



CSE 343 – 3rd Presentation Report

Los Ingenieros Hermanos

Project Link:

<https://github.com/orgs/los-ingeneros-hermanos/repositories>

Group Members:

Muhammed Akif Sevilgen (Project Manager)

Burak Kocausta (Product Owner)

Ahmet Kadir Aksu

Ersel Celal Eren

Şamil Berat Delioğulları

Yusuf Arslan

Contents:

- a) User Stories _____
- b) Context Model _____
- c) Use Case Diagrams and Detailed Tables _____
- d) Sequence Diagrams _____
- e) Class Diagrams _____
- f) Test Cases (Updated) _____
- g) State Diagrams (NEW) _____
- h) Simple System Architecture (NEW) _____
- i) CMIS Database Backend API (Updated) _____
- j) Backend Tests (NEW) _____
- k) Web Tests (NEW) _____
- l) Mobile Tests (NEW) _____

User Stories and Related Scenarios

Story #1 – Profile Page of Community

Başak is part of the IAAS Community in GTU. IAAS brings together students studying, specializing, or researching in related fields such as agricultural and environmental sciences, etc.

She wants a list of all communities, and when a user clicks IAAS, the profile page of her community should open. It should include what they are doing, contact information, date of foundation etc. There must be tags for the community. For example, it can be scientific, agricultural etc. Students or newcomers can find the community by adding these tags to their interests.

Page must have an option to see upcoming and past events. Students can follow her community, and IAAS's followers should see their events, announcements on their main page. Moreover, she wants students to apply to her community through the application. Also, there must be an option to directly communicate with students through the app.

Scenario #1 – Profile Page of Community in GTU-CMIS App

Initial assumption: Community is signed in to GTU-CMIS application. Users of the app who want to view communities profile pages are signed up and logged in mobile or web applications.

Normal: User clicks communities tab in the main page, then sees all communities names. Click one of them, then a profile page of the community will be open. There is a button for applying, if the user is not a part of the community, it proceeds. If a community has a google form, or another form link, it will be shared. If not, default application form will be used. There must be a restriction to send application forms that can be sent by a user to a particular community, and communities can organize a time interval that they can only accept applications on that time interval.

Each community has defined tags, according to these tags users can search the communities through these tags.

Users can directly message to the community, there will also be a button on the profile page for this. On one part of the page, an overview or description of the community can be seen. They are written by communities. There could be default descriptions that can be provided by application. There is an option to see events that the community organized. There must be a button or another sign to see past or future events. And there must be a follow button for the user. Following, and applying is different.

What can go wrong: There could be a broken application link that is shared by the community, in this case the user can have an option to inform the community. Community has no past or future events. User sees the empty event page. Users can abuse the application form option, in this case every user has time out for applying, and communities have timelines for applications. User applies to the community and cannot get feedback for a long time. A warning option can be sent to communities for giving feedback after a definite number of days are passed after application.

Other activities: Another user may do the same things at the same time for a particular community.

System state on completion: User applied to the community, application is added to the application list of the community. User followed community, and community is added to the user's "followed communities" list. The community's follower number has increased. Direct message is sent to the community, and it is seen, can be answered by the community.

Story #2 – Profile Page of User

Can is a new student in Gebze Technical University. His department is computer engineering, but he is in the preparatory class now. He wants to join student clubs in order to meet new people and take part in organizations. His desire is, seeing the list of the communities with their events and organizations, so he could choose the best clubs that match his hobbies and interests. Also, he would like to see the events and announcements of communities he is interested in.

He wants to create a profile, add his information, hobbies, and interest, and also his project idea - which he needs team members to carry out-.

Scenario #2 – Profile Page of User in GTU-CMIS App

Initial assumption: The user has successfully launched the GTU-CMIS app(web or mobile)

Normal: After the user launches the app, he/she sees posts of GTU communities on the homepage and if he/she wants to follow one of the communities he/she is asked to log in or register. If the user has an account, he/she needs to enter his/her email address and his/her password for his/her account. If the user does not have an account, he/she needs to create a profile, then the user chooses an avatar, enters his/her information, hobbies and interests, and his/her project idea if he/she has any. After the user creates a profile, the application recommends the user appropriate communities and teams according to the hobbies and interests of the user. Of course, the user can list all of the communities when he/she clicks on all communities button.

The user goes to his homepage, sees the announcements of the communities he/she follows. The user chooses global, then he/she sees the posts of all communities. The user chooses membership, then he/she sees the posts of the communities which he/she is a member of on his/her homepage.

The user can directly interact with communities via message option.

The user adds some events from a community page into his/her calendar to follow upcoming events.

The user bookmarks a post. The user goes to the bookmarks page to see the posts he/she bookmarked.

The user searches for a specific community with the search button.

What can go wrong: Some users can give misinformation about themselves. Some users can lie about their project ideas, or some users can send spam messages to others.

System state on completion: There is a user with a profile picture, his hobbies and interests, and his project idea. The event which the user chooses, added on to the user's calendar. The posts user bookmarked, are seen on the bookmarks page of the user. The communities user followed add on the following communities of the user.

Story #3 – Announcement and Calendar System

Buse is part of a drone team in Gebze Technical University. The team participates in global and local competitions. She is responsible for making announcements and keeping track of events in the team.

She wants to share the upcoming events and calendar to the members of the team. The members get notifications for the upcoming events if they want, and they can check the calendar for the events.

Scenario #3 – Announcement and Calendar System in GTU-CMIS App

Initial assumption: The user has logged in GTU-CMIS web or mobile application. The user has permission to make announcements and add events into the community's calendar.

Normal: The user enters his/her community's page and clicks on the new announcement button, upload pictures and add text to the post, then shares on the page. The announcement will be seen on the communities' page and also users who follow the community will see the announcement on their homepage.

The user navigates to the calendar tab of the community. He/she chooses a date and then clicks on the “+” button to create a new event. The application asks users to enter title, description, and upload photos. After the user completes these, clicks the “publish” button.

The user chooses an event in the calendar and then clicks the “delete” button to delete the event.

What can go wrong: When creating an event or post, if the title or description parts are empty, the “publish” button will not be available.

System state on completion: The announcements are seen on the homepage of the users who followed the community. The event is added on the community's page and calendar and the event is shared as a post. The users can like or bookmark the events. The users are able to add the event to their calendar. There will be an attendance option on the events and the members of the club can mark themselves as willing to attend.

Story #4 – Management of Community

Aslı is part of the Social Science Community. It is a community that operates in the fields of social sciences such as psychology, sociology, economy, history, geography.

She wants to organize a business model for her community. Organizing means that she can add/remove members, or change the role of any member in the system, also she can assign some tasks for the members of the community like arrange an announcement, meeting, or send mails about some topic to these people etc. Permissions of these members cause the possibility of doing some actions or restriction of doing these particular actions. She needs to make some of the members qualified. In this way she can create a hierarchical relationship between members. Some of the meetings can be compulsory for the community, and it is nice to hold attendances of members in the community. Even if it is not compulsory, overall participation of each member should be seen on the application.

Scenario #4 – Management of Community in GTU-CMIS App

Initial assumption: Members are contributors of the same community with admin. Admin created the community.

Normal: The admin of the community/team changes permissions and makes some members more authorized. There are 3 types of users in the community system, which are admin, authorized member, member. Authorized members, admin can do actions which are adding, removing, updating, assigning. These actions are for the management of members. When the admin opens the management screen, he/she sees all the members, and assigns these tasks to which member that he/she wants. Additional actions are updating calendar, removing members, accepting applicants, publishing an announcement, updating information about the community. Members which can assign custom tasks, send this custom task, and other members that assigned this task are informed about that. The user finishes a task, he/she commits the information that task is finished with clicking the done button.

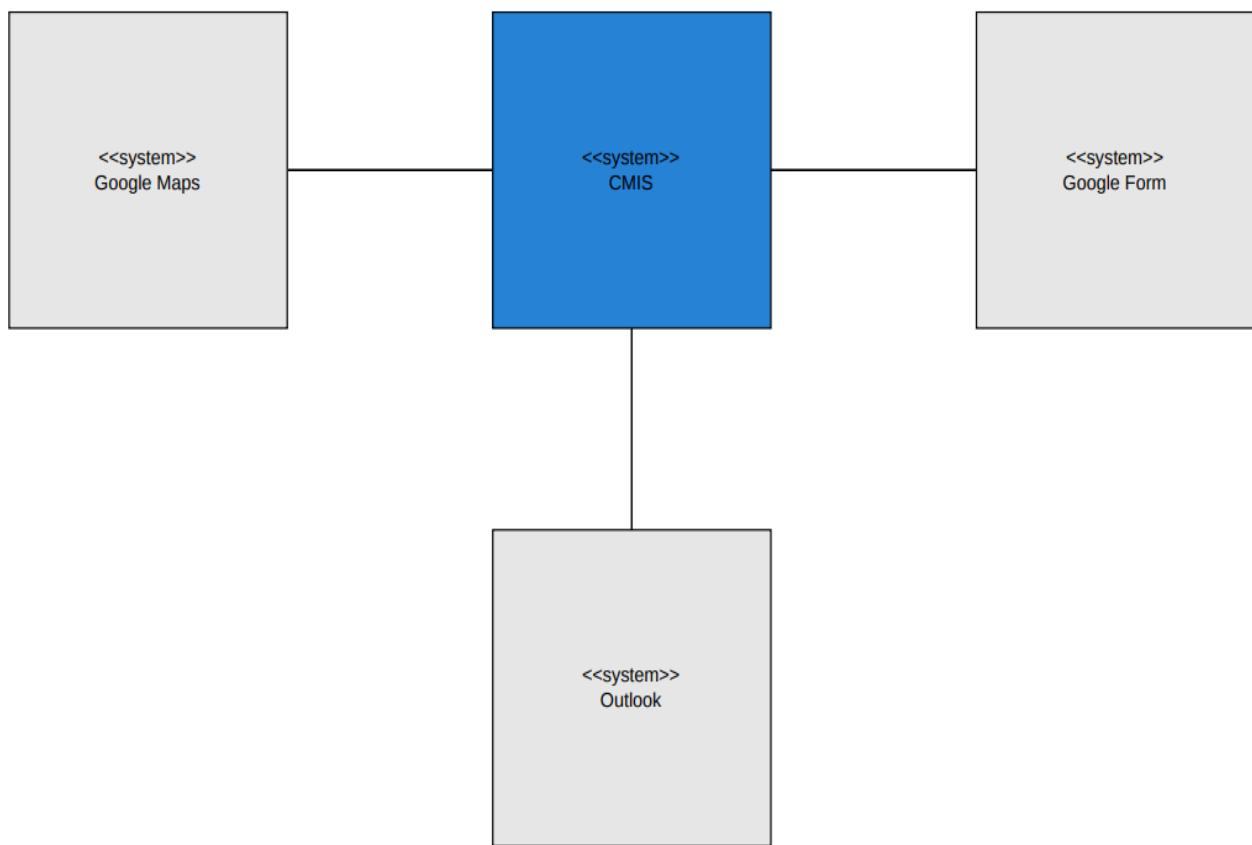
Qualified members can keep track of each member's contribution, participation in the community's activities. Event is planned by an authorized member, then members see the post about that event, and some of them click the button which means that member wants to attend the event. After the event is done, the qualified member can edit the attendance form for all members and see the overall participation of each member at any time. Every member has a profile in the management screen. Admin or authorized members open the management screen, and all members are seen on that screen. Authorized members, and admin can do the actions, which are permitted for that user, on that screen.

What can go wrong: Admin loses account, or becomes unavailable this can cause some management error, and it can be solved through contacting the admins of the applications.

Other Activities: While a member wants to perform an action, another member of the community is trying to perform the same action at the same time. While the admin is withdrawing a specific action from a member, that member might be performing that action during withdrawing.

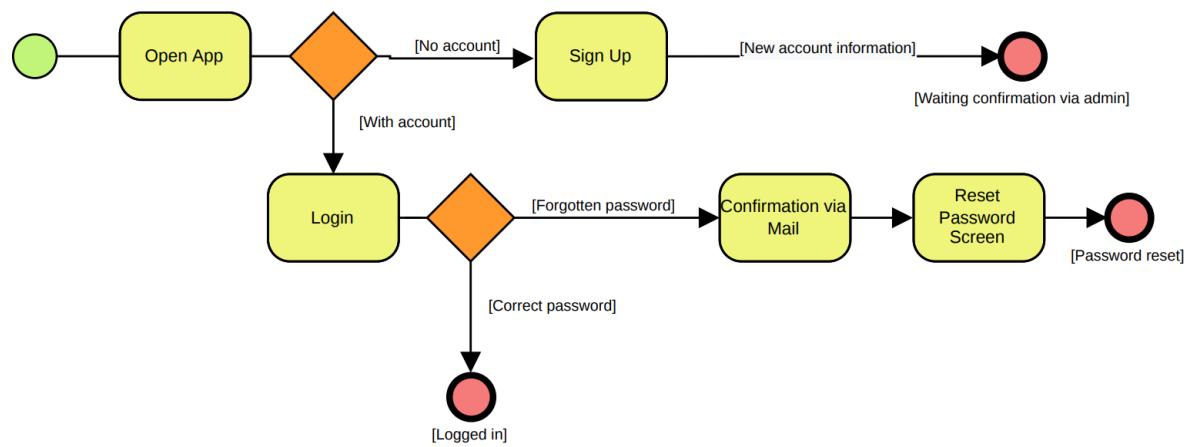
System state on completion: Each member's role is defined by the admin. Members/Qualified Members wanted to do an action, and the app performed this action, therefore the database is affected by this action. After an event is planned, some of the members attend, then an event is done, attendance is edited by an authorized member, then it is saved on the members participation data

Context Model

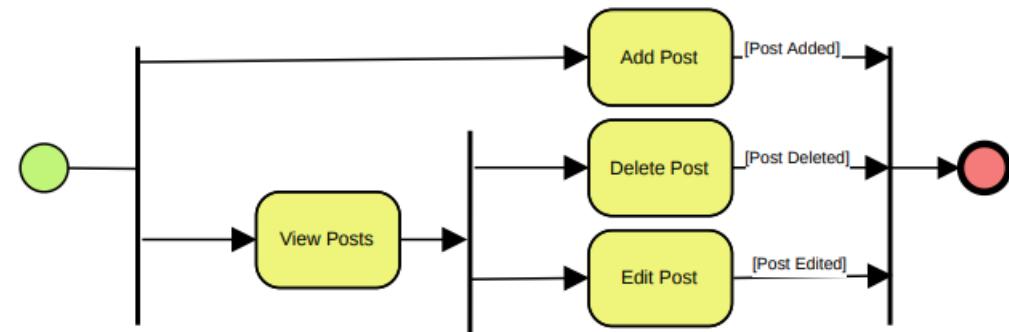


Process Models

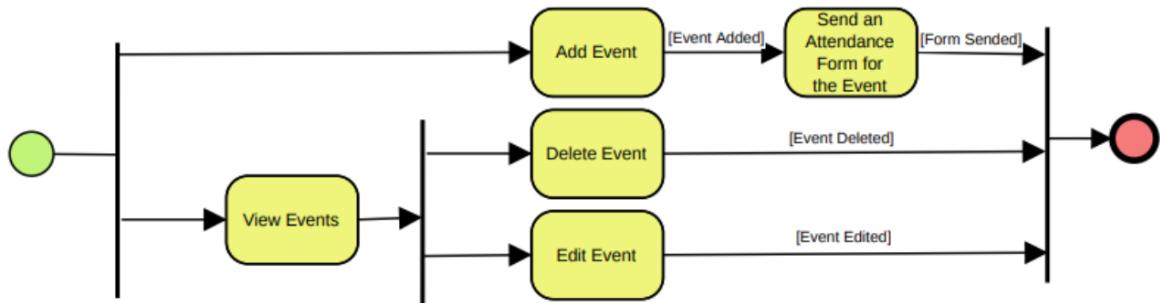
Community Login/Register



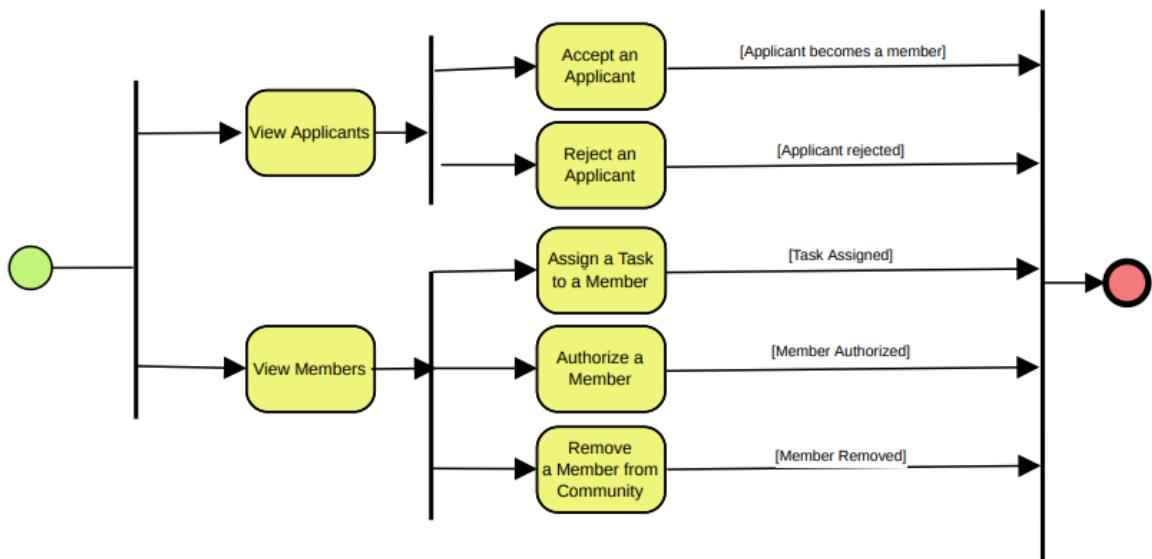
Community Manage Posts



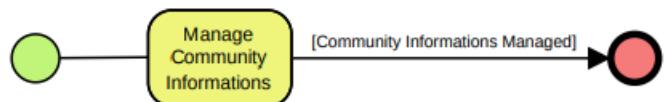
Community Manage Events



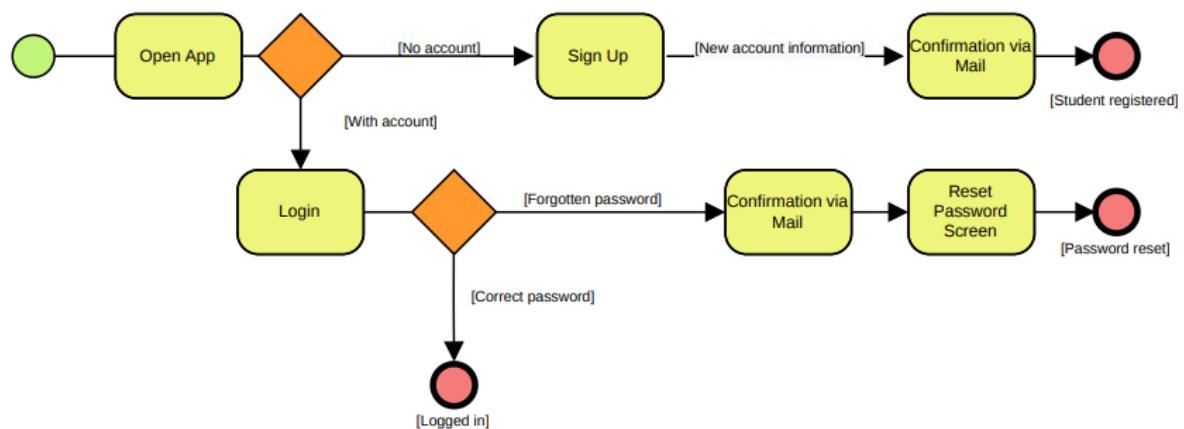
Community Member Management



Community Profile Management

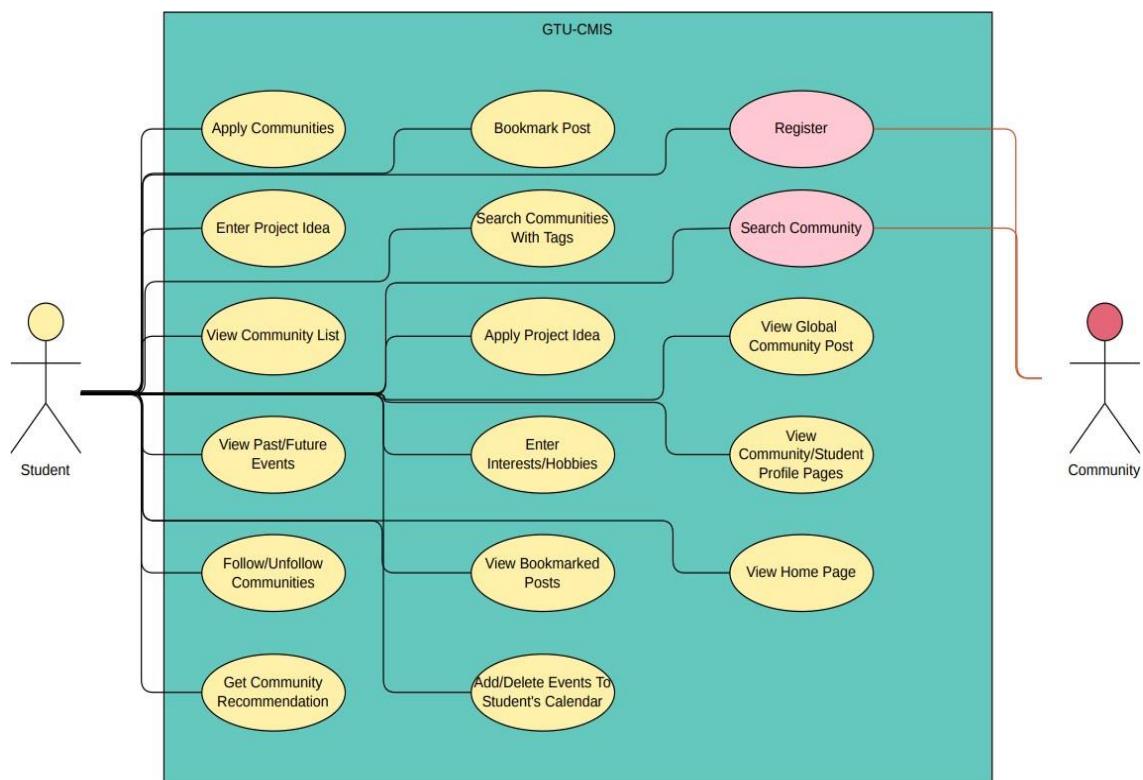


User Login/Register

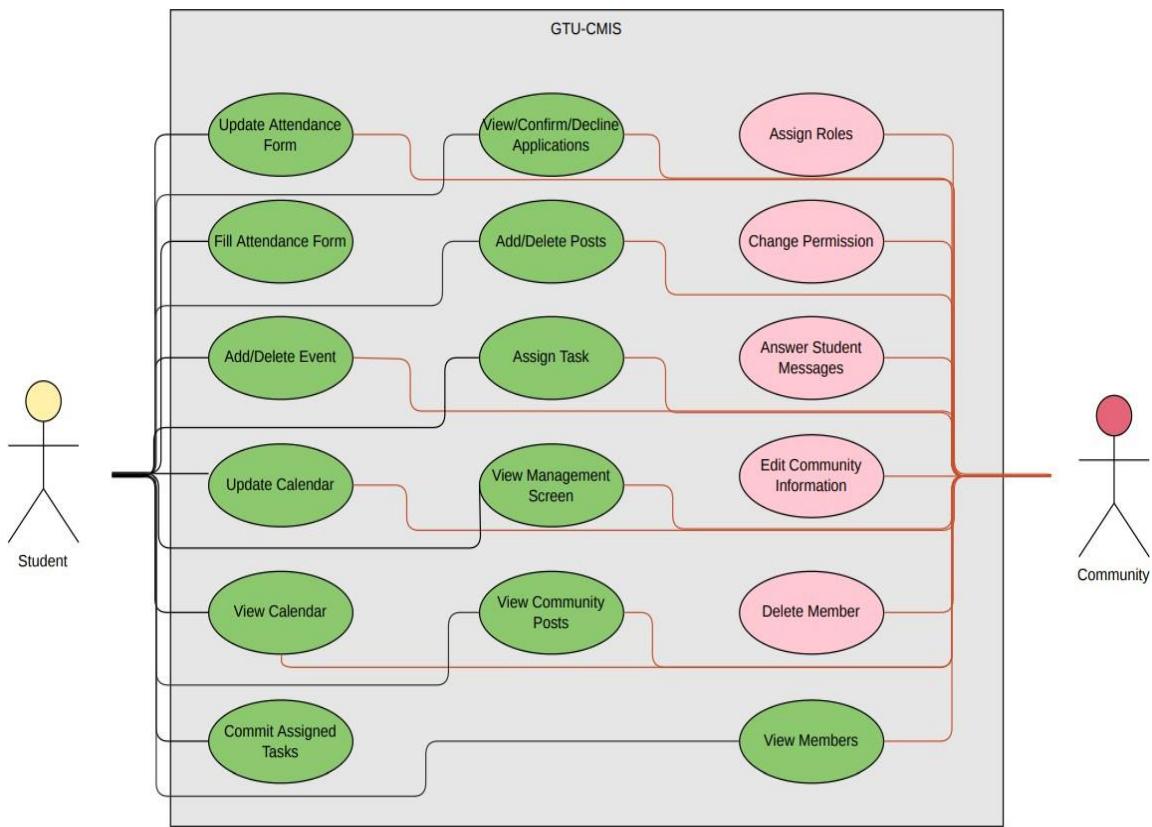


Use Case Diagrams

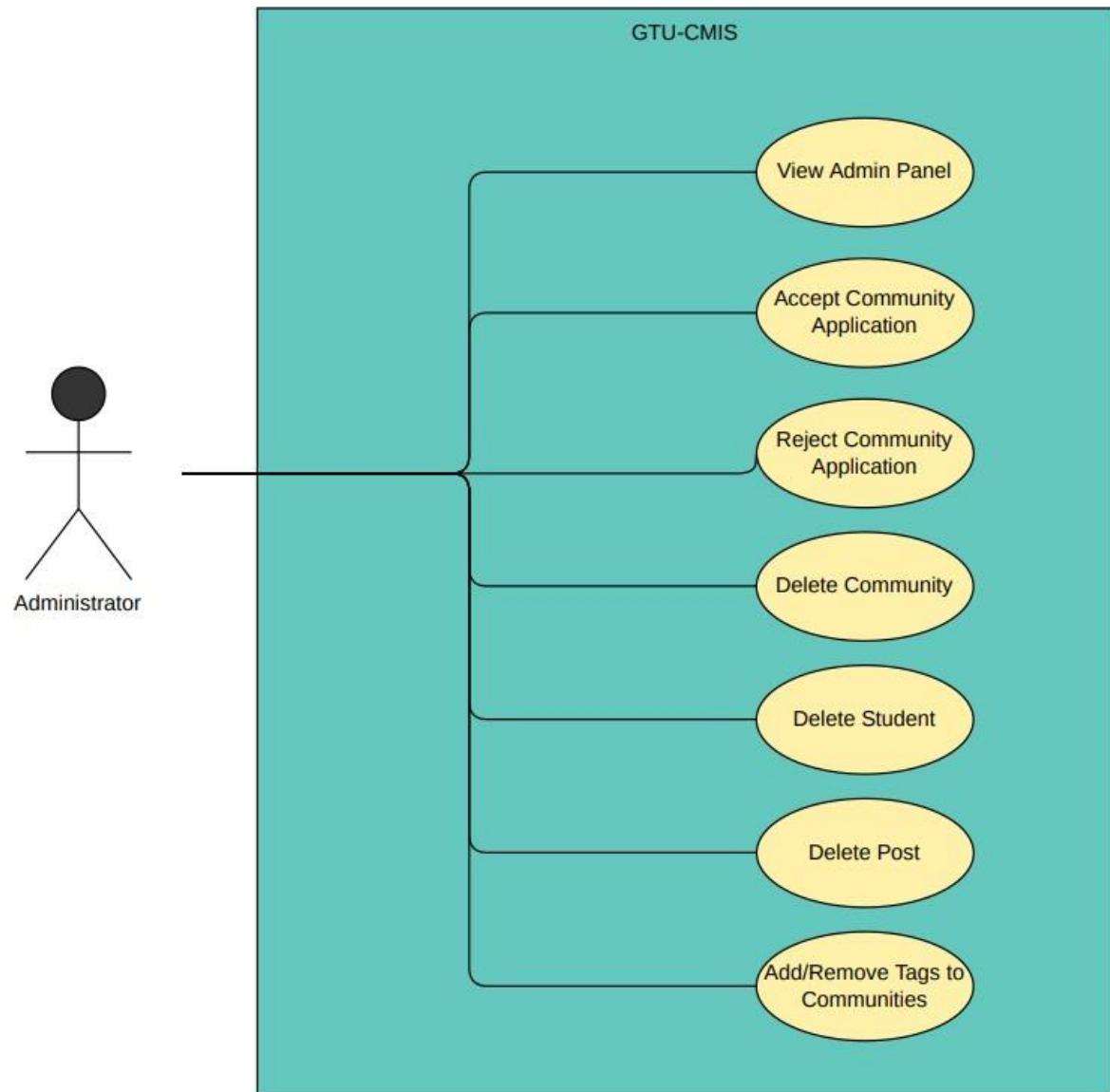
1- Student-Community



2- Student-Community



3- Administrator



Detailed Tables

System	CMIS
Use Case	Apply Communities
Actors	Student
Description	A Student may want to apply a community to become a member of that community. Application can be done with regular form, or community can provide GoogleForm for application. Student must wait till application is accepted. After becoming member, Student can benefit from community's events, announcement, management.
Data	Student's applied communities list, community's waiting application list. Community is added to the applied communities list, and Student is added to waiting application list of communities after this application.
Stimulus	Student clicks the apply button on community profile.
Response	Student's application is added to the communities waiting applications list.
Comments	Student must be logged in to the application on mobile or web, also Student must not be a member of that community.

System	CMIS
Use Case	Enter Project Idea
Actors	Student
Description	A Student may have a project idea, but not have a partner to perform that idea. That idea can be described on the Student profile.
Data	Project Idea consists of text and title.
Stimulus	Student clicks add project idea button in his/her profile.
Response	Project idea is added to the Student Profile, and it can be viewed by other Students when they enter to that Student's profile.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	View Community List
Actors	Student
Description	A Student may want to see all communities in the application.
Data	List of communities which includes name, and image information. Each segment leads to that community's page if it is clicked.
Stimulus	Student clicked to the view community button.
Response	Community list is popped up to the Student screen, nothing is changed on the system.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	View Past/Future Events
Actors	Student
Description	A Student has an option to see past event that contributed by Student, or future events to contribute.
Data	Event includes text, title, community name, date, and location of the event.
Stimulus	Student clicked to the view events button, events screen will come up.
Response	Event screen pops up, nothing is changed on the system.
Comments	Student must be logged in to the application on mobile or web and must be applied to the community which has at least one event, otherwise viewed list will be empty.

System	CMIS
Use Case	Follow/unfollow Communities
Actors	Student
Description	A Student might want to follow a community. Following a community is different from becoming a member of the community. Following does not require an application, every Student can follow every community.
Data	Student's followed communities list, and community's follower list. If Student follows a community both lists will be updated.
Stimulus	Student clicked to the follow/unfollow button on community profile.
Response	Community is added/deleted to the Student's followed communities list, Student is added/deleted to the community's follower list.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	Get Community Recommendation
Actors	Student
Description	A Student may want to get recommendation according to the taginformation which are provided by Student.
Data	Certain tags which are classified according to the community diversion. Student and community can have these tags.
Stimulus	Student clicked on the get community recommendation button on the main page.
Response	All communities, popped up to the screen as similar to the community list but only recommended ones. Nothing is changed on the system.
Comments	Student must be logged in to the application on mobile or web, and taginformation must be provided by Student.

System	CMIS
Use Case	Bookmark Post
Actors	Student
Description	A Student may want to save some posts, and these posts are specified as bookmarked posts. These posts can be viewed anytime by Student.
Data	Post consists of title and text. Student has bookmarked post list, if Student wants to bookmark a post, it will be added to that list.
Stimulus	Student clicked to bookmark post button.
Response	Post is added to the Student's bookmarked post list.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	Search Communities with Tags
Actors	Student
Description	There is a search option which can be done with tags.
Data	Certain tags which are classified according to the community diversion. Student and community can have these tags.
Stimulus	Student chooses tags which are used to list communities.
Response	Matched communities will be shown to the student. Nothing is changed on the system.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	Apply Project Idea
Actors	Student
Description	A Student can view another Student's project idea and apply to that idea in order to become a participator. System does not provide a further process other than notifying the project idea owner.
Data	Project Idea consists of text and title.
Stimulus	Student clicks apply button on another Student's project idea screen.
Response	Project idea owner gets a notification, for that application.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	Enter Interest/Hobbies
Actors	Student
Description	A Student can enter interests, and hobbies for Student profile. Interests are tags, hobbies are in text format. Tags will be used for community recommendation.
Data	Hobbies are simple text, tags are predefined, and chosen by Student. Both information will be added to the hobbies/interest section on Student profile.
Stimulus	Student clicked to the edit button his/her profile page.
Response	Student's interest/hobbies section is updated.
Comments	Student must be logged in to the application on mobile or web.

System	CMIS
Use Case	View Bookmarked Posts
Actors	Student
Description	Student can view the bookmarked posts.
Data	Post consists of title and text. Student has bookmarked post list, if Student wants to bookmark a post, it will be added to that list.
Stimulus	Student clicks the bookmarked posts button on the main page.
Response	Bookmarked posts listed to the Studentscreen. Nothing is changed on the system.
Comments	Student must be logged in to the application on mobile or web. Student must have at least one bookmarked post to see the list.

System	CMIS
Use Case	Add/Delete Events to Student's Calendar
Actors	Student
Description	Student can add/delete event from personal calendar.
Data	Calendar consists of dates, and each event's title will be seen on that calendar's matched date. Event includestext, title, community name, date, and location of the event.
Stimulus	Student clicked event title on the calendar and delete or add option popped up to the screen.
Response	Event is added or deleted from Student.
Comments	Student must be logged in to the application.

System	CMIS
Use Case	Register
Actors	Student, Community
Description	It can be registered to the application as Student or community. Community must be approved by administrator.
Data	Student's first name, last name, email, password, image, description. Community's name, email, image, password, and description.
Stimulus	Sign in, button is clicked.
Response	Student is signed into the application, if community tried to sign in, request is sent to the administrator.
Comments	Student mails must have @gtu.edu.tr part.

System	CMIS
Use Case	Search Community
Actors	Student, Community
Description	Communities can be searched from search bar.
Data	Community name.
Stimulus	Student writes the community's name to the search bar, then clicks enter.
Response	Community that matched with the name will be shown to the Student. Nothing is changed on the system. If community does not exist, Student sees the error screen.
Comments	Student must be logged in to the application.

System	CMIS
Use Case	View Global Community Posts
Actors	Student
Description	Community's global posts can be seen by Student.
Data	Post consists of title and text.
Stimulus	Student clicked another community's profile, and posts are listed.
Response	Posts are listed to the screen. Nothing is changed on the system.
Comments	Student must be logged in to the application.

System	CMIS
Use Case	View Community/Student Profile Pages
Actors	Student, Community
Description	Profile page of community, and Student can be viewed by other communities and Students.
Data	Profile page includes each entities information such as name, image. If it is community, there is posts, follow button,apply button. These profiles are viewed differently, if it is viewed by Student, or community.
Stimulus	Student/Community entered the profile.
Response	Profile page is viewed by Student/community. Nothing is changed on the system.
Comments	Student must be logged in to the application.

System	CMIS
Use Case	View Home Page
Actors	Student
Description	Home page can be viewed by Student and community. Student will see communities which are followed or participated, posts. Community will see community contents.
Data	Posts which consist of text, title, and image.
Stimulus	Main page button is clicked.
Response	Main page information is shown to the Student/community.
Comments	Student must be logged in to the application.

System	CMIS
Use Case	Update Attendance Form
Actors	Student, Community
Description	Attendance form can be updated by authorized Student, or community.
Data	Form consists of Students which are expected to attend the event. AuthorizedStudent or community can delete some of them if expected Student is not attended.
Stimulus	Student/Community clicks the update attendance button on the event.
Response	Student's attendance state is updated.
Comments	Student must be logged in to the application. Student must be authorized member of the community.

System	CMIS
Use Case	Add/Delete Event
Actors	Student, Community
Description	Authorized members, and community can add/delete events. If it is Student's personal calendar it is not required to be authorized.
Data	Event includes text, title, community name, date, and location of the event.
Stimulus	Student/Community clicked to the segment of calendar.
Response	Event is added to the calendar, event is deleted from calendar.
Comments	Student must be logged in to the application. If calendar is community's calendar, Student must be authorized member of the community.

System	CMIS
Use Case	Update Calendar
Actors	Student, Community
Description	Authorized members, and community can update calendars.
Data	Calendar includes dates, and events. Event includes text, title, community name, date, and location of the event.
Stimulus	Student/Community clicked to the segment of calendar.
Response	Event on the calendar is updated.
Comments	Student must be logged in to the application. If calendar is community's calendar, Student must be authorized member of the community. There must be an existing event to update on calendar.

System	CMIS
Use Case	View Calendar
Actors	Student, Community
Description	Student/Community can view the calendar. It can be community calendar, or personal calendar. Personal calendars can only be viewed by owner of that calendar.
Data	Calendar includes dates, and events. Event includes text, title, community name, date, and location of the event.
Stimulus	Student/Community clicked the calendar symbol.
Response	Calendar is shown to the Student/community. Nothing is changed on the system.
Comments	Student must be logged in to the application. If calendar is community's calendar, Student must be authorized member of the community.

System	CMIS
Use Case	View Members
Actors	Student, Community
Description	Authorized members, and community can view the members of the community, on the management screen.
Data	List of members. First name, last name, email information is provided.
Stimulus	Student/Community opened the management screen.
Response	List of members are on the screen.
Comments	Student must be logged in to the application. Student must be authorized member of the community.

System	CMIS
Use Case	Commit Assigned Tasks
Actors	Student
Description	Student can change the task state to done.
Data	Task consists of title, text, and state.
Stimulus	Student drags the task to the done segment.
Response	Task's state is changed to done.
Comments	Student must be logged in to the application. Student must be member of the community, and task must be assigned to that Student.

System	CMIS
Use Case	View/Confirm/Decline Applications
Actors	Student, Community
Description	Authorized members of the community, and community can manage the applications. It can be viewed, confirmed, and declined.
Data	Applicant's which consists of Students. If decline, and confirm operations are performed, list will be decreased.
Stimulus	Student/Community clicked the applications button on management screen.
Response	Applications are viewed, confirmed, or declined. If it is only viewed, nothing is changed on the system. If confirmed, new member is added to the community.
Comments	Student must be logged in to the application. Student must be authorized member of the community.

System	CMIS
Use Case	Add/Delete Post
Actors	Student, Community
Description	Authorized Students, and community can delete/add post to the community.
Data	Post consists of title and text.
Stimulus	Student/Community clicked to the post and clicked to delete option. Student/Community typed text, title, and put image(optional) and clicked share button.
Response	Post is deleted from community. Post is shared on the community page.
Comments	Student must be logged in to the application. Student must be authorized member of the community.

System	CMIS
Use Case	Assign Task
Actors	Student, Community
Description	Authorized Members, and Community can assign tasks to the other members.
Data	Task consists of title, text, and state.
Stimulus	Student/Community clicked to the Student, then assign task button.
Response	Task is assigned to that Student.
Comments	Student must be logged in to the application. Student must be authorized member of the community. Also, task should not be assigned already.

System	CMIS
Use Case	View Management Screen
Actors	Student, Community
Description	Authorized Students, and community can see the management screen. Which consists of member list, and operations for each of the member on that list, delete, assign task, authorize, confirm/reject applications.
Data	Management screen consists of member list, applications, and operations.
Stimulus	Student/Community clicked to the management section.
Response	Management screen is shown to the Student/community.
Comments	Student must be logged in to the application. Student must be authorized member of the community.

System	CMIS
Use Case	View Community Posts
Actors	Student, Community
Description	Community's posts can be viewed by members, and community.
Data	Post consists of title and text.
Stimulus	Community profile is opened.
Response	Posts are shown to the Student/community. Nothing is changed on the system.
Comments	Student must be logged in to the application. Student must be member of the community.

System	CMIS
Use Case	Fill Attendance From
Actors	Student
Description	When event is scheduled, member can apply to the event, and member is automatically added to the attendance form.
Data	Event includes text, title, community name, date, and location of the event. Form consists of Students which are expected to attend the event.
Stimulus	Member clicks the event and apply option.
Response	Member is added to the event's applicant list.
Comments	Student must be logged in to the application. Student must be member of the community.

System	CMIS
Use Case	Assign Roles
Actors	Community
Description	Some of the members can be made authorized member through assigning roles by community account on the management screen.
Data	Members are reached through memberlist. State of the member is changed to authorized.
Stimulus	Community clicks the member's name and chooses make authorized option.
Response	Member becomes authorized.
Comments	That member is Student, and Student must be member of the community.

System	CMIS
Use Case	Change Permission
Actors	Community
Description	Community can make authorized members to regular member.
Data	Members are reached through memberlist. State of the member is changed to authorized.
Stimulus	Community clicks the member's name and chooses make unauthorized option.
Response	Member becomes unauthorized.
Comments	That member is Student, and Student must be member of the community.

System	CMIS
Use Case	Answer Student Messages
Actors	Community
Description	Community can answer the incoming messages.
Data	Message consists of only texts. Sender, and receiver.
Stimulus	Community clicked the messages and clicked the send button after typing.
Response	Message is sent to the Student.
Comments	Student is not required to be member of the community.

System	CMIS
Use Case	Edit Community Information
Actors	Community
Description	Community can edit its description, image, and tags.
Data	Editable information are image, tags, and description.
Stimulus	Community clicked the edit button.
Response	Community's information is changed.
Comments	Authorized members cannot do these change.

System	CMIS
Use Case	Delete Member
Actors	Community
Description	Some of the members are wanted to be removed from community because of various reasons which are unimportant for the system.
Data	Member information first name, last name, and member list of community.
Stimulus	Community clicked the delete button after clicking member on the management screen.
Response	Member is deleted from community.
Comments	Member must be participant of the community before deleting.

System	CMIS
Use Case	View Admin Panel
Actors	Administrator
Description	Administrator of the app can view the admin panel.
Data	List of Students, communities, and posts.
Stimulus	Administrator clicks the admin panel button.
Response	Admin panel is opened. Nothing is changed on the system.
Comments	Administrator must be logged in.

System	CMIS
Use Case	Accept Community Application
Actors	Administrator
Description	When community's sign into the application, it is not directly accepted. To community to be part of the application, it must be confirmed by administrator.
Data	Applicant list of communities.
Stimulus	Administrator clicks to the applicants list on the admin panel, then confirm is clicked.
Response	Community is successfully signed in.
Comments	Administrator must be logged in.

System	CMIS
Use Case	Reject Community Application
Actors	Administrator
Description	When community's sign into the application, it is not directly accepted. To community to be part of the application, it must be confirmed by administrator.
Data	Applicant list of communities.
Stimulus	Administrator clicks to the applicants list on the admin panel, then reject is clicked.
Response	Community is successfully rejected.
Comments	Administrator must be logged in.

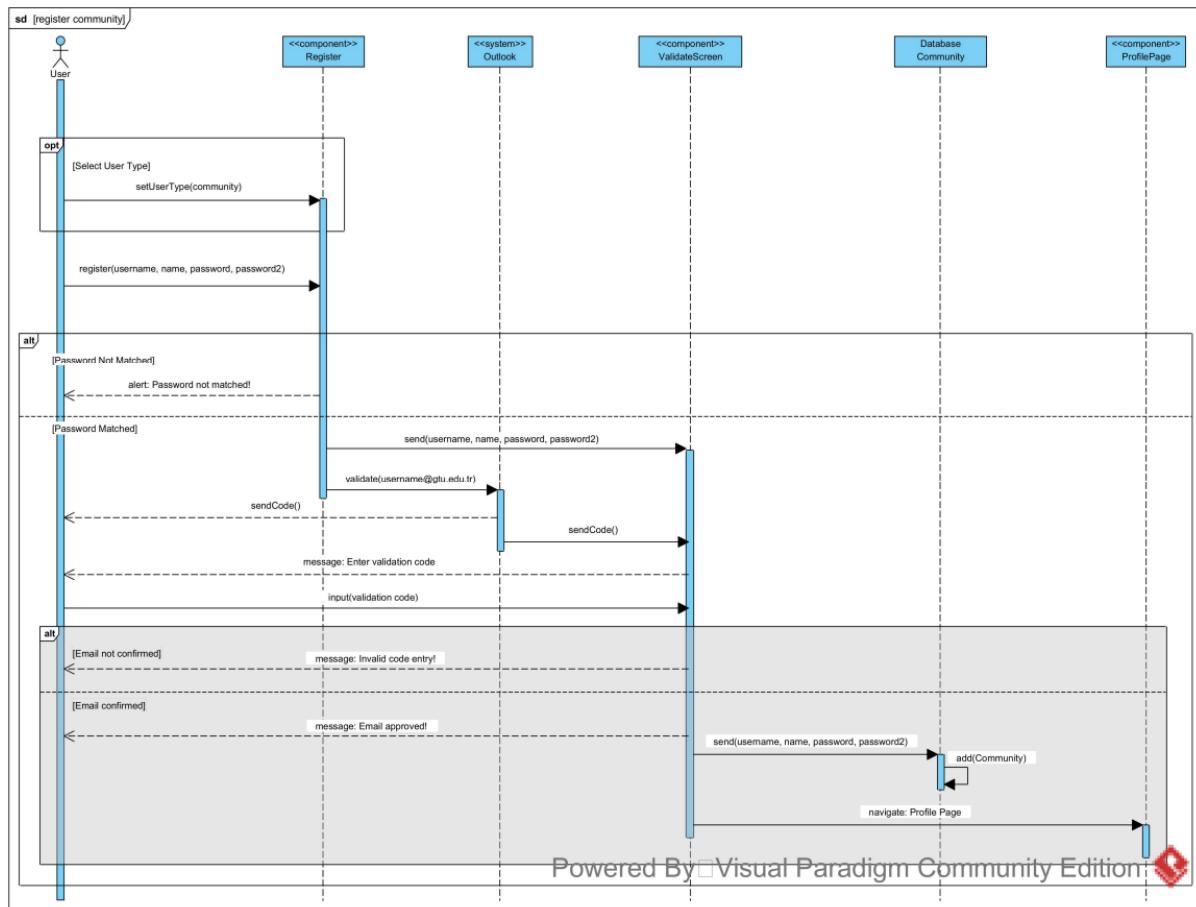
System	CMIS
Use Case	Delete Community
Actors	Administrator
Description	Administrator can delete any community from system.
Data	List of all communities.
Stimulus	Administrator clicked the communities tab, then clicks the delete option for one community.
Response	Community is deleted from system.
Comments	Administrator must be logged in.

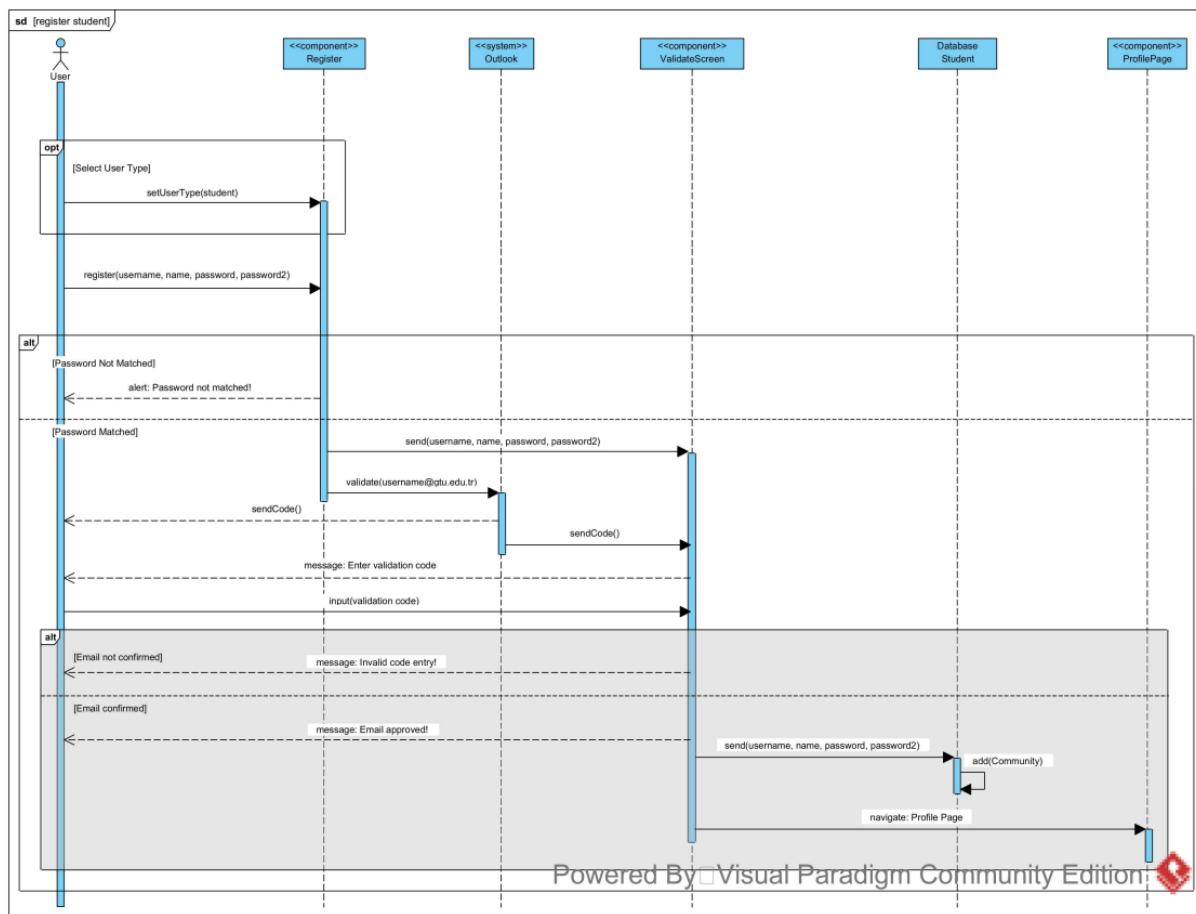
System	CMIS
Use Case	Delete Student
Actors	Administrator
Description	Administrator can delete any Student from system.
Data	List of all Students.
Stimulus	Administrator clicked the communities tab, then clicks the delete option for one Student.
Response	Student is deleted from system.
Comments	Administrator must be logged in.

System	CMIS
Use Case	Delete Post
Actors	Administrator
Description	Administrator can delete any post from system.
Data	List of all posts.
Stimulus	Administrator clicked the posts tab, then clicks the delete option for one post.
Response	Post is deleted from system.
Comments	Administrator must be logged in.

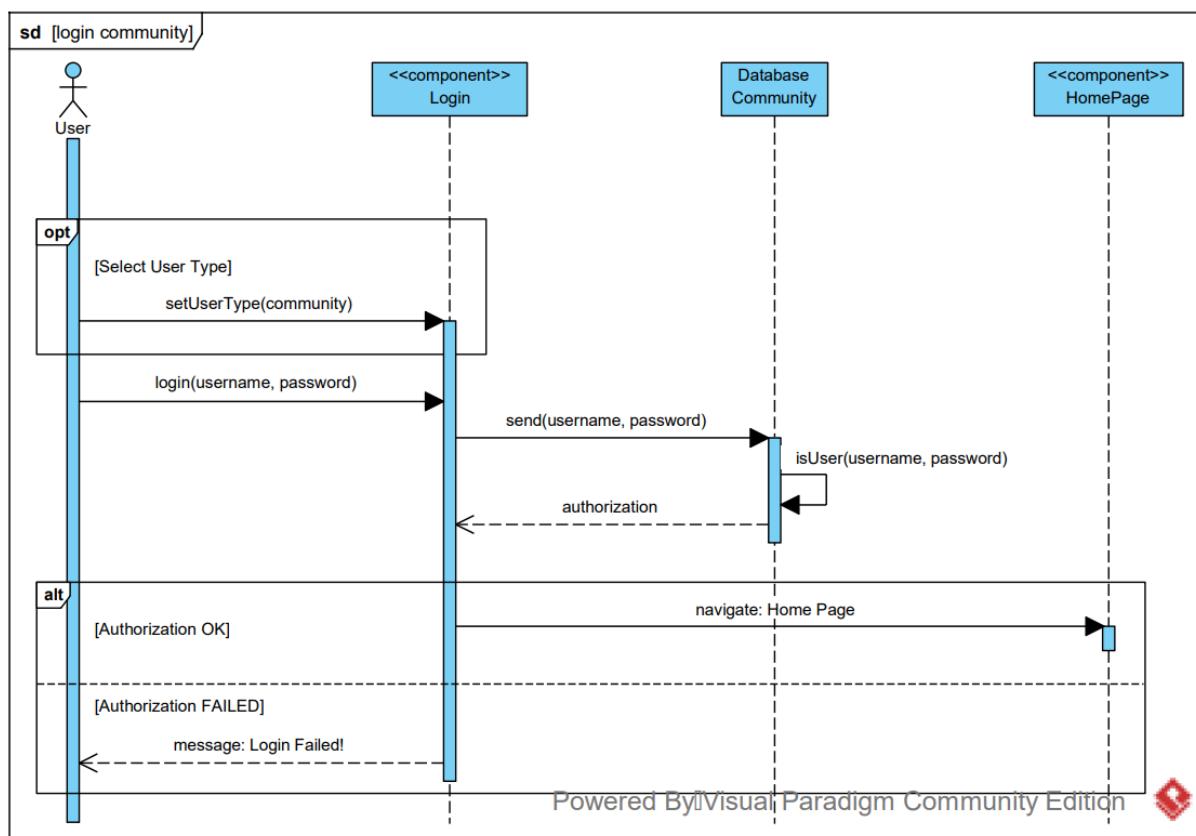
System	CMIS
Use Case	Add/Remove Tags to/from Community
Actors	Administrator
Description	Administrator can update the tags of any community. It is needed, because community can misuse the application, and it can be prevented by administrator.
Data	List of all communities, and their tags.
Stimulus	Administrator clicked to the community, and tags button, then added or removed tag.
Response	Tag is added or deleted from community.
Comments	Administrator must be logged in.

Sequence Diagrams

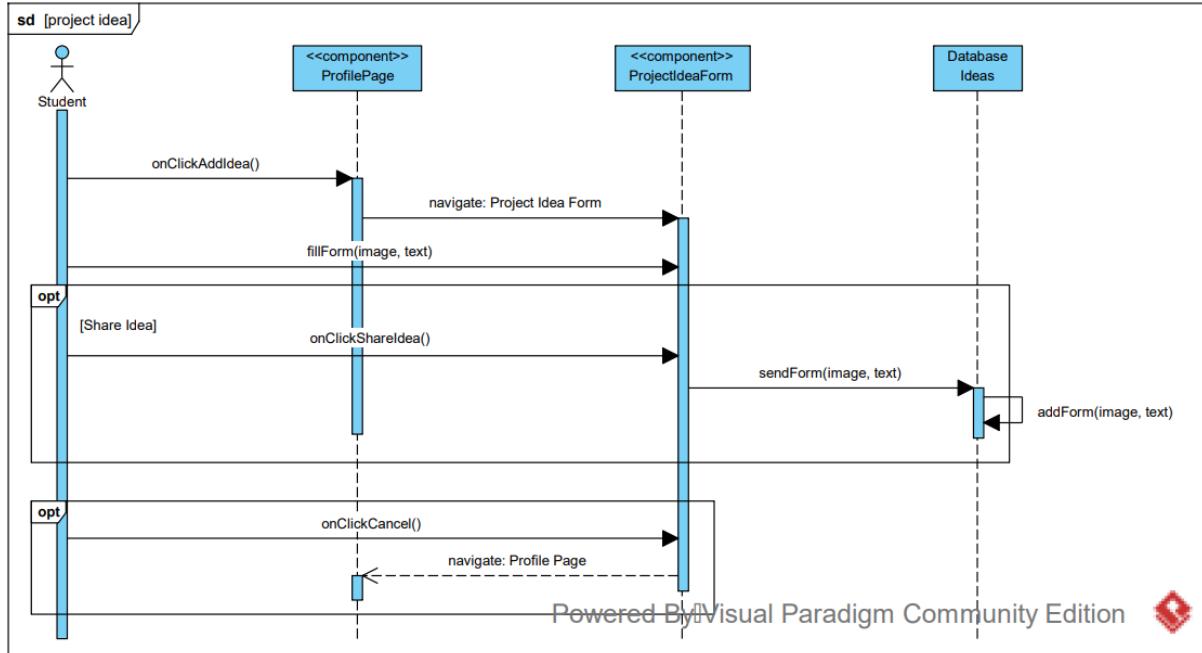
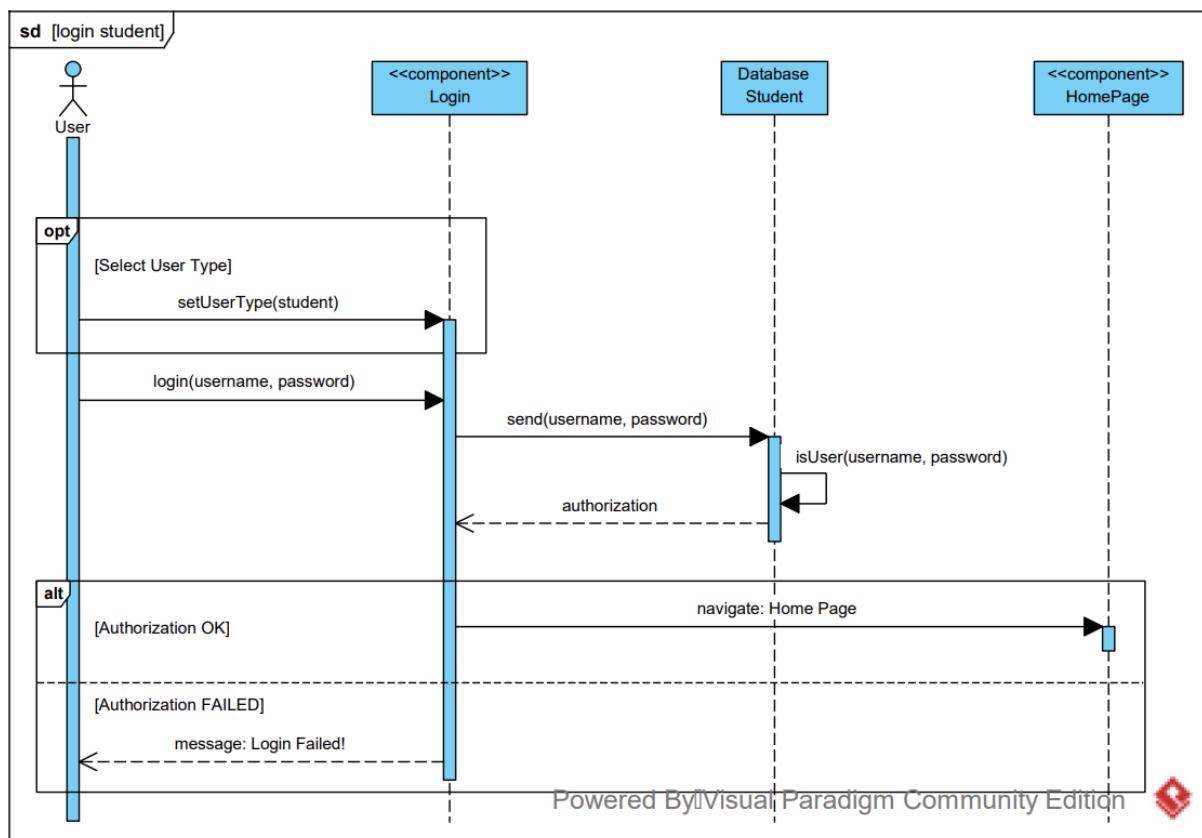


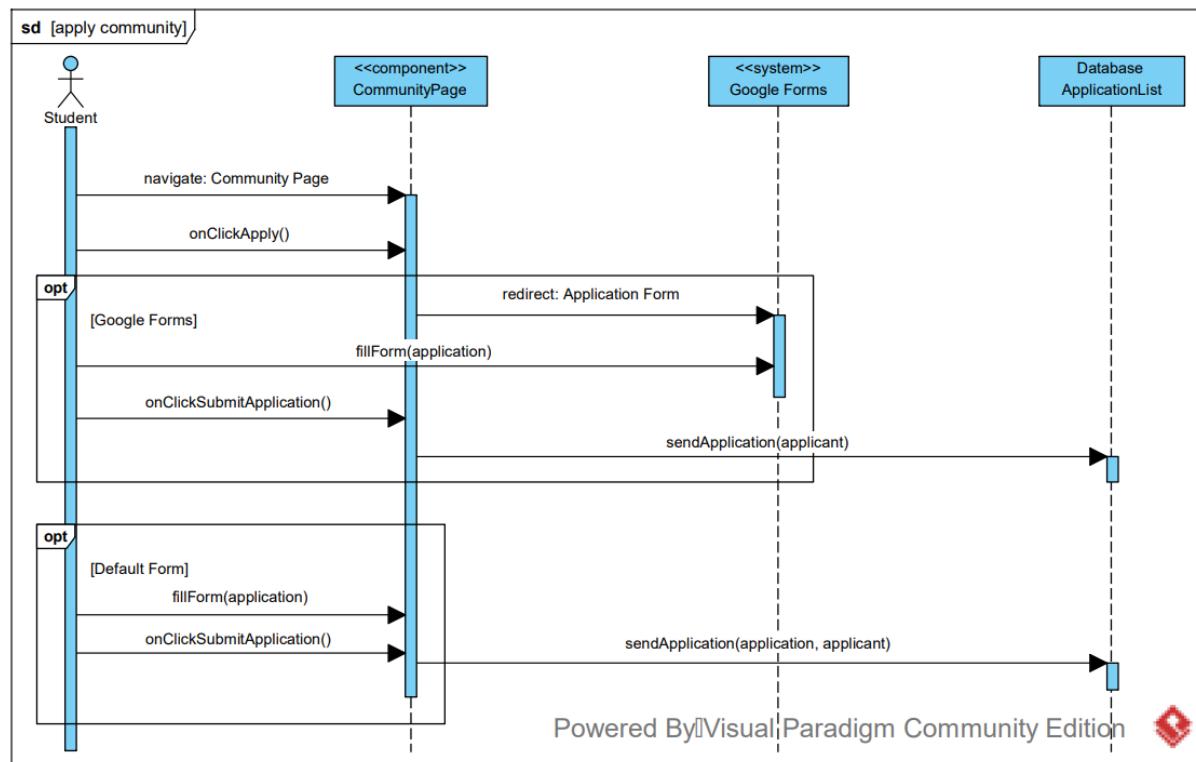
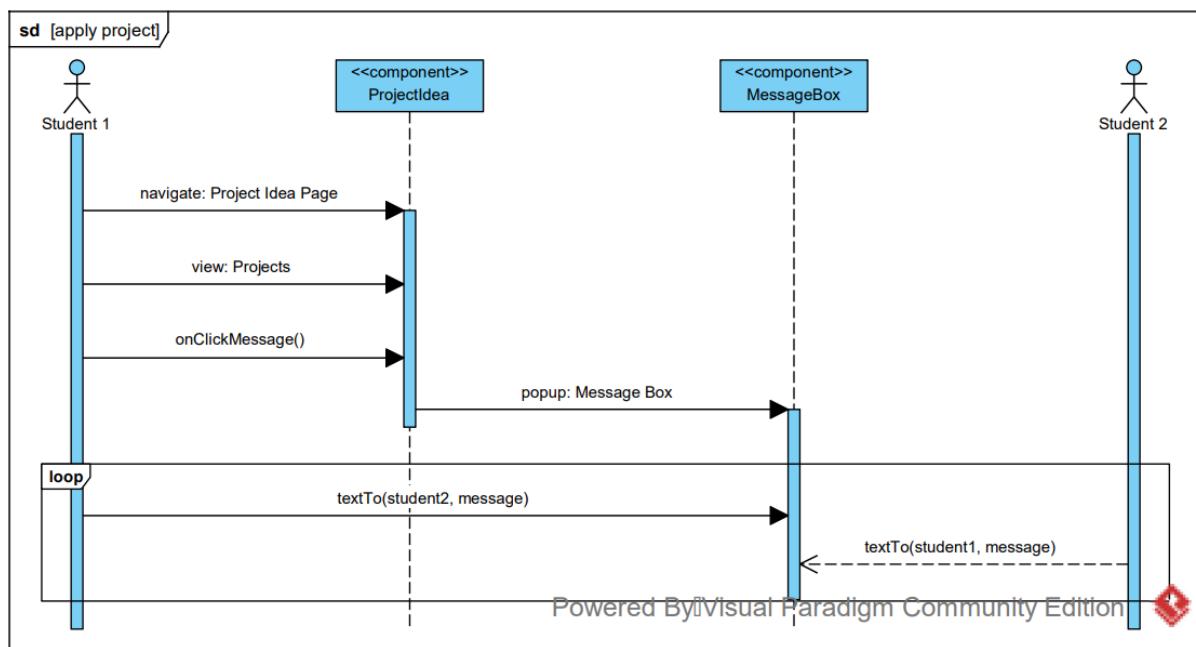


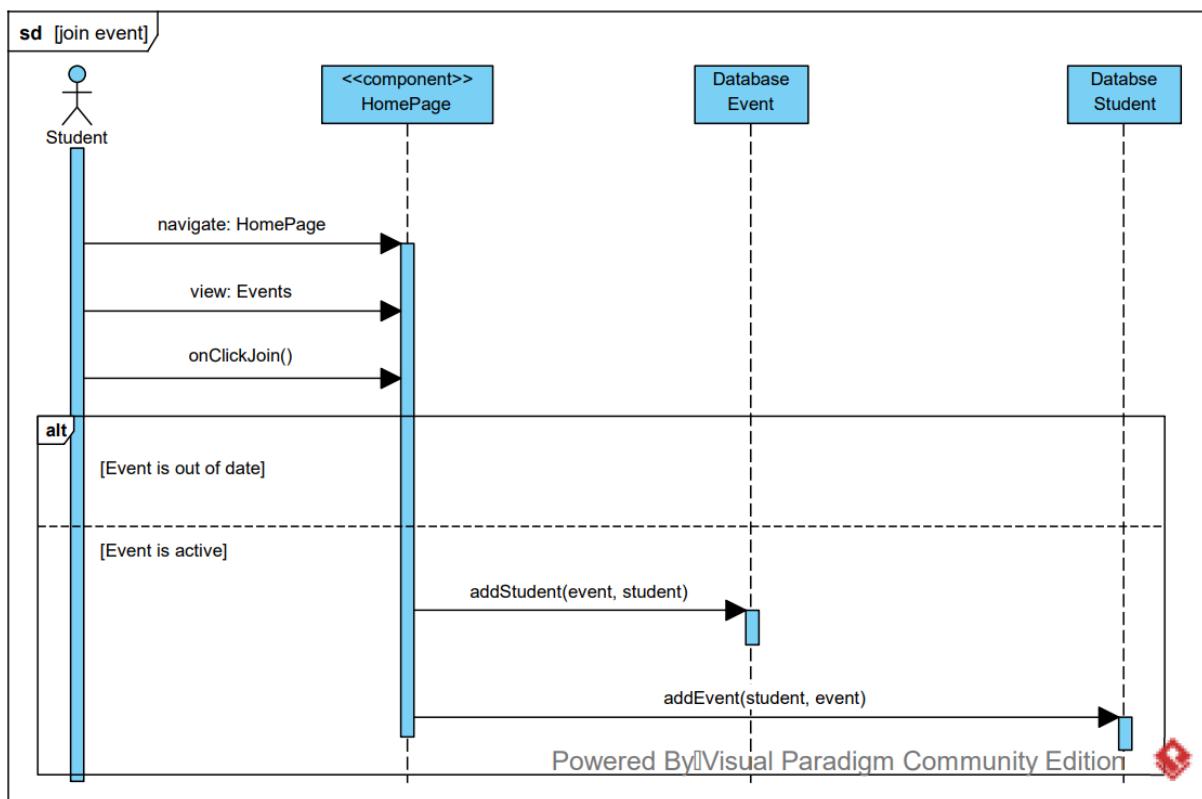
Powered By Visual Paradigm Community Edition

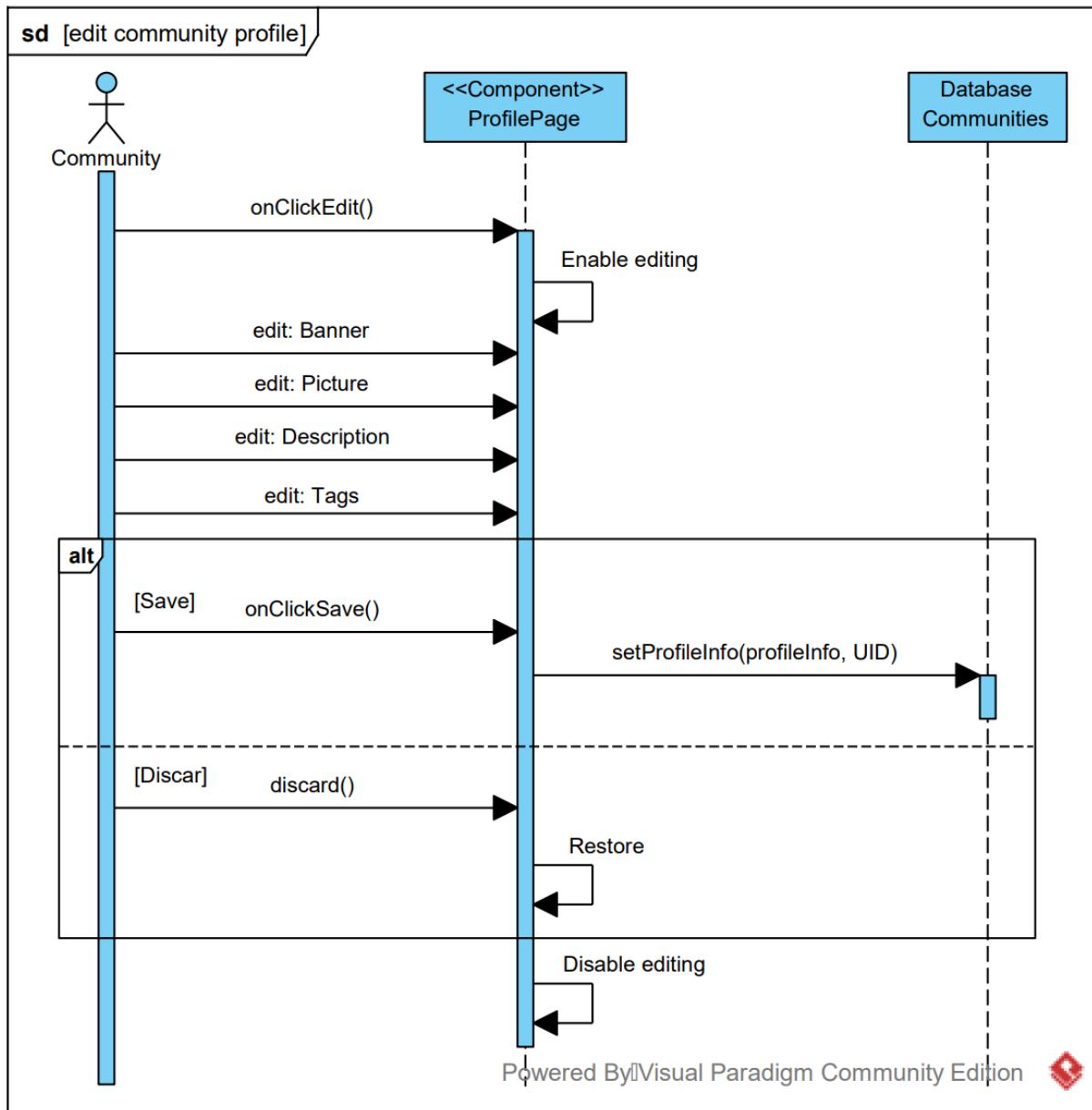


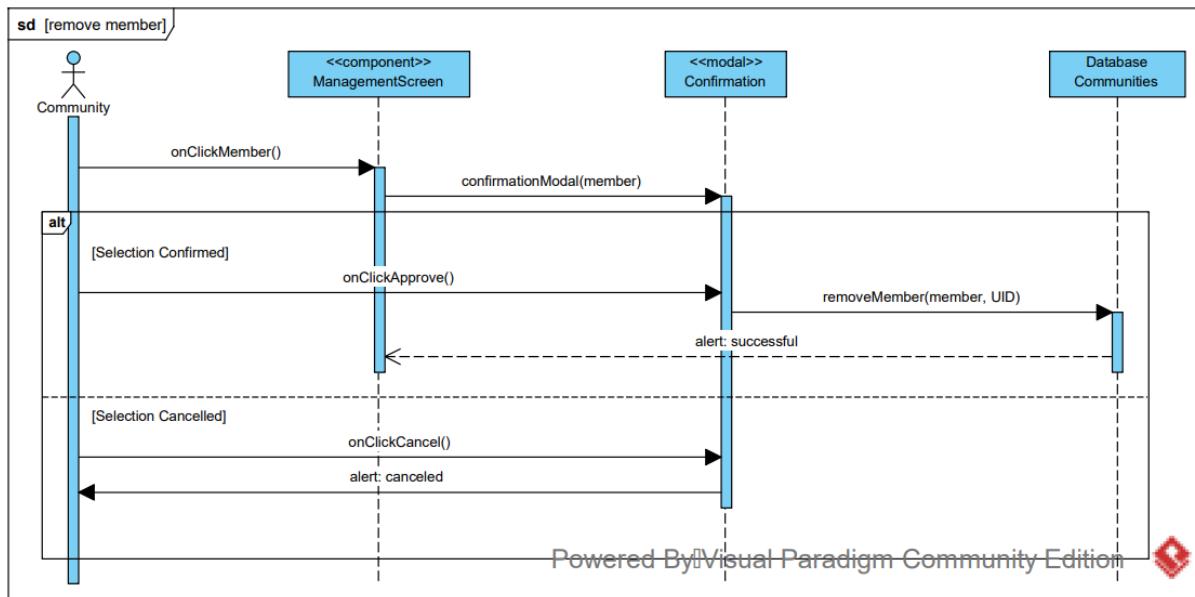
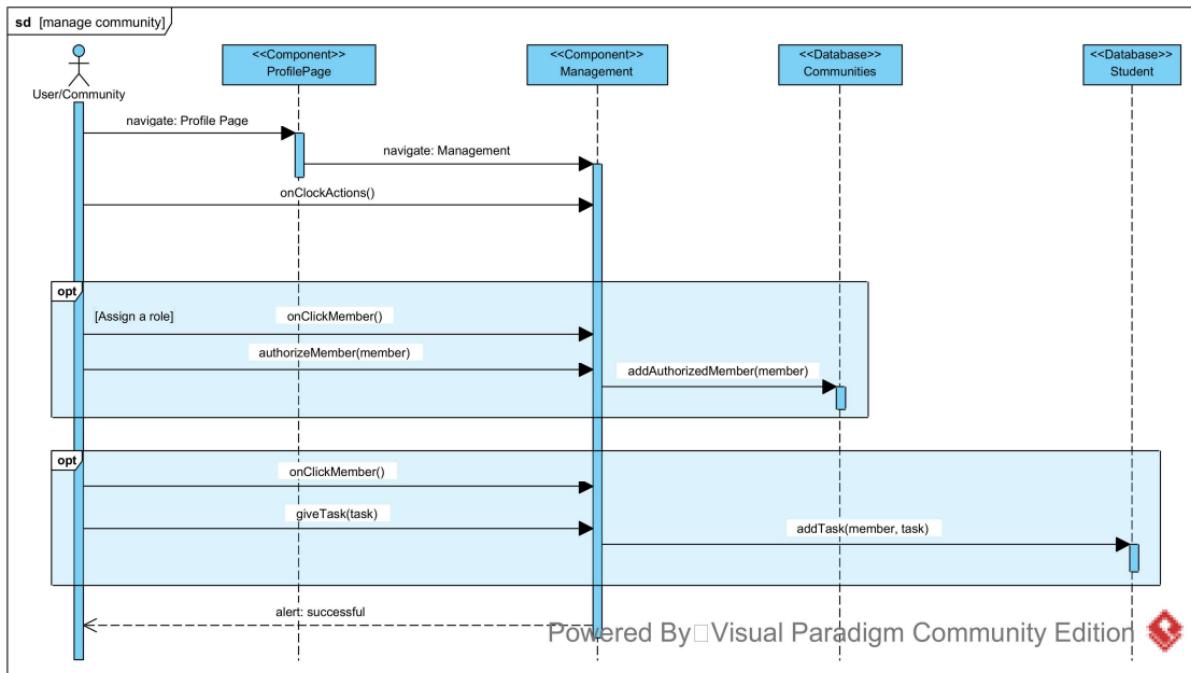
Powered By Visual Paradigm Community Edition

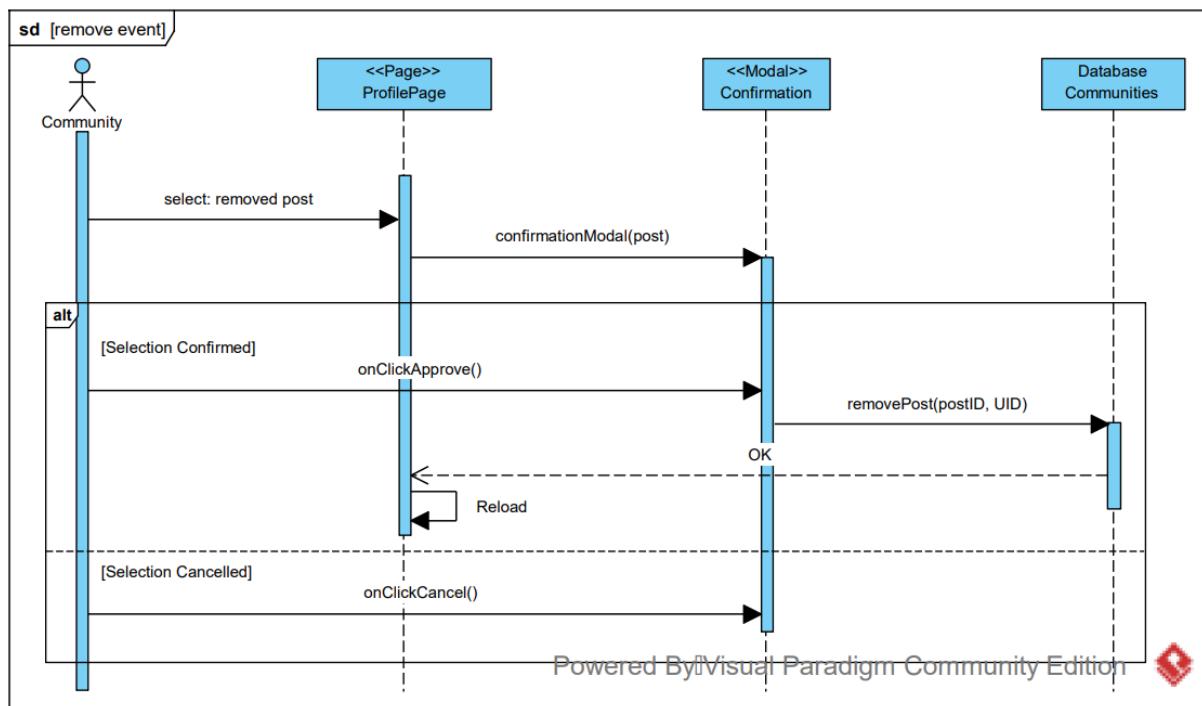
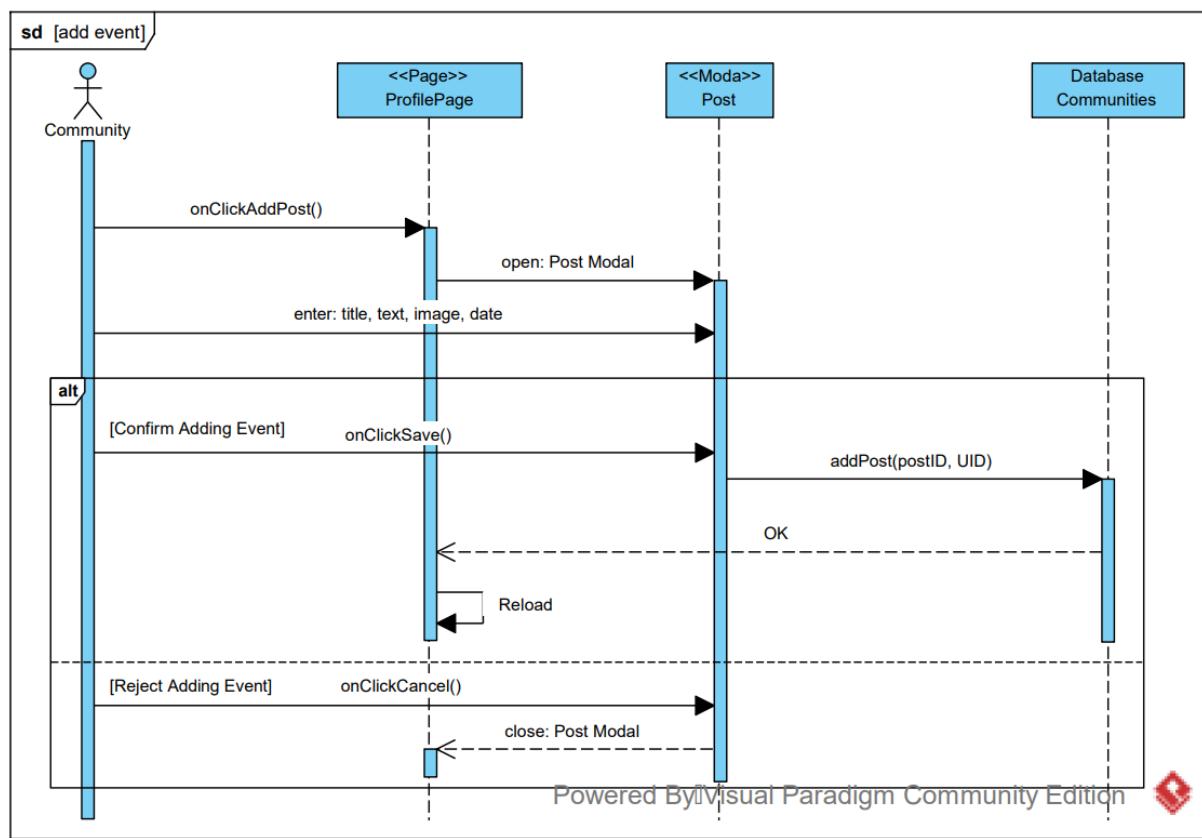


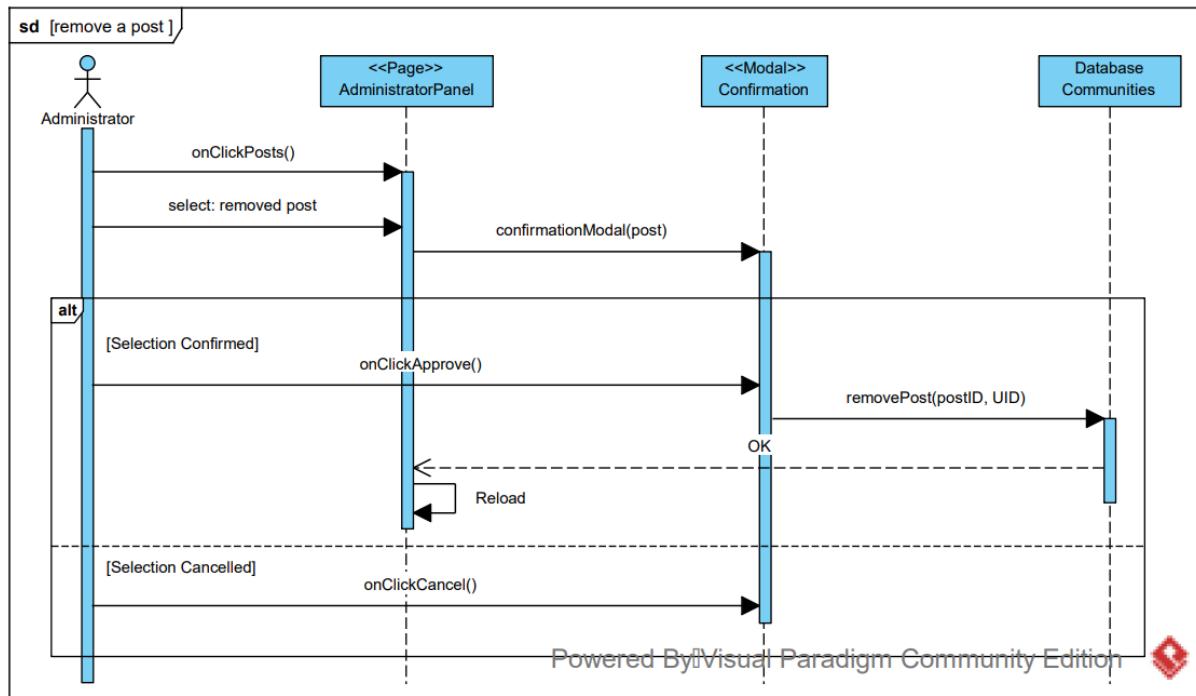
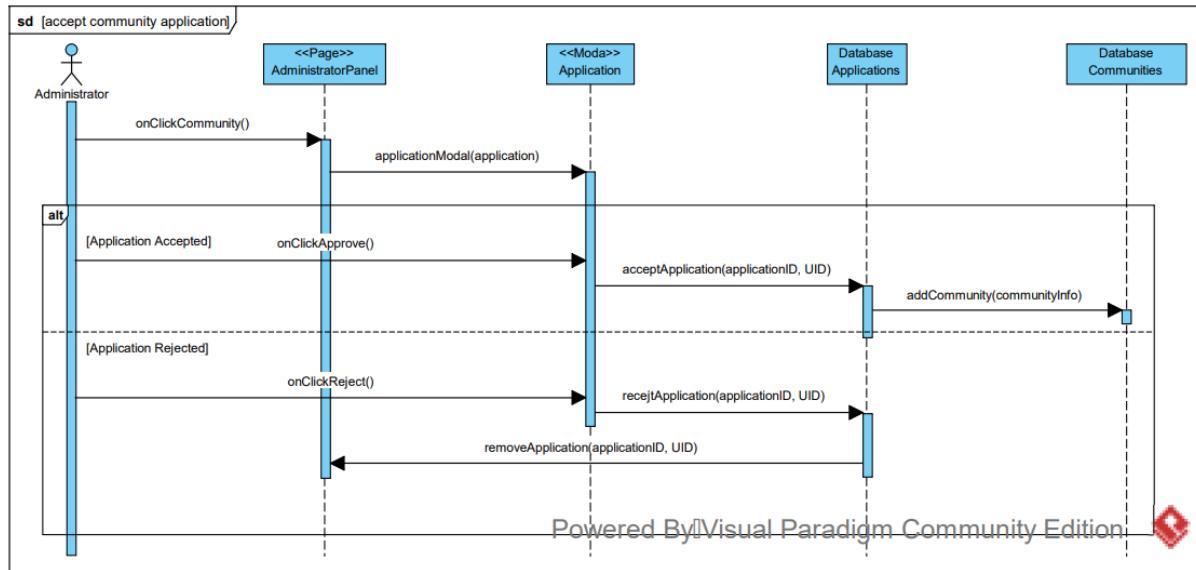




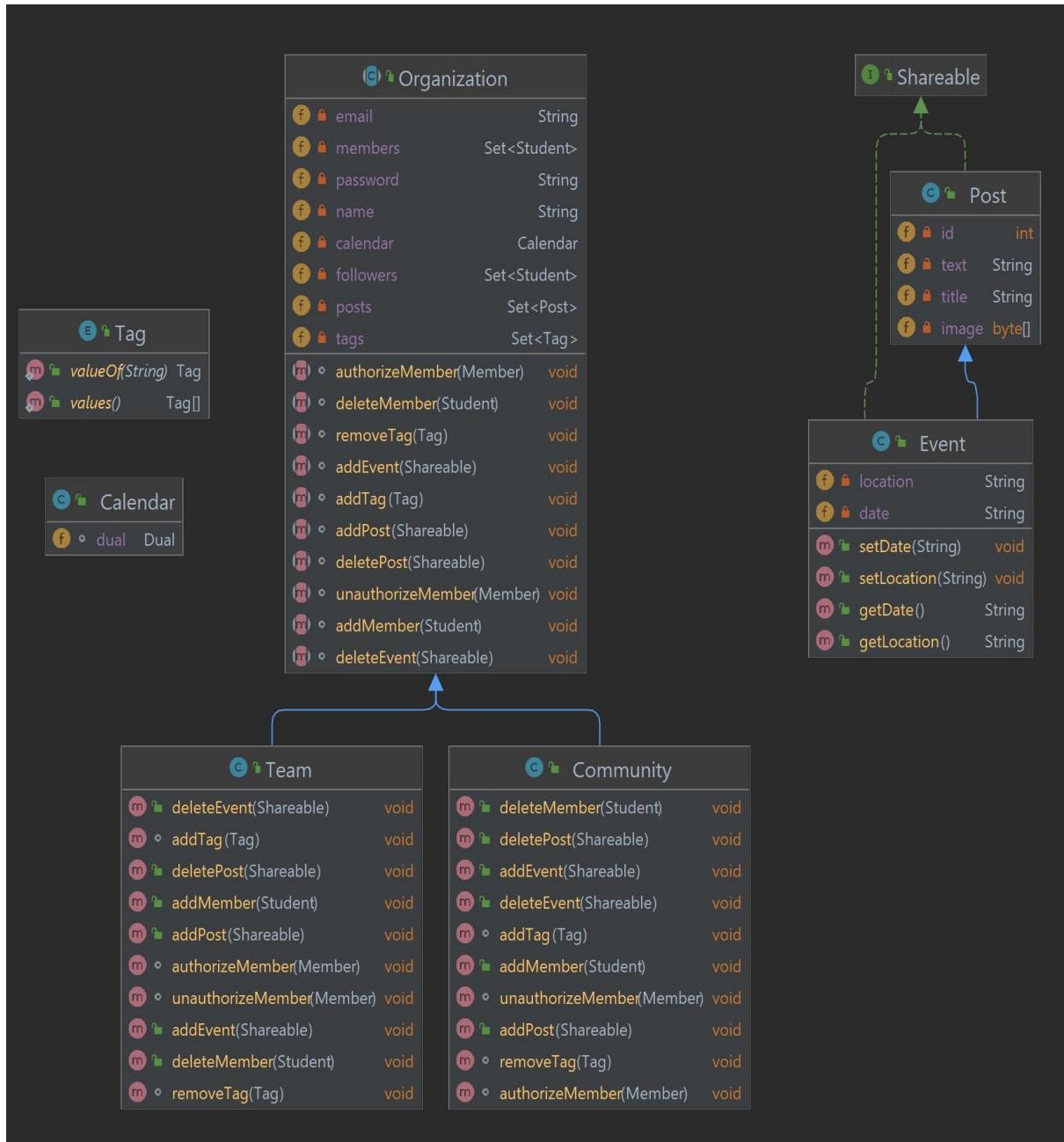


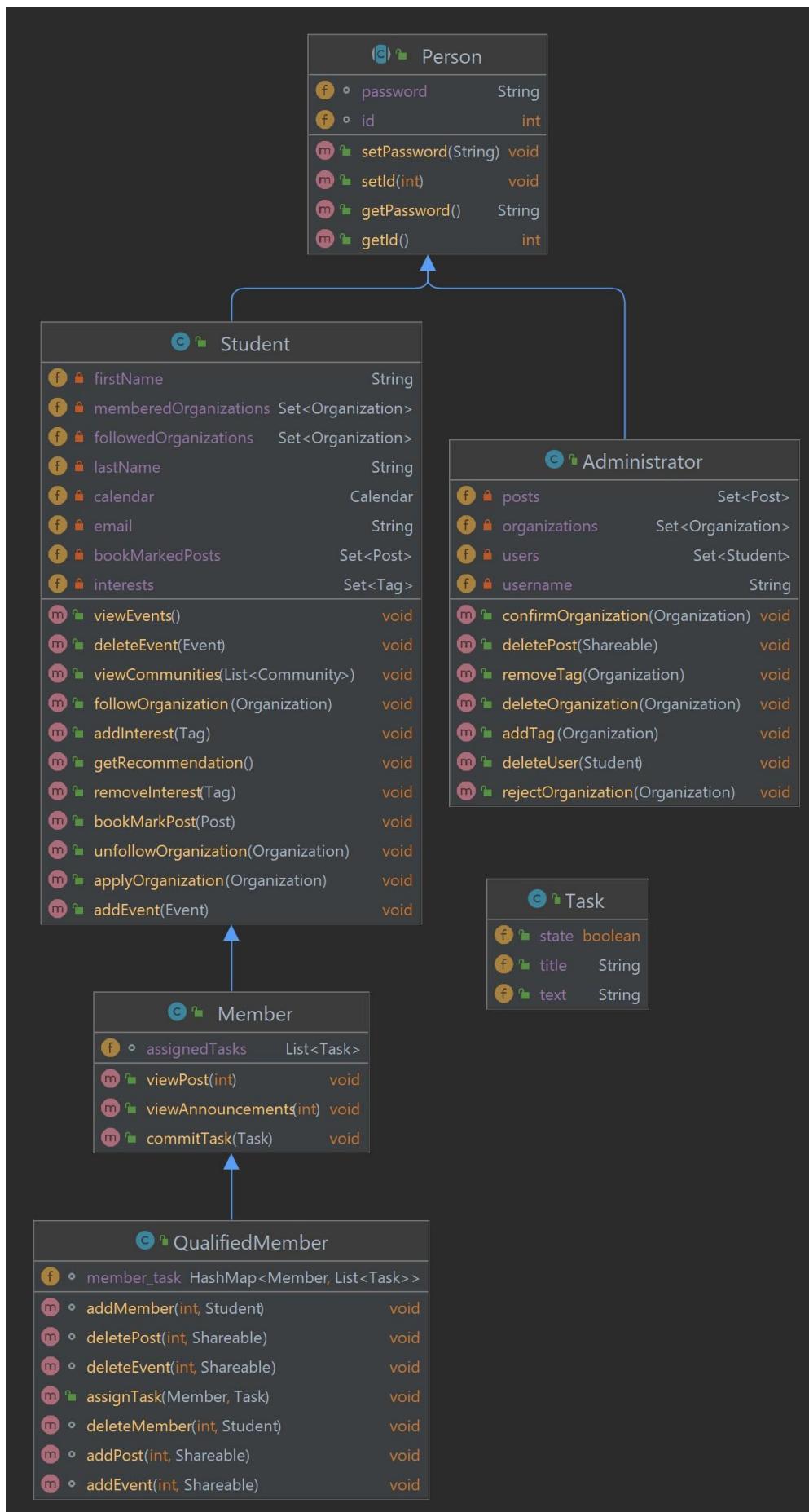






Class Diagrams





Test Cases

Test Case ID	HomePage-1	Test Case Description	View the app home page by entering the correct URL or by opening the mobile app.		
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass
S #	Prerequisites:				Test Data Requirement:
1	A web browser or mobile application				# No data requires
Test Scenario:		The user opens the web or mobile application.			
Test Conditions					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp	
1	Enter the URL to the browser or open the mobile app.	The application opens and the user displays the “global” posts on the home page.	The application opens and the user displays the “global” posts on the home page.	Pass	

Test Case ID	SignUp-1	Test Case Description	Sign-up to the app as a GTU community or team profile.				
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
<hr/>							
S #	Prerequisites:		S #	Test Data Requirement:			
1	Link for the sign-up page on the navigation		1	Username			
2	Link for the sign-up for community/team profile		2	Name of the community or the team			
			3	Correct e-mail (@gtu.edu.tr)			
			4	Incorrect e-mail			
			5	Password			
			6	Incompatible password			
<hr/> Test Scenario:		The user opens the sign-up page for communities/teams and sign-up for the community/team profile.					
<hr/> Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	Enter all data correctly	Successful sign-up		Successful sign-up	Done		
2	Enter incorrect e-mail	Show error message		Show error message	Done		
3	Enter incompatible passwords	Show error message		Show error message	Done		
4	Leave any input field empty	Show error message		Show error message	Done		
5	Press log-in button	Navigate to the log-in page		Navigate to the log-in page	Done		

Test Case ID	SignUp-2	Test Case Description	Sign-up to the app as a GTU student.				
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:				Test Data Requirement:		
1	Link for the sign-up page on the navigation		1	Username			
2	Link for the sign-up for student profile		2	Name and surname			
			3	Correct e-mail (@gtu.edu.tr)			
			4	Incorrect e-mail			
			5	Password			
			6	Incompatible password			
Test Scenario:		The user opens the sign-up page for the students and sign-up for the student profile.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	Enter all data correctly	Successful sign-up		Successful sign-up	Pass		
2	Enter incorrect e-mail	Show error message		Show error message	Pass		
3	Enter incompatible passwords	Show error message		Show error message	Pass		
4	Leave any input field empty	Show error message		Show error message	Pass		
5	Press log-in button	Navigate to the log-in page		Navigate to the log-in page	Pass		

Test Case ID	LogIn-1	Test Case Description	Log-in to a community/team account.		
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass
S #	Prerequisites:				S #
1	Link for the log-in page on the navigation				1
2	Link for the log-in as community/team.				2
					3
					4
Test Scenario:		The user opens the log-in page for the communities/teams and log-in to the community/team account.			
Test Conditions					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp	
1	Enter a valid username and a valid password	Successful login	Successful login	Pass	
2	Enter an invalid username or invalid password	Show an error message	Show an error message	Pass	
3	Leave any input field empty	Show an error message	Show an error message	Pass	
4	Press sign-up button	Navigate to the sign-up page	Navigate to the sign-up page	Pass	

Test Case ID	LogIn-2	Test Case Description	Log-in to a student account.					
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
S #	Prerequisites:				Test Data Requirement:			
1	Link for the log-in page on the navigation				1 Valid username			
2	Link for the log-in as student				2 Invalid username			
					3 Password			
					4 Invalid password			
Test Scenario:	The user opens the log-in page for the students and log-in to the student account.							
Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	Enter a valid username and a valid password	Successful login	Successful login	Pass				
2	Enter invalid username or invalid password	Show error message	Show error message	Pass				
3	Leave any input field empty	Show error message	Show error message	Pass				
4	Press sign-up button	Navigate to sign-up page	Navigate to sign-up page	Pass				

Test Case ID	Interests-1	Test Case Description	Listing communities/teams according to the user interests.					
Created By	Yusuf Arslan	Date Tested	1.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
<hr/>								
S #	Prerequisites:		S #	Test Data Requirement:				
1	First time log-in to the app		#	No data requires				
2	Existence of teams and communities in the program							
<hr/> Test Scenario:			The user log-in to the app the first time, then, a page where the user can select his/her interests opens.					
<hr/> Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	The user selects some of the fields of interest	The user is offered some communities or teams to follow.	The user is offered some communities or teams to follow.	Pass				
2	The user clicks the 'pass' button	Navigates to the home page.	Navigates to the home page.	Pass				

Test Case ID	ProjectIdea-1	Test Case Description	Sharing a project idea with the other people								
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass						
<hr/>											
S #	Prerequisites:		S #	Test Data Requirement:							
1	The user logged-in as a student		1	Title							
2	The user navigated his/her profile page		2	Text							
			3	Picture (optional)							
Test Scenario:		User shares his/her project idea via his/her profile.									
<hr/>											
Test Conditions											
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp						
1	The user clicks the 'add project idea' button	Project publishing window opens		Project publishing window opens	Pass						
2	User enters the project title, text, and a picture and then submits his/her idea	The project idea is added to the user's profile and project ideas page		The project idea is added to the user's profile and project ideas page	Pass						

Test Case ID	ProjectIdea-2	Test Case Description	Project Ideas page								
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass						
<hr/>											
S #	Prerequisites:		S #	Test Data Requirement:							
1	The user logged-in as a student		#	No data requires							
Test Scenario:		The user navigates the project ideas page and sees the owner of the ideas.									
<hr/>											
Test Conditions											
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp						
1	The user clicks the 'Project Ideas' button from the menu	Project ideas page opens		Project ideas page opens	Pass						

Test Case ID	Message-1	Test Case Description	Sending a message (from the student)		
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Not Executed
S #	Prerequisites:				Test Data Requirement:
1	The user logged-in to the app.				1 text
2	Existence of other profiles.				
Test Scenario:		The user clicks the message button to send a message to a user or a community/team.			
Test Conditions					
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp
1	The user clicks the message button.	The messages page appears on the screen.		#	Not Executed
2	The user selects a user to send a message.	Messages between users appear on the screen.		#	Not Executed
3	The user texts the message and presses the send button.	The message goes to the other user's message box.		#	Not Executed
4	User does not text the message to the box.	Send button does not work.		#	Not Executed

Test Case ID	UserProfile-1	Test Case Description	Navigating and displaying user profile page		
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass
S #	Prerequisites:				Test Data Requirement:
1	User logged-in				# No data requires.
2	Existence of the user on the app				
Test Scenario:		The user navigates and displays a user's profile page.			
Test Conditions					
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp
1	The user clicks on username	Profile page of the user opens		Profile page of the user opens	Pass

Test Case ID	CommunityProfile-1	Test Case Description	Navigating and displaying community profile page				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Not Executed		
S #	Prerequisites:	#	S #	Test Data Requirement:			
1	User logged-in		#	No data requires.			
2	Existence of the community on the app						
Test Scenario:		The user navigates and displays a community's profile page.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user clicks on community name	Profile page of the community opens		Profile page of the community opens	Pass		

Test Case ID	Follow-1	Test Case Description	Following a community or a team.				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:	#	S #	Test Data Requirement:			
1	Existence of communities/teams on the app		#	No data requires.			
2	Opened the community/team profile						
Test Scenario:		The user follows a community or team profile.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user clicks the community/team profile	Profile page of community/team opens		#	Pass		
2	The user clicks the 'follow' button	The user starts to follow the community/team		#	Pass		

Test Case ID	Search-1	Test Case Description	Searching any community/team with its name.				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	Working search bar		1	Community/team name exist in the app			
			2	Community/team name does not exist in the app			
Test Scenario:		The user makes a community/team search via the search bar.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	Type a community/team name that exists in the app	The profile page of that community/team opens		The profile page of that community/team opens	Pass		
2	Type a community/team name that does not exist in the app	Show an error message		Nothing shows	Pass		

Test Case ID	Search-2	Test Case Description	Searching any community/team with a tag.				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	Working search bar		1	A tag to search			
Test Scenario:		The user makes a community/team search via the search bar.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user enters a tag to the search bar	Related communities/teams will be shown		Related communities/teams will be shown	Pass		

Test Case ID	HomePage-2	Test Case Description	View the home page according to the following communities/teams					
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
Test Conditions								
S #	Prerequisites:		S #	Test Data Requirement:				
	1 The user logged-in to the app			# No data requires				
S #	2 The user has some following communities/teams							
Test Scenario:		The user opens the home page and changes the displayed screen from "global" to "followings."						
Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	The user changes the home page screen to the "followings."	Home pages are displayed with posts that communities that the following by the user.	Home pages are displayed with posts that communities that the following by the user.	Pass				

Test Case ID	Communities-1	Test Case Description	Displaying communities/teams profiles.					
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
Test Conditions								
S #	Prerequisites:		S #	Test Data Requirement:				
	1 The user opens the web or mobile application			# No data requires				
Test Scenario:		The user navigates the communities/teams page which includes all community and team profiles.						
Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	The user clicks the communities/teams icon from the menu	Communities/teams page opens	Communities/teams page opens	Pass				

Test Case ID	Like-1	Test Case Description	Giving a like to a post					
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Not Executed			
S #	Prerequisites:		S #	Test Data Requirement:				
	At least one of the communities/teams shared a post			No data requires				
Test Scenario:		The user clicks on the like icon on one of the community posts.						
Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	Logged-in user likes a post from one of the communities.	Like counter of the post is increased	Like counter of the post is increased	Pass				
2	Not logged-in user likes a post	The user is navigated to the log-in page.	The user is navigated to the log-in page.	Pass				

Test Case ID	Bookmarks-1	Test Case Description	Bookmarking a community post					
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
S #	Prerequisites:		S #	Test Data Requirement:				
	At least one of the communities shared a post			No data requires				
Test Scenario:		The user clicks on the bookmark icon on one of the community posts.						
Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	Logged in user bookmarks a post of one of the communities.	The post can be seen on the bookmarks page of the user.	The post can be seen on the bookmarks page of the user.	Pass				
2	Not logged-in user bookmarks a post	The user is navigated to the log-in page.	The user is navigated to the log-in page.	Pass				

Test Case ID	Bookmarks-2	Test Case Description	Displaying/editing bookmarks				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
<hr/>							
S #	Prerequisites:	#	S #	Test Data Requirement:			
1	The user logged-in		#	No data requires			
Test Scenario:		The user opens the bookmarks page and sees bookmarked posts.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user clicks the bookmarks button from the menu	Bookmarks page is opened		Bookmarks page is opened	Pass		
2	The user clicks the remove button	Bookmark removed from the bookmark list		Bookmark removed from the bookmark list	Pass		

Test Case ID	Calendar-1	Test Case Description	Displaying the calendar page.				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
<hr/>							
S #	Prerequisites:	#	S #	Test Data Requirement:			
1	The user logged-in		#	No data requires			
Test Scenario:		The user navigates to the calendar page and displays the calendar.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	User clicks the calendar link which is in the navigation bar	Calendar appears on the screen		Calendar appears on the screen	Pass		

Test Case ID	Event-1	Test Case Description	Creating an event (Community/Team)				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	Logged-in user		1	Event topic			
2	The user is an admin or authorized community/team member		2	Event description			
3	Navigated on calendar page		3	A picture (optional)			
4			4	Date and time			
Test Scenario:		The user adds an event.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	User opens the calendar page and clicks the '+' button.	The event publishing window opens.	The event publishing window opens.	Pass			
2	The user adds all required data and presses the publish button.	The event is published as a post and added to the community/team's calendar.	The event is published as a post and added to the community/team's calendar.	Pass			
3	Leave any required blank empty.	Publish button does not work.	Publish button does not work.	Pass			

Test Case ID	Event-2	Test Case Description	Deleting an event from the calendar (user)				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S # Prerequisites:							
1	Logged-in to the application as an admin or authorized member.		#	No data requires			
2	An event belonging to the community the user is a member of						
Test Scenario:		The user clicks the delete icon on an event in their calendar.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user clicks the calendar page.	The calendar page opened.	The calendar page opened.	Pass			
2	The user deletes an event from his/her calendar.	The event is deleted from the calendar.	The event is deleted from the calendar.	Pass			

Test Case ID	EditCommunityInfo-1	Test Case Description	Editing the community/team information.				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S # Prerequisites:							
1	The user logged-in to the app.		#	No data requires			
2	The user is the admin of the community/team.						
Test Scenario:		The user clicks the edit button on the community/team profile.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user navigates the community/team profile	Profile page is opened.	Profile page is opened.	Pass			
2	The user edits the information part of the community and clicks save.	The information is changed successfully.	The information is changed successfully.	Pass			

Test Case ID	Roles-1	Test Case Description	Labeling members role								
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass						
<hr/>											
S #	Prerequisites:	#	S #	Test Data Requirement:							
1	The user logged-in to the app.		#	No data requires							
2	The user is an admin of a community/team										
3	Existence of some members of the group										
Test Scenario:		The user clicks the edit button on the community/team profile.									
<hr/>											
Test Conditions											
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp						
1	The user navigates the community/team management page.	The management page opens.		The management page opens.	Pass						
2	The user clicks a member profile.	The profile page of the member opens.		The profile page of the member opens.	Pass						
3	The user clicks the assign role button and assigns a role.	The user will be labeled with his/her new role		The user will be labeled with his/her new role	Pass						

Test Case ID	Task-1	Test Case Description	Assigning tasks to the community members. (Community)				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	The user logged-in to the app.		#	No data requires			
2	The user is authorized or an admin in the community.						
Test Scenario:		The user assigns tasks to any member of the community on the community management page.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user opens the member list from the community/team management page	The list is opened	The list is opened	Pass			
2	The user chooses a member of the community and assigns a task by entering a text.	The task is assigned to the member and can be seen on his/her page by authorized members or admin	The task is assigned to the member and can be seen on his/her page by authorized members or admin	Pass			

Test Case ID	Task-2	Test Case Description	Viewing/editing a task				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Not Executed		
S #	Prerequisites:		S #	Test Data Requirement:			
1	The user logged-in to the app.		#	No data requires			
2	The user is a member of a community/team						
3	Assigned tasks to the user						
Test Scenario:		The user assigns tasks to any member of the community on the community management page.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user navigates to the tasks page.	Tasks page opened		Tasks page opened	Pass		
2	The user switches the toggle to done.	Task added to completed tasks		Task added to completed tasks	Pass		
3	The user switches the toggle to incomplete.	Task added to incomplete tasks		Task added to incomplete tasks	Pass		

Test Case ID	Permissions-1	Test Case Description	Editing permissions of the members in the community				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	The user logged-in to the app.		#	No data requires			
2	The user is the admin in the community.						
Test Scenario:		The user authorizes or deauthorizes the members.					
Test Conditions							
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp		
1	The user authorizes a member	The member is authorized.		The member is authorized.	Pass		
2	The user deauthorizes a member	The member is deauthorized.		The member is deauthorized.	Pass		

Test Case ID	Post-1	Test Case Description	Sharing a post (community/team)						
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass				
<hr/>									
S #	Prerequisites:			S #	Test Data Requirement:				
1	The user logged-in to the app.			1	Title				
2	The user is authorized or admin in the community.			2	Text				
				3	Photo				
<hr/>									
Test Scenario:		The user shares a new post							
<hr/>									
Test Conditions									
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp				
1	The user shares a new post with a text or photo or both for only members of the community.	The post is seen by the members of the community.		#	Pass				
2	The user shares a new post with a text or photo or both for global.	The post can be seen on the global homepage and the followers and the members of the communities can also see the post on their followings and memberships pages.		#	Pass				
3	The user shares a new post with no texts or photos	Error message shown to screen		#	Pass				

Test Case ID	Post-2	Test Case Description	Deleting a post (community/team)					
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
Test Scenarios:								
S #	Prerequisites:	#	S #	Test Data Requirement:				
1	The user logged-in to the app.		#	No data requires				
2	The user is authorized or an admin in the community.	#						
Test Scenario: The user deletes a post								
Test Conditions								
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user deletes the post	Post is deleted and cannot be seen by anyone.		Post is deleted and cannot be seen by anyone.	Pass			

Test Case ID	Attendance-1	Test Case Description	The user indicates that he/she will attend an event.					
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
Test Scenarios:								
S #	Prerequisites:	#	S #	Test Data Requirement:				
1	The user logged-in to the app.		#	No data requires				
2	The user should be a member of the community/team.	#						
3	A published event							
4	The event has not expired.	#						
Test Scenario: The user switches the toggle button for attending.								
Test Conditions								
Step #	Step Details	Expected Results		Actual Results	Pass / Fail / Not Exc. / Susp			
1	The member switches the attendance toggle.	The member is marked on the attendance form.		The member is marked on the attendance form.	Pass			

Test Case ID	Attendance-2	Test Case Description	Editing attendance form.		
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass
S #	Prerequisites:				Test Data Requirement:
1	The user logged-in to the app.				1 List of the community members' first and last names.
2	The user is authorized or an admin in the community.				
3	A published event				
4	The event should be done				
5	At least one member attended the event				
Test Scenario:		The user edits and views the attendance form.			
Test Conditions					
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp	
1	The user marks the member as not participating in the attendance form.	The user is deleted from the list of participants.	The user is deleted from the list of participants.	Pass	

Test Case ID	Join-1	Test Case Description	Applying to become a member of a community/team.				
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	The user logged-in to the app.		#	Community/team name			
2	At least one community/team						
Test Scenario:		Students apply to be a member of a community/team.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user types the community/team name into the search bar	Community/team profile opens	Community/team profile opens	Pass			
2	The user presses the join button	A request to join the community/team is sent.	A request to join the community/team is sent.	Pass			

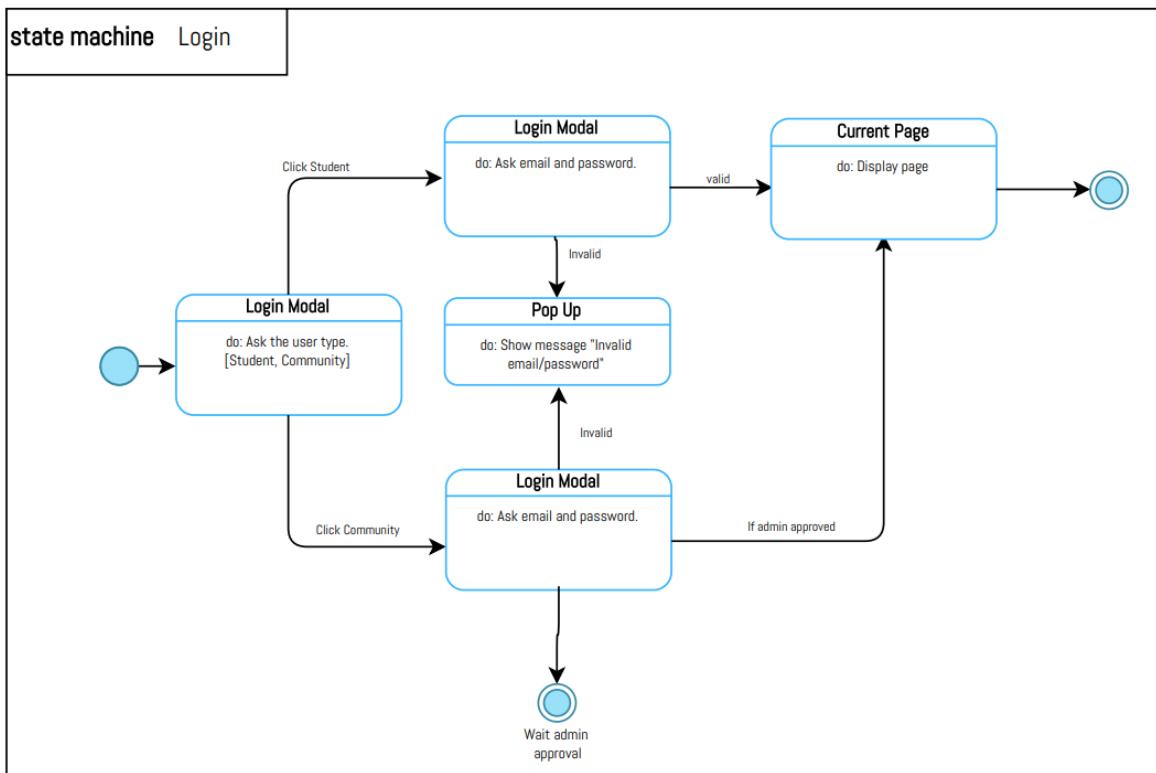
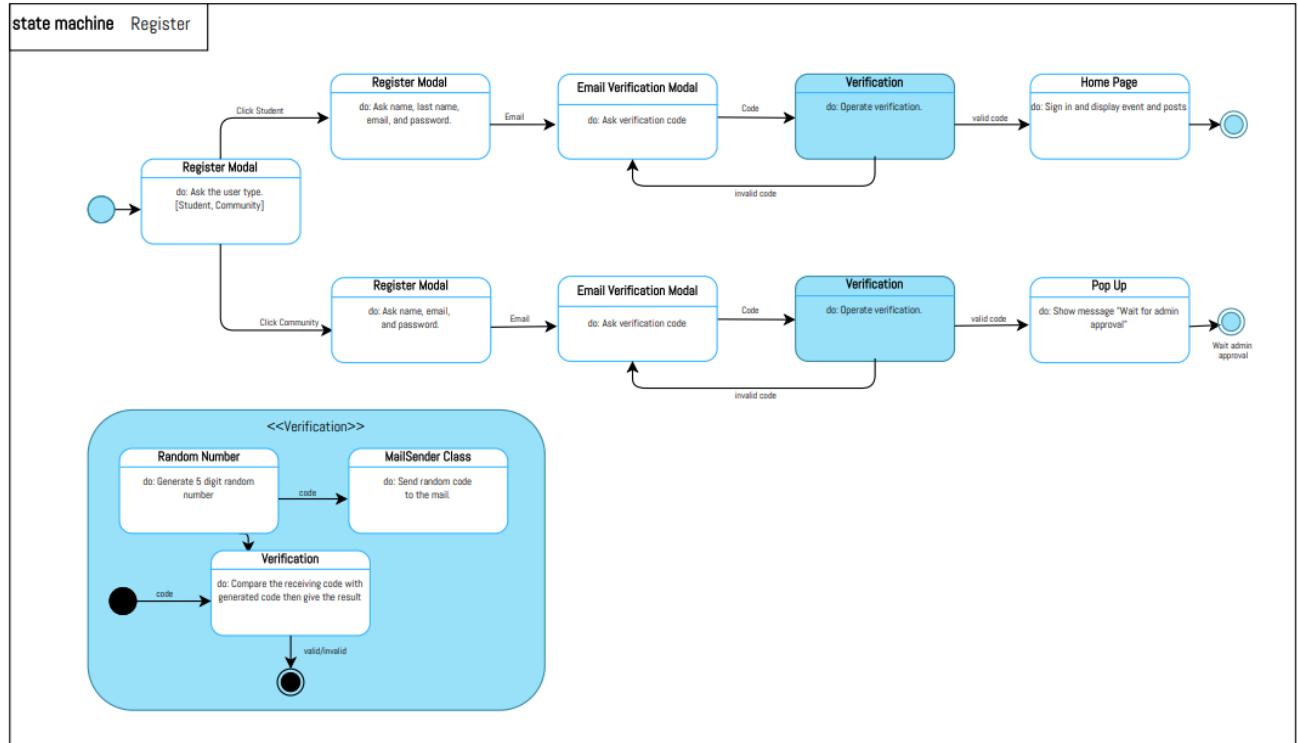
Test Case ID	Member-1	Test Case Description	An admin or authorized user approves/declines the applications.								
Created By	Yusuf Arslan	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass						
<hr/>											
S #	Prerequisites:		S #	Test Data Requirement:							
1	The user logged-in		#	No data requires							
2	The user is an admin or authorized member of the community/team										
3	At least one user sent a request to be a member										
4	Management page is opened										
Test Scenario:		Admin opens the application list and approves/declines applications.									
<hr/>											
Test Conditions											
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp							
1	User opens the applicant list and approves application	The applicant becomes a member of the community/team	The applicant becomes a member of the community/team	Pass							
2	User opens the applicant list and declines application	The user removed from the application list	The user removed from the application list	Pass							

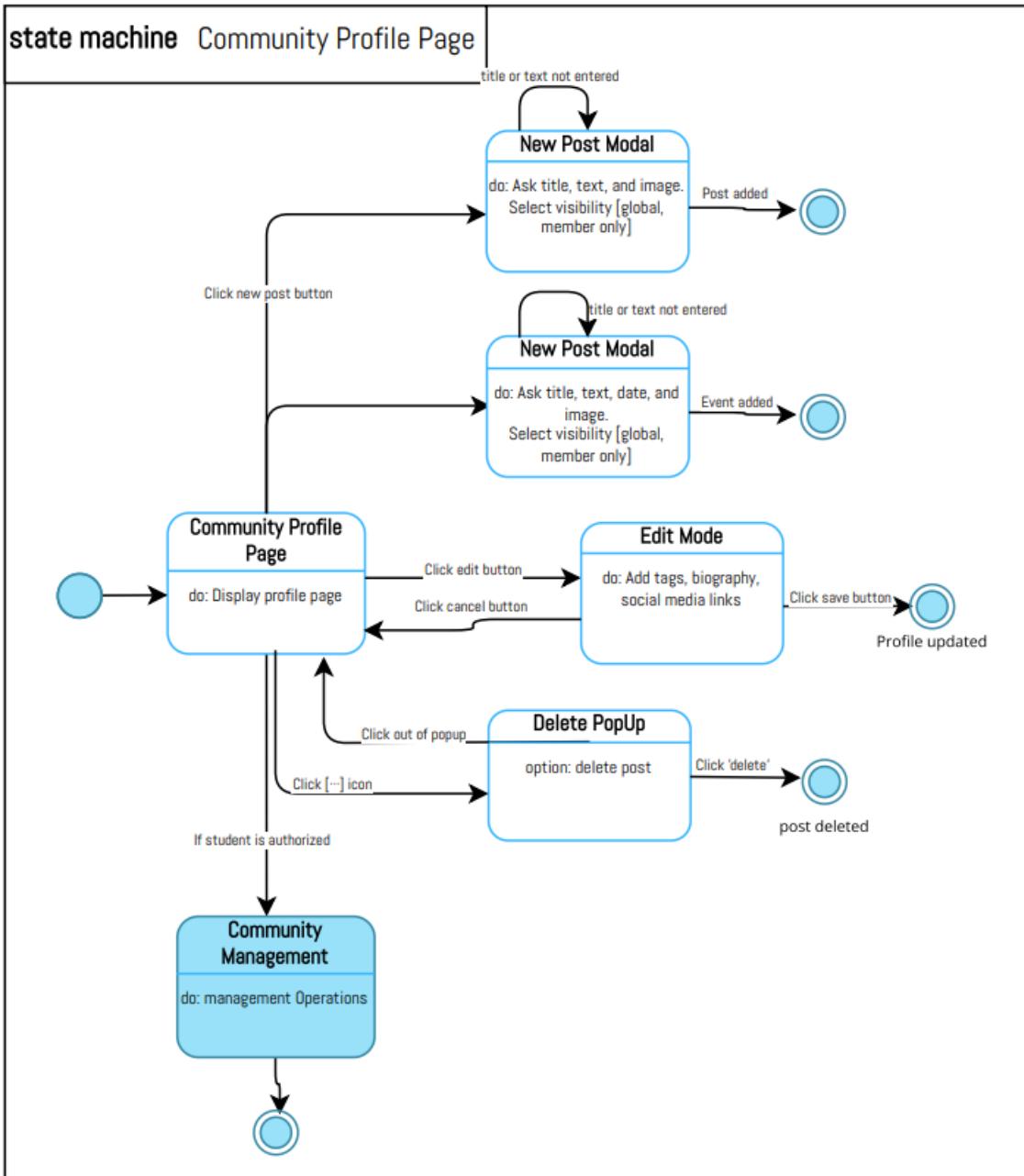
Test Case ID	Member-2	Test Case Description	Deleting a member from the community.					
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass			
<hr/>								
S #	Prerequisites:		S #	Test Data Requirement:				
1	The user logged-in to the app.		#	No data requires.				
2	The user is authorized or an admin in the community.							
<hr/> Test Scenario:			The user deletes a member from the community.					
<hr/> Test Conditions								
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp				
1	The user removes a member from the community	Removed user disappeared from the member list.	Removed user disappeared from the member list.	Pass				

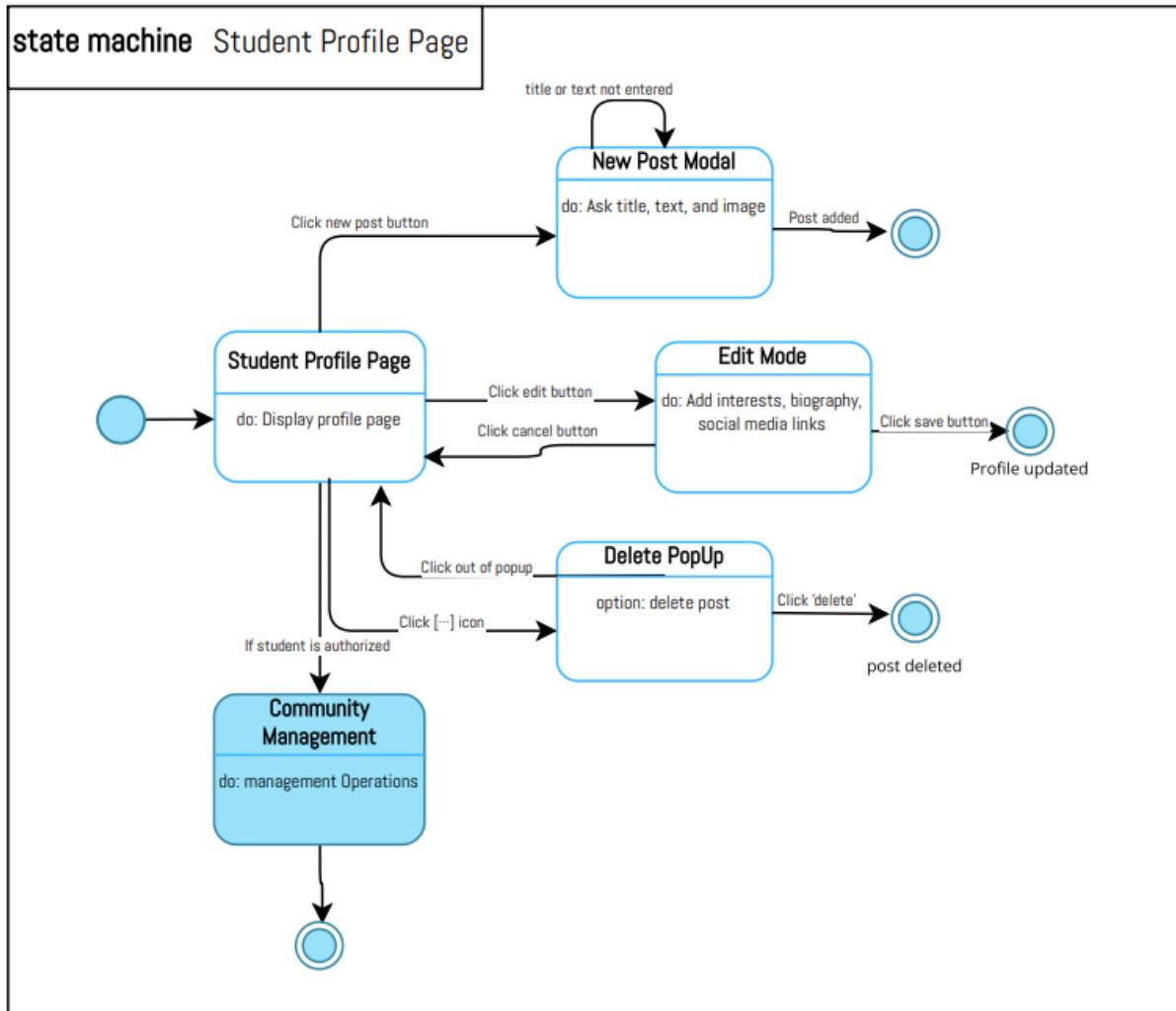
Test Case ID	Message-2	Test Case Description	Sending a message to a user (community)				
Created By	Ahmet Kadir Aksu	Date Tested	#	Result (Pass/Fail/Not Exc.)	Not Executed		
S #	Prerequisites:		S #	Test Data Requirement:			
1	Admin or authorized community/team member logged-in app		1	Text			
2	A user to send message						
Test Scenario:		The community sends a message to a user.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user clicks the message button.	The messages page appears on the screen.	#	Not Executed			
2	The user selects a user to send a message.	Messages between users screen appear on the screen.	#	Not Executed			
3	The user text the message and press the send button.	The message goes to the other user's message box.	#	Not Executed			
4	User does not text the message to the box.	Send button does not work.	#	Not Executed			

Test Case ID	LogOut-1	Test Case Description	Logging out				
Created By	Ahmet Kadir Aksu	Date Tested	5.01.2023	Result (Pass/Fail/Not Exc.)	Pass		
S #	Prerequisites:		S #	Test Data Requirement:			
1	The user logged-in to the app.						
Test Scenario:		The user logs out from the website/app.					
Test Conditions							
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not Exc. / Susp			
1	The user clicks the logout button	User logged out successfully	User logged out successfully	Pass			

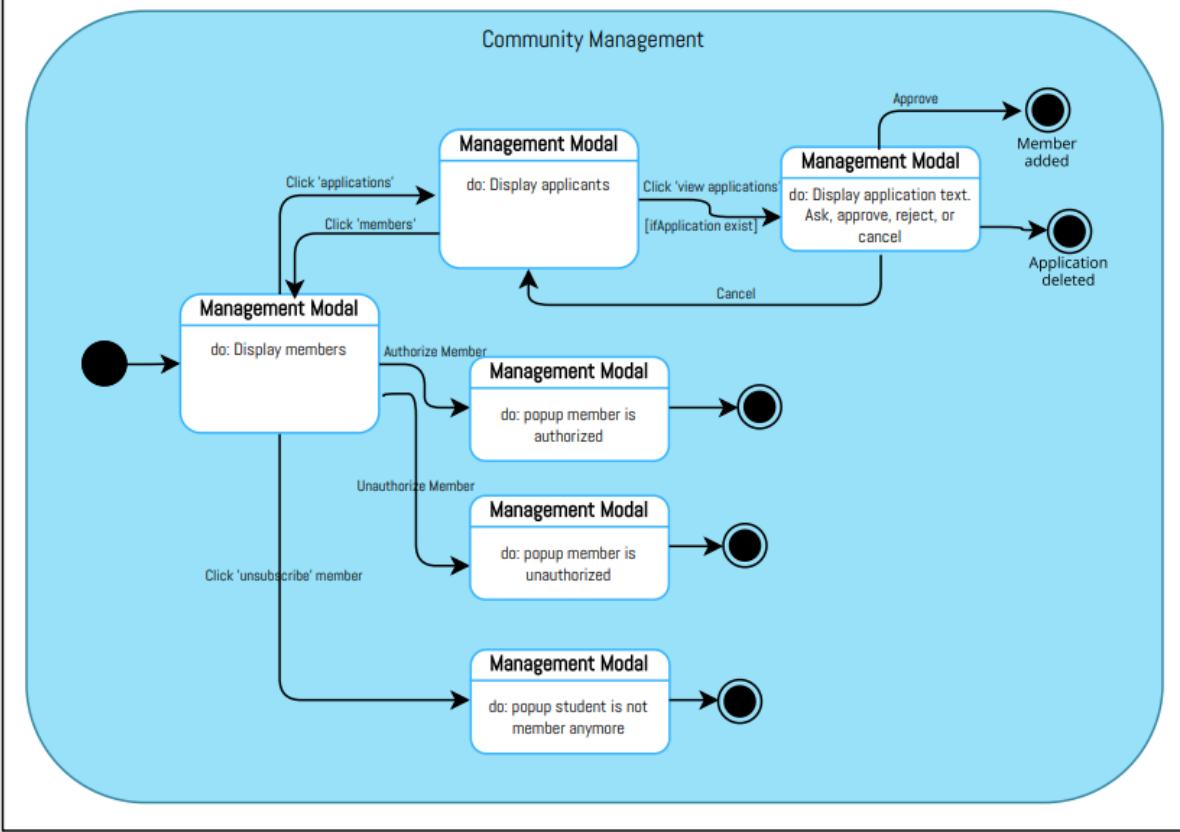
State Diagrams

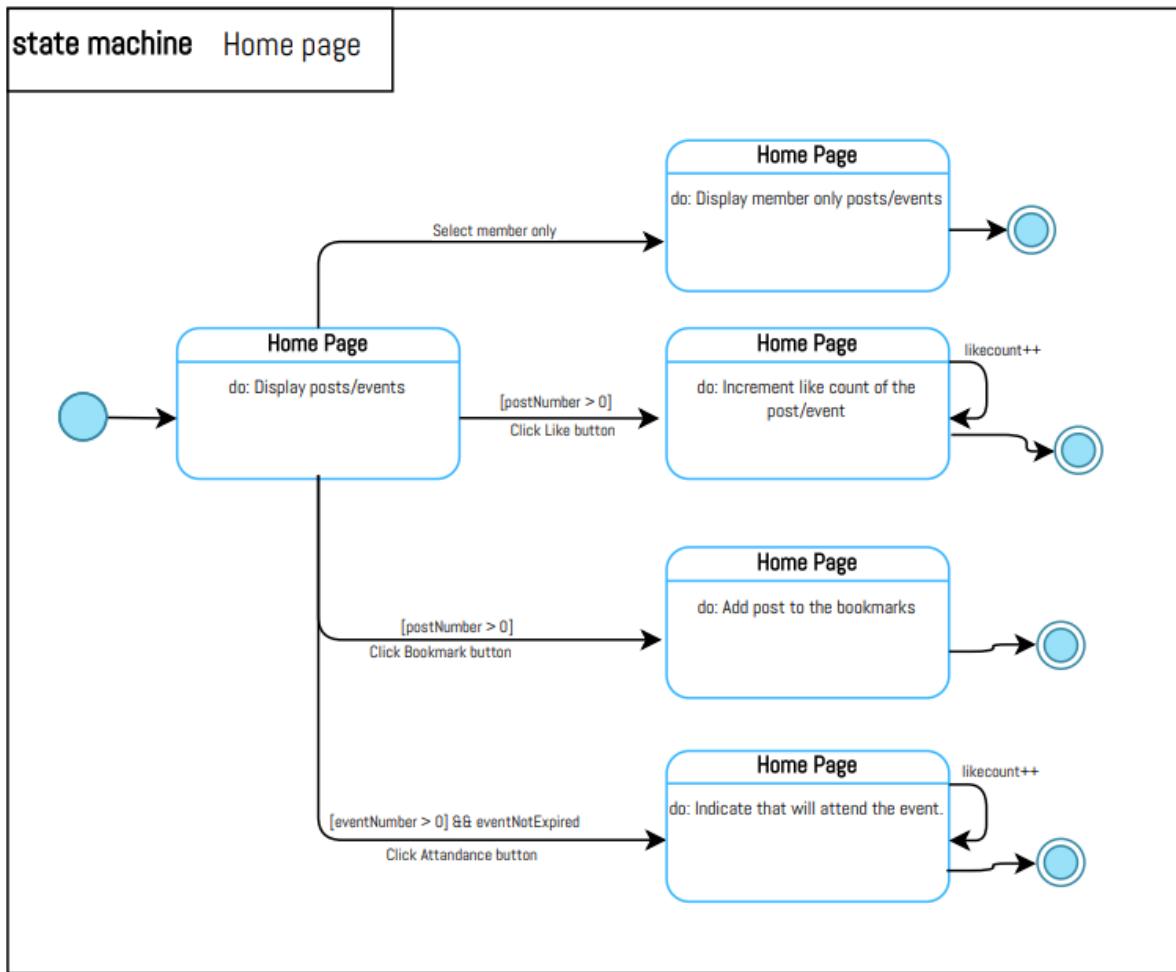


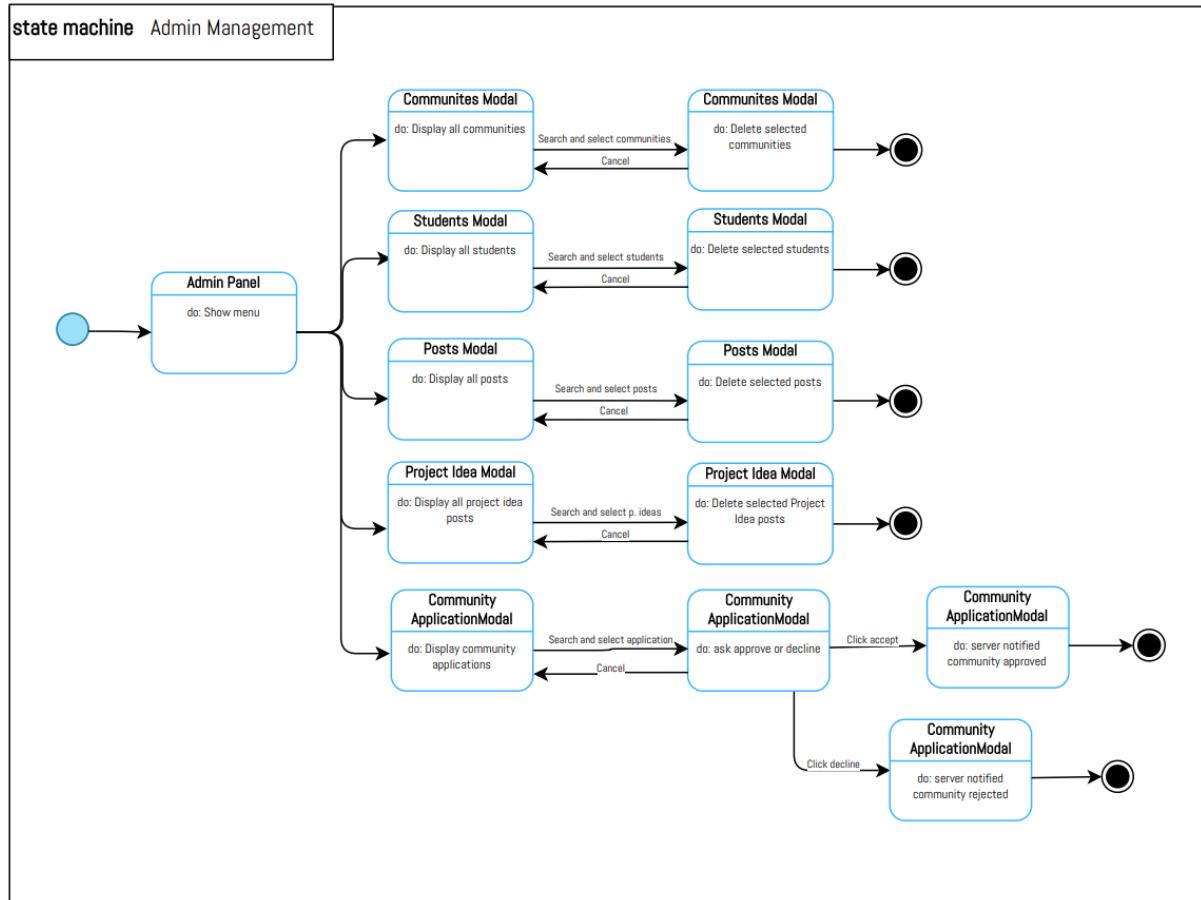




state machine Community Management







Simple System Architecture

Introduction

The purpose of this document is to describe the system architecture of the community management and interaction system for GTU students and communities. The system consists of a web app, mobile app, and admin panel, and is designed to enable students and communities to interact with each other and share information.

The main features of the system include:

- Profile pages for communities with profile picture, banner, tags, brief explanation about general purpose of the community, posts, and information about followers and members.
- Profile pages for students include profile picture, banner, interests, social media links, and project ideas, if any.
- The ability for communities to create and share posts, including events, management of members.
- The ability for students to browse and follow communities, apply for being member of communities, and see the posts of communities they are following, or they are member of, on their home page.
- An admin panel that allows the admin to manage the system and perform tasks such as community approval or rejection.

System Architecture

The community management and interaction system consist of a web app, mobile app, and admin panel, which are connected to a shared backend REST API and database.

The web app and mobile app provide the main interface for students and communities to interact with the system. They allow students to browse and follow communities, view, and create posts, and access their own profile information. The apps communicate with the backend REST API to retrieve and store data, using HTTP requests and responses.

The admin panel is a web-based interface that enables the administrator to manage the system. It allows the admin to view and edit user and community information, approve, or reject communities and perform other tasks as needed. The admin panel also communicates with the backend REST API to retrieve and store data.

The backend REST API is built using Spring Boot and connects to a MySQL database. It provides a set of APIs that enable the web and mobile apps, as well as the admin panel, to access data and perform operations on the database. The database stores user and community information and relations between them, as well as posts and events. The system is deployed on Azure cloud services, providing a secure and reliable hosting platform for the REST API and database.

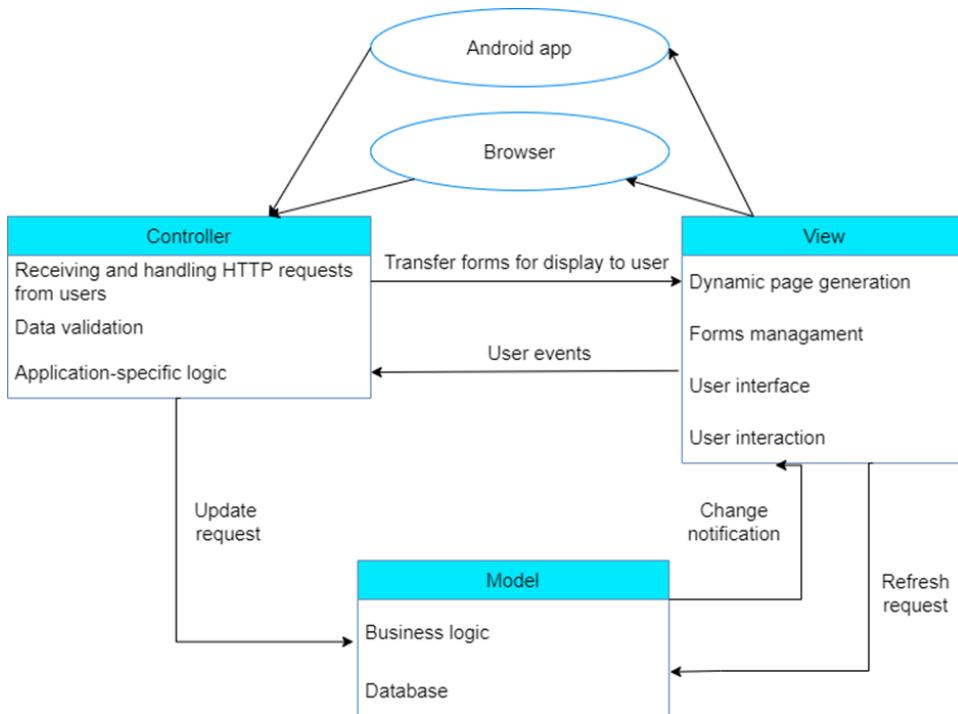


Diagram1: CMIS architecture using the MVC pattern

The MVC architecture diagram for the community management and interaction system consists of three main components: the model, the view, and the controller. The model represents the data and business logic of the system, and is responsible for storing and manipulating data, as well as implementing business rules. The view represents the user interface of the system and is responsible for displaying data to the user and providing a means for the user to interact with the system. The controller acts as the intermediary between the model and the view, and is responsible for receiving user actions, invoking actions on the model, and updating the view with the results. The MVC architecture enables a clear separation of concerns and promotes modular and maintainable code.

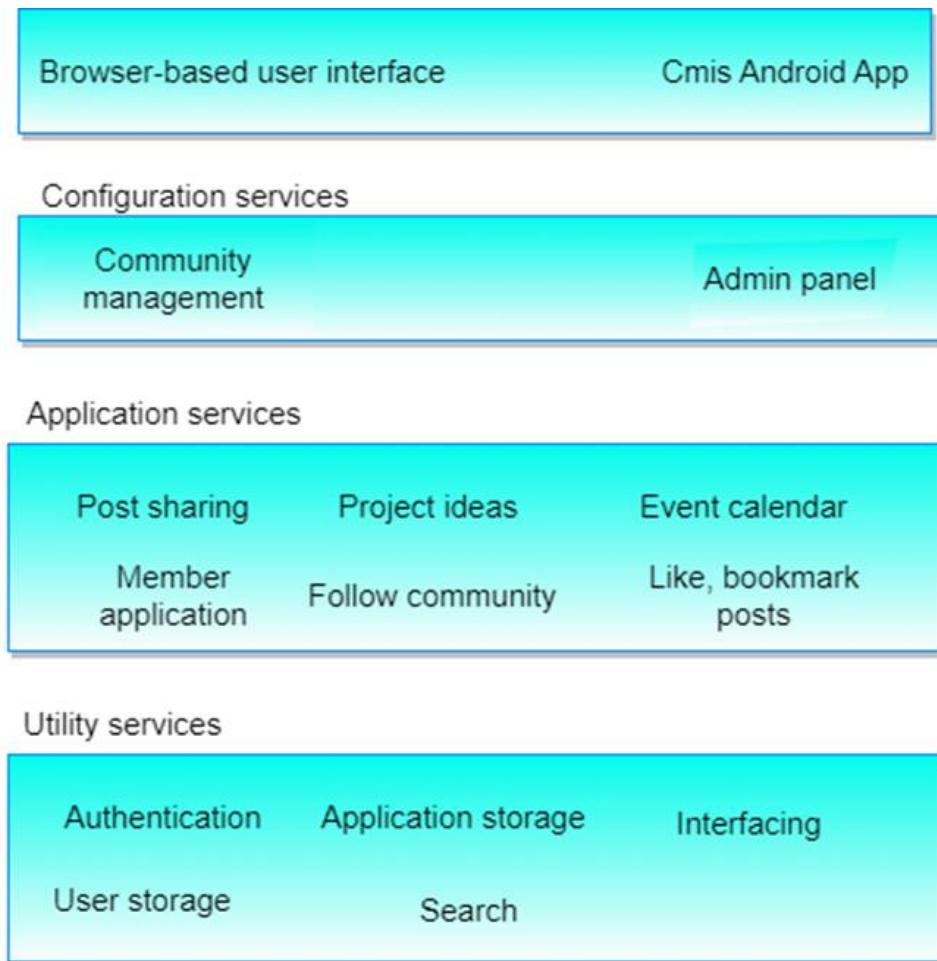


Diagram2: The layered architecture of CMIS

The layered architecture of the CMIS consists of four main layers: the top layer, the second layer, the third layer, and the fourth layer. The top layer includes the browser-based user interface and the CMIS Android app, which provide a means for users to interact with the system. The second layer, the configuration services layer, contains the community management and admin panel components, which allow for the management and configuration of the system. The third layer, the application services layer, includes the post sharing, project ideas, event calendar, member application, follow community, and like/bookmark posts components, which provide the core functionality of the CMIS. The fourth layer, the utility services layer, contains the authentication, application storage, interfacing, user storage, and search components, which provide support for the other layers and ensure the smooth operation of the CMIS.

Key Technologies and Frameworks

The community management and interaction system are built using the following technologies and frameworks:

Backend: Spring Boot REST API, MySQL database

Frontend: React for the web app, React Native for the mobile app

Deployment: Azure cloud services

The Spring Boot REST API is used to build the backend of the system, providing a set of APIs that enable the web and mobile apps to access data from the MySQL database. The MySQL database is used to store user and community information, as well as posts and events.

The web app and mobile app are built using React and React Native, respectively. These frameworks provide a set of components and tools for building user interfaces, allowing the system to be implemented in a modular and scalable way.

Finally, the system is deployed on Azure cloud services, providing a secure and reliable hosting platform for the REST API and database.

Advantages of the Selected Architecture

The selected architecture for the community management and interaction system has several advantages, including:

Scalability: The use of a REST API and a database allows the system to scale up or down as needed, without requiring significant changes to the architecture. This makes it easy to handle increased load or user demand and helps to ensure that the system can continue to operate effectively.

Maintainability: The modular design of the system, with separate components for the web and mobile apps, backend API, and admin panel, makes it easier to maintain and update the system over time.

Performance: The use of modern technologies and frameworks, such as Spring Boot and React, can help to ensure that the system performs well and provides a good user experience. The use of Azure cloud services also helps to ensure that the system is highly available and can handle traffic spikes.

Security: The use of a cloud hosting platform such as Azure helps to ensure that the system is secure and protected against threats such as cyber-attacks. The use of a database also allows for the storage of sensitive data in a secure and controlled manner.

Flexibility: The ability to implement the system as a web app and mobile app allows for flexibility in terms of user access and platform support. This can be beneficial for users who prefer to use one platform over the other, or for users who need to access the system from different devices

Conclusion

In summary, the community management and interaction system is a web app, mobile app, and admin panel that enables GTU students and communities to interact with each other and share information. The system is built using Spring Boot, React, and React Native, and is deployed on Azure cloud services.

The main advantages of the selected architecture include its scalability, maintainability, performance, security, and flexibility.

Thank you for reading this document. If you have any questions or comments, please don't hesitate to contact us.

References:

Spring Boot documentation: <https://spring.io/projects/spring-boot>

React documentation: <https://reactjs.org/>

React Native documentation: <https://reactnative.dev/>

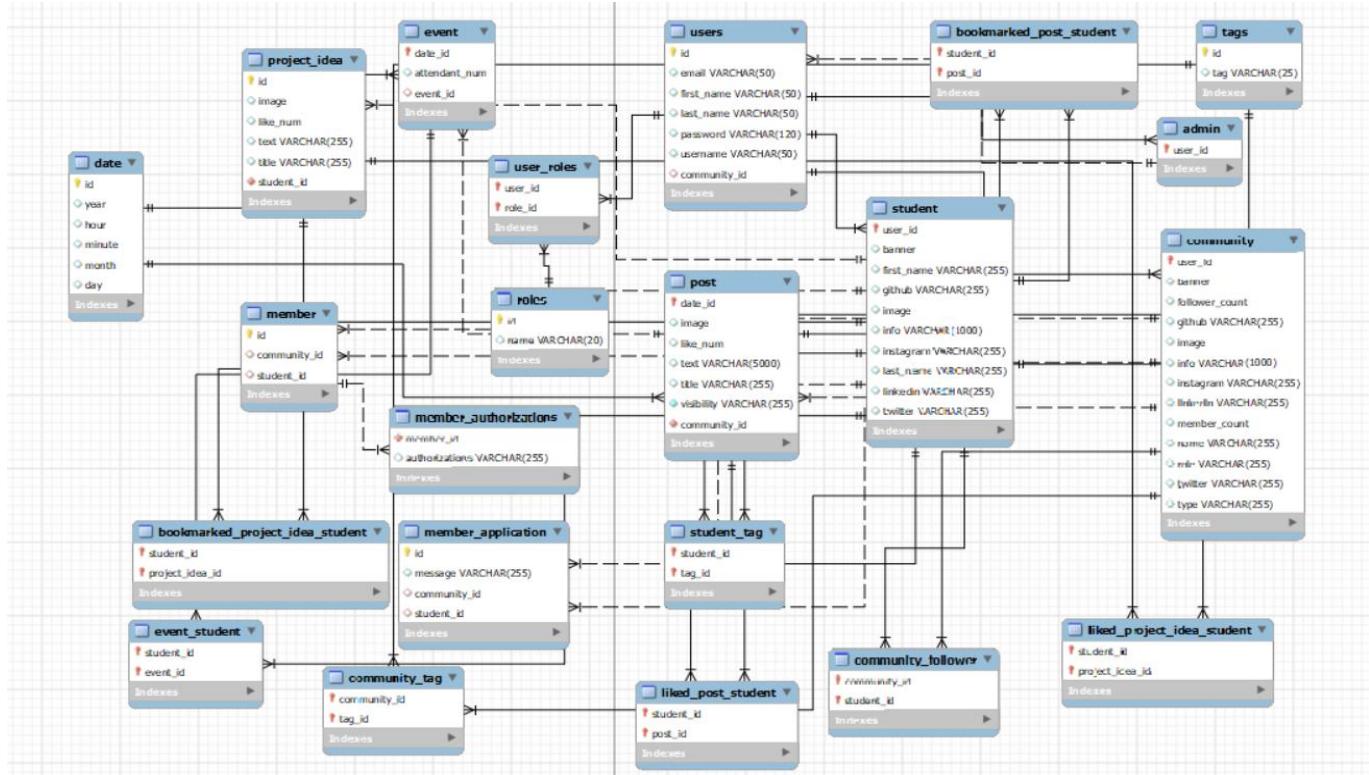
Azure cloud services documentation: <https://azure.microsoft.com/en-us/services/>

Backend

Documentation Link: [request_doc](#)

Cloud Link: [cloud](#)

Database Relation Diagram



Backend Server Requests

user-controller

PUT	/api/cmis/users/{id}/password
GET	/api/cmis/users
DELETE	/api/cmis/users
GET	/api/cmis/users/{id}
DELETE	/api/cmis/users/{id}
GET	/api/cmis/users/search
GET	/api/cmis/unverified/search
GET	/api/cmis/students/search
GET	/api/cmis/communities/search
GET	/api/cmis/communities/nameDesc
GET	/api/cmis/communities/nameAsc

student-controller

<code>GET</code>	/api/cmis/students/{id}	^
<code>PUT</code>	/api/cmis/students/{id}	▼
<code>DELETE</code>	/api/cmis/students/{id}	▼
<code>PUT</code>	/api/cmis/students/{id}/updateImage	▼
<code>POST</code>	/api/cmis/users/{userId}/students	▼
<code>GET</code>	/api/cmis/users/{id}/students	▼
<code>GET</code>	/api/cmis/students	▼
<code>DELETE</code>	/api/cmis/students	▼
<code>GET</code>	/api/cmis/students/{studentId}/memberOf	▼
<code>GET</code>	/api/cmis/students/{studentId}/isMemberOf/{communityId}	▼
<code>GET</code>	/api/cmis/students/{studentId}/isFollowerOf/{communityId}	▼
<code>GET</code>	/api/cmis/students/{id}/image	▼
<code>GET</code>	/api/cmis/events/{eventId}/attendants	▼
<code>GET</code>	/api/cmis/communities/{communityId}/followers	▼

community-controller

<code>GET</code>	/api/cmis/communities/{id}	^
<code>PUT</code>	/api/cmis/communities/{id}	▼
<code>DELETE</code>	/api/cmis/communities/{id}	▼
<code>PUT</code>	/api/cmis/communities/{id}/updateImage	▼
<code>PUT</code>	/api/cmis/communities/{id}/updateBanner	▼
<code>POST</code>	/api/cmis/users/{userId}/communities	▼
<code>GET</code>	/api/cmis/students/{followerId}/followingCommunities	▼
<code>POST</code>	/api/cmis/students/{followerId}/followingCommunities	▼
<code>GET</code>	/api/cmis/teams	▼
<code>GET</code>	/api/cmis/tags/{id}/communities	▼
<code>GET</code>	/api/cmis/communities	▼
<code>DELETE</code>	/api/cmis/communities	▼
<code>GET</code>	/api/cmis/communities/{id}/image	▼
<code>DELETE</code>	/api/cmis/communities/{id}/image	▼
<code>GET</code>	/api/cmis/communities/{id}/banner	▼
<code>DELETE</code>	/api/cmis/communities/{id}/banner	▼
<code>DELETE</code>	/api/cmis/students/{followerId}/followingCommunities/{communityId}	▼

member-controller

GET	/api/cmis/communities/{communityId}/members/{studentId}	^
PUT	/api/cmis/communities/{communityId}/members/{studentId}	▼
DELETE	/api/cmis/communities/{communityId}/members/{studentId}	▼
PUT	/api/cmis/communities/{communityId}/memberApplications/{studentId}/accept	▼
PUT	/api/cmis/communities/{communityId}/memberApplications/{applicantId}/reject	▼
POST	/api/cmis/communities/{communityId}/apply/{studentId}	▼
POST	/api/cmis/communities/{communityId}/adminAddMember/{studentId}	▼
GET	/api/cmis/communities/{communityId}/members	▼
DELETE	/api/cmis/communities/{communityId}/members	▼
GET	/api/cmis/communities/{communityId}/memberApplications	▼
GET	/api/cmis/communities/{communityId}/memberApplications/{studentId}	▼
GET	/api/cmis/communities/{communityId}/authorizedMembers	▼
DELETE	/api/cmis/communities/{communityId}/cancelApplication/{studentId}	▼

post-controller

POST	/api/cmis/students/{studentId}/posts/{postId}/like	▼
GET	/api/cmis/students/{studentId}/bookmarkedPosts	▼
POST	/api/cmis/students/{studentId}/bookmarkedPosts	▼
GET	/api/cmis/communities/{communityId}/posts	▼
POST	/api/cmis/communities/{communityId}/posts	▼
DELETE	/api/cmis/communities/{communityId}/posts	▼
GET	/api/cmis/students/{studentId}/likedPosts	▼
GET	/api/cmis/posts	▼
GET	/api/cmis/posts/{studentId}/private	▼
GET	/api/cmis/posts/{postId}/likes	▼
GET	/api/cmis/posts/{postId}/likeNum	▼
GET	/api/cmis/posts/{postId}/isLikedByStudent/{studentId}	▼
GET	/api/cmis/posts/{postId}/isBookmarkedByStudent/{studentId}	▼
GET	/api/cmis/posts/{id}	▼
DELETE	/api/cmis/posts/{id}	▼
GET	/api/cmis/posts/sortTitleDescending	▼
GET	/api/cmis/posts/sortTitleAscending	▼
GET	/api/cmis/posts/sortLikeNumDescending	▼
GET	/api/cmis/posts/sortDescending	▼
GET	/api/cmis/posts/sortAscending	▼
GET	/api/cmis/posts/search	▼
GET	/api/cmis/posts/searchTitle	▼

<code>GET</code>	/api/cmis/posts/global	
<code>GET</code>	/api/cmis/posts/communities/{communityId}/private	
<code>DELETE</code>	/api/cmis/students/{studentId}/bookmarkedPosts/{postId}	
tag-controller		
<code>GET</code>	/api/cmis/students/{studentId}/interests	
<code>POST</code>	/api/cmis/students/{studentId}/interests	
<code>GET</code>	/api/cmis/communities/{communityId}/tags	
<code>POST</code>	/api/cmis/communities/{communityId}/tags	
<code>GET</code>	/api/cmis/tags	
<code>GET</code>	/api/cmis/tags/{id}	
<code>DELETE</code>	/api/cmis/tags/{id}	
<code>DELETE</code>	/api/cmis/students/{studentId}/interests/{tagId}	
<code>DELETE</code>	/api/cmis/communities/{communityId}/tags/{tagId}	
event-controller		
<code>GET</code>	/api/cmis/students/{id}/events	
<code>POST</code>	/api/cmis/students/{id}/events	
<code>GET</code>	/api/cmis/students/{studentId}/events/{eventId}/isAttended	
<code>GET</code>	/api/cmis/events	
<code>GET</code>	/api/cmis/events/{id}/eventDetails	
<code>GET</code>	/api/cmis/events/{eventId}	
<code>DELETE</code>	/api/cmis/students/{id}/events/{eventId}	
admin-controller		
<code>POST</code>	/api/cmis/admin/{id}/unverifiedCommunities/{userId}/decline	
<code>POST</code>	/api/cmis/admin/{id}/unverifiedCommunities/{userId}/accept	
<code>POST</code>	/api/cmis/admin/createCommunity	
<code>GET</code>	/api/cmis/admin	
<code>GET</code>	/api/cmis/admin/{id}/unverifiedCommunities	
auth-controller		
<code>POST</code>	/api/auth/signup	
<code>POST</code>	/api/auth/signout	
<code>POST</code>	/api/auth/signin	

Backend Test

1- Community Sign Up

Request + Response:

The screenshot shows a Postman request for a POST to `http://localhost:8070/api/auth/signup`. The Body tab contains the following JSON payload:

```
1 {  
2   "firstName": "Ayva Toplulu\u0111u",  
3   "email": "a.top@gtu.edu.tr",  
4   "password": "12345678",  
5   "role": ["community"]  
6 }
```

The response body is:

```
1 {  
2   "message": "User registered successfully!"  
3 }
```

Database:

	user_id	banner	follower_count	github	image	info	instagram	linkedin	member_count	name	role	twitter	type
▶	2	BLOB	0	NULL	BLOB	NULL	NULL	NULL	0	Ayva Top...	unverified	NULL	community
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2- Admin Accepts Community

Request + Response:

POST <http://localhost:8070/api/cmis/admin/1/unverifiedCommunities/2/accept>

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 2,
3    "firstName": "Ayva Toplulu\u011f",
4    "lastName": null,
5    "username": "a.top@gtu.edu.tr",
6    "email": "a.top@gtu.edu.tr",
7    "password": "$2a$10$tB9u.2kx42bdp5qQ8VSHHuUB0t06YBNkHpmKir9vFLSjl3PXglXv0",
8    "roles": [
9      {
10        "id": 3,
11        "name": "ROLE_COMMUNITY"
12      }
13    ]
14  }

```

Database:

user_id	banner	follower_count	github	image	info	instagram	linkedin	member_count	name	role	twitter	type
2	BLOB	0	HULL	BLOB	HULL	HULL	HULL	0	Ayva Top...	community	HULL	community
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3- Student Sign Up:

POST <http://localhost:8070/api/auth/signup>

Params Authorization Headers (9) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

Pretty Raw Preview Visualize JSON

```

1  {
2    "firstName": "Elma",
3    "lastName": "Tekne",
4    "email": "e.tekne22021@gtu.edu.tr",
5    "password": "elmatekne1234",
6    "role": ["student"]

```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "User registered successfully!"
3  }

```

Database:

Result Grid							Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	□
	id	email	first_name	last_name	password	username	community_id				
1	1	admin@gtu.edu.tr	ADMIN	NULL	\$2a\$10\$vUmhMmJQDzbUaSukh2YJ.5/f16kW....	admin@g...	NULL				
2	2	a.top@gtu.edu.tr	Ayva Topluluğ	NULL	\$2a\$10\$tB9u.2kx42bdp5qQ8VSHHuUB0t06YBN...	a.top@g...	NULL				
3	3	e.tekne22021@gtu.edu.tr	Elma	Tekne	\$2a\$10\$8dbdEcRhu6UhIxI9RZo3eDL1FXXR.xIf...	e.tekne2...	NULL				
	NULL	NULL	NULL	NULL	NULL	NULL	NULL				

4- Student Follows Community

POST ▼ http://localhost:8070/api/cmis/students/3/followingCommunities

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON ▾

```
1 [{"id": 2}]

2 [{"id": 2}]

3 ]
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "id": 2,
3   "user": {
4     "id": 2,
5     "firstName": "Ayva Topluluğ",
6     "lastName": null,
7     "username": "a.top@gtu.edu.tr",
8     "email": "a.top@gtu.edu.tr",
9     "password": "$2a$10$tB9u.2kx42bdp5qQ8VSHHuUB0t06YBNkHpmKir9vFLSj13PXglXv0",
10    "roles": [
11      {
12        "id": 3,
13        "name": "ROLE_COMMUNITY"
14      }
15    ]
16  },
17 }
```

Result Grid Filter Rows:

	user_id	banner	follower_count
▶	2	BLOB	1
*	NULL	NULL	NULL

5- Get All Communities Followed

GET <http://localhost:8070/api/cmis/students/3/followingCommunities>

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have any body content.

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 [  
2   {  
3     "id": 2,  
4     "user": {  
5       "id": 2,  
6       "firstName": "Ayva Toplulu\u0111\u011f",  
7       "lastName": null,  
8       "username": "a.top@gtu.edu.tr",  
9       "email": "a.top@gtu.edu.tr",  
10      "password": "$2a$10$tB9u.2kx42bdp5qQ8VSHuUB0t06YBNkHpmKir9vFLSjl3PXg1Xv0",  
11      "roles": [  
12        {  
13          "id": 3,  
14          "name": "ROLE_COMMUNITY"  
15        }  
16      ],  
17    },  
18    "info": null,  
19    "type": "community",  
20  }]
```

6- Apply Community

POST <http://localhost:8070/api/cmis/communities/2/apply/3>

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL **JSON**

```
1 "please accept me!"
```

Community Views Applications:

GET http://localhost:8070/api/cmis/communities/2/memberApplications

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "student": {
5       "id": 1,
6       "name": "John Doe",
7       "email": "john.doe@example.com"
8     },
9     "community": {
10       "id": 1,
11       "name": "Mathematics Department"
12     },
13     "message": "\"please accept me!\""
14   }
15 ]
```

7- Community Accepts Application

PUT http://localhost:8070/api/cmis/communities/2/memberApplications/3/accept

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1
```

```
1 {
2   "id": 3,
3   "user": {
4     "id": 3,
5     "name": "Jane Smith",
6     "email": "jane.smith@example.com"
7   },
8   "bookmarkedPosts": []
9 }
```

Community members GET:

GET <http://localhost:8070/api/cmis/communities/2/members>

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

1

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 [
2   {
3     "id": 1,
4     >     "student": { ...
5       },
6     "community": { ...
7       },
8     "authorizations": [
9       "NONE"
10      ]
11    }
12  ]
13 ]
```

8- Community Shares Post

POST <http://localhost:8070/api/cmis/communities/2/posts>

Params Authorization Headers (9) **Body** Pre-request Script Te

none form-data x-www-form-urlencoded raw binary

```
1 {  
2   "title": "announcement2",  
3   "text": "text2",  
4   "visibility": "global",  
5   "date": {  
6     "year": 2023,  
7     "month": 1,  
8     "day": 11,  
9     "hour": 1,  
10    "minute": 11  
11  }  
12 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "id": 1,  
3   "title": "announcement2",  
4   "text": "text2",  
5   "visibility": "global",  
6   "likeNum": 0,  
7   "event": [],  
8   "date": {  
9     "id": 1
```

GET Community Posts:

GET http://localhost:8070/api/cmis/communities/2/posts

Params Authorization Headers (7) Body Pre-request Script Tests Setting

none form-data x-www-form-urlencoded raw binary GraphQL

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 [ {  
2   "id": 1,  
3   "title": "announcement2",  
4   "text": "text2",  
5   "visibility": "global",  
6   "likeNum": 0,  
7   "event": [],  
8   "date": {  
9     "id": 1,  
10    "year": 2023,  
11    "month": 1,  
12    "day": 11,  
13    "hour": 0,  
14    "minute": 0  
15  },  
16 },  
17 >   "community": { ...  
18 },  
19   "image": null  
20 } ]
```

9- Community Adds An Event

The screenshot shows the Postman application interface. At the top, there is a header bar with 'POST' and a dropdown menu. Below the header, the URL 'http://localhost:8070/api/cmis/communities/2/posts' is entered. Underneath the URL, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body' (which is currently selected), 'Pre-request Script', 'Tests', and 'Settings'. Below these tabs, there are several options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (which is selected), 'binary', 'GraphQL', and 'JSON' (with a dropdown arrow). The main body area contains a JSON-formatted code block with line numbers from 1 to 17. The JSON represents a new event post with various properties like title, text, visibility, date, and an event object containing its own date.

```
1 {
2   "title": "event",
3   "text": "textevent2",
4   "visibility": "global",
5   "date": {
6     "year": 2026,
7     "month": 1,
8     "day": 14
9   },
10  "event": [
11    {
12      "date": {
13        "year": 2031,
14        "month": 2,
15        "day": 23
16      }
17    }
]
```

Get Community Posts:

GET http://localhost:8070/api/cmis/communities/2/posts

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have any body parameters.

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   {
3     "id": 1,
4     "title": "announcement2",
5     "text": "text2",
6     "visibility": "global",
7     "likeNum": 0,
8     "event": [],
9   >   "date": { ...
10    },
11   },
12   "community": { ...
13    },
14   },
15   "image": null
16   },
17   {
18     "id": 3,
19     "title": "event",
20     "text": "textevent2",
21     "visibility": "global",
22     "likeNum": 0,
23     "date": { ...
24      },
25      },
26      },
27      },
28      },
29      },
30      },
31      },
32      },
33      },
34      },
35      },
36      },
37      },
38      },
39      },
40      },
41      },
42      },
43      },
44      },
45      },
46      },
47      },
48      },
49      },
50      },
51      },
52      },
53      },
54      },
```

10- Student Likes a Post

POST <http://localhost:8070/api/cmis/students/3/posts/1/like>

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies (1) Headers (14) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```
1
2   "id": 1,
3   "title": "announcement2",
4   "text": "text2",
5   "visibility": "global",
6   "likeNum": 1,
7   "event": [],
8 >   "date": {...
15 },
16 >   "community": {...
45 },
46   "image": null
47 }
```

POST <http://localhost:8070/api/cmis/students/3/posts/3/like>

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "id": 3,
3   "title": "event",
4   "text": "textevent2",
5   "visibility": "global",
6   "likeNum": 1,
7 >   "event": [...]
20 ],
21 >   "date": {...
28 },
29 >   "community": {...
58 },
59   "image": null
60 }
```

Takes Back The Like:

The screenshot shows the Postman application interface. At the top, there is a header bar with a dropdown menu set to "POST" and a URL field containing "http://localhost:8070/api/cmis/students/3/posts/3/like". Below the header, there are tabs for "Params", "Authorization", "Headers (8)", "Body", "Pre-request Script", "Tests", and "Settings". The "Params" tab is currently active, showing a table titled "Query Params" with one row: "Key" and "Value".

Below the tabs, there are more tabs: "Body", "Cookies (1)", "Headers (14)", and "Test Results". The "Body" tab is selected and highlighted in orange.

Under the "Body" tab, there are four buttons: "Pretty", "Raw", "Preview", and "Visualize". The "Pretty" button is selected and highlighted in grey. To the right of these buttons is a "JSON" dropdown menu with a "Pretty" option selected. Further to the right is a copy icon.

The main content area displays a JSON response with line numbers on the left. The JSON structure is as follows:

```
1 {
2   "id": 3,
3   "title": "event",
4   "text": "textevent2",
5   "visibility": "global",
6   "likeNum": 0,
7   "event": [...]
8 },
9   "date": {...},
10  "community": {...},
11  "image": null
12 }
```

11- Student Attends An Event

POST <http://localhost:8070/api/cmis/students/3/events>

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {
2   "id": 2
3 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "id": 2,
3   "date": {
4     "id": 2,
5     "year": 2031,
6     "month": 2,
7     "day": 23,
8     "hour": 0,
9     "minute": 0
10   },
11   "attendantsNum": 1
12 }
```

12- Student Adds a Project Idea

POST <http://localhost:8070/api/cmis/students/3/projectidea>

Params Authorization Headers (9) **Body** Pre-request Script Tests Set

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {  
2   .... "title" : "Brilliant Idea",  
3   .... "text" : "Brilliant Text"  
4 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "id": 1,  
3   "title": "Brilliant Idea",  
4   "text": "Brilliant Text",  
5   "image": null,  
6   > "student": { ...  
85 },  
86   "likeNum": 0  
87 }
```

13- Get All Tags

The screenshot shows the Postman application interface. At the top, a header bar displays "GET" and the URL "http://localhost:8070/api/cmis/tags". Below this, a navigation bar includes tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", and "Set". The "Params" tab is currently selected. Under "Query Params", there is a table with columns "KEY" and "VAL", which is currently empty. Below the navigation bar, another set of tabs shows "Body", "Cookies (1)", "Headers (14)", and "Test Results", with "Body" being the active tab. The "Body" section contains a "Pretty" tab, a "Raw" tab, a "Preview" tab, a "Visualize" tab, and a "JSON" dropdown menu set to "JSON". To the right of the JSON dropdown is a copy icon. The main content area displays a JSON response with line numbers on the left:

```
1  [
2   {
3     "id": 1,
4     "tag": "TAG_ART"
5   },
6   {
7     "id": 2,
8     "tag": "TAG_BIOLOGY"
9   },
10  {
11    "id": 3,
12    "tag": "TAG_DRONE"
13  },
14  {
15    "id": 4,
16    "tag": "TAG_CHEMISTRY"
17  },
18  {
19    "id": 5,
20    "tag": "TAG_COMPUTER"
21  },
22  {
23    "id": 6,
```

Add Interest To Student:

POST <http://localhost:8070/api/cmis/students/3/interests>

Params Authorization Headers (9) **Body** Pre-request Script Tests ⚙

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {  
2   "id": 1  
3 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "id": 1,  
3   "tag": "TAG_ART"  
4 }
```

14- Another Student Likes the Project Idea:

The screenshot shows two API requests in the Postman application.

Request 1: Sign-In

- Method: POST
- URL: <http://localhost:8070/api/auth/signin>
- Body tab selected
- Body type: raw (JSON)
- JSON payload:

```
1 {
2   "email": "ahmet@gtu.edu.tr",
3   "password": "123ahmet"
4 }
```

Request 2: Like Project Idea

- Method: POST
- URL: <http://localhost:8070/api/cmis/students/4/projectidea/1/like>
- Params tab selected
- Query Params table:

KEY	VALUE
Key	Value
- Body tab selected
- Body type: raw (JSON)
- JSON payload:

```
1 {
2   "id": 1,
3   "title": "Brilliant Idea",
4   "text": "Brilliant Text",
5   "image": null,
6   "student": { ... },
7   "likeNum": 1
8 }
```

15- Student Bookmarks a Project Idea:

A screenshot of the Postman application interface. The request method is set to POST, and the URL is `http://localhost:8070/api/cmis/students/4/bookMarkedProjectIdeas`. The 'Body' tab is selected, showing a JSON payload with one item: `{"id": 1}`. The status bar at the bottom right shows "Status: 200 OK".

Get All Bookmarked:

A screenshot of the Postman application interface. The request method is set to GET, and the URL is `http://localhost:8070/api/cmis/students/4/bookMarkedProjectIdeas`. The 'Body' tab is selected, showing a JSON response with one item. The response body is a JSON object with fields: `"id": 1, "title": "Brilliant Idea", "text": "Brilliant Text", "image": null, "student": {"id": 3, "user": {"id": 3, "firstName": "Elma", "lastName": "Tekne", "username": "e.tekne22021@gtu.edu.tr"}}`. The status bar at the bottom right shows "Status: 200 OK".

16- Community Adds Tag:

A screenshot of the Postman application interface. The request method is set to POST, and the URL is `http://localhost:8070/api/cmis/communities/2/tags`. The 'Body' tab is selected, showing a JSON payload with one item: `{"id": 5}`. The status bar at the bottom right shows "Status: 200 OK".

17- Update Community Information

PUT <http://localhost:8070/api/cmis/communities/2>

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   .... "info" : "community info",
3   .... "linkedin" : "www.linkedin.com/profile/123415"
4 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "user": { ...
4     },
5   "info": "community info",
6   "type": "community",
7   "name": "Ayva Toplulu\u0111u",
8   "image": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4QAiRXhpZgAASUkqAAgAAAAABABIBAwABAAAAAAQAZgIwaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLzw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVT...
9 +Cjx40nhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bmM6bwV0YS8iIHg6eG1wdGs9I1hNUCDBb3J1IDQuNC4wLUV4aXYyIj4KIDX...
10 +C1AopPHIk7inF7XN1cm1wdG1vh1Rv7GY6YWJvdX09T1TKTCAsTHhthG5z0nRn7mY9Tmh0dHA61v9ucv5h7G917551h20vrdG1...
```

Result Grid											
user_id	banner	follower_count	github	image	info	instagram	linkedin	member_count	name	role	
2	BLOB	1	NULL	BLOB	commu...	NULL	www.linkedin.com/profile/123415	1	Ayva Top...	communi...	
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

18- Unfollow a community

DELETE <http://localhost:8070/api/cmis/students/3/followingCommunities/2>

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies (1) Headers (12) Test Results

Status: 204 No Content 1

GET http://localhost:8070/api/cmis/students/3/followingCommunities

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

1 []

Result Grid

	user_id	banner	follower_count	github	image	info	instagram	linkedin	member_count	name	role
▶	2	BLOB	0	HULL	BLOB	commu...	HULL	www.linkedin.com/profile/123415	1	Ayva Top...	commu...
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

19- Delete a Post

DELETE http://localhost:8070/api/cmis/posts/3

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Status: 204 No Content

Body Cookies (1) Headers (12) Test Results

GET http://localhost:8070/api/cmis/posts/

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

20- Update Password:

The screenshot shows a POST request in Postman to update a user's password. The URL is `http://localhost:8070/api/cmis/users/3/password`. The request body is a JSON object with two fields: `oldPassword` and `newPassword`.

```
1 {  
2   "oldPassword": "elmatekne1234",  
3   "newPassword": "elmatekne"  
4 }
```

The response body is a JSON object representing the updated user profile, including the user's ID, first name, last name, username, email, password (in hashed form), and roles.

```
1 {  
2   "id": 3,  
3   "firstName": "Elma",  
4   "lastName": "Tekne",  
5   "username": "e.tekne22021@gtu.edu.tr",  
6   "email": "e.tekne22021@gtu.edu.tr",  
7   "password": "$2a$10$luDx0spr.v.R.mSENEoYf.MdXHgTIkRluFATPNV001Yb6cim3YhMe",  
8   "roles": [  
9     {  
10       "id": 1,  
11       "name": "ROLE_STUDENT"  
12     }  
13   ]  
14 }
```

POST http://localhost:8070/api/auth/signin

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {  
2   "email": "e.tekne22021@gtu.edu.tr",  
3   "password": "elmatekne1234"  
4 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "path": "/api/auth/signin",  
3   "error": "Unauthorized",  
4   "message": "Bad credentials",  
5   "status": 401  
6 }
```

POST http://localhost:8070/api/auth/signin

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```
1 {  
2   "email": "e.tekne22021@gtu.edu.tr",  
3   "password": "elmatekne"  
4 }
```

Body Cookies (1) Headers (15) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "id": 3,  
3   "firstName": "Elma",  
4   "lastName": "Tekne",  
5   "email": "e.tekne22021@gtu.edu.tr",  
6   "roles": [  
7     "ROLE_STUDENT"  
8   ],  
9   "username": "e.tekne22021@gtu.edu.tr"  
10 }
```


Web Test

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Üyeler Üyelik Başvuruları

ayhan2 ayhanoglu Yetki Ver Üyelikten Çıkar

Şamil Berat Delioğulları Yetkisini Kaldır Üyelikten Çıkar

0 Takipçi

2 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

Üyelikten Çıkar

ayhan1 ayhanoglu

Vazgeç **Üyelikten Çıkar**

Şamil Berat Delioğulları Yetkisini Kaldır Üyelikten Çıkar

ayhan1 ayhanoglu Yetki Ver Üyelikten Çıkar

ayhan2 ayhanoglu2 Yetki Ver Üyelikten Çıkar

0 Takipçi

3 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Üyeler Üyelik Başvuruları

Şamil Berat Delioğulları	Yetkisini Kaldır	Üyelikten Çıkar
ayhan1 ayhanoglu1	Yetki Ver	Üyelikten Çıkar
ayhan2 ayhanoglu2	Yetki Ver	Üyelikten Çıkar

0 Takipçi [Edit](#)

3 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

[Instagram](#) [Twitter](#) [Facebook](#) [LinkedIn](#)

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Üyeler Üyelik Başvuruları

ayhan1 ayhanoglu1	Başvuruyu Görüntüle
ayhan2 ayhanoglu2	Başvuruyu Görüntüle
ayhan0 ayhanoglu0	Başvuruyu Görüntüle

0 Takipçi [Edit](#)

1 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

[Instagram](#) [Twitter](#) [Facebook](#) [LinkedIn](#)

cmis

The screenshot shows a digital platform interface. At the top, there is a navigation bar with icons for home, calendar, users, and settings, along with a user profile picture. Below the navigation bar is a large black banner with the text "GEEK DAY" in a stylized blue font, accompanied by a small 3D cube icon. The main content area has a light gray background. It features a header "GTÜ Bilgisayar Topluluğu". Below the header is a navigation menu with tabs: "Gönderiler" (selected), "Yaklaşan Etkinlikler", and "Yönetim". Under "Gönderiler", there is a sub-menu with "Üyeler" and "Üyelik Başvuruları". A user profile card for "Şamil Berat Delioğulları" is displayed, showing a small profile picture, the name, and two buttons: "Yetkisini Kaldır" and "Üyelikten Çıkar". To the right of the profile card is a sidebar with "Takip Et" (0 Takipçi) and "Üyelikten Çık" (1 Üye). Below these are some placeholder text and social media sharing icons (Instagram, Twitter, Facebook, LinkedIn).

cmis

This screenshot shows the same digital platform interface as the first one, but with different content in the "Gönderiler" section. The "Yeni Gönderi" tab is selected. There are two new posts listed:

- Yetki test**
test
Paz 7 Ocak 2023 23:26 ·
- Test Event**
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex lacus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum

To the right of the posts is a sidebar with "Takip Et" (0 Takipçi), "Üyelikten Çık" (1 Üye), some placeholder text about a user account, and social media sharing icons.

cmis

The screenshot shows a user profile for "GTÜ Bilgisayar Topluluğu". The top navigation bar includes icons for home, calendar, users, and settings, along with a profile picture of a person with glasses. Below the header is a decorative banner with the text "GEXKDAY" and a central logo. The main content area has tabs for "Gönderiler" (selected), "Yaklaşan Etkinlikler", and "Yönetim". A form for granting permissions is displayed, containing fields for "Yetki test" and "test", checkboxes for "Görüntü ekle" and "Etkinlik ekle", a dropdown for "Görünürlik" set to "Genel", and buttons for "Vazgeç" and "Gönder". To the right, there's a sidebar with "Takip Et" (0 Takipçi), "Üyelikten Çık" (1 Üye), and a text block about site statistics. Social media sharing icons for Instagram, Twitter, Facebook, and LinkedIn are also present.

The screenshot shows a modal dialog titled "Yetki Ver" (Grant Permission) for "Şamil Berat Delioğulları". It displays a message stating that the permission will be granted to the same users who already have it. The dialog includes a "Gönder" button, a "Yetki Ver" button (which is highlighted in red), and a "Vazgeç" button. The background of the page is dimmed, and the top navigation bar is visible at the top.

cmis

The screenshot shows a digital platform interface. At the top, there's a navigation bar with icons for home, calendar, users, and settings, along with a user profile picture. Below the navigation is a large banner with the text "GEEK DAY" in a stylized blue font, set against a dark background with a pixelated pattern. In the center of the banner is a circular logo featuring a 3D cube with binary code on its faces. Below the banner, the text "GTÜ Bilgisayar Topluluğu" is displayed. A horizontal menu bar follows, with tabs for "Gönderiler", "Yaklaşan Etkinlikler", and "Yönetim" (selected). Under "Yönetim", there are two sub-tabs: "Üyeler" and "Üyelik Başvuruları". A user profile card for "Şamil Berat Delioğulları" is shown, with buttons for "Yetki Ver" and "Üyelikten Çıkar". To the right, there are sections for "0 Takipçi" and "1 Üye", each with a small edit icon. Below these sections is a text block with placeholder Latin text and social media sharing icons for Instagram, Twitter, Facebook, and LinkedIn.

Gönderiler Yaklaşan Etkinlikler Yönetim

Üyeler Üyelik Başvuruları

Şamil Berat Delioğulları **Yetki Ver** **Üyelikten Çıkar**

0 Takipçi

1 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

A modal window titled "Üyelik Başvurusu" (Membership Application) is displayed. It shows a user profile for "Şamil Berat Delioğulları" and the message "Üyelik başvurusu". Below the message are three buttons: "Vazgeç" (Ignore), "Reddet" (Reject), and "Kabul Et" (Accept). The background of the main interface is dimmed, and the "Yönetim" tab is still visible at the top.

Üyelik Başvurusu

Şamil Berat Delioğulları

"Üyelik başvurusu"

Vazgeç **Reddet** **Kabul Et**

0 Takipçi

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

The screenshot shows a digital platform interface. At the top, there is a navigation bar with icons for home, calendar, users, and settings. Below the navigation bar is a large banner with the text "GEEK DAY" in a stylized blue font, set against a dark background with a pixelated pattern. In the center of the banner is a circular logo featuring a 3D cube. Below the banner, the text "GTÜ Bilgisayar Topluluğu" is displayed. A navigation menu below the banner includes tabs for "Gönderiler", "Yaklaşan Etkinlikler", and "Yönetim". The "Yönetim" tab is currently selected. Under "Yönetim", there are two sub-tabs: "Üyeler" and "Üyelik Başvuruları", with "Üyelik Başvuruları" being the active tab. A user profile card for "Şamil Berat Delioğulları" is shown, along with a purple button labeled "Başvuru Görenüle". To the right of the main content area, there is a sidebar with a "Takipçi" section showing "0 Takipçi" and a "Üye" section showing "0 Üye". Below these sections is a text block with placeholder Latin text: "Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu." At the bottom of the sidebar are social media icons for Instagram, Twitter, Facebook, and LinkedIn.

This screenshot provides a detailed view of a specific post titled "Üyelik Başvurusu". The post content is a large, empty rectangular box. Below the box, there are three buttons: "Vazgeç", "Kaydet", and "Başvuru Uygun Et". The "Başvuru Uygun Et" button is highlighted in red. Below the buttons, there is a text area containing placeholder Latin text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ut nisl, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex locus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum". At the bottom of the post area, there are two buttons: "Katılacağımı Belirt - 1 Katılımcı" and "Çar 10 Ocak 2023 15:30". Below the post area, there is a "Beğen" button with "0 Beğeni" next to it and a "Kaydet" button. To the right of the post, there is a sidebar with a "Takip Et" button, "0 Takipçi", a "Başvuru Düzenle" button, and "0 Üye". Below these buttons is another text block with placeholder Latin text: "Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu." At the bottom of the sidebar are social media icons for Instagram, Twitter, Facebook, and LinkedIn.

cmis

The screenshot shows a digital event platform interface. At the top, there are navigation icons for Home, Calendar, Groups, and Notifications, along with user profile pictures. The main header features a large, stylized 'GEEK DAY' logo with a central cube icon.

GTÜ Bilgisayar Topluluğu

Gönderiler **Yaklaşan Etkinlikler**

Test Event

GTÜ Bilgisayar Topluluğu
Paz 7 Ocak 2023 23:07 ·

Test Event
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex locus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum.

+ Katılacağımı Belirt - 1 Katılımcı **Çar 10 Ocak 2023 15:30**

Beğen · 0 Beğeni Kaydet

Takip Et 0 Takipçi **Başvuru Düzenle** 0 Üye

The screenshot shows the same digital event platform interface as the first one, but with a modal window open over the 'Test Event' post. The modal is titled 'Üyelik başvurusu' (Membership Application) and contains a large empty rectangular area for input.

Üyelik başvurusu

Test Event

GTÜ Bilgisayar Topluluğu
Paz 7 Ocak 2023 23:07 ·

Test Event
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex locus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum.

+ Katılacağımı Belirt - 1 Katılımcı **Çar 10 Ocak 2023 15:30**

Beğen · 0 Beğeni Kaydet

Takip Et 0 Takipçi **Üye Ol** 0 Üye

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

[Gönderiler](#) [Yaklaşan Etkinlikler](#)

Test Event
Paz 7 Ocak 2023 23:07

Test Event
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex lacus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum.

+ Katılacağımı Belirt - 1 Katılımcı Çar 10 Ocak 2023 15:30

[Beğen](#) • 0 Beğeni [Kaydet](#)

Takip Et 0 Takipçi **Üye Ol** 0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

Ocak, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
						1
2	3	4	5	6	7	8
9	10 Test	11	12	13	14	15
16	17 Event 15:30	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Şubat, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
		1	2	3	4	5

Ocak, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Şubat, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
		1	2	3	4	5



GTÜ Bilgisayar Topluluğu

Gönderiler

Yaklaşan Etkinlikler

 GTÜ Bilgisayar Topluluğu
Paz 7 Ocak 2023 23:07 ·

Test Event

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae oucator dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex lacus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nullus, ut porta lectus tellus ac nisi. Phasellus varius ipsum

Katılacağımı Belir! · 1 Katılımcı

Çar 10 Ocak 2023 16:30

Beğen · 0 Beğeni

Kaydet

 GTÜ Bilgisayar Topluluğu
Paz 7 Ocak 2023 23:04 ·

Test Başlığı

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi,

Takip Et · 0 Takipçi

Üye Ol · 0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.



cmis

mangalCom0

Gönderiler Yaklaşan Etkinlikler

mangalCom0
Paz 7 Ocak 2023 23:18 •

test 2
test
+ Katılacağını Belirt • 1 Katılımcı

Beğen • 1 Beğeni Kaydet

mangalCom0
Paz 7 Ocak 2023 23:17 •

test
test
+ Katılacağını Belirt • 1 Katılımcı

Beğen • 0 Beğeni Kaydet

Takip Et 0 Takipçi
Üye Ol 0 Üye

cmis

mangalCom0

Gönderiler Yaklaşan Etkinlikler

mangalCom0
Paz 7 Ocak 2023 23:18 •

test 2
test
+ Katılacağını Belirt • 0 Katılımcı

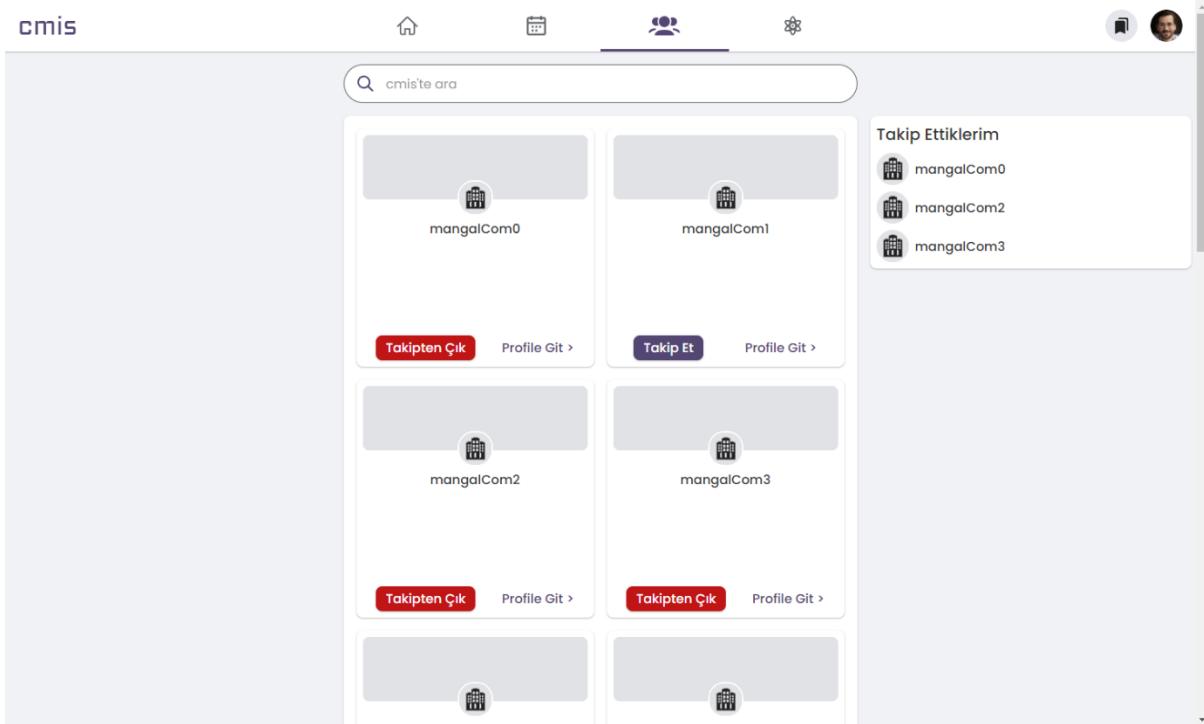
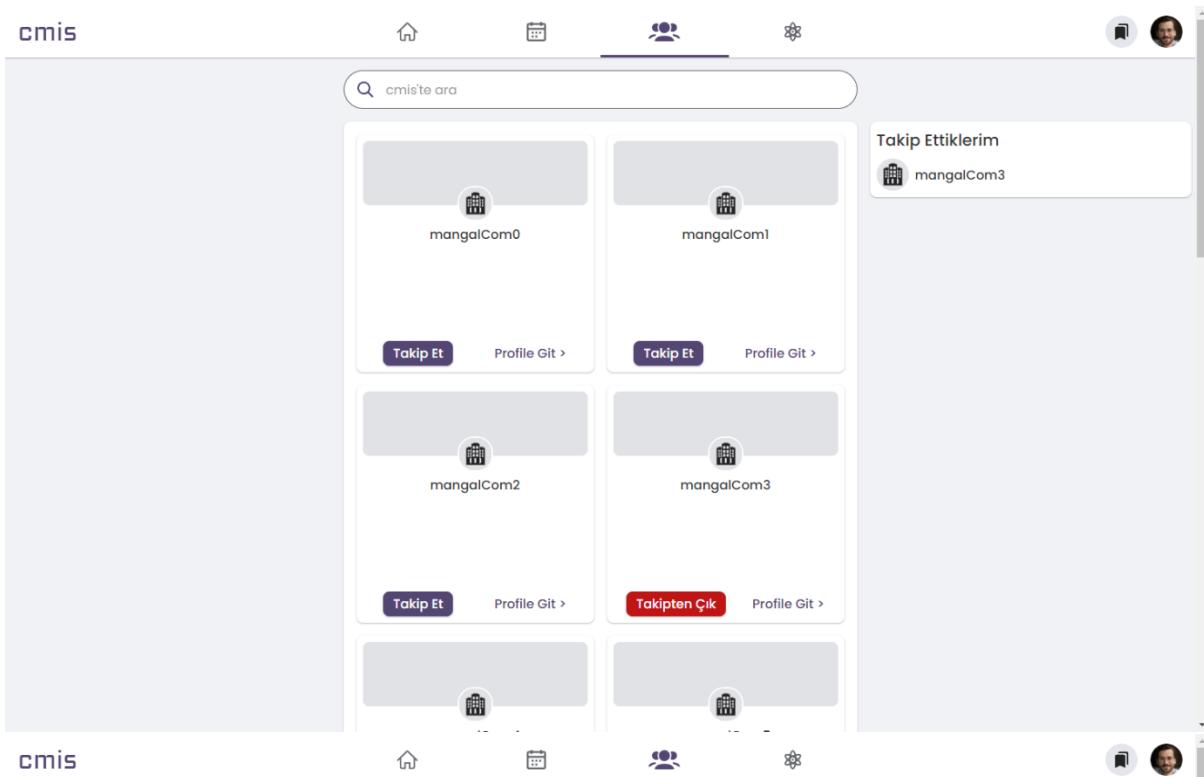
Beğen • 0 Beğeni Kaydet

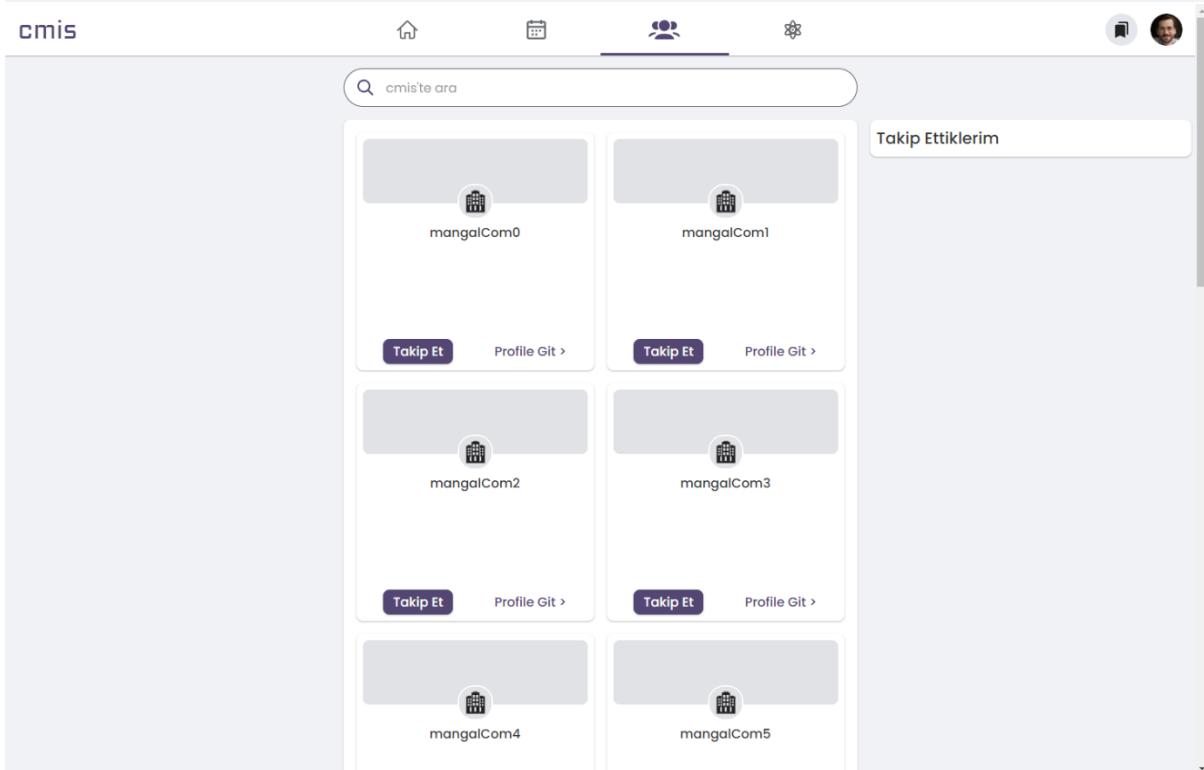
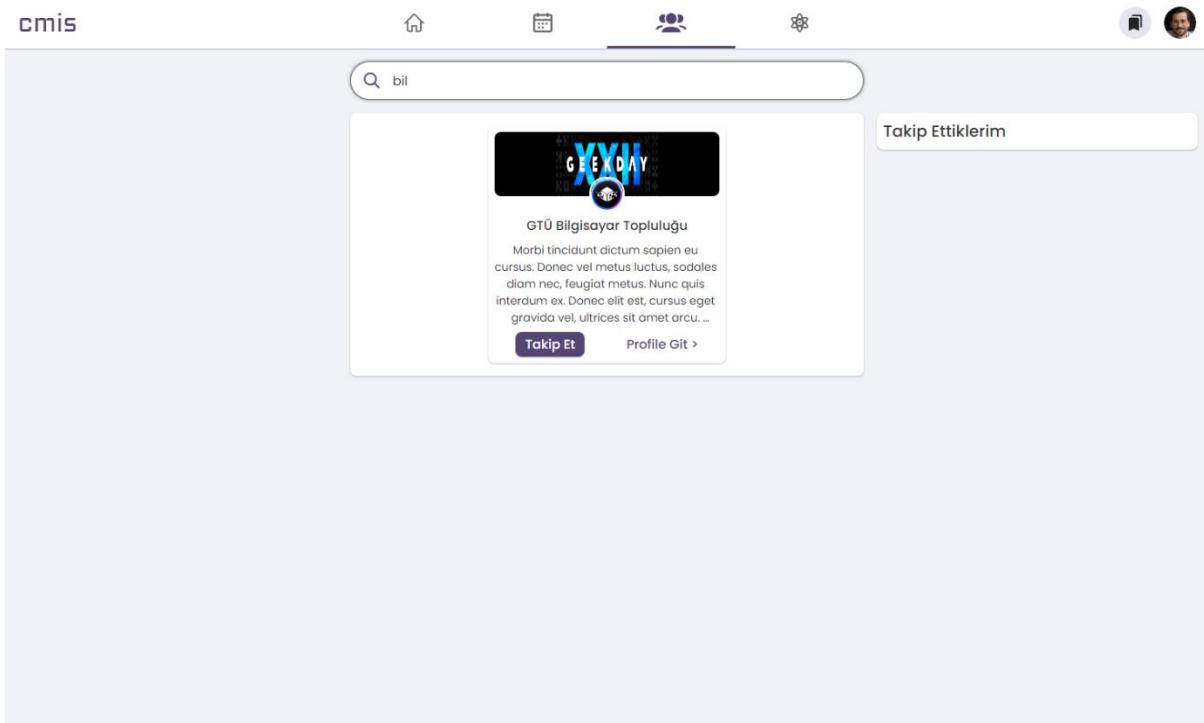
mangalCom0
Paz 7 Ocak 2023 23:17 •

test
test
+ Katılacağını Belirt • 0 Katılımcı

Beğen • 0 Beğeni Kaydet

Takip Et 0 Takipçi
Üye Ol 0 Üye





cmis

The screenshot shows a user profile for 'Şamil Berat Delioğulları'. The profile picture is a circular portrait of a man with a beard. Below the profile picture is the name 'Şamil Berat Delioğulları'. A section titled 'Askıda Projeler' (Recent Projects) contains a single item: 'Yeni Gönderi' (New Post). The post is from 'Şamil Berat Delioğulları' and is titled 'Öğrenci Başlık' (Student Title). The post content is a block of text: 'Vestibulum vitae ultrices lacus, a consectetur sem. Quisque bibendum mattis nisl, bibendum bibendum libero. Donec at ex eu quam sodales porta. Maecenas venenatis dui et lorem iaculis convallis.' Below the text is a list of file attachments with their names and file paths: 'topluluklar/3/gonderiler Tabs_3a7df4c159', 'topluluklar/3/yaklasan-etkinlikler Tabs_3a7df4c159', 'topluluklar/3/yonetim Tabs_3a7df4c159', 'topluluklar/3/gonderiler Tabs_3a7df4c159', 'topluluklar/3/yaklasan-etkinlikler Tabs_3a7df4c159', 'topluluklar/3/yonetim Tabs_3a7df4c159', 'topluluklar/3/gonderiler Tabs_3a7df4c159', 'topluluklar/3/yaklasan-etkinlikler Tabs_3a7df4c159', 'topluluklar/3/yonetim Tabs_3a7df4c159'. There are 0 likes and a 'Kaydet' (Save) button. To the right of the post is a sidebar with a placeholder text: 'Morbi tincidunt dictum sapien eu cursus.' followed by a text editor icon.

cmis

The screenshot shows a user profile for 'Şamil Berat Delioğulları'. The profile picture is a circular portrait of a man with a beard. Below the profile picture is the name 'Şamil Berat Delioğulları'. A section titled 'Askıda Projeler' (Recent Projects) contains a single item: 'Öğrenci Başlık' (Student Title). The post content is a block of text: 'iaculis lobortis at eget velit. Proin eu commodo lacus. Pellentesque eu efficitur quam. Cras a leo non ex dictum congue. Nulla magna ex, rhoncus sit amet viverra ut, tempor vitae nisi. Donec pulvinar tortor a efficitur tincidunt.' Below the text is a file attachment input field with the text 'Görüntü ekle' and a 'Choose File' button. The file path 'ss2.png' is shown. There are two buttons: 'Vazgeç' (Cancel) and 'Gönder' (Send). To the right of the post is a sidebar with a placeholder text: 'Morbi tincidunt dictum sapien eu cursus.' followed by a text editor icon.

cmis

Şamil Berat Delioğulları

Askıda Projeler

Yeni Gönderi

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

Şamil Berat Delioğulları

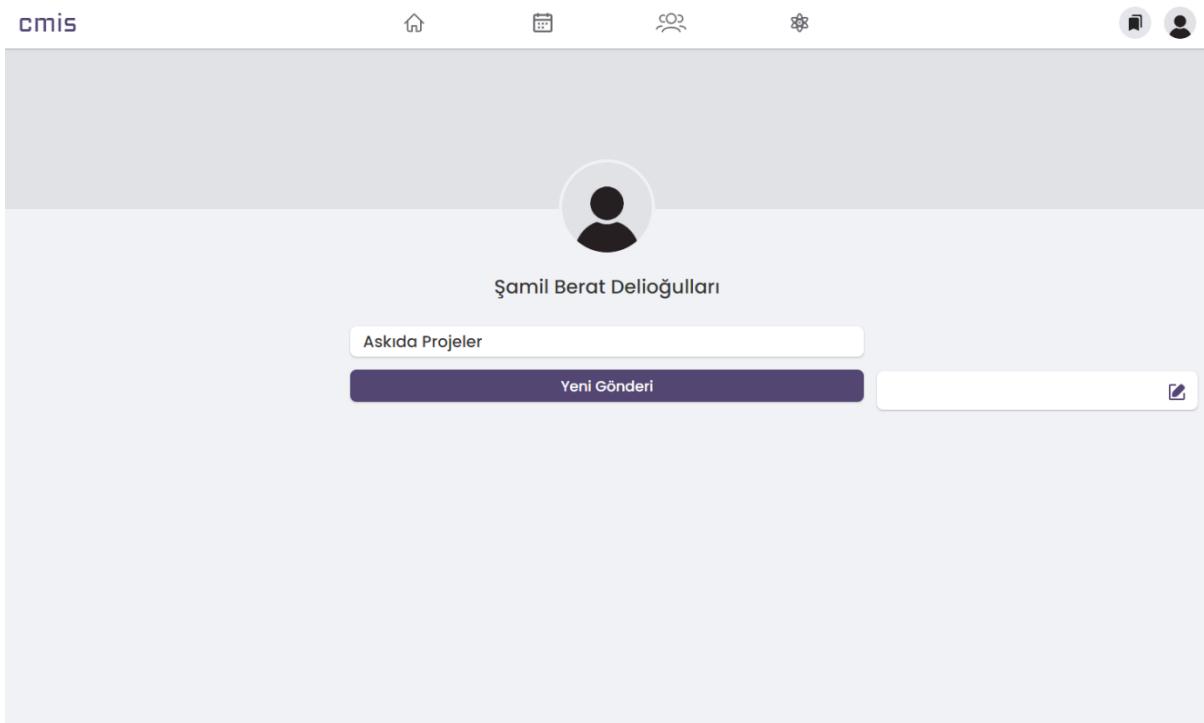
Askıda Projeler

Yeni Gönderi

sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

<https://twitter.com/home>

 <https://www.linkedin.com>



cmis

Şamil Berat Delioğulları

Gönderiler Yaklaşan Etkinlikler Yönetim

Ocak, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
2	3	4	5	6	7	8
9	10 Test	11	12	13	14	15
16	17 Event 15:30	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Şubat, 2023

Pzt	Sal	Çar	Per	Cum	Cmt	Paz

0 Takipçi Edit

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

The screenshot shows a user interface for a social media or event management platform. At the top, there are navigation icons for Home, Calendar, Groups, and Notifications, along with a user profile picture. Below the header, there are three tabs: 'Gönderiler' (Posts), 'Yaklaşan Etkinlikler' (Upcoming Events), and 'Yönetim' (Management). The 'Gönderiler' tab is selected, displaying a list of posts. The first post is titled 'Test Event' and is attributed to 'GTÜ Bilgisayar Topluluğu' (Paz 7 Ocak 2023 23:07). The post content is a long block of placeholder Latin text. It includes a timestamp ('Çar 10 Ocak 2023 16:30') and a 'Beğeni' (Like) button with a count of '0'. To the right of the post, there is a sidebar with '0 Takipçi' (0 Followers) and '0 Üye' (0 Members). A text area with placeholder text follows, and below it are social sharing icons for Instagram, Twitter, Facebook, and LinkedIn.

cmis

This screenshot shows the same web application interface as the previous one, but with the content of the 'Test Event' post highlighted in red. The post title is 'Test Event' and it is from 'GTÜ Bilgisayar Topluluğu' (Paz 7 Ocak 2023 23:04). The highlighted content is a block of placeholder Latin text. Below the post, there are input fields for 'Görüntü ekle' (Upload image), 'Etkinlik ekle' (Add event) with a date and time selector (10/01/2023 15:30), and a dropdown for 'Görünürlük' (Visibility) set to 'Genel'. There are also 'Vazgeç' (Cancel) and 'Gönder' (Send) buttons. To the right of the post, there is a sidebar with '0 Takipçi' (0 Followers), '0 Üye' (0 Members), and a text area with placeholder text, followed by social sharing icons.

cmis

Home Calendar People Notifications User



GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Yeni Gönderi

GTÜ Bilgisayar Topluluğu · Paz 7 Ocak 2023 23:04

Test Başlığı

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex lacus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum sapien, sit amet pulvinar dolor malesuada commodo. Aenean quis pretium nulla. Sed egestas ornare erat, a venenatis nulla ornare eget. Ut malesuada,

0 Beğeni

0 Takipçi

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

Home Calendar People Notifications User



GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Test Başlığı

sodales, turpis quam sagittis nunc, et ornare nunc quam et turpis. Sed ac finibus tellus. Sed eget pellentesque diam. Vestibulum a nibh a libero hendrerit feugiat ut est est. Integer sollicitudin lacus nec lacus sodales aliquet.

Görüntü ekle 02lcg0Rt9G3g...56623239.jpg

Etkinlik ekle

Görünürlik:

0 Takipçi

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

GEEK DAY

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Yeni Gönderi

0 Takipçi

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

0 Takipçi Vazgeç Kaydet

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

<https://www.instagram.com>
 <https://twitter.com/home>
 <https://github.com/los-in>

cmis

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Yeni Gönderi

0 Takipçi Vazgeç Kaydet

0 Üye

Instagram Twitter GitHub LinkedIn

cmis

GTÜ Bilgisayar Topluluğu

Gönderiler Yaklaşan Etkinlikler Yönetim

Başlık

Görüntü ekle
 Etkinlik ekle
Görünürlik:

Vazgeç Gönder

0 Takipçi

0 Üye

cmis

Home Etkinlikler Gönderiler

Gönderiler Yaklaşan Etkinlikler Yönetim

Növ Gönderi

GTÜ Bilgisayar Topluluğu
Paz 7 Ocak 2023 23:04 ·

Test Başlığı

Lore ipsum dolor sit amet, consectetur adipiscing elit. Morbi ut varius nisi, vitae auctor dolor. Curabitur dignissim, urna vel finibus lobortis, lorem velit vehicula quam, et pulvinar dui arcu non mauris. Pellentesque ex lacus, vulputate nec urna at, dictum suscipit dui. Aenean varius odio ac tristique cursus. Duis vel metus sed mauris dignissim eleifend nec sit amet sapien. In eu venenatis lectus. Sed nec purus consectetur, dictum metus in, fermentum sem. Cras tempor, ante at lobortis tristique, sapien metus scelerisque nulla, ut porta lectus tellus ac nisi. Phasellus varius ipsum sapien, sit amet pulvinar dolor malesuada commodo. Aenean quis pretium nulla. Sed egestas ornare erat, a venenatis nulla ornare eget. Ut malesuada, diam eu efficitur sodales, turpis quam sagittis nunc, et ornare nunc quam et turpis. Sed ac finibus tellus. Sed eget pellentesque diam. Vestibulum a nibh a libero hendrerit feugiat ut et est. Integer sollicitudin lacus nec lacus sodales aliquet.



0 Takipçi

0 Üye

Morbi tincidunt dictum sapien eu cursus. Donec vel metus luctus, sodales diam nec, feugiat metus. Nunc quis interdum ex. Donec elit est, cursus eget gravida vel, ultrices sit amet arcu. Aliquam at commodo diam. Nam bibendum eget sapien id lacinia. Praesent sagittis tortor eu magna feugiat, et elementum elit maximus. Cras at aliquam nulla, sit amet scelerisque arcu.

cmis

Home Etkinlikler Gönderiler

Gönderiler Yaklaşan Etkinlikler Yönetim

Növ Gönderi

GTÜ Bilgisayar Topluluğu

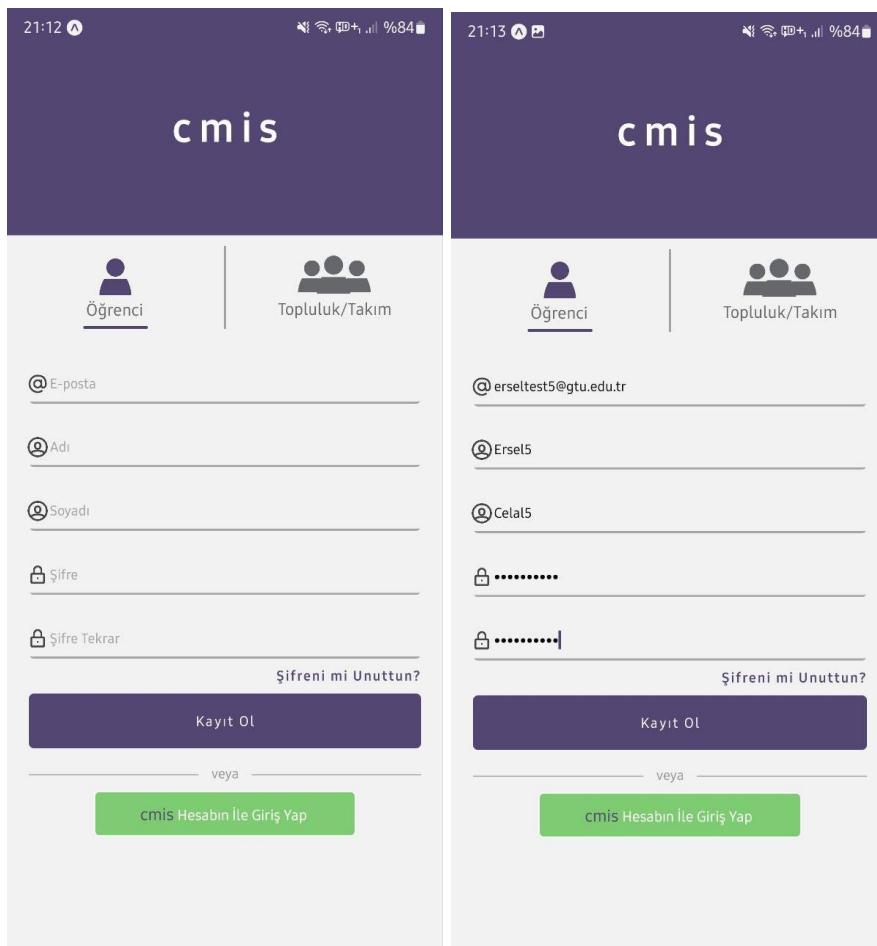


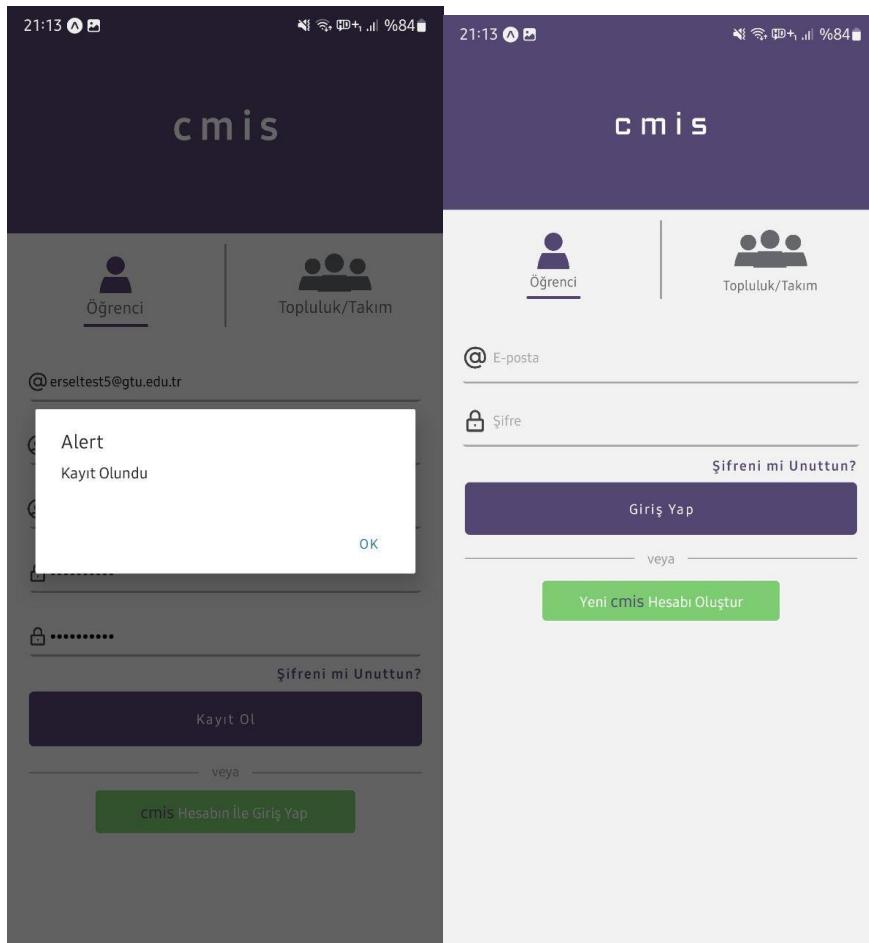
0 Takipçi

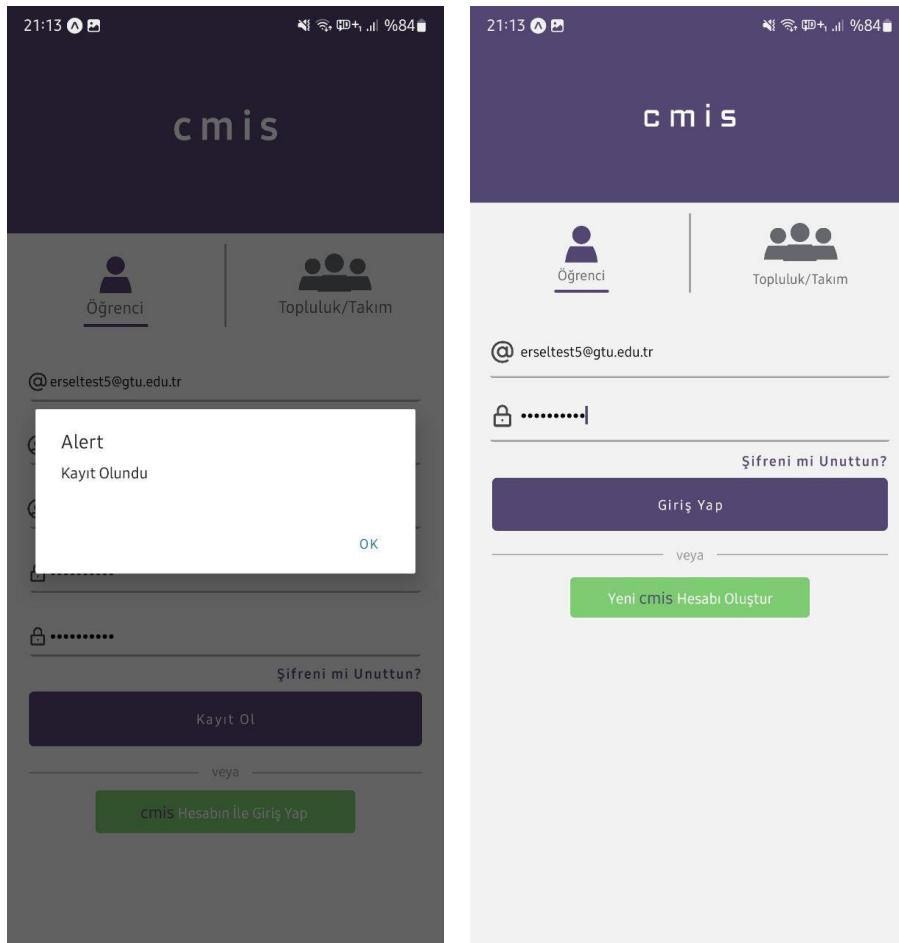
0 Üye

Mobile Test

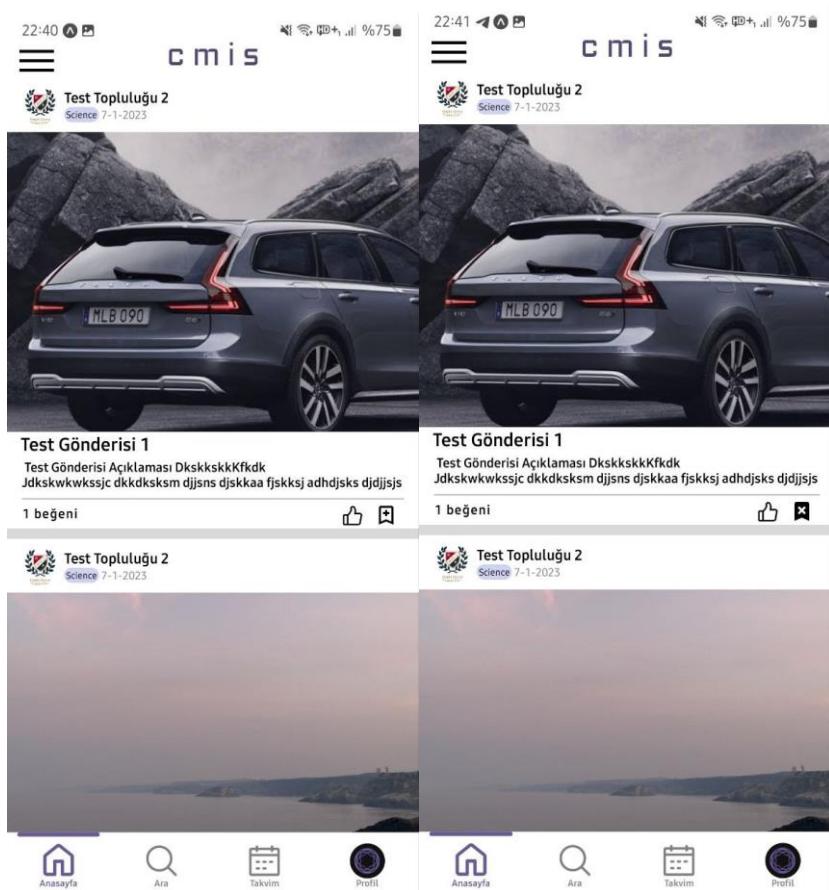
-Register Screen and Login Screen



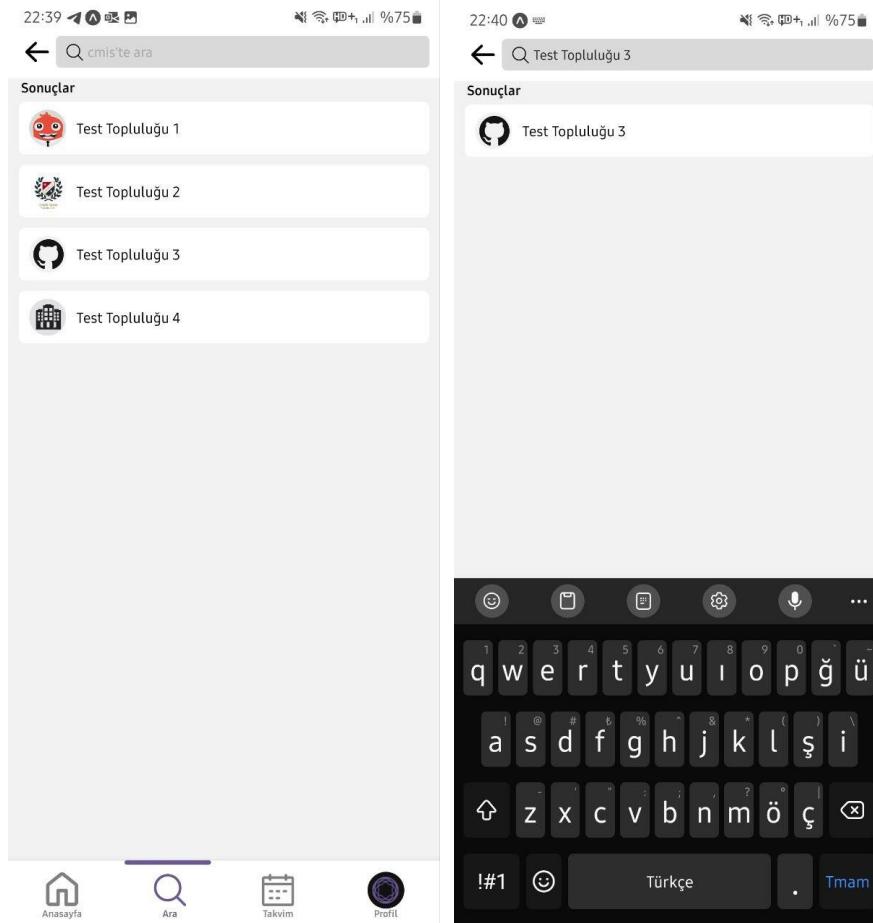




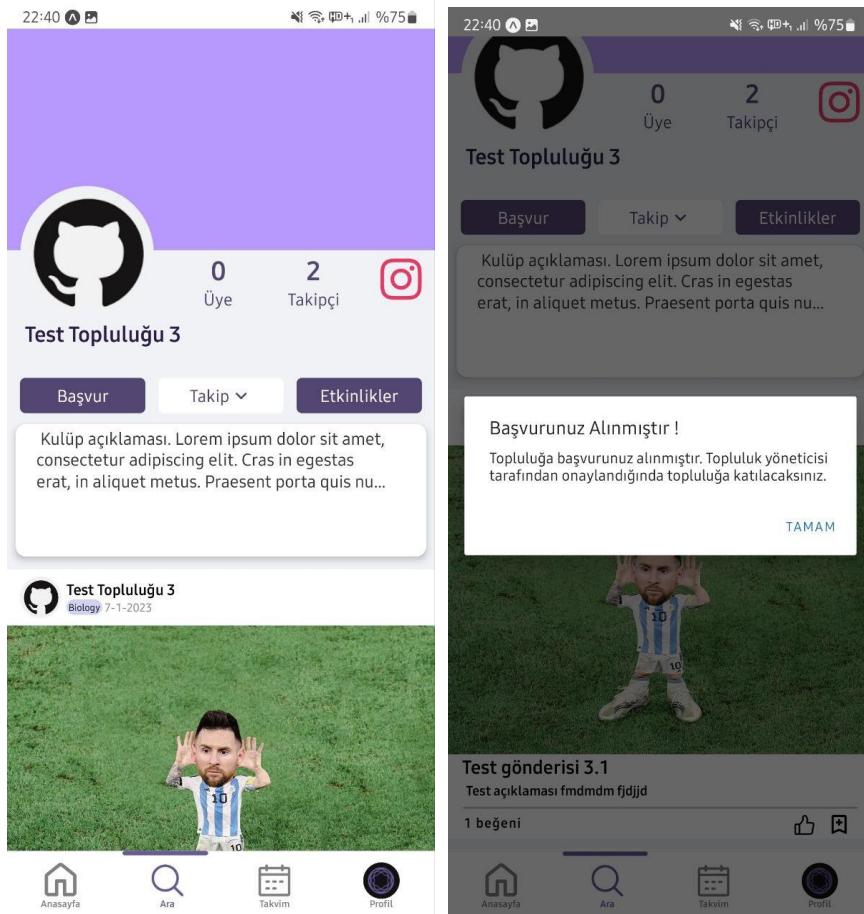
-Home Screen Viewing Posts



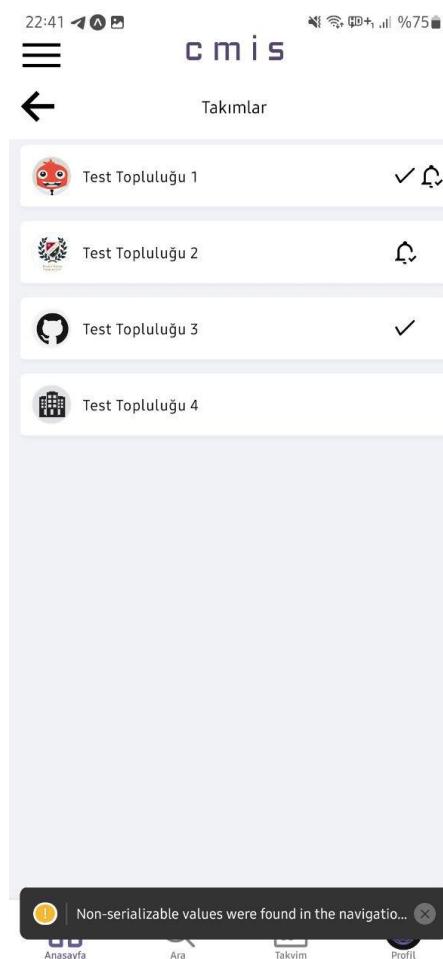
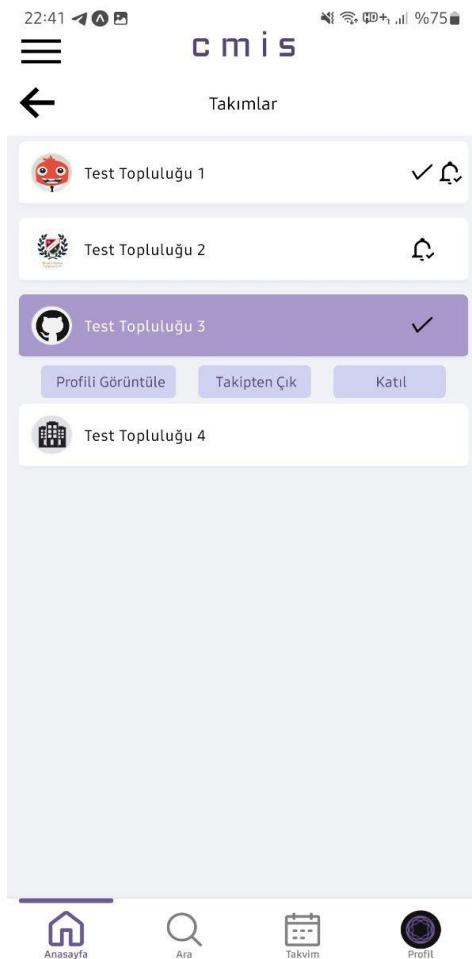
-Search Screen

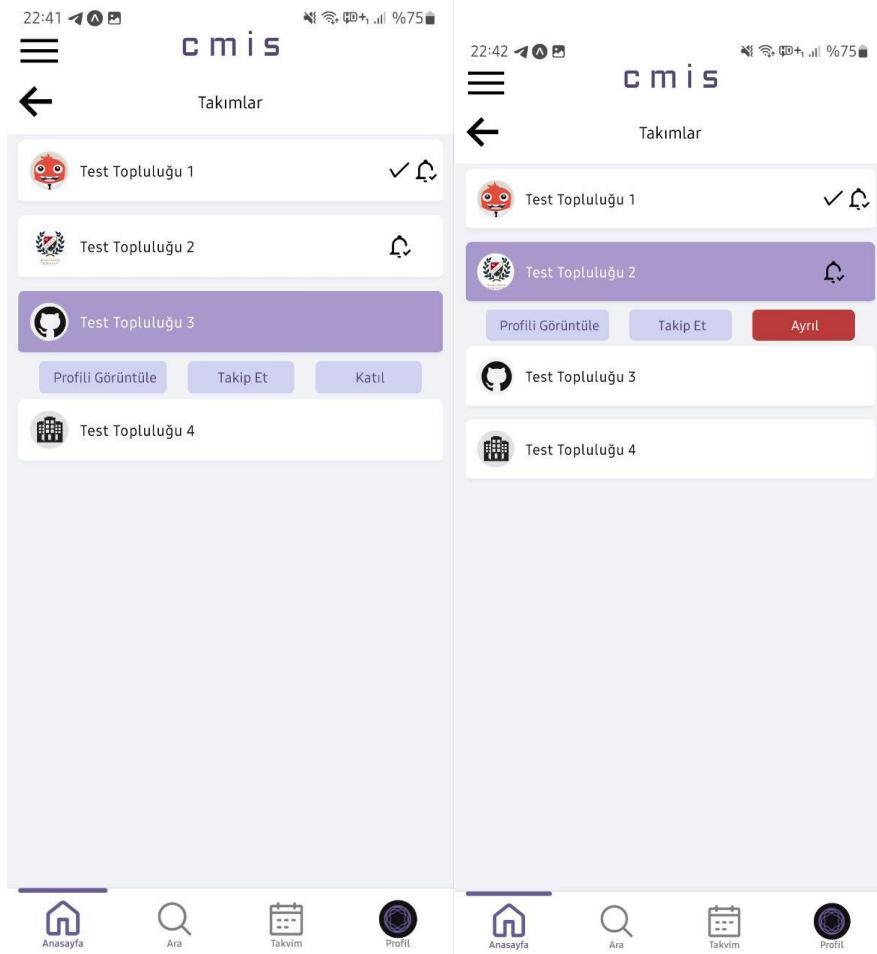


- Visiting a Community Profile and Apply

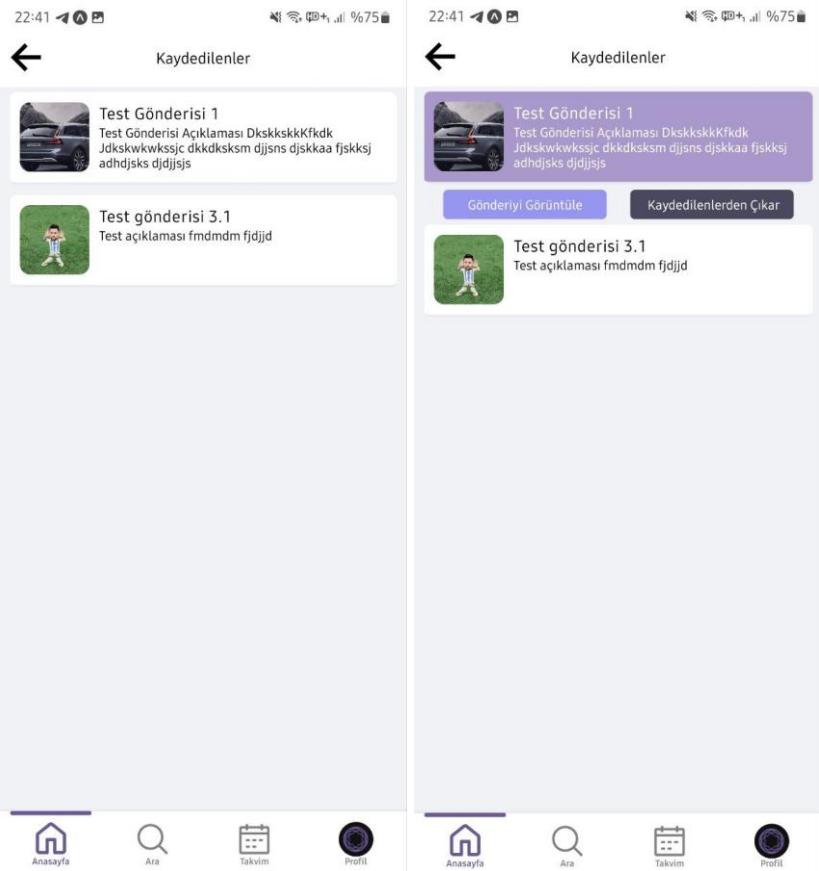


- Listing Teams, Follow, Apply and Visit Profile Buttons

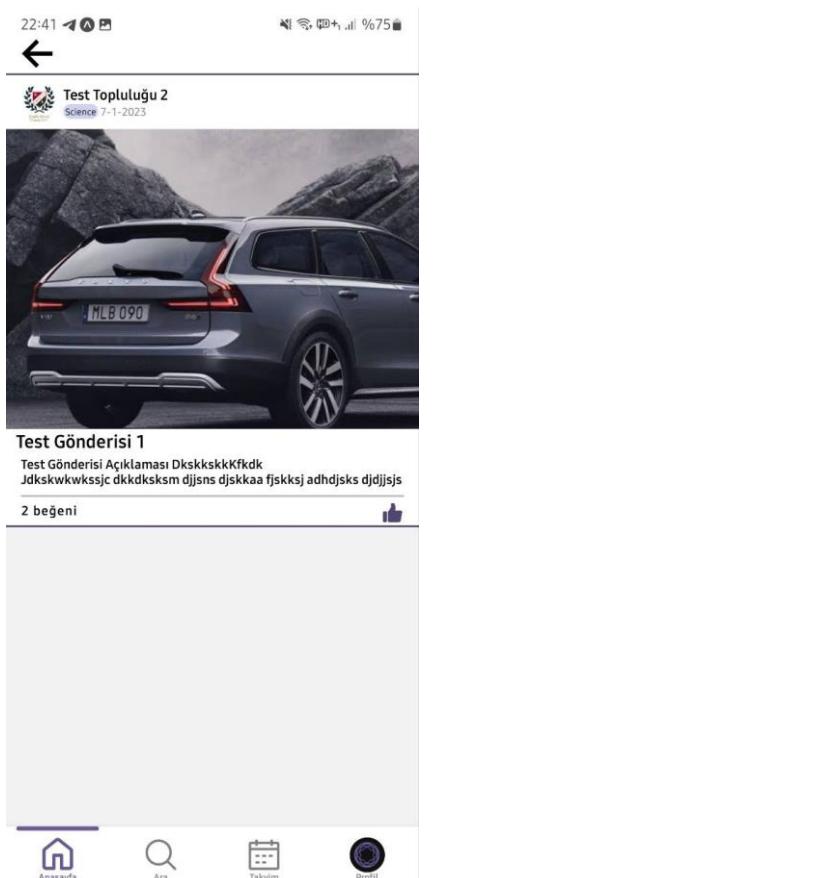




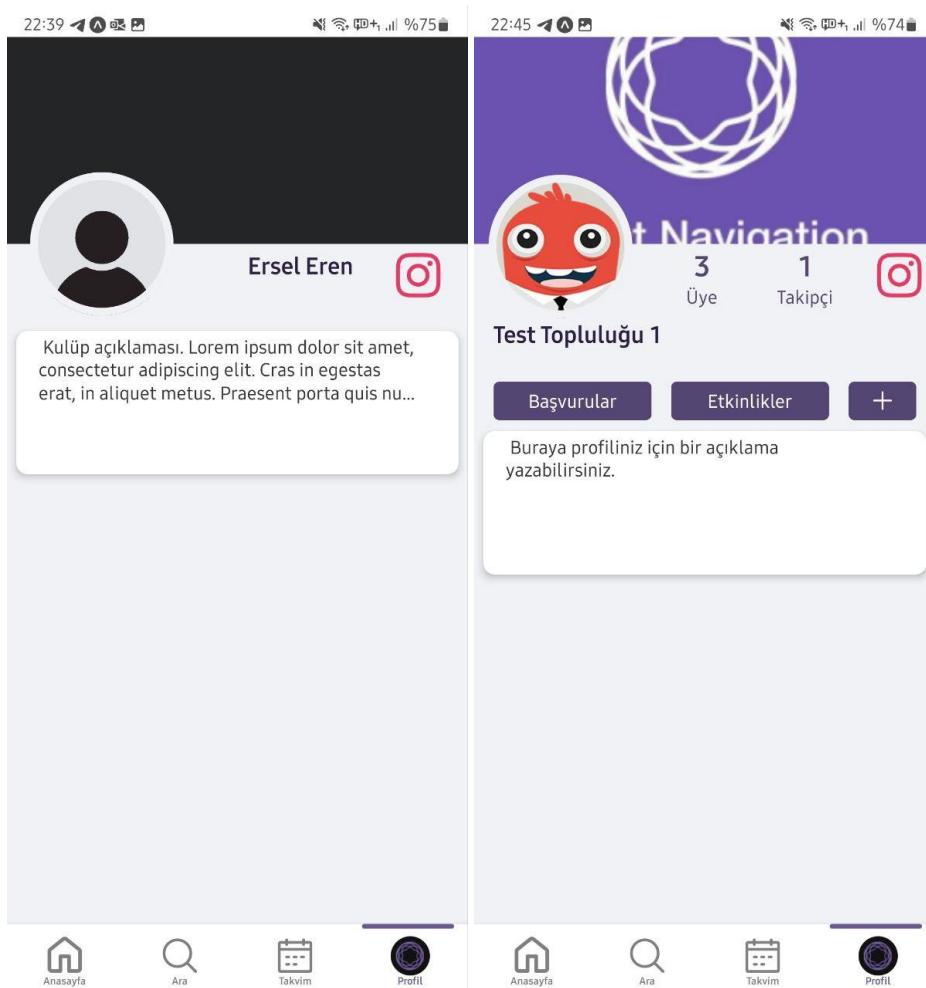
-Bookmarks List, Removing from Bookmarks and Open Post



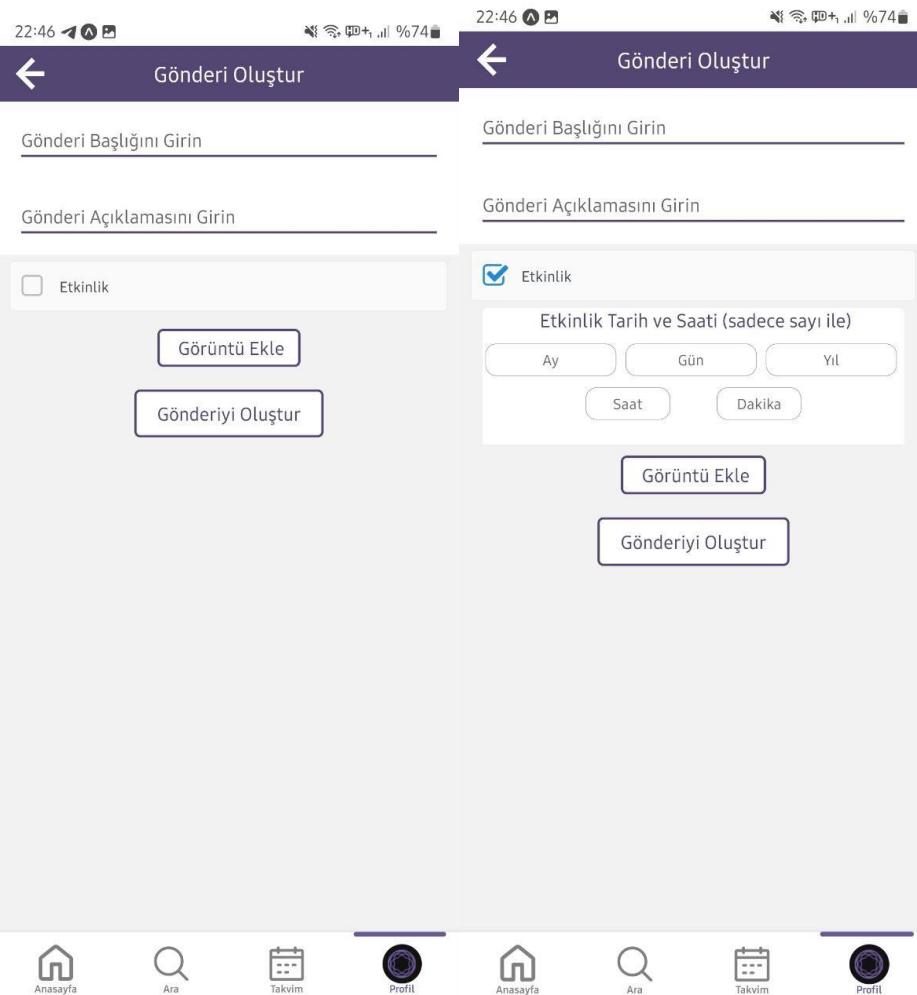
- Bookmarked Post View



-Student and Community Profile



- Creating Post



The image consists of two side-by-side screenshots of a mobile application interface.

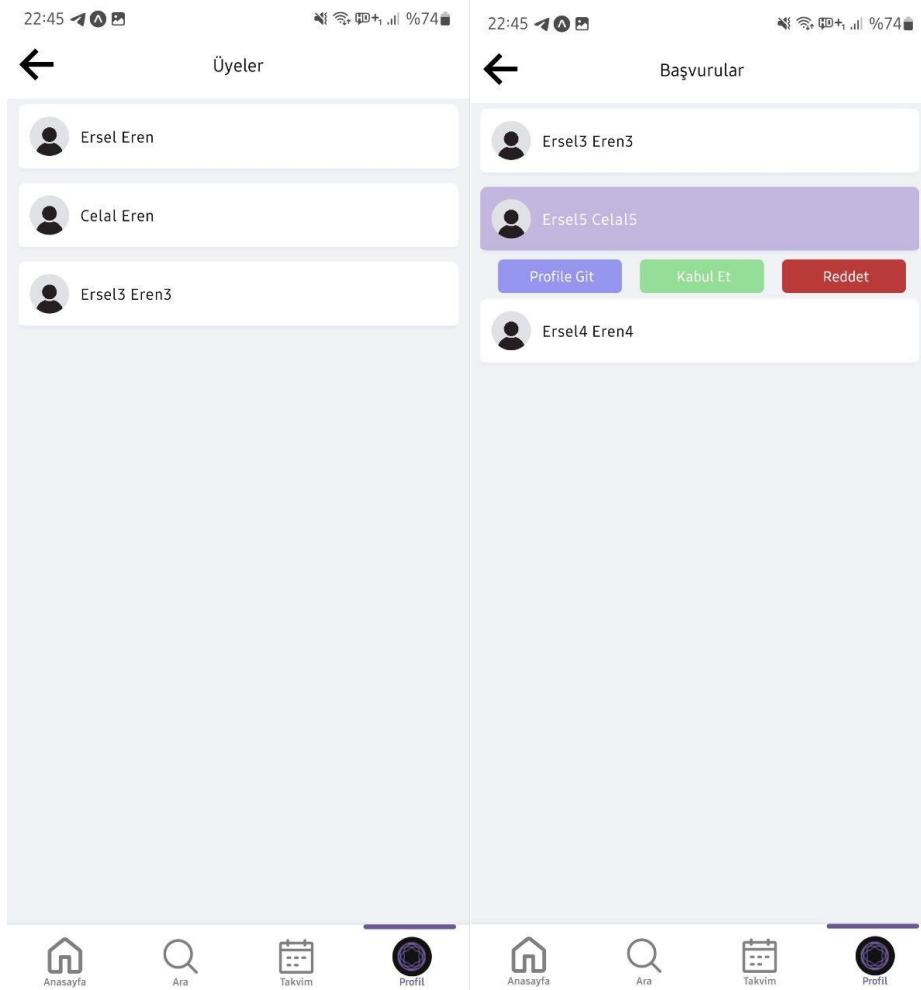
Left Screenshot (Posting Screen):

- Header:** 22:47, battery level 74%.
- Title:** Gönderi Oluştur (Create Post).
- Form Fields:**
 - Gönderi Başlığını Girin (Enter Post Title)
 - Gönderi Açıklamasını Girin (Enter Post Description)
 - Etkinlik (Event) - A checkbox with a checked state.
 - Etkinlik Tarih ve Saati (sadece sayı ile) (Event Date and Time (only numbers))
 - Buttons for Ay (Month), Gün (Day), Yıl (Year), Saat (Hour), and Dakika (Minute).
 - Görüntü Ekle (Add Image) button.
- Image Preview:** An illustration of a muscular, brown-skinned character wearing a yellow vest with a green emblem.
- Post Creation Button:** Gönderiyi Oluştur (Create Post) button.
- Bottom Navigation:** Icons for Anasayfa (Home), Ara (Search), Takvim (Calendar), and Profil (Profile).

Right Screenshot (Community Profile Screen):

- Header:** 22:51, battery level 73%.
- User Info:** Üye (User) and Takipçi (Follower).
- Title:** Test Topluluğu 1 (Test Community 1).
- Buttons:** Başvurular (Applications), Etkinlikler (Events), and a plus sign (+) button.
- Description:** Buraya profiliniz için bir açıklama yazabilirsiniz. (You can write a description for your profile here.)
- Community Profile Card:**
 - Icon: A red emoji-like face.
 - Name: Test Topluluğu 1
 - Category: Compute
 - Date: 7-1-2023
- Image Preview:** The same illustration of the muscular character.
- Event Details:**
 - Etkinlik Test Başlığı 1 (Event Test Title 1)
 - Etkinlik Test Açıklaması. Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use.
 - Date: 24.2.2023 - 21:45
 - + Katıl (Join) button.
- Engagement Metrics:** 0 beğeni (0 likes) and 0 katılım (0 participants).
- Bottom Navigation:** Icons for Anasayfa (Home), Ara (Search), Takvim (Calendar), and Profil (Profile).

- Listing members of community by pressing member count in profile
- Listing Member Applications



- Community profile user listing teams. Can click “Follow” and “Join” buttons

22:47

Wi-Fi 4G+1.11 %74



c m i s



Takımlar



Test Topluluğu 2



Test Topluluğu 3



Test Topluluğu 4

Profilini Görüntüle

Takip Et

Katıl



Anasayfa



Ara

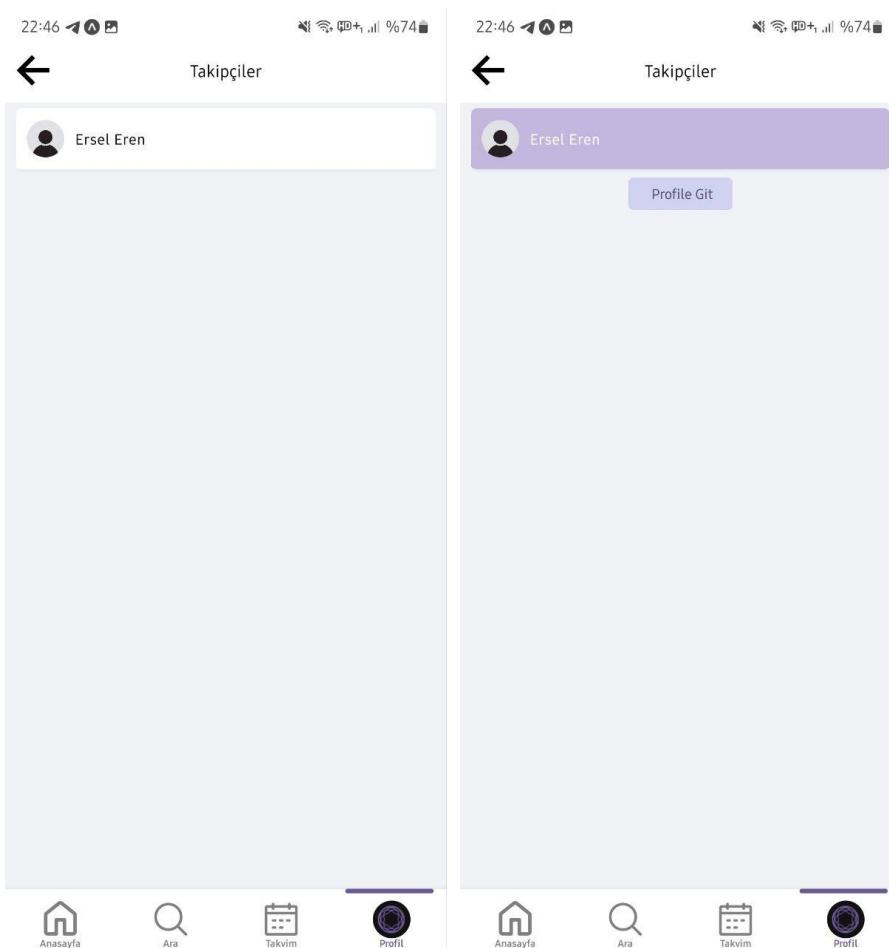


Takvim



Profil

- Community Profile User views its followers and redirects to profile



-Community User Views another

22:47 🔍 74% 22:40 🔍 75%

Test Topluluğu 4

Üye 0 Takipçi 0

Test Topluluğu 3

Üye 0 Takipçi 2

Kulüp açıklaması. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras in egestas erat, in aliquet metus. Praesent porta quis nu...

Başvur Takip Etkinlikler

Kulüp açıklaması. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras in egestas erat, in aliquet metus. Praesent porta quis nu...

Başvur Takip Et Etkinlikler +

Test Topluluğu 3
Biology 7-1-2023

Test gönderisi 3.1

Test açıklaması fmfdmdm fjdjjd

1 beğenisi

Anasayfa Ara Takvim Profil

Project Backlog Document

1- First Sprint

SPRINT: First Two Weeks			Dates: 14.10.2022 – 28.10.2022	
ID	Task	Priority	Assignee	Status
20	Find requirements	Low	Burak Kocausta Ahmet Kadir Aksu	Done
21	Finish module 1 in Meta's Introduction to Front-End Development course.	Medium	Yusuf Arslan	Done
22	Continue learning Spring Boot	High	Ahmet Kadir Aksu Burak Kocausta	Done
23	Create backend repository	Lowest	Burak Kocausta	Done
24	Finish 3rd and 4th weeks of Meta's React Basic course.	Highest	Yusuf Arslan	Done
27	Create the GitHub repository for the web app	Medium	Şamil Berat Delioğulları	Done
28	Begin to learn React Native	Medium	Ersel Celal Eren Muhammed Akif Sevilgen	Done
29	Make hand-on web development practices in Coursera.	Medium	Yusuf Arslan	Done
30	Learn spring security	Medium	Ahmet Kadir Aksu Burak Kocausta	Done
31	Learn rest api more	Medium	Burak Kocausta	Done
33	Finish spring course	Medium	Burak Kocausta	Done
37	Create the GitHub mobil repository	Medium	Muhammed Akif Sevilgen	Done
38	Learn mysql scripts	High	Ahmet Kadir Aksu Burak Kocausta	Done

2- Second Sprint

Sprint: Preparing First Presentation			Dates: 29.10.2022 – 05.11.2022	
ID	Task	Priority	Assignee	Status
46	At least one well-defined user story and its related scenarios	Highest	Burak Kocausta Ahmet Kadir Aksu	Done
47	Project backlog document with task priorities and their rationales	Medium	Ersel Celal Eren Muhammed Akif Sevilgen	Done
48	Use case diagram(s) for the defined scenarios	High	Burak Kocausta Şamil Berat Delioğulları	Done
50	Documented Test Cases or Generated Unit Tests	High	Yusuf Arslan Ahmet Kadir Aksu	Done
51	Preparing presentation file	Lowest	Muhammed Akif Sevilgen	In Progress
52-54	Dummy/Simulated Graphical Web Interface	High	Yusuf Arslan Şamil Berat Delioğulları	Done
53-55	Dummy/Simulated Graphical Mobil Interface	High	Ersel Celal Eren Muhammed Akif Sevilgen	Done

3- Third Sprint

SPRINT: Starting the Development				Dates: 09.11.2022 – 22.11.2022	
ID	Task	Priority	Assignee	Story Points	Status
64	Create Database Table, and Entity for Community	High	Ahmet Kadir Aksu	10	Done
65-68	Generate Login and Registration Page for Mobile	Medium	Ersel Celal Eren	6	In Progress
66-69	Generate Login and Registration Page for Web	Medium	Yusuf Arslan	6	In Progress
67	Creating Login and Registration Page Interface in Figma	Medium	Yusuf Arslan Ersel Celal Eren	4	Done
76	Creating NavBar (header) for Web	Medium	Şamil Berat Delioğulları	4	Done
77	Creating NavBar (header) for Mobil	Medium	Muhammed Akif Sevilgen	4	To Do
78	Creating Community Profile for Web	Medium	Şamil Berat Delioğulları	6	To Do
82	Improving and Developing the Interfaces in Figma	Medium	Yusuf Arslan Ersel Celal Eren	8	Done
85	Create Database Table, and Entity for User	High	Burak Kocausta	10	Done
88	Create Communities and Teams Pages	Medium	Şamil Berat Delioğulları	6	Done

4- Fourth Sprint

SPRINT: Connecting Frontend to Backend					Dates: 24.11.2022 – 08.12.2022	
ID	Task	Priority	Assignee	Story Points	Status	
65-68	Generate Login and Registration Page for Mobile	Medium	Ersel Celal Eren	6	Done	
66-69	Generate Login and Registration Page for Web	Medium	Yusuf Arslan	6	Done	
77	Creating NavBar (header) for Mobil	Medium	Muhammed Akif Sevilgen	4	In Progress	
78	Creating Community Profile for Web	Medium	Şamil Berat Delioğulları	6	In Progress	
91	Implement JWT	Highest	Burak Kocausta	12	Done	
92	Holding images at database	High	Ahmet Kadir Aksu	12	In Progress	
94-95-96	Implementing controllers	Medium	Ahmet Kadir Aksu Burak Kocausta	8	Done	
97	Preparing backlog development	Low	Muhammed Akif Sevilgen	1	Done	
98	Preparing use case diagrams and their detailed tables	High	Burak Kocausta	5	Done	
99	Preparing context model	Medium	Burak Kocausta	2	Done	
100	Preparing process model diagrams	Medium	Muhammed Akif Sevilgen	12	Done	
102	Preparing class diagrams	Medium	Burak Kocausta	1	Done	
104-105-106	Completing test reports	Low	Ahmet Kadir Aksu Ersel Celal Eren Şamil Berat Delioğulları	12	Done	
109-110-111	Preparing sequence diagrams for major state changes or activities defined in use case diagrams	Medium	Şamil Berat Delioğulları Yusuf Arslan Ersel Celal Eren	12	Done	
114	Adding locations to events	Very Low	Yusuf Arslan	1	In Progress	
115	Designing admin panel on Figma	Low	Yusuf Arslan	1	Done	

5- Fifth Sprint

SPRINT: Finishing Phase			Dates: 17.11.2022 – 06.01.2023	
ID	Task	Priority	Assignee	Status
56	Turkish character bug in cloud	medium	Ahmet Kadir Aksu	Done
57	Add Unit Tests -Backend	high	Burak Kocausta	Done
58	Add event application -Backend	medium	Burak Kocausta	Done
59	Add like option -Backend	lowest	Burak Kocausta	Done
60	Update Post Response -Backend	medium	Burak Kocausta	Done
61	Fixing Authentication Bug	high	Burak Kocausta	Done
62	Add Date, visibility Field To Post	medium	Burak Kocausta	Done
63	Implement Apply Community, Accept Member -Backend	high	Ahmet Kadir Aksu	Done
64	Add openapi -Backend	medium	Burak Kocausta	Done
65	Implement e-mail verification - Backend	low	Burak Kocausta	Done
66	Add Tags -Backend	medium	Burak Kocausta	Done
67	Handle Request from Backend for Mobile	medium	Ersel Celal Eren	Done
68	Posts	medium	Ersel Celal Eren	Done
69	Header Bar	medium	Ersel Celal Eren	Done
70	Options Drawer Page for Mobile	medium	Ersel Celal Eren	Done
71	Profile Page for Mobile	medium	Ersel Celal Eren	Done
72	Calendar Page for Mobile	medium	Ersel Celal Eren	Done
73	Search Page for Mobile	medium	Ersel Celal Eren	Done
74	Home Page for Mobil	medium	Ersel Celal Eren	Done
75	Arrange Community, Post get requests -Backend	medium	Ahmet Kadir Aksu	Done
76	Implement Event for Community - Backend	medium	Burak Kocausta	Done
77	Implement Administrator Entity, Controller -Backend	medium	Burak Kocausta	Done
78	Administrator Panel	medium	Yusuf Arslan	Done
79	Project Idea Page for Web	medium	Yusuf Arslan	Done
80	Implement Project Idea -Backend	medium	Ahmet Kadir Aksu	Done
81	Adding Searching Functionality	medium	Şamil Berat Delioğulları	Done
82	Bookmarks Page for Web	medium	Yusuf Arslan	Done
83	Events Page for Web	medium	Şamil Berat Delioğulları	Done
84	User profile page for Web	medium	Şamil Berat Delioğulları	Done
85	Homepage for Web	high	Yusuf Arslan	Done
86	Community Register Application	high	Burak Kocausta	Done
87	Community Member Implementation	high	Ahmet Kadir Aksu	Done
88	Backend API Deployment	high	Ahmet Kadir Aksu	Done