DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

System Design: Device to Administer Motor Task Training to Lab Animals

Revisions

Revision	Author	Changes	Date
001	Ang Li Hao Xuan Maocheng Shi Ruichen Chen	First Attempt	2021-03-11
002	Ang Li Hao Xuan Maocheng Shi Ruichen Chen	Addressed the Feedback from Client (Dr. Keith Fenrich)	2021-03-12

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Table of Contents

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Acronyms

Acronym	Full Description
AMTTLA	Administer Motor Task Training to Lab Animals
CSS	Cascading Style Sheets
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
ID	Identify Document
IP	Internet Protocol
LAN	Local Area Network
R/W	Read/Write
SQL	Structured Query Language

References

[1] D.-K. Electronics, “LTH-301-32,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/lite-on-inc/LTH-301-32/3198349. [Accessed: 11-Mar-2021].

[2] D.-K. Electronics, “MCP3008-I/P,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/microchip-technology/MCP3008-I-P/319422. [Accessed: 11-Mar-2021].

[3] D.-K. Electronics, “SEN-14729,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/sparkfun-electronics/SEN-14729/9555603. [Accessed: 11-Mar-2021].

[4] D.-K. Electronics, “SEN-13879,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/sparkfun-electronics/SEN-13879/6202732. [Accessed: 11-Mar-2021].

[5] D.-K. Electronics, “MCP4151-103E/P,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/microchip-technology/MCP4151-103E-P/1874217. [Accessed: 11-Mar-2021].

[6] D.-K. Electronics, “28BYJ-48 Series,” *28BYJ-48 Series - MikroElektronika - Stepper Motors | Online Catalog | DigiKey Electronics*. [Online]. Available: https://www.digikey.ca/catalog/en/partgroup/28byj-48-series/59691. [Accessed: 11-Mar-2021].

[7] D.-K. Electronics, “ULN2003APG,CN,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/toshiba-semiconductor-and-storage/ULN2003APGCN/871122. [Accessed: 11-Mar-2021].

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

[8] Gavinlyonsrepo, "gavinlyonsrepo/RpiMotorLib," *GitHub*. [Online]. Available: https://github.com/gavinlyonsrepo/RpiMotorLib. [Accessed: 11-Mar-2021].

[9] D.-K. Electronics, "3873," *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/3873/1528-2689-ND/9603612?utm_medium=email&utm_source=oce&utm_campaign=4251_OCE21RT&utm_content=productdetail_CA&utm_cid=1733681&so=68350897&mkt_tok=MDI4LVNYSy01MDcAAAF7k8qDv61mJWdMGzxWZAoukEgmEXUyyqs_x35F-PTd4kNkp7dYxfhX6tasBG1fPCAa8kkFtwYMmFzm77hbTPjtf2Ecma0vtqY64Nl-C87U. [Accessed: 11-Mar-2021].

[10] P. S. Foundation, "spidev," *PyPI*, 02-Jun-2020. [Online]. Available: https://pypi.org/project/spidev/. [Accessed: 11-Mar-2021].

[11] A. S. App-Generator, "app-generator/flask-datta-able," *GitHub, 2020*. [Online]. Available: https://github.com/app-generator/flask-datta-able. [Accessed: 11-Mar-2021].

[12] J. S. F.- js.foundation, "jQuery.ajax()," *jQuery.ajax() | jQuery API Documentation*. [Online]. Available: https://api.jquery.com/jquery.ajax/. [Accessed: 11-Mar-2021].

[13] Pallets, "Welcome to Flask¶," *Welcome to Flask - Flask Documentation (1.1.x)*, 2010. [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/. [Accessed: 11-Mar-2021].

[14] Sqlalchemy, "The Python SQL Toolkit and Object Relational Mapper," *SQLAlchemy*. [Online]. Available: https://www.sqlalchemy.org/. [Accessed: 11-Mar-2021].

[15] F. Corp., "Scan a network in seconds," *Advanced IP Scanner - Download Free Network Scanner.* [Online]. Available: https://www.advanced-ip-scanner.com/. [Accessed: 11-Mar-2021].

[16] J. S. F.- js.foundation, "jQuery.ajax()," *jQuery.ajax() | jQuery API Documentation*. [Online]. Available: https://api.jquery.com/jquery.ajax/. [Accessed: 11-Mar-2021].

[17] D.-K. Electronics, "Raspberry Pi 4 Model B," *DigiKey*. [Online]. Available: https://www.digikey.ca/en/product-highlight/r/raspberry-pi/raspberry-pi-4-model-b. [Accessed: 12-Mar-2021].

[18] C. Ruichen, L. Ang, S. Maocheng, X. Hao, *Preliminary Design Document,* ECE 492, Dept. Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, February 12, 2021.

[19] D.-K. Electronics, "Assembled Pi T-Cobbler Breakout for Raspberry Pi," *Assembled Pi T-Cobbler Breakout for Raspberry Pi - Adafruit Industries LLC - Evaluation Boards - Expansion Boards, Daughter Cards | Online Catalog | DigiKey Electronics*. [Online]. Available: https://www.digikey.ca/catalog/en/partgroup/assembled-pi-t-cobbler-breakout-for-raspberry-pi/67419. [Accessed: 13-Mar-2021].

[20] D.-K. Electronics, "2N2222," *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/stmicroelectronics/2N2222/654412. [Accessed: 13-Mar-2021].

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

[21] D.-K. Electronics, “LTH-301-32,” *DigiKey*. [Online]. Available: https://www.digikey.ca/en/products/detail/lite-on-inc/LTH-301-32/3198349. [Accessed: 11-Mar-2021].

[22] D.-K. Electronics, “ROB-12438 Datasheet by SparkFun Electronics,” *Digi*. [Online]. Available: https://www.digikey.be/htmldatasheets/production/1872001/0/0/1/rob-12438.html. [Accessed: 13-Mar-2021].

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

1 Purpose

This document describes the system design for the AMTTLA. It is based on the Preliminary Design [18] where an initial high-level system design was presented, but the focus was on the minimum viable end-to-end (walking skeleton) functionality. This document updates and extends that work to provide the necessary detail to build an early (alpha) version of the string-pulling system.

2 Preliminary Design Summary

The AMTTLA Design [18] focused on the walking skeleton needed to validate the design adopted.

As seen in figures 1 and 2, the walking skeleton system comprises the hardware components and software/firmware components. The hardware components mainly consist of the Raspberry Pi and string-pulling device. The Raspberry Pi executes all software/firmware components of the project excluding the web browser application. The string-pulling device is the mechanism designed for the training specifications. It contains the stepper motor, load cell, and optical sensor. The software/firmware components mainly consist of the web server, hardware manager, and database. The web server hosts a web page allowing the user to customize configurations on the string-pulling device and retrieve training information. The hardware manager controls the hardware and records the real-time training data. It interacts with the stepper motor controller, load cell signal processor, and optical sensor processor, which are firmware components. The database stores training information.

Figure 1 also illustrates the critical communication paths (yellow lines) between the system components. The Raspberry Pi is connected to the LAN via the router where devices under the same network can access the webpage Pi hosts. The Raspberry Pi GPIO port sends control signals to the stepper motor and retrieves data from the load cell and optical sensor.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

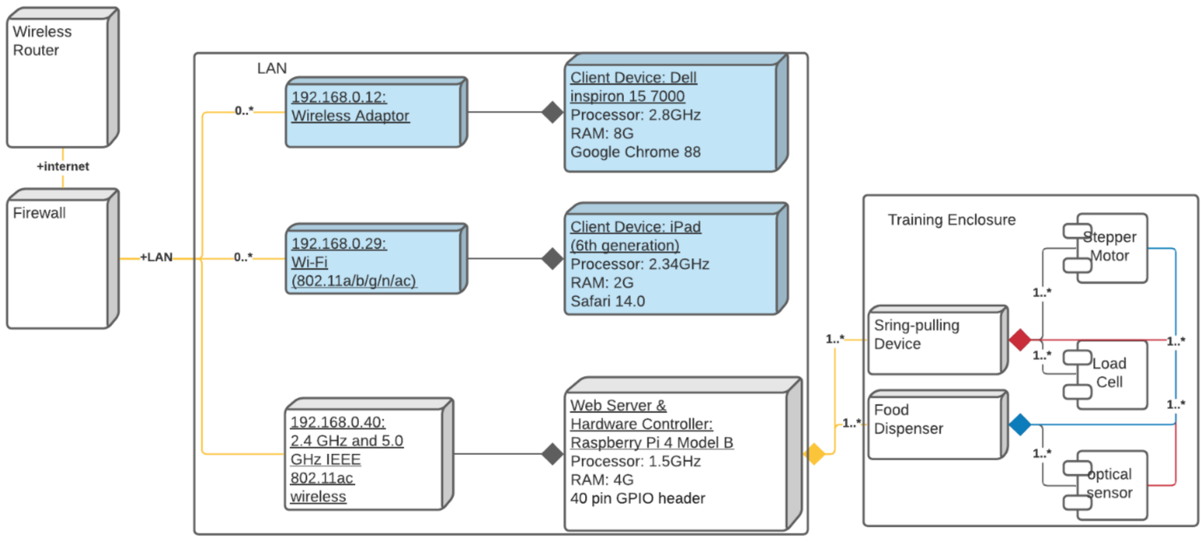

Figure 1: The system deployment diagram of the primary design document, where the yellow line indicates the critical communication paths.

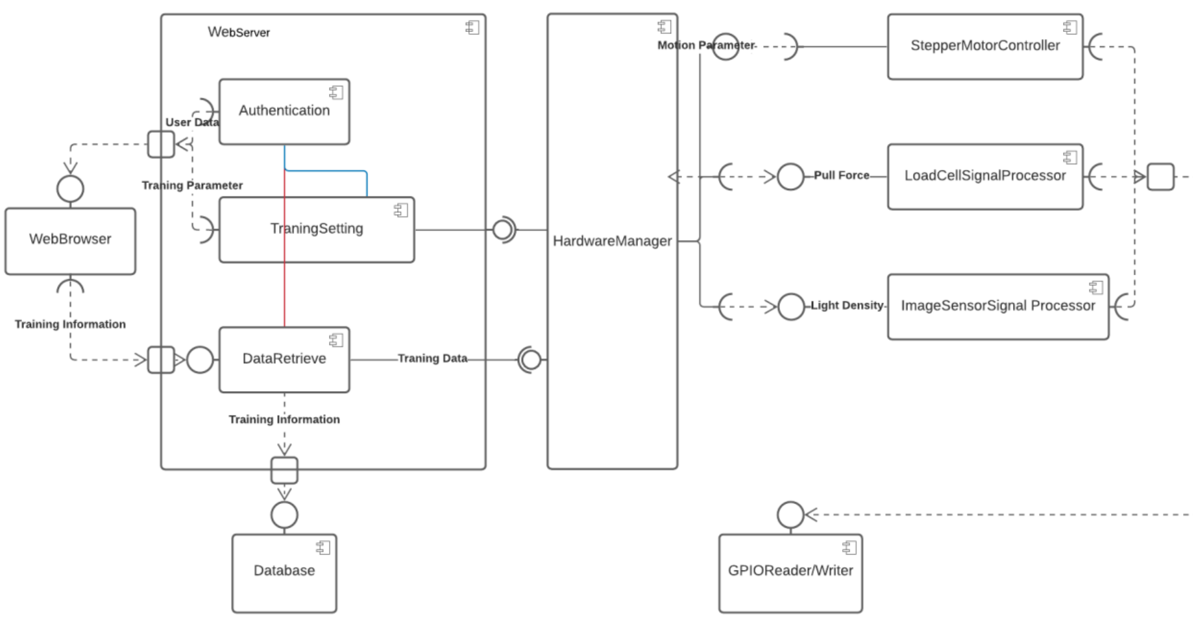

Figure 2: The system component diagram of the primary design document.

3 Main Design Goals

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

The objective of AMTTLA is to develop an automated animal training system that can administer and evaluate performance in the string-pulling task. To fulfill the requirements, we divide our design into four critical goals.

The first goal is to design a food dispenser that distributes food pallets when the training subject completes one training task. The reward motivates the subject to complete the specified training tasks.

The second goal is to design a web server that executes on the Raspberry Pi. It hosts a webpage that can be accessed by the user under the same LAN. The webpage allows users to configure the training parameters (i.e., the requirements to complete one training task) and retrieve the training record.

The third goal is to design a hardware manager which takes the user input and interprets the parameters to control the hardware. It also records the real-time data collected by the sensors. Notice that both the second and third design goad requires support from the database.

The fourth goal is to design a robust string-pulling device that controls the force threshold for one training subject to pull the string and collects real-time training metadata (i.e., pull force, pull distance, pull velocity, and the number of tasks completed) through various sensors.

4 System Design

The AMTTLA system comprises two main subsystems, the client device and web server (which is integrated with hardware manager). The purpose of the system design is to give a clear overview of the relationship between our implementation components, which is critical for further product modification and improvement. Major nodes and components in each subsystem are identified below.

4.1 System Architecture

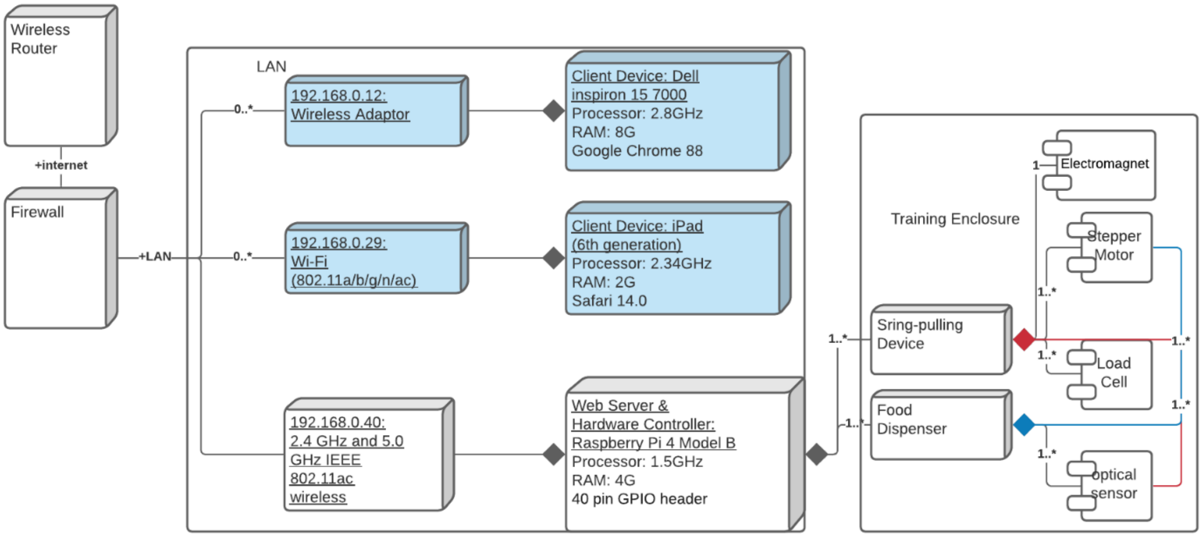

Figure 3 Updated deployment diagram. The modification we made was that we added the electromagnet to the string-pulling device, which was used to control the required pull force.

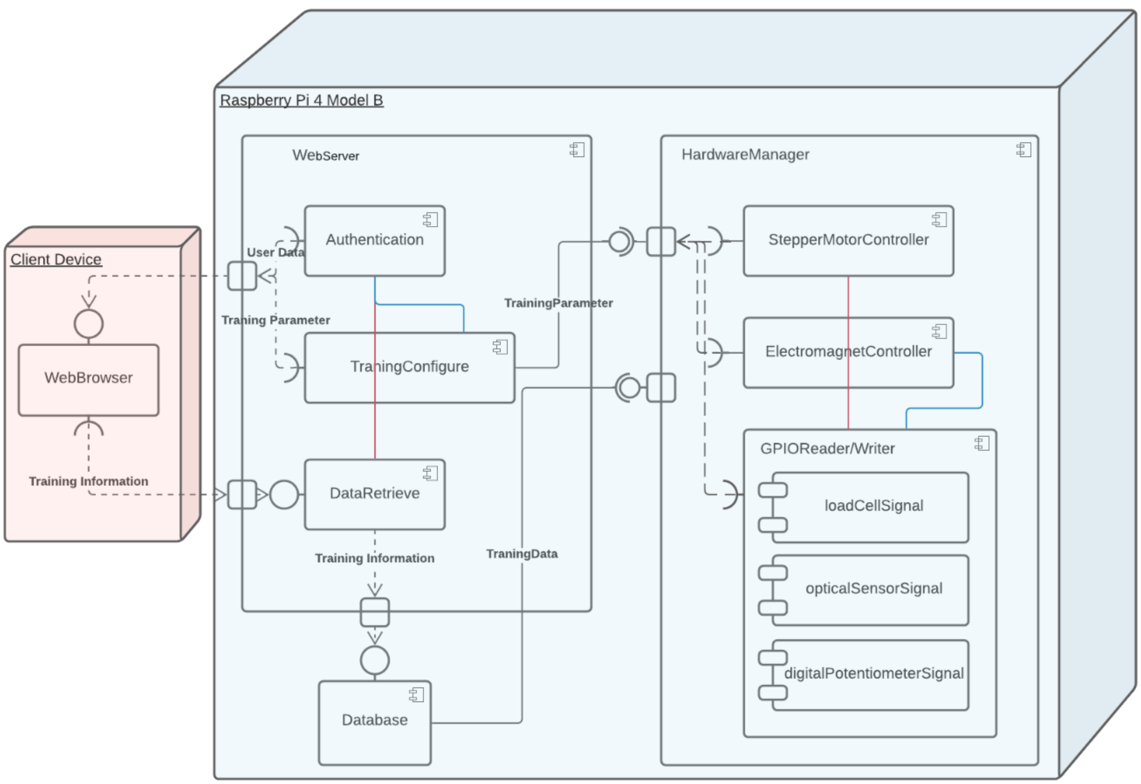

Figure 4 Updated component diagram. The modification we made was that we added the electromagnet controller in the hardware manager. We also integrate the load cell signal processer, optical sensor processer, and digital potentiometer controller into the GPIO reader/writer. Their corresponding hardware components do not require an external driver library and are directly connected to the Raspberry Pi GPIO port.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

4.1.1 Hardware Components

The Hardware comprises the following components:

Table 1 Hardware Components

HC #	Hardware Component Title	Description
HC-01	Raspberry Pi 4 Model B	Executes all software components listed in figure 4, excluding the web browser application.
HC-02	Wireless Adapter: Wi-Fi (802.11a/b/g/n/ac), 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, etc.	Allows the device to access Wi-Fi.
HC-03	String-pulling Device	It consists of the force controller and pulley system. The former (includes the electromagnet and load cell) controls the force threshold pulling the string. The latter (includes load cell) collects the real-time training data.
HC-04	Food Dispenser	Dispenses a fixed amount of food pellets to the rats/mice.
HC-05	Stepper Motor	Rotates the floor of the food dispenser to deliver the food pellets.
HC-06	Electromagnet	Exerts a normal force on the pulley to produce a friction force to control the required pull force.
HC-07	Load Cell	Measures the pull force on the string.
HC-08	Optical Sensor	It is used on the string-pulling device and food dispenser. For the string-pulling device, it measures the pull velocity and the pull distance. For the food dispenser, it checks whether one food pellet delivery attempt succeeds.
HC-09	Client Device	Accesses the webpage under the LAN through Wi-Fi.
HC-10	Wireless Router	Initializes LAN connection.

4.1.2 Firmware Components

The Firmware comprises the following components:

Table 2 Firmware Components

FC #	Firmware Component Title	Description
FC-01	GPIO Reader/Writer	Reads signals from the load cell and optical sensor. Writes signals to the digital potentiometer (which is integrated with the electromagnetic and controls the electromagnetic output force).
FC-02	Stepper Motor Controller	Controls the stepper motor.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

FC-03	Electromagnet Controller	Switch on/off the electromagnet.

4.1.3 Software Components

The Software comprises the following components:

Table 3 Software Components

SC #	Software Component Title	Description
SC-01	Web Server	Hosts a web page allowing the user to customize settings on the string-pulling device.
SC-02	Authentication	Protects the training system.
SC-03	Training Configure	Transmits the custom settings from the web server to the hardware manager.
SC-04	Data Retrieve	Queries training information from the database.
SC-05	Hardware Manager	Controls the hardware. Records the real-time training data into the database.
SC-06	Database	Stores the training data and system information (e.g., user account, experiment ID, subject ID).
SC-07	Web Browser	Accesses the webpage.

5 System Requirements

Table 4 Functional Requirements

FR #	Functional Requirement Description
FR-01	The training system shall host the webpage under the LAN, where the user can set training preferences.
FR-02	Once the training preferences are received, the sting-pulling device shall be configured to meet the task requirements (e.g., adjust the force threshold pulling the string).
FR-03	The sensors shall collect the real-time metadata, where the latter is then interpreted as training data.
FR-04	The training data shall be stored at the local database, where the user can query a certain piece of training information from.
FR-05	The food dispenser shall deliver a fixed amount of food pellets when the subject completes one training task.
FR-06	The stored training data shall be transmitted through LAN
FR-07	The training system shall be easily attached/detached to/from the training enclosure.
FR-08	The training system shall accommodate different sizes of subjects, from mice to rats.
FR-09	The string in the system shall be easily replaced. It shall tolerate edge cases (e.g., animal chewing or biting).
FR-10	The user shall log in to access the training system, while the administrator can manage the access permission of all accounts that the training system registered.

Table 5 Performance Requirements

PR #	Performance Requirement Description	Related FRs
PR-01	The training system shall be set up (i.e., connecting the system to the LAN and running the software) within 5 minutes.	FR-01
PR-02	Once be booted, the training system shall keep functioning for at least one week without any crashing.	FR-01 to FR-06
PR-03	The training system shall respond to incoming user commands (setting training preferences and querying training data) within 30 seconds.	FR-01, FR-04
PR-04	Once the training preferences are received, the sting-pulling device shall be set within 30 seconds.	FR-02
PR-05	The sensors shall collect the real-time metadata of the subject with a satisfying precision and minimum latency (e.g., 0.5 seconds).	FR-03
PR-06	There is no data lost during a sudden outage	FR-04
PR-07	The hardware of the system should last reasonably long without mayor part damage or malfunctioned.	

Table 6 System Requirements

SR #	System Requirement Desc	FR#	PR#	Notes
SR-01	Raspberry Pi shall be able to send and receive data through LAN.	FR-01, FR-06	PR-01, PR-03	
SR-02	Raspberry Pi shall be able to R/W signals through its GPIO ports.	FR-02, FR-03	PR-04, PR-05	
SR-03	Raspberry Pi shall have enough GPIO ports connecting the motor/sensors.	FR-02, FR-03		
SR-04	The motor/sensors shall be able to accept/generate the analog signal generated by Raspberry Pi.	FR-02, FR-03		
SR-05	Raspberry Pi shall be able to work long enough without crashing.		PR-02	
SR-06	Raspberry Pi shall be able to store a reasonable amount of training data.	FR-04	PR-06	Through external storage devices, e.g., SD cards.

SR-07	The client device shall be able to send and receive data through LAN.	FR-01, FR-06	PR-01, PR-03	
SR-08	The string-pulling device shall accommodate different sizes of subjects, from mice to rats	FR-08		
SR-09	The string in the string-pulling device shall be easily replaced.	FR-09		
SR-10	The food dispenser shall be able to function long enough without getting stuck.	FR-05	PR-02	Decreasing the area where food will rub against the structure.

6 Detailed Hardware Design

Figure 3 illustrates the deployment diagram of our design. We utilize the webpage as one GUI module. The user can review, modify, and retrieve the training setting/data when the client device HC-09 and Raspberry Pi 4 Model B HC-01 [17] are under the same LAN (connected to the same wireless router HC-10).

- This suggests that any client device supporting the web browser SC-07 and has a wireless adapter HC-02 can serve as a front-end (here, we use laptop and iPad as an example).
- Meanwhile, we utilize one Raspberry Pi as the back-end server, which executes all software components listed in figure 4 excluding the web browser application. It can communicate with the webpage user through its built-in wireless adapter, as well as communicate with the string-pulling device HC-03 and food dispenser HC-04 through its built-in GPIO ports.
- The string-pulling device consists of an electromagnet HC-05 [9] to control the required pull force, a load cell HC-06 [3] to measure the current full force, and an optical sensor HC-07 [1] to measure the pull velocity. The electromagnet is connected to the output port of the Raspberry Pi GPIO module. In contrast, the load cell and optical sensor are connected to the input port of the Raspberry Pi GPIO module.
- The food dispenser consists of a stepper motor HC-05 [6] to deliver the food pellet and an image sensor to check whether one delivery attempt succeeds. The stepper motor is connected to the input port of the Raspberry Pi GPIO module. The connection setting of the image sensor is the same as in the string-pulling device.

The hardware requires custom-designed mechanism are the food dispenser and string-pulling device. They are designed and manufactured by modeling and then 3D-printing.

6.1 Food Dispenser

The shell of the food dispenser is a half-cylinder with an oblique cut on it. Four convex screw holes appeared around the shell are used to connect and fix the blue base to the body. The main hole at the bottom is for connecting the stepper motor [6] to the yellow floating floor in the middle to rotate it during loading/dispensing the food pellet. There are also three small screw holes placed around the main hole at the bottom. The front one is used to connect the green funnel part to drop the food, and the two rear holes are used to fix the stepper motor to the base to rotate the inner part.

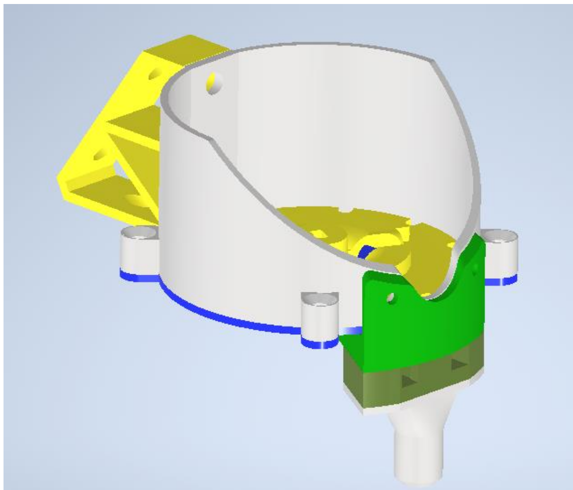

Figure 6: Food dispenser front view.

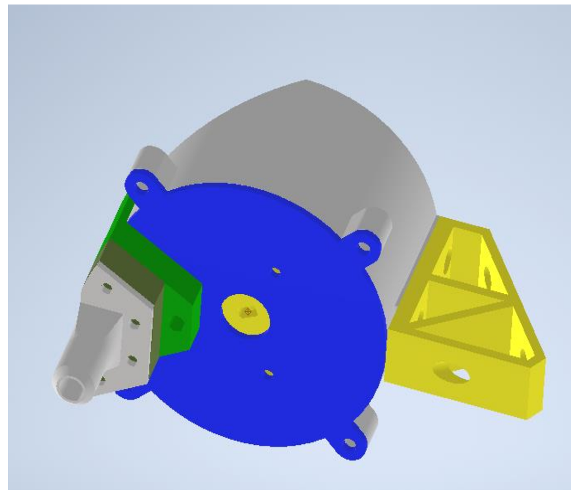

Figure 5: Food dispenser bottom view.

The yellow plate inside the main body is called the floating floor. It is responsible for rotating, carrying the food in the edge cut, and dropping food to the funnel part. Since the entire device is placed in a tilted position as shown in figure 9, the food will gather in the bottom. And when the floating floor rotates, it carries food in the edge cut and brings it to the funnel at the top to drop it. The small rectangular cut in the very center is in the shape of the stepper motor connector. In actual deployment, the stepper motor will be screwed to the bottom of the base and connected to the floating floor through the main hole at the bottom of the base. Meanwhile, the two long strip openings in the center are for accessing three screw holes at the bottom.

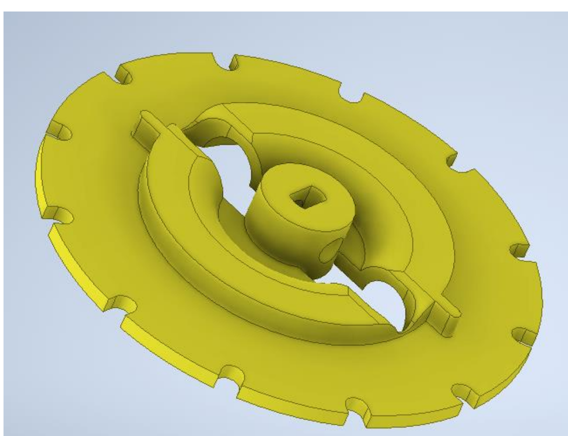

Figure 8: Floating floor top view.

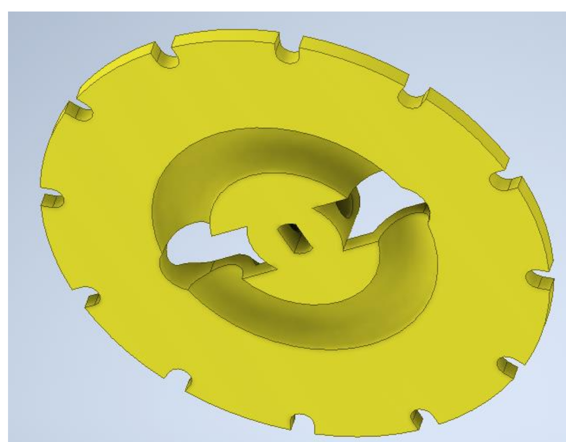

Figure 7: Floating floor bottom view.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

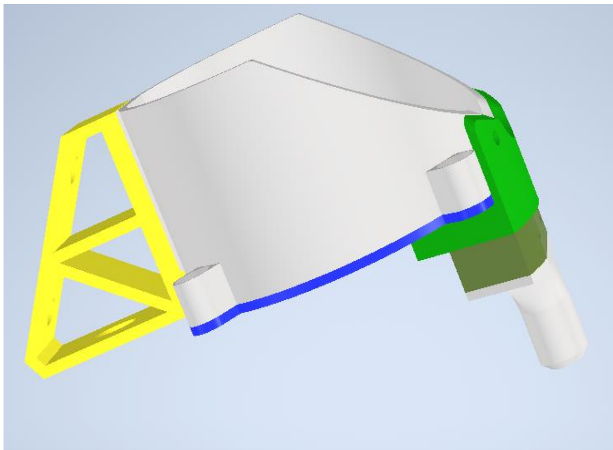

Figure 9: Food dispenser working position.

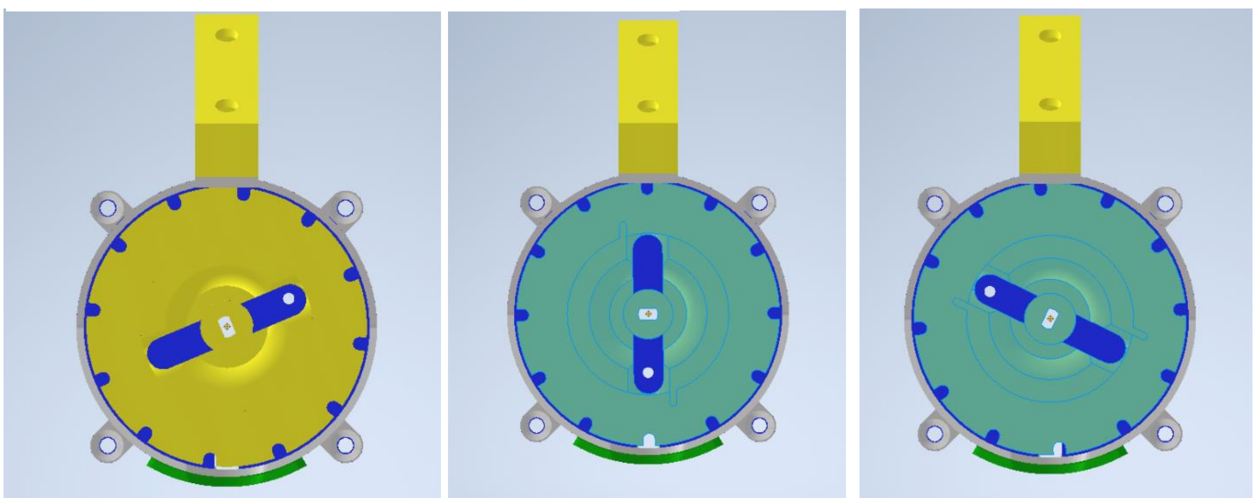

Figure 10: Accessing three screw holes at the bottom.

As illustrated in figure 6, the funnel part at the front consists of three parts. The top light green part acts as an interface to the main body. Two screws at the side and one screw at the bottom are used to fix it to the main body. The middle dark green part has two square holes is for placing an optical sensor that detects whether food pellets are going through. The bottom white part is the actual funnel part that will control the position that the food drops. The yellow skeleton on the left is used for connecting the whole structure to the aluminum frame, which will hold the structure in the training enclosure.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Figure 12: Detailed structure of the bottom funnel part.

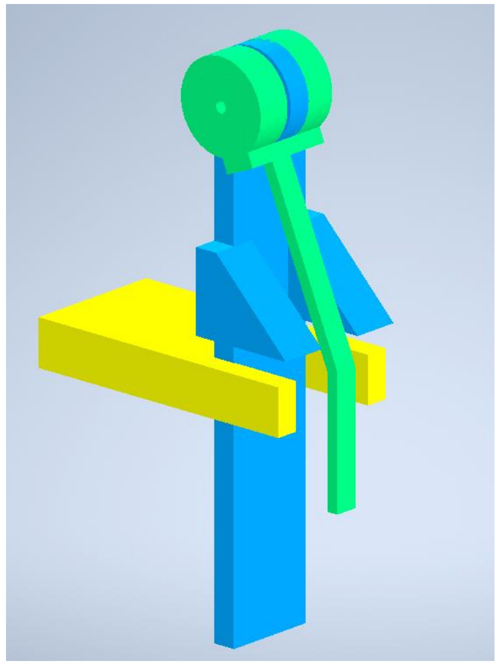

Figure 11: Basic structure of the string-pulling device.

6.2 String-pulling Device

For the string-pulling device, the detailed machenism is still needed to be finalized. The basic struct will be similar to figure 11. An electromagnet will be attached to the blue structure and attract the metal block attached to the green structure. The pulley will be attatched to the blue structure. And by adjusting the electromagnet output power, the friction between the pulley and the green structure will increase/decrease accordingly, which makes the pully harder/easier to be rotated. The string will be placed on the pulley.

The load cell connects the blue and yellow structures to measure the pull force on the string (when the subject pulls the string, the entire blue and green structure is experiencing a downward force), where the yellow structure is firmly attached to the training enclosure.

The pulley should have hollows so that when it rotates to different angles the optical sensor can notice the light density changing. Then by measuring the pulley diameter and the light density variation time, the software can calculate the pull velocity and pull distance.

6.2.1 Alternative Solution (Plan B)

As discussed in section 9 Risks and Mitigations, the design of an electromagnetic control system requires tedious and fragile electric circuit design, which means using the electromagnet to attract the blue and yellow structure may fail. Here we proposed an alternative solution, where the stepper motor will be attached to the blue structure. The motor’s gearhead will pull the green structure through a robust string. Thus, the friction between the green structure and the pulley will be adjusted as the motor rotating clockwise/counterclockwise.

7 Detailed Software/Firmware Design

7.1 Firmware Design

The firmware components are implemented as Python classes with support packages.

- GPIO Reader/Writer FC-01 transmits signals between all other firmware components and physical GPIO ports in Raspberry Pi. This library is included in the Pi [ref]. It directly maps GPIO analog signals from sensors (namely the load cell, optical sensor, and digital potential meter) into appropriate digital signals to the hardware manager for further computation. We use the optical sensor model LTH-301-32 [1] with MCP3008 driver [2], use the load cell model SEN-14729 [3] with HX711 [4] driver, and use the digital potential meter model MCP4151 [5].
- Stepper motor controller FC-02 maps arguments passed by the hardware manager into appropriate GPIO analog signals to administrate the stepper motor. We use the stepper motor model 28BYJ-48 [6] with ULN2003APG driver [7]. The corresponding Python driver library is RpiMotorLib [8].
- Electromagnet controller FC-03 maps arguments passed by the hardware manager into appropriate GPIO analog signals to administrate the electromagnet. We use the electromagnet model 3873 [9] (no driver required). The corresponding Python support library is Spidev [10].
- Firewall FC-04 is configured to allow the communication between the client device and Raspberry Pie. It is done by editing the router's configuration page.

7.2 Software Design

7.2.1 Web Server

We apply Flask Dashboard Datta Able [11] as our back-end server template, which provides fully developer-centric code for SC-01 to SC-04. Adapted from the template, we code UI elements for the web page in the HTML file, where the user request response can be supported with the Ajax jQuery method [12] written in JavaScript. Meanwhile, the Flask python script [13] serves as an engine consisting of several routers responsible for web page rendering and event handling. We use SQLAlchemy toolkit [14] to manage our database, which is an object-relational mapper that both web server and hardware manager can utilize.

Listed below are the required functionalities and their proposed solutions for the web server.

- Web Server SC-01 hosts the web page at the LAN IP address assigned by the physical router (in figure 3, it is 192.168.0.40) with a specified port number (e.g., 5000), where their combination (192.168.0.40:5000) enables the user to access the posted data through their internet browser under the same LAN. Flask package provides methods for hosting the web page at one

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

unreserved port. On the other hand, once the Raspberry Pi is connected to LAN, the user can search its IP address with IP Scanner software [15].

- Authentication SC-02 is the subcomponent of the web server. It provides the interface to register the user, which prevents the potential laboratory data breach. When the product is released, we will provide backdoor administrator accounts for registering other normal users. Those backdoor accounts will be coded in the configure file in the software package, where the client can modify later. The Flask Dashboard Datta Able template already provides minimum user registration options, and we can adapt our implementation from there.
- Training configure SC-03 is the subcomponent of the web server. It communicates the user input training parameter (i.e., training duration, required pull force, pull distance, and maximum allowable time window to complete one task) to the hardware manager. Once the user enters the parameter and hits the submit button, jQuery will get the input data and post it to one specific Flask handling routine. The parameter is recorded to the database, and the hardware manager is invoked (as the yellow area in figure 14 illustrates). The routine ends when the hardware manager finishes the process (i.e., timeout), and jQuery will report the status to the web page.
- Data retrieve SC-04 is the subcomponent of the web server. It allows the user to query one specific piece of information in the database. This procedure is similar to what we performed in training configure, where jQuery transmits data between the web page and Flask router (as the yellow area in figure 13 illustrates).

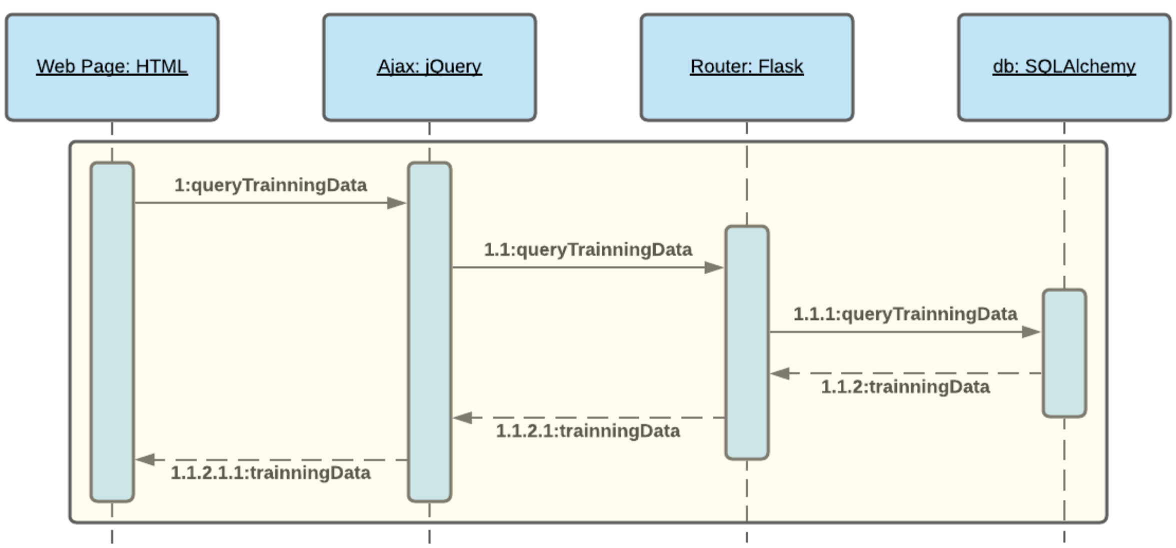

Figure 13: Sequence diagram for querying training record.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

7.2.2 Hardware Manager

The implementation of the hardware manager SC-05 is straightforward as it does not require sophisticated computing algorithms. However, it must be tested thoroughly within the training enclosure as most of our designed hardware is sensitive and fragile. It is integrated with firmware components FC-01 to FC-04 and has direct access to all hardware components. Moreover, we decide to authorize the hardware manager to write training data into the database (rather than communicating the metadata back to the web server to process) to minimize the transmission delay.

Listed below are the required functionalities and their proposed solutions for the hardware manager.

- The hardware manager is implemented as a Python class which takes the training parameter received from the web server as arguments and interprets them into an appropriate numerical value (e.g., translates the required pull force into the digital potentiometer input value, where the latter controls the electromagnetic strength). It then configures the corresponding hardware (namely the stepper motor and digital potentiometer) by invoking the firmware component.
- The hardware manager also interprets the real-time metadata received from the hardware (namely the loadcell and optical sensor) into appropriate training data (e.g., translates the load cell reading into the pull force), records it into the database, and performs further action accordingly. The green area in figure 14 illustrates the above procedure, where the hardware manager keeps querying the real-time metadata if there is remaining training time. Meanwhile, if the data collected so far indicates that the training subject completes one task, the hardware manager will dispense food pellets as a reward and resets the training goal. Since the received data is technically discrete, it is critical to balance the query interval (e.g., $1000ms$) and the storage space to maximize the measurement precision while minimizing the redundancy information.

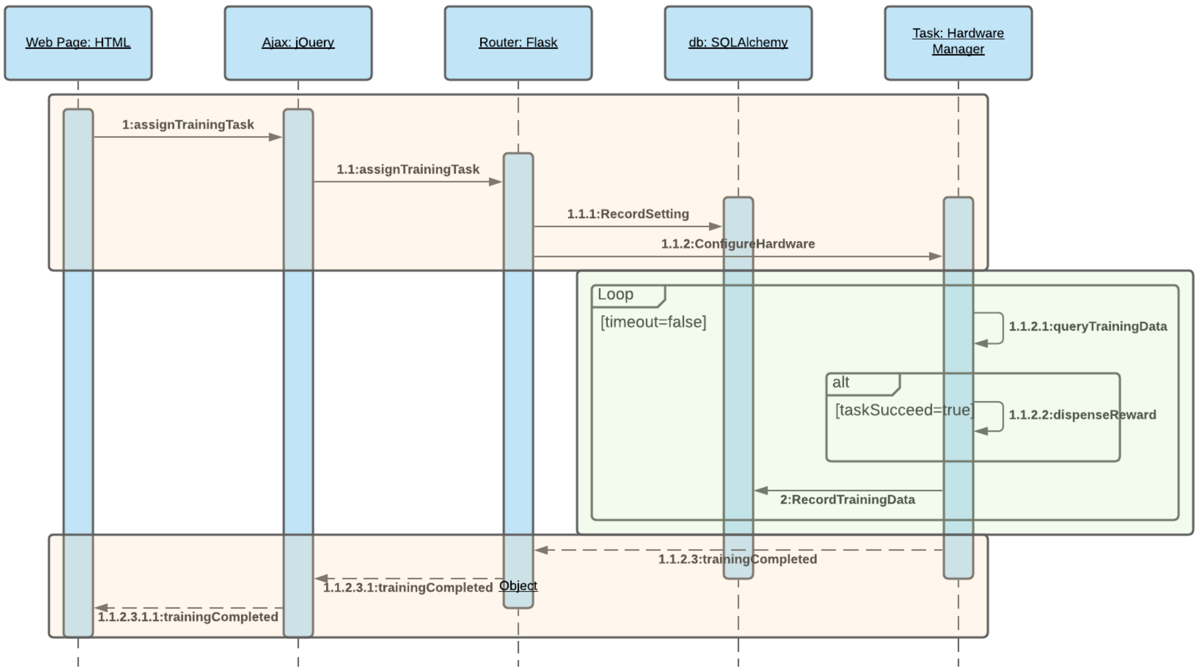

Figure 14: Sequence diagram for conducting one training experiment.

7.2.3 Other Software Components

- Database SC-06 stores all the training information, R/W by the web server and hardware manager, and all the account records. Figure 15 illustrates the table schematic of our database, which is in the form of the SQLite file. It is organized in a way that is robust for experimental investigation. The configure table records the training parameter, the experiment table records the training information, the subject table records the training subject information (there might be multiple subjects participate in one experiment), and the metadata table records the real-time training data. We decouple the configuration from the experiment, so the user does not need to re-enter the training parameter every time a new experiment starts.

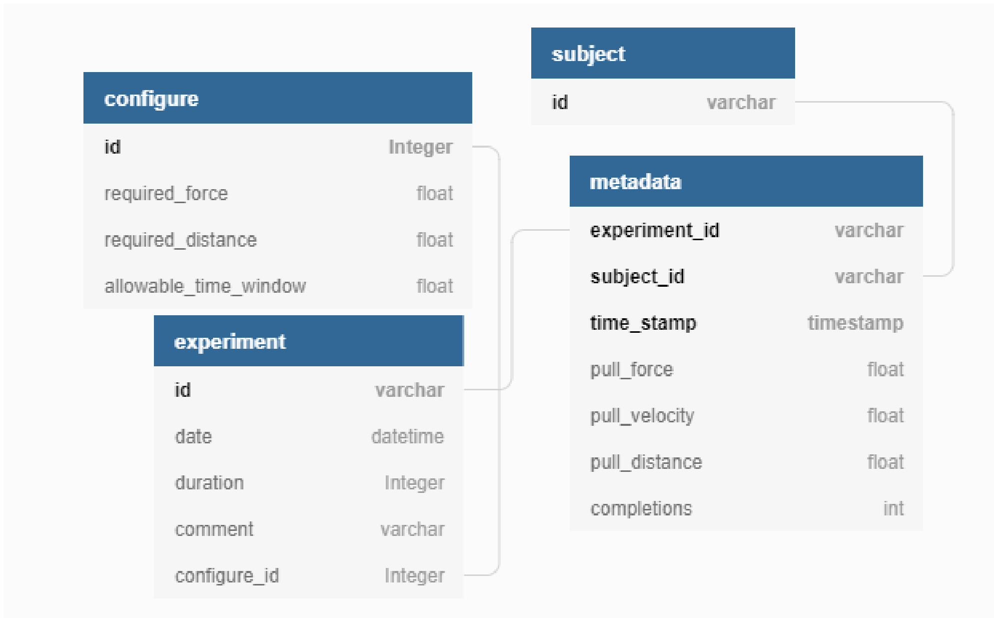

Figure 15: Table schematic of our database, generated by dbdiagram.io [16].

- Web Browser SC-07 provides an interface for the user to access the server. It can be any commercially used application (e.g., Google Chrome, Safari, etc.).

8 Landed Production Cost

Table 7 Landed Production Cost Budget Estimate for units based on production of 1000

Component	Mfr P/N	Mfr	Qty	Unit Price	Extended Price
Raspberry Pi 4 (4G RAM) [17]	99467	CanaKit	1	CDN$ 134.99	CDN$ 134.99
Raspberry Pi GPIO Breakout Expansion Board [19]	B07425G6D6	UCEC	1	CDN$ 11.99	CDN$ 11.99
Transistor 2N2222 [20]	2N2222 PBFREE	Central Semiconductor Corp	3	CDN$ 1.60	CDN$ 4.80
100Ω resistor [21]	CF14JA100R	Stackpole Electronics Inc	2	CDN$ 0.005	CDN$ 0.01
Electromagnet [9]	3873	Adafruit Industries LLC	1	CDN$ 14.39	CDN$ 14.39

3D-printed Dispenser			1	CDN$ 50(estimate)	CDN$ 50
Smooth Belt 1/8" [22]	ROB-2152	Actobotics	1	CDN$ 4.99	CDN$ 4.99
Analog-to-Digital Converter [2]	MCP3008-I/P	Microchip Technology	1	CDN$ 2.65	CDN$ 2.65
Digital Potentiometer [5]	MCP4151-502E/P	Microchip Technology	1	CDN$ 1.08	CDN$ 1.08
3D-printed Pulley System			1	CDN$ 50(estimate)	CDN$ 50ca
Load Cell [3]	4541	Adafruit Industries LLC	1	CDN$ 5.37	CDN$ 5.37
Load Cell Amp [4]	SEN-13879	SparkFun Electronics	1	CDN$ 14.39	CDN$ 14.39
Stepper Motor 28BYJ-48 [6]	28BYJ-48	Longrunner	1	CDN$ 4.198	CDN$ 4.198
Total Cost					CDN$ 297.858

9 Risks and Mitigations

Table 8 Risk assessment.

Risk ID	Risk Name	Risk Category	Description	Mitigation	Risk Level
R-01	Magnetic Field Problem	Technical	The magnetic field produced by the electromagnet will affect the performance of some circuit components that require high precision.	Put away the electromagnet from devices like load cell and optical sensor.	Medium
R-02	Magnetic Force Control Problem	Technical	The digital potentiometer resistance is large compared with the electromagnet, making it difficult to control the magnet's voltage.	Increase the number of electromagnet or replace electromagnet with a stronger one.	High

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

R-03	Friction Coefficient Problem	Technical and Cost	Our design plans to use the electromagnet to provide different normal forces (i.e., the force between the green structure and pulley in figure 11) to get different friction forces. Since the required force is minimal (around 0.5 newtons), we need a small friction coefficient, which may need expensive materials.	Use lever structure to reduce the normal force.	Medium

R-02 explanation: the digital potentiometer is the way we used to adjust the voltage across the electromagnet. The digital potentiometer we used has 5 $k\Omega$ with 258 steps (about 20 $\Omega/step$). But electromagnet is a low resistance component, having a resistance around 20 Ω. Therefore, as the potentiometer increasing its output resistance from 0 to 5000, the voltage among the electromagnet drops quickly at the beginning. Our mitigation to this problem is only used in the middle steps where the voltage changing continuously. This solution will narrow the range of the force.

10 Certification and Licensing

The final design of this product will only be used in laboratory.