

Detailed Design Document - Hog Ridaahs!

- When the user first enters the website, they are greeted with the root or “home” page. It serves as the central location from which the user can search for other pages, papers, and posters.
- It has a search bar that the user can use to navigate the site, or request data from the server (papers, posters, etc).
- Subpages, which branch off from the root page, include the following:
 - Student portfolio page: a place where a specific student can review their published papers or posters
 - Faculty/editor page: a place for editors and members of faculty to perform necessary functions, such as reviewing papers or moderating user content.
 - Journal directory page: a place for the general public to review the published works of students and faculty.
- Currently, we are utilizing a single database design for the website. However, as the project expands, it may become better to break up the user information, pdf files, and posters into separate databases, allowing for simplicity and security.
- We are using MongoDB for the database and it is set up on Atlas. Currently, the data is stored in BSON format, which is MongoDB’s unique way of storing data. When interacting with the Spring Boot code, this data is converted to “Plain Old Java Objects” (POJOs), before it can be converted into JSON and served to the client.
- The backend is built using Spring Boot, which provides a structured and efficient way to handle HTTP requests, interact with the database, and manage business logic.
- Spring Boot will provide code for a REST API that will facilitate communication between the client and server. The data itself will be, when possible, sent using the JSON format, along with HTTP status codes.
- Within the files related to application server functions, there are several different types of naming conventions used, each of which is dependent on which the code within the file is supposed to do. The major ones are: Repositories, Service, and Controllers. The “repository” file serves as a connection between the Spring Boot code and the database. The service class contains functions related to the specific type of data that is being handled. Finally, controllers handle the reception and processing of HTTP requests.
 - For example, there is code that controls the processing of user data that is stored in MongoDB. Therefore, we have a User Repository class, that helps connect the code to the actual database. There also exists a User Service class, which contains functions related to managing user data. Finally, there is a User Controller class, which deals with HTTP requests, such as getting information from the database and sending it to the client.

- The frontend is built with a central App.jsx file containing the structural code for the website, serving as the foundation for the entire thing. It is paired with a central App.css file which contains the universal styling of the website.
- To achieve unique functions and styling for each subpage of the website, we have .jsx and .css files unique to each desired subpage within a 'components' folder in React (e.g., 'LoginPage.jsx + LoginPage.css, HomePage.jsx + HomePage.css, etc.)

Additional Notes

- New developers should review this document before making structural changes.
- Ensure all changes align with the documented design to maintain consistency.
- Future discussions should be recorded here for transparency and reference.