Criterion B: Design

Contents:

1. Database

1.1. ER Diagram

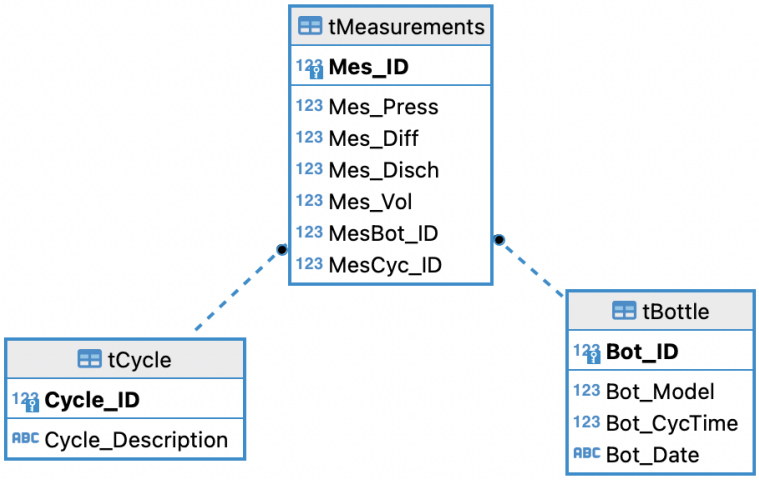

Figure 1 - ER Diagram for database

1.2. Database dictionary

Table 1 - Database dictionary

Table	Column	Data Type	PK or FK	Description
tCycle	Cycle_ID	Integer	PK	Primary key of tCycle
	Cycle_Description	Text		Description of cycles
tMeasurement	Mes_ID	Integer	PK	Primary key of tMeasurement
	Mes_Press	Real		Stores pressure values
	Mes_Diff	Real		Stores differential pressure values
	Mes_Disch	Real		Stores discharge rate values
	Mes_Vol	Real		Stores water volume values
	MesBot_ID	Integer	FK	Foreign key for tBottle
	MesCyc_ID	Integer	FK	Foreign key for tCycle
tBottle	Bot_ID	Integer	PK	Primary key of tBottle
	Bot_Model	Integer		Stores a 0 if the model is conventional. Stores a 1 if the model is novel
	Bot_CycTime	Integer		Time of cycles for this bottle
	Bot_Date	Text		Stores the date and time of tested bottle

2. GUI

2.1. Launch Window

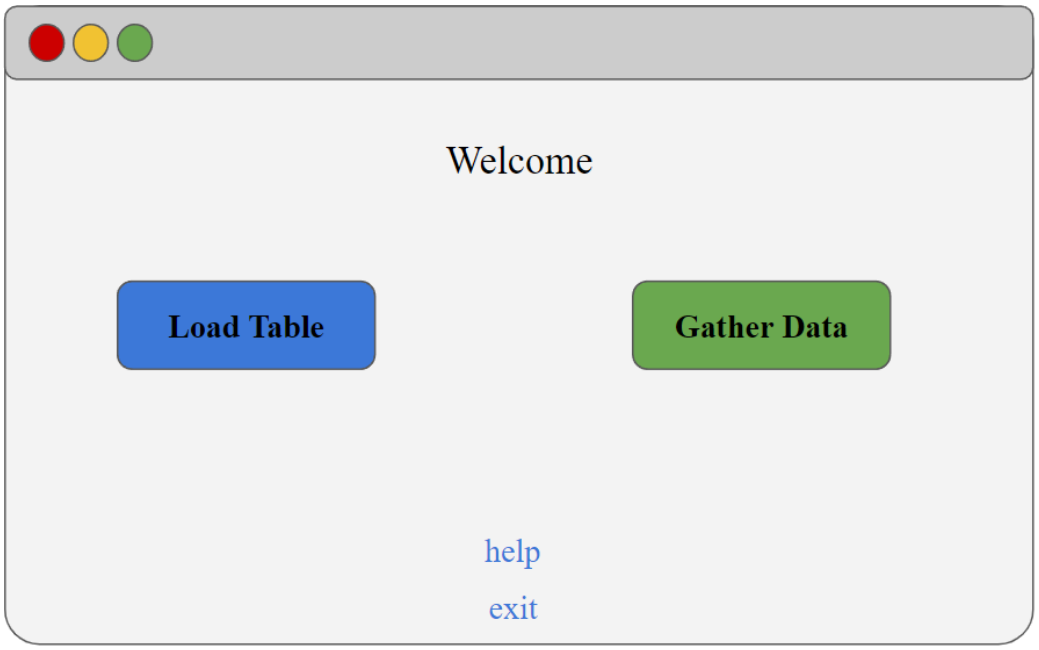

Figure 2 - Main GUI

- This is the launch window once the application is run.
- Load Table and Gather Data are two buttons that will each open separate menus.
- Help button provides documentation for user
- Exit button exits the application

2.2. Gather Data Menu + Save Data + Save Table

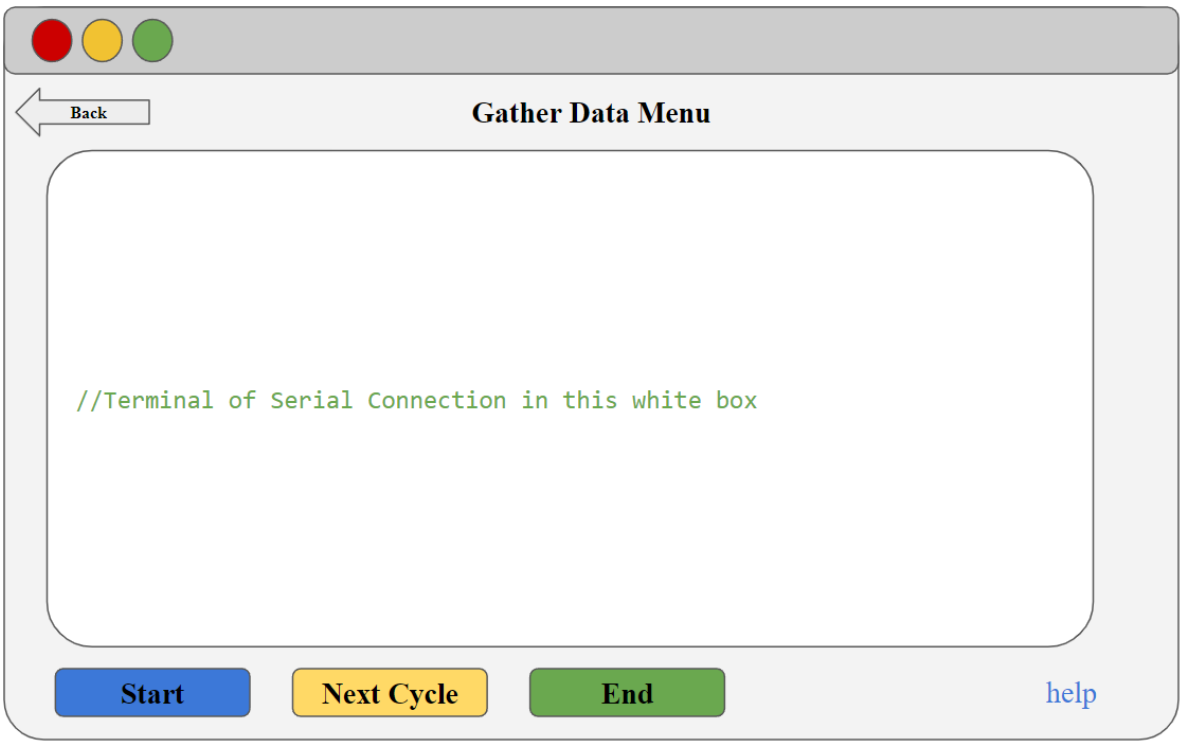

Figure 3 - Gather Data Menu

- This menu provides the interface to gather data when the Arduino is connected to the computer.
- Press Start to launch the interface
- Press Next Cycle to start a new cycle
- Once the user has done enough cycles, he needs to press the button “End”
    - This button will pop up the “Save Data” window

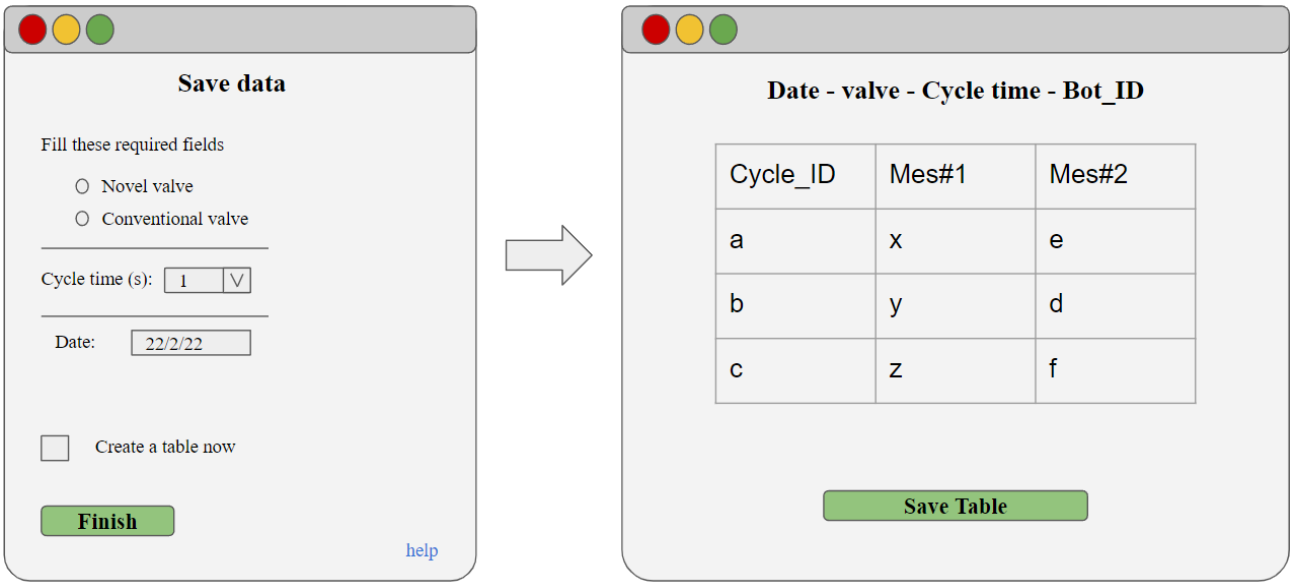

Figure 4 - Save Data/Create Table

- User fills this information before saving the data in the database.
- The Finish button will send data to the database.
- Press Save Table to save the table as an excel file on the user’s computer.

2.2. Load Table Menu + Save Table

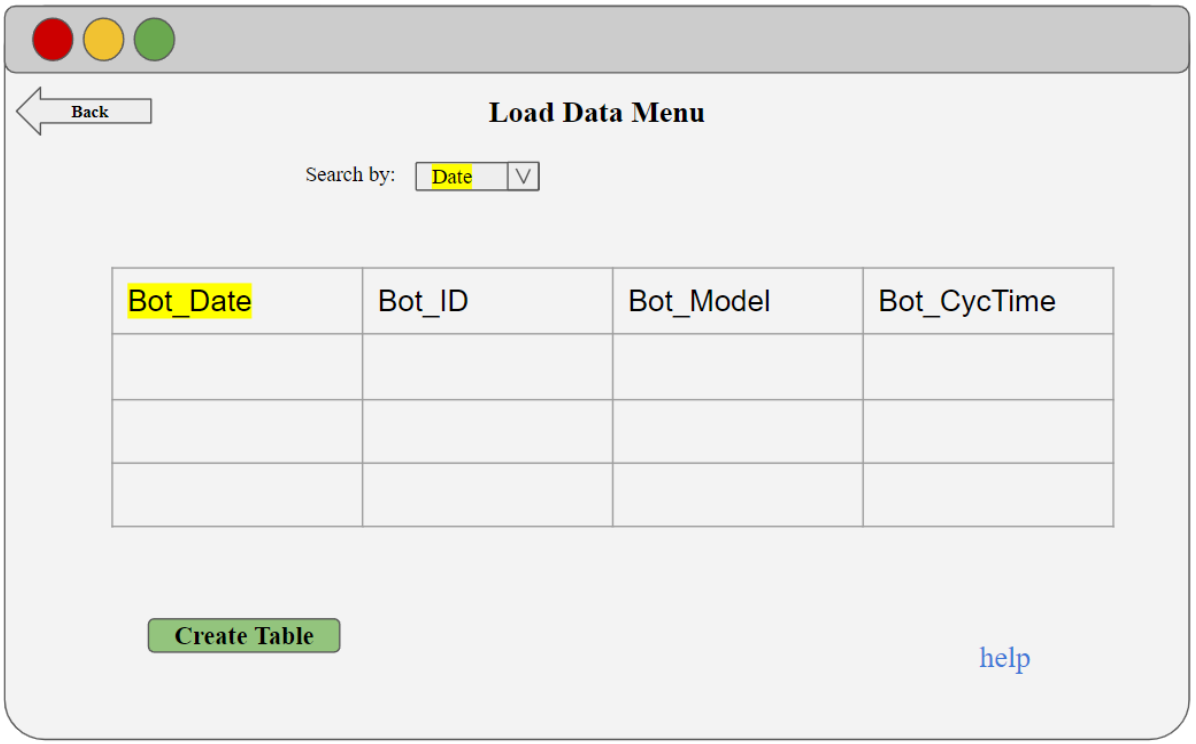

Figure 5 - Load Data Menu

- This menu is used to search the data that needs to be processed in tables.
- Select the bottle you want to load data from.
- Press on Create Table to create a table, and a window will open with a table.

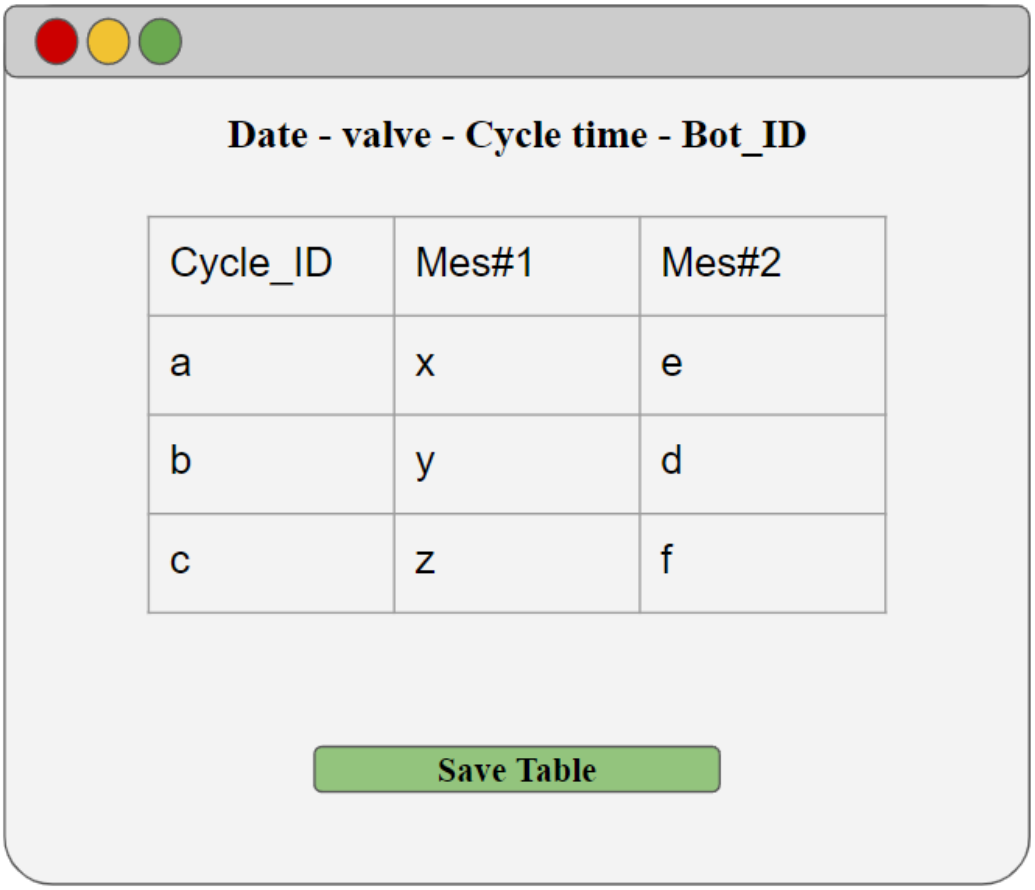

Figure 6 - Create Table

- Press Save Table to save the table as an excel file on the user's computer
- Pressing Save Table will also open the Launch Window.

3. Algorithms

3.1. Arduino program - Flowchart

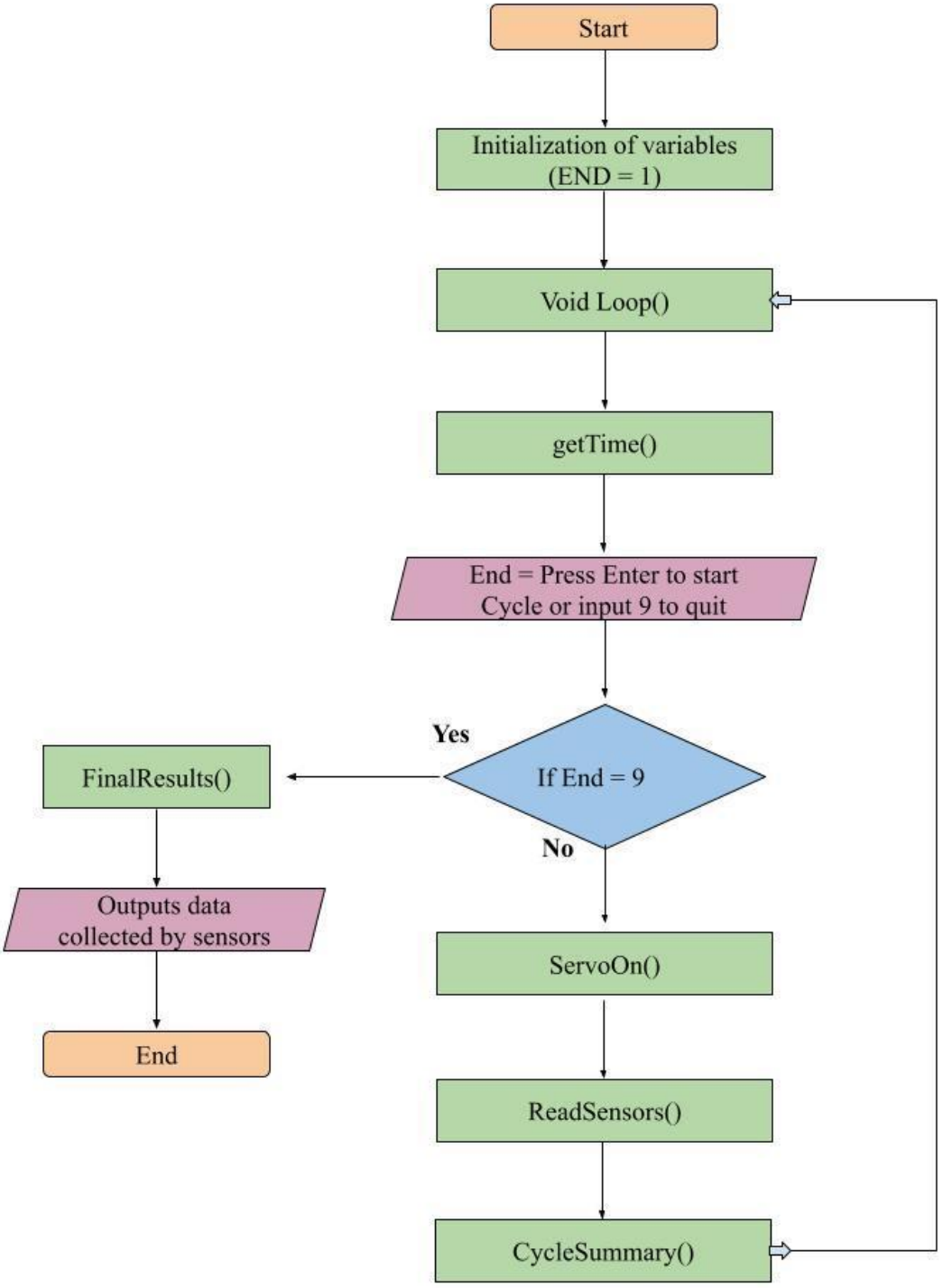

Figure 7 - Arduino Flowchart

- This algorithm will actuate the servo motor to press on the cap of the bottle upon user input
- As the cap is pressed, the sensors collect data, and the program stores them in arrays.
- The user ends the arduino void loop by pressing on the 'End' button. This will make End = 9.

3.2. Java stores data read from COM port

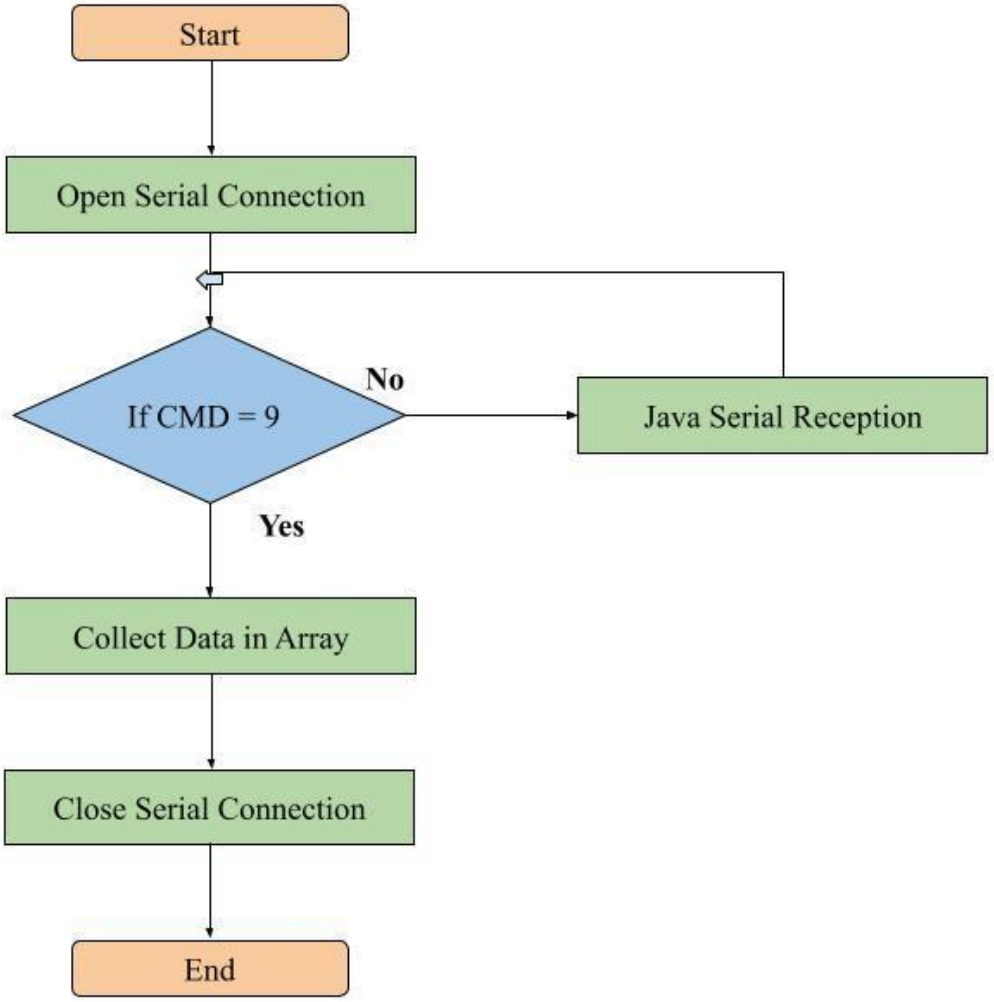

Figure 8 - Java stores measurements

- This algorithm first opens a serial connection with the COM port of the Arduino.
- A while loop occurs for the Java program to receive the information from the COM port; the while loop ends when the 'End' button is pressed. This will make End = 9.
- The program stores the data in arrays and finally the connection is closed.
- This algorithm is running concurrently to the Arduino program, as both programs are constantly exchanging information through the Serial connection.

3.3. Java creates database + stores data in database

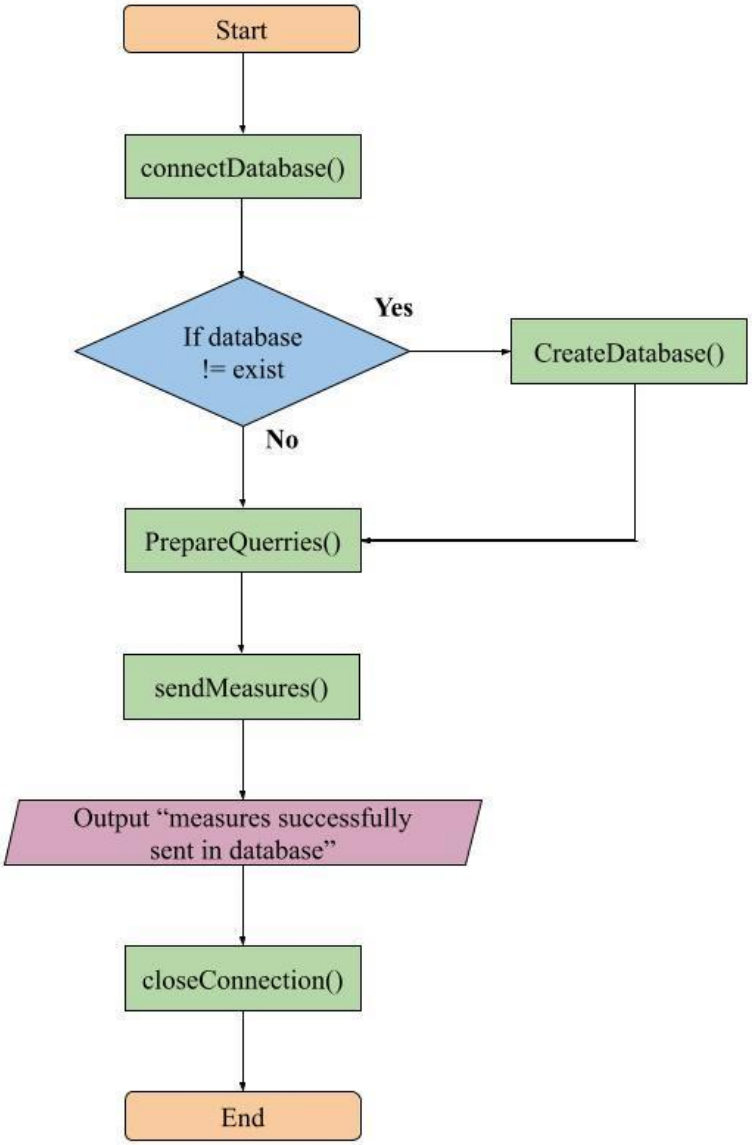

Figure 9 - Java send measurements to database

- This algorithm first opens a connection with a database. If no database exists, then it creates a database.
- Then, the program prepares the queries and then executes them based on the data collected in the Arduino.

3.4. Java creates views of the database + Saves Table

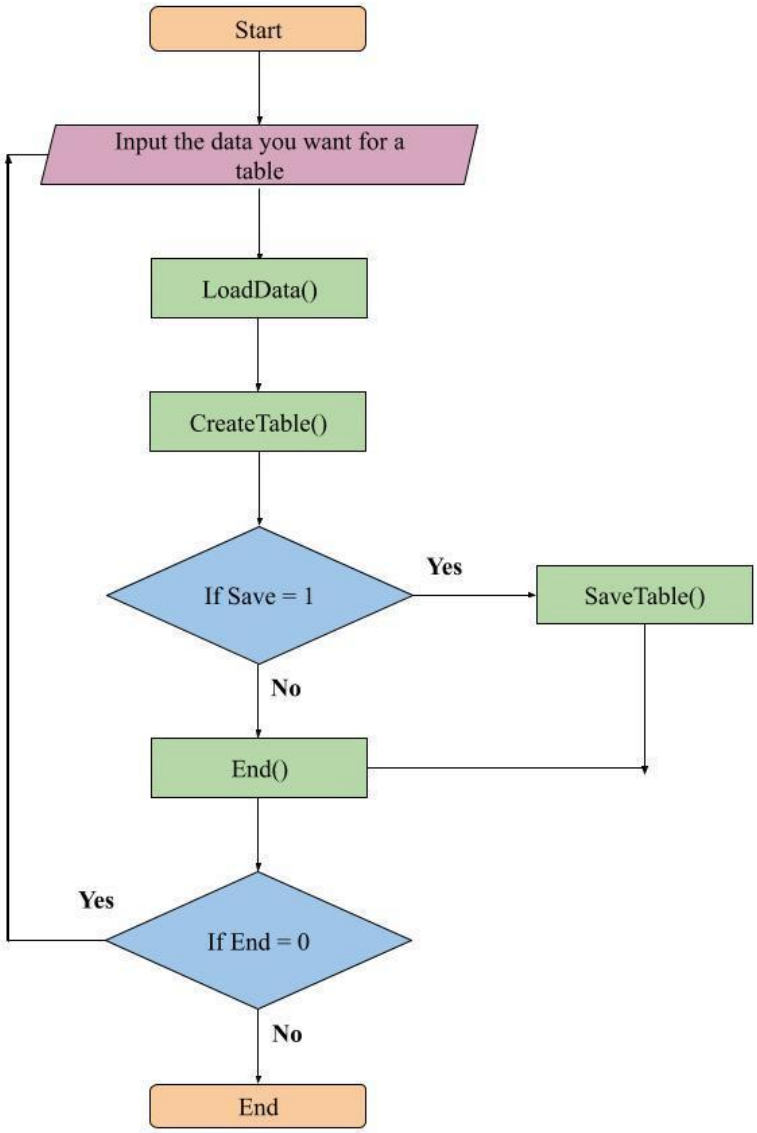

Figure 10 - Java creates views of database

- This algorithm provides an interface for the user to decide the data that he will process in a table.
- The program creates the table and asks whether the table needs to be saved.
- The user creates tables until he wants to end the application

4. Data Structures

4.1. Arrays

1. In the Arduino program, each measurement's average has its own array. The indexes of the arrays refer to the cycle number.
    a. Efficient to store the averages while the program is running
    b. Easy to print these averages in loops.
2. In the Java reception of data, I will be using an array for each measurement to store the data. This will make the transfer to the database efficient. I will create loops to insert the measurements from the arrays with SQLite prepared statements.

4.2. 2-Dimensional Array

1. To create JTables, I will load the data from the database in 2-dimensional arrays to create a table in an efficient manner.

5. External Design

5.1. Actuation design

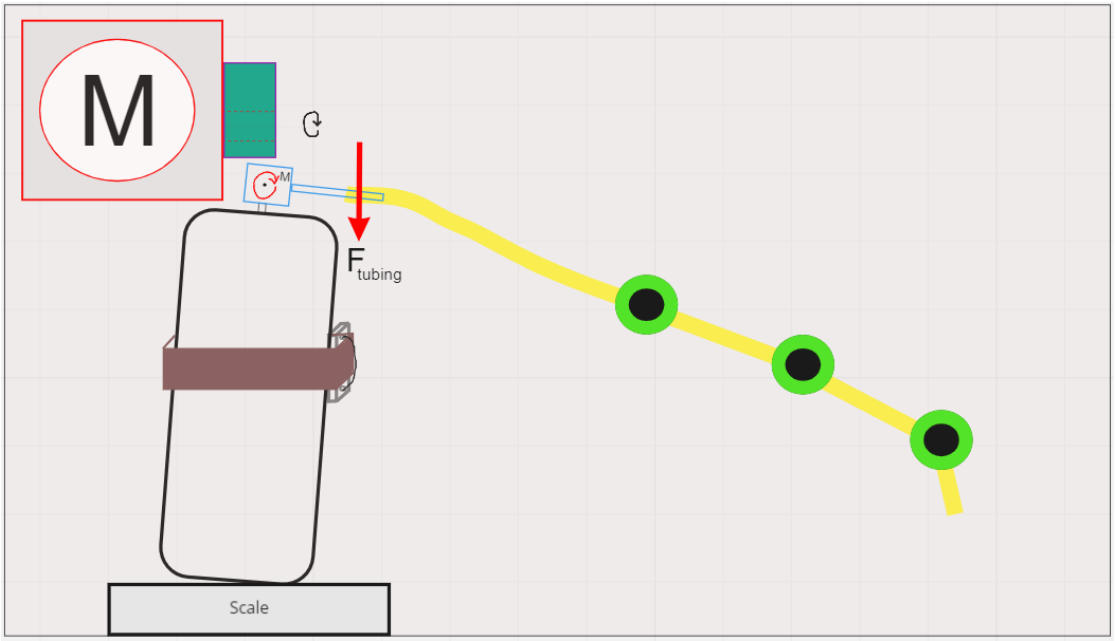

Figure 11 - Actuation design

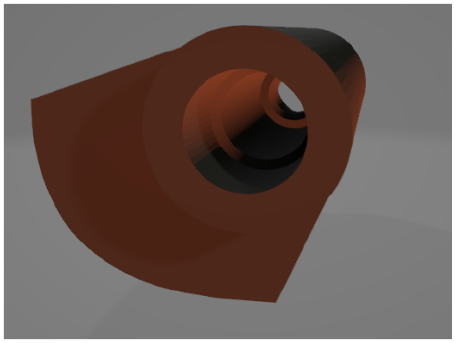

Figure 12 - Design of cam that pushes down on the cap

- Figures 11 and 12 show the bottle and the position of the motor (M) that push down on the cap of the bottle with a cam.

Figure 13 - First iteration of the actuation design.

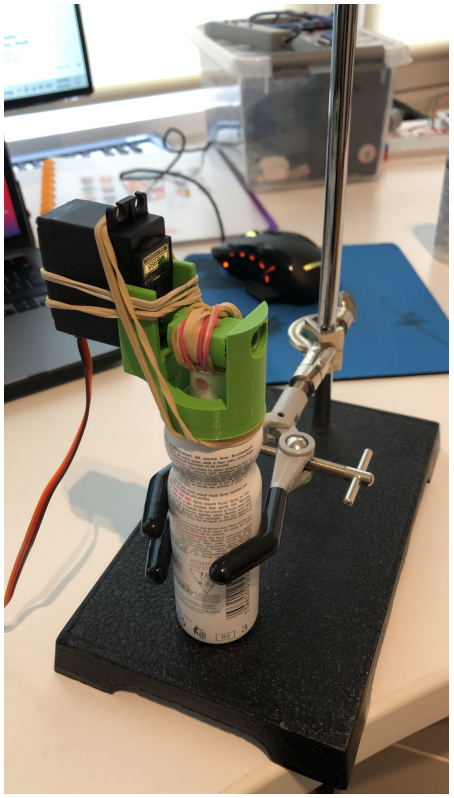

Figure 14 - Second iteration of the design - Used a more powerful Servo motor.

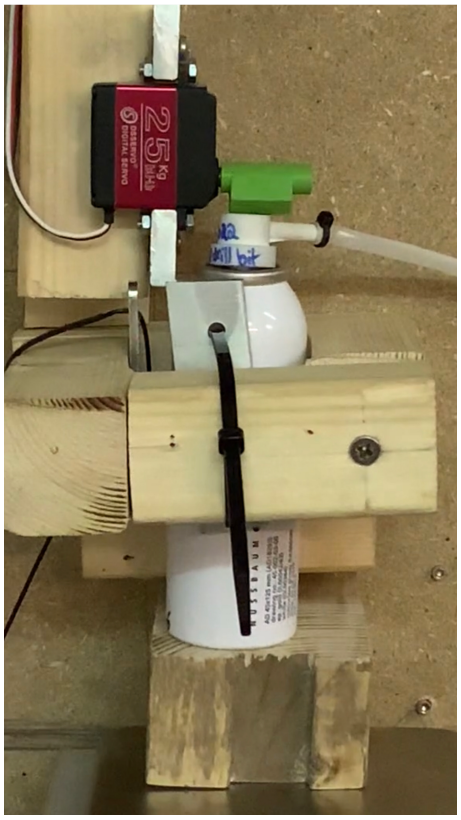

Figure 15 - Final prototype of the actuation

- I bought a more powerful servo than the second iteration.
- I fix the servo with metal, allowing the servo to use its full torque and preventing the servo to move

5.3. Sensor design (helicopter view)

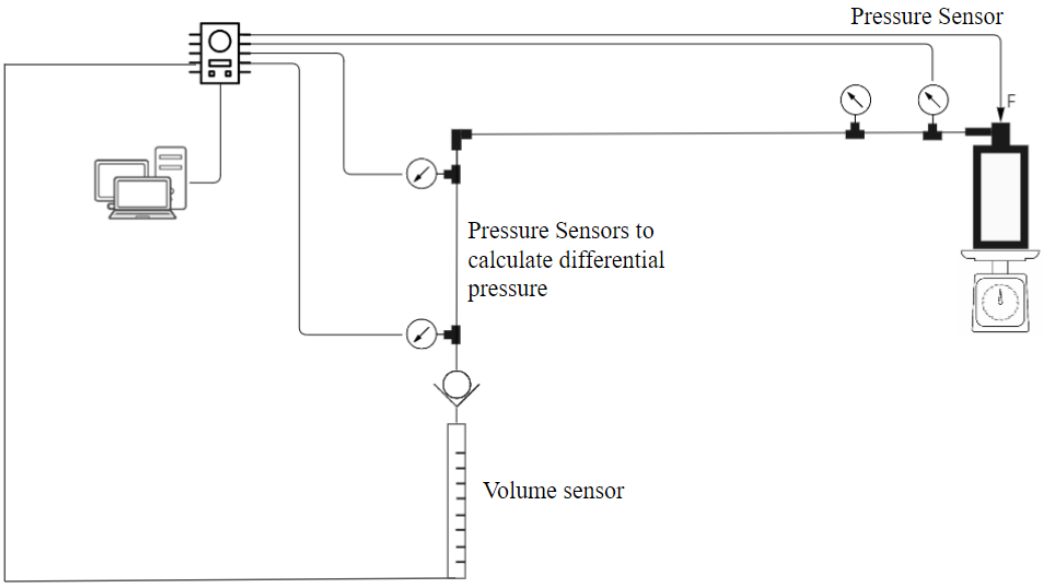

Figure 16 - Helicopter view of experimentation setup

- This figure shows the placement of the sensors in the experiment.

6. Test plan

Table 2 - Test Plan

# of success criteria	Test Date	Test Method	Outcome
1	05/20/22	Using a RDBMS, my client and I can find out whether the database is created successfully after running the application.	Successful
2	04/29/22	At my client's lab, test whether the mechanism of the servo presses successfully on the different caps. Also, test whether my client can change the actuation time on command (as seen in *Appendix 1.3 and 1.4*)	Successful
3	10/02/22	Using a RDMS, my client and I can confirm whether the cycle time is successfully stored.	Successful
4	10/02/22	For this criterion, my client and I first need to make sure that the average of each measure per cycle is computed successfully in the Arduino program. Then, we need to check whether the measurement is successfully stored in the database using a RDBMS	Successful
5	10/02/22	For this criterion, my client and I first need to make sure that the average of each measure per cycle is computed successfully in the Arduino program. Then, we need to check whether the measurement is successfully stored in the database using a RDBMS	Successful
6	10/02/22	For this criterion, my client and I first need to make sure that the average of each measure per cycle is computed successfully in the Arduino program. Then, we need to check whether the measurement is successfully stored in the database using a RDBMS	Successful
7	xx/xx/xx	At the lab, my client and I could test the radio connection by having an Arduino in a different room to see if its practical implementation works.	Not successful
8	06/ 01/22	This can be tested by printing a success message to the serial monitor when the Java connection to the COM port is achieved	Successful
9	10/02/22	My client will test whether he can create tables successfully with the developed interface. In addition, there are help buttons on each window to provide documentation to the user if he cannot use the interface successfully	Successful
10	xx/xx/xx	This can be tested by saving a table from the interface and then seeing if it is successfully saved on the user's computer.	Not successful