**Project Structure Document**

The Bug Tracking System project will follow a standard Maven project structure:  
.

├── src

│ ├── main

│ │ ├── java

│ │ │ └── com.example.bugtracker

│ │ │ ├── BugTrackerApplication.java

│ │ │ ├── config

│ │ │ │ └── WebMvcConfig.java

│ │ │ ├── controller

│ │ │ │ ├── UserController.java

│ │ │ │ └── BugController.java

│ │ │ ├── model

│ │ │ │ ├── Bug.java

│ │ │ │ └── User.java

│ │ │ ├── repository

│ │ │ │ ├── BugRepository.java

│ │ │ │ └── UserRepository.java

│ │ │ └── service

│ │ │ ├── BugServiceImpl.java

│ │ │ └── UserServiceImpl.java

│ │ └── resources

│ │ ├── static

│ │ └── templates

│ └── test

│ └── java

│ └── com.example.bugtracker

│ ├── controller

│ │ └── BugControllerTest.java

│ └── service

│ ├── BugServiceTest.java

│ └── UserServiceTest.java

└── pom.xml

# ****Explanation of Directories and Files****

* **src/main/java**: This directory contains all the Java source files for the project.
  + **com.example.bugtracker**: This is the root package for the project.
    - **BugTrackerApplication.java**: This is the main class that contains the **main()** method to start the Spring Boot application.
    - **config**: This directory contains the configuration files for the Spring Boot application.
    - **controller**: This directory contains the REST API controller classes for the Bug and Admin functionalities.
    - **model**: This directory contains the Java POJO classes for the Bug and User models.
    - **repository**: This directory contains the Spring Data JPA repositories for the Bug and User entities.
    - **service**: This directory contains the service classes that implement the business logic for the Bug and User functionalities.
* **src/main/resources**: This directory contains the non-Java files for the project.
  + **static**: This directory contains the static assets (such as CSS, JavaScript, and images) for the web application.
  + **templates**: This directory contains the HTML templates for the web application.
* **src/test/java**: This directory contains all the unit and integration tests for the project.
* **pom.xml**: This file contains the Maven project configuration and dependencies.

This project structure separates the code into different layers for better organization and maintainability. The controller layer handles the HTTP requests and responses, the service layer contains the business logic, and the repository layer interacts with the database. The test directory contains the unit and integration tests for the project to ensure that the code is functioning correctly.