TED UNIVERSITY

CMPE492

Senior Project 2

Peer2Share

Low-Level-Design

Team Members: YAPRAK DENİZ KEVİNÇ – 31157371828

SALİH IŞIK – 13789110358

MURAT KAAN GÖKYILDIZ – 24919417872

MERT GÖKÇE – 24016271698

Submission Date: 11/3/2024

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.

The design goal of our project is for users to be able to access course notes and upload course notes reliably and quickly using user-friendly UI.

The constraints of our project are keeping safe of users’ data, reporting sensitive/copyrighted content for maintaining high level of user experience and ensure that web application is available for users quickly.

1.1. OBJECT DEGISN and TRADE-OFFS

1.1.1. Maintainability v Availability

Availability is the application is accessible continuously for users. Maintainability means the web application can be updated in future, corrects errors inside and adds new properties. If too complex and modular structures are used for making web application maintainable, availability feature can be affected negatively. It is required to balance them.

1.1.2. Functionality v Usability

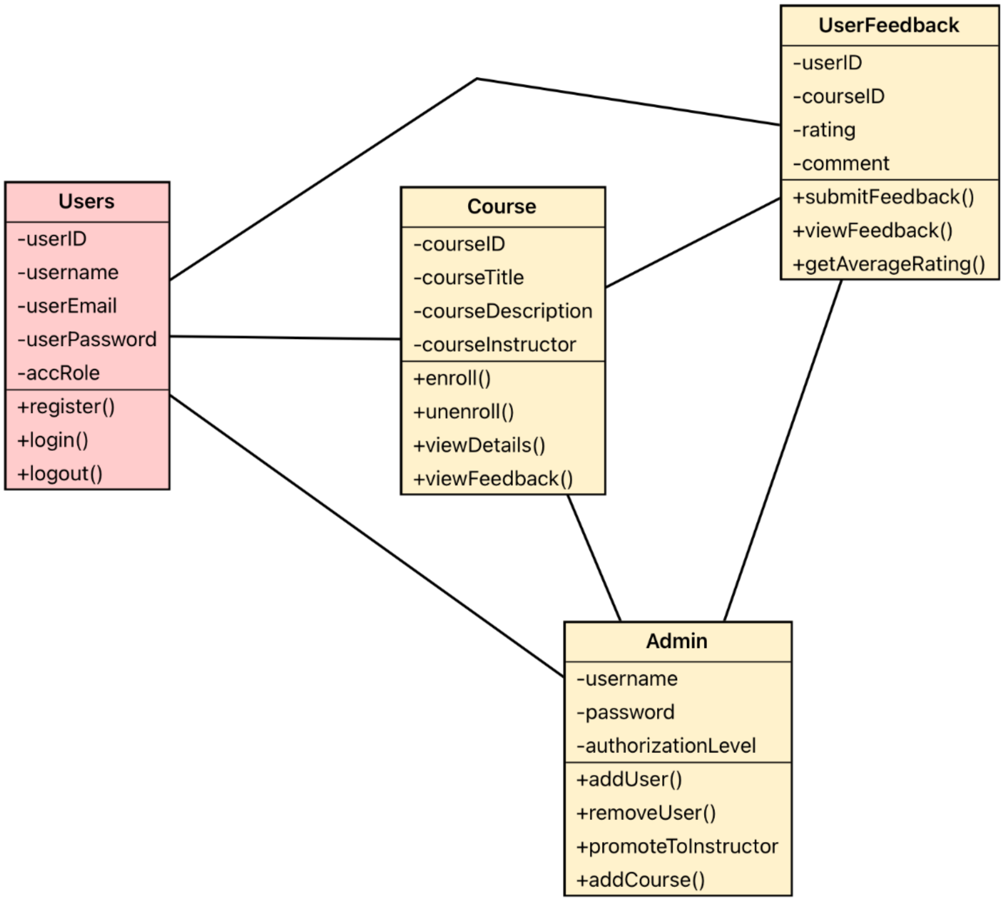
Functionality is the features and properties of our application. Usability is the feature that shows how easily, quickly, and effectively users can use the application. More functionalities ensure the application can provide solutions for a variety of needs. However too much functionality makes the application complex and difficult for users to understand.

1.2.Interface Documentation Guidelines

We have explained and visualized the design of our project using UML standards. Even though we do not have an implemented system yet, our project is still in shaping and development.

1.3. Engineering Standards

UML, which is the accepted standard, is used for our project. With respect to UML design principles, classes, entities, and diagrams are used to explain and visualize the Peer2Share project.



1.4. Definitions, Acronyms and Abbreviations

UI: User Interface, UML: Unified Modeling Languages

2. PACKAGES

The packages in our project are used as themes to manage different components of the project. Since we use Django in the backend part, we mainly use Django REST Framework, and we use it to manage our APIs. Requests and Beautiful Soup are used for data exchange from external sources and web scraping. Celery and Redis are used to manage asynchronous processes. PostgreSQL is the main database management system of the project. Below are the other packages we may use in the future and the details of the packages we use.

2.1.Packages Currently in Use and in Trail

1)Django: It is a high-level framework for developing web applications with Python. Django provides many of the tools needed to quickly build secure and scalable web applications.

2)Django REST Framework: Django REST Framework is a framework for creating and managing RESTful APIs for Django-based web applications.

3)Requests: It is a library used to make HTTP requests with Python. It is widely used to exchange data from external sources. It supports HTTP methods such as GET, POST, PUT, DELETE.

4)Beautiful Soup: It is a Python library used to extract data from HTML and XML files. It can be used for web scraping. It is useful for analyzing pages, scraping data and processing data.

5)Celery: It is a Python-based, distributed task queue manager. It is used for managing and running asynchronous tasks. Ideal for running long-running processes in the background.

6)PostgreSQL: It is an open-source relational database management system. It provides advanced data integrity, data consistency and performance. It is supported by many frameworks and applications such as Django.

2.2.Packages That Can be Used in the Future

1)Redis: It is an open-source key-value storage system and message broker. It provides a fast, flexible, and durable data structure. It supports asynchronous operations using tools such as Celery.

2)Django Channels: It is a tool used to provide real-time communication in Django projects. It offers support for the WebSocket protocol and asynchronous coding. It can be used for real-time chat, notifications, and games.

3)Django Allauth: Provides user authentication and account management for Django-based web applications. You can easily integrate functions such as registration, login, password reset and login with social media accounts.

4)Django Guardian: It is a tool that facilitates permission management for objects in Django projects. It provides model-based permissions and allows you to control user access to objects.

5)Django Filter: Used to filter data using query parameters in Django based web applications. Provides a customizable filtering interface.

6)Django Debug Toolbar: Used to monitor and debug performance during development in Django projects. It visually presents HTTP requests, queries, templates, and various other performance metrics.

7)Django Storages: Used for storing files in Django projects. It provides integration with external storage services such as Amazon S3, Google Cloud Storage, Azure Blob Storage.

3. CLASS INTERFACE

Since it is very suitable to develop further improvements to our project. We haven't created any code for the system yet. We will implement our project and make improvements to it. Therefore, we don't have any code class analysis available. But we can portray to you the main structure of our project.

3.1.Model

a) User Class

This class is the base class of all user types. It has attributes such as id, username, password and so on. Users will be able to upload their lecture notes to the website. Make comments about lecture notes and rate them.

Attributes

int id

String nickname

String password

Comments []

Likes []

Dislikes []

UploadedLectures []

Methods

UploadLectureNote

Like

Dislike

b)Admin Class

This class will be able to delete certain users and lecture notes from the dashboard and edit users profile credentials.

Attributes

int id

String nickname

String password

Methods

DeleteUser

UpdateUser

DeleteLectureNote

UpdateLectureNote

3.2.User Interface Layer Class Interfaces

a)LoginPage

loginRequest(nickname, password): On click it will send a request to the server to check the user credentials. If credentials are correct then redirected to the main page.

b)RegisterPage

registerRequest(nickname, password, universityName): On click it will register the user with the provided credentials.

c)MainPage

Search(lectureNoteName): When users type a specific lecture note, matching lecture notes will be listed for the users search interest.

EditProfile(newName, newPassword): Users will be able to edit their credentials.

UploadLectureNote(lectureNote): Users will be able to upload documents in pdf format.

d)LectureNotePage

Comment[]: Users will type the comments to the textfield that is provided at the bottom of the lecture pdf. When the comment button is pressed, the comment will be sent to server.

Like[] & Dislike[]: Similarly for the like and dislike button, it will record the rating on the server.

4.GLOSSARY

Application Programming Interface (API): A set of protocols, tools, and definitions that allows different software applications to communicate with each other, enabling data exchange and functionality implementation.

Back-End: The part of the website or application on the server side that manages the database and server functions.

Cascading Style Sheets (CSS): A style sheet language used for describing the presentation of a document written in HTML.

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.

Database Management System (DBMS): Software used to create, update, and query databases, managing storage and retrieval of platform data.

Django: A high-level Python web framework used for rapid development and clean design in building web applications, serving as the backend framework for the Peer2Share project.

Framework: A structured set of tools, libraries, and best practices used to develop peer2share.

Front-End: The visible side of the website or application with which the user interacts directly.

Git: A distributed version control and source code management system used in software development processes.

GitHub: A web-based platform for version control and collaboration, used for managing the source code of the Peer2Share project and facilitating teamwork among developers.

Hypertext Markup Language (HTML): The standard markup language for creating web pages and web applications.

JavaScript: A high-level programming language commonly used alongside HTML and CSS to create dynamic and interactive web pages.

JSON Web Tokens (JWTs): A standard used to securely transmit information over the internet.

Peer2Share: An interactive platform that allows users to share and exchange knowledge, notes, and other educational materials with their peers and the community.

PostgreSQL: A highly stable database management system that supports both SQL and JSON querying.

Python: The object-oriented, high-level programming language.

Responsive Design: Creating web pages that provide optimal viewing and interaction experiences across different screen sizes.

User Authentication: The process by which new users register and existing users log in to the platform securely.

- 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.

5.REFERENCES

1-ISO Code of Ethics and Conduct.

2-Django Documentation: https://docs.djangoproject.com/en/5.0/

3-Wikipedia. (n.d.). Git : https://en.wikipedia.org/wiki/Git

4-Wikipedia. (n.d.). PostgreSql: https://en.wikipedia.org/wiki/PostgreSQL

5-[8]. Hostinger Tutorial (Aug 29, 2023), What Is a DBMS: Understanding Database Management Systems: https://www.hostinger.com/tutorials/dbms

6- https://www.django-rest-framework.org

7- https://pypi.org/project/beautifulsoup4/

8- https://docs.celeryq.dev/en/main/django/first-steps-with-django.html