**McDonalds Meal Generator Test Plan**

Version 1.0

Authors:

Brandon Polite, Mike Pursh, Tiana Poteat, Enrich Rogers

Delaware Technical Community College

CSC214-401 Computer Science III

November 17th, 2020

**Version History**

|  |  |  |
| --- | --- | --- |
| Date | Version Number | Description |
| 11/17/2020 | 1.0 | ● Created the document and added core elements |

**Table of Contents**

[**1. Introduction**](#_heading=h.gjdgxs) **4**

[1.1. Purpose](#_heading=h.30j0zll) 4

[1.2. Project Overview](#_heading=h.1fob9te) 5

[1.3. Audience](#_heading=h.3znysh7) 5

[**2. Test Strategy**](#_heading=h.2et92p0) **6**

[2.1. Test Objectives](#_heading=h.tyjcwt) 6

[2.2. Test Assumptions](#_heading=h.3dy6vkm) 7

[2.2.1. Key Assumptions](#_heading=h.1t3h5sf) 7

[2.2.2. General Assumptions](#_heading=h.4d34og8) 7

[2.3. Test Principles](#_heading=h.2s8eyo1) 7

[2.4. Data Approach](#_heading=h.17dp8vu) 7

[2.5. Scope and Levels of Testing](#_heading=h.3rdcrjn) 8

[2.5.1. Exploratory](#_heading=h.26in1rg) 8

[2.5.1. Functional Test](#_heading=h.lnxbz9) 8

[2.5.2. User Acceptance Test???](#_heading=h.35nkun2) 9

[**3. Execution Strategy**](#_heading=h.1ksv4uv) **9**

[3.1. Test Cycles](#_heading=h.44sinio) 9

[3.2. Validation and Defect Management](#_heading=h.2jxsxqh) 10

[3.3. Test Metrics](#_heading=h.z337ya) 10

[3.4. Defect Tracking and Reporting](#_heading=h.3j2qqm3) 10

[**4. Test Management Process**](#_heading=h.1y810tw) **11**

[4.1. Test Management Tools](#_heading=h.2xcytpi) 11

[4.2. Test Design Process](#_heading=h.1ci93xb) 11

[4.3. Test Execution Process](#_heading=h.3whwml4) 12

[4.4. Communications Plan and Team Roster](#_heading=h.2bn6wsx) 12

[4.4.1 Role Expectations](#_heading=h.qsh70q) 12

[4.4.2. Main Developer](#_heading=h.3as4poj) 12

[4.4.3. Testing Lead](#_heading=h.1pxezwc) 13

[4.4.4. Support](#_heading=h.2p2csry) 13

[**5. Test Environment**](#_heading=h.49x2ik5) **13**

# 1. Introduction

## 1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the McDonalds Meal Generator. The document introduces:

* Test Strategy: Descriptions of parameters and data to be given to the program to test its various functions.
* Execution Strategy: describes how the test will be performed and the process to identify and report defects, and to fix and implement fixes
* Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

## 1.2. Project Overview

This software will be able to allow a user to login and generate random meals that they can order at a McDonalds location. The program will also allow the user to view nearby McDonald's locations as well as define specific parameters to narrow the scope of their meal generation. The software will also allow the user to favorite or dislike meals to either view them again in their favorites list, or exclude them from future meal generation.

## 1.3. Audience

* Project team members perform tasks specified in this document, and provide input and recommendations on this document.
* The instructor of CSC214 Computer Science III will assess the document and give feedback on the competency of the outlined testing strategy for the software.
* Students and users of the program can assess the testing strategy and give feedback on aspects of the program that may need further testing. They can also read this document to gain further understanding of the inputs and functions of the program

# 2. Test Strategy

## 2.1. Test Objectives

* Verify complete functionality of program requirements outlined in the McDonalds Meal Generator SRS document.
* Reveal any defects that need to be addressed before final deployment of the software
* Deliver a bug free software.

## 2.2. Test Assumptions

### 2.2.1. Key Assumptions

* Tester is on an operating system that supports Java SDK 14
* Keyboard and mouse will be used to interact with the program

### 2.2.2. General Assumptions

* Performance testing is beyond the scope of this project
* User understands general conventions of software use
* Project development team adequately defined test cases to encompass all possibilities for a given field

## 2.3. Test Principles

* Procedures, inputs and results will be repeatable and measurable
* Success or failure to respond properly to an input must be properly defined
* Testing results must be consistent
* All testers will have similar/identical test environments

## 2.4. Data Approach

* While testing functionality, all data fields and functions will be tested via human interaction and direct usage, developers must refer to test case tables to ensure all expected inputs are tested
* The underlying data processing functions will be tested autonomously via program functions

## 2.5. Scope and Levels of Testing

McDonalds Meal Generator will be a functional utility to assist the user in deciding upon the meal they would like to eat from McDonalds, thus functional testing will be needed to ensure the program is stable and responds accordingly to unexpected inputs.

### 2.5.1. Exploratory

**Purpose:** ​The purpose of this test is to make sure the software is in a runnable state before the next level of testing can start

**Scope:** ​Login Prompt

**Testers:** ​All four group members

**Method:** ​This exploratory testing is carried out in the application without any test scripts and documentation.

**Timing:** ​At the beginning of each cycle.

### 2.5.1. Functional Test

**Purpose:** ​Ensure functionality of each portion of the program. Testing will be carried out by entering various inputs and evaluating whether the expected output is given. Testers will refer to the test case tables contained within the google sheets document “McDonalds Meal Generator Test Cases”. [Link to google sheets document (view only).](https://docs.google.com/spreadsheets/d/1x0HI3nCSNhjLBIsO9uHw9xVRcKy2e4vNH81SC2OCwSc/edit?usp=sharing)

**Scope:** ​

* Login Menu
* Meal Generator Menu
* Meal Generator Results
* Favorites Page
* Locations Page
* System navigation

**Testers:** ​Testing cases will be distributed among the four developers to carry out.

**Method:** ​Follow test case tables provided in “McDonalds Meal Generator Test Cases.sheets” link above

**Timing:** ​After the exploratory test is completed.

### 2.5.2. User Acceptance Test

**Purpose:** ​Use the program and analyze if the program that were to be developed was in fact delivered, all functions and tools accounted for.

**Scope:** Entire Program

**Testers:** ​All Developers

**Method:** Perform qualitative analysis on program functionality while using the program thoroughly

# 3. Execution Strategy

* Testing execution will be divided into cycles for organization and to ensure fixes are properly implemented
* Testing execution will not be complete until all test cases pass with a desirable output.

## 3.1. Test Cycles

* There will be two distinct cycles for functional testing. Both cycles will test both automated inputs and functionality as well as user inputs and functionality.
* Cycle 1, Failure Analysis: The purpose of the first test cycle is to find the bugs and issues with the software and analyze what caused them.
* Cycle 2, Fix Analysis: The purpose of the second cycle is to test whether the fixes implemented for the failures identified in cycle one were adequate. If not, new fixes will need to be implemented and cycle 2 must be repeated

## 3.2. Validation and Defect Management

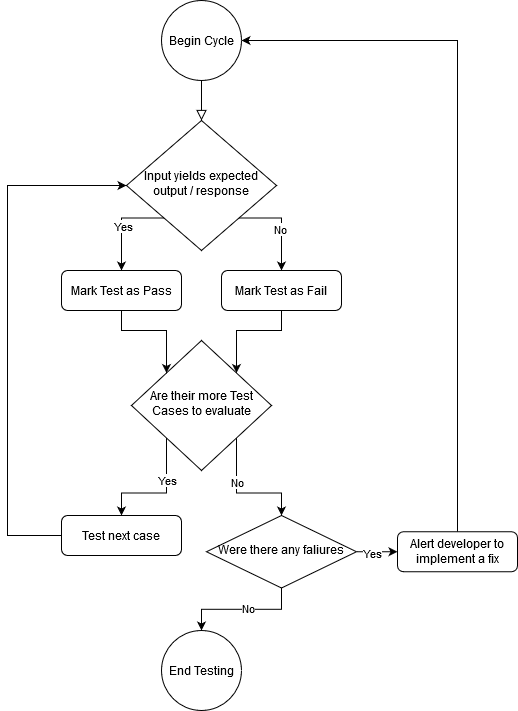
* Validation of functionality will be determined by whether the software responds as it is intended to with various inputs.
* The software’s responses will be tracked in “McDonalds Meal Generator Test Case.sheets” link in section 2.5.1.
* It is the responsibility of the tester to adequately communicate discovered defects.
* It is the responsibility of the developer who implemented the functionality to design a fix for the defect

## 3.3. Test Metrics

* Tests will receive a pass/fail status depending on systems response to the inputs.
* After testing cycles are completed, all test cases must have a passed status for all possible inputs.

## 3.4. Defect Tracking and Reporting

The following flowchart depicts the Defect Tracking Process:



# 4. Test Management Process

## 4.1. Test Management Tools

* Google Sheets testing case document “McDonalds Meal Generator Test Cases”.
* Developer communication group chat.
* Each Developer must be responsible for updating and accessing their test cases in the test cases document.
* Each Developer must be responsible for communicating software failures to the developer group chat.

## 4.2. Test Design Process

* Test cases will be defined based on the system requirements outlined in the McDonalds Meal Generator SRS document.
* All developers must agree that the test case table is adequate for testing the entirety of the software function.

## 4.3. Test Execution Process

* Ensure Software compiles and displays a login menu
* Each tester will be assigned roughly the same number of test cases
* Each tester must complete assigned test case tables and identify any software failures, inputs will be completed via keyboard and mouse
* Tester must report failure to appropriate developer to implement a fix
* After fixes are implemented, complete test case tables once more to ensure fixes were adequate.
* Test case results must be confirmed and approved by all developers before they are marked as final.

## 4.4. Communications Plan and Team Roster

### 4.4.1 Role Expectations

* All four group members will participate in the testing and development process.
* All Members are both developers and testers

### 4.4.2. Developer

* Responsible for developing a specific function.
* If a failure is found within a particular developer’s implementation of a function, it is that developers responsibility to design a solution to resolve the failure in future use.

### 4.4.3. Tester

* Responsible for completing a number of test case tables contained within “McDonalds Meal Generator Test Cases.sheets” (Link in section 2.5.1)
* Responsible for reporting discovered failures to appropriate developer to obtain a solution.

# 5. Test Environment

McDonalds Meal Generator is being developed in Java JDK 14. All testers must use an IDE supporting said JDK revision. As java is universally supported on many operating systems, the operating system of each user is irrelevant.