Low Level Design for Lead API

1. General Information

Purpose and Scope

The document includes a solution on implementation of Lead APIMicroservice to address CRUD operations and holds lead specific data.

The solution is described in such detail that is possible to understand the business value, functionality and make high level effort estimation.

Any part of the proposed design may subject to change implementation.

Input

This document is prepared based on the HLD referenced and sent task by Emil Frey Digital d.o.o..

Terminology

Term/Concept	Abbreviation	Definition

Revision Information

Revision	**Date**	**Signature**	**Comments**
V1	23.03.2023	Irem AKTAS	Initial Document has been created

Patent Aspects

The solution does not include any patentable part.

2. Technical Solution

This solution proposes the design of Lead Micro-service which exposes basic APIs (CRUD).

The base API path should be **/efd/api/v1/lead**

Lead API is responsible for create, update, delete lead and search with first name and last name options of lead in Postgres DB.

Entity Model contains Lead and Exception Model. A view of the domain model can be seen in the following logical View – Domain Model Diagram.

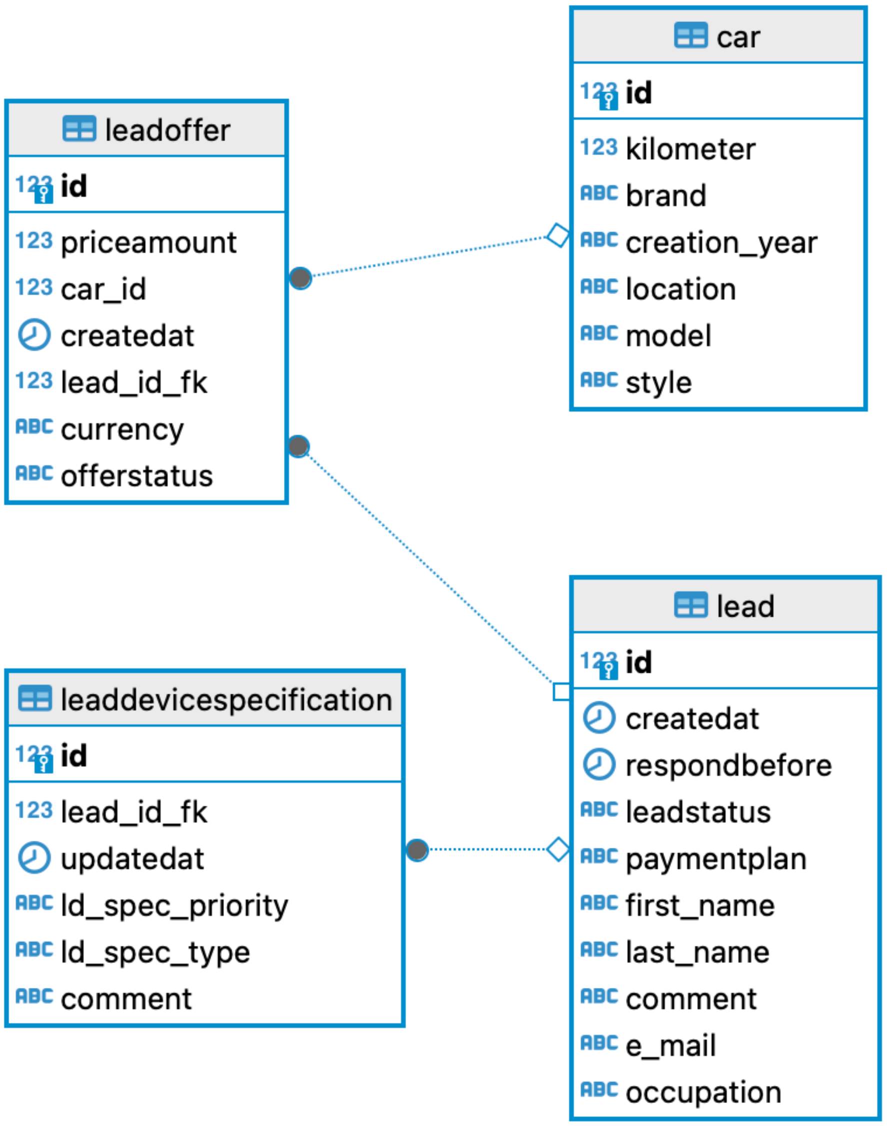


Figure 1 Lead API Logical View, Entity Domain Model

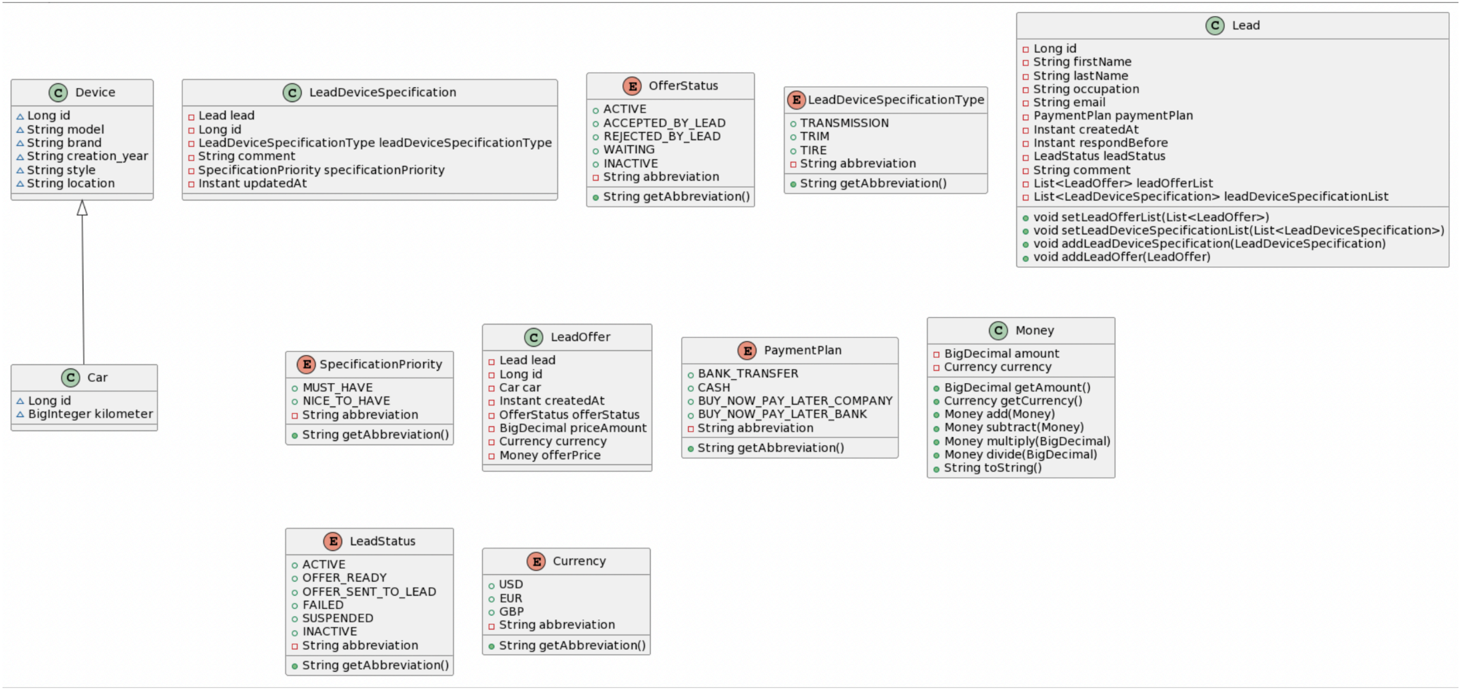
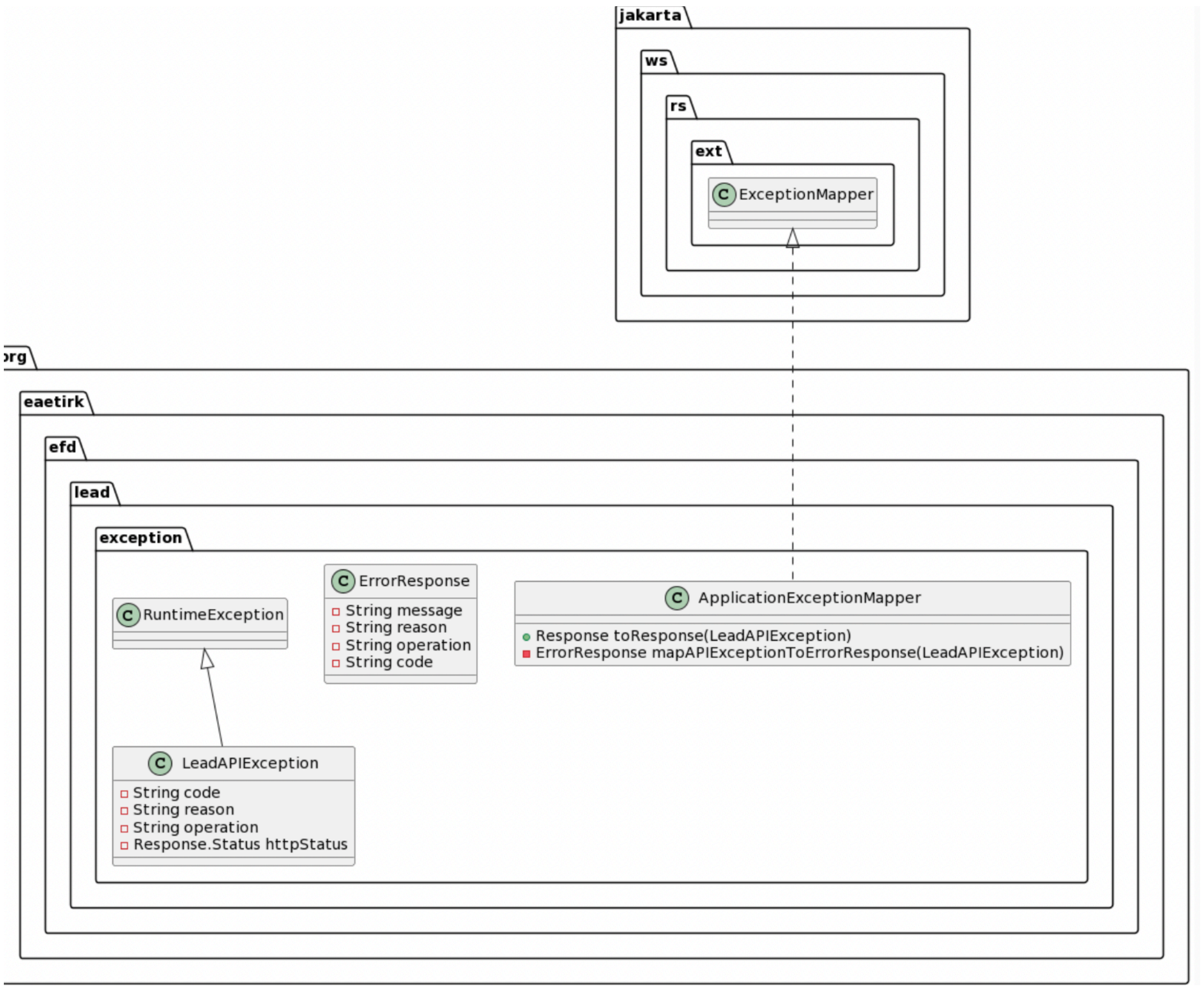


Figure 2-3 Lead API Entity Class Model



3. Technology Stack

Quarkus IO should be used as a development Framework.

Hibernate ORM and Panache Repository are used to inherit entity management methods which are inherited by PanacheRepositoryBase.

RESTEasy is used to expose REST APIs.

Quarkus SmallRye OpenTracing is used for Log Tracing.

SmallRye Fault Tolerance is used for Resilient Management.

SmallRye OpenAPI is used for API documentation : http://localhost:8082/q/swagger-ui/

Rest Assured is used for unit testing.

4. Entity Model Use Case

Lead Offer is model to represent offer to Lead. Each lead offer has price and and car specification. Car derives from Device.

Lead Device Specification is created to keep Lead's specifications and desires related to the Car as a comment and type.

Lead can have more offers and specify desired car specifications.

Each offer has its status to keep lead and company relationship.

5. Sample Lead APIs

Create Lead

POST : http://localhost:8082/efd/api/v1/lead

Update Lead

PUT: http://localhost:8082/efd/api/v1/lead/{{lead_id}}

If User wants to chance Offer, LeadSpecification, Offer ID and/or LeadDeviceSpecification ID should be given in the update request

Get Lead by ID

GET:http://localhost:8082/efd/api/v1/lead/{{lead_id}}

Get Leads

GET:http://localhost:8082/efd/api/v1/lead

Search Lead by First Name

GET: http://localhost:8082/efd/api/v1/lead?first_name=Irem

Search Lead by Last Name

GET:http://localhost:8082/efd/api/v1/lead?last_name=Aktas

Search Lead by First and Last Name

GET:http://localhost:8082/efd/api/v1/lead?first_name=Irem&last_name=Aktas

Filter Lead contains character in First Name or in Last Name

GET: http://localhost:8082/efd/api/v1/lead/contains?first_name=Ir&last_name=Ak

Filter Lead contains character in First Name

GET:http://localhost:8082/efd/api/v1/lead/contains?first_name=Ir

Filter Lead contains character in Last Name

GET:http://localhost:8082/efd/api/v1/lead/contains?last_name=Ak

In The project folder you can find postman collection and APIs sample usages.