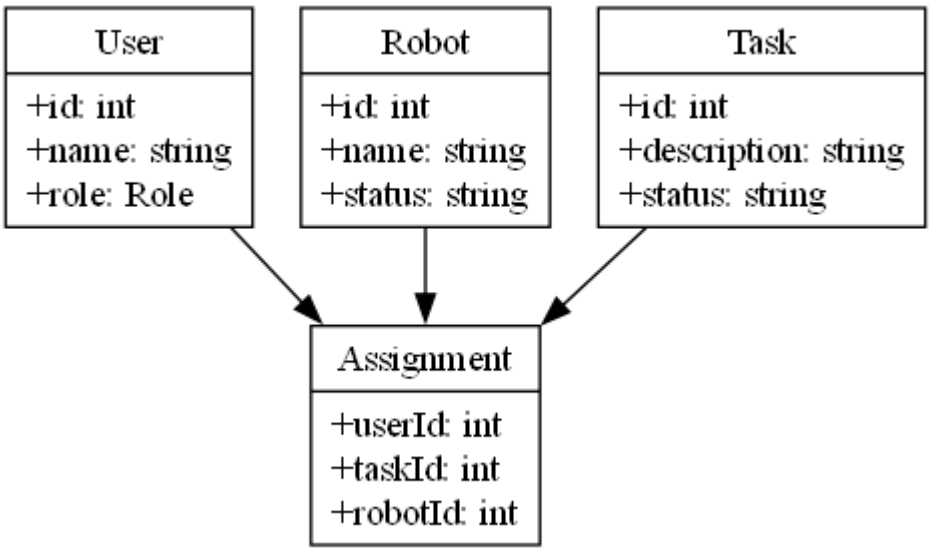
Deliverable 2
Mester Darius
Robot Management

Domain Model



Architectural Style:

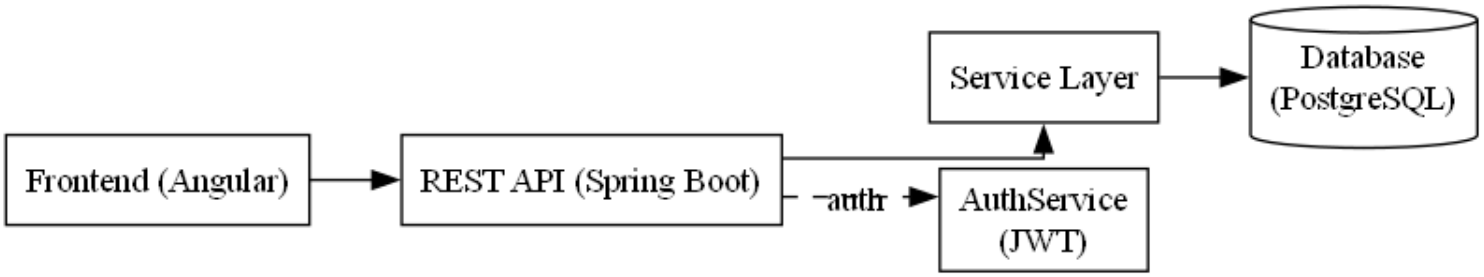
It is a **Layered Architecture** with a **Client-Server model**. This architectures separate the components (modules) like UI, API, Logic (Services), and Database Access.

Pattern Used:

The system follows the **MVC (Model-View-Controller)** pattern:

Model: Represents business entities such as **`User`**, **`Robot`**, **`Task`**, and their relationships.
View: Angular-based frontend displaying user/admin dashboards and managing interaction.
Controller: Spring Boot REST controllers handling HTTP requests and bridging UI with backend logic.

High-Level Structure:



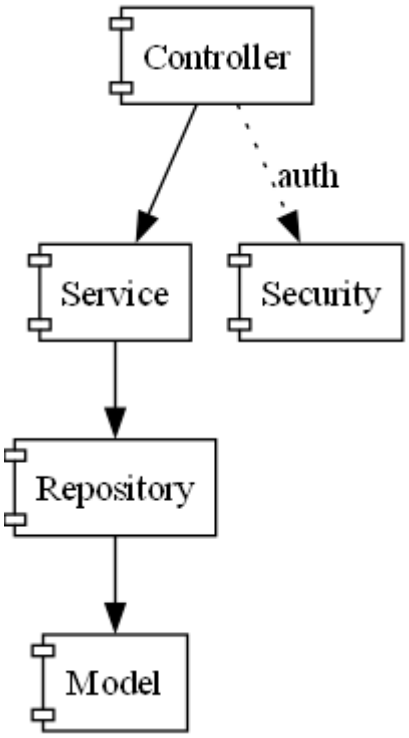
Authentication:

`AuthService` uses **JWT (JSON Web Tokens)** for authentication. When trying to login, the frontend receives a JWT and it is used in all the Services of the REST API.

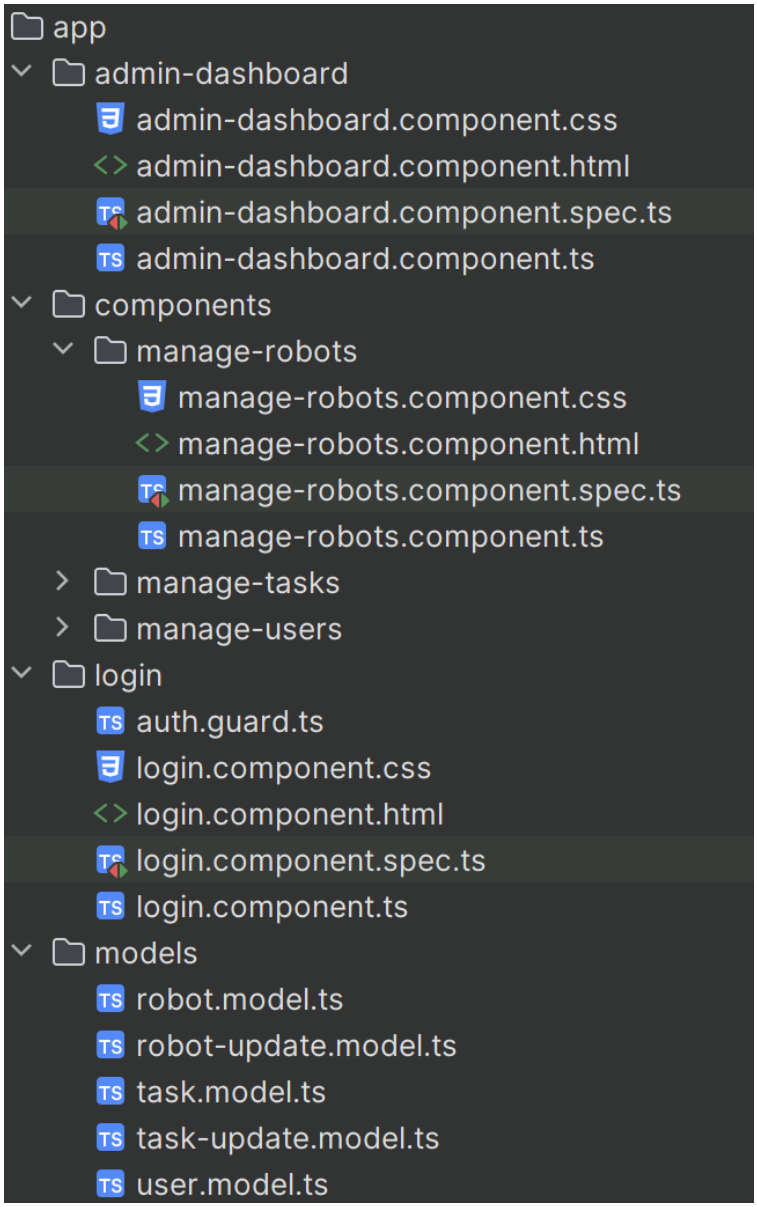
Motivation for Choice:

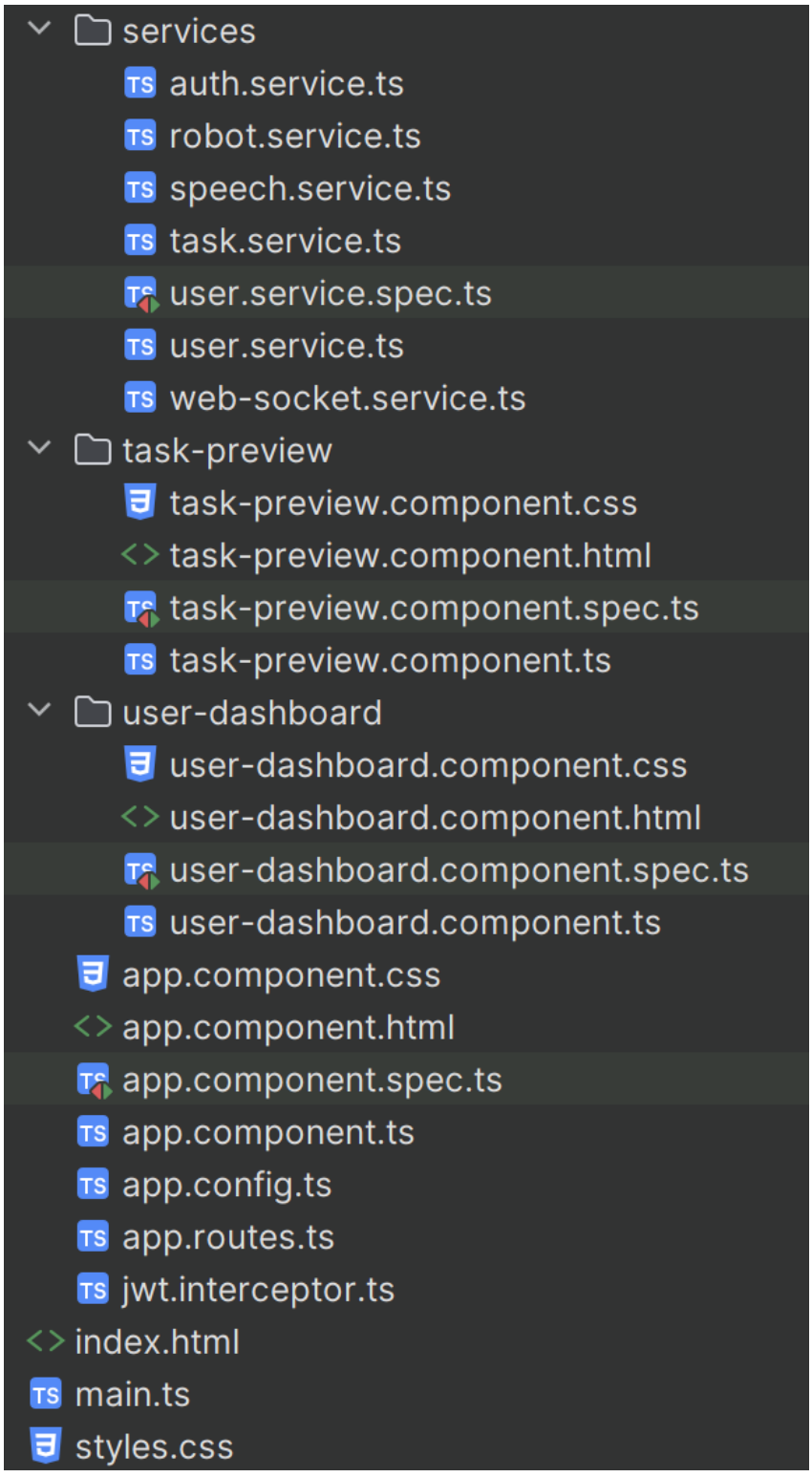
Modularity: The components are separated from each other and can be reused in other components.
Scalability: Backend services can be scaled on their own.
Security: JWT and service authorization are making sure that data access is secured.
Front-End Flexibility: Angular is a component-based framework => flexible.
Maintainability: Modules reduce code duplication and is easier to debug when there is a change.
Reusability: Services and models can be reused for both user and admin interfaces. (I added checks to see if it is an admin or user inside the services).

Package Design - Package Diagram

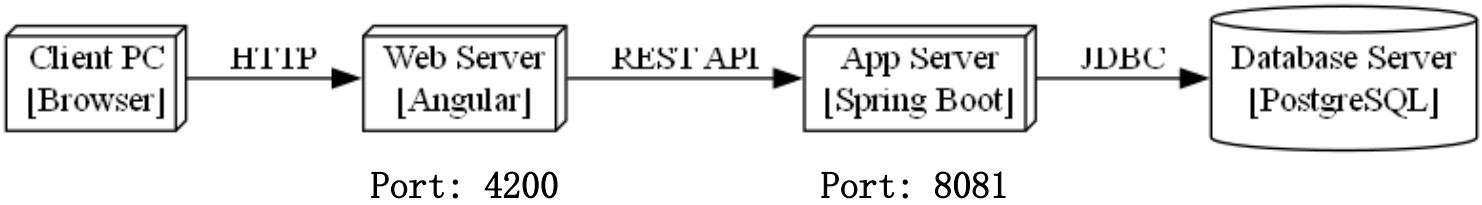


Component Diagram





Deployment Diagram



Port: 4200 Port: 8081