# Developer Manual

## CRAISELION : Handong Team Meeting Archiving and Exchange Web Platform

Version: 1.0
Date: 2024-06-08
Client: Student
Confidential: No
developer: Seokjae Ma, Donggyu Kim, Sechang Jang, Junhyeok Choi, Minseo Lee

### Team

| Name | Email | ID | Role |
|---|---|---|---|
| Seokjae Ma | 21800239@handong.ac.kr | 21800239 | Project Manager |
| Donggyu Kim | 22000063@handong.ac.kr | 22000063 | Scrum Master |
| Sechang Jang | 21900628@handong.ac.kr | 21900628 | Documenter |
| Junhyeok Choi | 21900764@handong.ac.kr | 21900764 | Developer |
| MinSeo Lee | 22100503@handong.ac.kr | 22100503 | Developer |

### Version

| Version | Date | Description |
|---|---|---|
| 1.0 | | |

# Table of Contents

# 1 Introduction

## 1.1 Project Background

Handong University hosts a variety of Residential Colleges (RCs), within which numerous team meetings are regularly held. These meetings play a crucial role in fostering community formation and interaction among students. However, the current system faces several significant challenges.

First, there are difficulties in exchanging information between teams within the same RC, and even more so between different RCs. This lack of interaction within and between RCs hinders the development of a community-centered culture at Handong, which the university aims to promote.

Second, there is a scarcity of guidance provided to team leaders, leading to challenges in leadership and team management. This situation diminishes the efficiency and effectiveness of team activities, negatively affecting student engagement and enthusiasm.

Third, the preservation of materials and information generated during team meetings throughout the year is not adequately addressed. Due to the absence of a proper archiving system, important documents are often lost, posing significant challenges to long-term project management and the reuse of materials.

To resolve these issues, we propose a web platform that facilitates communication and exchange among teams or RCs and preserves information and materials related to team meetings, thereby aiding in the development of the entire community at Handong University. This platform will serve as an essential resource for the student support team and the students (team executives), who are the main stakeholders, in enhancing the team culture and leadership development within Handong University.

# 2. Frontend build method

This project was bootstrapped with Create React App.

## Available Scripts

In the project directory, you can run:

### `npm install`

Need to download all dependencies and required ones.

### `npm start`

Runs the app in the development mode.
Open http://localhost:3000 to view it in your browser.

The page will reload when you make changes.
You may also see any lint errors in the console.

### `npm test`

Launches the test runner in the interactive watch mode.
See the section about running tests for more information.

### `npm run build`

Builds the app for production to the `build` folder.
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.
Your app is ready to be deployed!

See the section about deployment for more information.

### `npm run eject`

**Note: this is a one-way operation. Once you `eject`, you can't go back!**

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Learn More

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

### Code Splitting

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

### Analyzing the Bundle Size

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

### Making a Progressive Web App

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

### Advanced Configuration

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

### Deployment

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

### `npm run build` fails to minify

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify

# 3. Backend build method

Spring Boot is a framework built on top of the Spring Framework that simplifies the development of production-ready applications by providing a set of conventions and defaults. It aims to streamline the setup process and eliminate much of the boilerplate configuration traditionally associated with Spring applications. Key features of Spring Boot include:

- **Auto-Configuration**: Automatically configures your Spring application based on the dependencies you have added to the project.
- **Embedded Server**: Comes with embedded servers like Tomcat, Jetty, or Undertow, allowing you to run your application as a standalone Java application.
- **Production-Ready Features**: Includes features like metrics, health checks, and externalized configuration to help you manage your application in a production environment.
- **Starter POMs**: Provides a set of "starter" POMs to simplify the inclusion of dependencies.

### How to Start a Spring Boot Application

To start a Spring Boot application, follow these steps:

You can set up a Spring Boot project using several methods, such as Spring Initializr, IDE support, or manually. Here, I'll cover using Spring Initializr and a manual setup with Gradle.

1. **Visit Spring Initializr**: Go to start.spring.io.
2. **Fill Out Project Metadata**: Select the following options:
   - **Project**: Gradle Project
   - **Language**: Java
   - **Spring Boot**: 3.2.5
   - **Packaging**: Jar
   - **Java**: 17

### Running the Spring Boot Project Locally

#### Running the Application

Using Gradle Wrapper

If you have the Gradle Wrapper (`gradlew`), run the following command in your project root directory:

```
./gradlew bootRun
```

Alternatively, you can build the project into a JAR file and run it:

1. **Build the Project**: ./gradlew build
2. **Run the JAR File**: java -jar build/libs/your-project-0.0.1-SNAPSHOT.jar

## Deploying to a Server

### Deploying to a Standalone Server

If you have a standalone server (like Tomcat), you can deploy your Spring Boot application as a WAR file. However, Spring Boot applications are typically packaged as executable JARs, which include an embedded server. This makes deployment simpler.

1. **Build the Executable JAR**: ./gradlew build
2. **Copy the JAR File to the Server**:
    a. Copy the `build/libs/your-project-0.0.1-SNAPSHOT.jar` file to your server.
3. **Run the JAR File on the Server**: java -jar your-project-0.0.1-SNAPSHOT.jar