Task Manager

By Muhammad Al Ghali

1st Muhammad Al Ghali
Student at University of Michigan-Dearborn
Dearborn, United States of America
Mhdghali@umich.edu

***Abstract*—Every Project Require resources, be it human resources, computing resources, and management resources. One way into enhancing management of projects is using a task manager to track progress and deadlines.**

Keywords—component, Task Management, Project Management.

I. Introduction

A lot of people start a project and get into development without engineering the solution step by step from requirements to maintenance. One of the issues faced in the development process is tracking the progress of the engineers. In this project, I will be developing a project-task tracking system. This project is a tool used in almost all companies worldwide to keep track of their project progress and track members' tasks in each project.

II. Software Engineering Method

A. Motivation for method use

Due to time limitations and complete knowledge of the problem which we are facing, which is task management, the requirements are known and don't usually change.

B. The method selected

The method usually used in this case is the Waterfall method, this method usually proceeds in a strict order without and overlapping and iterative steps.

III. Method Steps

Every software engineering method has its own step, and for you to achieve the outcome you expect you should follow these steps.

The steps of our method, the Waterfall method are seven: concept, requirements elicitation, design, implementation, testing, installation, and maintenance and deployment.

A. Concept and Literature Survery

The concept of Task management and project management aren't new concepts. Much software has been invented in this area of research. So, in my software I tried to do some enhancements that would work well in a small business environment.

Table 1 illustrates our features in contrast to other software's features.

B. Requirments

- As an admin and a manager, I would want to start projects and be able to add people to the project.
- As an admin and a manager, I would like to assign a role for my employees in a project.
- As a user, I would like to be able to send comments on a task.
- As an employee, I would like to view all my tasks in different projects on my dashboard and their respective deadlines.
- As an employee assigned a task, I should be able to edit the status of a task.
- As a manager, I should be able to view who is the team on my project and what is everyone's role.

Table 1. Our feature and others software's features

Our Software	Quire Tasks [1]	Asana [2]	Google tables [3]	Plutio [4]
Simple Dashboard	Tasks and Subtasks	Regular users get: Projects	Many workspaces and collaborators	Timeline/Gant view
Table view for Every project's task	Progress reports	Tasks	Reports, automated actions using bots, triggers	Kanban View
Task Deadlines	Recurring tasks	Basic dashboard	Simple task trackers	List view
Role dependent dashboard	Apps for IOS and android	Basic search	Project tracker	Calendar view
Commenting on tasks	Calendar	Premium: many Dashboards	Kanban boards	Dashboards
Only admins can add users		Private teams and projects	Employee directory	Projects
Roles and permissions		Start dates and deadlines		Team chat
Web based app				Time-tracking
User has access to only his/her tasks				

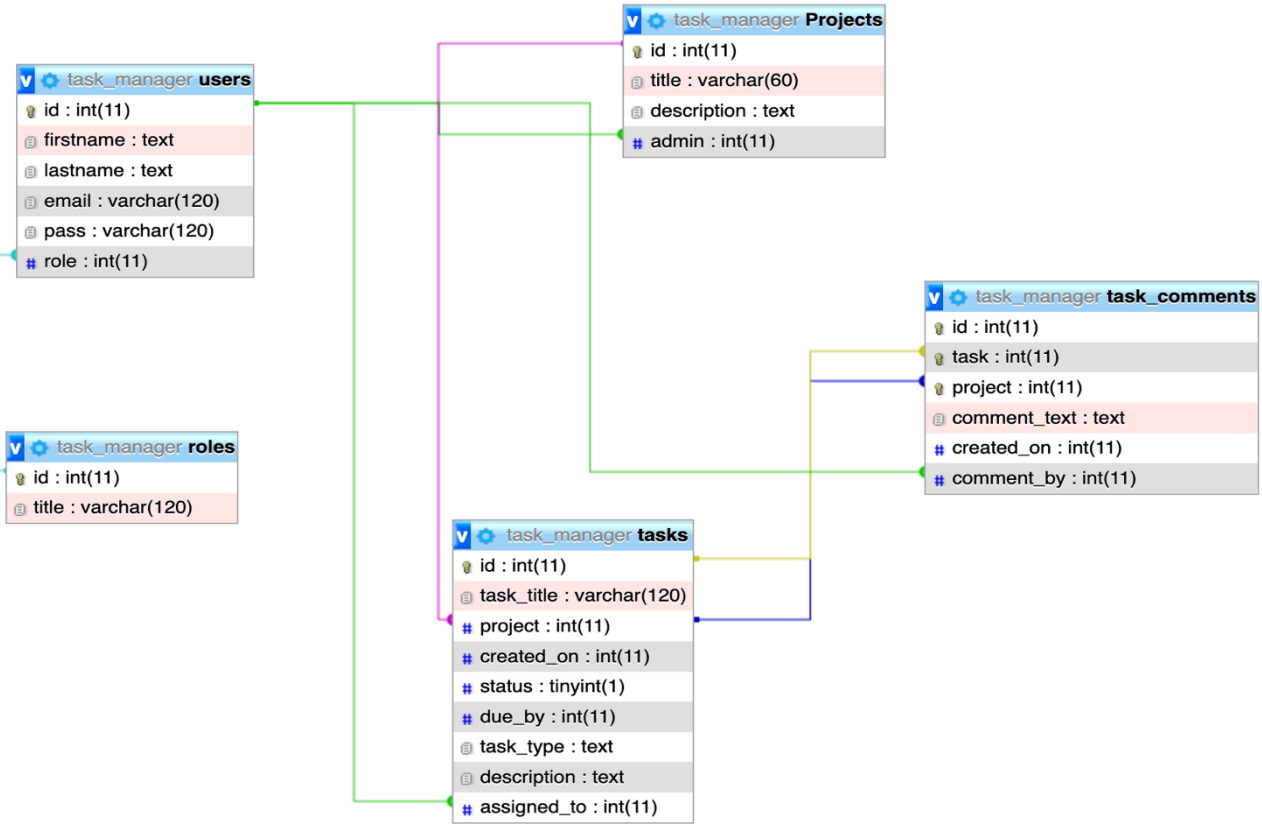

Figure 1. ERD diagram for the database

C. Design

Figure 1 presents the ERD demonstrating the database structure. First, we have the main table the Users table, which has the attributes provided for each user. Second, we have the table of roles. The roles are 3 only system_admin, manager, employee. Then, we have the projects table, which contains every project's id, title, description, and who is the manager of the project. Last but not least, we have the task table, which presents each task, it shows the id of the task, its title, the name of the project that this task belongs to, the date it was created, the status whether not started or in progress or complete, the due date, the task type which is basically the role of the person, the description of the task, and finally the person who the task is assigned to from the users' table. Finally, we have the comments table, which presents each comment's id, the task being commented on, the project the task belongs to, the comment itself, the date it was commented, and who authored the comment from the users' table.

Figure 2 below presents the flowchart of the whole project. The flowchart demonstrates a step-by-step route for every user role. There are three routes:

1- Employee route: After launch, if logged in the user gets redirected to a dashboard that shows his/her tasks. The Employee has only 4 function which are adding a comment on a task, editing task status, changing password, and viewing all the tasks assigned to him/her.

2- Manager route: The manager route has all the functionality for a regular employee, but with few additions. The additions are that the manager can create a project and can create tasks, and the manager views the projects he is managing with all the tasks within unlike the regular employee who can only view his/her tasks.

3- System Admin Route: The most powerful route has all the privileges the others have and few extras. The extras are that the admin only can add new users and assign their respective role and can also view all projects and all tasks present in the system.

D. Implementation

- The first step in implementation is building the database using MySQL [5].
- The second step was building up the queries needed for the CRUD API operations. Example: adding a new user, creating a new project, creating a task, and assigning it to an employee
- The third step after building the queries needed for the data retrieval and sending is building the server and making the APIs using Nodejs [6], and Express.js [7].

Figure 2. The flowchart of the project

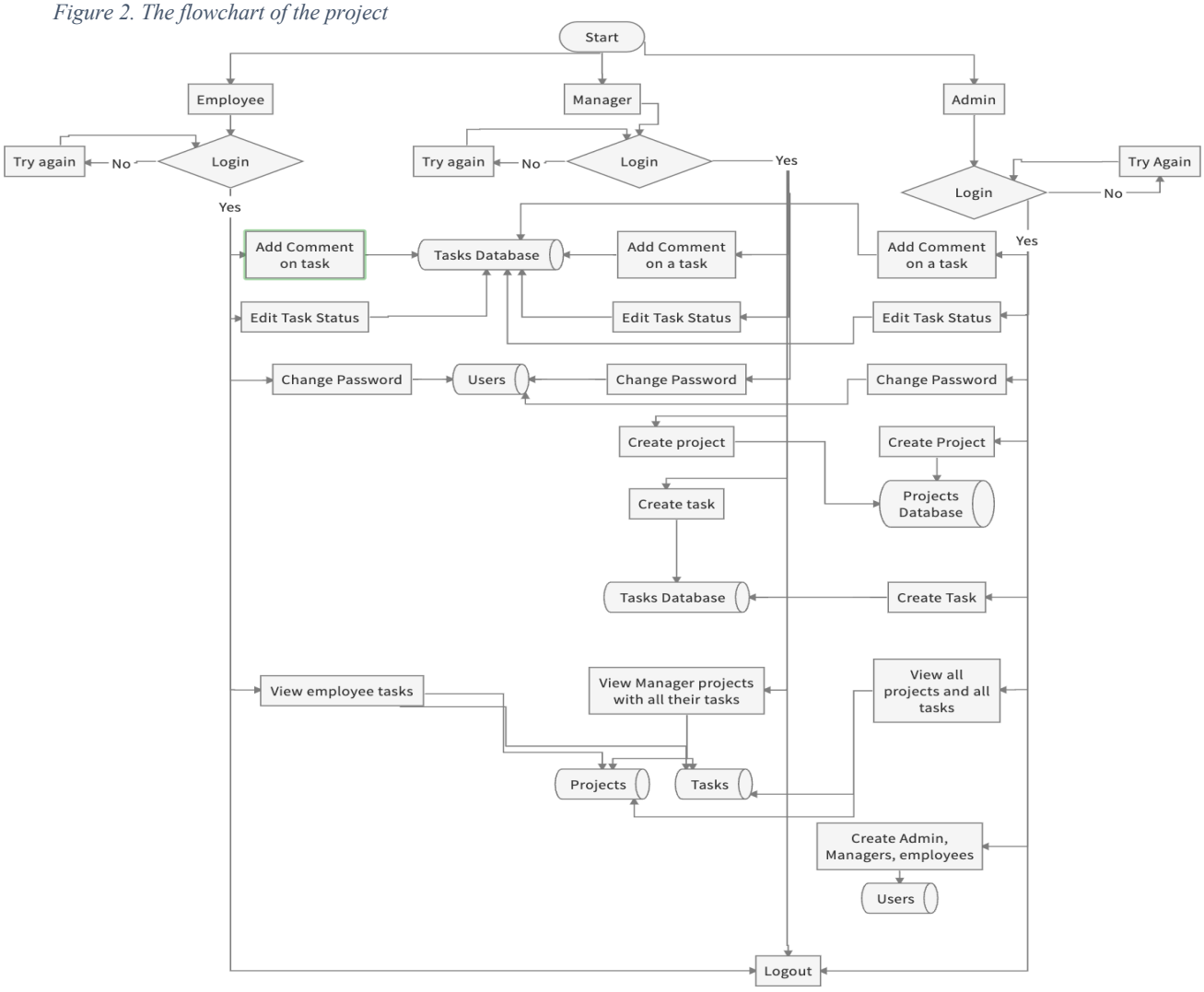

Figure 3. Functionality table by role

Function Available / Role	Employee	Manager	System Administrator
Add New Users	No	No	Yes
Get All Employees	No	No	Yes
Change Password	Yes	Yes	Yes
Add/Delete Projects	No	Yes	Yes
Add/Delete Tasks	No	Yes	Yes
Edit Task Status	Yes	Yes	Yes
Get Projects with Tasks	Yes	Yes	Yes
Get Projects and Tasks Managed by User	No	Yes	Yes
Get All Projects and All Tasks	No	No	Yes
Get/Add Comments	Yes	Yes	Yes

- The fourth step was building a user interface with good

user experience. This was built using ReactJs [8] a library made by facebook, and a CSS library for ReactJs called Rsuitejs [9].

- The last step was implementing the API calls from the frontend. All the calls were made using axios [10] which is a promise-based HTTP client.
- In the frontend to display a dashboard according to role, after we receive the user information on login, we check the role and according to role the frontend displays the targeted interface.
- In the backend when a user logs in the frontend send a request to get tasks, and according to the role of the user the user receives either the tasks he is assigned only, or the tasks with the projects he is managing, or he gets all the tasks and projects.

Figure 3. demonstrates what function each user has depending on his/her role.

E. *Testing*

- *In the testing of this software a combination of black-box and white box testing has been applied in addition to regression testing.*
- *After building each query and api routes, every independent query and route was tested at least once, even logical decisions like checking if the user has privileges to do a certain request or not.*
- *After finishing the full project, every functionality was tested in order to make sure the software as a whole works or not.*
- *As an extra layer of assurance regression testing was used, after adding any feature the software was tested to check If the feature breaks the app or works correctly.*
- *After all these tests many bugs where found and fixed, but many bugs could still be undiscovered.*

F. *Installation*

1. To install MySQL, you could visit: https://www.apachefriends.org/
2. To run the frontend and backend NodeJS should be installed on your laptop from: https://nodejs.org/en/
3. After downloading the previous two, you could go to the frontend and backend and just type in the command line npm install then npm start, this will install all the packages required in this project and will run the full project.
4. In the backend some changes should be applied for the express server to connect to MySQL server.

G. *Maintenance and deployment*

a) This phase is currently not applicable since the project is not going to be deployed

b) Steps to deploy the project on your device:

1- After installing Nodejs and the MySQL server, launch the xamp manager and turn on the Apache web server and MySQL database.

2- After running the MySQL server get the port number which is usually 3306 and input it in the backend in the databaseManager.js file

3- Then from the code files import the sql code to the phpMyAdmin which you can access by typing in the browser localhost/phpMyAdmin

4- After making sure that the database is setup run the server by getting into the backend folder and typing in the console npm install then npm run

5- After that get into the frontend file and access the api.js file and put the link that your backend is running on and add to it /api and then in the console type npm install and then type npm run

6- Finally, you're all setup for the project. For any assistance contact: mhdghali@umich.edu

Results

This project is a small step in the project-task management field, this software works for a small businesses and small team projects. For the project to be used for large scale business a lot of services need to be upgraded.
In this project I tried enhancing the task management tool by making a light frontend that adapts to users according to his/her role without making separate pages for every role

Conclusion

In conclusion, software management is a process itself and it should be taken care of to succeed in building a project.

References

[1] Quire: https://quire.io/
[2] Asana : http://www.asana.com/
[3] Google Tables: https://tables.area120.google.com/about
[4] Plutio: https://plutio.com/
[5] Xamp in order to get MySQL: https://www.apachefriends.org/
[6] NodeJS: https://nodejs.org/en/
[7] ExpressJs: https://expressjs.com/
[8] Reactjs : https://reactjs.org/
[9] Rsuitejs : https://rsuitejs.com/
[10] Axios for api calls in frontend : https://axios-http.com/docs/intro