İhsan Doğramacı Bilkent University

Department of Computer Engineering

CS319 Object Oriented Software Engineering

ProCheck

Analysis Report Iteration 2

Group 1E

Lara Fenercioğlu - 21802536

Kimya Ghasemlou - 21801412

Sebahattin Utku Sezer - 21802798

Gökhan Taş - 21802136

Bedirhan Sakinoğlu - 21802576

Instructor: Eray Tüzün

Teaching Assistant(s): Elgun Jabrayilzade, Erdem Tuna

Table of Contents

1. Introduction

There has always been a miscommunication between instructors and students during the project development process. It is hard for instructors to organize the project deadlines and project requirements and inform the students about the changes at the same time, and it is hard for students to follow deadlines and their projects' progression. Moreover, it is always difficult to arrange meetings among team members and this situation leads to difficulties while developing a project. Also, lack of knowledge about the other groups' work prevents groups from working on creative and various ideas. Considering all these drawbacks, we came up with a solution. Classroom helper is a web-based application that students and instructors will use for university projects. The main goal is to help students to form their project groups, organize their project meetings, give and get feedback from both instructors and students. All these features will eventually ease the process of the project.

2. Overview

2.1. Register - Login Screen

Before using our application, users should register according to their roles. There will be two roles: Students and Instructor/TA. Having registered the system, users will log in with their emails and passwords then, they will be directed to the dashboard which will be the main page for the users.

2.2. Dashboard

After the instructor registered, s/he can create a class with an access code for the student to join the class. Students will use this code and their dashboard will be refreshed according to the given code. Students' and instructors' dashboards will be slightly different from each other since both of them are not allowed to use some functions such as giving feedback or group formations. In the dashboard, there will be several functionalities which are written below.

2.2.1. Calendar

Each group will have a calendar apart from their personal calendar where they can see each upcoming artifact deadline and these deadlines will be assigned to each group's calendar by the instructor. Also, groups will be able to edit their calendar and view the calendar. Group members can add their available times in order to arrange a meeting easily together.

2.2.2. Progress Bar

Progress bar module at students' dashboards informs students about their progress about the project according to their overall submissions. Each time students upload a submission to the system, the progress bar will be updated according to the submission's weight to the overall project.

2.2.3. Upcoming Deadlines

In every Student's dashboard, each student can see the deadlines of their upcoming assignments. If their assignments are open for submission, Students can click the link under that assignment which will lead them to the submission page. If there is one day (24 hours) left for a submission, the module will warn the students with a color change on that submission.

Every Instructor and TA can modify the “Upcoming Deadlines” module as they wish. Instructors and TA’s can add a deadline for submission to the module, delete any deadline for submissions, or arrange a submission’s deadline and name.

2.2.4. Artifact Review

At the end of each artifact submission (analysis report, design report etc.) each group must give reviews to 2 groups. These 2 groups will be assigned randomly at the end of each submission. Also, these reviews will be made anonymously and groups will not know which group they are reviewing. These reviews will be made on a template question form. This will be mandatory for every group. There will be two buttons that represent whether reviews are done or not. After the group gives a review to the anonymous group, these buttons will be checked. Under these buttons, there will be links that make the groups anonymous and able to review these artifacts personally. This functionality will allow each group to get feedback from their classmates so that they can improve their ideas.

2.2.5. Submit Assignment and Instructor Feedback

Instructors will create submission buttons and determine each artifact’s deadline at the beginning of the semester and groups will see each assignment on their upcoming deadlines part on the dashboard. They will be clickable links and one student from each group will add artifact submission through these links. It is enough for one group member to upload the assignment. After one loads the assignment, all the other group members will also be seen as if they uploaded as well. After each submission, instructors and ta’s will start giving feedback on submission pages. This feedback will have a follow-up discussion so that the group members can discuss the feedback with instructors and TA’s.

2.2.6. Peer Review

After completing all parts of the projects the instructor will activate the peer review option for groups to assess their teammates about their performance during the whole semester. These reviews will be made on a template question form. This feature will only be used by the student.

2.2.7. Statistics About the Course

Students can follow the statistics such as the average and the curve of each assignment from their dashboard’s statistics module. It will also help the instructor with handling the calculation of curve and average just after entering all of the grades on their own and making it available for the students.

2.2.8. Group Formation

Group formation can be performed by clicking on one of the members of the class which doesn’t have a group and sending an invitation to the person to join the group or the students which don’t have a group can send a request to join a group. The instructor will determine a group formation deadline and after this deadline those who don’t have a group will be assigned to groups randomly. This feature will only be used by the student.

2.2.9. Chat

In the right side of the dashboard there will be a sliding bar which includes chat functions. There will be options such as creating channels for group communications, joining channels and one-to-one chatting for all members of the class. These options can be used for group formation, announcements from teacher to students and ask course related questions from teachers and TA's.

3. Functional Requirements

- **Log In**

  The first thing that the user will see is the Register - Login screen. If the user has registered already, s/he can enter the program by entering the username and password.

- **Register**

  In case of not having an account before, the user can click Register and create a new account by entering the information and the user type (Instructor/Student).

- **Create Class**

  The instructor can create a new class after logging in to the application in case of not having one already. After the creation of the new class, a random unique key will be generated so the instructor can provide the key for the students to join the class.

- **Artifact Review**

  Artifact review is one of the main themes in the student helper. Students will be able to give artifact reviews to other groups for further improvements. By default the number of required artifact reviews that each group should give is two. At the end of the artifact review process, students will be able to see artifact reviews given to them and instructors will be able to see each group's own artifact review made by the other groups.

- **Group Formation**

  The basis of the student helper program is on team work. The instructor must set the preferred group size at the beginning of the semester. Students, after joining the class, need to form a group. After the formation of a group any group member can send an invitation to a student that doesn’t have a group yet and the student that doesn’t have a group yet can accept the invitation or send a request to any group to join them, after the acceptance from the group the student will be a part of the group. At the end of group formation deadline, if any students left without a group the program will assign them to groups that are not filled yet.

- **Peer Review**

  At the end of the project, each member of a group must review the others in the group. This will be done by a template scoring form and there will be also written questions for members to share their thoughts about a specific member. This will help the instructor to understand and evaluate which members do participate in the group work and which don’t.

- **Edit Project Settings**

  Instructors will be able to change the project settings such as how many members should be in a group or what are the important deadlines for the project.

- **Give Feedback & Grade**

  After a group adds a submission, then the instructor can give feedback in order to point out what is missing, wrong or correct. The instructor can also grade that submission.

- **Create Assignment**

  Only instructors can create an assignment so that the students of that class can be informed about their assigned tasks. After creating an assignment, the instructor can edit the assignment.

- **Add Submission**

  Only students can add submissions. One member from each group must upload an artifact before the deadline of that submission, after the submission from one of the group participants, it will be seen as submitted to all group members.

- **Announce**

  Only instructors can announce. These announcements can include the deadlines of the artifacts or some comments related to the project.

- **Chat**

  Users can send messages to the participants of the class privately.There will be a specific channel for each group in the class. Students will also have the chance to talk with the group members that s/he belongs to, in their group channel. Additionally, Students will be in the general chat where they can ask a question to instructors publicly in the FAQ channel or in the main channel.

4. Non-Functional Requirements

Usability

The entering interface of the application has playful icons which help users to have motivation to study. Additionally, the user interface is based on purple color and according to conducted research, purple color has a relaxing nature and is the color of peace. Since our application is focused on group work, this color preference will give positive energy to have a peaceful work environment. Also, every functionality can be accessed through the main dashboard of each user. This will provide easy access to the functionalities of the program. Furthermore, users can see the statistics of the class, students can see their progress in the project and these are all features that help users to follow the track of their work.

Availability

Most of the functionalities in the application are the requirements for the other functionalities thus not all features will be available to users whenever they want. They have to complete some tasks in order to access other functionalities. Additionally, not all functionalities will be available until the instructor opens such as assignment submission. This will create a dependency between functionalities and users won’t be able to access some operations when wanted for use.

Response Time

Dashboards will be shown in less than 0.5 seconds. All functionalities will work in this short period of time in order to increase user experience. Since our application will be based on a database, the queries we handle will fasten the performance.

Reliability

Users’ data will be stored privately. Encapsulation will be used in order to implement an information hiding mechanism to increase security. Additionally, there will be exceptional cases for wrong password, invalid submission type etc. These exceptions will be checked in the presence of an invalid input.

Supportability

The code must be clean and there must be comments that define each feature of the code so that other developers can understand the code easily and can implement further features. Also, by considering the UML diagrams, the flow and the implementation of the program can be understood. Additionally, this project will be developed on Github so the communication between the developers will be held easily. In case of an emergence of a bug, any contributor can create a new issue to fix the problem.

5. System Models

5.1. Use Case Models

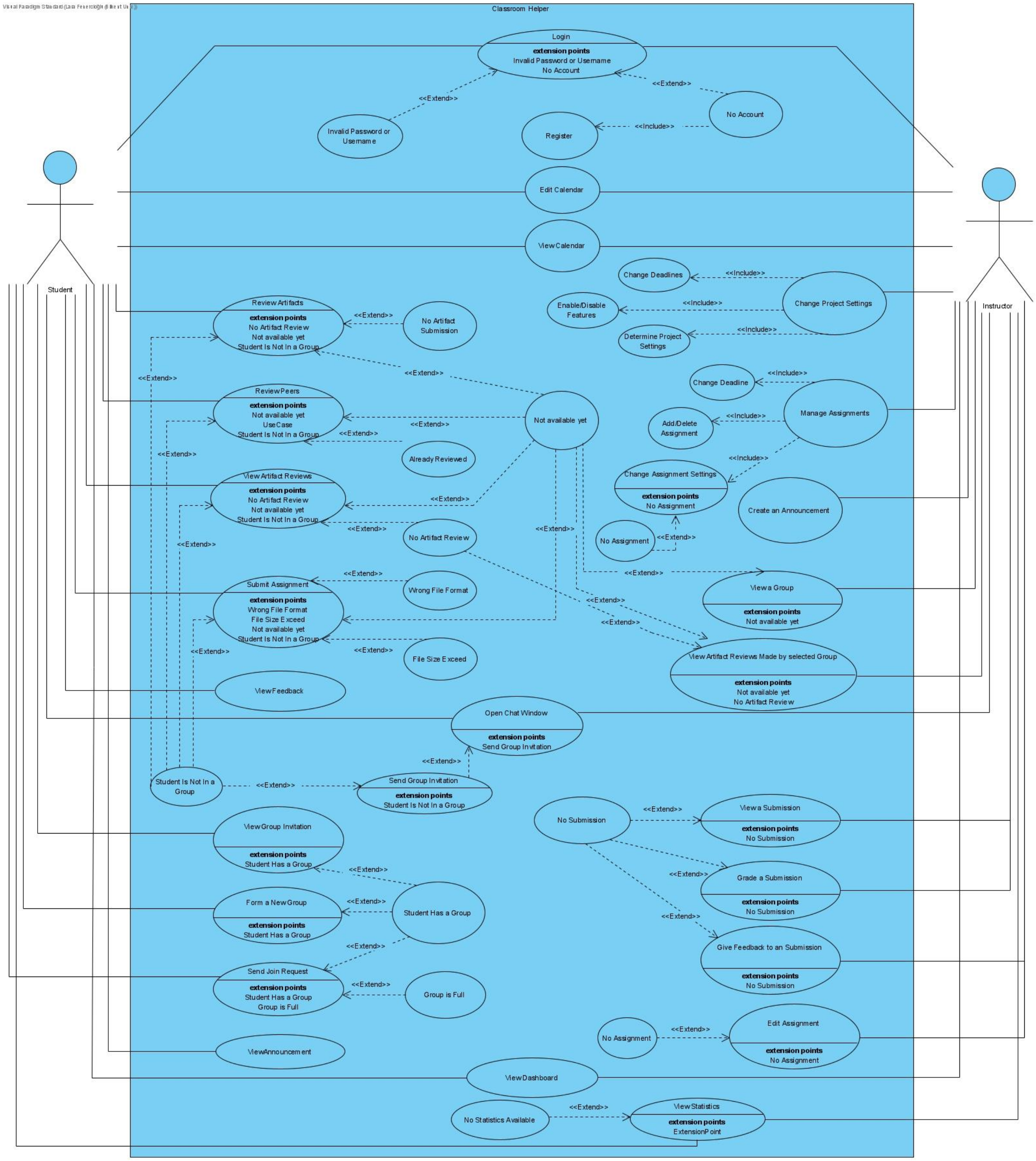


Figure 1 https://imgur.com/yMvWu6G

Textual Use Case Description

Name: Login

Participating Actors: Student, Instructor/TA

Entry Condition: Open the app

Exit Condition:

- Actor successfully logs in
- Actor does not have an account, proceeds to Register

Flow of Events:

1. Actor enters ID
2. Actor enters password
3. Actor presses “LogIn” button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet

Name: Invalid Password or Username

Participating Actors: Student, Instructor/TA

Entry Condition: Actor enters an invalid username or password at the Login screen

Exit Condition: Actor returns to login screen after notification

Flow of Events:

1. Actor enters incorrect ID or password
2. Actor clicks on “LogIn” button
3. A notification appears

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User enters a username or password which doesn’t correspond to any user in the database

---

Name: No Account

Participating Actors: Student, Instructor/TA

Entry Condition: Actor should not have an account

Exit Condition: Actor creates new account

Flow of Events:

1. Actor opens the app
2. Actor does not have an account

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User doesn’t has an account yet

Name: Register

Participating Actors: Student, Instructor/TA

Entry Condition: Actor should not have an account

Exit Condition: Actor registers to the system

Flow of Events:

1. Actor does not have an account and tries to open an account
2. Actor creates a new account with a username and password.

Special/Quality Requirements:

- Server must be up

- User must be connected to the internet
- User doesn’t has an account yet and clicks on register

Name: Edit Calendar

Participating Actors: Student, Instructor/TA

Entry Condition: Actor clicks to edit calendar button on the dashboard

Exit Condition: Actor clicks the previous page button

Flow of Events:

1. Actor clicks the edit calendar button
2. Calendar will open in a new page in a bigger format
3. Actor can add edit his/her own calendar
4. Actor presses the previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User must be registered and signed in to the system
- User must click on calendar to edit calendar

Name: View Calendar

Participating Actors: Student, Instructor/TA

Entry Condition: Actor clicks to view calendar button on the dashboard

Exit Condition: Actor clicks the previous page button

Flow of Events:

1. Actor clicks on the button
2. Calendar will appear as pop-up

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User must be registered and signed in to the system
- User must click on calendar to edit calendar

Name: Review Artifacts

Participating Actors: Student

Entry Condition: Actor clicks to the “Review Artifact” button

Exit Condition:

- Actor completes review
- Actor clicks submit button

Flow of Events:

1. Actor clicks on review artifact button
2. Artifact submission of a random group and a poll for reviewing artifact comes up
3. Actor completes the review
4. Actor clicks submit button

Special/Quality Requirements:

- Actor must be in a group
- the artifact must be submitted
- User must be registered and signed in to the system
- User must be connected to the internet
- Server must be up
- Artifact Review must be enabled by the Instructor/TA

Name: No Artifact Submission

Participating Actors: Student

Entry Condition: Actor will review an artifact but there is no submission yet

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor clicks review artifacts
2. Actor sees that group did not upload any assignment to review
3. Actor presses the go to the previous page button

Special/Quality Requirements:

- Actor must be in a group
- the artifact is not submitted
- User must be registered and signed in to the system
- User must be connected to the internet
- Server must be up
- Artifact Review must be enabled by the Instructor/TA

Name: Review Peers

Participating Actors: Student

Entry Condition: Actor clicks the “Click for Peer Review” button at Student dashboard

Exit Condition: Actor is done with the review and submits it

Flow of Events:

1. Actor clicks the “Click for Peer Review” button for desired peer
2. Actor enters the grade for selected Student
3. Actor enters the feedback for selected Student
4. Actor submits the review after filling it

Special/Quality Requirements:

- Actor must be in a group
- User must be registered and signed in to the system

- Server must be up
- User must be connected to the internet
- Peer review must be enabled by the Instructor/TA

Name: Not Available Yet

Participating Actors: Student, Instructor/TA

Entry Condition:

- Student tries to review artifact when there is no available artifact
- Student tries to review peers when there is no available review
- Student tries to view artifact when there is no available artifact
- Student tries to submit assignment when it is not opened yet
- Instructor/TA tries to view a group when there is no group yet
- Instructor/TA tries to view artifact review made by selected group when there is no available review

Exit Condition: Actor presses the close button

Flow of Events:

1. Actor tries to do an action which is not available yet
2. Error message is displayed
3. Actor clicks the close button

Special/Quality Requirements:

- User must be registered and signed in to the system
- User must be connected to the internet
- Server must be up
- The functionality is not enabled by the Instructor/TA

Name: Already Reviewed

Participating Actors: Student

Entry Condition: If the peer has already been reviewed

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor clicks review peer button
2. Actor sees that this peer has already been reviewed
3. Actor presses previous page button

Special/Quality Requirements:

- User must be registered and signed in to the system
- Server must be up
- User must be connected to the internet
- User must be in a group
- The artifact has been reviewed already

Name: View Artifact Reviews

Participating Actors: Student

Entry Condition: Actor clicks on View Artifact Button in Dashboard

Exit Condition: Actor clicks on previous page button or ok button

Flow of Events:

1. Actor clicks on View Artifact Reviews button
2. Actor selects the artifact review s/he wants to view
3. Actor clicks on exit or previous page button

Special/Quality Requirements:

- Actor must be in a group
- Server must be up

- User must be registered and signed in
- Artifact reviews for the assignment must be done
- Artifact reviews must be enabled by the Instructor/TA

Name: No Artifact Review

Participating Actors: Student, Instructor/TA

Entry Condition: If the desired artifact has no review so far

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor clicks the button to see the artifact reviews
2. Actor sees that artifact has no review so far
3. Actor presses the previous button to exit

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User must be registered and signed in
- Artifact review must be enabled by the instructor
- No artifact reviews has been done to the assignment

Name: Submit Assignment

Participating Actors: Student

Entry Condition: Actor clicks on add submission button

Exit Condition: Submission completed

Flow of Events:

1. Actor enters the submission page
2. Actor uploads the assignment from his/her computer

3. Actor presses the previous page when submission completes

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- There must be an assignment created by instructor
- Student clicks on submit assignment

Name: Wrong File Format

Participating Actors: Student

Entry Condition: Actor attempts to upload a file in wrong format

Exit Condition: Notification shows up

Flow of Events:

1. Actor attempts to upload an assignment
2. Uploaded file’s format is different from what Instructor/TA determined
3. Notification shows up

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in to the system
- Actor must be in a group
- Assignment upload must be enabled by Instructor/TA
- Student uploads a file with a wrong format

Name: View Feedbacks

Participating Actors: Student

Entry Condition: Actor clicks the button to see the feedbacks

Exit Condition: Actor clicks the previous button to exit

Flow of Events:

1. Actor clicks the view feedback button
2. Actor sees the feedback
3. Actor clicks the previous button to exit

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User must be registered and signed in to the system
- User clicks on view feedback

Name: View Announcements

Participating Actors: Student

Entry Condition: Actor clicks the button to see the announcements

Exit Condition: Actor clicks the previous button to exit

Flow of Events:

4. Actor clicks the view announcement button
5. Actor sees the announcement
6. Actor clicks the previous button to exit

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- User must be registered and signed in to the system
- User clicks on announcement button

Name: File Size Exceed

Participating Actors: Student

Entry Condition: Actor attends to upload a file which exceeds the size limit determined by Instructor/TA

Exit Condition: Notification shows up

Flow of Events:

1. Actor attempts to upload an assignment
2. Uploaded file's size is over the limit determined by Instructor/TA
3. Notification shows up

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in
- Actor must be in a group
- Assignment upload must be enabled by Instructor/TA
- Actor submits a file with a size over the limit determined by Instructor/TA

Name: Open Chat Window

Participating Actors: Student, Instructor/TA

Entry Condition: Actor tries to open chat window

Exit Condition: Actor closes the chat window

Flow of Events:

1. Actor opens the chat window
2. After s/he is done, actor closes the chat window

Special/Quality Requirements:

- Server must be up

- User must be connected to the internet
- Actor must be registered and signed in to the system

Name: Create An Announcement

Participating Actors: Instructor/TA

Entry Condition: Actor presses the create announcement button

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor adds a new announcement
2. Actor publishes this announcement
3. Actor finishes the creation and presses previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Manage Assignments

Participating Actors: Instructor/TA

Entry Condition: Actor attends to edit or add an assignment

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor presses manage assignments button
2. Actor can either add a new assignment submission or edit an existing one
3. After editing or adding, Instructor/TA presses previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Change Project Settings

Participating Actors: Instructor/TA

Entry Condition: Actor clicks to button to change project settings

Exit Condition: Actor is done with changing the settings and saves them

Flow of Events:

1. Actor clicks to button to change project settings
2. Project settings screen opens up
3. Actor can change the following settings:
    a. Change peer review function's visibility by clicking enable or disable button
    b. Change group formation function's visibility by clicking enable or disable button
    c. Change the weight of the specified assignment
    d. Change the deadline of the specified assignment

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Change Deadlines

Participating Actors: Instructor/TA

Entry Condition: Actor attempts to change a deadline

Exit Condition: Actor finishes the change and presses the previous page button

Flow of Events:

1. Actor enters this use case from change project settings
2. Actor changes the beginning and ending deadline of the project
3. Actor finishes the change and returns to the previous page

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Enable/Disable Features

Participating Actors: Instructor/TA

Entry Condition: Actor clicks to Enable/Disable Features button

Exit Condition: Actors saves the changes by clicking save button

Flow of Events:

1. Actor clicks to Enable/Disable Features button
2. Actor can click “Enable” or “Disable” buttons to change these features:
    a. Peer Review
    b. Group Formation
    c. Artifact Review

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Determine Project Settings

Participating Actors: Instructor/TA

Entry Condition: The actor clicks on the project settings button

Exit Condition: The actor clicks on the exit button

Flow of Events:

1. The actor clicks on the project setting button
2. The actor chooses one of the options such as changing the deadline and adding delete assignment.

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Change Deadline

Participating Actors: Instructor/TA

Entry Condition: The actor clicks on change project settings and then clicks on change deadline button

Exit Condition: The actor clicks on the apply button and goes back to the dashboard

Flow of Events:

1. The actor clicks on the change project setting
2. The actor clicks on the change deadline button
3. The actor sets the new deadline
4. The actor clicks on apply changes button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet

- Actor must be registered and signed in as the instructor

Name: Add/Delete Assignment

Participating Actors: Instructor/TA

Entry Condition: Actor clicks on add or delete assignment

Exit Condition: Addition or deletion is completed

Flow of Events:

1. Actor clicks on add or delete assignment
2. Actor selects add option or delete option for an assignment

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Change Assignment Settings

Participating Actors: Instructor/TA

Entry Condition: Actor wants to manage assignments

Exit Condition: Actor exits from the manage assignments windows

Flow of Events:

1. Actor clicks on the manage assignment
2. Actor changes the assignment setting.
3. Actor clicks exit button to return back to the dashboard

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: View A Group

Participating Actors: Instructor/TA

Entry Condition: Actor wants to review a group

Exit Condition: Actor clicks on the exit button

Flow of Events:

1. Actor clicks on a group to review their submission etc.
2. After the Actor is done, they click the exit button to return the dashboard.

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: View Artifact Reviews Made By Selected Group

Participating Actors: Instructor/TA

Entry Condition: Actor wants to see a review made by a group

Exit Condition: Actor press the exit button

Flow of Events:

1. Actor presses on a group to see their reviews and their reviewed artifacts
2. After Actor is done, they exits from the review window

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: No Assignment

Participating Actors: Instructor/TA

Entry Condition: If there is no assignment made by the instructor, then the instructor cannot edit an assignment.

Exit Condition: Actor clicks previous page button

Flow of Events:

1. Actor tries to change assignment settings
2. Actor gets an error message since there is no assignment to change
3. Actor press the previous page button to exit

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: View A Submission

Participating Actors: Instructor/TA

Entry Condition: The instructor clicks on the view button in the submission part to see the submission made by a specific group.

Exit Condition: Actor clicks on exit button or clicks on previous page button

Flow of Events:

1. Actor Clicks on view button
2. Submission is displayed on screen

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

- There must be an uploaded submission

Name: No Submission

Participating Actors: Instructor/TA

Entry Condition: If a submission does not exist then the actor cannot view, grade or give feedback to a submission.

Exit Condition: Actor presses the previous page button

Flow of Events:

1. Actor attends to view, grade or give a feedback to a submission
2. Actor sees that there is no submission available to view
3. Actor presses previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor

Name: Grade A Submission

Participating Actors: Instructor/TA

Entry Condition: Actor clicks on the grade button for an assignment to submit the grade for it.

Exit Condition: Grading completed

Flow of Events:

1. Actor attends to grade one of the group's submission.
2. After reading and checking the submission, the Actor loads a grade for that submission.
3. Actor finishes grading for one group and presses the previous page button.

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor
- There must be an uploaded submission

Name: Give Feedback To A Submission

Participating Actors: Instructor/TA

Entry Condition: The instructor clicks on the give feedback button for an assignment.

Exit Condition: Actor finishes the feedback process and presses the previous page button.

Flow of Events:

1. Actor attends to give a feedback to one of the group's submission
2. Actor views the assignment and reads
3. Actor writes a feedback for the group and publishes for that group to see
4. Actor finishes giving feedback and presses previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as the instructor
- There should be a submission

Name: Student Is Not In A Group

Participating Actors: Student

Entry Condition: If the student is not in a group he/she cannot

- Submit an assignment
- View artifact reviews
- Review peers

- Review artifacts.

Exit Condition: Student press the close button

Flow of Events:

1. Students tries to do an invalid action
2. Students sees error message in the entered action window
3. Student closes the window to exit

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student
- The group formation deadline should be over
- The Actor must not be registered to any group

Name: Send Group Invitation

Participating Actors: Student

Entry Condition: Actor clicks on a student who doesn't have a group by looking from the chat and chooses to send a group invitation.

Exit Condition: Invitation completed and pressed the previous page button

Flow of Events:

1. Actor enters chat
2. Actor finds another student that does not have a group currently
3. Actor sends an invitation for the student to join their group
4. After invitation sent, student pressed previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet

- Actor must be registered and signed in as student
- Student must be in a group
- The receiver of the invitation must not be in a group
- Group should not be full

Name: View Group Invitation

Participating Actors: Student

Entry Condition: Actor click on the view group invitation button

Exit Condition: Actor presses previous page button

Flow of Events:

1. Actor clicks the view button to see the invitation that a group has sent
2. If Actor accepts the invitation, actor automatically joins the group, if declines then Actor continues to be groupless
3. Actor presses the previous page button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student
- The actor must have an invitation sent from a group

Name: Form A New Group

Participating Actors: Student

Entry Condition: Actor doesn’t have a group and clicks on the form a new group

Exit Condition: Actor successfully creates a group

Flow of Events:

1. Actor clicks to form a group button

2. Actor sees that he forms a group

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student
- The student shouldn't be part of a group
- The group formation deadline is not over

Name: Student Has A Group

Participating Actors: Student

Entry Condition: If the Actor has a group he/she cannot do the following:

- View group invitation
- Form a new group
- Send a join request to a group

Exit Condition: Actor closes the page that s/he entered

Flow of Events:

1. Actor tries to do an action which is not valid for her/him
2. Actor sees new window with the message
3. Actor presses the exit button

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student
- The student must be part of a group

Name: Send Join Request

Participating Actors: Student

Entry Condition: Actor clicks on the send join request for a group

Exit Condition: Actor clicks on the send button on the pop-up screen to send the request

Flow of Events:

1. Actor clicks on the send join request button from dashboard
2. Actor can fill a note to be sent to the group
3. Actor clicks on the send button and returns to the dashboard

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student
- The student shouldn't be part of a group
- The receiver group of the request must not be full

Name: Group Is Full

Participating Actors: Student

Entry Condition: The selected group to send join request is already full

Exit Condition: Display Notification

Flow of Events:

1. Actor sends join request to a group
2. Selected group is already full
3. Display notification

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- Actor must be registered and signed in as student

- The student shouldn’t be part of a group
- The receiver group of the request must be full

Name: View Dashboard

Participating Actors: Student, Instructor/TA

Entry Condition: User clicks on view dashboard button

Exit Condition: User closes the page

Flow of Events:

1. User wants to see the dashboard so clicks on view dashboard
2. User can see the dashboard
3. User closes the page

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet

Name: View Statistics

Participating Actors: Student, Instructor/TA

Entry Condition: User clicks on view statistics button

Exit Condition: User closes the page

Flow of Events:

1. User wants to see the statistics of a course so clicks on view statistics
2. User sees the statistics of a course
3. User is done with looking so closes the page

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet

- There should be some data to display the statistics

Name: No Statistics Available

Participating Actors: Student, Instructor/TA

Entry Condition: User wants to see statistics but there is no available data to show

Exit Condition: User clicks previous page button

Flow of Events:

1. User clicks on view statistics
2. User can't see because there is no statistics to display
3. User gets warning
4. User goes to previous page

Special/Quality Requirements:

- Server must be up
- User must be connected to the internet
- There is no data to display statistics

5.2. Dynamic Models

5.2.1. Sequence Diagrams

Scenario 1: Artifact Review Sequence Diagram

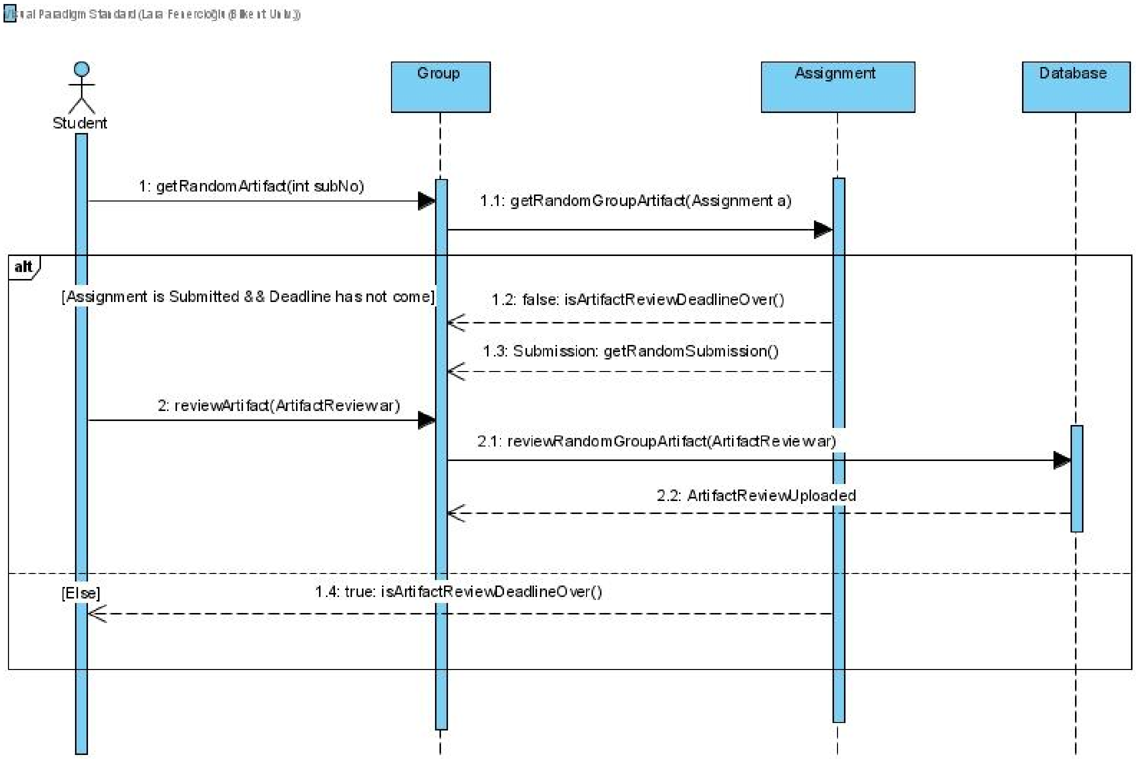

Visual Paradigm Standard(Lara Fenercioglu(Bilkent Univ.))
Student
Group
Assignment
Database
1: getRandomArtifact(int subNo)
1.1: getRandomGroupArtifact(Assignment a)
alt
[Assignment is Submitted && Deadline has not come]
1.2: false: isArtifactReviewDeadlineOver()
1.3: Submission: getRandomSubmission()
2: reviewArtifact(ArtifactReview ar)
2.1: reviewRandomGroupArtifact(ArtifactReview ar)
2.2: ArtifactReviewUploaded
[Else]
1.4: true: isArtifactReviewDeadlineOver()

Figure 2: https://imgur.com/PPwQEZX

Scenario 2: Group Formation with Request Sequence Diagram

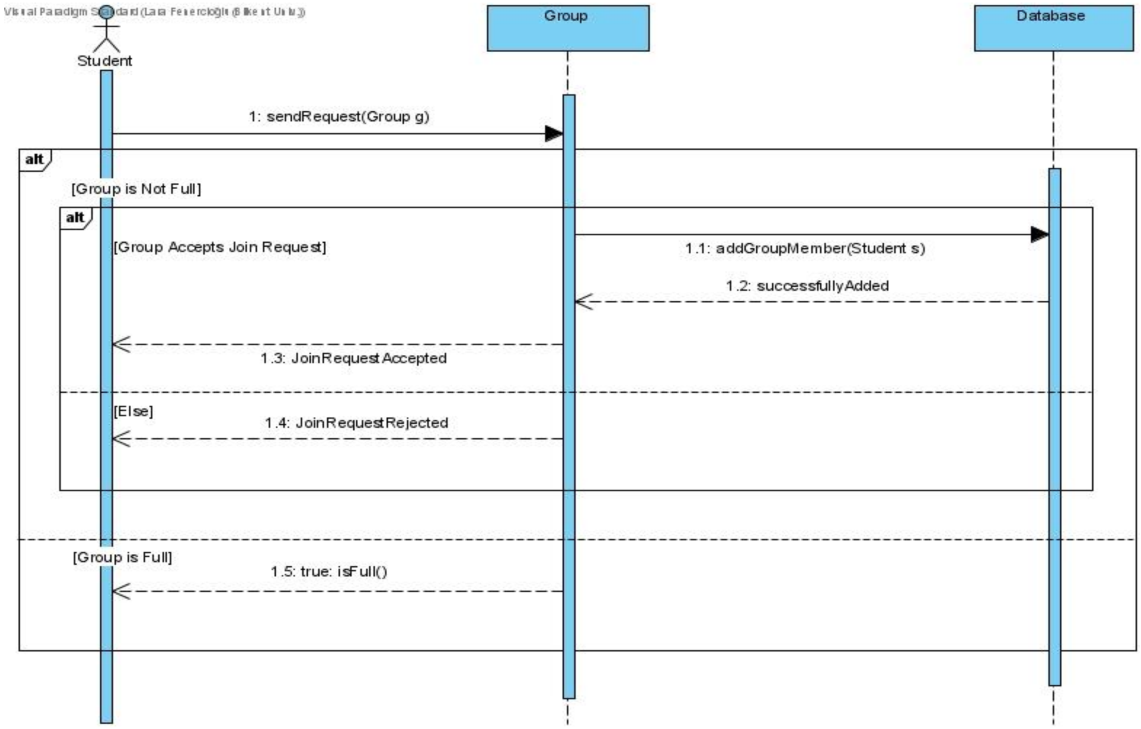


Figure 3: https://imgur.com/N94ayxI

Scenario 3: Group Formation with Invitation Sequence Diagram

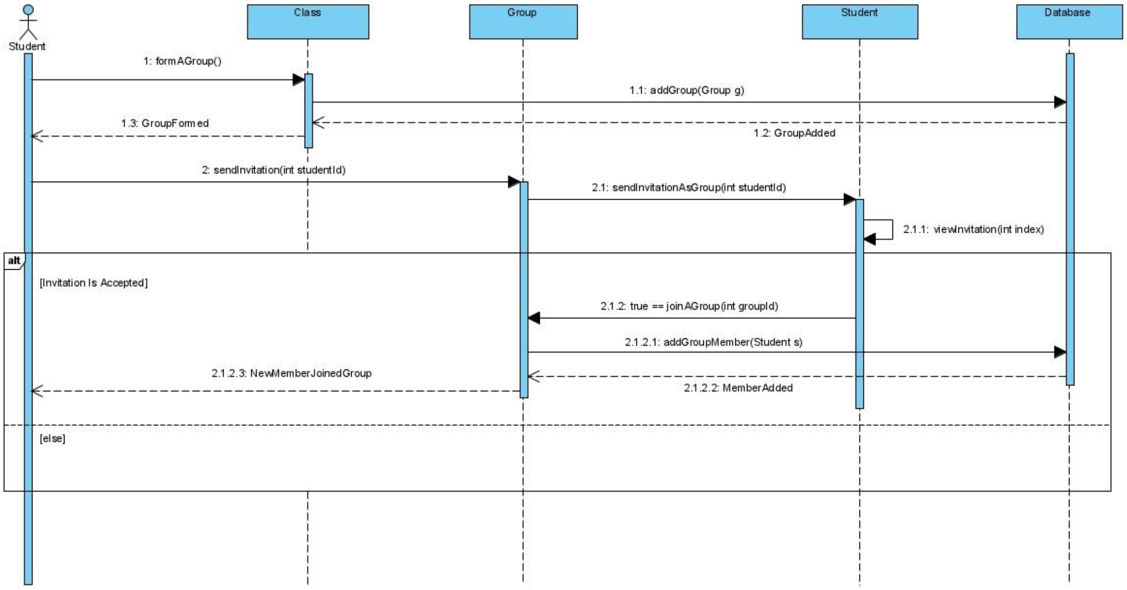

Student
Class
Group
Student
Database
1: formAGroup()
1.1: addGroup(Group g)
1.3: GroupFormed
1.2: GroupAdded
2: sendInvitation(int studentId)
2.1: sendInvitationAsGroup(int studentId)
2.1.1: viewInvitation(int index)
alt
[Invitation Is Accepted]
2.1.2: true == joinAGroup(int groupId)
2.1.2.1: addGroupMember(Student s)
2.1.2.3: NewMemberJoinedGroup
2.1.2.2: MemberAdded
[else]

Figure 4: https://imgur.com/KeFRGJO

Scenario 4: Peer Review Sequence Diagram

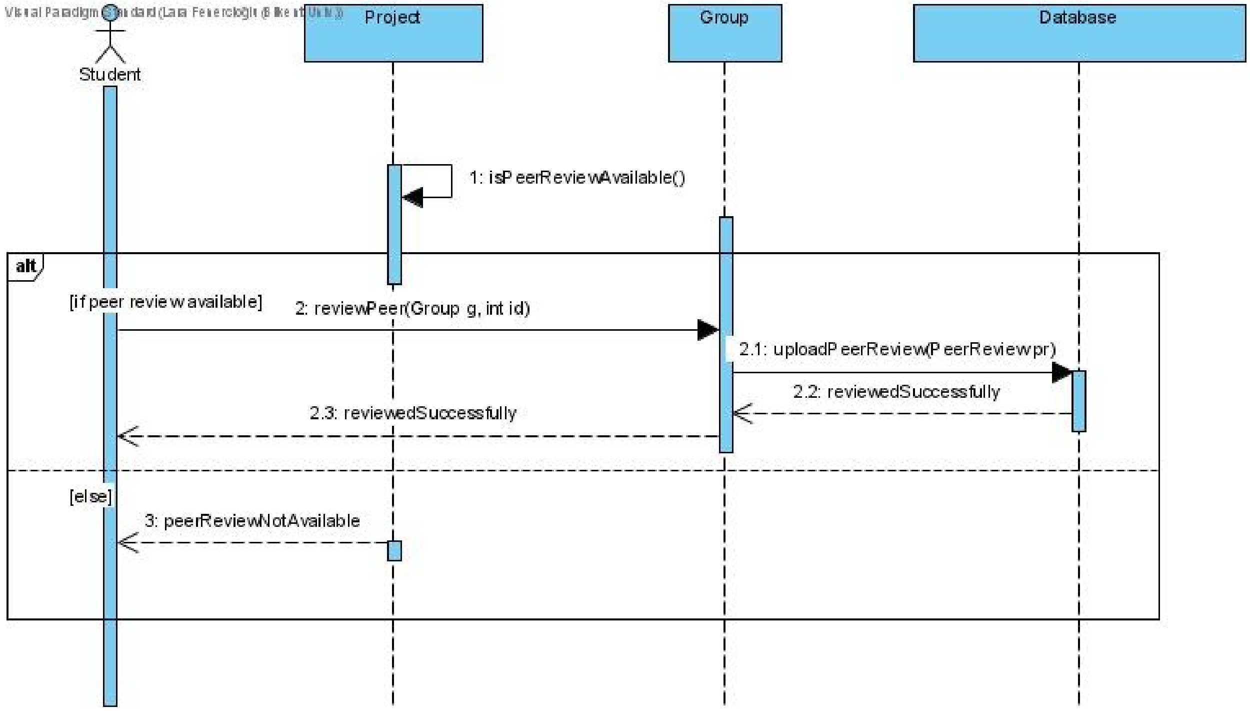

Visual Paradigm Standard (Lara Fenercioğlu (Bilkent Univ.))
Student
Project
Group
Database
1: isPeerReviewAvailable()
alt
[if peer review available]
2: reviewPeer(Group g, int id)
2.1: uploadPeerReview(PeerReview pr)
2.2: reviewedSuccessfully
2.3: reviewedSuccessfully
[else]
3: peerReviewNotAvailable

Figure 5: https://imgur.com/95aBCkb

Scenario 5: Add Submission Sequence Diagram

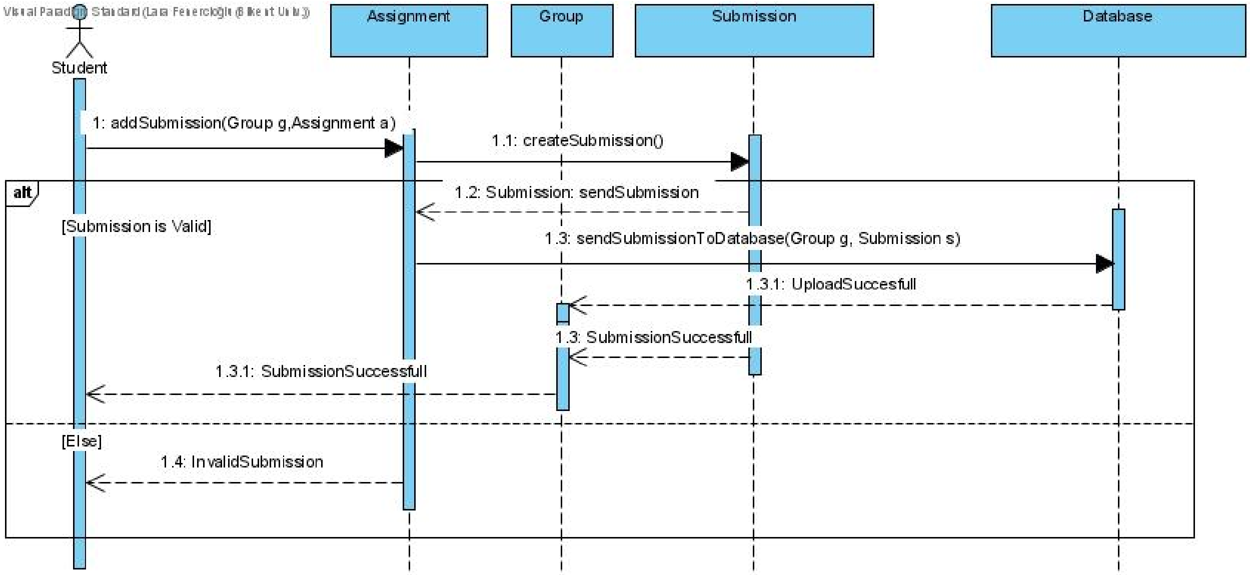

Visual Paradigm Standard (Lara Fenercioğlu (Bilkent Univ.))
Student
Assignment
Group
Submission
Database
1: addSubmission(Group g,Assignment a)
1.1: createSubmission()
alt
1.2: Submission: sendSubmission
[Submission is Valid]
1.3: sendSubmissionToDatabase(Group g, Submission s)
1.3.1: UploadSuccesfull
1.3: SubmissionSuccessfull
1.3.1: SubmissionSuccessfull
[Else]
1.4: InvalidSubmission

Figure 6: https://imgur.com/n4RUkrR

Scenario 6: Give Instructor Feedback Sequence Diagram

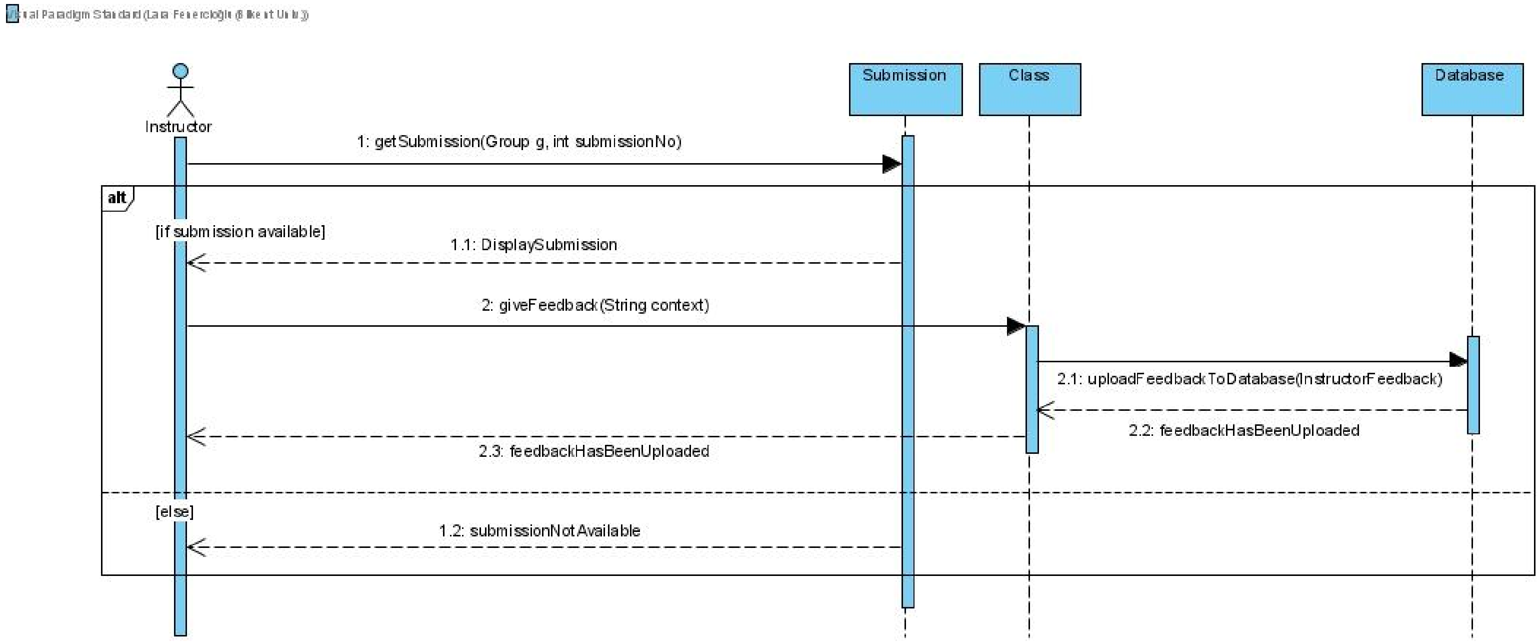

Visual Paradigm Standard (Lara Fenercioglu (Bilkent Univ.))
Instructor
Submission
Class
Database
1: getSubmission(Group g, int submissionNo)
alt
[if submission available]
1.1: DisplaySubmission
2: giveFeedback(String context)
2.1: uploadFeedbackToDatabase(InstructorFeedback)
2.2: feedbackHasBeenUploaded
2.3: feedbackHasBeenUploaded
[else]
1.2: submissionNotAvailable

Figure 7: https://imgur.com/qRVJnmP

5.2.2. Activity Diagram

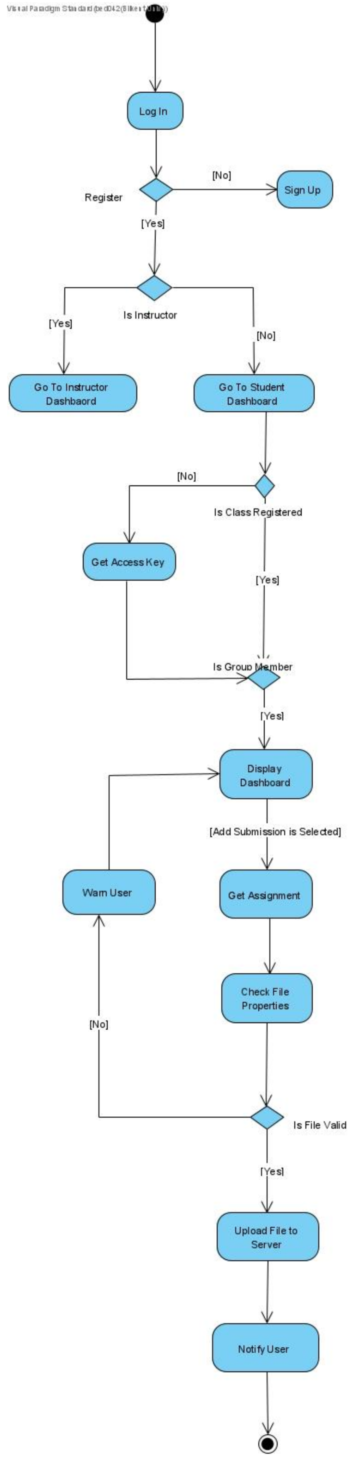


Figure 8: https://imgur.com/FbatR2R

Figure 8 explains the series of action flows that would denote the adding a submission scenario. The initial node starts with the logIn action and if the user has not registered to the application, s/he has to signUp. If a user has registered already, the program will check whether the user is an instructor or a student. If the user is a student type then the program will check whether the user has registered into the class or not. If not, then the student has to enter the class access key which is provided by the instructor. Then the program will check whether the student has a group or not. If the student registered a class before then the program will directly check whether the student has a group or not. After this condition's result, the student will go to the dashboard. However, based on if the condition of having a group or not, the functionalities in the dashboard will differ. If the student has a group then by selecting addSubmission the student will be able to upload an assignment. The system will check whether the submission is appropriate for file properties. If not, the system will warn the user and s/he will be directed to the dashboard. If submission is valid, then the files that are uploaded will be sent to the server and afterwards, the system will notify the user to indicate that the submission has been completed successfully.

5.2.3. State Diagram

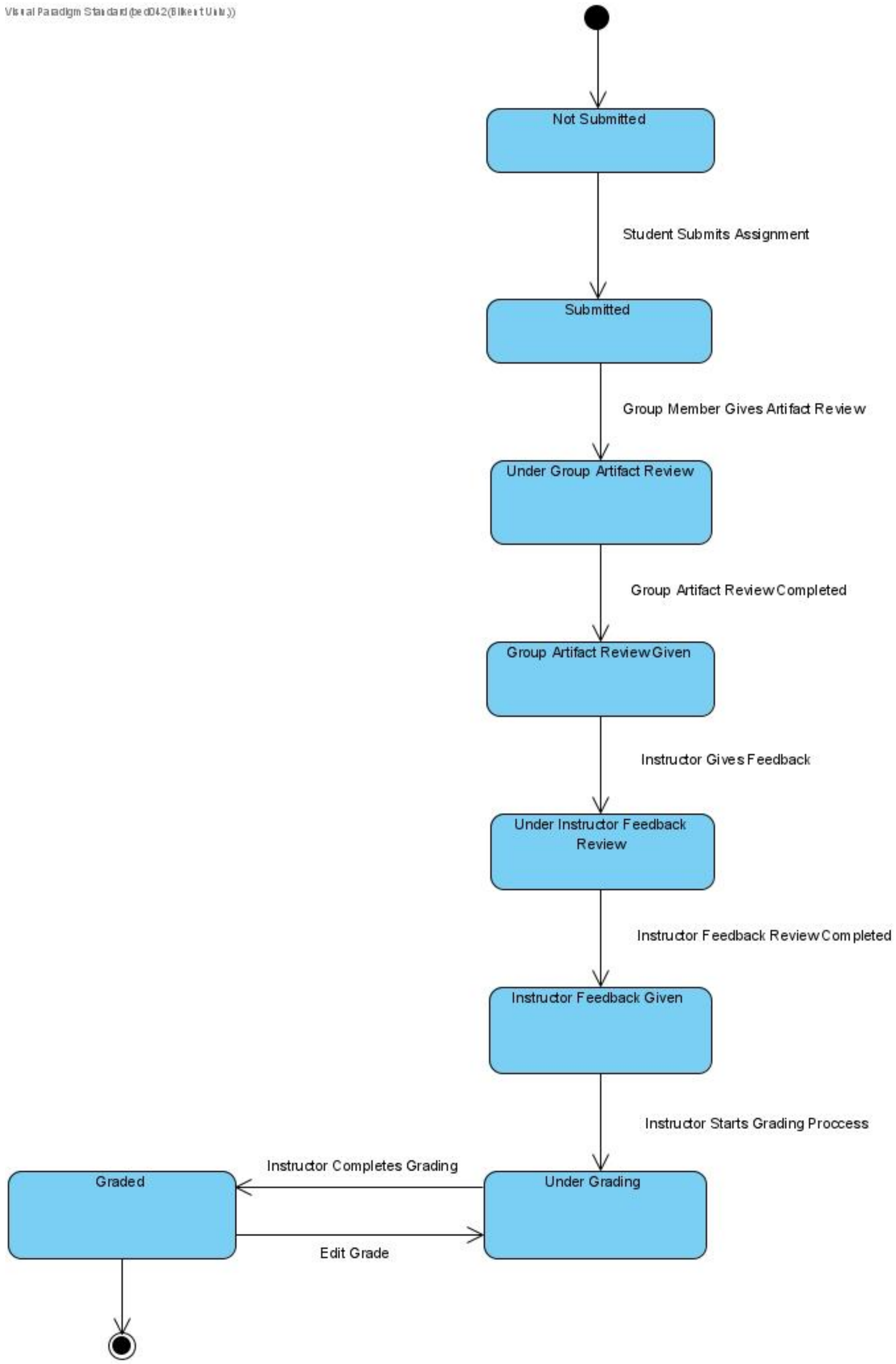

Visual Paradigm Standard(bed042(Bilkent Univ.))
Not Submitted
Student Submits Assignment
Submitted
Group Member Gives Artifact Review
Under Group Artifact Review
Group Artifact Review Completed
Group Artifact Review Given
Instructor Gives Feedback
Under Instructor Feedback Review
Instructor Feedback Review Completed
Instructor Feedback Given
Instructor Starts Grading Process
Under Grading
Instructor Completes Grading
Edit Grade
Graded

Figure 9 https://imgur.com/RWSkGkT

Figure 9 explains what states do throughout the Assignment process. At first the state of the assignment is not submitted. When a student submits an assignment its state will change to submitted state. After this state, when a member from another group wants to do the artifact review to the assignment its state will go to the under group artifact review while it's being reviewed, and when the artifact review is completed it will go to the group artifact review given. Submitting the group artifact review only by one group member is enough. After this state the instructor will be able to give feedback and while he gives the feedback it will be in under instructor feedback review and when the feedback is given it will go to the Instructor feedback given state. The instructor will be able to give a grade after this state, and it will go to the under grading state, when grading completed it will go to the graded state. The instructor will be able to edit the grade and go back to the under grading state.

5.3. Object and Class Models

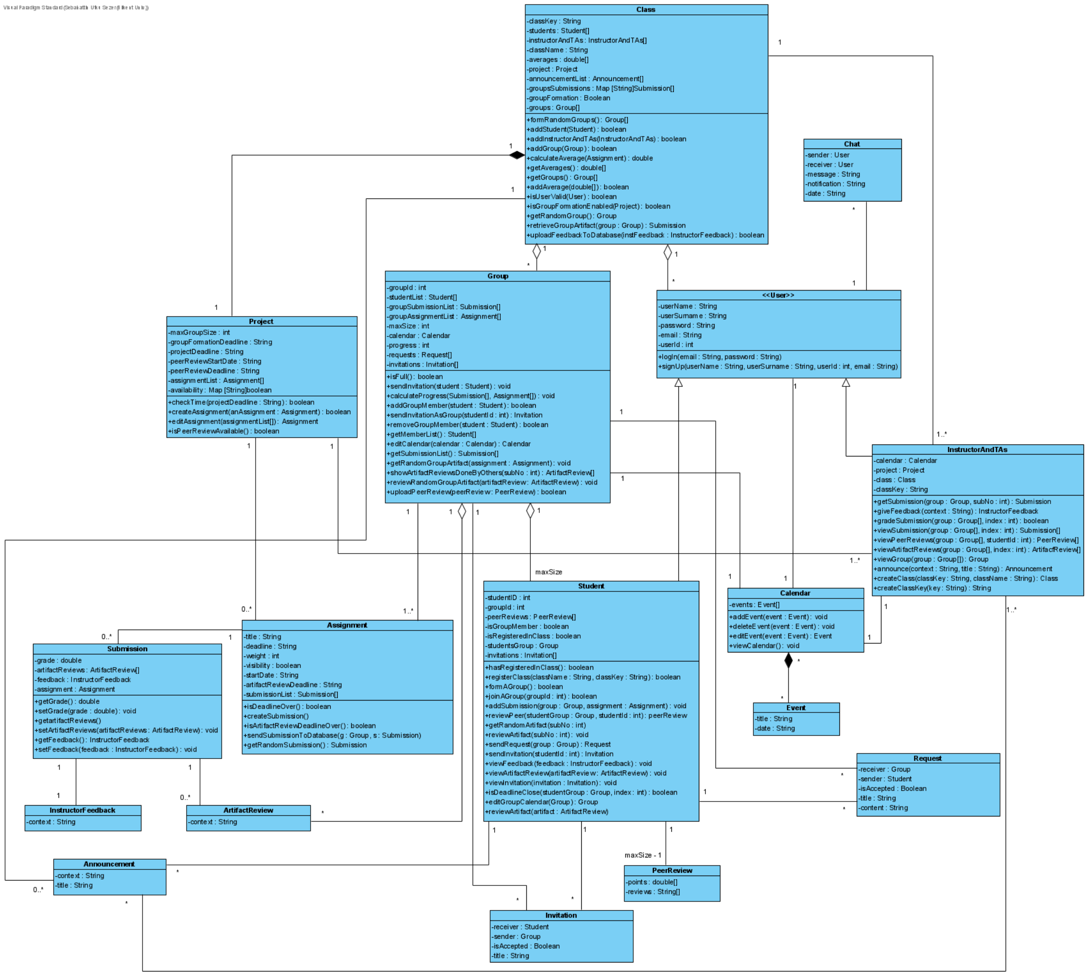

Class
-classKey : String
-students : Student[]
-instructorAndTAs : InstructorAndTAs[]
-className : String
-averages : double[]
-project : Project
-announcementList : Announcement[]
-groupsSubmissions : Map [String]Submission[]
-groupFormation : Boolean
-groups : Group[]
+formRandomGroups() : Group[]
+addStudent(Student) : boolean
+addInstructorAndTAs(InstructorAndTAs) : boolean
+addGroup(Group) : boolean
+calculateAverage(Assignment) : double
+getAverages() : double[]
+getGroups() : Group[]
+addAverage(double[]) : boolean
+isUserValid(User) : boolean
+isGroupFormationEnabled(Project) : boolean
+getRandomGroup() : Group
+retrieveGroupArtifact(group : Group) : Submission
+uploadFeedbackToDatabase(instFeedback : InstructorFeedback) : boolean
Chat
-sender : User
-receiver : User
-message : String
-notification : String
-date : String
Project
-maxGroupSize : int
-groupFormationDeadline : String
-projectDeadline : String
-peerReviewStartDate : String
-peerReviewDeadline : String
-assignmentList : Assignment[]
-availability : Map [String]boolean
+checkTime(projectDeadline : String) : boolean
+createAssignment(anAssignment : Assignment) : boolean
+editAssignment(assignmentList[]) : Assignment
+isPeerReviewAvailable() : boolean
Group
-groupId : int
-studentList : Student[]
-groupSubmissionList : Submission[]
-groupAssignmentList : Assignment[]
-maxSize : int
-calendar : Calendar
-progress : int
-requests : Request[]
-invitations : Invitation[]
+isFull() : boolean
+sendInvitation(student : Student) : void
+calculateProgress(Submission[], Assignment[]) : void
+addGroupMember(student : Student) : boolean
+sendInvitationAsGroup(studentId : int) : Invitation
+removeGroupMember(student : Student) : boolean
+getMemberList() : Student[]
+editCalendar(calendar : Calendar) : Calendar
+getSubmissionList() : Submission[]
+getRandomGroupArtifact(assignment : Assignment) : void
+showArtifactReviewsDoneByOthers(subNo : int) : ArtifactReview[]
+reviewRandomGroupArtifact(artifactReview : ArtifactReview) : void
+uploadPeerReview(peerReview : PeerReview) : boolean
<<User>>
-userName : String
-userSurname : String
-password : String
-email : String
-userId : int
+logIn(email : String, password : String)
+signUp(userName : String, userSurname : String, userId : int, email : String)
InstructorAndTAs
-calendar : Calendar
-project : Project
-class : Class
-classKey : String
+getSubmission(group : Group, subNo : int) : Submission
+giveFeedback(context : String) : InstructorFeedback
+gradeSubmission(group : Group[], index : int) : boolean
+viewSubmission(group : Group[], index : int) : Submission[]
+viewPeerReviews(group : Group[], studentId : int) : PeerReview[]
+viewArtifactReviews(group : Group[], index : int) : ArtifactReview[]
+viewGroup(group : Group[]) : Group
+announce(context : String, title : String) : Announcement
+createClass(classKey : String, className : String) : Class
+createClassKey(key : String) : String
maxSize
Student
-studentID : int
-groupId : int
-peerReviews : PeerReview[]
-isGroupMember : boolean
-isRegisteredInClass : boolean
-studentsGroup : Group
-invitations : Invitation[]
+hasRegisteredInClass() : boolean
+registerClass(className : String, classKey : String) : boolean
+formAGroup() : boolean
+joinAGroup(groupId : int) : boolean
+addSubmission(group : Group, assignment : Assignment) : void
+reviewPeer(studentGroup : Group, studentId : int) : peerReview
+getRandomArtifact(subNo : int)
+reviewArtifact(subNo : int) : void
+sendRequest(group : Group) : Request
+sendInvitation(studentId : int) : Invitation
+viewFeedback(feedback : InstructorFeedback) : void
+viewArtifactReview(artifactReview : ArtifactReview) : void
+viewInvitation(invitation : Invitation) : void
+isDeadlineClose(studentGroup : Group, index : int) : boolean
+editGroupCalendar(Group) : Group
+reviewArtifact(artifact : ArtifactReview)
Calendar
-events : Event[]
+addEvent(event : Event) : void
+deleteEvent(event : Event) : void
+editEvent(event : Event) : Event
+viewCalendar() : void
Event
-title : String
-date : String
Assignment
-title : String
-deadline : String
-weight : int
-visibility : boolean
-startDate : String
-artifactReviewDeadline : String
-submissionList : Submission[]
+isDeadlineOver() : boolean
+createSubmission()
+isArtifactReviewDeadlineOver() : boolean
+sendSubmissionToDatabase(g : Group, s : Submission)
+getRandomSubmission() : Submission
Submission
-grade : double
-artifactReviews : ArtifactReview[]
-feedback : InstructorFeedback
-assignment : Assignment
+getGrade() : double
+setGrade(grade : double) : void
+getartifactReviews()
+setArtifactReviews(artifactReviews : ArtifactReview) : void
+getFeedback() : InstructorFeedback
+setFeedback(feedback : InstructorFeedback) : void
Request
-receiver : Group
-sender : Student
-isAccepted : Boolean
-title : String
-content : String
InstructorFeedback
-context : String
ArtifactReview
-context : String
Announcement
-context : String
-title : String
maxSize - 1
PeerReview
-points : double[]
-reviews : String[]
Invitation
-receiver : Student
-sender : Group
-isAccepted : Boolean
-title : String

Figure 10: https://imgur.com/GQJ3vrf

5.3.1. Explanation

There are two main classes which aim to regulate the whole process, **Class** and **Project** classes. Class object mainly regulates the **Student** and group, it also adds new instructors and students to the class for them to start their semester processes. **Group** class focuses on what students can do when they are in a group. Group can add new members, give feedback to other group's artifacts, calculate the process of the whole project and so on. Project class also has the assignment list which consists of **Assignment** objects. Assignment objects can be regulated from the **InstructorsAndTAs**. Each Assignment object holds **Submission** objects which are basically groups' submissions to that assignment, which then can be seen from the Instructors for assessment and other groups to get an **ArtifactReview**. Instructors can use the **Announcement** class to create an announcement for all students. When a student makes a change on their **Calendar** by adding a new **Event** to it, all the group member's calendar will change and add the same Event into it. Students also can use the **Chat** class to chat with each other and chat as a group.

5.4. User interface - Navigational Paths

Login Screen

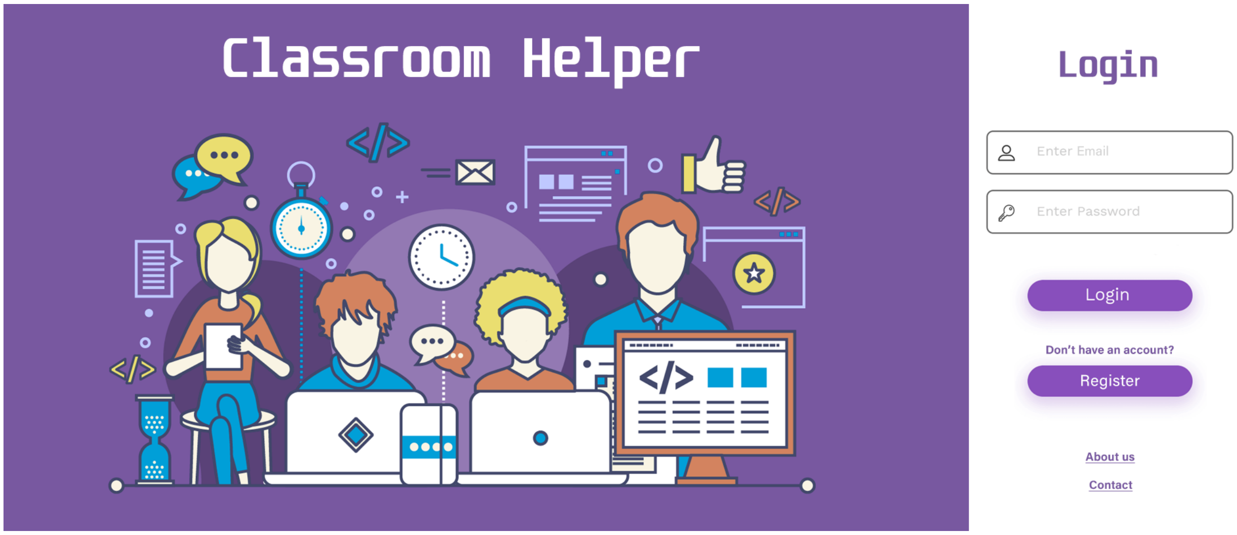
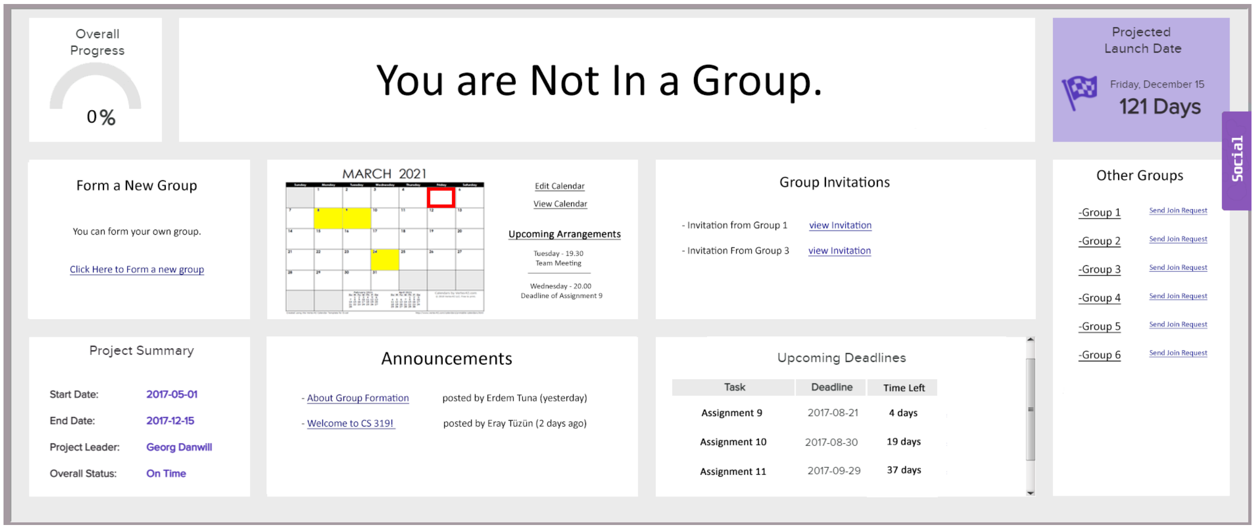


Figure 11

Figure 11 shows the Login Interface which will be the main page when a user enters the application. The user can either register from here or login to the system. If the user hasn't registered before therefore don't have an account, s/he will click register and enter the necessary information. Afterwards the system will save the user info. If the user has already registered then s/he will enter email and password to login to the system.

Dashboard of students with no group.



Overall Progress

0%

You are Not In a Group.

Projected Launch Date

Friday, December 15

121 Days

Social

Form a New Group

You can form your own group.

Click Here to Form a new group

MARCH 2021

Edit Calendar

View Calendar

Upcoming Arrangements

Tuesday - 19.30
Team Meeting

Wednesday - 20.00
Deadline of Assignment 9

Group Invitations

- Invitation from Group 1 view Invitation
- Invitation From Group 3 view Invitation

Other Groups

-Group 1	Send Join Request
-Group 2	Send Join Request
-Group 3	Send Join Request
-Group 4	Send Join Request
-Group 5	Send Join Request
-Group 6	Send Join Request

Project Summary

Start Date:	2017-05-01
End Date:	2017-12-15
Project Leader:	Georg Danwill
Overall Status:	On Time

Announcements

- About Group Formation posted by Erdem Tuna (yesterday)
- Welcome to CS 319! posted by Eray Tüzün (2 days ago)

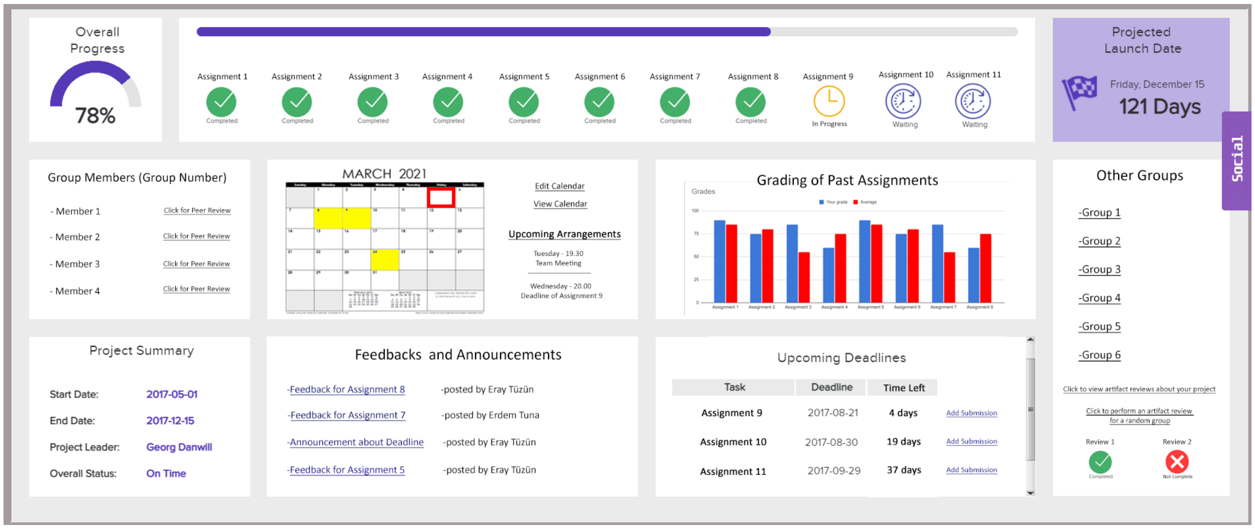
Upcoming Deadlines

Task	Deadline	Time Left
Assignment 9	2017-08-21	4 days
Assignment 10	2017-08-30	19 days
Assignment 11	2017-09-29	37 days

Figure 12

If the student is not in a group then some functionalities will not be visible to this student. However, s/he will be able to view other groups and send join requests to them so that s/he can join a group. Also groups can send invitations to individual students to join their group so students will be able to see these invitations. The student will be able to see announcements made by the instructor.

Student Dashboard



Overall Progress

78%

Assignment 1	Assignment 2	Assignment 3	Assignment 4	Assignment 5	Assignment 6	Assignment 7	Assignment 8	Assignment 9	Assignment 10	Assignment 11
Completed	Completed	Completed	Completed	Completed	Completed	Completed	Completed	In Progress	Waiting	Waiting

Projected Launch Date

Friday, December 15

121 Days

Social

Group Members (Group Number)

- Member 1	Click for Peer Review
- Member 2	Click for Peer Review
- Member 3	Click for Peer Review
- Member 4	Click for Peer Review

MARCH 2021

Edit Calendar

View Calendar

Upcoming Arrangements

Tuesday - 19.30
Team Meeting

Wednesday - 20.00
Deadline of Assignment 9

Grading of Past Assignments

Other Groups

-Group 1

-Group 2

-Group 3

-Group 4

-Group 5

-Group 6

Click to view artifact reviews about your project

Click to perform an artifact review for a random group

Review 1	Review 2
Completed	Not Complete

Project Summary

Start Date:	2017-05-01
End Date:	2017-12-15
Project Leader:	Georg Danwill
Overall Status:	On Time

Feedbacks and Announcements

-Feedback for Assignment 8 -posted by Eray Tüzün

-Feedback for Assignment 7 -posted by Erdem Tuna

-Announcement about Deadline -posted by Eray Tüzün

-Feedback for Assignment 5 -posted by Eray Tüzün

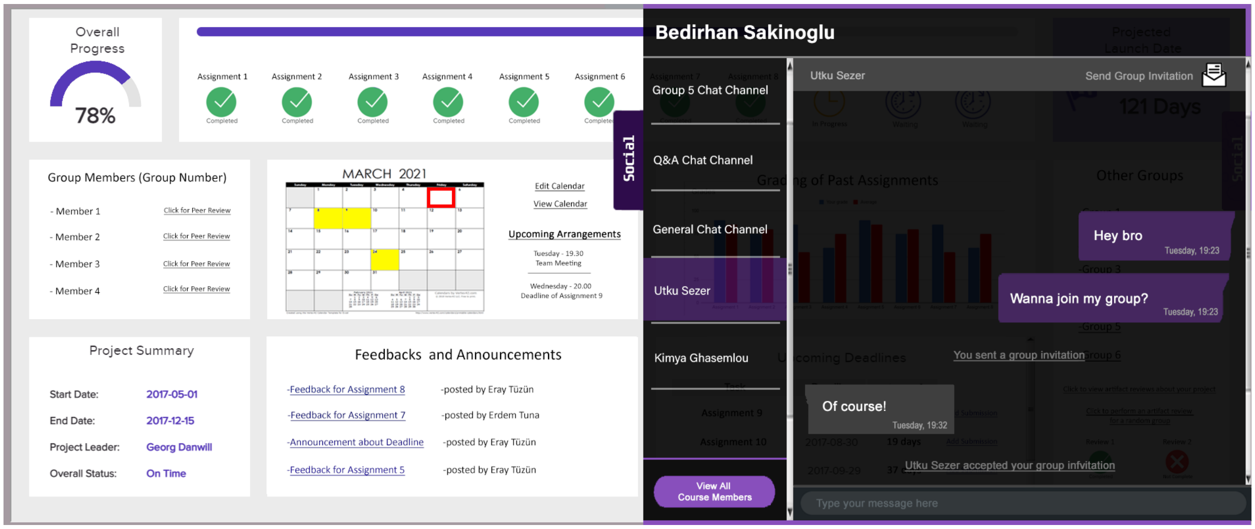
Upcoming Deadlines

Task	Deadline	Time Left	
Assignment 9	2017-08-21	4 days	Add Submission
Assignment 10	2017-08-30	19 days	Add Submission
Assignment 11	2017-09-29	37 days	Add Submission

Figure 13

Figure 13 shows how the dashboard is when a student is in a group. Students can see which members are in his/her group, see the group's progress in time and also can see the group calendar. This calendar is editable by the members of the group so they can add new events for their project meetings as well. Additionally, feedback for the artifacts given by the instructor, announcements made by the instructor can be seen. Statistics for the course can be seen in the graph. Also, upcoming deadlines and their submission links are in the dashboard. Finally, the students can review artifacts of other group's and see which are reviewed and which are not.

Chat



Group Members (Group Number)

- Member 1
- Member 2
- Member 3
- Member 4

Upcoming Arrangements

Tuesday - 19.30
Team Meeting

Wednesday - 20.00
Deadline of Assignment 9

Project Summary

Start Date:	2017-05-01
End Date:	2017-12-15
Project Leader:	Georg Danwill
Overall Status:	On Time

Feedbacks and Announcements

-Feedback for Assignment 8	-posted by Eray Tüzün
-Feedback for Assignment 7	-posted by Erdem Tuna
-Announcement about Deadline	-posted by Eray Tüzün
-Feedback for Assignment 5	-posted by Eray Tüzün

Bedirhan Sakinoglu

Utku Sezer

Hey bro
Tuesday, 19:23

Wanna join my group?
Tuesday, 19:23

You sent a group invitation

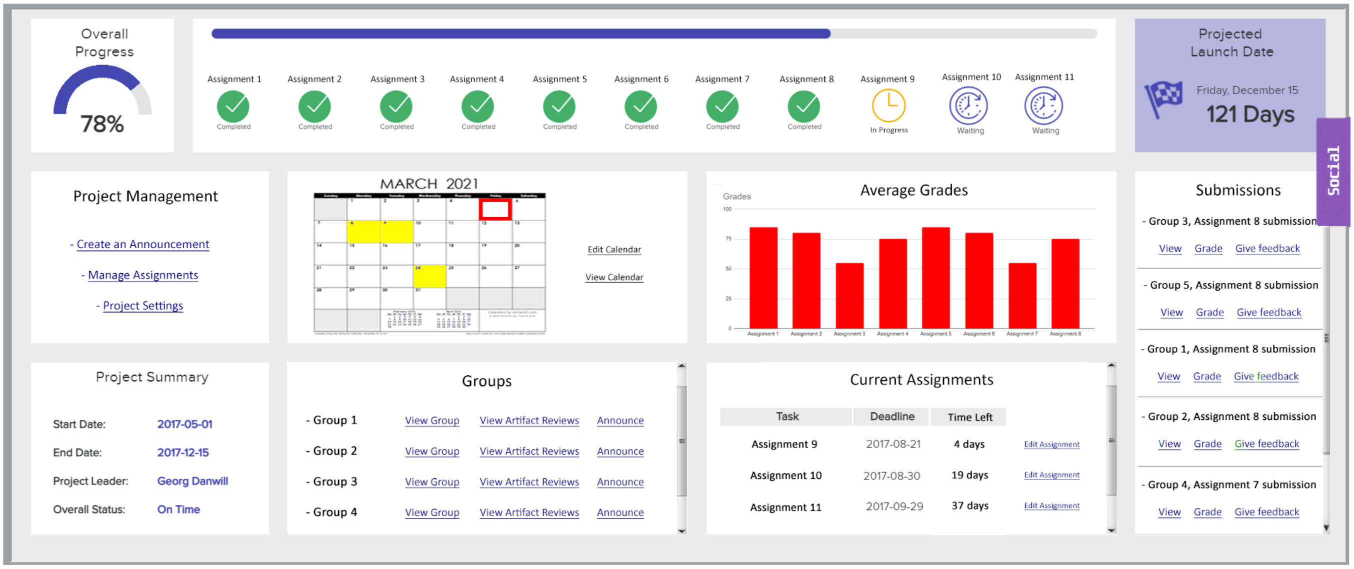
Of course!
Tuesday, 19:32

Utku Sezer accepted your group infvitation

Figure 14

Figure 14 shows how chat functionality works. By sliding the "Social" bar, the user can see the chat. There will be group channels for each group, general chat channels and individual chat channels.

Instructor Dashboard



- Group 3, Assignment 8 submission
  View Grade Give feedback
- Group 5, Assignment 8 submission
  View Grade Give feedback
- Group 1, Assignment 8 submission
  View Grade Give feedback
- Group 2, Assignment 8 submission
  View Grade Give feedback
- Group 4, Assignment 7 submission
  View Grade Give feedback

Project Summary

Start Date:	2017-05-01
End Date:	2017-12-15
Project Leader:	Georg Danwill
Overall Status:	On Time

Groups

- Group 1	View Group	View Artifact Reviews	Announce
- Group 2	View Group	View Artifact Reviews	Announce
- Group 3	View Group	View Artifact Reviews	Announce
- Group 4	View Group	View Artifact Reviews	Announce

Current Assignments

Task	Deadline	Time Left	
Assignment 9	2017-08-21	4 days	Edit Assignment
Assignment 10	2017-08-30	19 days	Edit Assignment
Assignment 11	2017-09-29	37 days	Edit Assignment

Figure 15

Figure 15 shows the instructor's dashboard. It is different from student's. Instructors can create announcements for the groups, manage and edit the assignments and also change details about the project. Instructors will also have a calendar to add events in everyone's calendar so that each group is aware of the upcoming deadlines. They can also view each group, their artifact reviews and create a specific announcement for a group. Instructors will have a graph chart so that they can see averages of each artifact. Finally, instructors will be able to each artifact submission, grade them and give feedback to them.

6. Improvement Summary

We've started correcting our mistakes from top to the bottom while adding what we've missed in the first iteration. We added some additional use cases according to the feedback that we've received from our instructor. For example, there were no use cases related to viewing the dashboards or statistics. So we've added those use cases. There were some minor mistakes in the sequence diagrams such as method names. We've changed those according to our updated version of the class diagram. We've also changed the activity diagram because we've misunderstood the main function of the activity diagram. So, we created the scenario of adding a submission for the activity diagram and added activities and decision nodes according to this scenario. Additionally, we've changed the state diagram for a single assignment object to display what states do assignment take in the application. Lastly, we've changed the class diagram as well. There were less multiplicities and some incorrect relations between classes so we fixed them. Also, we've added new classes and new operations in existing classes. Overall we had some diagram issues due to some misunderstandings of the functionalities of the diagrams so we corrected them.

7. Conclusion

We are currently using systems such as moodle and slack as tools in the class works and group projects. However, these tools are insufficient from the aspect of being a whole system and needing to use them interchangabily. This new proposed system is supposed to provide all the neccesary functions such as group and instructor-student communications, submission of the assignments, required artifact reviews and peer reviews for the group members. This is a system which will make it easier to communicate and do class work specially for the courses which contain teamwork.

Since the time for development of this application was limited, we prefer to only include the crucial and some beneficial functionalities but it can be expanded to include other functionalities when it is required.

8. Glossary & References

[1] Brügge, B., & Dutoit, A. H. (2004). *Object oriented software engineering using UML, patterns, and Java* (Third ed.). Upper Saddle River, NJ: Pearson Education.