Chemical and Inventory Management System
SoFab Inks
Team 3

Hilton Benson, Blake Hourigan, CJ Johnstone, Maggie Jackey

November 29, 2024

Contents

1 Introduction/Executive Summary

1.1 Introducing SoFab Inks

SoFab Inks is a chemical manufacturing startup that was spun-out from the University of Louisville, Conn Center for Renewable Energy Research with support from the US DoE. SoFab inks focuses on accelerating the commercialization of Perovskite Solar Cells through the development and manufacturing of functionalized inks that improve cell efficiency, reduce module cost, and enable scalable manufacturing. [1]

1.2 SoFab Inks Inventory Management Problem

SoFab Inks is doing important work in the field of solar cell technology, helping to drive humanity towards a cleaner, more energy abundant future. The problem, however, is that the team currently faces issues with managing inventory. These challenges prevent SoFab's talented team from working on the most important aspects of their work. These challenges include expending valuable energy on menial tasks like locating inventory, managing a growing number of shipments manually, scouring inventory entries found across several Google Sheets or handwritten labels to pinpoint important product information, and more.

To aid SoFab in these challenges, this semester Team 3 was tasked with developing a more efficient method of managing inventory items. To accomplish this, the team employed various existing software solutions, including database software, CRUD[1] user interface software, and solutions to containerize this software together into one package. The team also developed custom software to provide features not available in the existing software solutions.

CAC 3. Communicate effectively in a variety of professional contexts

2 System Description

The following sections describe the specifications that were formulated as a result of communication between Team 3 and the SoFab Inks team.

[1]Database term for Create, Read, Update, Delete.

2.1 Needs Assessment/System Requirements

Following assignment to this project, the team assembled and met virtually with the SoFab Inks team for brief self-introductions, an overview of the current issues facing SoFab, and to gain an initial insight of what the SoFab team was looking for in a solution to these issues.

The SoFab team described the current state of inventory management at the company which included problems such as shipments arriving to incorrect customers, an inability to pinpoint important information about products as they progressed through the manufacturing process, and difficult to track remaining volumes of chemical products. In these discussions, SoFab also emphasized the importance of a solution begin *simple and easy-to-use*. As the team is primarily constructed of chemical engineers or business people, technology was not a strength. The company also made it clear that the ability to generate item labels for internal tracking of should be a priority. These labels would allow members of their team to easily scan a barcode to view the details of an item. Finally, it was also made clear that the company also desired the ability to generate shipment labels for their products as well. These labels would differ slightly from their internal tracking label counterparts with the inclusion of chemical hazard information. This information would be included as a way to reduce harm for customers handling SoFab's chemicals upon receipt.

After this initial discussion, Team 3 began to brainstorm potential solutions to the problems presented. Immediately, 3 main pieces of software jumped out at the team as critical pieces to what would be the final product.

1. **A Database System** - In order to move the company away from the usage of Google Sheets and toward a more efficient, safe and redundant, user-friendly solution, Team 3 knew it would be necessary to select an existing database software solution. While the team was unsure of what specific solution would be chosen, it was sure that one of these solutions would be required.
2. **A User-Friendly Database Interface** - While a database solution would be an incredible improvement on its own, it would be useless to the SoFab team if there were not a simple and easy way to interact with the underlying data. Again, the team was unsure of what specific solution would be chosen, but a few requirements from discussions with SoFab were clear.
    - **Clean, Simple, Easy-to-Use** - As previously discussed, SoFab was clear that a simple and easy-to-use solution was of paramount importance for the day-to-day usage of the product.
    - **Free and Open-Source** - While this requirement was not mentioned in the initial discussions with SoFab, this requirement jumped

out as important to Team 3 because this would help to avoid incurring additional costs beyond the development cost that SoFab had already paid.

3. **Software to Generate Internal and Shipment Labels** - After discussing the need for barcode and label generation software, the team found that it would likely be necessary to build custom software to meet the needs of the client. The requirements were quite specific, and would not be available in any existing commercial product. These requirements certainly would not be made available in any *free and open-source* software.

2.2 Initial System Specification

(External design document) EAC 1. Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics and CAC 1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions)

2.3 Final Specifications

(finalized internal design document) EAC 1. Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics and CAC 1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions and CAC 6. Apply computer science theory and software development fundamentals to produce computing-based solutions)

2.4 System Diagrams

Detail all interfaces between the environment and the components EAC 2. Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors)

2.5 Hardware Overview Diagram

CAC 2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline)

2.6 Software Overview Diagram

CAC 2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline)

2.7 Economical, Technical, and Time Constraints

3 Detailed Implementation

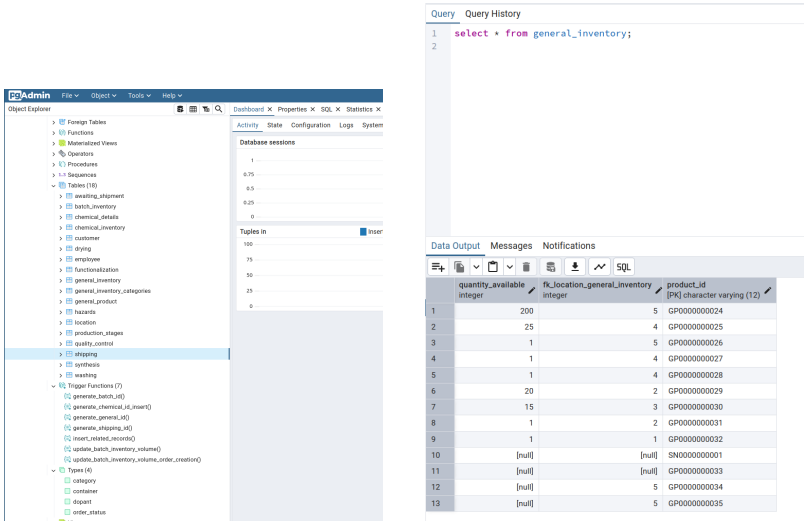
3.1 Hardware Detailed Implementation

3.2 Software Detailed Implementation

3.2.1 Docker-Compose

3.2.2 PostgreSQL

3.2.3 PgAdmin4



(a) Viewing existing tables in PgAdmin interface.

(b) Running PostgreSQL queries inside of PgAdmin4.

Figure 1

3.2.4 Budibase

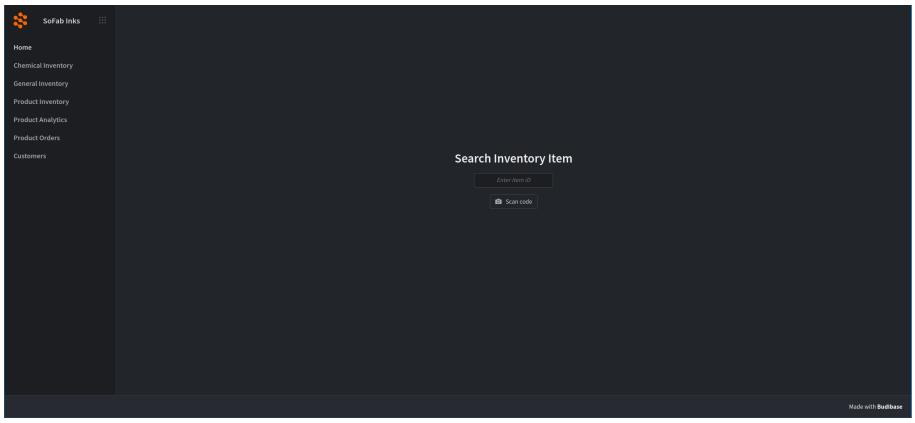


Figure 2: Budibase home page. Global search that searches all types of inventory via QR code or direct ID entry.

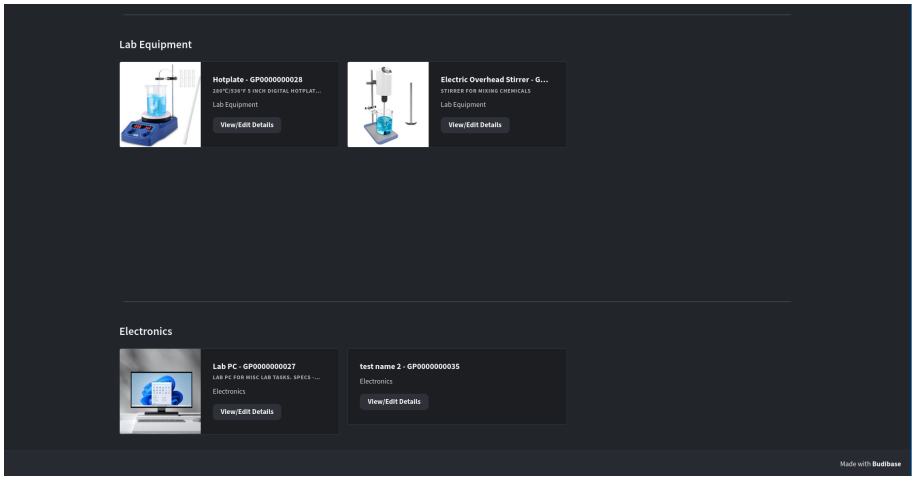


Figure 3: Budibase general inventory page. Keep track of general lab items.

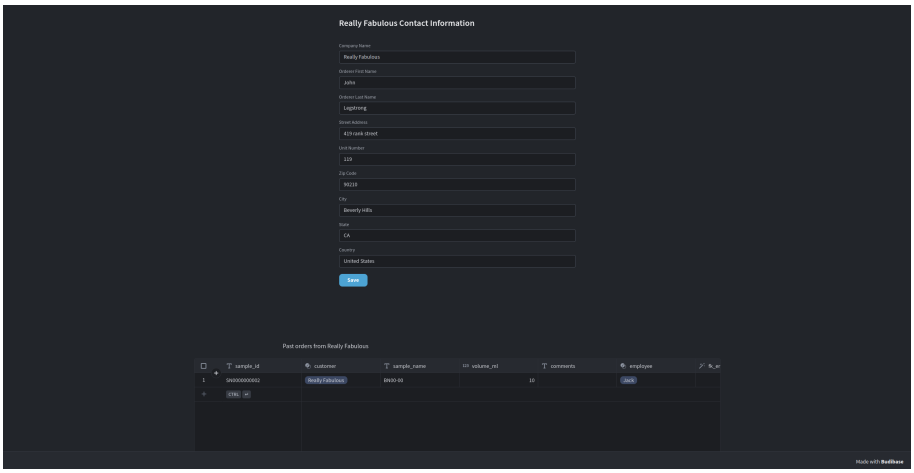


Figure 4: Keep track of customer information with Budibase. View customer information or view all orders originating from a customer

3.2.5 Python Database Insertion/Label Generation

```
class App(Tk):
    def __init__(self, controller):
        # initializing the Tk class instance
        super().__init__()

        self.title("SoFab Inventory Managment System")

        self.style = Style(theme="darkly")

        self.controller = controller
        self.controller.set_view(self)

        self.notebook = ttk.Notebook()
        self.notebook.pack(fill="both", expand=True)
        self.notebook.bind("<<NotebookTabChanged>>", self.
            on_tab_selection)

        # filling the notebook (top tabs) with frames

        item_type_tables = self.controller.
            get_item_type_tables()
```

```
class HazardPrecautionFrame(tk.Frame):
    def __init__(self, parent, controller, warning_dict,
        images=False):
        super().__init__(parent)
```

```
def generate_checkboxes(self, images=False):
    self.checkboxes_frame = tk.Frame(self)
    self.checkboxes_frame.grid(row=0, column=0)

    for item in self.warning_items:
        if images:
            image = Image.open(item[1])
            resized_image = image.resize((100, 100), Image.
                LANCZOS)
            image = ImageTk.PhotoImage(resized_image)
            item = item[0]
        else:
            image = None
        var = tk.BooleanVar()
        checkbox = ttk.Checkbutton(
            self.checkboxes_frame,
            text=item,
            image=image,
            compound="left",
            variable=var,
            command=lambda var=var: self.parent.
                update_text_box(),
        )
        checkbox.image = image

        checkbox.pack(anchor="w", fill="x")
```

EAC 1. Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics and EAC 2. Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors, and CAC 6. Apply computer science theory and software development fundamentals to produce computing-based solutions)

4 Test/Evaluation Experimental Procedure and Analysis of Results

EAC 6. Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions)

5 Societal Impact of Project/Legal and Ethical Considerations

include legal and ethical considerations

CAC 4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles)

6 Contribution of Project to Society/Expected Effects

CAC 4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles)

7 Engineering Standards, Constraints, and Security

EAC 1. Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors)

8 Conclusions

9 Recommendations for Future Work

References

[1] SofaBinks. (n.d.). About SofaBinks. Retrieved November 29, 2024, from `https://www.sofabinks.com/about`

Appendices

A Customer Contact Information

B Data Sheets

C Additional Drawings and Diagrams

D Source Code

E Experimental and/or Simulation Test Results

F Software Installation Instructions

G User Manual

H Quotes, Including Ordering Information

I White Papers