Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Course Project

YACM: Yet Another Club Manager

Analysis Report

Ali Emre Aydoğmuş, Cemal Faruk Güney, Deniz Berkant Demirörs, Enis Özer, Yekta Seçkin Satır

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

Progress Report
December 18, 2021

Contents

Introduction

This is a project that can help club executives run their clubs, students find interesting club activities and faculty advisors see what their club is up to. The project will consist of two parts. One of them is a website. In the website student club executives will be able to create posts in which they can announce events and create polls to share their new activities and use chat to communicate with club members. The students that are not in the club boards will have the option to follow some clubs and see their posts on their feed, they will be able to comment on these posts and receive answers and if they are a member of that club they will have access to the club chat. Faculty advisors will be able to see their club's activities from the past few months and they will have the option to veto any upcoming event if they see it inappropriate. The mobile application part of the project will have all the features that the web application has and some more. In the mobile application there will be a QR scanner which students will use to scan the assigned QR code of the activity they have attended. With these features we are trying to make life in the university easier for all users.

Functional Requirements

Sign Up

The users need accounts to use most features of the system such as creating posts, participating in polls, subscribing to student clubs. When they have opened the application or entered the website for the first time they are expected to create an account. For creating the account the students and faculty advisors need to provide their name, surname, Bilkent email address, their Bilkent ID number, and a valid password. After the process is done the account will be created in the database with the given information. Users will use the same account for the website and the mobile application.

Login

After they have created their accounts users will have to log in. If the login is successful, login credentials to the browser will be saved and it will make the app skip the login activity using the saved information for future entries. The login pop-up will have two input text boxes: One for email and the other for password. Users are expected to fill these boxes with the email and password they have used for creating their accounts. If the email and password pair matches with an account in the database and if the email is verified, the login will be successful and the user will proceed to language selection. In language selection, there will be two options: English and Turkish. Users will select one language on their first login and this preference will be saved to their account until they change it on the settings part.

Posts

Posts in this application are created by club board members. A post can be used for a poll or an event announcement. Event announcement posts can contain at most 8 images as well as text. Users will be able to access posts of their clubs from their feed, from the club's profile, from the explore window, or if they have the URL to the post they can see it directly. Every user with an account has the opportunity to comment on a post. The comments will be visible to anyone who can access the post and they will remain under the post content. There will be a button that opens a menu on the top-right of the post. This menu will contain buttons for editing (manager), deleting (manager) or vetoing (advisor) the post, and copying the link of a post. The users will pin posts to access them easily on the pinned posts tab. Students that can see a club's post are expected to determine if they will participate in the event or not. If they think that they will, they should declare their participation by clicking a button. This feature is important for events with limited seating. If there is no room for another student in the event the student will be notified. A student list with the names of the students that declared participation will be sent to the club board members.

Students that are a member of a club can vote on that club's polls. Club board members can create polls to get the opinion of other students in the club on certain issues. The polls will be anonymous. There can be at most 5 options in a poll. A student who is not authorized to vote on the poll will still be able to see the post that contains the poll.

Posts contain information about the start date of the event and the number of available seats if it is limited. One day after the start date of the event posts will be deleted from user feeds.

If the viewer is not enrolled in the club which has created the post there will be a subscribe button in the post. Students will be able to subscribe to clubs using this button easily.

Subscriptions

All students with accounts will be able to subscribe to any club of their interest. Subscribed students will receive notifications from the new events that are organized by the clubs they are subscribed to. Subscribe button will be available on club profiles and posts from unsubscribed clubs.

Feed

Feed is basically a list of posts that are available for the user. The main page of the application will have the feed with the posts from subscribed clubs for the active student. After the student sees all posts in his/her feed the application will recommend them new posts from clubs that are similar to their subscribed clubs. For example, if a student is a member of ACM and has seen all of their posts in his/her feed the next thing in their feed can be the newest post of IEEE. Application will determine the similarity of clubs using the tags that are assigned to clubs by us developers.

The user will also be able to access the feed of a certain club from the club profile. This feed will be the same with the main page feed however, it will only contain posts from one club. If the student sees every post in this feed, no more posts will be recommended by the application.

User Profiles

Every user will have a profile page. The pop up will have information about the user name. Students have the option to add a profile picture to represent themselves. If they do not want to upload a profile picture they will have a default picture.

Profile page of a user will also display the QR code generated uniquely for that user. This QR code will be used when entering a club event to identify people who take GE250/251 rather than using a conventional card scanner.

Users can also access the clubs that they are subscribed to from their profile pages. They will have the ability to go to the club profile of a certain subscribed club by selecting the club from the list of subscribed clubs.

Additionally, users have the opportunity to edit their profile. They can change their profile pictures and passwords using this feature. For editing the profile they should press a button on their profile page. To change their passwords users have to enter their old password and type a valid new password. After this process the user will have to enter the application with their new password.

Club Profiles

Each club that is registered to the system will have a profile page. The club profile page will consist of club name, club profile picture, club description, club specific feed and club chat. The club feed and club chat will not be visible at the same time. The page opens with the club feed open initially, the browsing user has to select change from the navbar to see the club chat.

In the club feed photos or other content, such as polls, of the club's posts will be displayed in a grid. The photos will be clickable and if clicked it will take the user to the post. If the user that is viewing the club profile is not a member of the club they will not be able to see the club chat.

In the club profile page there will be a “Manage Club” button that is only visible to authorized club board members. Using this button the board members will be able to create or delete posts and kick club members that are not board members.

If the user clicks the club profile picture they will see the list of students that are subscribed to the club. In the list students will be represented with their profile pictures and names.

Club Chats

Club chats are accessible from club profile pages by club members. All of the club members are also members of the club chat. A student can only see the messages sent after they have joined the club. Anyone that can see the messages can send new messages. The user interface for club chat contains the profile picture of the message sender in the left-hand side and the content of the message in the right-hand side. Content of the message can be text, a link to a post or an image.

For club board members there will be a mute button on the right side of every message. Authorized board members will be able to mute club members using this button if they are causing trouble in the group chat.

Explore

There will be an explore window in the application. In this window users will be able to search clubs with text inputs. Users can click on the search results to enter the corresponding club profile.

In the explore window under the search bar recent posts will be displayed in a grid. These posts can be created by any club; it does not matter if the user is subscribed to them or not. They will be sorted with respect to the date of their creation. If the user clicks on one of the posts in the grid they will be taken to the relevant post page.

On the right side of the window there will be a list containing suggested clubs for the user. Application will suggest these clubs depending on the user's subscribed clubs' tags. If the user is subscribed to clubs with similar tags to 'Club A', it will be a suggestion on the list.

QR Scanner

There will be a QR scanner in the mobile application. The feature will be used by club board members at the beginning of an event to check who is participating in the event. At the start of an event the students will ideally open the application and show their QR codes in the profile page. Then, the list of participants will be sent to the club board via email after the event is done. This information will be used by the club board to determine which members are attending the events and which members are inactive.

Notifications

The users will automatically receive notifications whenever an event that they have declared participation is in a certain time or a poll that they have voted is concluded. The students will be able to see received notifications in the notifications window where the notifications will be listed. Each notification in the notifications window will take the user to the corresponding post.

Non-Functional Requirements

Usability:

The system must have a usable interface. This is going to be achieved by designing the UI in an intuitive and common way. Large social media platforms can be used as a guide. The user should feel familiar in the first encounter.

Performance:

The users should have smooth experience with the platform. The changes in the database should reflect to necessary places almost instantly. For example if a club manager changes the time of an event, members who are going to join that event should be notified immediately.

Security:

All users who enter the system must be either students or instructors of Bilkent University. This is achieved by forcing the students to sign in via their Bilkent mail address and giving instructors their accounts directly. If a student suspects that his/her account has been stolen he/she can change or regain the account using his/her Bilkent mail. There is no anonymity other than visitors who cannot do anything other than viewing the events.

Constraints

- An Object Oriented Programming Language should be used in the implementation of the project.
- Project has to be a web-app.

System Models

Use Case Model (Click here to open the diagram in a higher quality)

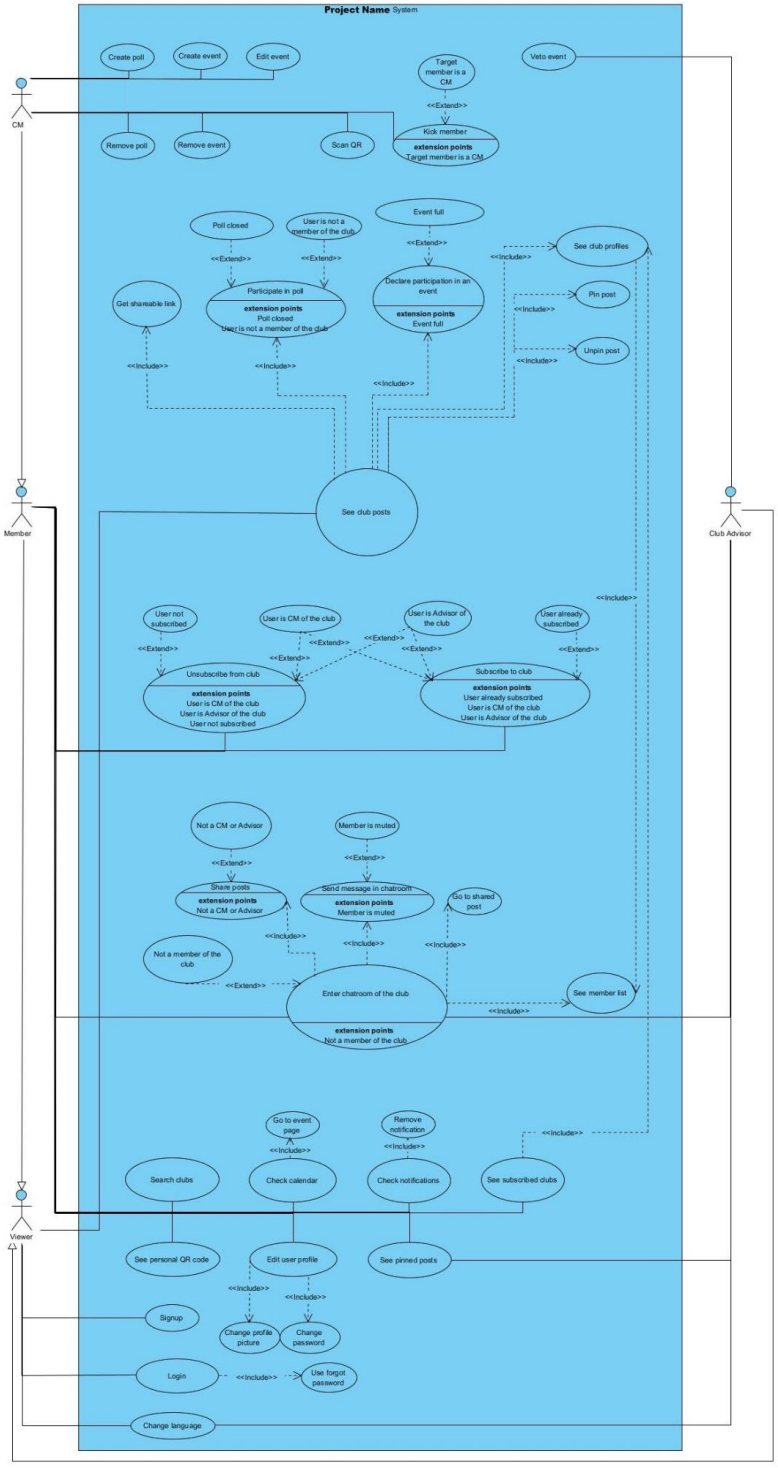

Figure 1: General Use-Case Diagram

Below are the textual descriptions of each use case, moving top-down according to the diagram.

Use case name: Create poll
Participating actors: CM
Flow of events:
1) User adds a question and possible answers.
2) User saves the poll.
Entry condition: User must be CM of that club and logged in.
Exit condition: A new poll is created and all members of that club are notified.

Use case name: Create event
Participating actors: CM
Flow of events:
1) User specifies the event place and time.
2) User adds a title and description.
3) User sees if the event overlaps with any other event whether from this club or from other clubs.
4) User may add a picture or pictures describing that event.
5) User may add GE250/251 info: telling if any chances of getting points are present.
6) User saves the event.
Entry condition: User must be CM of that club and logged in.
Exit condition: A new event is created and all members of that club are notified.

Use case name: Edit event
Participating actors: CM
Flow of events:
1) User edits one or more properties of an event (photos, title, description, time, GE250/251 info).
2) User saves the changes.
Entry condition: User must be CM of that club and logged in.
Exit condition: The event has been edited and all members of that club are notified.

Use case name: Remove poll
Participating actors: CM
Flow of events:
1) User removes the poll
Entry condition: User must be CM of that club.
Exit condition: The poll has been removed.

Use case name: Remove event
Participating actors: CM
Flow of events:
1) User removes the event.
Entry condition: User must be CM of that club.
Exit condition: The event has been removed and all members of that club have been notified.

Use case name: Scan QR
Participating actors: CM
Flow of events:
1) The user scans a member's QR code.
2) User is notified that the scan is complete.
Entry condition: User must be CM of that club.
Exit condition: Owner of the scanned QR will be marked as present for that event.

Use case name: Kick member
Participating actors: CM
Flow of events:
1) User chooses which member to kick.
2) User kicks the member.
Entry condition: User must be CM of that club.
User is currently in a tab that displays the list of members.
Target member is not a CM.
Exit condition: Member has been kicked.

Use case name: Veto event
Participating actors: Advisor
Flow of events:
1) User is shown a list of events of the club they are an advisor of that are planned to take place.
2) If deemed inappropriate, the user may click on the "veto" button.
3) The event is cancelled, the event post is removed, event's participants and club CMs are notified.
Entry condition: User is viewing the veto screen.
Exit condition: User has vetoed an event.
User closed the veto screen.

Use case name: See club posts
Participating actors: Viewer
Flow of events:
1) User sees a list of club posts.
2) User scrolls through the list provided.
3) User may interact with the posts.
Entry condition: User has entered a tab that displays club posts.
Exit condition: User moves to another tab or goes to a profile.

Use case name: Get shareable link
Participating actors: Member or Advisor
Flow of events:
1) User clicks the "get link" button under a post.
2) User is prompted that the link is copied to the clipboard.
Entry condition: User must currently be in a tab that displays club posts.
Exit condition: The post is copied to the clipboard.

Use case name: Participate in a poll
Participating actors: Member
Flow of events:
1) User sees the votes cast so far.
2) User votes on an answer on a poll post.
3) User sees their choice on their screen.
4) User may change their vote.
Entry condition: User must currently be in a tab that displays club posts.
User is a member of the club that has posted the poll.
The poll has not yet ended.
Exit condition: User has cast a vote on the poll.

Use case name: Declare participation in an event
Participating actors: Member
Flow of events:
1) User sees the number of participants and the capacity of the event.
2) User clicks on the "participate" button that notifies the user's participance in the event.
3) User may give up on their participation.
Entry condition: User must currently be in a tab that displays club event posts.
The event's capacity should not be full.
Exit condition: User has changed their participation status on an event.

Use case name: See club profiles
Participating actors: Viewer
Flow of events:
1) User clicks on a club's profile picture or name.
2) User is taken to that club's profile.
3) User may see and interact with the club's posts.
Entry condition: User is on a tab that they can click on a club.
Exit condition: User moves to another tab.

Use case name: Pin Post
Participating actors: Member or Advisor
Flow of events:
1) User sees if the post is pinned or not by looking at the "pin" button.
2) User clicks on the "pin" button to pin the post.
3) The post is added to the "Pinned Posts" section.
Entry condition: User must currently be in a tab that displays club event posts.
Exit condition: Post is pinned and added to "Pinned Posts".

Use case name: Unpin Post
Participating actors: Member or Advisor
Flow of events:
1) User sees if the post is pinned or not by looking at the "pin" button.
2) User clicks on the "pin" button to unpin the post.
3) The post is removed from the "Pinned Posts" section.
Entry condition: User must currently be in a tab that displays club event posts.
Exit condition: Post is unpinned and is removed from "Pinned Posts".

Use case name: Unsubscribe from a club
Participating actors: Member
Flow of events:
1) User clicks the "unsubscribe" button of a club.
2) The user leaves the club.
Entry condition: User is currently viewing a club's profile page.
User is subscribed to the club.
User is not the CM of the club.
User is not the Advisor of the club.
Exit condition: User is no longer a member of the club.

Use case name: Subscribe to a club
Participating actors: Member
Flow of events:
1) User clicks the "subscribe" button of a club.
2) The user becomes a member of the club.
Entry condition: User is currently viewing a club's profile page.
User is not already subscribed to the club.
User is not the CM of the club.
User is not the Advisor of the club.
Exit condition: User is a member of the club.

Use case name: Enter chatroom of a club
Participating actors: Member or Advisor
Flow of events:
1) User clicks the "chat bubble" button of a club.
2) The user enters the chatroom of the club.
3) User can see previous messages of the chatroom.
Entry condition: User is currently viewing a club's profile page.
User is a member of the club.
Exit condition: User is in the chatroom.

Use case name: Share posts to the chatroom
Participating actors: CM or Advisor
Flow of events:
1) User clicks the "plus" button in the chatroom.
2) User clicks on the "share posts" option.
3) User is displayed the posts of the club that owns the chatroom.
4) User chooses a post and clicks send.
5) The selected post is sent to the chatroom.
Entry condition: User is currently inside the chatroom of a club.
User is a CM or an Advisor of the club.
Exit condition: A post among the club's posts is sent to the club's chatroom.

Use case name: Send message to the chatroom
Participating actors: Member or Advisor
Flow of events:
1) User clicks on the "message" bar.
2) User types a message and presses enter or clicks the send button.
3) User's message is displayed in the chatroom.
Entry condition: User is currently inside the chatroom of a club.
User is not muted.
Exit condition: User successfully sends a message to the chatroom.

Use case name: Go to shared post
Participating actors: Member or Advisor
Flow of events:
1) User presses on the event message.
2) User is taken to the event post.
3) User may interact with the post as usual.
Entry condition: User is currently inside the chatroom of a club.
Exit condition: User is taken to the post.

Use case name: See member list
Participating actors: Member or Advisor
Flow of events:
1) User clicks on "see members" clickable.
2) User is shown a list of members of the related club.
3) User may look at the members on the list.
4) User may try to kick a member if the user is a CM.
Entry condition: User is either at the club profile or the chatroom.
Exit condition: User closes the list.

Use case name: Search clubs
Participating actors: Member or Advisor
Flow of events:
1) User clicks on the search bar on the "Explore Tab".
2) User writes the name of the club that they want to search.
3) The results of the search are shown to the user in a scrollable manner.
Entry condition: User is in the explore tab.
Exit condition: User has done a search and the results are displayed.

Use case name: Check calendar
Participating actors: Member or Advisor
Flow of events:
1) The events of that month are shown in a calendar, on their respective days.
2) On each day of the calendar, logos of clubs that the user is subscribed to and are hosting events and the user is subscribed to will be shown.
Entry condition: User must be logged in.
Exit condition: The calendar is successfully shown to the user.

Use case name: Go to event page
Participating actors: Member or Advisor
Flow of events:
1) User clicks on the logo of the club.
2) User is taken to the event post.
3) User may interact with the post as usual.
Entry condition: User must be viewing the calendar.
Exit condition: User is successfully sent to the event page.

Use case name: Check notifications
Participating actors: Member or Advisor.
Flow of events:
1) User clicks on the notifications button shown on the page.
2) Notifications will be shown in chronological order.
Entry condition: User must be logged in.
Exit condition: User sees the notifications.

Use case name: Remove notification.
Participating actors: Member or Advisor.
Flow of events:
1) User clicks the "delete" icon of a notification.
2) The notification will get deleted.
3) The notification will disappear in the UI, if there is any.
Entry condition: User must be viewing the notifications page.
Exit condition: Notification will be deleted successfully.

Use case name: See personal QR code
Participating actors: Member
Flow of events:
1) User specific QR code is shown on the screen.
Entry condition: User must be logged in.
Exit condition: QR code is successfully shown.

Use case name: See subscribed clubs
Participating actors: Member or Advisor.
Flow of events:
1) A list of subscribed club accounts is shown to the user.
2) User may click on their club name or picture to go to their profile.
Entry condition: User must be logged in.
Exit condition: Subscribed club accounts have been shown successfully.

Use case name: See personal QR code
Participating actors: Member
Flow of events:
1) User specific QR code is shown on the screen.
Entry condition: User must be logged in.
Exit condition: QR code is successfully shown.

Use case name: Edit user profile
Participating actors: Member or Advisor
Flow of events:
1) User views their profile.
2) User may edit his/her profile picture or password.
Entry condition: User must be logged in.
Exit condition: Changes (if occurred) are saved.

Use case name: Change profile picture.
Participating actors: Member or Advisor
Flow of events:
1) User uploads a new profile picture.
1) User confirms the picture change.
Entry condition: User must be viewing profile page.
Exit condition: Profile picture has been changed successfully.

Use case name: Change password.
Participating actors: Member or Advisor
Flow of events:
1) User writes the current password.
1) User writes the new password twice.
2) User confirms the password change.
Entry condition: User must be viewing profile page.
Exit condition: Password has been changed successfully.

Use case name: See pinned posts
Participating actors: Member or Advisor
Flow of events:
1) User clicks on the "pinned posts" tab.
2) User is shown a list of the posts that they've pinned.
3) User may click on one of the posts to go to the post page.
Entry condition: User is logged in.
Exit condition: User has changed the tab.

Use case name: Signup
Participating actor: Viewer
Flow of events:
1) User writes their Bilkent mail and password to be used.
2) User submits the credentials.
3) User is warned if the mail address is not a Bilkent mail address or an existing account is using the mail address.
4) If the mail address is an unused Bilkent mail address a verification code is sent to the mail address.
5) User submits the verification code.
6) User gets warned if the code is incorrect.
7) User is acknowledged that signup is complete.
Entry condition: User must be in the login page.
Exit condition: User successfully signs up.

Use case name: Login
Participating actors: Viewer
Flow of events:
1) User writes their credentials and submits them.
2) If credentials are wrong an error message is shown.
3) If credentials are correct the user successfully logs in.
Entry condition: User is on the login page.
Exit condition: User is successfully logged in.

Use case name: Use forgot password
Participating actors: Member or Club Advisor.
Flow of events:
1) User writes their mail address.
2) A verification code is sent to the user's mail address.
3) User submits the verification code.
4) User is warned if the code is incorrect.
5) User is asked to write the new password if the verification code is correct.
6) User submits the new password.
Entry condition: User must be viewing the login page.
Exit condition: User's password has been successfully changed.

Use case name: Change language
Participating actors:
Flow of events:
1) User clicks the "change language" button.
2) User chooses the language by clicking on a flag.
3) Language has changed.
Entry condition: User must be viewing the login page or must be logged in.
Exit condition: Language is changed successfully.

Class Diagram (Click here to open the diagram in a higher quality)

Figure 2: Class Diagram

User:
Once a user is verified, an instance of this class is constructed. A user can pin to posts to get notifications about them. A user has a list of clubs that they are a member of.

Authorized User:
Extends User class. An authorized user has functions like deletePost(), to provide different types of authorized user needs like Club Admins and Advisors.

Club:
A club has a list of members, a list of admins and an Advisor. Notifications can be sent to all members or only to Admins by Club class. A club has posts and a chat.

Post:
Post is an abstract class. They have to be in a club and can only be posted by authorized users of that club.

Event:
An event is in the form of a post that has to have a date planned and a participant list, and a QRCode. It may have prerequisite events and a specified duration.

Poll:
A Poll is a type of post that has choosable options.

Option:
Option is a simple combination of a text and the number of votes it got.

QRCode:
Holds the data of a QR Code to track participation.

Notification:
Holds the data of notifications.

Chat:
Chat instances are to be used in Posts and Clubs. It has a list of ChatContent.

Message:
Contains data of a message that has a text, author and a date. Can have a post and acts as a link to that post in the chat environment internally in the application.

TextMessage:
Contains a text to act as the message.

Calendar:
Contains dates of the events of clubs that the user has subscribed to.

Dynamic Models

State Diagrams

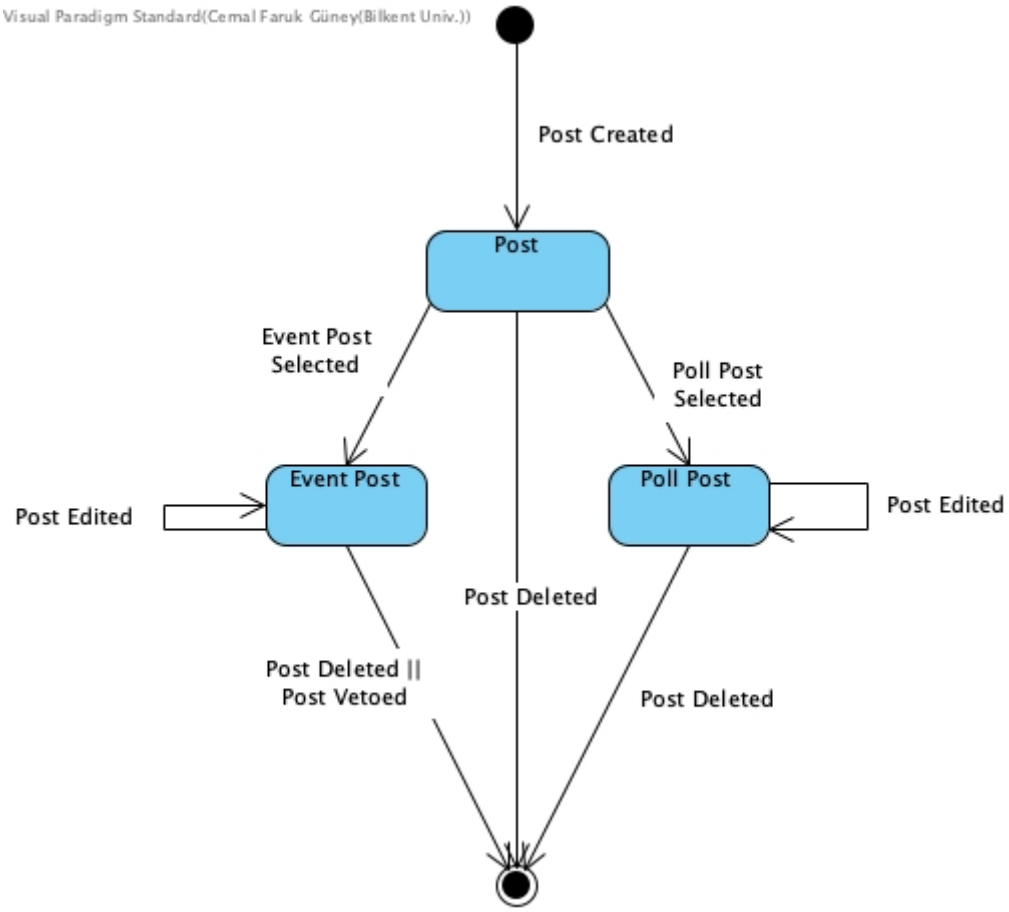

Figure 3: Post State Diagram

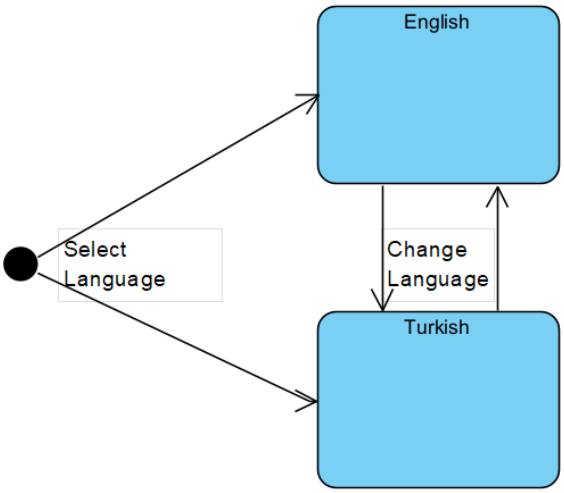

Figure 4: Language State Diagram

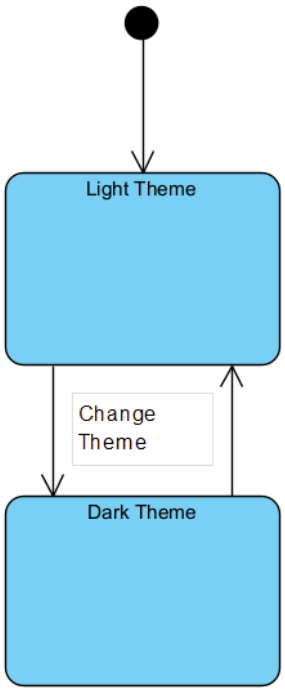

Figure 5: Theme State Diagram

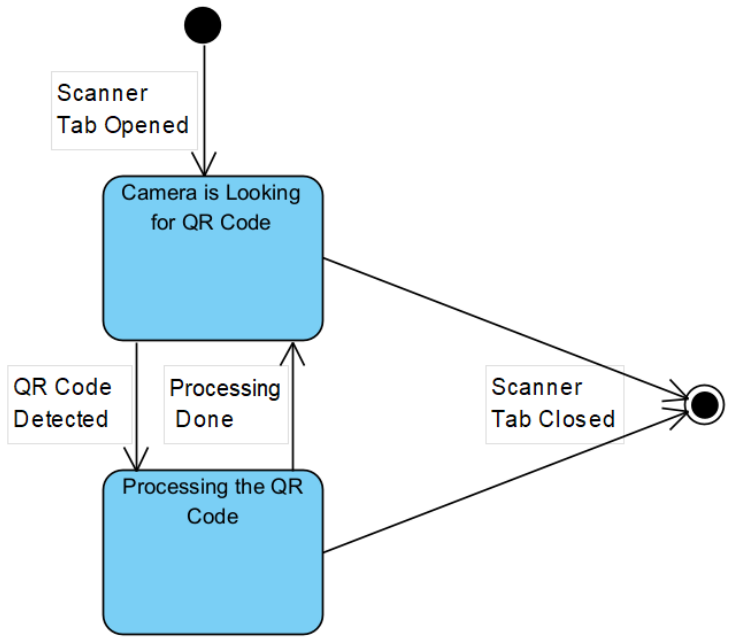

Figure 6: QR Scanner State Diagram

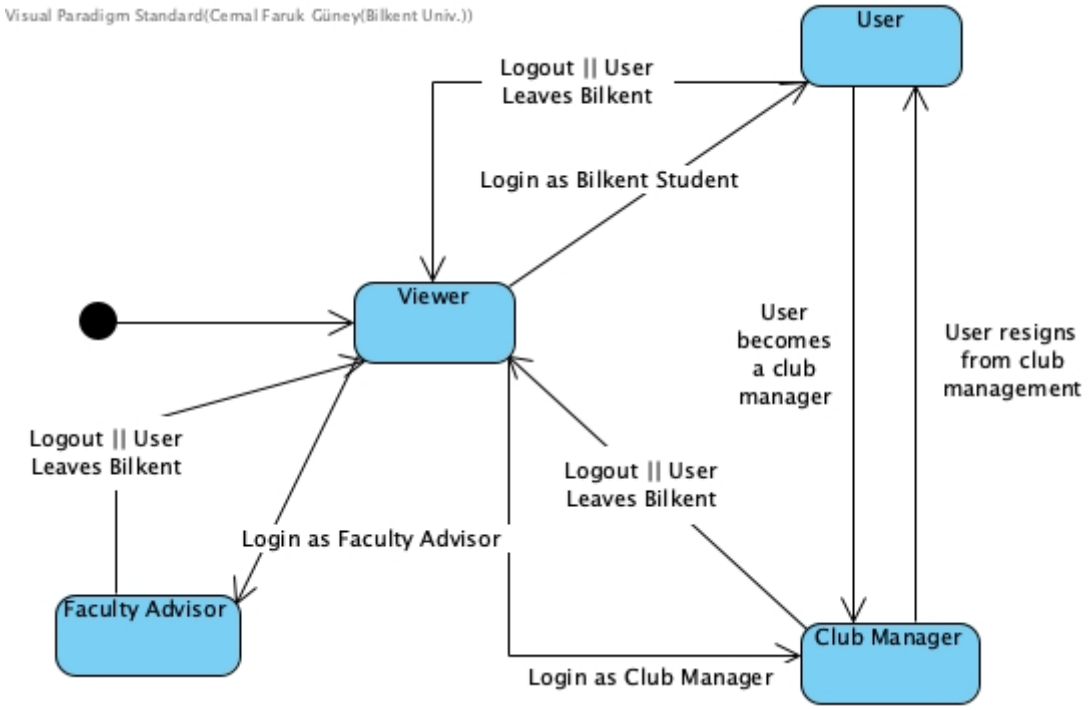

Figure 7: User State Diagram

Activity Diagrams

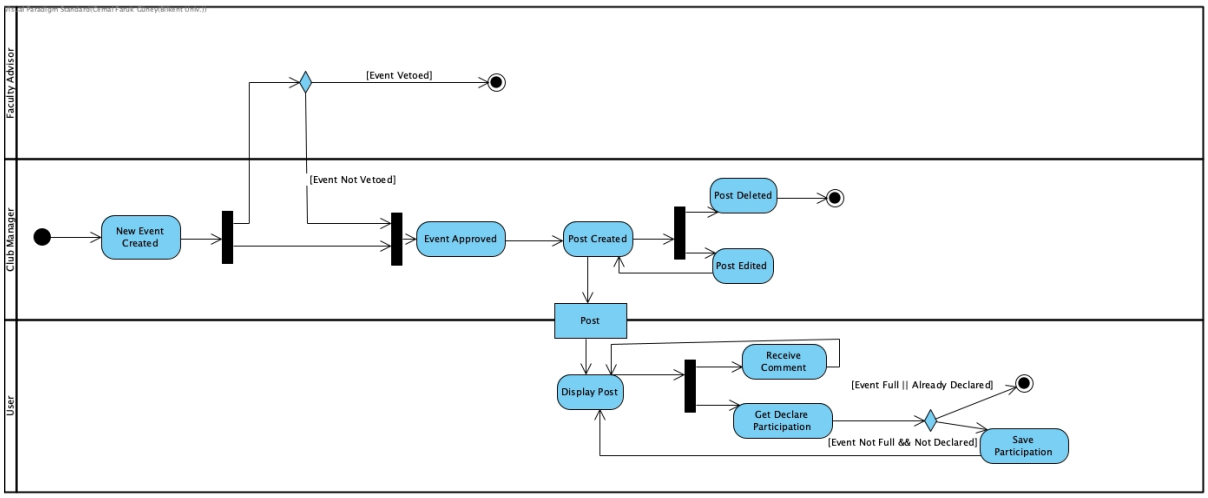

Figure 8: Event Activity Diagram

Every event post in the system is created by club managers. The event is reviewed by the faculty advisor of the club and he/she has the option to veto the event. If the event is vetoed the post will be deleted. Otherwise, club managers will be able to edit posts. Students will be able to see newly created posts. They have the option to comment or declare participation. If they try to declare participation to an event and the event is not full their participation will be saved and expected.

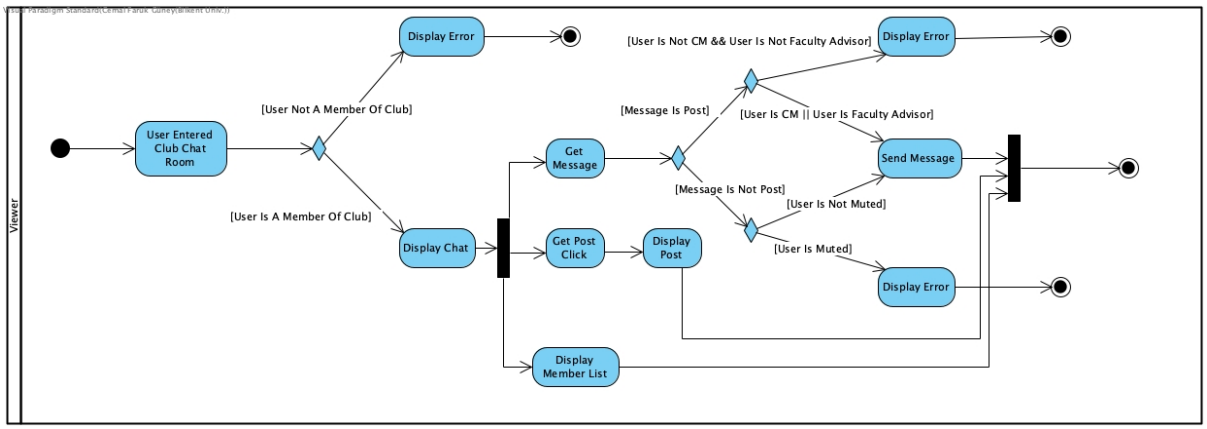

Figure 9: Chat Activity Diagram

Every person that uses the application has the ability to go to the chat screen in the club profile. However if the user is not logged in or they are not a member of the club the application displays an error. Otherwise, if they are a logged in member of the certain club they can see the chat. In the chat screen the club member has 3 different options. First option is sending a message if the user is one of the managers of the club they can send a text message or a link to a post. Normal members of the club can only send text messages if they are not muted in the chat. The second option is clicking a post link, the user is navigated to the post page after they click the link. The third option is viewing the member list from the chat. This list shows the members of the club.

Sequence Diagrams

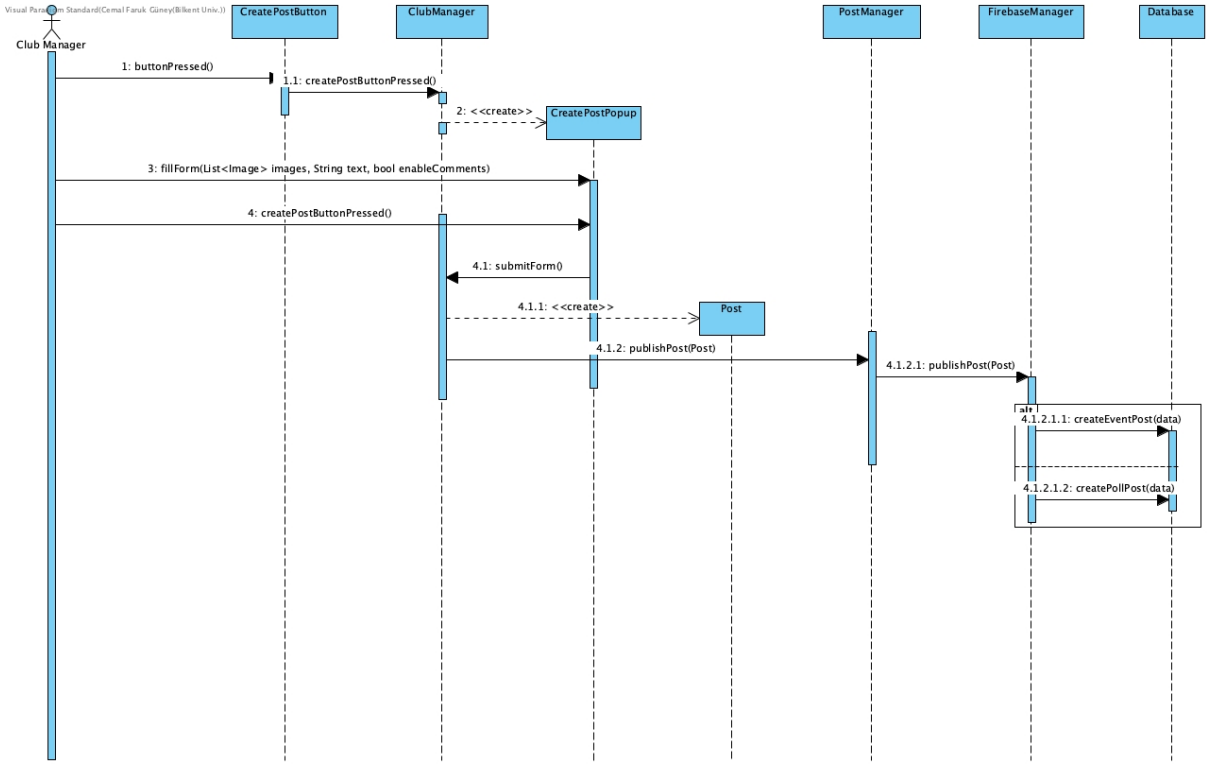

Figure 10: Create Post Sequence Diagram

Posts are created when a Club Manager presses the CreatePostButton*. After the button is pressed an instance of ClubManager is called. The ClubManager then displays the CreatePostPopup*. The Club Manager fills the fields in this pop-up and presses the “Create Post” button at the bottom of the pop-up. ClubManager gets the information from this pop-up and creates a new Post object with the given information. ClubManager calls the PostManager’s publishPost method using the new Post object as a parameter. Then, the PostManager calls the FirebaseManager with its publishPost method. FirebaseManager creates the post in the Database according to the type of the post (event or poll).
*These two classes are not included in the class diagram because these are boundary objects created using Flutter Buttons and Views and they have no name (CreatePostButton and CreatePostPopup).

User Interface

Figure 11: Not logged in event post

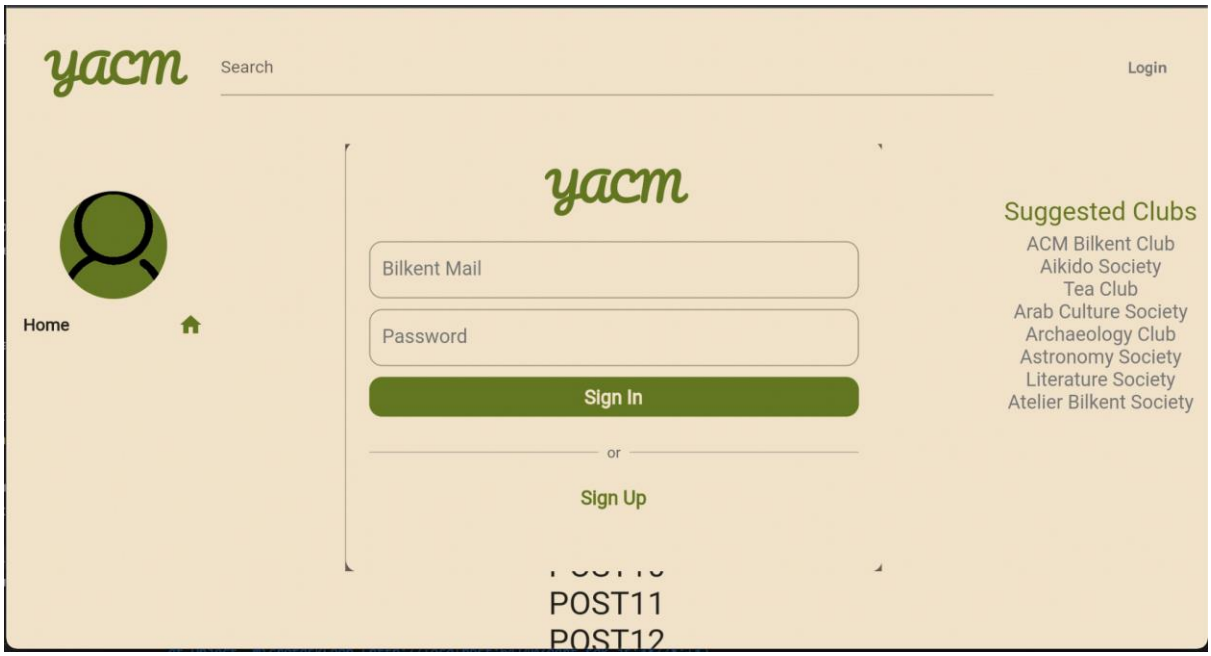

Figure 12: Login Screen

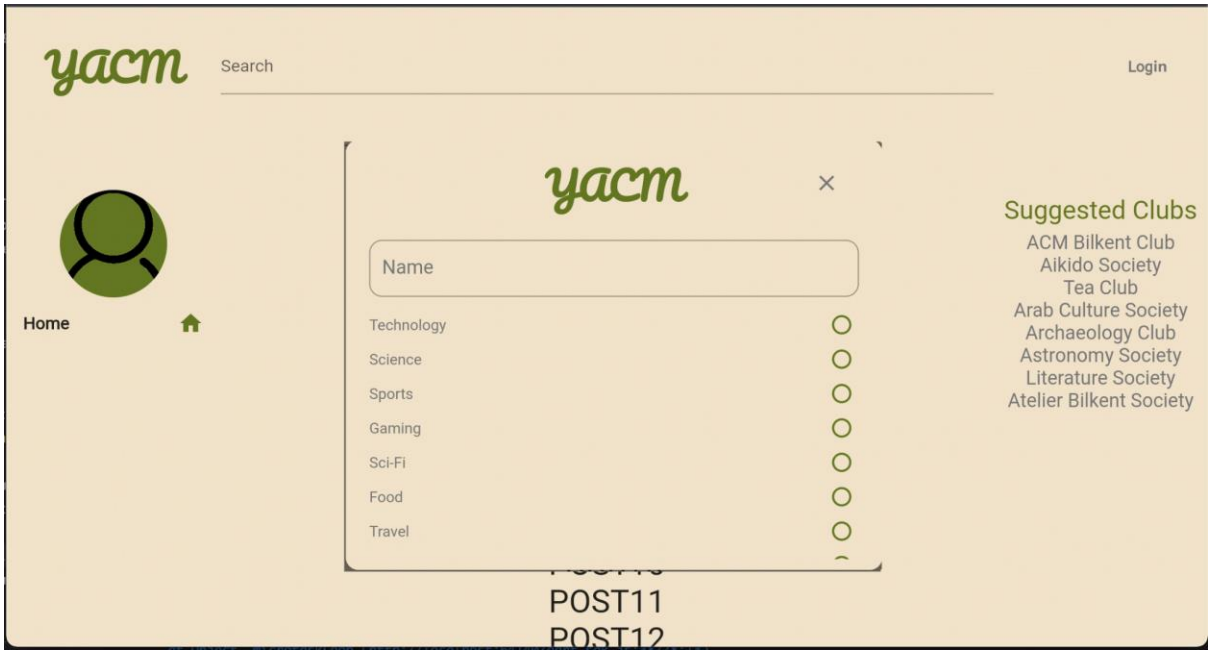

Figure 13: Choosing interests

Figure 14: Register information under interests

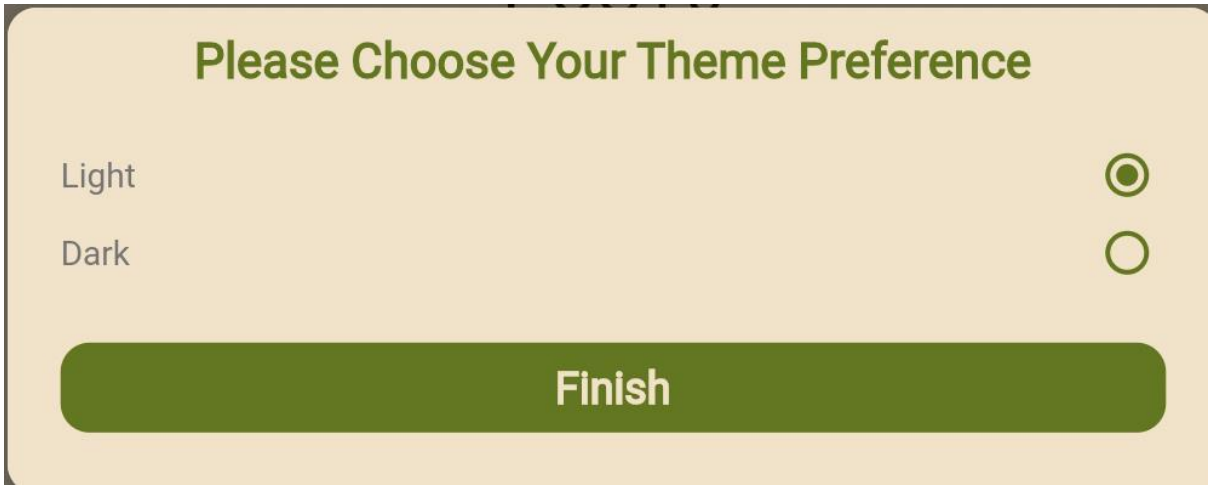

Figure 15: Theme preference window

Figure 16: Home Tab

Figure 17: Pinned Tab

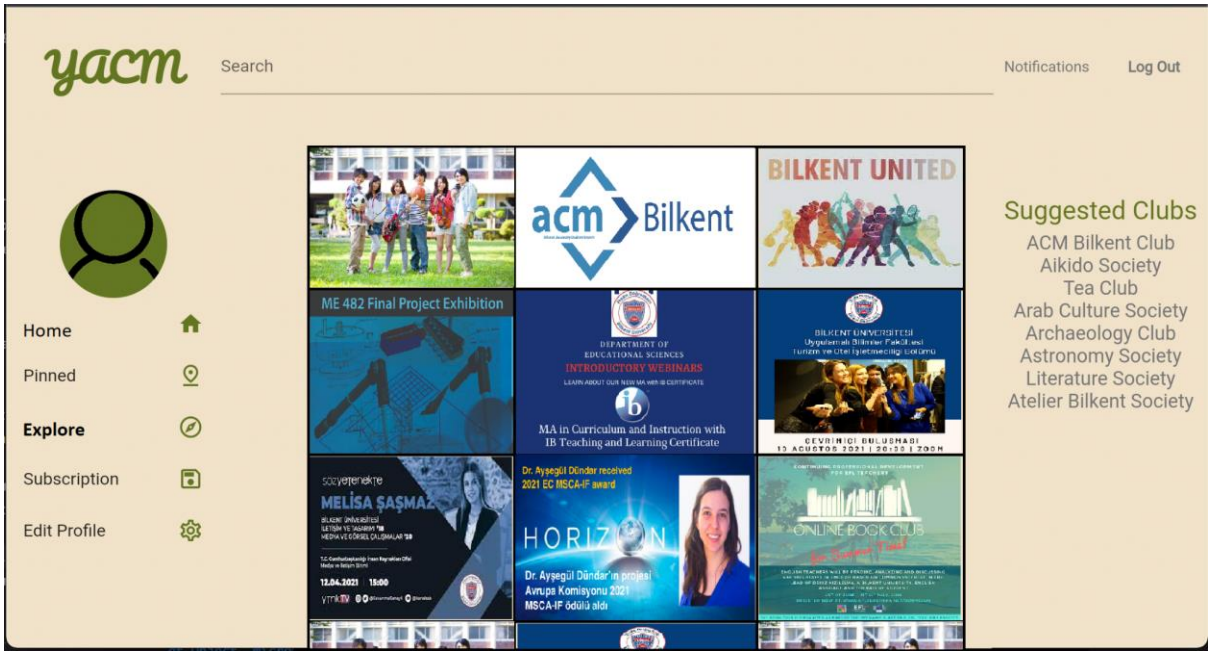

Figure 18: Explore Tab

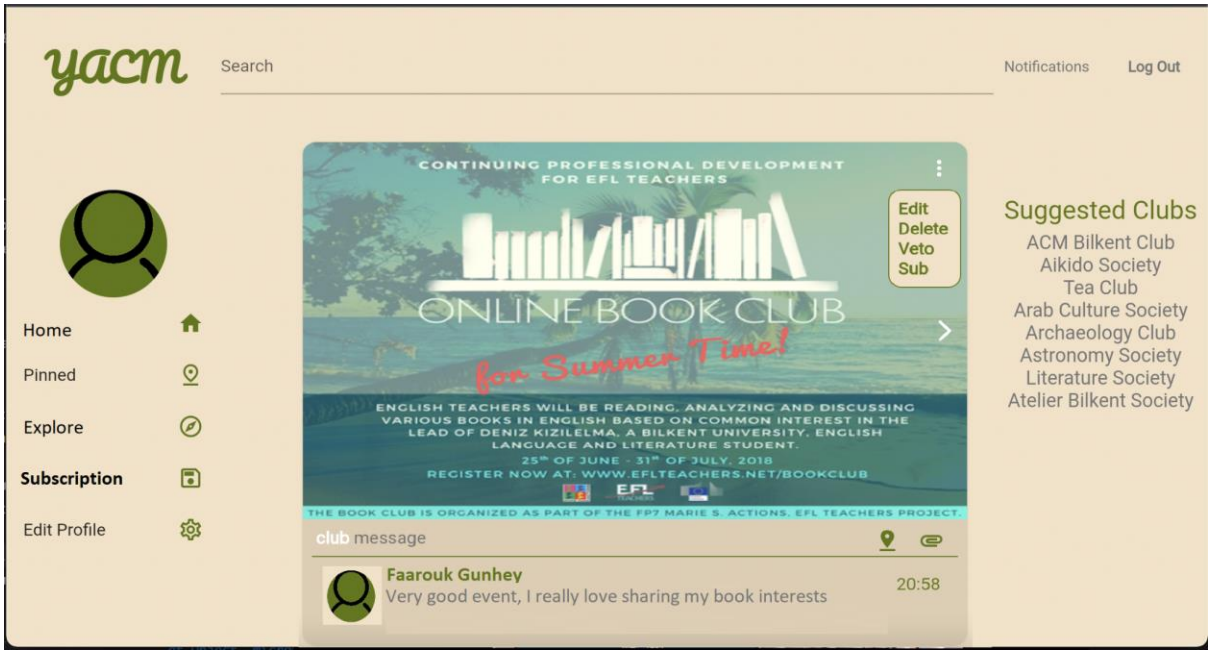

Figure 19: Subscription Tab

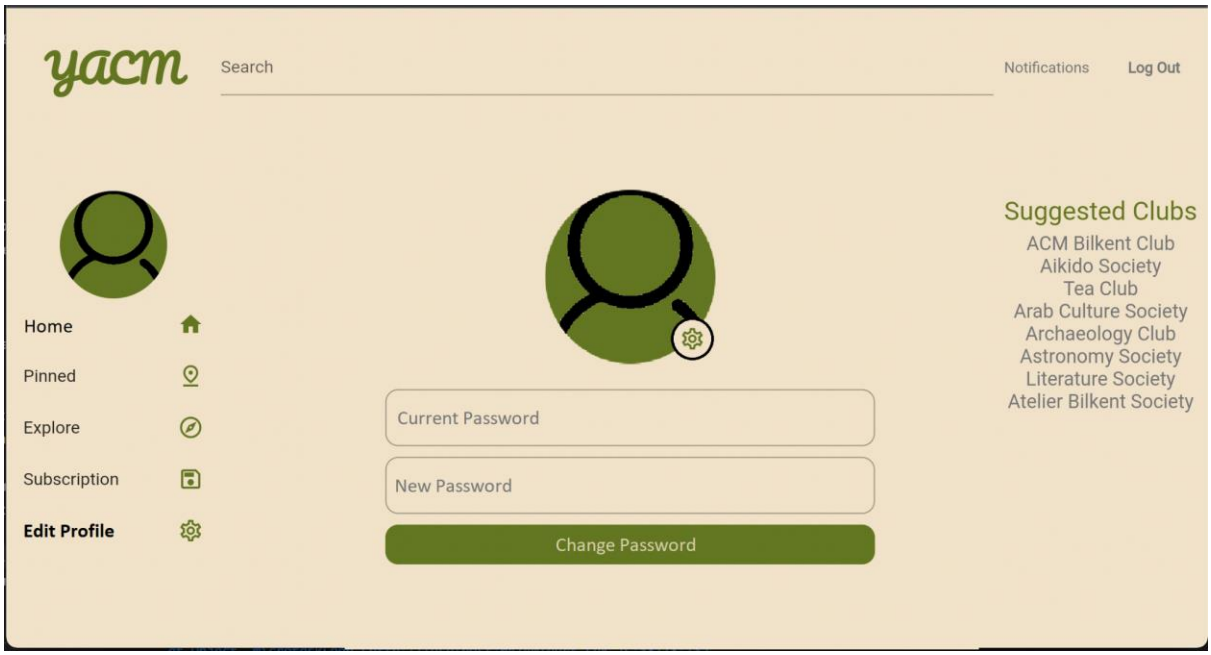

Figure 20: Edit Profile Tab

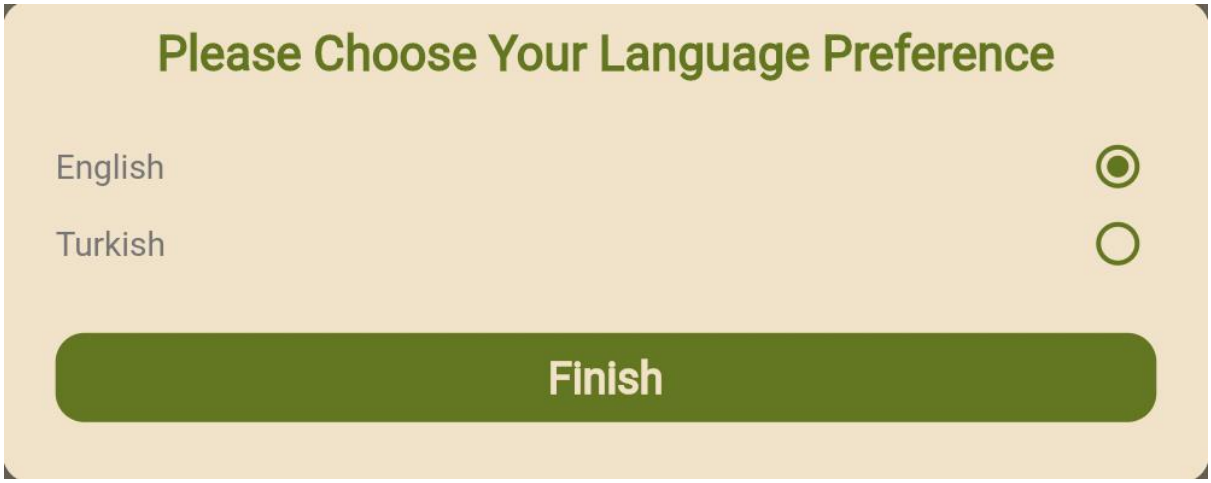

Figure 21: Choose language screen

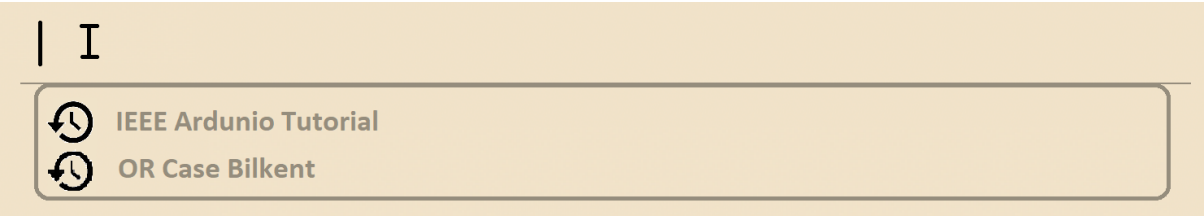

Figure 22: Search history when search is clicked

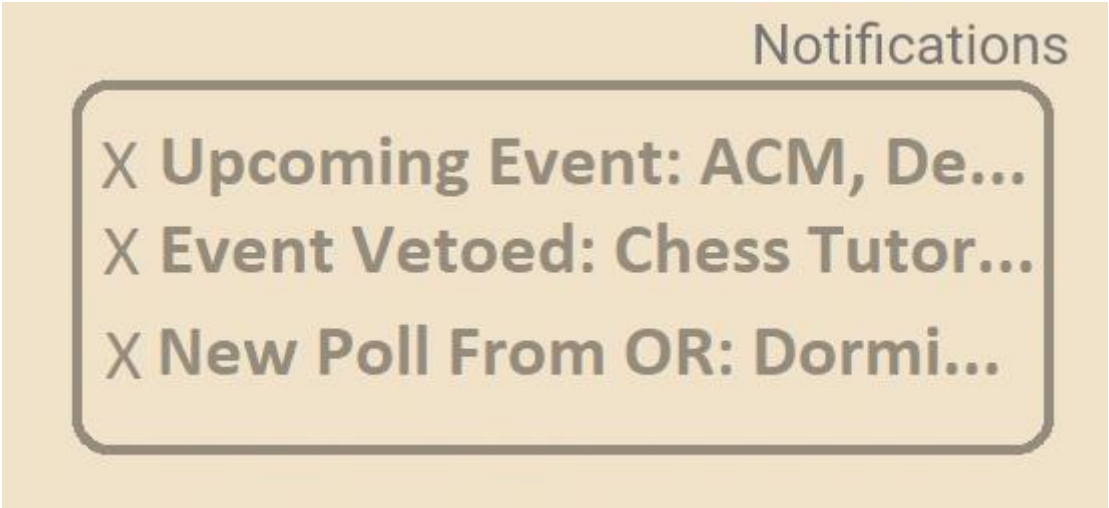

Figure 23: Notifications pop-up

Figure 24: Club profile

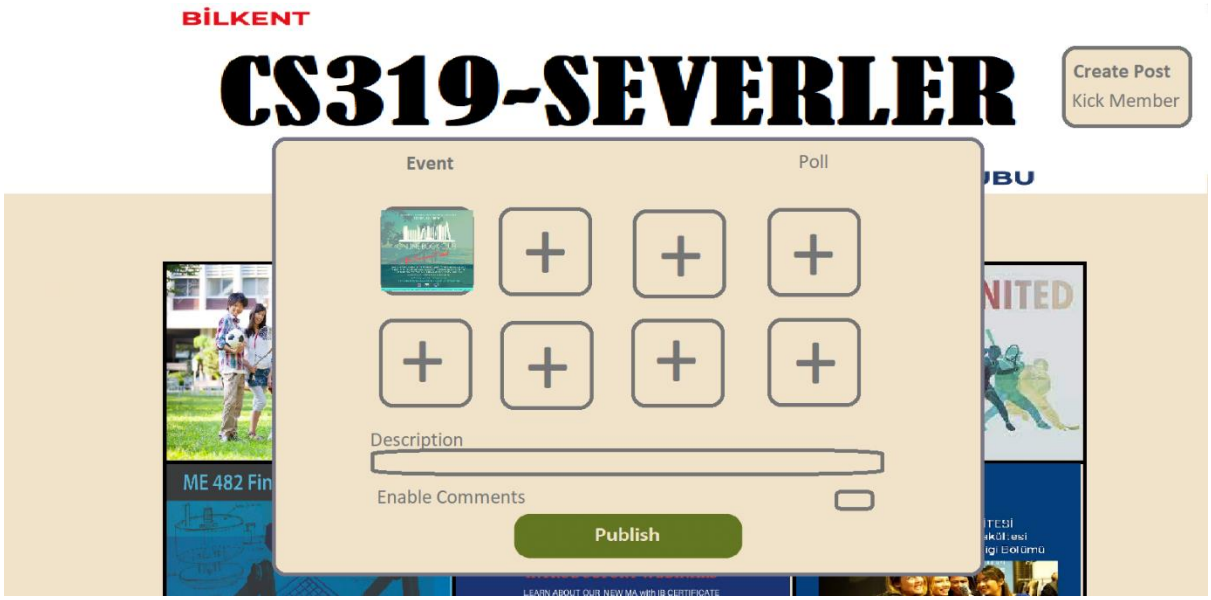

Figure 25: Creating an event post screen

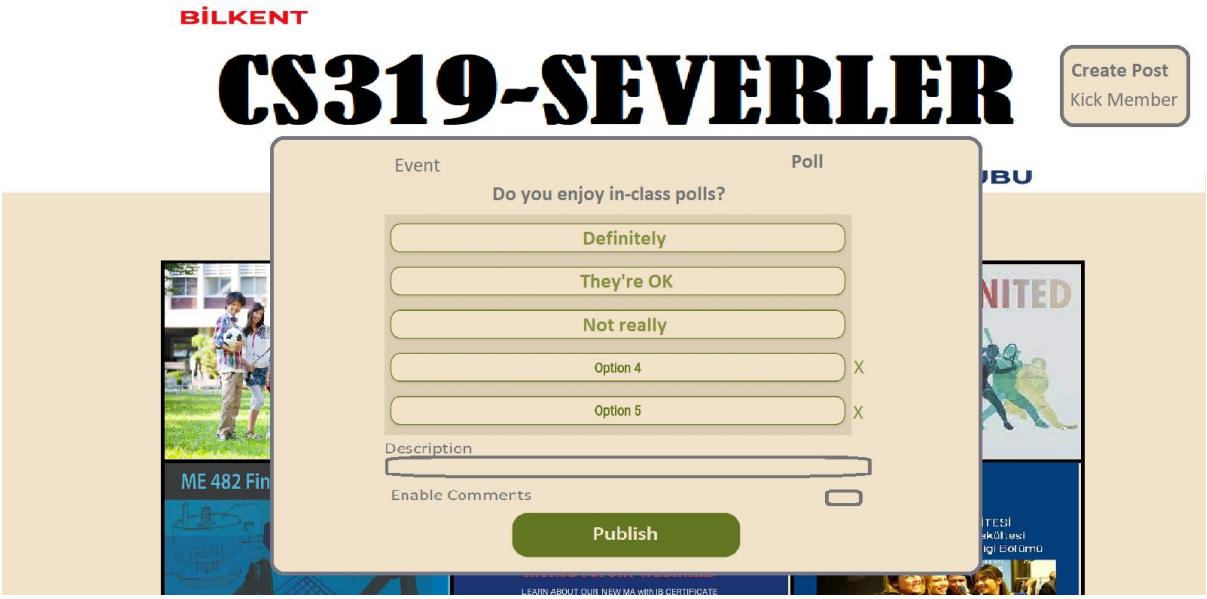

Figure 26: Creating a poll post screen

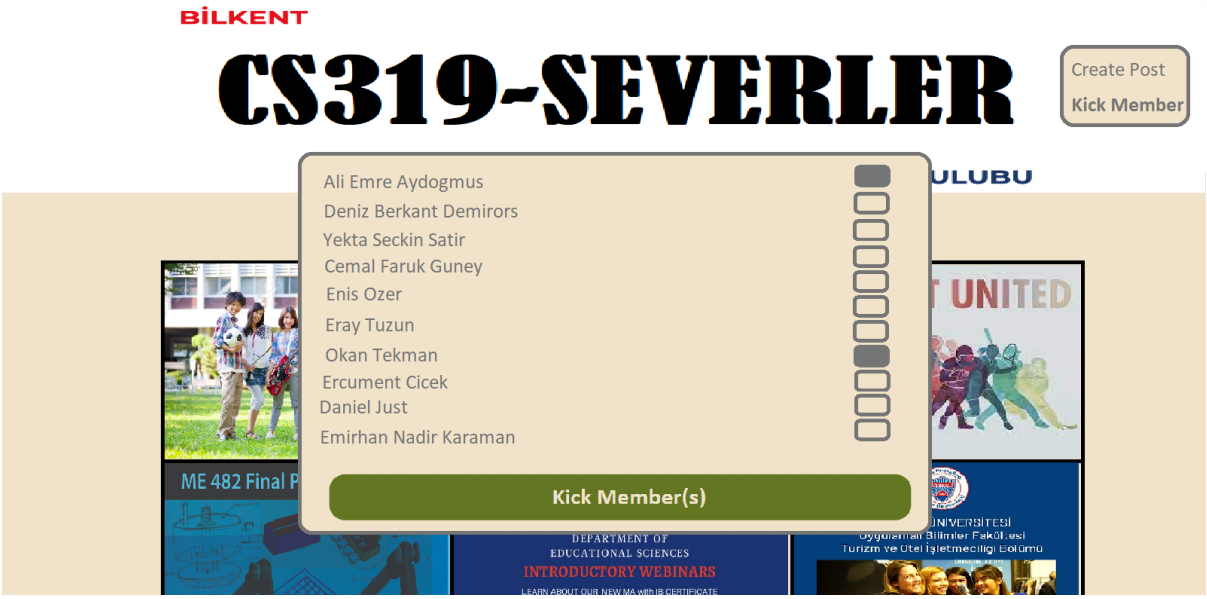

Figure 27: Kicking Members Screen

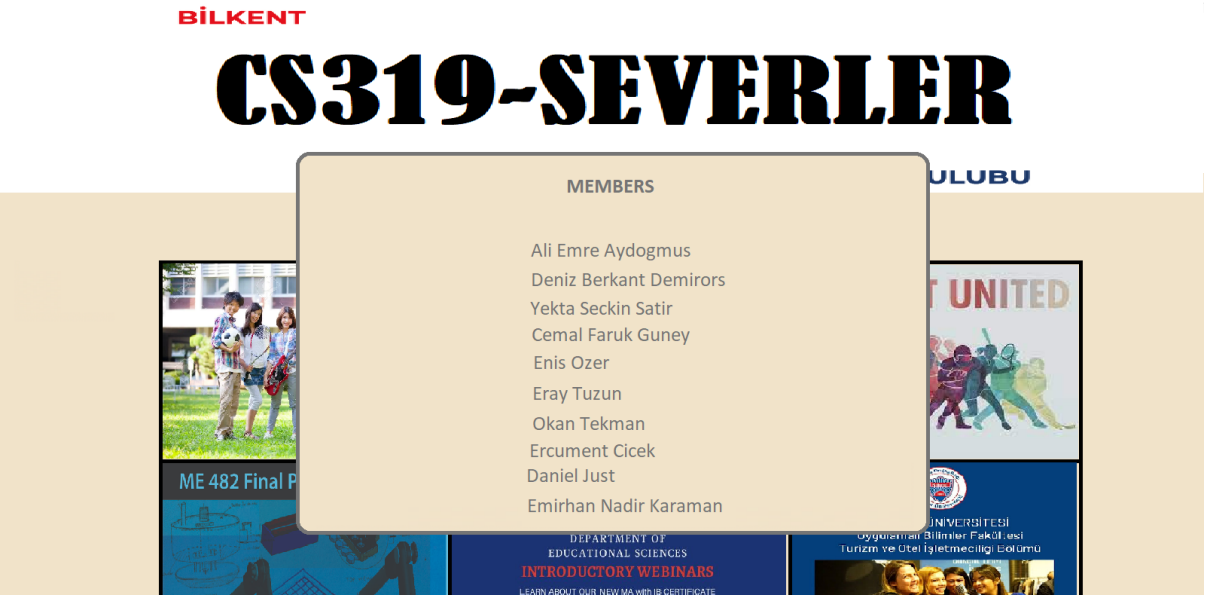

Figure 28: Member list screen

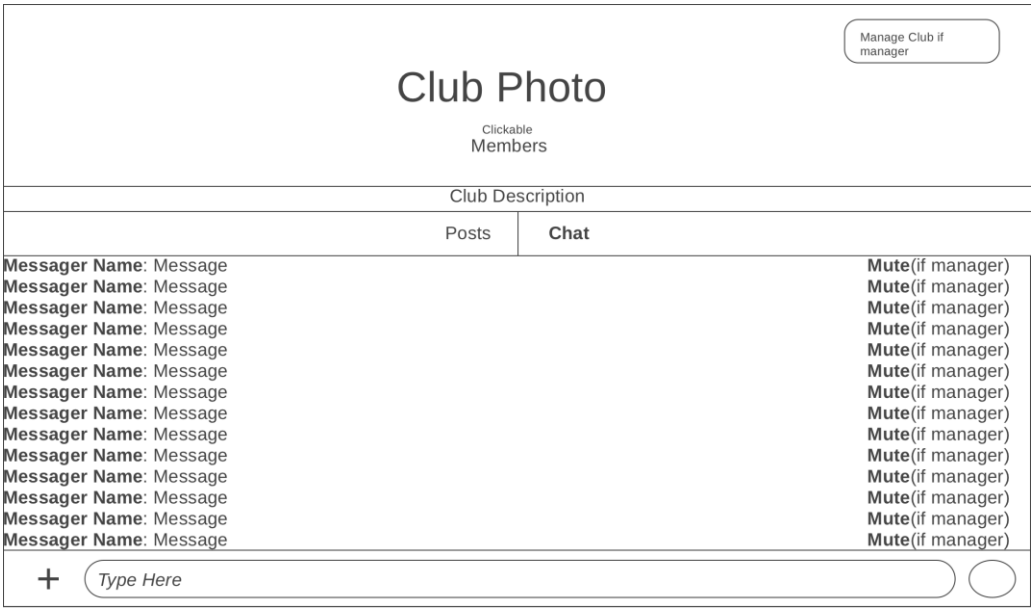

Figure 29: Club chat screen

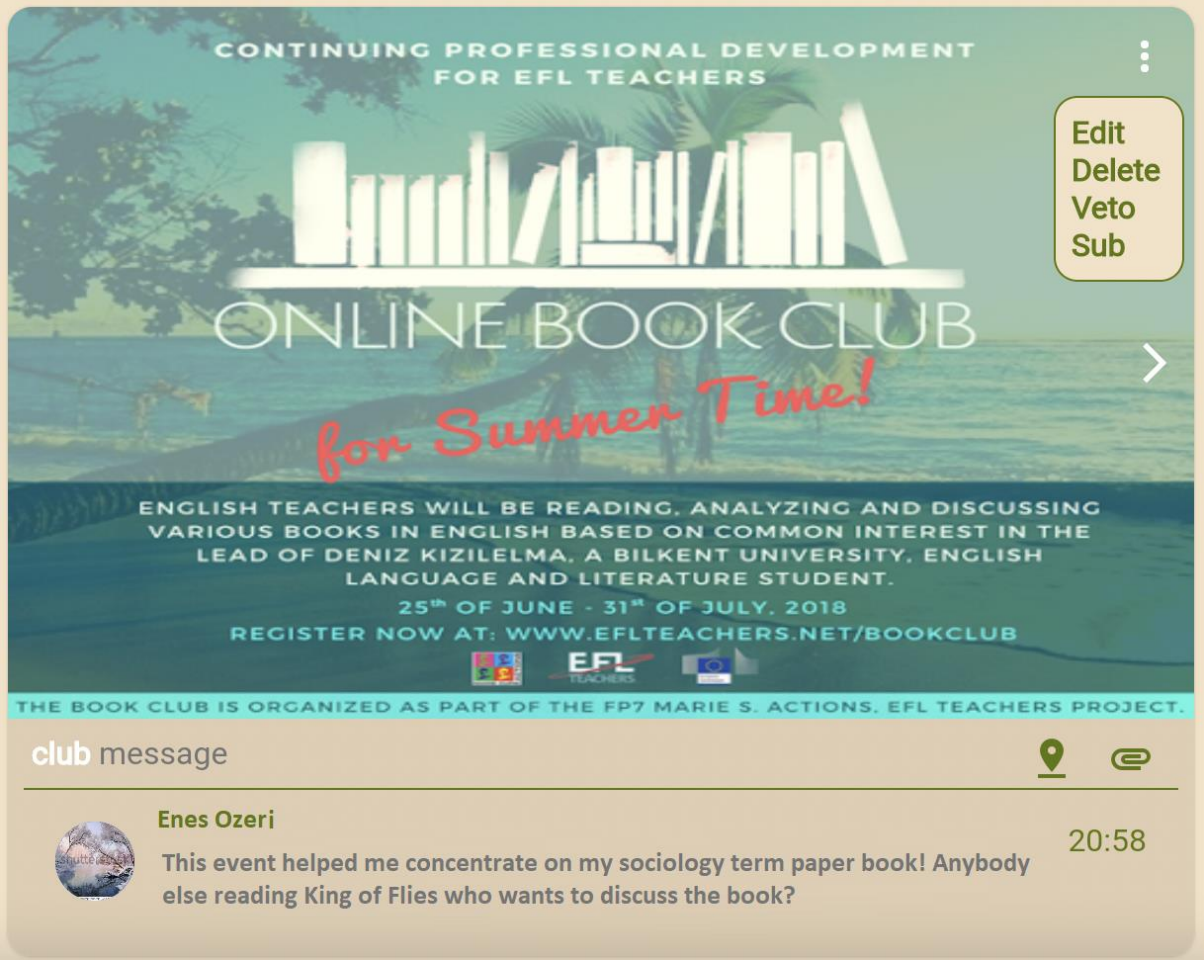

Figure 30: An Event Post

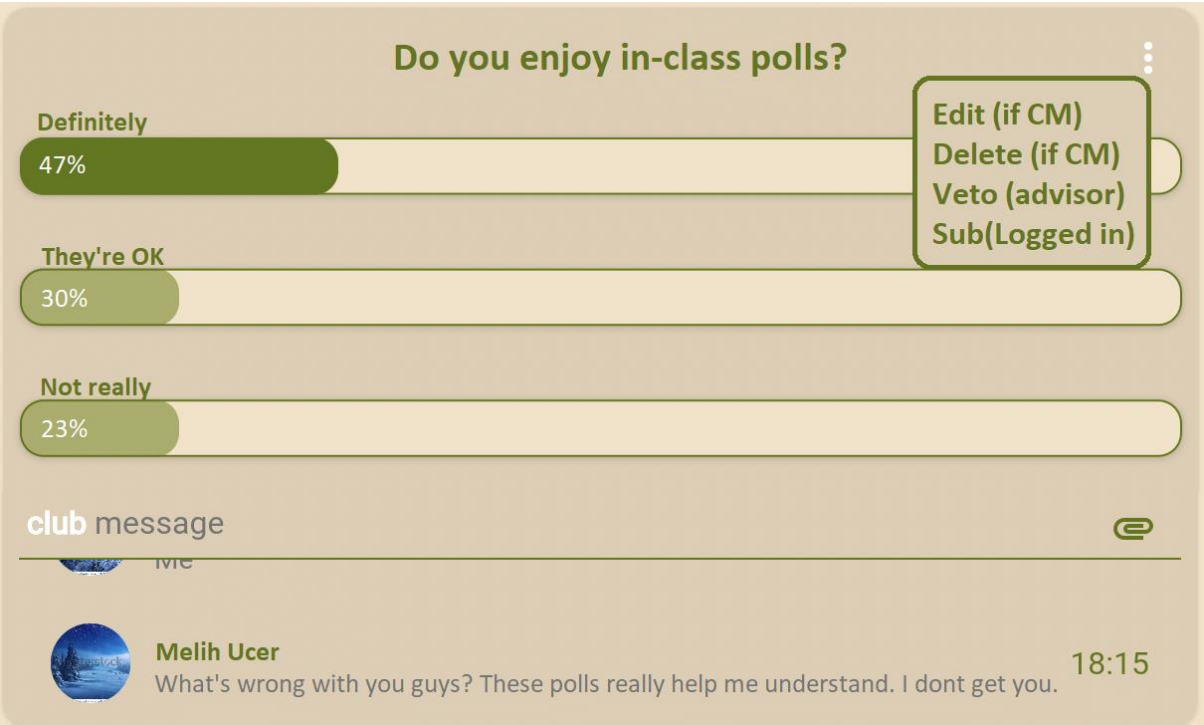

Figure 31: A poll post