



**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
PULCHOWK CAMPUS**

**OOAD Case Study Report**



**Submitted By:**

Nisan Shrestha

Pramish Aryal

Pranab Ratna Gubhaju

Group: 076-BCT-AB

(076BCT040)

(076BCT046)

(076BCT048)

**Submitted To:**

Department of Electronics and  
Computer Engineering

# Undergraduate Project Database System

## Introduction:

Undergraduate Project Database System is a project aiming to act as a central archival system for Projects that are submitted by the students of IOE in their course of study.

The system will act as a centralized database for storing references to Past Projects completed and theses submitted.

## Functional Requirements:

### R.1: Login

**Description:** Login function allows users to enter their Username or Email (provided to them by their institution) and Password that they have used to register to verify their identity. It checks whether the entered Email or Username and password are valid or not and returns appropriate validation message or error and session token for successful logins. This option is only available when there is no active logged in session running.

#### R.1.1: Select Login Option

- *Input:* The login option has been selected.
- *Output:* User is prompted to enter Username/Email and password.

#### R.1.2: Request Login

- *Input:* User has entered the Username/Email and Password.
- *Output:* Upon validation display login successful message, other options such as Create projects, Update Projects, Delete Project, Create Team and Join Team become available. Incase of unsuccessful validation, the user is prompted with an error message.
- *Processing:* The entered Username/Email and Password are checked against the entries in the database and appropriate validation status is returned.

### R.2: Sign Up

**Description:** Sign Up feature allows users to create an account for our database. Users should provide valid username, email provided by their institution and strong password. After signing up, a verification mail will be sent to the registered email. The verified account is allowed to

access various features like add project, update project , delete project , create team and join other teams.

#### R.2.1: Select Sign Up

- *Input:* Sign up has been selected.
- *Output:* A prompt appears where the user has to enter username, email and password.

#### R.2.2: Sign up

- *Input:* User has to enter a valid username, email and password.
- *Output:* After data is provided, an output account is created but it has to be verified.
- *Processing:* After an account is created, a verification mail is sent to the registered email.

### R.3: Browse Projects

**Description:** Browse Project functionality allows users to view all verified projects. This feature is the main part of the system and is available to all users. It doesn't matter if the user is logged in or not.

#### R.3.1: Select Browse Project Option

- *Input:* Browse has been selected.
- *Output:* Gives list of all the projects that have "Verified Status". If the count is over 50, only the first 50 projects are shown at once.
- *Processing:* Access database of projects and return verified projects only.

### R.4: Create Project

**Description:** Create project allows users to create a new project entry by entering details like Project Team, Project Name, Project Supervisor, Project Description, Links and other necessary details. The created projects are added to the database and assigned pending status. This option is only available when the user is logged in.

#### R.4.1: Select Create Project Option

- *Input:* User has clicked on the Create Project option
- *Output:* The user is taken to a form page where they are prompted to enter the project details.

#### R.4.2: Submit Project Entry for Review

- *Input:* User has entered all the necessary information about the project and clicked on the submit button
- *Output:* The user is given a project added message.

- **Processing:** A new database entry is created and verification Review Pending status is assigned to it.

## R.5: Update Project

**Description:** Update project allows users to edit an existing entry of a project of which they are a team member. The user may edit any of the required details about the project and submit for review. The projects are moved to pending status if they were previously published. This option is only available when the user is logged in.

### R.5.1: Select Update Project Option

- *Input:* User has clicked on the Update Project option
- *Output:* The user is shown a list of all projects of which they are a team member.
- *Processing:* Projects by team that the user is part of are looked up in the database.

### R.5.2: Select Project to Edit

- *Input:* User has selected a project entry that they would like to edit.
- *Output:* A form displaying all the project details are shown. Users are allowed to make changes in the form.

### R.5.3: Submit Edits for Approval

- *Input:* User has pressed submit edits button.
- *Output:* A edits submitted message is shown to the user and redirected to the home page.
- *Processing:* The projects are assigned Pending Review Status and changes submitted by the user are reflected in the database entry.

## R.6: Delete Project

**Description:** Delete project allows users to delete an existing entry of a project of which they are a team member. This option permanently deletes the entry from the database. This option is only available when the user is logged in.

### R.6.1: Select Delete Project Option

- *Input:* User has clicked on the Delete Project option.
- *Output:* The user is shown a list of all projects of which they are a team member.
- *Processing:* Projects by team that the user is part of are looked up in the database.

### R.6.2: Select Project to Delete

- *Input:* User has selected a project entry that they would like to delete.

- *Output:* User is prompted to enter their password for verification. Upon successful validation, Project Successfully deleted message is shown.
- *Processing:* User's password is verified using a database lookup. Upon verification of password the selected project is deleted from the database.

## R.7: Filter Projects

**Description:** Filter Project allows users to filter through the projects in the database based on certain criteria. The filter project function would allow users to narrow down the list of projects by applying one or more filters. Users choose to filter according to department, year, college and project categories. Admin can filter verified and not verified projects. This feature is available for logged as well as not logged users.

### R.7.1: Select filter option

- *Input:* Criteria are chosen from the given select box available in filter option.
- *Output:* List of projects that meet the entered criteria and are verified are displayed.
- *Processing:* Access database of projects and display the projects where criteria chosen by the user is true.

## R.8: Search Projects

**Description:** Search project functionality allows users to search for projects based on their needs.

Search can be done based on project name, student name, supervisor name, year, project category, department and college. By implementing a search function, users can quickly and easily find the projects that are most relevant to their needs, without having to manually review the entire database. It will save time and improve the user experience.

### R.8.1: Input data in search bar and press search

- *Input:* Data to be searched is entered in the search bar.
- *Output:* List of projects that have information searched in the search bar are displayed.
- *Processing:* Our system accesses the project database and gives the project whose information matches with that given by the user in the search bar.

## R.9: Create Team

**Description:** Teams own projects and consists of one or more team members. A user can be a member of multiple teams. Create team allows users to create a new team by entering the team name. The user who requested the creation is automatically added to the team. A team code is generated upon creation that is unique and private to each team and can be used to join the team. This option is only available when the user is logged in.

#### R.9.1: Select Create Team Option

- *Input:* User has selected the create team option.
- *Output:* User is prompted to enter a team name.

#### R.9.2: Submit Team Name

- *Input:* User has submitted a team name for creation.
- *Output:* Team Creation Successful message will be displayed along with a generated unique Team Code, incase of error failure message is displayed.
- *Processing:* Create a new team entry with the user as member and generate a unique team code. Send success flag unless error is generated.

### R.10: Join Team

**Description:** Join team option allows users to join an existing team by using the private team code which is available to an existing team member.

This option is available only to logged in team members.

#### R.10.1: Select Join Team option.

- *Input:* User has selected the join team option.
- *Output:* User is prompted to enter a team join code.

#### R.10.2: Select Join Team option.

- *Input:* User has entered the team code and submitted.
- *Output:* Success message is displayed if the team join code is valid. If the team join code is invalid a join unsuccessful message is displayed.
- *Processing:* The entered team code is looked up on the database. If the team join code valid exists then a success flag is returned and the user is added to the team. If the team join code is invalid a join unsuccessful flag is returned.

### R.11: Delete Team

**Description:** Delete project allows users to delete an existing entry of a project of which they are a team member. This option permanently deletes the entry from the database. This option is only available when the user is logged in.

#### R.11.1: Select Delete Team Option

- *Input:* User has clicked on the Delete Team option.
- *Output:* The user is shown a list of all teams of which they are a team member.

- *Processing:* Teams the user is part of are looked up in the database.

#### R.11.2: Select Team to Delete

- *Input:* User has selected a team that they would like to delete.
- *Output:* User is prompted to enter their password for verification. Upon successful validation, Team Successfully deleted the message is shown.
- *Processing:* User's password is verified using a database lookup. Upon verification of password the selected team is deleted from the database.

### R.12: Moderate Team

**Description:** Moderate Team functionality allows administrators to manage team and team members. This functionality is to be used when there is a team that has no projects in its contribution for over a set period of time after registration. It allows administrators to tidy up the database.

#### R.12.1: Select Moderate Team option

- *Input:* Admin clicks on Moderate Team Option.
- *Output:* Teams with no contribution after a certain period of registration are listed.
- *Processing:* Find teams with no projects uploaded for over a certain period of time after registration from the database.

#### R.12.1: Select desired Team

- *Input:* Admin clicks on the desired team to be deleted.
- *Output:* Admin is prompted to enter the master password. On validation Team Deletion Success message is displayed and the team is deleted from the database, in case of unsuccessful validation, error message is displayed.
- *Processing:* Master password is verified and the selected team is deleted from the database. If the password is incorrect, error flags are returned.

### R.13: Moderate Projects

**Description:** Moderate Projects functionality allows administrators to manage project entries. This functionality is to be used when the projects submitted by the users are not upto the standards or there are incorrect data filled by the users. It allows administrators to tidy up the database and make sure only correct projects are published.

#### R.13.1: Select Moderate Projects option

- *Input:* Admin clicks on Moderate Project Option.
- *Output:* All Projects are listed with options to filter or search. [R.7, R.8]
- *Processing:* List projects uploaded with filters or search applied.

#### R.13.2: Select desired project

- *Input:* Admin clicks on the desired project to be deleted.
- *Output:* Admin is prompted to enter the master password. On validation Success message is displayed and the project is deleted from the database, in case of unsuccessful validation, error message is displayed. Project Rejection message is sent to project team members emails.
- *Processing:* Master password is verified and selected project is deleted from the database. If the password is incorrect, error flags are returned.

### R.14: Verify Projects

**Description:** Verify Projects functionality involves confirming the authenticity and accuracy of the information provided for projects in a database. This functionality is only available to the administrator. All projects provided by the team are verified by the administrator before it is available to the public. The administrator checks members name and their email, and team data to make sure that correct information is provided. Once a project is verified it will be published and it can be browsed by other users.

#### R.14.1: Select Verify Project Option

- *Input:* Admin clicks on Verify Project Option.
- *Output:* All Projects with Review Pending status are listed.
- *Processing:* Search for projects with Review Pending status and return them all.

#### R.13.2: Select Desired Project

- *Input:* Admin clicks on the desired project to be verified.
- *Output:* Selected project is verified by setting its status to Verified and success message is displayed, in case of failure an error message is displayed.
- *Processing:* The project selected is given a Verified status and success message is returned.



## **Non-functional Requirement:**

### **R.1: Performance Requirements**

**R.1.1:** The loading time of the user end home page, with at least a 8Mbps bandwidth, must not exceed 5 seconds.

**R.1.2:** Login information must be verified within 10 seconds.

**R.1.3:** Search and filter queries must be returned within 5 seconds.

### **R.2: Design constraints**

**R.2.1:** The end system must be available via any modern browser and available via the internet.

**R.2.2:** The system will be developed using Node.js with ejs which supports server-side rendering and uses MongoDB as the database.

**R.2.3:** The system must be deployable in any web server with a maximum of 16GB of ram and handle up to 200 concurrent users.

### **R.3: User Interface**

**R.3.1:** The graphical user interface (GUI) should have a consistent design throughout, ensuring that all elements of the interface have a similar appearance and function in a similar way.

**R.3.2:** Proper color should be assigned to call to action buttons.

### **R.4: Availability**

**R.4.1:** The system must be up and running around the clock with downtime not more than 0.01%.

**R.4.2:** The system must be accessible to everyone with an internet connection.

### **R.5: Security**

**R.5.1:** Administrators will have access to all modules of the system.

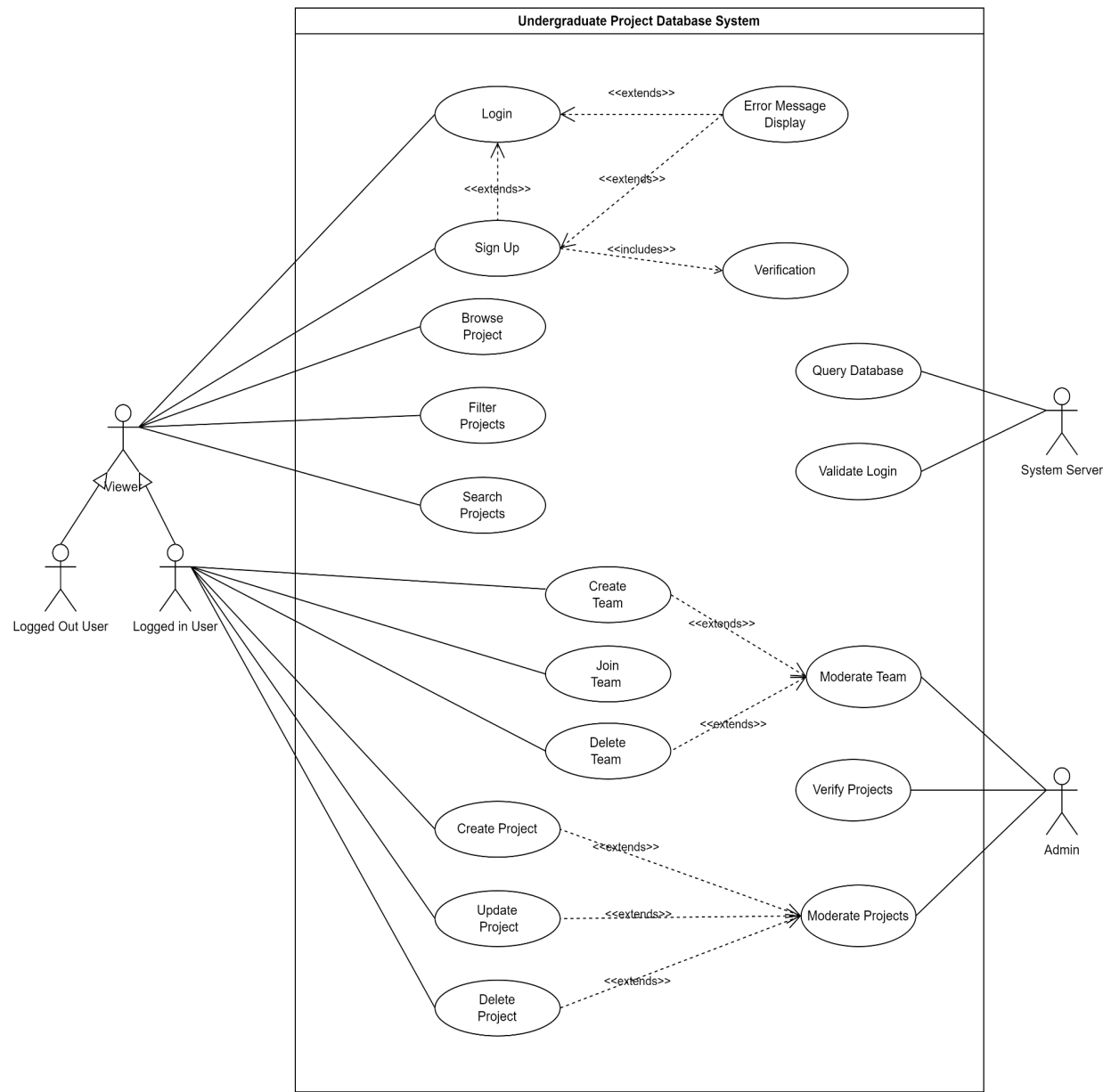
**R.5.2:** Projects created by one team can only be deleted and updated by team members of that project.

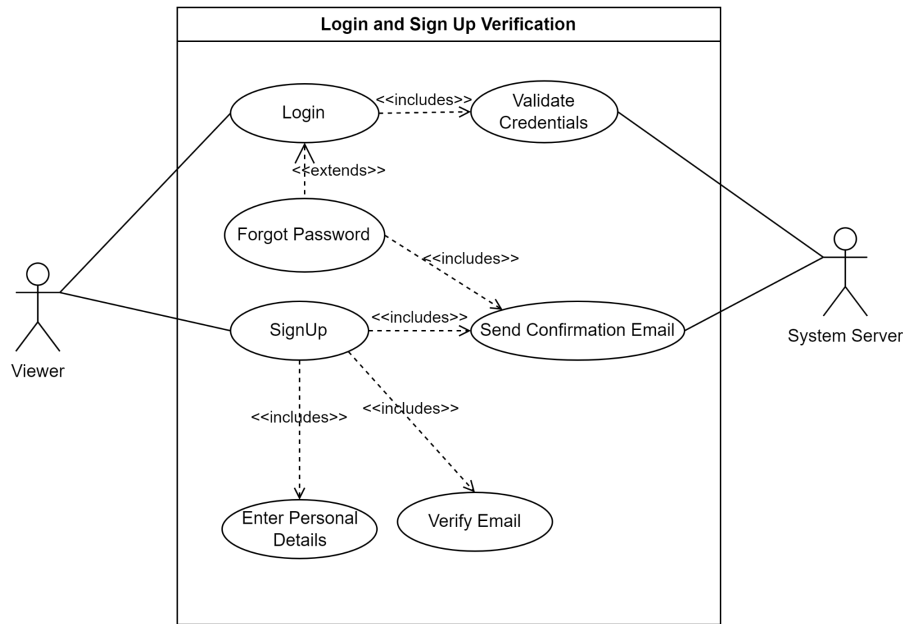
**R.5.3:** All delete and modify operations can only be carried out by logged in users.

**R.5.4:** User session token will be made invalidated after 7 days from the date of issue.

# Use Case Descriptions and Diagrams

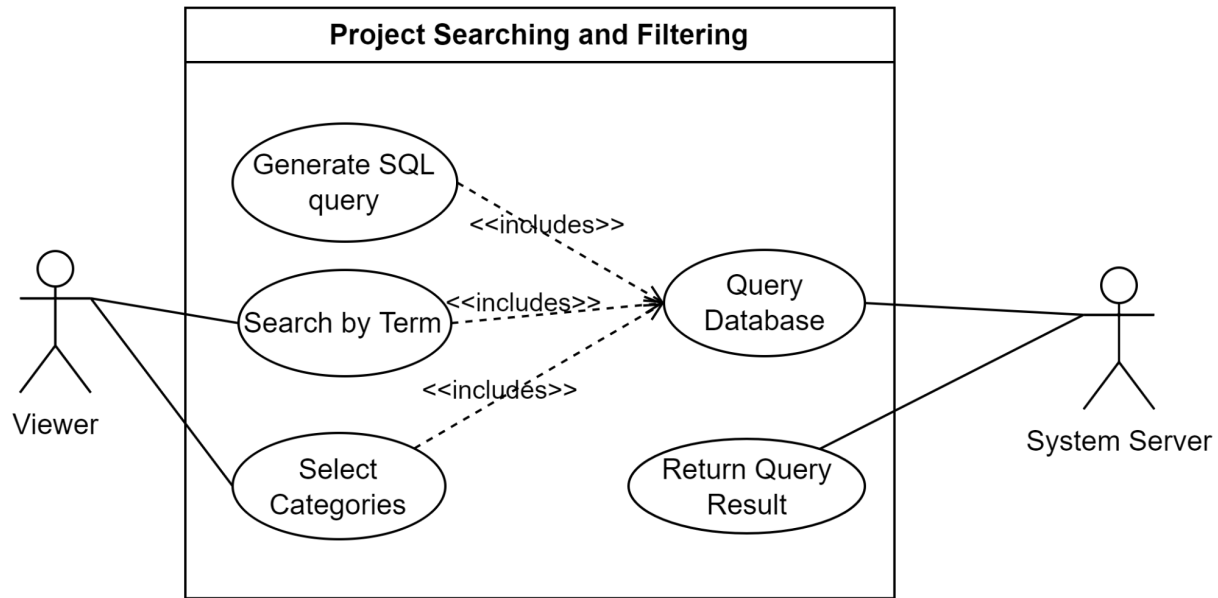
Overall Use Case Diagram:





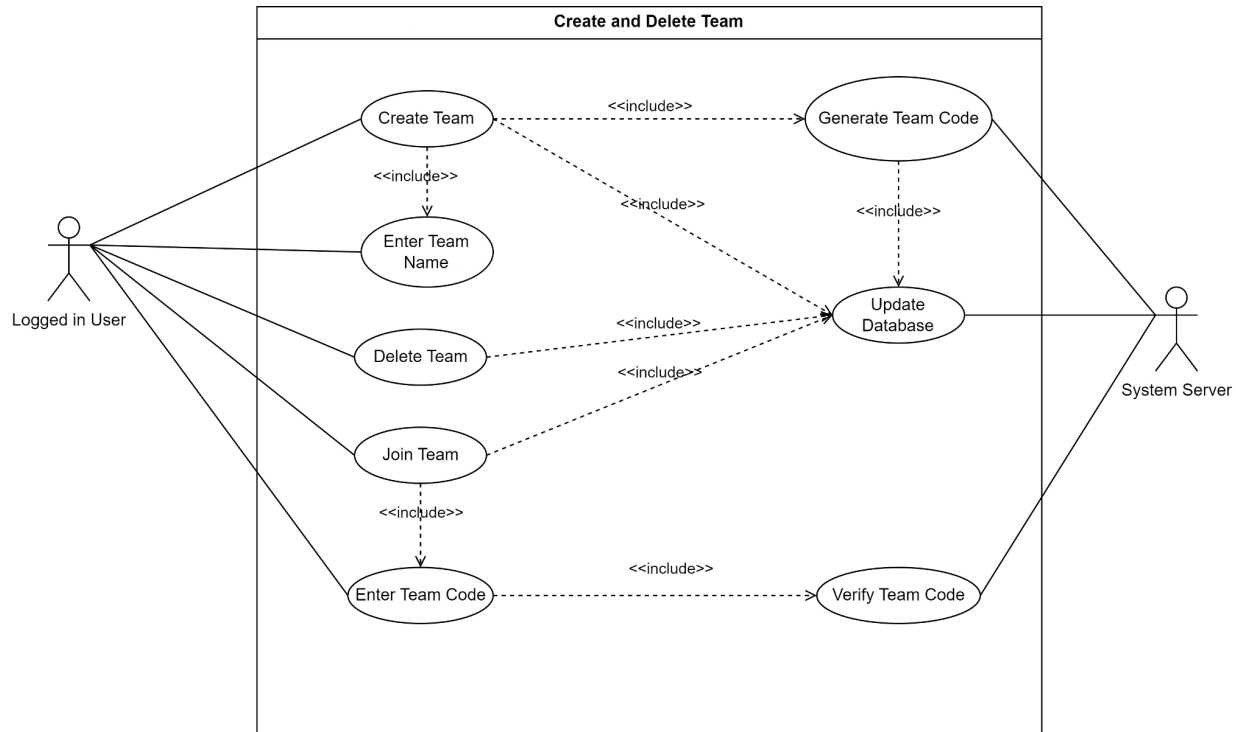
## Description:

<b>Use Case Name:</b>	Login/Sign Up Verification
<b>Participating Actors:</b>	Viewer, System
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• User has a valid institution email.</li> <li>• User has an access to the internet.</li> </ul>
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. User opens the Home Page and clicks on Login option.</li> <li>2. User enters their Email and Password.</li> <li>3. System checks the credentials and logs the user in if they are correct.</li> <li>4. If the user has forgotten their password, they can select the "Forgot Password" option and enter their email.</li> <li>5. System sends a password reset link to the provided email and the user can reset their password using the link.</li> <li>6. If the user does not have an account, they can select the "Sign Up" option.</li> <li>7. The User provides their personal information and an email confirmation link is sent.</li> <li>8. User verifies their email and creates a password to complete the registration.</li> </ol>



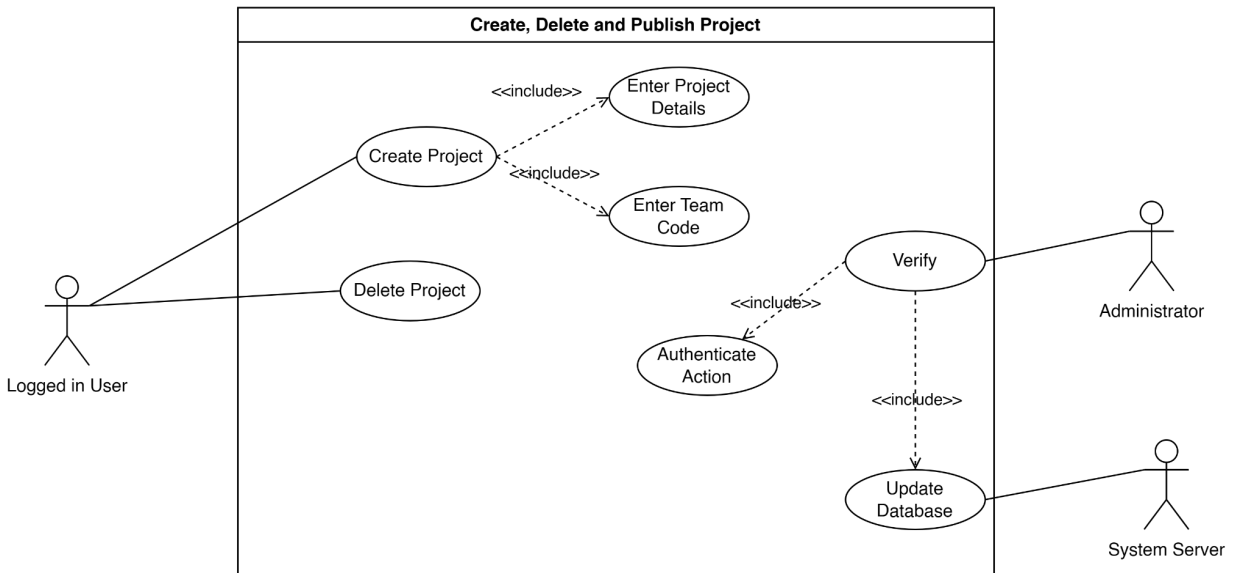
## Description:

<b>Use Case Name:</b>	Search and Filter Project
<b>Participating Actors:</b>	Viewer, System Server
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• User is on the home page.</li> <li>• The user has access to the internet.</li> </ul>
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. User opens the Home Page and clicks on the Search field.</li> <li>2. User enters a search term.</li> <li>3. System queries the database using a SQL query it generates.</li> <li>4. System returns a list of results to the frontend.</li> <li>5. Users may then choose categories to filter query results.</li> <li>6. System performs a second database query using a new SQL query.</li> <li>7. System returns the filtered query result.</li> </ol>



## Description:

<b>Use Case Name:</b>	Creating, Joining and Deleting Teams
<b>Participating Actors:</b>	Logged in User, System Server
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• User has selected the Teams Menu.</li> <li>• User is logged in.</li> </ul>
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. User opens the Teams page.</li> <li>2. User can Select to Create, Join or Delete existing Team.</li> <li>3. Upon selecting Create Team, User enters Team name.</li> <li>4. System generates unique team code and team entry in the database.</li> <li>5. System return the team code to the user.</li> <li>6. To join a team the user enters a team code in the join menu.</li> <li>7. System adds the member to the team in the database.</li> <li>8. Team deletion can be initiated by any member.</li> <li>9. Deleting removes the team entry form database and frees the team code.</li> </ol>

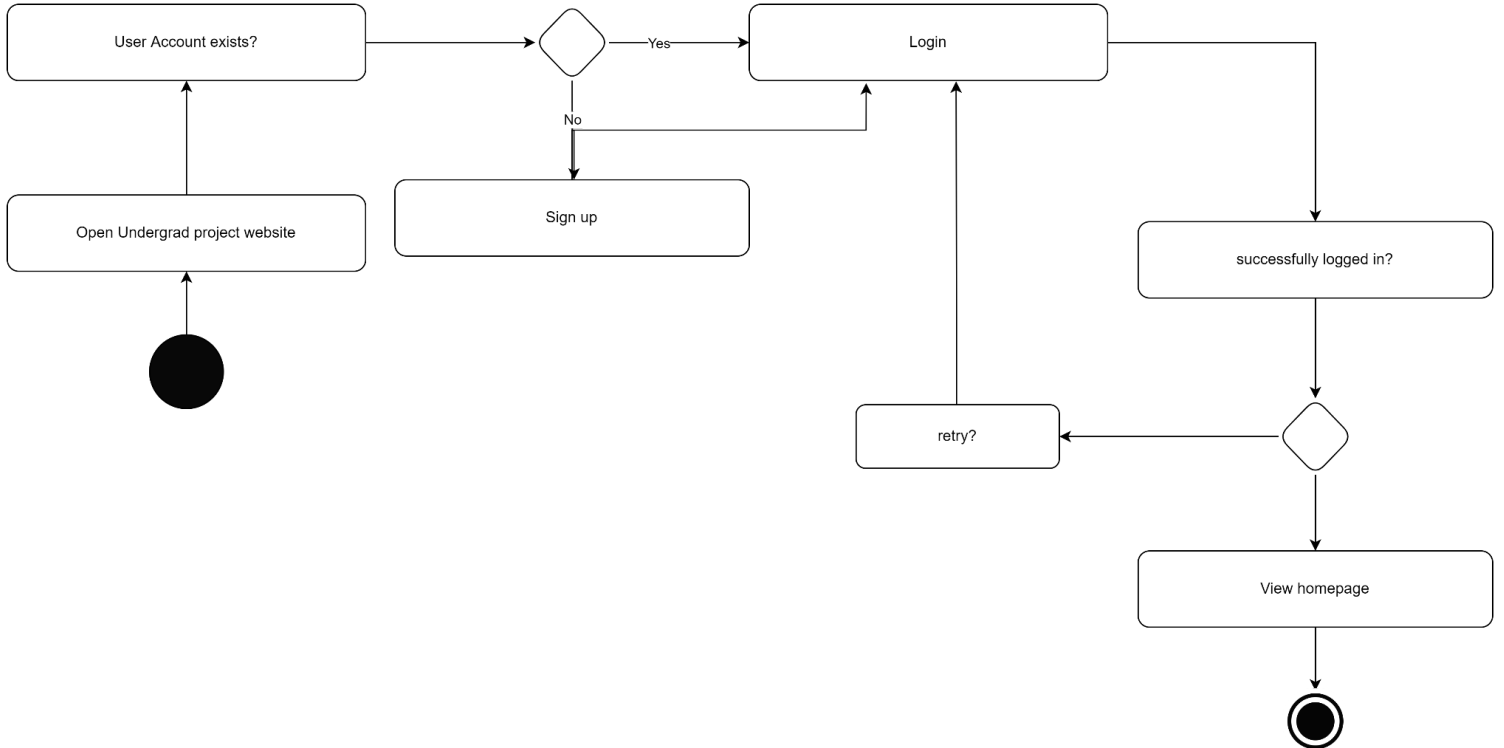


## Description:

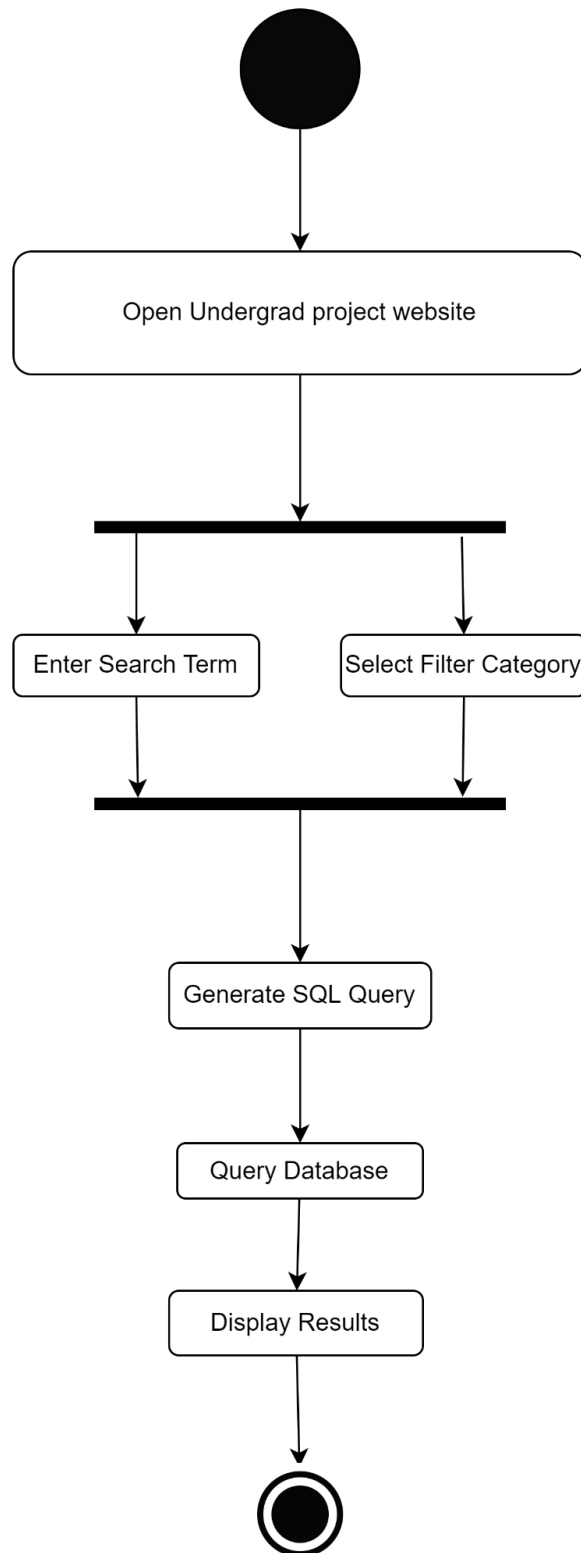
<b>Use Case Name:</b>	Creating, Deleting and Publishing Project
<b>Participating Actors:</b>	Logged in User, Administrator, System Server
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>• User has to be logged in.</li> <li>• User has selected the project menu.</li> </ul>
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. User opens the Project Page.</li> <li>2. User can select create or delete project.</li> <li>3. Upon selecting create project option, user can enter project details which include project name, project members, year, category, supervisor and repository link.</li> <li>4. After entering project details, user has to enter team code.</li> <li>5. Then user has to authenticate the action for creating project.</li> <li>6. Then the recently created project is to be verified by the admin after which it is viewable publicly.</li> <li>7. Upon selecting deletion of project, first we have to authenticate the delete action and project is deleted from database.</li> </ol>

# Activity Diagrams

## Login and Sign up

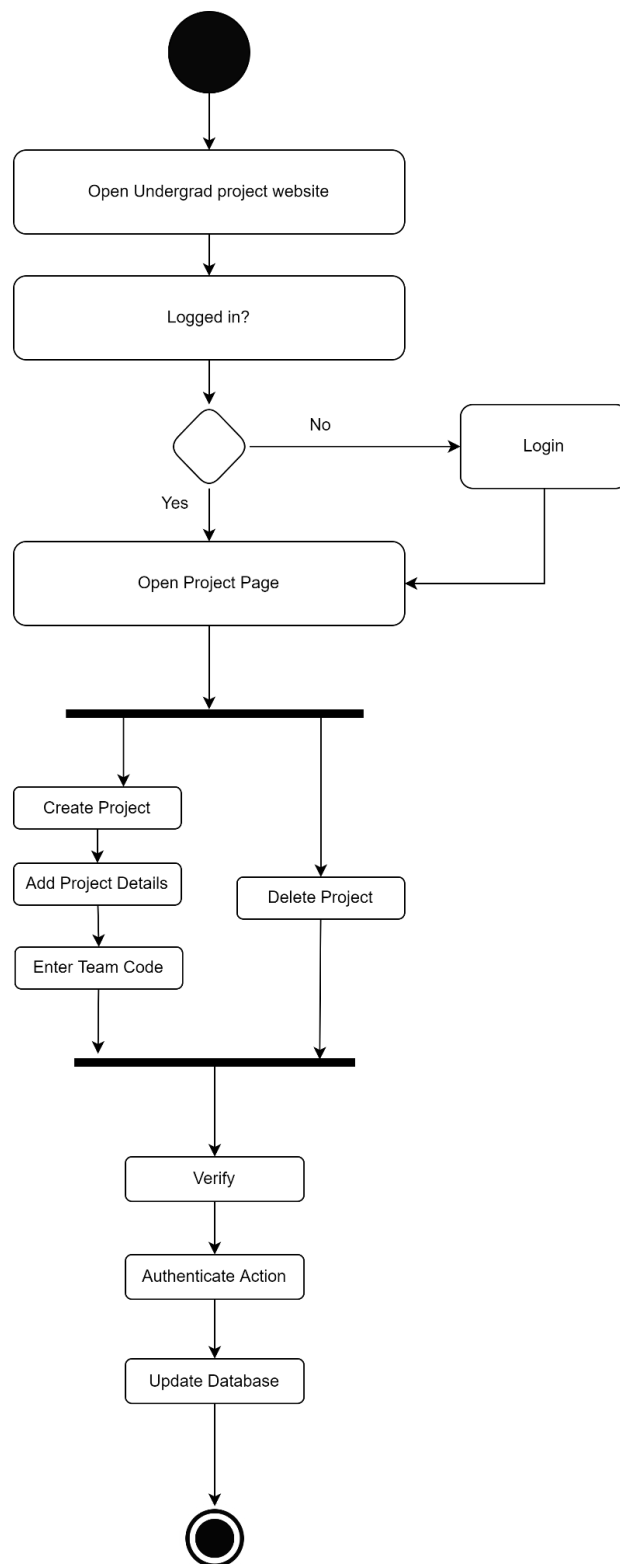


# Search and filter

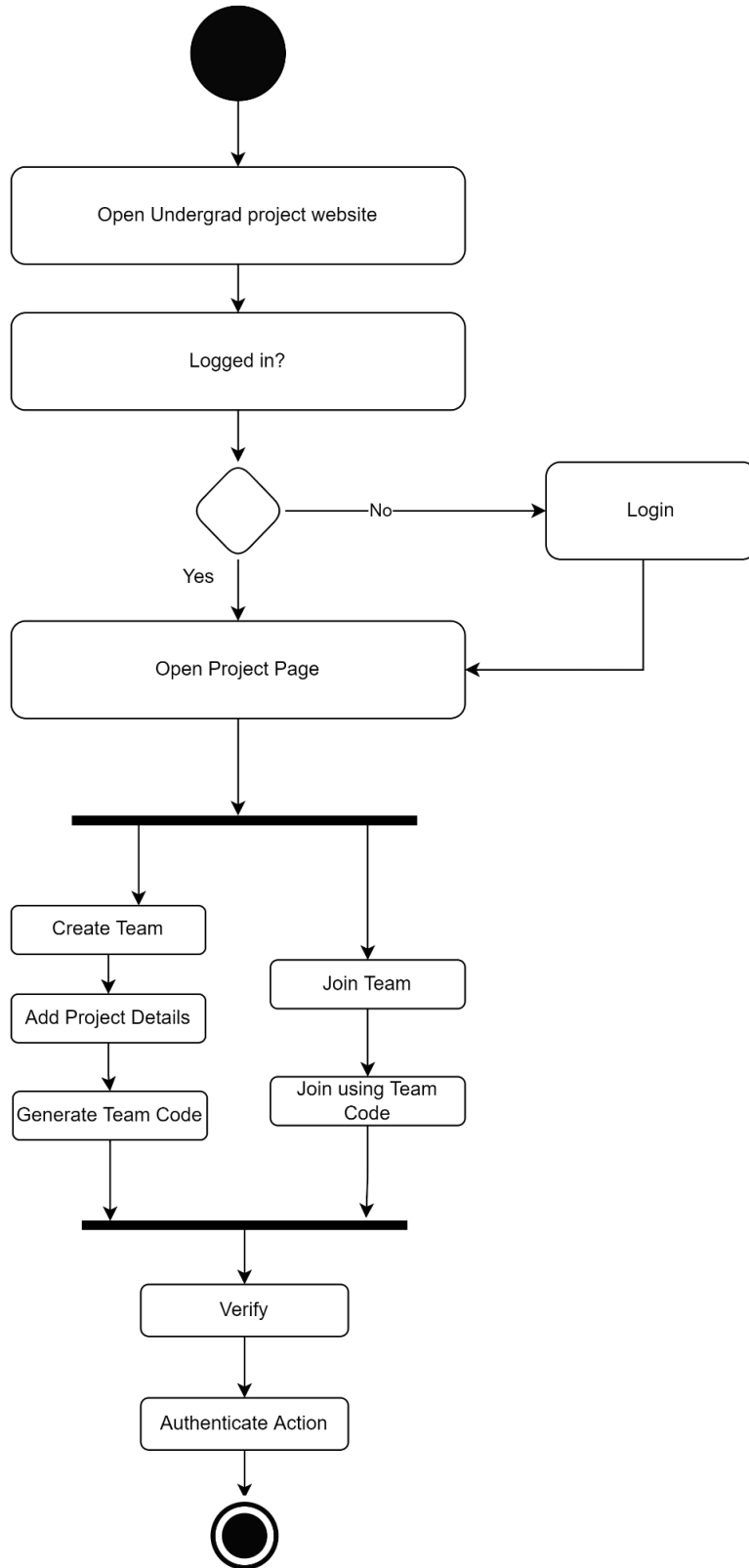




# Create, Delete and Publish Project



# Create/Join Team



# Sequence Diagrams

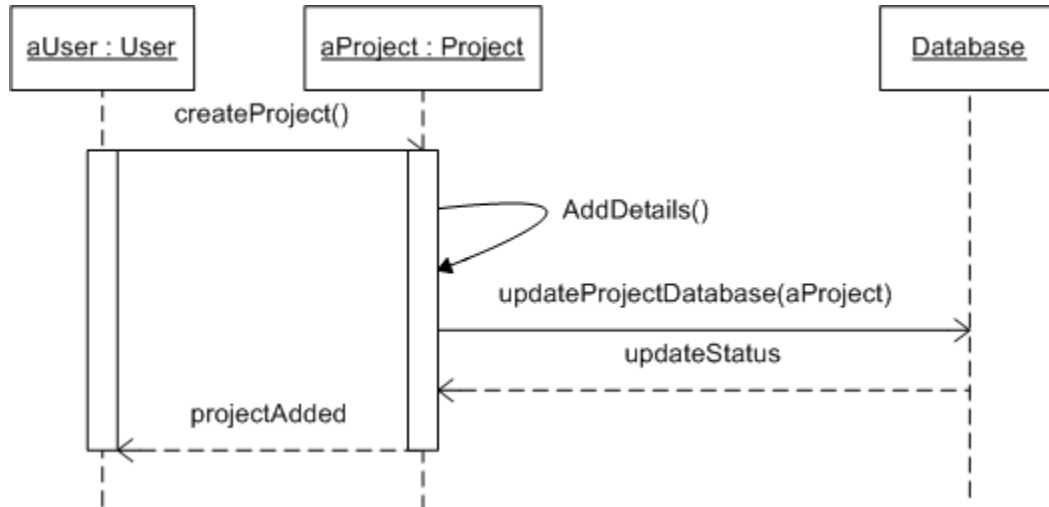


Fig: Sequential Diagram for creating project

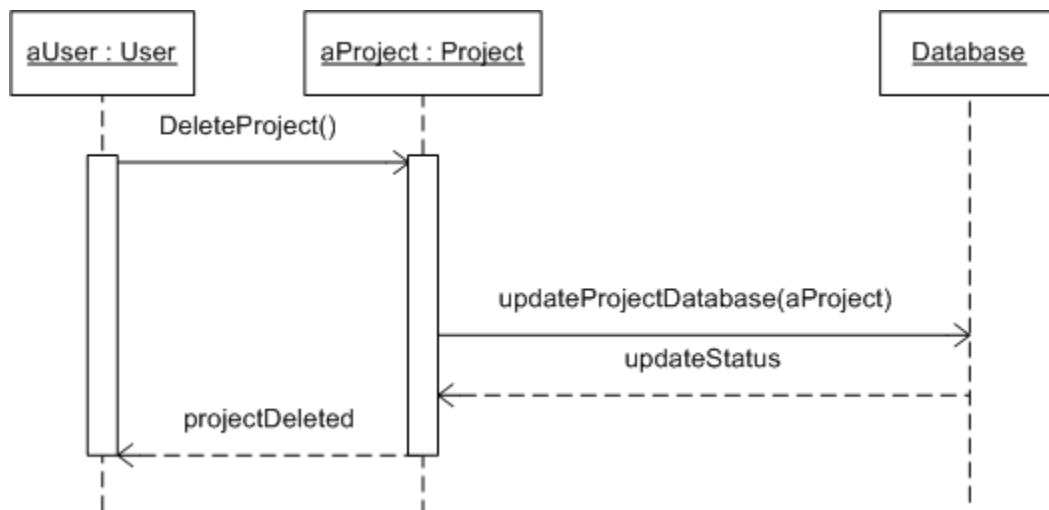


Fig: Sequential Diagram for deleting project

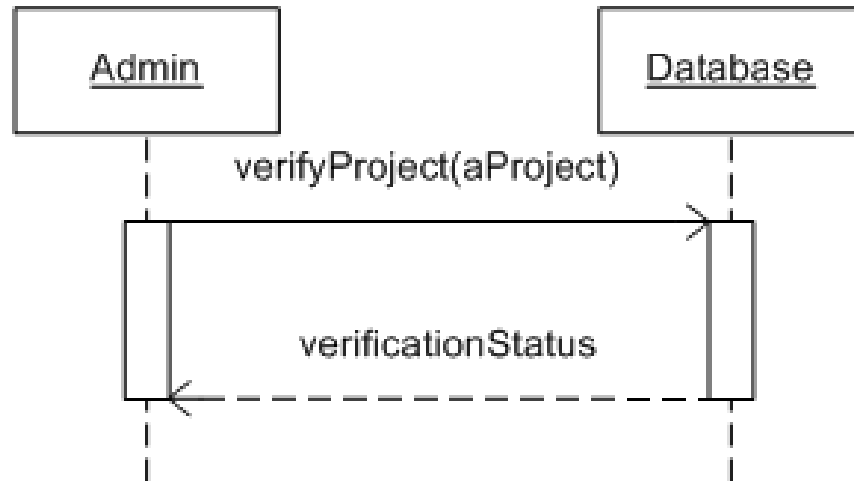


Fig: Sequential Diagram for verifying project

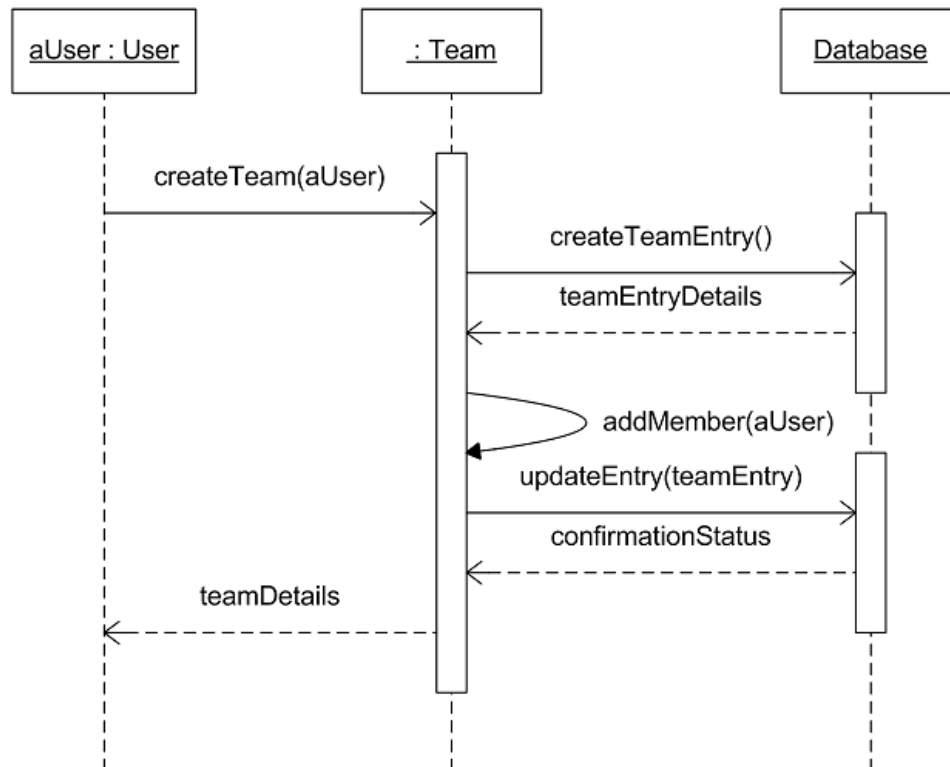


Fig: Sequential Diagram for creating team

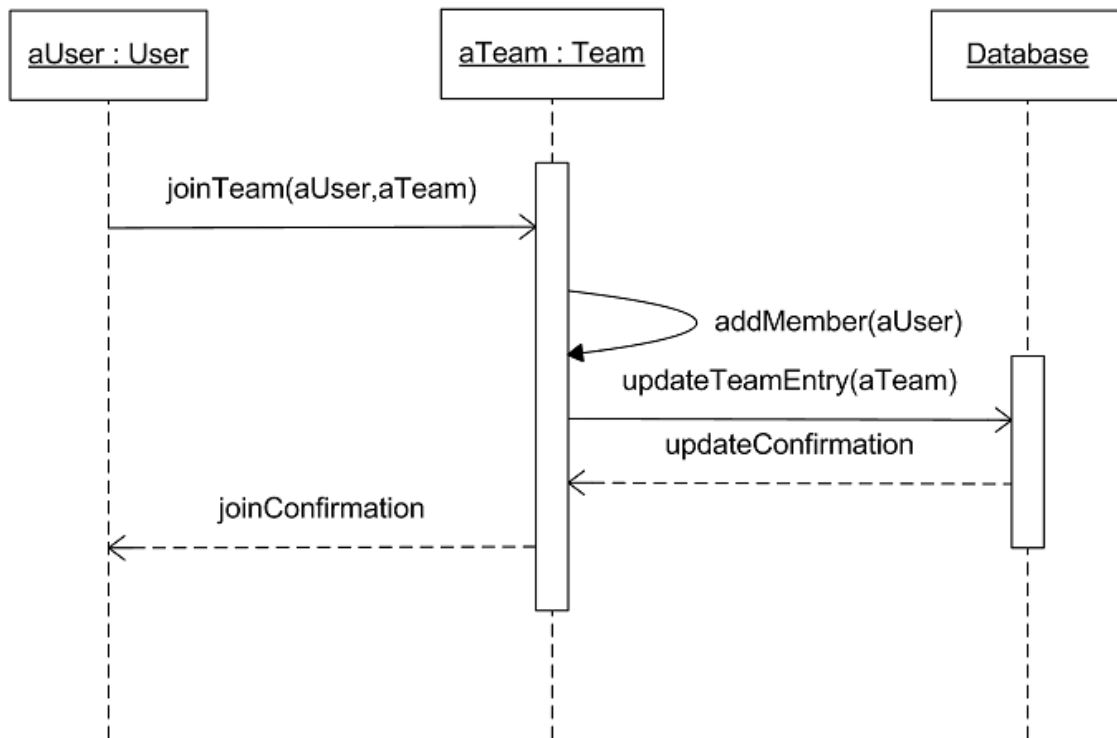


Fig: Sequential Diagram for joining team

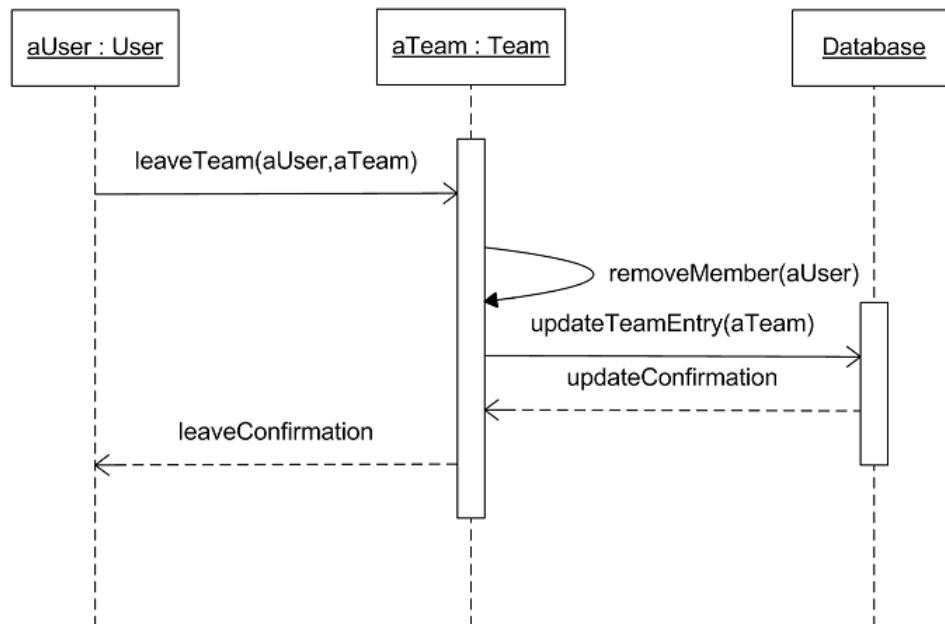


Fig: Sequential Diagram for leaving team

# Class Diagram

