Is it Poison? – Specification

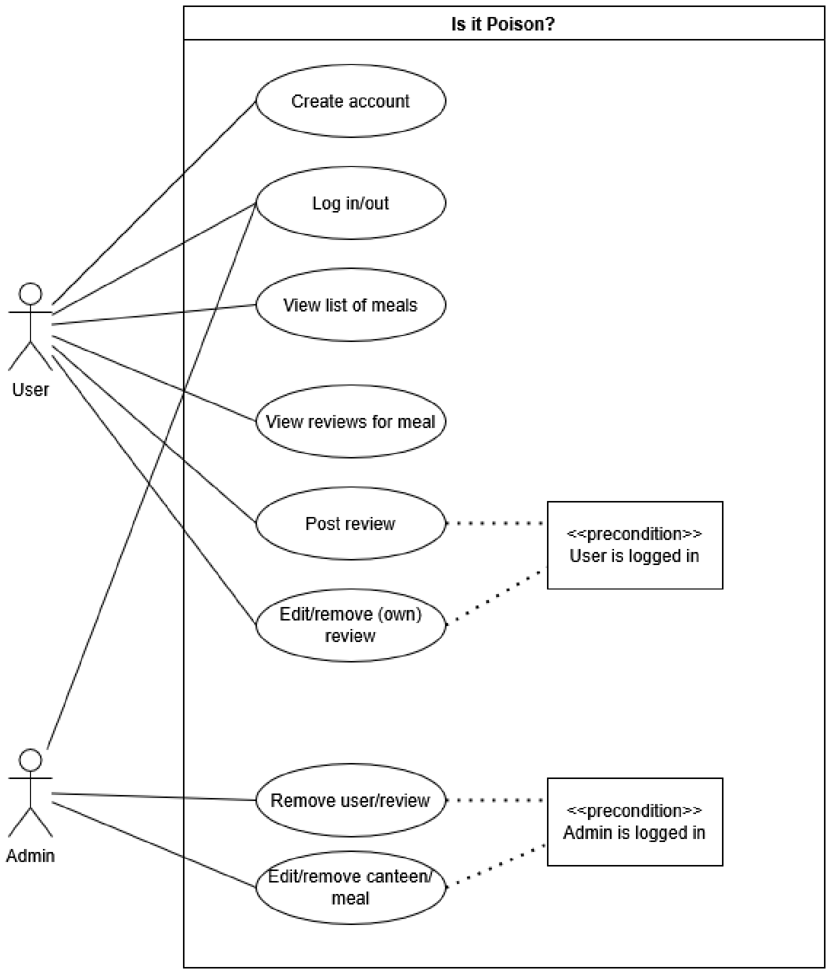
Is it Poison? is a web application that allows users to read and post reviews for meals served in university canteens (Eat & Meet, FreeFood, FaynFood, FiitFood). The target group are university students and staff who eat at the canteens and tend to choose what they want to eat beforehand. The application will provide additional information from other users that they can use to make their decision.

Requirements

Roles:

- **User** – Can view a list of meals with various filters (only specific canteen, only meals served today…) and sortings (alphabetically, by date served, by average rating). Can view individual reviews for each meal. If logged in, can also post reviews (stars only or stars + text), and later edit or remove them from their profile page.
- **Admin** – Can remove inappropriate reviews/users. Can perform CRUD operations on individual meals and canteens in the database.

Use case diagram:



Data model

Entities:

Canteen

- Attributes:
    - id
    - name – unique

Canteen_detail

- Attributes:
    - canteen_id
    - location – string
    - monday_open – time (null means that the canteen is not open that day)
    - monday_close – time (null means that the canteen is not open that day)
    - …
    - sunday_open – time (null means that the canteen is not open that day)
    - sunday_close – time (null means that the canteen is not open that day)

Meal

- Attributes:
    - id
    - name – unique
    - canteen_id – which canteen served this meal
    - last_served – date when this meal was last served
    - uploaded – date when this meal was added to database
- Note: If same meal is served in multiple canteens, these are separate Meals with same name

Weekly_meal

- Weekly_meals is a list of all meals available for each day of the current week
- The list is updated at the beginning of a week
- Attributes:
    - meal_id
    - weekday – int between 0 and 6

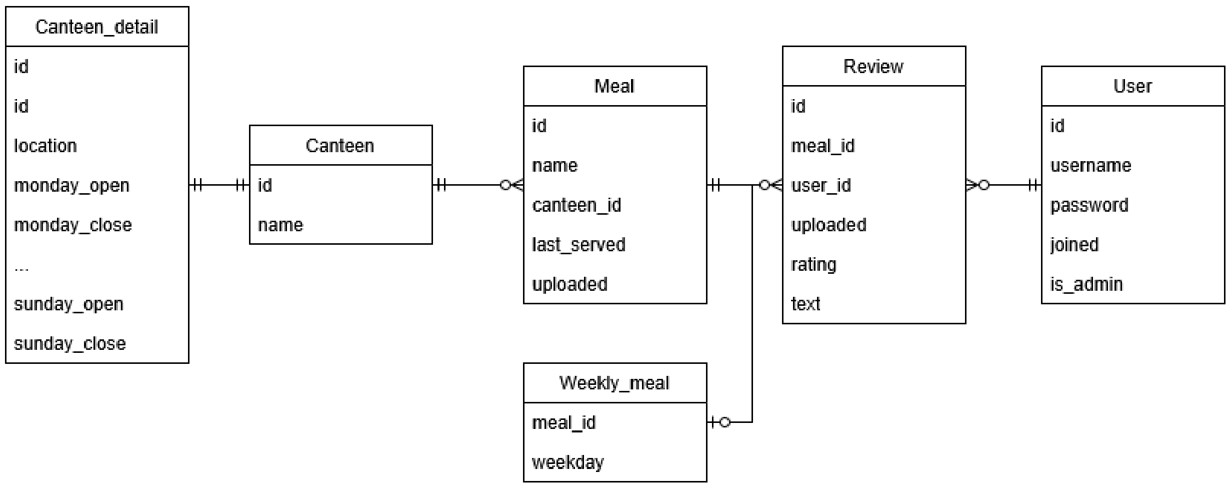
User

- Attributes:
    - id
    - username – unique
    - password – hash + salt
    - joined – date when the user registered
    - is_admin – bool

Review

- Attributes:
    - id
    - meal_id – meal that this review is associated with
    - user_id – user that posted the review
    - uploaded – when the meal was uploaded
    - rating – stars rating between 0 and 10 (10 is 5 stars, 9 is 4 and half stars…)
    - text – optional text accompanying the stars rating

Entity-relationship model:



Architecture

- Client-server, SPA (Single Page Application)

Technological requirements

- Client-side: React 19, TypeScript, HTML5, CSS3, React Bootstrap
- Server-side: node.js 22, express.js 5, TypeScript
- Database: PostgreSQL 17
- Client-server interface: Rest API
- Hosting: render.com?
- Supported browsers: Chrome, Firefox, Edge?

Future work

- Logged in users can like reviews and all users can sort reviews by number of likes.
- Logged in users can report inappropriate reviews. Admin has a page with reports, where they can resolve them.
- Logged in users can submit proposals for adding new meals that are missing in the database. Admin has a page with proposals, where they can resolve them.
- Logged in users can choose to delete their own account.

Time schedule

- Week 1-3
    - Create scaffolding for client and server app that utilizes required technologies
    - Make server serve SPA client to browser
    - Connect server to database
    - Create database schema
- Week 4
    - Create client component structure
        - Add all components displayed when not logged in and with hardcoded values
        - There will be enough CSS for layout of components but nothing else
- Week 5
    - Create server-side API for functionality when not logged in and connect it with client
- Week 6
    - Add components and mock functionality with hardcoded values when logged in as either User or Admin
- Week 7
    - Add login functionality (both client and server-side)
- Week 8
    - Add API and functionality for User role when logged in
- Week 9 (Beta version)
    - Add the rest of CSS
    - Set up hosting
- Week 10
    - Add functionality for Admin role
- Week 11 (Final version)
    - Finishing touches