TÀI LIỆU ĐẶC TẢ YÊU CẦU PHẦN MỀM (Software Requirement Specification - SRS)

Ứng dụng quản lý và sắp xếp công việc

Phiên bản 1.0.0

Thành viên nhóm 12: Hoàng Đại Nghĩa

Hà Văn Kiên

1.Giới thiệu

1.1. Mục đích

Tài liệu này đặc tả các yêu cầu phần mềm cho hệ thống "Ứng dụng Quản lý Công việc" . Tài liệu hướng đến việc giúp nhóm phát triển hiểu rõ chức năng và yêu cầu kỹ thuật để xây dựng và bảo trì hệ thống.

1.2. Phạm vi

Ứng dụng cung cấp nền tảng trực tuyến để người dùng tổ chức, cộng tác và theo dõi công việc qua hệ thống bảng (board), cột (column) và thẻ (card). Người dùng có thể tạo tài khoản, xác thực qua email, tạo board, phân quyền truy cập (public/private), mời thành viên, và chỉnh sửa hồ sơ cá nhân.

1.3. Từ điển thuật ngữ

|  |  |
| --- | --- |
| **Từ viết tắt** | **Giải nghĩa** |
| SRS | Software Requirements Specification |
| UI | User Interface (Giao diện người dùng) |
| JWT | JSON Web Token |
| API | Application Programming Interface |
| MUI | Material UI |
| CRUD | Create, Read, Update, Delete |

1.4. Tài liệu tham khảo

* React: https://react.dev
* Material UI: https://mui.com/material-ui/
* Dnd-kit: https://docs.dndkit.com/
* MongoDB
* Express.Js: https://expressjs.com/

1.5. Về ứng dụng quản lý công việc

- Dự án này nhằm mục đích phát triển một ứng dụng quản lý công việc lấy cảm hứng từ Trello, cung cấp các tính năng để cá nhân và nhóm nhỏ có thể tổ chức, theo dõi và cộng tác hiệu quả trong các nhiệm vụ và dự án. Ứng dụng sẽ tập trung vào sự đơn giản, trực quan và khả năng phản hồi trên các nền tảng web.

- Cốt lõi của ứng dụng là tính năng kéo thả vô cùng trực quan và thân thiện với người dùng, dễ sử dụng ngay cả với người mới

2. Phân tích yêu cầu

2.1. Đặt vấn đề bài toán

Trong môi trường làm việc hiện đại, nhu cầu tổ chức, phân chia công việc và cộng tác nhóm ngày càng cao. Các công cụ quản lý tác vụ đã chứng minh hiệu quả trong việc giúp nhóm làm việc linh hoạt và có tổ chức. Ứng dụng được xây dựng nhằm:

* Cho phép người dùng quản lý công việc thông qua mô hình **Board - Column - Card**
* Hỗ trợ **đăng ký tài khoản, xác thực email, đăng nhập an toàn**
* Cung cấp khả năng **cộng tác nhóm** với quyền phân chia rõ ràng
* Giao diện **thân thiện, dễ dùng, hỗ trợ dark/light/system mode**
* Hỗ trợ **quản lý hồ sơ người dùng**

Mục tiêu là xây dựng một ứng dụng web thân thiện, bảo mật, linh hoạt, có khả năng mở rộng và phù hợp với nhóm người dùng cá nhân hoặc nhóm nhỏ.

2.2. Tác nhân, người dùng

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả** |
| **Người dùng chưa đăng nhập** | Có thể đăng ký tài khoản mới, xác thực email, đăng nhập |
| **Người dùng đã đăng nhập** | Có thể sử dụng đầy đủ tính năng quản lý board, cột, thẻ, chỉnh sửa profile, v.v. |
| **Chủ sở hữu board** | Là người tạo board, có toàn quyền quản lý board, mời hoặc xóa thành viên |
| **Thành viên board** | Người được mời vào board, có thể thêm/xóa card, tương tác trong giới hạn quyền |
| **Hệ thống (System)** | Tự động gửi email xác thực, kiểm tra xác thực, bảo vệ API |

2.3. Chức năng, phân tích chức năng (theo tác nhân)

2.3.1. Người dùng chưa đăng nhập

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| Đăng ký | Nhập email, tên, mật khẩu để tạo tài khoản |
| Nhận email xác thực | Nhận liên kết xác nhận qua email |
| Xác thực tài khoản | Kích hoạt tài khoản qua liên kết |
| Đăng nhập | Nhập email và mật khẩu để đăng nhập hệ thống |

2.3.2. Người dùng đã đăng nhập

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| Tạo board | Tạo bảng làm việc mới, chọn tên và quyền (public/private) |
| Xem danh sách board | Hiển thị các board sở hữu hoặc được chia sẻ |
| Tạo cột trong board | Thêm các danh sách công việc trong board |
| Tạo thẻ trong cột | Thêm công việc cụ thể dưới dạng card |
| Xoá cột, thẻ | Quản lý và xóa các phần tử trong board |
| Mời thành viên | Nhập email để mời người khác vào board |
| Chỉnh sửa hồ sơ | Đổi tên hiển thị, avatar, mật khẩu |
| Chuyển đổi theme | Chuyển giữa dark/light/system mode |
| Đăng xuất | Thoát phiên làm việc hiện tại |

2.2.3. Chủ sở hữu board

|  |  |
| --- | --- |
| **Chức năng bổ sung** | **Mô tả** |
| Quản lý thành viên | Mời hoặc xoá người dùng khỏi board |
| Đổi quyền truy cập | Đổi giữa public/private |
| Xoá board | Xoá vĩnh viễn board (nếu cần) |

2.3.4. Thành viên board

|  |  |
| --- | --- |
| **Chức năng giới hạn** | **Mô tả** |
| Tương tác với card | Tạo, sửa, xoá card nếu được cấp quyền |
| Tạo cột (nếu cho phép) | Có thể tạo cột nếu chủ sở hữu cho phép |
| Xem board | Chỉ xem nội dung nếu board ở chế độ public hoặc đã được mời |

2.3.5. Hệ thống

|  |  |
| --- | --- |
| **Chức năng hệ thống** | **Mô tả** |
| Gửi email xác thực | Tự động gửi email chứa liên kết xác minh |
| Xác thực phiên JWT | Kiểm tra token để cho phép truy cập API |
| Bảo vệ route API | Chặn truy cập trái phép đến các endpoint |

3. Đặc tả và thiết kế

3.1. Usecase và mô tả

- Usecase chính:

* + Đăng ký tài khoản
  + Xác thực email (Gặp lỗi do không thể đăng kí được tài khoản Brevo cung cấp dịch vụ gửi mail tích hợp)
  + Đăng nhập
  + Tạo board
  + Mời thành viên ( do phần xác thực email có lỗi nên phần này cũng chưa thể hoạt đông)
  + Tạo column/card
  + Xoá column/card
  + Di chuyển các column với nhau
  + Di chuyển các card qua các column khác nhau
  + Quản lý profile
  + Chuyển đổi dark/light mode

Usecase 1:

|  |  |
| --- | --- |
| **Thành phần** | **Nội dung** |
| **Tên Use Case** | Đăng ký tài khoản |
| **Mã Use Case** | UC01 |
| **Tác nhân** | Người dùng nhấn vào nút “Đăng ký” |
| **Mô tả** | Cho phép người dùng đăng ký vào hệ thống bằng email và mật khẩu |
| **Tiền điều kiện** | Tài khoản email phải chính xác và có tồn tại |
| **Hậu điều kiện** | Hệ thống xác thực thành công và gửi một mail cho bạn để xác thực và active tài khoản |
| **Luồng chính** | 1. Người dùng nhập email và tạo mật khẩu sau đó xác nhận mật khẩu  2. Nhấn nút “Đăng ký”  3. Hệ thống lưu thông tin tạm thời và gửi email xác thực (qua verifyToken).  4. Hệ thống kiểm tra thông tin email có tồn tại không và gửi đến một mã yêu cầu người dùng ấn vào link đó để active tài khoản  5. Sau khi user nhấn vào link hệ thống cập nhật trường isActive = true. |
| **Luồng ngoại lệ** | 1. email không tốn tại hoặc bị nhập sai --> Hệ thống yêu cầu nhập lại  2. xác nhận mẩu khẩu không trung với mật khẩu tạo ở trên --> hệ thống yêu cầu nhập lại |

Usecase 2:

|  |  |
| --- | --- |
| **Thành phần** | **Nội dung** |
| **Tên Use Case** | Đăng nhập |
| **Mã Use Case** | UC02 |
| **Tác nhân** | Người dùng nhấn vào nút “đăng nhập” |
| **Mô tả** | Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu |
| **Tiền điều kiện** | Người dùng đã đăng ký tài khoản |
| **Hậu điều kiện** | Hệ thống xác thực thành công và chuyển đến trang board |
| **Luồng chính** | 1. Người dùng nhập email và mật khẩu --> API/Login  2. Nhấn nút “Đăng nhập”  3. Server sinh AccessToken, RefreshToken để xác thực tài khoản  4. Nếu đúng HttpOnly Cookie / LocalStorage --> đăng nhập thành công |
| **Luồng ngoại lệ** | Email hoặc mật khẩu sai → hiển thị thông báo lỗi, yêu cầu nhập lại |

Usecase 3:

|  |  |
| --- | --- |
| **Thành phần** | **Nội dung** |
| **Tên Use Case** | Tạo board |
| **Mã Use Case** | UC03 |
| **Tác nhân** | Người dùng đã đăng nhập |
| **Mô tả** | Cho phép người dùng tạo 1 bảng |
| **Tiền điều kiện** | Người dùng đã đăng nhập thành công và đến phần boards |
| **Hậu điều kiện** | Một bảng mới được tạo thành công, lưu vào hệ thống, người tạo là admin của bảng |
| **Luồng chính** | 1. Người dùng nhấn vào ô “Create a new board”  2.Hệ thống yêu cầu bạn nhập title(tên bảng) và description(mô tả), chọn quyền truy cập của bảng là public hay private  3. Nhấn Create để tạo bảng  4. Hệ thống tạo bản ghi trong Board, thêm user vào ownerIds |
| **Luồng ngoại lệ** | 1. Nếu nhập dưới 3 kí tự --> hệ thống báo không hợp lê  2. Nếu không nhập title, description hệ thống sẽ báo lỗi yêu cầu phải điền đầy đủ |

Usecase 4:

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| **Tên Use Case** | Tạo column (Cột) |
| **Mã Use Case** | UC04 |
| **Tác nhân (Actor)** | Người dùng đã đăng nhập |
| **Mô tả ngắn gọn** | Cho phép người dùng tạo một cột mới trong bảng để phân loại công việc |
| **Tiền điều kiện** | Người dùng đã đăng nhập và đang ở trong một bảng hợp lệ |
| **Hậu điều kiện** | Một cột mới được thêm vào bảng hiện tại, hiển thị trong giao diện |
| **Luồng chính** | 1. Người dùng nhấn vào ô “Add new column”  2.Nhập tên column  3.Nhấn “Add”  4. Hệ thống tạo column --> gửi API --> lưu vào Column(gắn boardID) |
| **Luồng phụ / Ngoại lệ** | 1. Tên clumn trống --> Hệ thống báo lỗi và yêu cầu phải nhập vào ô chứa |

Usecasr 5:

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| **Tên Use Case** | Tạo card (thẻ) |
| **Mã Use Case** | UC05 |
| **Tác nhân (Actor)** | Người dùng đã đăng nhập |
| **Mô tả ngắn gọn** | Cho phép người dùng tạo thẻ trong một cột để biểu diễn một tác vụ cụ thể |
| **Tiền điều kiện** | Người dùng đã đăng nhập và đã chọn một cột cụ thể |
| **Hậu điều kiện** | Một thẻ mới được tạo và hiển thị trong cột được chọn |
| **Luồng chính** | 1. Người dùng nhấn “Add new card” trong cột  2. Nhập tên thẻ  3. Nhấn “Add”  4. Hệ thống tạo card --> gửi API --> lư vào Card(gắn columnID) |
| **Luồng phụ / Ngoại lệ** | 1.Tên card trống → Hệ thống báo lỗi và yêu cầu nhập vào ô chứa |

Usecase 6:

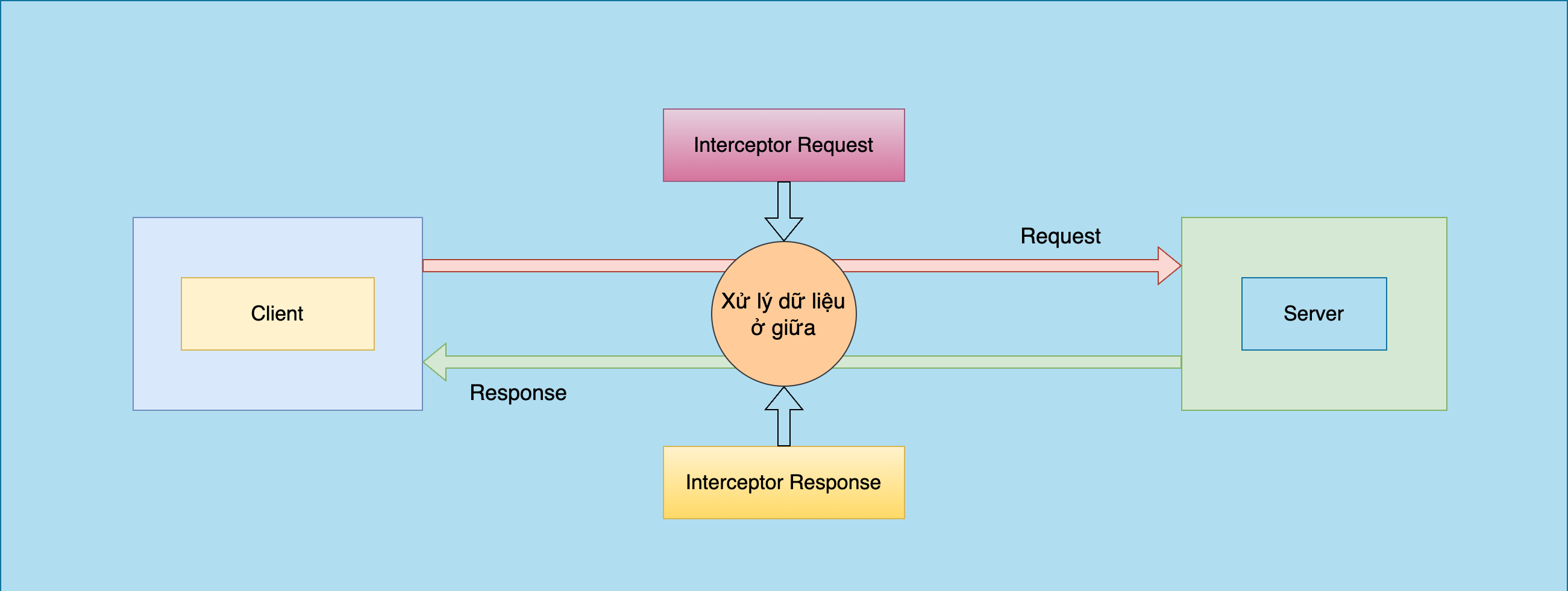
|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| **Tên Use Case** | Kéo thả column (cột) |
| **Mã Use Case** | UC06 |
| **Tác nhân (Actor)** | Người dùng đã đăng nhập |
| **Mô tả ngắn gọn** | Cho phép người dùng thay đổi vị trí các cột trong bảng bằng cách kéo thả |
| **Tiền điều kiện** | Người dùng đã đăng nhập và đang ở trong bảng có ít nhất 2 cột |
| **Hậu điều kiện** | Thứ tự các cột được cập nhật và hiển thị đúng vị trí mới trên giao diện |
| **Luồng chính** | 1. Người dùng nhấn giữ một cột bất kỳ  2. Kéo cột đến vị trí mong muốn  3. Thả chuột để hoàn tất thao tác  4. Hệ thống cập nhật thứ tự và lưu vào cơ sở dữ liệu |
| **Luồng phụ / Ngoại lệ** | 1. Người dùng kéo nhưng không thả đúng vị trí --> cột trở về vị trí cũ |

Usecase 7:

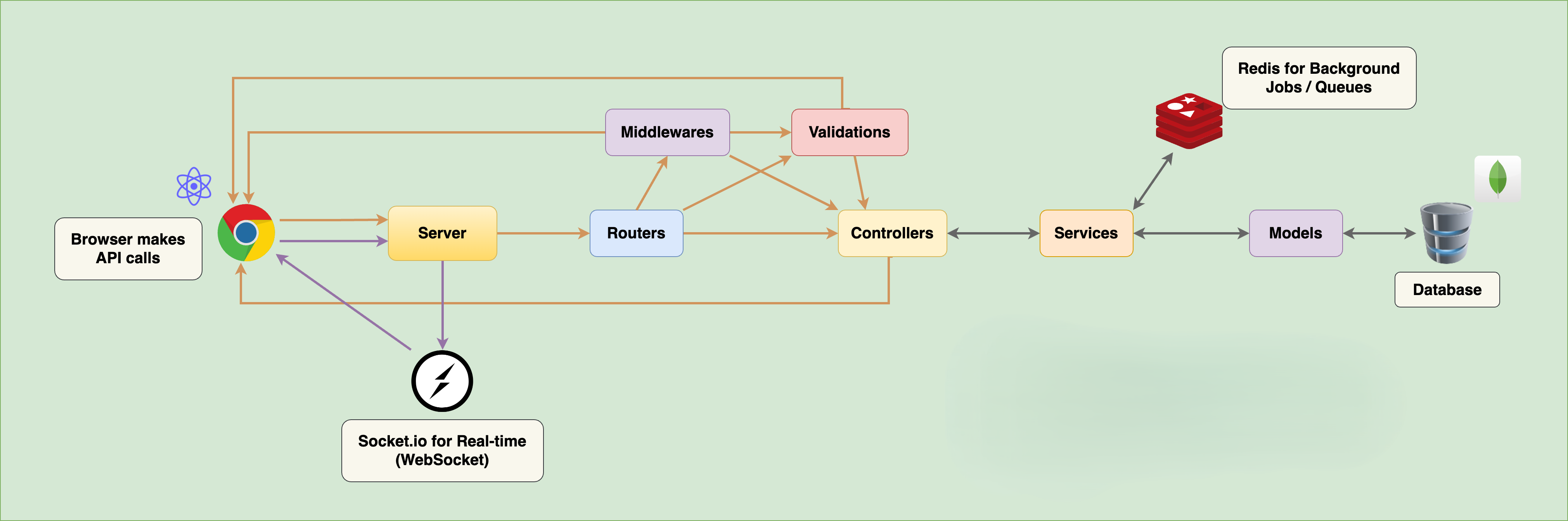
|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| **Tên Use Case** | Kéo thả card (thẻ) |
| **Mã Use Case** | UC07 |
| **Tác nhân (Actor)** | Người dùng đã đăng nhập |
| **Mô tả ngắn gọn** | Cho phép người dùng di chuyển thẻ giữa các cột hoặc thay đổi thứ tự trong cùng một cột |
| **Tiền điều kiện** | Người dùng đã đăng nhập và bảng có ít nhất một cột với một thẻ |
| **Hậu điều kiện** | Vị trí của thẻ được thay đổi, hiển thị đúng trên giao diện và lưu vào hệ thống |
| **Luồng chính** | 1. Người dùng nhấn giữ một thẻ  2. Kéo thẻ đến vị trí mới (trong cùng cột hoặc cột khác)  3. Thả chuột  4. Hệ thống cập nhật vị trí thẻ trong cơ sở dữ liệu |
| **Luồng phụ / Ngoại lệ** | 1.Người dùng kéo ra ngoài khu vực hợp lệ --> thẻ trở lại vị trí cũ |

3.2.Sơ đồ tham khảo cho mục này:

-Sơ đồ Axios Interceptors: Dùng cho phần mô tả Use Case đăng nhập, xử lý request/response



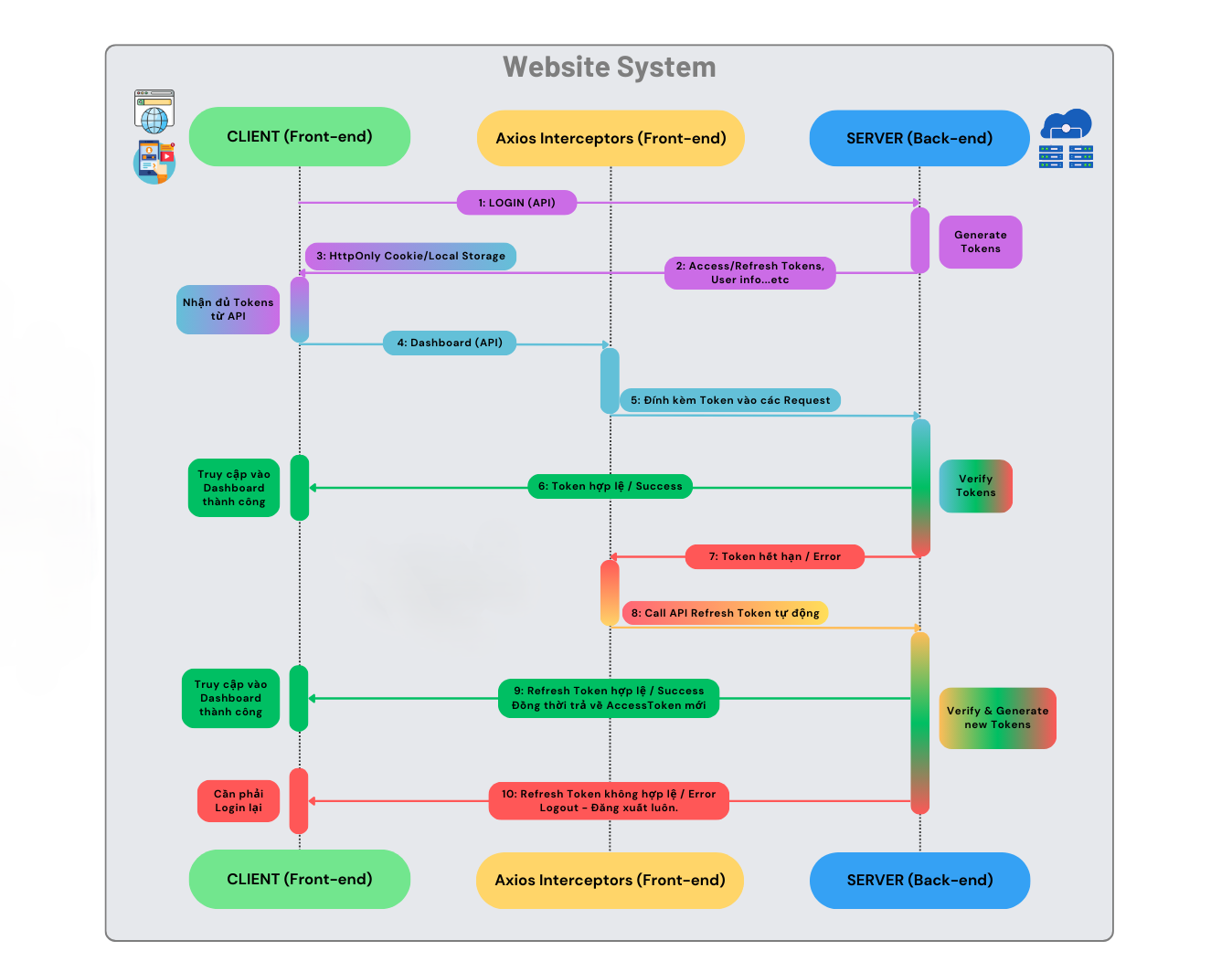
-Sơ đồ BE code flow: Kiến trúc hệ thống backend, data flow (đã rõ các khối Router, Middleware, Controller, Service, Model)



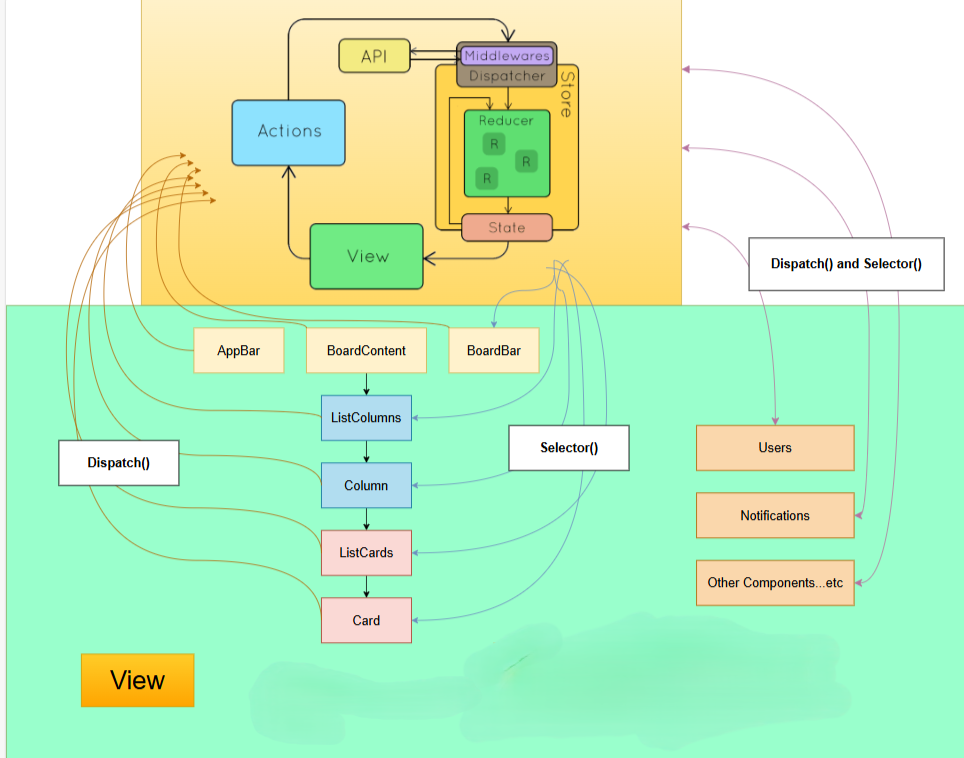
-Sơ đồ Data Scheme: CSDL (đã có đầy đủ các collections: User, Board, Column, Card, Invitation)



-Sơ đồ JWT-Jsonwebtoken-Axios-Interceptors:



-Sơ đồ Redux-lifecycle:

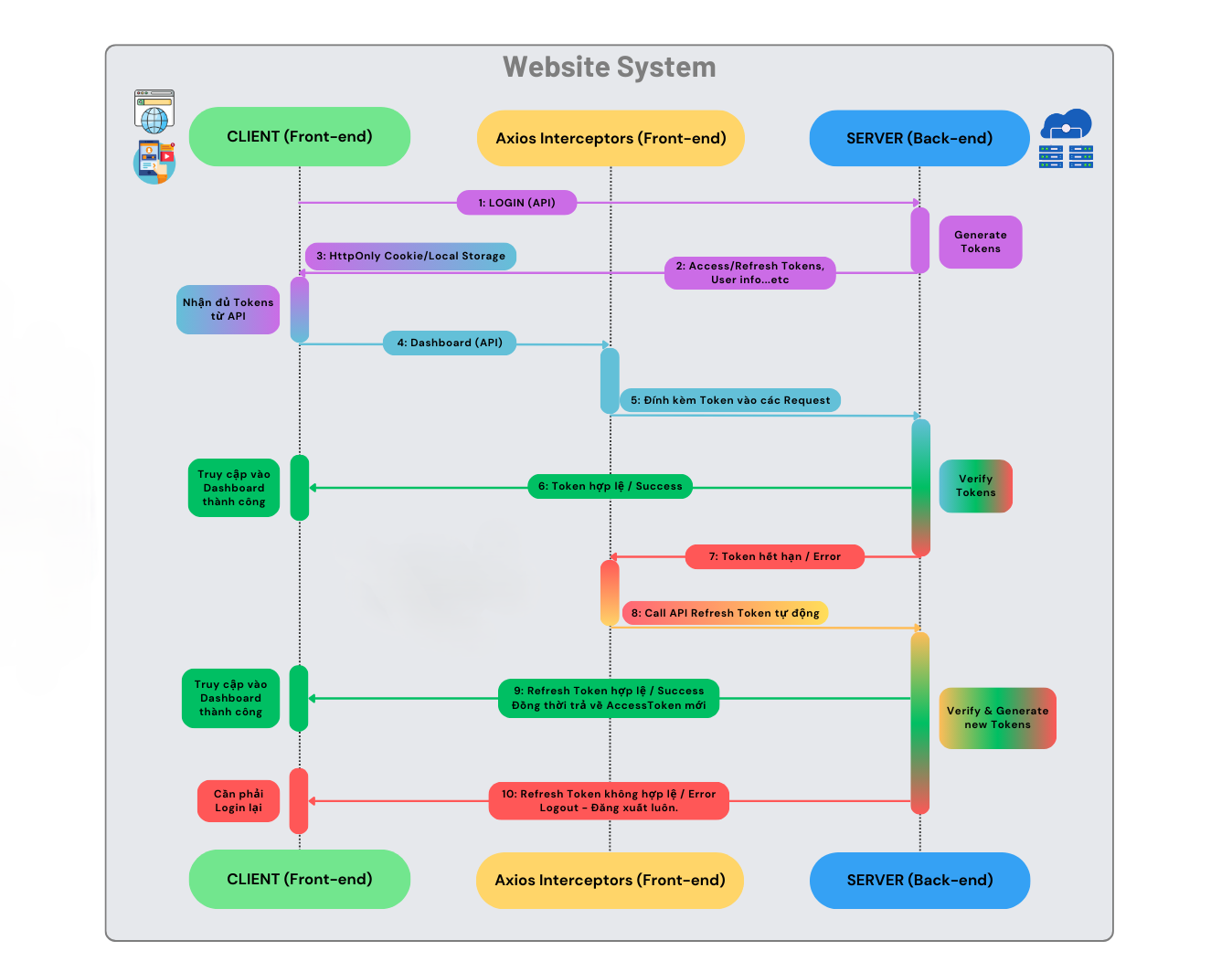


3.3. Flow – Sequence Diagram: Đăng nhập & Xác thực JWT

3.3.1. Mục đích

- Trình bày chi tiết quy trình xác thực người dùng khi đăng nhập, sử dụng cơ chế AccessToken, RefreshToken kết hợp với Axios Interceptors.

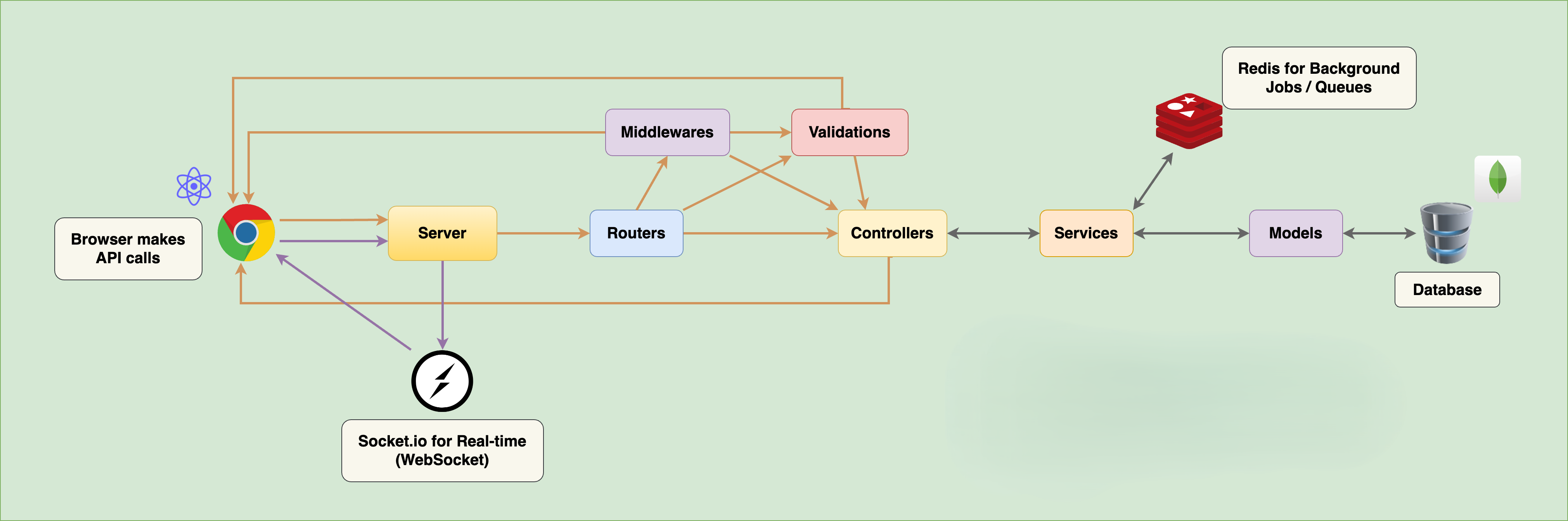
3.3.2. Luồng chính

1. Người dùng nhập thông tin đăng nhập.
2. Gửi yêu cầu POST /login tới server.
3. Server xác thực thông tin, sinh AccessToken và RefreshToken.
4. Gửi token về client (qua cookie hoặc localStorage).
5. Khi gọi API sau đó, client tự động gắn AccessToken vào header.
6. Nếu token hợp lệ → server phản hồi dữ liệu.
7. Nếu token hết hạn → interceptor gọi API /refresh-token.
8. Server xác thực RefreshToken, gửi lại AccessToken mới.
9. Nếu RefreshToken không hợp lệ → buộc đăng xuất.  
   

3.4. Data Flow & Cơ sở dữ liệu (MongoDB)

3.4.1. Data Flow Diagram – Backend Architecture  
 - Luồng xử lý chính:

1. Client (React App) gửi request qua Axios.
2. Server Node.js nhận request → qua Routers.
3. Router gọi Controllers, đi qua Middlewares và Validations.
4. Controller gọi Services.
5. Services tương tác với Models (Mongoose).
6. Models truy xuất/ghi dữ liệu từ MongoDB.
7. Với các công việc nền như gửi email → sử dụng Redis Queue.

Sơ đồ kiến trúc backend:  


3.4.2. MongoDB Schema

|  |  |  |
| --- | --- | --- |
| **Collection** | **Các trường chính** | **Ghi chú** |
| **User** | email, password, username, isActive, verifyToken, avatar, role | Xác thực, phân quyền |
| **Board** | title, type, ownerIds, memberIds, columnOrderIds | Phân quyền sở hữu |
| **Column** | title, boardId, cardOrderIds | Gắn vào từng board |
| **Card** | title, columnId, boardId, description, attachments, members, comments | Có hỗ trợ file & comment |
| **Invitation** | inviterId, invitedId, boardInvitation, status | Gửi lời mời thành viên |

Tất cả collections đều có \_destroy, createdAt, updatedAt hỗ trợ logic "xoá mềm" và audit.

4. UI Flow - Luồn giao diện người dùng

4.1. Các màn hình chính:

|  |  |
| --- | --- |
| **Giao diện** | **Chức năng** |
| Trang đăng nhập / đăng ký | Nhập thông tin tài khoản |
| Giao diện xác thực email | Nhấn liên kết từ mail |
| Dashboard | Hiển thị danh sách board |
| Giao diện Board | Hiển thị column, card (drag & drop) |
| Modal tạo board | Tạo board mới (public/private) |
| Modal mời thành viên | Nhập email để mời người dùng |
| Trang Profile | Đổi avatar, đổi mật khẩu, tên |
| Giao diện chế độ sáng/tối | Chuyển dark/light/system mode |

4.2. Mô tả UI Flow (từ sơ đồ Redux-lifecycle)

Luồn điển hình của một thao tác:

View (BoardPage)

↓

Dispatch(Action) → API gọi BE

↓

Middleware xử lý → gọi Reducer cập nhật Store

↓

Selector() truy xuất dữ liệu

↓

Cập nhật lại View

- Cấu trúc component chính:

* AppBar – chứa đổi mode + profile
* BoardContent – gồm ListColumns
  + Column – chứa nhiều Card
    - Card – có popup hiển thị chi tiết

Tất cả dùng Redux để điều phối trạng thái.

5. Kết quả và đánh giá

5.1. Kết quả đạt được

- Sau quá trình phát triển và thử nghiệm, hệ thống quản lý công việc do nhóm phát triển đã hoàn thiện với các chức năng chính sau:

#### **Tính năng chức năng:**

* **Xác thực tài khoản qua email:** Hệ thống yêu cầu người dùng xác thực email để kích hoạt tài khoản sau khi đăng ký.
* **Quản lý board:** Người dùng có thể tạo board mới, chỉnh sửa, phân quyền private/public, và mời thành viên.
* **Quản lý cột và thẻ:** Cho phép tạo, xoá, sắp xếp cột và thẻ trong từng board. Có hỗ trợ kéo thả và lưu trữ trạng thái.
* **Quản lý người dùng:** Người dùng có thể chỉnh sửa thông tin cá nhân như avatar, tên hiển thị, và mật khẩu.
* **Chế độ hiển thị:** Hệ thống hỗ trợ chuyển đổi giữa chế độ sáng, tối và tự động (theo hệ điều hành).
* **Hệ thống phân quyền cơ bản:** Người tạo board có quyền mời/xoá thành viên và chỉnh sửa nội dung.

#### **Về kỹ thuật:**

* **Frontend:** Sử dụng React kết hợp Material UI để đảm bảo giao diện hiện đại, responsive.
* **Backend:** Node.js tổ chức theo mô hình MVC rõ ràng, sử dụng MongoDB làm cơ sở dữ liệu chính.
* **Authentication:** Xác thực bảo mật bằng JWT và sử dụng AccessToken + RefreshToken để nâng cao trải nghiệm và bảo mật.
* **Redux:** Quản lý trạng thái hiệu quả với Redux Toolkit và lifecycle chuẩn.
* **Socket (dự kiến):** Ứng dụng có cấu trúc sẵn sàng tích hợp Socket.io cho tính năng realtime.

### **5.2. Đánh giá**

#### **Ưu điểm:**

* **Tính năng cốt lõi hoàn thiện:** Hệ thống đã có hầu hết các chức năng quan trọng cho một ứng dụng quản lý công việc.
* **Kiến trúc rõ ràng, mở rộng dễ dàng:** Backend phân lớp hợp lý, dễ nâng cấp và bảo trì.
* **UX/UI thân thiện:** Giao diện đơn giản, dễ sử dụng, màu sắc hiện đại, hỗ trợ đa chế độ sáng/tối.
* **Bảo mật:** Việc áp dụng JWT + refresh token giúp tăng tính bảo mật cho phiên làm việc.

#### **Hạn chế:**

* **Chưa có realtime:** Chức năng kéo thả hoạt động tốt, nhưng thay đổi chưa được đồng bộ thời gian thực giữa nhiều người dùng.
* **Thiếu tính năng nâng cao:** Chưa hỗ trợ tính năng nhắc việc, deadline, đính kèm file lớn, hoặc phân quyền chi tiết trong từng board.
* **Chưa tối ưu hiệu suất lớn:** Hiện tại hệ thống hoạt động tốt với số lượng người dùng vừa phải, chưa thử nghiệm hiệu năng cao.

### **Kết luận**

Ứng dụng quản lý công việc do nhóm xây dựng đã hoàn thành tốt mục tiêu ban đầu: xây dựng hệ thống có chức năng tương tự Trello với trải nghiệm người dùng tốt và kỹ thuật hiện đại. Tuy chưa đầy đủ toàn bộ tính năng của sản phẩm thương mại, ứng dụng vẫn là nền tảng vững chắc để tiếp tục phát triển trong tương lai.