**NEW USE CASES🡪**

1. **ASSIGN EVENT :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to schedule an event for the members of their organizations.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects a date from the calendar and the organization name.
      2. Server fetches the details of selected organization and generates an event form.
      3. Admin fills the details of the form (like event title, event description etc.) and schedules an event.
      4. Server stores the details of the event in the storage file.
   5. **Alternate Flow :-**
      1. Invalid organization is selected. In such a case error message is displayed.
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
2. **ASSIGN TASKS :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to assign tasks to members of their organization.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects one of their organizations.
      2. Server fetches the details of the selected organization and generates a form for the admin.
      3. Admin selects the member to whom he/she wants to assign the task and provides the details of the task.
      4. Server stores the details of the task in the storage file.
   5. **Alternate Flow :-**
      1. Invalid organization is selected. In such a case error message is displayed.
   6. **Pre – Conditions :-** User should have admin privileges and the user should be logged into their account.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
3. **BECOME ADMIN:**
   1. **Actors :**- User
   2. **Introduction :**- This use case documents the procedure that allows a user to become an admin.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. User selects the admin option.
      2. Server checks the admin status of the user.
      3. If the user is not an admin then the status of the user is changed to admin in the storage file.
   5. **Alternate Flow :-** none
   6. **Pre – Conditions :-** User should be logged in to their account.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
4. **CREATE CLASS :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to create their own organization.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Server generates a code for the organization and asks the admin to enter name, description and member limit for the organization.
      2. After admin enters the details , server stores the details of the organization in storage file backend and displays the success message.
   5. **Alternate Flow :-**
      1. Invalid class limit is entered. In such a case error message is displayed.
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
5. **DO TASK :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to complete assigned tasks.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects a completed sub-task.
      2. Server increments the admin’s monthly stats and deletes the selected sub-task from the set of sub-tasks in the storage file.
   5. **Alternate Flow :-** none
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
6. **EDIT CLASS :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to edit an existing organization’s details.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects an organization to edit.
      2. Server fetches the details of the selected organization, generates a form and populates it with the current details of that organization.
      3. Admin makes changes to the details.
      4. Server then stores the changes in the storage file.
   5. **Alternate Flow :-**
      1. Invalid organization is selected. In such a case error message is displayed.
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
7. **JOIN CLASS :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to join other organizations.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin enters the code of the organization he/she wants to join.
      2. Server matches the code with list of codes of existing organizations .
      3. If the code matches with one of the codes in the list then the admin’s user number is stored with other members of that organizations.
   5. **Alternate Flow :-**
      1. Invalid code is entered by the admin. In such a case error message is displayed to the admin
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
8. **UNENROLL :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to unenroll from an organization.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects the organization to unenroll from.
      2. Server takes the organization number and searches for the user-number of the admin in the organization’s group.
      3. If the admin is found then the admin’s user-number is removed from the group.
   5. **Alternate Flow :-**
      1. Admin’s user number is not found in the organization’s group. In such a case error message is displayed to the user.
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-** JSON file is updated
   8. **Related use cases :**- none.
9. **VIEW PERFORMANCE :**
   1. **Actors :**- Organization Admin
   2. **Introduction :**- This use case documents the procedure that allows an admin to analyze the performance of the members of their organization.
   3. **Special Requirements :-** none.
   4. **Basic Flow :-**
      1. Admin selects the name of one of the members of the organization.
      2. Server fetches the monthly stats of the selected member.
      3. Server generates a graph based on the stats.
   5. **Alternate Flow :-** none
   6. **Pre – Conditions :-** User should have admin privileges.
   7. **Post –** **Conditions :-**  none
   8. **Related use cases :**- none.