CS 319
OBJECT ORIENTED SOFTWARE ENGINEERING
FINAL REPORT
PEER-REVIEW

GROUP 1D

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

1.0 INTRODUCTION

Design of the project Pire was built on making a classroom helper for CS319 lessons. First and most important goals were creating a web page which helps students to create their project groups and at the end of the semester review their group members. It is also considered that reviewing other groups' works should have importance.

Main design goals are reached in this project. Pire software enables students to create a group, join a group, remove from a group in case people may change their minds while creating their groups. Also software helps instructors to see groups and people in those groups. It is also possible that the instructor can start a peer review period. When the peer review period starts by the instructor the “Peer Review” button becomes available to be seen by students and students can give points and add comments on their group members. These reviews are only available to be seen by the instructor. The instructor also can end the peer review period. It is possible to add project related document links to be easily found by the instructor. These links are available to be seen in the group page. It is also possible to make reviews to other groups every individual file which are uploaded by them. Reviews of documents are visible by everybody.

In addition to these features, it is also possible to create polls by the instructor and answer poll questions by the students. So, the feature makes it easy to get feedback from students.

It was planned to add poll functions inside the groups to decide their meeting times or project related features. But since it did not have higher priority and time was not enough, it was not implemented.

It was planned to add a schedule function to help students project deadlines but time was not enough to implement it.

2.0 LESSONS LEARNT

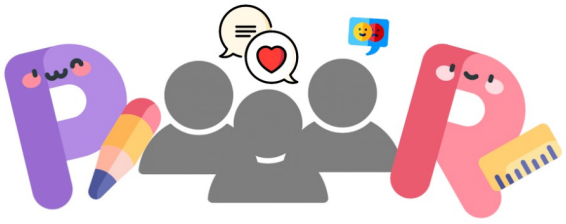
During this project design and implementation stages it is observed that the design part has a huge importance in the project. Making good design is the one of the hardest parts. Since implementation should depend on the design, every case should be planned before and this makes the design part even harder. Good design gives easier and fast implementations where lacking design make it harder to implement.

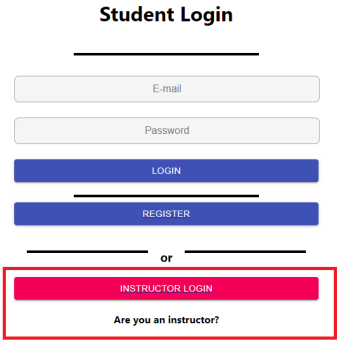
Also the importance of work allocation is observed in the project. In real software engineering life it is known that work allocation will have more importance since everybody cannot do every job. It is also experienced in this project. The project was between a very small project and a small project, nevertheless work allocation contributed to time saving. In this project everybody had the different interests and professionalism for separate

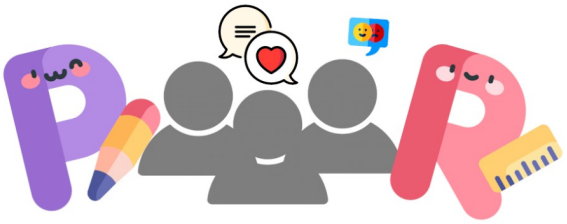
implementation parts, thus these interests and professionalisms are taken into consideration while making work allocation.

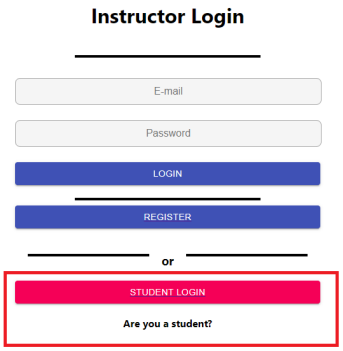
3.0 USER'S GUIDE

- Pire includes 2 types of users. If the user is a student, s/he can register and login by using the first page. If the user is an instructor then the user should use Instructor Login button and can register and login by this page.

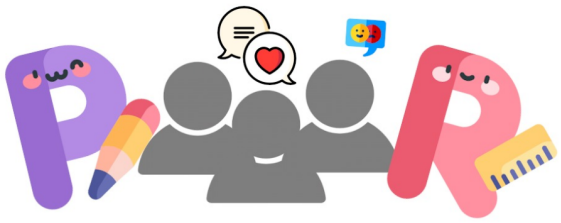


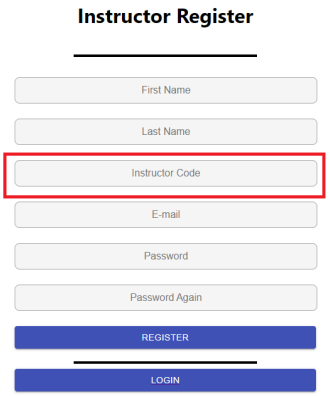




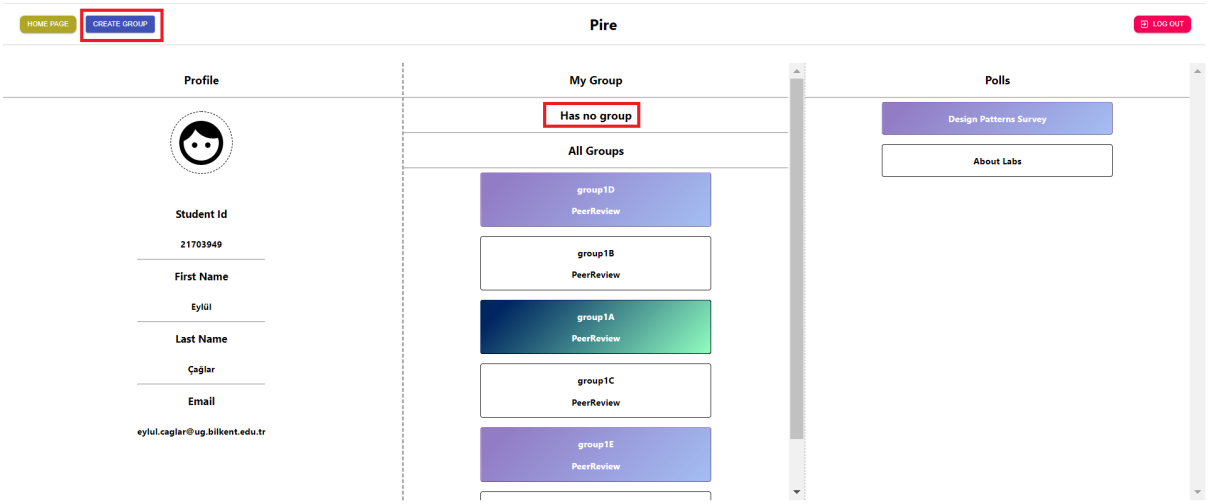


- In the instructor register instructor should enter Instructor Code which is provided by the Pire owners.





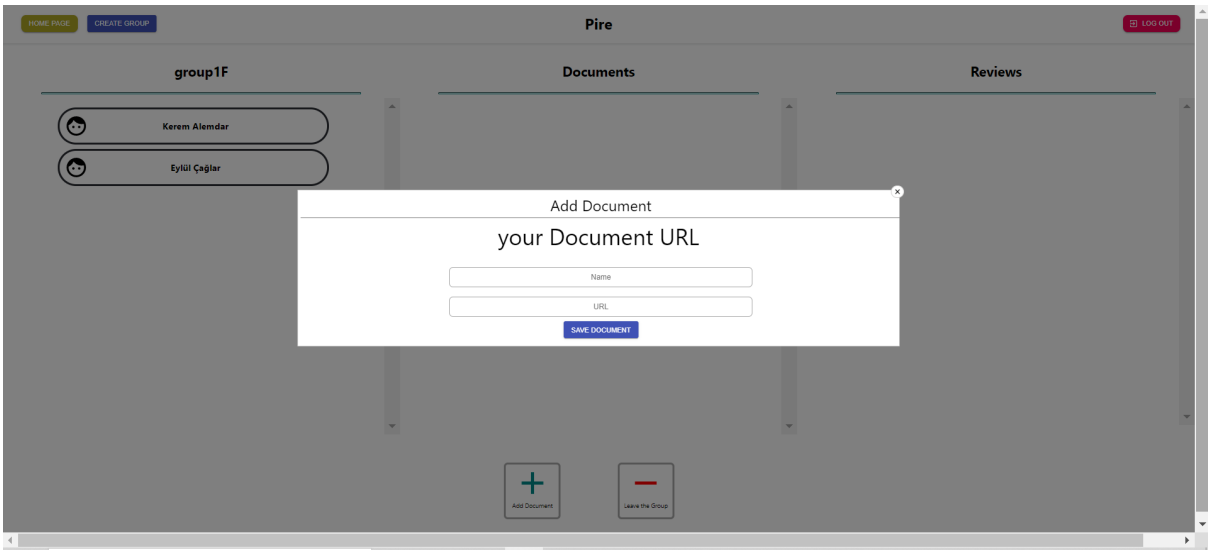
- -If a student does not have any group, the student can create a group by clicking the "Create Group" button.



- A student can join a group by going to the group page and clicking the "Join the Group" button.

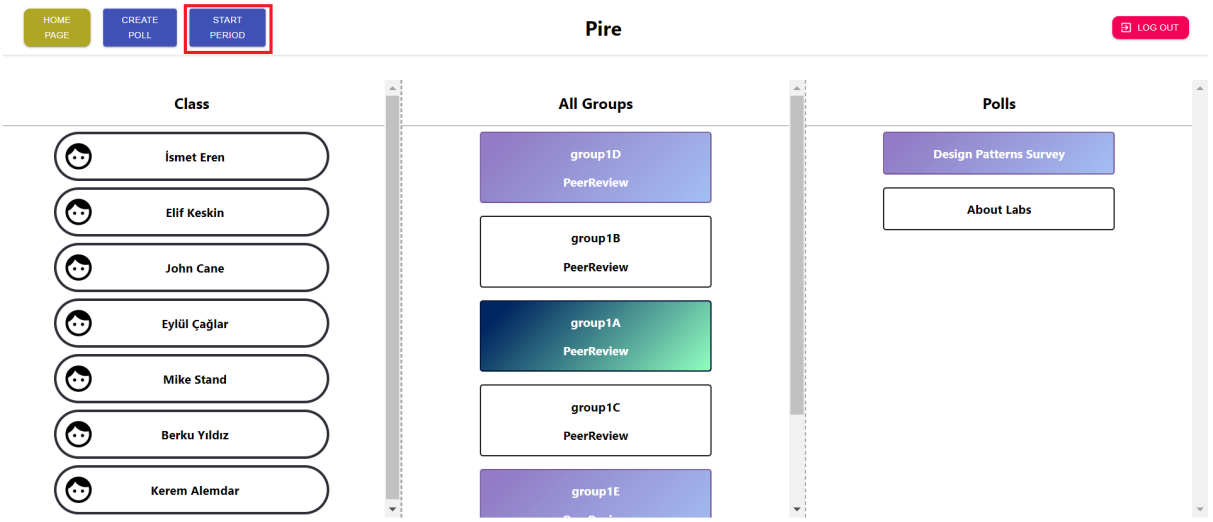
- A student can remove from the group by clicking the "Leave the Group" button.

- A student can add their works url by clicking the “Add Document” button.

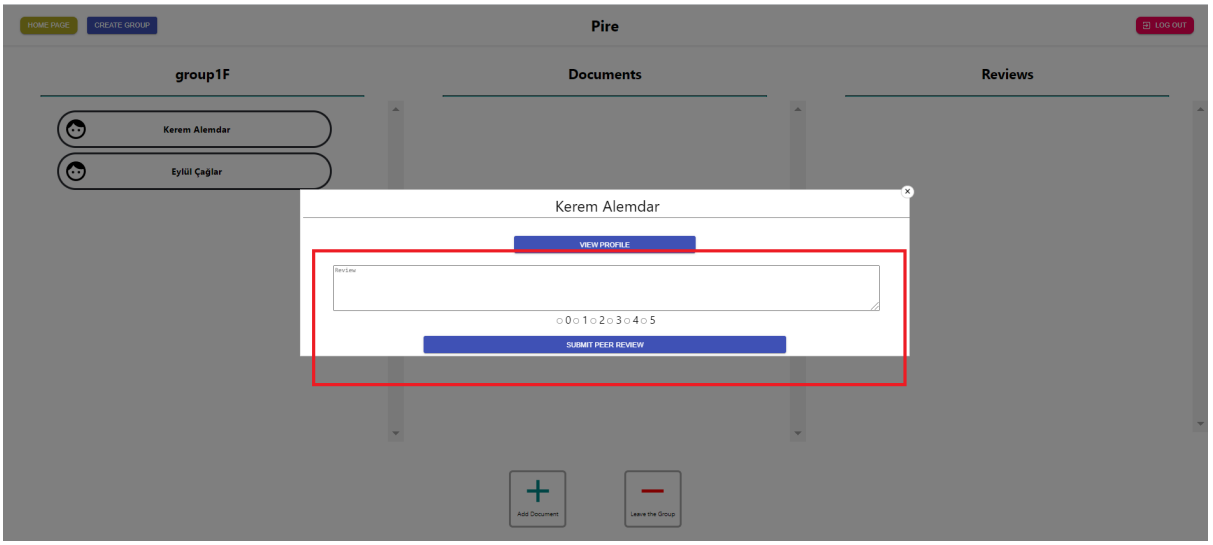


- A student only can see the profile of their peers before the review period starts.

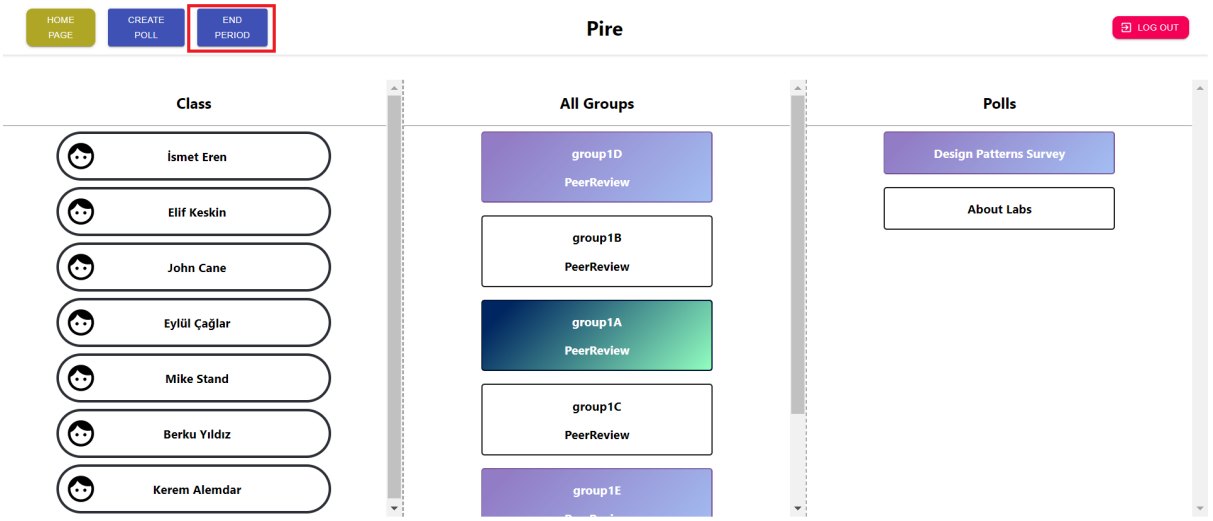
- The instructor can activate the peer review period by clicking the "Start Period" button.



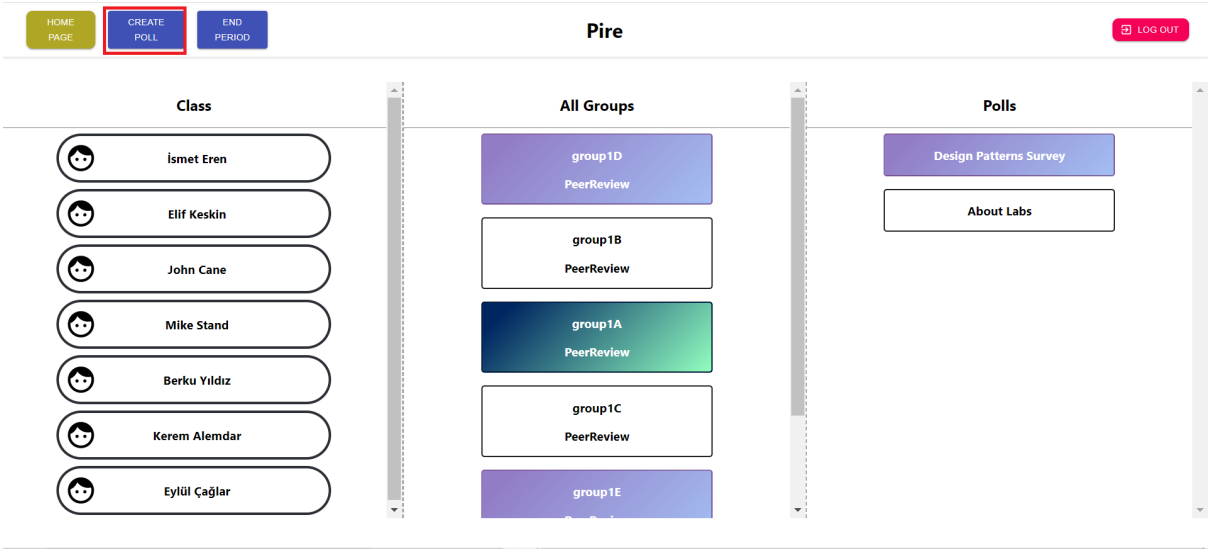
- A student can review his/her group members if the review period is active.

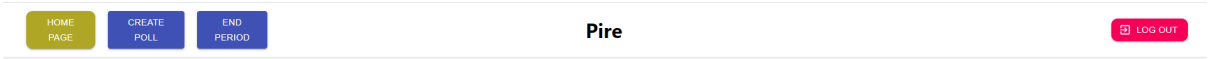


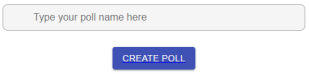
- The instructor can deactivate the peer review period by clicking the "End Period" button.

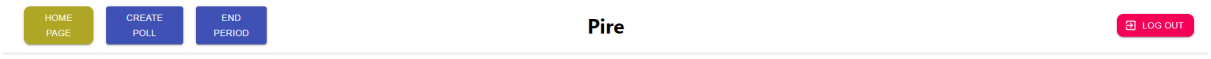


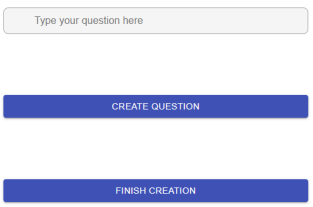
- The instructor can create polls by using the "Create Poll" button and add many questions to the polls.



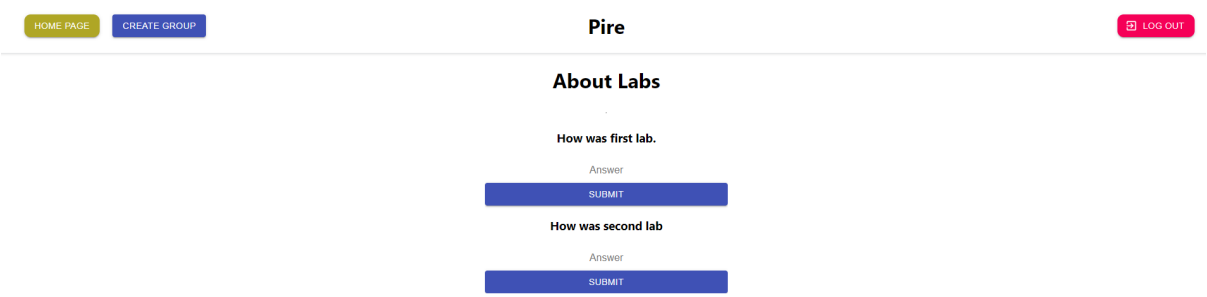




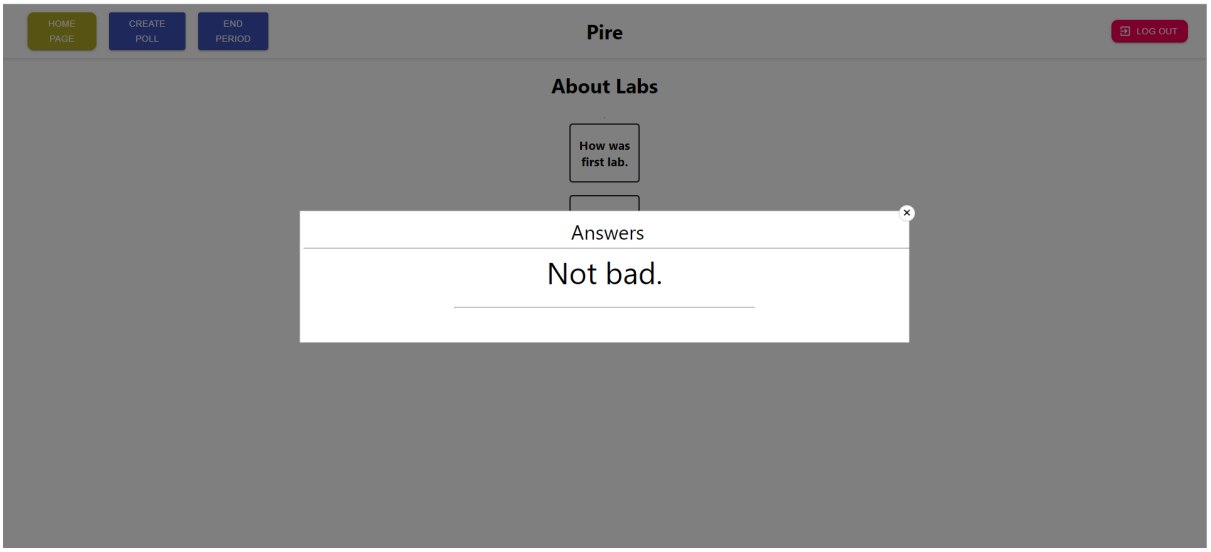




- A student can answer poll questions by clicking the particular poll.

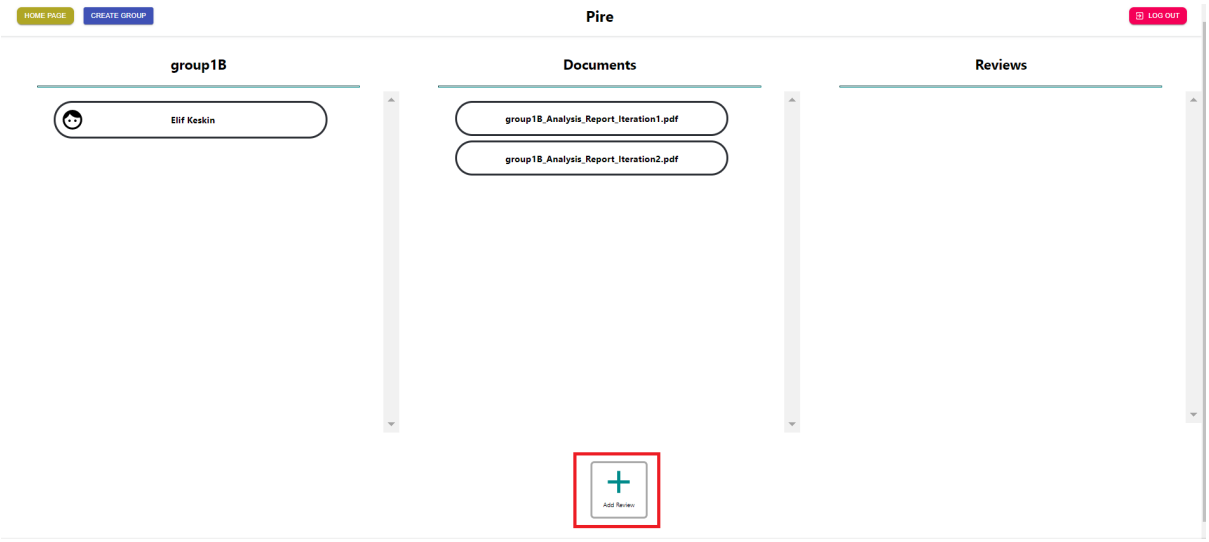


- The instructor can view students' answers to the polls.



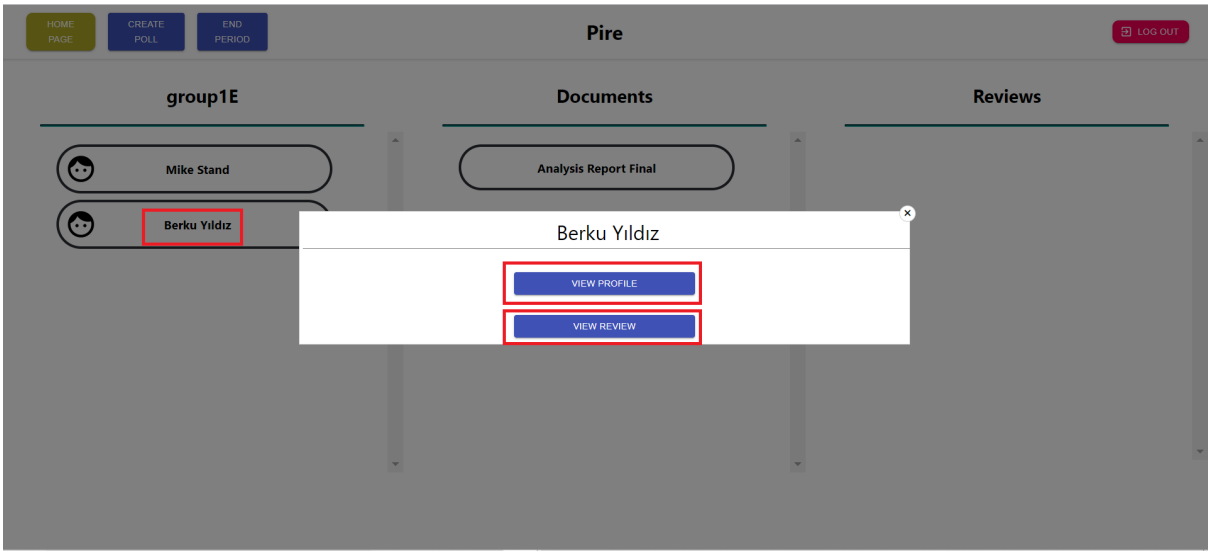
- A student can add review for other groups'

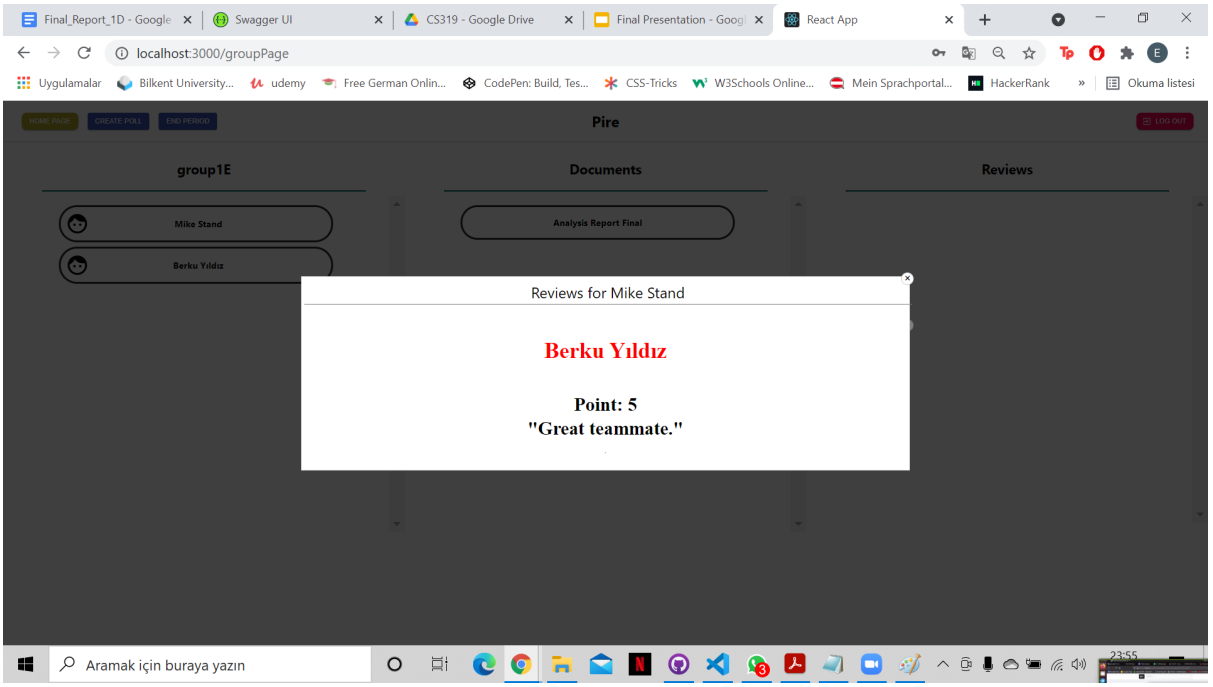
every document one by one, by using the "Add Review" button.



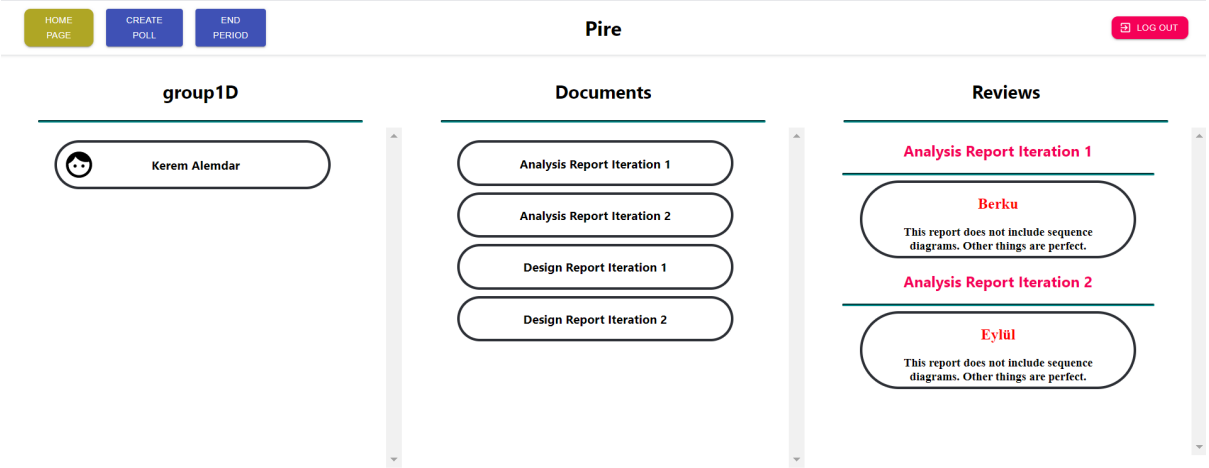
- Every given review for the documents is listed on the right hand side of the Group Page.

- The instructor can see reviews given to the particular student by clicking the name of the student and then clicking the "View Review" button.
- The instructor can see the particular students' profile by clicking the name of the student and then clicking the "View Profile" button.





- This view is from instructors' account and it is the Group Page of an example group group1D.



4.0 BUILD INSTRUCTIONS

Pire application contains 3 different parts which should be run separately. First one is front end server which is React.js application, second part is the backend server which is java spring-boot web application which requires OpenJDK 11 and the last part is PostgreSQL database server version of the PostgreSQL is 11.

4.1 Frontend Server

In order to set up a frontend application, Node.js must be installed to the operating system. The installation guide is different for all operating systems. Therefore the further information can be seen from these links: install Node.js. After installing Node.js, packages and their dependencies have to be installed by using

```
npm install
```

After installation the packages application is ready to run with npm start comment.

```
npm start
```

4.2 Backend Server

In order to set up a backend application, **Apache Maven 3.6.3** and **OpenJDK-11** must be installed to the operating system. The installation guide is different for all operating systems. Therefore the further information can be seen from these links: install maven and install OpenJDK. Installing OpenJDK-11 and java in Debian which is a linux distribution as follows:

```
sudo apt-get install openjdk-11-jdk
```

After this set JAVA_HOME environment variable:

```
export JAVA_HOME=/usr/lib/jvm/openjdk-11-jdk
```

Before the application is run make sure that applications have corrected database connection configuration in application-defoult.yml file. It is under `/pire_backend/src/main/resources/` file. After the database connection is set the application can be runned with `mvn spring-boot:run` command on the command line. This command should run inside of the backend application folder.

4.2.1 Database Setup

PostgreSQL-11 should be installed on the system. Installation guide changes from operation system to operation system therefore see this link to installation guide for specific operation system. If the system will be runned in linux distribution called Debian the steps are as follows:

1. Add the deb file to your source list

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

2. Import the repository signing key

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
```

3. Update the system

```
sudo apt-get update
```

4. Last step install the PostgreSQL

```
sudo apt-get -y install postgresql
```

After PostgreSQL is installed, connect to your database. Start your database with this command:

```
/usr/lib/postgresql/11/bin/pg_ctl start
```

Then connect to PostgreSQL with following command:

```
psql postgres
```

Then create a new database with following command:

```
create database smartblock;
```

The database tables, indexes and schemas are provided in the project folder (link). Create the tables, indexes and schemas accordingly.

5.0 WORK ALLOCATION

Kerem Alemdar: Worked on class diagrams and state diagrams, reviewed use case, sequence and activity diagrams. Worked on the frontend part of the implementation and frontend-backend connection. Implemented poll function and worked on instructor and student functions.

Kaan Ateşel: Worked on UI templates and solution domain class diagrams. In addition to this review class diagrams on the analysis report. Contributed to create the database table and schemas with other group member M. Berk Yıldız. In addition, implemented the backend server and deployed both backend server and database to the linux system.

Eylül Çağlar: Contributed all reports and project design part, worked on the Use Case Diagram and Sequence Diagrams. Reviewed Class Diagram and State Diagram. Worked on the frontend part of implementation. Created Instructor pages and reviewed Student pages, made some minor changes to fix the Student UI.

İsmet Alp Eren: Worked on class diagrams and state diagrams, reviewed all reports and use case, activity and sequence diagrams. Implemented group, main and profile pages and reviewed other frontend parts. Worked on all css files and backend-frontend connections.

Muharrem Berk Yıldız: Contributed all reports and project design part, worked on activity diagrams. Prepared database ER diagram. Reviewed use case, class diagram. Worked on connection between backend and frontend. Made the trailer of the web page.