TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 492 - Senior Project

Test Plan Report

peer2share

Team Members: YAPRAK DENİZ KEVİNÇ – 31157371828

SALİH IŞIK – 13789110358

MURAT KAAN GÖKYILDIZ – 24919417872

MERT GÖKÇE – 24016271698

Contents

1. Introduction

This test plan has been created to define the test strategy, scope, and requirements of the peer2share web project. The test plan will determine how test activities will be managed and implemented during all phases of the project.

1.1 Scope

This test plan has been created to guide the testing process of the peer2share web application. The tests specified below are designed to check the key components and functionalities included in the project scope:

a) User Interface Tests: The user interface, compatibility and usability of the website will be tested.

b) Functional Tests: The basic functions and modules of the system will be tested. Then check for acceptance criteria.

c) Performance Tests: Response time, load tolerance and overall performance of the web application will be tested.

d) Security Tests: Security of the application, authentication, authorization, and data security will be tested.

e) Usability Tests: User experience and usability of the website will be tested.

1.2 Test Requirements

a) User Interface Requirements:

The compatibility of the website in different browsers should be tested. The user interface should be easy to use and be a user-friendly experience.

b) Functional Requirements:

Users must be able to create an account, log in and use the functions. Basic functions (creating a user profile, sharing content, like other contents, make comments other contents e.g.) must work correctly.

c) Performance Requirements:

The response time of the website should be tested underload and should be within acceptable limits. Load tests should determine the capacity of the website to be used by multiple users at the same time.

d) Security Requirements:

User data must be secured, and authorizations must be implemented correctly. The application should be regularly scanned for security vulnerabilities and updates should be made.

e) Usability Requirements:

The website should be easy to navigate, easy, and understandable for users. Appropriate feedback should be provided to the user in case of errors.

2. Test Methodology

Test Methodology section indicates test techniques that implied for Peer2Share web application.

2.1 Overview

Testing a software includes confirming and evaluating the software does its intended functions. Testing provides numerous advantages such as improving software performance, preventing bugs. Both smallest units and entire system should be tested in order to ensure that system works correctly. Also all units should contribute the whole system.

2.2 Test Levels

We planned to test the Peer2Share according to five different test techniques. These are Unit Testing, Integration Testing, System Testing, Performance Testing.

2.2.1 Unit Testing

Unit Testing was planned to take place while Peer2Share web application was being developed. The smallest units of our application such as classes will be tested using Unit Testing methodology.

2.2.2 Integration Testing

The Integration testing technique was planned to take place in middle stages of software development phase. We planned to combine some of modules and components in our application to test their functionality and correctness.

2.2.3 System Testing

We planned to use the System Testing method at the end of the development phase and when it is over. We aimed to test the integrity of the system with System testing method. We measure compliance with user expectations, holistic functionality, usability and general performance of Peer2Share.

2.2.4 Performance Testing

We planned to apply Performance Test after the application development phase is done. We aimed to perform this test using popular tools such as Apache JMeter, LoadRunner etc. We will simulate real user interactions with using one of these test tools. For instance user login/logout, routing through different pages, downloading pdf file, commenting the file etc.

3. Risk and Issues

During the testing process of the peer2share web application, several risks and issues may arise that could potentially impact the project's timeline, quality, and overall success.

o Testing the application across different browsers, devices, and operating systems to ensure compatibility and consistent user experience may reveal compatibility issues that need to be resolved.

o Ensuring the security of user data, authentication mechanisms, and overall application security is crucial. Any security vulnerabilities discovered during testing must be addressed promptly to prevent potential data breaches. If the security vulnerabilities provided by the Django framework are not sufficient to close them, we will additionally close these vulnerabilities

ourselves. For this, we can detect the vulnerability on the website by performing a Pentest and taking precautions against it.

o Identifying performance bottlenecks and optimizing the application's performance under various load conditions can be a complex task, especially if the application experiences unexpected spikes in user traffic. For this, we can conduct load testing to simulate high traffic conditions on peer2share application. This can help uncover potential bottlenecks under heavy loads.

4. Test Environment

The application will undergo testing in both staging and production environments to simulate real world usage. The testing infrastructure includes various devices such as desktop computers, laptops, tablets and smartphones running different operating systems mostly windows, but also we will on macOS, İOS and Android. Although the main testing environment of the project will be google chrome, we will also use supported web browser for testing include Mozilla Firefox, Safari and Microsoft Edge. As well compatibility testing will be conducted to ensure seamless functionality across different browser versions. The backend will be tested on servers configured with appropriate database systems, ensuring data integrity and security. Test data will be generated to simulate various user interactions and scenarios covering different file formats for lecture notes (at the same time it will be checked if the files are harmful). At the same time, we will be able to test the performance of our application on computers equipped with different hardware. All of these ensure that the application is tested under conditions similar to those it will encounter in real world usage.

5. Test Schedule

This section specifies schedule of testing process. We aimed to test different functionality for Peer2Share.

Peer2Share is opened in a web browser	05.04.2024	Tested
User can register to the system	07.04.2024	Tested
Registered user can login the system	07.04.2024	Tested
User can logout securely	08.04.2024	Tested
User can see course materials (.pdf files)	17.04.2024	Tested
User can display details for the course material	21.04.2024	Tested
User can download the pdf file	28.04.2024	Tested
User can upload a pdf file to the system	13.05.2024	Will Be Tested
User can add comment on a specific course material	15.05.2024	Will Be Tested

User can add like/dislike to a specific course material	16.05.2024	Will Be Tested
Users' interaction is correctly affected the database	20.05.2024	Will Be Tested
Performance Testing of the Peer2Share	25.05.2024	Will Be Tested
Testing on different browsers	30.05.2024	Will Be Tested
Penetration Tests for different senarios	03.06.2024	Will Be Tested

6. Procedures

We planned to implement different test methods in a specific order.

- In early stages in software development, we implement Unit Tests. So, smallest units and components of our application will be verified and bugs and errors will be detected. For instance, how specific input interacts with the database in backend side.
- Integration testing will follow Unit Testing and it will be implied in the middle stages of software development. Functionality of combinations of different components will be checked in this method. For instance, how backend returns to the client for a specific input after database operations.
- We planned to do System Testing after Integration testing. We will test whole system by using this method. For instance, student registration, login, displaying a course material (a pdf file), download the file and logout.
- We planned to do Performance Testing after System Testing. By Performance Testing, we will evaluate how Peer2Share web application works under various loads such as many users' interaction with the system. This method will allow us to under which parts should we optimize and improve user experiences.