



# CS 319 - Object-Oriented Software Engineering

## CATCH UP

### Project Analysis Report

**Group 1H**

İlke Kaş, 21803184  
Zeynep Büşra Ziyagil, 21802646  
Bilgehan Akcan, 21802901  
Yaren Yılmaz, 21803561  
Ömer Onat Postacı, 21802349

**Instructor:** Eray Tüzün

**Teaching Assistants:** Erdem Tuna, Elgun Jabrayilzade

# **Contents**

<b>Introduction</b>	<b>3</b>
<b>Proposed System</b>	<b>3</b>
Overview	6
Functional Requirements	8
Non-functional Requirements	17
System Models	20
Use-Case Model	20
Scenarios	21
Class Model	35
Dynamic Models	50
User Interface	66
<b>Conclusion</b>	<b>83</b>
<b>References</b>	<b>85</b>

# Analysis Report

*CatchUp: Classroom Helper*

## 1 Introduction

As group 1H, we are designing a classroom helper web-based application named “CatchUp”. Classroom helper applications ensure instructors, teaching assistants and students to easily interact with each other to complete the project of the ongoing semester. As we are the students of Bilkent University, there might be a more understandable application which organizes the semester by organizing the group assignments, group formations, projects, reports, etc.

Undoubtedly, there are many examples of classroom-helper applications with the same features. However, due to the fact that a classroom-helper application is appropriate for object-oriented design, we created the general features of the project as follows:

- Dashboard
- Message
- Announcements
- Profiles
- Calendar in which the students and teaching assistants can see his/her due dates for upcoming project assignments.
- Current grade calculations (average calculations)

- Instructor's timetables for students and groups to make a reservation  
(like office hours but editable)
- Course Pages

Also, the specified features of instructors, students, and teaching assistants are respectively:

#### **For Instructors:**

- Can create course pages,
- Assign artifacts (due date-close assignment),
- Give feedback,
- Grade assignments,
- Example project artifacts,
- Project Description,
- Can create polls,
- Reports for assignments after due dates of homework for example how many groups do the given assignment in time.
- Can see statistics

#### **For Students:**

- Can enroll course pages,
- Can view other groups' pages,
- Their profiles will include peer review rates and comments,
- Default review questions,

- Review other group's work,
- Assess Team Members,
- Due date reminder that will notify each student,
- General message groups,
- Group formation request to teaching assistants,
- All of them are admin in their groups

### **For Groups:**

- Deadline request for extension,
- Notice for students for searching groups. For example, when a group needs one more person with specific requirements, they give ads,
- Add member to the group,
- Remove member from the group,
- Private message between group members

### **For TAs:**

- Can enroll course pages,
- Grading for each assignment,
- Give feedbacks,
- Group formation according to the requests,
- Can delete a group but can't add a person to an existing group

At the implementation stage of the project, we will use HTML5, CSS3, JavaScript and React.js for the frontend developing, whereas, Spring framework of Java, and PostgreSQL for backend developing. We will use AWS as a cloud.

## **2 Overview**

### **2.1. Dashboard**

The dashboard screen will be displayed after the logging in process for every user, while it can be accessed by pressing Catchup logo. This page will include the general course page which they can see other groups and their own group page.

### **2.2. Message**

All users (teaching assistants, instructor, and students) can compose messages to each other. However, there are two types of message platforms. Group chats and direct messages. For example, by group chat there will be a conversation in which all students and teaching assistants enrolled in the course can see the written message. Therefore, students and teaching assistants enrolled in the course can interact with each other. Also, another chat will be created by groups chatting feature when the project groups get created and each project group can interact with each other to design their project. Direct messaging ensures all users can interact with each other privately. None of the other users can see the messages between the two users.

### **2.3. Announcements**

Instructors can make announcements to the general course page. None of the students have permission to change this page. However, instructors will have permission to announce annunciations.

## **2.4. Course Page**

There will be a course page that includes course materials which are respectively: Information about the basic course requirement and link to group pages, etc. The layout of the course page will consist of the materials divided into weeks.

## **2.5. Group Pages**

Groups are formed by teaching assistants according to the request of students. Each group has unique names, participants, reviews and rates given by other groups, uploaded assignments. The grades of these assignments can only be seen by group members. Participants can create group meetings and these meetings can be seen by all members of the group in their calendar. Meetings can be edited by all of the members of the group. Also students can see feedback of the instructors and TAs from their group pages.

## **2.6. Unique Profiles**

Each user has their own unique profiles with distinct features. Student profiles consist of name, surname, department information, group information, student id, mail address, notification, calendar, grade and peer review. Teaching Assistant profiles consist of name, surname, location of the office, notification, project group requests, calendar, id and mail address. Instructor profiles consist of name, surname, course id, timetable, notification, mail address, location of the office. Others can send messages by pressing the send message button.

## **2.7. Calendar**

The calendar will help all of the students and teaching assistants, to keep track of their upcoming events such as group meetings, assignment deadlines, grading deadlines, feedback deadlines, etc. Students will receive notifications one day before the due date through a pop-up. Students can also be able to the one day reminder at their student profile.

## **2.8. Instructor's Timetables**

The instructor's timetable will be visible in the instructors' unique profiles for all the users to be aware of the times that an instructor is available. Both students and teaching assistants can send the message to the instructor after looking at the free hours from this page to make an appointment with the instructor. Instructors can edit this timetable weekly according to his/her appointments in that week if he/she wants.

# **3 Functional Requirements**

There will be functional requirements for each user type (instructor, student, teaching assistant, group).

## **3.1. Instructor**

### **3.1.1. Sign Up**

Instructors will sign up from their sign up part which wants information from the instructor such as name, surname, password and email address.

### **3.1.2. Edit Instructor Profile**

Instructors will be able to change their profile by editing features such as name, surname, password, location of the office and timetables. Besides, they can add profile photos.

### **3.1.3. Create Course Page**

Instructors will create the course page to be enrolled in by students and teaching assistants. As mentioned before, the course page will include information about the basic course requirements, assigned artifacts, link to group pages, announcements etc. After the course page is created, every course page will have unique code for students and teaching assistants to enroll in this course page. Instructors can determine the content of the page.

### **3.1.4. Assign Assignments (Artifacts of The Project)**

Instructors will assign project related reports, source codes, demo dates etc. Also, there will be determined deadlines for students. The deadline, assignment type will be determined by the instructor. Instructors will determine the due date by entering date and time specifically. Instructors can upload assignments with a simple explanation and with or without a file. Additionally, instructors will assign deadlines for teaching assistants to grade assignments.

### **3.1.5. Edit Assignment**

Instructors can edit the assignment features. For example if they want to change the due date or want to extend the deadline they are able to change it through there. The changes will be done in the necessary section of the assignment page.

### **3.1.6. Give Feedback**

Instructors can give feedback to the given assignments. Firstly, the instructor will enter the course page. Then, under the week section that homework's deadline is on, instructors can click assignment and the selected assignment page will be opened. From the assignment page, instructor can see which groups have uploaded their homeworks before and after the deadline in order to indicate late submissions. Then, the instructor can click these group links of the submissions and the system will direct him/her to the chosen group's page assignment part in order to give feedback.

### **3.1.7. Grading Assignment**

Firstly, the instructor will enter the course page. Then, under the week section that homework's deadline is on, instructors can click assignment and the selected assignment page will be opened. From the assignment page, instructors can see which groups have uploaded their homeworks before and after the deadline in order to indicate late submissions. Then, the instructor can click these group links of the submissions and the system will direct him/her to the chosen group's page assignment part in order to grade. Assignments can only be graded if a feedback is given to this assignment earlier.

### **3.1.8. Share Example Project Artifacts**

Instructors can share example project artifacts from previous semesters.

### **3.1.9. Message**

Instructors can receive and send messages to all users privately. It is useful especially for students to ask the instructor about reserving a meeting or office hour after looking at their time tables. They are also able to have group chats.

### **3.1.10. Create Polls**

Instructors can create polls for getting the opinions of the students about the progress of the course. They can create the poll from the course page. When they click “create poll”, the system will display pop-up for creation of the poll. Instructors will enter the question and options for students to choose. Results will be shown to the instructor at the end of the time that is determined by the instructor during the creation phase of the poll. This ensures the course to be more interactive.

### **3.1.11. Report Assignment Statistic**

Instructors can see the statics of the assignments (artifacts of the project) as separate reports for each assignment consisting of the average grade, the number of students who upload the assignment on time, peer reviews. The instructor will be shown a button to see statistics. Every time the instructor clicks the button the report will be generated from scratch. By that, if the instructor makes any change in grades, information will be updated in the report. The report will be shown in the course page of the instructor under the assignment part.

## **3.2. Student**

### **3.2.1. Sign up**

Students will sign up from their sign up section which wants information from the students such as name, surname, password, mail address, student number, department and grade.

### **3.2.2. Edit Student Profile**

Students will be able to change their profile by editing features such as name, surname, password, photos.

### **3.2.3. Enroll Course**

Each student will enroll in the course to know more about course related topics (artifacts of the project), interact with the instructor, teaching assistants or the students that take the same course. However, if they do not have the course code, they can't enroll in the course. After enrolling in a course, students will be able to access the course page. They enroll courses from their dashboards.

### **3.2.4. Upload Assignment**

Students can upload artifacts to the system in order to be graded and given feedback from teaching assistants, students and instructors. Students use the corresponding assignment's uploading place in their group's page and only the group members are authorised to upload an assignment. However, the instructor, teaching assistants and other students can view the submission to give feedback.

### **3.2.5. Send Message**

Students can send messages to all other users. They can either use the general messaging groups which enables them to communicate with students, instructors and teaching assistants, private messaging feature to communicate with someone privately, or the group messaging feature that helps group members to interact with each other. Hereby, the course progress will be more interactive. Also, students can send a message privately to the instructor to arrange an appointment within the available time slots in the instructor's timetable given in his/her profile or ask questions.

### **3.2.6. Request to Form Group**

Undoubtedly, there will be groups for the projects. After students form their groups unofficially, they send a request including each group members' id and name of the group to the teaching assistants to confirm their group formation and make it official.

### **3.2.7. Find Groups for Students**

A student having no group who sees the advertisement of other groups on the course page under the advertisements part can communicate with the group which gives the advertisement via message. In case of agreement of the student and the group, the group is able to add the student to their group. Besides that, students can give advertisements individually if they are in search of a group. Advertisements will be shown in the course page.

### **3.2.8. Assess Team Members (Peer Review)**

Each student can assess their group members. They can assess them by writing comments and giving rates out of 10. They can submit the peer reviews just once. The comments will be visible in the assessed student's page anonymously. Also, the rates given by the group members will be averaged and shown in the student's profile. However, instructors can see the owner of these comments and rates. At the end of each assignment, group members will give each other feedback and grades.

### **3.2.9. Review and Give Feedback to Group's Work**

Students can give feedback and rates to others and their own groups' work to be more interactive. Thereby, they can encourage themselves to develop better artifacts. The works done by each group can be rated out of 10. However, the given points and feedback will be anonymous for students, while visible for the instructor. Only their group name will be visible while grading. They can reach these group pages from the general course page.

### **3.2.10. Submit Polls**

Students will submit polls which are created by their course instructor. The poll will only be able to submit in the time period stated by the instructor. Polls will be shown in the course page. Students will be able to submit the poll just once.

## **3.3. Teaching Assistant**

### **3.3.1. Sign up**

Teaching assistants will sign up from their sign up part which wants information from the TA's such as name, surname, password, mail address.

### **3.3.2. Edit Teaching Assistant's profile**

Teaching assistants will be able to change their profile by editing features such as name, surname, password, location of the office, photos.

### **3.3.3. Enroll Course**

Each teaching assistant will enroll in the course by using the course code created by the instructor. If they do not have course code, they can not enroll in the course. They enroll courses from their dashboards.

### **3.3.4. Grade Assignments and Give Feedback**

Teaching assistants can grade the assignments and give feedback to them (artifacts of the project). Firstly, the teaching assistant will enter the course page. Then, under the week section that homework's deadline is on, instructors can click assignment and the selected assignment page will be opened. From the assignment page, instructor can see which groups have uploaded their homeworks before and after the deadline in order to indicate late submissions. Then, the teaching assistant can click these group links of the submissions and the system will direct him/her to the chosen group's page assignment part in order to grade and give feedback. It is mandatory to give feedback and grade every assignment before the deadline given them by the instructor.

### **3.3.5. Assign Assignments**

Teaching assistants will assign project related reports, demo dates etc. Also, there will be determined deadlines for students. The deadline, assignment type will be determined by the teaching assistants. Teaching assistants will determine the due date

by entering it specifically. Teaching assistants can upload assignments with a simple explanation and with or without a file.

### **3.3.6. Edit Assignment**

Teaching assistants can edit the assignment features. For example if they want to change the due date or want to extend the deadline they are able to change it through there. The changes will be done in the assignment section of the assignment page.

### **3.3.7. Manage Requests**

Teaching assistants will receive requests from students who decide their group members and name. If the group does not exceed the maximum member limit, teaching assistants will confirm the request and create the group page. However, they will not be able to add or remove members from the group page. They are also able to delete the group pages. After accepting the request, the assistant will enter the id of group members and name of the group. After that the group page will be created automatically by the system for participants. It is possible to see these groups from the course page of any users.

## **3.4. Groups**

### **3.4.1. Request Deadline Extension**

Students can make a request in order to extend the due date of an assignment. The statistics of how many students request deadline extensions will be visible on the course page. Groups can request this from their group page.

### **3.4.2. Advertise for Group Member**

In the case of missing group members, students can place an advertisement on the course page to find remaining group members by specifying the qualifications they look for in the potential group member. Thanks to this feature, group formation can be completed in an easier way and a new group member who bridges the specific gap related to the project can be added.

### **3.4.3. Add Member**

In case of agreement with a student to join the group, groups can add the student and the students becomes capable of submitting reports or messaging with the other group members within the group chat window.

### **3.4.4. Remove Member**

When a student wants to leave the group or group members decide not to be a group member with the specific student anymore, the group can remove the student.

## **4 Non Functional Requirements**

### **4.1. Usability**

#### **4.1.1 User-Friendly Interface**

The interface will not be complex, instead it will be simple and provide quick access to essential features and commands. Interface requires minimal explanation to understand how to use it.

#### **4.1.2 Group With No Member**

Teaching assistants cannot create groups with no members. Besides that, if all members of one group leave the group, the system will automatically delete this group.

#### **4.1.3 Enrollment Failure**

When a student enters the wrong course code, or the email of the instructor the system does not allow the student to enroll in the course and show error to the user.

#### **4.1.4 Invalid Password or Email Address**

When a user enters the wrong password or email address, the system will not accept this information and show error to the user.

#### **4.1.5 Time Limitation Of Polls**

There is limitation for the time of the polls and this is determined by the instructor. After time out for polls, students cannot make changes in their choices.

#### **4.1.6 Poll submission once**

Students have only one attempt to submit their poll answers. After submitting they can not change the options they chose.

#### **4.1.7 Peer Review Limitation**

Students can make only one attempt to rate their group members' works. After submitting they can not change the grades or feedback they have given.

## **4.2 Reliability**

### **4.2.1 Exception Handling**

Program will have an exception handling mechanism to prevent errors. It will display error messages if the error is caused by the users.

## **4.3 Performance**

### **4.3.1 Maximum Submission Size**

Files with size 15 MB can be accepted through the submissions of assignments. Files that exceed that limit will give error.

## **4.4 Security**

### **4.4.1 Time out for Accounts**

Time out can occur during the session of the user. In this case, to protect against certain types of computer attacks, the system will close the users' accounts.

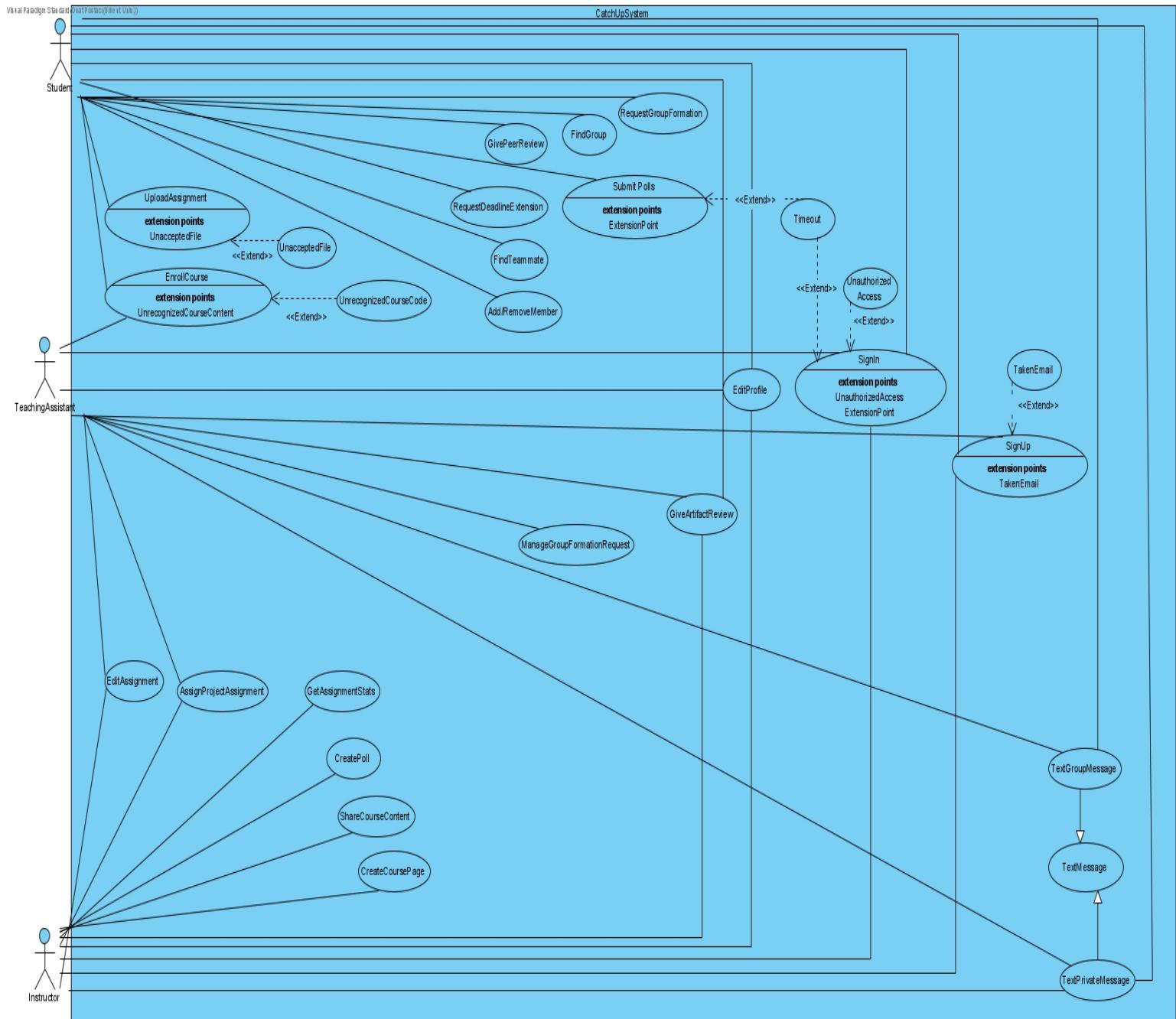
## **4.5. Compatibility**

### **4.5.1 Assignment types (pdf,txt,source codes)**

System will only accept submissions in forms such as pdf, txt or files that contain source codes (such as .l , .htm, .html). Otherwise, the submission files will not be accepted.

# 5 System Models

## 5.1 Use-Case Model



**Figure 1 (Use Case Diagram)**

### **5.1.1 Use-Case #1**

Use Case: Sign Up

Primary Actor(s): Student, Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor wants to register to the system.
- System checks the entered information and registers the actor to the system.

Entry Condition:

- There is no entry condition.

Exit Condition:

- The player is directed to the “Sign In” page.
- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Actor presses the “Sign Up” button.
2. Actor enters his/her name, surname, email, and password to register to the system.
3. If the entered information is verified by the system, then the actor registers to the system.

### **5.1.2 Use-Case #2**

Use Case: Sign In

Primary Actor(s): Student, Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor wants to sign in to the system.

- System checks the entered information and lets the actor into the application.

Entry Condition:

- There must exist an account with entered email and password.

Exit Condition:

- The actor is directed to the “Dashboard” page.

Success Scenario Event Flow:

1. Actor enters an email, and password to enter the system.
2. If the entered information is verified by the system, then the actor enters the system.

### **5.1.3 Use-Case #3**

Use Case: Create Course

Primary Actor(s): Instructor

Stakeholders and Interests:

- Instructor wants to create a course with a course page.

Entry Condition:

- The instructor should be registered to the system.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Instructor will press the “+” button from his/her dashboard.
2. Instructors will enter information such as the course name.

3. System will generate unique course code and courses will be created.

#### **5.1.4 Use-Case #4**

Use Case: Enroll in Course

Primary Actor(s): Student, Teaching Assistant

Stakeholders and Interests:

- Actor wants to enroll in the system.
- System checks the entered information and lets the actor enroll in the course.

Entry Condition:

- There should be a course which was created by the instructor.
- Students or Teaching Assistants should be signed in for this operation.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. The actor (student or teaching assistant) presses the “+” button from the dashboard.
2. The actor (student or teaching assistant) enters the course code.
3. If entered information (course code) is recognized by the system, then the actor can enroll in the course.

#### **5.1.5 Use-Case #5**

Use Case: Assign Project Assignment

Primary Actor(s): Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor wants to assign a new project assignment.

Entry Condition:

- There should be a course which was created by the instructor.
- Actors (teaching assistants or instructors) should be signed in.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. The actor (instructor or teaching assistant) enters the “Course Page” from the dashboard.
2. The actor (instructor or teaching assistant) presses the “+” button then chooses the “Add Assignment” button from the course page.
3. The actor (instructor or teaching assistant) assigns a new project assignment by giving title, description, due date to it. Besides, actors can upload files that describe the assignment.

### **5.1.6 Use-Case #6**

Use Case: Create Polls

Primary Actor(s): Instructor

Stakeholders and Interests:

- Instructor wants to create a new poll.

Entry Condition:

- There should be a course which was created by the instructor.
- Instructors should be signed for this operation.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. The actor (instructor) enters the “Course Page” from the dashboard.
2. The actor (instructor) presses the “Create New Poll” button from the course page and is directed to the poll creation page.
3. The actor can write questions and multiple choices for answers.

Besides, s/he can specify the due date of this poll.

### **5.1.7 Use-Case #7**

Use Case: Edit Assignment

Primary Actor(s): Instructor, Teaching Assistant

Stakeholders and Interests:

- The actor( instructor, teaching assistant) wants to edit the assignment.

Entry Condition:

- The actor should be signed in.
- There should be a course which was created by the instructor.
- There should be an existing assignment assigned by an instructor or teaching assistant.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. The actor (instructor or teaching assistant) presses the “Edit” button from the assignment page and edits its information such as due date, file, description, etc.

Alternative Scenario Event Flow:

1. If the teaching assistant gets notification from groups to extend the deadline and decides to do it, S/he can achieve this by editing the assignment.

### **5.1.8 Use-Case #8**

Use Case: Share Example Project Artifacts

Primary Actor(s): Instructor

Stakeholders and Interests:

- Instructor wants to share example project artifacts to the students and TAs.

Entry Condition:

- There should be a course which was created by the instructor.
- Instructors should be signed in.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Instructor presses the attach button from the course page and shares example project artifacts by uploading documents from his/her desktop .

### **5.1.9 Use-Case #9**

Use Case: Upload Assignment

Primary Actor(s): Student

Stakeholders and Interests:

- Student wants to upload the assignment assigned for him/her.

Entry Condition:

- Students should be signed in.
- There should be a course which was created by the instructor.
- There should be an existing assignment assigned by an instructor or teaching assistant.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Student enters the group page from the dashboard
2. Under the assignment in the group page, students will press the “upload” button.
3. Then, students submit assignments by uploading documents from his/her desktop.

### **5.1.10 Use-Case #10**

Use Case: Request Deadline Extension

Primary Actor(s): Students

Stakeholders and Interests:

- Students want an extension for his/her assignment deadlines.

Entry Condition:

- There should be a course which was created by the instructor.
- There should be an existing assignment assigned by an instructor or teaching assistant.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. Student presses the “Group Page” button from the dashboard.
2. Student presses the “Request Deadline Extension” button from the group page.

### **5.1.11 Use-Case #11**

Use Case: Request Group Formation

Primary Actor(s): Student

Stakeholders and Interests:

- Students want to request a group formation from teaching assistants.

Entry Condition:

- There should be at least one student in the group.
- The student should be signed in.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Student enters the “Course Page” from the dashboard.
2. Students press the “Request Group Formation” button from the course page.
3. A pop-up window will be shown for the request form which takes text as input from students.

### **5.1.12 Use-Case #12**

Use Case: Give Peer Review

Primary Actor(s): Student

Stakeholders and Interests:

- The student wants to review their group members.

Entry Condition:

- There should be a group formed by different students.
- The student should be signed in.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. Student presses the “Group Page” button from the dashboard.
2. Student chooses the group assignment which s/he wants to assess.
3. Student clicks the “peer review” button to give feedback and grades to his/her teammates in a pop-up window.

### **5.1.13 Use-Case #13**

Use Case: Add/Remove Member

Primary Actor(s): Student

Stakeholders and Interests:

- Student wants to add a group member or remove a member from a group.

Entry Condition:

- There should be a group formed by a specific number of students.
- Student should be signed in.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. Student enters the “Group Page” from the dashboard.
2. Student presses the “Add” button in the group page then searches the id of the other student that he/she wants to add and clicks the plus button next to the searched id.  
Student presses the “Remove” button in the group page next to the group members in order to remove an existing member from the group.

#### **5.1.14 Use-Case #14**

Use Case: Find Group

Primary Actor(s): Student

Stakeholders and Interests:

- Student wants to find a group.

Entry Condition:

- There should be existing groups created by different students.
- The student should be signed in.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Students enter the “Course Page” from the dashboard.
2. Students press the “Give Ad” button from the course page to be visible to other groups/students.

#### **5.1.15 Use-Case #15**

Use Case: Submit Polls

Primary Actor(s): Student

Stakeholders and Interests:

- Students want to submit a poll created by the instructor.

Entry Condition:

- There should be a course which was created by the instructor.
- There should be an existing poll created by the instructor.
- Students should be signed in for this operation.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. The student enters the “Course Page” from the dashboard.
2. Poll is visible at the top of the course page.
3. The student answers the poll and presses “Submit” buttons.

Quality requirements:

- Students can only make one poll submission.
- Students can make a submission only before the due date which is given by the instructor.

### **5.1.16 Use-Case #16**

Use Case: Manage Group Formation

Primary Actor(s): Teaching Assistant

Stakeholders and Interests:

- The Teaching Assistant wants to manage the requested group formations by students.

Entry Condition:

- Teaching assistant should be signed in.
- There should be a course for assistants to form groups.

Exit Condition:

- User has received a message about the success of the operation..

Success Scenario Event Flow:

1. The teaching assistant enters the “Course Page” from the dashboard.
2. The teaching assistant receives group formation requests which are given to him/her as notification.  
Then the teaching assistant clicks the “Create Group” button on the course page and creates a group with wanted student members by using their ids given in the notification.
3. The teaching assistant can remove any group.

### **5.1.17 Use-Case #17**

Use Case: Give Artifact Review

Primary Actor(s): Student, Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor (student, teaching assistant, instructor ) wants to review the artifacts of the groups.

Entry Condition:

- There should be a course created by the instructor.
- There should be groups formed by students.
- There should be artifacts submitted by groups

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Actor enters the “Course Page” from the dashboard and from here s/he can view all group links.
2. After clicking the group link, s/he will be directed to the selected group page .
3. Under the assignments, the actor can view these group’s artifacts by clicking the document link.
4. The actor can give grades and feedback to these artifacts from the group pages of others.

### **5.1.18 Use-Case #18**

Use Case: Edit Profile

Primary Actor(s): Student, Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor (student, teaching assistant, instructor ) wants to edit their personal profiles.

Entry Condition:

- The actor should have signed up.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Actor presses the “Profile” button which is at the right corner of every page that leads the actor to the profile page.
2. Actor presses the “Edit Profile” button in the profile page and edits his/her profile.

### **5.1.19 Use-Case #19**

Use Case: Text Message

Primary Actor(s): Student, Teaching Assistant, Instructor

Stakeholders and Interests:

- Actor (student, teaching assistant, instructor ) wants to text a message to a specific person or a group.

Entry Condition:

- There should be a group.
- There should be at least two actors.

Exit Condition:

- User has received a message about the success of the operation.

Success Scenario Event Flow:

1. Actor presses the “Message” button which is at the right corner of every page that leads the actor to the message page.
2. Actors can either choose to text message to group message, or direct message and text message.

## 5.2 Object and Class Model

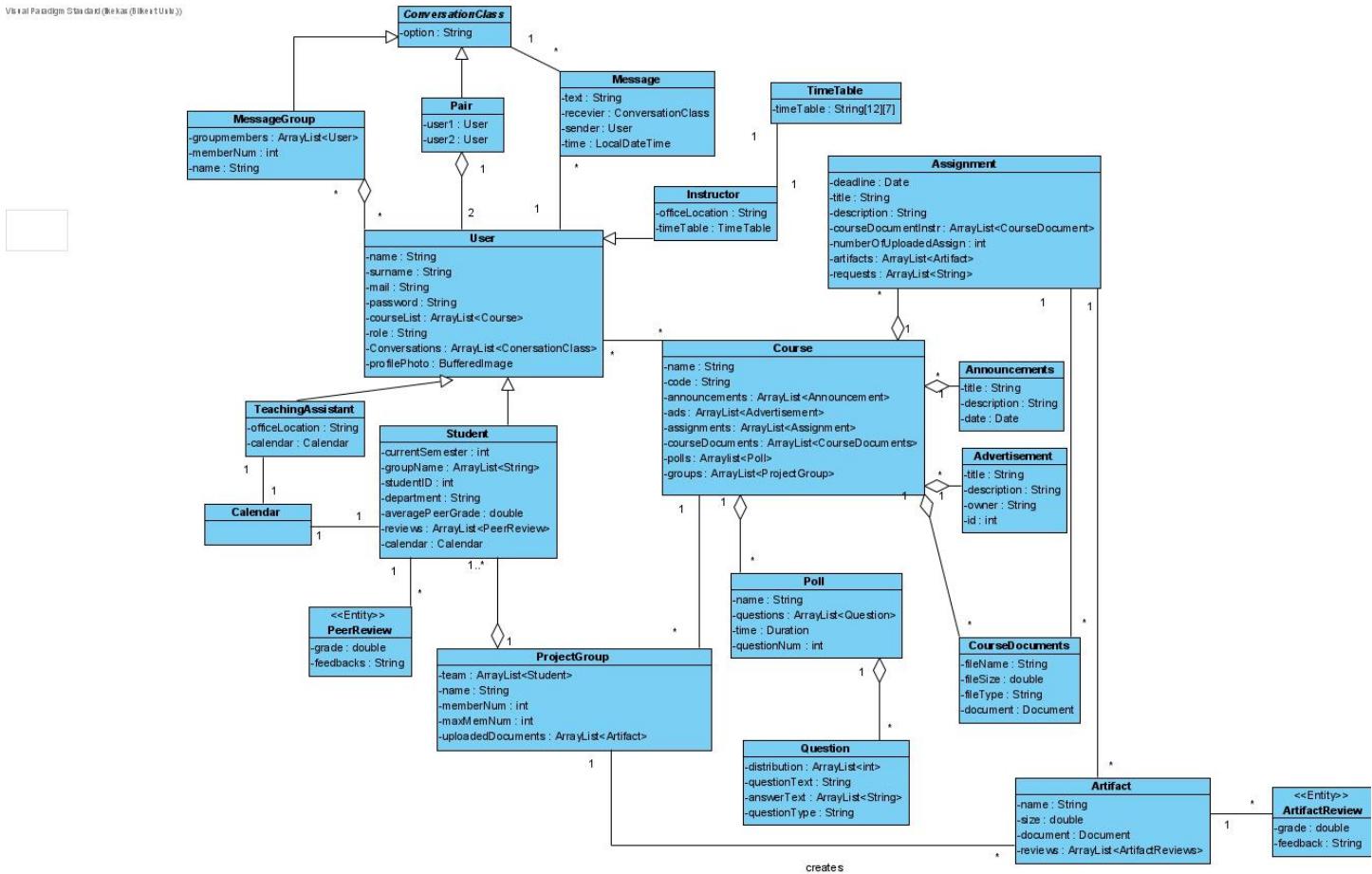


Figure 2 (Class Diagram)

### 5.2.1 User Class

This class represents and defines the features and actions of all the user options in the Catch Up. All users in the system have name, surname, mail address, password and a list of courses.

#### Operations:

- public String getName()
- public void setName(String name)
- public String getSurname()

- public void setSurname(String surname)
- public String getMail()
- public void setMail(String mail)
- public String getPassword()
- public void setPassword(String password)
- public ArrayList<Course> getCourseList()
- public void setCourseList(ArrayList<Course> courseList)
- public String getRole()
- public ArrayList<ConversationClass> getConversations()
- public void setRole(String role)
- public BufferedImage getPhoto()
- public void setPhoto(BufferedImage profilePhoto)
- public boolean gradeAssignment(ArtifactReview artifactReview,  
double grade)
- public boolean giveFeedback(Artifact artifact, String feedback)
- public void editProfile(String changedItem, String input)
- public boolean sendMessage(String text, LocalDateTime time,  
ConversationClass receiver)
- public boolean addCourse(Course course)
- public boolean removeCourse(String courseCode)
- public boolean createMessageGroup(int memnum, String name,  
ArrayList<User> groupMembers)
- public boolean createDM(User user1)
- public void addPhoto(String path)
- public void editFeedback(ArtifactReview artifact, String feedback)

- public void editGrade(ArtifactReview artifact, double grade)

### **5.2.2 Student Class**

This class is a subclass of the user class. Besides the user features, students have the information of their current semester, group name, id, department and peer grade average in the system.

#### **Operations:**

- public int getCurrentSemester()
- public void setCurrentSemester(int currentSemester)
- public void setReview(ArrayList<PeerReview> review)
- public ArrayList<PeerReview> getReview()
- public ArrayList<String> getGroupName()
- public void setGroupName(ArrayList<String> groupName)
- public int getStudentID()
- public void setStudentID(int studentID)
- public String getDepartment()
- public void setDepartment(String department)
- public double getAveragePeerGrade()
- public void setAveragePeerGrade(double grade)
- public Calendar getCalendar()
- public void setCalendar(Calendar calendar)
- public boolean givePeerGrade(Student student, double grade)
- public boolean givePeerFeedback(Student student, String feedback)
- public boolean givePeerReviewFeedback(string feedback)

- public boolean givePeerReviewGrade(PeerReview review, double grade)
- public boolean enrollCourse(String courseCode)
- public boolean uploadAssignment(Assignment assignment, Artifact artifact)
- public void requestFormGroup(String assistantMail, String text)
- public boolean giveAdvertisement(String title, String text)
- public boolean removeAdvertisement(int id)
- public boolean submitPoll(Poll poll, ArrayList<String> text)
- public boolean addMeetingTime(String title, Date time)
- public boolean removeMeeting(String title)
- public Artifact scanDesktop()
- public boolean hasReviewed(Student student)
- public void editAssignment(Assignment assignment, Artifact artifact)
- public boolean addToCalendar(Date date, String text)

### **5.2.3 ProjectGroup Class**

Project groups consist of students. Thus, this class has the list of students that are in the project group. Also, every project group has a name, number of current members and number of maximum members that can be in a group.

#### **Operations:**

- public String getName()
- public void setName(String name)
- public ArrayList<Student> getTeam()
- public int getMemberNum()

- public void setMemberNum(int memberNum)
- public int getMaxMemNum()
- public void setMaxMemNum(int maxMemNum)
- public ArrayList<Artifact> getUploadedDocuments()
- public boolean giveAdvertisement(String title, String text)
- public boolean removeAdvertisement(int id)
- public void requestDeadlineExtension(Assignment assignment)
- public boolean addMember(Student student)
- public boolean removeMember(Student student)
- public void addAssignment(Assignment assignment, Artifact artifact)
- public boolean addMeetingDate(String title, Date time)
- public boolean removeMeeting(String title)
- public void editAssignment(Assignment assignment, Artifact artifact)

#### **5.2.4 Instructor Class**

This class is a subclass of the user class. Besides the user features, instructors have their office location and time table information in the system.

##### **Operations:**

- public String getOfficeLocation()
- public void setOfficeLocation(String officeLocation)
- public TimeTable getTimeTable()
- public void setTimeTable(TimeTable timeTable)
- public String createCourse(String courseName)

- public boolean assignAssignments(CourseDocuments document, Date duedate, String title, String description)
- public boolean editAssignment(Assignment assignment, String changedItem, String input)
- public boolean shareCourseDocument(Course course, CourseDocuments document)
- public CourseDocument scanDesktop()
- public Poll createPoll(String name, int questionNum, Date duedate)
- public void seeAssignmentReport(Assignment assignment)
- public void assignDueDateToAssistant(TeachingAssistant assistant, Assignment assignment, Date duedate)
- public Question createQuestion(Poll poll, String question, String questionType)
- public void addAnswerToQuestion(Question question, String answerText)
- public boolean createAnnouncement(Course course, String title, String description, Date date)
- public void seePollReport(Poll poll)
- public void seePeerReview(ProjectGroup group)
- public void seeDeadlineExtension(Assignment assignment)
- public void removeAssignment(Assignment assignment)
- public void removePoll(Poll poll)
- public void removeQuestion(Question question)
- public void editTimeTable(String date, String time, String title, String editType)

### **5.2.5 TeachingAssistant Class**

This class is a subclass of the user class. Besides the user features, teaching assistants have their office location information in the system.

#### **Operations**

- public String getOfficeLocation()
- public void setOfficeLocation(String office)
- public Calendar getCalendar()
- public void setCalendar(Calendar calendar)
- public boolean enrollCourse(String courseCode)
- public boolean assignAssignments(CourseDocument document, Date duedate, String title, String description)
- public void editAssignment(Assignment assignment, String changedItem, String input)
- public boolean formGroup( ArrayList<Student> students , String name, int memberNum , Course course)
- public boolean deleteGroup(ProjectGroup group)
- public CourseDocument scanDesktop()
- public void removeAssignment(Assignment assignment)

### **5.2.6 Course Class**

All users in the system can have many courses. These courses have a name and a unique code. Also, there are many assignments, ads, course documents and polls in every course.

#### **Operations**

- public String getName()
- public void setName(String name)
- public String getCode()
- public void setCode(String code)
- public ArrayList<Announcements> getAnnouncements()
- public ArrayList<Advertisement> getAds()
- public ArrayList<Assignment> getAssignments()
- public ArrayList<CourseDocuments> getCourseDocuments()
- public ArrayList<Poll> getPolls()
- public boolean addGroup(ProjectGroup group)

### **5.2.7 Poll Class**

Polls can be created by instructors for students to reflect their opinions on course subjects. Each poll has a name, certain number of questions and a time duration to be answered.

#### **Operations**

- public Duration getTime()
- public void setTime(Duration time)
- public String getName()
- public void setName(String name)
- public ArrayList<Question> getQuestions()
- public int getQuestionNum()
- public void setQuestionNum(int num)

### **5.2.8 Question Class**

This class represents poll questions. Every question has a question type, question text to ask the question, list of answer text to demonstrate different answer options and list of distribution for each answer option.

#### **Operations**

- public String getQuestionText()
- public void setQuestionText(String text)
- public ArrayList<String> getAnswer()
- public void addAnswer(String answer)
- public void setDistribution(ArrayList<int> distr)
- public ArrayList<double> calculateDistributionPercentage()
- public String getQuestionType()
- public void setQuestionType(String questionType)

### **5.2.7 Advertisement Class**

This class is for individual students and groups to give ads for a team or a team member to make group formation easier.

#### **Operations**

- public String getTitle()
- public void setTitle(String title)
- public String getDescription()
- public void setDescription(String description)
- public String getOwner()
- public void setOwner(String owner)
- public int getID()

- public void setID(int id)

### 5.2.8 Assignment Class

This class is for instructors to give project related assignments such as reports to project groups and for project groups to submit the assignment. Each assignment has a title, description, a deadline. Also, it has two different course documents list one for instructors to give assignment details with a document such as pdf and another list that includes each groups' submitted assignments. Finally, assignments have grades and feedback that provides artifact and peer review features to the system.

#### Operations

- public Date getDeadline()
- public void setDeadline(Date deadline)
- public String getTitle()
- public void setTitle(String title)
- public String getDescription()
- public void setDescription(String description)
- public ArrayList<String> getRequests()
- public void setRequest(ArrayList<String> requests)
- public ArrayList<CourseDocument> getCourseDocument()
- public int getNumberOfUploadedAssign()
- public void setNumberOfUploadedAssign(int number)
- public ArrayList<Artifact> getArtifacts()
- public void addCourseDocument(CourseDocument doc)
- public void addRequest(String request):

- public void sendRequestNotification(Instructor instructor)

### **5.2.9 CourseDocuments Class**

This class is for instructors to share project related slides or assignments in the system. Each course document has a file name, file size and a file type. It is for uploading files to the related course by all users.

#### **Operations**

- public String getFileName()
- public void setFileName(String fileName)
- public double getFileSize()
- public void setDocument(Document document)
- public Document getDocument()
- public void setSize(String size)
- public String getFileType()
- public void setType(String type)

### **5.2.10 Announcement Class**

This class is for instructors to make announcements to all of the course's students.

Each announcement has a title, description and a date.

#### **Operations**

- public String getTitle()
- public void setTitle(String title)
- public String getDescription()
- public void setDescription(String description)
- public Date getDate()
- public void setDate(Date announcementDate)

### **5.2.11 TimeTable Class**

This class is for instructors to show their time table at the system so that students can see the empty time slots of the instructor in order to ask them for an appointment.

#### **Operations**

- public void setTimeTable(String[12][7] timeTable )
- public String[12][7] getTimeTable()
- public boolean addActivity(int row,int col, String activity)
- public boolean removeActivity(int row, int col)

### **5.2.12 Calendar Class**

This class is for students and teaching assistants to see the upcoming assignments or meetings on a calendar with due dates.

#### **Operations**

- public boolean addActivity(Date date, Time time, String title)
- public boolean removeActivity(Date date)
- public ArrayList<String> sendNotificationInfo()

### **5.2.13 Message Class**

This class is for messaging in the system. Messages have a sender, a receiver, time and a text that includes the message information.

#### **Operations**

- public String getText()
- public void setText(String text)
- public User getReceiver()
- public void setReceiver(ConversationClass receiver)

- public User getSender()
- public void setSender(User sender)
- public LocalDateTime getTime()
- public void setTime(LocalDateTime time)

### **5.2.16 ConversationClass Class**

This class is for representing people that take roles in messaging. Each conversation class can have one or more messages.

- public void setOption(String option)
- public StringgetOption()

### **5.2.14 Pair Class**

This class is the subclass of conversation class. It is for private messages between two users. Thus, every pair has user1 and user2.

#### **Operations**

- public User getUser1()
- public void setUser1(User user1)
- public User getUser2()
- public void setUser2(User user2)

### **5.2.15 MessageGroup Class**

This class is the subclass of conversation class. It is for group messages in the system. Message groups consist of one or more users. Every message group has a name, number of members and list that includes the users in the group.

#### **Operations**

- public ArrayList<User> getGroupMembers()
- public void setGroupMembers(ArrayList<User> groupMembers)

- public int getMemberNum()
- public void setMemberNum(int memberNum)
- public String getName()
- public void setName(String name)
- public boolean addMember(Student student)
- public boolean removeMember(int studentId)

### **5.2.16 Artifact Class**

This class is for representing artifacts of the projects that are created and uploaded by students.

#### **Operations**

- public String getName()
- public void setName(String artifactName)
- public double getSize()
- public void setSize(String size)
- public void addReview(ArtifactReview review)
- public ArrayList<ArtifactReview> getReviews()
- public void setReviews(ArrayList<ArtifactReview> reviews)
- public void setDocument(Document document)
- public Document getDocument()
- public String getType()
- public void setType(String type)
- public boolean giveArtifactFeedback(String feedback)
- public boolean giveArtifactGrade(double grade)
- public int calculateStudentGradeAverage()

### **5.2.17 ArtifactReview Entity Class**

This class is for representing the reviews given to an artifact.

#### **Operations**

- public double getGrade()
- private String getFeedback()
- public void setGrade(double grade)
- private void setFeedback(String feedback)

### **5.2.18 PeerReview Entity Class**

This class is for representing the reviews given to an artifact.

#### **Operations**

- public double getGrade()
- private String getFeedback()
- public void setGrade(double grade)
- private void setFeedback(String feedback)

## 5.3 Dynamic Models

### 5.3.1 Sequence Diagrams

#### 5.3.1.1 Sign Up

**Scenario:** User without an account presses the sign up button on the home page. Then the registration form is opened. After, the information that the user entered was checked from the authentication server. If information is sufficient and correct registration is confirmed, then, information is added to the database. Else if information already exists in other account, registration is denied.

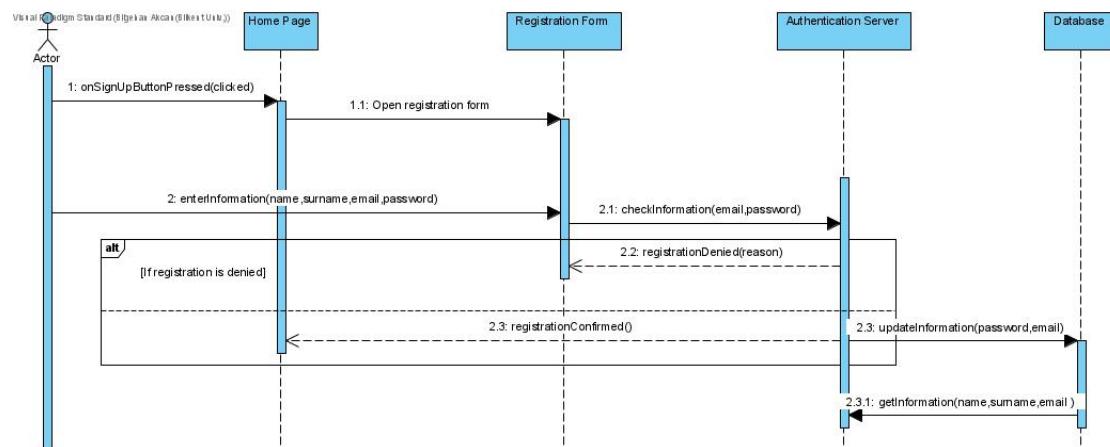
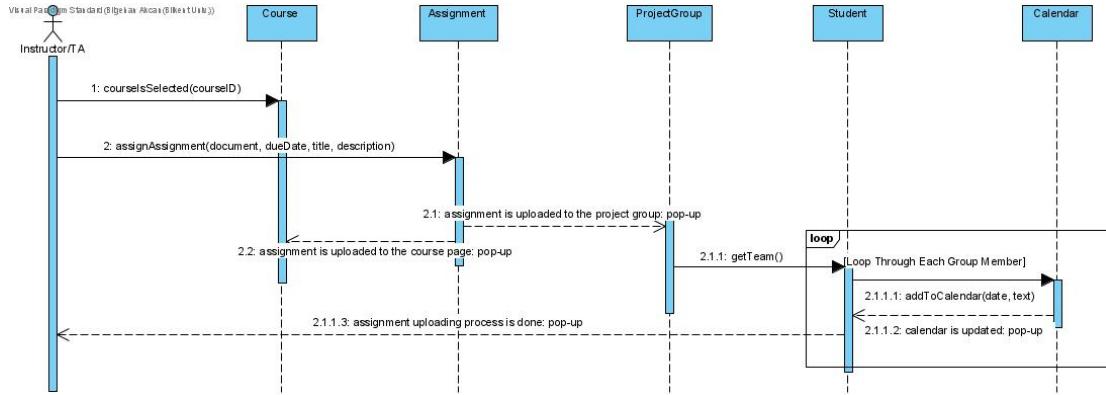


Figure 3

#### 5.3.1.2 Assign Assignments

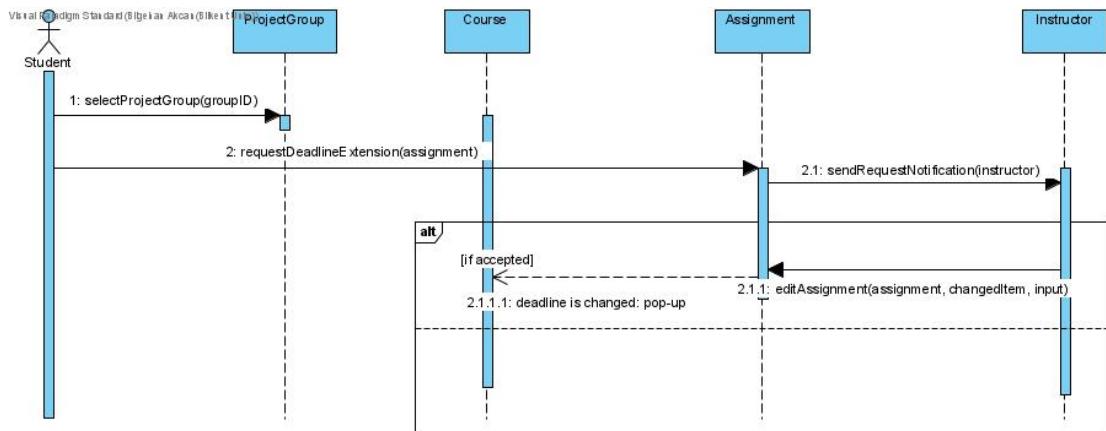
**Scenario:** In order to give assignments, firstly, the instructor or TA selects which course to be assigned. Then, they assign the assignment to the selected course. After, that assignment is added to the projects' group pages and general course page. Afterwards, from team to each student due date is added in their calendar by using a loop.



**Figure 4**

### 5.3.1.3 Request Deadline Extension

**Scenario:** In order to request deadline extension, a student first selects the project group. Then, from the assignment part, she/he requests an extension for the deadline. This request reaches the instructor. If the instructor decides to extend the deadline, he/she edits the assignment and the deadline is changed a pop-up message will be displayed in the course page.

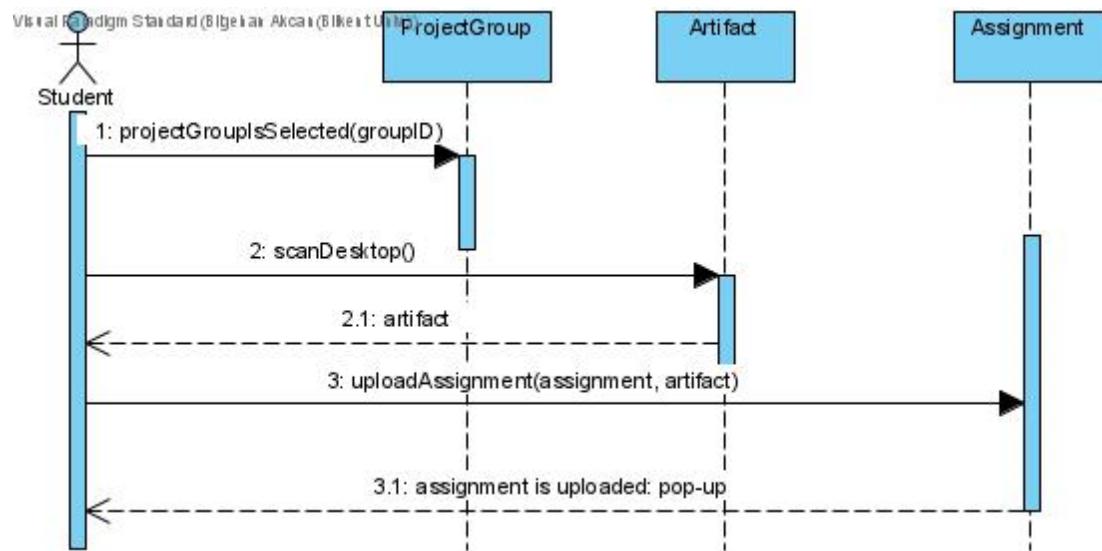


**Figure 5**

### 5.3.1.4 Upload Assignments

**Scenario:** In order to upload the assignment, the student needs to press the project group button that leads her/him to the project group. Then, by scanning the desktop, the document will be chosen. Then, the student can upload the artifact to the relevant

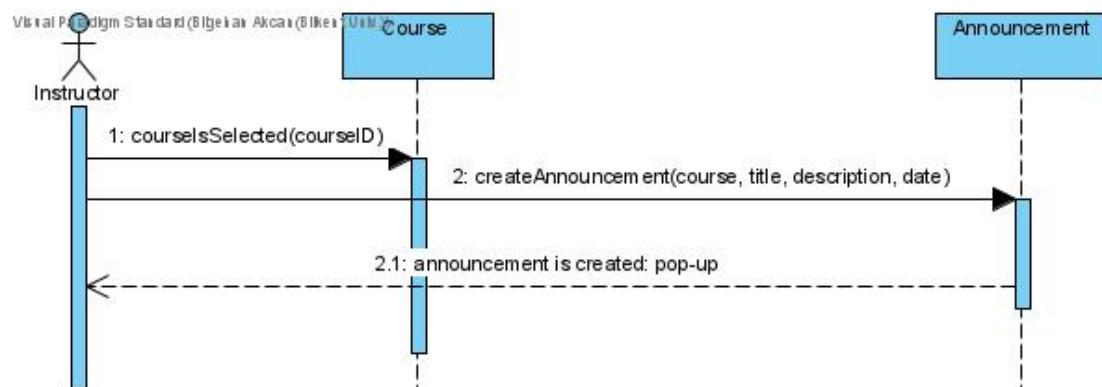
place by assignment object. After that assignment is uploaded, a message will be displayed.



**Figure 6**

### 5.3.1.5 Make Announcements

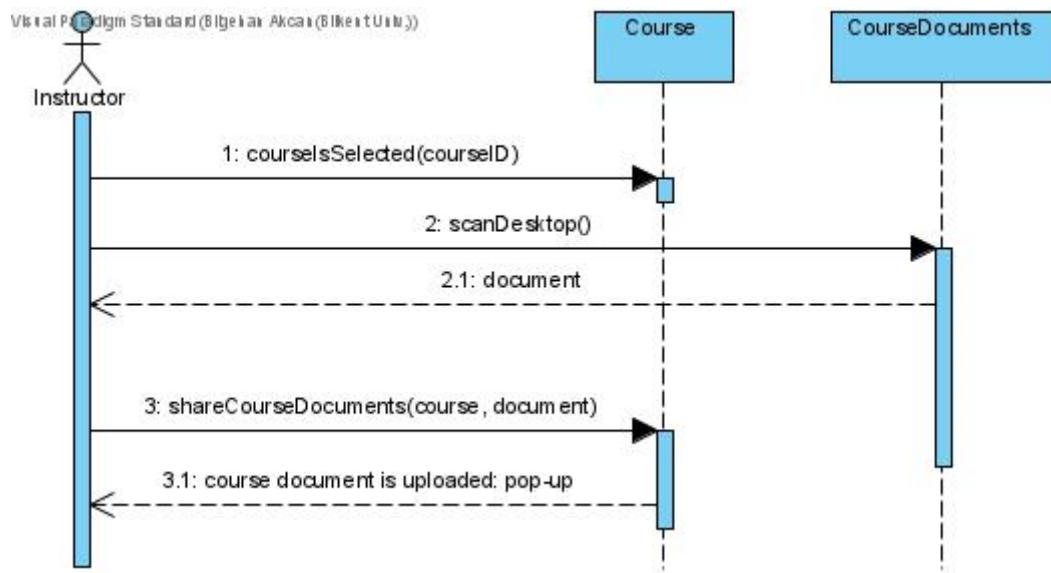
**Scenario:** In order to make an announcement, the instructor selects the relevant course. Then, he/she creates an announcement and created pop-up is forwarded to instructor.



**Figure 7**

### 5.3.1.6 Share Course Documents

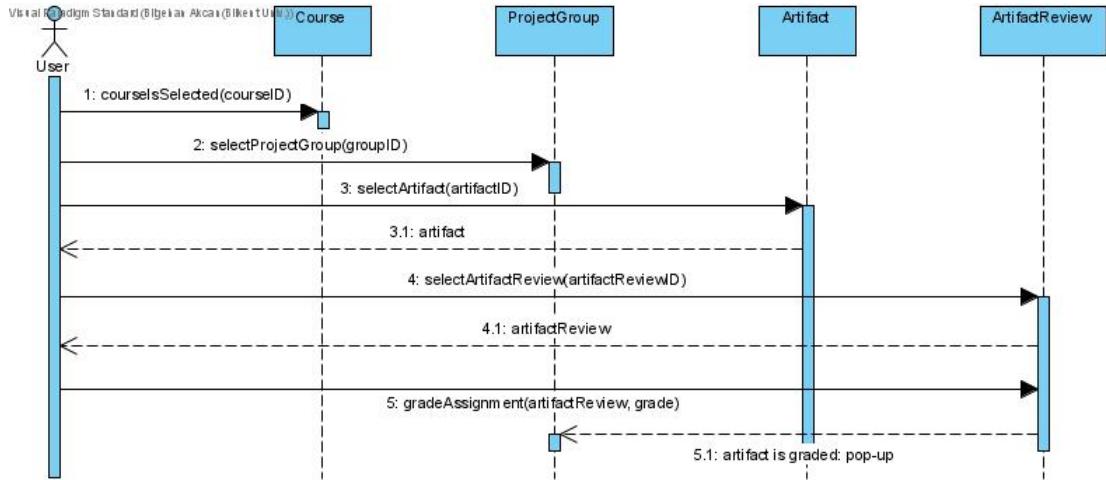
**Scenario:** First of all, the instructor needs to choose the relevant course. Then, he/she will share the course document by scanning the desktop. Then, documents will be added to the course by the share course documents function. After the course document is uploaded, a message will be displayed.



**Figure 8**

### 5.3.1.7 Give Grades

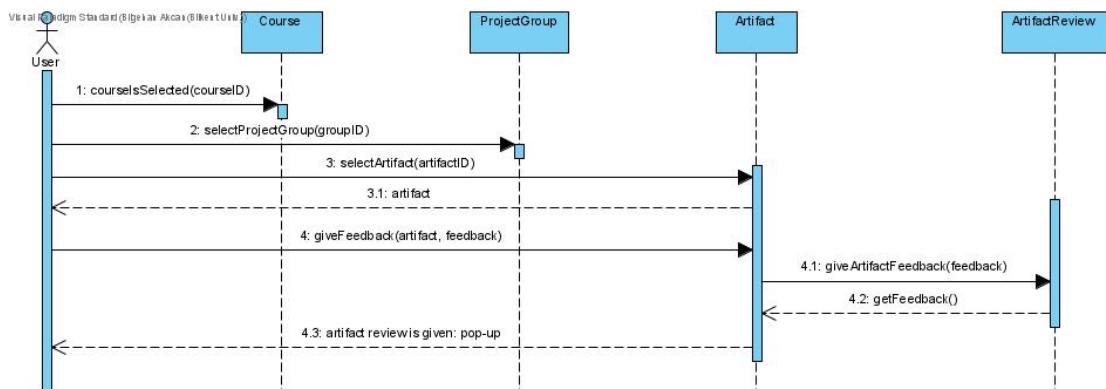
**Scenario:** In order to give grades first the relevant course needs to be selected. After that the project group which it's work will be graded needs to be selected. Then, the chosen group's artifact can get graded by forwarding to artifact review class. After that artifact is graded, a message will be displayed.



**Figure 9**

### 5.3.1.8 Give Feedback

**Scenario:** In order to give grades first the relevant course needs to be selected. After that the project group whose work will be evaluated by feedback needs to be selected. Then, the chosen group's artifact can receive feedback by forwarding to artifact review class. After that artifact is given feedback, a message will be displayed.

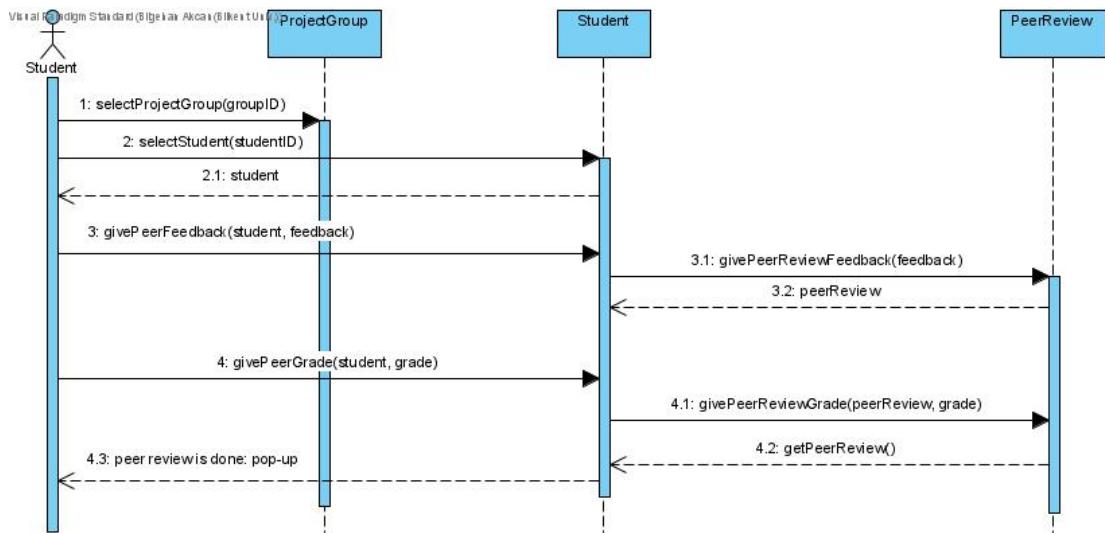


**Figure 10**

### 5.3.1.9 Peer Review

**Scenario:** In order to give feedback students must choose the project group first. Then for a particular student in the group, actor students will give reviews. These

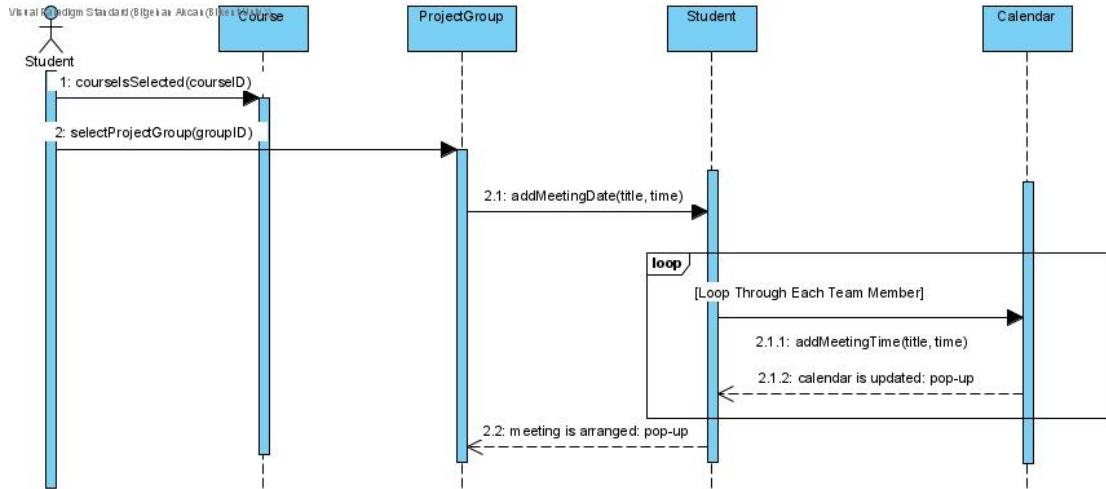
reviews can be feedback or a grade and implemented to the student object. After that peer review is done a pop-up message will be displayed.



**Figure 11**

### 5.3.1.10 Project Group Meeting

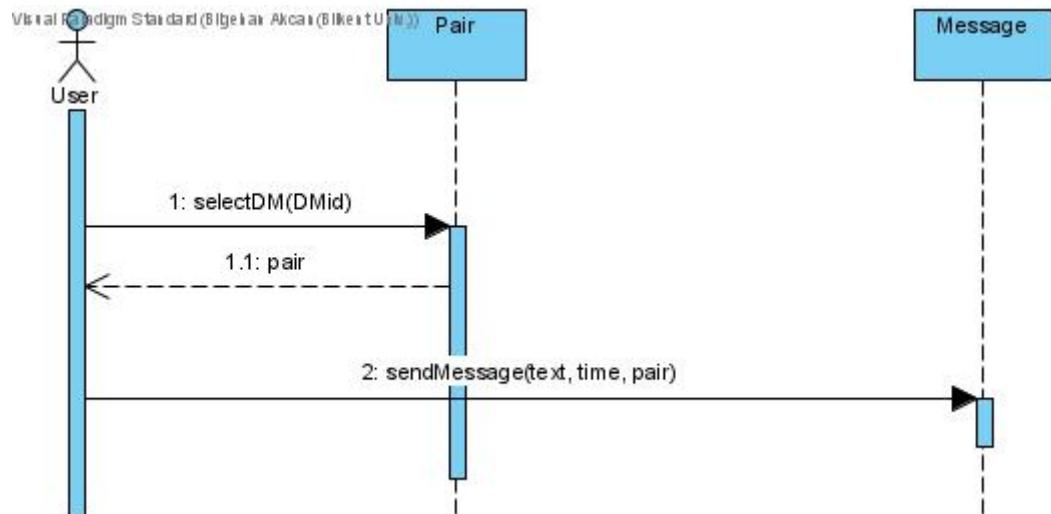
**Scenario:** In order to arrange a group meeting a student needs to choose the relevant course first. Then he/she chooses the project group one belongs to. Then the decided date will be given for students of the group and by a loop the meeting will be added to the calendars of each student by a loop. After that's done the calendar is updated and a message will be displayed in the student. Then the meeting is arranged message will be sent to the group.



**Figure 12**

### 5.3.1.11 Send Direct Message

**Scenario:** In order to send a message first users need to choose who to send the message and by choosing that messaging pair takes place. After the pair is indicated the message object is written.



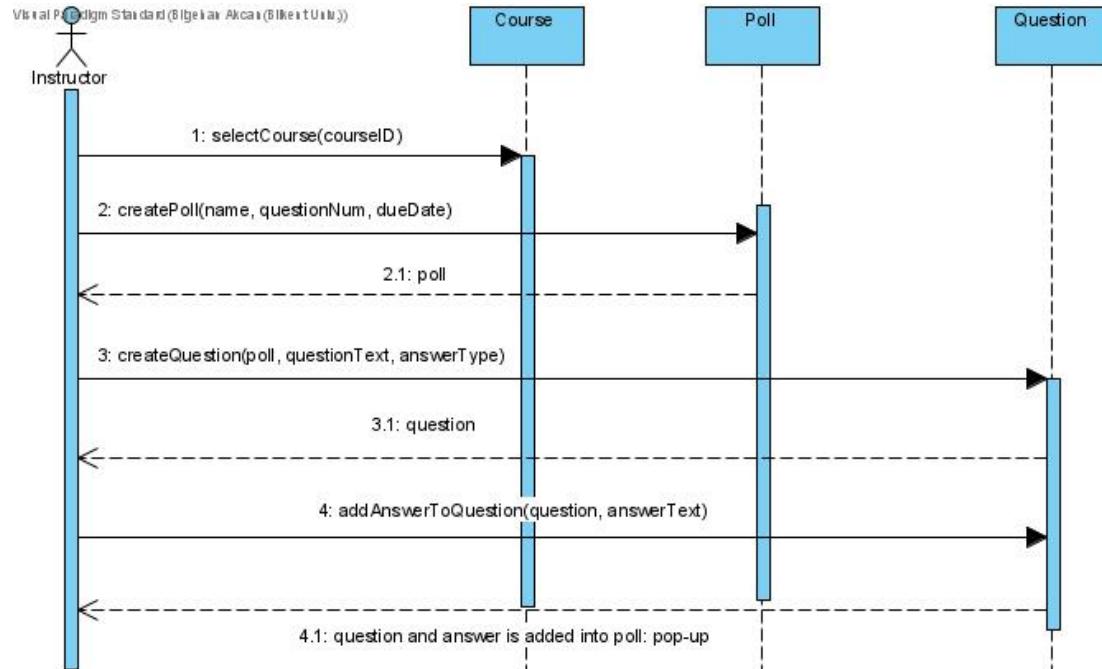
**Figure 13**

### 5.3.1.12 Create Poll

**Scenario:** First instructor chooses which course he/she wants to create a poll about.

Then the instructor chooses to create a poll and after the poll is created question

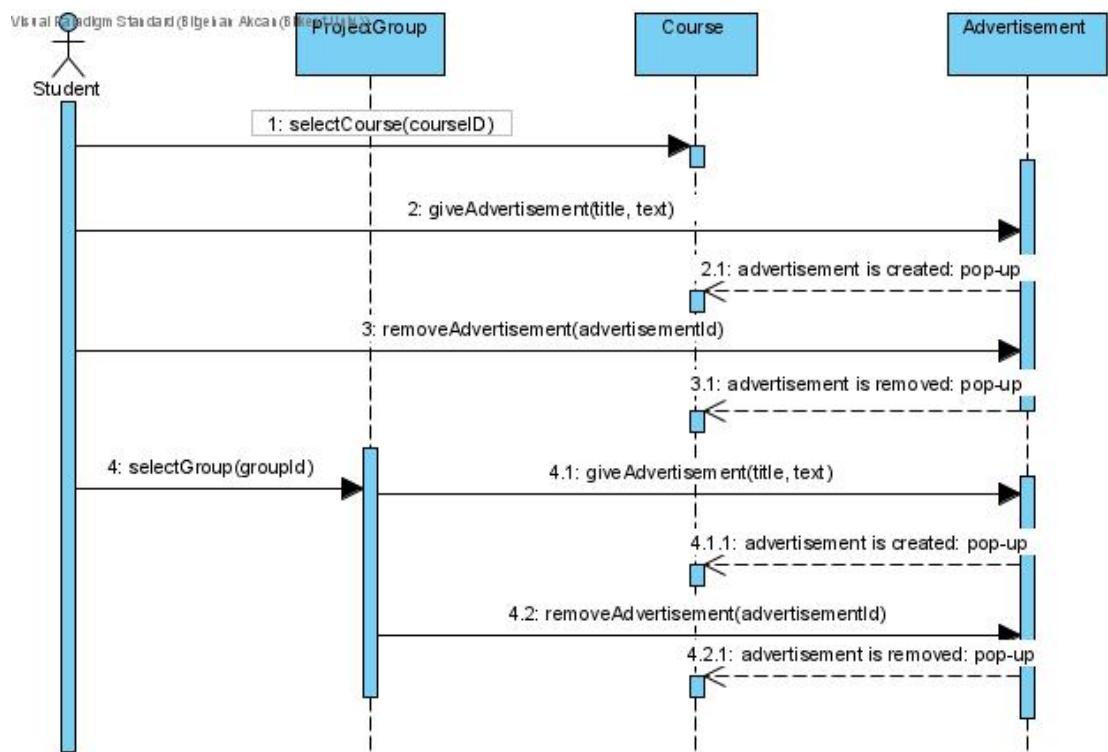
objects are created next. After questions are implemented the answers might be necessary therefore the instructor enters the answers necessary ones. Then, the questions and answers are added to the poll message will be displayed.



**Figure 14**

### 5.3.1.13 Advertisement

**Scenario:** The actor, who is a student in this scenario, in order to give individual advertisement, selects the course and creates the advertisement of his/her. It can be removed at the same place if the student is found a group or wants to remove it. In order to give group advertisement, the student goes to one's own group page then selects the course and creates the advertisement of the group. It can be removed at the same place if the group is full or wants to remove it.

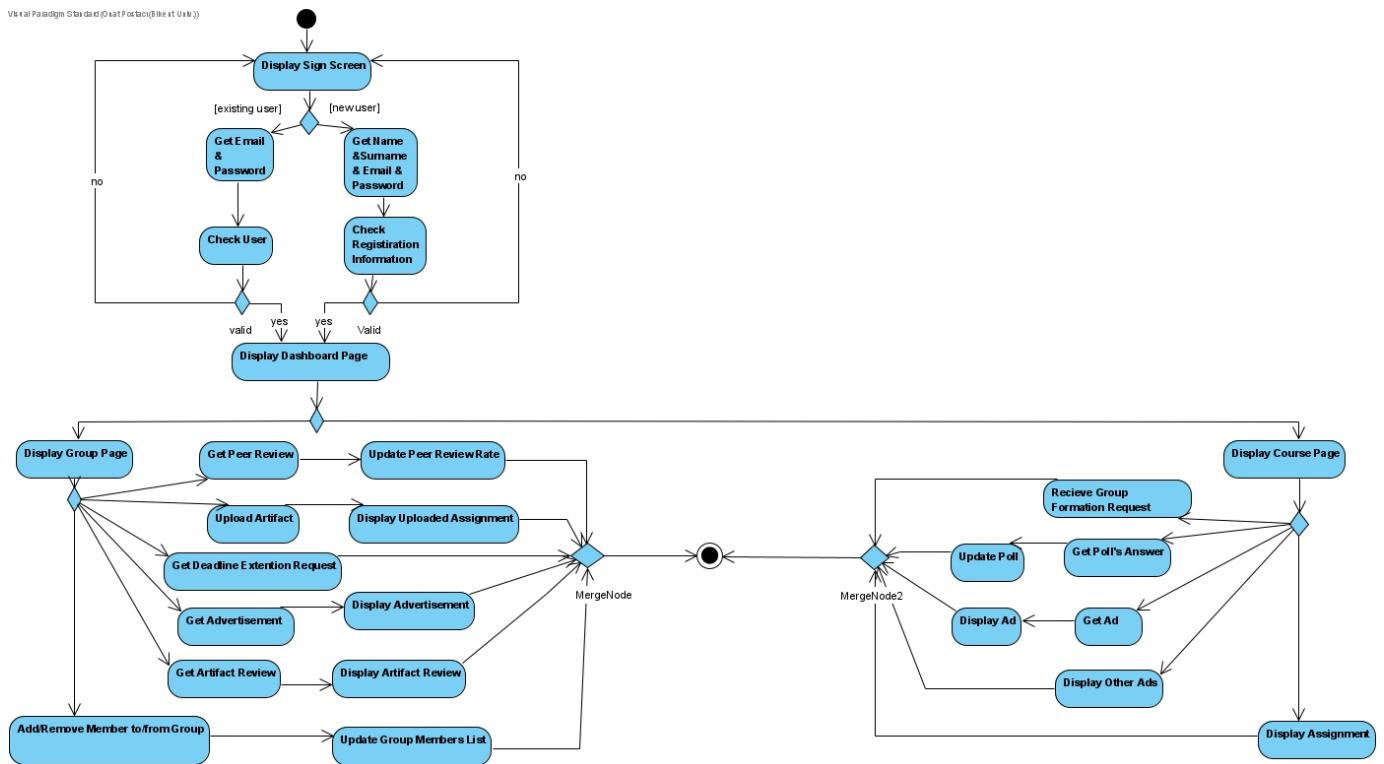


**Figure 15**

### 5.3.2 Activity Diagrams

Activity diagrams represent the sequencing and coordination of lower level behaviors

[1]. We have 3 types of activity diagrams for our project. First one is for students:



**Figure 16 (Activity Diagram For Students)**

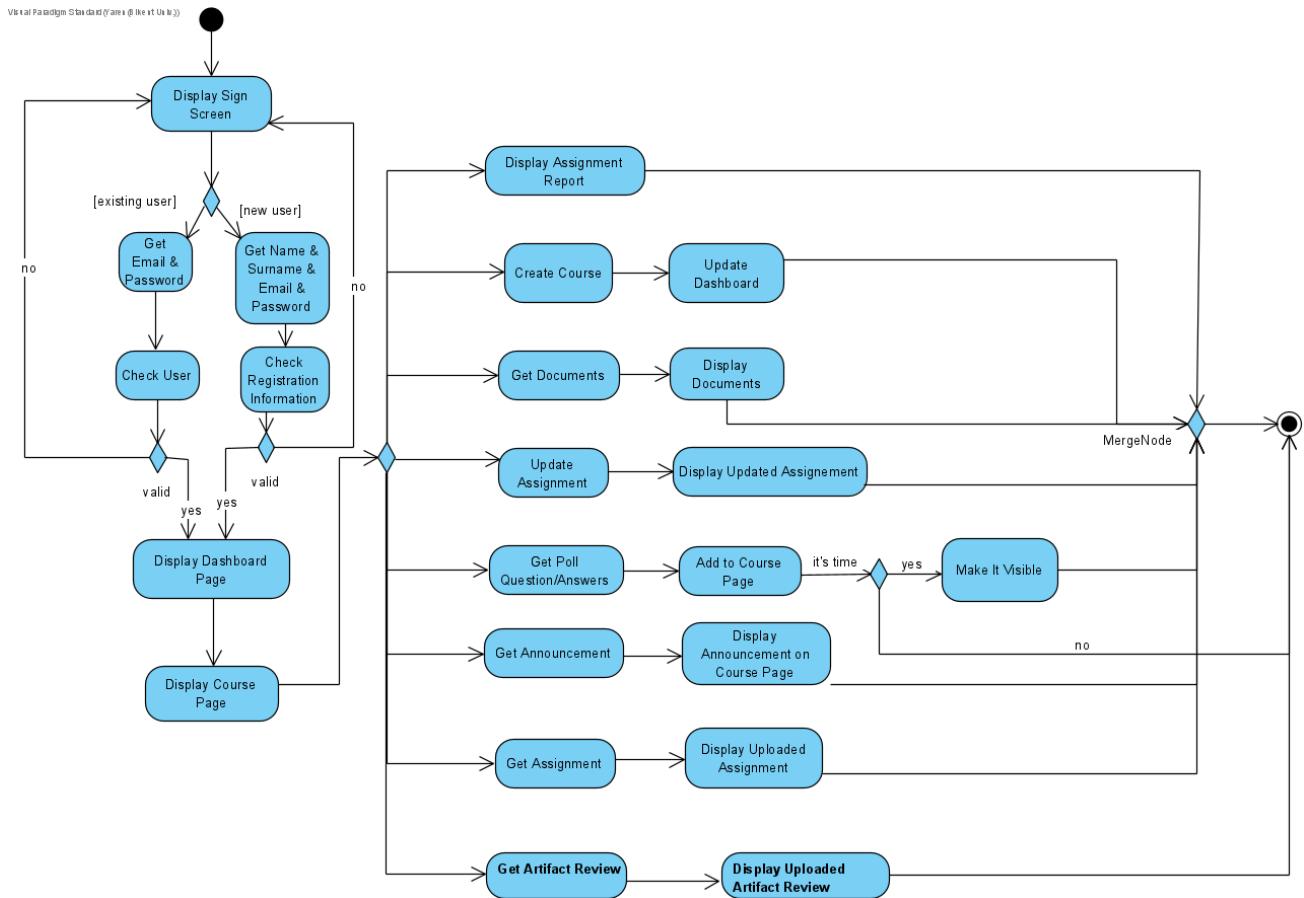
Initially, the system displays the opening screen of our web application. According to the choice of the student, the system gets the information that is needed to sign in or sign up to the application. If the information entered by the user is valid, the system displays the dashboard. Otherwise, it displays the opening screen again. After validation is done, according to the decision of the user, the system displays either group page or course page. If the group page is selected, there are 6 paths of action sequences that can be taken by the system:

- Peer Review: Peer review entered by the user is caught and the system updates the average peer review grade displayed on the profile page of the user.
- Uploading Artifact: The artifact uploaded by the user is caught and the system updates the group page to show the uploaded artifact on the group page.
- Deadline Extension: Deadline extension request is gotten by the system and conveyed to the instructor.
- Advertising: The advertisement to find a group member is gotten and displayed on the course page by the system.
- Reviewing artifact: The artifact review entered by users is gotten and displayed on the group page by the system.
- Organizing group: The student is added to the group or removed from the group. Then, the updated group list is displayed on the group page by the system.

If the course page is selected, there are 4 paths of action sequences that can be taken by the system:

- Group Formation: After students decide who to group with, they create a group formation request and send it to the teaching assistants. The system gets the group formation request and conveys it to the teaching assistants.
- Answering Poll Question: The system gets the answers given by the students to the poll question and updates the poll results.
- Advertising: The advertisement to find a group is gotten and displayed on the course page by the system.
- Assignments: The system displays all the assignments given by the instructor on the course page.

The second one is for instructors:

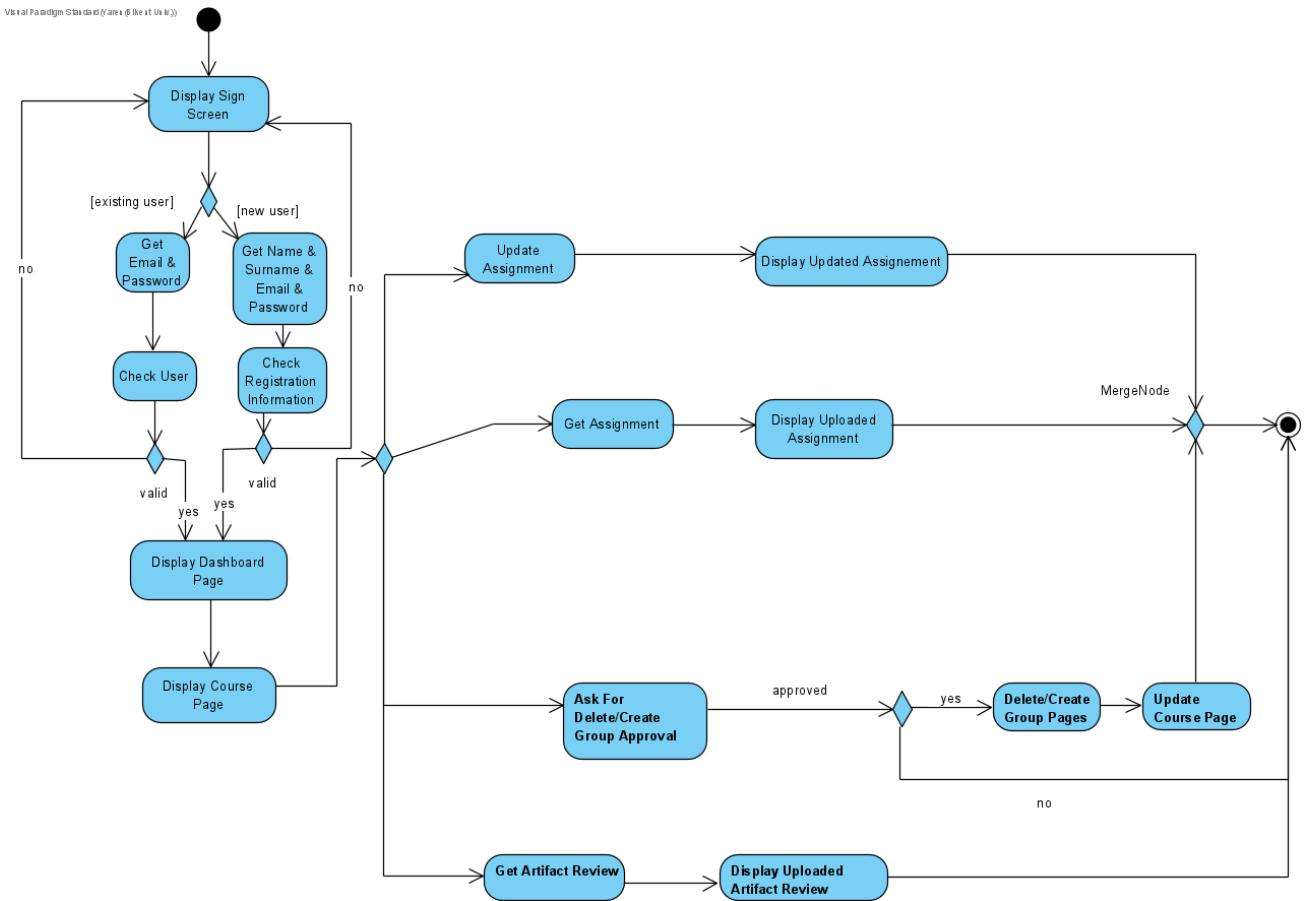


**Figure 17 (Activity Diagram For Instructor)**

Initially, the system displays the opening screen of our web application. According to the choice of the instructor, the system gets the information that is needed to sign in or sign up to the application. If the information entered by the user is valid, the system displays the dashboard. Otherwise, it displays the opening screen again. After validation is done, the system displays the course page. From that moment on, there are 7 paths of action sequences that can be taken by the system:

- Uploading Assignment: The system gets the assignment uploaded by the instructor and updates the course page to display the assignment.

- Announcements: The system gets the announcements formed by the instructor and updates the course page to display the announcement.
- Creating Poll: The system gets the poll's questions and answers formed by the instructor. If it is the time to share the poll with students, the system makes it visible on the course page. Otherwise, the process is terminated.
- Updating Assignment: The system updates the assignment file and updates the course page to display the updated assignment.
- Documents: The system gets the documents uploaded by the instructor and updates the course page to display documents.
- Creating Course: The system creates the course with a course page and updates the dashboard to show the new course too.



### **Figure 18 (Activity Diagram for Teaching Assistants)**

Initially, the system displays the opening screen of our web application. According to the choice of the teaching assistant, the system gets the information that is needed to sign in or sign up to the application. If the information entered by the user is valid, the system displays the dashboard. Otherwise, it displays the opening screen again. After validation is done, the system displays the course page. From that moment on, there are 3 paths of action sequences that can be taken by the system:

- Updating Assignment: The system updates the assignment file and updates the course page to display the updated assignment.
- Uploading Assignment: The system gets the assignment uploaded by the teaching assistant and updates the course page to display the assignment.
- Editing Groups: The system asks the user to approve the request of deleting/creating groups. If it is approved, the system deletes/creates the group page and updates the course page to add/delete the group number to the existing groups list. If it is not approved, the process is terminated.

### 5.3.3 State Diagrams

#### 5.3.3.1 Assignment State Diagram

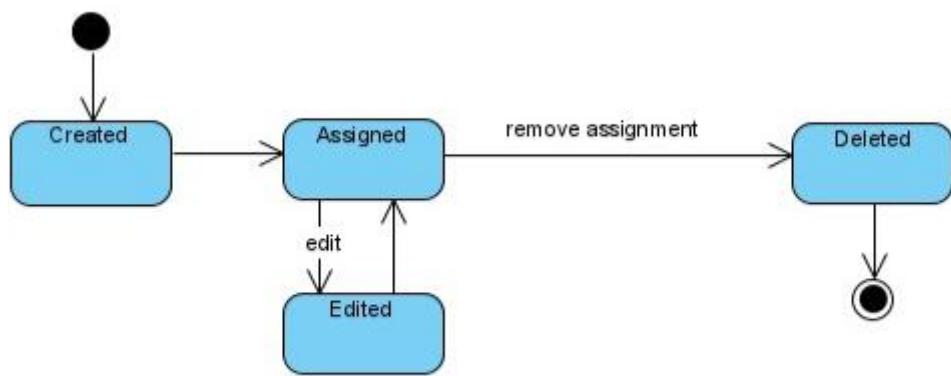
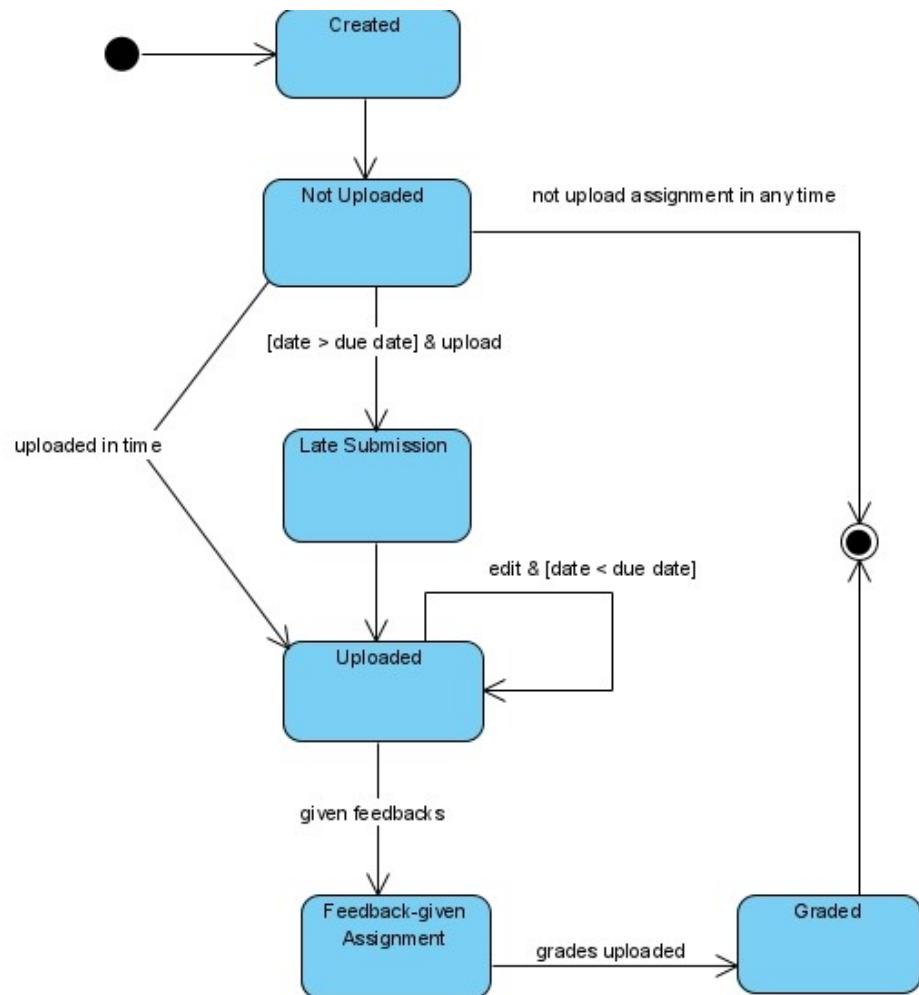
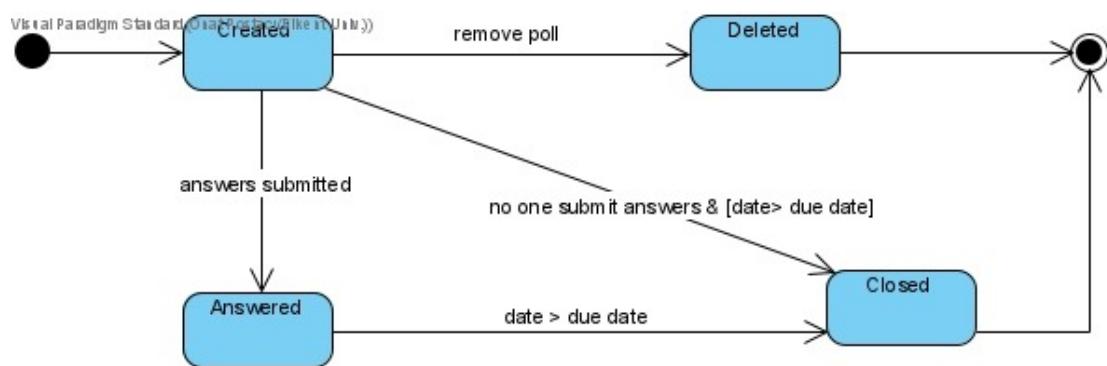


Figure 19



**Figure 20**

### 5.3.3.2 Poll State Diagram



**Figure 21**

### 5.3.3.3 Ad State Diagram

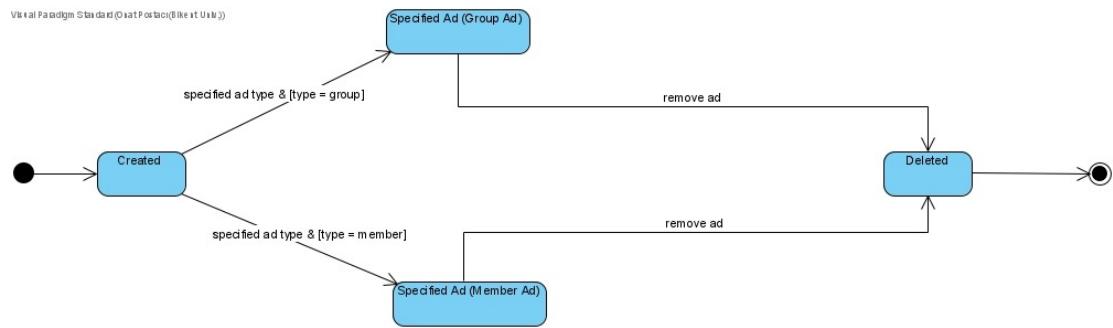


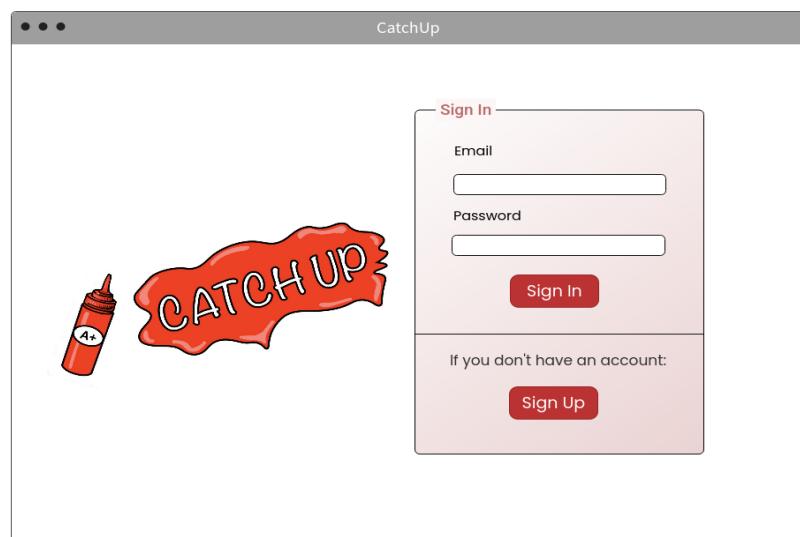
Figure 22

## 5.4 User Interface

### 5.4.1 Sign In Page

Users will sign in to the web page server by entering their email and password. If they try to enter with the wrong password or email they will receive an error message. If they do not have an account they will click the sign up button and will be forwarded to sign up page.

Figure 23



### 5.4.2 Sign Up Page

Users who have not registered yet will sign up through this page. The users have to choose their profile kind and they will enter the data according to their choice.

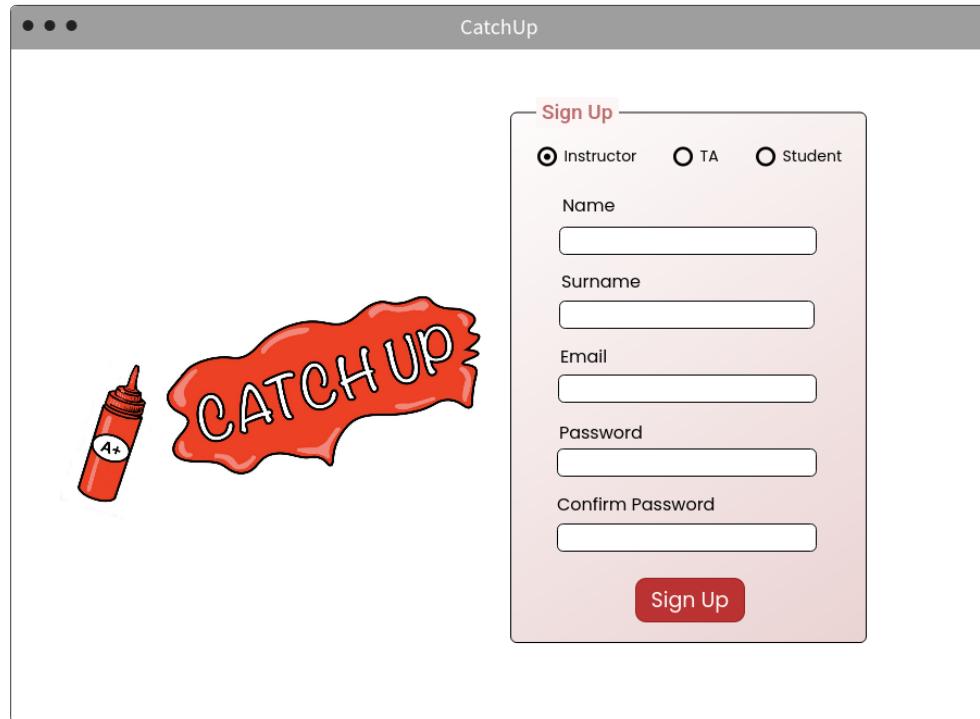
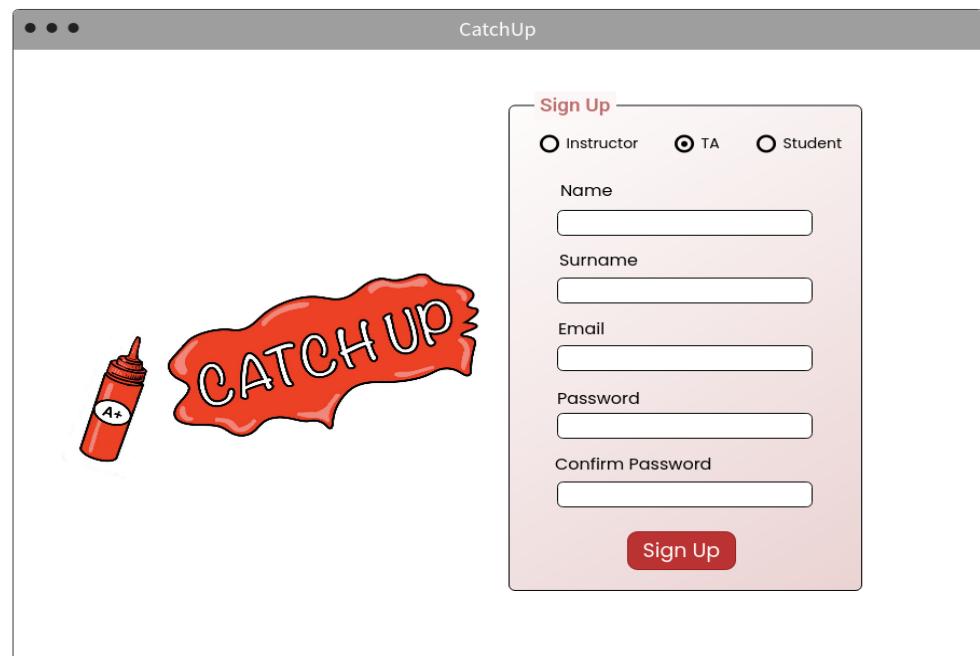
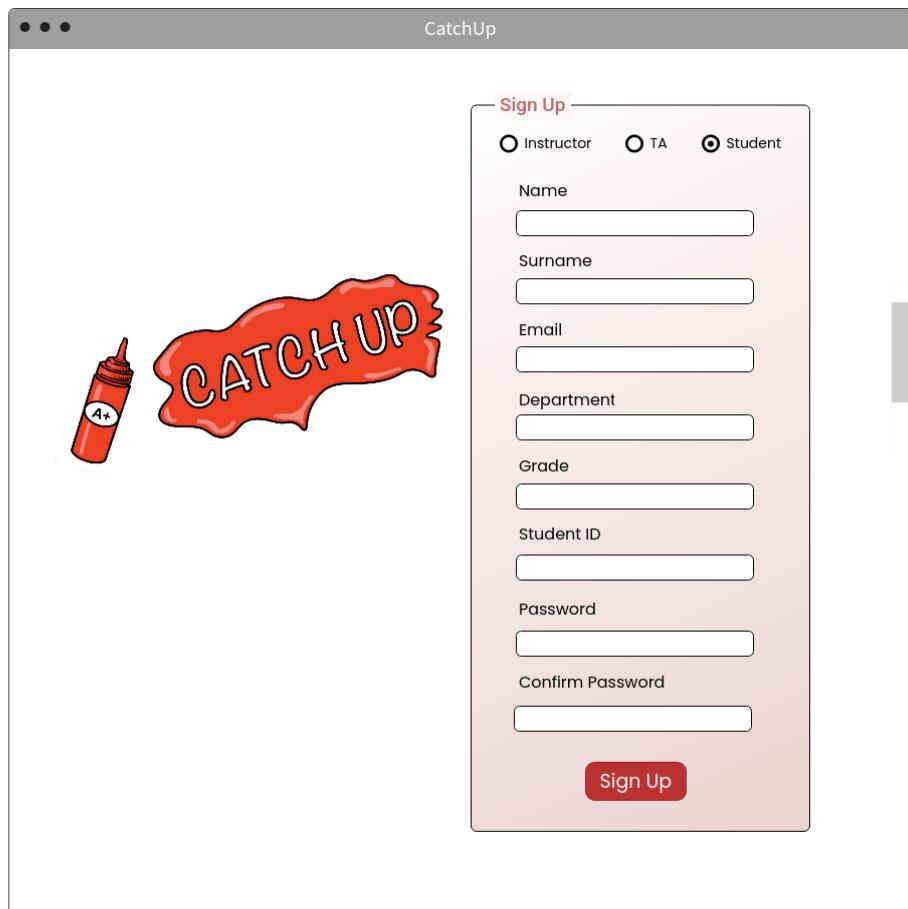


Figure 24



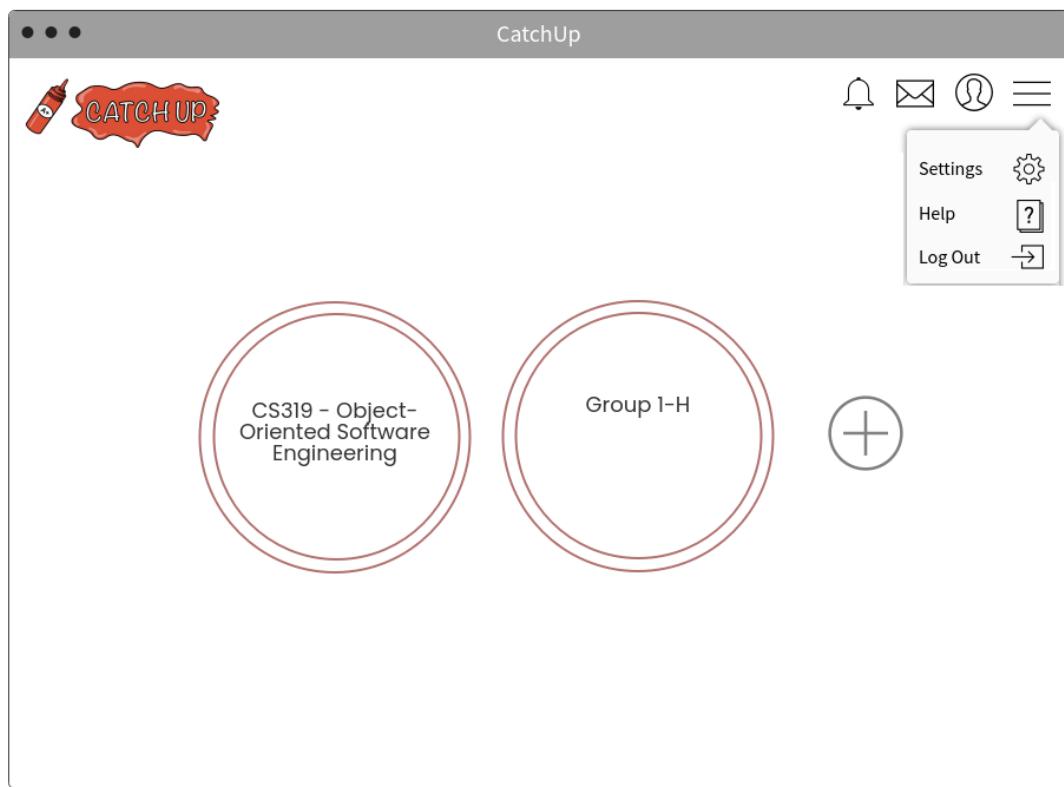
**Figure 25**



**Figure 26**

### 5.4.3 DashBoard Pages

Users will be forwarded to the dashboard page in order to choose which page they will enter. If the user is an instructor as in the first page example they will encounter a plus button to add courses and see their courses. If it is a student as in the second page they will encounter a plus button to enroll courses through code and they will see their own project group.



**Figure 27**



**Figure 28**

#### 5.4.4 Message Pages

Users will enter the message page through the message button above in every page or when they click to send message buttons at profiles. There are two kinds of messaging scenarios: one is direct messaging in the first page and other one is the group chats as in the second page.

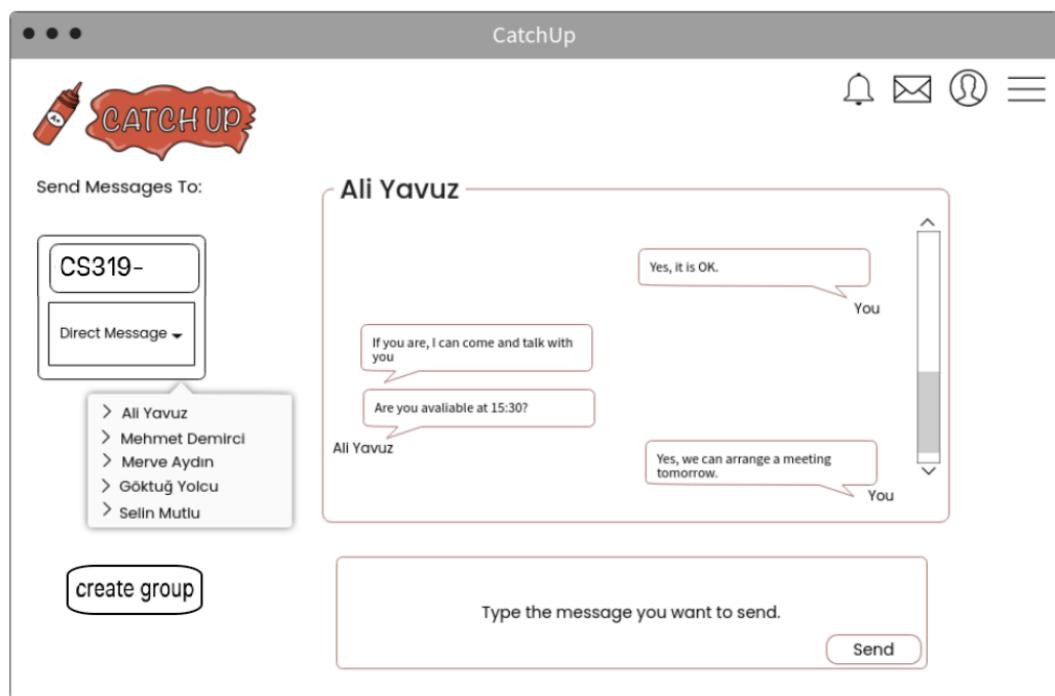


Figure 29

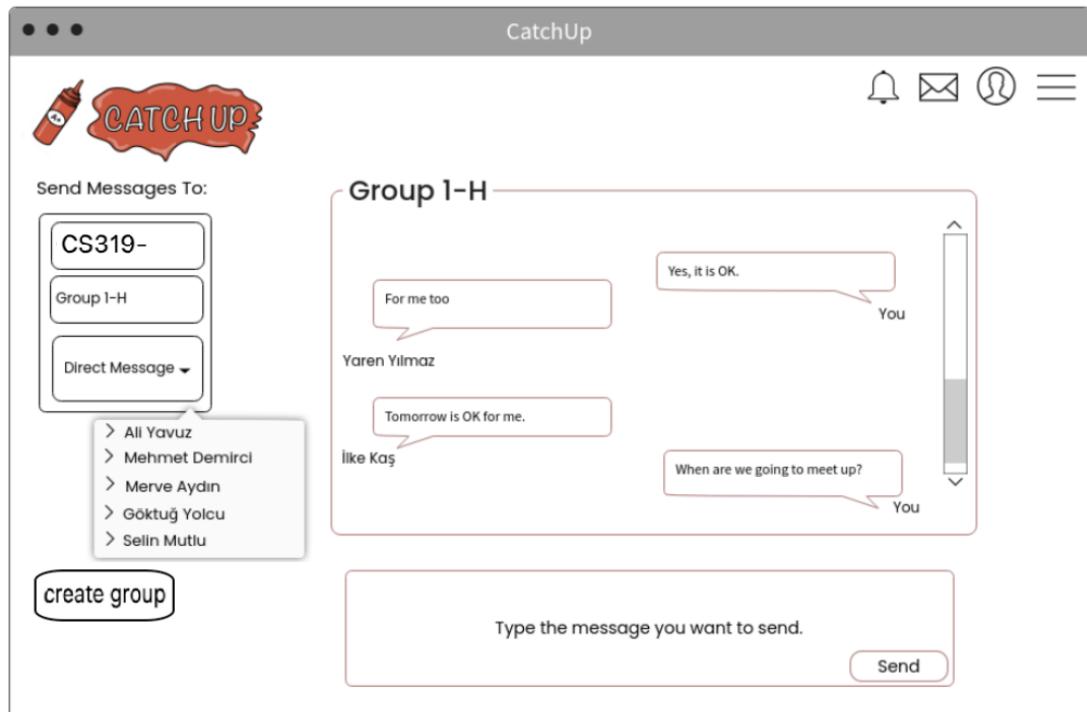
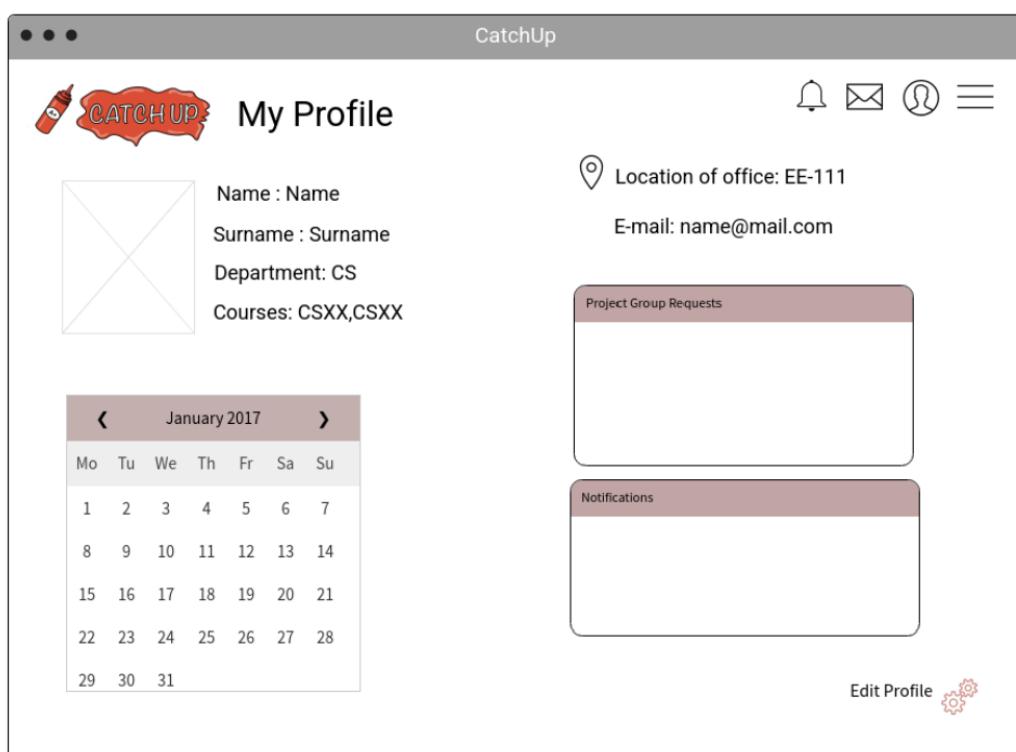


Figure 30

### 5.4.5 Own Profile Pages

Users will see their profile pages differently according to their account kind but they all will see registering information and notifications. For instance in a student's profile one will see s/her comments, peer reviews and a calendar. In the instructor profile they will add their time table to indicate their free time to see students. In the Teaching assistant's page there will be project group requests..



**Figure 31**

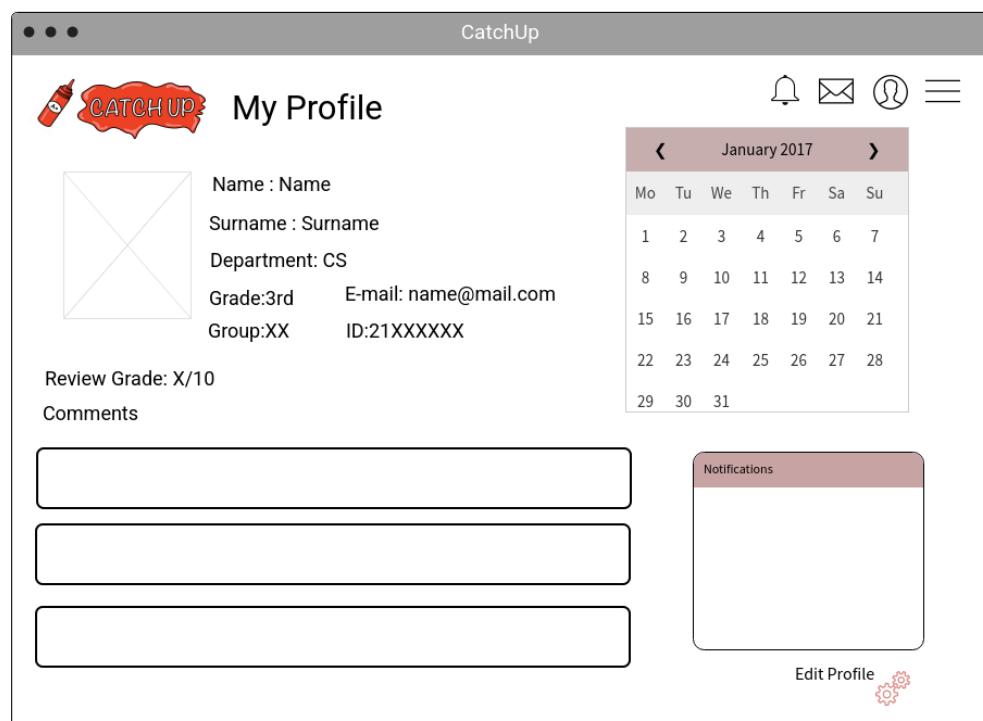


Figure 32

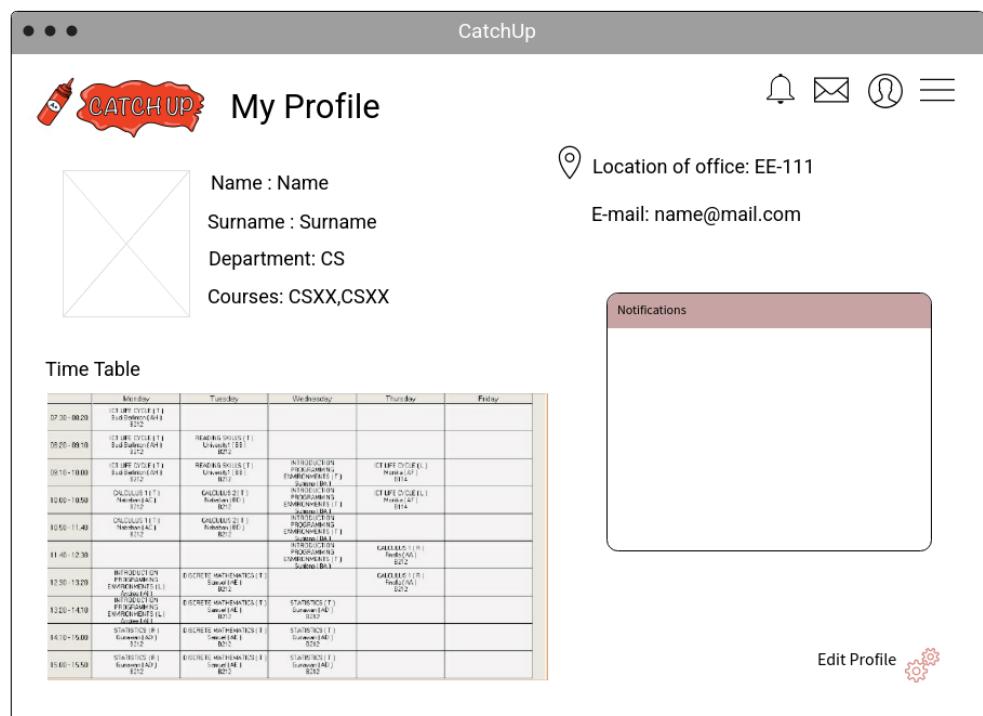
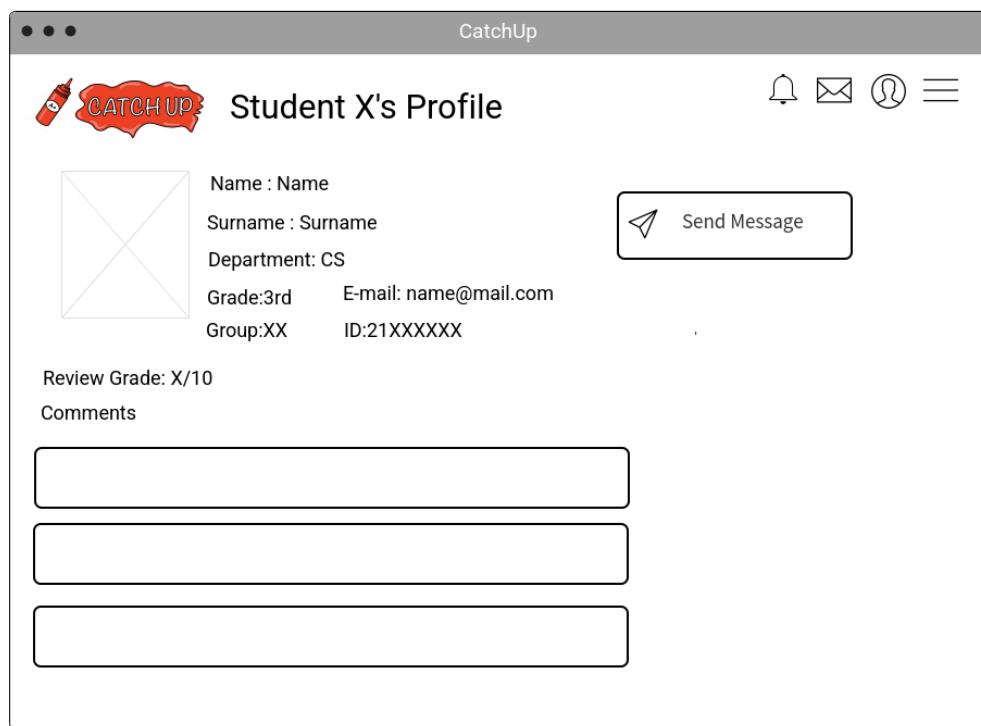


Figure 33

#### 5.4.6 Profile Pages

Users will have their unique profiles according to the information they entered while signing in. The instructor, teaching assistant and student profiles are different from one another. For instance in student profile s/her comments and peer review will be shown. In the instructor profile there will be a time table to see their office hours or make an appointment.



**Figure 34**

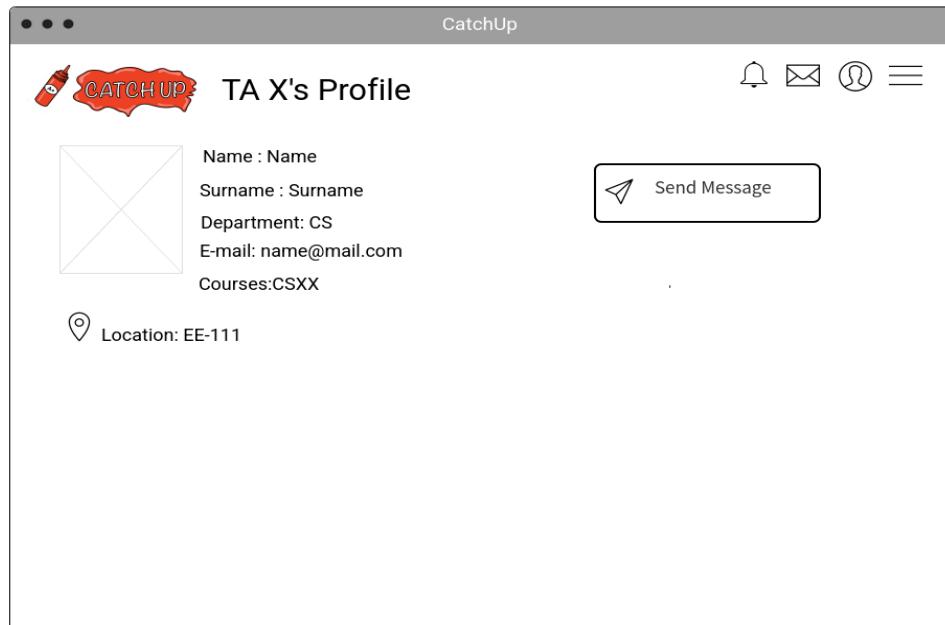


Figure 35

CatchUp

**Instructor X's Profile**

Name : Name  
Surname : Surname  
Department: CS  
E-mail: name@mail.com  
Courses:CSXX

Location: EE-111

	Monday	Tuesday	Wednesday	Thursday	Friday
07:30 - 08:20	(CT LIFE CYCLE (1)) EndSemester 1				
08:20 - 09:10	(CT LIFE CYCLE (1)) EndSemester 1	READING SKILLS (1) Unknown (EE-111)			
09:10 - 10:00	(CT LIFE CYCLE (1)) EndSemester 1	READING SKILLS (1) Unknown (EE-111)	INTRODUCTION TO COMPUTER PROGRAMMING ENVIRONMENT (1)	(CT LIFE CYCLE (1)) Math 101	
10:00 - 10:50	CALCULUS 1 (1) Numerical 1	CALCULUS 2 (1) TakeHome 1 (EE-111)	INTRODUCTION TO COMPUTER PROGRAMMING ENVIRONMENT (1)	(CT LIFE CYCLE (1)) Math 101	
10:50 - 11:40	CALCULUS 1 (1) Numerical 1	CALCULUS 2 (1) TakeHome 1 (EE-111)	INTRODUCTION TO COMPUTER PROGRAMMING ENVIRONMENT (1)	(CT LIFE CYCLE (1)) Math 101	
11:40 - 12:30			INTRODUCTION TO COMPUTER PROGRAMMING ENVIRONMENT (1)	CALCULUS 1 (1) Math 101	
12:30 - 13:20	INTRODUCTION PROGRAMMING ENVIRONMENT (1)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	STATISTICS (1) Unknown (EE-111)	CALCULUS 1 (1) Math 101	
13:20 - 14:10	INTRODUCTION PROGRAMMING ENVIRONMENT (1)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	STATISTICS (1) Unknown (EE-111)	CALCULUS 1 (1) Math 101	
14:10 - 15:00	STATISTICS (1) Unknown (EE-111)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	STATISTICS (1) Unknown (EE-111)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	
15:00 - 15:50	STATISTICS (1) Unknown (EE-111)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	STATISTICS (1) Unknown (EE-111)	DISCRETE MATHEMATICS (1) Unknown (EE-111)	

Figure 36

## **5.4.7 Course Pages**

Users will be shown different kinds of course pages according to their profile abilities but they all see weekly materials and project groups through these pages.

### **5.4.7.1 Student Course Pages**

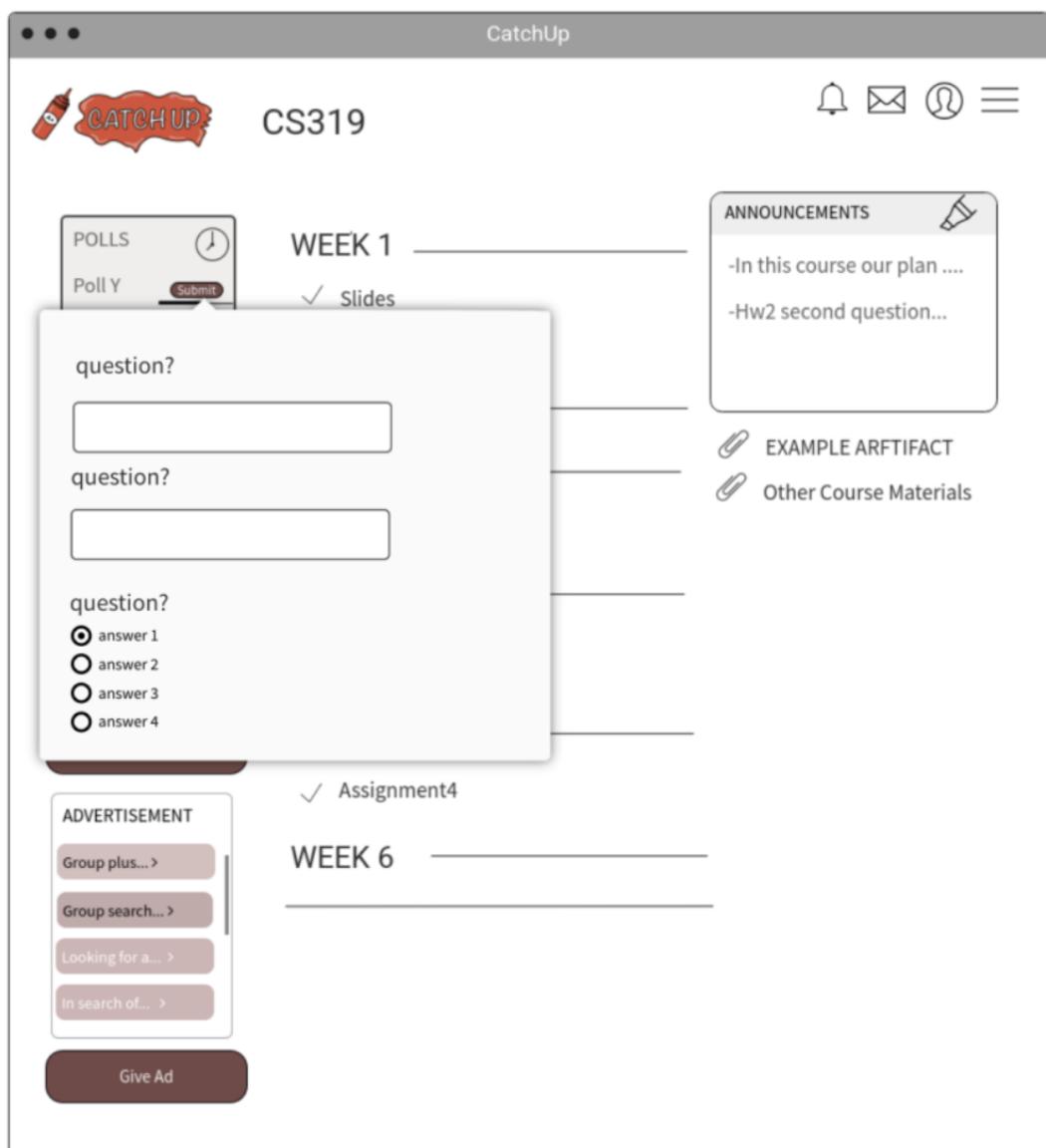
As in these pages which are for students, there will be a request extension button,a request group formation button and a give ad button. Polls above the page will be clickable and when clicked there will be a pop up window in which questions will be shown. Syllabus, example artifact and other materials will be clickable and they forward students to pdf links. Due assignment will be shown in a list form with radio buttons to choose which of them extension requests are for. Advertisements will be clickable and group links will also be clickable.

The screenshot shows the 'CatchUp' mobile application interface. At the top, there is a header bar with three dots on the left and the word 'CatchUp' on the right. Below the header is a decorative section with a red paintbrush icon and the word 'CATCH UP' in a bubbly font.

The main content area is titled 'CS319'. On the far right of the header are icons for a bell, envelope, profile, and menu.

The interface is organized into several sections:

- POLLS**: A box containing 'Poll Y' with a 'Submit' button and 'Poll X' below it.
- PROJECT GROUPS**: A box listing '1A >', '1B >', '2C >', and '4A >'. Below it is a button labeled 'Request Group Formation'.
- ADVERTISEMENT**: A box listing 'Group plus... >', 'Group search... >', 'Looking for a... >', and 'In search of... >'. Below it is a button labeled 'Give Ad'.
- ANNOUNCEMENTS**: A box containing the text: '-In this course our plan ....' and '-Hw2 second question...'. It also features a pencil icon.
- WEEKLY LOGO**: Large horizontal bars representing weeks, each with a small icon and a descriptive text:
  - WEEK 1: ✓ Slides, ✓ Notes
  - WEEK 2: (empty)
  - WEEK 3: ✓ Assignment2, ✓ Assignment3
  - WEEK 4: ✓ Notes, ✓ Slides
  - WEEK 5: ✓ Assignment4
  - WEEK 6: (empty)
- EXAMPLE ARTIFACT**: A section with a paperclip icon and the text 'Other Course Materials'.



**Figure 38**

#### 5.4.7.2 Instructor Course Pages

Instructors will see groups through this page and assign assignments through this page by pressing the plus button and choosing add assignment. they can also add other materials through this plus button above each week.they can create polls through this page and see the results. They can add announcements through clicking the pencil sign and they can add syllabus and example artifacts by clicking the

paperclip sign. They can edit the assignments by choosing one of the above in the box.

The screenshot shows the CatchUp application interface for a course titled "CS319".

**Left Side:**

- POLLS:** Shows two polls: "Poll Y" (radio button) and "Poll X" (radio button). Buttons: "Create New Poll" and "remove poll".
- PROJECT GROUPS:** Lists four groups: "1A", "1B", "2C", and "4A".
- SEE STATS:** Includes a bar chart icon.

**Top Center:** Course title "CS319".

**Top Right:** Icons for Bell, Mail, User, and More.

**Middle Section:** A timeline from "WEEK 1" to "WEEK 6". Each week has a list of completed items (marked with a checkmark and a green checkmark icon) and an "Add" button (a plus sign inside a circle).

- WEEK 1:** Slides, Notes. Add buttons: "Add Assignment" and "Add other content".
- WEEK 2:** No items listed.
- WEEK 3:** Assignment2, Assignment3. Add button.
- WEEK 4:** Notes, Slides. Add button.
- WEEK 5:** Assignment4. Add button.
- WEEK 6:** No items listed.

**Right Side:**

- ANNOUNCEMENTS:** Buttons: "urse our plan ...", "nd question...".
- EXAMPLE ARFTIFACT:** Buttons: "Other Course Materials", "SHARE COURSE CONTENT".
- ASSIGNMENTS:** List: Assignment2 (radio button), Assignment3 (radio button), Assignment4 (radio button selected). Button: "see dealine reqs".

Figure 39

### 5.4.7.3 Teaching Assistant Course Pages

Teaching assistants will see project groups, polls, announcements and other materials through this page. However, they can only edit the assignments by choosing one of the assignments given in the box in **Figure 39**.

The screenshot shows a course page titled "CatchUp" for "CS319". On the left, there's a "CATCH UP" section with a paintbrush icon. Below it is a "POLLs" box containing "Poll Y" and "Poll X". To the right, there are six weekly sections labeled "WEEK 1" through "WEEK 6", each with a plus sign (+) at the end. "WEEK 1" lists "Slides" and "Notes". "WEEK 2" has no listed items. "WEEK 3" lists "Assignment2" and "Assignment3". "WEEK 4" lists "Notes" and "Slides". "WEEK 5" lists "Assignment4". "WEEK 6" also has a plus sign (+) at the end. On the far left, there's a "PROJECT GROUPS" sidebar with four items: "1A", "1B", "2C", and "4A". A "Form groups" button is located below this sidebar. In the top right corner, there are icons for a bell, envelope, user profile, and menu. To the right of the weekly sections, there are three boxes: "ANNOUNCEMENTS" (listing "In this course our plan ...." and "Hw2 second question..."), "EXAMPLE ARFTIFACT" (with a pencil icon), and "Other Course Materials" (with a clipboard icon). At the bottom right, there's an "ASSIGNMENTS" box containing "Assignment2" (radio button), "Assignment3" (radio button), and "Assignment4" (radio button, marked with a dot).

**Figure 40**

#### 5.4.8 Group Pages

There will be two kinds of group pages. One of them will be for group members and the other one is for non-members to give feedback and grade. In the first type of group page which is for group members, they can submit assignments, arrange group meeting dates, see their feedback given to their submitted artifacts, and access peer reviews and grades. In the second type of group page which is for non-members, users can give feedback to artifacts uploaded by the group members.

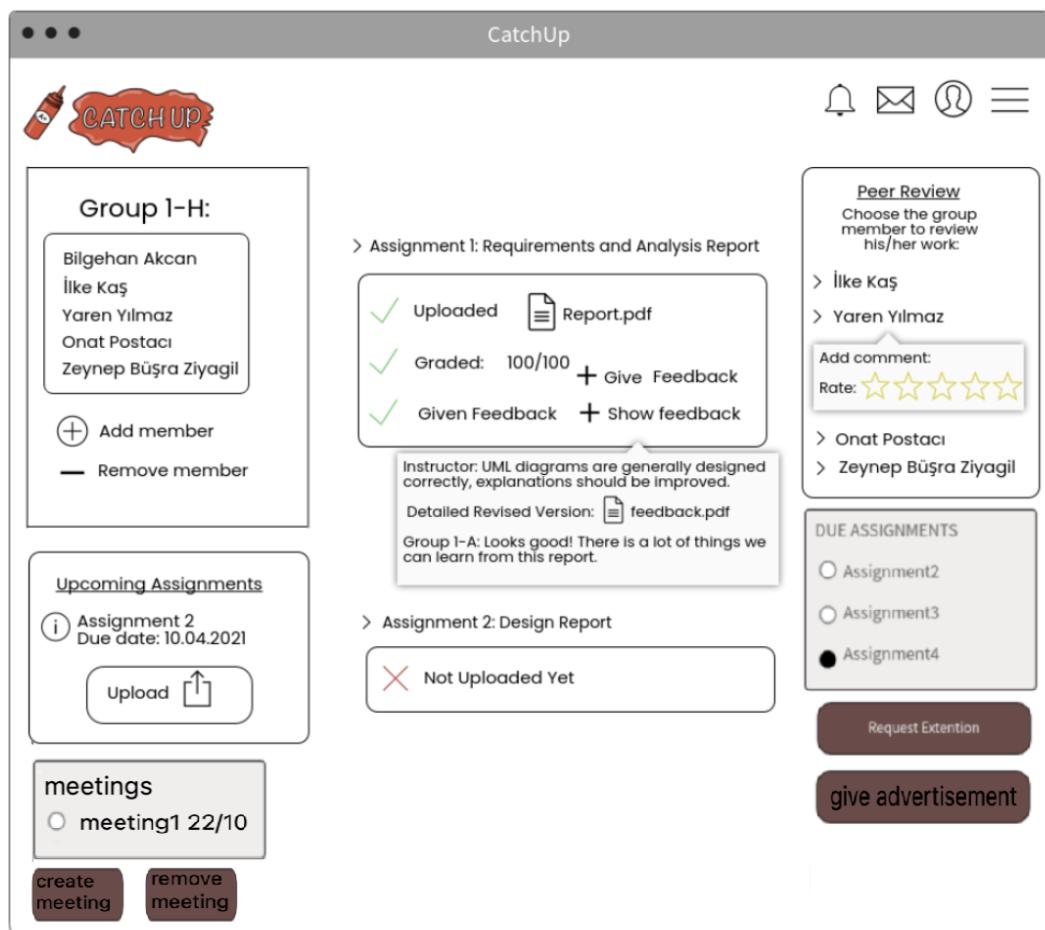
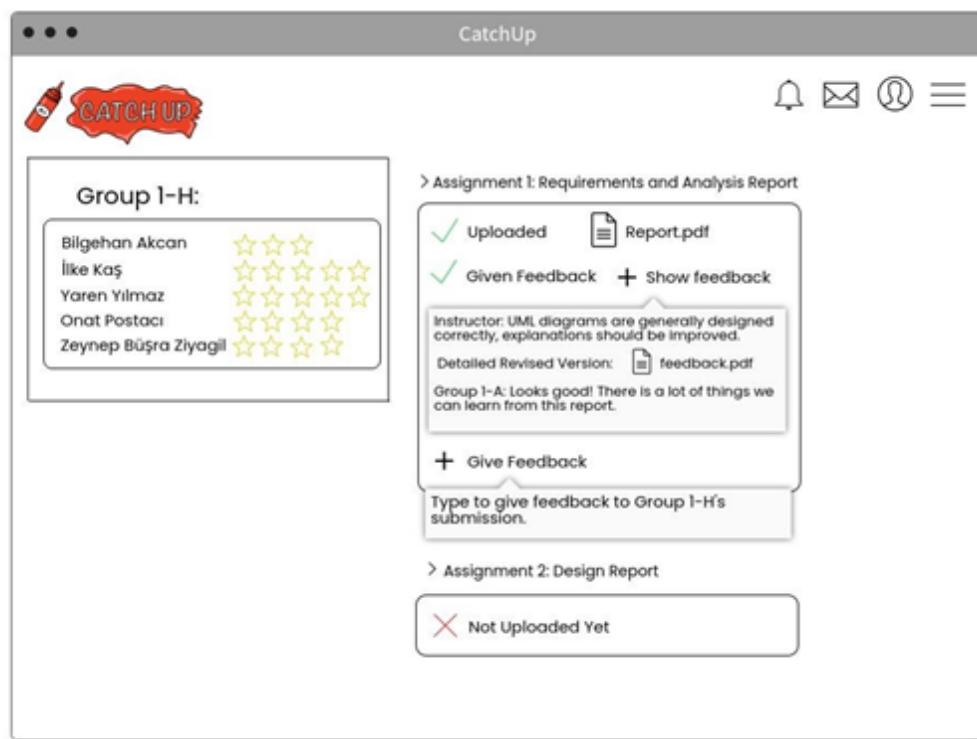


Figure 41



**Figure 42**

## 6 Conclusion

*CatchUp* is a web-based application that helps instructors, teaching assistants and students to interact with each other. Also, it helps students to form groups for their projects and enables them to rate each other at the end of the project thanks to peer review feature.

*CatchUp* suggests many features for different kinds of users. It enables instructors to create different course pages, assign projects and assignments, give feedback to an submitted artifact, grade assignments, create polls and share with students online, and see assignment statistics. *CatchUp* allows teaching assistants to select and enroll in course pages, grade and give feedback to artifacts, accept or deny group formation

requests and manage group formation processes. Finally, it enables students to select and enroll in course pages, upload artifacts, give rates and feedback to group members and review peers' works, send messages to group members, and form groups.

In order to organize the project and make the implementation part easier, several types of UML diagrams are used. To show the object model of the project, a class diagram is drawn and given above. To show the functional model of the project, a use case diagram is drawn and given above. Also, to show the dynamic behavior of objects that take part in the project, state diagrams, sequence diagrams and activity diagrams are drawn and given above. All these diagrams make the project more understandable for both stakeholders and users who want to know the context.

## 7 References

- [1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.
- [2] Produle, “Wireframe Tools, Prototyping Tools, UI Mockups, UX Suite, Remote designing,” *MockFlow*. [Online]. Available: <https://www.mockflow.com/>. [Accessed: 23-Mar-2021].
- [3] “Peer review,” *Wikipedia*, 07-Apr-2021. [Online]. Available: [https://en.wikipedia.org/wiki/Peer\\_review](https://en.wikipedia.org/wiki/Peer_review). [Accessed: 23-Apr-2021].