Milestone 5

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 5:

Task	Progress	To Do
Update database structure	50%	N/A
Define sections	100%	N/A
Create pie chart	100%	N/A
New page for past results	100%	N/A
Create API and Lambda functionalities for pie charts	100%	N/A
Create API and Lambda functionalities for past results page	100%	N/A

Discussion (Milestone 5):

- Update Database Structure:
    - The database structure needs to be updated to hold the information that the Machine Learning will produce and send to the website as CSV files. Currently Alec, the team member working on the ML is still working on producing a model that can identify idiopathic toe walking (ITW) steps and unsure of what data will be used and unsure of what the data the ML will produce.
    - The database was updated with a new table to hold the data for the pie chart. The table is as follows: id, patient_id, session_id, session_date, Normal, Mild, Moderate, Severe, Total. Normal, mild, moderate, and severe hold the number of steps that were taken that fall under the conditions of normal (no toe walking), mild (slight toe walking), etc. patient_id holds the foreign key to what patient the data belongs to.

Session_id holds the identifier for what session this data belongs to in patient_session_data. The total is the total number of steps taken during the session and the date is important because it is what allows the pie chart to be sorted by month.

- Define Sections:
  - The group requested that the sections on the patient dashboard page be changed and moved around. The changes include renaming EMG Analysis to “Gastrocnemius EMG Activity”, Removing the Gait Analysis with Step Classification section, replace the graph under Last PT Session Details with the pie chart, create a new section for doctors only that holds average ITW foot angle over one gait cycle graph. This means that the only things that are currently visible to a patient are the patient information, clinician information, treatment goals, progress graph, monthly session details, and severity assessment bar.

- Add a Pie Chart:
  - A pie chart was added to hold the percentage of steps taken during all the sessions for the current months separated by the number of steps taken that were severe, moderate, mild, and normal. The pie chart is under the Monthly Session Details section and is visible to both doctors and patients. To create the pie chart, I first have to add the new table to the database to hold the data used in the pie chart. The details of the table are discussed in the Update Database Structure section of this report. The Lambda function DragDropHandler was also edited to be able to upload the new format of CSV to the table from the website and from the Raspberry Pi. A new javaScript method was made to create the pie chart called fetchAndDisplayPieChartData(). When a user logs into their profile the login function sends two pieces of information to the patient or clinician dashboard html files. In this case it is the patient dashboard that receives the username and user type (patient or clinician). The first thing that fetchAndDisplayPieChartData() does it parse the userData to retrieve the username and user type. Then the current date is retrieved which is then used to retrieve all sessions of data for that username within the current month by using the API created for the pie chart. Once the data is collected the JSON body that is returned by the API is parsed. After the data is parsed the loading elements are deleted and the string of numbers are converted to the Number data type and the createPieChart() method is called. createPieChart() is where the actual pie chart is made starting with assigning the colors to the normal, mild etc. colors. Then the percentage for each step type is calculated and finally the actual pie chart is created.

- New Page for Past Results:
  - The past results page is to look at the pie charts by month to see your progress over time. The page can be accessed by the “See Past Results” button located underneath the key for the pie chart. After being redirected to the new page all the pie charts of the patient are displayed. The results can be filtered to look at only a section in time. When clicking the calendar button for the start or end dates a drop-down menu appears that is all months within a year, and you can scroll up or down to change the year. After a user selects their timeline, they can click apply filters. This will trigger a change in the months’ pie charts that are displayed. The user can then click the “Reset Filters” button which will remove the filters they selected, and they can once again view all of their results. The user also has the option to click the “Back to Dashboard” button at anytime redirecting the user back to the main profile page. The filters utilize the same API that is used on the main page to retrieve the pie chart data from the database and the same process is used to create the pie charts with the only change being that createPieChart() is called for every month that contains data.

- Create API and Lambda Functions for Pie Charts:
  - The Lambda and API handle retrieving the data that is used to create the pie charts. The API GetPieChartData is a POST method that is used as a trigger for the GetPieData Lambda function. The Lambda function first parses through the body that was sent through the API getting the username, month, and year identifiers. Next Lambda connects to the database and after a successful connection queries the database to retrieve the pie chart data for that user during the month and year that was sent. After receiving the data from the database, the lambda organizes the data into a JSON format to be sent back to the API and then to the HTML file as a JSON response body.
  - The DragDropHandler Lambda function needed to be updated to allow the CSV files containing the pie chart data to be uploaded to the database through either the drag and drop page or from the Raspberry Pi. I had some trouble with this because the format of the session CSV data is different from the pie chart CSV data. One way I could have solved this problem was by creating two separate lambdas and having the user select which kind of data they were uploading to the database but the user might not know what kind of data they have so I decided to keep it as one API and Lambda function for ease of use. The way the Lambda was edited to handle both types of CSVs is when a new file is requesting to be uploaded

the lambda function checks if there are only two rows of data or more because the pie chart data will always be two rows of data and the session data will always be more than two rows of data. If there is more than two rows the Lambda continues exactly how it worked previously but if there is only two rows the Lambda will check the new table for the last session number, parse through the data using the new format, and upload the data into the new table.

- Create API and Lambda Functions for Past Results:
  - Before I started working on the past results page, I thought I was going to make a new lambda function to handle pulling multiple months of pie chart data at once but after I completed the first pie chart API and Lambda functions I realized that I could just call the same API/ Lambda function multiple times to retrieve the information to create multiple pie charts. This also helps with the filtering system because I can vary the number of months being called easily.

Plans for Milestone 6:

Task	Progress	To Do
User and/or Developer Manual	0%	Create user and/or dev manual
Demo Video	0%	Start creating a demo video
SmartStride Poster	10%	Complete my section in the SmartStride poster
Update User Guide	50%	Talk to SmartStride team to create a user guide for device
Update About Us Page & Home Page Description	30%	Work with SmartStride to create a more accurate description
System Testing & Evaluation	15%	Create a user survey, find testers, conduct login and data testing

Discussion (Milestone 6):

- User and/or Developer Manual:
  - The User Manual will be for two kinds of users: patients and clinicians. Both patients and clinicians will share certain sections of the manual like how to sign up, log in, find the device guide etc. The patients will have a

  - specific section of the manual for using the app on the Raspberry Pi, how to manually upload their data, and how to read the pie chart and severity bar. The clinicians will have their own sections within the manual explaining how to add or drop a patient, how to read the clinician only data, how to view a specific patients' data, how to add or remove goals from specific patients etc.
  - The developer manual will contain diagrams of the website structure with webpage explanations, database structure with table descriptions, as well as a diagram for how the API, Lambda, Amplify, RDS, CloudWatch, IAM Roles, VCP, and security groups work together. It will also contain a description of how to connect the developer's device to the database so they can access the database.

- Demo Video:
  - The demo video will showcase the key features of the website including the sign up, login, uploading data, patient and clinician dashboards, past results, the Raspberry Pi application, how the data that is uploaded changes the figures featured on the patient dashboard in real time.

- SmartStride Poster:
  - Since I joined the SmartStride team I will have a limited section of the poster to explain/ show my goals, features, design, evaluation results and limitations I experience while working on the project. I will try to be as concise as possible in order to ensure that all the information will be included on the poster. I will also be in discussions with my SmartStride group to see if I will have one small section of the poster or if my project descriptions will be combined with the descriptions of the SmartStride device.

- Update User Guide:
  - Currently the user guide page exits and has very minimal and outdated information that has changed. I will be working closely with my group to create a more comprehensive guide on how to use the SmartStride device along with how to use the Raspberry Pi and links to my own User Manual.

- Update About Us and Home Page Descriptions:
  - The About Us and Home Page Descriptions are currently outdated and do not include important information about collaborations and device purposes. It does not capture an accurate description of the website or the

SmartStride project and because of this will all need to be rewritten in order to accurately reflect our goals of the device.

- System Testing and Evaluation:
  - The Systems testing and evaluation will be completed in three parts: Usability, Reliability, and Accuracy. This is how I will be evaluating if the system is successful or not.
  - To test reliability, I will be testing the data upload through the drag and drop handler Lambda function to see if the data is lost or not uploaded successfully the evaluation will be the comparison of the total attempts and successful attempts to produce a rate of success in percentage form. Since this will be sensitive patient data when in production a success rate of 99% is ideal, 95% - 98% is acceptable but could use improvement if possible, and below 95% percent would be unsuccessful and result in the system needing many improvements. I will also be testing the login functionality to ensure the website is functioning properly as this reflects usability. This again will be the comparison of the total attempts and successful attempts to get a percentage of success. 99% or higher is always ideal but an acceptable range would be 90%-98% while anything lower would be unacceptable.
  - To test the accuracy of the system we will using the tests that are described in the reliability section but we will be comparing the data that was uploaded with the original data to ensure that the data is accurate to what was supposed to be uploaded we again will compare the total number of data sets that were uploaded to the number of datasets that were uploaded accurately to find a success rate. This should be 98% or higher to be considered successful.
  - Usability will be determined by a user survey after the user has performed several tasks such as sign up, log in, find the user help page etc. This will help determine if the website is friendly to new users and can help point out flaws in the architecture of the website if something is consistently hard to find. The user survey will contain questions such as “How easy was it to find the sign-up page? * * * * *” five stars being no problem and one star being impossible. It will also contain a section at the bottom of the survey for additional comments where the user can provide feedback if they would like. The star rating of each task will be averaged with four stars and above being acceptable, three and two stars need work starting with two-star items and one star being unacceptable.

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.
- Documentation is currently being completed

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: __________