Milestone 4

Title:

SmartStride: Toe-Walking Rehab

Names & Emails:

Cianna Grummer cgrummer2019@fit.edu

Alec Anzalone aanzalone2021@fit.edu

Kiera Ceely kceely2021@fit.edu

Bela Perdomo iperdomo2021@fit.edu

Caleb Phillips cphillips2021@fit.edu

Faculty Advisor:

Dr. Gu gul@fit.edu

Progress of Milestone 4:

Task	Progress	To Do
Setup AWS's IoT or S3	50%	N/A
Connect Raspberry Pi to lambda function	100%	N/A
Create a new webpage for drag and drop	100%	N/A
Create drag and drop functionalities	100%	N/A
Connect device to website	33.33%	N/A

Discussion (Milestone 1):

- Setup AWS's IoT or S3:
  - The purpose of the S3 Bucket was to catch the CSV data from the raspberry pi, then hold the data till the lambda function can parse the data to input it into the database. The S3 is also for holding large amounts of data from multiple raspberry pies to make sure no data is lost during uploading or processing. The cost of uploading and holding data in the S3 bucket exceeds the budget of the project because of this the S3 Bucket was not implemented into the system but will be included in the future plans part of the project upon completion. All necessary functionalities including uploading and parsing of the CVS data can be handled on a smaller scale by the API and lambda functions which are ample enough for proof of concept and small-scale testing.
- Connect Raspberry Pi to lambda function:
  - The process of connecting the raspberry pi to the lambda function was easier than expected due to the fact that the S3 Bucket was left out due to

pricing factors. Excluding the S3 bucket means that the Pi could directly connect to the API that handles the Drag & Drop functionalities utilizing the same lambda functions which simplified the amount of work needed. The API and Lambda function used for uploading data through the Pi and for the Drag and Drop functions are talked about more in depth under the *Drag and drop functionalities* section of this report. The Raspberry Pi functionality was expanded to create an app for a user-friendly upload. This simple app was completed in python using tkinker to create the GUI for the user to interact with. Upon opening the app, the user is prompted to enter their username and click the button to start monitoring. The app monitors any new files uploaded into the app's data folder. This folder will be where the data from the ML is saved upon completion of the signal processing. When a new file is found to upload the app will display a massage to the user: "New file found 'Name of CSV'" followed by "Uploading 'Name of CSV'" after a few seconds a new message will be displayed with either "Successful Upload of file 'Name of CSV'" or "Error: 'Error code/ error message'". The response is dependent on the lambda response returned through the API.

- Webpage for drag and drop:
    - The webpage for drag and drop is accessible through a patient's dashboard by clicking the "Upload CSV File" button and the webpage will redirect the user to a screen with a title of "Upload Files". From here the user can drag and drop the csv file into the upload box, click the upload box and file explorer will appear on the screen prompting the user to choose a file to upload, or a back to dashboard button can be clicked allowing the user to return to the patient dashboard. Once the CSV file has been selected through either method a bar will appear under the upload box with the name of the file on the left and a upload loading bar on the right with the word "Uploading…". Once the file has been successfully uploaded the loading bar will be replaced with the size of the file with units and a green checkmark with the word "Uploaded" in green. If the upload was unsuccessful a red X will appear with the word "Error" followed by the error message from the lambda function.
- Drag and drop functionalities:
    - The drag and drop API handler handles both the drag and drop and raspberry pi uploads of CSV data. The API will send the data along with the username of the patient to the lambda function that handles the drag and drop and the raspberry pi uploads as well. The lambda from here will check that the username of the patient exists and then will check if that patient already has a session of data uploaded and will check what number that session is. If the patient has no session of data uploaded the lambda will upload the data as session one. If the patient has session data uploaded already it will check the session ID number of that data and increment it

by one to upload the data into the database underneath the name of the patient. Upon successful or unsuccessful completion, the lambda will return a response code through the API to be displayed to either the drag and drop upload or the Raspberry Pi app.

- Connect device to website:
    - Full testing of the device to the website upload was unable to be completed due to the breaking of the IMU sensors on the device. The IMUs are imperative to this step because they are the sensors that collect the data from the patient that is then used by the machine learning aspect of the project that outputs upload able data to the data folder in the raspberry pi app that accesses the API of the website to send the data to the lambda function to be uploaded to the database. This is part of the task list was left at 33.33% completion because there are three main parts of connecting the device to the website. The first is successfully collecting data from the sensors on the sock and uploading it to the raspberry pi. The second part is the machine learning completing the signal processing of the uploaded data and the third part is uploading the processed data to the database. The Machine Learning is not completed to the point of testing and the setback of the sensors breaking means we cannot currently collect data from the device. The upload from the Pi to the database is completed and successfully hence the 33.33% of completion.

Plans for Milestone 5:

Task	Progress	To Do
Update database structure	15%	Talk with group to see what data will be used
Define graphs	35%	Make necessary changes to HTML
Create pie chart	10%	
New page for past results	40%	Add the API calls to HTML and redirection links, refine looks
Create API and Lambda functionalities for pie charts	0%	Create Lambda and functions as well as API
Create API and Lambda functionalities for past results page	0%	Need to create lambda and API functions

Discussion (Future Milestones):

- Update database structure:
    - The database structure will need to be updated according to the data that is deemed helpful to be displayed. Currently the data that is being displayed is all IMU data. After signal processing is completed much of these data points will be

rendered useless and will not be included in the CSV file uploads. Since the uploaded files are going to be changed the data that is being held in the database will need to change to reflect this. Currently I am in talks with my group to try and narrow down the exact data that will be used to create a more accurate database.

- Define graphs:
    - The graphs that are shown to the patient and to the practitioners will need to be changed. The current graphs are not helpful in the rehabilitation process of the patients. These changes include:
        - Rename EMG Analysis to "Gastrocnemius EMG Activity"
        - Remove "Gait Analysis with step classification" tile
        - Remove figures from "Last PT Session Details"
        - Pie Chart Visible to Patient and Doctor
        - Pie Chart will go under "Last PT Session Details"
        - Will put average ITW foot angle under "Last PT Session Details" (doctor)
        - The time series plot showing averages over 1 gait cycle for ITW vs Normal Step (x = time, y = angle (degrees) (doctor)
- Create pie chart:
    - The pie chart will be a percentage of all steps taken with differing severities. The whole of the pie chart will be all steps taken and the different sections will be Normal, Mild, Severe, and Extremely Severe. Note that the titles of these sections will change after discussions with my group. The severity of the steps will be determined during the machine learning signal processing stage that occurs on the raspberry pi. These values will be taken and compared to the total steps taken to determine the separation of the pie chart.
- New page for past results:
    - When opening a patient's page, a user will see the current months data that has been collected but cannot see past results. To amend this issue a button will be added to the patient dashboard to redirect the user to a past results page. This page will contain all the months of data from that user that has been collected in the past. This page will also contain the current month's data.
- Create API and Lambda functionalities for pie charts:
    - A new API and Lamba function will need to be created to handle the displaying and creating of the pie charts for the patient dashboard. The lambda function will hold the code to collect the data from the current month including total steps and number of steps that were mild, severe etc. Once this data is collected the percentage of each section can be calculated and sent back to the webpage through the API and displayed through CSS, HTML, and JavaScript.
- Create API and Lambda functionalities for past results page
    - The past results page will need a new API and lambda function to pull all the necessary data separated by the months they were collected. After this data is pulled the lambda will send the data through the API to the HTML code where a

combination of HTML, CSS and JavaScript will be used to organize and display the data to the user.

Meeting Dates:

Every Tuesday and Thursday 11am-12pm

Every Friday 12pm-1pm

Client Feedback:

- The IMUs are broken and are currently in the process of being fixed.
- The ML has had a few issues slowing down the development.

Advisor Meetings:

Every Tuesday and Thursday 11am-12pm

Evaluation by Faculty Advisor:

Task for Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Cianna Grummer	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____