YET ANOTHER LIST APP

Jake Allen, Evan Berryman, Sam Hobbs

Github: https://github.com/jake-allen/YALA
Trello issue tracking: https://trello.com/b/8JTgqSMS/yala

Test Documentation

Test Strategy: Use JUnit to test the functionality of basic operations throughout UserInterface and ItemSearch classes.

Test Data: Before every test, UserInterface ui is logged in with Jake's account (jake@jakeallen.com, password), and in almost every test a new list is added to the user's list storage (with names such as "TestList"). Any newly created lists are removed from the ListStorage, so every test can operate on a clean slate with the user's list storage. Also, for item-related use cases, a custom item "Nintendo Wii U Console" from Store "Amazon" with quantity 3 is used.

Test Cases:

- addAndDeleteList() – adds list "TestAdd" to the user's ListStorage; if the adding fails, or a separate search through the ListStorage reveals the absence of the list, the test fails. A sub-function deleteList() is called to test deleting this same list from ListStorage, failing if the list is still there. deleteList() is not a separate test because JUnit calls tests randomly, and deleteList() depends on an existing list

These functions both add a list to start and delete it to finish:

- copyList() – copies the list "TestList" as a new list "CopyList" – failing if the copy doesn't work, or the copied list is not in the ListStorage
- addItem() – adds an item to "TestList", failing if the added item is not present
- deleteItem() – adds and deletes item from "TestList", failing if the item is still in the list
- crossItem() – crosses off the newly added item, failing if the item's quantity is not negative (as any crossed item should have their quantity be)
- uncrossItem() – uncrosses the newly crossed-off item, failing if the item's quantity is negative (a non-crossed item always has positive quantity)

These last functions do not involve adding new lists:

- storeSelect() – initializes an ItemSearch object and has it switch to a table of the first store – while Swing is not normally visible to the user in JUnit, this test calls isTableVisible() on the ItemSearch, which reports if the table would otherwise be visible in a normally run program. The test fails if this returns false
- itemSearch() – the ItemSearch object's filters are set to only set items of type "Bread" & extra "wheat" to be the only items visible in the table; if "Bread Factory Bread wheat" is not the first item, the test fails

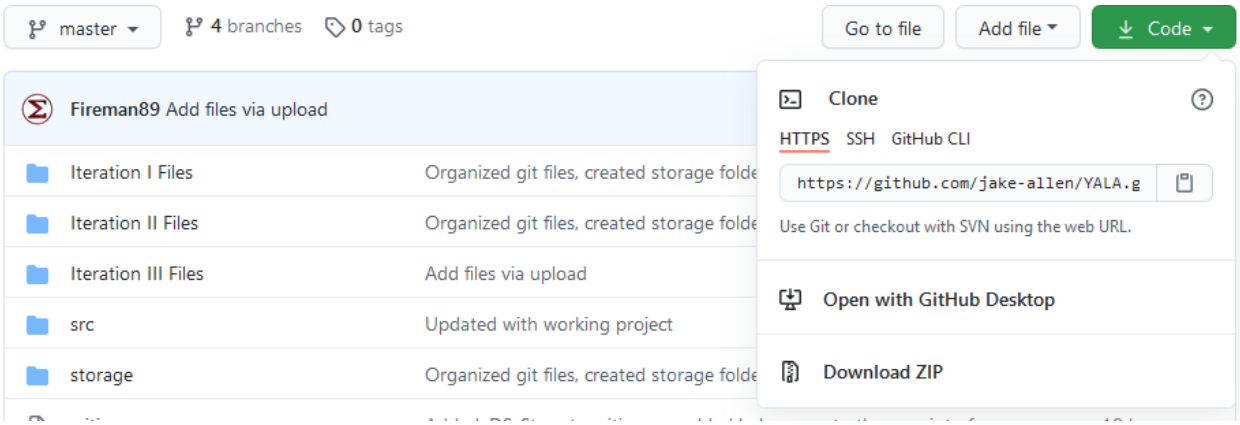
Traceability Matrix:

Requirements	Add List	Copy List	Delete List	Add Item	Select Store	Set Filter	Cross Item	Uncross Item	Delete Item
TC 1: addAndDeleteList()	✓		✓						
TC 2: copyList()	✓	✓	✓						
TC 3: addItem()	✓		✓	✓					
TC 4: deleteItem()	✓		✓	✓					✓
TC 5: crossItem()	✓		✓	✓			✓		
TC 6: uncrossItem()	✓		✓	✓				✓	
TC 7: storeSelect()					✓				
TC 8 itemSearch()						Incomplete			

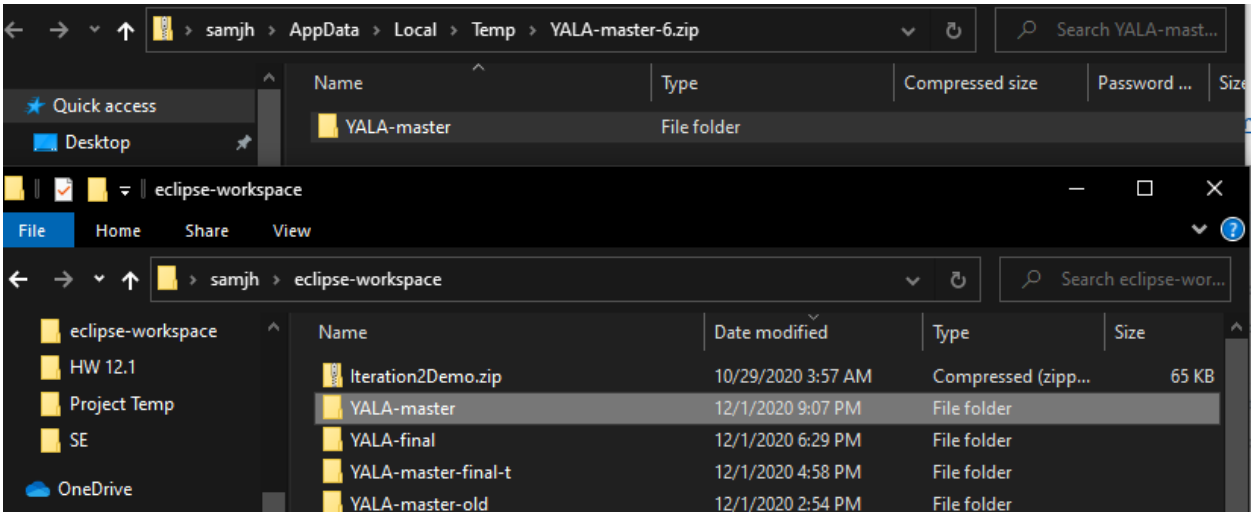
Installation Guide

Importing source code to Eclipse

1. Download the ZIP of the project at https://github.com/jake-allen/YALA.git.



2. Drag the folder to your Eclipse workspace.



3. Import YALA-master as an existing Maven project into Eclipse

4. This project will not run correctly unless it has the dependency JavaSE-1.8, and the JUnit test UserInterfaceTest needs JUnit 5. To add these, go into Project -> Properties -> Libraries and add the library “JUnit 5” and edit the library “Java SE 1.6” to instead be “Java SE 1.8.”

5. The program will now run as intended via UserInterface.java, and UserInterfaceTest.java provides a series of tests for the program.

Using runnable JAR

1. Download the ZIP of the project at https://github.com/jake-allen/YALA.git.

2. Move the YALA-master file to a location of your choice.

3. Click on the YALA.jar file in the YALA-master folder. It will not work if you move the jar outside of the folder as it depends on files inside of subfolders.

Issue Tracking and Commits

Git Commits

Total: 100 commits

Jake Allen: 49 commits

Sam Hobbs: 29 commits

Evan Berryman: 22 commits

Issues

General: 12 items

Jake Allen: 36 items

Sam Hobbs: 27 items

Evan Berryman: 25 items

Hours

Total: 129 hours

Jake Allen: 44 hours

Sam Hobbs: 43 hours

Evan Berryman: 42 hours