

# Project 1 System Test Plan

Document Author: Ethan Browne

Date: 9/23/2023

---

## Introduction

The System Test Plan provides test cases that run several different paths through the Project Backlog requirements. The main focus of these test is to check conflicts between different tasks.

To run the test:

1. Right click on BacklogGUI class in the Package Explorer.
2. Select Run As > Java Application

## Test Files

### ***tasks1.txt***

#### # 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.

## System Tests

Test ID	Description	Expected Results	Actual Results
Test 1: Valid File (UC 1)	Preconditions: None 1. Run BacklogGUI 2. Select File and then Load 3. Import test-files/tasks1.txt	The Current Product is: Shopping Cart Simulation  The tasks id for the product are:	<b>After following the steps as instructed the Task information displayed matched that of the Expected Results. The other</b>

	<ol style="list-style-type: none"> <li>4. Check Results</li> <li>5. Press the Drop Down Arrow next to current product</li> <li>6. Check Results</li> </ol>	<p>1,2,3,5,8, and 10</p> <p>The task states for the product are: Backlog, Owned, Processing, Verifying, Done, and Rejected</p> <p>The task types for the product are: Feature, Feature, Knowledge Acquisition, Bug, Feature, and Feature</p> <p>The task titles for the product are: Express Carts, Regular Carts, Java Swing, Calculating Wait Time, Special Carts, and Flatbed Carts</p> <p>In the drop down menu the other product, WolfScheduler, is display</p>	<p><b>product was also displayed in the drop down menu. This passes the test.</b></p>
Test 2: Add Task (UC 9)	<p>Preconditions: Test 1</p> <ol style="list-style-type: none"> <li>1. Click the Add task button</li> <li>2. Enter the following information in             <ol style="list-style-type: none"> <li>a. Product: Shopping Cart Simulation</li> <li>b. Title: Test Task</li> <li>c. Task Type: Feature</li> <li>d. Task Creator: Me</li> <li>e. Task Notes: Hello World!</li> </ol> </li> <li>3. Click Add Task to Backlog</li> <li>4. Check Results</li> </ol>	<p>The task should be added at the end of the task list with an ID of 11, a state of Backlog, a type of Feature, and a title of Test Task. All other tasks remain the same.</p>	<p><b>After following the steps as instructed a task was added to the bottom of my list with a correct task id, state, type, and title. This passes the test.</b></p>

Test 3: Remove Task (UC 10)	Preconditions: Test 1 1. Click on Task ID 10 2. Press the delete Task Button 3. Check Results	Task ID 10 has been removed from the product. All other tasks remain the same.	<b>After selecting task 10 and pressing the delete task button the task was deleted from the list. No other tasks were affected. This passes the test.</b>
Test 4: Edit Product (UC 5)	Preconditions: Test 1 1. Press the edit Product button 2. Enter the name "Amazing Product" 3. Check Results	The product is renamed from Shopping Cart Simulation to Amazing Product.	<b>After following the steps the product was renamed to "Amazing Product". This passes the test.</b>
Test 5: Select New Active Product (UC 7)	Preconditions: Test 1 1. Press the Drop Down Arrow next to current product 2. Click the WolfScheduler Product 3. Check Results	<p>The list of tasks should update to the new products tasks.</p> <p>The tasks id for the product are: 2,5, and 6</p> <p>The task states for the product are: Owned, Backlog, and Done</p> <p>The task types for the product are: Feature, Feature, and Feature</p> <p>The task titles for the product are: Weekly Repeat, Add event, and Add Course</p>	<b>After following the steps as instructed the WolfScheduler product was successfully displayed with all valid information. This passes the test.</b>