Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Peer Review System

Analysis Report

Berke Ceran, Güven Gergerli, Lara Merdol, Akmuhammet Ashyralyyev, Sıla Saraoğlu

Instructor: Eray Tüzün
Teaching Assistant(s): Elgun Jabrayilzade, Erdem Tuna

Contents

1 Introduction

All engineers participate in many small and large-scale projects during their educational life and careers. These can be individual or group projects that are assigned by either instructor or team manager. In the career stage, the projects are divided into small parts that will be assigned to a specific group. Hence, the group members will have to not only collaborate but also with other groups to deliver a good project. To do this, all members have to have good engineering and communication skills. These skills are developed in colleges where students are assigned to course projects. The instructor assigns the project to the group or an individual in the course. Also, for the group projects, members are evaluating other teammates for their performance if the instructor asks them to do so. As a result, these projects help students to develop themself and prepare them for their working life.

The first thing that needs to be done is to establish groups if they are not individual projects. This process can be done by an instructor or students can be allowed to choose their teammates. During the semester, the team will have many deliverables that are needed to be presented to the instructor such as reports or presentations. These tasks are done as a team and each team member will contribute to some extent in this process. At the end of the semester, each member can be asked to evaluate other members about their performance which will be one of the criteria for the instructor for grading. Furthermore, each student of the course is asked to evaluate all other group projects(artifacts) as well. This application aims to ease this process for instructors and students. This application will mainly perform similar tasks that are described above. It will help instructors to create groups for the project, it will assign a peer review form for each member and present it to the instructor as well as other artifacts, and it will give opportunity to students to review other projects of the course as well.

2 Proposed System

2.1 Overview

Peer Review System is a web application that helps both students and instructors for grouping and reviewing projects. Each person that has an account will be able to login into the system where he can join the class as well as to the project groups. Also, if the user has no account on the system, he/she will be able to sign up. The system will be

online, hence, no application will be needed to be installed and all information will be stored online.

2.1.1 Type of Users

The application refers to three kinds of users in its content which are instructor, TA and student. The access of some features or datas in application is open to specific types of users according to their permissions.

2.1.1.1 Instructor

Instructor is the user that can only be assigned by the admin. When the Instructor of the section enrolls, the system automatically recognizes it and converts it's user type to the Instructor. The section can only have one instructor.The web application is designed in a way that maximizes the authority of the Instructor to control students performance. Instructors are able to operate most of the functions that are discussed in section 2.1 such as group formation, reviving systems or even the enrollment of the students in the section. The other important feature that applications provide is the accessibility of the Instructor to the student's personal profiles which provides the information if they do their task or not also their reviews and information. Instructor also has a team of Teaching Assistants that he can designate. That type of user has a permission to review both the TA's and students.

2.1.1.2 Teaching Assistant (TA)

Teaching assistants are the helpers of the Instructor, they have permission to some features that the Instructor can perform such as reviewing students and groups, access to the group list, notify students and view their tasks. There can be more than one TA in the section however delegation of the TA's is performed by the instructor. No other user can become Ta unless they are chosen by the instructor. Students can communicate with the TAs by the chat function for their questions that they do not have to ask the Instructor.

2.1.1.3 Student

The vast majority of the users in the application are composed of students. They have specific tasks that no other type of user has such as group formation, peer review, and uploading tasks. After their enrollment, they are required to form a group. Students have both some individual tasks and group tasks. They can also have access to the other students' profile pages but they cannot see some data such as others reviews. They can use the chat function to communicate with their peers, TA's and the Instructor.

2.1.2 Welcome Page

When the browsers enter the web application, they first encounter the welcome page that includes information about the application. It also contains buttons that navigate browsers to "About", "Terms", "Privacy", "Our Policy", "Contact Us","Sign In" pages. Browsers who want to register to the application or users can go to the sign-in page to enter the system.

2.1.2.1 Sign-up System

Instructors, TA's and students can create their account from the same sign-up segment in the sign-in page. By typing their name, surname student ID, school email address and password that they determine, they can easily register to the system. After their registration, the system assigns their user types and redirects them to the specific dashboard page according to their user type. System allows only one ınstructor per section and this type of user can only be created if admins give permission.

2.1.2.2 Login/ Administration System

Users who create their account earlier can enter the application by the Login segment in the sign-in page. They will be asked for their username and password where their username will be their emails. If the username and password match among the registered users, they will be allowed to enter the system and will be directed to their Dashboard Screen. On the other hand, the login for administration will be directed to a different path after their login since they will have a different interface than other users. Login segment also contains a button called "forget password?" for the users who want to reset their password. Users can change their password by entering their username and initiating the resetting process.

2.1.3 Dashboard Screen

When a user enters the system, he/she will encounter the main dashboard. The dashboard will be different for the instructor, TA's and student. For all of the users, it will have many common features as well as varying ones. The common features will be the search bar, frequently asked questions (FAQ) feature, profile link, course description link, chat link, and the main timeline. By clicking any group title, the user will be navigated to the group profile page. Also, by clicking to profile, the user will be navigated to his/her profile, as well. On the other hand, there will be a chat part which will navigate the user to the chat page if it will be clicked by the user. In the middle of the

page, there will be the main timeline where anyone can write or submit something which will be visible to all people in the same section. On the other hand, there will be other sections where users can see other sections' groups and their work and give artifact review. The search bar will search specific text in the system such as a person, group, project, etc. On the other hand, the main dashboard will have a notification button that will pop up all notifications that are sent to the user by the instructor or TAs.

2.1.3.1 Dashboard Screen For Instructor

For the Instructors, section segments in the dashboard will be separated into the section that they are teaching and other sections that they are not. So, Instructors will be able to browse between the sections. If the instructor navigates to section that he/she teaches, then the main timeline will change according to that section and be able to perform as a instructor, however if he/she navigate other section the domain of the task that he/she can do limited to artifact review and viewing profiles and groups. Start group formation button will be visible to the instructor if he/she teaches in that section. Furthermore, there will be a "review" button next to the group formation button that navigate instructor to the page where he/she can arrange questions for the reviews.

2.1.3.2 Dashboard Screen For TA

Dashboard Screen for TA is mostly similar to the Instructor's, however due to the fact that TAs are not able to activate group formation period and create review questions, in their page there are not any buttons that navigate them to those systems.

2.1.3.3 Dashboard Screen For Students

 For students, the dashboard page has a button that navigates the student to the group formation page if the student is not assigned to a group. Otherwise, for students with an assigned group, the dashboard will have a group profile button that navigates them to their own group. Furthermore, students are able to view other groups' pages by clicking any group title. Students are also able to look at other sections groups and participants and perform artifact review just like other users.

2.1.4 Group Formation Event

Group formation is one of the most crucial events in the application, which ensures the students constitute their groups in an easy way. At the beginning of the semester, after

all students are enrolled in their specific section automatically by the instructor, the instructor enables the group formation event. Students from the same section are able to view each other's profile page for recognizing their classmates by the course list provided to them on the Course Page. Also, students can communicate with other students by the chat feature to get to know each other and finding favorable teammates. All students can create a new group unless the number of groups exceeds the maximum limit of group number. After one participant creates the group, that specific student becomes the leader of the group. Other students can send a request for joining that group. If the request is accepted by the group leader and if the number of participants in the group does not exceed the limiting number determined by the instructor, the student becomes a member of that specific group. Group members can also send requests to the other students at the section to invite them to the group. Students can also exit the group from their group profile page if they withdraw the course or want to change their group. Students that are not leaders can change their groups only in group formation period after that system does not allow them to change. Furthermore, in the case of the situation that there are students who cannot settle down to any group at the end of the group formation period, the instructor is able to interfere with the process and add those students to groups that do not exceed the maximum number of participants. Another type of group formation event that the application provides is random distributive group formation. At the beginning of the period, the instructor can also choose to form groups randomly, in that case, students are automatically located in groups not by joining or creating new groups. The system automatically distributes them with respect to not to exceed the max number of a participant in the group. This function provides the instructor to administer the process in a way that he or she wants.

2.1.5 Review System

Peer Review application presents three kinds of review system in its content to supply all kinds of actors which are TAs, instructors and the students to be reviewed and to give feedback on different scales.

2.1.5.1 Peer Review System

Peer Review System is one of the review systems in the application in which participants in the same group review each other. It is a single-blind review in which the reviewed student does not know who reviewed him or her. The peer-review form is constituted by the instructor. The application provides the instructor to form questions in three different types. While the instructor sets up the questions, he or she can put

multiple-choice, open-ended, or rating questions or can choose a combination of those kinds of questions. After that process instructor posts it to every student in the section. When students receive that form, they are entailed to fill that form for each of the teammates in their group. The instructor is able to control students from the student list to determine if anyone remains that does not fill the form for all teammates. If the instructor detects a student who does not fill the form, the instructor can send a notification to that specific student as a warning purpose. Students can see their teammate's reviews about them on their profile page but cannot see who is the reviewer. Furthermore, students cannot see reviews that other students take too. Peer reviews only can be seen by a reviewed person, reviewer, and instructor.

2.1.5.2 Artifact Review

Artifact Review System is another review system in the application in which groups work is evaluated by every user in the application. This review contains only one question that is a scale question which scale can be determined by the instructor. This application has an attach file feature that group leaders can attach their reports or other informative files for the purpose of Ta's, the instructor, and other students outside the group, can look and review accordingly. This assessment can be given by students in the section to groups only if they do not participate in that group. Group members, instructors, Ta's, and also the other students can see the average assessment point of the group on the specific group page which can be accessed from the dashboard.

2.1.5.3 Lecture Review

Lecture review is the third review system in the application in which students evaluate their instructors, course, and TA's in a single-blind review. The type of the form is similar to the form which is used in the peer review system. Questions for that assessment are chosen by the instructor. The application provides the instructor three different types of questions to form the review. While the instructor sets up the questions, he or she can put multiple-choice, open-ended, or rating questions or can choose a combination of those kinds of questions. After that process instructor posts it to every student in the section. It is not a mandatory assignment; students can fill or not fill the form. The purpose of that review is to take feedback from students to detect some errors about the course and also to assess Ta's and fix possible problems for the next semesters. By the Lecture Review system, not only students, but all users in the system also find a chance to take feedback about their performance during the semester. Lecture reviews

can only be seen by the Instructor and TA's on the course page but they can just see the review, not the reviewer name.

2.1.6 Message System

All of the users; instructors, teaching assistants, and students are going to be able to send messages preferably to one or more users if they are members of the same section. Messages are going to include the information of message sending time and the name of the user who sent the message. Users are going to be able to see the messages sent by them on the right side of the screen and the messages sent by other users on the left side of the screen which is going to be more convenient for the user to track. Moreover, it will be optional for a user to delete one or more messages that are specified by them so that the user will not see the message anymore. In addition, a user also has an option to delete the whole chat log from their chats. However, if a user prefers to delete messages or chat logs, it is not going to be deleted in other users' messages or chat logs but only going to be deleted from the user who preferred to delete them. All of the chat logs that the user has can be found in the message section of the application and when a chat is clicked, all of the chat logs are displayed to the user. Application also has a function that allows users to like each other's message and reply to it.

2.1.6.1 Private Chat

Any user is allowed to start a private chat with another user if both are members of the same section. All of the messages sent in the private chat will be only accessible and visible to 2 members of the chat. A private chat is initiated between two users by a message sent by one user to another. After the chat is initiated, the chat can be found on the messages section of the application for both users. When entered into a private chat, the information of the user that is in the chat will be available on the top of the chat and when it is clicked by the user, they will be automatically transferred to the other user's profile page.

2.1.6.2 Group Chat

A group chat is a type of chatting where there can be more than two users on the same chat. Group chat is only doable between the group members. Users can not create new chat groups with other students. Every group in the section has a chat group by default. When a message sent by a user is sent to the group chat, every member of the group can access the message. Every message consists of the name of the user that sent the message and the sent time of the message. Same as private chat, the messages sent by

the other users are displayed on the right and the messages sent by the user themselves are displayed on the left. The name of the group will be available on the top of the chat and when it is clicked by the user, they will be automatically transferred to the group information which consists of the name of the group and each group member. When any of the group members is clicked, the user will be automatically transferred to the profile of the user.

2.1.7 Search Page

There will be a search bar at the top of the application which can be written text on it. The bar will be available anytime on the top of the application all the time. When a user clicks on the search bar, type a keyword, and clicks enter, the search results will be displayed on the search page. Students can search users, groups, artifacts, and announcements. For instance, when the name of an instructor, teaching assistant, or a student is typed into the search bar, the searched user is given a clickable profile on the search page. When there is more than one result in the search, all of the results are listed one after another.

2.1.8 Profile Page

Every user; namely instructors, teaching assistants, and students have a unique personal profile page. It consists of the name of the user, profile picture, other written information about them, the group they are in, and the artifact they made (if exists). Also, another type of profile page is the group profile page. A group profile page is created when the groups are formed and consist of information about the group name, written information about the group, the members of the group, and the teaching assistants that are assigned to that group. The common artifact if it exists will be also displayed on the group profile page.

2.1.8.1 Personal Profile Page

Any personal page can be found via, private or group chats, search page, group profile page, and artifact members. The personal page consists of the information whether the user is an instructor, a teaching assistant, or a student. The personal profile pages will be editable by its user. The name of the user is on the top and every user has an option to add written information below it such as their GitHub or Gmail links so that other users can easily access it. Apart from that, if a student user has formed a group the information of the group is also displayed on the profile page which is clickable so that from a

personal profile page any user can access the group profile page with ease. When a group shares their artifact, it also will be available on every personal profile page.

2.1.8.2 Group Profile Page

A group profile page is created when a group is formed by the students. The group profile page consists of a group name at the top and an option to add written information below it such as the group's GitHub or Gmail document links so that other users can easily access it. Also, there will be a list of the students that are in that group which all of them are going to be clickable so that every group member's personal profile page is accessible on the group profile page. Just as student group members', there are also the instructors and teaching assistants that are responsible for the group will have their profile pages clickable. When the group shares an artifact it will be displayed on their group profile page. Group profile page can be only edit by the group leader.

2.1.9 FAQ Page

There will be a Frequently Asked Question button, or mostly mentioned as FAQ, at the top of the application near the search bar. The FAQ button will be accessible from everywhere and whenever the user clicks on the FAQ button, they are going to be directed to the FAQ page. The FAQ page consists of an easy tutorial of the application divided into sections so that a user who wants to learn something about a specific feature of the application can easily find it. Below the tutorials, there will be the most frequently asked question and the answer to them. If a student cannot find the question in the FAQ section will also have an option to send a question to the administration which will be below the FAQ section. Just by typing the question into the text field and clicking send, the message question will be delivered to the administration. When the administration answers the question, both the question and the answer will be added anonymously to the FAQ so that other students can have access to the answer.

2.2 Requirements

2.2.1 Functional Requirements

1. Home: The home page of the application-which users will see first when they enter the application- consists of the main information about the application, sign up, sign out, help, and credits links.

2. Administration: The users will be upgraded to instructor by the administration. Then, the instructor will provide the course's TAs' and students' emails to the system to permit them to administrate into the system. If the instructor does not provide a students' email, that student cannot access the system. Then, all the other users need to administrate to the system to sign in to the application.

3. Sign In: After a user is administered to the system, that user should log in to the system to access other system functionalities.

4. Frequently Asked Questions: For users with no prior knowledge of the application, there will be a tutorial and frequently asked question page to inform the user about the features of the application and how to apply each of them.

5. Credits: This option will include a little introduction of developers with their contact information.

2.2.1 Non-Functional Requirements

1. Extensibility:

The student evaluation system we will build will be designed in a way that can be expanded and can be easily added to new things, so when new elements or features are desired to be added, new features can be added without much trouble and major changes on the main code. Object-oriented design concepts will be used to make a project so flexible and easily developed.

2. Maintainability:

The codes will be coded in a maintainable way for the programmers who want to add to the source code of the student evaluation system or want to understand the source codes. Code block features such as what the code blocks are or what are they associated with will be clearly stated within the comment blocks.

3. Usability:

The student evaluation system we will make will have a user-friendly interface that can be easily followed and understood by users who will use the system. The keywords or various feature names to be used in this interface will be chosen from the words or explanations that users frequently use or encounter in daily life and it will be aimed for users to follow them comfortably.

4.Reliability:

Private data such as various private information of users, uploads they make into the system, messaging channels, or various notifications will be stored in an online database and these private data will not be shared with anyone and will not be lost for any reason such as power loss, connection down and sign out.

3 System Models

3.1 Use-Case Model
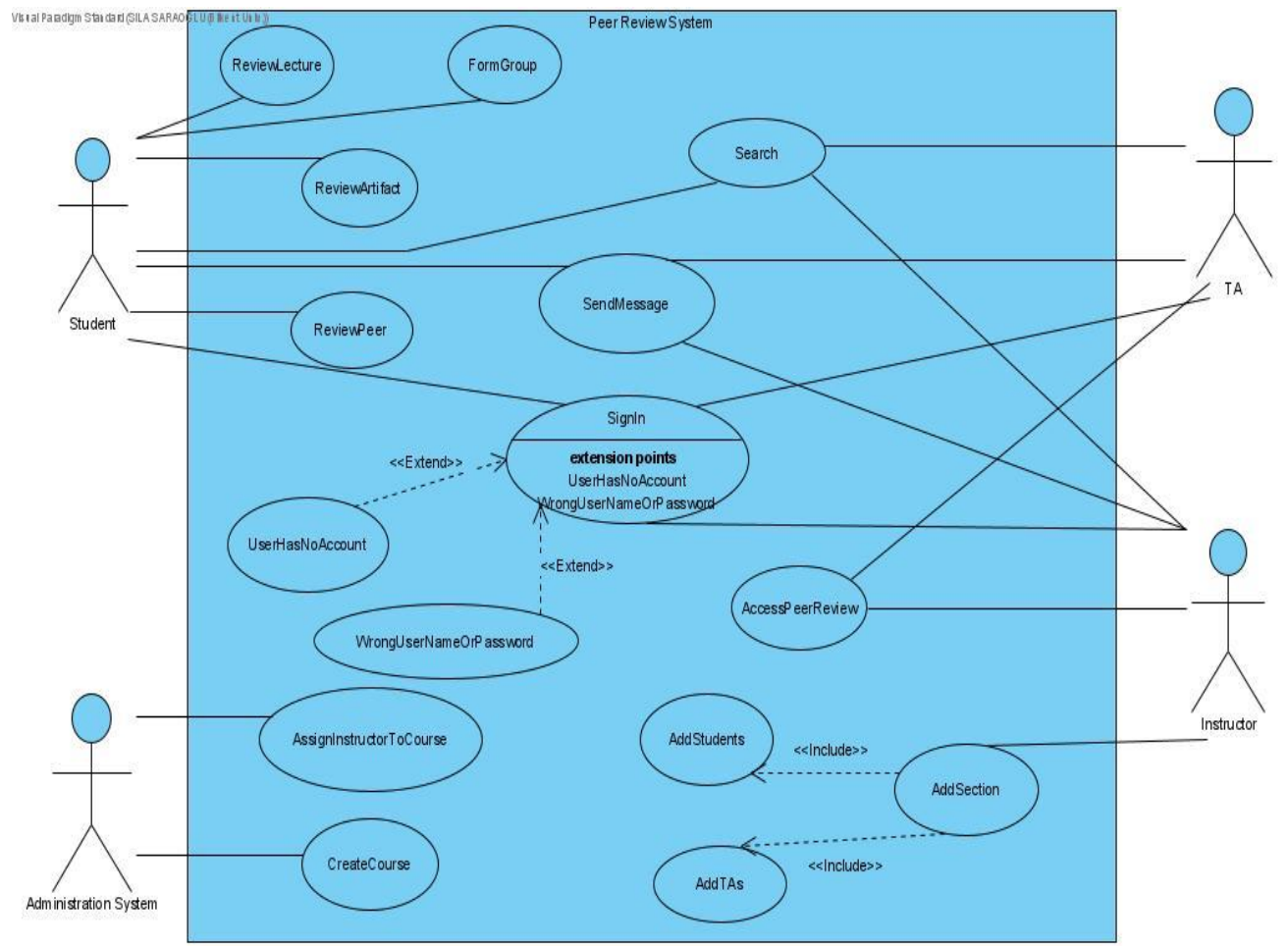
Use Case#1

1.Name: FormGroup

2.Participating Actor: Student

3. Entry Condition:
   ● Student does not have a group already.

4.Exit Condition:

- Student has a group.

5. Flow of Events:
   1. The student creates a group and adds its name.
   2. The student sends requests to other students in the course list.
   3. If other students accept, he/she is added to the group.
   4. If the number of members reaches the maximum limit of students for a group, the member adding process stops.

6. Special Requirements

The group formation process stops after the deadline specified by the instructor.

Use Case#2

1.Name: ReviewArtifact

2.Participating Actor: Student

3. Entry Condition:
   - Student does not belong to the group which created the artifact.

4.Exit Condition:
   - Student reviewed the artifact.

5. Flow of Events:
   1. Student examines the artifact by looking at the attached file for the artifact.
   2. Student writes his/her feedback for the artifact.
   3. Student sends the feedback.

6. Special Requirements

The feedback should be anonymous and does not include any slang language.

Use Case#3

1.Name: ReviewPeer

2.Participating Actor: Student

3. Entry Condition:
   - The Student and reviewee are in the same group.

4.Exit Condition:
   - Student reviewed his/her peer in the group.

5. Flow of Events:
   1. Student writes his/her feedback for the peer.
   2. Student sends the feedback.

6. Special Requirements

The feedback should be anonymous and does not include any slang language.

Use Case#4

1.Name: ReviewLecture

2.Participating Actor: Student

3. Entry Condition:
   ● Student belongs to the course.

4.Exit Condition:
   ● Student reviewed the course.

5. Flow of Events:
   3. Student writes his/her feedback for the lecture.
   4. Student sends the feedback.

6. Special Requirements

The feedback should be anonymous and does not include any slang language.

Use Case#5

1.Name: SendMessage

2.Participating Actor: Student, TA, Instructor

3. Entry Condition: -

4.Exit Condition:
   ● Actor sends the message.

5. Flow of Events:
   1. Actor selects the group or person to send a message to.
   2. Actor writes the message.
   3. Actor sends the message.

6. Special Requirements

The message transportation should not take longer than one minute.

Use Case#6

1.Name: Search

2.Participating Actor: Student, TA, Instructor

3. Entry Condition: Actor searches a valid name that is present in the system.

4.Exit Condition:
   ● Actor finds the searched student/TA/Instructor/Group.

5. Flow of Events:
   1. Actor writes the name he/she wants to search.
   2. Actor sees the information about the name he/she searched for.

6. Special Requirements

The time between the search process and providing the information about the searched name should take no longer than one minute.

Use Case#7

1.Name: SignIn

2.Participating Actor: Student, TA, Instructor

3. Entry Condition: Actor is already registered to the system.

4.Exit Condition:

- Actor is signed in.

5. Flow of Events:

1. Actor writes his/her email and password.
2. If the email and password is registered in the system, the actor signs in to the system.

6. Special Requirements

The sign in process should not take longer than one minute.

Use Case#8

1.Name: UserHasNoAccount

2.Participating Actor: Communicates with Student, TA, Instructor

3. Entry Condition: This use case extends SignIn use case. It is initiated by the system whenever the actor is not registered in the system.

4.Exit Condition:

- Actor returns back to the SignIn use case.

5. Flow of Events:

1. System navigates the user to the signing up process.
2. System asks for the information needed for the signing up process.
3. User enters his/her information.
4. Systems registers the user to the system and displays the acknowledgement to the user.

6. Special Requirements

The email the user uses to sign up must be valid and not already in the system. The password needs to include numbers and letters.

Use Case#9

1.Name: WrongUserNameOrPassword

2.Participating Actor: Communicates with Student, TA, Instructor

3. Entry Condition: This use case extends SignIn use case. It is initiated by the system whenever the actor enters a wrong email or password.

4.Exit Condition:

- Actor returns back to the SignIn use case.

5. Flow of Events:

1. System sends acknowledgement to the user that the email or password is wrong.

6. Special Requirements

Use Case#10

1.Name: AssignInstructorToCourse

2.Participating Actor: Administration System

3. Entry Condition:

The particular course is created, the instructor has not been assigned to this course yet.

4.Exit Condition:

Instructor has assigned the particular course.

5. Flow of Events:

1. The Administration System assigns an Instructor to a particular course which was created before.

6. Special Requirements:

Instructors should be assigned to a particular course at the beginning of the semester. The instructor assigned to the course can access the course in the duration of the specific course period.

Use Case#11

1.Name: CreateCourse

2.Participating Actor: Administration System

3. Entry Condition:

A course that is not in the system can be created.

4.Exit Condition:

A course is created.

5. Flow of Events:

1. The administration system creates a new course which is not in the system.

6. Special Requirements: The courses to be created will be held during the school term.

Use Case#12

1.Name: AccessPeerReview

2.Participating Actor: TA, Instructor

3. Entry Condition:·

There is a peer assessment document entered by the students into the system.

4.Exit Condition:

TA and Instructor have accessed peer reviews.

5. Flow of Events:

1.Instructor or TA go to their personal profile page.

2. When Instructors or TA open their personal profile page, they access peer reviews of students.

6. Special Requirements -

Use Case#13

1.Name: AddStudents

2.Participating Actor: Instructor

3. Entry Condition:·

The course to which the student will be added must have been created and the student has not already been assigned.

4.Exit Condition:·

Students are assigned to a particular course.

5. Flow of Events:

1. The instructor upgrades a user to a student.

2. The instructor adds this student to the course.

3. The instructor adds the student to the section.

6. Special Requirements:

Adding students to a specific course should be done before the instructor loads various tasks.

Use Case#14

1.Name: AddSection

2.Participating Actor: Instructor

3. Entry Condition:

A section that is not in the system can be added to the system.

4.Exit Condition:

A section is added to the particular course.

5. Flow of Events:

1. The Instructor adds different sections to a particular course.

6. Special Requirements:

Section addition process is determined by the number of students.

Use Case#15

1.Name: AddTAs

2.Participating Actor: Instructor

3. Entry Condition: The course to which the TA will be added must have been created and the TA is not already to this course.

4.Exit Condition:

      TAs are assigned to particular courses.
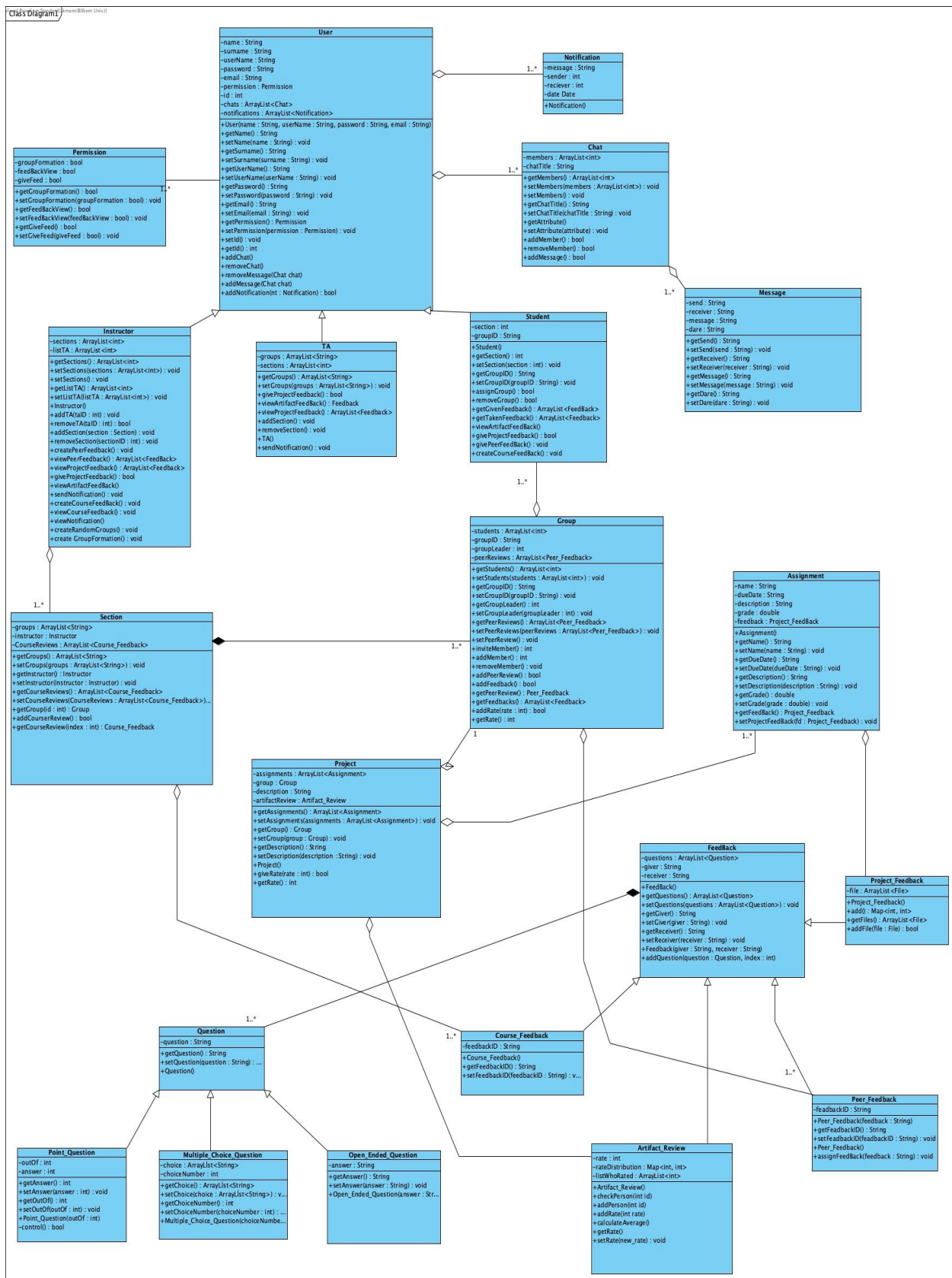
5. Flow of Events:

      1. The instructor upgrades a user to a TA.

      2. The instructor adds this TA to a course.

6. Special Requirements

Adding TA to a specific course should be done before the instructor loads various tasks.

3.2 Object and Class Model

User
- -name : String
- -surname : String
- -userName : String
- -password : String
- -email : String
- -permission : Permission
- -id : int
- -chats : ArrayList<Chat>
- -notifications : ArrayList<Notification>
- +User(name : String, userName : String, password : String, email : String)
- +getName() : String
- +setName(name : String) : void
- +getSurname() : String
- +setSurname(surname : String) : void
- +getUserName() : String
- +setUserName(userName : String) : void
- +getPassword() : String
- +setPassword(password : String) : void
- +getEmail() : String
- +setEmail(email : String) : void
- +getPermission() : Permission
- +setPermission(permission : Permission) : void
- +setId() : void
- +getId() : int
- +addChat()
- +removeChat()
- +removeMessage(Chat chat)
- +addMessage(Chat chat)
- +addNotification(nt : Notification) : bool

Notification
- -message : String
- -sender : int
- -reciever : int
- -date Date
- +Notification()

Permission
- -groupFormation : bool
- -feedBackView : bool
- -giveFeed : bool
- +getGroupFormation() : bool
- +setGroupFormation(groupFormation : bool) : void
- +getFeedBackView() : bool
- +setFeedBackView(feedBackView : bool) : void
- +getGiveFeed() : bool
- +setGiveFeed(giveFeed : bool) : void

Chat
- -members : ArrayList<int>
- -chatTitle : String
- +getMembers() : ArrayList<int>
- +setMembers(members : ArrayList<int>) : void
- +setMembers() : void
- +getChatTitle() : String
- +setChatTitle(chatTitle : String) : void
- +getAttribute()
- +setAttribute(attribute) : void
- +addMember() : bool
- +removeMember() : bool
- +addMessage() : bool

Message
- -send : String
- -receiver : String
- -message : String
- -dare : String
- +getSend() : String
- +setSend(send : String) : void
- +getReceiver() : String
- +setReceiver(receiver : String) : void
- +getMessage() : String
- +setMessage(message : String) : void
- +getDare() : String
- +setDare(dare : String) : void

Instructor
- -sections : ArrayList<int>
- -listTA : ArrayList<int>
- +getSections() : ArrayList<int>
- +setSections(sections : ArrayList<int>) : void
- +setSections() : void
- +getListTA() : ArrayList<int>
- +setListTA(listTA : ArrayList<int>) : void
- +Instructor()
- +addTA(taID : int) : void
- +removeTA(taID : int) : bool
- +addSection(section : Section) : void
- +removeSection(sectionID : int) : void
- +createPeerFeedback() : void
- +viewPeerFeedback() : ArrayList<Feedback>
- +viewProjectFeedback() : ArrayList<Feedback>
- +giveProjectFeedback() : bool
- +viewArtifactFeedBack()
- +sendNotification() : void
- +createCourseFeedBack() : void
- +viewCourseFeedback() : void
- +viewNotification()
- +createRandomGroups() : void
- +create GroupFormation() : void

TA
- -groups : ArrayList<String>
- -sections : ArrayList<int>
- +getGroups() : ArrayList<String>
- +setGroups(groups : ArrayList<String>) : void
- +giveProjectFeedback() : bool
- +viewArtifactFeedBack() : Feedback
- +viewProjectFeedBack() : ArrayList<Feedback>
- +addSection() : void
- +removeSection() : void
- +TA()
- +sendNotification() : void

Student
- -section : int
- -groupID : String
- +Student()
- +getSection() : int
- +setSection(section : int) : void
- +getGroupID() : String
- +setGroupID(groupID : String) : void
- +assignGroup() : bool
- +removeGroup() : bool
- +getGivenFeedback() : ArrayList<Feedback>
- +getTakenFeedback() : ArrayList<Feedback>
- +viewArtifactFeedBack()
- +giveProjectFeedback() : bool
- +givePeerFeedback() : void
- +createCourseFeedBack() : void

Group
- -students : ArrayList<int>
- -groupID : String
- -groupLeader : int
- -peerReviews : ArrayList<Peer_Feedback>
- +getStudents() : ArrayList<int>
- +setStudents(students : ArrayList<int>) : void
- +getGroupID() : String
- +setGroupID(groupID : String) : void
- +getGroupLeader() : int
- +setGroupLeader(groupLeader : int) : void
- +getPeerReviews() : ArrayList<Peer_Feedback>
- +setPeerReviews(peerReviews : ArrayList<Peer_Feedback>) : void
- +setPeerReview() : void
- +inviteMember() : int
- +addMember() : int
- +removeMember() : void
- +addPeerReview() : bool
- +addFeedback() : bool
- +getPeerReviews() : Peer_Feedback
- +getFeedbacks() : ArrayList<Feedback>
- +addRate(rate : int) : void
- +getRate() : int

Assignment
- -name : String
- -dueDate : String
- -description : String
- -grade : double
- -feedback : Project_Feedback
- +Assignment()
- +getName() : String
- +setName(name : String) : void
- +getDueDate() : String
- +setDueDate(dueDate : String) : void
- +getDescription() : String
- +setDescription(description : String) : void
- +getGrade() : double
- +setGrade(grade : double) : void
- +getFeedBack() : Project_Feedback
- +setProjectFeedBack(fd : Project_Feedback) : void

Section
- -groups : ArrayList<String>
- -instructor : Instructor
- -CourseReviews : ArrayList<Course_Feedback>
- +getGroups() : ArrayList<String>
- +setGroups(groups : ArrayList<String>) : void
- +getInstructor() : Instructor
- +setInstructor(Instructor : Instructor) : void
- +getCourseReviews() : ArrayList<Course_Feedback>
- +setCourseReviews(CourseReviews : ArrayList<Course_Feedback>)...
- +getGroupId() : int) : Group
- +addCourserReview() : bool
- +getCourseReview(index : int) : Course_Feedback

Project
- -assignments : ArrayList<Assignment>
- -group : Group
- -description : String
- -artifactReview : Artifact_Review
- +getAssignments() : ArrayList<Assignment>
- +setAssignments(assignments : ArrayList<Assignment>) : void
- +getGroup() : Group
- +setGroup(group : Group) : void
- +getDescription() : String
- +setDescription(description : String) : void
- +Project()
- +giveRate(rate : int) : bool
- +getRate() : int

FeedBack
- -questions : ArrayList<Question>
- -giver : String
- -receiver : String
- +FeedBack()
- +getQuestions() : ArrayList<Question>
- +setQuestions(questions : ArrayList<Question>) : void
- +getGiver() : String
- +setGiver(giver : String) : void
- +getReceiver() : String
- +setReceiver(receiver : String) : void
- +Feedback(giver : String, receiver : String)
- +addQuestion(question : Question, index : int)

Project_Feedback
- -file : ArrayList<File>
- +Project_Feedback()
- +add() : Map<int, int>
- +getFiles() : ArrayList<File>
- +addFile(file : File) : bool

Question
- -question : String
- +getQuestion() : String
- +setQuestion(question : String) : ...
- +Question()

Course_Feedback
- -feedbackID : String
- +Course_Feedback()
- +getFeedbackID() : String
- +setFeedbackID(feedbackID : String) : v...

Peer_Feedback
- -feadbackID : String
- +Peer_Feedback(feedback : String)
- +getFeedbackID() : String
- +setFeadbackID(feedbackID : String) : void
- +Peer_Feedback()
- +assignFeedBack(feedback : String) : void

Point_Question
- -outOf : int
- -answer : int
- +getAnswer() : int
- +setAnswer(answer : int) : void
- +getOutOf() : int
- +setOutOf(outOf : int) : void
- +Point_Question(outOf : int)
- -control() : bool

Multiple_Choice_Question
- -choice : ArrayList<String>
- -choiceNumber : int
- +getChoice() : ArrayList<String>
- +setChoice(choice : ArrayList<String>) : v...
- +getChoiceNumber() : int
- +setChoiceNumber(choiceNumber : int) : ...
- +Multiple_Choice_Question(choiceNumbe...

Open_Ended_Question
- -answer : String
- +getAnswer() : String
- +setAnswer(answer : String) : void
- +Open_Ended_Question(answer : Str...

Artifact_Review
- -rate : int
- -rateDistribution : Map<int, int>
- -listWhoRated : ArrayList<int>
- +Artifact_Review()
- +checkPerson(int id)
- +addPerson(int id)
- +addRate(int rate)
- +calculateAverage()
- +getRate()
- +setRate(new_rate) : void

1-Permission:

The Permission Class is used to determine which tasks 3 types of users registered in the application will be assigned (Student, Instructor, or TA).

2-User:

In the application, the User Class makes users register or access their own pages after registration. According to the permission class, users can be of 3 types, student, TA, or instructor.

3-Notification:

The Notification Class enables various notifications to be created and sent. It will also store the message.

4-Chat:

Chat Class allows users in the application to message with each other or to send and receive files with each other.

5-Instructor:

The Instructor Class is specific to the instructor and allows the instructor-related work to be done.

6-TA:

The TA Class is specific to the instructor and allows the instructor-related work to be done.

7-Student:

The Student Class is specific to the instructor and allows the instructor-related work to be done.

8-Message:

Message Class shows the properties of the messaging channel in the Chat class.

9-Section:

Section Class is used to determine the various sections determined by the instructor and their various features.

10-Group:

Group Class allows the various students to perform tasks related to the groups they have and the characteristics of these groups.

11-Assignment:

The Assignment Class helps the instructor to set tasks or project-related responsibilities to students or various groups.

12-Project:

Project class determines the properties of projects owned by various groups.

13-FeedBack:

FeedBack Class enables the instructor to give feedback to various groups about their work and tasks. This Class is divided into 3 separate private classes.

14-Project_Feedback:

Project_Feedback Class contains feedback related to the projects owned by the groups.

15-Course_Feedback:

Course_Feedback Class contains feedback related to the given courses from the instructor.

16-Peer_Feedback:

Peer_Feedback Class contains feedback related to between group members in a particular group.

17-Question:

The Question Class allows setting up the questions to be asked while giving feedback. This Class is divided into 3 separate private classes.

18-Point_Question:

The Point_Question Class allows determining the point question among the questions to be created while giving feedback.

19-Multiple_Choice_Question:

Multiple_Choice_Question Class allows determining multiple-choice questions among the questions to be created while giving feedback.

20-Open_Ended_Question:

Open_Ended_Question Class allows determining open-ended questions among the questions to be created while giving feedback.

21-Artifact_Review:

The Artifact_Review Class allows various groups to give feedback and various scoring to other groups' projects at the end of the term.

3.3 Dynamic Models

3.3.1 Sequence Diagrams

3.3.1.1 Artifact Review

Scenario: User wants to give a rate to a Project.
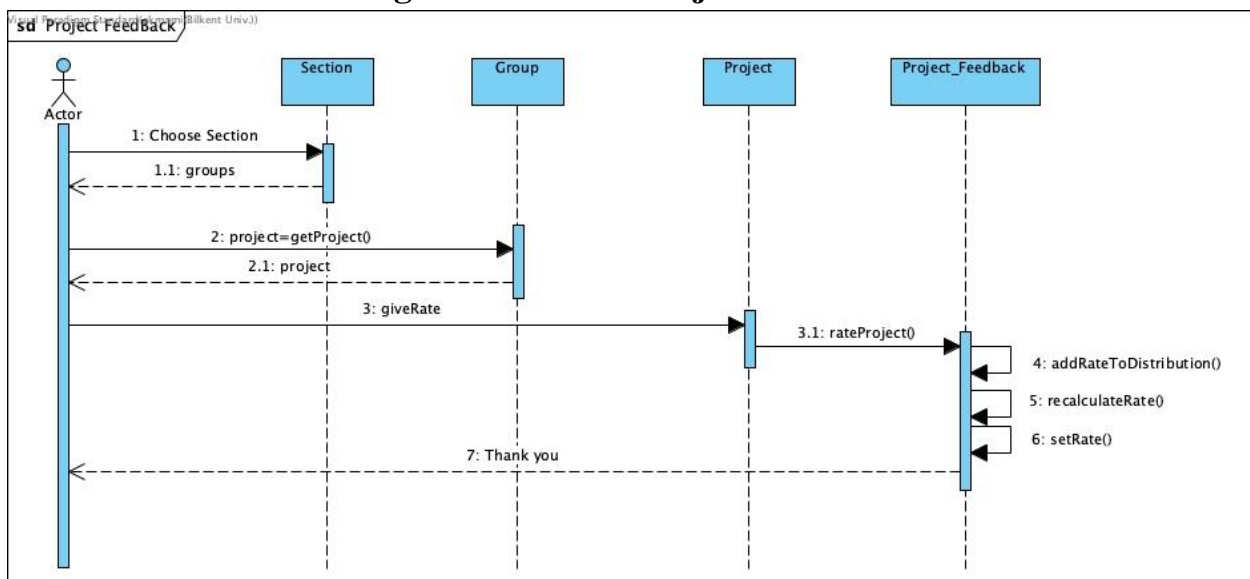
User logs in to the system and clicks a section link -on the dashboard screen- to see the groups of that section. Then the user clicks one of the group links to see the

project of that group - on the group profile page-. The user clicks the projects description link where the Group class will retrieve its project with the help of the Project class. Users will see the description of the project. If the user wants, they can give a rate via the stars symbols in the group profile page. Then, this rate will be implemented to the specific group's project feedback. Project_Feedback class will recalculate the project's rate again. After all the process, a display message "thank you" will be visible to the user.

3.3.1.2 Perform Peer Review

Scenario: Student wants to review his/her peer in the group.
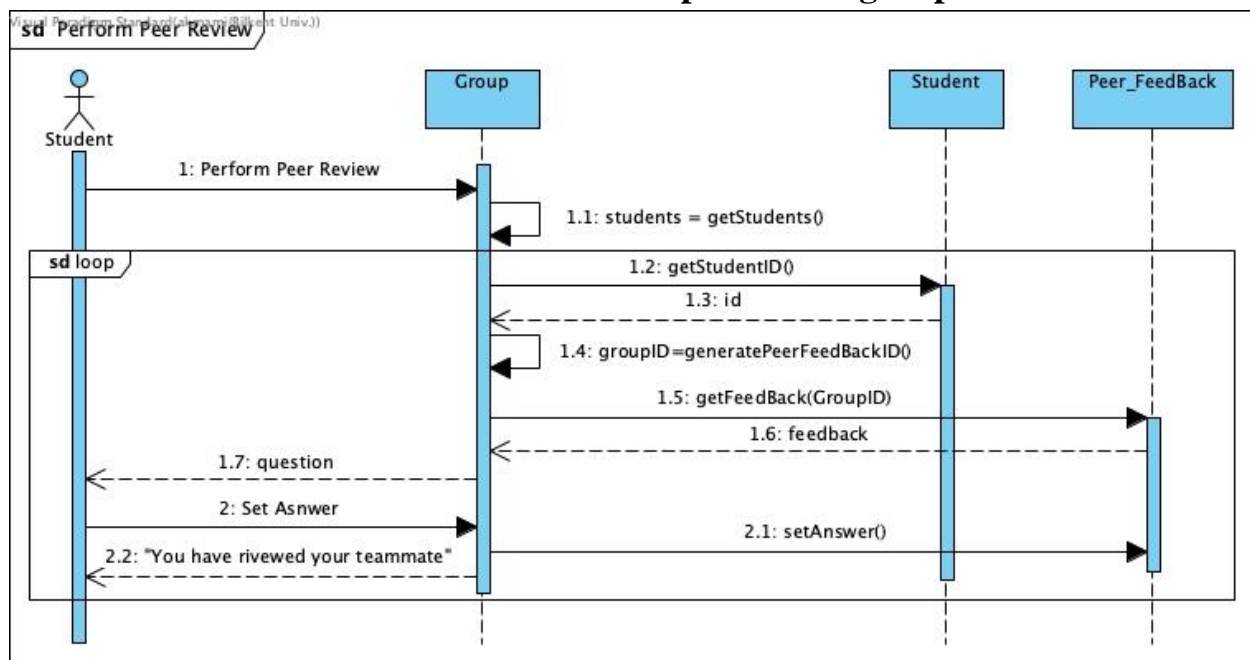
In the personal page of the student, the student reviews his/her peers in alignment. First, the Group class of the student retrieves the student list in that group. Then, Group class gets the first student's id in the group by the help of the student class. Then, according to this id, the feedback id will be generated feedback (Ali=21800 and Ayse=21900 then feedback id=2180021900 if Ali gives feedback to Ayse) and with that id, the feedback will be retrieved from the Group class where all peer feedbacks will be stored. Then, the peer feedback questions, retrieved from Peer_Feedback class, will be displayed to the user to set the answers. This answer will be saved into Peer_Feedback class with the feedback giver, feedback id and group id. Getting one student on the list and generating it's feedback and answer will be implemented in a loop until the reviewing of all the members of the group will be finished.
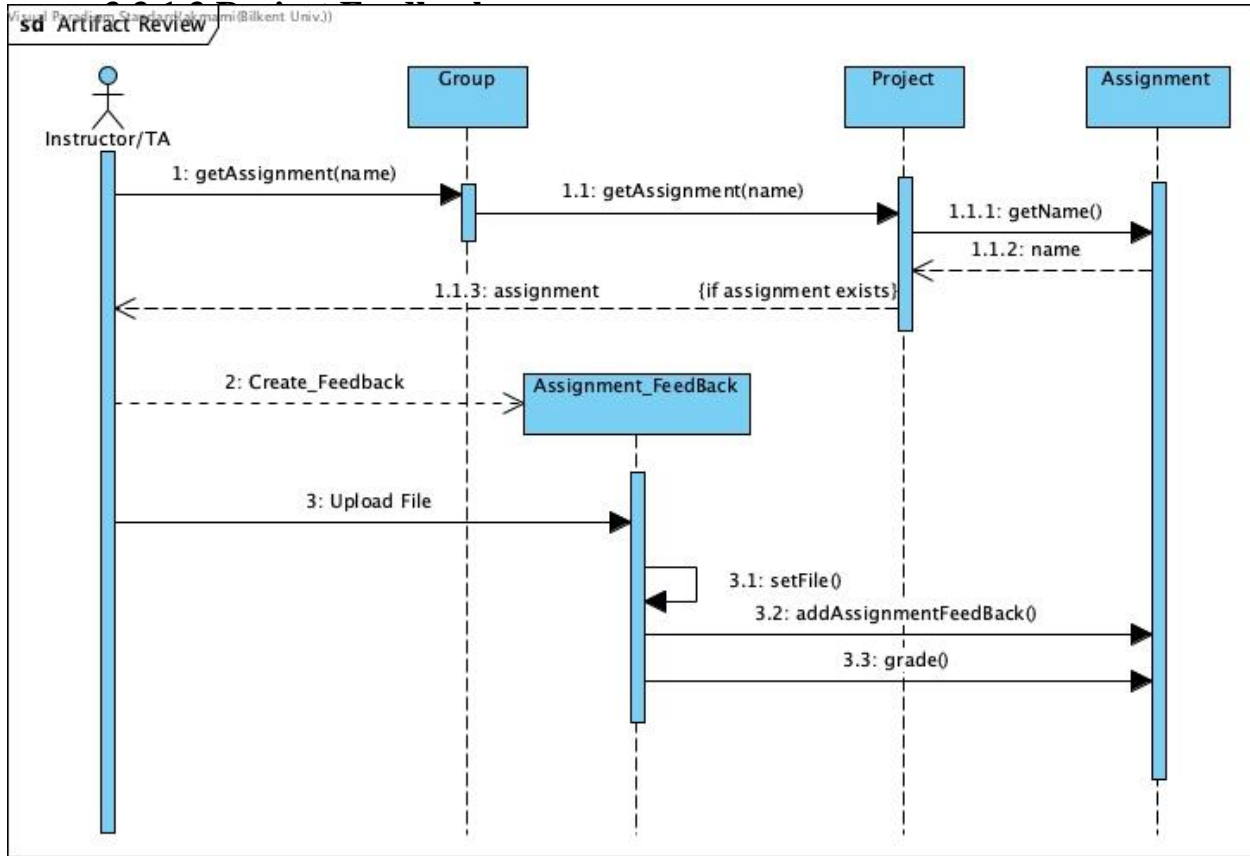
Scenario: The instructor or TA gives feedback to the assigned documents of the project.

Instructor/TA will get the assignment first by clicking the assignment which will find the assignment by its name. Then, the Instructor/TA creates a feedback file with the help of the Assignment_Feedback class. The Instructor/TA will upload the feedback file into the system. The Assignment_Feedback class will set this file and add the feedback and its grade -with respect to group id- into the assignment object of this Assignment class which will be stored in project class.

3.3.1.4 Group Formation

Scenario: The student forms a group.

Students click on the send request button on the group profile page. The request is created by the Group class and a method is sent to Leader class to get the request's answer. The request will be seen in the group profile page as well however; only group leaders can accept or deny the request. Then, if the leader accepts the request, the Group class will add the member student in the group via addMember method with the member

student's id. The member student will be also assigned to the particular group by the method assignGroup in the Student class.
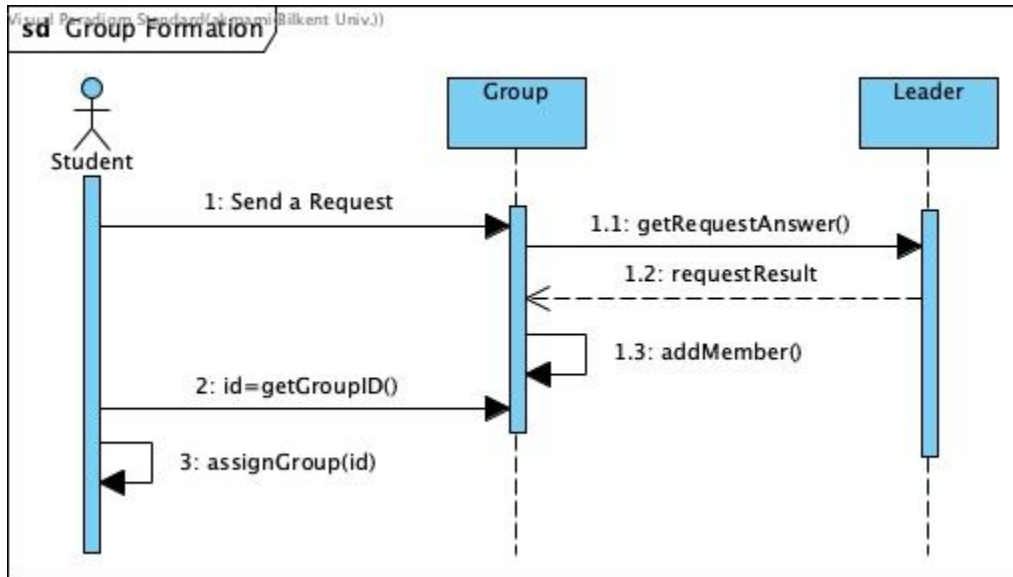
3.3.1.4 Assign Peer Review

Scenario: The instructor assigns peer review to students.

Instructor is in the "assign peer review" page. First, the instructor will add questions and its answers by the help of the "add" button on the page. Instructor class will call Question class by the method createQuestion to create a feedback question and question will be stored in string format in Question class. However, the question object will be stored in a list. Creating questions will be done in a loop as long as the instructor continues to add questions. Then, the instructor clicks on the Submit button on the page to send the feedback questions to students in the instructor's section. In Group class, createFeedback method will be called to create and set peer feedback for each two students in the group. Each peer feedback will have an id to differentiate all peer feedbacks. In PeerFeedback class, these ids will be created by the method createFeedbackID and the questions will be set by the setQuestions. In this way, the PeerFeedback class will also hold the questions of the feedback as well. After all the feedbacks are specified by an id -after loop process- this unique peerFeedback object will be added to Group class with the method addPeerFeedback.
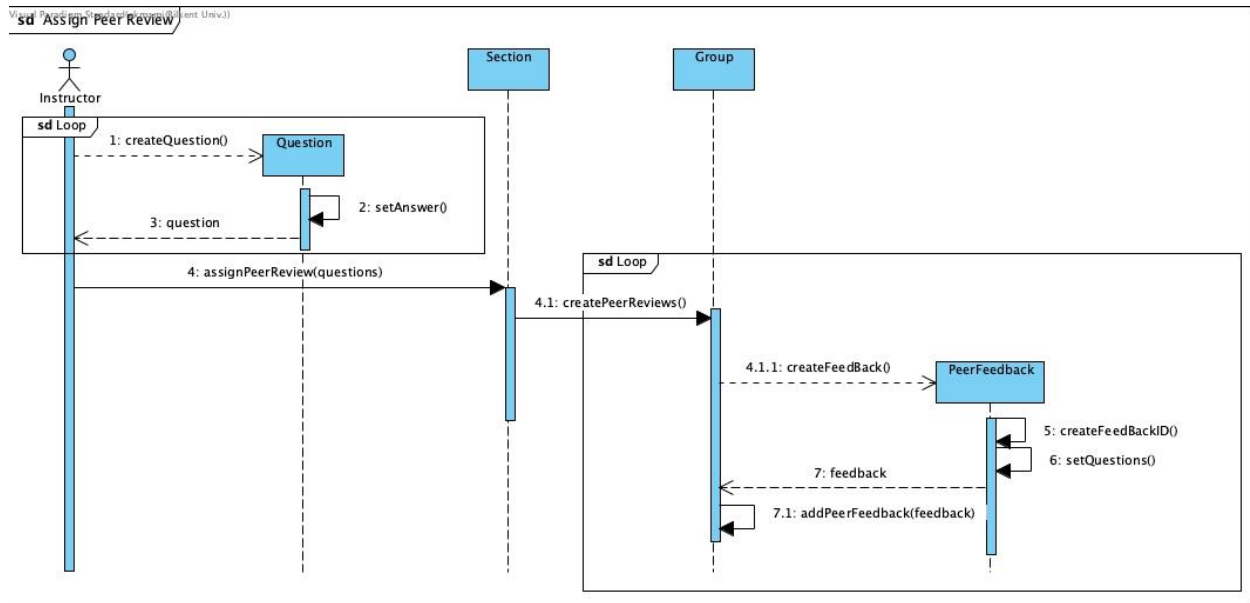
3.3.2 State Machine Diagrams

3.3.2.1 Assignment Class State Machine Diagram

First, the Assignment class will be created by the instructor or TA by the "create assignment button" on the group profile page. Then, by clicking the send button, assignments will be sent to groups and will be in the process of waiting for the responses. After the due date is passed, the assignment class will be in the state ResponseAssignment. In this state, students are now uploaded the assignments and waiting for feedback or a grade from the instructors. If instructor or TA gives feedback, the class goes to the FeedbackGivenAssignment state or if instructor or TA grades, the class goes to GradedAssignment state. Instructors or TAs can give grades or feedback multiple times. Then, at the end of the semester after the letter grades are given to students the assignment class will be closed and last grades or feedback could not be changed by instructor or TAs anymore.
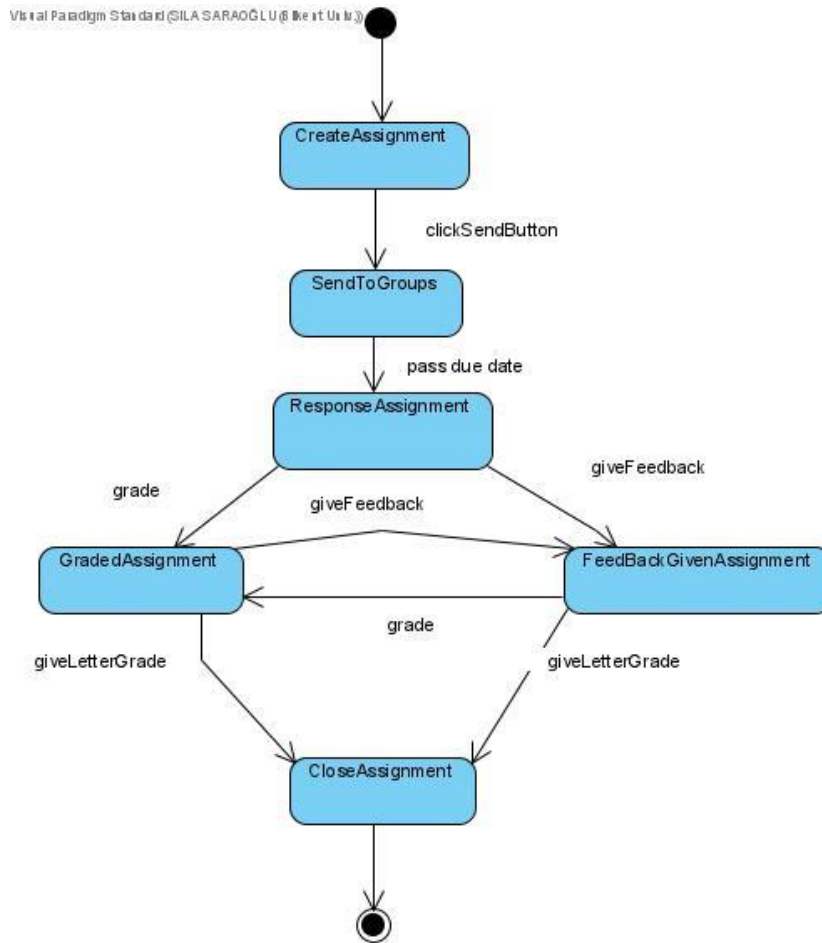
3.3.2.1 Group Class State Machine Diagram

First state of the Group class is the StartFormation which is initiated by the instructor by clicking on the "start group formation" button. Then, if the instructor decides to randomly assign students and clicks on the random button, Group class will be in the state of AssignRandom. Otherwise, if the instructor leaves the group formation to students, the Group class will be in the state of GroupFormation. Until the group is full, members can be added to the group via inviting other students or accepting requests that come from other students. After the due date of group formation, the Group class will be in the state of FormationClosed state. In this state, the formation is finished, students will have to continue the semester with their current groups. At the end of the semester, the groups will be closed.

clickLeaveToStudentsButton

GroupFormation

StartFormation

inviteMember

acceptRequest

clickRandomButton

GroupIsFull

AssignRandom

passDueDate

PassDueDate

FormationClosed

end of semester

CloseGroup
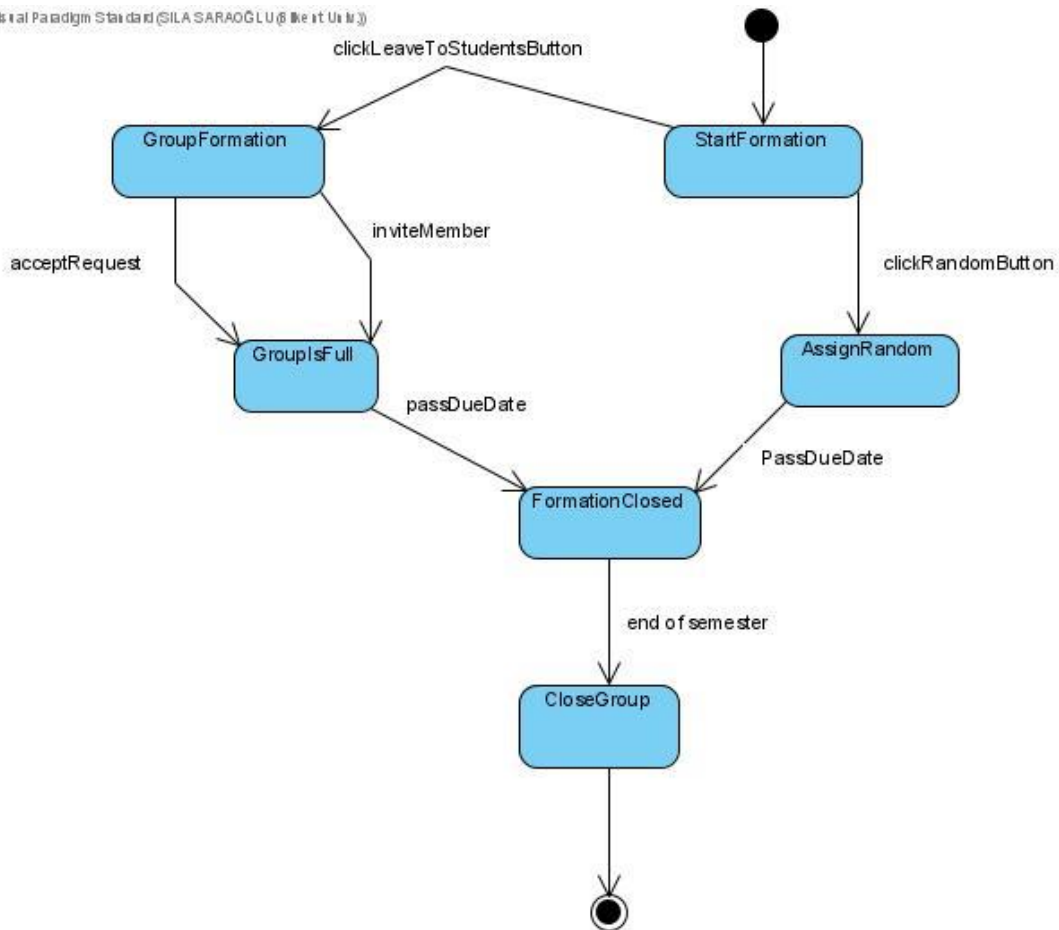
3.3.3 Activity Diagram

The Activity Diagram starts with the initial state which directs the user to the welcome page. In the welcome page, there is a fork that can lead the user to several options depending on the choice. There are several information about the application that are About, Terms, Privacy, Our policy and Contact us. Besides the user can go directly into FAQ and Login Page too. If the user goes to the Login Page there are two options: The user can sign up to the environment or if the user is registered to the environment previously, then can directly Log in to the environment. Also if the user is registered however forgot their password, then can regain their password and login again.

After logged in to the environment, the user is directed to the Dashboard which is the main page of the application. In the dashboard there are several actions that can be done by the user indicated with a fork. The user can check the notifications in the notification page, can go directly to the chat page, personal profile page, group profile page, and can create a new group. If the user is an instructor, can also go to the Peer review creation page and can start the group formation period.

Students in the profile page can fill the Peer Review of other students if the peer review is submitted. Students can also create a new group if the instructor started the group formation. The instructor can create a peer review by adding 3 different question types optionally and submit it. After students fill the Peer Review, Instructors can access to student peer reviews in their Personal Profile Page.

Students also can go to the group profile page from dashboard and there is another fork after the group profile page. In the group profile page students can invite and remove members, send requests to other users, leave the group or submit artifacts. Moreover, students can access peer review notifications and chat pages through that page too.

3.4 User Interface

3.4.1 Navigational Path

Visual Paradigm Standard(SILA SARAOĞLU(İlkent Univ.))
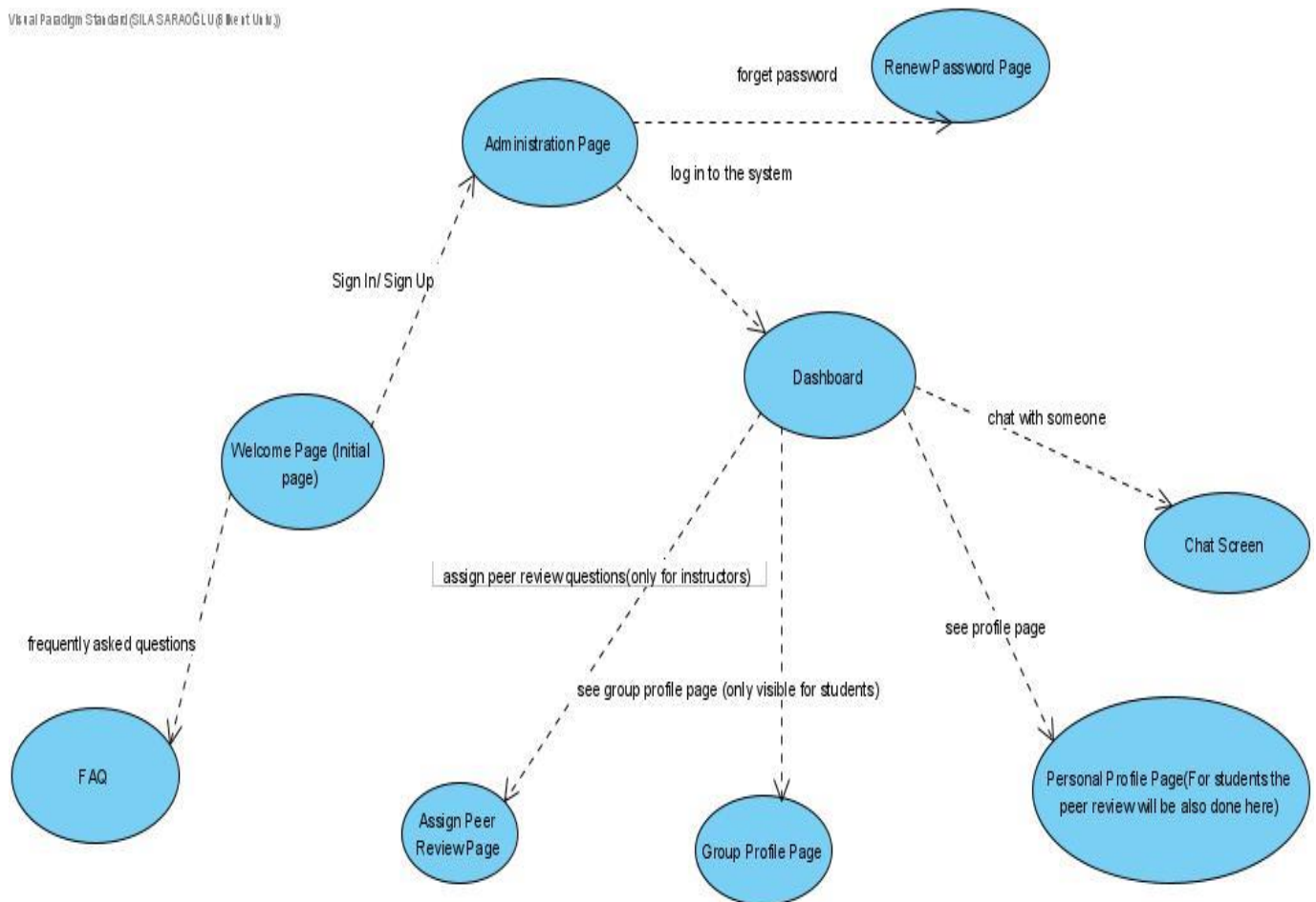
3.4.2 Screen Mock-ups

3.4.2.1 Welcome Page
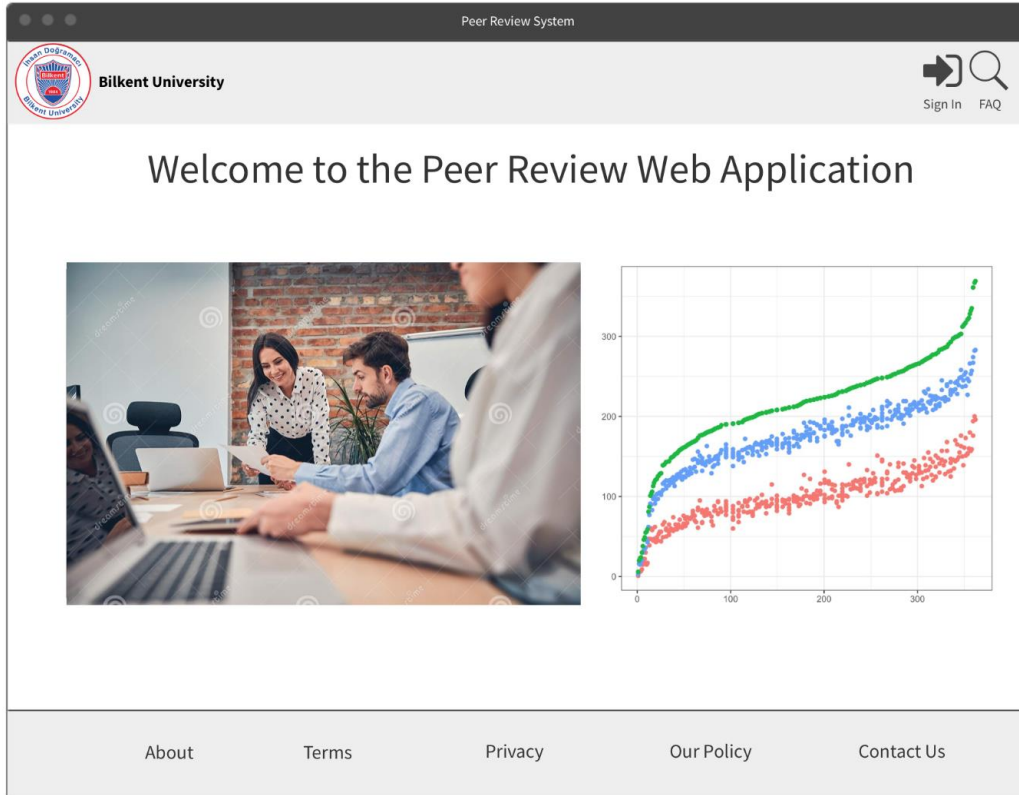
Welcome page is the start page of the Peer Review System Web Application, which consists of 7 clickable buttons and links; "About", "Terms", "Privacy", "Our Policy", "Contact Us", and "Sign In" which navigates the user to signup/login page and "FAQ button" which navigates to the frequently asked questions page which includes both a tutorial and answers to common questions about the application. At the center of the page, there will be the statistics of the app (total users, growth, etc.)

3.4.2.2 Signup and Login Page

Signup and login page is the page where users can sign up to the system via filling the information on the left or users can log in to the system via filling the information on the right if the user has got an existing account. The buttons on the navigation bar bottom of the page functions the same as the home page. A "forget password" link is visible near the login button. This link will navigate the user to a different page where he/she can change his/her password via email. When the user presses the sign up button if the information he/she is given valid then the user is signed up. If the user has an existing account and gives the right information and presses the login button then the user is navigated to the dashboard screen.

3.4.2.3 Dashboard

In Dashboard Screen, users can search an artifact, a group or a personal profile (Student/TA/Instructor) at the search bar on the top of the screen near the notifications button. The notifications button provides an exclamation to notify users about the new events. Via the "Profile" button, users can see their own profile while the "Group" button will be visible for students that will navigate to their group profile. "Create New Group" button will be visible in the students' dashboard as well, which will help the student to form a group when the instructor will assign group formation. The "Peer Review" and "Start Group Formation" buttons will be visible in instructors' dashboard to assign peer review questions to students and start the group formation process. Also, at the bottom, the all groups at that section will be visible and only the instructor/TA can assign or remove group members by clicking the links,, so few students left without groups can be manually placed to the groups. At the right of the page, a chat button is visible to chat with other users. At the middle of the page, there will be a main timeline where everyone can post anything while it will be only visible to the

students at that section.
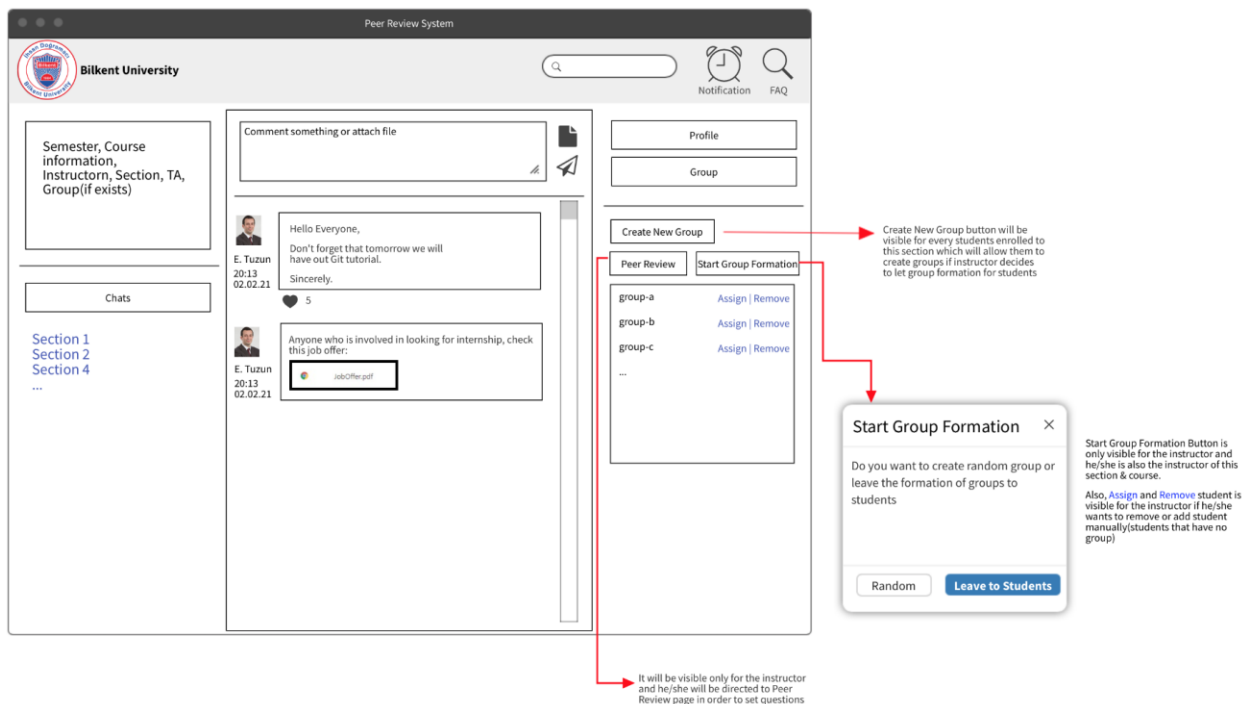
3.4.2.4 Chat Screen
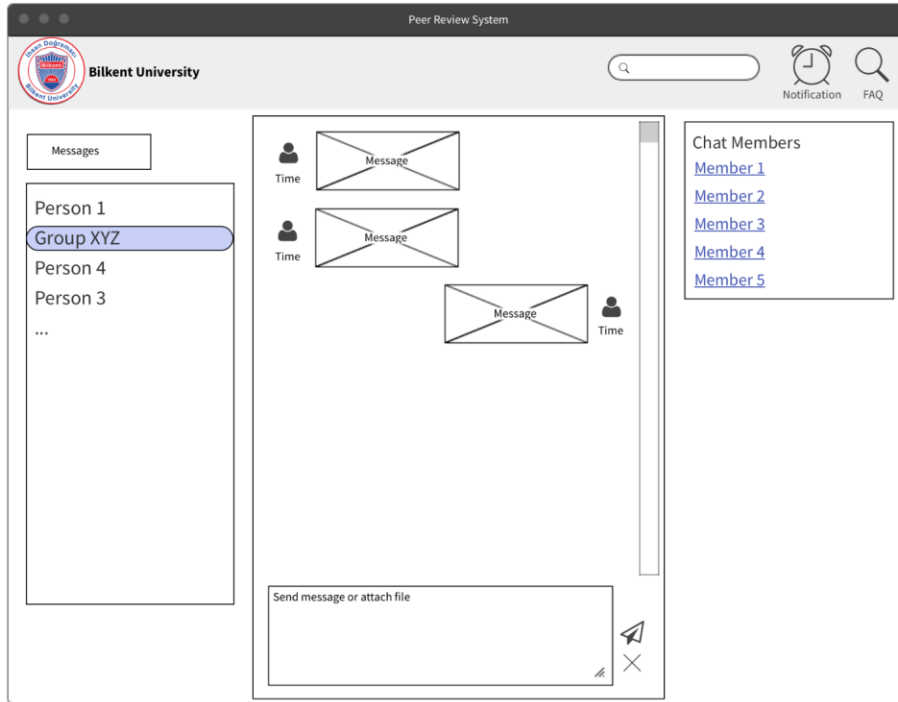
      The chat screen allows users to chat with different users personally or as a group. The top navigation bar is the same as the dashboard screen. All chats will be created manually when students will be assigned to the section and group chats will be created/deleted when students form/join a group. Students will not be allowed to create chat manually. Also, at the right side of the page, there will be the members of the group chat or person who is chatting right at that moment.

3.4.2.5 Student Profile Page

The student profile page provides sensitive information about the student and a peer review feature at the bottom. The profile picture and information will be visible for all people (enrolled to course) while the Peer Review part will be visible only for the Instructor, so the instructor will be able to see all reviews done to him/her by his/her group members.

Beside the information and profile picture, there will be an Instructor's section which will be visible only for the instructor of the section (other instructors will not see this). Instructor will be able to see his/her sections and inside them, see all groups or students (depending how the instructor will choose to list). So, instructors can go to the group page or student profile page by clicking them.

3.4.2.7 Group Profile Page

The group

profile page will be visible if the student presses the "Group" button in the dashboard screen. If the user has a group, at the middle the group name, its members and the relevant documents will be visible. The groups here can upload documents relevant to their group project to present it to TAs and instructors - via the "Add New Document" link on the right side of the screen. Moreover, instructors can upload feedback as well to these documents via the "Upload File" link at the middle of the page. Feedback will be visible only for the members and Instructor and TA's. Also, Each member can perform peer review by clicking "Peer Review" button (it will navigate to Peer Review Page).

3.4.2.8 Assign Peer Review Page

The assigned peer review page will be only visible to the instructor of the course. The instructor can edit the peer review questions and add them to the questions' list. There are 3 different types of questions to edit to the system. When the instructor clicks on the submit button, the peer review document and its questions are saved to assign students.

3.4.2.9 Pop-up Messages

4 Conclusion

Online resources are the indispensable tools for improving course efficiency. As the web industry progressed, developers built various kinds of tools for education purposes. One of the most crucial tools that instructors might need during the course period is reviewing applications. Especially for the group project assignments, those tools are very essential. Group projects require those applications for several reasons. First of all, during the term group projects there might be some inequities between the peers of the group. For instance, one of the participants will be obligated to do all of the work oneself. Peer review systems are barriers that prevent those kinds of annoying situations from happening or at least to have consequences. On the other hand, reviews are essential for the students to notice their faults and improve themselves. Besides that, taking feedback is not just beneficial for the students, instructors and TA's can also be evaluated and improve themselves for the next semesters. Peer Review System web application contains all of those review systems in its content for ensuring all actors in the class a good and beneficial experience.

Furthermore, this application also contains other helper systems in it content to ease the course period for all of the actors. Such as, it contains group formation events, that peers can form their own group under the surveillance of the instructor. Forming groups is always a painful process for the students, this application offers an easy way for students to do that task without needing any other application. The system has a message and chat functions to ensure the communication between the users. As discussed above the system has several functions and mechanisms for ensuring the best for the period.

Peer review application is recommended for the all-course instructors, if they want to improve their efficiency and reduce their unnecessary workload. It is a helper tool which is both an improver and facilitator for the process.

5 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.