System Analysis and Design – ITSE 1380

# CINEMA TICKETING SYSTEM

## PROJECT REPORT

Intructors:
Mr. Huy Nguyen
Mr. Liem Ha
Students:

| | |
|---|---|
| 01-13-0-00022 | Hien Nguyen |
| 01-13-1-20002 | Thao Pham |

Saigon Institute of Technology – Spring 2015

# Abstract

This project works on a simple cinema managing system. It serves the employees to carry out common tasks as well as view insights. It is a website which allow customer to check movie time table, about our cinema, new release movie, and also allow member to book ticket. Our purpose of develop this cinema system and cinema website because it has bring benefits to both company and also the customer.

# Acknowledgements

We wish to send our thanks to our instructors, Mr. Nguyen Quoc Huy and Mr. Ha Thanh Liem, for their guidance.

## List of Tables

## List of Figures

# Table of Contents

# Planning phase

## Feasibility Study

### Technical Feasibility

| Criteria | Advantage | Disadvantage |
|---|---|---|
| *Familiarity with the application* | The users are familiar with the functions of a movie ticket booking system. | Several similar systems have already been built. |
| *Familiarity with the technology* | The developers are familiar with the project's technologies. | - |
| *Project size* | - | CMS has a smaller size than established systems of the same type.<br>There are only 2 developers. |
| *Compatibility* | The company’s existing communications infrastructure matches the project's requirements. | - |

Table 1: Technical Feasibility

### Economic Feasibility

Estimated break-even point is 2 years since establishment.

### Organizational Feasibility

The system is predicted to be accepted by its users and incorporated into the ongoing operations of the organization.

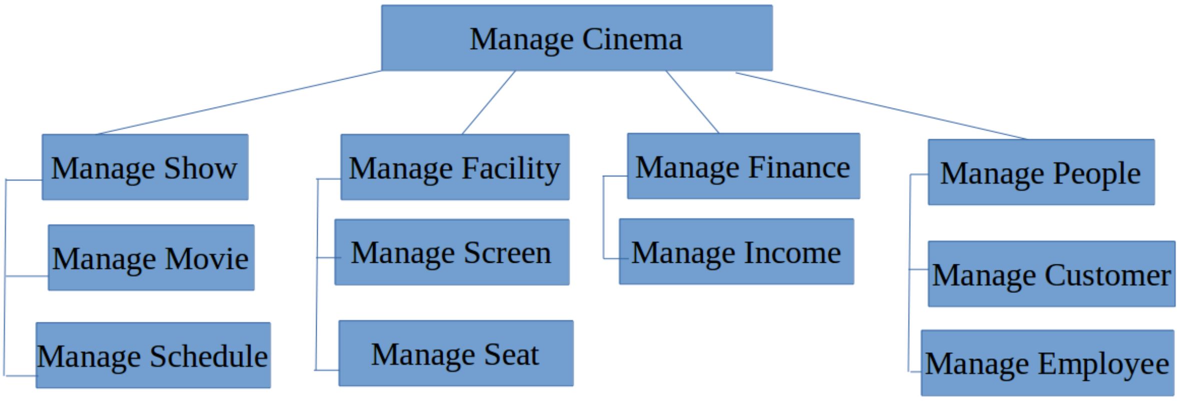
### Project Plan (using Gantt Chart)

(view PDF file or project file)

# Analysis phase

## Business Flow Diagram

Figure 1: BFD



## Requirements Determination

# Functional requirements:

1 **Search and browse**
  **1.1** The system will allow users to browse movies by predefined categories.
  **1.2** The system will allow users to search for movies by title, main cast, director and genre.
  **1.3** The system will allow users to watch trailer(s) of a movie selection.
  **1.4** The system will allow members to add movie selections to their favorite list.

2 **Read and Comment**
  **2.1** The system will allow members to post comments to showing movies.
  **2.2** The system will allow members to read and see related news.
  **2.3** The system will allow members to post comments on articles.

3 **Book tickets**
  **3.1** The system will enable the members to specify the movie choice and seat selection(s) to purchase ticket(s).
  **3.2** The system will allow members to view their booking history.

4 **Manage account information**
  **4.1** The system will enable the user to create an account (if desired) that will store user data and booking information.
  **4.2** The system will allow members to view and update their personal information.
  **4.3** The system will allow members to change their password or request a password reset.

5 **Promote**
  **5.1** Based on the member's previous bookings and favorite list, movie recommendations will be generated and displayed on future visits to the web site.

6 **Sell and reserve tickets (for moderators)**
  **6.1** The system will allow the moderator to reserve tickets and cancel reservations within permitted time.
  **6.2** The system will allow the moderator to print tickets for unfinished sessions.

7 **Manage**

**7.1** The system will provide a control panel for administrators to manage accounts, user interactions, movie information, sessions, news, pricing and system maintenance.

**7.2** The system will allow moderators to hide offending comments and submit a report to the administrators on the member who violates the rules.

**8 Reports (visible to administrators only)**

**8.1** The system will provide the administrator with different types of income report (daily, weekly, monthly, based on movies, based on predefined time slots).

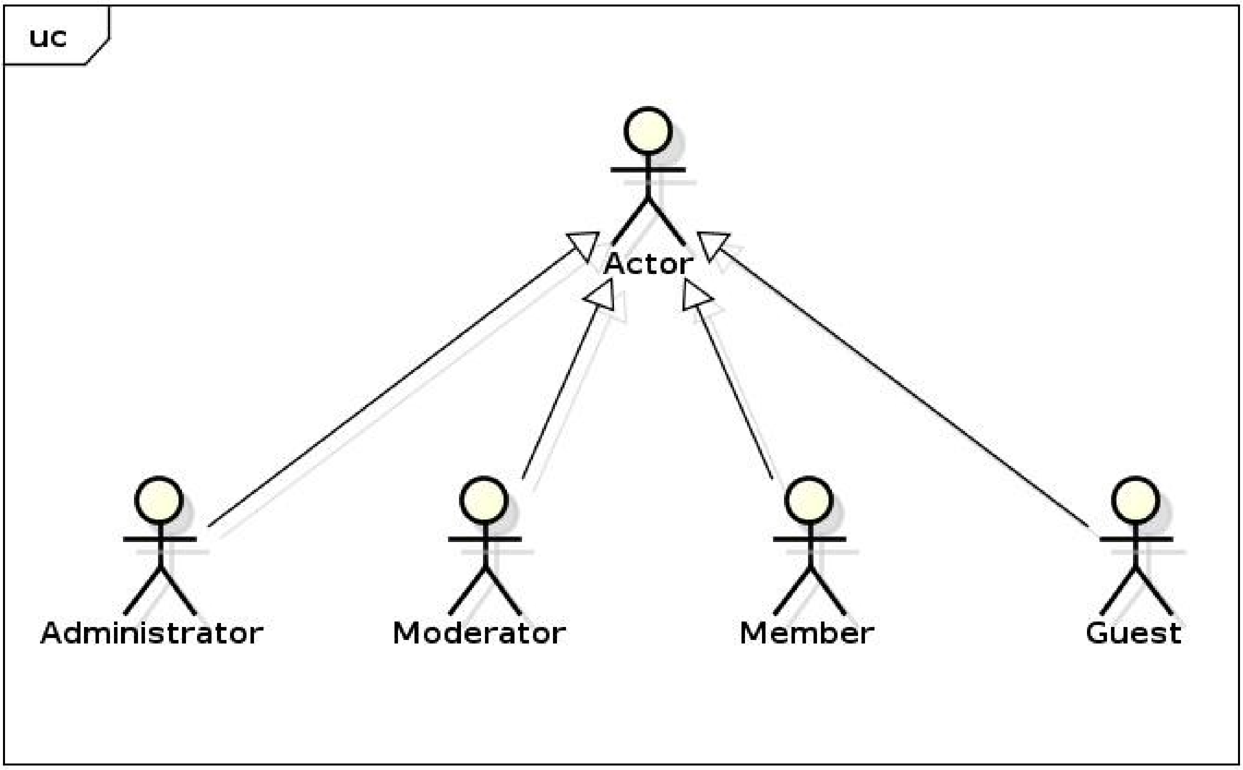
## Non-functional requirements:

| Non-functional requirement | Detail |
|---|---|
| Operational | 1. The system can run on hand-held devices.<br>2. The system should be able to work on common Web browsers (Internet Explorer 7 and later, Mozilla Firefox, Google Chrome, Opera, Safari).<br>3. The system must be able to work with different operating systems (i.e., Linux; Windows 7).<br>4. The system must be able to import and export Excel spreadsheets.<br>5. The system must enable real-time database updates. |
| Performance | 6. Any interaction between the user and the system should not exceed 4 seconds.<br>7. The system supports 400 simultaneous users.<br>8. The system should be available for use 24/7, with the exception of scheduled maintenance.<br>9. Scheduled maintenance shall not exceed one 2-hour period each month.<br>10. The system will have 99% uptime performance. |
| Security | 11. Only administrators can view and export insights, only administrators and moderators can print tickets.<br>12. All users logging in will be required to authenticate.<br>13. Secured and encrypted Internet protocols must be used for administering activities.<br>14. Data will be encrypted from the user’s computer to the Web site to provide secure ordering.<br>15. The system includes all available safeguards from viruses, worms, Trojan horses, etc. |
| Cultural and Political | 16. The system will operate in Vietnamese and English.<br>17. Administrators can delete comments, block or deactivate accounts.<br>18. Personal information is protected in compliance with the Data Protection Act. |

Table 2: Non-functional requirements

# Use Case Analysis

Figure 2: Primary actors



Actor

| No. | Actor | Description |
|---|---|---|
| 1 | Administrator | Has all privileges on the system. |
| 2 | Moderator | Has the privileges on ticket-related matters, can hide comments and submit reports on members. |
| 3 | Member | Is the main actor who comes to the web site to browse movies/news and purchase tickets online. |
| 4 | Guest | Has similar privileges to those of a member, except for purchasing tickets online and adding movies to favorite list. |

Table 3: Description of primary actors

# Use case diagram

## Administrator

Figure 3: Use cases for administrator

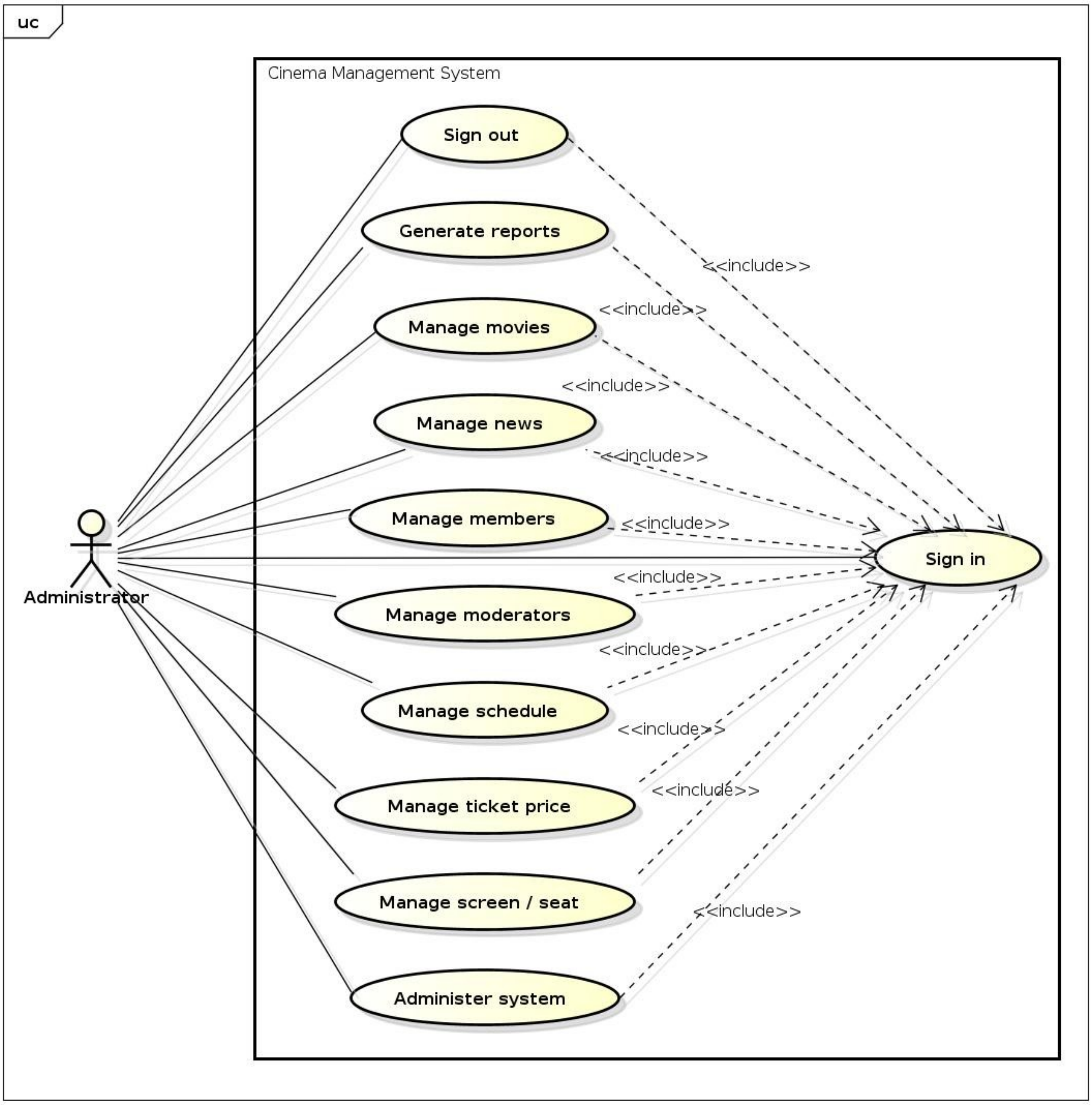


Figure 4: UC Administrator: Manage movies

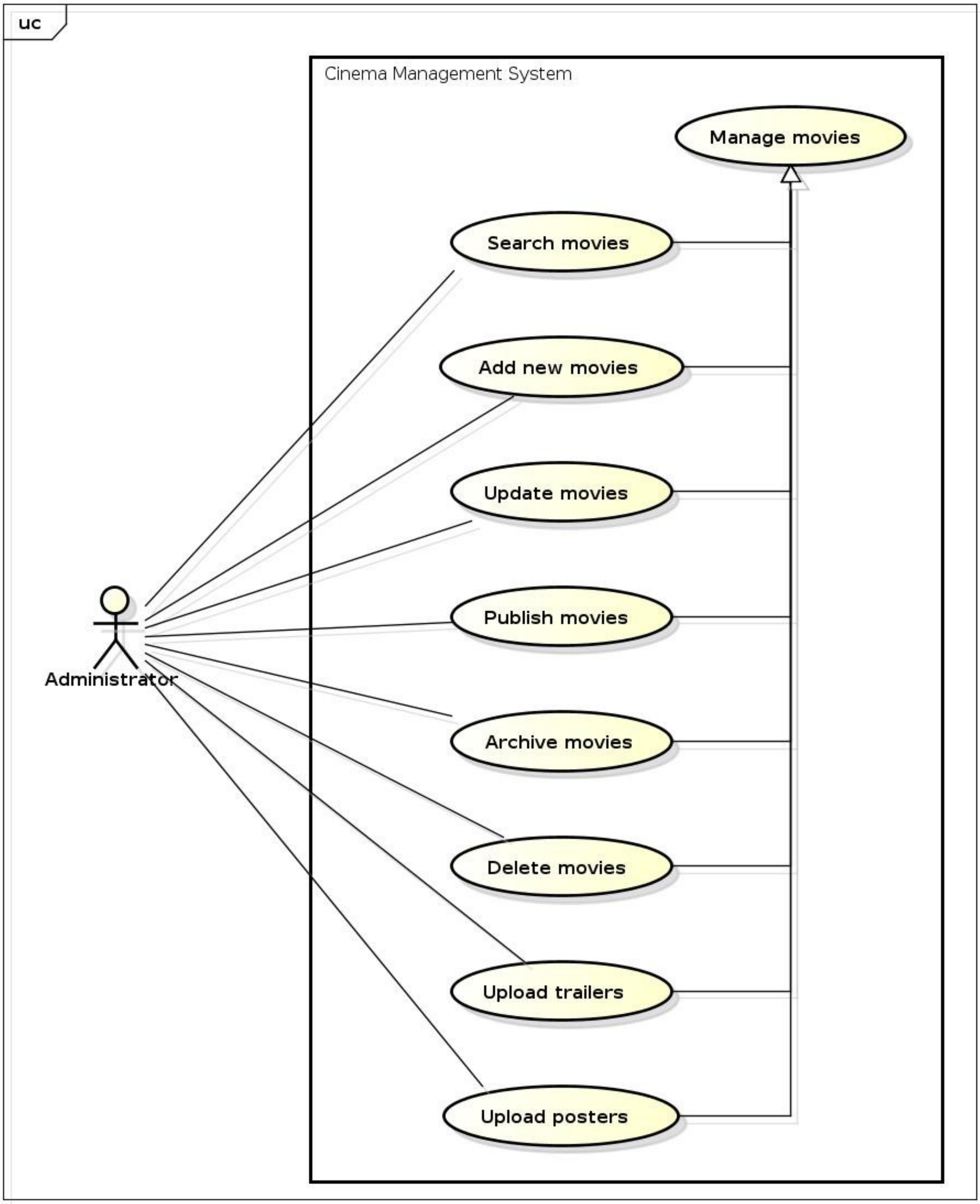


Figure 5: UC Administrator: Manage news

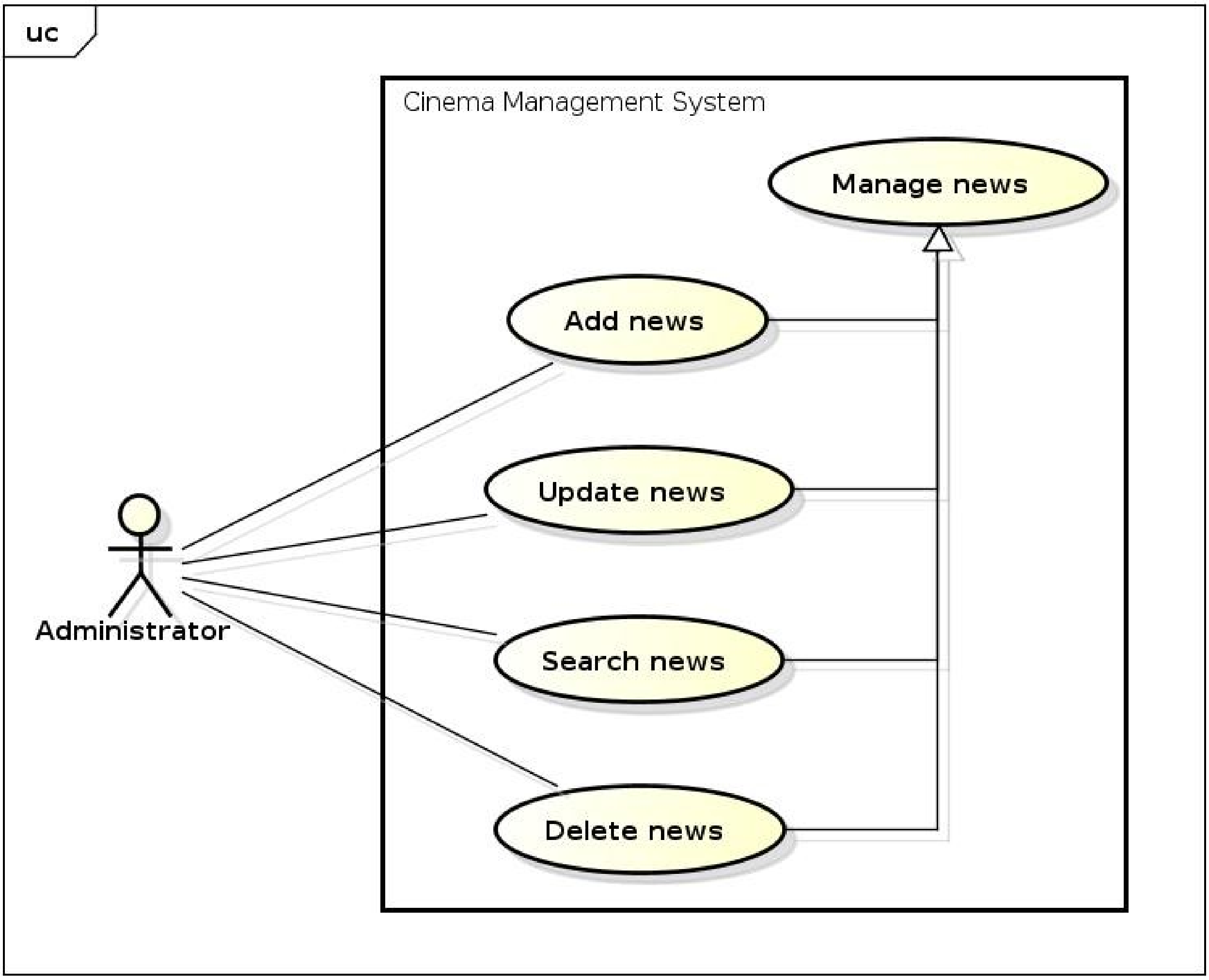


Figure 6: UC Administrator: Manage members

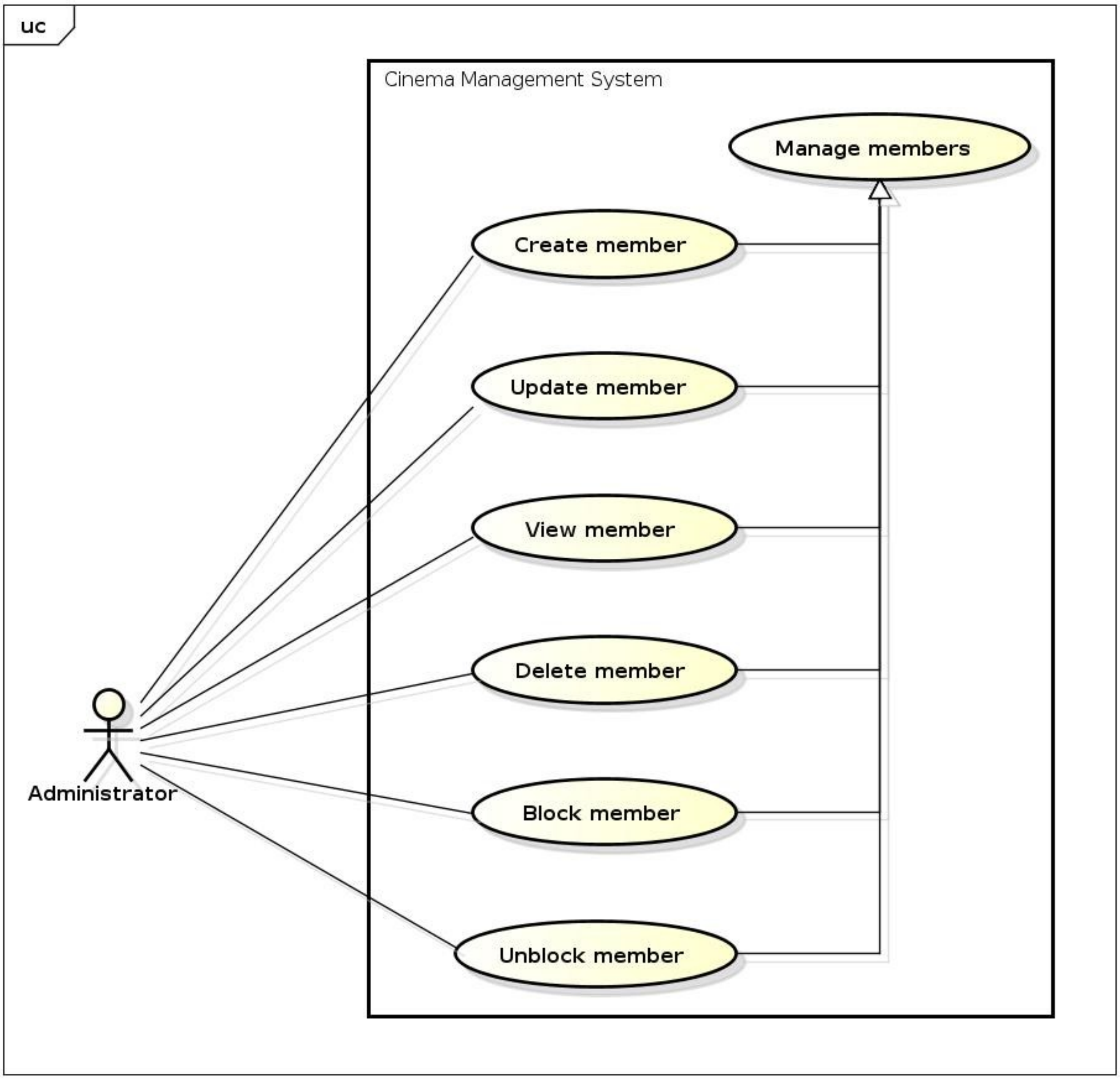


Figure 7: UC Administrator: Manage moderators

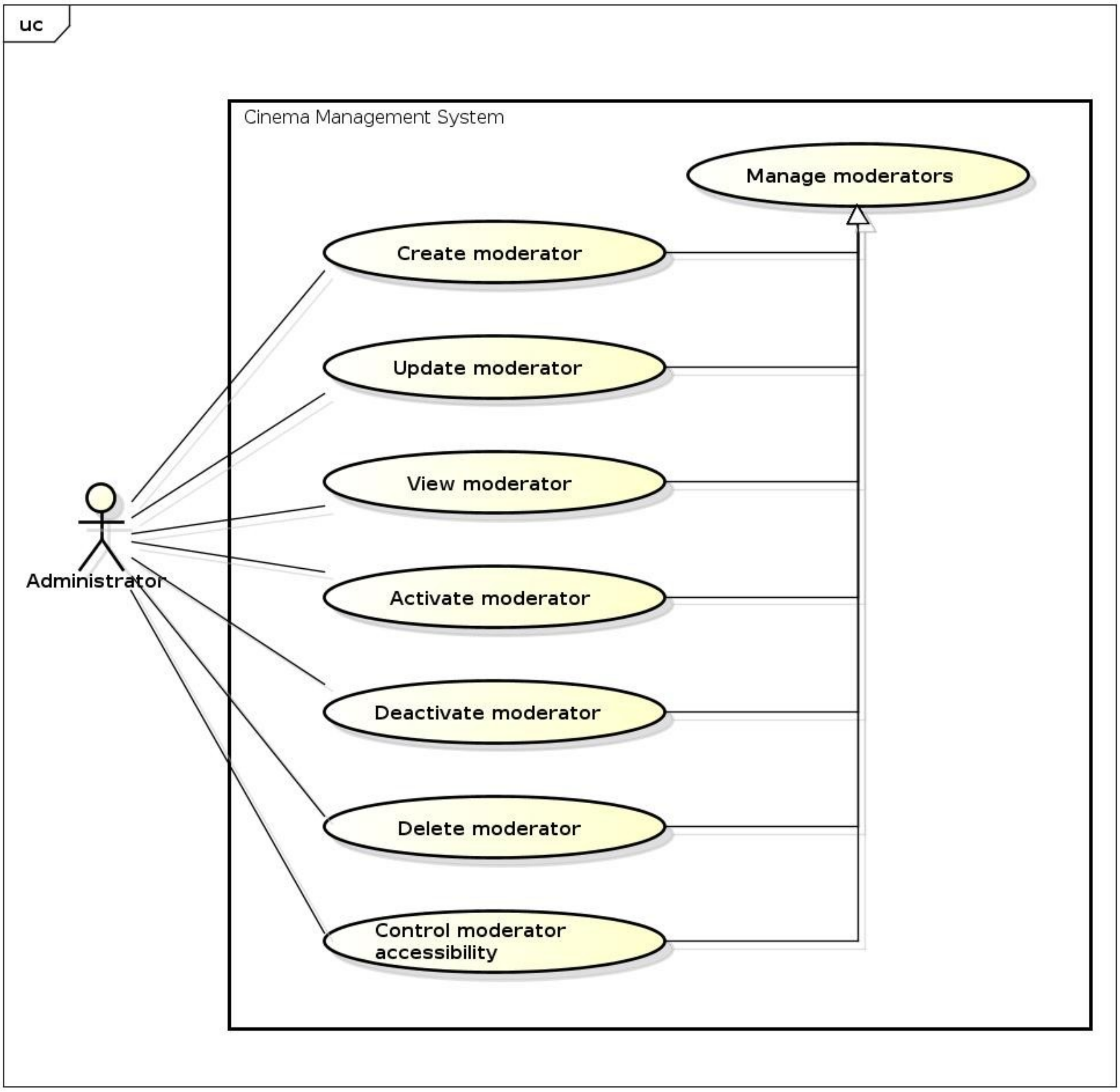


Figure 8: UC Administrator: Manage schedule

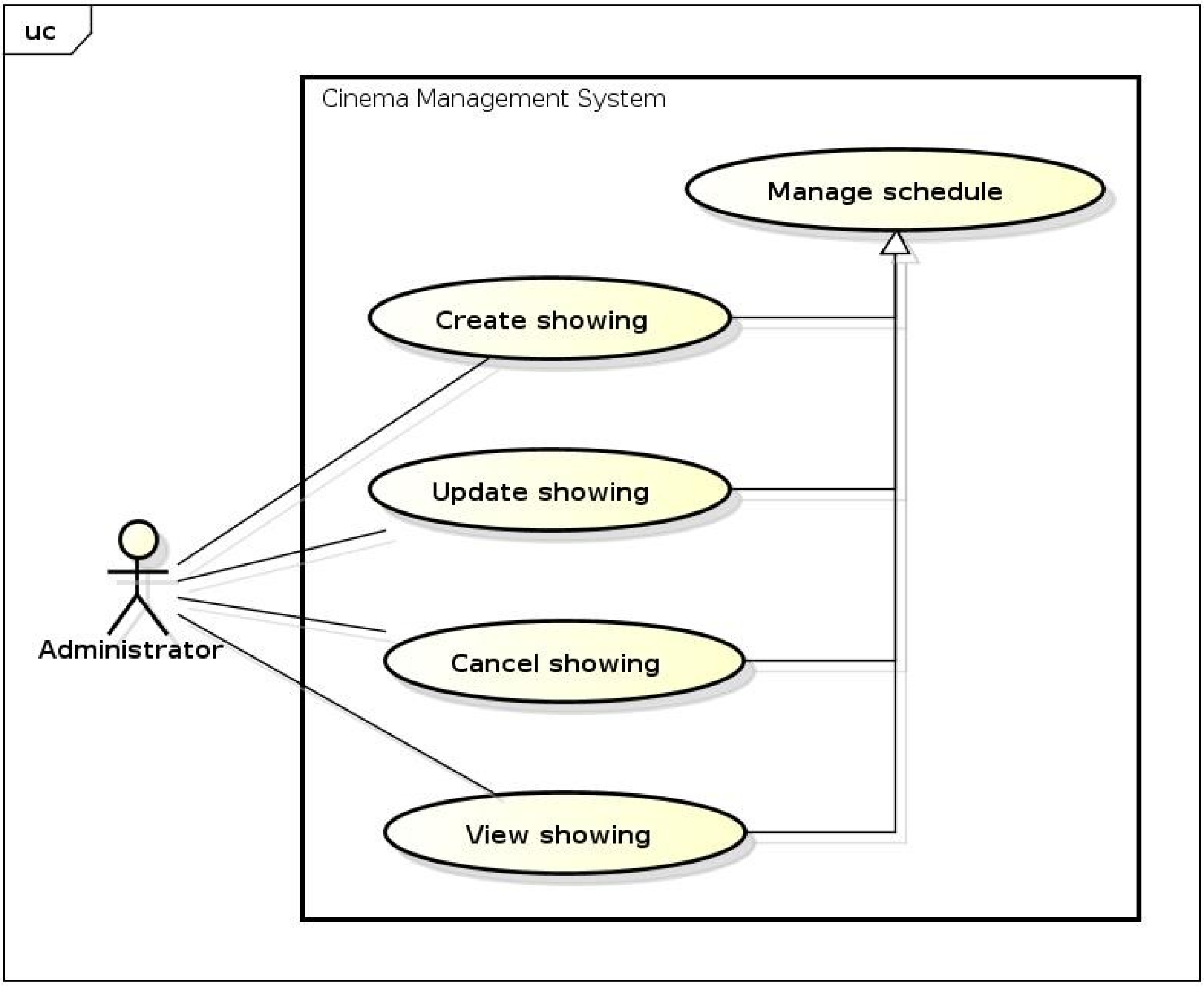


Figure 9: UC Administrator: Manage ticket price

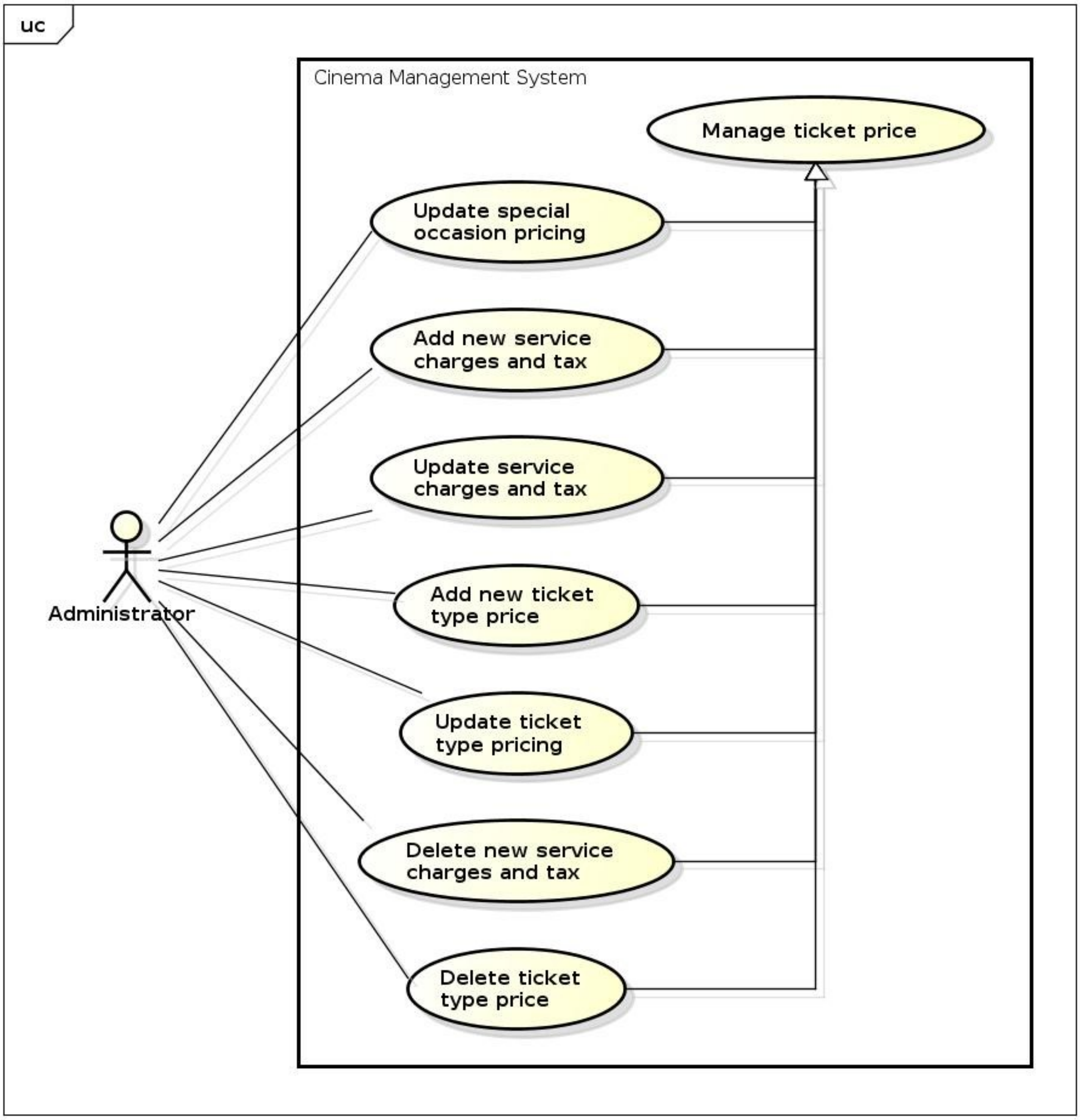

uc
Cinema Management System
Manage ticket price
Update special occasion pricing
Add new service charges and tax
Update service charges and tax
Add new ticket type price
Update ticket type pricing
Delete new service charges and tax
Delete ticket type price
Administrator

Figure 10: UC Administrator: Manage screen / seat

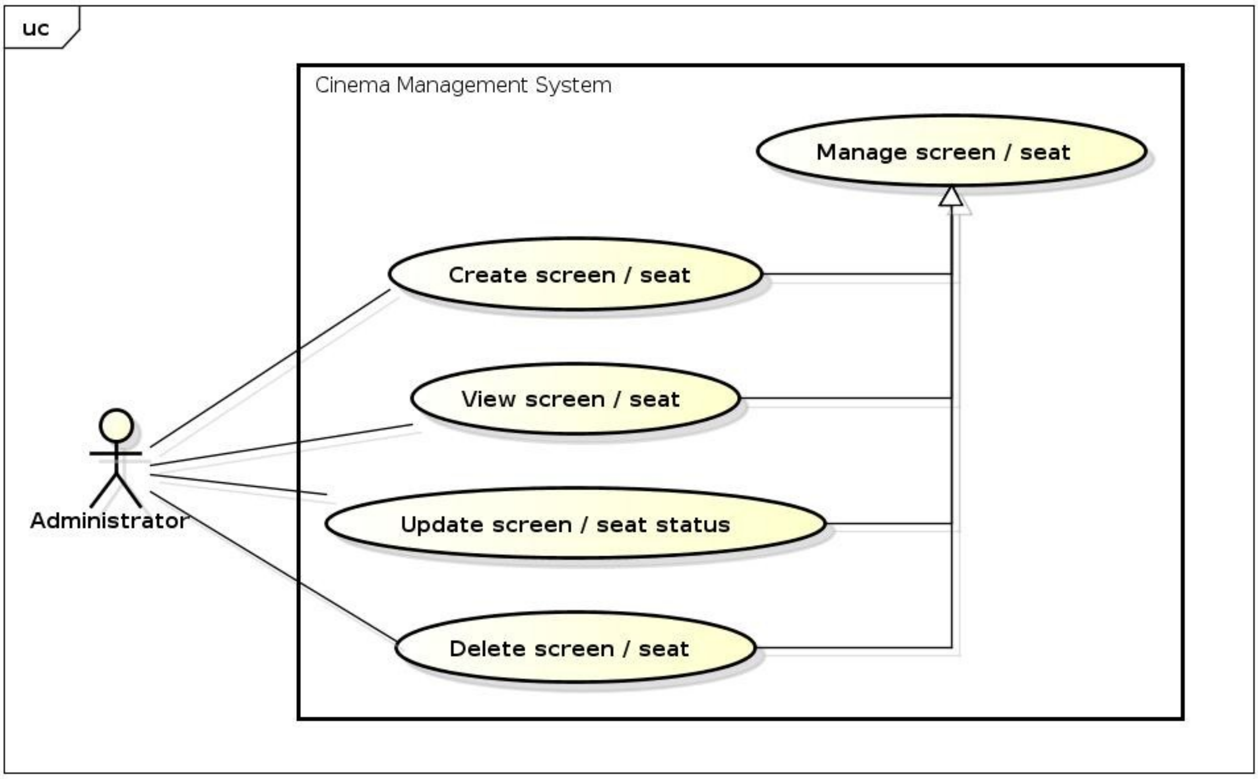
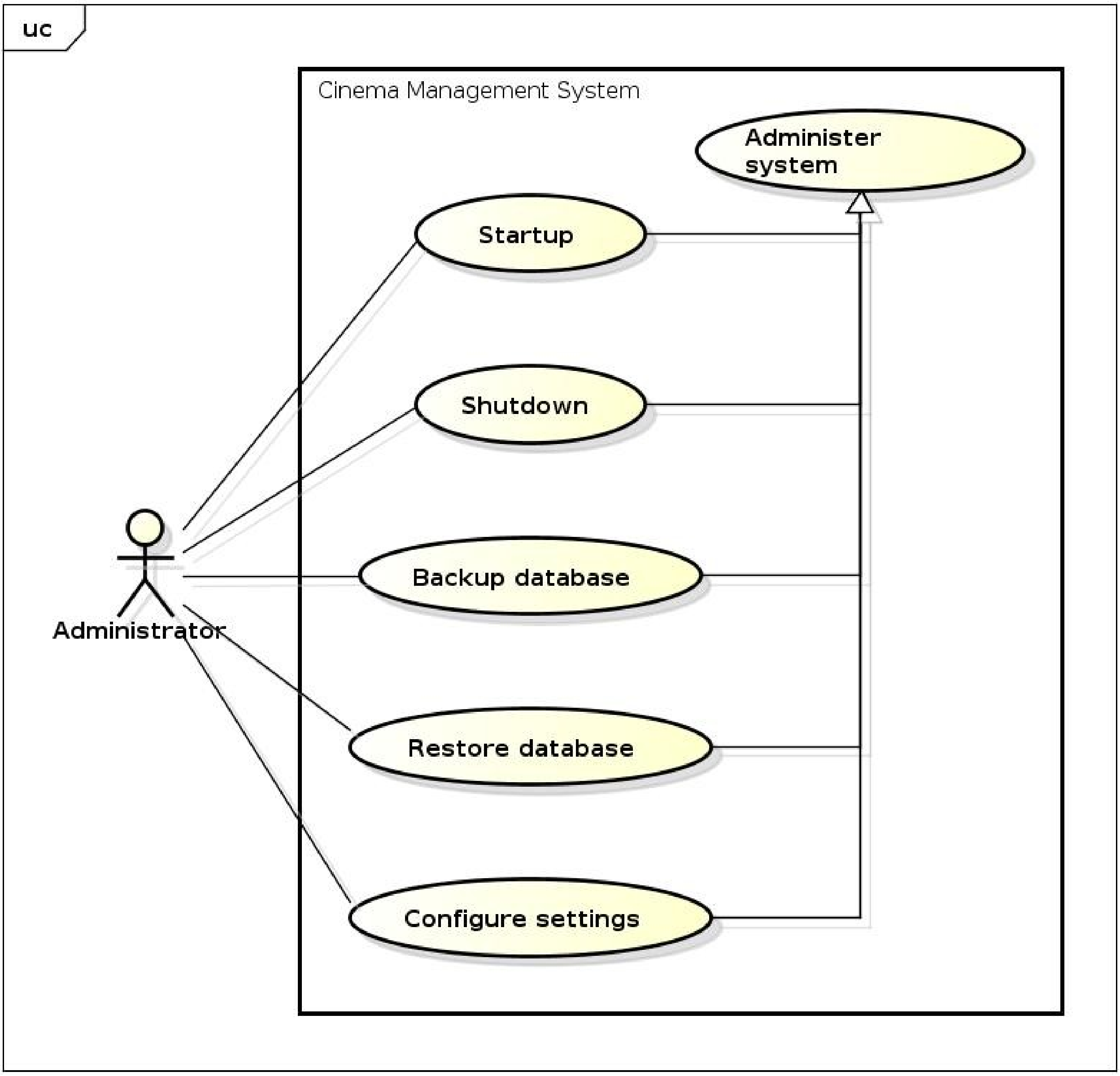
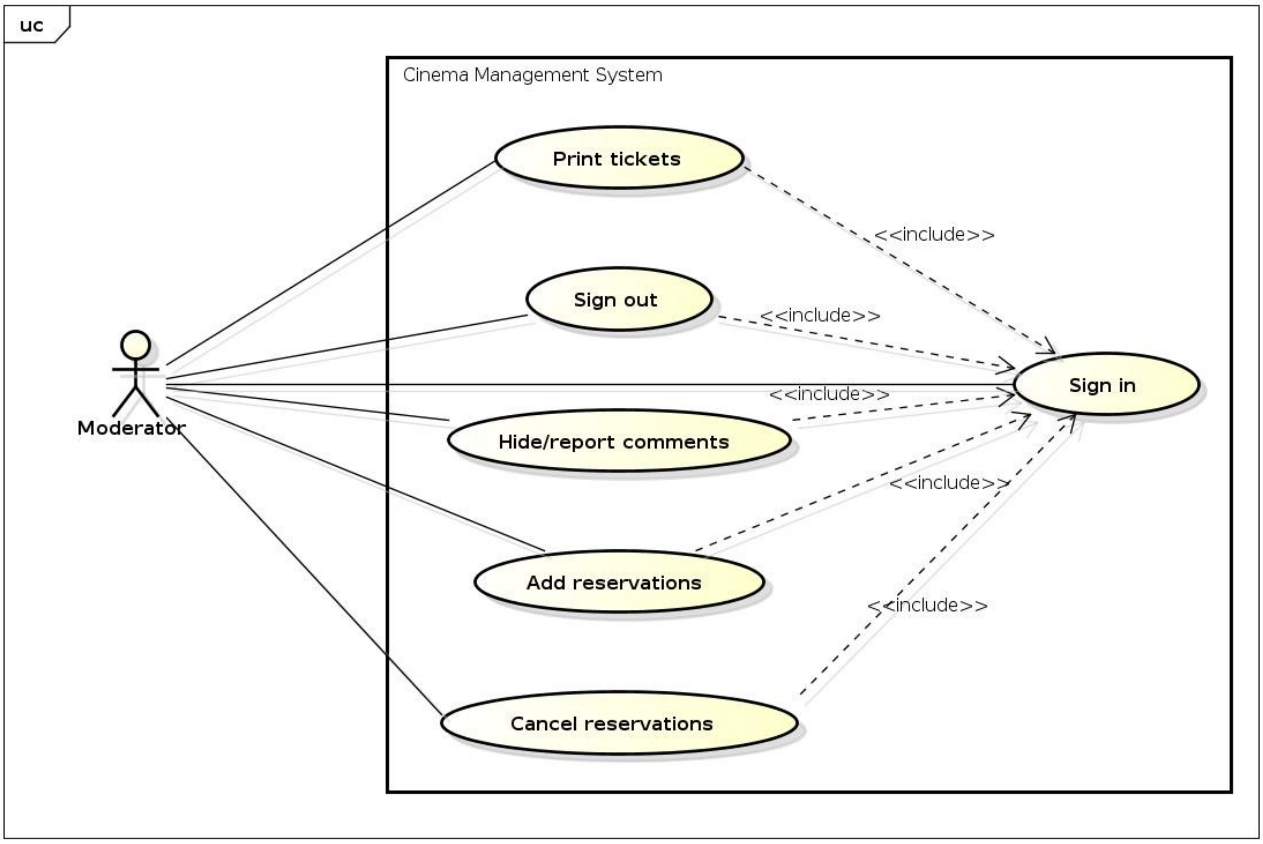


Figure 11: UC Administrator: Administer System



## Moderator

Figure 12: Use cases for Moderator



## Member

Figure 13: Use cases for member

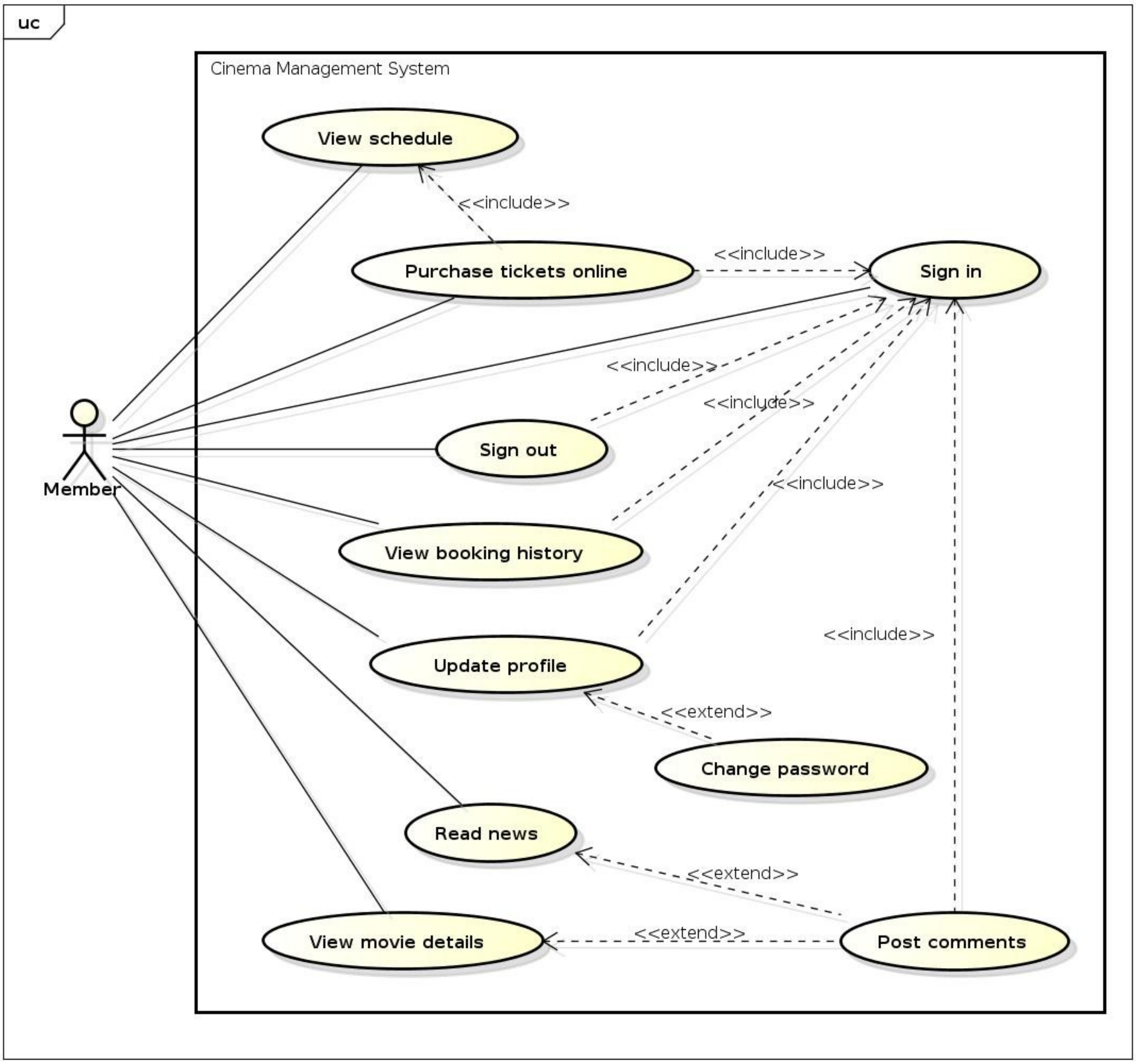


Figure 14: UC Member: Purchase tickets online

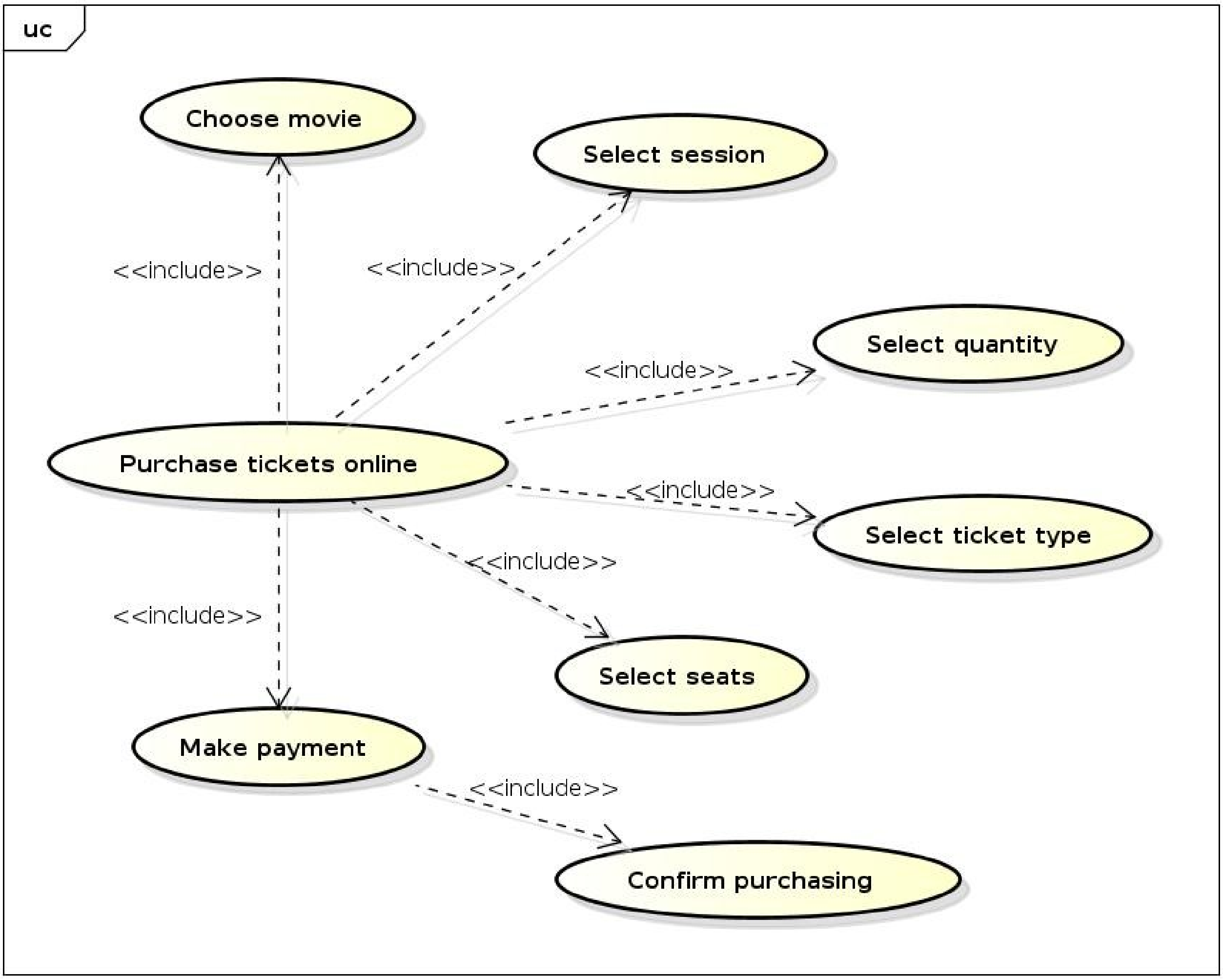
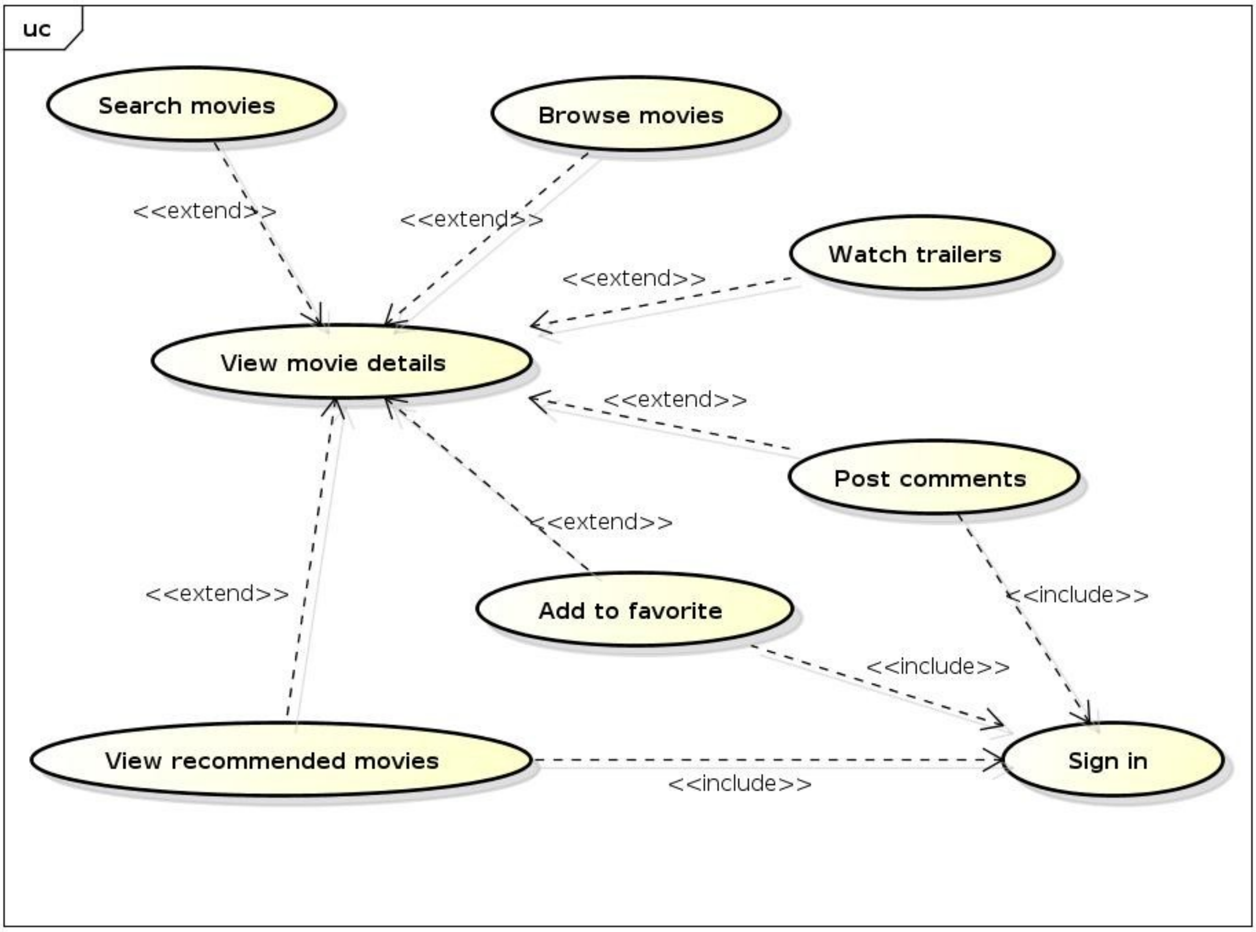
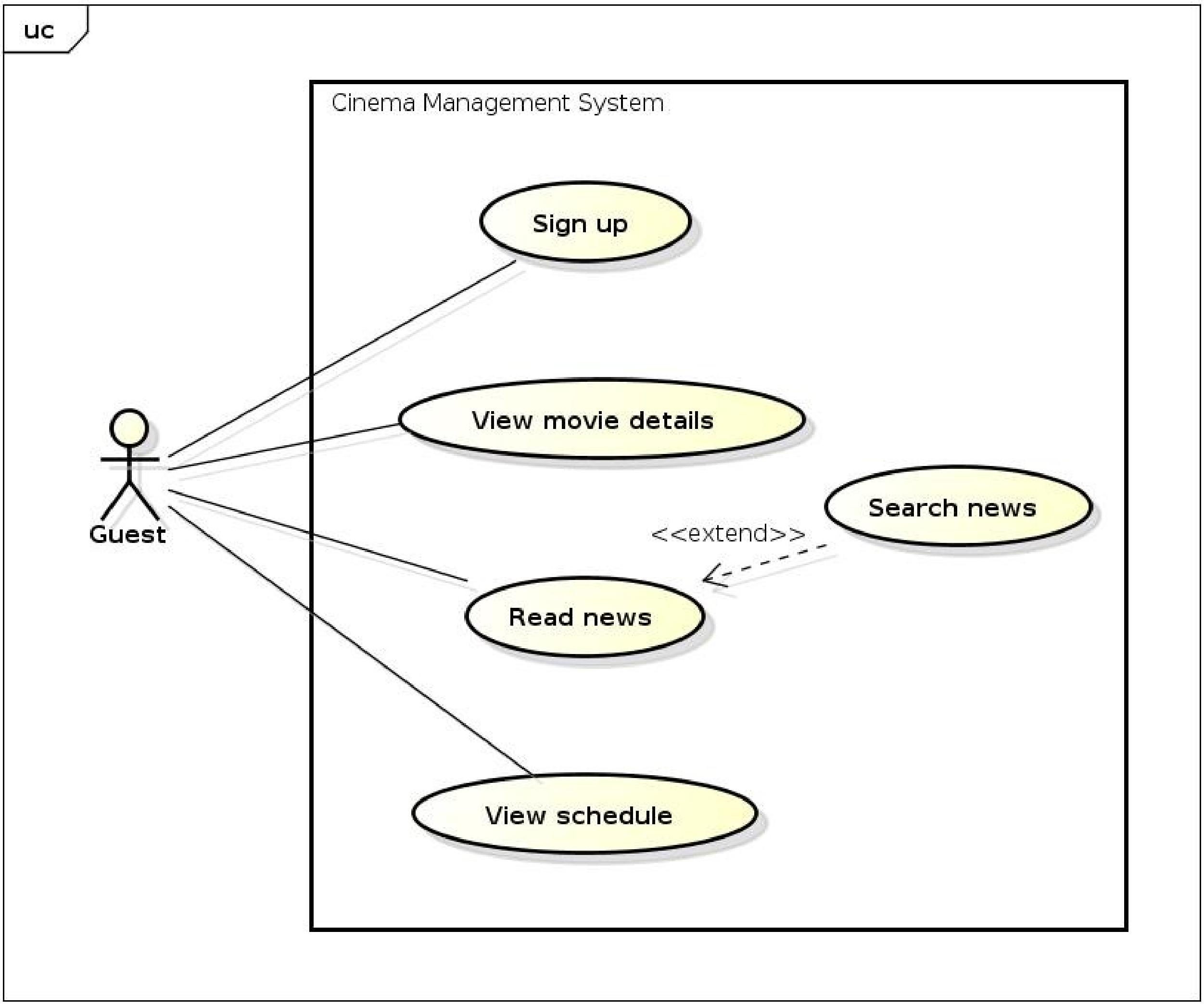


Figure 15: UC Member and Guest: View movie details



## Guest

Figure 16: Use cases for guest



## Use case specification

**List of available use case specifications:**

1. Add new movie
2. Update Movie
3. Delete Movie
4. View showing
5. Add screen
6. Update screen
7. Delete screen
8. Add moderator
9. Delete moderator
10. Update Moderator
11. Search
12. Add FAQ
13. Update FAQ
14. Delete FAQ
15. Add schedule
16. Update schedule
17. Delete schedule
18. Block member
19. Add showing room
20. Update showing room
21. Delete showing room
22. Search movie
23. Create account
24. Book ticket
25. View purchased history
26. Update profile
27. Change password
28. View movie detail

| | | | |
|---|---|---|---|
| **Use Case Name:** Add new movie | | **ID:** | **Priority:** |
| **Actor :** Administrator | | | |
| **Description:** The use case describe how the administrator add new movie | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1 Movie datastore is available on-line<br>2 Poster datastore is available on-line | | | |
| **Normal Course:**<br>**1.** Administrator specify the movie’s information by filling the form and then submit.<br>**2.** When movie added successfully, the new movie will be stored in movie database and the system will display successful notification.<br>**3.** The administrator will be asked to upload the poster for the new movie.<br>**4.** For each poster uploaded successfully, the new poster will be stored in poster database and the system will notify the administrator . | | **Information for Steps:**<br>-movie title, movie summary, movie description, movie duration, movie cast list,movie director, movie age rate.<br>-Notification<br>-posters.<br>-Notification | |
| **Post-conditions:**<br>New movie is stored<br>New poster is stored | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name**: Update Movie | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>**1.** The admin is authenticated<br>**2**. The movie database is available on-line | | | |
| **Normal Course:**<br>**1** To update the movie, the admin click on the movie title link to transmit movie id.<br>**2** The system will search and display all existing movie information of inputted id in movie database and the admin can modify it.<br>**3** The admin can modify new information or change the status of the movie and submit it to the system<br>**4** When all information is valid, the system will show successful notification and new movie information of the input id in movie database will be updated. | | **Information for Steps:**<br>Movie Id<br>Existing movie information<br>New information<br>Successfully updating notification | |

| | | | |
|---|---|---|---|
| **Post-conditions:**<br>1. The selected movie in the database will be updated . | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name**: Delete Movie | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>**1.** The admin is authenticated<br>**2**. The movie database is available on-line | | | |
| **Normal Course:**<br>**1.** To delete the movie, the admin click on checkboxes in the list of movie to transmit the list of movie Id he/she want to delete.<br>**2.** The system will search and delete one by one movie in movie database.<br>**3.** If there is any movie that's showed already, the system won't let the admin to delete and show unsuccessful notification.<br>**4.** If theses movie haven't showed yet, the system will allow the admin to delete and display the successfully deleting notification. | | **Information for Steps:**<br>▪ Movie Ids<br>▪ Existing movies<br>▪ Unsuccessfully deleting notification<br>▪ Successfully deleting notification | |
| **Post-conditions:**<br>1. The movie in the database will be deleted . | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** View showing | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** the use case describe how to check room status | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. Schedule datastore is available online<br>2. Room-Movie datastore is available on-line<br>3. Seats datastore is available on-line<br>4.Ticket datastore is available on-line | | | |

<table>
<tr><td colspan="2"><b>Normal Course:</b><br>1. The system will require the admin to select the showing movie in the combo box to transmit the movie id.<br>2. The system will check the existing room that show the selected movie in Movie-Room database and show the list of showing Room in combo box for this movie<br>3. The system will continue requiring the admin to select the showing room for selected movie.<br>4. The system will check existing schedules of the selected room in Schedule database and show the list of schedules.<br>5. The system will require the admin to select the showing schedule<br>6. The system will display the map of seats of the rooms for that schedule with colored seats.</td><td colspan="2"><b>Information for Steps:</b><br>▪ Movie Id<br>▪ List of Showing Room<br>▪ Showing Room Id<br>▪ List of schedules<br>▪ Schedule Id<br>▪ Map of seats for selected schedule</td></tr>
<tr><td colspan="4"><b>Post-conditions:</b><br>1. The map of rooms with many status of seats will be showed</td></tr>
<tr><td colspan="4"><b>Summary</b></td></tr>
<tr><td><b>Inputs</b></td><td><b>Source</b></td><td><b>Outputs</b></td><td><b>Destination</b></td></tr>
</table>

<table>
<tr><td colspan="2"><b>Use Case Name:</b> Add screen</td><td><b>ID:</b></td><td><b>Priotity:</b></td></tr>
<tr><td colspan="4"><b>Actor :</b> Administrator</td></tr>
<tr><td colspan="4"><b>Description:</b> The use case describe how to add room</td></tr>
<tr><td colspan="4"><b>Trigger:</b><br><b>Type:</b></td></tr>
<tr><td colspan="4"><b>Pre-conditions:</b><br>1. The admin is authenticated<br>2. The room database is available on-line</td></tr>
<tr><td colspan="2"><b>Normal Course:</b><br><b>1.</b> To add 1 more room, the system requires the admin to input the room information including room name, room type.<br><b>2.</b> The system will check if the information is valid or not and check existing room name in room database<br><b>3.</b> If room name in database exist already, the system will notify the admin and asked him/her to change the name<br><b>4.</b> After finish creating room, the system will require the admin to create seats for the room. The admin will enter seat codes and each seat code must be unique.<br><b>5.</b> When all information is  valid, the system will show successful notification and the new room will be stored in room database and its status will be good.</td><td colspan="2"><b>Information for Steps:</b><br>▪ Room information<br>▪ Existing list of rooms<br>▪ Notification<br>▪ List of seat codes<br>▪ Successfully room-creating notification</td></tr>
<tr><td colspan="4"><b>Post-conditions:</b><br>1. A new room will be stored in database.<br>2. New seats will be stored in database.</td></tr>
<tr><td colspan="4"><b>Summary</b></td></tr>
<tr><td><b>Inputs</b></td><td><b>Source</b></td><td><b>Outputs</b></td><td><b>Destination</b></td></tr>
</table>

| | | | |
|---|---|---|---|
| **Use Case Name:** Update screen | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** The use case describe how to update room | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The admin is authenticated<br>2. The room database is available on-line | | | |
| **Normal Course:** | | **Information for Steps:** | |
| **1** To update the room, the admin click on the room name link to transmit room id. | | ▪ Room Id | |
| **2** The system will search and show the existing information of selected room in room database. | | ▪ Existing room information | |
| **3** If the admin want to add more seats, he/she click on function "add seat" and input the seat name. | | ▪ function<br>existing seat names | |
| **4** The system will check if the inputted seat name exists or not in seat database. | | ▪ successfully seat-adding notification | |
| **5** If the information is valid, the new seat will be stored in seat database of selected room and 1 notification will be sent to the admin. | | ▪ New information | |
| **6** The admin can modify new information or change the status of the room and submit it to the system | | ▪ Successfully updating notification | |
| **7** When all information is valid, the system will show successful notification and new information of selected room will be stored in room database | | ▪ | |
| Post-conditions:<br>1. New information of the chosen room will be updated. | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Delete screen | | **ID:** | **Priotity:** |
| **Actor :**Administrator | | | |
| **Description:** The use case describe how the admin to delete rooms | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The admin is authenticated<br>2. The room database is available on-line | | | |
| **Normal Course:** | | **Information for Steps:** | |
| **1.** To delete the room, the admin click on checkboxes in the list of room to transmit the list of movie Id he/she want to delete. | | ▪ Room Ids | |
| **2.** The system will search and delete one by one room in room database. | | ▪ Existing rooms | |
| **3.** If there is any room that's showed any movie already, the system won't let the admin to delete and show unsuccessful notification. | | ▪ Unsuccessfully deleting notification<br>Successfully deleting notification | |

| | | | |
|---|---|---|---|
| **4.** If theses rooms haven't showed yet, the system will allow the admin to delete and display the successfully deleting notification. | | | |
| **Post-conditions:** | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Add moderator | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>**1.** The admin is authenticated<br>**2.** The moderator database is available on-line | | | |
| **Normal Course:**<br>**1.** To add 1 more moderator, the admin needs to fill in the form with including information: username, password, firstname, lastname, address, phone ,email …<br>**2.** The system will check if the username and email address exist in the moderator database or not<br>**3.** If all information is valid, new moderator will be stored in moderator database. However, this moderator account haven't still logged in yet . 1 activating link will be sent to the moderator email<br>**4.** When the moderator click on this activating link, the system will update moderator status and the moderator can login their account | | **Information for Steps:**<br>Moderator Information<br>Existing moderator information<br>Activating Link<br>Moderator status | |
| **Post-conditions:**<br>1. New moderator will be stored in moderator database. | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Delete moderator | | **ID:** | **Priotity:** |
| **Actor : Administrator** | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The admin is authenticated<br>2. The moderator database is available on-line | | | |
| **Normal Course:**<br>**1.** The administrator clicks on checkbox to select the list of moderator he/she want to delete.<br>**2.** The system will find one by one moderator in the list of deleted moderator in the moderator database to delete.<br>**3.** If there is any moderator who work already, the system won't allow the admin to delete him/her and show unsuccessfully deleting notification.<br>**4.** As finished, new moderator database will be updated and the system will notify the admin. | | **Information for Steps:**<br>▪ List of moderator id<br>▪ Existing moderator id<br>▪ Unsuccessfully deleting notification<br>▪ Successfully deleting notification | |
| **Post-conditions:**<br>1. The moderators will be deleted in moderator database. | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

|  |  |  |  |
|---|---|---|---|
| **Use Case Name:** Update Moderator |  | **ID:** | **Priotity:** |
| **Actor : A**dministrator |  |  |  |
| **Description:** The use case describe how to update a moderator |  |  |  |
| **Trigger:**<br>**Type:** |  |  |  |
| **Pre-conditions:**<br>1. The admin is authenticated<br>2. The moderator database is available on-line |  |  |  |
| **Normal Course:**<br>**1.** To update the moderator, the admin click on the moderator name link to transmit moderator id.<br>**2.** The system will search and show the existing moderator information of selected moderator in moderator database.<br>**3.** The admin can modify with new information or change the status and submit it to the system<br>**4.** When all information is valid, the system will show a successful notification, one more notification will be sent to the moderator's email whose id is inputted id. New information for the inputted moderator's id will be updated in moderator database |  | **Information for Steps:**<br>▪ Moderator Id<br>▪ Existing moderator information<br>▪ New information<br>▪ Successfully updating notification |  |
| **Post-conditions:**<br>1. The selected movie in the database will be updated . |  |  |  |
| **Summary** |  |  |  |
| **Inputs** | **Source** | **Outputs** | **Destination** |

|  |  |  |  |
|---|---|---|---|
| **Use Case Name:** Check Income | | **ID:** | **Priotity:** |
| **Actor :** | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The movie database is available on-line.<br>2. The room database is available on-line.<br>3. The movie-room is available on-line.<br>4. The schedule database is available on-line.<br>5. The ticket database is available on-line. | | | |
| **Normal Course:**<br>**1.** The system will require the admin to select some options  check the income including: check by movie, check from date to date, check by schedule, check by room… in form.<br>**2.** The system will check if inputted information exist or not and calculate the total price for all existing ticket. | | **Information for Steps:**<br>▪ Search information<br>▪ Total price. | |
| **Post-conditions:**<br>1. The total pirce will be displayed | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Search | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** The use case describe how the administrator search information | | | |
| **Trigger:** **Type:** | | | |
| **Pre-conditions:**<br>Member datastore, Moderator datastore, Movie datastore are available on-line | | | |
| **Normal Course:** | | **Information for Steps:** | |
| The administrator will choose the category he/she want to search by like movie, member… | | Category | |
| If admin choose search movie: | | | |
| **a.** The system will ask admin to enter the title of the movie or key word of the title of the movie. | | Title movie/ some keyword of title of movie | |
| **b.** The system will check if the input name or words that matched with the existing movie title in movie datastore or not. | | Existing movie titles | |
| **c.** If the input name or words matches with any existing movie title in the movie datastore, the list of movie with similar title will be displayed | | List of movie having similar title | |
| If admin choose search moderator: | | | |
| **a.** The system require admin to enter the fullname/first name or lastname or moderator id | | Moderator Id | |
| **b.** if admin enter moderator id: | | | |
| o the system will check existing moderator id in moderator database. | | Existing moderator id | |
| o If the input id appear in the database, the system will display exactly the moderator the admin want to find. Otherwise, nothing display | | Moderator | |
| **c.** if admin enter name: | | Moderator name | |
| o the system will check exsiting moderator name in moderator database. | | Existing moderator names | |
| o The system will display list of moderators whose name are similar to input name. | | List of Moderator having similar names | |
| If admin choose search member: | | | |
| **a.** The system require admin to enter the member username | | Member Id | |
| **b.** The system will check existing member username in member database. | | Existing member Id | |
| **c.** If input member username exist in member database, The system will display exactly the member | | Member | |
| **Post-conditions:** | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Add FAQ | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** The usercase describe how to add FAQ | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The admin is authenticated.<br>2. FAQ database is available on-line. | | | |
| **Normal Course:**<br>**1.** To add new FAQ, the system will require the admin to fill in the form with question and answer of FAQ<br>**2.** New FAQ will be stored in FAQ database and the system will notify the admin with successful message. | | **Information for Steps:**<br>▪ FAQ information<br>▪ Successfully FAQ-creating notification. | |
| **Post-conditions:**<br>1. New FAQ will be stored in FAQ database. | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Update FAQ | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1. The admin is authenticated.<br>2. FAQ datastore is available on-line. | | | |
| **Normal Course:**<br>**1.** To update new FAQ, the admin will click on the FAQ name number to transmit the FAQ Id.<br>**2.** The system will search and display the information of inputted id in FAQ database.<br>**3.** The admin can modify the information by replacing the old value with new value<br>**4.** If all information is all valid, the system will show successful notification and the FAQ of inputted id will be updated. | | **Information for Steps:**<br>▪ FAQ Id<br>▪ FAQ information<br>▪ Modified information<br>▪ Successful notification | |
| **Post-conditions:** | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Delete FAQ | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description:** | | | |
| **Trigger:**<br>**Type:** | | | |

<table>
<tr><td colspan="4"><b>Pre-conditions:</b><br>1. The admin is authenticated.<br>2. FAQ datastore is available on-line.</td></tr>
<tr><td colspan="2"><b>Normal Course:</b><br><b>1.</b> To delete FAQ or list of FAQ, the admin will click on checkboxes in list of FAQ to transmit the FAQ Ids.<br><b>2.</b> The system will search and delete one by one FAQ in FAQ database.</td><td colspan="2"><b>Information for Steps:</b><br>FAQ Ids<br>Existing FAQ</td></tr>
<tr><td colspan="4"><b>Post-conditions:</b></td></tr>
<tr><td colspan="4"><b>Summary</b></td></tr>
<tr><td><b>Inputs</b></td><td><b>Source</b></td><td><b>Outputs</b></td><td><b>Destination</b></td></tr>
</table>

| | | | |
|---|---|---|---|
| **Use Case Name:** Add Schedule | | **ID:** | **Priotity:** |
| **Actor : Administrator** | | | |
| **Description:** The use case describe how to add new schedule. | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>1.The admin is authenticated<br>2. Schedule database is available on-line<br>3. room-movie is available on-line | | | |
| **Normal Course:** | | **Information for Steps:** | |
| **1.** The system requires admin to choose the movie by click on the movie title in combo box to transmit the movie id | | Movie id | |
| **2.** The system will find and display all of the room that is showing the inputted movie in room-movie database as combo box | | List of showing room | |
| **3.** The admin will click on his/her desired room to transmit the id | | Room-Movie Id | |
| **4.** The system will display all of occupied time and the function to add new schedule | | List of occupied time | |
| **5.** The admin will enter the time for new schedule | | Time | |
| **6.** The system will check if the inputted time is collided with the gap between starting time and ending time in schedule database | | List of starting time and ending time | |
| **7.** If the new schedule is collided, the system will send unsuccessful notification to admin | | Unsuccessful notification | |
| **8.** If the new schedule is available, the system will be stored it in schedule database and list of new ticket will be create in ticket database, with new status the price of ticket will base on the time. | | List of new tickets | |
| **Post-conditions:**<br>1. New Schedule will be stored<br>2. List of tickets will be stored | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name:** Update schedule | | **ID:** | **Priotity:** |
| **Actor : Administrator** | | | |
| **Description: The use case describe how to update the schedule** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>**1. The admin is authenticated**<br>**2. Schedule database is available on-line** | | | |
| **Normal Course:**<br>**1.** The admin will click on the schedule to transmit its id.<br>**2.** The system will search and show the information of inputted in schedule database.<br>**3.** The admin can change time and status of movie. The default status when created is new, and when its current time is greater than showing time, the status will be automatically updated to finished.<br>**4.** If all modified information is valid, the system will notify the system will automatically send mail to all the customer who booked tickets at for that schedule via email. | | **Information for Steps:**<br>▪ Schedule Id<br>▪ Schedule information<br>▪ Modified information<br>▪ Notification to email | |
| **Post-conditions:** | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

| | | | |
|---|---|---|---|
| **Use Case Name**: Delete Schedule | | **ID:** | **Priotity:** |
| **Actor :** Administrator | | | |
| **Description** | | | |
| **Trigger:**<br>**Type:** | | | |
| **Pre-conditions:**<br>**1.** The admin is authenticated<br>**2**. The schedule database is available on-line | | | |
| **Normal Course:**<br>**1.** To delete the schedule, the admin will click on checkboxes in the list of schedule to transmit the list of schedule Id he/she want to delete.<br>**2.** The system will search and delete one by one schedule in schedule database.<br>**3.** If there is any schedule that's bought tickets already, the system won't let the admin to delete and show unsuccessful notification.<br>**4.** If theses movie haven't showed yet, the system will allow the admin to delete and display the successfully deleting notification. | | **Information for Steps:**<br>▪ Schedule Ids<br>▪ Existing schedules<br>▪ Unsuccessfully deleting notification<br>▪ Successfully deleting notification | |
| **Post-conditions:**<br>2. The movie in the database will be deleted . | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |

<table>
<tr><td colspan="2">**Use Case Name:** Ban Member</td><td>**ID:**</td><td>**Priotity:**</td></tr>
<tr><td colspan="4">**Actor :** Administrator</td></tr>
<tr><td colspan="4">**Description:** The use case describe how admin ban members</td></tr>
<tr><td colspan="4">**Trigger:**<br>**Type:**</td></tr>
<tr><td colspan="4">**Pre-conditions:**<br>**1.** The admin is authenticated<br>**2.** The member database is available on-line</td></tr>
<tr><td colspan="2">**Normal Course:**<br>**1.** To ban members, the system will require the admin to member username. Members who were banned already won appear their name in the list.<br>**2.** The status of member will be turn to banned and when member try to login their account, the system will show their status.</td><td colspan="2">**Information for Steps:**<br>▪ Member username<br>▪ Notification for being banned</td></tr>
<tr><td colspan="4">**Post-conditions:**</td></tr>
<tr><td colspan="4">**Summary**</td></tr>
<tr><td>**Inputs**</td><td>**Source**</td><td>**Outputs**</td><td>**Destination**</td></tr>
</table>

<table>
<tr><td colspan="2">**Use Case Name**: Add Showing Room</td><td>**ID:**</td><td>**Priotity:**</td></tr>
<tr><td colspan="4">**Actor :** Administrator</td></tr>
<tr><td colspan="4">**Description:** The use case describe how the admin create showing room</td></tr>
<tr><td colspan="4">**Trigger:**<br>**Type:**</td></tr>
<tr><td colspan="4">**Pre-conditions:**<br>1. Room database is available on-line<br>2. Movie database is available on-line</td></tr>
<tr><td colspan="2">**Normal Course:**<br>**1.** The system will display a list of rooms with good status in room database as combo box for the admin to select.<br>**2.** The system will display a list of all movie with showing status in movie database as combo box.<br>**3.** The admin will choose his/her options for room and movie<br>**4.** The system will check if the selected room showing the selected movie exist or not in the movie-room database or not.<br>**5.** If existed, the system will display notification for failing to create a new showing.<br>**6.** If not, the new showing will be stored in movie-room database with showing status and the system will display successful notification</td><td colspan="2">**Information for Steps:**<br>▪ List of rooms with good status<br>▪ List of movies with showing status<br>▪ Slected movie and rooms<br>▪ List of showing rooms with status showing<br>▪ Unsuccessful notification<br>successful notification</td></tr>
<tr><td colspan="4">**Post-conditions:**<br>**1. New showing room is stored in movie-room database**</td></tr>
<tr><td colspan="4">**Summary**</td></tr>
<tr><td>**Inputs**</td><td>**Source**</td><td>**Outputs**</td><td>**Destination**</td></tr>
</table>

|  |  |  |  |
|---|---|---|---|
| **Use Case Name:** Update showing room |  | **ID:** | **Priotity:** |
| **Actor : Admin** |  |  |  |
| **Description:** The use case describe how the admin to update showing rooms |  |  |  |
| **Trigger:**<br>**Type:** |  |  |  |
| **Pre-conditions:**<br>1. Movie-room database is available on-line |  |  |  |
| **Normal Course:**<br>**1.** The system will display a list of showing movie based on the way admin selected a status (all,showing,fisnished).<br>**2.** The admin can change the status of 1 showing room and its new status will be updated in movie-room database. |  | **Information for Steps:**<br>▪ List of showing movies<br>▪ Status of showing room |  |
| **Post-conditions:** |  |  |  |
| **Summary** |  |  |  |
| **Inputs** | **Source** | **Outputs** | **Destination** |

|  |  |  |  |
|---|---|---|---|
| **Use Case Name**: Delete showing room |  | **ID:** | **Priotity:** |
| **Actor :** Administrator |  |  |  |
| **Description** |  |  |  |
| **Trigger:**<br>**Type:** |  |  |  |
| **Pre-conditions:**<br>**1.** The admin is authenticated<br>**2**. The movie-room database is available on-line |  |  |  |
| **Normal Course:**<br>**1.** To delete the showing room, the admin will click on checkboxes in the list of schedule to transmit the list of showing room Id he/she want to delete.<br>**2.** The system will search and delete one by one showing room in movie-room database.<br>**3.** If there is any showing room that's has schedules already, the system won't let the admin to delete and show unsuccessful notification.<br>**4.** If theses movie haven't showed yet, the system will allow the admin to delete and display the successfully deleting notification. |  | **Information for Steps:**<br>▪ Showing room Ids<br>▪ Existing showing room<br>▪ Unsuccessfully deleting notification<br>▪ Successfully deleting notification |  |
| **Post-conditions:**<br>3. The movie in the database will be deleted . |  |  |  |
| **Summary** |  |  |  |
| **Inputs** | **Source** | **Outputs** | **Destination** |

|  |  |  |  |
|---|---|---|---|
| **Use Case Name:** Search Movie |  | **ID:** | **Priority:** |
| **Actor : Guests/Members** |  |  |  |
| **Description:** This use case describes how the end-user search their desired movie. |  |  |  |

**Trigger:**
**Type:**

**Preconditions:**

**1.** Movie database is available and on-line

| **Normal Course:** | **Information for Steps:** |
|---|---|
| **5.** Customers specify the way they want to search· by choosing showing time, movie. | Search Selection |
| **6.** Based on the selected way, the system will show text for searching movie by name or time options for searching movie by time. | |
| **7.** If search by movie, Customer will be ask to enter movie name. | Movie name |
| **8.** The system will check for any movie that matches with the input name. | Existing movie |
| **9.** If there are any movies with the name corresponding with the input name, the system will display list of movie with this name. Otherwise, nothing displays. | List of movie |
| **10.** If search by time, Customer will be asked to select date and time slot. | Date, time |
| **11.** The system will check if there are any existing movies showing at input time | Existing movie |
| **12.** The list of movies at input time will be displayed if there is existing movie at that time. Otherwise, nothing displays | List of movie |

**Post conditions:**

3 List of movie is displayed

**Summary**

| **Inputs** | **Source** | **Outputs** | **Destination** |
|---|---|---|---|
| Search Selection<br>Movie name<br>Existing movie<br>Date, time | Customers<br>database | List of Movie | |

| | | | |
|---|---|---|---|
| **Use Case Name:** Create Account | | **ID:** | **Priority:** |
| **Actor :** Customers | | | |
| **Description:** This use case describe how to create a new account for customers | | | |
| **Trigger:**<br>**Type:** | | | |
| **Preconditions:**<br>**1.** Customers database is available and on-line | | | |
| **Normal Course:** | | **Information for Steps:** | |
| Customers will specify their information including: username, password, first name, last name, address, email, phone, id card… | | Customer’s information | |
| The system will check the existing username in customers account. | | Existing username | |
| The system will check the existing email in customers account. | | Existing email | |
| If inputted username or email existed in the system. The system will notify the customers and ask them to enter again. | | Notification | |
| If all information are valid, the system will automatically send the activate link to customer’s email. | | Activating link | |
| As the customers enter activating link, the system will notify congratulation message and customers can login their account | | Notification | |
| **Post conditions:**<br>1 New customer is stored. | | | |
| **Summary** | | | |
| **Inputs** | **Source** | **Outputs** | **Destination** |
| Customer’s information<br>Existing username<br>Existing email | | Notification<br>Activating link<br>Notification | |

| | | | |
|---|---|---|---|
| **Use Case Name:** Book ticket | **ID:** | **Priority:** | |
| **Actor : Customers** | | | |
| **Description: This use case describe how to book ticket online** | | | |
| **Trigger:** **Type:** | | | |
| **Preconditions:**<br>**1.** Member is authenticated<br>**2.** ticket database is available and on-line<br>**3.** Customer database is available an on-line | | | |
| **Normal Course:** | **Information for Steps:** | | |
| The system will require the member to choose a movie from list of current showing movies | Movie selection | | |
| The system will ask the member to choose the date and time they would like to watch | Date and time | | |
| The system will show the list of simulated seat positions on the screen that look like real positions. Red seats are occupied already, and white seats are still available. | List of seats | | |
| The customer chooses their desired seats by clicking on it. It just only work with available seats. | Desired seats | | |
| The system will send the customers the confirmation for his/her selection. | Confirmation | | |
| The system will display the information of customer’s booking and show the total money. | Booking information | | |
| When customers agree to make payment. He will be offered the way to pay, paying online or paying later at ticket sellers. | Paying option | | |
| If customers choose paying online: | | | |
| They will be forwarded to paying agency. They will make their payment there. | | | |
| When they finish their transaction, the system will store their booking information and update paying status of ticket database into sold status. | Paying status | | |
| The system will send confirmation code and notification though customer email. | Confirmation code and notification | | |
| If customers choose paying later: | | | |
| **d.** The system will store their booking information and paying status. The paying status in ticket database will be updated into booked status. | Paying status | | |
| **e.** The system will also send confirmation code to customers email. | Confirmation code | | |
| **f.** The system will send the notification that require customers to get tickets they bought before 1 hours the movie play. If customers don’t come at that time, ticket database will return to available status and their confirmation code after that time will be invalid. | Notification | | |
| **Post conditions:** ticket data store will be updated | | | |
| **Summary** | | | |
| **Inputs**<br>Movie Selection<br>Date and time<br>Selected seats | **Source** | **Outputs**<br>List of seats<br>Confirmation<br>Notification | **Destination** |

| | | | |
|---|---|---|---|
| Payment method | | | |
| **Use Case Name:** View purchased history | **ID:** | **Priority:** | |
| **Actor : Customers** | | | |
| **Description:** The use case describe how customers see purchased history. | | | |
| **Trigger:**<br>**Type:** | | | |
| **Preconditions:**<br>**1.** Member is authenticated<br>**2.** Ticket database is available and on-line<br>**3.** Member database is available and on-line | | | |
| **Normal Course:**<br>**d.** Based on the id of member , the system will find tickets they bought in ticket database.<br>**e.** The system will check existing booking customer in ticket database<br>**f.** If the system can find any member id in ticket database, it will show list of ticket bough by that member .<br>**g.** Each ticket in list of tickets will have different Id; the end-user will enter its id by clicking on it.<br>**h.** For each ticket Id, the system will find information including movie title, day, time, and price. | **Information for Steps:**<br>Member Id<br>Customer booking<br>List of ticket<br>Ticket Id<br>Information of the ticket. | | |
| **Post conditions:**<br>**8** List of tickets bought by current user will be displayed | | | |
| **Summary** | | | |
| **Inputs**<br>Member Id<br>Customer booking<br>Ticket Id | **Source** | **Outputs**<br>List of ticket<br>Information of the ticket. | **Destination** |

| | | |
|---|---|---|
| **Use Case Name:** Update profile | **ID:** | **Priority:** |
| **Actor : Customers** | | |
| **Description: The use case show how to modify account** | | |
| **Trigger:**<br>**Type:** | | |
| **Preconditions:**<br>**1.** Member is authenticated<br>**2.** Member database is available and on-line | | |
| **Normal Course:**<br>To modify the account, customer need to click on their name in hyperlink to transmit his/her id<br>Based on the member Id, the system will find all information of this member id but user id in member database<br>The customer will be able to change their present information by entering new information. The system will check if these new information are | **Information for Steps:**<br>Customer Id<br>Member information<br>New information | |

| | | | |
|---|---|---|---|
| valid or not. If valid, these information will be updated in member database<br>If updating successfully, The system will send notification for the customer. | | Notification | |
| **Post conditions:**<br>3. The user database with new information will be update | | | |
| **Summary** | | | |
| **Inputs**<br>Customer Id<br>New information | **Source** | **Outputs**<br>Member information<br>Notification | **Destination** |

| | | |
|---|---|---|
| **Use Case Name:** View movie detail | **ID:** | **Priority:** |
| **Actor : All** | | |
| **Description: The use case describe how the user view the information of 1 movie** | | |
| **Trigger:**<br>**Type:** | | |
| **Preconditions:**<br>1. The movie database is available on-line | | |
| **Normal Course:**<br>**d.** To view information of a movie, user needs to click on movie title in hyperlink to transmit the movie id.<br>**e.** Based on the inputted movie id, the system will search, display existing movie information of inputted id and the use case will end. | **Information for Steps:**<br>Movie Id<br><br>Existing movie information | |
| **Post conditions:**<br>4. Existing information of inputted id movie will be display. | | |

| | | | |
|---|---|---|---|
| **Summary** | | | |
| **Inputs**<br>Movie Id | **Source** | **Outputs**<br>Existing movie information | **Destination** |

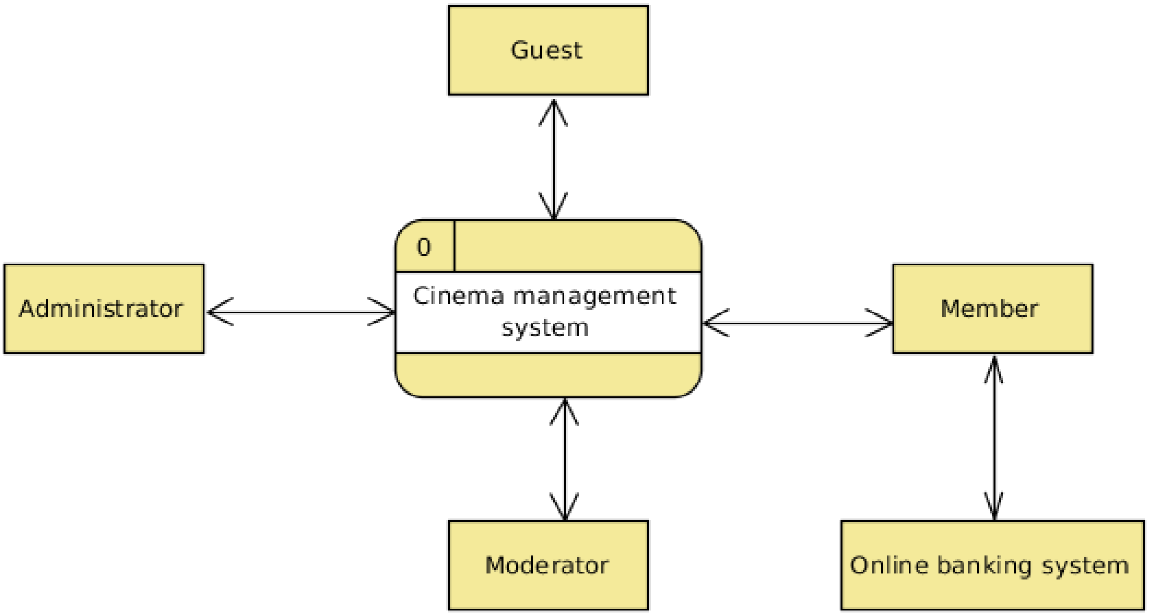
| | | |
|---|---|---|
| **Use Case Name:** Change Password | **ID:** | **Priority:** |
| **Actor :** All | | |
| **Description:** The use case describe how one user change his/her password | | |
| **Trigger:**<br>**Type:** | | |
| **Preconditions:**<br>**7.** The user is authenticated<br>**8.** The user database is available on-line | | |
| **Normal Course:**<br>**3.** The user need to enter his/her current password to confirm it first and then enter new password and confirmed password.<br>**4.** Based on the inputted current password, the system will search if this password exists in | **Information for Steps:**<br>Current password, new and confirmed new password<br><br>Current password | |

<table>
<tr><td colspan="2">user database or not.<br>5. If the current password is different from the inputted current password, the system will display notification for wrong password and the use case will finish.<br>6. If the inputted current password is correct, the system will update the password into the inputted new password and show the notification for successful process and end the use case.</td><td colspan="2">Notification for wrong password<br>Successful notification for changing the password</td></tr>
<tr><td colspan="4"><b>Post conditions:</b><br>5. New password of current user will be updated</td></tr>
<tr><td colspan="4"><b>Summary</b></td></tr>
<tr><td><b>Inputs</b><br>Current password, new and confirmed new password<br>Successful notification for changing the password</td><td><b>Source</b></td><td><b>Outputs</b><br>Current password<br>Notification for wrong password</td><td><b>Destination</b></td></tr>
</table>

# Process Modeling

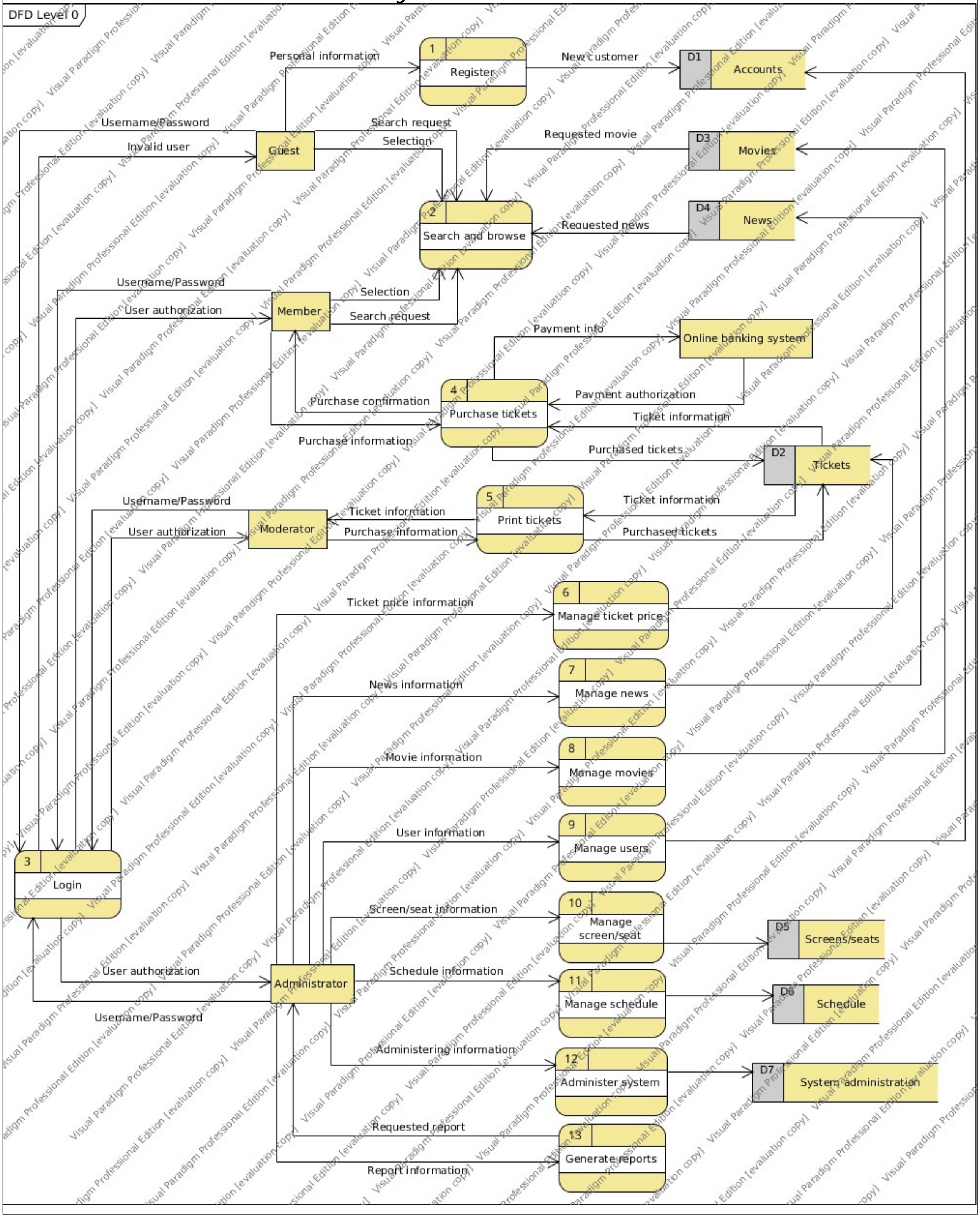
## Context diagram

Figure 17: Context diagram



# Data flow diagram

Figure 18: DFD level 0



# Data modeling

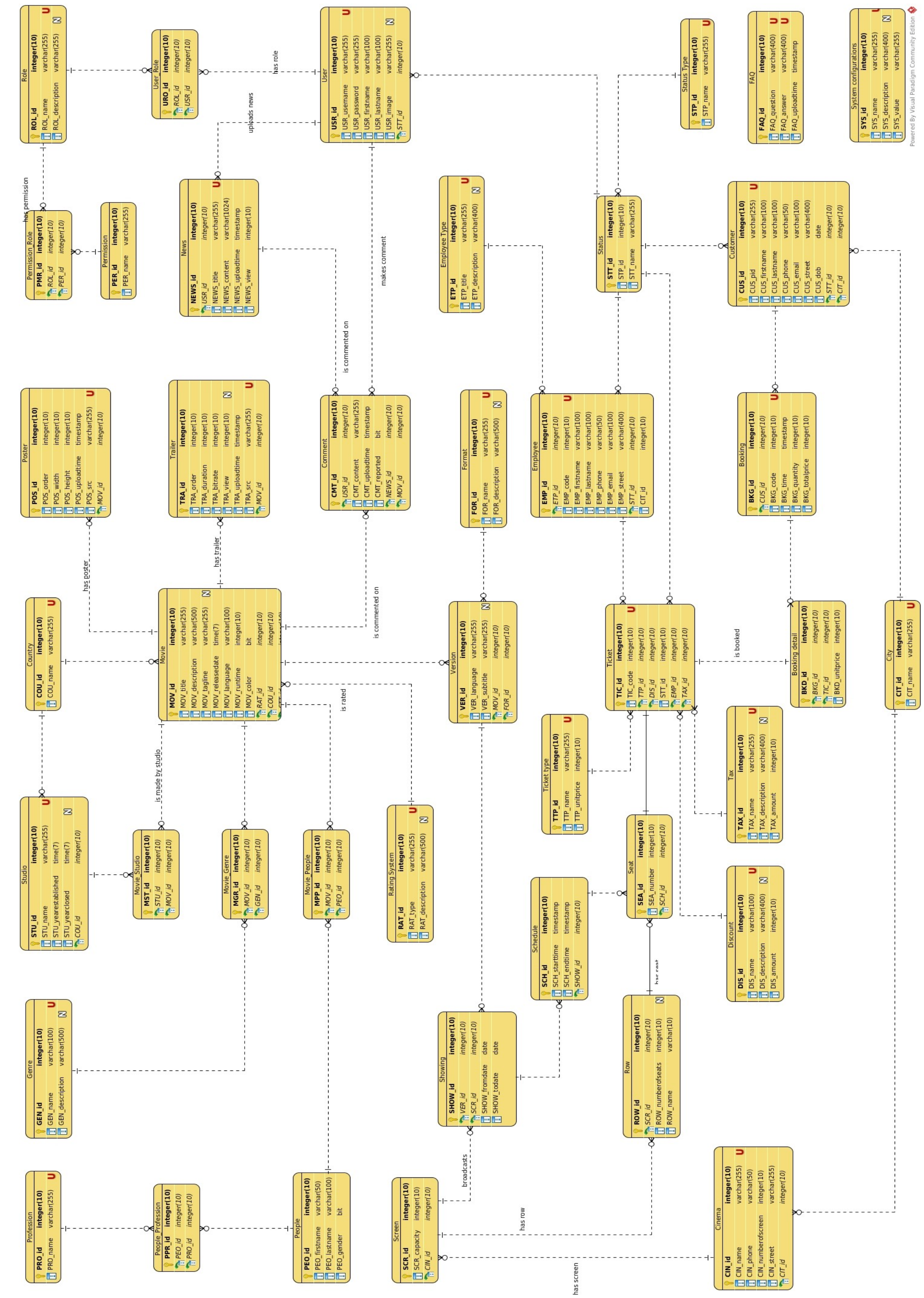


Figure 19: ERD global view

Figure 20: ERD: Movie information

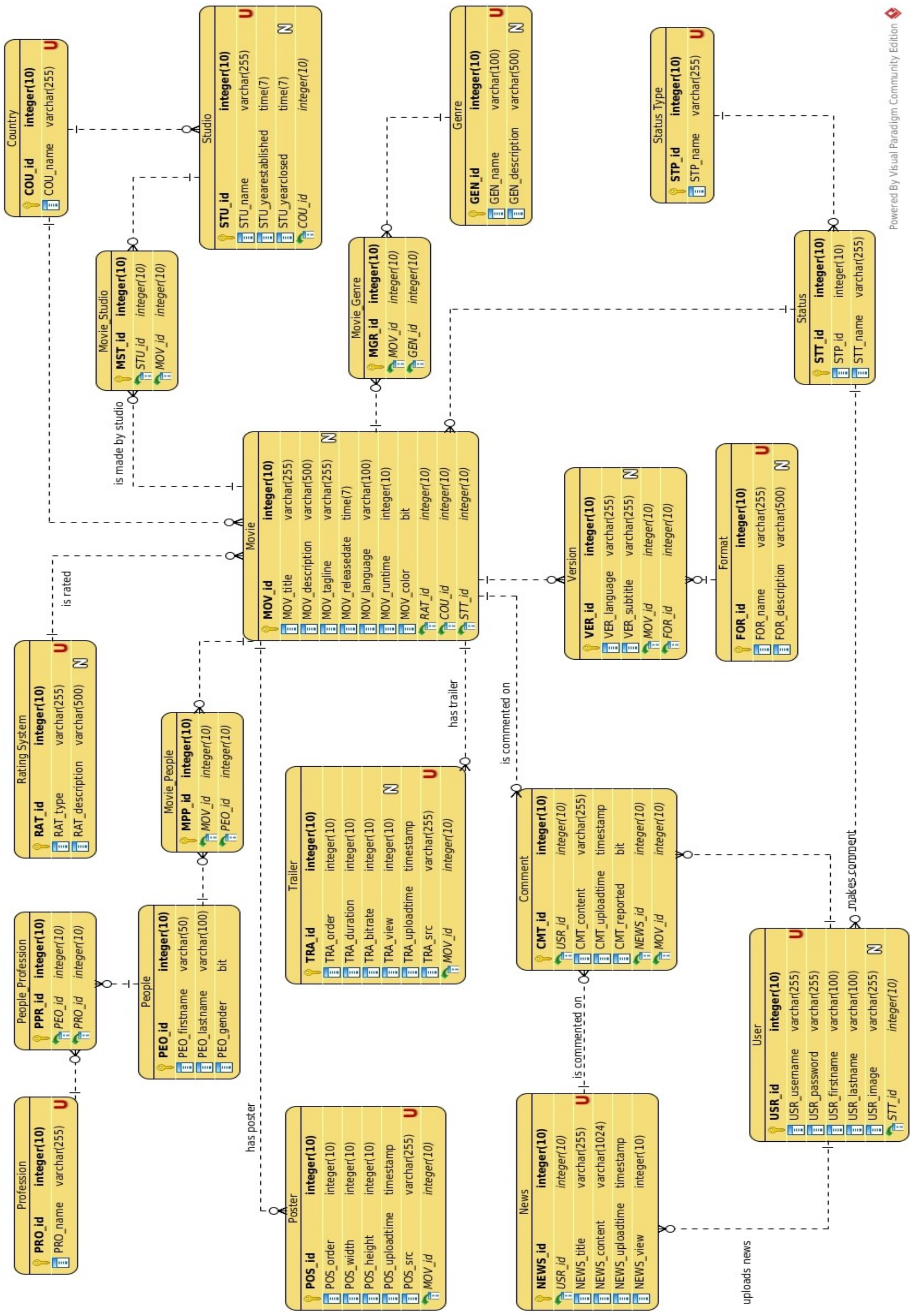


Figure 21: ERD: Movie schedule and booking

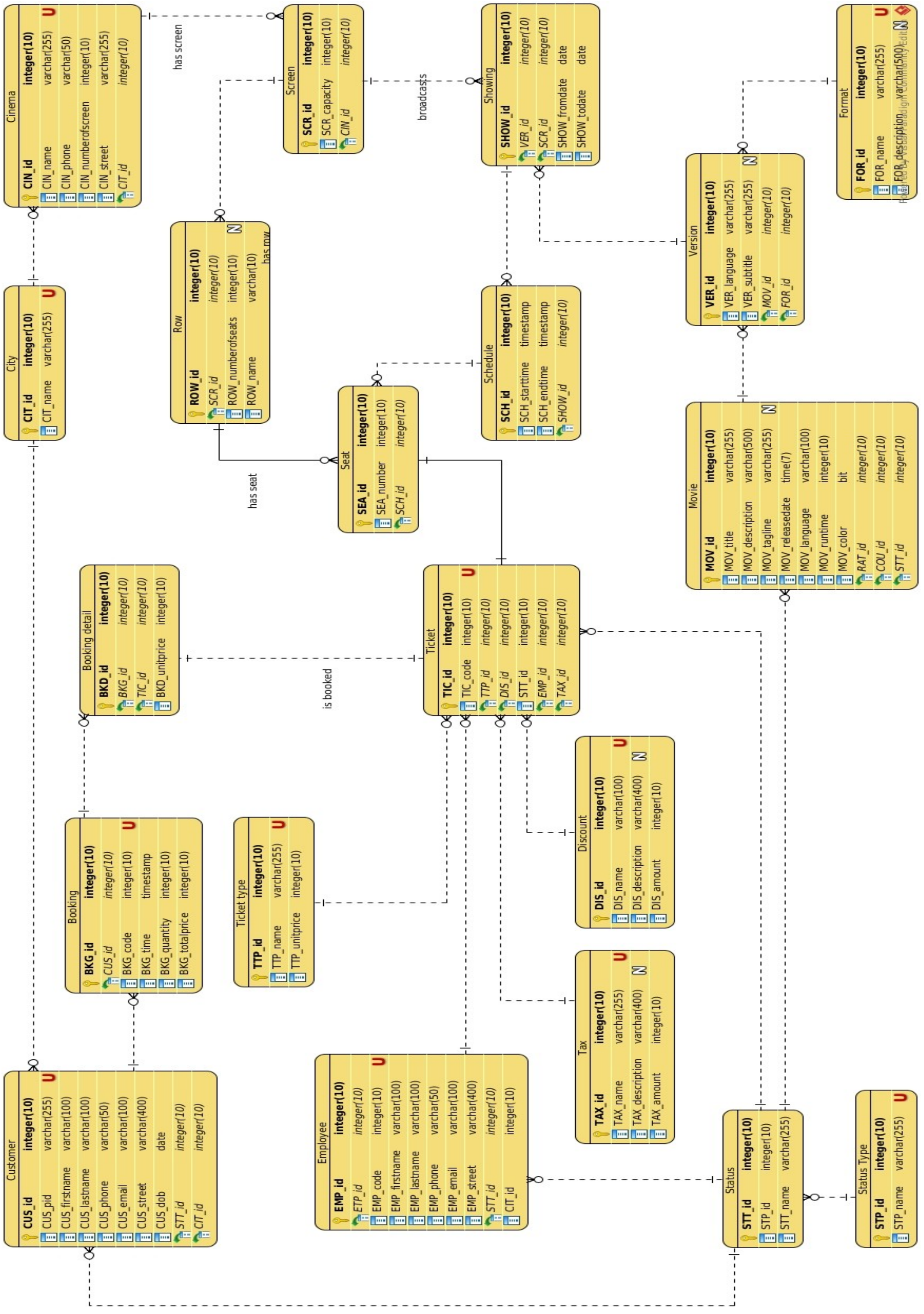
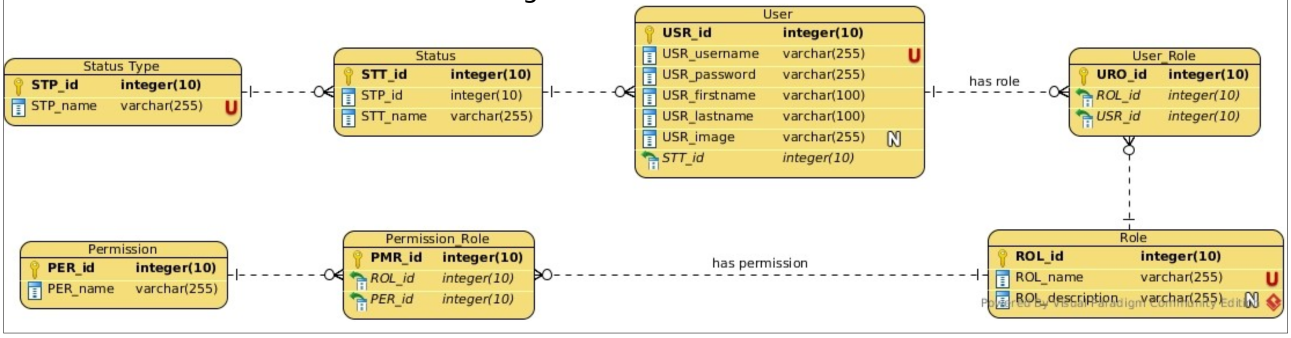


Figure 23: ERD: Users



# Design phase

## Acquisition Strategy

To implement this project we choose custom development because of the following:

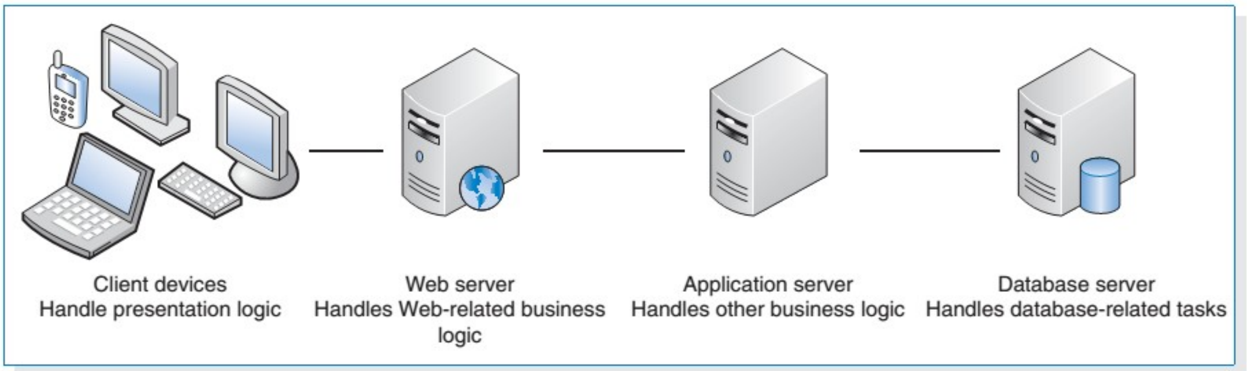
A custom application would be easier to change to include components that take advantage of current technologies that can support such strategic efforts.

Custom development also allows developers to be flexible and creative as well as builds technical skills and functional knowledge among the developers.

## Architecture Design

We use *n-tiered architecture (specifically 4-tiered architecture)* for our project since it distributes the work of the application among multiple layers of more specialized server computers and thus balance the load on the different servers better. This architecture is scalable and reliable. It is also open for further development.

Figure 24: Architectural structure


Client devices
Handle presentation logic
Web server
Handles Web-related business logic
Application server
Handles other business logic
Database server
Handles database-related tasks

Based on the above architecture, we suggest the following hardware and software specifications:

| | **Standard Client** | **Standard Web Server** | **Standard Application Server** | **Standard Database Server** |
|---|---|---|---|---|
| Operating system | Windows<br>Mozilla | Linux | Linux | Linux |
| Special software | Real Audio<br>Adobe Acrobat Reader | Apache | Java | Oracle |

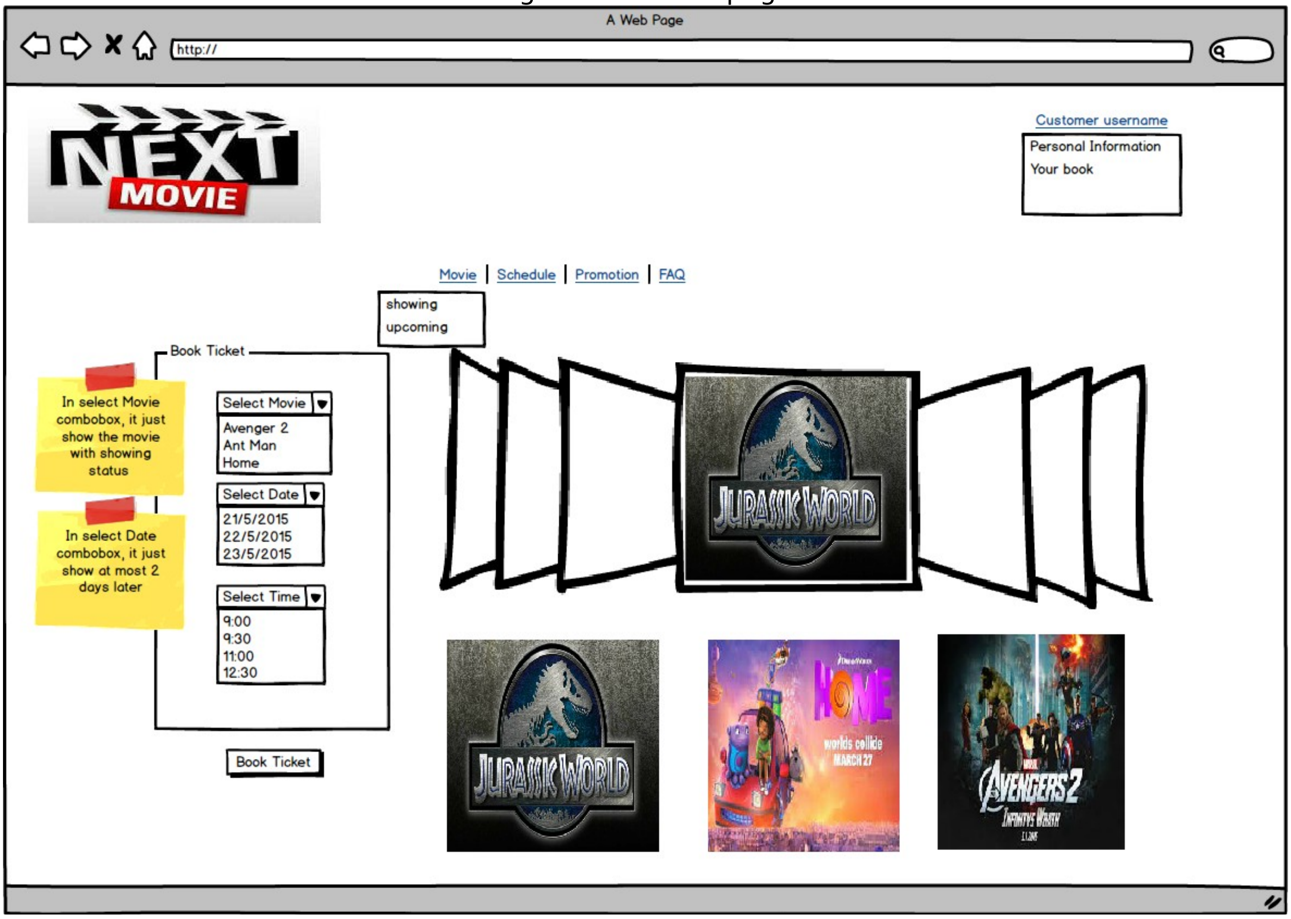
|  | Standard Client | Standard Web Server | Standard Application Server | Standard Database Server |
|---|---|---|---|---|
| Hardware | 250-GB disk drive<br>Intel ®- Core TM i3-2100 dual core processor<br>14-inch LCD Monitor | 500-GB disk drive<br>Dual-core Xeon | 160-GB disk drive<br>Quad-core Xeon | 1-TB disk drive<br>RAID<br>Quad-core Xeon |
| Network | Always-on Broadband, preferred<br>Dial-up at 56 Kbps, possible with some performance loss | Dual 100 Mbps Ethernet | Dual 100 Mbps Ethernet | Dual 100 Mbps Ethernet |

Table 4: Hardware and software specifications

## User Interface Design

(See mockup folder for more protoypes)

Figure 25: Home page

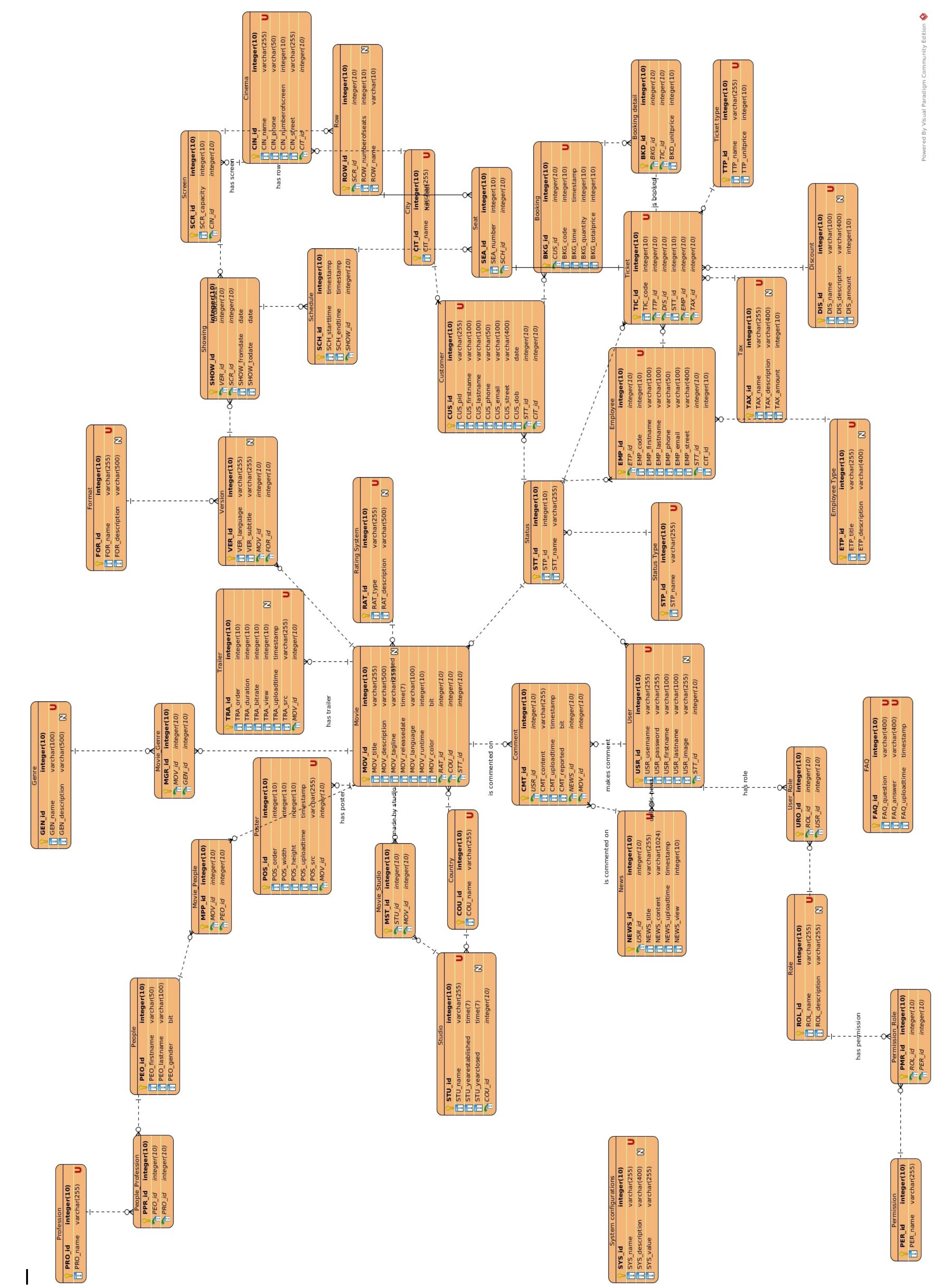


In select Movie combobox, it just show the movie with showing status

In select Date combobox, it just show at most 2 days later

## Data Storage Design

Figure 26: Physical Data Mode


Cinema
Row
Screen
City
Seat
Booking
Booking detail
Ticket type
Discount
Ticket
Tax
Employee
Employee Type
Customer
Schedule
Showing
Version
Format
Rating System
Status
Status Type
Trailer
Movie
Movie Genre
Genre
User
Comment
User Role
FAQ
Poster
Movie People
Movie Studio
Country
Studio
News
Role
Permission Role
Permission
People
People Profession
Profession
System configurations
has screen
has row
is booked
has trailer
has poster
made by studio
is commented on
makes comment
is commented on
has role
has permission
Powered By Visual Paradigm Community Edition