**SETSTATS**

A picture containing antenna

Description automatically generated

Mark Byrne Richard Collins Liam Denning Florian Scheunert

**Table of Contents**

Contents

[Team Members 3](#_Toc89021606)

[What is SetStats? 3](#_Toc89021607)

[What are our goals? 3](#_Toc89021608)

[Github Link: 3](#_Toc89021609)

[Hardware Required 4](#_Toc89021610)

[What we did - 5](#_Toc89021611)

[Data the device will be collecting: 6](#_Toc89021612)

[How the data will be gathered: 6](#_Toc89021613)

[How we are going to be processing the data collected: 6](#_Toc89021614)

[How the device and data are going to be secured: 6](#_Toc89021615)

[Frequency with which each sensor can record a value and how the sensors work: 6](#_Toc89021616)

[Paper Prototype 9](#_Toc89021617)

[Paper prototype page descriptions: 10](#_Toc89021618)

[Connection: 11](#_Toc89021619)

[User Profiles 12](#_Toc89021620)

[Software-Testing 16](#_Toc89021621)

[Hardware-Testing 16](#_Toc89021622)

[User-Testing 16](#_Toc89021623)

**About Us**

# Team Members

**Mark Byrne: Liam Denning:**

A picture containing person, person, male

Description automatically generatedA person in a yellow shirt

Description automatically generated with low confidence**Role:** Database developer **Role:** Web Developer

**Richard Collins Florian Scheunert**

**Role:** IoT Developer **Role:** Tester/ Designer

A person in a suit and tie

Description automatically generated with medium confidenceA picture containing person, indoor, posing

Description automatically generated

# What is SetStats?

SetStats is an application created to help everybody improve their form in the gym. Whether you are a new user or a 20-year veteran, SetStats will surely help you. Trainers can use it to see how their trainees are doing and trainees can use it to find a trainer to watch them.

# What are our goals?

* We want to create a platform where you can track progress and improve your form.
* Beginners will have a platform to watch and learn from more experienced lifters through our tutorial section.
* SetStats hope to avoid injury by making sure lifters are not lifting too heavy or with incorrect form.
* Trainers should be able to see all of their trainees’ stats and with the help of SetStats they will have a better idea of how the trainee can improve their training

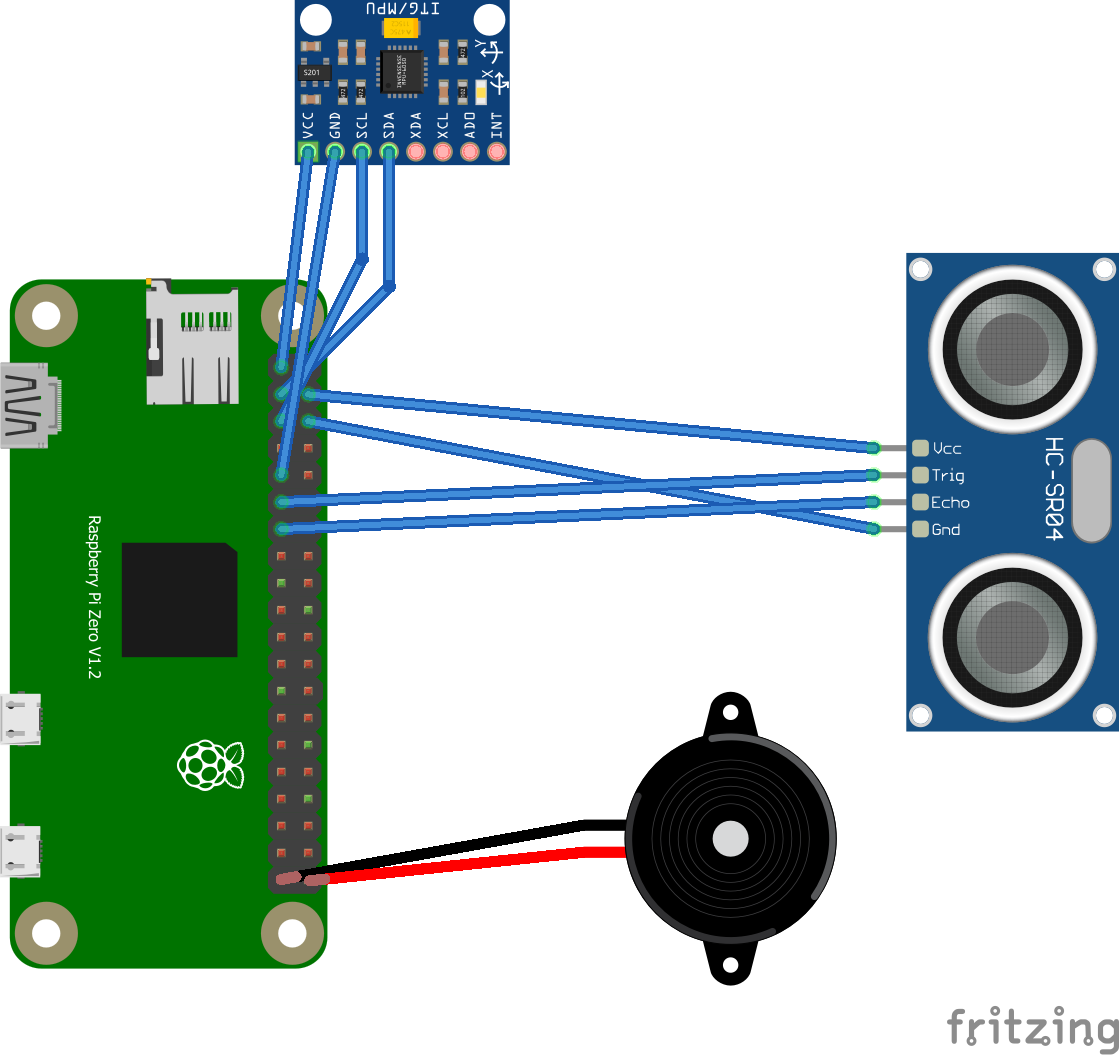
## Github Link:

<https://github.com/MarkByrneDKIT/SETSTATS->

**Hardware**

# Hardware Required

1. **Raspberry Pi Zero** – this will be used as it is the more compact option whilst also having low power consumption. It will collect, process and further send the sensor data to a database on the Amazon Web Services, where it will be stored for later use.
2. **Accelerometer (MPU-6050)** – used to measure the displacement on the x-axis (sway). This will be used to determine if the lift was considered bad and could possibly injure the user.
3. **Ultrasonic Sensor (HC-SR04)** – is being used to measure the distance on the y-axis (height) very accurately. This data can then be compared to the x-axis to pinpoint at what time the lift becomes bad.
4. **Peizo Buzzer –** This will be used to alert the user if the accelerometer’s data reads past our tolerance limit.
5. **Power Bank (5000 mAh)** – the power source which will be connected via micro-USB, it can be recharged and reused.
6. **Female to Female connectors –** the connectors will be used to connect each sensor to the raspberry pi.



**How we will power it and how we will connect it to the internet Is it mobile?**

The system will be connected to an external rechargeable battery source (Power Bank), which will give long lasting life due to the low power consumption of the Pi. It will be connected to the gym’s Wi-Fi or your phone’s hotspot. It would be mobile as long as you’re connected to a phone’s hotspot.

# What we did -

**Florian** Researched what modules would best suit our needs for the project, and found that an accelerometer would do the job. Florian was also responsible for buying and testing the piezo buzzer. Link - https://www.aliexpress.com/item/32819406470.html?spm=a2g0s.9042311.0.0.1b324c4dWNSvpP

**Mark** bought the accelerometer off Amazon and tested it out. He ran into a problem that we wouldn’t be able to track the movement of the bar on the x-axis(sway). Link -https://www.amazon.co.uk/gp/product/B00PIMRJX6/ref=ppx\_yo\_dt\_b\_asin\_title\_o05\_s00?ie=UTF8&psc=1

**Richard** found an alternative option, which would be by using an ultrasonic distance sensor in tandem with the accelerometer.

**Liam** then bought ultrasonic sensor to make sure it works by testing code from the internet. Link - https://www.amazon.co.uk/gp/product/B01DM8MRTS/ref=ppx\_yo\_dt\_b\_asin\_title\_o04\_s00?ie=UTF8&psc=1

**Richard** was responsible for buying and setting up the raspberry pi zero, he wrote code and connected the sensors to collect data at the same time and to display it for a prototype. Link - https://www.amazon.co.uk/gp/product/B08H1M94BK/ref=ppx\_yo\_dt\_b\_asin\_title\_o09\_s00?ie=UTF8&psc=1

**Data**

# Data the device will be collecting:

* Height of the bar (y co-ordinate)
* Sway of the bar (x co-ordinate)
* Time
* Number of reps and sets
* Highest point (in which bar was lifted)

# How the data will be gathered:

The data will be sent from the device to a database hosted on a server. Every 0.75 seconds the sensor picks up a change in distance/sway etc the data will be sent to the database and stored in a table for that lift.

# How we are going to be processing the data collected:

The data collected will be send to a database and then further pulled/displayed onto a website using chart.js. The data will be shown as a line graph.

# Frequency with which each sensor can record a value and how the sensors work:

Both sensors will collect data every 0.75 seconds.

**Accelerometer (MPU-6050) –** converts mechanical energy into electrical energy, it measures change in motion(acceleration)

**Ultrasonic Sensor (HC-SR04) –** Sends out a soundwave at a frequency of 40KHz and it travels through the air and if there is an object it will bounce back to the sensor.

# Data Security:

**Database**

Diagram, table

Description automatically generated

**Trainee Table:**

The trainee table contains all the information of the regular user who uses the application. They have their own id and linked to them is a trainer id, but they don’t necessarily have to have a trainer. Their password will be hashed for security. You will be able to access all the data for each user through their unique trainee id.

**Trainer Table:**

The trainer table contains all of the data of a trainer who uses the app. The trainer can have multiple trainees and we can use sql to grab them uses the foreign trainer\_id in the trainee table. With this you can then view the history of each trainee so the trainer will be able to view the progress of their trainee.

**Current Session Table:**

The lift details table will contain the data for a specific trainees current lift, the x and y will be updated a lot as the user moves the bar, while the rep, set and time will change on a slower pace but still quickly enough. The data will help us for the history page of the trainee so they can see when they trained.

**Current Lift Table:**

At the moment the xy is stored in geometry LineString which means a real-time graph is not possible right now.  The lift\_id is how we link the currents\_session table and we use the lift\_id to determine which lift it is in each rep. After each lift is completed the data is stored in xy.

**History Table:**

The history table will be used for the trainee to be able to look at their progress overtime by looking at previous sessions. This will be able to grab the users best lift of the session and the rest of the data like reps and sets and how long the session was. The trainer can use this data also to be able to observe their trainee’s improvements.

**Session Table:**

The session table is what stores the data that the history table grabs useful data from the current\_session and current\_lift table are stored here after there session is done.

# Database Implementation:

With the AWS ec2 instance, we use ubuntu which is where we can store the database using the MySql install. With this we also use the secure installation of Mysql so we can disable a feature like anonymous user and also we can create a root password that is hashed. We can have the choice to disable remote logins so that only on the device can we login to the MySql.

We import the database into MySql and call it "SetStats" with this we can access all the tables and the data inside of them.

Now on the python file for the flask app, we import flask\_mysqldb and install this onto the AWS ec2 instance. Using mysqldb we can write into the database once it is connected. We will be putting the queries into the flask app and inserting the data grabbed from the sensors into the database along with user registrations.

We can now also read data from the "SetStats" database so with this we can tell if a user is registered to the database so we are able to log them in, also we can now display the history of each trainee using SELECT statements and dates sorted by descending so we get the newest sessions first in the list.

# Database Security:

On the instance we use mySql for our database. In order to stop un-authorised users from entering the database we set up a root for mySql. To do this we create a hashed password for the root user using mysql\_secure\_install and this will then disable anybody from entering the database without the password, the non-root user is then disabled also. In the code connecting to the database we then need to login as root and grab the password, you can do this by hardcoding the password into the code, but the better option would be to grab the password using os.getenv(). This will grab the password and not allow somebody who managed to access the code to be able to view the password.

**User Interface**

# Paper Prototype

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceGraphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Diagram

Description automatically generated with medium confidenceChart

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

# Paper prototype page descriptions:

**1. Login Page**

User uses this page to login by entering their username and password that they registered with.

**2. Register Page 1**

User will use this page to begin their registration by entering their details like their username, password, and email. They will use these details to log into their account.

**3. Register Page 2 (About You)**

The second register page is to get the users physical details such as their height, weight and age. This data can be used for trainers and for the user to track their progress.

**4. Main Menu Page**

The main menu page contains the links to most of the applications pages, this is where the user will be able to navigate the app.

**5. Before Lift Page**

This screen is where the user has setup the device and is getting ready to lift, the user can click the start button to begin the session.

**6. During Lift Page**

This is what the user see’s while they are lifting, the graph displays the position of the bar compared to the optimal form line, the user can also see details such as the time of the session, what rep they are on and what set they are on.

**7. Lift Summary Page**

This is what the user can see after their session is finished or when they click on this session from their history. The summary page shows the users best lift of the session, their max weight lifted and time of session, along with the reps they done and sets.

**8. Session History Page**

This is where the user and their trainer can see the trainee’s session history, click one of the days and you will be linked to the lift summary page of that session.

**9. Side Bar Menu**

This is where the user can sign out and view other links to parts of the application which are not displayed on the main men u page and this side bar can be accessed from any page on the app.

**10. Trainees Page**

This is the page where the trainer can see their trainees and once they click on their name, they can see the trainee’s session history page.

# Connection:

Connection to the device and app is through a Wi-Fi connection where the device is connected either to your hotspot/gym Wi-Fi. If there is no connection, a message should be displayed on the app telling the user that there is no solid connection and that they need to setup the device correctly.

**Users**

# User Profiles

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generatedGraphical user interface, text

Description automatically generated

Diagram

Description automatically generated**Use Case Diagram**

**Testing**

# Software-Testing

The software can be tested with suitable Unit tests. There are test frameworks for Python as well as JavaScript which covers all our used languages. The unit tests can be used to ensure all coded methods work as intended and help by finding bugs or improve the ability to add new features later more simply.

# Hardware-Testing

Once we have a running prototype the accuracy and measurements of the sensors need to be checked. In order to do that we track the data provided by the sensors and check whether the data is plausible. For example, someone can test lift with the prototype and someone else will measure the lifted height by hand and compare the results to the measurements of the ultra-sonic sensor.

We should also test the actual battery capability. How long it will last, how durable the setup is and how the sensors react do different unpredicted movements. For example, dropping the handle, carrying the handle sideways, spinning it etc. We need to make sure that the system does not break when doing unexpected things.

In order to get a good overview, it is necessary to work on a decent test concept which tracks all information and the results of the tests for further improvements and easy access.

# User-Testing

When the initial testing is done, and we have a running prototype we need to make sure to let other people from our different target groups test our app in a real scenario.

We could ask friends or just people and trainers in the gym to try out our app. This not only helps to resolve problems me might not have noticed yet, but it also makes gathering feedback to the overall usage of our app way easier. Everyone should be given a decent questionnaire to give feedback about the usage, design, and functionality of our app. This helps to get a real user perspective from all target groups on our app. Only through the perspective of different users we can spot missing features we did not think about and find problems we on our own had never seen.

People that should test our app in this state are experienced lifters, casual gym users, people who never lifted before and trainers.

**MoSCoW**

* **Must** - Accelerometer and Ultrasonic sensor data i.e., time, x and y co-ordinates. Feedback on lift
* **Should** - Drawing a live graph with data, saving after each rep. Use accelerometer to calculate tilt on bar.
* **Could** - Use heartbeat sensor to track heartrate/pulse during rep. Have a real-time graph from data. Have screen display in front of the user wirelessly.
* **Would** - Make mobile app. co2 sensor to measure air quality.