Text to Motion Database

Test Report

Brendan Duke
Andrew Kohnen
Udip Patel
David Pitkanen
Jordan Viveiros

April 7, 2017

Contents

1 Preface

1.1 Revision History

Table 1.1: Revision History

Date	**V**ersion	**N**otes
March 14, 2017	0.0	File created
March 23, 2017	0.1	Initial template completed
March 26, 2017	0.2	Completed tables and organized sections

1.2 List of Figures

This document does not utilize figures in order to display our results.

List of Tables

2 Introduction

2.1 Purpose of Document

The purpose of this manuscript is to document the testing that has been performed on the Text-to-Motion web application. The testing that has been performed largely follows the test plan that was provided in the Test Plan document.

2.2 Scope of Testing

Following good design procedures, the Text-To-Motion web application has been modularized into 3 conceptual modules. The first module is a web framework that allows users to access key features like uploading an image, view a previous image, among others. The database is the second module in the application as it is used to store events from the website like the previously listed features. The third module is the most expansive and is the deep learning module that runs pose estimation on the images and pictures uploaded.

Following the application's modular decomposition, the tests have been decomposed into 3 conceptual tests, in addition to automated testing.

The first stage of testing is the minimum viable product, and was done to ensure that a base level of functionality was provided. This is where we test to see that all of the software modules have been created and can work together. With this in mind the types of testing that were done to ensure a minimum viable product involved both system and unit testing.

The second set of tests is Solution Constraints testing or Functional tests. In this set of tests, the performance of the deep learning algorithm is rigorously tested and quantified. In addition to testing the deep learning algorithm we test the database, and search capabilities with more specific parameters than those introduced in the minimum viable product.

In the third set of tests, we get a few users to test the non-functional requirements of the finished product. Examples of these requirements include: usability, look and feel requirements and learnability.

The final set of tests were the automated tests ran on the http-server in order to verify the GET and POST calls to and from the server.

The final section in this document includes a traceability table that helps to organize and explain how tests are connected to the requirements that were created in the software requirements document we created earlier.

3 Preliminary Testing

3.1 Minimum Viable Product

3.1.1 TextToMotion Availability

Description

This test is done to ensure that the TextToMotion Database (https://brendanduke.ca/) is functioning as intended and available to the public. These tests will be done by first accessing the home page of https://brendanduke.ca/, and then navigating to other pages within the website to ensure they are meeting the availability requirements.

Results

Test Num	**T**est	**I**nitial State	**E**xpected Output	**A**ctual Output	**R**esult
3.1	Input brendanduke.ca to view the home page	Google.ca	The home page of brendanduke.ca	The home page of brendanduke.ca	Pass
3.2	Add ../Account/Register to the URL or click ”Register”	brendanduke.ca	The user registration page	The user registration page	Pass
3.3	Add ../Account/Login to the current URL or click ”Login”	brendanduke.ca	The user sign in page	The user sign in page	Pass
3.4	Add ../ImagePoseDraw to the current URL or click ”ImagePoseDraw”	brendanduke.ca	The ImagePoseDraw page	The ImagePoseDraw page	Pass
3.5	Add ../Demo to the current URL or click ”Demo”	brendanduke.ca	The Demo page with the camera feed	The Demo page with the camera feed	Pass
3.6	Add ../TextToMotion to the current URL or click ”TextToMotion”	brendanduke.ca	The TextToMotion page	The TextToMotion page	Pass

3.1.2 Updating the database

Description

The database allows users to upload images to the database through the website through brendanduke.ca/ImagePoseDraw. The test will be considered a pass if an image can be uploaded successfully and then viewed through the interface within ImagePoseDraw.

Input Data

Throughout this section the input data is referenced by ”Name”.jpeg and the website location is referenced through ../ImagePoseDraw which represents brendanduke.ca/ImagePoseDraw.

Input	Description
AverageGuy.jpg	A picture of a man standing still and facing the camera
AverageGirl.jpg	A picture of a woman standing still and facing the camera
TomBrady.jpg	A picture of football star Tom Brady

Results

In order to get to the URL brendanduke.ca/ImagePoseDraw/Create the user can navigate to brendanduke.ca/ImagePoseDraw and click the buttons associated with adding a new image or video. After the upload is complete a user will be taken back to the ImagePoseDraw where they can search for the uploaded media.

Test Num	**T**est	**I**nitial State	**E**xpected Output	**A**ctual Output	**R**esult
3.6	Upload TomBrady.jpeg using the ImagePoseDraw functionality	.. /ImagePoseDraw/Create	Land on ../ImagePoseDraw	Landed on ../ImagePoseDraw	Pass
3.7	Upload AverageGirl.jpeg using the ImagePoseDraw functionality	.. /ImagePoseDraw/Create	Land on ../ImagePoseDraw	Landed on ../ImagePoseDraw	Pass
3.8	Upload AverageGuy.jpeg using the ImagePoseDraw functionality	.. /ImagePoseDraw/Create	Land on ../ImagePoseDraw	Landed on ../ImagePoseDraw	Pass

3.1.3 Verifying Pose Estimation

Description

After the user has uploaded an image for pose estimation they should be able to view the pose estimation that has been run on the image. In order to verify that some form of pose estimation has been run on the image the user must view their upload and have visible annotations made to the image to be considered a pass.

Results

An image once uploaded can be found within brendanduke.ca/ImagePoseDraw/Details/N, where N represents the number of the uploaded image. This can also be achieved by searching through the uploads.

Test Num	**T**est	**I**nitial State	**E**xpected Output	**A**ctual Output	**R**esult
3.9	View TomBrady.jpeg from ../ImagePose-Draw/Details/30	.. /ImagePose-Draw	An image of Tom Brady with pose estimated limbs	Tom Brady with pose estimated limbs	Pass
3.10	View AvergaeGirl.jpeg from ../ImagePose-Draw/Details/26	.. /ImagePose-Draw	An image of a female with pose estimated limbs	A female with pose estimated limbs	Pass
3.11	View AverageGuy.jpeg from ../ImagePose-Draw/Details/27	.. /ImagePose-Draw	An image of a male with pose estimated limbs	A male with pose estimated limbs	Pass

3.1.4 Search by Name or Description

Description

The web interface has the ability to search through the uploaded images and videos based on their name or information within the associated description. The test will be considered a pass if the name is input in the search bar and image is returned.

Results

Test Num	Test	Initial State	Expected Output	Actual Output	Result
3.12	Input ”Tom” in the search bar within ../ImagePoseDraw	.. /ImagePoseDraw	A list of figured including the Tom Brady image that was uploaded	A single return of the Tom Brady image	Pass
3.13	Input ”Average” in the search bar within ../ImagePoseDraw	.. /ImagePoseDraw	A list of results including the AverageGirl image	Two results, one of which was Average Girl	Pass
3.14	Input ”Guy” in the search bar within ../ImagePoseDraw	.. /ImagePoseDraw	A list of results including the AverageGuy image	Two results, one of which was Average Guy	Pass

3.2 Solution Constraints Testing

3.2.1 Deep Learning Methods Test

Description

In order to provide a proper demonstration of the deep learning mechanics that can be associated with pose estimation we will meet with Dr.Taylor to ensure that we are implementing the Bulat et al paper properly. This test will be considered a pass if Dr.Taylor is satisfied with the implementation of the previously mentioned paper.

Results

Test Num	Test	Result
3.15	Meet with Dr.Taylor in order to verify the integrity of the deep learning implementation within the TextToMotion Database.	Pass

3.2.2 Standard Data Format Test

Description

An automated test that checks if the human pose data used for the project is standard and compatible with existing software libraries.

Results

This test was technically a failure. The data was not converted to a format compatible with libraries. Instead the data is stored in JSON strings, which is a standard format, but not compressed as an ideal format such as HDF5 would be.

4 Functional Requirements

4.1 Input Data

Throughout this section the input data is referenced by ”Name”.jpeg and the website location is referenced through ../ImagePoseDraw which represents brendanduke.ca/ImagePoseDraw.

Input	Description
E9FY2.MP4	A short video of a woman eating a sandwich
U4XV9.MP4	A short video of a man waking up and getting out of bed
Z1A0Q.MP4	A short video of a man sitting on a stool

4.2 Supported Video Encoding Test

4.2.1 Description

Tests whether the ReadFrames API is able to decode MP4 files. If we are able to run pose estimation on the video, then the ReadFrames API is able to process the frames.

Results

Test Num	Test	Initial State	Expected Output	Actual Output	Result
4.1	Running ReadFrames API on E9FY2.MP4, to run pose estimation	ReadFrame API	A compiled E9FY2.MP4 with pose estimation executed	E9FY2.MP4 that has been pose estimated	Pass
4.2	Running ReadFrames API on U4XV9.MP4, to run pose estimation	ReadFrame API	A compiled U4XV9.MP4 with pose estimation executed	U4XV9.MP4 that has been pose estimated	Pass
4.3	Running ReadFrames API on Z1A0Q.MP4, to run pose estimation	ReadFrame API	A compiled Z1A0Q.MP4 with pose estimation executed	Z1A0Q.MP4 that has been pose estimated	Pass

4.3 Frame Reading Timestamp Accuracy Test

4.3.1 Description

Tests whether the timestamps on the frames returned by the ReadFrames API match their temporal position in the original video stream. Our input data is identical to the previous test.

4.3.2 Results

Test Num	Test	Initial State	Expected Output	Actual Output	Result
4.4	Run ReadFrames API on E9FY2.MP4, to verify the timestamps match	ReadFrame API	Matching timestamps at 5 seconds	Matching timestamps at 5 seconds	Pass
4.5	Run ReadFrames API on U4XV9.MP4, to verify the timestamps match	ReadFrame API	Matching timestamps at 10 seconds	Matching timestamps at 10 seconds	Pass
4.6	Run ReadFrames API on Z1A0Q.MP4, to verify the timestamps match	ReadFrame API	Matching timestamps at 6 seconds	Matching timestamps at 6 seconds	Pass

4.4 Human Pose Estimation Data Quality Test

4.4.1 Description

Test to ensure the data quality produced by the human pose estimator component was acceptable.

A set Charades videos will be processed by the human pose estimator, and skeleton animations corresponding to the generated human pose data will be created (this is a scoped part of the software pipeline). A double-blind test will be ran, wherein testers will be shown random mixed sets of the skeleton animations produced by McMaster Text to Motion, together with skeletons from actual motion capture data coming from CMU"s motion capture lab. Testers will indicate whether they think the motion capture data came from actual motion capture, or from the pose estimation software.

The McMaster Text to motion Results should be guessed as accurate at similar rates to the Charades tests.

4.4.2 Results

We provide the ratio of Text-to-Motion images guessed as accurate compared to the Charades Images.

Test	**C**harades/ TextToMotion	**R**esult
Nick was shown a set of videos and asked to determine which was generated by the TextToMotion Database	1/1	Pass
Drew was shown a set of videos and asked to determine which was generated by the TextToMotion Database	7/5	Pass
Sarah was shown a set of videos and asked to determine which was generated by the TextToMotion Database	5/4	Pass

4.5 Database Output Full Range Coverage Test

4.5.1 Description

Tests to be sure all entries in the database can be successfully searched for. The videos provided from earlier tests are put into the database, and have been renamed for testing purposes.

4.5.2 Input Data

Throughout this section the input data is referenced by ”Name”.jpeg and the website location is referenced through ../ImagePoseDraw which represents brendanduke.ca/ImagePoseDraw.

Input	Description
`Waking_Up.mp4`	Given tags sleeping, boy, man, sleepy, getting up, table
`Eating.mp4`	Given tags sandwich, eating, girl, woman, table
`Stool.mp4`	Given tags man, stool, corner, sitting

Results

The given set of videos appeared in the returned list of videos from a database search.

Test Num	**T**est	**I**nitial State	**E**xpected Output	**A**ctual Output	**R**esult
4.7	Search for the Waking_Up.mp4 within ../ImagePoseDraw by searching ”Waking”	.. /ImagePoseDraw	A list of results with Waking_Up in the results	A single result of the Waking_up.mp4	Pass
4.8	Search for the Eating.mp4 within ../ImagePoseDraw by searching ”Eat”	.. /ImagePoseDraw	A list of results with Eating in the results	A single result of the Eat.mp4	Pass
4.9	Search for the Stool.mp4 within ../ImagePoseDraw by searching ”Stool”	.. /ImagePoseDraw	A list of results with Stool in the results	A single result of the Stool.mp4	Pass

4.6 Database No False Positives

4.6.1 Description

Tests that the database search does not return any false positives, such as videos or images that do not contain searched words. The same videos from the previous test will be used with the same tags. Thus, we will search with tags other than those provided. If no videos appear, then the test is a success.

Results

Using the nonsensical keywords from our input data, no search results were returned, meaning that the test passed.

Test Num	**T**est	**I**nitial State	**E**xpected Output	**A**ctual Output	**R**esult
4.10	Search for ”Raptor” within ../ImagePose-Draw	.. /ImagePose-Draw	A list of results that don’t contain the input data	No results were returned	Pass
4.11	Search for ”Exponent” within ../ImagePose-Draw	.. /ImagePose-Draw	A list of results that don’t contain the input data	No results were returned	Pass
4.12	Search for ”Glasses” within ../ImagePose-Draw	.. /ImagePose-Draw	A list of results that don’t contain the input data	No results were returned	Pass

5 Non-Functional Requirements

5.1 Usability

5.1.1 Description

In order to determine the usability of the Text-to-Motion database, a small sample of users were asked to use the website to perform some predetermined actions and answer questions afterwards.

Before the participants were asked to preform any actions they were given a minute to familiarize themselves with the interface, but were not given any guidance or tips from the development team.

Once the time was up they were asked to upload an image on mobile or desktop through their webcam, a URL or from a file saved within the computer.

While performing the required action the participant's time was recorded and used to determine if a requirement had passed or failed.

Upon completion of the task the users were asked to rate the style and design of the website on a scale from 1 to 10.

5.1.2 Results

The results from the participants can be seen throughout the Non-Functional Requirements along with a pass or fail based on the requirements description.

5.2 Look and Feel Requirements

5.2.1 Colour Scheme

Description

A test to see if the colour scheme of the website is visually appealing. Making the colour scheme of the website visually appealing to a larger audience is important as it will help keep users on the website and ideally provide a reason to reference it to a friend or colleague.

When testing the colour scheme anything above a 6 was considered a pass. We chose anything above a 6 because if the users were to rate it above a 6 it means they are within the upper percentage and enjoyed the colour scheme. This may result in them recommending the website to others and providing more traffic.

Results

Test Num	**U**ser	**R**ating	**R**esult
5.1	Nick	7	Pass
5.2	Drew	9	Pass
5.3	Sarah	7	Pass

5.3 Style Requirements

5.3.1 Minimalistic Web Design

Description

The website interface should be minimal and should inform the user of valid actions through visual means. This is a modern requirement in website design, the more minimalistic design you can provide the better it will be for the user experience and any new technologies use to enforce this will help drive traffic.

While testing the minimalistic design anything above a 7 was considered a pass. This required a higher rating then the colour scheme as we believed it would be a key factor in providing the user with an enjoyable experience and help provide a greater amount of traffic through uploads.

Results

Test Num	**U**ser	**R**ating	**R**esult
5.4	Nick	9	Pass
5.5	Drew	9	Pass
5.6	Sarah	8	Pass

5.4 Ease of Use Requirements

5.4.1 Upload/Download

Description

Through the web interface a user should be able to upload a picture using either a mobile phone camera, URL, or saved file. The participant will start on the Home page and be asked to upload an image through one of the methods previously mentioned.

In order for this to be considered a pass it should take the users 45 seconds or less to complete the upload process and click the button. The required time of 45 seconds was chosen due to the amount of time that was required to find an image, URL, or file while running trials throughout the development.

Results

Test Num	**T**est	**I**nitial State	**U**ser	**T**ime	**R**esult
5.7	Upload an image from a URL	.. /ImagePose-Draw/Create	Nick	32 seconds	Pass
5.8	Upload an image from a mobile device	.. /ImagePose-Draw/Create	Nick	38 seconds	Pass
5.9	Upload a file stored within the Desktop	.. /ImagePose-Draw/Create	Nick	27 seconds	Pass

5.4.2 Text Box Functionality

Description

The user should be able to input a descriptive word or phrase into a text-box from within the web interface in order to search for a video. With a the website being part of a larger database the ability to search for an image that the user just uploaded is an important factor in determining the usability for users.

In order to complete this task the users were asked to search for a specific word and display the results. Any time below 10 seconds will be considered a pass. Each user was allocated 10 seconds because we did not want the action of typing the keyword to be the bottleneck when searching for a database entry.

Results

Test Num	**T**est	**I**nitial State	**U**ser	**T**ime	**R**esult
5.10	Search for ”Woman”	.. /ImagePose-Draw	Nick	4 seconds	Pass
5.11	Search for ”f”	.. /ImagePose-Draw	Drew	3 seconds	Pass
5.12	Saerch for ”the”	.. /ImagePose-Draw	Sarah	4 seconds	Pass

5.5 Learning Requirements

5.5.1 Usability Tests

Description

The user should be able to interact with the website without prior knowledge. They will given a minute to explore the website. After that time the participants were asked to rate the usability on a scale of 1-10. An average of 6 is required for a pass.

Results

The users' usability ratings allowed the site to pass this test.

User	**R**ank	**R**esult
Nick	6	Pass
Drew	8	Pass
Sarah	8	Pass

5.6 Politeness and Understandability Requirements

5.6.1 Hiding the Inner Workings

Description

Users should not be able to see the deep learning model and its training when using the pose estimation. When prompted the website should display the correct skeletons without any low-level detail. Once uploaded the participants were asked if they saw anything that seemed out of place or any information on the deep learning process, if they did not it will be considered a pass.

Results

Users indicated that the deep learning model was encapsulated from their view, and hence this test passed.

Test	**U**ser	**R**esult
Uploading an image from URL	Nick	Pass
Uploading an image from mobile	Drew	Pass
Uploading an image from desktop	Sarah	Pass

5.7 Speed and Latency Testing

5.7.1 External Database Connection Response Time

Description

The web interface should be able to connect to an external database and store or query items. In order for this test to be considered a pass the confirmation of the image being uploaded would have to occur within 30 seconds so that additional resources are not wasted by the database. Testing this will occur by uploading an image and testing the total time taken.

Input Data

Input	Description
Image from a URL	An image of a male
Image that was saved within the desktop	An image of Seth Rogan

Results

According to the response times of this automated test, the database queries were executed fast enough for the test to pass.

Test	**T**ime	**R**esult
Uploading the image from a URL	28 seconds	Pass
Uploading an image from desktop	29 seconds	Pass

5.7.2 Website Search Responsiveness

Description

When given a word or phrase the web interface will be able to respond with an image or video of a pose or action within a two minutes.

Input Data

Input	Description
Creepy	Searched using a tag within the description
Seth Rogan	Searched using a tag within the description

Results

Test	**T**ime	**R**esult
Search for “Seth”	1 second	Pass
Search for “Creepy”	2 seconds	Pass

6 Other Relevant Testing

Again we test this on the same files we have used in the previous tests: `TomBrady.jpg`, `AverageGirl.jpg` and `AverageGuy.jpg`.

6.1 Precision and Accuracy

6.1.1 Bone and Joint Position

Description

The pose estimation should accurately predict the placement of joints and bones of the person in the provided photo. This will be determined with visual means with an uncertainty range of 20 pixels.

Input Data

Input	Description
`TomBrady.jpg`	A picture of a man standing still and facing the camera
`AverageGirl.jpg`	A picture of a woman standing still and facing the camera
`TomBrady.jpg`	A picture of football star Tom Brady

Results

We were able to qualitatively confirm that these tests passed for the given input images. A more rigorous "PCKh" metric is used to formally evaluate the performance of our deep learning model on single-person pose estimation.

Test	**R**esult
Tom Brady	Pass
Average Girl	Pass
Average Guy	Pass

6.1.2 Deep Learning Model

Description

The **PCKh** metric, used by the MPII Human Pose Dataset, defines a joint estimate as matching the ground truth if the estimate lies within 50% of the head segment length. The head segment length is defined as the diagonal across the annotated head rectangle in the MPII data, multiplied by a factor of 0.6. Details can be found by examining the MATLAB evaluation script provided with the MPII dataset.

If our model can achieve 80% total PCKh then the test is considered a pass.

Results

Our model achieves 85% PCKh, thus this test is a pass.

Test	**PCKh Value**
r ankle	68.907%
r knee	77.201%
r hip	83.583%
l hip	84.444%
l knee	77.419%
l ankle	69.055%
pelvis	89.776%
thorax	98.071%
upper neck	97.823%
head top	96.557%
r wrist	81.288%
r elbow	87.205%
r shoulder	92.918%
l shoulder	92.828%
l elbow	85.278%
l wrist	80.719%
Total PCKh	**85.195%**

6.2 Reliability and Availability Requirements

6.2.1 Software Availability

Description

The software component of the project should be available at all times. If we can have an event regularly occur then it will be considered a pass. To do this we arranged to have the pose estimation algorithm automatically called on to process a single image at 4 hour intervals, and record the time.

Results

The software successfully processed the image at the specified 4 hour intervals over a period of 2 days.

6.2.2 Website Availability

Description

The software component of the project should be available at all times with the exception of maintenance and migration. When we make a web server call we should receive an HTTP verified response. To do this we will have three users sending a HTTP POST request to the server. If they receive a response the test is considered a pass.

Results

All three users were able to receive responses to their HTTP POST requests, therefore we consider this test a pass.

6.3 Robustness or Fault-Tolerance Requirements

6.3.1 Web Interface Error Handling

Description

The web interface should respond to unhandled exceptions by throwing the corresponding error messages. If an exception is thrown and an error message is displayed then the test is considered a pass.

Input Data

Input	Description
Scenario 1	Try uploading an image while the image storage service (Amazon S3) is not available.
Scenario 2	Trying to search an image on the database while the database server is down.

Results

For Scenarios 1 and 2, appropriate exceptions were thrown and error pages displayed to the user. Therefore this test is considered to have passed.

6.4 Capacity Requirements

6.4.1 Multiple Connections

Description

The web interface should be able to serve multiple connections. If the interface can support 5 connections at once it is considered a pass.

Input Data

Input	Description
Andrew	The connection to the website corresponding to tester Andrew
Brendan	The connection to the website corresponding to tester Brendan
Udip	The connection to the website corresponding to tester Udip
Jordan	The connection to the website corresponding to tester Jordan
Dave	The connection to the website corresponding to tester Dave

Results

The 5 users were able to successfully connect to the website, and run queries simultaneously without experiencing significant slow down or waiting.

6.4.2 Database Capacity

Description

The database should contain at least 5GB of data in order to facilitate growth.

Results

The database successfully handled us uploading a total of 5.1GB of data in the form of pictures of various qualities. These pictures were of randomly acquired photos from Google, which were combined to form the data used in this testing.

6.5 Scaling of Extensibility Requirements

6.6 Operational Environment Requirements

6.6.1 Linux Friendly TensorFlow

Description

The web interface should be run on a Linux friendly server that can access the TensorFlow model either directly or indirectly. By creating an interface that successfully runs on a Apache or nginx server this test will be considered a pass.

Results

Our service is successfully running on a production Linux server using nginx.

6.6.2 Export Types

Description

The project should be able to export multiple types of media (JPEG, PNG, etc) in order to support all major operating systems. We will use `TomBrady.jpg` for this test.

Results

This test was a guaranteed pass due to the fact that our website stores images as a base64, meaning it can be converted into any type.

Test	**R**esult
PNG	Pass
JPEG	Pass
BMP	Pass

7 Traceability

Table 7.1: Traceability Matrix for Test-Requirement Relationships

Test	**D**escription	**R**equirement
3.2, 4.2, 4.3, 5.8.2	Tests which measured performance Accuracy of Deep Learning Algorithm	Req 1, 8, 23(Speed and Latency)
4.1, 5.3.1, 5.7.1, 6.4.2	System tests, measuring reliability of the web framework.	Req 7, 12, 17, 20, 30, 38, 39
4.4-4.6	Unit and Systems Tests Grading Database Search Performance	Req # 9, 10, 11,
4.5, 6.3.1, 6.3.1, 6.4.1	Security and Data Integrity Tests	Req # 10, 27, 29
4.1.1, 5.8.2, 6.4.1	Proper Formatting Tests	Req # 7, 23, 29
4.2.1-5.8.1,6.1.1-6.2.2	User Interface Tests	Req # 13-21, 24-26

7.1 Modules

Similarly, the following is a traceability table explicitly relating test cases to modules:

Table 7.2: Tests and Modules Relationships

Test #	Module
3.1(all subsections), 3.2.1, 4.4-4.6, 5(all sections), 6.2(all subsections) - 6.4(all subsections), 6.6.2	ASP.NET and DB
3.1.3, 4.3, 5.8.2, 6.1, 6.5	TensorFlow Models
3.13, 3.2.2, 4.1-4.2, 5.8.2, 6.2(all subsections), 6.6.1	Python HTTP Server

8 Changes After testing

The first of our major changes would likely be to the website interface. Though our testers reviewed it favourably – there were numerous references – but we were also given consistent criticism that an alternate colour scheme might be a slight improvement. As such we have agreed to experiment with those improvements.

Testing also revealed that a JavaScript application to allow continuous requests to the HTTP server as well as utilize a mobile platform would have greater applicability to the Guelph team. This improvement would come alongside a function on the website to display statistics. This would help insure greater availability. The ability to track the database live would be both useful to the testing team as well as an entertaining aspect for the product demo.

The last major improvement we wish to provide is an improved JavaScript plugin for drawing the skeleton on top of an image. Testing revealed that there was flicker with the drawn skeletons, and as such, could lead to false negatives, where a skeleton that appears inaccurate is actually just limited by our current skeleton drawing methods.