

USER CASES- ADMIN, CLIENT, MANAGER, STAFF

1.ADMIN CASES

Use Case Name	UC 001 - Managing users
Summary	This use case describes a set of actions that the admin of the system can perform in relation with user management.
Dependency	Admin UC - Add a new user Admin UC - Edit an existing user Admin UC - Remove an existing user
Actors	Primary actor: Admin Secondary actor: User being affected
Preconditions	1. The admin has been logged in successfully.
Description of main sequence	1. Admin navigates to the user management section. 2. Admin selects the user level he wants to make changes at(client, staff, manager). 3. Admin selects the action he wants to perform at this specific user. 4. Admin adds, edits or removes information regarding this user. 5. Admin saves changes.
Description of the alternative sequence	Invalid input detected 1. Admin navigates to the user management section. 2. Admin selects the user level he wants to make changes at(client, staff, manager). 3. Admin selects the action he wants to perform at this specific user. 4. Admin modifies or adds user details but the data is invalid. 5. Upon attempting to save changes, the system validates the input and detects the invalid data. 6. The system displays the appropriate message telling exactly where error is being displayed.

	7. Admin is prompted to correct the data before saving the changes again.
Non functional requirements	Performance: The system saves the changes immediately Security: The admin should be the only one to change these permissions Usability: The user interface should be easy to understand and not overwhelming
Postconditions	System reflects the changes made by the admin, including newly created users, edited users and removed users.

Use Case Name	UC005: Changing Specific User Permissions
Summary	This use case describes the ability of the admin to give unique permission to users who originally would not have access to such abilities or restrict them from abilities they would originally have
Dependency	Admin UC: Add Users Admin UC: Manage Users
Actors	Primary actor: Admin Secondary actor: Manager, Staff, Client
Preconditions	1. The admin has been logged in successfully
Description of main sequence	2. The admin navigates to the “user management” section in his panel 3. The admin selects the option to “edit user information” 4. The admin chooses the user they want to edit 5. The admin selects which permissions they want the user to have or not have 6. The admin saves changes 7. If the changes save correctly, the admin gets a message
Description of the alternative sequence	1. The admin navigates to the “user management” section in his panel

	<ol style="list-style-type: none"> The admin selects the option to “edit user information” The admin chooses the user they want to edit The admin selects which permissions they want the user to have or not have The admin saves changes If the changes fail to save, the system gives the admin a message, and gives him the option to retry or to cancel the changes If the changes save correctly, the admin gets a message
Non functional requirements	<p>Performance: The system saves the changes immediately</p> <p>Security: The admin should be the only one to change these permissions</p> <p>Usability: The user interface should be easy to understand and not overwhelming</p>
Postconditions	After choosing to go back, the admin should be sent back to the financial reports section

Use Case Name	UC006: View Financial Reports
Summary	This use case describes the ability of the admin to access a detailed report of the finances of each cinema during any specific period of their liking.
Dependency	<p>Staff UC: Ticket selling</p> <p>Client UC: Booking Tickets</p> <p>Admin UC: Add Users</p> <p>Admin UC: Manage Users</p> <p>Manager UC: Staff Spending</p>
Actors	Primary actor: Admin
Preconditions	<ol style="list-style-type: none"> The admin has been logged in successfully
Description of main sequence	<ol style="list-style-type: none"> The admin navigates to the “financial management” section in his panel The admin selects the option to view financial reports The admin chooses which cinema/s

	<p>they want a report on</p> <ol style="list-style-type: none"> The admin chooses the period for which they want a revenue report The system retrieves the financial data and displays it to the admin
Description of the alternative sequence	<ol style="list-style-type: none"> The admin navigates to the “financial management” section in his panel The admin selects the option to view financial reports The admin chooses which cinema/s they want a report on If the admin enters an invalid period the application will let him know that he should add a valid one The admin enters a valid period The system retrieves the financial data and displays it to the admin
Non functional requirements	<p>Performance: The system generates the financial report quickly</p> <p>Security: The reports should be only visible to the admin</p> <p>Usability: The information should be presented in a way that is both detailed and easy to understand</p>
Postconditions	After choosing to go back, the admin should be sent back to the financial reports section

Use Case Name	UC007: Financial Feedback
Summary	This use case enables the admin to give feedback to the managers regarding their cinema's performance and give them information for budgeting, price changes and plans for the future
Dependency	None
Actors	<p>Primary actor: Admin</p> <p>Secondary actor: Manager</p>
Preconditions	<ol style="list-style-type: none"> The admin has been logged in successfully The admin has up to date reports on the financial situation of the cinema/s The manager has the permission to

	get feedback from the admin
Description of main sequence	<ol style="list-style-type: none"> 1. The admin navigates to the “financial management” section in his panel 2. The admin selects the option to give feedback to the managers 3. The admin chooses which cinema/s they want to give feedback to 4. The admin chooses which manager to give feedback to 5. The admin writes their feedback 6. The admin sends the feedback
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The admin navigates to the “financial management” section in his panel 2. The admin selects the option to give feedback to the managers 3. The admin chooses which cinema/s they want to give feedback to 4. The admin chooses which manager to give feedback to 5. The admin writes their feedback 6. If the admin encounters a loss of connection or system crash while writing their message, it will get saved as a draft
Non functional requirements	<p>Performance: The system sends the feedback fast</p> <p>Security: The feedback is sent only to the managers the admin chooses and no one else can read them</p> <p>Reliability: Feedback data should be accurately captured, stored, and transmitted without loss or corruption.</p>
Postconditions	The app should let the admin know that the feedback message was sent, and then send him back to the “financial management” section

Use Case Name	UC008: Managing Cinema Facilities
Summary	This use case describes the process for the cinema owner to manage and maintain the facilities of their cinemas.
Dependency	None

Actors	Primary actor: Admin
Preconditions	<ol style="list-style-type: none"> 1. The admin has been logged in successfully
Description of main sequence	<ol style="list-style-type: none"> 1. The admin navigates to the "Facilities Management" section in their dashboard. 2. The admin views a list of all facilities within their cinemas (e.g., screens, seats, sound systems, projectors). 3. The admin selects a specific facility to manage. 4. The admin can perform actions such as scheduling maintenance, repairs, or upgrades for the selected facility. 5. The admin confirms the changes and updates the facility status.
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The admin navigates to the "Facilities Management" section in their dashboard. 2. The admin selects a specific cinema to manage. 3. The admin views a list of all facilities within their cinemas (e.g., screens, seats, sound systems, projectors). 4. The admin can perform actions such as scheduling maintenance, repairs, or upgrades for the selected facility. 5. If a facility requires immediate attention (e.g., equipment breakdown), the admin can expedite the maintenance process by marking it as urgent. 6. If there are budget constraints or resource limitations, the admin can prioritize maintenance tasks based on criticality. 7. The admin confirms the changes and updates the facility status.
Non functional requirements	<p>Performance: The system should provide real-time updates on facility status and maintenance schedules.</p> <p>Usability: The facilities management interface should be intuitive and easy to navigate.</p> <p>Reliability: The system should accurately</p>

	track maintenance activities and ensure timely resolutions.
Postconditions	After choosing to go back, the admin should be sent back to the financial reports section

2.CLIENT CASES

UC Name	ClientUC1 : View Movies
Summary	The client should be able to browse all currently showing movies, or movies that are confirmed to be available in the future using the app.
Dependency	None
Actors	The primary actor in this case is the client.
Preconditions	The client must have the app downloaded. The app must have a list of movies that it's displaying
Description of the Main Sequence	<ul style="list-style-type: none"> • <i>Step 1: The client opens the app</i> • <i>Step 2: They click on "Currently showing"</i> • <i>Step 3: They browse all the current movies that are showing, finding whichever one interests them</i>
Description of the Alternative Sequence	<ul style="list-style-type: none"> • <i>Step 1: The client opens the app</i> • <i>Step 2: They click on "Showing soon"</i> • <i>Step 3: They browse all the movies that are confirmed to be showing in cinemas in the future, finding whichever one interests them.</i>

Non functional requirements	<p><i>The user interface must respond quickly to the user input, within a second.</i></p> <p><i>The display when browsing through the movies must be optimised for both, and change depending on if the user is viewing on landscape or portrait mode</i></p> <p><i>The system should be able to handle letting the user scroll through multiple movies at a time</i></p> <p><i>Clicking on a movie should take the user to it's page quickly.</i></p>
Postconditions	<p><i>The client is able to see all the movies that are currently screening or that are confirmed to be screening in the future.</i></p>

UC Name	<i>ClientUC2 : View movie information</i>
Summary	<p><i>The client should be able to view the information available for any given movie by clicking on it's page in the app. This information includes it's title, poster, main actors involved, director, description, trailer, etc.</i></p>
Dependency	<i>ClientUC1 : View Movies</i>
Actors	<i>The primary actor in this case is the client.</i>
Preconditions	<p><i>The client must have the app downloaded.</i></p> <p><i>The app must have a list of movies that it's displaying along with available information on each movie</i></p> <p><i>The client must select the movie they want to see information on</i></p>

Description of the Main Sequence	<ul style="list-style-type: none"> • Step 1: The client opens the app • Step 2: They find the movie they're interested in • Step 3: They're able to see all available information on the movie
Description of the Alternative Sequence	<ul style="list-style-type: none"> • Step 1: The client opens the app • Step 2: They find an upcoming movie they're interested in • Step 3: They see a screen telling them that more information on the movie will be available soon
Non functional requirements	<p>The user interface must respond quickly to the user input, within a second.</p> <p>The display when showing the information must be optimised for both, and change depending on if the user is viewing on landscape or portrait mode</p> <p>The information must be laid out in an easy to understand and intuitive way</p>
Postconditions	The client is able to view all currently available information on the movie they're interested in.

UC Name	<i>ClientUC3 : Book tickets in app</i>
Summary	<i>The client should be able to book tickets using the app, selecting their preferred seating, screening and cinema. This information then is sent to the staff through the app.</i>
Dependency	<i>ClientUC1 : View Movies</i>
Actors	<i>The primary actor in this case is the client who sends the information through the app. The secondary actor is the staff member who receives the information.</i>

Preconditions	<p><i>The client must have the app downloaded.</i></p> <p><i>The app must have a listing of all current movies, their times of showing, the seating, and the prices.</i></p> <p><i>The client must select the movie they want to watch.</i></p>
Description of the Main Sequence	<ul style="list-style-type: none"> • <i>Step 1: The client opens the app</i> • <i>Step 2: They find the movie they're interested in</i> • <i>Step 3: They're able to see all the available dates, times, cinemas and seats for the movie they want, and then book a ticket right there.</i>
Description of the Alternative Sequence	<ul style="list-style-type: none"> • <i>Step 1: The client opens the app</i> • <i>Step 2: They find the movie they're interested in</i> • <i>Step 3: They try to book a ticket, but are met with a screen telling them that they're seat was already taken, while showing them all other available seats they can choose.</i>
Non functional requirements	<p><i>The user interface must respond quickly to the user input.</i></p> <p><i>The information on the app must always be up to date.</i></p> <p><i>Everything should be as easy and intuitive as possible.</i></p> <p><i>The seating information must be updated as quickly as possible when there's a change, so 2 clients don't book the same seat</i></p>
Postconditions	<p><i>The client is able to select the movie going experience they want, and book a ticket.</i></p>

UC Name	<i>ClientUC4 : Pay in app</i>
----------------	-------------------------------

Summary	<i>The client should be able to pay for everything using the app, and then this information is processed in the app, and if the purchase is successful, the information about what was purchased should be sent to the staff, and the payment info should be sent to finance management.</i>
Dependency	<i>ClientUC1 : View movies ClientUC3 : Book tickets</i>
Actors	<i>The primary actor is the client, who makes the purchase, and the secondary actors are the staff who receive the information about the purchase, and finance management who gets the information on the payment.</i>
Preconditions	<i>The client must have the app downloaded. The app must have a listing of all current movies, their times of showing, the seating, and the prices. The client must select the movie they want to watch.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> • Step 1: The client opens the app • Step 2: They find the movie they're interested in • Step 3: They select the date, time, cinema, seat for the movie • Step 4 : They press "Make purchase" • Step 5 : They enter their credit card information, completing the transaction
Description of the Alternative Sequence	<ul style="list-style-type: none"> • Step 1: The client opens the app • Step 2: They find the movie they're interested in • Step 3: They're select the date, time and cinema for the movie • Step 4 : They press "Pay in theatre"

Non functional requirements	<p><i>The user interface must respond quickly to the user input.</i></p> <p><i>The app must accept purchases in all popular modes of online payment (paypal, mastercard, visa, etc)</i></p> <p><i>The payment process must be easy and intuitive.</i></p> <p><i>Everything must be secure, insuring that the payment process is as safe for the client as possible.</i></p>
Postconditions	<p><i>The client is able to book a ticket through the app, either paying online, or in person (in which case they cant reserve a seat)</i></p>

UC Name	ClientUC5 : Preorder snacks in app
Summary	The client should be able to preorder the snacks, beverages or both that they want, and then pick them up from a staff member at the cinema.
Dependency	ClientUC1 : View movies ClientUC3 : Book tickets ClientUC4 : Pay in app
Actors	The primary actor is the client, who makes the order, and the secondary actors are the staff who receive the information about the purchase.
Preconditions	<p>The client must have the app downloaded.</p> <p>The app must have a listing of all current movies, their times of showing, the seating, and the prices.</p> <p>The client must have selected and booked a ticket for a movie.</p>

Description of the Main Sequence	<ul style="list-style-type: none"> ● Step 1: The client opens the app ● Step 2: They find the movie they're interested in ● Step 3: They book a ticket for it. ● Step 4 : They press "Preorder snacks/beverages" ● Step 5 : They select what they want to order ● Step 6 : They enter in their payment information, completing the purchase
Description of the Alternative Sequence	<ul style="list-style-type: none"> ● Step 1: The client opens the app ● Step 2: They find the movie they're interested in ● Step 3: They book a ticket for it. ● Step 4 : They press "Preorder snacks/beverages" ● Step 5 : They select what they want to order. ● Step 4 : They press "Pay in theatre"
Non functional requirements	<p>The user interface must respond quickly to the user input.</p> <p>The app must accept purchases in all popular modes of online payment (paypal, mastercard, visa, etc)</p> <p>The app must have all current snacks available, as well as the promotional deals.</p> <p>The payment process must be easy and intuitive.</p> <p>Everything must be secure, insuring that the payment process is as safe for the client as possible.</p>
Postconditions	<p>The client is able to preorder snacks through the app, which then they can easily go and pick up at the theatre before the movie starts, making their experience faster.</p>

Use Case Name	ClientUC6:Recieve notifications
Summary	This use case describes the ability of a client to view upcoming movies with details and receive notifications or updates.
Dependency	None
Actors	Primary actor: Clients
Preconditions	1.The client has been logged in successfully.
Description of main sequence	1.Client navigates to the “Upcoming Movies” section. 2.Client views a list of upcoming movies with their details, such as release dates, trailers, summaries and cast information. 3.Client selects a movie to view more details. 4.Client sets reminders for release dates or marks movies as favorites. 5.Client receives notifications or updates for the selected movies.
Description of the alternative sequence	1. If the client experiences a connectivity issue, the app displays a message indicating the inability to load upcoming movies. 2.The client may choose to retry the list or wait for the connection to be restored.
Non functional requirements	-Reliability Reminders should be reliable, ensuring clients receive notifications at the right time to stay informed about their favourite movie. -Performance The app should load the list of upcoming movies quickly, so clients do not have to wait too long to see what is new.
Postconditions	After using the app, clients have seen the list of movies and set reminders for their favourites.

Use Case Name	ClientUC7:Enroll in loyalty programs
Summary	This use case describes the ability of a client to enroll in cinema loyalty programs through

	the app, providing necessary information such as name, email, and phone number.
Dependency	Client case: Claim benefits and apply promotions
Actors	Primary actor: Clients
Preconditions	1.The client has been logged in successfully.
Description of main sequence	1.Client navigates to the “Loyalty programs” section. 2.Client selects the option to enroll in a loyalty program. 3.Client provides necessary information such as name, email and phone number. 4.Client submits the enrollment form. 5.The system processes the enrollment request and adds the client to the loyalty program. 6.Client receives confirmation of enrollment.
Description of the alternative sequence	If the client encounters an error: 1.The system displays an error message asking the client to put the information again. 2.Client modifies the necessary fields and resubmits the form. 3.The system reprocesses the enrollment request.
Non functional requirements	-Security The information of the client should be securely stored and managed. -Performance The process of enrolling should be easy to understand and navigate quickly for clients.
Postconditions	After using the app, clients are successfully enrolled in the loyalty program.

Use Case Name	ClientUC8:Claim benefits and apply promotions
Summary	<p>Once clients are enrolled in loyalty programs, they can claim benefits directly from the app. This could include redeeming rewards points for free tickets or concessions, accessing exclusive discounts or promotions, or receiving personalized offers based on their preferences and past activity.</p> <p>Additionally, even without enrollment, clients can still enjoy various marketing promotions by viewing the list of prices in the app.</p>
Dependency	<p>Client Case: Enroll in loyalty programs</p> <p>Client Case: Contact customer support</p>
Actors	<p>Primary actor: Clients</p> <p>Secondary actor: Staff</p>
Preconditions	1.The client has been logged in successfully.
Description of main sequence	<p>1.Client navigates to the “Loyalty Program” or “Promotions” section.</p> <p>2. If the client is enrolled in the loyalty program:</p> <ul style="list-style-type: none"> -Client views the desired benefit, such as redeeming reward points for free tickets, accessing exclusive discounts, or receiving personalized offers. -Client selects the desired benefit. -Client follows the instructions to claim the benefits. -The system processes the request and applies the benefit to the client’s account. <p>3.If not enrolled:</p> <ul style="list-style-type: none"> -Client views the list of prices or promotions available in the app. -Client selects the desired promotion. -Client follows the instructions to apply the promotion. -The system processes the request and applies the promotion to the client’s account.
Description of the alternative sequence	<p>If the app has technical issues:</p> <ul style="list-style-type: none"> -Client receives a notification informing them for the issue. -Client is asked to try again later or contact customer support.
Non functional requirements	<p>Security</p> <ul style="list-style-type: none"> -The personal information of the client, such as loyalty program details and transaction history should be securely stored and managed. <p>Availability of instructions</p> <ul style="list-style-type: none"> -The app should provide clear instructions for clients to claim benefits and apply for promotions.

Postconditions	After using the app, client has successfully claimed benefits or applied promotions.

Use Case Name	Client UC9:Contact customer support
Summary	Clients should have access to a customer support feature within the app, allowing them to easily reach out for assistance with any issues. They can chat with someone from staff in real-time, or find answers to common questions. They can also see what's happening with their request and get updates through the app.
Dependency	Client case: Claim benefits and apply promotions
Actors	Primary actor: Clients Secondary actor: Staff
Preconditions	1.The client has been logged in successfully.
Description of main sequence	1.Client navigates to the “Help” or “Support” section. 2. If the client selects the “Support” section : -Client initiates a chat with a staff member. -Client describes the issue or the question and sends the message. -Staff member responds in real time. 3.If the client selects the “Help” section : -Client browses through the FAQ section. -Client searches for relevant topics or questions. -Client reads the provided information.
Description of the alternative sequence	1.If the client cannot find answers to their question in the FAQ or knowledge base: -The app asks the client to submit a support request. -Client fills out a form detailing their issue or question. -The app notifies the client that their request has been submitted and will be addressed by staff service.

	<p>2.If the client does not receive updates on their support request:</p> <ul style="list-style-type: none"> -The app provides a "Check Status" option within the support section. -Client selects this option to view the status of their request. If there are no updates, the app informs the client that their request is still being processed.
Non functional requirements	<p>-Security The conversation between the client and the staff should be secure and can't be seen by anyone else.</p> <p>-Performance The messages get sent in real time, without any delay.</p> <p>-Usability The user interface should be easy to understand and not overwhelming.</p>
Postconditions	<p>After finishing the conversation with the staff member, the chat gets saved in the system database, and the user goes back to the previous page.</p>

3.MANAGER CASES

Use Case Name	Cinema revenue
Summary	This use case describes the involvement of Managers in tracking the revenue of the cinema using the cinema software
Dependency	None
Actors	Primary actor- Manager
Preconditions	1. The manager has logged on successfully
Description of main sequence	<ol style="list-style-type: none">1. The manager opens the performance tab from the main view.2. The manager chooses the “Overall Revenue” button3. The manager enters a period or clicks the “All Time” button4. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Description of the alternative sequence	<ol style="list-style-type: none">1. The manager opens the performance tab from the main view.2. The manager chooses the “Overall Revenue” button3. If the manager enters a period that is not valid the system alerts him to enter a valid period4. The manager enters a period or clicks the “All Time” button5. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Non functional requirements	Security: The overall revenue should be viewed only by the manager Usability: The overall revenue should be displayed in an easy-to-read format
Postconditions	After choosing to go back, the Manager should be sent back to the Performance section

Use Case Name	Tracking Staff Member Income
Summary	This use case describes the involvement of Managers in tracking the income made by staff using the cinema software
Dependency	Staff UC: Ticket selling
Actors	Primary actor- Manager Secondary actor- Staff
Preconditions	<ol style="list-style-type: none"> 1. The manager has logged on successfully 2. The Staff account already exists
Description of main sequence	<ol style="list-style-type: none"> 1. The manager opens the performance tab from the main view. 2. The manager picks from a series of already existing staff. 3. The manager chooses the "Income" button. 4. The manager enters a period or clicks the "Overall Income" button 5. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The manager opens the performance tab from the main view. 2. The manager picks from a series of already existing staff. 3. The manager chooses the "Income" button. 4. If the Manager enters a period that is not valid the system alerts him to try a valid period. 5. The manager enters a period or clicks the "Overall Income" button 6. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Non functional requirements	<p>Security: The staff income should be viewed only by managers.</p> <p>Performance: The program should display the information quickly, to make it possible to check a vast number of staff faster.</p>
Postconditions	After choosing to go back, the Manager should be sent back to the Performance section

Use Case Name	Tracking Staff Member Spending
Summary	This use case describes the involvement of Managers in tracking the spending made by staff using the cinema software
Dependency	None
Actors	Primary actor- Manager Secondary actor- Staff
Preconditions	<ol style="list-style-type: none"> 1. The manager has logged on successfully 2. The Staff account already exists
Description of main sequence	<ol style="list-style-type: none"> 1. The manager opens the performance tab from the main view. 2. The manager picks from a series of already existing staff. 3. The manager chooses the "Spending" button. 4. The manager enters a period or clicks the "Overall Spending" button 5. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The manager opens the performance tab from the main view. 2. The manager picks from a series of already existing staff. 3. The manager chooses the "Spending" button. 4. If the Manager enters a period that is not valid the system alerts him to try a valid period. 5. The manager enters a period or clicks the "Overall Spending" button 6. If the manager has entered a period the system displays the data during that period, otherwise it shows the all-time data
Non functional requirements	<p>Security: The staff spending should be viewed only by managers.</p> <p>Performance: The program should display the information quickly, to make it possible to</p>

	check a vast number of staff faster.
Postconditions	After choosing to go back, the Manager should be sent back to the Performance section

Use Case Name	Add a new film
Summary	This use case describes the involvement of Managers in adding a new film to the catalogue of the cinema using the cinema software
Dependency	None
Actors	Primary actor- Manager
Preconditions	1. The manager has logged on successfully
Description of main sequence	<ol style="list-style-type: none"> 1. The manager opens the manage catalogue tab from the main view 2. The manager chooses the “Add new Film” button 3. The manager enters the information regarding the film 4. The manager clicks the “Save” button 5. The system validates the data entered and adds the film if they are correct
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The manager opens the manage catalogue tab from the main view 2. The manager chooses the “Add new Film” button 3. The system displayed the type of information needed 4. The manager enters the information regarding the film 5. The manager clicks the “Save” button 6. The system validates the data entered shows which ones are entered incorrectly and asks the user to re-enter if they are incorrect

Non functional requirements	Security: Only the manager can add new films to the catalogue Performance: The program should process the values entered quickly, to make the work of adding films more efficient
Postconditions	After choosing to go back, the Manager should be sent back to the Manage Catalogue section

Use Case Name	Delete an existing film
Summary	This use case describes the involvement of Managers in deleting an existing film from the catalogue of the cinema using the cinema software

Dependency	None
Actors	Primary actor- Manager
Preconditions	<ol style="list-style-type: none"> 1. The manager has logged on successfully 2. The film to be deleted already exists
Description of main sequence	<ol style="list-style-type: none"> 1. The manager opens the manage catalogue tab from the main view 2. The manager chooses the “Delete Film” button 3. The manager picks from a series of already existing films 4. The system displays all of the information regarding the film 5. The manager clicks the “Delete” button 6. The system displays a message that informs the manager that the film is deleted
Description of the alternative sequence	None
Non functional requirements	<p>Security: Only the manager can delete existing films from the catalogue</p> <p>Performance: The program should handle deleting films rapidly to make the work of deleting films more efficient</p>
Postconditions	After choosing to go back, the Manager should be sent back to the Manage Catalogue section

Use Case Name	Modifying an existing film
Summary	This use case describes the involvement of Managers in modifying the data of an existing film from the catalogue of the cinema using the cinema software
Dependency	None
Actors	Primary actor- Manager

Preconditions	<ol style="list-style-type: none"> 1. The manager has logged on successfully 2. The film to be modified already exists
Description of main sequence	<ol style="list-style-type: none"> 1. The manager opens the manage catalogue tab from the main view 2. The manager chooses the “Modify Film” button 3. The manager picks from a series of already existing films 4. The system displays all of the information regarding the film 5. The manager modifies the data that he/she wants 6. The manager clicks the “Save” button 7. The system validates the data entered and modifies the film if they are correct
Description of the alternative sequence	<ol style="list-style-type: none"> 1. The manager opens the manage catalogue tab from the main view 2. The manager chooses the “Modify Film” button 3. The manager picks from a series of already existing films 4. The system displays all of the information regarding the film 5. The manager modifies the data that he/she wants 6. The manager clicks the “Save” button 7. The system validates the data entered shows which ones are entered incorrectly and asks the user to re-enter if they are incorrect
Non functional requirements	<p>Security: Only the manager can modify the data of existing films from the catalogue</p> <p>Performance: The program should process the values entered quickly, to make the work of modifying films more efficient</p>
Postconditions	After choosing to go back, the Manager should be sent back to the Manage Catalogue section

UC Name	User Login
----------------	------------

Summary	This use case allows users to authenticate themselves and gain access to the system. It verifies the identity of users attempting to log in and ensures that only authorized individuals can access the system.
Dependency	None
Actors	Primary Actor: Manager Secondary actor: Staff
Preconditions	The user must have a valid username and password.
Description of the Main Sequence	1-The user enters their username and password. 2-The system validates the entered credentials against stored records. 3-If the credentials are valid, the system grants access to the user, allowing them to proceed to use the system.
Description of the Alternative Sequence	If the entered credentials are invalid, the system prompts the user to re-enter their username and password, or it may provide an error message indicating the incorrect credentials.
Non functional requirements	1-Secure encryption of passwords to ensure confidentiality. 2-Authentication mechanism should be efficient to provide a seamless user experience. 3-The system should be resilient against common security threats like brute force attacks or credential stuffing.
Postconditions	The user gains access to the system upon successful authentication, allowing them to perform their intended actions based on their role and permissions within the system.

UC Name	Access System via Mobile App
Summary	This use case allows users to access the system using a dedicated mobile application, providing flexibility and convenience for users who need to manage staff schedules or track attendance while on the go.
Dependency	User Login
	Primary Actor:Manager

Actors	Secondary actors: Staff
Preconditions	The mobile application is installed on the user's device, and the user must be logged into the system.
Description of the Main Sequence	1-The user launches the mobile application on their device. 2-The user enters their username and password into the mobile app. 3-The system authenticates the user's credentials and grants access to the mobile interface. 4-The user can then view staff schedules, record attendance, or perform other relevant tasks using the mobile app.
Description of the Alternative Sequence	If the mobile network connection is poor or unavailable, the user may experience delays in accessing the system or encounter issues with data synchronization.
Non functional requirements	1-The mobile interface should be responsive and user-friendly, optimized for various screen sizes and devices. 2-Offline access capabilities should be provided to allow users to perform essential tasks even when not connected to the internet. 3-The mobile app should adhere to security best practices to ensure the confidentiality and integrity of data transmitted between the device and the system.
Postconditions	The user gains access to the system upon successful authentication, allowing them to perform their intended actions based on their role and permissions within the system.

UC Name	Create/Edit Schedule
Summary	This use case empowers managers to create or modify schedules for staff members. It facilitates the efficient allocation of shifts to ensure adequate coverage while taking into account factors such as employee availability and business needs.
Dependency	User Login
Actors	Manager
Preconditions	The manager must be logged into the system.

Description of the Main Sequence	<p>1-The manager selects the staff member or group for whom the schedule needs to be created or modified.</p> <p>2-The manager specifies the date and time slots for shifts, assigning staff members accordingly.</p> <p>3-The system updates the schedule to reflect the changes made by the manager.</p>
Description of the Alternative Sequence	<p>If conflicts arise, such as overlapping shifts or unavailable staff members, the manager may be prompted to resolve these conflicts before finalizing the schedule.</p>
Non functional requirements	<p>1-The scheduling interface should be intuitive and user-friendly to streamline the process for managers.</p> <p>2-The system should support recurring schedules as well as one-time events for flexibility in scheduling.</p> <p>3-It should provide real-time updates and notifications to keep staff members informed of any changes to their schedules.</p>
Postconditions	<p>The schedule for the selected staff member or group is successfully created or updated in the system, ensuring that shifts are allocated effectively to meet operational requirements.</p>

4.STAFF CASES

UC Name	Selling Tickets
Summary	This use case outlines the process for staff to sell tickets to customers.
Dependency	None
Actors	Primary actor: Staff Secondary actor: Customer
Preconditions	Staff member is logged into the system. Ticket selling functionality is accessible.
Description of the Main Sequence	1.Staff member navigates to the ticket selling section of the system. 2. Staff member selects the movie, showtime, and seat(s) based on customer preference. 3.Staff member confirms the ticket details and calculates the total price. 4.Staff member collects payment from the customer. 5.Staff member issues the ticket to the customer. 6. Staff member updates the system to reflect the sold ticket.
Description of the Alternative Sequence	If the selected seats are not available, the staff member informs the customer and helps them choose alternative seats.

Non functional requirements	<p>1. Performance: The system should handle ticket sales to minimize customer waiting time.</p> <p>2. Usability: The ticket selling interface should be intuitive for staff members to navigate.</p> <p>3. Security: Staff members should only have access to sell tickets and not perform unauthorized actions.</p>
Postconditions	The customer receives the ticket, and the system updates the available seat inventory accordingly.

UC Name	Handling Customer Inquiries
Summary	This use case describes how staff handle customer inquiries and assistance requests.
Dependency	None
Actors	<p>Primary actor: Staff</p> <p>Secondary actor: Customer</p>
Preconditions	<p>Staff member is logged into the system.</p> <p>Customer inquiries functionality is accessible.</p>
Description of the Main Sequence	1.Staff member receives a customer inquiry through in-person interaction, phone call, or online chat.

	<p>2. Staff member listens to the customer query and gathers necessary information.</p> <p>3. Staff member provides relevant information or assistance to resolve the customer's issue.</p> <p>4. If necessary, staff member escalates the inquiry to a manager or higher authority.</p>
Description of the Alternative Sequence	If the staff member is unable to resolve the inquiry immediately, they inform the customer of the steps being taken to address the issue and provide an estimated resolution time.
Non functional requirements	<p>1. Responsiveness: Staff members should respond promptly to customer inquiries.</p> <p>2. Communication: Staff members should effectively communicate with customers to understand and address their concerns.</p> <p>3. Professionalism: Staff members should maintain a professional demeanor while assisting customers</p>
Postconditions	The customer's inquiry is resolved satisfactorily, or appropriate steps are taken to address the issue.

UC Name	Checking Ticket Validity
Summary	This use case outlines the process for staff to check the validity of tickets presented by customers..
Dependency	None
Actors	Primary actor: Staff

	Secondary actor: Customer
Preconditions	<p>Staff member is logged into the system.</p> <p>The customer provides their ticket.</p>
Description of the Main Sequence	<ol style="list-style-type: none"> 1. Staff member navigates to manage ticket section. 2. Staff member scans the ticket or manually searches the ticket by ticketId. 3. Staff member checks the validity of the ticket. (If it exists in the database or not). 4. If the ticket is valid, the staff member changes the status of the ticket. 5. If the ticket is invalid, staff member may create a new ticket for the client.
Description of the Alternative Sequence	<p>Invalid ticket format</p> <ol style="list-style-type: none"> 1. Staff member navigates to manage ticket section. 2. Staff member scans the ticket or manually searches the ticket by ticketId but the format is incorrect. 3. Staff member will stay on the same page and the system will provide an error message that the format of the ticket Id is not correct

Non functional requirements	<p>1. Reliability: The ticket scanning system should accurately detect valid and invalid tickets.</p> <p>2. Efficiency: The ticket validation process should be swift to minimize customer waiting time.</p> <p>3. Customer Service: Staff members should handle invalid ticket situations courteously and efficiently.</p>
Postconditions	Customers with valid tickets are allowed entry, while appropriate action is taken for customers with invalid tickets.

UC Name	Reporting Technical Issues
Summary	This use case describes how staff report technical issues encountered with cinema equipment or software.
Dependency	None
Actors	<p>Primary actor: Staff Member</p> <p>Secondary actor: Manager</p>
Preconditions	<p>Staff member is logged into the system.</p> <p>Technical issue reporting functionality is accessible..</p>
Description of the Main Sequence	<p>1.Staff member navigates to the Technical Issue section within the system.</p> <p>2.Staff member creates an issue.</p>

	<p>3.Staff member enters a detailed description of the problem.</p> <p>4.System will notify the staff member, type of technician via email.</p> <p>5.If a staff member, type of technician, resolves the technical issue, the system will notify the staff member via email.</p> <p>6. If a staff member, type of technician, resolves the technical issue, it must report to the manager.</p>
Description of the Alternative Sequence	<p>1.Staff member navigates to the Technical Issue section within the system.</p> <p>2.Staff member creates an issue.</p> <p>3.Staff member enters a detailed description of the problem.</p> <p>4.System will notify the staff member, type of technician via email.</p> <p>5.Technician has not solved the issue for 7 days, then the status of the issue changes to important.</p>
Non functional requirements	<p>1. Clarity: Staff members should provide clear and detailed descriptions of technical issues to facilitate resolution.</p> <p>2.Timeliness: Staff members should report technical issues promptly to minimize disruption to cinema operations.</p> <p>3. Collaboration: Staff members should collaborate with technical support teams to resolve reported issues effectively.</p>
Postconditions	<p>Technical support teams receive the reported issue and initiate appropriate troubleshooting and resolution procedures.</p>

UC Name	Making a Transaction
Summary	This use case outlines the process for staff to make a new transaction composed of tickets and snacks.
Dependency	None
Actors	Primary actor: Staff
Preconditions	Staff member is logged into the system.
Description of the Main Sequence	<ol style="list-style-type: none"> 1. The staff member navigates to the Transaction section in the system. 2. The staff member creates a new Transaction. 3.The staff member selects the type and number of tickets the customer wants to purchase. 4. The system returns available seating and pricing options. 5. New ticket(s) are created. 6. Optionally, the staff member adds the customer's selected snacks to the transaction. 7. The system calculates the total cost of the transaction.. 8. The system displays the bill for the transaction.
Description of the Alternative Sequence	<ol style="list-style-type: none"> 1. The staff member navigates to the Transaction section in the system. 2. The staff member creates a new Transaction. 3.The staff member selects the type and number of tickets the customer wants to purchase.

	<p>4. The system returns available seating and pricing options.</p> <p>5. There are no seats available.</p> <p>6. System won't allow to make a new ticket for this movie in this hour.</p>
Non functional requirements	<p>1. Performance: The system should handle ticket sales to minimize customer waiting time.</p> <p>2. Usability: The ticket selling interface should be intuitive for staff members to navigate.</p> <p>3. Security: Staff members should only have access to sell tickets and not perform unauthorized actions.</p>
Postconditions	<p>The customer receives the ticket(s), snack(s) and the system updates the available seat inventory accordingly and generates a new bill in the end.</p>