**Document Control Project**

1. **Create Initial Interface dummy**
2. **Create Navigation bar components:**

* **Home**
* **About**
* **Feature**
* **Login**
* **Logout**
* **Search**

1. **Home: Just Allay Pharmaceuticals**
2. **About: Description About interface users**
3. **Feature: Steps of feature how to use document control**
4. **Login: User Login with credentials**
5. **Logout: Logout from interface**
6. **Successful Login:** 
   * 1. **Username**
     2. **Password**
     3. **Stay singed in**
     4. **Login Button**
     5. **Cancel Button**
     6. **Signup Button**
     7. **Forgot Password or Reset Password**
7. **Signup Card:**
   * 1. **Email**
     2. **Password**
     3. **Confirm Password**
     4. **Submit Button**
     5. **Cancel Button**
8. **Find Document Card:**
   * 1. **Filter by**
     2. **Document Control No**
     3. **Department**
     4. **Submit**
     5. **Cancel**
9. **Create New Document**
   * 1. **Document Control No**
     2. **Department**
     3. **Created By**
     4. **Checked By**
     5. **Approved by**
     6. **Submit**
     7. **Cancel**

**Architecture**

Diagram

Description automatically generated

The communication between the front-end and back-end will be implemented using REST API. On the front-end side, we will have an HTTP Client and the back-end will be handling those HTTP requests using Spring REST Controller. The angular application on the front-end side will create a fully-functional user interface to manage posts (adding, edit, search). All data will be saved in the MySQL database, integrated with the application using Spring Data JPA.

**1. Front End Development**: Front end interface (Document Control) communicates middle wire as following:

* Send Request to read data
* Send Request to Create data
* Put Request to update Data
* Delete request from database

**Framework:**

* Angular 12
* TypeScript
* JavaScript
* HTML5
* CSS
* Bootstrap

Components Created:

* Home
* About
* Features
* Login
* Logout
* Sign up
* Add New Document
* Find Document
* Error Handling
* File Management
* Admin
* Models:
  + - User
* Service
  + - User Authentication
* App Routing
* App Component
* App Module
* Assets

**2. Middle Wire Architecture:** Communicate both Front End and Back End for following:

* HTTP request: GET, PUT, POST, DELETE
* Firmware use: Spring Boot, MVC, Microservices
* Technology :
  + - Java 8
    - Spring Boot
    - Maven
    - MySql

**3. Back End:** Database Layer for storage replies request to middle wire

**Firmware:** MySQL, SQL, Oracle, Mongo DB, AWS Cloud