



# CS 319 Term Project

*CluBil: Student Club Manager*

## Project Analysis Report

**İdil YILMAZ - 21703556**

**Doğukan Ertunga KURNAZ - 21702331**

**Turgut Alp EDIS - 21702587**

**Taha Batur ŞENLİ - 21901857**

**Argun MURADOV - 21901305**

**Cemhan Kaan ÖZALTAN - 21902695**

Instructor: Eray Tüzün

Teaching Assistant(s): Elgun Jabrayilzade, Emre Sülün, Erdem Tuna, Muhammad Umair Ahmed and Cevat Aykan Sevinç

Oct 31, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object-Oriented Software Engineering course CS319.

# Contents

<b>1 Introduction</b>	<b>4</b>
<b>2 Proposed System</b>	<b>4</b>
<b>2.1 Overview</b>	<b>4</b>
<b>2.1.1 Landing Page</b>	<b>4</b>
<b>2.1.2 Login Page</b>	<b>5</b>
<b>2.1.3 Event List Page</b>	<b>5</b>
<b>2.1.4 Calendar of Registered Events</b>	<b>5</b>
<b>2.1.5 Student Profile Page</b>	<b>6</b>
<b>2.1.6 View Clubs Page</b>	<b>6</b>
<b>2.1.7 Club Profile Page</b>	<b>6</b>
<b>2.1.8 Sysadmin Account</b>	<b>7</b>
<b>2.1.9 Club Advisor Account</b>	<b>7</b>
<b>2.2.10 Search Bar</b>	<b>8</b>
<b>2.2 Functional Requirements</b>	<b>8</b>
<b>2.2.1 Login</b>	<b>8</b>
<b>2.2.2 Events</b>	<b>8</b>
<b>2.2.3 Clubs</b>	<b>8</b>
<b>2.2.4 Students</b>	<b>9</b>
<b>2.2.5 Budget</b>	<b>9</b>
<b>2.2.6 Profile</b>	<b>9</b>
<b>2.2.7 Search</b>	<b>9</b>
<b>2.3 Non-functional Requirements</b>	<b>10</b>
<b>2.3.1 Reliability</b>	<b>10</b>
<b>2.3.2 Performance</b>	<b>10</b>
<b>2.3.3 User-Friendly Interface</b>	<b>10</b>
<b>2.3.4 Safety</b>	<b>10</b>
<b>2.3.5 Maintainability</b>	<b>11</b>
<b>2.4 Pseudo Requirements</b>	<b>11</b>
<b>3 System Models</b>	<b>12</b>
<b>3.1 Use-Case Diagram</b>	<b>12</b>
<b>3.2 Scenarios</b>	<b>12</b>
<b>3.3 Object and Class Diagram</b>	<b>28</b>
<b>3.4 Activity Diagrams</b>	<b>29</b>
<b>3.5 Sequence Diagrams</b>	<b>32</b>
<b>3.6 State Diagrams</b>	<b>33</b>
<b>3.7 Mockup Diagrams (UI Models)</b>	<b>34</b>
<b>4 Improvements and Summary</b>	<b>57</b>
<b>4.1 General</b>	<b>57</b>
<b>4.2 Overview of the System</b>	<b>57</b>
<b>4.3 Changes to Actors</b>	<b>57</b>
<b>4.4 Functional Requirements</b>	<b>57</b>

<b>4.5 Non-Functional Requirements</b>	<b>57</b>
<b>4.6 Use Case Diagram</b>	<b>57</b>
<b>4.7 Activity Diagrams</b>	<b>57</b>
<b>4.8 Sequence Diagrams</b>	<b>58</b>
<b>4.9 State Diagrams</b>	<b>58</b>
<b>4.10 Class Diagrams</b>	<b>58</b>
<b>Glossary</b>	<b>59</b>
<b>References</b>	<b>59</b>

## **1 Introduction**

The program we've developed is a web-based application designed to help Bilkent University students and student clubs. Our team created this project with hopes of easing the interaction between student clubs, students, and the Student Activities Center (SAC). As such, the program can be used by students, board members of clubs and administrators of SAC to hasten the process of organizing, approving, and announcing club events.

The application provides students neat and quick access to all the club events planned in the university as well as the chance to enroll in clubs or specific activities. On the other hand, student clubs, alongside managing their clubs' profiles, can quickly create events and submit application forms to SAC. Finally, SAC can use the system to easily coordinate events as well as student clubs.

## **2 Proposed System**

This is where you provide the details of the results of your analysis work.

### **2.1 Overview**

#### **2.1.1 Landing Page**

The first screen the program's users will see will be the landing page, which will display all the upcoming public events open to everyone. It will also include online events marked as public alongside links to the external meeting app, such as Zoom. It is an essential feature because it will help clubs with promoting activities open to other universities. Examples to this case are large summits or activities organized alongside fellow clubs from other universities. This page will also include a login button taking the user to the application's login feature. From the point of view of clubs, it will be beneficial to advertise their own clubs, because landing page will be a field that brings together the supply and demand of both university students and clubs.

### **2.1.2 Login Page**

The login page of the program will be accessible from the top bar of the landing page for the user. Here, two login options will be displayed side by side, one of which is for students and the other for clubs. The student login will have text fields for taking a student ID and a password and the club login will be done with an assigned username and password combination. In addition, a sysadmin login which will not be shown explicitly will also be available for the usage of the Student Activities Center and other administrative personnel. All account creations will be done through the sysadmin account, similar to the STARS system of Bilkent University, to prevent unnecessary account flooding.

### **2.1.3 Event List Page**

The Events List Page will be the first page seen by logged in students on the platform. Similar to the GE250/251 page in the STARS system, this page will display all upcoming university-restricted face-to-face and online activities in list form. The user will be able to view information such as the name, date, time, duration, location (platform name for online events), the organizing club and the total/available quota for the event.

The list will be in chronological order, starting from the closest event. Filters at the top of the page will let the student filter the events by keyword and filter online or face-to-face activities specifically.

The “show more” button within the list will show a pop-up window, with additional description of the event, relevant URLs (registration form, Zoom link etc.), and the “Register to Event” button through which a student can register to the event.

### **2.1.4 Calendar of Registered Events**

In the Calendar page, students will be able to view their registered events in a monthly calendar view. The events represented in the calendar will be clickable buttons which will again display the “show more” pop-up described in the previous section. Students can use this popup to cancel their registration to the event and delete the event from their calendar.

### **2.1.5 Student Profile Page**

In the profile page, the students will be able to view their personal information, profile picture, joined clubs and relevant tags (if they have a specific position within the club), and an “about me” section where they can add a biography. In this page, students can also see their inbox with messages and club-restricted event invites from their clubs.

### **2.1.6 View Clubs Page**

In a separate page, students can look throughout all the active clubs within the university in list form. They can also filter this list to clubs they have joined already, or simply by searching for a club. The “show more” button for each club will open a pop-up that will display the logo, purpose and upcoming activities for the specific club. The students will also be able to Join or Leave the club throughout this pop-up window.

### **2.1.7 Club Profile Page**

In the profile section of the club, there will be a club logo and/or a photo of the club managers and informative information about the club for students who want to participate in events. The ability to edit a profile and change a password just below the club name will become available to club administrators. When the club manager wants to edit the profile, the Edit Profile button is clicked, and then the profile name, student ID, Club information, and profile photo can be edited in the pop-up that opens. The club manager who wants to make a password change can type his/her new password after confirming his/her old password with the pop-up that appears when s/he clicks the button.

In a frame where the Upcoming events are visible, the club administrator can create a new event for the club( Create Event), as well as delete one or more of the created events( Delete Event ) .Each time a new event is created, the club to which the event belongs, the date it will be held and the time between which it will take place can be seen in the Upcoming Events frame, if the user wants to get more detailed information about any event, More section can be used.If the club administrator wants to add an event, clicking on the Add Event button will open a pop-up.The event name, date and time of the event, location and description information; quota for how many people the event will be, type of event (public, campus only or private), photo

of the event (such as banner) will be in this pop-up. If the club administrator wants to delete an event, clicking on the Delete Event button opens a pop-up and lists all the events that can be deleted here.

The club's finances can be checked from the "Finance" tab on the top bar. This page, also visible by the SAC admin, contains detailed information about the club's sources of revenue, expenditures and the cost of each event. It also has a pie chart that displays how the club's total budget is appropriated to the events. The club user can add or delete events in this panel. When adding an event, its initial budget and how much of that budget is actually used can be specified as well. On the other hand, the SAC admin has access to this panel. They can view how the club's budget is spent to decide whether to approve or delete the club's proposed events. Additionally, the SAC administrator can edit a club's total budget, increasing or decreasing it. However, this is done from the SAC admin panel and the SAC admin must provide a reason for the change.

### **2.1.8 Sysadmin Account**

The sysadmin account will mainly be used by the Student Activities Center employees for administrative purposes. When a new student is admitted to the university, their account will be created through this sysadmin account with their student ID as the username and an initial password which may later be changed. Similarly, when a new club is formed, their registration will be made through the sysadmin account. This account will also be able to delete the listed types of accounts in the case of a student leaving the university or a club being closed. Organized student club events will also be confirmed and created by administrative personnel through this account, along with the deletion of denied events.

### **2.1.9 Club Advisor Account**

Student Club Advisors play a key role in directing, helping and reviewing student clubs. Using their accounts, Club Advisors can view proposed events of their assigned club and use their vote of confidence for a specific event. They can also see their budget, incomes and spendings. Moreover, advisors can write reviews to the event requests made by their club.

### **2.2.10 Search Bar**

The search bar can find any club pages or activities that is available for the students. By using the search bar, students can use the search bar to participate in the clubs and activities of these clubs or to leave the clubs they participate in.

## **2.2 Functional Requirements**

### **2.2.1 Login**

- Students, clubs, SAC admins and club advisors can login with a unique ID and password initially provided by the system.
- The system will deny entry and return an error message if the user's login credentials are incorrect.

### **2.2.2 Events**

- Students can register or unregister an event.
- Each event has a name, quota, time slot, date, location, label and budget. However, budget is given when the clubs calculate their budgets.
- Student clubs can create an event as an event request and it is approved by SAC. Also, after the creation of the request, the club advisor can inspect that event and may give feedback to the SAC.
- Whether the event is approved or pending, the student club can delete that event if that event belongs to the student club.
- If the event is not approved by SAC, it will be deleted automatically.
- After the event is approved, the student club can inform their members with a private invitation.
- After the event, students can give feedback about their experiences.
- The system will refuse an event form filled by the club if the club enters a location that's already been reserved.
- The system will warn the club user if they try to schedule two different events at the same time. However, this warning can be overridden.
- The system will warn the student if the time of two different registered events collide. However, this warning can be overridden.
- The system will display a warning message if the club user tries to use more than the remaining budget while creating an activity in the finances tab.

### **2.2.3 Clubs**

- Each club is created as an account by the SAC.
- Students can view, join and leave the club.
- Club accounts can be deleted by the SAC.
- Each club can create and delete events and label them as public, campus-only and private.
- Each club has members which are student and club advisors.

#### **2.2.4 Students**

- Each student is created and can be deleted by SAC.
- Each student can personalize their calendar which consists of the registered events of the student.
- Each student can list events and clubs and filter them.

#### **2.2.5 Budget**

- Each club has a budget and the budget is determined by the SAC.
- Club advisors of the club can see the budget and expenses of the club.
- The club cannot overcome the budget but it can request the budget change to the SAC. However, the system will provide a warning to the SAC admin if they do not include a reason for the budget change.
- The events can be added with their costs to the expense list to calculate the remaining budget. However, The system will display a warning message if the club user tries to use more than the remaining budget while creating an event.

#### **2.2.6 Profile**

- All the users of the program, which are students, clubs, club advisors and SAC have a profile.
- All users can customize their profile picture, name, design and change their passwords. However, the system will decline a password change if the user enters their old password. Besides, the system will decline a username change if the user enters their old name.

#### **2.2.7 Search**

- Every user can search the specific event and specific club from the search bar.
- SAC, clubs and their club advisors can search the event request. However, SAC can search all event requests.
- Only SAC can search students.

## **2.3 Non-functional Requirements**

### **2.3.1 Reliability**

The system should function all the time, 24/7 unless it is taken down for maintenance by the sysadmin. The system will have a capacity of users that can use it at the same time, and this capacity is determined by the budget of the university and number of people in the university.

### **2.3.2 Performance**

Pop ups will open in less than 3 seconds. The page load should be under 5 seconds. The system should update the status of a user's participation in an event or membership of a club in 2 seconds after a change happens.

### **2.3.3 User-Friendly Interface**

User-Friendly interface is an important non-functional requirement for our software because our software aims to reach everyone in the university. The UI will be designed to be friendly to people from all backgrounds, regardless of their understanding of technology. In the mockups part of this report, we tried our best to reflect the combination of functionality and the ease-of-use. Our mockups were designed with the consideration of the technical possibilities.

### **2.3.4 Safety**

If a user wants to join an event that is marked as private by the student club, the student club receives a notification of the join request and can approve or deny it. Unless it is approved, the user can't see the result of their join request. All types of users will be provided with their initial credentials (username and password) by the sysadmin and the system will only accept the entry requests of these provided credentials. The users can then change their passwords in the system, which will be saved by the system. The password should be 8 characters in length and must contain one digit and one uppercase letter. If a password that isn't recognized by the system is entered, the login request will be denied and blocked by the system after three failed attempts. The system will automatically log the user out if there is no interaction for more than 6 minutes.

### **2.3.5 Maintainability**

Because our project will be developed with future extensions in mind, OOP technique will be used while writing the code. Whole codebase will be managed with the version control tools like Git, which enables us to revert the code when needed. Because deployment will be automatic when the new commit is made, the project will be easier to maintain.

## **2.4 Pseudo Requirements**

- The system must be programmed using Object-Oriented-Programming (OOP).
- The application must be web-based.
- The system must use GitHub repositories to store the code and track progress.
- The system must be designed using Universal Markup Language (UML).
- The system must satisfy the needs of Bilkent University SAC Administrator.

### 3 System Models

#### 3.1 Use-Case Diagram

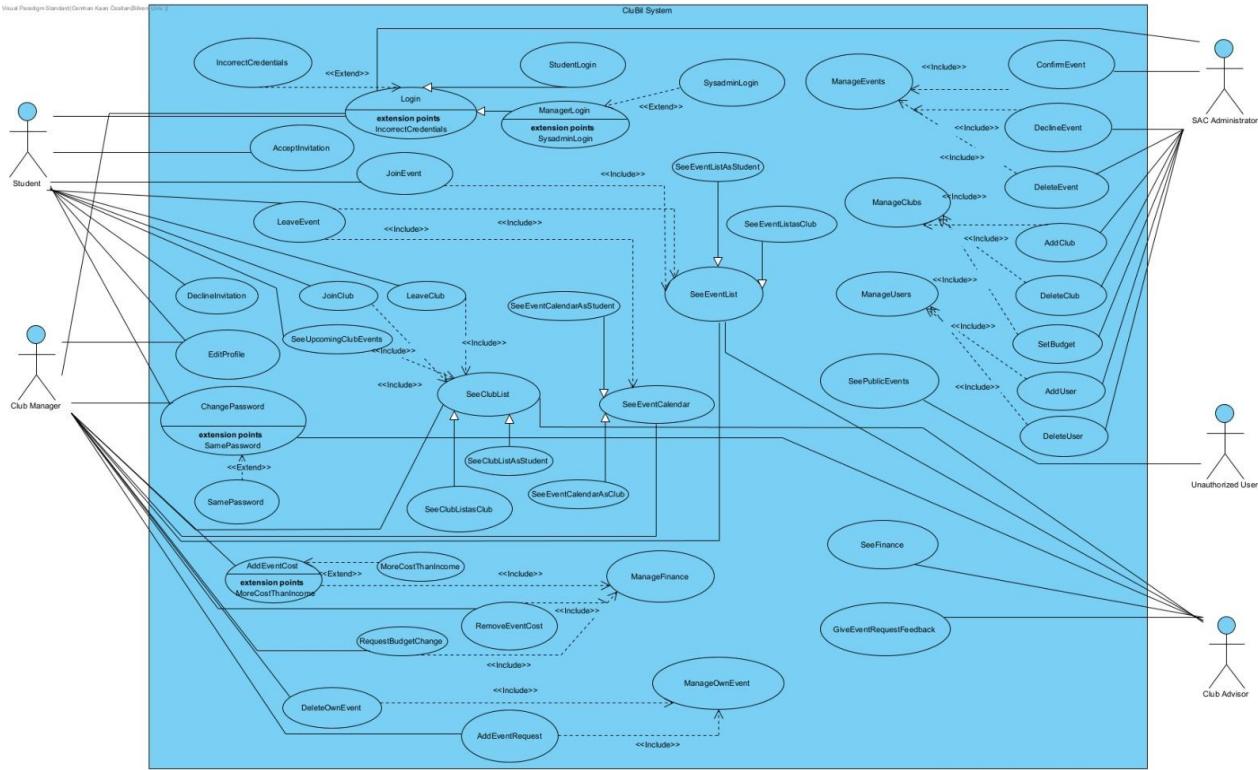


Figure 1: Use Case diagram

#### 3.2 Scenarios

1.

- **Use Case :** EditProfile
- **Participating Actor:** Initiated by Student OR  
Initiated by Club Manager
- **Entry Condition:**
  - The user logs into the system.
  - The user clicks the name at the top bar.
- **Exit Condition:**
  - The user clicks on "Save Changes" button.
  - The user exits from the window without saving the changes.
- **Flow of events:**
  - The user clicks the "Edit Profile" button.
    - The program responds by opening a popup including the information about the user.
  - The user sets the new information in the screen.
  - The user clicks "Save Changes" button.

- The program responds by updating the user's information in the profile page and database.
  - **Quality Requirements:**
    - The user must not remain inactive for longer than 6 minutes.
- 2.
- **Use Case :** RequestBudgetChange
  - **Participating Actor:** Initiated by Student Club
  - **Entry Condition:**
    - Student Club opens the finance tab of the club.
  - **Exit Condition:**
    - Student Club submits a budget change request form to the Student Activities Center (SAC).
  - **Flow of events:**
    - Student Club clicks on the “Request Budget Change” button on the finance page.
      - The program responds by opening a popup of a form which includes the requested new budget and the reason for the requested change. The old budget is also shown in the popup.
    - The student club fills the form and submits it by clicking on the “Submit” button on the bottom.
      - The program responds by sending the request to the SAC Administrator as a notification.
  - **Quality Requirements:**
    - The user must not remain inactive for longer than 6 minutes.
    - At the start of the flow of events, this use case includes the ManageFinance use case. The ManageFinance use case initiates when the user opens their club's finances.
- 3.
- **Use Case :** JoinEvent
  - **Participating Actor:** Initiated by Student
  - **Entry Condition:**
    - Students log into the system.
  - **Exit Condition:**
    - Student joins the event
  - **Flow of events:**
    - Student selects an event from the events list, which is the first page seen after login.
      - The program responds by opening a popup including detailed information about the event.
    - Students click on the “Join” button on the top right corner.
      - The program responds by updating the user's status in their calendar and also changing the clicked “Join” icon to the “Leave” icon.
  - **Quality Requirements:**

- The user must not remain inactive for longer than 6 minutes.
- At the start of the flow of events, this use case includes the SeeEventList use case. The SeeEventList use case initiates when the user opens the list of all events.

#### 4.

- **Use Case :** LeaveEvent
- **Participating Actor:** Initiated by Student
- **Entry Condition:**
  - Student logs into the system OR,
  - Student opens their event calendar.
- **Exit Condition:**
  - Student leaves the event.
- **Flow of events:**
  - Student selects an event.
    - The program responds by opening a popup including detailed information about the event.
  - Student clicks on the “Leave” button on the top right corner.
    - The program responds by updating the user’s status in their calendar and also changing the clicked “Leave” icon to the “Join” icon.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.
  - At any given point during the flow of events, this use case can include the SeeEventList OR SeeEventCalendar use cases. The SeeEventList use case is initiated when the user opens the list of all events. The SeeEventCalendar is initiated when the user clicks on the Calendar tab.

#### 5.

- **Use Case:** JoinClub
- **Participating Actor:** Student
- **Entry Condition:** Student is logged in
  - Student has not joined the selected club before
- **Exit Condition:** Student clicks the join club button
- **Flow of Events:**
  - Student clicks on the “Clubs” button
    - The program displays the club list to the student
  - Student selects the club they would like to join and clicks “Show More” button
    - The program shows the pop-up that shows the detail information about the selected club
  - Student clicks the “join” button in the pop-up
    - The program adds the student to the club and the student joined the club
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

- It includes the seeClubList that shows all clubs to the student.

**6.**

- **Use Case:** LeaveClub
- **Participating Actor:** Student
- **Entry Condition:** Student is logged in  
Student is a member of the club
- **Exit Condition:** Student clicks the leave club button
- **Flow of Events:**
  - Student clicks on the “Clubs” button
    - The program displays the club list to the student
  - Student selects the club they would like to leave and clicks “Show More” button
    - The program shows the pop-up that shows the detail information about the selected club
  - Student clicks the “leave” button in the pop-up
    - The program removes the student to the club and the student left the club
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.
  - It includes the seeClubList that shows all clubs to the student.

**7.**

- **Use Case:** SeeUpcomingClubEvents
- **Participating Actor:** Initiated by Student
- **Entry Condition:**
  - Student opens the list of all clubs tab.
- **Exit Condition:**
  - Student sees the events specific to a single club.
- **Flow of events:**
  - Student selects a club from the list of all clubs tab.
    - The program responds by opening a popup including detailed information about the club and all of the club’s upcoming events.
  - From here the student can click on any upcoming event and view more information about them.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.
  - At the start of the flow of events, this use case includes the SeeClubList use case. The SeeClubList use case initiates when the user opens the list of all clubs registered to the program.

**8.**

- **Use Case:** AcceptInvitation
- **Participating Actor:** Initiated by Student
- **Entry Condition:** Student is logged in

- Student has received a private invitation from a club
    - Student is clicked on their name at the top bar
  - **Exit Condition:** Student accepts the invitation
  - **Flow of Events:**
    - Student clicks on the “My Profile” button
      - The program displays the profile page of the student
    - Student Selects the invitation they would like to respond to from the “Notifications” section
    - Student clicks one of the “Accept” buttons from the view.
      - The program notifies the club account according to the respond from the user.
  - **Quality Requirements:**
    - The user must not remain inactive for longer than 6 minutes.
    - The invited event must not be full.

## 9.

- **Use Case:** DeclineInvitation
- **Participating Actor:** Initiated by Student
- **Entry Condition:** Student is logged in
  - Student has received a private invitation from a club
  - Student is clicked on their name at the top bar
- **Exit Condition:** Student declines the invitation
- **Flow of Events:**
  - Student clicks on the “My Profile” button
    - The program displays the profile page of the student
  - Student Selects the invitation they would like to respond to from the “Notifications” section
  - Student clicks one of the “Decline” buttons from the view.
    - The program notifies the club account according to the respond from the user.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

## 10.

- **Use Case:** SeeEventCalendarAsClub
- **Participating Actor:** Club Manager
- **Entry Condition:** Club manager clicks the Calendar button
- **Exit Condition:** Club Manager goes to another tab.
- **Flow of Events:**
  - Club Manager clicks the Calendar button
  - Club Manager sees the events created by their club on a calendar.

## 11.

- **Use Case:** SeeEventCalendarAsStudent
- **Participating Actor:** Inherited from SeeEventCalendar
- **Entry Condition:** Inherited from SeeEventCalendar
- **Exit Condition:** Inherited from SeeEventCalendar
- **Flow of Events:**
  - Student clicks the Calendar button
  - Students see the events created by their club on a calendar.

## 12.

- **Use Case:** SeeEventCalendar
- **Participating Actor:** Initiated by Club Manager OR Initiated by Student
- **Flow of Events:**
  - The user activates the “event calendar” function by clicking the button in the top bar menu.
  - The program responds by displaying the calendar that shows the event belonging to the user or being joined by the user to the user.
  - The user can choose the event and click the more button at the end column.
  - The program responds by displaying the detailed information about the selected event.
- **Entry Condition:** The user clicks the Calendar button
- **Exit Condition:** The user clicks another button
- **Quality Requirements:**
  - The response from the user has to be received no later than 6 minutes.

## 13.

- **Use Case:** ChangePassword
- **Participating Actor:** Student, Club Manager
- **Entry Condition:** Change Password button is clicked on the Profile page
- **Exit Condition:** Password is changed
- **Flow of Events:**
  - Change Password button is clicked
  - User enters their old and new password
  - Password is updated

## 14.

- **Use Case:** SamePassword
- **Participating Actor:** Student, Club Manager
- **Entry Condition:** This use case extends ChangePassword and is initiated when the user tries setting the new password as the old one
- **Exit Condition:** Error message is displayed.
- **Flow of Events:**
  - User tries setting the new password as the old one

- Password is not accepted and no updates are made in the database
- An error message is displayed

## 15.

- **Use Case:** SeeClubListasClub
- **Participating Actor:** Inherited from SeeClubList use case
- **Entry Condition:** Inherited from SeeClubList use case
- **Exit Condition:** Inherited from SeeClubList use case
- **Flow of Events:**
  - Club manager clicks the Clubs button and goes to the Clubs tab
  - All existing clubs are listed

## 16.

- **Use Case:** SeeClubListasStudent
- **Participating Actor:** Inherited from SeeClubList use case
- **Entry Condition:** Inherited from SeeClubList use case
- **Exit Condition:** Inherited from SeeClubList use case
- **Flow of Events:**
  - Student clicks the Clubs button and goes to the Clubs tab
  - All existing clubs are listed

## 17.

- **Use Case:** SeeEventListAsClub
- **Participating Actor:** Inherited from SeeEventList use case
- **Entry Condition:** Inherited from SeeEventList use case
- **Exit Condition:** Inherited from SeeEventList use case
- **Flow of Events:**
  - Club manager clicks the Events button and goes to the Events tab
  - All existing events are listed

## 18.

- **Use Case:** SeeEventListAsStudent
- **Participating Actor:** Inherited from SeeEventList use case
- **Entry Condition:** Inherited from SeeEventList use case
- **Exit Condition:** Inherited from SeeEventList use case
- **Flow of Events:**
  - Student clicks the Events button and goes to the Events tab
  - All existing events are listed

## 19.

- **Use Case:** AddEventCost
- **Participating Actor:** Club Manager

- **Entry Condition:** Event to add a cost to exists
- **Exit Condition:** Cost is added to the event or the given cost exceeds the budget
- **Flow of Events:**
  - Club Manager clicks the Add Event button on the Finance tab
  - Club Manager enters the event name, cost and the spent amount
  - Event cost is either added or denied due to the cost being higher than total budget

## 20.

- **Use Case:** MoreCostThanIncome
- **Participating Actor:** Club Manager
- **Entry Condition:** This use case extends AddEventCost and is initiated when the cost of an event exceeds the club budget.
- **Exit Condition:** Event cost is denied
- **Flow of Events:**
  - Club Manager sets the event cost too high during budget addition
  - The event cost is denied and not added to the database
  - An error message is displayed
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

## 21.

- **Use Case:** RemoveEventCost
- **Participating Actor:** Club Manager
- **Entry Condition:** Event to delete the cost of exists and already has a cost
- **Exit Condition:** Event cost is deleted
- **Flow of Events:**
  - Club Manager clicks the Delete Event button on the Finance tab
  - Club Manager selects the event to delete the cost of
  - Cost of the selected event is deleted
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

## 22.

- **Use Case:** AddEventRequest
- **Participating Actor:** Club Manager
- **Entry Condition:** Club Manager clicks the Add Event button on the Profile page
- **Exit Condition:** Event proposal is added
- **Flow of Events:**
  - Club Manager clicks the Add Event button on the Profile page
  - Club Manager enters event details on a pop up
  - Event proposal is added for confirmation or deletion by the SAC
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**23.**

- **Use Case:** DeleteOwnEvent
- **Participating Actor:** Club Manager
- **Entry Condition:** Club Manager clicks the Delete Event button on the Profile page
- **Exit Condition:** The confirmed event is deleted
- **Flow of Events:**
  - Club Manager clicks the Delete Event button
  - Club Manager selects the event to be deleted
  - Selected event is deleted from the database
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**24.**

- **Use Case:** AddClub
- **Participating Actors:** Initiated by SAC Administrator
- **Flow of Events:**
  - SAC Administrator clicks Manage Clubs button.
  - In the Manage Clubs page, there will be clubs with their code and their events' names in addition to their operations.
  - When SAC Administrator clicks Icon with and addition sign
    - The program responds by opening a pop-up which contains profile name, ID, password and User Type for SAC Administrator to fill.
  - SAC Administrator fills the form and, in the end, clicks the Add Profile Button.
    - The program receives the form and updates the list of clubs with the new clubs' ID and name.
    - The club's password is also stored afterwards.
  - The new profile can be added to the club's list.
- **Entry Condition:**
  - The SAC Administrator opens the Manage Clubs tab.
- **Exit Condition:**
  - The SAC Administrator clicked on another tab.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**25.**

- **Use Case:** DeleteClub
- **Participating Actors:** Initiated by SAC Administrator
- **Flow of Events:**
  - The SAC Administrator opens the Manage Users tab.
    - The program responds by taking the SAC Adminisrtator to the tab.
  - SAC Administrator clicks on the minus icon at the far end of a row for a club.
    - The program responds by removing the club from the database, including all of its events, profile details, finances and password.
- **Entry Condition:**

- The SAC Administrator opens the Manage Clubs tab.
- **Exit Condition:**
  - The SAC Administrator clicked on another tab.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

## 26.

- **Use Case:** SetBudget
- **Participating Actors:** Initiated by SAC Administrator
- **Flow of Events:**
  - The SAC Administrator opens the Manage Clubs tab of the program.
    - The program responds by taking the SAC Administrator to the tab.
  - SAC Administrator clicks on the money icon at the end of the row of a club, in operations section.
    - The program responds by opening a popup form for the SAC Administrator to fill.
  - SAC Administrator fills the form.
    - The program receives the form and updates the budget with the new monthly budget with its reason.
    - The club's new budget is also stored afterwards.
- **Entry Condition:**
  - The SAC Administrator opens the Manage Clubs tab.
- **Exit Condition:**
  - The SAC Administrator clicked on another tab.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

## 27.

- **Use Case:** ConfirmEvent
- **Participating Actor:** SAC Administrator
- **Entry Condition:** Event is not yet confirmed by the SAC
- **Exit Condition:** Event is confirmed
- **Flow of Events:**
  1. SAC Administrator clicks on the confirm button next to an event
  2. Event is confirmed and reaches the approved status
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.
  - The SAC Administrator must click on the Manage Events tab. During the flow of events this invokes OpenManageEventsTab Use Case.

## 28.

- **Use Case:** DeclineEvent
- **Participating Actor:** SAC Administrator
- **Entry Condition:** Event is not yet confirmed by the SAC

- **Exit Condition:** Event is declined and its proposal is deleted
- **Flow of Events:**
  - SAC Administrator clicks on the decline button next to an event
  - Event is declined and deleted from the database
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**29.**

- **Use Case:** DeleteEvent
- **Participating Actor:** SAC Administrator
- **Entry Condition:** Event to be deleted exists.
- **Exit Condition:** Event is deleted.
- **Flow of Events:**
  - SAC Administrator clicks delete event button next to an event.
  - Event is deleted from the database.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**30.**

- **Use Case:** AddUser
- **Participating Actors:** Initiated by SAC Administrator
- **Flow of Events:**
  - The SAC Administrator clicks on the “Add User” button on the top bar, which is marked as the person icon with a plus sign next to it.
    - The program responds by opening a popup form for the SAC Administrator to fill.
  - SAC Administrator fills the form and, in the end, must select the user type as “Student.”
    - The program receives the form and updates the list of students with the new student’s ID and name.
    - The student’s password is also stored afterwards.
- **Entry Condition:**
  - The SAC Administrator opens the Manage Users tab.
- **Exit Condition:**
  - The SAC Administrator clicked on another tab.
- **Quality Requirements:**
  - The user must not remain inactive for longer than 6 minutes.

**31.**

- **Use Case:** DeleteUser
- **Participating Actors:** Initiated by SAC Administrator
- **Flow of Events:**

- The SAC Administrator opens the Manage Profiles tab.
  - The program responds by taking the SAC Administator to the tab.
- SAC Administrator clicks on the minus icon at the far end of a row for a club.
  - The program responds by removing the user from the database, including all of its events, profile details, finances and password.
- **Entry Condition:**
  - The SAC Administrator opens the Manage Users tab.
- **Exit Condition:**
  - The SAC Administrator clicked on another tab.
- **Quality Requirements:**
  - The SAC Administrator must not remain inactive for longer than 6 minutes.

**32.**

- **Use Case:** SeeClubList
- **Participating Actor:** Initiated by Club Manager OR Initiated by Student OR Initiated by Club Advisor
- **Flow of Events:**
  - The user activates the “club list” function by clicking the button in the top bar menu.
  - The program responds by displaying the list of clubs to the user.
  - The user can choose the club and click the more button at the end column.
  - The program responds by displaying the detailed information about the selected club.
- **Entry Condition:** The user clicks the club list button
- **Exit Condition:** The user clicks another button
- **Quality Requirements:**
  - The response from the user has to be received no later than 6 minutes.

**33.**

- **Use Case:** SeeEventList
- **Participating Actor:** Initiated by Club Manager OR Initiated by Student OR Initiated by Club Advisor
- **Flow of Events:**
  - The user activates the “event list” function by clicking the button in the top bar menu.
  - The program responds by displaying the list of events to the user.
  - The user can choose the event and click the more button at the end column.
  - The program responds by displaying the detailed information about the selected event.
- **Entry Condition:** The user clicks Events button
- **Exit Condition:** The user clicks another button
- **Quality Requirements:**
  - The response from the user has to be received no later than 6 minutes.

### **34.**

- **Use Case:** SeePublicEvents
- **Participating Actor:** Unauthorized User
- **Flow of Events:**
  - Unauthorized User opens the application webpage for the first time in a session
  - The program responds by opening the landing page containing all public events such as Zoom meetings.
- **Entry Condition:** Unauthorized User initially opens the application webpage.
- **Exit Condition:** Unauthorized User sees the events on the landing page.

### **35.**

- **Use Case:** SeeFinance
- **Participating Actor:** Club Advisor
- **Flow of Events:**
  - Club Advisor opens the Finance page.
  - The program responds by opening the Finance page containing the budget and expense chart and list of events done by the club.
  - Club Advisor sees the chart and can give feedback to the club according to the current situation of the club finance.
- **Entry Condition:** Club Advisor initially clicks the Finance button at the top bar.
- **Exit Condition:** Club Advisor clicks another button.

### **36.**

- **Use Case:** GiveEventRequestFeedback
- **Participating Actor:** Club Advisor
- **Flow of Events:**
  - Club Advisor opens the EventRequests page by clicking the button at the top bar
  - The program responds by opening the event requests page containing all event requests that are pending for approval by SAC as listed.
  - Club Advisor selects one of the event requests from the list.
  - The program shows the pop-up that shows the selected event in detail.
  - Club Advisor writes the feedback about the event at the end of the pop-up.
  - Club Advisor clicks the send button.
  - The program sends the feedback to SAC.
- **Entry Condition:** Club Advisor clicks Event Requests button at the top bar.
- **Exit Condition:** Club Advisor clicks another button or completes the flow of events.

### **37.**

- **Use Case:** ManageFinance
- **Participating Actor:** Club Manager
- **Flow of Events:**
  - Club Manager enters the amount determined for the club's budget

management together with the reason, at the same time, manager can change the amount determined by the budget change request button and enter a new amount.

- The budget determined in the Finance section is seen and an activity is carried out accordingly.
- **Entry Condition:** Club Manager clicks Button Change Request button left of the bar.
- **Exit Condition:** Club Manager clicks another button or completes the flow of events.
- **Quality Requirements:**  
The budget provided by the school cannot be exceeded.

### 38.

- **Use Case:** ManageOwnEvent
- **Participating Actor:** Club Manager
- **Flow of Events:**
  - Club Manager sends a request to add events that are/will be edited for clubs or to delete added events.
- **Entry Condition:** Club Manager can manage events with Add Event Request button or Delete Event Request button.
- **Exit Condition:** Club Manager clicks another button or completes the flow of events.
- **Quality Requirements:**  
An event that is not added for the club cannot be deleted.

### 39.

- **Use Case:** ConfirmEvent
- **Participating Actor:** SAC Administor
- **Flow of Events:**
  - When the club manager wants to add an event for the club, the SAC Administrator confirms this event or not.
  - The program responds by showing accepted events in the event list.
- **Entry Condition:** Club Manager can add events by using Add Event Request Button.
- **Exit Condition:** SAC Administor can confirm events and it can be seen in the Event List.
- **Quality Requirements:**  
Events that are not approved by the club, students and advisor will not be accepted.

### 40.

- **Use Case:** ManageEvent
- **Participating Actor:** SAC Administor
- **Flow of Events:**
  - SAC Administor accepts a request to add events that are/will be edited for clubs or to rejects events.
  - The program responds by showing accepted requested events in the event list.
- **Entry Condition:** SAC Administor can manage events by adding or rejecting.

- **Exit Condition:** SAC Administrator can confirm events and it can be seen in the Event List, otherwise it cannot.

#### 41.

- **Use Case:** ManageClubs

- **Participating Actors:** Initiated by SAC Administrator

- **Flow of Events:**

***For Addition***

- SAC Administrator clicks Manage Clubs button.
- In the Manage Clubs page, there will be clubs with their code and their events' names in addition to their operations.
- When SAC Administrator clicks Icon with and addition sign
  - The program responds by opening a pop-up which contains profile name, ID, password and User Type for SAC Administrator to fill.
- SAC Administrator fills the form and, in the end, clicks the Add Profile Button.
  - The program receives the form and updates the list of clubs with the new clubs' ID and name.
  - The club's password is also stored afterwards.
- The new profile can be added to the club's list.

***For Deletion:***

- The SAC Administrator opens the Manage Users tab.
  - The program responds by taking the SAC Adminisrtator to the tab.
- SAC Administrator clicks on the minus icon at the far end of a row for a club.
  - The program responds by removing the club from the database, including all of its events, profile details, finances and password.

- **Entry Condition:**

- The SAC Administrator opens the Manage Clubs tab.

- **Exit Condition:**

- The SAC Administrator clicked on another tab.

- **Quality Requirements:**

- The user must not remain inactive for longer than 6 minutes.

#### 42.

- **Use Case:** ManageUsers

- **Participating Actors:** Initiated by SAC Administrator

- **Flow of Events:**

***For Addition***

- The SAC Administrator clicks on the “Add User” button on the top bar, which is marked as the person icon with a plus sign next to it.
  - The program responds by opening a popup form for the SAC Administrator to fill.
- SAC Administrator fills the form and, in the end, must select the user type as “Student.”

- The program receives the form and updates the list of students with the new student's ID and name.
- The student's password is also stored afterwards.

***For Deletion***

- The SAC Administrator opens the Manage Profiles tab.
  - The program responds by taking the SAC Administator to the tab.
- SAC Administrator clicks on the minus icon at the far end of a row for a club.
- The program responds by removing the user from the database, including all of its events, profile details, finances and password.

**- Entry Condition:**

- The SAC Administrator opens the Manage Users tab.

**- Exit Condition:**

- The SAC Administrator clicked on another tab.

**- Quality Requirements:**

The user must not remain inactive for longer than 6 minutes.

### 3.3 Object and Class Diagram

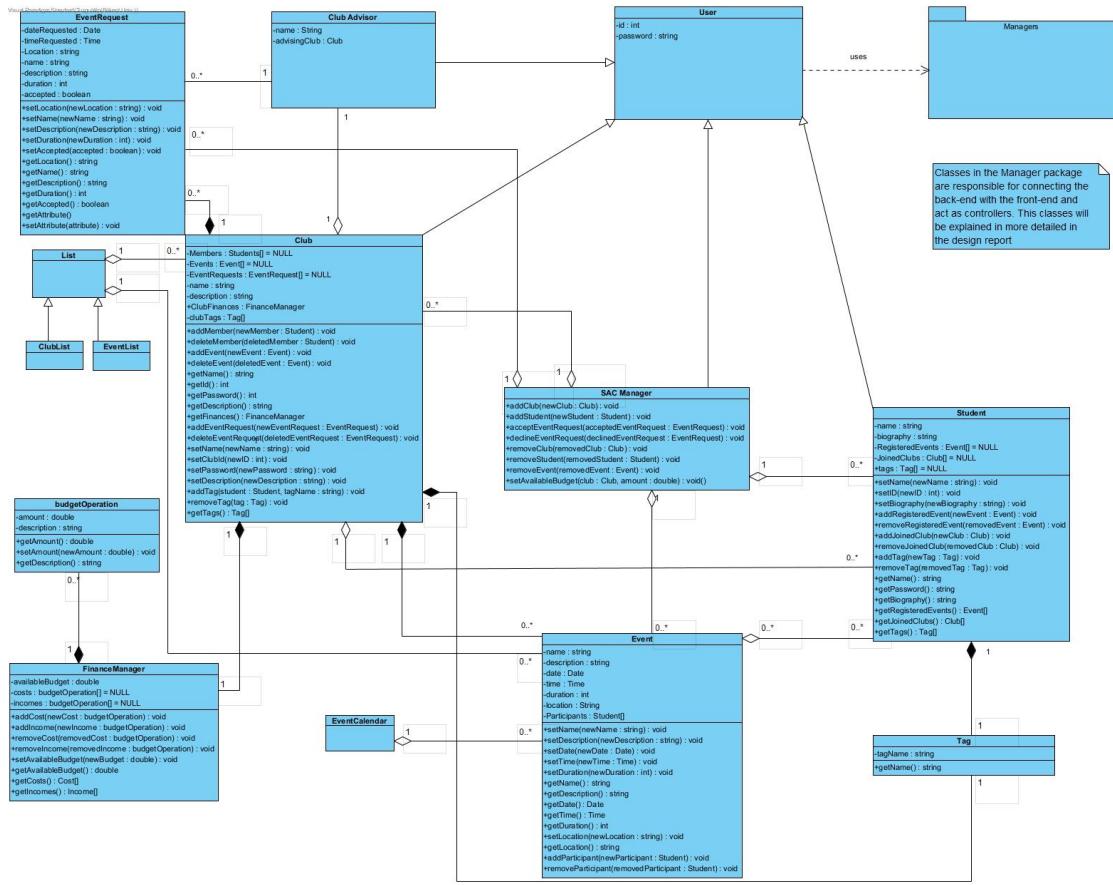


Figure 2: Class diagram

### 3.4 Activity Diagrams

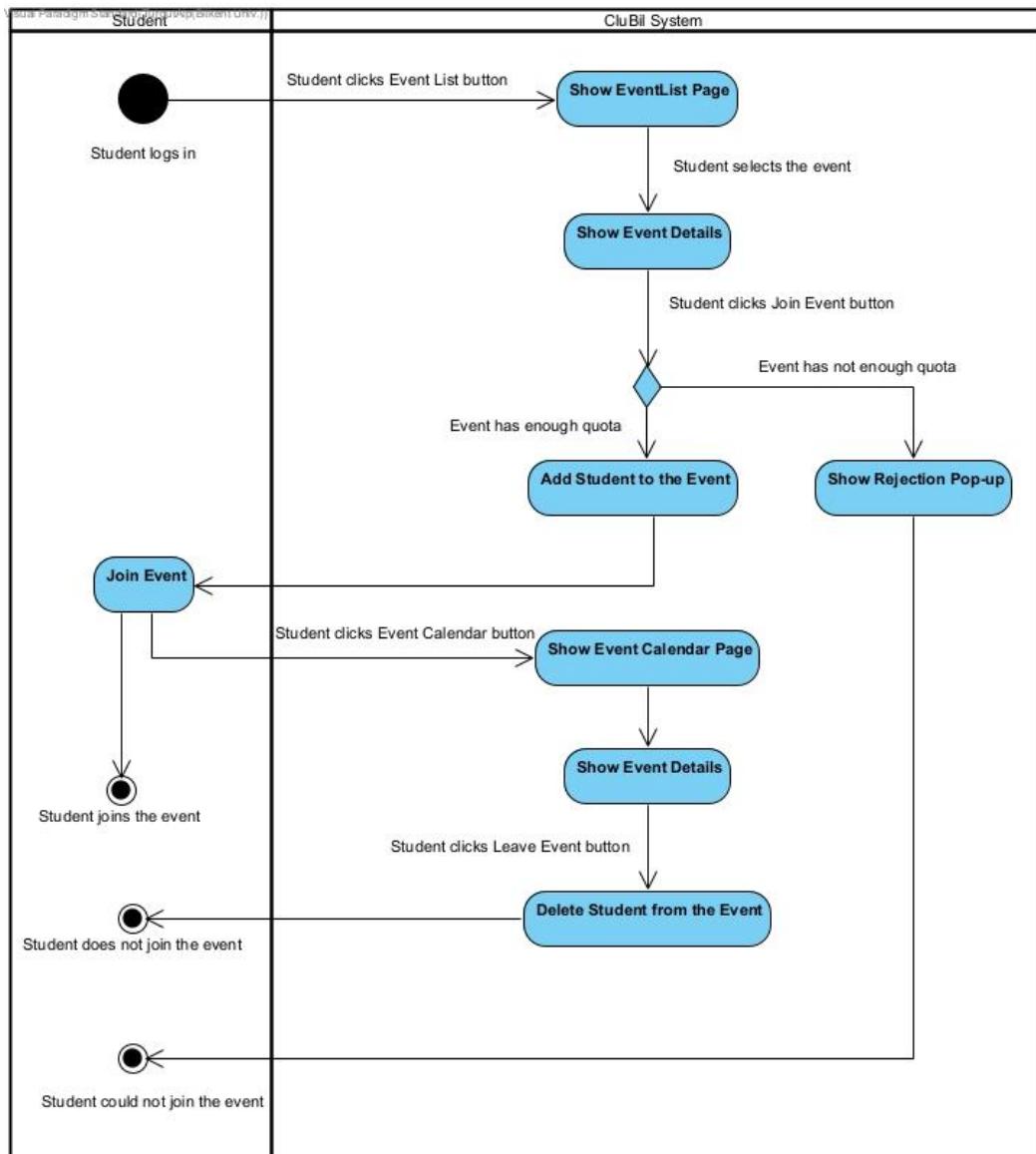


Figure 3: Join and Leave Event Activity Diagram

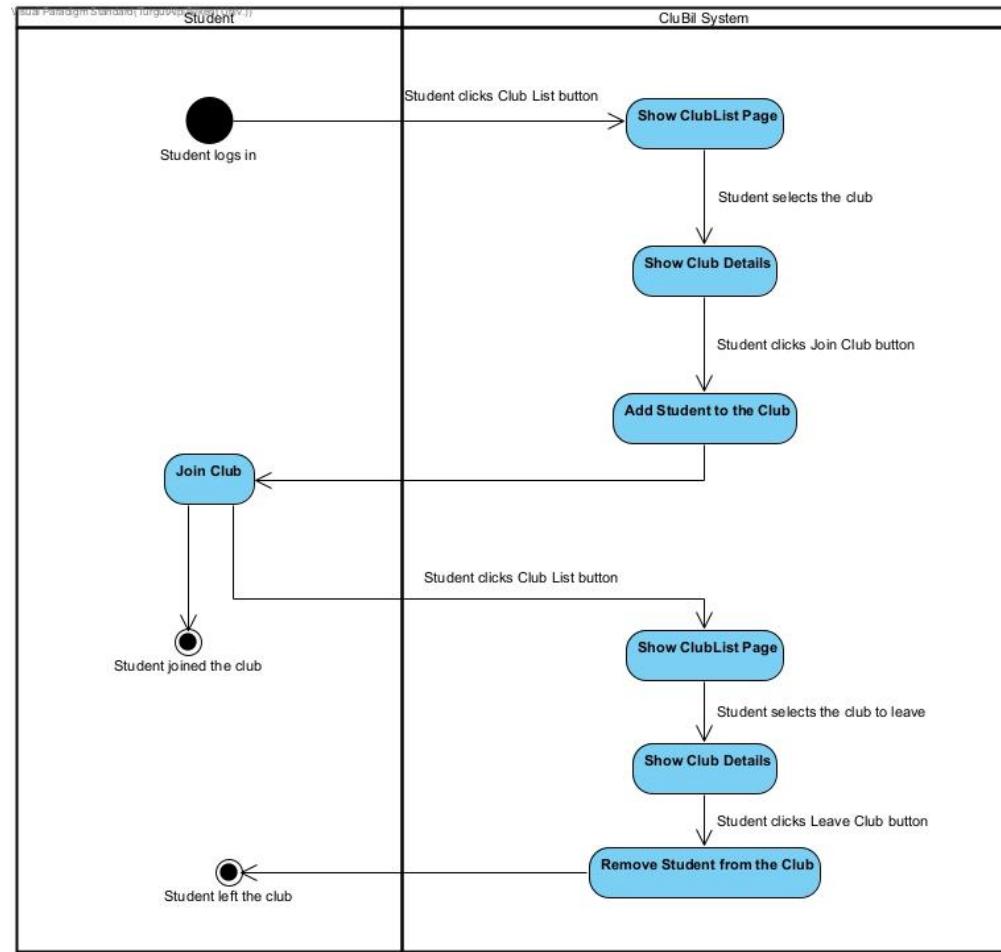


Figure 4: Join and Leave Club Activity Diagram

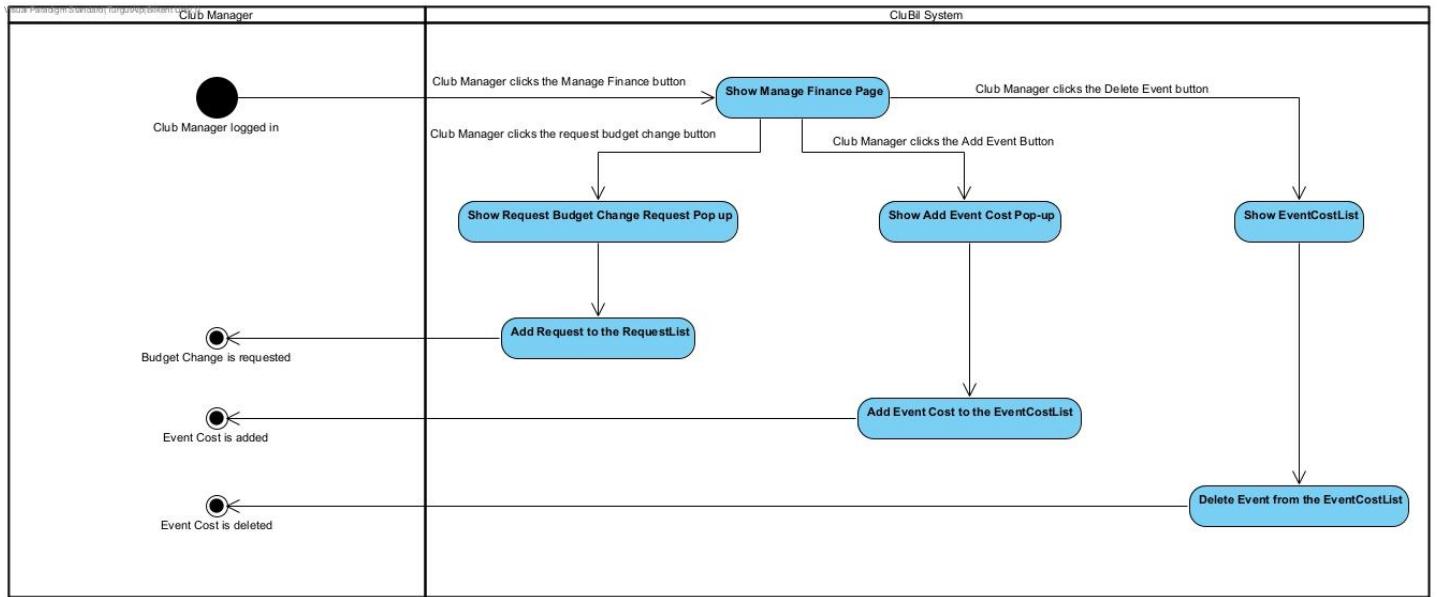


Figure 5: Manage Finance Activity Diagram

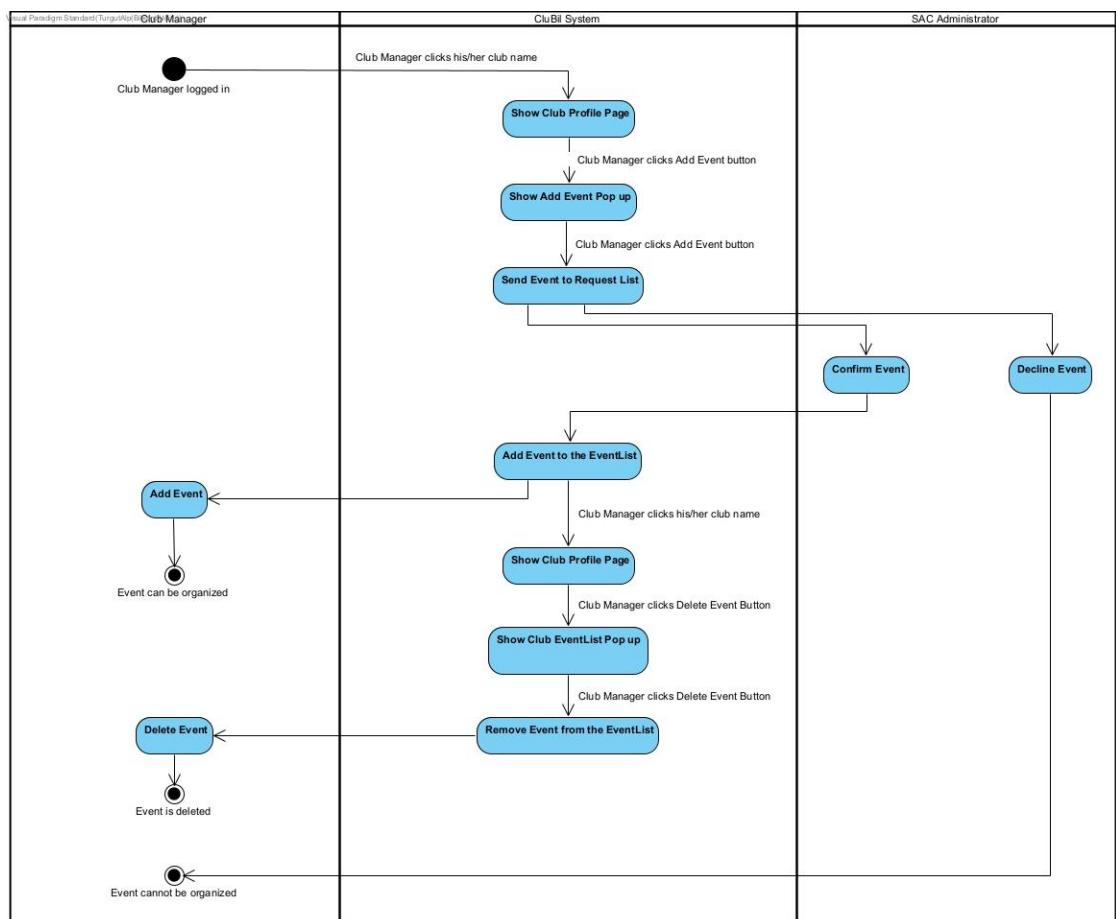


Figure 6: Manage Event Activity Diagram

### 3.5 Sequence Diagrams

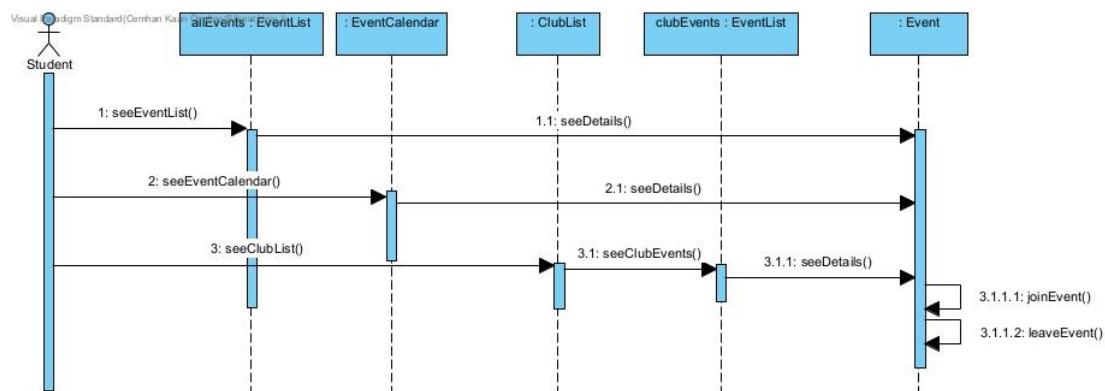


Figure 7: Student Join Event Sequence Diagram

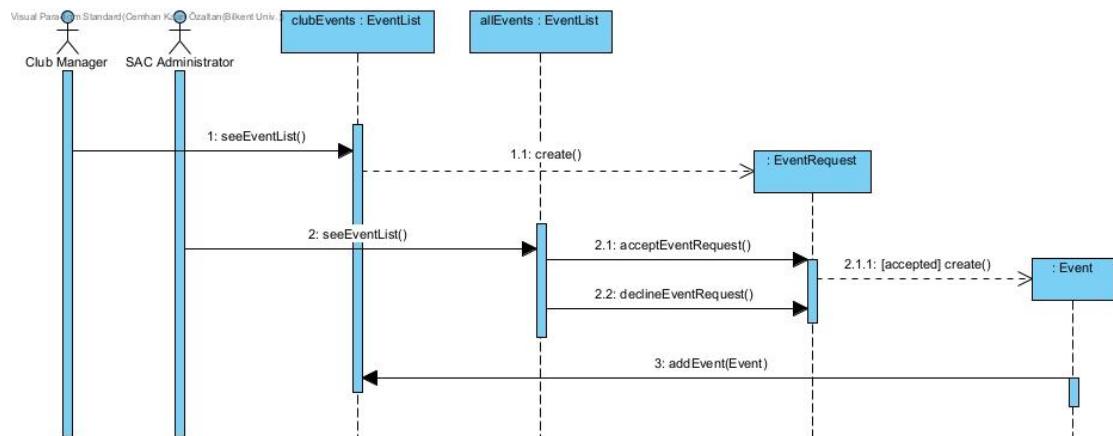


Figure 8: Event Request and Acceptance Sequence Diagram

### 3.6 State Diagrams

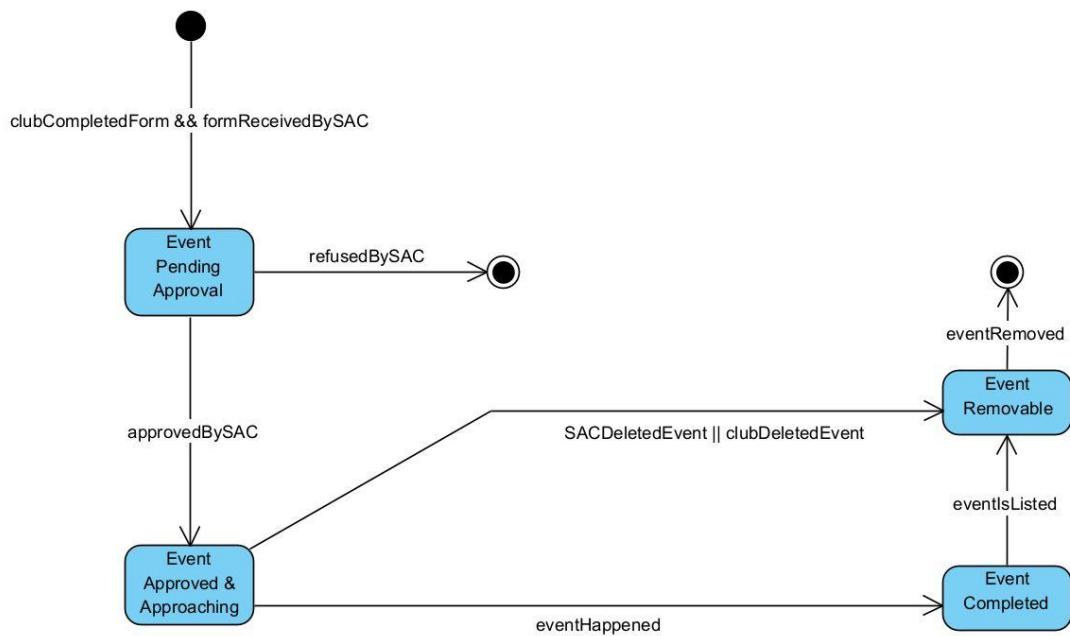


Figure 9: Event State Diagram

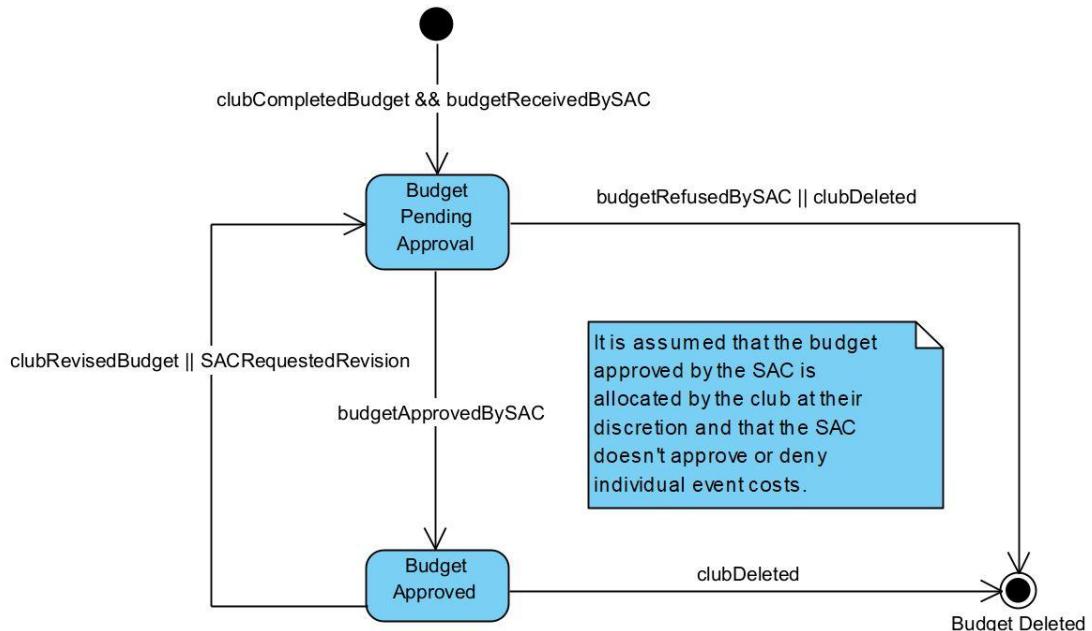


Figure 10: Budget State Diagram

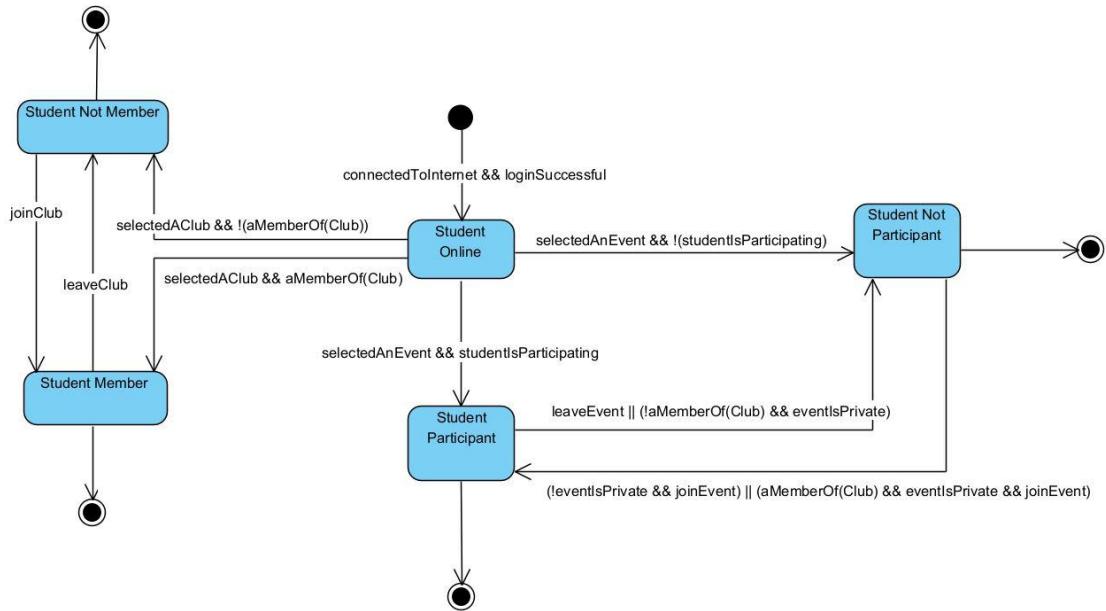
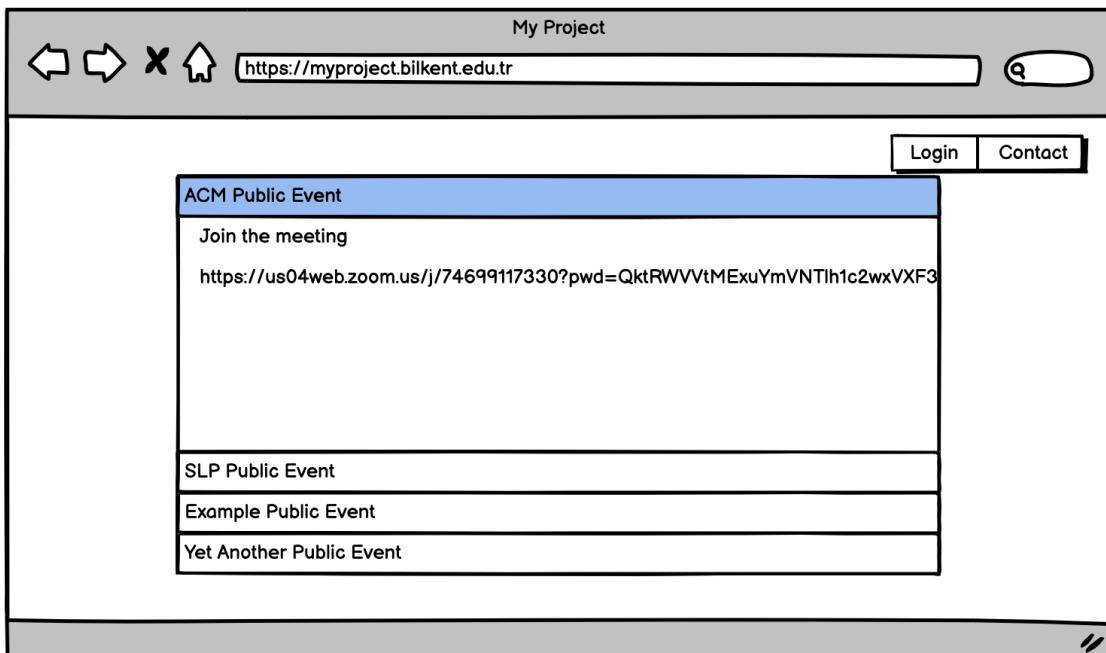


Figure 11: Student State Diagram

### 3.7 Mockup Diagrams (UI Models)



My Project

https://myproject.bilkent.edu.tr

Login Contact

### Student Login

Username:

Password:

### Club Login

Username:

Password:

My Project

https://myproject.bilkent.edu.tr

Login Contact

Contact with us

If you have any trouble please contact us at example@example.com

Made with <3 and Cola Zero and Filter Coffee

My Project

https://myproject.bilkent.edu.tr

Event List
Calendar
Clubs
 
Jack London

Event Name	Club	Date	Time Block	Details
Pizza Partisi	ACM	27.10.2021	18.00-20.00	
BEE with Kariyer.net	OR	28.10.2021	17.30-19.30	
Taylan Yıldız ile Söylesi Etkinliği	Atatürkçü Düşünce Topluluğu	28.10.2021	18.30-19.30	
Linux 102	Google Developer Student Club	29.10.2021	17.30-21.00	
Felsefe Atölyesi	Bilkent Felsefe Topluluğu	29.10.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	

My Project

<https://myproject.bilkent.edu.tr>

Event List	Calendar	Clubs	Finance	search	ACM	User Icon	Print
Event Name	Club	Date	Time Block	Details			
Pizza Partisi	ACM	27.10.2021	18.00-20.00	<b>i</b>			
BEE with Kariyer.net	OR	28.10.2021	17.30-19.30	<b>i</b>			
Taylan Yıldız ile Söylesi Etkinliği	Atatürkçü Düşünce Topluluğu	28.10.2021	18.30-19.30	<b>i</b>			
Linux 102	Google Developer Student Club	29.10.2021	17.30-21.00	<b>i</b>			
Felsefe Atölyesi	Bilkent Felsefe Topluluğu	29.10.2021	17.30-18.30	<b>i</b>			
Example Event	Example Club	01.11.2021	17.30-18.30	<b>i</b>			
Example Event	Example Club	01.11.2021	17.30-18.30	<b>i</b>			
Example Event	Example Club	01.11.2021	17.30-18.30	<b>i</b>			
Example Event	Example Club	01.11.2021	17.30-18.30	<b>i</b>			

My Project

<https://myproject.bilkent.edu.tr>

Event List	Calendar	Pizza Partisi		Jack London	User Icon	Print
Event Name		Pizza Partisi	Leave	Time Block	Details	
Pizza Partisi		V-01		18.00-20.00	<b>i</b>	
BEE with Kariyer.net		27.10.2021		17.30-19.30	<b>i</b>	
Taylan Yıldız ile Söylesi		18.00-20.00		18.30-19.30	<b>i</b>	
Linux 102		ACM		17.30-21.00	<b>i</b>	
Felsefe Atölyesi		The pizza party will be held in V-01. It was only for club members.		17.30-18.30	<b>i</b>	
Example Event				17.30-18.30	<b>i</b>	
Example Event				17.30-18.30	<b>i</b>	
Example Event				17.30-18.30	<b>i</b>	
Example Event				17.30-18.30	<b>i</b>	

My Project

<https://myproject.bilkent.edu.tr>

Pizza Partisi

Event List Calendar

Event Name

Pizza Partisi

BEE with Kariyer.net

Taylan Yıldız ile Söylesi

Linux 102

Felsefe Atölyesi

Example Event

Example Event

Example Event

Example Event

Pizza Partisi  
V-01  
27.10.2021  
18.00-20.00  
ACM

The pizza party will be held in V-01. It was only for club members.

Join Available Quota 31/50

Time Block Details

Time Block	Details
18.00-20.00	i
17.30-19.30	i
18.30-19.30	i
17.30-21.00	i
17.30-18.30	i

Jack London

Join Available Quota 31/50

Time Block Details

Time Block	Details
18.00-20.00	i
17.30-19.30	i
18.30-19.30	i
17.30-21.00	i
17.30-18.30	i

My Project

Event List Calendar Clubs Finance search ACM

https://myproject.bilkent.edu.tr

NOVEMBER 2021

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1 Pizza Partisi	2	3	4
5	6	7	8	9	10	11

My Project

Event List Calendar Clubs search Jack London

https://myproject.bilkent.edu.tr

NOVEMBER 2021

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1 Pizza Partisi	2 BEE with Kariyer.net Taylan Yıldız ile Söyleşi	3 Linux 102 Felsefe Atölyesi	4
5	6	7	8	9	10	11

My Project

https://myproject.bilkent.edu.tr

Event List Calendar Clubs

search

Jack London

Operations

Code	Name	Operations
ACM	Association for Computing Machinery	Show More
SLP	Street Lamp Project	Show More
IEEE	IEEE Student Branch	Show More
FLSFT	Philosophy Society	Show More
CAYYK	Tea Club	Show More
RNKDK	Think Colorfully Club	Show More
HYDST	Animals' Friends Club	Show More
BDOST	Outdoor Sports Society	Show More
Example Club Code	Example Club Name	Show More

My Project

<https://myproject.bilkent.edu.tr>

Event List	Calendar	Clubs	Finance	search	ACM	User Profile	Logout
Code		Name			Operations		
ACM		Association for Computing Machinery			Show More		
SLP		Street Lamp Project			Show More		
IEEE		IEEE Student Branch			Show More		
FLSFT		Philosophy Society			Show More		
CAYYK		Tea Club			Show More		
RNKDK		Think Colorfully Club			Show More		
HYDST		Animals' Friends Club			Show More		
BDOST		Outdoor Sports Society			Show More		
Example Club Code		Example Club Name			Show More		

My Project

<https://myproject.bilkent.edu.tr>

Event List	Calendar	IEEE		Jack London	User Profile	Logout
Code		IEEE Student Branch			Operations	
ACM				Join	Show More	
SLP					Show More	
IEEE		<b>Purpose</b>			Show More	
FLSFT		IEEE, an association dedicated to advancing innovation and technological excellence for the benefit of humanity, is the world's largest technical professional society. It is designed to serve professionals involved in all aspects of the electrical, electronic, and computing fields and related areas of science and technology that underlie modern civilization.			Show More	
CAYYK					Show More	
RNKDK					Show More	
HYDST					Show More	
BDOST					Show More	
Example Club Code		<b>Upcoming Events</b>			Show More	
		Event Name	Date	Time Block	Details	
		Webinar	27.10.2021	18.00-20.00		



My Project

<https://myproject.bilkent.edu.tr>

Event List    Calendar

Budget Change Request

ACM

 ACM

Monthly Budget 25000 TRY

Budget Change Request

Event Name: Pizza Party

Current Monthly Budget 25000 TRY

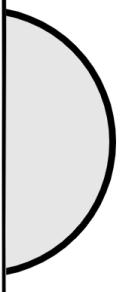
New Monthly Budget:

The reason:

Submit

40 %60

Delete Event -



My Project

<https://myproject.bilkent.edu.tr>

Event List    Calendar

Add Event

ACM

 ACM

Monthly Budget 25000 TRY

Budget Change Request

Event Name: Pizza Partisi

Add Event

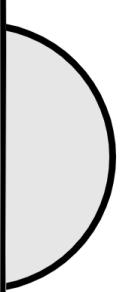
Event Name:

Budget:

Spent:

40 %60

Delete Event -



My Project

https://myproject.bilkent.edu.tr

Event List Calendar Clubs Finance search ACM

Club Info Notifications

ACM

Jack London accepted to join private event

Upcoming Events Create Event + Delete Event -

Event Name	Club	Date	Time Block	More
Pizza Partisi	ACM	27.10.2021	18.00-20.00	
Example Event	ACM	01.11.2021	17.30-18.30	
Example Event	ACM	01.11.2021	17.30-18.30	

Change Password Edit Profile

My Project

https://myproject.bilkent.edu.tr

Event List    Calendar

Change Password

ACM

Old Password

New Password

Change Password

to join private event

Delete Event

Time Block	More
0-20.00	i
0-18.30	i
0-18.30	i

Change Profile

Edit Profile

My Project

https://myproject.bilkent.edu.tr

Event List    Calendar

Add Event

ACM

Quota

Event Type

Select Event Type

Public  
Campus Only  
Private

Upload Photo  
(Optional)

Create Event

to join private event

Delete Event

Time Block	More
0-20.00	i
0-18.30	i
0-18.30	i

Change Profile

Edit Profile

My Project

<https://myproject.bilkent.edu.tr>

Event List    Calendar

Add Event

Club Info

ACM

Change Password

Edit Profile

Event Name: [ ]

Date: [ ]

Time: [ ]

Location: [ ]

Description: [ ]

to join private event

Delete Event [ ]

Time Block More

0-20.00	[i]
0-18.30	[i]
0-18.30	[i]

My Project

<https://myproject.bilkent.edu.tr>

Event List    Calendar

Club Info

ACM

Change Password

Edit Profile

Delete Pizza Partisi

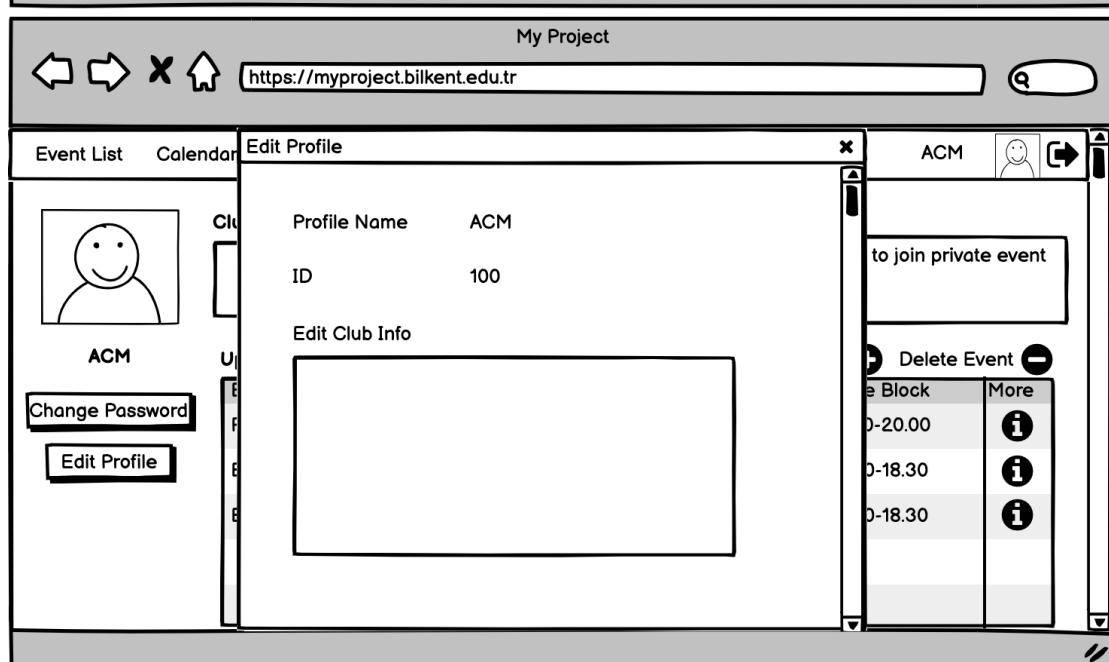
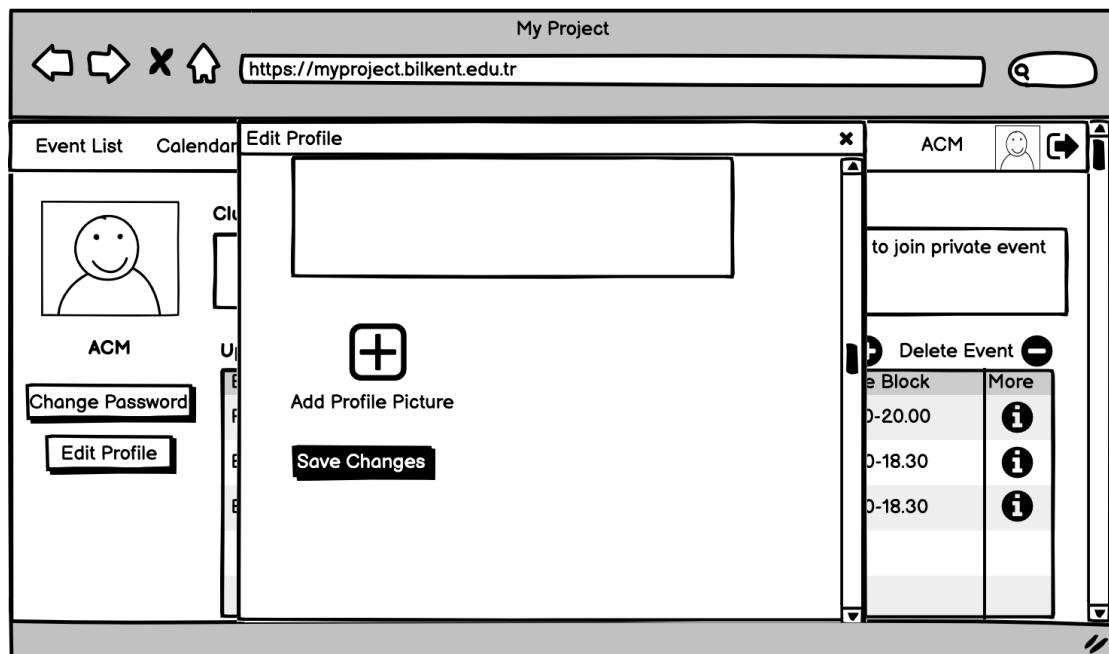
Event Name	Club	Date	Time Block	Delete
Pizza Partisi	ACM	27.10.2	18.00-20.00	[ ]
Example Event	ACM	01.11.20	17.30-18.30	[ ]
Example Event	ACM	01.11.20	17.30-18.30	[ ]

Join private event

Delete Event [ ]

Time Block More

0.00	[i]
30	[i]
30	[i]



My Project

https://myproject.bilkent.edu.tr

Event List    Calendar

Jack London

Fu  
ID:  
Join  
Ro

Old Password

New Password

Change Password

Edit Profile

Change Password

Jack London

Event

My Project

https://myproject.bilkent.edu.tr

Event List    Calendar

Jack London

Fu  
ID:  
Join  
Ro

Add

Change Password

Edit Profile

+

Add Profile Picture

Save Changes

Jack London

Event

My Project

https://myproject.bilkent.edu.tr

Event List    Calendar

Edit Profile

Jack London

Fu  
ID:  
Join  
Ro  
Ab

Profile Name    Jack London

Set new profile name

ID    21700000

Change Password

Edit About Me

Edit Profile

49

My Project

<https://myproject.bilkent.edu.tr>

Manage Events	Add Profile	Admin
Code	 <span>Add Profile</span>	Operations
ACM		
SLP		
IEEE		
FLSFT		
CAYYK		
RNKDK		
HYDST	Profile Name <input type="text"/> ID <input type="text"/> Password <input type="text"/> User Type <input type="button" value="Please Select User Type"/>	
BDOST	Admin Student Club	
Example Club Code		

The screenshot shows a web-based application titled "My Project" with the URL <https://myproject.bilkent.edu.tr>. The main menu on the left includes "Manage Events" and "Manage Budget", with "Manage Budget" currently selected. The budget management section displays a current monthly budget of 25000 TRY and a placeholder for a new monthly budget. A large text area is available for entering a reason. On the right, there is an "Operations" panel containing a grid of 10 budget items, each represented by a dollar sign icon and a minus sign icon.

My Project

<https://myproject.bilkent.edu.tr>

Manage Events	Manage Clubs	Manage Profiles	Admin
Code	Name	Operations	
ACM	Association for Computing Machinery		
SLP	Street Lamp Project		
IEEE	IEEE Student Branch		
FLSFT	Philosophy Society		
CAYYK	Tea Club		
RNKDK	Think Colorfully Club		
HYDST	Animals' Friends Club		
BDOST	Outdoor Sports Society		
Example Club Code	Example Club Name		

My Project

<https://myproject.bilkent.edu.tr>

Manage Events	Manage Clubs	Manage Profiles	search	Admin
Event Name	Club	Date	Time Block	Operation
Pizza Partisi	ACM	27.10.2021	18.00-20.00	
BEE with Kariyer.net	OR	28.10.2021	17.30-19.30	
Taylan Yıldız ile Söylesi Etkinliği	Atatürkü Düşünce Topluluğu	28.10.2021	18.30-19.30	
Linux 102	Google Developer Student Club	29.10.2021	17.30-21.00	
Felsefe Atölyesi	Bilkent Felsefe Topluluğu	29.10.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	
Example Event	Example Club	01.11.2021	17.30-18.30	

My Project

<https://myproject.bilkent.edu.tr>

Manage Events	Add Profile	Admin
Profile Name Jack London Example Student Name	 <input type="button" value="Add Profile"/> Profile Name: <input type="text"/> ID: <input type="text"/> Password: <input type="text"/> User Type: <input type="button" value="Please Select User Type"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">           Admin            Student            Club         </div>	Operations        

My Project

<https://myproject.bilkent.edu.tr>

Manage Events	Manage Clubs	Manage Profiles	 <input type="text" value="search"/>	Admin
Profile Name Jack London Example Student Name		ID 21700000 21X00000	Operations	     

My Project

<https://myproject.bilkent.edu.tr>

Event List	Clubs	Finance	search	Club Advisor		
Event Name	Club	Date	Time Block	Details		
Pizza Partisi	ACM	27.10.2021	18.00-20.00			
BEE with Kariyer.net	OR	28.10.2021	17.30-19.30			
Taylan Yıldız ile Söylesi Etkinliği	Atatürkçü Düşünce Topluluğu	28.10.2021	18.30-19.30			
Linux 102	Google Developer Student Club	29.10.2021	17.30-21.00			
Felsefe Atölyesi	Bilkent Felsefe Topluluğu	29.10.2021	17.30-18.30			
Example Event	Example Club	01.11.2021	17.30-18.30			
Example Event	Example Club	01.11.2021	17.30-18.30			
Example Event	Example Club	01.11.2021	17.30-18.30			
Example Event	Example Club	01.11.2021	17.30-18.30			

My Project

<https://myproject.bilkent.edu.tr>

Event List	Clubs	Finance	search	Club Advisor		
Code	Name	Operations				
ACM	Association for Computing Machinery					
SLP	Street Lamp Project					
IEEE	IEEE Student Branch					
FLSFT	Philosophy Society					
CAYYK	Tea Club					
RNKDK	Think Colorfully Club					
HYDST	Animals' Friends Club					
BDOST	Outdoor Sports Society					
Example Club Code	Example Club Name					

The screenshot shows a web-based application interface. At the top, there's a header bar with icons for back, forward, refresh, and a search bar containing the URL <https://myproject.bilkent.edu.tr>. Below the header is a navigation menu with tabs: Event List, Clubs, Finance (which is highlighted in blue), and Club Advisor. To the right of the menu is a search input field and a user profile icon.

On the left side, there's a sidebar with the ACM logo and the text "Monthly Budget 25000 TRY". The main content area contains a table with three columns: Event Name, Budget, and Spent. The table has two rows: "Pizza Partisi" with a budget of 2000 and spent 1800, and "Example Event" with a budget of XXXX and spent XXXX. To the right of the table is a pie chart divided into two equal halves, labeled "Spent %40" and "Remaining %60".

The image shows a wireframe of a web application interface. At the top, there's a header bar with icons for back, forward, close, and refresh, followed by the URL <https://myproject.bilkent.edu.tr>. Below the header is a navigation bar with links for Event List, Clubs, Finance, a search bar, and a Club Advisor section featuring a user icon. The main content area includes a user profile picture, user details (Full Name: Club Advisor, ID: 00000, Joined Clubs: ACM, Role: Club Advisor), and sections for Club Advisor and About Me. On the right, there's a large Notifications box containing a message from ACM and a feedback button.

My Project

https://myproject.bilkent.edu.tr

Event List Clubs Finance

Club Advisor

search

Club Advisor

Full Name: Club Advisor  
ID: 00000  
Joined Clubs: ACM  
Role: Club Advisor

Notifications

ACM wants to create an example event

Give Feedback

Club Advisor

About Me

Change Password

Edit Profile

My Project

https://myproject.bilkent.edu.tr

Event List Clubs

Club Advisor

Change Password Edit Profile

Fu ID: Join Ro Ab

Give Feedback

Event Name: Example ACM Event

Give Feedback

Submit

Club Advisor

My Project

https://myproject.bilkent.edu.tr

Event List Clubs

Club Advisor

Change Password Edit Profile

Fu ID: Join Ro Ab

Change Password

Old Password

New Password

Change Password

Club Advisor

My Project

https://myproject.bilkent.edu.tr

Event List Clubs

Club Advisor

Fu ID: Join Role

Change Password Edit Profile

Profile Name Club Advisor  
Set new profile name   
ID 00000

Edit About Me

This screenshot shows the 'Edit Profile' page. It includes fields for 'Profile Name' (set to 'Club Advisor') and 'ID' (set to '00000'). There is also a large text area for 'Edit About Me' which is currently empty. On the left sidebar, there are links for 'Event List', 'Clubs', 'Change Password', and 'Edit Profile'. A placeholder image for a profile picture is shown.

My Project

https://myproject.bilkent.edu.tr

Event List Clubs

Club Advisor

Fu ID: Join Role

Change Password Edit Profile

Add Profile Picture

Save Changes

This screenshot shows the same 'Edit Profile' page as above, but with additional options. It features a large empty box for adding a profile picture, indicated by a plus sign icon. Below this is a 'Save Changes' button. The rest of the interface remains the same, with the sidebar and basic profile information.

## **4 Improvements and Summary**

### **4.1 General**

- The indentations and spacing errors have been fixed.
- Additionally, the paragraph alignments have been corrected.

### **4.2 Overview of the System**

- The format of the overview section is changed.
- The content of the overview is changed with the functional requirements.

### **4.3 Changes to Actors**

- A Club Advisor actor is added.
- New two use cases are added for the new actor.

### **4.4 Functional Requirements**

- The content is changed and combined with the overview section.
- The format of the functional requirements is fixed.

### **4.5 Non-Functional Requirements**

- The content of the non-functional requirements is changed.
- Usability is removed from this section.
- Safety and user-friendly interface are added as new non-functional requirements.

### **4.6 Use Case Diagram**

- New actor and use cases are added to the diagram.
- The UI elements are removed.
- SeeEventList, SeeEventCalendar are added as common use cases.
- ManageEvents, ManageFinance, ManageOwnEvent, ManageClubs and ManageUsers are added.

### **4.7 Activity Diagrams**

- Activity diagrams are changed as the representation of the system activity.

- The diagrams that have common activity steps are concatenated as one activity diagram.

## 4.8 Sequence Diagrams

- Sequence diagrams previously included UI elements and database related classes. These were removed and diagrams were reworked.
- Two sequence diagrams displaying the most important functionalities, namely joining clubs and creating events, were added.

## 4.9 State Diagrams

- The state diagrams were completely overhauled. We realized that our selection of objects were completely wrong and overhauled the diagrams accordingly. The new objects are the student, club budget and the event.
- Student objects can transition between joining or leaving clubs and events.
- Budget objects transition between approved or not by the SAC.
- Event objects undergo the process of being approved by SAC, following which they either happen or are marked as deletable. The end state is the deletion of the event.

## 4.10 Class Diagrams

- The new class is created for Club Advisor actor.
- List is created as the parent class of EventList and ClubList.

## **Glossary**

- SAC: Student Activities Center

## **References**

- [1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.