MOVEMENT SMOOTHNESS APP

Approved: ___ Songqing Yue, Ph.D _____ Date: _04/01/2019_____
Paper Advisor

MOVEMENT SMOOTHNESS APP

A Seminar Paper

Presented to

The Graduate Faculty

University of Wisconsin-Platteville

In Partial Fulfillment

Of the Requirement for the Degree

Master of Science in Computer science

By

Nadine L Gadjou Kengne
2019

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Qi Yang, for the patient, guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

Special thanks to my teacher and supervisor Dr Songqing Yue as well as Dr Jeffrey Cowley. who gave me the golden opportunity to do this wonderful project on the topic (**Movement Smoothness App**). This Project also helped me in doing a lot of Research, and I came to know about so many new things.

I am grateful to all of those with whom I have had the pleasure to work on this and other related projects.

I would also like to thank my family and friends who helped me a lot in finalizing this project within the limited time frame.

I must express my gratitude to my sisters Doris, Nidel, Stephanie, Gladis and my brother Brice for their continued support and encouragement.

Abstract

MOVEMENT SMOOTHNESS APP

Nadine L Gadjou Kengne

Under the Supervision of Songqing Yue, Ph.D.

TABLE OF CONTENTS

Chapter I Introduction

Smartphones have become an unavoidable tool in many countries and even an important part of life. Its applications are used nationwide by millions of users every time. Although they are becoming more and more affordable. Most smartphones are composed of a processor, graphics chip, advanced connectivity and an inertial motion unit (IMU), with a 3D-accelerometer, magnetometer and gyroscope as standard features. Moreover, smartphones contain more technology such as a screen display, an audio system or a haptic feedback system that enables interaction with the user of the device. To use all these functionalities, smartphones can run specific software, called “applications”. All these features in the smartphones have widely been used as tools for scientific and clinical research, especially in the healthcare and physical activity monitoring fields. This work proposes an application that use the built-in motion sensors in a smartphone to provide measurable data to clinicians, therapists, and coaches, for people suffering from fine motor skills disabilities or injuries.

Project Description

This application will use the built-in motion sensors in a smartphone to gather data about the smoothness of a given phone movement. The idea is that the existing sensors in the cell phone can be used as a budget measurement system that can provide data (remotely) to clinicians, therapists, and coaches. To do this the phone would be grasped in the hand or attached to the body while a person executes a given movement.

Summary

Clinicians, therapists, and coaches are examples of people who would be reviewing the data saved from the application, in order to evaluate the users fine motor movements. Smoothness in motion can be accurately evaluated by looking at the jerk in a movement, or the

derivative of acceleration. The jerk is the third derivative of position-time data, which can be calculated using raw data gathered from the internal sensors.

Core Features

- User can create phone movement tests with a description detailing how to run the test
- User can run tests that will record phone movement data in a specific time interval
    - The app will utilize the internal accelerometer to record position/time data
    - Data is then formatted to produce the jerk metric (third derivative of position-time data)
    - Jerk data is used to quantify “smoothness” in a hand movement
- Movement data will be saved to be reviewed by person administering the test
- User can compare new trial results with old results to identify progress/trends
- User can view number of submovements in a given test trial
- User can run tests independently and submit data for analysis

Goal of Project

The goal of this app is to gather information about the quality of a person's motor movement. The project Idea was given to us by Prof Cowley from HHP. The main goal of this project is to provide measurable data to clinicians, therapists, and coaches, for people suffering from fine motor skills disabilities or injuries. This application will be used by patients to track and record fine movement to provide information on their injury or disability, and this data will be able to be viewed by their clinician, therapist, or coach in order for them to monitor progress of therapy. We expect to learn how to create a mobile application in the Android environment, and how to access accelerometer and other built in phone mechanisms in order to record this data. Hopefully this app can provide valuable data to people suffering from ailments affecting

their fine motor skills.

Example movements

Phone is resting on a table/counter. Phone is picked up and moved to a shelf or other location. The phone is held in the hand with the hand resting at the side. Phone is lifted to ear/mouth/opposite shoulder/behind the head.

Example of Usage

A therapist has a patient with Parkinson's and creates a movement test with this app. The test is composed of the patient moving the smart phone between three specific locations in a set amount of time. The patient is then given a week to perform the test 10 times. At the end of the 10th test, the gathered data are store in an online database and will be query by the therapist for evaluation and analysis.

Chapter II System Design and Implementation

System Requirements

To successfully develop an application, we must first define all the requirement that we want our application to satisfied. Below are some of our system requirements

- The admin must be able to create tests.
- The admin must be able to edit tests.
- The admin must be able to delete tests
- The user must be able to run tests.
- The admin must be able to view test results
- The system must be able to record test results and display them graphically
- The admin must be able to view plots of recorded movement data

System Design

Figure 1 show the navigation diagram of the application. On open, the user is prompt with a *select user screen* where he has the choice between the user tab of the Admin tab.

If the user tab is selected, the next screen is the *user test list screen* where all the test to be performed are listed. When a test is selected, the next screen prompted is *user test info screen* containing the test description, and a button to run the test. When the button is pressed the next screen is the *run test screen* with a stopwatch displaying the running time and a button to complete the test.

If the admin tab is selected, the next screen prompted is the *admin test list tab* where all the tests are listed and an *add test* button that allow the admin to navigate to a *add test tab*. When a test is selected, the next screen is the *admin test info screen* containing the test description and

the list of all the test trial performed by the user. When a trial is selected, the next screen is *the trail info screen.*

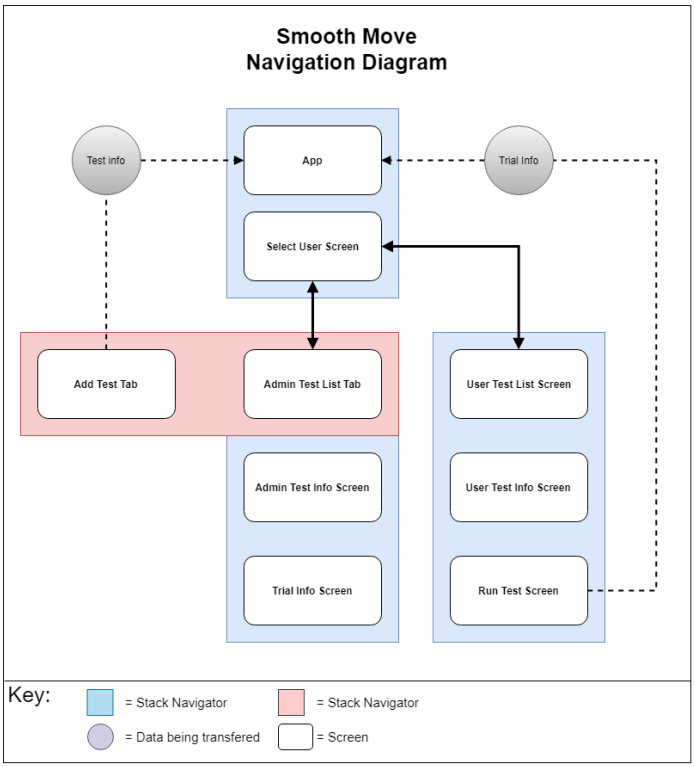


Figure1: navigation Diagram

Implementation

The implemented application satisfied to all the requirements mention early on the requirement section. To create test, the admin needs to specify some dynamic information such as the test's name, the test's duration the number of trial required for a particular test, the maximum speed that the patient must reach to succeed the test, and the test description (describing the type of movement the patient or user need to perform).

When a test is successfully added, it should appear on the admin test list screen and user test list screen.

If a test is selected from the user side, the next screen (user test info screen) show the test description, the number of trials require for that test, and a button to run the test.
In the *run test screen*, we use react accelerometer to record the phone accelerometer data. we use those data to calculate the speed by using the momentum formula (the quantity of motion of a moving body). We also use those data to calculate the jerk [1].

If a test is selected from the admin side, the next screen shows the test description and the list of trials already performed by the user. That list shows the trial number, the trial date and the time. Formatted using *the react native moment* library. The list is empty if no trials have been performed yet. When a trial is selected, the admin is able to see the summary result of the trial such as (the test duration, weather the test passed or failed, the maximum speed reach, the jerk cost, the average speed reach, the graph to visualize the speed peak, and the jerk). We used a *react native pure chart* library to plot the graphs.

When a user is performing a test, the accelerometer data (given in a three-dimensional values x, y, z) are stored into a MongoDB database. We use a mongo DB script to query and save those data into excel document which would be evaluate with MATLAB to find the total jerk cost. With formula given bellow [2].

$$Jerk - Cost = \frac{1}{2}\int_0^T \left\{ Jerk_x^2(t) + Jerk_y^2(t) + Jerk_z^2(t) \right\} dt$$

$$Jerk(t) = \frac{d\{accl(t)\}}{dt}$$

To Calculate the speed that the phone is moving with, we have used the momentum formula (which is the quantity of motion of a moving body). we have used accelerometer data generated by the library call React-Accelerometer (which generates data in the form of three-dimensional values x, y, z representing the acceleration in m/s2 and generate every 100 ms).

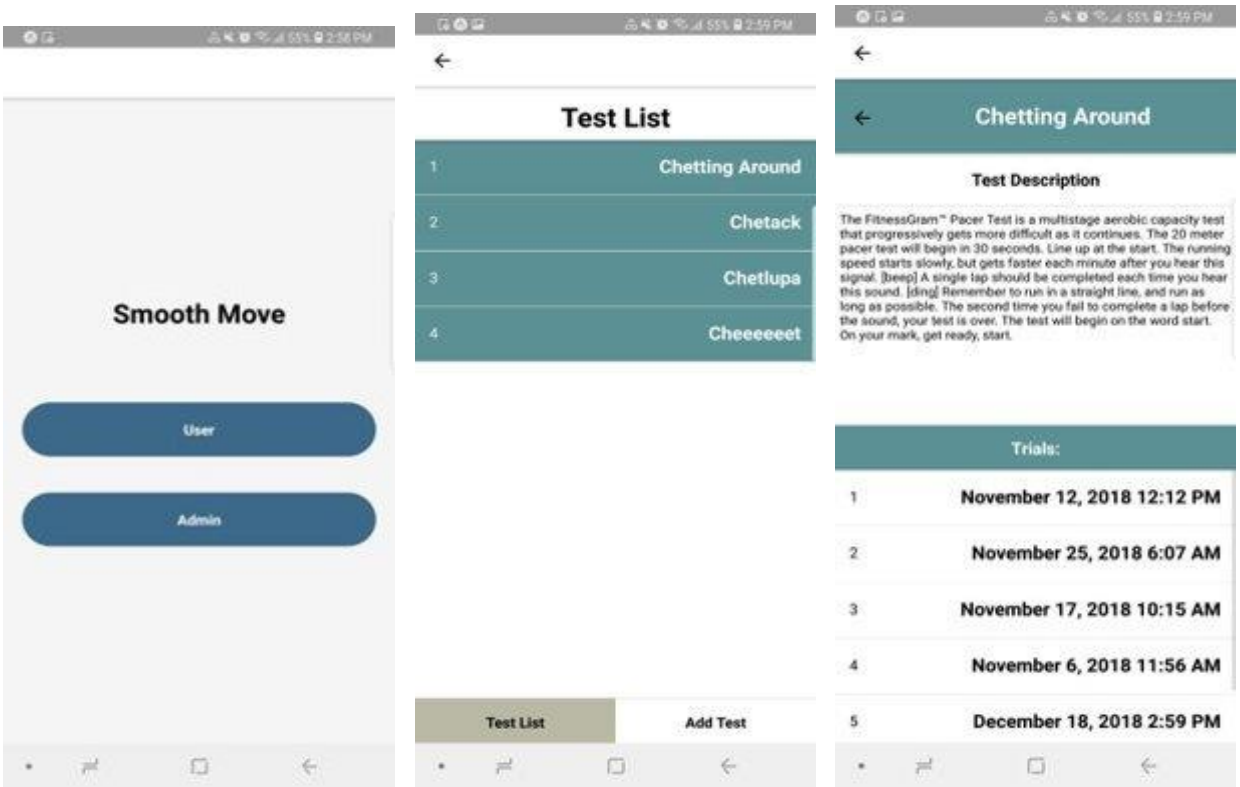
The Axios library is used to send data to a mongoDB. For the entire application, we created a database name “data”. It contains collections which are the equivalence of table in a relational database. Thus, we have one collection for each test. Each collection gives information of the current test such as: the testID, the trialID and the different X, Y, Z values

Libraries Used

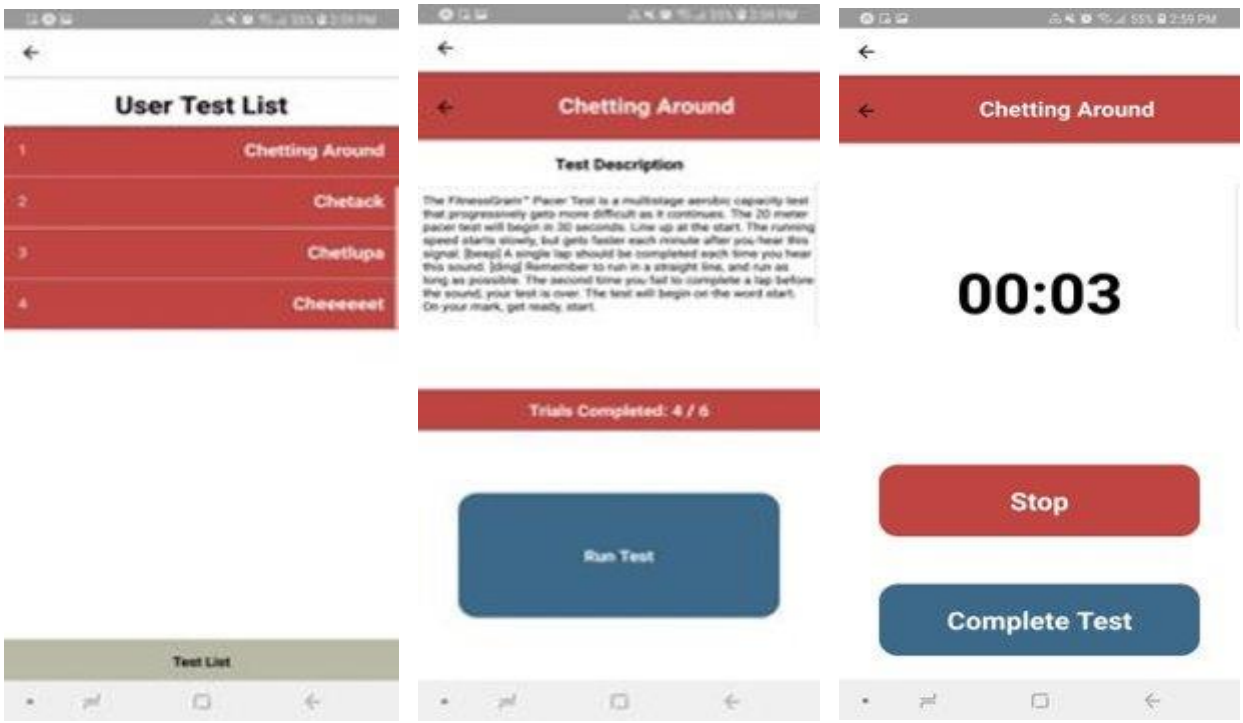
- Moment.js, for creating and formatting date/time objects
- React-Accelerometer: used to record accelerometer data
- Swipeout: used for deleting functionality of trials and tests
- Axios: used to send data to a server.

- Use of react -accelerometer to calculate the speed of a moving phone

Admin Screenshots



User Screenshots



Lessons learned

- Use of nested navigators to manage routes between screens
- How to store and manage data
    - All data store in App.js
    - All methods for modifying those data also in App.js
    - Methods send to child classes as screenprops

Run the application

To run the application, you need to cd in the project directory and install all the following dependencies:

Npm: with `npm install`

Navigation: with `npm i react-navigation`

Moment: with `npm i react-moment`

Swipeout: with `npm i react-native-swipeout`

React native pure chart: with `npm i react-native-pure-chart`

Axios: with `npm i axios`

To run the server, you need to first install the mongoDB, then lunch the mongodb using the command “mongod” to connect to your local application. By default, it is on port 2717.
To lunch the server, cd into the folder call express- Backend and run the command “node app.js”.

The database (“data”) has three collections (accData, accData2, accData3)
To see data stored in the collection accData, use de command (db.accData.find()).

Once the data are successfully store in the database, you can query the data and send the output in an excel document using the command *mongo data mongoScript.js > out.csv*
here data is the name of the database. The data store on the .csv file can be used by the clinicians to analyze new insight

Chapter III Conclusions and Recommendations

In this project we have used JavaScript and react native to build a native Mobil app that will allow people suffering from fine motor skills disabilities or injuries to provide measurable data to clinicians, therapists, and coaches. Base on the above explanations and the implemented app, the following conclusions and recommendations can be drawn. The app allows the patient to see how fast he is moving and whether he has reached the threshold set by the admin for that test. The app allows the admin to create or deleted a test, to view the trial result, to delete a trial and to see the result of all the tial performed by the user as well as the speed peak graph of each trial.

Although the app is functional, it only cares about a single user with multiple tests and multiple trials. In the future we plan to add more pages to allow the admin to handle multiple users and adjust the color style for each test already performed by the user.

References

https://developer.android.com/reference/android/hardware/SensorEvent.html

I Minami, K. Oogai, T.Nemoto, T. Nakamura, Y. Igarashi, N. Wakabayashi. Measurement of jerk-cost using a triaxial piezoelectric accelerometer for the evaluation of jaw movement smoothness. "*journal of oral rehabilitation 2010"*

https://docs.expo.io/versions/latest/