Specification - Capacity Planner

The goal of this web application is to provide a lightweight tool for homework planning. The primary target users are students who need to plan and track of their homework (including historical ones). Additionally, other students can see other students' homeworks and add them to their list of homeworks, if that homework is listed as public.

Functional requirements

Roles

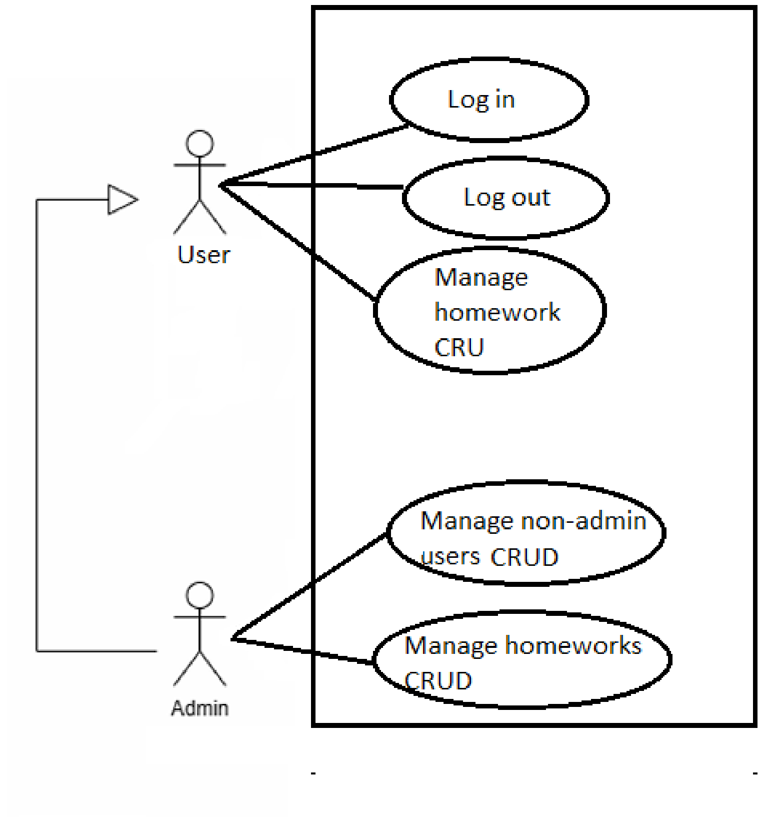
The application distinguishes two roles:

- **User –** a student. They want to see and manage their homeworks.
- **Admin** - manages users and homeworks. Admin accounts are managed at the database level.

Use Case Diagram

Notes:

- **CRUD** = Create, Read, Update, Delete
- **"Log in"** and **"Log out"** use cases are not user goals, but they are included in the diagram for completeness. All other use cases require the corresponding role (primary actor) login to complete (they have the precondition "[role] is logged in").
- **Delete:** Some delete functionalities for the admin role may be implemented as a "soft" delete due to incoming references. This will be further refined during implementation.


Log in
Log out
Manage homework CRU
User
Manage non-admin users CRUD
Manage homeworks CRUD
Admin

Data model

The following conceptual data model contains the entities, attributes and relations.

Entities and attributes

Attributes that are self-describing are not mentioned explicitly in this section.

User

- A user is identified uniquely by personal id.
- Constraints & rules:
    - An employee has exactly one non-admin account and at most one admin account associated.

Account

- An account used for authentication to the application. The username is unique.
- Attributes:
    - *password:* Stored as a hash value
    - *is_admin*: True in case of admin account, false in case of non-admin account
- Constraints & rules:
    - Each account belongs to exactly one employee.

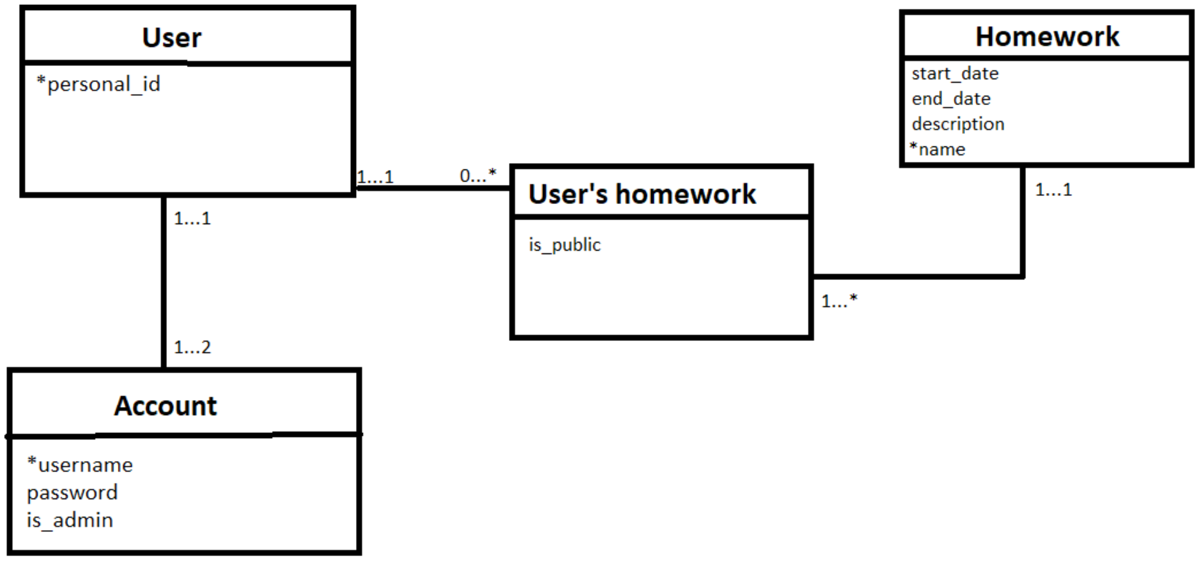
Homework

- Homework identified by its unique name.
- Attributes:
    - *start_date:* The date when a given homework is added to the list of current homeworks
    - *end_date:* The date when a given homework is done, can be NULL
    - description: Contains what subject this homework belongs to and a brief description of what the student is supposed to do in this homework

User's homework

- An assignment of a user to a homework as a user who needs to do this homework.
- Attributes:
    - *start_date:* The date when a given homework is added to the list of current homeworks
    - *end_date:* The date when a given homework is done, can be NULL.

ER model



Architecture

- The application will be based on the client-server architecture and it will use the SPA (Single Page Application) approach.

Technological requirements

- Client-side: React 18, JavaScript, HTML5, CSS3
- Server-side: node.js 23, express.js 4.21.2, JavaScript
- Database: PostgreSQL 16
- Interface client - server: Rest API
- Hosting: render.com
- Supported browsers: Chrome, Firefox

Time schedule
Will get filled out next week, I don't know what am I even going to be doing exactly (never worke with frontend/backend before)

Week 4
- …

Week 5
- …

Week 6
- …

Week 7
- …

Week 8
- …

Week 9 (Beta version)
- …

Week 10
- …

Week 11 (Final version)
- …

Week 4
- Setup of the developmental environment, mock frontend

Week 5
- Api, backend

Week 6
- Work on databases

Week 7
- Authentification

Week 8
- Deployment

Week 9 (Beta version)
- End-to-end private homeworks without descriptions

Week 10
- homeworks with descriptions

Week 11 (Final version)
- publichomeworks