### **Contributors:Demir Doruk Dilek,Tuna Yılmaz,Ahmet Kaan Tırhış**

|  | **Demir Doruk Dilek** | **Tuna Yılmaz** | **Ahmet Kaan Tırhış** |
| --- | --- | --- | --- |
| **Deployment Overview** | **X** |  | **X** |
| **Deployment Process** | **X** |  | **X** |
| **Configuration Plan** | **X** | **X** |  |

### 

### 

### **Deployment Plan for To-Do List Application**

#### **0. Content Table**

#### **Deployment Overview**

#### **Deployment Process**

#### **Configuration Plan**

#### **1. Deployment Overview**

The deployment of the To-Do List application consists of a backend server (Spring Boot) and a frontend application that provides a user interface for interacting with tasks. The backend manages tasks (add, edit, delete, update tags), while the frontend interacts with the backend using REST API calls. The application will be hosted on a local machine for testing purposes, but can be deployed to a cloud platform for production.

* **Backend**: Spring Boot (Java) with JPA for database management, using H2 as an in-memory database for local deployment or for production.
* **Frontend**: JavaFX for a desktop-based GUI.
* **Deployment Tool**: Maven or Gradle for building the application.

#### **2. Deployment Process**

The following steps outline the process of deploying the To-Do List application:

1. **Set up Environment**:  
   * Install JDK 17 or higher for building and running the Spring Boot application.
   * Install Maven or Gradle (depending on project setup).
   * Install JavaFX SDK for GUI.
   * Install a local database (H2 or MySQL) or configure cloud-based database services.
2. **Build Backend Application**:  
   * Navigate to the backend directory.
   * Run mvn clean install ( Marven build ) to compile and package the backend code.
   * Ensure dependencies are installed and the application can be packaged into an executable .jar file.

1. **Database Configuration**:  
   * For local development, configure the application to use H2 update application.properties for production deployment.
   * The database contain tables to manage tasks and their associated properties such as descriptions, tags, deadlines, etc.
2. **Deploy Backend**:

Deploy the Spring Boot application by running the packaged .jar file:  
  
 java -jar target/todolist-application.jar

* + For local testing the backend should be accessible at http://localhost:8080 or a cloud endpoint for production.

1. **Frontend Deployment**:  
   * For local deployment, simply run the MainView.class class using java command to start the JavaFX application.
   * For production deployment, package the frontend along with the backend into a single application (optional for cloud deployment).
2. **Test Deployment**:  
   * Test the deployment by opening the frontend, interacting with the application, and ensuring that tasks can be added, edited, deleted, and retrieved from the backend.

#### **3. Configuration Plan**

* **Backend Configuration**:  
  + Database configurations: Set up a database (H2 for development for production) in application.properties.
  + API URLs: Ensure the frontend communicates with the backend's correct API endpoints.
  + Logging: Configure logging to output errors, information, and debug logs for development and troubleshooting.
* **Frontend Configuration**:  
  + The frontend will connect to the backend API running on localhost:8080 or the production API endpoint.
  + Task display should update in real-time (with a timer in the frontend).
  + The application should include robust error handling and data validation for user input.