Issue Tracker Specification

This project focuses on creating a tool for effective task management and error tracking in teams. The goal is to enable teams to better coordinate their activities, record problems and track their solutions in real time. The main benefits of the system are increased transparency of work, better planning and the ability to quickly respond to complications that arise.

The application will support filtering and sorting tasks according to various criteria, such as status, priority, creation date or assigned user. Team leaders will be able to create projects and assign tasks to them, which will allow for better organization of work. The application will also provide project progress reports, including graphs and statistics. The application will support notifications, e.g. by email, that will alert users to changes in tasks (e.g. adding a new comment)

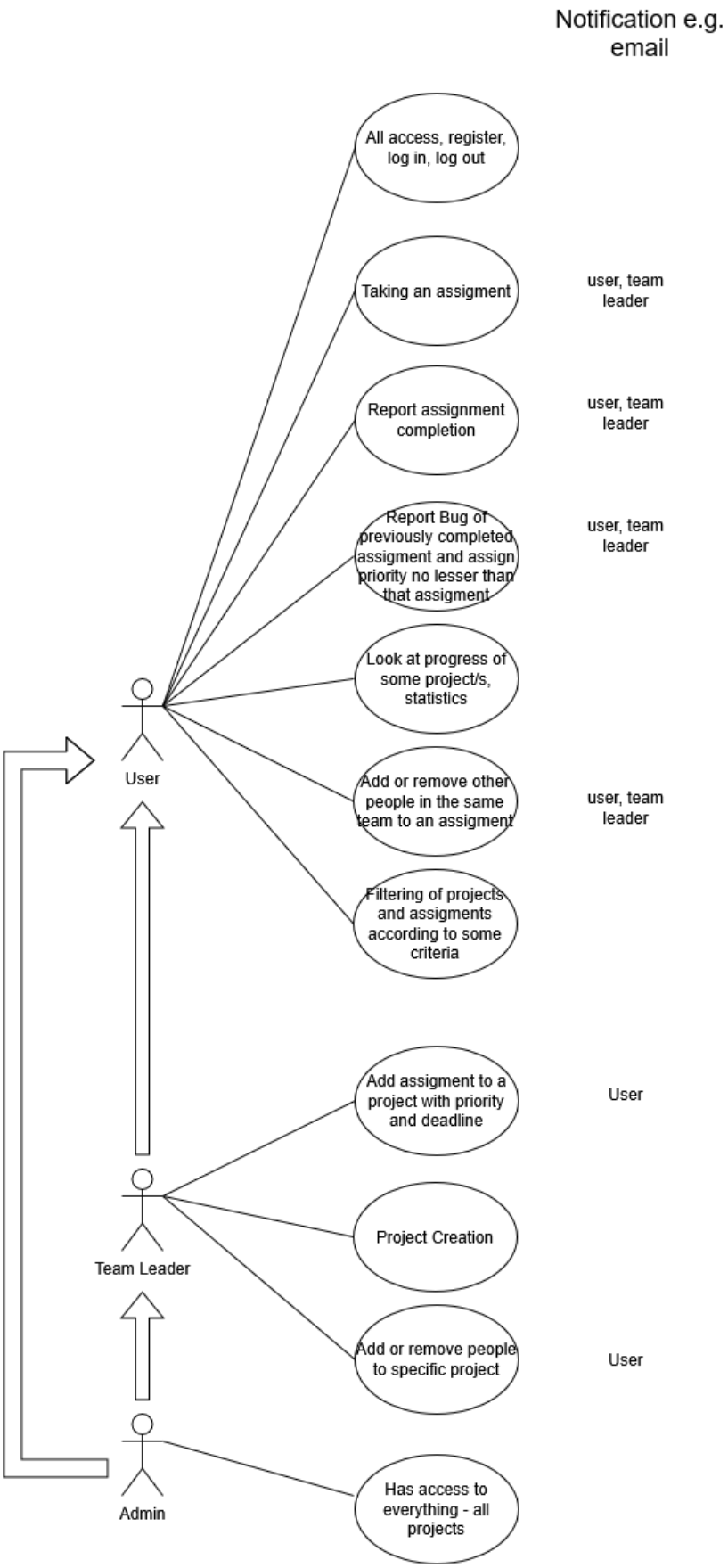
Functional requirements

Roles

The application distinguishes three roles:

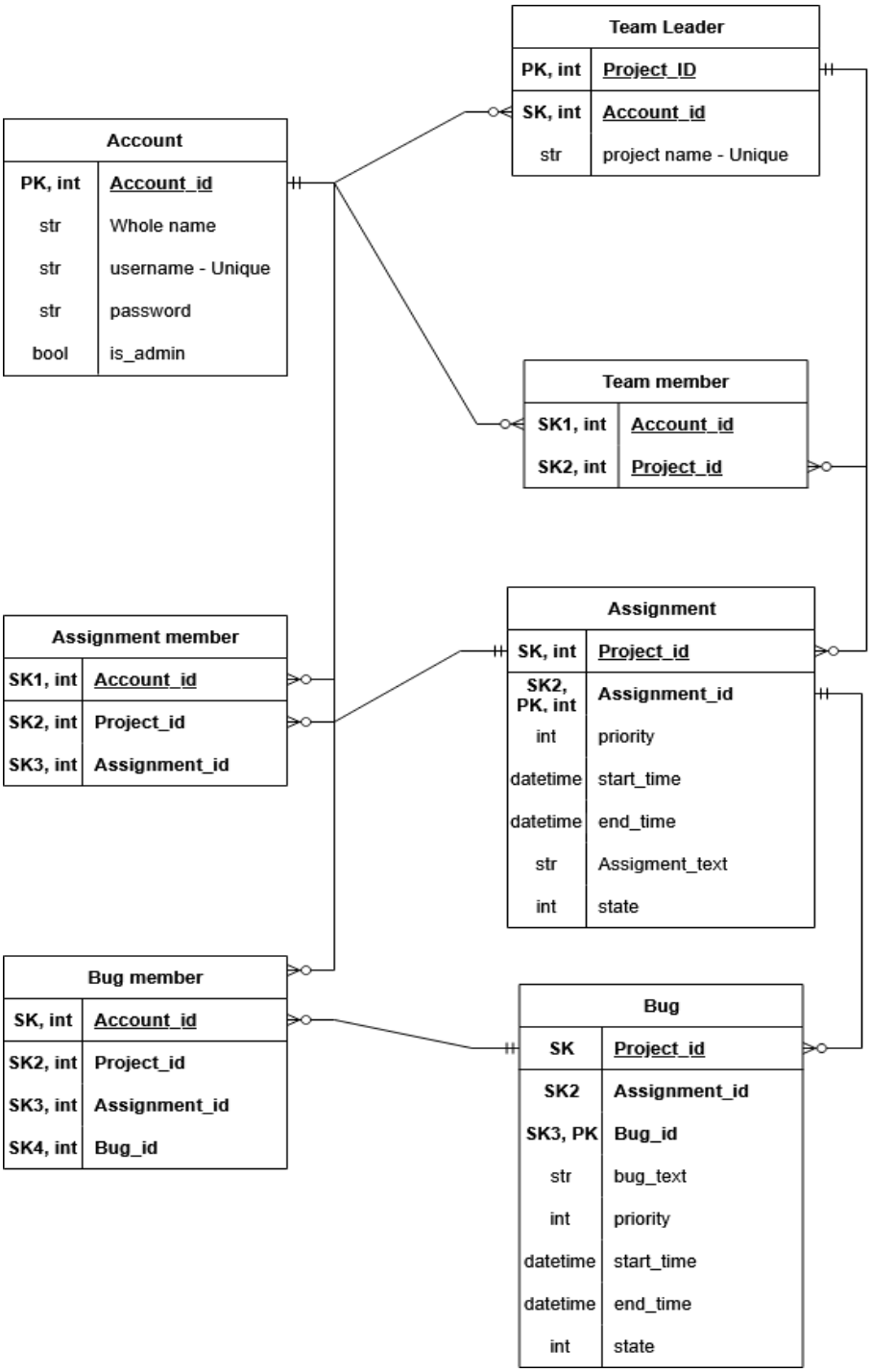
- **User** – User of the application. They want to see teams they have been included in ,team assignments, take them, filter, see progress, add other people in the team to an assignment, report completion, report bug...
- **Team leader** - their goal is to assign the team members to projects and to track current assignments by team / project. Team leaders manage the members of their team. Can do everything that user can do
- **Admin** – Has access to everything, all teams, all projects. Has access to all functions for user and team leader

Use Case Diagram:



- All the notifications happen according to action in use case. Other notification can be when assignment is close to deadline and so on ...

ER model:



ER Connection explanation:

-Every row of the table account with main account_id key can be 0 or multiple times in team Leader table as secondary key

-Every row of the table team leader with main leader_id key can be 1 time with it`s secondary key inside table account as primary key

-And so on... same connections between account to team member, assignment member, bug member and connections between team leader to team members, assignments, bugs

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: ???
- Hosting: render.com
- Supported browsers: Chrome, Firefox

Future work

Time schedule

Week 4

- First time installing React, Vite, getting my bearings, trying out various code,

Week 5

- Log in, log out, register implementation, PostgreSQL creation of all databases

Week 6

- Create draft of GUI, start of implementation for team leader – project creation with priorities, member assignment together with notification system

Week 7

- Finish team leader implementation, start user implementation, like taking an assignment, report assignment completion, report bug, notification system

Week 8

- Progress graphs and statistics, sorting function to criterias priority, state, team, members, start time, end time, finish notification system

Week 9 (Beta version)

- Finish GUI, Admin account – admin interface, finishing all tasks from previous weeks

Week 10

- ...

Week 11 (Final version)

- ...