

# Project State Document

Dirty Water

---

University of Washington Tacoma  
Jeffrey Weiss  
TCSS 360

---

<https://dirtywater.github.io>  
[dirtywater365@gmail.com](mailto:dirtywater365@gmail.com)  
<https://discord.gg/GM3xgYg>

---

## **TEAM:**

Caleb Wheeler  
Michelle Brown  
Jim Phan  
David Guerrero

## Introduction

This document provides the details of the progress of our team's DIY application in the final stage of our first iteration. We successfully implemented the majority of the functionality for our top user stories, as well as a few requirements added after the initial design stage. We JUnit tested many of the core classes and ensured that the program is performing as intended. We had to resolve a few miscommunications in regards to coding, but overall communication was good and we pulled the project together with most of what we had hoped to accomplish.

## Table of Contents

<b>Title</b>	<b>pg.</b>
Introduction	2
Table of Contents	2
Requirements Addressed	3
How to run the Application	4
Contributions	5
Build and Tests	7
Source Catalog	9
Emergency Contacts	10

## Requirements Addressed

State of completion of the requirements collected and addressed over the iteration:

- |          |   |
|----------|---|
| US01.    | As a DIYer, I want to store data and measurements for my DIY projects so I can refer to them any time.<br><span style="color: green;">Complete.</span>  |
| US02.    | As a homeowner, I want to document my home's energy efficiency.<br><span style="color: red;">Attempted, but no where near complete.</span>  |
| US03.    | As a user, I want to make price calculations based on the data I assemble so I can estimate project costs.<br><span style="color: green;">Complete.</span>  |
| US07.    | As a user, I want to compare prices and technical information about products being considered.<br><span style="color: orange;">Somewhat complete. Products being offered are not real.</span>   |
| US11.    | As a user, I want to export my data in the form of a report so I can save it as a backup of my data or forward the report to contractors, energy audit professionals, retailers, or promotional programs.<br><span style="color: green;">Complete.</span>                                   |
| newUS01. | As a user I want to import settings to synchronize with other devices<br><span style="color: green;">Complete.</span>   |
| newUS02. | As a user I want to enter settings such as my first name and email address<br><span style="color: green;">Complete.</span>  |
| newUS03. | As a user I want to export settings for synchronization to other devices<br><span style="color: green;">Complete.</span>  |
| NF08.    | There is a consistent look among the different screens.<br><span style="color: orange;">Somewhat complete. Could refine appearance in later iteration.</span>   |
| NF09.    | On each screen, users have the choice to go to the previous screen and the main menu (among any other options you might provide).<br><span style="color: green;">Complete.</span>   |
| NF10.    | Users are unaware of the location of the files that hold the persistent data. No part of the interface should cause users to deal with specifying the location. Instead, the location of these files should be initialized by your program.<br><span style="color: green;">Complete.</span> |

# How to run the Application

## Set Up:

1. Find the folder marked "DirtyWaterApplication"
2. Ensure that the following are inside the folder: a folder named "images" which contains 9 images, DirtyWaterApp.jar, simstore.csv, testAbout.txt
3. If none of these are inside the folder, you can find them inside the source code. Please contact Jim if issues occur

Start the program.

## Creating a New Project:

1. Click the hammer icon to open the projects panel
2. Click "New Project"
3. Enter the project title
4. Save the project when finished

## Adding materials to a project:

1. After a project has been created or selected from the existing projects list
2. Click the cart icon to open the shop panel
3. Click to add the desired materials
4. Materials can be removed by going to the existing project and clicking materials

## Adding receipts to a project:

1. Select edit on the project from the list of existing projects
2. Click the receipt button
3. Click the Add receipt button
4. Enter data for the receipt and click save

## Changing User Settings:

1. Click the Gear icon
2. Change the name in the first textbox and the email with the second
3. After making changes click save

## Contributions

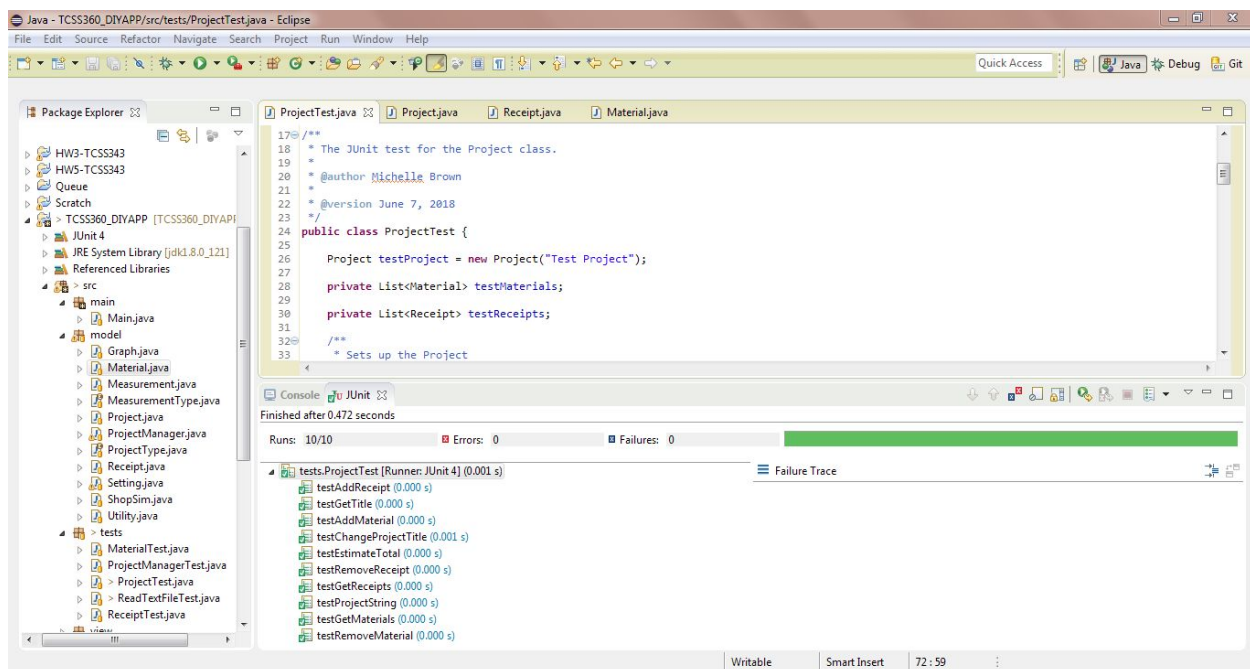
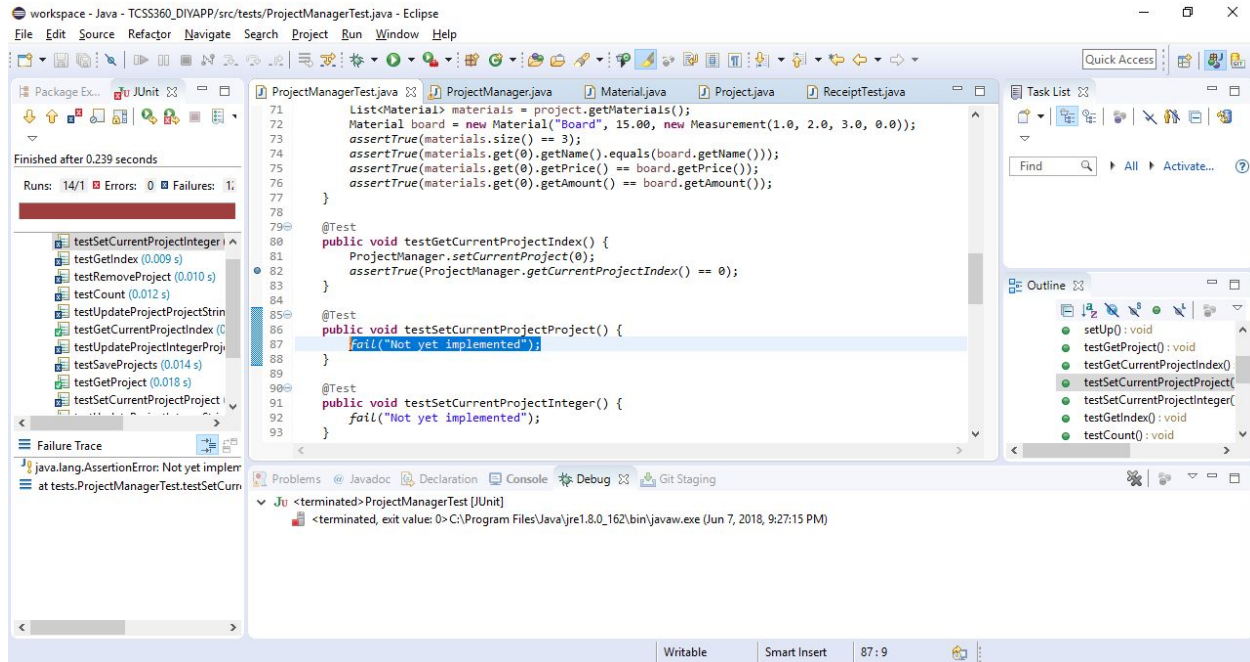
coding = authorship of all or an overwhelming majority of statements coded.

editing = bug fixes and refactoring.

Person	Class–Method	Activity
Jim	Main	Wrote class that starts the program
Jim	Graph	Wrote class that wasn't used
David	Material	Wrote class
Jim	Material - compareTo - MaterialComparator	Added comparator
Michelle	Measurement	Wrote class
Jim	Measurement	Refactored class as an outer class instead of inner
Michelle	MeasurementType	Wrote enum
Caleb	Project	Wrote class
David	Project - removeMaterial	Refactored to remove one
Caleb	ProjectManager	Wrote class
Caleb	ProjectType	Wrote enum
Michelle	Receipt	Wrote class
Jim	Receipt - getComparator - ReceiptComparator	Added comparator
Caleb	Receipt - Receipt	Edited second constructor
Caleb	Setting	Wrote class
David	ShopSim	Wrote class
Caleb	Utility	Wrote most of class
David	Utility - readTextFile	Wrote method that reads text file and returns contents as String
Jim	GUI	Wrote class
Jim	MainFrame	Wrote class
Michelle	MainFrame - createSidePanel - createButton - createPanel	Added images and paths Added icons Added home and about page

Caleb	MainFrame	Refactored changing panels and added panel enumerator
Caleb	MainFrame - PAGE	Refactored in page enum
Caleb	AboutPanel	Wrote class
Jim	DisplayPanel	Wrote class as a placeholder for code
Jim	GraphPanel	Wrote incomplete class that wasn't used
Michelle	HomePanel	Wrote class
Caleb	ProjectPanel	Wrote most of class
Michelle	ProjectPanel - AddReceiptsPanel - actionPerformed - openReceiptsPage	Wrote method Added piece of code to add a receipt Moved piece of code to method to be reused
David	ProjectPanel - ProjectAskPanel - ProjectMaterialsPanel	Refactored to get multiple of same material to show up as one
Jim	ReportPanel	Wrote class
Michelle	SettingsPanel	Wrote class
David	ShopPanel	Wrote most of class
Caleb	ShopPanel - ShopPanel	Refactored shop items buttons actionlistener
David	MaterialTest	Wrote test
Jim	ProjectManagerTest	Wrote test
Michelle	ProjectTest	Wrote test
Michelle	ReadTextFileTest	Wrote test
Jim	ReceiptTest	Wrote test
Jim	(all)	Fixed encapsulation issues
Michelle	(all)	Fixed missing Javadoc including @author credit and @version

# Build and Tests



## Test Logs

Report class log.

While testing the Report class, issues came out where the Receipt's title would appear as null. After checking the Receipt class, found that when initializing the Receipt class, it did not have its title updated. This resulted in a null value. As such, the values were fixed.

JLabel issue log.

When trying to display text using a JLabel, the text displays on the frame on a single line. However JLabels use HTML so we could display an HTML formatted string to insert new lines into a JLabel.



# Source Catalog

A catalog of the java files created in this iteration, including JUnit tests.

## **MAIN**

Main

## **MODEL**

Graph

Material

Measurement

MeasurementType

Project

ProjectManager

ProjectType

Receipt

Setting

ShopSim

Utility

## **VIEW**

GUI

MainFrame

## **PANEL**

AboutPanel

DisplayPanel

GraphPanel

HomePanel

ProjectPanel

ReportPanel

SettingsPanel

ShopPanel

## **TESTS**

MaterialTest

ProjectManagerTest

ProjectTest

ReadTextFileTest

ReceiptTest

\*source packages are in bold

## **Emergency Contacts**

Phone numbers of team members that will be available for up to one week after the iteration is complete in case of problems with the application.

Jim Phan                      (206) 370-4416

Caleb Wheeler              (360) 349-8532