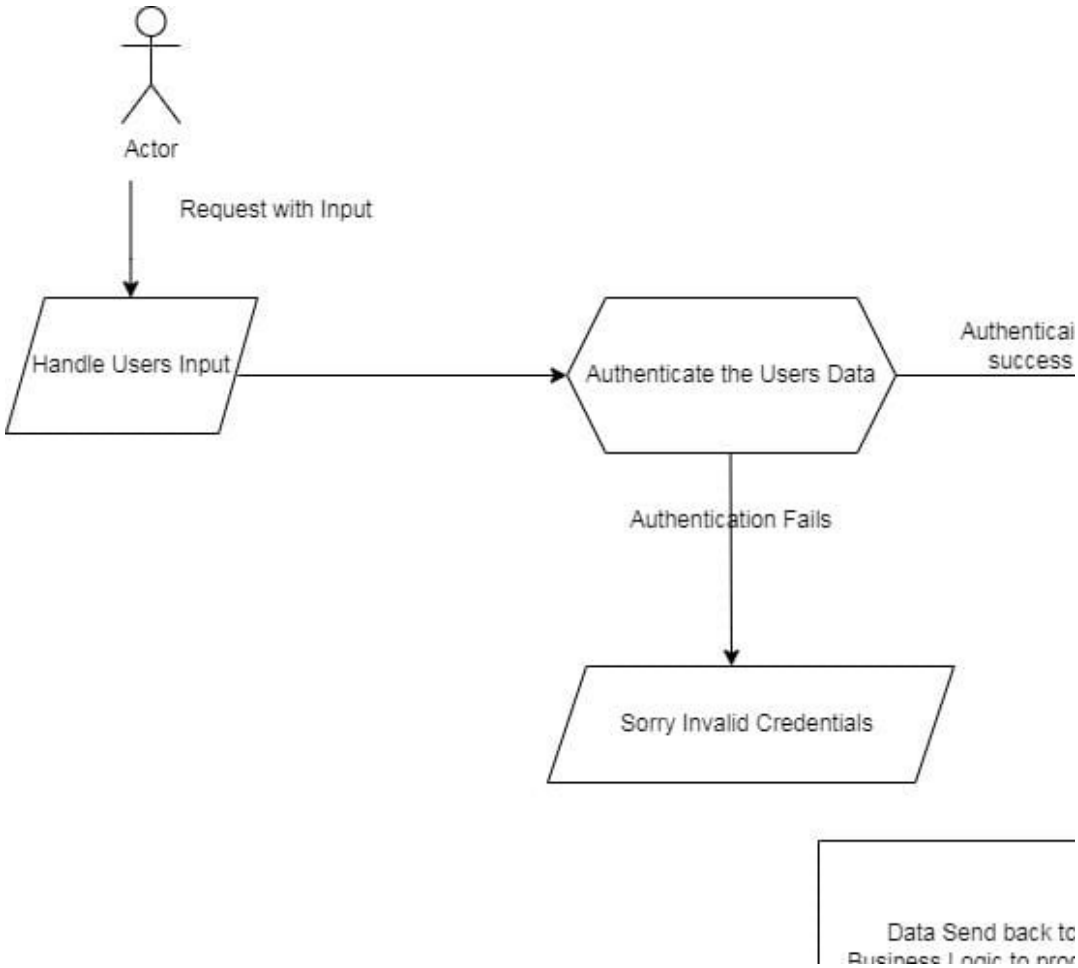
Architecture Diagram



The Car Vault App architecture is designed to efficiently handle user authentication, session management, and data operations using Node.js, Express.js, and MongoDB. The architecture ensures secure access, smooth data flow, and robust error handling. Below is a detailed description of each component in the architecture:

1. **User Interaction (Actor):**
   - The user interacts with the application by sending a request, typically including input such as login credentials or other data.
2. **User Input Handling:**
   - The application first captures and processes the user's input. Initial validations are performed to ensure that the data is correctly formatted and complete.
3. **User Authentication:**
   - The system authenticates the user by verifying the credentials against stored data. If the credentials are valid, the authentication is successful, and the process continues. If not, the user is informed of invalid credentials, and no further action is taken.

4. **JWT Token Issuance:**
    - Upon successful authentication, the system generates a JWT (JSON Web Token). This token is used to authorize the user for subsequent requests, ensuring secure communication between the client and the server.
5. **Business Logic Layer:**
    - The authenticated request is passed to the business logic layer, implemented in Node.js using the Express.js framework. This layer contains the core functionalities of the application, such as processing user requests, performing computations, and handling data flow.
6. **Database Interaction (MongoDB):**
    - The business logic layer interacts with MongoDB to perform necessary database operations. These operations may include querying data, updating records, or storing new information. MongoDB serves as the primary data store for the application.
7. **Database Operation Results:**
    - If the database operation is successful, the relevant data is returned to the business logic layer for further processing.
    - In case of a database error or exception, the error is captured, and an appropriate message is sent back to the business logic layer.
8. **Response Handling:**
    - The business logic processes the data (or error) received from the database and prepares a response for the user. The response is then sent back to the user, completing the request cycle.

Key Features:

- **Authentication & Authorization:** Secure user authentication using JWT ensures that only authorized users can access specific features.
- **Scalable Business Logic:** The business logic layer is modular and can handle various tasks, making the application scalable and maintainable.
- **Efficient Data Handling:** MongoDB provides a flexible and efficient data store, ensuring that the application can handle large amounts of data seamlessly.
- **Error Handling:** Comprehensive error handling ensures that any issues during authentication or data processing are managed gracefully, providing users with clear feedback.