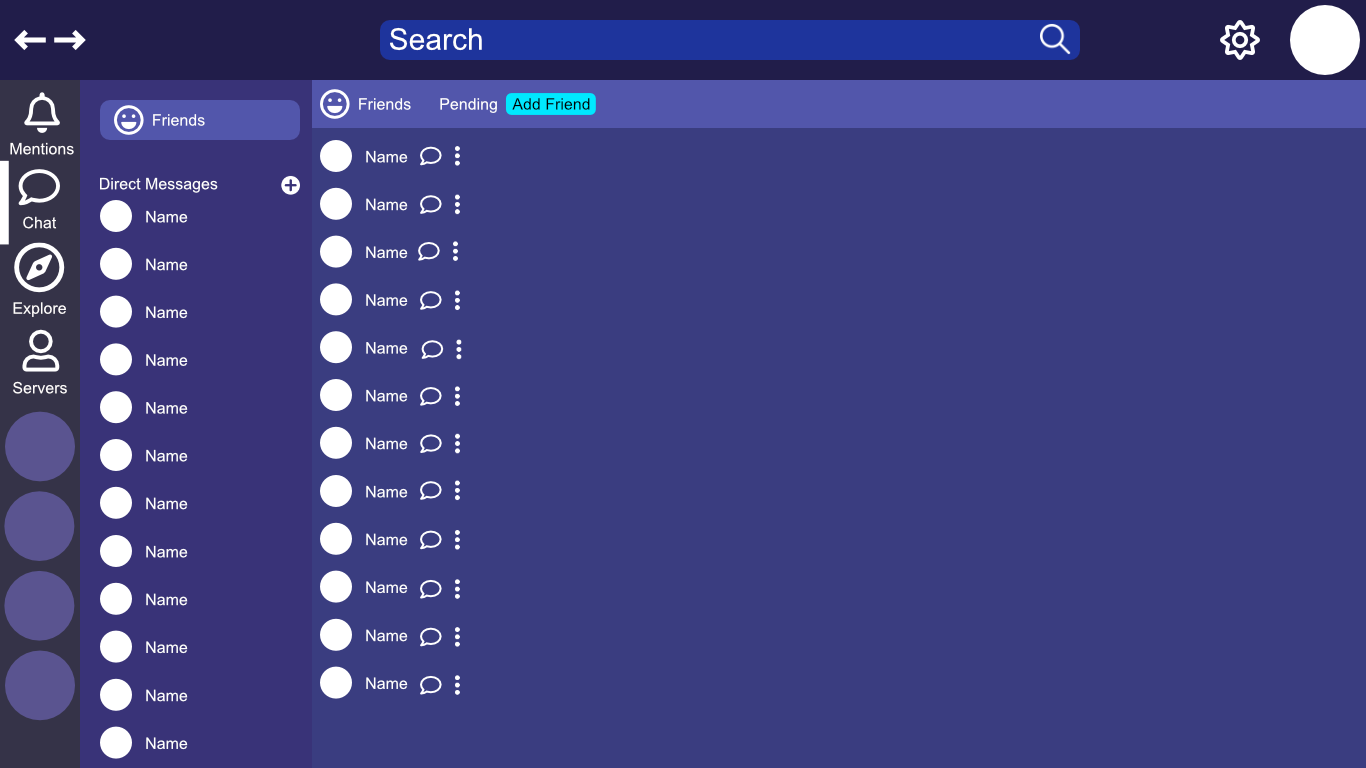
**Parlantos High Level Design**

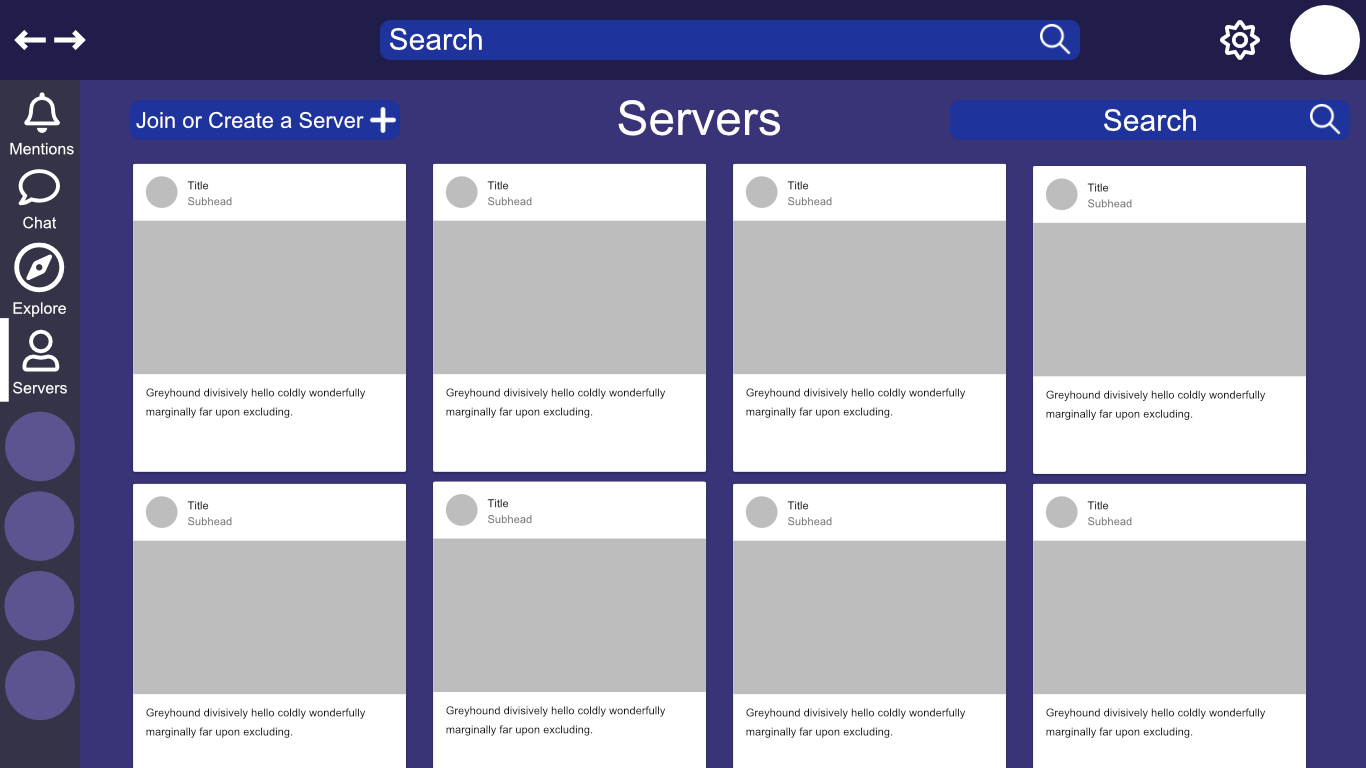
**Application Flow**



The application starts with the login page that will allow the user to login or be directed to create an account to login. 

After logging in the users will be taken to the friend’s page and that users top servers, if they exist, will be populated on the sidebar and any friends they have will populate in the friends view.

After this initial authentication process the user will have the options to access the different parts of the application that will trigger different requests to the back end to retrieve the related data.



When the user clicks on the server’s button on the sidebar they will be taken to the server search screen where they can search the servers that they are in and have the option to join or create a server.

**Other Tabs**

The mentions tab will take the user to a screen to see all of their mentions. The explore tab will take the user to a screen to filter different servers to find servers to join.

**Basic backend services flow**

**Messaging Service Flow**

Database

Client

Messaging Orchestrator

RabbitMQ

Queue

Discord Bot

Instant messaging in the app will work through a websocket connection with a server that will communicate with other web services to persist data to the database and optionally update a discord server with the same message. The orchestrator will listen to a RabbitMQ queue and communicate with the websocket connection to send data to the client. The orchestrator will only listen to the RabbitMQ queue to retrieve data and will send data to other web services.

**Storing Other Information**

Other information related to different users, server name, channels will be updated in the database and pushed to a queue. This data will be retrieved to make up the information that the client pulls down and the queue will be used to update each client in real time and will allow users to receive messages from all of the servers that they are in.

**Syncing Discord and Parlantos**

Discord and Parlantos will be synced via parallel rest endpoints that will be hit to update the information that is in each server, so each server contains the same content.