LiftOff

Authors: Sydney Dunn, Jasmine Yew, and Zhichun Zhao
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—Weightlifting involves many techniques and having bad form can lead to injury, especially with heavy compound lifts, such as deadlifting. To help people improve their deadlifting and squatting techniques, we aim to create a device that detects back orientation and spine curvature, along with a separate device with weight detection. Our devices will be connected to a mobile app that provides feedback on how well users performed the exercise and how they can improve.**

***Index Terms*—Design, lifting, sensors, web app**

1 INTRODUCTION

Oftentimes, people are apprehensive about lifting weights because they are unsure of proper techniques and are afraid of hurting themselves, especially when lifting heavy weights. Even amongst veteran weightlifters, having the reassurance that they are doing an exercise properly, especially with heavy weight or one they have never done before, can increase their confidence and improve their performance with their lifts. Not only do we want to help people improve their lifting techniques, but our project also aims to help reduce injury with two of the main lifts – dead-lifting and squatting.

Our project consists of a wearable device that will track the user's back orientation as well as relative weight, and report it back to our web application. The wearable device will act similarly to a harness with sensors on it. The weight-tracking device will sit in the shoe, and be accompanied by an ankle strap. On the web application, users will be able to see feedback on their forms and track their progress.

Current solutions a gym goer might use to identify problems with their lift include setting up cameras to film themselves, or hiring a personal trainer to offer them coaching. While we considered a computer-vision-based approach, many gym goers might wear bulky clothes that inhibit our ability to analyze it, and the setup angle becomes critically important in being able to detect an arch. With a physical sensing system, we are able to offer more customized and potentially discrete support, as it can be worn under clothing. We also will offer this qualitative feedback at a much lower cost, than that of a trainer, and users analyzing their own lifts using a camera, might miss or lack the knowledge required to fully understand if they are doing something wrong, and potential improvements.

2 USE-CASE REQUIREMENTS

The primary user we will be considering is someone interested in learning how to deadlift or squat, that does not have significant prior experience. For this user, we have identified the following use case requirements. First, the system must be able to detect the user's back orientation. As a subset of this, the system must also be able to detect the presence of a lower back arch. The system must classify these back orientations i.e. (upright, slanted at 80 degrees, slanted at 60 degrees) correctly. In turn, the user wants to be able to track their progress with ease, so they desire an automated lifted weight detection. This will then be recorded in the web application, to allow them to track their workout. As a part of this, the user will want to be able to track their progress over time. This might come in the form of a graph illustrating the weight they lifted over a particular time period for a certain lift. Additionally, the user wants correct and timely feedback on whether or not they performed their lift correctly. It is imperative to the user that the feedback be correct, and not result in them having a worse form than before.

Outside of these basic requirements with regards to accuracy and tracking, we must also consider the convenience of our product to the typical gym goer. Most gym goers normally spend a fixed amount of time in the gym(e.g. 1 hour) so our device setup time must not significantly eat into their workout time. Therefore, we require an easy and simple setup, ideally taking less than 2 minutes. Also, the user would not want to have to recharge their device mid workout, or have to worry about turning it on and off between sets to save battery. Therefore the device must have enough power to last the entire workout. It must also be rechargeable as replacing batteries is both bad for the environment and annoying for the user. The device itself should also be unobtrusive, so as not to hinder or impair the user's typical range of motion, or otherwise negatively influence their lifting form. Furthermore, as the user will be actively working out, they likely will be sweating a lot, and therefore will want the device they are using to be sweat and water-resistant.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

As illustrated in Fig. 1, our system has both a software and a hardware component. One of the devices will be on an ankle strap with a sensor worn under the foot that will detect the amount of pressure being placed on it. The other device will be the back device that the user can strap onto

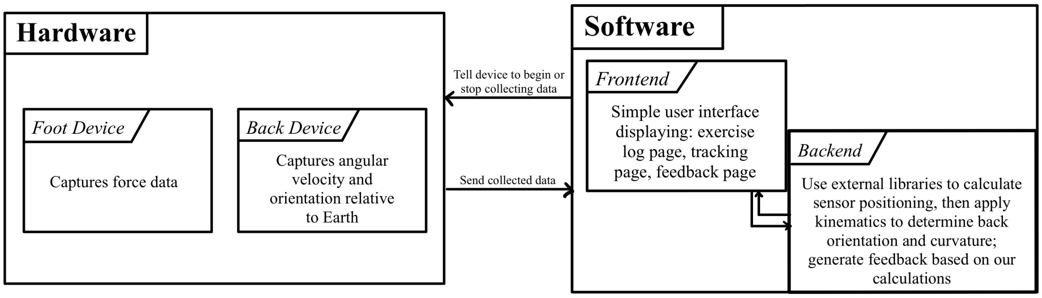

Figure 1: The overall architecture of our system

their upper body. This back device will consist of several gyroscopes and accelerometers that will capture angular velocity and the user's back orientation relative to Earth. This data that is read by our devices will then be sent to the software via Wi-Fi where data will be processed.

In the backend, we will use Quartenion libraries to convert the data into useful vector data that we will afterward use to detect back orientation and curvature. Our web app will be created using Django. Therefore, the frontend will consist of HTML/CSS templates, while the backend consists of Python code to generate the webapp.

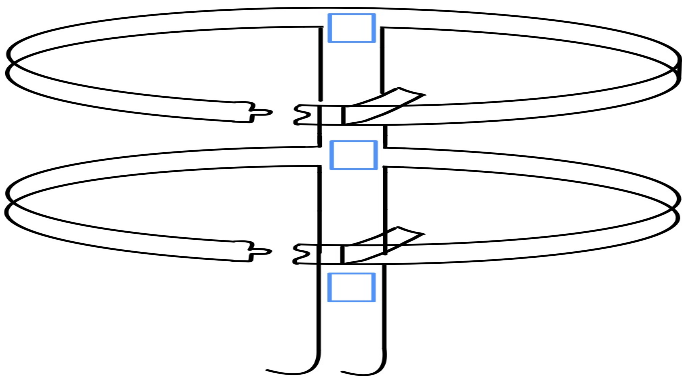

Figure 2: A hand-drawn rendering of the harness for the back orientation sensing unit

In figure 2, we illustrate the apparatus by which our back orientation sensing unit will be placed on the user. Essentially, we imagine it a bit like a backpack, the user will be able to fasten the two shown straps onto their chest. Then, at the bottom, they will be able to hook it into their pants and/or lifting belt if they so desire. Through this form factor, we hope to barely impact the range of motion, as it places no restriction on the arms, as well as allows for maximal adjustability via the straps. The three blue squares indicate the positioning of the sensors. Their relative position will be adjustable, such that the user can make sure they are placed at approximately the base of their neck(the C-spine), their mid back(the T-pine), and their low back(the L-spine).

In order to sense the weight our user is lifting,we will place a sensor inside of the shoe as illustrated in figure 3. The force-sensing resistor, demarcated as A301 will be placed inside the sole of the shoe. Then wires will run up the back of the shoe, to the strap around the user's ankle. This will be a velcro strap that contains an Arduino and battery to allow it to send data to the web application for processing.

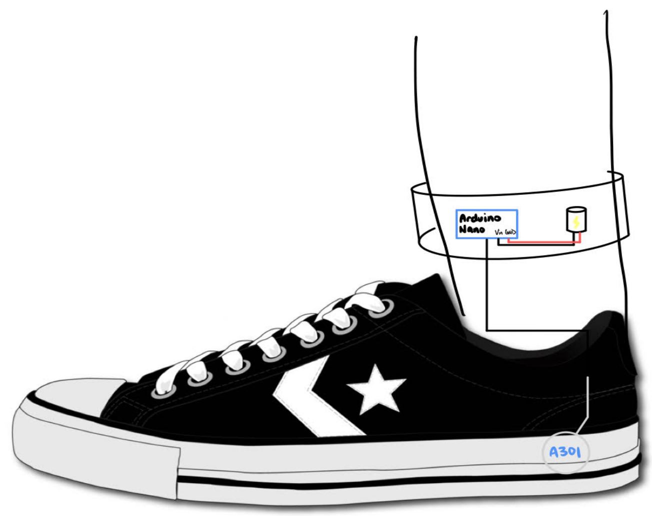

Figure 3: A hand-drawn rendering of the ankle strap for the weight detection unit

4 DESIGN REQUIREMENTS

To accurately measure if our product meets the user case requirement, we created a quantified requirement for each use case requirement. First, we considered what an acceptable delay in producing our feedback is. For our users, we decided that real-time feedback is not desirable, as that might result in them getting distracted when they see that they are doing it wrong, potentially resulting in them injuring themselves. Thus, we decided that the system must display the result at the end of a set, which we estimated to be about 7 seconds, or approximately the time required to squat or deadlift for five repetitions. Additionally, for our users, the correctness of our system is of the utmost importance. Therefore, we established a required 90% to ensure accuracy in our detection of back orientation and lower back arch. In more concrete terms, this means classifying the angle of the back relative to the ground(i.e. standing upright is 0 degrees, while being completely hinged at the hip is 90 degrees). This information must be accompanied by a time signature, such that the user could potentially see a replay of their exact squat or deadlift movement. Taking on a larger view, our system must be able to take these lift signatures and classify them into categories of good or several different versions of "bad" such that when a "bad" form is detected, the user will receive an appropriate piece of feedback. Again, as it is imperative we don't make life worse for the user, this must happen with 90% accuracy.

Furthermore, as discussed in our use case requirements, the setup and ergonomics of our system must be unobtrusive to the user. In quantifying this, we identified that it must take approximately the length of a rest period, or less than 2 minutes to set up. The form factor for the primary computer must be able to fit either on a weight lifting belt, or clip onto their clothes, meaning it should be less than 2.5 inches. The sensors and components having direct contact with users' bodies must be water and sweat resistant. To build in buffer time, the battery life needs to be at least 3 hours long such that it can last the average gym session (2 hours) and then some. Finally, the data tracking and logging function must count the number of reps completed with an error margin of +/- 1, and be able to record the amount of weight used in either kg or lb. This record of the weight used must be accurate within 5lb, as 2.5lb is the typical smallest weight increment available at gyms.

5 DESIGN TRADE STUDIES

In coming up with our design, we considered many other potential options. Below, we will discuss why we chose to use a Web Application rather than a mobile app, as well as our choice of using a wearable device over computer vision.

5.1 User Feedback Interface Design Study

As shown in Figure 7 which matches the block diagram described in Figure 6, we plan to use a web application instead of a mobile application as our user interface. The reason why we prefer using a web application is it is more versatile than a mobile app, as it can be used on more devices such as iPads and laptops. On top of that, our team has prior experience in developing web applications, whereas mobile app development is an entirely new field to the team.

5.2 Detection Method Design Study

For the wearable part of our system, we plan to use three sensors along the spine line instead of using computer vision to detect the form of users' workouts. The reason behind this decision is that sensors maintain higher accuracy than computer vision particularly when users are wearing bulky clothes in cold weather. Besides, it is more convenient for users to set up a wearable device as they work out in the gym so they will not have a constrained location for different workouts or move the camera with them during their training period.

In deciding which sensors to use for our project, we investigated a couple of different options. We looked most closely at whether or not a 9DoF IMU or a 6DoF IMU would be acceptable for our purposes. Essentially, a 9DoF IMU contains a magnetometer as well as the gyroscope and accelerometer found in a 6DoF IMU. Essentially, the magnetometer serves to correct angle drift caused by the Earth's magnetic field. As in our system, accuracy is of the utmost importance, and a couple of degree difference could mean classifying something as correct, rather than incorrect, we decided it would be better to err on the side of caution, and not attempt to correct this via math, but instead via direct sensing.

Another consideration is how to attach the sensors to the user. One option is to build them into a piece of clothing, have them individually strapped, or the harness like structure that we ultimately settled on. We ruled

6 SYSTEM IMPLEMENTATION

6.1 Back Orientation and Arch Detection Unit

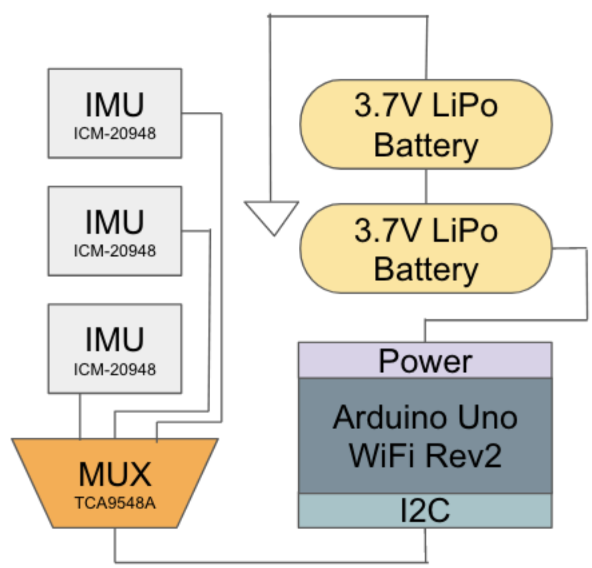

Figure 4: A block diagram of the back sensing unit, consisting of 3 Inertial Measurement Units(IMUs)

The back sensing unit consists of 3 IMUs which allow us to learn the angular velocity of the sensor relative to the earth's gravity and orientation. These three units are then piped into a MUX, as they use the I2C protocol, but unfortunately, the chosen sensors are hardwired to only support 2 I2C addresses, so a MUX is needed to allow support for 3 units. These are then connected to an Arduino Uno Wifi Rev2 unit. Also to reduce computations needed at the web server, the intention is to have the sensing units convert the raw data into quaternions, then forward those quaternions via the Arduino's wifi to our AWS EC2 instance. Additionally, as the system must be battery powered, and as the Arduino requires at least 5V of input voltage, we have decided to connect 2 3.7V Rechargeable LiPo batteries in series.

6.2 Weight Sensing Unit

The weight sensing unit will consist of 1 Force Sensitive Resistor, that can measure up to 4448N or 1000lb of force. To allow for sensing up to this range, we will build the op-amp circuit drawn above, which allows us to measure a wider range of voltage drops (and thus force) across the resistor. Additionally, to communicate with the main device, this will be connected to an Arduino Nano IoT which will then connect to the Arduino Wifi Rev2, where the data will be compiled and sent to the web server. Similar to the Back Unit, the Weight Sensing Unit also requires 2 3.7V LiPo rechargeable batteries connected in series to deliver the required voltage.

6.3 Web Application

The web application will be created using Django and deployed onto AWS EC2. As shown in Fig. 7, the frontend of our web app has multiple pages: login and registration page, exercise log page, set tracking page, feedback page, and user settings page. To create a good user interface, we will be incorporating Bootstrap into our HTML/CSS templates.

For the implementation of the login and registration page, we will use Django's Forms and Models to securely store users' usernames and passwords. On the exercise log page, we will display the user's previous records of exercises that we have stored in our MySQL server. When the user clicks on their previously done exercises, we will display the data we gathered from that time. Using Django's ORM, we can create and store Models of workouts when we are on the set tracking page to add the data to the database. On the tracking page, the user presses a start button on the app when they begin their set. In Fig. 6, we show the steps that will be taken to generate the feedback after the user begins their set.

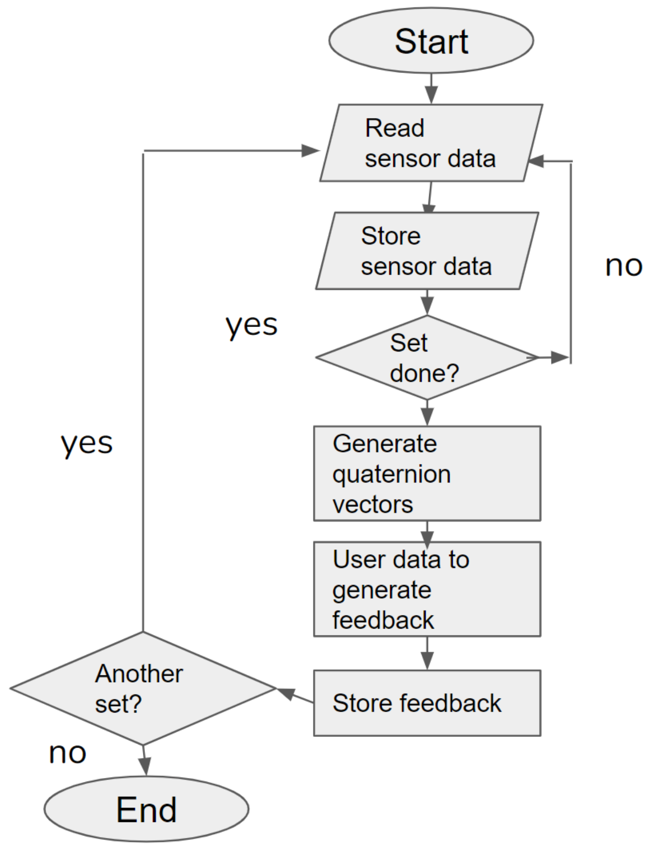

Figure 6: A flow chart of how we will process users' data from the sensors to generate feedback

In the backend implementation of the feedback page, there will be different switch cases for feedback generation depending on the vector data. For example, if the vectors show a bend where one vector's direction is up and the adjacent vector's direction is down, we may map this scenario to feedback that tells the user their spine should be kept neutral. In addition, we will have links to videos explaining how to do the exercise properly on the page.

7 TEST & VALIDATION

7.1 Tests for Back Orientation and Arch Detection

To test the back orientation and arch detection, we plan to hold back in different positions and check if the data produced by sensors is consistent. We will do the above procedures with various sensor placements on the back, and with people of different heights.

7.2 Tests for Weight Detection

We will test the weight detection's accuracy by running multiple trials where we will be comparing the data generated by the system with the actual weight of the user

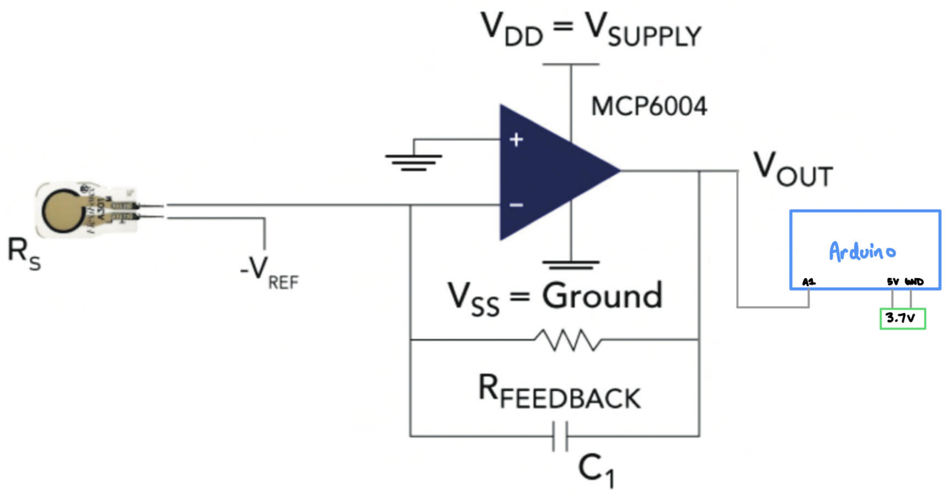

Figure 5: A circuit diagram of our weight sensing unit

plus the weight they are holding. The difference should be within plus or minus ten percent of the actual weight.

7.3 Tests for Latency

To test the latency, we would time how long it takes for the system to generate feedback for multiple trials. The time it takes should be less than 7 seconds, which is about the time for users to do a squat plus 2 seconds of buffer time.

7.4 Tests for Feedback Generation

To test if the system gives informative and actionable feedback, we plan to provide our own data to ensure conditional coverage such that the feedback will match the data received. For example, we will do a proper squat and a wrong squat and compare to see if the feedback generated for the users is different and correct.

7.5 Tests for Data Tracking and Logging

To ensure data is logged correctly, we plan to use branch coverage to ensure that when we are counting reps or tracking time, everything is correctly updated. We will do this by passing in several sets of test data, then asserting that the values stay consistent with what we expected.

7.6 Tests for User Experience

To check if our system is easy and straightforward enough to set up, we plan to recruit people who are unfamiliar with our project to follow the setup instructions and time how long it takes.

7.7 Tests for Additional Features

We plan to use different testing plans for use case requirements. To fulfill the water and sweat-resistant requirement, we plan to do research on the data table provided by the manufacturer of the parts. We will also buy sweat-resistant sensors and test them to see if the reading stays consistent through the same motion when they are in contact with sweat. To test the battery life and time between charging, we plan to power our system on and wait until it dies. We will record the time it takes to exhaust the battery.

8 PROJECT MANAGEMENT

8.1 Schedule

The schedule is shown in Fig. 8. We are currently on track to finish our project by April 20th.

8.2 Team Member Responsibilities

Our responsibilities for this project are split into three categories: hardware, web app, and algorithms. We partitioned the work and assigned different categories for each person depending on their area of expertise. Sydney will be in charge of the hardware due to her experience from taking multiple hardware classes. Therefore, she will be building and connecting the hardware devices together and implementing data processing to read the data from the devices. Since Rachel has experience with web app development, she will be primarily in charge of web app development and will be developing and deploying our web app. Jasmine will work on the algorithms that determine back orientation, back curvature, and weight carried. Since the algorithms rely on the devices and web app being set up, Jasmine will also assist with the hardware and web development since

Table 1: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Inertial Measurement Unit	ICM-20948	SparkFun/InvenSense	3	$18.50	$55.50
Multiplexer	TCA9548A	SparkFun/TexasInstruments	1	$12.95	$12.95
Wifi Enabled Microcontroller	Uno Wifi Rev 2	Arduino	1	$53.80	$53.80
Wifi Enabled Microcontroller	Nano 33 IoT	Arduino	2	$24.00	$48.00
Cables	Qwicc Connect Kit	SparkFun	1	$8.95	$8.95
Force Sensitive Resistor	A301	Tekscan	4	$15.44	$61.79
3.7V LiPo Battery	LIPO785060	PKCELL	4	$14.95	59.80
LiPo Battery Charger	259	AdaFruit	1	$12.50	$12.50
Quick Release Buckle	B07W64DBL8	MAGROW	1	$7.65	$7.65
Hook and Loop Strap	B09XMVVRH8	YMCRLUX	1	$12.99	$12.99
					$333.93

she has experience with both embedded systems and web app development.

8.3 Bill of Materials and Budget

Please see Table 1 for a table of our purchased parts.

8.4 Risk Mitigation Plans

Initially, we planned to create a mobile app. However, this posed a significant risk because none of us had experience with mobile app development. We ultimately mitigated this risk by changing to web app development. Currently, our main risks are:

- Reading proper data from our sensor devices
- Being able to connect our device to our web app
- The wearable back device not sitting properly on the user's back

To mitigate these risks, we have and are currently still doing heavy research on how the parts interact. We have also assigned two people in our team, Sydney and Jasmine, to work on data processing. With two people working on reading data, we are more likely to ideate a proper implementation that reads useful data from our devices and can conduct more tests. Additionally, if our Arduino is unable to connect to our web app, we may pivot and use a Raspberry Pi. To mitigate the risk of the back device not laying properly on the person, we hope to use adjustable straps with buckles. We may also use velcro if the sensors do not properly attach to the user's back.

9 RELATED WORK

There are currently no exercise form-checking devices that are available on the market or have been documented. However, there exists several software and apps, such as FormCheck, that incorporate machine learning to check if the user properly executes a rep of any exercise. FormCheck uses computer vision and 3D motion tracking to read users' movements and machine learning to interpret their data. By reading data from the users' cameras, they are able to detect human limbs and their movement.

10 SUMMARY

Our system is intended to act like a training wheel gently guiding people into different workout forms. The challenges that could be present in the process of developing this system is to accurately detect, evaluate, and report the data that the users input as they do a repetition of a workout movement. The accuracy of the system will need to be thoroughly tested and could be limited by the hardware and datasets available to the team.

Glossary of Acronyms

- AWS - Amazon Web Services
- IMU - Inertial Measurement Units
- I2C - Inter-Integrated Circuit
- MUX - Multiplexer
- ORM - Object Relational Mapping

References

1. *FlexiForce Standard Model A301 Datasheet.* Tekscan. https://www.tekscan.com/resources/product/flexiforce-a301-datasheet

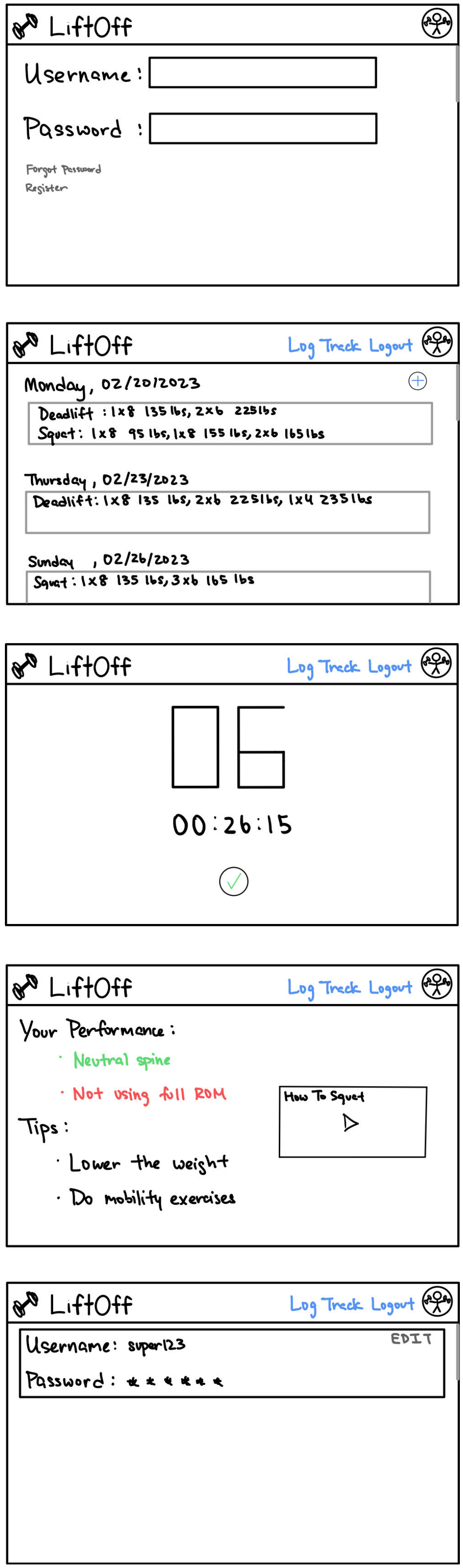

Figure 7: Web App Wireframe

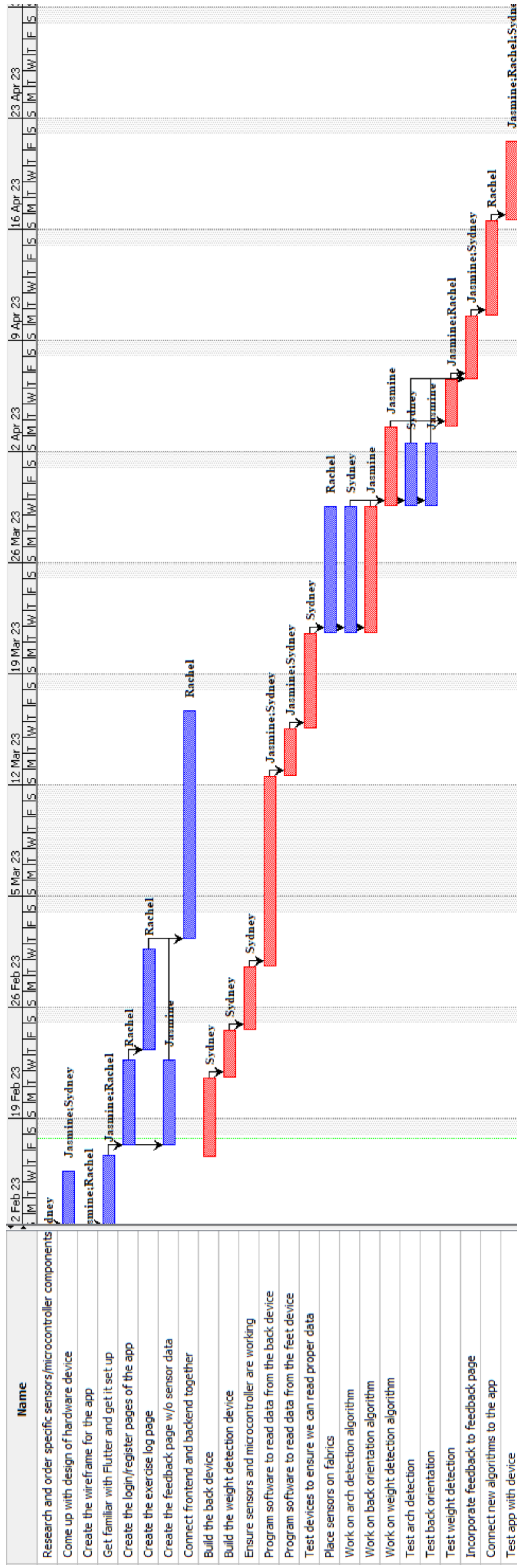

Figure 8: Gantt Chart