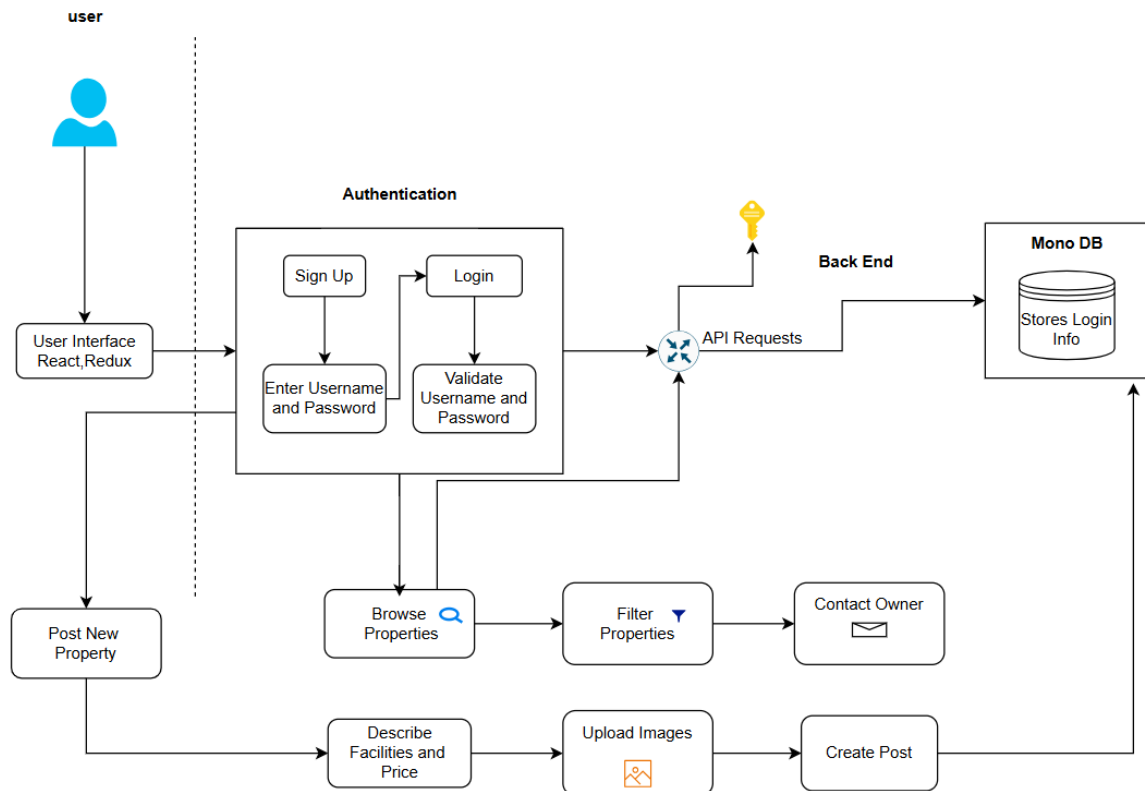


Requirement Gathering and Analysis Phase

Solution Architecture

Date	06 July 2024
Team ID	SWTID1720069738
Project Name	HOUSE RENT
Maximum Marks	4 Marks

Solution Architecture Diagram:



Business Problem

Users need a platform where they can efficiently search for, view details of, and manage properties. This involves both searching for properties based on various criteria (filters or name search) and managing their own property listings.

Solution:

Our tech solution utilizes React with Redux on the frontend for building responsive interfaces and managing application state efficiently. On the backend, Node.js , Express.js powers our API development, ensuring scalability and robust handling of CRUD operations for property management. MongoDB paired with Mongoose serves as our database solution, offering flexibility in data modeling and storage. Authentication is managed securely using JWT tokens.

Components:

1. **Authentication Module:** Manages user login and signup securely using JWT tokens for authentication and middleware for session validation.
2. **User Database (MongoDB):** MongoDB serves as the scalable and flexible database for storing user profiles and session data.
3. **User Interface (UI) for Login/Signup:** Provides a responsive and intuitive interface for users to authenticate through login and signup forms.
4. **User Profile Management:** Enables users to view and update their profile information such as name, contact details, and avatar.
5. **API Gateway:** The API component serves as the backend interface for handling CRUD operations and data exchange between the frontend and database.

Interactions:

- Validates the JWT tokens and checks user roles or permissions against access control lists (ACLs) stored in a database.
- Users interact with UI to search properties, UI communicates with backend API for CRUD operations (POST, PUT, DELETE).
- Users access and update profiles via UI, backend API handles profile data.
- Users initiate contact via UI, backend sends emails through integrated service (email).
- Centralized logging tracks system activities and errors for monitoring and troubleshooting.
- Routes incoming requests from clients to the appropriate backend services based on the endpoint and type of request (e.g., authentication, file operations).