

Use Case Analysis

Use Case: User Registers to the Website

Actor: User

Description: This use case outlines the steps a user takes to register for an account on the CodeTrekker platform.

Preconditions:

- User accesses the CodeTrekker website.
- User does not have an existing account on the platform.

Postconditions:

- User successfully completes the registration process and gains access to CodeTrekker's features.

Main Success Scenario:

User Accesses Registration Page:

- User navigates to the registration page or clicks on the "Register" button on the CodeTrekker homepage.

User Enters Registration Details:

- User fills in the required fields in the registration form, including handle, email address, and password.

User Submits Registration Form:

- User reviews the entered details for accuracy and completeness.
- User clicks on the "Register" or "Sign Up" button to submit the registration form.

System Validates Registration Details:

- CodeTrekker system validates the submitted registration details to ensure they meet the required format and criteria.
- The system checks for unique handle/email, strong password requirements, and other validation rules.

System Creates User Account:

- Upon successful validation, CodeTrekker creates a new user account for the registered user.

User Receives Registration Confirmation:

- CodeTrekker displays a confirmation message to the user, indicating that the registration process was successful.

Alternative Flows:

A1. Invalid Registration Details:

- If the user submits incomplete or invalid registration details, CodeTrekker prompts the user to correct the errors.
- User revises the registration details and resubmits the registration form.

A2. handle/Email Already Exists:

- If the entered handle or email address already exists in the CodeTrekker database, the system displays an error message indicating that the username/email is already taken.
- User must choose a different handle/email to proceed with registration.

Use Case: User Logs In

Actor: User

Description: This use case outlines the steps a user takes to log in to their account on the CodeTrekker platform.

Preconditions:

- User has registered for an account on the CodeTrekker platform.
- User has valid login credentials (handle/email and password).

Postconditions:

- User successfully logs in to their account and gains access to CodeTrekker's features.

Main Success Scenario:

User Accesses Login Page:

- User navigates to the CodeTrekker program or opens the CodeTrekker mobile application.
- User choose the "Login" or "Sign In" operation to access the login page.

User Enters Login Credentials:

- User enters their handle/email and password into the corresponding fields on the login form.

User Submits Login Form:

- User reviews the entered login credentials for accuracy.
- User clicks on the "Login" or "Sign In" button to submit the login form.

System Validates Login Credentials:

- CodeTrekker system validates the entered login credentials to ensure they match the records in the platform's database.
- The system checks for correct handle/email and corresponding password.

System Authenticates User:

- Upon successful validation, CodeTrekker authenticates the user's identity and verifies their login credentials.
- If the credentials are valid, the system grants access to the user's account and associated features.

User Accesses Dashboard:

- CodeTrekker redirects the user to their account dashboard or home page after successful login.
- User can now access their profile, participate in contests, solve problems, and engage in community activities.

Alternative Flows:

A1. Invalid Login Credentials:

- If the user enters incorrect login credentials, CodeTrekker displays an error message indicating that the login attempt failed.
- User may retry entering the correct credentials or choose the "Forgot Password" option to reset their password.

Use Case: User Submits a Problem on CodeTrekker

Actor: User

Description: This use case outlines the steps a user takes to submit a problem to the CodeTrekker platform.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User has a problem ready for submission.

Postconditions:

- The problem is successfully submitted to CodeTrekker for review and inclusion in the platform.

Main Success Scenario:

User Navigates to Problem Submission Page:

- User accesses the "Submit a Problem" section on the CodeTrekker platform.

User Enters Problem Details:

- User fills in the required fields, including problem title, description, input/output specifications, constraints, and sample test cases.
- Optionally, user may include problem difficulty, tags, and time/memory limits.

User Submits the Problem:

- User reviews the entered details for accuracy and completeness.
- User clicks on the "Submit" button to submit the problem to CodeTrekker.

System Processes Problem Submission:

- Upon successful validation, CodeTrekker processes the submitted problem.
- It assigns a unique identifier (problem ID) to the problem and stores it in the platform's database.

User Receives Submission Confirmation:

- CodeTrekker displays a confirmation message to the user, indicating that the problem has been successfully submitted.
- Optionally, the user may receive an email notification confirming the submission.

Alternative Flows:

A1. Invalid Problem Details:

- If the user submits incomplete or invalid problem details, CodeTrekker prompts the user to correct the errors.
 - User revises the problem details and resubmits the problem.
-

Use Case: User Shows Problem Set and Sorts It

Actor: User

Description: This use case outlines the steps a user takes to view a problem set on the CodeTrekker platform and sort it based on difficulty level, either from hard to easy or from easy to hard.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User wants to access and sort a problem set based on difficulty level.

Postconditions:

- User successfully views the problem set and sorts it based on the selected criteria.

Main Success Scenario:

User Navigates to Problem Set Page:

- User accesses the "Problem Sets" section on the CodeTrekker platform.

User Accesses Sorting Options:

- On the problem set page, user looks for sorting options available.
- User identifies options to sort the problem set based on difficulty level, such as "Sort by Difficulty".

User Chooses Sorting Preference:

- User selects the desired sorting preference, either "Hard to Easy" or "Easy to Hard."
- User clicks on the corresponding sorting option to apply the chosen preference.

System Sorts Problem Set:

- CodeTrekker system sorts the problem set based on the user's chosen preference.
- If "Hard to Easy" is chosen, the system arranges the problems from the most difficult to the easiest.
- If "Easy to Hard" is chosen, the system arranges the problems from the easiest to the most difficult.

System Displays Sorted Problem Set:

- CodeTrekker presents the sorted problem set to the user on the problem set page.
 - User can view the arranged list of problems, starting from the selected difficulty level (hard or easy) based on the chosen sorting preference.
-

Use Case: User Searches for a Handle

Actor: User

Description: This use case outlines the steps a user takes to search for a specific handle (handle) on the CodeTrekker platform.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User wants to find information about a specific handle.

Postconditions:

- User successfully retrieves information related to the searched handle, if it exists.

Main Success Scenario:

User Accesses Search Functionality:

- User navigates to the search bar or the "Search" section of the CodeTrekker platform.

User Enters Handle:

- User types the handle (handle) they want to search for into the search bar.

User Initiates Search:

- User enters to initiate the search operation.

System Retrieves Handle Information:

- CodeTrekker system retrieves information related to the searched handle from the platform's database.
- This may include user profile details, submissions history, solved problems, contests participated in, etc.

System Displays Search Results:

- CodeTrekker presents the search results to the user in a structured format.
- User can view information related to the searched handle, including profile details and relevant statistics.

User Interacts with Search Results:

- User can click on the search result corresponding to the handle they were searching for to view detailed information.
- User may navigate to the profile page of the searched handle to view additional details and activities.

Alternative Flows:

A1. Handle Not Found:

- If the searched handle does not exist on CodeTrekker, the system displays a message indicating that the handle could not be found.
- User may refine their search criteria and try again or explore other handles.

Use Case: User Creates a Team

Actor: User

Description: This use case describes the steps a user takes to create a team on the CodeTrekker platform.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User wants to create a team for collaborative problem-solving or contest participation.

Postconditions:

- User successfully creates a new team on CodeTrekker, which can be joined by other users.

Main Success Scenario:

User Navigates to Team Creation Page:

- User accesses the "Create Team" section or choose the "Create Team" option the CodeTrekker platform.

User Enters Team Details:

- User fills in the required fields in the team creation form, including team name, description, and team type (e.g., competitive programming team, study group).

User Invites Team Members:

- User adds other CodeTrekker users to the team by entering their handles .

User Submits Team Creation Form:

- User reviews the entered details for accuracy and completeness.
- User choose the "Create Team"option to submit the team creation form.

System Validates Team Creation Details:

- CodeTrekker system validates the submitted team creation details to ensure they meet the required format and criteria.

System Creates Team:

- Upon successful validation, CodeTrekker creates a new team with the specified details.

User Receives Team Creation Confirmation:

- CodeTrekker displays a confirmation message to the user, indicating that the team creation process was successful.

Alternative Flows:

A1. Invalid Team Creation Details:

- If the user submits incomplete or invalid team creation details, CodeTrekker prompts the user to correct the errors.

Use Case: User Follows a Handle

Actor: User

Description: This use case outlines the steps a user takes to follow (friend) another user's handle on the CodeTrekker platform.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User wants to follow another user's handle.

Postconditions:

- User successfully follows the selected handle.

Main Success Scenario:

User Navigates to Profile of Handle:

- User accesses the profile page of the handle they want to follow.
- This can be done by searching for the handle .

User Selects Follow Option:

- On the profile page of the handle, user choose on the "Friend" option. privacy settings.

System Updates User's friend List:

- Upon successful validation, CodeTrekker adds the selected handle to the user's list of followed handles.

User Receives Confirmation:

- CodeTrekker displays a confirmation message to the user, indicating that the follow operation was successful.

Alternative Flows:

A1. Handle Already Followed:

- If the user is already following the selected handle, the "Follow" or "Friend" button may be disabled or replaced with an option to unfollow.
- User may choose to unfollow the handle or cancel the operation.

Use Case: User Favorites a Submission

Actor: User

Description: This use case outlines the steps a user takes to favorite (bookmark) a submission on the CodeTrekker platform for future reference.

Preconditions:

- User is logged in to the CodeTrekker platform.
- User wants to save a particular submission for later reference or review.

Postconditions:

- User successfully adds the submission to their list of favorites, making it easily accessible for future viewing.

Main Success Scenario:

User Navigates to Submission:

- User accesses the submission they want to favorite.

User Selects Favorite Option:

- On the submission page, user choose the "Favorite" option.

System Updates User's Favorites List:

- Upon successful validation, CodeTrekker adds the selected submission to the user's list of favorites.

User Receives Confirmation:

- CodeTrekker displays a confirmation message to the user, indicating that the favorite operation was successful.

Alternative Flows:

A1. Submission Already Favorited:

- If the user has already favorited the selected submission, the "Favorite" or "Bookmark" button may be disabled or replaced with an option to remove from favorites.
- User may choose to remove the submission from favorites or cancel the operation.

