

CSCI 362 Deliverable 2: SugarLabs

The Chocolate L'Eclercs

Alex Skiff, Blaine Billings, Carson Barber
Chase Myers, Justin Willis

1 Deliverable 2: Introductory Test Plan

1.1 Testing Process

We have divided the process of our test plan for our five introductory test cases into the evaluation of three main subsystems. They are described as follows:

1. **Profile:** The **profile** subsystem of SugarLabs includes all files relevant to the creation of a user's profile, including the age, gender, etc. This will be evaluated based on input/output matching as well as on exception handling for invalid input.
2. **Journal:** The **journal** subsystem of SugarLabs includes all files relevant for the upkeep of a user's journal. Much like with the first subsystem, this will be evaluated based on input/output matching as well as on exception handling for invalid input.
3. **Activity:** The **activity** subsystem of SugarLabs includes all files relevant to tracking a user profile's assigned activities, essentially providing running assessment list for the user. This will be evaluated based on correctness of logged information with regards to simulated performed actions.

1.2 Requirements Traceability

For the aforementioned subsystems, the following files will be evaluated. This information is included for the ease of tracking the testing process and the evaluated files as well as the understanding of what processes in specific are being investigated.

1. **Profile:** colorpicker.py, agepicker.py, genderpicker.py for color, age, and gender selection, respectively, in the user profile creation process.
2. **Journal:** journalactivity.py, journalentrybundle.py, journalwindow.py for activity logging, entry bundling, and graphical output, respectively, in the user journal editing and viewing.
3. **Activity:** buddy.py for the activity assignment in the user activity view.

1.3 Tested Items

In this section, we described the specific items mentioned in Section 1.2 and give an in-depth description of how they are to be tested through our framework. Figure 1 shows a visualization of three three general subsystems and their associated test cases.

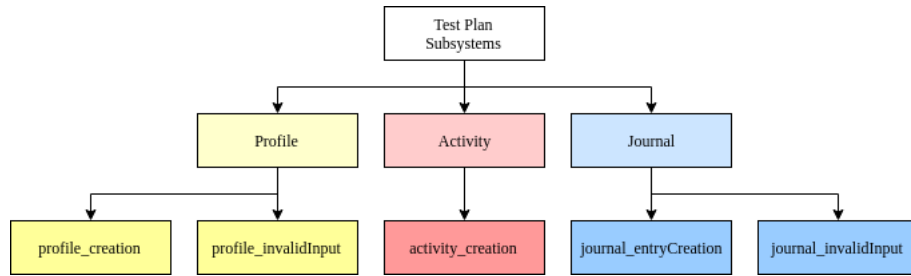


Figure 1: Overview of test plan subsystems and their associated test cases. (made using draw.io)

1.3.1 Profile

The testing of this subsystem and the respective colorpicker.py, agepicker.py, and genderpicker.py files will begin with the creation of a test_profile.py program with specific test_profile.creation and test_profile.invalidInput functions as test cases. These test cases go as follows:

1. **test_profile.creation:** This function will consist of the selection of a valid color and a gender and the input of a valid age. The test will be passed if profile creation is carried through with according input and without the throwing of an exception.
2. **test_profile.invalidInput:** This function will consist of the input of an invalid color, gender, or age. The test will be passed if profile creation with this input throws an exception.

1.3.2 Journal

The testing of this subsystem and the respective journalactivity.py, journalentry-bundle.py, and journalwindow.py files will begin with the creation of a test_journal.py program with specific test_journal.entryCreation and test_journal.invalidInput functions as test cases. These test cases go as follows:

1. **test_journal.entryCreation:** This function will consist of the opening of the journal entry window followed by the inputting of a valid journal entry name. The test will be passed if profile creation is carried through with according input and without the throwing of an exception.
2. **test_journal.invalidInput:** This function will consist of the opening of the journal entry window followed by the inputting of an invalid journal entry name (special characters, escaped characters, etc.). The test will be passed if profile creation with this input throws an exception.

1.3.3 Activity

The testing of this subsystem and the respective buddy.py file will begin with the creation of a test_activity.py program with specific test_activity.creation

function as a test case. These test case goes as follows:

1. **test_activity_creation:** This function will consist of the opening of the creation of a new activity and its assignment to a specific user profile. The test will be passed if activity creation is carried through with according input, assigned to the relevant profile, and without the throwing of an exception.

1.4 Testing Schedule

The tentative schedule for our testing plan has been defined as follows:

- 11/9/2018:** Completion of automated testing framework
- 11/19/2018:** Completion of design and implementation of testing framework; Evaluation of twenty-five test cases
- 11/28/2018:** Completion of code injection of 5 faults; Testing of chosen twenty-five test cases with faulty code.

1.5 Test Recording Procedures

Each of the implemented test cases will be coded in python and run through the terminal as described in the first deliverable. For easy auditing and understanding of previous tests, this code will include the date, time, and current software version with all output automatically being written to a log file for that specific date. These will be aggregated in a **testing** folder in the root directory of this testing framework's project.

1.6 Hardware and Software Requirements

As of the time of this document being written, Sugar Labs requires the local operating system to be either Debian 0.110 or higher, Fedora, or Ubuntu 16.04 or higher. In addition, Python 2.7, the main codebase of the software, must be installed alongside the sugar-artwork, sugar-datastore, and sugar-toolkit-gtk3 packages which can all be obtained directly from the terminal. For building the project, the packages intltool, libglib2.0-dev, and gtk+-3.0 should also be installed on the system.

1.7 Constraints

Though there are not many constraints limiting the testing process of this project, we must plan to be punctual in every step of our testing framework's implementation as required by the general outline of this course. Due to the nature of this project (in it being carried out in an educational setting), we need not worry about budgeting, staff, or weekly meeting presentations, all of which are often necessitated in government or industry settings.

1.8 System Tests

The system tests will primarily consist of checking relevant software installations. The following software, shown with their according verification commands, will be used to check that the system is up-to-date and able to run SugarLabs:

- **Python:** `python -v`
- **intltool:** `intltool --help`
- **autoconf:** `autconf -h`
- **libglib2.0-dev:** `dpkg --get-selections | grep libglib2.0-dev`
- **gtk+-3.0:** `dpkg --get-selections | grep gtk-3.0`