

Dev Rai - 2024301022

TE COMP – Division D, Batch D

SE LAB

Experiment 2: UML Diagrams

Use Case Diagram:

Table 1: List of Actors

Actor	Description
User	A customer of the theatre. A User can register for a new account, log in, and manage their profile. Their primary functions include browsing the list of available movies, watching trailers, and booking tickets for a screening. After booking, they can print their e-ticket. They also have the ability to reset their password if forgotten.
Admin	An employee with administrative privileges who manages the backend of the system. The Admin is responsible for managing the movie catalog, including adding or deleting movies. They also manage movie screenings (showtimes). A key role for the Admin is managing user accounts and viewing/exporting system reports for business analysis.

Table 2: List of Use - Cases

#	Use Case	Description
UC1	Register	Allows a new user to create a personal account in the system by providing necessary details like name, email, and password.
UC2	Login	Enables existing Users and Admins to access their respective dashboards by authenticating with their credentials.
UC3	Manage Profile	Allows a logged-in User to view and update their personal information, such as name, contact details, or password.
UC4	Forgot Password	An optional function for a User who cannot remember their password. This use case initiates the password recovery process, typically by sending a verification link to the user's registered email.
UC5	Reset Password	The second step in password recovery. After verifying their identity via the "Forgot Password" process, the User can set a new password for their account.
UC6	Browse Movies	Allows a User to view the list of currently showing or upcoming movies, read their descriptions, see ratings, and check showtimes.
UC7	Watch Trailer	An optional feature that extends "Browse Movies". While viewing a movie's details, a User can choose to watch its promotional trailer.
UC8	Book Ticket	The core function for a User to select a movie, a specific screening (date and time), choose their seats, and proceed to checkout. This process requires payment to be completed.
UC9	Process Payment	A mandatory part of the "Book Ticket" process. This use case handles the financial transaction, allowing the user to enter payment details and confirming the payment.
UC10	Print E-Ticket	An optional function that extends "Book Ticket". After a successful booking, the User can choose to view and print a digital copy of their ticket.
UC11	Manage Movies	An Admin-only function to manage the entire movie catalog. This includes adding new movies, editing existing movie details (e.g., synopsis, cast), and removing movies from the system.
UC12	Delete Movie	An optional function that extends "Manage Movies". It allows an Admin to permanently remove a specific movie record from the system.
UC13	Manage Users	An Admin-only function to oversee all user accounts in the system, including viewing user details, deactivating accounts, or resetting user passwords.
UC14	Manage Screenings	An Admin-only function to create, update, or delete movie showtimes. This includes assigning a movie to a specific screen, setting the date, and time.
UC15	Cancel Screening	An optional function that extends "Manage Screenings". It allows an Admin to cancel a specific upcoming showtime, which may involve notifying affected customers.

UC16	View Report	An Admin-only function to generate and view various system reports, such as sales reports, booking statistics, and movie performance metrics.
UC17	Export Report	An optional feature that extends "View Report". While viewing a report, an Admin can choose to export the data into a file format (e.g., CSV, PDF) for offline analysis or record-keeping.

Use Case Scenarios:

Table 3: UC1 – User Registration

Use Case:	UC1. User Registration	
Goal:	To create a new user account in the Theatre Management System.	
Actors:	User	
Pre-condition:	The user is not registered in the system.	
Post-condition:	The user's account is created, and they are ready to log in.	
Mainline Scenario:	Actor Actions 1. The new user accesses the system's homepage and selects the "Register" option. 3. The user provides required information (name, email, password) and confirms the registration.	System Actions 2. The system displays a registration form. 4. The system validates the provided information. 5. If validation is successful, the user's account is created, and the system displays a success message.
Alternate Flows:	3a. If the provided email address is already registered, the system displays an error message prompting the user to choose a different one.	

Table 4: UC2 – Login

Use Case:	UC2. Login
Goal:	To gain authenticated access to the system.

Actors:	User, Admin											
Pre-condition:	The actor has a registered account.											
Post-condition:	The actor is successfully logged in and gains access to their respective dashboard.											
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The registered actor accesses the "Log In" page.</td> <td></td></tr> <tr> <td>2. The actor provides their registered email address and password.</td> <td></td></tr> <tr> <td>3. The actor confirms the login.</td> <td>4. The system validates the provided credentials.</td></tr> <tr> <td></td> <td>5. If validation is successful, the actor is logged in, and they are redirected to their dashboard.</td></tr> </tbody> </table>	Actor Actions	System Actions	1. The registered actor accesses the "Log In" page.		2. The actor provides their registered email address and password.		3. The actor confirms the login.	4. The system validates the provided credentials.		5. If validation is successful, the actor is logged in, and they are redirected to their dashboard.	
Actor Actions	System Actions											
1. The registered actor accesses the "Log In" page.												
2. The actor provides their registered email address and password.												
3. The actor confirms the login.	4. The system validates the provided credentials.											
	5. If validation is successful, the actor is logged in, and they are redirected to their dashboard.											
Alternate Flows:	3a. If the provided credentials are incorrect, the system displays an error message.											

Table 5: UC3 – Manage Profile

Use Case:	UC3. Manage Profile									
Goal:	To allow a logged-in user to update their personal information.									
Actors:	User									
Pre-condition:	The user is logged into their account.									
Post-condition:	The user's profile information is successfully updated.									
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The user navigates to their "Profile" or "Account Settings" page.</td> <td>2. The system displays the user's current profile information in an editable form.</td></tr> <tr> <td>3. The user makes changes to their profile information and confirms the update.</td> <td>4. The system validates and saves the updated information to the database.</td></tr> <tr> <td></td> <td>5. The system displays a "Profile Updated" success message.</td></tr> </tbody> </table>	Actor Actions	System Actions	1. The user navigates to their "Profile" or "Account Settings" page.	2. The system displays the user's current profile information in an editable form.	3. The user makes changes to their profile information and confirms the update.	4. The system validates and saves the updated information to the database.		5. The system displays a "Profile Updated" success message.	
Actor Actions	System Actions									
1. The user navigates to their "Profile" or "Account Settings" page.	2. The system displays the user's current profile information in an editable form.									
3. The user makes changes to their profile information and confirms the update.	4. The system validates and saves the updated information to the database.									
	5. The system displays a "Profile Updated" success message.									

Alternate Flows:	3a. If the user tries to submit invalid information, the system displays an error message indicating the required corrections.
------------------	--

Table 6: UC4 – Forgot Password

Use Case:	UC4. Forgot Password									
Goal:	To initiate the password recovery process.									
Actors:	User									
Pre-condition:	The user is on the login page and does not remember their password.									
Post-condition:	A password reset link is sent to the user's registered email.									
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The user selects the "Forgot Password" option on the login page.</td> <td>2. The system displays a page asking for the user's email address.</td> </tr> <tr> <td>3. The user enters their registered email address and submits the form.</td> <td>4. The system verifies the email exists in the database.</td> </tr> <tr> <td></td> <td>5. The system generates and sends a unique password reset link to the user's email and displays a confirmation message.</td> </tr> </tbody> </table>		Actor Actions	System Actions	1. The user selects the "Forgot Password" option on the login page.	2. The system displays a page asking for the user's email address.	3. The user enters their registered email address and submits the form.	4. The system verifies the email exists in the database.		5. The system generates and sends a unique password reset link to the user's email and displays a confirmation message.
Actor Actions	System Actions									
1. The user selects the "Forgot Password" option on the login page.	2. The system displays a page asking for the user's email address.									
3. The user enters their registered email address and submits the form.	4. The system verifies the email exists in the database.									
	5. The system generates and sends a unique password reset link to the user's email and displays a confirmation message.									
Alternate Flows:	3a. If the provided email address is not found in the database, the system displays an error message.									

Table 7: UC5- Reset Password

Use Case:	UC5. Reset Password	
Goal:	To set a new password for the user's account.	
Actors:	User	
Pre-condition:	The user has clicked a valid password reset link received via email.	
Post-condition:	The user's password is changed.	

Mainline Scenario:	Actor Actions	System Actions
	1. The user clicks the reset link from their email.	2. The system validates the link and displays a form to set a new password.
	3. The user enters and confirms their new password.	4. The system validates that the passwords match and meet security requirements.
		5. If validation is successful, the system updates the user's password and redirects them to the login page with a success message.
Alternate Flows:	3a. If the passwords do not match or are too weak, the system displays an appropriate error message.	

Table 8: UC6 – Browse Movies

Use Case:	UC6. Browse Movies	
Goal:	To view the list of available movies and their details.	
Actors:	User	
Pre-condition:	The user is on the system's main page.	
Post-condition:	The user has viewed information about available movies.	
Mainline Scenario:	Actor Actions	System Actions
	1. The user navigates to the "Movies" or "Now Showing" section.	2. The system retrieves and displays a list of currently showing movies.
	3. The user selects a specific movie from the list.	4. The system displays a detailed page for the selected movie, including its synopsis, cast, and available showtimes.
Alternate Flows:	None	

Table 9: UC7 – Watch Trailer

Use Case:	UC7. Watch Trailer
Goal:	To watch the promotional trailer for a selected movie.

Actors:	User					
Pre-condition:	The user is viewing the details page for a specific movie (UC6).					
Post-condition:	The user has viewed the movie trailer.					
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The user selects the "Watch Trailer" option.</td> <td>2. The system opens a video player and begins streaming the trailer.</td> </tr> </tbody> </table>	Actor Actions	System Actions	1. The user selects the "Watch Trailer" option.	2. The system opens a video player and begins streaming the trailer.	
Actor Actions	System Actions					
1. The user selects the "Watch Trailer" option.	2. The system opens a video player and begins streaming the trailer.					
Alternate Flows:	1a. The "Watch Trailer" option is unavailable if no trailer is linked to the movie.					

Table 10: UC8 – Book Ticket

Use Case:	UC8. Book Ticket											
Goal:	To initiate the password recovery process.											
Actors:	User											
Pre-condition:	The user is logged in and has selected a movie and a specific showtime.											
Post-condition:	A ticket is successfully booked.											
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The user chooses a showtime for a movie.</td> <td>2. The system displays the seating chart for that screening.</td> </tr> <tr> <td>3. The user selects their desired seat(s).</td> <td>4. The system displays a summary of the selection and total cost.</td> </tr> <tr> <td>5. The user confirms to proceed to payment.</td> <td>6. Upon receiving payment success confirmation, the system finalizes the booking.</td> </tr> <tr> <td></td> <td>7. The system generates an e-ticket, displays a confirmation page, and sends a confirmation email.</td> </tr> </tbody> </table>	Actor Actions	System Actions	1. The user chooses a showtime for a movie.	2. The system displays the seating chart for that screening.	3. The user selects their desired seat(s).	4. The system displays a summary of the selection and total cost.	5. The user confirms to proceed to payment.	6. Upon receiving payment success confirmation, the system finalizes the booking.		7. The system generates an e-ticket, displays a confirmation page, and sends a confirmation email.	
Actor Actions	System Actions											
1. The user chooses a showtime for a movie.	2. The system displays the seating chart for that screening.											
3. The user selects their desired seat(s).	4. The system displays a summary of the selection and total cost.											
5. The user confirms to proceed to payment.	6. Upon receiving payment success confirmation, the system finalizes the booking.											
	7. The system generates an e-ticket, displays a confirmation page, and sends a confirmation email.											
Alternate Flows:	3a. If a selected seat becomes unavailable during the process, the system displays an alert and prompts the user to choose another seat.											

Table 11: UC9 – Process Payment

Use Case:	UC9. Process Payment											
Goal:	To handle the financial transaction for a ticket booking.											
Actors:	User											
Pre-condition:	The user has confirmed their booking details and is at the payment stage.											
Post-condition:	The payment is processed successfully or fails.											
Mainline Scenario:	<table border="1"><thead><tr><th>Actor Actions</th><th>System Actions</th></tr></thead><tbody><tr><td></td><td>1. The system displays a secure payment form.</td></tr><tr><td>2. The user enters their payment information and confirms the transaction.</td><td>3. The system securely transmits the payment data to a payment gateway.</td></tr><tr><td></td><td>4. The system receives a success response from the gateway.</td></tr><tr><td></td><td>5. The system records the transaction and confirms payment success to the "Book Ticket" use case.</td></tr></tbody></table>		Actor Actions	System Actions		1. The system displays a secure payment form.	2. The user enters their payment information and confirms the transaction.	3. The system securely transmits the payment data to a payment gateway.		4. The system receives a success response from the gateway.		5. The system records the transaction and confirms payment success to the "Book Ticket" use case.
Actor Actions	System Actions											
	1. The system displays a secure payment form.											
2. The user enters their payment information and confirms the transaction.	3. The system securely transmits the payment data to a payment gateway.											
	4. The system receives a success response from the gateway.											
	5. The system records the transaction and confirms payment success to the "Book Ticket" use case.											
Alternate Flows:	4a. If the gateway returns a failure response, the system displays an error message to the user.											

Table 12: UC10 – Print E-Ticket

Use Case:	UC10. Print E-Ticket					
Goal:	To obtain a printable version of a booked ticket.					
Actors:	User					
Pre-condition:	The user has a successfully booked ticket in their account.					
Post-condition:	The user is presented with a printable ticket.					
Mainline Scenario:	<table border="1"><thead><tr><th>Actor Actions</th><th>System Actions</th></tr></thead><tbody><tr><td>1. The user navigates to their booking history and selects the "Print Ticket" option for a booking.</td><td>2. The system retrieves the booking details.</td></tr></tbody></table>		Actor Actions	System Actions	1. The user navigates to their booking history and selects the "Print Ticket" option for a booking.	2. The system retrieves the booking details.
Actor Actions	System Actions					
1. The user navigates to their booking history and selects the "Print Ticket" option for a booking.	2. The system retrieves the booking details.					

		3. The system formats the details into a printable e-ticket layout and opens the browser's print dialog.
Alternate Flows:	None	

Table 13: UC11 – Manage Movies

Use Case:	UC11. Manage Movies									
Goal:	To add or update movies in the system catalog.									
Actors:	Admin									
Pre-condition:	The Admin is logged into the administrative panel.									
Post-condition:	The movie catalog is updated.									
Mainline Scenario:	<table border="1"> <thead> <tr> <th>Actor Actions</th> <th>System Actions</th> </tr> </thead> <tbody> <tr> <td>1. The Admin navigates to the "Manage Movies" section.</td> <td>2. The system displays a list of all movies.</td> </tr> <tr> <td>3. The Admin selects "Add New Movie" or "Edit" for an existing movie.</td> <td>4. The system displays a form for movie details (empty for new, pre-filled for edit).</td> </tr> <tr> <td>5. The Admin enters or modifies the movie's details and submits the form.</td> <td>6. The system validates the data, saves the changes to the database, and displays a success message.</td> </tr> </tbody> </table>		Actor Actions	System Actions	1. The Admin navigates to the "Manage Movies" section.	2. The system displays a list of all movies.	3. The Admin selects "Add New Movie" or "Edit" for an existing movie.	4. The system displays a form for movie details (empty for new, pre-filled for edit).	5. The Admin enters or modifies the movie's details and submits the form.	6. The system validates the data, saves the changes to the database, and displays a success message.
Actor Actions	System Actions									
1. The Admin navigates to the "Manage Movies" section.	2. The system displays a list of all movies.									
3. The Admin selects "Add New Movie" or "Edit" for an existing movie.	4. The system displays a form for movie details (empty for new, pre-filled for edit).									
5. The Admin enters or modifies the movie's details and submits the form.	6. The system validates the data, saves the changes to the database, and displays a success message.									
Alternate Flows:	5a. If the submitted data is invalid, the system displays an error message.									

Table 14: UC12 – Delete Movie

Use Case:	UC12. Delete Movie	
Goal:	To remove a movie from the system.	
Actors:	Admin	
Pre-condition:	The Admin is viewing the list of movies.	

Post-condition:	The selected movie is removed from the system.		
Mainline Scenario:	Actor Actions	System Actions	
	1. The Admin selects the "Delete" option for a movie.	2. The system displays a confirmation prompt.	
	3. The Admin confirms the deletion.	4. The system removes the movie record from the database.	
		5. The system refreshes the movie list and displays a success message.	
Alternate Flows:	3a. If the Admin cancels the deletion at the prompt, no action is taken.		

Table 15: UC13 – Manage Users

Use Case:	UC13. Manage Users		
Goal:	To view and manage user accounts.		
Actors:	Admin		
Pre-condition:	The Admin is logged into the administrative panel.		
Post-condition:	A management action is performed on a user account.		
Mainline Scenario:	Actor Actions	System Actions	
	1. The Admin navigates to the "Manage Users" section.	2. The system displays a list of all registered users.	
	3. The Admin selects a user to view their details or perform an action (e.g., "Deactivate").	4. The system executes the requested action (e.g., updates the user's status).	
		5. The system displays a confirmation of the action taken.	
Alternate Flows:	None		

Table 16: UC14 – Manage Screenings

Use Case:	UC14. Manage Screenings
-----------	-------------------------

Goal:	To create or update movie showtimes.		
Actors:	Admin		
Pre-condition:	The Admin is logged into the administrative panel..		
Post-condition:	The schedule of movie screenings is updated.		
Mainline Scenario:	Actor Actions	System Actions	
	1. The Admin navigates to the "Manage Screenings" section.	2. The system displays the current schedule.	
	3. The Admin selects to add a new screening.	4. The system displays a form to create a new screening.	
	5. The Admin selects a movie, screen, date, and time, then submits the form.	6. The system validates for scheduling conflicts, saves the new screening, and displays a success message.	
Alternate Flows:	5a. If a scheduling conflict is detected, the system displays an error message.		

Table 17: UC15 – Cancel Screening

Use Case:	UC15. Cancel Screening		
Goal:	To cancel an upcoming movie screening.		
Actors:	Admin		
Pre-condition:	The Admin is viewing the schedule of screenings.		
Post-condition:	The selected screening is cancelled.		
Mainline Scenario:	Actor Actions	System Actions	
	1. The Admin selects the "Cancel" option for a specific screening.	2. The system displays a confirmation prompt.	
	3. The Admin confirms the cancellation.	4. The system updates the screening's status to "Cancelled" and may trigger notifications/refunds for booked users.	
Alternate Flows:	3a. The Admin cancels the action at the prompt; no changes are made.		

Table 18: UC16 - View Report

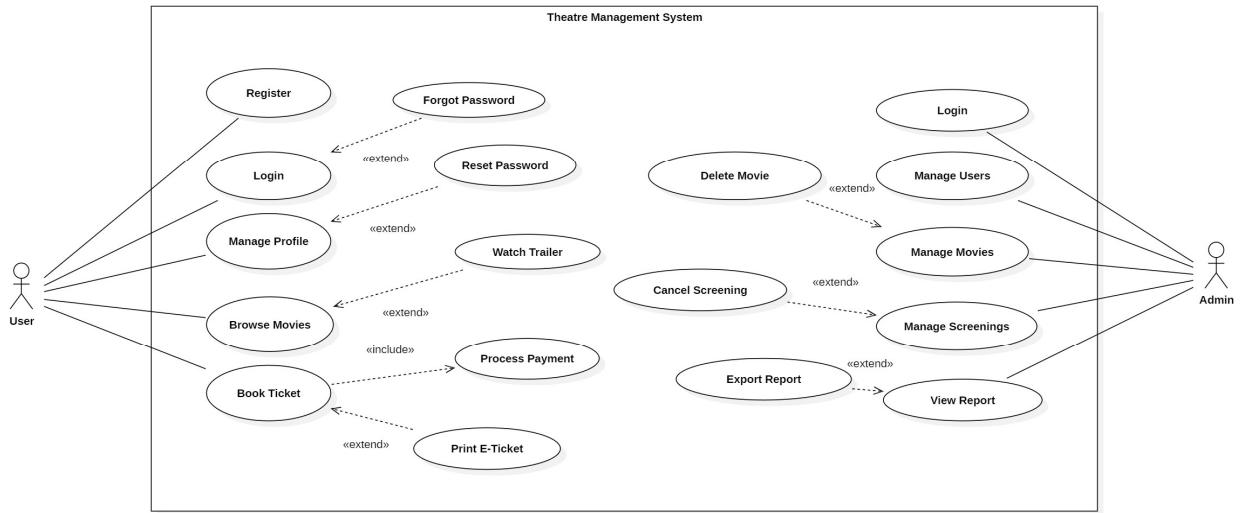
Use Case:	UC16. View Report										
Goal:	To view analytical reports on system activity.										
Actors:	Admin										
Pre-condition:	The Admin is logged into the administrative panel.										
Post-condition:	The Admin has viewed the requested report.										
Mainline Scenario:	<table border="1"><thead><tr><th>Actor Actions</th><th>System Actions</th></tr></thead><tbody><tr><td>1. The Admin navigates to the "Reports" section.</td><td></td></tr><tr><td>2. The Admin selects the type of report and applies any filters (e.g., date range).</td><td>3. The system processes the request, retrieves the relevant data, and aggregates it.</td></tr><tr><td></td><td>4. The system displays the generated report in a chart or table format.</td></tr></tbody></table>			Actor Actions	System Actions	1. The Admin navigates to the "Reports" section.		2. The Admin selects the type of report and applies any filters (e.g., date range).	3. The system processes the request, retrieves the relevant data, and aggregates it.		4. The system displays the generated report in a chart or table format.
Actor Actions	System Actions										
1. The Admin navigates to the "Reports" section.											
2. The Admin selects the type of report and applies any filters (e.g., date range).	3. The system processes the request, retrieves the relevant data, and aggregates it.										
	4. The system displays the generated report in a chart or table format.										
Alternate Flows:	None										

Table 19: UC17 – Export Report

Use Case:	UC17. Export Report										
Goal:	To download a report for offline analysis.										
Actors:	Admin										
Pre-condition:	The Admin is currently viewing a generated report (UC16).										
Post-condition:	A report file is downloaded to the Admin's computer.										
Mainline Scenario:	<table border="1"><thead><tr><th>Actor Actions</th><th>System Actions</th></tr></thead><tbody><tr><td>1. The Admin selects the "Export" option.</td><td></td></tr><tr><td>2. The Admin chooses a file format (e.g., PDF, CSV).</td><td>3. The system generates the report file in the selected format.</td></tr><tr><td></td><td>4. The system initiates a file download in the Admin's browser.</td></tr></tbody></table>			Actor Actions	System Actions	1. The Admin selects the "Export" option.		2. The Admin chooses a file format (e.g., PDF, CSV).	3. The system generates the report file in the selected format.		4. The system initiates a file download in the Admin's browser.
Actor Actions	System Actions										
1. The Admin selects the "Export" option.											
2. The Admin chooses a file format (e.g., PDF, CSV).	3. The system generates the report file in the selected format.										
	4. The system initiates a file download in the Admin's browser.										

Alternate Flows:	None
------------------	------

Use Case diagram:



Conclusion:

From the above experiment, I learned how to identify and define different use case scenarios for a movie ticket booking system and represent them systematically. Each use case clearly specifies its purpose, involved actors, preconditions, postconditions, main flow, and alternate flow, which together provide a detailed understanding of the system's functionality. By analyzing these scenarios, we could implement a use case diagram that visually depicts the interactions between users, the system, and external services, ensuring clarity in system requirements and supporting effective software design.