EXP 8: To design test cases for performing black box testing for the given project

Black-Box Testing

Black-Box Testing Documentation

Project Name: Course Scheduler System
Testing Type: Black-Box Testing
Version: 1.0

1. Introduction

1.1 Objective

This document provides a detailed overview of the Black-Box testing performed on the Course Scheduler System. Black-Box testing is focused on evaluating the system's outputs based on given inputs without any concern for the internal workings or implementation details of the system. The goal of this testing is to ensure the system behaves as expected, satisfying the functional requirements and providing a seamless experience to users.

1.2 Scope

The scope of the black-box testing covers the following core modules of the Course Scheduler System:

- **Classroom Management**: Managing classroom resources, ensuring valid inputs for classroom creation and deletion.
- **Teacher and Subject Allocation**: Managing teacher assignments to subjects, ensuring valid assignments and conflict-free schedules.
- **Timetable Generation**: Automatically generating timetables based on given data, ensuring accurate scheduling with no conflicts.
- **User Authentication**: Validating user login, handling password resets, and ensuring correct session management.
- **Error Handling**: Ensuring that error messages are displayed in the correct contexts when invalid data is provided.

This testing does not involve examining the internal structure of the system but focuses purely on user-level interaction and the system's response.

1.3 Testing Environment

The black-box testing was performed in the following environment:

- **Operating System:** Windows 10
- **Web Server:** Apache 2.4
- **Database:** MySQL 8.0
- **Browser Compatibility:** Google Chrome, Mozilla Firefox

2. Testing Methodology

Black-Box testing was conducted using a functional-based approach, where the system's behavior was validated by providing different types of input and validating the corresponding output. The following testing methods were employed:

- **Boundary Testing**: This ensures that the system handles edge cases such as maximum capacity, minimum input values, and invalid data appropriately.
- **Usability Testing**: Ensuring the system is user-friendly and that error messages are clear, understandable, and helpful.
- **Performance Testing**: Ensuring that the system performs well even with large volumes of data, such as multiple classrooms, teachers, and subjects.
- **Security Testing**: Testing authentication and user access controls to ensure that unauthorized access is prevented.

Each module was tested by preparing specific input data, running the test cases, and comparing the system’s response against the expected outputs. The results were then documented.

3. Test Cases

Classroom Management

Test Case ID	Description	Input	Expected Output	Result
TC1	Add classroom with valid data	Classroom name: "Room 101", Capacity: 30	Classroom added successfully	Pass
TC2	Add classroom with missing name	Capacity: 30	Error message: "Classroom name is required"	Pass
TC3	Add classroom with invalid capacity	Name: "Room 102", Capacity: -10	Error message: "Capacity must be positive"	Pass
TC4	Add classroom with excessive capacity	Name: "Room 103", Capacity: 1000	Error message: "Capacity exceeds limit"	Pass
TC5	Delete classroom with valid ID	Classroom ID: 5	Classroom deleted successfully	Pass
TC6	Delete classroom that does not exist	Classroom ID: 999	Error message: "Classroom not found"	Pass

Subject Management

Test Case ID	Description	Input	Expected Output	Result
TC7	Add subject with valid data	Subject name: "Mathematics", Code: "MATH101"	Subject added successfully	Pass
TC8	Add subject with duplicate code	Subject name: "Physics", Code: "MATH101"	Error message: "Subject code already exists"	Pass
TC9	Delete subject with active allocations	Subject ID: 101	Error message: "Cannot delete subject with active allocations"	Pass

TC10	Update subject details	Subject ID: 101, New name: "Advanced Math"	Subject updated successfully	Pass

Teacher Management

Test Case ID	Description	Input	Expected Output	Result
TC11	Add teacher with valid data	Teacher name: "John Doe", Qualifications: "PhD in Mathematics"	Teacher added successfully	Pass
TC12	Add teacher with duplicate ID	Teacher ID: 101, Name: "Jane Smith"	Error message: "Teacher ID already exists"	Pass
TC13	Delete teacher with active subject allocations	Teacher ID: 101	Error message: "Cannot delete teacher with active subject allocations"	Pass
TC14	Update teacher information	Teacher ID: 101, New qualifications: "PhD in Physics"	Teacher information updated successfully	Pass

Timetable Generation

Test Case ID	Description	Input	Expected Output	Result
TC15	Generate timetable with complete teacher and subject data	Class: "Math101", Teacher: "John Doe", Subject: "Mathematics"	Timetable generated successfully	Pass
TC16	Generate timetable with missing teacher allocation	Class: "Math102", Teacher: None, Subject: "Physics"	Error message: "Teacher allocation required"	Pass
TC17	Generate timetable with conflicting teacher schedules	Class: "Math101", Teacher: "Jane Smith" with schedule overlap	Error message: "Conflict in teacher schedule"	Pass

User Authentication

Test Case ID	Description	Input	Expected Output	Result
TC18	Login with valid credentials	Username: "johndoe", Password: "password123"	User logged in successfully	Pass
TC19	Login with invalid credentials	Username: "janedoe", Password: "wrongpass"	Error message: "Invalid credentials"	Pass
TC20	Password reset request	Registered email: "johndoe@email.com"	Password reset link sent	Pass
TC21	Account lockout after multiple failed login attempts	Multiple failed login attempts	Error message: "Account locked due to multiple failed login attempts"	Pass

4. Summary of Results

Total Test Cases: 21
Passed: 21
Failed: 0
Remarks: All test cases for the Course Scheduler System passed successfully. The system correctly handles valid inputs and displays appropriate error messages for invalid scenarios. No critical defects were identified during the testing process.

5. Conclusion

Black-box testing indicates that the Course Scheduler System functions as expected, with all features performing correctly. Error handling is robust, and the user interface is intuitive, providing clear feedback to the user. The system performs well under various scenarios, including boundary cases, and handles invalid inputs appropriately.