<HealthApp>

SYSTEM TEST PLAN

HealthApp

Author: Nishith Desai

Project Manager: Tarun Dagar

Date: December 01, 2022

Table of Contents

[INTRODUCTION 3](#_Toc485964763)

[1. TESTING SCOPE 3](#_Toc485964764)

[2. TESTING OBJECTIVES 3](#_Toc485964765)

[2.1 Core Features to be Tested 4](#_Toc485964766)

[2.2 Non-Functional Features to be Tested 4](#_Toc485964767)

[2.3 Features not to be Tested 5](#_Toc485964768)

[3. TEST PROCESS DEFINITION 5](#_Toc485964769)

[3.1 Test Process Phases 5](#_Toc485964770)

[3.2 Testing Tasks and Deliverables 6](#_Toc485964771)

[4. APPROACH TO SYSTEM TESTING 7](#_Toc485964772)

[4.1 Approach to Functional Testing 7](#_Toc485964773)

[4.2 Approach to Non-Functional Testing 7](#_Toc485964774)

[5. ENTRY/EXIT CRITERIA 7](#_Toc485964775)

[6. SYSTEM TEST ENVIRONMENT 8](#_Toc485964776)

[7. ROLES AND RESPONSIBILITIES 8](#_Toc485964777)

[8. TEST CYCLES AND SCHEDULE 9](#_Toc485964778)

[9. RISKS AND CONTINGENCIES 9](#_Toc485964779)

# INTRODUCTION

The primary purpose of the System Test Plan document is to establish a common understanding among the "HealthApp" project stakeholders about the scope, objectives, and approach to performing the system testing. In addition, the document covers such topics as environmental needs, testing entry/exit criteria, test schedule, roles and responsibilities, and risks and contingencies.

# 1. TESTING SCOPE

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

The functional scope includes the following modules:

* Account Management
* Fitness Tracking​
* Nutrition Tracking
* Coach Connect​
* Dashboard.

The technical scope includes the following architectural components:

* Web server
* Application server
* Database server
* Content server

# 2. TESTING OBJECTIVES

The main objective of system testing is to validate the implementation of the system features for compliance with their functional and non-functional requirements. The system test cases should include negative, i.e., challenging testing conditions in order to be effective in finding software defects. This section describes the features to be tested and the features that will be out of testing scope.

The list of project documents that will be used as a basis for designing test cases includes:

* Business Requirements Document (BRD)
* "HealthApp" Functional Requirements
* Supplementary Requirements Specification
* Requirements Composition Table (RCT)
* Database design and data dictionary specifications

## 2.1 Core Features to be Tested

Account Management

* Account Authentication
* Account Summary
* Customer Support

Fitness Tracking

* List Fitness Routine
* Track Steps
* Track Calories

Nutrition Tracking

* List food nutrition
* Track food intake
* Modify food intake
* Remove food intake

Coach Connect​

* List Of Coaches
* Subscribe Coach
* Schedule Appointment
* Access Routine

Dashboard

* Display fitness activity
* Display diet analysis
* List user goals

In addition to the above core features, testing will cover crosscutting concerns applicable to the context of individual core features, see the Requirements Composition Table for reference.

2.2 Non-Functional Features to be Tested

The non-functional testing requires some special tooling to monitor performance characteristics, which is not available on this project.

2.3 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.

# 3. TEST PROCESS DEFINITION

## 3.1 Test Process Phases

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

## 3.2 Testing Tasks and Deliverables

Each phase in the test process is further defined in terms of tasks and deliverables as shown in the table below.

|  |  |  |
| --- | --- | --- |
| **Process Phase** | **Tasks** | **Deliverables** |
| Test Planning | * Define the scope, objectives, and approach to system testing | System Test Plan document |
| Test Design | * Detail the approach to system testing * Specify required test data * Design test-case specifications * Setup a test management system | * Test Design Specification * Test-Case Specifications * Test Management System HPQC |
| Test Preparation | * Setup the test environment * Migrate the system into the test environment * Provision test data * Setup a defect tracking system | * The system under test is up and running in the test environment * Test data available in the QA environment * Defect Tracking System is ready for the test cycle |
| Test Execution | * Test the system and find and report defects | * Defect reports reported in the defect tracking system * The system has been completely tested * Test Summary Report produced and approved |
| Test Reporting | * Produce defect metrics * Report test execution progress * Produce a test completion report | * Test Summary Report * Defect metrics * Test execution status reports |

# 

# 4. APPROACH TO SYSTEM TESTING

## 4.1 Approach to Functional Testing

The System Test will be performed based on the black-box techniques. This means, first, that the external functional specifications or business rules will be used as a primary source to design test conditions. Secondly, testing will be executed from the user perspective, i.e., considering the system as a black box and entering input data and evaluating results via the user interface.

The system features identified above can be classified by the following types of business logic – GUI, Field Edits, Field Dependencies, and General Business Rules. Each type can have its own test logic that can be reused across the system. Test conditions can be designed using conventional techniques, such as boundary analysis, equivalence partitioning, decision tables, etc. The detailed test logic for each pattern of business rules will be described in the test design specification.

## 4.2 Approach to Non-Functional Testing

All non-functional test objectives specified above can be tested using the black-box approach, i.e. from the user perspective. The volume test should be performed for a complete production scenario that covers the allocation steps, sending emails, and uploading and committing results. The portability and extreme layout tests should cover all functions (menu options) of the system and validate that each function works under the specified test conditions.

# 5. ENTRY/EXIT CRITERIA

The **Test Entry criteria** is used to formally evaluate the conditions necessary to begin test execution, it includes the following conditions:

* Development tasks and integration testing have been completed
* System Test Plan document has been approved
* QA environment is ready
* QA team members have access to the QA environment
* Test case specifications have been completed and reviewed
* Release Notes document has been sent to the QA team

The **Test Exit criteria** is used to evaluate the conditions necessary to conclude that testers can stop test execution and the system is ready for the final user acceptance testing, it includes the following conditions:

* All requirements, in scope of testing, are covered by test cases
* All test cases have been executed
* Zero defects of Critical and Hi-severity remain open
* Open defects of Medium and Low severity have known work-around
* A Test Completion Report has been produced and communicated to stakeholders
* QA testing sign-off has been provided

# 6. SYSTEM TEST ENVIRONMENT

The system test will be performed on mobile devices on the following OS:

1. Android

2. iOS

The backend server is deployed on the AWS EC2 connected to RDS

(US-EAST-1).

# 7. ROLES AND RESPONSIBILITIES

The project roles involved in system testing include the following:

|  |  |
| --- | --- |
| **Project Role** | **Role Responsibilities** |
| Project Manager | Responsible for the overall project timelines, review and approval of the System Test Plan, escalation of issues. |
| QA Environment Manager | Responsible for procurement and support of the QA environment. |
| Developers | Responsible for producing a working software build, build migration to the QA environment, communicating release notes, investigating and fixing software defects. |
| Test Manager | Responsible for developing a System Test Plan document, planning the testing tasks, maintaining the test repository in HPQC tool, coordinating test execution, producing a Test Completion Report. |
| QA Lead | Responsible for developing test cases, overseeing test execution, conducting defect review calls, providing test execution metrics and reports. |
| Testers | Responsible for developing and executing test cases, reporting defects and re-testing defect fixes. |

# 8. TEST CYCLES AND SCHEDULE

System testing will be executed in three cycles:

* Cycle 1 - Focuses on testing the Account Management Module;
* Cycle 2 - Focuses on testing the Fitness Tracking Module;
* Cycle 3 - This cycle concentrates on testing the Nutrition Tracking Module;
* Cycle 4 - This cycle concentrates on testing Coach Connect Module;
* Cycle 5 - This cycle concentrates on testing Dashboard Module;

See the timelines of the testing cycles in the project plan.

# 9. RISKS AND CONTINGENCIES

This section highlights a few potential risks and contingencies that maybe happened during the system testing.

* A lack of testing resources can result in more time needed to complete test case specifications.
* Changes to the implementation scope or existing functional requirements can impact the test execution schedule.
* Too many defects can delay the completion of test execution.
* Instability of the test environment can impact the test execution schedule.