**Use Cases Detailing interaction between User and System**

Use Case 1: User Registration and GitHub Integration

**Use Case 1: User Registration and GitHub Integration**

**Actors**: New User

**Preconditions:**

* The user has access to a web browser and an internet connection.
* The user has a GitHub account.

**Triggers:**

* The user wants to start using Zube.io to manage tasks in their GitHub repositories.

**Actions:**

1. The user navigates to the Zube.io website.
2. The user clicks on "Sign Up" or "Log In."
3. The user selects the option to sign up or log in using their GitHub account.
4. The user authorizes Zube.io to access their GitHub account.

**Expected Outcomes:**

* The user is registered or logged in to Zube.io.
* Zube.io is connected to the user's GitHub account, allowing access to repositories.

A screenshot of a computer screen

Description automatically generated

**Use Case 2: Creating a New Zube Project Linked to a GitHub Repository**

**Actors**: Registered User

**Preconditions:**

* The user is logged into Zube.io.
* The user has at least one repository in their GitHub account.

**Triggers:**

* The user wants to create a new project in Zube.io linked to a GitHub repository.

**Actions:**

1. The user clicks on "Create New Project" in Zube.io.
2. The user selects a GitHub repository to link to the new Zube project.
3. The user configures the project settings, such as project name and description.
4. The user saves the new project.

**Expected Outcomes:**

* A new project is created in Zube.io.
* The selected GitHub repository is linked to the new Zube project.
* Issues from the GitHub repository are imported into the Zube project.

A diagram of a person with different colored circles

Description automatically generated

**Use Case 3: Adding and Managing Tasks on the Kanban Board**

**Actors**: Project Member

**Preconditions:**

* The user is a member of a Zube project linked to a GitHub repository.

**Triggers:**

* The user wants to add or manage tasks on the Kanban board.

**Actions:**

1. The user navigates to the Kanban board of the Zube project.
2. The user creates a new issue/task on the board.
   * The user enters the issue title, description, assigns team members, sets due dates, and adds labels.
3. The user moves existing issues to different columns to reflect their current status.
4. The user updates issue details as needed.

**Expected Outcomes:**

* New issues are created and displayed on the Kanban board.
* Existing issues are moved to appropriate columns (e.g., "In Progress," "Review," "Done").
* Issue details are updated and synchronized with the GitHub repository.

A diagram of a project

Description automatically generated

### **Use Case 4: Synchronizing Issues Between Zube.io and GitHub**

**Actors**: Project Member

**Preconditions:**

* The user is a member of a Zube project linked to a GitHub repository.
* There are issues in the linked GitHub repository.

**Triggers:**

* The user wants to ensure issues are synchronized between Zube.io and GitHub.

**Actions:**

1. The user creates or updates an issue in GitHub.
2. The user checks the corresponding Zube project for synchronization.
3. The user updates an issue in Zube and verifies changes in GitHub.

**Expected Outcomes:**

* Issues created or updated in GitHub are reflected in Zube.io.
* Issues created or updated in Zube.io are reflected in GitHub.
* Changes are synchronized in real-time or near real-time.

A diagram of a person with circles and lines

Description automatically generated

**Use Case 5: Automating Workflow Transitions**

**Actors**: Project Admin

**Preconditions:**

* The user is an admin of a Zube project linked to a GitHub repository.

**Triggers:**

* The user wants to automate workflow transitions based on specific actions.

**Actions:**

1. The user sets up GitHub Actions or other automation tools.
   * The user defines rules such as moving an issue to "Review" when a pull request is opened.
2. The user commits code or opens a pull request in GitHub.
3. The user observes the issue transition in Zube.io.

**Expected Outcomes:**

* Workflow transitions are automated based on the defined rules.
* Issues are moved to appropriate columns in Zube.io based on GitHub actions.

A diagram of a person with circles and text

Description automatically generated

**Use Case 6: Notification Setup for Task Updates**

**Actors**: Project Member

**Preconditions:**

* The user is a member of a Zube project linked to a GitHub repository.

**Triggers:**

* The user wants to stay updated on task changes and progress.

**Actions:**

1. The user configures notification settings in Zube.io.
   * The user selects notification preferences (e.g., email, in-app notifications) for different events such as issue creation, status changes, comments.
2. The user performs actions that trigger notifications (e.g., updates an issue, moves an issue to a new column).
3. The user receives notifications according to their preferences.

**Expected Outcomes:**

* The user is notified of relevant changes and updates in tasks.
* Notifications help the user stay informed and up-to-date with project progress.

A diagram of a project

Description automatically generated