# Night's Watch
# Software Design

## CSCI-P465/565 (Software Engineering I)

### Project Team

**Shantanu Kotambkar**

**Rahul Velayutham**

**Varun Machingal**

**Adarsh Bhandary**

---

# 1. Overview

This section provides an overview of the testing approach used to verify the software product.

## 1.1 Test Objectives

Testing approach used in testing the system should check for security vulnerabilities in the system, which might be improper technique to encrypt password, or incorrect transfer of data. Testing should identify the correctness of the overall output of the system, along with the restrictions on the system.

In Night's watch, testing approaches should point out the flaws in authentication, retrieval or posting of data, real time rendering mechanism, if there exists any.

## 1.2 Test Environment

The environment we explored and discussed for testing in Django was Unit test module built in to the python standard library. However, like the sprint 1, sprint-2 and sprint 3 were limited, we did more of manual oriented testing.

Operating systems – Mac OS X, Windows 10 were used while testing.

Hosting options- We have hosted on IU Servers ( SILO).

### 1.3 Test Personnel

Rahul Velayutham and Varun Machingal have been involved in the manual testing in this sprint.

### 1.4 Acceptance Criteria

Acceptance Criteria is to pass all the test cases mentioned in section 2.

# 2. Test Cases

The test cases are the partitioning of the verification of the software into manageable sections. Often these sections correspond to the set of active use case scenarios, but can be organized as the test developer sees fit. Test cases should be in place to cover all of the software verification methods. Even non-execution based testing methods (i.e., inspection/analysis) may be detailed here. The intent is for the test case procedures to provide a repeatable verification of the software specification.

For each test case describe the following:
1. Username registration:
   a. Description:
      - In this test case, the test must be carried out to check for an already existing user. This is important to store data distinctively for distinct user.

   b. Initial Condition:
      - The user should enter the data required in registering the user.

   c. Input data to be tested: Username.

   d. Procedure:
      - The SQL table is expected to give an error as a duplicate record is being added. Username should be a Unique key in SQL Database.

2. Password registration:
   a. Description:
      - In this test case, the password that is being registered must might certain criteria's. They are:
         o Password cannot include username
         o Password must be at least 8 characters long
         o Password must include alphabets.

3. Username check at login:

a. Description:
- In this test case, the test must be carried out to check the validity of the user.

b. Acceptance criteria:
- The username should match a username present in the database thus indicating validity of user.

4. Password check at login:
   a. Description:
   - In this test case, the password is to be checked. The password should match the password associated with the username in the database.
   b. Acceptance criteria:
   - The password should be encrypted and decrypted to check for correct password, if the password doesn't exactly match, it should throw an incorrect password/username error.

Sprint – 2

5. Submitting without tokens
   a. Description:
       i. In this test case, we will try to submit the form without tokens
   b. Acceptance Criteria
       i. The from should not accept without a token key being entered.

6. Invalid Tokens:
   a. Description:
       i. In this test case, all ways in which invalid tokens are taken from users is rejected
   b. Acceptance Criteria
       i. The from should not accept with an invalid token entered.

Sprint – 3

7. Add Sensors:
   a. Description:
       i. In this test case, the sensor to be added should have proper sensor number, and that needs to be checked.
   b. Acceptance Criteria:
       i. The form should return an error message.

8. Manage User and Sensors info:
   a. Description
       i. In this test case, we will check if deletion of sensors or updating user without selecting anything works
   b. Acceptance criteria:
       i. The form should return a message no sensor detected and no user selected respectively.

9. Display Weather Information:
    a. Description:
        i. Check if appropriate weather.txt file is present.
    b. Acceptance criteria:
        i. An error message saying file is missing should get printed
10. Data accumulation:
    a. Description:
        i. Check if new data is pulled correctly
            a. set date ahead of time
            b. set date behind time
    b. Acceptance criteria:
        i. on the same date a pull should not happen

11. Accuracy of data:
    a. Description:
        i. Data is pulled from weather-mine accurately and for Indiana Bloomington.
    b. Acceptance criteria:
        i. Manually test and verify the data.
12. Display Map Information Window:
    a. Description:
        i. Verify that following data is displayed correctly:
            a. sensor image
            b. latitudes and longitudes
            c. light intensity
            d. battery level

    b. Acceptance criteria:
        i. Manually test the data and verify its success.

13. Meter and sensor light intensity widget:
    a. Description:
        i. Verify sensor light intensity widget diagrams shows the right light intensity level on the meter:
    b. Acceptance Criteria:
        i. Manually test the meter reading on display along with the corresponding light widget.

Sprint 4:

14. Display Other User Profiles:
    a. Description:
        i. Display other user profiles apart from the logged in user.
    b. Acceptance Criteria:
        i. Logged in user should not be able to edit the other user profile.
        ii. The other user profile should not override current user profile.

15. Search users using Username, First Name, or Last Name:
    a. Description:
        i. Display a list of users matching the searched user query.
    b. Acceptance Criteria:
        i. Upper case and Lowercase distinction should be avoided.
        ii. Search should display every user in search list that may contain the query string.
        iii. It should give an error to user, if no such user exists!

16. Search users using Keywords:
    a. Description:
        i. Display a list of users matching the searched Keyword.
    b. Acceptance Criteria:
        i. Upper case and Lowercase distinction should be avoided.
        ii. Search should display every user in search list that may contain the query string.
        iii. It should give an error to user, if no such user exists!
        iv. It should search every possible field in user Profile, like user Bio, favorite sensor, location etc.

17. Group Chat:
    a. Description:
        i. Create a dynamic chat-box, to allow users post messages to be seen by every other user in the system.
    b. Acceptance Criteria:
        i. The chat should use CSRF token to provide proper encryption.
        ii. Users should get messages dynamically, and don't need to refresh the page.
        iii. Every user should be able to post the messages in chat
        iv. The chat should be real time
        v. New chat messages should be displayed directly, that is user doesn't need to scroll all the way down.
        vi. User can scroll past all the past messages.
        vii. While scrolling, user shouldn't be redirected to the latest chat message.

18. Token Validation:
    a. Description:
        i. Update the design, to add a token decorator
    b. Acceptance Criteria:

           i. checks if token was entered or not

19. Retrieving Data from Web API:
    a. Description:
        i. To retrieve the weather data, sensor data from web api published by the dark sky team.
    b. Acceptance Criteria:
        i. The weather widget should get access to weather api.
        ii. Retrieval of data, should happen upon set interval, as soon as the data from the sensor is published.
        iii. The data should get mined properly and stored in database.

# Revision History

| Revision | Date | Change Description |
|---|---|---|
| Test Cases for Authentication | 10/01/2017 | Manual Testing was done for Sprint 1. |
| Test Cases for Token based Authentication | 10/13/2017 | Manual Testing was done for Sprint 2. |
| Test case for Weather, Sensor Display & Manage Users and Sensors | 10/29/17 | Manual Testing was done for Sprint 3. |
| Test cases for Displaying Other user profiles, searching other users using keywords, Group Chat, Token Validation Retrieving data from web Api | 11/12/17 | Manual Testing was done for Sprint 4. |
| | | |