**Unit Test Plan**

|  |  |
| --- | --- |
| Module #: | Application #: |
| Tester: Willy | Test Manager: Biombo Arielle/Mende Jennifer |
| **Module Overview** | |
| Describe the purpose of the module.  The purpose of the User Management module is to handle user-related operations such as creating new user accounts, retrieving user details, and fetching past purchase history. This module ensures that users are managed efficiently and that their data is stored and retrieved accurately. | |
| **Module Inputs** | |
| Describe the inputs to the modules   **createUser(id, name)**: Takes a user ID and name to create a new user.   **getUser(id)**: Takes a user ID to retrieve user details.   **getUserDataBeforeValidation(id, externalService)**: Takes a user ID and an external service to retrieve user data before validation. | |
| **Module Outputs** | |
| Describe the outputs from the module.   **createUser**: Returns a promise that resolves with the user data if successful.   **getUser**: Returns user details based on the provided ID.   **getUserDataBeforeValidation**: Returns user data fetched from an external service before performing validation. | |
| **Logic Flow** | |
| Include a logic flow diagram. | |
| **Test Data** | |
| List all test cases to be executed. | |
| Positive Test Cases   **TC\_COVERAGE\_01**:   * **Test**: Call createUser with valid ID and name. * **Expected Outcome**: User is created, and user data is returned.    **TC\_COVERAGE\_02**:   * **Test**: Call getUser with valid ID. * **Expected Outcome**: Correct user details are returned.    **TC\_COVERAGE\_03**:   * **Test**: Call getUserDataBeforeValidation with valid ID and external service. * **Expected Outcome**: User data is retrieved from the external service. | |
| Number each test case. Indicate the test to be performed and expected outcome | |
| Negative Test Cases   **TC\_NEGATIVE\_01**:   * **Test**: Call createUser with an invalid username (e.g., without '@'). * **Expected Outcome**: An "Invalid Username" error is thrown.    **TC\_NEGATIVE\_02**:   * **Test**: Call createUser with an already existing username. * **Expected Outcome**: A "User already exists" error is thrown.    **TC\_NEGATIVE\_03**:   * **Test**: Call getUser with an invalid or non-existent ID. * **Expected Outcome**: An error indicating that the user does not exist is thrown. | |
| Listin valid data selections | |
| Interface Modules | |
|  **Output Data**: The user creation process outputs user data.   **Data Input**: User details and IDs are input for retrieving user information.   **Internal Program Interface**: Interfaces with internal user data storage.   **External Program Interface**: Interfaces with external services for fetching user data. | |
| Test Tools   **Unit Testing Framework**: Vitest   **Mocking Library**: sinon (for mocking external services)   **Code Coverage Tool**: Istanbul | |
| Identify software used for unit testing.  Identify names and locations of software for future regression testing.   tests/unit/   tests/integration/ | |
| Archive Plan | |
| Specify location of archived data.  Define procedures required to obtain access to this data.   **Location of Archived Data**: ./archive/   **Access Procedures**: Data can be accessed by the testing team through the CI/CD pipeline artifacts storage. | |
| Updates | |
| Identify how unit test plan will be updated.  **Unit Test Plan Updates**: The unit test plan will be updated as new features are added or changes are made to the module. Updates will be tracked using version control in the repository. | |