İhsan Doğramacı Bilkent University

Department of Computer Engineering

CS319 Object Oriented Software Engineering

Classroom Helper

Project Analysis Report

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

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 can log in with their emails and passwords. They will be redirected 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 vs.) 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 determined by the instructor. 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. The template for peer review will be made and available by the instructor. 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 Login screen. In case of not having an account before, the user can go to Sign Up screen and create a new account by entering the information and the user type (Instructor/Student). If the user has an account already s/he can enter the program by entering the username and password.

- **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, what are the important deadlines for the project etc.

- **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.

Performance

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 progressed 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

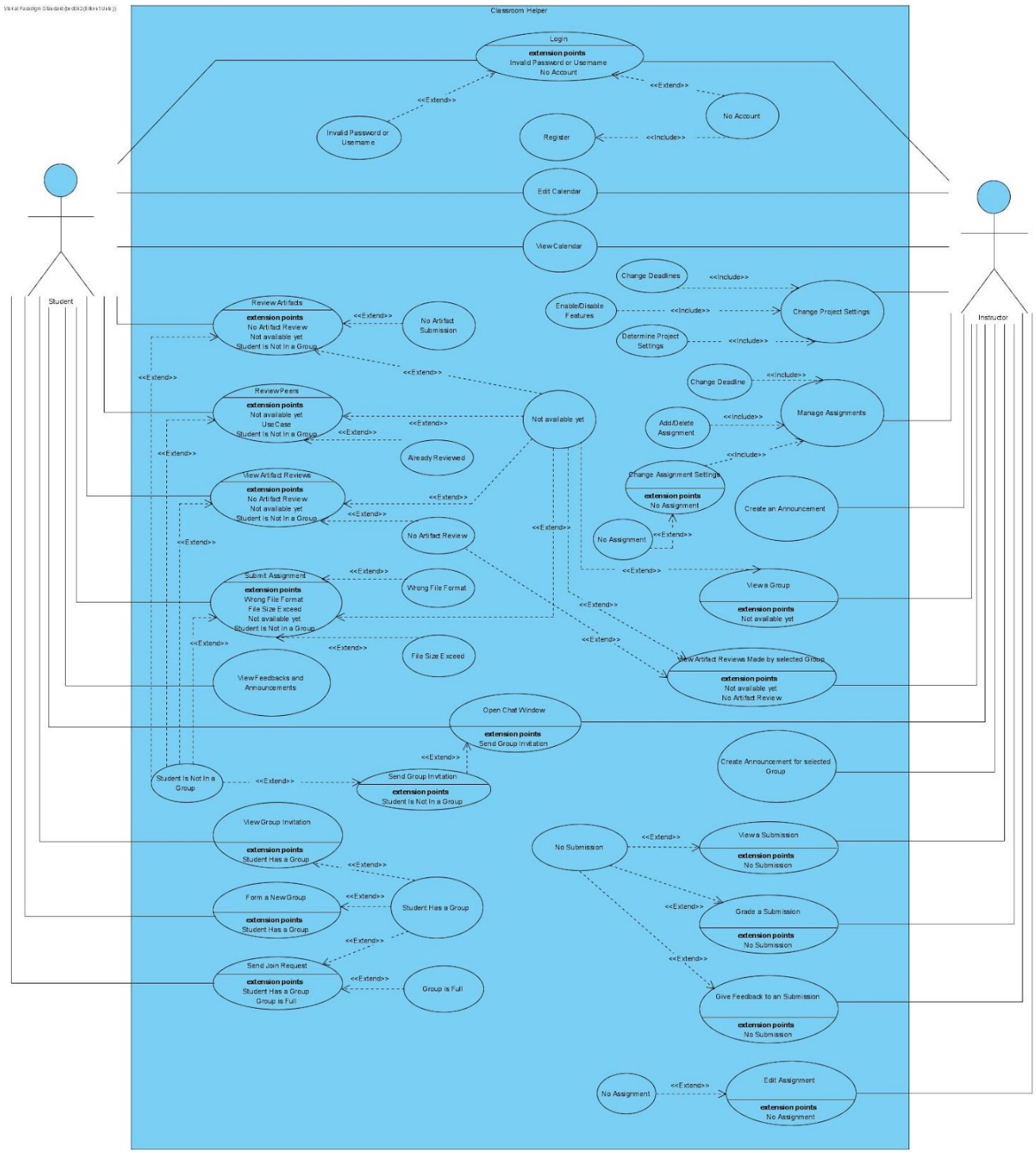

Figure 1

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 “Log In” button

Special/Quality Requirements: Server must be up

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 “Log In” button
3. A notification appears

Special/Quality Requirements: Server must be up

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

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

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:

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:

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
- 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:

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:

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:

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:

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
- There must be existing artifact reviews
- 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:

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:

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:

- Actor must be in a group
- Assignment upload must be enabled by Instructor/TA

Name: View Feedbacks And Announcements

Participating Actors: Student

Entry Condition: Actor clicks the button to see the announcements and feedbacks

Exit Condition: Actor clicks the previous button to exit

Flow of Events:

1. Actor clicks the view announcement or view feedback button
2. Actor sees the feed back or announcement
3. Actor clicks the previous button to exit

Special/Quality Requirements:

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:

- Actor must be in a group
- Assignment upload must be enabled 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:

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:

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:

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

Special/Quality Requirements:

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:

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:

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:

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:

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:

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 does some configurations
3. Actor clicks exit button to return back to the dashboard

Special/Quality Requirements:

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:

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:

Name: Create Announcement For Selected Group

Participating Actors: Instructor/TA

Entry Condition: Actor clicks the button to create announcement for desired group

Exit Condition: Actor submits the announcement for that group

Flow of Events:

1. Actor clicks button to create announcement for desired group
2. Actor writes announcement for that group
3. Actor submits the announcement

Special/Quality Requirements:

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:

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:

- 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:

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:

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:

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:

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:

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:

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:

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:

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:

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:

5.2. Dynamic Models

5.2.1. Sequence Diagrams

Scenario 1: Artifact Review Sequence Diagram

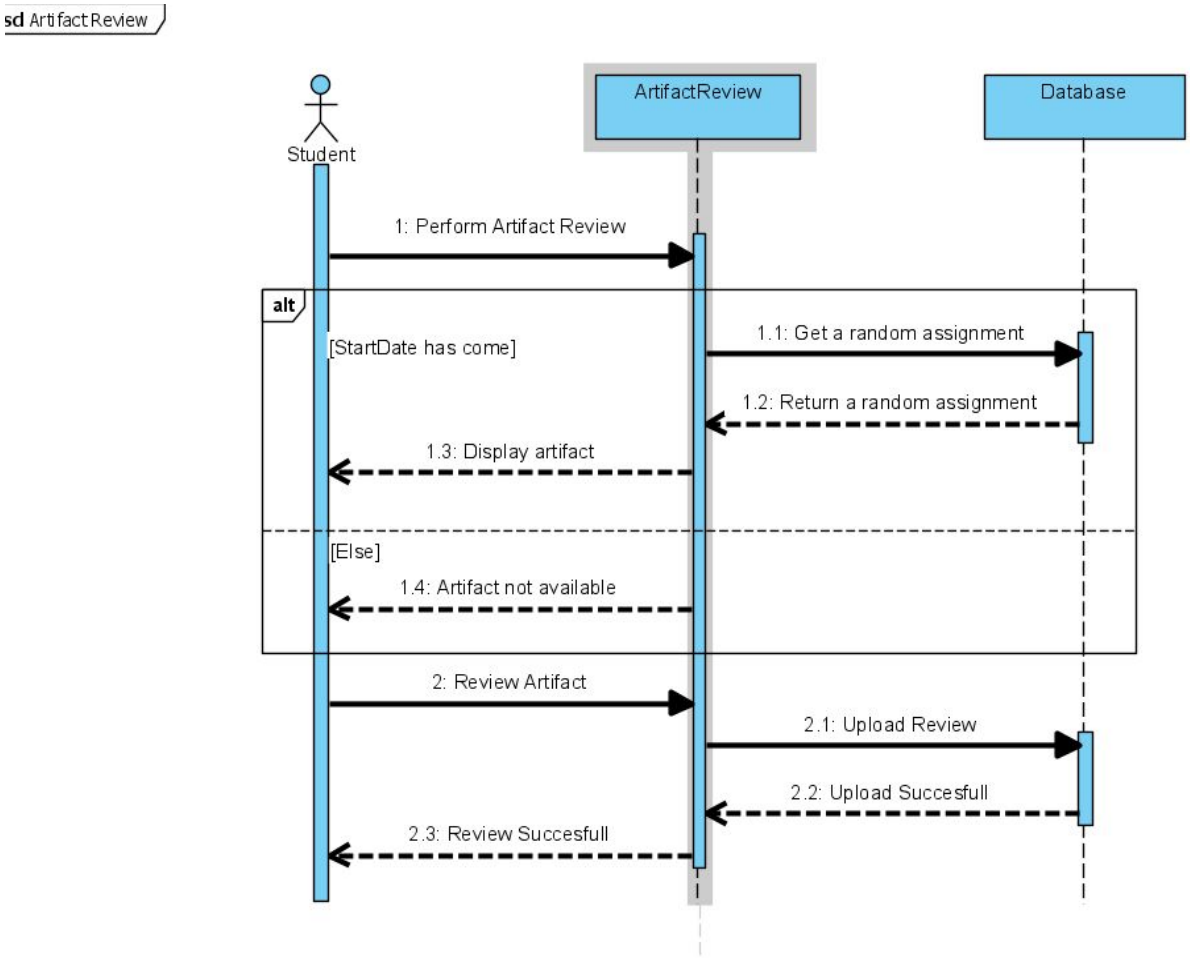

Figure 2

Scenario 2: Group Formation Sequence Diagram

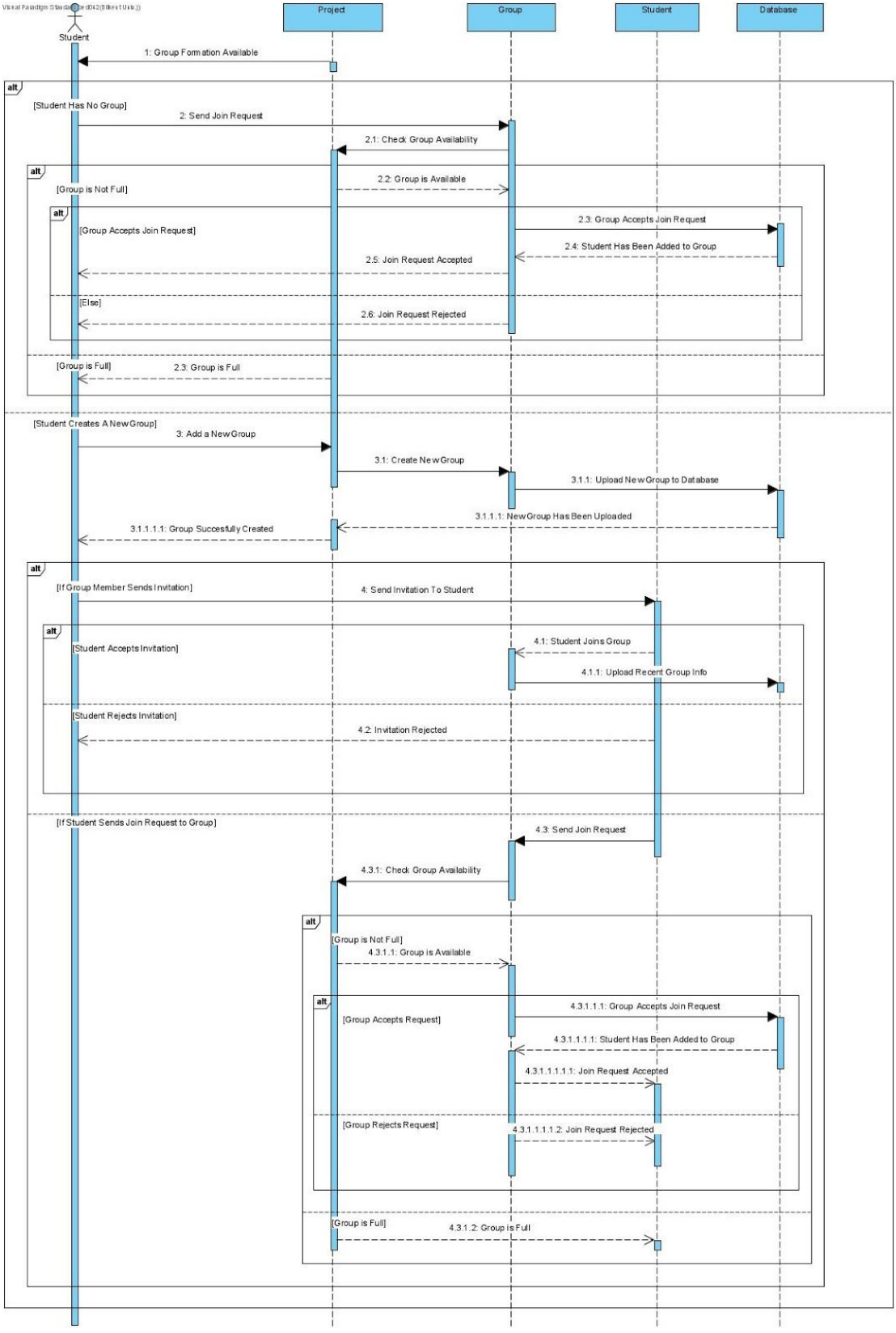

Figure 3

Scenario 3: Peer Review Sequence Diagram

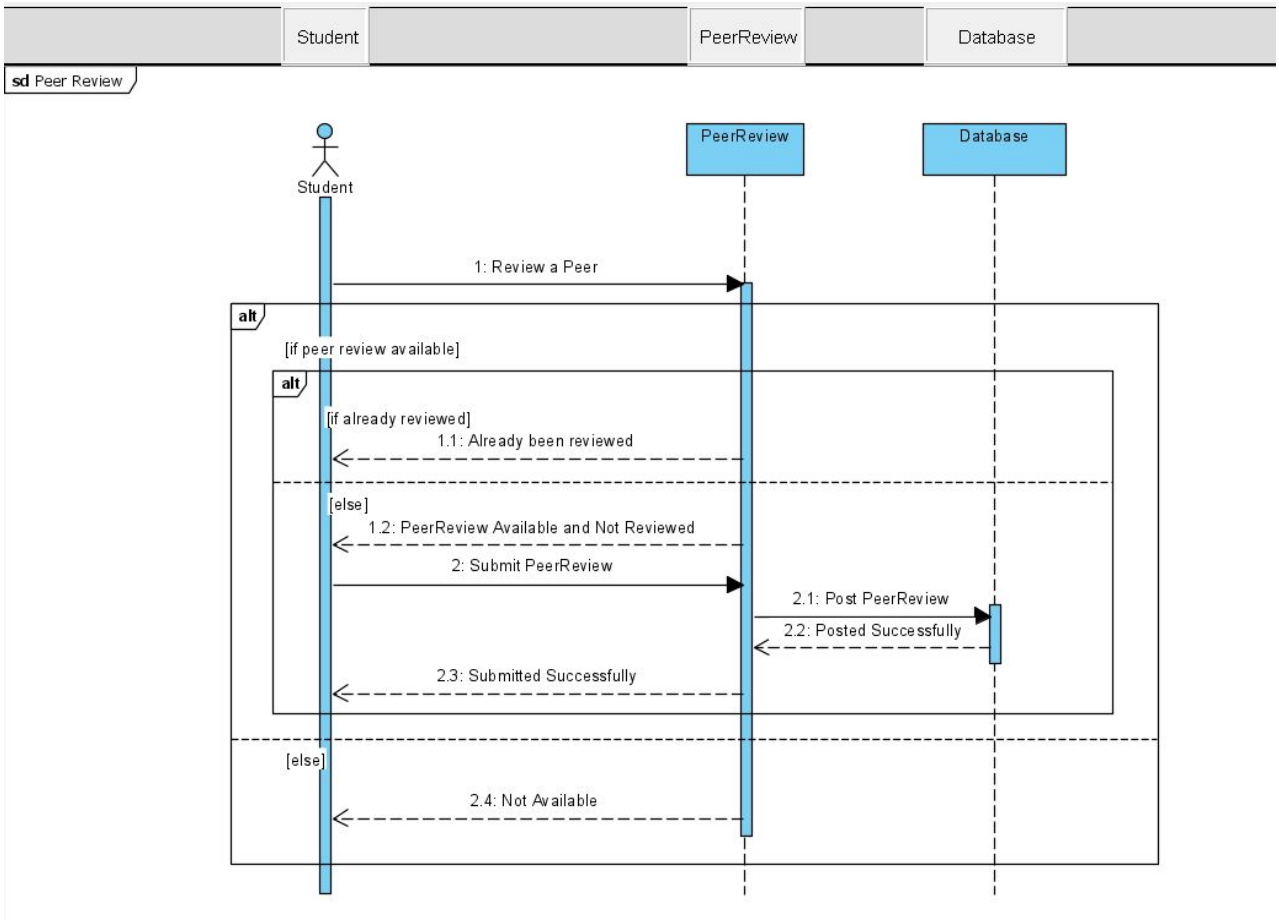

Figure 4

Scenario 4: Add Submission Sequence Diagram

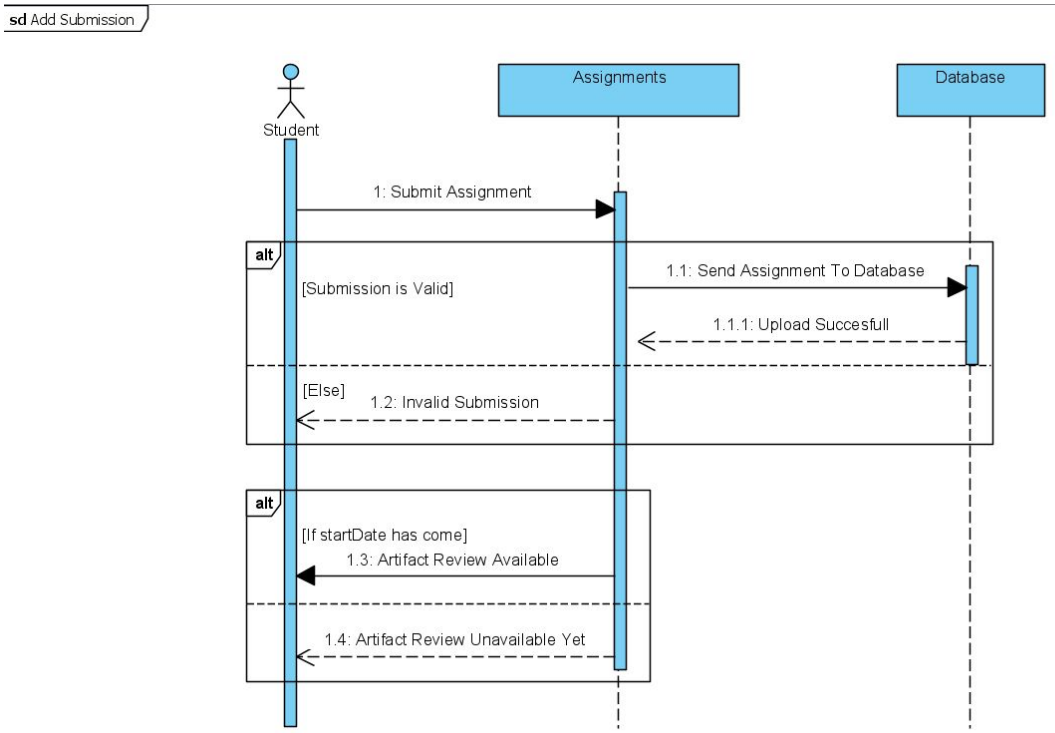

Figure 5

Scenario 5: Give Feedback Sequence Diagram

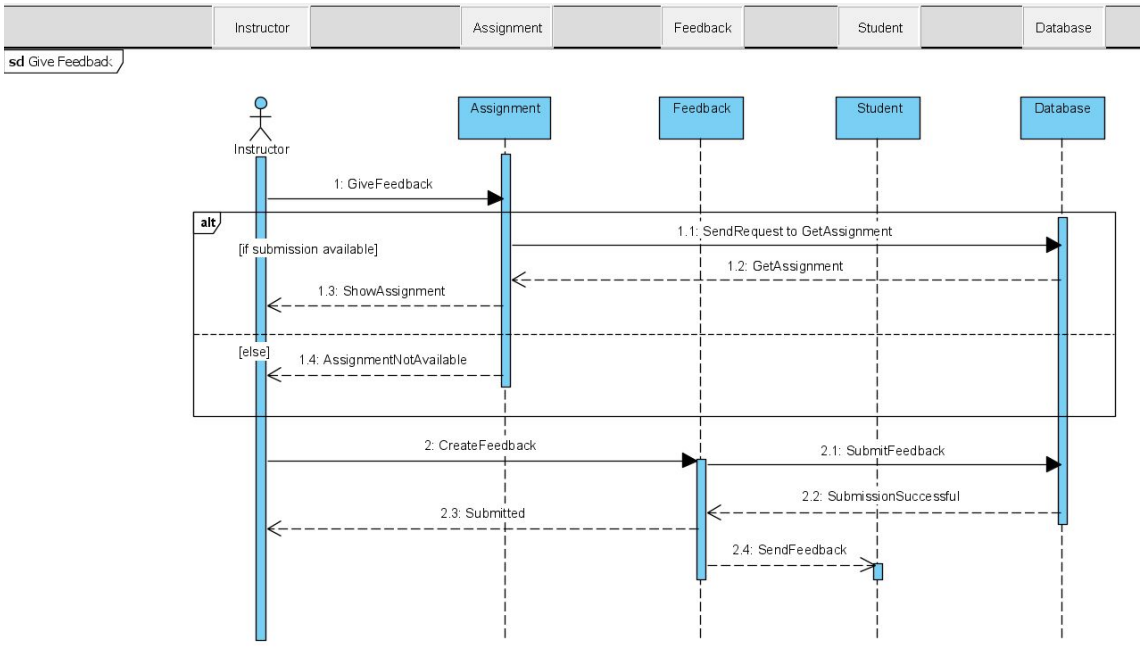

Figure 6

5.2.2. Activity Diagrams

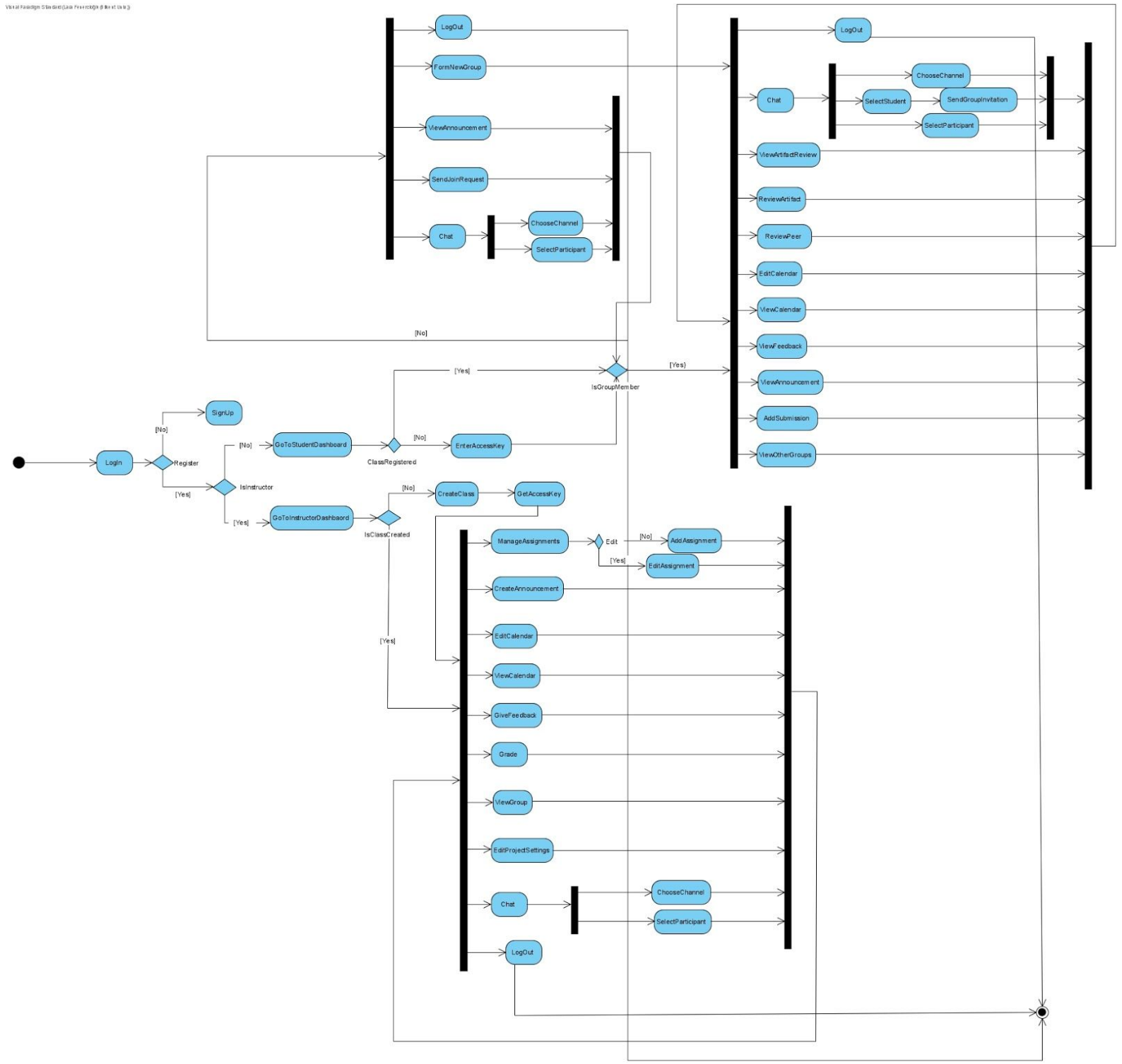

Figure 7

The above diagram explains the series of action flows that would denote the classroom helper.

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. This type identification will be kept in the database after the signUp process. 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. As we mentioned before, if the student has no group then his/her dashboard will have less functionality than those who have formed a group. Students without a group can form a new group without any member at first. Then this action will enable the student dashboard to have more functionality such that a group can do. Students without a group can also send a request to the groups. If the group accepts that request that isGroupMember condition will return true and the student will directly go to the dashboard again but with more functionality such that a group can do. Furthermore, students can chat with each other thanks to select participant action. Students can choose a channel where they can chat with all the users in one place. Students can also view announcements that the instructor/TA shared. If a student wants to exit the program, then s/he can just log out and the action flow finishes. If the student has a group then his/her dashboard will have more functionality such as reviewing an artifact, viewing an artifact review, reviewing a peer at the end of the semester, editing a calendar, viewing a calendar and viewing a feedback made by the instructor/TA. Students can also view an announcement, add a submission created by the instructor/TA or view other groups. Students with a group can use the chat function as well. A student can select

a student and then send an invitation to his/her group. Students can chat with other participants as well. Also students can choose any channel to message with her/his group, or chat with all the participants in one place.

Additionally, if the user type is Instructor/TA then the user will go to the instructor dashboard. Afterwards, if the instructor/TA hasn't already created a class then the instructor/TA will create a class and will get an access key. the instructor/TA will later send this access key to the students on their own (e.g. via email). From the manage assignment option in the dashboard the instructor will be able to create new assignments or edit the previously created ones. The instructor can make announcements, submit grades and give feedback for the assignments which will be uploaded by the students. The instructor can also edit project settings and these edits include changing the deadlines, determining the size of the groups, enabling peer review functionality and disabling group formation functionality. By the time instructor/TA disables the group formation functionality, students without group or incomplete groups will be turned into completed groups by a specific algorithm. He/She can also view the members of groups from the dashboard. There will be a calendar which shows the deadlines for assignments and instructors can edit or view the calendar in a more detailed and bigger version.

5.2.3 State Diagram

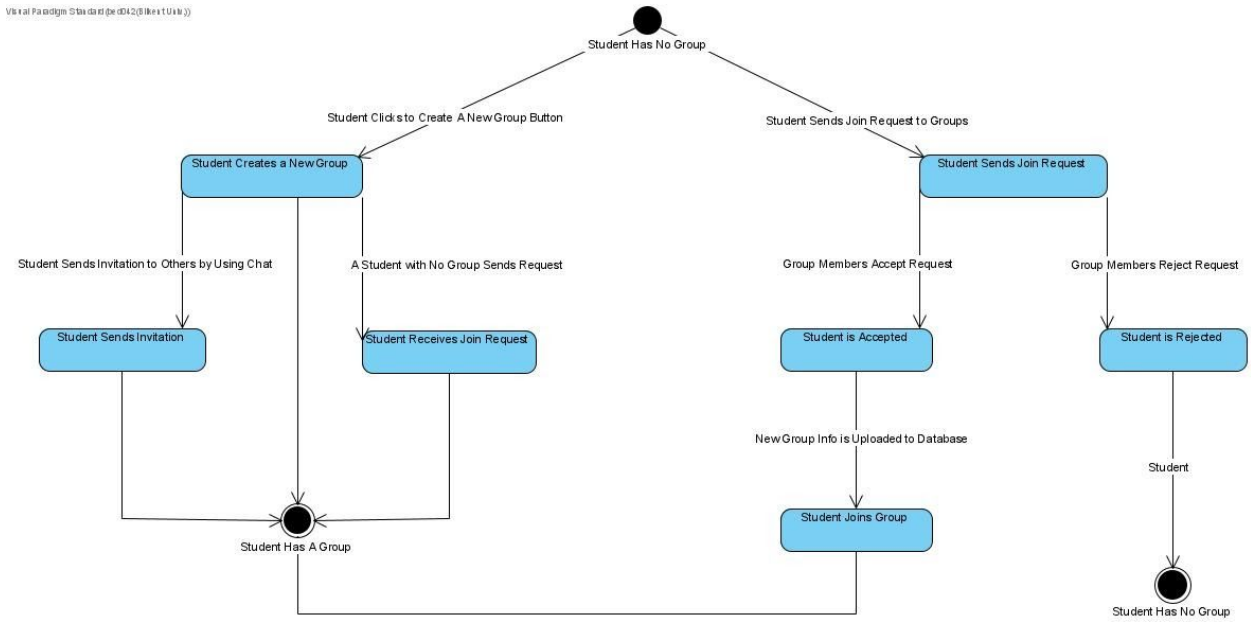

Figure 8

5.3. Object and Class Models

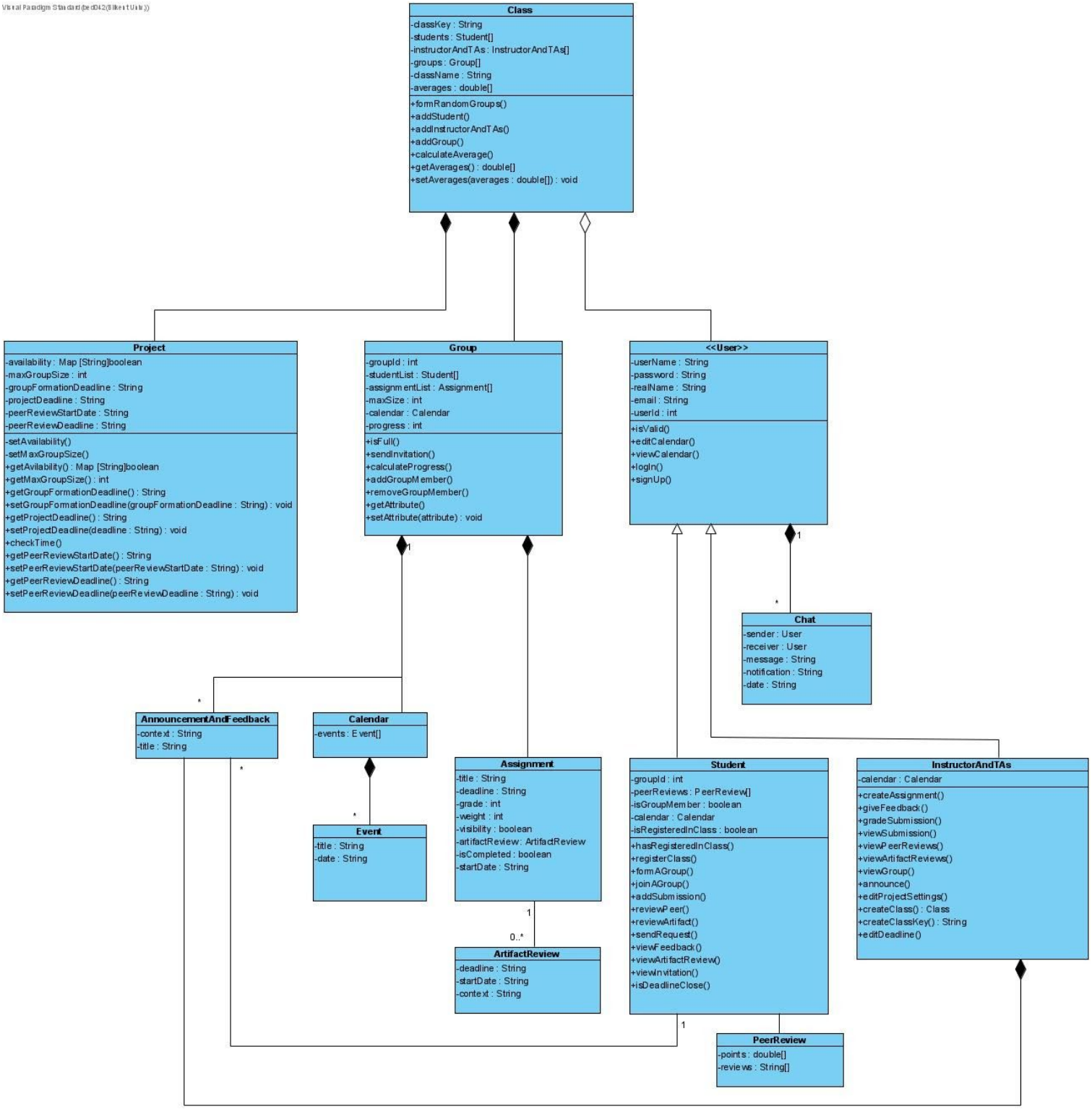

Figure 9

5.4. User interface - Navigational Paths

Login Screen

Figure 10

Dashboard of students with no group.

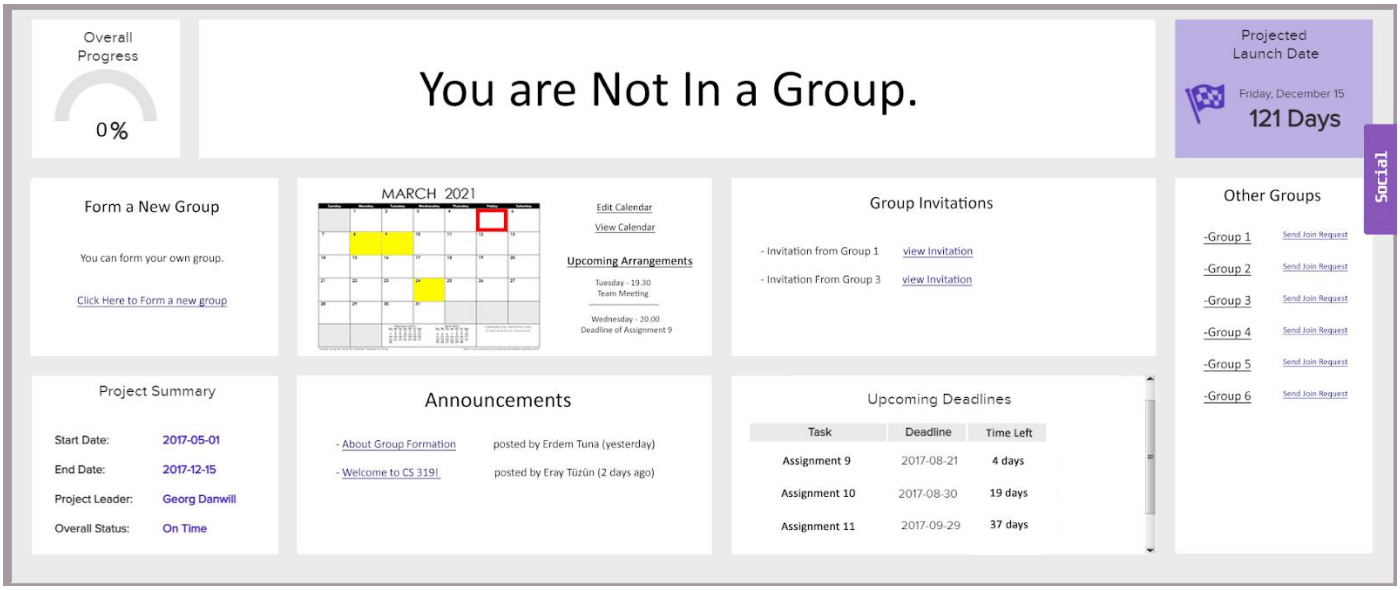

Figure 11

Student Dashboard

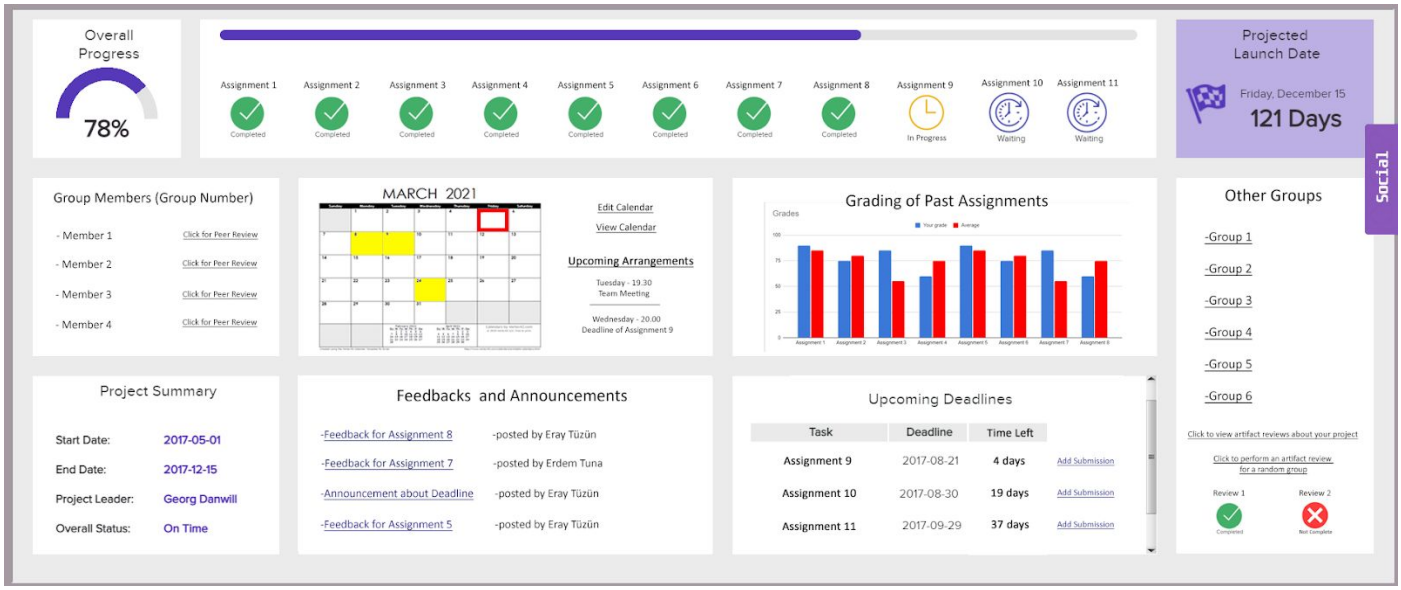

Figure 12

Chat

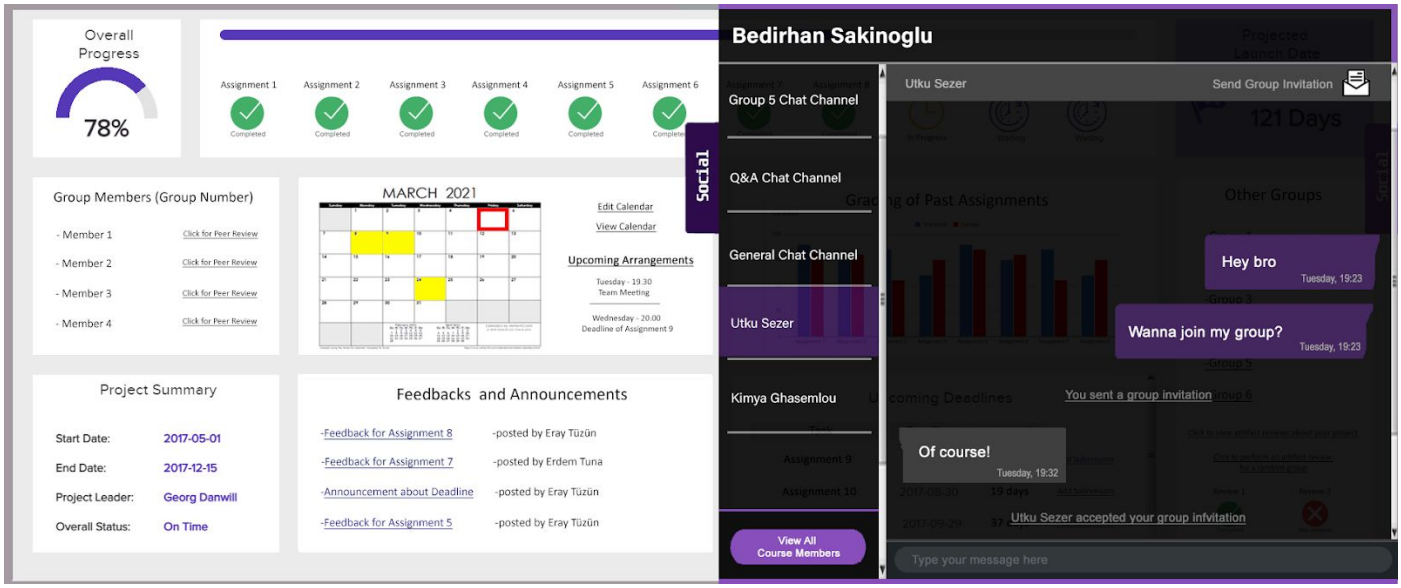

Figure 13

Instructor Dashboard

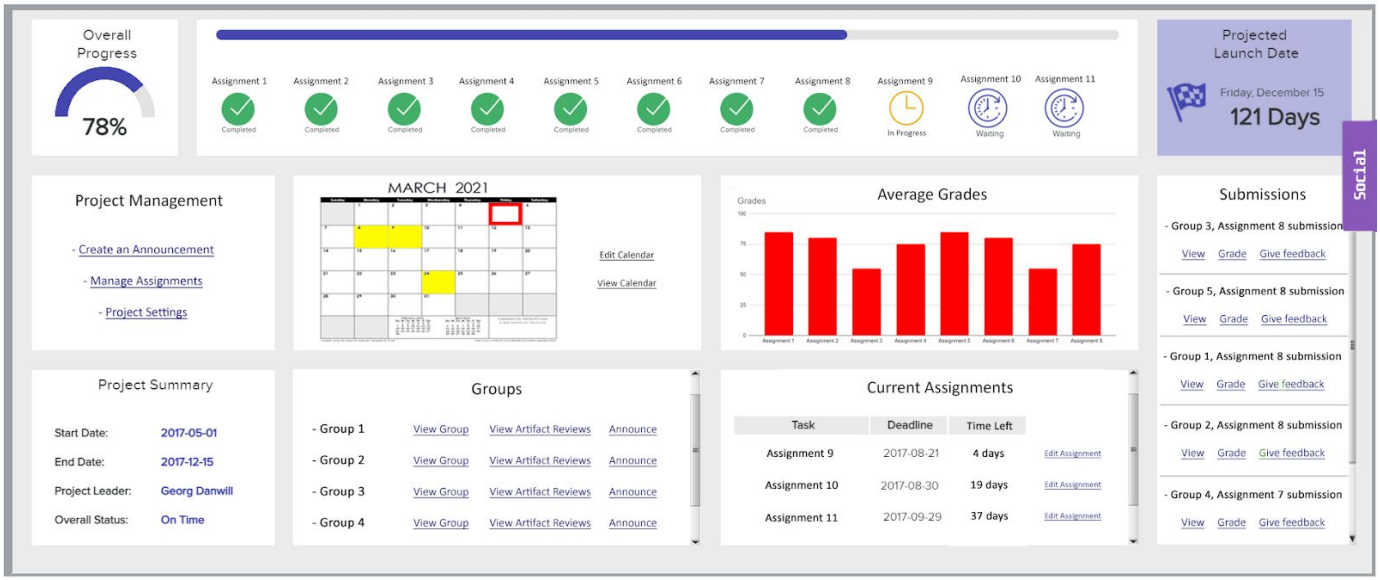

Figure 14

6. 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.