AutoErasing

Authors: Jiayi Wang, Wenqi Deng, Xiaoyu Chai
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A system capable of taking pictures of and erasing and projecting on a section of a whiteboard upon instructions from a web application. A Raspberry Pi will be communicating with a web application and controlling the camera and motors. This product would reduce the effort of manual erasing for instructors, as well as provide capabilities for instructors to project back the erased content.**

***Index Terms*—Virtual Board, Automated Erasing, Stepper Motors, Solenoid, Raspberry Pi, Web application**

1 INTRODUCTION

In lecture scenarios, while instructors do not have enough board space to write on, they would usually have to erase the content of the board multiple times during one lecture, and it's difficult for them to refer back to some specific notes from the erased content. In order to solve this problem, our team works on a project that could provide unlimited "virtual board" space, that is to automate the process of taking pictures, erasing the board, and uploading the content to a website, which allows instructors to display the erased content back to the board again when there's a need to refer to some erased content.

While there exist some automatic board erasers with the functionality to erase the entire board, our project provides additional flexibility by allowing instructors to erase part of the board instead of the whole board. On our web application, instructors can customize the board into different sections of different sizes, based on their writing preferences. These sections can be erased independently. Also, our project provides a novel way of assisting instructors by taking pictures of the erased content and making it available for later use through the website.

The basic goal of the project is to save instructors' effort in repetitively erasing the whiteboard during lectures. Aside from that, we want our project to extend beyond a mechanical system to a helpful teaching tool. Our camera functionality, should be able to provide instructors much convenience in referring back to lecture content.

2 USE-CASE REQUIREMENTS

Based on the application or problem we have described in the Introduction (section 1), we have identified the following use case requirements considering the needs of instructors and students in classes.

2.1 Latency

In order for our erasing system to be useful during classes, we would expect that our erasing functionality will take no longer than the time it takes for a human to erase a board of the same area. After manually doing repeated erasing testing, we concluded that the time it takes for a human to erase one section of the board we will use for demonstration (18" $\times$ 24") is 10s. So our system will aim at most 10-second latency for erasing features.

For our image capture functionality, in order to provide the students with a seamless experience in the note-taking process, we have set a goal of uploading captured images to the website within 3 seconds since this is the average time taken for a website on the standard internet connection to load. This means that once students refresh the website after the content gets erased, they will be able to see the uploaded images and continue taking notes.

2.2 Cleanness

Another important factor we consider is that our system should effectively clean the board, determined by whether observers believe the board is clean to continue writing on it or not. We are aiming at a 100% success rate because if the system fails to erase the board cleanly, the instructor has to send erasing instructions again or manually erase it, which fails to achieve our use cases.

2.3 User experience

Regarding user experience, we are aiming to build an easy-to-navigate user interface for new users to get familiarized with within 15 minutes, which is the average time taken for users to get started with relatively straightforward applications with intuitive user interfaces and limited features. This includes the time taken for them to learn how to configure classes, boards and camera, choose which section to erase, and access images of the board captured.

2.4 Picture Readability

For the picture-taking functionality, we expect the content on the whiteboard to be readable in the picture, just as they are reading it on the whiteboard. In other words, users should be able to recognize 100% of the content on the picture, just as they are sitting in front of the board.

2.5 Scalability

In order for our project to have a wider application, it should be able to work on classroom boards of different di-

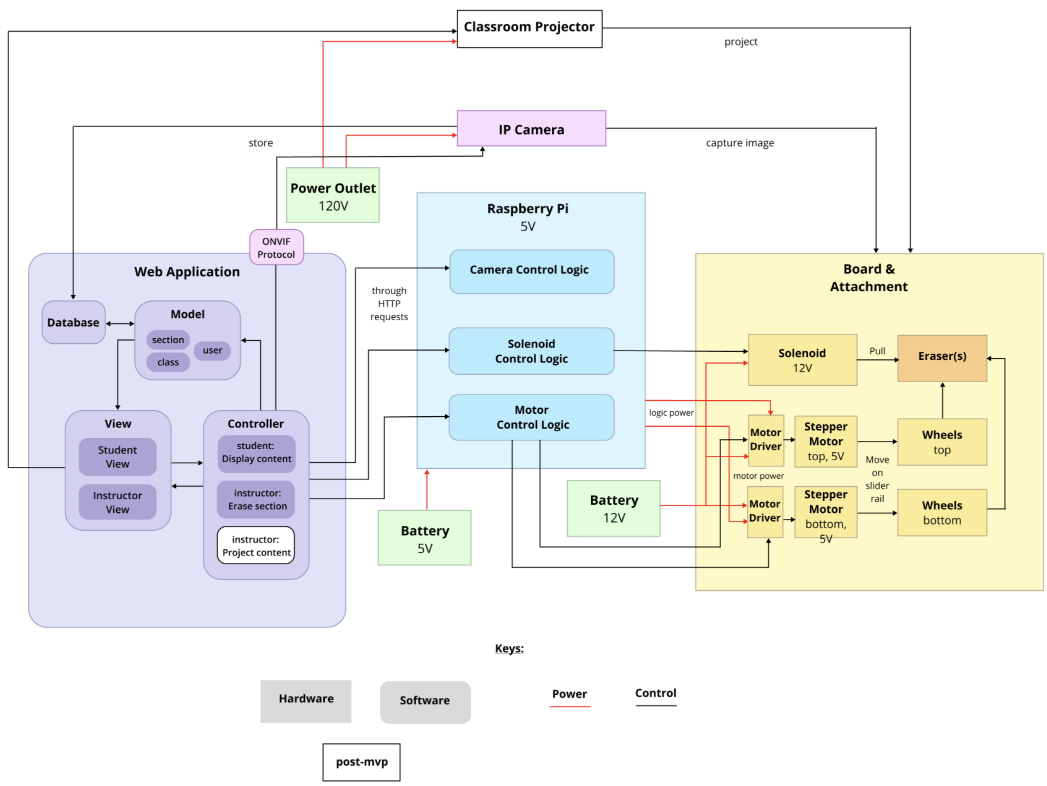


Figure 1: Block Diagram for the Entire System, see Fig. 6 for a full-page version

mensions. The web application should allow instructors to set the actual size of the board, and the erasing and image-capturing functionality controlled by RPi should function normally according to board dimension settings.

2.6 Battery Life

For the power supply, since our system should function as manual erasing for at least one lecture without charging, we decided to aim for at least 80 minutes of battery life while using it to simulate the frequency of erasing during lectures, while 80 minutes is how long typically a lecture will last.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

3.1 Components on Whiteboard

Our system is designed to be an external attachment to classroom boards. For this project, we will buy a whiteboard and assemble components onto it. These attachments include erasers, metal plates, slider rails, and wheels. The slider rails will be 3D printed as subparts at Tech Spark and combined together into a long slider rail based on the dimensions of the boards, and attached to the top and bottom of the whiteboard supported by metal corners. Wheels will be connected to motors and will rotate on the slider rail. A long eraser will be controlled by the wheels and move across the board to erase. The metal plate serves as a support that holds erasers and other components together.

3.2 Hardware Components

3.2.1 Stepper Motor

Connecting to the motor driver and then Raspberry Pi, two stepper motors will be used to drive the erasers. The movement will be horizontal along the slider rails. Raspberry Pi will keep track of the current position of the erasers and send specific instructions to the motor through a motor driver to move the erasers.

3.2.2 Solenoid

We plan to use a solenoid to push or release the erasers on the board. When the erasers are moving to their designated position, the solenoid will pull back, allowing space between the erasers and the whiteboard so that no content on the board will be accidentally erased. When erasing, the solenoid will push the erasers against the board and apply appropriate pressure to eraser the board cleanly.

3.2.3 IP Camera

A wireless IP camera, which will only be connected to the power supply and placed on or attached to the classroom podium facing the board, will be used to record the content of the whiteboard right before erasing happens. The web application will be able to access real-time video streaming at the designated IP address over a network in the backend, and upload the content to the website on both student view for them to review and take notes and instructor view for them to refer back as well as use in a potential projection feature we are considering as a post-MVP implementation.

3.3 Hardware Controller

We plan to use Raspberry Pi as a microcontroller to control the hardware components including stepper motors and solenoids. Raspberry Pi will also be communicating and receiving information from the web application, and execute erasing and image capturing instructions.

3.4 Web Application

Our web application, including an instructor view and student view, will be set up using the Django framework and deployed on an EC2 AWS server. It will allow instructors to create classes, invite students, and configure board settings and camera positioning before lectures (section division, number of boards etc.). During lectures, instructors can navigate to the page of the current class and select the section of the board to be erased. Through ONVIF protocol to access the IP address of the camera, the backend will be able to obtain the captured picture of the section before erasing and command the hardware components to physically erase the section. At the same time, it will display the captured picture in a list at the side of the current page in student view and instructor view, for students to refer to if they wish to continue note-taking and for instructors to keep track of content taught previously and potentially project it back to the board if they want to refer to topics discussed previously. The website will serve as the main interface for users to interact with our system and use each of the functionalities, as well as a central information hub that will store important data in its database and retrieve it to display to the users when needed.

4 DESIGN REQUIREMENTS

In response to our latency requirement, we will have a minimum speed requirement for the motor. During an erase action, the motor needs to first move to the starting position, then move back and forth twice to clean the board. Thus the erasers need to travel 5 times the width of one section. Therefore the speed of erasers will be

$$v = \frac{d}{t} = \frac{5 * 18'' * 0.025''/m}{10s} = 5 * 18 * 0.025m = 0.225m/s \quad (1)$$

Our wheel has a diameter of 80mm, which means that one revolution of the motor would give

$$C = 2\pi r = 2 * \pi * 0.08m = 0.5m \quad (2)$$

Thus, the speed of the motor in rev/s would be

$$\frac{0.225m/s}{0.5m/rev} = 0.45rev/s \quad (3)$$

Our image capturing functionality should take at most 3 seconds from the start of the erasing process to the image being uploaded to the website to ensure a seamless experience for our users, especially the students, to continue viewing the erased notes on their own devices. This process goes through multiple subsystems including the latency in the camera module, the latency in communication between the backend of the website and to camera, and the latency of retrieving the images in the database and uploading them to display at the frontend. To ensure we have minimal latency experienced by our users, we have set realistic latency goals for each of our subsystems: camera module taking pictures (2s), sending pictures to the website backend (500ms), retrieving the images in database (300ms) and upload it to display at the frontend (200ms).

In addition, the image captured must be readable under various potential lighting conditions. Our camera needs to have a high resolution of at least 720P, and the ability to tilt to avoid glares under special circumstances.

The entire system should be able to support all the functionalities for at least an entire lecture, which means the batteries need to continue powering the system for at least 80 minutes, which is the most common length of a single lecture.

Lastly, our web application should display all the information and options users might need to use and allow them to interact with all the features without significant difficulties. The instructors should find it easy to create a class, add students, configure the board with the current board setting given, Erase a certain section of the board, and access the captured image of the first section, while the students should also find it convenient to access the captured images.

5 DESIGN TRADE STUDIES

5.1 Hardware

5.1.1 Motor

We picked a stepper motor instead of a regular DC motor because it provides easier and better position control of the erasers. Since our product will not be erasing the entire board, position control is important because we don't want to erase content on another section accidentally. Our previous experience in DC motor position control using Proportional-Integral-Derivative control was very unstable and had oscillations. We've also considered a servo motor, but it's hard to adjust the speed. Also, there is not much cost difference between stepper motor and regular motor. Therefore we decided to use a stepper motor in our project.

5.1.2 Solenoid

Since solenoids cannot be powered on for long, otherwise it would get very hot, we decided to use a pull-type solenoid in order to minimize the time required for the solenoid to be on. We think that the time it takes to quickly move the eraser to the desired position, which is the time required for pulling, should be less than the time it takes to erase the board, which is the time required for pushing.

For other specifications, our solenoid needs at least 1 cm stroke length, to account for the elasticity of erasers and ensure that erasers will not be touching the board when the solenoid is pulling. We also tested the force it takes when a human presses the erasers on the board, which is 5N. Other than that, the dimension of the solenoid should be as small as possible, so as to not add too much weight to the system.

5.1.3 Erasing Mechanism

In terms of erasing mechanism, we first considered using a robot arm to move the erasers, because it imitates actual human behavior in erasing. However, we realized that it is too complex and expensive. We then turn to think of moving an eraser linearly across the board using slider rails. We decided to only aim at horizontal movement because of the potential weight issue in vertical movement. We worry that the weight of our entire components could be pulling the erasers down, which makes it difficult to move vertically up. Also, in real life instructors rarely erase only the upper half or lower half of the board, which suggests that deleting vertical movement would not affect our use case requirement much.

5.1.4 IP Camera

After researching the different options of cameras online, our team has chosen to use a wireless IP camera. While USB cameras provided in the inventory can only transmit data through the USB connection, we decided to purchase the IP camera, mainly because it can send and receive data over an IP network, which enables it to communicate with our web application through the internet. Specifically for the model we have chosen, it can pan 320°, and tilt 90° vertically, allowing it to adjust to the angle that best captures the specified section on board, which offers more flexibility than the Raspberry Pi HQ Camera. In addition, it has a 1080P HD resolution, which will improve the readability of the captured images.

5.1.5 Raspberry Pi

In our system, we need a microcontroller to control the movement of hardware components. We decided to use Raspberry Pi 4, instead of Arduino or other microcontrollers because this version of Raspberry Pi supports onboard wireless networking, which enables it to communicate with the web application of our system and serve as a central hub to transform general action commands into different logics acting on different hardware components

5.2 Software

5.2.1 Web Application

For the interface for users to interact with, we decided to create a website for users to view and modify the information stored in the database, as well as send commands to communicate with the hardware components of our system. The main reason for choosing a web application is its accessibility across different devices, which eliminates the need to download, install, and manually update our application. In this way, it takes minimal setup time and ensures any student or instructor with any device can connect to it with an internet connection. We also plan to investigate how to integrate our web application with classroom tablets, so instructors won't need to use additional devices during the lecture to interact with the system.

5.2.2 AWS Deployment

Regarding our web application, we have the option to either run it locally or deploy it to a cloud server. Our team has decided to deploy it to Amazon EC2, since this option provides more scalability and availability. Since we want to provide the capability for instructors and students to access the captured images after classes with a different network connection, it's necessary to deploy the website for public access.

5.2.3 OAuth Login

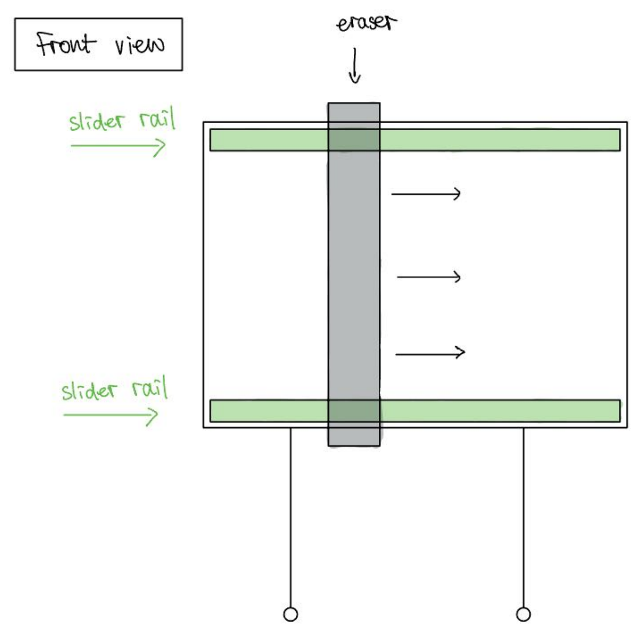
As users need to sign in to the website to use the functionalities of our systems, we have also considered different options for users to register and sign in and eventually decided to use third-party APIs, namely Google OAuth, for our users to sign in with email, while it will automatically register new users at the first time signing in. With Single Sign-On enabled, OAuth login can save the set-up time for our users to access the service. In addition, combined with two-factor authentication, this protocol can improve the reliability of authentication of users' identities, so our web application can manage access for different groups of users, for example, limiting access to hardware components at CMU to CMU faculties only.

5.2.4 Django

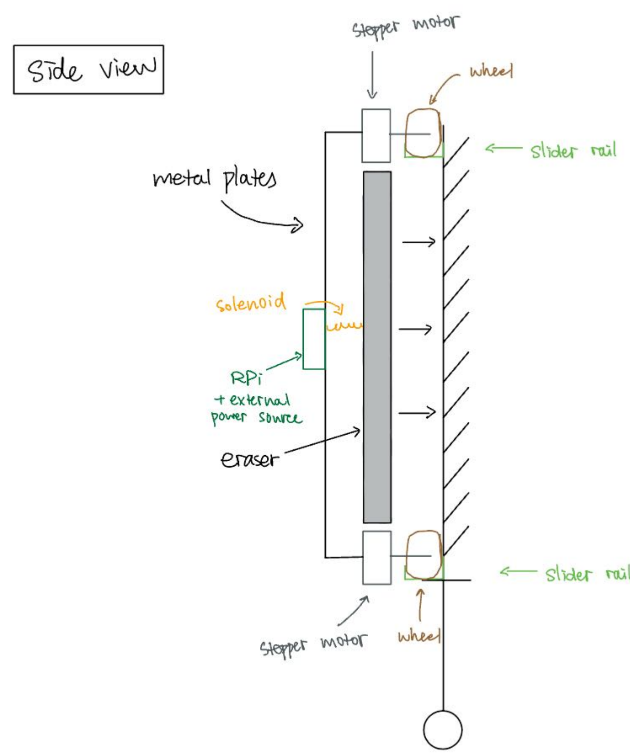
For the development and implementation of our web application, our team has chosen Django framework, which is a Python-based web framework that offers a well-structured Object-Relational Mapping system to simplify the operations interacting with the database where data models are defined using Python classes. Django also includes a wide range of reusable components and libraries, reducing the need for third-party plugins and allowing for easier setup and development.

6 SYSTEM IMPLEMENTATION

6.1 Whiteboard Attachment Wiring and Power Supply



(a) Front-view of Whiteboard



(b) Side-view of Whiteboard

Figure 2: Diagram of Physical System (a) Front view (b) Side view.

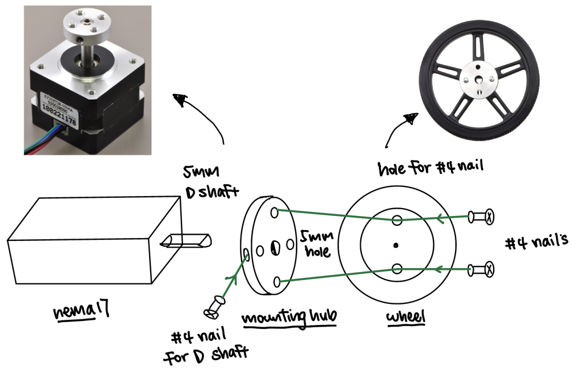


Figure 3: Connection between motor and wheel

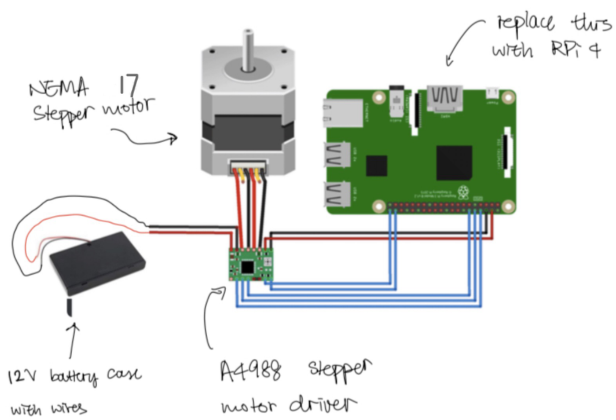


Figure 4: Connection between motor and driver

Above are diagrams of our whiteboard setup. Two slider rails would be nailed tightly to the whiteboard on top and bottom. It will be U-shape, to keep the wheels on track. We plan to 3D-print the rails because it is easier for us to adjust the dimensions to fit the whiteboard and wheels and generalize our system to other settings. The wheels will be connected to the stepper motors with 5mm D-shaft, through the help of an aluminum mounting hub for 5mm shaft. The motors will be able to drive wheels to rotate on the rail parallel to the board, moving the line of erasers held together by a metal plate to the designated position and cleaning the board. We will use wires to connect the Raspberry Pi to motors and solenoids so the microcontroller can directly control those components to complete the erasing mechanism.

Aiming to capture images before the content gets erased, our IP camera, connecting to the power outlet, will be placed on or attached to a platform such as a podium or a desk for instructors, and adjusted to the right angle that can best film the board using its built-in pan and tilt function.

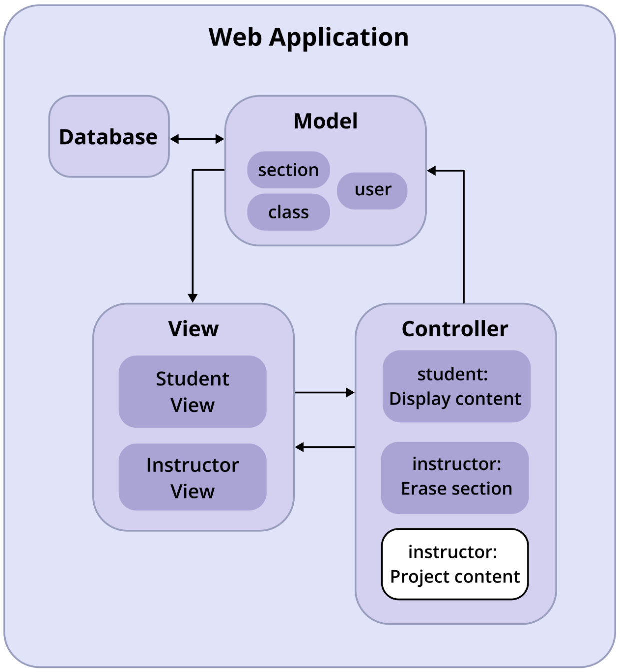
Regarding the power supply of our system, we will use Raspberry Pi to supply power to the camera and motors with a voltage of 5V. But for the solenoid, since it requires a voltage of 12V, we will use an external power source, namely a battery for its portability, to power it. In addition, we will also use a 5V battery to power our Raspberry Pi.

6.2 Image Capture Mechanism

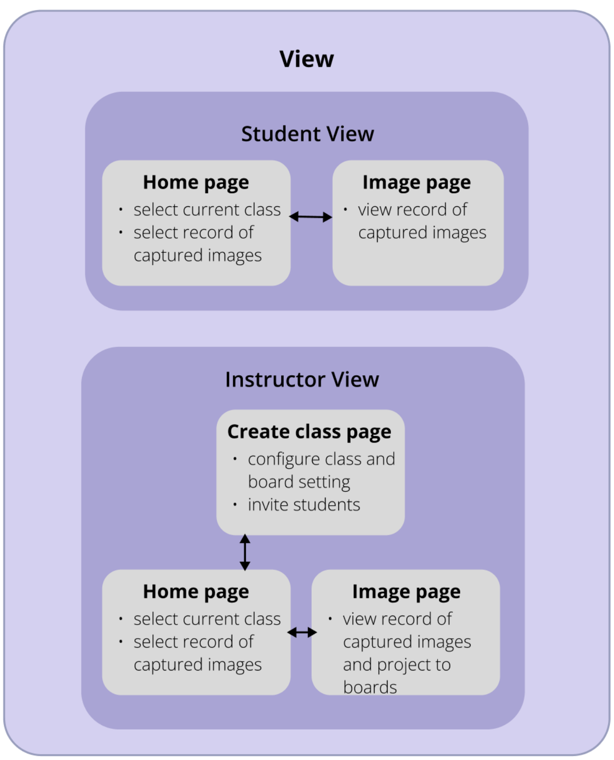
A camera module will be incorporated into our design to capture the content of the specified section of the board before it is erased by uploading an image at that moment to the web application for users' reference. We have decided to use an IP camera, which has an IP address that makes it accessible over the Internet. With the camera model we've chosen, the built-in pan and tilt functionalities can help to adjust to the angle facing the specified sections upon receiving the corresponding command from the web application. While commanding the Raspberry Pi to erase the content on board, the web application will also access the real-time video streaming of the IP camera at the designated IP address in the backend, and upload the content at that moment right before erasing captured to the website.

In our design, the physical movement of the entire metal plate holding the eraser, Raspberry Pi will rely on stepper motors, which will move the plate horizontally across the board through wheels. After taking the image, the web application will send information about the left and right coordinates of the section to be erased. Once receiving the coordinates from the web application, RPi will turn the solenoid to the pull position so it holds the erasers back to ensure it does not touch the board, and move the metal plate to the left coordinate. Our system will then control the motor and solenoid to physically erase the board. The physical movement of board erasers depends on the actions of the stepper motors and the solenoid, driven by RPi. After image capture is completed, RPi will then turn the solenoid to the push position, applying appropriate pressure to the erasers, while the motors will move the erasers back and forth twice within range to erase the content. Then it would signal back to the web application that the erasing is completed.

6.3 Web application Implementation



(a) An MVC Layout of the Web Application



(b) Zoom-in of views

Figure 5: Web Application description. (a) A high-level MVC layout. (b) A brief overview of pages and interactions.

Serving as the interface for users to interact with different functionalities and the center of communication, our web application has a frontend and backend. The website frontend will use HTML to build the skeleton of the pages and specify elements on different pages, and CSS to format the way different elements appear on the page to make it more user-friendly. For the backend, we have chosen to set up the website using the Django framework and program the functionalities in Python. The instructors will need to create a class, add students and configure the board setting and camera positioning before accessing other system functionalities. Once receiving the command of erasing a certain section from the users, the backend will refer to the board setting stored in the database that is inputted by users to calculate the corresponding left and right coordinates of the section and send it to Raspberry Pi for it to control the movement of motors. At the same time, it will use the left and right coordinates as well as other manually inputted parameters about its relative positioning from the board stored in the database to calculate the pan and tilt angle so it can film the section of the board. We will use the ONVIF protocol to control the pan and tilt of the camera and save a picture to the database. Then, the Django framework will retrieve the image files from the database and display them on the frontend of our web application.

7 TEST & VALIDATION

7.1 Tests for Erasing Cleanness

In order to test the effectiveness of our erasing system, we plan to recruit participants to judge whether different board sections after erasing are clean enough multiple times. Our goal is to have participants agree system erasing is clean at all times. If we fail to achieve this goal, we will adjust the speed of the stepper motor and the pressure of the solenoid, and re-perform the test again until we achieve our goal.

7.2 Tests for Picture Readability

To test the readability of the picture taken from the camera, we plan to conduct a comparison test. Participants will be asked to read the text content on the board, once directly looking at the whiteboard and once looking at the pictures. We will do tests under different lighting conditions to ensure the glare doesn't influence the readability. We expect that the content they read will be exactly the same. If such a test fails, we will first try to adjust the angle and position of the camera or purchase a new camera with better resolution.

7.3 Tests for Low Latency

In order to meet the requirement of low latency, we will measure the latency time of each subsystem such as motor and solenoid for unit testing, while also measuring the time it takes for the entire system to complete erasing a certain section. We aim to erase one section of the board we will purchase (18"x 24") in less than 10s as the reason discussed before. If we fail, we will identify latency bottlenecks based on our unit tests and optimize the latency based on which part is identified.

In addition, for our image capturing picture feature to meet the low latency requirement as well, we will also measure the time between erasing start and the image being uploaded to the internet on the backend, aiming for a latency of at most 3 seconds, which is the average time for a website to load. In other words, the students should expect to see the uploaded images after reloading the website at the start of erasing.

We will repeat both experiments, calculate the average latency time for both features and compare them with our goals.

7.4 Unit Tests

7.4.1 Motor

We will unit test the functionality of the motor controlled by RPi. We will write RPi testing program that moves motor in a certain direction to a certain distance, and see if the actual movement of erasers is accurate. We would expect an error within 5mm in final position.

7.4.2 Solenoid

We will unit test the solenoid controlled by RPi. We will write RPi programs that switches the solenoid between pull and push positions every 2s (the longest time for the eraser to move across a section to pass the latency test). We would expect that the solenoid successfully switches position and does not overheat within 30s.

7.5 Tests for User Power Supply

To measure the power consumption, we plan to simulate the frequency of erasing the board during a typical 80-minute lecture by going through some of the previous lecture recordings we can access online and command our system to erase whenever the instructors erase the board during the recordings. We will repeat the experiment for at least 3 times and record whether if the batteries succeed in supporting the system to run through the entire class for each individual experiment, and we hope to achieve a 100% success rate.

7.6 Tests for Usability

We will recruit two groups of participants to test both the instructor view and student view of our website. For both of the two groups, we hope the participants could take an average of less than 15 minutes to understand how to use every functionality on the website and collect their feedback on the usability of the website. For our primary stakeholder instructors, we plan to ask them to perform the following tasks:

- Create a class
- Add students with ID X, Y, Z in the class
- Configure the board with the current board setting given
- Erase the first section of the board
- Access the captured image of the first section

Then, for our secondary stakeholder students, we plan to have them complete the following actions:

- Navigate to the class page for class X
- Access the captured image of the first section

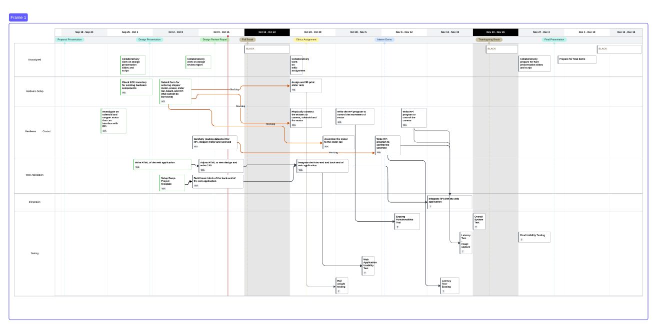
After completing those tasks, we will collect feedback from the participants by asking them a series of questions, about the different aspects of our test metrics:

- Latency:
  – *"Did you feel any noticeable lag in erasing/image capturing?"*
- User Experience:
  – *"Is the website intuitive to use?"*
  - *"Would you prefer using this system or manually erasing?"*
- Accessibility:
  – *"Were you able to access the images easily?"*

By asking those questions, we hope to understand the usability of our system from users' perspective. We plan to conduct the test with at least 3 users in each group (3 instructors and 3 students). We hope to receive a standard response of "No" for latency and "Yes" for user experience and accessibility, with at most 1 non-standard response for each section.

8 PROJECT MANAGEMENT

8.1 Schedule



Our schedule is shown above. Refer to Fig. 7 for a full-page version. The arrows indicate that a previous task is blocking a subsequent task.

8.2 Team Member Responsibilities

Xiaoyu: Raspberry Pi programming

Jiayi: Web application back-end setup

Wenqi: Web application front-end setup, camera control and image uploading functionality

All: Integration of system, whiteboard setup, testing.

Table 1: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Stepper Motor	1207	Pololu	2	$18.95	$37.90
Stepper Motor Driver	1182	Pololu	2	$13.95	$27.90
Stepper Motor brackets	2257	Pololu	2	$3.95	$7.90
Solenoid	FLT20190821M-0051	Fielect	1	$14.99	$14.99
Raspberry Pi	RPi 4 8GB	Raspberry Pi	1	$0	$0
Camera	AT-200DW	Alptop	1	$39.99	$39.99
Wheel	1424	Pololu	1	$5.75	$5.75
Mounting Hub	1203	Pololu	1	$8.95	$8.95
Whiteboard	WB3624L	Zhengzhou Aucs Co.,Ltd.	1	$31.79	$31.79
Erasers	SAN81505-8	Newell Rubbermaid Office Products	1	$20.99	$20.99
Battery	DURMN21B4PK	Duracell Distributing, Inc	1	$6.48	$6.48
Steel braces	N351-461	National MFG/Spectrum Brands HHI	1	$30.83	$30.83
Steel Corners	N/A	Blingstar	1	$7.99	$7.99
Slider rail	N/A	Tech Spark	1	$0	$0
					$241.46

8.3 Bill of Materials and Budget

Refer to table 1. The majority of our budget goes to the whiteboard and other mechanical components for building the erasing system. We are considering 3d-printing some materials, in order to save more budget if necessary. In the future, we might spend more budget on buying whiteboards of different sizes to verify our design. We saved some budget by borrowing materials from ECE inventory. Table 1.

8.4 Risk Mitigation Plans

One major risk of our project is the mechanical design. No member of our team has experience in mechanical engineering, so we are unsure whether our design for the erasing mechanism will work out. To mitigate that risk, we had many discussions about the design of physical systems and prepared some back-up plans. We have also allocated sufficient time in our schedule for the whole group to assemble the mechanical system.

Another risk is that we do not have experience in wireless communication from web applications to Raspberry Pi. Our risk mitigation plan is that Xiaoyu, who has experience in embedded systems, will research that topic.

We are also worried about the actual erasing effect of moving erasers across the board. To mitigate this risk, we will repeatedly adjust the speed of the motor and pressure of the solenoid, and choose the combination that has the best effect. If necessary, we can also buy different motors and solenoids to test their effect.

9 RELATED WORK

We found several similar automated erasing products online. One of them used the Internet of Things to move erasers across the board upon button press to erase the entire board. Other products have similar mechanisms. However, all of the products we researched are just about mechanically erasing the board. In comparison, our product would allow instructors to take pictures and reuse erased content through integration with web applications and projectors.

10 SUMMARY

In conclusion, our product will be a system on the whiteboard that can receive instructions from a web application to take pictures and erase the whiteboard. This project can be very helpful to instructors because it can not only reduce their effort of repetitively erasing the board but also allow them to refer back to the content they wrote on the board that had been erased through simple clicks on the web application.

Some major challenges we expect include building the mechanical system for erasing, integrating RPi with web application, controlling the camera to take clear pictures, etc. Our group members will dedicate time and effort to overcome these challenges and we believe that we can successfully build this whiteboard-erasing tool that meet the need of instructors.

Glossary of Acronyms

- RPi – Raspberry Pi
- AWS - Amazon Web Services
- API - Application Programming Interface
- ONVIF - Open Network Video Interface Forum

References

Muthusamy, Suresh & Meenakumari, R & Raghavendran, P.s.Raghavendran & Gowrishankar, V &

Karthikeyan, P & Vadivel, Surendar & Jagan, C & Prashant, N. (2018). Design and Development of Automatic Whiteboard Cleaner for Effective Cleaning Mechanism using Arduino. IJIREEICE. 6. 30-33. 10.17148/IJIREEICE.2018.696.

Ghazali, Nurzaliza, Ooi Kuan Zhe, & Rosmani Ramli. Design And Development Of Automatic Whiteboard Eraser For Effective Cleaning Mechanism using Internet Of Things (IOT). Politeknik & Kolej Komuniti Journal of Life Long Learning [Online], 5.1 (2021): 43-52. Web. 15 Oct. 2023

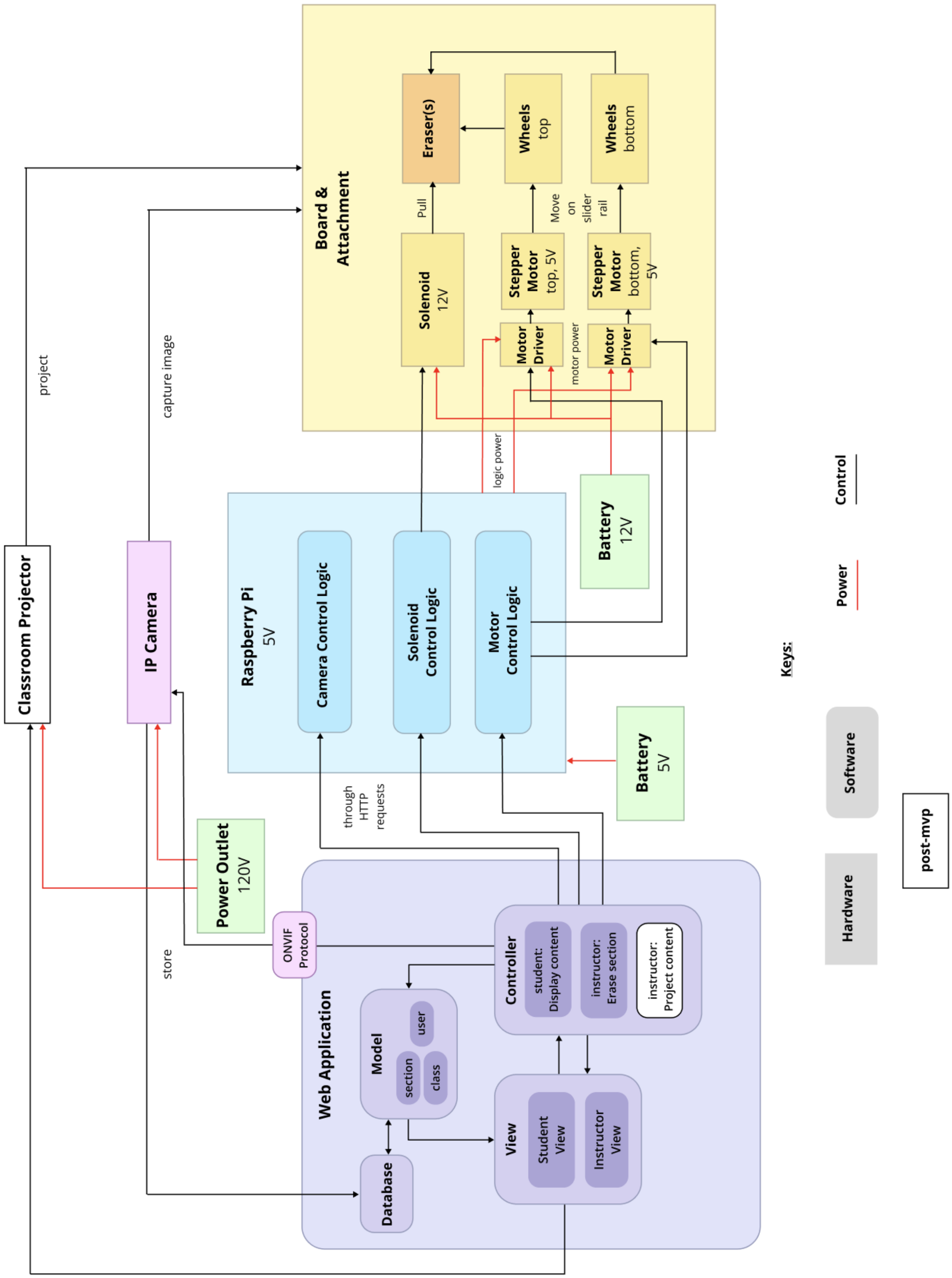


Figure 6: A full-page version of the same system block diagram as depicted earlier.

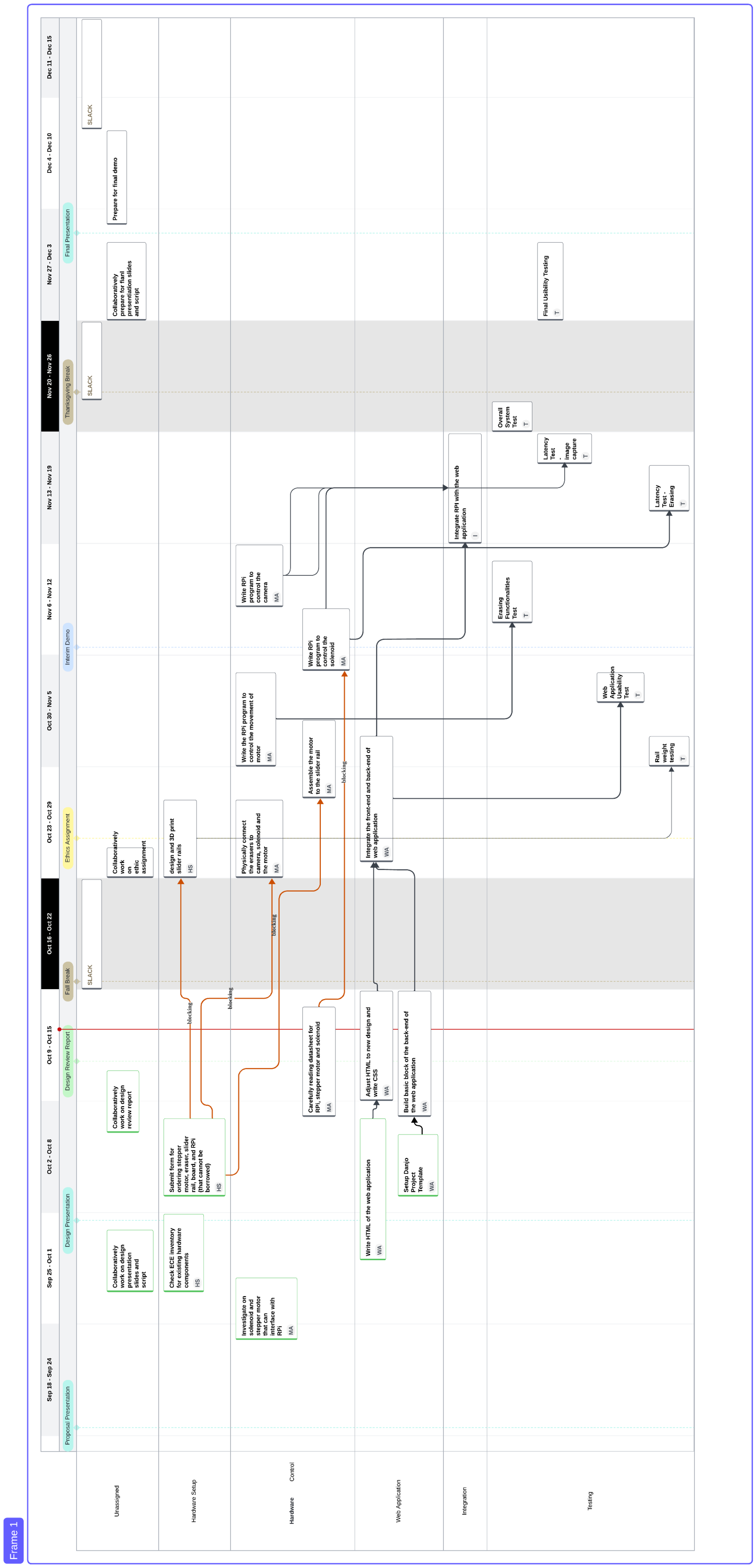


Figure 7: Gantt Chart