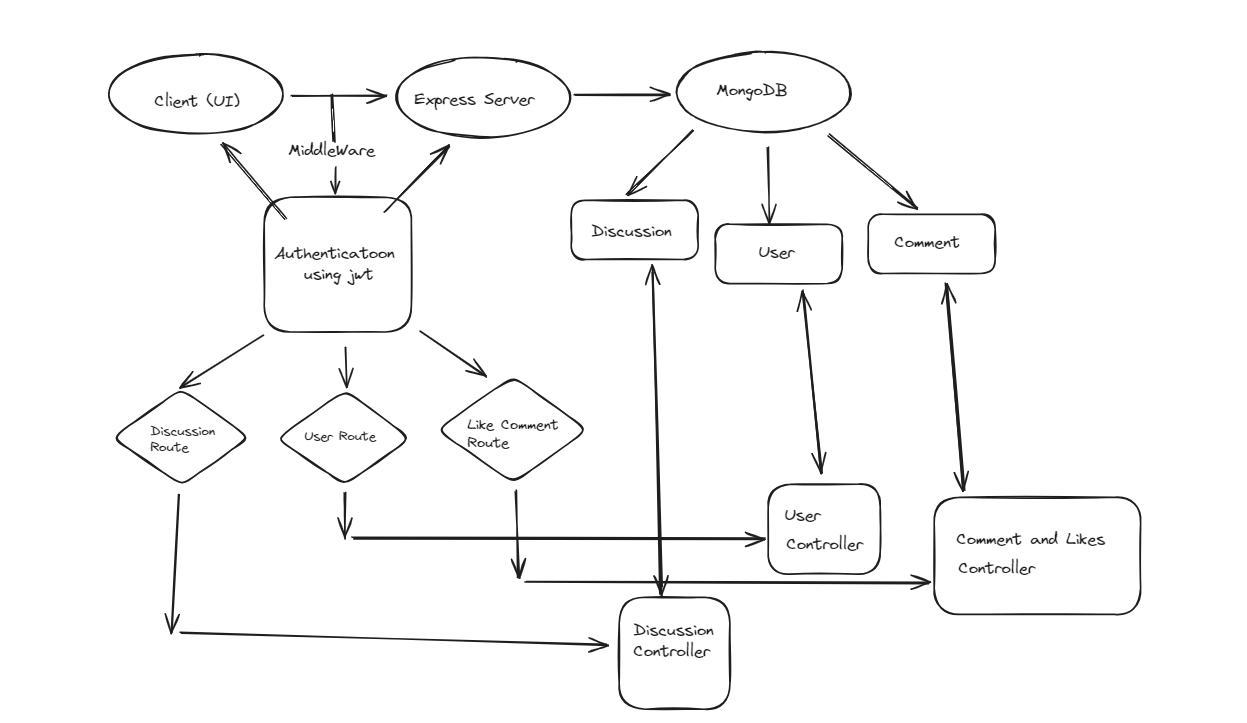
**Detailed Description of Each Component**

1. **User Service**
   * **Description**: Handles user-related operations like signup, login, and following other users.
   * **Components**:
     + **User Model**: Schema defining user attributes (name, mobile, email, password, following, followers).
     + **User Controller**: Contains methods for signup, login, and following users.
     + **User Routes**: Defines API endpoints for user-related operations.
     + **Middleware**: Authentication middleware for protecting routes.
2. **Discussion Service**
   * **Description**: Manages the creation, retrieval, updating, and deletion of discussions (posts).
   * **Components**:
     + **Discussion Model**: Schema defining discussion attributes (text, image, hashtags, user, comments, likes, viewCount).
     + **Discussion Controller**: Contains methods for creating, updating, deleting, and viewing discussions, and liking discussions.
     + **Discussion Routes**: Defines API endpoints for discussion-related operations.
3. **Comment Service**
   * **Description**: Manages comments on discussions, including creation, updating, deletion, and liking of comments.
   * **Components**:
     + **Comment Model**: Schema defining comment attributes (text, user, discussion, likes).
     + **Comment Controller**: Contains methods for adding, updating, deleting, and liking comments.
     + **Comment Routes**: Defines API endpoints for comment-related operations.
4. **Search Service**
   * **Description**: Handles searching for users and discussions using various criteria (name, tags, text).
   * **Components**:
     + **Search Controller**: Contains methods for searching users and discussions.
     + **Search Routes**: Defines API endpoints for search-related operations.
5. **Middleware**
   * **Auth Middleware**: Verifies JWT tokens to authenticate users before accessing protected routes.

**Diagram Illustrating the system architecture**