Dunwoody College of Technology

**Test Plan for**

**Java Coffeemaker**

Confidential and Proprietary Information of Sarah Fox

Contents

1.0 Introduction 3

1.1. Objective 3

1.2. Project Description 3

1.3. Process Tailoring 3

1.4. Referenced Documents 3

2.0 Assumptions/Dependencies 3

3.0 Test Requirements 3

4.0 Test Tools 4

5.0 Resource Requirements 4

6.0 Test Schedule 5

7.0 Risks/Mitigation 5

8.0 Metrics 5

**[9.0 DEFINITIONS 5-6](#_Definitions_and_Acronyms)**

# Introduction

## Objective

This document describes the test plan for the CoffeeMaker application and includes information on what is to be tested, and how the testing is to be accomplished. Specifically, this document describes the tests to be performed, the testing schedule, resource required, dependencies, and test tools. This is a living test plan and must be changed to reflect team needs and requirements as they arise.

The Test Plan is an aggregation of information, which describes the entire test activity for this project. It covers the entire testing effort (unit, development test, system verification test, and Beta). It identifies the product requirements, schedules, resource requirements (people, effort and equipment), quality, assumptions, exclusions, and risks.

A preliminary Test Plan is prepared for the Project Team during the System Phase of PEAQ Process. This Test Plan will be updated in the earliest possible time of the Implementation Phase, so that progress can be tracked during implementation.

## Project Description

This product is an automated coffeemaker that allows the user to select a coffee beverage of their choosing using an interactive, digital interface. The user will be able to choose from up to three existing recipes and modify them to their specifications as well as create a new, custom recipe that they can save to the system to be ordered at a later date. The coffeemaker will accept payment and display the purchase status

## Process Tailoring

This project will use software development and management processes as a guideline. The following tests are planned:

* Module testing
* Unit testing
* User Acceptance Testing
* System Integration Testing

Since there is no plan yet to do the following tests, though they will be added at a later date as the physical implementation of this project makes progress:

* Functionality Testing
  + Graphical User Interface Testing to look for accessibility, responsiveness, and efficiency of the GUI that the user interacts with to order their beverage
  + Status and error messages will be tested
* User Experience Testing

## Referenced Documents

Referenced Documents

1. Software Requirements Specification
2. Software Development Plan
3. Structured Software Test Planning at DataCard, participant manual from   
    Benchmark Laboratories Incorporated, August 1996
4. UML Class Diagram
5. Sequence Diagram

# Assumptions/Dependencies

All assumptions for carrying out this test effort successfully are listed here. Some requirements assumptions might be necessary in order to scope the test activities. Also, assumption of responsibility to conduct unit, integration, SVT, regression, and beta tests.

# Test Requirements

It is assumed that there will be a working version of Java 15 or later installed as well as access to JUnit 4 or later which can be found in these two files:

1. hamcrest-core-1.3.jar
2. junit-4.13.2.jar

The code is expected to be completed by November 24th, 2021 in order to meet the test schedule. Access to the Gradle build tool will also work for running the application.

# Test Tools

1. Various hardware and software test tools apart from the deliverable product are used to assist in the testing process. These include:
   * defect control reporting and tracking software (PVCS Tracker)
   * computer hardware
   * access to the command line interface
   * access to JUnit 4
2. In future iterations of this project there will be machinery and hardware for mechanical engineers to test as well as User Acceptance Testing to be completed on the user interface, by the user, on the machine itself

# Resource Requirements

Based upon the test requirements identified in Section 3.0 and the tools development identified in Section 4.0, an estimate of resources required to accomplish the tests as performed are testers, users, and an environment by which to execute and run the Java code and JUnit tests. Refer back to Section 2.0 for more information.

# Test Schedule

Based upon the resource requirements identified in Section 5.0, a test schedule can be planned. The test schedule must be compatible with the project overall schedule. Coordination with the Project Manager and Development Lead Engineers is essential in planning a realistic and workable schedule.

|  |  |  |
| --- | --- | --- |
| **Test Sequence** | **Start** | **Finish** |
| 1. Unit test individual modules | 11/19 | 11/24 |
| 1. Test Development | 11/19 | 11/24 |
| 1. Module Availability | 11/19 | 11/24 |
| 1. User Acceptance Testing | TBD | TBD |
| 1. System Integration Testing | TBD | TBD |

Once the logic is finalized on the backend of the CoffeeMaker application, the user interface will need to be tested with UAT and SIT will be needed once the other parts of the coffee maker are assembled to ensure the various parts work together safely and accurately.

# Risks/Mitigation

[List all potential risks in this section. There still might be risks even in a good plan. These should be identified and a mitigation plan developed.]

# Metrics

The following metrics data will be collected. Some will be collected prior to, and some after product shipment.

**Prior to shipment:**

Effort expended during DVT, SVT and Regression

Number of defects uncovered during DVT, SVT and Regression, and development phase each defect is attributable to:

Test tracking S-Curve

PTR S-Curve

**After shipment:**

Number of defects uncovered and development phase each defect is attributable to:

Size of software

# Definitions and Acronyms

SVT System Verification Testing

SIT System Integration Testing

UAT User Acceptance Testing

DVT Design Verification Testing

PTR Problem Tracking and Resolution