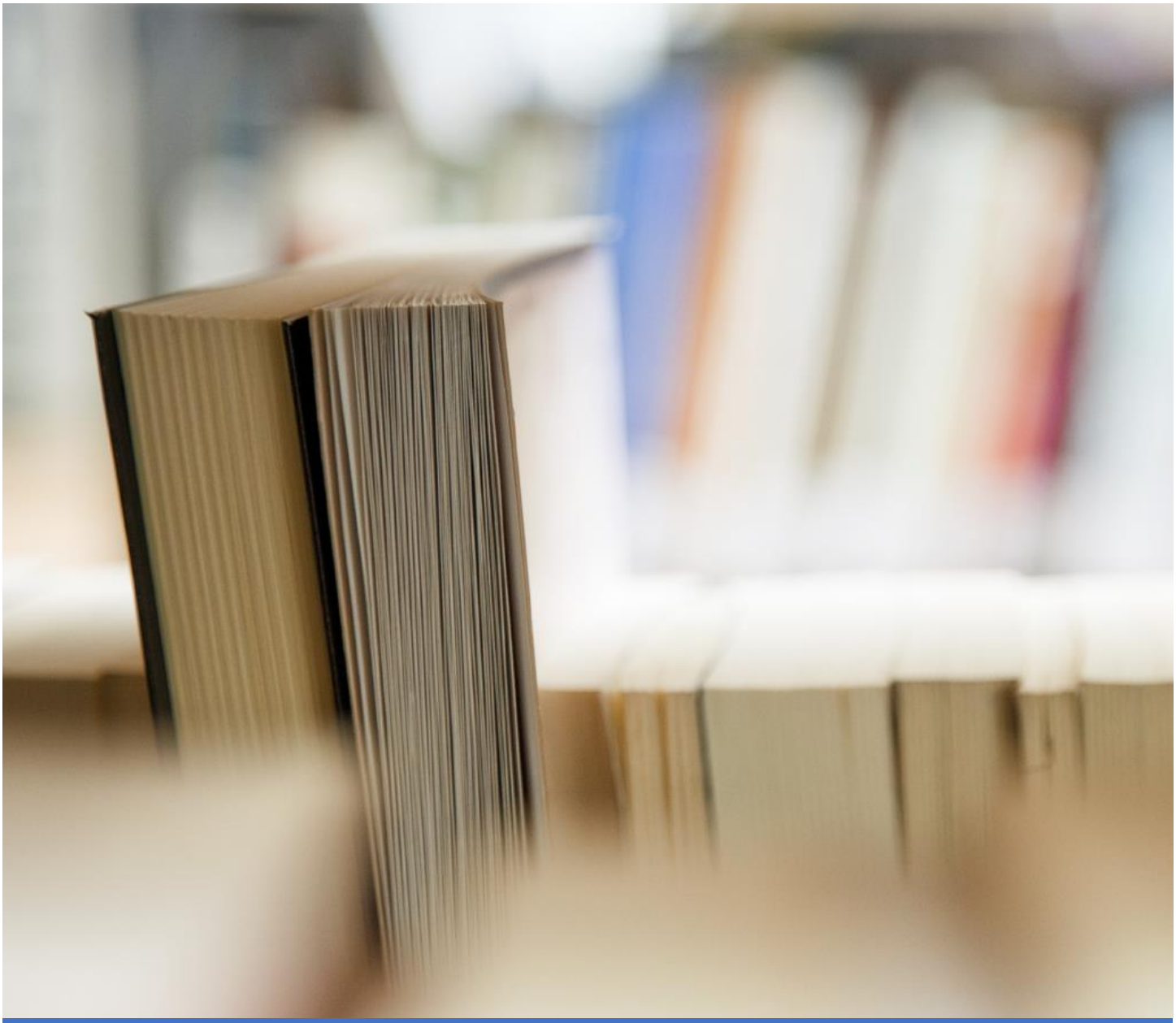


ToDo LIST WEB APPLICATION

Design Document

Wondimu Gebre

11/29/20

Individual Track Project

Contents

1. JavaScript framework selection

Now a day a lot of frameworks are on the market and these all framework has their own pros and cons. Choosing the right framework is sometimes confusing and it needs deep research. To have the right framework is very important and that will cut unnecessary costs and speed up the development process.

First, I researched and analyzed different web technologies in different perspectives and in terms of building a simple ToDo List web application. Finally, I decided to use Angular for my ToDo list front end application and I listed out the reasons below why I should be using Angular: -

1.1 Angular Advantages over React and Vue

- ➢ In 2018 Stack Overflow survey, it is ranked the second most used technologies.
- ➢ Boasted in the detailed documentation.
- ➢ Supported by Google
- ➢ Filters – contains multiple filters to format data of different data types.
- ➢ Intuitive and declarative interface
- ➢ Component-based architecture
- ➢ CLI – automates the whole development process
- ➢ Dependency injection – assists the developer in creating components, resolving their dependencies and providing them to the other components as required
- ➢ Popularity – apps built with Angular – YoutubeTV, Google Cloud, Netflix and Udacity,
- ➢ Directives – allow the developer to build custom HTML tags

1.2 Downsides or limitations of React framework

- ➢ Angular offer three 3 files for component HTML, CSS and JS, while React offers only two files for components.
- ➢ Angular offers the whole thing from routing to the template, unlike Angular React does not

Provide everything in the official library.

➢ React doesn’t have a form validation and handling like Angular. In Angular, I can easily define a customer validators and use built Angular validators, for example, min/max, email, pattern etc.

➢ JSX React’s documentation is disliked by many developers.

1.3 Downsides or limitations of Vue frameworks

➢ Vue has only one file for the component, in that reason I prefer to use Angular to work on three components.

➢ Limited resources and Fewer tutorials, according to me, in order to do one project in a fast and concurrent way, first we need to check if this technology has enough resources

Or not. According to many developers and my finding, Vue doesn't have enough resources.

➢ Lack of experienced developers - Always better to have experienced people in some topics or technology area to get advice and feedback. In the case of Vue, don't get a lot of experienced developers.

➢ Language barriers, I found most of the codes and examples in Chinese language.

In conclusion, there is no better framework here, all framework has its pros and cons. Three of them are very fast frameworks. But I have to choose only one framework for my project. In that reason, I choice Angular because of the above reasons. And, Angular framework is preferable to access information and data security than Vue and React.

2. Backend technologies

2.1 Why Spring Boot?

- It provides a powerful bach to manage REST endpoints.
- In Spring boot, everything is auto configured; no manual configurations are needed.
- It reduces overall development time and increase efficiency by having a deafult configuration for unit test.
- It provides a very good support to create a DataSource for Database.
    - ✓ Just adding the dependencies and doing the configuration details is enough to create a DataSource and connect the Database.

3. Database Technologies

3.1 Why MySQL database?

- It is a free and open source relational database management system.
- It is highly extensible, reliable, compatible with all major hosting providers and easy to manage.
- Connectivity and security, it is fully networked, and I can share my data with anyone, anywhere. And, it supports encrypted connections using the secure Sockets Layer protocol.
- Open distribution and source code, it is easily obtainable; simply use my browser if I don't understand how something works.
- In addition, fonts provide free MySQL access for its students.

4. Software architecture

4.1 Level 1: Context diagram

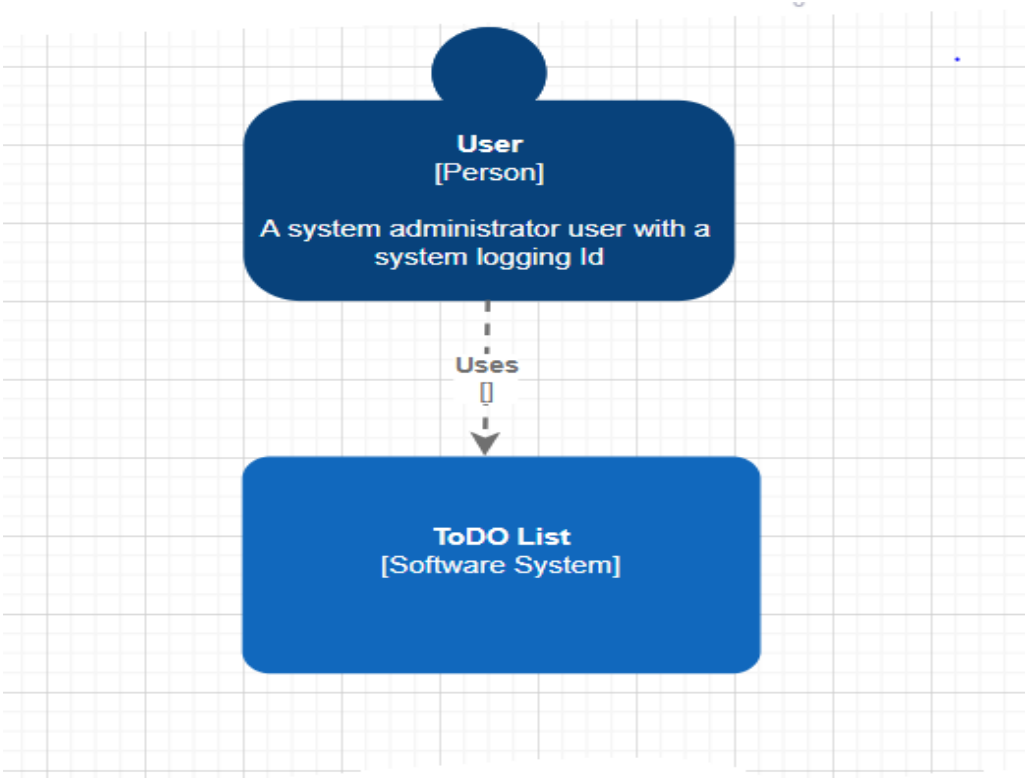


Diagram 1 - Context Diagram

The above diagram is according to the user view and, it shows how the system fits into the world.

The admin/manager of the company uses ToDo List system to do CRUD functionalities, for assigning roles and tasks to employees, and categorize tasks within departments.

4.2 Level 2: Container Diagram

The diagram below shows a container which zooms into the software system and shows the container which creates the software system. Moreover, in this diagram, I have attempted to demonstrate the technological decisions that will be used to create the system.

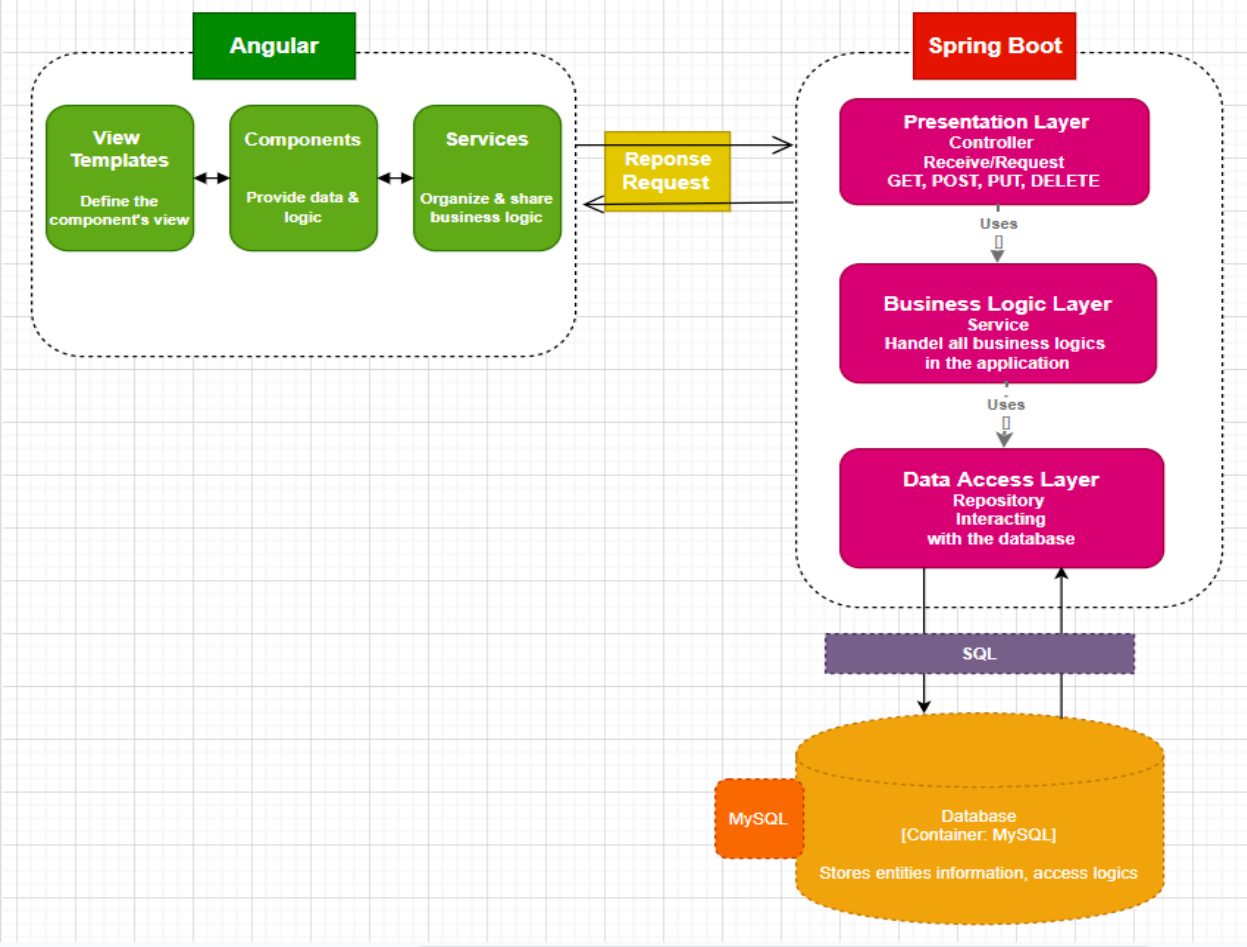


Diagram 2 – Container diagram

The backend application has three different layers: **A presentation layer, a Business logic layer, and a Data access layer/Persistence layer.**

The **Presentation layer** has controller classes and, these controller classes receive requests from the client and maps that request and handle it. It means that the client requests are handled in these controller classes. After that, it calls the service logic if required. These classes are put in a controller package in my project.

The Business logic layer has service classes and, in these service classes, all the business logic (data process, data transformations, and cross-record validations) performs. After performing the business logic uses the data access layer to perform the logic on the data that is mapped to JPA with model classes. And, these classes are put in the Service package in my project.

The Data Access layer/ Persistence layer contains all the storage logic (database queries) and translates business objects from and to database rows. It is mainly responsible to interact with the database. This layer has repository classes and, this layer is responsible for CRUD operations on a data source, which is a relational database. It is implemented using Spring Data JPA. Repository classes are put in a repository package in my project.

Technologies:

The frontend application is an Angular JavaScript framework application that works in a web browser and, it is delivering and presents all features of the ToDo List backend app. The backend application is a Java, Spring Boot application. And, to organize and persist data, the system uses MySQL database technology.

Remarks: Front side tools and technologies used: Angular 9, TypeScript, NodeJS and NPM, IDEA, Angular CLI, and bootstrap 4+.

1. Why NodeJS in Angular, it allows me to spin up a lightweight webserver to host the application locally on my system.
2. Why NPM, it gives me angular CLI (angular command-line interface)

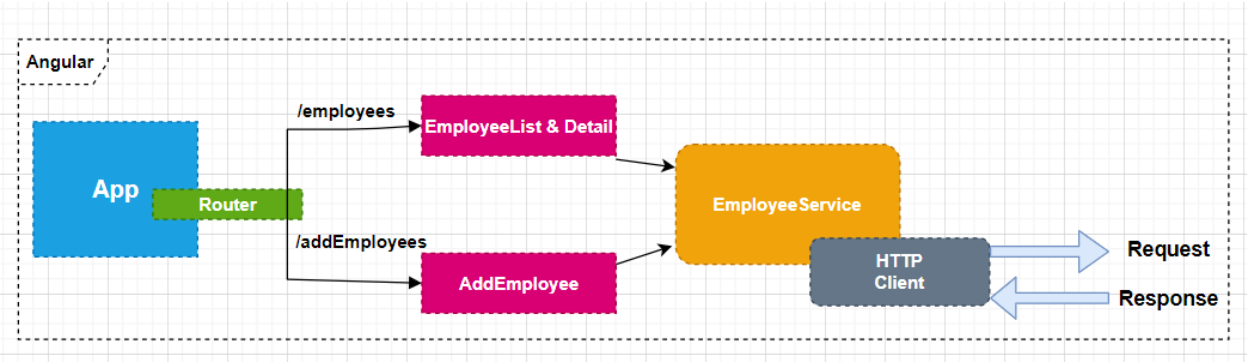
Remarks: The backend technologies used: Spring Boot 2.3.4 +, Spring Data JPA (Hibernate ORM core version 5.4.2l), Maven 3.2 + JDK 1.11, Embedded Tomcat 9.0.38 +,

Remarks: Database technologies used: MySQL

4.3 Level 3: Component diagram

4.3.1 Front-end Angular Application

The below diagram shows only employee components within the Angular 9 front-end application. Other components are also following the same process as the Employee component.



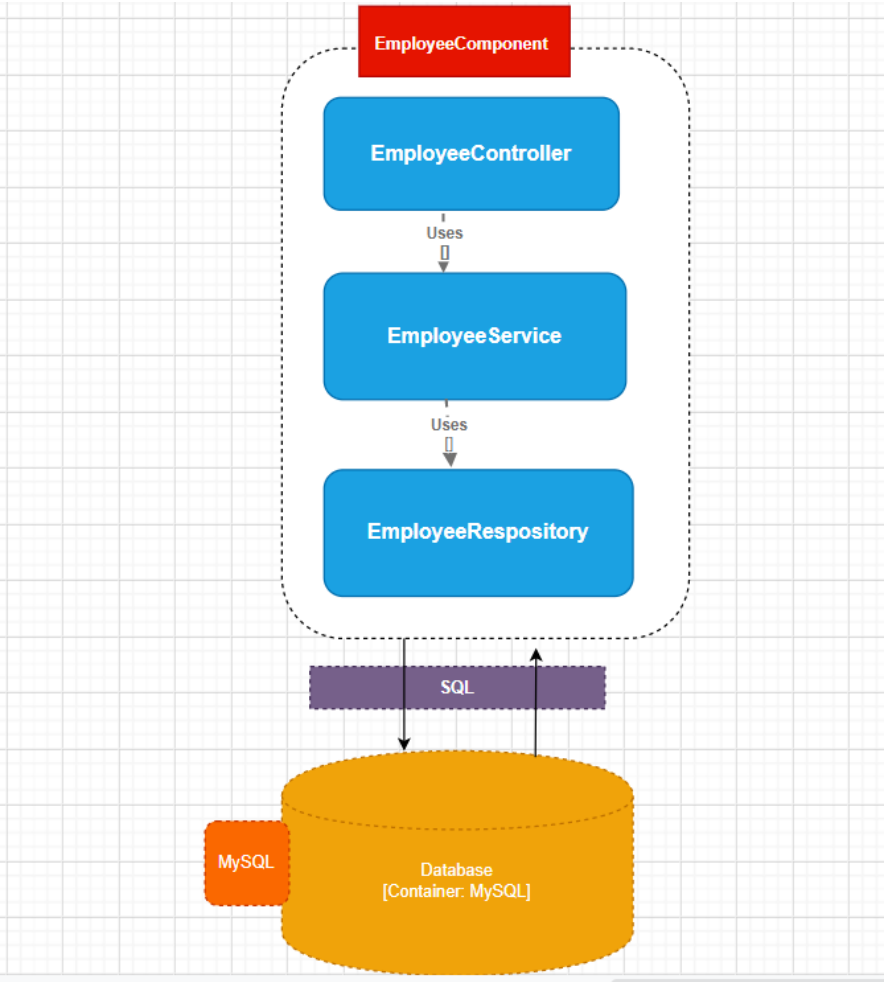
The App component is a container with a router-outlet. It has a navigation bar that links routes through the router link.

- EmployeeList component gets and displays the employee.
- EmployeeDetail component is for editing Employee's detail based on :id.
- AddEmployee component for adding/submission a new Employee.

These Components call EmployeeService methods which use Angular HTTPClient to make HTTP requests and receive responses. EmployeeService layer is responsible to handle all business logics.

4.3.2 Backend Application

In below diagram, I will try to demonstrate employee component in different layers. All other components are also having the same diagram as employee component.



In the Employee component presentation layer, there is an EmployeeController class that provides APIs for creating, retrieving, updating and, deleting Employees.

In the business logic layer, there is an EmployeeService class and, in this service class, all the business logic (data processing, data transformations, and cross-record validations) performs.

Data Access layer/ Persistence layer at the Employee component has an EmployeeRepository interface to interact with Employees from the database. This EmployeeRepository interface extends JpaRepository to use JpaRepository methods like (save(), findById(), findAll(), delete(), deleteById()). EmployeeREspository interface is responsible for CRUD operations on a data source, which is a relational database. It is implemented using Spring Data JPA.

Back-end overview

Spring Boot exports REST Apis using Spring Web MVC & interacts with MySQL Database using Spring Data JPA. Below I would like to show only Employee APIs that Spring App will export. And, all other entities use the same overview as Employee

Methods	URLs	Actions .
POST	/employees	create new Employee
GET	/employees	retrieve all Employees
GET	/employees/:id	retrieve an Employee by :id
PUT	/employees/:id	update an Employee by: id
DELETE	/employees/:id	delete an Employee by :id

5. ERD

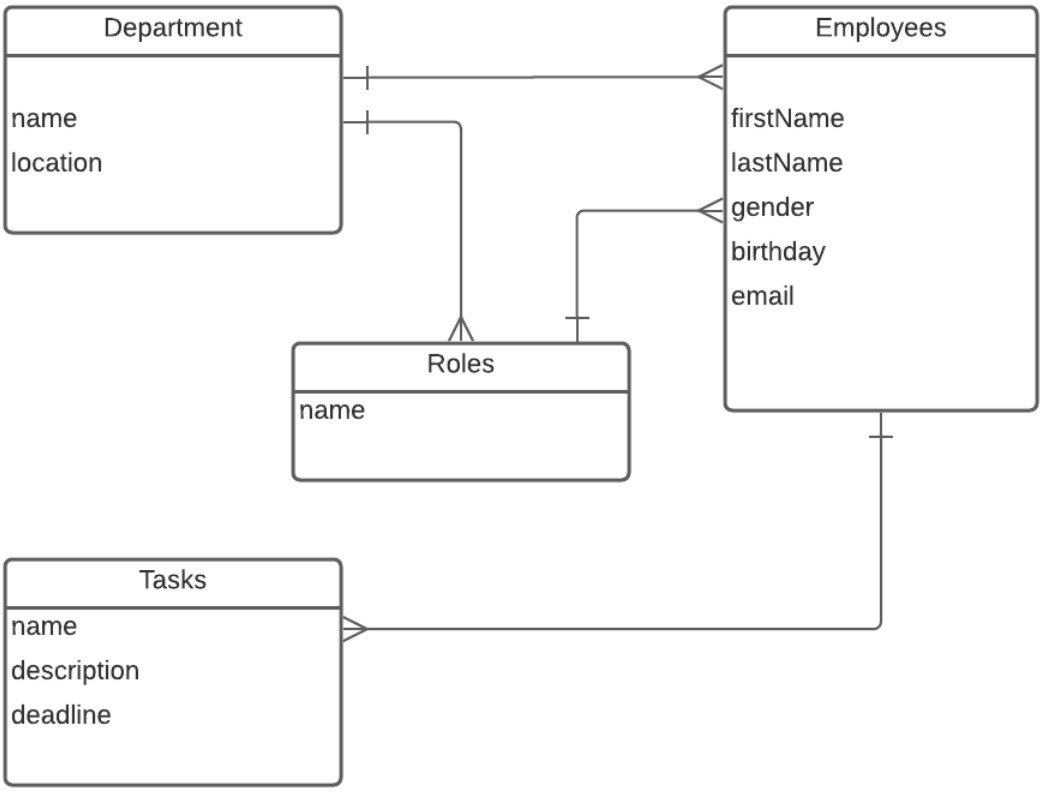


Diagram 5 - ERD

6. Ci/CD Integration test report Screenshot

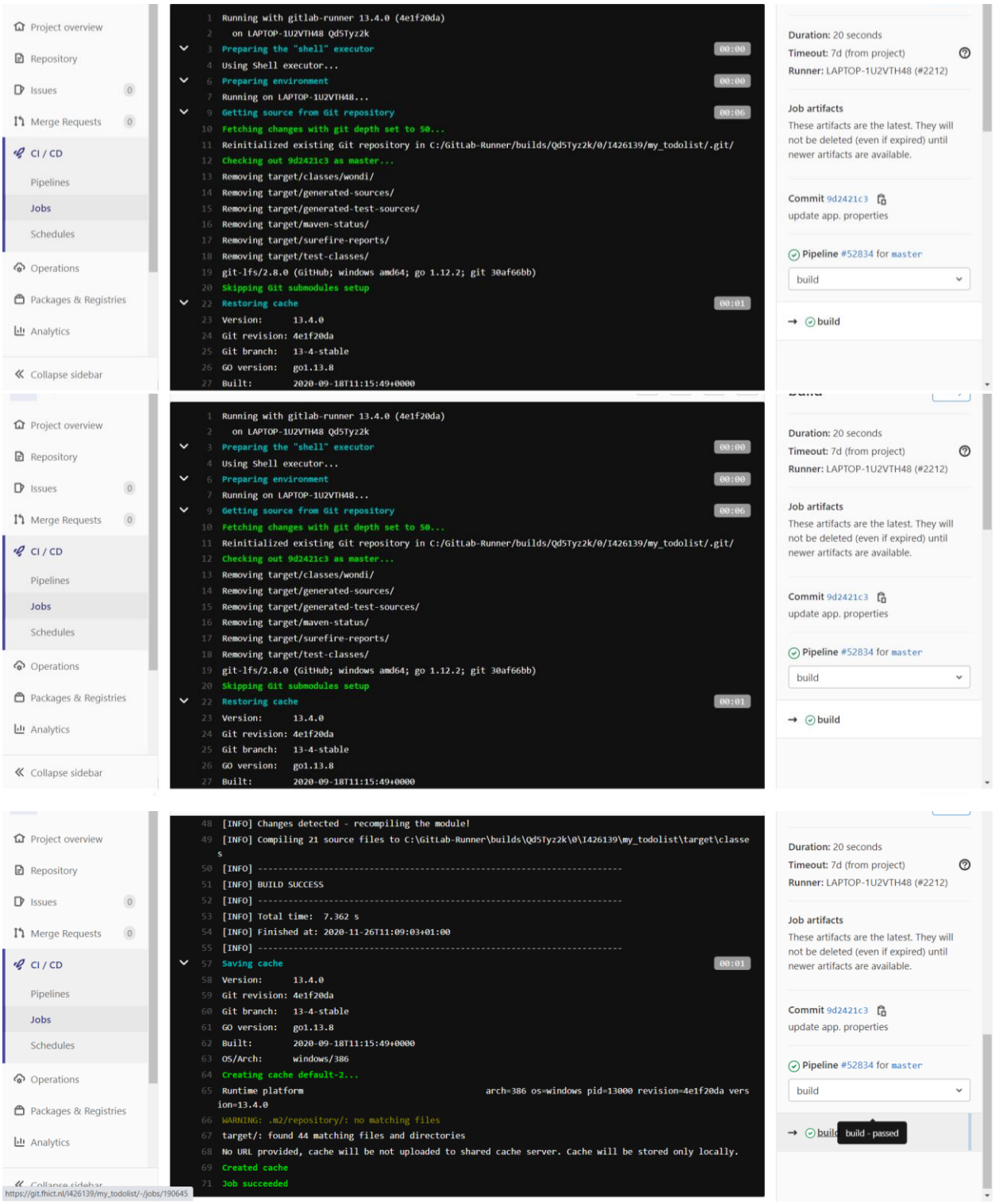


Diagram 6 – CI/CD integration test result

7. CI/CD integration Diagram

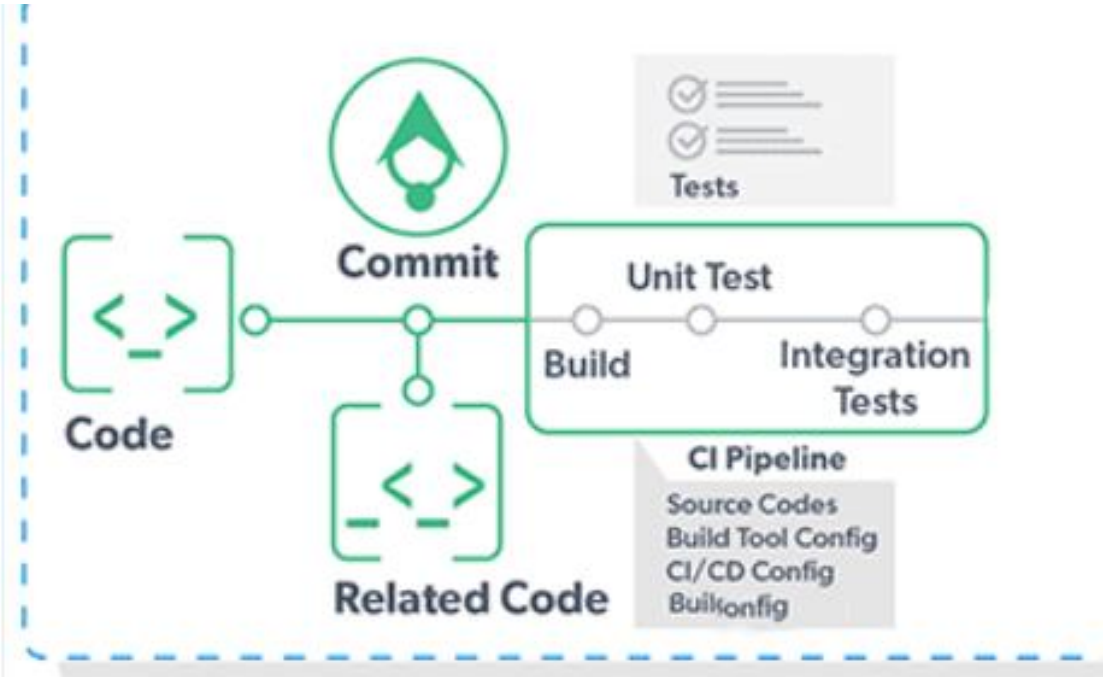
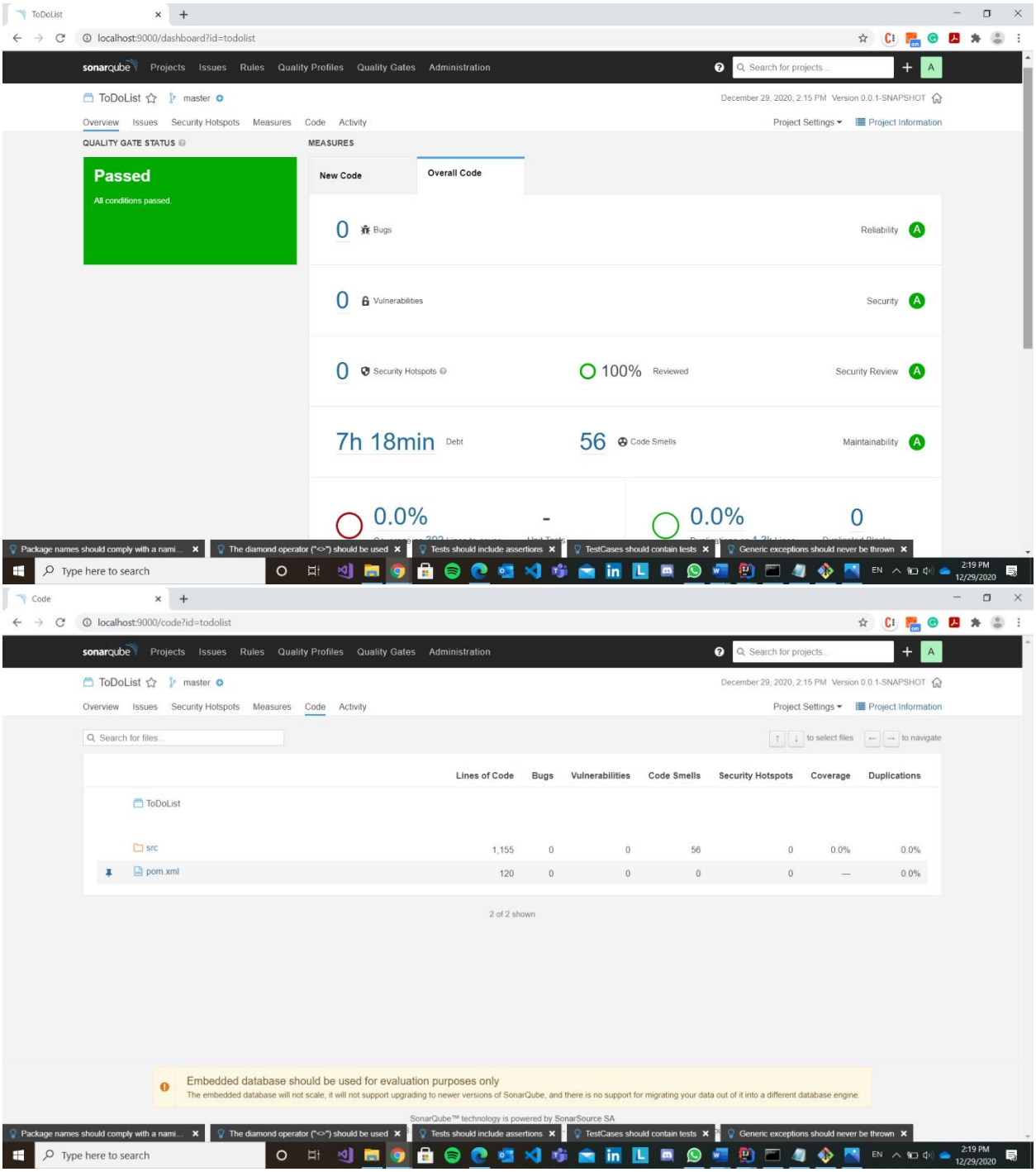


Diagram 7 – Ci/CD integration Diagram

8. SonarQube test report screenshot



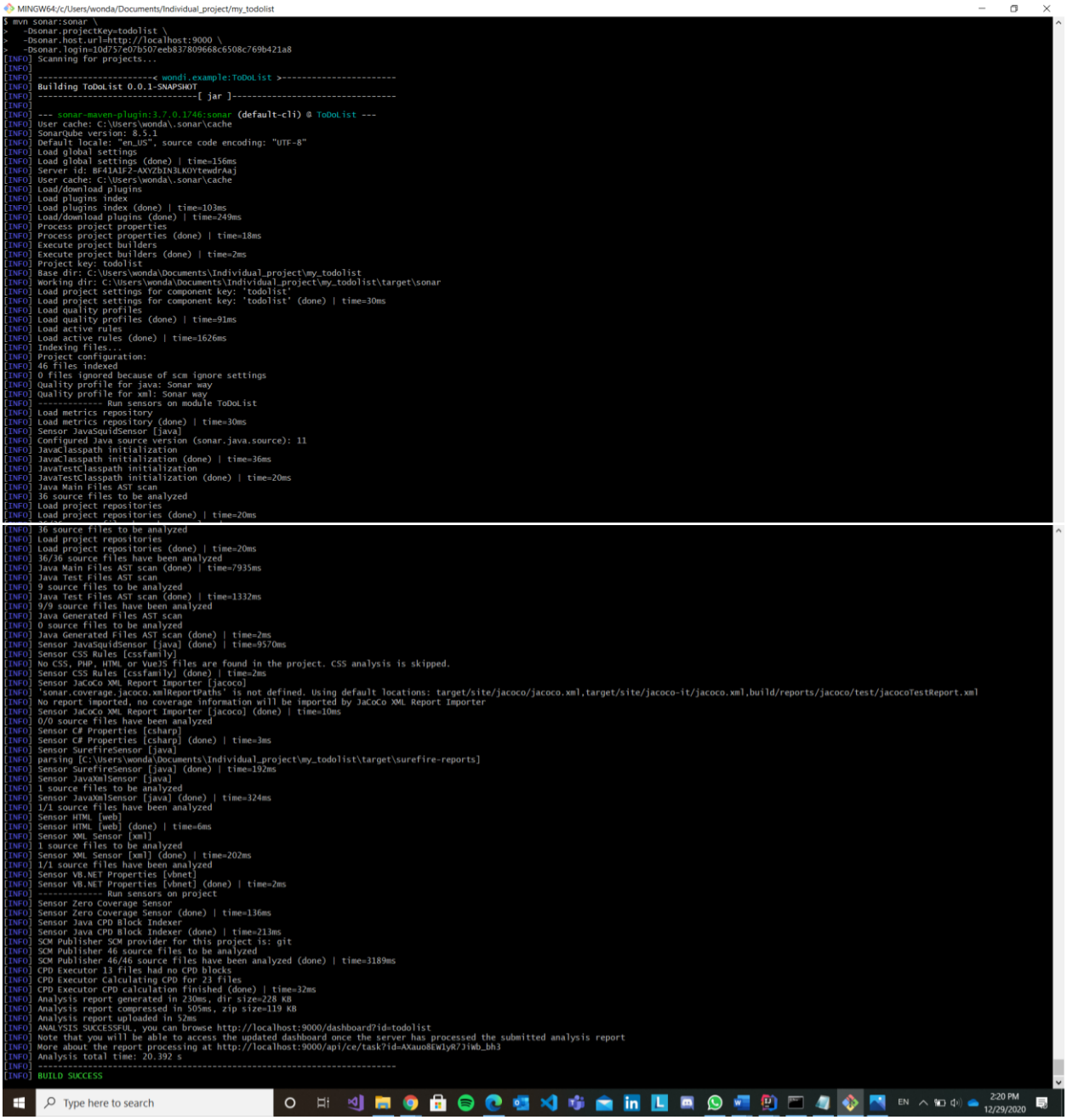


Diagram 8 – SonarQube report screen shot

I. Reference

- *Angular*. (n.d.). Angular Tour of Hero. Retrieved March 9, 2020, from https://angular.io/tutorial
- *MySQL Tutorial - Learn MySQL Fast, Easy and Fun.* (2020, March 30). MySQL Tutorial. https://www.mysqltutorial.org/
- *Spring | Guides*. (n.d.). Spring. Retrieved October 1, 2020, from https://spring.io/guides#tutorials
- *GitLab CI/CD*. (n.d.-b). GitLab. Retrieved September 8, 2020, from https://docs.gitlab.com/ee/ci/