Bilkent University

Department of Computer Engineering

CS319
Object-Oriented Software Engineering

Classroom Helper

Group 1G

Analysis Report

Tuna Dalbeler (21802539), Ezgi Saygılı (21802871), Funda Tan (21801861),
Selahattin Cem Öztürk (21802856), Onur Korkmaz(21802925)

Instructor: Eray Tüzün

Teaching Assistants: Elgun Jabrayilzade, Erdem Tuna

1.Introduction

Our project is intended to simplify the group formation and peer review processes in classes, which is called a Classroom Helper. With our application, finding groups and reviewing and grading their groups or the other groups in the classroom will be easier. Moreover, the Classroom Helper tool will facilitate the communication between students, TAs and teachers (TAs and teachers will be considered as instructors) with the student forums and synchronous chat. Classroom Helper tool facilitates in-group activities with an in-group work distribution tool and a deadline calendar. Also, our tool provides some features for instructors such as analytic tools. We will use Java to implement our project.

2.Proposed System

2.1 Overview

Group Formation:

Group formation consists of two options. First option is, a student can send a join request to a group. The student can also use the “bundle me with a friend” option. With the “bundle me with a friend” option, students can send a join request to a group together. Second option is inviting students to an existing group. Students can invite other students to their group.

If the group is not full, students may send join requests or invitation requests, otherwise, they cannot. Instructor may start or end the group formation process.

Instructors can choose to make random groups.

Peer Review and Peer Grading :

The peer-review feature enables students to review other groups’ artifacts. The peer-grade feature enables students to grade their teammates and the other groups. Instructors can see the reviews and grades of the students and the groups.

Synchronous Chat:

The synchronous chat function enables synchronous communication between instructors and students.

Messaging:

Messaging includes private messaging between members of the classroom and group messaging between the people in the group and general messaging between people in the course.

Student Forums:

Student forums allow students and isntructors to ask questions and post answers.

Tasks:

Students may create tasks or delete tasks using tasks functionality. With tasks, students can distribute the work among the group members.

Upload Assignment:

Students can upload assignments using the upload assignment tool. By doing this, other groups can see their assignments to review and grade.

Calendar:

The Calendar shows due dates and dates that may be important. Users can add events to the calendar or delete events from the calendar.

Group Meetings Log:

Students can record their meetings using the group meetings log.

Information:

Displays information about the people in a classroom.

2.2 Functional Requirements

From both instructors' and students' point of view:

Login to course: Entering the website link user is navigated to the login screen asking their email and password. User either logins by entering his/her email and password or switches to sign up page.
Sign up: Users will sign up with a sign up page. They will provide email and password.
Sign up to a course: Users can sign up to a course by using course code (created in the course creation process).
Options: Users are able to modify their profiles: contact information, name etc.
Course page: Users have a courses page displaying the courses they signed up.
All groups page: Users have an all groups page displaying all groups in a course.
Display course & participant information: Users can display the course information and contact information of students and instructors.
Private/group/general chat: Users can chat synchronously in three ways: person to person(private), as a group or as a course participant(general). Private chat is either between two students, two instructors or one student and one instructor. Group chat is between any students in a group. General chat is between all students and instructors in the course.
Forums: Allows all users, students and instructors, to share questions and answers among each other.
Calendar: Displays the deadlines for the assignments given by instructors.
Review Groups: All users are able to review each group in the "all group"'s page. Displays the participant list and information about the group.
View statistics for grades: All users are able to view the statistics computed by comparing the grades between groups(artifact-review) and instructors in Gradebook. Also displays assignment grades for groups, this is visible for participants in a particular group and instructors in the course.

Only from instructors' point of view:

Create a course: The course will be created by an instructor and a sign up code will be provided. With this code students and instructors (if there are other instructors such as other teachers and assistants) will sign up to courses.

Assignments page: Allows instructors to create assignments through assignments page. This assignment will be displayed on the calendar as an event. Instructors can upload a file and specify criterias.
View student dashboard: All instructors are able to see the detailed reviews about all students in the course.
View statistics for grades: In addition to viewing the statistics computed by comparing the grades between groups and instructors, also displays the detailed statistics regarding in-group peer review and assignment grades in Gradebook.
Analytic tools for instructors: Allows to import-export grades. Import review questions to be used in in-group peer review and artifact review, in addition to the template questions provided by the tool.
Assign unassigned students to groups: Allows instructors to assign remaining students to groups that have available seats.

Only from students' point of view:

My Group Page: Students have my group page displaying group information, join requests, group chat, assignment upload link and calendar displaying deadlines for tasks created by group members and assignments given by instructors. Group members can create tasks to divide the workload among themselves and review the assignment file uploaded by the instructor.
Calendar: In addition to displaying the deadlines for the assignments given by instructors, also displays the deadlines for the group tasks created through in-group work distribution tools.
Form a group: A student can create a group himself/herself and manage invitations, join and bundle me with friend requests. A student can send an invitation to others to join his/her group through an unassigned students list. He/she can accept or decline a join/bundle me with a friend request.
Find a group: All groups created are visible in all groups' pages. Students can review groups, if there are available seats and the student is not already in a group, he/she can send a join/bundle me with friend request to a group.
In-group work distribution tool: Group members are able to create their own tasks and specify deadlines on a calendar. They can specify the task and the responsible member.
Group meetings notepad: Students can keep a record of the key points in a group meeting.

In group peer review: Enables group members to review and grade each other anonymously. This is only visible to person reviewing, not visible to group members or other groups.
Peer review of artifacts: Enables reviewing and grading other groups' work anonymously. This is only visible to group reviewing and grading, not visible to other groups.
Review and Upload assignment: Students can see the assignment file and upload the assignments given by the instructor through their group page.

2.3 Nonfunctional Requirements

Extendibility: Classroom Helper tool will have an extendible design to enable adding new features or extending the current features without having to modify the legacy code substantially. This efficiency will be achieved by employing Object-Oriented Programming principles.

Maintainability: Classroom Helper tool will have a maintainable design to repair or replace faulty or worn-out components without having to replace still working parts, maximize a product's useful life, make future maintenance easier and cope with a changing environment. It will be achieved by using plenty of comments and forming a clear and easy-to-understand structure for design and implementation.

Usability: Classroom Helper tool will have a simple and user-friendly interface supporting a range of user actions which will enable users to quickly grasp the use of the tool. User interface components and verbal interactions will be similar to that of contemporary communication tools. Names, messages, buttons, icons and graphics will require a basic and intuitive understanding so that they will be tailored to meet the naive users' requirements.

Reliability: The program should have a secure connection and should use HTTPS protocol. Unfortunately, since none of the group members are experienced in web security, data security over the web cannot be a first priority for us. [Security Requirement]
The program should have a minimum 95% uptime (1 hour in a day might be down). [Mean time between failures requirement]
The system should backup once in 3 days to protect against power loss. [Data loss tolerance]

Performance Requirements: The System should support at least 50 concurrent users and ,to allow a smooth experience, all user inputs must be acknowledged within 5 second and the Classroom Helper tool must respond to a user action within 5 second.

2.4 Pseudo Requirements

Implementation Requirements: Classroom Helper will use React for frontend, Java for backend and mySQL for database.

Client Software Requirements: Since the program will use React in frontend, the client's browser should support it. Chrome, Firefox, Safari should be the latest. Other platforms might work but will not be officially supported. Older versions of any kind of browser will not be supported because of time constraints.

Legal Requirements: The program must conform to KVKK (Kişisel verileri koruma kanunu - Data Protection Regulations).

3. System models

3.1 Use case model

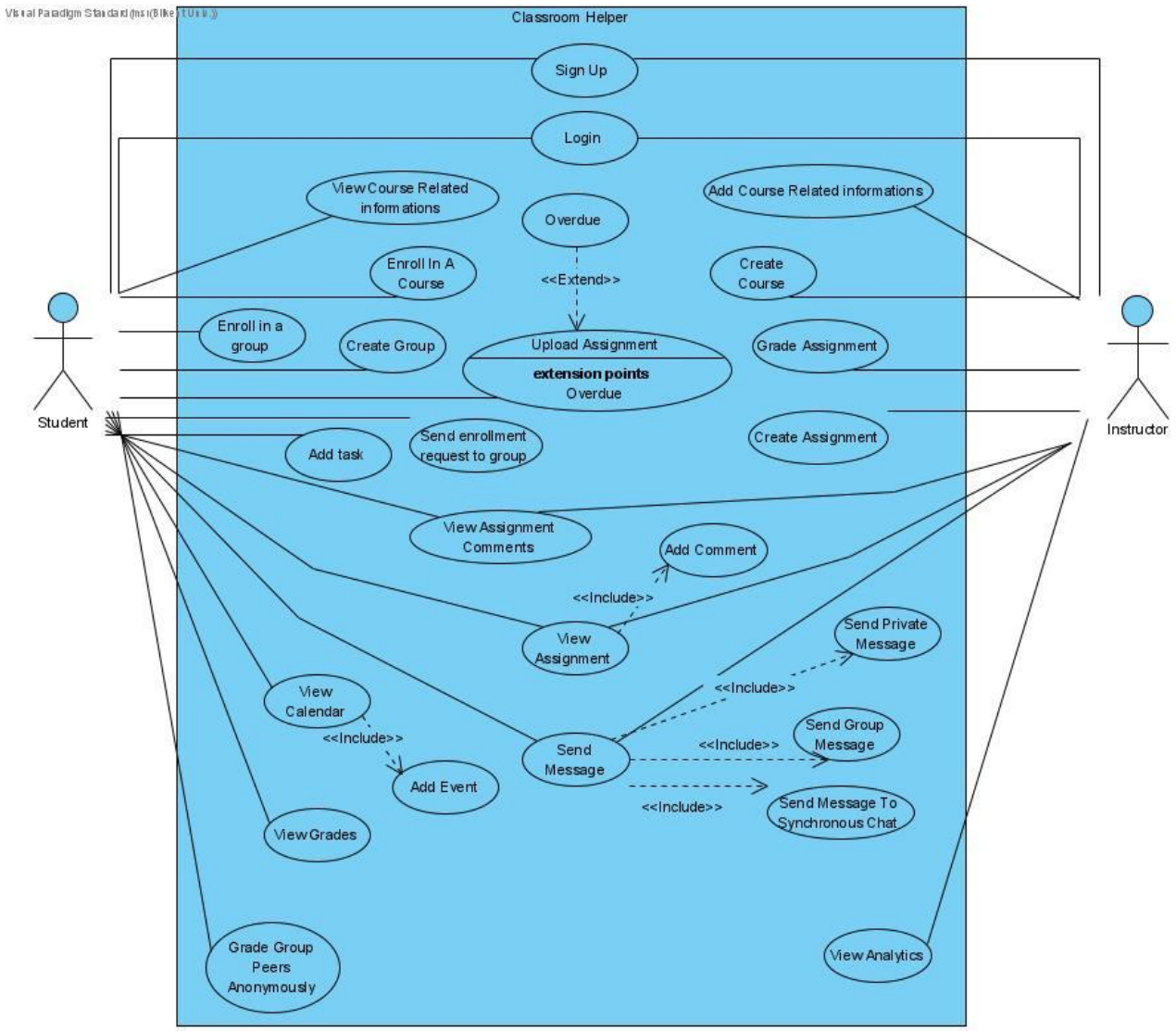

Figure 1 - Use Case diagram

Use Case #1

Use case name: Create Group.

Participating actors: Student.

Stakeholders and interests:

- Student navigates to Create Group page.
- The system displays the Create Group screen.

Pre-conditions:

- The student must be logged in to the system.
- The student must not be in another group.

Post-conditions:

- No post conditions.

Entry-conditions:

- The student navigates to Create Group page.

Exit-conditions:

- Student cancels the group creation.
- Student creates the group.

Success Scenario Event Flow:

1. Student navigates to All Groups page.
2. System displays the All Groups page.
3. Student navigates to Create Group page.
4. System displays the Create Group page.
5. Student fills out the necessary informations.
6. Student creates the group .
7. System adds the new group to the All Groups page.

Alternative Event Flows:

1) If student wants to cancel the create group process
    a) Student navigates to All Groups page.
    b) System displays the All Groups page.
    c) Student navigates to Create Group page.
    d) System displays the Create Group page.
    e) Student cancels the group creation process.
    f) System displays the All groups page.

Use Case #2

Use case name: Send Enrollment Request to a Group.

Participating actors: Student.

Stakeholders and interests:

- Student send Enrollment Request to a Group.
- System adds the request to Group Page.

Pre-conditions:

- The student must be logged into the system.
- Student must not be in another group.

Post-conditions:

- Student added to the group as a member.

Entry-conditions:

- Student sends an Enrollment Request to a Group.

Exit-conditions:

- No exit conditions.

Success Scenario Event Flow:

1. Student navigates to all group page.
2. System displays the all groups page.
3. Student sends an Enrollment Request to a Group.

4. System adds the request to the Group Page.

Alternative Event Flows:

1. No alternative event flow.

Use Case #3

Use case name: Upload Assignment.

Participating actors: Student.

Stakeholders and interests:

- Student Uploads an Assignment.
- System shows the uploaded file on the corresponding group page.

Pre-conditions:

- The student must be enrolled in a group.
- The assignment must be uploaded before the due date.

Post-conditions:

- No postconditions.

Entry-conditions:

- Student navigates to Upload Assignment Page.

Exit-conditions:

- Student cancel the Upload Assignment process.
- Student uploads a file to the system.

Success Scenario Event Flow:

1. Student selects the All Groups button.
2. System displays the All Groups page.
3. Student clicks on him/her Group Page.
4. System displays the Group Page.

5. Student navigates to Uploads Assignment Page.
6. System displays the Upload Assignment Page.
7. Student selects the file.
8. Student uploads the file.
9. System displays the file on the group page.

Alternative Event Flows:

1) If student wants to upload an assignment by directly going to their Group Page.
    a) Student navigates to My Group Page.
    b) System displays the Group Page.
    c) Student navigates to Uploads Assignment Page.
    d) System displays the Upload Assignment Page.
    e) Student selects the file.
    f) Student uploads the file to the system.
    g) System displays the file on the Group Page.
2) If the student wants to cancel the upload process.
    i) Student navigates to My Group Page.
    ii) System displays the Group Page.
    iii) Student navigates to Uploads Assignment Page.
    iv) System displays the Upload Assignment Page.
    v) Student cancels the process.
    vi) System displays the Group Page.

Use Case #4

Use case name: View Assignment.

Participating actors: Student and Instructor.

Stakeholders and interests:

- Student or Instructor navigates to the uploaded file to view it.
- System displays the uploaded assignment.

Pre-conditions:

- Student or Instructor must be on the corresponding Group Page.

- There must be an uploaded file for an assignment.

Post-conditions:

- No postconditions.

Entry-conditions:

- Student or Instructor navigates to uploaded file to view it.

Exit-conditions:

- Student or Instructor exits from the uploaded file page.

Quality requirements:

> At any point during the flow of events, this use case includes the Add Comment use case. This Add comment use case is initiated when the Student or Instructor invokes the add comment function that will be also displayed on the review assignment page. When this use case is invoked, the system will add the comment on the file.

Success Scenario Event Flow:

1. Student or Instructor selects the all groups button.
2. System displays the all groups page.
3. Student or Instructor navigates to the Uploaded Assignment Page.
4. System displays the Uploaded Assignment Page.
5. Student or Instructor exits the Uploaded Assignment Page.
6. System returns to the Group Page.

Alternative Event Flows:

1. No alternative event flow.

Use Case #5

Use case name: Grade group peers anonymously.

Participating actors: Students.

Stakeholders and interests:

- Student grades the peer.
- The system displays the peer grade screen.

Pre-conditions:

- The student must be in a group.

Post-conditions:

- No postconditions.

Entry-conditions:

- Student clicks the grade your peer button.

Exit-conditions:

- Student cancels the process.
- Student submits the grade.

Success Scenario Event Flow:

1. Student navigates to All Group Page
2. System displays the All Group Page.
3. Student navigates to his/hers group page.
4. System displays the group page.
5. Student navigates to the Peer Grade Page.
6. System displays the Peer Grade Page.
7. Student gives the grades of his/her peers.
8. Student submits the grade.
9. System displays My Group Page.

Alternative Event Flows:

1. If a student wants to cancel the peer grading process:
    a. Student navigates to All Group Page
    b. Student clicks on his/her Group Page.
    c. System displays the Group Page.
    d. Student navigates to Peer Grade Page.
    e. System displays the Peer Grade Page.
    f. Student cancels the process.

g. System displays my group page.

Use Case #6

Use case name: Grade Assignment.

Participating actors: Instructor.

Stakeholders and interests:

- Instructor press the grade assignment icon.
- The system displays the grading screen.

Pre-conditions:

- Instructor must be on a group page.
- There must be an uploaded assignment.

Post-conditions:

- No postconditions.

Entry-conditions:

- Instructor press the grade assignment icon.

Exit-conditions:

- Instructor press the “cancel” button.
- Instructor press the “upload the grade” button.

Success Scenario Event Flow:

1. Instructor press the all groups button.
2. System displays the all groups page.
3. Instructor went to a group page by pressing on it.
4. System displays the group page.
5. Instructor press the grade assignment button.
6. System displays the grading page.
7. Instructor gives enters the grade.
8. Instructor press the upload the grade button.
9. System displays the group page.

Alternative Event Flows:

1) If a Instructor wants to cancel the create a group process:
    a) Instructor press the all groups button.
    b) System displays the all groups page.
    c) Instructor went to a group page by pressing on it.
    d) System displays the group page.
    e) Instructor press the grade assignment button.
    f) System displays the grading page.
    g) Instructor presses the cancel button.
    h) System displays the group page.

Use Case #7

Use case name: Send Private Message.

Participating actors: Student and Instructor.

Stakeholders and interests:

- Student or Instructor wrote a message and press the send button.
- The system displays the message in the chat box.

Pre-conditions:

- Student or Instructor must be enrolled in a course.

Post-conditions:

- No postconditions.

Entry-conditions:

- Student or Instructor press chat icon that is next to the users' name.

Exit-conditions:

- Student or Instructor press the send button.

Success Scenario Event Flow:

1. Student or Instructor press the people button.
2. System displays the people page.
3. Student or Instructor press the chat icon that is next to another user's name.
4. System displays private chatbox.
5. Student or Instructor wrote a message and press the send button.
6. System shows the message in the private chat box.

Alternative Event Flows:

1. If Student or Instructor wants wrote a message in the group page
    a. Student or Instructor press all groups button.
    b. System displays the all group page.
    c. Student or Instructor presses the group.
    d. System displays the group page.
    e. Student or Instructor press the chat icon that is next to the another user's name.
    f. System displays private chatbox.
    g. Student or Instructor wrote a message and press the send button.
    h. System show the message in the private chatbox.

Use Case #8

Use case name: Send Group Message.

Participating actors: Student.

Stakeholders and interests:

- Student wrote a message and press the send button.
- The system displays the message in the group chat.

Pre-conditions:

- Student must be enrolled in a group.

Post-conditions:

- No postconditions.

Entry-conditions:

- Student, goes into the my group page.

Exit-conditions:

- Student press the send button.

Success Scenario Event Flow:

1. Student, press the my group button.
2. System displays the group page.
3. Student wrote a message to the group chat and press the send button.
4. System shows the message in the group chat.

Alternative Event Flows:

1. If Student or Instructor wants to write a message on the group page.
    a. Student press the all groups button.
    b. System displays the all group page.
    c. Student presses his/her group.
    d. System displays the group page.
    e. Student wrote a message to the group chat and press the send button.
    f. System shows the message in the group chat.

3.2 Sequence Diagrams

Sequence Diagram structure is affected by a web page from a university (Pearce, *Client-Server Architectures*).

Scenario 1: Login

The user enters their credentials to the client. The Client creates a request and sends the email and password to the backend. RequestListener processes the request and creates a request handler. This RequestHandler creation process' purpose is to handle more than one request in parallel. Handler sends credentials to UserManager and UserManager checks if the user exists in the database. If user email - password does exist in the database sends the user a declined message. If it exists create a UserSession. UserSession holds the data about what the user did and auth token etc. Then PageManager prepares the homepage with users data. Then the homepage returns to the client.

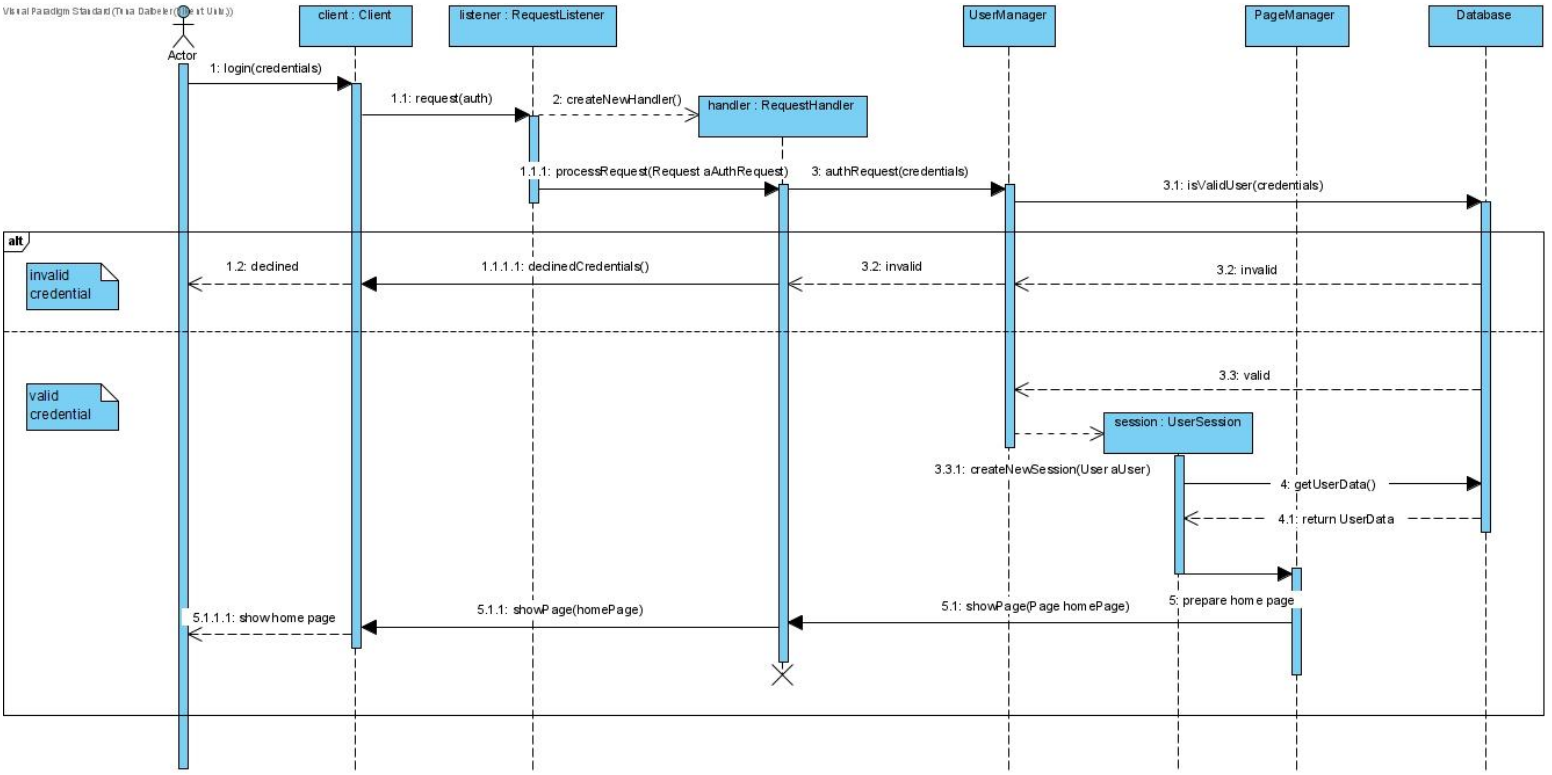

Figure 2 - Sequence Diagram - Login Sequence

Scenario 2: Join a Group

The user clicks the “All Groups” button. A page request sent to the server. UserSession checks if the user is eligible to access the page. If not, return an error. Error process is omitted from the diagram for the sake of simplicity. If the request is okay send the page to the user. On the All Groups page, the user clicks join and the join request is sent to the server. The handler passes the request to UserSession. UserSession makes necessary checks (like if the user is already in a group or not). Then UserSession passes the request to AllGroupsManager. AllGroupsManager is the class that holds all groups and records groups history, also handles adding and deleting a group, also checks if the group choosing period is over or not (Group choosing period is when people can freely form groups. Later will be disabled by the instructor or an admin account.). If the request is okay, records the request into the database and returns a successful message.

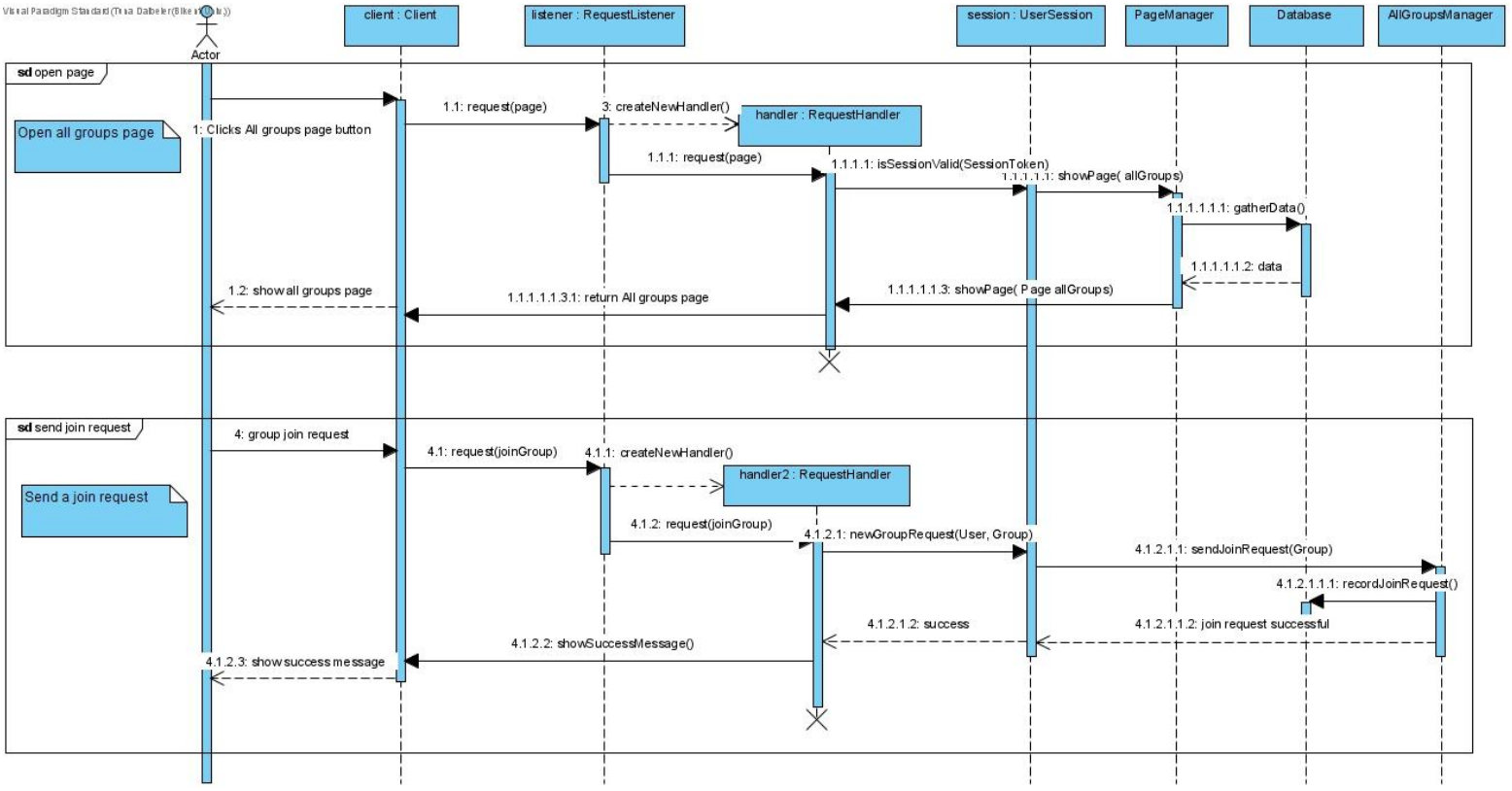

Figure 3 - Sequence Diagram - Request for all groups page and join a group sequence

Scenario 3: Upload Assignment

After the user got into the upload page. User clicks upload file and chooses a file and sends an upload request. After that file, passed through Handler and UserSession, reaches the user's group. The Group records the file into the system.

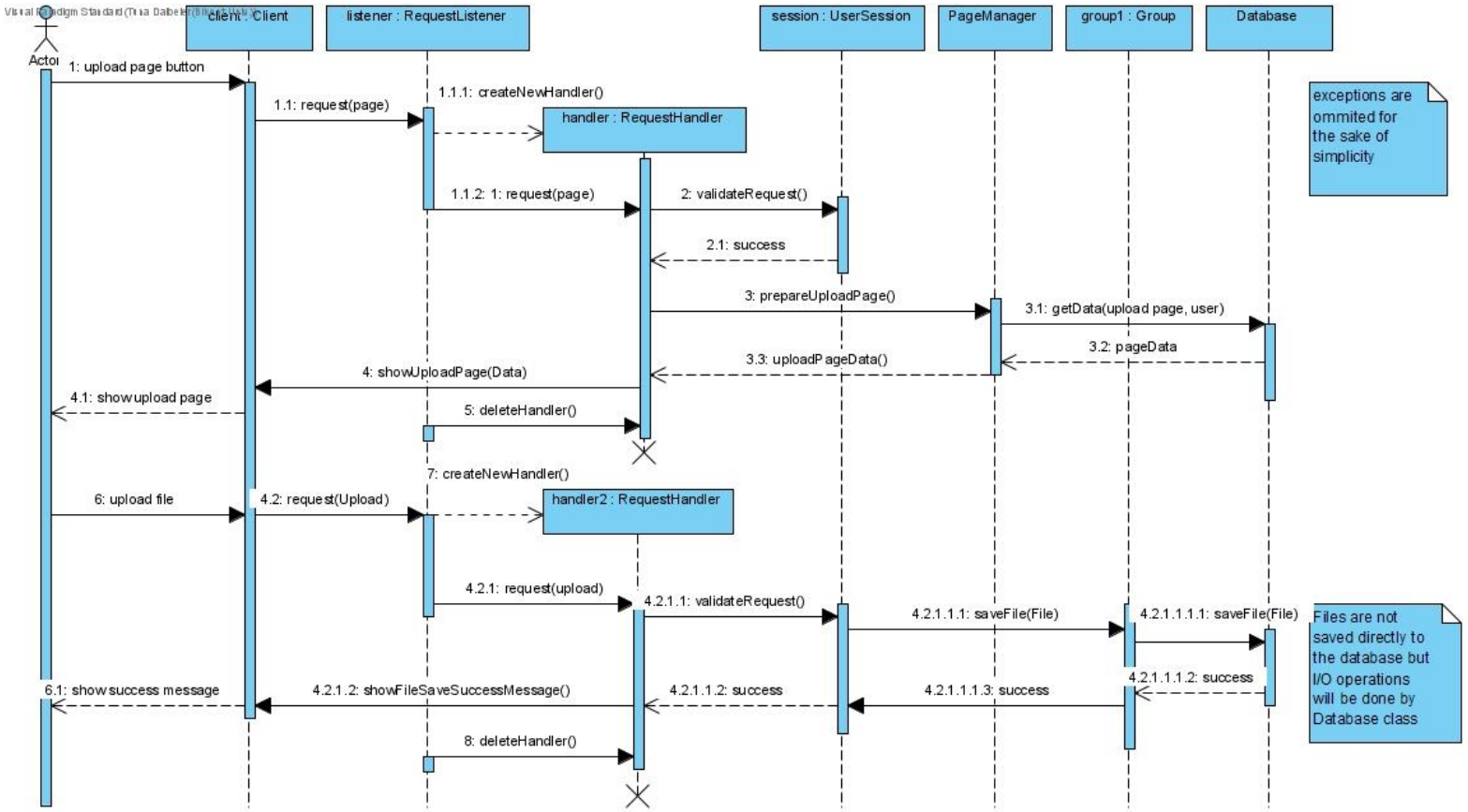

Figure 4 - Sequence Diagram - Request for upload page and uploading a file

Scenario 4: Tasks

Assumed the user is on the group page now. The student clicks the create task button and fills the form opened on the page. In the form, the student enters task info (who is assigned to the task, what is the due date, name of the task and details etc.)Then, the task request is sent to the server. The server passes the task info to a group class. Group class opens a new task object and records info to the object. Then the task gets recorded to the database and returns the user to a success message.

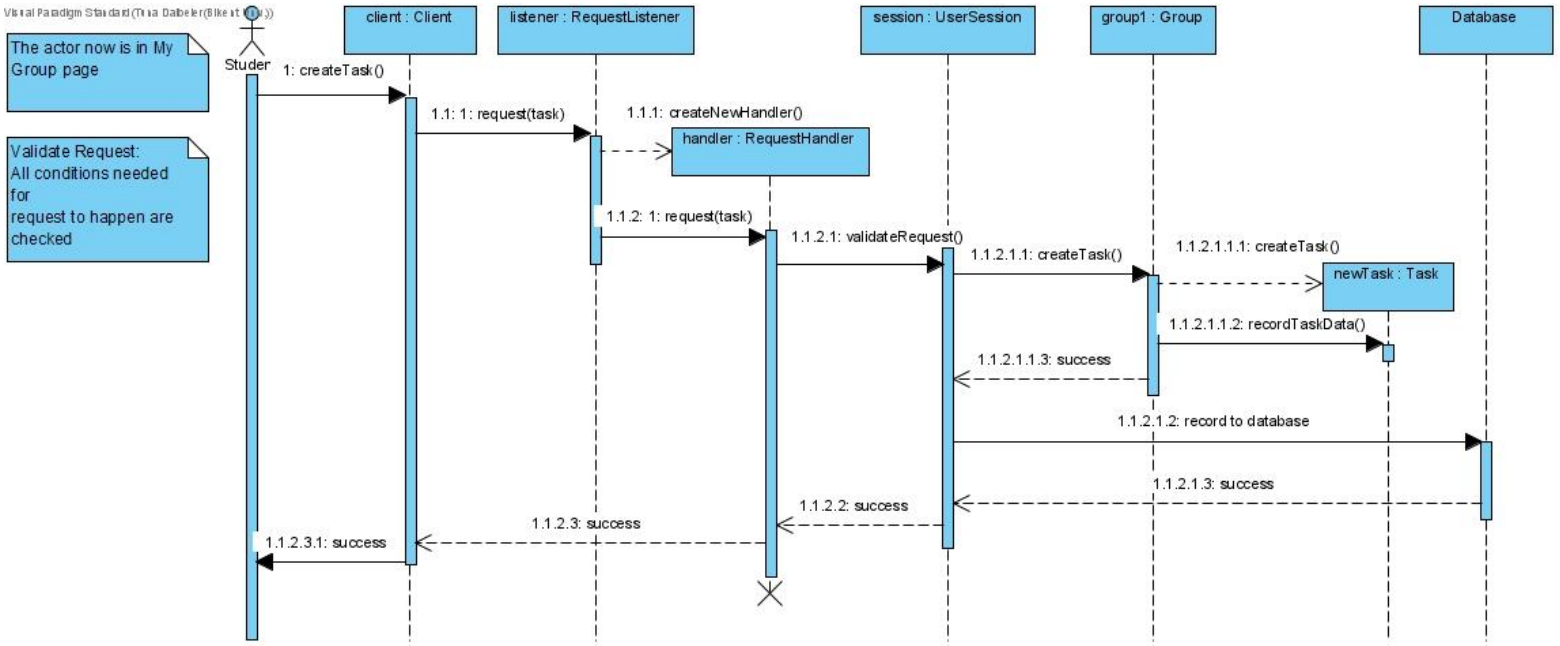

Figure 5 - Sequence Diagram - Creating a task

Scenario 5: In Group Peer Review

Same process with a couple of scenarios. A form sent from the client recorded into the database.

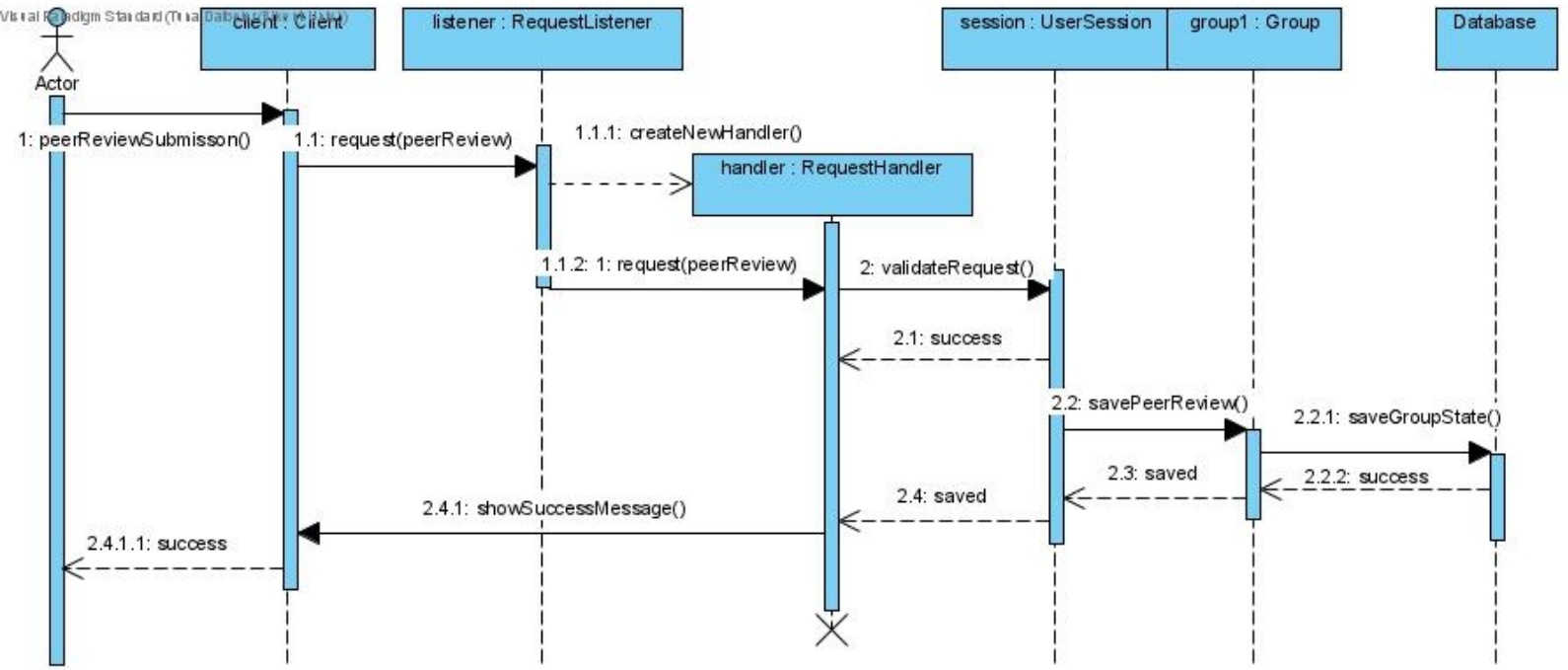

Figure 6 - Sequence Diagram - Peer Review

Scenario 6: Review of the Artifacts

First part of the diagram: An instructor creates a grading criteria for an assignment. This grading criteria includes a criteria - to be used in “review of artifacts” by students. This request gets recorded via AssignmentController. AssignmentController is the part of the controller system from MVC design. It might include more than one class depending on implementation.

Second part of the diagram: A student submits a review form. This form’s contents goes through AssignmentController to Assignment to Artifact. Gets saved in Artifact.

Third part of the diagram: A student requests comments (reviews) from the system. System retrieves comments from Artifact class. Then, because a student requested, anonymize the comments and return anonymised comments to the client.

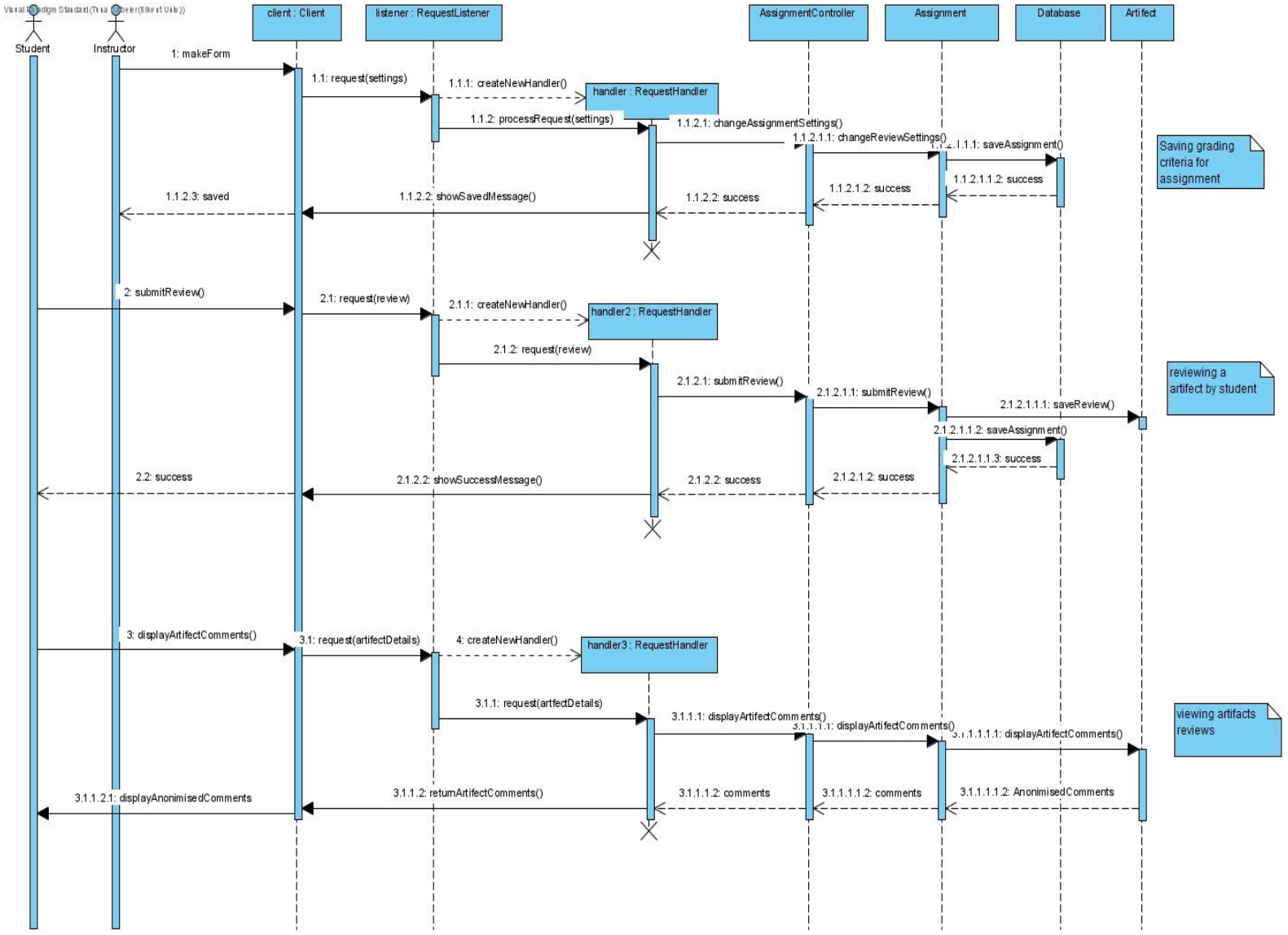

Figure 7 - Sequence Diagram - Review of artifact

3.3 Activity Diagrams

Activity Diagram 1: Login

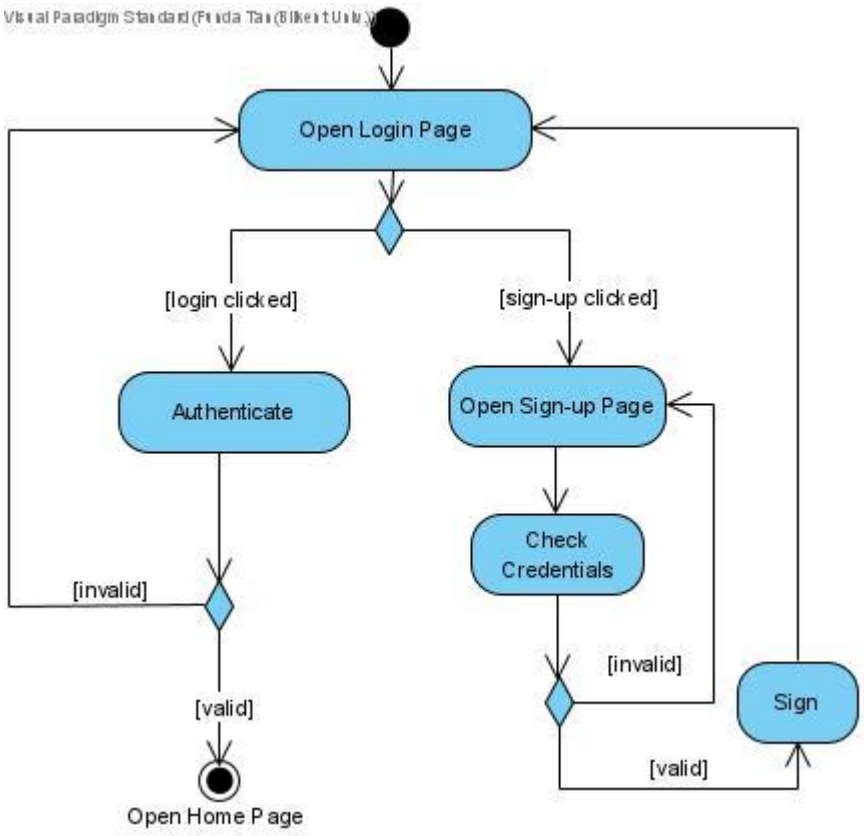

Figure 8 - Activity Diagram - Login

When the user opens the system, the login page is displayed. Users can be students or instructors at this point.

Case 1.1: If the login is clicked, the system will authenticate the user.

Case 1.1.1: If the authentication is invalid, system displays login page again.

Case 1.1.2: If the authentication is valid, system open “Home Page”.

Case 1.2: If the sign-up is clicked, the system will display the sign-up page then checks the credentials.

Case 1.2.1: If credentials are invalid, system displays the sign-up page again.

Case 1.2.2: If the credentials are valid, the system opens the login page again.

Activity Diagram 2: Group Formation

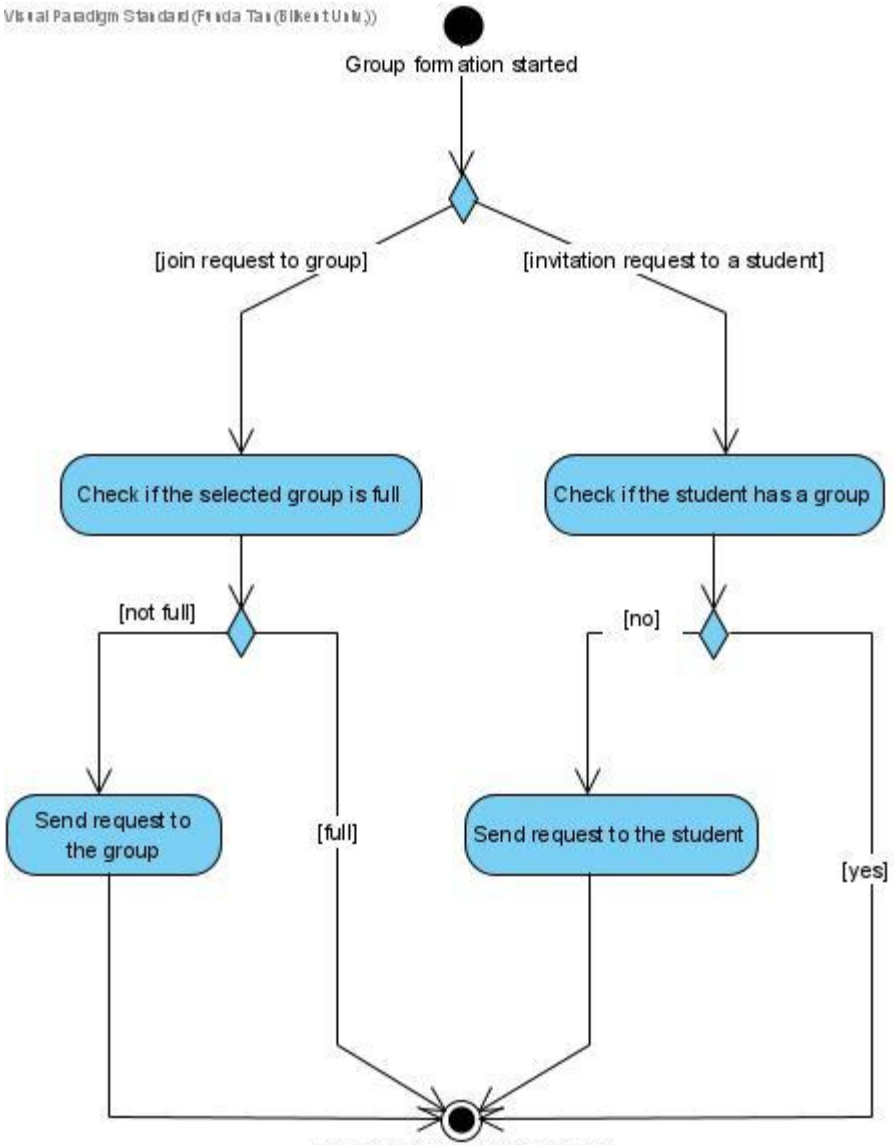

Figure 9 - Activity Diagram - Group Formation

For the group formation, there are two cases possible.

Case 1: Sending a join request to a group.

Case 1.1: If the group is not full, a request is sent from the user to the group.

Case 1.2: If the group is full, request will not be sent

Case 2: Sending an invitation request to a student.

Case 2.1: If the student does not have a group, system will send the request to the student.

Case 2.2: If the student has a group, the invitation will not be sent.

Activity Diagram 3: Artifact Review

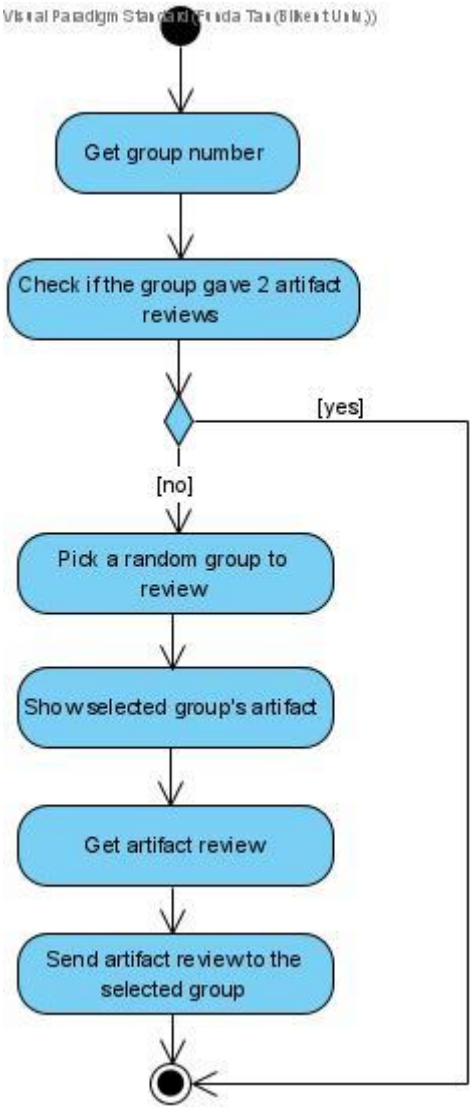

Figure 10 - Activity Diagram - Artifact Review

For the artifact review, there are two cases possible.
Case 1: System gets the group number, then checks if the group gave 2 artifact reviews. If no, the system picks a random group to review. Then, the system shows the selected group's artifact and gets the artifact review. Finally, the system sends the artifact review to the selected group.
Case 2: System gets the group number, then checks if the group gave 2 artifact reviews. If yes, the group cannot give any artifact reviews to the other groups.

Activity Diagram 4: Peer Review

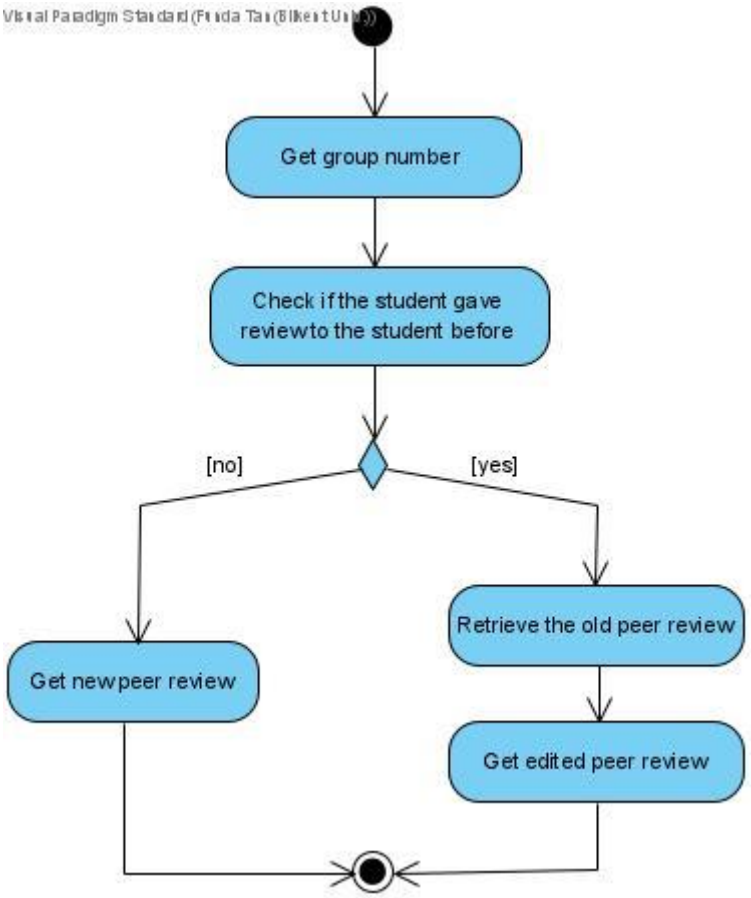

Figure 11 - Activity Diagram - Peer Review

For the peer review, there are two cases possible.
Case 1: System gets the group number, then checks if the group gave 2 artifact reviews. If no, the system picks a random group to review. Then, the system shows the selected group's artifact and gets the artifact review. Finally, the system sends the artifact review to the selected group.
Case 2: System gets the group number, then checks if the group gave 2 artifact reviews. If yes, the group cannot give any artifact reviews to the other groups.

3.4 State Diagram

State diagram for Course Project

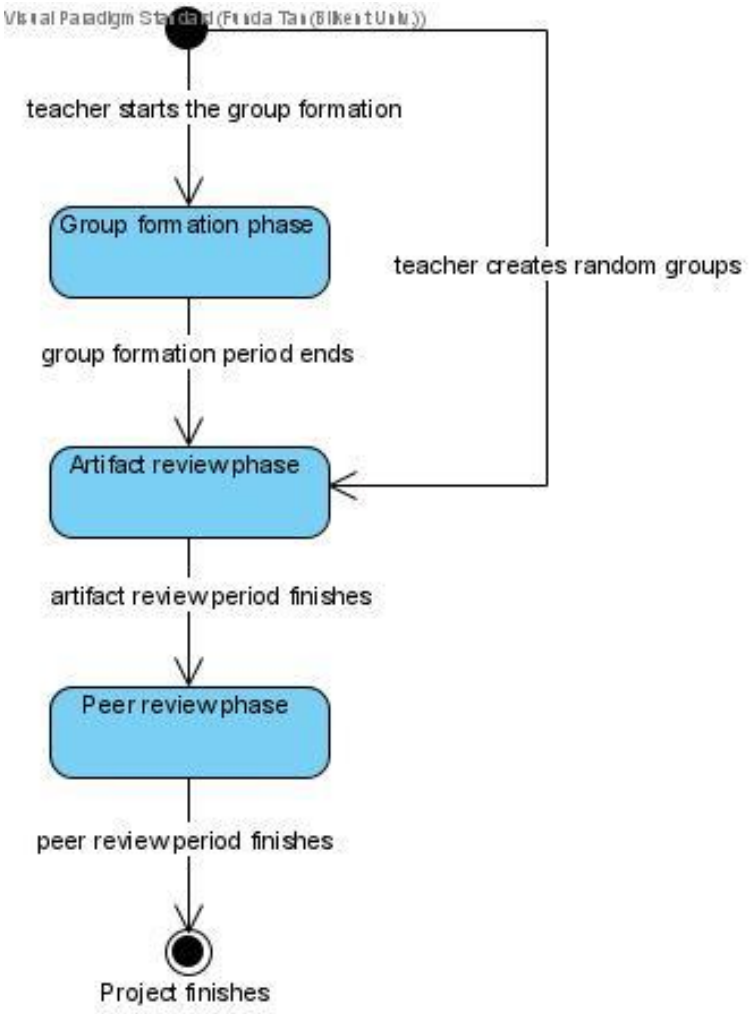

Figure 12 - State Diagram - State for Course Project

In the beginning of a course project, the instructor may start the group formation or create random groups. If the instructor creates random groups, the group formation phase will be skipped because random groups will be created and the course project will directly go to the artifact review phase. If the instructor starts the group formation, the course project will go to the group formation phase. Then, when the group formation period ends, the course project will begin the artifact review phase. If the artifact review period finishes, the course project will go to the peer review phase. Finally, when the peer review period finishes, the project finishes.

State diagram for Assignment

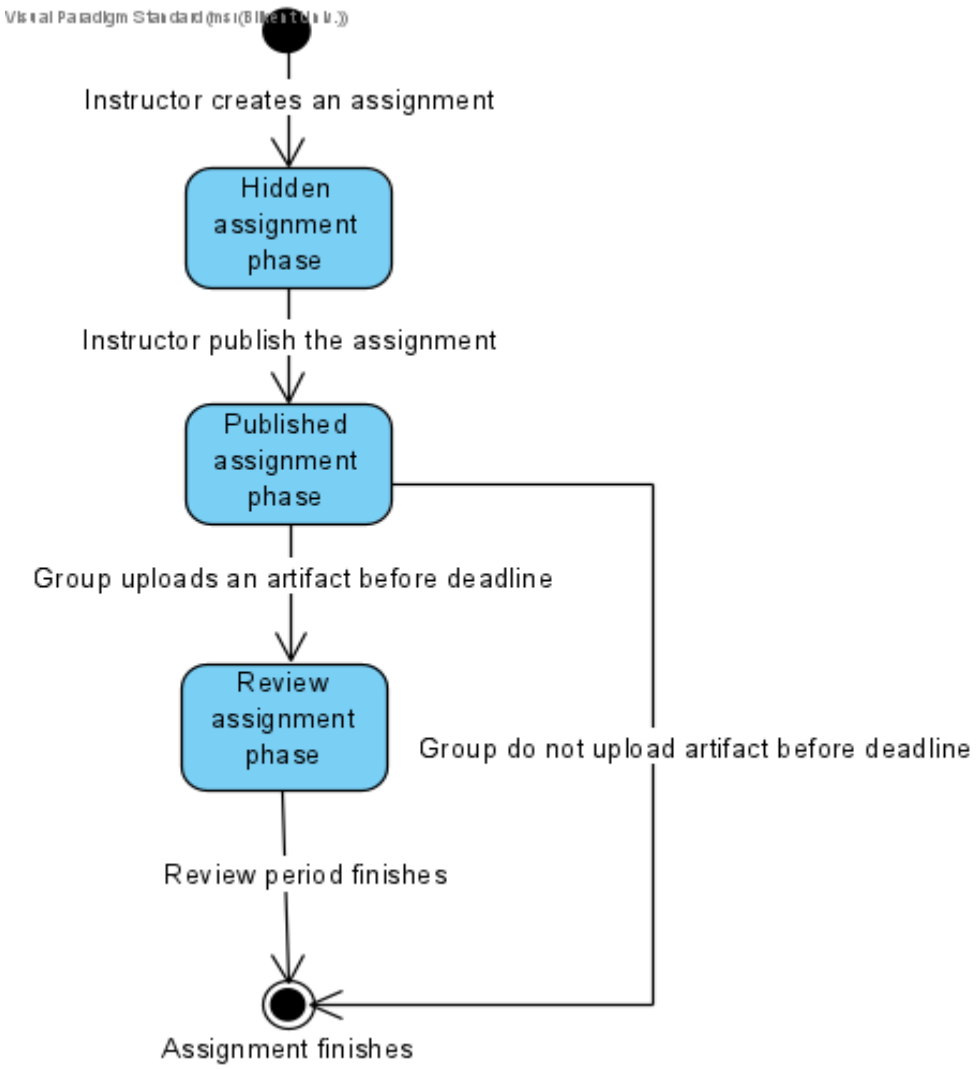

Figure 13 - State Diagram - State for Assignment

During the project, the instructor creates assignments. After the creation, the assignment is not visible to the students until the instructor decides to publish the assignment. After the publication of the assignment, groups can upload their artifacts to the system before the due date of the assignment. If there is an artifact that is uploaded by the group, the assignment goes to the review phase. Finally, when the review period finishes, the assignment finishes.

3.5 Object and Class Model

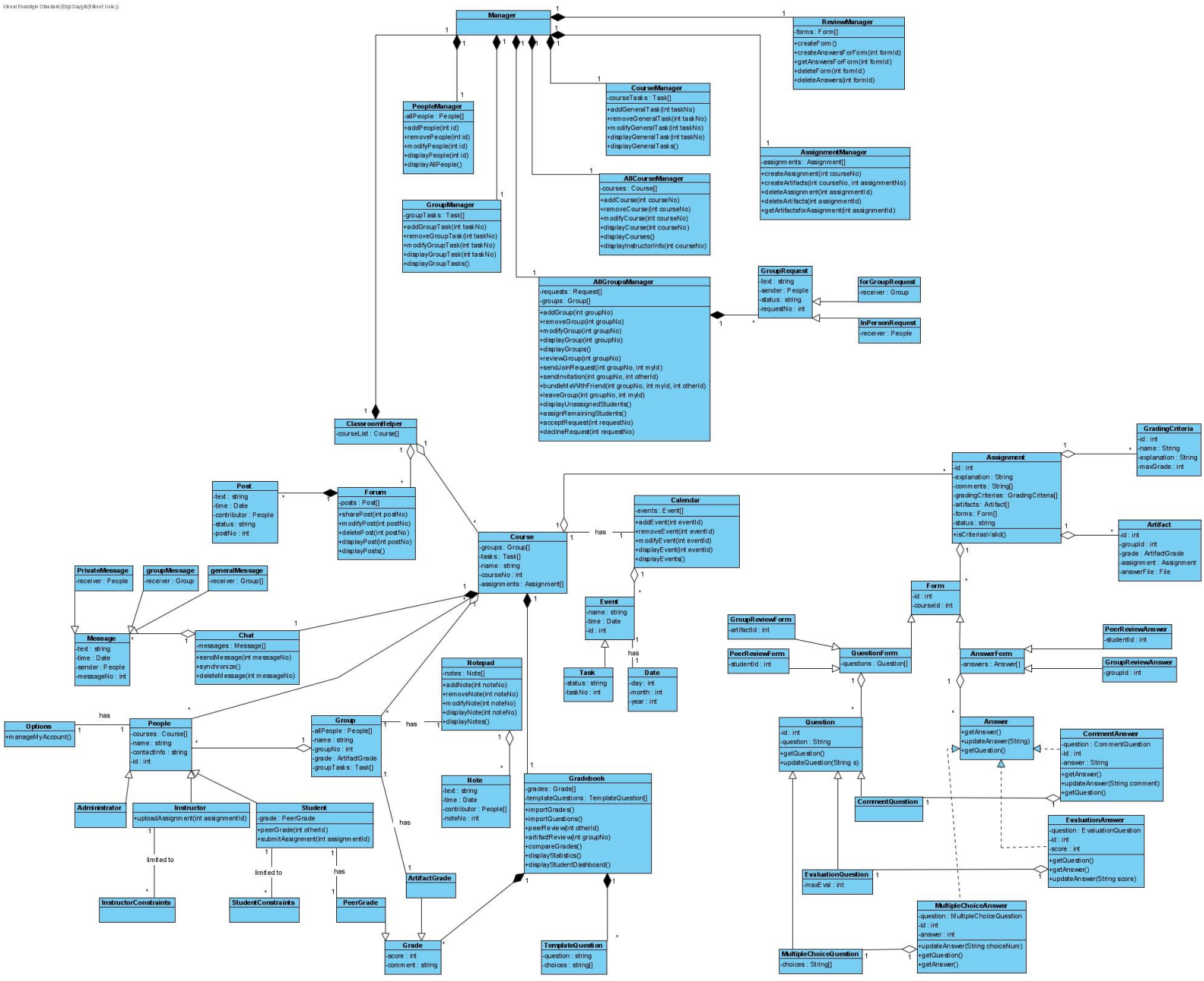

Figure 14 - Class Diagram

The class diagram of the “Classroom Helper” tool is illustrated above.

privateMessage class: A subclass of Message class with an addition of receiver as People specification.

groupMessage class: A subclass of Message class with an addition of receiver as Group specification.

generalMessage class: A subclass of Message class with an addition of receiver as a set of Groups specifications.

Message class: Defines a message with text, time, sender and message number to be used in chat. Specifies time as Date and sender as People.

Chat class: Defines a chat platform:

1) Provides a synchronous chat.
2) Enables private, group and general messaging.

Options class: Enables People to modify their name, contact information and ID.

Note class: Defines a note with text, time, contributor and note number to be used in Notepad. Specifies time as Date and contributor as a set of People.

Notepad class: Defines a note-taking platform for keeping record of the key points in a group meeting. Enables to modify and display Notes.

Post class: Defines a post with text, time, contributor, status and post number to be used in Forum. Specifies time as Date and contributor as People.

Forum class: Defines a forum platform which enables sharing Posts.

InstructorConstraints class: Defines constraints/limits for Instructor users.

StudentConstraints class: Defines constraints/limits for Student users.

Administrator class: A subclass of People class. Defines user as Administrator.

Instructor class: A subclass of People class. Defines user as Instructor. Enables Instructors to upload assignments.

Student class: A subclass of People class with an addition of peer grade specification. Specifies peer grade as PeerGrade. Defines user as Student. Enables Students to peer grade other students in his/her group and submit assignments.

People class: Users are people. Defines People with name, contact information, ID and a set of Courses.

Group class: People form groups. Defines a Group with name, group number and a set of People.

Course class: Course consists of groups. Defines a Course with name, course number and a set of Groups.

Date class: Defines a date with day, month and year to be used in events.

Event class: Defines an event with name, time and id. Specifies time as Date.

Task class: Subclass of Event class with an addition of number and status specifications.

Calendar class: Defines a calendar which enables to modify and display Events.

TemplateQuestion class: Defines a template question with a question and a set of choices to be used in peer review and artifact review.

PeerGrade class: A subclass of Grade class. Defines the grade of a student given through peer review to be used in Gradebook.

ArtifactGrade class: A subclass of Grade class. Defines the grade of a group given through artifact review to be used in Gradebook.

Grade class: Defines a grade with a score and comments.

Gradebook class: Defines a student dashboard platform. Defines the following operations:

1) Provides artifact review functionality. Enables groups to review and grade each other anonymously.

2) Provides in-group peer review functionality. Enables group members to review and grade each other anonymously.

3) Provides statistics by comparing Grades given through artifact review with imported grades given by Instructor.

4) Contains template review questions and imports new review questions from the user file provided.

GroupRequest class: A subclass of Request class with an addition of receiver as Group specifications.

InPersonRequest class: A subclass of Request class with an addition of receiver as People specifications.

Request class: Defines a request with a text, request number, status and sender to be used in grouping. Specifies sender as People.

PeopleManager class: Manages People by adding, removing, modifying and displaying the People.

GroupManager class: Manages a Group by adding, removing, modifying and displaying the group Tasks.

AllGroupsManager class: Manages all Groups by adding, removing, modifying, reviewing and displaying the groups. Also, defines the following operations:

1) Send join Request to a Group
2) Send invitation Request to People
3) Send bundle me with friend Request to a Group
4) Accept/decline a Request
5) Leave a Group
6) Display unassigned Students
7) Assign remaining Students

CourseManager class: Manages a Course by adding, removing, modifying and displaying general Tasks.

AllCourseManager class: Manages all Courses by adding, removing, modifying and displaying Courses.
AssignmentManager class: Manages all assignments by creating assignments and their corresponding artifacts, deleting assignments and their corresponding artifacts and displaying assignments.
ReviewManager class: Manages all forms by creating forms and their corresponding answers, deleting forms and their corresponding answers and displaying answers.
Manager class: Puts all manager classes together for simplicity.
Assignment class: Defines an assignment with an id, explanation, comments, grading criterias, artifacts, forms and status. Specifies grading criterias as a set of GradingCriterias, artifacts as a set of Artifacts and forms as a set of Forms.
GradingCriteria class: Defines a grading criteria with an id, name, explanation and maximum grade.
Artifact class: Defines an artifact with an id, group id, artifact grade, assignment and answer file. Specifies artifact grade as ArtifactGrade, assignment as Assignment and answer file as File.
Form class: Defines a form with an id and course id.
QuestionForm class: A subclass of Form class with an addition of questions specification. Specifies questions as a set of Questions.
AnswerForm class: A subclass of Form class with an addition of answers specification. Specifies answers as a set of Answers.
GroupReviewForm class: A subclass of QuestionForm class with an addition of artifact id specifications.
PeerReviewForm class: A subclass of QuestionForm class with an addition of student id specifications.
GroupReviewAnswer class: A subclass of AnswerForm class with an addition of group id specifications.
PeerReviewAnswer class: A subclass of AnswerForm class with an addition of student id specifications.
Question class: Defines a question with an id and question statement.
CommentQuestion class: A subclass of Question class.
EvaluationQuestion class: A subclass of Question class with an addition of maximum evaluation value specification.
MultipleChoiceQuestion: A subclass of Question class with an addition of choices specification.
Answer class: Defines an interface for CommentAnswer, EvaluationAnswer and MultipleChoiceAnswer classes.

CommentAnswer class: A class implementing Answer class with an addition of question, id and answer specifications. Specifies question as CommentQuestion.
EvaluationQuestion class: A class implementing Answer class with an addition of question, id and score specifications. Specifies question as EvaluationQuestion.
MultipleChoiceQuestion class: A class implementing Answer class with an addition of question, id and answer specifications. Specifies question as MultipleChoiceQuestion.

3.6 User interface - navigational paths and screen mock-ups

3.6.1 Navigational Paths

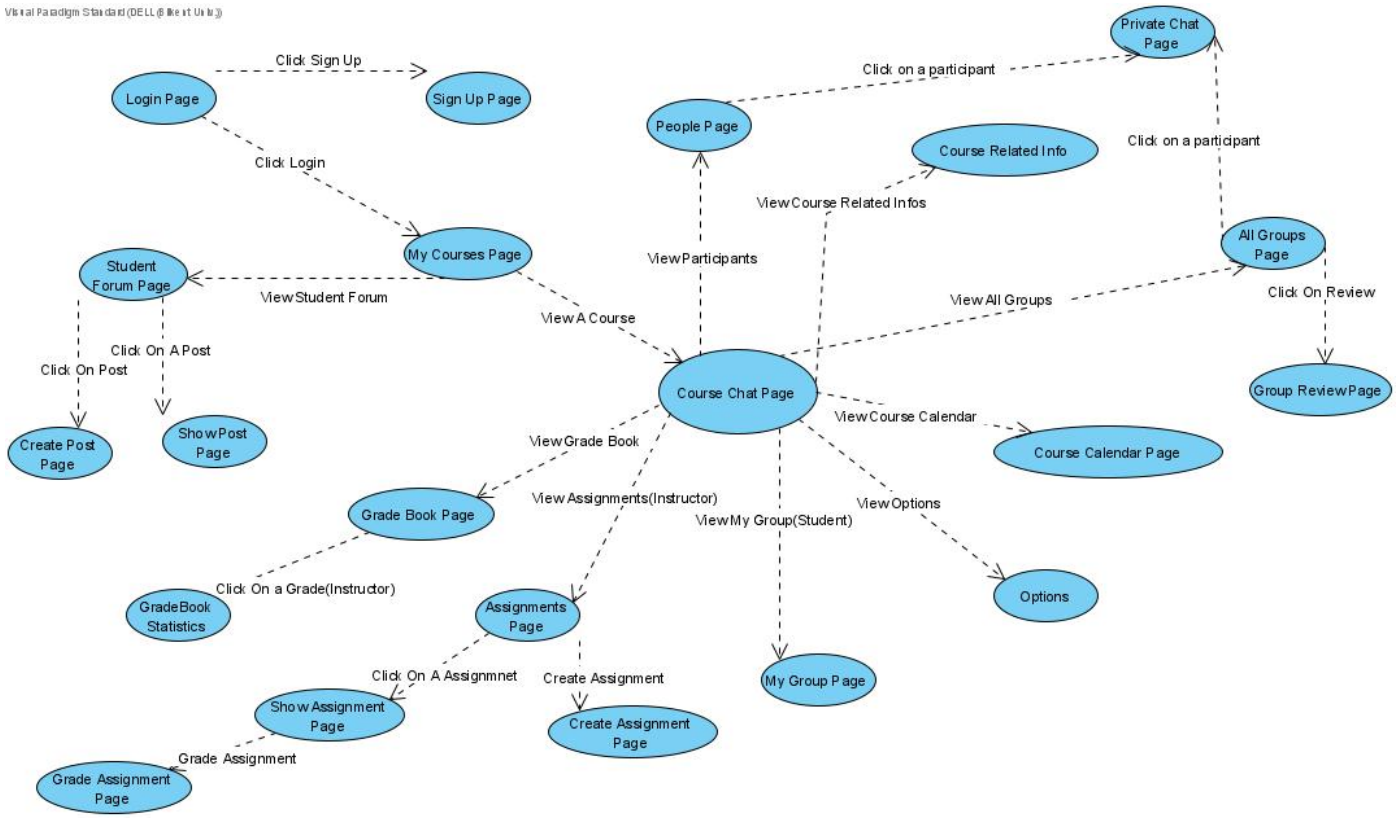

Figure 15 - Navigational Paths

3.6.2 Screen Mock-Ups

3.6.2.1 Login Page

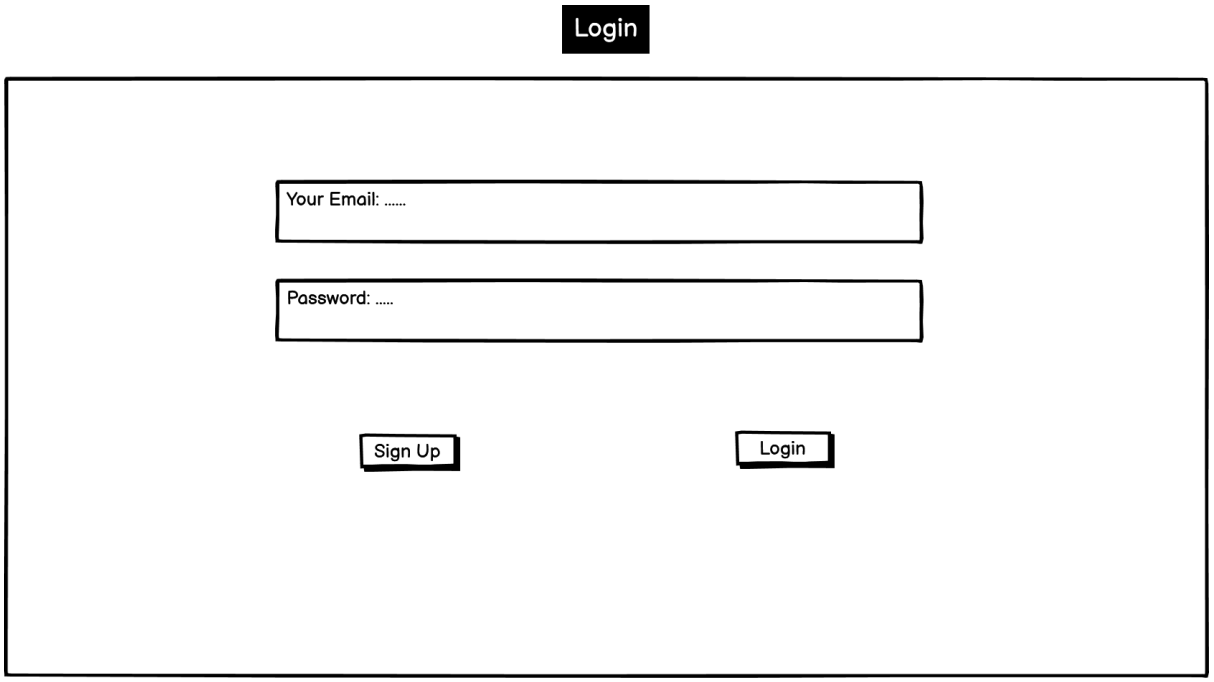

Figure 16 - Screen Mock- Ups - Login page

The initial page is the login page. If the sign up button is clicked, then the sign up page opens, if the user enters valid email and password, then the home page opens up.

3.6.2.2 Sign Up Page

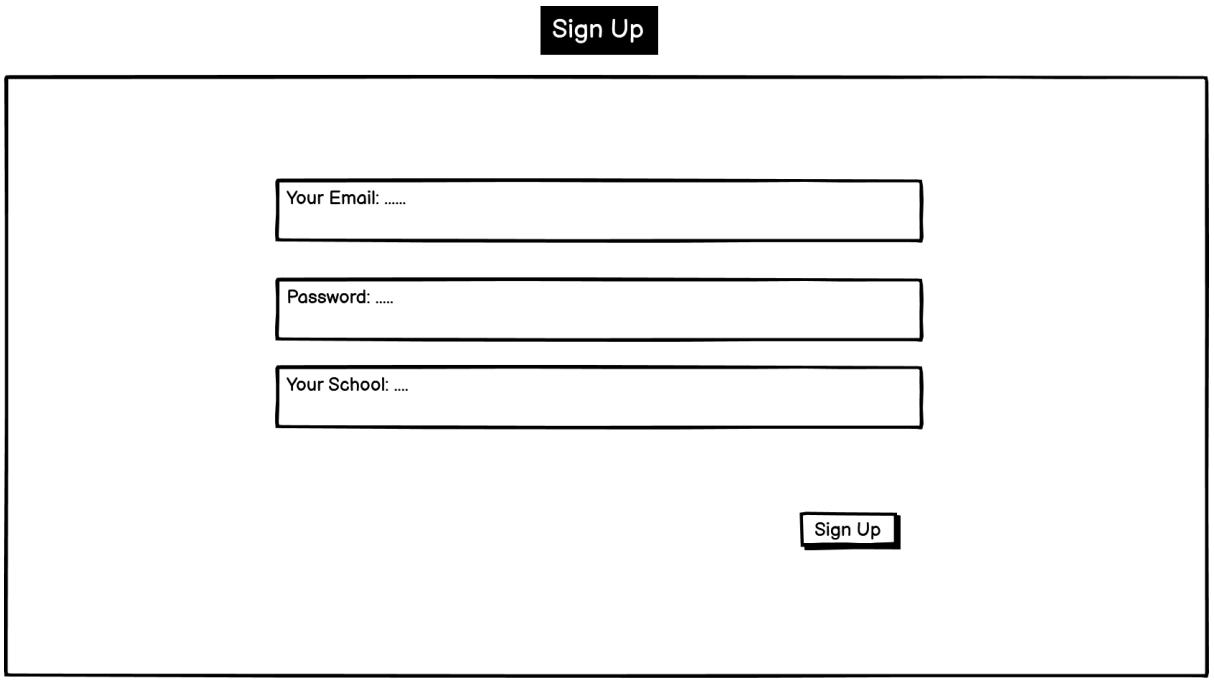

Figure 17 - Screen Mock- Ups - Sign up page

3.6.2.3 My Courses Page

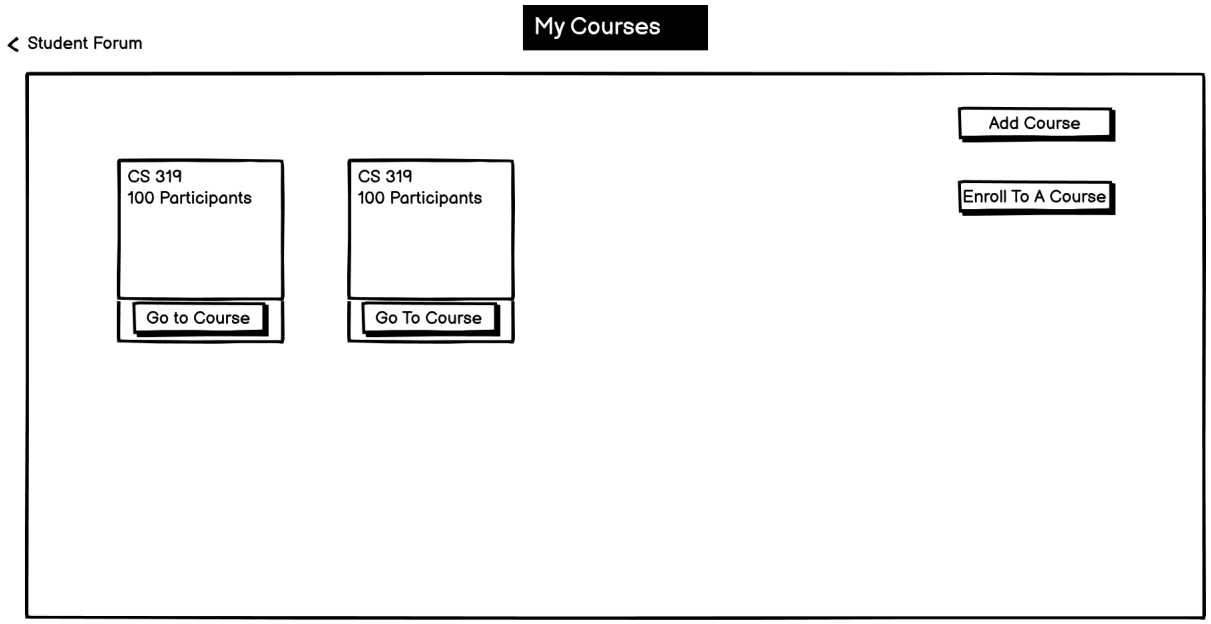

Figure 18 - Screen Mock-Ups - My Courses Page

This is the homepage for the application. It shows enrolled courses. All courses have buttons which direct the user to the course chat page. There are two buttons which enable creating a course or enrolling to a specific course. Users can switch to the student forum by student forum button.

3.6.2.4 Course Chat Page

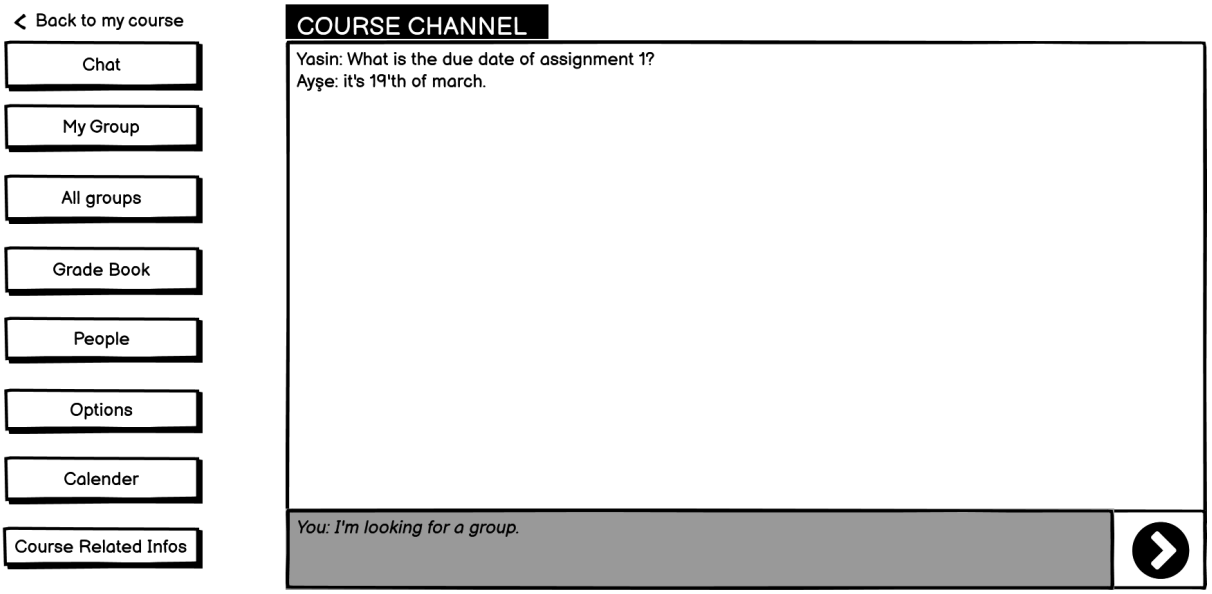

Figure 19 - Screen Mock-Ups - Course chat page

This is the page where the users interact with the synchronous chat of course. There is a menu for all course related pages which enables transitions between pages. This menu has eight buttons Chat, My Group(Assignments for instructors), All Groups, Grade Book, People,

Options, Calendar and Course Related info. Also, there is a button which allows the user to return to the home page. The Course Chat page is where the Chat button is linked.

3.6.2.5 My Group Page

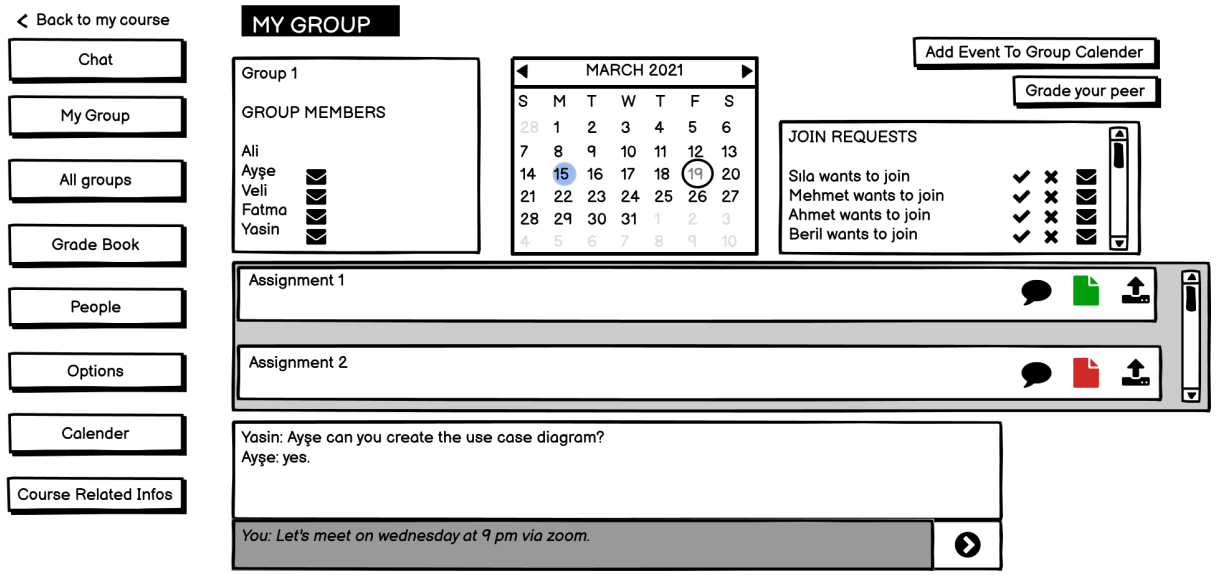

Figure 20 - Screen Mock-Ups - My Group Page

My Group page is where the students interact with his/her group. The student can see group mates, tasks, task calendar, assignments, join requests and group chat on this page. Students upload given assignments and organize group tasks on this page.

3.6.2.6 All Groups Page

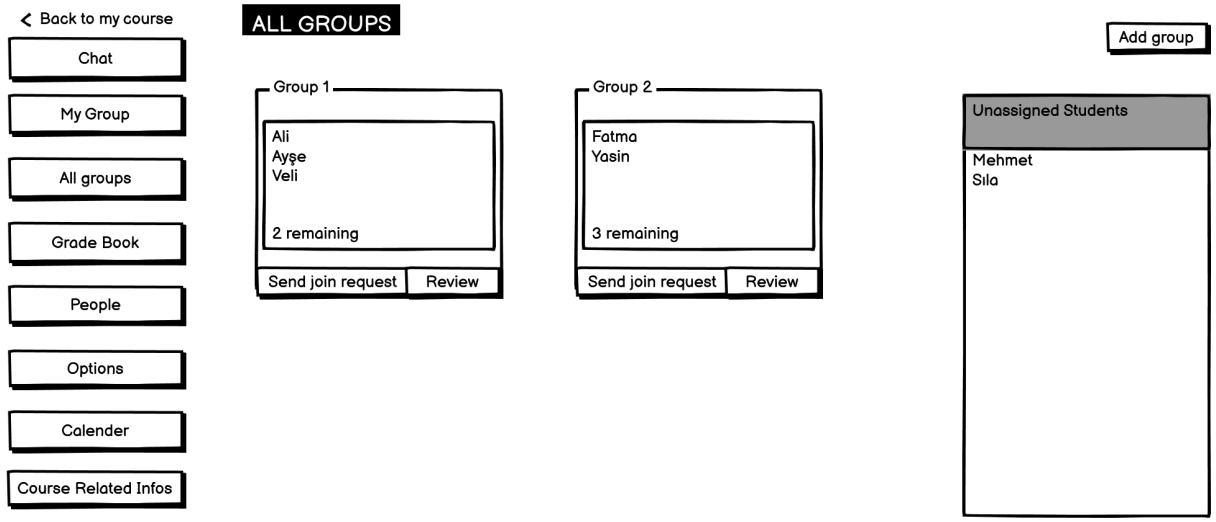

Figure 21 - Screen Mock-Ups - All Groups Page

This is the page where users can interact with groups of the course. They can see created groups, their members and unassigned students. Students can create groups by the Add

group button if they did not join a group. Also, every group has two buttons: Send join request and Review. Send join button creates a join request to join that group. Review opens a page which shows the work of the group which will be reviewed.

3.6.2.7 Group Review Page

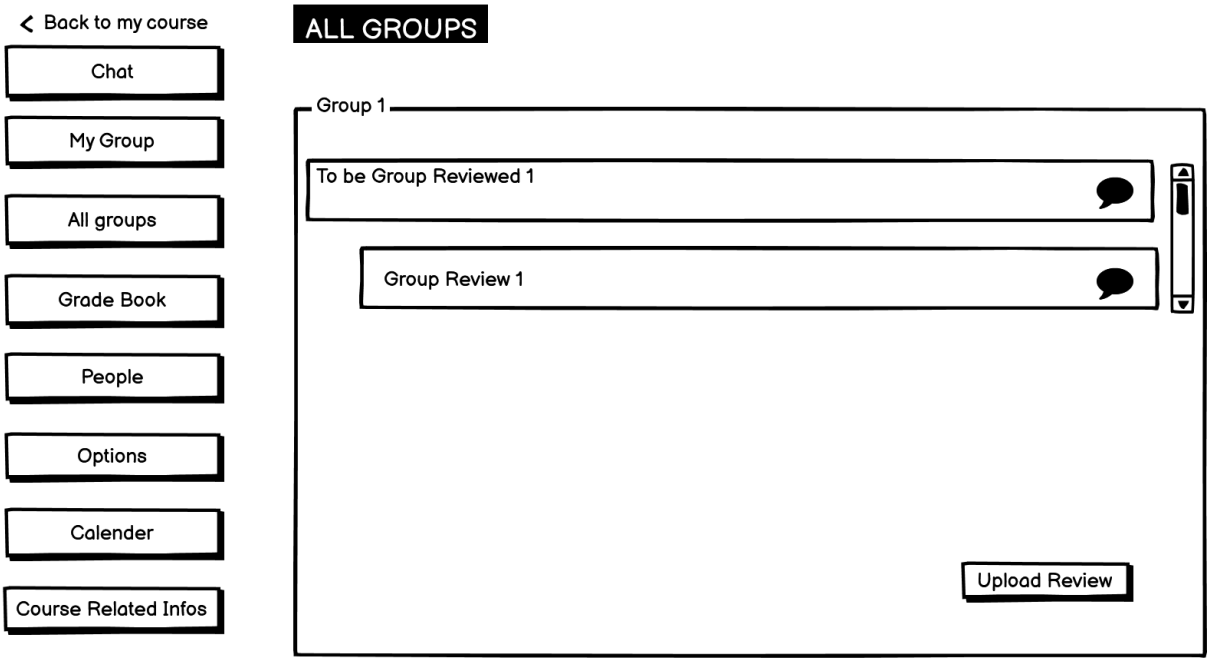

Figure 22 - Screen Mock-Ups - All Groups

This page allows students to get other groups' to be reviewed assignments and review them. Uploading a review is done by the Upload Review button.

3.6.2.8 Grade Book page

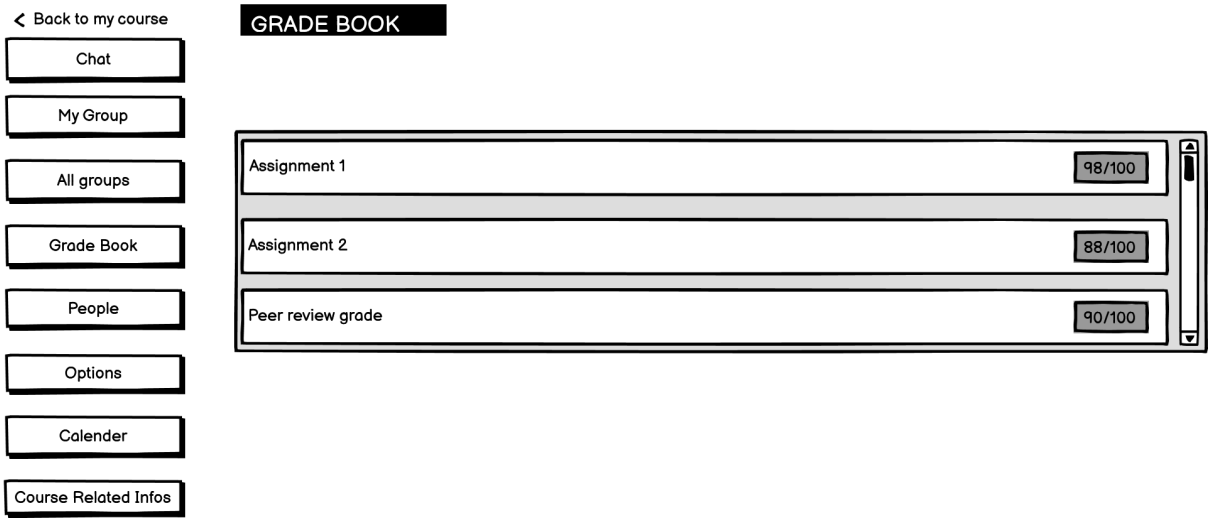

Figure 23 - Screen Mock-Ups - GradeBook Page

This page shows graded assignments. Students will see grades they get from their assignments. Instructors will see a statistics button for every assignment and when clicked application will direct the instructor to the statistics page of the assignment.

3.6.2.9 Grade Book page continues (Instructor view)

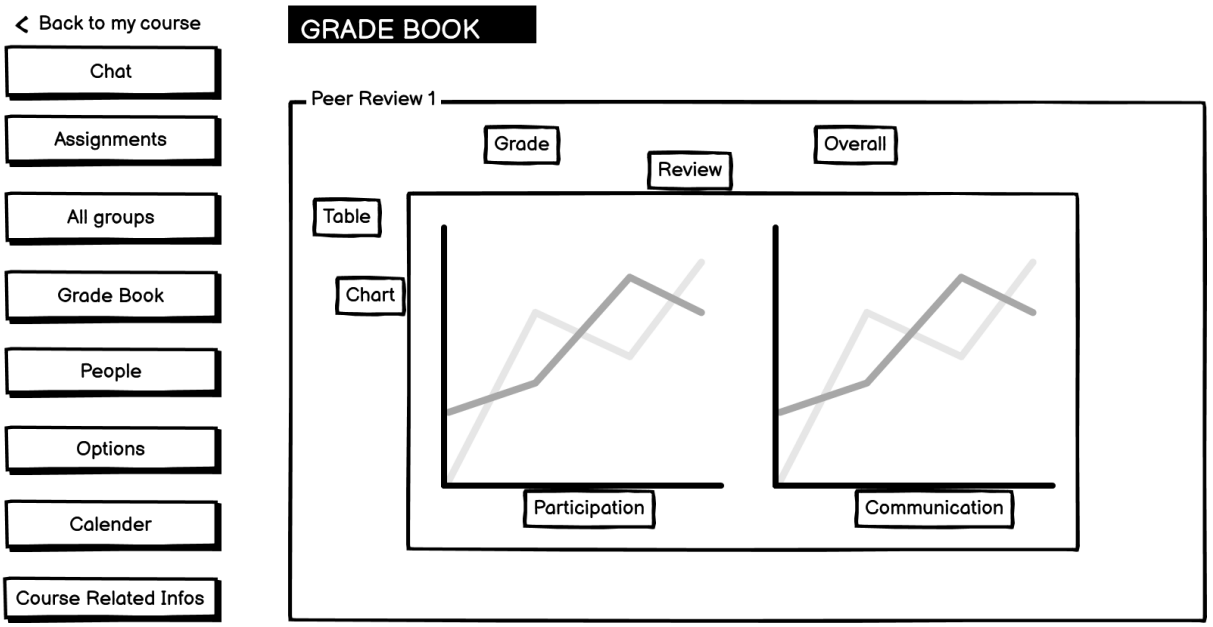

Figure 24 - Screen Mock-Ups - GradeBook Page (Instructor)

Statistics page shows statistics of grades of students. There may be several options for demonstration of data.

3.6.2.10 People Page

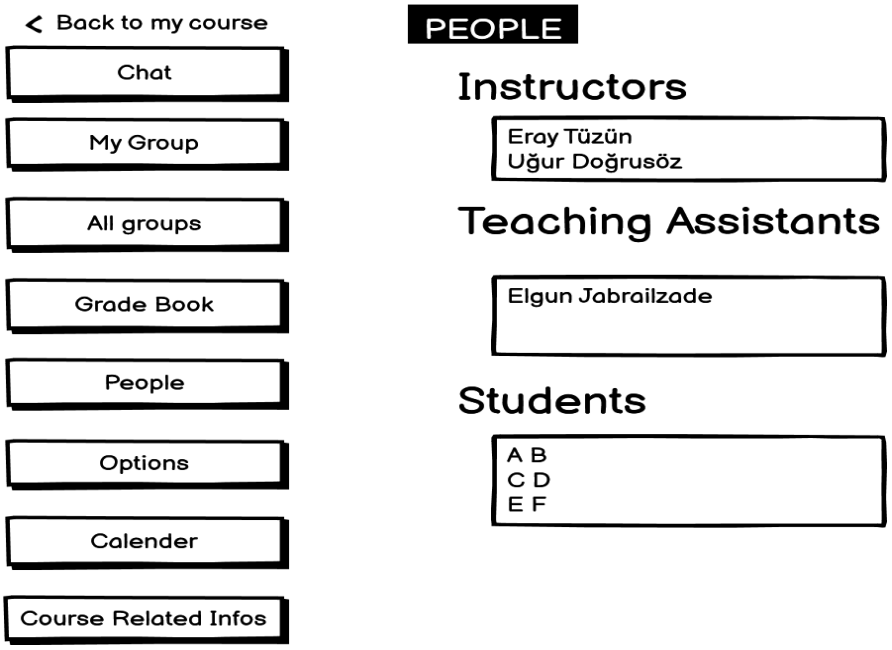

Figure 25 - Screen Mock-Ups - People Page

This page shows participants of the course in different categories such as instructors and TAs. Users can click on the names and message them. When clicked, they will be directed to the Private Chat Page.

3.6.2.11 Private Chat Page

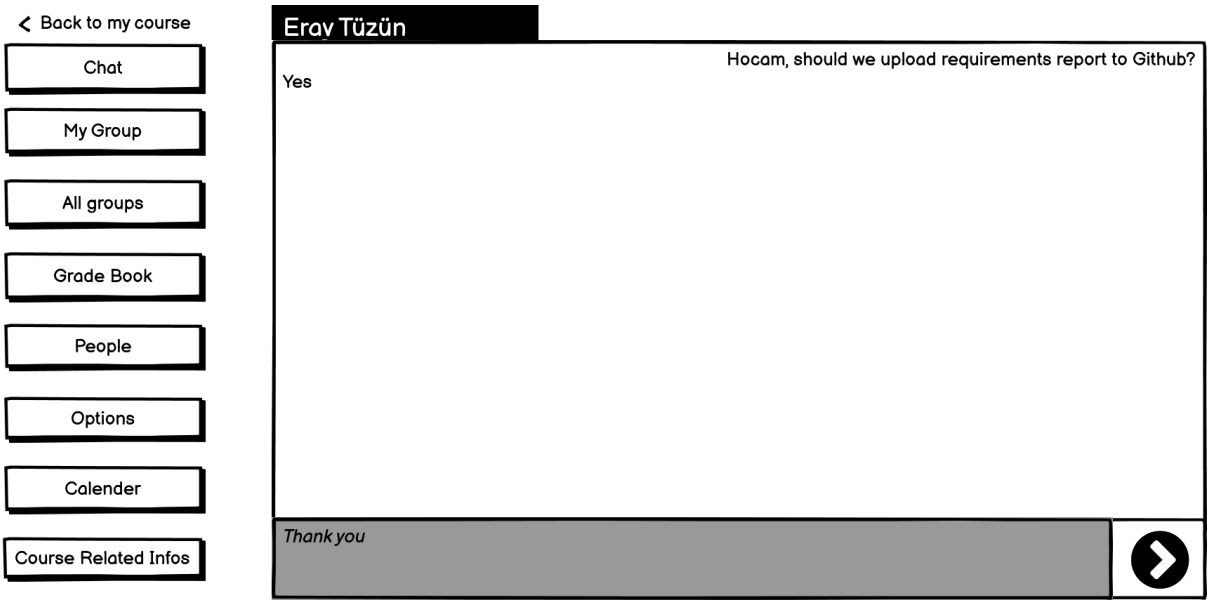

Figure 26 - Screen Mock-Ups - Chat

This page allows the user to message with other participants of the course.

3.6.2.12 Course Calendar Page

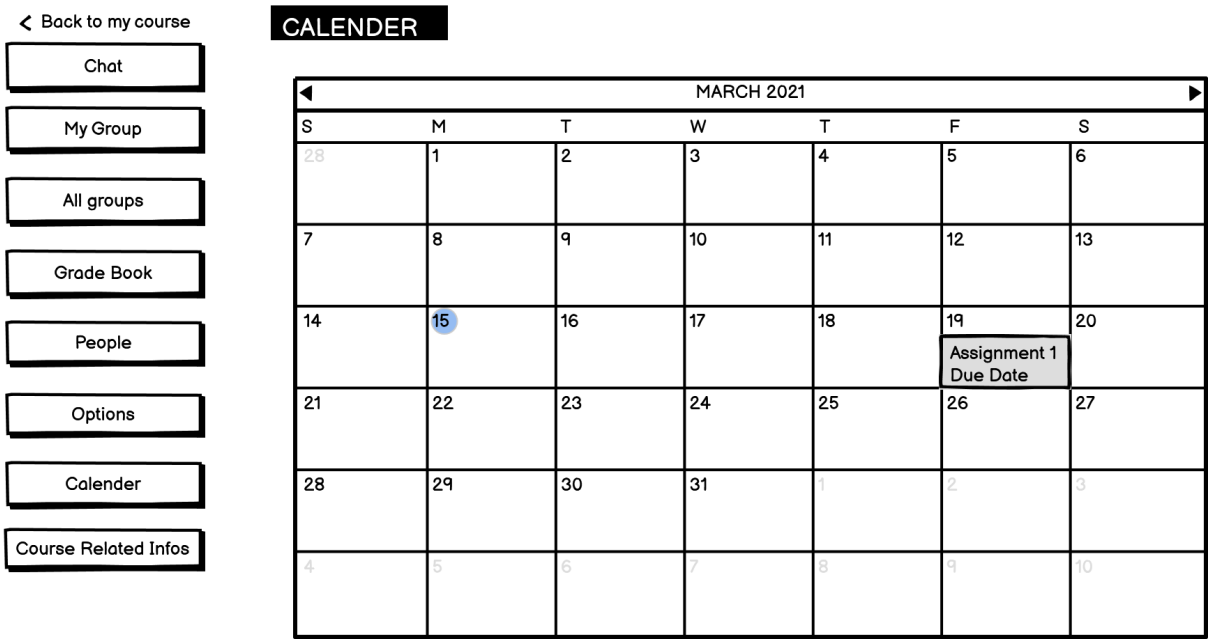

Figure 27 - Screen Mock-Ups - Course Calendar

This page shows event dates for the course as assignment due dates, task due dates.

3.6.2.13 Course Related Info

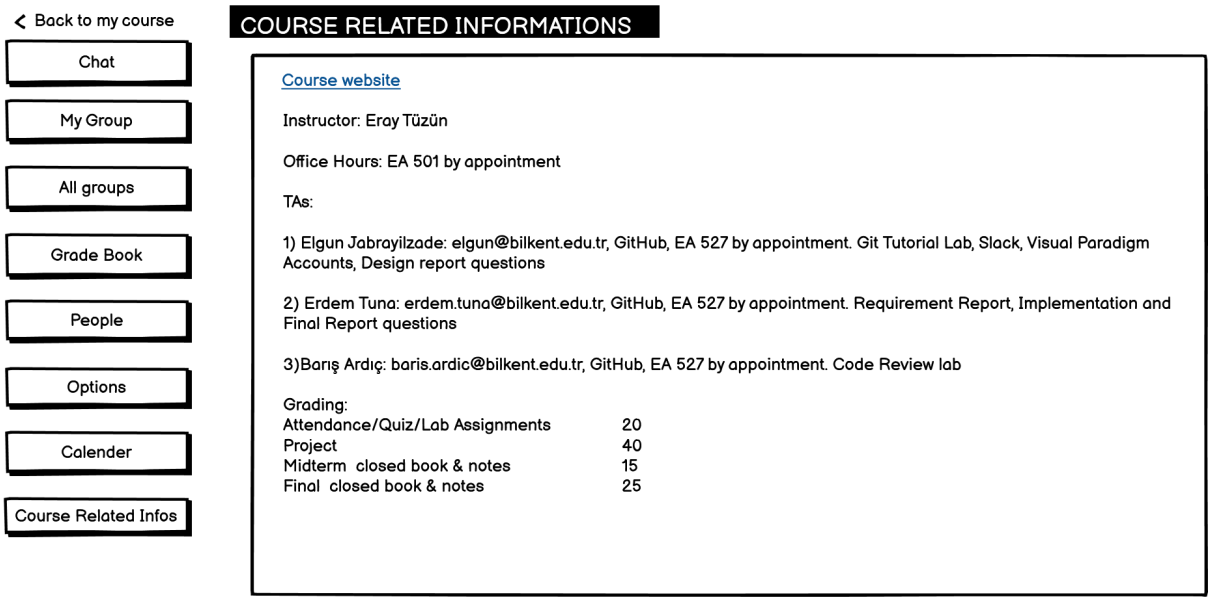

Figure 28 - Screen Mock-Ups - Course Related Info

This page shows the information about the course and contact information of instructors.

3.6.2.14 Assignments Page (Instructor view)

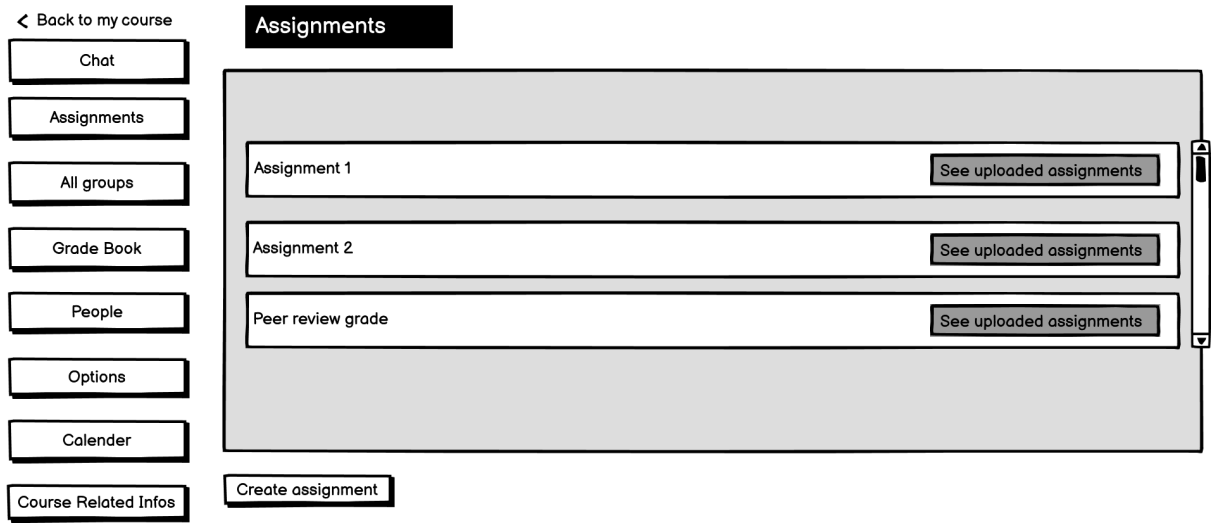

Figure 29 - Screen Mock-Ups - Assignments Page

For instructors of course, there is the Assignments section instead of My Groups section in the menu. The instructor can create and reach assignments. When the Create Assignment button is clicked, the Create Assignment Page opens. Also, every assignment has their own “See Uploaded Assignments” which is linked to “Show Assignment Page”.

3.6.2.15 Create Assignment Page

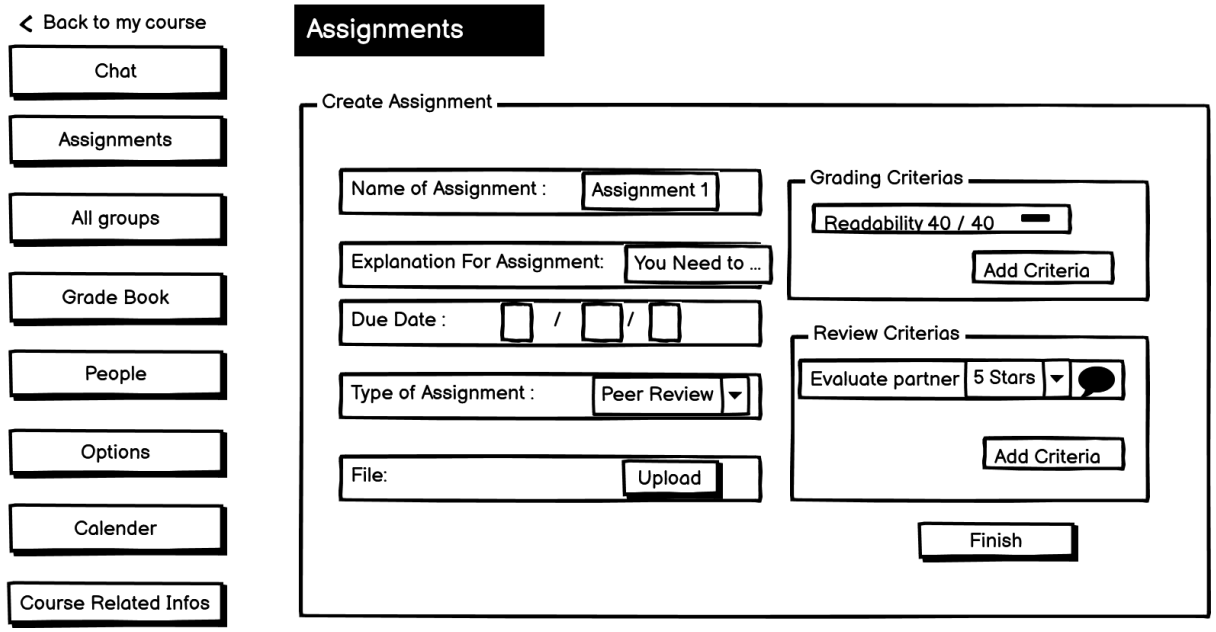

Figure 30 - Screen Mock-Ups - Create Assignment (Instructor)

This page allows the instructor to create an assignment. The instructor may specify important characteristics of the assignment such as name, due date,explanation for the assignment and also if there is a pdf file for the assignment. Also, the instructor may specify the type of the assignment, such as individual homework, peer review or group review. Then there could be criterias which will be added by the instructor for easy grading and demonstrating the grades. The types of criterias would be grading criterias which will allow the instructor to grade the assignment in an organized way. Also, there are review criterias for review assignments which will allow students to review each other in an organized way. After clicking finish, assignment will be created and the application will return to “Assignments Page”.

3.6.2.16 Show Assignment Page

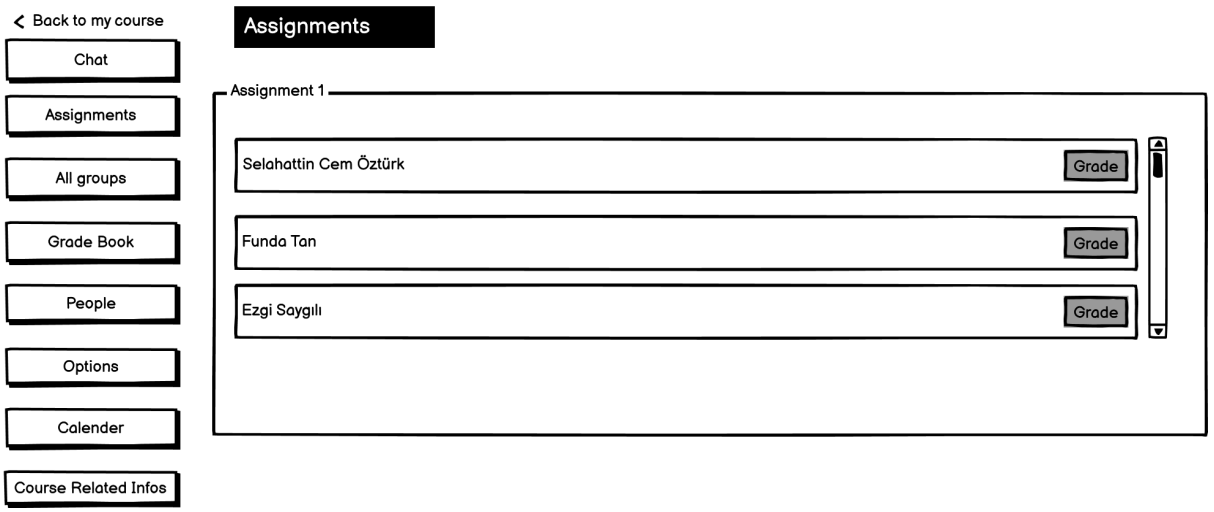

Figure 31 - Screen Mock-Ups - Show Assignment

This page shows grades of students and if they are not graded, there will be a Grade button which will direct the instructor to “Grade Assignment Page”. Also, the instructor will be able to get uploaded files for the assignment by clicking on names.

3.6.2.17 Grade Assignment Page

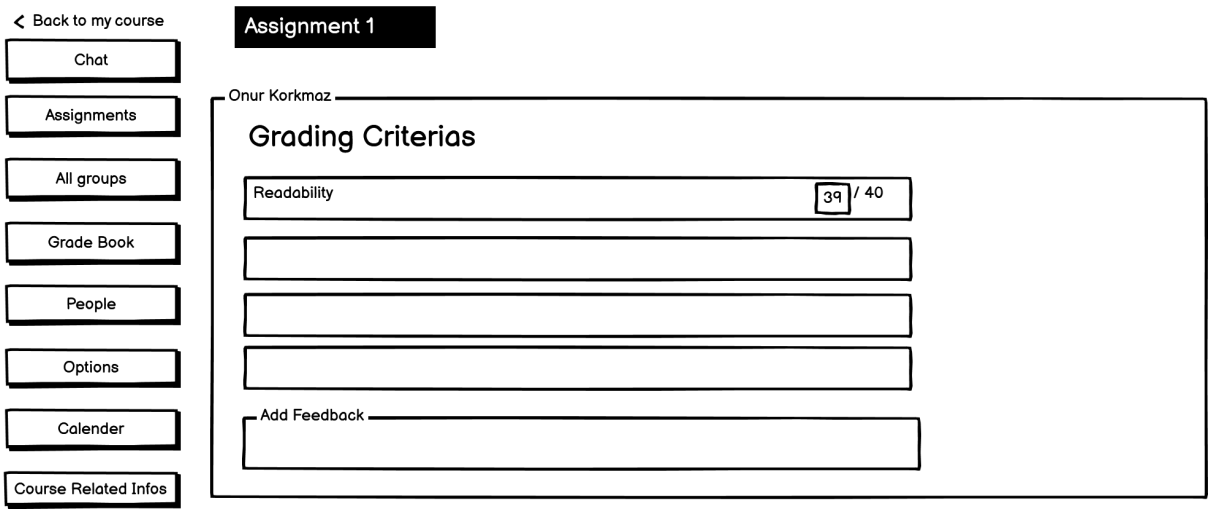

Figure 32 - Screen Mock-Ups - Grade Assignment (Instructor)
This page allows the instructor to grade assignments. The page will show criterias that the instructor attached to the assignment as grading criterias. The instructor will be able to specify grades. Also, the instructor may add feedback for the grade.

3.6.2.18 Student Forum Page

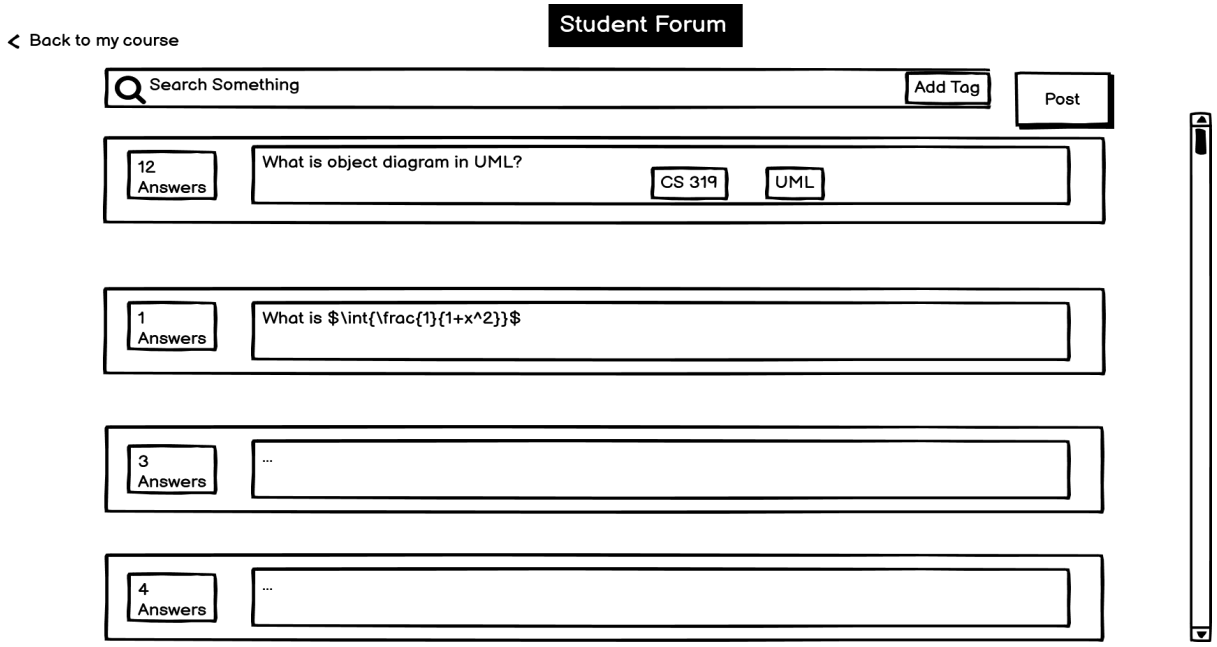

Figure 33 - Screen Mock-Ups - Student Forum

Student forum page is a section of the application which is independent from courses' pages. It can be reached by "My Courses Page" and it is linked to that page. This page shows questions that are asked in the forum. Post button is linked to "Create Post Page" and

also when clicked on a question, the user is directed to “Show Post Page”. Also the user may search posts by keywords.

3.6.2.19 Create Post Page

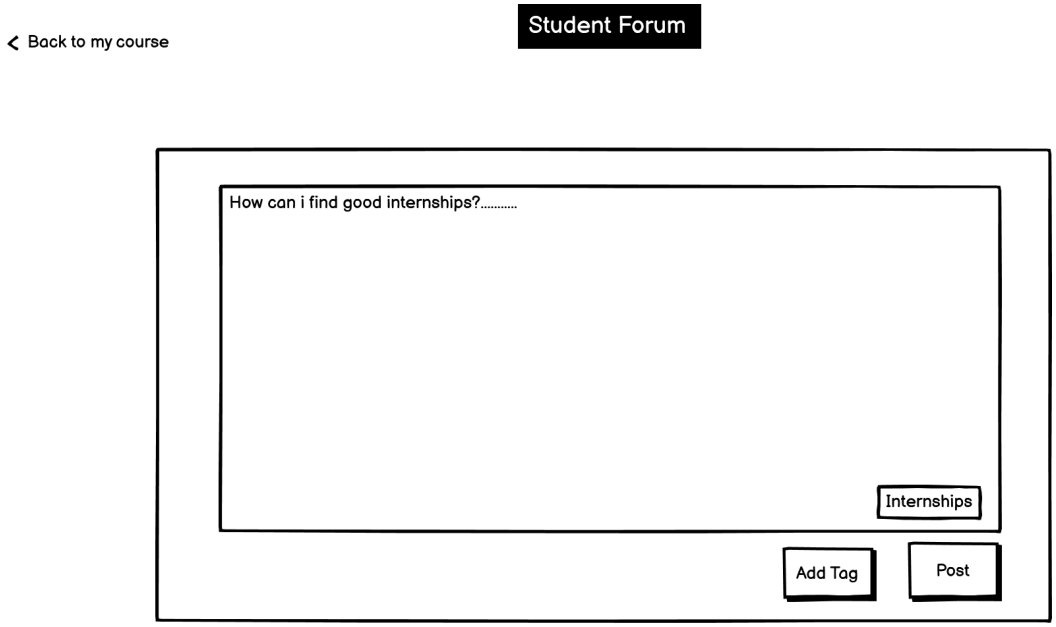

Figure 34 - Screen Mock-Ups - Create Post Page

This Page allows the user to create a post and write the question of the post.

3.6.2.20 Show Post Page

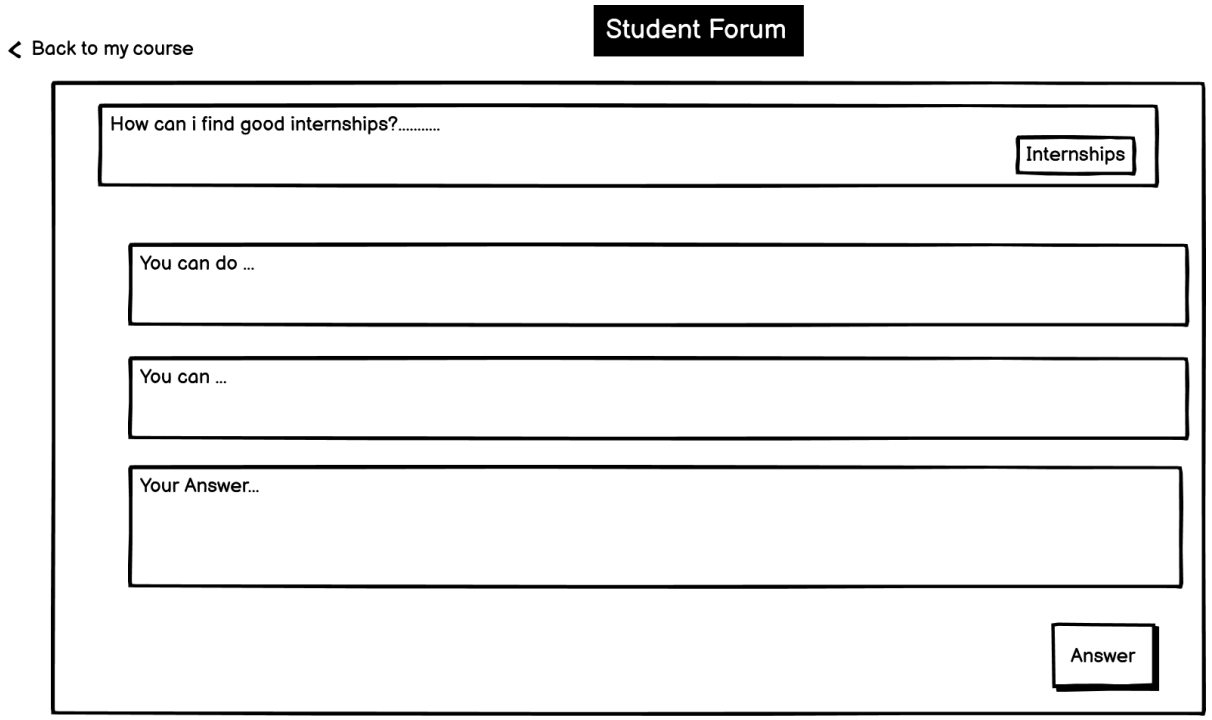

Figure 35 - Screen Mock-Ups - Show Post Page

This page shows the question of a forum post and its answers. User can write his/her answer using the text field and post it by clicking the Answer button.

3.6.2.21 Artifact Review Page for Students

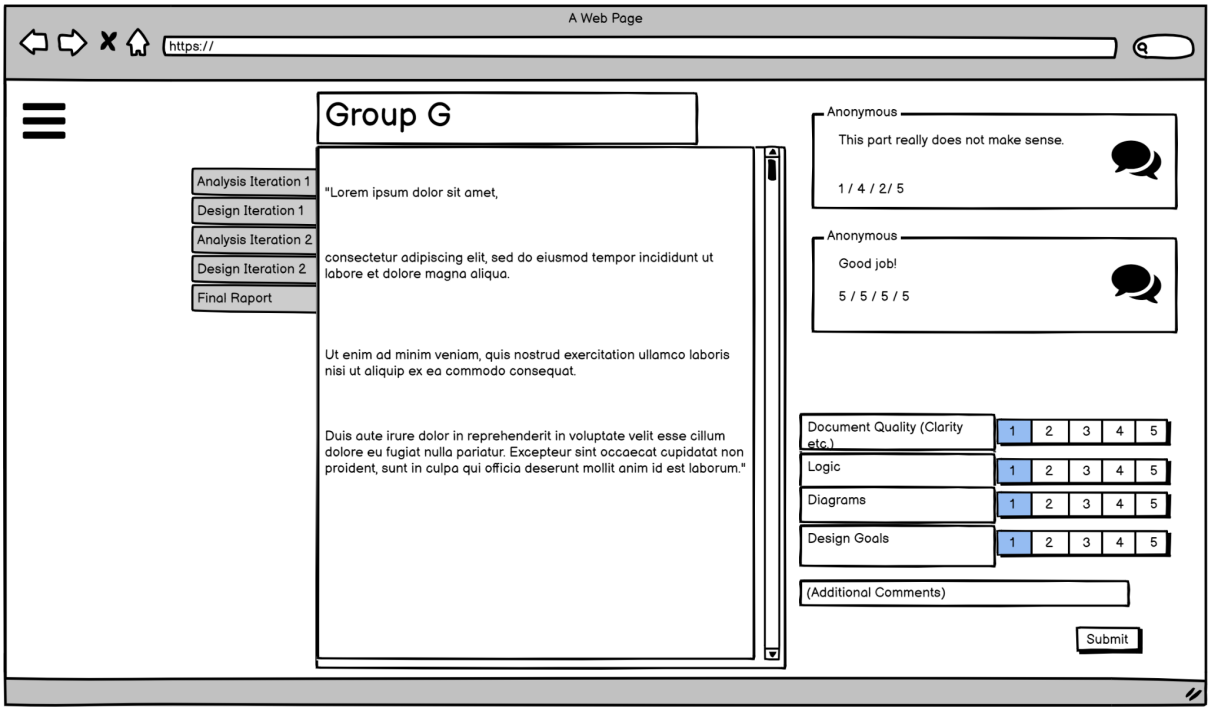

Figure 36 - Screen Mock-Ups - Artifact Review Page for Students

4. Improvement Summary

- Added a sequence diagram for review of the artifacts (Figure 7).
- Activity diagrams are updated.
- Added two state diagrams for course project and assignment (Figure 12 and Figure 13).
- Added grading criteria, Artifact, and Assignment classes to class diagram.
- Use case diagram is updated.
- Sequence diagrams updated.
- Added a new mockup for an artifact review page for students (Figure 36).

5. Glossary & References

J. Pearce, “Client-Server Architectures,” *San José State University*. [Online]. Available: http://www.cs.sjsu.edu/~pearce/oom/ood/distArch/server. [Accessed: 25-Apr-2021].

UI Designs are inspired by these web sites (login required to see applications):

https://mlm.pearson.com/northamerica/masteringphysics/

https://moodle.bilkent.edu.tr/2020-2021-spring/login/index.php

Designs made through Balsamiq Wireframes:

https://balsamiq.com/