**F114130 House Manager Project Documentation**

**Table of contents**

1. Introduction
2. Project structure
3. Usage
4. Database Model
5. Setup and Configuration
6. Future enhancements
7. Conclusion

**1. Introduction**

The HouseManager project is a Java-based application developed to manage information related to residential buildings, apartments, residents, and building management. The application utilizes Java technologies, Hibernate for data persistence, and JavaFX for the user interface. It is designed to provide functionality for creating, updating, and deleting information about buildings, apartments, residents, building management and monthly payments.

**2. Project Structure**

The project follows a modular structure, with various components encapsulated into separate classes and modules. The key components of the project include:

- **Entities:** Classes representing entities in the application, such as `Building`, `Apartment`, `ApartmentResidents`, `BuildingManagement`, `MonthlyPayment`.

- **DAO (Data Access Object) Classes:** Responsible for handling database operations. Examples include `BuildingDAO`, `ApartmentDAO`, `ApartmentResidentDAO`, `BuildingManagementDAO` and `MonthlyPaymentDAO`.

- **User Interface (JavaFX):** The user interface is implemented using JavaFX. It includes features for adding, updating, and deleting buildings, apartments, and residents.

- **Database Configuration:** The project uses Hibernate for data persistence. Configuration files, such as `hibernate.cfg.xml` and `hibernate.properties`, define the database connection details.

- **Session Management:** Uses the SessionFactoryUtil class, which in turn utilizes the Hibernate `SessionFactory` for managing database sessions.

**3. Usage**

The HouseManager application provides the following functionalities:

- Building Management: Add, update, and delete information about residential buildings.

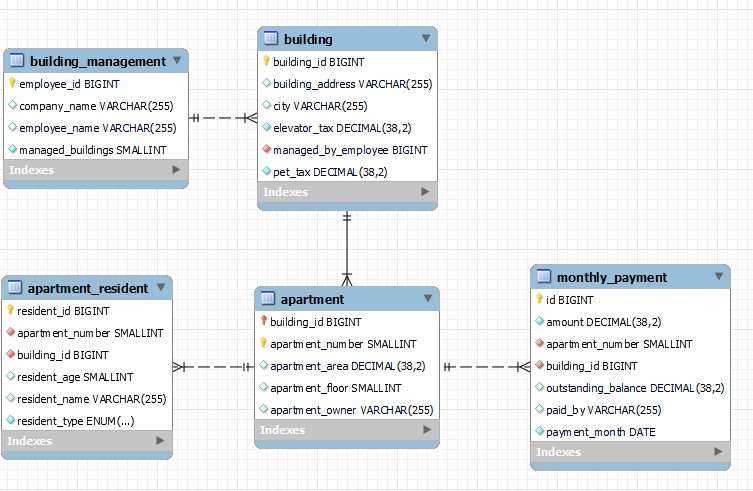
- Apartment Management: Create, modify, and remove details about individual apartments within buildings.

- Resident Management: Handle information about residents, including their names, ages, types, and association with specific apartments.

- Building Management Information: Track details about employees responsible for managing buildings.

- Monthly Payment Management: Track information about paid monthly taxes.

**4. Database Model**

****

**5. Setup and Configuration**

To run the HouseManager project, follow these steps:

1. **JavaFX Configuration:** Ensure that the JavaFX SDK is correctly installed and configured. Update the `vmArgs` section in the `launch.json` file with the correct path to the JavaFX SDK.

2. **Database Configuration:** Adjust the `hibernate.properties` file to match your database connection details, including the JDBC URL, username, and password.

3. **Dependencies:** Make sure that all required dependencies, including Hibernate and Lombok, are correctly added to the project.

4. **Run the Application:** Execute the application using the `gradlew run` command or through the configured launch configurations in Visual Studio Code.

**7. Future Enhancements**

Possible future enhancements to the HouseManager project may include:

- User Authentication: Implement a user authentication system to secure access to the application.

- Additional Features: Add features such as reporting, search functionality, and improved UI design.

- Error Handling: Enhance error handling to provide more informative messages to users.

**8. Conclusion**

The HouseManager project serves as a comprehensive solution for managing information related to residential buildings and residents. Its modular structure allows for easy maintenance and future expansion. The project demonstrates the use of Java, Hibernate, and JavaFX technologies to create a functional and user-friendly application for building and resident management.