

Team JAB Deliverable 2

Testing Process

Although we will not be performing major system tests for our requirements of this project, we will be testing individual sub-systems within Sugar Labs. The main sub-system that we will be testing is the modeling system. Within the modeling system we will be testing the integrity of methods that help control notifications, friend system, and the journaling system. In each system we have chosen specific methods and objects that will need to be tested and validated.

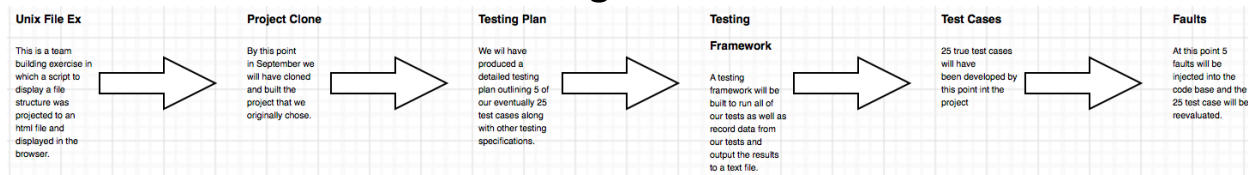
Requirements Traceability

Throughout the Sugar Labs project, there is very little documentation as to what certain classes and methods are doing. So, when we begin testing certain methods we will be creating some requirements of our own to satisfy this as we would otherwise not have any. If there is documentation as to what a method is doing or what requirement it need to satisfy, then we will develop our tests to mimic what the requirement demands. We will state in the test case what requirement we are testing.

Tested Items

The component folders that we are currently testing are the model and journal folders. In these folders we are specifically testing the notifications.py, model.py, and friends.py. We currently have plans to use more components from these folders in future tests.

Testing Schedule



Test Recording Procedures

All test results will export to an external .txt file. If a test passes, the txt file will simply record a "Pass" next to the test name/number. If a test fails, the txt file will display a "Fail" next to the test name/number as well as an error output if one is available.

Software Requirements

Currently, the only software requirements are being able to run python files as well as having a version of UNIX installed. This is because Sugar Labs creates a desktop environment which requires a UNIX environment. As we continue working on this project of software requirements will expand.

Constraints

We are all full time students with a full slate of classes. We will try to create a thorough test schedule and adhere to it. Although creating a strict testing schedule is very helpful, it is still possible to run into unforeseen issue. If any issues arise, we should be able to quickly resolve them.

Test Cases

Test Case 1 –

Title: Test Notifications Counter

Description: Ensures that the Counter for notifications works properly
Preconditions: All proper setup and config has been performed by user. The current notification count is at 0.

ID: test_notifications_counter

Requirement: Notifications received should be counted

Component: This tests the Notifications module

Method: This tests the Notify method.

Test Inputs: Notify("Journal", "23", "", "TestNotification", "", "", "", 30)

Expected Outcomes: Notify._notification_counter == 1

Test Case 2 –

Title: Test Remove Friends 1

Description: This test ensures that a User's Friends list stays correct after various modifications. **Preconditions:** All proper setup and config has been performed by user. A set of "Friends" object is instantiated by the driver found [here](#). The AddFriends Test has already been performed and verified

ID: test_remove_friends

Drivers: friend_driver.py

Requirement: Friends should be removable from the friends list

Component: This tests the Friends module.

Method: This tests the remove_friends method.

Test Inputs: make_friend(friend1), make_friend(friend2), remove_friend(friend1)

Expected Outcomes: friends{} = {friend 2}

Test Case 3 –

ID: test_make_friends_valid_buddy

Drivers: friend_driver.py

Requirement: Valid BuddyModel objects should be added to the Friends' _friends attribute.

Component: This tests the Friends module.

Method: This tests the make_friends method.

Test Inputs: BuddyModel("nickOne", "keyOne", "accountOne", "idOne")

Expected Outcomes: _friends[keyOne: FriendBuddyModel("nickOne", "keyOne", "accountOne", "idOne")]

Test Case 4 –

ID: test_make_friends_none

Requirement: None type objects should not be accepted as friends.

Component: This tests the Friends module.

Method: This tests the make_friends method.

Test Inputs: None

Expected Outcomes: TypeError

Test Case 5 –

ID: test_make_friends_string

Requirement: Strings should not be accepted as friends.

Component: This tests the Friends module.

Method: This tests the make_friends method.

Test Inputs: ['hello', 'goodbye', 'apple', '', 'five', 'testing inputs']

Expected Outcomes: AttributeError

Test Case 6 –

ID: test_make_friends_invalid_buddy

Requirement: BuddyModel objects lacking a valid key attribute should not be accepted as friends.

Component: This tests the Friends module.

Method: This tests the make_friends method.

Test Inputs: BuddyModel("nickOne", None, "accountOne", "idOne")

Expected Outcomes: AttributeError