



TED UNIVERSITY

CMPE491

Senior Project 1

Peer2Share

High Level Design

Group – 3

Team Members:

YAPRAK DENİZ KEVİNÇ – 31157371828

SALİH IŞIK – 13789110358

MURAT KAAAN GÖKYILDIZ – 24919417872

MERT GÖKÇE - 24016271698

Submission Date: 07/01/2024

TABLE OF CONTENTS

1. Introduction	3
1.1 Purpose of the System	3
1.2 Design Goals	3
1.2.1 Availability	3
1.2.2 Scalability	3
1.2.3 Performance	3
1.2.4 Community Engagement.....	3
1.2.5 Privacy and Security	3
1.3 Definitions, acronyms, and abbreviations.....	4
1.3.1: Definitions	4
1.3.2: Acronyms and Abbreviations.....	4
1.4 Overview	4
2. Proposed Software Architecture	5
2.1 Overview	5
2.2 Subsystem Decomposition	5
2.3 Hardware/Software Mapping.....	6
2.4 Persistent Data Management.....	7
2.5 Access control and security	7
2.6 Global software control	8
2.7 Boundary conditions	8
2.7.1 Initialization	8
2.7.2 Termination	8
2.7.3 Failure.....	8
3.References	9

1. Introduction

1.1 Purpose of the System

Peer2share systems' goal is to develop a user-friendly, interactive web platform that encourages academic collaboration among university students and also people who eager to learn course materials. Through our web application, people, especially university students, can access academic information in the various branches they want to learn and share their ow course notes on single platform.

1.2 Design Goals

1.2.1 Availability

Availability will be maximized by minimizing the requirements for those who want to use peer2share. It will be enough for the user to use our application to have a running computer, a phone or tablet and internet access.

1.2.2 Scalability

Our system is designed to effectively handle an expanding user base with offering scalability for infrastructure and software architecture. New features will be able to integrated easily into our application. Furthermore, without compromising performance, system will be able to handle larger data quantities.

1.2.3 Performance

Web application will have fast load times and effective data processing because of high performance feature. We aim to obtain performance by optimizing every part of our system.

1.2.4 Community Engagement

Users can meet each other in the comment section below the course notes uploaded to our system. They can comment on the uploaded content and make suggestions / criticism.

1.2.5 Privacy and Security

Ensuring the privacy and security of users is crucial. In order to achieve it, encryption of data, authentication and authorization processes will be applied.

1.3 Definitions, acronyms, and abbreviations

1.3.1: Definitions

- Peer2Share: A interactive platform that users have opportunities to share and exchange their knowledge, notes, etc. with their peers and community.
- User-friendly: User-friendly is a easy to use or understand to software.
- Web- based solution: A web-based solution is a software you access over the internet via a web browser.
- PostgreSQL: Highly stable database management system is an advanced, enterprise-class open source relational database that supports both SQL and JSON querying.
- HTTP POST Request: The POST request method requests that a web server accept the data enclosed in the body of the request message, most likely for storing it.
- User Authentication: The process by which new users register and existing users login to the platform, ensuring secure access to their accounts.
- Git: Git is a speed-oriented, distributed version control and source code management system used in software development processes.
- Content Management System: Software managing the addition, evaluations, and presentation of the content. Includes features like user comments, ratings, and identification of popular content.
- JSON Web Tokens: It is a standard used to transmit information securely over the internet.
- Database Management System: Software used to create, update, and query databases, managing storage and retrieval of platform data.

1.3.2: Acronyms and Abbreviations

- UI: User Interface
- SQL: Structured Query Language
- TCP/IP: Transmission Control Protocol/Internet Protocol
- SSD: Solid-State Drive
- HDD: Hard Disk Drive
- DBMS: Database Management System
- JSON: JavaScript Object Notation
- JWTs: JSON Web Tokens
- HTTP: Hypertext Transfer Protocol

1.4 Overview

We have designed our project as an intra-university and inter-university course sharing and collaboration platform. The core ethos of our project centers on harnessing the most effective and user-friendly learning methodology—peer education. Our overarching goal is to transcend institutional boundaries, granting users unfettered access to a vast repository of resources both within and beyond their academic institutions. This inclusivity empowers individuals to cultivate interdisciplinary insights, enriching their comprehension of post-university programs and diverse departments.

On this dynamic stage, users participate in various activities. The registration of new members is easier, and existing users conveniently and securely access their accounts. The home page is like a place where users browse through different subjects, using search and filter to browse according to their interests. After identifying a particular subject, users investigate university-level information about the topic, the contents of courses on that topic, and links to resources. Profile Management involves user-specific profiling, peer following, and interaction history tracking. Additionally, the platform enables content engagement through commenting, rating, and linking to resources.

2. Proposed Software Architecture

2.1 Overview

The Peer Education Platform is conceived as a comprehensive web-based solution that facilitates collaborative learning and knowledge sharing among peers. The architecture of the platform is designed to guarantee scalability, security and robust performance while providing users with a seamless interaction experience.

2.2 Subsystem Decomposition

1) User Registration and Login:

The user accesses the home page.

New users register to the platform by filling out the registration form.

Existing users access their accounts by logging in.

2) Home Page:

Users can browse general topics and popular content on the home page.

They can find the topics they want with topic search and filtering options.

3) Topic Details:

Users are directed to the detail page when they click on a specific topic.

The subject detail page contains university level information, course contents and resource links.

4) Profile Management:

Users can create and edit their profiles and follow other users.

The profile page includes user information, followed users and interaction history.

5) Adding and Reviewing Content:

Users can comment on topics, rate them, and add their own resource links.

The platform identifies popular content based on users' interactions and reviews.

6) Content Comment Section:

Users can select specific content and comment on it.

Comments are displayed on the content page and provide interaction with other users.

7) Data Security and Privacy:

The platform securely stores user data and protects it with encryption methods.

Users control data sharing by setting their privacy preferences.

2.3 Hardware/Software Mapping

1) User Interface (UI):

Hardware: Various devices such as computers, laptops, tablets, and smartphones.

Software: Web browsers (Google Chrome, Mozilla Firefox, etc.). Web technologies such as HTML, CSS, JavaScript, and Python provide the user interface of the platform.

2) Subject Information Repository:

Hardware: Servers and database storage units.

Software: Database management system PostgreSQL. Database management software is used to store topic contents and user information.

3) User Business Logic:

Hardware: Servers and computers providing processing power.

Software: Server-side application software (backend) that manages the business logic of the platform, usually written in languages such as Python and Node.js.

4) Connection and Communication:

Hardware: Network infrastructure, routers, and switches.

Software: Includes network protocols (HTTP, TCP/IP) and communication software used to provide communication.

5) Database Management:

Hardware: Database servers and storage units.

Software: Database management system (DBMS) software is used to create update and query databases.

6) User Registration and Login:

Hardware: User devices and servers.

Software: User authentication software, session management software.

7) Content Addition and Evaluation:

Hardware: User devices and servers.

Software: Content management system, software that manages commenting and evaluation processes.

8) Version Control:

Software: Git is used for version control in peer2project Project.

2.4 Persistent Data Management

1) Database Management:

Hardware: Database servers and storage units.

Software: Database management system (PostgreSQL) software is used to create update and query databases.

2) Data Storage and Backup:

Hardware: SSDs, HDDs, storage, and backup devices.

Software: Automated data backup software manages regular data backup processes and data storage software ensures that data is stored efficiently.

3) Data Security:

Software: Data encryption algorithms, software containing security protocols, software for monitoring and correcting security vulnerabilities.

4) Data Access Controls:

Software: Data access authorization software controls users' access to and editing of specific data, thus ensuring data security.

2.5 Access control and security

We will use JSON Web Tokens (JWTs) for access control. When users login to our server address /api/login as an HTTP POST request. If the username and password are correct, the server generates a token that identifies the logged-in user. The token is signed cryptographically, which makes it impossible to falsify. The backend responds with a status code indicating the operation was successful and returns the token with the response. Browser saves the token and when user makes new requests, server checks the user's authorities and permits or rejects user requests.

2.6 Global software control

In our applicationn client-server model will be used. This model is the most used architecture in networking. Client relies on sending a request to another program in order to access a service that made available by a server. The server runs one or more programs that share resource with the client. In our project user will constantly send requests to server to login or signup to the website. To download or upload lecture notes to website or making comments. In order to for these to happen user and server will communicate through requests.

2.7 Boundary conditions

2.7.1 Initialization

Users will need internet connection to view the application. When user types the domain name on the browser user will be directed to the main page of the website. Since the server-client model will be used in the application, the server will be ready before the user make any request. In the main page user can make any prefereable request that user wants.

2.7.2 Termination

When user closes the browser or visit any other URL the application will terminate. If the user quits the page in a middle of a process the process will be lost.

2.7.3 Failure

If power connection is lost failure will ocur during the process. User has to redo or restart the process in order to proceed.

3. References

- [1]. Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.
- [2]. Wikipedia. (n.d.). Git : <https://en.wikipedia.org/wiki/Git>
- [3]. Wikipedia. (n.d.). PostgreSQL: <https://en.wikipedia.org/wiki/PostgreSQL>
- [4]. Wikipedia. (n.d.). POST (HTTP): [https://en.wikipedia.org/wiki/POST_\(HTTP\)](https://en.wikipedia.org/wiki/POST_(HTTP))
- [5]. JSON.org (n.d.) Introducing JSON: <https://www.json.org/json-en.html>
- [6]. TechTarget Networking, TCP/IP:
<https://www.techtarget.com/searchnetworking/definition/TCP-IP>
- [7]. Wikipedia. (n.d.) JSON Web Token: https://en.wikipedia.org/wiki/JSON_Web_Token
- [8]. HostingerTutorial (Aug 29, 2023), What Is a DBMS: Understanding Database Management Systems: <https://www.hostinger.com/tutorials/dbms>
- [9]. ResearchGate: https://www.researchgate.net/figure/Overview-of-the-proposed-software-architecture_fig1_221594330
- [10]. Medium. (Sep 24, 2022). How to design software architecture?:
<https://medium.com/tradeling/how-to-design-software-architecture-c39eb5481a07>