

* + 1.  food & drink

### 

|  |  |
| --- | --- |
| Description | Food&Drink |
| Connect From | Book ticket page |
| Connect To | movie-overview page |
| Validation | Not blank |
| Events | - when click "Add New" button, Add New food & drink to the database  - when click "Edit" button, Update the list on database  -when click "Delete" button, Delete product categories |

* + 1. Room management

### 

|  |  |
| --- | --- |
| Description | Room Management |
| Connect From | Book ticket page |
| Connect To | movie-overview page |
| Validation | Not blank |
| Events | - when click "update " button, update price room  - when click "Edit" button, Edit and update new room data on database |

REVIEW 3 – TASK SHEET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Contents** | **Member** | **Activity Plan** | **Preparation** | |
| **Start Date** | **Actual Days** | **Status** |
| **Client** | | Phung Van Phuc | 12/06/2021 | 15 | Completed |
| 1 | Login, Register |
| 2 | User Profile |
| 3 | Change Password |
| 4 | Fogot pasword |
| 4 | My deal and point |
| **Admin** | |
| 5 | User Manage: view, update, delete, locked |
| 6 | Admin account manager: view, add, update, delete, locked |
| 7 | Membership and Point: view and caculate |
|  | | | | | |
| **Client** | | Tran Pham Thanh Hoang | 12/06/2021 | 15 | Completed |
| 1 | Now Showing |
| 2 | Coming Soon Film |
| 3 | Information and film’s trailer |
| 4 | Comment and rating film |
| **Admin** | |
| 5 | Manage File:view, add, edit, delete |
| 6 | Mange comment, rating: view , add ,edit ,delete |
|  | | | | | |
| **Client** | | Le Tuan Xuyen | 12/06/2021 | 15 | Completed |
| 1 | Homepage |
| 2 | Reserve Ticket |
| 3 | Book Food and drink |
| 4 | Payment page |
| **Admin** | |
| 5 | Manage Order :ticket, food and drink |
| 6 | Manage Food and Drink: view, add, edit, delete |
| 7 | Manager Ticket: view ,add, edit, delete |
| **Client** | | Nguyen Nhat Nam | 12/06/2021 | 15 | Completed |
| 1 | About Us |
| 2 | Category Film |
| 3 | Contact Form |
| **Admin** | |
| 4 | Manage Room: view, add, edit ,delete |
| 5 | Manager Category film: view, add, edit ,delete |
| 6 | Dashboard Homepage: revenue statistics |