Sample Manual Testing Exam (CA 2) – Xray & Jira Practical Task

Duration: 1 hour 30 minutes

Instructions:

- Use **Xray in Jira** to create and manage test cases.
- You will be testing an application based on the **Software Requirements Specification (SRS)** provided.
- Although you do not have access to the actual application, you must cover all the testing types by **designing test cases, linking them together logically, and managing defects.**
- Ensure that all test cases are linked properly in Xray, demonstrating how different types of testing connect.
- Submit your work by **organizing everything in Xray and providing a Jira project link or report export.**

Section 1: Test Case Design & Linkage (60 Marks)

☞ **Task:** Design and document test cases for the following **10 testing types.**

📌 Step 1: Create test cases in Xray

- Write at least **one test case for each testing type** based on the SRS.
- Ensure that each test case follows the format: **Test ID, Test Summary, Pre-conditions, Steps, Expected Result.**
- Link each test case to a related **requirement** in Xray.

📌 Step 2: Establish linkages between test types

- Ensure **unit tests** connect to **integration tests** (unit-tested components should interact).
- **Integration tests** should lead into **system tests** (to check the whole system).
- **System tests** should feed into **regression tests** (to ensure stability after changes).
- **Performance, load, security, and compatibility tests** should support the overall system.
- **UAT tests** should ensure business needs are met at the final stage.

☞ **Expected Deliverables:**

- **Test cases documented in Xray** and linked to relevant requirements.
- **Traceability established** between different test types.

Section 2: Test Execution Planning (20 Marks)

☞ **Task:** Create a **Test Execution Plan** in Xray.

📌 Step 1: Define execution priorities

- Assign **priority levels** to test cases (High, Medium, Low).
- Ensure that **critical functionalities** are tested first.

📌 Step 2: Organize test execution cycles

- Group test cases into **execution cycles** based on testing types.
- Ensure test cycles logically **progress from unit to UAT.**
- Create a structured plan showing the execution sequence.

☞ **Expected Deliverables:**

- **Test Execution Plan created in Xray.**
- **Execution cycles structured to reflect the logical testing flow.**

Section 3: Defect Logging & Linking (20 Marks)

☞ **Task:** Log a defect in Jira based on a failed test case.

📌 Step 1: Identify a test failure

- Choose any **test case** from the above list and assume it fails.
- Document the failure **clearly in Xray.**

📌 Step 2: Report the defect in Jira

- Create a Jira bug and include:
  - **Defect summary & description.**
  - **Steps to reproduce.**
  - **Expected vs actual result.**
  - **Severity level (Critical, High, Medium, Low).**
  - **Attachments (mocked screenshots).**

📌 Step 3: Link the defect

- **Link the defect** to the **failed test case** in Xray.
- **Trace the defect back** to the requirement it affects.

INTRODUCTIONS

How to Log a Defect Based on the SRS Alone

📌 *Step 1: Identify a Test Failure*

- Since you don't have the real application, assume that a test case **fails** based on what is expected from the SRS.
- Example: If the SRS states that **"The login button should redirect users to the dashboard upon successful authentication,"** you can assume a failure where clicking the login button **does nothing** or **redirects to the wrong page.**
- Document this failure **clearly in Xray** as a test case failure.

📌 *Step 2: Report the Defect in Jira*

👉 Create a bug in Jira and include the following:

- **Defect Summary:**
  Login button does not redirect to the dashboard after successful authentication.
- **Description:**
  As per the SRS, after entering valid credentials and clicking the login button, the user should be redirected to the dashboard. However, clicking the login button does not redirect the user anywhere.
- **Steps to Reproduce:**
  1. Navigate to the login page.
  2. Enter valid username and password.
  3. Click the login button.
  4. Observe the result.
- **Expected vs. Actual Result:**
  - **Expected:** User should be redirected to the dashboard.
  - **Actual:** Clicking the login button does nothing (or redirects to the wrong page).
- **Severity Level:** *High* (since login is a critical function).
- **Attachments:**
  - Add a **mocked screenshot** (e.g., an image of a login screen with a failed login attempt or a text-based error message).

📌 *Step 3: Link the Defect in Xray*

- **Link the defect** to the failed test case in Xray.
- **Trace the defect back** to the **SRS requirement** stating how login should work.

👉 Example of linking in Xray:

- Test Case: *Verify login functionality* → **FAILED**
- Linked Defect: *BUG-123: Login button not working*
- Linked Requirement: *REQ-001: Login Functionality* (from the SRS)