CS691 – Computer Science, Fall 2020

Pace University



SYSTEM TEST PLAN

Unit-Fit

Authors: Team 2

Project Manager: Pratesh Gartaula

Date: 11/10/2020

**Table of Contents**

[INTRODUCTION 3](#_heading=h.30j0zll)

[TESTING SCOPE 3](#_heading=h.3znysh7)

[TESTING OBJECTIVES 3](#_heading=h.tyjcwt)

[Features to be Tested 3](#_heading=h.1t3h5sf)

[Features not to be Tested 4](#_heading=h.2s8eyo1)

[TEST PROCESS DEFINITION 4](#_heading=h.17dp8vu)

[Test Process Phases and Tasks 4](#_heading=h.26in1rg)

[Deliverables 5](#_heading=h.35nkun2)

[APPROACH TO SYSTEM TESTING 5](#_heading=h.1ksv4uv)

[Approach to Functional Testing 5](#_heading=h.2jxsxqh)

[ENTRY/EXIT CRITERIA 6](#_heading=h.3j2qqm3)

[Entry Criteria 6](#_heading=h.1y810tw)

[Exit Criteria 6](#_heading=h.4i7ojhp)

[ENVIRONMENTAL NEEDS 6](#_heading=h.1ci93xb)

[ROLES AND RESPONSIBILITIES 7](#_heading=h.3whwml4)

[TEST CYCLES AND SCHEDULE 8](#_heading=h.qsh70q)

[RISKS AND CONTINGENCIES 8](#_heading=h.1pxezwc)

# INTRODUCTION

This document describes the System Test Plan that provides a common understanding among the “Unit Fit” project stakeholders on the scope, objectives, and approach to performing the system testing. Also, the document explains the features to be tested, testing entry/exit criteria, resource and responsibilities, and testing schedule.

# TESTING SCOPE

The testing scope includes two perspectives - the functional scope and technical scope.

In line with our expected functions of our system, the scope of system testing includes following functional aspects:

* Build Profile
* User’s account
* Searching
* Listing a workout videos

The corresponding modules in the Unit Fit system is Home page(create an account and sign in to account)

The technical scope includes the following components that are associated with the expected functions:

* Web browser
* Server
* Database

# TESTING OBJECTIVES

The primary focus of this System Test Plan is functional testing with the objective to evaluate the system implementation stability. The non-functional testing requires some special tooling to monitor performance characteristics, which is not available on this project.

The basis for developing functional tests and evaluating the system functionality includes the following sources:

* Business Requirements Document (BRD)
* User Stories (functional requirements)
* Requirements Composition Table (supplementary requirements)

## Features to be Tested

This section lists all core features that will be tested grouped by the application modules below.

**01.01 Create an Account**

* To test whether a user can create an account
* To test whether the data entry fields are validated
* To test whether it addresses the core feature behaviour when it generate data used by other core feature of the same application

**01.02 Sign-in into account**

* To test whether a user can sign in to an existing account
* To test whether the data entry fields are validated
* To test whether it populates default values based on another field value
* To test whether whether data coming from external applications and indicates which of the core features consume such data
* To test whether it addresses the core features behaviour when it takes data in from another core feature of the same application

**02.01 Select a Goa**l

To test whether after creating an account the users would have a verify of options to choose from as their goal. (Example: Weight loss, Gain muscle etc)

**02.02 Browse the type of workout**

To test whether the users select the type of workout video that they would be more interested in. (Example: Zumba, kick boxing etc)

**02.03 Level of Intensity**

To Test whether a user select the level of intensity for their workout based on their goal and the type of workout they have selected from.

**03.01 Diet Planner**

To test whether the app allows a user to plan out their daily meal schedule and based on the user's goal it would provide the min and max calorie intake.

**03.02 Recipes**

To test whether a user with a range of simple recipes to choose from and a few cooking videos from youtube.

**04.01 Workout statistics**

To test whether users can keep track of their daily workout and provide them with their workout history and hours spent.

**05.01 Freemium**

To test whether user can only access a set of basic features such as pre-provided workout type and intensity.

**05.02 Paid Subscription**

To test whether subscribed users can access all the premium features.

**06.01 Ads and Promo's**

To test whether a user can access any promotions/offers that is provided by Unitfit.

**07.01 Modify General User's Information**

To test whether a user can modify existing user's information.

**07.02 Modify Goal**

To test whether a user can change the existing goal.

**07.03 Add a New Card**

To test whether a user can add a new card to a user's account, so that a user can select that card while he is going through the payment process.

**07.04 Contact a Customer Support**

To test whether a user can email or call a customer support team.

**07.05 Reset the password**

To test whether a user to reset the password or change the existing password

**07.06 Delete an Account**

To test whether a feature a user can delete an account.

Besides the core features in the scope of testing, the function testing also will cover crosscutting concerns that are applicable to the context of the individual core features (refer to the RCT).

## Features not to be Tested

As mentioned above, system performance will not be tested for the lack of required tools. Also, usability and security will not be tested as well.

# 

# 

# 

# 

# TEST PROCESS DEFINITION

## Test Process Phases and Tasks

The test process consists of five phases, which include test planning, design, preparation, execution, and reporting. Each phase has a few tasks as defined below:

* Test Planning
  + Define scope and objectives of testing
  + Define roles and responsibilities
  + Define testing approach
* Test Design
  + Identify test ideas, define an approach to designing test cases
  + Develop test case specifications
  + Measure test coverage
  + Determine requirements for test data
* Test Preparation
  + Setup a test environment
  + Provision test data
  + Install the software in the test environment
* Test Execution
  + Execute all test cases
  + Find and report software defects
  + Evaluate the system stability
  + Validate all target features
* Test Reporting
  + Summarize and report the test execution results
  + Report defect metrics
  + Evaluate the test exit criteria
  + Create a test completion report, submit for stakeholder approval
  + Obtain stakeholder signoff on system testing

## Deliverables

On this project, the test process deliverables include:

* System Test Plan document
* Test Design specifications
* Test Case specifications
* Software Defects
* Test Execution Logs
* Test Completion Report

# APPROACH TO SYSTEM TESTING

## Approach to Functional Testing

The overall approach to functional testing will be based on the Black-box method:

* Test cases will be designed using some formal black-box techniques such as boundary-value analysis, equivalent-class partitioning, cause-effect graphing, decision tables, and state-transition testing, where applicable.
* Test execution will be conducted manually, from the user perspective and based on formal test case specifications.

The test execution results will be captured and reported in test execution logs.

entry /exit criteria en

# ENTRY/EXIT CRITERIA

This section defines both Entry and Exit Criteria for test execution and is intended to establish a common understanding about the conditions when the test execution can start and when it can stop.

## Entry Criteria

The test Entry Criteria include the following items:

* The application build is produced and deployed to the test environment
* The system test plan is produced and approved
* The test environment is ready for testing
* Test Designs and test case specifications are completed

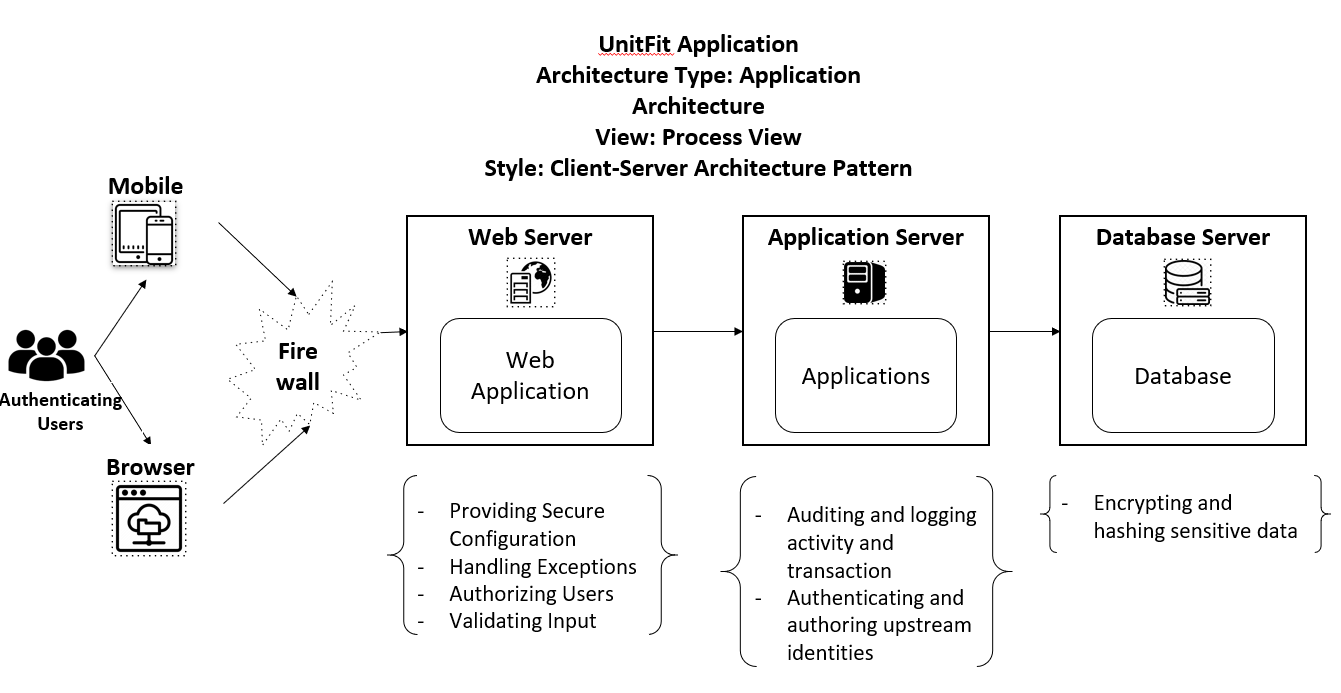
## Exit Criteria

The test Exit Criteria include the following items:

* All test cases have been executed
* Zero defects of Critical and High-severity remain open
* Open defects of Medium and Low severity have known work-arounds
* Test Summary report is produced and published

# ENVIRONMENTAL NEEDS

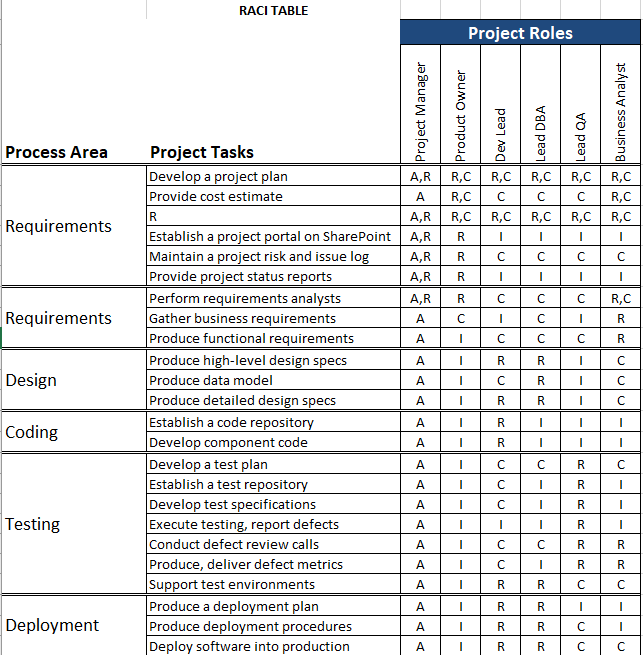
The Test Environment is available to start test execution which includes a laptop, localhost server, and Chrome browsers. The architecture of the test environment is shown below:



# ROLES AND RESPONSIBILITIES

The project team has five members that are assigned various project roles including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA. Their responsibilities are defined in the table below.

|  |  |
| --- | --- |
| **Project Role** | **Role Responsibilities** |
| Project Manager | Reviewing and approving the System Test Plan, test design specifications.  Managing the test environment preparation.  Tracking the testing schedule and results. |
| Lead QA Analyst | Designing a test plan, establishing a test repository, developing test case specifications, executing testing and reporting defects. |
| Product Owner | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Developer | Establishing and maintaining the test environment, assisting a Lead QA Analyst throughout the testing process. |
| DBA | Assisting the Lead Developer in establishing and maintaining the test environment. |



# TEST CYCLES AND SCHEDULE

The system test execution will be conducted as three test cycles that are aligned with three application modules as follows:

Cycle 1. User Experience I

* This cycle concentrates on testing the first part (input and filtering) of the User Experience Module

Cycle 2. User Experience II

* This cycle concentrates on testing the second part (generating and viewing recipes) of the User Experience Module.

Cycle 3. Payment and User Register/Login

* This cycle concentrates on testing the Payment and User Register/Login Module.

See the schedule of the test execution cycles in the project plan.

# RISKS AND CONTINGENCIES

This section highlights a few potential risks and contingencies that may happen during the system testing.

* Limited testing resources may result in a delay.
* Any changes on the scope objectives can cause a delay or extra work.
* A large number of defects require a longer time to fix the system
* Collaboration of the Team has an impact on the testing progress.