Test Plan for Advanced Software Engineer

Mohammad Hesham Elsayed, Mark Mounir, Omar Hosny, Zeyad Abdelnasser Elzayaty
Supervised by:
Dr. Salwa Osama
Dr. Nada Shorim

May 21, 2024

Table 1: Document version history

Version	**Date**	**Reason for Change**
1.0	17-Mar-2024	Test Plan First version is defined.
1.1	17-May-2024	Test Scenarios "Add cart" and "Teacher form" have been updated.
1.2	18-May-2024	Test Scenarios "Add course" and "Add course content" have been added.
1.3	19-May-2024	Test Scenarios "Add to wishlist" has been added.

GitHub: https://github.com/MohammadHishamm/Advanced-Software-Eng

Contents

1 Introduction **3**
1.1 Purpose 3

2 Test Scenario "Add to cart" **3**
2.1 User Actions and Behaviors 3
2.2 Technical Aspects 3
2.3 Scenarios of System Failure 4
2.4 References 4
2.5 Test Cases 4

3 Test Scenario "Teacher form" **5**
3.1 Possible Users' Actions and Behaviors 5
3.2 Technical Aspects of the Requirement 5
3.3 Possible Scenarios of System Failure 5
3.4 Test Cases 6

4 Test Scenario "Add Course" **6**
4.1 Possible Users' Actions and Behaviors 6
4.2 Technical Aspects of the Requirement 6
4.3 Possible Scenarios of System Failure 6
4.4 Test Cases for "Add Course" Scenario 7

5 Test Scenario "Add Content" **7**
5.1 Possible Users' Actions and Behaviors 7
5.2 Technical Aspects of the Requirement 7
5.3 Possible Scenarios of System Failure 8
5.4 Test Cases for "Add Content" Scenario 8

6 Test Scenario "Add to Wishlist" **8**
6.1 Possible Users' Actions and Behaviors 8
6.2 Technical Aspects of the Requirement 9
6.3 Possible Scenarios of System Failure 9
6.4 Test Cases for "Add to Wishlist" Scenario 9

1 Introduction

1.1 Purpose

The purpose of this document is to outline the testing strategy and procedures for verifying the functionality of the "Add cart", "Teacherform", "Add course", "Add course content", "Add to wishlist" features in the application. It provides a structured approach to ensure that the features meet the specified requirements, functions correctly, and maintains expected behavior across different scenarios. Additionally, this document serves as a reference for stakeholders involved in the testing process, including developers, testers, and project managers, to understand the scope, objectives, and resources required for testing the "add cart" functionality effectively.

2 Test Scenario "Add to cart"

2.1 User Actions and Behaviors

- **User Interaction**:
  - Users navigate to the course page or product listing where the "add to cart" functionality is available.
  - They may be logged in or not logged in to the system.
- **Actions**:
  - Logged-In User: Clicks on the "Add to Cart" button/link associated with a specific course or product.
  - Not Logged-In User: Attempts to add a course to the cart without logging in.
- **Behaviors**:
  - Successful Addition: The course/product is added to the user's cart, and a confirmation message is displayed.
  - Existing Course/Product: If the course/product is already in the cart, the system prevents duplication and notifies the user.
  - User Not Logged In: If the user is not logged in, the system prompts the user to log in or register before adding items to the cart.

2.2 Technical Aspects

- **Frontend Interaction**:
  - The "Add to Cart" button triggers an HTTP request to the backend.
  - The request includes the course/product ID and user session information (if logged in).
- **Backend Processing**:

    - The backend controller (`cartController`) handles the request.
    - It retrieves the necessary data from repositories (e.g., `UserRepository`, `CoursesRepository`) to validate the request.
    - The system checks if the course/product is already in the user's cart and performs the necessary actions.

- **Data Persistence**:
    - The system updates the user's cart in the database (if applicable) to reflect the addition of the course/product.

- **Response**:
    - The system sends an appropriate response to the frontend indicating the success or failure of the operation.

2.3 Scenarios of System Failure

- **Invalid Course/Product ID**:
    - If the provided course/product ID is invalid or does not exist, the system should handle the error gracefully and notify the user.

- **Database Connection Failure**:
    - If there is a failure in connecting to the database or updating the cart, the system should log the error and display a generic error message to the user.

- **Session Management Issues**:
    - If there are issues with user session management (e.g., session timeout, session invalidation), the system should redirect the user to the login page or prompt them to re-authenticate.

- **Concurrency Issues**:
    - If multiple users attempt to add the same course/product to their carts simultaneously, the system should handle concurrency issues to prevent data inconsistency or race conditions.

2.4 References

- Use Cases in Software Requirements Specification (SRS)
- Sequence Diagrams in Software Design Document (SDD)

2.5 Test Cases

Test Cases for the scenario mention in section 2 shown in Table 2

Table 2: Test Cases for "Add to Cart" Scenario

Test Case ID	Test Case Description	Functional Requirement Code	Test Data	Expected Result
TC01	Add New Course to Cart	FR001	Valid course ID, Logged-in user session	Course added to cart successfully
TC03	Add Course to Cart (User Not Logged In)	FR001	Valid course ID, No user session	User not logged in

3 Test Scenario "Teacher form"

In this test scenario, we focus on testing the functionality of the "Teacher form" feature in the application. The purpose of this feature is to allow users to submit data via a form to become instructors in the system.

3.1 Possible Users' Actions and Behaviors

Users interact with the application by accessing the "Teacher form" page, filling out the required fields (such as personal information, qualifications, etc.), and submitting the form. They may also navigate away from the page without submitting the form.

3.2 Technical Aspects of the Requirement

The backend of the application should handle form submissions by validating the input data, updating the user's role to "instructor", and storing the instructor details in the database. It should also handle error conditions, such as invalid input data or database errors, gracefully.

3.3 Possible Scenarios of System Failure

- **Invalid Input Data:** If the user submits invalid or incomplete data, the system should provide appropriate error messages and prompt the user to correct the input.
- **Database Error:** In case of a failure while saving the instructor details to the database (e.g., database connection issues or constraints violation), the system should handle the error gracefully and notify the user about the issue.
- **Session Timeout:** If the user's session expires while filling out the form, the system should redirect them to the login page and prompt them to log in again. Any data entered in the form prior to the session timeout should be preserved.

This test scenario ensures that the "Teacher form" feature functions correctly and meets the requirements specified in the Software Requirements Specification (SRS) and Sequence diagrams in the Software Design Document (SDD).

3.4 Test Cases

Test Cases for the scenario mention in section 3 shown in Table 3

Table 3: Test Cases for "Teacher form" Scenario

Test Case ID	Test Case Description	Functional Requirement Code	Test Data	Expected Result
TC03	Submitting Valid Data	FR02	Valid user input, logged-in user session	Instructor role assigned to user, Instructor details saved successfully
TC04	Submitting Invalid Data	FR02	Invalid user input, logged-in user session	Error message displayed, user prompted to correct input

4 Test Scenario "Add Course"

This scenario involves users interacting with the "Add Course" feature, where instructors can add new courses via a form submission. The technical aspects include verifying form validation, session handling, and associating the course with the correct instructor. Possible system failures include issues with form validation, session management, and database interactions.

4.1 Possible Users' Actions and Behaviors

Users interact with the application by accessing the "Add Course" page, filling out the required fields (such as course title, description, etc.), and submitting the form. They may also navigate away from the page without submitting the form.

4.2 Technical Aspects of the Requirement

The backend of the application should handle form submissions by validating the input data, associating the course with the logged-in instructor, and storing the course details in the database. It should also handle error conditions, such as invalid input data or database errors, gracefully.

4.3 Possible Scenarios of System Failure

- **Invalid Input Data:** If the user submits invalid or incomplete data, the system should provide appropriate error messages and prompt the user to correct the input.

- **Database Error:** In case of a failure while saving the course details to the database (e.g., database connection issues or constraints violation), the system should handle the error gracefully and notify the user about the issue.
- **Session Timeout:** If the user's session expires while filling out the form, the system should redirect them to the login page and prompt them to log in again. Any data entered in the form prior to the session timeout should be preserved.

4.4 Test Cases for "Add Course" Scenario

Table 4: Test Cases for "Add Course" Scenario

Test Case ID	Test Case Description	Functional Requirement Code	Test Data	Expected Result
TC05	Submitting Valid Data	FR003	Valid course data, logged-in instructor session	Course details saved successfully, Course associated with instructor
TC06	Submitting Invalid Data	FR003	Invalid course data, logged-in instructor session	Error message displayed, user prompted to correct input
TC07	Session Timeout	FR003	Valid course data, session timeout	User redirected to login page, Data entered preserved

5 Test Scenario "Add Content"

This scenario involves instructors interacting with the "Add course content" feature, where instructors can add new course content via a form submission. The technical aspects include verifying form validation, session handling, and associating the course with the correct instructor. Possible system failures include issues with form validation, session management, and database interactions.

5.1 Possible Users' Actions and Behaviors

Instructors interact with the application by accessing the "Add Course Content" page, filling out the required fields (such as video playlist, title, etc.), and submitting the form. They may also navigate away from the page without submitting the form.

5.2 Technical Aspects of the Requirement

The backend of the application should handle form submissions by validating the input data, associating the course content with the correct course, and storing the content details in the database. It should also handle error conditions, such as invalid input data or database errors, gracefully.

5.3 Possible Scenarios of System Failure

- **Invalid Input Data:** If the user submits invalid or incomplete data, the system should provide appropriate error messages and prompt the user to correct the input.
- **Database Error:** In case of a failure while saving the course content details to the database (e.g., database connection issues or constraints violation), the system should handle the error gracefully and notify the user about the issue.
- **Session Timeout:** If the user's session expires while filling out the form, the system should redirect them to the login page and prompt them to log in again. Any data entered in the form prior to the session timeout should be preserved.

5.4 Test Cases for "Add Content" Scenario

Table 5: Test Cases for "Add Conten" Scenario

Test Case ID	Test Case Description	Functional Requirement Code	Test Data	Expected Result
TC05	Submitting Valid Data	FR004	Valid course content data, logged-in instructor session	Course content details saved successfully, Content associated with correct course
TC06	Submitting Invalid Data	FR004	Invalid course content data, logged-in instructor session	Error message displayed, user prompted to correct input
TC07	Session Timeout	FR004	Valid course content data, session timeout	User redirected to login page, Data entered preserved

6 Test Scenario "Add to Wishlist"

This scenario involves students interacting with the "Add to Wishlist" feature, where students can add a course to their wishlist via a form submission. The technical aspects include verifying session handling, course retrieval, wishlist association, and response messages. Possible system failures include issues with session management, course retrieval, wishlist management, and database interactions.

6.1 Possible Users' Actions and Behaviors

Students interact with the application by accessing the "Add to Wishlist" page, selecting a course to add to their wishlist, and submitting the request. They may also navigate away from the page without adding the course to the wishlist.

6.2 Technical Aspects of the Requirement

The backend of the application should handle course addition to the wishlist by verifying the session, retrieving the course and student information, checking if the course is already in the wishlist, and saving the updated wishlist to the database. It should also handle error conditions, such as invalid session, course retrieval failures, and database errors, gracefully.

6.3 Possible Scenarios of System Failure

- **Invalid Session:** If the user is not logged in or the session is invalid, the system should provide an appropriate message indicating that the user needs to log in.
- **Course Retrieval Failure:** In case of a failure while retrieving the course details (e.g., course not found), the system should handle the error gracefully and notify the user about the issue.
- **Database Error:** If there is a failure while saving the wishlist details to the database (e.g., database connection issues or constraints violation), the system should handle the error gracefully and notify the user about the issue.
- **Course Already in Wishlist:** If the course already exists in the user's wishlist, the system should notify the user that the course is already in the wishlist.

6.4 Test Cases for "Add to Wishlist" Scenario

Table 6: Test Cases for "Add to Wishlist" Scenario

Test Case ID	Test Case Description	Functional Requirement Code	Test Data	Expected Result
TC01	Adding Valid Course to Wishlist	FR001	Valid course ID, logged-in student session	Course added to wishlist successfully, success message displayed
TC02	Course Already in Wishlist	FR001	Course ID already in wishlist, logged-in student session	Error message displayed indicating course already in wishlist
TC03	User Not Logged In	FR001	Valid course ID, no logged-in session	Error message displayed indicating user needs to log in
TC04	Course Not Found	FR001	Invalid course ID, logged-in student session	Error message displayed indicating course not found
TC05	Database Error During Save	FR001	Valid course ID, logged-in student session, simulate database error	Error message displayed indicating failure to save wishlist