

# Technical documentation

## Technologies

Technologies used in this project are as follows: Vaadin, HSQLDB, Java

Vaadin was chosen because it provides a simple and powerful way of creating and managing UI components of the web application, without having to write any HTML. Modifying and getting information out of the components is also really simple using vaadin, as the components are defined in your java file; this means that you can basically access all of your UI components by just using the variable on which the component is defined. This also makes it really smooth to implement and use different classes which are needed for functionalities of the application.

For database I used HSQLDB. This database solution was used because it allows the project to be really easily tested and ran as you can use the in-memory database. Of course this also means that the information stored in database is not persistent, as the database clears when the application shuts down. Using HSQLDB also makes it really easy to import the project to other users, and it also doesn't require the user to have any kind of database server up and running.

## Design

The design of this project consists of three different java classes: MyVaadinUI is the class which handles the UI, Logic-class handles the business logic (there isn't much of logic in this application though, but in larger project it would be really important) and DbHandler handles all the database queries and connection.

In short this project consists of three different layers; UI, business logic and database. The design choices were made so that the UI would never directly communicate with databases, but instead does this by using the business logic layers as a sort of an interface between the UI and database. This kind of design solution also makes the project to be much easier to be modified if you wanted to expand it as the "layers" are separated, which would mean that you can modify for example UI without affecting the other two layers that much. It's also much safer to not connect UI straight into the database layer, as you can handle the user inputs in the logic layer as well to prevent some abuse cases on database.

All in all the design choices made when designing the architecture of this project were made keeping the eye on these few things:

- Management of the code
- Expanding and/or modifying the functionalities
- Security