# TEST PLAN

VR Boxing Training System for Parkinson’s

*Not Implemented Version 1.8*

**Project Client:** David Blacker

**Project Supervisor:** Shri Rai

**Team Members:**

Aaron Thomson

Liam Kinsella

Kiera Gibson

Jessica Preston

Jake Oorschot

Caleb Macmile

Davinder Singh

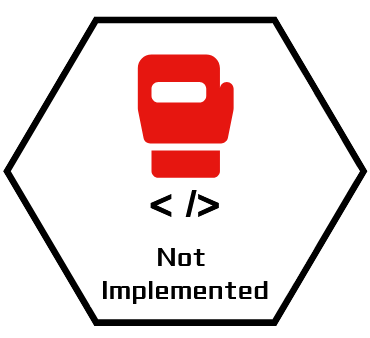


Table of Contents

[**TEST PLAN**](#_heading=h.ab2evmbuaqx7) **0**

[**Version History**](#_heading=h.lk435b3coqyx) **2**

[**Introduction**](#_heading=h.rcxglb96jd8p) **3**

[**Project Team Roles**](#_heading=h.4icmthfhjdq0) **4**

[**Requirements for the Project**](#_heading=h.4aehu1p82521) **5**

[Functional Requirements](#_heading=h.e4jupo1bcwo3) 5

[Non-Functional Requirements](#_heading=h.yjo6bzij8wbn) 5

[**VR Application Testing**](#_heading=h.n8p597g3768x) **6**

[VR Functional Testing](#_heading=h.kuctv1x1go2d) 6

[VR Application Non-Functional Testing](#_heading=h.xw5jglb72shy) 8

[**Database Testing**](#_heading=h.r6a2nel7wikb) **9**

[Database Functional Testing](#_heading=h.2s5qcrv4fp60) 9

[Database Non-Functional Testing](#_heading=h.l2dcl3lgemaw) 11

[**Web Application Testing**](#_heading=h.1o2t2k78830n) **13**

[Web Application Functional Testing](#_heading=h.a8canuq18nk) 14

[**Security Testing**](#_heading=h.dyd93jr810wu) **20**

[**Web Application Performance Testing**](#_heading=h.5ssmulp8sxhj) **21**

[**Network Connectivity Testing**](#_heading=h.cg00ap8qc13b) **24**

[**Unit Tests**](#_heading=h.u8eussi4tvgz) **25**

[Testing Hardware and Software](#_heading=h.pzplced6uwr) 25

[**Test Completion Criteria**](#_heading=h.bd69795v9n3l) **26**

[**Bug Detections**](#_heading=h.5vjar37hlm6w) **26**

[**Conclusion**](#_heading=h.3v87bbyxzqil) **27**

# Version History

**31st October 2021 - Version 1.9**

* Caleb VR application testing

**29th October 2021 - Version 1.8**

* Jake Final Web application functional testing
* Jess Final Web application useability testing

**27th October 2021 - Version 1.7**

* Jess took out Database testing and added reasoning. Renamed unfixed bug list at the end of the document.

**26th October 2021 - Version 1.6**

* Document review and adding more test results - Jess

**25th October 2021 - Version 1.5**

* Jake completed Web application tests
* Added some additional web application tests

**26th September 2021 - Version 1.4**

* Jess reviewing the document and adding more tests for VR
* Jake added some tests, proofread document
* Davinder added tests in web application part

**17th September 2021 - Version 1.3**

* Davinder added more functions of VR application and database to test

**13th September 2021 - Version 1.2**

* Jess and Davinder continuing to work on the Test Plan

**8th September 2021 - Version 1.1**

* Jess added and changed elements of the Test Plan

**8th September 2021 - Version 1.0**

* Davinder created the Test Plan document

# Introduction

This Test Plan serves as a guide for testing the VR Boxing Training System for Parkinsons in its second iteration and accompanies the Design Document. The Design Document focuses on the multiple facets of design for the system, whilst this Test Plan provides specific details of the scope, resourcing and strategy for testing all features of the system. This document can be referred to by internal stakeholders, such as the Project Supervisor, the Project Team, external stakeholders such as the Client, and future project teams who may continue this project. The Requirements and Analysis (R&A) Document also accompanies this Test Plan and is the main point of reference to understand the major functional and non-functional requirements of this project.

**Project Overview**

The aim of this project is to develop a VR boxing system which works as a treatment for patients affected with Parkinson’s disease. The deliverables of this project are a VR application compatible with VR Quest 2 Headset, a web application and database.

**Objective of this Document**

This Test Plan specifies the scope, strategy, resources, and timetable for all of the project VR Boxing Training System for Parkinson’s testing operations. It identifies the objects to be tested, the features to be tested, the types of testing to be conducted, the individuals responsible for testing, the resources and timeline needed to accomplish testing, and the plan's risks.

* Ensure that the VR application is working including all functional requirements.
* Ensure there is connectivity between the VR application, database and web server.
* Ensure that the Database is working and performing daily backups of all data processed through it. Database testing will essentially be performed via web application and VR testing.
* Ensure that high level software code is running as expected.
* Before going live, bugs and issues are found, documented and resolved.

**Test Strategy**

The objective of testing is to verify the functionalities of the VR Boxing System for Parkinsons. This includes testing the functional and non-functional requirements of the deliverables, as specified in the Requirements and Analysis (R&A) Document. A brief requirements list can be found below, however the R&A document should be referred to, for understanding all requirements of this project.

# Project Team Roles

The below team roles have been specified at the beginning of the development lifecycle. This is to ensure that sufficient human resourcing is assigned to each major deliverable for testing. It also ensures that more than one person is testing the major deliverables.

|  |  |  |
| --- | --- | --- |
| Role | Name | Tests |
| VR Application Developer | Kiera | VR Application |
| VR UI | Caleb | Website |
| Database and Web App developer, Project Manager | Aaron | Website |
| VR Application Developer | Liam | VR Application |
| Tester | Jake | Website |
| Tester | Davinder | Website, VR Application |
| Tester | Jess | Website |

# Requirements for the Project

The major functional and non-functional requirements can be found below for the project. It is imperative that this Test Plan accommodates for the testing of these functionalities, to ensure user acceptance, usability, and client satisfaction. For more detailed explanations of requirements, please refer to the Requirements and Analysis Document (R&A).

## Functional Requirements

1. Capture patient movement data
2. Allow web access for clinicians to view and configure treatment plans per patient
3. Allow web access for patients to view account information
4. Store session data on remote server
5. How to box mode - stance and punch training
6. VR Application and web server can send/retrieve data between each other
7. Clinician specified treatment plan
   1. Static target boxing
   2. Dynamic opponent boxing
8. Web Application allows for clinicians to input, delete and update patient data.
9. The database can store data and perform CRUD operations.
10. Secure Authentication for users in the Web Application

## Non-Functional Requirements

1. Application size must fit on hardware (Oculus Quest)
2. VR Application performance is acceptable (FPS isn't too low)
3. The database adheres to Privacy Act 1988
4. VR and Web Application UI is easy to use
5. Web server and database is secure
6. Patient database has backups
7. Additions to Gameplay
   1. Shouting/Chanting
   2. Foot positions
8. Game polish
   1. Music, SFX, VFX

# VR Application Testing

VR Application Testing refers to the testing of the major features within the VR Game. This includes the testing of functional and non-functional requirements, which can be found in the R&A document in detail.

## 

## VR Functional Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functionalities | Description | Expected Output | Test | Result (Pass/Fail) |
| Patient Login to VR Game | Patients enter the account details to log-In into the system on VR. | Access allowed/Access denied | Log in as Mizz Erry (000227) | Pass |
| VR Game successfully loads for a Patient’s boxing session | Boxing sessions for Parkinson’s treatment should be loaded and played by the Patient | Boxing session successfully loaded and playable. | Play TestOne session assigned to Mizz Erry | Pass |
| VR application compatibility | VR application can be connected with Oculus Quest 2 | The VR application connects to the Oculus Quest 2 | Live Demo from Unity | Pass |
| Tutorials | Patient requests to explore a list of tutorial sessions in order to understand basic positions for boxing | Tutorials are displayed and are able to be used by the Patient. | Menu Navigation and Video Player Test | Pass |
| Patient’s movement data is logged | An administrator/clinician request to access patient’s movement data | Patient movement data is logged and saved to the Database. | Unit Testing - playing boxing game | Pass - JSON data file is created. |
| VR Game captures patient movements accurately | Patient movement must be captured in near real-time, to ensure the game is functional and can be played properly. | Patient movement is reflected within the VR game in near real-time, without major delay. | Static training completed | Mixed - If user looks directly down the arms block the screen and the game |
| Game Exit | The game can be quit | The game closes | Quit the game | Fail - Functionality does not exist |
| Menu Testing | The game menu is responsive and changes scenes appropriately | The menu can open and be interactive | Observe menu functionality and flow | Pass |
| Profile Generation | A patient profile is to be generated in the game based on existing database data regarding the patient. | The patient can view their profile within the VR Game | Log in as Mizz Erry and view profile | Fail - Functionality does not exist |
| Static Boxing Game | Play the static boxing game | The static boxing game should be interactive and display a score. | Completed static training | Mixed - Game is interactive, but does not display a score |
| Dynamic Opponent Boxing Game | Play the dynamic opponent boxing game | The dynamic game should have an interactive opponent that will spar back at the player. | Played dynamic training | Pass |
| Displaying of Exercises | The VR requests for exercises and displays them within the VR Game | Exercises should be displayed to the patient if assigned | Viewed and completed exercises inside Session TestOne | Pass |
| Display of error message if patient account does not exist | Error message prompting the patient to create an account appears if they do not have an account | Display error message and prompt for account creation | Enter non-existing user pin | Fail - Does not prompt for account creation, instead displays 0 sessions remaining |
| Updating of sessions | When a player has successfully completed their assigned sessions, the game must update this by sending data to the web server, which commits this to the database. | Database is updated with sessions completed. | Completed and viewed session TestOne | Pass |

## VR Application Non-Functional Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functionalities | Description | Expected Output | Test | Result (Pass/Fail) |
| The VR Game has a user-friendly interface | The VR application is easy to use. All users can use all features of VR Game without having any technical knowledge | All feature options are clearly shown and well designed | Live Demo | Pass |
| Size of VR Game | The VR Game is not too large in size. Users with small memory devices can use this application. | The VR Game application works on low storage devices without any issues |  |  |
| The VR application performance | The VR application must have good performance to interact with the user. The | The VR application have high FPS so, every scene goes smooth on screen | Complete dynamic training | Pass - Game feels smooth to use and doesn’t cause headaches |
| Game Polish | The VR Game should have VFX, SFX and music where appropriate | VFX, SFX and Music is played where specified. | View all scenes | Fail - SFX and Music exists, but is not implemented |

## Web Application Testing

Web Application Testing refers to the testing of the functional and non-functional requirements for the web app. This testing is undertaken in three parts: functionality testing, usability testing and security testing. The functionality and usability tests checked for the web applications ability to meet the functional and non-functional requirements of the project, while the security testing was added for good measure that the application can safely store patient information. The web app is used by Clinicians, Administrators and Clients. These users have varying permissions when accessing the app, along with potential compatibility and stability differences depending on the platform they use to view the web application.

**Functionality Requirements Testing:**

* Clinicians should be able to view patient session history, patient movement data, assign treatment plans, and more.
* Patients should be able to login and view their personal health details and account settings, alongside their game session history and movement data.
* Administrators have the highest privileges within the website, and will have access to all website functionalities, including adding Clinicians, editing and deleting accounts and viewing all data.

**Performance Testing:**

* These are general tests performed by all users, to test the accessibility and usability of the website. This involves testing the website from multiple devices (laptop, computer and mobile) to test the user interface, and ensure the website works as expected when elements are clicked, and data is entered.

## 

## Web Application Functionality Testing

|  |  |  |  |
| --- | --- | --- | --- |
| User | Tests | Expected Output | Result  (Pass/Fail) |
| Clinicians | Log-In into system with personal account | The clinicians are able to log into the system using their username and password | Pass |
| Clinicians | Assign sessions to patients | The clinicians are able to assign sessions to their patients | Pass |
| Clinicians | Create activity to be used in sessions   * rest, dynamic and static types created * no name & named activities * absurd hit counts, hit locations and hit durations | Clinicians can use an activity creator to set targets for sessions | Pass - dynamic activity not configurable |
| Clinicians | Remove old sessions and activities | The clinicians are able to remove and add sessions and activities | Pass |
| Clinicians | View patient’s movement data | The clinicians are able to see patient’s recovery performance | Fail - no sessions in table |
| Clinicians | View session history   * see upcoming sessions | The clinicians are able to look on number of repeats done by patient on any particular treatment exercise | Pass - unable to test completed sessions |
| Clinicians | Filter tables on display   * tested sessions * tested activities * tested patients | The clinicians should be able to filter the displayed information from any given table | Pass |
| Clinicians | Add patients   * names with number * weight & heights with letters * large height and weight value * future D.O.B * All character types in Notes | The clinicians are able to add more patients for treatment | Pass |
| Clinicians | View Patient’s information   * filter table by patient | The clinicians are able to look at the session information of particular patients | Pass - Untested completed session reviews |
| Clinician | Update Patient’s information | The clinicians are able to edit the account information of patients | Pass |
| Clinicians | Delete patients | The clinicians are able to remove any patient once they have completed their treatment | Pass |
| Clinicians | Log-out | The account should be logged out and met with the login screen | Pass |
| Clinicians | Web tutorial | Clinicians have a how to use page for the web application | Pass |
| Head Clinicians | Add other clinicians   * names with & without numbers * only used first/last name | The head clinicians are able to add other clinicians users | Pass |
| Head Clinician | Delete clinicians | The head head clinicians are able to delete unwanted clinician users | Pass |
| Head Clinicians | Assign sessions to patients | The head clinicians are able to assign sessions to their patients | Pass |
| Head Clinicians | Create activity to be used in sessions   * rest, dynamic and static types created * no name & named activities * large hit counts, hit locations and hit durations * set hit count to letter | head clinicians can use an activity creator to set targets for sessions | Pass - dynamic activity not configurable |
| Head Clinicians | Remove old sessions and activities | The head clinicians are able to remove and add sessions and activities | Pass |
| Head Clinicians | View patient’s movement data | The head clinicians are able to see patient’s session performance | Untested - no completed sessions to review |
| Head Clinicians | View session history   * see upcoming sessions | The head clinicians are able to look on number of repeats done by patient on any particular treatment exercise | Pass - completed sessions untested |
| Head Clinicians | Filter tables on display   * tested sessions * tested activities * tested patients * tested clinicians | The clinicians should be able to filter the displayed information from any given table | Pass |
| Head Clinicians | Add patients   * names with number * weight & heights with letters * large height and weight value * future D.O.B * All character types in Notes | The head clinicians are able to add more patients for treatment | Pass |
| Head Clinicians | View Patient’s information   * filter upcoming sessions by patient | The head clinicians are able to look at the session information of patients | Pass - completed sessions untested |
| Head Clinicians | Update Patient’s information | The head clinicians are able to edit the account information of patients | Pass |
| Head Clinicians | Delete patients | The head clinicians are able to remove any patient once they have completed their treatment | Pass |
| Head Clinicians | Web tutorial | Head clinicians have a how to use page for the web application | Pass |
| Administrators | Log-In into system with account pin | Administrators can successfully log-in into the system using their username and password | Pass |
| Administrators | Add other users   * clinicians * patients * head clinicians * admins | Administrators can add other user roles according to need | Pass |
| Administrators | Remove other users   * clinicians * head clinicians * admins * patients | Administrators are able to remove other users at anytime | Pass |
| Administrators | View and modify clinician information | Administrators can update and view information of clinicians | Pass |
| Administrators | View and add Patient’s information | Administrators are able to view and update information of patients | Pass |
| Administrators | Assign and remove treatment plans for patients | Administrators can add and remove treatment sessions for patients | Pass |
| Administrators | View patient’s movement data | Administrators are able to access improvement of all registered patients | Fail - no completed sessions to test |
| Administrators | View session history | Administrators can view number of sessions completed by all patients according to their treatment plan | Fail - no completed sessions to test |
| Administrators | Delete patients | Administrators can remove any patients once the treatment is completed | Pass |
| Administrators | View Patient Usernames | Administrators can view patient usernames to allow patients to login etc. | Pass |
| Administrator | Web tutorial | Admins have a how to use page for the web application | Pass |
| Patients | Log-In into the system with patient role account | Patients are able to log-in into the system using their account pin | Pass |
| Patients | View account details | Patients can view their account information once they are logged into the system | Fail |
| Patients | Update account information from settings page | Patients can update their account information from the settings page. | Fail |
| Patients | Web tutorial | Patients have a how to use page for the web application | Pass |
| Patients | View session plans | Patients are able to see the list of session plans coming up and completed | Pass |
| Patients | View movement data | Patients can look their recovery performance throughout the treatment | Fail - no completed sessions to test |
| Patients | View session history   * see upcoming sessions | Patients can view the number of sessions done by them according the treatment plan | Pass - no completed sessions to test |
| Patients | Log out | Can log out - be presented with login page | Pass |

## 

## 

## Security Testing

Brief security-related tests were performed to identify any security misconfigurations and vulnerabilities. Nmap and Nikto were used to enumerate the web application. The results of these tests show that the Murdoch firewall is sufficient at blocking port scanning and suspicious web application scanning.

**Nmap Results:**

* Firewall blocks too many retransmission attempts so the Murdoch firewall is sufficient.

**Nikto results:**

* Few security headers missing:
  + + The anti-clickjacking X-Frame-Options header is not present.
  + + The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
  + + The X-Content-Type-Options header is not set.

**Cookie re-use:**

* Manipulation of the cookie field could allow an attacker to take over another user’s session. This may be done via logging in as Patient A, where Attacker B steals Patient A’s cookie string, and uses this value to send requests to the server. The server was not susceptible to this attack.



**Injection of predefined username when creating a new Patient:**

* Providing a 6 digit number as a username and sending the request to the server generates an error. This is good, as an attacker cannot set their PatientID/Username.

**Exhaust Server Resources:**

* If an attacker (e.g. disgruntled Clinician or attacker with stolen credentials) wanted to overwhelm the server, it is possible by sending a bulk amount of requests to /cgi-bin/ajax/ajaxsavepatient.cgi with a string such as:

patientid=&firstname=jess&lastname=preston&dob=06%2F02%2F2001&condition=swag+test&height=175&weight=52

* This will populate the database with patients with the same information. This could only be performed by a logged in Administrator or Clinician.

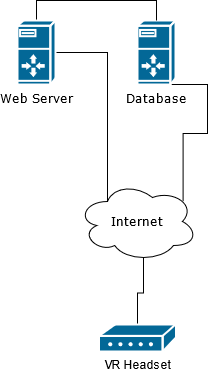


## 

## Web Application Performance Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Functionalities | Description | Expected Output | Test | Result(Pass/Fail) |
| Web logs | The web application should be able to track site performance and traffic record | The web application is tracking traffic record and performance. | view sessions page | Pass |
| Role display | Users should be able to see their role under their account information on the application. | User can identify their role easily on the web application | log in and view settings | Fail |
| Data Security | The web application must be a secure place where users can enter their personal information confidentially. | All information of users is secured and encrypted. Traffic cannot be sniffed. | Patient signs into the website. | Pass - 6 digit not secure, 8 character policy should be implemented. |
| Data input sanitization | The web application should not be susceptible to SQL inject, buffer overflows, etc. | All inputs from the user should be sanitised and inputs should only be accepted in a “regular” format. | Input fuzzing & SQL strings sent in place of user inputs to the system. | Pass |
| Compatibility | The web application should be compatible with all browsers and operating systems. | The web application supports all browsers so users can access it through any suitable browser and OS. | The website is accessed from a mobile. | Pass |
| Compatibility | The web application should be compatible with all browsers and operating systems. | The web application supports all browsers so users can access it through any suitable browser and OS. | The website is accessed from a laptop. | Pass |
| Website Sessions | The website should remember the user when the tab closes. The user should not have to login again within a short timespan (4 hours) | The user should remain logged-in | The user accidentally closes the browser tab, and re-opens the website in a new tab. | Fail |
| User Interface - Clicking Elements | When elements are clicked (nav bar, buttons) they should be responsive. | User is navigated to another page when the navbar is clicked. | Navbar clicked by the user. | Pass |
| User Interface - Data Entry | When users enter in data into the website - the website should respond with success/failure accordingly | The website generates a success message when the data is saved to the database. | New users created, new sessions created, user/sessions edited | Pass - some feedback is given to users |
| Data Validation at login | The input fields should reject bad input on the client side | The website rejects bad user input on the client side and server side. | User enters garbage data into login input | Pass |
| Data Validation at patient creation | The input fields should reject bad input on the client side | The website rejects bad user input on the client side and server side. | User enters wrong/garbage/invalid data into input fields. | Pass |
| How to use | The application should have a “how to” section for users to understand the application. | How to on the dashboard and pop up tooltips. | Check for how to page  Check pop ups | Fail |

# Network Connectivity Testing



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network | Description | Expected Output | Actual Output | Result  (Pass/Fail) |
| VR Application<>Database | VR application is connected to database in order to receive and send information | Connected/Not connected | VR Application is able to read and write data to the database. | Pass |
| Web Server<>VR Application | Web Server is connected with VR Application | Connected/Not connected | Web server activities/sessions can be sent to the VR Application | Pass |
| Database <> Web Server | Database is connected with web server to backup files and data | Connected/Not connected | Connected. Data from database can be displayed on webpages | Pass |

## Testing Hardware and Software

The below list documents the required tools that are necessary for the testing of the VR Application, Database and Web Application. According to the procurement section of the Project Management Plan, these assets are available to the project team at no expense.

* Movement tracker for VR Headset
* Quest 2 VR Headset
* Unity test runner
* Unity
* Automation Tools
* Visual studio

# Test Completion Criteria

It is important that testing is conducted throughout the Agile Development Lifecycle, to ensure code functionality, application usability and functionality. Tests are considered complete when they have been executed and documented in this Test Plan. Once documented in this Test Plan, it is expected that the tests are reviewed by other project team members to verify and ensure consistency. Test results are then viewed by developers to fix any bugs, defects or issues.

**The following are the criteria that must be achieved to complete testing:**

* Tests are 100% completed and documented within the Test Plan
* All manual and automated test cases have been completed.
* All bugs are fixed or will be fixed in the next release, and documented.

# Conclusion

This document provided the test strategies and plans to ensure that all important components of this project are working as expected. This document helps to detect if any feature of deliverable components is not working. The test plan establishes rules that must be followed during the whole testing procedure. The bug detection list will also assist future teams when undertaking iterations of the boxing for health project.