**Software Test Plan (STP) for User Authentication**

**Health Tracker App**  
**Course: Software Validation and Testing, Winter 2025**  
**Submitted by:**

Saleem Yousef - 213523418  
Mohammed Khalil - 212134274

**1. Introduction**

**Purpose:**  
The purpose of this STP is to ensure the proper functioning, security, and performance of the user authentication feature in the Health Tracker App. This includes verifying that users can register, log in, log out, and reset their passwords securely and efficiently.

**Scope:**  
This STP covers the testing of the user authentication feature, including functional requirements like login, logout, and password reset, as well as non-functional requirements such as system response time and security.

**2. Test Items**

* User registration using email and password
* Login functionality
* Logout functionality
* Password reset process
* Error handling for incorrect credentials

**3. Features to be Tested**

[Describe the features that will be tested, including functional and non-functional requirements]

**Functional Requirements:**

* Users can successfully register an account using valid credentials.
* Users can log in with valid credentials.
* Users can log out securely.
* Users can reset their password when forgotten.

**Non-Functional Requirements:**

* Authentication response time should be within 2 seconds.
* The system should be resilient to SQL injections and other attacks.

**4. Approach**

[Describe the testing strategy, including the types of testing to be performed and the rationale for each]

**Testing Strategy and Rationale:**

* Unit Testing: Validate individual components such as registration and password reset functions.
* Integration Testing: Test the interaction between user authentication components and the backend database.
* System Testing: Ensure the complete authentication process works end-to-end.
* Security Testing: Test for vulnerabilities like SQL injection, brute force attacks, and session hijacking.

**5. Pass/Fail Criteria**

[Define the criteria for passing or failing the test cases]

**Pass Criteria:**

* All test cases execute successfully as per the expected results.
* No critical or high-severity defects remain unresolved.

**Fail Criteria:**

* Any test case fails due to unresolved defects or non-compliance with requirements.
* Performance metrics fall below defined thresholds.

**6. Test Deliverables**

* Test cases
* Test scripts
* Test data
* Test execution report
* Defect report

**7. Schedule**

[Provide a timeline for the testing activities. For example:]

| **Activity** | **Start Date** | **End Date** |
| --- | --- | --- |
| Write test Cases | 2025-01-10 | 2025-01-15 |
| Develop Test Scripts | 2025-01-16 | 2025-01-20 |
| Execute Unit Tests | 2025-01-21 | 2025-01-22 |
| Execute Integration Tests | 2025-01-23 | 2025-01-25 |
| Execute System Tests | 2025-01-26 | 2025-01-28 |
| Execute Performance Tests | 2025-01-29 | 2025-01-30 |
| Execute Security Tests | 2025-01-31 | 2025-02-01 |
| Compile Test Report | 2025-02-02 | 2025-02-03 |

**8. Environmental Needs**

[Describe the hardware, software, and test data requirements]

**Devices:**

* Android
* iOS devices

**Tools:**

* Testing frameworks (e.g., Selenium, Postman)
* Security testing tools (e.g., OWASP ZAP)

**Test Data:**

* Valid and invalid user credentials
* Sample email addresses for password reset

**9. Risks and Mitigation**

[List potential risks and how they will be mitigated]

**Risk 1:** Delays due to incomplete backend setup  
**Mitigation:** Collaborate with the backend team to ensure readiness before testing.

**Risk 2:** Security vulnerabilities in third-party libraries  
**Mitigation:** Perform a thorough dependency review before testing.

**Risk 3:** Insufficient test coverage  
**Mitigation:** Regularly review and update test cases.

**10. Approvals**

[Include the names and roles of those who prepared, reviewed, approved and approval date of the STP ]

Prepared by: Saleem Yousef, Mohammed Khalil  
Reviewed by: Saleem Yousef  
Approved by: Saleem Yousef  
Approval Date: 22/01/2025

**Software Test Description (STD) for Daily Food Intake Logging**

**1. Introduction**

[Write 3-4 sentences describing the purpose of the STD]

The purpose of this STD is to define detailed test cases for the Daily Food Intake Logging feature of the Health Tracker App. This feature allows users to log food items, view nutritional information, and add custom entries, aiding in dietary tracking and health improvement.

**2. Test Cases**

[Provide detailed test cases for the feature. Each test case should include the following fields: Test Case ID, Title, Objective, Pre-conditions, Test Steps, Expected Results and Post condition]

Test Case 1: Log Food from Database

* Test Case ID: TC001
* Title: Log an existing food item from the database
* Objective: Verify that users successfully log food items from the database.
* Pre-conditions:
  + The app is installed and functioning.
  + The user is logged into their account.
  + The food database is populated with test entries.
* Test Steps:

1. Open the "Log Food" section of the app.
2. Search for an existing food item, e.g., "Banana."
3. Select the food item.
4. Specify quantity (e.g., 1 medium banana).
5. Save the entry.

* Expected Results:
  + The food item is successfully logged and displayed in the daily food log with accurate nutritional information.
* Post-conditions:
  + The food entry is saved in the user's daily log.

Test Case 2: Add Custom Food Entry

* Test Case ID: TC002
* Title: Add and log a custom food item
* Objective: Ensure users can add custom food items with nutritional details.
* Pre-conditions:
  + The user is logged into their account.
* Test Steps:

1. Open the "Log Food" section of the app.
2. Select "Add Custom Food."
3. Enter the food details (e.g., "Homemade Sashimi – 28g").
4. Save the custom entry.
5. Log the custom food item to the daily log.

* Expected Results:
  + The custom food item is saved and can be logged with correct nutritional information.
* Post-conditions:
  + The custom food entry appears in the user’s database and daily log.

Test Case 3: Search for a Food Entry

* Test Case ID: TC003
* Title: Search for a Food Item
* Objective: Verify that a user can search for a food item in the database.
* Pre-conditions:
  + The user is logged into the app.
  + The food database contains the items being searched for.
* Test Steps:
  + 1. Navigate to the “Food Logging” page.
    2. Enter a food name in the search bar.
    3. Select the food item from the search results.
    4. Enter the quantity.
    5. Click the “Save” button.
* Expected Results:
  + The food item is displayed with nutritional information.
  + The food item is logged successfully.
* Post-conditions:
  + The food item is stored in the database and displayed in the user’s daily log.

Test Case 4: View Food Log History

* Test Case ID: TC004
* Title: View Food Log History.
* Objective: Verify that a user can view their food log history.
* Pre-conditions:
  + The user is logged into their account.
  + The user has logged at least one food entry.
* Test Steps:
  + 1. Navigate to the “Food Log History” page.
    2. Select a date from the calendar.
* Expected Results:
  + The food items logged on the selected date are displayed.
  + The total calories for the day are calculated and displayed.
* Post-conditions:
  + The user can view their food log history for the selected date.

Test Case 5: Edit a Logged Food Item

* Test Case ID: TC005
* Title: Edit a Logged Food Item
* Objective: Verify that a user can edit a logged food item.
* Pre-conditions:
  + The user is logged into the app.
  + The user has logged at least one food item.
* Test Steps:
  + 1. Navigate to the “food Log History” page.
    2. Select a logged food item.
    3. Click the “Edit” Button.
    4. Modify the quantity or nutritional details.
    5. Click the “Save” button.
* Expected Results:
  + The food item is updated with the new details.
  + The total calories for the day are recalculated.
* Post-conditions:
  + The updated food item is stored in the database and displayed in the user’s daily log.

Test Case 6: Delete a Logged Food Item

* Test Case ID: TC006
* Title: Delete a Logged Food Item
* Objective: Verify that a user can delete a logged food item.
* Pre-conditions:
  + The user is logged into their account.
  + The user has logged at least one food item.
* Test Steps:
  + 1. Navigate to the “Food Log History” page.
    2. Select a logged food item.
    3. Click the “Delete” button.
    4. Confirm the deletion.
* Expected Results:
  + The food item is removed from the log.
  + The total calories for the day are recalculated.
* Post-conditions:
  + The food item is removed from the database and no longer appears in the user’s daily log.

**3. Test Data**

[Provide the test data that will be used for testing]

**Food Database Entries:**

* Banana (Calories: 105, Carbs: 27g, Protein: 1.3g)
* Pasta (Calories: 200, Carbs: 40g, Protein: 8g)

**Custom Food Entry:**

* Food Name: Homemade Sashimi – one piece (28g)
* Calories: 35
* Protein: 5.3g
* Carbohydrates: 0.3g
* Fats: 1.2g

**4. Environmental Needs**

[Describe the hardware, software, and other environmental requirements for testing]

* **Devices:**
* android devices (version 10+)
* iOS devices (version 14+)
* **Backend Requirements:**
* Access to the food database.
* Stable internet connection.
* **Testing Tools:**
  + Selenium for UI testing.
  + Postman for API testing.
* **Test Environment:**
  + Development environment with mock data for testing.

**5. Conclusion**

This Software Test Description (STD) provides a comprehensive set of test cases for the **Daily Food Intake Logging** feature of the Health Tracker App. By executing these test cases, we can ensure that the food logging functionality meets both functional and non-functional requirements. The test cases cover logging, searching, adding custom food items, and managing the food log history.

**Software Test Report (STR) for Exercise Activity Tracking**

**1. Test Summary**

**Objective:**

The objective of this testing effort was to verify the functionality, performance, and reliability of the **Exercise Activity Tracking** feature in the Health Tracker App. This includes ensuring that users can log exercise activities, track calories burned, and view their exercise history.

**Scope:**

The scope of this testing effort includes:

* Logging exercise activities (e.g., running, cycling).
* Tracking calories burned based on user input (e.g., duration, intensity).
* Viewing exercise history and progress over time.

**Approach:**

* **Unit Testing**: To validate individual components (e.g., calorie calculation logic).
* **Integration Testing**: To ensure the exercise tracking feature interacts correctly with the backend database.
* **System Testing**: To verify the end-to-end functionality of the feature.
* **Performance Testing**: To measure the response time of the exercise tracking feature.
* **Security Testing**: To ensure the feature is secure against potential vulnerabilities.

**2. Test Results**

[Provide a summary of the test results in the table format below. Include the test case ID, description, execution date, outcome, and actual results for each test case]

| **Test Case ID** | **Description** | **Execution Date** | **Outcome** | **Actual Results** |
| --- | --- | --- | --- | --- |
| TC101 | Log a Running Activity | 2025-01-21 | Pass | Running activity was logged successfully, and calories burned were calculated. |
| TC102 | Log a Cycling Activity | 2025-01-21 | Pass | Cycling activity was logged successfully, and calories burned were calculated. |
| TC103 | View Exercise History | 2025-01-22 | Fail | Exercise history was not displayed for the selected date. |
| TC104 | Edit a Logged Exercise Activity | 2025-01-22 | Pass | Exercise activity was edited successfully, and calories burned were recalculated. |
| TC105 | Delete a Logged Exercise Activity | 2025-01-23 | Pass | Exercise activity was deleted successfully, and the history was updated. |

**3. Defects**

[Document any defects found during testing. Use the table format below for clarity]

| **Defect ID** | **Severity** | **Priority** | **Description** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| D001 | High | High | Exercise history not displayed for the selected date. | 1. Log an exercise activity. 2. Navigate to the exercise history page. | Exercise history should be displayed for the selected date. | Exercise history was not displayed. | Open |
| D002 | Medium | Medium | Incorrect calorie calculation for high-intensity activities. | 1. Log a high-intensity exercise activity (e.g., sprinting). 2. Check calories. | Calories burned should be calculated accurately based on intensity. | Calories burned were underestimated for high-intensity activities. | Open |

**4. Evaluation**

**Testing Objectives Met:**

* The majority of the testing objectives were met. The exercise logging, editing, and deletion functionalities worked as expected.
* However, the exercise history feature failed to display data, and the calorie calculation for high-intensity activities was inaccurate.

**Software Quality Assessment:**

* The **Exercise Activity Tracking** feature is mostly functional, but there are critical issues that need to be addressed before release.
* The feature meets functional requirements for logging and managing exercise activities but falls short in displaying history and calculating calories accurately.

**Limitations:**

* Testing was limited to a simulated environment, which may not fully replicate real-world usage.
* Only a subset of exercise activities (e.g., running, cycling) was tested due to time constraints.

**5. Conclusion**

The testing effort for the **Exercise Activity Tracking** feature was largely successful, with most test cases passing. However, critical defects were identified in the exercise history and calorie calculation functionalities. These issues must be resolved to ensure the feature meets user expectations and functional requirements.

**Recommendations:**

* Fix the exercise history display issue (Defect ID: D001).
* Review and correct the calorie calculation logic for high-intensity activities (Defect ID: D002).
* Conduct additional testing for other exercise activities (e.g., swimming, weightlifting) to ensure comprehensive coverage.
* Perform regression testing after fixes are implemented to ensure no new issues are introduced.