# Document of the Use Case Diagram of the project “CodeNestIIT”

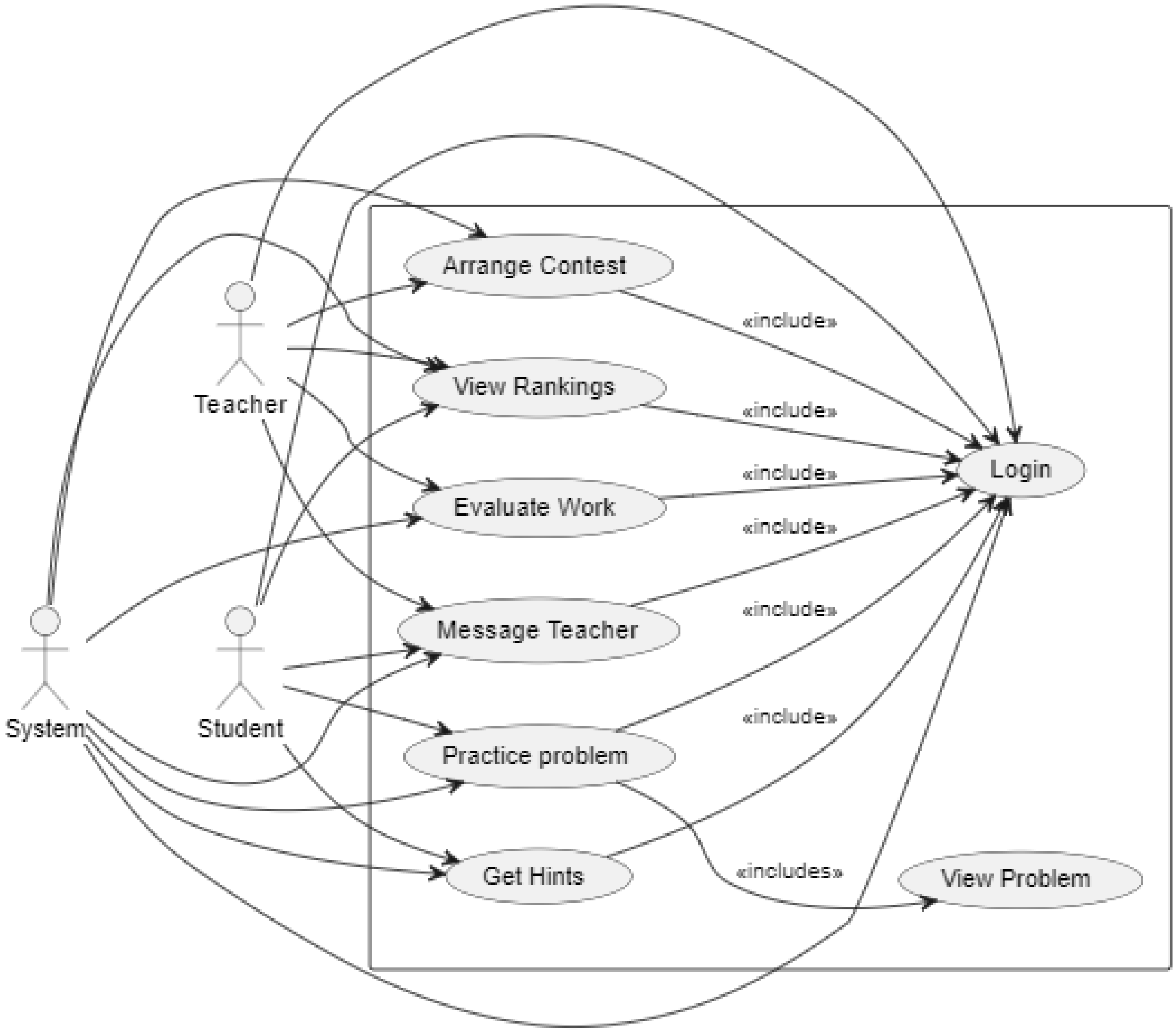

Fig: Use Case Diagram

| | |
|---|---|
| **Use Case No.** | 01 |
| **Use Case** | Login |
| **Description** | Allows users (students, teachers, administrators) to access the platform using their assigned user IDs and passwords. User IDs are provided by the system, eliminating the need for individual registration. |
| **Primary Actors**<br>**Secondary Actor** | Student, Teacher<br>System |
| **Preconditions** | • Users must have a valid user ID provided by the system.<br>• Users must remember their assigned passwords. |
| **Flows** | 1. User navigates to the platform's login page.<br>2. Enters the provided user ID in the designated field.<br>3. Inputs the associated password.<br>4. Submits the login form for authentication. |
| **Alternative flows** | If the provided user ID or password is incorrect, the system prompts the user to re-enter the correct credentials. |

| | |
|---|---|
| **Use Case No.** | 02 |
| **Use Case** | View Problem |
| **Description** | Enables users, specifically students, to explore and access a curated collection of C programming problems from authoritative sources such as "Teach Yourself C" and "C-The Complete Reference." This feature provides a comprehensive set of exercises to enhance learning. |
| **Primary Actors** | Student |
| **Secondary Actors** | Teacher |
| **Preconditions** | • Student must be logged in with a valid user ID.<br>• A list of available problems must be accessible to the student. |
| **Flows** | 1. The user, whether a student or a teacher, logs into the platform.<br>2. Navigates to the "View Problems" section of the platform.<br>3. Explores a categorized list of C programming problems.<br>4. Selects a specific problem of interest to view its statement and details. |

| | |
|---|---|
| **Use Case No.** | 03 |
| **Use Case** | Practice Problem |
| **Description** | Enables students to actively engage in practicing C programming by selecting and solving problems from the platform's extensive collection. This feature fosters a hands-on approach to learning. |
| **Primary Actors** | Student |
| **Preconditions** | • Student must be logged in with a valid user ID.<br>• A list of available problems must be accessible to the student. |
| **Flows** | 1. Student navigates to the "Practice" section of the platform.<br>2. Browses through the available C programming problems.<br>3. Selects a specific problem of interest.<br>4. Attempts to solve the chosen problem using the programming interface.<br>5. Submits the solution for evaluation. |
| **Alternative flows** | If the solution is incorrect, the system provides feedback, including hints and suggestions for improvement. |

| | |
|---|---|
| **Use Case No.** | 04 |
| **Use Case** | Get hints |
| **Description** | Allows students to request guidance while solving C programming problems. User can take up to three hints. The gap between first and second hints will be 30 minutes. After that user will be allowed to see the third one after one hour. |
| **Primary Actors** | Student |
| **Preconditions** | • Student must be logged in with a valid user ID.<br>• Student must submit once before requesting hints |
| **Flows** | 1. Student navigates to the problem-solving interface.<br>2. Attempts to solve a C programming problem.<br>3. Requests hints for additional guidance. |

| | |
|---|---|
| **Use Case No.** | 05 |
| **Use Case** | View Rankings |
| **Description** | Allow users, both students, and teachers, to access and explore rankings based on performance within the platform. |
| **Primary Actors** | Student, Teacher |
| **Preconditions** | • Users must be logged into the platform with valid user IDs.<br>• Ranking data should be available for the respective user and class. |
| **Flows** | 1. The user, whether a student or teacher, logs into the platform.<br>2. Navigates to the "View Rankings" section.<br>3. Explores individual and class rankings based on performance metrics. |

| | |
|---|---|
| **Use Case No.** | 06 |
| **Use Case** | Message Teacher |
| **Description** | Enables students to establish direct communication with teachers within the platform to discuss about a problem. |
| **Primary Actors** | Student, Teacher |
| **Preconditions** | • The student must be logged into the platform with a valid user ID.<br>• Teachers must have accounts and be accessible for communication. |
| **Flows** | 1. The student navigates to the "Message Teacher" section.<br>2. Selects the teacher they wish to communicate with.<br>3. Composes and sends a message, seeking clarification or guidance. |
| **Alternate flows** | If a teacher is not available for messaging, the system provides a notification and suggests alternative communication methods or times. |
| **Use Case No.** | 07 |
| **Use Case** | Arrange Contest |
| **Description** | Empowers teachers to organize programming contests within the platform. Teachers, logged in with valid credentials, can define contest parameters, select problems, and invite students to participate. |
| **Primary Actors** | Teacher |
| **Preconditions** | • The teacher must be logged into the platform with a valid user ID.<br>• Problems for the contest must be available in the platform's database. |
| **Flows** | 1. The teacher navigates to the "Arrange Contests" section.<br>2. Defines contest parameters, including duration and rules.<br>3. Selects C programming problems from the platform's collection for the contest.<br>4. Invites students to participate by specifying contest details. |
| **Alternate flows** | If no suitable problems are available, the system notifies the teacher and suggests alternatives or allows the addition of new problems. |

| | |
|---|---|
| **Use Case No.** | 08 |
| **Use Case** | Evaluate Work |
| **Description** | Enables teachers to assess students' solutions. |
| **Primary Actors** | Teacher |
| **Preconditions** | - The teacher must be logged into the platform with a valid user ID.<br>- Students must have submitted their solutions for evaluation. |
| **Flows** | 1. The teacher navigates to the "Evaluate Work" section.<br>2. Selects a specific assignment or problem for evaluation.<br>3. Reviews submitted solutions, assigns grades, and provides constructive feedback.<br>4. Updates the evaluation status, allowing students to view their assessed work. |
| **Alternative flows** | If a student's solution requires additional clarification, the teacher may ask further information before completing the evaluation. |