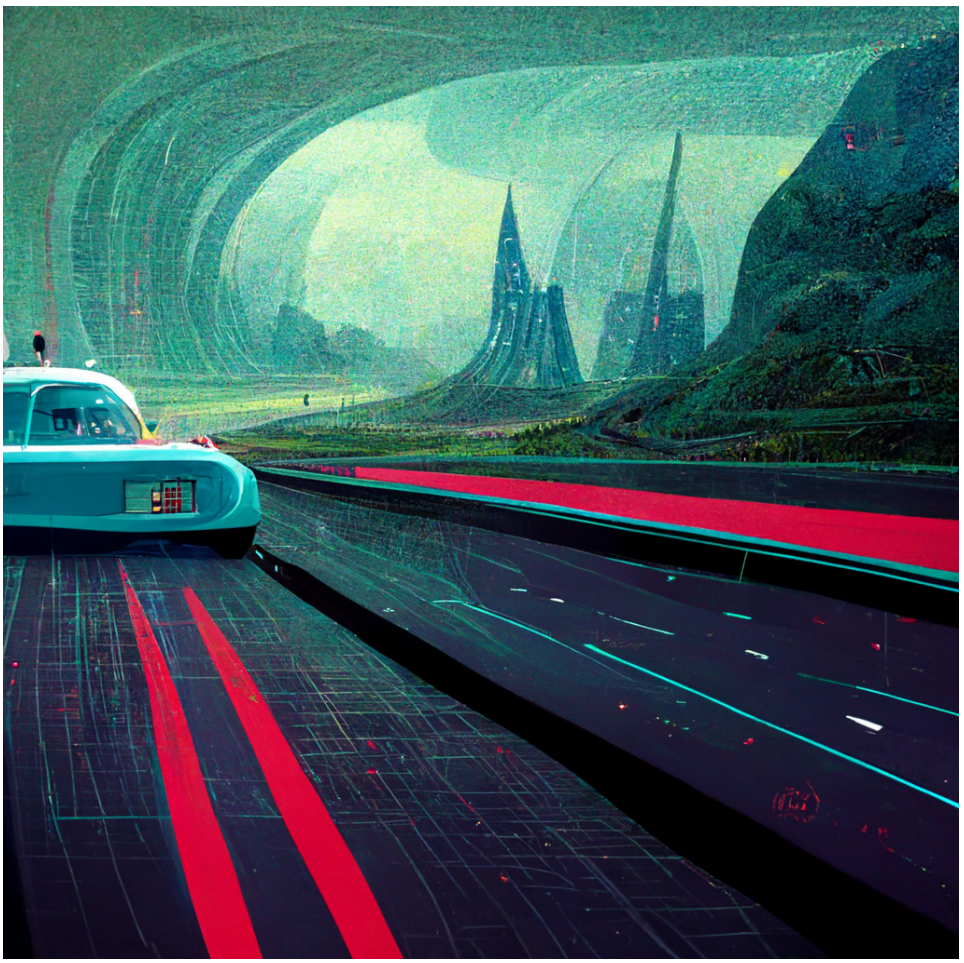
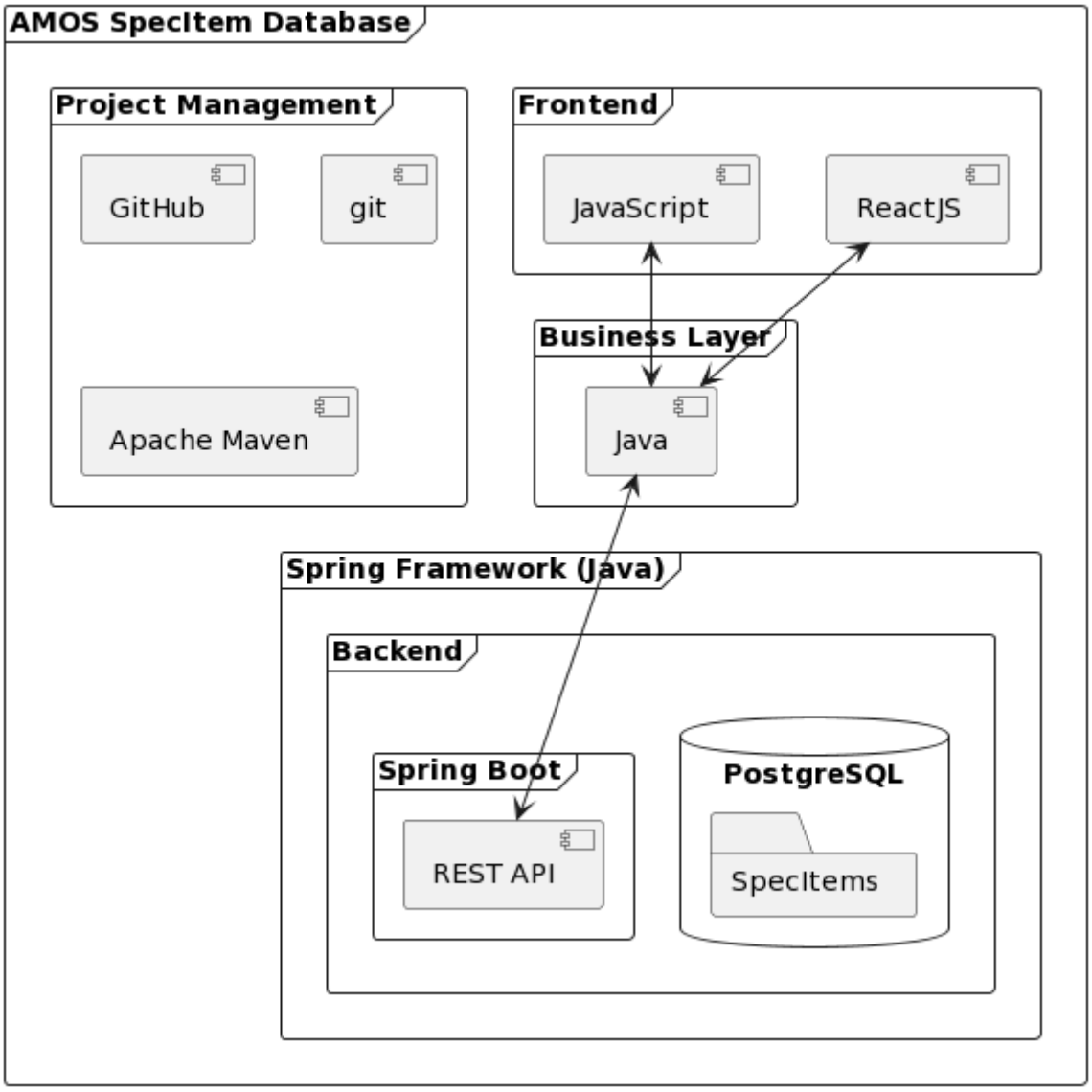
System Architecture Description



Project 4 - SpecItem Database

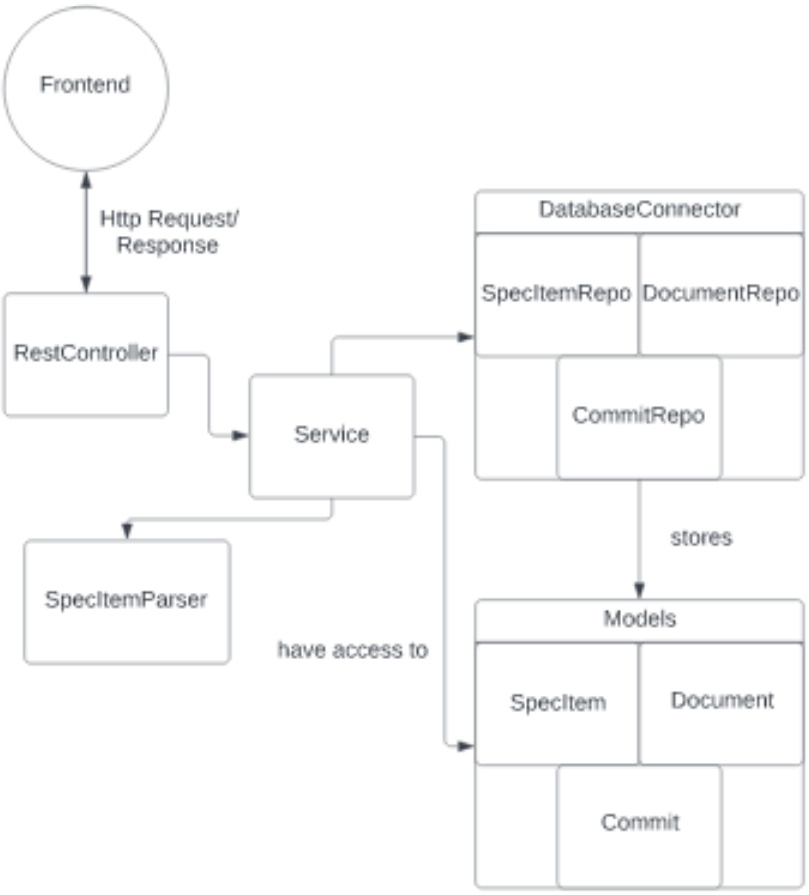
AMOS WS 22/23

Runtime Component Diagram



For the SpecItem Database project we have the task to create a webpage where the user could get, upload or delete documents, containing the so-called specification items, that were uploaded in a Database. For the project versioning we will use GitHub where we will store our code, and as a project management tool Apache Maven was our choice, because it provides support for the Java Spring Boot, and has a good documentation. We have decided to use ReactJS as our main frontend JS framework in order to build an interactive user interface efficiently and with less code. The frontend will send HTTP requests to the Spring Framework, which is using Java, and these requests will be handled in the backend from the Spring Boot. After which it should be decided whether to execute the request and allow access to the DB or deny it. As a database system we went for PostgreSQL because we will need an object-relational one and again it has useful features.

Code Component Diagram



As stated by the industry partner, users should be able to get, edit and add SpecItems to the Database via User Interface. So we implement both client and server sides. Our architecture is a classical MVC-Architecture.

- The end user interacts with the View Component, we use ReactJS for it.
- The frontend and the RestController (MVC-Controller) communication will be handled via HTTP Protocols.
- Our MVC-Controller has two Services:

1) Parsing the text files to transform the unstructured SpecItems to our specific database format.

2) Manipulating the Models inside our PostgreSQL database via Springboot DatabaseConnector according to HTTP Requests.

We have 3 different types of Models:

1. Documents
2. SpecItems
3. SVN Commit

Summary of the Underlying Technology Stack

Frontend

Tool	Version	Summary
JavaScript	-	Used to create a Web Application
ReactJS	18.2.0	JS library for building user interfaces

Backend

Tool	Version	Summary
Spring Boot/ Data JPA	2.7.5	Used for REST API
PostgreSQL	15	Database that stores the SpecItems

Business Layer

Tool	Version	Summary
Java	11	Service between Frontend and Backend

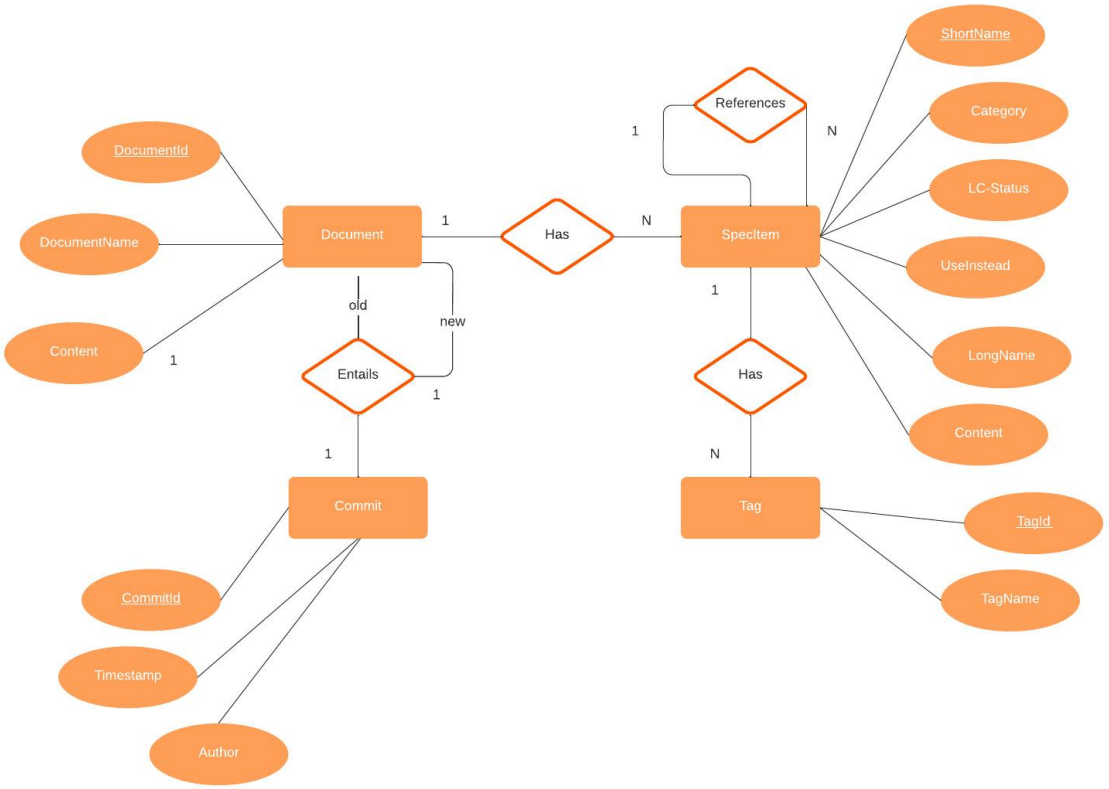
Project Management

Tool	Version	Summary
GitHub	-	Version Control
Apache Maven	3.8.6	Build Tool

ER-Diagram

The following picture presents the ER-Diagram. In the figure one can distinguish the following entities:

1. Document
2. SpecItem
3. Tag
4. Commit



Particular entities are interconnected via a series of relationships:

1. A single document contains (has) one or more SpecItems
2. A SpecItem can refer to one or more other SpecItem. It is also possible that a given SpecItem has no references.
3. A SpecItem can have zero, one, or more Tags.
4. A given commit entails a single document.

The described relationships can be translated to 7 relations (tables) in the database (the underlined attributes are meant to form the primary key):

1. Document(DocumentId, DocumentName, Content)
2. Commit(CommitId, Timestamp, Author)
3. DocInCommit(CommitId, DocumentId)
4. DocSpecItem(DocumentId, SpecItemId)
5. SpecItem(ShortName, Category, LC_State, UseInstead, LongName, Content)
6. Tag(TagId, TagName)
7. SpecItemTags(SpecItemId, TagId)