Tazmeen Afroz
22P-9252
Assignment 3

13 May 2024

Testing Strategy for Virtual Student Support Assistant (VSSA)

1. **Type of Testing:**

   - **Unit Testing:** Verifies individual software units in isolation. This type of testing is crucial for identifying issues early in the development process. It can be both static (examining code for errors) and dynamic (executing the unit to test its functionality).
   - **Component Testing:** Tests the interaction and integration between multiple software units. This testing ensures that different parts of the software work together as expected. It can involve both static and dynamic.
   - **System Testing:** Validates the entire VSSA application's functionality. This testing ensures that the system as a whole meets the specified requirements. It primarily involves dynamic testing, where the system is executed under various scenarios to test its functionality and usability.

1. **Strategy:**

   - **Black-Box Testing:** Focuses on the functionality of the software from the user's perspective. It simulates user interactions to validate the system's functionality without examining the internal code structure. This approach is primarily used in dynamic testing scenarios, such as system testing.
   - **White-Box Testing:** Involves examining the internal structure of the software, including the code and data flow. This testing is aimed at identifying logical errors and ensuring that the software behaves as expected under various conditions. It can be both static (static analysis) and dynamic (dynamic analysis).
   - **Static Testing:** This approach involves reviewing the software without executing it. It includes activities such as code reviews, static

analysis tools, and documentation reviews. Static testing is crucial for identifying potential issues early in the development process, such as logical errors, security vulnerabilities, and compliance issues. In the context of VSSA, static testing is applied in unit testing and component testing to ensure the correctness of individual units and their interactions.

- **Dynamic Testing:** Dynamic testing involves executing the software to test its behavior under various conditions. It includes activities such as functional testing, integration testing, and system testing. Dynamic testing is essential for validating the functionality, performance, and usability of the software. In the VSSA testing strategy, dynamic testing is primarily used in system testing to ensure the entire application functions as expected and meets user requirements.

1. **Reasons for Selection:**

   - Unit Testing (White-Box Testing - Primarily Static): Identifies potential issues early in the development process by examining the code for errors and logical flaws.
   - Component Testing (White-Box & Black-Box Testing): Ensures proper data exchange and functionality between integrated units by both examining the code for issues and executing the components to test their behavior.
   - System Testing (Black-Box Testing - Primarily Dynamic): Validates the entire VSSA application's functionality and identifies usability issues, integration problems, or inconsistencies with the SRS before deployment by simulating user interactions and testing the system's behavior under various scenarios.

Test Cases:

Test Case 1: **Information Access - (Unit Testing - White-Box)**

- **Input:** Test code with a query "What are the deadlines for registration?"
- **Test Steps (Static):** Review the code responsible for handling information access functions.
- **Expected Output:** VSSA displays a clear and accurate message with registration deadlines.
- **Objective:** Verify the unit responsible for handling information access functions correctly.

Test Case 2: **Course Selection - Prerequisite Checking (Component Testing - White-Box)**

- **Input:** Test the component responsible for course selection and prerequisite checking with various course data.
- **Test Steps (Static):** Review the code for retrieving course information and prerequisite data from the database.
- **Expected Output:** VSSA correctly identifies and displays course prerequisites.
- **Objective:** Verify the functionality of the component responsible for prerequisite checking during course selection in VSSA.

Test Case 3: **User Authentication and Security (System Testing - Black-Box)**

- **Input:** Users attempt to log in with valid and invalid credentials.
- **Test Steps (Dynamic):** Simulate login attempts with valid usernames and strong passwords. Attempt logins with invalid usernames or passwords (typos, incorrect case).
- **Expected Output:** VSSA grants access to authorized users and rejects invalid login attempts. User data remains secure with encryption protocols.
- **Objective:** Confirm VSSA adheres to security best practices to protect user data and prevent unauthorized access.