

COFFEE MAKER TEST PLAN

EMAD ARMITI

Table of Contents

- INTRODUCTION3**
 - 1. OBJECTIVES3
- 2. SCOPE3**
- 3. ASSUMPTIONS / RISKS4**
 - 3.1. ASSUMPTIONS4
 - 3.2. RISKS4
- 4. TEST APPROACH.....4**
- 5. TEST TOOLS.....4**

Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

1. Objectives

The Coffee Maker is an app used to create drink recipes, it's written in Java programming language. It includes many functionalities such as adding the recipe ingredients and creating recipe books.

Phase 1 of the project will deliver TCR (Test Case Recipe) with functionality to create automated tests.

Team Members

| Resource Name | Role |
|---------------|------------------|
| Emad Armiti | Testing Engineer |

2. Scope

The initial phase will include all 'must have' requirements. These and any other requirements that get included must all be tested. At the end of Phase 1, a tester must be able to do these functionalities:

1. Retrieving recipe ingredients
2. Creating recipes
3. Setting recipe ingredients (coffee, milk, sugar)
4. Setting recipe info (price, name)
5. Creating recipe books
6. Adding recipes to recipe books
7. Changing recipe book content

3. Assumptions / Risks

3.1. Assumptions

This section lists assumptions that are made specific to this project.

1. The Coffee Maker app is for creating and editing recipes and combining recipes in recipe books
2. The interaction with the app is by using console

3.2. Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

| # | Risk | Impact | Trigger | Mitigation Plan |
|---|------|--------|---------|-----------------|
| 1 | | | | |

4. Test Approach

Automated testing using JUnit to validate and cover these functions:

1. Get functions (GetAmtChocolate, getAmtCoffee, getAmtMilk, getAmtSugar, getPrice, and getName)
2. Set functions (setName, SetAmtChocolate, setAmtMilk, setAmtSugar, and setPrice)
3. Recipe book functions (addRecipe, editRecipe, and deleteRecipe)
4. Other (equal)

5. Test Tools

1. Eclipse IDE
2. JUnit Lib