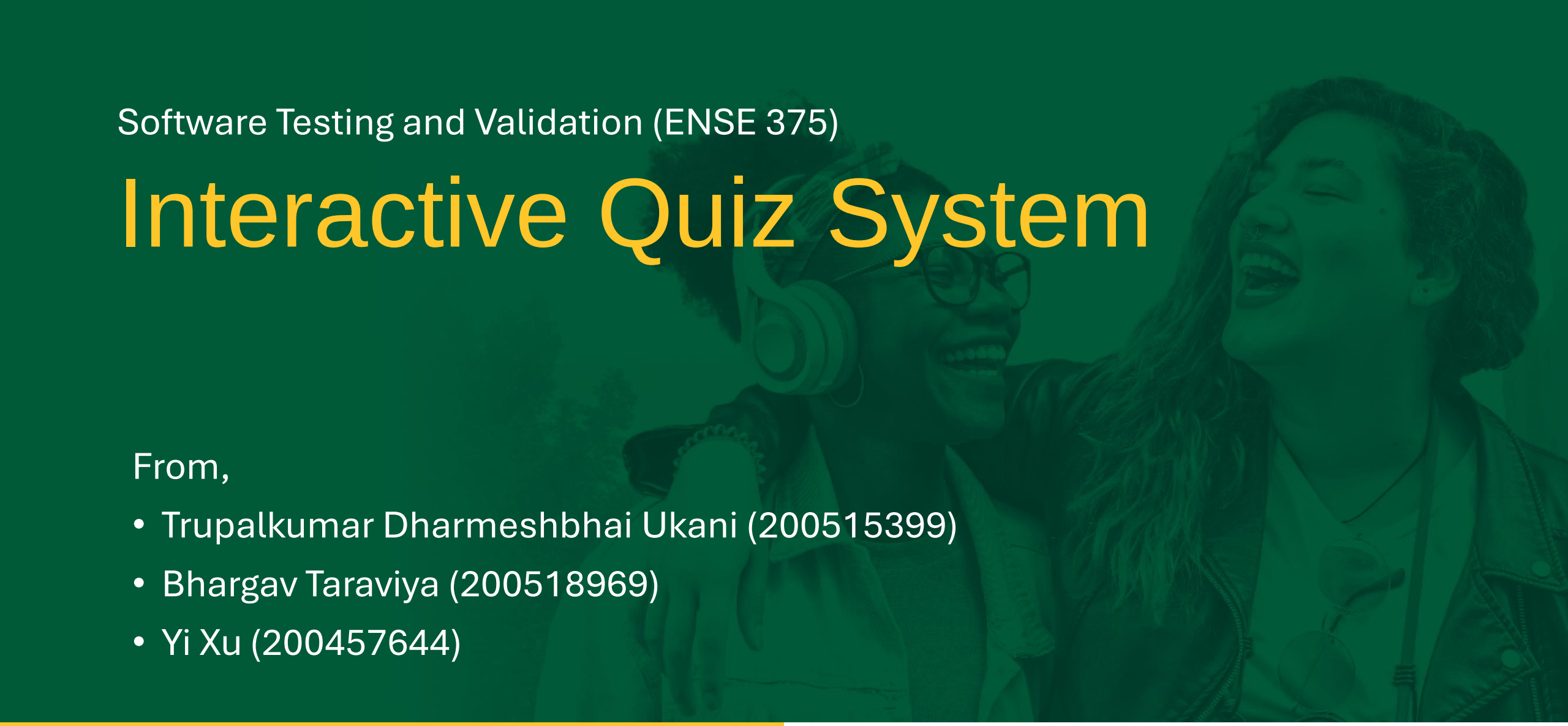
Software Testing and Validation (ENSE 375)

Interactive Quiz System



From,

- Trupalkumar Dharmeshbhai Ukani (200515399)
- Bhargav Taraviya (200518969)
- Yi Xu (200457644)

31st July 2024

Software Testing and Validation (ENSE 375)

University of Regina | Go far, together.

Agenda

- Introduction
- Problem Definition
- Design Requirements
- Solutions
- Testing and Demonstration
- Project Management
- Conclusion and Future Scope

Introduction

This project focuses on designing and developing the Interactive Quiz System, a Java-based application for creating and managing quizzes. It is evident that a wide range of users will benefit from improved management of quizzes provided by the Online Quiz System, which focuses on a simple, efficient, and accessible service.

Our goal is to simplify the quiz administration process, allowing users to easily create, manage, and take quizzes.

Rationale:

- Traditional paper quizzes are time-consuming for feedback and hard to save for long periods.
- Motivation to provide an efficient and effective quiz management system that benefits users.

Problem Definition

Issues with Traditional Methods:

1. Inconvenience: Manual distribution and evaluation of paper-based quizzes is labor-intensive and error-prone.

2. Limited Accessibility: Traditional methods do not support remote access and management, hindering flexibility.

3. Complexity: Existing digital solutions are often complex and not user-friendly.

Benefits of a Digital Solution:

1.Improved Efficiency: Easy quiz creation, distribution, and evaluation.

2.User-Friendly Interface: Simplifies quiz management.

3.Immediate Result: Provides result immediately after the completion of the quiz.

Design Requirements - Functions

- Edit Quiz: Users can modify existing quizzes, including updating quiz name and questions.
- Create Quiz: Users can create new quizzes by adding various types of questions.
- Delete Quiz: Users can remove existing quizzes from the system.
- Add Question: Users can add new questions to quizzes, with different question types such as multiple choice and true/false.
- Edit Question: Users can update the content of existing questions in the quiz.
- Take Quiz: Users can participate in quizzes. The questions will be presented, and answers chosen by users will be recorded.
- Score Quiz: Automatically calculate and display score immediately after finishing the quiz, providing immediate feedback on performance.

Design Requirements - Objectives

- User-Friendly: The system should provide an intuitive and easy-to-navigate interface to improve user interaction and satisfaction.
- Efficient: The system should ensure a smooth and seamless user experience without causing errors.
- Responsive: The system should offer real-time scoring and immediate feedback upon quiz completion.
- Reliable: The system should function correctly and consistently under normal usage conditions, maintaining data integrity and accuracy.
- Maintainable: The system should be designed with clean and modular code, facilitating easy maintenance and future enhancements.
- Secure: The system should store user data and quiz content securely to prevent unauthorized access and ensure data privacy.

Design Requirements - Constraints

- Economic factors: Since there is no budgets for this project, we will keep this project in a small size.
- Regulatory compliance: Users are required to sign in to the system for using the service, and we will prevent personal account information from leaking.
- Reliability: The quiz system can be maintained and managed to ensure an excellent user experience.
- Ethics: The grading criteria for the quizzes are equal for all users.

Solutions – 1 & 2

Solution 1

Application Design: Basic structure focusing on essential functionalities such as a login system, adding quizzes, deleting quizzes, and viewing all quizzes.

Analysis: Supported only adding quizzes without the ability to add questions and answers, making the system incomplete and unmanageable through a CLI.

Reason for Rejection: Significant drawbacks and unmanageable for multiple users and quizzes.

Solution 2

Application Design: Included login system identifying admins and regular users, added functionalities for creating, editing, deleting, and managing quizzes.

Analysis: Improved over Solution 1 but had limitations without input validations, leading to potential risks and poor user experience.

Reason for Rejection: Required further enhancement for better usability and validation.

Solutions – Final Solution

Application Design: Comprehensive functionality for an online quiz system, with validations for username and password. Admins see error messages if any issues exist.

Analysis: Addressed limitations of previous solutions, providing a complete feature set with improved user experience.

Comparison of Solutions:

Functionalities/Solutions	Solution 1	Solution 2	Final Solution
Login System	✓	✓	✓
Admin and User Separation		✓	✓
Quiz Creation	✓	✓	✓
Quiz Deletion	✓	✓	✓
Quiz Execution		✓	✓
Quiz Edition		✓	✓
Input Validation			✓

Testing and Demonstration

Test Requirements for Each Testing Technique:

1. **Boundary Value Testing**: Ensure the system handles edge cases for username and password lengths.
2. **Equivalence Class Testing**: Validate consistent processing of valid and invalid inputs.
3. **Decision Tables Testing**: Verify correct system behavior under various input combinations.
4. **State Transition Testing**: Ensure proper transitions between different states (e.g., login/logout).
5. **Use Case Testing**: Test typical user scenarios like creating, taking, and scoring quizzes.

Testing and Demonstration

Test Cases to Satisfy the Test Requirements:

1.Boundary Value Testing:

1. Usernames: Test 0,1, 2,14, 15, 16 characters.
2. Passwords: Test 5 ,6,7, 14, 15, 16 characters.

2.Equivalence Class Testing:

1. Usernames: Valid (1-15), Invalid (<1, >15).
2. Passwords: Valid (6-15), Invalid (<6, >15).

3.Decision Tables Testing:

1. Login: Combinations of valid/invalid usernames and passwords.

4.State Transition Testing:

1. Verify login and logout transitions.

5.Use Case Testing:

1. Create quiz, take quiz, receive feedback.

Testing and Demonstration

Testing Results:

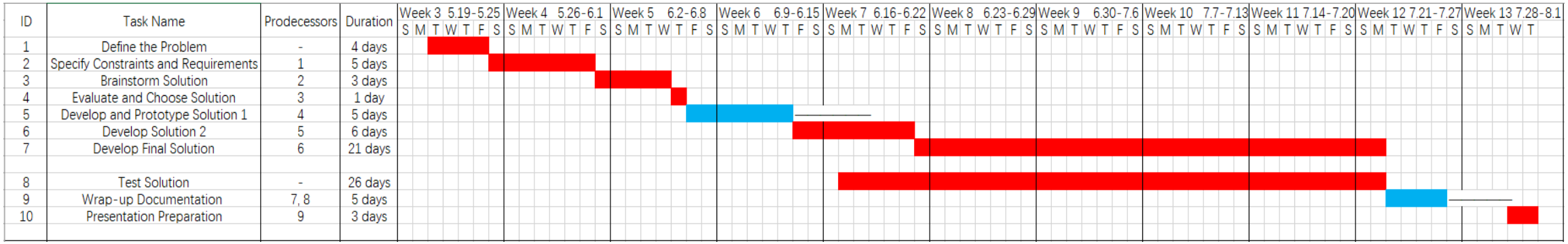
1. **Boundary Value Testing**: Passed, edge cases handled correctly.
2. **Equivalence Class Testing**: Passed, consistent processing of inputs.
3. **Decision Tables Testing**: Passed, correct behavior for all combinations.
4. **State Transition Testing**: Passed, seamless transitions.
5. **Use Case Testing**: Passed, meets user requirements efficiently.

Testing and Demonstration

DEMO

Project Management

- Gantt Chart



ID	Task Name	Prodecessors	Duration
1	Define the Problem	-	4 days
2	Specify Constraints and Requirements	1	5 days
3	Brainstorm Solution	2	3 days
4	Evaluate and Choose Solution	3	1 day
5	Develop and Prototype Solution 1	4	5 days
6	Develop Solution 2	5	6 days
7	Develop Final Solution	6	21 days
8	Test Solution	-	26 days
9	Wrap-up Documentation	7, 8	5 days
10	Presentation Preparation	9	3 days

- The critical path is in red, and the slack is the black lines connected to non-critical activities.
- Weekends are excluded from the schedule.
- Testing is integrated throughout the coding process.

Conclusion and Future Work

Conclusion:

• The ENSE 375 project taught us how to design and develop software step-by-step, enhancing our understanding of software testing through practical application.

• We achieved several key design functions and objectives, including:

- A login system that distinguishes between admins and regular users.
- Admin capabilities to create, delete, edit, and view quizzes.
- A quiz system that supports adding questions with multiple options.
- A testing system that provides notifications to users.
- User functionalities to take quizzes and view results.

Future Work:

• Implement a graphical user interface (GUI) for an enhanced user experience.

• Add animations related to quiz scores for better engagement.

Thank you!