CS-319 TERM PROJECT

Peer-Review

Analysis Report

Group Members:

Eylül Çağlar (Group Leader) 21703949

İsmet Alp Eren 21703786

Kaan Ateşel 21703694

Kerem Alemdar 21702133

Muharrem Berk Yıldız 21802492

Instructor: Eray Tüzün

Teaching Assistant(s): Elgun Jabrayilzade, Erdem Tuna

Table of Contents

1.   Introduction

In this project, an application which enables students to make peer review and form a group will be implemented. The application will be a desktop application.

The purpose of application is facilitate the follow-up of instructors on students in term projects and to facilitate group formation. Considering the teachers' command of technology and their computer knowledge, an application will be developed with complex changeable options that have given the user more authority to make changes on.

Finding a project group is sometimes very hard and students need to contact TA's and instructors. This application will make everybody's life easier than before because it enables students to form their project groups directly by contacting class members. Since everybody in class does not know each other this application is valuable for this purpose.

On the other hand, instructors sometimes want to hear about their students' opinions about their friends' performance. This application will enable students to review their friends and will enable instructors to collect data by these reviews.

2.   Proposed System

2.1   Functional requirements

2.1.1 User Registration
- Users should be able to register for an application through his phone number. On the first run of the application user should register with his phone number.

2.1.2 User Profile
- Users should be able to configure his profile by setting their profile picture and could add additional information about his background, his previous projects and his abilities.

2.1.3 Groups

- Users could be able to manage their different project groups and these groups are linked to their profiles and could be seen which projects the user involved.

2.1.4 Contacts

- Users are able to add contacts to their contact list either by phone number of going to someone else's profile in project groups.

2.1.5 Send Message

- Users could send private messages to people who have mutually added each other to their contact lists. It is also possible to send messages to related groups.

2.1.6 Send Attachments

- In addition to messages, users are able to send documents and make those files fix on top.

2.1.7 Forming a Group

- Users should be able to create groups and can ad inside of the application to look for possible contributors for the project.

2.1.8 Create Poll

- Users could be able to create new polls or use old polls to determine what should be the upcoming events or what should be the way they will choose as a group.

2.1.9 Calendar

- Calendar includes both global and local activities where global means all class deadlines and local means all group deadlines.

2.1.10 Exporting

- Exporting group grades as an organized document for submitting grades to SRS efficiently.

2.1.11 Review And Drop Comments

- Users can review other member's tasks and make comments on them.

2.1.12 To do list and Progress Bar

- User can see the to do list and choose some of the and mark with his name to indicate that to do item will be made by that member. Also, members can see the progress of the project.

2.2 Nonfunctional requirements

2.2.1 Usability

Peer reviewing takes a fair amount of time .Any additional burden on users even extends this time requirement. Pire eases this process, helping students to give feedback to their team members by providing assessment templates, team activity graphs.Meanwhile, instructors and teaching assistants can create templates for peer reviewing. Students are given a class code which is generated by instructors. Class codes determine students' class while registering without any effort. Student groups have built in a to do list to keep track of objectives and plan their work throughout the project.

2.2.2 Supportability

Pire is written in Java meaning clients can be run on many platforms (Windows, Linux, Mac) thanks to JVM.

2.2.3 Reliability

All the data is saved and stored in real time to the remote database and each user data including chat messages, user credentials and documents.

2.3 Pseudo Requirement

All related software associated with Pire will be written using Java.

2.4. System models
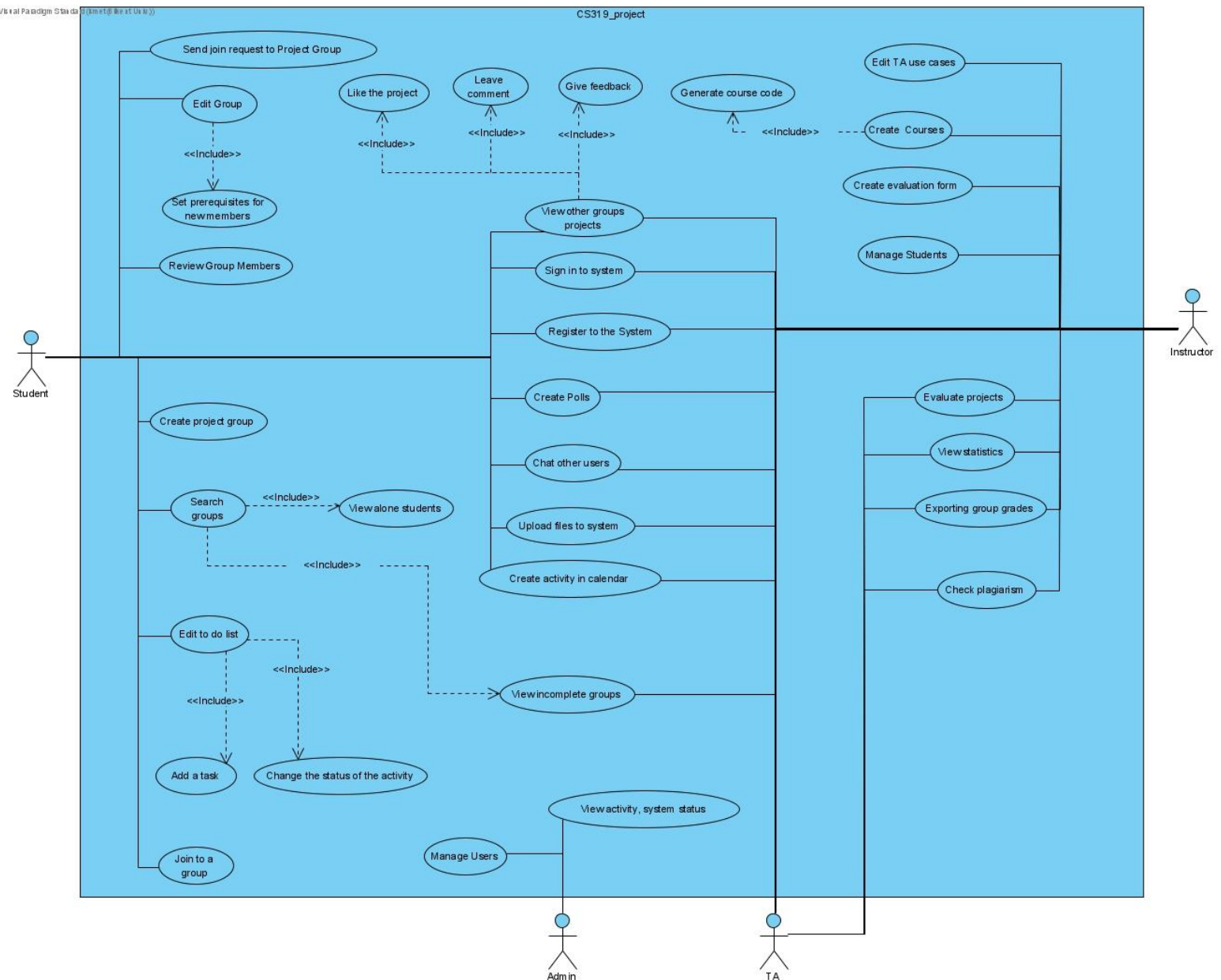
2.4.1. Use Case Diagram

Figure 1: Use Case Diagram showing the use cases of actors.

This application has four actors: Student, Instructor, TA and Admin.

Admin: Admin only has two actions which **View System Status** and **Manage Users.**

View System Status: Admin can see the system status that include how many students use the system, what are the rush times etc.

Manage Users: Only the manager can give ….. to the instructor, TAs and students.

Instructor VS TA: Both instructor and TA have almost the same number of actions but instructor has more administrative functions.

Create Course: Only instructors can create courses and publish that course.

Edit TAs' Use Cases: Instructors can edit TAs' permissions and authority.

Create Evaluation Form: Instructors can create an evaluation form and publish that form when group evaluation date has begun.

Manage Students: Instructors can see the students who are idle and who are not. After most of the students found their group, instructors can assign idle students randomly or manually to new groups or not completed groups.

2.4.2. Dynamic models

2.4.2.1 Activity Diagrams

2.4.2.1.1 Authentication Activity Diagram

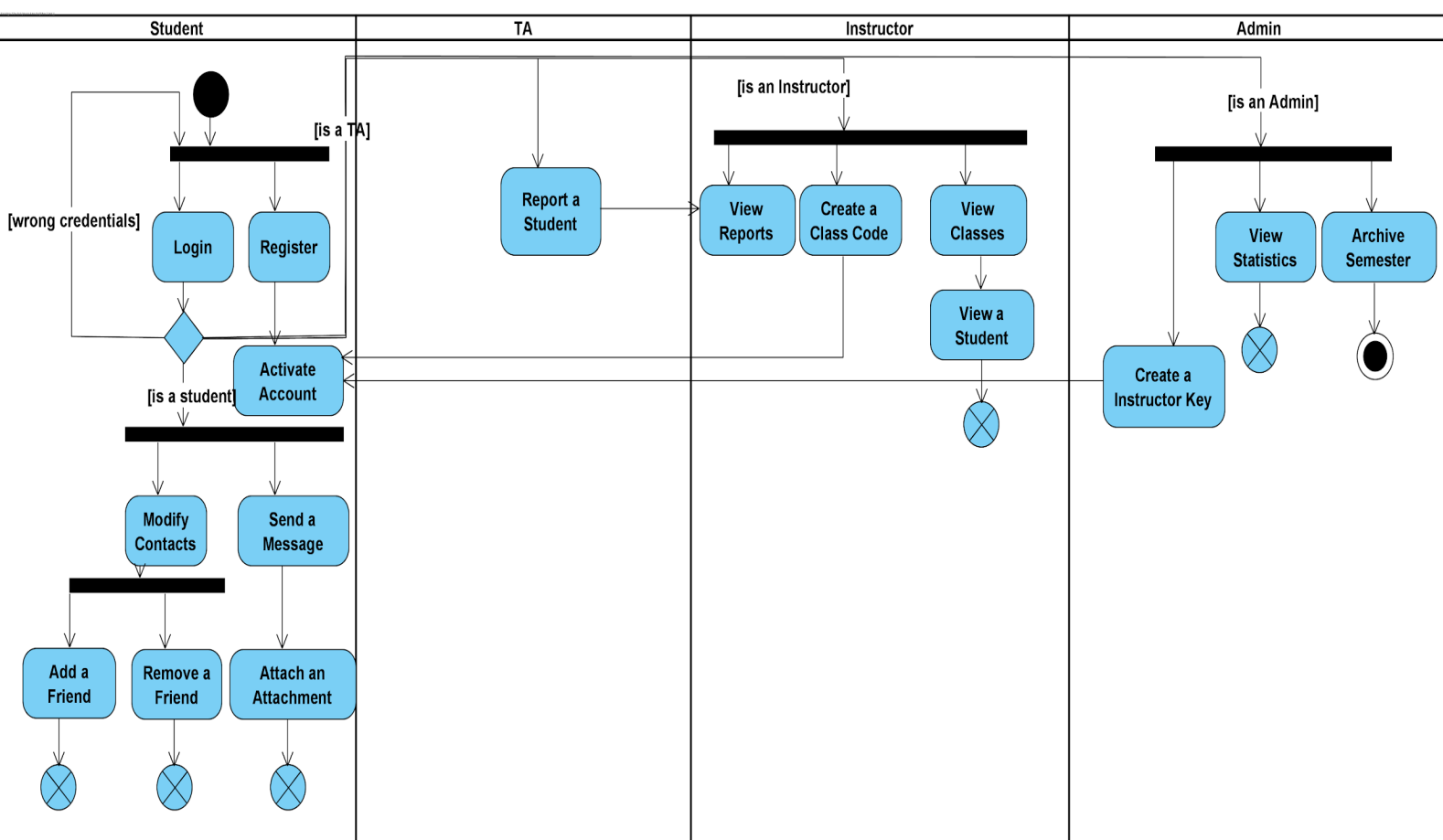**Figure 2:** Activity Diagram showing the authentication progress of the application.
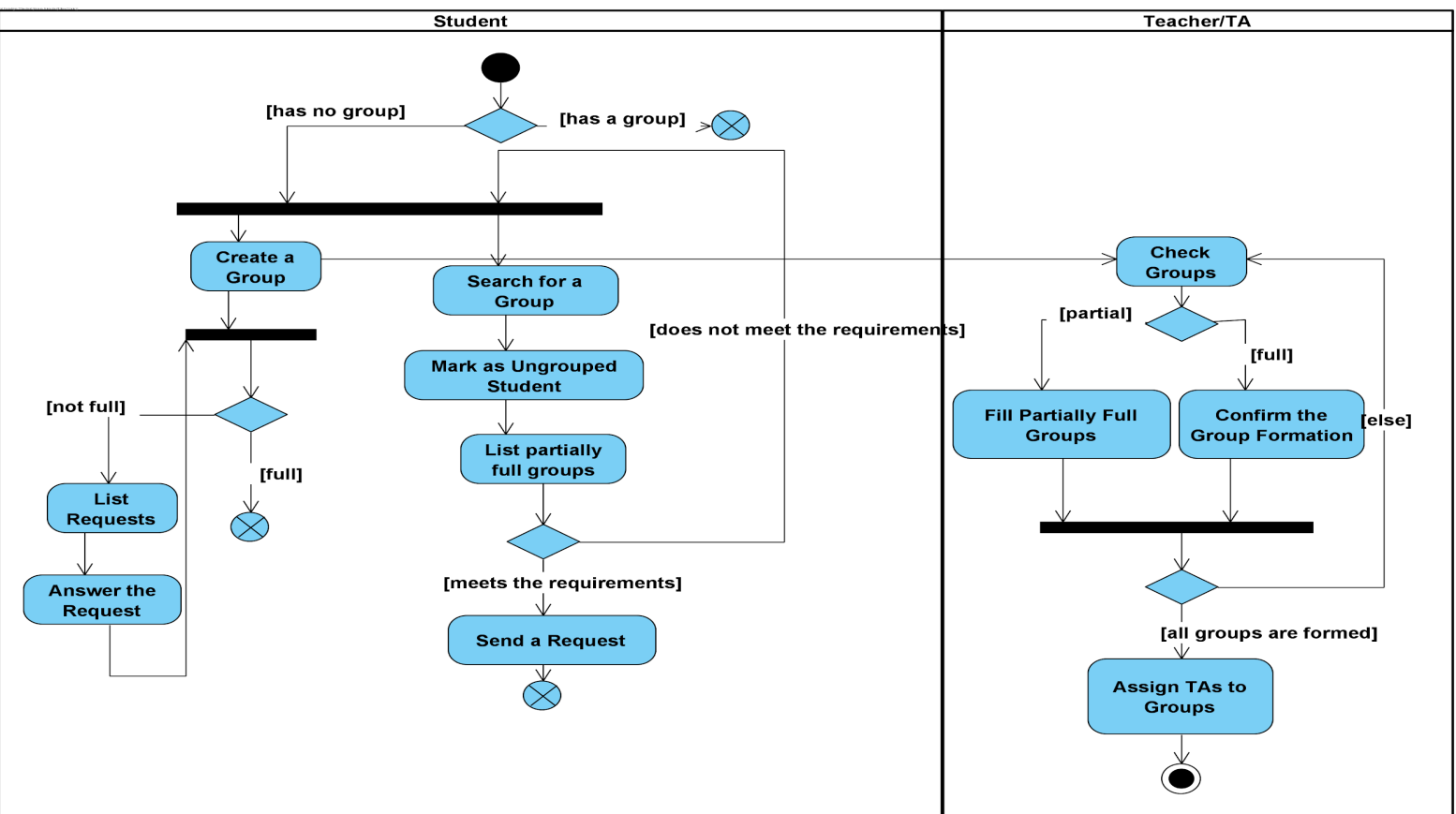
2.4.2.1.2 Group Management Activity Diagram

Figure 3: Activity Diagram showing the group progress of the application.
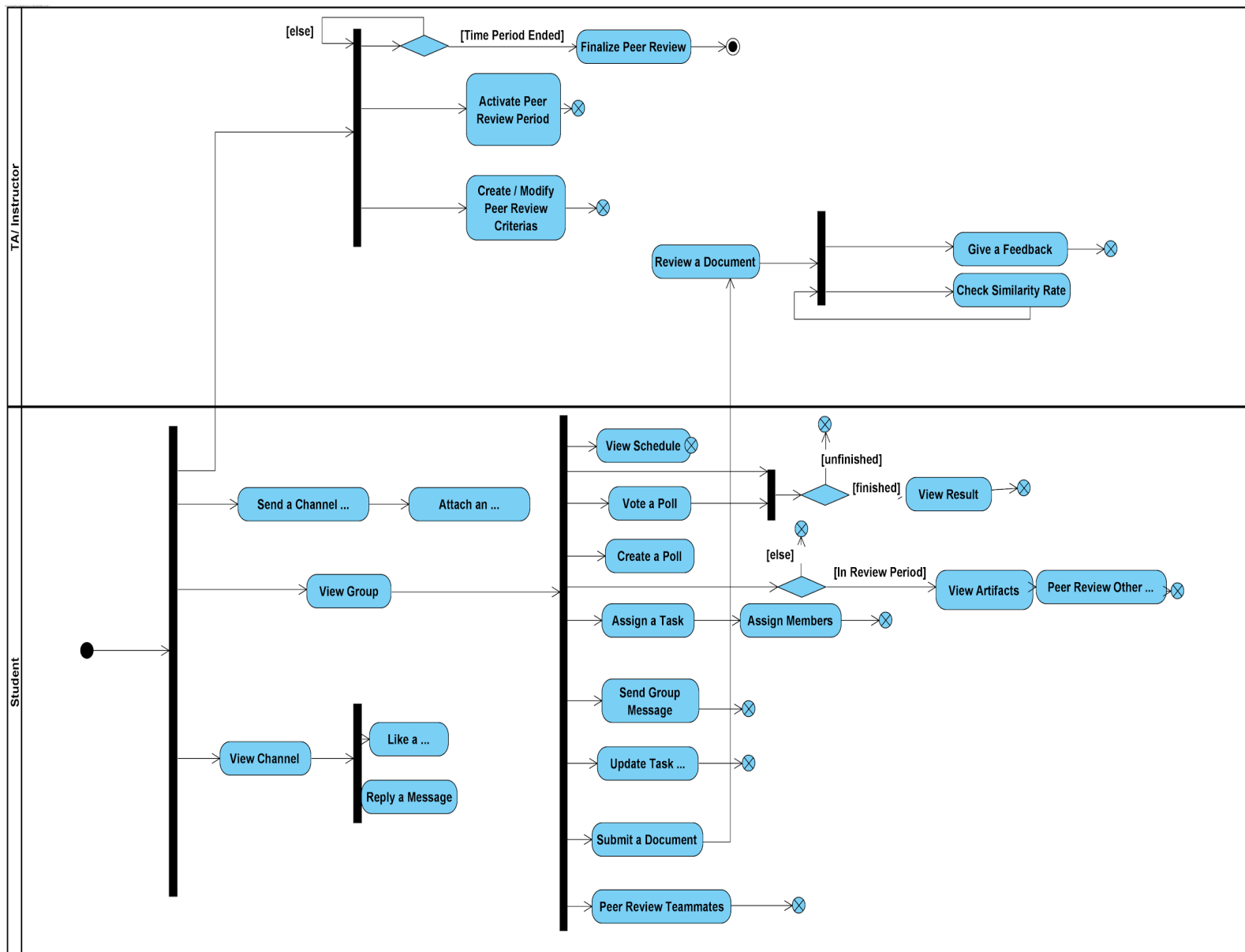
2.4.2.1.3 Reviewing Activity Diagram

Figure 4: Activity Diagram showing the authentication progress of the application.
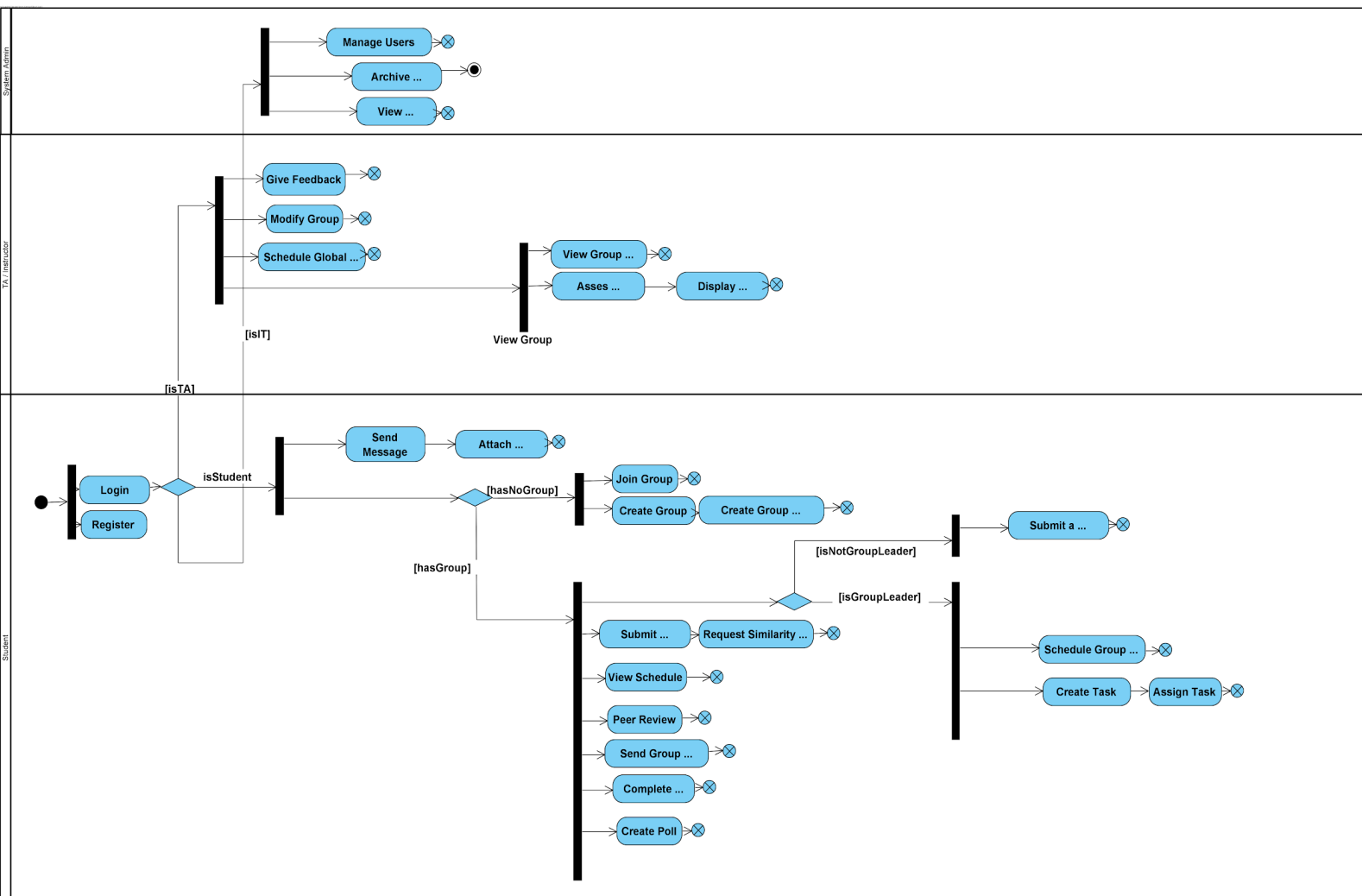
2.4.2.1.4 System Activity Diagram

Figure 5: Activity Diagram showing the authentication progress of the application.

2.4.2.2 State Diagram
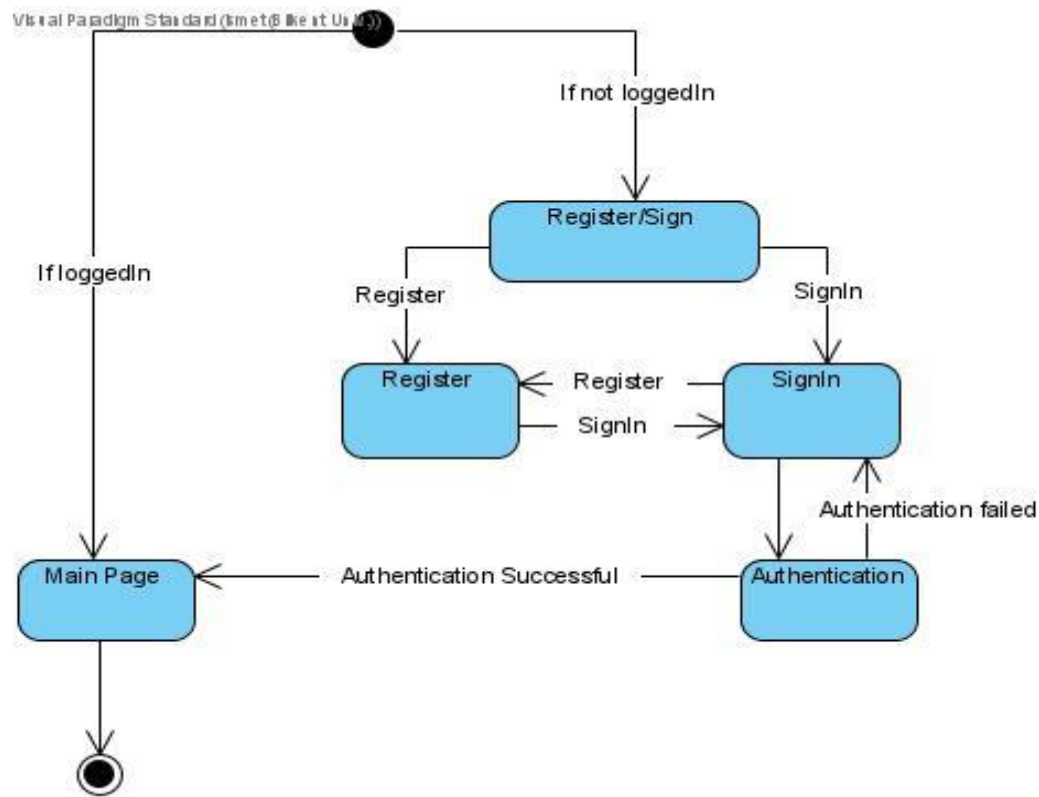
2.4.2.2.1 Authentication State Diagram

Figure 6: State Diagram showing the authentication states and transitions between them.

Initial State: After the user launches the application, if the user already logged in before authentication state will direct the user automatically to the **Main Page.** On the other case which user logging in first time then authentication state will present two choices to the user.

Register: User will indicate his/her information, password etc. and then if all inputs satisfy the conditions (min length of password etc.) application will direct the user to **SignIn Page**.

SignIn: If the user doesn't have an account yet, he/she can go to the Register Page or continue with SignIn stage. After needed inputs are filled, the authentication manager will check the account from the database and if the account exists in the database, the authentication manager will let the user to the main page.
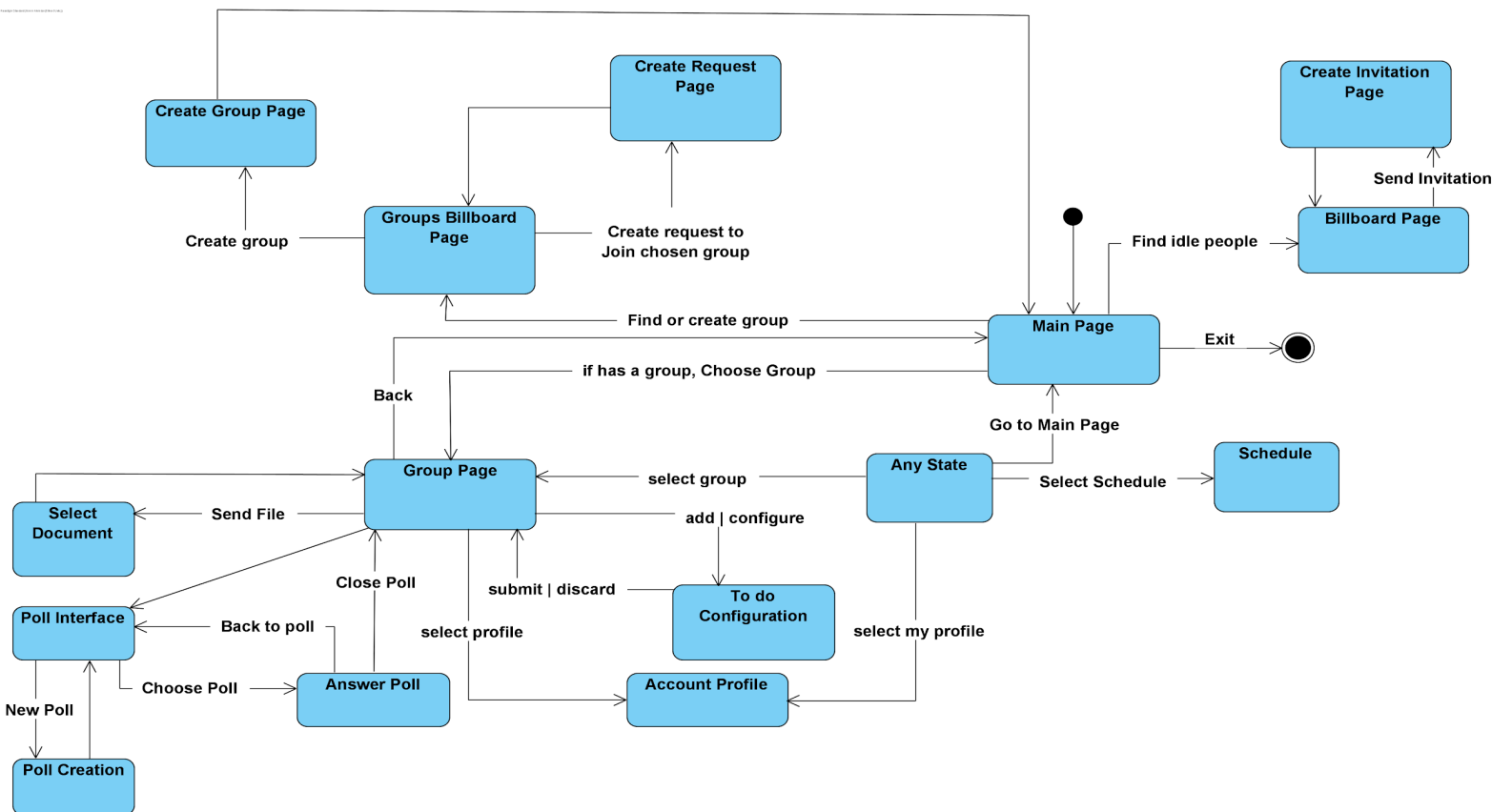
2.4.2.2.2 Main State Diagram

Figure 7: State diagram showing the main page states of the applications.

Any State: Any state represents the top bar of our application. Regardless of which screen you are, you could use a top bar. At any state, you could view your profile, schedule, and you could go to the main page.

Main Page: It was the end of the previous state diagram. It is what you will encounter when you log in to our application. If a user wanted to join a new group or to create a group, by clicking the group billboard page button, he would be directed to the Group Billboard Page. If the user has already joined the existing group by clicking search for new members, he will be directed to the Billboard Page. Users

could see their own groups. If they select one of their groups, they will be directed to the Group Page.

Group Page: In this page, the user could see members of the group,his group chat, group documents, already created polls which belong to that group and to do list according to his group. If a user selects one of his group members he will be directed to their Profile Page.

Group Billboard Page: In this page you could create or search for a new group. All groups will be listed, you could select a group and start to create a join request. If you want to create your own group you could go to create a group page by clicking create group.

Create Group Page: User will indicate, group name, group subject and requirements to join the group. All of them are also adjustable later. When he is created he will be directed to the main page.

Create Request Page: User should add his CV and comments. After all required submissions are done and he clicks send request, he will be directed to the group billboard page for an alternative request if he wants.

Select Document: Windows based pop up for uploading file.

Poll Interface: In this page, users would rather create a new poll or answer an already existing one by selecting it from the poll list.

Poll Creation: User writes his question and multiple choices for which he wanted to poll. After the creation user will be directed to the Poll Interface again, and could see his poll in the poll list.

Answer Poll: Users will answer the poll which he chose in Poll Interface and submit their answer.

To do Configuration: Users could add a new task or configure any existing task.

Account Profile: Users will be directed to the group members profile which he chose in the Group Page.

Billboard Page: If a user has an already existing group, he could search for people to send them an invitation. In this page, they will see all students looking for a team.

Create Invitation Page: On this page users could add comments and indicate his group project subject. After sending the invitation, users will be directed to Billboard Page again to look for alternative group members.

2.4.2.3 Object and Class Model

Student
- -studentId : string
- -phoneNumber : string

Instructor

TA

Account
- -firstname : string
- -groups : Group[]
- -secondname : string
- -email : string
- -comments : string
- -password : string
- -accountType : string
- -completedCourses : string[]
- -accountId : string
- -reviews : InGroupReview[]
- -directMessages : Message[]
- -invitations : Invitations[]
- +joinGroup() : void

Invitation
- -sender : Group
- -receiver : Account
- -receiveDate : string
- -message : string

Schedule

inGroupReview
- -comments : string
- -reviewer : Account
- -reviewed : Account
- -grade : int
- -date : string

Review
- -comments : string[]
- -reviewer : Account
- -grade : int[]
- -date : string

Group
- -members : Account[]
- -groupName : string
- -groupId : int
- -documents : Document[]
- -messages : Message[]
- -polls : Poll[]
- -prerequisites : string
- -tasks : Task[]
- -requests : Request[]
- -schedule : Schdule
- +addTask() : void
- +deleteTask() : void
- +configureTask() : void
- +createPoll() : void
- +configurePoll() : void
- +deletePoll() : void
- +fireMember() : void

Message
- -senderName : string
- -receiverName : string
- -content : string
- -date : string
- -isSeen : boolean
- -priority : string
- -color : string

Task
- -owner : string
- -task : string
- -dueDate : string
- -ownerId : string
- -percentage : int

Poll
- -question : string
- -choices : string[]
- -title : string
- -result : int[]

Document
- -documentName : string
- -documentComments : string
- -date : string
- -senderName : string

Request
- -sender : Group
- -receiver : Account
- -message : string
- -sendDate : string

Figure 8: Class Diagram showing the class structures of the program.

Account: There are three types of account; Student, TA, Instructor. Account has personal information such as (name, password, email etc.) and other properties shown above.

Student: Students have two additional properties; studentId and phone number that are only visible by some people(group members and instructor) for urgency.

Invitation: Class that was created to invite students or TAs to the group. Account has an invitation list that keeps the not accepted invitations.

Schedule: Account has a schedule that contains both personal and groups' deadlines as marked with different colors.

In Group Review: Differently from the **Review**, in group review will be used to make a review to his/her group members and this type of reviews will be kept in **Account**.

Message: Message has a composition relationship with both Account and Group. Message class will be used in Account to keep direct messages. For the group, messages will be visible by that group's members.

Group: Group has a composition relationship with Task, Document, Review, Poll, Request and Message.

Task: Task can be created by all members but it has to be approved by the group leader. After a task approved by the leader, that task will be syncrozied with personal schedules.

Poll: Poll will be created by only the group leader.

Request: Each group has a request list that keeps the request sended by idle(not have group) students.

2.4.2.4 Sequence Diagram

2.4.2.4.1 Student Join Request
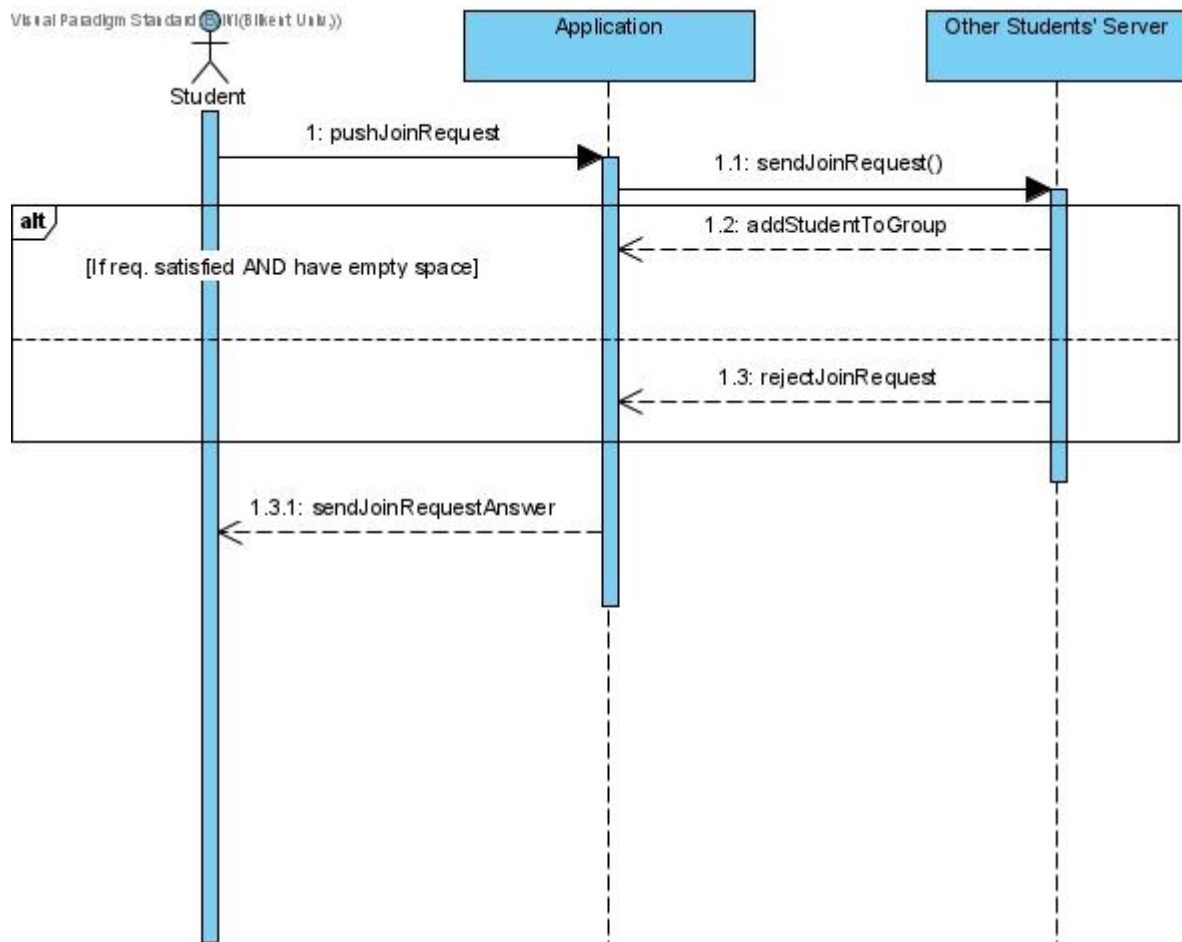
Figure 9: Sequence diagram showing the students' join request steps

- When a student pushes the Join Request button for a group, other groups'
  students see the request and can decide whether to add the student to the
  group or not.

2.4.2.4.2 Student Peer Review
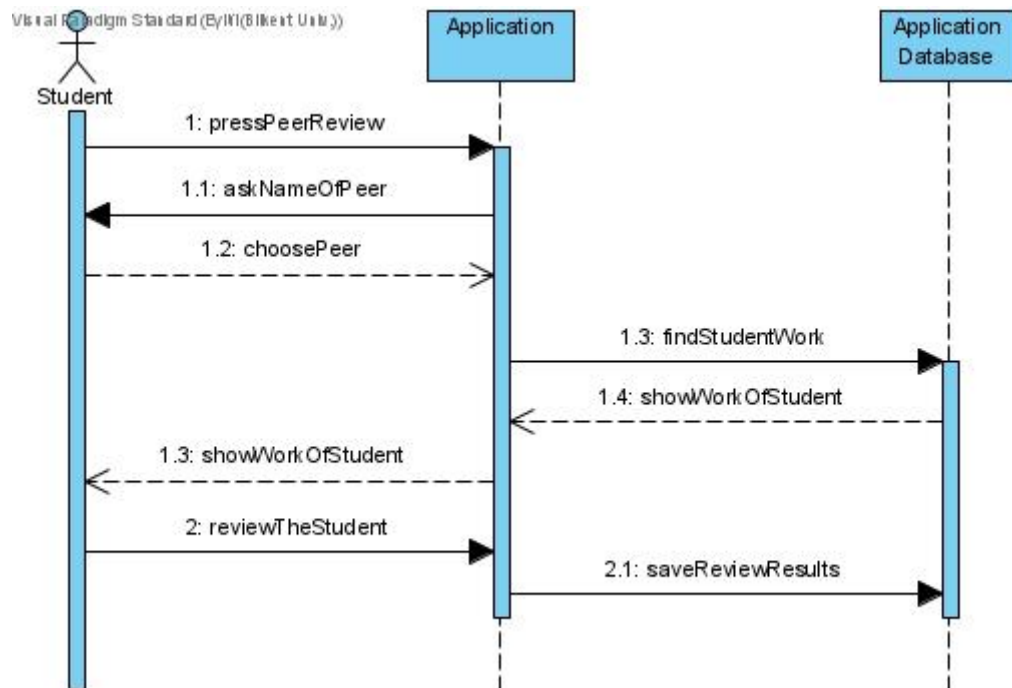
Figure 10: Sequence diagram showing that students' peer review steps

- When a student presses the Peer Review button, names of group members are displayed.
- Students can choose a group member to review.
- Works done by chosen group members will be displayed.
- After that the student can review the work.
- Review will be saved.

2.4.2.4.3 Student Group Review
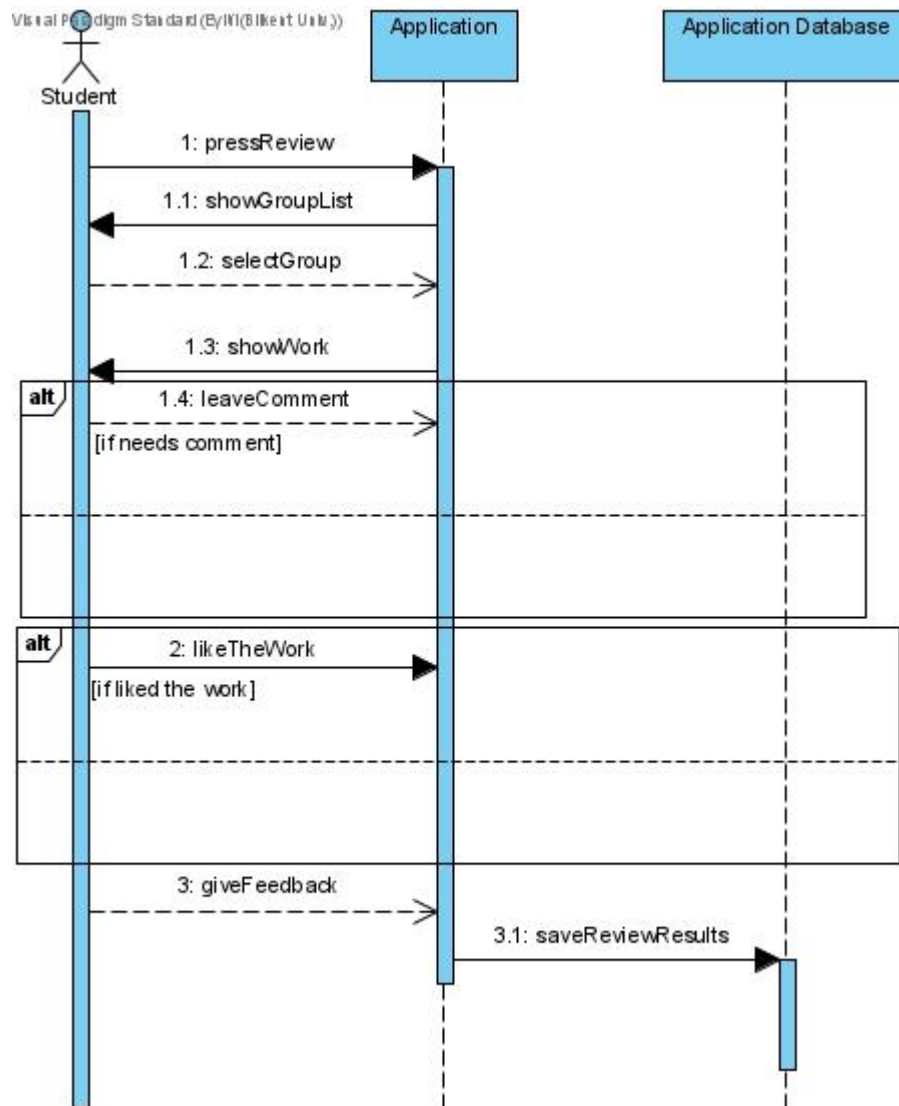
Figure 11: Sequence diagram showing that students' other groups reviewing steps

- When the Review button is pressed, the application gives a list of works done by other groups.
- Students can select a work which s/he has not reviewed yet.
- If a student feels that s/he should make a comment, leave a comment.
- If a student likes the work, can press the like button.
- Students may evaluate the work by using a feedback feature.
- All review steps will be saved when students press the Submit button.

2.4.2.4.4 Student Poll Creation
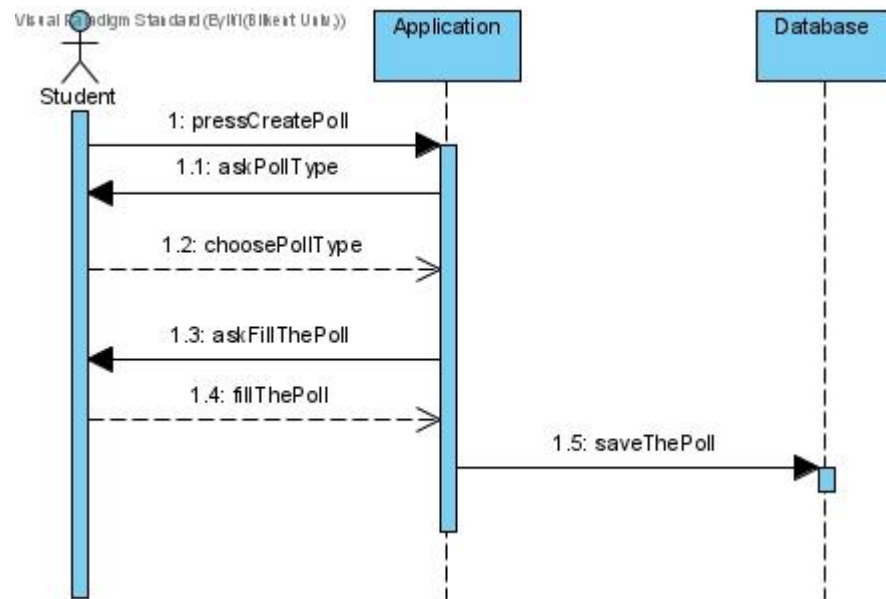
Figure 12: Sequence diagram showing that students' poll creation steps

- A student can press Create Poll and choose which type of poll to be used.
- After selecting the poll type, students can write the question and can easily save the poll. After then other group members can see the poll.

2.4.2.4.5 Instructor Evaluation Form Creation
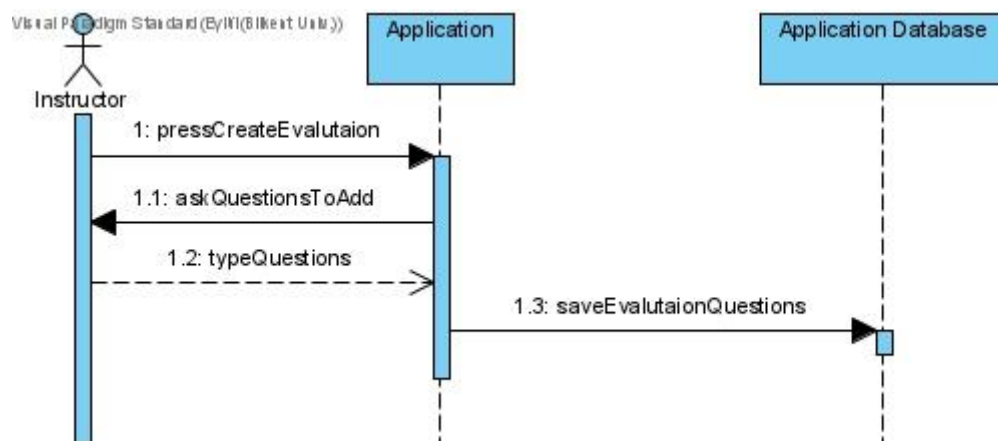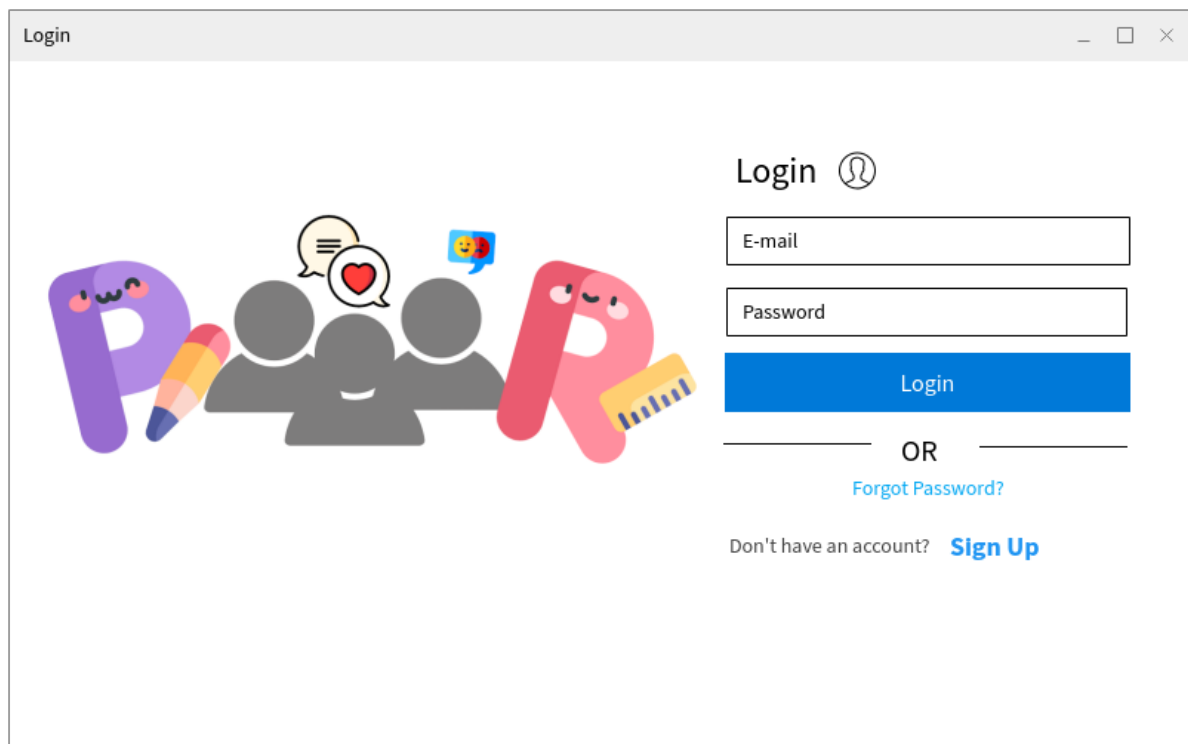
Figure 13: Sequence diagram showing that instructors' evaluation form creation steps.
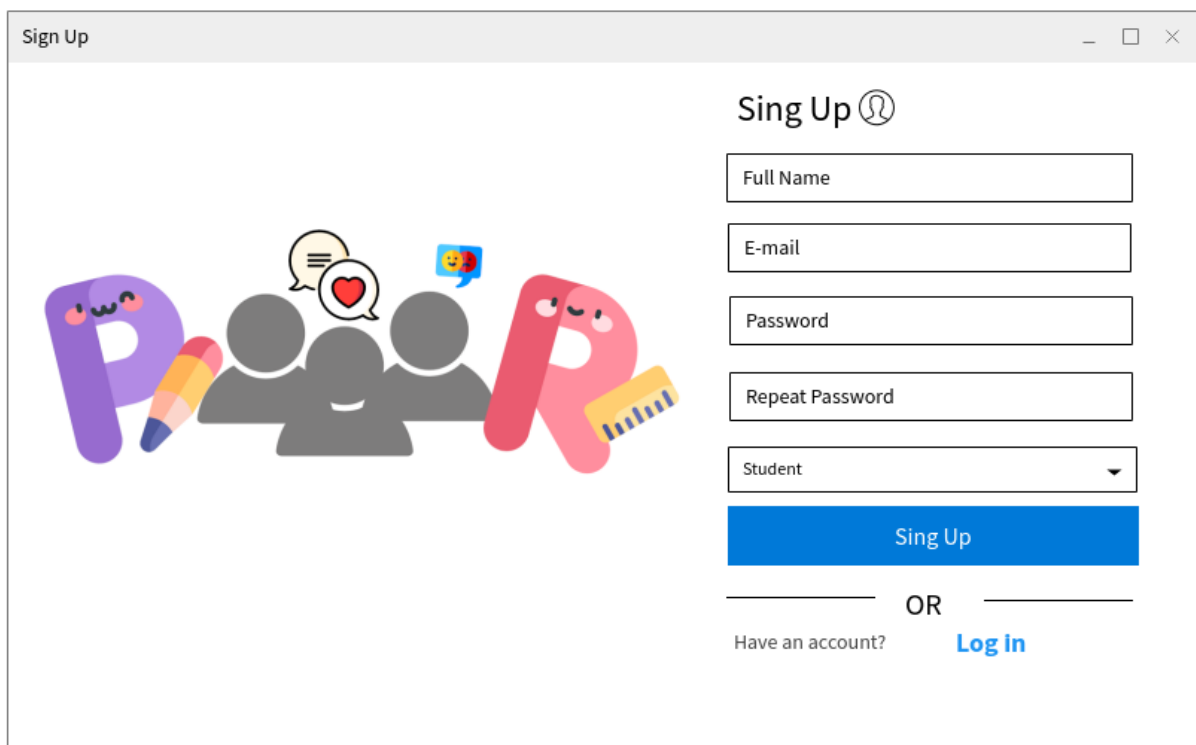
- Instructors can create an evaluation form by clicking Create Evaluation button.
- Button will enable instructors to type questions.
- Questions will be saved and can be seen by students.

2.4.4 User interface

Figure 14: Login page

Figure 15: Sign Up page

First Peer Review Form

Please update your assessments for yourself and your team members:

5: Excellent, 4: Good, 3 Fair, 2: Poor, 1: Very Poor

Yıldız Tilbe:

Contributes and functions effectively on a team: [5 ▼]

Helps creating a collaborative and inclusive team-work environment: [5 ▼]

Works with team mates to establish goals and plan tasks: [5 ▼]

Any Other Comments:

[   ]

Uzay Heparı:

Contributes and functions effectively on a team: [5 ▼]

Helps creating a collaborative and inclusive team-work environment: [5 ▼]

Works with team mates to establish goals and plan tasks: [5 ▼]

Any Other Comments:

[   ]

Sezan Aksu:

Contributes and functions effectively on a team: [5 ▼]

Helps creating a collaborative and inclusive team-work environment: [5 ▼]

Works with team mates to establish goals and plan tasks: [5 ▼]

Any Other Comments:

[   ]

Figure 15: Peer Review page

Figure 16: Poll

Figure 17: Poll

Figure 18: Chatl

Figure 19: Create Poll

Figure 20: Create Evaluation Form

Figure 21: Activity Creation page

Figure 22: Editing evaluation pages

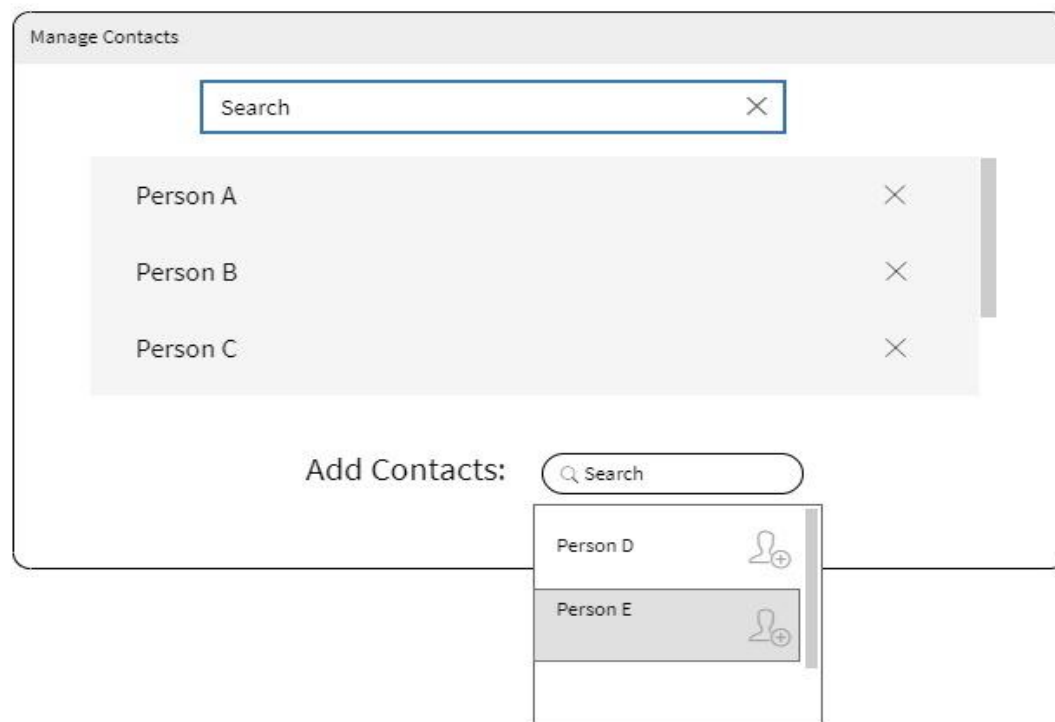**Figure 23:** Profile Page
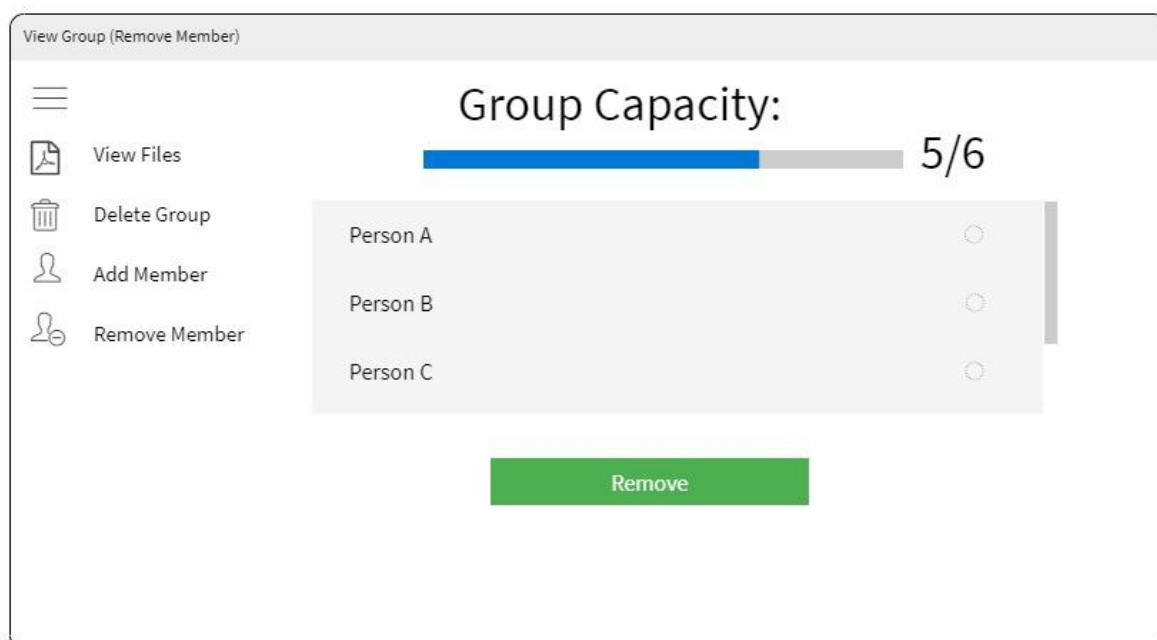
Figure 24: Manage contacts page

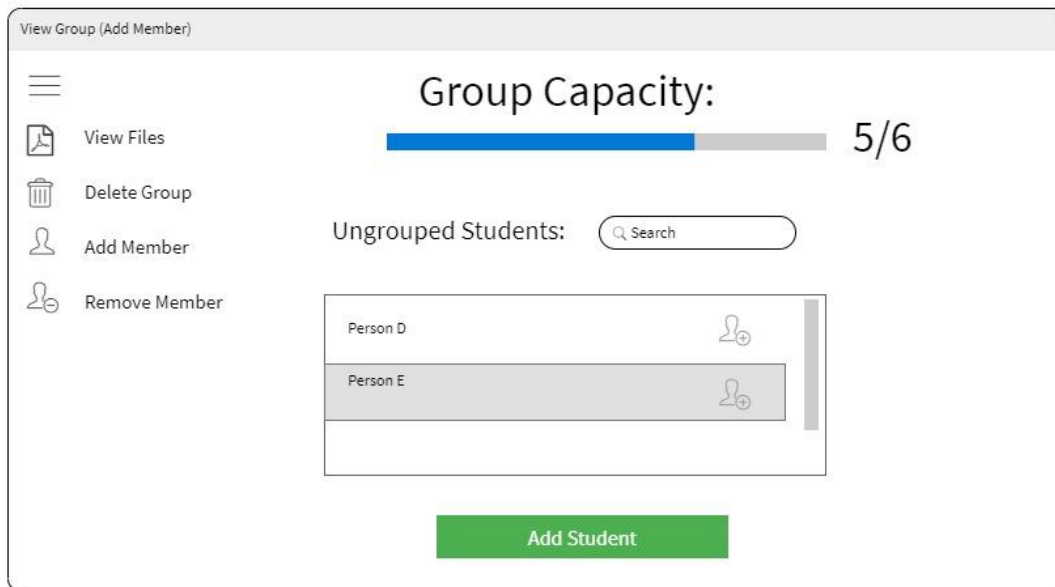**Figure 25:** Fire member page

Figure 26: Add member page

3. Glossary & references

[1] Produle, "Wireframe Tools, Prototyping Tools, UI Mockups, UX Suite, Remote designing," *MockFlow*. [Online]. Available: https://mockflow.com/. [Accessed: 16-Mar-2021].