Şur-Up : Upload your Projects
Project Proposal Report
Project Tracking Software

By Grup Surup **Group 27**

Muhammet Said Demir	21602021	sec.3
Zeynep Nur Öztürk	21501472	sec.3

04.03.2019

Table of Contents

1. Introduction..3
2. Why database System is required?..3
3.Functional Requirements..4
4.Non-Functional Requirements..6
5. Limitations..6
6. Entity-Relationship Diagram..7
7. Descriptions of the Relations..7

1.Introduction//detail

This report aimed to examine our database project “Project Tracking Software”. In this report, we have examined our database project under several topics. Therefore, we used the following headings in our proposal which are functional requirements, non-functional requirements, limitations, and Entity-Relationship Model diagram to make our project report explanatory and understandable.

In this database design, we aimed to create an efficient, sustainable and user-friendly database system with innovatively added features. Before the designing process, we have examined similar Project Tracking Softwares such as Trello and Jira. During the design process, we focused to create an effective number of attributes to provide simple and easy usage for the users. Firstly, we have provided all the required features to the users and then, we tried to add useful additional entity-sets and relations in order to increase features in our database project.

The users can create teams, boards, lists, and cards. Leader can arrange meetings and can promotethe empleyee as a leader to a team. When the user arranges meetings he also arranges calendars of the users based on meetings. When the user creates cards, user writes a work needed to be done, therefore, this work has a deadline, user also creates calendar. User can edits cards and calenders (deletes, moves to other lists, rewrites, etc.) User can assign any card to any user, this causes changes the assigned user’s calendar. User has archieve, and can sends cards to the archieve. Every user can send a message to another user. When a board is created by a user, boards has reports. Also a leader can promote or fire employees different than other users.

Briefly, our system allows users to create a project and add new project members to the project, then manage this project using with some entity sets, such as calendar and board. Additionally, while managing the project, users can also create meetings while controlling project reports and can assign tasks into the project as pre-works for the development process.

As some explanations for our headings, in limitations part we have explained some of the restrictions that we added for customers due to easy use and efficiency of the system and Entity-Relationship Model diagram section describes our conceptual Design and specifies the entity sets and relationships in our database project.

2. Why Database System is Required?

Our system “Project Tracking Software” requires persistent storage to store data’s, processes and user interactions of the project. For example, after the user created a project, we need to save this project somewhere to protect our data's. Therefore, we need an efficient database design to store required data’s. In addition, there will be more information about the projects, users and the process of the continuing project or finished project during the development process of our design. Thus, we need to come up with an effective storage system. As we discussed, our project has several entity sets and relation sets for managing the system. Therefore, we needed to design a relational database system to create an efficient database model while storing these necessary components of our design.

3. Functional Requirements

3.1 User

User entity represents the users which are enrolled the system. Users will have name, password and e-mail. Users have calendar to adjust their schedule or follow the important events such as meetings etc. Users can create team, board, card also. Users might have 1 calendar, many teams. There are 2 type of user. These are leader and employee.

3.2 Team

Teams have names. They also have users, boards, and meetings. These teams represents a group of the user that responsible from any specific job(board). Sometimes a team might not have a board(job). For example, some users can create team firstly and define the job after that. Teams might have many user, many boards and many meetings.

3.3 Board

Board represents the projects, tasks. It has name, deadline, privacy, starred, background. Board deadlines actually specify that which board considered as big project or mini tasks such as sprints. Therefore, users can create a sprint(small task) or a big project. Privacy demonstrates whether the board is open to everyone because some users or companies can enroll the system and they might want to hide their jobs. Board might have 1 team, many lists and reports.

3.4 List

List represents upcoming tasks, finished tasks, or the tasks that still continue. List keeps these type of the small task chain. List has name and may have many cards attached to the list.

3.7 Card

Card represents the smaller tasks in boards. These cards contains the name, description that explains detail of task, label that shows the emergency or priority of task, deadline and comment.

3.8 Meeting

Meeting has time, date, meeting place and team notes. Teams can follow the meeting info from the boards they have. Also members of team can read the meeting notes to follow important decisions.

3.9 Report

Report has the name, type and description. They consist of informations about the board the project.

3.10 Calendar

Calendar that shows the personal schedule of the users. Users can follow the events, tasks etc according to their calendars and can adjust these accordingly.

4. Non-functional requirements

4.1. Usability

- User should have a basic understanding of betting to use the system.
- The system should have a basic tutorial when they sign up to the system.

4.2 Reliability

- the system should be avaible 24/7 and 365 days to the users
- User's password should contain at least 6 characters with the one number and one special character.

4.3 Performance

- Database query shoul not take more than 1 second.
- The system should support high number of concurrent users.

4.4 Packaging

- The system should be web based and should not require any installation(neither desktop nor browser extension)

4.5 Portability

- This system needs to work on different operating systems, browsers such as Chrome, Mozila, etc., and most of the smartphones, tablets, and different operating systems.

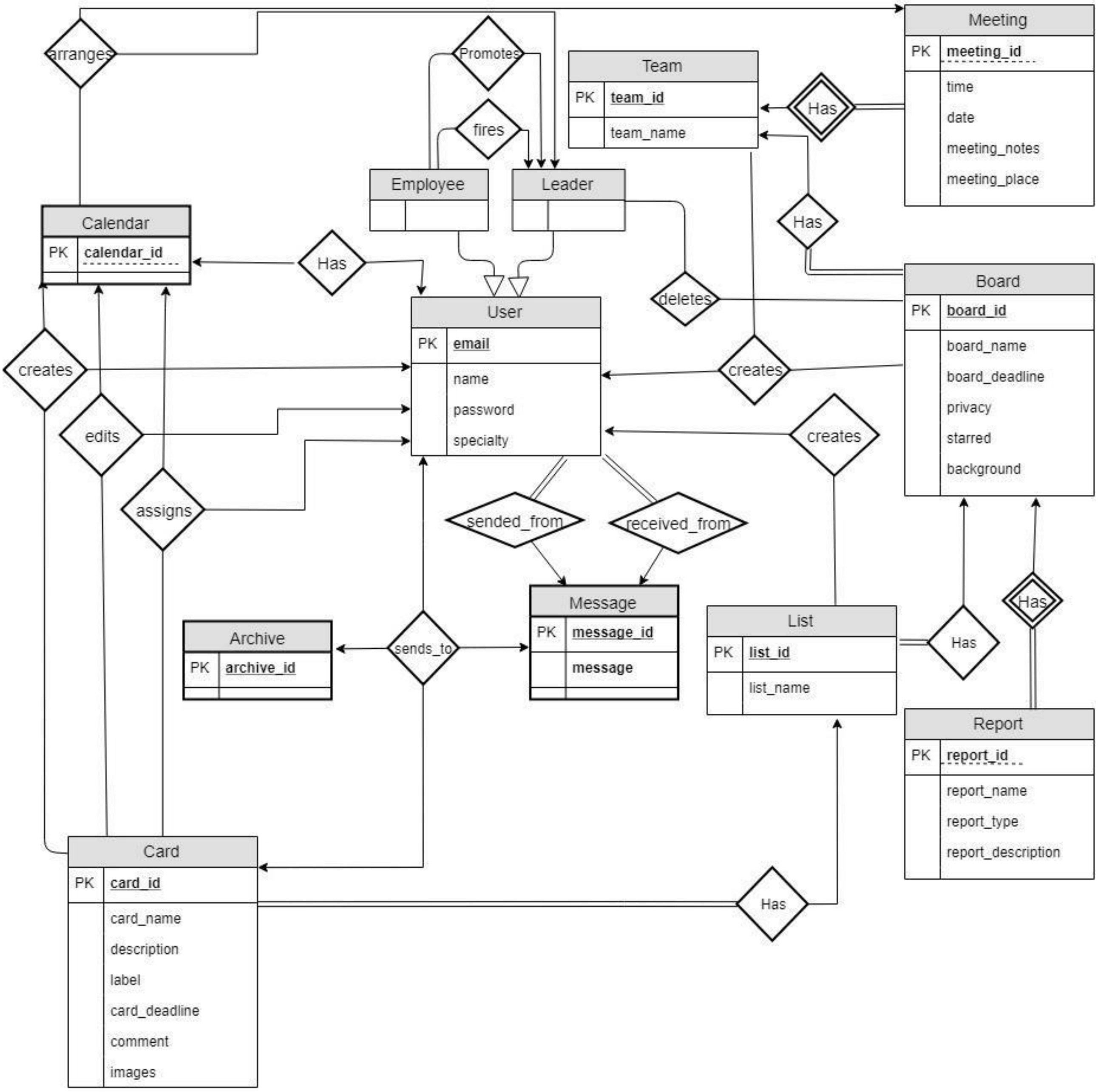
4.6 Authentication

- This system includes the end-user and administrative interfaces. Given permissions to the users should be specified and prevent them drom some extra actions that they are not allowed.

5. Limitations

- At most 100 people can use the system concurrently.
- A password can be at least 6 characters and at most 8 characters
- Teams have at least one leader.

6. Entity-Relationship Diagram



7. Descriptions of the Relations

7.1 Promotes relation between Leader and Employee

It’s a one to many relation which indicates that a leader can promote an employee to be a leader as well.

7.2 Fire relation between Leader and Employee

It’s a one to many relation which indicates that a leader can fire an employee from the team.

7.3 Has relation between User and Calendar

It’s a one to one relation which indicates that a user can have a calendar which holds the information of due dates and meetings.

7.4 Has relation between User and Archive

It’s a one to one relation which indicates that a user can have an archive to store cards.

7.5 Sends_message relation between User and User

It’s a one to one relation which indicates that a user can send message to another user.

7.6 Has relation between User and Team

It’s a many to many relation which indicates that there can be many users within a team and also a user can be within many teams.

7.7 Has relation between Team and Meeting

It’s a one to many relation which has total participation on the meeting side which indicates that a team can have multiple meetings within itself but the meetings can be assigned only to one team.

7.8 Arranges relation between Leader, Calendar and Meeting

It’s a ternary relation which indicates that a leader can arrange a meeting which updates the calendar of the team members.

7.9 Has relation between Team and Board

It’s a one to many relation which indicates that a team can own multiple boards whereas a board can be owned by only one team. Also, board has total participation which means that a board can be assigned to only one team, not 0.

7.10 Creates relation between User, Team and Board

It’s a ternary relation which means that a user can create team or create board. When the user creates a board without creating a team first, the system creates an automatic team which only has 1 member which is the user itself as the leader. Also, when the user creates a team, it assigns an empty board which the user can edit it.

7.11 Has relation between Board and Report

It's a one to many relation which has total participation on report side which indicates that a board can host a multiple reports but a report can be assigned to only one board.

7.12 Has relation between Board and List

It's a one to many relation which has a total participation on List side which says that lists must be assigned to boards. A board can have multiple lists such as ToDo/Finished/Graphics etc. But a list can only be assigned to a single board.

7.13 Creates relation between User and List

It's a one to many relation which indicates that a user can create lists, but a list can be created by only one particular user.

7.14 Has relation between List and Card

It's a one to many relation which has total participation on cards side which indicates that multiple cards can be assigned to a certain list whereas a card cannot be assigned more or less than one list.

7.15 Creates relation between User, Card and Calendar

It's a ternary relation which indicates that User can create multiple cards which changes the calendar of the user due to the fact that the due date of the cards will be shown in the calendar of that particular user.

7.16 Edits relation between User, Card and Calendar

It's a ternary relation which indicates that a user can edit the card, which is to say; add images, comments, change due date, name, label etc. Thus, it may change the Calendar of the assigned users as well.

7.17 Assign relation between User, Card and Calendar

It's a ternary relation which indicates that a user can assign other users to cards which directly updates the calendar of the assigned users.

7.18 Sends_to relation between User, Card and Archive

It's a ternary relation which indicates that a user can send the specific card to that user's specific archive in order to save it in the archive.

You can find our project on the web,
Our address is: https://github.com/SaidDemir/-ur-up/blob/master/Group27_proposal.pdf