CIS 5690 – Advanced Systems Project

Task Tracker: Microservices – Based Task Management System

By

Poojitha Puppala (700746591)

Big Data Analytics and Information Technology

UNIVERSITY OF CENTRAL MISSOURI

Abstract

In the fast-paced work environments of today, effective task management is essential to increasing both individual and group productivity. Nevertheless, current task management systems frequently provide difficult challenges because of their complex interfaces and unnecessary features, intimidating customers who are looking for a simple and efficient tool. The ability of individuals and small teams to collaborate and plan activities together is limited in the lack of a user-friendly, flexible task management system with essential capabilities. The goal of this project is to close this gap by designing and building a user-friendly, lightweight task management system that is specific to the needs of individual users as well as smaller teams. The system is made with an easy-to-use interface in mind, utilizing modern technology to provide flexibility and user-friendliness. Through the integration of both functionality and simplicity, the suggested system aims to enable users to go through their duties with ease, promoting increased productivity and collaboration in a variety of work environments.

TABLE OF CONTENTS

1. Project Description

In contemporary work environments, effective task management is paramount for individual and team productivity. Existing task management solutions often present a significant barrier due to their complexity and a surplus of features, making them overwhelm for users seeking a straightforward and efficient tool. The absence of a lightweight, user-friendly task management system with essential functionalities hampers the ability of small teams and individuals to organize and collaborate on tasks seamlessly. This system should cater to the specific needs of smaller teams or individual users, providing an intuitive interface while leveraging modern technologies to ensure flexibility and ease of use.

Key Functionalities:

User Registration and Authentication:

Users can register and log in securely to access the task management system.

Task Creation:

Users can create tasks and set due dates on them.

Task Assignment:

Users can assign tasks to team members and reassign any time.

Task Prioritization:

Tasks can be prioritized for efficient time management.

Task Listing:

An intuitive dashboard displays a list of tasks displayed in states like TODO, In-Progress, QA, and Done.

Task Filtering:

Users can filter the tasks with filtering options based on status, due date, and assignee.

Real-time Updates:

Users receive real-time updates on task status changes, ensuring effective collaboration.

Microservices Architecture:

The system will be built on a microservices architecture, separating concerns for user management, task management, and authentication.

2. Technologies/Tools

Backend (Microservices):

Java with Spring Boot: Used Spring Boot to facilitate the creation of microservices, guaranteeing scalability and smooth integration for quick deployment.

MySQL Database: Protect user information, task details, and system configurations in MySQL to guarantee safe archiving and retrieval.

Spring Security: Used Spring Security to fortify the application by integrating strong permission and authentication mechanisms for improved security.

Frontend:

React: Used React to create dynamic user experiences that encourage responsiveness and interaction for effective task management.

Redux: Used Redux to centralize state management while enhancing data flow control and front-end application performance.

Communication:

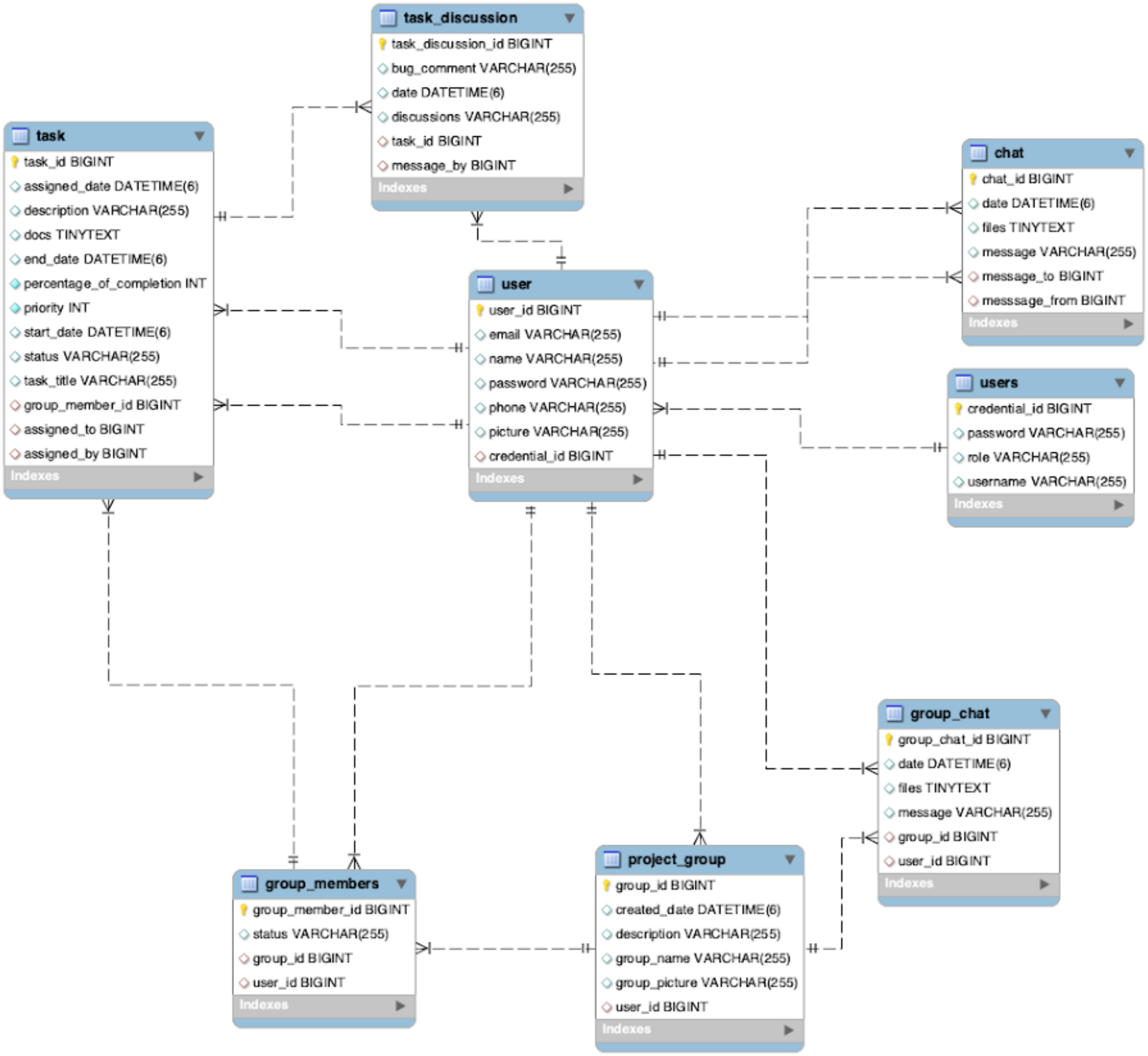
RESTful APIs: Encourages smooth connection between the front-end and microservices to provide effective data interchange and interoperability.

Testing:

JUnit and Mockito: Used JUnit and Mockito to conduct thorough unit testing to ensure the functionality and dependability of microservices and to ensure smooth operation in a variety of scenarios.

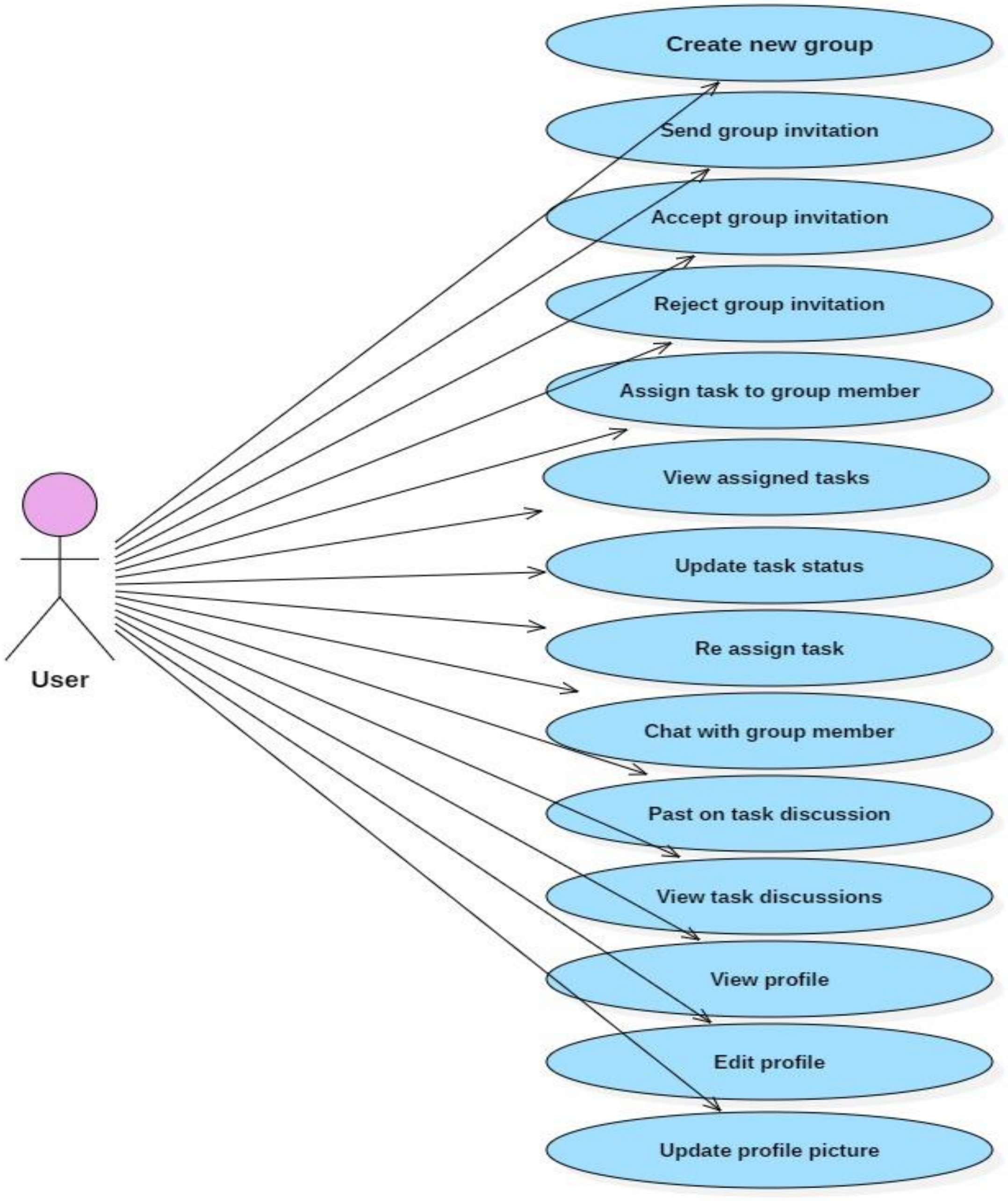
3. System Design and Architecture

a. E-R Diagram

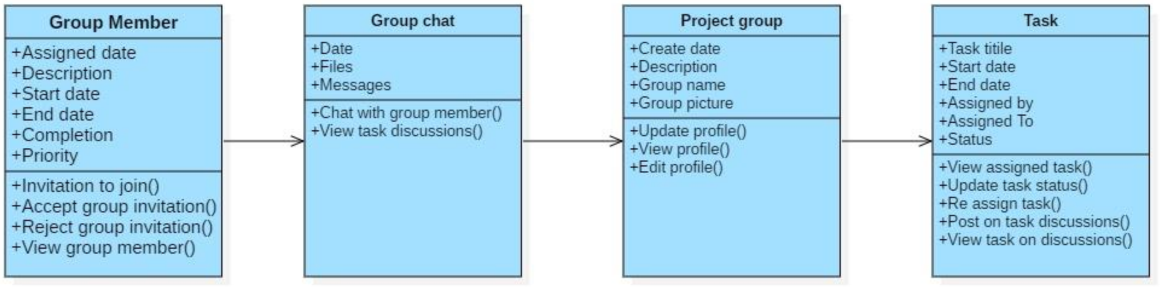


b. UML Diagrams

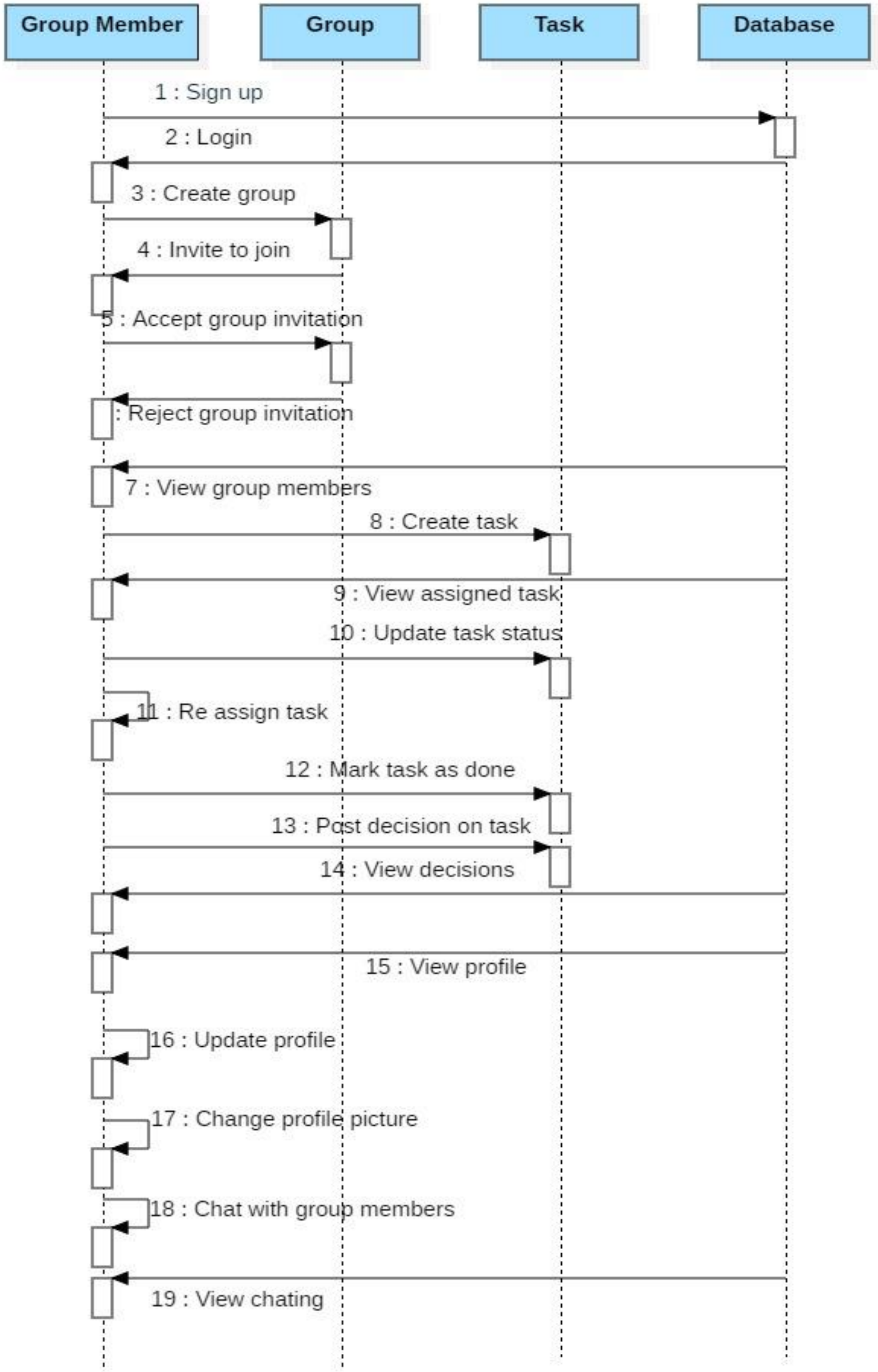
i. Use Case Diagram



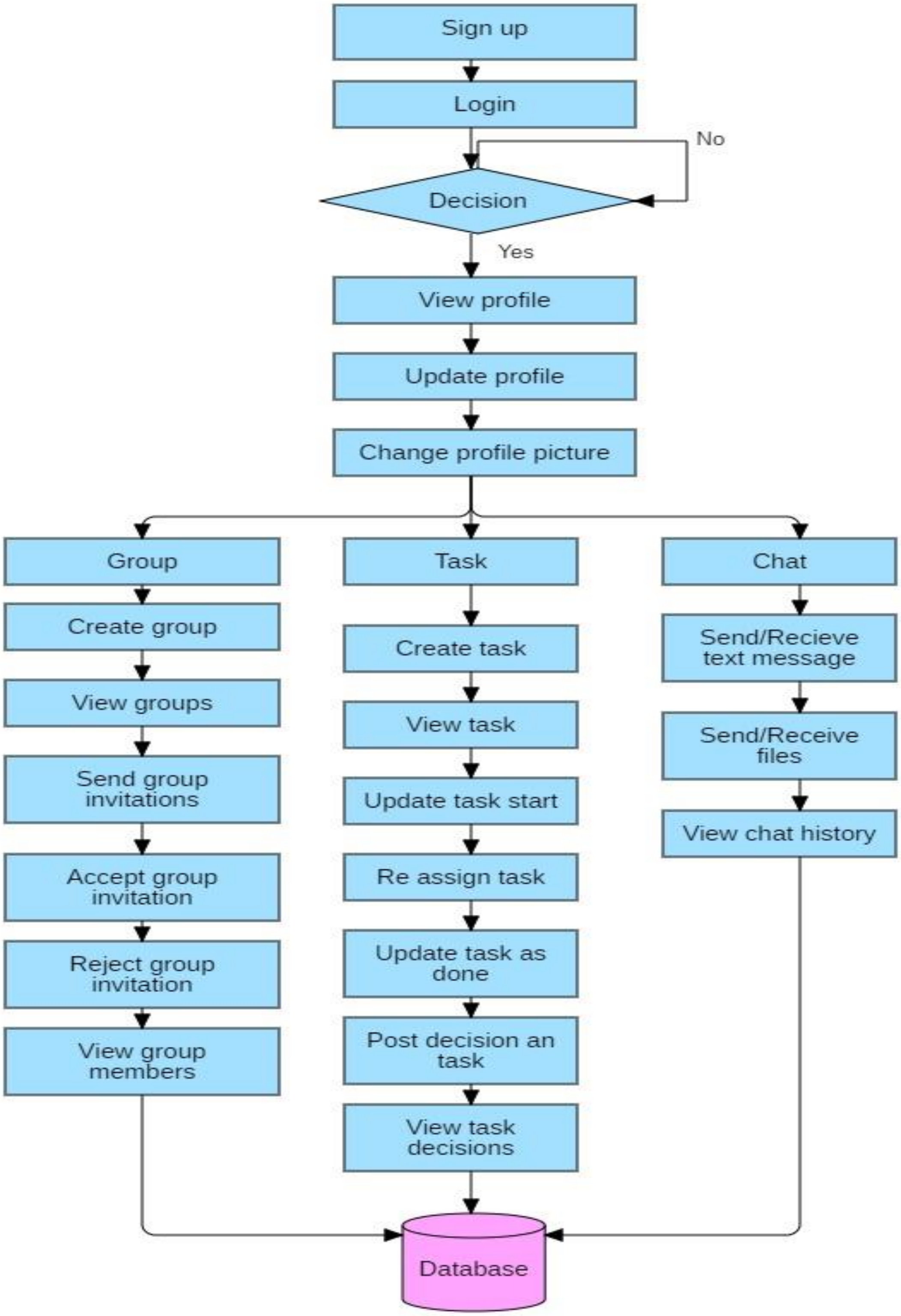
ii. Class Diagram



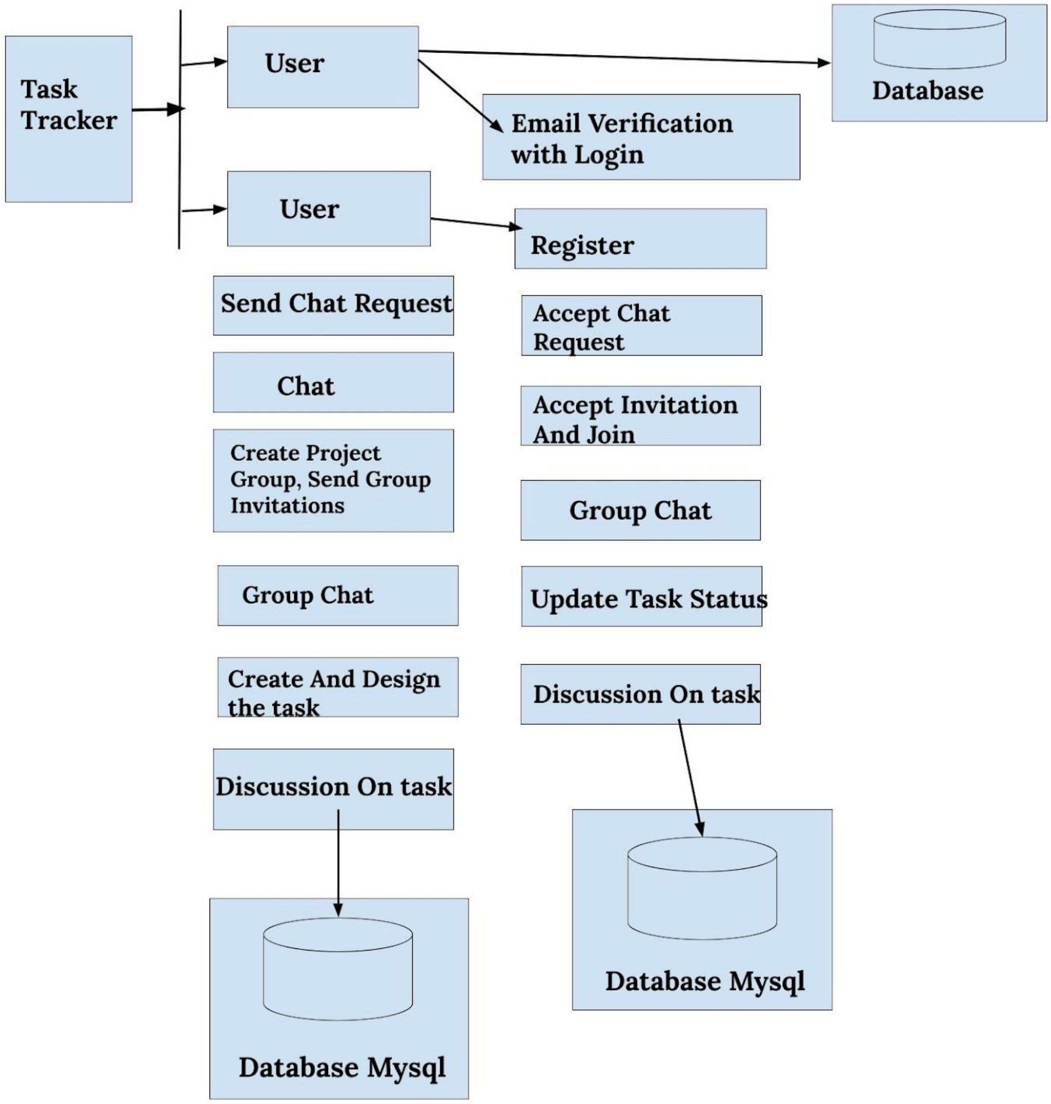
iii. Sequence Diagram


Group Member
Group
Task
Database
1 : Sign up
2 : Login
3 : Create group
4 : Invite to join
5 : Accept group invitation
: Reject group invitation
7 : View group members
8 : Create task
9 : View assigned task
10 : Update task status
11 : Re assign task
12 : Mark task as done
13 : Post decision on task
14 : View decisions
15 : View profile
16 : Update profile
17 : Change profile picture
18 : Chat with group members
19 : View chating

iv. Flow Chart Diagram



v. Workflow Diagram



4. Relationship between Components

S.No	**Module Name:**	**Template**	**URL**	**DB Table**	**Code Logic**
1	Create Group	createGroup.jsx	@postMapping(create group)	Groups	Here the user will create the group for users who are registered into the website by clicking the group link the users will directly navigate to the group page.
2	User Registration	userRegistration.jsx	@post Mapping(Registration)	Users	After creating the user registration the user will fill the form of registration and click on the register button then the page

					redirects to the login page.
3	User Login	userLogin.jsx		Users	Here the user will login into the website. With registered details, after entering the details the page is redirected to the homepage of users.
4	Group Invitations	groupInvites.jsx	@Get Mapping(Group invitations)	Invitations	Here the developer will do the business logic to develop the code logic to send the invitations to team members and maintain the group details.
5	Assign Task	task.jsx	@post Mapping(assign task)	Task	The assigned task will be stored in

					the tasks table and the task details will be placed and maintained by the developer that helps the developer to develop.
6	Post Task Discussion	groupChat.jsx	@Post Mapping(task discussion)	Task Discussions	The discussions will be created and send to users and post the discussions to users and may add the informations to website.
7	Update Profile	editProfile.jsx	@get Mapping(update profile)	Users	The profiles will be updated by users themselves and maintain the

					profile should be clear.
8	Reassign Task	taskDetails.jsx	@get Mapping(re assign task)	Task	The user will send the task and maintain the assigned task will be reassigned to users. That code will be assigned to users.

5. Project Implementation

User management:

User management will provide the user to login into the website and maintain the user details into the website. And the database will add the amount of data within the task table. The user will add and enter the website within the Users can register and log in securely to access the task management system. After registration the user will create the task and create the group for team members and invite them by sending the invitations and then the user will have to send the invitations and get it accepted by users.

Task management:

The task management will manage and send the task to users and, after accepting the invitation of the groups the user will must get into the website and send the tasks to users, the team members will do the work with the task management will send the requests to avoid the amount of the task. And the task management will do the and manage the team updates and task deadline. The task management will work on the page they sent in the group requests.

Chat management:

The chat management will manage the team to communicate with the team and interlink the persons to do the project in the proper way. Through the chat the team member will track the work progress and do communication with team members. After accepting the users group invitation and chatting with the team members about task progress and for queries also updates.

Discussion management:

The discussion management will discuss the team members work and project. The discussion between the management will put the task on the target and discuss the task and maintain the process in line. The discussions will be clear and maintained by the discussions.

Group Management:

The group will create and send the group requests to group members and invite them into the discussion. The user will send the task and maintain the assigned task will be reassigned to users. That code will be assigned to users and manage the group tasks.

The implemented steps to maintain the process very clearly and maintained by the users and team members.

We used the following technologies when constructing this application: HTML, CSS, React, and bootstrap for the frontend, Spring and MySQL for the backend development and data storage. To build the application code stable. We utilize a Spring Suite Tool to write the code. Data will also be stored and saved in the database MySQL workbench.

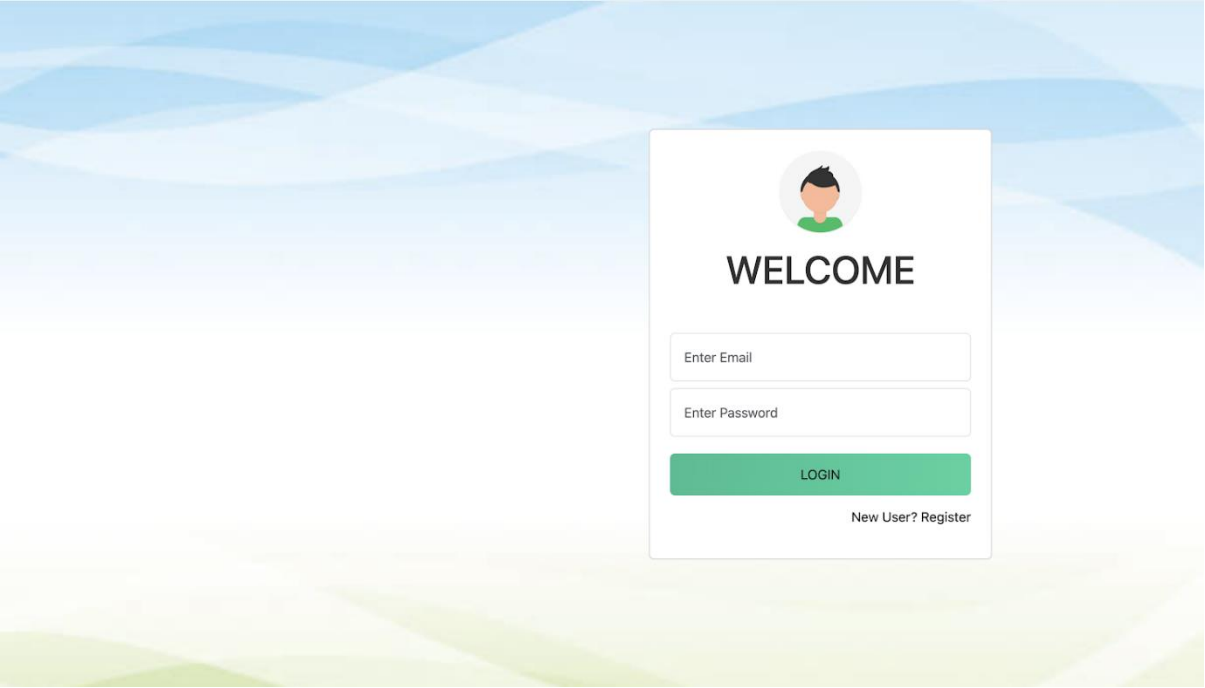
The database and other implementations will be clearly mentioned and started to work on the task. Users can filter the tasks with filtering options based on status, due date, and assignee.

Real-time Updates: Users receive real-time updates on task status changes, ensuring effective collaboration.

Microservices Architecture: The system will be built on a microservices architecture, separating concerns for user management, task management, and authentication.

6. Project Functionalities

Website Homepage:



WELCOME

Enter Email

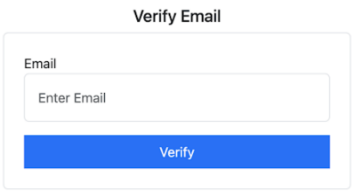
Enter Password

LOGIN

New User? Register

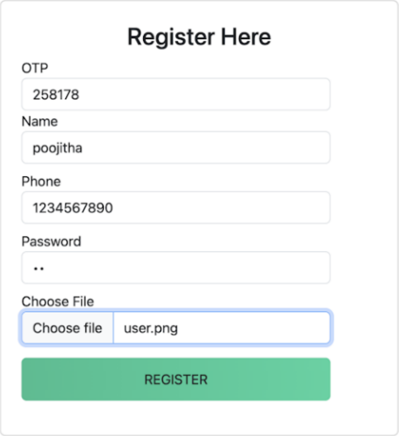
The above screenshot depicts the homepage of the website **Task Tracker** where existing users can login into website with their credentials such as email and password which were used to register their account. New users have an option to register their account.

Email Verification:



To Register for an account, users need to verify their email address like above. After User enters their email id and click verify, the user will receive the **One Time Password** (OTP). With that **verification** code users will verify their email account which allows the registration process to begin.

User Registration:



Register Here

OTP: 258178

Name: poojitha

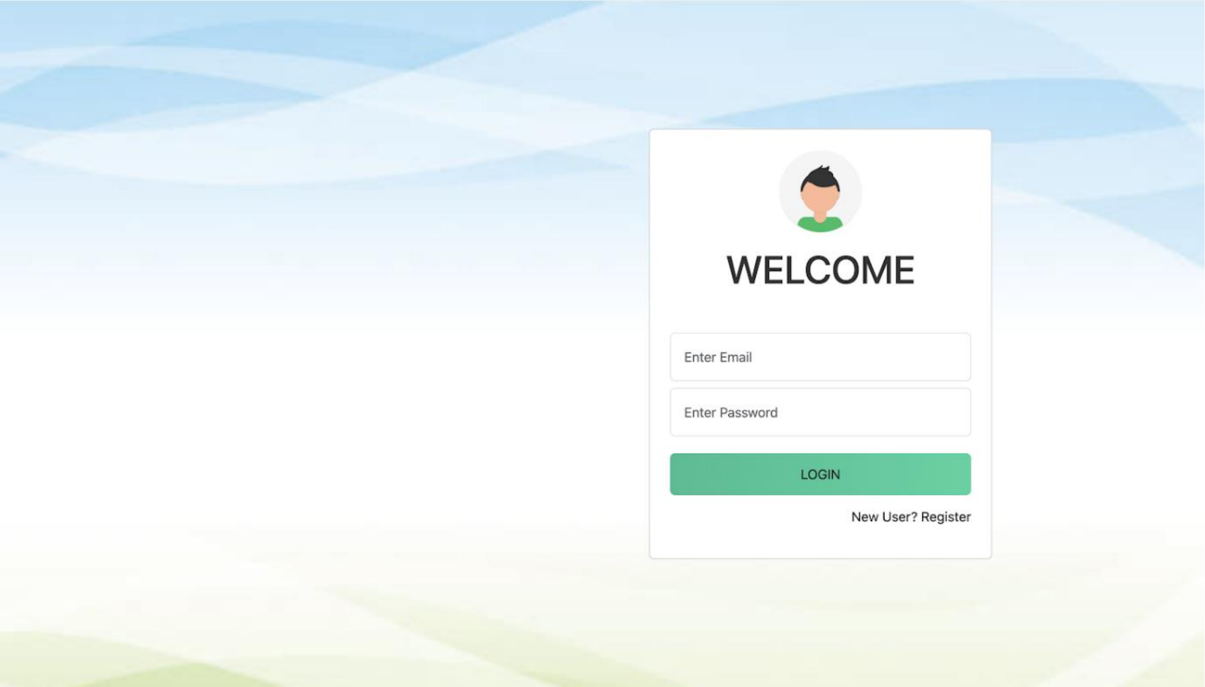
Phone: 1234567890

Password: ••

Choose File: user.png

After email verification the user will be redirected to registration page on the website where they can enter their personal information like **name**, **email**, **phone** number, **password** and upload **profile picture** along with their OTP for verification. Finally, user can click **register** button for registration.

Login Page:



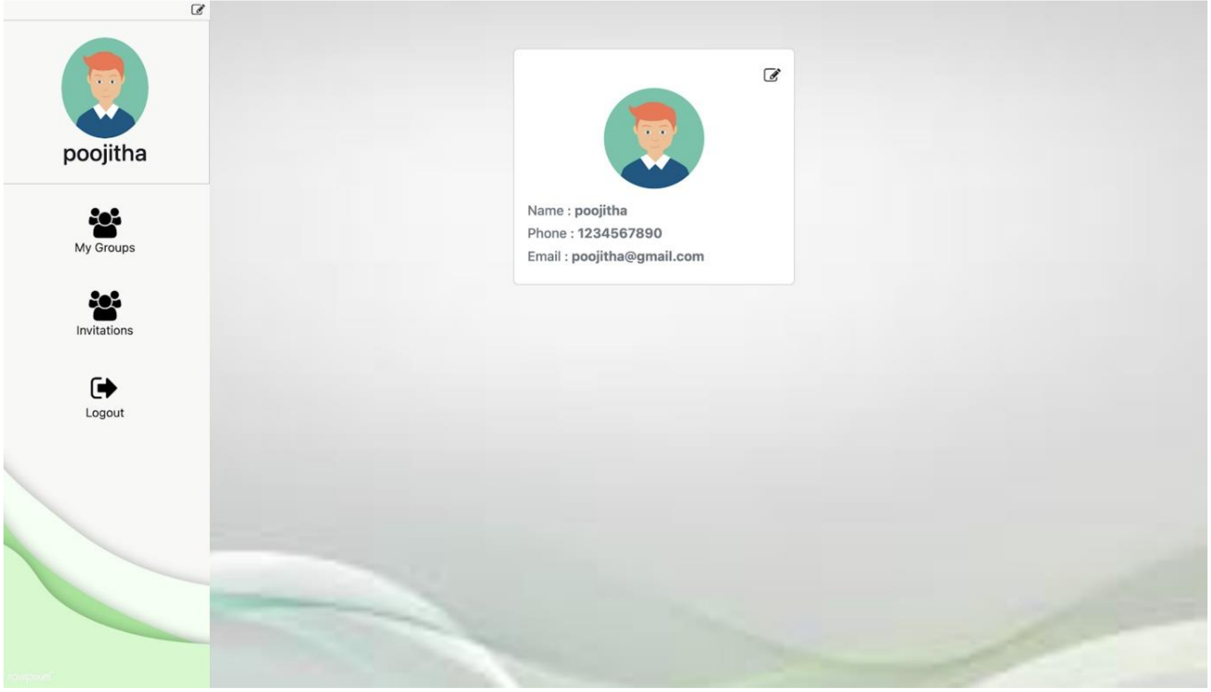
WELCOME

Enter Email

Enter Password

After successful Registration the users will be redirected to login page on the website, where user will login using their credentials like **email** and **password** and click **Login** button.

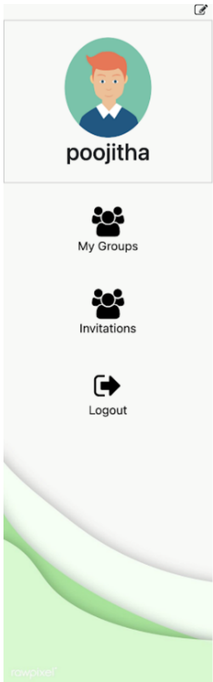
User Homepage:

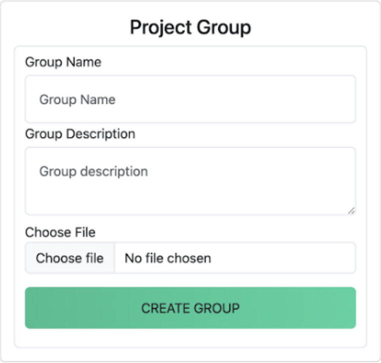


Name : poojitha
Phone : 1234567890
Email : poojitha@gmail.com

After logging into the website, the user will enter the homepage. Here the user's profile will be displayed. Their information like name, phone number, email address and their profile picture are displayed. Also, there is an **edit** option provided on the top right corner to edit their **information** and upload a new profile picture. Additionally, there is a menu on the left side of the screen where users can view the **dashboard** of groups or projects, can see pending **invitations** for the groups and **logout** option.

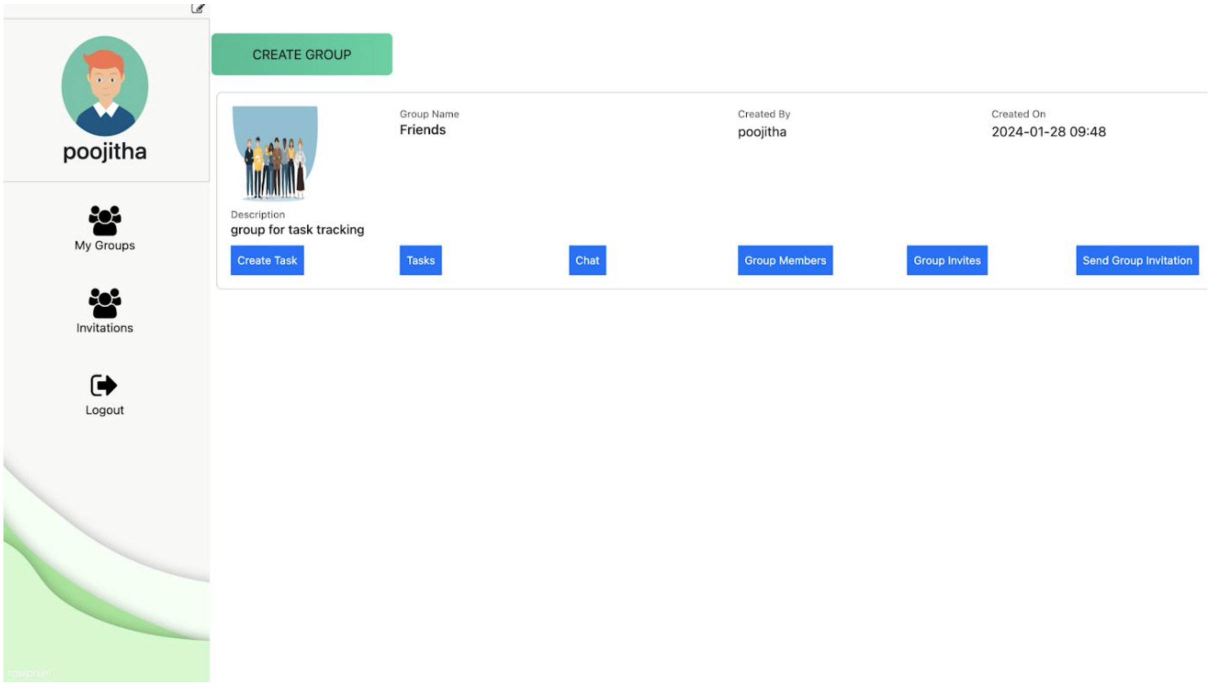
Creating Group:





In My Groups Dashboard the users will have an option **create group** for creating a new project group which helps to communicate with the team members and to track the progress of work of a project. While creating a new project group the user should give a group **name** and its **description** and group **logo**. A new project group will be created after clicking create group button.

Group Dashboard:



Group Name
Friends

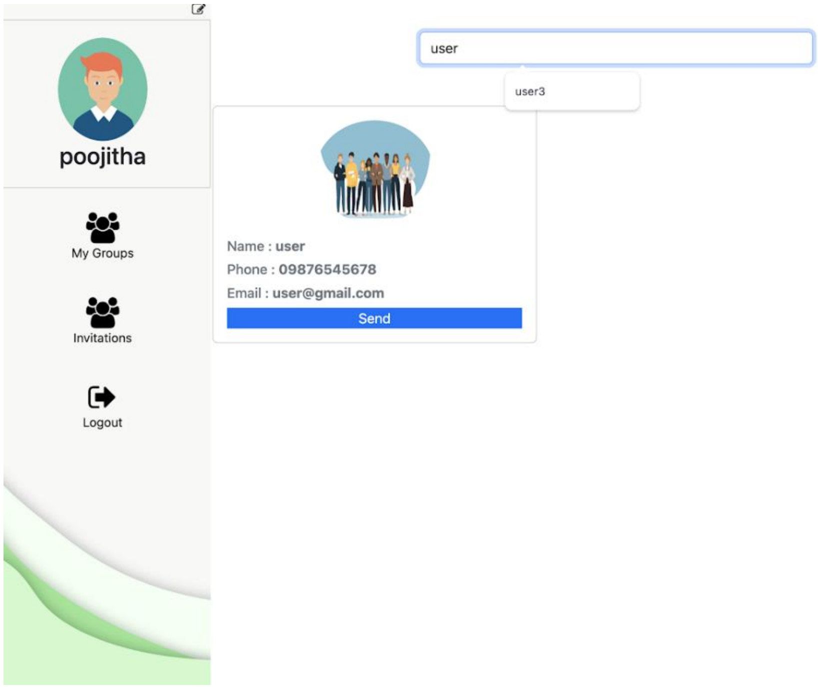
Created By
poojitha

Created On
2024-01-28 09:48

Description
group for task tracking

After creating the group, the user can view the created group in My Groups dashboard with details like group name, description logo, created by and the date when it was created. The user who created the group will send the invitation to all the team members involved in that specific project by using **Send Group Invitation** option. The list of team members who are invited into that group and yet to accept their invitations can be viewed by clicking **Group Invites** button. And all the team members who accepts the invitation and joins the group will be listed in **Group Members** option. These group members will have access to create the tasks using **Create Task** option and assign the tasks to team members and communicate about the project through the **Chat** option provided. All the tasks create with that group are listed in **Tasks** option where group members can analyze the progress of the project.

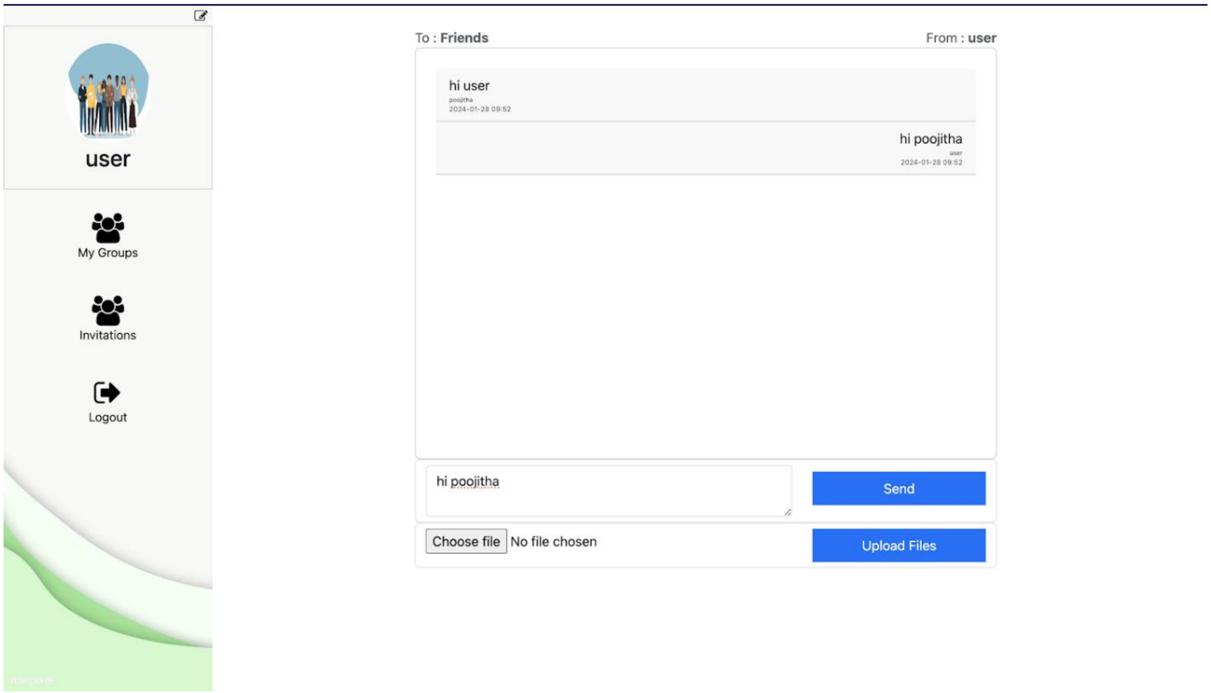
Sending group invitations:



Name : **user**
Phone : **09876545678**
Email : **user@gmail.com**

By clicking the Send Group Invite button in the project group dashboard, the user can send the invitations to the other team members involved in that project. It can be done by searching their name or email in the **search bar**, the list of suggested users will be shown where the user can click the **send** button for the selected user. The invitations are sent to the invited team members.

Group Chat:



To : **Friends** From : **user**

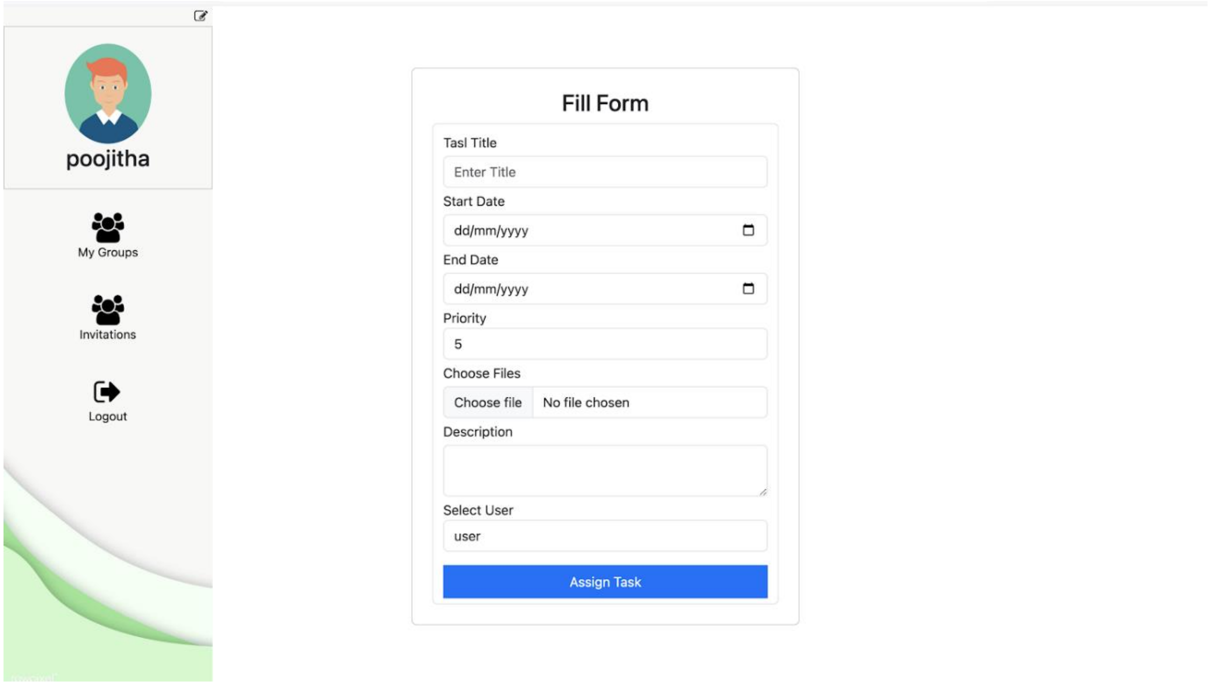
hi user
poojitha
2024-01-28 09:52

hi poojitha
user
2024-01-28 09:52

hi poojitha

Through the **chat** option provided in the group dashboard the user will be able to communicate with all the team members by sending **messages** and can **upload files** related to project or other tasks.

Create and Assign Task to Team members:



Fill Form

Tasl Title
Enter Title

Start Date
dd/mm/yyyy

End Date
dd/mm/yyyy

Priority
5

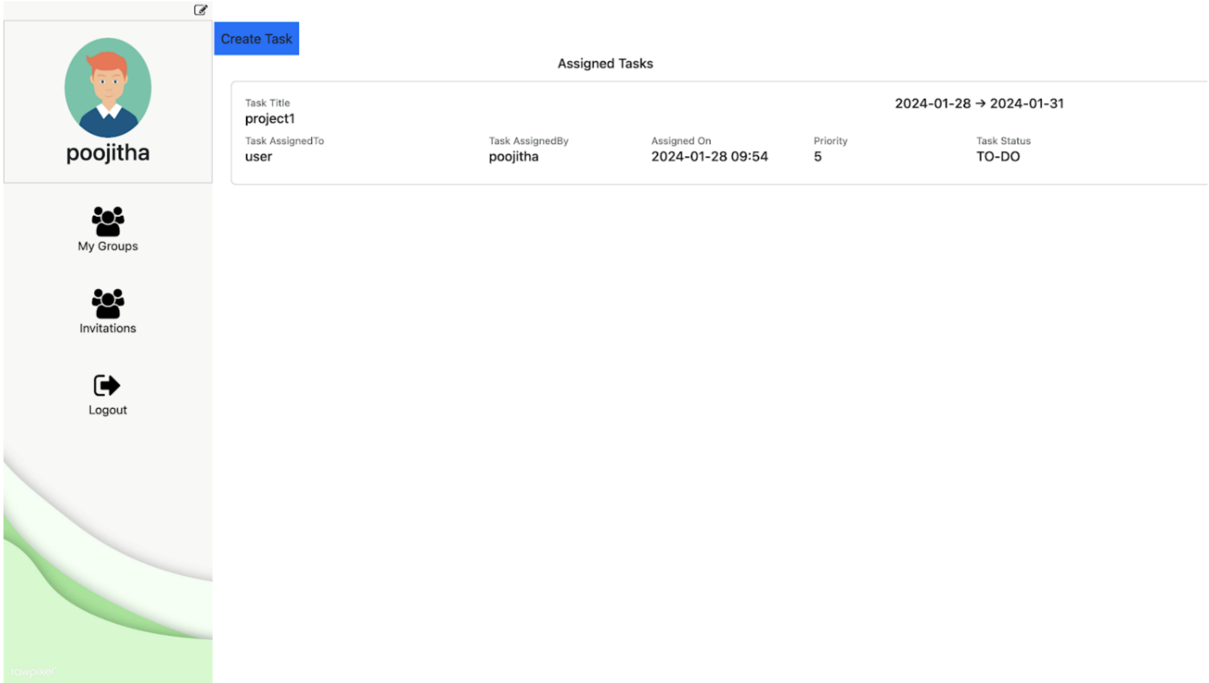
Choose Files
Choose file No file chosen

Description

Select User
user

By clicking Create Task button in Group dashboard, the user can create a new task and assign the task to team member while creating that task. A form will display right after clicking create task button, where the information is entered about the task like task **title**, start date, **end date** as a deadline, **priority** of the task (from 1 to 5, 1 as a high priority), description, upload required document and assign the task to a selected team member. After entering all the data, click **Assign Task** button.

View Assigned Tasks:



Assigned Tasks

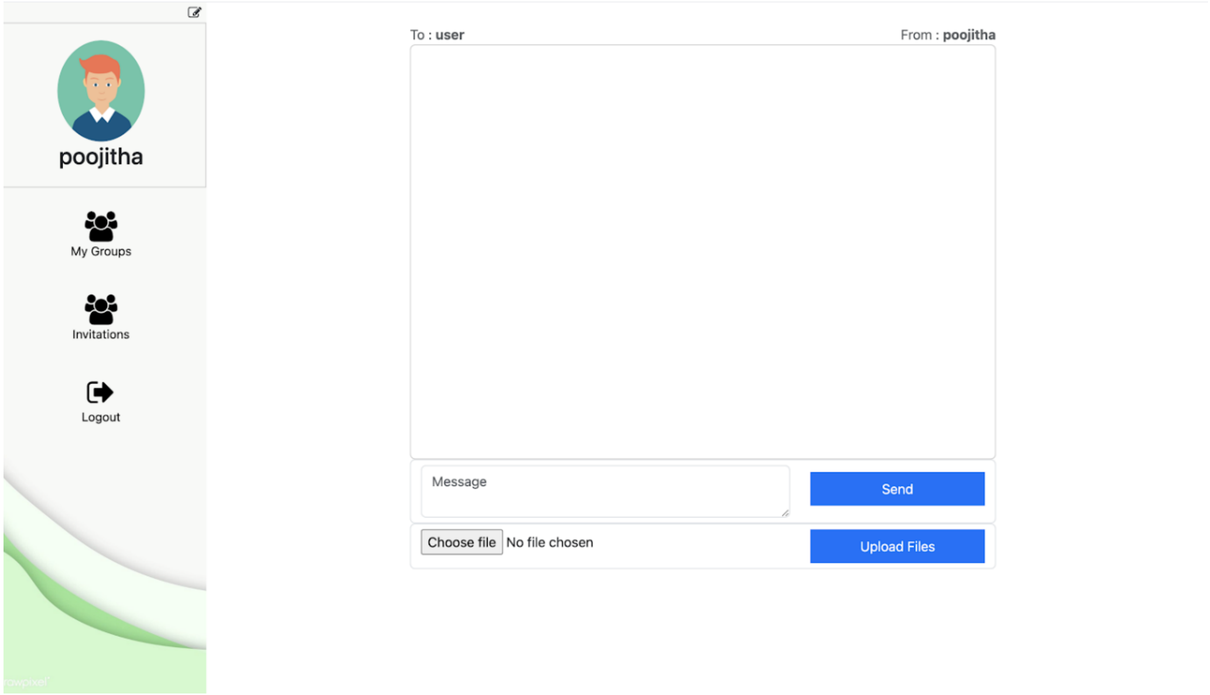
Task Title
project1

2024-01-28 → 2024-01-31

Task AssignedTo	Task AssignedBy	Assigned On	Priority	Task Status
user	poojitha	2024-01-28 09:54	5	TO-DO

After creating the task, it can be viewed in Tasks option in group dashboard or assigned tasks option provided in the Assigned team member section in the Group Members option. The details like task title, start date, **deadline**, **priority**, assigned by, assigned to, assigned on and status are shown. Let us consider the user who created the task as team lead and the user who is assigned to task as a team member. Initially after creating the task the status of the task is **TO-DO**. By clicking the task title, it is further displayed, where description, related files, and chat discussion area are shown where team lead and team members of the task and other team members of the project group can comment on the tasks. And a button will be displayed to team members like move to **In-Progress** or move to **QA** which changes the status of the task from TO-DO to In-Progress and In-Progress to QA. This is option is given to all the team members of the project group. On the other hand, the team lead user will have a different options like **Report a Bug** and move to **Done**. Through this a bug can be raised by using Report a Bug option and it is shown in the comments section. By doing this the status of task again changes from QA to In-Progress. The team member will work on the bug and again move the task to QA, and this process will continue until the team lead is satisfied about the work. And finally, the task can be moved to DONE state by the team lead.

Individual Chat:



To : user

From : poojitha

In Group Members tab, which is provided in the group dashboard, the team members profiles will be displayed. For each profile or team member we have options to view their tasks through **Assigned Tasks** option and a **Chat** option is given to chat with them. This chat will be personalized to the user and that selected team member. The team members can send messages and files personally to each other for a better understanding and communication.

7. Deployment Steps

There are two ways that to implement this project into the system throughout the deployment process.

a. Manual Process

For manual deployment process, the user needs to install the required software to run the project in their system.

1. Java

   Get the Java Development Kit (JDK) from a reliable source or the official Oracle website.

   To finish the installation procedure, launch the downloaded installer and adhere to the on-screen directions.

   Configure the JAVA_HOME environment variable to point to the JDK installation path following a successful installation.

2. An IDE like IntelliJ or Eclipse

   Visit IntelliJ or Eclipse's official website to download the installation package that is appropriate for your operating system.

   Follow the installation wizard's instructions when you launch the installer that you downloaded.

   After it's finished, open the IDE, and adjust any further settings to suit your needs.

3. MySQL Workbench

   To get the MySQL Workbench installer, go to the MySQL Community Edition download page or the official MySQL website.

Run the installer file that you downloaded to start the installation procedure.

To configure MySQL Workbench, including configuring the connection to your MySQL server, follow the installation prompts.

Once the installation has completed successfully, open MySQL Workbench and start creating the required tables for this project.

4. Spring Tool Suite

   First search the spring tool suite4 click on the IDE you're using and download it.

   Extract the zip file and install the STS.

   Spring tool 4 launcher box will appear there and click on launch button, it launches the STS.

Steps to Project Import:

First, we must unzip the folder containing the project code.

1. Open Spring Tool Suite and go to file and
2. Click on import then select Existing maven project.
3. Browse the root directory and select the extracted project folder and click on the finish button the project will be imported.
4. Now open the workbench and create the database, which is mentioned in the project.
5. After creating the database, now go to the property management applications page and right click in that and click on run as and select "3 spring boot app" the project will be run in the console and tables will be created in the database.
6. Goto vs code and open the folder which is available with frontend code right click on the folder open in vs code.

7. Now take the terminal and run the npm start command and run the project.
8. Then the application will be opened.

b. GIT Deployment

First **login into git** with your personal login credentials like email and password.

1. After login the git dashboard will appear .
2. Click on your repository button and then the new repository button will choose.
3. In that repository give the project name on the folder field and scroll down and click on the create folder button.
4. Now go to file explorer and create new folder in that.
5. Copy before extracted code and paste on that new folder.
6. Now right click on the new folder just we created before and select the git bash option it will redirect to git and run the commands with email and password. Which are available in the repository.
7. After running the commands, the authentication will be done and succeed.
8. After authentication the link will be generated.

Link: https://github.com/pooji1/Task-Tracker

Copy this above URL and do the cloning process:

Cloning:

Using the cloning method, we can extract the code from the URL.

Cloning process:

1. Go to file explorer and create empty new folder.
2. Right click on that folder and git bash here option.
3. Run the commands git clone "paste here Link" and press enter.
4. After cloning the code will run command "ls".
5. Now the project data will be stored in that empty folder.

8. Conclusion

This project aims to create a robust, scalable, and user-friendly Simple Task Management System with microservices architecture. By leveraging Java, Spring Boot, React, and associated tools, we aim to deliver a solution that enhances task organization and collaboration for both individuals and teams. The microservices approach ensures modularity, scalability, and maintainability, making the system adaptable to evolving requirements.

The successful implementation of this project will provide users with an intuitive and efficient tool for managing tasks, thereby contributing to improved productivity and collaboration in various professional settings.