RecNet

Akshat Sharma, Anay Kulkarni, Neel Makhecha, Swasti Choubey

295P Keystone Project

Winter 2022

1 Introduction

Our internet activity is heavily influenced by our personalities and our interests are reflected in the content we consume and who we follow. Our preferences across music, food, books and entertainment have a huge impact on the kind of people we hang out with. It is through this network of people with similar interests we often stumble upon other interesting content.

Many platforms leverage this idea to match people, suggest friends, partners, and more. For example, platforms like Spotify and YouTube use this fact to power recommendations through collaborative filtering. However, this data is shared by the users themselves, which often results in an inorganic definition of a users' personality and interests.

This can be fixed if we collect the actual data instead of creating it. Our idea is to build an engine that uses user centric data collected from multiple platforms like Spotify, Reddit, and Youtube to make cross-platform recommendations and find other users with similar taste profiles.

RecNet allows you to link multiple platforms such as Spotify, YouTube, and Reddit to learn about users through the music they listen to, videos they watch and like, and the content they share and follow. This gives us another opportunity to even make content recommendations of a specific platform based on the data from other platforms, potentially making these recommendations even more relevant to the users' interests. The idea stemmed from the use of Collaborative Filtering used in Recommender Systems. The goal of Recommender Systems is to provide the users with a list of recommended items that they prefer, or predict how much they might prefer each item. We referred to a few papers [1] [2] while conceptualizing RecNet that you can refer to as well. One of the biggest concerns of today's social media and dating platforms is catfishing. With RecNet, our goal is to eliminate (or at least reduce to a great extent) such activities by collecting "real" data, hence, letting people make genuine connections safely.

2 User Features

2.1 Social media and recommendations in one place

Users should be able to sign-up to RecNet and link existing Spotify, Reddit or other accounts for a more personalized and unique experience. Users can also unlink previously linked applications.

2.2 User and Content Recommendations

Access to user feed with panels/tabs for content recommendations and friend suggestions. Get the most out of the application by getting tailored friend recommendations based on your combined social media usage.

2.2.1 Find recommendations across categories tailored to your interests

Browse the content recommended by us or filter the user feed by platform, type of content (books, music, subreddit) and platform specific genres.

2.2.2 Find like-minded friends

Add friends to your feed based on similar interests across platforms. Chat with them and see other content they are following.

2.2.3 Real-time Feedback

Your liking or disliking a recommendation reflects in the model real-time, immediately changing any future recommendations.

3 Interface

The following are mere suggestions of what the user interface could look like. Developers have full autonomy over the final design. For the complete design refer to www.figma.com/RecNet.

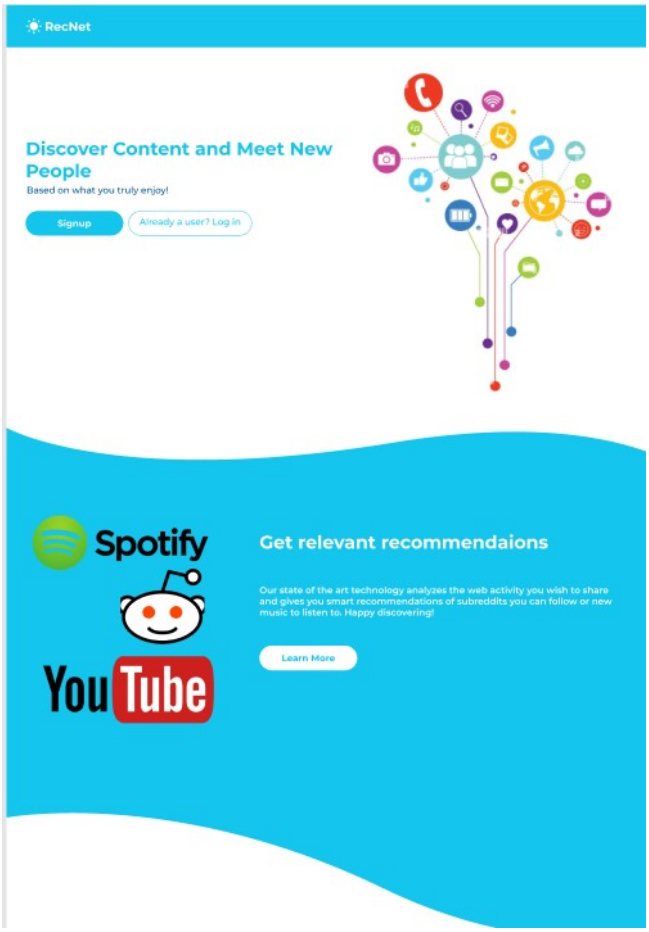

Figure 1: Landing Page Example

Figure 1 shows the welcome page where unauthorized users can find out about RecNet, reach out to developers and either login or sign-up. Once an authorized user logs out, they should be redirected back to the landing page.

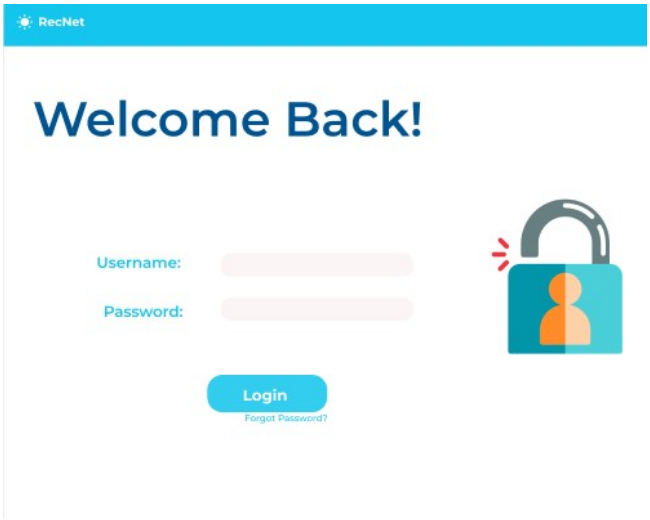

Figure 2: Login Page Example

The Login Page should be functional and minimalist. We need a username and password pair as the input which will be used to validate the user. An example has been shown in Figure 2. If authorized, they will move on to their feed otherwise they will be redirected to the registration page.

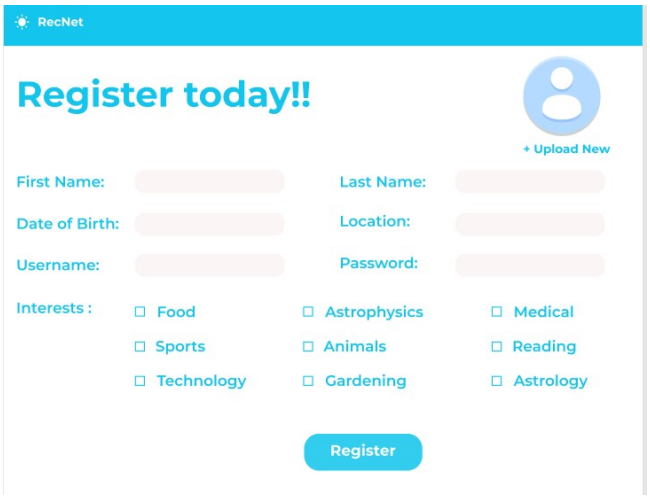

Figure 3: Registration Page Example

The registration page should allow the user to build their profile which means that besides asking for a username and password, other information such as name, location and interests should also be taken as input. It is up to the developers what information they would like from the users but an example has been provided in Figure 3.

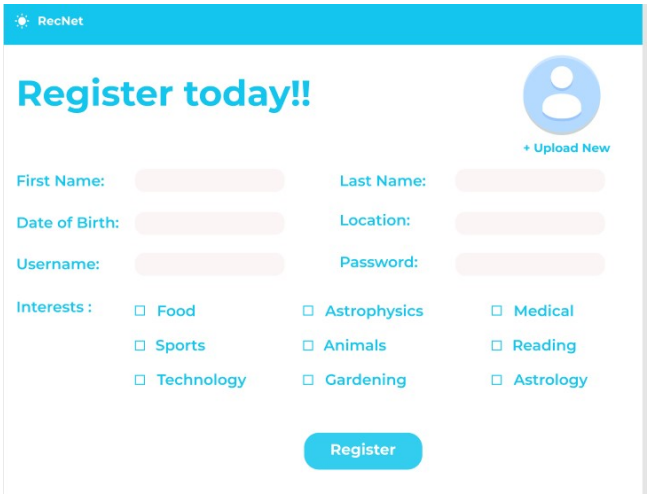

Figure 4: Friend Suggestion Page Example

The page should display other users that are recommended by the model. Apart from the profile and name, there should also be a short description stating why the user was recommended to them as shown in Figure 4.

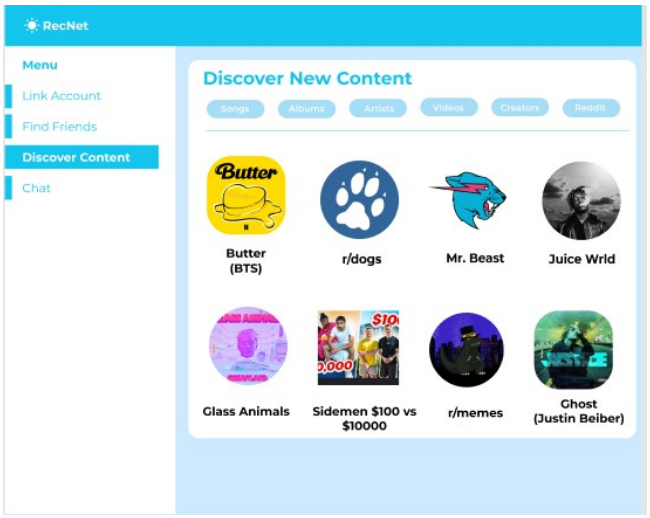

Figure 5: Content Recommendation Page Example

Similar to the Friend Suggestions page but this should display the content recommended by the model. The main page should display all content but the user should be allowed to filter through the recommendations. Refer to Figure 5 for an example.

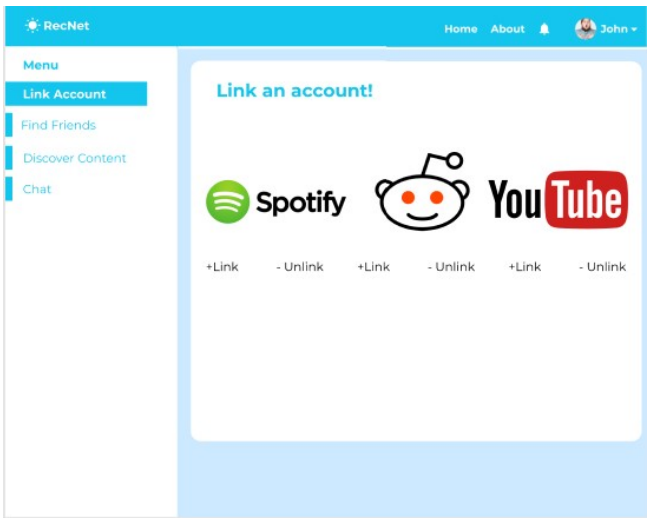

Figure 6: Account Link/Unlink Page Example

As exemplified in Figure 6, this page displays which account, from Spotify, Reddit and Youtube, is linked to RecNet by the user. It could be one or more and the user should be able to link or unlink the connected accounts.

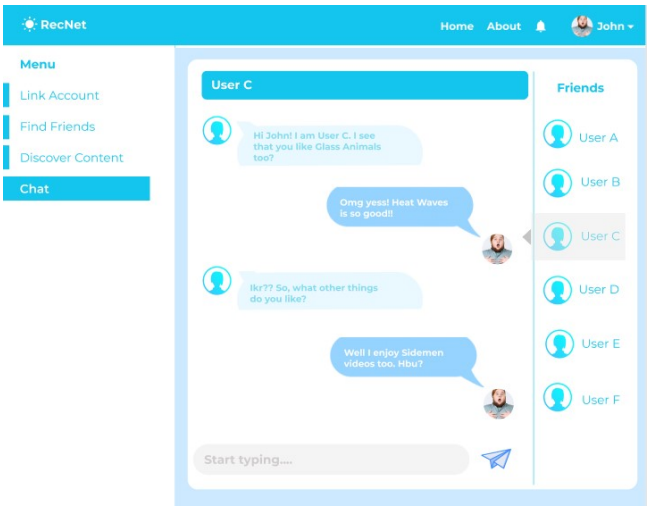

Figure 7: Friend Suggestion Page Example

A basic chat room. An example has been provided in Figure 7 but it is up to the developers. It should be easy to navigate and should display who a user is chatting with.

4 Architecture

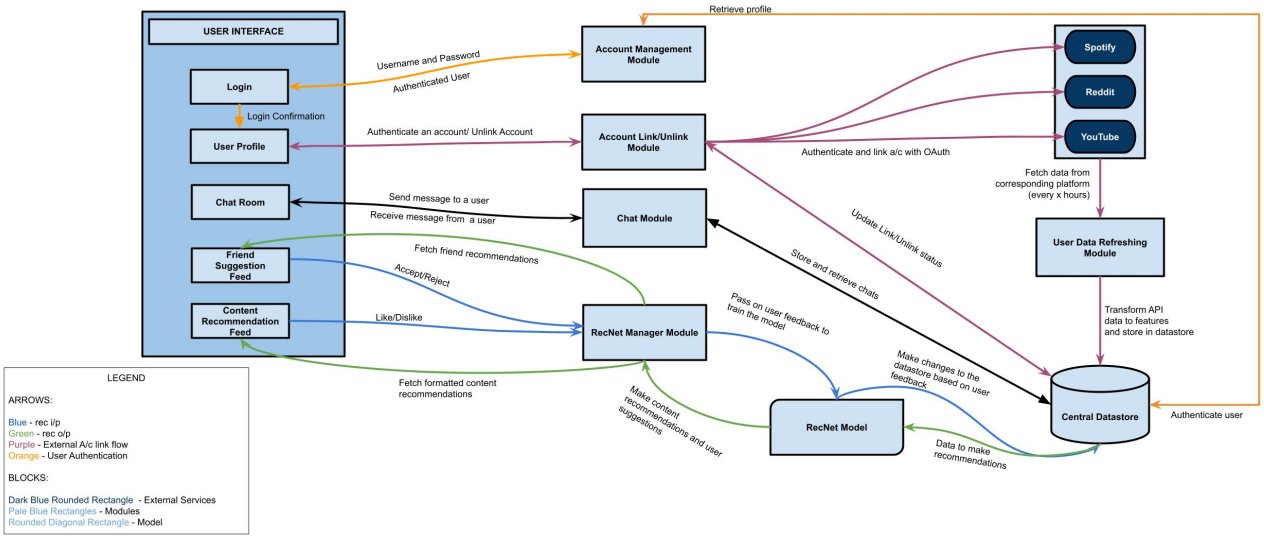

Figure 8: Architecture Diagram

Each block shown in Figure 8 has been explained in detail in Section 5 Modules. A clearer image has been provided in the Diagrams folder of this Git Repository: swchoubey/RecNet.

5 Modules

5.1 User Interface Module

Inputs: Data entered by user
Outputs: None
The user interface should be easy to navigate. A returning user will input login credentials and be redirected to the feed from where they can navigate as they wish. A new user should be redirected to the registration page where they will be requested to fill a form in order to set up a profile. Apart from authorization, the user interface also includes the content and friend recommendation feed along with the chat module, all of which have been explained in detail in the following sub-sections.

5.1.1 Login Module

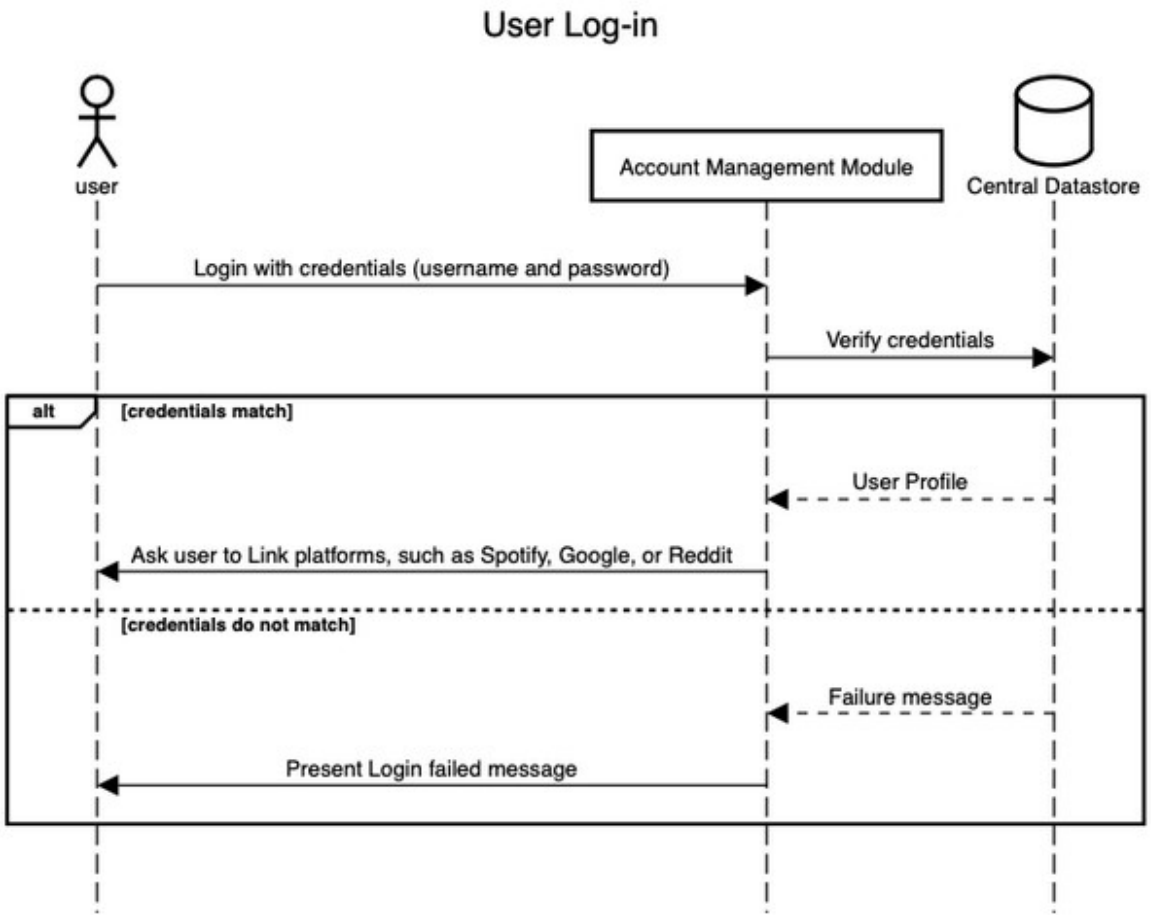

Figure 9: User Login

The landing page requests the user to either login or sign-up. Each new user has to sign up to authenticate access to the application. As shown in Figure 8, the login credentials are pushed through the Account Management Module to the central data-store and once authenticated, the user is redirected to the profile page. Figure 9 depicts the sequence of action performed by the user and the responses in a log-in process involving Account Management Module

5.1.2 User Profile

Only authenticated users can access the User Profile page which allows them to set a username, share any information they wish to, set a profile picture, add some interests, and link their Reddit, Spotify and/or Google

accounts. If the users wish to link any account then a request is pushed through the Account Link/Unlink Module to the requested external application as shown in Figure 8. The response from the application is then stored in the central datastore and a status response (Success/Failure) is pushed back to the profile page.

5.1.3 Chat Interface

A simple interface where a user can choose from a list of their friends who they wish to chat with and either start a new thread or continue one. This is an optional feature, it is up to the developers whether they want to include it or not.

5.1.4 Friend Suggestions Feed

The dashboard will be partitioned into friend suggestions and content recommendations. In the friend suggestions section of User A, those users would be suggested who have a similarity score (with A) close to one. The users will be displayed in a decreasing order of similarity upto a certain threshold value.

5.1.5 Content Recommendation Feed

The content recommendation feed will be available on the dashboard and display music, artists, subreddits, videos and content creators that a user can try out based on their previous activities. Users will be allowed to filter through the content by the above mentioned categories as well. Recommendations will also be made to them based on the content consumed by users with whom they have a similarity score close to 1 or above a certain threshold value.

5.2 Account Management Module

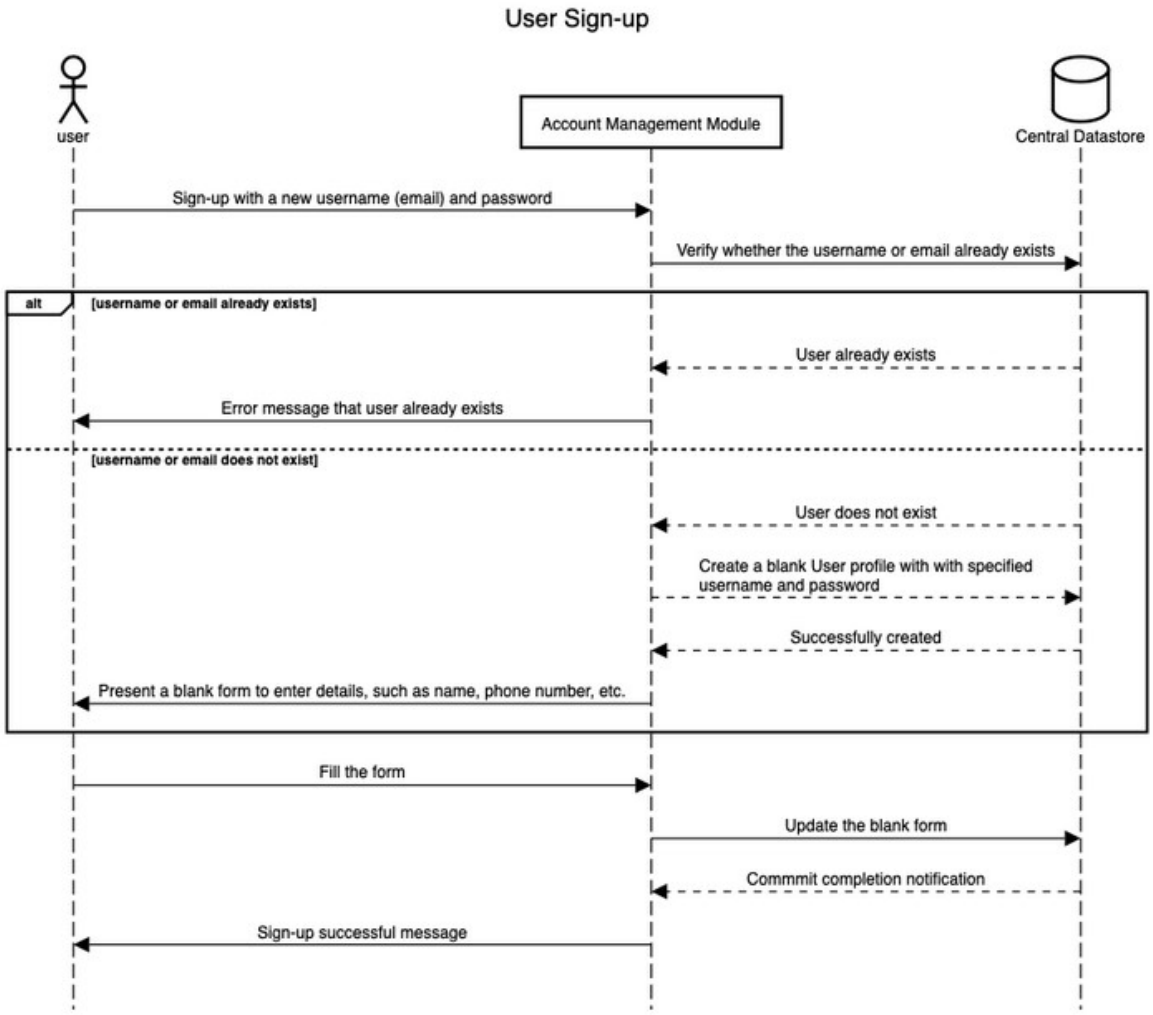

Figure 10: Account Management Module

Inputs: Username and Password from the Login Interface
Outputs: Authentication status (success or failure)
As indicated in Figure 8, the primary responsibility of the Account Management Module is to authenticate the user. So to provide basic functionalities of verifying the username and password, and then notifying the frontend about the authentication status. Additionally, this module should also provide functionalities of creating new users, updating user login information, such as password reset, account deletion, etc. Figure 10 depicts the sequence of actions and responses in a sign-up process involving Account Management Module

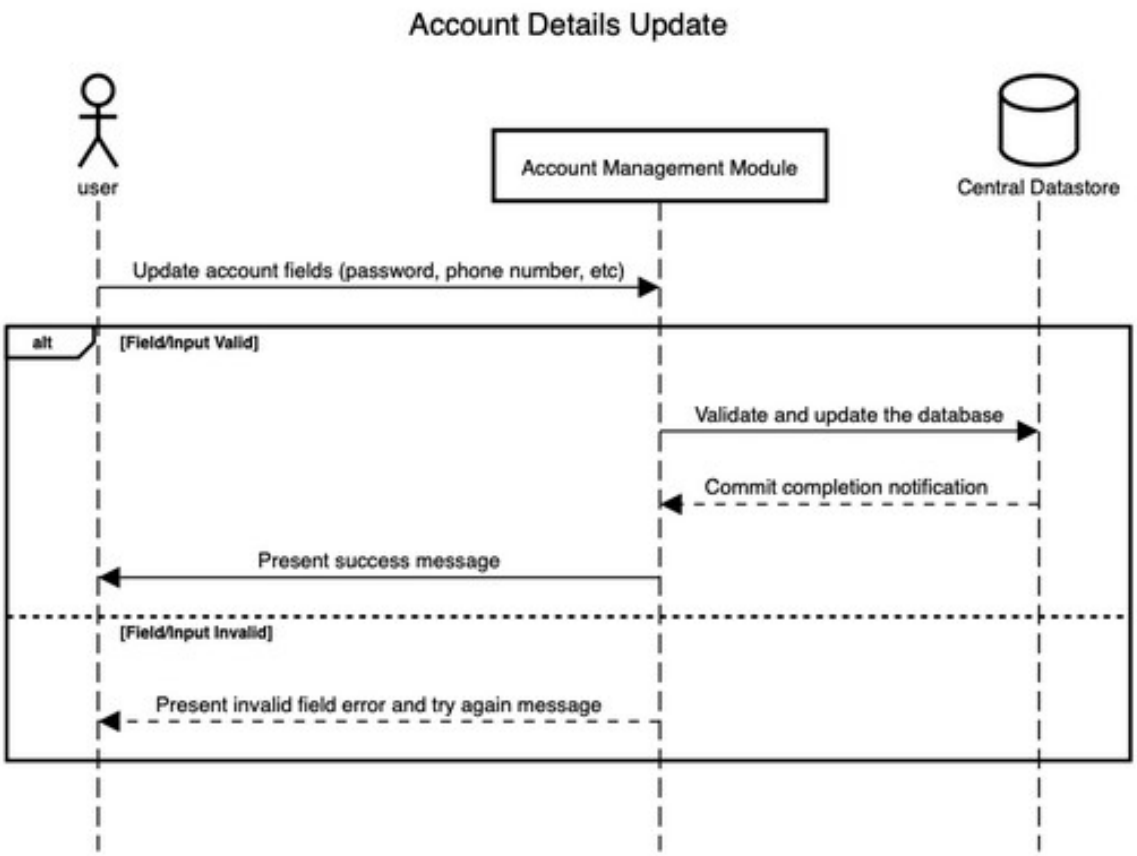

Figure 11: Account Details Update

Figure 11 depicts the sequence of actions and responses of the process of updating account details involving Account Management Module

5.3 Account Link/Unlink Module

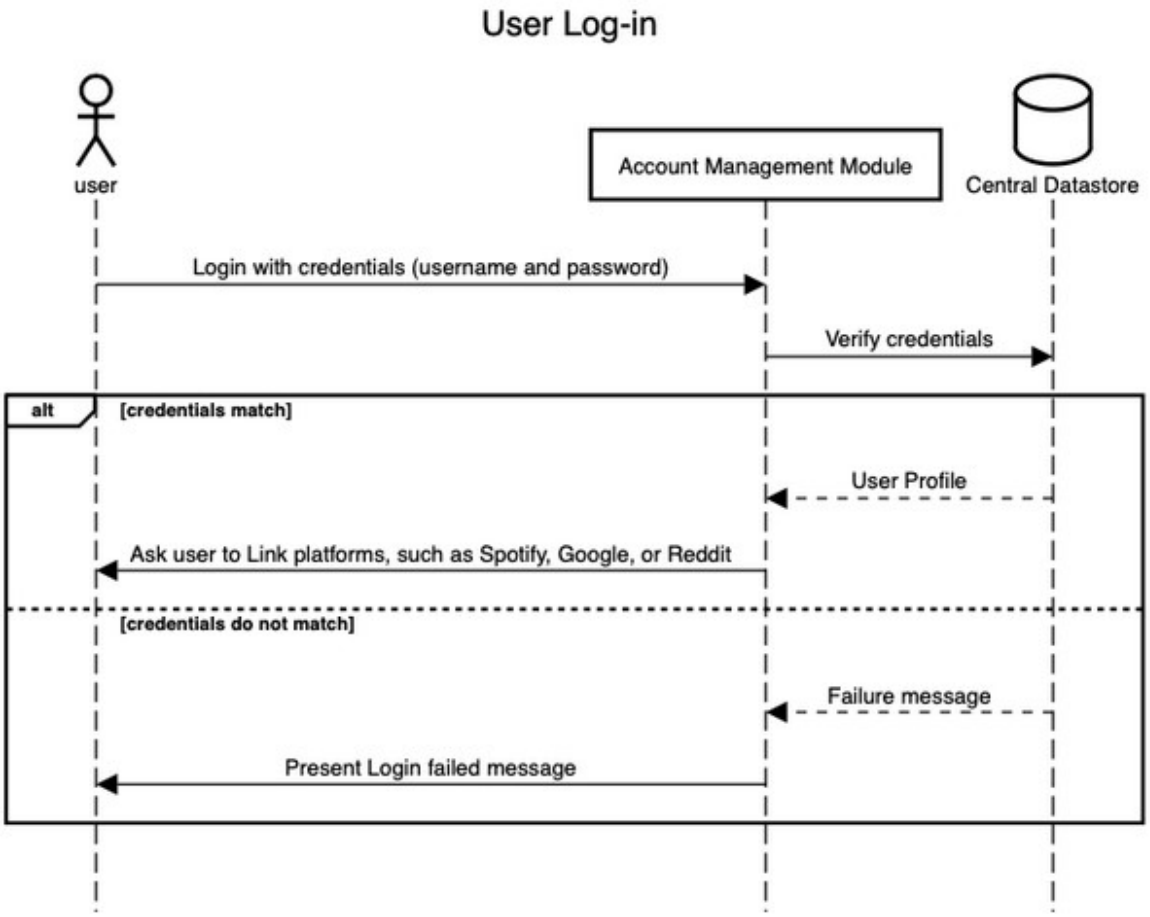

Figure 12: User Login

Inputs: User Profile, Action Type [Link / Unlink]
Outputs: Status - Success/Failure

This module is responsible for linking or unlinking a user's third-party platform account - Spotify, Reddit or Google. All these platforms let us use the OAuth2.0 authentication to receive tokens on behalf of users, giving us access to their information in the security scope granted to us by the user. The APIs required here include a Authorization callback URL to receive authorized tokens and the ability to revoke these permissions. Figure 12 depicts the user account linking/unlinking data flow

5.4 Chat Module

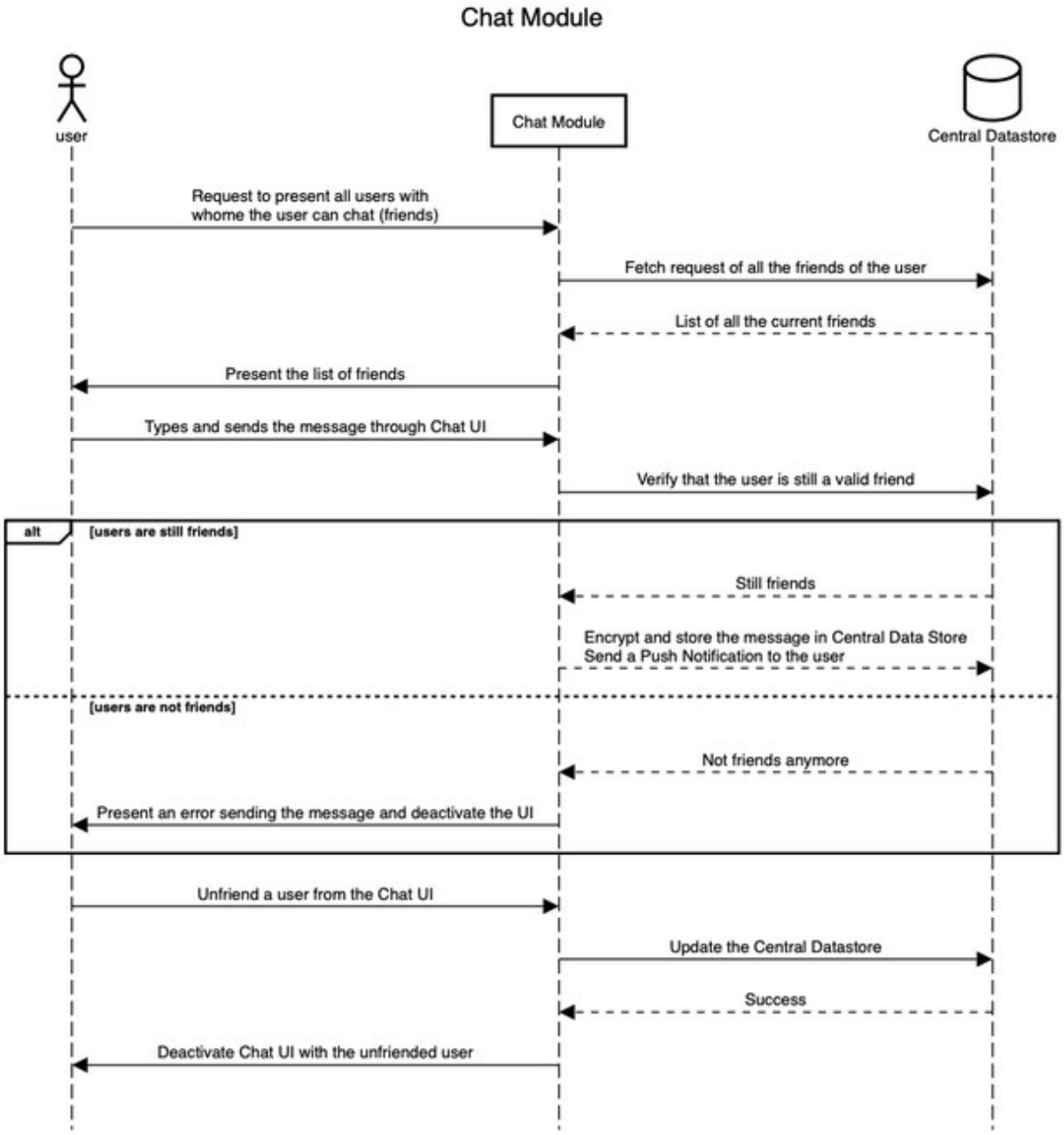

Figure 13: Chat Module

Inputs: Recipient user, message type, message content
Outputs: Status (Success/Failure/Denied)

The Chat module should provide necessary backend services to send and receive text messages (or possibly multimedia messages) to and from connected friends, and to save all the previous conversations to the central datastore. Communication should be allowed only when two users are actively friends, i.e. if one of them unfriends the other user, further communication should be stopped. Figure 13 depicts the working of the chat module, including flow of the messages, and the process to unfriend the users.

5.5 RecNet Manager Module

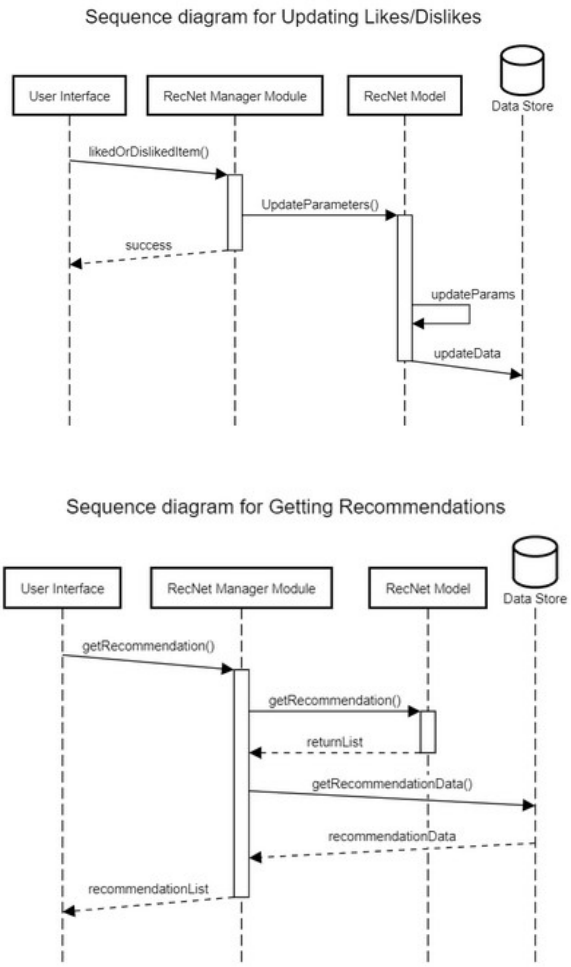

Figure 14: RecNet Manager Module

Recommendation API

Inputs: User ID, Recommendation for [Friend or Content]

Outputs: List of Data - Recommendations

Recommendation Feedback API

Inputs: User id, item, Liked or Disliked

Outputs: None

This module is responsible for providing an interface to the RecNet model and for making sure the input and output are in proper format for use. It formats the input and output before passing on the data from the UI to the model and from the model to the UI.

Recommendation API

It gets the user profile and sends the data on to the RecNet model module and fetches content/friend recommendations and parses and formats the content before passing it on to the UI.

Recommendation Feedback API

It takes in information as to if the user liked or disliked the suggestion and passes on the data to the model to make updates to the suggestions. The top sequence diagram in Figure 14 depicts the dataflow after a user likes or dislikes a suggestion. The bottom of Figure 14 shows the communication between the modules for getting the recommendation list for a particular user.

5.6 RecNet Model Module

Recommendation Module
Inputs: User Id, Recommendation Type [Content/Friend]
Output: List of Recommendations
Recommendation Feedback Module
Inputs: User id, Item id, Liked/Dislike
Output: Success [true / false]

This module is used for loading the pretrained model from storage and returning recommendations based on the user choices. It has the following responsibilities:

Recommendation Module
It makes friend suggestions based on user details as input to the model and previous data stored for the user. This module returns new friend suggestions or new content (spotify songs, subreddits, etc.) based on the likes and dislikes of the user.

Recommendation Feedback Module
It updates the user preferences based on if the user liked or disliked a specific recommendation by updating the importance associated with the categories of the recommendation.

5.7 User Data Refreshing Module

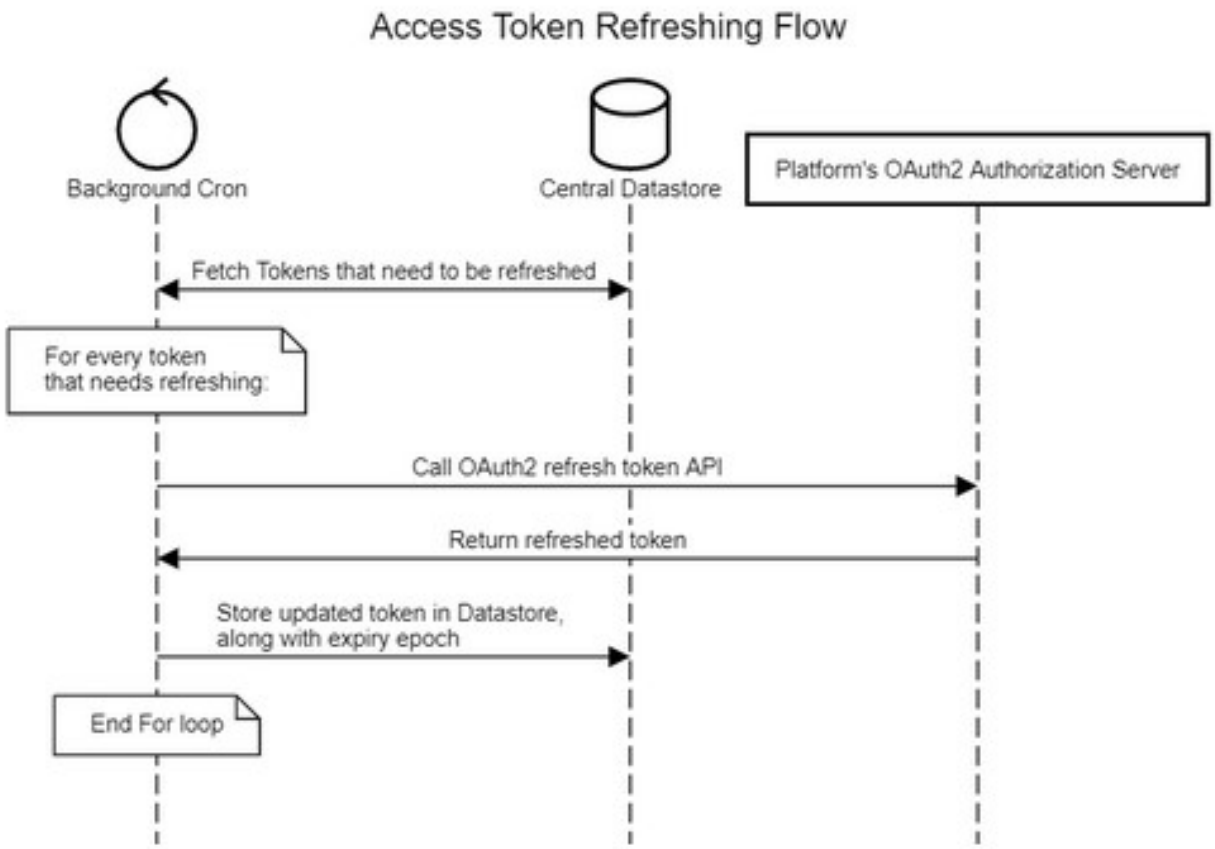

Figure 15: User Data Refreshing Module

Access Token Refreshing Module

Inputs: Platform Type, User Profile identifier (used to fetch refresh token)

Outputs: A refreshed access token, stored in the DataStore

User Feature Refreshing Module

Inputs: Platform Type, User Profile identifier (used to fetch access token)

Outputs: Refreshed transformed features for given platform, stored in the central datastore.

This module is responsible for refreshing user data from the integrated third-party platforms and for maintaining access tokens. It includes the following components:

Access Token Refreshing Module

Periodically runs every few hours to refresh access tokens basis the TTL of the corresponding platform.

User Feature Refreshing Module

Periodically runs every few hours to fetch data from an integrated platform's API, to transform that data into aggregated features and to store it in the central data-store.

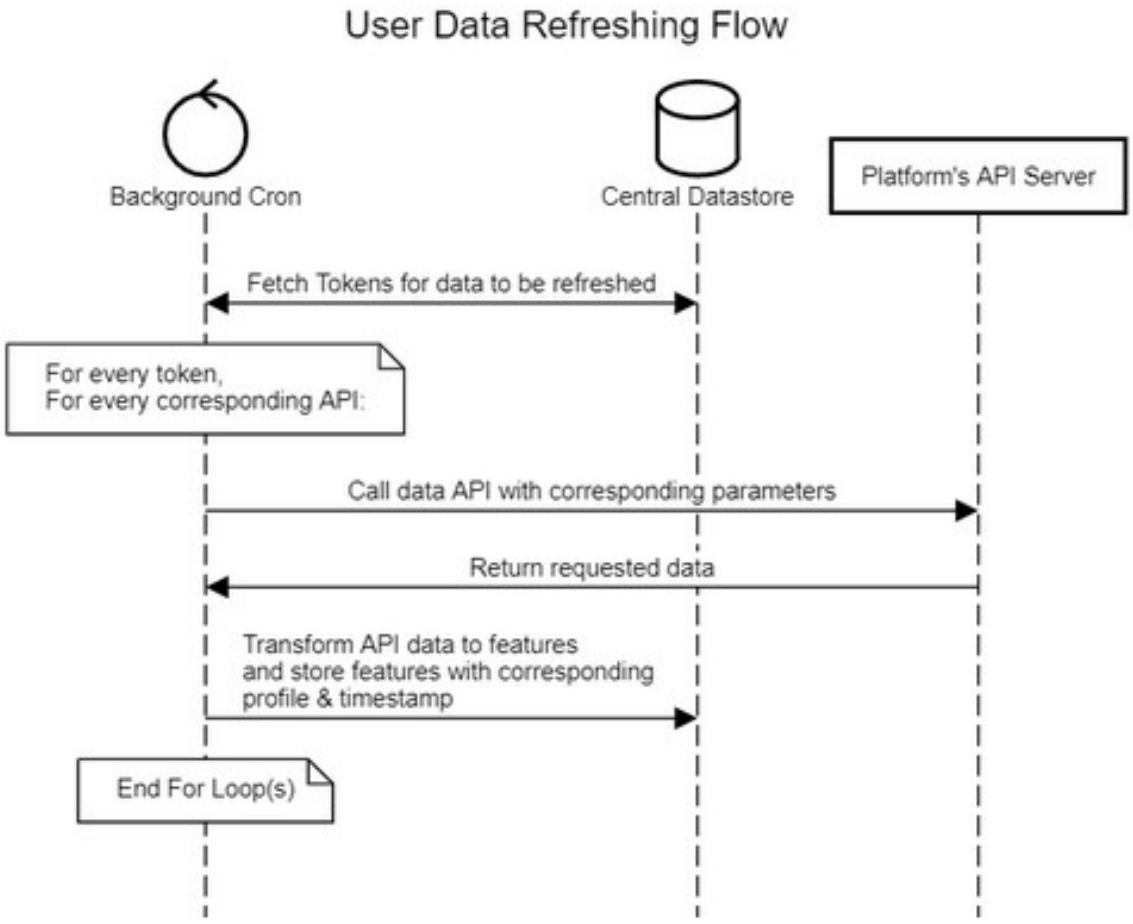

Figure 16: User Data Refreshing Flow

Figure 15 depicts the data flow of the periodic job that refreshes the user token(s) and Figure 16 depicts the data flow for the periodic task that refreshes user features in the Data Store.

5.8 Central Data-Store

Inputs: Records to be stored/updated, filters to retrieve data on *Outputs*: Status of store/update, Matched records The central database stores the following:

- User account details including name, encrypted passwords and friends added.
- Answers to the User profile questionnaire.
- Platform specific features along with timestamps for consumption by the RecNet model.
- Friend suggestions, content recommendations generated by the
- RecNet model and the user's feedback on those recommendations.
- User specific RecNet model parameters tuned through the feedback provided.

6 Feature Mapping

In this section, we have tabulated each block of Figure 8, the architecture diagram, to explain their purpose. Table 1 and Table 2 summarize each code block, which component they belong to and their usage.

Sub-feature	**Component**	**Usage Description**
Sign Up/Login	UI Component: Login Page	User Authentication using a username and password.
	Account Management Module	
Ability to link/unlink platforms	UI Component: User Profile Page	Link Spotify, Youtube and/or Reddit account to share web activity.
	Account Link/Unlink Module	

Table 1: Social Media and recommendations in one place

Sub-feature	**Component**	**Usage Description**
Friend Suggestion Feed	UI Component: Friend Recommendation Feed	The friend feed displays suggestions from the RecNet model.
	RecNet Manager module, RecNet Model Module	
Content Recommendation Feed	UI Component: Content Recommendation Feed	Recommend content based on previous activity and activity of added friends.
	RecNet Manager module, RecNet Model Module	
Real-time feedback	UI Component: Friend Recommendation Feed, Content Recommendation Feed	The feedback from the friend suggestions page or content recommendations flows through the RecNet manager to the RecNet mode, allowing us to tweak future recommendations.
	RecNet Manager module, RecNet Model Module	
Chat with friends	UI Component: Chat Interface, Chat Module	Simple chat interface where a user can choose a friend and start/continue conversation.
	Chat Module	

Table 2: User and Content Recommendations

7 Database Schema

Figure 17: Database Schema

References

[1] Da Cao et al. "Cross-Platform App Recommendation by Jointly Modeling Ratings and Texts". In: *ACM Trans. Inf. Syst.* 35.4 (July 2017). ISSN: 1046-8188. DOI: 10.1145/3017429. URL: https://doi.org/10.1145/3017429.

[2] The Anh Dang and Emmanuel Viennet. "Collaborative filtering in social networks: A community-based approach". In: *2013 International Conference on Computing, Management and Telecommunications (ComManTel)* (2013), pp. 128–133.