## **Project Title and Scope**

### **1.1. Project Title:**

**RetroHub: Retroboard and Discussion Forum for Team Collaboration**

### **1.2. Project Scope:**

RetroHub is a web-based collaborative platform designed to streamline feedback collection and discussions within teams, organizations, or educational groups. The system allows a **manager (team creator)** to create teams, invite members, and initiate retrospectives through a **retroboard**. Team members can submit **feedback** (anonymously or with identity) under categories such as *positive*, *negative*, or *improvement*. Each feedback item can link to a **dedicated discussion thread** to facilitate focused conversations rather than general chat.

The platform aims to:

* Promote structured retrospection and team dialogue
* Allow anonymous yet trackable feedback
* Support transparency and resolution through discussions
* Provide a simple UI for both team creators and members

## **2. User Requirements**

### **2.1. Functional Requirements:**

* **User Signup/Login**
  + All users can register and log in.
* **Team Creation**
  + Managers can create a team and assign it a unique name and description.
* **Team Invitation**
  + Managers can invite members via email.
* **Team Membership**
  + Invited users can join teams after signup or directly via invite link.
* **Retroboard Creation**
  + Managers create a retroboard for the team.
* **Feedback Submission**
  + Members can submit feedback categorized into positive*,* negative*,* and improvement*.*
  + Feedback can be submitted anonymously.
* **Discussion Forum**
  + Each feedback card can be linked to a dedicated discussion thread.
  + Members can post messages in discussions.
* **Role Management**
  + A user can be a manager in one team and a member in another.

### **2.2. Non-Functional Requirements:**

* The system must be responsive and accessible via modern browsers.
* Feedback data should be securely stored and protected.

## **Module Specifications**

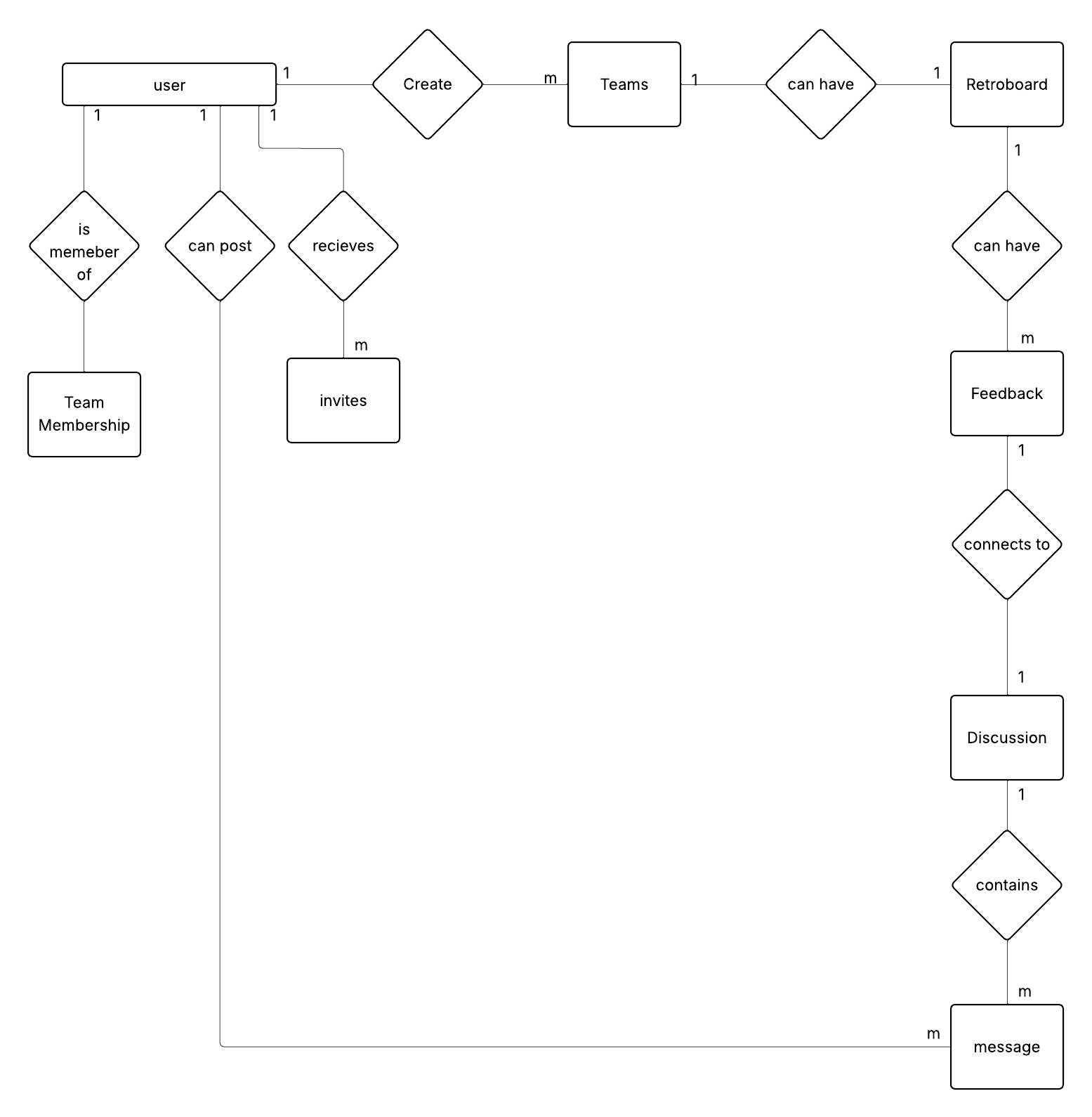
| **Module** | **Description** |
| --- | --- |
| **User Management** | * Manages user sign-up, login, and session tracking. * All users are stored in a common User table, and can act as team creators or members. |
| **Team Management** | * Allows managers to create teams and stores team details. * Each team is linked to the user who created it. |
| **Invitation System** | * Enables managers to invite team members via email. * Tracks invitation status (pending, accepted, rejected) for each user and team. |
| **Team Membership** | * Manages the many-to-many relationship between users and teams. * Tracks user roles (manager or member) and join timestamps. |
| **Retroboard Module** | * Each team can have one or more retroboards created by the manager. * Retroboards act as a container for categorized feedback. |
| **Feedback System** | * Allows team members to post categorized feedback (*positive*, *negative*, *improvement*), optionally anonymously, linked to a retroboard. |
| **Discussion Module** | * Allows users to start a discussion on any feedback item. * Ensures every feedback can optionally link to a discussion thread. |
| **Messaging System** | * Enables team members to participate in threaded discussions by posting messages. * Each message is linked to a user and a discussion. |
| **Access Control** | Ensures users access only the teams and features relevant to their roles. Manages visibility and permissions based on membership type. |

## **4. Technologies to be Used**

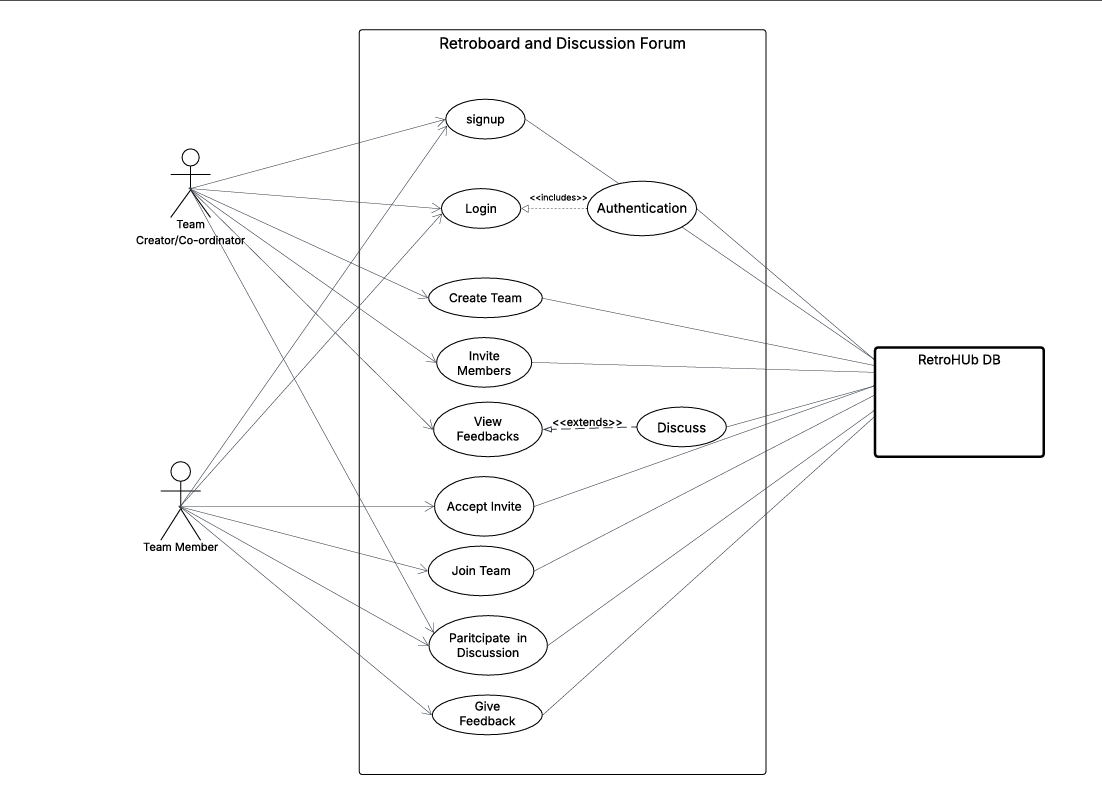
| **Category** | **Technology** |
| --- | --- |
| **Frontend** | React.js |
| **Backend** | Java Servlets, JSP |
| **Database** | PostgreSQL |
| **ORM/Database Access** | JDBC |

**5. Diagrams:**

**5.1. ER Diagram**



**5.2. Use Case Diagram:**

****

**6. Data Dictionary**

**1. User**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| User | user\_id | UUID / INT | Primary Key, Not Null, Unique |
|  | name | VARCHAR(100) | Not Null |
|  | email | VARCHAR(150) | Not Null, Unique |
|  | password | VARCHAR(255) | Not Null |
|  | created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**2. Team**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Team | team\_id | UUID / INT | Primary Key, Not Null, Unique |
|  | team\_name | VARCHAR(100) | Not Null, Unique per manager |
|  | description | TEXT | Optional |
|  | created\_by | UUID / INT | Foreign Key → User(user\_id) |
|  | created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**3. TeamMembership**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| TeamMembership | membership\_id | UUID / INT | Primary Key, Not Null, Unique |
|  | user\_id | UUID / INT | Foreign Key → User(user\_id), Not Null |
|  | team\_id | UUID / INT | Foreign Key → Team(team\_id), Not Null |
|  | role | ENUM | Values: 'manager', 'member', Not Null |
|  | joined\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**4. Invite**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Invite | invite\_id | UUID / INT | Primary Key, Not Null |
|  | email | VARCHAR(150) | Not Null |
|  | team\_id | UUID / INT | Foreign Key → Team(team\_id), Not Null |
|  | invited\_by | UUID / INT | Foreign Key → User(user\_id), Not Null |
|  | status | ENUM | 'pending', 'accepted', 'rejected' |
|  | invited\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**5. Retroboard**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Retroboard | retroboard\_id | UUID / INT | Primary Key, Not Null |
|  | team\_id | UUID / INT | Foreign Key → Team(team\_id), Not Null |
|  | title | VARCHAR(100) | Not Null |
|  | created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**6. Feedback**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Feedback | feedback\_id | UUID / INT | Primary Key, Not Null |
|  | retroboard\_id | UUID / INT | Foreign Key → Retroboard(retroboard\_id) |
|  | user\_id | UUID / INT | Foreign Key → User(user\_id), Nullable if anonymous |
|  | type | ENUM | 'positive', 'negative', 'improvement' |
|  | message | TEXT | Not Null |
|  | anonymous | BOOLEAN | Default: FALSE |
|  | created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**7. Discussion**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Discussion | discussion\_id | UUID / INT | Primary Key, Not Null |
|  | feedback\_id | UUID / INT | Foreign Key → Feedback(feedback\_id) |
|  | created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |

**8. Message**

| **Entity** | **Attribute** | **Data Type** | **Constraints** |
| --- | --- | --- | --- |
| Message | message\_id | UUID / INT | Primary Key, Not Null |
|  | discussion\_id | UUID / INT | Foreign Key → Discussion(discussion\_id) |
|  | user\_id | UUID / INT | Foreign Key → User(user\_id) |
|  | message\_text | TEXT | Not Null |
|  | posted\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP |