Bilkent University

Department of Computer Engineering

CS 353 - Database Systems

Term Project - Proposal

Project Name:	Tasks & Managers
Group No:	26
Group Members:	Cihan Erkan Merve Sağyatanlar Çağla Sözen Murat Tüver

Table of Contents

1.Introduction

1.1. Webpage

Project reports will be available on:

https://caglasozen.github.io/Tasks-Managers/

1.2. System Definition

In this project, we will design and implement a project tracking system that aims to aid project management by representing the hierarchical structures of companies as well as the structure of projects. This system will provide a user-friendly interface to track the progress of projects by structuring them into items such as boards, lists, cards. Cards may be added, removed and updated in a simple manner using the user-interface.

1.3. Necessity of the Database

A project management system should be able to incorporate large numbers of users, projects, project items and their associations to provide different privileges. Such a system should also be able to manipulate the information stored in these entities to provide the capability of user operations on project items and projects. Complexity of providing such abilities with a file system is not trivial and providing a certain flexibility to these abilities becomes even more sophisticated. Hence, utilising a database system is essential for providing a functional project management system in an efficient and effective manner.

1.4. Utilization of the Database

In the database system, data of users, boards, cards and their associations in teams and companies will be stored. Also, the database will ease access to data, its manipulation and ease providing safety to the system.

2. Functional Requirements

Tasks&Managers work on the concept of different functionalities for different types of users to achieve a successful project tracking system.

2.1. User Requirements

Tasks & Managers will be capable of reflecting the hierarchical order of a company. Therefore, it includes two main types of users, **privileged** and **standard** to clearly define the hierarchical structure along with the different requirements of different users. However, a user registered on Tasks&Managers does not have a single user type associated with its user account. If a user is associated with several user types, then that user has the privileges for the user type they have for the associated projects or teams. In other words, a user may have different privileges in different projects or teams according to their user type for that specific project or team. Each type of user represents different authorities and therefore have different limitations. The distinct functional requirements for the different types of users is provided in the following subsections.

2.1.1. Privileged User Requirements

Privileged users are divided into two subtypes, **manager** and **team leader**. These subtypes have different capabilities within the system and therefore have different limitations.

2.1.1.1. Manager Requirements

- Managers can create new **projects**.
- Managers can create **teams** with different tasks and add them to their **projects**.
- Managers can add members to **teams** with or without an assigned role**.**
- Managers can elect a **team leader** to be in charge of that team.
- Managers can add **comments** on **teams** in order to inform them further.
- Managers can observe **maps.**
    - A map may represent different information about the project according to its **content**.
        - A map can indicate the overall **progress** of the project in terms of team progresses.
        - A map can display all of the **teams** participating in the projects.
        - A map can show the **progress** of each team.
        - A map can inform manager about **issues** related to each team.
        - A map can show **lists** of teams and their tasks.
        - A map can show **members** of teams.

2.1.1.2. Team Leader Requirements

- Team leaders can create new **boards**.
- Team leaders can assign **roles/ratings** to users.
- Team leaders can create new **lists** and add them to a board.
- Team leaders can create **cards**, add a task description to them and add a **prerequisite** card if needed.
- Team leaders can assign team **members** to **cards**.
- Team leaders can edit **cards** by changing card names, assigning a different team member to that card or altering the task description.
- Team leaders can add **comments** to cards or list to make clarifications or to give feedback to team members.
- Team leaders can also assign a **card**, to themselves.

2.1.2. Standard User Requirements

- Standard users can change **status of cards** assigned to them.
- Standard users can add **comments** to their cards for further implementation notes.
- Standard users can create **issues** related to their tasks such as asking for clarifications, getting help from a different user or to learn more about a linked implementation.

2.2. Other Requirements

- Boards and lists has a **progress bar** to indicate how close they are to be completed.
- Lists has card so that task can divided in to smaller subtasks.
- A card may have **prerequisite cards** that block editing the status of a card before the prerequisite cards status is complete.
- Cards have **status** to reveal whether it is done.
- When a **company** is signed up to the system, their company id's can be used to enable users to enroll into companies.

3. Non-Functional Requirements

3.1. Response Time

Tasks & Managers will be a system that requires fast response to queries to be able to offer a high performance to the users.

- Queries should be highly optimized.
- Database management system used in the implementation should not provide poor transfer speed.

3.2. Safety

Tasks & Managers hosts users of varying privileges. Each user should be able to perform only the operations within their given privileges.

- There should be no projects created by a non-manager user.
- There should be no boards or cards created by a member user within a team. Boards and cards can only be created by the corresponding team leaders.
- No team leader user should be assigned to a team by a non-manager user.
- No member user should be assigned to a project by a non-manager user.
- No member or team leader user should be assigned to a card by a non-team leader user.
- A company in the system should not have more employees enrolled in the system than the specified employee number.

3.3. Scalability

Tasks & Managers will be a system that operates on the existence of users, items and their interactions. Therefore, the system should neither limit the number of users nor the number of user-created items.

- Number of users and user-created items should only be limited by the database management server used in the implementation.

3.4. User-Friendly Interface

Tasks & Managers will be a system that will be used by users of varying levels and professions during complex project implementations for communication and progress tracking. For this reason, user-friendliness of the application is crucial to provide easy communication between units and represent the status of the project in a more comprehensible manner.

- Creating projects and boards should be easy to perform. A user, having the correct permissions, should be able to create a project or a board by clicking no more than 3 buttons.
- Adding lists and cards should be easy to perform. A user, having the correct permissions, should be able to add a list or a card by no more than 1 button click.
- Adjusting the placement of cards should be easy to perform. A user, having the correct permissions, should not have alignment difficulties in card placement. A card should automatically align itself to a list if ½ of the card body is on the list.
- Managing cards should be easy to perform. A user, having the correct permissions, should be able to edit/manage a card by no more than 1 button click.
- Managing lists should be easy to perform. A user, having the correct permissions, should be able to edit/manage a list by no more than 2 button clicks.
- Managing/Aligning multiple cards should be easier to perform. A user, having the correct permissions, should be able to manage/align all cards in a list by no more than 2 button clicks. Also managing/aligning multiple (*N*) cards should not exceed *N+2* clicks.

4. Limitations

- A company has 0 or more employees.
- A user can be employed by 0 or 1 company.
- A user might have different roles in same project.
- 1 or more teams can work on the same project.
- A project has exactly one manager.
- A project has one or more map.
- Each team has exactly one team leader.
- Each team has 0 or more members.

- Each team works on exactly one project.
- Each team has 1 or more boards
- Each board has 1 or more lists.
- Each board belongs to exactly one team.
- Each list contains 1 or more cards.
- Each list belongs to exactly one board.
- Each card has 0 or 1 user assigned.
- A team member can be assigned to 0 or more cards.
- A team member can be in 0 or more teams.

5. Conceptual Design of the Database

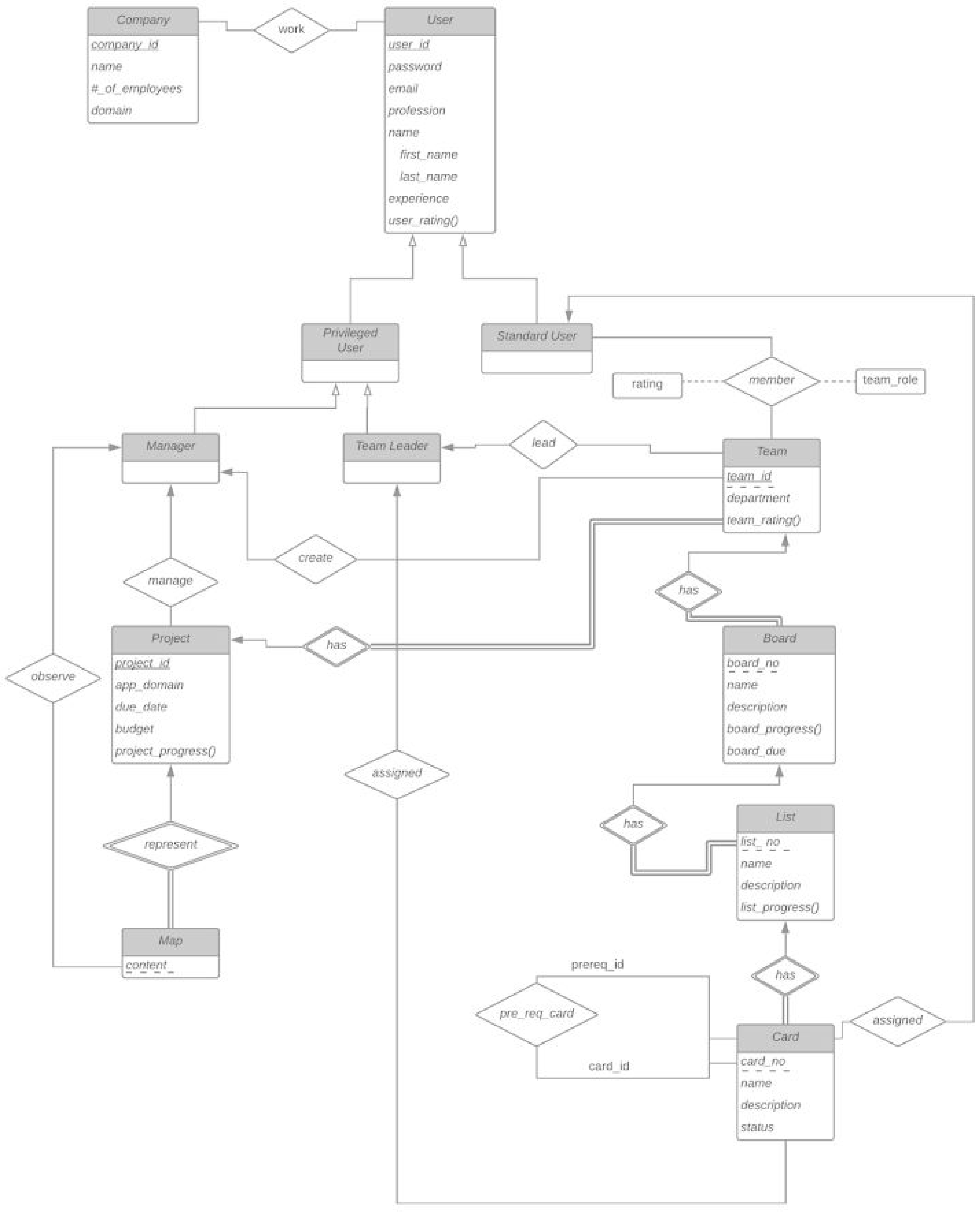
5.1. Description

Conceptual design of the database is explained below as also represented in the E/R diagram.

- In order to be able to use Tasks&Managers, creating a **User** account is compulsory. All user accounts have *user_id, password, e-mail, profession, name, experience* and *user rating*. *User_id*'s are unique attributes of users so that they can be distinguished. *E-mail* and *password* will be used for logging into the system. Additionally, *e-mails* will be used to confirm user accounts to adjust the appropriate privileges. *User rating* will be derived from the users' ratings in different teams. A user may be a **Standard User** or a **Privileged User.** Privileged users may be **Team Leaders** or **Managers.** Different types of users have different rights in the system to provide safety within the system.
- Users in the system may be enrolled in **Companies.** Companies have *company_id, name, #_of_employees, domain* attributes. Each company will be uniquely identified by the *company_id. #_of_employees* will identify the number of employees in the company that may be enrolled in the system or not. *domain* attribute will identify the sector of the company.
- Main entities that users interact with in Tasks&Managers are **Projects**. Projects have *project_id, app_domain, due_date, budget, project_progress* attributes. Each project will be uniquely identified by the *project_id. project_progress* will be derived from the progress of boards in the project.
- Information and status of each project will be represented by several **Maps.** These maps will be uniquely identified by their *content* within the same project.
- **Teams** will be working on a project. Teams will be created by a manager. Each team has *team_id, department* and *team_rating. team_id* will uniquely identify each team within projects. *team_rating* will be derived from the ratings of all members of the team. Each team member will have a *role* and a *rating* within the team. Each team will have a **Team Leader** assigned by managers. Team Leaders will assign members to the team they are leading.

- Teams will have **Boards.** Boards have *board_no, name, description, board_progress,* and *board_due. board_no* will uniquely identify boards within teams. *board_progress* will be derived from the progress of lists in the board.
- Boards will have **Lists.** Lists have *list_no, name, description* and *list_progress. list_no* will uniquely identify lists within boards. *list_progress* will be derived from the status' of cards in the list.
- Lists will have **Cards.** Lists have *card_no, name, description* and *status. card_no* will uniquely identify cards within lists. A card may have a *pre_req_card*, such that the card with the *prereq_id* should have its status as "complete" before the card with *card_id.*

5.2. E/R Diagram



Company
company_id
name
#_of_employees
domain
work
User
user_id
password
email
profession
name
first_name
last_name
experience
user_rating()
Privileged User
Standard User
rating
member
team_role
Manager
Team Leader
lead
Team
team_id
department
team_rating()
create
manage
has
Project
project_id
app_domain
due_date
budget
project_progress()
observe
has
Board
board_no
name
description
board_progress()
board_due
assigned
represent
has
List
list_no
name
description
list_progress()
Map
content
prereq_id
pre_req_card
card_id
has
Card
card_no
name
description
status
assigned