

## USER CASES- ADMIN, CLIENT, MANAGER, STAFF

### 1.ADMIN CASES

<b>Use Case Name</b>	<b>UC 001 - Managing users</b>
<b>Summary</b>	This use case describes a set of actions that the admin of the system can perform in relation with user management.
<b>Dependency</b>	<b>Admin UC</b> - Add a new user <b>Admin UC</b> - Edit an existing user <b>Admin UC</b> - Remove an existing user
<b>Actors</b>	<b>Primary actor:</b> Admin <b>Secondary actor:</b> User being affected
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. The admin has been logged in successfully.</li></ol>
<b>Description of main sequence</b>	<ol style="list-style-type: none"><li>1. Admin navigates to the user management section.</li><li>2. Admin selects the user level he wants to make changes at(client, staff, manager).</li><li>3. Admin selects the action he wants to perform at this specific user.</li><li>4. Admin adds, edits or removes information regarding this user.</li><li>5. Admin saves changes.</li></ol>
<b>Description of the alternative sequence</b>	<b>Invalid input detected</b> <ol style="list-style-type: none"><li>1. Admin navigates to the user management section.</li><li>2. Admin selects the user level he wants to make changes at(client, staff, manager).</li><li>3. Admin selects the action he wants to perform at this specific user.</li><li>4. Admin modifies or adds user details but the data is invalid.</li><li>5. Upon attempting to save changes, the system validates the input and detects the invalid data.</li><li>6. The system displays the appropriate message telling exactly where error is being displayed.</li></ol>

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

<b>Use Case Name</b>	<b>UC002</b> - Editing client information
<b>Summary</b>	This use case describes the involvement of admin editing the information of the clients using the cinema software.
<b>Dependency</b>	<b>Admin UC</b> - Add a new user <b>Admin UC001</b> - Manage users
<b>Actors</b>	<b>Primary actor:</b> Admin <b>Secondary actor:</b> Client
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The admin has been logged in successfully.</li> <li>2. The client account already exists.</li> <li>3. UC002 has been executed successfully.</li> </ol>
<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>1. Admin navigates to the client management section within the user management interface.</li> <li>2. Admin selects the client whose information is needed to be edited.</li> <li>3. Admin modifies the information of the client.</li> <li>4. Admin saves the changes.</li> </ol>
<b>Description of the alternative sequence</b>	<b>Client not found</b> <ol style="list-style-type: none"> <li>1. Admin navigates to the client management section within the user management interface.</li> <li>2. Admin searches the client whose information is needed to be edited by</li> </ol>

	<p>their name.</p> <ol style="list-style-type: none"> <li>The system is unable to locate this client due to incorrect search terms or this client account may be deleted.</li> <li>The system displays the appropriate message indicating that this client cannot be found.</li> <li>Admin can choose to search again using a different criteria or adding a new client account if necessary.</li> </ol>
<b>Non functional requirements</b>	<p><b>Performance:</b> The system saves the changes immediately</p> <p><b>Security:</b> The admin should be the only one to change these permissions</p> <p><b>Usability:</b> The user interface should be easy to understand and not overwhelming</p>
<b>Postconditions</b>	The client information is updated within the system according to changes made by the admin.

<b>Use Case Name</b>	<b>UC003</b> - Deleting a staff member
<b>Summary</b>	This use case details the process for an admin to delete a staff member's account after he is no longer working at this cinema.
<b>Dependency</b>	<b>UC001</b> - Managing users
<b>Actors</b>	<p><b>Primary actor:</b> Admin</p> <p><b>Secondary actor:</b> Staff member</p>
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>The admin has been logged in successfully.</li> <li>The staff member's account already exists.</li> <li>UC003 has been executed successfully</li> </ol>
<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>Admin navigates to the staff member management section within the user management interface.</li> <li>Admin searches and selects the staff member who will be deleted by their name.</li> <li>Admin clicks the delete option.</li> </ol>

	<ol style="list-style-type: none"> <li>Admin confirms the deletion of the staff member.</li> <li>The system deletes the staff member's account and displays a confirmation message to the admin.</li> </ol>
<b>Description of the alternative sequence</b>	<p><b>Staff member not found</b></p> <ol style="list-style-type: none"> <li>Admin navigates to the staff member management section within the user management interface.</li> <li>Admin searches the staff member who will be deleted by their name.</li> <li>The system is unable to locate this staff member's account due to incorrect search terms or this staff member's account may be deleted.</li> <li>The system displays the appropriate message indicating that this staff member cannot be found.</li> <li>Admin can choose to search again using a different criteria.</li> </ol> <p><b>Cancellation of deletion</b></p> <ol style="list-style-type: none"> <li>Admin navigates to the staff member management section within the user management interface.</li> <li>Admin searches and selects the staff member who will be deleted by their name.</li> <li>Admin clicks the delete option.</li> <li>Admin does not confirm the deletion of the staff member.</li> <li>The system returns to the previous screen without deleting the staff member's account.</li> </ol>
<b>Non functional requirements</b>	<ol style="list-style-type: none"> <li><b>Security:</b> Only authorized admins should have the ability to delete staff member accounts to prevent unauthorized access and potential misuse.</li> <li><b>Data Integrity:</b> The system should ensure that all related data to the staff member is appropriately handled to maintain data integrity.</li> </ol>

<b>Postconditions</b>	If the deletion is confirmed, the staff member's account is permanently removed from the cinema management system. All access rights and associations related to the staff member are revoked or appropriately handled according to the system's data integrity policies.
-----------------------	---

<b>Use Case Name</b>	<b>UC003 - Revoking Specific User Permissions</b>
<b>Summary</b>	This use case describes the ability of the admin to revoke a unique permission to a specific user who before would have access to this ability.
<b>Dependency</b>	<b>Admin UC:</b> Add Users <b>Admin UC:</b> Manage Users
<b>Actors</b>	<b>Primary actor:</b> Admin <b>Secondary actor:</b> Other user(Manager, Staff, Client)
<b>Preconditions</b>	1. The admin has been logged in successfully
<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “user management” section in his panel</li> <li>2. The admin selects the option to “edit user information”</li> <li>3. The admin chooses the user they want to edit</li> <li>4. The admin selects which permissions they want the user not have anymore</li> <li>5. Admin confirms the revocation of this user’s permission.</li> <li>6. The admin saves changes</li> <li>7. If the changes save correctly, the admin gets a message</li> </ol>
<b>Description of the alternative sequence</b>	<b>Cancellation of revocation</b> <ol style="list-style-type: none"> <li>1. The admin navigates to the “user management” section in his panel</li> <li>2. The admin selects the option to “edit user information”</li> <li>3. The admin chooses the user they want to edit</li> </ol>

	<ol style="list-style-type: none"> <li>4. The admin selects which permissions they want the user not have anymore</li> <li>5. Admin does not confirm the revocation of this user's permission.</li> <li>6. The system displays the appropriate message indicating that the changes are not saved.</li> <li>7. Admin can choose to search again using a different criteria.</li> <li>8. The system returns to the previous screen without revoking the user's permissions.</li> </ol>
<b>Non functional requirements</b>	<ol style="list-style-type: none"> <li>1. <b>Security Measures:</b> Ensure the revocation process is secure to prevent unauthorized reinstatement of permissions</li> </ol>
<b>Postconditions</b>	The user's permissions have been successfully updated to exclude the revoked permissions, limiting their access to previously accessible functionalities or data.

<b>Use Case Name</b>	<b>Define Permissions</b>
<b>Summary</b>	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 permission to users whose user level should otherwise have it
<b>Dependency</b>	<b>Admin UC:</b> Add Users <b>Admin UC:</b> Manage Users
<b>Actors</b>	<b>Primary actor:</b> Admin <b>Secondary actor:</b> Manager, Staff, Client
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>2. The admin has been logged in successfully</li> </ol>
<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>8. The admin navigates to the "user management" section in his panel</li> <li>9. The admin selects the option to "edit user information"</li> <li>10. The admin chooses the user they</li> </ol>

	<p>want to edit</p> <ol style="list-style-type: none"> <li>11. The admin selects which permissions they want the user to have or not have</li> <li>12. The admin saves changes</li> <li>13. If the changes save correctly, the admin gets a message</li> </ol>
<b>Description of the alternative sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “user management” section in his panel</li> <li>2. The admin selects the option to “edit user information”</li> <li>3. The admin chooses the user they want to edit</li> <li>4. The admin selects which permissions they want the user to have or not have</li> <li>5. The admin saves changes</li> <li>6. 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</li> <li>7. If the changes save correctly, the admin gets a message</li> </ol>
<b>Non functional requirements</b>	<p><b>Performance:</b> The system saves the changes immediately</p> <p><b>Security:</b> The admin should be the only one to change these permissions</p> <p><b>Usability:</b> The user interface should be easy to understand and not overwhelming</p>
<b>Postconditions</b>	After choosing to go back, the admin should be sent back to the financial reports section

<b>Use Case Name</b>	<b>View Revenue Reports</b>
<b>Summary</b>	This use case describes the ability of the admin to access a detailed report of the revenues of each cinema during any specific period of their liking.
<b>Dependency</b>	<p><b>Staff UC:</b> Ticket selling</p> <p><b>Client UC:</b> Booking Tickets</p>
<b>Actors</b>	<b>Primary actor:</b> Admin
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The admin has been logged in successfully</li> </ol>

<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “financial management” section in his panel</li> <li>2. The admin selects the option to view revenue reports</li> <li>3. The admin chooses which cinema/s they want a report on</li> <li>4. The admin chooses the period for which they want a revenue report</li> <li>5. The system retrieves the revenue data and displays it to the admin</li> </ol>
<b>Description of the alternative sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “financial management” section in his panel</li> <li>2. The admin selects the option to view revenue reports</li> <li>3. The admin chooses which cinema/s they want a report on</li> <li>4. If the admin enters an invalid period the application will let him know that he should add a valid one</li> <li>5. The admin enters a valid period</li> <li>6. The system retrieves the revenue data and displays it to the admin</li> </ol>
<b>Non functional requirements</b>	<p><b>Performance:</b> The system generates the revenue report quickly</p> <p><b>Security:</b> The reports should be only visible to the admin</p> <p><b>Usability:</b> The information should be presented in a way that is both detailed and easy to understand</p>
<b>Postconditions</b>	After choosing to go back, the admin should be sent back to the financial reports section

<b>Use Case Name</b>	<b>View Expense Reports</b>
<b>Summary</b>	This use case describes the ability of the admin to access a detailed report of the expenses of each cinema during any specific period of their liking.
<b>Dependency</b>	<p><b>Admin UC:</b> Add Users</p> <p><b>Admin UC:</b> Manage Users</p>
<b>Actors</b>	<b>Primary actor:</b> Admin
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The admin has been logged in</li> </ol>



	successfully
<b>Description of main sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “financial management” section in his panel</li> <li>2. The admin selects the option to view expense reports</li> <li>3. The admin chooses which cinema/s they want a report on</li> <li>4. The admin chooses the period for which they want a expense report</li> <li>5. The system retrieves the expense data and displays it to the admin</li> </ol>
<b>Description of the alternative sequence</b>	<ol style="list-style-type: none"> <li>1. The admin navigates to the “financial management” section in his panel</li> <li>2. The admin selects the option to view expense reports</li> <li>3. The admin chooses which cinema/s they want a report on</li> <li>4. If the admin enters an invalid period the application will let him know that he should add a valid one</li> <li>5. The admin enters a valid period</li> <li>6. The system retrieves the expense data and displays it to the admin</li> </ol>
<b>Non functional requirements</b>	<p><b>Performance:</b> The system generates the expense report quickly</p> <p><b>Security:</b> The reports should be only visible to the admin</p> <p><b>Usability:</b> The information should be presented in a way that is both detailed and easy to understand</p>
<b>Postconditions</b>	After choosing to go back, the admin should be sent back to the financial reports section

<b>Use Case Name</b>	<b>Financial Feedback</b>
<b>Summary</b>	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
<b>Dependency</b>	None

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

## 2.CLIENT CASES

Track upcoming movies:

Use Case Name	Track upcoming movies
Summary	This use case describes the ability of a client to view upcoming movies with details and set reminders or mark movies as favorites to 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	<b>-Reliability</b> Reminders should be reliable, ensuring clients receive notifications at the right time to stay informed about their favourite movie. <b>-Performance</b> The app should load the list of upcoming

	movies quickly, so clients do not have to wait too long to see what is new.
<b>Postconditions</b>	After using the app, clients have seen the list of movies and set reminders for their favourites.

#### Enroll in loyalty programs:

<b>Use Case Name</b>	<b>Enroll in loyalty programs</b>
<b>Summary</b>	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.
<b>Dependency</b>	<b>Client case:</b> Claim benefits and apply promotions
<b>Actors</b>	Primary actor: Clients
<b>Preconditions</b>	1.The client has been logged in successfully.
<b>Description of main sequence</b>	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.
<b>Description of the alternative sequence</b>	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.
<b>Non functional requirements</b>	<b>-Security</b> The information of the client should be securely stored and managed.

	<b>-Performance</b> The process of enrolling should be easy to understand and navigate quickly for clients.
<b>Postconditions</b>	After using the app, clients are successfully enrolled in the loyalty program.

#### Claim benefits and apply promotions :

Use Case Name	Claim benefits and apply promotions
<b>Summary</b>	<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. For instance, clients will have discounts on special occasions or festive days. Moreover, clients can avail discounts through partner programs like Vodafone coins, where they can redeem coins for discounts on movie tickets or snacks. Furthermore, certain limited groups, such as students or groups over 20 people will have discounts as well.</p>
<b>Dependency</b>	<b>Client Case:</b> Enroll in loyalty programs <b>Client Case:</b> Contact customer support
<b>Actors</b>	Primary actor: Clients Secondary actor: Staff
<b>Preconditions</b>	1.The client has been logged in successfully.

<b>Description of main sequence</b>	<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"> <li>-Client views the desired benefit, such as redeeming reward points for free tickets, accessing exclusive discounts, or receiving personalized offers.</li> <li>-Client selects the desired benefit.</li> <li>-Client follows the instructions to claim the benefits.</li> <li>-The system processes the request and applies the benefit to the client’s account.</li> </ul> <p>3.If not enrolled:</p> <ul style="list-style-type: none"> <li>-Client views the list of prices or promotions available in the app.</li> <li>-Client selects the desired promotion.</li> <li>-Client follows the instructions to apply the promotion.</li> <li>-The system processes the request and applies the promotion to the client’s account.</li> </ul>
<b>Description of the alternative sequence</b>	<p>If the app has technical issues:</p> <ul style="list-style-type: none"> <li>-Client receives a notification informing them for the issue.</li> <li>-Client is asked to try again later or contact customer support.</li> </ul>
<b>Non functional requirements</b>	<p><b>Security</b></p> <ul style="list-style-type: none"> <li>-The personal information of the client, such as loyalty program details and transaction history should be securely stored and managed.</li> </ul> <p><b>Availability of instructions</b></p> <ul style="list-style-type: none"> <li>-The app should provide clear instructions for clients to claim benefits and apply for promotions.</li> </ul>
<b>Postconditions</b>	<p>After using the app, client has successfully claimed benefits or applied promotions.</p>

**Contact customer support:**

<b>Use Case Name</b>	<b>Enroll in loyalty programs</b>
----------------------	-----------------------------------

<b>Summary</b>	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.
<b>Dependency</b>	<b>Client case:</b> Claim benefits and apply promotions
<b>Actors</b>	Primary actor: Clients Secondary actor: Staff
<b>Preconditions</b>	1.The client has been logged in successfully.
<b>Description of main sequence</b>	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.
<b>Description of the alternative sequence</b>	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. 2.If the client does not receive updates on their support request: -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.
<b>Non functional requirements</b>	<b>-Security</b> The conversation between the client and the staff should be secure and can't be seen by anyone else. <b>-Performance</b> The messages get sent in real time, without any delay. <b>-Usability</b> The user interface should be easy to understand and not overwhelming.
<b>Postconditions</b>	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.
--	---



### 3.MANAGER CASES

<b>Use Case Name</b>	<b>Tracking Staff Member revenues</b>
<b>Summary</b>	This use case describes the involvement of Managers in tracking the revenues (Income – Spending = Revenue) made from staff using the cinema software
<b>Dependency</b>	<b>Staff UC:</b> Ticket selling <b>Staff UC:</b> Stocking ???
<b>Actors</b>	<b>Primary actor-</b> Manager <b>Secondary actor-</b> Staff
<b>Preconditions</b>	The manager has logged on successfully The Staff account already exists
<b>Description of main sequence</b>	<ol style="list-style-type: none"><li>1) The manager opens the performance tab from the main view.</li><li>2) The manager picks from a series of already existing staff.</li><li>3) The manager chooses the “Revenue” button.</li><li>4) The manager enters a period or clicks the “Overall Revenue” button</li></ol>
<b>Description of the alternative sequence</b>	<ol style="list-style-type: none"><li>1) The manager opens the performance tab from the main view.</li><li>2) The manager picks from a series of already existing staff.</li><li>3) The manager chooses the “Revenue” button.</li><li>4) If the Manager enters a period that is not valid the system alerts him to try a valid period.</li><li>5) The manager enters a valid period or clicks the “Overall Revenue” button.</li></ol>
<b>Non functional requirements</b>	<b>Security:</b> The staff revenue should be viewed only by managers. <b>Performance:</b> The program should display the information quickly, to make it possible to

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

<b>Use Case Name</b>	<b>Tracking Staff Member Income</b>
<b>Summary</b>	This use case describes the involvement of Managers in tracking the income made from staff using the cinema software
<b>Dependency</b>	<b>Staff UC:</b> Ticket selling
<b>Actors</b>	<b>Primary actor-</b> Manager <b>Secondary actor-</b> Staff
<b>Preconditions</b>	1) The manager has logged on successfully 2) The Staff account already exists
<b>Description of main sequence</b>	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
<b>Description of the alternative sequence</b>	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
<b>Non functional requirements</b>	<b>Security:</b> The staff income should be viewed only by managers. <b>Performance:</b> The program should display the information quickly, to make it possible to check a vast number of staff faster.
<b>Postconditions</b>	After choosing to go back, the Manager should be sent back to the Performance

	section
--	---------

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

	section
--	---------

**4.STAFF CASES**