

CS335
Group Project on Software Engineering
Group 24

Topic- Media

FlixNet Streaming Service

Contributors:

Josh Martin

Riona Lawless

Matthew O'Rourke

Contents

User Stories	3
Functional Requirements.....	4
Non-Functional Requirements	4
Use Case Diagram	4
Sequence Diagrams	5
Reset Login	5
Save Payment Details	6
Delete User Account.....	7
Get User Information	7
Set Interests and receive recommendations.	8
Publish Content to the streaming service.....	8
Remove Content from Streaming Service	9
Class Diagram	9
Architecture Diagram	10
User Interface	11
Test Design	11
Conclusion	13

User Stories

- As a user I want to be able to create an account so that I can use the service.
The system should provide a user-friendly interface for users to create their accounts seamlessly.
- As a user I want to be able to retrieve my login information when I forget it so that I can continue to use my account. The system should provide a mechanism for users to retrieve their login information (e.g., username or password) when forgotten. This allows users to regain access to their accounts without the need for manual intervention from support staff.
- As a user I want to be able to update my subscription plan so that I can get access to a greater range of features of the service. I should be able to choose between a free basic plan and a more premium plan without advertisements.
- As a user I want to be able to store my payment details so that I can be charged automatically for my subscription. This will enhance the reliability of my experience using the streaming service as every time I use the service, I can be assured my subscription will be still active without me having to manually renew it. This is a functional requirement.
- As an admin I want to be able to remove user accounts so that I can maintain a clean and efficient user database. This ensures greater optimization for active users as well as decreasing security risk from inactive accounts.
- As an admin I want to be able to view all user account information so that I can manage current users. By having a complete overview of user accounts, I can identify trends, address issues promptly, and make informed decisions to optimize the user experience. This also allows me to monitor engagement patterns and identify high value users.
- As a user I want to be able to set my interests so that I receive better recommendations. This ensures that the recommendations that I receive are relevant to my tastes as well as enhancing my satisfaction with the platform by showing my content I will enjoy. I will also be more likely to recommend the service to friends if I have a better experience with the content I am provided.
- As a user I want to receive recommendations based on my interests as well as my viewing history. This will save me time as I won't have to go searching far for the content I want to watch as well as allowing me to pick up where I left off on content I haven't yet finished watching.
- As a content provider I want to be able to publish new media on the site so that it is viewable by users. This will allow me to expand my audience and maintain an active presence on the site.
- As a content provider I want to be able to withdraw my content from the platform so that it is no longer available for streaming. This allows me to maintain control over its availability for streaming, whether it's due to licensing agreements, content updates, or strategic considerations.

Functional Requirements

The system shall store account information of all current users.

The system shall allow new users to register with the streaming service.

The system shall allow users to update their account information.

The system shall allow users to update their subscription type.

The system shall allow users to recover their login information in cases where they have forgotten.

Non-Functional Requirements

The system shall notify users about important account activity like subscription changes, payment errors, change in account information and system updates.

The system shall store passwords securely using encryption techniques.

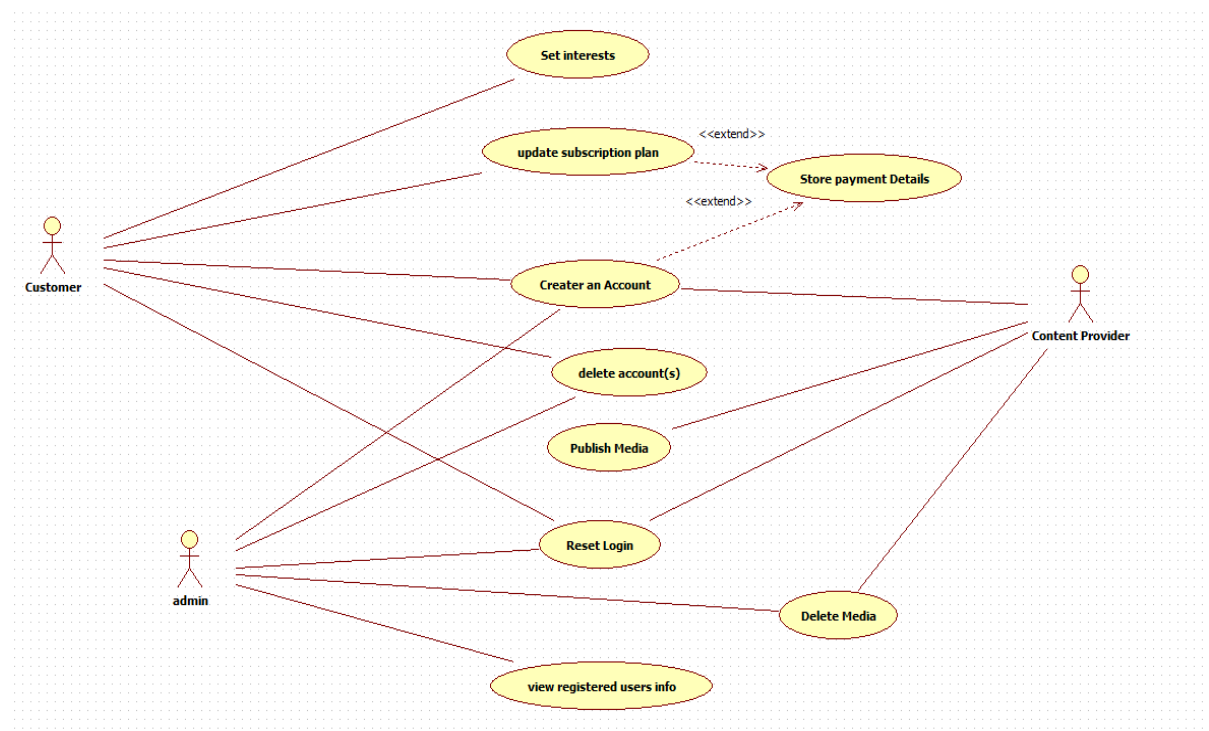
The system shall enforce strong password standards on users making new passwords.

The system shall be able to handle a large number of users and account transactions.

The system shall be capable of scaling to accommodate for future growth.

The system shall regularly back up user accounts data to prevent losses in the event of system failure.

Use Case Diagram



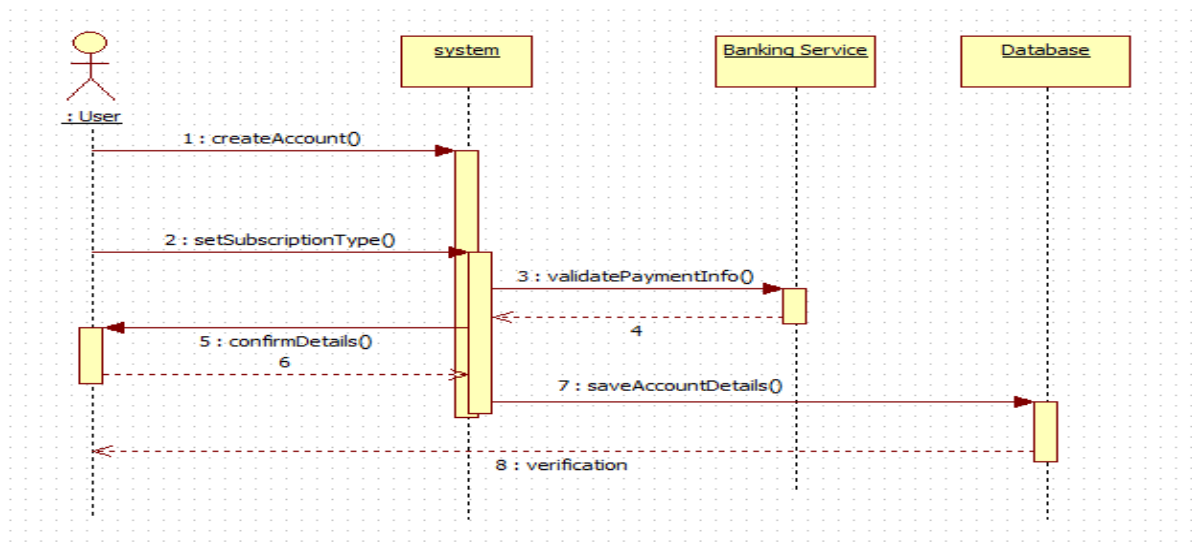
The customers can create or delete their account, update their subscription plan, set their interests, and reset their login.

The admins can create or delete accounts, reset login, view information about registered users and delete content.

The content providers can create an account, reset login information, publish media, and delete media.

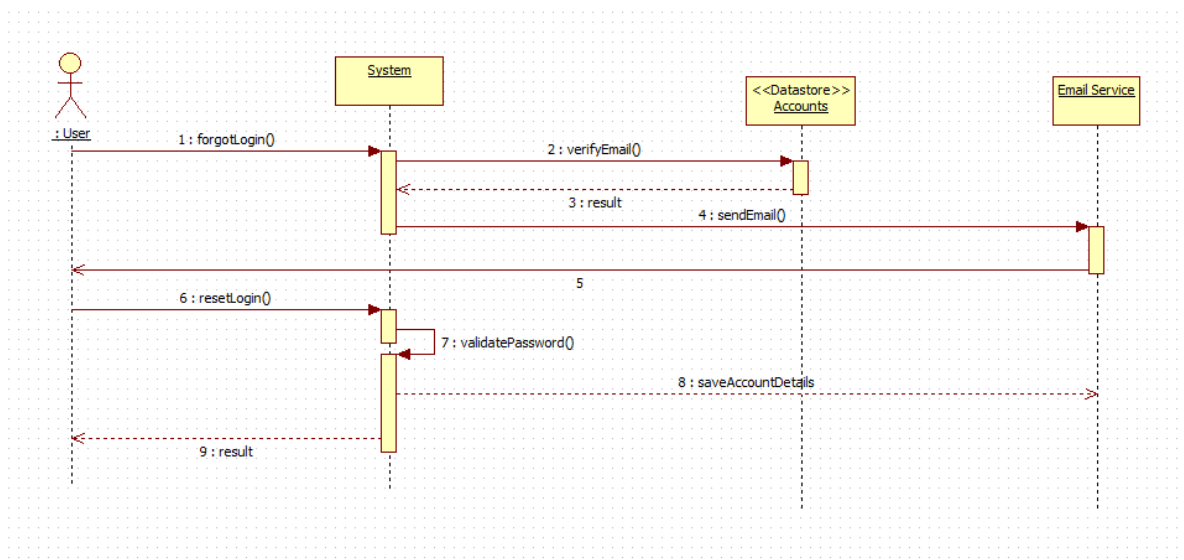
Sequence Diagrams

Create an account.



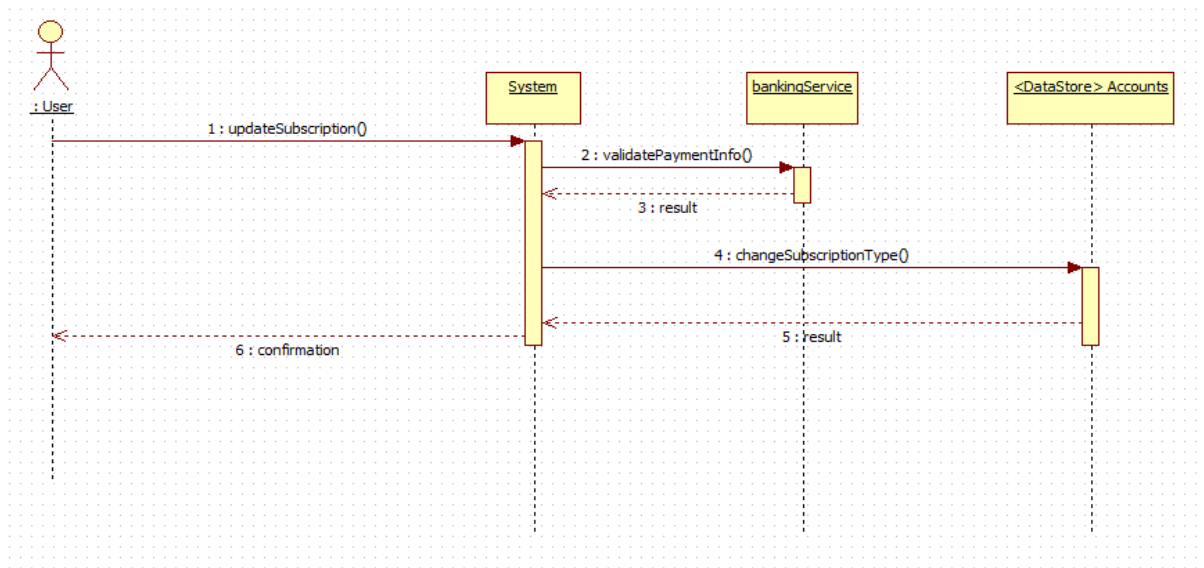
The user enters their new account details which are sent to the system. They are then prompted to select the type of subscription they want as well as their payment information. This payment info is then validated with their banking service and the user is asked to confirm their new account details. These details are then stored in the account information database.

Reset Login



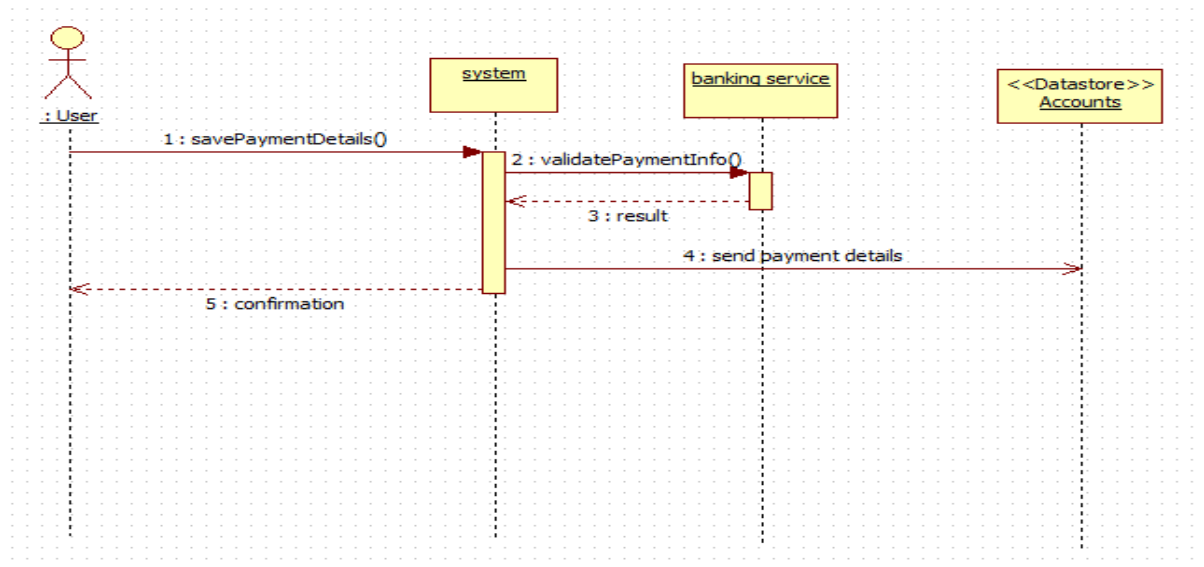
The user enters their email to the system which checks the database and sends an email to the user if their email is in the database. The user then enters a new password for their account which is validated to make sure it adheres to the strong password standards. The account details are updated in the database with the user's new information.

Update Subscription



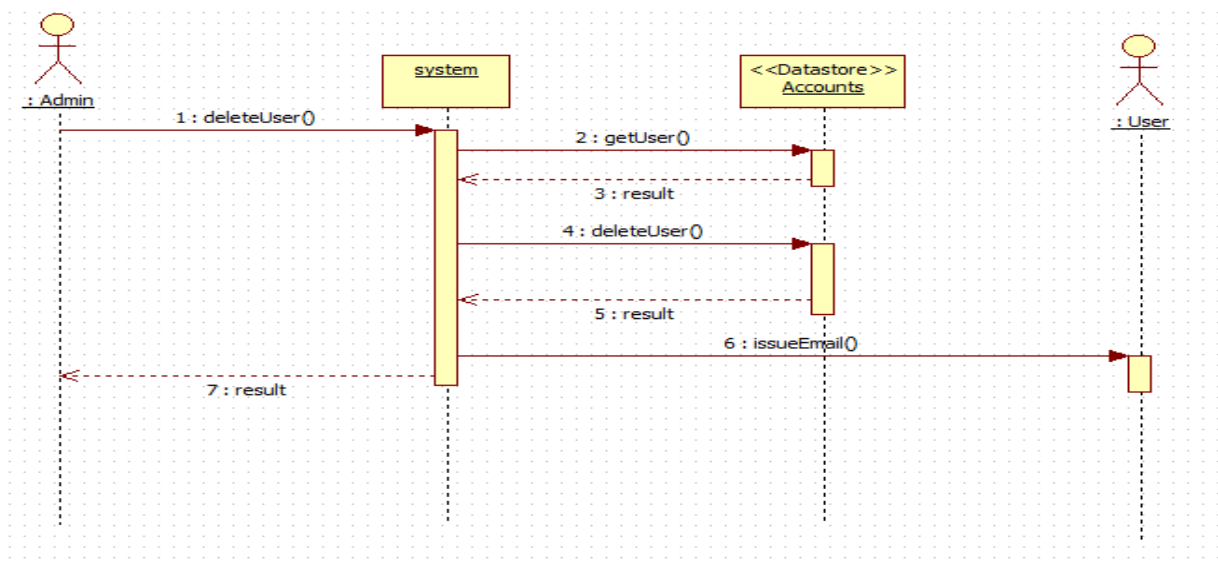
The user's new subscription type is sent to the system. Their payment information is then validated and the subscription type for their account in the database is changed.

Save Payment Details



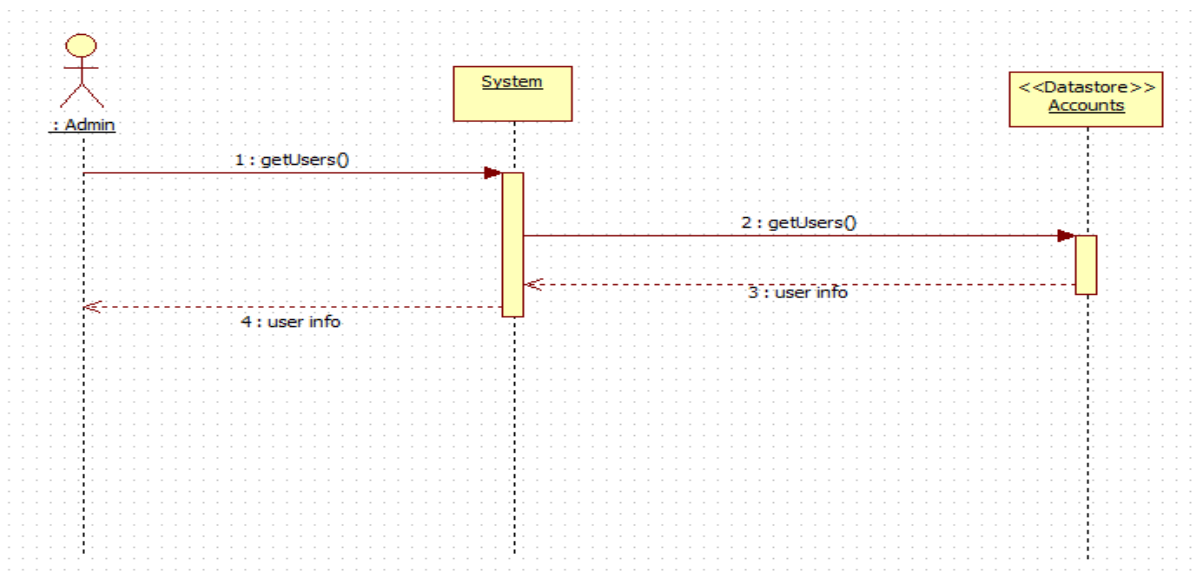
The user sends their payment details to the system. These payment details are then validated and saved in the account database.

Delete User Account



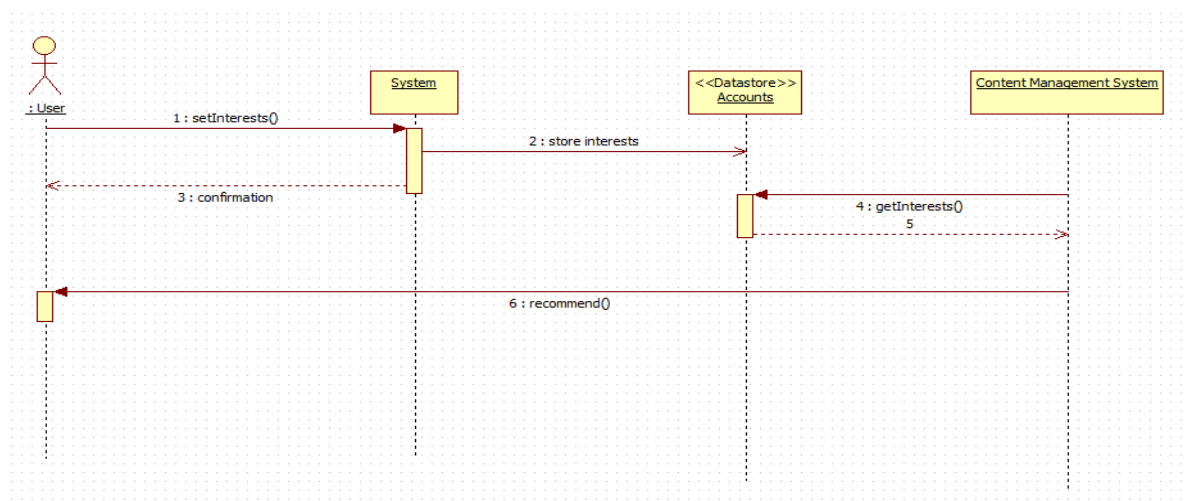
The admin provides the details of the account to be deleted to the system. The system then checks for this account and deletes any matches. An email is then issued to the user stating that their account has been terminated.

Get User Information



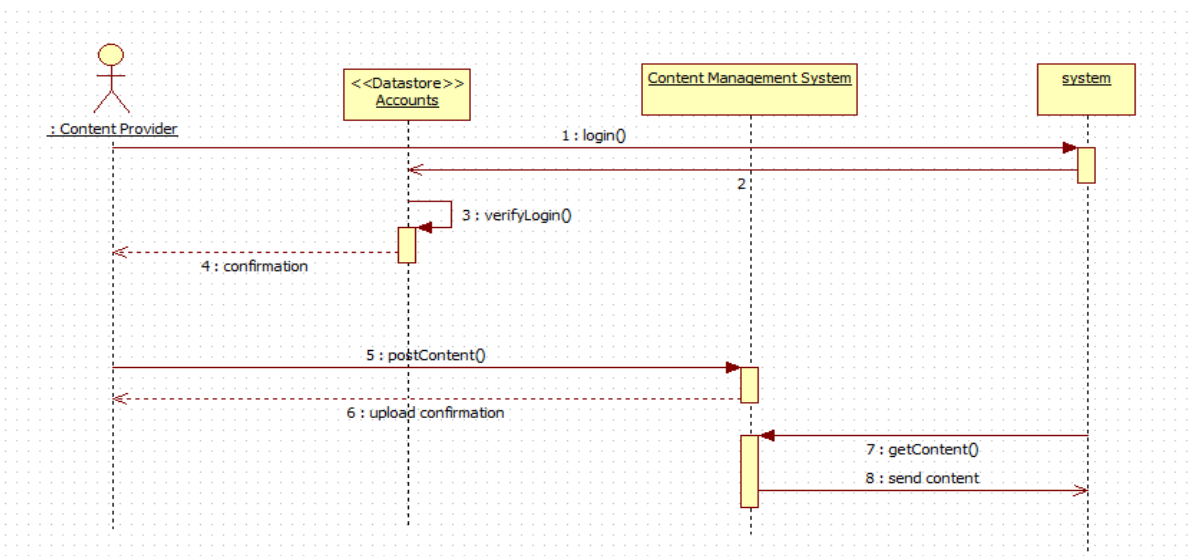
The admin requests user information from the system which then fetches this data from the accounts database and returns it to the admin.

Set Interests and receive recommendations.



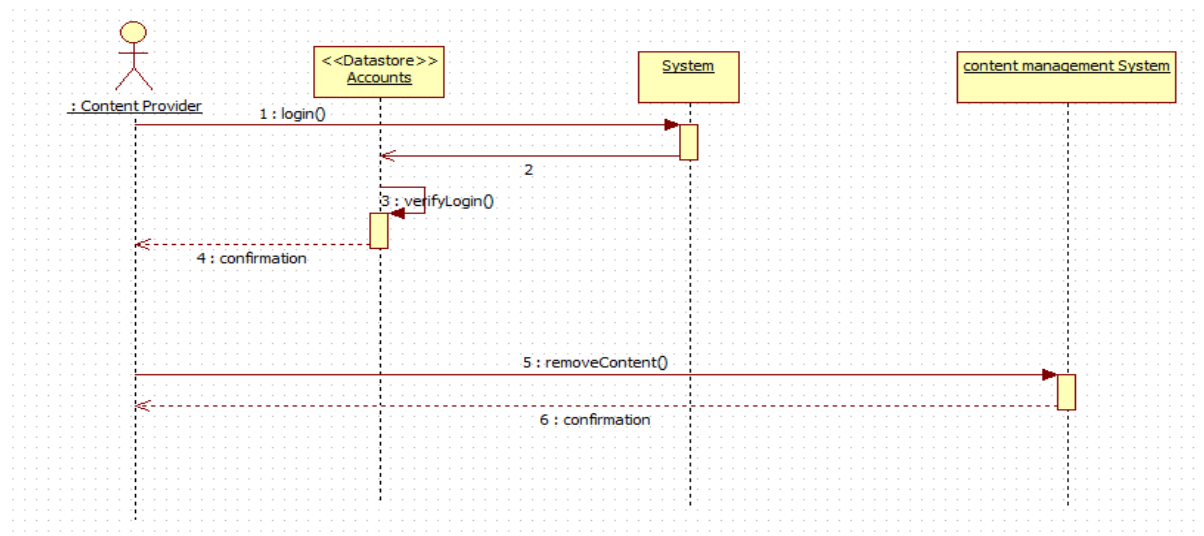
The user sends their interests to the system which stores these in the user accounts database. The Content Management System then uses these interests to find content suited for the user.

Publish Content to the streaming service.



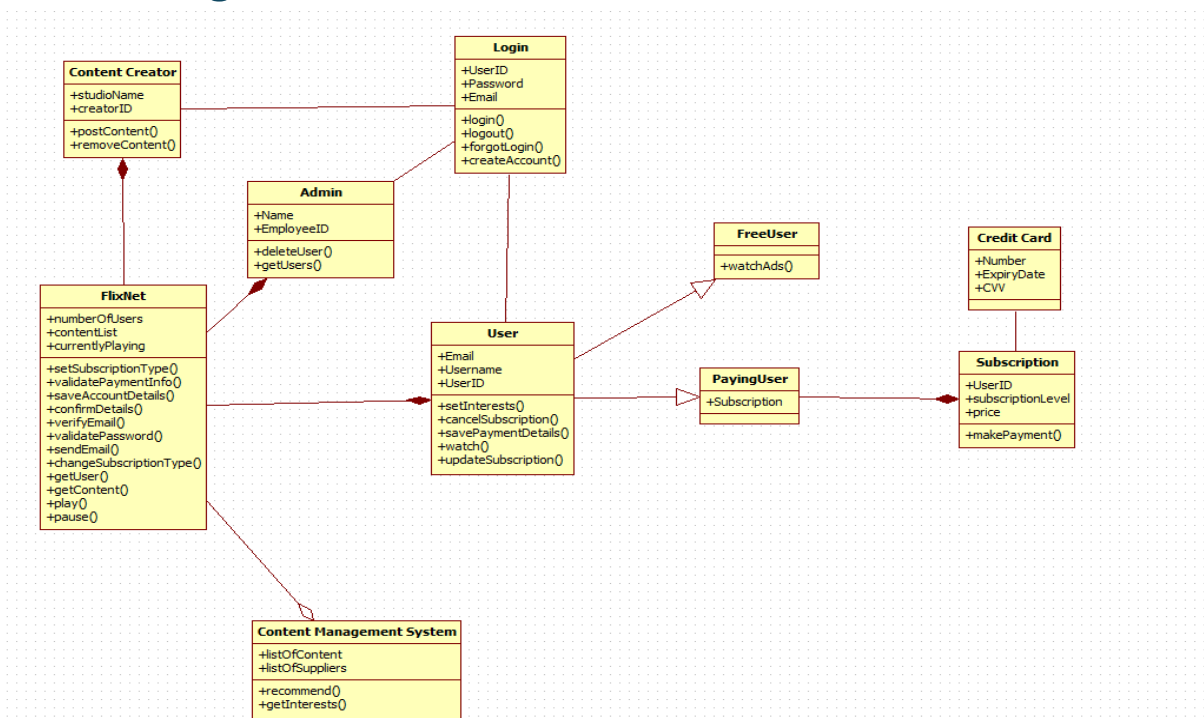
The content provider logs in to the system using their login credentials which are then verified against the credentials in the database. The Content provider can then send content to the content management system which can be fetched by the system to push to users.

Remove Content from Streaming Service



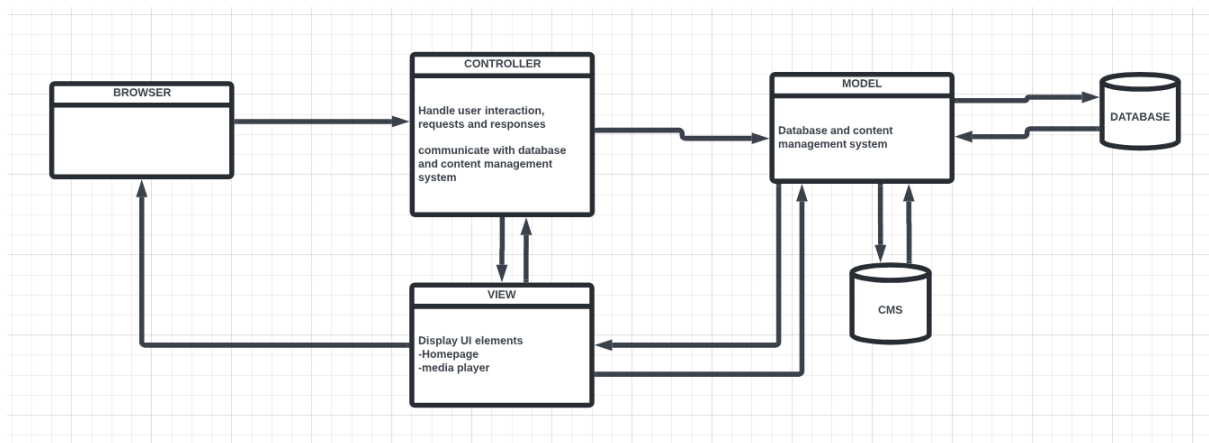
The content provider logs in to the system using their login credentials which are then verified against the credentials in the database. The content provider can then select content to be removed from the content management system.

Class Diagram



The FlixNet system provides the typical features of a streaming service. The FlixNet class represents the core system and its features. These features are then expanded by different classes based on the type of user you are. These users cannot exist without the FlixNet class, so the relationship is composition. The content Management system provides and manages the content provided to the streaming service by different content creators. These content management systems can be used by many streaming services and are not dependent on the FlixNet class to exist.

Architecture Diagram



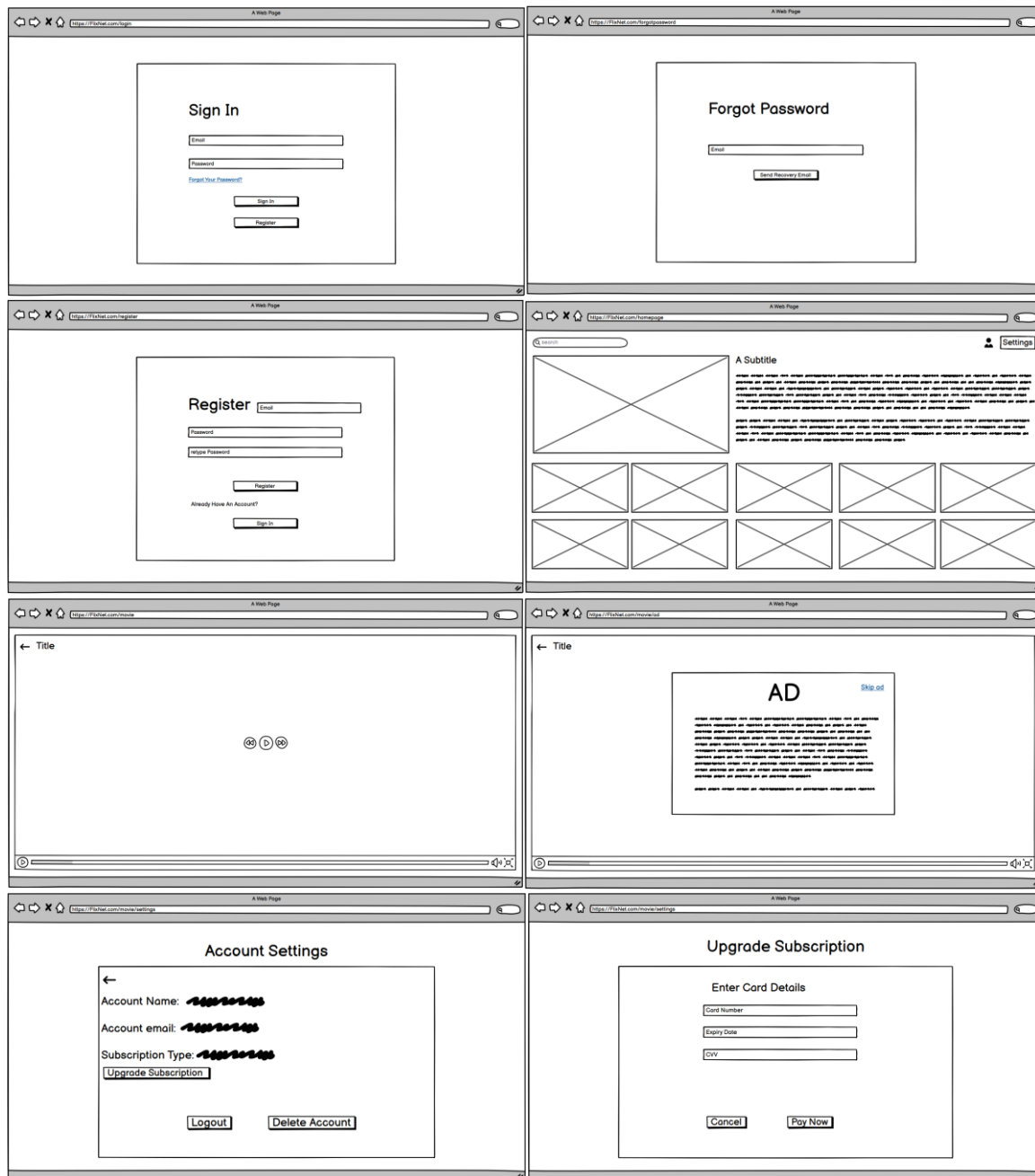
The browser is the interface through which the user interacts with the streaming service. It sends requests to the server and receives responses, rendering the UI based on these responses.

The controller is the middleman between the UI and the backend data stores. It processes user interaction and invokes the appropriate actions. It manages the flow of data between the model and the view.

The model represents the data structures of the system. It is responsible for querying and manipulating the database as well as managing content stored in the CMS.

The view is responsible for presenting the UI to the user based on data received from the model. It receives data from the controller and renders it for display in the browser.

User Interface



The mockup above shows how a typical user would navigate the interface of the streaming service.

Signing in, they are greeted by the homepage with all the relevant content suggestions based on their interests. They can also use the search feature to find specific content. From the homepage they can choose a video to watch, which takes them to the media player, or they can navigate to their account settings where they will be able to set their interest or edit account details.

Test Design

The tests designed below are aimed to discover defects in expected and critical areas before the service becomes available to end user.

These tests also demonstrate to the customer that the software meets its expected requirements.

Scenario	Test Step	Expected Result	Actual Outcome
Verify that a user can create an account	Provide valid user information on the registration page (email, password, username)	User account should be successfully created, and the user should be logged in.	The account is created, and the user is logged in.
Verify that a user can create an account	Provide invalid user information on the registration page (nonexistent email)	The account creation should fail and notify the user as to why	The account creation fails, and the user is notified as to why
Verify that users can retrieve forgotten login information	Request password reset or username reminder	The user should receive an email with instructions on how to reset their password/ reminder of their username	The user receives an email with instructions on how to reset their password/ reminder of their username
Verify that a user can update their subscription plan.	Enter valid payment details on the 'upgrade subscription' page	The user's account should be upgraded from free to premium	The user's account is upgraded from free to premium
Verify that a user can update their subscription plan.	Enter invalid payment details on the 'upgrade subscription' page	The user should be notified that invalid payment details were entered and be prompted to reenter them	The user is notified that invalid payment details were entered and is prompted to reenter them
Verify that a user can save their payment details to their account	Enter valid payment details and check to box to "save payment details"	Payment details should be successfully stored, and the user's subscription should be automatically renewed each month	Payment details are successfully stored, and the user's subscription is automatically renewed each month
Verify that a user can save their payment details to their account	Enter invalid payment details and check to box to "save payment details"	Payment details should fail to be stored and the user should be prompted to reenter their payment details	Payment details fail to be stored and the user is prompted to reenter their payment details
Verify that an admin can remove user accounts	Admin selects a user account for deletion	User account should be successfully	User account is successfully

		removed from the database	removed from the database
Verify that admins can view all user accounts	Access the user accounts details from the admin view	Admin should be able to see all information about users registered.	Admin can see all information about users registered.
Verify that a user can set their interests	Select interests from the account settings page	User interests should be successfully saved to their account	User interests are successfully saved to their account
Verify that users receive relevant recommendations based on their interests	Analyze the recommendations provided to a user account and compare with their set interests	User should receive relevant recommendations based on their set interests	User receives relevant recommendations based on their set interests
Verify that content providers can publish new media to the site	Publish new media to the site and check if it is viewable by user accounts	User accounts should be able to view the new media	User accounts are able to view the new media
Verify that content providers can remove their content from the platform	Select the content to be removed and delete it	The content should be removed from the site and users will no longer be able to view it	The content is removed from the site and users can no longer view it

Conclusion

In conclusion, The FlixNet streaming service aims to offer a wide range of features for all kinds of users, with a focus on user experience, security, and scalability.

Features such as account creation, password recovery, subscription management and personalized recommendations are aimed at the users to ensure their experience with the platform is positive and to encourage them to engage further with it.

Administrators will be able to maintain the service efficiently and effectively with the administrative features of the system, such as user account management and content moderation as well as access to all the registered user information. The use case diagram, sequence diagrams, class diagram and architecture diagram clearly depict the system's structure and flow of interaction so administrators can easily comprehend the system.

Overall, the outlined requirements and architectural design aim to create a robust, user-friendly, and secure streaming service for all types of users.