# Project description

## What is this project about

Usually when working with a company you have to open a consistent group of sites and applications, such as: an app for time management, a platform for communicating with the team and the company's site. Glassbowls goal is to streamline the task of opening and managing multiple instances of sites and apps. The concept is that the user should be able to store a School of Fishes, a group of links to sites and apps. Let’s say the user interacts with the school assigned by their company, this would open all the basic pages and apps at once. This is useful for kickstarting new or inexperienced users of digital platforms or simplifying complex management routines.

## Example work flow

The user would start by creating a new School and inserting Fish into it while being able to customize what the Fish do and the visual style of the app. Once created the School would then be simply used with a click, all the while being readily accessible and easily shareable with other users.

# Requirements

## Functional requirements

- A fully fledged user system.
- Must integrate a full "school" system.
- Must have a Restful back-end Front-end must be SPA/PWA.
- The school system must be shareable and include a marketplace.
- The users have to be able to be managed in groups with roles.

## Non-functional requirements

- Front-end must be as light-weight as possible.
- Must be user-customizable.
- User personal data must be kept as low as possible.
- The application must be able to "function" without access to the backend.
- Must be secure.

# Technologies

## Development

### Editors:

Visual Studio Code: HTML, JS & CSS
IntelliJ: Java
Stackedit: MD
Google Suite:PDF & CSV
GIMP: IMGs

### Version Control:

Git
GitHub
GitHub Desktop
GitHub IntelliJ Integration

### Integration:

Netlify
Tomcat
Docker

### Front End:

Vue JS: JS Framework
Quasar: Framework
SASS: With SCSS syntax, CSS extension
Material Design: CSS Framework

### Back End:

Spring Web MVC: Java Framework
Spring Boot: Framework
MySQL: Database
Hibernate: ORM

## Management

### Storage:

Google Drive
GitHub

### Communication:

Discord

### Management:

Trello

## Why did we choose these technologies

One of the core reasons for the usage of the technologies is the familiarity the team has with them. It is also worth mentioning most if not all the technologies are freemium and allow easy development for this specific use case.

- As for code editors the team uses IntelliJ for Java, VisualStudio Code for HTML/CSS/JS.
- The team utilizes GitHub as the hub for version control, locally each member utilizes GitHub Desktop as a Git manager as well as GitHub integrations on the editors
- The environment is built upon Docker containers, these being managed by the wonderfull Docker-Compose. The containers run by Docker are a HTTPD host for the page, a Java server for the API and a MySQL db server.
- The front-end is built utilizing Quasar Dev which compiles into Vue.js the styling is done with Material Design.
- For the back-end SpringBoot is used to compile the API which will be connected to the MySQL database using the Hibernate ORM.
- The team utilizes the Github platform to manage the project's version control. Google Drive is also used as a storage medium.
- The team utilizes the Discord application as a communication channel as well as for meetings.
- The Trello timeboard is used as a task manager.

## Deadlines

- Login must be fully implemented before April 5.
- School and Fishes must be fully implemented before May 3
- User groups and sharing must be fully implemented before May 17
- After May 17 extra features and polishing de application