D1: Use Cases & Non-Functional Requirements & Tech. Stack
D2- Class/Sequence/Activity/State Diagrams
Mockup UI Design

S3T11

BilFind

Serhat Merak **(22002414)**
Mirza Özgür Atalar **(22003455)**
Kaan Türkoğlu **(22102105)**
Ege Karaahmetoğlu **(22102622)**
Orhun Aysan **(22103005)**

Contents

Use Case Diagram 4

Textual Use Case 5

Authentication 5

Login 5

Register 6

Forgot Password 7

Post Management 8

Create Post 8

Delete Post 9

Edit Post 10

Search Post 11

View Post 12

Add to Favorites 13

Start Chat 14

Profile 15

Logout 15

View Others Profile 16

View Own Profile 17

Edit Profile 18

Delete Profile 19

Change Password 20

Control Panel 21

Ban User 21

Handle Support Request 22

Destroy Post 23

Manage Admin 24

Make Announcement 25

Others 26

Create Support Request 26

Report 27

Send Message 28

Send E-mail 29

View Chat Box 30
Class Diagram 31
Sequence Diagrams 32
Login Sequence Diagram 32
Create Post Sequence Diagram 33
Ban User Sequence Diagram 34
Activity Diagrams 35
Create Post Activity Diagram 35
Communication Activity Diagram 36
Report Post Activity Diagram 37
State Diagrams 38
Post State Diagram 38
Report State Diagram 39
Mockup UI 40
Login Page 40
Sign Up 41
Forgot Password 42
Main Page 43
Detailed Post 44
User’s Favorite and Own Posts 45
Create Post 46
Chat Menu 47
Report Submission Pop-Up 48
Post Menu Pop-Up 49
Error Page 50
Non-Functional Requirements 51
1) Performance: 51
2) Security: 51
3) Usability: 51
4) Reliability: 52
5) Maintainability: 52
Tech Stack 53

Use Case Diagram

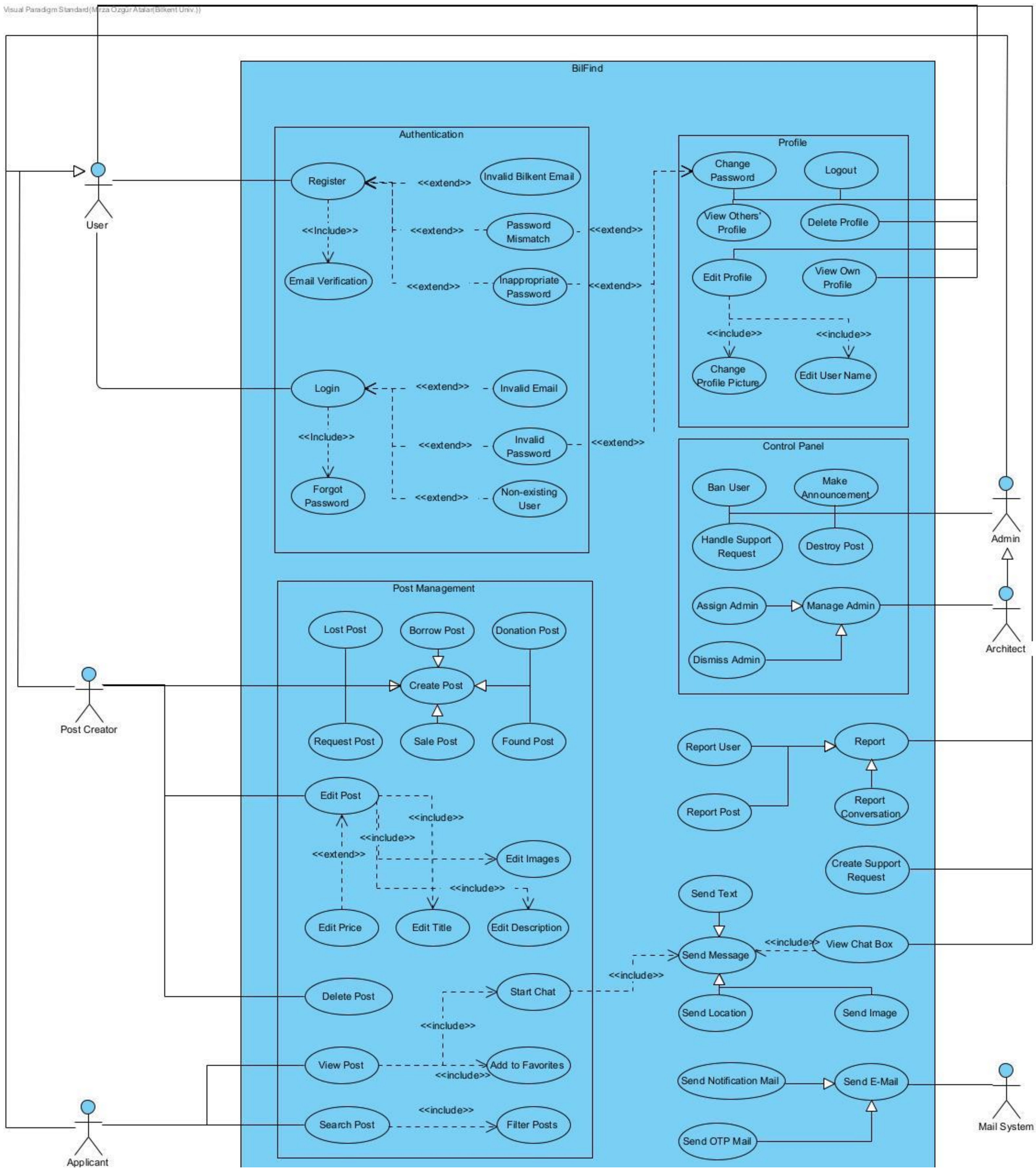


Figure 1 Use Case Diagram

Textual Use Case

Authentication

Login

Use Case Name	Login	
Actors	User, Admins	
Description	This use case is provided when the users want to sign in the system. The user has to login into the system to be able to perform any kind of action.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user tries to enter the system. The user enters the required Bilkent email and password.	**System Response** **Step 2:** The system checks whether the email is in the database. **Step 3:** Check whether the user matches an email-password combination. **Step 4:** The user is redirected to the home page.
Alternate Courses	**Step 1:** If the user enters their password wrong or if there is no specified user with the given credentials, the login page is reloaded so that the validation process starts over. **Step 2:** An error message is shown to the user as a pop-up if the email is not registered to the system or the password-email combination is wrong.	
Pre-condition	-	
Post-condition	If the login is successful, the user will be redirected to the home page.	
Assumptions	None	

Register

Use Case Name	Register	
Actors	User	
Description	This use case is shown when the users want to sign up for the system. Also, by sending an email to confirm whether the user is a Bilkent University student, this use case increases security.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: The user who wants to join the BilFind is shown the register page. **Step 2:** The user enters the necessary information so that he/she can be validated.	
		Step 3: The system checks whether the new user is a member of Bilkent University by sending an email to him/her.
	Step 4: The user submits the code taken from the email.	
		Step 5: If the code is valid, user is registered into the system.
		Step 6: The user is redirected to the home page
Alternate Courses	**Step 1:** If the user does not enter matching passwords upon confirmation or the password does not satisfy the requirements, the system will reload the register page stating the problem. **Step 2:** The user sees an error if the provided email is not a Bilkent Email.	
Pre-condition	The user must be a member of Bilkent University.	
Post-condition	User will be redirected to the home page	
Assumptions	The user is a member of the Bilkent community.	

Forgot Password

Use Case Name	Forgot Password	
Actors	User	
Description	This use case occurs when a member of BilFind forgets their password.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user clicks the “Forgot Password” option on the login screen. **Step 2:** The user enters his/her Bilkent email. **Step 5:** The user clicks the link from his/her email. **Step 6:** Enter the new password for his account and send it to the backend. **Step 8:** The user can log in with the new password.	**System Response** **Step 3:** Member details are fetched from the database to check whether the user exists. **Step 4:** An email containing a reset password link is sent to the user. **Step 7:** New password and reset password token is validated and saved to the database.
Alternate Courses	**Step 1:** The email verification code is re-sent if a problem occurs while validating the user. **Step 2:** If no member has the provided user ID, a pop-up occurs to inform the user. After that, the user is redirected to the login page.	
Pre-condition	The user is already on the system, so an email can be sent to validate and allow the user to get a new password	
Post-condition	None	
Assumptions:	The user is already registered to BilFind.	

Post Management

Create Post

Use Case Name	Create Post	
Actors	Post Creators	
Description	This use case describes the process of putting new posts on the market. This includes posting all kinds of posts which are lost, borrow, donation, request, sale, and found posts. Moreover, we have one use case of creating a post even though there are multiple variations of a post to increase usability.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks on the create new post (+) button.	
	Step 2: The user selects the information and photos of the post.	
	Step 3: The user hits the “post” button.	
		Step 4: Post details are pushed to the database.
		Step 5: The system sends the post creator back to the post detail page.
	Step 6: This use case concludes when the user is redirected to the post detail page that he created.	
Alternate Courses	**Step 1:** If a problem occurs during validating the user, the user is redirected to the login screen to fix this issue.	
Pre-condition	The user has to be signed in, and the given post ID must be valid.	
Post-condition	Details of the new post are shown on the webpage.	
Assumptions	None	

Delete Post

Use Case Name	Delete Post	
Actors	Post Creator	
Description	This use case describes the process of removing the existing post from the market by its owner.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks the delete post button.	
		Step 2: Data about posts are deleted from the database.
		Step 3: The system directs the post creator to the post list page.
	Step 4: This use case concludes when the user is redirected to the post list page after deleting successfully	
Alternate Courses	**Step 1:** If a problem occurs during validating the user, the user is redirected to the login screen to fix this issue. **Step 2:** If the request sender user does not own the given post id, the user sees an error.	
Pre-condition	The user has to be signed in, and that user must own the given post id.	
Post-condition	Details of the new post are shown on the webpage.	
Assumptions	None	

Edit Post

Use Case Name	Edit Post	
Actors	Post Creator	
Description	This use case describes the process of editing a post which includes editing the images, title, description, and price.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks on “edit post“ button which is on the post page.	
	Step 2: After the editing process is over, the new information is posted to the database.	
		Step 3; The database is updated with the given data
		Step 4: Success response is returned to the user
	Step 5: This use case concludes when the user is done with editing the post according to the needs.	
Alternate Courses	**Step 1:** If a problem occurs during validating the post creator, the post creator is redirected to the login screen to fix this issue.	
Pre-condition	The post creator has to be signed in and the post must exist..	
Post-condition	None	
Assumptions	The post creator is a member of BilFind	

Search Post

Use Case Name	Search Post	
Actors	Applicants	
Description	This use case describes the process of applicants searching for a desired post using the search bar or the filters given.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when an applicant uses the search bar or filters in the search screen	
		Step 2: Posts are listed according to the given filters and provided text taken from the applicant
		Step 3: The list of posts is returned to the client from the backend.
	Step 4: This use case concludes when the applicant sees the list of posts.	
Alternate Courses	**Step 1:** If a problem occurs during validating the applicant, the applicant is redirected to the login screen to fix this issue. **Step 2:** If there is no post based on the given filters, the applicant sees a message indicating it.	
Pre-condition	The applicant has to be signed in.	
Post-condition	The list of post is shown on the webpage.	
Assumptions	None	

View Post

<table>
<tr><td><b>Use Case Name</b></td><td colspan="2">View Post</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Applicants</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This use case describes the process of applicants viewing the post details listed on the page.</td></tr>
<tr><td><b>Typical Course of Events</b></td><td><b><u>Actor Action</u></b>
<b>Step 1:</b> This use case is initiated when an applicant clicks on the post card listed on the page.

<b>Step 4:</b> This use case concludes when the applicant is redirected to the post detail page with the desired post.</td><td><b><u>System Response</u></b>

<b>Step 2:</b> Post details are fetched from the database.

<b>Step 3:</b> The resulting post object is returned to the applicant.</td></tr>
<tr><td><b>Alternate Courses</b></td><td colspan="2"><b>Step 1:</b> If a problem occurs during validating the applicant, the applicant is redirected to the login screen to fix this issue.

<b>Step 2:</b> If there is no post with the provided user id on the post detail page, the post not found label is indicated for the applicant.</td></tr>
<tr><td><b>Pre-condition</b></td><td colspan="2">The applicant has to be signed in, and the given post id must be valid.</td></tr>
<tr><td><b>Post-condition</b></td><td colspan="2">Post details are shown on the webpage.</td></tr>
<tr><td><b>Assumptions</b></td><td colspan="2">None</td></tr>
</table>

Add to Favorites

Use Case Name	Add to Favorites	
Actors	Applicants	
Description	This use case describes the process of users saving a post as a favorite.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when an applicant clicks the heart button on a post. **Step 3:** This use case concludes when the applicant sees the red heart icon on the post card.	**System Response** **Step 2:** The post is saved for the applicant's favorite list in the database.
Alternate Courses	**Step 1:** If a problem occurs during validating the applicant, the applicant is redirected to the login screen to fix this issue.	
Pre-condition	The applicant has to be signed in, and that applicant must not own the given post id.	
Post-condition	The red heart is seen on the card	
Assumptions	None	

Start Chat

Use Case Name	Start Chat	
Actors	Applicant	
Description	Applicants can send private messages to each other.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when an applicant wants to communicate with the post owner or reply to the taken messages. **Step 4:** This use case concludes when the applicant sees the sent message.	**System Response** **Step 2:** The message is saved to the database. **Step 3:** Push notification is sent to the target applicant.
Alternate Courses	**Step 1:** If a problem occurs during validating the applicant, the applicant is redirected to the login screen to fix this issue.	
Pre-condition	The applicant has to be signed in, and the target user must be valid.	
Post-condition	Applicant sees the sent message to user	
Assumptions	None	

Profile

Logout

Use Case Name	Logout	
Actors	User, Admins	
Description	This use case is provided when the user wants to exit from the system.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user wants to exit from the system.	**System Response** **Step 2:** The user is redirected to the login page.
Alternate Courses	None	
Pre-condition	The user had already signed in to the system.	
Post-condition	If the login is successful, the user will be redirected to the login page.	
Assumptions	None	

View Others Profile

Use Case Name	View Others Profile	
Actors	User	
Description	This use case describes the process of viewing other user’s profiles	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user clicks on the profile photo on the post **Step 3:** This use case concludes when the user sees the profile details.	**System Response** **Step 2:** The page is redirected to the profile details screen.
Alternate Courses	**Step 1:** If a problem occurs while validating the user, the user is redirected to the login screen to fix this issue. **Step 2:** If a problem occurs while validating the profile to be shown, the user is redirected to the post screen.	
Pre-condition	The user has to be signed in.	
Post-condition	None	
Assumptions	The user is a member of BilFind	

View Own Profile

Use Case Name	View Own Profile	
Actors	User	
Description	This use case describes the process of the user viewing his/her own profile.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks on the profile button from the menu.	
		Step 2: The page is redirected to the profile screen.
	Step 3: This use case concludes when the user sees the profile details.	
Alternate Courses	**Step 1:** If a problem occurs while validating the user, the user is redirected to the login screen to fix this issue. **Step 2:** If a problem occurs while validating the profile to be shown, the user is redirected to the home screen.	
Pre-condition	The user has to be signed in.	
Post-condition	None	
Assumptions	The user is a member of BilFind	

Edit Profile

Use Case Name	Edit Profile	
Actors	Users	
Description	This use case describes the process of editing the user profile which includes changing profile picture and user name	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks on “edit profile“ button which is on the profile page.	
	Step 2: After the editing process is over, the new information is posted to the database.	
		Step 3; The database is updated with the given data
		Step 4: Success response is returned to the user
	Step 5: This use case concludes when the user is done with editing the profile according to the needs.	
Alternate Courses	**Step 1:** If a problem occurs during validating the user, the user is redirected to the login screen to fix this issue.	
Pre-condition	The user has to be signed in.	
Post-condition	None	
Assumptions	The user is a member of BilFind	

Delete Profile

Use Case Name	Delete Profile	
Actors	User	
Description	This use case describes the process of removing the existing profile from the system by its owner.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks the delete profile button.	
		Step 2: Data about the profile is deleted from the database.
		Step 3: The system directs the user to the login page.
	Step 4: This use case concludes when the user is redirected to the login page after deleting successfully	
Alternate Courses	**Step 1:** If a problem occurs during validating the user, the user is redirected to the login screen to fix this issue.	
Pre-condition	The user has to be signed in	
Post-condition	Profile is deleted from the database.	
Assumptions	None	

Change Password

Use Case Name	Change Password	
Actors	User, Admin	
Description	This use case describes the process of changing the existing password of the user by its owner.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user clicks the change password button or enters invalid passwords and needs to change the password. **Step 4:** The user enters the new password that he/she wants to set.	**System Response** **Step 2:** Data about the password is sent to the database and the previous password is deleted. **Step 3:** The system sends a notification to the user about whether the new password is accepted or not and if it is accepted it directs the user to the login page. On the other hand, if it is not accepted or an error occurs, the user is shown a pop-up error message
Alternate Courses	**Step 1:** If the new password is not accepted by the system or an error occurs, the user is shown a pop-up error message	
Pre-condition	A password for the user must already exist	
Post-condition	Password is changed in the database.	
Assumptions	None	

Control Panel

Ban User

Use Case Name	Ban User	
Actors	Admin	
Description	This use case occurs when the admin removes a user from the system if needed.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case happens when the admin decides to remove a user from the system.	
		Step 2: The user is deleted from the database, and a successful response is returned to the user.
	Step 3: Admin sees a success message.	
Alternate Courses	**Step 1:** If a problem occurs during validating the admin, the admin is redirected to the login screen to fix this issue. **Step 2:** If a problem occurs during validating the user, the admin is shown a pop-up message that states the condition of the user.	
Pre-condition	The admin must be logged in and the user to be banned must exist	
Post-condition	Admin will be shown a pop-up message.	
Assumptions	None	

Handle Support Request

Use Case Name	Handle Support Request	
Actors	Admin	
Description	This use case occurs when a support request reaches an admin.	
Typical Course of Events	**Actor Action** **Step 1:** A user sends a support request **Step 3:** The admin answers the needs of the users	**System Response** **Step 2:** This request is shown to the admin. **Step 3:** The answer of the admin is shown to the related user.
Alternate Courses	**Step 1:** If a problem occurs during validating the admin, the admin is redirected to the login screen to fix this issue. **Step 1:** If a problem occurs while sending a support request or the answer to this request, both the admin and the user receives a notification.	
Pre-condition	The admin must be logged in to the system	
Post-condition	None	
Assumptions:	None	

Destroy Post

Use Case Name	Destroy Post	
Actors	Admin	
Description	This use case occurs when an admin wants to delete a post.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when an admin wants to delete an inappropriate post	
	Step 2: Admin hits the “ban” post button.	
		Step 3: The post is deleted from the database, and a successful response is returned to the user.
	Step 4: Admin sees a success message	
Alternate Courses	**Step 1:** If a problem occurs during validating the admin, the admin is redirected to the login screen to fix this issue.	
Pre-condition	The user should be an admin	
Post-condition	None	
Assumptions:	The user is already registered to BilFind.	

Manage Admin

Use Case Name	Manage Admin	
Actors	Architect	
Description	This use case occurs when an architect wants to assign or dismiss an admin.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when an architect wants to assign or dismiss an admin. **Step 2:** Architect hits either assign admin or dismiss admin buttons.	**System Response** **Step 3:** The admin is assigned or dismissed based on the choices architect made
Alternate Courses	None	
Pre-condition	None	
Post-condition	None	
Assumptions:	None	

Make Announcement

Use Case Name	Make Announcement	
Actors	Admin	
Description	This use case occurs when an admin wants to make an announcement.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when an admin wants to make an announcement about the application. **Step 2:** Admin hits the announce button and enters the content of the announcement.	**System Response** **Step 3:** The system sends notifications to all the registered users about the announcement.
Alternate Courses	None	
Pre-condition	The user should be an admin.	
Post-condition	None	
Assumptions:	None	

Others

Create Support Request

Use Case Name	Create Support Request	
Actors	User	
Description	Users can send emails to admins when they encounter a problem.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: This use case is initiated when a user clicks the ‘?’ icon.	
		Step 2: The message retrieved from the user is saved to the database.
		Step 3: The success response is returned to the user.
	Step 4: This use case concludes when the user sees a success message.	
Alternate Courses	**Step 1:** If a problem occurs during validating the user, the user is redirected to the login screen to fix this issue.	
Pre-condition	The user has to be signed in.	
Post-condition	User sees a popup message.	
Assumptions	None	

Report

<table>
<tr><td><b>Use Case Name</b></td><td colspan="2">Report</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Applicants</td></tr>
<tr><td><b>Description</b></td><td colspan="2">This use case describes the process of reporting an inappropriate post, user, or conversation. This use case will help applicants to have a friendlier environment, therefore, increase usability.</td></tr>
<tr><td><b>Typical Course of Events</b></td><td><b>Actor Action</b>
<b>Step 1:</b> This use case is initiated when an applicant finds a post, user or conversation inappropriate and reports it.
<b>Step 4:</b> This use case concludes when the applicant completes the report process.</td><td><b>System Response</b>
<b>Step 2:</b> Post, user or conversation details are fetched from the database.
<b>Step 3:</b> Report details are posted to the database, and a notification is sent to the admins.</td></tr>
<tr><td><b>Alternate Courses</b></td><td colspan="2"><b>Step 1:</b> If a problem occurs during validating the applicant, the applicant is redirected to the login screen to fix this issue.
<b>Step 2:</b> If there is no post, user or conversation related to the report, the user sees a message indicating it.</td></tr>
<tr><td><b>Pre-condition</b></td><td colspan="2">The applicant has to be signed in.</td></tr>
<tr><td><b>Post-condition</b></td><td colspan="2">The applicant is redirected to the post’s page.</td></tr>
<tr><td><b>Assumptions</b></td><td colspan="2">None</td></tr>
</table>

Send Message

Use Case Name	Send Message	
Actors	User	
Description	Users can send messages to other users so that they can communicate with each other. There are three types of messages that can be sent which are send text, send location and send image.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when a user decides to send a message to other users. **Step 2:** The user selects the type of the message to be sent and enters the proper input. Then the user clicks on the send button.	**System Response** **Step 3:** System informs the user about the status of the given message such as sent/received
Alternate Courses	**Step 1:** If a problem occurs during sending the message, the user is shown a pop-up stating the error and status of the message	
Pre-condition	The user has to be signed in.	
Post-condition	User sees a popup message.	
Assumptions	None	

Send E-mail

Use Case Name	Send E-mail	
Actors	Mail System	
Description	The mail system is responsible for sending two types of emails which are OTP mails and notification mails. OTP mails are usually sent when a user tries to register to the system and notification mails are used for various different purposes such as confirmation.	
Typical Course of Events	**Actor Action** **Step 1:** This use case is initiated when the Mail System is asked to send notification or OTP type mails.	**System Response** **Step 3:** The third-party system returns whether the email is successfully sent or not.
Alternate Courses	**Step 1:** If a problem occurs during sending the email, Mail System receives an error and tries to resend the email.	
Pre-condition	None	
Post-condition	None	
Assumptions	None	

View Chat Box

Use Case Name	View chat box	
Actors	User	
Description	Users can see all the messages that are previously sent through a chat box.	
Typical Course of Events	**Actor Action**	**System Response**
	Step 1: The user clicks to chat box from the menu	
		Step 2: The system fetches the previous conversations and show them to the user
	Step 3: The user is shown the conversation that are done with the other users in a list format	
Alternate Courses	None	
Pre-condition	The user has to be signed in.	
Post-condition	User sees all the old conversations	
Assumptions	None	

Class Diagram

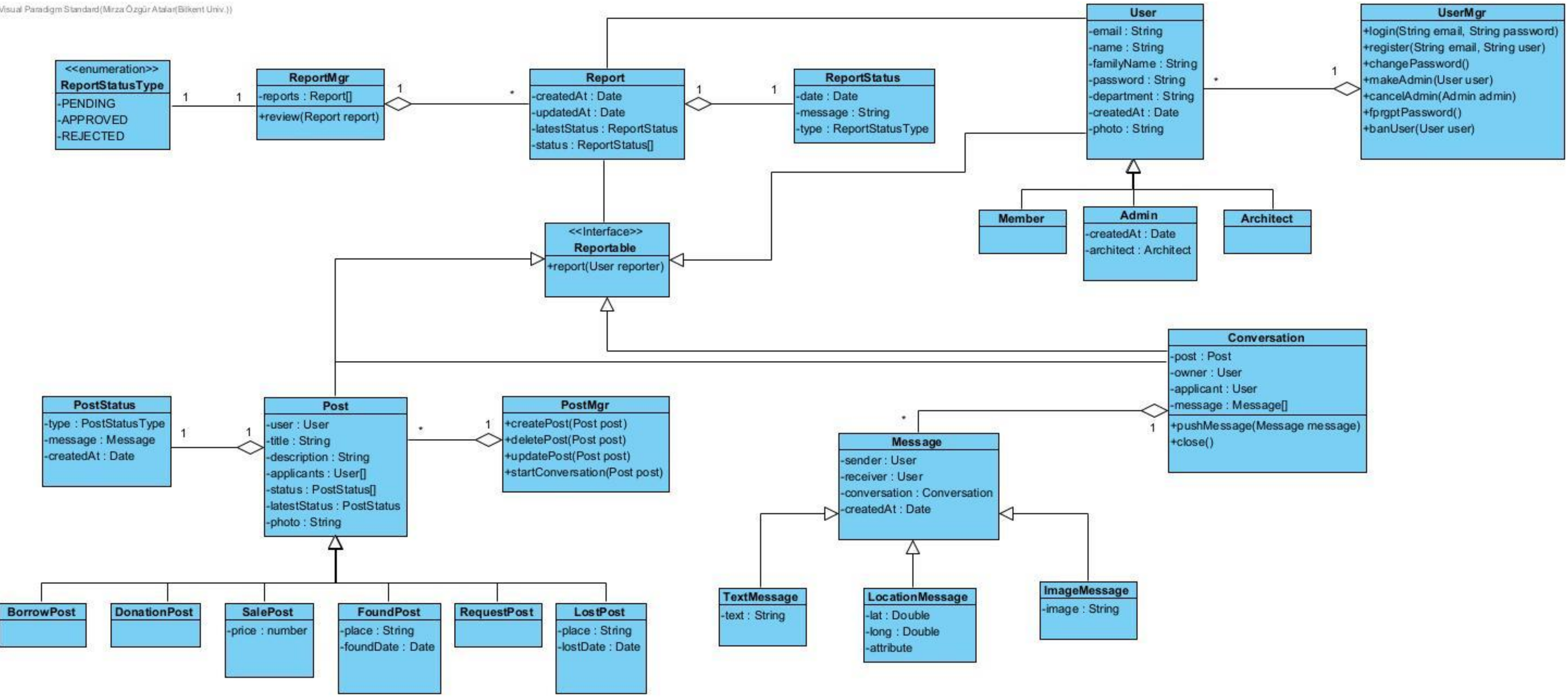


Figure 2 Class Diagram

Sequence Diagrams

Login Sequence Diagram

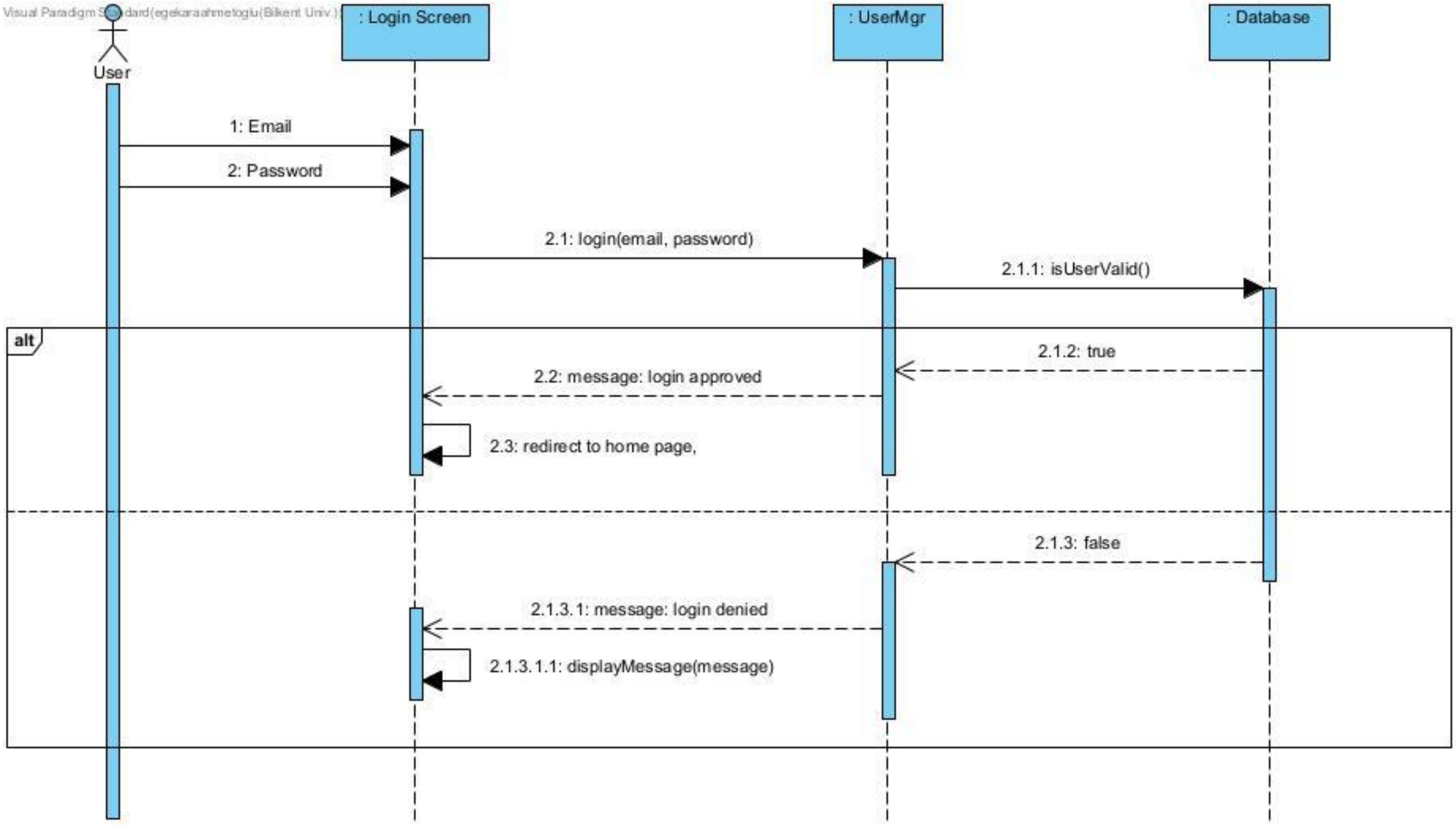

Visual Paradigm Standard(egekaraahmetoglu(Bilkent Univ.))
User
: Login Screen
: UserMgr
: Database
1: Email
2: Password
2.1: login(email, password)
2.1.1: isUserValid()
alt
2.1.2: true
2.2: message: login approved
2.3: redirect to home page,
2.1.3: false
2.1.3.1: message: login denied
2.1.3.1.1: displayMessage(message)

Figure 3 Login Sequence Diagram

Create Post Sequence Diagram

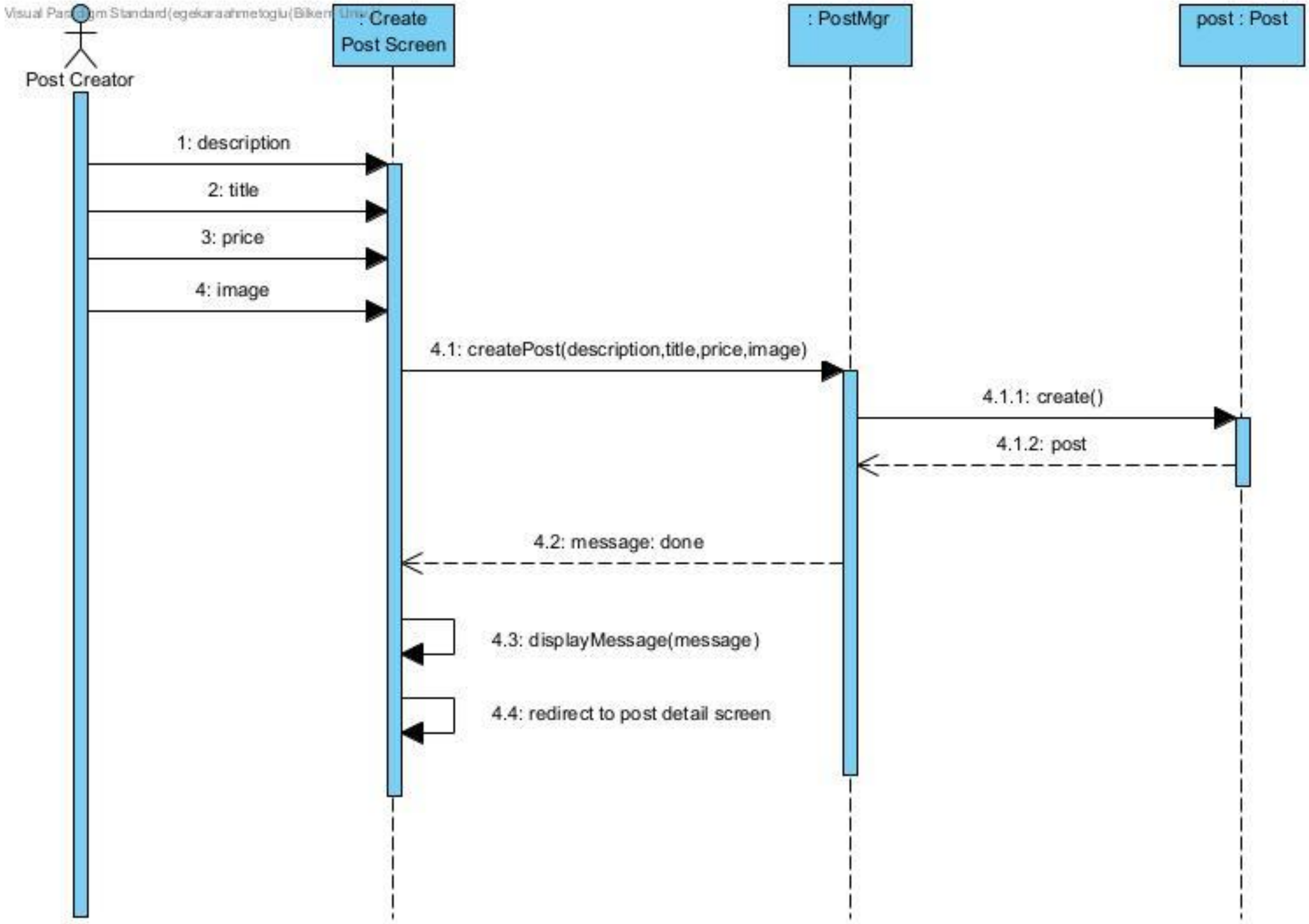

Visual Paradigm Standard(egekaraahmetoglu(Bilkent Univ.))
Post Creator
: Create Post Screen
: PostMgr
post : Post
1: description
2: title
3: price
4: image
4.1: createPost(description,title,price,image)
4.1.1: create()
4.1.2: post
4.2: message: done
4.3: displayMessage(message)
4.4: redirect to post detail screen

Figure 4 Create Post Sequence Diagram

Ban User Sequence Diagram

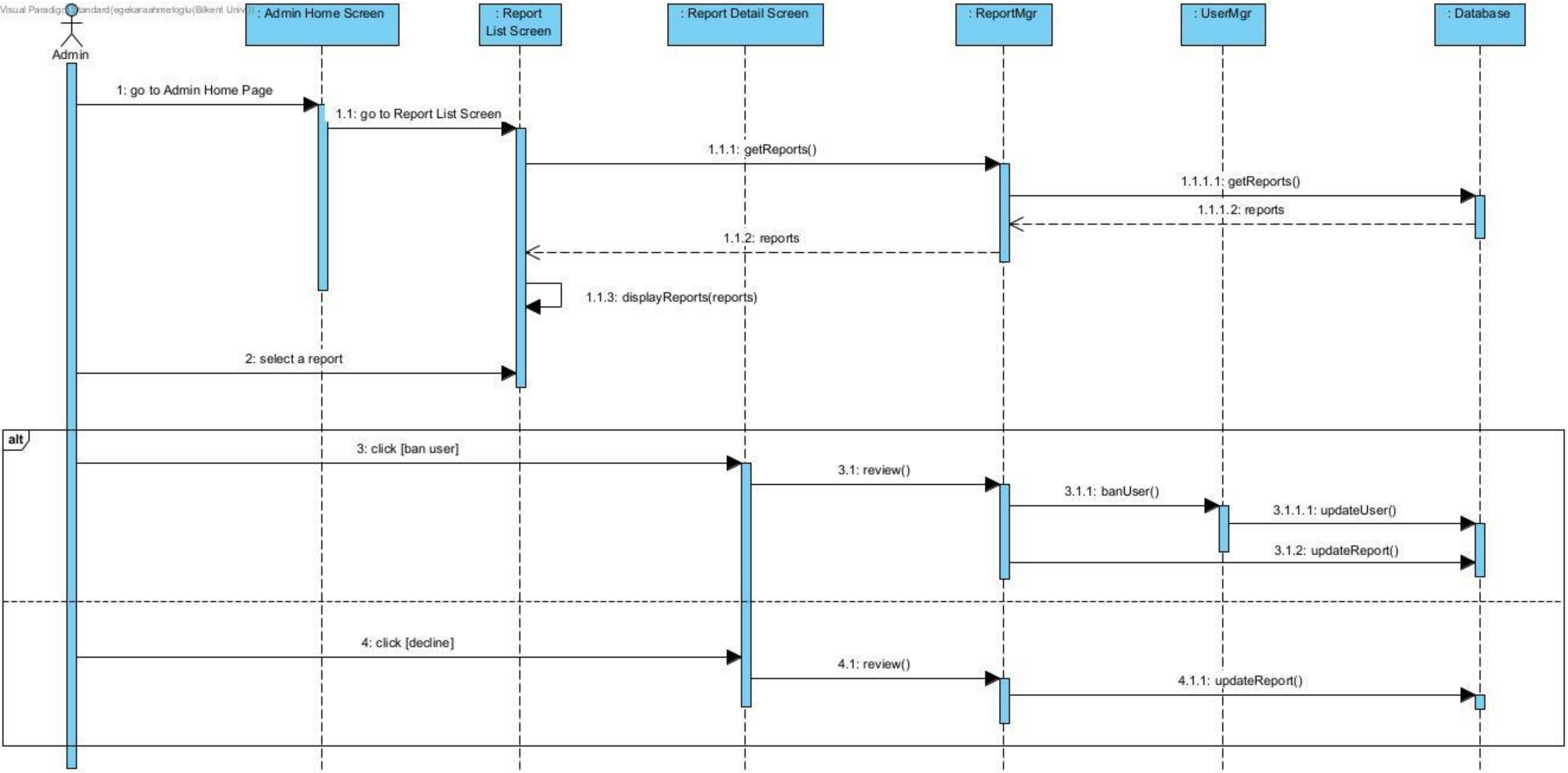


Figure 5 Ban User Sequence Diagram

Activity Diagrams

Create Post Activity Diagram

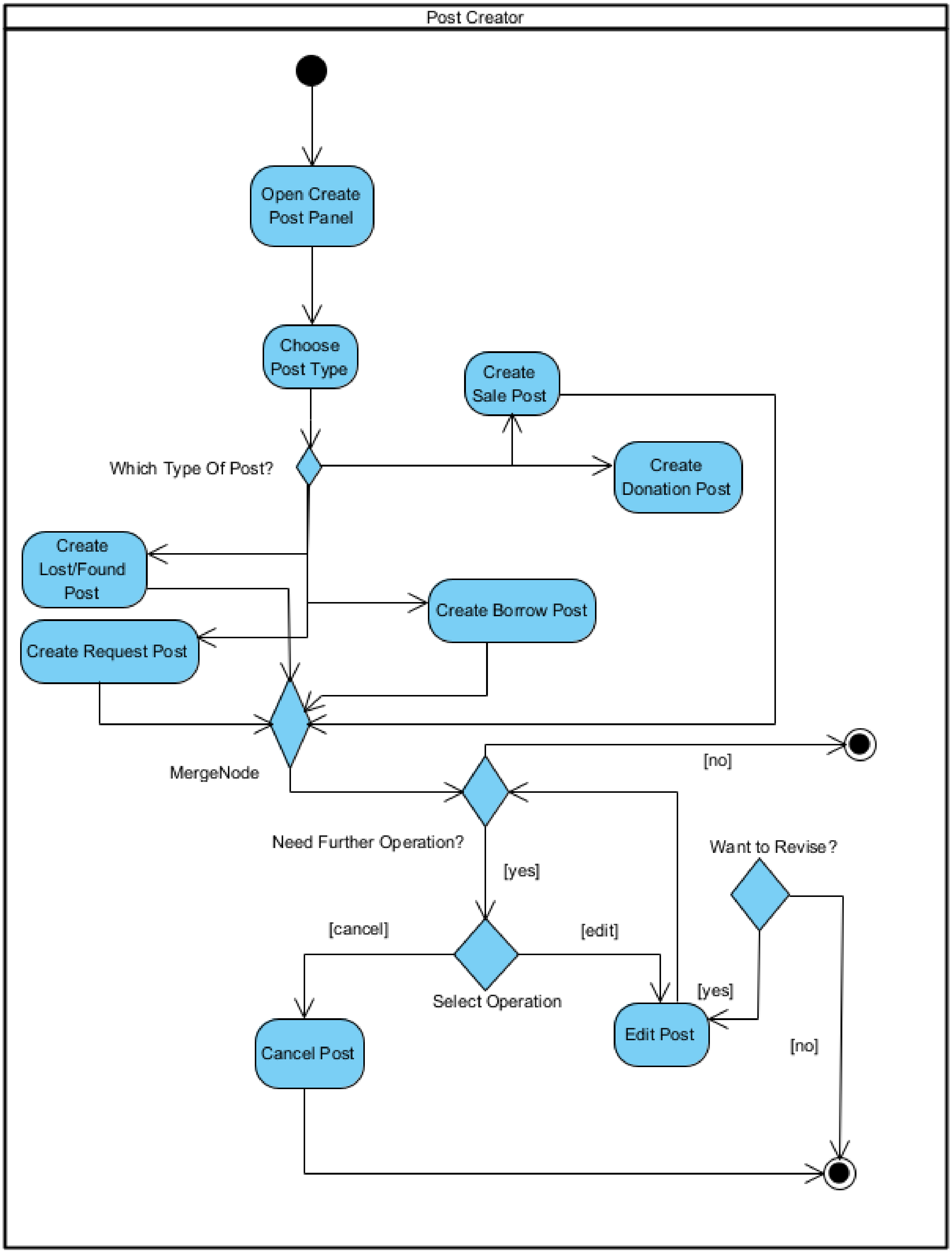


Figure 6 Create Post Activity Diagram

Communication Activity Diagram

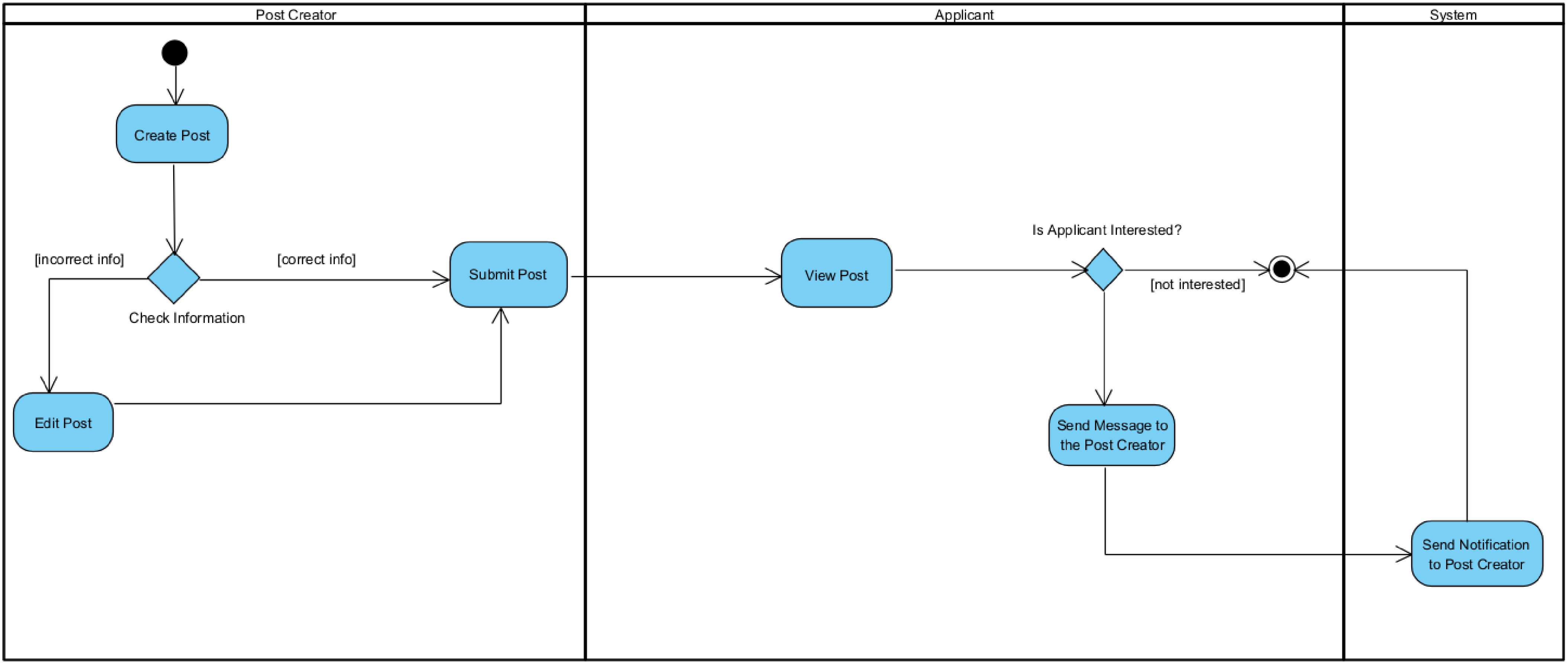


Figure 7 Communication Activity Diagram

Report Post Activity Diagram

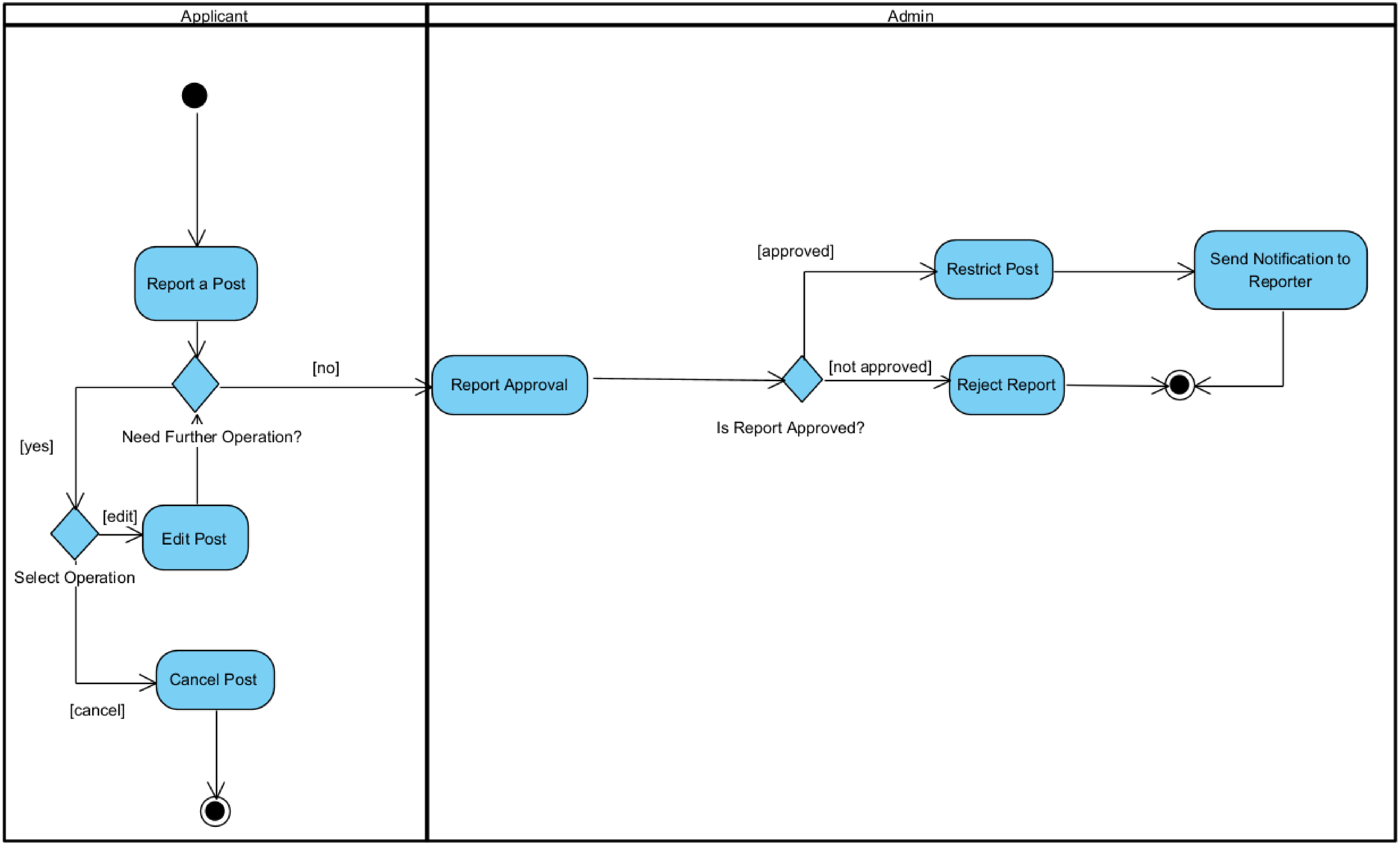


Figure 8 Report Post Activity Diagram

State Diagrams

Post State Diagram

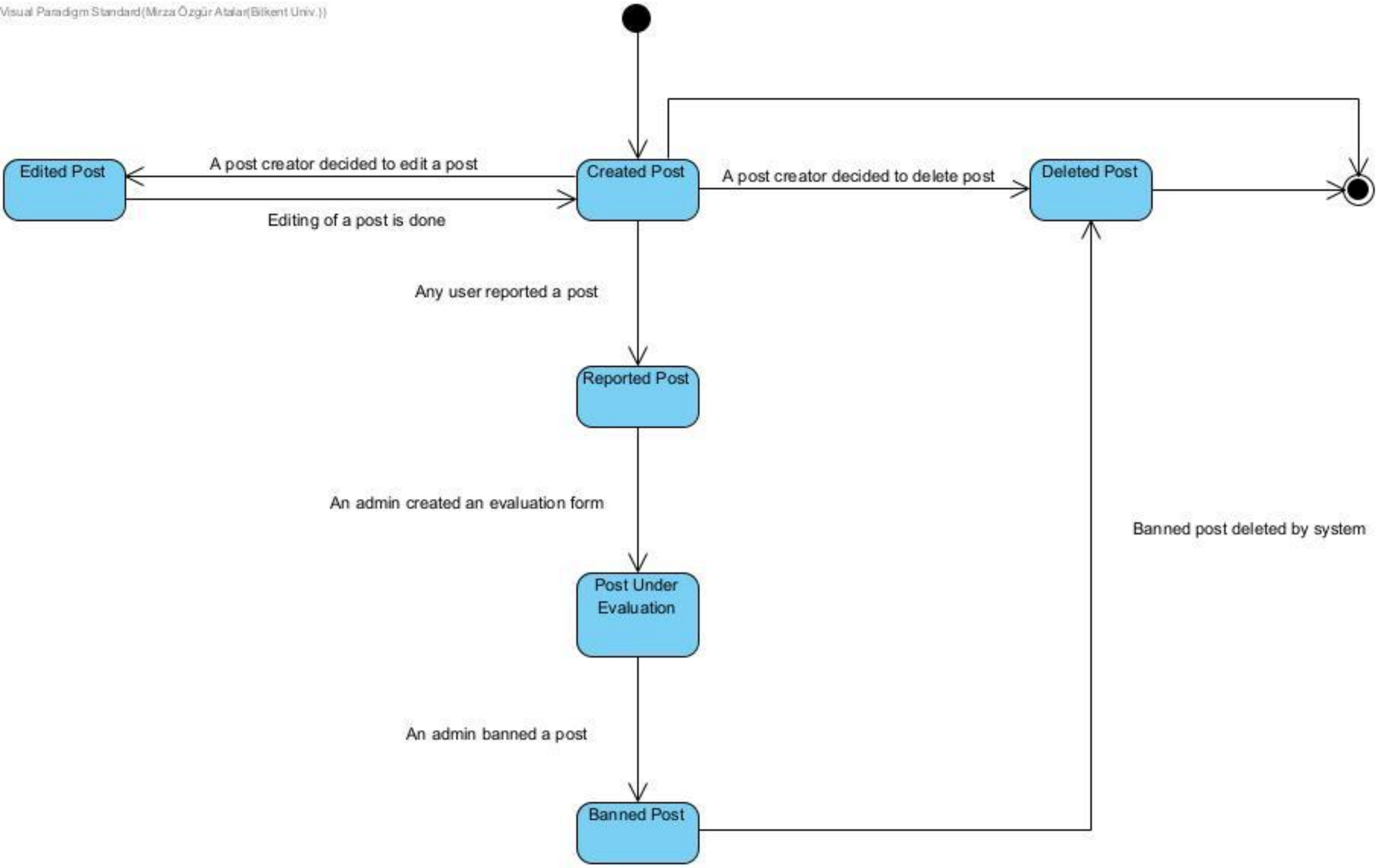

Visual Paradigm Standard(Mirza Özgür Atalar(Bilkent Univ.))
Edited Post
A post creator decided to edit a post
Editing of a post is done
Created Post
A post creator decided to delete post
Deleted Post
Any user reported a post
Reported Post
An admin created an evaluation form
Post Under Evaluation
An admin banned a post
Banned Post
Banned post deleted by system

Figure 9 Post State Diagram

Report State Diagram

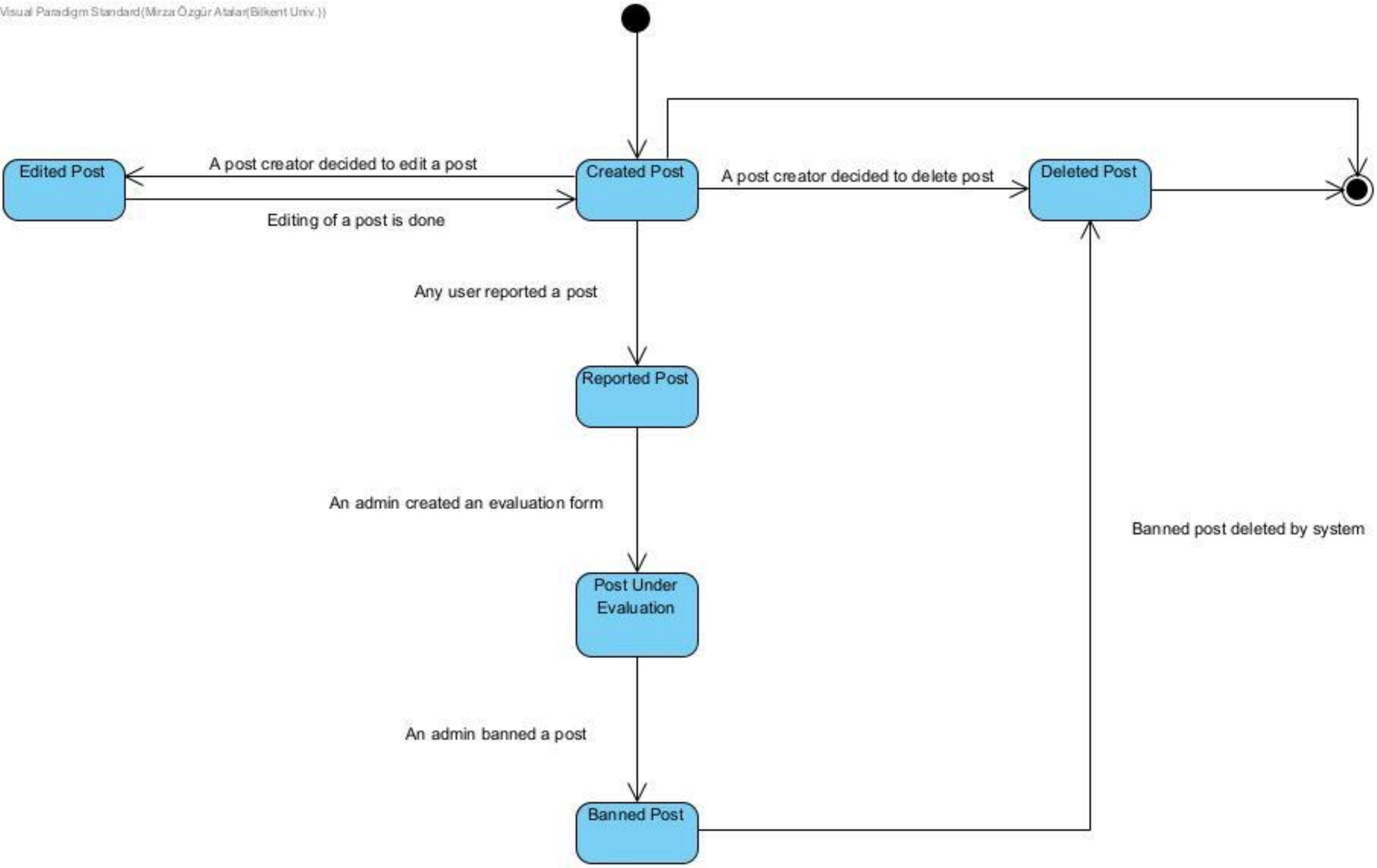
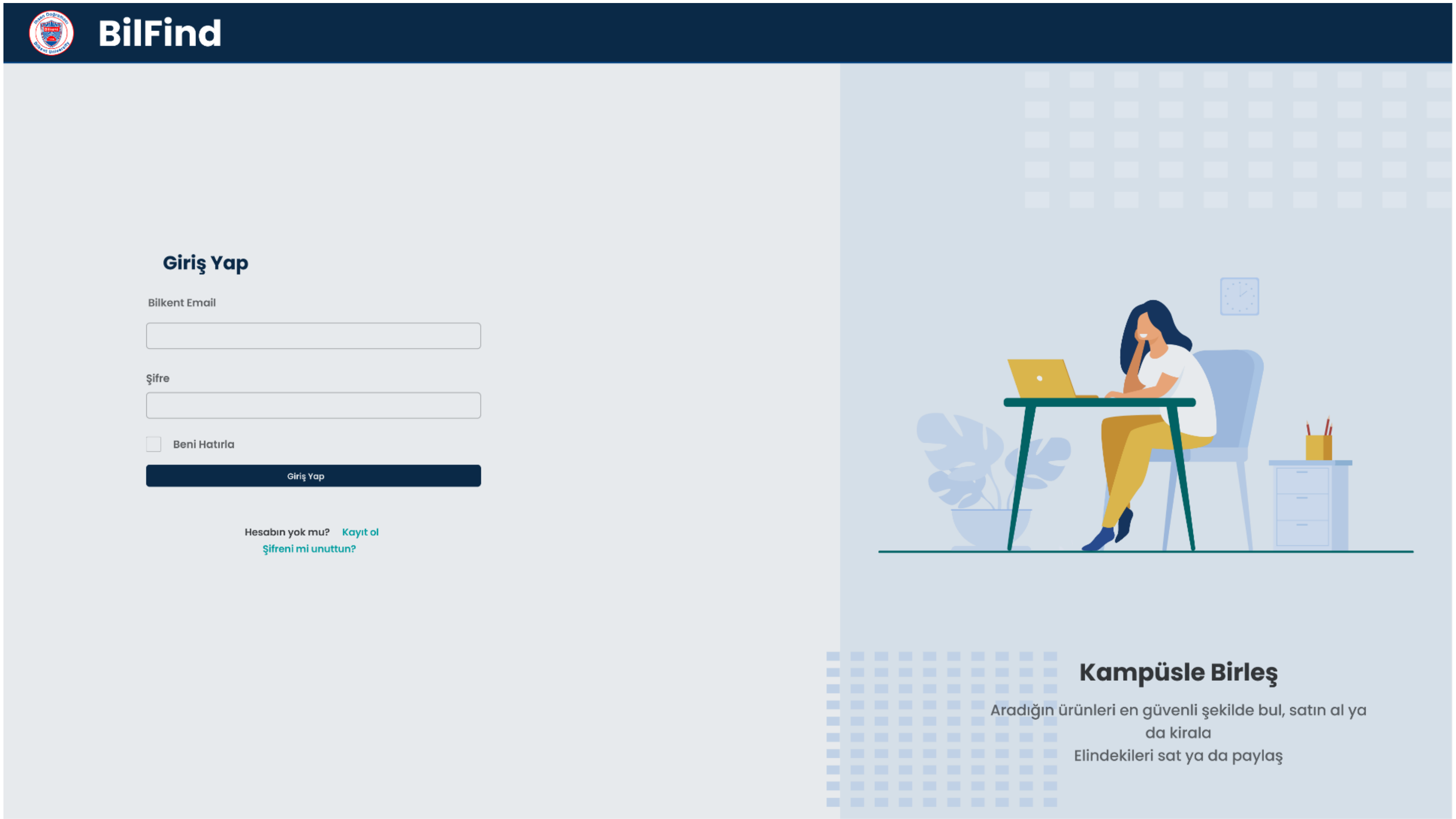

Visual Paradigm Standard(Mirza Özgür Atalar(Bilkent Univ.))
Edited Post
A post creator decided to edit a post
Editing of a post is done
Created Post
A post creator decided to delete post
Deleted Post
Any user reported a post
Reported Post
An admin created an evaluation form
Post Under Evaluation
Banned post deleted by system
An admin banned a post
Banned Post

Figure 10 Report State Diagram

Mockup UI

Login Page

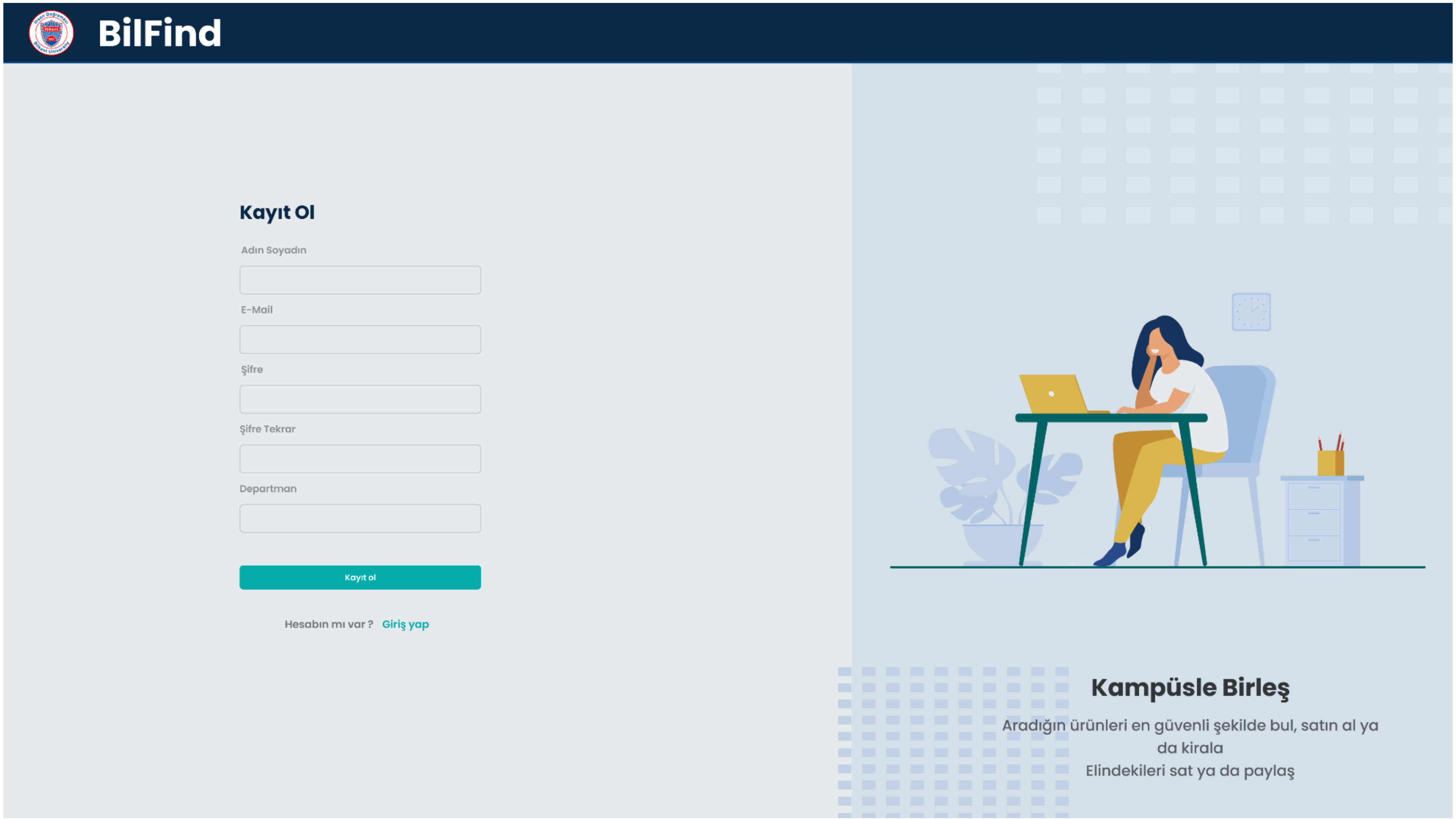


Kampüsle Birleş

Aradığın ürünleri en güvenli şekilde bul, satın al ya da kirala
Elindekileri sat ya da paylaş

Figure 11 Login Page

Sign Up



BilFind

Kayıt Ol

Adın Soyadın

E-Mail

Şifre

Şifre Tekrar

Departman

Kayıt ol

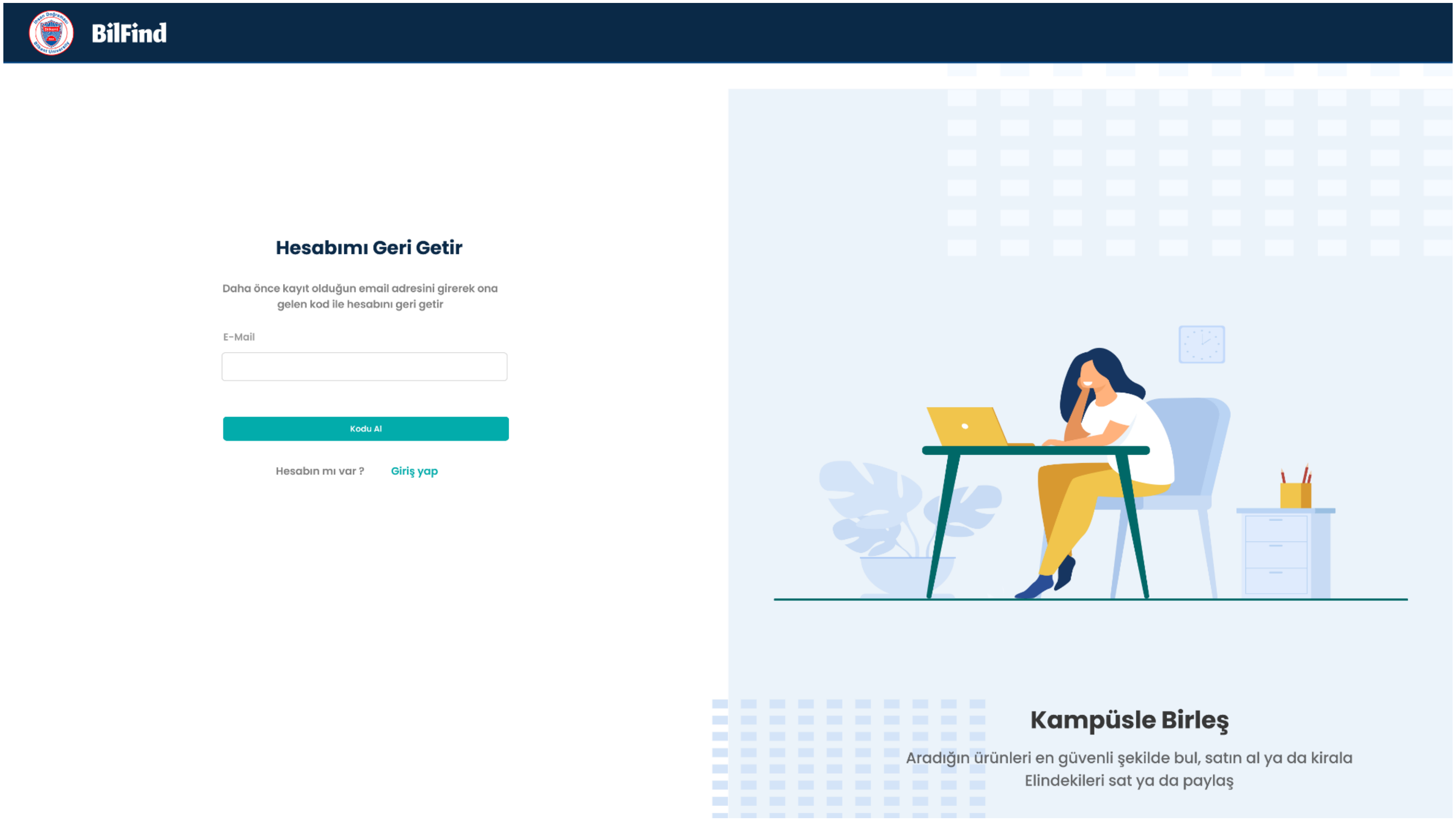
Hesabın mı var ? Giriş yap

Kampüsle Birleş

Aradığın ürünleri en güvenli şekilde bul, satın al ya da kirala
Elindekileri sat ya da paylaş

Figure 12 Sign Up Page

Forgot Password



Hesabımı Geri Getir

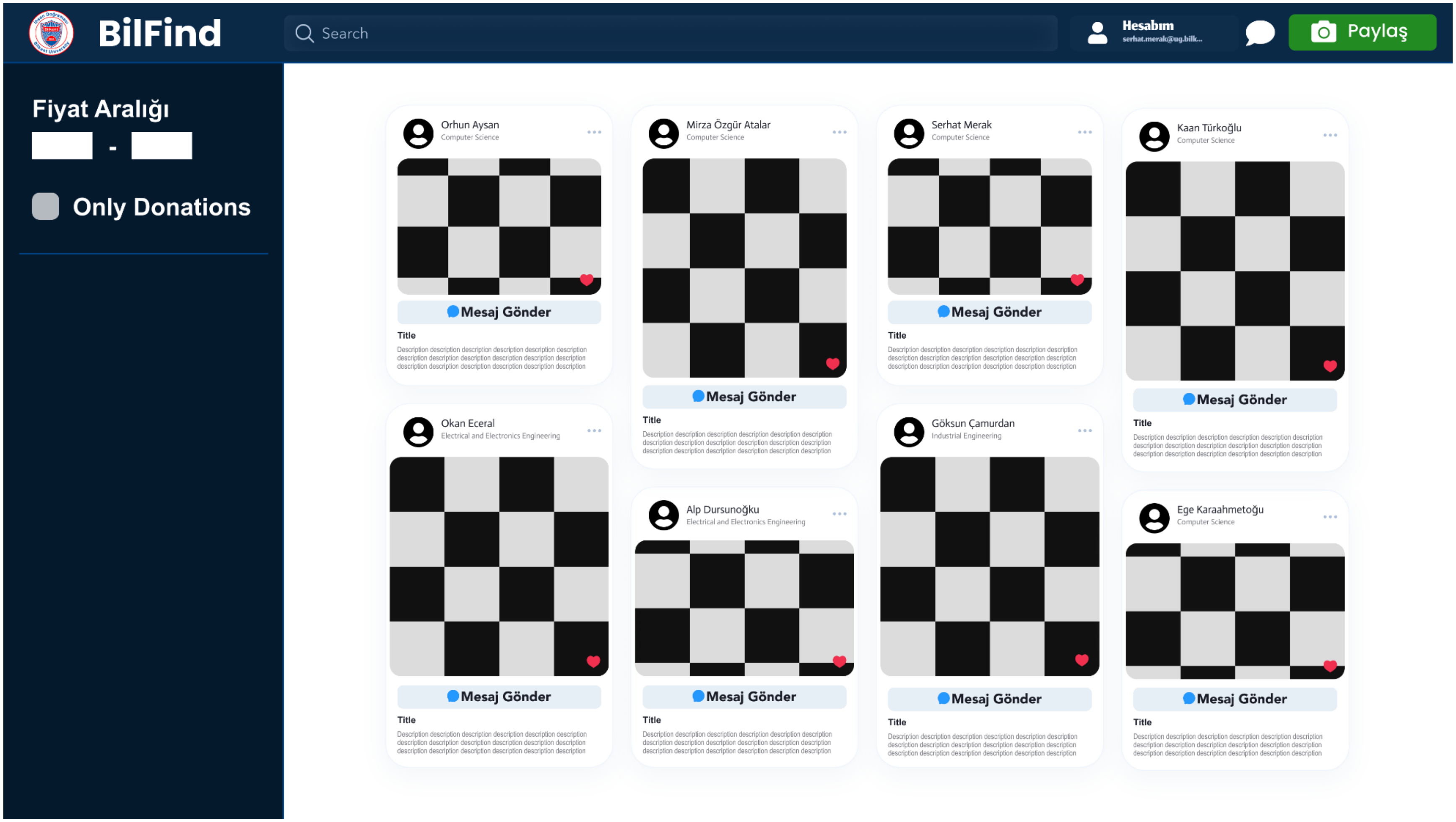
Daha önce kayıt olduğun email adresini girerek ona gelen kod ile hesabını geri getir

Kampüsle Birleş

Aradığın ürünleri en güvenli şekilde bul, satın al ya da kirala
Elindekileri sat ya da paylaş

Figure 13 Forgot Password Page

Main Page



Orhun Aysan
Computer Science
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Okan Eceral
Electrical and Electronics Engineering
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Mirza Özgür Atalar
Computer Science
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Alp Dursunoğku
Electrical and Electronics Engineering
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Serhat Merak
Computer Science
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

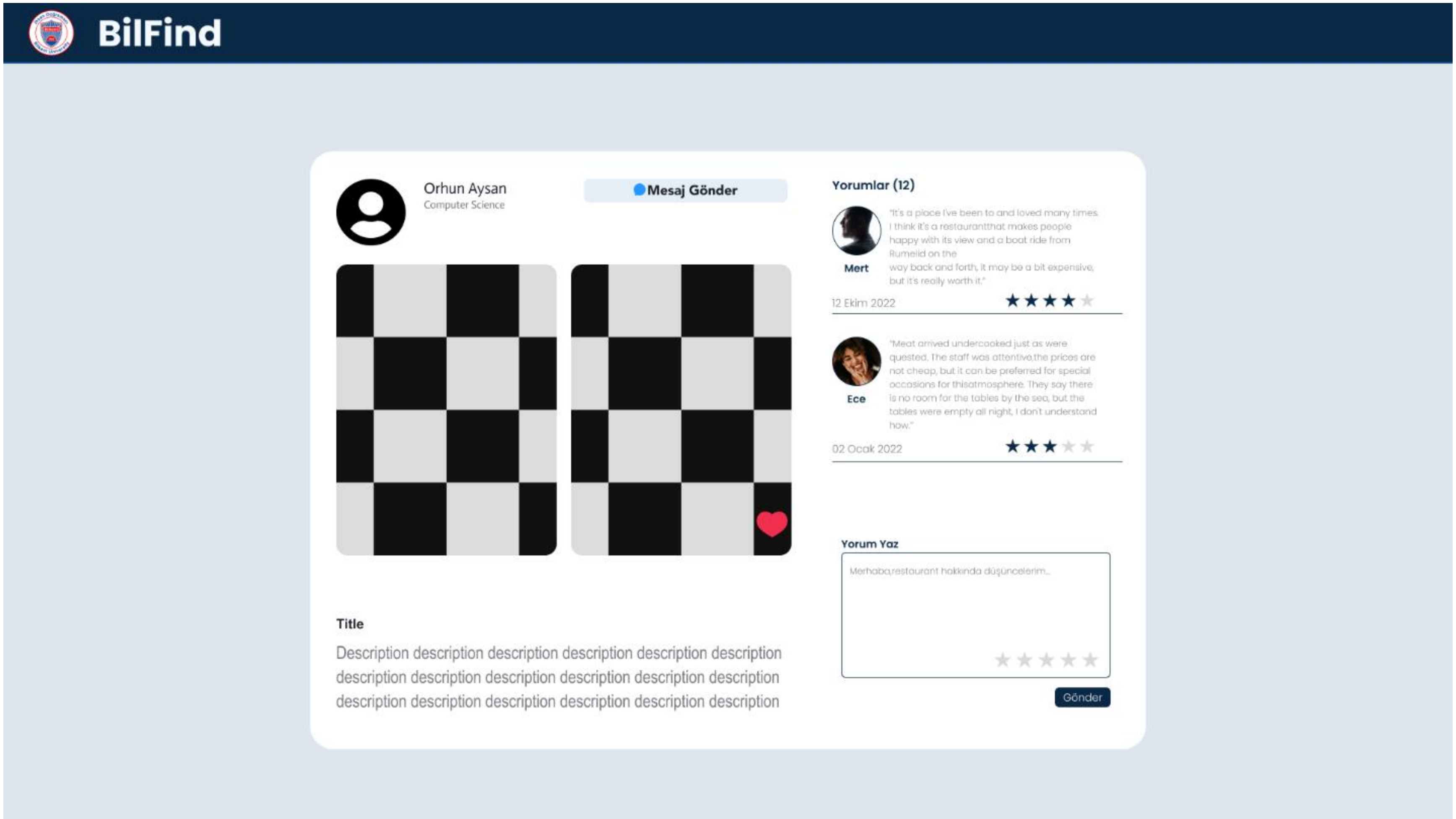
Göksun Çamurdan
Industrial Engineering
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Kaan Türkoğlu
Computer Science
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Ege Karaahmetoğu
Computer Science
Mesaj Gönder
Title
Description description description description description description description description description description description description description description description description description description

Figure 14 Main Page

Detailed Post



"It's a place I've been to and loved many times. I think it's a restaurantthat makes people happy with its view and a boat ride from Rumelid on the
way back and forth, it may be a bit expensive, but it's really worth it."

"Meat arrived undercooked just as were quested. The staff was attentive,the prices are not cheap, but it can be preferred for special occasions for thisatmosphere. They say there is no room for the tables by the sea, but the tables were empty all night, I don't understand how."

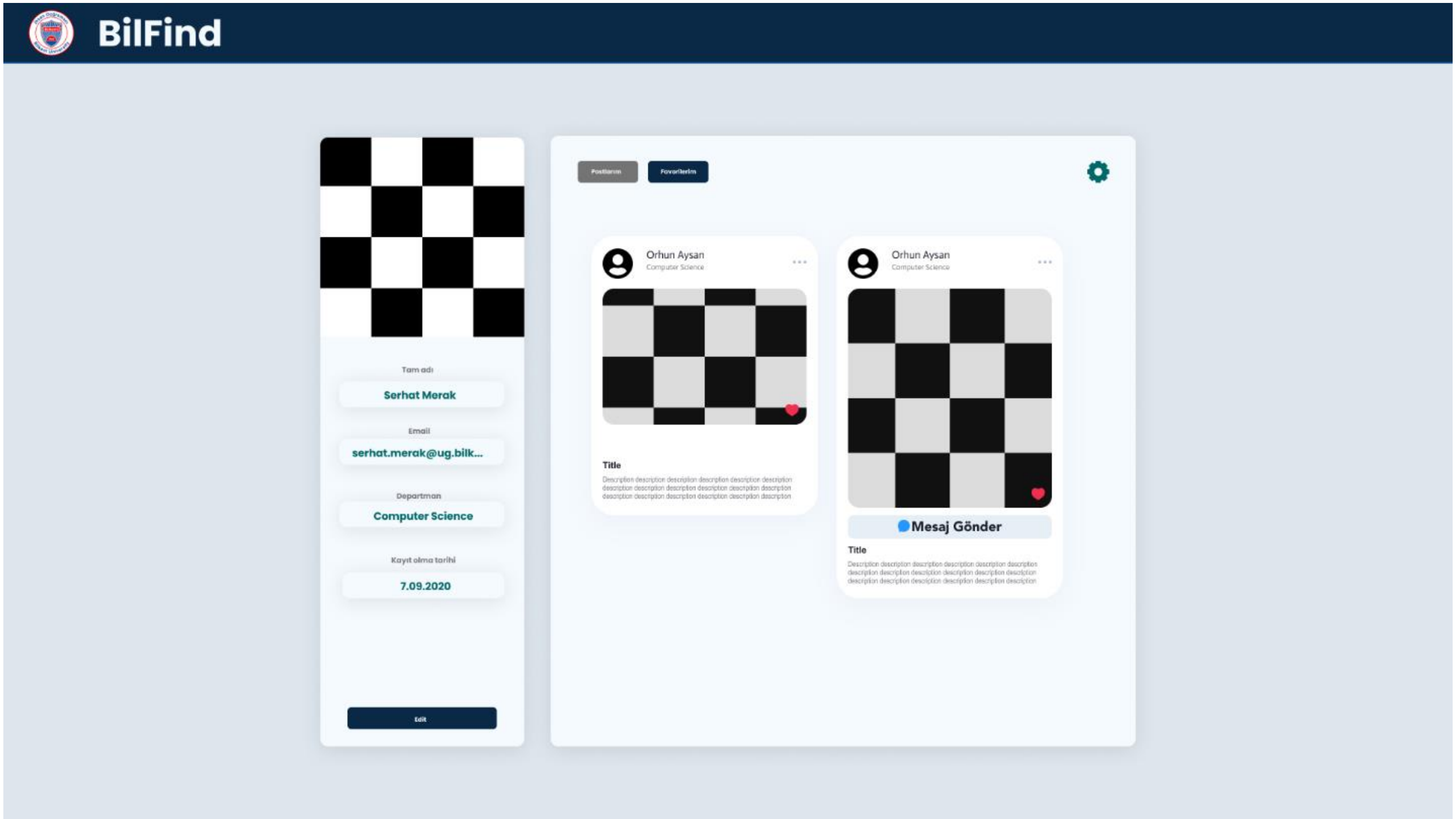
Merhaba,restaurant hakkında düşüncelerim...

Title

Description description description description description description description description description description description description description description description description description description

Figure 15 Detailed Post Page

User’s Favorite and Own Posts



Tam adı
Serhat Merak

Email
serhat.merak@ug.bilk...

Departman
Computer Science

Kayıt olma tarihi
7.09.2020

Orhun Aysan
Computer Science

Title

Description description description description description description description description description description description description description description description description description description

Orhun Aysan
Computer Science

Title

Description description description description description description description description description description description description description description description description description description

Figure 16 User's Favorite and Own Posts Page

Create Post

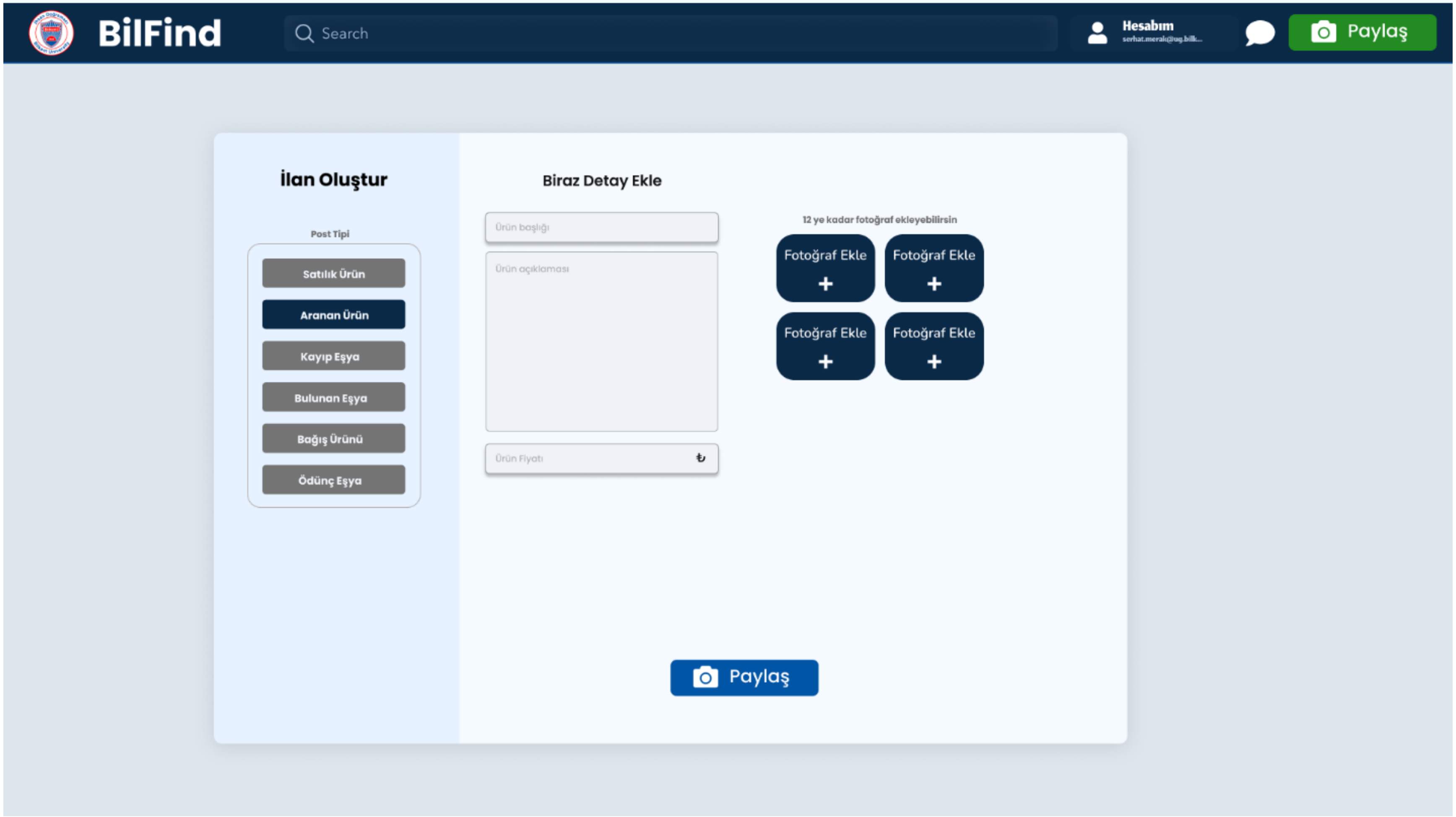
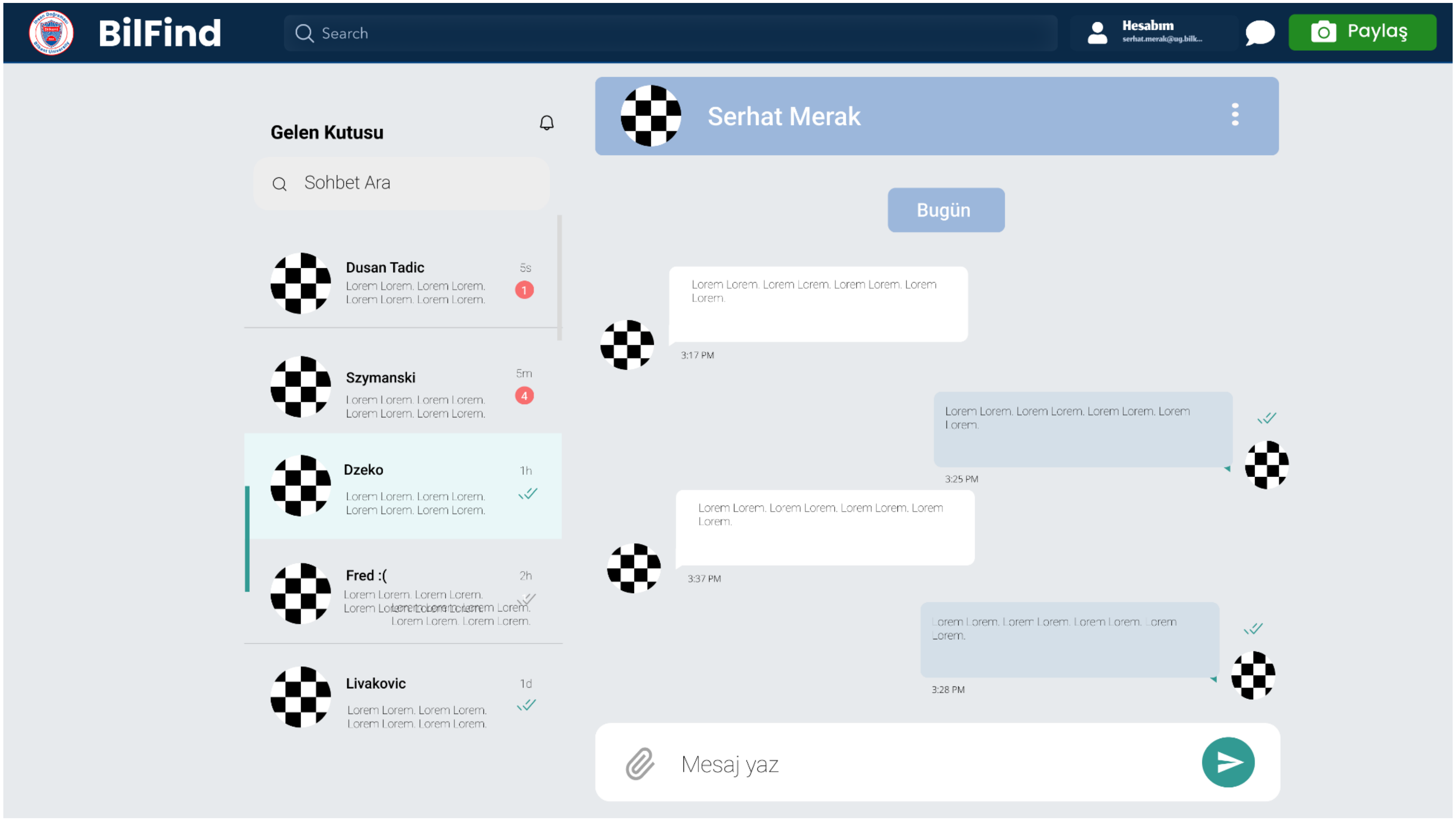


Figure 17 Create Post Page

Chat Menu



Dusan Tadic 5s 1
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.

Szymanski 5m 4
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.

Dzeko 1h
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.

Fred :(2h
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.

Livakovic 1d
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.

Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.
3:17 PM

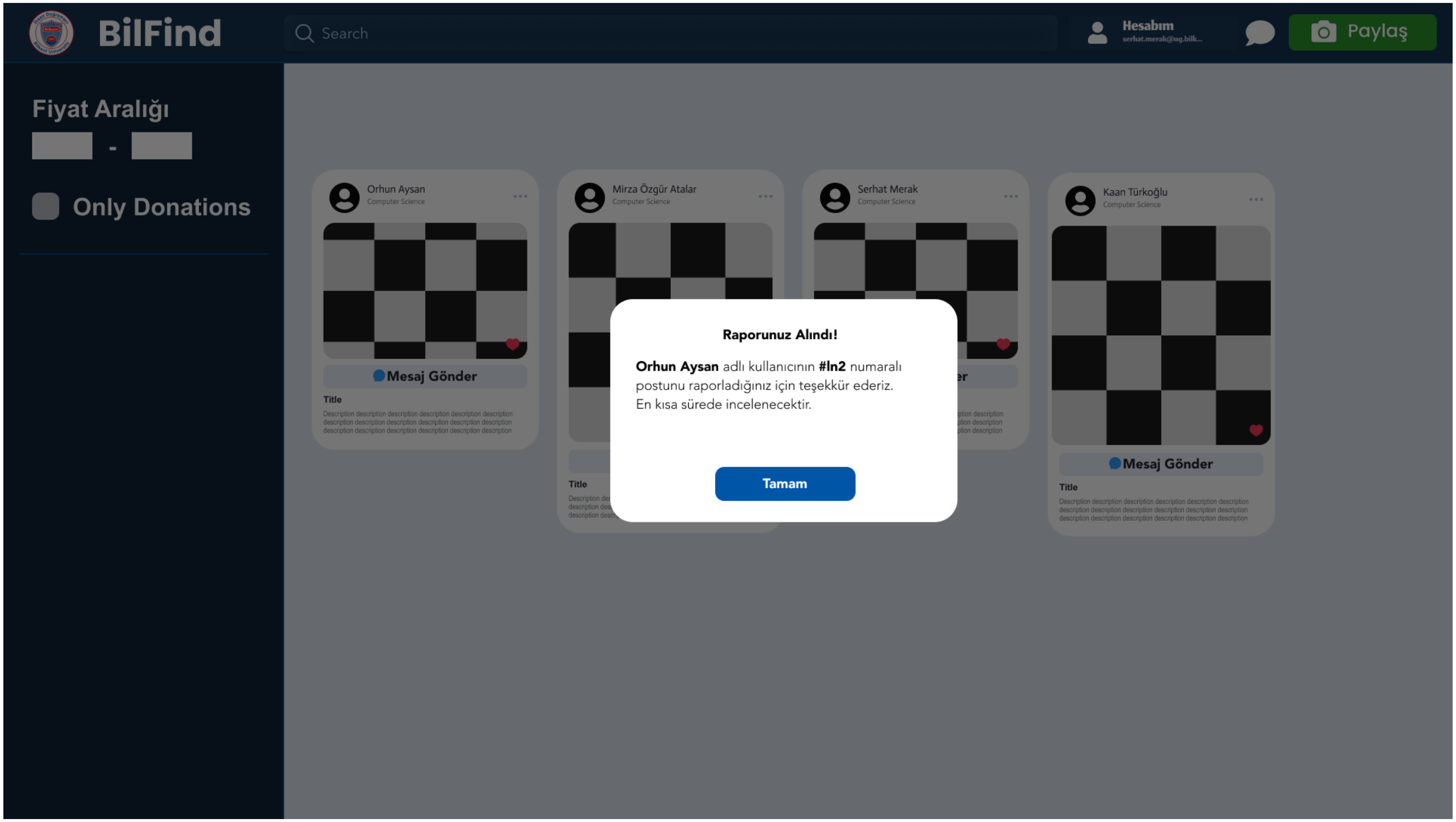
Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.
3:25 PM

Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.
3:37 PM

Lorem Lorem. Lorem Lorem. Lorem Lorem. Lorem Lorem.
3:28 PM

Figure 18 Chat Menu Page

Report Submission Pop-Up



Raporunuz Alındı!

Orhun Aysan adlı kullanıcının **#In2** numaralı postunu raporladığınız için teşekkür ederiz. En kısa sürede incelenecektir.

Figure 19 Report Submission Pop-Up Sample

Post Menu Pop-Up

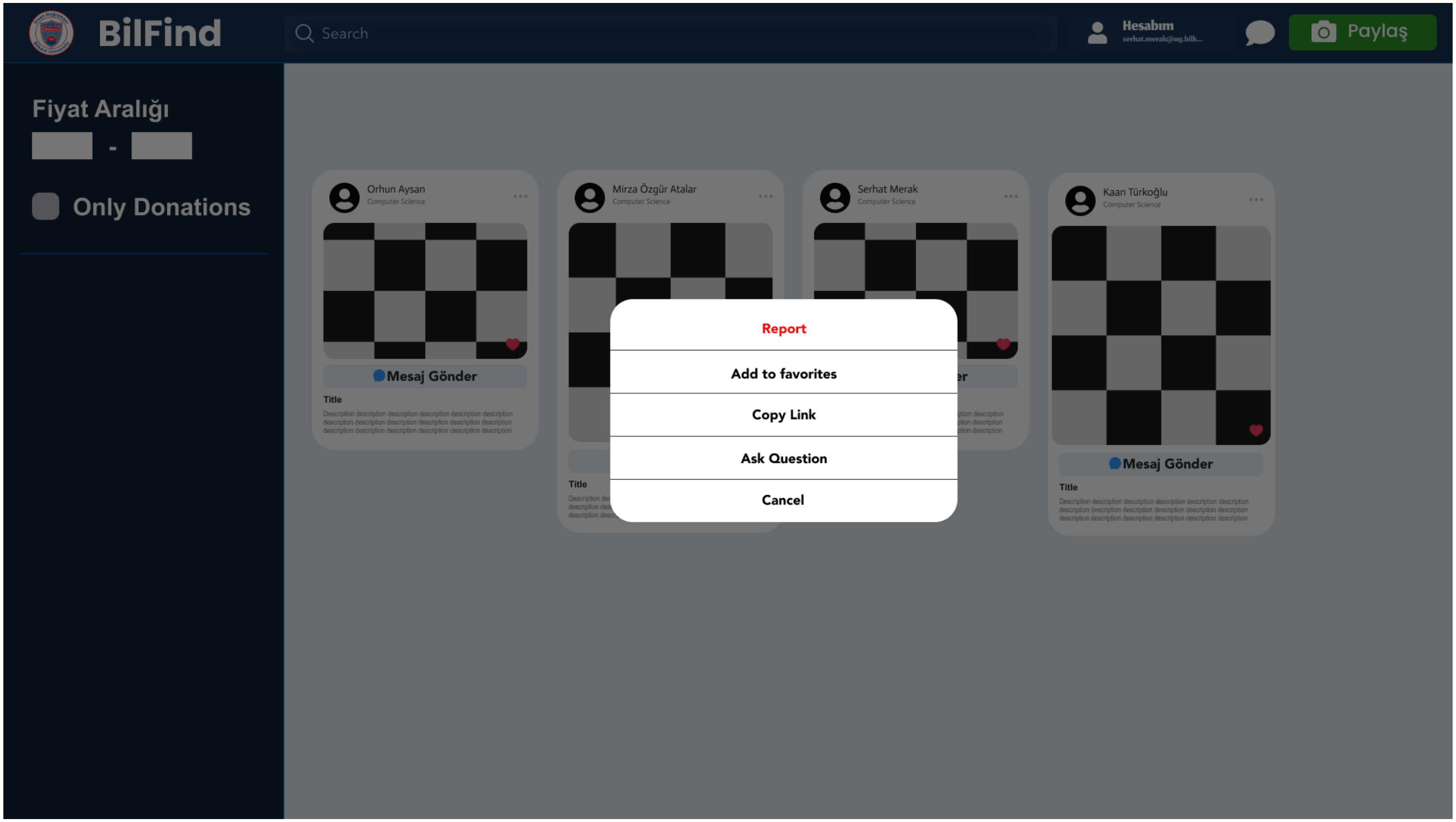
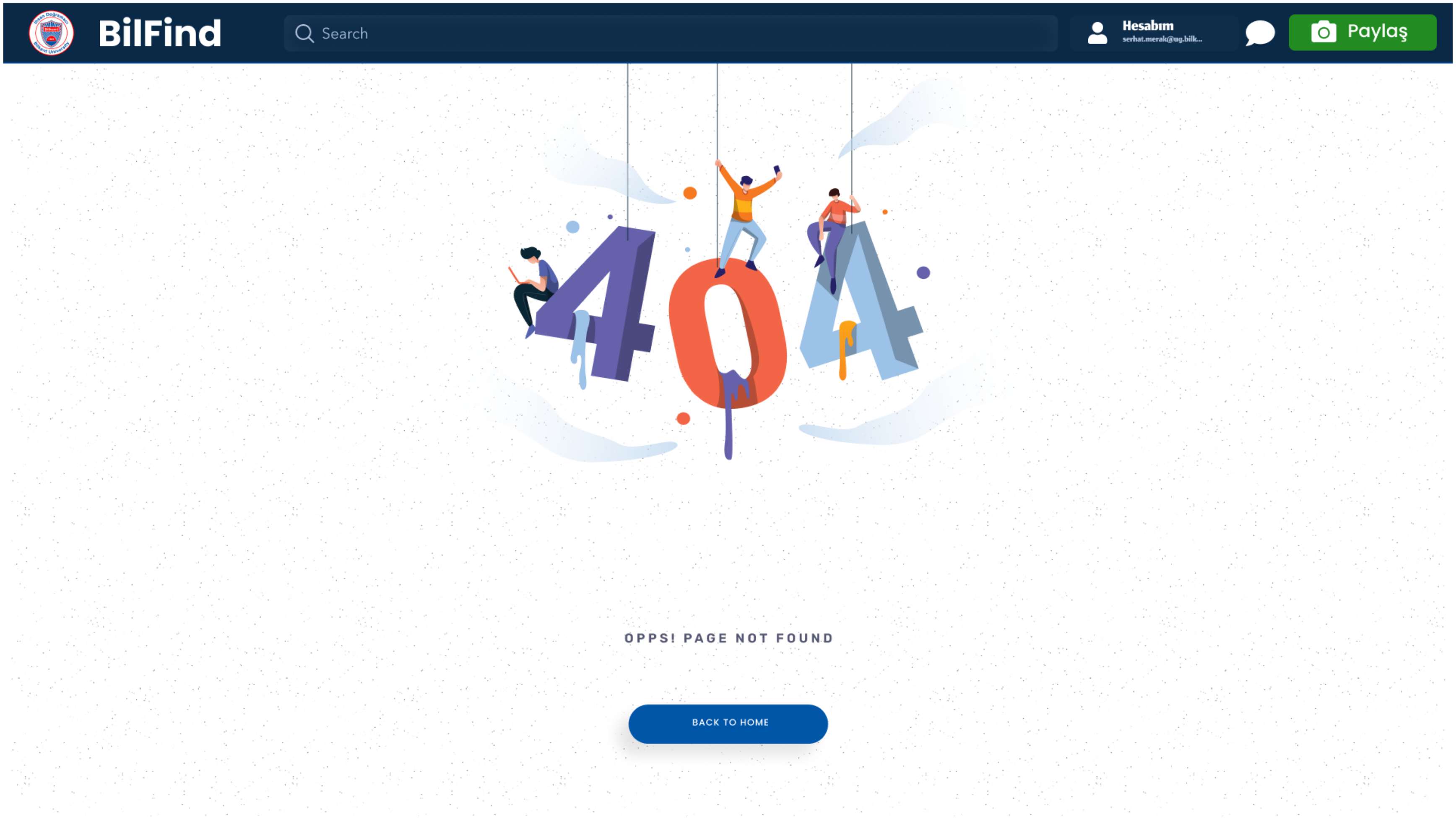


Figure 20 Post Menu Pop-Up Sample

Error Page



OPPS! PAGE NOT FOUND

Figure 21 Error Page

Non-Functional Requirements

1) Performance:

- **Response Time:** The application should have fast response times for user interactions, with a goal of under 5 seconds for most of the actions both in the front end and the back end.
- **Scalability:** The system should be scalable to accommodate a growing number of users and items in the marketplace.
- **Load Handling:** The application should be able to handle a large number of concurrent users without degrading performance.

2) Security:

- **Data Security:** User data, especially personal and financial information, will be securely stored and transmitted with the help of modern and secure encryption and database systems such as MongoDB.
- **Authentication:** Secure user authentication will be provided, and only users from Bilkent University will be authenticated via their Bilkent-assigned e-mails. Third-party members unrelated to Bilkent University will not be able to interfere with the user data.
- **Privacy:** User privacy and data protection must be a top priority, complying with relevant data protection regulations.

3) Usability:

The application should have a user-friendly and visually appealing interface, to make users comfortable upon using the application. Also, in the interface, each app feature should be easily distinguished and provide clear information about themselves.

4) Reliability:

- **Availability:** The system should be available 24/7 with minimal downtime for maintenance.
- **Data Integrity:** All user data, listings, and messages should be stored securely and reliably. Data loss or corruption should be prevented via frequent database backups.

5) Maintainability:

Throughout the implementation, Object-Oriented Programming will be used to extend and improve the features in time. Moreover, the codebase will be well-documented and follow coding standards for easy maintenance and for further development. These approaches will also make debugging and fixing bugs easier, as it will facilitate finding errors in the code.

Tech Stack

We decided on our tech stack primarily considering the limited time and the flexibility requirements. We also wanted to use modern technologies in the continuously developing world.

Flutter (Frontend):

In our front-end application, we decided to use Flutter for its flexibility. Our main product will be a web application. However, in today's world, we observed that the mobile application for several needs at Bilkent University would be as efficient as the web application. Therefore, we needed to create a web, Android, and iOS application in a limited time, so Flutter is one of the best options to ensure this aim. It is possible to adopt the written code to both mobile and web applications with only small adjustments. Also, it is obvious that almost every application follows similar components. Therefore, with the very rich open-source library, Flutter was the best option for us to develop our application with the least effort.

NodeJS (Backend):

Our backend will be structured over NodeJs technology supported with multiple libraries. By using **ExpressJs**, we will create a flexible and well-structured RESTful API for our application. Since we will write the backend considering the Model View Controller (MVP) structure, it is quite readable and scalable to write with the Express library. However, using solely JavaScript in the backend may produce some ambiguity among the group members because it would be harder to maintain code as it grows without type-checking. Therefore, we decided to utilize **TypeScript** for our backend project to use static typing and Object-Oriented Concepts in the backend architecture.

MongoDB (Database):

In our application, we observed that we need flexible data models because even though we are planning the whole development process, it is likely for data models to evolve over and over. Therefore, MongoDB’s support for flexibility was the key factor in our decision. Additionally, with the query optimization support, it is easier to scale the datasets while protecting the efficiency. Also, MongoDB is very suitable for startups because it is free up to a certain level