Deployment Document

System Requirements

- Hardware Requirements
- Processor: 1 GHz or higher
- RAM: 1 GB or higher
- Storage: Sufficient disk space to store the application and database

Software Requirements

- Operating System: Any OS that supports Java, Apache Tomcat, and MySQL
- Java Development Kit (JDK): Version 8 or higher
- Apache Tomcat: Version 8 or higher
- MySQL Database: Version 5.7 or higher
- MySQL Connector/J: Latest version

Dependencies

If the recipients of Maven-based web app are using Eclipse, they can follow these steps to build the Maven dependencies:

1. Make sure the Java Development Kit (JDK) is installed on their system. They can download and install it from the Oracle website (https://www.oracle.com/java/technologies/javase-jdk11-downloads.html).

2. Install the Maven Integration for Eclipse plugin if they don't have it already. They can install it by following these steps:

- Open Eclipse go to "Help" -> "Eclipse Marketplace".
- In the "Eclipse Marketplace" dialog, search for "Maven Integration for Eclipse".
- Click "Go" and locate "Maven Integration for Eclipse" in the search results.
- Click "Go" next to "Maven Integration for Eclipse" and follow the instructions to install the plugin.

3. Once the Maven Integration for Eclipse plugin is installed, they can import your Maven-based web app into Eclipse by following these steps:

- Go to "File" -> "Import".
- In the "Import" dialog, expand "Maven" and select "Existing Maven Projects".
- Click "Next" and browse to the root directory of your Maven project that contains the `pom.xml` file.
- Select the project and click "Finish" to import it into Eclipse.

4. Eclipse will automatically recognize the Maven project structure and start downloading the required dependencies. The progress can be seen in the "Maven" tab at the bottom of the Eclipse window.

5. Once the dependencies are downloaded, the project can be built. Right-click on the project in the "Package Explorer" or "Project Explorer" view and select "Run As" -> "Maven build".

6. In the "Edit Configuration" dialog, set "Goals" to `clean install` and click "Run".

7. Eclipse will execute the Maven build process, which compiles the source code, resolves dependencies, and generates the necessary artifacts.

After following these steps, Eclipse will have built the Maven dependencies for your web app, and the recipients can work with the project within their Eclipse IDE.

Deployment:
Import the WAR file into eclipse: File->Import->Type in WAR->Next->SELECT the WAR file, ||||||Add

You will need to install MySql, and use the included buildb.sql script to generate the Database.

Inside the com.project.db.DatabaseConnection class the login, and password for your MySql need to be set.

Run as server, SELECT the Apache in the project folder

Troubleshooting:
The JDBC may be windows only it may be the case that it needs to be swapped out with a Mac JDBC, to do so place the macJDBC in the lib folder under
`Src->main->webapp->WEB-INF->lib/, and in the Apache../lib/ folder`

A. Outside Libraries

a. Chart.js
b. MySQL Connector

B. Server Settings

a. Apache Tomcat v10.0
b. apache-tomcat-10.0.27.tar extracted

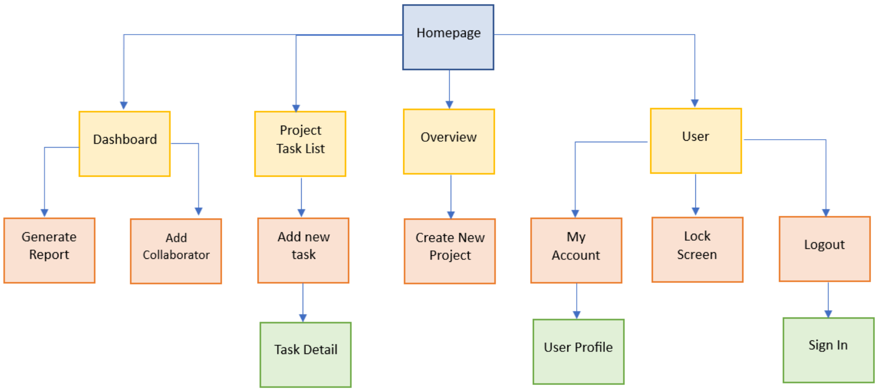
C. Java Download/Install:

a. mysql-connector-j-8.0.33.jar
b. Apache-tomcat-10.0.27.tar

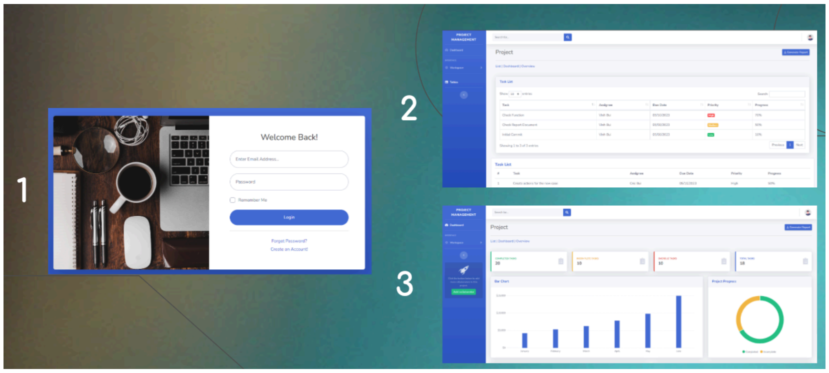
D. Detailed Design Document:

a. Mockup Page Designs
    i. Wireframe Link: https://wireframe.cc/pro/pp/032517f72638114

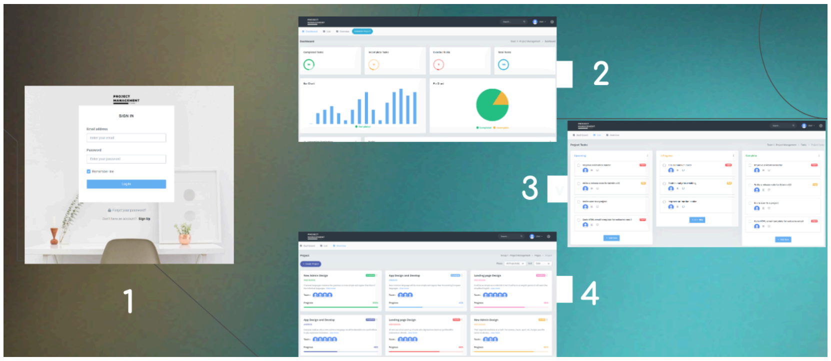
Sitemap



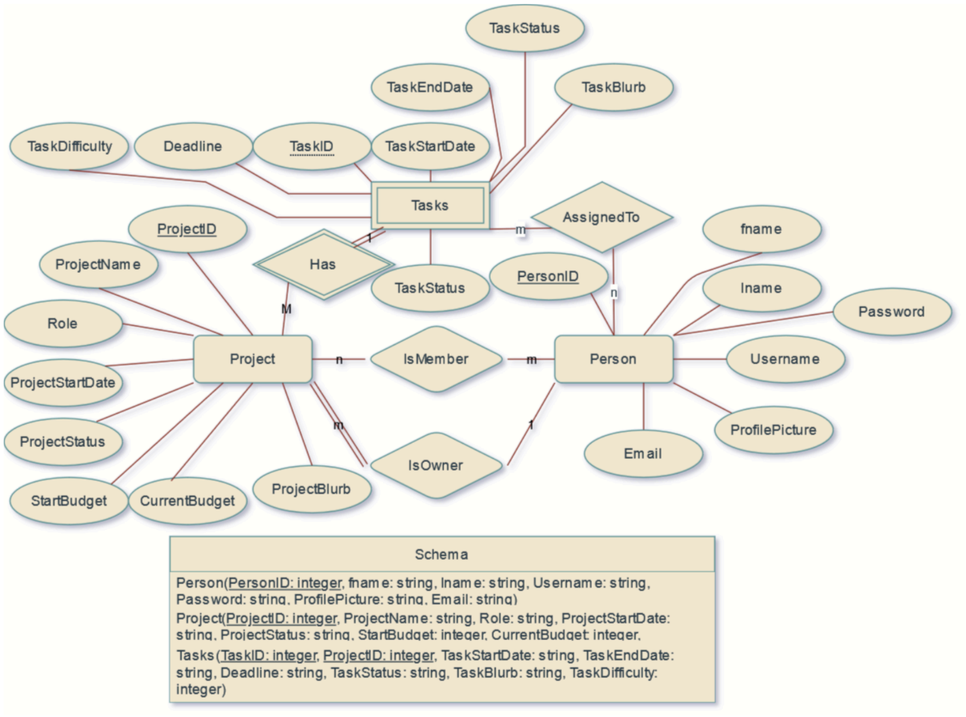
First Design



Final Design



Backend Database



Schema

Person(PersonID: integer, fnam e: string, lnam e: string, Usernam e: string, Password: string, ProfilePicture: string, Email: string)

Project(ProjectID: integer, ProjectName: string, Role: string, ProjectStartDate: string, ProjectStatus: string, StartBudget: integer, CurrentBudget: integer,

Tasks (TaskID: integer, ProjectID: integer, TaskStartDate: string, TaskEndDate: string, Deadline: string, TaskStatus: string, TaskBlurb: string, TaskDifficulty: integer)