

CSCI 362 Deliverable 4: SugarLabs

The Chocolate L'Eclercs

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

1 Deliverable 4: Complete Testing Specifications

1.1 Documentation

As described in the previous section, the test cases have been implemented and saved into the directory structure as follows: each test case executable file has been saved in the `/testing-framework/testCaseExecutables/` folder; each test case description has been saved in the `/testing-framework/testCases/` folder; a short description as well as the inputs for the respective executable file have been documented in each of these text files.

1.2 Test Case Specifications

Below are detailed the specifications for all twenty-six test cases. Each brief description contains four main parts - the test case id number, the requirement being tested, the component being tested, and the method being testing.

1. **Activity Chooser:**
 - **Test ID:** 1
 - **Requirement:** Checks ActivityChooser titles and constructor
 - **Component:** `jarabe.model.activitychooser.py`
 - **Method:** `set_title`
2. **Age Calculator:**
 - **Test ID:** 2
 - **Requirement:** Checks for valid timestamp calculation of a birthdate
 - **Component:** `jarabe.intro.agepicker.py`
 - **Method:** `calculate_age`
3. **Age Calculator Invalid:**
 - **Test ID:** 3
 - **Requirement:** Checks for invalid timestamp calculation of a birthdate
 - **Component:** `jarabe.intro.agepicker.py`
 - **Method:** `calculate_age`
4. **Buddy Color:**
 - **Test ID:** 4
 - **Requirement:** Checks for color change ability
 - **Component:** `jarabe.model.buddy.py`
 - **Method:** `set_color`
5. **Buddy Key:**
 - **Test ID:** 5

- **Requirement:** Checks for key change ability
 - **Component:** jarabe.model.buddy.py
 - **Method:** set_key
6. **Buddy Nickname:**
 - **Test ID:** 6
 - **Requirement:** Checks for nickname changes
 - **Component:** jarabe.model.buddy.py
 - **Method:** set_nick
 7. **Change Brightness:**
 - **Test ID:** 7
 - **Requirement:** Can set brightness levels
 - **Component:** jarabe.model.brightness.py
 - **Method:** set_brightness
 8. **CMD Help:**
 - **Test ID:** 8
 - **Requirement:** Testing system command line
 - **Component:** jarabe.controlpanel.cmd.py
 - **Method:** cmd_help
 9. **CMD Restart:**
 - **Test ID:** 9
 - **Requirement:** Testing the system command line
 - **Component:** jarabe.controlpanel.cmd.py
 - **Method:** node_restart
 10. **Help Website:**
 - **Test ID:** 10
 - **Requirement:** Checks for the link to the help website
 - **Component:** jarabe.view.viewhelp.py
 - **Method:** get_social_help_server
 11. **HTTP Open:**
 - **Test ID:** 11
 - **Requirement:** Checks that http requests can be made
 - **Component:** jarabe.util.httprange.py
 - **Method:** open
 12. **HTTP Size:**
 - **Test ID:** 12
 - **Requirement:** Checks that http requests are valid
 - **Component:** jarabe.util.httprange.py
 - **Method:** size
 13. **Journal Comment:**
 - **Test ID:** 13
 - **Requirement:** Checks for journal comments
 - **Component:** jarabe.journal.expandedentry.py
 - **Method:** add_row
 14. **String isASCII:**
 - **Test ID:** 14
 - **Requirement:** Checks to make sure strings can be represented in

- ascii code
 - **Component:** jarabe.desktop.keydialog.py
 - **Method:** string_is_ascii
- 15. **String isHex:**
 - **Test ID:** 15
 - **Requirement:** Checks to make sure the hex string comparison is working
 - **Component:** jarabe.desktop.keydialog.py
 - **Method:** string_is_hex
- 16. **String Hex Converter:**
 - **Test ID:** 16
 - **Requirement:** Checks the string hex converter
 - **Component:** jarabe.desktop.keydialog.py
 - **Method:** string_to_hex
- 17. **Friends Model:**
 - **Test ID:** 17
 - **Requirement:** Checks for the friends model creation
 - **Component:** jarabe.model.friends.py
 - **Method:** get_model
- 18. **Add Friend:**
 - **Test ID:** 18
 - **Requirement:** Checks for errors when adding a friend
 - **Component:** jarabe.model.friends.py
 - **Method:** add_friend
- 19. **Has Friend:**
 - **Test ID:** 19
 - **Requirement:** Checks for added friends to appear as added
 - **Component:** jarabe.model.friends.py
 - **Method:** add_friend
- 20. **String Normalization:**
 - **Test ID:** 20
 - **Requirement:** Tests string normalization (removing non-ascii characters)
 - **Component:** jarabe.util.normalize
 - **Method:** normalize
- 21. **Serial Number Generation:**
 - **Test ID:** 21
 - **Requirement:** Checks for randomized serial generation
 - **Component:** jarabe.model.session
 - **Method:** have_systemd
- 22. **System Session Support:**
 - **Test ID:** 22
 - **Requirement:** Checks for school session logging support
 - **Component:** jarabe.model.session
 - **Method:** have_systemd
- 23. **Get System Session:**

- **Test ID:** 23
 - **Requirement:** Checks session logging
 - **Component:** jarabe.model.session
 - **Method:** get_session
24. **Set Volume:**
- **Test ID:** 24
 - **Requirement:** Changes the system volume
 - **Component:** jarabe.model.sound.py
 - **Method:** set_volume
25. **Mute Volume:**
- **Test ID:** 25
 - **Requirement:** Mutes the system volume
 - **Component:** jarabe.model.sound.py
 - **Method:** set_muted
26. **Speech Manager:**
- **Test ID:** 26
 - **Requirement:** Creates a speech manager utility
 - **Component:** jarabe.model.speed.py
 - **Method:** get_speech_manager

1.3 Team Evaluation

In this deliverable our team had to finish flushing out all 25 of the test cases that our Testing Framework would be automatically testing as well as make any changes to the framework that we found necessary. We decided that the way the framework was set up was perfectly fine for the rest of the project. From here we had a large task ahead of us, planning out and executing 20 more test cases on top of the 5 that we had for the previous deliverable. We got off to a decent start knocking a few out pretty quickly and started to hit a bit of a wall. Our biggest problem was many of the classes and files this system used are graphical and required additional imports that were not needed for a successful build. That issue combined with the system being written in python, it made it very difficult to understand some of the files because even if we could get them imported for test cases, we were not sure what all of the parameters meant because of python being a dynamically typed language. We still managed to get all 25 test cases and some of them were able to test graphical components of the system without actually needing the graphics to appear on screen. Sugar Labs would be a great tool and idea, but with the lack of documentation and instructions it turned out to be much larger of a task to come in and get test cases written than we initially expected. Overall we are happy with what we have accomplished up to this point with the system, it would have just been much easier to understand Sugar Labs as a whole if the development team had more documentation throughout the code.