# 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.

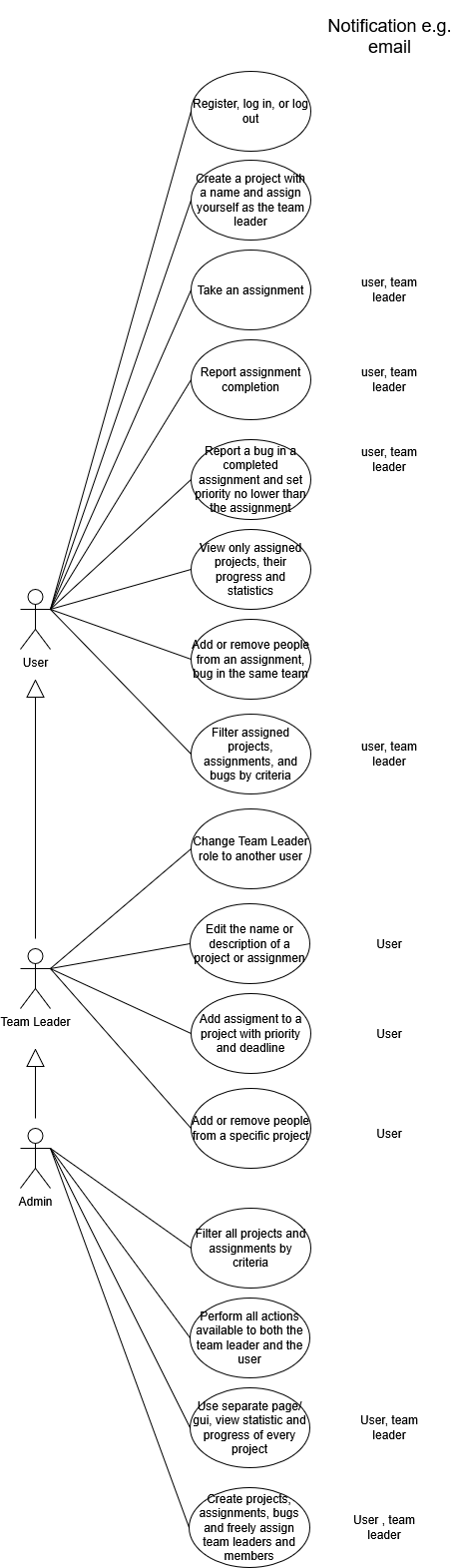
# Roles :

● **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

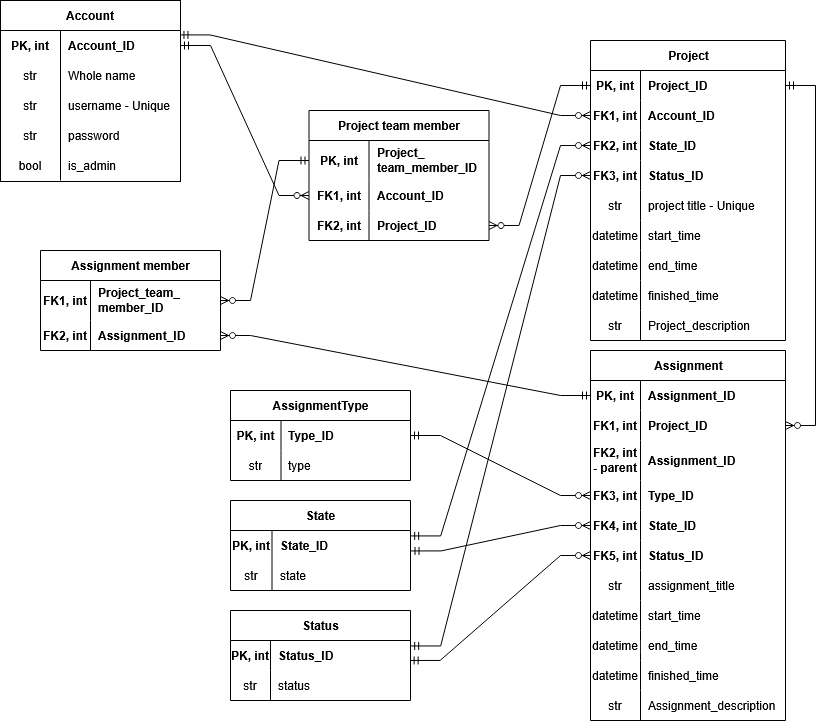
● **Admin** – Has access to everything, all teams, all projects. Has access to all functions for user and team leader

# Use Case Diagram:



* All the notifications happen according to action in use case.
* Time based notification as well - closer the deadline is, the priority is higher and email is sent
* Possibly one thread where it will constatly check database, update it and send notifications

# ER model:



# Data model explanation:

## Account:

* User is uniquely identified by primary key Account\_ID.
* Username is unique atribute, which is needed to distinguish different users at authorization
* Whole name attribute for entire name of the user
* Password for authorization
* Is\_admin bool value says if this account is admin accoun in case of truth
* Constraints & rules:
  + User has exactly one non-admin account and at most one admin
  + Admin account can not be team leader, member of any team or assignment member
  + One non-admin account (user) can be leader of multiple teams in relation with Project by Account\_ID
  + One non-admin Account (user) can be member of multiple teams in relation with Project team member by Account\_ID

## Project:

* Project uniquely identified with primary key Project\_ID
* Project\_title is also unique
* With foreign\_key as Account\_ID in relation with Account specifies Team Leader
* With foreign\_key as State\_ID specifies state of the project inside table State
* With foreign\_key as Status\_ID specifies status of the project inside table Status
* Start\_time specifies time when project was created
* End\_time specifies deadline of the project
* Finished\_time specifies time when Project was completed, otherwise it has empty string
* Project\_description gives discreption of the entire project and it`s goals
* Constraints & rules:
  + One Project has exactly one Account as Team Leader with Account\_ID
  + One Project can have multiple Project team member/s with Project\_ID
  + One Project can have multiple Assignment/s with Project\_ID
  + Project will have state according to difference between end\_time and start\_time, the lesser time remaining will determine more higher priority gradually. (low, low-middle, middle, higher, high). In any time one project will have one state with State\_ID.
  + Project will have status according to the start\_time, end\_time and finished\_time. If there is still time remaining between end\_time and start\_time it will be ongoing project. If there is no finished\_time and it is after deadline, it will be flagged as ongoing after deadline . The same will apply for finishing the project, it could be flagged as finished after deadline or finished in time. In any time project have one status with Status\_ID

## Project team member:

* Says which Account/s belongs to which Project/s
* Project team member uniquely identified as Project\_team\_member\_ID
* Foreign key Account\_id with connection to Account
* Foreign key Project\_id with connection to Project
* Constraints & rules:
  + One row of table of Project team member has exactly one row in Project with Project\_ID
  + One row of table of Project team member has exactly one row in Account with Account\_ID
  + Project team member can be member of multiple assignments in connection with Project\_team\_member\_ID in Assignment member – assignment/s must be from the same Project as Project team member of particular project with Project\_ID

## Assignment:

* Primary key Assignment\_ID uniquely specifies assignment
* Foreiqn key Project\_ID specifies which Project the Assignment belongs to
* Foreiqn key Assignment\_ID specifies which particular Assignment is his parent in case of state not being set to task inside AssignmentType - bug was reported (or maybe something else)
* Has foreiqn keys for Type\_ID, State\_ID and Status\_ID to know exact type, status and state from other tables
* Assignment\_title is short title text for the assignment
* Start\_time specifies time when Assignment was created
* End\_time specifies deadline of the Assignment
* Finished\_time specifies time when Assignment was completed, otherwise it has empty string
* Assignment\_text specifies string, actual assignment, what is needed to do and it`s goals
* Constraints & rules:
  + Only Team Leader can add first Assignment/s with foreign key Assignments\_ID set to NULL
  + One Assignment row with Project\_id has one row in Project – one assignment belongs to one exact project
  + One Assignment can have multiple Assignment member/s with Assignment\_ID
  + Created Assignemnt/s must have lesser or equal deadline (end\_time) of the Project
  + Project team members and Team Leader can bug report finished Assignment/s with foreiqn key Assignment\_ID related to original primary key Assignment\_ID. Deadline can be over the deadline of the project, but the time difference between start\_time and end\_time for fixing the bug must be lesser or equal to the time difference of the parent Assignment. In this case state inside Assignment type is bug and not task.
  + Priorities and status will work the same way as for Project
  + Project team member/s of and Team Leader can report a bug of an Assignment and create new Assignment where Type\_ID and foreiqn key to parent Assignment\_ID will be set appropiately

## Assignment members:

* Connection between Project team members and Assignments
* Has two foreiqn keys: Project\_team\_member\_ID, Assignment\_ID
* Says which Project team member/s belongs to which assignment/s
* Constraints & rules:
  + One Assignment member row has one row in Project Team member by Project\_team\_member\_ID.
  + One Assignment member must already be a Project team member before it can be Assignment member
  + One Assignment member row belongs to one specific Assignment row with Assignment\_ID
  + Assignment member and Assignment must belong to same Project

# Future Work:

* More Assignment Types to task and bug
* More Roles for members of the team and asssignments

# 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

Week 4

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

Week 5

● Rewriting specification. GUI draft for: user, team leader, admin. Recreation in React components

Week 6

● 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

● Basic application functionality implemented end-to-end in both directions, Implemented authentication, bug fixing

Week 9 (Beta version)

● Finish GUI

Week 10

● Implementing statistics and progress of assignments and projects

Week 11 (Final version)

● Admin Interface