TED UNIVERSITY

TEST PLAN REPORT

CMPE 492

Senior Design Project 2

Çağla Yıldız 1431404597

Deniz Zeynep Ersoy 2743946522

Tuğçe Nilay Öztekin 1004013348

TABLE OF CONTENTS

1. Introduction

1.1 Purpose of the Test Plan Report

The Test Plan Report is a thorough document that describes the goals, strategy, parameters, and timetable of the Lead the Board's testing efforts. Its goal is to give a thorough summary of the testing approach. Thus, this Test Plan Report outlines the testing approach, resources, and schedule for the Lead the Board project.

1.2 Scope

Depending on the project, system, and testing objectives, the testing's scope might change. The Lead the Board project is a gamification application designed to increase productivity in production through positive competition. The scope of our testing will include unit testing, usability testing, integration testing, user interface (UI) testing and user acceptance testing.

1.3 Features to be Tested

The features that will be tested include:

- User Registration
- Badge Earning Functionality
- Progress Tracking
- Leaderboard Functionality
- User Profile Management

1.4 Testing Methodology

Our testing methodology will include:

- **Unit Testing:** This will involve testing individual components of our system in isolation. The purpose of unit testing is to validate the behavior of each unit independently and ensure that it meets its specified requirements. On the scope of unit testing, we test methods.

- **Integration Testing:** This will involve testing how different components of our system work together. The purpose of integration testing is to uncover defects that may arise due to integration issues between different components and to ensure that they work together correctly as a cohesive system. On the scope of integration testing, we conduct tests on our modules.

- **User Acceptance Testing:** User interface elements must be tested to ensure that they are presented correctly, react to user interactions effectively, and follow all design criteria and guidelines. It examines visual components across many displays and devices for alignment, layout, responsiveness, and consistency. Additionally, verifying user input, managing errors, and navigating the user interface are all included in UI testing. Since we don't have real users, we act as user and conduct the test in that way.

- **Usability Testing:** Usability testing is a technique used to assess a product, system, or interface's usability and effectiveness from the viewpoint of its target users. It focuses on evaluating how simple it is for users to learn, comprehend, and use the system, as well as how satisfied they are with the overall user experience. The objectives of usability testing are to pinpoint problems with usability, suggest areas that may be improved, and confirm if the system satisfies user requirements and expectations. With usability testing, we may improve

customer happiness, boost product usability, and promote adoption by doing usability testing on our designs.

- **User Interface (UI):** A software testing method known as user interface (UI) testing focuses on analyzing a system's graphical user interface elements, such as buttons, menus, forms, and other interactive features. The basic goal of UI testing is to make the user interface functionally sound aesthetically pleasing and offer a satisfying user experience. User interface elements must be tested to ensure that they are presented correctly, react to user interactions effectively, and follow all design criteria and guidelines. It examines visual components across many displays and devices for alignment, layout, responsiveness, and consistency. Additionally, verifying user input, managing errors, and navigating the user interface are all included in UI testing.

1.5 Test Schedule

Our test schedule can be found as follows:

Tests	Weeks
Unit Testing	Week 3-4
Integration Testing	Week 5-8
User Acceptance Testing	Week 9-10
Usability and User Interface Testing	Week 10-11

1.6 Roles and Responsibilities

The roles and responsibilities for testing are as follows:

Responsibilities/ Roles	Deniz Zeynep Ersoy	Çağla Yıldız	Tuğçe Nilay Öztekin
Unit Testing	✓	✓	✓
Integrity Testing		✓	✓
User Acceptance Testing	✓		✓
Usability and User Interface Testing	✓	✓	

2. Unit Testing

2.1 Objectives

The objectives of unit testing are to ensure that individual methods of our system function correctly and meet the design goals.

2.2 Test Cases and Scenarios

We will create test cases and scenarios for each component of our system, including:

- **User Registration:** Verify that users can successfully register and log in to the system.
- **Badge Earning Functionality:** Verify that users can earn badges for completing tasks and that the badge is displayed on their profile.
- **Progress Tracking:** Verify that progress is accurately tracked and displayed on the user's profile.
- **Leaderboard Functionality:** Verify that the leaderboard accurately displays user rankings based on productivity.
- **User Profile Management:** Verify that users can update their profile information and view their progress.

3. Integration Testing

3.1 Objectives

The objectives of integration testing are to ensure that different components of our system work together correctly and meet the design goals.

3.2 Test Cases and Scenarios

We will create test cases and scenarios for how different components of our system interact with each other, including:

- **User Registration + Badge Earning Functionality:** Verify that badges are correctly awarded to users who complete registration.
- **Progress Tracking + Leaderboard Functionality:** Verify that progress is accurately tracked and displayed on the leaderboard.

4. User Acceptance Testing

4.1 Objectives

The objectives of user acceptance testing are to ensure that our system meets user needs.

4.2 Test Cases and Scenarios

We will have actual users test our system using a set of predefined tasks, including:

- Registering for an account
- Earning a badge
- Viewing progress on their profile
- Interacting with the leaderboard

5. Control Procedures

5.1 Version Control Software

We will use Git to manage changes made during development and ensure that all code changes are properly documented. We chose Git because of some of the advantages it has over the other Version Control Software.

Git is efficient at managing processes including branching and merging. Developers can simply build, switch between, and combine branches because to its lightweight branches. This makes it perfect for feature branching, concurrent development, and keeping various codebase versions.

Git is a distributed version control system (DVCS), which means that every developer has a full local copy of the repository on their computer. This enables developers to work individually and offline before synchronizing their modifications with the main repository. In distributed or remote development environments, it provides more flexibility and quicker operations.

Moreover, Git is renowned for its efficiency and effectiveness. It is designed to effectively maintain history and modifications while handling massive codebases. Large-scale projects often use it because operations like committing, branching, and moving between branches can often be done quickly. Thus, we chose Git as our Version Control Software to work collaboratively in Lead the Board.

6. Conclusion

In conclusion, our testing plan includes several types of testing to ensure that our Lead the Board application is secure, scalable, and meets user needs. We will use a password-based authentication system, keep access logs, and test for vulnerabilities to ensure data security. We will also use an event-driven control design pattern to make the application more reactive and flexible. Our testing plan includes unit testing, system and integration testing, performance testing, and user acceptance testing. Finally, we will use Git for version control and assign specific roles and responsibilities for each type of testing.