ASDs

Data design

Product description

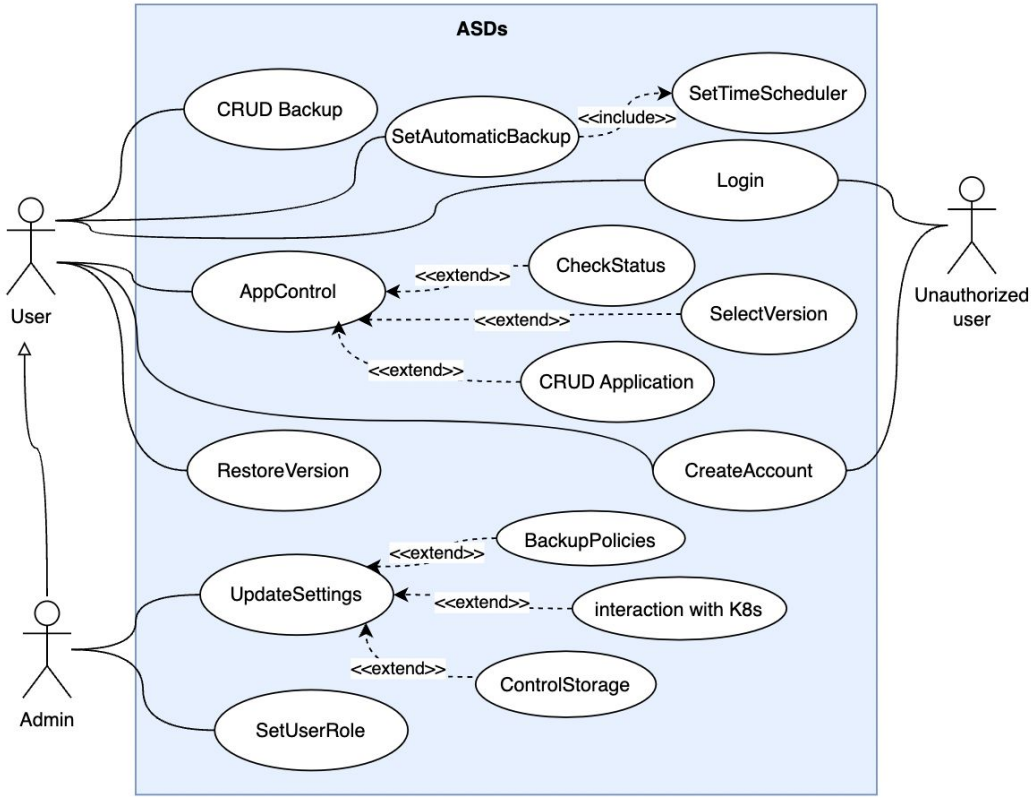
An application continuity service dynamically backs up and restores the state of applications running within the framework on K8s. The service uses the DSL of the framework to determine the relevant application state and its backup policies. Developers of an application are able to backup and restore a specific version from images and relevant state from the service. The application state is stored externally on the given S3-compatible object storage.

Team: Gorelyi Mikhail, Lukashin Daniil, Derezovskiy Ilya, Sigal Lev

Repo: https://github.com/gorelyi-code/advanced_software_designers

Report: a link to this slides within project repo/doc storage

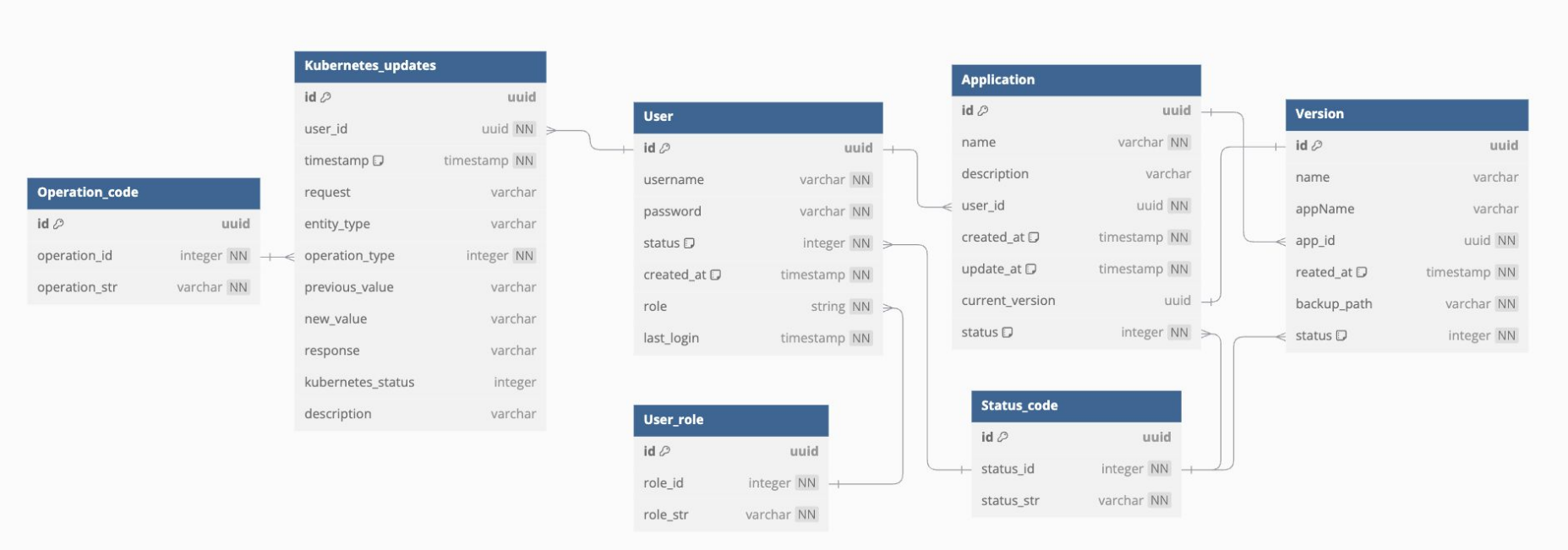
Use case diagram



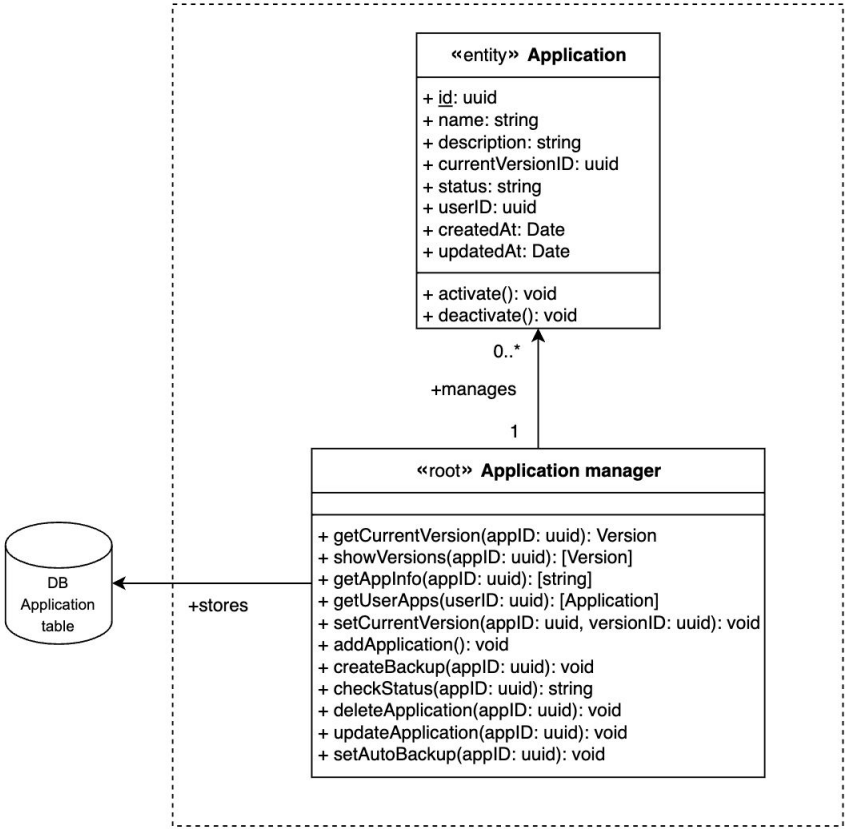
CRUD: create, read, update, and delete

Textual description in github:
https://github.com/gorelyi-code/advanced_software_designers/blob/main/Use%20case%20diagram/main_scenarios.md

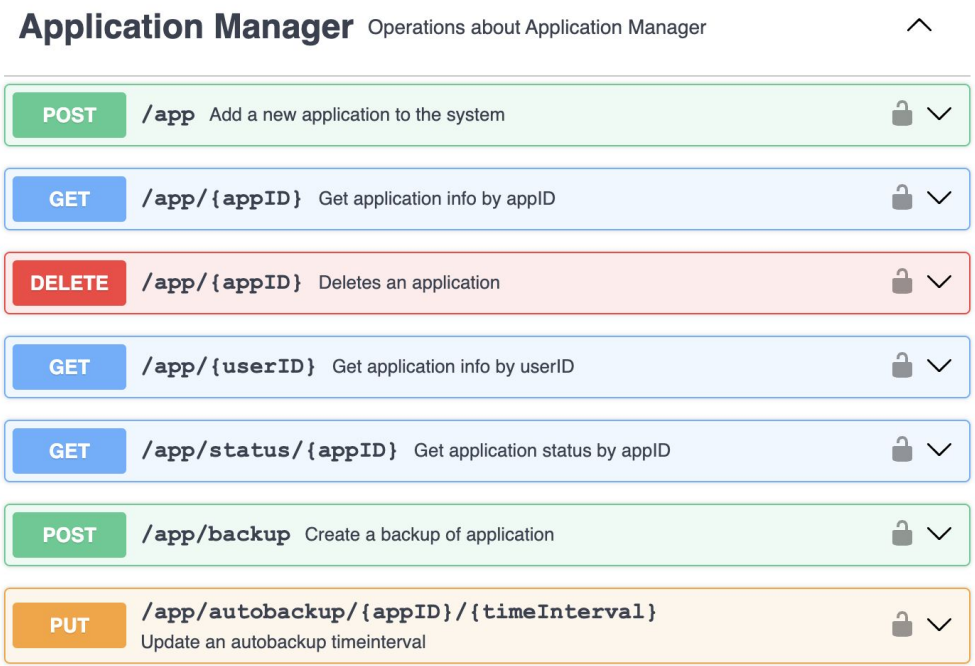
Physical schema



Aggregate Application Manager

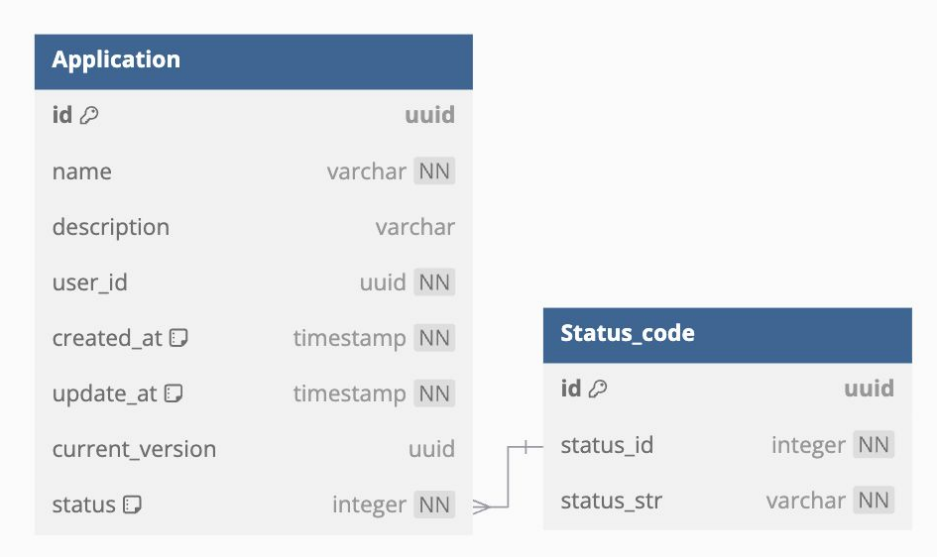


API usage Application Manager



https://github.com/gorelyi-code/advanced_software_designers/blob/main/hometasks/task_10/src/Application-manager-api.yaml

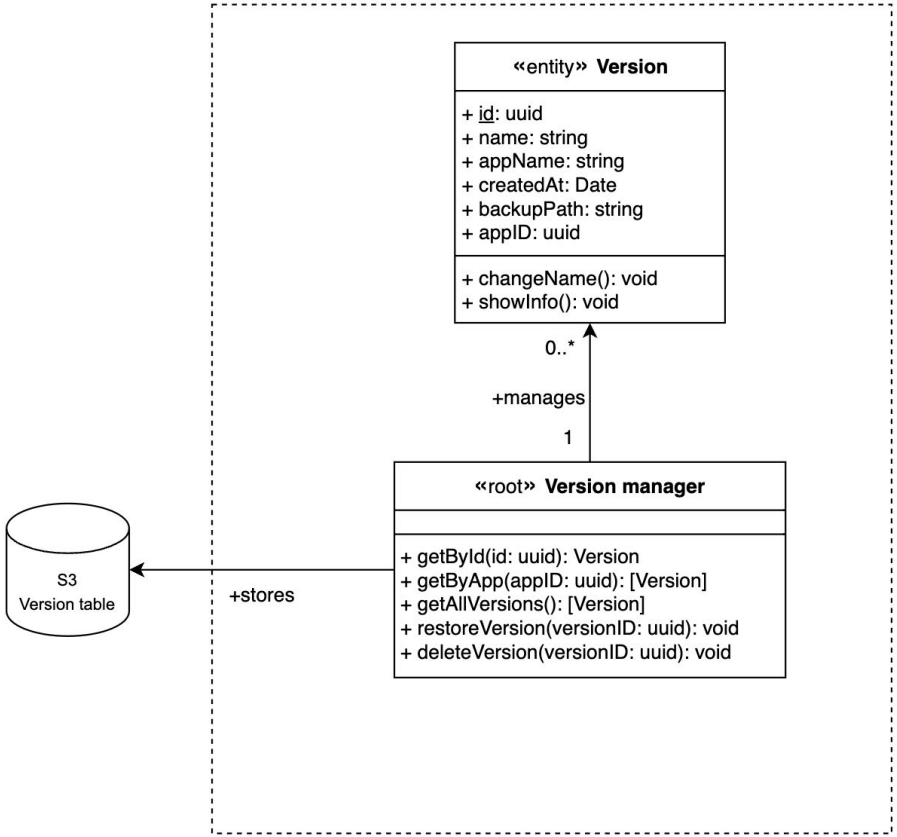
Physical schema for Application Manager



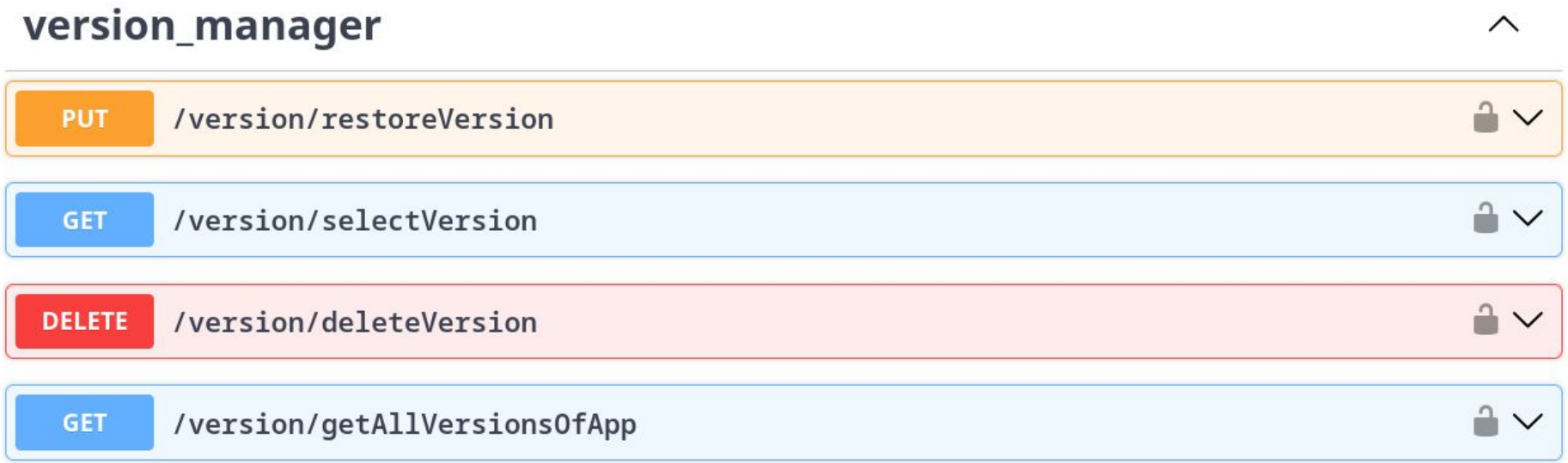

```
CREATE TABLE "Application" (
  "id" uuid PRIMARY KEY,
  "name" varchar UNIQUE NOT NULL,
  "description" varchar,
  "user_id" uuid NOT NULL,
  "created_at" timestamp NOT NULL DEFAULT (now()),
  "update_at" timestamp NOT NULL DEFAULT (now()),
  "current_version" uuid,
  "status" integer NOT NULL DEFAULT 0
);
```

```
CREATE TABLE "Status_code" (
  "id" uuid PRIMARY KEY,
  "status_id" integer NOT NULL,
  "status_str" varchar NOT NULL
);
```

Aggregate Version Manager

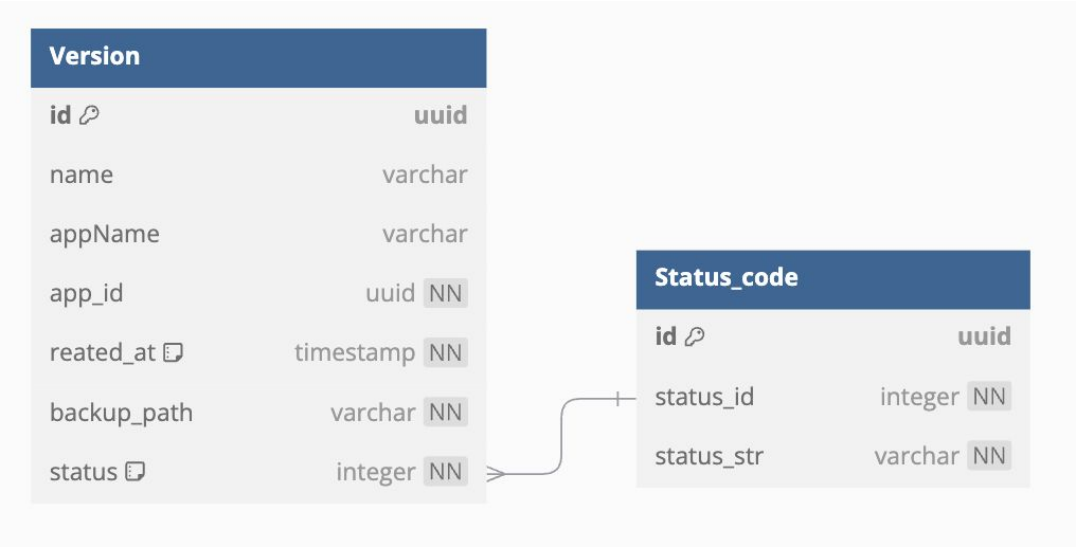


API usage Version Manager



https://github.com/gorelyi-code/advanced_software_designers/blob/main/hometasks/task_10/src/Version-manager-api.yaml

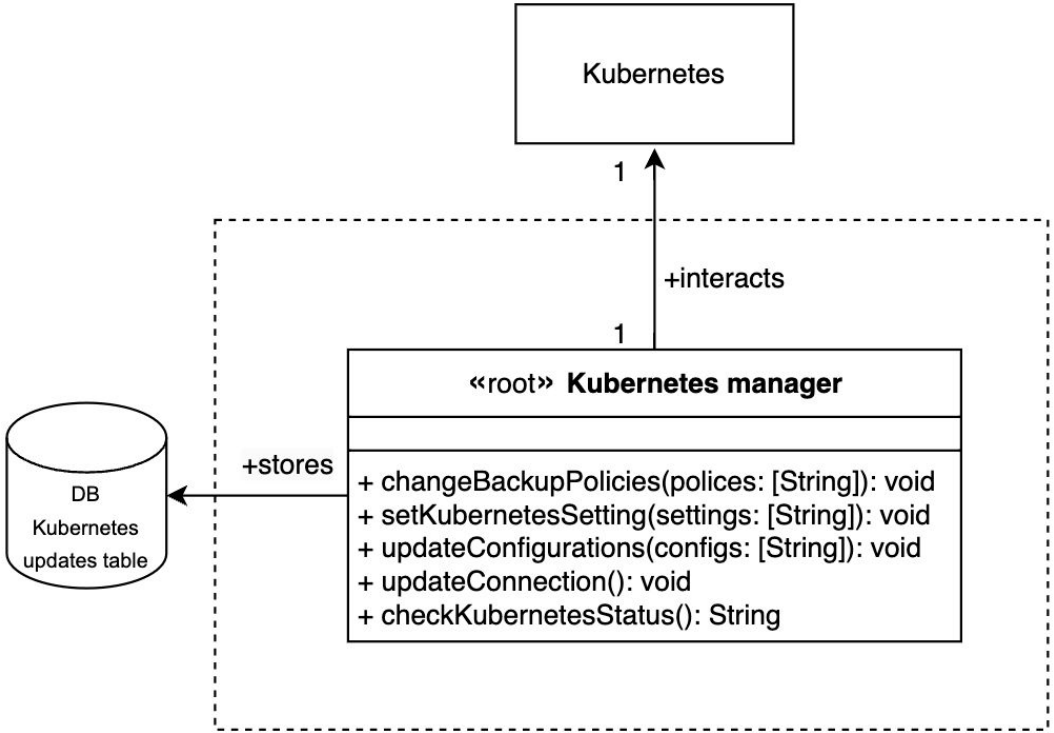
Physical schema for Version Manager



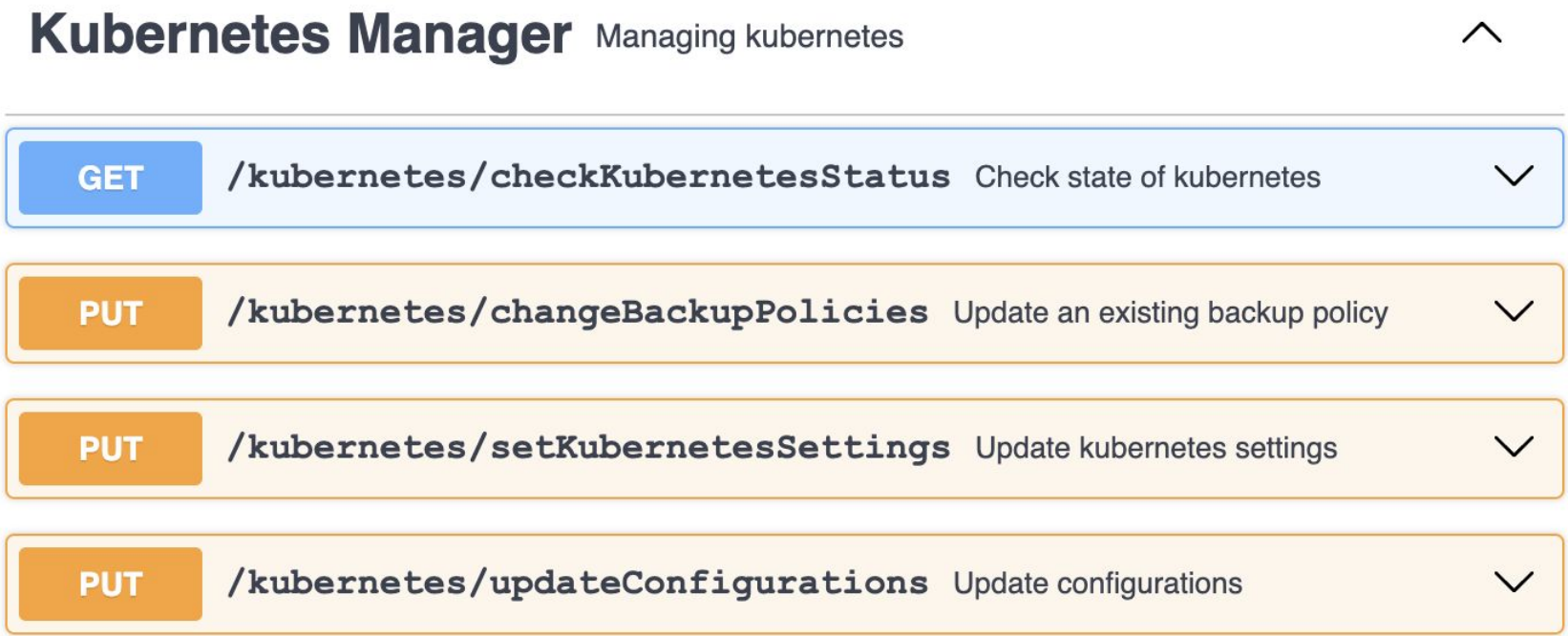

```
CREATE TABLE "Version" (
  "id" uuid PRIMARY KEY,
  "name" varchar,
  "appName" varchar,
  "app_id" uuid NOT NULL,
  "reated_at" timestamp NOT NULL DEFAULT (now()),
  "backup_path" varchar NOT NULL,
  "status" integer NOT NULL DEFAULT 0
);

CREATE TABLE "Status_code" (
  "id" uuid PRIMARY KEY,
  "status_id" integer NOT NULL,
  "status_str" varchar NOT NULL
);
```

Aggregate Kubernetes Manager

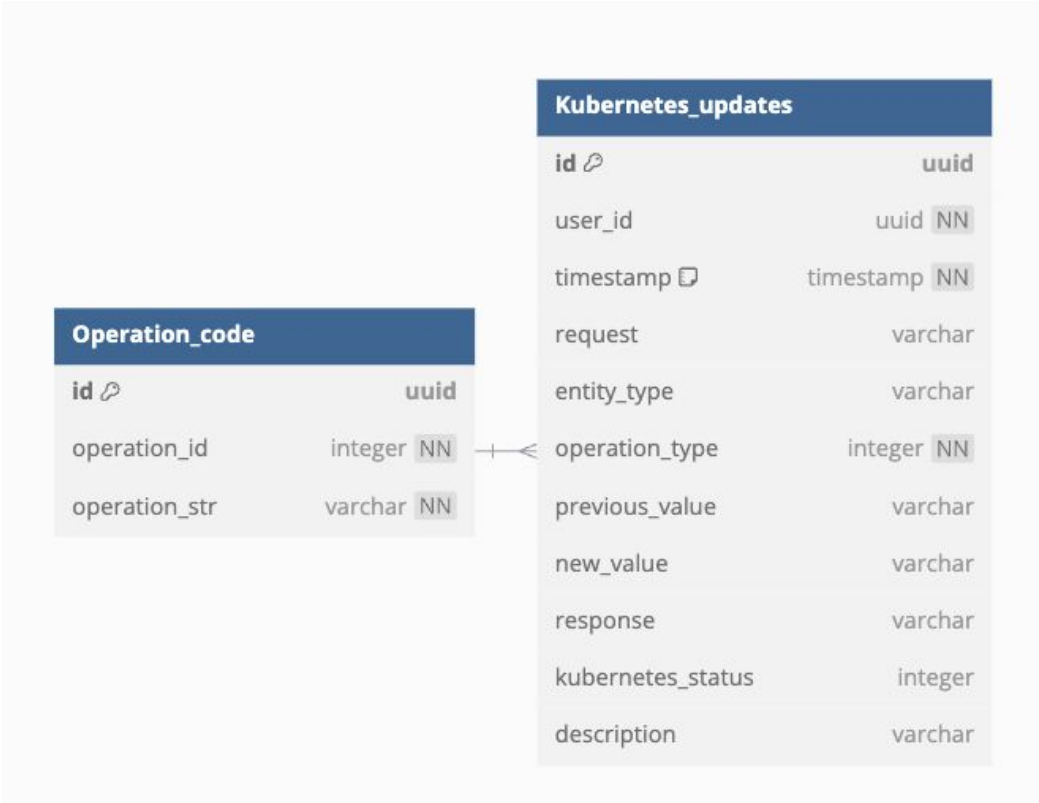


API usage Kubernetes Manager



https://github.com/gorelyi-code/advanced_software_designers/blob/main/hometasks/task_10/src/kubernetes_manager.yaml

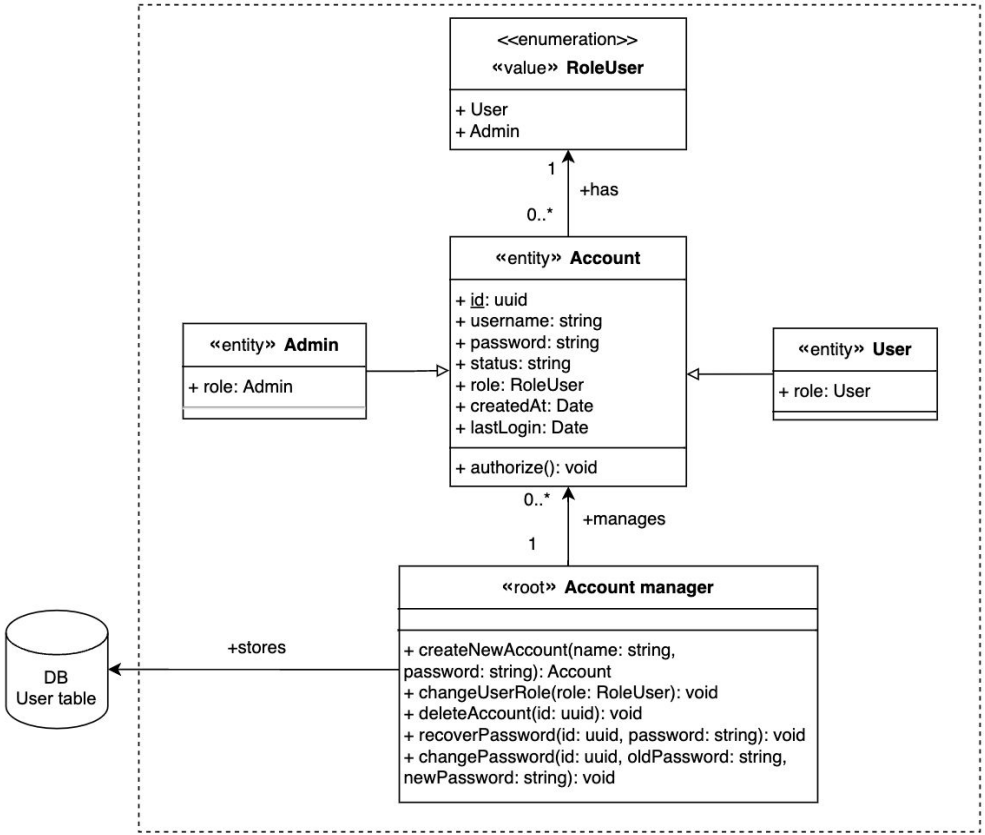
Physical schema for Kubernetes Manager



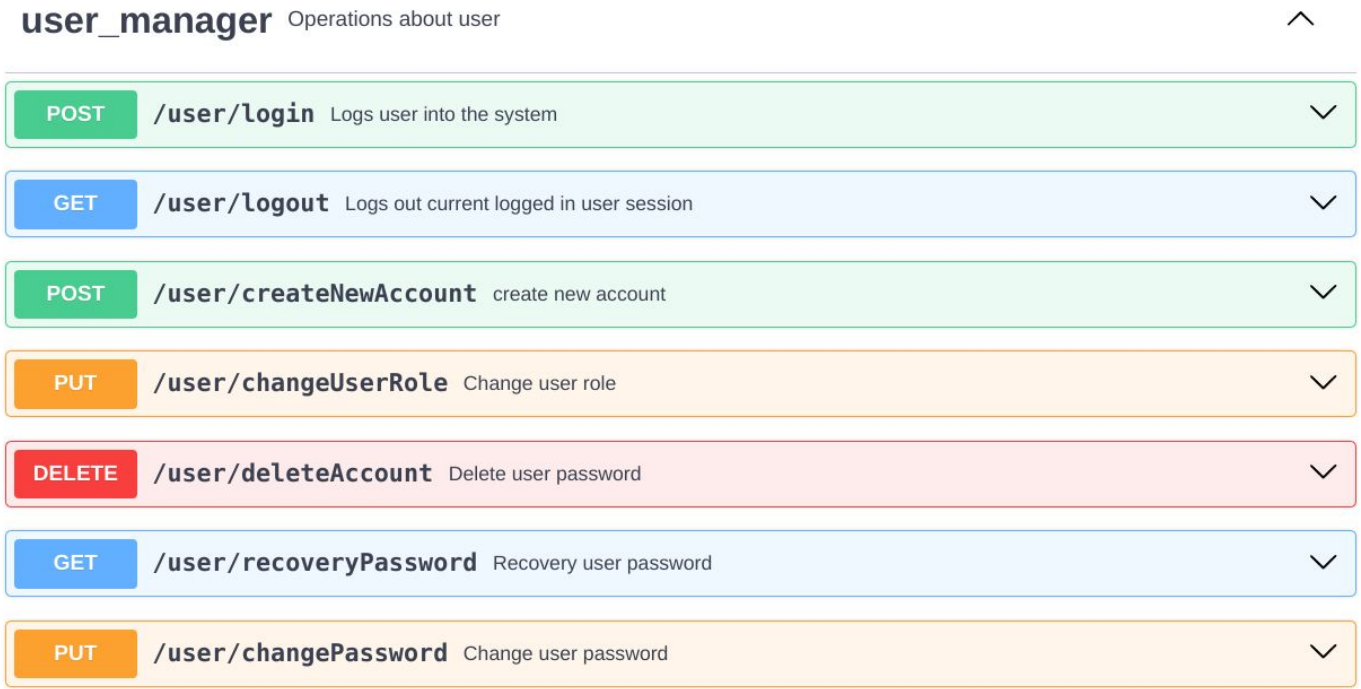

```
CREATE TABLE "Kubernetes_updates" (
  "id" uuid PRIMARY KEY,
  "user_id" uuid NOT NULL,
  "timestamp" timestamp NOT NULL DEFAULT (now()),
  "request" varchar,
  "entity_type" varchar,
  "operation_type" integer NOT NULL,
  "previous_value" varchar,
  "new_value" varchar,
  "response" varchar,
  "kubernetes_status" integer,
  "description" varchar
);

CREATE TABLE "Operation_code" (
  "id" uuid PRIMARY KEY,
  "operation_id" integer NOT NULL,
  "operation_str" varchar NOT NULL
);
```

Aggregate Account Manager

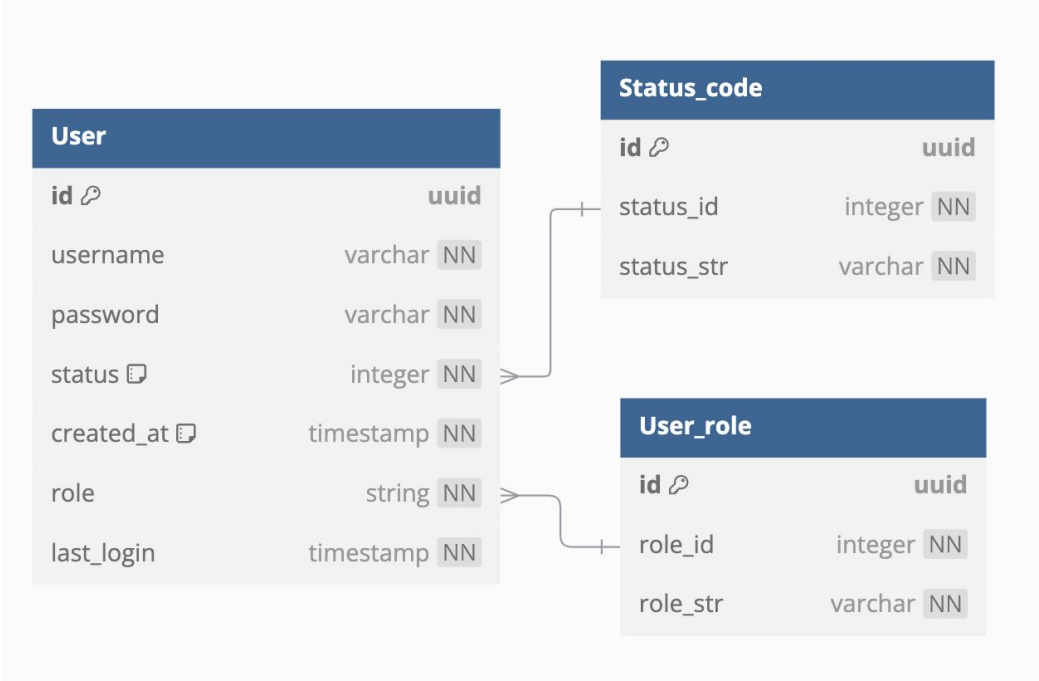


API usage Account Manager



https://github.com/gorelyi-code/advanced_software_designers/blob/main/hometasks/task_10/src/Account-manager-api.yaml

Physical schema for Account Manager




```
CREATE TABLE "User" (
  "id" uuid PRIMARY KEY,
  "username" varchar UNIQUE NOT NULL,
  "password" varchar NOT NULL,
  "status" integer NOT NULL DEFAULT 0,
  "created_at" timestamp NOT NULL DEFAULT (now()),
  "role" string NOT NULL,
  "last_login" timestamp NOT NULL
);

CREATE TABLE "User_role" (
  "id" uuid PRIMARY KEY,
  "role_id" integer NOT NULL,
  "role_str" varchar NOT NULL
);

CREATE TABLE "Status_code" (
  "id" uuid PRIMARY KEY,
  "status_id" integer NOT NULL,
  "status_str" varchar NOT NULL
);
```

Team work

Gorelyi Mikhail - Kubernetes Manager

Lukashin Daniil - Account Manager

Derezovskiy Ilya - Application Manager

Sigal Lev - Version Manager