**Buffalo Concussion Treadmill Test Digitalization**

| **Joshua Kent**  for Submission  Atlanta, United States  jkent40@gatech.edu | **Sami Belhareth**  for Submission  Atlanta, United States  sbelhareth3@gatech.edu | **Karan Singh**  for Submission  Atlanta, United States  karanjs@gatech.edu |
| --- | --- | --- |

**INTRODUCTION**

The Buffalo Concussion Treadmill Test (BCTT) is a metric used by trained professionals (physical therapists, certified trainers, etc.) to help patients recovering from concussions and usually those suffering from post-concussive symptoms (PCS). This project will digitalize the traditional test into a mobile iOS application and web application such that it can be conducted more efficiently. Furthermore, the digital setup will permit the use of the data for health informatics and clinical decision support in future work.

Overall, the project will consist of three main aspects. First, the IOS application. This will be the patient-facing interface that will prompt them along the BCTT and capture biometric information like their heart rate. Second, the front-end of the web server. This will require designing an interface for the physical therapists to input notes, capture data about the patient for the BCTT, and have a portal to all previous tests. Third, the back-end of the web server. This will entail receiving the data from the iOS app, saving it in a database, and using it to track performance.

**PREVIOUS/RELATED WORK**

While there has not been substantial literature pertaining to the digitization of the BCTT, there has been work in proving the efficacy and validity of the BCTT, particularly literature detailing the relationship between BCTT performance over time and concussion recovery [1].

Using Google Scholar, we searched for the following pairs of boolean search terms: “Buffalo concussion treadmill test” AND “digital”; “Buffalo concussion treadmill test”

AND “FHIR” (acronym for fast healthcare interoperability resources, a standard in the electronic medical record field). There were no relevant search results that included these terms of interest, indicating that there is little or no work pertaining to implementing the Buffalo concussion treadmill test in a digital format.

The protocol used for conducting the BCTT test currently [2] will guide our work.

**OUR WORK**

**Web**

The web portion of the project concerns maintaining the back-end as well as front end logic. The back-end for the service is responsible for maintaining storage, individual test sessions while providing API contracts for the wearable (iOS) and the front-end client. To deploy the web application, we create a local network NodeJs server instance for the front-end client and wearable device to interface.

The front-end part of the web application is purposed for the physician assisting the patient in the treadmill test session. Along with providing an interface for the physician the front-end is also responsible for fetching all the patient history from the NodeJs server.

The diagram in Fig 1. (see appendix) represents the flow of communication between all three instances of the whole application.

**iOS**

The IOS portion of the project is comprised of three main aspects: (1) getting the necessary biometric data from the Apple Watch (2) creating a session between the watch and phone so this information can be communicated in real time and (3) sending this data to the web application to be used by the clinician while conducting the BCTT.

In respect to the first goal, we have successfully created a watchOS application (runs on Apple Watch) that continuously reads the user’s heart rate. On first use, the application prompts the user for permission to access their health data. After this approval, we use Apple’s HealthKit framework to query for their heart rate in real-time and display it right on the watch.

In terms of the second goal, we use WatchConnectivity to communicate between the WatchKit extension and its companion IOS app. We transfer the heart rate from the watch to the iPhone through the messaging functionality that is built into the WCSession. Once on the phone, the data is displayed so it can be viewed from both devices.

The third facet of the IOS part of the project has not yet been addressed; in the coming weeks we will focus on this area. More details regarding the implementation of how we will do this can be found in the Future Work section.

**DISCUSSION**

**Web**

The major challenge for the web application is designing a system that has 3 end-points (client, server and wearable peripheral) as opposed to traditional 2-endpoint client-server applications. Since the wearable sensor data and the monitoring front end both need to sync in real time, the server must provide abstracted contracts to both these clients.

We propose a design (Fig. 1) that conceptualizes this abstraction between the front-end client and the wearable client, in this design the server acts as a synchronization mechanism between the front-end monitoring client and the sensor-enabled wearable client. Both these clients will periodically push (wearable) and pull data (front-end monitor) from the server. This real-time periodic sync is aimed to help the physician to monitor the patient’s status at every moment.

**iOS**

This project has taught us many new skills that we’ve previously not known until this point. Though we have IOS development experience, no team member has dealt with reading a user’s heart rate from an Apple Watch, creating companion WatchOS and IOS applications, or communicating data through both apps with a WCSession. This process has allowed us to read a lot of documentation published by Apple in the past years that not only strengthens our knowledge of mobile development, but also provides us an opportunity to implement concepts of ubiquitous computing that we’ve only learned about–not applied. In addition to the new concepts aforementioned, this work has let us learn about privacy requirements, transfer speeds, etc.

**FUTURE WORK**

**Web**

With the communication design between all three end-points established, the next task for the web portion of the project is to implement these contracts along with storage logic on the back-end to enable the full application end-to-end.

**iOS**

The majority of the future work for the IOS portion of the project lies in the third aspect previously described in the Our Work section. We need to take the heart rate data from the iPhone and send it to a web application to be stored with exertion scale recordings and clinician notes. For purposes of this class, we will run the web application locally and therefore can use a simple HTTPS request to communicate the information. This will be the last major goal.

After this is completed, we will focus on the aesthetics and design of the WatchOS and IOS application. We will beautify the software and make sure it has a functionable and efficient flow. Time dependent, we may add different prompting or timing features on the phone to further improve the user experience.

**CONCLUSION**

We have developed an iOS app in parallel with a webserver and front end capable of addressing the needs of a digital BCTT.

The iOS app is currently able to read heart rate data using the Apple Watch’s sensors, display it on both the Apple Watch and the iPhone that the Watch is connected to, and relay that information to an endpoint of interest. Future work will entail refining the user interface to address the needs of the physician and the patient.

The web app is currently able to deploy a local network NodeJS server instance in order to connect with both the front end and the Apple Watch. Future work will entail further development on the front end to accept input from the clinician.

**REFERENCES**

[1] Mohammad N. Haider, John J. Leddy, Charles G. Wilber, et al. 2019. The predictive capacity of the buffalo concussion treadmill test after sport-related concussion in adolescents. Frontiers in Neurology 10.

[2] John J. Leddy and Barry Willer. 2013. Use of graded exercise testing in concussion and return-to-activity management. Current Sports Medicine Reports 12, 6: 370–376.

**APPENDIX**

[1] Fig. 1. Flow of communication between front-end back-end and wearable peripherals.

