VIRTUAL REALITY SYSTEM FOR TREATING EATING DISORDERS

By

NANCY YEN NGAN PHAM

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelor's degree

With Honors in

Biomedical Engineering

THE UNIVERSITY OF ARIZONA

M A Y 2 0 2 0

TEAM 19085

MEMBERS AND CONTRIBUTIONS:

All members contributed to the overall system design, filming, and documentation.

NANCY YEN NGAN PHAM: Narrative design, acceptance test plan, video film editing

ALEXANDER FELIPE REYES: Simulation software graphical user interface design

STEVEN LONG DINH: Narrative design and assistance in simulation software design

ZAYNAH ARZIE KMEID: Narrative design and acceptance test plan

KARM JAMIL AL HAJHOG: Calorie counting system and system data analysis

CURT JOHN BAUTISTA BANSIL: Sensor and graphical user interface software design

PROJECT SPONSOR: DR. MARVIN SLEPIAN AND THE ARIZONA CENTER FOR BIOMEDICAL INNOVATION (ACABI)

PROJECT MENTOR: HEATHER HILZENDEGER

Project goal: To develop a virtual reality system aimed at treating eating disorders with physiological monitoring and self-assessment capabilities.

Approximately 30 million individuals in the United States suffer from eating disorders. Eating disorders have the highest mortality rate of any mental illness. Our group is attempting to make a dent in the health impact of eating disorders via the use of "virtual reality therapeutics."

The EDVRS is a multi-level, multi-branching experience in which users diagnosed with eating disorders will be exposed to increasingly complex food and social scenarios through a virtual reality program created in Unity and displayed on the Oculus Go, a head-mounted VR display. The system designed is a novel treatment approach that contains a progressive, self-directed software exposure experience to "desensitize" and acclimate individuals suffering from anorexia or bulimia.

The therapy will include physiological monitoring and self-assessment to determine the effect of the treatment. Monitoring of the patient's heart rate and galvanic skin response will occur with the biosensors of a wearable device and utilized to characterize their stress response. These metrics will be displayed in real time and will allow the user and administrator to decide the speed of progression through the treatment. The data will be stored for future use and analysis to identify best practices in treating these disorders.

Table of Contents

1.0 Scope

This final report contains development and operation information for our product the Virtual Reality System for Treating Eating Disorders. Documentation includes the final design, technical data package, acceptance tests and results, verification and system requirement status, final budget report, and our conclusions on the project.

The purpose of our project was to develop a virtual reality system aimed at treating eating disorders with physiological monitoring and self-assessment capabilities. Approximately 30 million individuals in the United States suffer from eating disorders. Eating disorders have the highest mortality rate of any mental illness. Our group is attempting to make a dent in the health impact of eating disorders via the use of "virtual reality therapeutics."

The EDVRS is a multi-level, multi-branching experience in which users diagnosed with eating disorders will be exposed to increasingly complex food and social scenarios through a virtual reality program created in Unity and displayed on the Oculus Go, a head-mounted VR display. The system is designed as a novel treatment approach that contains a progressive, self-directed software exposure experience to "desensitize" and acclimate individuals suffering from anorexia or bulimia.

The therapy will include physiological monitoring and self-assessment to determine the effect of the treatment. Monitoring of the patient's heart rate and galvanic skin response will occur with the biosensors of a wearable device and data will be utilized to characterize their stress response. These metrics will be displayed in real-time and will allow the user and administrator to decide the speed of progression through the treatment. The data will be stored for future use and analysis to identify best practices in treating these disorders.

Using the Engineering V Model, we have defined the project using a concept of operations and finalized the design using our system requirements. Once this was outlined, we tested the subsystems. Next, we verified and validated our subsystems to ensure we met the goals as established. To complete the project in its entirety, we need to integrate the calorie counting and level up system into our simulation code and finally integrate the entire system with verification and validation of the system as a whole. We were unable to complete these due to COVID-19.

2.0 Technical Data Package

2.1 Introduction

2.1.1 Abstract

The technical data package of this project includes information about the design of the Virtual Reality System for Treating Eating Disorders. The goal of this system is to help support patients with anorexia and bulimia to become more comfortable and develop a healthier connection with food. The system uses 360-degree video that will place the user in scenes where they will be exposed to different situations that vary in difficulty. The user will also be given several choices at each level that include options for food type and amount. To move to the next level, the user must fall into a certain calorie count range that is calculated using their weight, sex, and age. Moreover, to keep track of the progress of the patients, psychological data will be self-reported by the patient following the simulation. We will be recording the heart rate and galvanic skin response of the patient to monitor the parts of the virtual reality simulation that correspond with increased psychological reactions. Furthermore, after each simulation, the user will fill out a self-evaluation form to give feedback directly from the user's perspective. All these features that are included in the design will help the patients with eating disorders to react better in similar situations they may face, and give a clearer view to their physician of how they are progressing.

Disclaimer: The following the sections reflect design ideas and concepts however, they have yet to be implemented into our system:

- EDVRS Graphical User Interface (2.5.2.5.2)
- Level System and Self-Evaluation (2.5.2.5.3 & 2.5.2.5.4)
- Verifying Level-up (2.5.3.4.4)
- Complete system integration and verification

2.1.2 Abbreviations

EDVRS: Virtual Reality System for Treating Eating Disorders
VR: virtual reality
GSR: galvanic skin response
HR: heart rate
GUI: graphical user interface
TCP: transmission control protocol
LED: light-emitting diode
BTLE: Bluetooth-Low Energy

2.2. Referenced Documents

2.2.1 EDVRS System Requirements Document, 11/24/2019, rev(-) 10001
E4 Wristband User's Manual

2.3 System Architecture

2.3.1 System Block Diagram & Architecture

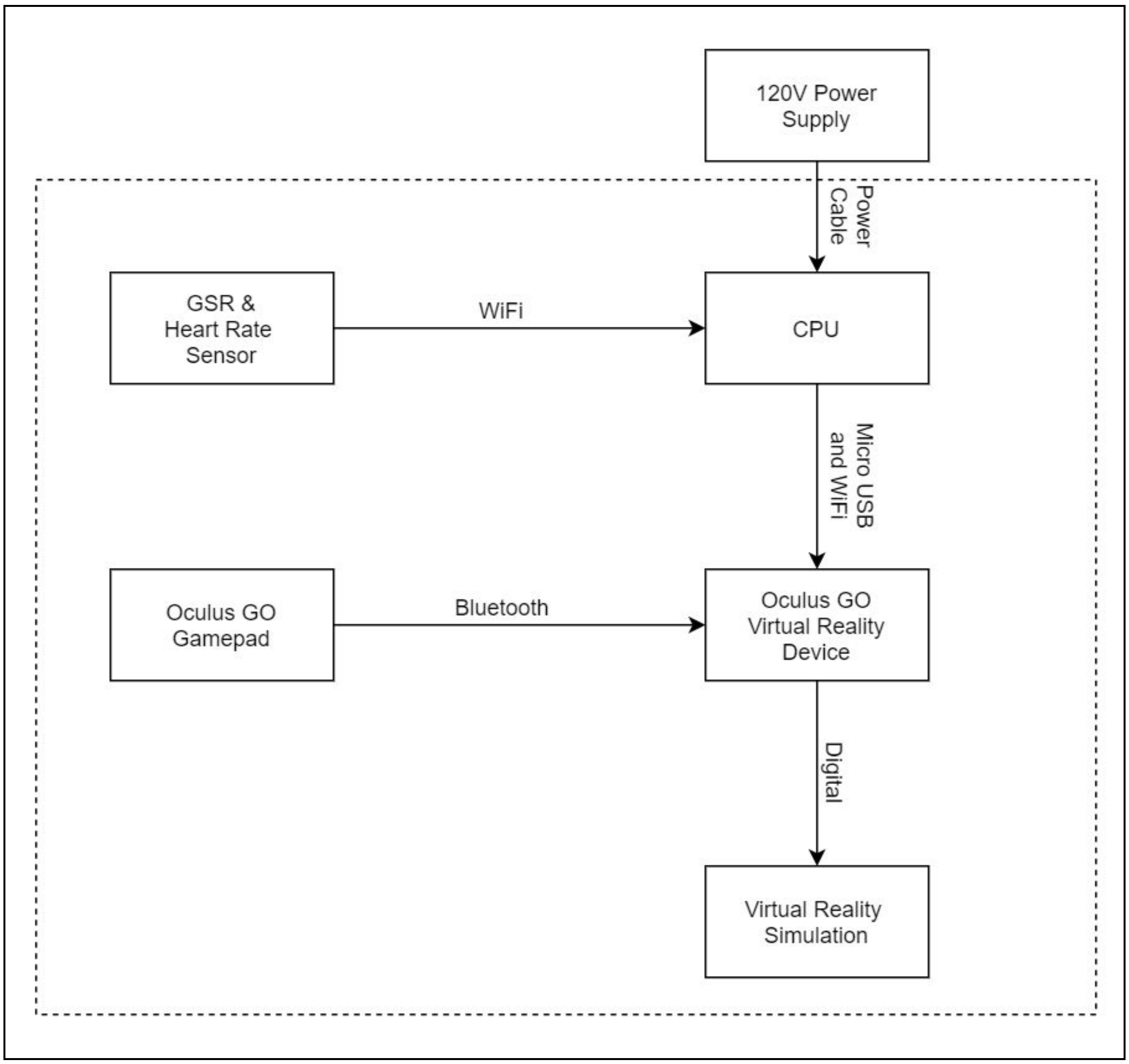


Figure 1. System Block Diagram of the EDVRS

The dashed square in Figure 1 represents the boundary of our system. All the modules inside the square are within our control, while everything outside the square is an input to our system that is outside of our control. Our system consists of 5 main components. The first component, our CPU, receives input from a 120V power supply through a power cable. This provides our CPU with power to function. The CPU receives input from the next system component, the sensor module, in the form of a wifi signal. This consists of our heart rate and GSR sensors, which feed physiological data from the patient to the computer. The CPU is connected to the Oculus GO Virtual Reality Device through a micro USB cable and Wi-Fi. The micro USB cable is used to charge the device and upload our simulation software to it. The Oculus Go Gamepad sends a Bluetooth signal to the Oculus Go VR Device which is composed of the user’s inputs during the simulation, allowing them to make food selections. Lastly, the Oculus Go VR Device is connected to our Virtual Reality Simulation through a digital signal as the VR Device displays the simulation to the user.

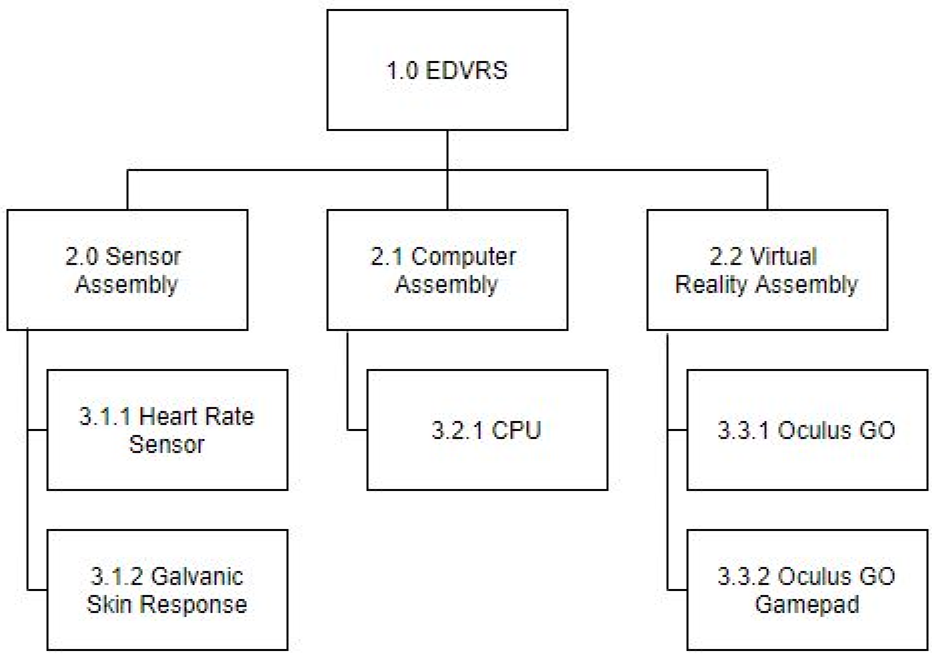


Figure 2. System Architecture of the EDVRS

A diagram of the system architecture for the Eating Disorder Virtual Reality Device. The top level represents the system as a whole. Tier 2 contains the three subsystems, which are the Sensor, Computer, and Virtual Reality subassemblies. Tier 3 contains the components of the subsystems.

2.3.2 System Requirements

VRS for Treating Eating Disorders Verification Matrix	Verification Method T= Test A= Analysis D= Demonstration I = Inspection			
Requirements	**T**	**A**	**D**	**I**
1.0 Performance				
1.1 The system shall run the simulations at a minimum rate of 60 frames per second as required by the Oculus Store, a necessary platform for mass distribution and consumer use.	X	X		
1.2 The wireless devices shall follow IEEE 802.11 wireless communication standards.	X			
1.3 The system shall be created using Unity version 2019.2.7.				X
1.4 The system shall operate on the first generation of the Oculus Go Virtual Reality platform.				X
1.5 The external computer shall be a portable device.				X
1.6 The external computer shall operate on the Windows 10 OS.				X
1.7 The external computer shall use a standard 120-volt outlet for power.				X
1.8 The external computer shall upload the virtual reality simulation to the Oculus Go using a micro-USB cable.				X
1.9 The galvanic skin sensor shall read a standard clinical minimum of 0 μS and a maximum of 25 μS.				X
1.10 The galvanic skin response sensor shall read in accordance with the Empatica E4 Technical Specifications Document.		X		
1.11 The heart rate sensor shall read a standard clinical minimum heart rate of 50 beats per minute to a maximum of 200 beats per minute.				X
1.12 The heart rate sensor shall read with +/- 10% error rate with respect to a gold standard ECG to detect the change in heart rate.	X			
2.0 Response Time				
2.1 The heart rate sensor shall sample at a rate of at least one sample every five seconds to capture trends throughout the simulation, which is supported by our research that heart rate will change gradually rather than instantly.	X	X		
2.2 The galvanic skin response sensor shall sample at a rate of at least one sample every five seconds to capture trends throughout the simulation, which is supported by our research that the galvanic skin	X	X		

response will change gradually rather than instantly.				
2.3 The computer shall store data from the heart rate sensor.				X
2.4 The computer shall store data from the galvanic skin response sensor				X
3.0 VR Simulation				
3.1 The system shall have three scenarios.			X	
3.2 The scenarios shall each have at least three different branching endings.			X	
3.3 The system shall offer at least two options per branch for the user.			X	
3.4 The system shall require the user to be within +/- 15% of their ideal calorie intake before unlocking the next difficulty level.	X		X	
3.5 The system shall provide the user with a self-evaluation upon completion of a level.			X	

Table 1. System Requirements and Verification Summary

2.3.3 Traceability to Subsystems

VRS for Treating Eating Disorders Requirements Decomposition Summary			
System Requirements	**Subsystems**		
	Sensor Assembly	**Computer Assembly**	**VR Assembly**
1.1 The system shall run the simulations at a minimum rate of 60 frames per second as required by the Oculus Store, a necessary platform for mass distribution and consumer use. (T/A)			(T-Direct) The Oculus Go shall run the simulation at a minimum rate of 60 fps as required by the Oculus Store.
1.2 The wireless devices shall follow IEEE 802.11 wireless communication standards(T)	(T-Direct) The sensors shall be capable of using IEEE 802.11 wireless communications standard.	(T-Direct) The computer shall be capable of using the IEEE 802.11 wireless communication standards.	(T-Direct) The Oculus Go shall be capable of using the IEEE 802.11 wireless communication standards.
1.3 The system shall be created using Unity version 2019.2.7.(I)		(I-Direct) The Unity version on the computer shall be checked.	(I-Direct) The Unity version is 2019.2.7 shall be checked.

1.4 The system shall operate on the first generation of the Oculus Go Virtual Reality platform. (I)			(I-Direct) The Oculus Go Virtual platform shall run on the first generation.
1.5 The external computer shall be a portable device. (I)		(I-Direct) The computer shall be portable.	
1.6 The external computer shall operate on the Windows 10 OS. (I)		(I-Direct) The computer shall operate on Windows 10.	
1.7 The external computer shall use a standard 120-volt outlet for power.(I)		(I-Direct) The computer shall operate using a 120-volt outlet for power.	
1.8 The external computer shall upload the virtual reality simulation to the Oculus Go using a micro-USB cable. (I)		(I-Direct) The simulation shall be uploaded via micro-USB to the Oculus Go.	(I-Direct) Oculus shall accept a micro-USB cable
1.9 The galvanic skin sensor shall read a standard clinical minimum of 0 μS and a maximum of 25 μS. (I)	(I-Direct) The sensor shall be able to read a galvanic skin response between 0 μS and 25 μS.		
1.10 The galvanic skin response sensor shall read in accordance with the Empatica E4 Technical Specifications Document.	(A-Direct) The sensor shall be tested in accordance with the data provided in the E4 Empatica study.		
1.11 The heart rate sensor shall read a standard clinical minimum heart rate of 50 beats per minute to a maximum of 200 beats per minute. (I)	(I-Direct) The sensor shall be able to read a heart rate range between 50 and 200 beats per minute.		

1.12 The heart rate sensor shall read with +/- 10% error rate with respect to a gold standard ECG to detect the change in heart rate. (T)	(T-Direct) The sensor shall be tested at the same time as a gold standard heart rate sensor to verify accuracy.		
2.1 The heart rate sensor shall sample at a rate of one sample every five seconds to capture trends throughout the simulation, which is supported by our research that heart rate will change gradually rather than instantly. (T/A)	(T/A-Direct) Samples shall be taken at a rate of 0.2 Hertz		
2.2 The galvanic skin response sensor shall sample at a rate of one sample every five seconds to capture trends throughout the simulation, which is supported by our research that GSR will change gradually rather than instantly. (T/A)	(T/A-Direct) Samples shall be taken at a rate of 0.2 Hertz		
2.3 The computer shall store data from the heart rate sensor. (I)		(I-Derived) The computer shall receive and store information from the heart rate sensor.	
2.4 The computer shall store data from the galvanic skin response sensor. (I)		(I-Derived) The computer shall receive and store information from the GSR sensor.	
3.1 The system shall have three scenarios. (D)			(D-Direct) The system shall have three scenarios.
3.2 The scenarios shall each have three different branching endings. (D)			(D-Direct) The scenarios shall each have three different branching endings.
3.3 The system shall offer two options per branch for the user. (D)			(D-Direct) The system shall offer two options per branch for the user.

3.4 The system shall require the user to be within +/- 15% of their ideal calorie intake before unlocking the next difficulty level. (D/T)			(D/T-Direct) The system shall require the user to be within +/- 15% of their ideal calorie intake before unlocking the next difficulty level.
3.5 The system shall provide the user with a self-evaluation upon completion of a level(D)			(D-Direct) The system shall provide the user with a self-evaluation upon completion of a level.

Table 2. System Requirements Decomposition Summary

2.4 Prediction of Performance

2.4.1 Heart Rate Sensor Performance

The normal heart rate range is between 60-100 beats per minute. As the E4 wristband can sample at a frequency of 64Hz, it is expected to accurately capture the subject's heart rate in real-time. The program will take a heart rate reading initially to establish a baseline for the patient. As the patient is exposed to more stress-inducing scenarios, their heart rate is expected to rise. Judging by the sample rate that the E4 sensor is capable of, the sudden increases should be clearly shown and not affect the readings in any manner. The accuracy of the sensor must be within +/- 10% of a Gold Standard ECG.

Age	Target Heart Rate Zone (50-85%) (BPM)
20	100-170
30	95-162
35	95-157
40	90-153
45	88-149
50	85-145
55	83-140
60	80-136

65	78-132
70	75-128

Table 3. Expected 'Normal' Heart Rates for varying Ages

2.4.2 Galvanic Skin Response Sensor Performance

The normal range for galvanic skin response is between 1-20 microsiemens. This reading increases with the activity of the patient's sweat glands. As the patient sweats more, skin conductivity is increased. The increase of sweat is not an instantaneous occurrence, so the 4Hz sample rate of the E4 GSR sensor will be adequate in measuring this change. The accuracy of the sensor will read in accordance with the Empatica E4 Technical Specifications Document

2.4.3 Oculus Go Frame Rate Performance

The Oculus store requires a minimum frame rate of 60 frames per second in order to distribute an application through them. The Oculus Go can operate at a maximum of 72 frames per second, giving 12 frames of leeway in development. As the program is being designed in Unity, the frame rate can be hard-coded, but must still be tested to see if it is running optimally on the hardware. As the device is displaying almost entirely filmed footage and not having to render complex models, we predict that the 60 frames per second minimum will be met.

2.4.4 Calorie Intake Measurement Performance

Using the Mifflin-St Jeor equation, a user's ideal daily calorie consumption will be determined. This is based on their weight, height, age, and sex. This amount will then be cut in half to account for the fact that the simulation is representative of half a day's caloric consumption. The calculated quantity shall then be used as their baseline for the simulation. This equation is widely accepted and utilized by medical professionals and providers within this field, however, the equation does not account for inaccuracies resulting from diseases or other biological factors that may affect the patient's ideal intake. We recognize these possible variabilities but accept the equation as professionals in the field do. Given the purpose of our system, the equation satisfies the objective.

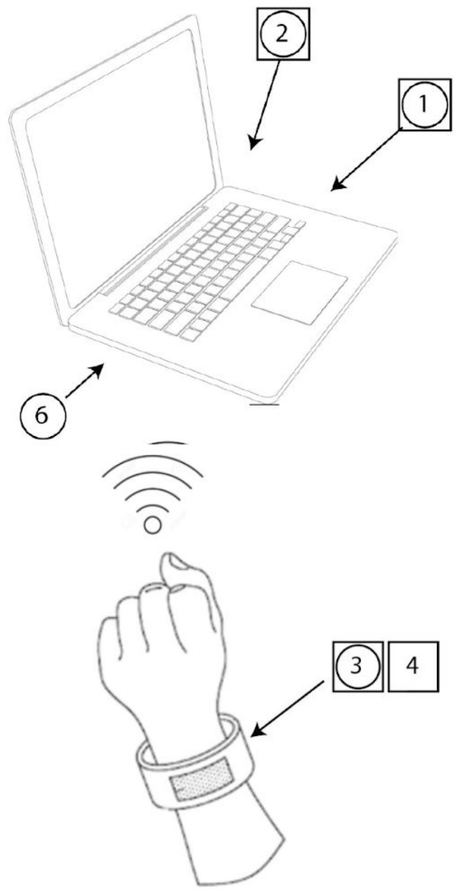
2.4.5 Calorie Threshold Performance

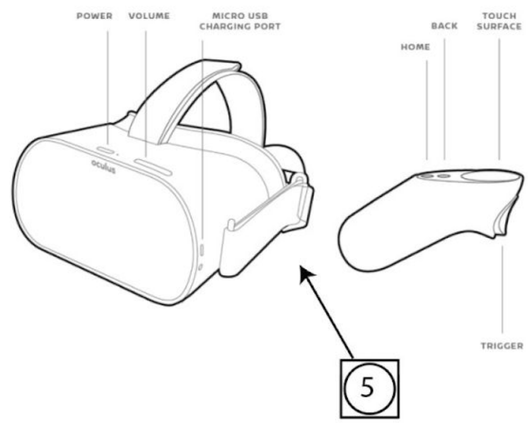
Using the ideal calorie intake calculated with the Mifflin-St Jeor equation, a threshold is created for the user that measures +/- 15% of half of that amount. The +/- 15% threshold will be applied to 50% of the ideal calorie intake calculated. Fifty percent of the daily intake is used because the simulation will be run on the main meal only (dinner). Fifteen percent is used because 20% lower is defined as losing weight while 20% higher is defined as gaining weight, and the intent is to help the patient maintain their weight. As with the calorie intake measurement, there can be other biological factors that can affect the percentage needed to maintain weight, but for the purposes of this simulation, the amount chosen will suffice. Moreover, for anyone using the simulation, a ratio of their BMI to 15.6 will be multiplied to the predefined calories in the system to make the simulation usable by everyone and accepting of all BMI values.

2.5 Drawings

2.5.1 Assemblies & Schematics

2.5.1.1 EDVRS Top Assembly Drawing





Find	Part Description	Vendor	Part #	Qty.
1	EDVRS Software (Unity)	Team 19085	100400	1
2	Computer	ASUS	ASUS F510U - X540UADS51	1
3	Empatica Wristband	Empatica	E4 - Rev. 2	1
5	Oculus Go - All-In-One VR (64 GB)	Facebook Technologies	301-00104-01	1
6	Bluetooth dongle for wristband connection	Silicon Labs	BLED112	1

Notes:

1. Connect the Oculus Go to the computer via micro-USB. Build the file to the Oculus Go using Android Debugger Bridge.

2. Power on computer. Connect the E4 dongle to the computer and ensure connectivity with the wristband. At the end of a data collection session, the data will be uploaded to the computer. Once downloaded, it can be viewed from a desktop application and stored for later analysis.

3. The E4 Wristband is both our heart rate sensor and galvanic skin sensor.

4. Power on the device and ensure connectivity to the computer.

5. Power on headset. Navigate to treatment application, select, and enter.

Company: Team 19085		Title: Virtual Reality Treatment System for Eating Disorder Top Assembly	
		Drawing No: 100100	Rev(-)
			Sheet 1 of 1

Figure 3. EDVRS Top Assembly Drawing

2.5.1.2 Parts

Part Number	Document
100100	Virtual Reality System
100100-ATP	VRS Acceptance Test Procedure
100200	Sensor Assembly
100210	E4 Wristband
100300	Computer Assembly
100310	ASUS F510U
100400	VRS GUI/Display
100410	User Choice
100420	Play Submenu
100430	Settings Submenu
100440	Self Evaluation
100500	VR Assembly
100510	Oculus Go

2.5.1.3 Virtual Reality Narrative

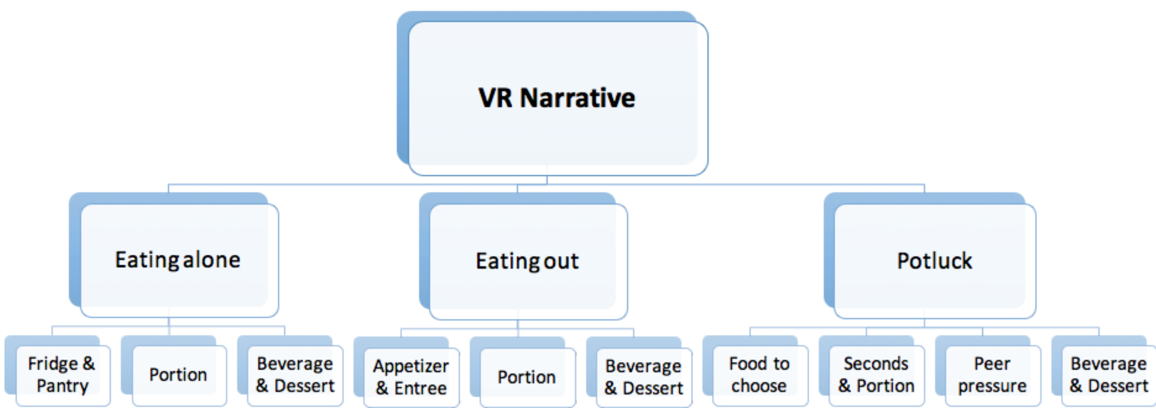


Figure 4. EDVRS Narrative Scenarios and Decision Branches

Shows the basic structure of our simulation narrative. The home screen will give the user 3 level options ranging from easiest to hardest (eating alone, eating out, potluck). Once the user selects a level, they enter into that simulation. In that simulation, they will be prompted to make decisions relating to food: meal size, appetizers, type of food, beverage choice, etc. The figure above reflects this branching system and what each scenario will prompt the user to make choices about.

2.5.2 Software Design Documents (SDD)

2.5.2.1 Scope

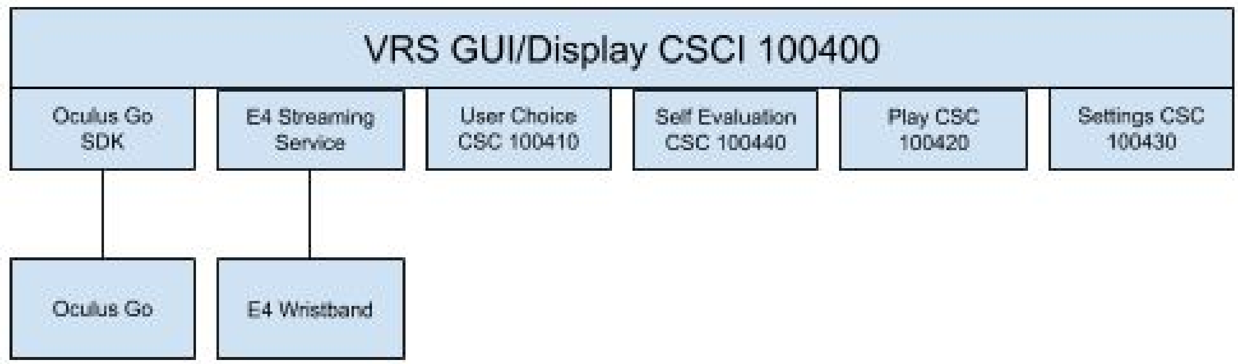


Figure 5. Software Architecture

GUI Reference	Inputs	From	Outputs	To	Description
Oculus Go	Click on the application icon	User (GUI)	run EDVRS application	Oculus Go SDK	The call to run the VR application for the user
E4 Wristband	Connect to the dongle and start a session	E4 Wristband	obtain data from streaming service	Windows application	The process to stream data to a web application for a session
User Choice	Choice	User (GUI)	save choice to data file	Data file	Saves the user's choice for the branch to a data file, as well as the updated total calorie count the user has after making that choice
Self Evaluation	Rating	User (GUI)	save user's rating to data file	Data file	Save the user's rating for a question to a data file
Self Evaluation	Feedback	User (GUI)	enter script feedback	Data file	Save the user's feedback for the scenario to data file
Sys Utility	Play	User (GUI)	calls "Play" submenu	GUI (display)	Changes from GUI main menu to "Play" submenu
Play	Level 1	User (GUI)	executes "Level	GUI	Runs "Level 1"

			1" scenario for user	(display)	scenario for user
Play	Level 2	User (GUI)	executes "Level 2" scenario for user	GUI (display)	Runs "Level 2" scenario for user
Play	Level 3	User (GUI)	executes "Level 3" scenario for user	GUI (display)	Runs "Level 3" scenario for user
Sys Utility	Settings	User (GUI)	calls "Settings" submenu	GUI (display)	Changes from GUI main menu to "Settings" submenu
Settings	Gender	User (GUI)	toggle gender	GUI memory and GUI (display)	Sets the user's gender
Settings	Age	User (GUI)	enter script age value, or increase/decrease value via arrows	GUI memory and GUI (display)	Sets the user's age
Settings	Weight	User (GUI)	enter script weight value, or increase/decrease value via arrows	GUI memory and GUI (display)	Sets the user's weight
Settings	Height	User (GUI)	enter script height value, or increase/decrease value via arrows	GUI memory and GUI (display)	Sets the user's height
Settings	Exit	User (GUI)	makes cleanup and exit call for application	GUI (display)	Exits the application

Table 4. Software Interface Description Table

For the software architecture of the Virtual Reality System, the above interfaces in **Figure 5** are used. The Oculus Go will interact with the Oculus Go SDK to interface with the VRS GUI. This is mainly for building and running the application. Regarding the sensor data, the E4 Wristband will send data via the E4 Streaming Service, which will then be displayed on an external viewing device via a Windows application. The User Choice, Self Evaluation, Play, and Setting CSC's will interface with the VRS GUI as well.

This Virtual Reality System for Treating Eating Disorders SDD will delineate the software decisions made for the VR and Sensor System using Unity and the E4 Streaming Service, respectively. In the Data Acquisition sections, this SDD will explain the process required to collect the sensor data and display them to an external viewing device. In the User Interface sections, this SDD will outline what the user will see when using the application, as well as how they will be able to interact with it. Furthermore, the Level System sections will discuss the different scenarios (levels) that the user will be exposed to, and the process that will run in the background to determine whether or not the user passes a level. Lastly, the Self-Evaluation sections will discuss the self-evaluation interface that the user is to answer after completing a level.

2.5.2.2 Referenced Documents

1. Critical Design Review: Virtual Reality System for Treating Eating Disorders, rev (-), 12/2/2019
2. Preliminary Design Review: Virtual Reality System for Treating Eating Disorders, rev (-), 10/31/2019
3. System Requirements Document: Virtual Reality System for Treating Eating Disorders, rev (-), 10/22/2019

2.5.2.3 CSCI Design:

The E4 input shall take the form of data packets sent from the E4 Server through a TCP client. The data packets will be parsed and then added as a data point to their respective graphs. The output shall be heart rate and galvanic skin response, transmitted to an external display.

According to Empatica, the E4 extracts heart rate data at a sampling rate of 64 Hz, and galvanic skin response data at a sampling rate of 4 Hz.

The VR program shall be uploaded to the Oculus Go using Android Debugger Bridge. Unity/JavaScript shall build User Choice, Play, Settings, and Self-Evaluation CSC's. Additionally, the user's pass/fail progress for the level will be calculated as each food decision is made.

2.5.2.4 CSCI Architecture:

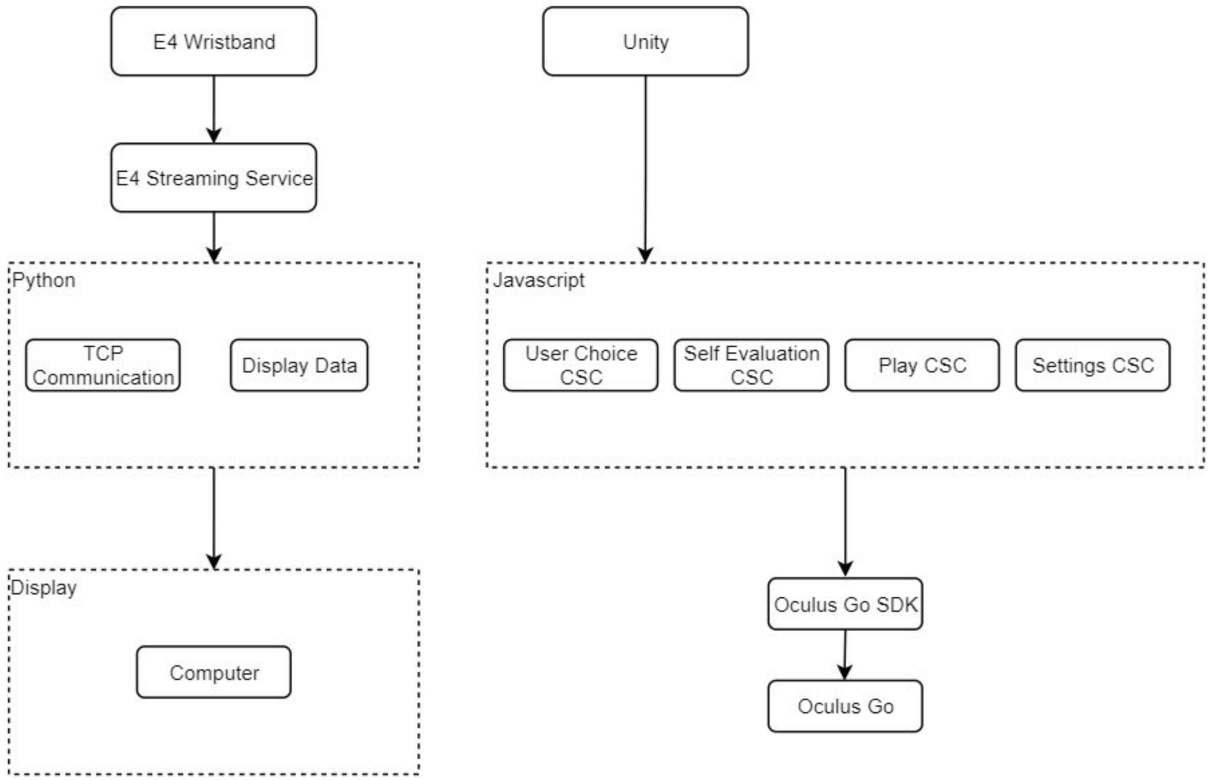


Figure 6. Software Flowchart

2.5.2.5 CSCI Detailed Designs:

2.5.2.5.1 Data Acquisition

The E4 streaming server is a Windows application that can be downloaded from Empatica's developer site. Once the streaming server has been installed, it must be configured at its first run. To connect an E4 to the server, the following steps can be followed:

1. Make sure to keep the E4s not too close to each other and the dongle. The radios can saturate when the antennas are too close, resulting in communication problems.
2. Turn on the E4 Wristband. The LED starts to blink green.
3. After a few seconds, the E4 is visible in the list, showing the device ID and, once bound, the associated TCP connections.
4. If manual BTLE is disabled, the E4 is automatically connected. Otherwise, a button is shown in the "Actions" row to allow for selective connection over BTLE.

5. While the connection is being established, the E4 LED turns light blue. Finally, the LED turns blue upon a successful connection.
6. After around 40 seconds, the LED will be turned off to save battery.

Then, to connect a TCP client to the server and start receiving data, the following steps can be followed:

1. A TCP client establishes a connection to the E4 streaming server, using the configured connection parameters.
2. The TCP client binds to a device connected over BTLE through the device_connect command.
3. The bound TCP client subscribes to a channel with the device_subscribe command.

The TCP client will now start receiving the data as data packets.

The table below outlines the possible commands that can be sent through the client to communicate with the server:

Command	Format	Description
device_discover_list		The client requests the list of Empatica E4 devices, that are in range and not connected over BTLE. The server responds with the number of discovered devices and a list of device info. If **Manual BTLE** is not set, this command is not needed since discovery and connection over BTLE are handled automatically.
device_list	device_list	The client requests the list of Empatica E4 devices that are in range and connected over BTLE. The server responds with the number of connected devices and a list of device info.
device_connect	device_connect <DEVICE_ID>	The client sends a connection request to a specific device. The E4 streaming server binds the client connected by TCP to the device connected over BTLE. The bound client needs to subscribe to channels with the

		device_subscribe command in order to start receiving data.
device_disconnect	device_disconnect	The client sends a device disconnection request. The client will be disconnected from the currently connected device and close the TCP connection to the E4 streaming server. The device will remain connected to the E4 streaming server over BTLE.
device_connect_btle	device_connect_btle <DEVICE_ID> [<TIMEOUT>]	The client sends a BTLE connection request for a specific device. The server will connect to the device over BTLE if it has been discovered and if allowed by the API key. If **Manual BTLE** is not set, this command is not needed since discovery and connection over BTLE are handled automatically.
device_disconnect_btle	device_disconnect_btle <DEVICE_ID>	The client sends a BTLE disconnection request for a specific device. The server will disconnect the device from BTLE. This command terminates active connections as well as stops reconnection attempts to devices that have temporarily lost BTLE connection.
device_subscribe	device_subscribe<STREAM> <STATUS>	To start or stop receiving data from a given stream, the client sends a data subscription request specifying the desired stream.
pause	pause <STATUS>	To temporarily suspend and resume the data streaming (without disconnecting or turning off the device), the client sends a pause requests

Table 5. TCP Message Protocol

The following stream subscriptions can be made:

- acc - 3-axis acceleration
- bvp - Blood Volume Pulse
- gsr - Galvanic Skin Response
- ibi - Interbeat Interval and Heartbeat
- tmp - Skin Temperature
- bat - Device Battery
- tag - Tag taken from the device (by pressing the button)

The code snippet below shows the code for initializing TCP Communication using C#:

```
// The port number for the remote device.
private const string ServerAddress = "127.0.0.1";
private const int ServerPort = 28000;

// ManualResetEvent instances signal completion.
private static readonly ManualResetEvent ConnectDone = new ManualResetEvent(false);
private static readonly ManualResetEvent SendDone = new ManualResetEvent(false);
private static readonly ManualResetEvent ReceiveDone = new ManualResetEvent(false);

// The response from the remote device.
private static String _response = String.Empty;

// Container for responses from the remote device.
private static ConcurrentQueue<string> ResponseQueue = new ConcurrentQueue<string>();

// Boolean initialized to false, and will be set to true by another object when data acquisition is to be terminated
private static bool terminateAcquisition = false;
private static bool clientStarted = false;

public static void StartClient()
{
    // Connect to a remote device.
    try
    {
        // Establish the remote endpoint for the socket.
        var ipHostInfo = new IPHostEntry { AddressList = new[] { IPAddress.Parse(ServerAddress) } };
        var ipAddress = ipHostInfo.AddressList[0];
        var remoteEp = new IPEndPoint(ipAddress, ServerPort);

        // Create a TCP/IP socket.
        var client = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

        // Connect to the remote endpoint.
        client.BeginConnect(remoteEp, (ConnectCallback), client);
        ConnectDone.WaitOne();

        // Obtain list of devices
        PushCommand(client, "device_list");

        // Parse response from server to obtain device ID
```

Figure 7. TCP Communication using C#

The data obtained from the server will be graphed in C3. The below code sample shows the code for adding a data point to the graphing interface from the application:

```
// Add data to relevant graph
cPoint newPt = new cPoint
{
    x = dtNow.Milliseconds,
    y = dataVal
};

if (this.InvokeRequired)
{
    this.Invoke(new MethodInvoker(delegate
    {
        DataSource ds = display.DataSources[0];
        AppendAtEnd(ds, newPt);
        ds.OnRenderYAxisLabel = RenderYLabel;
        bvpFile.WriteLine(newPt.y);
    }));
}
else
{
    DataSource ds = display.DataSources[0];
    AppendAtEnd(ds, newPt);
    ds.OnRenderYAxisLabel = RenderYLabel;
    bvpFile.WriteLine(newPt.y);
}
```

Figure 8. Creating Plot using Python’s matplotlib library

2.5.2.5.2 User Interface

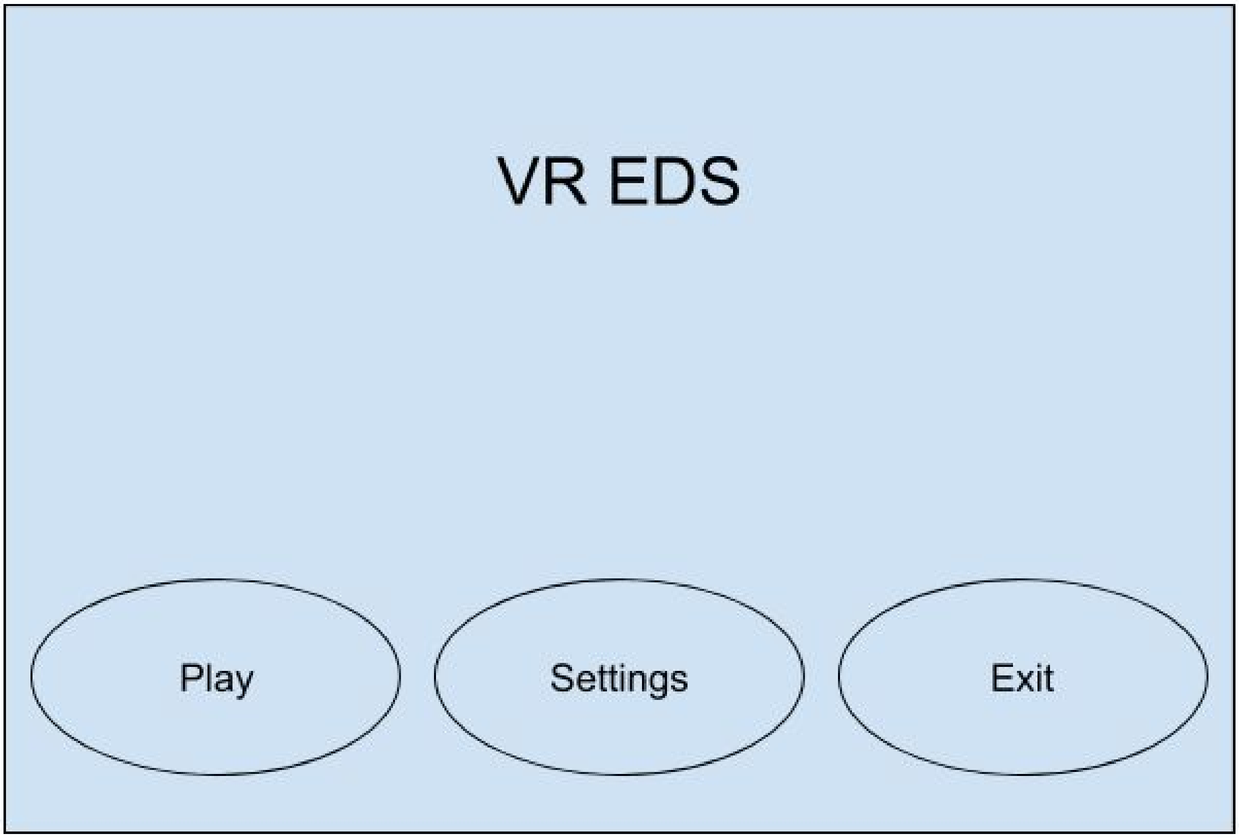


Figure 9. GUI CSCI (100400) Sample GUI Start Menu Display

This is the design concept for the main menu that the user first sees when running the application. This menu has the title of the application at the top, and three buttons at the bottom: a Play button, a Settings button, and an Exit button. Clicking the Play and Settings buttons will take the user to the Play and Settings submenu, respectively. Clicking the Exit button will cause the execution of a cleanup and exit call for the application. For the rest of these pages, the design concepts will be talked about. The GUI interface is still to be implemented in the project.

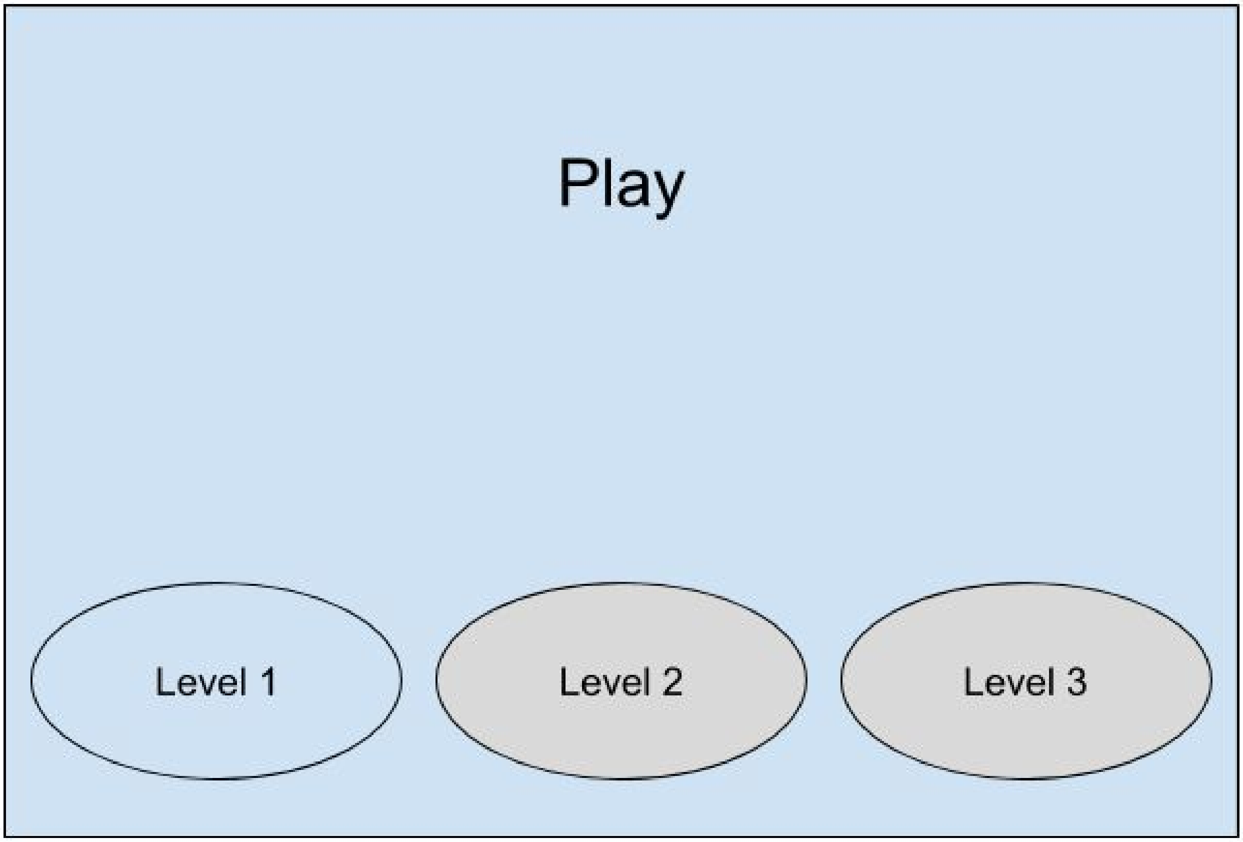


Figure 10. Play Submenu (100420)

Regarding the Play submenu, clicking a level button will start a scenario for the user to play as it relates to the narrative. Different names will be used for the buttons in the final design, but for easy understanding, the preliminary design above just named the buttons as Levels 1/2/3 to make it easy to see how the scenarios will be written out. Level 1 corresponds to the “Eating Alone” scenario, Level 2 to the “Going Out to Eat” scenario, and Level 3 to the “Going to a Potluck” scenario.

A design decision was made that the buttons would be at the same spots as the main menu buttons, and start where the Play button is located in the main menu to make it easy for the user to click on Level 1 right away. Levels 2 and 3 must be locked for the user until they pass Level 1, so those buttons will be grayed out and inaccessible until the user has unlocked it by successful level completion.

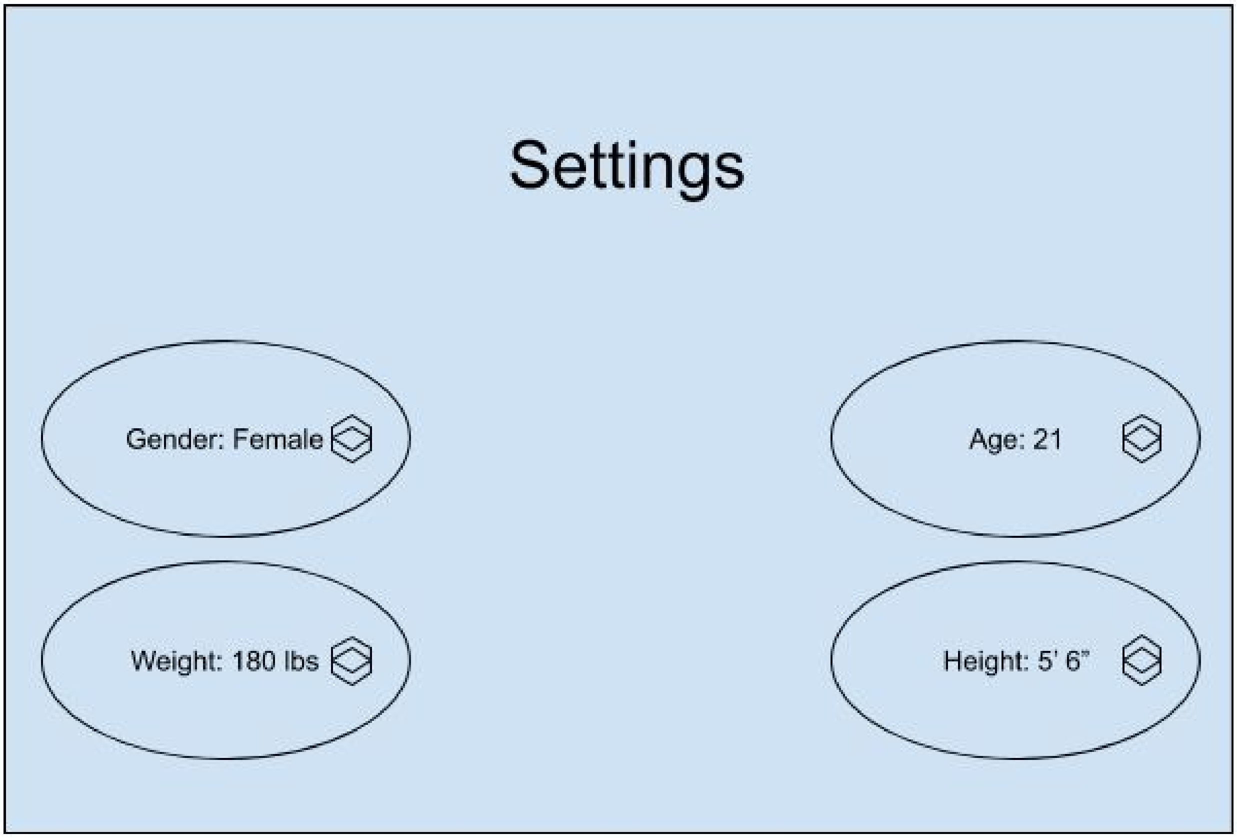


Figure 11. Settings Submenu (100430)

Figure 11 shows the preliminary design for the Settings submenu. At the start of each session, the user will be able to update their parameters if desired. Regardless of whether or not they choose to update their parameters, if they decide to edit them at a later point during the session, they can go to the Settings submenu to do so. When the user enters the submenu, they will be able to edit their parameters by using the up/down arrows or clicking on the circle and entering the desired value.

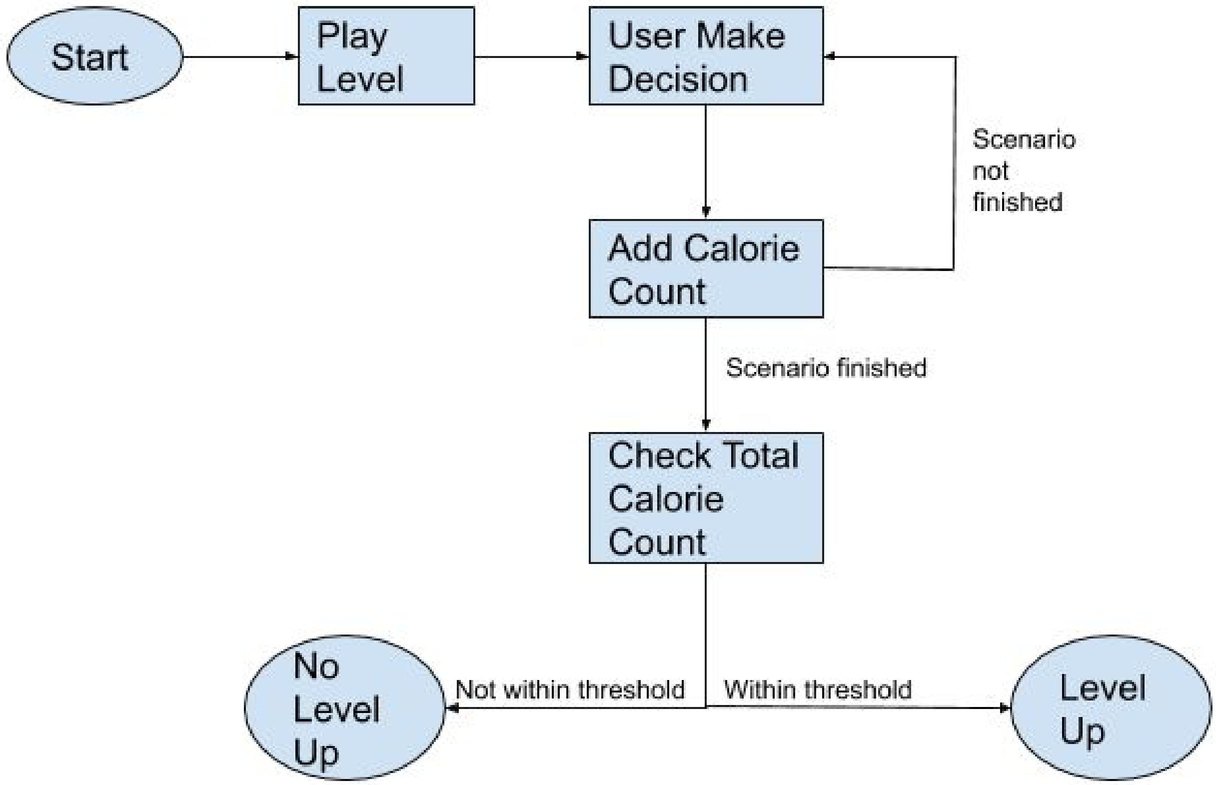


Figure 12. User Choice Routine

Figure 12 shows the routine that the application will follow when the user plays a level. Essentially, the user will make food decisions until the scenario has finished, and if they are within the threshold of their ideal total calorie intake, then they pass the level. Otherwise, they fail the level and cannot advance. The design of the level system is explained in more detail in the section below.

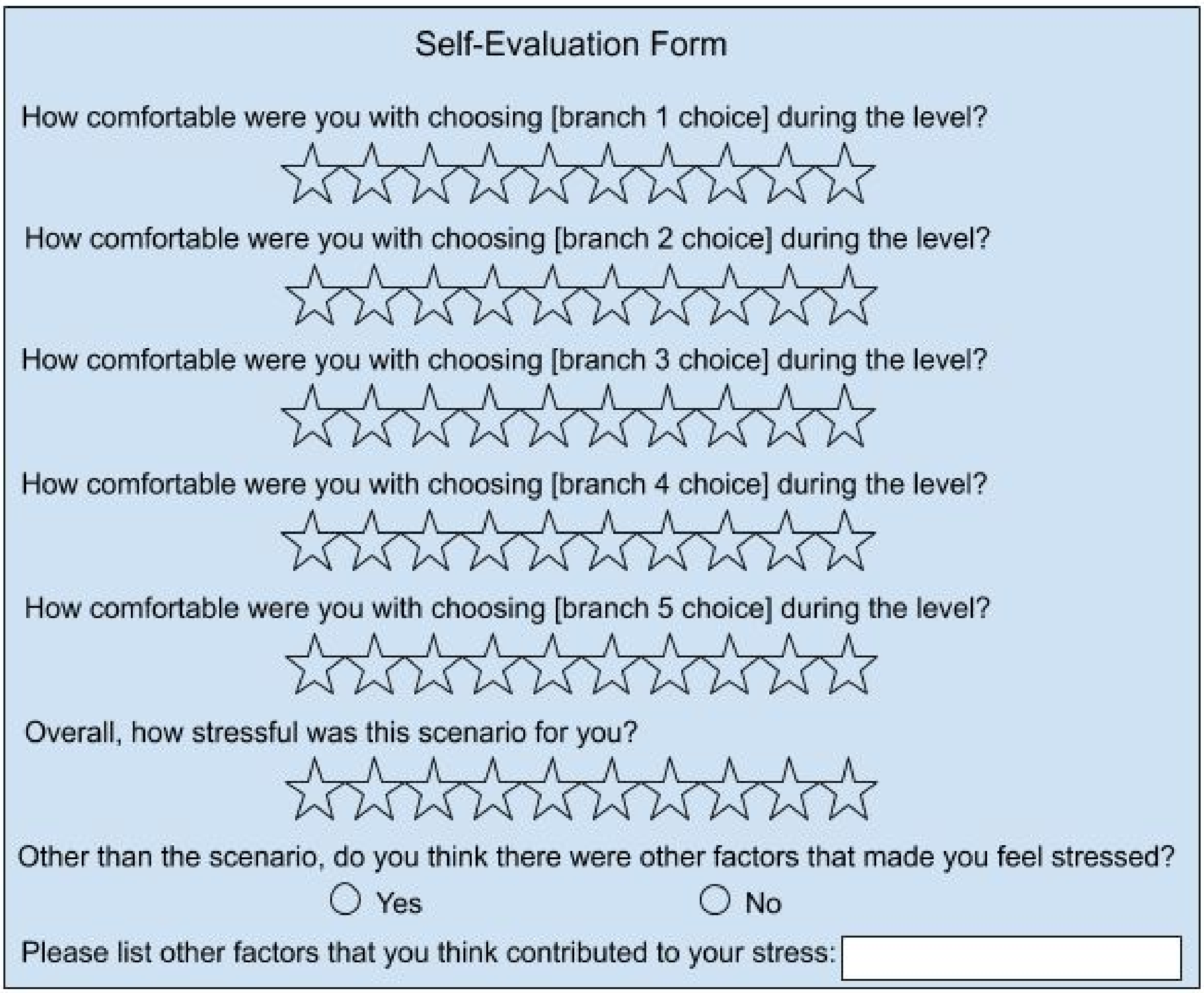
2.5.2.5.3 Level System

System Requirement 4.3.4 states that the system shall require the user to be within +/- 15% of their ideal calorie intake before unlocking the next difficulty level. The value of the ideal calorie intake depends upon the user's age, weight, height, and gender. This value is called the BMR value, and the equation to calculate it is as follows: BMR (kcal / day) = 10 * weight (kg) + 6.25 * height (cm) - 5 * age (y) + s (kcal / day), where s is +5 for males and -161 for females. At the start of the user's first session with the application, they will be requested to input the relevant info to calculate the BMR value. And at each subsequent sign-in, they will be prompted to update their information.

In order for the user to progress to the next level, the sum of their caloric count must fall within a +/- 15% threshold of 50% of the BMR value calculated. The 50% comes from the fact that the BMR value is an ideal calorie intake for a whole day, which generally means three meals. The user will only be eating one full meal per level, but each meal will either be lunch or dinner, which are generally large meals. They will also have the opportunity to eat appetizers and desserts, which will lead to greater calorie intake. Thus, it was decided that reaching a +/- 15% threshold of 50% of the user's BMR value would be the best way of deciding whether the user passes or fails a level.

2.5.2.5.4 Self-Evaluation

As per System Requirement 4.3.5, the EDVRS system shall provide the user with a self-evaluation upon completion of a level. **Figure 13** below shows the preliminary design of the form.


Self-Evaluation Form

How comfortable were you with choosing [branch 1 choice] during the level?

How comfortable were you with choosing [branch 2 choice] during the level?

How comfortable were you with choosing [branch 3 choice] during the level?

How comfortable were you with choosing [branch 4 choice] during the level?

How comfortable were you with choosing [branch 5 choice] during the level?

Overall, how stressful was this scenario for you?

Other than the scenario, do you think there were other factors that made you feel stressed?

Yes No

Please list other factors that you think contributed to your stress:

Figure 13. Self-Evaluation Form

The intended design for the self-evaluation form is that when a session starts, a .txt file will be created inside a directory for Self-Evaluation choices. The current plan is to have the file be inside the "Program Files" directory: "C:\Program Files\EDVRS\SelfEvaluation\". Each new file will be named using a timestamp of the time at which the session started: "Session_[Timestamp].txt". Inside this file, each new section will be denoted by a new section mark- "=="- or something similar. Then, the level number for the current self-evaluation will be written under the new section mark. Under the level number, a string indicating whether the user passed or failed the level will be written. Lastly, each answer to the self-evaluation questions will be written in a new line for each one. The format for writing the answers will be the following: [Question]: [Number ranking] or [Yes/No] or [String input].

2.5.2.6 Requirements Traceability: (system requirements to CSCI to units)

Functional Requirement	System Requirement	Test Method
1. The system shall simulate scenarios in virtual reality.	1.1 The system shall have three scenarios	D
2. The system shall be created using the Unity engine.	2.1 The system shall be created using Unity version 2019.2.7.	I
3. The system shall operate on the Oculus Go Virtual Reality platform.	3.1 The system shall operate on the first generation of the Oculus Go Virtual Reality platform.	I
4. The system shall contain a multi-choice, branching narrative exposing the user to different scenarios involving food.	4.1 The system shall allow the user to unlock three different levels.	D
	4.2 The system shall allow the user to unlock three different levels. To confirm this requirement, the system will be demonstrated by going through all possible levels.	D
5. The system shall allow the user to level up and unlock more difficult scenarios.	5.1 The system shall require the user to be within +/- 15% of their ideal calorie intake before unlocking the next difficulty level.	T/A/D

6. The system shall allow the user to provide a self-evaluation.	6.1 The system shall provide the user with a self-evaluation upon completion of a level.	D
7. The system shall collect physiological data about the user.	7.1 The heart rate sensor shall sample at a rate of at least one sample every five seconds to capture trends throughout the simulation, which is supported by our research that heart rate will change gradually rather than instantly.	T/A
	7.2 The galvanic skin response sensor shall sample at a rate of at least one sample every five seconds to capture trends throughout the simulation, which is supported by our research that galvanic skin response will change gradually rather than instantly.	T/A
	7.3 The computer shall store data from the heart rate sensor.	I
	7.4 The computer shall store data from the galvanic skin response sensor	I
8. The system shall feed the collected physiological data to an external computer to be saved for future reference.	8.1 The system shall provide the user with a self-evaluation upon completion of a level.	T
	8.2 The external computer shall operate on the Windows 10 OS.	I
	8.3 The external computer shall upload the virtual reality simulation to the Oculus Go using a micro-usb cable.	I

Table 6. Functional Requirement Traceability to CSCI

2.5.3 Acceptance Test Procedure (ATP)

2.5.3.1 Introduction

2.5.3.5.1.a. This procedure outlines the acceptance tests for the accuracy and sampling rate of our heart rate sensor. This test will verify that our heart rate sensor can record data within a +/- 10% error rate of a gold standard ECG and that the sampling rate collects at least one sample every five seconds. We will be performing this test in a lab that has access to a gold standard ECG.

2.5.3.5.2.a. This procedure outlines the acceptance test for the sampling rate of our galvanic skin response sensor. This test will verify that our galvanic skin response sensor can record data at a sampling rate of at least one sample every five seconds. We will be performing this test in a lab that has access to a medical-grade GSR sensor.

2.5.3.5.3.a. This procedure outlines the acceptance test for the frame rate of the Oculus Go. This test will verify that our Oculus Go can run simulations at a minimum rate of 60 frames per second. We will be testing this using the OVR metric tool which collects information about the frame rate of the application and stores it in a csv file.

2.5.3.5.4.a. This procedure outlines the acceptance test for the level-up components of our simulation. This test will verify that our application will allow or prevent the user from advancing through the simulation depending on whether or not they fall within the calorie threshold. If the user is within the +/- 15% threshold from their ideal calorie intake, they unlock the next level; otherwise, the next level remains locked and the user cannot advance. The ideal calorie intake will be a predefined number to minimize the variability of the test and enhance the accuracy of the test. We will be performing this test by ensuring the user may only advance if they fall within this threshold.

2.5.3.5.5.a. This procedure outlines the acceptance test for the Wireless Communication of our system. This test will verify that our Oculus Go, computer, and E4 wristband are all able to wirelessly communicate with one another and connect to wireless communication frequencies (Wi-Fi). This test will be performed by connecting all of the components together and monitoring the connection of each device to Wi-Fi. Note: The Oculus Go and E4 sensor will both individually communicate with the computer, but they do not communicate with one another.

2.5.3.2 Revisions

2020 Jan 19 - rev(-)
2020 Apr 17 - rev(-)

2.5.3.3 Contents

Section	Reference	Description
Cover	2.5.3	This document was written and compiled on January 19, 2020. Team 19085, Acceptance Test Procedure Document for the EDVRS.

Introduction	2.5.3.1	Introduction to Acceptance Test Plans
Revisions	2.5.3.2	2020 Jan 19 - rev(-), 2020 Apr 17 - rev(-)
Contents	2.5.3.3	Lab Tests Independent Sensor, VR, and Computer Assembly Tests
Subprocedures		
Lab Tests		
1.0	2.5.3.4.1	Procedure number. 2.5.3.4.1.1 Title of procedure. Accuracy of Heart Rate Sensor Test
2.0	2.5.3.4.1.a.	Introduction
3.0	2.5.3.4.1.b.	Referenced documents
4.0	2.5.3.4.1.c.	Required test equipment
5.0	2.5.3.4.1.d.	Table of tests
6.0	2.5.3.4.1.e.	Step-by-step procedure
7.0	2.5.3.4.1.f.	Support Requirements
8.0	2.5.3.4.1	Procedure number. 2.5.3.4.1.2 Title of procedure. Heart Rate Sensor Sampling Test
9.0	2.5.3.4.1.a.	Introduction
10.0	2.5.3.4.1.b.	Referenced documents
11.0	2.5.3.4.1.c.	Required test equipment
12.0	2.5.3.4.1.d.	Table of tests
13.0	2.5.3.4.1.e.	Step-by-step procedure
14.0	2.5.3.4.1.f.	Support Requirements
15.0	2.5.3.4.2	Procedure number. 2.5.3.4.2.1 Title of procedure. Galvanic Skin Response Sensor Sampling Test
16.0	2.5.3.4.2.a.	Introduction
17.0	2.5.3.4.2.b.	Referenced documents
18.0	2.5.3.4.2.c.	Required test equipment
19.0	2.5.3.4.2.d.	Table of tests
20.0	2.5.3.4.2.e.	Step-by-step procedure
21.0	2.5.3.4.2.f.	Support Requirements
Independent Sensory, VR, and Computer Assembly Tests		

22.0	2.5.3.4.3	Procedure number. 2.5.3.4.3.1 Title of procedure. Oculus Go Frame Rate Data Sheet
23.0	2.5.3.4.3.a.	Introduction
24.0	2.5.3.4.3.b.	Referenced documents
25.0	2.5.3.4.3.c.	Required test equipment
26.0	2.5.3.4.3.d.	Table of tests
27.0	2.5.3.4.3.e.	Step-by-step procedure
28.0	2.5.3.4.3.f.	Support Requirements
29.0	2.5.3.4.4	Procedure number. 2.5.3.4.4.1 Title of procedure. Calorie Counting Level-Up Test
30.0	2.5.3.4.4.a.	Introduction
31.0	2.5.3.4.4.b.	Referenced documents
32.0	2.5.3.4.4.c.	Required test equipment
33.0	2.5.3.4.4.d.	Table of tests
34.0	2.5.3.4.4.e.	Step-by-step procedure
35.0	2.5.3.4.4.f.	Support Requirements
36.0	2.5.3.4.5	Procedure number. 2.5.3.4.5.1 Title of procedure. Wireless Communication Test (E4 wristband to computer)
37.0	2.5.3.4.5.a.	Introduction
38.0	2.5.3.4.5.b.	Referenced documents
39.0	2.5.3.4.5.c.	Required test equipment
40.0	2.5.3.4.5.d.	Table of tests
41.0	2.5.3.4.5.e.	Step-by-step procedure
42.0	2.5.3.4.5.f.	Support Requirements
43.0	2.5.3.4.5	Procedure number. 2.5.3.4.5.2 Title of procedure. Wireless Communication Test (Oculus Go to computer)
44.0	2.5.3.4.5.a.	Introduction
45.0	2.5.3.4.5.b.	Referenced documents
46.0	2.5.3.4.5.c.	Required test equipment
47.0	2.5.3.4.5.d.	Table of tests
48.0	2.5.3.4.5.e.	Step-by-step procedure

49.0	2.5.3.4.5.f.	Support Requirements
	Data Sheets	
50.0	2.5.3.4.1.1	Accuracy of Heart Rate Sensor Data Sheet
51.0	2.5.3.4.1.2	Sampling of Heart Rate Sensor Data Sheet
52.0	2.5.3.4.2.1	Sampling of Galvanic Skin Response Sensor Data Sheet
52.0	2.5.3.4.3.1	Oculus Go Frame Rate Data Sheet
54.0	2.5.3.4.4.1	Calorie Counting Level-Up Data Sheet
55.0	2.5.3.4.5.1	Wireless Communication Data Sheet (E4 Wristband to computer)
56.0	2.5.3.4.5.2	Wireless Communication Data Sheet (Oculus Go to computer)

2.5.3.4 Test Plan

2.5.3.4.1 Verifying Accuracy and Sample Rate of Heart Rate Sensor

2.5.3.4.1.b. Referenced Documents: EDVRS System Requirements Document, 11/24/2019, rev(-)

2.5.3.4.1.c. Required test equipment: The list of required test equipment includes:

Description	**Model Number**	**Accuracy**
E4 Wristband	Rev. 2	+/- 10%
Computer	ASUS F510U	N/A
Test Subject	N/A	N/A
Biostamp ECG	N/A	N/A
ECG Operator	N/A	N/A

2.5.3.4.1.d. Table of Tests

Test #	Test	Requirement
2.5.3.4.1.1	Accuracy	+/- 10% error rate
2.5.3.4.1.2	Sample Rate	At least one sample every five seconds

2.5.3.4.1.e. Step-by-Step Procedure

a. Turn on ECG and E4 wristband (our heart rate sensor).
b. ECG operator will set up the test subject with the gold standard ECG.
c. The subject will then put on the E4 wristband.
d. First, we will record the heart rate of the patient while they are sitting still. This will be used as the resting baseline for the remainder of the test.
e. Data will be collected from both the ECG and the E4 during this 30 second time period.
f. Next, we will have the subject jog in place for 30 seconds while recording their heart rate.
g. Allow the subject's heart rate to return to resting baseline.
h. Finally, we will have the subject do jumping jacks in place for 30 seconds while recording their heart rate.
i. For each experiment, we will compare the data from the gold standard ECG to the data from our E4 sensor.
j. For the accuracy test, we will calculate the error rate between our sensor and the gold standard ECG.
k. All of the tests should be within a +/- 10% error rate in order for our test to be considered successful (mark pass/fail on the data sheet)
l. For the sampling rate, we should see at least six samples recorded in the span of 30 seconds, from our E4 wristband (mark pass/fail on the data sheet)

2.5.3.4.1.f. Support Requirements - Biostamp operator

2.5.3.4.2 Verifying Galvanic Skin Response Sensor Sampling Test

2.5.3.4.2.b. Referenced Documents: EDVRS System Requirements Document, 11/24/2019, rev(-)

2.5.3.4.2.c. Required test equipment: The list of required test equipment includes:

Description	**Model Number**	**Accuracy**
E4 Wristband	Rev. 2	+/- 10%
Computer	ASUS F510U	N/A
Test Subject	N/A	N/A

Medical Grade GSR	Grove - GSR_Sensor V1.2	N/A

2.5.3.4.2.d. Table of Tests

Test #	Test	Requirement
2.5.3.4.2.1	Sample Rate	At least one sample every five seconds

2.5.3.4.2.e. Step-by-Step Procedure

a. Turn on GSR and E4 wristband (our galvanic skin response sensor).
b. Set up the test subject with the medical grade GSR sensor.
c. The subject will then put on the E4 wristband.
d. First, we will record the GSR of the patient while they are sitting still. This will be used as the resting baseline for the remainder of the test.
e. Data will be collected from both the gold standard GSR and the E4 during this 30 second time period.
f. Next, we will have the subject jog in place for 30 seconds while recording GSR.
g. Allow the subject’s galvanic skin response to return to baseline.
h. Finally, we will have the subject do jumping jacks in place for 30 seconds while recording their GSR.
i. For each experiment, we will compare the data from the gold standard GSR to the data from our E4 sensor.
j. For the sampling rate, we should see at least six samples recorded in the span of 30 seconds, from our E4 wristband (mark pass/fail on the data sheet).

2.5.3.4.2.f. Support Requirements - N/A

2.5.3.4.3 Verifying Frame Rate of Oculus Go

2.5.3.4.3.b. Referenced Documents: EDVRS System Requirements Document, 11/24/2019, rev(-)

2.5.3.4.3.c. Required test equipment: The list of required test equipment includes:

Description	**Model Number**	**Accuracy**

Oculus Go	All-In-One VR (64 GB)	N/A
Computer	ASUS F510U	N/A
OVR Metric Tool	1.4	N/A

2.5.3.4.3.d. Table of Tests

Test #	Test	Requirement
2.5.3.4.3.1	Frame Rate	> 60 Frames per Second

2.5.3.4.3.e. Step-by-Step Procedure

a. Install the OVR Metrics tool to the headset using the Android Debugger Bridge with the following command: “adb shell am start omms://app”.
b. Enable Report Mode Usage to allow the OVR tool to track frame rate with the following command in Android Debugger Bridge: “abs shell setprop debug.oculus.omms.enableCSV”.
c. Run the Virtual Reality System for Treating Eating Disorders program on the Oculus Go.
d. After completing a session of the virtual reality simulation, open the OVR Metrics tool in the headset and select the drop down menu in the upper right corner. Select View Recorded Sessions to see a performance graph of the previously recorded session. This graph will include frame rate.
e. This graph must display a frame rate of 60 frames or greater per second to pass (mark pass/fail on the data sheet).

2.5.3.4.3.f. Support Requirements - N/A

2.5.3.4.4 Verifying Level-up

2.5.3.4.4.b. Referenced Documents: EDVRS System Requirements Document, 11/24/2019, rev(-)

2.5.3.4.4.c. Required test equipment: The list of required test equipment includes:

Description	**Model Number**	**Accuracy**
Oculus Go	All-In-One VR (64 GB)	N/A
Computer	ASUS F510U	N/A

2.5.3.4.4.d. Table of Tests

Test #	Test	Requirement
2.5.3.4.4.1	Level-Up	+/- 15% of ideal caloric intake

2.5.3.4.4.e. Step-by-Step Procedure

a. Input ideal caloric intake (pre-determined) and compare it to recorded caloric intake according to simulation. Determine whether this falls within the +/- 15% threshold.
b. If yes, check to see that the user can advance.
c. If no, check to see that the user cannot advance.
d. According to level-up criteria as defined in 5.3.5.4.a., ensure appropriate outcome (mark pass/fail on the data sheet).

2.5.3.4.4.f. Support Requirements - N/A

2.5.3.4.5 Verifying Wireless Communication

2.5.3.4.5.b. Referenced Documents: EDVRS System Requirements Document, 11/24/2019, rev(-)

2.5.3.4.5.c. Required test equipment: The list of required test equipment includes:

Description	**Model Number**	**Accuracy**
Oculus Go	All-In-One (64GB)	N/A
Computer	ASUS F510U	N/A
E4 Wristband	Rev. 2	N/A

2.5.3.4.5.d. Table of Tests

Test #	Test	Requirement
2.5.3.4.5.1	Wireless Communication [E4 Wristband to Computer]	Devices adhere to IEEE 802.11 wireless communication standards.
2.5.3.4.5.2	Wireless Communication [Oculus Go to Computer]	Devices adhere to IEEE 802.11 wireless communication standards.

2.5.3.4.5.e. Step-by-Step Procedure

a. Power computer, E4 wristband, and Oculus Go.
b. Open the E4 streaming server and connect to the E4 wristband.
c. Send the client a request to receive data packets.
d. Ensure that the windows application receives data, confirming connection between computer and E4 wristband (mark pass/fail on the data sheet).
e. Connect Oculus Go to the computer via a micro-usb.
f. Open unity with the simulation files loaded on the system.
g. Take note of output file location.
h. Build file to the Oculus Go using Android Debugger Bridge.
i. Locate Application in the Oculus Go under the other applications folder (mark pass/fail on data sheet).

2.5.3.4.5.f. Support Requirements - N/A

2.5.3.5 Acceptance Test Data Sheets for Test 2.5.3.4.1.1-2.5.3.4.5.2

2.5.3.4.1.1 Data Sheet

2.5.3.4.1.1 Accuracy of Heart Rate Sensor Data Sheet			
Referenced ATP Paragraph 2.5.3.4.1.a			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Accuracy of Heart Rate Sensor test			
Unit Under Test (UUT): Name: Heart Rate Sensor Assembly Part number: 100200 Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: • E4 Heart Rate Sample • Biostamp Sample	Requirement (SR, with Tolerances):	Test Equipment Error:	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): $\frac{E4\ sample}{Biostamp\ Sample}$			
Signatures: Tester__________________________Customer________________________________			

2.5.3.5.1.2 Data Sheet

<table>
<tr><td colspan="4">2.5.3.4.1.2 Sampling of Heart Rate Sensor Data Sheet</td></tr>
<tr><td colspan="4">Referenced ATP Paragraph 2.5.3.4.1.a</td></tr>
<tr><td colspan="4">Analysis of Referenced (for verification by T/A): none</td></tr>
<tr><td colspan="4">Name of Test: Heart Rate Sensor Sampling test</td></tr>
<tr><td colspan="4">Unit Under Test (UUT):
Name: Heart Rate Sensor Assembly
Part number: 100200
Serial number: n/a</td></tr>
<tr><td colspan="2">Results (Pass/Fail):</td><td colspan="2">Date of Test:</td></tr>
<tr><td>Recording of Test Measurement:
• Number of E4 samples recorded in 30 seconds</td><td>Requirement (SR, with Tolerances):</td><td>Test Equipment Error</td><td>Adjustment Test Limit:</td></tr>
<tr><td colspan="4">Computations, (Include Analyses Results, if any): $\frac{\textit{Number of E4 samples recorded in 30 seconds}}{\textit{30 seconds}}$</td></tr>
<tr><td colspan="4">Signatures:

Tester__________________________________Customer________________________</td></tr>
</table>

<table>
<tr><td colspan="5">Computations - Heart Rate Accuracy</td></tr>
<tr><td colspan="5">Record the Biostamp sample and calculate the acceptable range (+/-10%). All recorded values for the E4 must fall within the range to pass.</td></tr>
<tr><td>Biostamp value</td><td>Min. values:</td><td>Max. values:</td><td>E4 value</td><td>Within range?</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Computations - Heart Rate Sampling
Record E4 samples for 30 seconds. There must be a minimum of six samples recorded within this

window to pass. (one sample/five seconds)	
Number of samples	Pass?

2.5.3.4.2.1 Data Sheet

2.5.3.4.2.1 Sampling of Galvanic Skin Response Sensor Data Sheet			
Referenced ATP Paragraph 2.5.3.4.1.a.			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Galvanic Skin Response Sensor Sampling test			
Unit Under Test (UUT): Name: Galvanic Skin Response Sensor Assembly Part number: 100200 Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: • Number of E4 samples recorded in 30 seconds	Requirement (SR, with Tolerances):	Test Equipment Error	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): $\frac{\text{Number of E4 samples recorded in 30 seconds}}{\text{30 seconds}}$			
Signatures: Tester_________________________________Customer___________________________			

Computations - GSR Sampling		
Record E4 samples for 30 seconds. There must be a minimum of six samples recorded within this window to pass. (1 sample/5 seconds)		
Test Data		Pass?
Time (sec)	Data Value	

2.5.3.4.3.1 Data Sheet

2.5.3.4.3.1 Oculus Go Frame Rate Data Sheet			
Referenced ATP Paragraph 2.5.3.4.3.a.			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Oculus Go Frame Rate test			
Unit Under Test (UUT): Name: VR Assembly Part number: 100510 Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: • Frame rate per second	Requirement (SR, with Tolerances):	Test Equipment Error	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): N/A			
Signatures: Tester_________________________________Customer__________________________			

Computations - Oculus Go Frame Rate		
Record Oculus Go Frames Per Second during simulation. Take the overall average, it should be above 60 FPS to pass.		
Sample Data		Pass/Fail
Time Stamp	Average Frame	

	Rate	

2.5.3.4.4.1 Data Sheet

2.5.3.4.4.1 Calorie Counting Level-Up Data Sheet			
Referenced ATP Paragraph 2.5.3.4.4.a.			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Level-Up Test			
Unit Under Test (UUT): Name: VRS GUI/Display Assembly Part number: 100400 Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: • Calorie count total	Requirement (SR, with Tolerances):	Test Equipment Error	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): $\frac{\text{calories consumed}}{\text{ideal calorie intake}}$			
Signatures: Tester__________________________________Customer_________________________			

Computations - Caloric Intake			
Record the ideal and actual caloric intake from simulation and determine if the actual falls within the threshold (+/- 15%) to level-up			
Ideal Caloric Intake (pre-determined)	Actual Caloric	Within threshold?	Correct Subsequent

	Intake		Action?

2.5.3.4.5.1 Data Sheet

2.5.3.4.5.1 Wireless Communication Data Sheet [E4 wristband to computer]			
Referenced ATP Paragraph 2.5.3.4.5.a			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Wireless Communication test			
Unit Under Test (UUT): Name: Sensor, Computer, and VR Assembly Part number: 100200, 100300, 100510 Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: N/A	Requirement (SR, with Tolerances):	Test Equipment Error:	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): N/A			
Signatures: Tester_______________________________Customer______________________			

2.5.3.4.5.2 Data Sheet

2.5.3.4.5.2 Wireless Communication Data Sheet [Oculus Go to computer]
Referenced ATP Paragraph 2.5.3.4.5.a
Analysis of Referenced (for verification by T/A): none
Name of Test: Wireless Communication test
Unit Under Test (UUT): Name: Sensor, Computer, and VR Assembly Part number: 100200, 100300, 100510

Serial number: n/a			
Results (Pass/Fail):		Date of Test:	
Recording of Test Measurement: N/A	Requirement (SR, with Tolerances):	Test Equipment Error:	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): N/A			
Signatures: Tester________________________________Customer________________________			

3.0 Verification Results

We checked to see that our system requirements were satisfied. Our tests and analyses have been verified as discussed throughout this section. Our requirements verified by demonstration and inspection such as those in the performance requirements (1.3-1.9), response time (2.3 and 2.4), and the simulation (3.1-3.5) have also been fulfilled as defined.

3.1 Analyses & Models

Requirement Title	System Requirement	Model	Analysis Prediction	Margin
3.1.1 Frame Rate	> 60 FPS	Geometric	72 FPS	12 FPS
3.1.2 Heart Rate Sample	> 0.2 Hz	Geometric	64 Hz	63.8 Hz
3.1.3 GSR Sample	> 0.2 Hz	Geometric	4 Hz	3.8 Hz
3.1.4 GSR Accuracy	According to Empatica E4 Technical Specifications Document	Geometric	Within medical grade standard	N/A
3.1.5 Calorie Counting and Level-Up	User falls within +/- 15% of their ideal calorie intake (IDC) to level-up	Geometric	Ideal calorie intake +/-15%	IDC -14% <IDC< IDC +14%

Table 7. Table of Models

3.1.1 Frame Rate

The Oculus Go can operate at a maximum of 72 frames per second, and the Oculus Store requires a minimum of 60 frames per second to distribute an application through them. Although the margin is not very large, our simulation primarily consists of displaying filmed footage which does not require rendering of complex models so we expect to be within the 12 FPS margin.

3.1.2 Heart Rate Sample Rate

The E4 wristband can sample heart rate at a speed of 64 Hz. However, we will be taking samples every 0.2 Hz due to our knowledge of how the sympathetic nervous system does not change instantaneously. These recordings will be taken to monitor stress, which is regulated by the sympathetic nervous system, and will not show changes at the speed at which the wristband can sample. A 0.2 Hz sampling rate will be sufficient in showing any notable trends.

3.1.3 Galvanic Skin Response Sample Rate

For the GSR sampling rate, the E4 wristband samples at a rate of 4Hz. Changes in galvanic skin response also occur with an increase in stress and are regulated by the sympathetic nervous system. They will not change instantaneously so we will record GSR changes at a sampling rate of 0.2 Hz which will be sufficient in showing any notable trends.

3.1.4 Galvanic Skin Response Accuracy

For GSR accuracy the Empatica E4 Technical Specifications Document specifies that the accuracy is sufficient for a medical-grade recording device. The recorded data will be used to better understand the correlation between stress and eating disorders. Therefore, according to the specifications, the E4 wristband will be sufficient in showing any notable trends.

3.1.5 Calorie Counting and Level-Up

To level up, the user must be able to choose a food combination for their main meal that will allow them to maintain their weight. To gain weight, a person must eat 20% more than their ideal daily intake, and to lose weight, they must eat 20% less. For this reason, we chose our threshold to be within +/-15% of their ideal intake; to insure the user is practicing maintaining weight. Furthermore, the user will be experiencing eating only one meal in the simulation which is the main meal, and for this, the system will only be considering 50% of the ideal daily intake.

Therefore, when the user chooses food combinations, they should be within +/-15% of half of the ideal daily intake for their body.

Moreover, our system will apply a ratio of (user's BMI/ 15.6) to the calorie ranges in the simulation to ensure that everyone is able to pass all the levels. Because our system has predefined food with specific calories which makes it would be limiting for people with varying BMIs to use without scaling the calories. The calorie ranges in our system are perfectly suitable for someone who consumes 1200 daily and with a BMI of 15.6 which is low and considered underweight (BMI<18). This is an appropriate amount of calorie to the simulation since we are targeting Anorexic and Bulimic patients. However, a person with much higher or lower BMI, would not be able to pass any of the levels without scaling the calorie ranges because the calories would be lower or higher than they need to pass.

4.0 Acceptance Test Results

4.2.5.3.5 Acceptance Test Data Sheets for Test 4.2.5.3.4.1.1-4.2.5.3.4.5.2

4.2.5.3.4.1.1 Data Sheet

4.2.5.3.4.1.1 Accuracy of Heart Rate Sensor Data Sheet			
Referenced ATP Paragraph 2.5.3.4.1.a			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Accuracy of Heart Rate Sensor test			
Unit Under Test (UUT): Name: Heart Rate Sensor Assembly Part number: 100200 Serial number: n/a			
Results (Pass/Fail): Pass		Date of Test:4/27/2020	
Recording of Test Measurement: • E4 Heart Rate Sample • Biostamp Sample	Requirement (SR, with Tolerances):	Test Equipment Error:	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): $\frac{E4\ sample}{Biostamp\ Sample}$			
Signatures: Tester Karm Al Hajhog Customer Unavailable due to COVID-19			

4.2.5.3.4.1.2 Data Sheet

4.2.5.3.4.1.2 Sampling of Heart Rate Sensor Data Sheet			
Referenced ATP Paragraph 2.5.3.4.1.a			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Heart Rate Sensor Sampling test			
Unit Under Test (UUT): Name: Heart Rate Sensor Assembly Part number: 100200 Serial number: n/a			
Results (Pass/Fail): Pass		Date of Test:4/27/2020	
Recording of Test Measurement: ● Number of E4 samples recorded in 30 seconds	Requirement (SR, with Tolerances):	Test Equipment Error	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): $\frac{\textit{Number of E4 samples recorded in 30 seconds}}{\textit{30 seconds}}$			
Signatures: Tester__Karm Al Hajhog___________Customer_Unavailable due to COVID-19__			

Computations - Heart Rate Accuracy				
Record the Biostamp sample and calculate the acceptable range (+/- 10%). All recorded values for the E4 must fall within the range to pass.				
Biostamp value	Min. values:	Max. values:	E4 value	Within range?
83	74.7	91.3	83.35	Yes
83	74.7	91.3	83	Yes
83	74.7	91.3	82.89	Yes
84	75.6	92.4	82.75	Yes
84	75.6	92.4	82.57	Yes
84	75.6	92.4	82.5	Yes
84	75.6	92.4	82.17	Yes
84	75.6	92.4	81.62	Yes

84	75.6	92.4	81.08	Yes
84	75.6	92.4	80.54	Yes
84	75.6	92.4	80.04	Yes
84	75.6	92.4	79.54	Yes
78	70.2	85.8	79.03	Yes
78	70.2	85.8	78.47	Yes
78	70.2	85.8	77.94	Yes
78	70.2	85.8	77.44	Yes
78	70.2	85.8	76.94	Yes
78	70.2	85.8	76.53	Yes
78	70.2	85.8	76.17	Yes
75	67.5	82.5	75.83	Yes
75	67.5	82.5	75.51	Yes
75	67.5	82.5	75.21	Yes
75	67.5	82.5	74.97	Yes
75	67.5	82.5	74.8	Yes
75	67.5	82.5	74.63	Yes
75	67.5	82.5	74.52	Yes
75	67.5	82.5	74.37	Yes
75	67.5	82.5	74.27	Yes
75	67.5	82.5	74.24	Yes
75	67.5	82.5	74.13	Yes
75	67.5	82.5	74.04	Yes
72	64.8	79.2	73.96	Yes
72	64.8	79.2	73.84	Yes
72	64.8	79.2	73.68	Yes
72	64.8	79.2	73.63	Yes
72	64.8	79.2	73.56	Yes
72	64.8	79.2	73.53	Yes

72	64.8	79.2	73.5	Yes
72	64.8	79.2	73.49	Yes
72	64.8	79.2	73.59	Yes
72	64.8	79.2	73.63	Yes
72	64.8	79.2	73.64	Yes

Computations - Heart Rate Sampling	
Record E4 samples for 30 seconds. There must be a minimum of six samples recorded within this window to pass. (one sample/five seconds)	
Number of samples	Pass?
8723	yes
8540	yes
8174	yes
8723	yes
8234	yes

4.2.5.3.4.2.1 Data Sheet

<table>
<tr><td colspan="4">4.2.5.3.4.2.1 Sampling of Galvanic Skin Response Sensor Data Sheet</td></tr>
<tr><td colspan="4">Referenced ATP Paragraph 2.5.3.4.1.a.</td></tr>
<tr><td colspan="4">Analysis of Referenced (for verification by T/A): none</td></tr>
<tr><td colspan="4">Name of Test: Galvanic Skin Response Sensor Sampling test</td></tr>
<tr><td colspan="4">Unit Under Test (UUT):
Name: Galvanic Skin Response Sensor Assembly
Part number: 100200
Serial number: n/a</td></tr>
<tr><td colspan="2">Results (Pass/Fail): Pass</td><td colspan="2">Date of Test: 4/17/2020</td></tr>
<tr><td>Recording of Test Measurement:
● Number of E4 samples recorded in 30 seconds</td><td>Requirement (SR, with Tolerances):</td><td>Test Equipment Error</td><td>Adjustment Test Limit:</td></tr>
<tr><td colspan="4">Computations, (Include Analyses Results, if any): $\frac{\textit{Number of E4 samples recorded in 30 seconds}}{\textit{30 seconds}}$</td></tr>
<tr><td colspan="4">Signatures:</td></tr>
</table>

Tester______Curt Bansil___ _____Customer__Unavailable due to COVID-19__

Computations - GSR Sampling					
Record E4 samples for 30 seconds. There must be a minimum of six samples recorded within this window to pass. (1 sample/5 seconds)					
Test Data		Pass?	Test Data		Pass?
Time (sec)	Data Value		Time (sec)	Data Value	
0	0	Pass	15.25	3.192667	Pass
0.25	0.836457	Pass	15.5	3.187543	Pass
0.5	1.18386	Pass	15.75	3.138867	Pass
0.75	1.622013	Pass	16	3.069696	Pass
1	2.099904	Pass	16.25	3.078662	Pass
1.25	2.588144	Pass	16.5	3.118372	Pass
1.5	2.854095	Pass	16.75	3.164486	Pass
1.75	2.84769	Pass	17	3.195229	Pass
2	2.848971	Pass	17.25	3.259276	Pass
2.25	2.845128	Pass	17.5	3.29258	Pass
2.5	2.837442	Pass	17.75	3.263119	Pass
2.75	2.836161	Pass	18	3.282333	Pass
3	2.832319	Pass	18.25	3.288738	Pass
3.25	2.823352	Pass	18.5	3.240062	Pass
3.5	2.82079	Pass	18.75	3.25159	Pass
3.75	2.814385	Pass	19	3.211881	Pass
4	2.823352	Pass	19.25	3.188824	Pass
4.25	2.831038	Pass	19.5	3.170891	Pass
4.5	2.837442	Pass	19.75	3.155519	Pass
4.75	2.846409	Pass	20	3.128619	Pass
5	2.850252	Pass	20.25	3.087629	Pass

5.25	2.855376	Pass	20.5	3.073539	Pass
5.5	2.851533	Pass	20.75	3.081224	Pass
5.75	2.843847	Pass	21	3.103	Pass
6	2.837442	Pass	21.25	3.111967	Pass
6.25	2.828476	Pass	21.5	3.105562	Pass
6.5	2.828476	Pass	21.75	3.08891	Pass
6.75	2.819509	Pass	22	3.064572	Pass
7	2.813104	Pass	22.25	3.050482	Pass
7.25	2.805419	Pass	22.5	3.044077	Pass
7.5	2.800295	Pass	22.75	3.07482	Pass
7.75	2.801576	Pass	23	3.132462	Pass
8	2.797733	Pass	23.25	3.165767	Pass
8.25	2.807981	Pass	23.5	3.190105	Pass
8.5	2.801576	Pass	23.75	3.218286	Pass
8.75	2.788766	Pass	24	3.215724	Pass
9	2.719595	Pass	24.25	3.178576	Pass
9.25	2.729843	Pass	24.5	3.160643	Pass
9.5	2.68501	Pass	24.75	3.146553	Pass
9.75	2.670919	Pass	25	3.152957	Pass
10	2.64402	Pass	25.25	3.158081	Pass
10.25	2.674762	Pass	25.5	3.143991	Pass
10.5	2.700381	Pass	25.75	3.114529	Pass
10.75	2.691415	Pass	26	3.126057	Pass
11	2.743933	Pass	26.25	3.167048	Pass
11.25	2.760586	Pass	26.5	3.188824	Pass
11.5	2.76699	Pass	26.75	3.211881	Pass
11.75	2.764429	Pass	27	3.256714	Pass
12	2.8336	Pass	27.25	3.25159	Pass
12.25	2.92839	Pass	27.5	3.242624	Pass

12.5	3.036391	Pass	27.75	3.211881	Pass
12.75	3.146553	Pass	28	3.178576	Pass
13	3.255433	Pass	28.25	3.161924	Pass
13.25	3.327166	Pass	28.5	3.132462	Pass
13.5	3.352785	Pass	28.75	3.07482	Pass
13.75	3.364313	Pass	29	3.029987	Pass
14	3.331009	Pass	29.25	3.005649	Pass
14.25	3.309233	Pass	29.5	2.98003	Pass
14.5	3.269523	Pass	29.75	2.964658	Pass
14.75	3.259276	Pass	30	2.951849	Pass
15	3.222128	Pass			

4.2.5.3.4.3.1 Data Sheet

4.2.5.3.4.3.1 Oculus Go Frame Rate Data Sheet			
Referenced ATP Paragraph 2.5.3.4.3.a.			
Analysis of Referenced (for verification by T/A): none			
Name of Test: Oculus Go Frame Rate test			
Unit Under Test (UUT): Name: VR Assembly Part number: 100510 Serial number: n/a			
Results (Pass/Fail): Pass		Date of Test:4/23/2020	
Recording of Test Measurement: • Frame rate per second	Requirement (SR, with Tolerances):	Test Equipment Error	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): N/A			
Signatures: Tester___Alexander Reyes____Customer__Unavailable due to COVID-19__			

Computations - Oculus Go Frame Rate					
Record Oculus Go Frames Per Second during simulation. Take the overall average, it should be above 60 FPS to pass.					
Sample Data		Pass/Fail	Sample Data		Pass/Fail
Time Stamp	Average Frame Rate		Time Stamp	Average Frame Rate	
1910	61	Pass	56909	60	Pass
2905	60	Pass	57913	60	Pass
3927	60	Pass	58914	60	Pass
5902	52	Fail	60041	60	Pass
6903	58	Fail	60916	57	Fail
7906	60	Pass	61915	58	Fail
8909	60	Pass	62917	61	Pass
9905	60	Pass	63913	60	Pass
10907	60	Pass	64913	60	Pass
11906	60	Pass	65914	60	Pass
12903	60	Pass	66914	60	Pass
13904	60	Pass	67916	60	Pass
14903	60	Pass	68917	60	Pass
15907	60	Pass	69914	60	Pass
16905	60	Pass	70916	61	Pass
17905	60	Pass	71913	60	Pass
18902	60	Pass	72918	60	Pass
19993	60	Pass	73917	59	Fail
20904	60	Pass	74919	60	Pass
21905	60	Pass	75920	60	Pass
22905	60	Pass	76915	60	Pass
23910	60	Pass	77917	60	Pass
24907	60	Pass	78915	61	Pass
25905	60	Pass	80050	60	Pass
26905	60	Pass	80916	60	Pass

27904	60	Pass	81919	60	Pass
28903	58	Fail	82918	60	Pass
29905	59	Fail	83932	60	Pass
30907	60	Pass	84914	55	Fail
31908	60	Pass	85914	57	Fail
32907	60	Pass	86918	60	Pass
33909	58	Fail	87916	60	Pass
34908	59	Fail	88921	60	Pass
35907	60	Pass	89917	60	Pass
36909	60	Pass	90921	60	Pass
37910	60	Pass	91917	60	Pass
38911	60	Pass	92920	60	Pass
40034	60	Pass	93917	60	Pass
40909	60	Pass	94916	60	Pass
41908	60	Pass	95920	60	Pass
42909	60	Pass	96921	60	Pass
43909	60	Pass	97918	60	Pass
44911	60	Pass	98918	60	Pass
45913	60	Pass	100044	60	Pass
46913	60	Pass	100917	60	Pass
47911	60	Pass	101917	60	Pass
48911	60	Pass	102918	60	Pass
50910	56	Fail	103920	60	Pass
51911	60	Pass	104921	60	Pass
52915	60	Pass	105922	60	Pass
53913	60	Pass	106921	60	Pass
54909	60	Pass	107443	60	Pass
55910	50	Fail	Total Average	60	Pass

4.2.5.3.4.4.1 Data Sheet

Due to the challenges with integrating our system, we have not been able to perform test **2.5.3.4.4** for calorie counting and level-up. Although we have created the equation for calorie counting and a system that allows the user to level-up based on the calorie count (section **3.1.5**), we have not yet integrated it into our code. We will be unable to finish this test until the system has been integrated in Unity, including the calorie counting and level-up function.

4.2.5.3.4.5.1 Data Sheet

<table>
<tr><td colspan="4">4.2.5.3.4.5.1 Wireless Communication Data Sheet [E4 wristband to computer]</td></tr>
<tr><td colspan="4">Referenced ATP Paragraph 2.5.3.4.5.a</td></tr>
<tr><td colspan="4">Analysis of Referenced (for verification by T/A): none</td></tr>
<tr><td colspan="4">Name of Test: Wireless Communication test</td></tr>
<tr><td colspan="4">Unit Under Test (UUT):
Name: Sensor, Computer, and VR Assembly
Part number: 100200, 100300, 100510
Serial number: n/a</td></tr>
<tr><td colspan="2">Results (Pass/Fail): Pass</td><td colspan="2">Date of Test: 3/1/2020</td></tr>
<tr><td>Recording of Test Measurement:
N/A</td><td>Requirement (SR, with Tolerances):</td><td>Test Equipment Error:</td><td>Adjustment Test Limit:</td></tr>
<tr><td colspan="4">Computations, (Include Analyses Results, if any): N/A</td></tr>
<tr><td colspan="4">Signatures:

Tester___Curt Bansil_______________Customer_Unavailable due to COVID-19__</td></tr>
</table>

4.2.5.3.4.5.2 Data Sheet

4.2.5.3.4.5.2 Wireless Communication Data Sheet [Oculus Go to computer]
Referenced ATP Paragraph 2.5.3.4.5.a
Analysis of Referenced (for verification by T/A): none
Name of Test: Wireless Communication test

Unit Under Test (UUT): Name: Sensor, Computer, and VR Assembly Part number: 100200, 100300, 100510 Serial number: n/a			
Results (Pass/Fail): Pass		Date of Test: 2/1/2020	
Recording of Test Measurement: N/A	Requirement (SR, with Tolerances):	Test Equipment Error:	Adjustment Test Limit:
Computations, (Include Analyses Results, if any): N/A			
Signatures: Tester__Alex Reyes____________Customer__Unavailable due to COVID-19__			

5.0 Final Budget

5.1 Itemized Budget Expense Report

Item	Amount
GoPro Stand	$12.99
Dongle Adaptor	$21.49
Shirts	$125.00
Public Union House Filming	$179.00
GoPro Fusion w/ SD cards	$368.36
GSR and HR Sensor	$1219.32
Total Spent	**$1926.16**
Total Budget	**$4000.00**

5.2 Budget Distribution

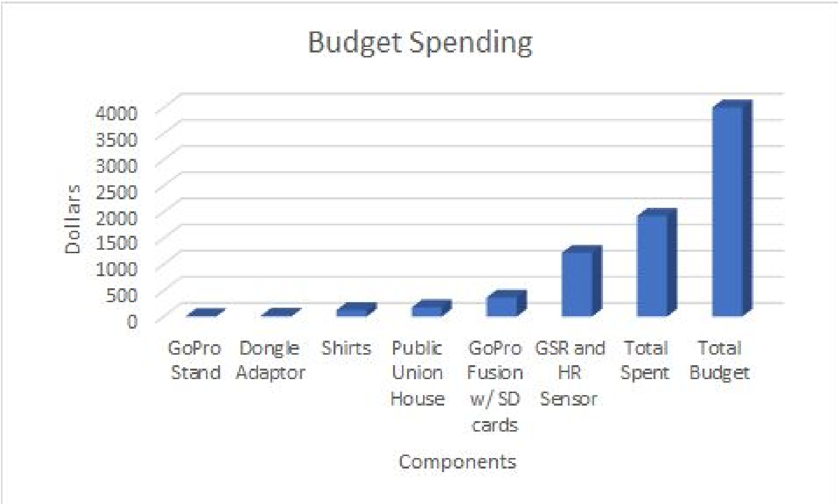


Figure 14. Budget Expenditure

A graphical representation of the project costs.

6.0 Lessons Learned

Throughout this past year, we have designed and developed a system aimed at treating eating disorders. Though we were only able to complete one scenario with its associated components (without the GUI and calorie counting software integration and full system integration), we believe we have created a solid foundation that future work can be built upon. After these are completed, we would like to conduct a trial on test patients.

During the development of the project this past year, our team has learned some valuable lessons on communication, teamwork, and the engineering process (V Model). In regards to communication, we learned the value of establishing and defining the scope and expectations with your customer. Only once you have outlined these can you begin to design and develop your product. We also learned that different customers have varying expectations for the level of involvement. We appreciated our sponsor's more hands-off approach in this regard as we possessed more autonomy and freedom in our design. Again, this lesson was important in communicating and addressing the changes resulting from COVID-19.

Another key takeaway from this project was to think critically and think like an engineer. With the creative sovereignty we had, we needed to conduct extensive research for our system requirements and components. We learned the importance of justifying and supporting each design decision we made throughout the development process. This was certainly a valuable learning experience because we had to fully integrate this logic and reasoning into the design of our system. In doing so, we thoroughly engaged in the design process, utilizing skills that will be beneficial in our future careers.

In regards to our system, we have learned how to design an effective system to treat eating disorders using best practices in the field as suggested by nutritionists and therapists. This also contributed to our lesson on justifying decisions like an engineer. We realized it was not our entire responsibility to treat the disorder but rather to design a useful and appropriate solution.

We also learned about adaptation. Changes happen in engineering all the time and it is important to reevaluate and accommodate. From changes to our system components and requirements to the current global situation, we have needed to adjust to move forward and complete the project as satisfactorily as possible.

The final lesson we learned was one of teamwork and resilience. Despite all the obstacles and the current global pandemic situation, we have learned your team is your most valuable resource. Their insights and contributions are essential to making your team work and to create a product you can all be proud of. The path you all take to the final product can be wholesome, with both fun and educational aspects. Teamwork is such an essential and distinctive trait of engineering. Together we grew as engineers, teammates, and people; undoubtedly important for our future endeavors.

This project and the class has taught us a lot about how to be an engineer and teammate. We have been able to preview and practice the fundamental skills we will use in our engineering careers.