**Testing plan: Anxiety app.**

**Prepared by:**

Nigel brennan.

Festus Ikenna.

**Table of contents:**

1. introduction
2. objectives and tasks
   1. objectives
   2. tasks
3. scope
4. testing strategy
   1. unit testing

-code coverage testing.

* 1. Integration Testing
  2. System Testing
  3. User Acceptance Testing
  4. Performance Testing

1. Hardware requirements
2. Test schedule
3. Control procedures
4. Features to be tested
5. Features not to be tested
6. Resources/roles and responsibilities
7. Schedules
8. Dependencies
9. Risks/assumptions
10. Tools
11. Approvals
12. introduction:
    1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the anxiety app. The document introduces:

• Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).

• Execution Strategy: describes how the test will be performed and 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)

* 1. Project Overview

We are developing a mobile phone app that works with a fitbit to detect the physiological symptoms of anxiety attacks. We will use sensors to track the physiological effects of the stress response and use a rules engine to decipher when a user is experiencing an anxiety attack as opposed to mild stress or exercise etc, it will then:

Launch a graphic with helpful advice for the user.

This advice will be time and location specific i.e. It will not suggest to have a cup of tea if the user is experiencing an anxiety attack at 2am.

The app will graph the anxiety attack, which will be shared with the users therapist. This will allow the user and the therapist to see over time the frequency of these attacks and see if the therapy is effective or not. It will also allow the user and therapist to identify triggers for the users anxiety attacks.

1. objectives and tasks
   1. objectives

the tasks required for the successful testing of this app are listed below. The responsibilities of the testing has been outlined in each testing method section below(4.0).

communication.

The project partners will communicate through facebook messenger as this allows for quick and effective communication in text or speech form but also the sending and receiving of images and videos.

* 1. tasks

**testing.**

The testing process is detailed below. This task will be handled by both project partners.

**Post-testing.**

This task will be dealing with the results and findings from the testing task. This will be handled by both partners.

**Problem reporting**

This task deals with any problems encountered with testing and the any need for retesting. This will be handled by both project partners.

1. scope

**general:**

what is being tested?

The functionality of the app with the fitbit and the usability of the app.

Existing interfaces?

The therapist interface and the user interface.

Integration of all functions?

Functional testing will be performed using synthetic data so we can ensure that the rules engine is operating correctly when it receives the sensor data we want it to identify.

**Tactics:**

How you will accomplish items in general section?

We will accomplish the testing of the functionality of the app with the fitbit by first testing for a response from the fitbit web api. Then testing that the information returned from the web api is correct and finally making sure its is timely. Ie in 5 minute intervals.

We will accomplish the usability of the app for the therapist and user through usability testing and unit testing. Ensuring all elements of the interface are working as expected. We will create test reports to document the results and any bugs that are found.

Positive and negative testing of the rules engine using synthetic anxiety data and exercise data. the expected result is that the anxiety attack causes the notification method to run and the graph to be updated. The exercise data is expected to have no effect.

4.0 TEST STRATEGY

Test Objectives

The objective of the test is to verify that the functionality of anxiety app works according to the specifications. The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing. The final product of the test is twofold:

• A production-ready software;

• A set of stable test scripts that can be reused for Functional test execution.

4.1. Test Assumptions

Key Assumptions

• Production like data required and be available in the system prior to start of Functional Testing

• Exploratory Testing would be carried out once the build is ready for testing

• Performance testing is not considered for this estimation.

• All the defects would come along with a snapshot JPEG format

• The project will provide test planning, test design and test execution support

• During Functional testing, testing team will use preloaded data

• The Test Team will perform Functional testing on anxiety app

• User acceptance test execution will be performed by end users(personas) and Testers .

2.3. Test Principles

• Testing will be focused on meeting the project requirements and quality.

• There will be common, consistent procedures for supporting testing activities.

• Testing processes will be well defined, yet flexible, with the ability to change as needed.

• Testing activities will build upon previous stages to avoid redundancy or duplication of effort.

• Testing environment and data will emulate a production environment as much as possible. • Testing will be a repeatable, quantifiable, and measurable activity.

• Testing will be divided into distinct phases, each with clearly defined objectives and goals. • There will be entrance and exit criteria.

* We will be using IEEE standards documents to ensure our testing is to the highest standard.

1. 29119-1-2013, Software and Systems Engineering - Software Testing - Part 1: Concepts and Definitions
2. 29119-2-2013, Software and Systems Engineering - Software Testing - Part 2: Test Processes
3. 29119-3-2013, Software and Systems Engineering - Software Testing - Part 3: Test Documentation
4. 829-2008, IEEE Standard for Software and System Test Documentation
5. 1008-1987 - IEEE Standard for Software Unit Testing

4.4. Data Approach

• In functional testing, anxiety app will contain pre-loaded test data and which is used for testing activities.

4.5. Scope and Levels of Testing:

1. **Unit tests.**

Definition: ‘’ **UNIT TESTING** is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output’’

PURPOSE:

the purpose of this test is to test individual input components of app such as buttons and text boxes.

SCOPE: First level navigation.

TESTERS: Nigel Brennan and Ikenna Festus.

METHOD: this testing method writes unit tests using Junit to automate the tests on the individual pages i.e MainActivity.java test and FirstPage.java test etc.

TIMING: after each activity is completed in the android studio project.

**TEST ACCEPTANCE CRITERIA**

1. Approved Unit test cases designed.

2. Test cases approved and signed-off prior to start of Test execution

3. Development completed, unit tested with pass status and results shared to Testers.

4. Test environment with application installed, configured and ready to use state.

* 1. –**Code coverage testing.**

DEFINITION: ‘’Code coverage is a measure which describes the degree of which the source code of the program has been tested. It is one form of white box testing which finds the areas of the program not exercised by a set of test cases. It also creates some test cases to increase coverage and determining a quantitative measure of code coverage.’’

PURPOSE:

the purpose of this test is to test the amount of code used when the app is run. This will give us an idea of the statement coverage, branch coverage and condition coverage

SCOPE: First level navigation.

TESTERS: Nigel Brennan and Ikenna Festus.

METHOD: this testing method uses android studios built in code coverage tester to test the amount of code that is used during execution and the code that is not.

TIMING: after each activity is completed in the android studio project and unit tests are completed.

**TEST ACCEPTANCE CRITERIA**

1. Test coverage level approved.

2. Development completed, unit tested with pass status and results shared to Testers.

3. Test environment with application installed, configured and ready to use state.

2. **Integration testing**

DEFINITION: ‘’ Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.’’

PURPOSE:

Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

Scope: after unit testing. Second level navigation.

TESTERS: Nigel Brennan &Ikenna Festus.

METHOD: After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities.

TIMING: after unit tests and coverage testing is completed.

**TEST ACCEPTANCE CRITERIA**

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.

2. Test cases approved and signed-off prior to start of Test execution

3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects

4. Test environment with application installed, configured and ready to use state

3. **System testing.**

DEFINITION: ‘’ System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements’’

PURPOSE: System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. The functionality of the software is tested from end-to-end also.

Scope: after integration testing. Third level navigation.

TESTERS: Nigel Brennan &Ikenna Festus.

METHOD: The test will be performed according to Functional scripts, which are stored in test report\_functional.

TIMING: after integration test is completed.

**TEST ACCEPTANCE CRITERIA**

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.

2. Test cases approved and signed-off prior to start of Test execution

3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects

4. Test environment with application installed, configured and ready to use state

4.**User Acceptance testing.**

DEFINITION: **‘’** Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user’s needs. This requires the product be tested both internally and externally, meaning you’ll need to get it into the hands of your end users for beta testing along with those of your QA team. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.’’

PURPOSE: the purpose of this stage of testing is to get user feedback on the functionality of the app and that the app satisfies users requirements.

SCOPE: Validation level.

TESTERS: Nigel Brennan and Ikenna Festus.

METHOD: This method will use Personas created by the testers to represent the apps user base so we can understand how this app performed in meeting user requirements and functionality.

TIMING: final stage of testing. After unit testing and functional testing is complete.

**TEST ACCEPTANCE CRITERIA**

1. Approved users for testing. User target base document completed.

2. User sample approved and signed-off prior to start of Test execution.

3. Test environment with application installed, configured and ready to use state.

4. TESTERS NOTE: because this is a student project we will use personas that represent real users instead of actual users for ethics reasons and data protection.

***That is all the test methods we expect to complete under the third year project time frame. If there is time we will do 1 final stage of testing:***

5.**Performance Testing.**

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations.

* **Load testing** is the process of putting increasing amounts of simulated demand on our software to verify whether or not it can handle what it’s designed to handle.
* **Stress testing** takes this a step further and is used to gauge how our software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios.
* **Endurance testing,** also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how our system will behave under sustained use. A critical piece of endurance testing is that it helps uncover memory leaks.
* **Spike testing** is a type of load test used to determine how our software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

1. hardware requirements

laptop

android phones

fitbit charge 2 hr.

7.0 test schedule

Test Milestones: testing begins.

Estimated time:

Test Milestones: app runs fully with no crashes or unexpected behaviour.

Estimated time:

Test Milestones: test reports are completed and bugs have been reported and fixed.

Estimated time:

Test Milestones: finished project that is submitted. Testing continues to ensure successful demo.

Estimated time:

8.0 control procedures

8.1 problem reporting

The format that will be used to report any problems discovered in testing.

* 1. change requests.

9.0 Features to be tested

Identify all the software features and combinations of the software features that will be tested.

Login -therapist

Login – user

Register - therapist

Register- user

Forgot password- therapist

Forgot password – user

Email verification – therapist and user.

Database test

Cloud storage test

Graph data test, view graph test

Notification graphic test

Fitbit web api test

Data received from web api test

Rules engine test

The individual components

The integrated modules

The system as a whole

That the requirements are met by the finished app.

10.0 Features not to be tested

Identify all the features and significant combinations of features which will not be tested along with the reasons.

The fitbit hardware. It will be assumed that the hardware we own is fully functional.

11.0 Resources/roles and responsibilities

Nigel brennan – testing/development.

Ikenna festus – testing/development.

12.0 schedules

12.1 major deliverables:

* Test Plan
* Test Cases
* Test Incident Reports
* Test Summary Reports

13.0 dependencies

significant constraints:

* The deadline for this project.
* The scope of the project.
* Testing resource availability.

14.0 **risks/assumptions**

* This is a student project for third year so time for testing is limited.
* Delays in creating all the features and completing all the deliverables for the project could impact the time available for testing.

1. **tools**

* android studio.
* junit.
* robotium.

1. **Approvals**

* Mark roantree - project supervisor.
* Nigel brennan – project partner
* Ikenna festus – project partner.