Project Management Platform

Răzvan Giurgiu
SD - 30432_1

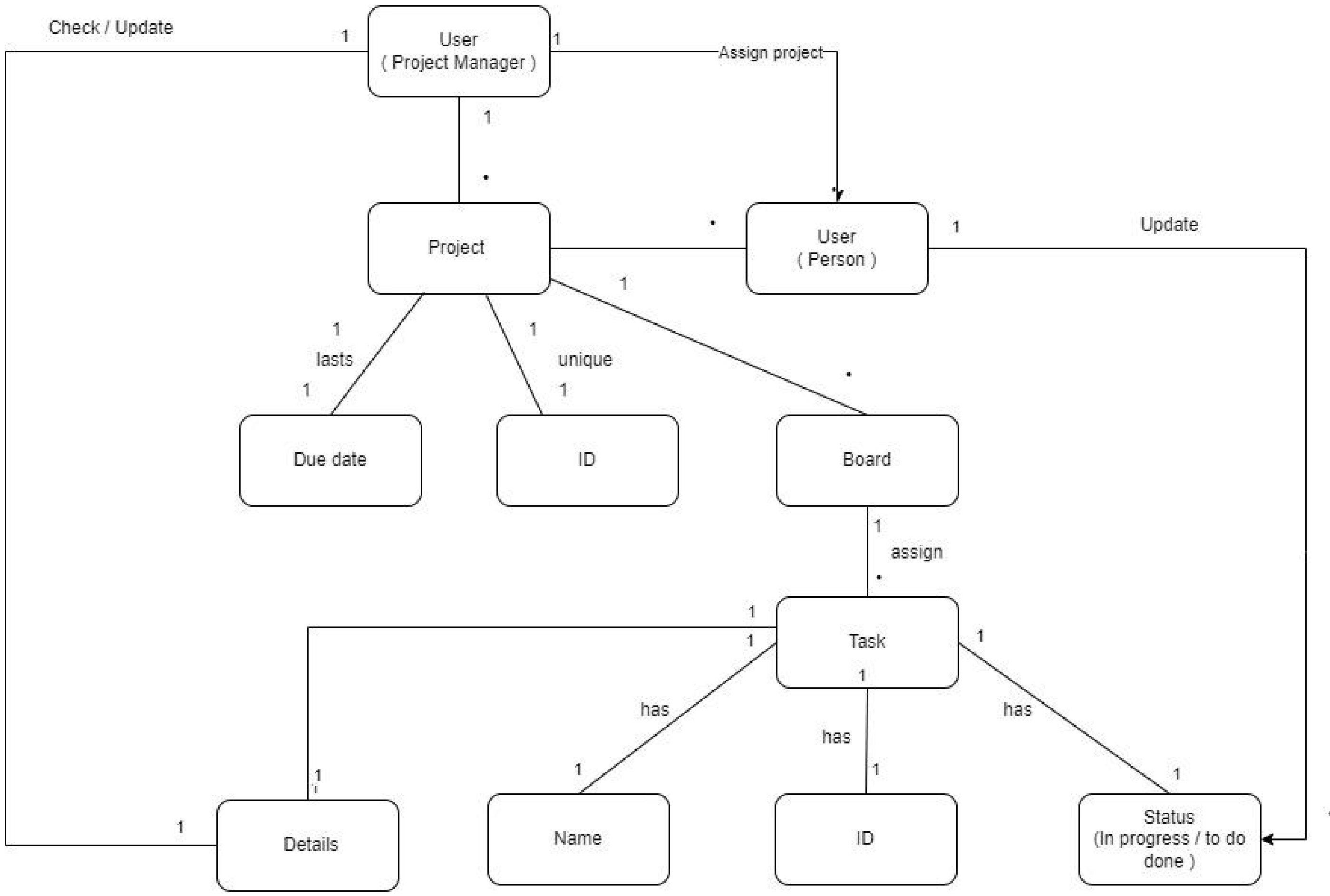
Content

Project specification

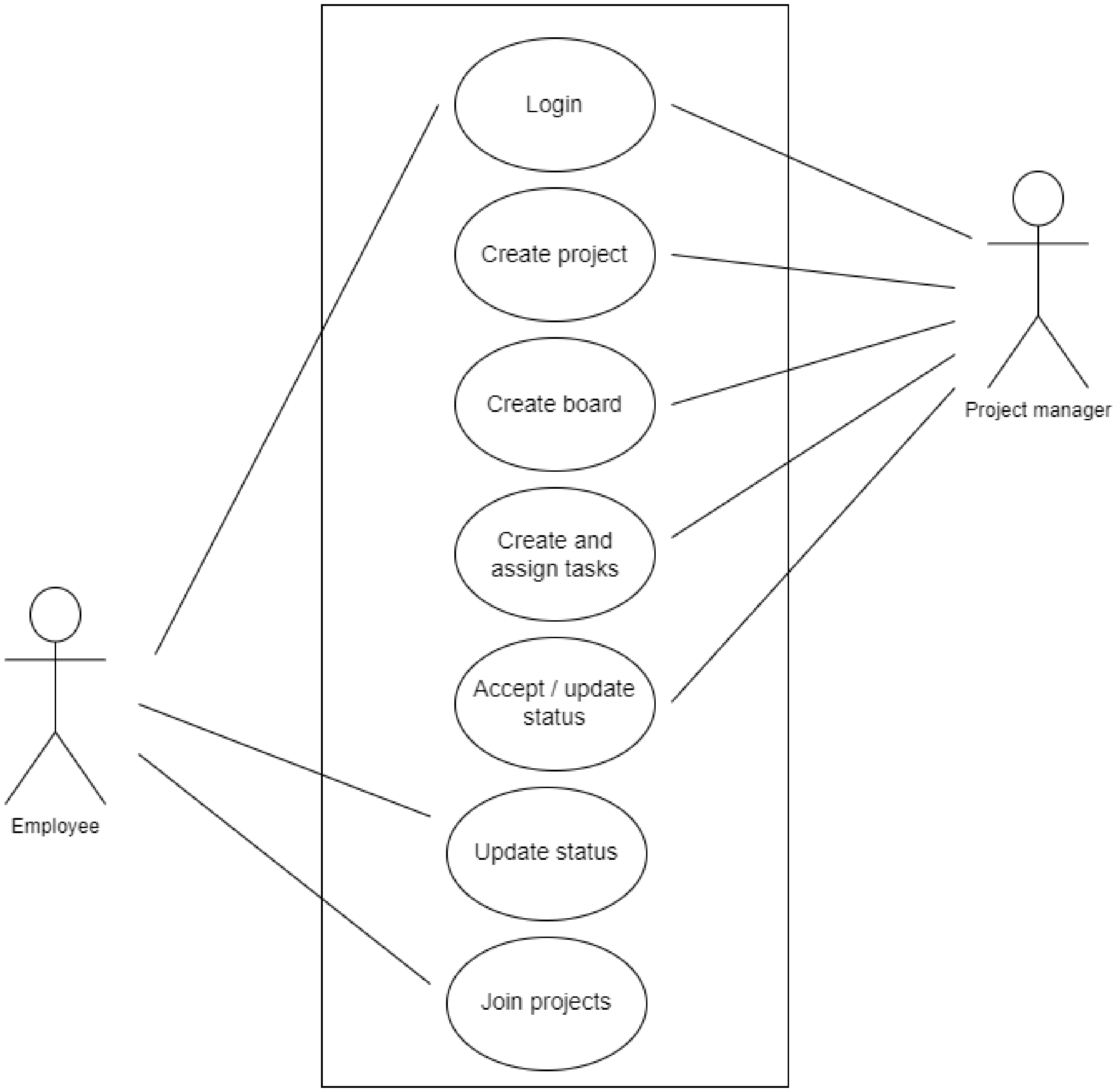
For the laboratory final project I would like to create a Project Management Platform, in a similar approach with Trello, Asana, etc, using Java and React. Project management software is software used for project planning, scheduling, resource allocation and change management. It allows project managers (PMs), stakeholders and users to control costs and manage budgeting, quality management and documentation and also may be used as an administration system. Project management software is also used for collaboration and communication between project stakeholders.

Main functionalities of the app will be to create new projects with specific attributes like name, id, description, start and end date and assign multiple users to this project. For each project there will be one or more boards where tasks will be created, updated and tracked for results. Users assigned to these boards can change the status of a task, ask for feedback and mark them as resolved.

Domain model diagram



UML use case diagram



Use case identification

Use case name: Create project
Level: User-Goal .
Main actor: Project manager .
Main success scenario: Users can successfully create a new project where people will be assigned to different boards depending on the needs of the project. One project can have multiple boards but at least one person must be assigned to each board.
Extension: Someone can move another person’s tasks and mark them as done.

Use case name: Accept / update status
Level: User-Goal .
Main actor: Employee .
Main success scenario: An employee marks the status as done but the project manager checks it and asks the employee to do some changes. The user receives a notification and does the appropriate tweaks so the task can be marked as done.

Architectural design

The application will use a layered architecture in order to organize the components in a better way and group together related classes. The layers are the following: **Presentation Layer**, **Business Layer**, **Persistence Layer** and **Database Layer**, each having a specific purpose.

1. **Presentation layer:** contains all of the classes responsible for presenting the UI to the user.

2. **Bussiness layer:** contains all the business logic that is required by the application to meet its functional requirements.

3. **Persistence layer:** it's an intermediary layer between the business logic of the application and the database layer. It contains the queries and the logic for processing and accessing the database.

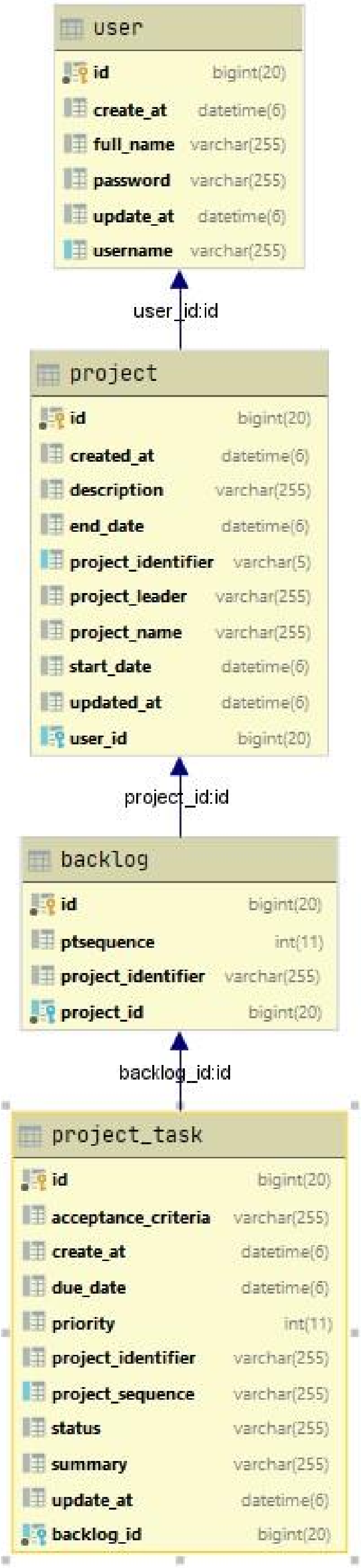
4. **Database layer:** contains all the classes mapped to the database.

Keeping this in mind, the project will have 4 packages: **Controller** – with all the presentation classes, **Service** – with all the bussiness layer classes which contain the functional requirements of the application, **Repository** – with the classes which directly access the database and **Model** – containing the model classes which directly map the database tables: User, Client, Tennis Court Manager, Tennis Court, Reservation.

Non-functional requirements

1. Availability: Application must be reliable for the users and maintain data since it's deployment
2. Security: Password must be stored encrypted and the process of making will use DTO's
3. Scalability: Scalability will be ensured by using AWS servers
4. Performance: Pages must load in less than 1 sec and a minimal number of requests must be sent to the server each time a new function is loaded.
5. Availability across devices: Cross-platform responsive design for desktop, android and ios

Database diagram



user	
id	bigint(20)
create_at	datetime(6)
full_name	varchar(255)
password	varchar(255)
update_at	datetime(6)
username	varchar(255)

project	
id	bigint(20)
created_at	datetime(6)
description	varchar(255)
end_date	datetime(6)
project_identifier	varchar(5)
project_leader	varchar(255)
project_name	varchar(255)
start_date	datetime(6)
updated_at	datetime(6)
user_id	bigint(20)

backlog	
id	bigint(20)
ptsequence	int(11)
project_identifier	varchar(255)
project_id	bigint(20)

project_task	
id	bigint(20)
acceptance_criteria	varchar(255)
create_at	datetime(6)
due_date	datetime(6)
priority	int(11)
project_identifier	varchar(255)
project_sequence	varchar(255)
status	varchar(255)
summary	varchar(255)
update_at	datetime(6)
backlog_id	bigint(20)

Design Constraints

My project is Client – Server web application developed in **Java** with the **Spring** framework. For the front end, I will be using **Typescript** with the **Angular** framework**.**

The data from the application will be stored in a **MySQL relational database** and mapped to the system using the **Java Persistence API** and **Hibernate.**

My app will be a cross-platform responsive web application that will resize based on the device it's accessed from