**TEST PLAN**

**‘Squat Mate’**

**Abstract**This document provides an overview of the project and the product test strategy, a list of testing deliverables and plan for development

**Table of Contents**

**Introduction** ……………………………………………………………………………………………… 3

**1 Test Strategy** ………………………………………………………………………………………… 3 1.1 Scope of testing ……………………………………………………………………….. 3  
 1.2 Test Types ……………………………………………………………………………….. 4  
 1.3 Risks and Issues ……………………………………………………………………….. 5  
 1.4 Test logistics ……………………………………………………………………………. 5

**2 Test Objective** ……………………………………………………………………………………… 5

**3 Test Criteria** …………………………………………………………………………………………. 63.1 Pass criteria …………………………………………………………………………….. 6  
 3.2 Suspension criteria ………………………………………………………………….. 6

**4 Test Approach** ……………………………………………………………………………………… 7

**5 Resource Planning** ……………………………………………………………………………….. 7 5.1 System resources …………………………………………………………………….. 7

**6 Test Environment** …………………………………………………………………………………. 7

**7 Responsibilities** …………………………………………………………………………………….. 8

**8 Schedule** ……………………………………………………………………………………………….. 8

**Introduction**

The Test Plan is designed to describe the scope, approach, resources and schedule of all testing activities of the Squat Mate project.

The plan identifies items to be tested, features to be tested, the types of testing to be performed, our personnel responsible for testing, the schedule required to complete testing, and the risks associated with the project.

1. **TEST STRATEGY**
   1. **Scope of Testing**

**Features to be tested**

All features of the ‘Squat Mate’ that were defined in the Minimum Viable Function requirements need to be tested.

|  |  |
| --- | --- |
| Name | Description |
| MVF1:  Arduino Ethernet Connectivity | This feature is part of the Arduino hardware. This requires a connection between the Arduino and the Ethernet Shield module, with the appropriate code to create a stable data connection. |
| MVF2:  Arduino Sensor Output | This feature is part of the Arduino hardware. This requires a connection between the Arduino and the HC-SR04 Ultrasound, with the appropriate code to output the two variables required in the project; distance and velocity. |
| MVF3:  Repetition Counter | This feature is part of the companion application. This allows the user to keep track of the number of repetitions they have completed in any given set, without reference to the movement being a success (or not). |
| MVF4:  Form Checker | This feature is part of the companion application. A history of recent lifts is returned to the user, indicating if those reps were successful in meeting the squat depth standard (as set by the user following calibration). |
| MVF5:  Velocity Reading | Using the HC-SR04 Ultrasound we can measure a users lift velocity. This feature will process the data received by the sensor via our Arduino device and display it through the companion application at the completion of a successful rep. |
| EVF2:  Additional inclusions:  c. Historical data tracking | This feature is part of the companion application. Expanding on “MVF4 – Form Checker”, this data is permanently stored for future retrieval and analysis. |

**Features not tested**

Remaining Extended Viable Function features of the ‘Squat Mate’ were not tested because they were not included in the project requirements.

* Form checker additional inclusions
  + Light indicators
  + Sound indicators
  + Vibration feedback
* Companion application additional inclusions
  + User profiles
  + Weight tracking
  + Data graphing
  + Data import/export
* Customisable exercise routines
  + Adjustable training plans
  + Additional exercises and lifts

**1.2 Test Types**

In the project ‘Squat Mate’, there are several types of testing methodolgy that should be conducted.

1. Unit Testing

* A type of software testing where individual units of a software are tested, with the purpose to validate that each unit of code performs as expected
* A unit may be an individual function, method, procedure, module or object

1. Integration Testing

* Large projects are divided into smaller parts, with each part subject to multiple iterations
* At the end of an iteration, a module has been developed or enhanced
* This module is integrated back into the software architecture and the entire system is test all together

1. System Testing

* Testing the fully integrated application including all external peripherals ino rder to check how components interact with one another and with the system as a whole
* Verify every input in the application has the desired output

1. Agile methodology

* A project management process, where demands and solutions evolve through the collaborative effort of self-organising and cross-functional teams (and their customers/users)

**1.3 Risks and Issues**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Risk Name | Risk Category | Risk Description | Likelihood | Severity | Degree of Impact | Mitigation Steps | Further Action |
| COVID-19 related procurement delays. | Planning - Procurement | Due to the impacts of a current global pandemic procuring hardware for electronics may become a factor. Lead times for many electronic items both locally distributed and globally imported have been increasing. There is a risk that hardware parts that have been identified as required for this project, as well as parts that have not yet been identified as critical, may not arrive in time to meet project deadlines. | Likely | Moderate | High | To mitigate the risk of this a wide variety of sensors, programmable logic controllers, and connection types can be ordered in advance.  Look at alternate designs to achieve the same functionality.  Consult with peers outside of this project for items they may have around if lead times are excessive. | Risk has not eventuated, no further action taken beyond mitigation steps. |
| Project Scope is Unachievable | Planning – Scope Management | Due to this being an entry level subject there is a risk that the team may have overestimated our capacity to achieve the project deliverables. | Possible | Major | High | Regular and periodic check ins with group team consultant to help check that we are not attempting to plan for a project that is unachievable. | Risk has not eventuated, no further action taken beyond mitigation steps. |
| Team member absenteeism | Execution – Resourcing | For any group project there is a risk of absenteeism, within a university group project this may be temporary or permanent and may occur without warning due to personal, professional or environmental circumstances. | Likely | Moderate | High | Regular team meetings twice a week will help keep the team informed of upcoming risks from everyone. It will also help flag more proactively if a team members involvement is not available or performing. | Risk has not eventuated, no further action taken beyond mitigation steps. |
| Confusion on project direction | Execution - Planning | Without a clear “north star” individual aspects of the project may stray from the intended outcomes. | Unlikely | Moderate | Moderate | Regular communication and check ins to make sure everyone is proceeding in the correct direction | Risk has not eventuated, no further action taken beyond mitigation steps. |
| Incorrect technology choices | Execution - Hardware | As this project has a large research focus there is a risk that the technology chosen in the planning stage may not be adequate for us to achieve or demonstrate particular MVFs. | Likely | Major | Extreme | Ensure that the team proceeds with an agile approach to the product development, if required pivot technology choices. | We have been unable to create a design that adequately demonstrates the products functionality using the Bluetooth module, as a result we have pivoted the project towards using ethernet hardware for connectivity. |

**1.4 Test Logistics**

**Who will test?**

The project should use its group members as the tester, to save on project costs and to further the learning of the members involved.

**When will the test occur?**

The tester will start the test execution of each MVF following the developer reporting that each of the MVF is ready.

**2. TEST OBJECTIVE**

The test objectives are verify the functionality of the ‘Squat Mate’, found at <https://youevenlift.net>, where the testing will focus on guaranteeing all MVF’s operations work normally in a realtime environment.

**3. TEST CRITERIA**

**3.1 Pass Criteria**

This specifies the criteria that denote a successful completion of a test phase for each MVF. The pass rate is 100% and achieving a pass rate is mandatory.

|  |  |  |
| --- | --- | --- |
| Name | Pass | Fail |
| MVF1:  Arduino Ethernet Connectivity | A stable data connection is present.  The Arduino receives a response from an external IP address. | The data connection is not stable and regular drop outs occur.  The Arduino is not able to reach an external IP address. |
| MVF2:  Arduino Sensor Output | The Arduino and HC-SR04 Ultrasound creates a serial port outputs of the distance and velocity variables. | Distance and velocity are not returned, and there is no output available to the Arduino user. |
| MVF3:  Repetition Counter | A counter variable is displayed to the user on the companion app.  This counter is automatically incremental with each movement (squat) by the user.  It is able to be reset to zero. | No counter variable available to the user on the companion app. The count is not incremental with movement by the user.  The counter is unable to be reset. |
| MVF4:  Form Checker | Distance readings are displayed to the user on the companion app.  Lifts that meet the calibation distance are marked “Success”.  Lifts that do not meet the calibation distance are marked “Fail”. | No distance reading is available to the user on the companion app.  Comments around the status of a lift relative to the calibration distance are inaccurate. |
| MVF5:  Velocity Reading | Velocity readings are displayed to the user on the companion app. | No velocity reading is available on the companion app. |
| EVF2:  Additional inclusions:  c. Historical data tracking | Data is stored in an SQL database.  The user can specify the number of their most recent lifts to be displayed in a table. | Information is not stored in an SQL database.  User information fails to be returned to the user in any format. |

**3.2 Suspension Criteria**

If team members report that test cases failed, suspend testing until the development team creates a solution to the failed cases.

**4. TEST APPROACH**

The project is using an agile approach, with weekly iterations. At the end of the week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go. Unit tests will be a key part of the development process.

**5. RESOURCE PLANNING**

**5.1 System Resources**

|  |  |  |
| --- | --- | --- |
| No. | Resources | Descriptions |
| 1 | Arduino and components | Need an **Arduino**, **Ethernet Shield** and **HC-SR04** Ultrasound. All components need appropriate wiring together. |
| 2 | Server | Need a web server with running SQLi server |
| 3 | Test tool | Develop a test tool which can generate the test result to the predefined form and perform a test execution |
| 4 | Network | Setup a LAN with 1 internet line |
| 5 | Computer | 1 computer with minimum Windows 7, 1GB RAM, Intel Pentium or AMD Athlon processor at 1GHz or greater |
| 6 | Smart Phone | Minimum of iOS 7.0 or Android 8.0 |

**6. TEST ENVIRONMENT**

The test environment should be setup as per the figure below.

A close up of a logo

Description automatically generated

**7. Responsibilities**

This section is included to show who is responsible for which functions (MVF’s).

Identify all the group members responsible for test-related managing, designing, preparing, executing, witness, checking and resolving:

* Detail who will prepare test cases, test data and expected results in each MVF Test Stage
* Detail who will execute the testing

|  |  |
| --- | --- |
| Role/Project Members | Responsibility |
| Test Manager | Manage the whole project  Define project directions  Track testing to project schedule |
| System Analyst | Coordinates all test activities and accumulation of test results |
| Test Team | Execute Test Cases, log results, reports defects |
| Development Team | Code software according to the software requirements of all MVF’s |
| SQA members | Take charge of quality assurance  Confirm whether the testing process is meeting specified requirements.  Review and approve Test Plan |

**8. Schedule**

<https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>

<https://www.guru99.com/testing-methodology.html>

<https://zenkit.com/en/blog/agile-methodology-an-overview/>

<https://en.wikipedia.org/wiki/Iterative_and_incremental_development>

<https://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/>

<https://www.wikihow.com/Write-a-Test-Plan>

<https://strongqa.com/qa-portal/testing-docs-templates/test-plan>