CS-319 TERM PROJECT

Peer-Review

Analysis Report Final

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, a website which enables students to make peer review and form a group will be implemented. The application will be a web 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

- All users should be able to register for an application through the system. Users should register with provided TA code or Class code to register to the system and declare their account type.

2.1.2 User Profile

- All users should be able to configure his profile by setting their profile picture and can add additional information about his background, his previous projects and his abilities. So, groups can see his/her works and decide if they want his/her in their groups or not.

2.1.3 Groups

- Leader Student can manage their project groups and these groups are linked to their profiles and can be seen which projects the user is involved in. Groups can invite other students to join their groups through invitations, on the other hand ungrouped students can group join requests.

2.1.4 Contacts

- All users are able to add contacts to their contact list so they can easily send personal messages to each other.

2.1.5 Send Message

- All users can 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

- Students should be able to create groups.

2.1.8 Create Poll

- Students and the instructor can 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. For instance, if a group needs to decide a time to meet, creating a poll can be an easy way to see everybody's common free time.

2.1.9 Schedule

- Schedule includes both global and local activities where global means all class deadlines and local means all group deadlines. So, by seeing schedule all users can easily follow the course requirements, such as writing reports on time or online labs.

2.1.10 Exporting

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

2.1.11 Review And Drop Comments

- Peer review has two types, group peer review and teammate peer review.Students in each team can review each other based on question and consideration provided by a template which is created by instructors. For the peer review among other groups, students can review the artifacts of other groups once the review period starts.

2.1.12 To do list and Progress Bar

- Students 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.

2.1.13 Authentication Manager

- Users can instantly access the application without login every time they launch application tanks to the high level authentication manager if they logged in the system before.

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. Thanks to the supportability feature of this application, all users can easily access and use this peer review application without installing complex tools.

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. By using database all data will be accessible without using extra local storage and all data will be recovered from the database instantly. When database factor is considered, response time is important to maintain application efficiently. Therefore, the system is designed to have a maximum response time of less than 2 seconds. This ensures to keep the users’ attention in the application.

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

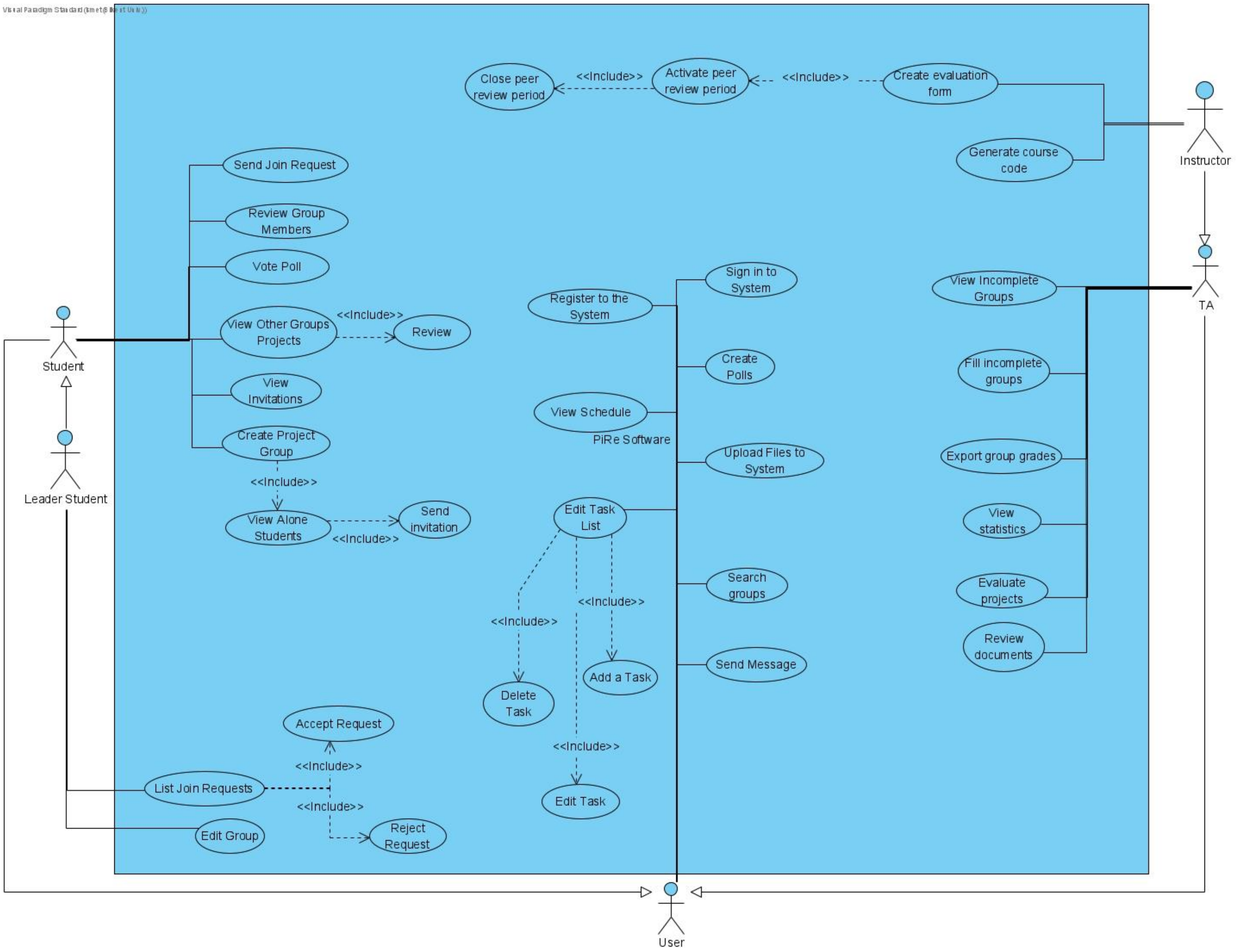


Figure 1: Use Case Diagram for Pire Software

Use Case Name: Vote Poll
Participating actors: Student
Flow of Events:

1. The student sees the “Polls” button on the computer screen.
2. The student clicks the “Polls” button.
3. Poll answer page is open
4. The student fills questions.
5. The student pushes the “Submit” button.
6. The answer is saved to the database.

Entry Conditions: Student chooses poll to vote.
Exit Conditions: Student votes poll or close poll without vote.

Use Case Name: View Invitations
Participating actors: Student
Flow of Events:

1. The student clicks the “Invitations” button.
2. List of invitations which are sent directly to the particular student will be shown.
3. If a student wants to join the group which has sent the invitation and if the student does not have any group yet the student will click the “Join” button. Else can click the “Reject” button.
4. If a student clicks the “Join” button, the case ends or a student closes the page, the case ends.

Entry Conditions: Student clicks invitations button and views invitations.
Exit Conditions: Student accepts invitation if s/he has no group or closes the page.

Use Case Name: Create Project Group
Participating actors: Student
Flow of Events:

1. A student clicks the “Create Group” button.
2. Create a group page opens.
3. A student fills the required information.
4. Student click the “Submit New Group“ button.
5. New group is saved to the database.
6. The student who created the group is added to as a group member.

Entry Conditions: Student clicks the create group button if s/he has no groups and creates groups.
Exit Conditions: Student creates group or closes the page.

Use Case Name: View Alone Students
Participating actors: Student
Flow of Events:

1. A student clicks the “List Alone Student” button.
2. The student redirected to the alone students list page.

Entry Conditions: After a student creates the group s/he can view the alone students or s/he can view these stıdents any time.
Exit Conditions: Student closes the page.

Use Case Name: Search Groups
Participating actors: Student
Flow of Events:

1. A student sees the groups list.
2. The students can see the list of unfilled groups.

Entry Conditions: Student clicks group billboard page.
Exit Conditions: Student closes the group billboard page.

Use Case Name: Send Join Request
Participating actors: Student
Flow of Events:

1. A student choses a group.
2. The student inspects the group
3. The student clicks the “Send Request” button.
4. Request sent.

Entry Conditions: Student choses a group to send a join request to the group if s/he has no group.
Exit Conditions: Students send requests to groups or exit without sending requests.

Use Case Name: View Incomplete Groups
Participating actors: TA or Instructor
Flow of Events:

1. TA or instructor clicks to “View Incomplete Groups”.
2. TA or instructor can see all incomplete groups in one page.
3. If TA or instructor closes the page, use case ends.

Entry Conditions: TA or instructor clicks to “View Incomplete Groups”.
Exit Conditions: TA closes the page.

Use Case Name: Edit Task List
Participating actors: Student, Instructor or TA
Flow of Events:

1. The student clicks the “Edit” button.
2. Edit task list page is open.
3. The student can fill or change the task attributes.
4. The student clicks the update task.
5. The task is updated in the database.

Entry Conditions: Student clicks the edit to do button.
Exit Conditions: Student edits the task or closes the page.

Use Case Name: Add a Task
Participating actors: Student, Instructor or TA
Flow of Events:

1. A student clicks the add task button.
2. The add task page is open.
3. The student fills the required fields.
4. The student clicks the “Add New Task” button.
5. New task is saved to the database.

Entry Conditions: Student clicks the add task button.
Exit Conditions: Student adds task or closes the page.

Use Case Name: Delete a Task
Participating actors: Student, Instructor or TA
Flow of Events:

1. A student clicks the task button.
2. The task is deleted from the database.

Entry Conditions: Student clicks the delete task button.
Exit Conditions: Student deletes the task or closes the page.

Use Case Name: Edit Task
Participating actors: Student, Instructor or TA
Flow of Events:

1. A student clicks the edit task button.
2. The student edits the task.
3. The student submits a task.
4. The task is edited.

Entry Conditions: Student clicks the edit task button.
Exit Conditions: Student closes the page or completes editing successfully.

Use Case Name: View Other Groups Projects
Participating actors: Student
Flow of Events:

1. Student view project groups list.
2. A student choses a group.
3. The student inspects the chosen group’s project.
4. The student closes the page.

Entry Conditions: Student chooses other groups’ projects to view.
Exit Conditions: Student closes the page.

Use Case Name: Review
Participating actors: Student
Flow of Events:

1. A student views the project.
2. The student types his/her evaluation.
3. The student clicks the “Submit” button.
4. The evaluation is saved.

Entry Conditions: Student reviews the project that was viewed.
Exit Conditions: Student closes the page.

Use Case Name: Review Group Members
Participating actors: Student
Flow of Events:

1. A student goes to the Group Page.
2. The student chooses a group member to review.
3. The student answers questions about review.
4. The student submits the review.

Entry Conditions: Student clicks the “Group Page” button.
Exit Conditions: Student submits the review.

Use Case Name: List Join Request
Participating actors: Leader Student
Flow of Events:

1. Leader Student clicks the “List Joint Requests” button.
2. Leader Student sees all requests on the screen.

Entry Conditions: Leader Student clicks the “Group Page” button.
Exit Conditions: Student submits the review.

Use Case Name: Approve Request
Participating actors: Leader Student
Flow of Events:

1. Leader Student clicks the “Approve” button.

Entry Conditions: Leader Student clicks the “Approve” button.
Exit Conditions: No condition.

Use Case Name: Reject Request
Participating actors: Leader Student
Flow of Events:

1. Leader Student clicks the “Reject” button.

Entry Conditions: Leader Student clicks the “Reject” button.
Exit Conditions: No condition.

Use Case Name: Edit Group
Participating actors: Leader Student
Flow of Events:

1. Leader Student clicks the “Edit Group” button.
2. If the Leader Student wants to delete a student, s/he selects that particular member and clicks the “Delete” button. Else closes the page.

Entry Conditions: Leader Student clicks the “Edit Group” button.
Exit Conditions: Leader Student closes the page.

Use Case Name: Sign In to System
Participating actors: Student, TA or Instructor
Flow of Events:

1. A student clicks the “Sign In” button.
2. The student enters required information.
3. The students click the “Sign In” button.
4. The student is registered to his&her main page.

Entry Conditions: Actor clicks to sign in button.
Exit Conditions: Actor signs in successfully or closes the page.

Use Case Name: Register to the System
Participating actors: Student, Instructor or TA
Flow of Events:

1. User clicks the “Register” button.
2. User enters name, surname, ID, email and password.
3. User clicks the “Register” button.

Entry Conditions: Actor clicks the register button.
Exit Conditions: Actor registers successfully or closes the page.

Use Case Name: Create Polls
Participating actors: Student, TA or Instructor
Flow of Events:

1. A student, TA or instructor clicks the “Create New Poll” button.
2. Poll page is seen by the user who clicks the button.
3. Type of polls will be listed and one of them will be chosen by the user.
4. Questions and possible answers will be typed by the user.
5. The user clicks the “Submit” button.
6. The user closes the page.

Entry Conditions: Actor clicks the “Create New Poll” button.
Exit Conditions: Actor closes the page.

Use Case Name: Upload Files to System
Participating actors: Student, TA or Instructor
Flow of Events:

1. User clicks the “Upload File” button.
2. User selects the file that s/he wants to upload.
3. User clicks the “Submit” button.
4. User closes the page.

Entry Conditions: Actor clicks the upload document button.
Exit Conditions: Actor uploads the document or closes the page.

Use Case Name: Send Message
Participating actors: Student, TA or Instructor
Flow of Events:

1. A user choses a contact to send a message.
2. The user writes a message.
3. The user clicks the send button.

Entry Conditions: Actor choses contact to send message.
Exit Conditions: Actor closes message page.

Use Case Name: View Schedule
Participating actors: Student, Instructor or TA
Flow of Events:

1. User clicks the “Schedule” button.
2. User sees the “Schedule” on the page.

Entry Conditions: Instructor clicks close peer review period button.
Exit Conditions: Instructor closes the page.

Use Case Name: Fill Incomplete Group
Participating actors: TA, Instructor
Flow of Events:

1. An actor clicks the “Fill Incomplete Groups” button.
2. Students with no groups are filtered and arranged with incomplete groups.
3. System completes the filling process.

Entry Conditions: TA clicks fill incomplete groups.
Exit Conditions: System completes filling process.

Use Case Name: View Statistic
Participating actors: TA
Flow of Events:

1. A TA clicks the “View Statistic” button.
2. The TA redirected to the view static page.
3. The TA views the statistics.
4. The TA closes the page.

Entry Conditions: TA clicks view statistic button.
Exit Conditions: TA closes the statistics page.

Use Case Name: Evaluate Projects
Participating actors: TA or Instructor
Flow of Events:

1. TA or instructor choose a project from the project lists.
2. The actor adds the mark and the feedback.
3. The actor submits the feedback and the mark by pushing the “Submit“ button.
4. Mark and feedback is saved in the database.

Entry Conditions: Actor choses project to evaluate it.
Exit Conditions: Actor closes the page or completes evaluation successfully.

Use Case Name: Review Documents
Participating actors: TA or Instructor
Flow of Events:

1. The actor pushes the “View” button.
2. View document page is open.
3. The actor can view the document.

Entry Conditions: Actor choses document to review.
Exit Conditions: Actor closes the page or completes review successfully.

Use Case Name: Generate Course Code
Participating actors: Instructor
Flow of Events:

1. An instructor clicks the “Generate Course Code” button.
2. The instructor enters the course code.
3. The instructor clicks the “Submit” button.
4. New course code is saved to the database.

Entry Conditions: Instructor clicks create course code button.
Exit Conditions: Instructor closes the page.

Use Case Name: Create Evaluation Form
Participating actors: Instructor
Flow of Events:

1. An instructor clicks the “Create Evaluation From” button.
2. The instructor redirected to create an evaluation form page.
3. The instructor enters the evaluation form name.
4. The instructor enters the questions.
5. The instructor clicks the “Submit Evaluation Form” button.
6. The evaluation form is saved to the database and students are informed about the evaluation form.

Entry Conditions: Instructor clicks create evaluation form button.
Exit Conditions: Instructor closes the page or completes creating form.

Use Case Name: Activate Peer Review Period
Participating actors: Instructor
Flow of Events:

1. An instructor clicks the activate peer review period.
2. Students are informed about the peer review period.
3. The instructor closes the page.

Entry Conditions: If peer review is not activated yet and instructor clicks activate peer review period button.
Exit Conditions: Instructor clicks close peer review period button.

Use Case Name: Close Peer Review Period
Participating actors: Instructor
Flow of Events:

1. An instructor clicks the close peer review period.
2. Students are informed that the review period is over.
3. The instructor closes the page.

Entry Conditions: Instructor clicks “Close Peer Review Period” button.
Exit Conditions: Instructor closes the page.

2.4.2. Dynamic Models

2.4.2.1 Activity Diagrams

2.4.2.1.1 Authentication Activity Diagram

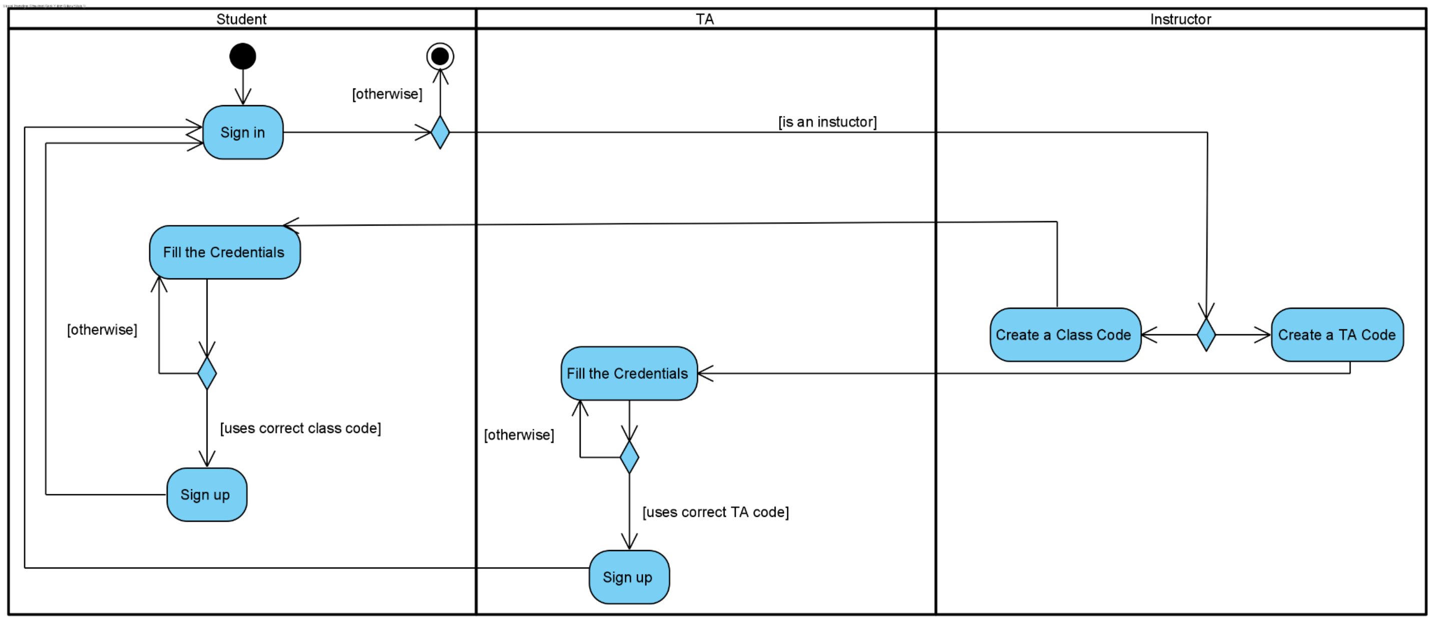


Figure 2: Activity diagram of authentication process.

Flow of Activity: User signs in. If the user is an instructor, the instructor has given two options, creating a class code or creating a TA code. Class code helps students to create an account. Meanwhile, TA Code enables the user to promote herself/himself to a TA. In the case of an incorrect code process is repeated until correct credentials are provided by the user.

2.4.2.1.2 Group Management Activity Diagram

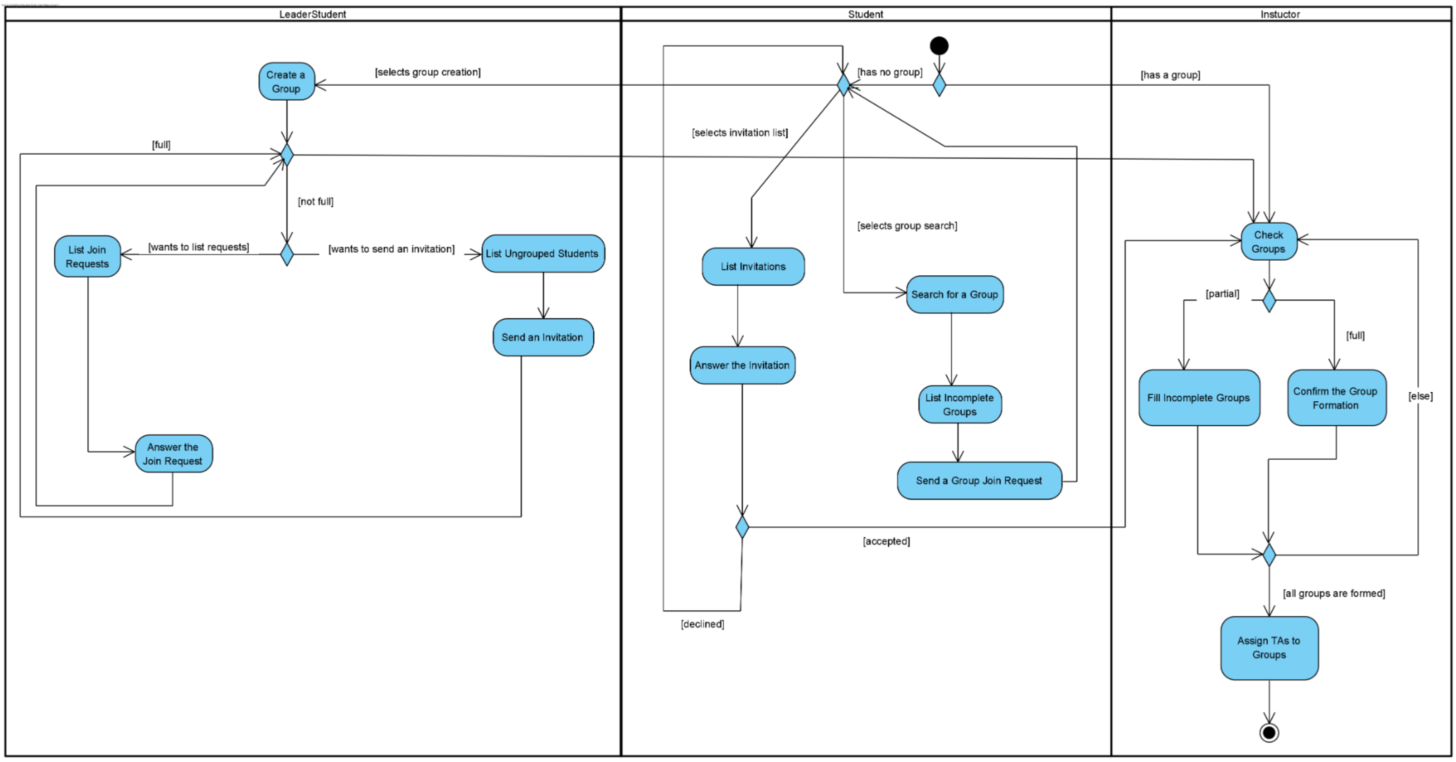


Figure 3: Activity Diagram of group management issues.

Flow of Activity: Student has two possible actions on the start of the flow. Students with a group require no further activity for students. On the other hand, students without a group either search for a group, view the list of incoming invitations from groups or create her/his own group. After sending a group request, students can send further requests to the other groups. If there are incoming invitations from groups, users can accept or decline invitations. Creating a group, changing actor student to a leader student. Leader Student can list incoming join requests or invite a student to his/her group. Ultimately, instructors assign TAs to the groups after filling incomplete groups or confirm the group formation.

2.4.2.1.3 Peer Reviewing Activity Diagram

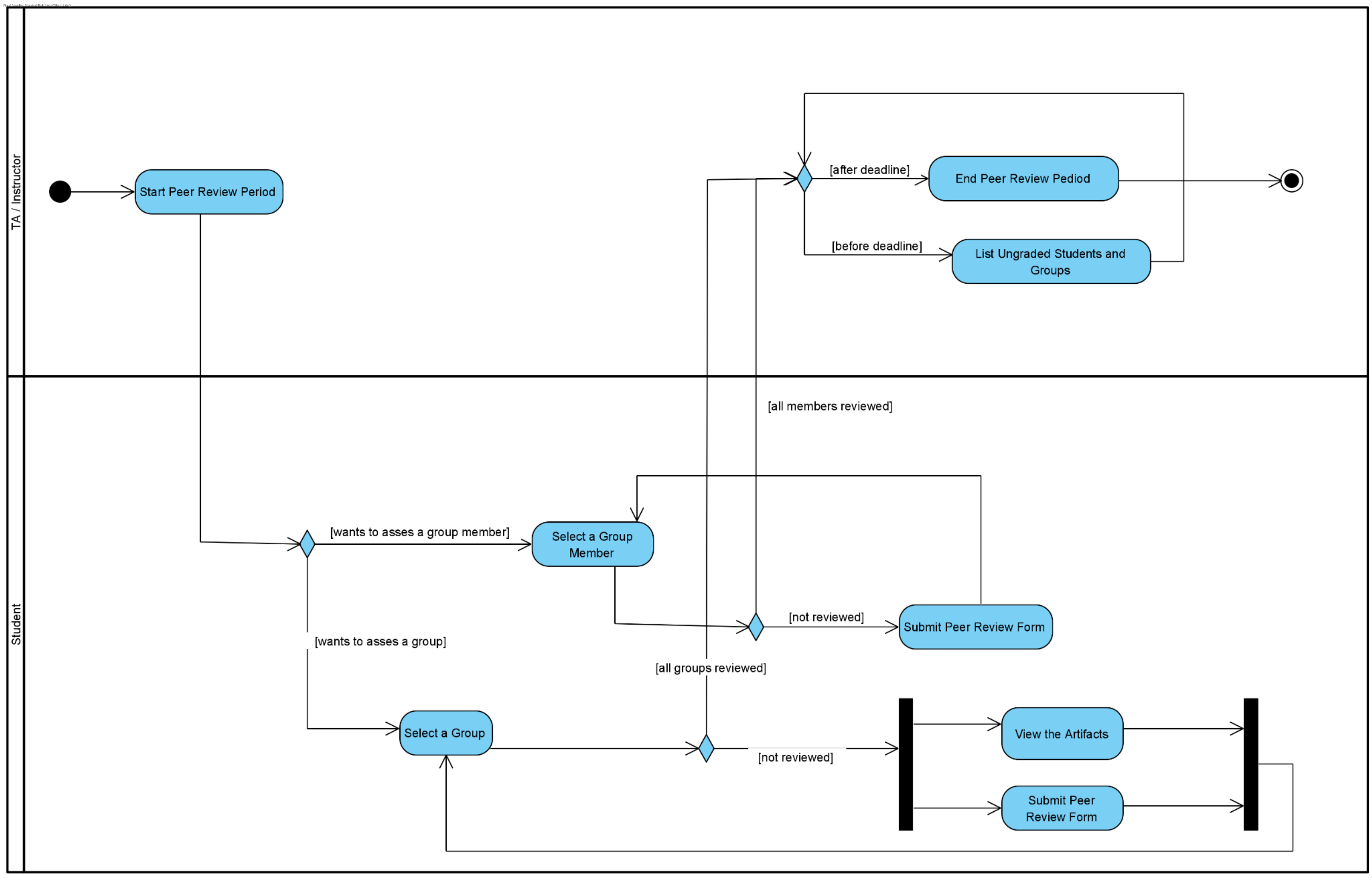


Figure 4: Activity diagram of peer review process.

Flow of Activity: Instructor starts peer reviewing period. Students either choose to assess a group member or another group from the class. If a student chooses to assess a group and it is not reviewed, students are able to view the artifacts of the chosen group and submit peer review form concurrently. On the other hand, If the student chooses to review a group member, the student selects a team member to review and submit the peer review form. If the review period is over, the instructor ends the review period after all the members and teams are reviewed. Otherwise the instructor lists ungraded students and groups.

2.4.2.1.4 Group Poll Management

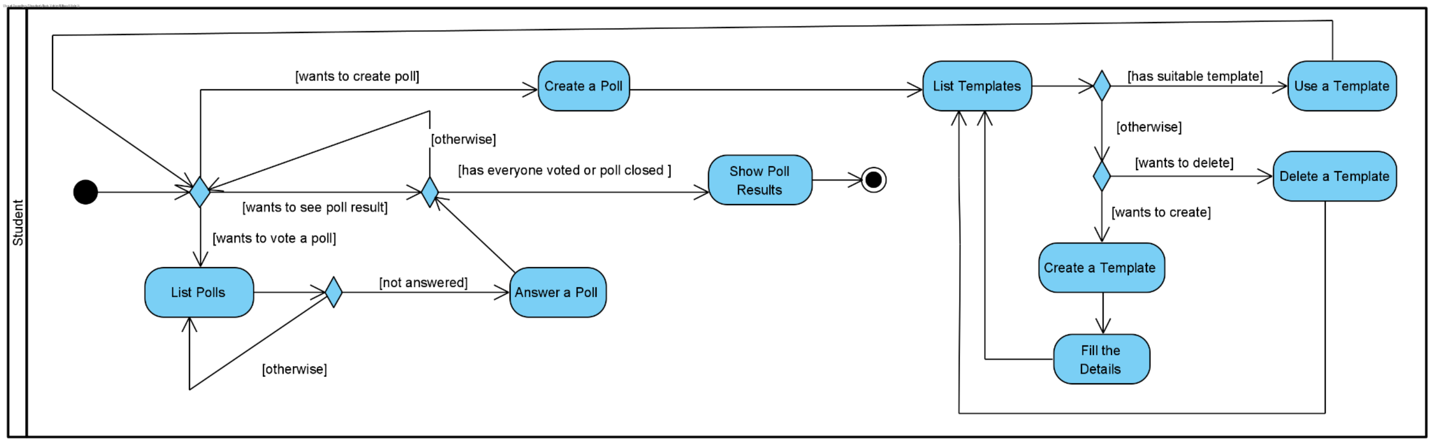


Figure 5: Activity diagram of in group poll management.

Flow of Activity: Students can either create a poll or list current polls to vote. For creating a poll, students list available templates if there is no proper template use, and students can create a template or delete a template. Ultimately, a template is chosen to use. Poll creators and other members can vote for the poll. After the poll is closed, students can display the poll result.

2.4.2.1.5 Global Poll Management

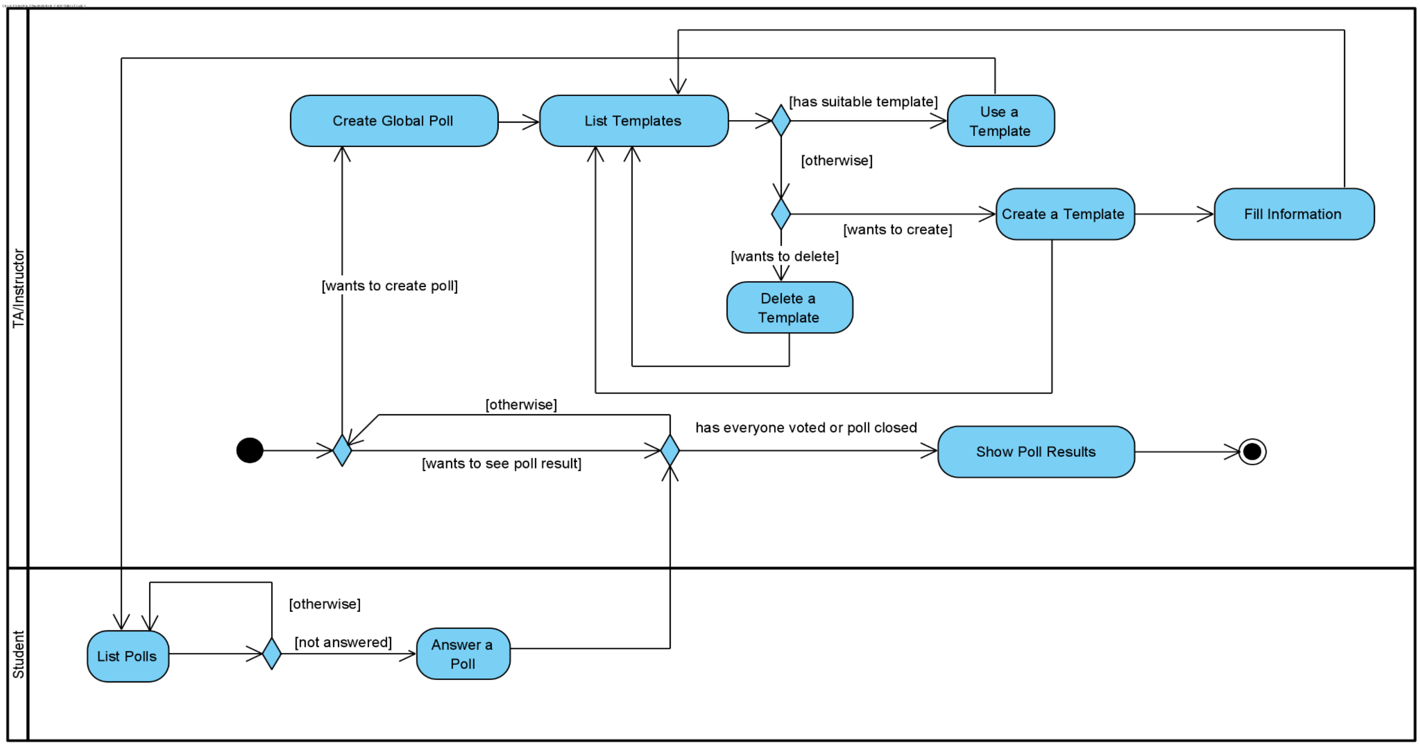


Figure 6: Activity diagram of global poll management.

Flow of Activity: Instructor creates a global poll. For creating a poll, instruction list available templates if there is no proper template use, and the instructor can create a template or delete a template. Ultimately, a template is chosen to use. After creation of a poll, students can list available polls and can vote for the poll. After the poll is closed or everyone is voted, an instructor can display the poll result.

2.4.2.1.6 Task Management

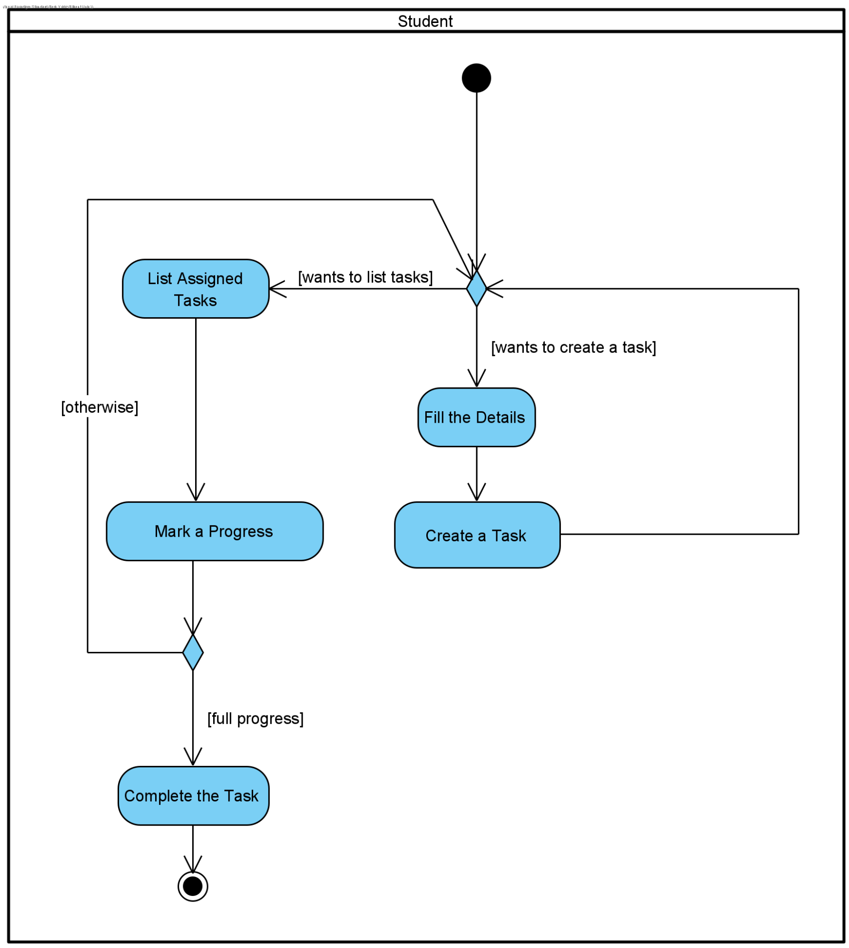


Figure 7: Activity diagram of task management.

Flow of Activity: Students either create a task or list current tasks assigned to him/her. If a student chooses to create a task, the student fills the details for the task then creates the task. On the other hand, students can view the assigned tasks and mark a progress. If the task has complete progress, students can complete the task.

2.4.2.2 State Diagram

2.4.2.2.1 Authentication State Diagram

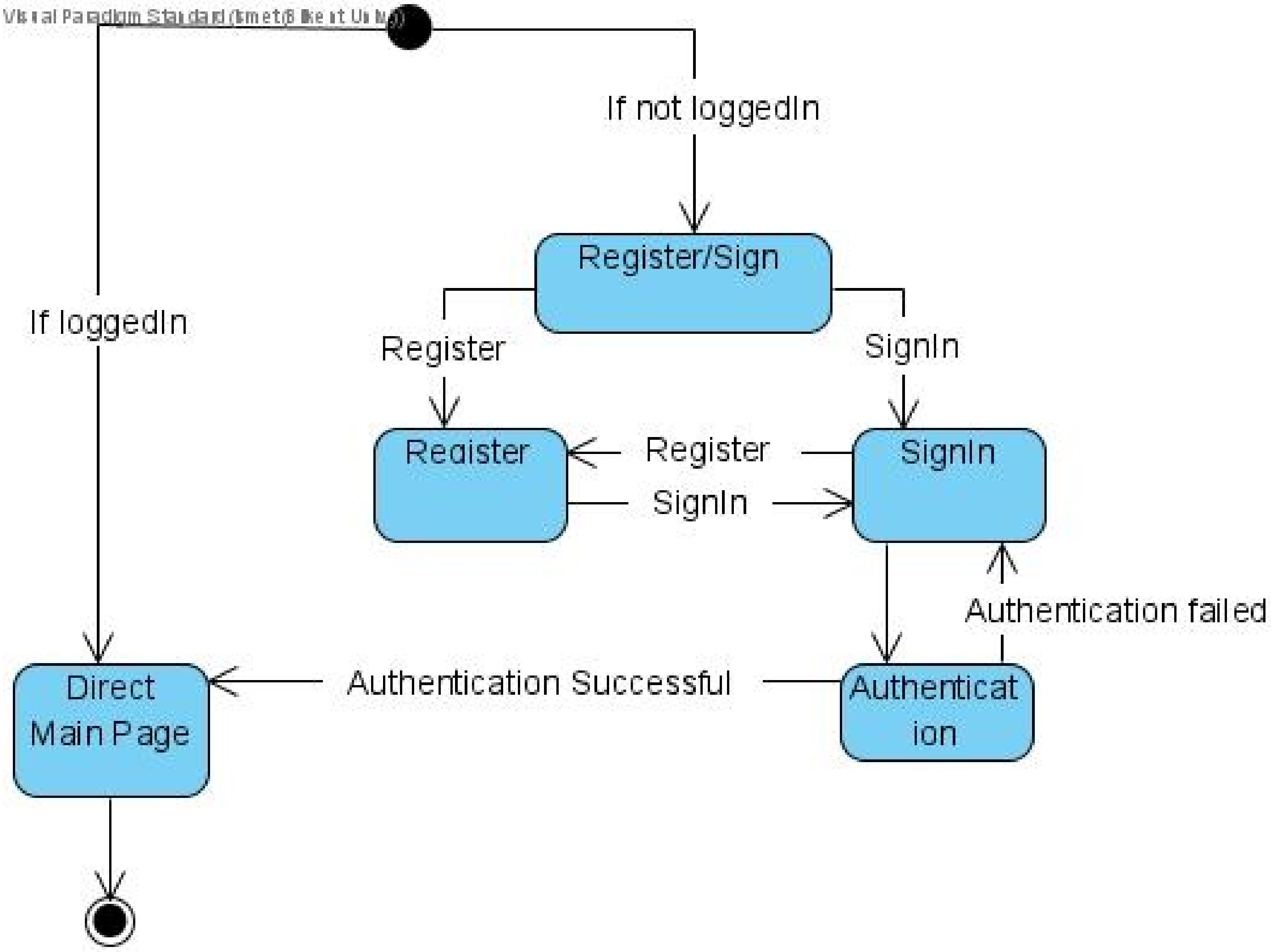


Figure 8: State Diagram showing the authentication states and transitions between them.

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 Group Formation State Diagram

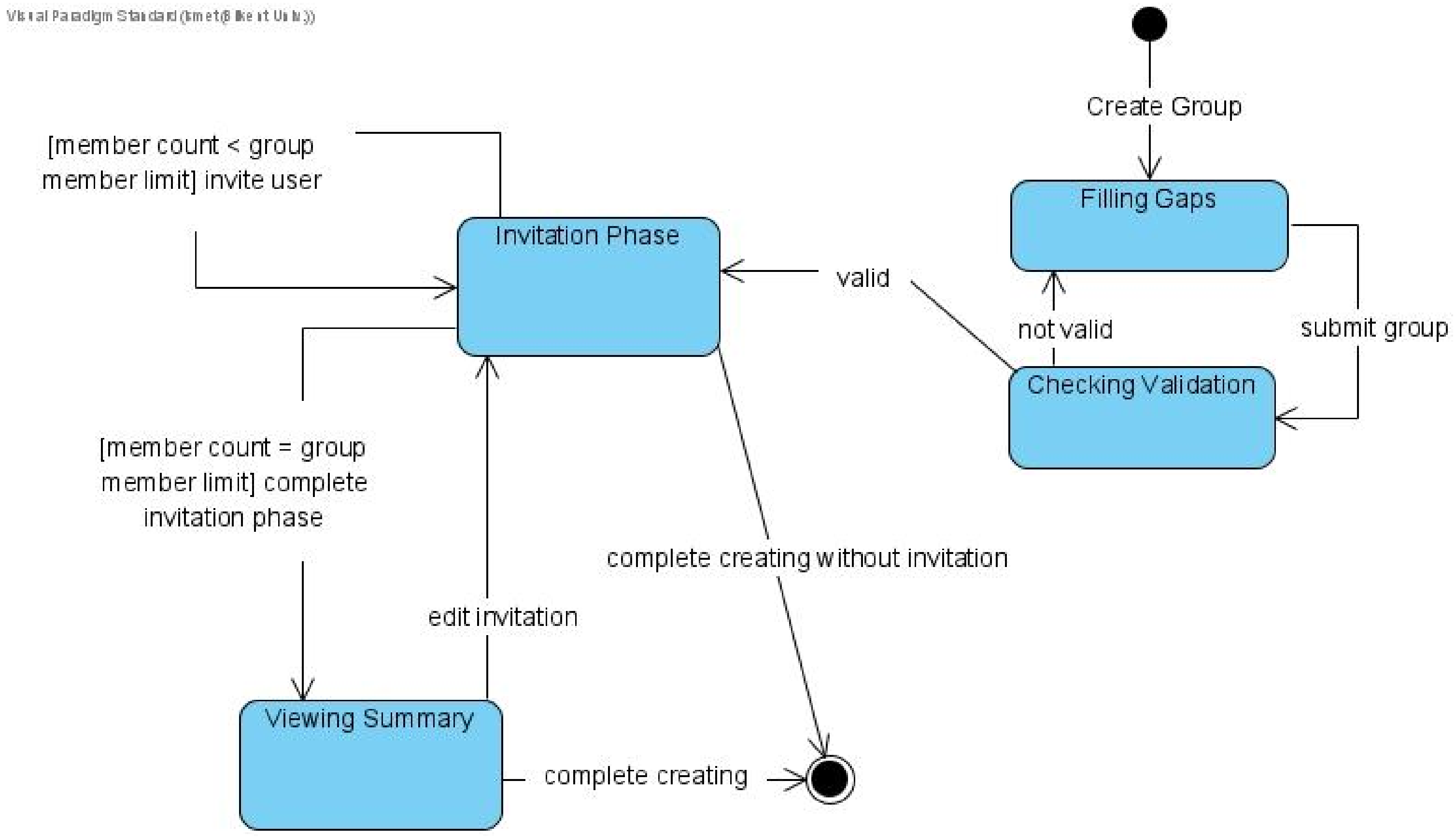

Visual Paradigm Standard(Ismet(Bilkent Univ.))
Create Group
Filling Gaps
submit group
Checking Validation
not valid
valid
Invitation Phase
[member count < group member limit] invite user
[member count = group member limit] complete invitation phase
edit invitation
complete creating without invitation
Viewing Summary
complete creating

Figure 9: State Diagram showing the group formation states

Filling Gaps: Student will fill the required gaps to create a group and then students will submit a form and will be directed to the **Checking Validation** state.

Checking Validation: System will check the inputs and if inputs are not valid then system will show error message and direct student to **Filling Gaps** state again until inputs will be valid. If inputs are valid, then student will be directed to the **Invitation Phase**.

Invitation Phase: Student will be able to invite other students to the newly created group until … number of students invited to the group. After the necessary number of students are invited and decide to complete the process or decide to complete the process without inviting other students, student will be directed to the **View Summary** state.

View Summary: The student will be able to complete the group formation process or return to the Invitation **Phase** again to edit invitations.

2.4.2.2.3 Group Evaluation State Diagram

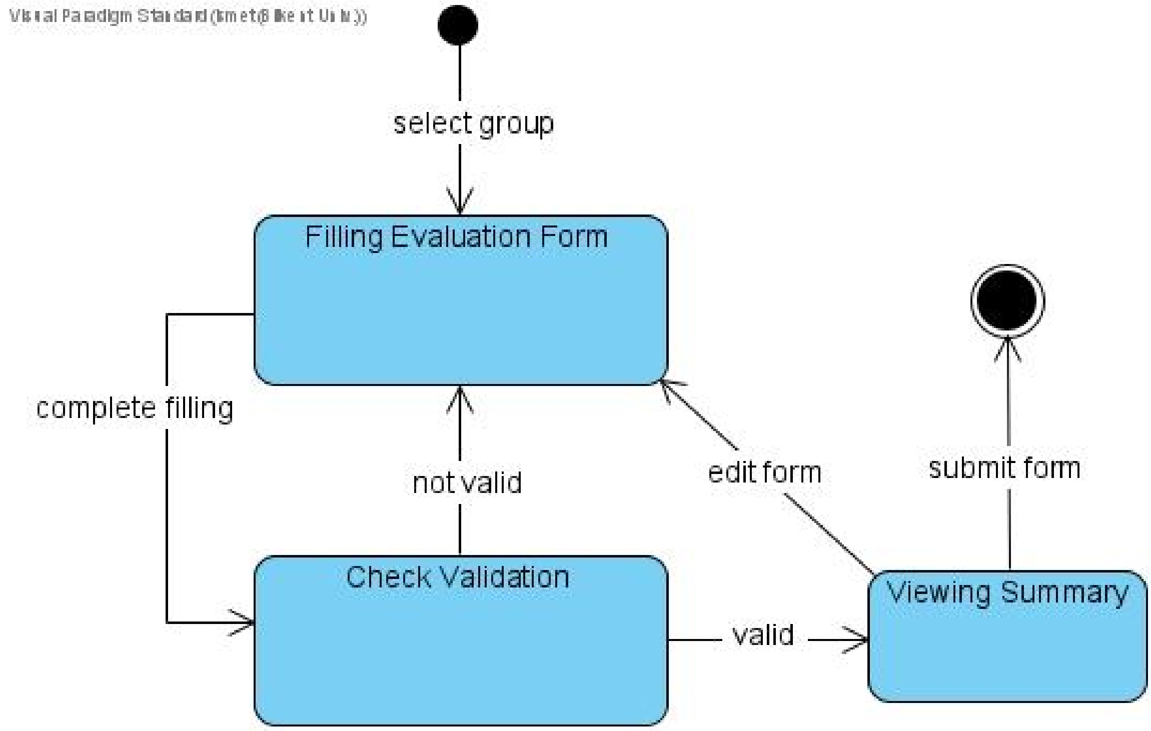


Figure 10: State Diagram showing the group evaluation states.

Filling Evaluation Form: User will fill the required gaps to submit evaluation form and then user will submit form and will be directed to the **Checking Validation** state.

Checking Validation: System will check the inputs and if inputs are not valid then system will show error message and direct student to **Filling Evaluation Form** state again until inputs will be valid. If inputs are valid, then student will be directed to the **Viewing Summary** state.

Viewing Summary: User will be able to view summary of the evaluation form and if wants to edit it will be able to return **Filling Evaluation Form** state. If the form is okay, then the user will submit the form.

2.4.2.2.4 Group Evaluation State Diagram

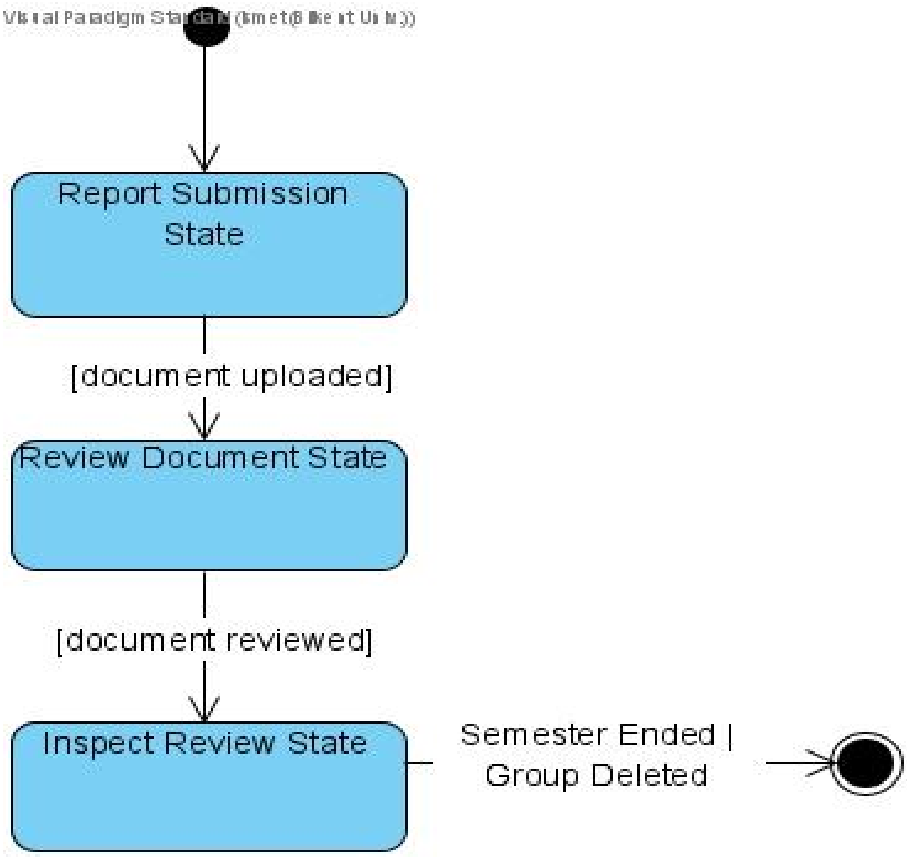


Figure 11: State Diagram showing the document review states.

Report Submission State : Report submission state is started by the instructor. In this state project groups can upload their report documents.

Review Document State : If a group uploaded their document, It remains on hold for the instructor's or TA's review.

Inspect Review State : After the document is reviewed, the group could inspect the feedback given to that document.

End point : If a group gets deleted or the semester is ended, document and review cannot be inspected anymore.

2.4.2.2.5 Create Poll State Diagram

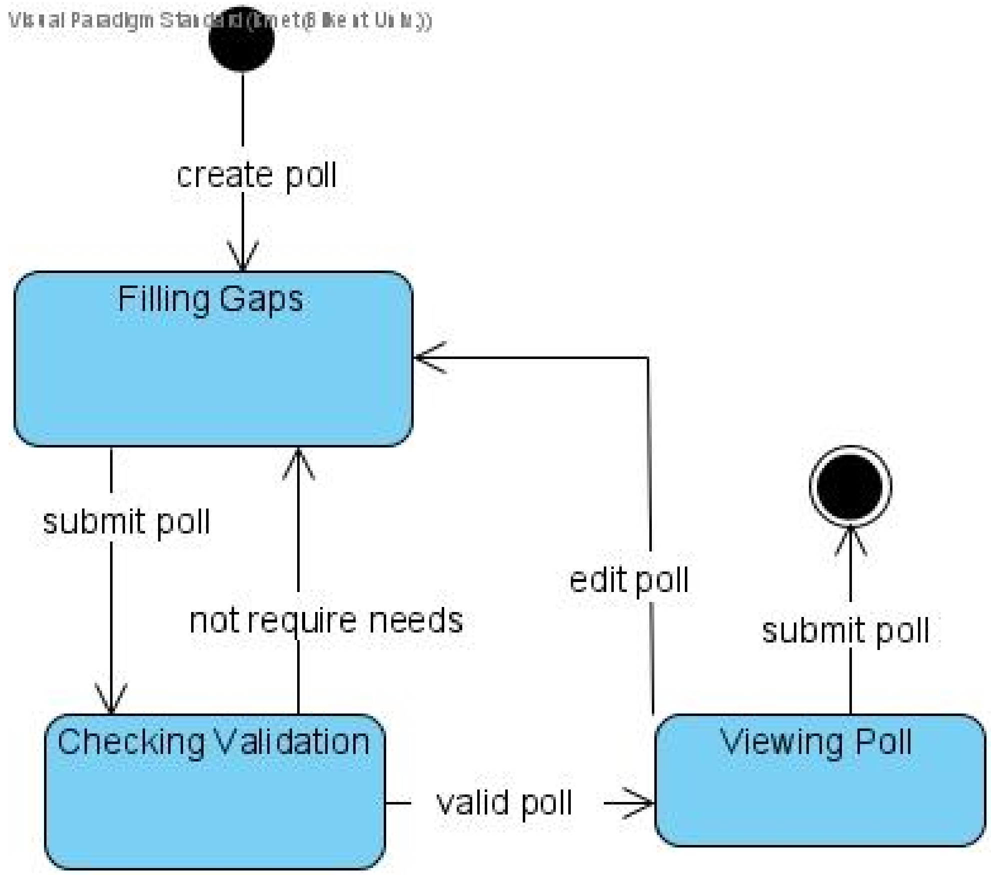


Figure 12: State Diagram showing the poll creation states.

Filling Gaps: User will fill the required gaps to create a poll and then the user will submit the poll and will be directed to the **Checking Validation** state.

Checking Validation: System will check the inputs and if inputs are not valid then system will show error message and direct student to **Filling Gaps** state again until inputs will be valid. If inputs are valid, then student will be directed to the **Viewing Poll** state.

Viewing Poll: User will be able to view a summary of the evaluation form and if wants to edit it will be able to return **Filling Gaps** state. If the form is okay, then the user will submit the form.

2.4.2.3 Class Diagram

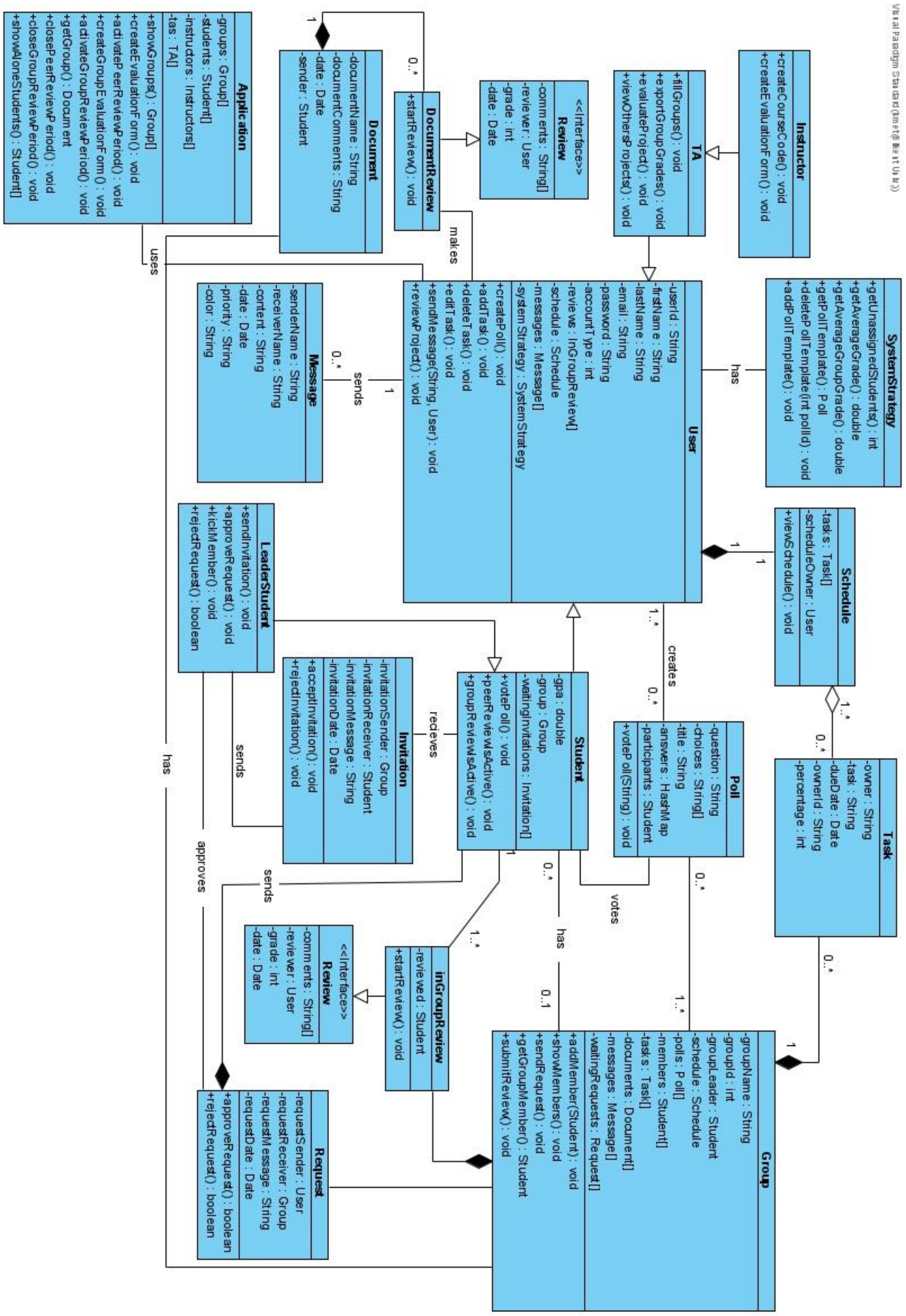


Figure 13: Class Diagram for Pire

User: There are four types of account; Student, Leader student, TA, Instructor. The user has personal information such as (name, password, email etc.) and other properties shown above.

Student: Students have additional properties; studentId, phone number, gpa etc. Phone number only visible by some people(group members and instructor) for urgency.

Leader Student: Type of student but has more functionality than regular student.

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

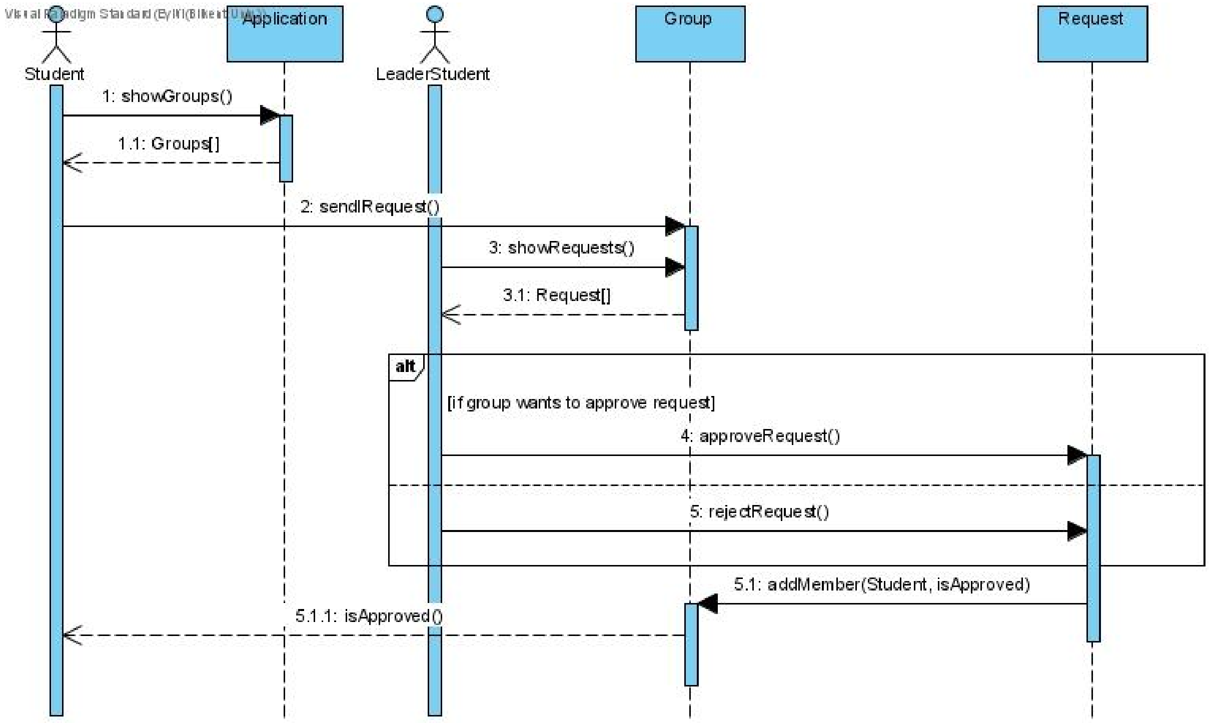


Figure 14: Sequence diagram showing the students' join request steps

- Students can see the list of groups and select a group.
- When a student wants to join a group s/he pushes the “Send Join Request” button, other groups' Leader Student can see the request and can decide whether to add the student to the group or not.

2.4.2.4.2 Student Peer Review

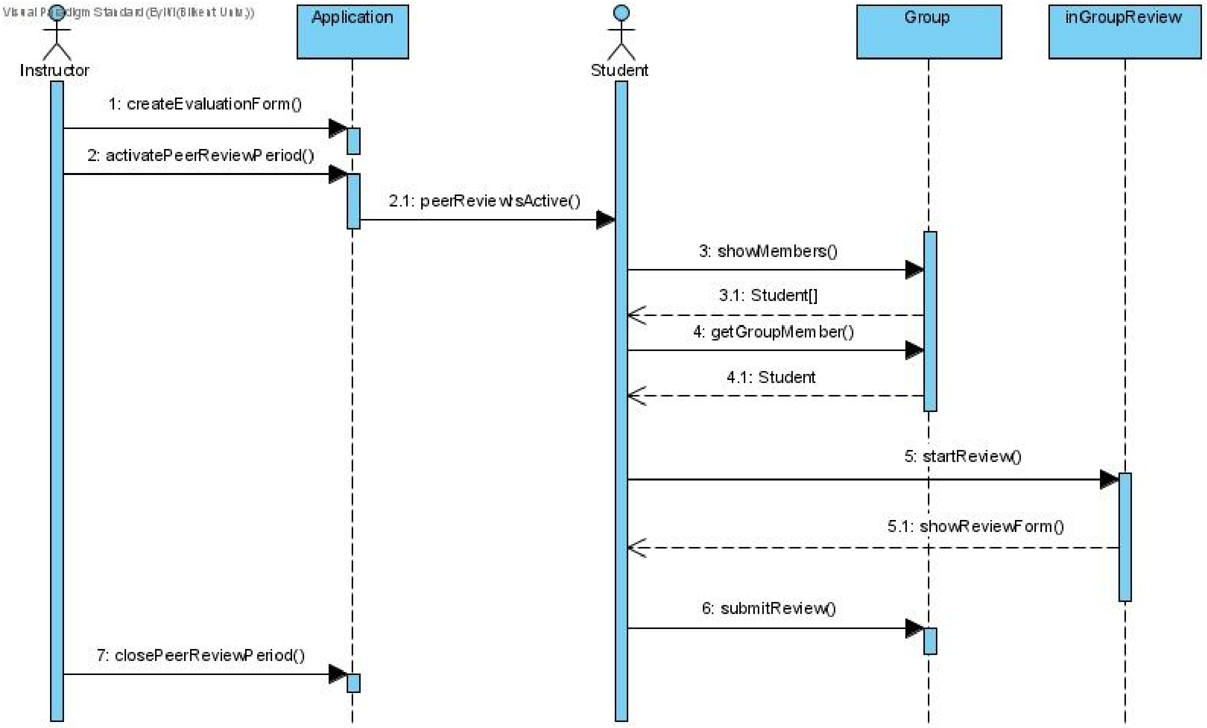


Figure 15: Sequence diagram showing that students' peer review steps

- Instructor creates an evaluation form.
- Instructor activates "Peer Review Period".
- Student selects the peer to review.
- Student submits his/her review depending on the evaluation form.

2.4.2.4.3 Student Group Review

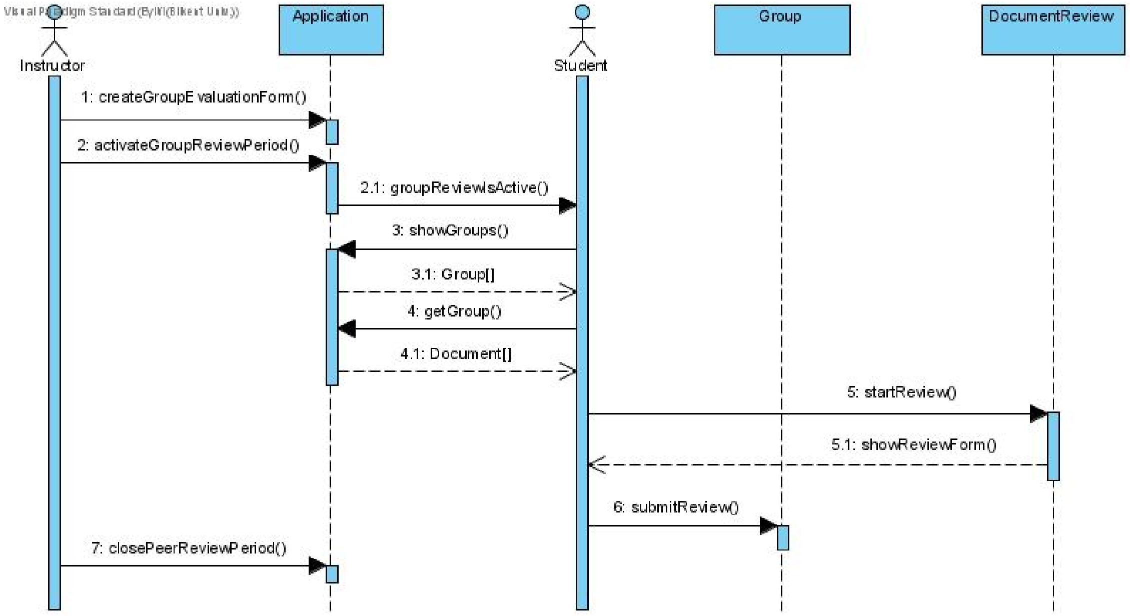


Figure 16: Sequence diagram showing that students' other groups reviewing steps

- Instructor creates a group evaluation form.
- Instructor activates "Group Review Period".
- Student selects a group to review.
- The student sees the documents of the particular group.
- Student submits his/her review depending on the evaluation form.

2.4.2.4.4 Group Invitation

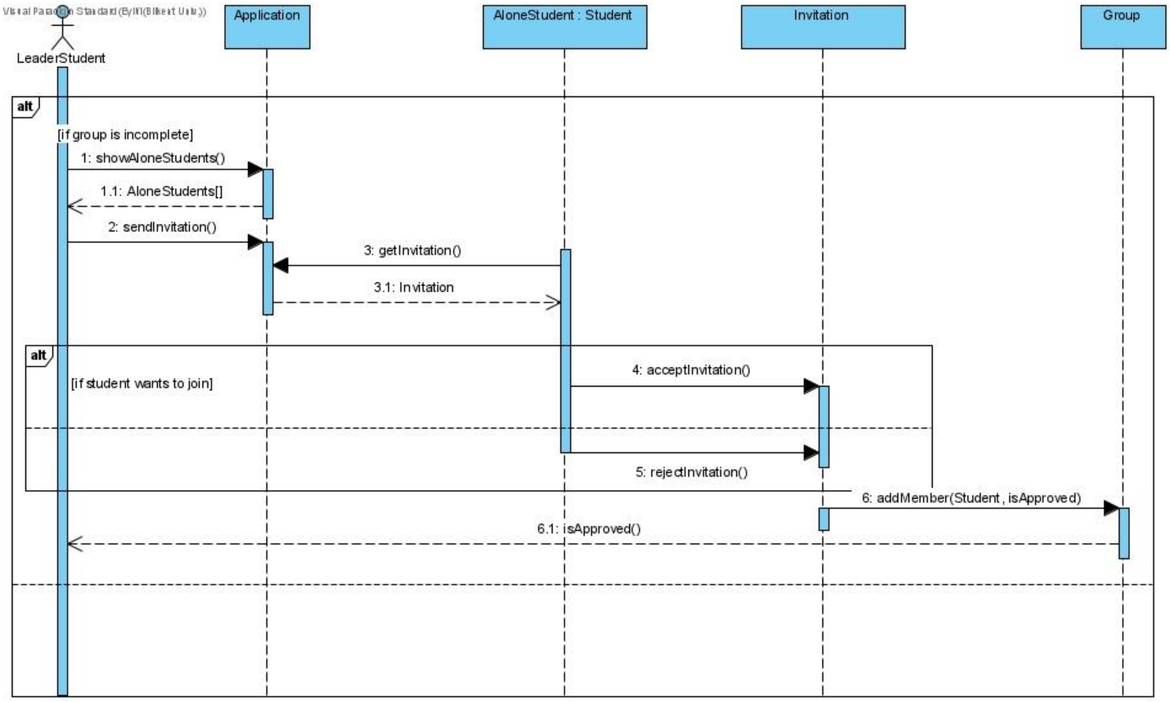


Figure 17: Sequence diagram showing that groups invitation steps

- A Leader Student can see the list of alone students.
- If a student wants to invite s/he to their group, send an invitation.
- Students' answers is seen by the Leader Student by isApproved() function.

3. User Interface

3.1 Login Page

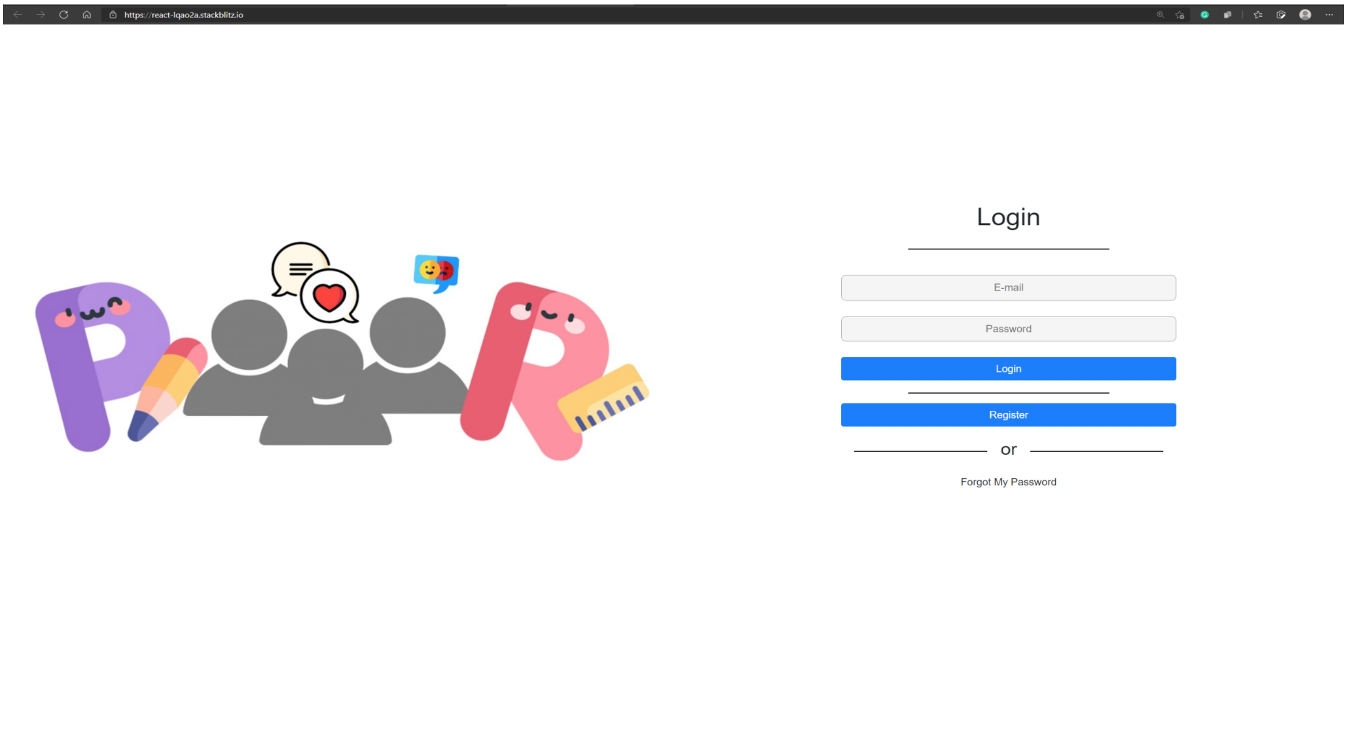


Figure 18: Login Page

3.2 Register Page

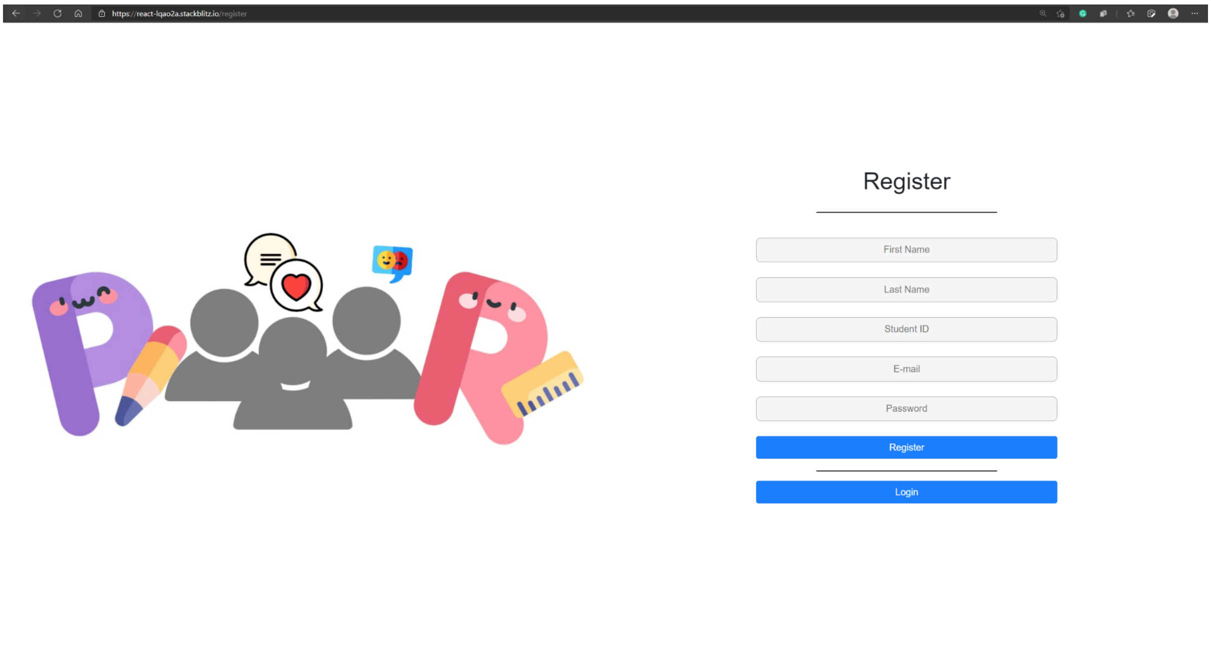
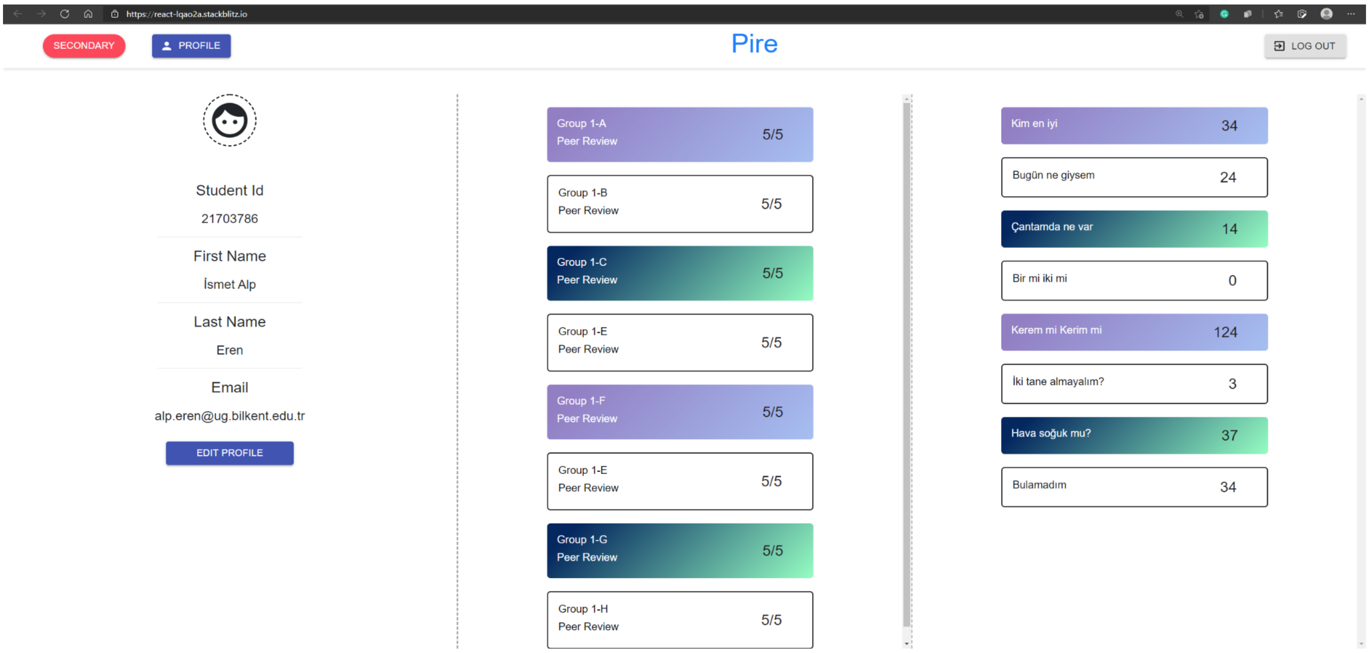


Figure 19: Register Page

3.3 Home Page



https://react-lqao2a.stackblitz.io

SECONDARY

PROFILE

Pire

LOG OUT

Student Id

21703786

First Name

İsmet Alp

Last Name

Eren

Email

alp.eren@ug.bilkent.edu.tr

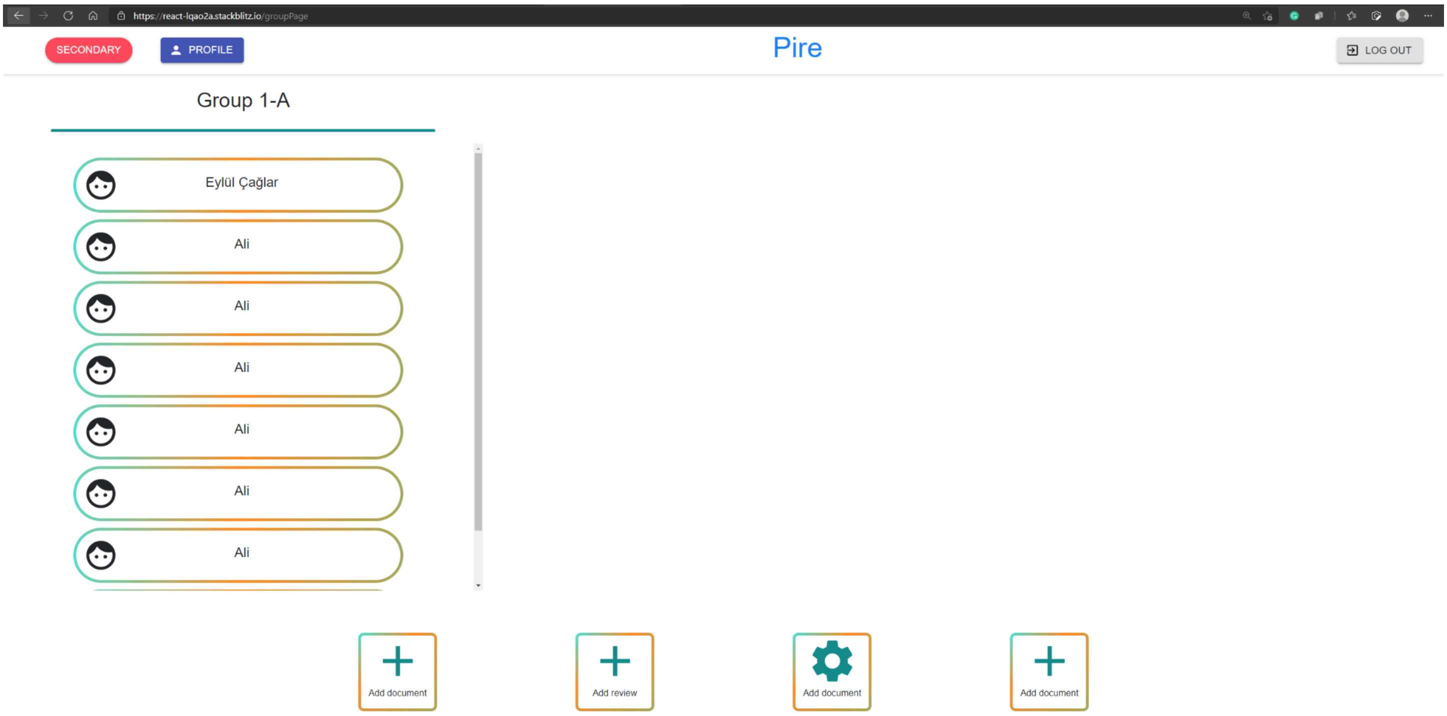
EDIT PROFILE

Group 1-A Peer Review	5/5
Group 1-B Peer Review	5/5
Group 1-C Peer Review	5/5
Group 1-E Peer Review	5/5
Group 1-F Peer Review	5/5
Group 1-E Peer Review	5/5
Group 1-G Peer Review	5/5
Group 1-H Peer Review	5/5

Kim en iyi	34
Bugün ne giysem	24
Çantamda ne var	14
Bir mi iki mi	0
Kerem mi Kerim mi	124
İki tane almayalım?	3
Hava soğuk mu?	37
Bulamadım	34

Figure 20: Home Page

3.4 Group Page



https://react-lqao2a.stackblitz.io/groupPage

SECONDARY

PROFILE

Pire

LOG OUT

Group 1-A

Eylül Çağlar

Ali

Ali

Ali

Ali

Ali

Ali

Add document

Add review

Add document

Add document

Figure 21: Group Page

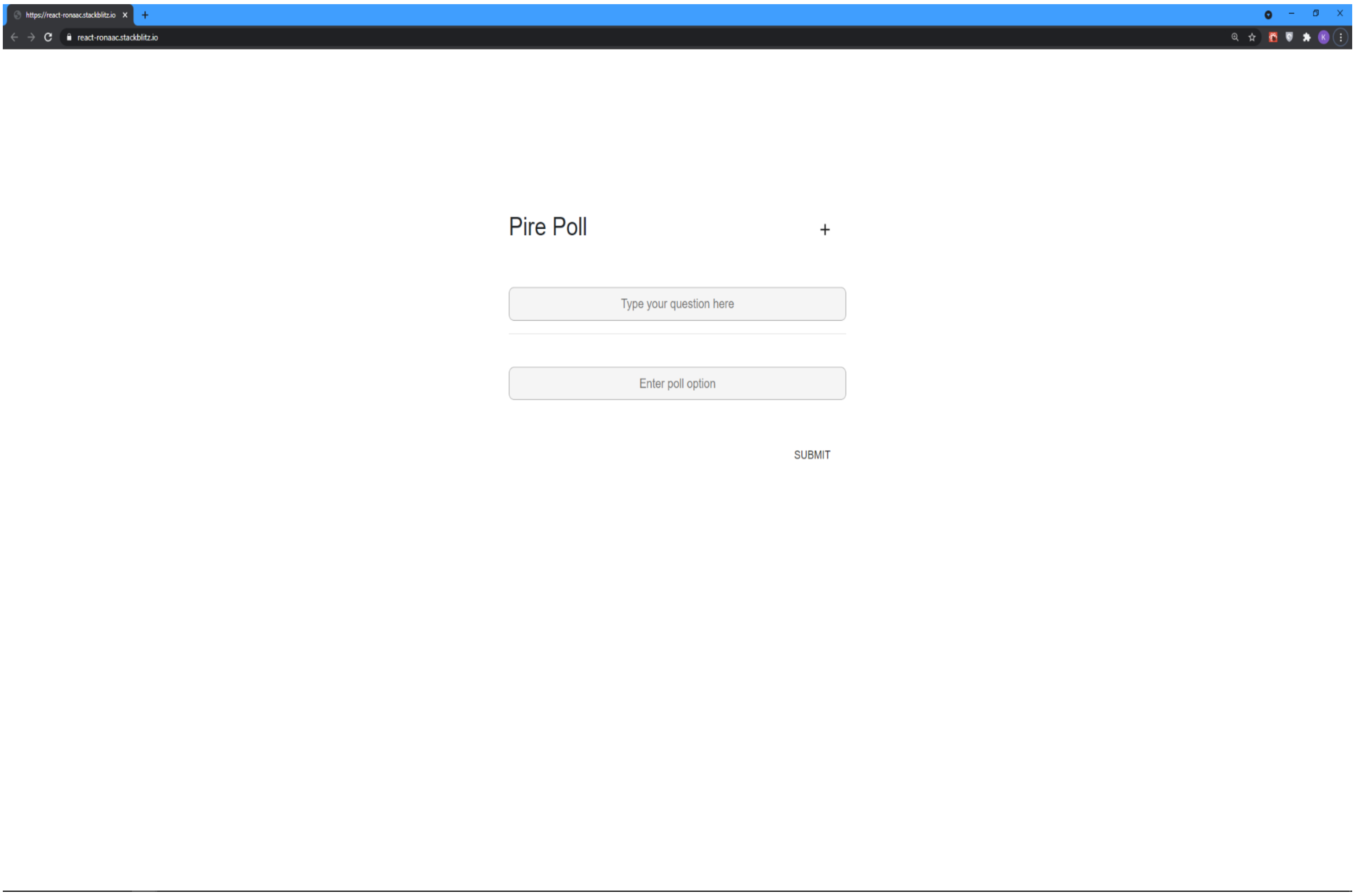

https://react-ronaac.stackblitz.io
react-ronaac.stackblitz.io
Pire Poll
+
Type your question here
Enter poll option
SUBMIT

Figure 22: Poll Creation Page

4. Glossary & references

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