York University - Lassonde School of Engineering

ENG 4000: Capstone Project

Team 8: Automated 3D-Printing Management System

Final Report

Team Members:

Luca Filippelli

Jean Granato

Luqmaan Irshad

Paras Kumar

Tharuveen Raveendran

Ethan Tran

April 21, 2025

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Like with any large-scale system, structure increases productivity. Additive manufacturing is no different. Manufacturing operations partially achieve an optimal level of organization by ensuring their inventory is properly stored and accounted for. Furthermore, as additive manufacturing materials degrade when stored improperly, an effective management system must ensure the inventory's storage environment is capable of minimizing degradation. As no commercially available product has harmonized inventory management with smart storage, additive manufacturing companies currently face the task of keeping track of their inventory while ensuring safe storage. In the case of emerging additive manufacturing companies, the structure and cost-saving benefits of a harmonized inventory management and storage system can provide the logistical safety required to facilitate further expansion. Pantheon 3DP, a student-run additive manufacturing company and the sponsor of this project, faces a similar set of challenges.

The purpose of the Automated 3D-Printing Management System is to produce a material management system which will enable additive manufacturing companies like Pantheon 3DP to effectively store and track their printing material inventory. The team's solution divides the project into two key components. The first component consists of a web application. This software solution has been created to ensure that Pantheon 3DP can visualize its current stock and receive notifications when its inventory is low. The application design ensures that the company can scale the solution with their growth by enabling Pantheon's technologists to create, add, delete and read an inventory item. This app leverages conventional software engineering practices such as the implementation of layered architecture and cloud deployment to create a reliable, secure and interactive solution. The second component consists of an inventory storage solution. An enclosed shelving unit has been augmented with microcontrollers, climate sensors and insulating material to create an ideal environment for the storage of common 3D printing materials. This storage solution communicates with the web application through a common Internet of Things protocol to provide Pantheon 3DP with warnings in the event the storage unit's current temperature or humidity risks material degradation. Further automation has been added through a load cell, which, when combined with QR identifiers, enables the company's technologists to efficiently update a material's stock.

The project was governed by the Agile Methodology. As such, an initial feature backlog and release pathway were established through consultation with the project's stakeholders to set the project's expectations over time. This schedule has largely been maintained except for a pivot away from integration with the company's currently incomplete order application. As the integration was meant to efficiently update material inventory, a hardware-based automation solution has been implemented. Reviewing the project's progress with the sponsor has been an integral part of maintaining transparency and collecting feedback. The project sponsor has fully reviewed the team's progress at the Alpha, Beta and final release stages, finding the progress to be exceeding their expectations.

The product's high scalability and robustness provide a strong foundation for future expansion. For instance, the use of cloud technologies for the web application allows for the seamless integration of further automation and monitoring features without the need for additional computing resources. Additionally, future iterations of this project may focus on developing a local hardware broker to bypass unencrypted third parties. Overall, the project provides the necessary harmonization to safely store and track additive manufacturing materials while empowering future development.

2. Requirements

2.1 Scope

The necessary scope of the solution includes:

- The development of a climate-controlled shelving unit
- The development of a web-based application to conduct material management, inventory visualization and notification facilitation
- The development of a communication system to facilitate the transfer of information between the shelving unit and the web application

Items which are out of scope include:

- The integration or optimization of Pantheon 3DP's in-progress applications

2.2 Compliance

Requirement compliance will be tracked through sprint reviews with the supervisor as well as biweekly meetings with the contact.

2.3 List of Requirements

As per the Agile Manifesto and best practice within Agile project management, User stories provided the focal point for the development of each requirement subsection [1]. A further criterion was outlined in our Vision Board. For more details, see Appendix A. The subsections were generated based on each key objective. In other words, requirements shall be divided into criteria surrounding storage, web application and communication between the two former systems. As with all Agile projects, the requirements are subject to some change given the results of the preliminary sprints and user feedback. The tables below outline the requirements by category: shelving requirements (Table 1), web application requirements (Table 2) and communication requirements (Table 3). Since the previous report, no further requirements have been added. Every requirement has been fulfilled and validated.

Table 1: Shelving Unit Requirements

Requirement Identifier	**Requirements** *(all completed by the end of the shelf design)*	**Source**	**Priority**	**Validation approaches and timing**	**Feature Validated?**
REQ 1 - Performance	The shelf material must be durable, resistant to wear and insulate the inventory from the external environment. The shelf will break down quickly over time if it is not durable to external elements and must be acclimated to an indoor workshop environment.	Pantheon 3DP Founder/CEO	Must have	Shelf Material: Rust/Corrosion resistant, shelf must be able to hold **80 kg** of load without deformation and be scratch/wear resistant (**500 Brinell** level of hardness) To be done and validated at the beginning of the winter semester.	**Yes,** the shelf is made of painted aluminium that is scratch-resistant and rust/corrosion proof. Validated.
REQ 2 - Functional	The shelf should be able to hold 40 material spools with **10-spool** tolerance.	Pantheon 3DP Founde	Should have	This requirement will be verified by physically testing the storage, as well as ensuring	**Yes,** the shelving unit passes the test of holding 50 spools.

	This is a requirement from our project contact, as they are using this shelf as a container for all their filament spools.	r/CEO		the volume of the container is **able** to hold **40** spools. To be done and validated at the beginning of the winter semester.	
REQ 3 -Functional/Performance	The shelf should always be accessible for users to remove and place **40** spools and **10** resin containers. Since all the material isn't used all at once, it is important for the spools to be placed and removed regularly from the shelf. This should also be organized in a way where the highest weighted material should go to an easily accessible shelf.	Panther on 3DP Founder/CEO	Should have	This will be verified through a test group, instructing them to remove spools and place them. We will collect user feedback. This will ensure that we have the right dimensions for each spool and container to store them without trouble. To be done and validated in the middle of the winter semester.	**Yes,** the shelf can meet the requirements of 10 resin containers and 40 spools.
REQ 4 - Functional/ Programmatic	The shelf should contain the necessary electrical components to track the unit's internal climate and communicate with the web application. This is important so that the web application can communicate with the shelf. The web application requires temperature and humidity data so that the company is aware of the storage environments to keep its inventory safe.	Panther on 3DP Founder/CEO	Must have	This will be tested through the web application once the wiring and code are completed. To be done and validated in the middle of the winter semester.	**Yes,** the unit contains an Arduino MKR WiFi 1010, connected to a DHT-11 Temperature and humidity sensor.
REQ 5 - Functional	The shelf should regulate humidity specific to material needs. Specifically for the spooled materials, the material must be stored at a regulated humidity of **20%** RH to prevent it from becoming brittle and wasted.	Panther on 3DP Founder/CEO	Must have	This will be tested by using a humidity sensor and hook it up to a micro-processor and see the logged readings of it after **6 hours**. To be done and validated at the middle to the end of the winter semester.	**Yes,** the shelf is completely sealed and contains 4 desiccant carriers to regulate the internal climate.
REQ 6 - Functional	It should block nearly 100% of UV light from entering the storage system (for resin) Resin materials should not come in contact with any external UV light as it will start to cure and harden the material and become waste.	Panther on 3DP Founder/CEO	Must have	This will be tested by using a LDR (light dependent resistor) and logging the data with a microcontroller. The data will validate whether or not the light insulation is working correctly. To be done and validated at the middle to the end of the winter semester.	**Yes,** the shelf has been completely sealed from light.

Table 2: Web Application Requirements

Requirement Identifier	Requirement text	Source	Priority	Validation approaches and timing	Feature Validated?
REQ 7 - System	The application must contain a database to act as a material "source of truth". In other words, the database must, in one instant, represent **the current state of all** materials, users and shelving units.	Pantheon 3DP Founder/ CEO	Must Have	Unit testing will be used to ensure the database elegantly handles errors, stores inputs as requested and is CRUD-compliant. This has been verified in November	**Yes,** this requirement has been validated through unit, integration and system testing. It continues to be validated after every change through a Continuous Integration testing pipeline.
REQ 8 - User	The web application should allow system administrators to create, read, update and delete material instances. This CRUD (create, read, update, delete) capability will enable the business to represent the intake of new printing materials on its management system.	Pantheon 3DP Founder/ CEO	Must Have	This will be verified using integration testing and feature testing. Integration testing will be used to ensure that the Hypertext Transfer Protocol (HTTP) requests from the client to the server return the expected status codes given an array of valid and invalid inputs. Feature testing will be used to ensure that the interface has the necessary components for users to complete these HTTP requests. This will be verified in January	**Yes,** this requirement has been validated through integration and system testing. It continues to be validated after every change through a Continuous Integration testing pipeline.
REQ 9 - System	The web application must provide a user hierarchy such that a **super administrator** can manage the permissions of administrators.	Pantheon 3DP Founder/ CEO	Should Have	Feature testing will be used to ensure that the superadministrator has the correct interface to update user permissions. Integration testing will be used to ensure that the interface triggers HTTP requests with the expected response codes for a diverse set of inputs.	**Yes,** this requirement has been validated through unit, integration and system testing. It continues to be validated after every change through a Continuous Integration testing pipeline.
REQ 10 - System	The application must provide visual interfaces to: - Monitor the company's inventory in real-time - Receive system notifications in real-time - Perform CRUD actions on existing inventory.	Pantheon 3DP Founder/ CEO	Must Have	This requirement will be verified using feature testing to assess the expected nature of the interfaces.	**Yes,** the interface has been validated through system testing. It continues to be validated after every change through a Continuous Integration testing pipeline.
REQ 11 - User	Users should be able to query for a material or set of materials by: - Material Type	Pantheon 3DP Founder/ CEO	Must Have	Unit testing will be used to validate the expected response of the persistence layer methods responsible for querying.	**Yes,** this requirement has been validated through unit, integration and system testing. It continues to be

	- Colour - Mass			Integration testing will validate the expected HTTP response code of the business layer methods, which are responsible for processing data between the persistence layer and the interface. Feature testing will be used to ensure that the user has a visual representation of their query.	validated after every change through a Continuous Integration testing pipeline.
REQ 12 - User	The system must provide the users with real-time notifications when a material's mass falls below fifty (50) grams.	Pantheon 3DP Founder/ CEO	Must Have	Integration testing will validate the expected response of a material mass listener	**Yes,** the requirement has been manually validated using a console and print statements. Integration tests will be formulated once the material warning system is complete.
REQ 13 - System	All notifications must be provided as a pop-up message within the application and through an email to each user.	Supervisor Meetings	Should Have	Feature testing will be used to ensure that a popup with the correct information is visible on the interface during a notification. Integration testing will be used to ensure a mailer event is properly triggered upon a notification	**Yes,** the email services have been validated through a mixture of manual testing (to ensure the emails are actually sent) as well as unit and integration testing to ensure the email services execute when expected with success (without actually sending an email to ensure we do not waste resources). The pop-up notifications have been verified through system testing.
REQ 14 - User	The web app must be able to provide the users with notifications once the humidity or temperature of a shelf falls outside of a safe threshold for	Pantheon 3DP Founder/ CEO	Must Have	Integration testing will validate the expected response of a shelf listener	**Yes,** this feature has been validated through system testing.
REQ 15 - User	The application should provide users with the capability to export all current shelf and material data to a Column Separated Values (CSV) file.	Pantheon 3DP Founder/ CEO	Could Have	Integration testing will validate the expected response of a CSV export service.	**Yes,** feature testing has been used to verify the requirement.

Table 3: Communication Requirements

Requirement Identifier	Requirement text	Source	Priority	Validation approaches and timing	Feature Validated?
REQ 16 - System	The microcontroller must communicate its humidity and temperature data with the web application in **real-time**.	Pantheon 3DP Founder/ CEO	Must Have	This will be tested by comparing the received values on the web application with the values of a physical thermometer and humidity sensor over **6 hours**	**Yes.** The feature has been validated both by reading the values from a physical temperature and humidity sensor, and values over the 6 hours have been logged by our application backend in the terminal output to show

					the values are within the expected range and our microcontroller is communicating with our application.
REQ 17 - System	The microcontroller must be able to measure the mass of an individual spool using a singular load cell.	Pantheon 3DP Founder / CEO + Professor Gerd Grau	Should Have	This will be validated by comparing the received values with the scale Pantheon 3DP uses. We will measure the weight of several known masses, each with different weights, to see whether the values are identical.	**Yes.** We have calibrated the scale readings to match the readings on the scale Pantheon 3DP uses, testing it with several objects of different masses and ensuring they are identical.
REQ 18 - System	The microcontroller must be able to send the current mass being measured by the load cell to the web application upon detecting a change in mass in **real-time**.	Pantheon 3DP Founder / CEO + Professor Gerd Grau	Should Have	This will be tested through integration tests on the software side and visual confirmation to assert that the communication and mutation works as expected	**Yes.** Through the use of integration tests and visually checking that the backend receives data in real-time from the scale when placing masses on the scale.

3. Baseline Design

3.1 Design Problem to be Solved

The essence of the problem lies in creating a harmonized system by which inventory can be visualized, tracked, monitored in real time and stored. The problem can effectively be captured by the following statement:

> "How might we **automate** the 3D printing *material management system* of Pantheon 3DP, and **integrate** it with the company's existing *order placement platform*, to **optimize** the company's *material waste* while **reducing the cost** surrounding material procurement?".

3.2 Decomposition of Product Features

With creating a complex system, it is important to decompose the requirements into smaller sections which are more suitable for the team. Using the Agile Methodology, we were able to form product increments as sprints which allowed us to track which product features would be completed. To decompose and optimize sufficiently, we had to split the features as software and mechanical to see how many smaller increments we needed.

For the web application, we broke down the systems into frontend and backend. With each sprint, we focused on one feature at a time. In our first sprint, we focused on creating a database that would mimic Pantheon's materials. We expanded upon this database by adding features to visualize each of Pantheon's materials. In the second sprint, we focused on developing a basic frontend mutation that would enable Pantheon's technologists to update existing inventory items. We also began the work of creating a basic warning system for low-material stock. The following sprints continued the work of enabling CRUD functionality for Pantheon's materials while adding further quality of life features such as email notifications. Feature planning was done during the team's regular sprint planning sessions and selected on a priority basis. This priority is outlined within our Now-Next-Later chart. This methodology is discussed in detail within the Project Management section.

For the shelving unit, the systems were broken down based on priority and feasibility. As the main core materials had to be bought, we had to structure our sprints based on what could be done before our Bill of Materials (BOM) was approved and materials were bought. In the first sprint, we worked on creating some designs, specifically a 3D model, of what the shelf should look like, which included dimensions and sensor placement. The second sprint focused on communicating with Pantheon and figuring out what shelf was needed and what kind of sensors would be feasible. This sprint also focused on creating a BOM and sending that in so it could be approved earlier. The final sprints split the shelving work into insulation-related tasks and circuitry-related tasks. The insulatory tasks centred upon establishing the ideal conditions for material storage, whereas the circuitry tasks focused on providing communications between the shelving unit and the web application.

3.3 Engineering Concepts and Theories Applied

3.3.1 Layered Architecture

The Layered Architecture pattern provides several layers of abstraction for software component interaction. Namely, these sections include the Database, Persistence, Business and Presentation layers. The Database layer acts as a vehicle for data definition and storage within the application. Above it, the Persistence layer provides an interface for database interaction without a strong dependency on the specifics of the database. The Business layer serves to implement business logic within the application while calling the Persistence layer components to receive data. Lastly, the Presentation layer is directly concerned with generating an interface for user interaction [2].

The strength of this pattern lies in its separation of concerns. Generally speaking, data must pass through each layer in succession such that no layer is skipped (Figure 1) [2]. This strict flow of data effectively abstracts a singular layer from its non-neighbours by removing its direct dependency on every other application component except for its neighbours. For example, despite indirectly receiving information from the database, the Presentation layer is not concerned with the database's schema as the Persistence and Business layers abstract this information.

Through its strict adherence to a structured flow of information, the Layered Architecture pattern enables an application to be easily verified and scaled. The pattern's ease of verification is by its separation of responsibilities between the layers. This division enables a developer to efficiently debug any errors through independently testing each layer. Layer-based independence also allows this pattern to be efficiently scalable.

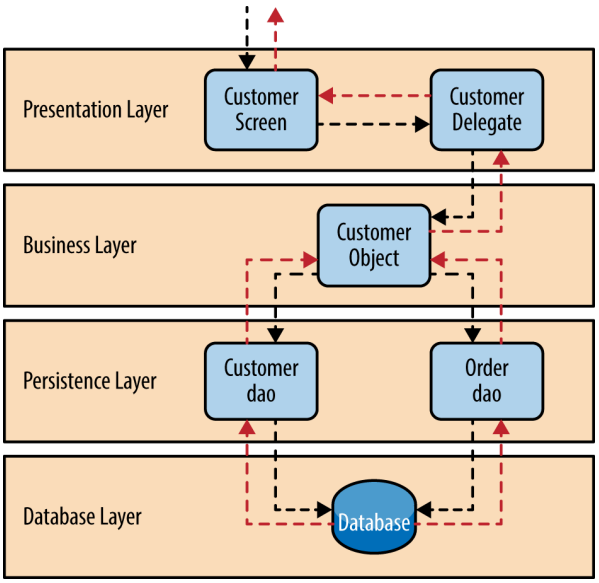


Figure 1: A generalized layered architecture diagram [2].

3.3.2 Repository Design Pattern

The Repository is a structural design pattern which facilitates the abstraction of a data model from its query tools [3]. This pattern typically operates within the persistence layer. By separating the query tools from an entity, one can effectively remove the dependence of the entity from business layer methods which call the repository. Therefore, the methods within the repository act as the primary link between the database layer and the business layer.

3.3.3 Representational State Transfer (REST) Protocol

The REST protocol is an architectural style used in the development of web applications. When implemented, REST ensures that an application is lightweight and scalable. This architectural style is used in the implementation of REST Application Program Interfaces (APIs) [4]. Pantheon Material Management makes use of the REST protocol to facilitate client-server communication by providing a series of server-side API endpoints for the client to call. This communication is enabled through the use of the Hypertext Transfer Protocol (HTTP) (Figure 2).

The key commands of HTTP include POST for creating new resources, GET for fetching them, PUT for updating existing resources, and DELETE for removing them. These commands are used by the client to standardize requests to the server. Within the server, Create, Read, Update, Delete (CRUD) operations are utilized to facilitate the necessary requests. The use of REST with the HTTP protocol ensures that the application maintains a standardized method of communication between the client and the server.

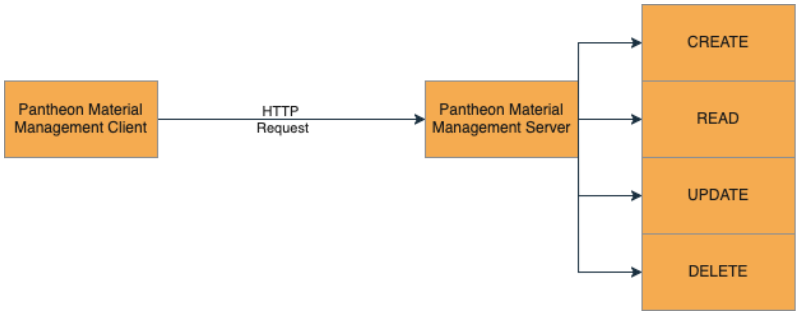


Figure 2: A client-server HTTP model.

3.3.4 Component-Based Architecture

Component-Based Architecture is a software paradigm which emphasizes the use of reusable and self-contained functional units which are capable of managing their state [5]. This paradigm is primarily used within the React frontend framework for developing user interfaces. By having a self-contained state, each component can effectively be used as a template to generate component instances within a user interface. This property of Component-Based Architecture facilitates the creation of a component instance hierarchy, which, in turn, increases the scalability of the system's user interface.

3.3.5 Software Testing

Three significant forms of testing exist to verify the functionality of various components within a software system. Unit testing is the most basic verification form. It focuses on verifying individual methods within classes to confirm their expected behaviour [6]. This type of testing ensures that developers can determine whether a software component works in isolation.

Integration testing increases the complexity of unit testing by verifying the interaction between multiple software components [6]. This form of testing ensures that developers can determine if multiple components work together in an expected way. A common form of integration testing in web development consists of simulating HTTP requests to an API endpoint to determine whether the client receives the expected response code. This test would combine the API controller with the necessary backend methods to facilitate a response.

Functional testing is the final major software testing form. This type of testing exists to verify the business requirements of a system. As such, functional testing primarily treats the system as a black box [7]. In the context of web development, functional testing can be achieved through the use of web automation tools such as Selenium. These tools simulate user interaction to ensure that specific user actions generate the expected outputs from the system.

Using each testing form within a software application can ensure that the entire system remains verifiable. As a system increases in complexity, this verifiability becomes important to determine that previously built components work as expected.

3.3.6 Internet of Things Integration

The Internet of Things (IoT) outlines a network of embedded devices which communicate with software to collect data [8]. This data can then be processed as a facet of an application's business logic to better serve the user. Most importantly, the IoT enables users to make more effective decisions based on real-time information. For instance, a user may decide to move their plants inside if a real-time temperature sensor alerts them to a significant drop in temperature.

The facilitation of real-time communication within an IoT system requires a robust and lightweight protocol such as MQTT. This protocol is based on the publish-subscribe methodology. In essence, a publisher will generate a message that a subscriber can receive through the use of a message broker. The use of a broker ensures that communication can be maintained if the subscriber and publisher are not properly functioning at the same time [9]. Furthermore, the low-overhead nature of MQTT enables the protocol to be lightweight. This property is ideal in an IoT environment given the presence of microcontrollers. As small microcontrollers are low-power, a lower overhead maximizes their transmission efficiency.

For scalability, we have designed the message similar to a data packet, where it contains a header, and the data. In our case, the header is the shelf ID. This allows us to separate requests from different shelves being published to the same broker and update the specific shelves accordingly.

The publish-subscribe nature of MQTT coupled with its lightweight communication capability allows for the protocol to synergize within an Event Driven Architecture.

3.3.7 Event-Driven Architecture

Event Driven Architecture (EDA) is a design model which facilitates the swift detection of and reaction to real-time events. An event in the EDA jargon describes an individual change of state within the application which is relevant to the given business logic [10]. Generally speaking, EDA enables software to be executed on the basis of events through a publish-subscribe methodology.

EDA is particularly useful for the facilitation of real-time notifications. For example, if a sensor detects a significant change in a shelving unit's temperature, EDA can facilitate a swift response through a user notification. To facilitate this architecture decision, we have elected to use the standard messaging protocol MQTT, for its lightweight data publishing via Topic Exchanges.

Passing information via MQTT is relatively simple. All that is required is to set up two clients. Typically, both can send and receive data, but for our purpose, we have one as the sender and the other as the receiver. The sender will create "topics" on the broker, which it uses to publish data to. The receiver can then subscribe to these topics to read the data being published by the sender. This transmission of data can be referred to as a "Topic Exchange" [11]. The Topic Exchange effectively limits communication to relevant components. In essence, the flow of data is as follows: collect sensor data→ sender publishes data to broker → receiver reads data from broker → receiver processes data.

To implement communication via MQTT in our application, we have opted to use a mixture of ArduinoMqttClient and paho-mqtt. ArduinoMqttClient is the supported MQTT library for Arduino devices and is relatively simple to use. It allows the Arduino (assuming it is connected to a network) to connect to an MQTT broker (like in our case, Mosquitto). Once connected, it can initialize topics and send data to those topics.

The paho-mqtt library is another client library, based on the MQTT protocol. It is best suited for lightweight messaging, such as signalling sensor data from our shelf to our application. Unlike a more robust message broker such as RabbitMQ, paho-mqtt simply allows one to connect to and interact with a broker. This workflow aligns better with the scope of this feature, as a lightweight library has all the capabilities needed to satisfy the requirements of this feature. Therefore, paho-mqtt enables us to remove the unnecessary bulk of the robust message handling provided by brokers like RabbitMQ, going for simplicity and efficiency.

3.3.8 Password Hashing

User accounts are typically password-protected to prevent unwanted access. For a password to be usable, it must be stored. The storage of passwords differs from the storage of typical data, as a password breach could risk the integrity of each user account.

The most optimal method of storing passwords requires hashing. A hash function is an algorithm which maps data to a fixed bit string. This function is typically difficult to reverse, thus decreasing the risk of an intruder reversing the hash function to obtain user passwords. For instance, if an intruder were to gain access to the password database and every password is stored as a hash, the risk of the passwords being compromised would be very low. To ensure that two passwords do not result in the same hash result, a salt can be used. Salting a hash refers to the process by which random data is added to each input, thus ensuring that every hash is unique [12]. Therefore, by storing passwords as a salted hash, developers can ensure that user accounts have a significantly lower chance of being breached.

3.3.9 Material Degradation

Without proper care, common additive manufacturing materials can degrade. Factors such as ultraviolet radiation exposure, temperature and humidity can influence a material's degradation process [13]. For instance, a high-humidity environment can cause common 3D printing materials to degrade as the material will begin to absorb the water within the air (Figure 3) [14]. In the context of additive manufacturing, material degradation can reduce the supply of viable raw material, thus negatively affecting the manufacturing process.

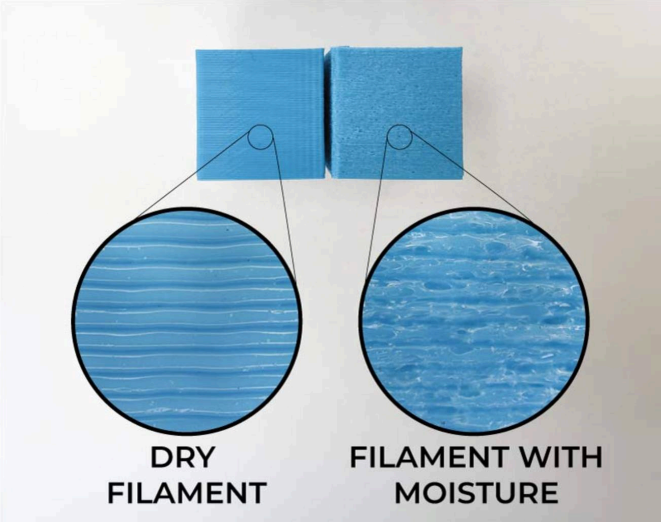


Figure 3: An example of material degradation due to high humidity [14].

While 3D printing materials radically differ, a generally acceptable temperature and humidity range can be developed by taking the tightest range of each material. For example, of all common Fused Deposition Modelling (FDM) materials, Acrylonitrile Butadiene Styrene (ABS) has the tightest ideal temperature maximum of 30 ℃, and Nylon has the tightest ideal humidity range of less than 50% [15]. By using the tightest storage conditions for each material, one can ensure that every material remains in an ideal state without a high risk of degradation.

As Pantheon 3DP also creates resin-based products, storage solutions must provide an ideal environment for resin. The stereolithography process uses UV curing to solidify the resin into a desired shape. Therefore, it is crucial to store the liquid resin in an opaque container [16]. Furthermore, liquid resin remains stable in an environment between 10 ℃ to 30 ℃ [17], [18]. Therefore, any ideal storage method must keep its temperature within the acceptable threshold for resin. By respecting the temperature and UV requirements of resin storage, one can increase the shelf life of the material.

To reduce the risk of material degradation, a storage solution should be able to maintain the ideal humidity and temperature for Pantheon 3DP's FDM and Stereolithography materials. Therefore, by taking the tightest bound of each material's temperature and humidity tolerance, one can determine that the ideal storage temperature lies below 30 ℃ and the ideal storage humidity lies between 0% to 49%.

3.4 As-Built Design

The final design consists of a web application, an insulated shelving unit and communications infrastructure for the components to respectively receive and send environmental information. The use of cloud hosting technologies for the project's software facilitates real-time communication with the user while providing the organization with a scalable service for companion projects such as an order application. Lastly, IoT integration within the shelf automates the inventory tracking process through the use of Quick Response (QR) code identifiers. This section will analyze the function of each component (and subcomponent) within the system.

3.4.1 Layered Architecture

Pantheon Material Management follows the traditional layered architecture model as defined in this report with a slight augmentation. The application includes a fifth controller layer to centralize all API endpoints, see Appendix AF2. This centralization ensures that one component is responsible for responding to the client.

Another notable, albeit slight, change from the textbook architectural diagram is the inclusion of a methodical approach to database access. In the traditional layered architecture, Data Access Objects would be used to facilitate access to the database. This facilitation would require the use of raw SQL queries. The use of SQL queries is suboptimal, as custom security methods would have to be created to avoid possible SQL injection attacks. Furthermore, raw SQL queries, as opposed to the use of an Object Relational Mapping (ORM) tool limit the scalability of the software system through their verbosity. Therefore, an ORM was added to the persistence layer to act as an efficient query tool. Furthermore, sophisticated ORMs such as SQLAlchemy have built-in security mechanisms to minimize the risk of SQL injection.

To ensure that the service layer is not directly dependent on one specific ORM tool, a repository pattern was implemented within the persistence layer. This pattern is responsible for decoupling database queries from the database representation itself. In the context of Patheon Material Management, the repository pattern also acts as an ORM query wrapper thus ensuring that the service layer is not directly dependent upon a raw ORM call.

3.4.2 Database Schema

Our database schema was developed to ensure entity relationships are properly modelled while providing the capability for efficient querying. This was achieved by using the relational modelling capability of SQL.

The database contains six primary entities responsible for maintaining information for specific portions of the application (Figure 4). The entities and their function are listed below:

1. **Material**: This entity models each material based on colour, name and mass. The material type and shelf location are modelled through a foreign key relationship with the Material_Type and Shelf entities.
2. **Material_Type:** This entity provides the application with the ability to group materials by their type. For example, this entity allows for the grouping of PLA materials through a PLA Material_Type instance.
3. **User:** The User entity enables the database to keep track of each active user. The principle of password hashing is applied to the User entity to emphasize the necessity of the security method in storage.
4. **User_Type:** The User_Type table allows the application to create a user hierarchy based on user types. Similar to Material_Type, this entity provides the capacity to group users.
5. **Shelf:** The Shelf entity allows for each shelf to be modelled as an individual entity capable of tracking its real-time temperature and humidity.

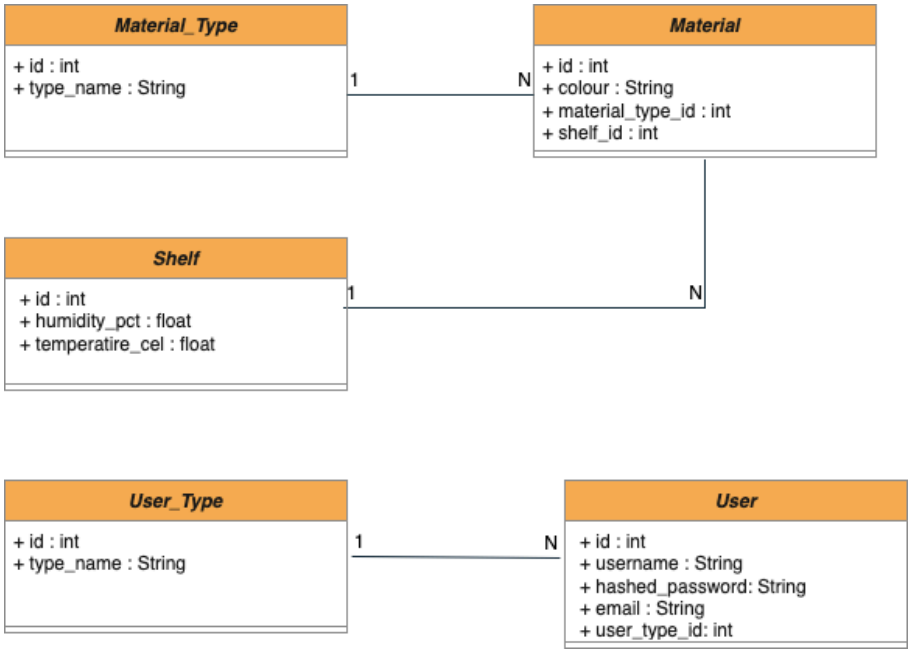


Figure 4: The Automated 3D-Printing Management System's database schema.

3.4.3 Fast-API Backend

The Fast-API backend exists to expose several API endpoints for clients to send requests to. In essence, the backend serves to receive inputs from the client, trigger the appropriate service methods and respond to the client's request with an HTTP code along with the necessary resources. These methods have been made available such that future developers can create requests to our server to retrieve data. A detailed API endpoint table has been made available in the Appendix AT3.

3.4.4 NextJS Frontend

A secure and user-friendly login page was implemented to authenticate users before accessing the application. It supports username and password authentication, allowing only authorized users to access the application. It includes features such as input validation, password hashing, which will be explained in a later section, and error handling to enhance security and usability. Additionally, the design aims to provide a smooth user experience by giving clear feedback for incorrect credentials and an option for password recovery (Figure 5).

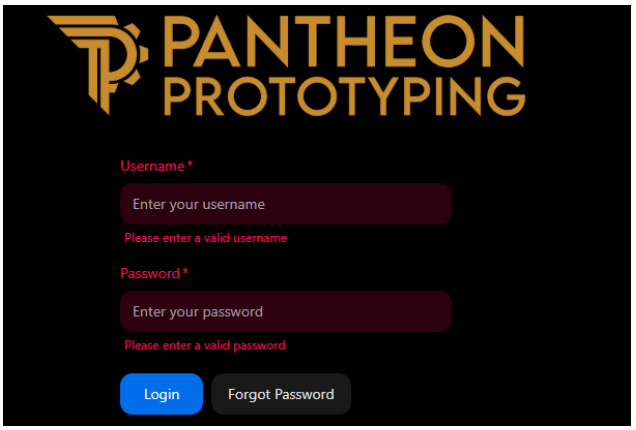


Figure 5: Void credentials login functionality example

The current state of the web application's frontend enables users to meaningfully perform CRUD operations on the materials, users and material types. The application makes use of tables to ensure that a user can view all existing entities in the system. The table allocates one row per entity with columns to represent the entity's attributes. The material table also provides a status to qualitatively indicate a material's level of inventory. This status indication is set to a green "In Stock" tag when a material has more than 50g of mass. Otherwise, the indicator is set to a yellow "Low Stock" tag. This attribute enables the user to be aware of low material quantities after the notification is issued. The table now includes a hyperlink to the supplier links associated with each material, making it easier to replenish inventory. On the far right-hand side, each table provides an edit button and a delete button to enable the user to make updates to a material and, if necessary, delete an entity. Again, the material table receives special treatment through a replenish/consume inventory button in each row to efficiently update a material's mass. A search bar is located at the top of the table to provide a search functionality if the inventory becomes large. Query parameters such as colour, status, shelf and even the material type, making it easier for Pantheon to locate certain materials and filter to their liking. The table includes functionality for exporting its data in Comma-Separated Values (.csv) format, enabling users to efficiently extract and utilize the information for external analysis or record-keeping. (Figure 6).

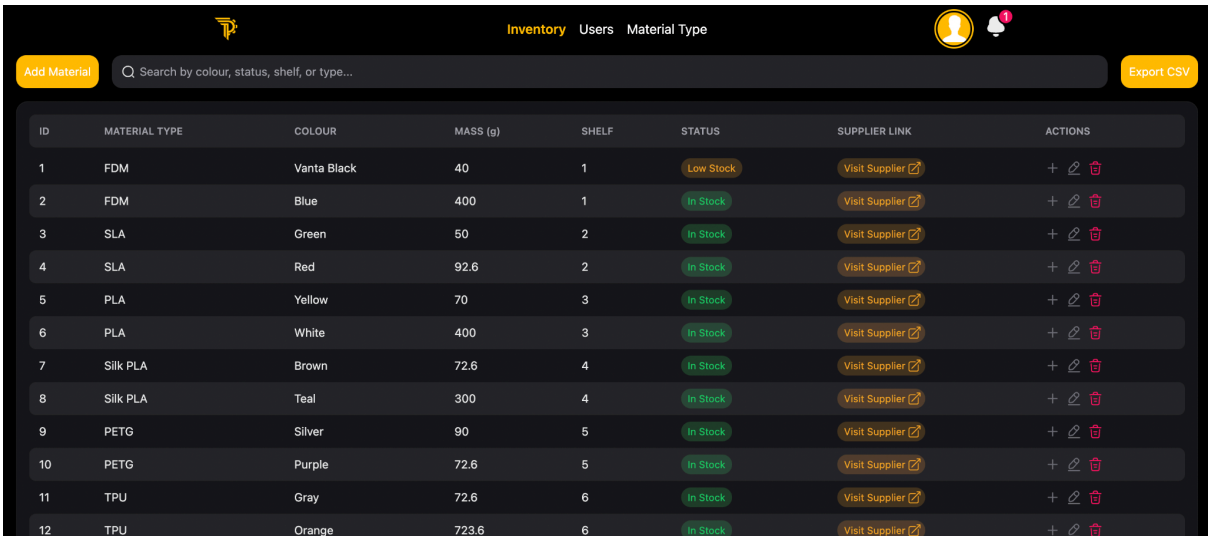


Figure 6: A material visualization example within the web application

Each table entry button acts as a method of activating its respective popup panel. For instance, the edit button will open a panel to edit the given entity (Figure 7). This window acts as the necessary user interaction to trigger the previously discussed edit and delete use cases. By including these buttons, the user has an efficient method of conducting basic entity manipulation through straightforward means.

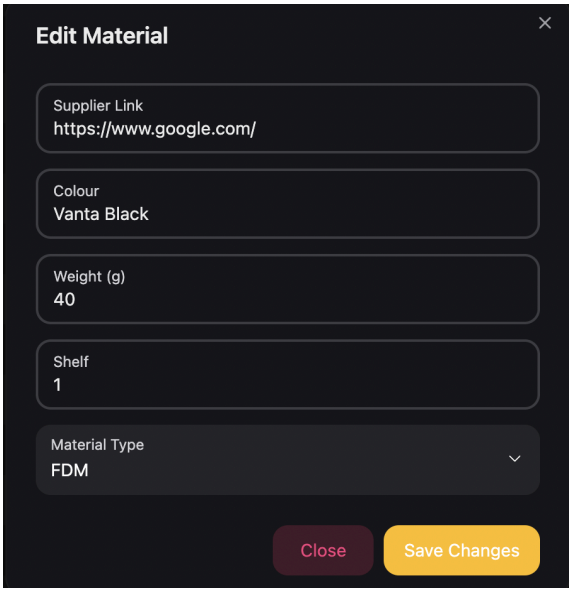


Figure 7: An update panel example within the web application

Pagination has been implemented to enhance data organization across various tables, including an inventory table, a user management table (restricted to super admins), and a material type table. This was essential in preventing clutter on the home screen while improving accessibility and usability. By leveraging the Next.js App Router with its navigation system, a seamless and structured pagination framework was established, making it easier for future developers to extend and maintain. (Figure 8).

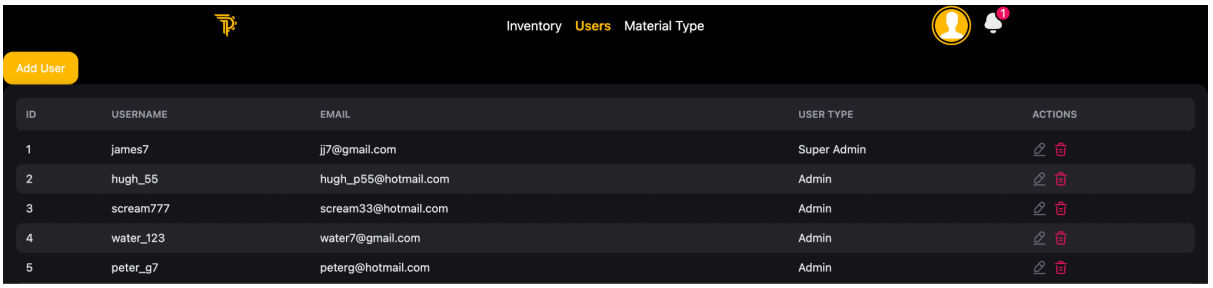


Figure 8: A user table example showcasing pagination

A dedicated user section was created, allowing users to view and edit their personal information through a separate user page. Users are able to update their email address, change their username, and even reset their password, ensuring they have full control over their account settings for a more personalized experience. (Figure 9).

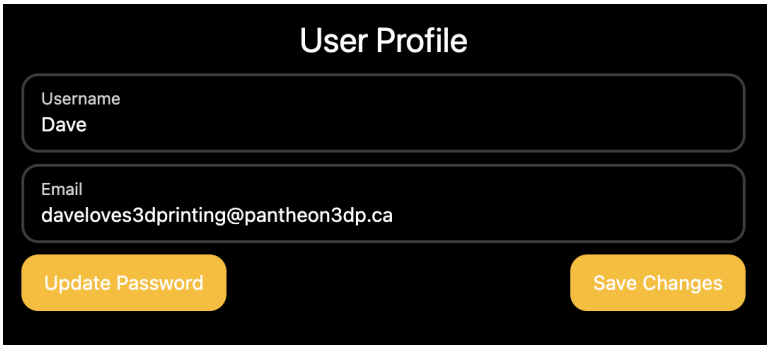


Figure 9: User profile page showcase

To facilitate the manipulation of material mass by Pantheon users, each material is assigned a unique QR code that links dynamically to a mobile-friendly interface. The QR code utilizes a URL containing the material's unique identifier (ID) from the database, enabling real-time access to its corresponding data. This dynamic linking strategy allows a single templated interface to accommodate all materials, significantly reducing space complexity and improving scalability. Through this interface, users can efficiently adjust the mass of the material, either increasing or decreasing it as required, after weighing it on the built Arduino scale (Figure 10).

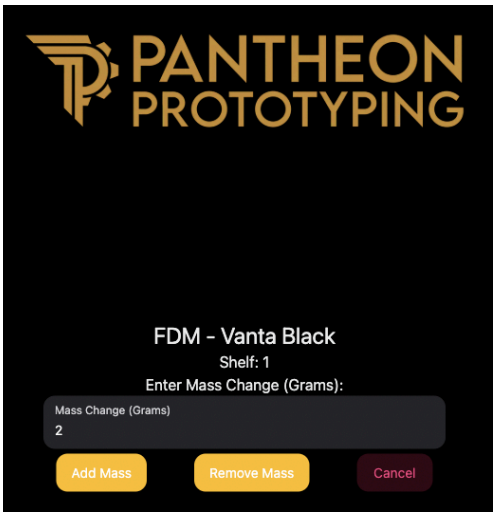


Figure 10: Mass manipulation page via QR code access

A real-time notification panel has been implemented to communicate critical alerts to users. This panel is connected to a live notification count and is designed to display warnings when monitored parameters breach predefined thresholds. Specifically, notifications are triggered when material quantities fall below their minimum acceptable levels or when environmental conditions such as shelf humidity or temperature deviate from the optimal thresholds (Figure 11).

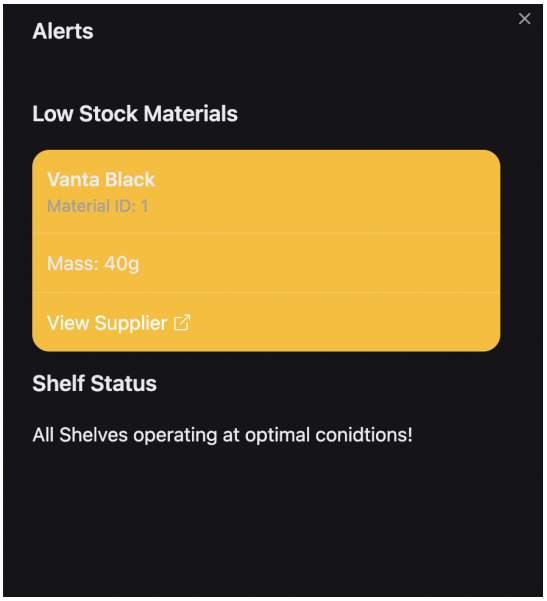


Figure 11: Notification panel for material and shelf alerts

3.4.5 User Hierarchy

In accordance with the project requirements, a two tiered user hierarchy was established to police data access. Firstly, a login page was created to provide limited access to registered users. User logins are completed by using a combination of a user's plaintext password and username. An email-triggered forgot password action is also included in order to send a registered user a temporary password in the event one is requested.

User access is further restricted through the delineation of Admins and Super Admins. Admins can perform CRUD actions on their user attributes, materials and material types. Super Admins have all of the features of Admins with the added benefit of having CRUD access for system users. It should be noted that access to user passwords has been restricted such that the only direct action that a user can take is to change their password or, in the case of superadmins, create a user with a default password. This access is enforced through the use of user session tokens and the privatization of API endpoints via IP blocking.

3.4.6 Use Cases

To provide a complete understanding of the software system, it is crucial to understand the MVP's existing use cases. In this context, use cases span the CRUD operation of entities (in this case users, materials and material types) in a table-like format. For this reason, the diagrams have been generalized to any instance of an entity.

3.4.6.1 Deleting An Instance

To delete an instance, an API endpoint will be made available to the client through a delete button on the table (Figure 12). For example, to delete a material, the user would press the delete button on the material's row. After providing the necessary user interaction, a delete action on the entity's controller will be triggered. This action calls a service which verifies that the user has the correct permissions to delete the entity. After verification, the material repository makes the correct query to the database.

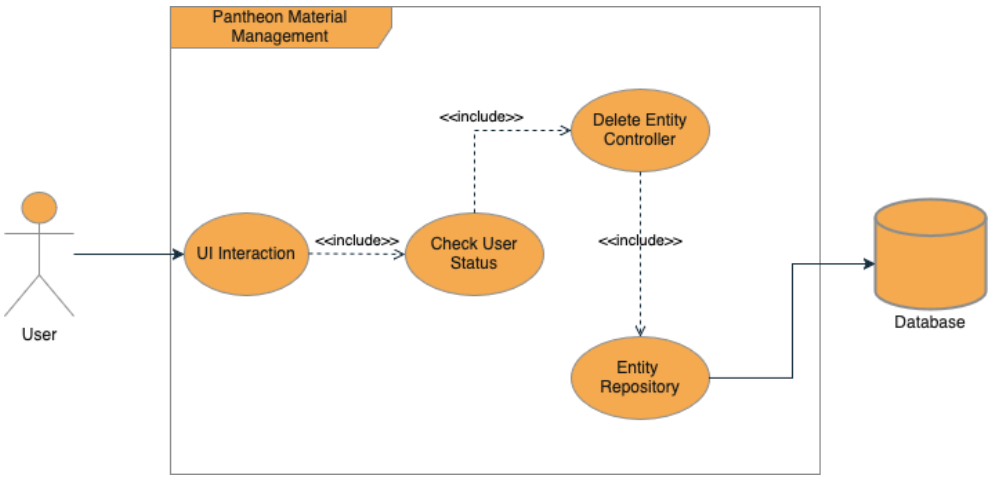


Figure 12: Use case diagram for an entity deletion command

3.4.6.2 Adding An Instance

To add an instance, an API endpoint will be made available to the client (Figure 13). After providing the necessary user interaction, an add action on the entity's controller will be triggered. This action calls a service which verifies that the user has the correct permissions to add the instance. After verification, the material repository makes the correct query to the database.

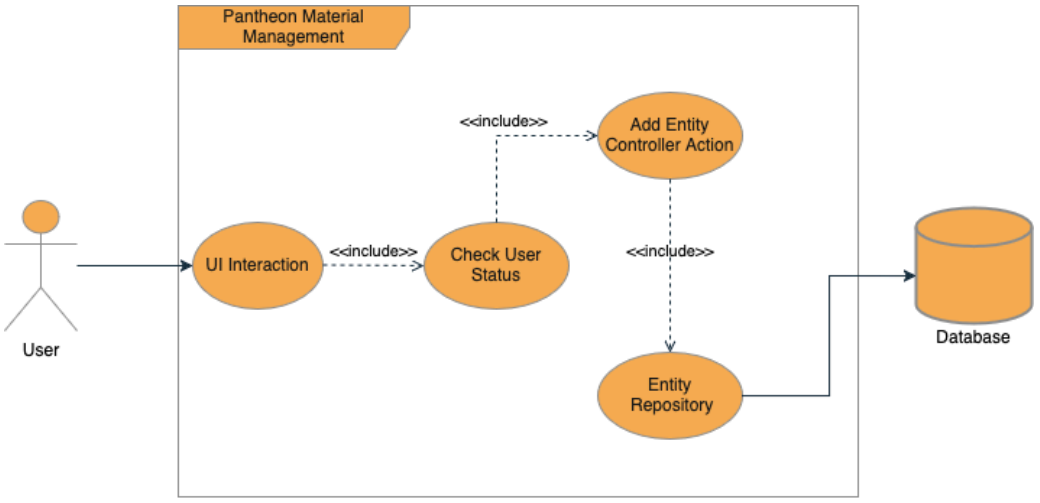


Figure 13: Use case diagram for an entity addition command

3.4.6.3 Updating An Instance

Instances can be updated through a similar series of events as deletion. An update button will be made available for each entity instance (Figure 14). This button will act as the necessary user interaction to provide new values for the instance and trigger a controller action. After triggering a controller action, a service will be called to determine whether the user can edit, providing an extra layer of access management. The entity's repository class will then be invoked to make the necessary database command.

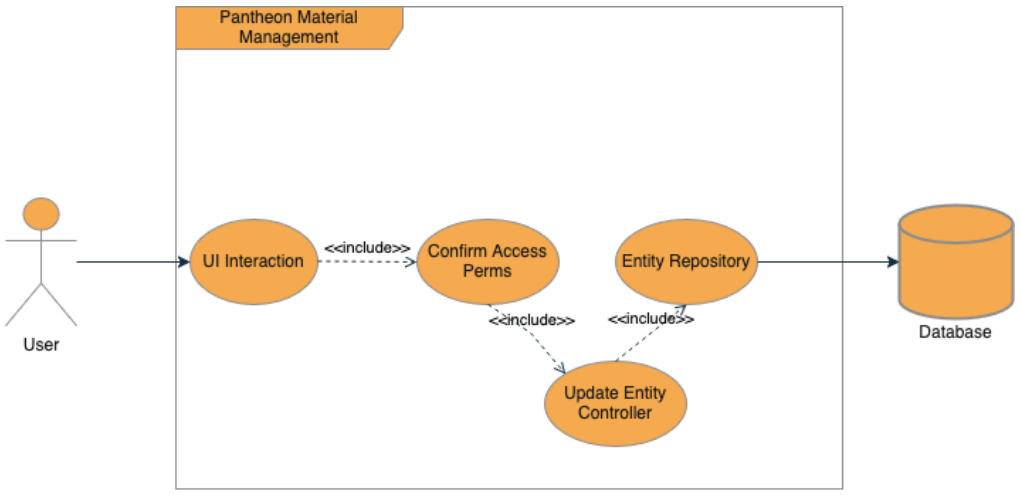


Figure 14: Use case diagram for an entity mutation command

3.4.7 Microcontroller and IoT Integration

By combining circuitry and software, we created an autonomous, multi-purpose device that has become the bridge between our physical and digital aspects of our design. The unit, being dubbed **MAI** (Monitor, Automate, Integrate), allows us to autonomously monitor the interior conditions of our shelf, integrates a scale that can be used to record the mass of materials and allows us to automate sending all of this data to our web application in real-time. MAI is fully online, connected to York University's AirYorkGUEST WiFi network, and makes use of IoT messaging protocols, which will be described later in this section.

As for the electronics, here is an overview of all the components we have used to create MAI (Figure 15 and Figure 16). Further details on each component will be listed below:

1. **Microcontroller:** Arduino MKR WiFi 1010.
2. **Scale:** HX711 24 BIT Precision AD Module + Tension-Based Strain Gauge Load Cell.
3. **Climate Sensor:** DHT11 Temperature and Humidity sensor.
4. **Scale Tare Button:** Tactile Push Button + Resistor (10kΩ).
5. **Other Components:** Breadboards (16.5cm x 5.5cm x 1 cm (main), 5.4cm x 8.7cm x 1cm (display)), and various wires (M to M, M to F and F to F depending on connection types).

We have selected the Arduino MKR WiFi 1010 as our microcontroller. This version of the Arduino allows for network connectivity via the WiFi umbrella of protocols. Its small profile allows us to directly place it on the breadboard, removing the requirement of designing a 3D-printed housing to secure the SoC and reducing the housings' overall footprint.

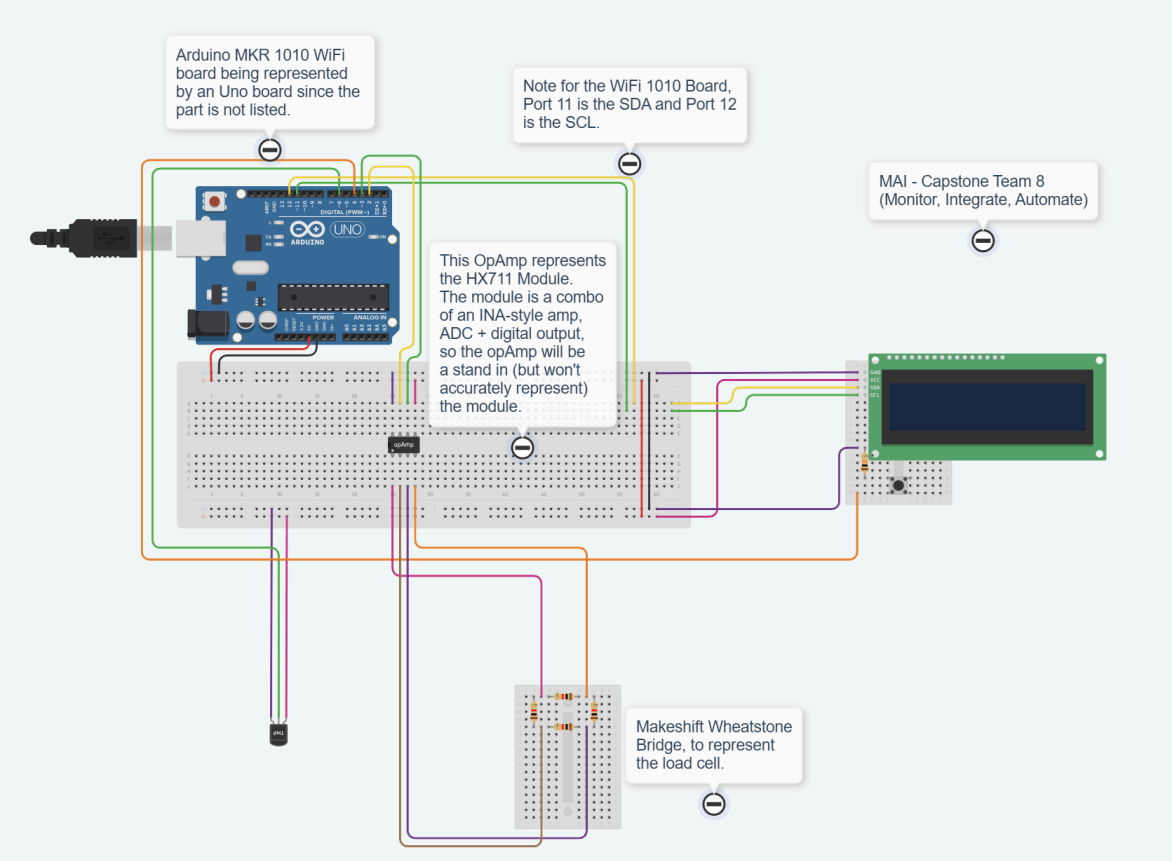


Figure 15: Circuit diagram of MAI.

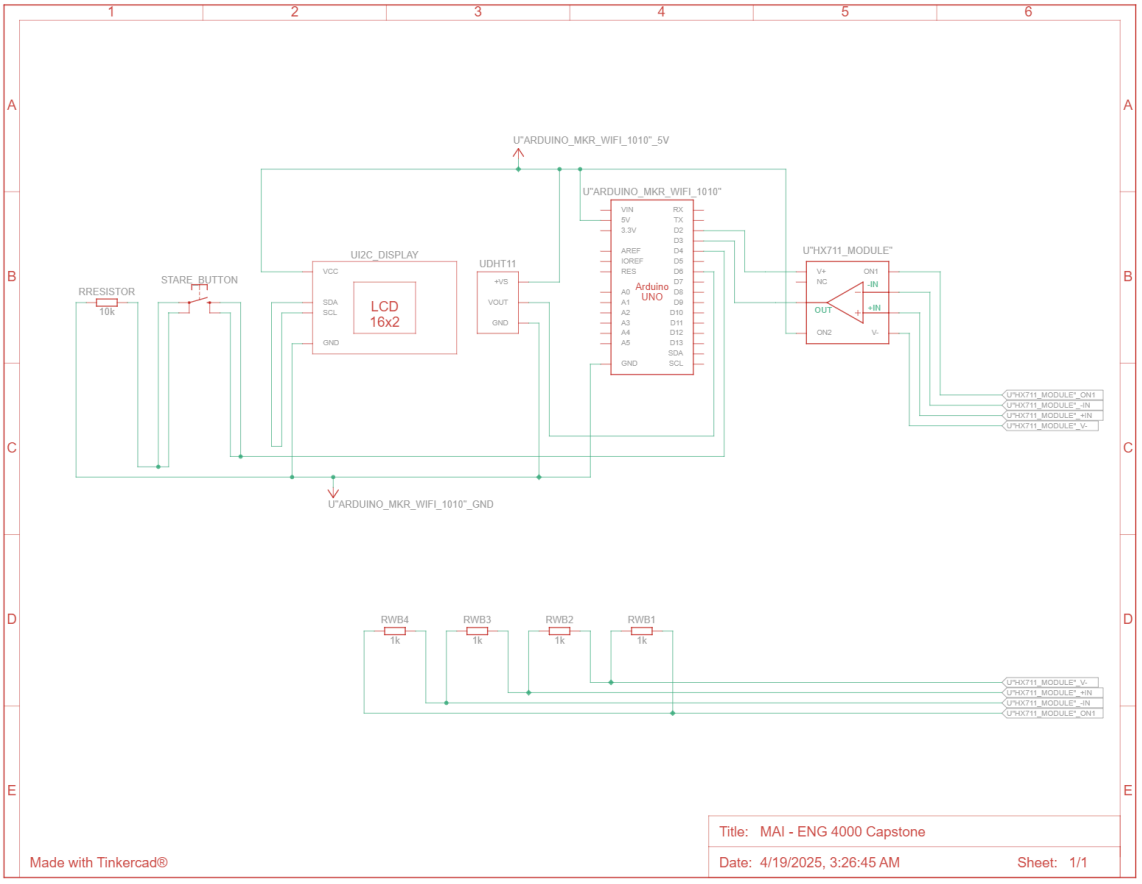


Figure 16: Schematic of MAI. The diagram may have small inaccuracies due to part-mimicking.

We have installed a load cell and HX711 module, which will allow us to measure the mass of spools and objects being placed on it. A load cell is a transducer that converts mechanical force into readable electrical signals [19]. Our specific device is a tension-based strain gauge load cell. This component measures the change in resistance from tension on the carriers, wires that strain gauges are bonded to. When tension is placed on the load cell, the carriers become thinner and longer, which increases resistance (Figure 17). These signals are passed on to our HX711 module, which reads and converts the signals into a digital reading. From here, we use a software-based calibration to attenuate the readings that match the mass of the object measured in grams.

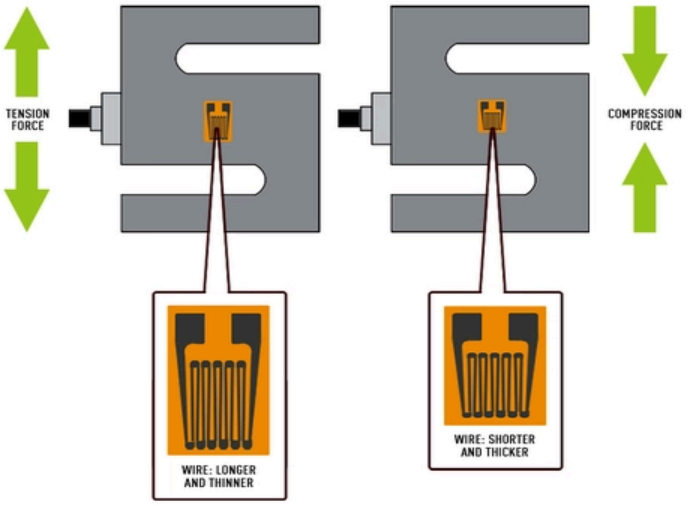


Figure 17: Tension-Based Strain Gauge Load Cell carriers changing shape due to applied forces [19]

The load cell is attached to two plates by using screws. One plate is attached to the bottom and the other to the top. This creates a base for the load cell to be placed on, as well as a platform to place objects to be measured. This unit sits on top of a specially printed housing, which contains other components of the scale, such as the display and tare button, to create an entire scale unit. The load cell we selected is rated to measure up to 5kg, which aligns with Pantheon's current needs, as the largest spools and resin containers are much lighter.

To replicate the tare function of a scale, we have used a tactile push button, in combination with a 10kΩ resistor to allow us to send a signal to the Arduino. Upon a button press, the Arduino will zero the current read value as the new baseline measurement. This can be helpful when needing to measure the mass of a specific component without including the mass of a component you have already accounted for. By holding down the button, a user can swap to one of three different measurement modes: standard / 0 mode, spool mode and resin mode. Long pressing the resin mode will loop the user back to the default 0 mode. The 0 mode is the setting that allows the scale to function as a basic scale, with no preset tare value. Spool mode is a special mode that automatically takes into account the mass of the spool and sets that mass as the zero value. Resin mode does the same thing, except it uses the mass of an empty resin container as its starting point. We have tailored the empty spool and resin container values based on the same equipment used by Pantheon 3DP, to make using the scale more convenient.

By using a DHT11 sensor, one could read both the temperature and humidity of the interior of the shelf. This is accomplished by fixing the sensor within the 3D-printed housing designed for it,

which we have attached to sit flush to the backside of the interior of the shelf wall, allowing it to get a consistent and precise reading (Figure 18).

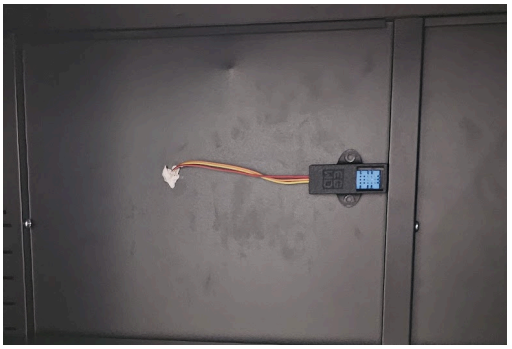


Figure 18: DHT11 in 3D-printed housing attached to shelf wall

MAI is fitted with an I2C LCD display, which allows us to display sensor data like temperature, humidity and the current measured mass all at once. The screen is also used to display errors, in the case that MAI is disconnected from the network or the MQTT broker is offline. We have also added a boot-up screen that sets up and initializes each component of the system, similar to a BIOS. The boot-up screen displays each step that is being performed, so in the case of a hang, we can determine which step the boot-up fails. The various wires and breadboards are used to foster the connections between components and provide a compact and flexible design, allowing for future extension. These components are protected by specially printed housings and wire covers that protect the system from the external environment and most minor damage.

With the ability to read sensor data and connect to a network, we can transmit the sensor data from the microcontroller to our application. This transfer is accomplished by utilizing paho-mqtt and the ArduinoMqttClient libraries to establish communication between the microcontroller and our application via the MQTT protocol.

In our use case, the DHT11 sensor records the temperature and humidity of the interior of our shelf. This data is then published by our Arduino to the broker, under the topics of temperature and humidity. The application subscribes to these topics, which allow it to receive the data being uploaded to the broker. We can then take the data from these topics and use it to update the shelf's temperature and humidity values in our database using backend controller methods that update the shelf's value (Figure 19).

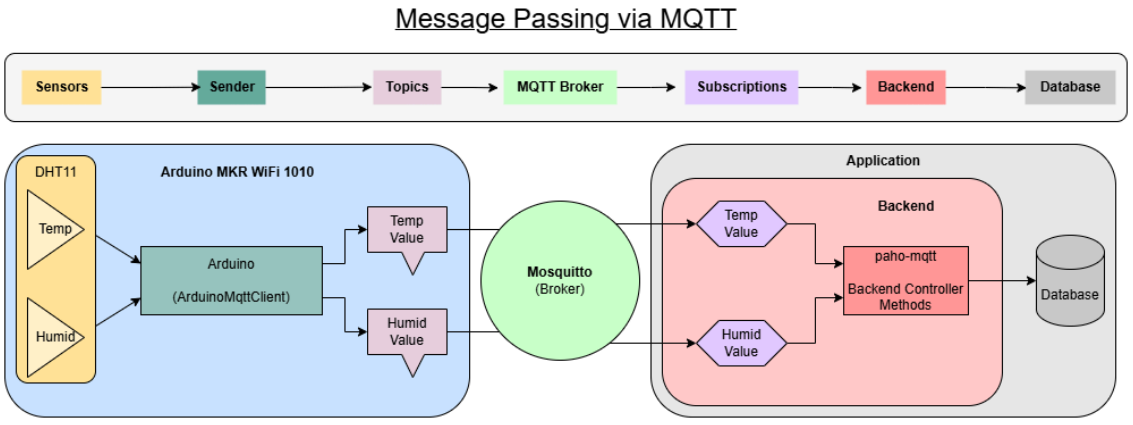


Figure 19: MQTT dataflow with a topic exchange

To ensure a consistent and constant uptime, we have programmed MAI with logic that allows it to reconnect to the WiFi network and the MQTT broker automatically, the moment connection is lost.

3.4.8 Shelving Design and 3D-Printed Housings

To regulate humidity and temperature, the storage vessel must maintain some level of enclosure. Structurally, the inclusion of shelving unit doors provides the necessary capability for enclosure (Figure 20). Although the doors provide some level of enclosure, they will not provide the necessary environmental regulation which is required to house Pantheon 3DP's raw materials. Therefore, insulating materials such as caulking and dehumidifying materials such as colour-changing desiccant have been included to regulate the shelf's internal environment. The shelving unit has been built in adherence to York University's health and safety guidelines. Furthermore, the insulation has been installed to mitigate the lack of enclosure caused by the shelf doors.

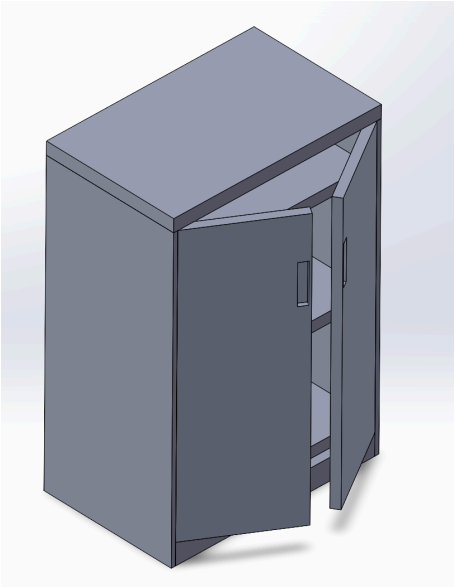


Figure 20: A SolidWorks rendering of the storage unit.

The shelving unit has been constructed using a kit that came with all the necessary components including all the panels and shelves (Figure 21). Once the shelving unit was assembled from the kit it was sealed to protect 3D printing filament from moisture and contaminants. Any edges where two sheets of aluminum meet creating a gap has been sealed with a silicone based sealant (Figure 22). The structure accommodates multiple spools of filament and is designed to optimize storage space while keeping the filament easily accessible.


Figure 21: The shelving unit with an insulated exterior with IoT housing (left)
Figure 22: Shelving unit with insulated interior (right)

To integrate environmental monitoring within the shelving unit, an Arduino equipped with temperature and humidity sensors has been mounted on the side panel. Additionally, a scale has been secured to the top surface using Velcro to allow for non-invasive and modular weight tracking of filament spools (Figure 21). To improve the enclosure's sealing capability, silicone caulking was applied along the front door seams. Duct tape flaps were added to seal the vertical gaps between the door edges and the cabinet frame, thereby minimizing external airflow and enhancing internal humidity control (Figure 23 and Figure 24).


Figure 23: Shelving unit with insulated top of the door (left)
Figure 24: Shelving unit with insulated bottom of the door (right)

For the moisture control within the shelving unit, a custom 3D-printed desiccant carrier was designed and fabricated (Figure 25). The carrier is intended to securely hold colour-changing desiccant while ensuring ease of replacement and maintenance. To minimize its spatial footprint inside the cabinet, a Velcro attachment point was added to the design, allowing the carrier to be mounted directly onto the interior face of the door (Figure 26). This mounting strategy preserves internal storage capacity while ensuring the desiccant remains effective in regulating humidity throughout the enclosure.

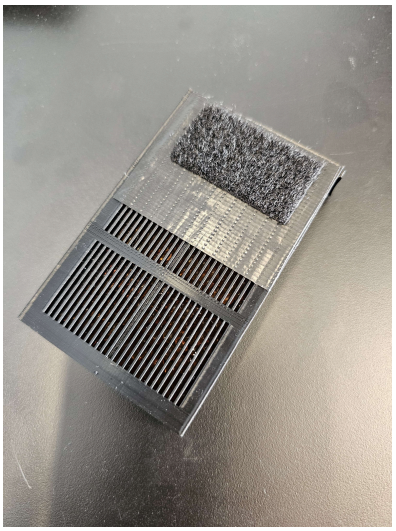



Figure 25: 3D-printed desiccant carrier (left)
Figure 26: Carrier velcro attachment (right)

3.4.9 Email Integration

Email notifications have been included as a feature within the application as a means of directly communicating with the users upon the initiation of certain events. Emails are sent when a user's password is updated, when a temporary password is generated, upon a temperature/humidity warning, during a sale for a specific material or when a material's stock is low. This functionality is achieved through an API call to SendGrid's automatic email platform [20]. The contents of the email are specified to their respective notification type through the use of the Wrapper design pattern [21].

3.4.10 Event Listening

The web application makes use of database listeners to appropriately trigger real-time notifications. This feature is implemented through the use of SQL Alchemy's backend *event.listen* method. By using *event.listen*, the backend can listen for specified changes to an entity within the database. This information is then forwarded from the backend server to the frontend server through WebSocket communication.

The material mass listener is one of the project's two event listeners. This component is responsible for triggering events when a material's mass is detected to be below 50g, a contact specified threshold. When an event is triggered, the backend triggers an email to be sent to every Super Admin through the mailer service. Furthermore, the backend sends a list of low stock materials to the frontend for display in the notification panel (Figure 27).

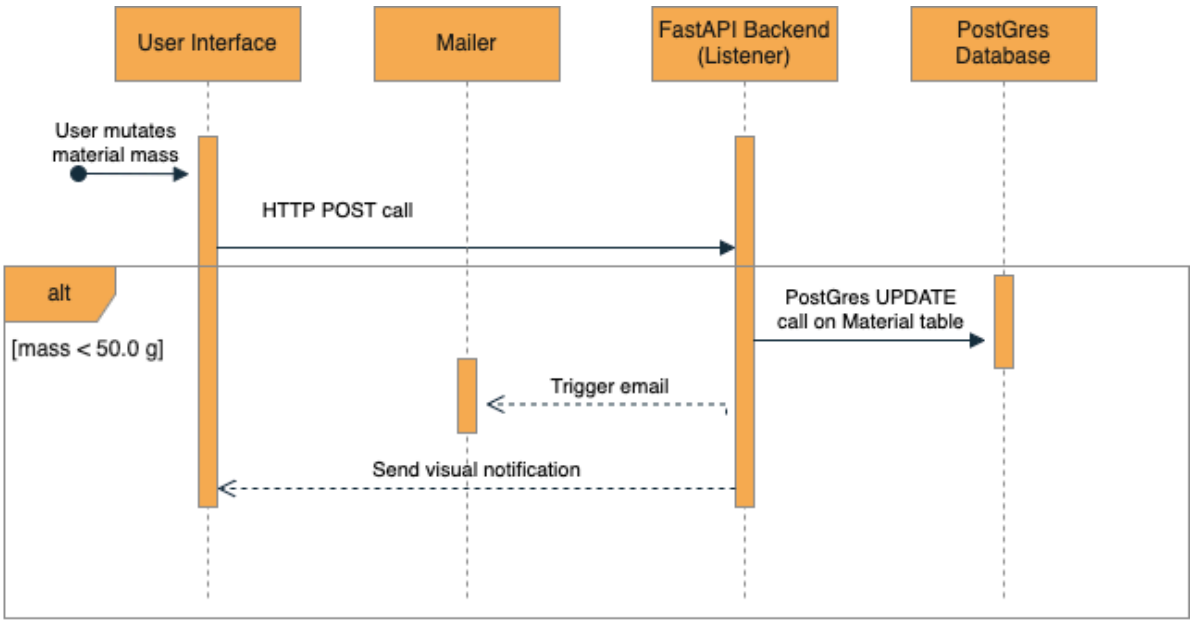


Figure 27: Sequence diagram outlining the mass listener's behaviour

The shelf listener triggers events upon an update to the Shelf database entity. Shelf database updates occur through MQTT-facilitated communication between an individual shelving unit and the backend server. When a shelf's mass is either detected to have a humidity greater than 20% or a temperature greater than 30 ℃, an event will be triggered. Once an event is triggered, an email notification will be sent to all Super Admins through the mailer service. Additionally, a visual notification will be present upon the frontend server through the notification panel (Figure 28). Much like the backend-to-frontend communication in the material mass listener, shelf-related notification panel updates are sent through web sockets.

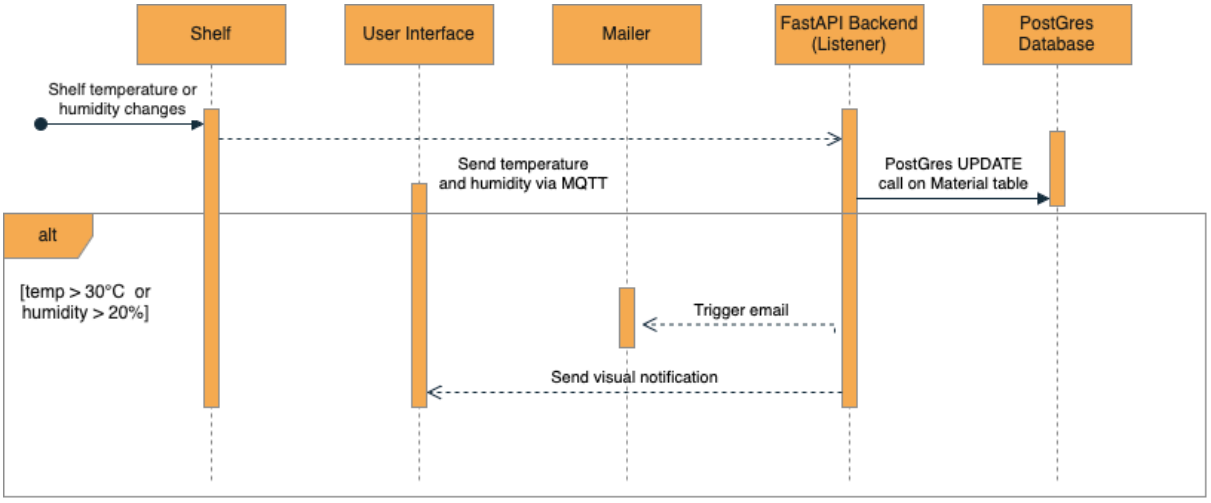


Figure 28: Sequence Diagram Outlining the Shelf Listener's Behaviour

3.4.11 Password Storage

As previously mentioned, passwords are unique in terms of datapoints as they cannot be stored as plaintext and must therefore be hashed. While many password hashing libraries exist within Python, Argon2 was selected due to its capacity to resist sophisticated parallel brute force attacks [22]. Therefore, this hashing selection ensures that the feature remains robust for the foreseeable future.

3.4.12 Continuous Integration Testing Pipeline

Continuous Integration is a software design practice which emphasizes the early verification of code changes. This concept was used to create a testing pipeline within our codebase. A Github Action is automatically triggered after a pull request is opened or after a merge into the main branch, such that a remote server runs the entirety of our software unit, integration and system (selenium) tests. When the pipeline is complete, a notification will be sent to the user. Should a test fail, the team's failure and issue tracking procedures will be used to swiftly resolve the deficiency. This system ensures that all new features are verified and do not break previously implemented features (Figure 29).

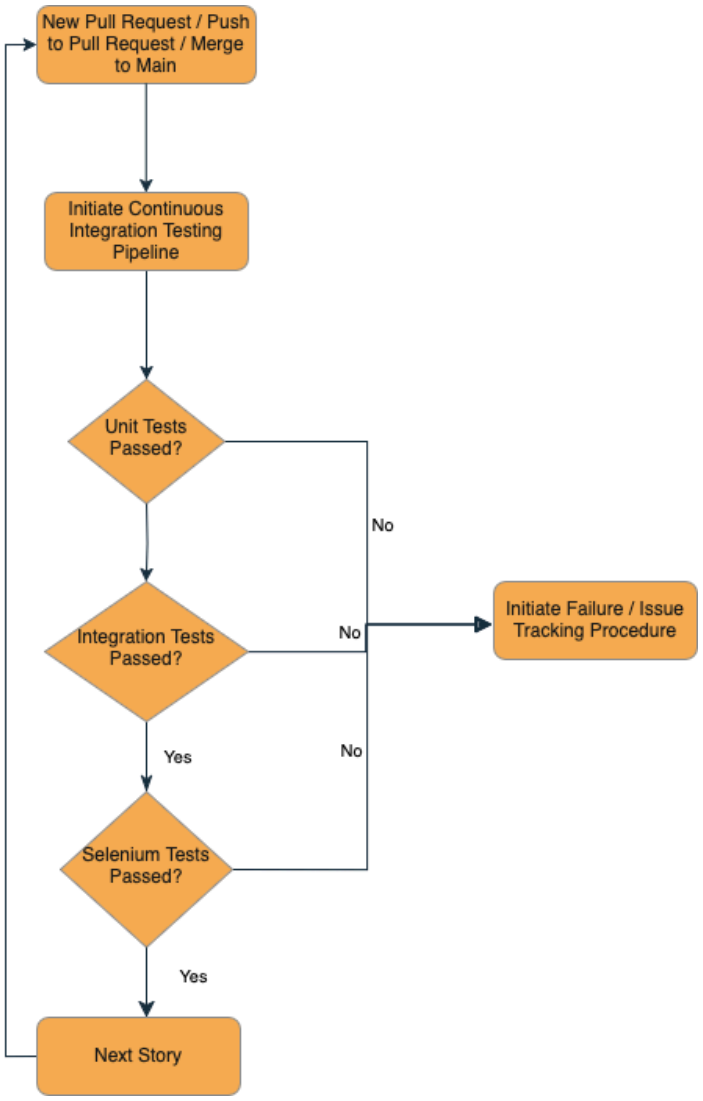


Figure 29: A Flowchart of the Testing Pipeline

3.4.13 Authentication Cookies

Authentication cookies were employed to enforce secure user login procedures, particularly due to the handling of sensitive credentials such as user passwords. Upon successful authentication, a session-based cookie is generated, containing an access token used to authorize requests to protected API routes. This mechanism ensures that only authenticated sessions can access sensitive user information, thereby preserving data integrity.

3.4.14 Handling Cross-Origin Resource Sharing

To mitigate Cross-Origin Resource Sharing (CORS) vulnerabilities, a reverse proxy was configured. This setup restricts API access exclusively to requests originating from the official frontend server, effectively preventing unauthorized third-party interactions and safeguarding the system against malicious data manipulation attempts.

3.4.15 Cloud Deployment

Cloud deployment technologies were used to ensure every software component is continually available with the potential for future scalability. The deployment process utilizes a remote database hosted via Supabase, while providing continuous deployment mechanisms for a Vercel-hosted frontend and a Microsoft Azure-hosted backend. Unlike a local database, the remote database acts as a source of truth for multiple devices. This centralized database position ensures that all system information is maintained with integrity. By hosting our FastAPI backend on Microsoft Azure, we guarantee the backend's availability while providing a mechanism for continuous update publishing. The backend and the database were connected via IPV4 as the current Azure service actively blocks IPV6 connections. This continuous deployment method is achieved by using a GitHub Action to deploy the updated backend upon a change to our repository's main branch (Figure AF16).

4. As-Built Design Compliance

4.1 System Hardware Budget and Component Justification

The strategic use of desiccant and sealing materials added significant value to the overall design while keeping costs low. Instead of purchasing a fully enclosed, climate-controlled cabinet, which can cost several hundred dollars, the team enhanced a standard $153.41 metal storage unit using $29.99 worth of reusable desiccant and $8.49 in velcro strips to mount the desiccant housing discreetly inside the door. To further regulate the internal environment, $14 worth of silicone caulking and duct tape were used to seal door edges and panel gaps, minimizing unwanted airflow and moisture intrusion. This insulation strategy created a cost-effective microclimate within the cabinet suitable for sensitive 3D-printing materials. The reusable nature of the desiccant, coupled with the permanent sealing improvements, ensures ongoing environmental stability with minimal future cost, making this one of the most budget-efficient features of the design.

4.2 Computational Resource Usage

The free subscriptions of Azure App Service, Supabase and Vercel were respectively tested to ensure they were capable of running the application with a high degree of performance. Although we initially budgeted for a paid remote server subscription, the feasible use of free subscriptions effectively aids in decreasing the overall cost of the project without sacrificing the solution's quality. The following subsections provide an analysis of each server's computational resource usage.

4.2.1 Remote Database Resource Use

The computational resource use of the remote backend was measured by using Supabase's reporting dashboard. The commitment of Pantheon's initial information to the database has accounted for 0.02 GB of disk space. This usage remains low considering that the free tier of Supabase allots 8 GB of

storage space. Furthermore, the relatively low average CPU utilization of 0.66% over one week starting on March 31, 2025 demonstrates the optimality of this free service. Therefore, the low CPU and disk use of the database, combined with the high ceiling of the free subscription, provides a robust long-term method of storing information within the system.

4.2.2 Backend Server Resource Use

We elected to use the B1 tier of Azure's App Service due to its guaranteed uptime and minimal cost. As Pantheon 3DP is a student-run organization, they are entitled to Microsoft's Azure Student Plan. This enables the project to use the B1 tier of the App Service at no cost [23], [24]. This plan allocates a one-core CPU with 1.75 GB of RAM. The average memory consumption of the server was measured using Azure's metric dashboard. Over one week of use, the average memory consumption of the server was 57.7 MB with a spike of 286.9 MB during intense load testing (Figure 30). This relatively low consumption represents the server's adequate memory capacity for the given application and capacity for scaling.

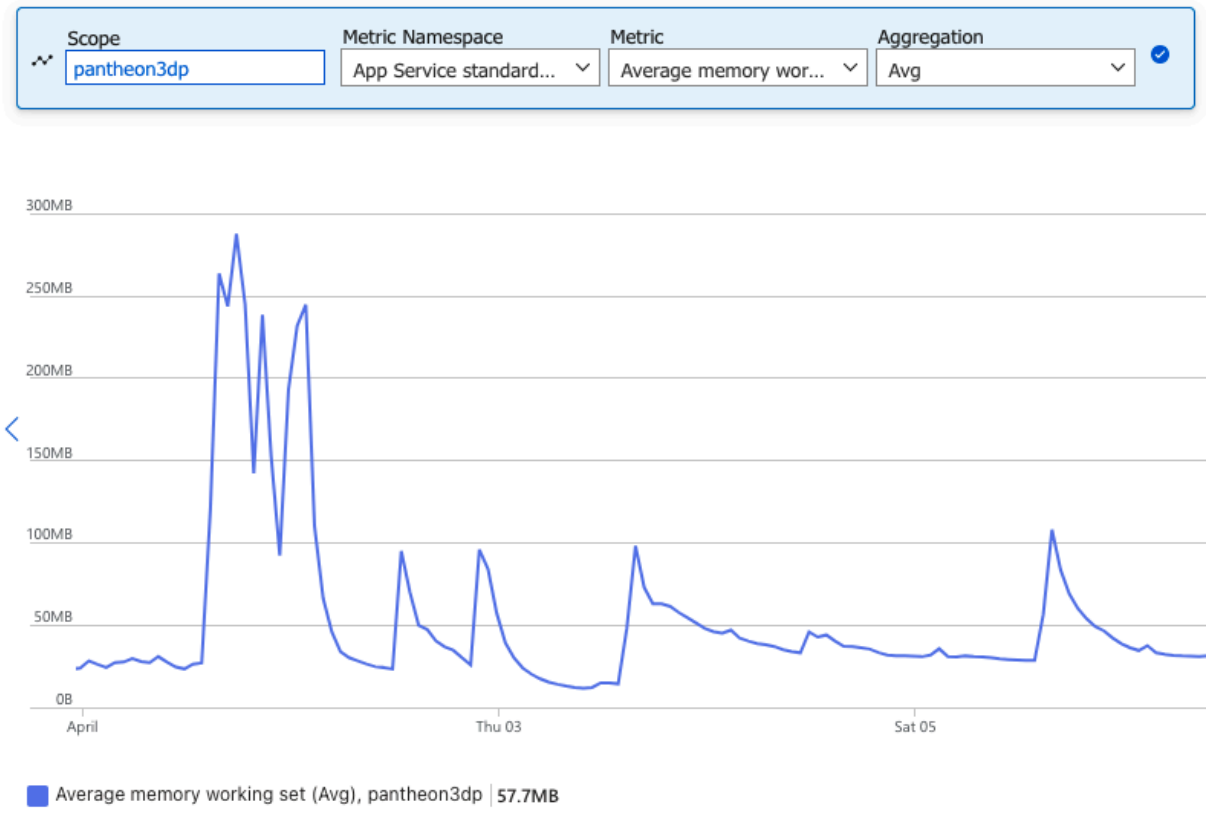


Figure 30: Backend server average memory use over one week

The response time of the backend server under load was tested via the *wrk* benchmarking tool. To complete this test, the server was flooded with simultaneous requests from 15 connections for 30 minutes. These 15 concurrent connections represent a significantly larger quantity of connections than the company's expectation of 3. The materials page was used as the request target as its centrality to the nature of the inventory management system will cause the page to be the most trafficked. The low average latency of 21.02ms per request demonstrates the backend's high performance in the face of a significantly higher than average concurrent load (Figure 31). Despite this low latency, the test result returned 45 timeouts in 1432008 requests. This low timeout rate can be attributed to temporary stalls

within the Azure network. Therefore, given the low latency and low timeout rate, the server can be considered to be effective under heightened load.

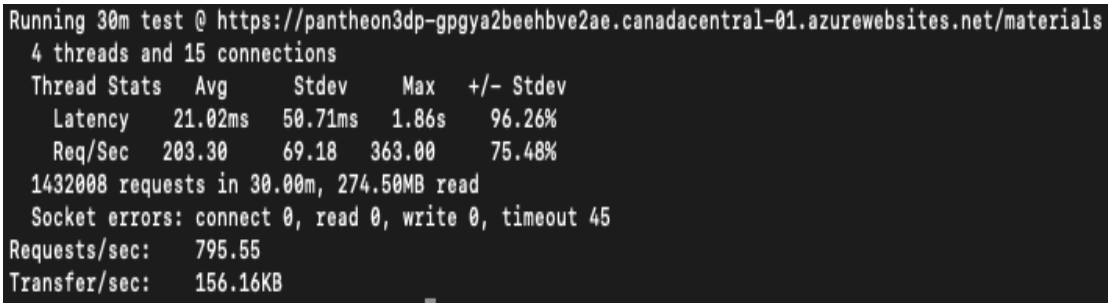


Figure 31: The results of load testing the backend server using Wrk

4.2.3 Frontend Server Speed Insights

To evaluate the performance and responsiveness of the frontend application, Vercel's built-in Speed Insights dashboard was utilized. The metrics are based on real user interactions collected over 7 days. One of the key indicators provided by this tool is the Real Experience Score (RES), which summarizes the overall quality of a user's experience on the site. During the evaluation period, the hosted application consistently achieved a perfect RES of 100 (Figure 32, Table 4). This score indicates that over 75% of visits provided a "great" user experience according to industry benchmarks. In addition to overall performance, route-level insights were analyzed. These metrics confirmed consistent responsiveness and stability across both static and dynamic routes within the application. Notably, the "/inventory" route, which represents one of the most frequently accessed pages in web applications, received over 624 visits and maintained a RES of 100. These findings, supported by both Speed Insights and manual testing, demonstrate that the frontend is highly optimized, responsive, layout-stable, and efficient under high traffic conditions. Such results were achieved through the strategic use of server-side rendering (SSR) and static site generation (SSG) capabilities offered by the Next.js framework, combined with performance-oriented development practices.

Table 4: Frontend Speed Insights Over 7 Days

Metric	Value	Notes
First Contentful Paint	0.52s	Fast time to visual feedback
Largest Contentful Paint	0.53s	Indicates fast loading of main content
Interaction to Next Paint	144ms	Responsive to user inputs
Cumulative Layout Shift	0.01	Stable visual layout
First Input Delay	7ms	Near-instantaneous input handling
Time to First Byte (TTFB)	0.1s	The backend/server is responding quickly

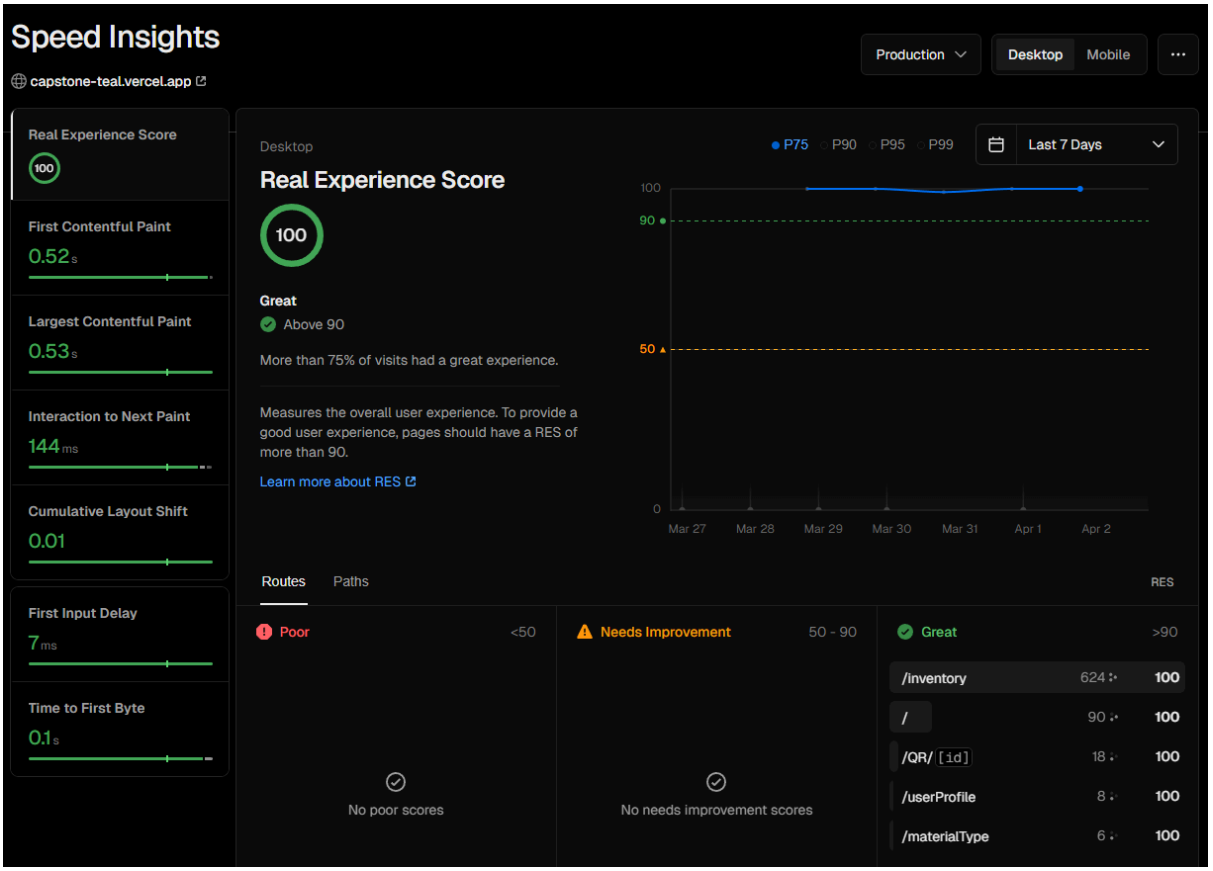


Figure 32: Speed Insights of Load Testing Commonly Used Routes

Fast Data Transfer was also considered during the selection of the appropriate Vercel hosting plan. The Pro plan provides up to 1 TB of fast data transfer, whereas the Hobby (free) plan is limited to 100 GB [26]. Following frontend load testing, Vercel's observability tools reported transfer metrics, which showed that for a small-scale application, the usage remained well within the limits of the Hobby plan. Over 7 days (scaled from 12-hour data, as the Hobby plan restricts long-term visibility), the application only consumed approximately 28 MB of outgoing and 1.41 MB of incoming fast data transfer (Figure 33). This indicates that the application is network-efficient and that the free tier is sufficient for its current scale.

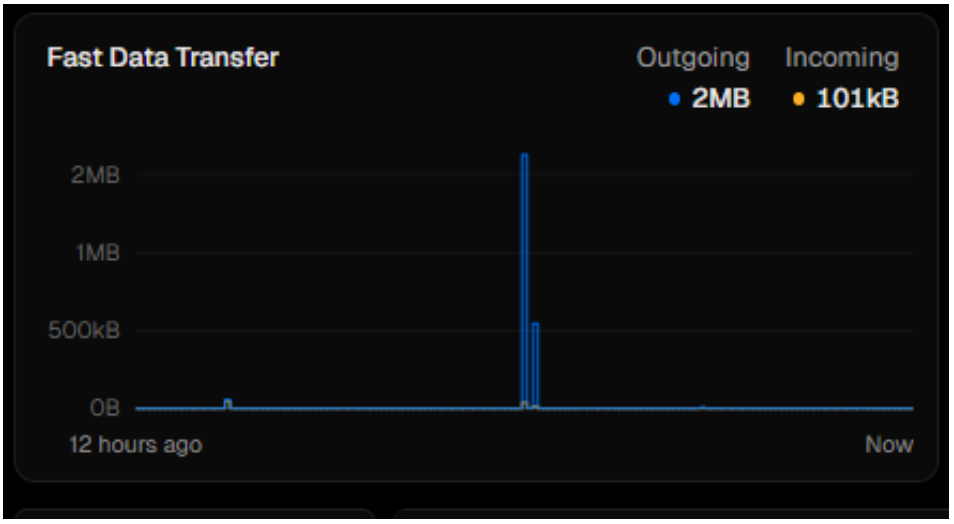


Figure 33: Fast Data Transfer Insights

4.2.4 Overall Computational Resource Consumption

The following table provides an organized summary of the solution's computational resource usage during the first week of its deployment (Table 5). As explained in the previous subsections, the project benefited from free-to-use software services while maintaining a high-quality standard. This careful balance between quality and cost ensures that the project's software components have an extensive longevity.

Table 5: Computational Resource Consumption

Resource	Consumption Over A 7-Day Period	Cost
Supabase PostgreSQL Remote Database	0.02 GB of storage	$0.00
Azure App Service (B1 Tier) Backend Server	57.7 MB (average)	$0.00
Vercel Frontend Server	28 MB Outgoing 1.41 MB Incoming	$0.00

4.3 Microcontroller Resource Use

The electronics used for MAI are all relatively cheap and low-power devices. Unlike actuators, sensors like the DHT11 and load cells require very little power draw, as well as our microcontroller, the Arduino MKR WiFi 1010 board. Embedded devices like these are designed for low power and are relatively cheap components, making them efficient in both power and budget costs. All of these components are powered by a 5V, 2A power supply. This would be the equivalent of keeping a smartphone plugged into the wall 24/7, which overall consumes little power compared to appliances like ovens and washing machines.

As for the IoT factor, the board is connected to York University's AirYorkGUEST WiFi, which is designed to handle traffic for users across the Keele campus. We have programmed the controller to only send updates when a change in values are detected, rather than a specific time interval. This helps limit the number of messages we send drastically, as going by time intervals (our initial concept of sending messages every 5 seconds) would have caused us to send around 51,840 messages per day (as illustrated by the formula below).

$$Number\ of\ Messages\ =\ Number\ of\ Messages\ Sent\ Per\ Day\ \times\ 3\ Sensors$$
$$Number\ of\ Messages\ =\ (24\ hours\ *\ (3600\ seconds\ in\ an\ hour\ /\ 5\ seconds\ per\ message))\ *\ 3$$
$$Number\ of\ Messages\ =\ 17280\ messages\ *\ 3\ sensors\ =\ 51,840\ messages$$

The messaging protocol we use to transmit our data, MQTT, is relatively lightweight. This allows us to take up minimal bandwidth for sending messages over the network. A device like this generates virtually no traffic, compared to conventional device use. For instance, a smartphone watching a video in High Definition uses around 500MB per hour, which is very commonplace in a campus of 50,000+ people in size [25]. Even going by the original design that sends messages every 5 seconds, that would be 259.2KB of data, or 0.25MB of data (as illustrated by the formula below), which would be equivalent to a few text messages using only plain text.

$$Data\ Used\ Per\ Day\ =\ Message\ Size\ *\ Number\ of\ Messages\ Per\ Day$$
$$Data\ Used\ Per\ Day\ =\ 5\ bytes\ (average\ size\ of\ our\ string\ message)\ *\ 51,840\ messages\ per\ day$$
$$Data\ Used\ Per\ Day\ =\ 259.2\ KB$$

Based on this data, it's easy to see that our microcontroller is essentially undetectable from a power and bandwidth perspective. With its relatively low cost of parts, it makes implementation of MAI very cheap, cost effective and inoffensive to introduce to Pantheon3DP's office.

4.4 System Strengths

Pantheon 3DP Printing Material Management stands out for its integrated inventory management apparatus with a centralized web application acting as its lynchpin. The system's physical storage solution provides the necessary insulation to ensure additive manufacturing materials can be stored without the risk of degradation. The insulating strength of the system was highlighted during a vandalism attempt when the unit, despite having a majority of its sealant removed, was still able to maintain an acceptable temperature and humidity level. The unit's space is maximized through the use of velcro-attached desiccant holders, which enable each shelf within the cabinet to be exclusively used for material storage without sacrificing the aforementioned insulating properties. The careful balance of space maximization and insulation ensures that the company has a long-term physical storage solution which can be relied upon.

The system's web application particularly excels at providing a streamlined interface for inventory visualization and manipulation without sacrificing safety, cost or reliability. The remote nature of the application enables the company to use the system on a multitude of internet-connected devices, thus increasing its accessibility. Furthermore, the web application's remote deployment enables the service to have uninterrupted activity, an important characteristic in real-time monitoring. Lastly, the use of relevant security measures such as a user hierarchy, password hashing and email notifications effectively prevents unauthorized access while providing real-time access-related notifications to the user. The strength of the application is compounded by its zero-cost nature. Therefore, the system's web application provides a reliable and user-friendly interface at no cost.

An often underrated strength of the web application is its scalability. By using Github Actions to establish continuous integration and continuous deployment pipelines, the system effectively streamlines the workflow of future development teams such that future features can be automatically verified and deployed. Moreover, the low memory and storage use of the application provides ample resources for future development by enabling successor teams to focus on creating features rather than redesigning the cloud infrastructure. Therefore, the scalable design of the current application provides the foundation for future development through continuous verification, continuous deployment and resource-efficient server use.

Lastly, the system's use of IoT technologies effectively facilitates real-time communication with the user, ensuring relevant updates are instantly sent. This strength is achieved through the use of the MQTT protocol. The low-latency nature of MQTT facilitates communication between the web application and the microcontroller without the need for a strong internet connection. Therefore, communications remain reliable even in the event of network instability.

By utilising electronics like sensors and our microcontroller, we are able to remotely monitor the conditions of our shelf and automate warning messages for instances where the conditions of the

shelf's interior environment are unfavourable. This gives peace of mind to our users that materials are being stored in conditions that optimize their longevity.

With the addition of the scale electronics, we can even automate part of the material consumption process. Users will weigh the mass of a spool after each print. By relaying the mass data to our application, we can accurately determine the amount of material used for a print, removing the effort required for a technician to manually calculate, subtract and enter the new mass of materials after a print.

4.5 System Weaknesses

Like with any tool, despite many strengths, weaknesses will always exist. This system is no exception. **The system is particularly limited in its MQTT broker, monolithic database usage and cloud limitation.** The most pressing weakness is the system's reliance on *test.mosquitto.org* as a MQTT broker. While we have not experienced significant outages while using the broker for one month, Eclipse, the broker provider, warns of potential restarts throughout the site's lifecycle. Therefore, while the broker has been reliable throughout the development and verification process, there may be a future instance of a brief outage resulting in temporarily delayed communication. Furthermore, the unencrypted nature of the broker may result in unwanted actors collecting temperature, scale and humidity-related data. While this information is not sensitive, its exposure nonetheless provides a security flaw in an otherwise secure system. Therefore, reliance upon *Mosquitto* as a broker poses minor reliability and security risks.

While the web application is highly scalable, its monolithic database reliance poses a minor weakness to future expansion. As the application scales, a singular database instance will inherently create bottlenecks due to the limited amount of available memory conflicting with an increase in memory-demanding connections. Using microservices, an architectural pattern by which individual services receive one database instance, may aid with feature scaling, although for the current application, the pattern is unnecessarily resource-intensive. Therefore, the monolithic database architecture, while optimal for the current set of features, limits the application's scalability.

The free nature of the cloud deployment system is a significant strength in maintaining the project's reliability, safety and accessibility. Ultimately, the ability to connect to the Azure backend via IPV6 was sacrificed for free of cost. The lack of IPV6 connection may decrease the system's performance as its features grow. For instance, as more software services begin to prefer IPV6 over IPV4, they may prioritize IPV6 compatibility. This weakness is heightened for the system's IoT features as services within the field, such as Contiki-NG and RIOT OS do not allow IPV4 connections [27], [28]. While this weakness is insignificant to the current solution's capabilities, it may limit the possible range of tools which future teams may use to scale the product.

4.6 Testing

4.6.1 System Failure Tracking

Our current software system failure tracking framework revolves around the Github platform. System modifications are developed during our biweekly sprint planning meetings. These modifications are decomposed into stories through a Jira storyboard. Github branches, along with pull requests are used to ensure that new features are not added into the master branch until at least two members of the

team have approved reviews. All modification stories are expected to contain the relevant test suites to verify their function.

Software feature verification is managed through the aforementioned CI pipeline. Should a test remotely fail, the team will be notified by Github via email (Figure 34). Upon the discovery of a failing test, a member of the development team manually assesses the issue. Should a bug be discovered, a Github Issue is added to the repository to track the form and status of the deficiency, and a story is added to the top of the team's backlog via Jira. Naturally, bugs are rapidly dealt with to ensure the pace of feature development remains constant.

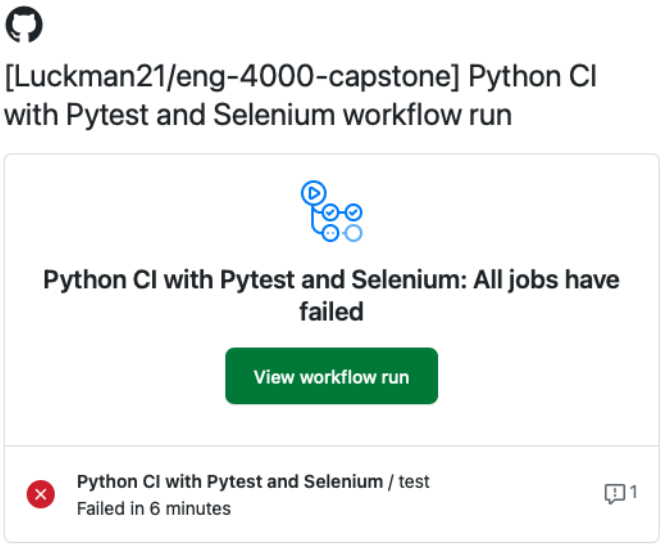


Figure 34: An example email notification for an unsuccessful code change

4.6.2 Software Testing Outcomes

As explained in section 3.3.5, a combination of unit, integration and system tests are typically used to verify individual software components and their interactions with one another and has been used in this project. For a code change to be officially implemented, it must pass any newly created tests along with every existing test to ensure the implemented feature synergizes with the rest of the system. Given that, by the end of the project, all tests pass, the software verification process can be deemed to be a strong reflection of the web application's adherence to the project requirements. Refer to AF1 for the test results.

4.6.3 Shelf Test Results

After extensive testing, the shelf can maintain a very consistent and optimal interior climate, despite the wildly changing environments it has been subjected to. The shelf unit sat in a garage for a week, with the garage door being opened and closed at various points in time. Over that week, the garage door was open during times of intense rainstorms and dry, hot weather days. Despite this, our shelf showed resistance to change. It maintained a very low relative humidity between 10-18% and 18-25°C, both of which are within the ideal ranges we targeted (Figure 35).

Figure 35: Govee Sensor showcasing the shelving unit's interior climate after 1 week of testing

4.6.4 Microcontroller Test Results

Testing our microcontroller requires a mixture of both software and physical testing. Unit tests are created to test every individual component functions on its own, while integration tests consist of field and endurance testing to ensure the circuitry is responsive and functioning. By logging uptimes and responsiveness via messages to our backend, we can determine whether the circuitry is alive and sending messages. We can validate the accuracy of the sensors and scale measurements via calibration tests. We compare the recorded values from our microcontroller to the ones we read from other external sensors that Pantheon uses that capture the same values. By performing these tests, we have ensured the values and readings fall within the accepted error ranges (+/-2°C, +/-1% RH, +/- 1g).

4.7 Non-Conformance Disposition

Several non-conformance dispositions were detected throughout the verification process of the project (Table 6). This section provides detailed information on the nature of each problem. In the following section, the solutions by which conformance was restored will be outlined.

Table 6: Summary of Non-Conformance Dispositions

Issue ID	Issue Title	Issue Description
Issue 1	Users List is Not Properly Protected	"Right now, if I'm an Admin, I can simply enter the user page by editing the URL, subverting our hierarchy." - Luca Filippelli
Issue 2	Microcontroller Information is Not Shelf Specific	"The microcontroller successfully sends information to the server, but we can't tell which shelf it belongs to." - Luqmaan Irshad
Issue 3	Weak Association Between a Material and a Material Type	"It seems like the materials can exist outside of a material type which doesn't make much sense." - Luca Filippelli
Issue 4	Unsecured Circuitry Causing Monitoring Gaps	"Since some of the components are not soldered, the connections may become loose if someone tugs on them."

		- Luqmaan Irshad
Issue 5	Slow Listener Execution on the Remote Server	“The mass and shelf listeners execute in real time on a local server but are incredibly slow on the remote servers.” - Luca Filippelli

4.7.1 Issue 1: Users List is Not Properly Protected

On March 2, 2025, after completing the quality assurance tests for Sprint 7, the team discovered that the user permissions were misconfigured. In the ideal state, Super Administrators would be the sole group of individuals with access to the site’s user manipulation functions. The misconfiguration at that time would have allowed unpermitted users to access the page by manipulating the URL in their browser, thus failing the user hierarchy requirement.

4.7.2 Issue 2: Microcontroller Information is Not Shelf Specific

On March 9, 2025, after testing the initial messaging capabilities of MAI, we identified a unique challenge that our current method of communication presented. Our backend application subscribes to MQTT topics, which are what MAI publishes sensor and scale mass data to. Since we are only designing one shelf (and by proxy, one MAI unit), we know any data being received by the application is meant for shelf 1. This allowed us to temporarily hardcode some of the logic to update shelf 1 only. The realization was, in the event of a scale-up, we wouldn’t be able to determine which MAI unit was publishing data. This is because MQTT data does not explicitly contain sender/receiver information in its messages. This means that the receiver would end up becoming one big queue of messages, with no information on which shelf the data is for. Since we hardcoded all the updating logic in our backend to shelf 1, we also needed to ensure that we could somehow determine which shelf the data belongs to, so we could properly attribute data to a shelf.

4.7.3 Issue 3: Weak Association Between a Material and a Material Type

During the testing process for Sprint 7, the team discovered that a material type can be deleted such that its previously associated materials have a null material type. This premise is illogical, given that a material’s type is an unchanging physical characteristic. For instance, PLA cannot transform into a resin-based material. Given the centrality of the inventory management system’s material tracking features, it was crucial to rectify this bug immediately.

4.7.4 Issue 4: Unsecured Circuitry Causing Monitoring Gaps

Throughout Sprints 7-9, the inclusion of scale functionality has added several complications to the circuitry. This feature was added very late into the project scope, due to Pantheon 3DP’s order application being indefinitely paused. This change resulted in unsecured wiring which increases the risk of tampering. The quality of the wires is lower than we would like due to most of them being used, and a lot of the wires were unideal in terms of length; some were too short, others were too long.

4.7.5 Issue 5: Slow Listener Execution on the Remote Server

During the quality assurance testing for Sprint 9, the team simulated the execution of the listeners. The material listener execution was induced by simulating a change in a material’s mass through the user interface. The shelf listener’s execution was simulated by sending a mock MQTT message with a

humidity value outside of the acceptable threshold. While the feature worked as expected when using a local database, both listeners took several seconds to respond to their given trigger actions when the system was connected to the remote database. The high listener latency ultimately subverts the real-time communication requirement, thus requiring a solution to maintain conformance.

4.8 Non-Conformance Resolutions

4.8.1 Issue 1: Users List is Not Properly Protected

The team discovered that the source of this bug was poor state management within the user page. Previously, the user's status was not being set when attempting to access the page. To rectify the issue, the team added a *UserRole* type to the page's state, thus ensuring that the user's status can be accurately verified should they attempt to gain access by modifying the site's URL. This solution was later verified in the system test suite and remains a fixture in the project's CI testing pipeline.

4.8.2 Issue 2: Microcontroller Information is Not Shelf Specific

To solve this challenge, we took inspiration from the HTTP protocol. HTTP data packets consist of two parts: the Header and the Data. The header is the part of the packet that stores metadata like source and destination IP addresses. The data segment, known as the payload, contains the actual data that is meant to be transmitted to the receiver. Following this structure, we decided to change the format of the message to resemble the HTTP data packet, instead of raw information. The message format is now as follows *shelf_id|data* (Figure 36).

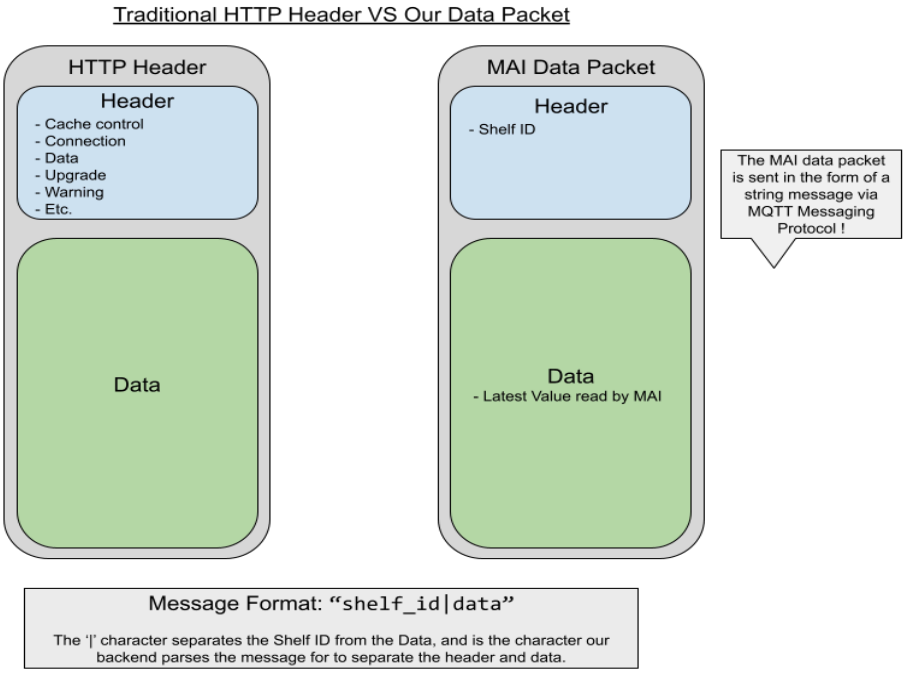


Figure 36: HTTP header VS our data packet

We've designed the backend methods to be able to parse this data packet, so it can identify the Shelf ID the data is addressed to, and update the correct shelf in the database. Using this approach, we can easily scale the application's messaging capabilities by allowing multiple shelves to send messages to our backend, using the same channels to communicate.

4.8.3 Issue 3: Weak Association Between a Material and a Material Type

This weak association was caused by a database misconfiguration, which allowed for a material type instance to be deleted even if it had an existing association with material instances. For instance, in this configuration, the PLA material type could have been deleted despite there being 3 PLA materials within the existing inventory. This bug was rectified by altering the database schema to ensure the material table's *material_type_id* attribute could not be null. The solution was then verified through unit testing.

4.8.4 Issue 4: Unsecured Circuitry Causing Monitoring Gaps

Alas, we worked with what we had access to and were still able to get the product running. Over time, we have worked on mitigating this issue through the use of soldering, creating housings for our circuitry and tying wires together so they are protected. This is still not a foolproof solution, but it does address some of the insecurities regarding the unsecured wiring.

4.8.5 Issue 5: Slow Listener Execution on the Remote Server

In the previous setup, the remote database used a common connection link. The team realized that this configuration could not possibly work for the listener as, unlike the rest of the application, the listener required an asynchronous database connection. Furthermore, the team discovered that the synchronous database connection was creating delays in asynchronous queries as the link was incapable of providing a persistent session. To resolve this issue, the team opted to use Supabase's session pooler connection link for every asynchronous database call as it provided a persistent session for the listeners.

5. Project Management (Agile Methodology)

5.1 Project Task Management Status

5.1.1 Agile Roadmap

To display our agile roadmap, we will focus on the Now-Next-Later chart as a reference. The general path in which features will be implemented is guided by a Now-Next-Later table; for more information, refer to Appendix AT2. This table categorizes each set of goals into three categories. The Now category emphasizes the immediate need of this objective set. The Next category outlines the subsequent goals which must be completed. Lastly, the Later section lists the lower priority time-weighted goals. Like with all agile objectives, the positions of each table item are subject to change as the team progresses through its sprints.

During February, the team updated the pathway starting from sprint 5 all the way to sprint 9 (Figure 37). The parallel sprint labelled "Luca Luxury Sprint" was added to depict the period during which a material sale web crawler project was completed. While the crawler integrates with this project's system, it is out of the scope of the project and therefore not included in its analysis. This pathway represents the alpha and beta release track that we used to present the project's timeline. The stories include the topics of notification and login features, cloud deployment, spool holders and sensor housing, climate stabilization, and rebranding to fit Pantheon 3DP.

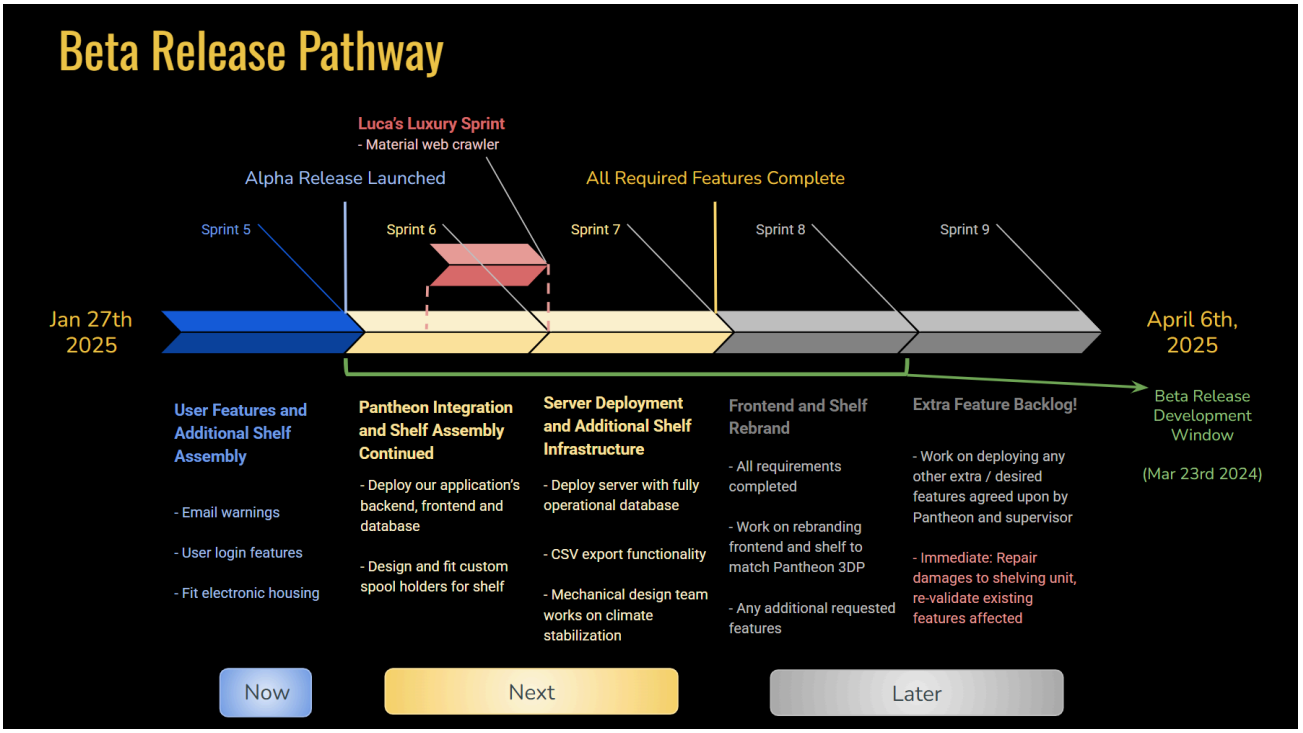


Figure 37: Updated feature release pathway

The updated beta release pathway primarily accounted for Pantheon's order application not being complete. This nonexistence caused the integration feature within the original Sprint 6 to be mute. Therefore, the team decided to implement a QR-based material updating system in place of the planned order app integration. The new release pathway was fully followed with two key exceptions. Firstly, the team learned from its delay of Sprint 3 that end-of-semester sprints will require more time due to the increased concentration of course deadlines and exams. As the team had several projects, assignments and exams during Sprint 9, its duration was extended by two weeks to ensure all tasks could be completed without exceeding the timed end of the sprint. This alteration ensured the sprint window is appropriate given the high-workload circumstances of the end of the semester. As for the second alteration, the team was impacted by vandalism to the shelving unit. For more information on this, please refer to 5.3.2. The damages caused an immediate addition of the following items to the final sprint: project repairs and feature revalidation for affected features. To see the original pathway that did not account for these deviations, please refer to Appendix AF14.

Finally, story points have been assigned on a scale of 1 to 10, where 1 represents a unit of work which can be individually achieved within an hour and 10 represents a unit of work which requires the whole team to complete. This point system is applied to each generated story during a sprint planning meeting to ensure the intensity of each task is quantified. If stories have dependencies on each other, Jira's flag tool is utilized to mark work which is blocked for the time being. When a story becomes unblocked, the flag is lifted, signifying that the work is ready to begin. For more information on the story point divisions and descriptions for each sprint, please refer to Appendix AF12.

5.1.2 Burndown Charts

To reflect on the team's performance throughout the sprints, we can refer to the burndown charts that were generated by the Jira platform. For more information, refer to Appendix AF3 to Appendix AF11. To simplify the chart, the two most important parameters to look at are the remaining work, which represents the number of story points left, and the guideline, which is the line of best fit representing the ideal burn rate. Another factor to note is the "planned end" line, which is a soft deadline of when the sprint is due.

Generally speaking, the team works at an effective pace. Throughout our sprints, a significant decline in story points is shown just before the sprint deadline, which indicates that a lot of tasks are finished at the team's desired pace. Despite the ideal pace, some lags in the burndown charts can be explained by the typical increase in the intensity of school work throughout a semester's lifetime. For example, during sprint 2, which occurred in November, a long lag time is displayed in the chart due to the team members' preparation for the examination period. Another key point seen in these charts is the periodic addition of story points during a sprint. This is largely due to addressing project sponsor feedback.

5.2 Resource Allocation Matrix (RAM)

RAMs help ensure project resources are effectively distributed to meet the needs of every decomposed task. Therefore, each RAM is completed during the sprint planning meeting for its corresponding sprint. Given the extensiveness of each RAM, this project's RAMs can be found in Appendix C.

Throughout the project, as part of the resource allocation process, the team tracked and reported total story points worked during each sprint, broken down by project month and sprint (Table 7, Table 8). As with the traditional form of the Agile Methodology, story points rather than hours are used to track estimated work intensity. As further work became unblocked from sprint to sprint, the team experienced a spike in story points during January, February and March. Overall, the estimates evolved to reflect a deeper understanding of task complexity and required effort as the project progressed.

Table 7: Summary of Story Point Divisions for Each Sprint

	Story Points for Each Sprint									
	Sprint 1 (Nov 1 - Nov 10)	**Sprint 2 (Nov 11 - Nov 22)**	**Sprint 3 (Dec 25 - Jan 12)**	**Sprint 4 (Jan 14 - Jan 26)**	**Sprint 5 (Jan 26 - Feb 10)**	**Sprint 6 (Feb 10 - Feb 24)**	**Sprint 7 (Feb 24 - Mar 10)**	**Sprint 8 (Mar 10 - Mar 24)**	**Sprint 9 (Mar 24 - Apr 21)**	**Total Story Points**
Luca	6	10	13	22	22	14	18	13	17	135
Luqmaan	3	5	0	12	4	7	12	6	14	63
Ethan	2	8	6	7	3	2	12	6	18	64
Jean	2	0	2	0	5	8	6	5	2	30

Paras	5	6	3	6	9	9	9	6	10	63
Tharuveen	4	3	5	6	7	10	3	5	9	5

Table 8: Summary of Story Point Divisions for Each Month

	Story Points for Each Month						
	November	**December**	**January**	**February**	**March**	**April**	**Total Story Points**
Luca	16	6	40	34	31	8	135
Luqmaan	8	0	14	15	19	7	63
Ethan	10	3	12	9	21	9	64
Jean	2	2	5	11	8	2	30
Paras	11	3	10	18	16	5	63
Tharuveen	7	5	10	14	11	5	52

5.3 Project Schedule

The team has made a full, up-to-date timeline of all of our planned sprints and releases (Figure 38). Being an Agile project, the dates and tasks were finalized near the start of their respective sprints. Initially, the timeline was shown as a Gantt chart, but since the project didn't have any overlapping sprints, a regular timeline was made. The timeline has been updated as of April 15th, 2025 and will be confirmed upon the completion of the capstone project. The gaps in November and December represent dates where the team respectively focused on final course assignments and celebrating the winter holidays. Also, during sprint 9, the month-long sprint was due to the team's final examinations.

Task	Status	Start date dd/mm/yy	End date dd/mm/yy	Milestone
Sprint 1	Completed	01/11/2024	09/11/2024	3D model Early Interface
Sprint 2	Completed	14/11/2024	22/11/2024	BOM Notification system
Sprint 3	Completed	25/12/2024	13/01/2025	Material Acquisition CRUD Expansion
Sprint 4	Completed	13/01/2025	26/01/2025	Shelving Unit Assembly Front-end back-end communications
Sprint 5	Completed	26/01/2025	10/02/2025	Insulation of shelving sensor calibration password hashing
Sprint 6	Completed	10/02/2025	24/02/2025	Circuitry and housing reassess redeploy options bug fixes Server Deployment
Sprint 7	Completed	24/02/2025	10/03/2025	Notification System Refinement Additional Shelf Infrastructure
Sprint 8	Completed	10/03/2025	24/03/2025	Front-end Shelf Rebrand
Sprint 9	In progress	24/03/2025	21/04/2025	Extra Feature Backing

Figure 38: Project timeline as of April 15, 2025

After the dates are updated in the timeline, the team finalizes the sprint milestones as per the project pathway. The tasks will be divided into stories following the process in 5.1.1. This process has already been implemented for sprints 1 through 9.

5.3.1 Stakeholder Feedback

Throughout the design process, we have maintained bi-weekly meetings with Nick DiScipio, the CEO of Pantheon 3DP. These meetings have enabled us to receive feedback on the current MVP features. As of now, Nick is satisfied with the current features and anticipated design path. For more information, refer to Appendix AF15.

5.3.2 Damages to our Shelf - Negligence From Other Groups and the Vandalism Incident

On April 3, 2025, the team discovered that the physical shelving unit was a victim of an act of vandalism. The extent of the vandalism was further confirmed by the unresponsiveness of the IoT infrastructure. The perpetrator removed the shelf's sealant, disassembled the scale forcefully using a screwdriver and vandalized the shelf using our caulking. This damage set back our team weeks for features and functions that were previously validated. For more detailed information on the incident, please refer to Appendix E.1, which includes a detailed overview of the damages, along with images.

Other minor damages have been caused to our shelf over time, mainly due to negligence. Many of these were from people using our shelf as a table, despite the signs and equipment on the shelf mentioning "CAPSTONE TEAM 8 - Do Not Touch". The extent of these damages is mainly cosmetic with dents to the top of the shelf from other Capstone teams dropping metal tools on our shelf as if it was a workstation.

As of the time of writing, we are still working with Security on identifying the culprits. Sending emails and orchestrating the communications have taken a lot of resources from us. If security can identify the culprits, we will pass on the information to the Capstone teaching team to follow up with the appropriate actions. In the meantime, we have worked tirelessly to repair all the damages and reverify the features of our project. The fair majority of features are functional again, but not without a lot of time invested. We have moved the shelf to a temporary location, and then to the Pantheon 3DP office, where it will stay as its permanent location now (except for having it moved to Vari Hall for Final Capstone day only, where it will then be returned to the office). The office is a much more secure room, with very limited access (only Pantheon employees, we must inform them ahead of time if we plan to drop by). The tradeoff to security is accessibility, but considering the alternative, we believe this is our best option to safeguard our work. Moving our shelf to the Pantheon office has had the unintended bonus effect of getting real field testing and ensuring all our features are functioning in the final location. This benefit brought the team a little light in the darkness.

5.4 Technical Design Budget

The bulk of the technical design budget for Pantheon Material Management has been allocated to procuring the necessary computer hardware and physical storage to facilitate efficient smart inventory storage. This portion of the budget amounts to CAD 408.76 which is $76.95 less than the original

allocation. This divergence is largely due to the fact that the shelf had to be downsized due to the building's regulations as well as the use of free hosting services as opposed to Amazon Lightsail.

The reallocated savings were used to enhance system functionality, including the integration of weight sensors and a basic user interface. These additions improve real-time inventory tracking and usability, offering better value without exceeding the original budget—provided that purchases continue to be approved through the updated bill of materials (BOM) by supervisors.

Moreover, a portion of the budget has been allocated to 3D-printing services in the event that sensor housings are required. At the beginning of November 2024, the team predicted that $100 was going to be spent on 3D-printing at Pantheon. However, due to some additional housing requests from the CEO, an extra $49.01 had to be allocated, which equates to a total of $149.01.

Therefore, the total technical budget amounts after all purchases and changes equate to **CAD 557.77**. The project team's bill of materials provides a detailed description of each budget line and a comparison of similar products.

6. Preliminary Business Case

Our solution follows the guidelines and demands of our stakeholders by designing a shelf to contain the 3D printing materials while creating a web-based application to maintain and administer the materials. While the general product has several weaknesses, its core features, coupled with its scalability, heavily outweigh its limitations. The following is a SWOT analysis of our project, which will list the system's strengths, weaknesses, opportunities, and threats (Figure 39).

Figure 39: An in-depth analysis of our as-built system

6.1 Strengths

The system provides an effective solution for storing and organizing 3D-printing materials. It allows users to store materials in a structured manner and keep track of their inventory. Additionally, it integrates temperature and humidity sensors to monitor and regulate environmental conditions, which helps preserve the quality of materials. The web application supports key features such as adding, deleting, and editing materials, material types and user accounts, making it flexible and user-friendly. Lastly, the use of cloud technologies aids in the real-time communication between the hardware and software components, providing a continuous stream of information to the user.

6.2 Weaknesses

One limitation of the current system is the basic environmental control logic, which does not differentiate between the specific requirements of various filament types. While the logic keeps all additive manufacturing materials from degrading, the baseline thresholding may prove to be overly restrictive should the company change the types of materials within their inventory. The system also relies on internet connectivity, which could hinder functionality in offline environments. Furthermore, mass updates to filament inventory still require some degree of manual input, which may lead to errors or inconsistencies over time.

6.3 Opportunities

There is strong potential to enhance the system with additional features. One opportunity is to incorporate an analytics dashboard that visualizes data trends and usage statistics, providing users with more insight into their material management habits. Another opportunity lies in implementing AI-driven recommendations that respond to sensor readings and environmental changes, helping users take proactive actions to protect their materials.

The overall scalability of the system increases the potential of feasible growth. This scalability is specifically present within the existing lightweight MQTT protocol for IoT communications as its topic specificity and low bandwidth requirement allows for multiple devices to easily be connected and communicate in real time. Furthermore, the use of no-cost cloud technologies coupled with the implementation of a low memory intensity application further facilitates growth by providing computational resources for the remote deployment of future features.

6.4 Threats

Several risks may affect the reliability of the system. The dependance on a third-party broker for connection between the shelf hardware and the web application could prevent real-time updates or alerts. Users might also forget or neglect to update the system after using materials, which can reduce the accuracy of the inventory. In addition, over time, sensor performance may degrade or fail, leading to incorrect environmental readings.

7. Deviations from Plan

A significant deviation from our original project plan was the cancellation of the planned integration with the company’s planned order management application. Initially, our system was designed to connect with a separate capstone group’s order app. This would have allowed our web application to automatically update material inventory whenever a customer’s print order was executed.

However, due to delays in the completion of the order app by the partner group, the integration could not be implemented. As a result, we had to redefine our automation strategy and seek alternative solutions for updating material data. To address this, we developed and implemented a QR-based scanning system, allowing users to scan a QR code on a filament spool using their phone and then commit the change in available inventory through the web app. While this method isn't fully automated, it provides a fast, intuitive, and mobile-friendly way to keep material data up to date without relying on the external system. This pivot ensured that our system remained operational, user-friendly, and adaptable, even in the absence of third-party integrations.

8. Failure Report

Project failure, in the context of our capstone, would have meant delivering a system that was either non-functional, failed to meet core objectives, or did not satisfy stakeholder needs. A major risk we encountered was our initial dependency on integrating with an external order management application being developed by another capstone team. This integration was a key part of our original plan, intended to automate the updating of filament mass and stock levels within our system. However, due to delays outside of our control, the external group was unable to complete their application in time for integration. This incomplete task effectively meant the project's application integration feature would have entirely failed.

To mitigate the significant gap in our system's functionality, we re-scoped that portion of the project and implemented an alternative solution, the aforementioned QR-based scanning feature. While this method does not provide the full automation we originally planned, it ensures the process remains quick, intuitive, and user-friendly.

By adapting to this change, we ensured the core functionality of our system, including real-time inventory management, environmental monitoring, and ease of use, remained intact. While the project could not achieve inter-application integration, our ability to pivot and implement a viable alternative allowed us to avoid failure, maintain stakeholder satisfaction, and deliver a complete and functional end-to-end system.

9. Team Performance

9.1 Overview and Assignment of Roles

Below is a table that describes our team structure (Table 10). No new changes to the team structure this term; roles have been defined and each member in the team has delved further into their roles, becoming more experienced and independent in taking on tasks assigned. To review ITP team dynamics from Progress Report II, please refer to Appendix F.2.

Table 10: Current Role Designations

Role / Responsibility	Member(s)
Team Leader	Luca, Luqmaan
Project Communications	Luca, Luqmaan
Agile SCRUM Master	Paras, Tharuveen, Ethan
Architect	Luca (Software), Paras (Frontend), Luqmaan (Microcontroller), Ethan (Mechanical)
Software Development Team	Luqmaan, Tharuveen, Luca, Paras
Mechanical Design Team	Ethan, Jean
Documentation	Luca (Jira, Confluence), Luqmaan (GitHub, Discord)
Deliverables	Jean, Paras

9.2 Individual Member Breakdowns

9.2.1.2 ITP Metrics for Luqmaan Irshad

To view reviews 1 and 2, please refer to Appendix F.1.1.

Review 3: Unlike the previous reviews, this period of time was especially challenging. Due to external factors and a heavier course load, I was not able to commit to the project as strongly as I did in the previous half of the project. Despite this, however, my team was still considerate enough to give me similar scores, hovering in the same ballpark as what I had scored previously. I have received my lowest score this review, with a 4.2 in Focus, but every other category remains in the 4.6+, fortunately. To improve individually, I aim to take more initiative on tasks to let people know when I need more help ahead of time. From a team perspective, I aim to become more active in making announcements for our team, like I was able to in the previous two reviews, to keep the team on track. Overall, I agree with how my team has rated me, but I am not satisfied with my performance this quarter. I aim to devote more time and put more effort into wrapping up this project to ensure I can continue to deliver the high standards I have set for myself since the beginning of the project.

Final Review: Overall, I am really proud of both our team and the growth I have had over this journey. From my last review, I have followed through with what I outlined to improve and gone beyond. I took initiative on many different tasks, fully focusing on the microcontroller bringup, leading the team by highlighting deadlines with announcements and keeping track of them, as well as being one of the main sources of communication between our team and external stakeholders in the project. I pushed the team to be creative by leading the project video we recorded for our final peer presentations. I have been taking initiative on recording meeting notes for our weekly SCRUMs, in charge of software deployment to our frontend and merging software features into our main branch on

GitHub. When we were afflicted with the damage incident to our shelf, I took sole initiative on facilitating all communication, applying pressure to ensure the process moved forward and doing everything I could to ensure this incident didn't set us back too far. Looking back at my growth between the first review to now, and even the previous review to now, I notice immense growth and improvement in my leadership, technical skills and commitment to the team. I am confident that if we were to have another ITP review, this would be my strongest, and more importantly, I feel this time that I am satisfied with my performance, despite the heavy course load and external circumstances that tried to limit me from doing my best.

I am proud of our team and seeing how far we have come in the project. From our humble beginnings, we have risen as potentially the strongest Capstone team this year. We are rigorous in testing and quality, meet deadlines on time and have completed all the required features of our project ahead of time, allowing us to implement bonus features based on our schedule. I have seen a lot of growth with each member of our team, and wholeheartedly thank them for allowing me the honour of being part of the team.

There's not much I would change about how we ran our team if we were to do things differently. As mentioned, we successfully finished the project ahead of schedule and delivered a quality product. Our meticulous breakdown of roles helped define each member's responsibilities on the team. The flexibility to role assignment we allowed each term allowed each member to find an area they could succeed in, subverting issues like burnout, and accounting for availability and technical experience much earlier. One small improvement I could suggest for each phase: as manager of our Discord server (used to facilitate remote communication between team members) would be to create a forum channel for weekly updates. Forum channels in Discord allow you to create a post with a heading, where other members can chime in to discuss based on the topic. Creating a forum for each week in the SCRUM would be nice to have, as members could post their weekly updates in text. We could increase the frequency of these updates to twice a week, allowing us to ensure blocking issues are caught faster than waiting until the weekly SCRUM meeting to discuss them. This change would propagate through each phase of the project, allowing members to take more initiative to address issues sooner. As for specific phase changes, here are a few key points we could have:

Phase 1 - Inquiring more about Pantheon3DPs software suite: We only noticed by phase 4 that Panthoen3DP was not using the order application we were planning on integrating with. Ensuring that this application was in use should have been one of the constraints we looked at before moving further in the process.

Phase 2 - Go into more detail on 2-way communication planning. Looking back, our planning on this concept was a bit weak, and required a lot more consideration. Device type, wiring, communication methods and even the quality of the sensors we plan to use.

Phase 3 - Prepare the backend listener and frontend trigger ahead of time. This feature required a lot more time to mature, and took us almost the entire term to solve all the bugs. The other option would have been to remove this feature from the MVP if the scope seemed too large.

Phase 4 - We had a little more room in the budget to play with, so it would have been nice to buy wires that better fit our project. The wires on the microcontroller were all taken from our old projects. Some were too small, some too long, and a lot of them were inconsistent with their connections. Soldering more wires would have been good, and maybe having a special prototype phase for this build would have helped us iron out any bugs, so we could have confidence to solder every part.

Phase 5 - It would be nice to have gathered focus group feedback from the future users of the application. Although we did collect and implement feedback directly provided from Pantheon3DP's CEO, Nick DiScipio (one of the future users), we missed out on valuable feedback and different perspectives by overlooking the rest of the employees in our user demo and testing presentation to Pantheon3DP.

Phase 6 - Going a step further than just documented code, I think it would have been nice to make time to create a user manual. This way new users can be onboarded quickly and refer to the manual to understand error codes better and how to fix any potential quirks that arise.

9.2.2 Luca Filippelli

9.2.2.2 ITP Metrics for Luca Filippelli

To view reviews 1 and 2, please refer to Appendix F.1.2.

Review 3: As usual, I have received an overall very positive review from my team, with similar top scores across each category. They've noticed the extra effort I've taken to ensure the project is running smoothly and have mentioned that without the leadership I have provided, our project would be behind schedule. Currently, it seems my weakest area is with focus, where I received a score of 4.6. To further drive self-improvement for the benefit of our team, I aim to continue to keep the team updated as one of the main project leaders and software architects. I am humbled by the comments provided by my team and even more motivated to work hard to ensure our project remains at its best!

Final Review: The team has significantly developed their field-specific skills throughout this project. Achieving exceeding metrics, while fulfilling every requirement despite the challenges throughout this project, is a major success. As in previous reviews, I have actively sought to efficiently finish my tasks, thus ensuring they do not act as blockers to future work. Once completed, I have sought to help my teammates wherever possible. This aid has helped the team grow, particularly through pair programming, the process by which multiple developers work on a problem together.

On the project management front, I have aided in ensuring that our project remains on schedule by chairing our stand-up meetings while staying in frequent contact with each team member in between meetings. I have also facilitated communication between the team and the project sponsor, ensuring that the team's questions can evolve into a meaningful conversation on the direction of the project. While I believe that the order application pivot may have been done more gracefully through initial consultation with its development team, the manner in which we provided similar functionality in the face of such an inhibitor is nothing short of spectacular. All in all, this project has made significant progress in harmonizing additive manufacturing material storage and inventory management. This team should be proud of their work over these several months. My opinion on the team's performance in each phase is as follows:

Phase 1 - The problem definition and requirement gathering phase was run rather methodically. The team did background research on the general problem and areas of improvement. The team then worked with the project supervisor to develop a requirements list which was verified by the project sponsor. Seeing as the requirements list is nearly unchanged since this page, it is safe to assume that the team did well.

Phase 2 - The team evaluated several solutions (as seen in the mid-term report) and began to form subteams to better handle each solution component. The team strongly benefited from feedback sessions with the project supervisor and sponsor to better understand the feasibility of our intended solution direction. If we were to redo this project, I would have liked to interview Pantheon's order application team to better understand their progress from the beginning of the project.

Phase 3 - While the design of the Minimum Viable Product went well, I would, in the future, have not initially included the material warning system in this release as the team required more time to understand it's integration in detail. While the underlying listener was moved to a dedicated sprint, limiting the Minimum Viable Product's scope from the start would have made the process less stressful.

Phase 4 - The Alpha Release significantly expanded upon the Minimal Viable Product's functionality and acted as a step forward in meeting the project's requirements. This phase allowed me to get a stronger understanding of continuous integration by establishing an end-to-end testing pipeline. While this may have been the phase to meaningfully evaluate the feasibility of custom infrastructure such as an in-house MQTT broker, such components would be difficult for an additive manufacturing company to maintain. Therefore, I am generally content with the way this stage was conducted.

Phase 5 - The Beta Release went somewhat slower than I expected. This seemed to be largely due to the team's conflicting commitments with other coursework. In a rerun of the project, I believe changing the duration of our sprints during high-intensity periods within a semester would allow the team to effectively complete their tasks with somewhat more leeway in terms of time.

Phase 6 - The team made significant progress in making the product more marketable. For instance, a dramatized video was produced to provide more audience engagement than a simple voice-over PowerPoint presentation. The team also recovered well from the act of vandalism which occurred, ensuring that the product was functional once more. Despite the interface being incredibly simple to use, developing a user onboarding post or manual may have been beneficial in making our documentation friendly to an employee who, rather than reading several pages of a report, wants a simple explanation on accessing certain features.

9.2.3 Tharuveen Raveendran

9.2.3.2 ITP Metrics for Tharuveen Raveendran

To view reviews 1 and 2, please refer to Appendix F.1.3.

Review 3: Since the last review, I have seen improvements in some aspects, and a couple setbacks. Overall, my performance this quarter was strong, as noted by the team. They mentioned my ability to be flexible helped fill in the gaps of our project that needed to be covered. I handled some of the smaller tasks that may not be visible at eye level, and also took on larger roles like working on the user profile systems. This is why I have been able to maintain my commitment (4.6) and standards (4.8) scores. Surprisingly, my focus has dropped (from 4.8 back to 4.2). I believe this is a result of my work style. I work well in teams, but filling the gaps can sometimes cause me to work independently. Due to this, I could have been more up to date with the latest features the group was pushing, but my focus was on the corner cases. To improve, I aim to help participate in recording meeting notes for discussions, so I can get a more holistic approach to the team's progress every week. I also aim to take on responsibility for bigger features that allow me more opportunities to collaborate with my group members, so I can stay in the know and help keep them on track.

Final Review: As we approach the end of the project, I can't help but appreciate the efforts of my team members. We have all worked so hard to get to this point. I feel like we did a lot of things correctly, like setting up an automated CI pipeline for testing, which helped catch bugs in code before deployment. We had good collaboration and all completed the project in a timely manner. If we were to change something from our process and redo the project, my suggestions would be:

Phase 1 - One area I feel we could have improved on for phase 1 was to do more research on Pantheon3DP. Understanding what existing software they have, the office space they have and what resources they could provide us may have helped us be a little more concise for the kinds of constraints we placed.

Phase 2 - When designing our solution, we should have looked into the types of hosting (backend, frontend, database) with more granularity. In phase 3, we started to look at different types of hosting services, and realized how many fees there were with our initial choices.

Phase 3 - Have a better way to plan out time to collaborate. Once we got to phase 3, we were deep into the fall semester, and it was difficult to find common time. We could try using a visual schedule planner like when2meet, which is a free online tool that shows everyone's overlapping availability [29].

Phase 4 - With more time, it would have been great to set up our own MQTT broker. The current broker we use can be unreliable. The free use comes at the cost of potential instability, as Eclipse (the owner of the server) runs experimental tests on it that can cause downtime. For most of the time, we have not had issues with the broker, but it's annoying when it pops up. Setting up our own broker would be a big time investment, and we would need to consider how to host it (maybe as a process in the backend), but with enough planning, I think this would benefit us in the long run.

Phase 5 - We received feedback from our supervisor to add a feature to our web application after we had completed our final sprint, as we wanted to collect feedback on the finished product. It would have been good to get his feedback before the final sprint so we could have time to add his request.

Phase 6 - From a PM side, we did a good job with sprint reviews at the beginning of each sprint. Looking back, it would have been beneficial to add a review to the end and middle of each sprint, to assess the story point distributions for each member. This way we could more evenly distribute tasks, and adjust story point weights accordingly. To be fair, the current breakdown is as-is due to time constraints (some members had less time due to having heavier course loads), but we may have been able to smooth out the curve a little better.

9.2.4 Paras Kumar

9.2.4.2 ITP Metrics for Paras Kumar

To view reviews 1 and 2, please refer to Appendix F.1.4.

Review 3: This review, I have received my strongest performance scores. I may have underestimated my performance this term due to the increased workload of my courses, job and external life factors. Despite this, it seems the team recognizes the effort I have put in to manage and guide the frontend development of our server. Being the lead on this task has helped me maintain my near perfect scores for commitment and standards (4.8) while averaging 4.6 over the rest of the categories. With my skills in frontend development becoming stronger, I look to only improve my capabilities (4.6) to further drive the development and quality of our frontend. To make improvements for the team and

my teamwork, I aim to be more communicative of my schedule, by letting others know in advance if I have work or classes scheduled so they can plan meetings accordingly.

Final Review: Overall, I believe we make a really strong team. As the frontend architect, integrating our backend and database went smoothly, by making use of a few niche workarounds to allow us to contribute to the software while allowing us to make use of free hosting tiers. I'm really pleased with how the project all came together, and am content with the deliverable we are giving Pantheon 3DP, as it meets all the requirements we set out from the beginning of the project. If I were to do things differently:

Phase 1 - One area we could have ideated better was the system's longevity. Our system should be good for the foreseeable future, but there are concerns if we consider much farther. Currently our Azure is eligible for the free tier since we are students, and since Pantheon 3DP is a student-led company, it too is eligible once we hand over full control and hosting of our application to them. But in the case that Pantheon decides to become a non-student-led company, costs will be incurred. Looking back, we could have opted to set up a localhost server to subvert costs, but Pantheon does not have a technician who handles DevOps, so it would not be maintained. We could also consider running background tasks of our application separately, so there isn't a dependency on the uptime of our main application. This may have other side effects, like making the configuration more complicated for Pantheon by requiring them to manage backend containers, instead of just monitoring the main applications' uptime. By designing with a greater emphasis on future scalability, we may have given these alternatives a second thought.

Phase 2 - For solution design, we should have looked at the options Pantheon had available. We didn't realize until later that Pantheon has an abundance of Microsoft Azure credits that we could use. This led us to looking at cheap cost solutions, and we didn't reconsider Azure as a solution until later when we realized this and figured out our solution would be eligible for the B1 tier.

Phase 3 - Our MVP was a successful launch overall, but I think one final feature that would have been good to include is the user profile. An MVP should represent the basic functionality of a product, and I believe users and the user hierarchy are a core aspect of our design. We didn't have much time to include many features for the MVP since we were all getting started, but if I were to reorder some of the story points, this would have been a feature I would have liked to have included for the MVP, instead of waiting until Beta Release.

Phase 4 - To make this process smoother, it would have been good to align better with Pantheon on their software. We had an issue when trying to convert the 3D models that the mechanical team had designed on their computer, since the file type they had was incompatible with their software, and upon conversion, caused the models to come out differently than they had designed them. Our biggest problem came from integration with the order application; although they were supposed to have one, the one they promised wasn't deployed. Further aligning with Pantheon on their software structure ahead of time would have helped us avoid these pitfalls that caused us delay.

Phase 5 - Overall our test suite was very robust, but some of the frontend Selenium tests were flaky. This caused us to rerun and double-check components that were functioning, just failing tests due to those tests being inconsistent at times. Refining our approval structure to run the CI pipeline several times may help better detect flaky tests and maximize development time.

Phase 6 - From a final product perspective, the application is very complete based on the requirements we laid out. However, the future scalability of our application is limited due to the cost-effectiveness of our application. Although the current resources being used to host our application are sufficient,

the heavy restrictions on features like no IPv6 support means that the room for scalability is limited. To alleviate this, we could have created a locally hosted application, which would allow us to scale our own server physically and remove most costs associated with hosting.

9.2.5 Ethan Tran

9.2.5.2 ITP Metrics for Ethan Tran

To view reviews 1 and 2, please refer to Appendix F.1.5.

Review 3: My performance in this review was satisfactory overall. I seem to have dropped in a few areas, but overall my scores are not too far off from the last review. My biggest strength this term has been my capabilities and standards, as highlighted by my group (4.6 and 4.8 respectively). The team mentioned my leadership on the BOM and mechanical side of the project were my strong points, which align with the high ratings I received for my capabilities and standards. That being said, I can definitely improve on my commitment. I have a lot of work to complete this semester, and because of this I have shifted focus off the project sometimes. This is why my focus and communication were some of my lowest scores (3.6, 3.8). To improve individually, I plan to devote more time to the project by scheduling hours each day to tackle my tasks. To improve within the team, I plan to take the initiative in communication, by reaching out first instead of waiting for a roadblock before being contacted by another group member. This will help reduce bottlenecks and stress on the team, since I can let them know of issues ahead of time, allowing us to plan around problems more effectively.

Final Review: I think we did a good job as a team. I'm happy with the progress we made and the final deliverable is fully operational. As the architect for our Mechanical Design team, I think we did a fair job. We essentially focused on the most important aspects of the design to include, and built off from there. I think with more time, we could have added more features and refined some aspects of the design.

Phase 1 - I think we could have gone into more detail on the technical constraints for the shelf. What environment would the shelf be subject to, and how drastically does the temperature and humidity change? These would have been good questions to ask at phase 1.

Phase 2 - As manager of our BOM, it would have been nice to lock in all the required components before the first BOM submission date. Due to changes in constraints and requirements, this wasn't exactly possible in all circumstances. In cases it was, however, it would have reduced the number of BOM updates we flooded the teaching team with.

Phase 3 - It would have been nice to have a constructed shelf instead of just a 3D model of it for our MVP. The bottleneck in this case was the BOM submission date, which was after the MVP due date. We couldn't afford to eat the cost ourselves, as the shelf costs around $400 on its own, which would have been a very expensive mistake in the case it wasn't approved. Perhaps we could have got a special request to advance the request so we could prepare it for the MVP.

Phase 4 - Making use of version control software would have been great, so we could enhance our iterative design process. The software team uses GitHub to version control their software, but for 3D modelling, it's not a great option. Looking into software designed for 3D modelling version control would allow us to design prototypes faster and revert to previous versions of our design if we decided we wanted to go back to a specific version.

Phase 5 - Creating a way to test our prints to ensure they could withstand some impact would be a nice bonus. The space within the office is tight, and it could be easy to accidentally hit the housings (which are designed to protect the electronics from these impacts). Printing out prototypes and designing tests around these kinds of simulations would be what I would do for next time.

Phase 6 - Again, the use of a version control software here would be really helpful. Allowing us to document the history of changes and versions that upgrade/modify the 3D models for improved mechanical use. It would also allow us to keep a repository of alternative prints, in the case the Pantheon team decides they want to use a different desiccant model for example that is curved to fit around spools better instead of the flat ones that attach to the walls of the shelf (both of which we designed, but we only use the flat ones).

9.2.6 Jean Granato

9.2.6.2 ITP Metrics for Jean Granato

To view reviews 1 and 2, please refer to Appendix F.1.6.

Review 3: This review received strong scores across all categories, with my highest rating in standards (5.0) and solid performance in commitment (4.0), communication (3.6), capabilities (4.4), and focus (3.8). One key takeaway for me is that while my team acknowledges my contributions, there is room to improve my communication to ensure better alignment with the group. I gave myself a higher rating than my peers in this category, which implies that I might need to be more deliberate about routinely asking for feedback and sharing changes. While I try to keep my team informed, I can be more proactive in engaging in discussions and clarifying expectations so that everyone is on the same page. To improve, I plan to focus on keeping my team informed about my progress and potential challenges earlier, so we can work more effectively together. As for team challenges, overall, we have been working well, but there are some hurdles in coordination and adaptability. Unexpected changes in workload might occasionally make it challenging to guarantee communication and assistance when needed. At times, balancing individual responsibilities while also staying actively involved in group decisions can be a challenge, especially when unexpected obstacles come up outside of this course. While our structured planning has helped, we could benefit from better backup strategies and proactive problem-solving. I want to contribute by being more available to assist teammates when they face obstacles and encouraging open discussions about workload distribution. By improving in these areas, I believe we can strengthen our teamwork and complete our project more effectively.

Final Review: Overall we did a great job as a team! We pushed through challenges and obstacles that faced us at every stage of the project. The software team did a great job with the application as it looks very professional and complete. Our mechanical deliverables are also of high quality. I had a great time working with this team, and appreciate all the support they gave me when I was tight on time and deadlines. If I were to go back and handle situations in our project differently:

Phase 1 - As a member of the mechanical design team, we should have looked into constraints regarding legal shelf storage and types. We ran into an issue in Phase 2 where our shelf was not approved since it did not meet the safety requirements of the Bergeron Garage (the shelf was too tall to not be anchored to a wall).

Phase 2 - I think we should have looked into alternative 3D printing options for any of the housings we designed, at least for the prototyping phase. This would have allowed us to be more flexible in

iterative design, since Pantheon's price per print is very high (due to the high quality of their prints). Making use of alternative printing solutions would have helped us facilitate this design better.

Phase 3 - For our MVP, similar to Ethan, I would have liked to build the shelf in time for the MVP due date. However, due to the BOM deadline being after the MVP due date, we were not able to purchase the shelf in time. Asking for a special advanced request would have been the way to go.

Phase 4 - I think we could have leaned more into iterative design. Due to the cost and accessibility of 3D printing for our team, it was a little more difficult to print and iterate on our designs physically without incurring too much cost and material waste. That being said, I think we could have looked more aggressively into other alternatives that allowed us to print and test our 3D models physically before creating the final order with Pantheon for the official prints we use in our shelves.

Phase 5 - Looking back, we could have designed our shelf testing to be more rigorous. The current testing method is sufficient, but it would be good to test the shelf's resilience in extreme conditions (when RH is close to 100%, during a storm, leaving the garage door open). But this is heavily dependent on the weather, and for most of the winter semester (when the shelf was built), we did not experience any extreme humidity scenarios.

Phase 6 - I think it would have been good to add branding / logos to more of our design. On the side of the shelf, in visible areas of our prints (like on buttons or above the display) to make the designs feel more official. This would have helped the product feel more polished and final.

9.3 Team Self Evaluation

Table 11: The Team's Self-Evaluation of the Project's Current Progress

Criterion For This Deliverable	Self-Evaluation Ranking (Compared to Mid-Project Report)	Justification
BOM	Exceeding → Exceeding	With our careful planning, management and alternative selections, we have been able to remain very cost effective, using under $600 of the $1000 budget we have access to. For context, general practice in project management states that at least 10% of the budget should be set aside. With our careful planning, we were able to achieve over 40% of the budget in savings to be set aside.
Story Points Completed	Meeting → Exceeding	Based on our Jira SCRUM board, we were able to complete 100% of our story points overall. Last time we evaluated this criterion in the mid-project report, we were at 98%, with some bugs preventing us from reaching full coverage. We're pleased to report that we have solved these issues, and have completed all story points. There are no more tasks in the backlog, and we have reached AFE (all features enabled).
Mini Capstone Day Video and Presentation	Meeting → TBD	Back in the mid-project report, we recorded a success with our presentation. Some key takeaways on areas we could improve; formatting for the presentation style, turning the video into a skit rather than a straightforward PowerPoint presentation.

→ Final Capstone Day Presentation		Despite this, we still had a very strong showing and were able to answer all of the Panel's questions effectively. The format for the final capstone day presentation is much different, and the date for this is past the report's deadline, so the final result is TBD. However, we are working hard to prepare ahead of time so we can knock our final presentation out of the park; we have a layout for our table arrangement drawn, and are working on a poster board to add to our booth for the presentation (please refer to Appendix AF15 for more details). We are also trying to have a 3D printer live print items, to draw people into our presentation. We can see this metric on the trajectory to "Exceeds" if all goes well.
Solution Presentation → Beta Release Presentation	Exceeding → Exceeding	Our time to show off our skit-style presentation occurred during our beta release presentation. Similar to the last presentation, we were able to handle all the questions targeted to our team with a high degree of effectiveness. The video was a hit, and kept viewers engaged, allowing us to achieve our target we set from the previous video presentation.
Final Product - Web Application	Meeting → Exceeding	In the last report, our software application was only in its MVP stage. Since then, our application has blossomed, becoming more robust, including all the required features and some additional bonus features we had time for. We have received very positive feedback from our project supervisor and from Pantheon3DPs CEO, Nick DiScipio about the current state of this deliverable. The beta release was a huge success, and with AFE, we are ready to fully deploy version 1.0 of our application to Pantheon.
Final Product - Shelf	(New) Meeting	Back in the MVP stage, our shelf was only a simple 3D model. Throughout the design and testing phase, we have completed bringup of the shelf, with AFE. The shelf meets the requirements we laid out at the beginning, and we have received good feedback after demonstrating it to Pantheon. As a team, we all agree that the shelf has more potential, and to exceed, we could have added a more active dehumidification/temperature regulation solution using electronics.
Final Product - MAI	(New) Meeting	The device that facilitates our IoT communications and monitors our shelf in real time has seen a lot of growth. It started as a simple monitoring tool to keep our application informed on the interior climate of the shelf in real-time. However, due to the lack of an order application from Pantheon, we had to expand the featureset. MAI was fitted with a scale that helps users track inventory use by recording the mass of a material, allowing the application to determine how much material was used in a print. We have received great feedback from Pantheon's CEO on this feature, as it alleviates the need to figure out deployment for the order application. As a team, we recognize that to have this feature exceed, we would have liked to incorporate a logging feature that allows the backend to export the messages sent by MAI over a set period of time.

10. Conclusion

This project aims to generate a material management system for an additive manufacturing company to track, store and manipulate inventory. This solution differs from existing products as it connects inventory management with climate-controlled material storage. The final design requirements have been achieved through the development of an IoT-integrated shelf and a web application. This design has been verified through a repeated series of test cases. While the system is limited by its use of third-party applications, its use of robust components increases the ease of future scalability while ensuring the existing features can operate with minimal maintenance. This project was able to achieve it's design successes through the use of the Agile Methodology. As of April 21, 2025, the project will have completed the entirety of its planned sprints on schedule. While some minor adjustments to the project path could have been made, the team has determined that the project's effective design and management strategy have sufficiently contributed to its completion.

11. Acknowledgements

We would like to thank our friends, family, peers, professors and mentors who enabled us to be where we are today. A special thanks to Professor Hung Viet Pham for his constant support throughout the project, Professor Gerd Grau for his shrewd advice and Pantheon3DP for sponsoring the project. It takes a village to complete such a feat and we would surely be the lesser without you.

12. References

Refer to the following page

Works Cited

[1] "Principles behind the Agile Manifesto." Manifesto for Agile Software Development. https://agilemanifesto.org/principles.html [accessed Oct. 24th, 2024].

[2] O'Reilly. "Software Architecture Patterns." O'Reilly. https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html [accessed Nov. 27th, 2024].

[3] GeeksforGeeks. "Repository Design Pattern." GeeksforGeeks. https://www.geeksforgeeks.org/repository-design-pattern/ [accessed Dec. 1st, 2024].

[4] Amazon. "What is a RESTful API?." AWS. https://aws.amazon.com/what-is/restful-api/ [accessed Nov. 27th, 2024].

[5] "React Architecture Patterns in ReactJS Apps." Angular Minds. https://www.angularminds.com/blog/react-architecture-patterns-in-reactjs-apps [accessed Nov. 27th, 2024].

[6] "Systems and software engineering — Vocabulary." Michigan State University. https://www.cse.msu.edu/~cse435/Handouts/Standards/IEEE24765.pdf [accessed Nov. 30th, 2024].

[7] "The different types of software testing." Atlassian. https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing [accessed Nov. 30th, 2024].

[8] "What is the Internet of Things (IoT)?." IBM. https://www.ibm.com/topics/internet-of-things [accessed Nov. 28th, 2024].

[9] Amazon. "What is MQTT?." AWS. https://aws.amazon.com/what-is/mqtt/ [accessed Nov. 28th, 2024].

[10] "What is event-driven architecture?." IBM. https://www.ibm.com/topics/event-driven-architecture [accessed Nov. 28th, 2024].

[11] "RabbitMQ tutorial - Topics." RabbitMQ. https://www.rabbitmq.com/tutorials/tutorial-five-python [accessed Nov. 28th, 2024].

[12] Okta. "Hashing Passwords: One-Way Road to Security." Auth0 by Okta | Blog. https://auth0.com/blog/hashing-passwords-one-way-road-to-security/ [accessed Nov. 28th, 2024].

[13] C. G. Amza, A. Zapciu, F. Baciu, M. I. Vasile, A. I. Nicoara. "Accelerated Aging Effect on Mechanical Properties of Common 3D-Printing Polymers." MDPI. https://www.mdpi.com/2073-4360/13/23/4132 [accessed Nov. 28th, 2024].

[14] “What is the ‘hygroscopicity’ of a filament?.” Recreus. https://recreus.com/gb/noticias/learn-with-recreus/moisture-in-filaments [accessed Nov. 28th, 2024].

[15] “How to store material.” UltiMaker. https://support.makerbot.com/s/article/1667410781492 [accessed Nov. 28th, 2024].

[16] “Guide to Stereolithography (SLA) 3D Printing.” formlabs. https://formlabs.com/blog/ultimate-guide-to-stereolithography-sla-3d-printing/?srsltid=AfmBOoonOovaMwRrX9SsQe8GHom7aC-_dsxahKzBaSN3FOrusQHfKe6B [accessed Nov. 28th, 2024].

[17] “Resin and Storage Safety.” 3D Systems. https://3dsystems.my.site.com/help/s/article/figure-4-modular-resin-storage-and-safety?language=en_US [accessed Nov. 28th, 2024].

[18] “ABS-like Resin V2.0 Grey 1KG.” ELEGOO. https://ca.elegoo.com/collections/resin/products/elegoo-abs-like-resin-2-0-1000g [accessed Nov. 28th, 2024].

Author initials. Last name. “Page title.” Website Name. URL (accessed Month Day, Year).

[19] “How Does a Strain Gauge Work?” Load Cell Central. https://www.800loadcel.com/load-cell-and-strain-gauge-basics.html [accessed Apr. 14th, 2025].

[20] “Email API quickstart: How to send email with python,” Twilio, https://www.twilio.com/docs/sendgrid/for-developers/sending-email/quickstart-python (accessed Feb. 27, 2025).

[21] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading etc.: Addison-Wesley, 1994.

[22] “What is argon2?#,” argon2, https://argon2-cffi.readthedocs.io/en/stable/argon2.html (accessed Feb. 27, 2025).

[23] “Build in the cloud free with Azure for Students” Microsoft. https://azure.microsoft.com/en-us/free/students [accessed Apr. 13th, 2025].

[24] “Azure App Service on Linux pricing” Microsoft. https://azure.microsoft.com/en-us/pricing/details/app-service/linux/ [accessed Apr. 13th, 2025].

[25] K. Sheehy. “How Much Cell Phone Data Do You Really Need?” nerdwallet. https://www.nerdwallet.com/article/finance/how-much-data-do-you-need [accessed Apr.18th 2025].

[26] “Account Plans on Vercel” Vercel. https://vercel.com/docs/plans [accessed Apr. 15th, 2025].

[27] “Contiki-NG: The OS for Next Generation IoT Devices” GitHub. https://github.com/contiki-ng/contiki-ng [accessed Apr. 15th, 2025].

[28] “RIOT - The friendly Operating System for IoT!” GitHub. https://github.com/RIOT-OS/RIOT [accessed Apr. 15th, 2025].

[29] “New Event Name” When2meet. https://www.when2meet.com/ [accessed Apr. 19th, 2025].

Appendix

Appendix A: Additional Tables and Figures

<table>
<tr><td colspan="4"><u>Purpose for creating the product</u>
<u>How the project/product aligns with the team's purpose, interests, capabilities</u>

Automate the Pantheon 3DP 3D printing material management system, and integrate it with the company’s existing order placement platform, to optimize the company’s material waste while reducing cost surrounding material procurement</td></tr>
<tr><td><u>Target group</u>

For various target customers and/or users
for the product/project, who are they and
are they a subset of a larger societal Segment

Pantheon 3DP machine operators, logistics staff, small business 3D Printing businesses</td><td><u>Problem solved by the product</u>

What is the group's articulation of the the problem that the product/project will Address

To automate the management of 3D-printing material to protect the inventory from environmental degradation, track inventory and warn of material shortages.</td><td><u>Product description, uniqueness, Feasibility</u>

What will the product/project do to address that problem (not how), and how does this differ from what any existing product/projects do? What suggests it will be possible to do this new thing?

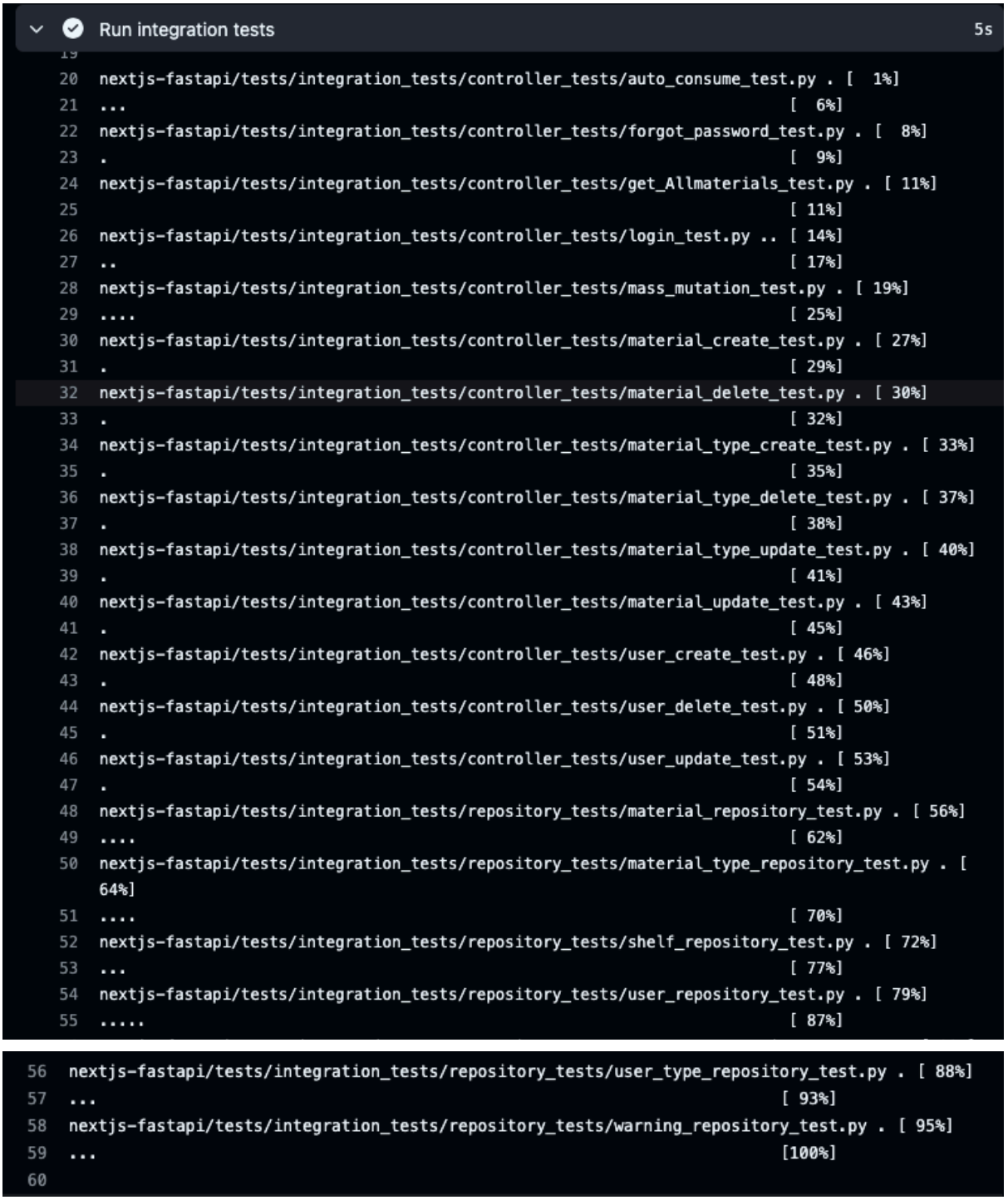
Our product will harmonize environment-controlled shelving with a material management application. This application will allow a user to view current material inventory while being alerted to unideal environmental storage conditions as well as low stock.</td><td><u>Product societal benefits</u>

What are the broader benefits of such a product/project (multi-bottom-line)

- Automated material tracking will ensure that the company is aware of its inventory in real time
- Low stock prompting will allow the company to know when to order more material to ensure they have sufficient stock
- Climate-controlled inventory storage will ensure that the inventory does not degrade.</td></tr>
</table>

<u>AT1: An Updated Product Vision Board</u>

```
13  ============================= test session starts ==============================
14  platform linux -- Python 3.9.21, pytest-8.3.5, pluggy-1.5.0
15  rootdir: /home/runner/work/eng-4000-capstone/eng-4000-capstone
16  plugins: postgresql-7.0.1, asyncio-0.26.0, anyio-4.9.0
17  asyncio: mode=strict, asyncio_default_fixture_loop_scope=None,
    asyncio_default_test_loop_scope=function
18  collected 39 items
19
20  nextjs-fastapi/tests/unit_tests/mailer/test_mailer.py ..........         [ 25%]
21  nextjs-fastapi/tests/unit_tests/model_u_tests/test_Material.py ........  [ 46%]
22  nextjs-fastapi/tests/unit_tests/model_u_tests/test_Material_Type.py .... [ 56%]
23                                                                           [ 56%]
24  nextjs-fastapi/tests/unit_tests/model_u_tests/test_Shelf.py .....        [ 69%]
25  nextjs-fastapi/tests/unit_tests/model_u_tests/test_User.py ...           [ 76%]
26  nextjs-fastapi/tests/unit_tests/model_u_tests/test_User_Type.py ...      [ 84%]
27  nextjs-fastapi/tests/unit_tests/password_hashing/password_hash_test.py . [ 87%]
28  ..                                                                       [ 92%]
29  nextjs-fastapi/tests/unit_tests/test_temp_password_randomizer.py ...     [100%]
30
31  ============================== warnings summary ===============================
32  nextjs-fastapi/db/model/base.py:3
33    /home/runner/work/eng-4000-capstone/eng-4000-capstone/nextjs-fastapi/db/model/base.py:3:
    MovedIn20Warning: The ``declarative_base()`` function is now available as
    sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at:
    https://sqlalche.me/e/b8d9)
34      Base = declarative_base()
35
36  nextjs-fastapi/tests/unit_tests/model_u_tests/test_Material_Type.py::test_set_name
37    /home/runner/work/eng-4000-capstone/eng-4000-capstone/nextjs-fastapi/tests/unit_tests/
    model_u_tests/test_Material_Type.py:64: SAWarning: Session's state has been changed on a non-
    active transaction - this state will be discarded.
38      session.rollback()
39
40  -- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
41  ======================== 39 passed, 2 warnings in 1.15s ========================
```

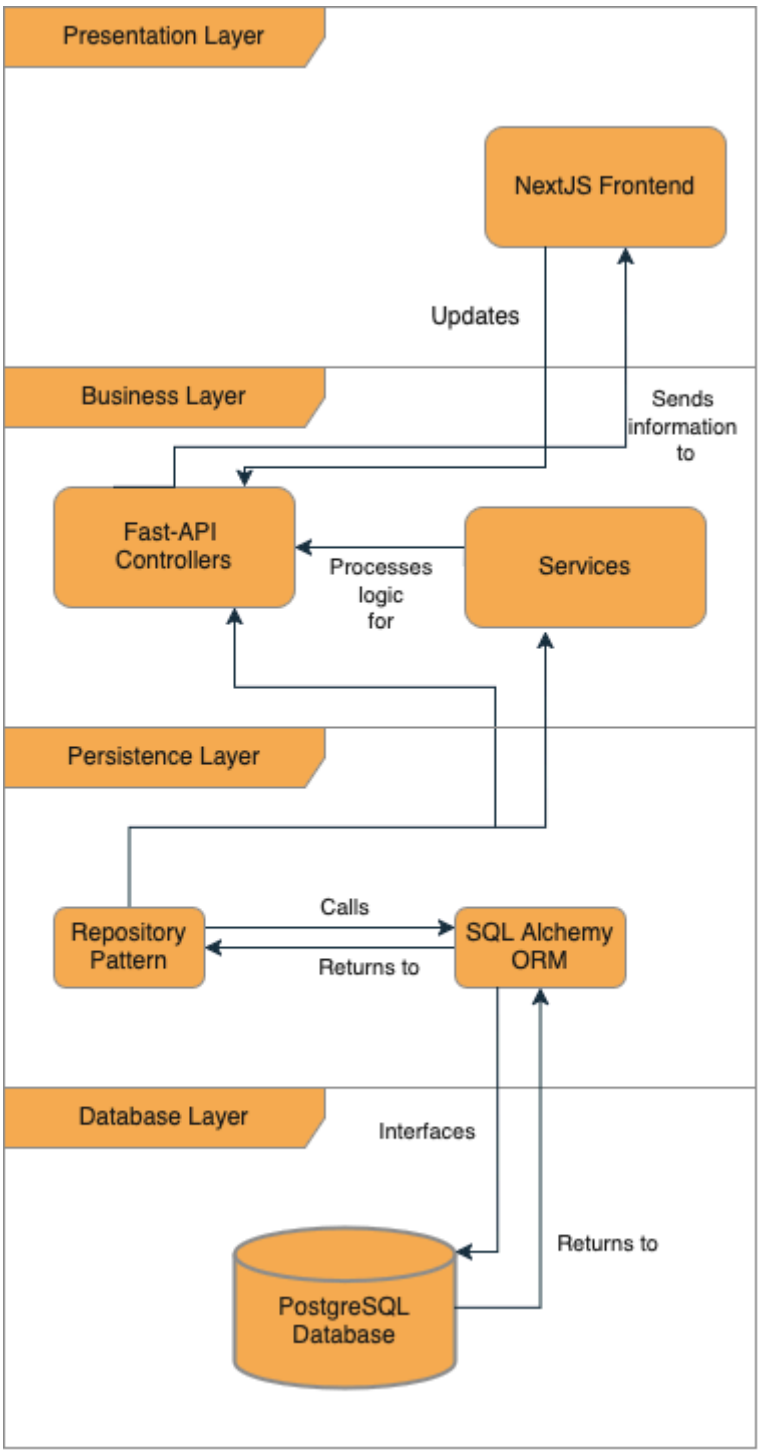

Run integration tests
5s
nextjs-fastapi/tests/integration_tests/controller_tests/auto_consume_test.py . [1%]
... [6%]
nextjs-fastapi/tests/integration_tests/controller_tests/forgot_password_test.py . [8%]
. [9%]
nextjs-fastapi/tests/integration_tests/controller_tests/get_Allmaterials_test.py . [11%]
[11%]
nextjs-fastapi/tests/integration_tests/controller_tests/login_test.py .. [14%]
.. [17%]
nextjs-fastapi/tests/integration_tests/controller_tests/mass_mutation_test.py . [19%]
.... [25%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_create_test.py . [27%]
. [29%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_delete_test.py . [30%]
. [32%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_type_create_test.py . [33%]
. [35%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_type_delete_test.py . [37%]
. [38%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_type_update_test.py . [40%]
. [41%]
nextjs-fastapi/tests/integration_tests/controller_tests/material_update_test.py . [43%]
. [45%]
nextjs-fastapi/tests/integration_tests/controller_tests/user_create_test.py . [46%]
. [48%]
nextjs-fastapi/tests/integration_tests/controller_tests/user_delete_test.py . [50%]
. [51%]
nextjs-fastapi/tests/integration_tests/controller_tests/user_update_test.py . [53%]
. [54%]
nextjs-fastapi/tests/integration_tests/repository_tests/material_repository_test.py . [56%]
.... [62%]
nextjs-fastapi/tests/integration_tests/repository_tests/material_type_repository_test.py . [64%]
.... [70%]
nextjs-fastapi/tests/integration_tests/repository_tests/shelf_repository_test.py . [72%]
... [77%]
nextjs-fastapi/tests/integration_tests/repository_tests/user_repository_test.py . [79%]
..... [87%]
nextjs-fastapi/tests/integration_tests/repository_tests/user_type_repository_test.py . [88%]
... [93%]
nextjs-fastapi/tests/integration_tests/repository_tests/warning_repository_test.py . [95%]
... [100%]

```
▸ Run set PYTHONPATH=nextjs-fastapi && pytest nextjs-fastapi/tests/feature_tests/
============================= test session starts ==============================
platform linux -- Python 3.9.21, pytest-7.2.2, pluggy-1.5.0
rootdir: /home/runner/work/eng-4000-capstone/eng-4000-capstone
plugins: asyncio-0.21.0, postgresql-7.0.1, anyio-4.9.0
asyncio: mode=strict
collected 40 items

nextjs-fastapi/tests/feature_tests/test_login_page.py ....               [ 10%]
nextjs-fastapi/tests/feature_tests/test_material_table.py .............  [ 42%]
nextjs-fastapi/tests/feature_tests/test_material_type_table.py .......   [ 60%]
nextjs-fastapi/tests/feature_tests/test_nav_bar.py .                     [ 62%]
nextjs-fastapi/tests/feature_tests/test_notif_panel.py .                 [ 65%]
nextjs-fastapi/tests/feature_tests/test_profile_page.py ..               [ 70%]
nextjs-fastapi/tests/feature_tests/test_qr_endpoint.py ..                [ 75%]
nextjs-fastapi/tests/feature_tests/test_sale_scraper.py ...              [ 82%]
nextjs-fastapi/tests/feature_tests/test_user_access.py .                 [ 85%]
nextjs-fastapi/tests/feature_tests/test_user_table.py ......             [100%]

```

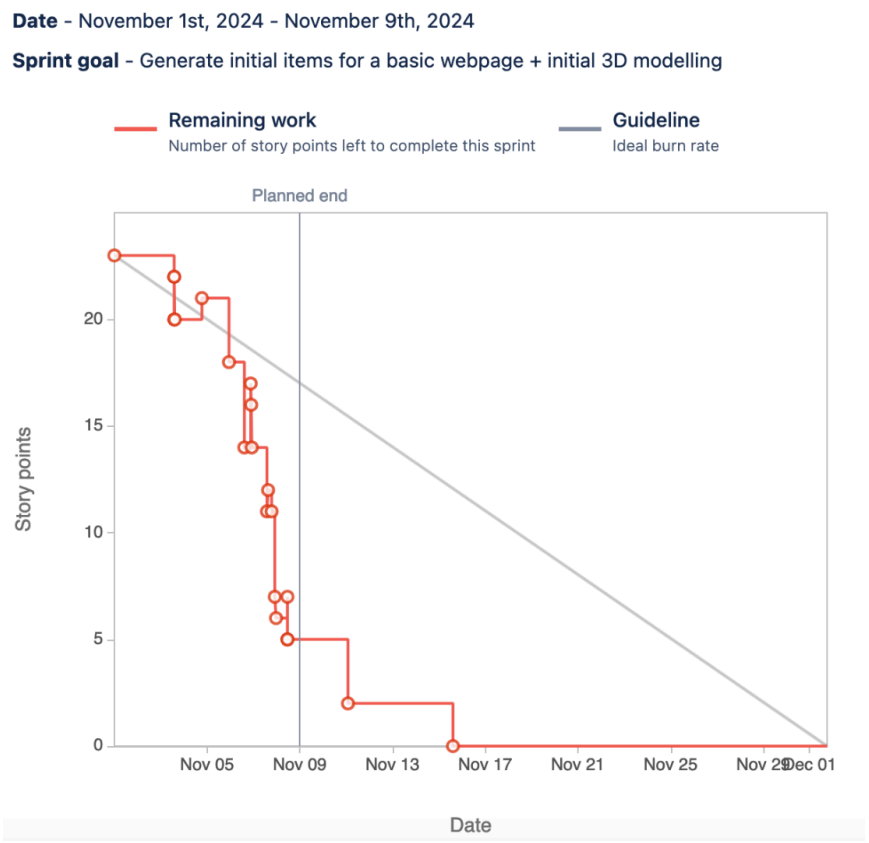
AF1: The outputs of Pantheon Material Management's test suite

NOW	NEXT	LATER
- Establish a database to represent the shelving units, warnings and materials - Design a CRUD-friendly front end to enable material visualization - Design a 3D model of the shelf frame and acquire the materials necessary using a BOM. - Design shelving so that it can accommodate the embedded climate tracking sensors. - Machine and build the frame itself based on engineering drawings - Develop a notification system for environmental thresholding and low inventory warnings	- Interconnect the shelving unit to the web application to receive real-time environmental data - Integrate a notification system with a mailer to email notifications - Expansion of the shelf frame - Have designated areas for specific materials based on dimensions and weight. - Improve the front end of the web application by enhancing the colours, visual design, and usability - Implement material searching for company inventory - Implement user hierarchy to facilitate access management - Implement pressure plates or any scale to read the weight of the spool and resin. - Develop a mobile page to automatically update a mass when a user interacts with the scale	- Implement predictive / web crawling tools to optimize material acquisition - Room for automated shelving - Package methods for extracting inventory information as files - Regular maintenance or replacement of the frame or features

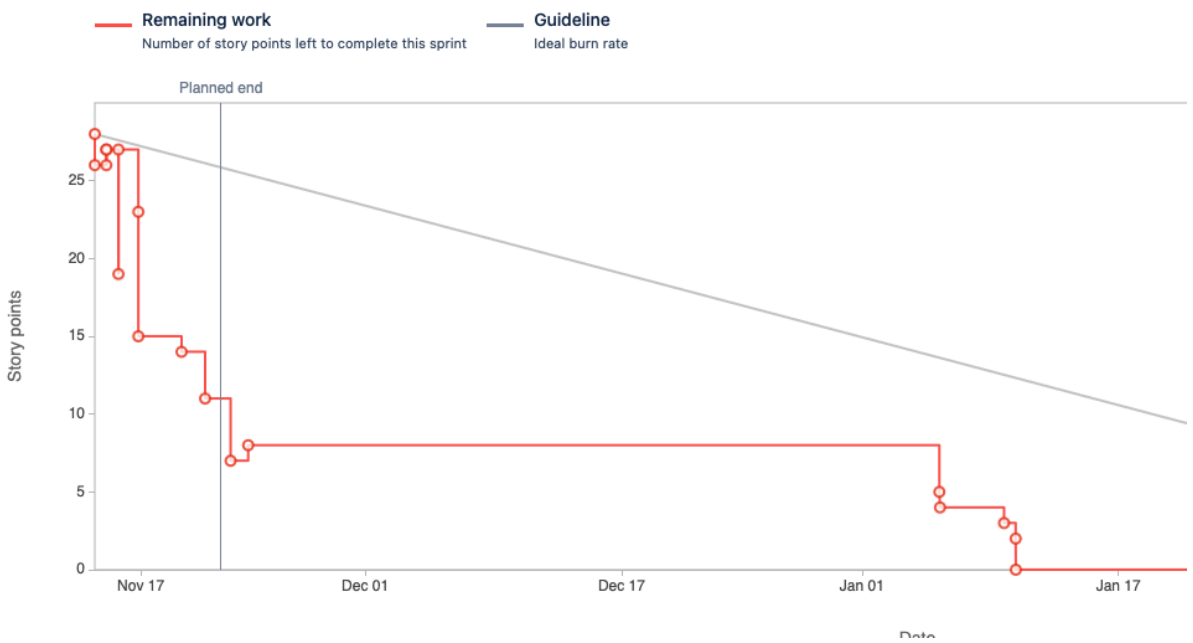
AT2: An Updated Now Next Later Table (the team has completed all three phases)



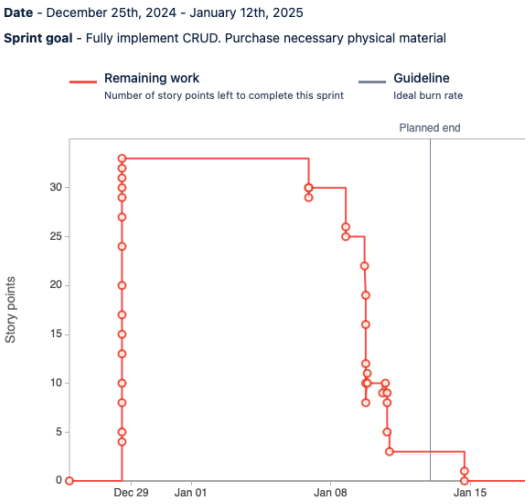
AF2: Pantheon Material Management's Layered Architecture diagram



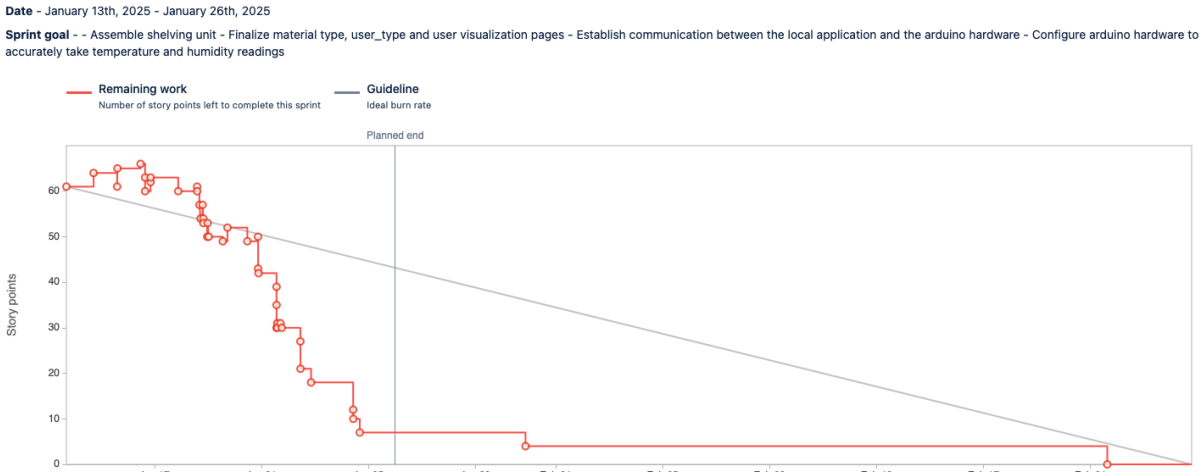
AF3: Sprint 1 Burndown Chart



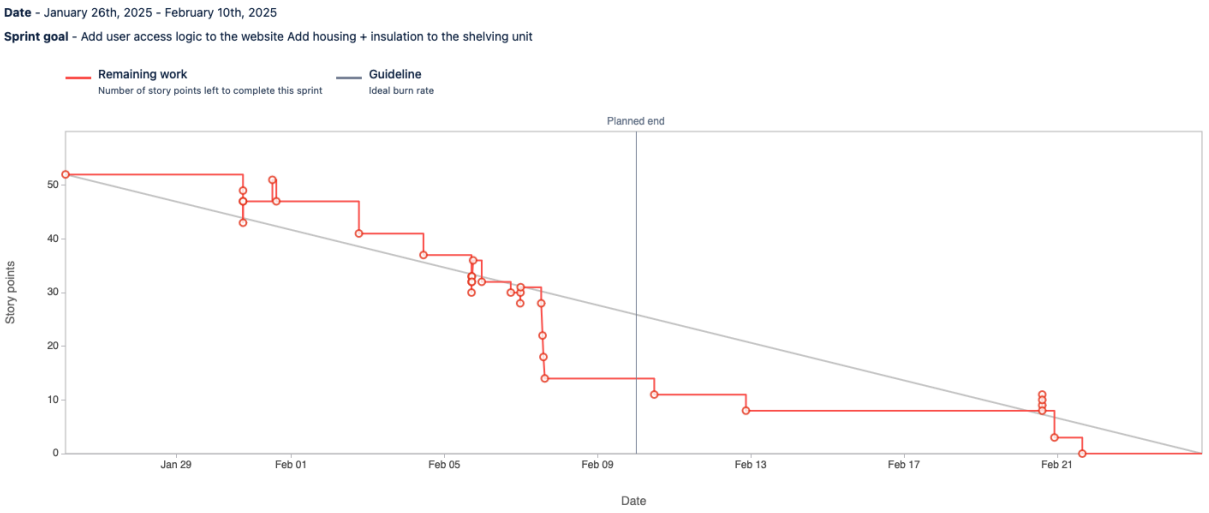
AF4: Sprint 2 Burndown Chart



AF5: Sprint 3 Burndown Chart



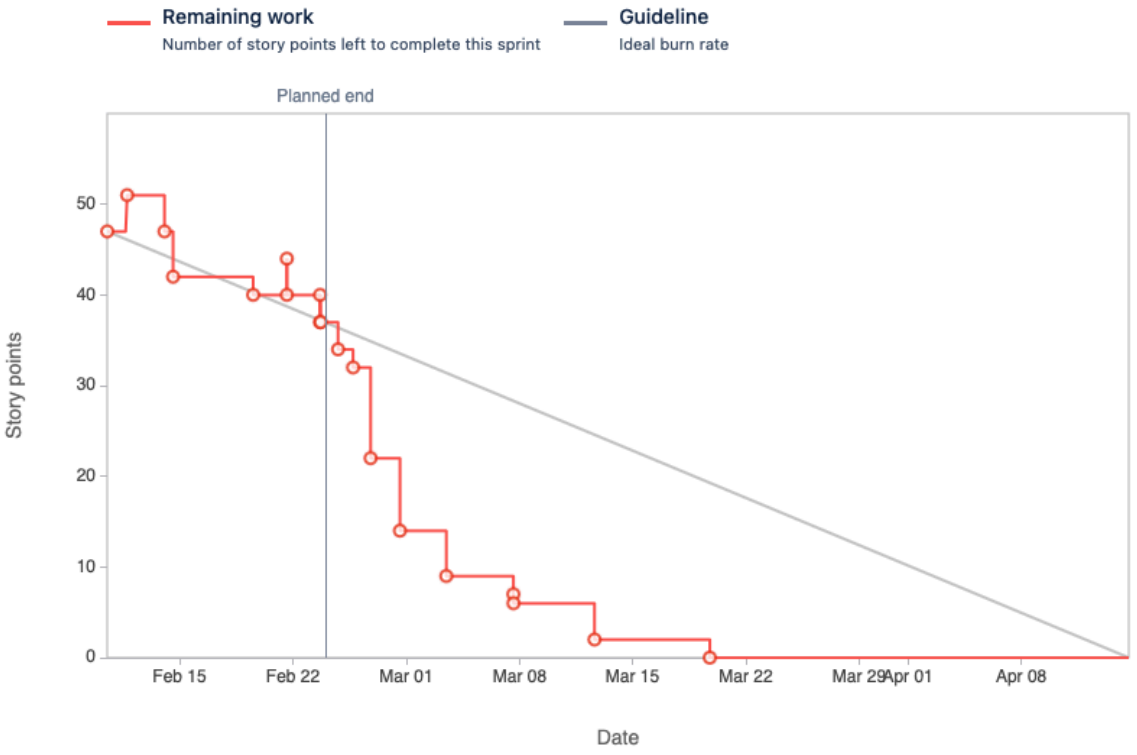
AF6: Sprint 4 Burndown Chart



AF7: Sprint 5 Burndown Chart

Date - February 10th, 2025 - February 24th, 2025

Sprint goal - - Insulate shelf and add circuitry - Deploy webapp - Write progress report - Edit application to client's requests



AF8: Sprint 6 Burndown Chart

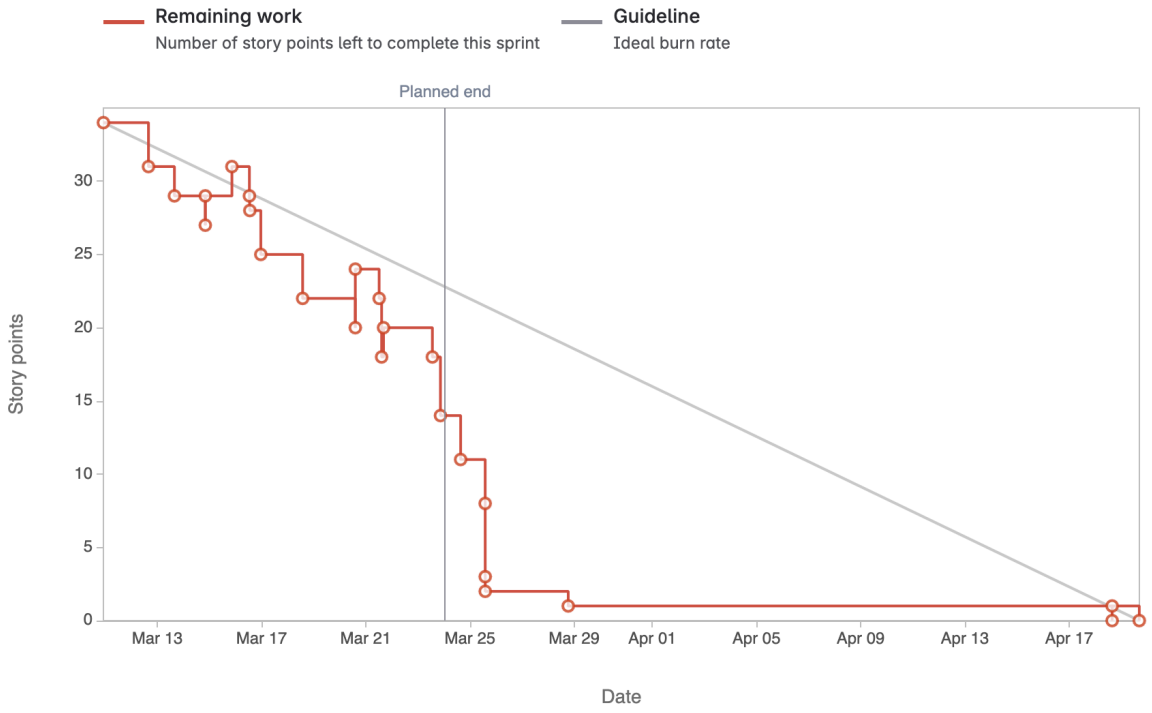
Date - February 24th, 2025 - March 10th, 2025

Sprint goal - - Finalizing backend methods for system warnings - Creating front-end interfaces for system warnings - Install load cells; determine method of associating spools efficiently - Install circuitry + filament holder - Software updates from alpha release feedback

AF9: Sprint 7 Burndown Chart

Date - March 10th, 2025 - March 24th, 2025

Sprint goal - - Backend / Frontend refactoring - Connect circuitry to shelving unit + verify - Add pantheon styling to website



AF10: Sprint 8 Burndown Chart

Date - March 24th, 2025 - April 21st, 2025

Sprint goal - - CAPSTONE DAY VIDEO - CAPSTONE DAY REPORT - Print out QR code links - Show the beta product to Nick for feedback - Minor software refactoring

AF11: Sprint 9 Burndown Chart


SCRUM Sprint 1 1 Nov – 9 Nov (9 issues)
0 0 22
Complete sprint
Generate initial items for a basic webpage + initial 3D modelling
SCRUM-1 Create DB DONE 2 LF
SCRUM-2 Visualize table of materials DONE 1 PK
SCRUM-3 Create DB wrapper class and populate with... DONE 3 LI
SCRUM-4 Create table query class DONE 3 PK
SCRUM-5 Create DB population script DONE 4 TR
SCRUM-6 Generate a rough estimate 3D models for t... DONE 2
SCRUM-11 Set up coding environment DONE 1 PK
SCRUM-7 Determine electrical components and incor... DONE 2 JG
SCRUM-12 Refactor DB references to go through an O... DONE 4 LF
Create issue
SCRUM Sprint 2 14 Nov – 22 Nov (12 issues)
1 7 22
Complete sprint
Implement BOM, complete mid term report, implement notification system.
SCRUM-13 Generate Bill of Materials DONE 8
SCRUM-9 Implement unit testing for the model classes IN PROGRESS... 1 LI
SCRUM-8 Implement feature testing for the basic tabl... IN PROGRESS... 3 LF
SCRUM-10 Add integration tests for the table controll... IN PROGRESS... 1 TR
SCRUM-19 Generate material event listener for mass ... IN PROGRESS... 2 LI
SCRUM-20 Add front end popup to react to a material... DONE 2 PK
SCRUM-21 Implement repository pattern + test DONE 4 LF
SCRUM-23 Add mass setter method in model DONE 1 LI
SCRUM-24 Generate controller method to update the ... DONE 3 LF
SCRUM-22 Add front end capabilities to edit the mass... DONE 4 PK
SCRUM-25 Midterm Report IN PROGRESS... - TR

SCRUM Sprint 3 25 Dec – 12 Jan (14 issues) — 0 0 29 — Complete sprint

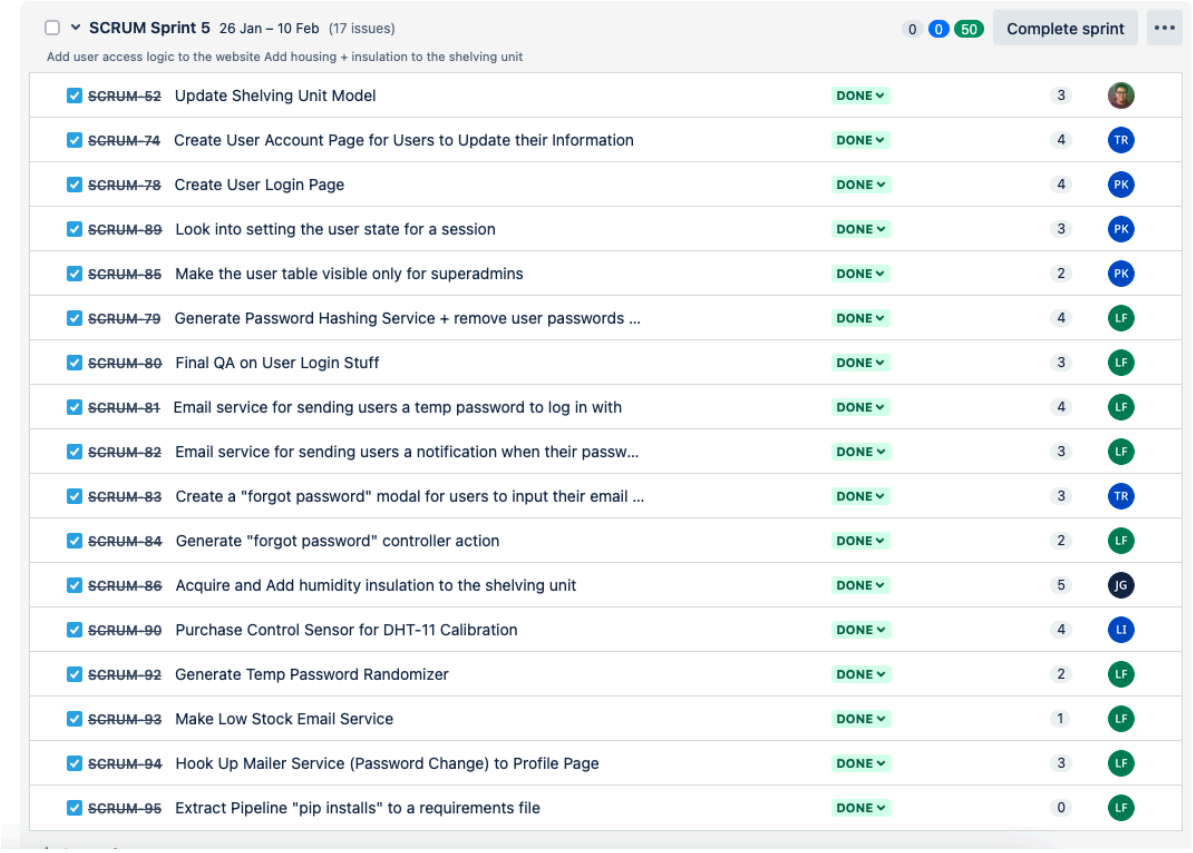
Fully implement CRUD. Purchase necessary physical material

~~SCRUM-29~~	Buy shelving unit & arduino supplies	DONE	4
~~SCRUM-30~~	Introducing controller endpoint for material update	DONE	3
~~SCRUM-35~~	Add a front end "create material" panel	DONE	3
~~SCRUM-31~~	Introducing controller endpoint for material deletion	DONE	3
~~SCRUM-32~~	Introducing controller endpoint for material creation	DONE	3
~~SCRUM-33~~	Add shelf textbox to material update panel and add column to mati...	DONE	2
~~SCRUM-34~~	Add front end confirmation of a material delete	DONE	2
~~SCRUM-36~~	Add a shelf attribute to the material entity	DONE	1
~~SCRUM-38~~	Final Verfication on Deletion	DONE	1
~~SCRUM-39~~	Final Verification on Mutation	DONE	1
~~SCRUM-40~~	Final Verification on Creation	DONE	1
~~SCRUM-41~~	Discuss Work Order With Nick to Resolve Shelving Concern	DONE	2
~~SCRUM-42~~	Look into Shelving Alternatives as a Backup	DONE	2

SCRUM Sprint 4 13 Jan – 26 Jan (19 issues) — 0 0 53 — Complete sprint

- Assemble shelving unit - Finalize material type, user_type and user visualization pages - Establish communication between the local application and the arduino hardware - C...

~~SCRUM-45~~	Display Table of Users: Front End Logic	DONE	3
~~SCRUM-46~~	Display Table of Users: Controller Endpoint + Backend Logic	DONE	3
~~SCRUM-48~~	Display Table of Material Types: Front End Logic	DONE	3
~~SCRUM-47~~	Display Table of Users: Connect Controller endpoint to front end + ...	DONE	3
~~SCRUM-49~~	Display Table of Material Types: Controller Endpoints + Backend Lo...	DONE	3
~~SCRUM-50~~	Display Table of Material Types: Connect Endpoint to Frontend + Ve...	DONE	3
~~SCRUM-51~~	Assemble Shelving Unit	DONE	7
~~SCRUM-60~~	Configure Microcontroller to Send Information over MQTT	DONE	4
~~SCRUM-61~~	Configure Local WebApplication to Recieve Information over MQTT	DONE	5
~~SCRUM-62~~	Update Shelf Database with Microcontroller Sensor Values	DONE	3
~~SCRUM-65~~	QA For Shelf Attribute Stories	DONE	3
~~SCRUM-66~~	Remove Material Name From Material and Replace With Supplier Li...	DONE	3
~~SCRUM-67~~	Add a Supplier Links Button to the Materials Table	DONE	1
~~SCRUM-68~~	Add Search Bar To Material Table To Filter By Name, Colour, Type, ...	DONE	3
~~SCRUM-69~~	Fix Model Unit Tests	DONE	1
~~SCRUM-70~~	Automate test execution with Github Action	DONE	3
~~SCRUM-71~~	Add shelf repo tests	DONE	1
~~SCRUM-72~~	Add ENV variables to automate DB connection	DONE	1
~~SCRUM-73~~	Add levenstien distance library to package.json	DONE	0



SCRUM Sprint 5 26 Jan – 10 Feb (17 issues) — 0 0 50 — Complete sprint

Add user access logic to the website Add housing + insulation to the shelving unit

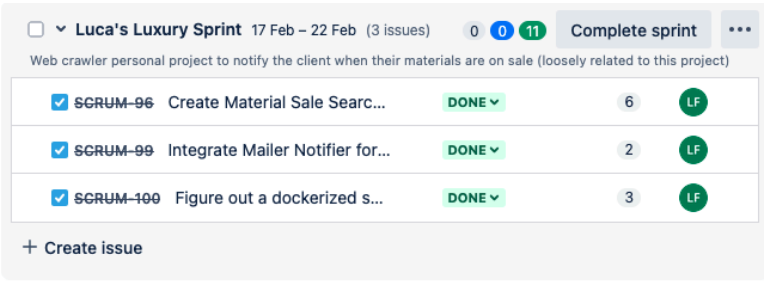
~~SCRUM-52~~	Update Shelving Unit Model	DONE	3
~~SCRUM-74~~	Create User Account Page for Users to Update their Information	DONE	4
~~SCRUM-78~~	Create User Login Page	DONE	4
~~SCRUM-89~~	Look into setting the user state for a session	DONE	3
~~SCRUM-85~~	Make the user table visible only for superadmins	DONE	2
~~SCRUM-79~~	Generate Password Hashing Service + remove user passwords ...	DONE	4
~~SCRUM-80~~	Final QA on User Login Stuff	DONE	3
~~SCRUM-81~~	Email service for sending users a temp password to log in with	DONE	4
~~SCRUM-82~~	Email service for sending users a notification when their passw...	DONE	3
~~SCRUM-83~~	Create a "forgot password" modal for users to input their email ...	DONE	3
~~SCRUM-84~~	Generate "forgot password" controller action	DONE	2
~~SCRUM-86~~	Acquire and Add humidity insulation to the shelving unit	DONE	5
~~SCRUM-90~~	Purchase Control Sensor for DHT-11 Calibration	DONE	4
~~SCRUM-92~~	Generate Temp Password Randomizer	DONE	2
~~SCRUM-93~~	Make Low Stock Email Service	DONE	1
~~SCRUM-94~~	Hook Up Mailer Service (Password Change) to Profile Page	DONE	3
~~SCRUM-95~~	Extract Pipeline "pip installs" to a requirements file	DONE	0

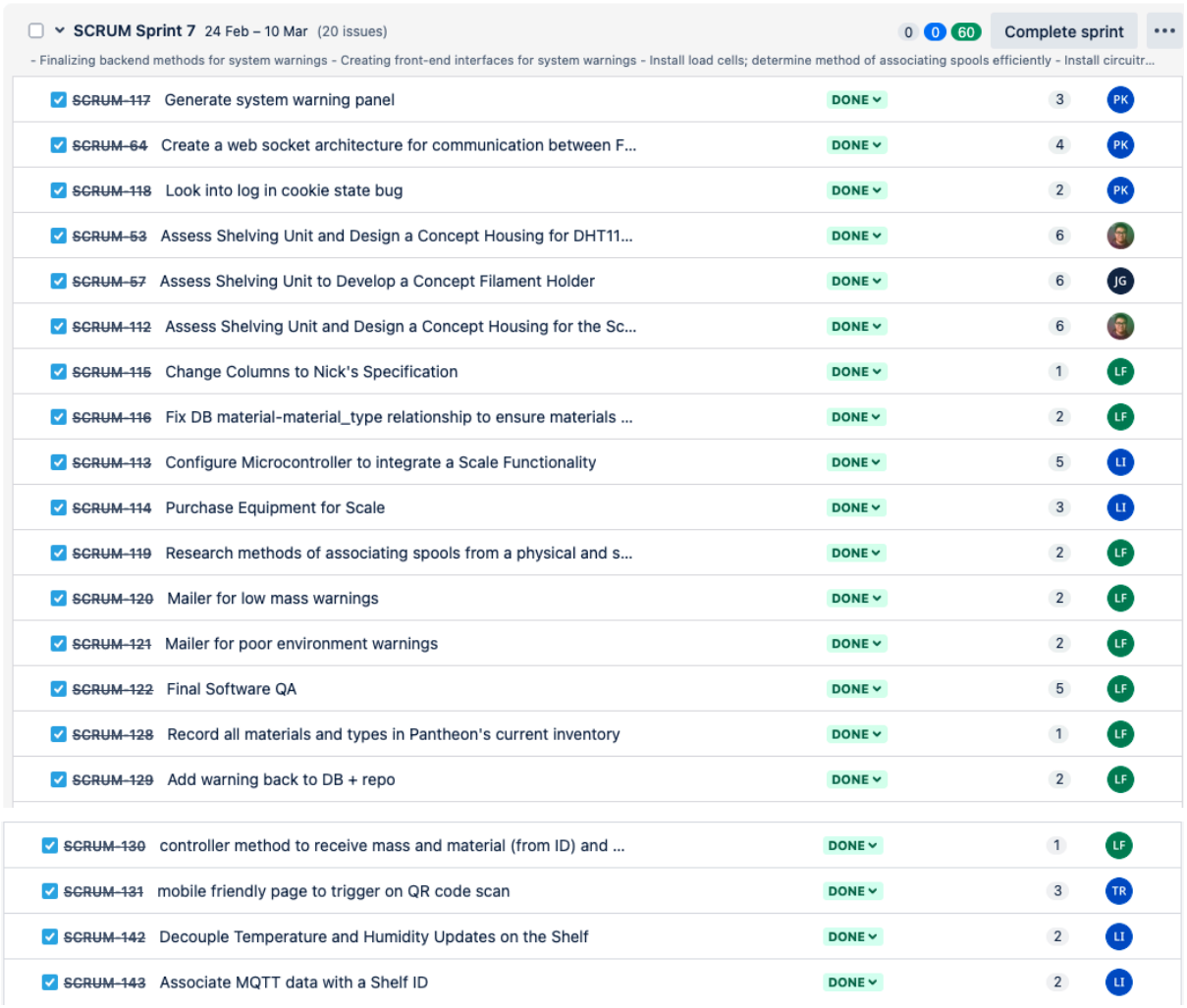
SCRUM Sprint 6 10 Feb – 24 Feb (12 issues) — 0 0 50 — Complete sprint

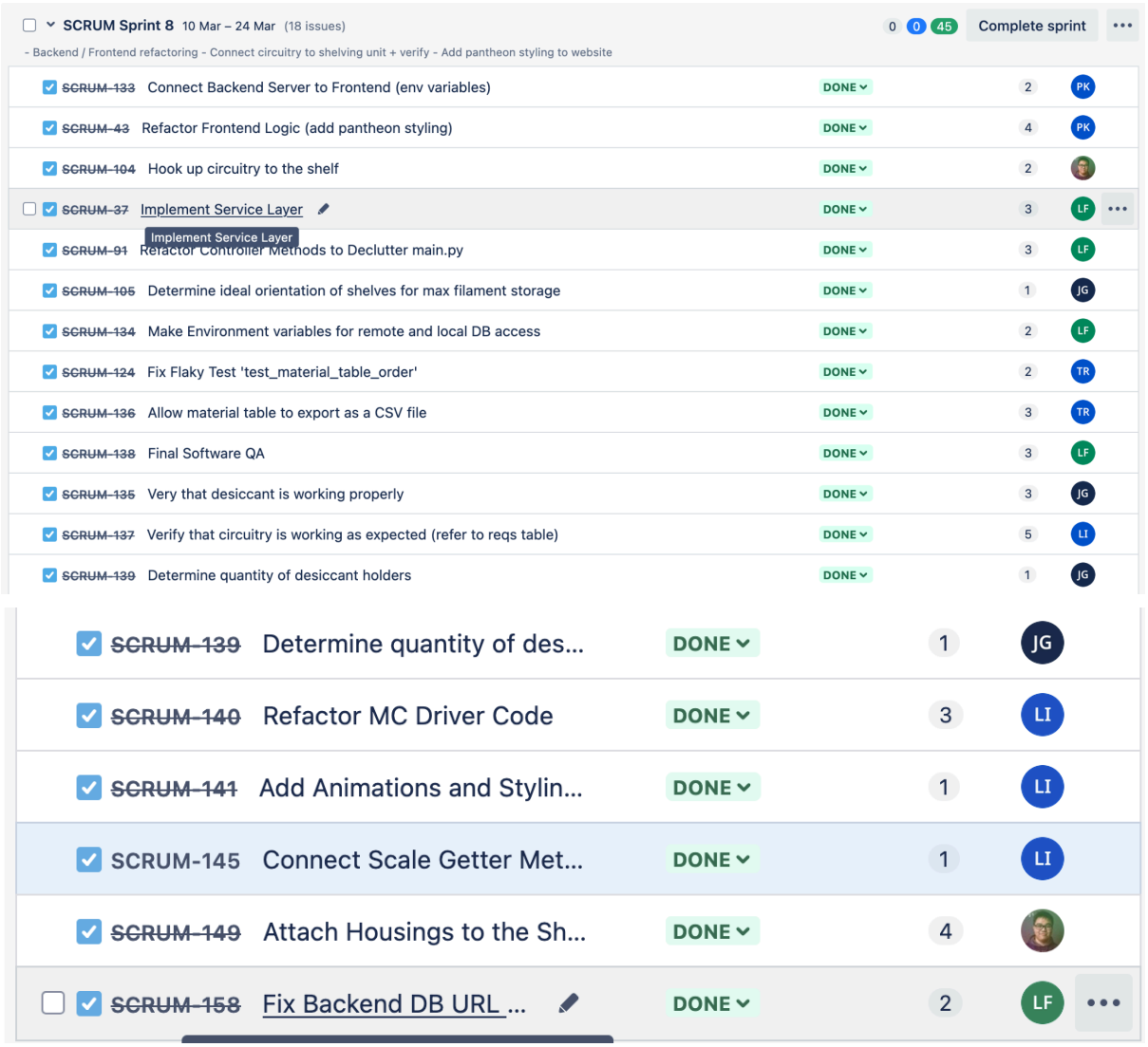
- Insulate shelf and add circuitry - Deploy webapp - Write progress report - Edit application to client's requests

~~SCRUM-102~~	Buy Dessicant + make holder model	DONE	2
~~SCRUM-103~~	Create dessicant filament holder model	DONE	4
~~SCRUM-58~~	Calibrate and Connect Arduino Sensors	DONE	3
~~SCRUM-59~~	Get WIFI Passkey for AirYorkGuest	DONE	2
~~SCRUM-107~~	Controller actions for mass replenishment and consumption	DONE	2
~~SCRUM-98~~	Reassess deployment options + present decision	DONE	5
~~SCRUM-97~~	Resolve Frontend Bugs For Deployment	DONE	8
~~SCRUM-108~~	Progress Report	DONE	10
~~SCRUM-106~~	Reformat material mass updates to account for adding or subt...	DONE	2
~~SCRUM-110~~	Change DB from SQLite to PostGres	DONE	4
~~SCRUM-126~~	Set Up Remote PostGres DB	DONE	3
~~SCRUM-101~~	Deploy the application backend + frontend (ensuring we can e...	DONE	5

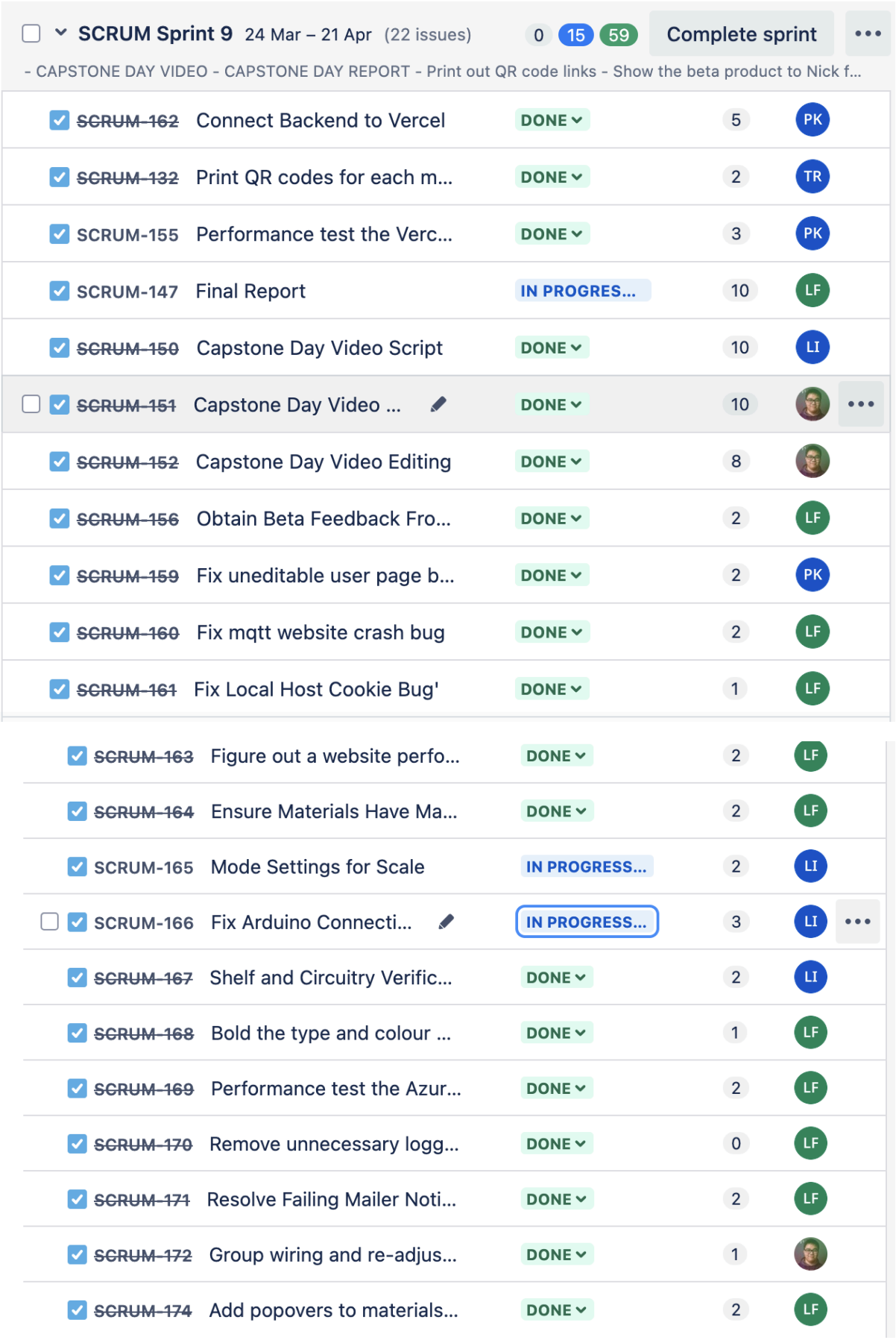
+ Create issue


Luca's Luxury Sprint 17 Feb – 22 Feb (3 issues)
0 0 11
Complete sprint
Web crawler personal project to notify the client when their materials are on sale (loosely related to this project)
SCRUM-96 Create Material Sale Searc... DONE 6 LF
SCRUM-99 Integrate Mailer Notifier for... DONE 2 LF
SCRUM-100 Figure out a dockerized s... DONE 3 LF
Create issue

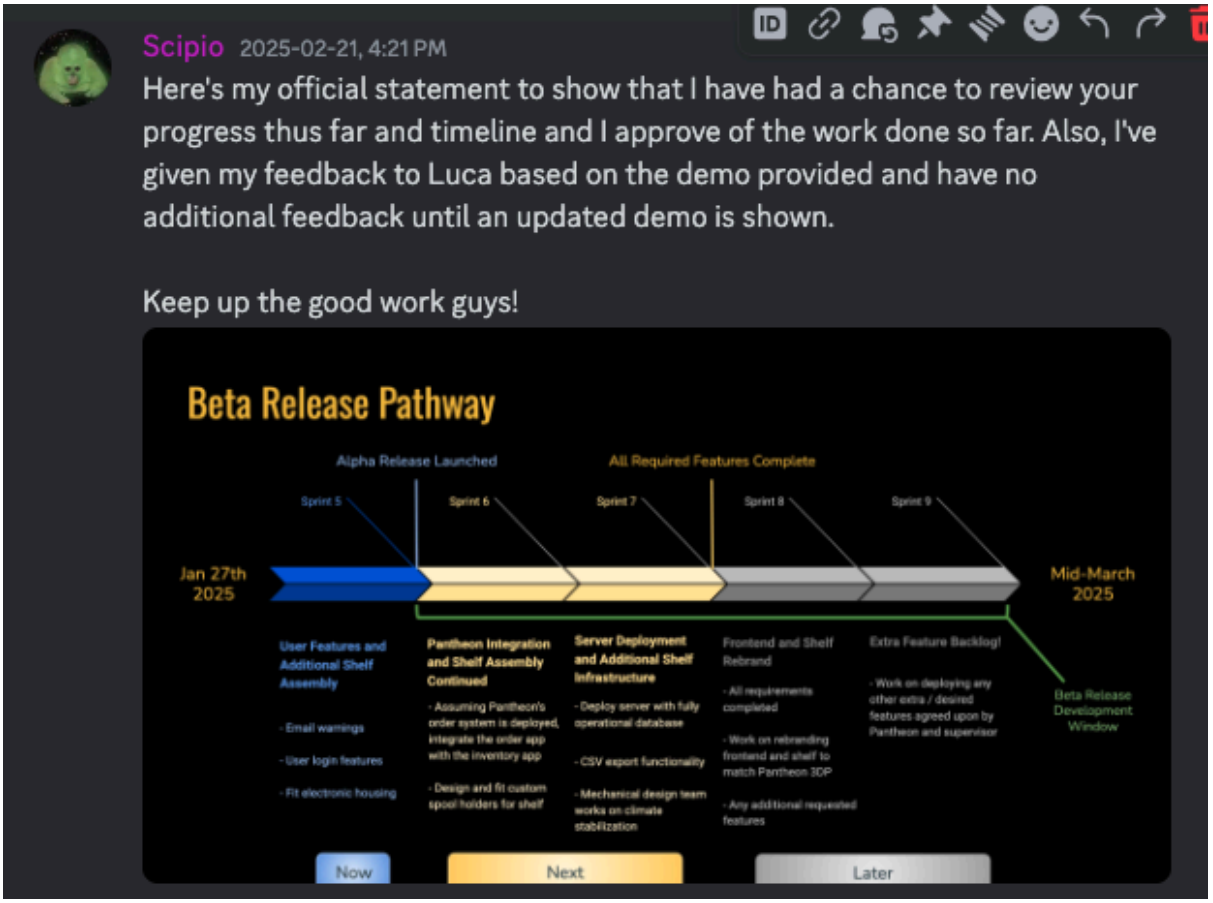

SCRUM Sprint 7 24 Feb – 10 Mar (20 issues)
0 0 60
Complete sprint
- Finalizing backend methods for system warnings - Creating front-end interfaces for system warnings - Install load cells; determine method of associating spools efficiently - Install circuitr...
SCRUM-117 Generate system warning panel DONE 3 PK
SCRUM-64 Create a web socket architecture for communication between F... DONE 4 PK
SCRUM-118 Look into log in cookie state bug DONE 2 PK
SCRUM-53 Assess Shelving Unit and Design a Concept Housing for DHT11... DONE 6
SCRUM-57 Assess Shelving Unit to Develop a Concept Filament Holder DONE 6 JG
SCRUM-112 Assess Shelving Unit and Design a Concept Housing for the Sc... DONE 6
SCRUM-115 Change Columns to Nick's Specification DONE 1 LF
SCRUM-116 Fix DB material-material_type relationship to ensure materials ... DONE 2 LF
SCRUM-113 Configure Microcontroller to integrate a Scale Functionality DONE 5 LI
SCRUM-114 Purchase Equipment for Scale DONE 3 LI
SCRUM-119 Research methods of associating spools from a physical and s... DONE 2 LF
SCRUM-120 Mailer for low mass warnings DONE 2 LF
SCRUM-121 Mailer for poor environment warnings DONE 2 LF
SCRUM-122 Final Software QA DONE 5 LF
SCRUM-128 Record all materials and types in Pantheon's current inventory DONE 1 LF
SCRUM-129 Add warning back to DB + repo DONE 2 LF
SCRUM-130 controller method to receive mass and material (from ID) and ... DONE 1 LF
SCRUM-131 mobile friendly page to trigger on QR code scan DONE 3 TR
SCRUM-142 Decouple Temperature and Humidity Updates on the Shelf DONE 2 LI
SCRUM-143 Associate MQTT data with a Shelf ID DONE 2 LI


SCRUM Sprint 8 10 Mar – 24 Mar (18 issues)
0 0 45 Complete sprint
- Backend / Frontend refactoring - Connect circuitry to shelving unit + verify - Add pantheon styling to website
SCRUM-133 Connect Backend Server to Frontend (env variables) DONE 2 PK
SCRUM-43 Refactor Frontend Logic (add pantheon styling) DONE 4 PK
SCRUM-104 Hook up circuitry to the shelf DONE 2
SCRUM-37 Implement Service Layer DONE 3 LF
Implement Service Layer
SCRUM-91 Refactor Controller Methods to Declutter main.py DONE 3 LF
SCRUM-105 Determine ideal orientation of shelves for max filament storage DONE 1 JG
SCRUM-134 Make Environment variables for remote and local DB access DONE 2 LF
SCRUM-124 Fix Flaky Test 'test_material_table_order' DONE 2 TR
SCRUM-136 Allow material table to export as a CSV file DONE 3 TR
SCRUM-138 Final Software QA DONE 3 LF
SCRUM-135 Very that desiccant is working properly DONE 3 JG
SCRUM-137 Verify that circuitry is working as expected (refer to reqs table) DONE 5 LI
SCRUM-139 Determine quantity of desiccant holders DONE 1 JG

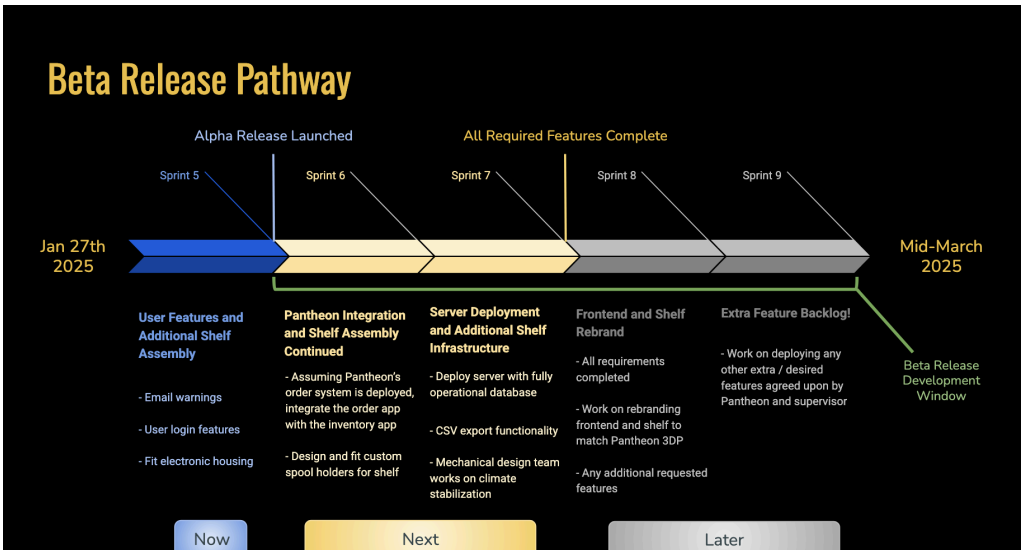
SCRUM-139 Determine quantity of des... DONE 1 JG
SCRUM-140 Refactor MC Driver Code DONE 3 LI
SCRUM-141 Add Animations and Stylin... DONE 1 LI
SCRUM-145 Connect Scale Getter Met... DONE 1 LI
SCRUM-149 Attach Housings to the Sh... DONE 4
SCRUM-158 Fix Backend DB URL ... DONE 2 LF



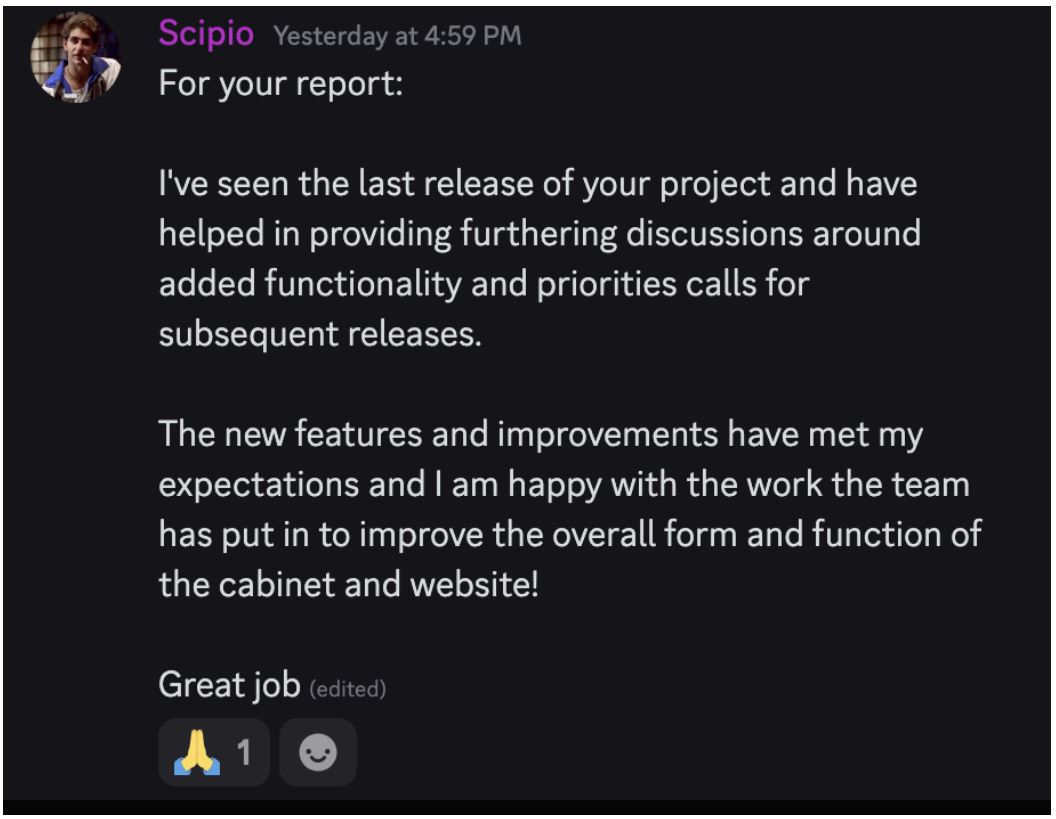
AF12: A Depiction of Each Sprint's Story Table (as of April 20, 2025)



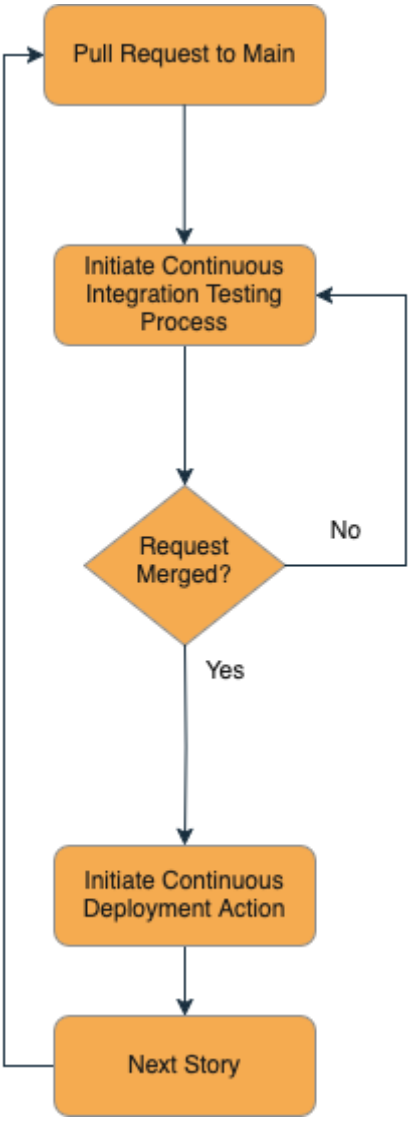
AF13: Evidence of stakeholder reviews (in addition to the supervisor meeting minutes). Discord is Nick's preferred method of communication.



AF14: January 2025 Agile Pathway



AF15: Final Contact Review on the Status of the Product



AF16: A Flowchart Depicting the Continuous Deployment Process for Updates to the Web Application

Endpoint	HTTP Method	Description
/access_management/login	POST	Verify a user's credentials and create a log in token
/access_management/logout	POST	Create a new cookie to end the user's session
/access_management/protected	GET	Get the user's token to verify their access permissions
/access_management/forgot_password	POST	Send a forgot password email to a valid user with a given email
/materials	GET	Return all materials
/material_types	GET	Return all material types

/users/all_users	GET	Return all users (renamed to ‘all_users’ to remove confusion with a frontend page)
/materials/create_material	POST	Create a material
/materials/delete_material/{entity_id}	DELETE	Delete a material given a material’s ID
/materials/update_material/{entity_id}	PUT	Update a material given its ID and parameters needed to be updated
/users/create_user	POST	Create a user
/users/delete_user/{entity_id}	DELETE	Delete a material given a user’s ID
/users/update_user/{entity_id}	PUT	Update a user given its ID and parameters needed to be updated
/material_types/create_mattype	POST	Create a material type
/material_types/delete_mattype/{entity_id}	DELETE	Delete a material type given its ID
/material_types/update_mattype/{entity_id}	PUT	Update a material type given its ID and parameters needed to be updated
/materials/replenish_mass/{entity_id}	PATCH	Add to the existing mass of a given material
/materials/consume_mass/{entity_id}	PATCH	Take from the existing mass of a given material
/qr/qr_display/{entity_id}	GET	Return a given material’s mass along with the current mass reading of the scale
/mqtt-status	GET	Returns the status of the material receiver and the shelf receiver (true if enabled, false if disabled)
/	GET	Root backend method

AT3: A Comprehensive List of All Implemented API Endpoints

Appendix B: Meeting Minutes

We initially used Confluence to store our meeting minutes. Upon the introduction of the supervisor meeting template, the necessary minutes have been converted to be course protocol conformant. Therefore, old supervisor meeting minutes and non-supervisor meeting minutes can be found on our Confluence page while supervisor meeting minutes can be found at this Google Drive link.

Appendix C: Resource Allocation Matrices

The following tables represent the RAMs used throughout the sprint planning process for each respective sprint. The following legend represents the symbolism of each letter within an individual RAM:

R = Responsible, A = Accountable, C = Consulted, I = Informed

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Create DB	R	A	I	I	C	C
Visualize a table of materials	A	C	I	I	R	C
Create a wrapper class and populate it with the necessary methods	A	R	I	I	C	C
Create table query class	C	I	I	I	R	A
Create population script	A	C	I	I	C	R
Generate a rough 3D model for the shelving unit	I	I	R	A	I	I
Set up the coding environment	C	C	I	I	R	C
Determine electrical components and incorporate it into the model	C	I	A	R	I	I
Refactor DB references to go through an ORM + add mass to the material object	R	C	I	I	C	C

CT1: RAM for Sprint 1

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Generate a Bill of Materials	C	C	R	C	I	I
Implement unit testing for the model classes	A	R	I	I	C	C
Implement feature testing for the basic table display	R	C	I	I	C	C

Add integration tests for the table controller class	A	A	I	I	C	R
Generate material event listener for mass dips below 50g	A	R	I	I	C	C
Add front end popup to react to a material dropping below 50 grams	A	C	I	I	R	A
Implement repository pattern + test	R	C	I	I	C	C
Add the mass setter method to model	A	R	I	I	C	C
Generate a controller method to update the DB upon a material's change in mass	R	C	I	I	C	C
Add front-end capabilities to edit the mass of a material	A	C	I	I	R	C
Midterm Report	A	A	A	A	A	A
Create Constants Class	A	R	I	I	C	C

CT2: RAM for Sprint 2

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Buying Materials	C	C	R	A	I	I
Assemble Shelving unit	I	I	R	R	I	I
Refactor notification system	C	R	I	I	A	C
Implement further CRUD functionality for material visualization	A	R	I	I	R	R

CT3: RAM for Sprint 3

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Display Table of Users: Front End Logic	A	C	I	I	R	C
Display Table of Users: Controller Endpoint + Backend Logic	R	A	I	I	C	C
Display Table of Material Types: Front End Logic	C	C	I	I	R	A
Display Table of Users: Connect Controller endpoint to front end + verification	C	C	I	I	A	R
Display Table of Material Types: Controller Endpoints + Backend Logic	R	A	I	I	C	C
Display Table of Material Types: Connect Endpoint to Frontend + Verification	C	C	I	I	A	R

Assemble Shelving Unit	C	C	R	A	I	I
Configure Microcontroller to Send Information over MQTT	A	R	I	I	C	C
Configure Local WebApplication to Receive Information over MQTT	A	R	I	I	C	C
Update Shelf Database with Microcontroller Sensor Values	A	R	I	I	C	C
QA For Shelf Attribute Stories	R	C	I	I	A	C
Remove Material Name From Material and Replace With Supplier Links Section	R	C	I	I	A	A
Add a Supplier Links Button to the Materials Table	R	C	I	I	A	A
Add Search Bar To Material Table To Filter By Name, Colour, Type, Mass	R	C	I	I	A	A
Fix Model Unit Tests	R	A	I	I	C	C
Automate test execution with Github Action	R	A	I	I	C	C
Add shelf repo tests	R	A	I	I	C	C
Add ENV variables to automate DB connection	R	A	I	I	C	C
Add levenstien distance library to package.json	R	A	I	I	C	C

CT4: RAM for Sprint 4

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Update Shelving Unit ModeL	C	C	R	A	I	I
Create User Account Page for Users to Update their Information	C	C	I	I	A	R
Create User Login Page	A	C	I	I	R	C
Look into setting the user state for a session	A	C	I	I	R	C
Make the user table visible only for superadmins	C	C	I	I	R	A
Generate Password Hashing Service + remove user passwords from table	R	A	I	I	C	C
Final QA on User Login Stuff	R	C	I	I	A	C
Email service for sending users a temp password to log in with	R	A	I	I	C	C
Email service for sending users a notification when	R	A	I	I	C	C

their password is changed						
Create a "forgot password" modal for users to input their email to recover their password	C	C	I	I	A	R
Generate "forgot password" controller action	R	C	I	I	C	A
Acquire and Add humidity insulation to the shelving unit	C	C	A	R	I	I
Purchase Control Sensor for DHT-11 Calibration	C	R	A	A	I	I
Generate Temp Password Randomizer	R	A	I	I	C	C
Make Low Stock Email Service	R	A	I	I	C	C
Hook Up Mailer Service (Password Change) to Profile Page	R	A	I	I	C	C
Extract Pipeline "pip installs" to a requirements file	R	A	I	I	C	C

CT5: RAM for Sprint 5

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Buy Dessicant + make holder model	I	C	A	R	I	I
Create dessicant filament holder model	I	C	A	R	I	I
Calibrate and Connect Arduino Sensors	C	R	A	A	I	I
Get WIFI Passkey from Edris	C	R	A	A	I	I
Hook up circuitry to the shelf	I	C	R	A	I	I
Determine ideal orientation of shelves for max filament storage	I	C	A	R	I	I
Controller actions for mass replenishment and consumption	R	A	I	I	C	C
Reassess deployment options + present decision	R	C	I	I	A	A
Deploy the application backend + frontend (ensuring we can easily push updates if needed)	A	C	I	I	R	A
Resolve Frontend Bugs For Deployment	C	C	I	I	A	R
Progress Report	R	A	A	A	A	A
Reformat material mass updates to account for adding or subtracting mass from the material's existing mass	A	C	I	I	R	A

Change DB from SQLite to PostgreSQL	R	A	I	I	C	C
Set Up Remote PostgreSQL DB	R	A	I	I	A	A

CT6: RAM for Sprint 6

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Assess Shelving Unit and Design a Concept Housing for DHT11 Sensor	I	C	R	A	I	I
Assess Shelving Unit to Develop a Concept Filament Holder	I	C	A	R	I	I
Create a web socket architecture for communication between Frontend and Backend	A	C	I	I	R	C
Assess Shelving Unit and Design a Concept Housing for the Scale	I	C	R	A	I	I
Change Columns to Nick's Specification	R	C	I	I	A	A
Fix DB material-material_type relationship to ensure materials delete when the material_type deletes	R	C	I	I	A	A
Configure Microcontroller to integrate a Scale Functionality	C	R	A	A	I	I
Purchase Equipment for Scale	C	R	A	A	I	I
Generate system warning panel	A	C	I	I	R	A
Look into log in cookie state bug	A	A	I	I	R	A
Research methods of associating spools from a physical and software standpoint	R	A	I	I	C	C
Mailer for low mass warnings	A	R	I	I	C	C
Mailer for poor environment warnings	A	R	I	I	C	C
Final Software QA	R	A	I	I	A	A

CT7: RAM for Sprint 7

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Connect Backend Server to Frontend (env variables)	A	C	I	I	R	C
Refactor Frontend Logic (add pantheon	C	C	I	I	R	A

styling)						
Hook up circuitry to the shelf	I	C	R	A	I	I
Implement Service Layer	R	A	I	I	C	C
Refactor Controller Methods to Declutter main.py	R	A	I	I	C	C
Determine ideal orientation of shelves for max filament storage	I	C	A	R	I	I
Make Environment variables for remote and local DB access	R	C	I	I	A	C
Fix Flaky Test 'test_material_table_order'	A	C	I	I	C	R
Allow material table to export as a CSV file	A	C	I	I	C	R
Final Software QA	R	C	I	I	A	C
Very that desiccant is working properly	I	C	A	R	I	I
Verify that circuitry is working as expected (refer to reqs table)	I	R	A	C	I	I
Determine quantity of desiccant holders	I	C	A	R	I	I
Refactor MC Driver Code	I	R	A	C	I	I
Add Animations and Styling to MC	I	R	A	C	I	I
Connect Scale Getter Method to QR Controller	I	R	A	C	I	I
Attach Housings to the Shelf	I	C	R	A	I	I
Fix Backend DB URL Overwrite Bug + Shelf Errors	R	A	I	I	C	C

CT8: RAM for Sprint 8

Scrum	Luca	Luqmaan	Ethan	Jean	Paras	Tharuveen
Connect Backend to Vercel	A	C	I	I	R	C
Print QR codes for each material	A	C	I	I	C	R
Reformat file names for consistency	R	A	I	I	C	C
Final Report	R	A	A	A	A	A
Capstone Day Video Script	C	R	I	I	I	A
Capstone Day Video Shoot	I	A	R	I	I	C

Capstone Day Video Editing	I	A	R	I	I	C
Performance test the Vercel frontend	A	C	I	I	R	C
Obtain Beta Feedback From Nick	R	C	I	I	A	C
Restrict QR calls to York's network	R	A	I	I	C	C
Fix uneditable user page bug	A	C	I	I	R	C
Fix mqtt website crash bug	R	A	I	I	C	C
Fix Local Host Cookie Bug'	R	A	I	I	C	C
Figure out a website performance standard	R	A	I	I	C	C
Ensure Materials Have Mandatory Type ID Field	R	C	I	I	A	C
Mode Settings for Scale	C	R	A	C	I	I
Fix Arduino Connection with AirYorkGUEST	A	R	I	I	C	C
Shelf and Circuitry Verification With Nick	C	R	C	A	I	I
Bold the type and colour of the material in warning email	R	A	I	I	C	C
Performance test the Azure backend	R	A	I	I	C	C
Remove unnecessary logging on listener manager	R	A	I	I	C	C
Resolve Failing Mailer Notif For Shelf Listener	R	A	I	I	C	C

CT9: RAM for Sprint 9

Appendix D: Risk Registers

D.1: Fall Risk Register

Below is a detailed explanation of each of the risks from the Fall sprints. Attached below is a risk register table, which we used to rank the severity and likelihood of each associated risk (DT1). We also included a legend below the table to describe each colour chosen (DT2).

- **Physical Risk 1 (PR-1)**
    - **Shelf Instability:** The shelf collapses due to the weight of the 3D printing materials. This would be considered a minor issue but unlikely to happen. The probable strategies are communicating with Pantheon to figure out the total weight of the materials and researching shelves that will accommodate the weight.
- **Physical Risk 2 (PR-2)**

    - **Material Degradation:** Materials deteriorating due to moisture or heat exposure within the shelf. This would be a significant and likely issue as it is one of Pantheon's main requests with this project. To solve this risk, we plan to attach sensors throughout the shelf to monitor the humidity and temperature and be able to send the data back to the web application.
- **Physical Risk 3 (PR-3)**
    - **Space Constraints:** The materials may likely take up a lot of space within the shelf. This risk would most likely happen but it is a very minor issue that can be dealt with very quickly. To satisfy this issue, we plan to measure the materials and compare them with the shelf dimensions to fit all of Pantheon's materials.
- **Physical Risk 4 (PR-4)**
    - **Difficult to Assemble:** Assembling the electronics such as the sensors or microcontrollers onto the shelf can lead to many issues and defects. This is possible but a minor issue that can occur when our team is assembling the shelf during the final sprints. To prevent these risks, we plan to create diagrams before assembling the actual to make sure there are no issues with attaching parts.
- **Programming Risk 1 (PR-5)**
    - **Integration:** Integrating the shelf with the web application will have issues with sending and receiving data. This is a severe risk that is very likely to happen as the main priority is integrating the order application with the shelf. The probable strategy to integrate is to use APIs and create integration tests.
- **Programming Risk 2 (PR-6)**
    - **Deployment Issues:** Deploying the web application during the final stage might lead to issues as team members have inexperience in deploying applications. This is a likely risk but will only have a moderate impact due to the number of computer engineers in the group. Our immediate strategy is to test deployment during the last couple of sprints to see if the web application can be self-standing.
- **Programming Risk 3 (PR-7)**
    - **Data Corruption:** Data can be corrupted if material data is lost during a software bug such as improper setup of the database. This would be a moderate risk which is possible to happen during the final sprints of this project. To mitigate this issue, we plan to incorporate integration, unit, and feature testing to make sure that data is not corrupted with the inclusion of test recovery processes.
- **Project Risk 1 (PR-8)**
    - **Feature Issues:** Features might not function as planned due to issues within the coding environment. Features like the notification or the filtering system might not work as planned which can cause delays in deploying the application within the deadline. This would be considered a minor but very unlikely issue. This is because our strategy is to test each feature frequently to ensure they work well together.
- **Project Risk 2 (PR-9)**
    - **Implementation:** Implementing the web application based on Pantheon's needs might become an issue due to time constraints and team member schedules. This risk would be likely to occur but it's a minor issue. To prevent this risk, we plan to schedule our tasks properly with the use of Jira and Discord to ensure we are not facing any struggles during the last couple of sprints.

		Impact				
		Negligible	**Minor**	**Moderate**	**Significant**	**Severe**
Likelihood	**Very Likely**	Low	Medium: PR-3	Medium	High	High: PR-5
	Likely	Low	Medium: PR-9	Medium: PR-6	High: PR-2	High
	Possible	Low	Medium: PR-4	Medium: PR-7	High	High
	Unlikely	Low	Low: PR-1	Medium	Medium	Medium
	Very Unlikely	Low	Low: PR-8	Low	Medium	Medium

DT1: Fall Risk Register

D.2: Winter Risk Register

Below is a detailed explanation of each of the risks. Attached below is a risk register table, which we used to rank the severity and likelihood of each associated risk (Table 23). We also included a legend below the table to describe each colour chosen (Table 24).

- **Physical Risk (PR-1)**
  - **Capacity Restraints:** The DHT11 sensor and the filament holder will take up some space within the shelf which will cause smaller capacity for the spools and filaments. The probable strategies were to find shelf sizes that were able to accommodate both the materials and the corresponding sensors/holders. This was considered a significant and likely issue which needed priority as we assembled the shelf.
- **Physical Risk (PR-2)**
  - **Scale Functionality:** We introduced the idea of a scale to be included with the shelf in order to weigh the mass of the spools/filaments. Therefore, the risk occurs with the scale being able to function collectively with the shelf. The probable strategy was to research and find the type of equipment needed for the scale to be able to integrate it into the shelf. This is a possible and moderate risk that we are dealing with as of right now.
- **Programming Risk (PR-3)**
  - **Web Socket Architecture:** Creating the architecture in order for the frontend (client) to communicate continuously with the backend (sever). This allows the server to push data without the client requesting a call every time. The probable strategy was to ensure that the connections were verified and data was being sent out continuously

without the client requesting an update. This is a very severe and likely issue that we are dealing with right now.

- **Programming Risk (PR-4)**
  - **Software Bugs:** With coding, the chances of bugs/warnings to occur is very high and it's important to regularly check to see if there are any new issues that need to be fixed. The probable strategy is to ensure that we have a final software QA to check if there are any bugs with certain codes or if a page from the website doesn't load as expected. This is a minor but a very likely issue that we have encountered during the final stages of the project.
- **Project Risk (PR-5)**
  - **Implementation:** Implementing the website to communicate with the shelf is a major risk that we are dealing with. To ensure the data from the shelf is communicating effectively with the website, we need to ensure the connections are secure and the calls are being handled as expected. This is a very severe and likely issue that is ongoing with the project.
- **Project Risk (PR-6)**
  - **Integration of Order App:** The main idea was to integrate Pantheon's order app into our website as of right now. However, they have been having issues with the app and the calls are not being handled properly so we have to come up with a new solution. The probable strategies to come up with a new solution is figuring out the timeframe we have and what functions/features can be created that ensures the main purpose is still being targeted. This is a significant and possible risk that needs to be attended to.

		Impact				
Likelihood		Negligible	Minor	Moderate	Significant	Severe
	Very Likely	Low	Medium: PR-4	Medium: PR-6	High	High: PR-5
	Likely	Low	Medium	Medium	High: PR-1	High: PR-3
	Possible	Low	Medium	Medium: PR-2	High	High
	Unlikely	Low	Low	Medium	Medium	Medium
	Very Unlikely	Low	Low	Low	Medium	Medium

DT2: Winter Risk Register

Colour	Description
Green	A minor issue that can be solved immediately
Yellow	Moderate issue that needs 1-2 people to look over
Red	A major issue that needs everyone’s attention

DT3: Risk Register Legend

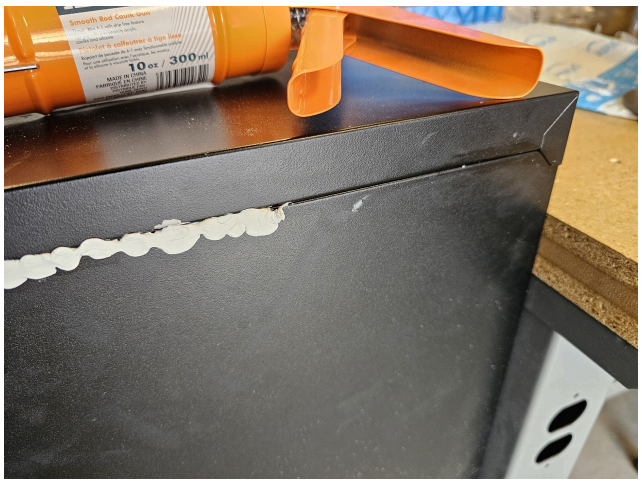
Appendix E: Project Vandalism Incident

E.1: Formal Email to Capstone Teaching Team and York Security

Below is a capture of the email message addressed to the Capstone Teaching Team and York Security. Also included is Pantheon3DP's CEO Nick DiScipio, our supervisor Professor Hung, and all the members of Capstone Team 8 (our group).

> *Hello,*
>
> *My name is Luqmaan Irshad and I am a student from ENG 4000 Capstone this semester. I have already got in contact with our course directors about this issue, and we were hoping you could look further into this incident, as our project has been damaged.*
>
> *The full extent of the damage is still being assessed, and we have not confirmed yet whether items have been stolen. This shelf and its materials belong to our Capstone Group (8) and Pantheon 3DP.*
>
> *As of now, here are the current assessed damages (a more thorough investigation will take place later today / next week):*
>
> *- **Caulking insulation removed** from the shelf; it **does not properly regulate the internal climate anymore** (it should be below 20% humidity), desiccant needs to be replaced, any printing material inside the cabinet could be compromised*

EF1 - Shelf Humidity is not regulated properly anymore after damages

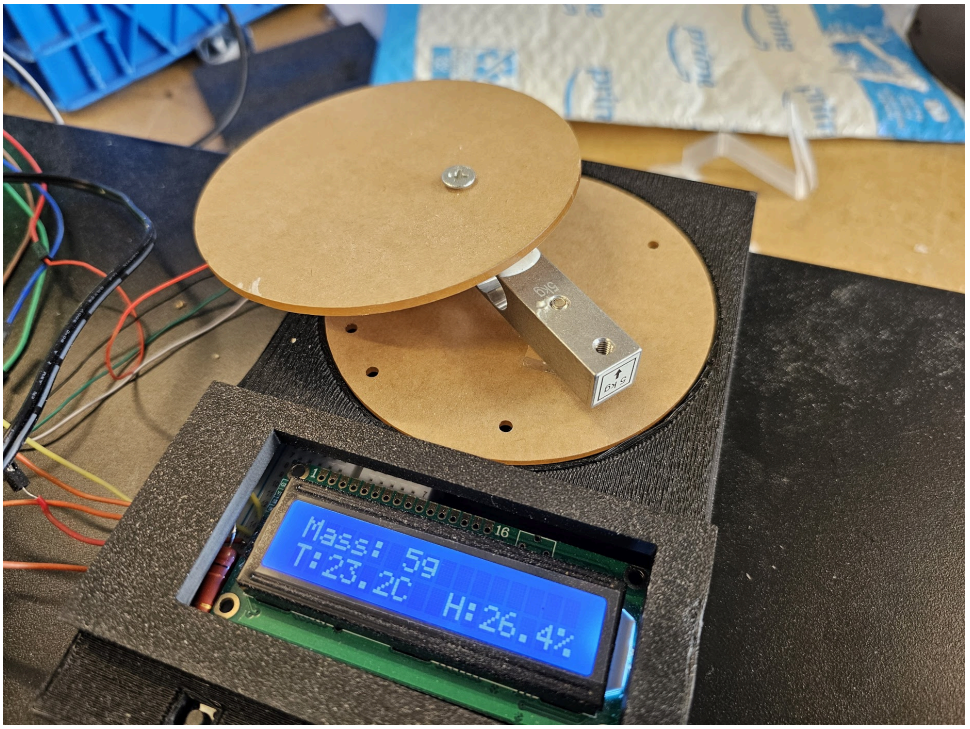


EF2: Caulking Insulation Removed From Shelf

*- **Electronics** forcefully **unplugged**; **extension cord** being used was **stolen**, **connector** to the Arduino was **yanked** out of place (demonstrates forceful behaviour)*

- Materials that were placed on top of our shelf are scattered across the room; most of the parts we had (empty spool rack, sign saying "Do not touch capstone group 8", empty Amazon Prime bag, etc) have been pushed to the side, some of it smeared with caulking as well

*- Scale disassembled; someone has found a **screwdriver to purposely unscrew and disassemble the scale assembly**. Scale needs to be tested again, **potential load cell damage** if improperly handled*



EF3: Scale Maliciously Disassembled Using A Screwdriver

*- **Shelf smeared** and sprayed **with caulking** (thankfully just cosmetic damage in this case)*

EF4: Shelf Vandalized With Our Team's Purchased Caulking. Possible "NM" Initials

*- We also noticed the **key to the cabinet was on the ground**. This key was taped to the bottom of the shelf in case of emergency access, but we found it lying on the ground and are still assessing whether there has been stolen / damaged items or damage to the shelf internally*

EF5: Key To The Shelf Found On The Ground

The financial cost of the damage is in the process of being calculated, but as for a time estimate, this could take our group an extra week to repair and reverify (assuming things work the first time). Any further assessed damages will cause an even longer delay, and may impede our completion towards final capstone day.

It may also be worthy to note, we have had previous issues in the past few weeks of people mistreating our shelf out of potential negligence; they have been using the top of our shelf as a work desk, carelessly dropping their metal tools and other work materials on it, and leaning on it. Due to this, we have noticed the top of our shelf has dents in the metal.

We have asked people politely to stop using our project as a desk, as they were causing damage to it. This, however, what was captured above is clearly more than just simple negligence, and is a deliberate attempt to sabotage and destroy some of our project (as demonstrated by the motive to unscrew parts, smear caulking, tear out insulation, etc.).

Another noteworthy piece of information is that the shelf was in perfect condition (other than the dents) before Friday around 5:00pm, as we had just finished recording the video for our Capstone presentation. Part of this presentation was recorded in the same garage, since the shelf resides there. Some time between then and Sunday (the next time I checked on the circuitry status), we noticed the Arduino had gone offline. Whether the Arduino unplugging was related or not is not yet clear, but it's entirely possible the damages took place right after we left on Friday.

The Capstone directors are considering taking action as well if we can figure out who was responsible (in the case this was someone from the class).

Thank you for taking the time to read about our concerns. We hope that this is the first and last time this happens to any group.

Please keep us updated on the status of this incident as the people responsible could try to damage other projects, and with our final capstone day coming in the next couple of weeks, could be absolutely detrimental.

Sincerely,

Luqmaan Irshad
217222365
luq21

E.2: Damage Report - Costs

Luckily, the damages were repairable, but they required a lot of time to fix. It has taken us over 3 weeks to fix the damages and retest affected features. The only monetary cost seems to be the caulking they used that we purchased, although with the price of the caulking, this could be considered negligible. The real impact of the damages were in the time required to fix all the damage, and the mental / emotional impact it has had on our team. Due to the incident, we have been forced to relocate our project early to Pantheon's office, which trades off access to our project in favour of security.

Appendix F: Previous ITP Ratings

F.1 Previous ITP Reviews

F.1.1 ITP 1 and 2 Reviews - Luqmaan Irshad

Review 1: The feedback I received was very positive, scoring 5.0 for almost every category (commitment, communication, capabilities, focus and standards). This means I have put on a strong start as a leader and communicator, as well as holding up to standards and providing technical expertise. My goals from here would be to use this as a starting platform and iterate on the few points I missed to aim for an even 5.0 across the board. In terms of the team, I think we have solid foundations and a good start. For continued growth, I look forward to seeing everyone take more initiative as the project develops.

Review 2: Similar to the first review, I received almost 5.0 scores in every category, even doing slightly better in the commitment component (up from 4.8 to 5.0). Despite external factors that caused me to request a reduction in my workload for a couple weeks, my team was understanding and was able to value the efforts I made to still contribute at maximum capacity. Overall, I am satisfied with my performance and will continue to push for the best in our team. In regards to our team, I think we are seeing great growth as everyone is now beginning their roles in full capacity. It's great to see the feedback I shared from the first review was taken into consideration. In spite of this, the team has grown in a positive direction, becoming more independent and forming structures and systems to regularly communicate and collaborate.

F.1.2 ITP 1 and 2 Reviews - Luca Filippelli

Review 1: Similar to Luqmaan, the feedback I received was very positive, scoring almost all 5.0s across the board. It seems the team trusts me as a capable and dependable group leader. The feedback received was to keep doing what I'm doing. To improve, I will continue to develop these behaviours for the sake of personal growth and to continue sharpening my skills as a leader. In terms of our team, I think we're doing a great job so far and can continue to improve by communicating more often to make sure we're covering all aspects of the design.

Review 2: Similar to review 1, the feedback I received was overwhelmingly positive. I improved in the aspects I scored 4.8 on, and attained an average score of 5.0 in every category for this review. My self-reflection aligns with these scores, as I have put a lot of effort in organizing and leading groups and software developments. I have taken on the additional role of being an architect on the software end, which requires good knowledge, skill and communication skills, all of which were rated high for this review. I will continue to enforce high standards across our team to achieve something great. As for the team, I think we're doing a good job getting most of our story points completed for each sprint. I have confidence our team will be able to continue producing quality work going forward.

F.1.3 ITP 1 and 2 Reviews - Tharuveen Raveendran

Review 1: I received very positive feedback for my ITP metrics review. I seem to underestimate myself when it comes to my communication and capabilities, based on the fact my team gave me higher ratings. I will continue to work hard and try to tackle the project with more confidence, seeing the group trusts and values my opinions and work. Currently, the team works well together, and I have no complaints from a team-based perspective.

Review 2: Similar to the last review, I have received positive feedback overall for my ITP metric review from my teammates. I have received an increase in my commitment (4.2 to 4.6), focus (4.2 to 4.8) and standards (4.6 to 4.8). I am proud of the increase in my commitment, as previously it was my lowest score. This is something I actively tried to improve, and it seems like my team agrees. I received specific feedback in the comments section of the report which mentioned my punctuality is an asset to the team, as it motivates others to arrive and complete tasks on time. I will continue to try to improve in communication by responding to messages and discussing and verbalizing my opinions and ideas more. In terms of the team, I believe we are doing good work, and everyone is taking responsibility for their roles. I look forward to continuing working with the team as we begin the next half of the project!

F.1.4 ITP 1 and 2 Reviews - Paras Kumar

Review 1: The feedback I received was overwhelmingly positive. I was told I have great technical skills and can have a good balance between the output I have towards the group and my outside time commitments, to make sure they don't interfere with my progress. To improve, I can continue to be more vocal about my opinions and offer solutions, as the team recognizes my expertise as valuable. There are no major issues or challenges I face in the group, and I look forward to continuing working with them!

Review 2: During this review, I received even more positive feedback from my team. I have improved in my commitment (4.2 to 4.8), communication (4.6 to 4.8) and capabilities (4.4 to 4.6). The comments I received were positive as well, mentioning things like "great work with the frontend

development", and even mentioned they saw growth in my initiative and leadership on the frontend development aspect of our project. I am happy that my team has recognized my efforts and skills, and they seem to show trust in my abilities. In terms of the team, I am content with our progress and work ethic. Everyone is getting their work done on time, and we cover for each other if someone cannot make it to a meeting. Looking forward to the next half of our project together!

F.1.5 ITP 1 and 2 Reviews - Ethan Tran

Review 1: Overall, I'm satisfied with the review I received. I have a good balance of skills across the board based on the ratings I received, with communication being my strongest suit. Since I updated the group ahead of time and communicated as the main architect for the mechanical aspects of the project, the group has ranked my communication skills as my strongest skill. Despite communication being my highest skill, one comment that was mentioned a few times was to be more vocal and confident in my ideas. The team seems to trust my skills, so I should try to voice my ideas more often. Going forward, I will try to deliver on this aspect, while also taking more of a leadership role as the main expert on the mechanical aspect of the project. I believe our team works well together and I'm happy with the current arrangement. I look forward to seeing what our group can do together.

Review 2: Overall, I am satisfied with my results for the second review. I was able to improve my commitment (4.2 to 4.6) and standards (4.0 to 4.4). Commitment now ties in with my Communication skills, which is my strongest skill in both reports. I feel like this represents my work well, as I took on the huge responsibility of being in charge of our team's BOM. The team seems to value my commitment because of this, which is why I believe my score improved here. Due to my thorough attitude when approaching the BOM, my standards have also improved, which was reflected in the score increase. At the moment, focus seems to be my lowest score at 4.2. I complete my work on time, but one thing I think I can do to improve in this aspect would be to look out for others and remind the team of upcoming due dates. Overall, I am satisfied with the team's performance, the software side has got a lot of work done. On the mechanical end, we've got all our required work done, and now wait for the BOM to be approved so we can start shelf construction.

F.1.6 ITP 1 and 2 Reviews - Jean Granato

Review 1: Overall, I'm pleased with the feedback I received from my peers, they highlighted my strengths across many teamwork dimensions. The highest scoring dimensions are Role clarity and Communication, I feel that it's quite accurate. I'm very good at articulating ideas and communicating through problems. I also feel that my high score in the role clarity dimension is accurate as I do follow the role that has been assigned to me. I handle a lot of the ideation and problem-solving on the mechanical aspect of the project. Some of my lower-scoring dimensions are commitment and standards. As someone who is constantly busy with extracurricular activities and other commitments, a challenge that I face is that I cannot give this project my full attention. I think that the feedback that has been received from my group is highly accurate and I can improve in various ways such as devoting more time to work on our project and giving time to be thorough with my work. I will work towards improving some aspects, and maintain others. It is a pleasure working with my team and we work together very well.

Review 2: This round of feedback I received from my peers was a pleasant surprise. The feedback they provided showed an improvement in nearly every teamwork dimension. My highest scoring dimensions are focus and standards which reflect my effort to provide high-quality work to the team. My peers mentioned their appreciation of my dedication to ensuring our work meets the requirements

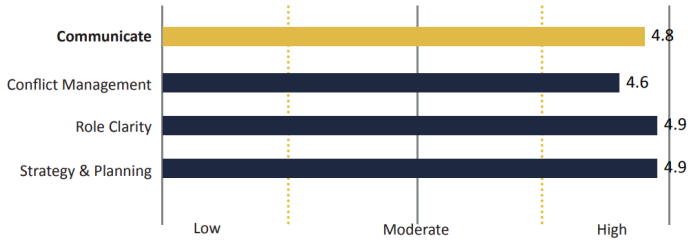
and my ability to anticipate challenges we might face (for the mechanical aspects). At the same time, my dimensions like commitment and communication are not low but could use improvement. My peers acknowledged my strong knowledge of 3D printing and the timely completion of tasks. They suggested our team could benefit from being more communicative in busy times so that they may help relieve some of the burden or know what to expect over that day. Going forward I will strive to maintain my strengths and improve on communicating more promptly when external obligations affect my availability. Overall I value the team, their feedback and ratings. I deeply enjoy working with my team and I know we are gonna come up with an amazing capstone project. According to my conflict management report, my ability to approach conflicts by seeking a cooperative mutual solution is strong. I'm good at fostering mutual respect and turning conflicts into opportunities for improvement. However, I see that there is room for me to grow for consistent role clarity and proactive communication. I will address my shortcomings by actively communicating, addressing disputes and clarifying my role in particular tasks. This will help further foster trust and collaboration creating a successful team environment.

F.2 ITP Team Dynamics

After reflecting on our individual and team ITP metrics, we've learned a lot about how our peers perceive us and the dynamics of our team. ITP breaks down its ratings into four separate categories; Communication, Adaptation, Relation and Education.

F.2.1 ITP Team Metrics: Communication

Communication deals with the ability to form a cooperative environment, along with developing clear goals and roles for the team to follow. From the last team breakdown, our overall communication score has improved from a 4.6 to a 4.8 (FF1). We improved in all aspects of communication, jumping from 4.4 to 4.6 on conflict management, 4.7 to 4.9 on role clarity and 4.6 to 4.9 on strategy and planning. Since our last review, we have added and clearly defined more roles for each group member (Adding project leaders, architects for software and mechanical, etc.) which have defined each team members' roles and contributions more vividly. Our strategy and planning has improved due to our consistency to host and record feedback during weekly meetings, which allows us to plan our sprints at least a month in advance with minimal change. Overall, our team is very strong with communication, and we can seek to improve by further taking initiative to drive conversations as soon as we have new information.

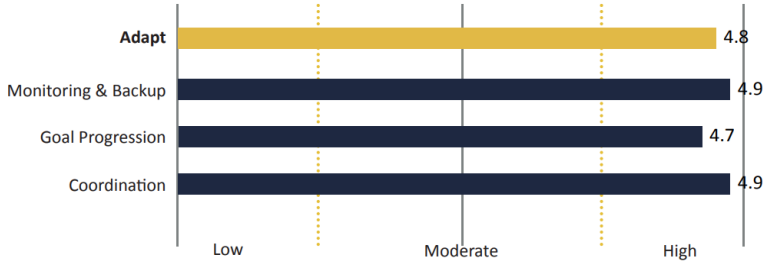


FF1: Communication statistics for ITP Team Dynamics.

F.2.2 ITP Team Metrics: Adaptation

Adaptation focuses on the flexibility to pivot based on modulating task priorities, as well as the ability to provide support to other members when they need it. Since our last review, we have seen an increase in our team's adaptiveness overall, jumping from 4.6 to 4.8 (FF2). We seem to have

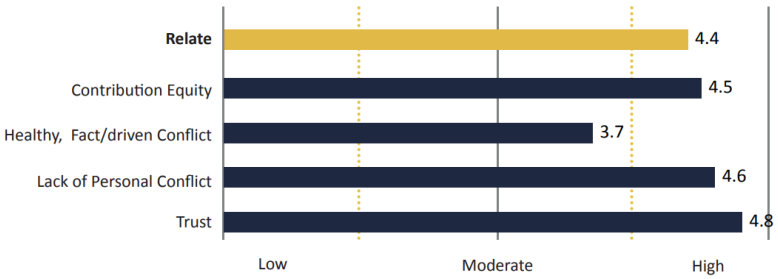
improved in every aspect of adaptation as well, jumping from 4.5 to 4.9 in monitoring and backup, 4.6 to 4.7 in goal progression and maintaining 4.9 in coordination. This semester has been particularly challenging for the team, as the semester has become very work intensive. Despite this, each group member has had the opportunity to cover for another at some point this term, which helped boost our monitor and backup score. With the evolution of our sprints and roles being more clearly defined, we are able to more accurately discern whether we are on track to completing our goals. Having members contribute their own stories to the sprint backlog has been critical in helping define goal progression for each sprint. Our coordination has always been a strength. We are able to complete a majority of our tasks each sprint, with the only bottlenecks being external factors / delays outside of our control. By accounting for these, we could possibly see a rise in our coordination from 4.9 to 5.0 by the end of our final review.



FF2: Adaptation statistics for ITP Team Dynamics.

F.2.3 ITP Team Metrics: Relation

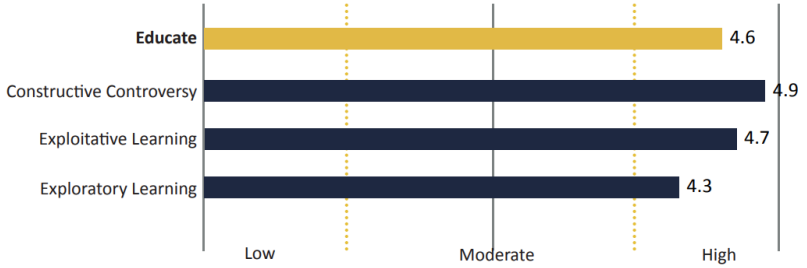
Relation stresses the importance of conflict resolving methods to complete tasks, as well as building trust within the team. This category has proven to be the most challenging for our team. We did maintain a very high score of 4.4 from our last review, but our reasonings for the score have shifted (FF3). Overall, it seems our team is more trustworthy and open to discussion (as resulted by our jumps in trust (4.7 to 4.8) and healthy conflict (3.4 to 3.7)). Despite this, however, our contribution equity has become a bit more skewed, which may have led to a drop in our lack of personal conflict (from 4.8 to 4.5 and 4.8 to 4.6 respectively). These dips are not very large however, and due to the added stress of this semester on our group, is a relatively tame drop from the previous rating. This is still a valid concern for our group to address, however, and we will be discussing the results in our upcoming sprint to identify these weak points.



FF3: Relation statistics for ITP Team Dynamics.

F.2.4 ITP Team Metrics: Education

Education in ITP metrics is the ability to learn from other members and provide each other with constructive feedback. This metric is recursive and reflective of the test itself, since the purpose is to provide each other feedback as well as rate the overall team's performance. Our "educate" score has received an increase since our last report, improving in every aspect (FF4). Our overall score has jumped from 4.4 to 4.6, backed by our increase to constructive controversy (4.7 to 4.9), exploitative learning (4.5 to 4.7) and exploratory learning (4.2 to 4.3). Since the start of the project, our team has been faced with learning many new technologies and skills to keep up with the demand of our sprint work. Our team has tackled these challenges head on, and learned many skills along the way, allowing us to effectively complete our stories, while also learning in the process. We are not afraid to give feedback or constructive criticism, asking members to review their PRs or designs before fully committing them, and we seek to learn more to improve our overall efficiency and project quality, hence our increase to exploitative and exploratory learning.



FF4: Education statistics for ITP Team Dynamics.

F.2.5 ITP Team Metrics: Summary and Conclusions

Overall, our team has been very successful at accomplishing our requirements, as we seem to be ahead of schedule. On top of this, we seem to have strengthened in just about every area, proving our success with the project is built upon our teams' overall strength. Each member's strengths play nicely off each other. We all have skills that are able to cover a weakness within the group, which increases the synergy between members. Our more clearly defined roles and ability to adapt give us each our own unique purpose on the team, while also allowing each member to be a team player to fill in the gaps and take initiative to drive innovation. Our strengths in communication, adaptation, relation and education within our group facilitate a strong team dynamic, where we can all leverage our individual strengths towards a common goal or objective. The team looks forward to closing off the project in the last chapter together!