



Bilkent University

Department of Computer Engineering

CS 319 - Object Oriented Software Engineering
Fall 2023
Section 1
Term Project

BilConnect

(Deliverable 1-2 Final)
Team 3
Members

Abdullah Samed Uslu
22003598

Celal Salih Türkmen
22102498

Emre Akgül
22102310

Murat Çağrı Kara
22102505

Umut Bora Çakmak
22101806

Table of Contents

Table of Contents	2
1. Non-Functional Requirements	4
2. Tech Stack	5
3. Use Case Diagram	6
1) Settings	6
a) Name: DeleteAccount	6
b) Name: EditAccount	6
c) Name: ChangeLanguage	7
d) Name: ToggleDarkMode	7
2) Moderation Services	8
a) Name: DeleteClubEventByPrivilege	8
b) Name: ViewChatLog	8
c) Name: ResolveReport	9
d) Name: DeletePostByPrivilege	9
e) Name: SuspendAccount	9
f) Name: ViewReports	10
g) Name: CreateTag	10
h) Name: RemoveTag	11
i) Name: EditTag	11
j) Name: ViewAllPosts	12
3) Club Event Calendar	12
a) Name: ViewEventDetail	12
b) Name: ViewEventCalendarbyMonth	12
c) Name: ViewEventCalendarbyDay	13
d) Name: EditClubEvent	13
e) Name: CreateClubEvent	14
f) Name: DeleteClubEvent	14
4) Second-hand Sales	15
a) Name: PostViewerViewChat	15
b) Name: SeeUserChatList	15
c) Name: ViewOwnPosts	16
d) Name: RateUser	16
e) Name: StartChat	16
f) Name: PostOwnerViewChat	17
g) Name: ViewSearchedPosts	17
h) Name: MarkAsFavoritePost	18
i) Name: DeletePost	18
j) Name: EditPost	19
k) Name: CreateDonationPost	19
l) Name: CreateBorrowingPost	20
m) Name: CreateTravellingPost	20
n) Name: CreateSecondHandItemPost	21

o) Name: CreatePetAdoptionPost	22
p) Name: CreateLostItemPost	22
q) Name: CreateEventTicketPost	23
r) Name: ReportEvent	24
s) Name: ReportPost	24
t) Name: ReportUser	25
u) Name: ViewMarkedAsFavoritePosts	25
v) Name: ViewPostDetail	25
5) Login/Logout/New Account Services	26
a) Name: Forgot Password	26
b) Name: Login	27
c) Name: CreateUserAccount	27
d) Name: CreateClubAccount	27
e) Name: Logout	28
4. Class Diagram	29
5. Sequence Diagrams	33
5.1 User Sends Report	33
5.2 User Shares Post	34
5.3 Moderator Resolves Report	35
5.4 Student Club Creates Club Event	36
6. State Diagrams	37
6.1 State Diagram For Chat	37
6.2 State Diagram For Post	38
6.3 State Diagram For Club Event	39
7. Activity Diagrams	40
7.1 Student Club Creating Event	40
7.2 Report Evaluation for Moderators	40
7.3 Post Creation	41
8. Mockups	42

1. Non-Functional Requirements

The non-functional requirements of BilConnect can be subdivided into six categories: speed, exclusivity, security, moderation, reliability, and expandability.

Speed

- The messaging functionality used in second-hand sales should provide a messaging experience that does not cause frustration for the user. The messaging should have a latency of less than 3 seconds.
- Posts on second-hand sales and club event posting have more tolerance towards latency, but still minimizing it to less than 15 seconds will prevent confused users from trying to repost cause they think the post didn't work.

Exclusivity

- This app is exclusive to the Bilkent student body, master and doctorate students, faculty members and teaching staff. Keeping it that way could be achieved by authenticating through the e-mail given by Bilkent University.
- Specifically creation of student club accounts should only be allowed after an authentication process.

Security

- This app will handle personal data such as name and e-mail. It will also contain inter-personal messages which is sensitive data. No data leaks should be allowed.
- The app should also be secure against spam. Since account numbers will be limited due to exclusivity, we can limit the number of posts per account to 20 per day. This will also ensure the identity of the app, preventing it from being run over by wholesalers.

Moderation

- Messaging, posts and club events should not be open to spam or any inappropriate content according to Bilkent University's code of ethics. This rule should be reinforced by a moderator.

Reliability

- The app should be reliable and should not be down for extended periods of time (2 hours for busy periods, 6 hours for non-busy periods, 24 hours for maintenance). It should especially be reinforced for the start of semesters. Maintenance should be done outside of busy periods. The app should not have bugs that may cause any data loss or data leak.

Expandability

- The app has potential for expansion, especially to graduates and other universities that might be considered for the app. To this end, the app should be easily expandable with its current small number of developers which is five.

2. Tech Stack

We will build BilConnect using a tech stack of popular and updated technologies. Our backend is powered by ASP.NET and C#, and our frontend is powered by React. The architecture used by this trio has many tutorials, templates and support forums on the web that we think that it will be very useful in our project.

Backend with ASP.NET

- BilConnect's backend will be built using ASP.NET, a web framework that meets our requirements of reliability and security. This framework efficiently operates the server-side processes, ensuring stability. Moreover, it is highly demanded in the industry, which makes it suitable for learning from many sources.

C# for Programming

- We will use C#, a popular object-oriented programming language, to program our app. C# is based on C++ and highly demanded in the industry. It offers high efficiency and many libraries, contributing to the reliability and performance of BilConnect.

SQL for Database

- Our data management system will use SQL, the most popular database language. SQL makes it possible for us to efficiently store, fetch, and manage various data, ensuring that BilConnect operates fast and securely.

Frontend using React

- We will be using React to code the frontend of BilConnect. It is a popular JavaScript library. It provides a fancy user interface, enhancing the appearance of our application. This enables us to create an engaging platform for the users of BilConnect.

3. Use Case Diagram

The use case diagram of the app is described in Figure 3.1 with the conventions of UML.

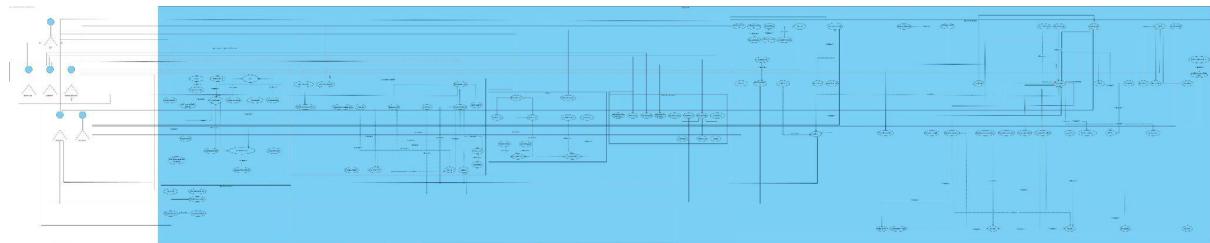


Figure 3.1: Use Case Diagram

1) Settings

This subsection is dedicated to explain the event flows of the use cases for settings.

- a) **Name:** DeleteAccount

Participating Actor: Initiated by *User*

Entry Condition:

- *User* clicks on the “Delete Account” button on the settings menu.

Exit Condition:

- Account successfully deleted.

Flow of Events:

- 1) *User* presses “Delete Account” button.
- 2) *User* confirms the deletion by pressing “Confirm” button.
- 3) A message screen confirming deletion is shown to the *User*.

Special/Quality Requirements: None.

- b) **Name:** EditAccount

Participating Actor: Initiated by *User*

Entry Condition:

- *User* enters setting menu from a dropdown menu on the main page.

Exit Condition:

- *User successfully changes account details.*

Flow of Events:

- 1) *User enters the settings menu.*
- 2) *User can toggle the page between profile photo and password change screens.*
- 3) *User can modify these screens to modify account details.*
- 4) *User confirms the changes using the current password of the account and with a “Confirm” button.*

Special/Quality Requirements: The current password must be correct. The new password must meet the password requirements. The new password cannot be the same with the old password. The passwords must be censored.

Additionally, the new photo must be in correct file format (.png) and up to 2048x2048px in quality. It must be cropped to be a square.

- c) **Name:** ChangeLanguage

Participating Actor: Initiated by User

Entry Condition:

- *User opens the right-top drop-box on the main menu and hovers on the English-Turkish switch.*

Exit Condition:

- *User stops hovering on the switch.*

Flow of Events:

- 1) *User press the switch to toggle the site-wide language from English to Turkish or vice versa.*

Special/Quality Requirements: None

- d) **Name:** ToggleDarkMode

Participating Actor: Initiated by User

Entry Condition:

- *User opens the right-top drop-box on the main menu and hovers on the Dark Mode switch.*

Exit Condition:

- *User* stops hovering on the switch.

Flow of Events:

- 1) *User* press the switch to toggle the color theme from dark to white or vice versa.

Special/Quality Requirements: None

2) Moderation Services

This subsection is dedicated to explain the event flows of the use cases for moderation services.

- a) **Name:** DeleteClubEventByPrivilege

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses the “Delete Event” button on event details.

Exit Condition:

- *Moderator* successfully deletes the event.

Flow of Events:

- 1) *Moderator* presses the “Delete Event” button.
- 2) *Moderator* confirms the deletion.

Special/Quality Requirements: None.

- b) **Name:** ViewChatLog

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses “View Chat Log” button in a post reporting another user.

Exit Condition:

- *Moderator* closes the chat window.
- *Moderator* resolves the report.

Flow of Events:

- 1) *Moderator* opens a User Report.
- 2) *Moderator* presses “View Chat Log” button.

- 3) *Moderator* is shown the chat log of the reporter and the reported.

Special/Quality Requirements: None.

- c) **Name:** ResolveReport

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses “Resolve Report” button.

Exit Condition:

- *Moderator* successfully resolves the report.

Flow of Events:

- 1) *Moderator* presses “Resolve Report” button on any kind of report.
- 2) *Moderator* confirms the report is resolved.
- 3) The report is removed from the report list and resolved.

Special/Quality Requirements: None.

- d) **Name:** DeletePostByPrivilege

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses “Delete Post” button in the post details.

Exit Condition:

- *Moderator* successfully deletes the post.

Flow of Events:

- 1) *Moderator* presses the “Delete Post” button.
- 2) *Moderator* confirms the deletion.

Special/Quality Requirements: None.

- e) **Name:** SuspendAccount

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses “Suspend User” button in a post reporting another user.

Exit Condition:

- *Moderator* successfully suspends the account.

Flow of Events:

- 1) *Moderator* presses the “Suspend User” button.
- 2) *Moderator* confirms the suspension.

Special/Quality Requirements: None.

f) **Name:** ViewReports

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* presses the exclamation mark button on the main page.

Exit Condition:

- *Moderator* views the details of a report.

Flow of Events:

- 1) *Moderator* enters the report list.
- 2) *Moderator* clicks on an individual report from the list.
- 3) *Moderator* views the details of the report.

Special/Quality Requirements: None.

g) **Name:** CreateTag

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* enters the “Create Tags” menu from the main page.

Exit Condition:

- *Moderator* successfully creates a new tag.

Flow of Events:

- 1) *Moderator* enters the “Create Tags” menu from the main page.
- 2) *Moderator* enters the name of the new tag.

- 3) *Moderator* indicates the tag as primary or secondary.
- 4) *Moderator* confirms the new tag.

Special/Quality Requirements: None.

- h) **Name:** RemoveTag

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* enters the “Remove Tags” menu from the main page.

Exit Condition:

- *Moderator* successfully removes the tag.

Flow of Events:

- 1) *Moderator* enters the “Remove Tags” menu from the main page.
- 2) *Moderator* selects the tag to be removed from the list.
- 3) *Moderator* presses “Delete” button.
- 4) *Moderator* confirms the deletion.

Special/Quality Requirements: None.

- i) **Name:** EditTag

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* enters the “Edit Tags” menu from the main page.

Exit Condition:

- *Moderator* successfully edits the tag.

Flow of Events:

- 1) *Moderator* enters the “Edit Tags” menu from the main page.
- 2) *Moderator* selects the tag to be edited from the list.
- 3) *Moderator* can change the primary-secondary status or name of the tag.
- 4) *Moderator* confirms the changes to the tag.

Special/Quality Requirements: None.

j) **Name:** ViewAllPosts

Participating Actor: Initiated by Moderator

Entry Condition:

- *Moderator* uses the search functionality.

Exit Condition:

- *Moderator* views the details of a post.

Flow of Events:

- 1) *Moderator* uses the search functionality.
- 2) *Moderator* is presented with posts that fit the search requirements that include removed posts (unlike search by *Second-hand user*).

Special/Quality Requirements: None.

3) Club Event Calendar

This subsection is dedicated to explain the event flows of the use cases for the club event calendar.

a) **Name:** ViewEventDetail

Participating Actor: Initiated by User

Entry Condition:

- *User* selects an event from the day version of the event calendar.

Exit Condition:

- *User* returns to the event calendar using the back button.

Flow of Events:

- 1) *User* enters event details.
- 2) *User* can leave event details using the back button.

Special/Quality Requirements: None.

b) **Name:** ViewEventCalendarbyMonth

Participating Actor: Initiated by User

Entry Condition:

- *User* clicks on the event calendar in the main page

Exit Condition:

- *User* views a specific day.

Flow of Events:

- 1) *User* enters the Event calendar page.
- 2) *User* can use arrows to navigate to months in the future and the past.
- 3) *User* can choose a specific day to view in detail.

Special/Quality Requirements: None.

c) **Name:** ViewEventCalendarbyDay

Participating Actor: Initiated by User

Entry Condition:

- *User* clicks on a specific day to view in detail from the month form calendar.

Exit Condition:

- *User* views event details.

Flow of Events:

- 1) *User* enters the Event calendar page in day form.
- 2) *User* clicks on an event.

Special/Quality Requirements: None.

d) **Name:** EditClubEvent

Participating Actor: Initiated by *Student Club*

Entry Condition:

- *Student Club* presses the “Edit Event” button when viewing their own event posts.

Exit Condition:

- *Student Club* successfully edits the event.

Flow of Events:

- 1) *Student Club* enters edit screen.
- 2) *Student Club* edits at least one of event name, place, time or detail.
- 3) *Student Club* confirms the changes.

Special/Quality Requirements: None.

e) **Name:** CreateClubEvent

Participating Actor: Initiated by *Student Club*

Entry Condition:

- *Student Club* presses the “Create New Event” button on the main page.

Exit Condition:

- *Student Club* successfully posts event.

Flow of Events:

- 1) *Student Club Manager* selects “Create New Event” on the main page.
- 2) *Student Club Manager* designates the time and place for the event as well as an explanation of the event.
- 3) *Student Club Manager* presses the confirmation button.

Special/Quality Requirements: None.

f) **Name:** DeleteClubEvent

Participating Actor: Initiated by *Student Club*

Entry Condition:

- *Student Club* presses the “Delete Event” button when viewing their own event posts.

Exit Condition:

- *Student Club* successfully deletes the event.

Flow of Events:

- 1) *Student Club* presses the “Delete Event” button.
- 2) *Student Club* presses “Confirm Deletion” on the confirmation window.
- 3) The system confirms the deletion of the event with a pop-up message.

Special/Quality Requirements: None.

4) Second-hand Sales

This subsection is dedicated to explain the event flows of the use cases for second-hand sales services.

a) **Name:** PostViewerViewChat

Participating Actor: Initiated by *Post Viewer*

Entry Condition:

- Through the chat list, *Post Viewer* accesses the chat of a post they do not own.

Exit Condition:

- *Post Viewer* finalizes the talks about a post.

Flow of Events:

- 1) *Post Viewer* selects a chat to view.
- 2) *Post Viewer* can send and receive messages through this channel with the post owner.
- 3) *Post Viewer* either choose to accept the post and finalize the talk, or not accept the offer.

Special/Quality Requirements: A maximum latency of 5 seconds is considered for live messaging.

b) **Name:** SeeUserChatList

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* presses the envelope button from the top left of the main page.

Exit Condition:

- *Second-hand User* selects a chat to view in detail.

Flow of Events:

- 1) *Second-hand User* enters the chat list page.
- 2) *Second-hand User* can view all their chats that are finalized or not.
- 3) *Second-hand User* can select a chat to view in detail.

Special/Quality Requirements: None.

c) **Name:** ViewOwnPosts

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* presses “View Own Posts” from a dropdown menu on the right.

Exit Condition:

- *Post Author* selects a post to view in detail.

Flow of Events:

- 1) *Post Author* enters the page that contains a list of their posts.
- 2) *Post Author* can select one of these posts to view in detail.

Special/Quality Requirements: None.

d) **Name:** RateUser

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* votes the user they are chatting with out of 5 stars.

Exit Condition:

- *Second-hand User* vote is successfully submitted.

Flow of Events:

- 1) *Second-hand User* selects stars out of five to vote for the user they are chatting with.
- 2) *Second-hand User* presses the submit button.

Special/Quality Requirements: None.

e) **Name:** StartChat

Participating Actor: Initiated by *Post Viewer*

Entry Condition:

- *Post Viewer* presses the envelope button in the details of a post of another *Post Author*.

Exit Condition:

- The chat is successfully created and *Post Viewer* enters the newly created chat.

Flow of Events:

- 1) *Post Viewer* presses the envelope button in the details of a post.
- 2) *Post Viewer* is redirected to the chat created with the posts owner.

Special/Quality Requirements: None.f) **Name:** PostOwnerViewChat**Participating Actor:** Initiated by *Post Author***Entry Condition:**

- Through the chat list, *Post Author* accesses the chat of a post they own.

Exit Condition:

- *Post Author* finalizes the talks about a post.

Flow of Events:

- 1) *Post Author* selects a chat to view.
- 2) *Post Author* can send and receive messages through this channel with the post owner.
- 3) *Post Author* finalizes the post's lifespan, removing it from the site and designating it as sold.

Special/Quality Requirements: A maximum latency of 5 seconds is considered for live messaging.g) **Name:** ViewSearchedPosts**Participating Actor:** Initiated by *Post Viewer***Entry Condition:**

- *Post Viewer* uses the search functionality.

Exit Condition:

- *Post Viewer* views the details of a post.

Flow of Events:

- 1) *Post Viewer* uses the search bar located at the top mid with any combination of primary, secondary tags and prompts.

- 2) Posts related to these are listed for *Post Viewer*.
- 3) *Post Viewer* can choose one of these posts.

Special/Quality Requirements: A latency of less than 7 seconds will be tolerated.

- h) **Name:** MarkAsFavoritePost

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* enters the details page of the post they want to favorite.

Exit Condition:

- *Second-hand User* successfully adds the post to their favorites.

Flow of Events:

- 1) *Second-hand User* presses the hearth-shaped button on the post details page.

Special/Quality Requirements: None.

- i) **Name:** DeletePost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* presses “Delete Post” button on the post details page.

Exit Condition:

- *Post Author* successfully deletes the post.

Flow of Events:

- 1) *Post Author* presses “Delete Post” button on the post details page.
- 2) *Post Author* presses “Confirm” on the confirmation pop-up.

Special/Quality Requirements: None.

- j) **Name:** EditPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* presses “Edit Post” button on the post details page.

Exit Condition:

- *Post Author* successfully edits the post.

Flow of Events:

- 1) *Post Author* presses “Edit Post” button on the post details page.
- 2) *Post Author* is redirected to a screen similar to post creation screen.
- 3) *Post Author* changes the variables of the post such as name and price.
- 4) *Post Author* presses confirm, finalizing the edit.

Special/Quality Requirements: None.

k) **Name:** CreateDonationPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Donation Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Donation Post as their primary tag and sets the attribute primary tag as Donation Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -description, title, secondary tag- that are required to present a Donation Post.
- 3) *Post Author* enters necessary fields to complete the donation post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

l) **Name:** CreateBorrowingPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Borrowing Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Borrowing Post as their primary tag and sets the attribute primary tag as Borrowing Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -title, description, secondary tag- that are required to present a Borrowing Post.
- 3) *Post Author* enters necessary fields to complete the borrowing post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

m) **Name:** CreateTravellingPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Travelling Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Travelling Post as their primary tag and sets the attribute primary tag as Travelling Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -departure place, destination place, time of the day that

travel takes place, maximum number of people that may travel in the particular trip- that are required to present a Travelling Post.

- 3) *Post Author* enters necessary fields to complete the travelling post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

- n) **Name:** CreateSecondHandItemPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Second Hand Item Sale Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Second Hand Item Sale Post as their primary tag and sets the attribute primary tag as Second Hand Item Sale Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -title, description, secondary tag, price information- that are required to present a Second Hand Sale Item Post.
- 3) *Post Author* enters necessary fields to complete the second hand item sale post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

- o) **Name:** CreatePetAdoptionPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Pet Adoption Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Pet Adoption Post as their primary tag and sets the attribute primary tag as Donation Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -description, title, secondary tag- that are required to present a Pet Adoption Post.
- 3) *Post Author* enters necessary fields to complete the pet adoption post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

- p) **Name:** CreateLostItemPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Lost Item Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Lost Item Post as their primary tag and sets the attribute primary tag as Lost Item Post.

- 2) BilConnect responds by presenting a UI that shows necessary fields -description, title, secondary tag- that are required to present a Lost Item Post.
- 3) *Post Author* enters necessary fields to complete the lost item post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

- q) **Name:** CreateEventTicketPost

Participating Actor: Initiated by *Post Author*

Entry Condition:

- *Post Author* is logged into the Bilconnect chooses Create Event Ticket Post as primary tag on the Create Post UI.

Exit Condition:

- *Post Author* finishes the creating post process.

Flow of Events:

- 1) *Post Author* chooses Create Event Ticket Post as their primary tag and sets the attribute primary tag as Event Ticket Post.
- 2) BilConnect responds by presenting a UI that shows necessary fields -description, title, secondary tag, price, place, the time cinema, concert, etc. will take place- that are required to present an Event Ticket Post.
- 3) *Post Author* enters necessary fields to complete event ticket post creation. Then the *Post Author* presses the Create Post button.
- 4) Otherwise, if the *Post Author* changes their mind about creating a new post, they click on CancelButton to exit the Create Post UI.
- 5) BilConnect receives the post and displays it on the *User* feed.
- 6) Otherwise, BilConnect does not receive any inquiry about a post.

Special/Quality Requirements: None.

r) **Name:** ReportEvent

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* presses the exclamation mark in the event description.

Exit Condition:

- *Second-hand User* successfully submits the report.

Flow of Events:

- 1) *Second-hand User* fills the details of the report with optional documents for proof.
- 2) *Second-hand User* selects report category.
- 3) *Second-hand User* submits the report.

Special/Quality Requirements: None.

s) **Name:** ReportPost

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* presses the exclamation mark in the post description.

Exit Condition:

- *Second-hand User* successfully submits the report.

Flow of Events:

- 1) *Second-hand User* fills the details of the report with optional documents for proof.
- 2) *Second-hand User* selects report category.
- 3) *Second-hand User* submits the report.

Special/Quality Requirements: None.

t) **Name:** ReportUser

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* presses the exclamation mark in the chat with another *Second-hand*.

Exit Condition:

- *Second-hand User* successfully submits the report.

Flow of Events:

- 1) *Second-hand User* fills the details of the report with optional documents for proof.
- 2) *Second-hand User* selects report category.
- 3) *Second-hand User* submits the report.

Special/Quality Requirements: None.

u) **Name:** ViewMarkedAsFavoritePosts

Participating Actor: Initiated by *Post Viewer*

Entry Condition:

- *Post Viewer* presses “View Favorite Posts” from a dropdown menu on the right.

Exit Condition:

- *Post Viewer* selects a post to view in detail.

Flow of Events:

- 1) *Post Viewer* enters the page that contains a list of their favorited posts.
- 2) *Post Viewer* can select one of these posts to view in detail.

v) **Name:** ViewPostDetail

Participating Actor: Initiated by *Post Viewer*

Entry Condition:

- *Post Viewer* clicks on a post.

Exit Condition:

- *Post Viewer* return to the main page.
- *Post Viewer* starts a chat with the *Post Author*.

Flow of Events:

- 1) *Post Viewer* can see details of the post such as name and description.

Special/Quality Requirements: None.

5) Login/Logout/New Account Services

This subsection is dedicated to explain the event flows of the use cases for Login/Logout and new account registration services.

- a) **Name:** Forgot Password

Participating Actor: Initiated by *User*

Entry Condition:

- *User* selects the “Forgot my Password” tab in the login page.

Exit Condition:

- *User* successfully changes their password.

Flow of Events:

- 1) *User* selects the “Forgot my Password” tab in the login page.
- 2) *User* enters their e-mail and confirms.
- 3) *User* enters the confirmation code sent to their e-mail.
- 4) *User* enters new password and confirms.

Special/Quality Requirements: The security code will be 6 digits, all number. Additionally, password must be strong, and must have at least 8 characters. The registered email must end with “bilkent.edu.tr” and must not be registered before. A new email verification code can only be sent after 60 seconds.

- b) **Name:** Login

Participating Actor: Initiated by *User*

Entry Condition:

- *User* enters the site domain.

Exit Condition:

- *Account* successfully enters their account.

Flow of Events:

- 1) *User* enters their account's email.
- 2) *User* enters their account's password.
- 3) Website directs the user to the home page of their account.

Special/Quality Requirements: The login process must be secured using HTTPS. Passwords should be stored encrypted and should never be displayed in any readable format.

- c) **Name:** CreateUserAccount

Participating Actor: Initiated by *Second-hand User*

Entry Condition:

- *Second-hand User* presses the sign up button in the log-in page.

Exit Condition:

- *Second-hand User* successfully opens an account.

Flow of Events:

- 1) *Second-hand User* enters the register page.
- 2) *Second-hand User* enters their account's username or email.
- 3) *Second-hand User* enters the verification code sent by email.
- 4) The website directs displays a message confirming successful account creation and redirects to the main page.

Special/Quality Requirements: Password must be strong, and must have at least 8 characters. The registered email must end with "bilkent.edu.tr" and must not be registered before. A new email verification code can only be sent after 60 seconds.

- d) **Name:** CreateClubAccount

Participating Actor: Initiated by *Student Club*

Entry Condition:

- *Student Club* presses the sign up button in the log-in page.

Exit Condition:

- *Student Club User* successfully opens an account.

Flow of Events:

- 1) *Student Club* enters the register page.
- 2) *Student Club* enters their account's username or email.

- 3) *Student Club* enters the verification code sent by email.
- 4) A *Moderator* checks the legitimacy of the club.
- 5) An e-mail is sent to the email used to register informing of registration success.

Special/Quality Requirements: Password must be strong, and must have at least 8 characters. The registered email must end with "bilkent.edu.tr" and must not be registered before. A new email verification code can only be sent after 60 seconds.

- e) **Name:** Logout

Participating Actor: Initiated by *User*

Entry Condition:

- *User* presses the logout button located at the top right dropdown menu.

Exit Condition:

- *User* successfully logs out.

Flow of Events:

- 1) *User* presses the logout button located at the top right dropdown menu.
- 2) *User* confirms they want to log out to the pop-out.

Special/Quality Requirements: None.

4. Class Diagram

The class diagram drawn with UML standards using Visual Paradigm is given in Figure 4.1.

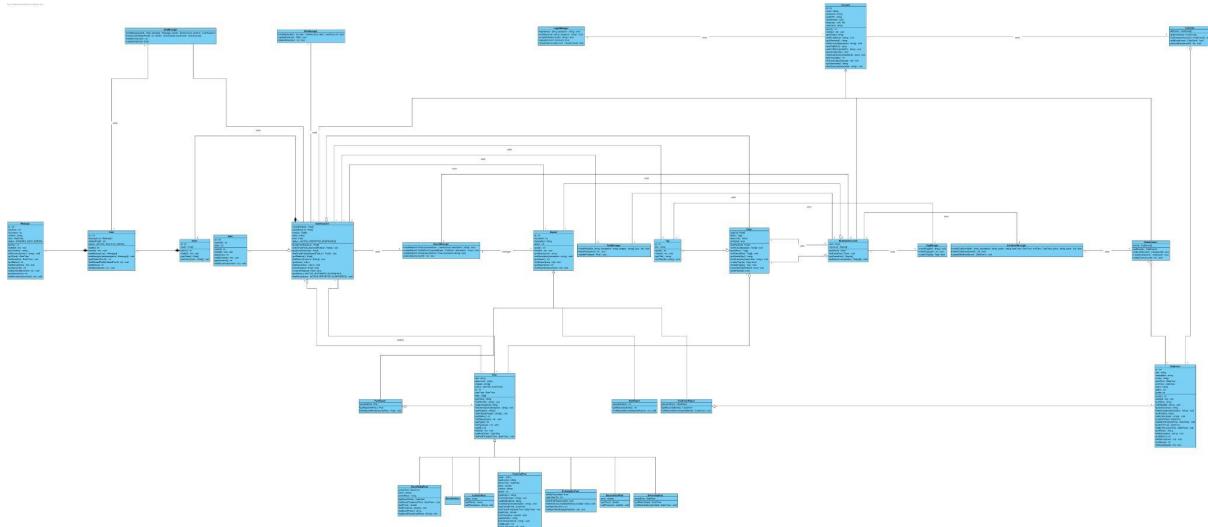


Figure 4.1: The class diagram showing the relations between the classes

The descriptions of the attributes, functions, and relations between the other classes in Figure 4.1 are given below.

Account: *Account* is an abstract class. *Account* class contains information common to all account types, such as ID, email, password, profile picture URL, username, dark mode preference, and language preference. Different account types inherit the *Account* class and implement different use cases accordingly. *UserAccount*, *ModeratorAccount*, and *ClubAccount* are children of the abstract *Account* class. *Account* class implements functions for reaching and changing information of accounts.

UserAccount: *UserAccount* is a subclass of *Account*. In addition to the inherited attributes from its parent class, it has attributes such as owned post list (container to store the posts owned by the account), saved post list (container to store the posts saved by the account), rate list (container to store the rates given to the account by the other accounts), inbox, and a personalized feed. It has getter and setter functions to fetch and mutate these attributes. Moreover, it can use *PostManager* class to create and manage posts, and *ReportManager* class to create and send reports.

ClubAccount: *ClubAccount* is a subclass of *Account*. In addition to the inherited attributes from its parent class, *ClubAccount* has an attribute to store events that are created by the student club that account belongs to. This class is designed for clubs to see their events on the calendar, create new events, and delete their events on the calendar.

ModeratorAccount: *ModeratorAccount* is a subclass of *Account*. In addition to the inherited attributes from its parent class, *ModeratorAccount* class stores a list of

reports created by *UserAccount* instances. *ModeratorAccount* also has a feed so that the moderator can supervise the activities of the instances of *UserAccount* directly. *ModeratorAccount* is a singleton class, so it will have only one instance. Using that instance, moderators can read reports, save reports as resolved, delete posts from feed and club events from the calendar, and suspend user accounts. Moderators are also responsible for adding new tags to be used and deleting tags that are already in use.

Feed: *Feed* is a container class to store the *Post* and *Tag* instances. Every *UserAccount* instance has a feed that can be personalized by following tags. It also has a search key attribute so that the posts might be filtered further. It also stores a boolean variable, *isFiltered*, to indicate if the filtering is turned on. It has getter and setter functions to fetch and mutate its attributes.

Post: *Post* is an abstract class with attributes title, description, a list of the URLs of the images it has, an enumerated status to keep the status of the post (such as active, inactive, deleted, etc.). It has some subclasses to create different instances of a post (such as second-hand sale, donation, request, etc.), and a unique id. It has getter and setter functions to fetch and mutate these attributes.

EventTicketPost: *EventTicketPost* is a class inherited from *Post*. It has extra attributes (place etc.) to specify a post that is a type of an event ticket.

DonationPost: *DonationPost* is a class inherited from *Post*. It specifies a post that is a type of a donation, basically similar with a second hand sale post with a zero price.

SecondHandPost: *SecondHandPost* is a class inherited from *Post*. It has extra attributes (price etc.) to specify a post that is a type of a second hand sale.

PetAdoptionPost: *PetAdoptionPost* is a class inherited from *Post*. It has extra attributes (vaccination status etc.) to specify a post that is a pet adoption announcement.

LostItemPost: *LostItemPost* is a class inherited from *Post*. It has extra attributes (place etc.) to specify a post that is a type of a lost item announcement.

TravelingPost: *TravelingPost* is a class inherited from *Post*. It has extra attributes (price, destination etc.) to specify a post that is a type of a traveling announcement to help drivers to find passengers to share the gas price.

BorrowingPost: *BorrowingPost* is a class inherited from *Post*. It has extra attributes (return date etc.) to specify a post that is a type of a borrowing announcement.

Calendar: *Calendar* class makes use of the singleton design pattern. Every user sees the same calendar. *Calendar* stores past and upcoming club events. The *Calendar* class has methods for reaching and manipulating club events on the calendar. *ModeratorAccount* and *ClubEventAccount* are responsible for manipulating the calendar.

ClubEvent: *ClubEvent* serves similarly to *Post*, with some differences. It has a unique id, a title, a description, a URL for a poster image, start and end times, place information, a quota, and an enumerated status (such as active, past, deleted, etc.), and getter & setter functions to fetch and mutate its attributes.

Inbox: Every *UserAccount* instance has one instance of *Inbox* object. This *Inbox* object stores its ID and chats for different posts (There might be more than one chat with the same user if the user is interested in more than one post).

Chat: It serves as a container for messages and additionally has a unique id, enumerated status and the id of the post that it is related to. It has getter and setter functions to fetch and mutate its attributes.

Message: The instances of the *Message* class are the messages sent by users to the other users in a chat. Every message instance holds its ID, senders' ID, receivers' ID, the content of the message, the sending time of the message, and an enumerated status (such as pending, sent, etc.). It also has getter and setter functions to manipulate its attributes.

Rate: The instances of the *Rate* class are the rates users give to the posts of other users. *Rate* class stores its ID, sender's ID, and star's value (1 to 5, 1 indicates worst experience, and 5 indicates best experience). Instances of rate class are stored in instances of the *UserAccount* class.

Tag: The instances of the *Tag* class are the tags that are used to filter the posts in a feed. It has a unique ID and a title, and getter & setter functions to view and manage its attributes.

Report: *Report* is an abstract class. It contains information common to all report subclasses, such as its ID, reporter's ID, description, and an enumerated status of the report (resolved, unresolved, etc.).

PostReport: *PostReport* is a subclass of the *Report* class. It is the report type dedicated to the problems with the posts. In addition to the attributes and functions inherited by its parent, it also stores the reported post and has getter and setter functions to manage it.

ClubEventReport: *ClubEventReport* is a subclass of the *Report* class. It is the report type dedicated to the problems with the club events. In addition to the attributes and functions inherited by its parent, it also stores the reported club event and has getter and setter functions to manage it.

UserReport: *UserReport* is a subclass of the *Report* class. It is the report type dedicated to the problems with the users. In addition to the attributes and functions inherited by its parent, it also stores the reported user and has getter and setter functions to manage it.

LoginManager: The *LoginManager* class is responsible for controlling account login, logout, and register operations. It has functions for login, logout, registering, and verifying emails. It also has a function to suspend user accounts, used by the moderator account.

PostManager: The *PostManager* class is responsible for controlling post sharing. It checks if the post is valid and creates the post if a user interacts with the UI element to create a post. If the post is invalid, *PostManager* does not share the post.

RateManager: The *RateManager* class is responsible for controlling user rating. When a user interacts with the UI elements to rate another user, the *RateManager* class acts as a controller class and sends the request to save the report in the database.

ClubEventManager: The *ClubEventManager* class is responsible for controlling club event sharing. It checks if the event is valid and creates the event if a student club interacts with the UI element to create an event. If the post is invalid, *ClubEventManager* does not share the event.

TagManager: The *TagManager* class is responsible for controlling tag creation. When the moderator interacts with the UI elements to create or delete a tag, the *TagManager* class acts as a controller class and sends the request to update the tag in the database.

ReportManager: The *ReportManager* class is responsible for controlling report sending and resolving operations. When a user interacts with the UI elements to send a report, it checks if the report is valid and saves the report in the database.

ChatManager: The *ChatManager* class is responsible for controlling chats and sending messages. When a user interacts with the UI elements to send a message, it manages all the operations to create a message, place it in a chat, and send it successfully.

5. Sequence Diagrams

In this section, four sequence diagrams are explained for a better understanding of the behaviors of the classes in BilConnect.

5.1 User Sends Report

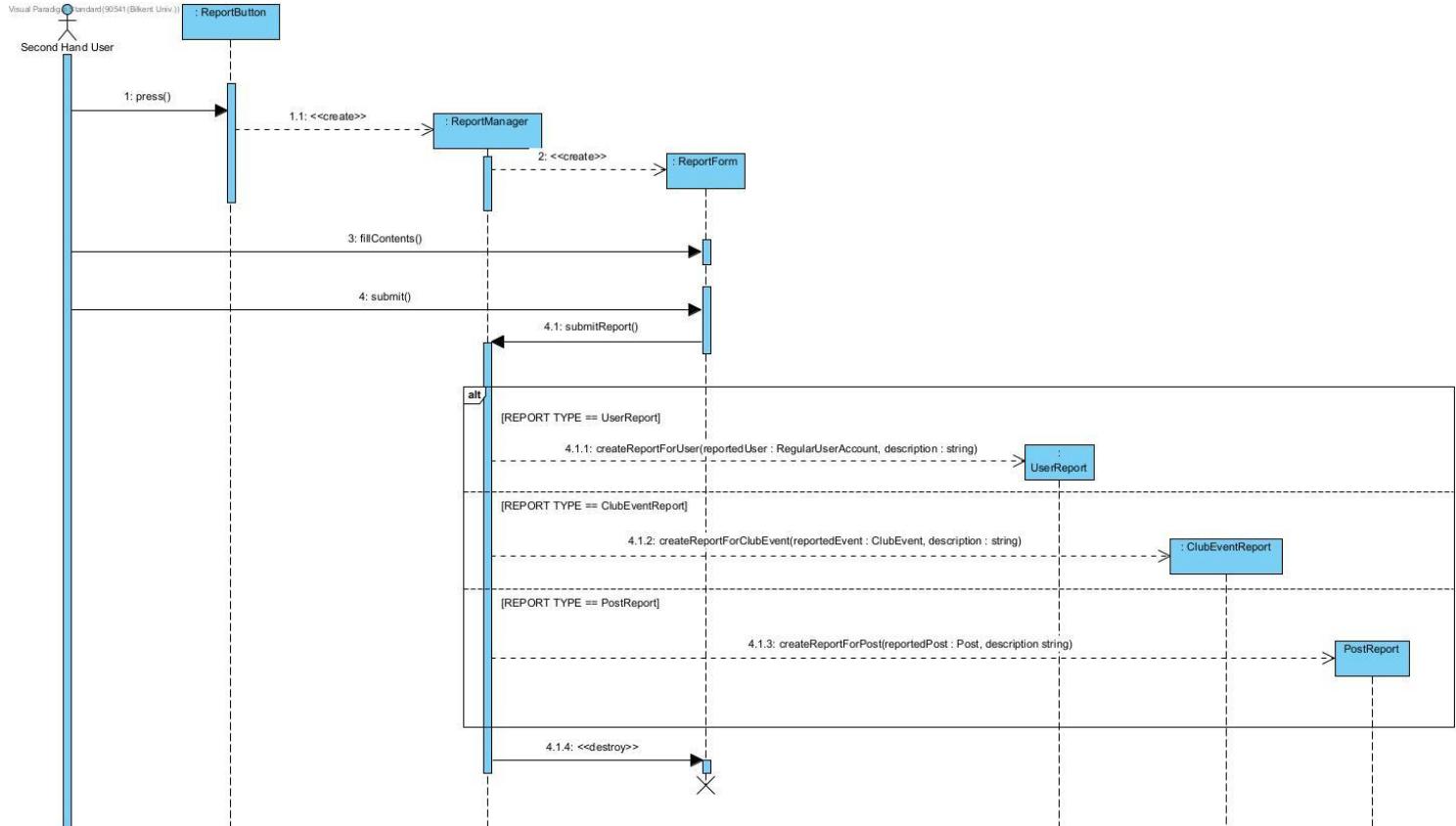


Figure 5.1: The sequence diagram describing a user sending a report

The user will press the report button next to an item (a post, a club event, or a user) they want to report. After the user presses the button, the *ReportManager* object will be created, and the *ReportManager* object will send a form for the user to fill out. The user will fill out the form and press the submit button. Depending on the type of report given (*UserReport*, *ClubEventReport*, *PostReport*), *ReportManager* will create the report by calling the appropriate function. Finally, *ReportManager* will destroy the report form because it is no longer in use.

5.2 User Shares Post

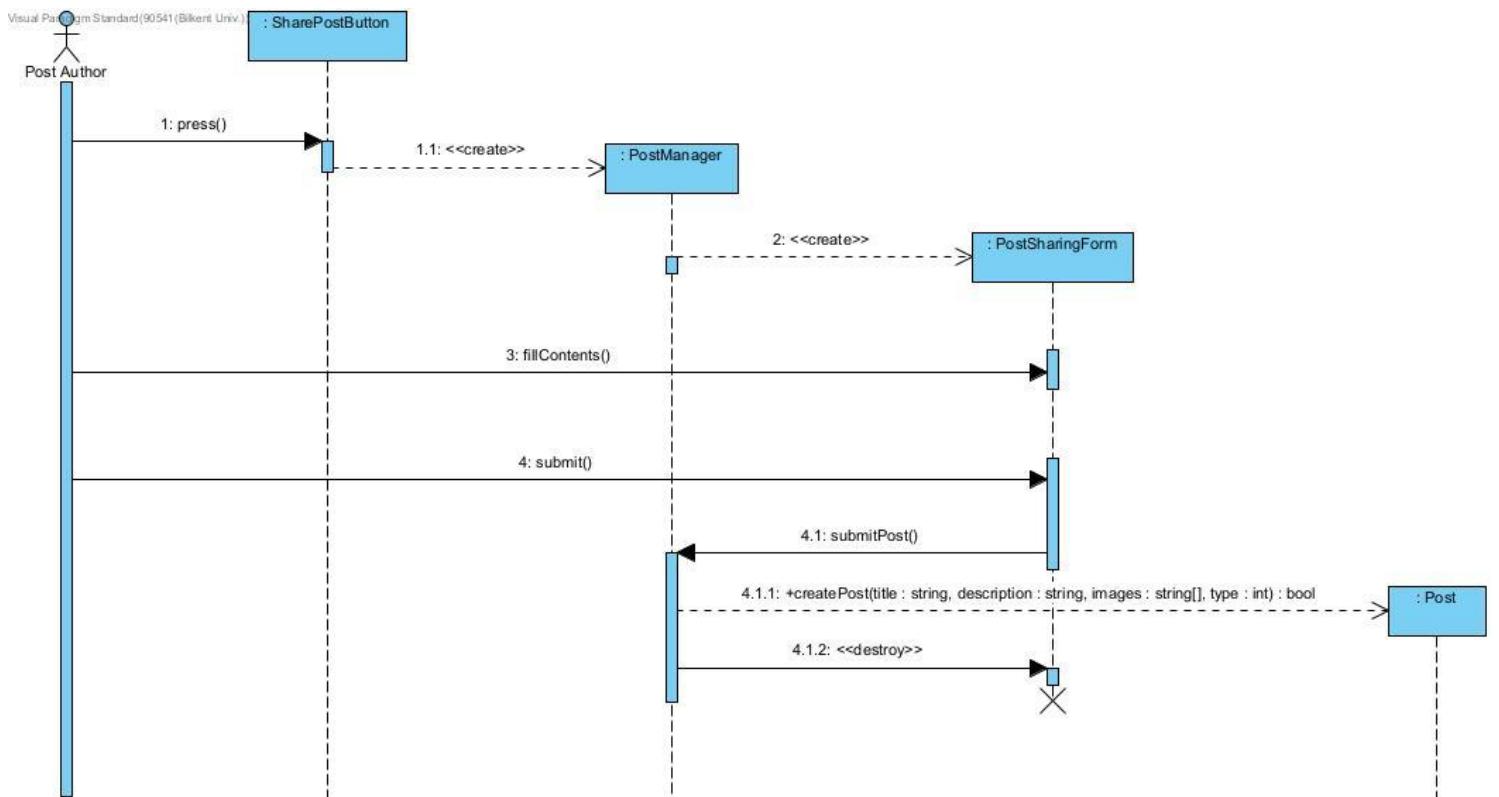


Figure 5.2: The sequence diagram describing a user sharing a post

The user will press the share post button to share a post. After the user presses the button, the *PostManager* object will be created, and the *PostManager* object will send a form for the user to fill out. The user will fill out the form, upload the pictures, write a title and description, and press the submit button. Then *PostManager* will create the post and save it to the system. If the post is invalid, *PostManager* will not share the post. After the post creation process, *PostManager* will destroy the post sharing form because it is no longer in use.

5.3 Moderator Resolves Report

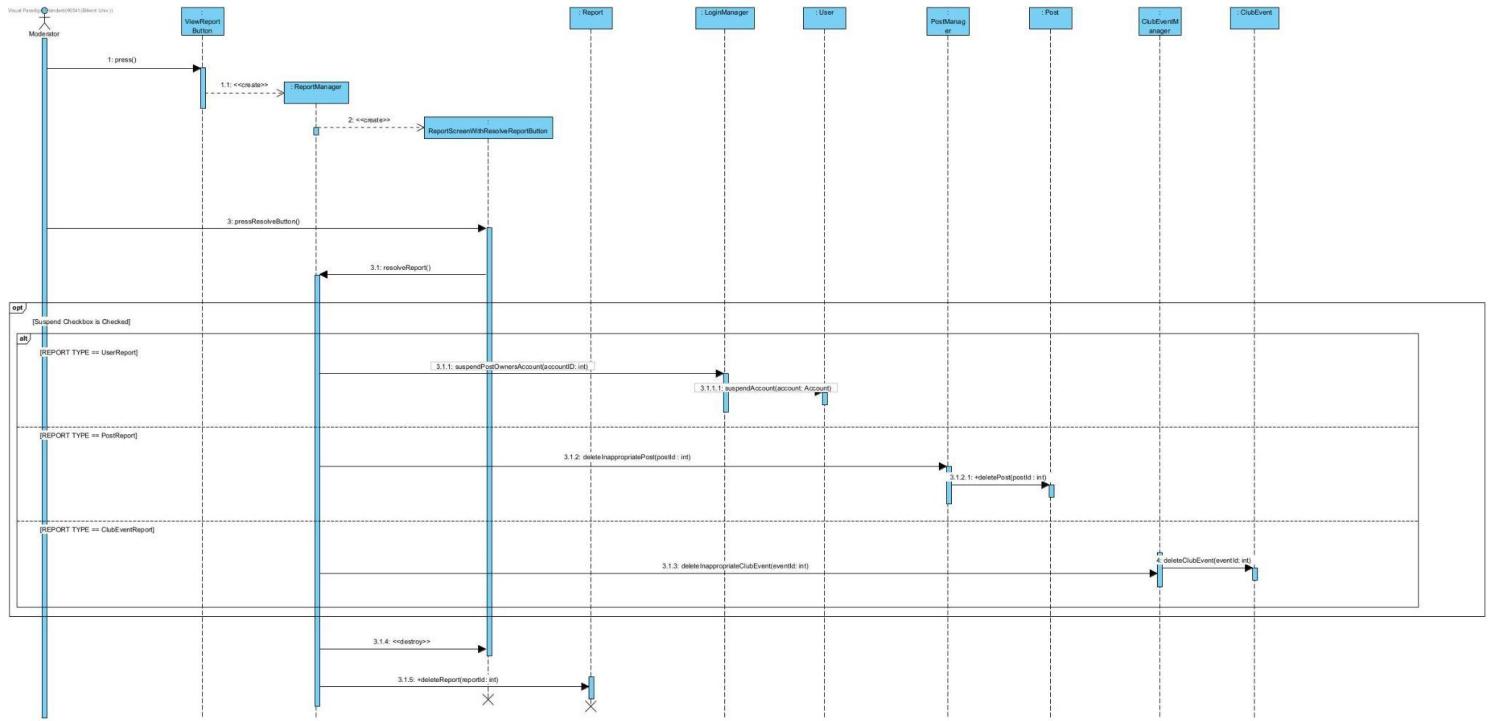


Figure 5.3: The sequence diagram describing the moderator resolving a report

In the reports screen, the moderator will select a report and press the “View Report” button to view the report. After pressing the button, the *ReportManager* object will be created, and then the moderator will be directed to a report resolving page. *ReportManager* object will show a UI element that contains the reported item and its description. The moderator will choose if the report is valid by either checking the checkbox or not then press the resolve report button. If the checkbox is checked depending on the type of report, the necessary manager will be notified and necessary action (suspending account, deleting post, deleting club event) will be taken. Finally *ReportManager* will destroy the report viewing screen and the report object because they are no longer in use.

5.4 Student Club Creates Club Event

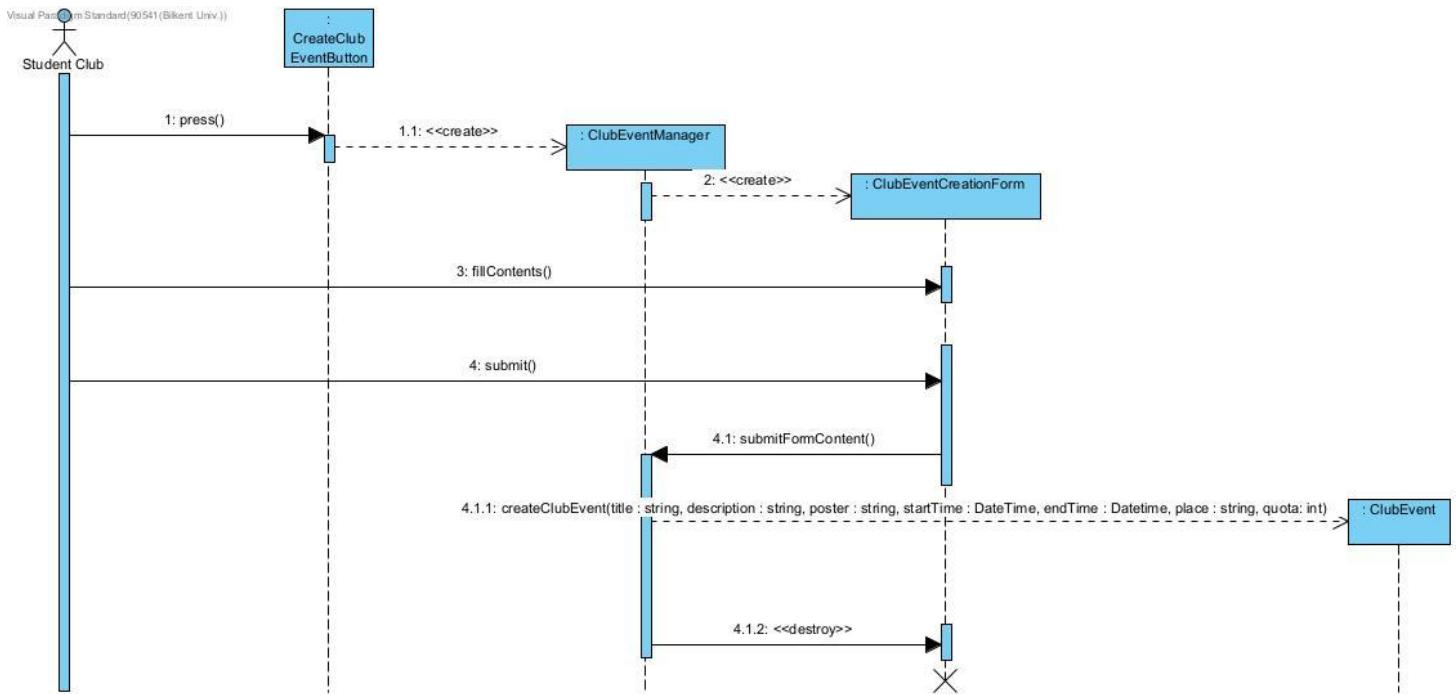


Figure 5.4: The sequence diagram describing a student club creating a club event

StudentClub will press the “Create Club Event” button to share a club event. After the student club presses the button, the *ClubEventManager* object will be created, and the *ClubEventManager* object will send a form for the user to fill out. The student club will fill out the given form by uploading the poster, writing a title, place, time, and description, and press the submit button. Then, *ClubEventManager* creates the event with information on form. Finally, *ClubEventManager* will destroy the form because it is no longer in use.

6. State Diagrams

6.1 State Diagram For Chat

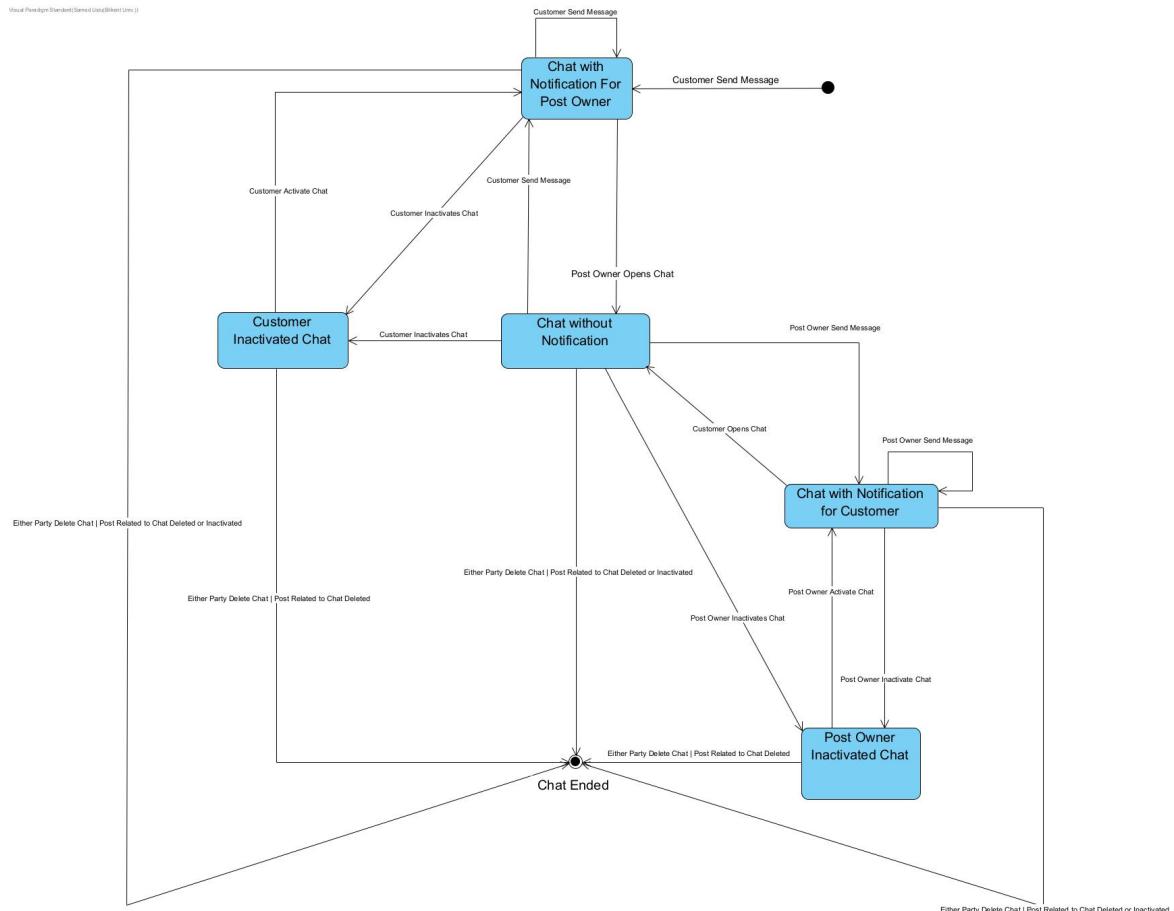


Figure 6.1: The state diagram describing the different states of the chat

This state diagram shows the main state transition for the chat feature of the application. The system handles chat initiation, notification, inactivation, and deletion through several states. Chat's life cycle begins after the customer sends the first message and the post owner is notified. Active chat state is divided into "*Chat with Notification for Post Owner*", "*Chat with Notification for Customer*" and "*Chat without Notification*" states. Depending on these states, chats representation in the inbox will be changed. In these states if either party decides to deactivate the chat they can do as long as they are not notified. Deactivate chat state is divided into "*Post Owner Inactivated Chat*" and "*Customer Inactivated Chat*". If the customer deactivates the chat only the customer can reactivate it. If the customer reactivates the chat, the post owner will be notified, same goes for the other way around. In all states, customers or post owners can delete the chat and if the post related to chat is deactivated or deleted, chat will also be deleted.

6.2 State Diagram For Post

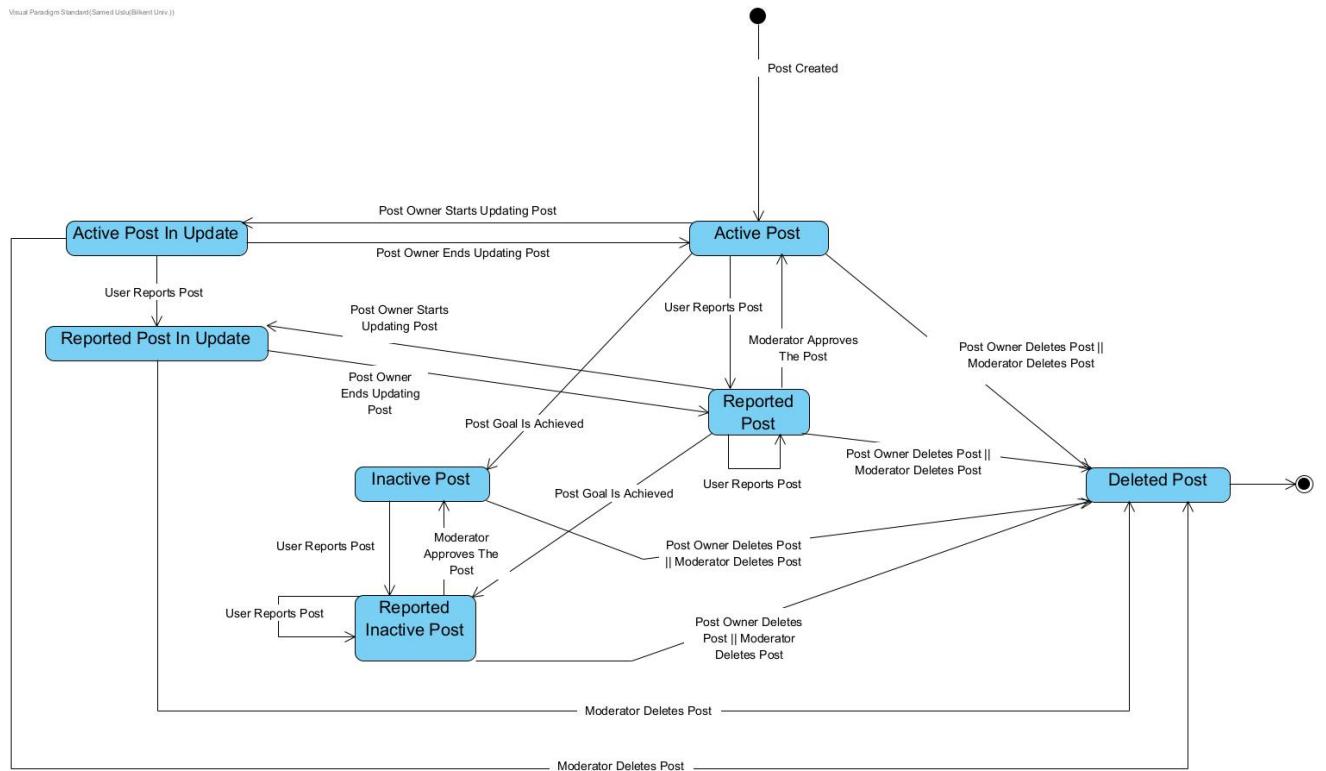


Figure 6.2: The state diagram describing the different states of the post

This state diagram shows the main state transition for the posts in the application. The initial state for the post is "*Active Post*". In this state of the post, the post can be deleted by the moderator and post owner, it can be updated and inactivated by the post owner, and another user can report it. When another user reports the "*Active Post*", it goes to the "*Reported Post*" state. Every "*Active Post*" operation can be applied to the post in this state. Additionally, the moderator can approve the post by resolving the issue to return to the "*Active Post*" state from this state. When the post owner starts to update the post, it stays in the "*Active Post In Update*" state until the moderator deletes the post or the post owner finishes the update. If the post reaches its goal, the "*Active Post*" state or "*Reported Post*" state goes to the "*Inactive Post*" or "*Reported Inactive Post*" state, which means the post is no longer open for new customers. Posts in the "*Inactive Post*" state can be reported and deleted. Posts in the "*Reported Inactive Post*" state can be reported, approved or deleted. When the post goes to the "*Deleted Post*" stage, it concludes its lifecycle.

6.3 State Diagram For Club Event

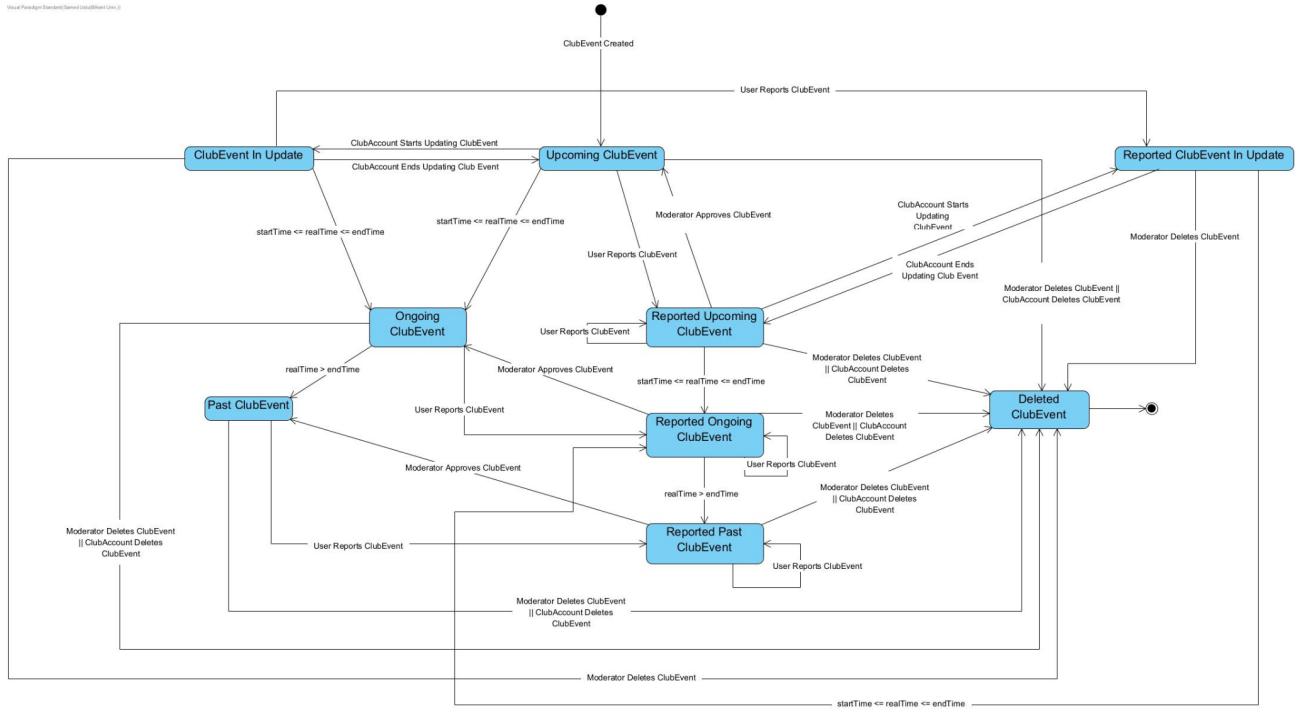


Figure 6.3: The state diagram describing the different states of the club event

This state diagram shows the main state transition for the club events in the application. The initial state for the club event is "*Upcoming ClubEvent*". In this state of the club event, the owner club account can update it, or the moderator or the owner club account can delete it. When the owner club account starts to update "*Upcoming ClubEvent*" or "*Reported Upcoming ClubEvent*", the club event goes to "*ClubEvent In Update*" or "*Reported ClubEvent In Update*" state. It stays in these states until the owner club account finishes the update operation or the time of the event comes. A user can report the club event in the "*Upcoming Club Event*", and the club event goes to the "*Reported Upcoming ClubEvent*" state. Every "*Upcoming ClubEvent*" operation can be applied to the club event in that state. In addition to the actions from the user, club events can change state with respect to time. When the time reaches the starting time of the "*Upcoming ClubEvent*" or the "*Reported Upcoming ClubEvent*", the event goes to the "*OngoingClubEvent*" or the "*Reported Ongoing ClubEvent*" state. When the time reaches the ending time of the "*Ongoing ClubEvent*" or the "*Reported Ongoing ClubEvent*", the event goes to the "*Past ClubEvent*" or the "*Reported Past ClubEvent*" state. The club event can be deleted in any of these states by the moderator or by the owner club account. When an event reaches the "*Deleted ClubEvent*" state, it reaches the final state.

7. Activity Diagrams

In this section, there are three activity diagrams explained briefly.

7.1 Student Club Creating Event

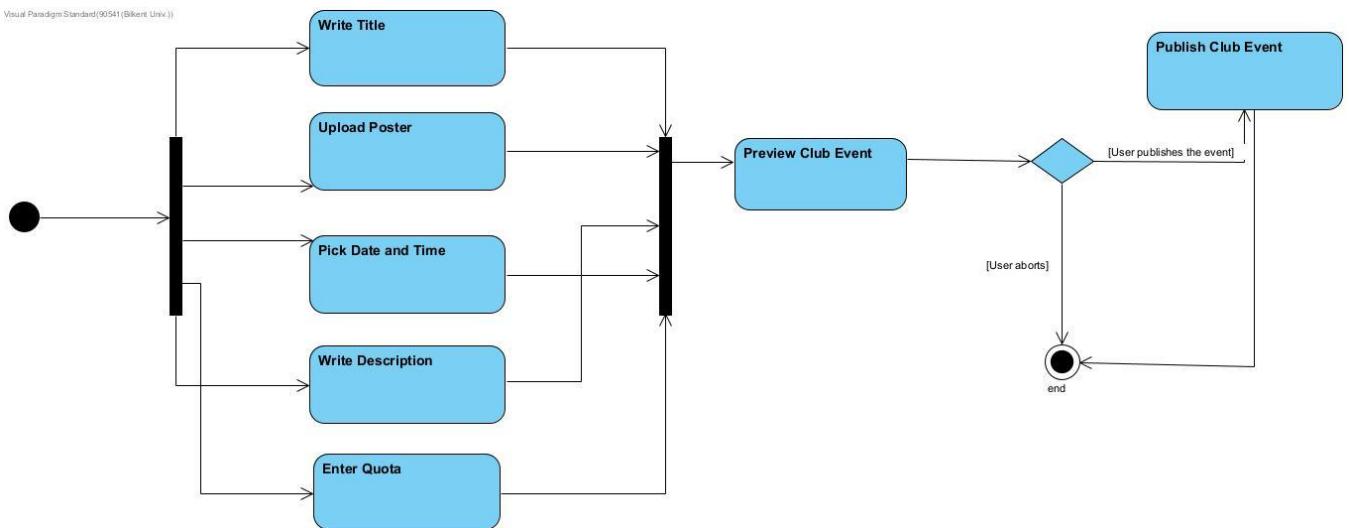


Figure 7.1: The Activity Diagram Representing *StudentClub*'s Event Creation

This diagram represents the process for adding a student club event. Date, time, event description, title, quota and poster information can be entered simultaneously. At any point in this process, the *Student Club* has the option to cancel the event and exit the action. Otherwise, *Student Club* can publish the event.

7.2 Report Evaluation for Moderators

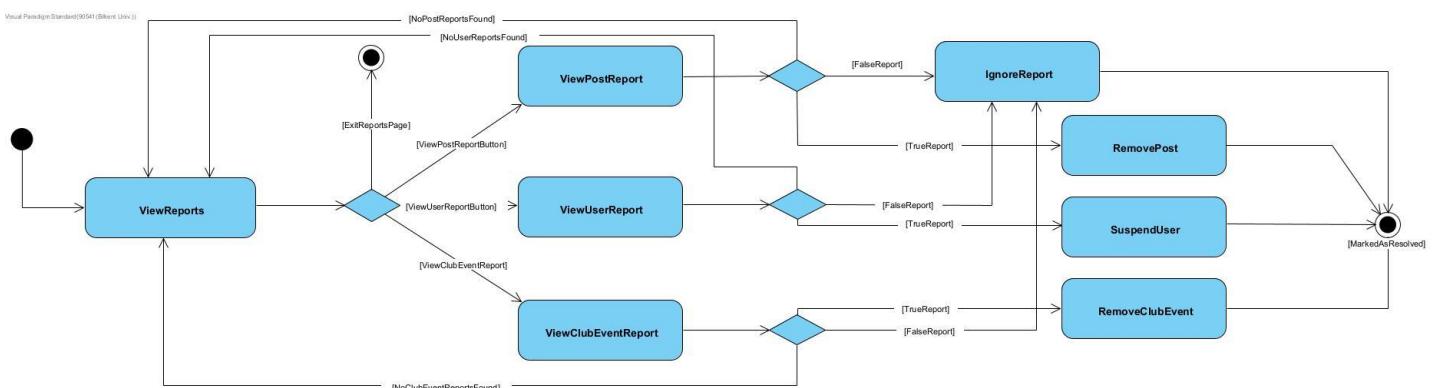


Figure 7.2: The Activity Diagram Representing Report Evaluation for Moderator

This diagram outlines the steps for handling reports. The action begins when a *Moderator* opens the reports page and views a list of categorized reports. The *Moderator* chooses a specific report type. In this action, the moderator can either ignore the report or mark it as resolved. Alternatively, depending on the report type, the *Moderator* may perform actions such as removing a post, suspending an account, or deleting a club event. The *Moderator* also has the option to exit the reports page from the action menu.

7.3 Post Creation

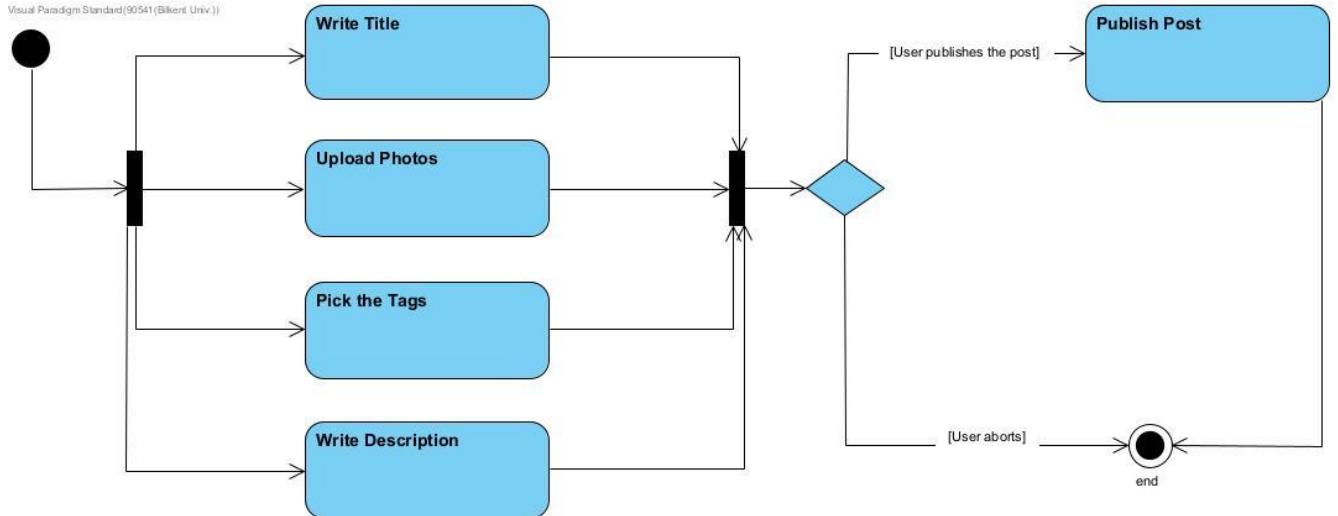


Figure 7.3: The Activity Diagram *Post Author's Representing Post Creation*

This diagram illustrates the process of creating a post. *Post Author* must complete various attributes -writing title, uploading photo/s, picking the tags, writing description-simultaneously when they initiate the post-creation process. Once all required information is provided, a *Post Author* can choose to publish the post. Otherwise, *Post Author* at any point can cancel the post creation process.

8. Mockups

Mockups are given in this section to give an insight into what BilConnect will look like. There are 48 figures with short descriptions in total.

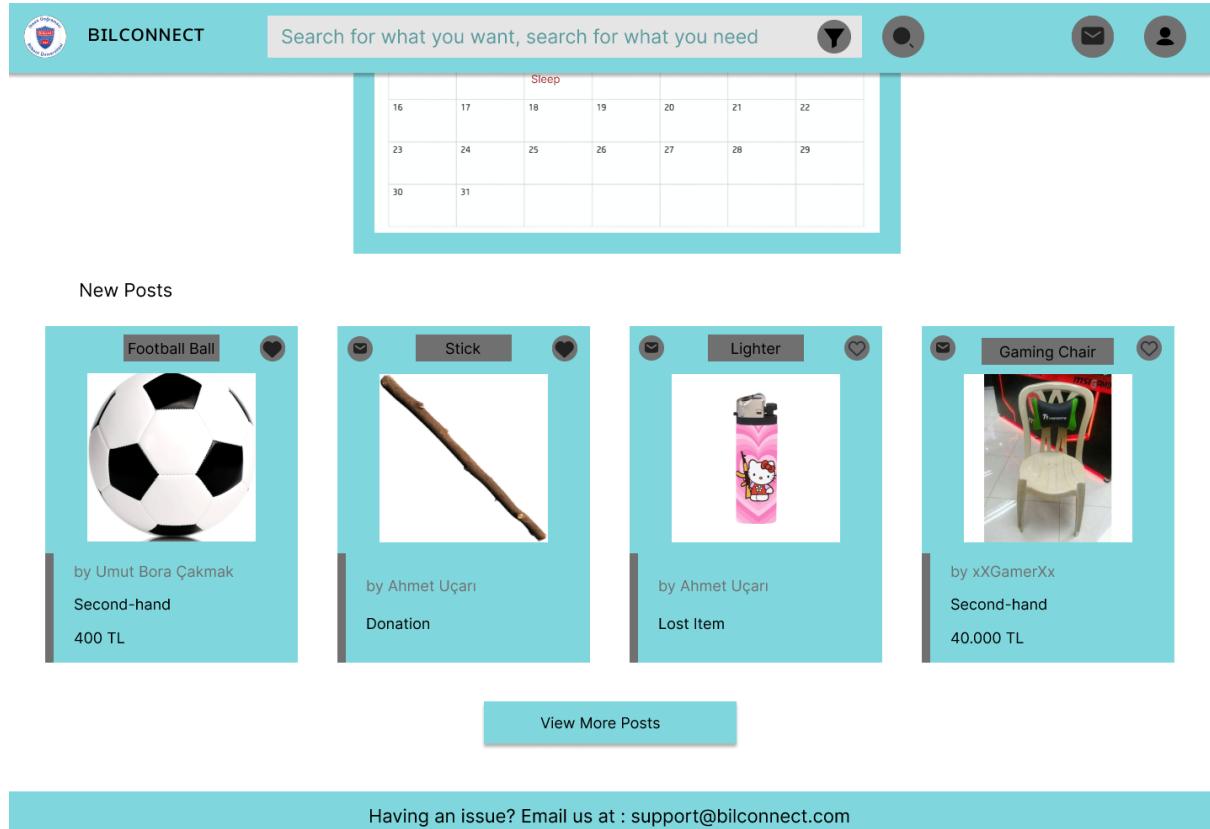


Figure 8.1: User Main Page

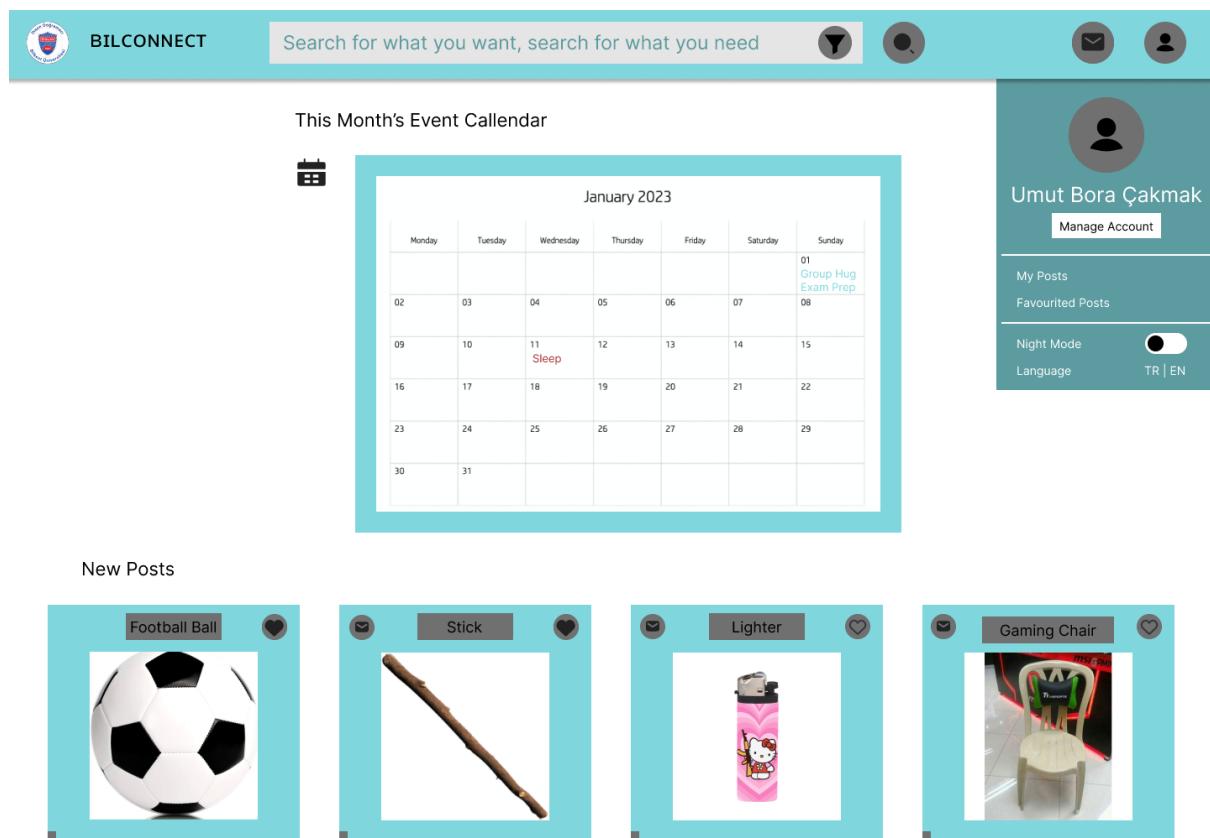


Figure 8.2: User Main Page displaying User Menu

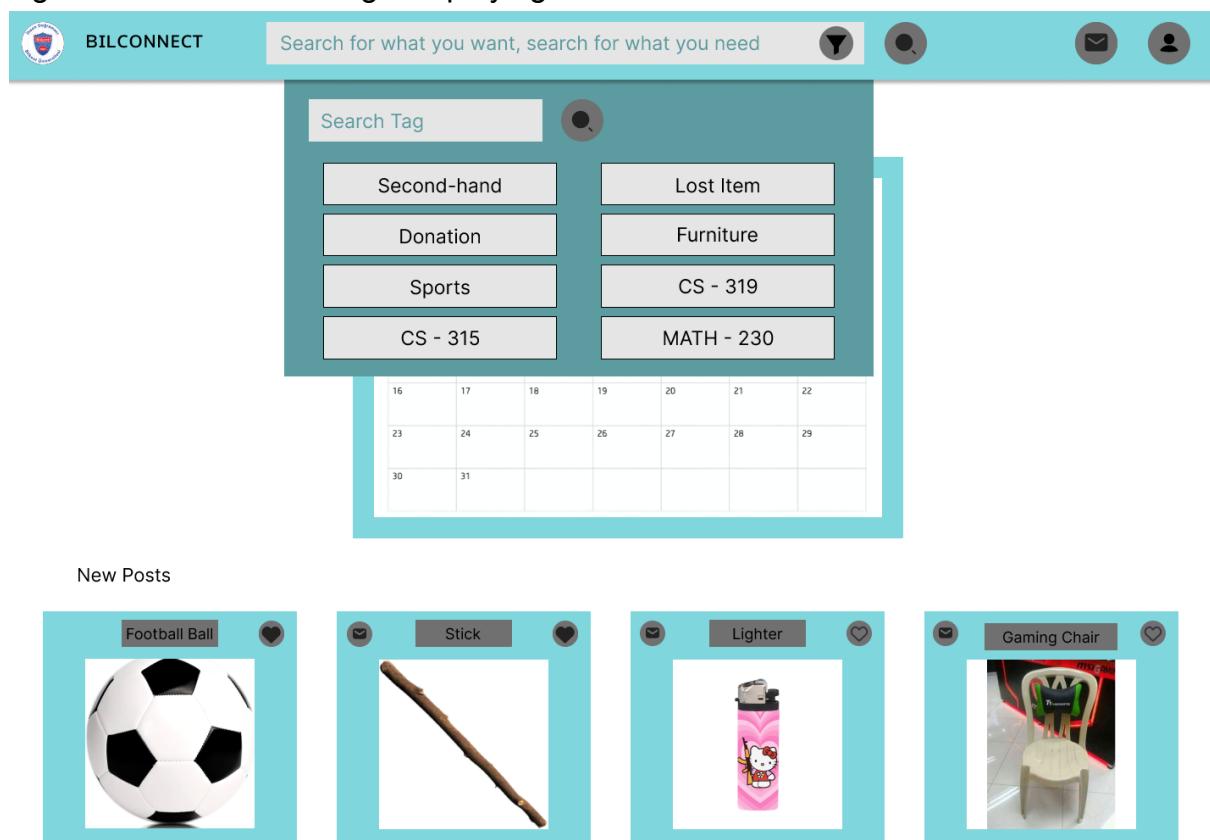


Figure 8.3: User Main Page displaying Filtering

 BILCONNECT Search for what you want, search for what you need    

Showing 1 result(s) for "chair"

Sort by: Date Ascending


Gaming Chair
 

by xXGamerXx
Second-hand
40.000 TL

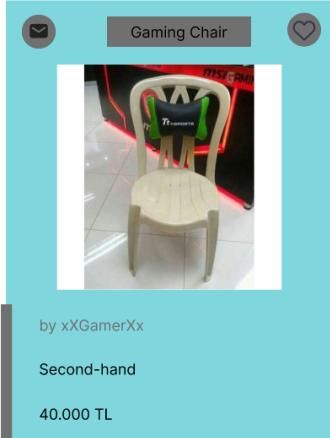
Having an issue? Email us at : support@bilconnect.com

Figure 8.4: Search with Prompt

 BILCONNECT Search for what you want, search for what you need    

Showing 2 result(s) for tag(s) Second-hand

Sort by: Price Descending


Gaming Chair
 

by xXGamerXx
Second-hand
40.000 TL


Football Ball


by Umut Bora Çakmak
Second-hand
400 TL

Having an issue? Email us at : support@bilconnect.com

Figure 8.5: Search with Tag

The screenshot shows a search result for a 'Gaming Chair'. At the top, there is a header bar with the BILCONNECT logo, a search bar containing the text 'Search for what you want, search for what you need', and three icons: a magnifying glass, a profile, and a message. Below the header, the title 'Gaming Chair' is displayed. Underneath the title are two category buttons: 'Second-hand' and 'Furniture'. The seller information is shown as 'Seller: xXGamerXx ★ ★ (3.0/5.0) 3 reviews'. The item details include 'Second-hand / Furniture' and a price of '40.000 TL'. The description states: 'Barely used gaming chair. It is in good condition. Head pillow in the photo is not included.' To the right of the description is a large image of a white wooden gaming chair with a green and black headrest pillow. To the far right of the screen are three circular icons: a heart, an envelope, and an exclamation mark.

Figure 8.6: Post Detail

The screenshot shows a post detail for a 'Football Ball'. At the top, there is a header bar with the BILCONNECT logo, a search bar containing the text 'Search for what you want, search for what you need', and three icons: a magnifying glass, a profile, and a message. Below the header, the title 'Football Ball' is displayed. Underneath the title are two category buttons: 'Second-hand' and 'Sports'. The seller information is shown as 'Seller: Umut Bora Çakmak ★ ★ ★ ★ ★ (5.0/5.0) 12 reviews'. The item details include 'Second-hand / Sports' and a price of '400 TL'. The description states: 'Football ball in good condition. Ideal for amateur usage.' To the right of the description is a large image of a black and white soccer ball. To the far right of the screen is a single circular icon with a heart symbol.

Figure 8.7: Post Detail of own Post

BILCONNECT Search for what you want, search for what you need

Mehmet Ulaş (Football Ball)
Ne yazık ki hayır. 21.04

Ahmet Uçarı (Stick)
Tabi ki alabilirsiniz ne... 16.04

Aslan Kaplan (Parrot)
We can talk about it. 13.16

John Williams (Coal)
Nope that doesn't fit me. 11.42

Zengin Mengin (Gold)
Sure where should... 12.10.2023

Mach Anderson (Phone)
Sorry. 11.10.2023

Tyrannosaurus Rex (Meat)
I will gladly buy mor... 23.09.2023

Football Ball Give to this user ! 400 TL

Remember to be respectful and adhere to site rules.

Merhaba. 21.03

Acaba 320 TL size uyar mı? 21.03

Ne yazık ki hayır. 21.04

Having an issue? Email us at : support@bilconnect.com

Figure 8.8: Chat with a buyer

BILCONNECT Search for what you want, search for what you need

Mehmet Ulaş (Football Ball)
Ne yazık ki hayır. 21.04

Ahmet Uçarı (Stick)
Tabi ki alabilirsiniz ne... 16.04

Aslan Kaplan (Parrot)
We can talk about it. 13.16

John Williams (Coal)
Nope that doesn't fit me. 11.42

Zengin Mengin (Gold)
Sure where should... 12.10.2023

Mach Anderson (Phone)
Sorry. 11.10.2023

Tyrannosaurus Rex (Meat)
I will gladly buy mor... 23.09.2023

Football Ball ! 400 TL

Remember to be respectful and adhere to site rules.

Merhaba. 21.03

Acaba 320 TL size uyar mı? 21.03

Ne yazık ki hayır. 21.04

O zaman 400TL, yarın 17.00, EE binası olur mu? 21.03

Evet olur. 21.04

You have finished this transaction.
If this was a mistake, [please click here](#) to revert this.

Having an issue? Email us at : support@bilconnect.com

Figure 8.9: End chat with buyer, designating them as the final buyer

BILCONNECT Search for what you want, search for what you need

Stick ! Donation

Mehmet Ulaş (Football Ball)
Ne yazık ki hayır. 21.04

Ahmet Uçarı (Stick)
Tabi ki alabilirsiniz ne... 16.04

Aslan Kaplan (Parrot)
We can talk about it. 13.16

John Williams (Coal)
Nope that doesn't fit me. 11.42

Zengin Mengin (Gold)
Sure where should... 12.10.2023

Mach Mach Anderson (Phone)
Sorry. 11.10.2023

Tyrannosaurus Rex (Meat)
I will gladly buy mor... 23.09.2023

Remember to be respectful and adhere to site rules.

Bu çubuğu ben alabilir miyim. 11.42

Tabiki alabilirsiniz nerede ne zaman buluşalım? 16.04

Having an issue? Email us at : support@bilconnect.com

Figure 8.10: Chat with a seller

BILCONNECT Search for what you want, search for what you need

Stick ! Donation

Mehmet Ulaş (Football Ball)
Ne yazık ki hayır. 21.04

Ahmet Uçarı (Stick)
Tabi ki alabilirsiniz ne... 16.04

Aslan Kaplan (Parrot)
We can talk about it. 13.16

John Williams (Coal)
Nope that doesn't fit me. 11.42

Zengin Mengin (Gold)
Sure where should... 12.10.2023

Mach Mach Anderson (Phone)
Sorry. 11.10.2023

Tyrannosaurus Rex (Meat)
I will gladly buy mor... 23.09.2023

Remember to be respectful and adhere to site rules.

Bu çubuğu ben alabilir miyim? 11.42

Tabiki alabilirsiniz nerede ne zaman buluşalım? 16.04

Yarın 15.00, kütüphane önü. 11.42

Ahmet Uçarı designated you as the buyer. Please rate your experience with the user.

If you believe this was an error, [please click here](#) to notify the seller.

Having an issue? Email us at : support@bilconnect.com

Figure 8.11: Chat end with seller, rating the seller



Figure 8.12: Settings Main Menu

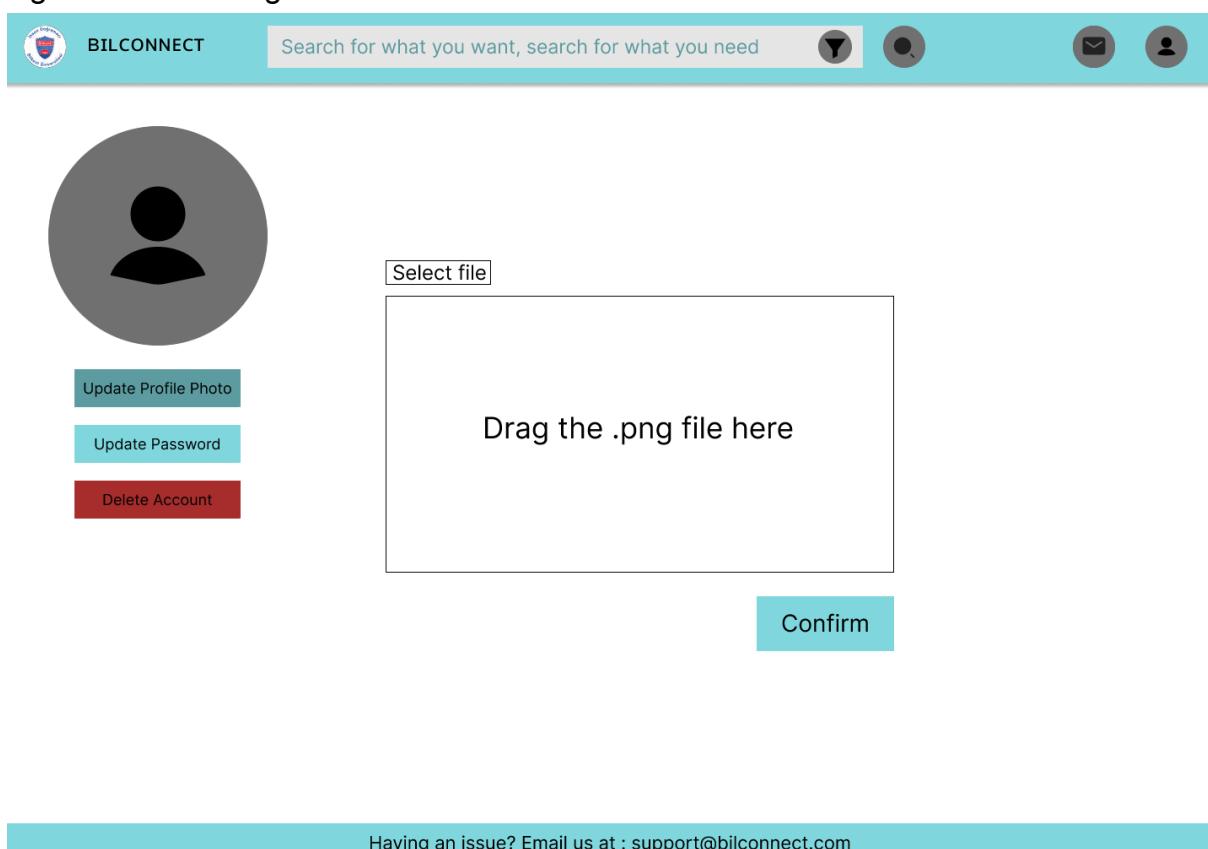


Figure 8.13: Settings Change Profile Picture



BILCONNECT

Search for what you want, search for what you need

Having an issue? Email us at : support@bilconnect.com

Profile Picture Placeholder

Enter old password:

Enter new password:

Enter new password again:

Email confirmation code:

Update Profile Photo

Update Password

Delete Account

Confirm

Password Requirements:

- Minimum 8 characters
- At least one capital letter
- At least one lower case letter
- At least one symbol
- At least one digit
- New password can't be same as the old one.

Fig 8.14: Settings Change Password



BILCONNECT

Search for what you want, search for what you need

Having an issue? Email us at : support@bilconnect.com

Profile Picture Placeholder

Enter password:

Email confirmation code:

Update Profile Photo

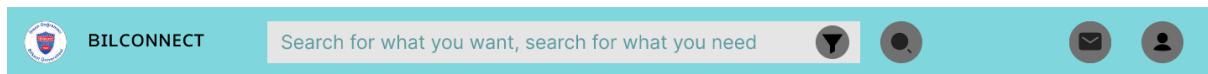
Update Password

Delete Account

Confirm

Your account deletion process will start after these details are provided.
If the account is not logged on for the week after the process starts, the account will be deleted.

Figure 8.15: Settings Delete Account



My Posts

Sort by: Date Ascending

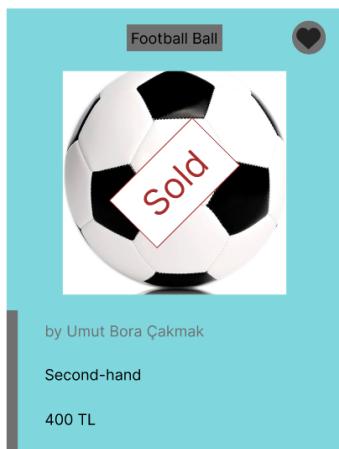


Figure 8.16: View Own Posts



Favourited Posts

Sort by: Date Ascending

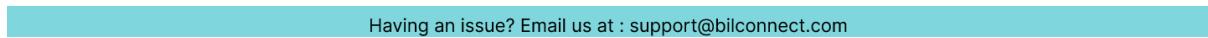


Figure 8.17: View Favorited Posts

The screenshot shows a product listing for a "Football Ball". The title "Football Ball" is at the top left, followed by a pencil icon for editing. Below the title are two category buttons: "Second-hand" and "Sports", with plus and minus icons for adjusting the count. A seller rating of 5.0/5.0 from 12 reviews is displayed. The item is categorized as "Second-hand / Sports" and has a price of "400 TL". The description states: "Football ball in good condition. Ideal for amateur usage." To the right is a large image of a black and white soccer ball. At the bottom left is a button labeled "Save Edits".

Having an issue? Email us at : support@bilconnect.com

Figure 8.18: Edit Post

The screenshot shows an edit post screen. The product name is "Enter Product Name" with a pencil icon. Below it are two category buttons: "Second-hand" and "Home-use", with plus and minus icons. A seller rating of 5.0/5.0 from 12 reviews is shown. The item is categorized as "Second-hand". A placeholder text "Enter Price in TL" is present, along with a price entry field containing "0". The description field contains the instruction: "Please describe the status of the product." To the right is an image of a clear glass with a colorful cartoon character sticker on it. At the bottom left is a button labeled "Post".

Having an issue? Email us at : support@bilconnect.com

Figure 8.19: Make New Post



Report User

Report Title

Report Category

Report Description

I recently had some problematic experiences with the user in question.

Please deal with the user per the site rules.

The problem is in the attachment, thank you for your time.

 problematic_actions.png 

Having an issue? Email us at : support@bilconnect.com

Figure 8.20: Report User



Report Event

Report Title

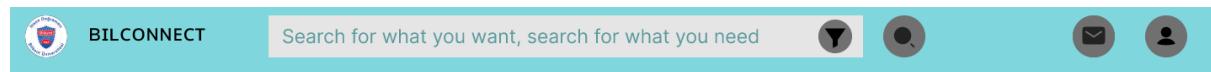
Report Category

Report Description

This event should not happen.

Having an issue? Email us at : support@bilconnect.com

Figure 8.21: Report Event



Report Post

Report Title

Report Category

Report Description

This post is against guidelines.

Having an issue? Email us at : support@bilconnect.com

Figure 8.22: Report Post



January 2023 Event Calendar

January 2023

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01 Group Hug Exam Prep					02	03
04	05	06	07	08	09	10
11 Sleep	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31



Having an issue? Email us at : support@bilconnect.com

Figure 8.23: See Event Calendar - Month

The screenshot shows the BILCONNECT website interface. At the top, there is a header bar with the BILCONNECT logo, a search bar containing the placeholder "Search for what you want, search for what you need", and three icons: a magnifying glass, a mail icon, and a user profile icon. Below the header, the main content area displays "January 1st 2023 Events". Two events are listed in a card format:

- Group Hug**
Club: CS Club
Location: Central Gym
Time: 20.00 - 21.00
- Exam Prep**
Club: CS Club
Location: Library
Time: 22.00 - 22.01

At the bottom of the page, there is a footer bar with the text "Having an issue? Email us at : support@bilconnect.com".

Figure 8.24: See Event Calendar - Day

The screenshot shows the BILCONNECT website interface, similar to Figure 8.23, but focusing on a single event: "Group Hug". The event details are displayed in a card format:

Group Hug

Club : CS Club
Location: Central Gym
Club : CS Club

Description
Group hug with the boys. Additionally, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum interdum, metus at feugiat fermentum, lectus lacus euismod tortor, nec porta nisl dolor vitae ex. Sed sollicitudin quam sed lorem aliquam, sagittis blandit purus ultrices. Aliquam sit amet enim gravida, viverra nunc sed, commodo enim. Vestibulum laoreet quis dolor nec dignissim. Mauris eu diam nec orci bibendum ornare nec non lacus. Donec ac maximus purus, quis vehicula felis. Mauris sit amet urna volutpat, efficitur magna id, iaculis magna. Suspendisse pretium condimentum libero, sit amet egestas ligula condimentum a. Nam sit amet ligula sit amet urna semper mollis et sed diam. In hac habitasse platea dictumst. Nullam ornare nulla quis nisl luctus consectetur. Donec rhoncus varius magna, ac condimentum dui maximus vel. Mauris in euismod lorem, ut convallis augue.

At the bottom of the page, there is a footer bar with the text "Having an issue? Email us at : support@bilconnect.com".

Figure 8.25: See Event Calendar - Event Detail

This Month's Event Callendar

January 2023

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
					01 Group Hug Exam Prep	
02	03	04	05	06	07	08
09	10	11 Sleep	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Events of My Club

Group Hug
Club: CS Club
Location: Central Gym
Date: 01.01.2023
Time: 20.00 - 21.00



Figure 8.26: Club Account Main Page

BILCONNECT

CONSETETUR DORIC TINCUS VARIUS MAGNA, AC CONDIMENTUM DURIA MAXIMUS VEL MIDVIS IN CULISIMO LOREM, ut convallis augue.

Exam Prep
Club: CS Club
Location: Library
Date: 01.01.2023
Time: 22.00 - 22.01

Description
Welcome to the annual CS Club CS-319 midterm preparation!
You can use this event to ask our highly capable club members about the problems you are having with example questions prepared by us, or about the course in general. As the midterm is easy, we allocated a total of 1 (one) minute for this prep session.
Example problems can be found in the attachment.

Example_Problems.zip

Add New Event

Having an issue? Email us at : support@bilconnect.com

Figure 8.27: Club Account Main Page Bottom Part

The screenshot shows the bottom portion of a club account page. At the top, there is a header bar with the BilConnect logo and a user icon. Below the header, the event details are listed in a table:

Club:	CS Club
Location:	Central Gym
Date:	01.01.2023
Time:	20.00 - 21.00

Below the table, there is a section titled "Description" containing a large block of placeholder text. At the bottom of the page are two buttons: "Upload Document" and "Save Event". A footer bar at the very bottom contains the text "Having an issue? Email us at : support@bilconnect.com".

Group Hug

Club: CS Club
Location: Central Gym
Date: 01.01.2023
Time: 20.00 - 21.00

Description

Group hug with the boys. Additionally, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum interdum, metus at feugiat fermentum, lectus lacus euismod tortor, nec porta nisl dolor vitae ex. Sed sollicitudin quam sed lorem aliquam, sagittis blandit purus ultrices. Aliquam sit amet enim gravida, viverra nunc sed, commodo enim. Vestibulum laoreet quis dolor nec dignissim. Mauris eu diam nec orci bibendum ornare nec non lacus. Donec ac maximus purus, quis vehicula felis. Mauris sit amet urna volutpat, efficitur magna id, iaculis magna. Suspendisse pretium condimentum libero, sit amet egestas ligula condimentum a. Nam sit amet ligula sit amet urna semper mollis et sed diam. In hac habitasse platea dictumst. Nullam ornare nulla quis nisl luctus consectetur. Donec rhoncus varius magna, ac condimentum dui maximus vel. Mauris in euismod lorem, ut convallis augue.

Upload Document

Save Event

Having an issue? Email us at : support@bilconnect.com

Figure 8.28: Edit Club Event



Group Hug
Club: CS Club
Location: Library
Date: 01.01.2023
Time: 22.00 - 22.01

Description

Welcome to the annual CS Club CS-319 midterm preparation!

You can use this event to ask our highly capable club members about the problems you are having with example questions prepared by us, or about the course in general. As the midterm is easy, we allocated a total of 1 (one) minute for this prep session.

Example problems can be found in the attachment.

Example_Problems.zip

[Upload Document](#)

[Delete Event](#)

[Save Event](#)

Having an issue? Email us at : support@bilconnect.com

Figure 8.29: New Club Event

The login page has a teal background. On the left, there are fields for 'Username' and 'Password' with a 'Login' button below them. On the right, there are links for 'Forgot Password' and 'Create Account'. A vertical line separates the two halves.

League of Legends

Having an issue? Email us at : support@bilconnect.com

Figure 8.30: Login



BILCONNECT

Student Account

Club Account

Email

Confirmation Code

[Resend](#)

Password

[Finish Registration](#)

Having an issue? Email us at : support@bilconnect.com

Figure 8.31: Create Account



BILCONNECT

Username

Confirmation Code

[Resend](#)

New Password

New Password Again

[Change Password](#)

Having an issue? Email us at : support@bilconnect.com

Figure 8.32: Forgot Password Screen

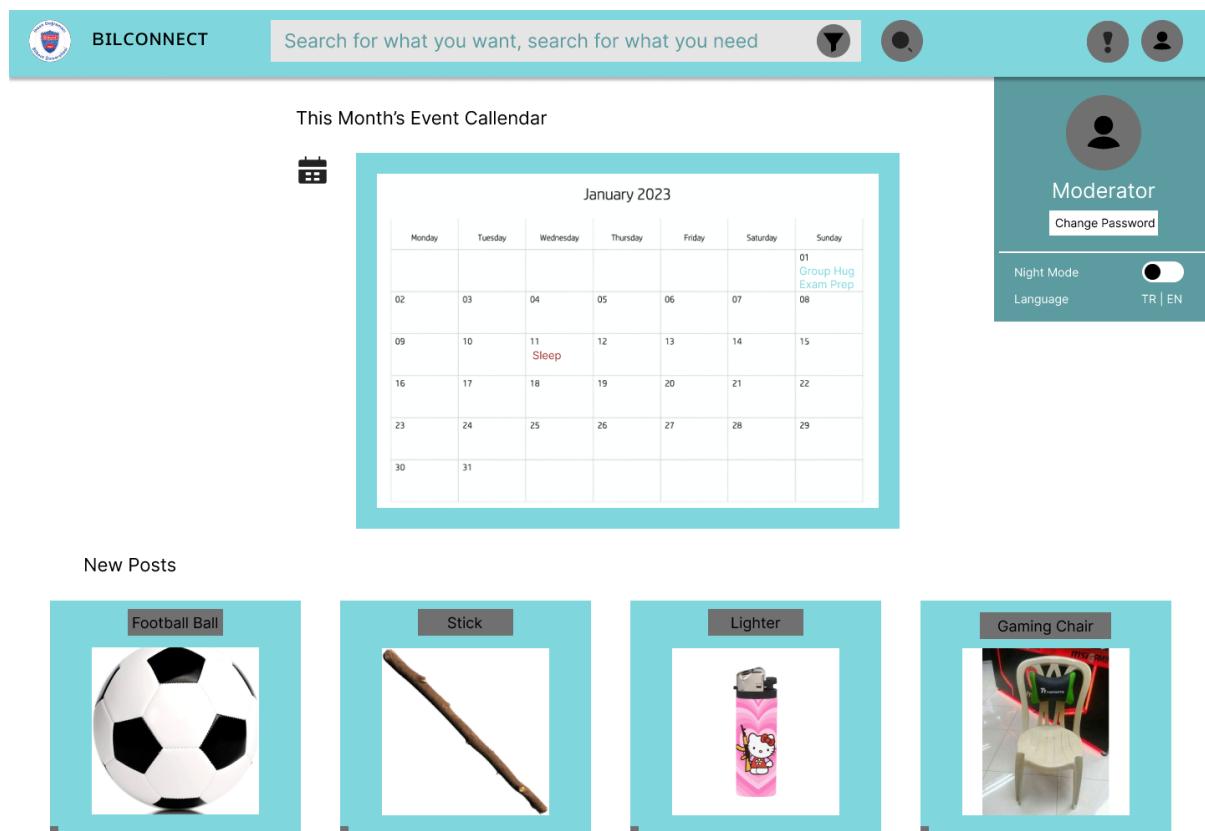


Figure 8.33: Moderator Main Screen

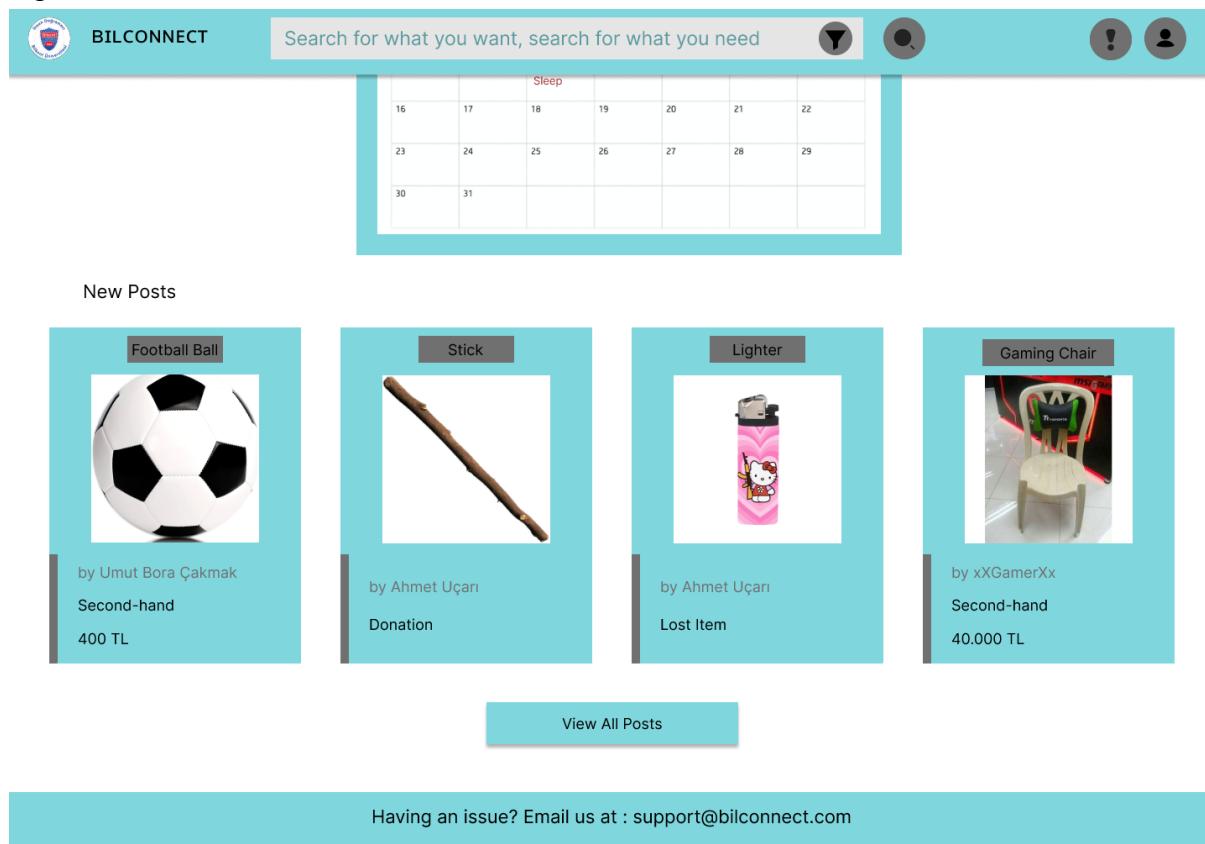


Figure 8.34: Moderator Main Screen Bottom

BILCONNECT

Search for what you want, search for what you need

! 🔎

CS - 319 Midterms Answer Key

Second-hand

CS - 319

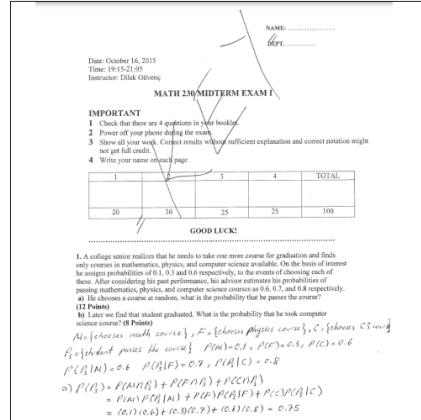
Seller: Cheater ★☆☆☆☆ (1.0/5.0) 3 reviews

Second-hand / CS - 319 Midterms Answer Key

1 TL

Description

I found this document for the upcoming CS-319 midterm.



Remove Post

Having an issue? Email us at : support@bilconnect.com

Figure 8.35: Moderator View Post Detail

BILCONNECT

Search for what you want, search for what you need

! 🔎

CS - 319 Midterms Answer Key

Second-hand

CS - 319

Seller: Cheater ★☆☆☆☆ (1.0/5.0) 3 reviews

Second-hand / CS - 319 Midterms Answer Key

1 TL

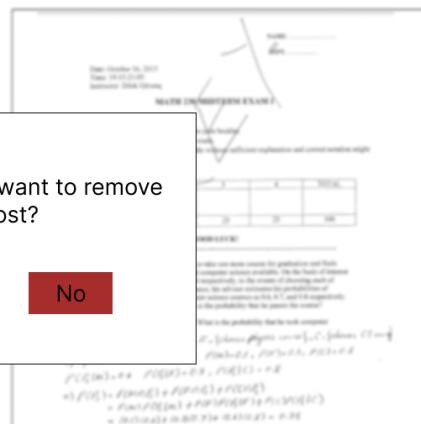
Description

I found this document for the upcoming CS-319 midterm.

Are you sure you want to remove this post?

Yes

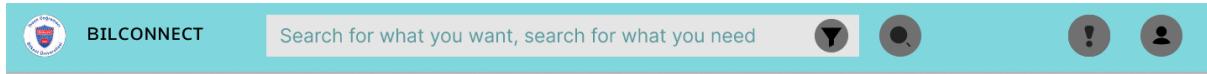
No



Remove Post

Having an issue? Email us at : support@bilconnect.com

Figure 8.36: Moderator View Post Detail - Confirmation



BILCONNECT Search for what you want, search for what you need

Group Hug

Club : CS Club

Location: Central Gym

Club : CS Club

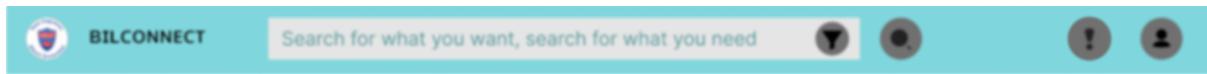
Description

Group hug with the boys. Additionally, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum interdum, metus at feugiat fermentum, lectus lacus euismod tortor, nec porta nisl dolor vitae ex. Sed sollicitudin quam sed lorem aliquam, sagittis blandit purus ultrices. Aliquam sit amet enim gravida, viverra nunc sed, commodo enim. Vestibulum laoreet quis dolor nec dignissim. Mauris eu diam nec orci bibendum ornare nec non lacus. Donec ac maximus purus, quis vehicula felis. Mauris sit amet urna volutpat, efficitur magna id, iaculis magna. Suspendisse pretium condimentum libero, sit amet egestas ligula condimentum a. Nam sit amet ligula sit amet urna semper mollis et sed diam. In hac habitasse platea dictumst. Nullam ornare nulla quis nisl luctus consectetur. Donec rhoncus varius magna, ac condimentum dui maximus vel. Mauris in euismod lorem, ut convallis augue.

[Remove Event](#)

Having an issue? Email us at : support@bilconnect.com

Figure 8.37: Moderator View Event



BILCONNECT Search for what you want, search for what you need

Group Hug

Club : CS Club

Location: Central Gym

Club : CS Club

Description

Group hug with the boys. Additionally, Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum interdum, metus at feugiat fermentum, lectus lacus euismod tortor, nec porta nisl dolor vitae ex. Sed sollicitudin quam sed lorem aliquam, sagittis blandit purus ultrices. Aliquam sit amet enim gravida, viverra nunc sed, commodo enim. Vestibulum laoreet quis dolor nec dignissim. Mauris eu diam nec orci bibendum ornare nec non lacus. Donec ac maximus purus, quis vehicula felis. Mauris sit amet urna volutpat, efficitur magna id, iaculis magna. Suspendisse pretium condimentum libero, sit amet egestas ligula condimentum a. Nam sit amet ligula sit amet urna semper mollis et sed diam. In hac habitasse platea dictumst. Nullam ornare nulla quis nisl luctus consectetur. Donec rhoncus varius magna, ac condimentum dui maximus vel. Mauris in euismod lorem, ut convallis augue.

Are you sure you want to remove this event?

Yes

No

[Remove Event](#)

mod tortor, nec porta nisl dolor
ultrices. Aliquam sit amet
quis dolor nec dignissim.
maximus purus, quis vehicula
a. Suspendisse pretium
t amet ligula sit amet urna
ornare nulla quis nisl luctus
ut convallis augue.

Having an issue? Email us at : support@bilconnect.com

Figure 8.38: Moderator View Event - Confirmation

BILCONNECT

Search for what you want, search for what you need

! 🔎

Showing 1 result(s) for tag(s) Removed

Sort by: Price Descending



Having an issue? Email us at : support@bilconnect.com

Figure 8.39: Moderator View Removed Posts

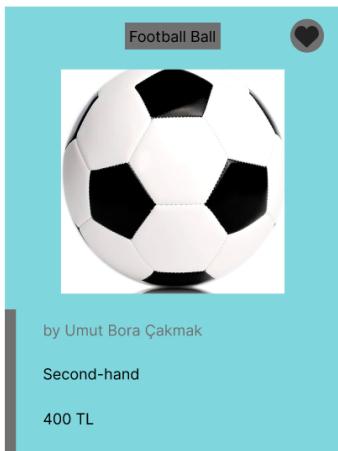
BILCONNECT

Search for what you want, search for what you need

! 🔎

Showing 1 result(s) for user Umut Bora Çakmak

Sort by: Price Descending



Having an issue? Email us at : support@bilconnect.com

Figure 8.40: Moderator Filter by User



BILCONNECT

Search for what you want, search for what you need






Reports

User Named xXGamerXx	Johny John
Inappropriate Username	
Post about cheating	Prof. Smart
Product Against Guidelines	
Midterm papers on sale	Prof. Smarter
User Against Guidelines	
Foul Language	Mary Mary
User Used Inappropriate Language	Resolved
Problematic Event	Club Hamilton
Event Against School Guidelines	

Having an issue? Email us at : support@bilconnect.com

Figure 8.41: Moderator View Reports



BILCONNECT

Search for what you want, search for what you need






Post Report

Report Title Post about cheating

Report Category Product Against Guidelines

Report Description

Post lists my midterm as the product, please take it down swiftly.

[See Post Details](#)

[Remove Post](#)

Having an issue? Email us at : support@bilconnect.com

Figure 8.42: Report - Post

63



Post Report

Report Title Post about cheating

Report Category Product Against Guidelines

Report Description

Post lists my midterm as the p

Are you sure you want to remove
this post?

Yes

No

See Post Details

Remove Post

Having an issue? Email us at : support@bilconnect.com

Figure 8.43: Report - Post - Removal Confirmation



Post Report

Report Title Midterm papers on sale

Report Category User Against Guidelines

Report Description

User should be banned, serious breach of guidelines

See Chats With
Reportee

See User Posts

Ban User

Having an issue? Email us at : support@bilconnect.com

Figure 8.44: Report - User

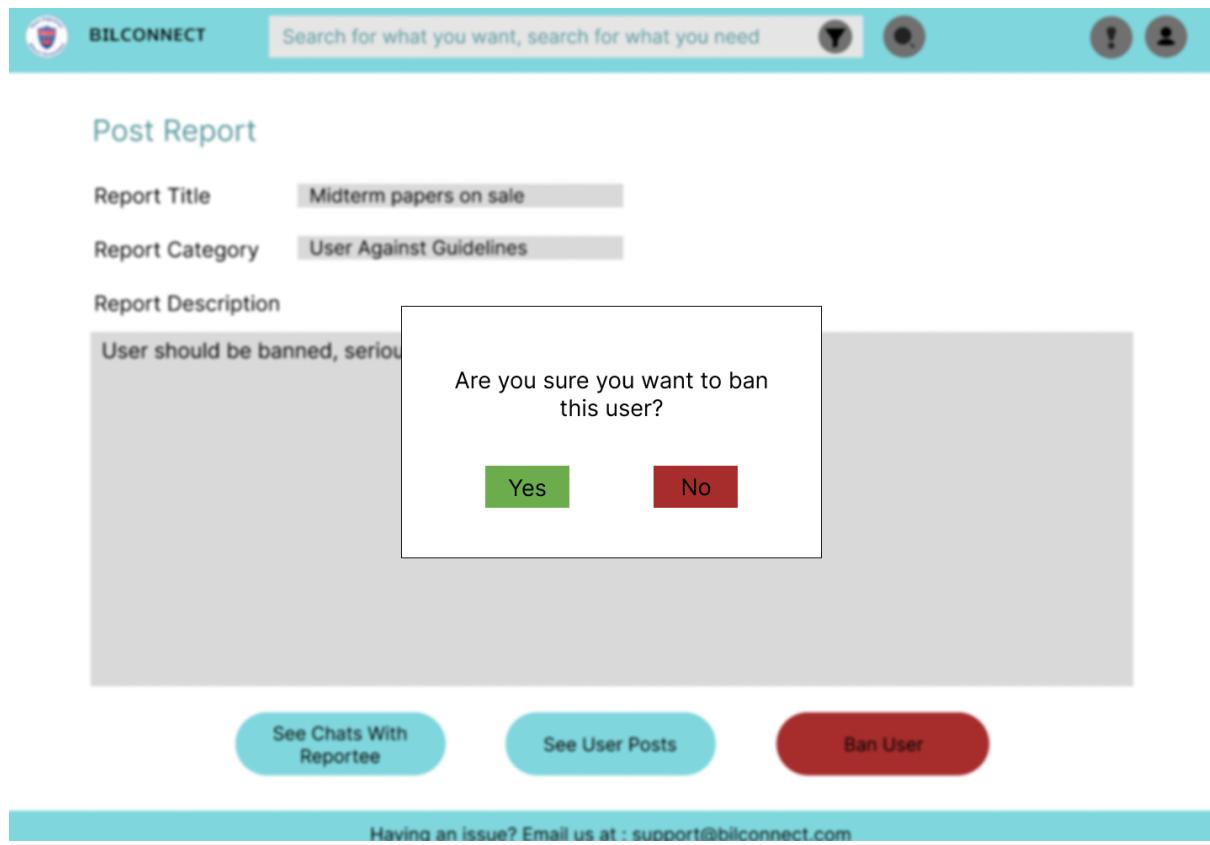


Figure 8.45: Report - User - Confirmation

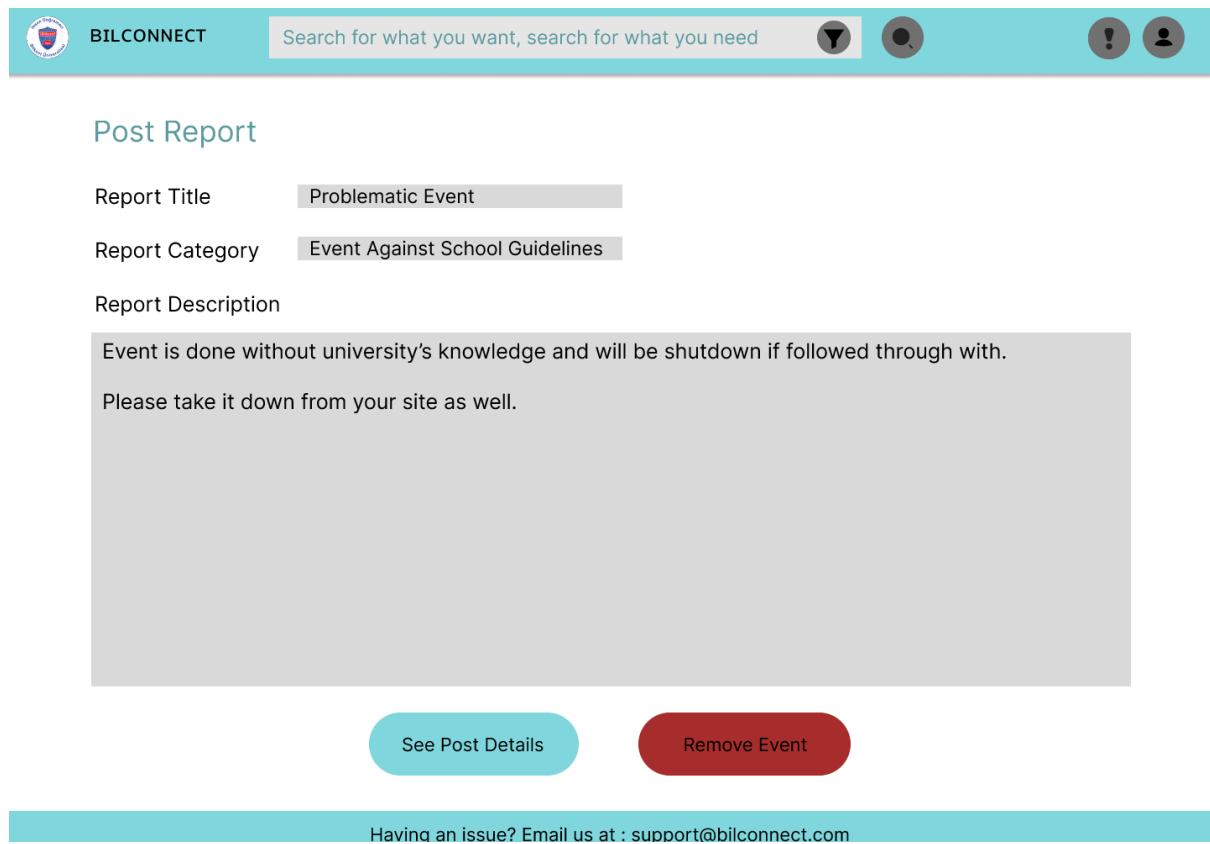
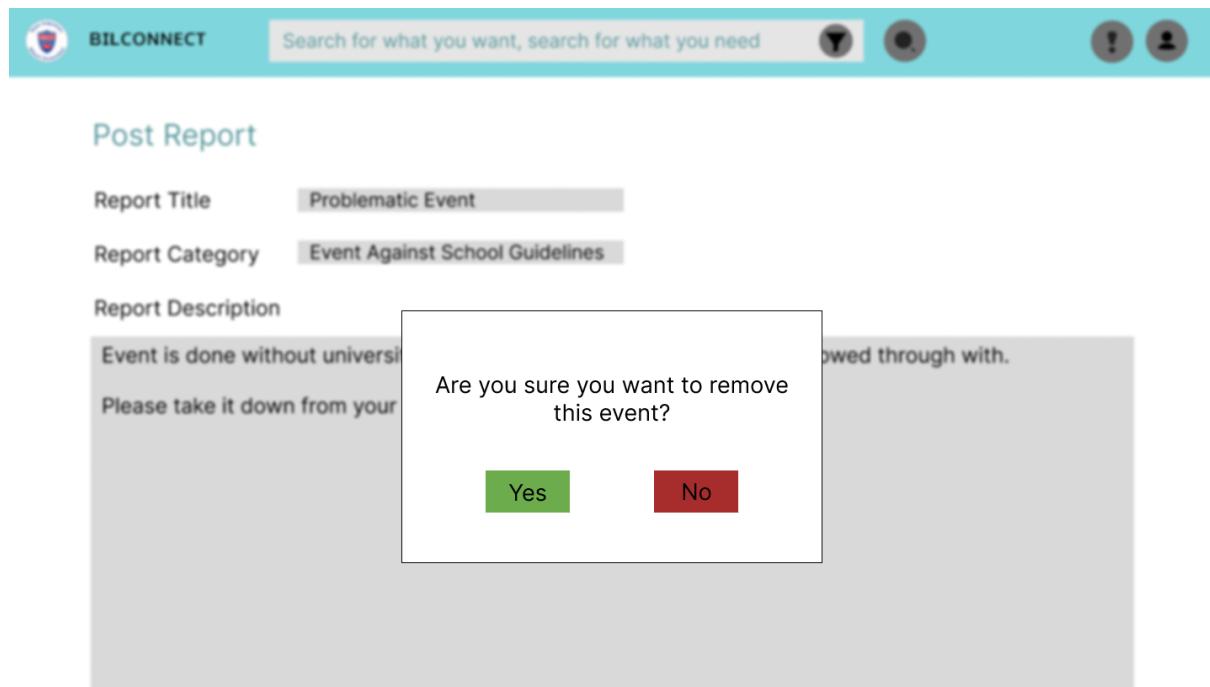


Figure 8.46: Report - Event



Having an issue? Email us at : support@bilconnect.com

Figure 8.47: Report - Event - Confirmation

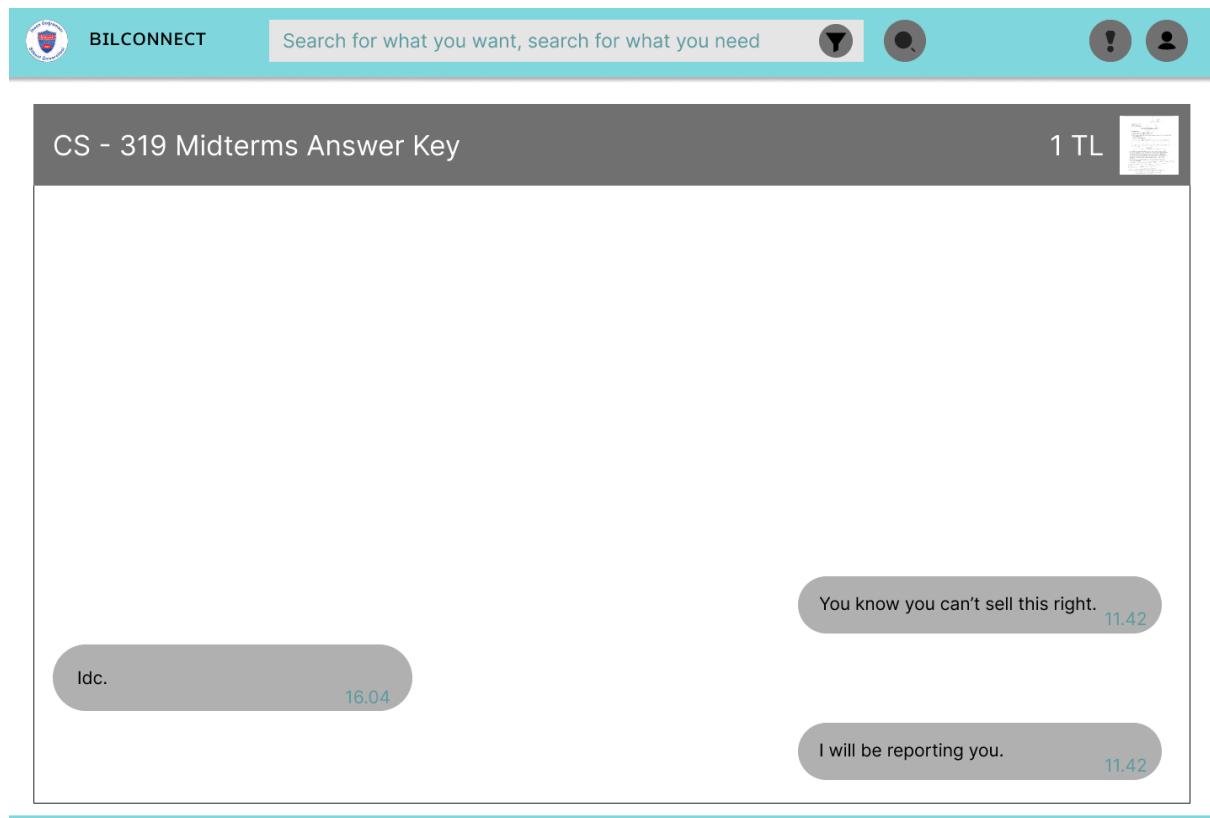


Figure 8.48: Report - View Message History

We used Figma to design our web app. For further details, refer to the Figma link below:

<https://www.figma.com/file/raDOosUb8R59PlqjYprqea/BilConnect-team-library?type=design&node-id=0%3A1&mode=design&t=JY2mSssEp6H1JZKZ-1>