



## CS 319 TERM PROJECT

*Section 3*  
*Group 3B*  
*UNICLAPP*

### Analysis Report

#### Project Group Members:

- |                      |          |
|----------------------|----------|
| 1. Ayberk Yaşa       | 21801847 |
| 2. Fatih Kaplama     | 21802755 |
| 3. Görkem Ayten      | 21802399 |
| 4. Mert Atakan Onrat | 21802520 |

Supervisor: Eray Tüzün

# Contents

<b>1. Introduction</b>	<b>4</b>
<b>2. Proposed System</b>	<b>4</b>
2.1 Overview	4
2.1.1 User Types	4
2.1.2 Sign Up & Sign In	5
2.1.3 Organizing Club Events	5
2.1.4 Event Tracker & Sync Event Calendar	5
2.1.5 Upcoming Events	5
2.1.6 Profile Page	6
2.1.7 Following	6
2.1.8 Campus Map	6
2.1.9 Explore	6
2.1.10 Leaderboard	6
2.1.11 Event History	6
2.2 Functional requirements	7
2.2.1 Sign Up	7
2.2.2 Sign In	7
2.2.3 View Profile	7
2.2.4 Edit Profile	7
2.2.5 Follow Student Club	7
2.2.6 Create Club Event	7
2.2.7 View Upcoming Events	8
2.2.8 Enroll In an Event	8
2.2.9 Cancel the Enrollment	8
2.2.10 Track the Events	8
2.2.11 Rate the Events	8
2.2.12 View the Events' Rates	8
2.2.13 View the Campus Map	9
2.2.14 View the PSI Score	9
2.2.15 View the Event History	9
2.2.16 View the Leaderboard	9
2.2.17 View Club Profile	9
2.2.18 Edit Club Profile	9
2.2.19 Rank Board Members	9
2.3 Nonfunctional requirements	10
2.3.1 Usability	10
2.3.2 Reliability	10
2.3.3 Performance	10
2.3.4 Supportability	10
2.3.5 Scalability	10
2.4 Pseudo Requirements	10
2.4.1 Implementation	10

2.5. System models	11
2.5.1. Use case model	11
2.5.2. Dynamic models	20
2.5.2.1 Sequence Diagrams	20
2.5.2.1.1 Follow Club Sequence Diagram	20
2.5.2.1.2 Organize Event Sequence Diagram	21
2.5.2.1.3 Sign In Sequence Diagram	22
2.5.2.1.4 Enroll in Event Sequence Diagram	23
2.5.2.1.4 Promote Student Sequence Diagram	24
2.5.2.2 Activity Diagrams	26
2.5.2.2.1 Organize Event Activity Diagram	26
2.5.2.2.2 Evaluate Event Activity Diagram	27
2.5.2.2.3 Promote Student Activity Diagram	28
2.5.2.2.4 Attend Event Activity Diagram	29
2.5.2.3 State Diagrams	30
2.5.2.3.1 State of an Event	30
2.5.2.3.2 State of a Student	31
2.5.3. Object and class model	32
2.5.4 User interface - navigational paths and screen mock-ups	35
2.5.4.1 Navigational Path	35
2.5.4.2 Screen Mock-ups	38
<b>3. Improvement Summary</b>	<b>50</b>
3.1 General	50
3.2 Overview	50
3.3 Non-Functional Requirements	50
3.4 Use Case Diagrams	50
3.5 Sequence Diagrams	50
3.6 Activity Diagrams	50
3.7 State Diagrams	50
3.8 Class Diagrams	51
<b>4. Glossary &amp; References</b>	<b>51</b>
4.1 Glossary	51
4.2 References	51

# 1. Introduction

We have decided to implement a student club manager for CS319 - Object-Oriented Software Engineering course's term project. This student club manager project is a website where students can follow the club activities they want. The project aims to provide students and student clubs of Bilkent University with an interactive environment.

The project enables students to follow the clubs that interest them, participate in the activities of the clubs they follow, and rate the activities they attend. Gathering students and student clubs on a single platform makes it easier for students to be informed about club activities and for student clubs to facilitate the processes of promoting student clubs' activities. Moreover, all these features contribute to the socialization of students.

All the user types and the features that will be found in the project are explained in detail in the Overview section to make the outline of the project more comprehensible. Section 2.2 describes what kind of actions actors can do and lists them. Section 2.3 describes some constraints affecting the system's and the user's behavior. Pseudo requirements are discussed in Section 2.4. Section 2.5 lists the use case diagram, class diagram, state transition diagrams, activity diagrams, and sequence diagrams. Finally, the context-specific terms used through the analysis report and references are provided in Section 3.

## 2. Proposed System

### 2.1 Overview

#### 2.1.1 User Types

The system has six kinds of users:

- Student
- Board Member
- Board Chairman
- Club Advisor
- OEM (Öğrenci Etkinlik Merkezi)
- Admin

**Student** can follow student clubs, view upcoming events, attend club activities, track events that students will attend, and view campus map to easily find out the location of the event.

**Board Member** can organize events, post the event announcement, and do whatever student can do.

**Board Chairman** can promote Student to Board Member, demote Board Member to Student, and do whatever board member can do.

**Club Advisor** has different rights than Student and Board Member. Club Advisor can view the event results performed by the club that he/she is the advisor of.

**OEM** can only view the pending events and approve or disapprove them.

**Admin** has the prerogative to access and edit all objects in the software as well as all rights of other types of users. Moreover, only the admin can create a Club Advisor, can promote Student to Board Chairman, and demote Board Chairman to Student.

### **2.1.2 Sign Up & Sign In**

The project will have a sign up and a sign in functionality. Thanks to these features of the application, each user will have a special account. The sign up and sign in page will welcome the club members who are responsible for the clubs' account and the students when entering the web application. On this page, there will be an e-mail and a password section that asks users to type this information to enter the application. Also, there will be a sign up option button for users to enter the application for the first time. To prevent a student from opening more than one account, students will only be able to register with Bilkent email or with their email registered in the Bilkent system. Once registered, all users except for club advisors will be ready to log into their own accounts and have a great experience. Since club advisors will be given accounts by the admin, they don't need to register.

### **2.1.3 Organizing Club Events**

This functionality is for the board members to organize events. Thanks to this feature of the application, board members will be able to easily create their activities. Each one will create an event by specifying the event information. Students will be able to easily see this event and indicate whether they will participate or not. In this way, students will not miss the events of the clubs they are interested in. In addition, each participating student will be able to rate that event at the end of the event. The overall score of the club will be determined by taking the average of these scores. By this way, the students can see the rate and have an idea about how effectively the examined club does their activities. Also, the club members can see the rate and have an idea about the impressions of the students. The clubs can evaluate their rates and work on how to improve their rates.

### **2.1.4 Event Tracker & Sync Event Calendar**

This feature of the application is to help students not to miss an event they desire to go to. On the event tracker page, students will be able to see the club events they marked as "Attending" on a calendar. Thus, they will be able to follow the events they will attend in an orderly manner. In addition to this feature, the calendar on the site will be synchronized one-way with the Google Calendar application, which is used by almost everyone, to make it easier for students to follow the club events they have marked as "Attending".

### **2.1.5 Upcoming Events**

This feature is for giving information about upcoming events. Thanks to the upcoming events page, students will be able to see the events added to the system by the student clubs that students follow. When they click on any upcoming event that appears on this page, they will be able to read detailed information about the event specified by the student club organizing the event. They will select the events they want to attend, from the events that they received detailed information about, and press the "Attending" button. Thus, the system will automatically add this event to the event tracker and trigger synchronization with Google Calendar. The upcoming events page works with the event tracker feature to help students easily learn information about the event they want to go to.

### **2.1.6 Profile Page**

This is the page where students can view personal information. This personal information includes a profile picture, full name, e-mail address, student ID, and department, respectively. Students can also edit their personal information on this page. In addition, each student has a PSI score that is calculated by a specific algorithm based on their activities on the events organized by student clubs. PSI is a personal social activity indicator. It is based on the frequency of attending student club events, the role of the student in the club, etc. Students will experience a very valuable social activity experience thanks to PSI. For instance, the student who has a greater PSI score has more say in the decisions taken jointly by the student clubs.

### **2.1.7 Following**

Students will be able to examine all the student clubs that are members of the application. The students can follow the student clubs that they want to be a part of and participate in their activities and events. Once a student club is followed, the students can see the upcoming events of the followed club. In this way, students can easily manage their calendars and participate in the activity to be carried out by the student club followed.

### **2.1.8 Campus Map**

On this page, there is a bird's eye view of the campus plan of Bilkent University. One of the biggest problems experienced by many students, especially newcomers, finding the building and classroom where student club activities will be held. This page aims to be a solution to this problem experienced by many students. Students will be able to choose the building and class number they want to go to from the list. Then, the selected location will be marked on the map. All these operations can also be done automatically by clicking on the location information on the page with Upcoming Events.

### **2.1.9 Explore**

On this page, students can explore all student clubs. They can view the filtering results by filtering as categories. They can also search student clubs by their name. Moreover, by clicking on the student clubs on this screen, they can view the student clubs' profiles. Finally, they can unfollow the club they follow from this screen and they can follow new clubs.

### **2.1.10 Leaderboard**

On this screen, the competitions between the student clubs will be displayed. Clubs can be sorted with regards to their total number of followers, their average rate, the total number of events, and the total number of participants in their events.

### **2.1.11 Event History**

This screen displays past events. The system automatically switches the event from the upcoming event state to the past event state. Past events seen on this screen can be rated. Moreover, events on this screen can be filtered by event categories. Finally, students can choose either past events of all clubs or past events of the following clubs.

## **2.2 Functional requirements**

### **2.2.1 Sign Up**

- Students, board members, and board chairmen sign up for the application with their Bilkent e-mail and students should set a password to his/her account.

### **2.2.2 Sign In**

- To sign in to the application with a student, a board member, and a board chairman account, students and board member should correctly enter their Bilkent e-mail and the password they created.
- To sign in to the application with a club advisor account, club advisors should correctly enter their account information provided by the admin.

### **2.2.3 View Profile**

- Students, board members, board chairmen, and club advisors can view their profiles by clicking the Profile button.
- Student, board member, and board chairman profiles contain their full name, ID, profile picture, department, email address, and PSI score, respectively.
- Club advisor profile contains club advisors' full name, e-mail address, profile picture, respectively.

### **2.2.4 Edit Profile**

- Students, board members, board chairmen, and club advisors can edit their profile information.
- Students, board members, board chairmen, and club advisors can update or delete their profile pictures.
- Students, board members, board chairmen, and club advisors can change their accounts' passwords.
- Students, board members, board chairmen, and club advisors can submit their HES codes.

### **2.2.5 Follow Student Club**

- Students, board members, and board chairmen can explore the clubs by filtering with respect to the categories of the clubs.
- Students, board members, and board chairmen can follow a club that they want to join or they desire to have some information about their events.
- If students, board members, and board chairmen want, they can unfollow any club that they follow.

### **2.2.6 Create Club Event**

- Board members and board chairmen can create their club's events.
- In this functionality, board members and board chairmen can explain where and when their events will occur.

- Board members and board chairmen can determine how many people can sign up for that event.
- Board members and board chairmen can specify the GE status and the F2F status of their events.

### **2.2.7 View Upcoming Events**

- Students, board members, and board chairmen can see the upcoming events of a club they want by entering the page of that club.
- Students, board members, and board chairmen can filter the upcoming events with regard to their categories.

### **2.2.8 Enroll In an Event**

- Students, board members, and board chairmen can register for events.
- Students, board members, and board chairmen can collect PSI points by enrolling and joining an event.

### **2.2.9 Cancel the Enrollment**

- If students, board members, and board chairmen give up to attend the event they are registered for, they can cancel that event registration.
- If students, board members, and board chairmen cancel their registration, they cannot collect PSI points.

### **2.2.10 Track the Events**

- Students, board members, and board chairmen can follow the activities they participate in the event tracker.
- Students, board members, and board chairmen can sync their Google Calendar with the event tracker.
- Students, board members, and board chairmen can label events as 1 to 5 based on their priority.

### **2.2.11 Rate the Events**

- Students, board members, and board chairmen can rate the events they have participated in.
- Students, board members, and board chairmen can write a comment while rating the event.

### **2.2.12 View the Events' Rates**

- Students, board members, and board chairmen can view the events rates they have participated in.
- Board members and board chairmen can view the evaluation of the event organized by their club.
- Club advisors can view the event evaluation of the club advised by them.



### **2.2.13 View the Campus Map**

- Students, board members, and board chairmen can view the Campus Map by clicking the Campus Map button.
- Students, board members, and board chairmen can also access this page from the event information.
- Students, board members, and board chairmen can select the building and classroom where they want to go.
- Students, board members, and board chairmen can transfer the coordinates of the location to another application such as Google Maps by clicking the Open With button.

### **2.2.14 View the PSI Score**

- Students, board members, and board chairmen can view their PSI in the area just below their profile pictures and Profile page.

### **2.2.15 View the Event History**

- Students, board members, and board chairmen can view the past events that they attended.
- Students, board members, and board chairmen can filter the past events with regards to their categories when viewing them.

### **2.2.16 View the Leaderboard**

- Students, board members, and board chairmen can view the leaderboard of all clubs in the application.
- Students, board members, and board chairmen can sort the clubs by the number of followers, the average rate, the number of events, or the number of total participants.

### **2.2.17 View Club Profile**

- Board members, board chairmen, and club advisors can view the club profile containing club profile picture, club name, the number of followers, the average rate, club description, club category, and board member information.

### **2.2.18 Edit Club Profile**

- Board members and board chairmen can edit the club profile by changing the club profile picture, club description, club category, and board member information.

### **2.2.19 Rank Board Members**

- Board chairman can promote students as board members and demote board members as students.

## **2.3 Nonfunctional requirements**

### **2.3.1 Usability**

- All pages should be displayed in the sidebar so that users can easily access them.
- The titles on the navigation bar, the titles on each page, and the labels on all buttons should be meaningful and self-explanatory so that users who do not read the user manual are able to understand how the website is used from the titles on the navigation bar, the titles on the screen and the labels on the buttons.
- All screens including pop-ups should be reached by being clicked at most 2.
- In order to be a user-friendly website, it is imperative to be consistent in the website layout and design by following a design pattern in which all screens accessed from the sidebar use the same main template.
- Except for the pop-ups, none of the screens on the website must be connected to each other, so users don't have to go backward on the website.
- Web Content Accessibility Guidelines [1] must be followed, which makes content in the website more accessible to users with disabilities.

### **2.3.2 Reliability**

- Users must access their upcoming and past events information %99 of the time without failure.
- Users who close the browser without properly logging out of their account will be automatically logged out.

### **2.3.3 Performance**

- The load time of each page must be less than 1 second.
- The average response time of each button must be about 0.1 seconds.
- The website must keep the above-mentioned times the same, up to 15000 simultaneous users.

### **2.3.4 Supportability**

- User feedback will be evaluated continuously and if there is any bug reported by users, it will be assigned to a developer within 24 hours.

### **2.3.5 Scalability**

- When the daily traffic of this website, which has 50000 visitors per day, exceeds this number, the bandwidth limit of this website that is allocated to the hosting plan should not be exceeded.

## **2.4 Pseudo Requirements**

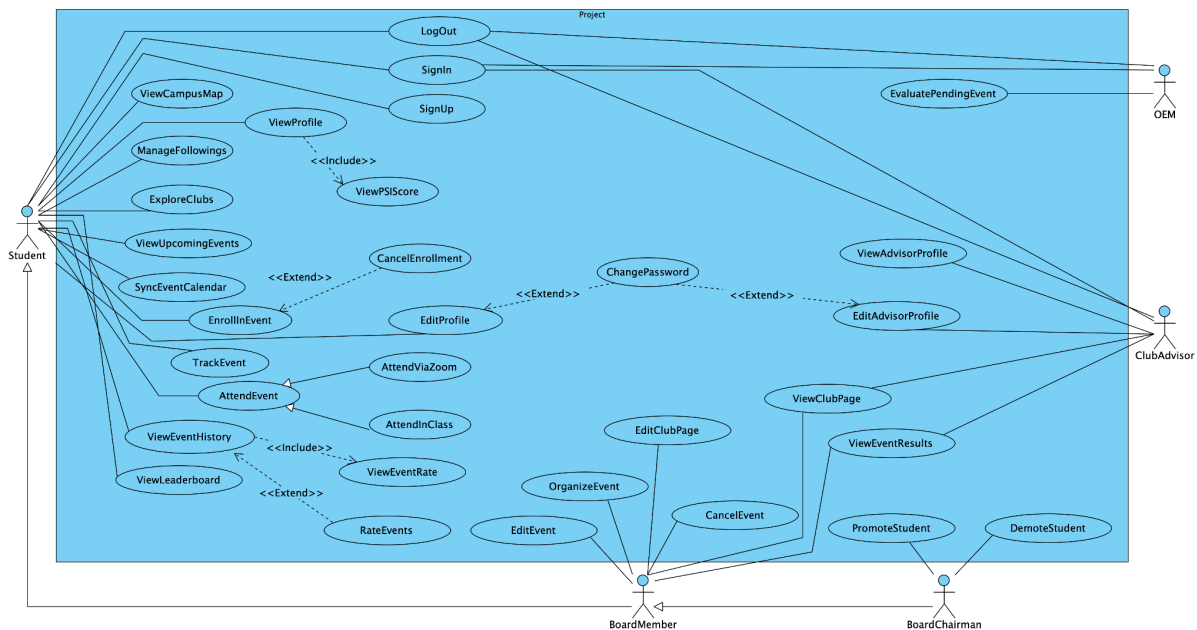
### **2.4.1 Implementation**

- Django [2] must be used at the back-end side.
- Vue.js [3] must be used at the front-end side.

- The web app must be used with any type of browser on a computer, i.e., the web app should not be used with a mobile device to get a neat layout.
- PostgreSQL [4] must be used for the database.

## 2.5. System models

### 2.5.1. Use case model



**Figure 1: Use Case Diagram of Project**

**Use Case Name:** SignIn

**Participating Actors:** Initiated by Student  
Initiated by Club Advisor  
Initiated by OEM

**Flow of Events:**

1. User enters a valid email registered in the system.
2. User enters a valid password registered in the system.
3. User logs in to the system.

**Entry Conditions:** User enters the web page.

**Exit Conditions:** User logs in to the system.

**Quality Requirements:** Email and password field cannot be empty.

**Use Case Name:** SignUp

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student enters a valid name.
2. Student enters an email registered in the Bilkent system.
3. Student enters a valid identification number.
4. Student enters a department registered in Bilkent.

5. Student enters a password.
6. Student signs up to the system.

**Entry Conditions:** Student starts filling out the form.

**Exit Conditions:** Student signs up to the system.

**Quality Requirements:** Students should enter an email registered in the Bilkent system and enter a valid identification number.

---

**Use Case Name:** LogOut

**Participating Actors:** Initiated by Student  
Initiated by Club Advisor  
Initiated by OEM

**Flow of Events:**

1. User logs out the system.
2. The system shut down the session of the user.

**Entry Conditions:** User logs out the system.

**Exit Conditions:** the system shut down the session of the user.

**Quality Requirements:** There must be internet connection.

---

**Use Case Name:** ViewCampusMap

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student selects the place s/he wants to learn its location.
2. System retrieves data from Google Maps and displays the location of that place on the screen.
3. Student views the location on the campus map.

**Entry Conditions:** Student views Campus Map.

**Exit Conditions:** Student closes Campus Map.

**Quality Requirements:** Students must select the place on the Bilkent Campus.

---

**Use Case Name:** ViewProfile

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student starts viewing profile.
2. System displays student's personal information such as name, id, profile picture, HES code.

**Entry Conditions:** Student views Profile.

**Exit Conditions:** Student close Profile.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** ViewPSIScore

**Participating Actors:** Communicates with ViewProfile.

**Flow of Events:**

1. Student starts viewing profile.
2. System retrieves data of PSI for that student and displays it on the screen

**Entry Conditions:** Student views Profile.

**Exit Conditions:** The system displays the data on the screen.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** ManageFollowings

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student searches a club by selecting the categories.
2. Student unfollows a club existing in the table of searching clubs.
3. System receives the new status of this club for this student and saves it.

**Entry Conditions:** Student starts searching for a club.

**Exit Conditions:** Student changes the status of this club for himself/herself.

**Quality Requirements:**

---

**Use Case Name:** EnrollInEvent

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student decides to enroll in the event.
2. System enrolls this student to that event and updates the participants of that event.

**Entry Conditions:** Student starts looking at events.

**Exit Conditions:** Student finishes the enrollment process.

**Quality Requirements:** It is not possible to register for events on the same date.

---

**Use Case Name:** CancelEnrollment

**Participating Actors:** Communicates with EnrollInEvent

**Flow of Events:**

1. Student gives up participating in the event.
2. System cancels the enrollment of this student for that event and updates the participants of that event.

**Entry Conditions:** Student starts looking at events that he/she will attend.

**Exit Conditions:** Student finishes the cancellation process.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** ExploreClubs

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student searches a club by selecting the categories.
2. Student follows a club existing in the table of searching clubs.
3. System receives the new status of this club for this student and saves it.

**Entry Conditions:** Student starts searching for a club.

**Exit Conditions:** Student changes the status of this club for himself/herself.

**Quality Requirements:** Only clubs registered to the Bilkent system can be displayed.

---

**Use Case Name:** ViewUpcomingEvents

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student searches events by selecting the predefined categories or by entering an event name.
2. Student selects either the following part which contains only the events that will be performed by the following clubs or the all part which contains all upcoming events.
3. The system displays events related to the above configurations.

**Entry Conditions:** Student starts searching for the upcoming events.

**Exit Conditions:** The system displays the related events.

**Quality Requirements:** Only clubs registered in Bilkent should be entered in the search section.

---

**Use Case Name:** SyncEventCalendar

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student enters an email and a password belonging to the Google account.
2. Student sends a request for the synchronization.
3. System checks the information entered and if it is valid it synchronizes with Google Calendar.
4. If the information is not valid the system shows an error message.

**Entry Conditions:** Student enters email and password.

**Exit Conditions:** The system synchronizes or shows an error message.

**Quality Requirements:** Email and password entered must be valid.

---

**Use Case Name:** ViewEventHistory

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student searches events by selecting the predefined categories or by entering an event name.
2. Student selects either the following part which contains only the events that were performed by the following clubs or the all part which contains all past events.
3. The system displays events related to the above configurations.

**Entry Conditions:** Student starts searching for the past events.

**Exit Conditions:** The system displays the related events.

**Quality Requirements:** Only clubs registered in Bilkent should be entered in the search section.

---

**Use Case Name:** RateEvents

**Participating Actors:** Communicates with ViewEventHistory

**Flow of Events:**

1. Student views event history that Student has attended.

2. Student selects the event to be rated.
3. The student votes for the selected event.
4. The system updates the average event rate accordingly and saves it.

**Entry Conditions:** Student selects the event to be rated.

**Exit Conditions:** The system updates the average event rate.

**Quality Requirements:** Students must vote for an event with a score between 0 and 5.

---

**Use Case Name:** ViewEventRate

**Participating Actors:** Communicates with ViewEventHistory

**Flow of Events:**

1. Student views event history that Student has attended.
2. The system shows the event rate for each event on the screen.

**Entry Conditions:** Student views the event history.

**Exit Conditions:** The system shows necessary information.

**Quality Requirements:** In order to display event rate, that event must be rated.

---

**Use Case Name:** EditProfile

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student opens profile.
2. Student changes the profile photograph, HES code, or password.
3. Student submits new information about the profile.
4. System checks this information and if it is valid, the system updates it and saves the new one.

**Entry Conditions:** Student opens profile.

**Exit Conditions:** The system updates the account information.

**Quality Requirements:** Changed information must be valid.

---

**Use Case Name:** ChangePassword

**Participating Actors:** Communicates with EditProfile.

**Flow of Events:**

1. Student enters the old password.
2. Student enters a new password.
3. Student confirms the new one by reentering the new password.
4. System checks this information and if it is valid, the system updates it.

**Entry Conditions:** Student starts filling out the form of changing the password.

**Exit Conditions:** The system updates the password of the account.

**Quality Requirements:** The old password entered by Student must match the password registered in the system. The two new passwords entered by Student must be identical to each other.

---

**Use Case Name:** TrackEvent

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student opens the Event Tracker.
2. System displays a calendar and synchronize section on the screen.
3. System displays the events that this student has enrolled
4. Student enters email and password.
5. Student updates the account.
6. System synchronized the events of this student with the google calendar linked to the e-mail address s/he entered.

**Entry Conditions:** Student opens Event Tracker.

**Exit Conditions:** The system synchronized the events.

**Quality Requirements:** Students should enter valid email and password to synchronize the account.

---

**Use Case Name:** AttendEvent

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student selects the date.
2. The system displays the events that this student has registered for the selected day.
3. This student selects and attends the event.

**Entry Conditions:** Student selects the date.

**Exit Conditions:** This student attends the event.

**Quality Requirements:** In order to open a tool, events must be performed online.

---

**Use Case Name:** AttendViaZoom

**Participating Actors:** Communicates with AttendEvent.

**Flow of Events:**

1. If the event is online, the student selects and attends the event via Zoom.
2. The system opens the zoom application and directs the student to the event room.
3. After this student attends the event, the system changes that student's event status to attended.

**Entry Conditions:** Student selects the event.

**Exit Conditions:** The system changes the student's event status.

**Quality Requirements:** The event must be online.

---

**Use Case Name:** AttendInClass

**Participating Actors:** Communicates with AttendEvent.

**Flow of Events:**

1. If the event is face to face, the student goes to the class which the event performs.
2. After this student attends the event, the system changes that student's event status to attended.

**Entry Conditions:** Student goes to the class.

**Exit Conditions:** The system changes the student's event status.

**Quality Requirements:** The event must be face to face.



---

**Use Case Name:** ViewLeaderboard

**Participating Actors:** Initiated by Student

**Flow of Events:**

1. Student opens the Leaderboard.
2. System displays a text field, type of sort selection section on the screen.
3. The system sorts the clubs according to the number of followers by default in a table.
4. The system shows the name, category, number of followers, total rates, total events and total participants of the clubs.
5. If the student chooses a different sorting type such as total rate, total events or total participants, the system sorts the clubs accordingly.

**Entry Conditions:** Student opens Leaderboard.

**Exit Conditions:** The system sorts clubs.

**Quality Requirements:** Only clubs registered in Bilkent should be entered in the search section.

---

**Use Case Name:** OrganizeEvent

**Participating Actors:** Initiated by BoardMember

**Flow of Events:**

1. BoardMember opens the Organize Event
2. System displays a section for event information such as event name, location, date and a table for upcoming events of the club.
3. The BoardMember enters the name, description, location, and date of the event.
4. The system checks the information and updates the upcoming events of that club on the screen.

**Entry Conditions:** The BoardMember enters the Organize Event.

**Exit Conditions:** The system updates the upcoming events.

**Quality Requirements:** There must be no other events on the date chosen by the board member.

---

**Use Case Name:** CancelEvent

**Participating Actors:** Initiated by BoardMember

**Flow of Events:**

1. BoardMember opens the Upcoming Events.
2. The BoardMember selects the event and cancels it.
3. The system detects this change and updates the upcoming events.

**Entry Conditions:** The BoardMember opens the Upcoming Events.

**Exit Conditions:** The system updates the upcoming events.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** EditEvent

**Participating Actors:** Initiated by BoardMember

**Flow of Events:**

1. BoardMember opens the Upcoming Events.
2. The BoardMember selects the event and clicks the edit button.
3. The BoardMember makes some adjustments.
4. The system detects this change and changes the information of this event
5. The system updates upcoming events.

**Entry Conditions:** The BoardMember opens the Upcoming Events.

**Exit Conditions:** The system updates upcoming events.

**Quality Requirements:** There must be no other events on the new date.

---

**Use Case Name:** ViewEventResults

**Participating Actors:** Initiated by BoardMember  
Initiated by ClubAdvisor

**Flow of Events:**

1. BoardMember opens the Event Results.
2. The system displays the name, description, location, date, and result field for the event in a table.

**Entry Conditions:** The boardMember opens the Event Results.

**Exit Conditions:** The system displays the information.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** ViewClubPage

**Participating Actors:** Initiated by BoardMember  
Initiated by ClubAdvisor

**Flow of Events:**

1. BoardMember opens the Club Profile.
2. System displays club's information such as name, followers, rate, profile photograph, description, category, and board members.

**Entry Conditions:** The boardMember opens the Club Profile.

**Exit Conditions:** The system displays the information.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** EditClubPage

**Participating Actors:** Initiated by BoardMember

**Flow of Events:**

1. BoardMember opens Club Profile.
2. The BoardMember changes the name, description, category, board member, or club profile photograph.
3. The BoardMember sends this information.
4. System checks this information and if it is valid it changes them and updates the Club Profile.

**Entry Conditions:** The BoardMember opens Club Profile.

**Exit Conditions:** The system updates the club profile.

**Quality Requirements:** Changed information must be valid.

---

**Use Case Name:** ViewAdvisorProfile

**Participating Actors:** Initiated by ClubAdvisor

**Flow of Events:**

1. Club Advisor opens the Profile page.
2. System displays the name, email, and profile photograph of this advisor.

**Entry Conditions:** The Club Advisor opens the Profile page.

**Exit Conditions:** The system displays all the information.

**Quality Requirements:** There must be internet connection.

---

**Use Case Name:** EditAdvisorProfile

**Participating Actors:** Initiated by ClubAdvisor

**Flow of Events:**

1. Club Advisor opens Profile page.
2. This advisor changes the name, email, password, or profile photograph.
3. This advisor sends this information.
4. System checks this information and if it is valid it changes them and updates the Club Advisor.

**Entry Conditions:** The Club Advisor opens Profile page.

**Exit Conditions:** The system updates the Club Advisor.

**Quality Requirements:** Changed information must be valid.

---

**Use Case Name:** PromoteStudent

**Participating Actors:** Initiated by BoardChairman

**Flow of Events:**

1. BoardChairmen opens Rank Board Member.
2. The BoardChairmen fills the form including the name, identification number, and email.
3. The BoardChairmen sends this information.
4. The system checks whether name, identification, and email entered are valid or not.
5. If the information is valid, the system changes the status of this student as a board member.
6. If not, the system displays an error message on the screen.

**Entry Conditions:** The BoardChairmen opens Rank Board Member.

**Exit Conditions:** The system changes the status of the student or displays an error message.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** DemoteStudent

**Participating Actors:** Initiated by BoardChairman

**Flow of Events:**

1. BoardChairmen opens Rank Board Member..
2. System displays the board members of the club.
3. The BoardChairmen demotes the board member.

4. The system changes the status of this board member as a student.

**Entry Conditions:** The BoardChairmen opens Rank Board Member..

**Exit Conditions:** The system changes the status of the board member.

**Quality Requirements:** There must be an internet connection.

---

**Use Case Name:** EvaluatePendingEvent

**Participating Actors:** Initiated by OEM

**Flow of Events:**

1. System displays the pending events on the screen.
2. OEM confirms the events.
3. The system changes the status of this event as confirmed.

**Entry Conditions:** The system displays the pending events.

**Exit Conditions:** The system changes the status of the event.

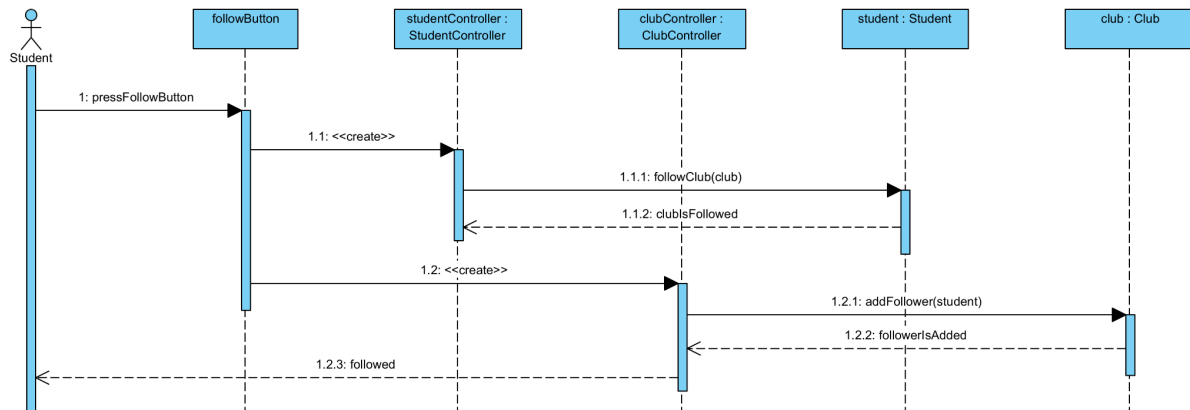
**Quality Requirements:** There must be an internet connection.

## 2.5.2. Dynamic models

### 2.5.2.1 Sequence Diagrams

#### 2.5.2.1.1 Follow Club Sequence Diagram

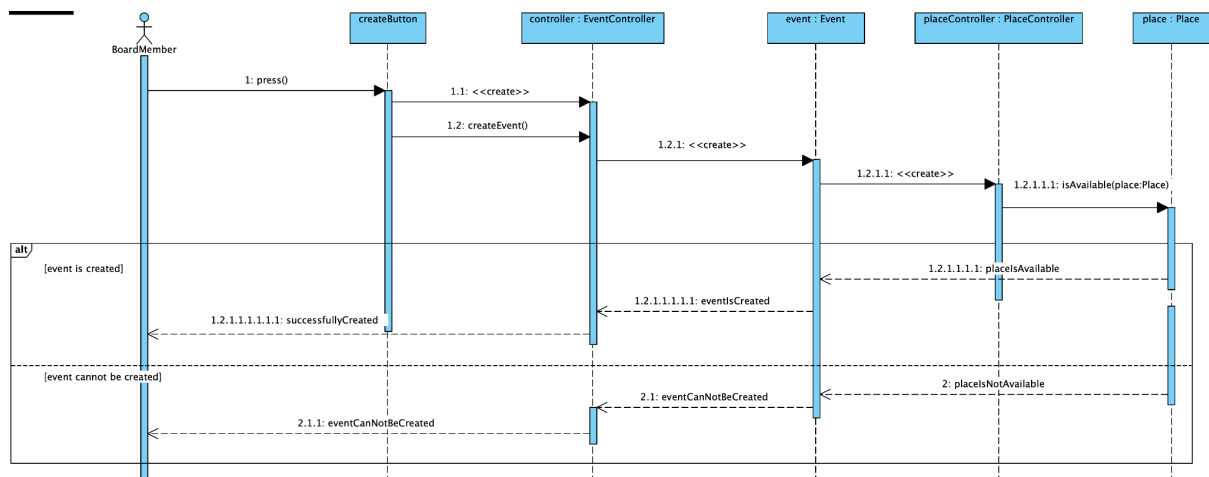
**Scenario:** Student views the Explore and sees the information of the student club from the list. The student follows the student club by clicking the Follow button. After the student clicks the follow button, the club followed by the student is added to the list of clubs followed by the student. After that, the number of followers of that club is incremented by one and this student is added to the followers list of that club.



**Figure 2:** Sequence diagram showing the sequence of events when a student follows a student club.

### 2.5.2.1.2 Organize Event Sequence Diagram

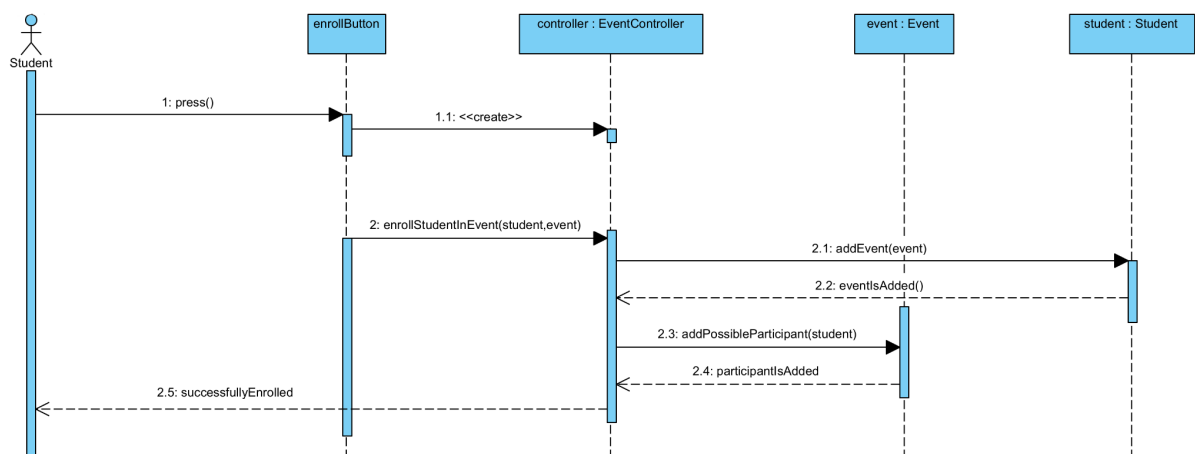
**Scenario:** Board Member creates a club event by filling in the necessary information such as name, place, date. If the place that BoardMember selected is available, an event is created. However, if the place is not available, the event cannot be created and BoardMember receives a warning message.



**Figure 3:** Sequence diagram showing the sequence of events when a board member creates a new student club's event.

### 2.5.2.1.3 Enroll in Event Sequence Diagram

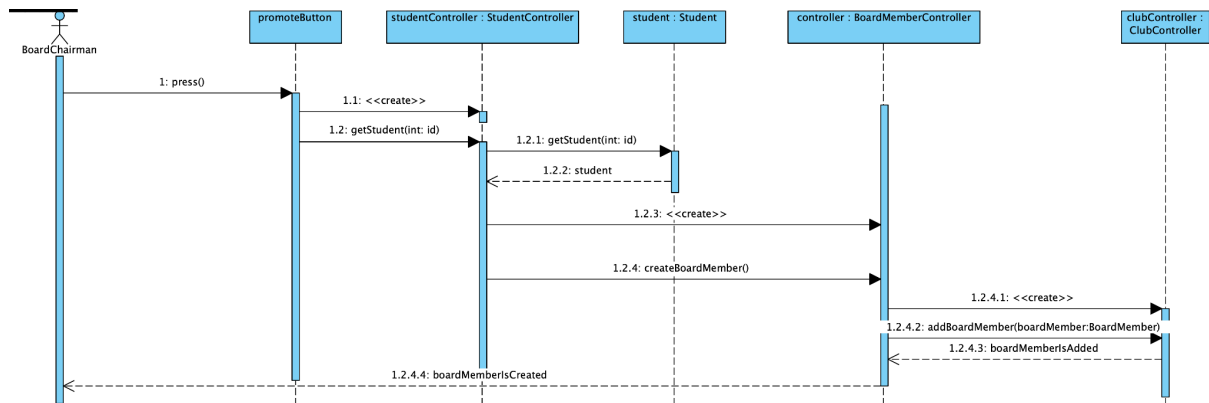
**Scenario:** The student views the upcoming events of different student clubs as a list. Student enrolls, in any event, he/she wants by clicking the enroll button. The system adds the event into the enrolledEvent list of the Student. Then if the event is added correctly, the system adds this student to the possibleParticipants list of events. If the student cancels his/her enrollment the system deletes the event from the enrolledEvent list of the Student and deletes this student from the possibleParticipants list of the event.



**Figure 4:** Sequence diagram showing the sequence of events when a student enrolls in a club event or cancels his/her enrollment.

#### 2.5.2.1.4 Promote Student Sequence Diagram

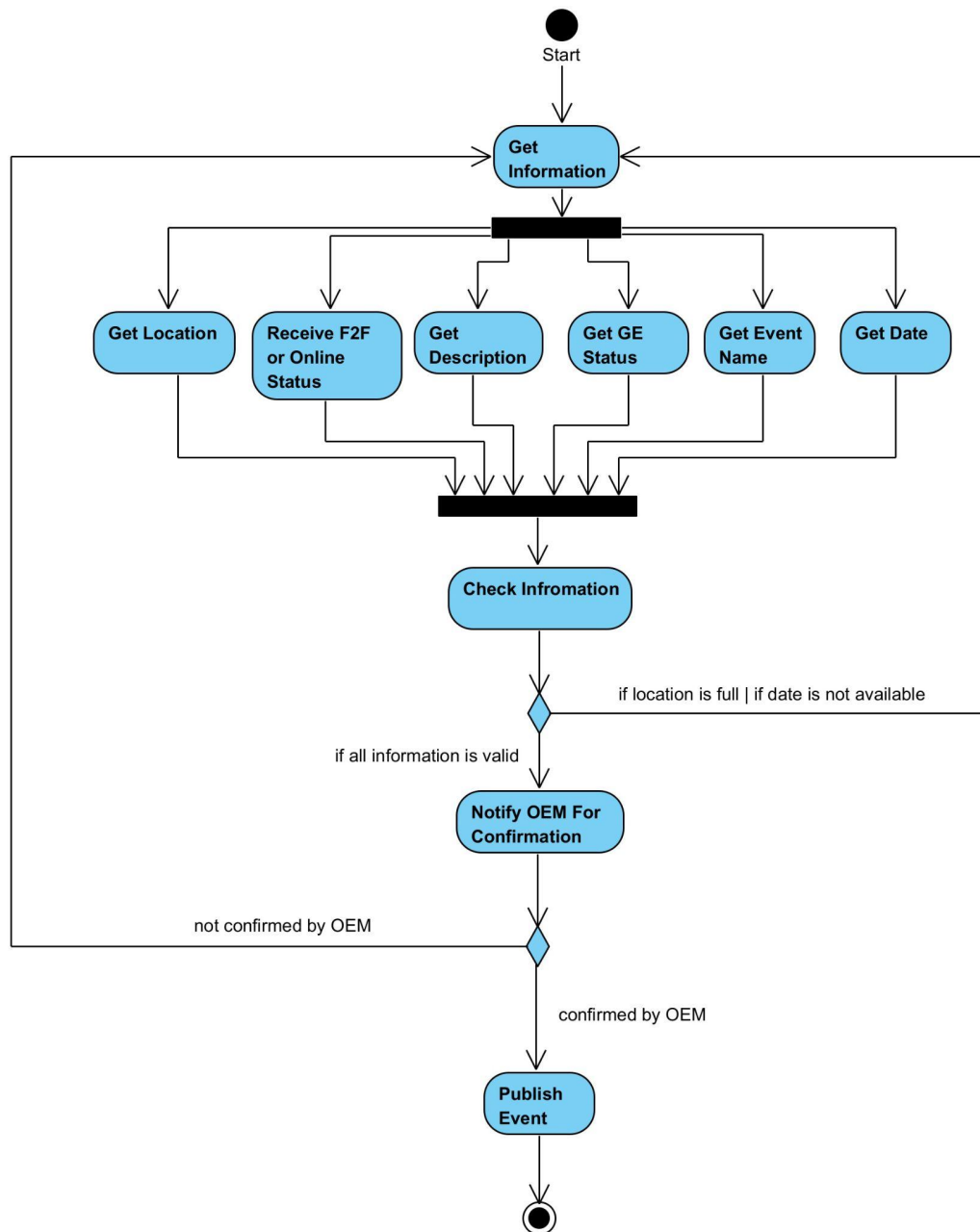
**Scenario:** The Board Chairman clicks the Rank BoardMember. The Board Chairman fills in the information of s student to become Board Member. If the given information is valid, the student will be promoted to Board Member and the System adds this student to the board member list of the student club.



**Figure 5:** Sequence diagram showing the sequence of events when a board chairman promotes a student as a board member

## 2.5.2.2 Activity Diagrams

### 2.5.2.2.1 Organize Event Activity Diagram



**Figure 6:** Activity Diagram for showing organize event activities.

#### Organize Event Activity Diagram Explanation:

When a Club Board Member organizes an event, s/he should fill in some information about the event. S/He should enter the event's location, F2F or online option, description, GE status, name and date. After entering all this information, the information must be checked.

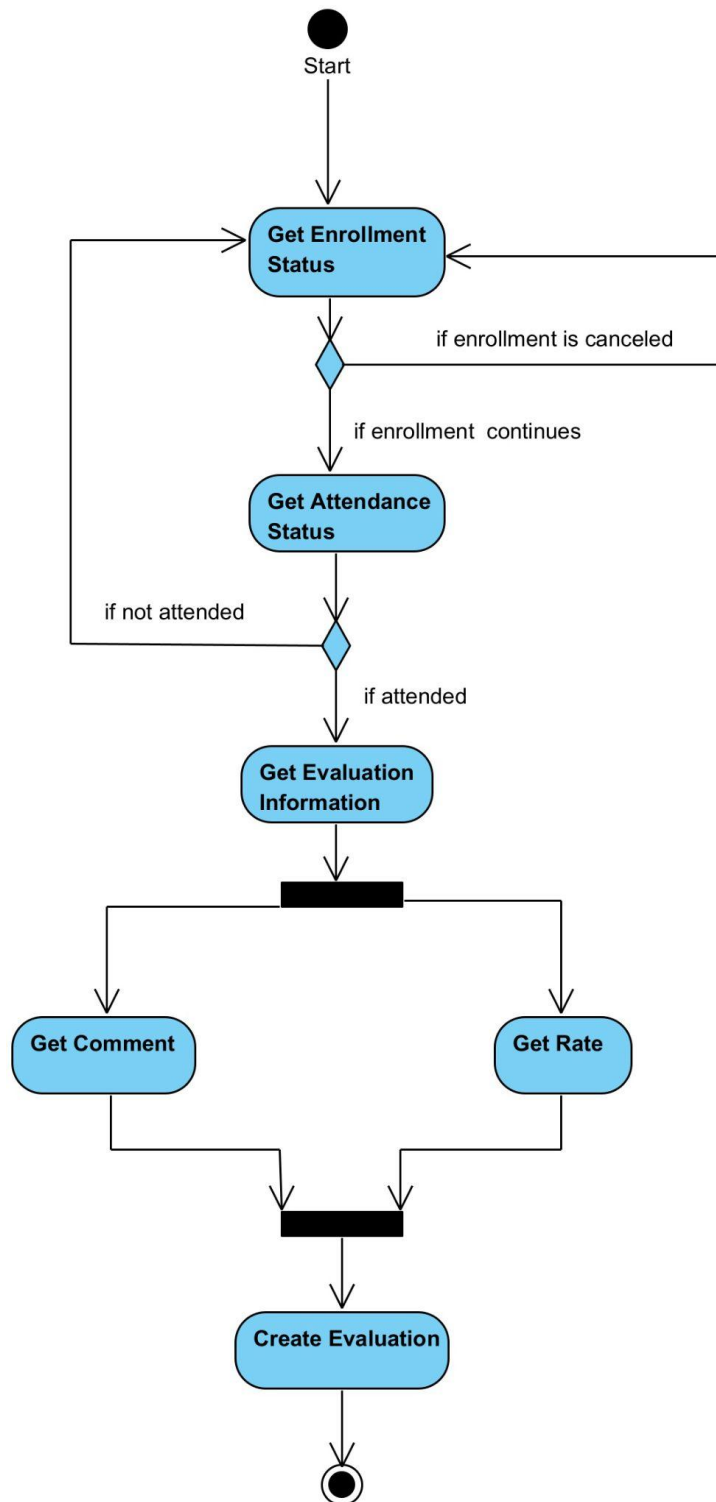
If the location is full or if the date is not available for an event, the board member should again enter another date or a location for the event.

If all the information is valid, the notification will be sent to the OEM for confirmation of the event.

If the OEM does not confirm the event, the new information will be taken from the Board member.

If the OEM confirms the event, the event will be published.

#### 2.5.2.2.2 Evaluate Event Activity Diagram



**Figure 7:** Activity Diagram for showing evaluate event activities.



### Evaluate Event Activity Diagram Explanation:

In order to be able to evaluate an event, the student must first enroll in an event.

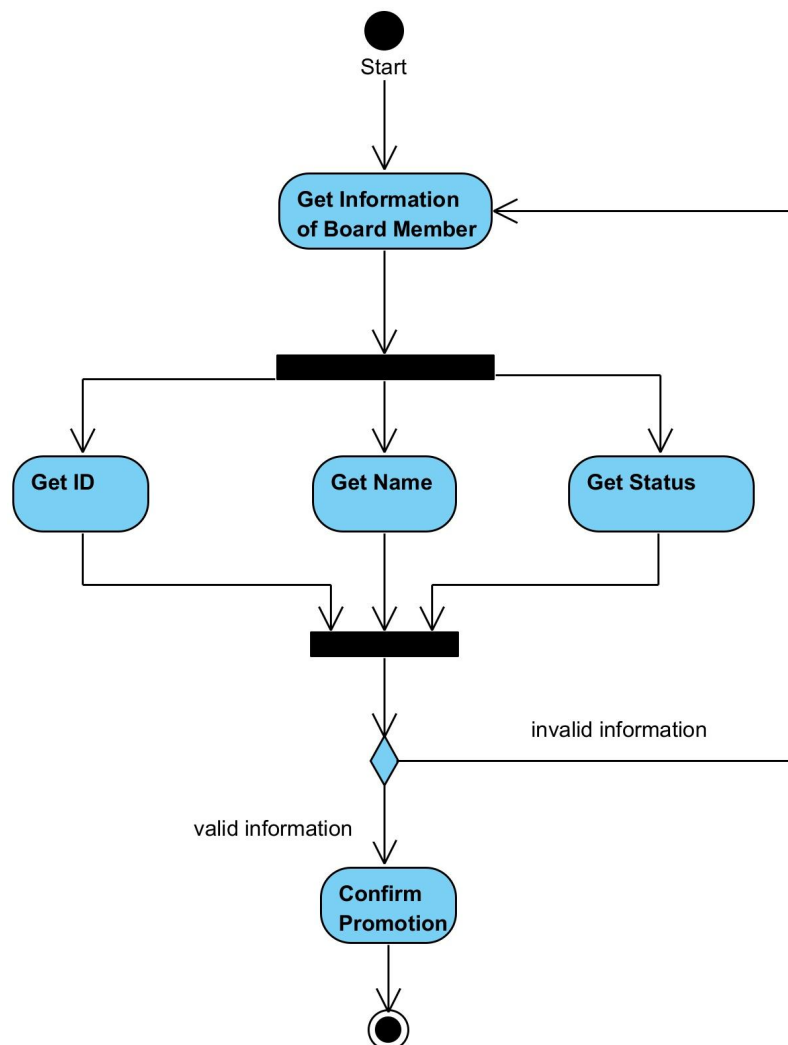
If the student cancels the enrollment for an event and to be able to evaluate an event, the student must again enroll for an event.

If the enrollment continues, the student must attend the event.

If s/he does not attend the event, then in order to be able to evaluate an event s/he must again enroll in an event.

If s/he attends the event, then in order to make an evaluation s/he should write a comment and give rate to the event.

### 2.5.2.2.3 Promote Student Activity Diagram



**Figure 8:** Activity Diagram for showing promote student activities.

### Promote Student Activity Diagram Explanation:

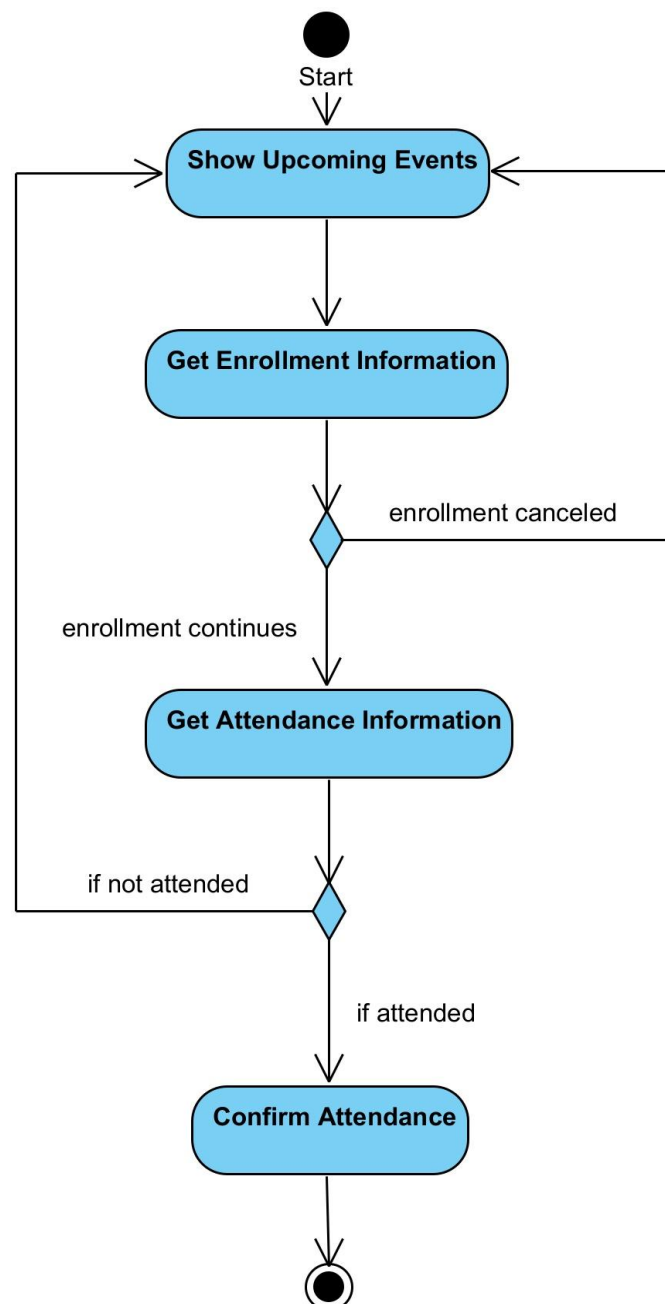
To create, edit and cancel events, a Student should be promoted to a Board Member by the Chairman.

Firstly the Chairman should enter the information of the student, the information should be ID, name and status. Status is the position of the student in the club.

If all information is not valid, the Chairman should enter the student's information again.

If all information is correct the promotion will be confirmed.

#### 2.5.2.2.4 Attend Event Activity Diagram



**Figure 9:** Activity Diagram for showing attending an event activities.

### Attend Event Activity Diagram Explanation:

When a student wants to attend an event, firstly, s/he must enroll in an event that s/he chooses from the upcoming events.

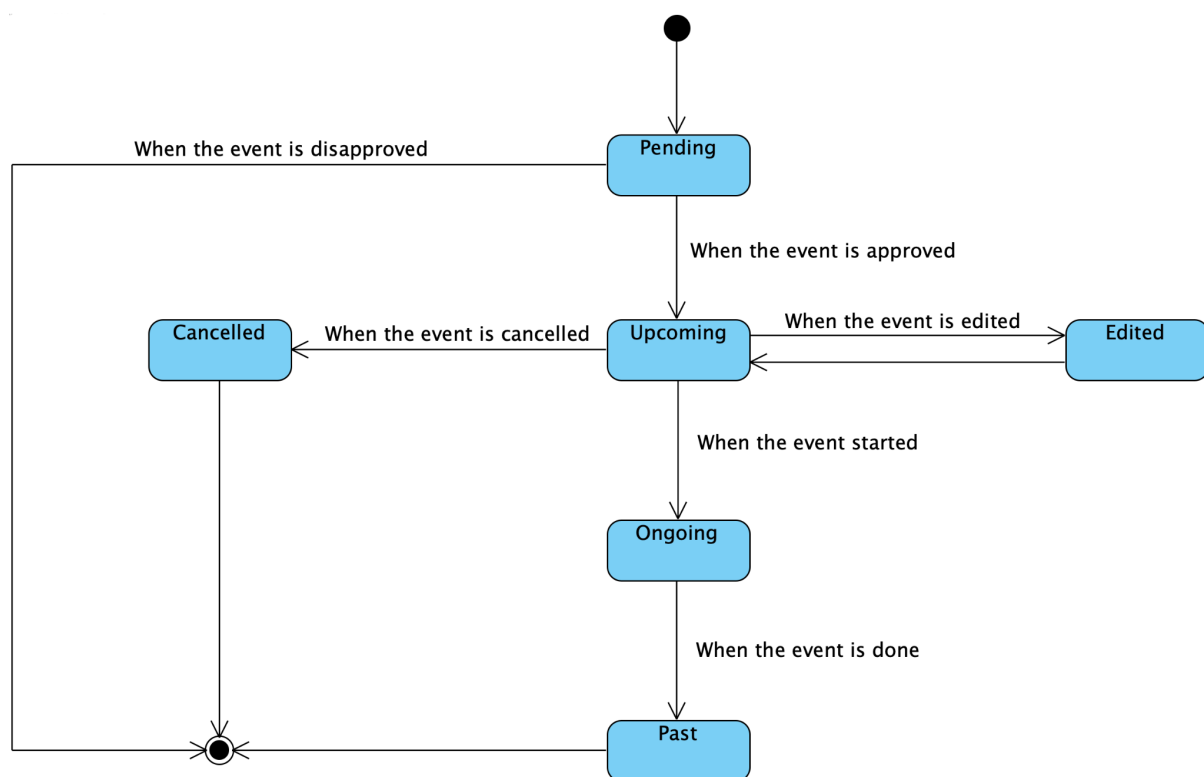
If the enrollment is canceled, the student must again choose an event from the upcoming events and enroll in this event.

If the enrollment continues, the student must attend the event. If the student does not attend the event that s/he enrolled, the student must again choose an event from the upcoming events and enroll in this event.

If the student attends the event, the attendance will be confirmed.

## 2.5.2.3 State Diagrams

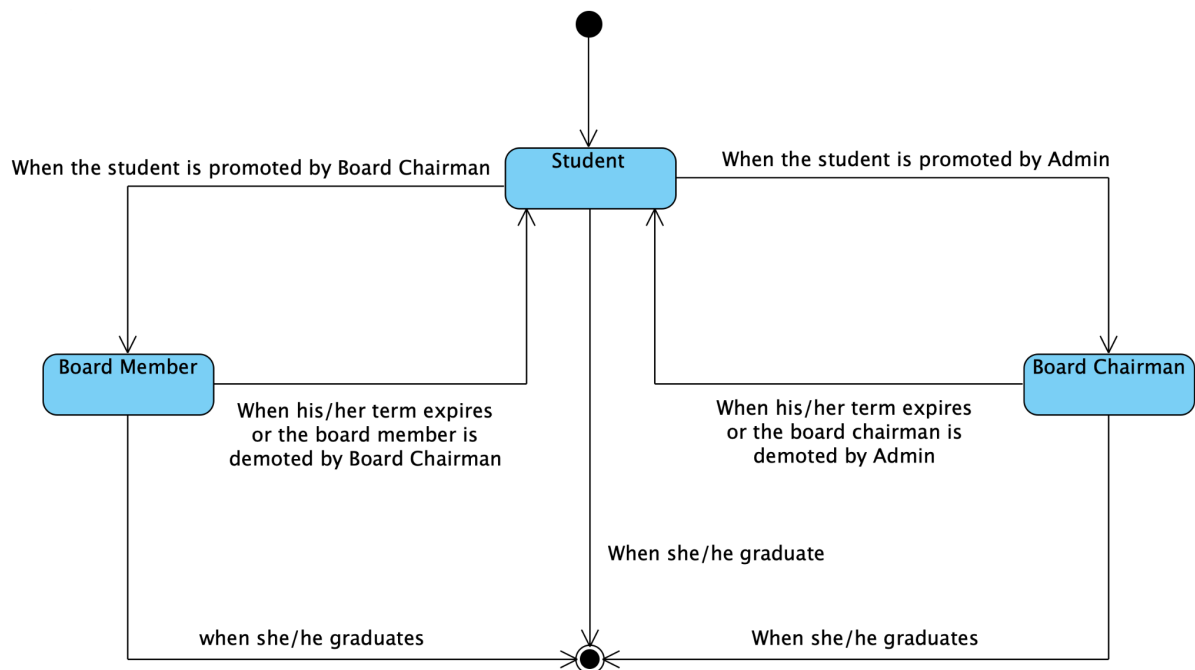
### 2.5.2.3.1 State of an Event



**Figure 10:** State Diagram of an Event object

The initial state of an event is pending. This state goes to the upcoming state when the event is approved by OEM, whereas this state goes to the final state when the event is disapproved by OEM. At the upcoming state, when the event is edited, the new state becomes edited. From here, the old state is turned back to after the edition process is done. At the upcoming state, when the event is cancelled, the upcoming state goes to the cancelled state. If the event is not cancelled and started, the state transitions into the ongoing state. When the event is done, the new state becomes past. Finally, both past state and cancelled state are directly linked to the final state.

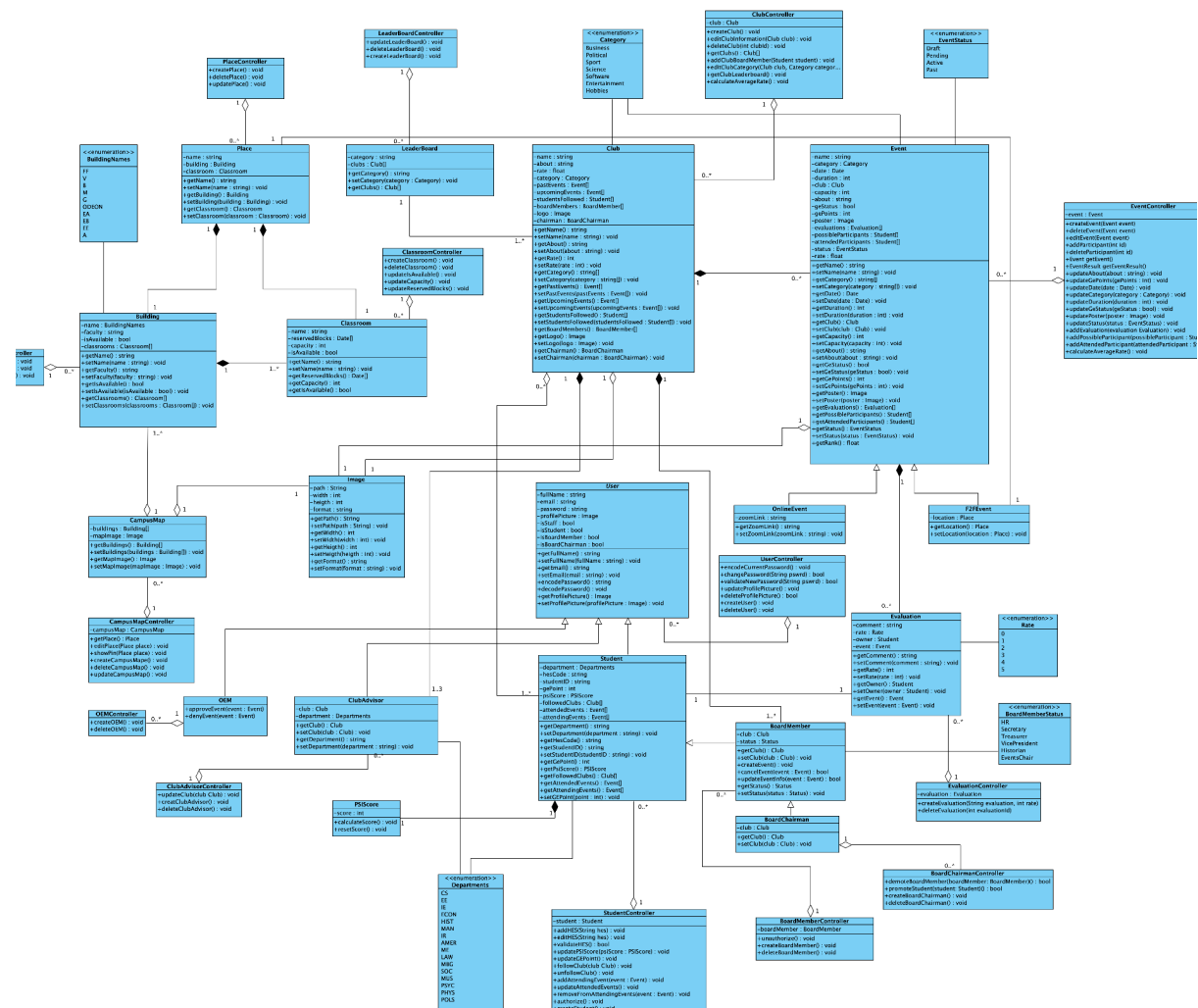
### 2.5.2.3.2 State of a Student



**Figure. 11:** State Diagram of a Student object

The initial state of a student is student. When the student is promoted by Admin, the new state of a student becomes board chairman. When his/her term expires or he/she is demoted by Admin, the old state which is student is turned back to. At the student state, when the student is promoted by the Board Chairman, the state goes to the board member state. When his/her term expires or he/she is demoted by the Board Chairman, it turns back to the old state which is student. Finally, when she/he graduates, each of these states goes directly to the final state.

### 2.5.3. Object and class model



**Figure 12:** The class diagram of the application  
(For the high resolution image: <https://ibb.co/Fh2DTjx>)

The class diagram of the web-based application is given above. The application's class diagram consists of 13 classes. This class diagram shows the relations between base classes.

**User Class:** This class is an abstract class and it is a superclass. Its subclasses are Student class and ClubAdvisor class.

**Student Class:** This class is a subclass of the User class and superclass of the Board Member class. 0 to many Event classes consists of 1 to many Student classes and 0 to many Club classes consists of 1 to many Student classes. Also, the Student class has associations with PSI Score and Evaluation classes.

**ClubAdvisor Class:** This class is a subclass of the User class. One Club class consists of 1 to 3 ClubAdvisor classes. Club Advisors can view many information about clubs and events.

**BoardMember Class:** This class is a subclass of the Student class. One Club class consists of 1 to many BoardMember classes. BoardMember classes can create many Events.

**BoardChairman Class:** This class is a subclass of BoardMember class. There is a one-to-one association between BoardChairman class and Club class. BoardChairman can promote Student to BoardMember and demote BoardMember to Student.

**LeaderBoard Class:** This club has one-to-many association with Club class.

**OEM Class:** This class is a subclass of the User class. OEM is responsible for approving Clubs' events.

**PSIScore Class:** This class has an association with the Student class. This class shows the PSI score of a student.

**ImageClass:** This class is for images. This class has associations with CampusMap, User and Event classes. An Event and a User class may have one image.

**Club Class:** This class consists of 0 to many Event classes, 1 to many BoardMember and Student classes and 1 to 3 ClubAdvisor classes.

**Event Class:** This class has associations with Place, Club, Image, Student and Evaluation classes. This class is for the information of the events. 0 to many Event classes may have 1 to many Student classes, one Place class and one Club class. One Event class may have one Image class and 0 to many Evaluation classes.

**OnlineEvent Class:** This class is a subclass of Event class. It contains necessary information about online events such as the Zoom link.

**F2FEvent Class:** This class is a subclass of Event class. There is a one-to-one relationship between Place class and F2FEvent class.

**Evaluation Class:** This class is for evaluations. It has associations with Event and Student classes. One Student class can have 1 Evaluation class and one Event class may have 0 to many Evaluation classes.

**CampusMap Class:** This class has associations with Building and Image classes. This class is for the campus map. One Campus map class has exactly one Image class and 1 to many Building classes.

**Place Class:** This class is a superclass. Its subclasses are Building and Classroom classes. Also, it has association with Event class. This class is for the place information of the events. One Place class may have 0 to many Event classes, 1 to many Building and Classroom classes.

**Building Class:** This class is a subclass of the Place class. It has associations with Classroom, Place and CampusMap classes. One Building class may have 1 to many

Classroom classes and one Place and CampusMap class consists of 1 to many Building classes. This class is for the building information of an event.

**Classroom Class:** This class is a subclass of the Place class. Also, it has an association with the Building class. One Building and Place class consists of 1 to many Classroom classes. This class is for the classroom information of an event.

**UserController Class:** This class is responsible for the operations of the User object such as delete, update and create.

**StudentController Class:** This class is responsible for the operations of the Student object such as delete, update and create.

**ClubAdvisorController Class:** This class is responsible for the operations of the ClubAdvisor object such as delete, update and create.

**BoardMemberController Class:** This class is responsible for the operations of the BoardMember object such as delete, update and create.

**BoardChairmanController Class:** This class is responsible for the operations of the BoardChairman object such as delete, update and create.

**ClassroomController Class:** This class is responsible for the operations of the Classroom object such as delete, update and create.

**BuildingController Class:** This class is responsible for the operations of the Building object such as delete, update and create.

**PlaceController Class:** This class is responsible for the operations of the user Place such as delete, update and create.

**EventController Class:** This class is responsible for the operations of the user Event such as delete, update and create.

**EvaluationController Class:** This class is responsible for the operations of the Evaluation object such as delete, update and create.

**ClubController Class:** This class is responsible for the operations of the ClubController object such as delete, update and create.

**LeaderBoardController Class:** This class is responsible for the operations of the LeaderBoard object such as delete, update and create.

**OEMController Class:** This class is responsible for the operations of the OEM object such as delete, update and create.

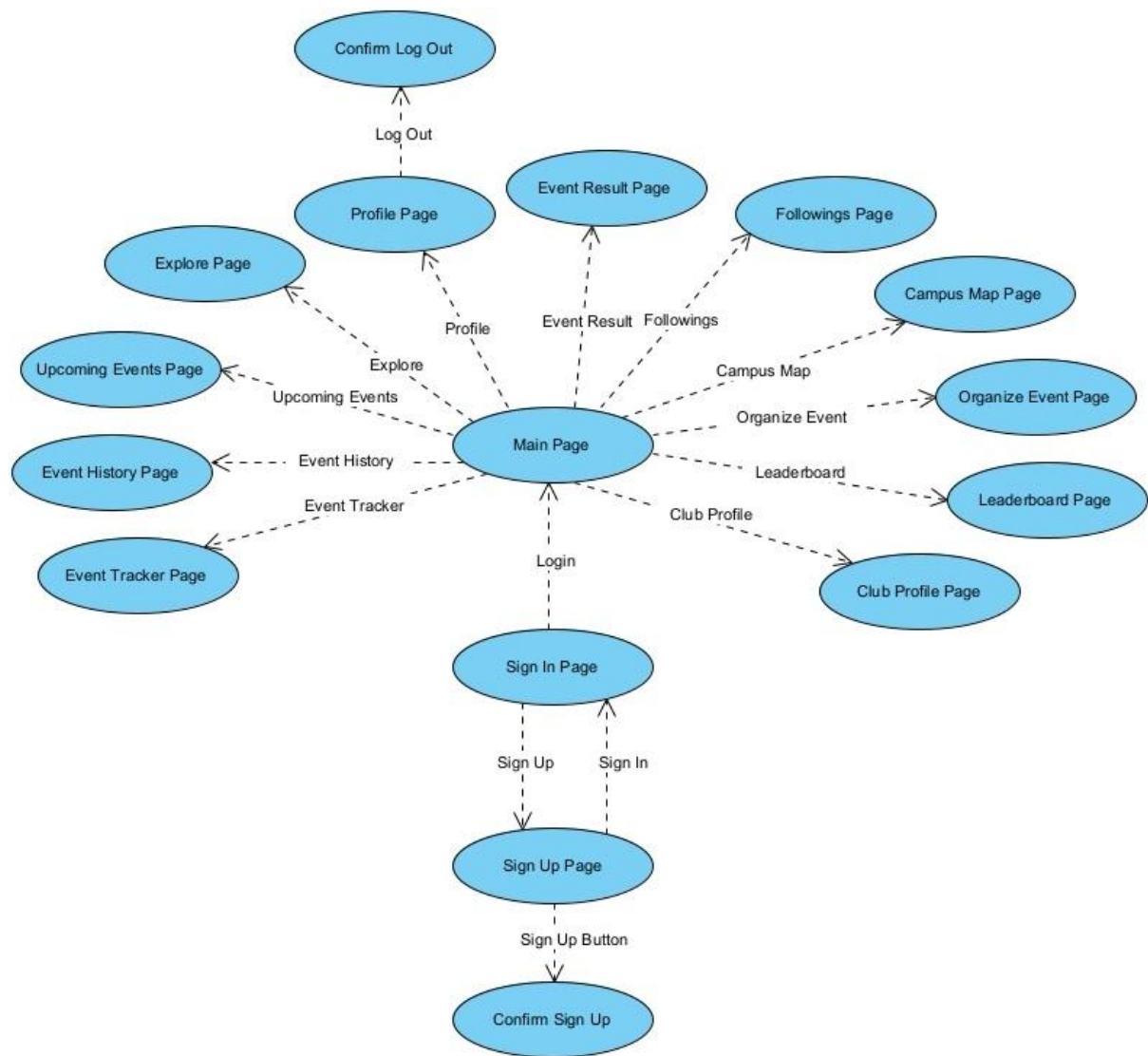
## 2.5.4 User interface - navigational paths and screen mock-ups

### 2.5.4.1 Navigational Path

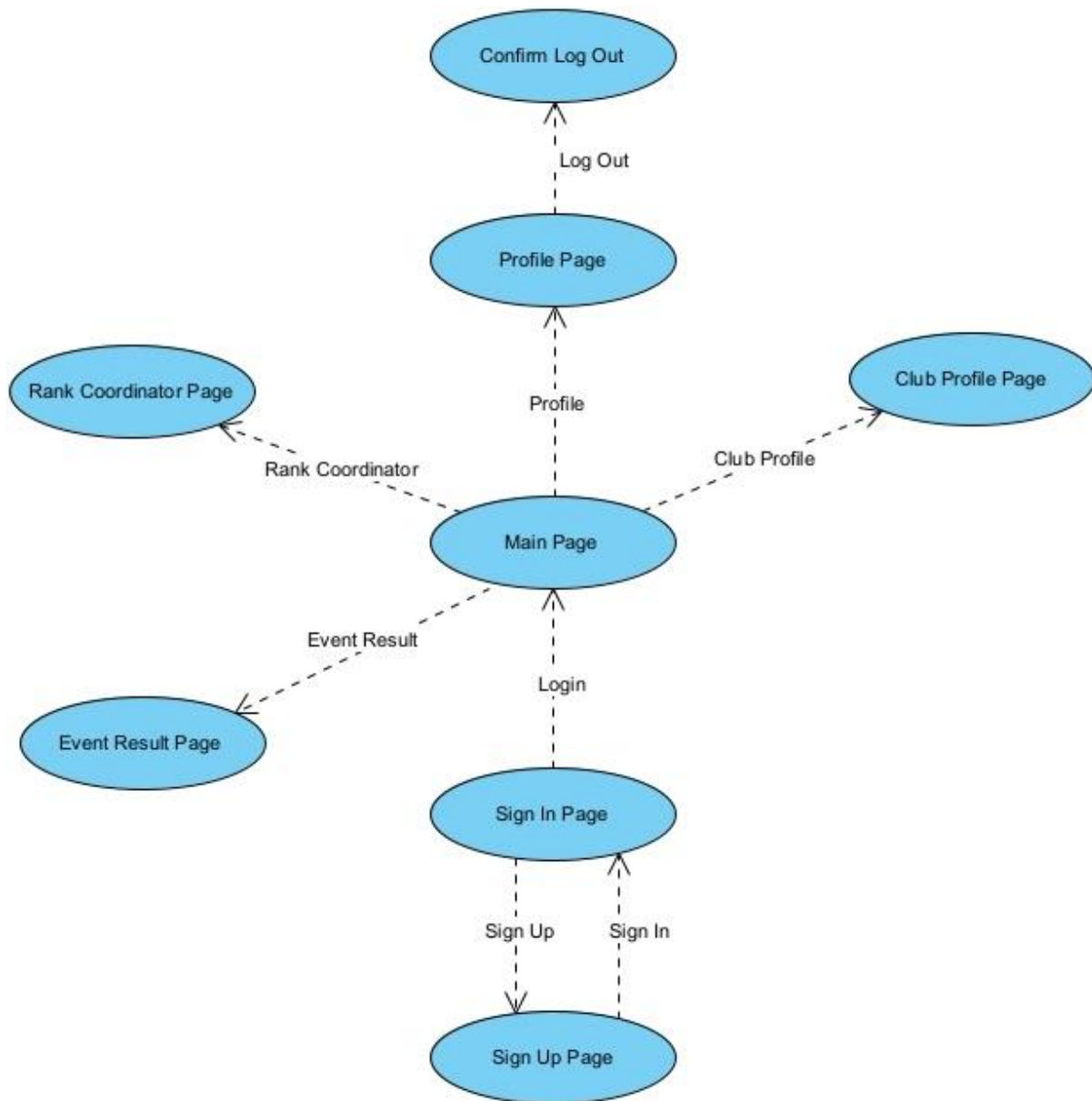


**Figure. 13:** The navigational path of the user interface screens for Student






**Figure 14:** The navigational path of the user interface screens for Board Member



**Figure. 15:** The navigational path of the user interface screens for Club Advisor

### 2.5.4.2 Screen Mock-ups



A screen mock-up for a 'Sign In' form. At the top, there are two tabs: 'Sign In' (selected) and 'Sign Up'. The form is centered and contains the following elements: an 'E-mail' label above a text input field containing 'tony.stark@ug.bilkent.edu.tr'; a 'Password' label above a text input field containing '\*\*\*\*\*'; a 'Forgot Password?' link below the password field; and a green 'Login' button at the bottom.

**Figure 16:** Sign In

This is the welcome screen that the user sees when he/she enters the website. If the user has an account, the user can successfully log in by pressing the login button after entering his email address and password correctly.



A screen mock-up for a 'Sign Up' form. At the top, there are two tabs: 'Sign In' and 'Sign Up' (selected). The form is centered and contains the following elements: a 'Name' label above a text input field containing 'Tony Stark'; an 'E-mail' label above a text input field containing 'tony.stark@ug.bilkent.edu.tr'; an 'ID' label above a text input field containing '21000000'; a 'Password' label above a text input field containing '\*\*\*\*\*'; a 'Department' label above a text input field containing 'Computer Science'; and a green 'Sign Up' button at the bottom.

**Figure 17:** Sign Up

If the student does not have an account he/she can create a new account by filling in the required information such as email, student ID located in the Sign Up screen. After

entering the information and clicking the Sign Up button, the accuracy of the information will be checked. If the entered information matches the student information in the Bilkent University database, a confirmation email will be sent to the entered email address.

Club Advisor accounts will be created by the system and account's information will be provided to them. Therefore, the Club Advisor does not need to create an account.



**Figure 18:** Main Page for Students

If the Student logs in successfully, this screen appears. The Students can see their profile picture, full name and PSI score at the upper left side of the screen. This screen consists of 8 different tabs.

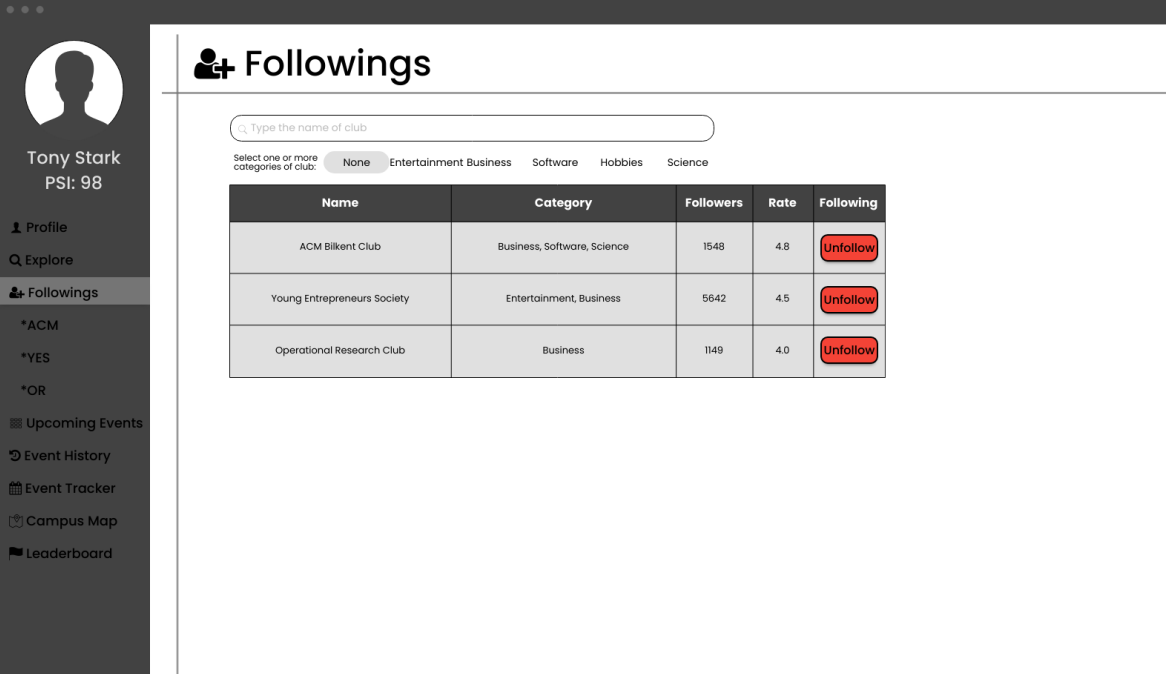
**Figure 19:** Profile Page for Students

In the Profile Tab Student can view him/her account information. Student can also update him/her account information by pressing Update Profile button. Since Hayat Eve Sığar (HES) code information is mandatory when entering club events, this field is required to be filled. Student can link to HES Code to him/her account by clicking the Submit button. Student can also change his/her HES Code. If the Student wishes, he/she can successfully update him/her password by entering the information in the change password panel. Lastly, Student can log out by clicking the red button at the top right of the screen.

Name	Category	Followers	Rate	Following
ACM Bilkent Club	Business, Software, Science	1548	4.8	Unfollow
Management and Economics Society	Business	6889	4.6	Follow
Astronomy Society	Science, Hobbies	1276	4.7	Follow
Young Entrepreneurs Society	Entertainment, Business	5642	4.5	Unfollow
E-Sports Society	Entertainment, Hobbies	2036	4.2	Follow
Science Fiction and Fantasy Society	Entertainment, Hobbies	891	4.3	Follow
Operational Research Club	Business	1149	4.0	Unfollow

**Figure 20:** Explore

The Student can view all Bilkent University student clubs with information of name, category, followers and rate. The student can filter student clubs according to their categories. The Student can follow and unfollow any student club he/she wants by clicking Follow or Unfollow buttons.



**Followings**

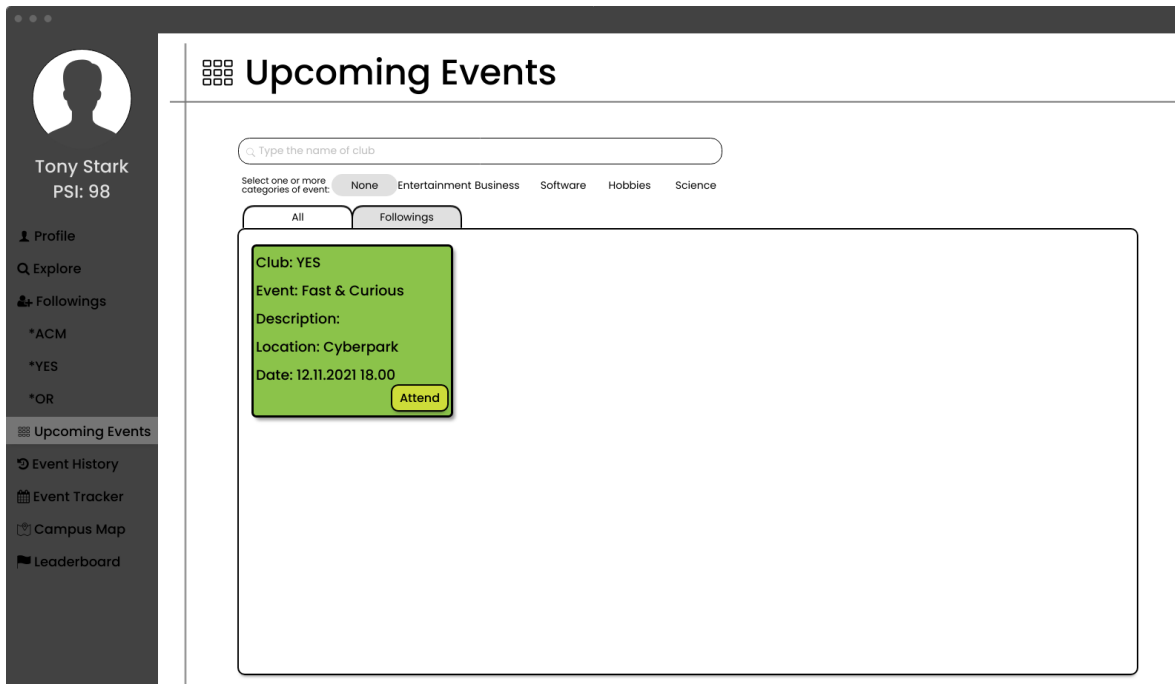
Type the name of club

Select one or more categories of club:

Name	Category	Followers	Rate	Following
ACM Bilkent Club	Business, Software, Science	1548	4.8	<input type="button" value="Unfollow"/>
Young Entrepreneurs Society	Entertainment, Business	5642	4.5	<input type="button" value="Unfollow"/>
Operational Research Club	Business	1149	4.0	<input type="button" value="Unfollow"/>

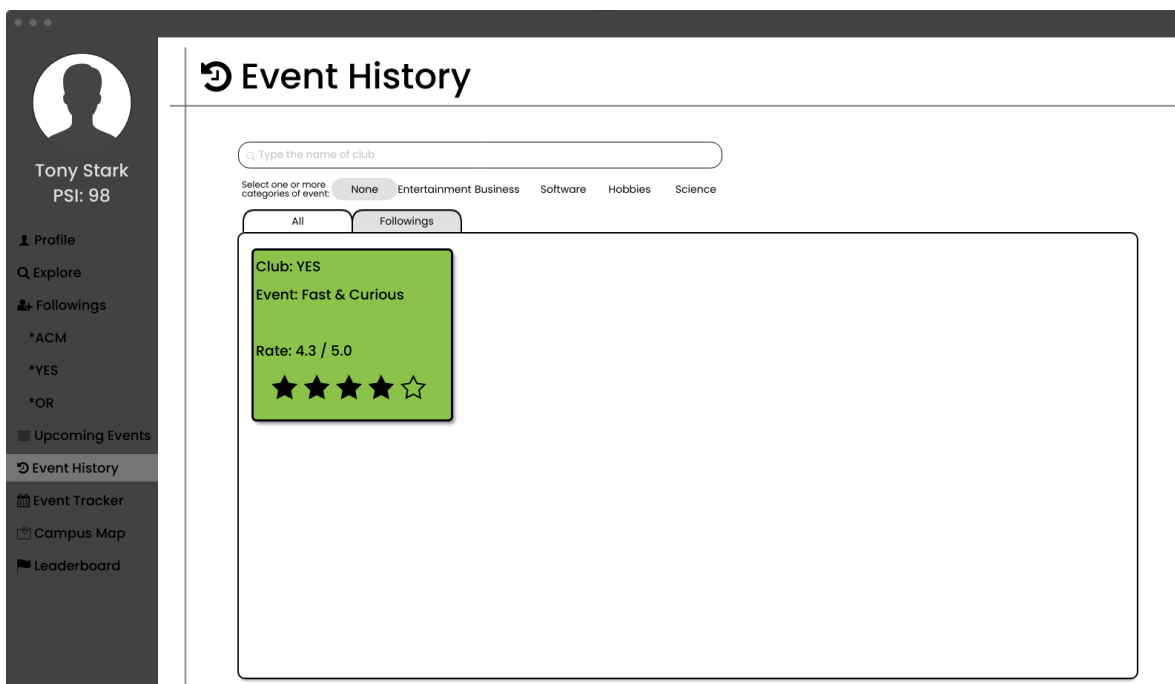
**Figure 21:** Followings

The Student can view just student clubs that he/she follows. The Student can also filter the clubs according to their categories. The Student can unfollow the student club he/she follows by clicking the Unfollow button.



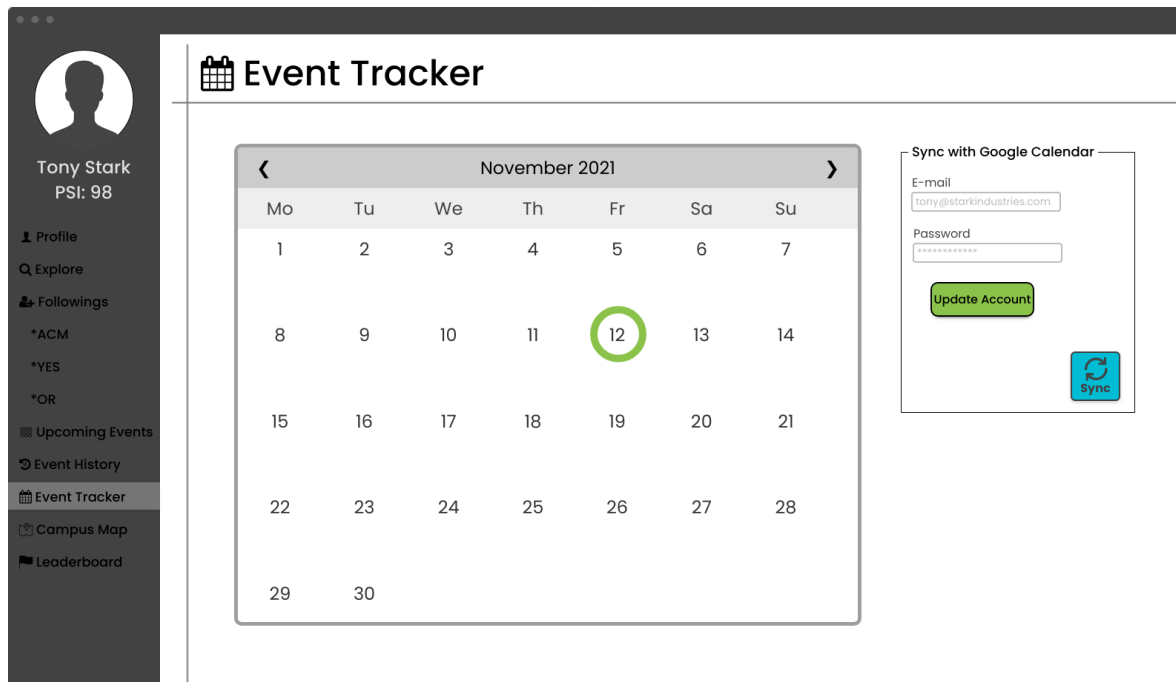
**Figure 22:** Upcoming Events

By default, the Student can view different events organized by different student clubs. If the Student clicks the Followings button he/she can view the only events organized by student clubs that he/she follows. The Student can announce their participation in the event by clicking the Attend button located on the event card. If the Student decides not to attend, he/she can change his/her status from will attend to will not attend.



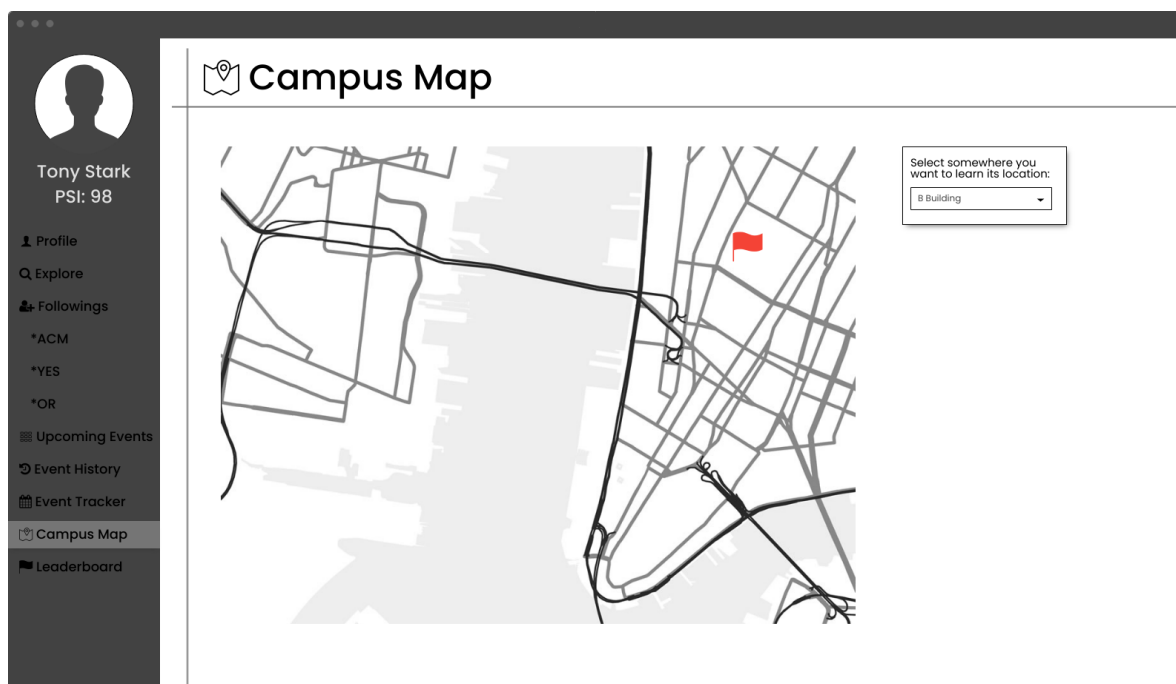
**Figure 23:** Event History

The Student can view the events he/she attended. Each event is represented as a card and this card contains the name of the student club, the name of the event and the rate given by students, respectively.



**Figure 24:** Event Tracker

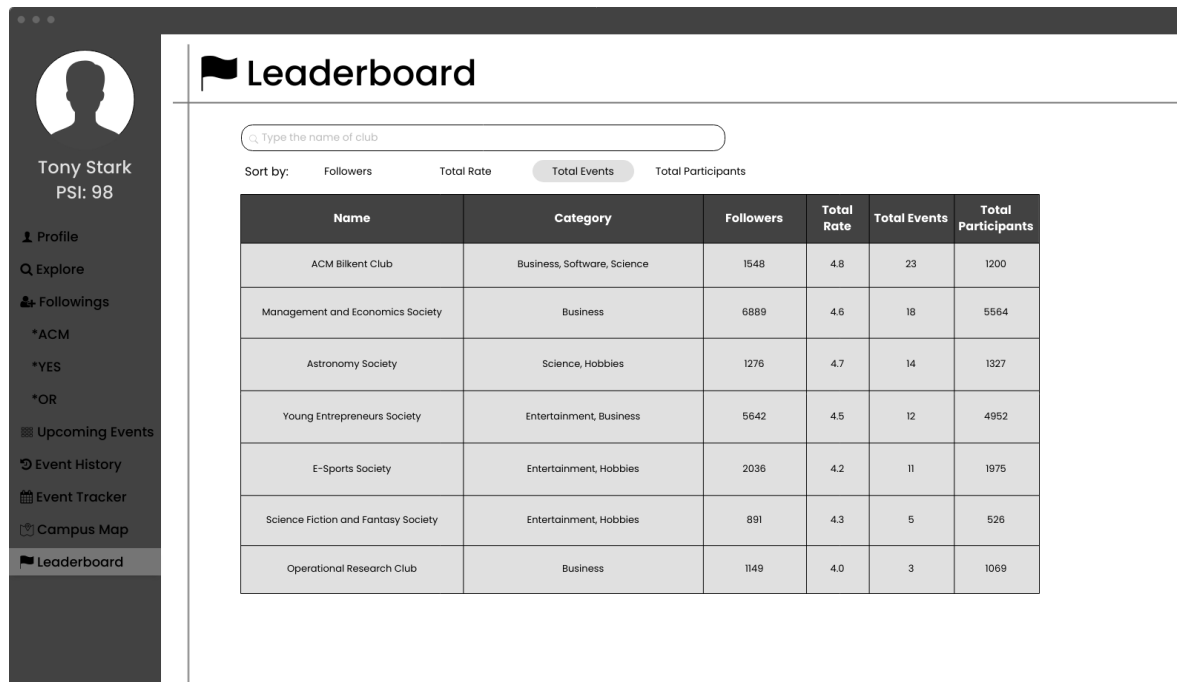
The Student can view the events to be attended as marked on the calendar. The Student can also synchronize this calendar with Google Calendar via filling the necessary information.



**Figure 25:** Campus Map

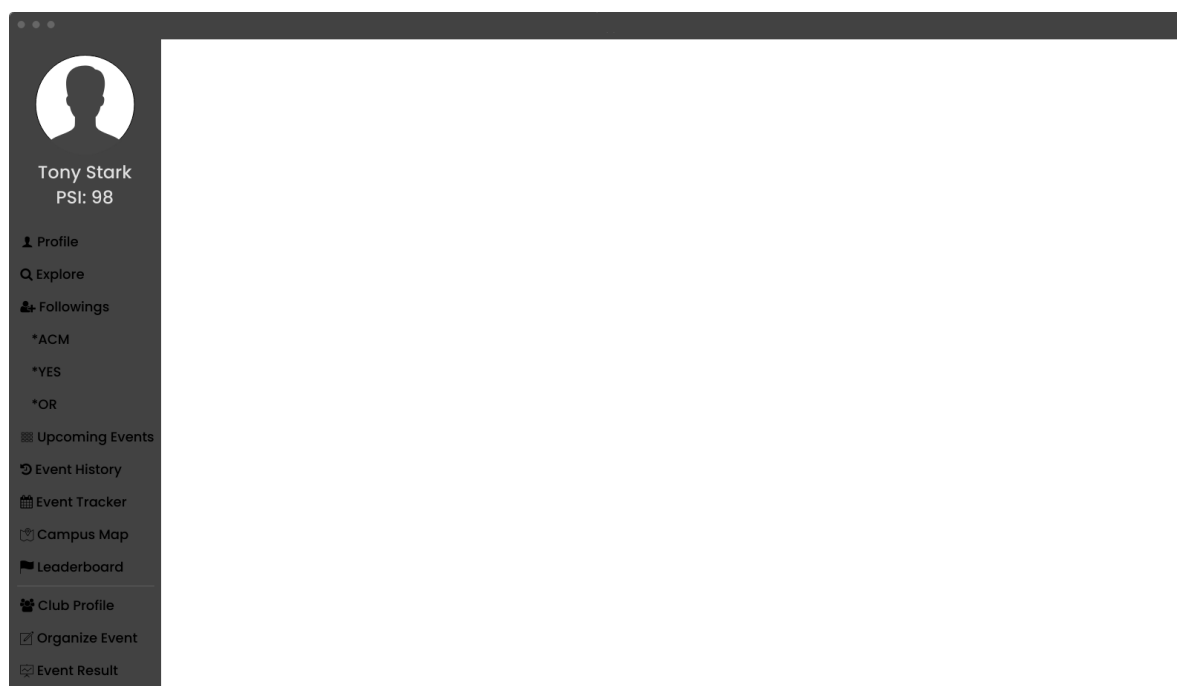


The Student can find the building located in Bilkent University by selecting from the drop-down menu at the top right of the page.



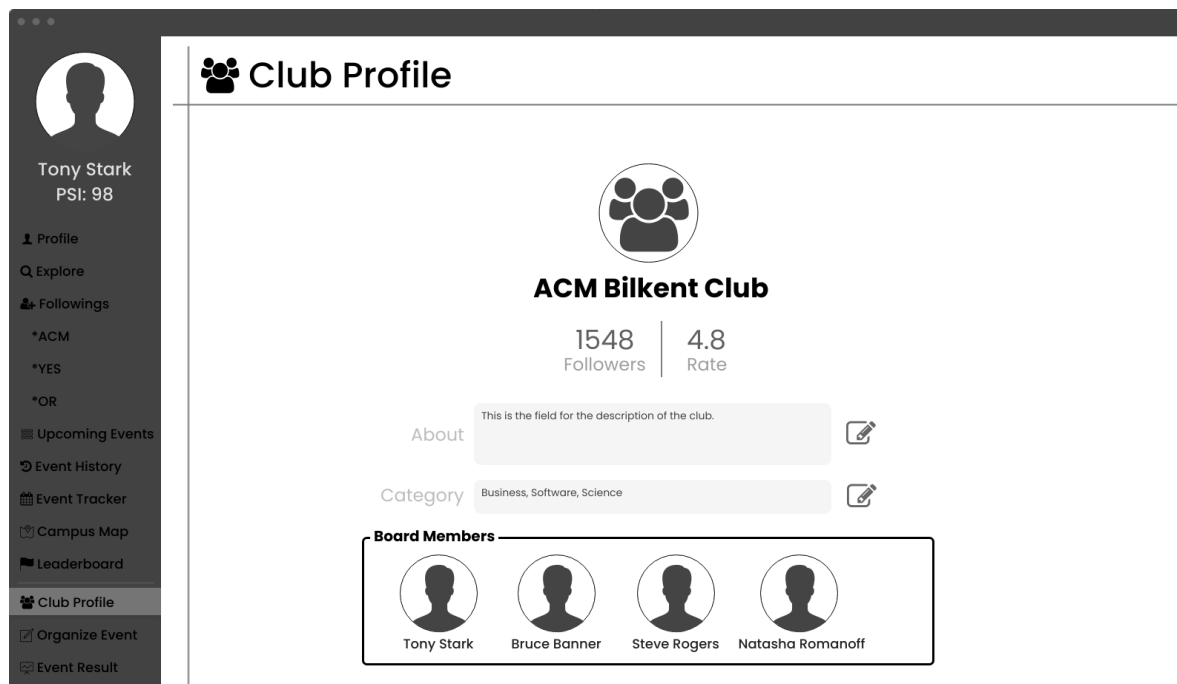
**Figure 26:** Leaderboard

The Student can view the list of student clubs with their information according to various criteria such as the number of followers, total rate, total events and total participants. The Student can select the criteria from the menu above the leaderboard.



**Figure 27:** Main Page for Board Member

Unlike the Student, the Board Member can view three additional menus. These are Club Profile, Organize Event and Event Result, respectively.



**Figure 28:** Club Profile

The Board Member can view and edit the information of the student club he/she is a board member of. The Board Member can edit the About and Category section. The Board Member can add new students or remove existing students in the Board Members section.

- Profile
- Explore
- Followings
- \*ACM
- \*YES
- \*OR
- Upcoming Events
- Event History
- Event Tracker
- Campus Map
- Leaderboard
- Club Profile
- Organize Event
- Event Result

## Organize Event

Event Name

Description

Location

Date

Create Event

### Your Club's Upcoming Events

Event Name	Description	Location	Date
Ankara Start-up Zirvesi	Business	B Building	26.11.2021

**Figure 29:** Organize Event

The Board Member can create an event by filling the necessary information such as Event Name, Description, Location and Date. Events added by the Board Member are displayed as a list and the Board Member can edit the information of the event or cancel the event.

- Profile
- Explore
- Followings
- \*ACM
- \*YES
- \*OR
- Upcoming Events
- Event History
- Event Tracker
- Campus Map
- Leaderboard
- Club Profile
- Organize Event
- Event Result

## Event Result

### Your Club's Past Events

Event Name	Description	Location	Date	Result
Introductory Meeting	Entertainment	Mayfest Area	16.09.2021	Show
Fast & Curious	Business	Cyberpark	13.10.2021	Show

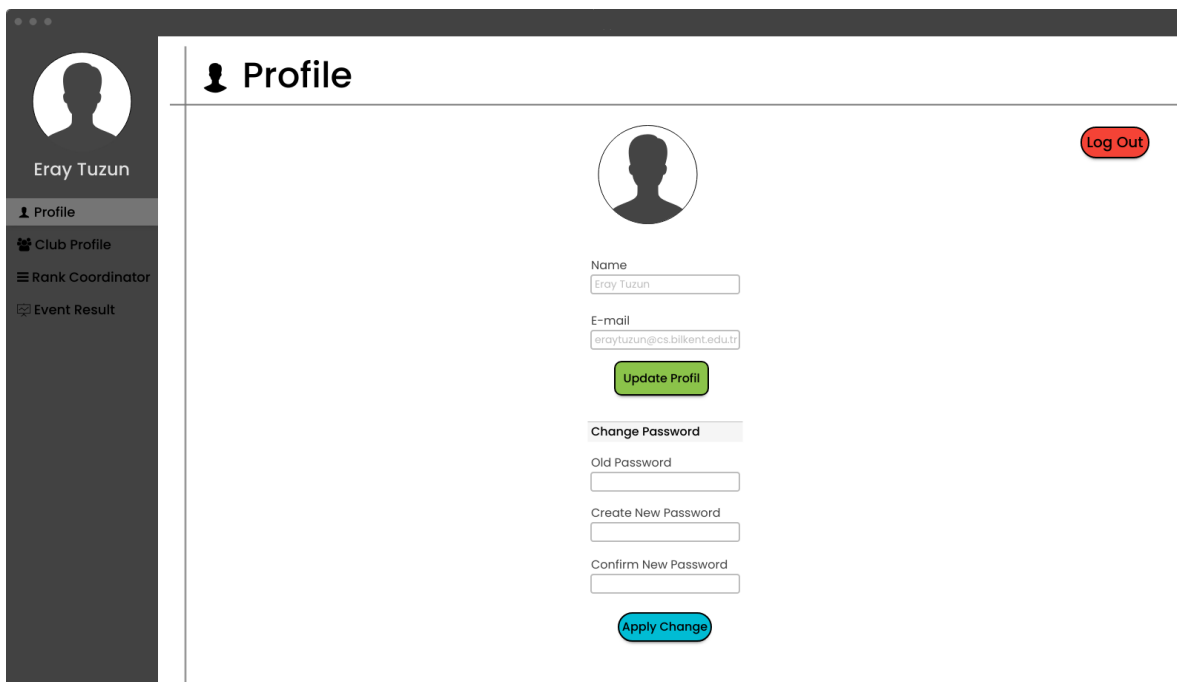
**Figure 30:** Event Result

In this screen, the Board Member can display all completed events as a list. The Club Board Member can view the details of the past events by clicking the Show button.



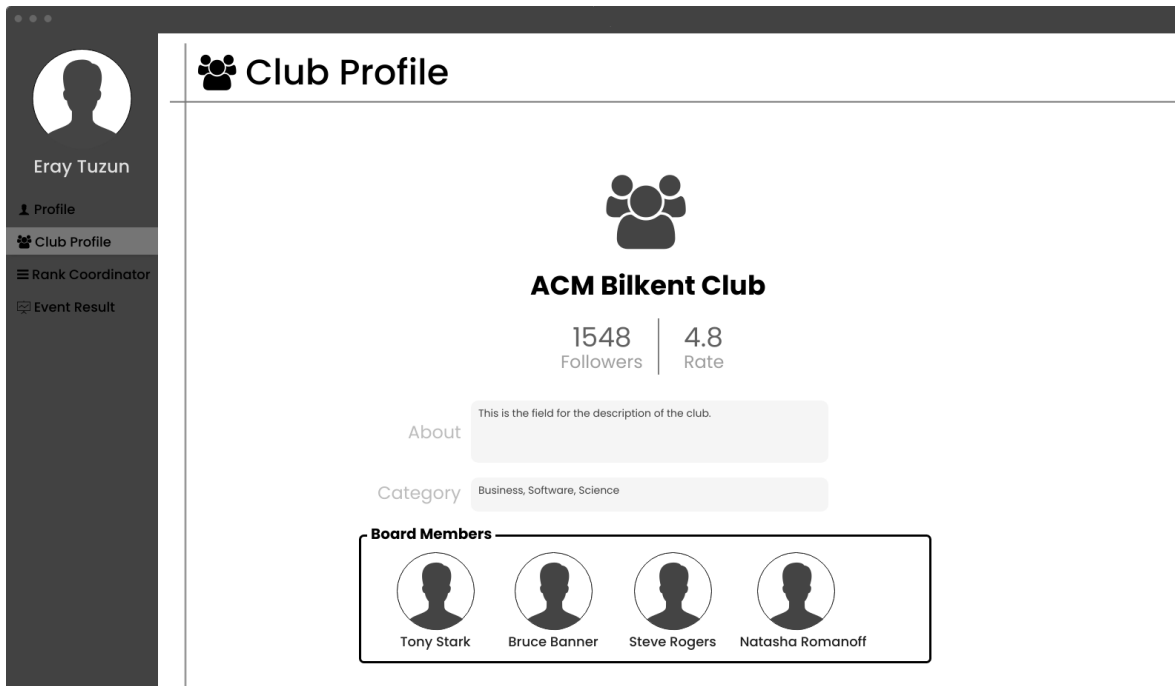
**Figure 31:** Main Page for Club Advisor

The Club Advisor can view 4 sub menus. These are Profile, Club Profile, Rank Board Member and Event Result, respectively.



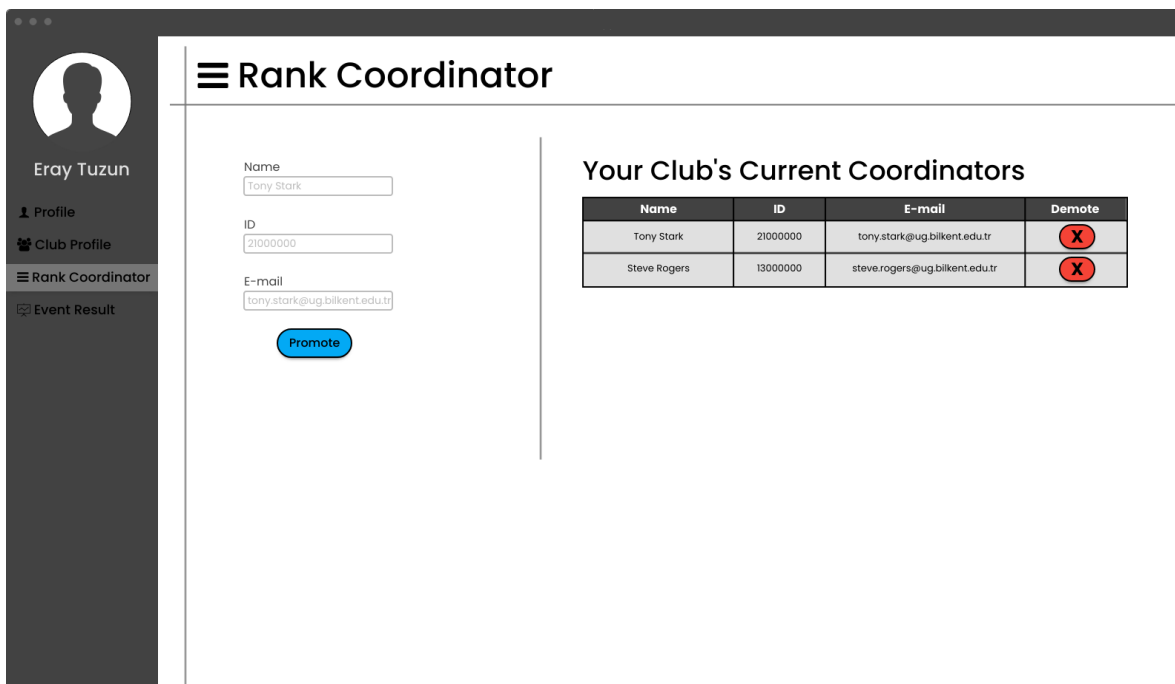
**Figure 32:** Profile Page for Club Advisor

The Club Advisor can view and edit their account's information. Unlike Student and Board Member, Club Advisor can change his/her email address. Lastly, Student can log out by clicking the red button at the top right of the screen.



**Figure 33:** Club Profile for Club Advisor

In this page the Club Advisor can display the information of the student club he/she is advisor of.



**Figure 34:** Rank Board Member

The Club Advisor can change the status of Student as Board Member by filling the information of student correctly. The Club Advisor can demote the Board Members by clicking the Demote button.

The screenshot displays a web application interface for a Club Advisor. On the left is a dark sidebar with a user profile icon and the name 'Eray Tuzun'. Below the name are four navigation links: 'Profile', 'Club Profile', 'Rank Coordinator', and 'Event Result' (which is highlighted). The main content area has a header 'Event Result' with a small icon. Below the header is the section 'Your Club's Past Events' containing a table with two rows of event data. Each row has a 'Show' button in the 'Result' column.

Event Name	Description	Location	Date	Result
Introductory Meeting	Entertainment	Mayfest Area	16.09.2021	Show
Fast & Curious	Business	Cyberpark	13.10.2021	Show

**Figure 35:** Event Result

Like the Board Member, Club Advisor can display the details of the completed events of the student club he/she is advisor of by clicking the Show button.

## **3. Improvement Summary**

### **3.1 General**

- Figure numbers have been added to each figure.

### **3.2 Overview**

- New actors have been added.
- ClubSupervisor actor has been changed to ClubAdvisor.
- ClubCoordinator actor has been changed to BoardMember.

### **3.3 Non-Functional Requirements**

- Scalability has been modified.

### **3.4 Use Case Diagrams**

- In the textual case, some of the flow of events have been changed by removing the UI stuff like buttons.
- In the use case, some extends and includes operations have been changed.
- New actors such as BoardMember, BoardChairmen, and OEM have been added.
- New use cases such as EvaluatePendingEvent have been added.
- Some use cases have been removed because their scope is so small.

### **3.5 Sequence Diagrams**

- Sequence diagrams have been completely reworked.
- New sequence diagrams have been added.

### **3.6 Activity Diagrams**

- Activity diagrams are updated.
- Sign in and sign up activity diagram is removed.
- Three new activity diagrams are added.
- Activity diagrams were written according to the user. However, this was found to be wrong and it was written according to the system in its new form.

### **3.7 State Diagrams**

- State diagram of event object and student object has been redrawn.
- Textual explanation of state diagrams of event object and student object has been added.

## 3.8 Class Diagrams

- Multiplicities have been reviewed.
- Controller classes have been added.
- Methods of classes have been reviewed.
- Attributes of classes have been reviewed.
- Enum classes have been added.



## 4. Glossary & References

### 4.1 Glossary

**PSI:** This term is the abbreviation of Personal Sociability Intelligence. This score indicates how social a person is. This score will be calculated by an algorithm determined by the system. The events that the user participates in, the clubs the user follows, the clubs the user is the board member of will affect the calculation of this score.

### 4.2 References

[1] "Web Content Accessibility Guidelines (WCAG) 2.1". <https://www.w3.org/TR/WCAG21/>. [Accessed: Oct 23, 2021].

[2] "Django". <https://www.djangoproject.com/>. [Accessed: Oct 23, 2021].

[3] "The Progressive JavaScript Framework". <https://vuejs.org/>. [Accessed: Oct 23, 2021].

[4] "PostgreSQL: The World's Most Advanced Open Source Relational Database". <https://www.postgresql.org/>. [Accessed: Oct 23, 2021].