**Test Plan for H2Now Application**

**Introduction**

This test plan outlines sections that we will go through and strategies that we will use to test our H2Now application to ensure it works as expected, meets all requirements, and delivers usability to all users. Our goal is to ensure that the application measures water intake reliably, delivers correct information and statistics, and alerts to the end users. The test plan covers functional, performance, usability, and compatibility testing for the whole application and hardware components.

**Features to be Tested**

The following features will be tested:

1. **Authentication**

* Ensure ability to create accounts and log in to them and encrypted storage of credentials

1. **Bottle connection**

* Test if the bottle successfully being connected with the application

1. **Intake measurement**

* Verify correct intake measurement based on formula using bottle incline that is measured with incline sensor

1. **Information and statistics display**

* Verify correct information and statistics delivery to the end user. This includes current intake, daily streak of achieved goal, and goal achievement consistency graph

1. **Alert delivery**

* Test if alerts being delivered to the end users with proper information

1. **Bottle, account, and application management**

* Ensure the end user can control bottle (correct and reset intake, edit bottle’s name / goal, reset statistics, and delete bottle), manage account (change information, log out, and delete it), and change application’s preferences (like units, light / dark mode, notifications)

**Testing Strategies**

1. **Functional Testing**

* Test application’s core features and verify that the results are as expected
* Strategy:
  + Test the account creation and bottle connection, and their management
  + Simulate different bottle use scenarios
  + Verify that application logs proper intake amount based on formula
  + Test alerts delivery

1. **Performance Testing**

* Measure application’s performance under different conditions and long use
* Strategy:
  + Conduct extended session of bottle use
  + Measure the application’s consistency and functioning within the time

1. **Usability Testing**

* Ensure that the application’s interface is easy to navigate and is intuitive
* Strategy:
  + Conduct tests with other end users to assess the understanding of an interface, the ease of navigation within the app, and use performance
  + Gather detailed feedback about their whole experience in the application

1. **Compatibility Testing**

* Verify that the application works as expected across different devices and environments
* Strategy:
  + Test the application on all machines (phones (iOS, Android) and computers (Windows, macOS, Linux))
  + Assess the reliability on network under various network conditions

**Test Case Scenarios**

1. Authentication

* Verify successful account creation, log in, and the secure storage of credentials
* Steps:

1. Create an account
2. Log in to the created account
3. Verify authentication system and that credentials are encrypted in the database
4. Bottle connection

* Test bottle’s connection with the application and account
* Steps:

1. Add new bottle and specify name and goal
2. Connect the bottle
3. Ensure connection between application and bottle
4. Intake measurement

* Verify correct intake measurement based on formula
* Steps:

1. Open the bottle and take a few sips of water
2. Verify calculations and measurements
3. Ensure the proper delivery of new intake information
4. Information and statistics display

* Verify correct information and statistics delivery to the end user
* Steps:

1. Take some sips of water until the goal is reached
2. Wait for the new intake calculations delivery
3. Verify correct goal achievement consistency graph rebuilding and daily streak update
4. Alert delivery

* Test if alerts being delivered with proper information
* Steps:
  + 1. Open the bottle and take a few sips of water
    2. Stop drinking, put the bottle, and ensure motor did the vibration
    3. Ensure that motor is vibrating to remind the end user to drink water as well

1. Bottle, account, and application management

* Verify rest of the features within the application
* Steps:

1. Test the bottle’s information change, reset, and deletion along with disconnection
2. Test the account’s data and credential change, and account’s deletion
3. Test the application’s preferences change features like light / dark mode, measurement units, and notifications

**Pass and Fail Criteria**

Application passes the tests if it meets the requirements, works as expected, meets the results, and behaves equally across different platforms. Otherwise, if the outcomes are not as expected, or application is crashing - the application fails the tests.

**Resources and Tools**

**Front-End**: React

**Back-End**: Flask, MySQL, AWS

**Hardware**:

1. MPU-6050 module (Accelerometer and gyroscope that detects motion, tilt, and rotation)
2. Raspberry Pi (Sensor data reader and processor. Server host)
3. Breadboard and jumper wires (Prototype wiring)
4. Small vibration motor / buzzer / LED (Feedback for drinking reminders and confirmation)

**Test devices**: Mobile phone and laptop

**Testing tools**:

* Visual Studio Code for debugging and performance test
* Breadboard for hardware