



Bilkent University

Department of Computer Engineering

---

# Object Oriented Software Engineering Course Project

*BilHub: A Classroom Helper*

## Analysis Report

Barış Ogün Yörük, Halil Özgür Demir, Mustafa Çağrı Durgut, Yusuf Miraç Uyar, Aybala Karakaya,  
Oğuzhan Özçelik

Instructor: Eray Tüzün

Teaching Assistant(s): Elgun Jabrayilzade, Erdem Tuna, Barış Ardiç

Progress Report

March 16, 2021

## Contents

1.	Introduction	1
2.	Overview	2
2.1.	Users	2
2.1.1.	Instructor	2
2.1.2.	Teaching Assistant (TA)	2
2.1.3.	Student	2
2.2.	Courses	3
2.3.	Sections	3
2.4.	Project Groups	3
2.5.	Assignments	4
2.6.	Reviewing	4
2.6.1.	Assignment and Project Reviewing	4
2.6.2.	Peer Reviewing	5
2.7.	Statistics	5
3.	Functional Requirements	5
4.	Non-Functional Requirements	6
5.	Pseudo Requirements	6
6.	System Models	7
6.1.	Use-Case Diagram	7
6.1.1.	Use-Case Descriptions	8
6.1.1.1.	Create Course and Add Participants	8
6.1.1.2.	Forming Groups	9
6.1.1.3.	Giving and Turning Assignments	10
6.1.1.4.	Giving and Seeing Feedback	12
6.1.1.5.	Showing and Editing Statistics	13
6.1.1.6.	Assigning and Doing Peer Grading	13
6.2.	Dynamic Diagrams	15
6.2.1.	Sequence Diagrams	15
6.2.1.1.	Creating Course and Adding Instructor/TA WF	15
6.2.1.2.	Adding Students to the Course Workflow	16
6.2.1.3.	Forming Groups	18
6.2.1.3.1.	Sending Join Group Request Workflow	18
6.2.1.3.2.	Join Request Answer Workflow	18
6.2.1.3.3.	Merging Groups Workflow	19
6.2.1.3.4.	Finalizing Group Workflow	20
6.2.1.4.	Assignments	21
6.2.1.4.1.	Giving Assignment Workflow	21

6.2.1.4.2.	Submitting Assignment Workflow	22
6.2.1.5.	Feedback	23
6.2.1.5.1.	Instructor Giving Feedback Workflow	23
6.2.1.5.2.	Student Giving Feedback Workflow	24
6.2.1.5.3.	Student Seeing Feedback Workflow	25
6.2.1.5.4.	Instructor Seeing Feedback Workflow	26
6.2.1.6.	Peer Grading	27
6.2.1.6.1.	Giving Peer Grade Assignment Workflow	27
6.2.1.6.2.	Performing Peer Grading Workflow	27
6.2.2.	Activity Diagrams	28
6.2.2.1.	Creating Courses and Adding Participants	28
6.2.2.2.	Forming Groups	29
6.2.2.3.	Giving and Turning in Assignments	30
6.2.2.4.	Giving Feedback	31
6.2.2.5.	Viewing Feedback	32
6.2.2.6.	Creating Statistics	32
6.2.2.7.	Peer Grading	33
6.2.3.	State Machine Diagrams	33
6.3.	Object and Class Models	36
6.4.	User Interface	37
6.4.1.	Navigation Bar	37
6.4.2.	Main Page	39
6.4.3.	Projects Page	41
6.4.4.	Project's Assignment Page	46
6.4.5.	Profile Page	51
6.4.6.	Courses Page	54
6.4.7.	Course's Assignment Page	60
6.4.8.	Create New Course	63
6.4.9.	Course Settings	65
6.4.10.	Settings	66
6.4.11.	Notification Page	68
7.	Glossary & References	72

# Analysis Report

*BilHub: A Classroom Helper*

## 1 Introduction

If you ever had a course that includes a group project, you are most probably familiar with this cult meme based on a scene from *Hangover*. Most of the time group projects are made by only one or two of its participants. The rest of the team meet with the project while they are presenting it to the instructor. Even if you did gather the “dream team”, running after one deadline or revision to another is more than frustrating and might make you miss the whole point of the project you build. What if there was a way to make this process less frustrating and more maintainable while not compromising the mental health of the team?

Well, now there is **BilHub**. BilHub is a comprehensive classroom helper which you can use with the courses that include a project, especially suitable for group projects. BilHub will make life easy for you whether you are a student, teaching assistant or the instructor of the course. You will have the ultimate overview of the progress you or your students are making, make the deadlines more catchable and the feedback more impactful.

Are you a person who doesn't have many friends and have no idea who to group with? Or an instructor who couldn't decide how to partition students into groups without there are no over or underpopulated ones? No worries, we got you covered as well. BilHub reinvents the group selections with a brand new way to form project groups with a maximum number of satisfied student and homogen sized groups. BilHub uses a technique we call **pseudogroups** in order to store the information of clustered students and join these clusters to form groups with an optimized algorithm.

Another key feature of BilHub is that it makes you learn from other groups' mistakes. Most of the time implementers of a project are so oriented with the details of the work they are pursuing that they can miss some of the main problems or vague parts of it. As an instructor you can allow your groups to review other people's work and give feedback to it. We believe that this feature will allow students to see what it's like to be in grader's shoes so that they will look at the work they have done more critically. Moreover you can assign your students to review their own project partners privately to detect any slackers in the

project. Overall, BilHub is the software you need if you are instructing a course which includes a group project.

## **2 Overview**

### **2.1 Users**

There are three types of user in BilHub; instructor, teaching assistant, and student. Each user has a profile which shows his/her previous courses and groups ( empty if there is no previous course ), information about himself/herself ( added by users himself/herself ). Different users may have their specific information in that page ( For example, assistants have courses that they are assistant of and instructors have their previous courses that they had given.). Each of them has different authority over each project.

#### **2.1.1 Instructor**

Instructors users have authority to create courses and assign teaching assistants and other instructors to courses. They can also do activities that teaching assistants can do. These activities are editing group formation settings( For example changing due date for group formations or changing maximum and minimum number of group members), adding new students to courses, assigning new assignments to project groups ( or individuals which are project groups consisting of one member), editing existing assignments (For example opening assignments for reviewing from other groups or fixing mistakes made while creating that assignment.) or opening peer reviewing inside groups.

#### **2.1.2 Teaching Assistant (TA)**

Teaching assistants are roles specific to courses. TAs are students who are authorized by instructors. They are able to perform new activities only in courses where they are authorized as TA. They can edit group formation settings, add new students to courses, assign new assignments to project groups, edit existing assignments or open peer reviewing inside groups.

#### **2.1.3 Student**

Students are common users in BilHub who can join courses, form groups and deliver assignments. Students can only enroll in a course if the instructor or TA registers students mail to the course page. When a student enrolled in a course, they attained to a one-person-pesude group by the system. They can form groups with other students in their sections until the due date which is called “the lock date” of group formation (for further details about

groups, see 2.4 “Project Groups”). While forming groups, they can view other students’ pages in order to get information. This will help students to choose group members that they can get along with. When groups are formed, they can wait for assignments. When assignments are posted, according to assignment type students can upload links, files or peer reviews(a type of assignment) into that assignment. Students can also view and review other groups’ assignments if the assignment is visible to them (setted by instructor or TA).

## 2.2 Courses

In BilHub, everything is based on courses. The new courses are added by the instructor users with 3 keywords: course code, year, and semester. For example CS319 2020-2021 Spring Semester will be a different course object than CS319 2020-2021 Fall Semester. Every course will have at least one instructor. An instructor can add other instructors and teaching assistants to the course with a role (being TA or instructor) assigned to them. Instructors and TAs can add students to the course. There will be one or more sections for each course. In addition to that, each course will have a lock deadline for the group formation, which means students will be allowed to form groups until a deadline determined by the instructor and after the deadline, BilHub will automatically form groups while aiming to make most people happy (see “2.4 Project Groups” for additional information about groups).

## 2.3 Sections

Whether there will be different sections or not is determined by the instructor’s decision. If a group formation between different STARS sections is allowed, the instructor will open only one BilHub section and all the students will belong to that section. If there is no permission to form a group with students from different STARS sections, the instructor is expected to open multiple BilHub sections and assign each student to one of the sections (this can be done according to STARS sections or the decision of the instructor). On the other hand, students from the same course but different BilHub sections will still be allowed to grade other sections’ projects.

## 2.4 Project Groups

Project groups are the groups formed by the students from the same BilHub sections (if they are allowed to form groups from different STARS sections, the instructor will open only one BilHub section, thus they will be in the same BilHub section). For the project groups, BilHub uses a technique we like to call **pseudogroups**. For a course, there will be a two deadline dates determined by the instructor which are called “add-drop date” and “the lock

date”. Initially, all students will belong to individual pseudogroups. Until the “add-drop date” students can join or merge with other pseudogroups. Those groups will be an “unformed group” which means it is not official yet. With the arrival of the “add-drop date” the number of groups with allowed numbers of students will be determined. After that, pseudo groups will be able to finalize their group if they fit the course restrictions. If every member of the group confirms the final group, the group will be a “formed group”. This process may only happen between “add-drop date” and “the lock date”. After the lock date, BilHub will use an algorithm to combine unformed groups. For example, if there are unformed three groups with two students each, and the course expects groups with three students each, BilHub will split one of the groups and assign each to different groups so that there will be two groups with three students each.

## **2.5 Assignments**

After creating project groups, TAs and instructors can assign different kinds of assignments to all project groups. When one assignment is assigned to a section, all the groups in that section are assigned with this assignment. Each assignment has different properties. Assignments may be reviewable( by students from other groups), deliverable (by using link or document), peer reviewing (peer reviewing between group members count as a different kind of assignment.). Also assignments can be altered during their lifetime. For example, until the due date of assignment, this assignment may only be visible to the group, however, after the due date, the instructor or TA can open assignment to review and other teams’ members can start to review this assignment. Students can submit assignments before the due date, provided that the file that they try to submit is supported by the system and the file size is within a specified limit.

## **2.6 Reviewing**

### **2.6.1 Assignment and Project Reviewing**

There will be a comments section for each group page and for their assignments. From that section, Instructors and TA’s can give feedback and grade both project groups and their assignments.

Instructors and TA’s can allow students to review other groups’ assignments and projects. The assignments and projects that can be reviewed are called “reviewable”. Students can review other groups’ assignments and projects by grading it and adding their

comments if the assignment or the project is reviewable. Whether they can only grade, only comment, or can do both are up to instructors and TAs.

### **2.6.2 Peer Reviewing**

Peer reviewing is an assignment type in which group members evaluate each other. Instructors and TAs can assign a peer reviewing assignment in the same way as other assignments. (see “2.5 Assignments” for additional information about assignments). Students can grade and comment about each other in peer reviewing assignments. Whether they can only grade, only comment, or can do both are up to instructors and TAs.

### **2.7 Statistics**

There will be a statistics section for each group page and for their assignments. From that section, students can see their grade statistics with different parameters. There will be several types of statistics charts and their visibility will be decided by the TA's and Instructors. The data of those charts (grades etc.) will be automatically taken from the comments section.

Example statistics chart:

- Grade comparison with other groups (for specific assignment or whole project)
- Average grade for the assignment
- Grade comparison between the assignments given to a specific group
- Average grades for different groups; Instructor, TA's, Student

## **3 Functional Requirements**

- Instructors can create courses.
- Instructors can add new instructors and TAs to the course.
- Instructors and TAs can add new students to the course.
- Instructors and TAs can grade student projects or specific project assignments.
- Instructors and TAs can comment on student projects or specific project assignments.
- Instructors and TAs can give new assignments to students and edit the existing ones.
- Instructors and TAs can choose which statistics are shown for a specific assignment.
- Instructors and TAs can make an assignment visible or invisible for the students.
- Instructors and TAs can make an assignment gradable by students.
- Instructors can determine the group size boundaries for the course project.



- Students can form groups. While doing this, they can join a group or merge their group with another.
- Students can indicate that their groups are finalized for the term and that no one can leave or join the group.
- Students can leave the groups if they are not finalized yet.
- Students can add information to their profile and change this information.
- Instructors and TAs can allow students to comment on a project or project assignment.
- Students can comment on projects and grade projects (both their project and others' projects).
- Students can grade other groups' project assignments.
- Students, TAs and instructors can view other students' profiles.
- Students can submit assignments.
- Students can set project information for their groups.

#### **4 Non-Functional Requirements**

- Any user who is familiar with the Moodle system should be able to use BilHub without difficulties [Usability Requirement].
- System crash does not cause data loss [Reliability Requirement].
- Any person who is not enrolled in Bilkent University cannot register the system [Reliability Requirement].
- System does not accept submissions bigger than 1 GB [Reliability Requirement].
- Up to 2000 people can be online at the same time [Performance Requirement].
- Servers should be online 7/24 during semester time [Performance Requirement].
- System continues to work autonomously through every different semester without needing a patch for every semester [Supportability Requirement].

#### **5 Pseudo Requirements**

- Must use object oriented programming language while implementing [Implementation Requirement].

## 6 System Models

### 6.1 Use-Case Diagram

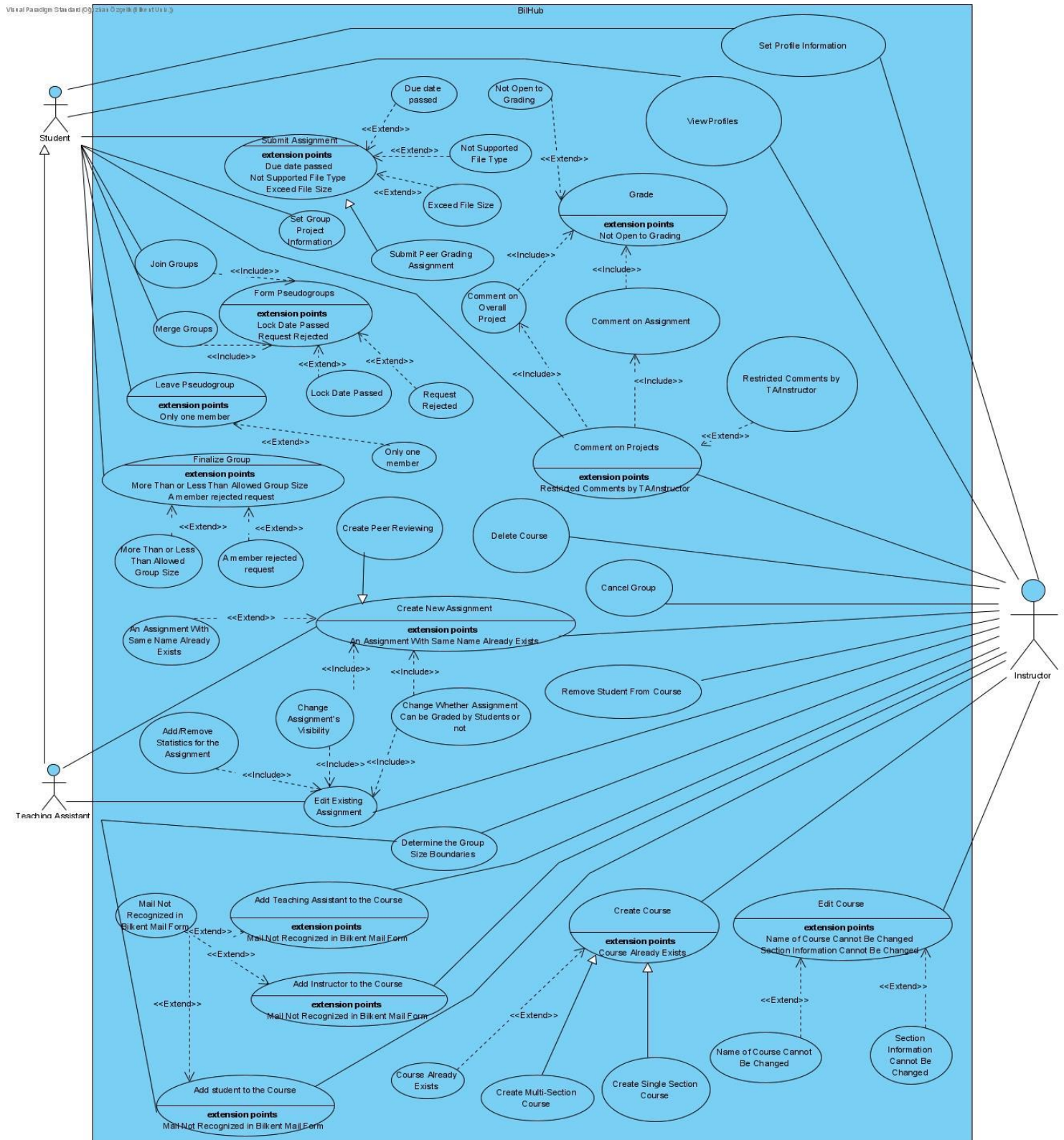


Figure 1: Use-Case Diagram

## 6.1.1 Use Case Descriptions

### 6.1.1.1 Use Case Name: Create Course and Add Participants

**Participating Actor:** Instructor, Teaching Assistant

**Stakeholders and Interests:** Instructor wants to create a new course and add TAs and students to that course.

**Entry Condition:** User must be an instructor to create a course.

**Exit Condition:** Instructor successfully adds all the students

#### Main Flow of events:

1. Instructor starts creating a course
2. Instructor specifies number of sections.
3. Instructor adds Instructors to the course.
4. Instructor adds TAs to the course.
5. Instructor adds students to the course.
6. Instructor defines group sizes for course.
7. TA adds additional students to the course.
8. All the participants are added to the course.

#### Alternative Flow of events:

- 2.1 System cannot recognize given mail.
  - a) Instructor fixes typo in mail.
  - b) Instructor adds TA to the course.
- 3.1 System cannot recognize given mail.
  - a) Instructor fixes typo in mail.
  - b) Instructor adds students to the course.
- 3.2 TA adds students to the course.
- 3.3 Given mail cannot be found by system.
  - a) Instructor fixes typo in mail.
  - b) Instructor adds student to the course.
- 5.1 Instructor changes group sizes for the course.

### 6.1.1.2 Use Case Name: Forming Groups

**Participating Actor:** Student

**Stakeholders and Interests:** Student need to form a group when they join a new course.

**Entry Condition:** Student should join a course.

**Exit Condition:**

- Lock date of group formation has come.
- Group is finalized by the co-decision of all the group members.

**Main Flow of events:**

1. Student sends a join request to another pseudogroup which consists of 1 student.
2. Student in that pseudogroup is notified about the join request .
3. That student accepts the join request and their pseudogroup size becomes 2.
4. Student sends a merge request to another pseudogroup with size of 2.
5. All students in both pseudogroups (except who sent request) are notified about merge request.
6. All of students who are notified and accepted the merge request.
7. Pseudogroups are merged.
8. All students in the pseudogroup are notified about the join request by another student.
9. All students in the pseudogroup accepted the join request and that student joined the group.
10. All students in the pseudogroup set their statement ready to form a group.
11. Group formed.

**Alternative Flow of events:**

- 3.1 Student declines the join request.
  - a) The student sends a join request to another group with size of 5.
  - b) That group finalizes their group before accepting a join request.
  - c) The student is notified about this.
  - d) One group sends a merge request to that student ( as his/her group consists of one member.).
  - e) All of the members in each group accept the merge request (except the one who sends it.).
  - f) Groups are successfully merged.

- g) Lock date comes and the group automatically finalizes as the number of group members satisfy the restrictions.

6.1 One of the students declines the merge request.

- a) Lock date comes before they reach the sufficient number of group members.
- b) There are two groups who need one for each member in order to reach a sufficient number of group members. So, the group with two members split in half and each of them is sent to one group.
- c) Groups are finalized.

8.1 One of the students declines the join request.

- a) Lock date comes before they reach the sufficient number of group members.
- b) As they need 1 member in order to reach the sufficient number of members, that member is given automatically by the system.
- c) Group is finalized.

9.1 Students cannot finalize group as they exceed the maximum number of group members.

- a) Lock date comes.
- b) One random student's group is changed by system.
- c) Group has sufficient number of students now.
- d) Group is finalized.

### **6.1.1.3 Use Case Name:** Giving and Turning in Assignments

**Participating Actor:** Student, Instructor

**Stakeholders and Interests:**

- Instructor wants to give an assignment.
- Student wants to turn in the assignment.

**Entry Condition:** Instructor and students are enrolled in the course.

**Exit Condition:** The assignment has been closed to submission.

**Main Flow of events:**

1. Instructor creates a new assignment.
2. Students are notified about the assignment.
3. Instructor makes the assignment gradable by other students.
4. Instructor adds statistics for the assignment.

5. Instructor edits a typo on the assignment description.
6. Student submits the assignment.
7. Last day of submission comes.

**Alternative Flow of events:**

- 1.1 Instructor cannot create the assignment because an assignment with the same name already exists.
  - a) Instructor creates an assignment with a different name.
  - b) Student submits the assignment.
  - c) Last day of submission comes.
- 3.1 Instructor does not allow other students to grade the assignment.
  - a) Student submits the assignment.
  - b) Last day of submission comes.
- 4.1 Instructor removes the statistics for the assignment.
  - a) Student submits the assignment.
  - b) Last day of submission comes.
- 6.1 Due date of the assignment comes.
  - a) Student tries to submit the assignment but s/he cannot because the last day of submission comes..
- 6.2 Student tries to submit the assignment but s/he cannot because the file type that s/he tries to submit is not supported.
  - a) Student submits another file with a supported type.
  - b) Last day of submission comes.
- 6.3 Student tries to submit the assignment but s/he cannot because the size of the file that s/he tries to submit exceeds the limit.
  - a) Student submits another file within the limits.
  - b) Last day of submission comes.
- 6.4 Instructor makes the assignment invisible.
  - a) Instructor makes the assignment visible again.
  - b) Student submits the assignment.
  - c) Last day of submission comes.

#### **6.1.1.4 Use Case Name:** Giving and Seeing Feedback

**Participating Actor:** Student, TA, Instructor

**Stakeholders and Interests:**

- Instructor and TA want to give feedback and grade to an assignment.
- Student wants to see the feedback given for the assignment and give feedbacks to other students.

**Entry Condition:**

- Student has submitted the assignment.
- Due date has passed.

**Exit Condition:** Feedback submitted.

**Main Flow of events:**

1. TA opens assignment to comments.
2. TA comments on the assignment and grades it.
3. Instructor comments on the assignment and grades it.
4. Students from other groups view the comments.
5. Students from other groups comment on the assignment.
6. Students in the group view all the comments and grades.
7. Student in the group comment on other group's assignment.
8. TA opens commenting on the overall project.
9. TA changes the project's commenting to anonymous.
10. Students comment on other projects.
11. Student and TA view each comment which are anonymous.

**Alternative Flow of events:**

- 1.1 Instructor opens assignment to comments.
- 5.1 Student cannot grade another student's assignment because this assignment was not allowed to be graded by students.
- 5.2 Student cannot comment on another student's assignment because this assignment was not allowed to be commented by students.

#### **6.1.1.5 Use Case Name:** Showing and Editing Statistics

**Participating Actor:** Instructor, TA

**Stakeholders and Interests:**

- Instructors and TAs may want to see overall statistics of course, project group or assignment.

**Entry Condition:** There should be already graded assignments or projects.

**Exit Condition:**

- Course is closed
- Showed statistic is deleted

**Main Flow of events:**

1. TA of the course grades all the assignments that groups are submitted.
2. TA opens a new statistic in the course page in order to see the histogram of grades.
3. Instructor opens a new statistic in the course page in order to see the table of all the grades in that assignment and average of it.
4. TA grades the second assignment in that course.
5. Instructor edits the table s/he opened in order to show second grades.
6. Instructor gives final grades to each project.
7. Instructor opens a new statistic in the course page in order to see the line chart of all the grades of assignments and projects of every group.
8. Instructor closes the course.

**Alternative Flow of events:**

- 8.1 Instructor closes the first statistic opened by the TA.
  - a) Instructor closes the table s/he opened.
  - b) Instructor closes the line chart s/he opened.

#### **6.1.1.6 Use Case Name:** Assigning and Doing Peer Grading

**Participating Actor:** Student, TA

**Stakeholders and Interests:**

- Some of the grade on the project comes from the peer grading.



**Entry Condition:** Project should be active.

**Exit Condition:**

- All the team members reviewed each other.
- TA or Instructor closed peer reviewing.
- Last day of peer reviewing has come.

**Main Flow of events:**

1. TA opened a peer reviewing assignment in group.
2. Students are notified about this.
3. All of the students give grades to and review each other.
4. Peer reviewing closed.

**Alternative Flow of events:**

- 3.1 Some of the students did not review.
  - a) Last day of peer reviewing comes.
  - b) Peer reviewing closed.
- 3.2 TA closes peer reviewing manually.
- 3.3 TA extends the due date of peer reviewing.
  - a) Most of the students give grades to each other.
  - b) Due date of peer reviewing comes.
  - c) Peer reviewing closed.

## 6.2 Dynamic Diagrams

### 6.2.1 Sequence Diagrams

#### 6.2.1.1 Creating Course and Adding Instructor/TA Workflow

**Scenario:** Instructor A wants to create a course. He can either create a sectionless course or a multisection one. In the first alternative, he creates a course: Course0, he creates one section, Section0 becomes sectionless. Then, he adds Section0 to Course0. Then he adds Instructor A (himself) to Course0 as instructor and Course0 is added into the courses of Instructor A. He adds Instructor B to Course0 as instructor and Course0 is added into the courses of Instructor B. He adds TA to Course0 as teaching assistant and Course0 is added into the instructed courses of TA. In the second alternative, he creates a multi section course (Course1), he creates non-sectionless two sections: Section1 and Section2. After the creation of Section1, he adds this to Course1 and after the creation of Section2, he adds this to Course1. Then he adds Instructor A (himself) to Course1 as instructor and Course1 is added into the courses of Instructor A. He adds Instructor B to Course1 as instructor and Course1 is added into the courses of Instructor B. He adds TA to Course1 as teaching assistant and Course1 is added into the instructed courses of TA.

**Created objects:** Course0 and Section0 in the course is sectionless alternative, Course1 and Section1 in the course has multiple sections alternative.

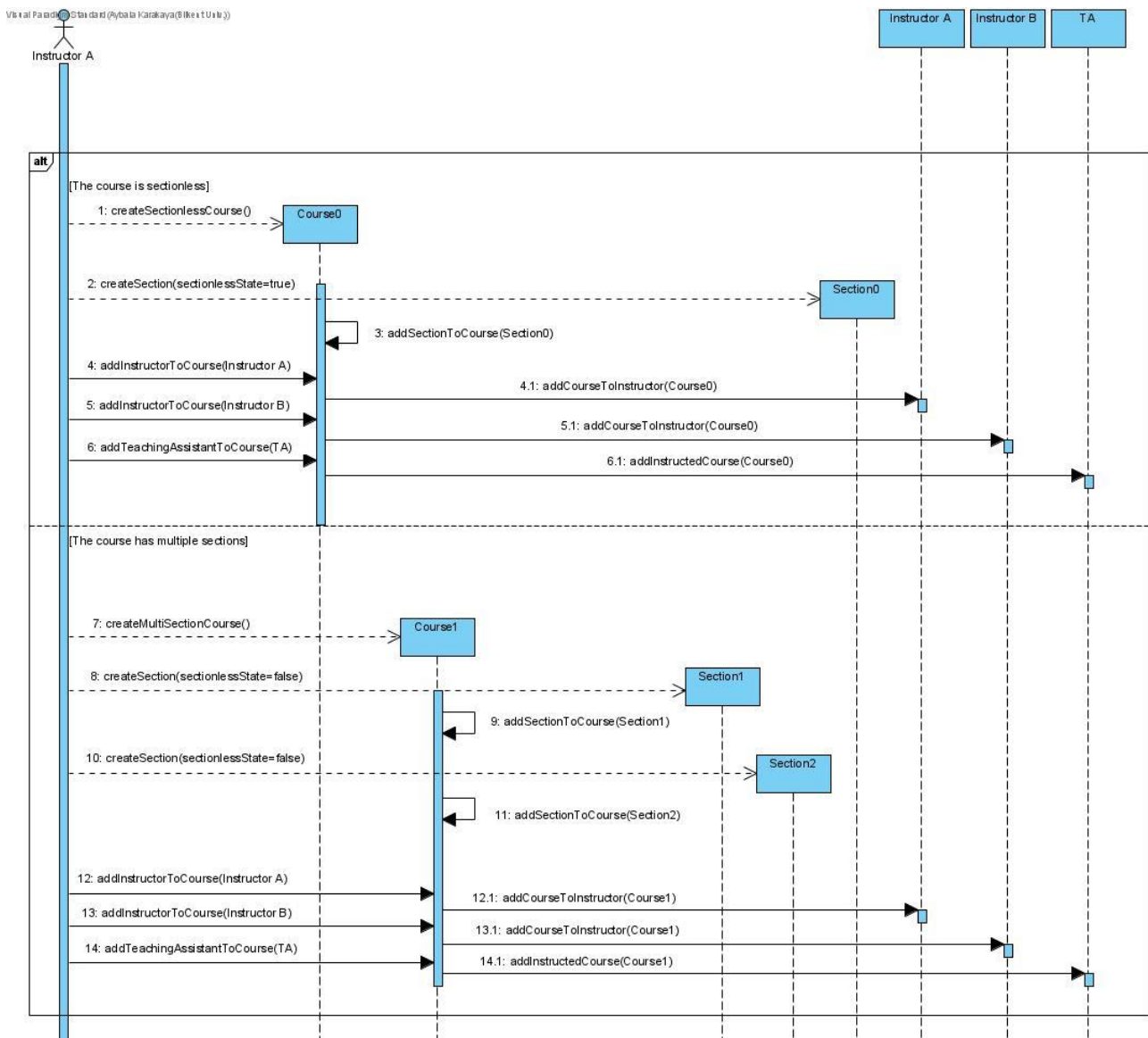


Figure 2: Sequence Diagram for Creating Course and Adding Instructors and TAs Workflow

### 6.2.1.2 Adding Students to the Course Workflow

**Scenario:** Instructor wants to add students to the course. In this scenario, there are two sections: Section1 and Section2. Instructor first submits the mail addresses of students in Section1 in a txt file. Section1 object gets the mail addresses that are not already in the database and the Student objects whose mail addresses are already in the database. Then, Student objects with the mail addresses that are not in the database are created. In this scenario, only Yusuf's mail address is not in the database. So, the Student object for Yusuf is created and Yusuf is added to Section1. After that, Student objects with mail addresses that were already in the database are added to the course. In this scenario, only Barış's mail address was in the database. So, Barış is added to Section1. Then, the instructor submits the mail addresses of students in Section2 in a txt file. Section2 object gets the mail

addresses that are not already in the database and the Student objects whose mail addresses are already in the database. Finally, all the students in Section1 are added to the Course one by one and in the meantime, Course is added to the students' courses. In this scenario, Yusuf is added to the Course and the Course is added to Yusuf. Then, Barış is added to the Course and the Course is added to Barış. Then, Student objects with the mail addresses that are not in the database are created and added to Section2. In this scenario, there are no mail addresses that are not in the database already. After that, Student objects with mail addresses that were already in the database are added to the course. In this scenario, there were no students whose mail address was in the database. So, no one is added

**Created objects:** Yusuf :: Student

UML Farklı Standart (UML) (Yayınla Karakaya (Bilgi UML))

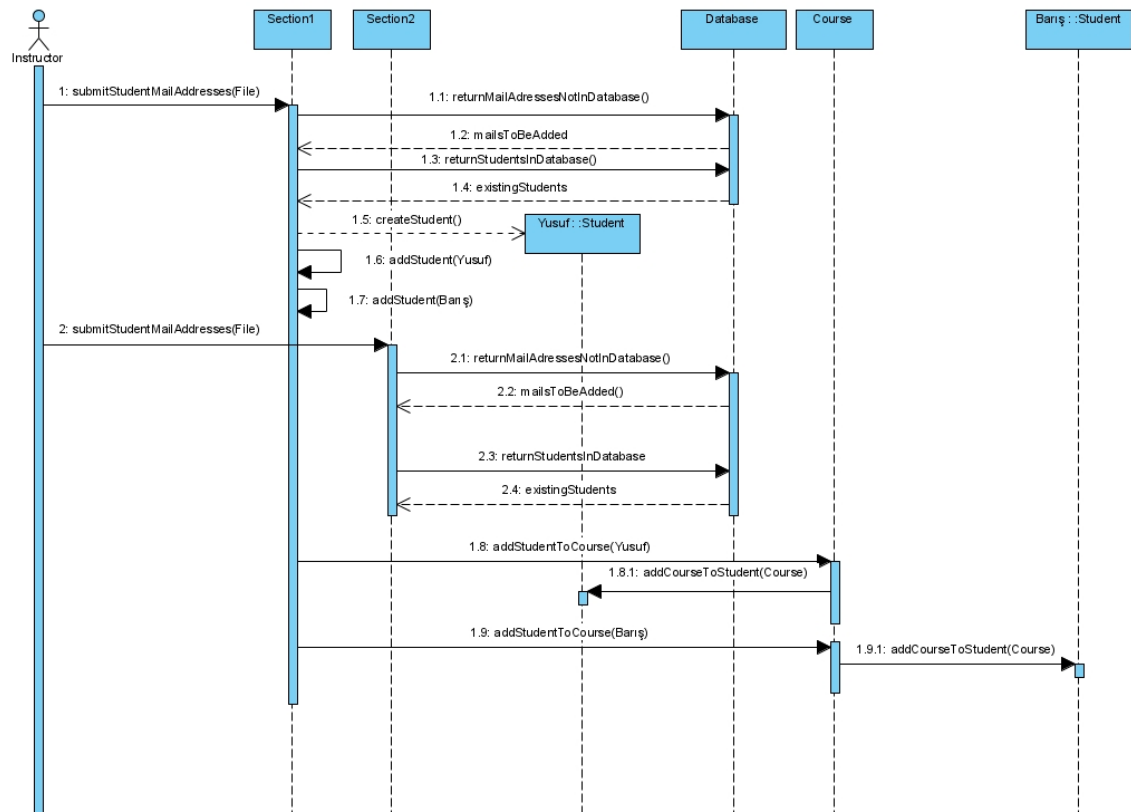


Figure 3: Sequence Diagram for Adding Students to the Course Workflow

### 6.2.1.3 Forming Groups

#### 6.2.1.3.1 Sending Join Group Request Workflow

**Scenario:** Student A wants to join a group, Group1. Group1 currently has one member, Student B. Student A creates a join request to join Group1. After a JoinRequest object is created, it is added to the outgoing join requests list of Student A and to the incoming join requests list of Group1. Then, a join request notification is sent to Student B.

**Created objects:** JoinRequest

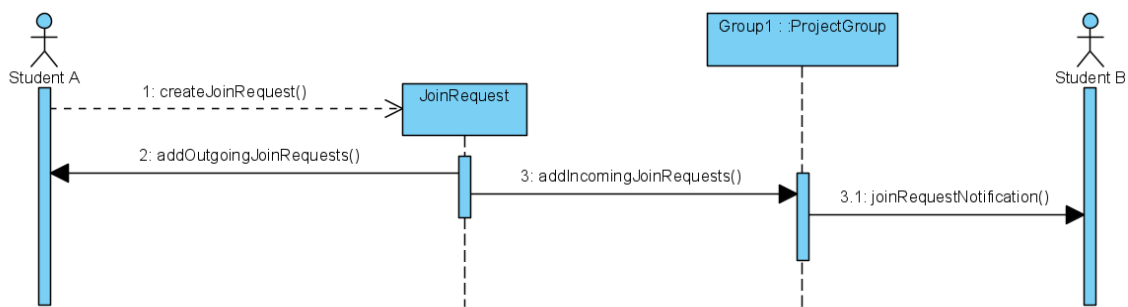


Figure 4: Sequence Diagram for Sending Join Group Request Workflow

#### 6.2.1.3.2 Join Request Answer Workflow

**Scenario:** Student B answers Student A's join request. Two possible scenarios, one for when she accepts the request and one for when she rejects the request are shown in the sequence diagram below. Either way, since there is only one member in Group1, voting finishes. If she accepts the request, Student A gets added to Group1 and then a notification about the answer of the join request is sent to Student A and B. Else, only a notification about the answer of the join request is sent to Student A and B.

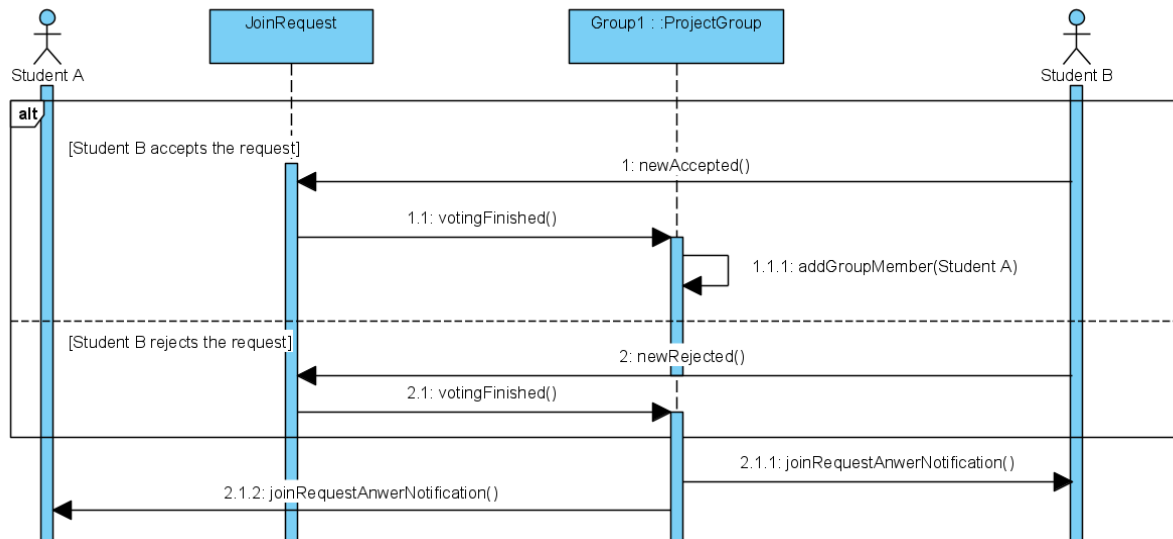


Figure 5: Sequence Diagram for Join Request Answer Workflow

### 6.2.1.3.3 Merging Groups Workflow

**Scenario:** Student A wants her group, Group1 to merge with Group2. Group1 has 2 members: Student A and Student B. Group2 has 2 members: Student C and Student D. First, Student A creates a merge request. This merge request object is added to the outgoing merge requests list of Group1 and a notification about the merge request goes to Student 2, since she is the only student in Group1 that has not sent the merge request. The merge request is also added to the incoming merge requests list of Group 2 and a notification about the merge request goes to all members of Group2 (Student C and Student D). In this scenario, all members of both groups accept the merge request (except for Student A who is the one who sent the request and because of that it is automatically assumed that she has accepted the request.) Student B, Student C, and then Student D accepts the merge request in this order. After all of them votes, the voting is finished. Then, the merging operation is performed inside the Group2 object. After two groups are merged, Group1 is deleted. Finally, a notification about the result of the merge request is sent to all members of both groups.

**Created objects:** MergeRequest

**Destroyed objects:** ProjectGroup (Group1)

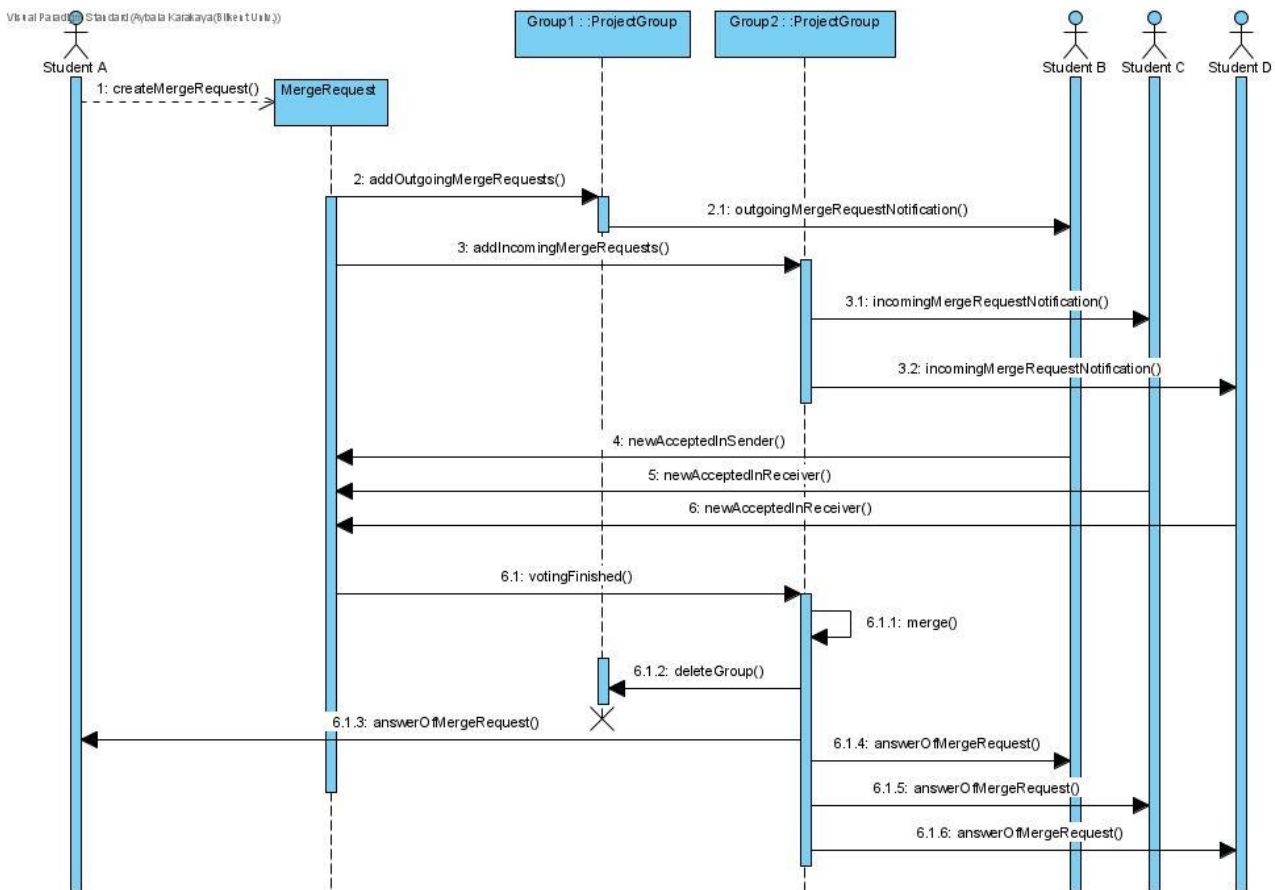


Figure 6: Sequence Diagram for Merging Groups Workflow

#### 6.2.1.3.4 Finalizing Group Workflow

**Scenario:** Student A wants her group, Group2 to finalize. Group2 has 4 members: Student A, Student B, Student C, and Student D. Student A requests a group finalization. Then, a notification about the group finalization request goes to the members of the group who have not requested the finalization. That is, the notification goes to Student B, Student C, and Student D. Two alternatives are shown below in this scenario: all students accept the finalization request or Student D rejects. In the first scenario, Student B, Student C, and Student D accepts finalization respectfully. Therefore, the ProjectGroup gets finalized and a notification about the finalization request result goes to all members of the group. In the second scenario, Student B accepts the finalization request but Student D rejects it. Thus, the finalization gets cancelled and Student C cannot vote anymore. A notification about the finalization request result goes to all members of the group. Student D decides to leave the group. Therefore, she leaves Group2 and her Student object is removed from Group2. Then, a group, Group3, with member Student D is created automatically.

**Created objects:** Group3 :: ProjectGroup (in the Student D rejects finalizing alternative)

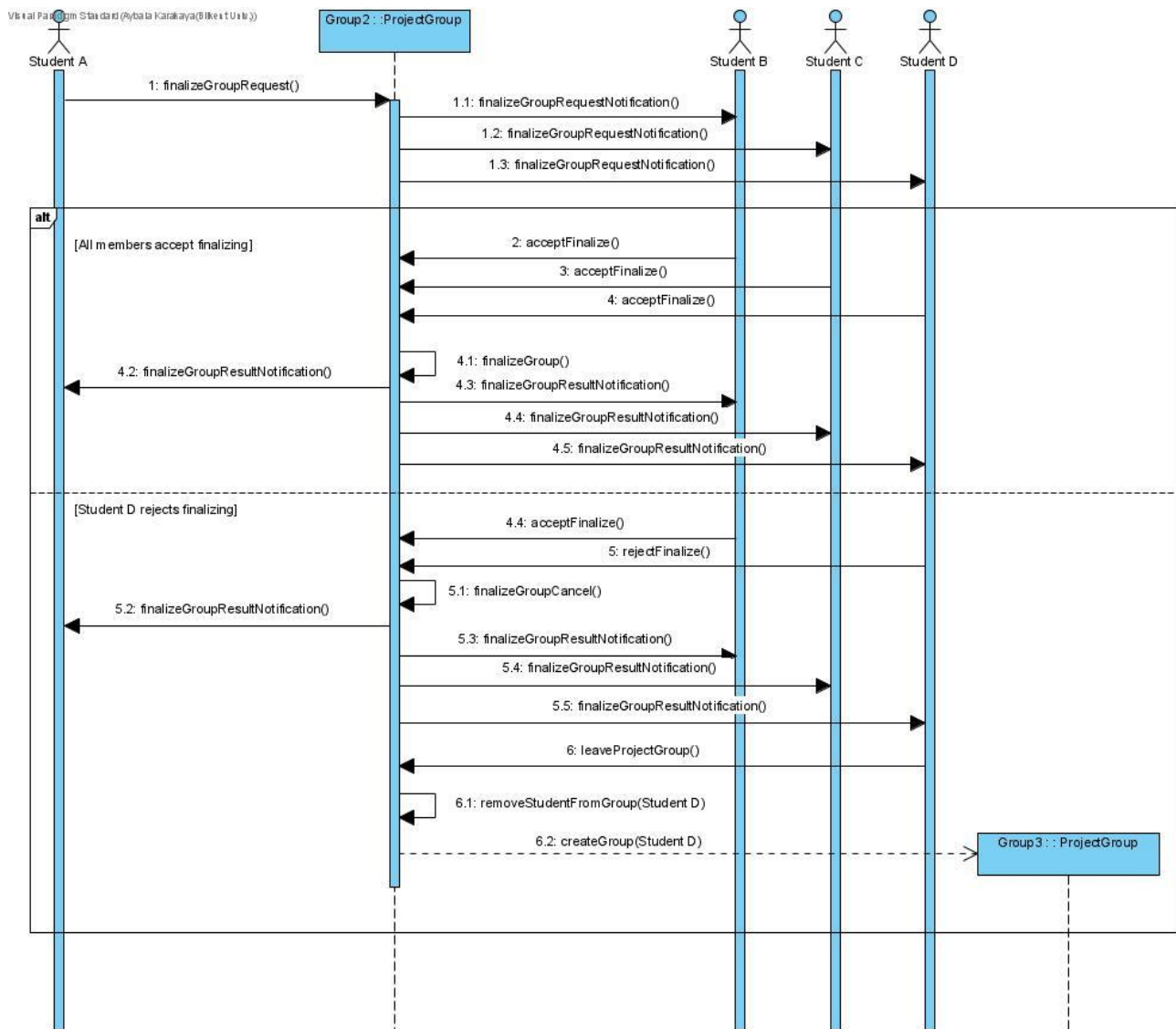


Figure 7: Sequence Diagram for Finalizing Group Workflow

## 6.2.1.4 Assignments

### 6.2.1.4.1 Giving Assignment Workflow

**Scenario:** Instructor wants to give an assignment. First, he creates an assignment. Then, he sets a description for the assignment, sets a due date, and specifies a maximum file size. Then, he adds the assignment to the course. Then, he wants to make the assignment gradable by other students, so he changes its grade state. After that, he adds statistics to the assignment. Then, he wants to change the visibility of the assignment and makes it visible by the student. Next, he publishes the assignment in the course page and a notification indicating there is a new assignment is sent to the students. However, he notices a typo in the assignment prompt. Thus, he edits the prompt by setting an assignment description.



### Created objects: Assignment

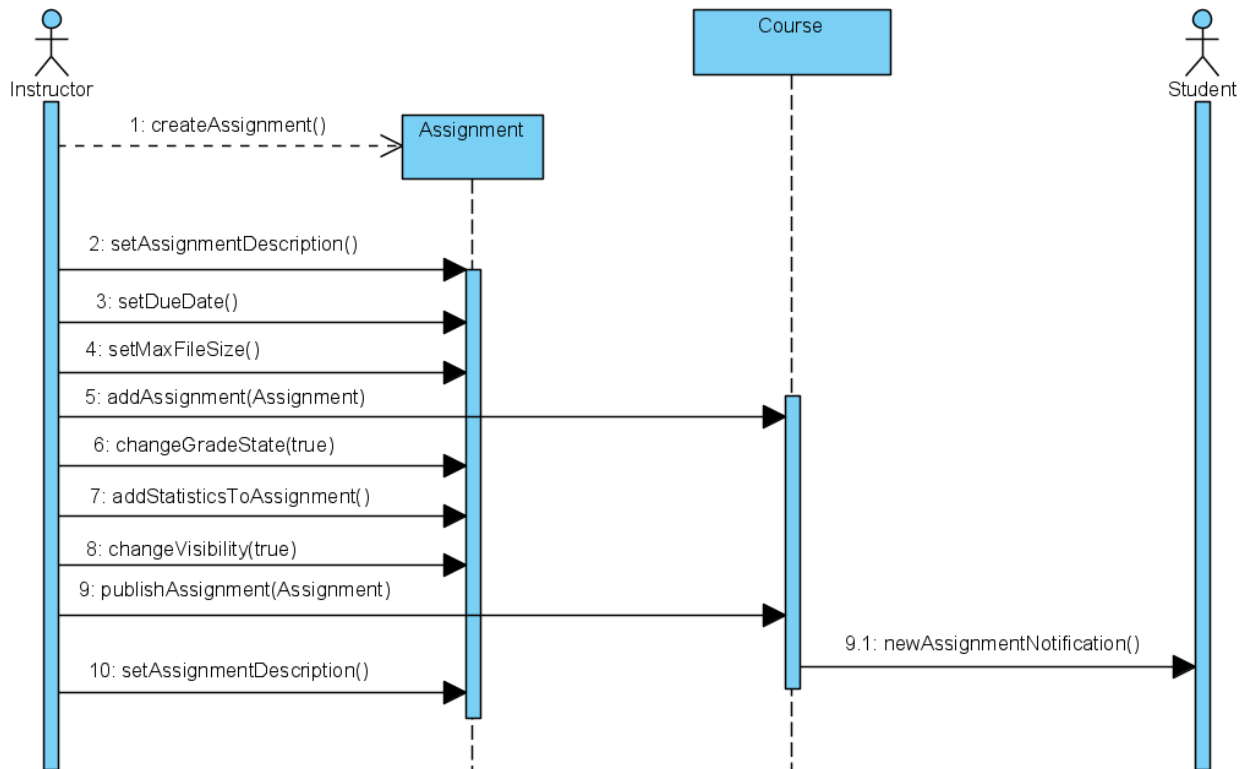


Figure 8: Sequence Diagram for Giving Assignment Workflow

#### 6.2.1.4.2 Submitting Assignment Workflow

**Scenario:** Student wants to submit an assignment. First, he creates a submission and uploads a file. Then, it is controlled if the file size is within the specified limits by first getting the maximum file size from the assignment of the submission and then comparing this maximum size with the file size. If it is not within the limits, the student is notified of the situation. In this scenario, a student uploads another file within the limits if this is the case. Then, the submission is added to the assignment and it is controlled if the last date of the assignment has passed or not. If the last date has not passed yet, the assignment gets submitted to the course and Student gets a notification. The submission is also added to the ProjectGroup of the Student. Else, the submission is removed from assignment, the student gets a notification about the last date having been passed, and then the submission is deleted.

**Created objects:** Submission

**Deleted objects:** Submission, if the last date of submission has passed.

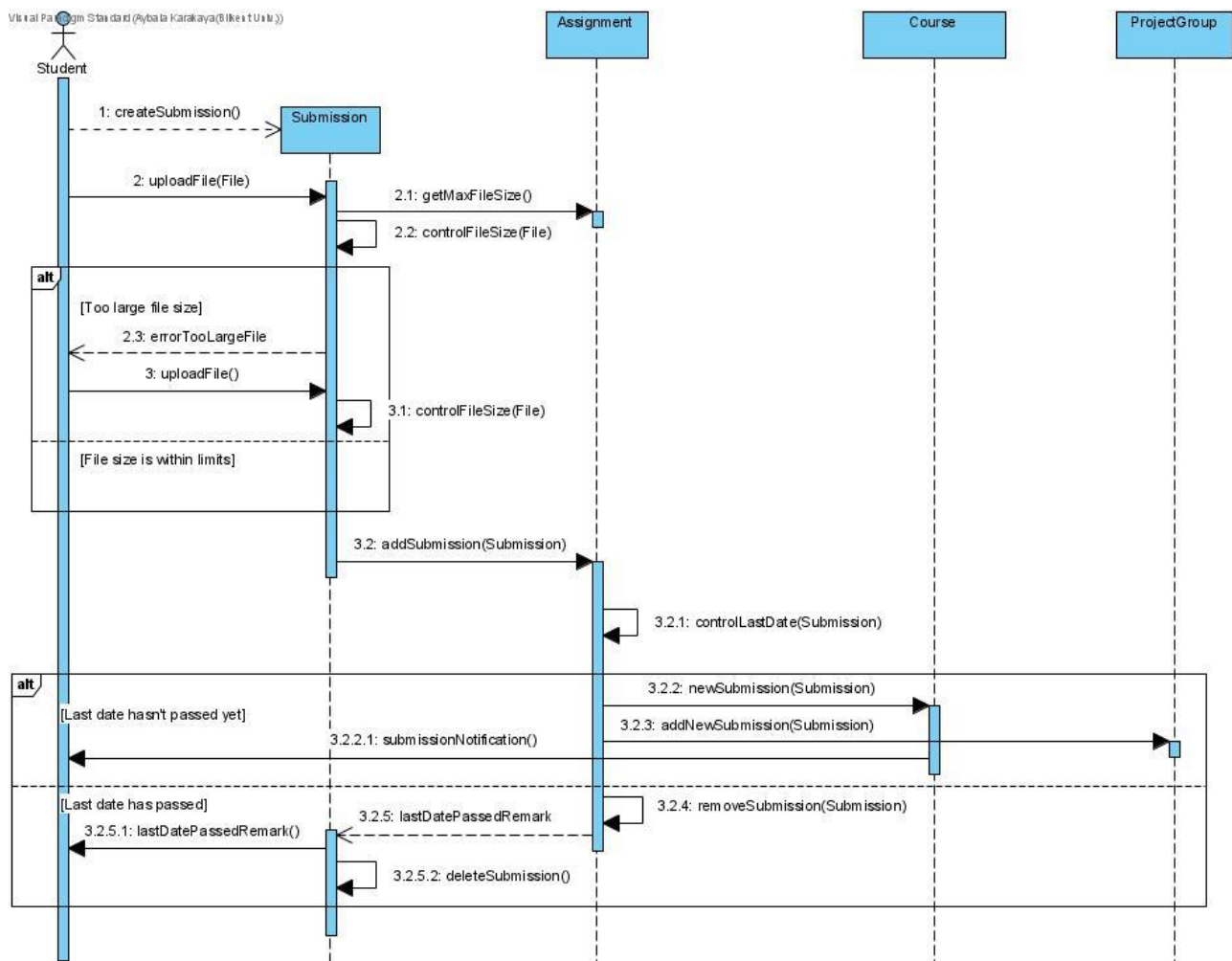


Figure 9: Sequence Diagram for Submitting Assignment Workflow

## 6.2.1.5 Feedback

### 6.2.1.5.1 Instructor Giving Feedback Workflow

**Scenario:** Instructor wants to give feedback to the student. He creates a comment. He sets comment text and comments about the submission, indicates the maximum possible grade for the assignment, and grades the submission. After he grades, the grade status of the comment becomes true, indicating that this comment includes a grade. Then, he adds this comment to the submission and the grade status of the submission changes and becomes “graded”. A notification stating that the submission has been graded is sent to the Student. After that, the instructor notices a mistake in his grading and changes the grade.

**Created objects:** Comment

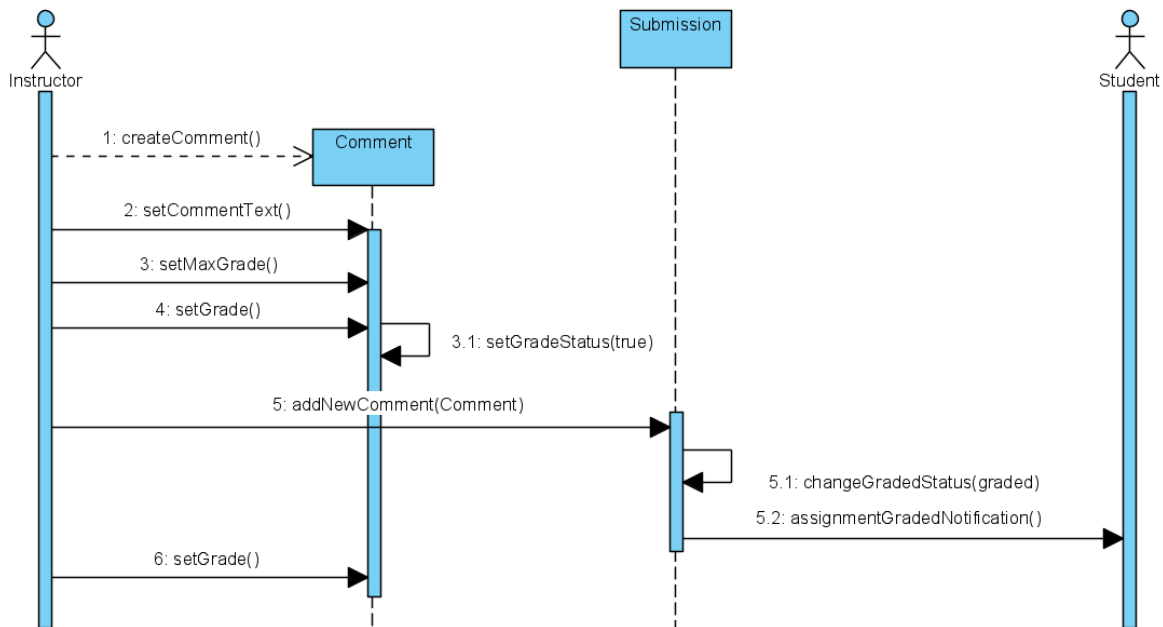


Figure 10: Sequence Diagram for Instructor Giving Feedback Workflow

#### 6.2.1.5.2 Student Giving Feedback Workflow

**Scenario:** Student wants to give feedback to another student. First, it is controlled if the assignment is reviewable by students. Reviewer Student wants this information from Submission object, which gets it from Assignment object. If the assignment is reviewable by the student, he creates a comment. He sets comment text and comments about the submission, and tries to grade the submission. After he grades, the grade status of the comment becomes true, indicating that this comment includes a grade. Then, he adds this comment to the submission. A notification stating that the submission got new feedback is sent to the student whose assignment has been reviewed. After that, the reviewer student edits his feedback but only the comment text after noticing another detail in the submission.

**Created objects:** Comment

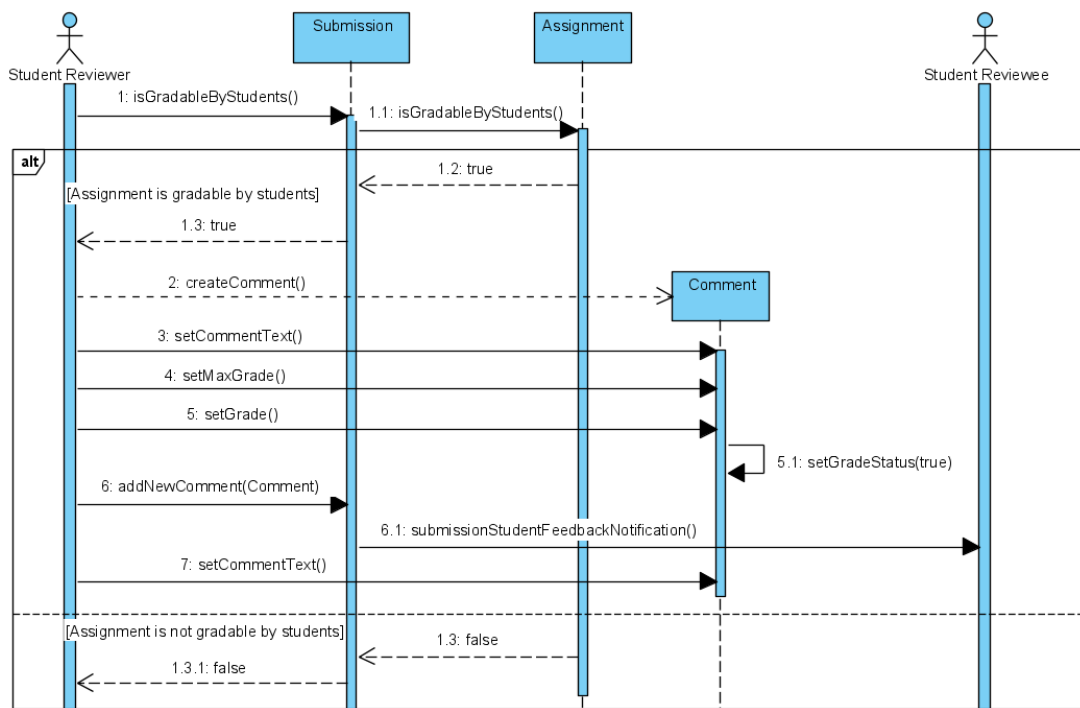


Figure 11: Sequence Diagram for Student Giving Feedback Workflow

### 6.2.1.5.3 Student Seeing Feedback Workflow

**Scenario:** Student wants to see the feedback given for a submission. First, it is controlled if the assignment feedback is visible to students. Student wants this information from Submission object, which gets it from Assignment object. If the feedback is visible, Student calls `displayFeedback()` on submission and then the comment is displayed. Else, student cannot see the feedback.

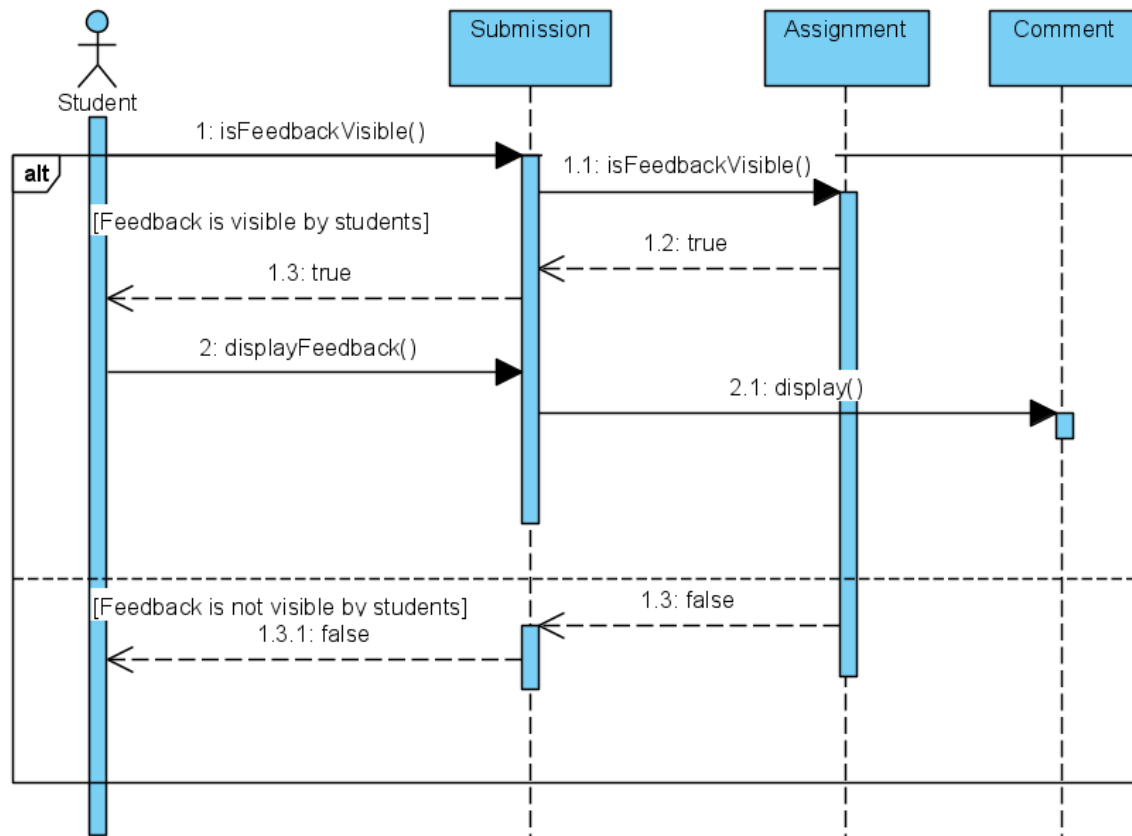


Figure 12: Sequence Diagram for Student Seeing Feedback Workflow

#### 6.2.1.5.4 Instructor Seeing Feedback Workflow

**Scenario:** Instructor wants to see the feedback given for a submission. He calls `displayFeedback()` on submission and then the comment is displayed.

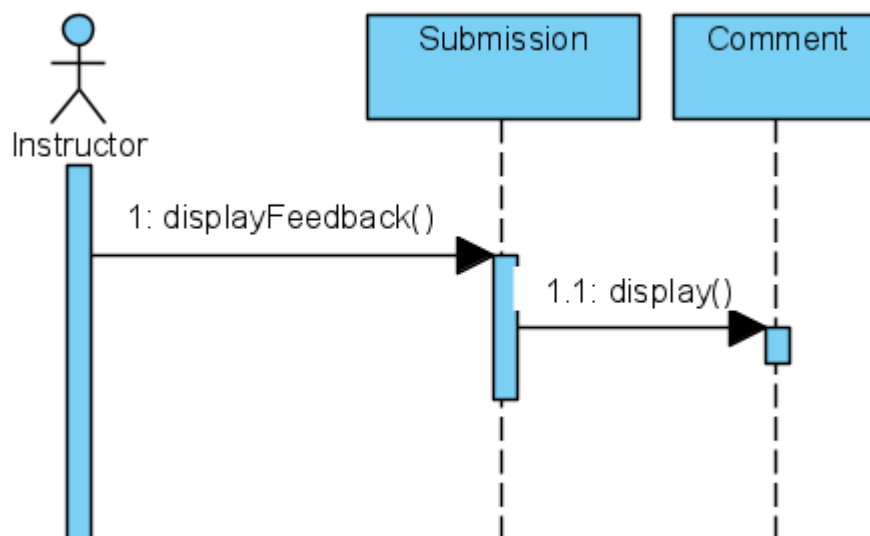


Figure 13: Sequence Diagram for Instructor Seeing Feedback Workflow

### 6.2.1.6 Peer Grading

#### 6.2.1.6.1 Giving Peer Grade Assignment Workflow

**Scenario:** Instructor wants to give a peer grading assignment. He creates a new peer grade assignment. Then, he sets an assignment description and a due date. Finally, he publishes the peer grade assignment for the Course and the peer grade assignment is added to the ProjectGroups' pages of the Course. In this scenario, there is only one ProjectGroup in the course.

**Created objects:** PeerGradeAssignment

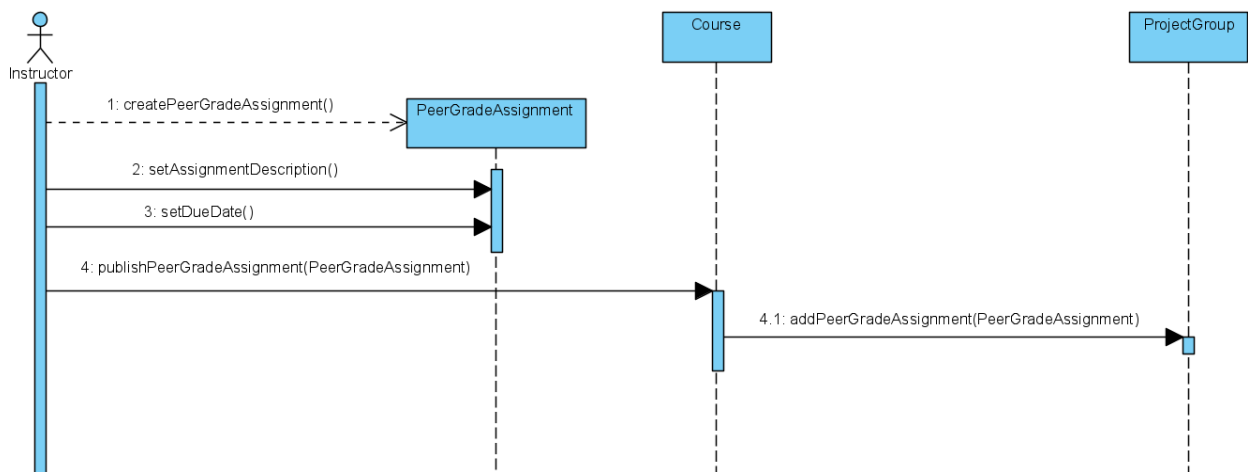


Figure 14: Sequence Diagram for Giving Peer Grade Assignment Workflow

#### 6.2.1.6.2 Performing Peer Grading Workflow

**Scenario:** Student wants to perform peer grading. She creates a peer grade and then adds the peer grade to the ProjectGroup.

**Created objects:** PeerGrade

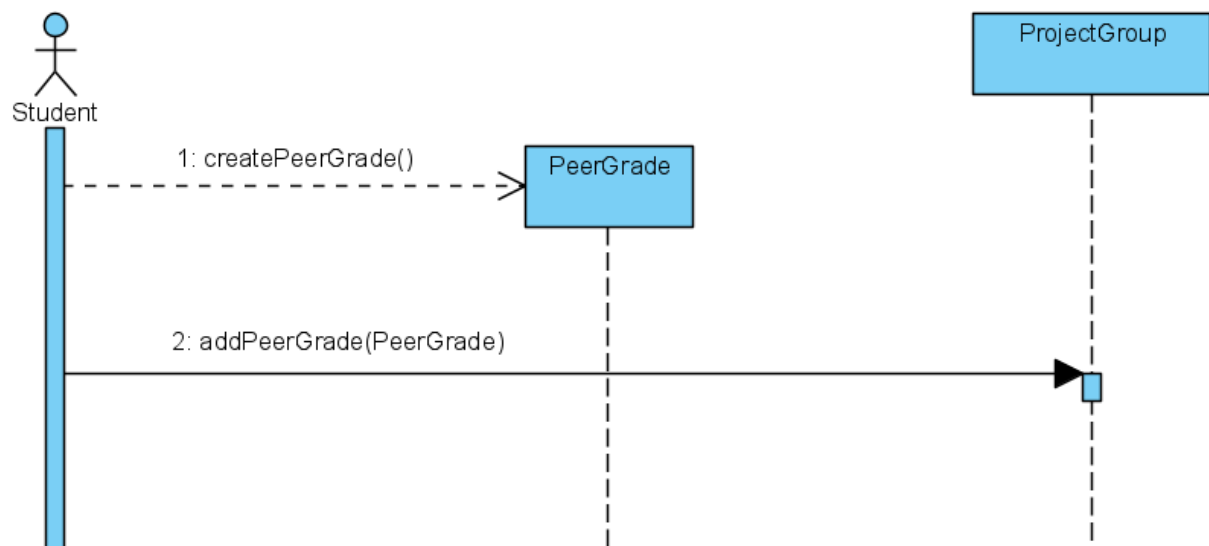


Figure 15: Sequence Diagram for Performing Peer Grading Workflow

## 6.2.2 Activity Diagrams

### 6.2.2.1 Creating Courses and Adding Participants

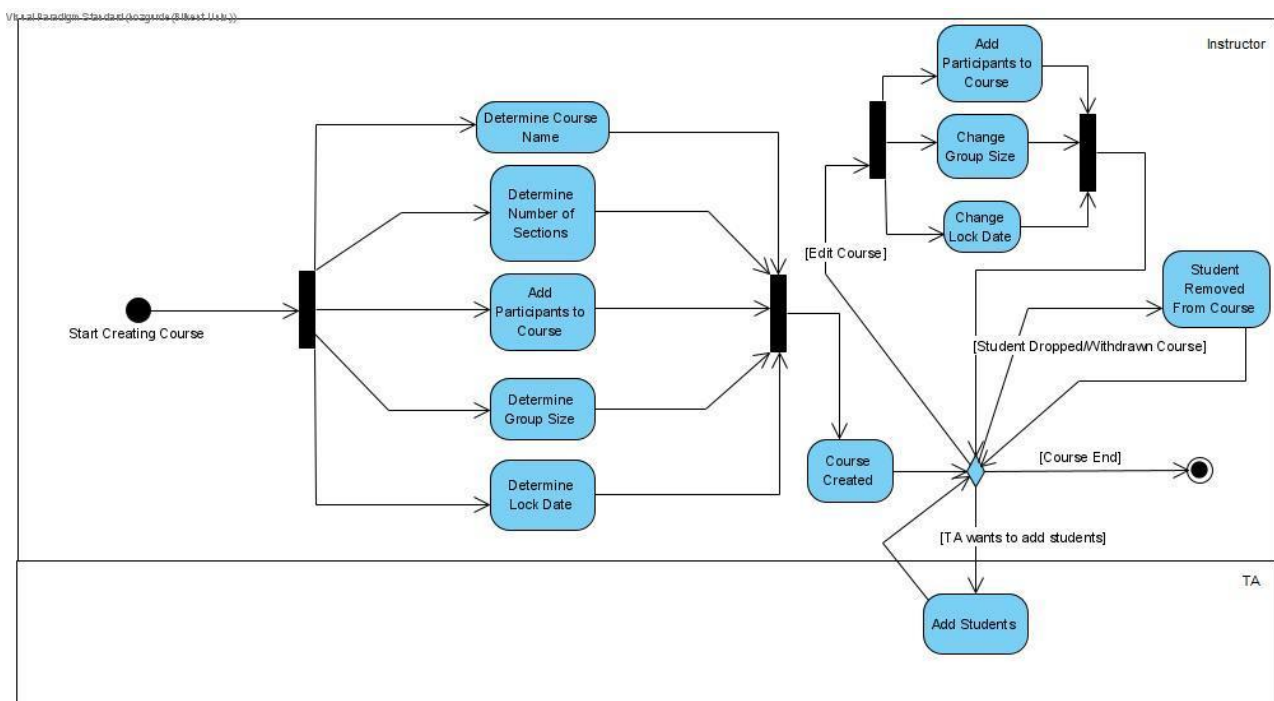


Figure 16: Activity Diagram for Creating Courses and Adding Participants

When instructor initialized course creation, s/he first need to determine some of the properties of the course. As each course must have name, number of sections, group size and lock dates, instructor must decide these while creating the course. Adding participants to course is optional initially. When course is created there may be different kind of decisions that can be made. TA can only add new students to group in this task. Instructor can change some information of the course if s/he wants to. However not all the information of the course

is editable. While group size and lock date can change, course name and section number cannot change. Also instructors can add new participants while s/he is editing the course. Instructors can also decide to remove students or TAs from the course if it is needed ( For example, one student dropped the course or withdrawn from the course.). When the course ends, these activities no longer exist for that course.

### 6.2.2.2 Forming Groups

Visual Paradigm Standard (602996) (Bilal et al.)

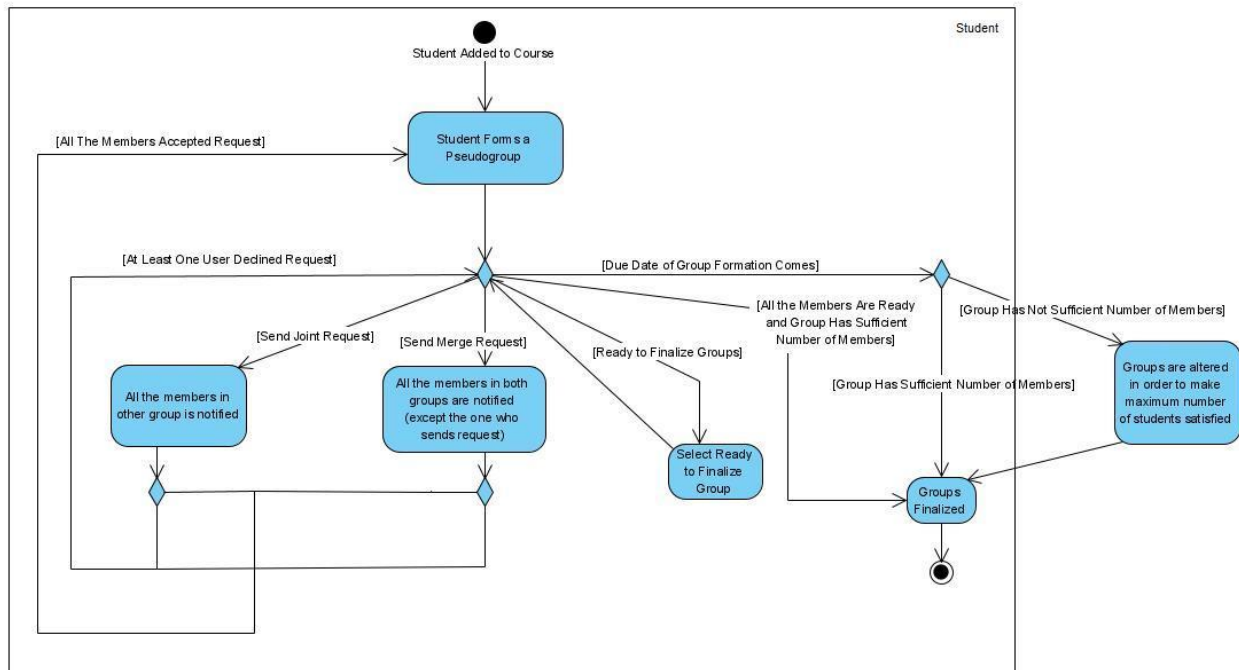


Figure 17: Activity Diagram for Forming Groups

Students need to be added to course to form groups. Firstly all of the students are counted as a one member pseudo groups. There are two different activities to change pseudo groups which are joining to a group and merge two different groups. These activities occur upon request. If student who wants to send a request is in a pseudogroup and the group which s/he wants to send a request to is pseudogroup, merge or join request can be sent.

When one student sends a join request to another group, there can be two different outcomes. Whether all of the members of that group accepts a join request, or at least one of the members of that group refuses the join request. When the request is accepted, a new pseudo group is formed by the joint of new student. If the request is declined, the student who sent the join request returns to the decision state.

When one student sends a merge request to another group, again there may be two different outcomes. Whether all of the members in each group except the one who sent the request, accepts the request, or at least one of the members in each group refuses to join



the request. When request accepted, a new group is formed by merging two groups. If the request is declined, the student who sent the join request returns to the decision state.

When one of the students wants to finalize the group, s/he becomes ready to finalize the group. When everyone in that group is ready and their group size is sufficient, group is finalized and cannot be changed by students.

Groups may not finalize before the deadline. In this case, there can be two outcomes. If group size is allowed, system will automatically finalize that group, else system distributes unformed students in order to satisfy maximum number of students and finalizes these unformed groups automatically.

### 6.2.2.3 Giving and Turning in Assignments

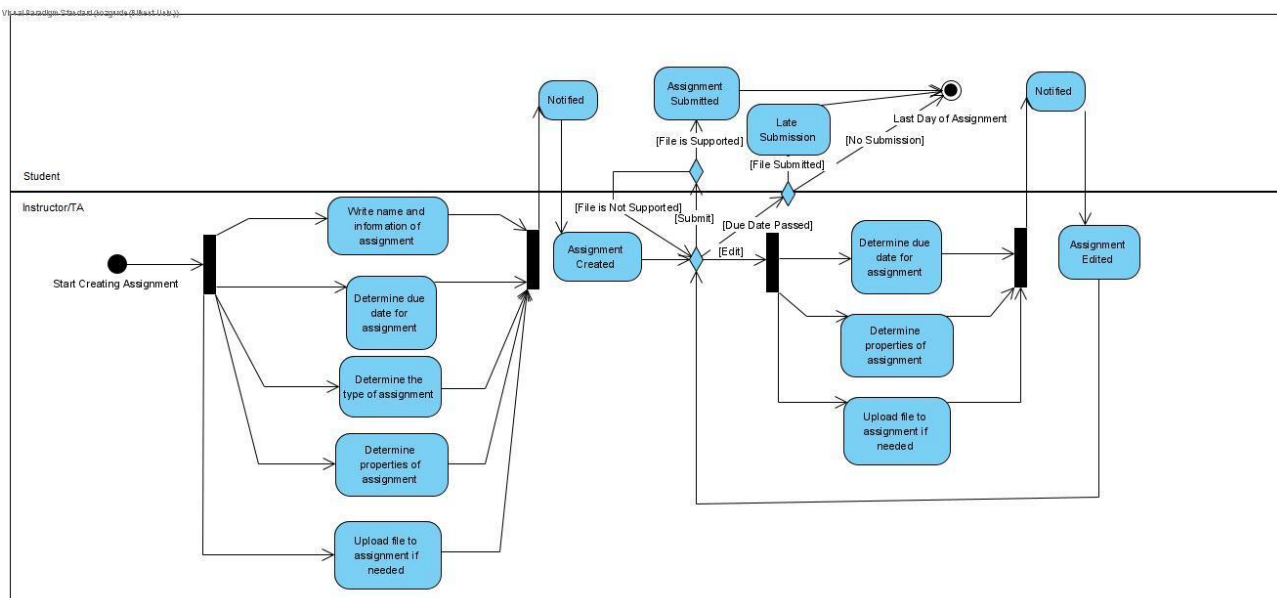


Figure 18: Activity Diagram for Giving and Turning in Assignments

In order to create an assignment, firstly TA or instructor should initialize this procedure. While creating the course, they have to write name and information of assignment, add due date to assignment, determine the type of assignment, determine the properties of the assignment. Also they can add files to assignment if they want to. When a new assignment is posted by a TA or instructor, students in that course are notified about it.

Students can upload files into assignments. Systems check for suitability of this file. If a file is not supported for that assignment, student cannot complete submission and returns to the decision stage. If the file is supported, submission is completed. If due date of assignment comes, the system starts to accept late submission for this assignment.

TAs or instructors can also edit submission. They can change due date, change properties of assignment or they can add new files or edit existing files. When they do this, all the students are notified about change.

### 6.2.2.4 Giving Feedback

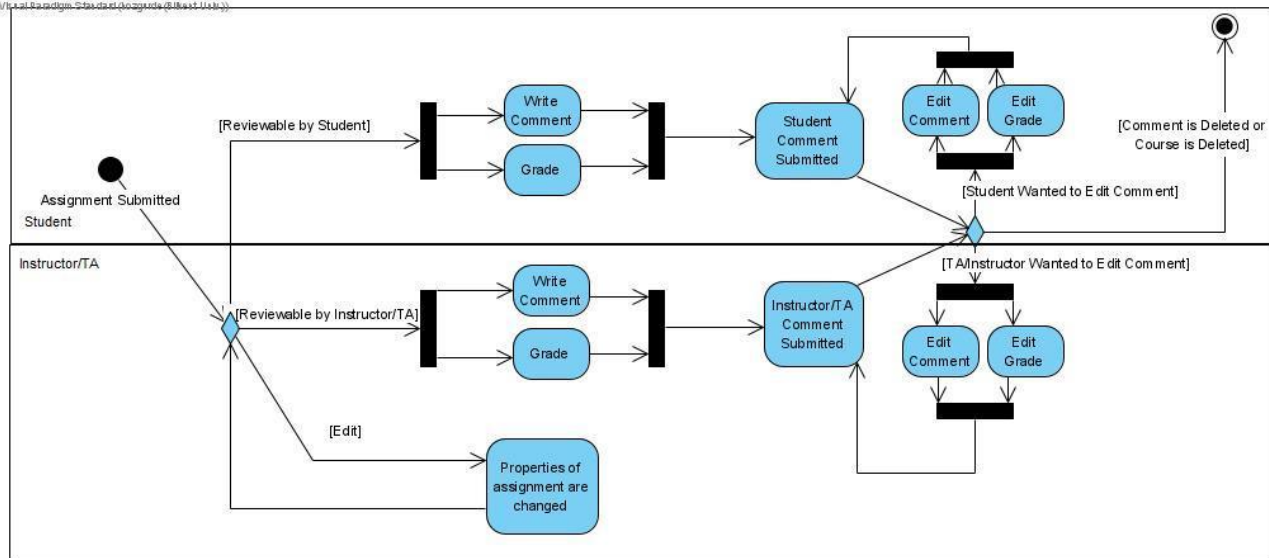


Figure 19: Activity Diagram for Giving Feedback

After the submission of any assignment, all the users in the course may start giving feedback to assignments. Instructor or TA may edit assignments and make it reviewable by students or TA/Instructor. If assignment is reviewable by instructor or TA, they can comment on this assignment or if assignment is reviewable by student, all the students in that course can comment on that assignment. Users give comments according to predefined properties of that assignment. For example they can only comment, only give grades or do both of them.

After any user comments on assignment, they can edit their comment before the end of course or deletion of comments.

### 6.2.2.5 Viewing Feedback

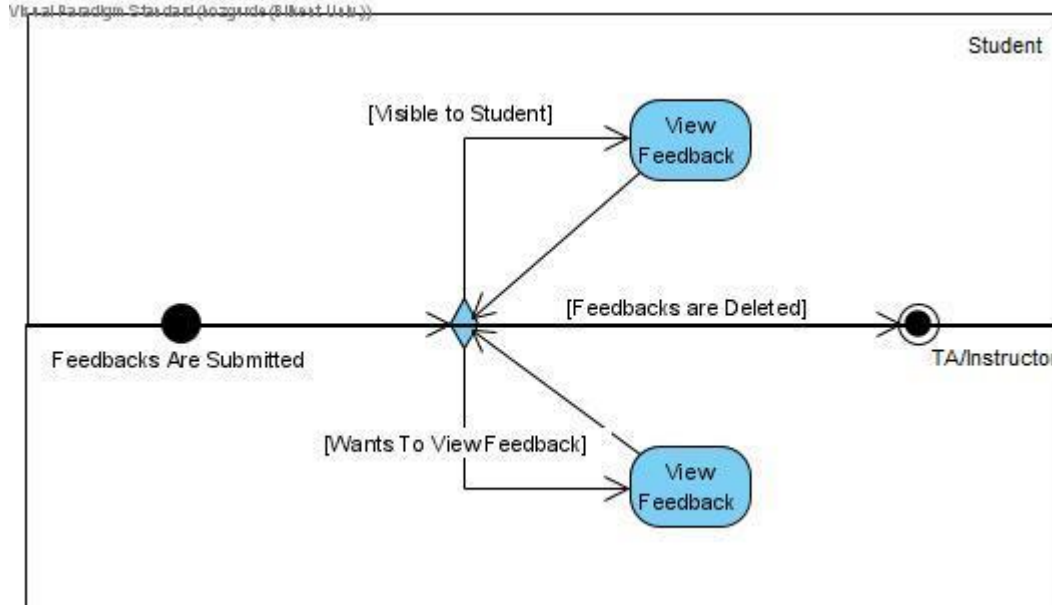


Figure 20: Activity Diagram for Viewing Feedback

After feedback is submitted, they can be viewed according to properties of assignment on which comment. While TAs and instructors can always view the comments of all users, students can only see if it is allowed to them. After feedback is deleted (either by deletion of course or deletion of assignment) it is no longer viewable.

### 6.2.2.6 Creating Statistics

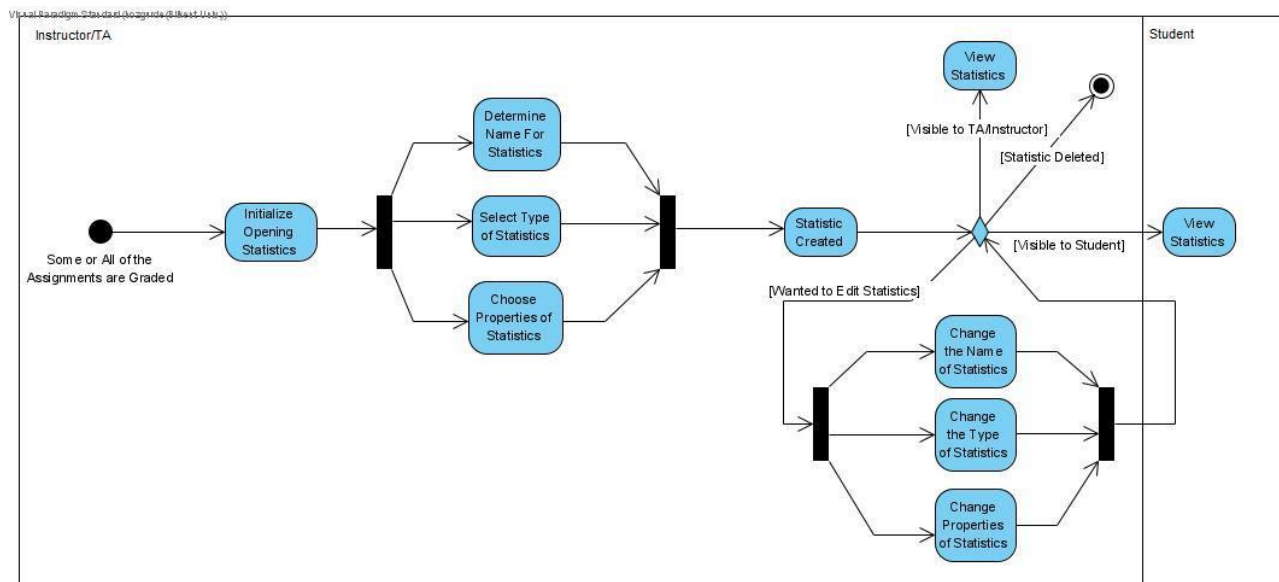


Figure 21: Activity Diagram for Creating Statistics

After some or all of the assignments are graded, instructors can create statistics in order to visualize all the grades given to the assignments. In order to open statistics, first the procedure must be initialized. While opening, instructor or TA must determine a name for the statistic, select the type of statistics and choose properties of statistics. When statistics

is created it can be edited by Instructor or TA. They can change the name of the statistic, type of the statistic and the properties of the statistic.

Visibility is one of the properties of statistics. It is always visible to TAs and instructors, however its visibility to students may be restricted by TAs or instructors. If it is visible to students, students can also view the statistics of assignments. When statistics is deleted it is no longer visible or editable.

### 6.2.2.7 Peer Grading

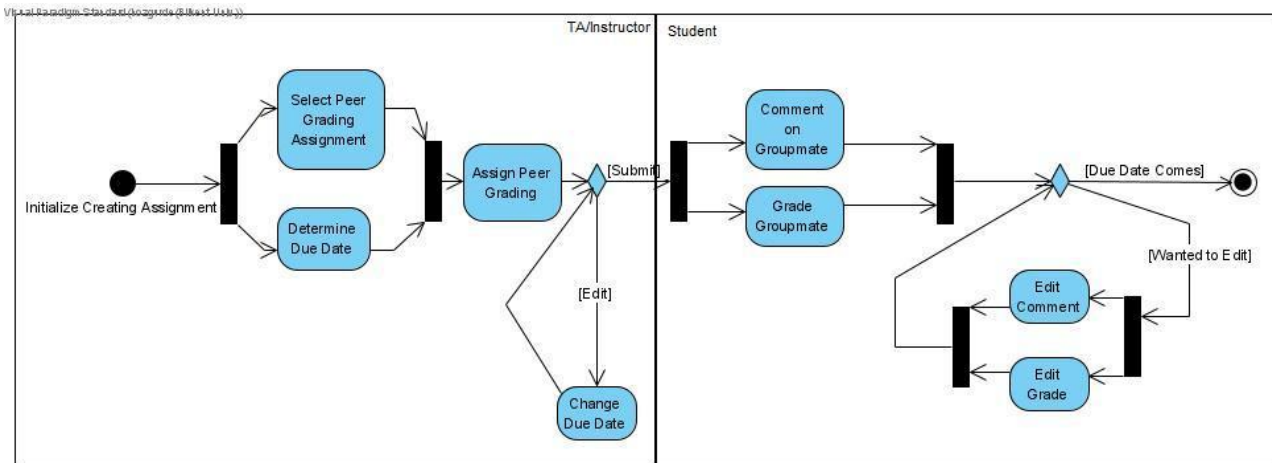


Figure 22: Activity Diagram for Peer Grading

While instructor or TA starts a new assignment they can choose to assign peer grading. When they did that, they only needed to determine the due date for these assignments. Ta or instructor can change the due date after peer grading assigned.

While submitting, students must comment on his/her group mates and give grades to them. After they make peer grading, they can edit comments they made on their group mates. When the due date has come, students no longer peer grade their group mates.

### 6.2.3 State Machine Diagrams

Find below the state machine diagrams for entities.

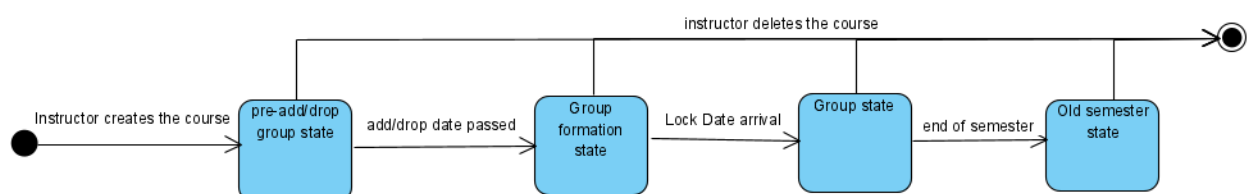


Figure 23: State Machine Diagram for a Course object

The state diagram above represents the entire potential life cycle of a *Course* object. It needs to be initiated by an instructor and can only -if wanted- be deleted by that instructor. Initially a course object starts in pre-add/drop state to let students know each other and look for potential teammates. After the add/drop date is passed students can form pseudogroups and lock their groups if eligible. After the pseudo groups converted to real groups either by student or the algorithm, a course object is now able to assignments, grading and commenting. After the end of semester, the course object goes into old semester state to allow students from the future to see previous work of others. This feature is up to the instructor.

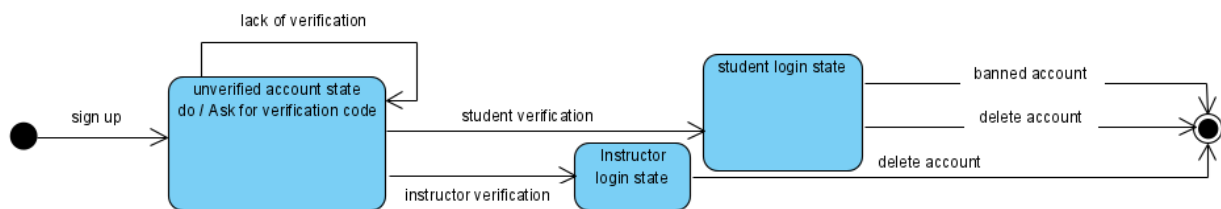


Figure 24: State Machine Diagram for a User object

The diagram above represents the potential life cycle of a user object. Once a user signs up with an edu extended mail, a unique verification code is sent to the entered mail address, and the account goes into unverified state. If the user verifies his/her account by entering the code, the system classifies the user as a student or an instructor depending on the mail entered. After a user is classified as an instructor or a student s/he can use all the features of that Course until the account is deleted.

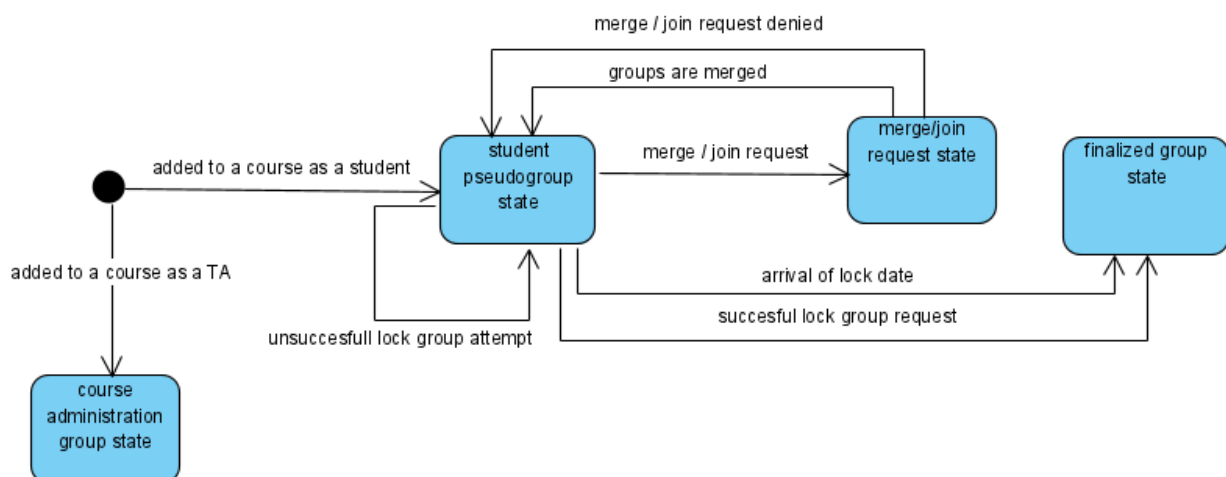


Figure 25: State Machine diagram for Student/TA objects

The state machine diagram above represents the life cycle of a user inside a course. Once a user is added to a course as a student it goes into pseudo group state with a pseudo group sized 1. After that, users can go into a merge/join request state either by sending or receiving a request. Note that a user or a pseudo group can be in a merge/join request state

with several groups. If groups decide to merge their pseudo groups, a new pseudo group with the members of both pseudo groups is created. Thereafter, if eligible pseudo groups may want to lock their group and go into finalized group state. Or with the arrival of lock date the system arranges the final group. In finalized group state users can see the assigned works, comments and grades, i.e. they can use the application fully pledged.

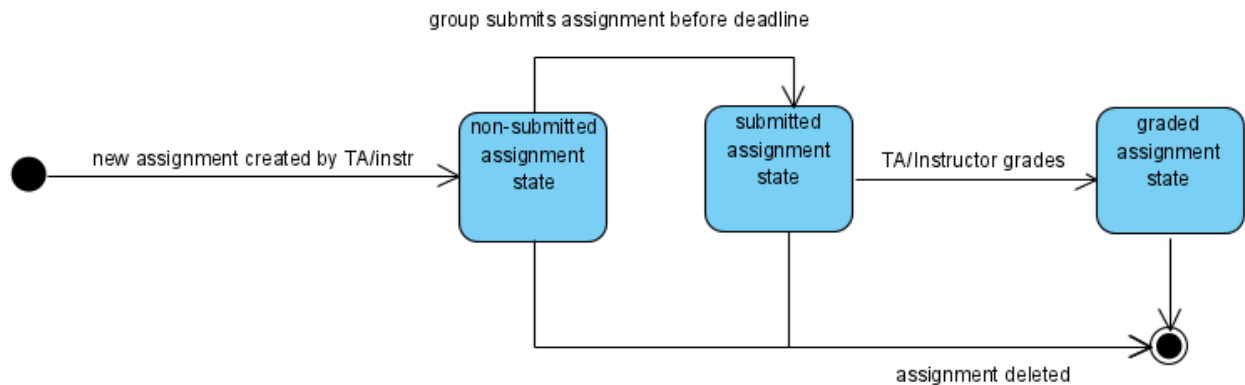


Figure 26: State Machine diagrams for assignment object

This state machine diagram above illustrates the life cycle of an assignment object. Before the submission or deadline of the assignment, the assignment stays in the non-submitted assignment state and can be changed by students. After submission of assignment or the deadline, students can not change the submission. And assignment becomes open to grading, after the grading assignment is done. An instructor may choose to delete it or leave it for observation of future semesters.

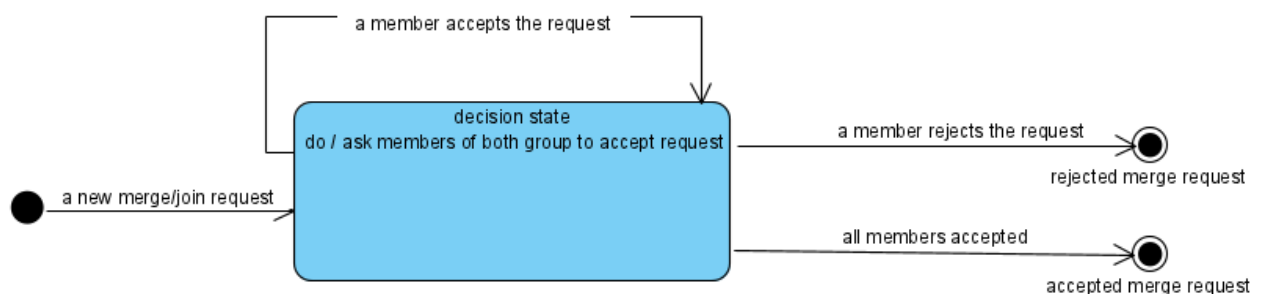


Figure 27: State Machine diagram for merge/join request object

The diagram above demonstrates the life cycle of a merge/join request. Request objects will be created whenever a request has been made to a pseudo group. If all of the members of pseudo groups accept the merge request a new pseudo group with members of old pseudo groups will be created. Thereafter the request object will be terminated.



## 6.3 Object and Class Models

Visual Paradigm Standard (Copyright © Springer Nature 2014)

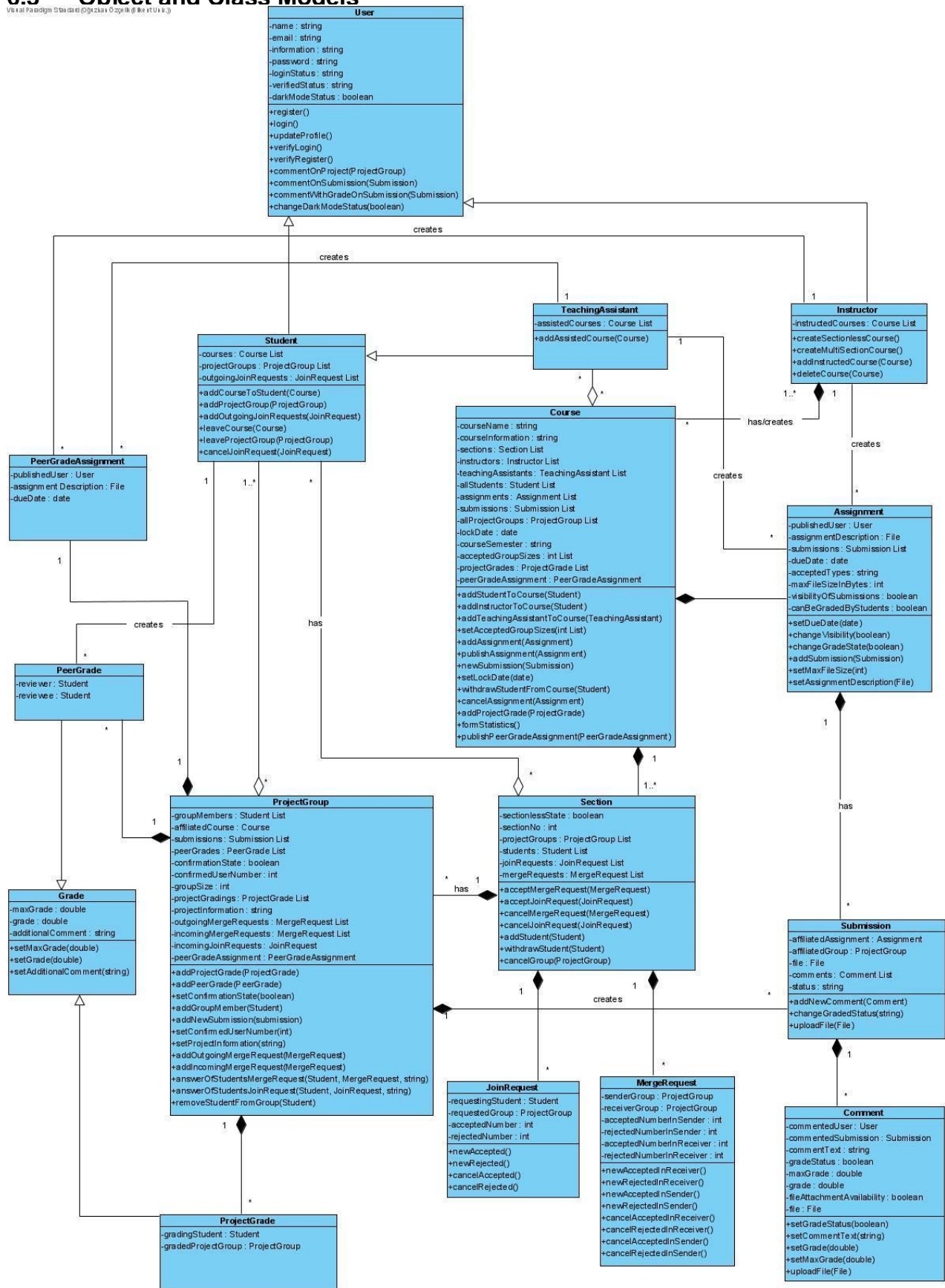


Figure 28: Object and Class Diagram for BilHub

## 6.4 User Interface

### 6.4.1 Navigation Bar

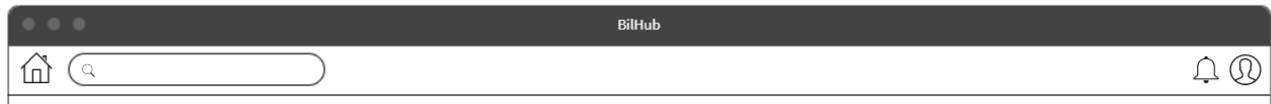




Figure 29 : Navigation Bar

 => Will send users to the main page.

 => Will send users to the notification page.

### Search Bar

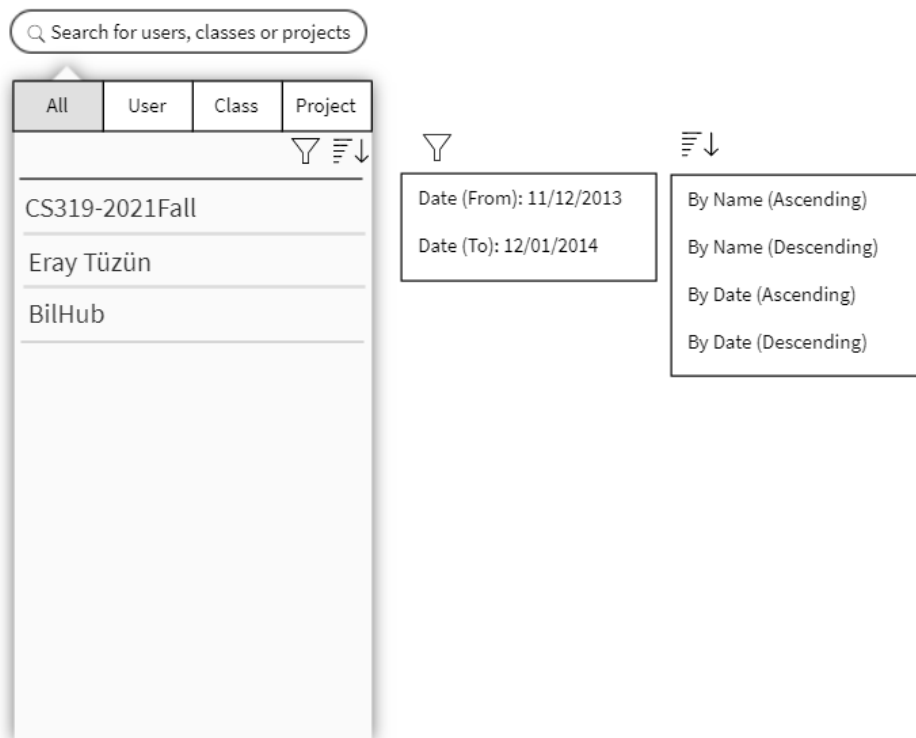



Figure 30: Search Bar

 Search for users, classes or projects => From the search bar, the users will be able to search for users, courses and projects. Additionally, they will be able to filter and sort their search results by date and name.



## Profile Options

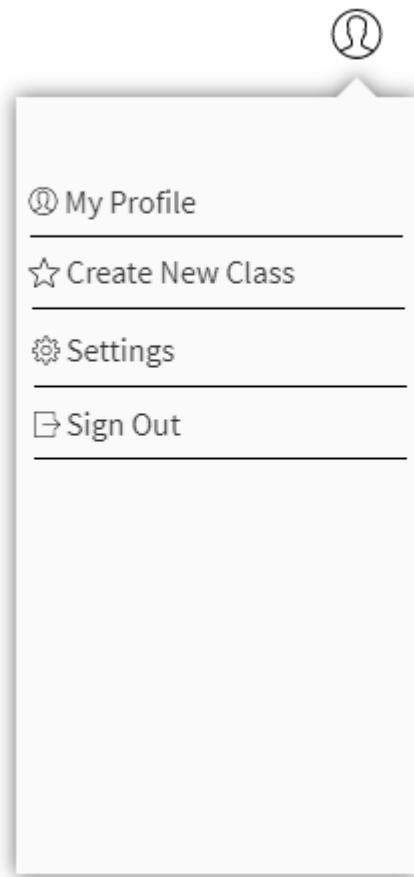


Figure 31: Profile Options

⇒ From the profile options, users will be able to access three different places. My profile will send them to their profile page. Sign out will sign them out from the website. Settings will send them to the settings page.

If the user is an instructor, then by clicking the create new course button, he/she will be able to create a new course.

## 6.4.2 Main Page

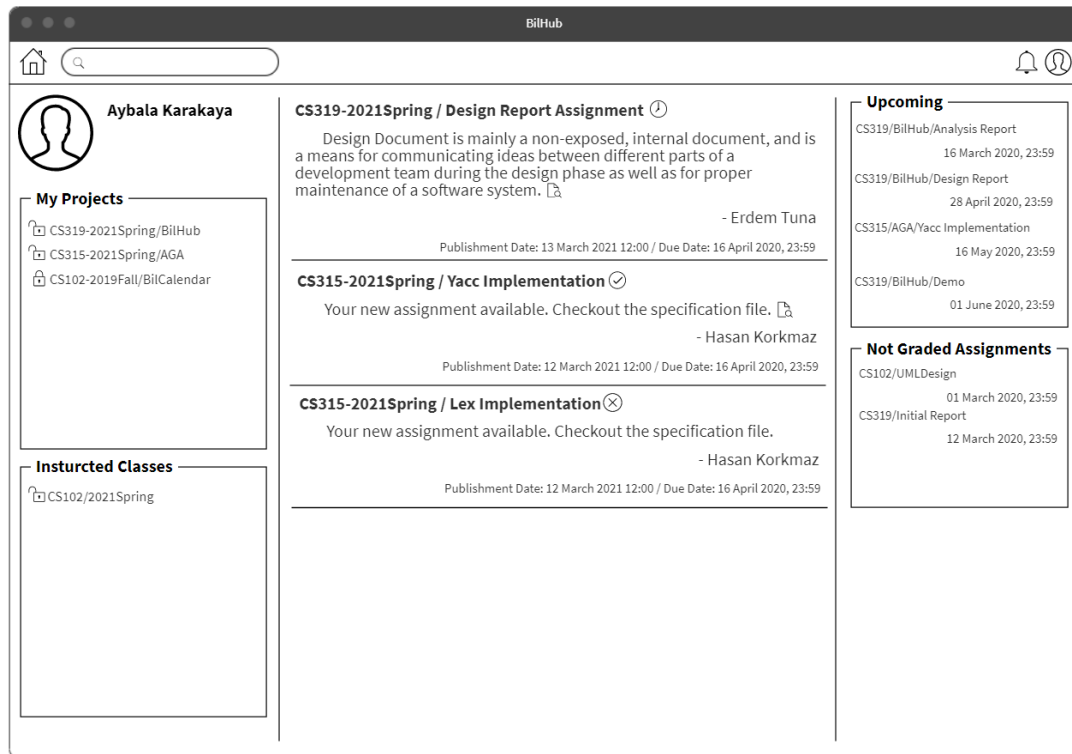


Figure 32: Main Page

### Left-part of the page





The name and the profile photo of the current user will be visible in the upper part. By clicking their name or photo, they will have access to their profile pages.

**Students or Teaching Assistants:** Users will see the names and the status of all their projects sorted from newer to older. By clicking the specific project's name, the user will be sent to that specific project's page. The icon of open lock means the project is active (current semesters projects), the icon of closed lock means the project is inactive (old semesters projects).

**Teaching Assistants or Instructors:** Users will see the names and the status of all their courses sorted from newer to older. By clicking a specific course name, the user will be sent to that specific course page. The icon of open lock means the course is active (current semesters courses), the icon of closed lock means the course is inactive (old semesters courses).

### Middle-part of the page:

All the assignments that are created will be visible with their name, description, file, the creator's name, and publication & due dates. They will be sorted by their publication dates from newer to older. By clicking a specific assignment, the users will be sent to that specific assignment's page.

-  => Means the assignment is submitted for the students and the assignment graded for teaching assistants and instructors.
-  => Means the assignment is not submitted for the students.
-  => Means the assignment waiting to be submitted from students for teaching assistants and instructors
-  => To download the assignment file.

### Right-part of the page:

**Students and Teaching Assistants:** All the assignments whose due dates closing will be visible in this part with their name and due dates. Due dates will be sorted by their due dates from closer to further. By clicking a specific assignment, the users will be sent to that specific assignment's page.

**Teaching Assistants or Instructors:** All the assignments that are not completely graded and whose due date is over will be visible by the user. Due dates will be sorted by their due dates from older to newer. By clicking a specific assignment, the users will be sent to that specific assignment's grading page.

### 6.4.3 Projects Page

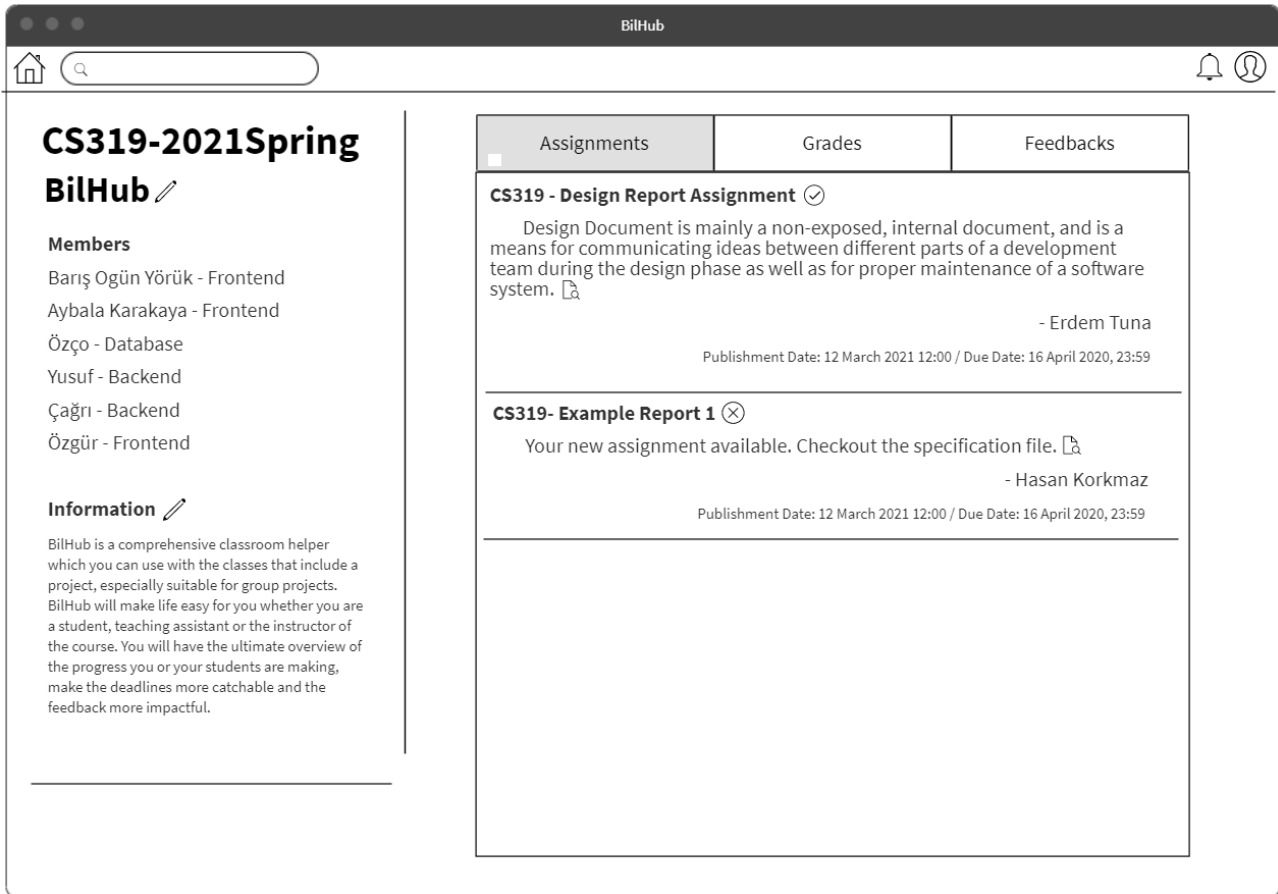




Figure 33: Project Page

#### Left-part of the page:

In the upper-part, course names will be visible. By clicking the course name the user will be sent to that course. Also the project name will be visible under the course name. By clicking , group members will have access to change the group name (Until determined time, after that time, this button will be invisible).

In the middle-part, the group members' names will be visible. By clicking their name, the user will be sent to that specific group member's profile page.

In the lower-part, there will be an information text about the group. The information text will be changeable by the group members via clicking  button.

#### Middle-part of the page:

There will be a tab bar that contains three choices to show information about the group.

**Assignment Section:**

Assignments	Grades	Comments
<b>CS319 - Design Report Assignment</b> ✓ Design Document is mainly a non-exposed, internal document, and is a means for communicating ideas between different parts of a development team during the design phase as well as for proper maintenance of a software system. 📎 <div>- Erdem Tuna</div> <div>Publication Date: 12 March 2021 12:00 / Due Date: 16 April 2020, 23:59</div>		
<b>CS319- Example Report 1</b> ✗ Your new assignment available. Checkout the specification file. 📎 <div>- Hasan Korkmaz</div> <div>Publication Date: 12 March 2021 12:00 / Due Date: 16 April 2020, 23:59</div>		
Empty row for visual consistency		

Figure 34: Project Assignment Section

Assignments section will show the assignments that belong to the project and will be visible with their name, description, file, the creator's name, and publication & due dates. They will be sorted by their publication dates from newer to older. By clicking a specific assignment, the users will be sent to the group assignment's page to see submission.

✓ => Indicates that the assignment was submitted by the group.

✗ => Indicates that the assignment was not submitted by the group.

**Grades Section:**

Assignments	Grades	Feedbacks
	Person	Grade
Project Instructor	Eray Tüzün	9.5/10
Instructors Average		9/10
TA	Erdem Tuna	9/10
TA	Elgun Jabrayilzade	7/10
Students Avarage		6.2/10
Total Avarage		7.1/10
BilHub Avarage		Class Avarage
7.1/10		6.5/10
<b>Final Grade: 38/40</b>		

Figure 35: Project Grades Section

The Grades section will show the final grade of the project. It will only be visible after the finalized project is graded. Project instructor grade and teaching assistants grades will be visible one by one. Other instructors' grades and students' grades will be displayed as an average. Final grade at the end of the section shows how much grade that project worthed.

**Comments Section:**







Assignments	Grades	Feedbacks
<b>SRS Result</b>   <p>Please download the complete feedback file. </p> <p>Grade: 9.5/10 -Elgun Jabrayilzade, 11 March 2021</p>		
<b>Instructor Comments</b> <p>They work hard and it was a great project.</p> <p>Grade: 9.5/10 -Eray Tüzün, 11 March 2021</p> <p>Wonderful job.</p> <p>Grade: 9/10 -Melih Demir, 06 Februray 2021</p>		
<b>TA Comments</b> <p>In the end, it was a good project. Congratulations</p> <p>Grade: 9/10 -Erdem Tuna, 12 March 2021</p> <p>Well job.</p> <p>Grade: 9/10 -Elgun Jabrayilzade, 27 January 2022</p>		
<b>Student Comments</b>   <p>This project is amazing.</p> <p>Grade: 9/10 -Mustafa Doğan, 17 April 2021</p>		
		

Figure 36: Project Feedbacks Sections

Comments section will show the comments that are sent by the users. SRS Result, Instructor's grades, Teaching assistant's grades and student's grades will be separately shown. The comments will be sorted by their dates from newest to oldest.


If the comment made by the current user he will be able to edit and delete his comments via  .

The SRS Result will be editable via   buttons by the teaching assistants or the instructors.


If there is no SRS Result yet the following button will be visible only to instructors and the TA's:


Submission	Grades	Feedbacks
<div>Add SRS Result</div>		

Figure 37: Project Add SRS Result Button

If the user have not commented yet, he/she will be able to comment via . Clicking this button will result in following pop-up:

**Comment to BilHub Project**
×

 Please write your comment below

The most important aspect of this project is simplicity. I like simple projects. This is one of those.
 

Add File

**Grade:**

Comment

Figure 38: Project Commenting Pop-up

Here the user could write his comment and give a grade to the project.

If the user is a teaching assistant or instructor, he/she will be able to add a file as a feedback.



## 6.4.4 Project's Assignment Page

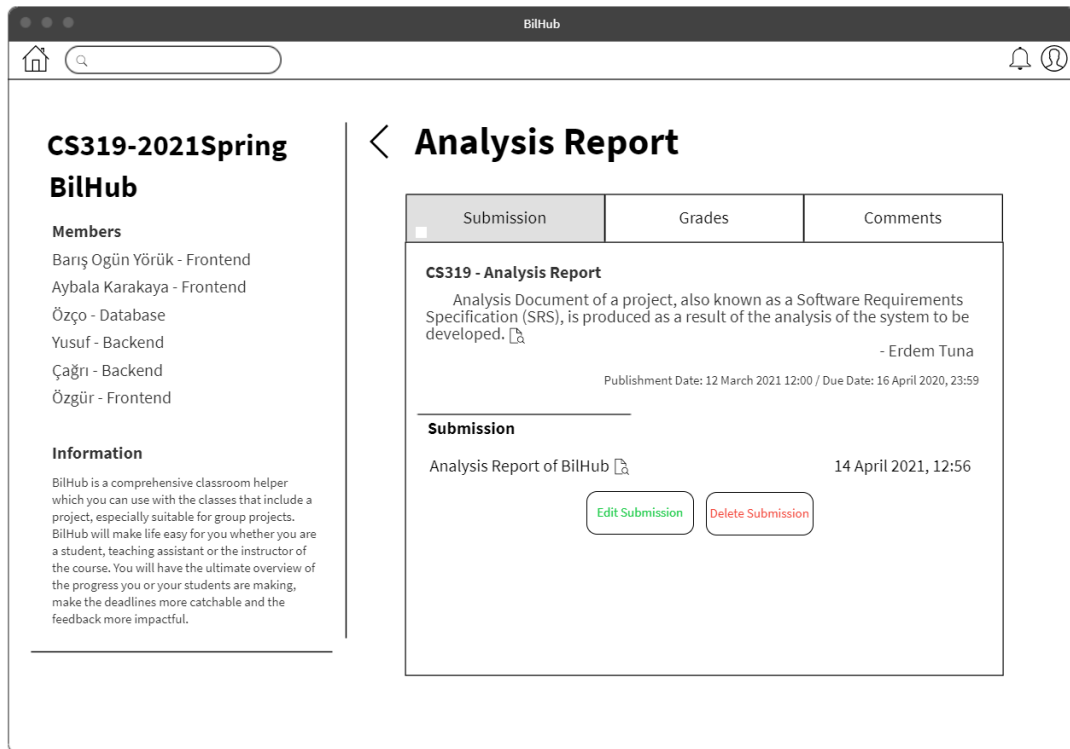


Figure 39: Project Submission Page

In the assignment page, the left-side of the website will be the same with the project page for giving information about which group is responsible for that specific submission.

In the rest of the page, there will be a top bar with three sections. Above the topbar, the assignment name will be displayed and in case of clicked to assignment name, the user will be directed to that assignment publication in the course page.

## Submission Section:

Submission	Grades	Feedbacks
<p><b>CS319 - Analysis Report</b></p> <p>Analysis Document of a project, also known as a Software Requirements Specification (SRS), is produced as a result of the analysis of the system to be developed. 📎</p> <p>- Erdem Tuna</p> <p>Publishment Date: 12 March 2021 12:00 / Due Date: 16 April 2020, 23:59</p> <hr/> <p><b>Submission</b></p> <p>Analysis Report of BilHub 📎 <span style="float: right;">14 April 2021, 12:56</span></p> <div style="display: flex; justify-content: center; gap: 20px;"> <span>Edit Submission</span> <span>Delete Submission</span> </div>		

Figure 40: Project Submission Section

The users will be able to see information about the assignment. Its name, description, file, creator and publication & due date. Below, they will be able to see the submission file and its submission date (If it has been submitted).

The members of the group will be able to edit and delete the submission with theses Edit Submission Delete Submission buttons (If there is no submission, the upload button will be visible).

**Edit File Submission for Assignment Report** ✕

i Please save your file name as groupNo\_sectionNo\_assignmentName.pdf

File Name:

Choose File

1\_1\_DesignReport.pdf
✕

Edit

Figure 41: Submission Edit Pop-up

With this pop-up, members will upload the submission file with the proper name.

**Grades Section:**

Submission	Grades	Feedbacks																					
<table><tr><th></th><th>Person</th><th>Grade</th></tr><tr><td>Project Instructor</td><td>Eray Tüzün</td><td>9.5/10</td></tr><tr><td>Instructors Average</td><td></td><td>9/10</td></tr><tr><td>TA</td><td>Erdem Tuna</td><td>9/10</td></tr><tr><td>TA</td><td>Elgun Jabrayilzade</td><td>7/10</td></tr><tr><td>Students Avarage</td><td></td><td>6.2/10</td></tr><tr><td>Total Avarage</td><td></td><td>7.1/10</td></tr></table>				Person	Grade	Project Instructor	Eray Tüzün	9.5/10	Instructors Average		9/10	TA	Erdem Tuna	9/10	TA	Elgun Jabrayilzade	7/10	Students Avarage		6.2/10	Total Avarage		7.1/10
	Person	Grade																					
Project Instructor	Eray Tüzün	9.5/10																					
Instructors Average		9/10																					
TA	Erdem Tuna	9/10																					
TA	Elgun Jabrayilzade	7/10																					
Students Avarage		6.2/10																					
Total Avarage		7.1/10																					
<table><tr><th>BilHub Avarage</th><th>Class Avarage</th></tr><tr><td>7.1/10</td><td>6.5/10</td></tr></table>			BilHub Avarage	Class Avarage	7.1/10	6.5/10																	
BilHub Avarage	Class Avarage																						
7.1/10	6.5/10																						
<b>Grade: 36/40</b>																							

Figure 42: Submission Grades Section

The Grades section will show the grades of the submission. It will only be visible after the submission is sent. Project instructor grade and teaching assistants grades will be visible one by one. Other instructors' grades and students' grades will be displayed as an average. Grade at the end of the section shows how much grade that assignment worthed.

## Comments Section:











Submission	Grades	Feedbacks
<b>SRS Result</b>   <p>Final feedback report. </p> <p>Grade: 36/40 -Erdem Tuna, 18 March 2021</p>		
<b>Instructor Feedbacks</b> <p>Good Analysis.</p> <p>Grade: 9.5/10 -Eray Tüzün, 11 March 2021</p>		
<b>Teaching Assistants Feedbacks</b> <p>Hello,</p> <p>I inspected your Readme.md file in Github. Thank you for your effort. Your features are sufficient. I will provide you with few more features. I think some of them are similar to the ones you listed, however, to make sure I am writing them as well. You can expand these ideas for your own use.</p> <ul style="list-style-type: none"> <li>• There should be artifact review functionality.</li> <li>• There should be student dashboard. For instance, it contains the reviews about that student.</li> <li>• Statistics should be displayed.</li> <li>• There are several types of review questions. Maybe some of them could be provided as templates.</li> </ul> <p>Grade: 9/10 -Erdem Tuna, 12 March 2021</p>		
<b>Student Feedbacks</b> <p>In second page, there is no functional analysis. Everything else is just fine.  </p> <p>Grade: 8/10 - Ahmet Yıldırım</p> <p>It seems okay to me. Nice Job</p> <p>Grade: 10/10 - Mehmet Yıldız</p>		
		

Figure 43: Submission Feedbacks Section

Comments section will show the comments that are sent by the users. SRS Result, Instructor's grades, Teaching assistant's grades and student's grades will be separately shown. The comments will be sorted by their dates from newest to oldest.

If the comment made by the current user he will be able to edit and delete his comments via  .

The SRS Result will be editable via   buttons by the teaching assistants or the instructors.

If there is no SRS Result yet the following button will be visible only to instructors and the TA's:

Submission	Grades	Feedbacks
<div>Add SRS Result</div>		

Figure 44: Submission Add SRS Result Button

If the user have not commented yet, he/she will be able to comment via ⊕. Clicking this button will result in following pop-up:

**Comment to BilHub's Analysis Report**

*i* Please write your comment below

The most important aspect of this project is simplicity. I like simple projects. This is one of those.

Add File

**Grade:** 9/10

Comment

Figure 45: Submission Comment Pop-up

Here the user could write his comment and give a grade to the project.

If the user is a teaching assistant or instructor, he/she will be able to add a file as a feedback.

## 6.4.5 Profile Page

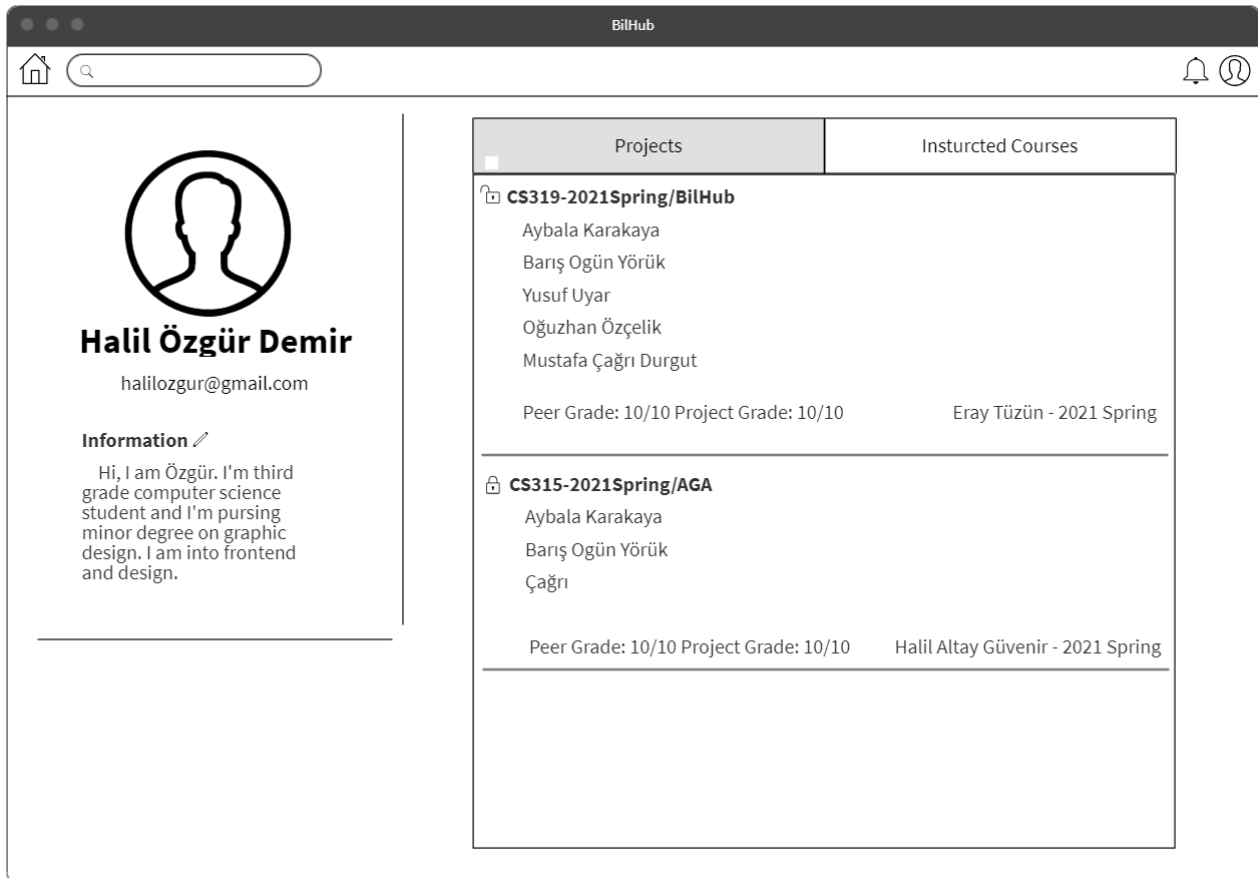



Figure 46: Profile Page

### Left Part of the Page:

In the left side, profile photo, name and the email of a specific user will be shown. Also that specific user will be able to write some sentences about himself in the information part. If the specific user is the current user he/she will see  button to edit his/her information.

### Middle Part of the Page:

In the middle and rest of the page there will be a tab bar with two choices maximum. If the user is only a student only the projects section will be visible. If the user is an instructor, only the instructed courses section will be visible. If a student is additionally TA, then both sections will be visible.

**Projects Section:**




Projects	Insturcted Courses
<div>  <b>CS319-2021Spring/BilHub</b>  Aybala Karakaya  Barış Ogün Yörük  Yusuf Uyar  Oğuzhan Özçelik  Mustafa Çağrı Durgut    Peer Grade: 10/10 Project Grade: 10/10 </div>	Eray Tüzün - 2021 Spring
<div>  <b>CS315-2021Spring/AGA</b>  Aybala Karakaya  Barış Ogün Yörük  Çağrı    Peer Grade: 10/10 Project Grade: 10/10 </div>	Halil Altay Güvenir - 2021 Spring

Figure 47: Profile Projects Section

In the projects section, the projects of the specific user will be shown with their project names, members' names, peer & project grade (if the project is finalized), semester time and instructor name. If the user clicks the project's name, he/she will be sent to the project's specific page. If the user clicks one of the members or the instructor, he/she will be sent to that person profile page.

 => Means project is finished (finalized)

 => Means the project is current (active)

**Instructed Courses Section:**

Projects	Instructed Courses
<b>CS484-2020Fall</b> <b>TA's:</b> Elgun Jabrayilzade, Erdem Tuna <b>Instructor:</b> Ayşe Karaman	

Figure 48: Profile Instructed Courses Section

In the instructed courses section, the instructed courses of a specific user will be shown with their course names and TA & instructor names. If the user clicks the course's name, he/she will be sent to the course's specific page. If the user clicks one of the TA or the instructor, he/she will be sent to that person profile page.



## 6.4.6 Courses Page

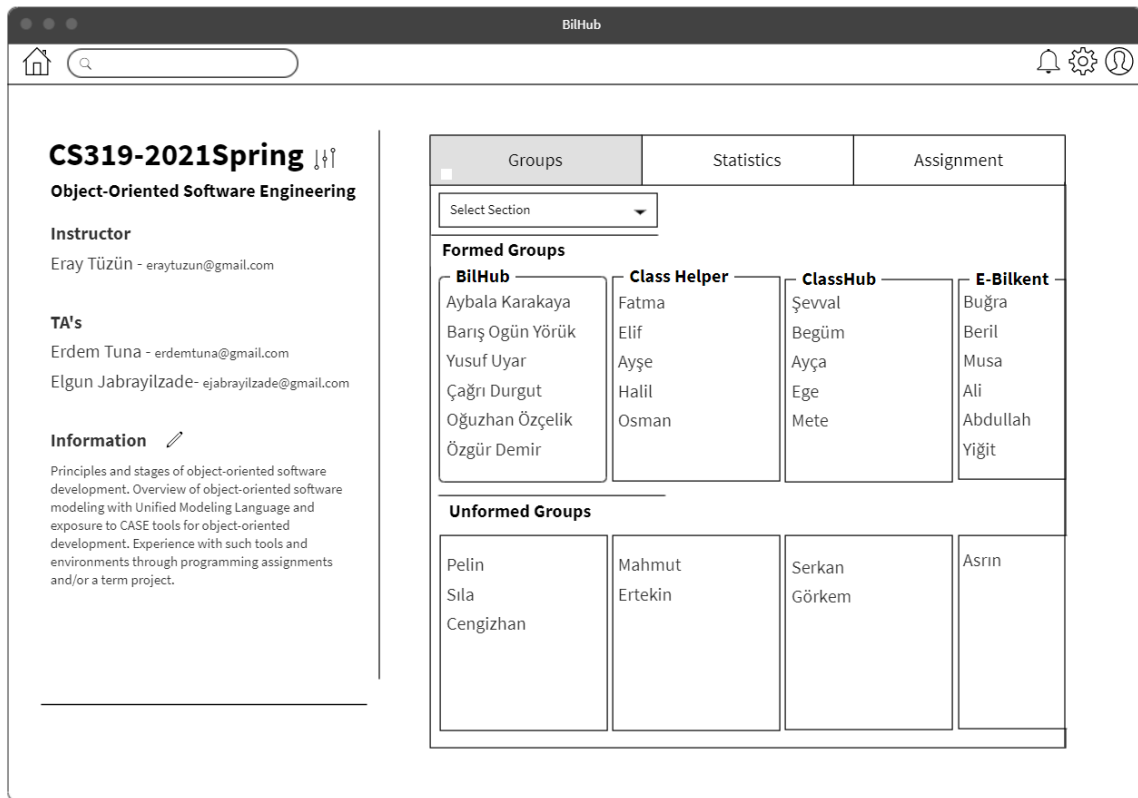


Figure 49: CoursesPage

### Left Part of The Page:

In the upper-part, the user will be able to see code of the course and the name of the course. Instructors may change the course settings via clicking ⚙️. In the middle part, instructors' names and teaching assistant's names will be visible. By clicking the name, the user will be directed to that user's profile page. In the bottom part information regarding the course will be given. Instructors or TA's will be able to change this information by clicking ✎.

### Main Part of The Page:

In the main part of the page, there will be a bar with three sections.

## Groups Section:

Groups	Statistics	Assignment
<div>Select Section ▼</div>		
<b>Formed Groups</b>		
<b>BilHub</b> Aybala Karakaya Barış Ogün Yörük Yusuf Uyar Çağrı Durgut Oğuzhan Özçelik Özgür Demir	<b>Class Helper</b> Fatma Elif Ayşe Halil Osman	<b>ClassHub</b> Şevval Begüm Ayça Ege Mete
<b>E-Bilkent</b> Buğra Beril Musa Ali Abdullah Yiğit		
<b>Unformed Groups</b>		
Pelin Sıla Cengizhan	Mahmut Ertekin	Serkan Görkem
Asrın		

Figure 50: Groups Sections

In the group section, the groups will be given. If the projects have not been started, they will be separated as formed groups and unformed groups. By clicking unformed groups, a student with no group will get a pop-up to join that group.

Request Group Join & Merge

*i* Please write a note and send a request

Hi, we are group that consists of three people. We want to merge our group with you.

Merge Request

Figure 51: Merge Request

Request Group Join

*i* Please write a note and send a request

Hi, I am Barış. I need a group. Please consider me.

Join Request

Figure 52: Join Request Pop-up

If the student is in one of the unformed groups, then he/she will be able to send a merge request. If the student is not in one of the unformed groups, then he/she will be able to send a join request.

If the student is in one of the unformed groups and he did click the his own group he may leave the group with the following pop-up, give ready to form the group and go to merge/join requests:

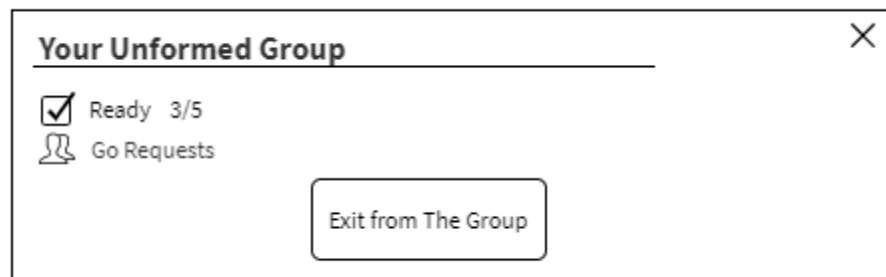


Figure 53: Unformed Group Pop-up

If the student is in one of the formed groups and he did click the his own group he may change his group name:

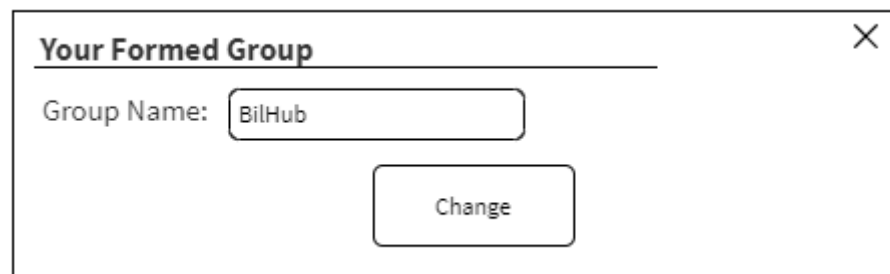


Figure 54: Formed Group

If the course has more than one section than the user will be able to stalk other groups with the dropdown menu above

When the projects has begun, the view of the groups section will be in the following way:

Groups	Statistics	Assignment
<div>Select Section ▼</div>		
<b>Groups</b>		
<b>BilHub</b> Aybala Karakaya Barış Ogün Yörük Yusuf Uyar Çağrı Durgut Oğuzhan Özçelik Özgür Demir	<b>Class Helper</b> Fatma Elif Ayşe Halil Osman	<b>ClassHub</b> Şevval Begüm Ayça Ege Mete
		<b>E-Bilkent</b> Buğra Beril Musa Ali Abdullah Yiğit

Figure 55: Groups Section

By clicking the group name, the user will be able to go to that group's page.

### Statistics Section:

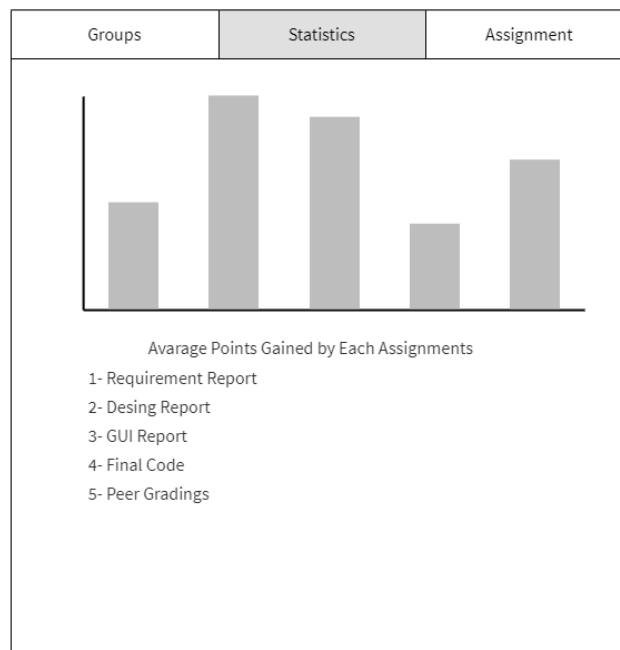


Figure 56: Course Statistics




In the statistics section, the user will be able to see several charts. In the example the user is able to see the average points gained by each assignment.

**Assignment Section:**

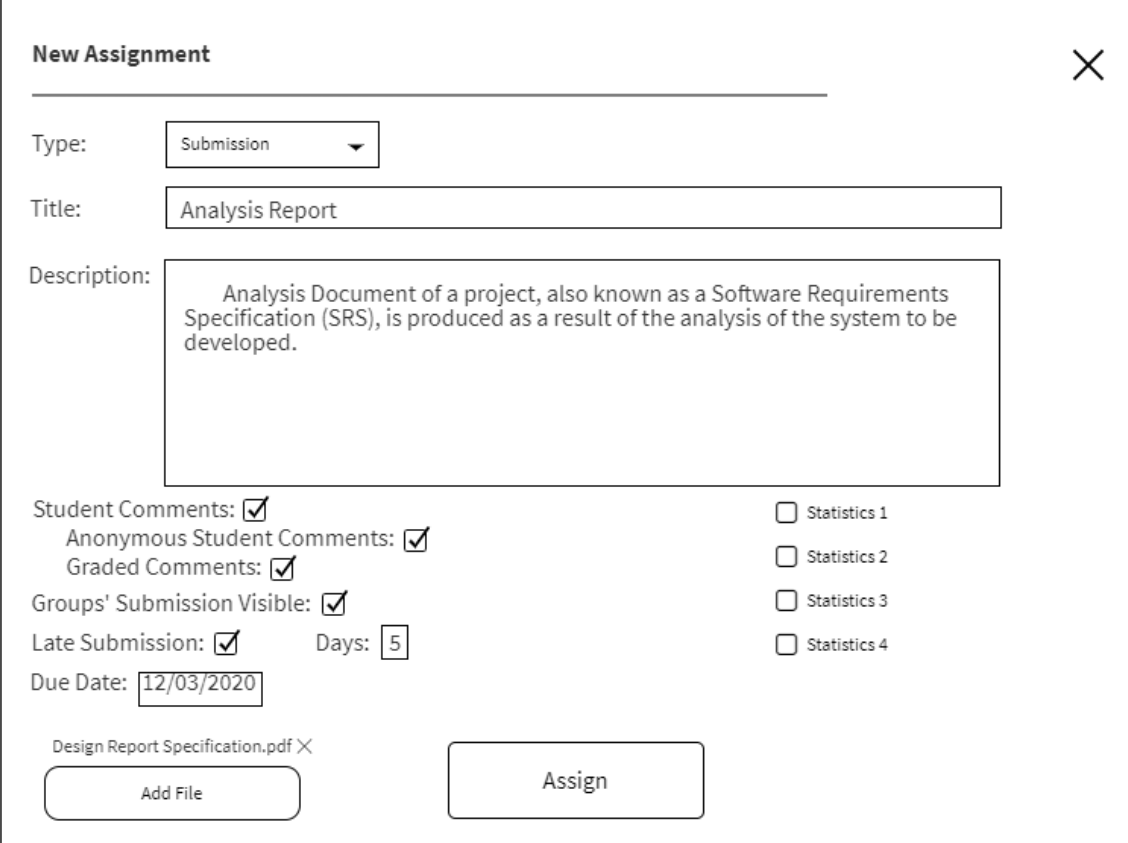
Groups	Statistics	Assignment
<div> <div>CS319 - Analysis Report</div> <div> <div>Analysis Document of a project, also known as a Software Requirements Specification (SRS), is produced as a result of the analysis of the system to be developed.</div> <div>- Erdem Tuna</div> <div>Publishment Date: 12 March 2021 12:00 / Due Date: 16 April 2020, 23:59</div> </div> </div> <hr/> <div> <div>CS319 - Design Report</div> <div> <div>Design Document is mainly a non-exposed, internal document, and is a means for communicating ideas between different parts of a development team during the design phase as well as for proper maintenance of a software system.</div> <div>- Elgun Jabrayilzade</div> <div>Publishment Date: 12 March 2021 12:00 / Due Date: 16 April 2020, 23:59</div> </div> </div> <hr/> <div></div>		

Figure 57: Course Assignment Section

In the assignment section, the users will be able to see all the assignments regarding the course's project. The assignment's name, description, creator, publication & due date will be displayed.

Teaching assistants and instructor will be able to create new assignment by clicking the  button. Also they will be able to change or delete the assignment by using the   buttons.

The new assignment pop-up be like in the following way:



**New Assignment** ✕

---

Type: Submission ▼

Title: Analysis Report

Description: Analysis Document of a project, also known as a Software Requirements Specification (SRS), is produced as a result of the analysis of the system to be developed.

Student Comments: ☒ Anonymous Student Comments: ☒ ☐ Statistics 1

Graded Comments: ☒ ☐ Statistics 2

Groups' Submission Visible: ☒ ☐ Statistics 3

Late Submission: ☒ Days: 5 ☐ Statistics 4

Due Date: 12/03/2020

Design Report Specification.pdf ✕

Add File
Assign

Figure 58: New Assignment Pop-up

This pop-up will be only visible to TA's and instructors. They will be able to choose the type of the assignment. They will be able to give a title, description. If the type is submission, they will also be able to give initial settings to the new assignment and choose which statistics to be shown. If the type is submission or peer grade, they can add a spec file to the new assignment. By clicking the assign button, the new assignment will be visible to all the users.

## 6.4.7 Courses's Assignment Page

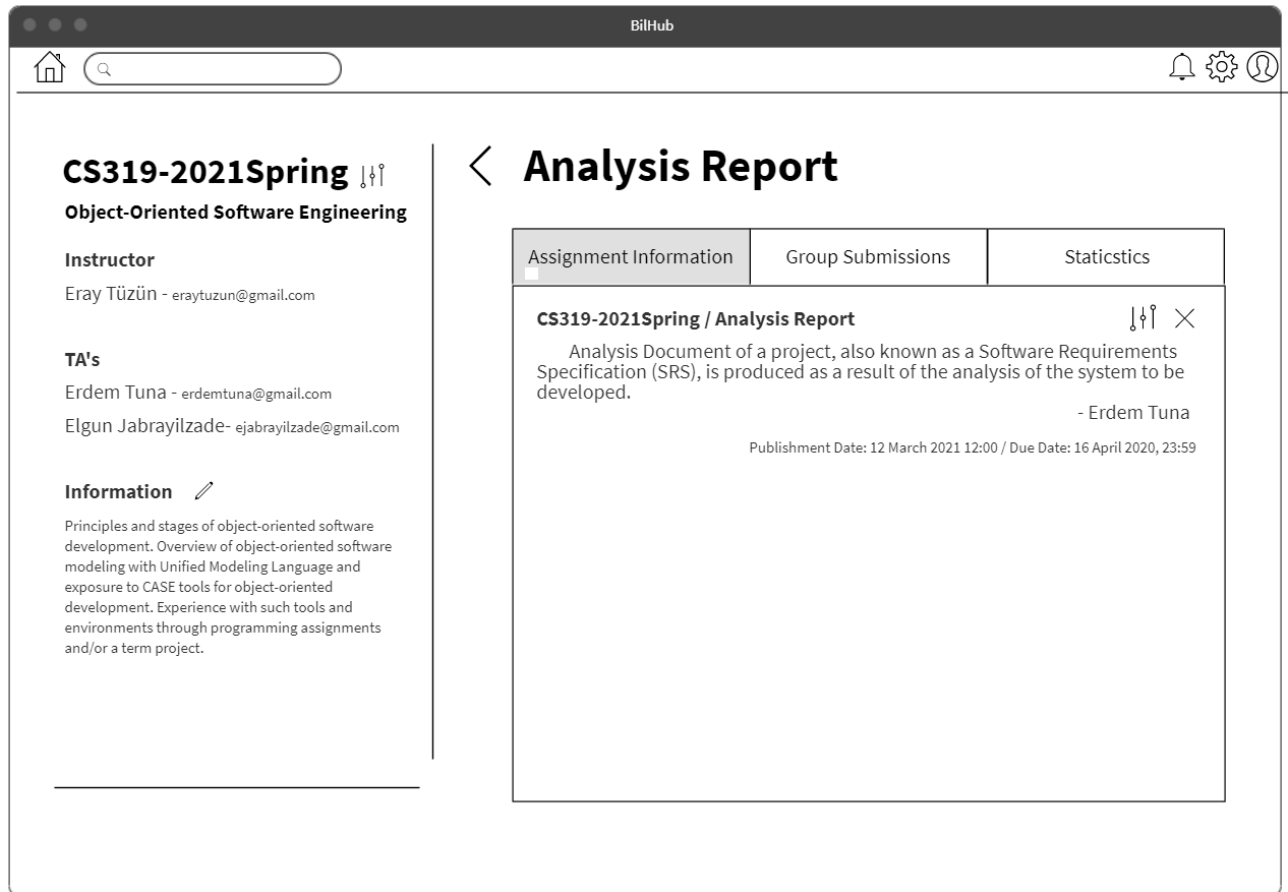


Figure 59: Course Assignment Page

In the course 's assignment the left side is the same with the course's page.

Main part of the page will contain a tab bar with three sections. On the top of it, the assignment name will be visible.

### Assignment Information Section:

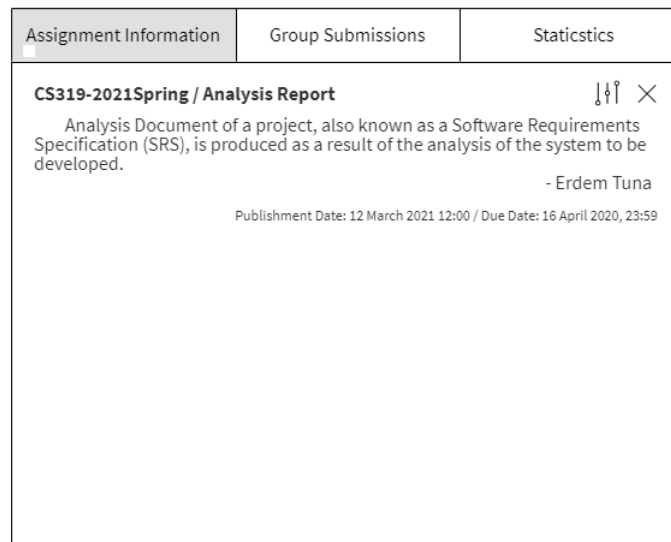


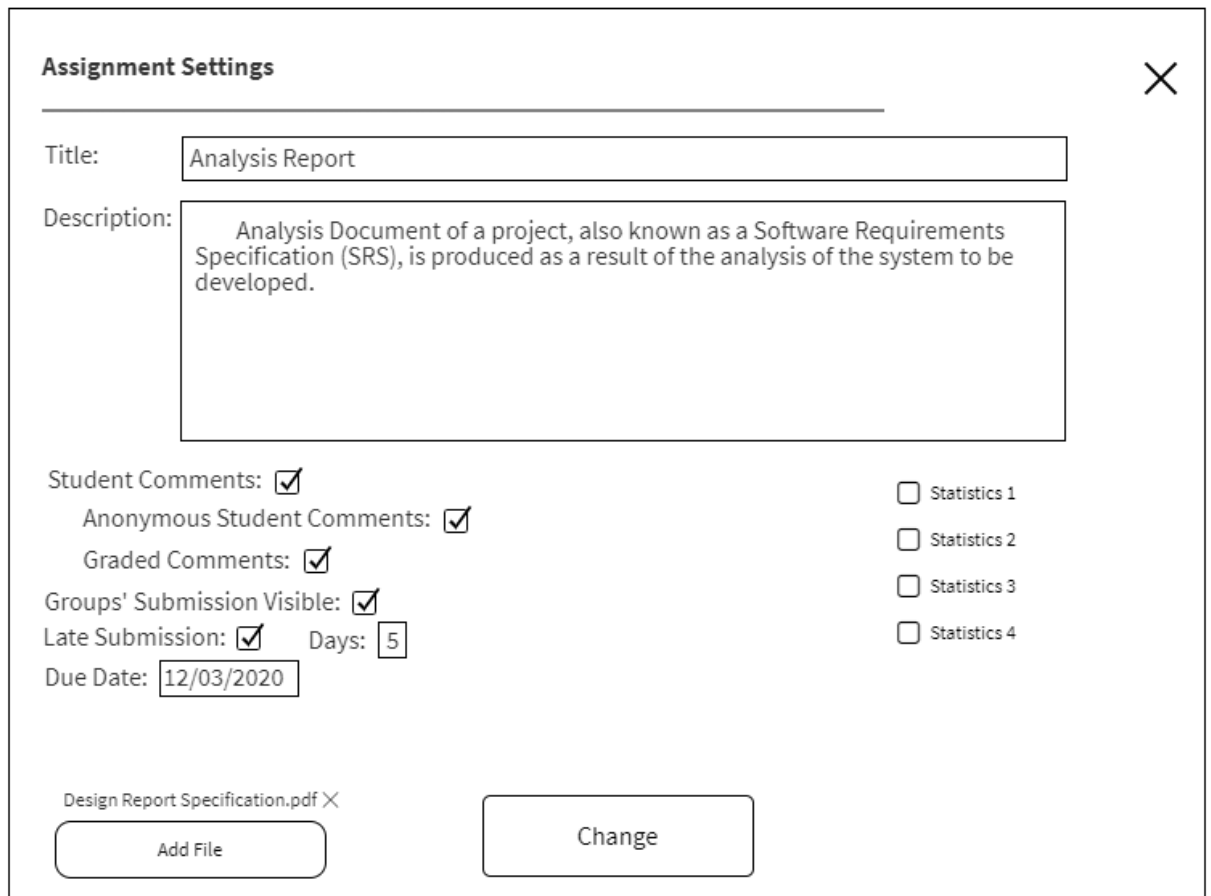


Figure 60: Assignment Information Section

In this section, there will be all the information about the assignment. It's course, name, description, file, creator, publication & due date will be visible. With   buttons, the TA's and the instructors will be able to change or delete the assignment.

Assignment settings pop-up will be like in the following way:



**Assignment Settings** ✕

---

Title:

Description: 

Analysis Document of a project, also known as a Software Requirements Specification (SRS), is produced as a result of the analysis of the system to be developed.

Student Comments: ☒ Anonymous Student Comments: ☒ ☐ Statistics 1

Graded Comments: ☒ ☐ Statistics 2

Groups' Submission Visible: ☒ ☐ Statistics 3

Late Submission: ☒ Days:  ☐ Statistics 4

Due Date:

Design Report Specification.pdf ✕

Add File

Change

Figure 61: Assignment Settings

This pop-up is very similar to the new assignment pop-up, but the user will not be able to change the assignment type.



## Group Submissions Section:






Assignment Information	Group Submissions	Statistics
<b>Graded</b>		
BillHub / 1_1_anaylsisReport.pdf  / Grade: 7/10		15 March 2021
Class Helper / 3_2_anaylsisReport.pdf  / Grade: 9/10		15 April 2021
<b>Submitted</b>		
BillHub / 1_1_anaylsisReport.pdf 		15 March 2021
Class Helper / 3_2_anaylsisReport.pdf 		15 April 2021
ClassHub / 2_2_anaylsisReport.pdf 		16 March 2021
<b>Not Submitted</b>		
AssignOnline		
SubOnline		
ProjectControl		

Figure 62: Group Submissions Section

In the group submissions sections, will consist of three parts. In the graded part, the groups whose submission files graded will be visible with their following grades. In the submitted part, the groups that submit their files will be visible with their group name, file download button and the date submitted. In the not submitted part, the groups that did not submit their files will be visible with their group name. By clicking the group name, the user will be sent to that group's assignment's page.

## Statistics:



Figure 63: Assignment Statistics

In the statistics section, the user will be able to see several charts. In the example the user is able to see the average points gained by each assignment.

## 6.4.8 Create New Course

**Create New Course**

Code:  Year:  Semester:  Name of the Course: CS319-2021Summer

Group Formation Between Sections: ☐ Number of Sections:  THESE CHOICES CANNOT BE CHANGED

Add Instructor: 

example@example.com

example@example.com

example@example.com

+

Add Teaching Assistants: 

example@example.com

example@example.com

example@example.com

+

Add Student as txt file:  Section: 

sec1\_students.txt

sec2\_students.txt

×

Add Student as a list: 

example@example.com

example@example.com

example@example.com

+

Group Formation Type:  ⓘ

Min:  Max:

Group Formation Date:

Figure 64: New Course Page


In this page, instructors will be able to create new courses. Creating a new course consists of four parts. First part, the instructor must enter the code, year and semester of the couser. This will result in the name of course automatically shown in the right part. Second part, the instructor will arrange the number of the sections or whether there should be a section at all (With group formation between sections checkbox). In the third part the instructor will add new instructors, TA's and students to the course. In the last part, the instructor will choose the group formation type and group formation date. By clicking the create new course button, the new course will be created. Other group formation interfaces as follow:




Group Formation Type:	Restricted ▼ ⓘ	
Group Size:	5	<i>Depending on the number of the students, groups may have 6 students.</i>

Group Formation Type:	Hard Coded ▼ ⓘ		
Number of Group Sizes:	3		
Group Size:	5	Number of group with this size:	2
Group Size:	4	Number of group with this size:	1
Group Size:	6	Number of group with this size:	3

Figure 65: Group Formation Settings

## 6.4.9 Course Settings





### Course Settings CS319-2021Summer

Code:  Year:  Semester:  Name of the Course: CS319-2021Summer

---



Add/Remove Instructor:

example@example.com  
example@example.com  
example@example.com

Add/Remove Teaching Assistants:



example@example.com  
example@example.com  
example@example.com

Add Student as txt file:

Add/Remove Student as a list:

example@example.com  
example@example.com  
example@example.com

Section:

---

### Change Groups

Section:

**Groups**

BilHub	Class Helper	ClassHub
Aybala Karakaya	Fatma	Şevval
Başış Ogün Yörük	Elif	Begüm
Yusuf Uyar	Ayşe	Ayça
Çağrı Durgut	Halil	Ege
Oğuzhan Özçelik	Osman	Mete
Özgür Demir		














Figure 66: Course Settings Page

In the course settings page the instructor will be able to make changes regarding courses and groups. In the first part, the instructor will be able to change the course name. In the second part, the instructor will be able to add/remove new instructors, TAs, and students. In the last part, the instructor will be able to change the groups by adding or removing students as well as creating new groups or removing one.

## 6.4.10 Settings

**Settings**

- Profile Settings >
- Account Settings
- Appearance
- Notifications

**Name**

Halil Özgür Demir

*Your instructor most probably wants your full name to be displayed.*

**Information**

Hi, I am Özgür. I'm third grade computer science student and I'm pursuing minor degree on graphic design. I am into frontend and design.

Github: github.com/ozgurdemir/  
 LinkedIn: linkedin.com/ozgurdemir/

*It is recommended that you put your github, linkedin, portfolio links to the description.*

Save Changes

Figure 67: Settings Page

The settings page will be formed out of four tabs.

## Settings

**Settings**

- Profile Settings >
- Account Settings
- Appearance
- Notifications

**Name**

Halil Özgür Demir

*Your instructor most probably wants your full name to be displayed.*

**Information**

Hi, I am Özgür. I'm third grade computer science student and I'm pursuing minor degree on graphic design. I am into frontend and design.

Github: github.com/ozgurdemir/  
 LinkedIn: linkedin.com/ozgurdemir/

*It is recommended that you put your github, linkedin, portfolio links to the description.*

Save Changes

Figure 68: Profile Settings

The first tab will be about profile settings where the user may change his/her name or information.

## Settings

Profile Settings

Account Settings >

Appearance

Notifications

E-mail Address

ozgurdemir@ug.bilkent.edu.tr

New Password

New Password Again

Change Password

You cannot change your bilkent e-mail address

Figure 69: Account Settings Section

In the second section, the user will be able to change his/her password.

## Settings

Profile Settings

Account Settings

Appearance >

Notifications

Dark Mode: ☒

Figure 70: Appearance Section

In the third section, the user will be able to open/close dark mode.

## 6.4.11 Notification Page

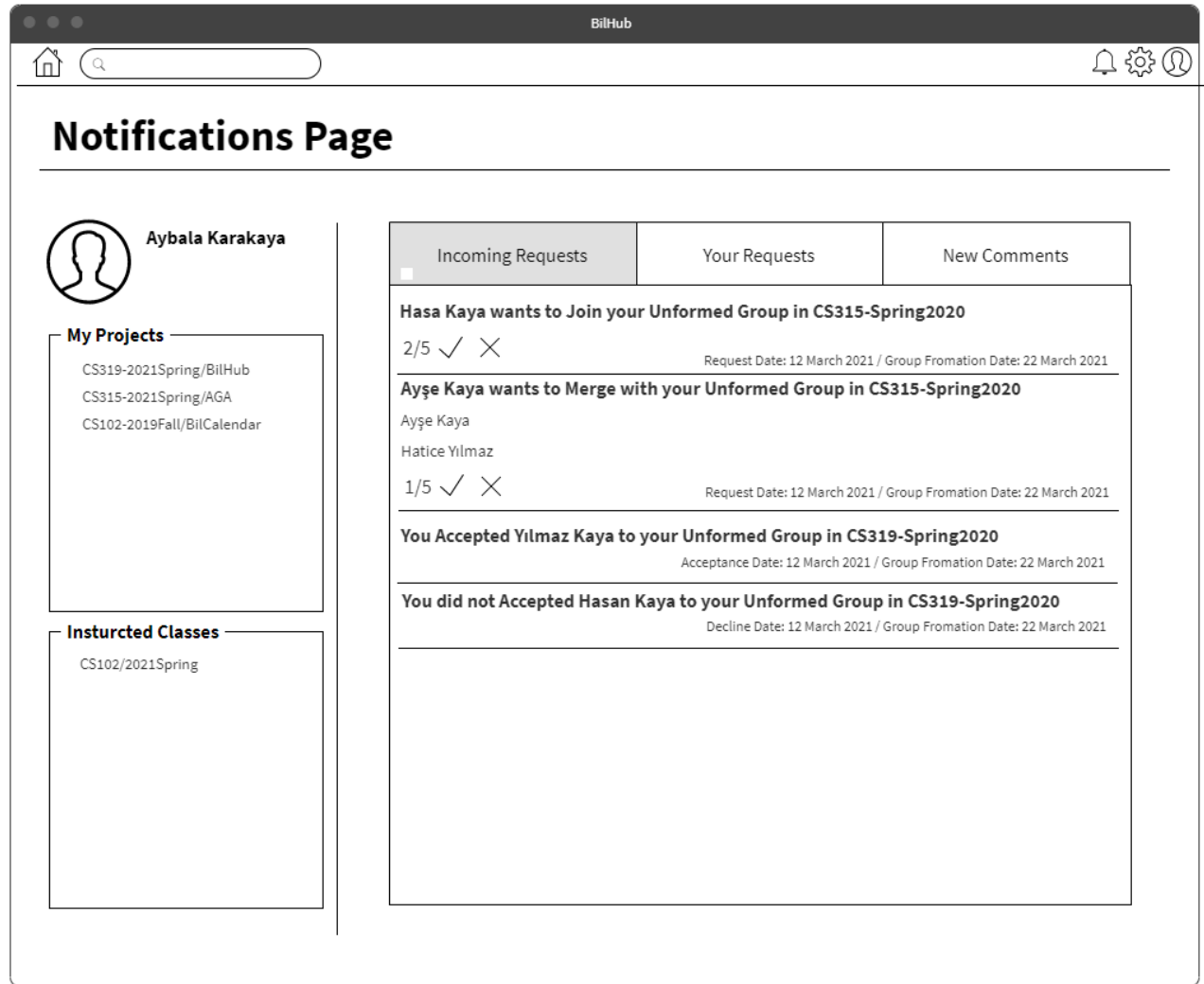


Figure 71: Notifications Page

In the notification page, the left-part of the page will be the same as the main page. In the main part, there will be a top bar with three sections.

## Incoming Requests Section:

Incoming Requests	Your Requests	New Comments
<b>Hasa Kaya wants to Join your Unformed Group in CS315-Spring2020</b> 2/5 ✓ ✕ Request Date: 12 March 2021 / Group Fromation Date: 22 March 2021		
<b>Ayşe Kaya wants to Merge with your Unformed Group in CS315-Spring2020</b> Ayşe Kaya Hatice Yılmaz 1/5 ✓ ✕ Request Date: 12 March 2021 / Group Fromation Date: 22 March 2021		
<b>You Accepted Yılmaz Kaya to your Unformed Group in CS319-Spring2020</b> Acceptance Date: 12 March 2021 / Group Fromation Date: 22 March 2021		
<b>You did not Accepted Hasan Kaya to your Unformed Group in CS319-Spring2020</b> Decline Date: 12 March 2021 / Group Fromation Date: 22 March 2021		

Figure 72: Incoming Requests Section

In the incoming requests page, there will be notifications about the user who wants your join/merge your unformed groups. Here, the user may accept or decline the request. If someone is accepted or declined, it will be also notified here.



## Your Requests Section:

Incoming Requests	Your Requests	New Comments
<b>You Accepted to the Unformed Group in CS315-Spring2020</b> Ayşe Yılmaz Fatma Yıldırım Acceptance Date: 12 March 2021 / Group Formation Date: 22 March 2021		
<b>You Requested to Join an Unformed Group in CS315-Spring2020</b> Ayşe Yılmaz Fatma Yıldırım 1/5 ✓ - Pending Request Date: 12 March 2021 / Group Formation Date: 22 March 2021		

Figure 73: Your Requests Section

In your request section, there will be notifications about your requests to the unformed group. You will be able to see how many people have accepted you. If you accept or decline, you will also get a notification.

## New Comments Section:

Incoming Requests	Your Requests	New Comments
SRS Results		
Instructor Comments		
<b>Eray Tüzün has Commented to your Analiz Report in CS319-2021Spring</b> Nice one Grade: 9.5/10 -Eray Tüzün, 11 March 2021		
<b>Ahmet Hasan has Commented to your Lex Report in CS315-2021Spring</b> Not bad Grade: 9/10 -Ahmet Hasan, 10 March 2021		
<b>Eray Tüzün has Commented to your Desing Report in CS319-2021Spring</b> Super Grade: 9/10 -Eray Tüzün, 11 March 2021		
Teaching Assistant Comments		
Student Comments		

Figure 74: Settings Page

In the new comments section, you will be able to see new comments to your projects or submissions. In the accordion, you may select the whose comments you want to see.

## 7. Glossary & References

### Glossary

**WF:** Workflow

**TA:** Teaching Assistant

**Pseudogroup:** Group formation before finalizing groups. There isn't any restriction on forming pseudogroups.

**Sectionless:** When instructor wants to open a course in which group formation between sections are allowed, instructor opens sectionless course.

### References

Object-Oriented Software Engineering, Using UML, Patterns, and Java, Third Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2010, ISBN: 0-13-606125-7.

Produte. (n.d.). Wireframe tools, prototyping TOOLS, UI Mockups, UX Suite, REMOTE DESIGNING. Retrieved March 16, 2021, from <https://www.mockflow.com/>

Visual Paradigm [Computer software].(Version 16.2.).