Overview

This technical design documentation provides information on the architectural design decisions of the product. It will illustrate the components of the system and their relation with each other. Users of the software will gain an understanding of how the product works as well as understand the front- and backend architecture decisions.

First, the architecture and data model will be discussed and afterwards it will go more into detail about the implementation of certain components and which technology was used.

Terminology

Here you can find some terms that are used in this documentation.

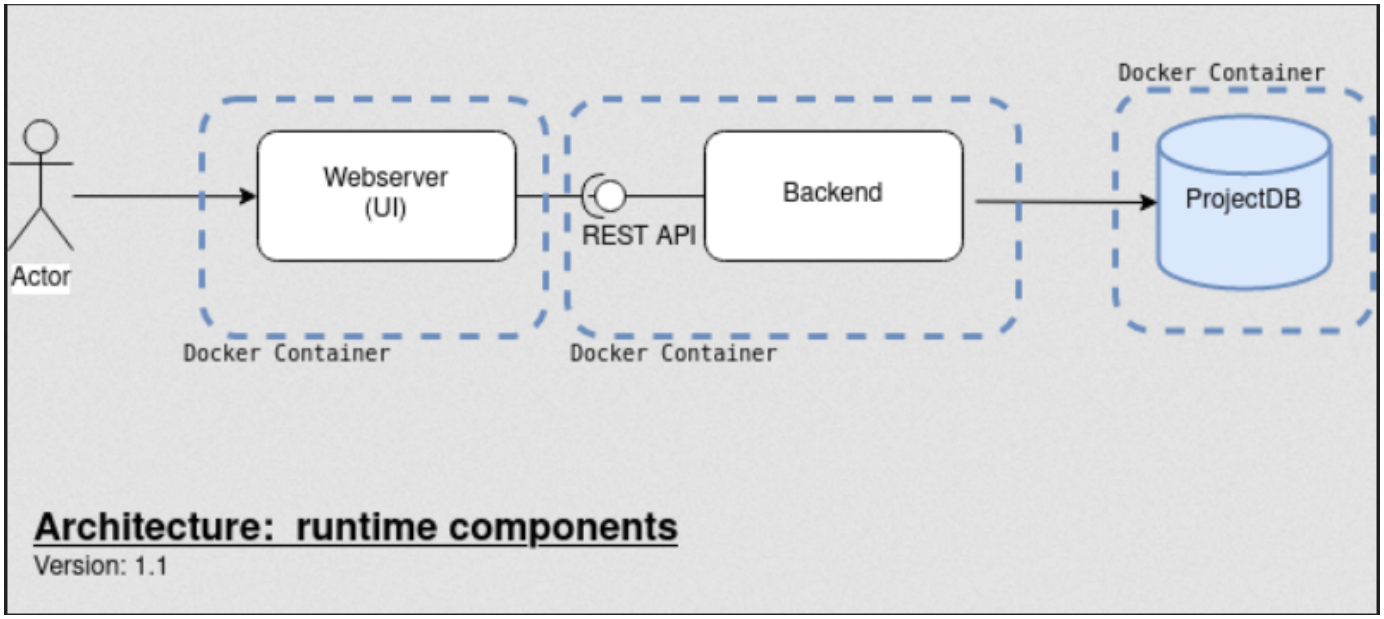
Term	Definition
Admin	The admin can create/delete users and assign roles
Consultant	The consultant is the main user of the software, which can conduct analysis, enter ratings and print charts.
Evaluation view	An Epic describes several issues which are grouped as one topic
Komplexitätskriterium/-treiber	Is a class of questions with the same topic
Marge	The margin whichs displays the y-axis in the in the pie chart
Piechart (äußerer Ring)	The customer complexity, for each Produkt three fields for the percentage of hoch, mittel & gering complex customer is given.
Piechart (innerer Ring)	The distribution of Bewertungen of Produktvarianten for one Produkt.
Piechart (Größe)	The volume of the piechart displays the criteria "credit volume" in the economical evaluation
Produktbereiche	Every product area has to sub areas (private and corporate) & and there are 1 to n products for each product area
Produkt	Every product has one to n product variants, which are divided in product areas

Rating	Hoch, Mittel & gering (with colours: red, yellow & green); is set by a consultant
Result view	The webpage with the charts
Projekt	A Projekt has one or many Produkte; Each Projekt has 1-n Produktbereiche
User	A User is ther overall class for Consultant, Admin and Project Manager
Gesamteinschätzung wirtschaftliche Bewertung	This value is used to determine the economical complexity of Produkt or a Produktvariante; This value is displayed in the inner piechart ring

Architecture & Design Principles

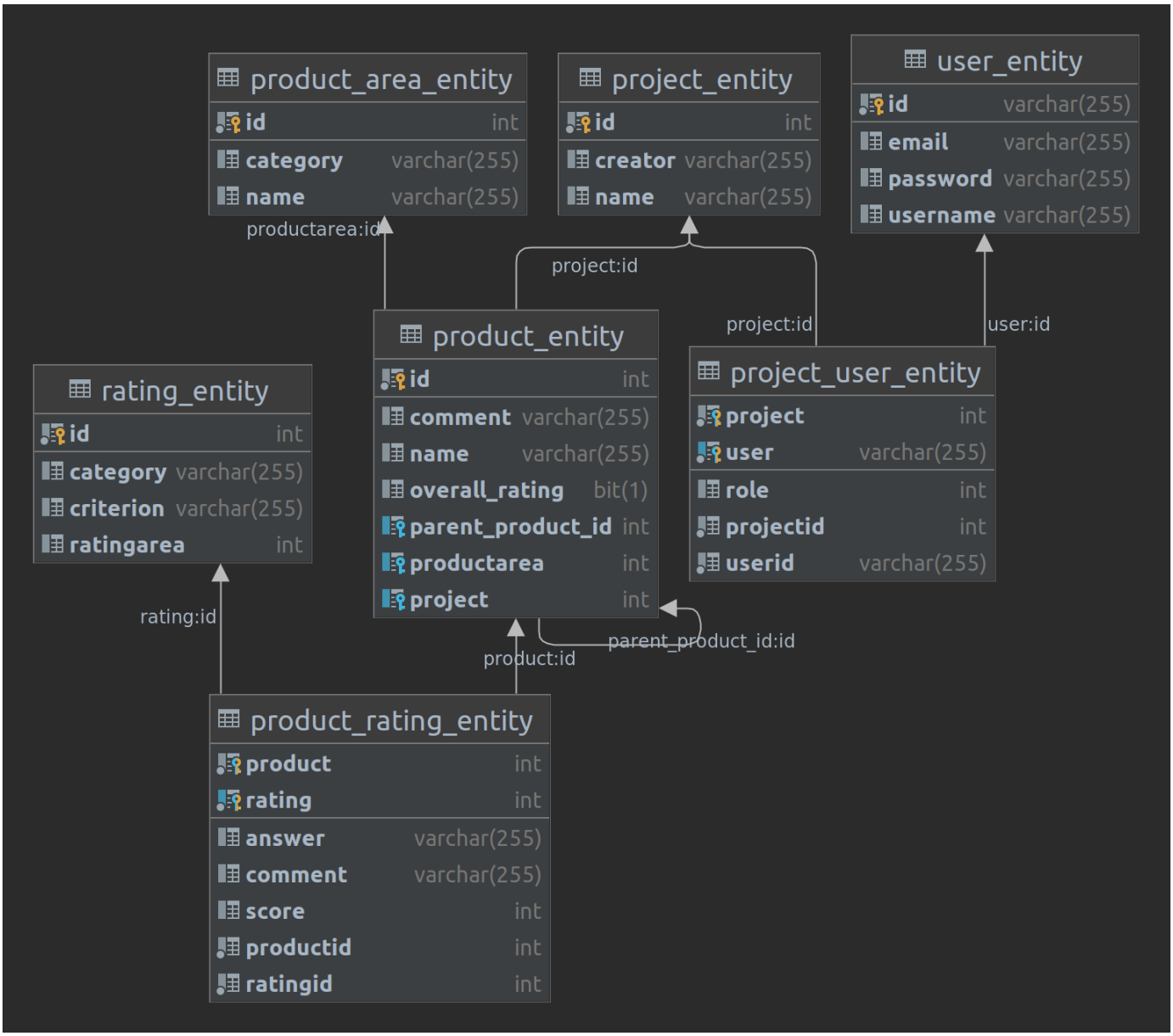
Software Architecture Diagrams

Runtime Components



The runtime components of the software consists of a frontend, backend and the database service. These three components are fully containerized with docker. The communication between front- and backend takes place over a REST API, which definition can be found below. In

Data Model



In this chapter we will focus on the data model of the database. It's easy to see that the Entity classes and the structure of the database are alike since these classes are used to fill data into the database.

which is a foreign key to already existing product id.

The **product_rating_entity** table links answers, comments and scores to rating from **ratings_entity** table and links both to a product. So changes on the answers would not affect a rating entity itself but product_rating entity.

Backend

The Backend is built in a Microservice Architecture style with Spring Boot, which are listed as follows:

Service	Description
ProductArea Service	Handles the product areas. You can get listed all the product areas or create new product areas here
ProductRating Service	Handles the ratings for products. You can create product ratings, update product ratings or find the rating for a specific product here.
Product Service	Handles the products. You can for example create & delete products or update & find products by the ID here.
Project Service	Handles the projects. You can create projects, find & update projects or get shown all projects here.
Rating Service	Handles the ratings. You can get all ratings here or get a rating for a specific rating area.
	Handles the user management, like create or delete a user, finding a

ProductDto	progressEconomic, List ratings, List productVariations, String comment, List resources	product or a product variation
ProductAreaDto	int id, String name, String category	describes Area of Product (see ProductDto)
ProductRatingDto	int ratingID, RatingDto rating	contains rating of a Product (see ProductDto)
RatingDto	int id, String criterion, String category, RatingArea ratingArea	explicit Rating of a Product (see ProductRatingDto)
ResultDto	int productID, String productName, List ratings, ScoreDto[] scores, int[] counts	contains "analyze" results
ScoreDto	Score score, int count	contains explicit score data (see ResultDto)

Frontend

React

The frontend is built with the React framework, a javascript framework for building web applications. The root location for the frontend source code is `/frontend/src` . This directory has the following contents:

Subdirectory	Description
`App.jsx`	App.jsx is the root Component of our React application. It mainly acts as a container for all other pages of our application.

update the application store. Further information about Easy Peasy can be found on their webpage.

UI Framework

We used Chakra UI to implement the above mentioned components in frontend. Chakra UI is a component library which provides React components using Chakra UI's default CSS styles. It offers basic building blocks such as Buttons, TextAreas and Flex boxes. Further information about Chakra UI can be found here.

Technology Stack

automates building, testing and the development.

6. **Other development tools**

- *Docker Engine*: Docker is for isolation of an application using containers.
  Docker CLI: It is the Command Line Interace using Docker.