TED UNIVERSITY

CMPE491

Senior Project 1

Peer Education

Project Analysis

Group – 3

Team Members: YAPRAK DENİZ KEVİNÇ – 31157371828

SALİH IŞIK – 13789110358

MURAT KAAN GÖKYILDIZ – 24919417872

MERT GÖKÇE - 24016271698

Submission Date: 25/11/2023

PROJECT ANALYSIS REPORT

1. INTRODUCTION

We have observed numerous times in ourselves and our social environment that students and people who are willing to learn have problems accessing notes and course content related to various courses. To solve this problem, we have decided to build a web application named “peer2share”. The aim of our platform is to enable people from different universities and people who are eager to learn to share and benefit from their course notes and materials.

2. PROPOSED SYSTEM

For now, it only works in our systems, our product is in alpha phase as we have not yet completed the idea phase. We plan to run our website on GitHub pages in 1-2 weeks.

2.1 OVERVIEW

This project is basically a web-based platform that I can use to obtain university level information on a subject. The main purpose is to connect students from different universities with students. We want to ensure that future generations have access to healthier and easier information by connecting the lower and upper classes with each other.

2.2 FUNCTIONAL REQUIREMENTS

- Users will register on the website using their credentials such as email and password.
- Users should have the ability to edit and update their profiles with personal information, including name, contact details, and educational background.
- Users will be able to upload their lecture notes to the platform. They will provide lecture notes in pdf format, lecture name, book that used in the class, lecturer’s name, and name of the teacher.
- Users who choose to “get lecture notes as an email” will receive a notification.
- Users will be able to like/dislike a specific lecture note upload.
- Users will make comments on specific lecture notes uploaded.

- Users will like/dislike comments on specific lecture note upload.
- Users who took the lecture will be able to send lecture notes. If the user has not taken a specific course, they cannot share notes to related courses.

2.3 NON-FUNCTIONAL REQUIREMENTS

- The system will be user-friendly.
- Users must be connected to the internet to reach the system.
- User passwords will be stored in an encrypted format within a database.
- The system is easy to maintain, upgrade and troubleshoot.
- The system is accessible 7/24 with minimal disruptions and errors.

2.4 PSEUDO REQUIREMENTS

- For the front end we use HTML, CSS, JavaScript and React and Next.js as frameworks.
- Back-end JavaScript, Python, and their frameworks Node.js and Django will be used.
- We will use PostgreSQL as a database.

2.5 SYSTEM MODELS

2.5.1 SCENARIOUS

Use Case (1): Signing up to the system.
Participating Actor: User
Entry Condition: User types of the web address to the search bar, then clicks sign up button.
Exit Condition: User logs off from the applications.
Flow of Events:

- User goes to address of our website.
- At first user doesn't have an account so first he/she clicks to the sign-up button.
- User Signs up to the application by filling credentials such as email (doesn't have to be a university email), nickname, university name (optional) and password.

- Then the system will ask for an email verification from the user.
- After the verification process, User can log in to the system with it account.
- Main page of the application opens.

Use Case (2): Logging in to the system, searching for notes and note page.

Participating Actor: User

Entry condition: User logs into the application

Exit condition: User closes or logs off the application.

Flow of Events:

- User enters email address and password then clicks the login button.
- In the main page User clicks on the search bar and start typing the lecture name to search the lecture notes.
- Users will be navigated to note page. In the page, on the left side there will be the name of the book in which users take their notes, on the middle side lecture note will be displayed, on the right side there will be two buttons. One for downloading the lecture note the other for adding the notes to favorites.
- After viewing the notes, at the bottom there will be like and dislike buttons.
- Users can like, dislike, and make comments on lecture notes.

Use Case (3): Logging in to the system, navigating to different pages and their usage.

Participating Actor: User

Entry condition: User logs into the application

Exit condition: User closes or logs off the application.

Flow of Events:

- When the user clicks on their icon in the top right, there will be options such as uploading; for uploading the lecture notes, favorites; for navigating to favorite lecture notes, account; for viewing the account information and editing the profile.

- When the user presses the favorite section, the user will be able the view the notes that saved in users favorites. When a user clicks on a specific note it will go directly to the note page.
- When the user presses the upload section, user will be navigated to the upload page. In the middle of the page there will be a “Select Note to Upload” button when it is clicked it will ask the pdf location to upload. Also, if users want to, they can drag their pdfs to the dotted area on the middle of the page to upload their files.
- When users press the account section, users will be navigated to account page. In this page users can see their credentials and can edit them.

Use Case (4): Logging in to the system with admin account.

Participating Actor: Admin

Entry condition: Admin logs into the application

Exit condition: Admin closes or logs off the application.

Flow of Events:

- Admin will log into the application through different login screens, which are specialized for only admins.
- After Admin logins it will be a dashboard on the left and there will be users and notes tab on the dashboard.
- In the home page users list will be shown.
- In each entry, their nickname, email address, their account creating date will be displayed and at the end of every entry there will be two buttons, one for deleting the user other for editing users’ credentials.
- In the notes tab, lecture note names along with creation date, and creator will be displayed.
- At the end of every entry there will be a delete button.

Use Case (5): Logging in to the system with teacher account.

Participating Actor: Teacher

Entry condition: Teacher logs into the application

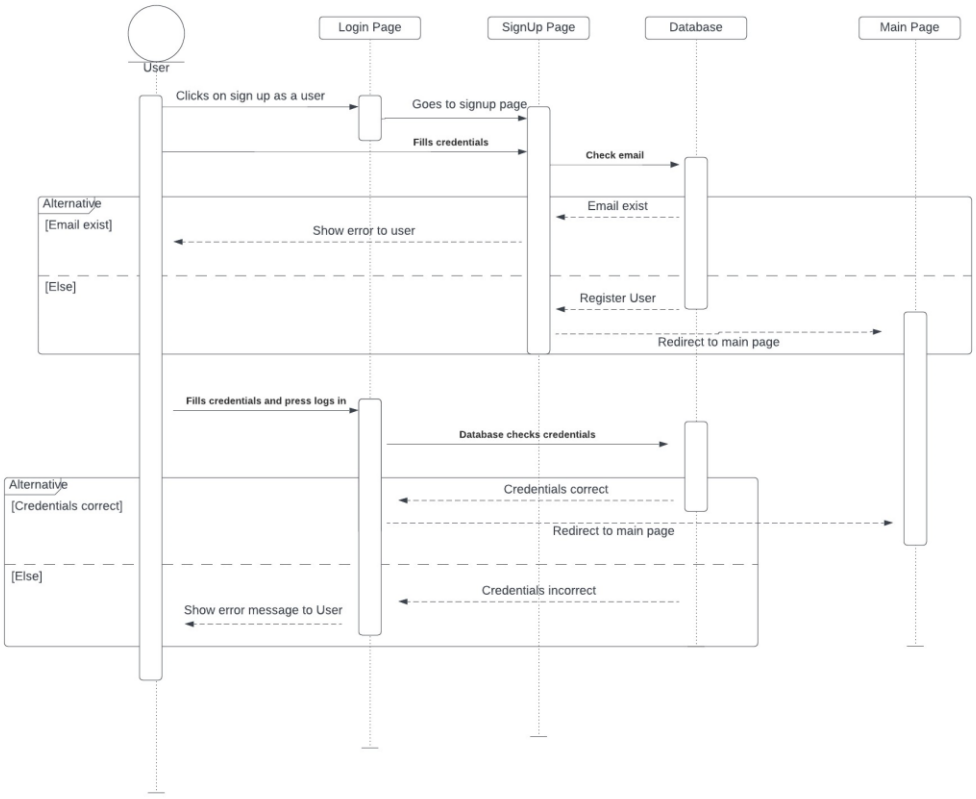
Exit condition: Teacher closes or logs off the application.

Flow of Events:

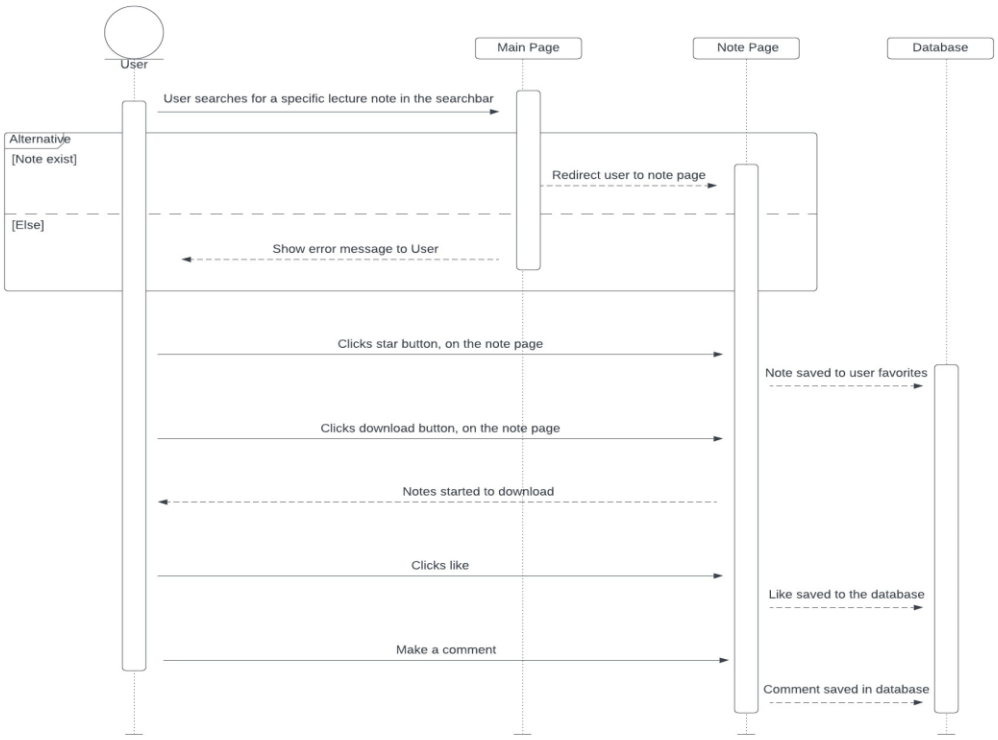
- Teacher will sign up and log in to the application just like any other user.
- After the teacher communicates with the application owners their role ‘Teacher’ will be given to them.
- Owners of the application will change teacher’s role to ‘Teacher’ through database.
- The teacher will be able to report a specific lecture note.

2.5.2 DYNAMIC MODELS

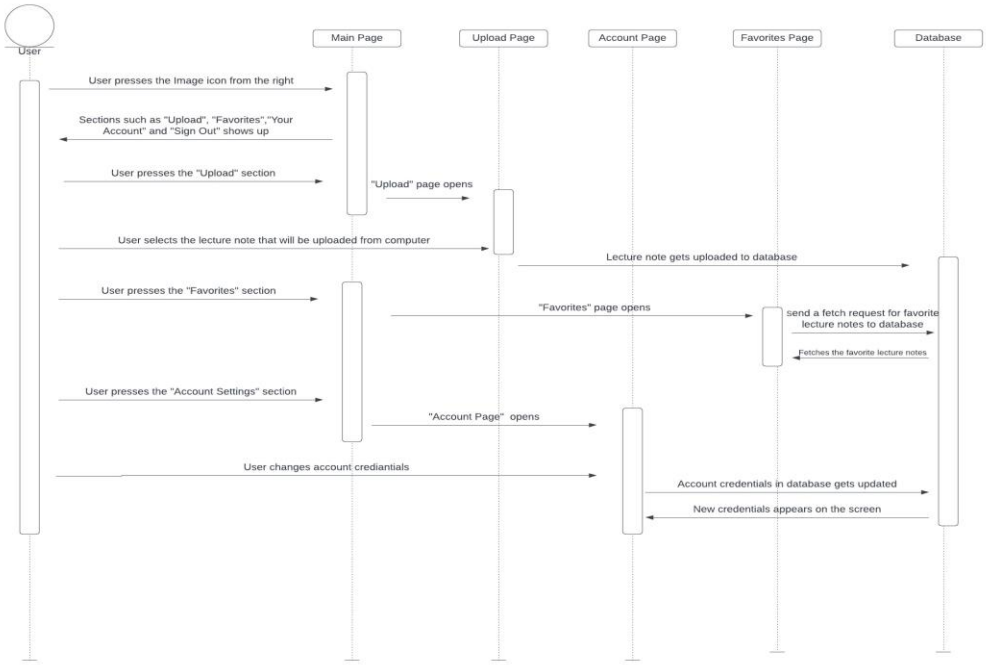
CASE (1)



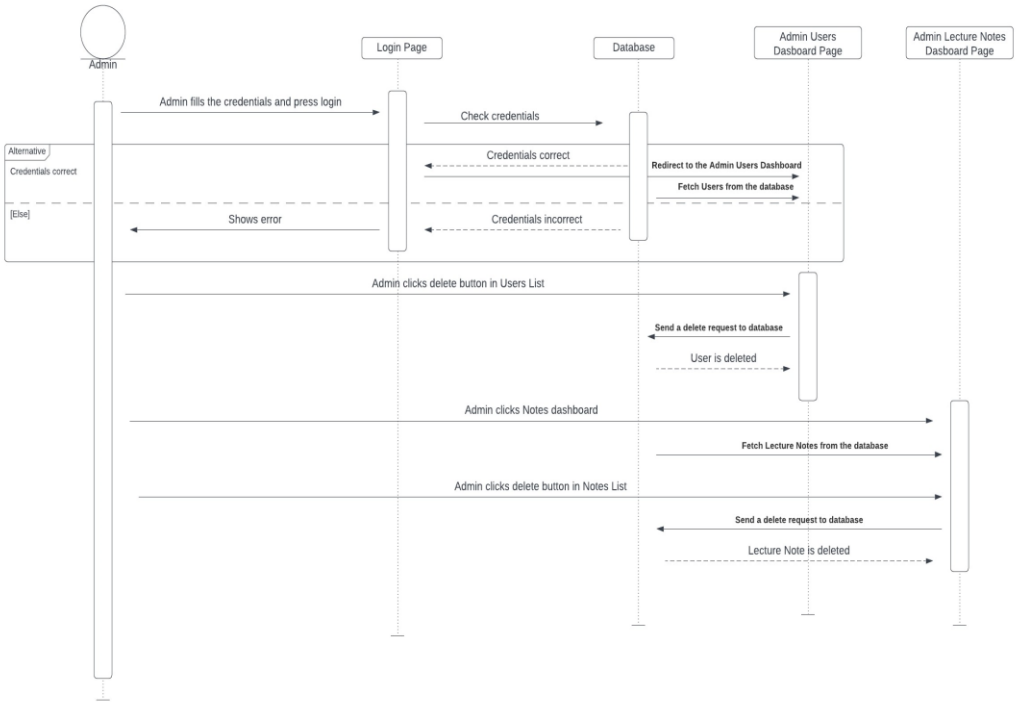
CASE (2)


User
Main Page
Note Page
Database
User searches for a specific lecture note in the searchbar
Alternative
[Note exist]
Redirect user to note page
[Else]
Show error message to User
Clicks star button, on the note page
Note saved to user favorites
Clicks download button, on the note page
Notes started to download
Clicks like
Like saved to the database
Make a comment
Comment saved in database

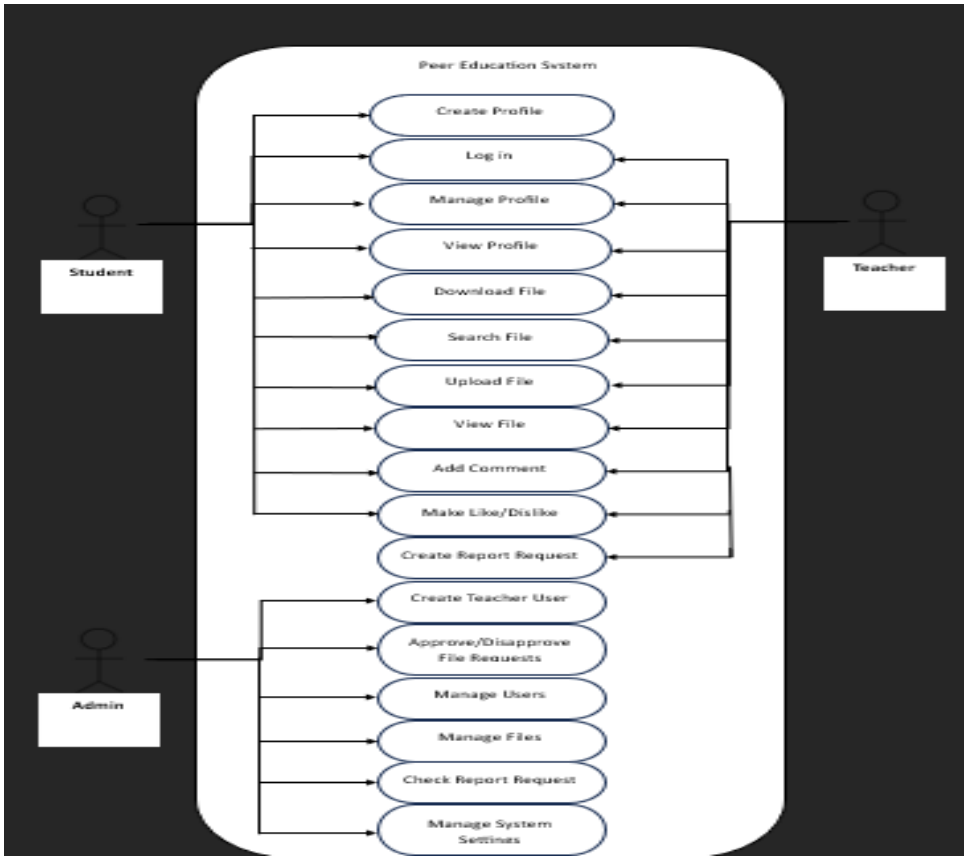
CASE (3)


User
Main Page
Upload Page
Account Page
Favorites Page
Database
User presses the Image icon from the right
Sections such as "Upload", "Favorites","Your Account" and "Sign Out" shows up
User presses the "Upload" section
"Upload" page opens
User selects the lecture note that will be uploaded from computer
Lecture note gets uploaded to database
User presses the "Favorites" section
"Favorites" page opens
send a fetch request for favorite lecture notes to database
Fetches the favorite lecture notes
User presses the "Account Settings" section
"Account Page" opens
User changes account crediantials
Account credentials in database gets updated
New credentials appears on the screen

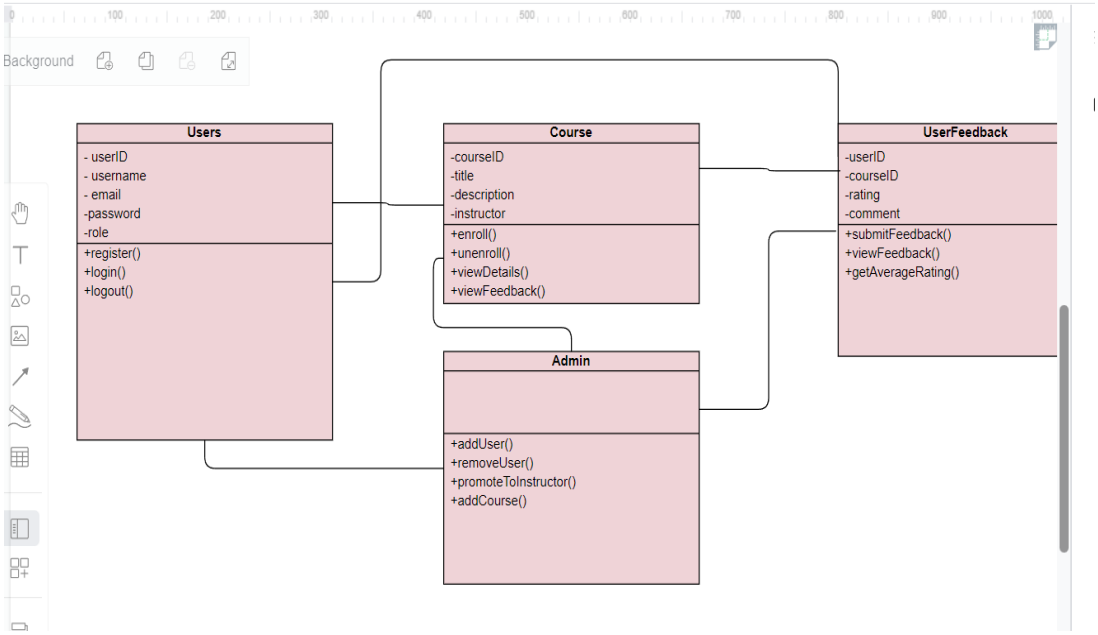
CASE (4)


Admin
Login Page
Database
Admin Users Dasboard Page
Admin Lecture Notes Dasboard Page
Admin fills the credentials and press login
Check credentials
Alternative
Credentials correct
Credentials correct
Redirect to the Admin Users Dashboard
Fetch Users from the database
[Else]
Shows error
Credentials incorrect
Admin clicks delete button in Users List
Send a delete request to database
User is deleted
Admin clicks Notes dashboard
Fetch Lecture Notes from the database
Admin clicks delete button in Notes List
Send a delete request to database
Lecture Note is deleted

2.5.3 USE CASE MODEL



2.5.4 OBJECT AND CLASS MODEL



2.5.5 USER INTERFACE

Now we do not have an eye appealing interface to offer, we have just finished the idea part and we continue to build the architecture in the back-end part and add something to web development.

3. GLOSSARY

- User: Anyone who wants to use our system and has information about it.
- Tutor: A person who has taken the course at that time or has good notes about the course and has passed it with a high grade (can also be a user).
- Controller: Although we think of university lecturers in the first stage, it is highly likely that the administrators will control the content.
- SEO (Search Engine Optimization): Methods used to optimize search engines.
- Responsive Design: It is used to create a responsive design for different screen sizes.
- Front-End: The visible side of the website or application with which the user interacts directly.
- Back-End: The part of the website or application on the server side that manages the database and server functions.
- API (Application Programming Interface): It is the link between two software, for applications such as email confirmation, SMS.
- Cookie: They are small particles added to track the user's session, it is one of the systems we will use to provide a better experience to the user.
- Framework: The infrastructure and architecture of our software and project.

4. REFERENCES

- https://www.webfx.com/web-development/glossary/
- https://www.lucidchart.com/pages/uml-use-case-diagram
- https://www.w3schools.com/whatis/
- https://richardsondx.medium.com/the-role-of-functional-programming-in-web-development-for-ruby-developers-82eee9570cf8
- https://www.oracle.com/tr/applications/what-is-saas/
- https://clutch.co/resources/web-development-glossary-75-essential-terms
- https://devdocs.io/javascript/
- https://www.w3schools.com/tags/

- https://www.w3schools.com/cssref/index.php