

Project 1 System Test Plan

Document Author(s): Jaden Abrams

Date: Oct 10, 2021

Introduction

This document contains the System Tests for Project 1, a product management system that tracks the development of various products through states of the tasks. The program can be launched through any Java IDE by running the file ProductBacklogGUI within the /src folder in the edu.ncsu.csc216.product_backlog.view.ui package. The tests within this document use text files in the test-files folder to load in products. The contents of the file we will be using for these tests, tasks1.txt, are shown below for easy reference.

```
# Shopping Cart Simulation
* 1,Backlog,Express Carts,F,jep,unowned,false
- [Backlog] Express carts always choose the shortest line. If there are
multiple shortest lines, an express cart chooses the one with the smallest
index.
* 2,Owned,Regular Carts,F,jep,sesmith5,false
- [Backlog] Regular carts always choose the shortest line excluding the express
register line (at index 0). If there are multiple shortest lines, a regular
cart chooses one with the smallest index.
- [Owned] Adding to sesmith5 backlog.
* 3,Processing,Java Swing,KA,sesmith5,sesmith5,false
- [Backlog] Learn more about Swing to debug GUI.
- [Owned] Adding to sesmith5 backlog.
- [Processing] Found Swing tutorials at
http://docs.oracle.com/javase/tutorial/uiswing/start/.
* 5,Verifying,Calculating Wait Time,B,jdyoung2,sesmith5,false
- [Backlog] Special carts are failing system tests associated with wait time.
- [Owned] Adding to sesmith5 backlog.
- [Processing] Replicated failure locally in unit test.
- [Verifying] Implementation complete. Requires peer inspection.
* 8,Done,Special Carts,F,jep,sesmith5,true
- [Backlog] Special carts always choose the shortest special register line.
If there are multiple shortest special register lines, a special cart
chooses one with the smallest index.
- [Owned] Adding to sesmith5 backlog.
- [Processing] Created hierarchy to prepare for other cart types.
- [Verifying] Implementation complete. Requires peer inspection.
- [Done] No problems found during inspection.
* 10,Rejected,Flatbed carts,F,jep,unowned,false
- [Backlog] Add flatbed carts to simulation.
- [Owned] Rejected. Flatbed carts won't fit through physical register stations.
# WolfScheduler
* 2,Rejected,Weekly Repeat,F,sesmith5,unowned,false
- [Backlog] Events should have a weekly repeat of every 1, 2, 3, or 4 weeks.
- [Owned] Weekly repeat is unnecessary when creating ideal week.
* 5,Backlog,Add Event,F,sesmith5,unowned,false
- [Backlog] Users can create events to identify places during their week where
```

they have scheduled activities other than class.

* 6,Done,Add Course,F,sesmith5,jctetter,true

- [Backlog] Users can add courses to their schedule.
- [Owned] Assigning to jctetter.
- [Processing] Creating Course class.
- [Processing] Adding error checking on course name.
- [Processing] Adding tests for Course.
- [Verifying] Request peer review.
- [Done] Updates meet requirements and test pass.

Test ID	Description	Expected Results	Actual Results
Test #1: Start Product Backlog	Preconditions: None 1. Run the program from an IDE, console, or executable.	Product Backlog starts up.	Product Backlog starts up with no active product.
Test #2: Load Products	Preconditions: Test 1 passes 1. Run the program 2. Navigate to File > Load 3. Navigate to tasks1.txt in the file system 4. open the file with the file explorer	The current product box populates with Shopping Cart Simulation and the lower box is filled in with tasks and their corresponding ID, state, type, and title.	The current product box populates with "Shopping Cart Simulation" with the lower grid filled in with the product's tasks and their corresponding IDs, states, types, and titles.
Test #3: Save Products	Preconditions: Test 1 and Test 2 pass 1. Navigate to file > save 2. put in the file to save the file to as "new_stp_tasks.txt" 3. click save	a new file "new_stp_tasks.txt" with the contents of tasks1.txt appears in the file explorer	a new file "new_stp_tasks.txt" with the contents of the GUI appears in the file explorer.
Test #4: Clear Backlog	Preconditions: Test 2 passes 1. Navigate to file > clear	The interface clears out, appearing as it did on start up.	The interface reverts to its appearance on start up.
Test #5: New Product	Preconditions: Test 1 passes 1. Click Add Product 2. Type "Vibe Check" in the Product Name? field. 3. click OK	The current product changes to Vibe Check.	The Current Product changes from nothing to Vibe Check

Document Revision History

Date	Author	Change Description
10/10/2021	Jaden Abrams	<ul style="list-style-type: none">• Created File• Wrote System tests