Optimizing Roofing Estimation: A Prototype for Automation and Workflow Integration

Design Document Report

Student: Silke Wittich

Student Number: 427704

Client Company: TPC Roofing

Direct Client: Elias Foreman

Class:

Major: Game Design

Graduation Supervisor: Douwe Terluin

Second Assessor: Rik Boer

Denver, CO - United States of America

10-03-2025

Management Summary

This report documents the design and development of a prototype roofing estimation tool for TPC Roofing, addressing inefficiencies in their current Excel-based manual workflow. The project aimed to improve accuracy, efficiency, and scalability by automating material pricing and production quantity calculations while laying the groundwork for future expansions.

The research phase included interviews with estimators, competitive analysis, and literature review on UI/UX and estimation methodologies. These insights highlighted critical pain points, such as manual data entry errors, slow supplier price updates, and inefficiencies in proposal generation. Based on this, multiple conceptual solutions were explored, ultimately leading to the selection of a Combined Estimation Dashboard as the most viable option.

The prototyping phase was conducted iteratively, refining backend calculations and developing an interface structure. Key milestones included:

- Transitioning from a multi-format automation approach (PDF + Excel) to an Excel-only focus to improve data consistency.
- Building a centralized SQLite database to store and manage pricing and production data, reducing reliance on fragmented spreadsheets.
- Developing terminal-based user inputs and validation mechanisms to ensure error-free data processing.
- Creating wireframe designs for a graphical user interface (GUI), aligning with best UI/UX practices and estimator feedback.

Evaluation involved peer reviews, usability testing, and system performance validation, validating the prototype's effectiveness in reducing manual errors and improving workflow efficiency. However, feedback emphasized the need for better error handling, user-friendly onboarding, and intuitive interface navigation, which were incorporated into the next development stages.

Ultimately, this project successfully began streamlining and set the stage for future streamlined core estimation processes, proving that automation can significantly reduce estimator workload and improve bid accuracy. While the current prototype focuses on backend automation, future iterations will integrate a fully interactive GUI, enhance automation for supplier price updates, and introduce proposal generation features, making it a scalable solution for both TPC Roofing and similar companies.

Preface

This report summarizes the work of my graduation project, where I worked closely with the lead estimator of TPC Roofing to develop a prototype aimed at streamlining the roofing estimation process. I would like to extend my gratitude to my supervisor Douwe Terluin for his guidance and constructive feedback during the Community of Learners (COL) meetings throughout the project. I am also deeply thankful to the client, Elias Foreman, who provided invaluable opinions, insights into his workflow, and allowed me endless creative freedom to design this prototype.

Creating this project was a journey for me. I started from scratch learning Python. With no previous experience, I tackled challenges using my basic coding experience. Over the course of this project, I developed a functional prototype while building a strong foundation in programming and data management. This experience has not only enhanced my technical skills but also sparked a deep interest in creating software solutions focused on automation and managerial data for businesses that I will eagerly explore further in the future. I hope this report not only demonstrates the value of the current project but also highlights its potential for future development that I will continue to work on until completion.

This resubmission includes substantial revisions based on feedback from assessors, ensuring clearer alignment with the project's objectives and evaluation criteria. The key updates made to this document include:

- **Strengthened Theoretical Framework** – Expanded research on accuracy, efficiency, and scalability, explicitly linking industry literature to design decisions.
- **Improved Concept Differentiation** – Clarified how each explored concept addressed different aspects of the estimation workflow, with additional wireframes to visually represent them.
- **Explicit Intermediary Testing** – Clearly documented iterative testing across different prototyping phases, detailing small validation tests, user feedback, and refinements.
- **Expanded UI/UX Research Integration** – Summarized relevant literature on user experience principles and directly linked findings to design requirements (DRs).
- **Refined Methodology & Justifications** – Added explanations for **why** specific research methods (e.g., interviews, UI studies) were used and how they shaped the final prototype.
- **Clarified Pivot & Scope Adjustments** – Outlined how and why the project scope shifted, emphasizing the decision to focus on core estimation automation before expanding into proposal generation.

These revisions comprehensively strengthen the project, ensuring clearer alignment with the competency criteria and demonstrating significant improvements across all assessed areas.

Contents

Design Process Report

Introduction

Competitive and accurate bidding is essential in the construction industry, directly impacting a company's ability to secure work and remain profitable. Research has explored ways to optimize bidding through data analysis and automation (Gates, 1967; Drew & Skitmore 1993; Fu et al., 2002). However, many companies still rely on manual, time-consuming estimation processes prone to errors. While automated bidding solutions exist (umnov.denis, 2023; SmartBuild Systems, 2022), they often require expensive subscriptions and training, making them impractical for smaller firms. As a result, improving accuracy and efficiency in estimation remains a critical challenge, particularly for small and mid-sized contractors.

This project addresses a central problem in the construction industry: improving the estimation workflow by transitioning from a manual Excel spreadsheet-based system to a partially automated prototype. This report documents the design process and results of my project prototype for TPC Roofing, a small family-owned business (Albuquerque,NM, USA) that is looking to expand. TPC Roofing is representative of many companies in the industry: small, with limited resources, and relying on an in-house manual estimation process. This document outlines the steps taken, challenges encountered, and the final outcomes of the project.

1. Project Overview

1.1 Client

1.1.1 Client Overview

TPC Roofing is a small roofing contractor specializing in residential and commercial projects. The company currently manages three to four projects ranging from $50,000 to $2,000,000 while generating one bid per week with a 10% acceptance rate. Their estimates are critical for on-site project management and securing new work. With plans to expand into Denver, CO, the company seeks to increase estimator efficiency and streamline communication to support growth. Despite its size, TPC Roofing shares challenges common to many contracting firms, making this project relevant beyond a single company.

1.1.2 Context and Problem Statement

Contracting companies worldwide rely on accurate and competitive bidding, which depends on estimators interpreting architectural plans, supplier data, and pricing fluctuations. Research confirms that bid accuracy correlates with contractor experience (Fu & Drew, 2003), making the loss of skilled employees a significant risk. Competitive bidding is crucial across industries, including government agencies managing public funds (Liu et al., 2022). Small businesses like TPC Roofing face additional challenges, relying on outdated tools such as spreadsheets and supplier databases, which are tedious

and error-prone. Inefficiencies in bid generation delay responses, reduce competitiveness, and hinder growth. TPC Roofing exemplifies thousands of similar companies needing an integrated, flexible system to reduce manual errors, improve productivity, and sustain business expansion.

For an expanded explanation, see here.

1.1.3 Opportunity and Goals

TPC Roofing operates in a highly competitive industry where speed and accuracy in estimates determine success (Evidentiary Document, Methods 1 and 2). Their reliance on a manual, Excel-based process hinders productivity, increases errors, and limits scalability. This project aims to automate estimation, centralize data, and enhance scalability, improving efficiency and accuracy.

This project not only addresses TPC Roofing's needs but also provides a scalable model for efficiency and accuracy applicable across industries. The company's small size enables direct user feedback and hands-on testing, ensuring the system effectively resolves inefficiencies in manual estimation processes. For expanded detail and steps to achieve these goals, see here.

1.2 Stakeholders and Target Audience

1.2.1 Stakeholders

TPC Roofing Estimators and Future Employees

Estimators use draft diagrams, technical plans and customer discussions to generate estimates. This project will have a direct impact on this group and future company growth by reducing delays and errors with streamlined workflow and data integration.

TPC Roofing Management

Management is responsible for operations and profitability. The use of effective tools will make planning and decision-making easier, reduce costs, and increase competitiveness in the bidding process, resulting in an increase in the number of bids accepted. Importantly, better workflow and more accurate estimates will be beneficial to project management across multiple subcontractors.

1.2.2 Target Audience

The target audience for this project are the TPC Roofing estimators and project manager. The current estimation system was designed by this team. The new system was customized to reflect the current work methods, reduce time for data entry, and improve accuracy and scalability as the company expands. (For workflow and challenges, see interviews).

1.3 Design Challenge

The plan involves developing features that integrate the necessary calculations in TPC Roofing's workflow, consequently increasing accuracy and efficiency. This would be built wholly with the existing processes of the company in mind, as well as keeping it user-friendly to estimators. The interaction for system design can be used as a model for generating future systems for other smaller companies.

1.3.1 Main Research Question

How can the manual estimation workflow at TPC Roofing be streamlined and partially automated to improve efficiency, accuracy, and scalability for the company?

1.3.2 Introduction to Key Aspects: Accuracy, Efficiency, Scalability

A well-designed estimation tool must prioritize accuracy to prevent costly errors, efficiency to streamline workflows, and scalability to support future growth. These aspects ensure reliability, reduce manual effort, and position TPC Roofing for long-term success.

1.4 Key Terms and Research Questions

ID	Research Question / Key Term	Method
K1	**Workflow Automation**	
SQ1	What are the current challenges in TPC Roofing's estimation workflow?	Client Interviews
SQ2	How can the automation of material pricing and production quantity estimation increase accuracy and efficiency?	Client Interviews Expert Interview Competitive Analysis
SQ3	How can the Python prototype replicate spreadsheet calculations for estimators?	Expert Interview Prototyping Testing
K2	**User-Centered Design**	
SQ4	What design features can ease user interaction while increasing efficiency?	Stakeholder Feedback Expert Interview Competitive Analysis
SQ5	How can the prototype's interface align with TPC Roofing's workflow?	Expert Interview Wireframes and Client Feedback
K3	**Scalability and Future Development**	
SQ6	What features are necessary to ensure the tool's scalability for additional users?	Literature Review Expert Interview Stakeholder Feedback
SQ7	What modifications would be needed to integrate supplier price updates automatically?	Prototyping Testing

Table 1 Key Terms, Research Questions, and Their Methods. Defined by Key Terms (K) and Sub Questions (SQ).

2. Current Prototype

2.1 Prototype Description

The prototype is designed to provide the solution to the Material Pricing and Production Quantities faced by TPC Roofing that were found during the client meetings and workflow analysis (Method 1 and 2). Through automating and centralizing the Excel-based workflow it simplifies the operations and maximizes the user-friendliness. It is based on the principle of scalability as it forms a basis for further developments such as updating supplier prices and proposal generation (see initial concept plans).

2.1.1 Key Features

1. **Material Pricing Automation:**
    - A centralized SQLite database replaces multiple disconnected Excel sheets, enabling dynamic retrieval of material prices and reducing manual updates.
2. **Production Quantities Workflow:**
    - The system mirrors TPC Roofing's existing workflow, automating calculations based on user-provided blueprint measurements while ensuring outputs align with manual processes.
3. **Dynamic Pricing Logic:**
    - The prototype supports customizable pricing through:
        - Default values from the database.
        - User-defined base prices for specific cases.
        - Inputs carried over from previous materials.
    - Discounted pricing is calculated based on quantity and optional extras, providing flexibility for estimators.
4. **Save and Load Functionality:**
    - Users can save progress and reload projects, allowing continuity for projects which usually lasts a few weeks.

2.1.2 Alignment with User Needs

By automating repetitive tasks, reducing manual errors, and maintaining familiarity with existing workflows, the prototype minimizes the learning curve for TPC Roofing's estimators. This can allow users to focus on higher-value tasks such as refining estimates and managing projects rather than navigating complex and fragmented workflows.

2.2 Prototype Development Process

2.2.1 Development Process

The project aimed to reduce the workload and automate the TPC roofing estimation process. Initially, high-end features like automated price updates were considered, but feedback led to a focus on material and production pricing, the most critical pain points and a foundation for future automation.

2.2.2 Design Decisions

1. **Platform Choice:**
    - Python was chosen for its flexibility and extensive library support (pandas, sqlite3, openpyxl), ensuring efficient handling of TPC's workflows (McKinney, 2012).
    - SQLite gave a lightweight, file-based database ideal for managing material pricing and production data (Learn SQL | CodeCademy, n.d.).
2. **Backend-Driven UI Design:**
    - The backend structure defined the UI wireframe for alignment with workflow and estimator needs.
3. **User-Centric Workflow:**
    - The prototype replicates TPC Roofing's existing Excel workflows to minimize disruption and the learning curve for estimators.
4. **Scalability:**
    - This supports future company and project expansions through collaborative work and proposal generation features, for better and easier training.

For more information see Tools and Technologies.

2.2.3 Challenges and Solutions

1. **Database Structure**
    - **Challenge:** Designing a framework that balances simplicity with flexibility for future expansions.
    - **Solution:** A modular approach centered on material categories and pricing enabled easy additions while preserving clarity.
2. **Workflow Replication**
    - **Challenge:** Accurately replicating Excel calculations that simultaneously increase efficiency and accuracy.
    - **Solution:** Custom logic was implemented that mirrors manual workflows, removes extra stages of calculation, reduces errors, and ensures continuity for estimators.

3. **Error Handling**
    - **Challenge:** Managing missing inputs and invalid data without affecting usability.
    - **Solution:** Validation checks and user-friendly prompts were added to guide users and prevent calculation errors or system crashes.

2.3 Future Iterations

The prototype lays a strong foundation by improving key aspects of TPC Roofing's workflow. Next steps will expand automation, streamline Excel sheets, and coding the user-friendly interface.

2.3.1 Planned Next Steps

1. **Automating Data Flow Between Excel Sheets:**
    - **Priority:** Ensuring the calculations work and implementing second excel sheet automation.
    - **Impact:** Reduces manual data entry and errors, improving efficiency.
2. **Programming the User Interface:**
    - The UI will be coded based on wireframe designs.
    - Users will be able to:
        - Input data (e.g., blueprint measurements, material selections).
        - View real-time calculations of production quantities and costs.
        - Save, load, and manage project data seamlessly.
    - The design of the UI will align with TPC Roofing's workflow to ensure ease of use while introducing the benefits of automation.

2.3.2 Future Goals

1. **Proposal Document Integration:**
    - Once automation is complete, future iterations will generate client-ready proposal documents.
2. **Further Automation and Scalability:**
    - Future updates will:
        1. Automate supplier price updates.
        2. Explore in-app colleague collaboration.

For more details, see here.

1. Phase 1 – Research

3.1 Set-Up

Research was conducted to understand the problem context and the users' needs. To gain deeper insight into the experiences of TPC Roofing's estimators, an Empathy Map and Persona were created based on interviews and workflow analysis. These helped identify key challenges, pain points, and opportunities for automation. While a summary of these insights is incorporated into the design process, the full Empathy Map and Persona and more information on those sections can be found in the Evidentiary Document.

3.2 Methods

Methods	Evidence
1. Client Interviews	Method 1
2. Expert Interview	Method 2
3. Empathy Map	Method 3
4. Persona	Method 4
5. SWOT Analysis	Method 5
6. Competitor Analysis	Method 6
7. Good, Better, Best Practices	Method 7
8. Literature Study	Method 8
9. Moscow Method	Method 9

Table 2 Methods and Evidence

3.3 Results

3.3.1 K1 - Roofing Workflow Automation

SQ1 - What are the current challenges in TPC Roofing's estimation workflow?

The research phase uncovered several critical challenges in TPC Roofing's current workflow through client and expert interviews. Insights from Elias highlight the key pain points in the estimation workflow. When first learning to estimate, the hardest challenge was understanding how various materials fit into a roofing system, as well as interpreting blueprint details accurately. Becoming proficient in estimation took approximately 2-3 years, with improvement largely due to training sessions and discussions with manufacturer representatives. If an automated system had been available to centralize material selection and streamline blueprint interpretation, it would have significantly reduced the learning curve for him as a new estimator.

1. **Manual Data Entry and Calculations**
    - Estimators spend 8–20 hours per project transferring measurements and data across multiple spreadsheets, a process prone to errors.
      (Client Interview 1)
2. **Tedious Pricing Updates**
    - Products must be manually updated for products from multiple suppliers. Supplier pricing updates require repetitive manual adjustments across numerous Excel

sheets, leading to frequent discrepancies.
(Expert Interview.)

3. **Proposal Preparation Delays**
    - Proposal generation involves manual input into Word templates, adding time and increasing risks of inconsistencies in bid documents.
      (Client Interview 2)
4. **Lack of Centralization**
    - The reliance on disconnected Excel sheets for production quantities, material pricing, and supplier updates creates inefficiencies and collaboration challenges.

SQ2* - How can the automation of material pricing and production quantity estimation increase accuracy and efficiency*?

Analysis of the workflow and industry practices identified several automation opportunities to address key inefficiencies:

1. **Automating Material Calculations:**
    - The current prototype automates material pricing and production quantity calculations using Python, reducing manual input and ensuring formula-driven accuracy.
2. **Integration of Current Second Excel Sheet:**
    - Currently the company uses two Excel sheets. The prototype automates the process of transferring material quantities from the primary calculation sheet into a secondary sheet, where total costs are calculated. This feature will address inefficiencies in the current workflow by reducing repetitive tasks. With this, the task will be automated so that the second calculation sheet is no longer necessary. (Project Framework 1.)
3. **Streamlining Supplier Price Updates:**
    - While not yet implemented, the proposed automation for supplier price updates would scan data from new supplier files and update pricing automatically. This feature was identified as a key need through competitor analysis, where similar features were either missing or underdeveloped in existing tools like RoofSnap and Roofr.
4. **Centralizing Data Management:**
    - By integrating all workflow steps into a centralized backend, the prototype sets the foundation for future scalability, including multi-user functionality and collaborative tools.

SQ3 - How can the Python prototype replicate Excel calculations for estimators?

The Python prototype replicates the manual calculations performed in Excel by leveraging libraries for Excel compatibility. These tools allow the prototype to mirror existing workflows, ensuring a seamless transition for estimators while automating repetitive tasks.

Key Approaches:

1. **Formula Replication:**
    - Excel formulas used for material pricing and production quantities were converted into Python functions. This ensures that calculations remain consistent with TPC Roofing's current methods.
    - Example: Calculations for discounted pricing and totals were integrated directly into Python's logic to reduce manual entry.
2. **Database Integration:**
    - The centralized SQLite database manages material data, replacing multiple Excel and PDF sheets. By integrating pricing and production data into the backend, the system ensures greater accuracy and scalability.
3. **Validation of Outputs:**
    - Outputs generated by the Python prototype were tested against Excel's results to ensure accuracy and reliability. This iterative testing process ensures that estimators can trust the automated calculations.
      (Evaluation Phase)

Future Outlook

At the heart of the automation process is the first step of replicating Excel calculations. Future actions including automated material integration and supplier updates will directly benefit the TPC Roofing Company by allowing them to eliminate the manual data input, diminish the errors, and increase the efficiency in the estimation work.

3.3.2 K2 – User-Centered Design

SQ4 - What design features can ease user interaction while increasing efficiency*?*

The main focus of the research phase was to implement a user-centered design that fits with TPC Roofing's workflows. The empathy map and persona shared insights that formed the basis of the key user needs:

1. **Familiarity with Current Processes:**
    - Estimators are more comfortable using devices that resemble their Excel workflows, so they can easily adapt to the new environment. The prototype is developed in a way that it follows the same logic and structure as before. (Empathy Map)

2. **Minimizing User Errors:**
    - The frustrating impact of human errors is contended with through the built-in automated calculations and input validation features. (Persona). Input validation is important in web applications. This is the enforcement of constraints that all inputs need to satisfy before being used for generating final outputs (Liu and Kuan Tan, 2008; Banach, 2022).
3. **Streamlined Workflow:**
    - The wireframe presents streamlined consolidated steps as a single tool that would have inputs and calculations centralized.
      (Wireframe)
4. **Efficiency-Driven Features:**
    - Outlining the features like project file storage and retrieval, and easy entry formats that are prioritized first for the sake of eliminating repetitive work and improving daily productivity.

SQ5 - How can the prototype's interface align with TPC Roofing's workflow?

Research into UI/UX design principles and insights from TPC Roofing estimators informed the interface design, ensuring alignment with their workflow:

1. **Wireframe Development:**
    - The prototype's wireframe was influenced by the backend logic, which replicates Excel calculations. This ensures that the interface not only mirrors existing processes but also integrates seamlessly with the backend.
      (Wireframe)
2. **Alignment with Pain Points:**
    - Estimators noted the inefficiencies of navigating multiple Excel sheets. The wireframe addresses this by centralizing inputs, calculations, and outputs in one cohesive interface.
      (Client Interview)
3. **Simplicity and Usability:**
    - Literature on UI/UX design principles guided the creation of a clean, uncluttered interface. Features like dropdown menus for material selection and automated pricing updates were prioritized to enhance usability without overcomplicating the design.
      (Literature Study)
4. **Future Scalability:**

- The interface design incorporates placeholders for future features, such as automated supplier price updates and proposal generation. This ensures that the tool remains adaptable as new functionalities are added.

Future Outlook

The user-centered design approach ensures that the prototype meets the immediate needs of TPC Roofing estimators while remaining flexible for future growth. The next steps involve coding the interface based on the wireframe, with a focus on integrating the backend logic and adding features like multi-user support and collaborative tools.

3.3.3 K3 – Scalability and Future Development

SQ6 - *What features are necessary to ensure the tool's scalability for additional users?*

The research phase identified scalability as a key consideration for TPC Roofing's prototype. Through client interviews, competitor analysis, and user-centered design principles, several features were identified as essential for future scalability:

1. **Multi-User Support:**
    - To support additional estimators or team members, the prototype will need features like user accounts, permissions, and individualized inputs. This will allow multiple users to work simultaneously while maintaining data integrity.
      (Client Interview and Competitor Analysis)
2. **Customizable Workflows:**
    - As TPC Roofing grows, the prototype should allow for adaptable workflows that can cater to various project types or user preferences. This includes features like modifiable templates and dynamic project settings.
      (Good, Better, Best Practices)
3. **Cloud-Based Collaboration:**
    - While the current prototype uses a local SQLite database, transitioning to a cloud-based system would enable real-time collaboration and remote access for team members working in different locations.
      (Expert Interview)
4. **Data Security:**
    - As the tool expands to accommodate more users, implementing secure access controls and encrypted data storage will be critical for protecting sensitive project information.

SQ7 - What modifications would be needed to integrate supplier price updates automatically?

Supplier price updates were identified as a major pain point during client interviews, and automating this process is a key opportunity for future development. The following modifications would be required to implement this feature:

1. **Parsing Supplier Files:**
    - The system would need the capability to parse files from suppliers (e.g., CSV, Excel, PDF) and identify updates to material pricing. Libraries like pandas and openpyxl are already integrated, providing a foundation for this functionality. (Rapid Prototype)
2. **Database Integration for Updates:**
    - Automating supplier updates would require enhancements to the SQLite database schema to track price changes and flag discrepancies for review.
3. **Color-Coded Indicators:**
    - Using color-coded visual indicators for updated, unchanged, or missing items would streamline the review process, as suggested in competitor analysis of similar tools.

Future Outlook

The prototype has been designed with scalability in mind, laying the groundwork for future enhancements like multi-user support and supplier integration. These features will ensure the tool grows with TPC Roofing's needs, providing greater efficiency, collaboration, and adaptability over time.

3.4 Design Requirements

Client interviews, user research, and competitor analysis were the design requirements' bases, and the empathy map and persona insights highlight user needs like replicating workflows, reducing manual tasks, and preventing errors. Apart from the changes from the previous iterations, the supplier price updates, and the proposal generation that are now focused only on the material price and production quantity of the supplier are addressed directly to the immediate pains, thereby scalability is ensured. The adjustments mentioned are in the demonstrative document and also in the main report.

Client Requirements	
CR1	The tool should automate calculations for material pricing and production quantities.
CR2	The tool should increase accuracy and efficiency by reducing manual errors and repetitive tasks by creating a single user interface.

CR3	The system should allow estimators to save and reload project data for seamless continuity.
CR4	The tool must be scalable to support additional future workflows, such as supplier price updates and proposal generation.
User Design Requirements	
UR1	The interface must align with the estimators' current Excel-based workflows to minimize the learning curve.
UR2	As a user, I want the system to provide clear feedback and prevent incorrect inputs to ensure accurate estimates (Liu and Kuan Tan, 2008; Banach, 2022).
UR3	As a user, I want the system to provide real-time calculations to ensure efficient decision-making.
UR4	As a user, I want the ability to input and update material pricing manually for specific cases.
Qualitative Design Requirements	
QR1	The system should prioritize ease of use by following core UI/UX principles (e.g., visibility, minimal clutter, clear visual cues).
QR2	The system should perform reliably, processing large data sets without noticeable lag.
QR3	The tool must maintain data consistency across all calculations to build user trust.
QR4	The design must remain flexible for future features like cloud integration or multi-user support.
Functional Design Requirements	
FR1	The prototype should automate material pricing calculations based on user inputs and database data.
FR2	The system must integrate with existing Excel workflows, supporting seamless transitions between old and new processes.
FR3	The database must support dynamic updates, such as material pricing changes or new material entries.
FR4	The system should facilitate error prevention through validation mechanisms for inputs and outputs.

Table 3 Design Requirements

Client interviews, user research, and competitor analysis were the most influential factors in shaping the design requirements. Insights from the empathy map and persona stressing the need for familiar workflows, less manual work, and error prevention were the basis for these. Previous iterations, such as supplier price updates and proposal generation, were modified for practicality, thus limiting the field to material prices and production amounts. This method not only provides a solution for existing discomforts but also creates a flexible structure for the development of scalability; these details are provided in the evidential document. For more details, see Project Framework

4. Phase 2 – Ideation and Conceptualization

4.1 Set-Up

Research, client feedback, and features were aligned by the conceptualization phase which was the guiding document for the project. I considered several concept ideas to improve the efficiency and accuracy of the estimation process at TPC. These ideas were generated with brainstorming techniques, the SCAMPER method, and stakeholder feedback.

4.2 Methods

During this period, the methods concentrated on features prioritization, tackling workflow pain points, creating sketch interfaces, and the incorporation of stakeholder feedback for the proper alignment with client needs.

Table of Methods:

Method	Evidence
MoSCoW Method	Method 9
Feature Brainstorming	Method 10
SCAMPER Method	Method 13
Wireframe Sketching	Method 12
Stakeholder Feedback	Method 10

Table 4 Phase 2 Methods and Evidence

4.3 Results

4.3.1 Idea Generation

Initial Concepts:

1. **Project Management App:** One of the initial concepts explored was a Project Management App, an internet-based dashboard aimed at improving collaboration between TPC Roofing estimators. This idea was based on client interviews, where challenges related to disconnected workflows and manual communication were highlighted. The proposed system would have combined real-time coworker activity tracking, integrated estimation tools, and cloud-based storage to streamline the process. However, due to technical complexity, multi-user synchronization challenges, and project scope limitations, this concept was deemed too ambitious for the available timeframe. Instead, I opted to focus on a smaller, more achievable aspect of the workflow, specifically material pricing and production quantity calculations, which directly addressed the client's immediate pain points.

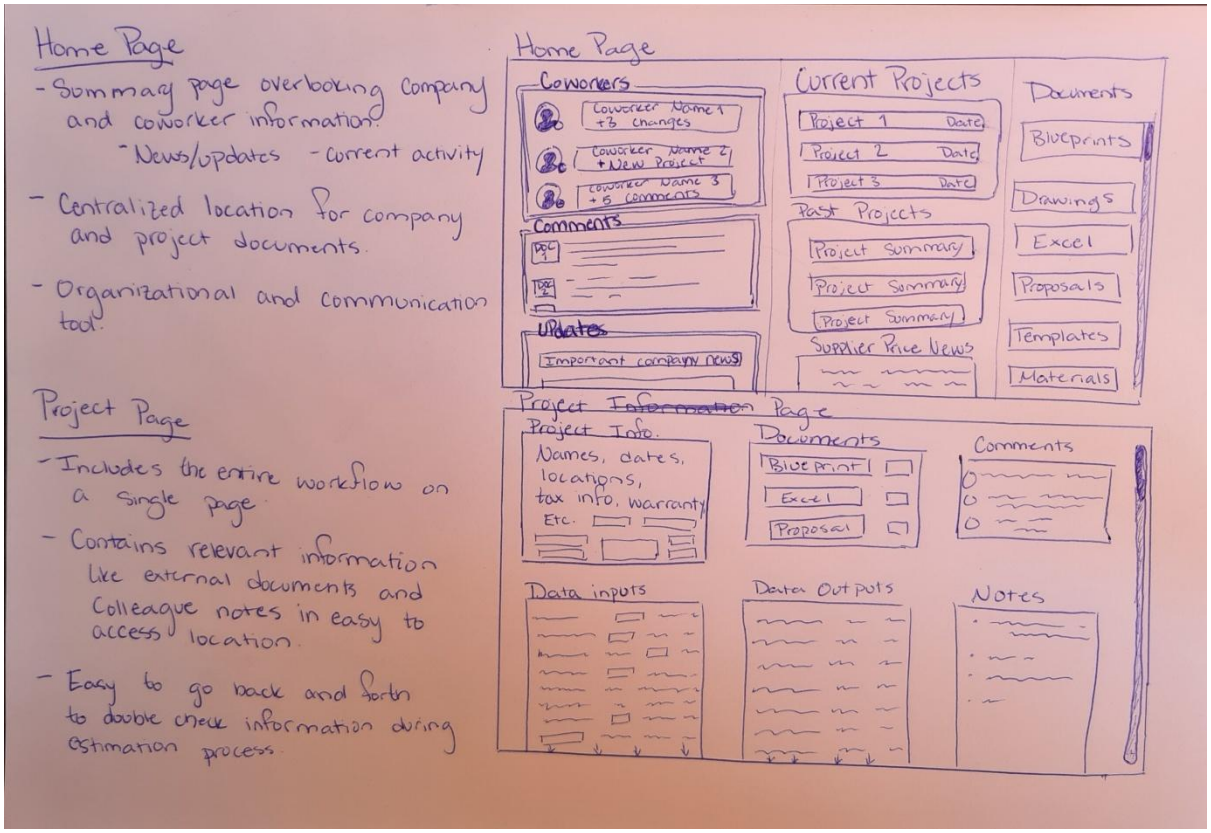


Figure 1 Sketch of Project Management App Layout.

More on this concept can be found here.

2. **Proposal Generator:** The Automated Bid Proposal Generator concept's goal to improve or all but eliminate the final step of the estimation workflow by automating bid document creation. Unlike the Project Management App, which included proposal templates as part of a broader collaborative system that would have to be filled in themselves, this concept focused exclusively on generating structured, formatted bid documents from finalized estimation data. While this would have reduced manual formatting time, it wouldn't have addressed inefficiencies in the estimation process itself. Since my project's goal was to improve both accuracy and efficiency in estimation, I decided to focus on earlier steps where errors and inefficiencies had a greater impact.

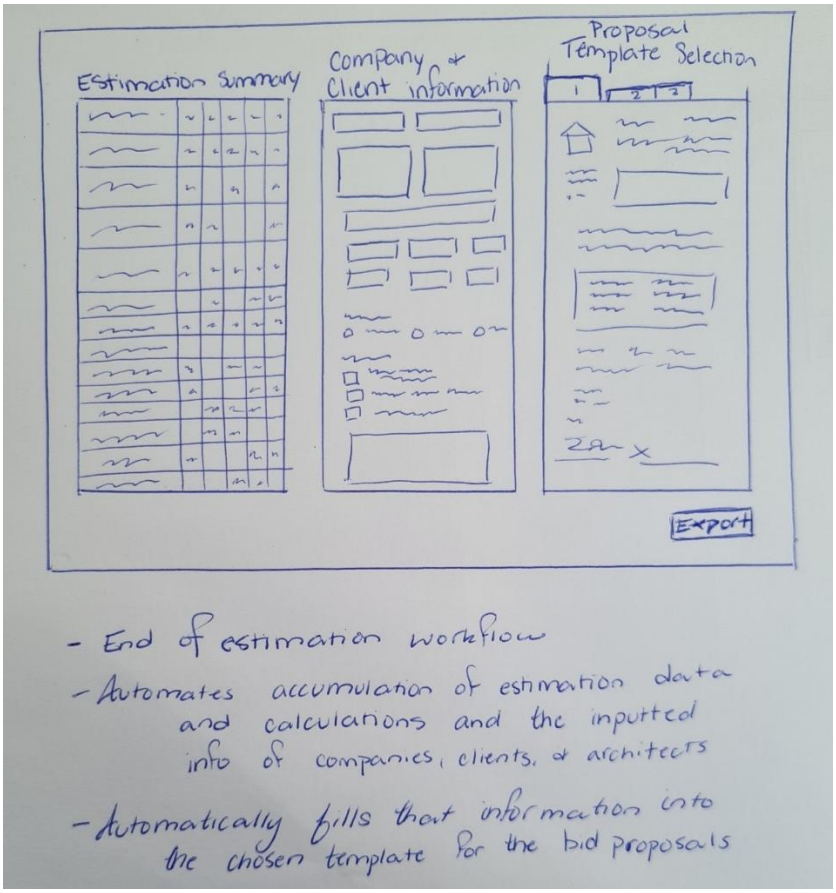


Figure 2 Sketch of Automated Bid Proposal Generator.

More can be found here.

3. **Combined Estimation Dashboard:** This concept was chosen as the final concept because it directly addresses estimation inefficiencies, unlike the previous concepts, which focused on collaboration (Project Management App) or final-stage document generation (Bid Proposal Generator). This system integrates multiple Excel sheets to automate material pricing updates, streamline production quantity calculations, and ensure supplier prices remain current. Client interviews highlighted that estimators struggled with manual price updates and fragmented workflows, making this concept the most effective solution for improving both accuracy and efficiency in the estimation process. By focusing on core estimation issues rather than auxiliary workflow improvements, this concept delivers the highest value within the project's scope.

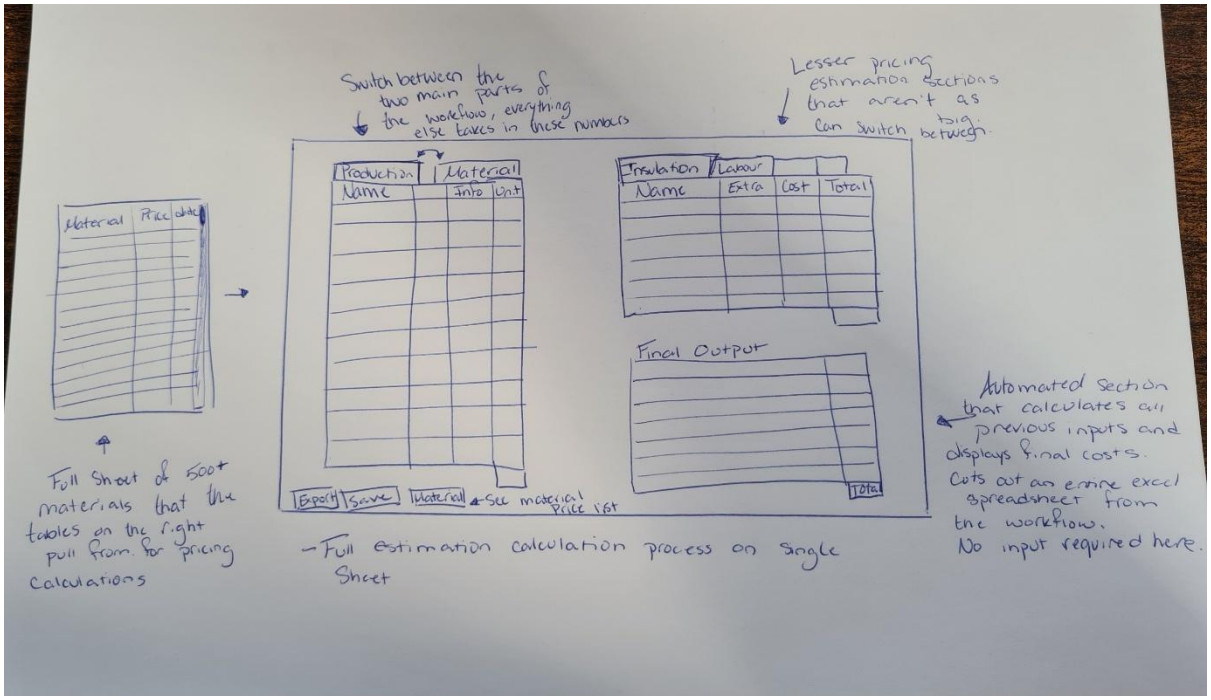


Figure 3 Sketch of the Estimation Tool.

More can be found in Brainstorming.

4.3.2 Concept Refinement

The three concepts were evaluated using the MoSCoW Method which categorizes features into Must Have, Should Have, Could Have, and Won't Have to determine feasibility and alignment with user needs. In the end, Combined Estimation Dashboard was chosen because it met all the Must Have requirements while remaining doable within the project's timeframe. The two other concepts lacked depth or required skillsets that were beyond what I could learn in the available time.

The Project Management App included collaboration and proposal templates, but these were categorized as "Should Have" rather than critical features for estimation accuracy and efficiency. The Automated Proposal Generator streamlined a later stage of the workflow but did not impact the accuracy of estimation itself, making it less impactful.

The Combined Estimation Dashboard directly addressed the core pain points identified in user research: pricing automation, production quantity calculations, and supplier price updates. All aligning with the most critical "Must Have" features. This made it the best choice for the project's goals.

Further refinements were made using the SCAMPER method to optimize the concept for feasibility, usability, and scalability. This process led to targeted adjustments, such as prioritizing pricing automation, modifying the user interface for simplicity, and refining workflow priorities. More on SCAMPER's impact can be found in the here.

Final Refinement & Wireframes

After refining the concept, the focus shifted to recreating TPC's existing workflow on a smaller scale. The prototype specifically replicates the material pricing and production quantity sections of the first Excel sheet, ensuring that core functionalities were properly implemented before automating the

second Excel sheet. This refinement allowed for better alignment with the key design decisions of accuracy, efficiency, and user-friendliness while ensuring that the core calculations and automation processes were functional.

Figures 5 and 6 below show wireframes drawn in Procreate, exploring designs for this project concept.

Wireframe Digital

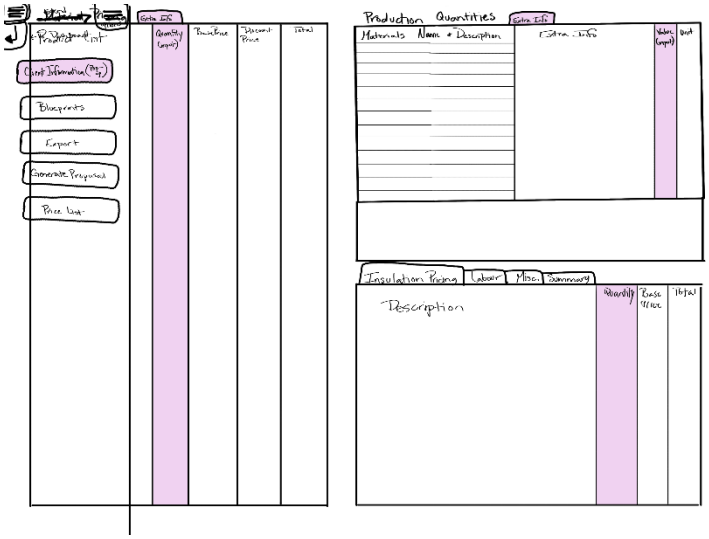


Figure 4 Digital Wireframe Sketch for the Work Page

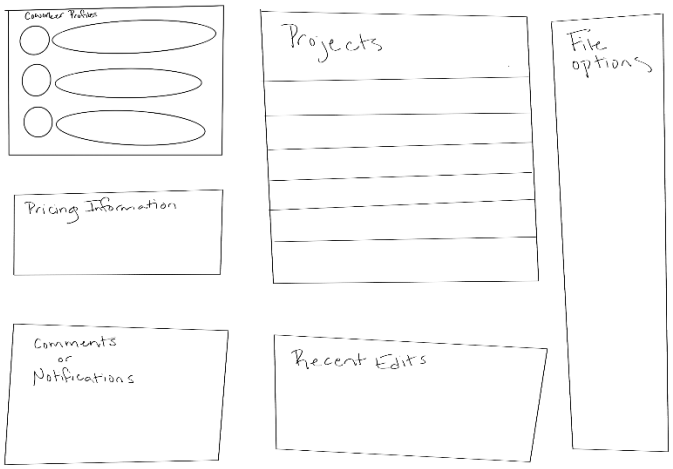


Figure 5 Digital Wireframe Sketch for the Home Page

4.3.3 Pivot and Final Concept

During the refinement process, the project underwent a significant pivot. The initial concept of a fully integrated estimation app was scaled back after further evaluating the project's scope, timeline, and feasibility. While the original vision included a broader feature set—including real-time collaboration, proposal generation, and full workflow automation—it became evident that addressing core estimation pain points first would result in a more practical and achievable solution.

The final concept prioritizes backend calculations and automation for material pricing and production quantity calculations, two of the most critical and time-consuming aspects of TPC's workflow. By automating these sections first, the system provides immediate benefits while also creating a scalable foundation for future expansion, such as proposal generation and cloud-based collaboration.

This refined approach ensures a focused, practical, and scalable solution that aligns with both TPC's needs and the project's constraints while setting the groundwork for future improvements.

For more details on the initial project plan and scope adjustments, see Project Framework 1.

5. Phase 3 – Prototyping

5.1 Set-Up

The development of a prototype was mainly based on the utilization of Python for its simplicity and libraries to construct modules that were in line with the customer's needs(Learn Python, n.d.; Maze, 2023).

5.2 Iteration 1: Multi-Format Automation (PDF + Excel)

5.2.1 Focus

The first iteration aimed to automate data extraction and integration from both PDF and Excel price sheets into an SQL database. This broad scope was chosen to address the diverse formats used by TPC Roofing and streamline their data workflows. Python was selected for its robust library support, particularly PDFPlumber for parsing PDF files and pandas for handling Excel data. For more, see Rapid Prototyping

5.2.2 Challenges

1. **PDF Parsing Issues:**
    - Despite initial success extracting data using PDFPlumber, significant challenges arose. Tables were misaligned, rows were incorrectly merged, and the parsing often resulted in data misclassification.
    - Extensive debugging and testing revealed that the inconsistencies in PDF formatting made this approach unreliable for the prototype's goal.
2. **Complexity of Handling Multiple Formats:**
    - Coding for the concurrent handling of both PDF and Excel files increased the complexity of the prototype. Edge cases of missing fields and different table layouts not only made the difficulty greater but also were an added factor.

5.2.3 Process

1. **PDF Parsing:**

- Before extracting cell data, first, the prototypes were built up using PDFPlumber.
- Tested small sections of the PDF file like rows were conducted to enclose the inconsistent parsing logic.
- Debugging the output by printing extracted data was done to find the errors, but there were still some inconsistencies.

```
  warn(msg)
Filtered Data read from Excel:   Unnamed: 0    TPO 80-mil Parapet Unnamed: 2 Unnamed: 3        0 Unnamed: 5  0.1 Unnamed: 7 Unnamed: 8 Unnamed: 9
0        NaN        TPO Coverstrip        NaN     100 LF         0     414.05    0          C        NaN        NaN
1        NaN    80-Mil Detail Roll        NaN       Each         0        NaN    0        NaN        NaN        NaN
2        NaN          Walkpath 24"        NaN       Each         0     724.16    0          C        NaN        NaN
3        NaN                   NaN        Per      Incl?  Quantity    $ / Qty  NaN        NaN        NaN        NaN
4        NaN  Flex Fast A&B w/ Gun        NaN        NaN         0    1502.16    0        NaN        NaN        NaN
Exception in Tkinter callback
```

Figure 6 Incorrectly Parsed Material Data

2. **Excel File Parsing:**

- Along with PDF parsing, the code for reading and processing Excel files was created using pandas
- The major objective was to parse column headers, filter irrelevant rows, and rename fields to match the database.
- The first trials included parsing one row and inputting it into an SQL database for workflow validation.

3. **Database Integration:**

- An SQL database was created for the storage of material pricing data.
- Both new entries and updates concerning product descriptions were handled in the code.
- Tested inserting and updating a small number of records to ensure the database queries worked as intended.

For more information see Prototyping.

Intermediary Test 1: PDF Data Extraction Validation

- **Setup:** Used PDFPlumber to extract material pricing data from a sample supplier PDF.
- **Results:** Parsing inconsistencies arose, causing misalignment in row extraction.
- **Analysis:** The extracted data did not consistently match the original table structure.
- **Steps Taken:** Additional debugging attempts were made, but inconsistencies persisted, leading to the decision to pivot towards an Excel-only focus in Iteration 2.

Intermediary Test 2: Excel Formula Accuracy Test

- **Setup:** Converted a section of TPC Roofing's manual pricing calculations from Excel formulas into Python functions and tested sample data.
- **Results:** Initial tests showed that the system successfully replicated calculations, but some formulas produced slight discrepancies due to rounding errors.
- **Analysis:** The errors stemmed from differences in how Excel and Python handle floating-point precision.
- **Steps Taken:** Adjusted calculations by implementing decimal-based rounding to ensure results matched Excel's outputs.

5.2.4 Outcome

- **Key Insights:**
    - The limitations of PDF parsing led to the decision to pivot to Excel-only workflows for subsequent iterations.
    - The Excel parsing functionality proved more reliable and laid the foundation for material pricing automation.

- **Prototype Deliverables:**
    - Laid the groundwork for delivering a functional prototype capable of reading Excel files, cleaning data, and integrating it into an SQL database.

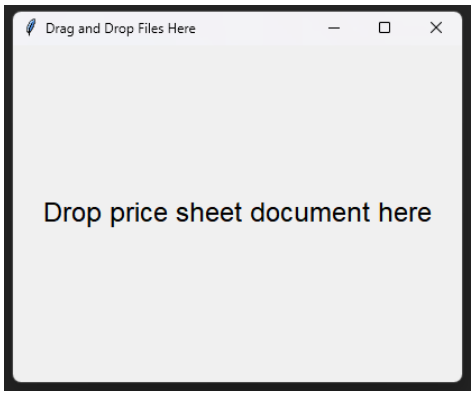


Figure 7 Drag and Drop Interface

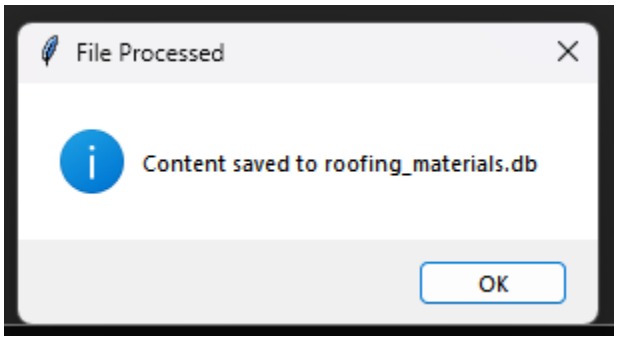


Figure 8 Successful File Process Interface

5.3 Iteration 2: Excel-Only Focus

5.3.1 Focus

This iteration shifted to focus exclusively on handling Excel files for parsing and SQL integration, allowing for a narrower scope and enabled more targeted development of critical workflows, such as material pricing and production quantities. The goal was to create a reliable prototype capable of automating these workflows while minimizing errors.

5.3.2 Challenges

1. **Refining Excel Parsing Logic:**
    - Adjusting parsing logic to handle edge cases, such as skipping irrelevant rows, renaming columns, and managing null values in the data.
    - Ensuring that all parsed data adhered to the database schema for consistency.
2. **Database Updates:**
    - Writing code to manage both new data entries and updates to existing records.

- Ensuring that updated prices replaced outdated ones without duplicating entries in the SQL database.

5.3.3 Process

- **Excel File Parsing:**
  - Expanded on the pandas logic from Iteration 1 to handle larger datasets and more diverse Excel file formats.
  - Tested parsing logic incrementally by extracting a single row, cleaning the data, and verifying the output.
  - Introduced logic for filtering irrelevant rows, filling missing values, and renaming columns to match database fields.
- **SQL Database Integration:**
  - Enhanced SQL queries to handle both new entries and updates efficiently.
  - Focused on testing scenarios where prices needed to be updated while retaining historical records of material descriptions.
  - Conducted small-scale tests to ensure accurate insertion and updating of records.

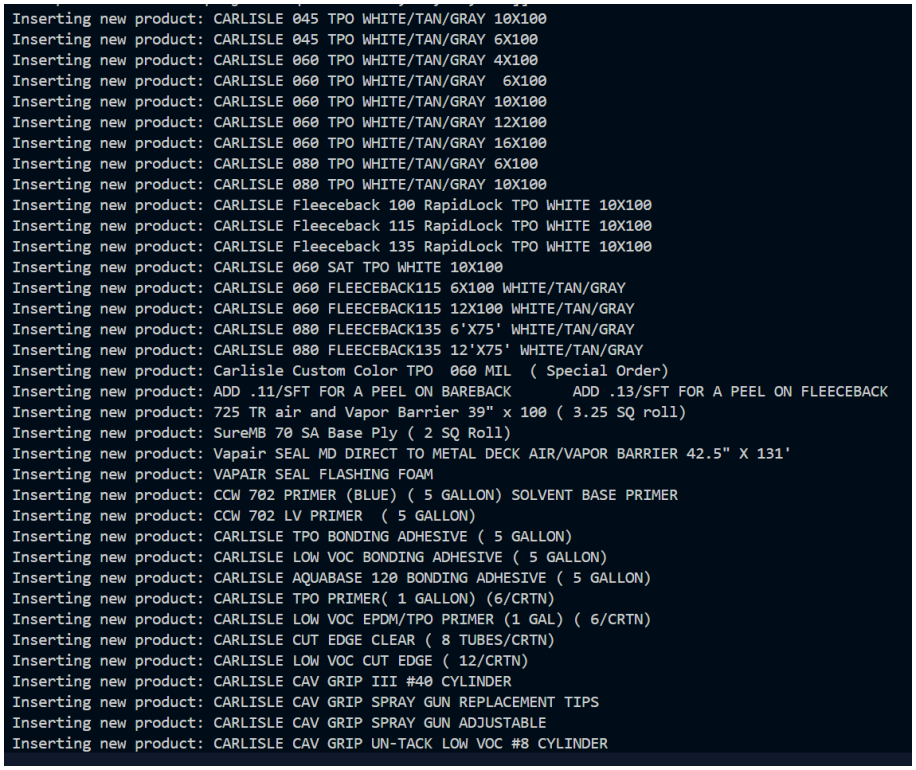
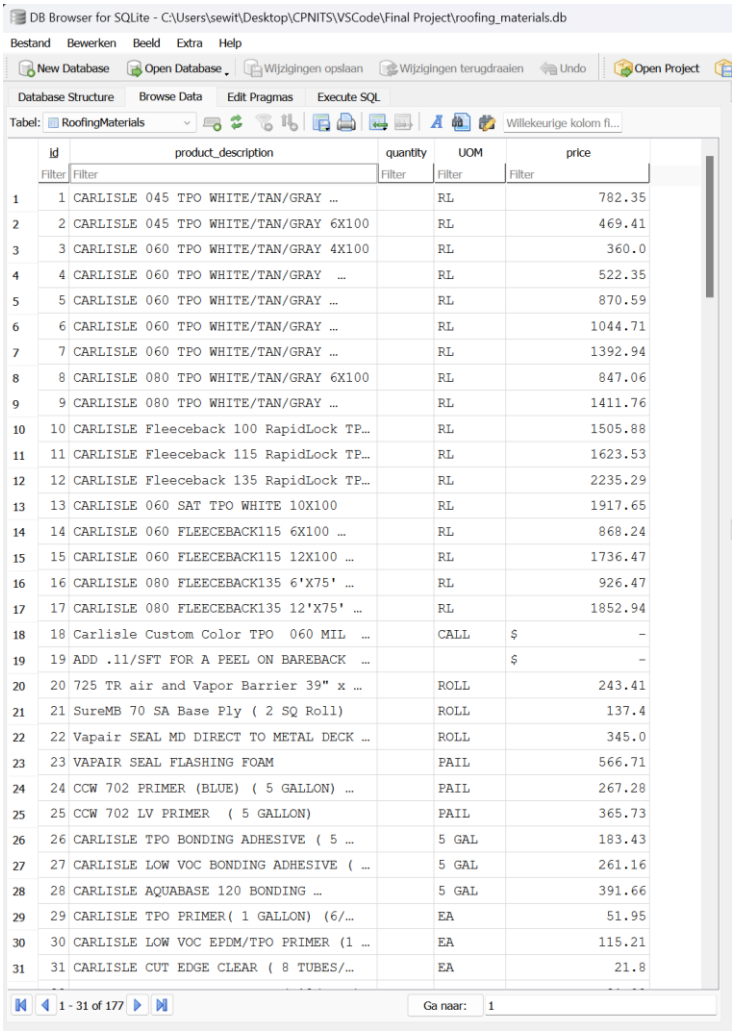


Figure 9 Terminal Update of SQL Database Inserts



	Id	product_description	quantity	UOM	price
	Filter	Filter	Filter	Filter	Filter
1	1	CARLISLE 045 TPO WHITE/TAN/GRAY …		RL	782.35
2	2	CARLISLE 045 TPO WHITE/TAN/GRAY 6X100		RL	469.41
3	3	CARLISLE 060 TPO WHITE/TAN/GRAY 4X100		RL	360.0
4	4	CARLISLE 060 TPO WHITE/TAN/GRAY …		RL	522.35
5	5	CARLISLE 060 TPO WHITE/TAN/GRAY …		RL	870.59
6	6	CARLISLE 060 TPO WHITE/TAN/GRAY …		RL	1044.71
7	7	CARLISLE 060 TPO WHITE/TAN/GRAY …		RL	1392.94
8	8	CARLISLE 080 TPO WHITE/TAN/GRAY 6X100		RL	847.06
9	9	CARLISLE 080 TPO WHITE/TAN/GRAY …		RL	1411.76
10	10	CARLISLE Fleeceback 100 RapidLock TP…		RL	1505.88
11	11	CARLISLE Fleeceback 115 RapidLock TP…		RL	1623.53
12	12	CARLISLE Fleeceback 135 RapidLock TP…		RL	2235.29
13	13	CARLISLE 060 SAT TPO WHITE 10X100		RL	1917.65
14	14	CARLISLE 060 FLEECEBACK115 6X100 …		RL	868.24
15	15	CARLISLE 060 FLEECEBACK115 12X100 …		RL	1736.47
16	16	CARLISLE 080 FLEECEBACK135 6'X75' …		RL	926.47
17	17	CARLISLE 080 FLEECEBACK135 12'X75' …		RL	1852.94
18	18	Carlisle Custom Color TPO 060 MIL …		CALL	$ -
19	19	ADD .11/SFT FOR A PEEL ON BAREBACK …			$ -
20	20	725 TR air and Vapor Barrier 39" x …		ROLL	243.41
21	21	SureMB 70 SA Base Ply (2 SQ Roll)		ROLL	137.4
22	22	Vapair SEAL MD DIRECT TO METAL DECK …		ROLL	345.0
23	23	VAPAIR SEAL FLASHING FOAM		PAIL	566.71
24	24	CCW 702 PRIMER (BLUE) (5 GALLON) …		PAIL	267.28
25	25	CCW 702 LV PRIMER (5 GALLON)		PAIL	365.73
26	26	CARLISLE TPO BONDING ADHESIVE (5 …		5 GAL	183.43
27	27	CARLISLE LOW VOC BONDING ADHESIVE (…		5 GAL	261.16
28	28	CARLISLE AQUABASE 120 BONDING …		5 GAL	391.66
29	29	CARLISLE TPO PRIMER(1 GALLON) (6/…		EA	51.95
30	30	CARLISLE LOW VOC EPDM/TPO PRIMER (1 …		EA	115.21
31	31	CARLISLE CUT EDGE CLEAR (8 TUBES/…		EA	21.8

Figure 10 SQL Database Table

- **Error Handling:**
    - Prototyped error messages for unsupported file types or invalid inputs.
    - Iteratively tested edge cases to ensure robust handling of unexpected scenarios.

Intermediary Test 3: Data Integrity Check for SQL Integration

- **Setup:** Tested the integration of parsed Excel data into the SQLite database.
- **Results:** Some records duplicated upon multiple uploads due to missing unique constraints.
- **Analysis:** The database structure lacked primary key constraints that prevented redundant entries.
- **Steps Taken:** Implemented a duplicate-checking mechanism to ensure that only new or modified records were updated.

5.3.4 Outcome

1. **Functional Prototype:**
    - Successfully created a robust mechanism for reading Excel files and incorporating them into the SQL database.
    - Handled the continuity of data extraction to database updating without any trouble, thus solving the major issues in TPC Roofing's workflow.
2. **Logic and Usability Improved:**
    - Standardized the Excel parsing logic for handling a vast range of data formats and special cases.
    - Improved database queries which allowed updating data faster and more consistently.
3. **Foundation for Future Workflows:**
    - These workflows that were built in this iteration are to be taken as the fundamental groundwork for the inclusion of user-interface functionalities in rest iterations..

See Iteration 1 & 2 , to view the full code.

5.4 Iteration 3: Placeholder Interface and Backend Refinement

5.4.1 Focus

Iteration 3 focused on refining backend calculations for material pricing and production quantities while introducing placeholder interface elements to validate workflows. The aim was to ensure backend functionality aligned with TPC Roofing's Excel workflows and to prepare for future integration with a graphical user interface.

5.4.2 Challenges

1. **Backend Logic Alignment:**
    - Replicating Excel workflows in Python required careful testing of individual calculations to ensure accuracy.
    - Managing dependencies between dropdown selections, user inputs, and backend calculations posed additional complexity.
2. **Interface Usability:**
    - Placeholder elements like dropdown menus were tested to validate how user inputs impacted database queries and calculations.
3. **Error Handling:**
    - Ensuring clear and actionable error messages for invalid inputs.

5.4.3 Process

Backend Refinement:

- Single-row tests replicated individual Excel formulas, ensuring seamless transitions from manual workflows to Python-based automation.
- Focused on calculations for production quantities and pricing to confirm backend accuracy.

```
measurements_data = [
    {"description": "Total Roof Area", "sub_description": " - Tear-Off", "is_manual": True, "unit": "SQ FT"},
    {"description": "Total Roof Area", "sub_description": " - Insulation", "formula": "input_store['Total Roof Area - Tear-Off']", "unit": "SQ FT"},
    {"description": "Total Roof Area", "sub_description": " - Membrane", "formula": "input_store['Total Roof Area - Tear-Off']", "unit": "SQ FT"},
```

Figure 11 Single Row of Code Asking User Input

```
# Create objects and collect inputs
production_quantities = []
previous_value = None
for data in measurements_data:
    pq = ProductionQuantity(
        data.get('description', ''),
        data.get("sub_description", ""),
        data.get("is_manual", False),
        data.get("extra_input", ""),
        data.get("value", 0),
        data.get("unit", ""),
        data.get("formula", None),
        data.get("is_editable", False),
        data.get("unit_dropdown", False),
    )
    pq.set_value(input_store, previous_value, available_units)
    production_quantities.append(pq)
    previous_value = pq.value


# Save project data to JSON file
save_to_json(project_file, input_store)

# Test that it works
print("\nFinal Results:")
print(f"{'=' * 20} Production Quantities {'=' * 20}")
print("Final Input Store Values:")
for key, value in input_store.items():
    print(f"{key}: {value}")

for pq in production_quantities:
    print(f"{pq.description}{pq.sub_description}: {pq.value} {pq.unit}")
print("=" * 40)
```

Figure 12 Code Collecting Inputs and Saving to JSON

```
Do you want to load an existing project or start a new one? (load/new): new
Enter file name: test12
Starting a new project.
Enter value for Total Roof Area - Tear-Off: 3200
Set value for Total Roof Area - Insulation: 3200.0
Set value for Total Roof Area - Membrane: 3200.0
```

Figure 13 Terminal Output of Single Row Input

Dropdown Menu Prototyping:

- Developed and tested dropdown menus for selecting material types or categories.
- Validated that dropdown selections dynamically updated backend queries and calculations.

```
terials_data = [
{
    "description": "Membrane Type Material 1",  # User selects from Membrane Type category in SQL
    "category": "Membrane Type",  # Category in the SQL database
    "material_ids": [1, 2, 7, 8],  # Restrict to material IDs 1-4
    "sub_category": "Field & Seams",  # Sub-description
    "extra_input": True,  # Indicates user can adjust the extra input value
    "extra_input_formula": "selected_material[2]",  # Default: Price from SQL database
    "quantity_formula": "(input_store['Total Roof Area - Membrane'] + (input_store['LN FT of Field Seam - Sheet Width (FT)'] / 2)) * 1.05",  # Formula for Quantity
    "is_manual_base_price": True,  # User inputs the base price
    "base_price_reference": None, #No reference for this material
    "discount_price": 0,  # Default user input
    "discount_price_input": True, #prompt for discount price input
    "total_formula": "(quantity * base_price) + discount_price"  # Formula for Total
},
```

Figure 14 Single Row Handling Material Pricing

```
for data in materials_data:
    if data["category"]:  # Only fetch materials if a category is defined
        print(f"Fetching materials for category '{data['category']}'...")

        # Fetch materials from SQL database
        available_materials = fetch_materials_by_category(db_file, data['category'])

        if not available_materials:
            print(f"No materials found for category '{data['category']}'.")
            continue

        # Filter materials based on IDs
        if "material_ids" in data:
            print(f"Filtering materials with IDs {data['material_ids']}...")
            available_materials = [item for item in available_materials if item[0] in data["material_ids"]]

        if not available_materials:
            print(f"No materials found with IDs {data.get('material_ids')} in category '{data['category']}'.")
            continue

        # Automatically select if there's only one material, otherwise prompt the user
        if len(available_materials) == 1:
            selected_material = available_materials[0]
            print(f"Automatically selected material: {selected_material[1]}")
        else:
            # Display drop-down options
            print("Available Materials:")
            for idx, item in enumerate(available_materials):
                print(f"{idx + 1}. {item[1]}")
            while True:
                try:
                    choice = int(input("Enter the number corresponding to the material: ")) - 1
                    if 0 <= choice < len(available_materials):
                        selected_material = available_materials[choice]
                        break
                    else:
                        print("Invalid selection. Please choose a valid number.")
                except ValueError:
                    print("Invalid input. Please enter a number.")
```

Figure 15 Calling on Materials from SQL Database

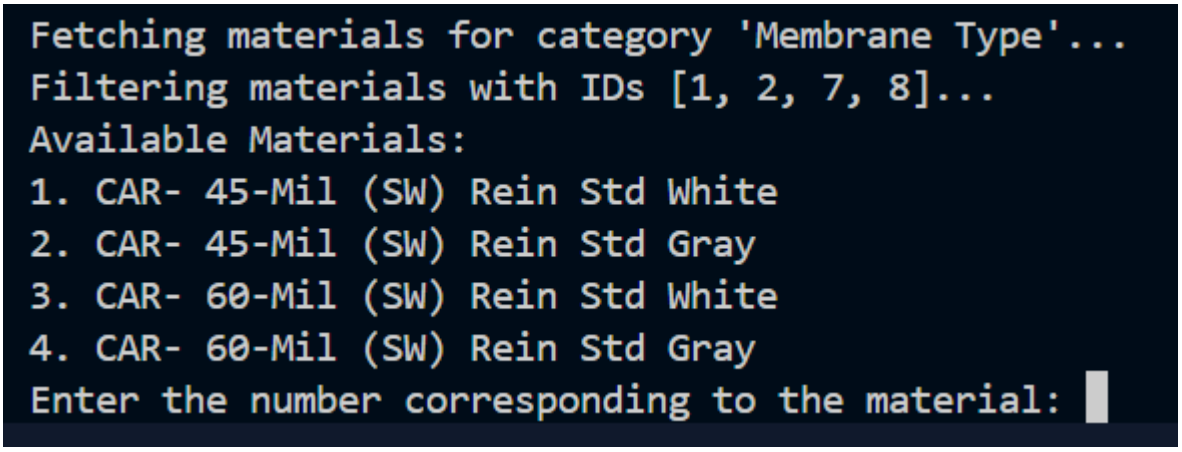


Figure 16 Terminal Output Providing Database Material Choices

Error Handling Enhancements:

- Iteratively tested error scenarios, such as invalid data, and refined terminal prompts to provide clearer instruction

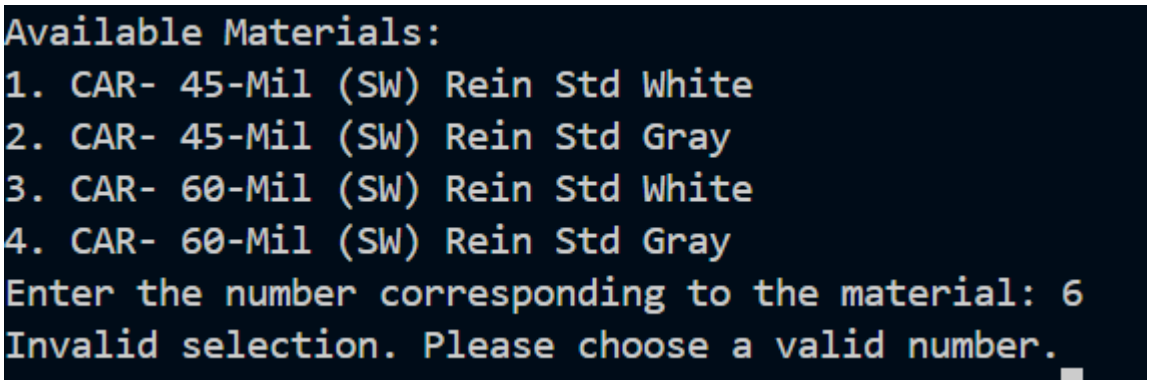


Figure 17 Error Response Upon Invalid Input

Code Process Breakdown

As a part of backend optimization the code was modularized into separate scripts with each one devoted to individual tasks in the program:

- **main.py:** The core of the program that controls user interaction and coordinates workflows.
- **material_pricing.py:** The section that takes care of the pricing calculations and validations of materials.
- **materials_query.py:** The interface to the SQL database that receives material data and options.
- **production_quantities.py:** The calculator of the production quantities that can be used manually or by formula.
- **project_io.py:** The module that is responsible for saving and loading project data which is in JSON format.

This structure was the first step in programming development. Each component was built, tested, and upgraded without being dependent on the others. The architecture also allowed scalability for the next versions, where additional workflows and user interface elements can be integrated.

See Iteration 3 to view the full code.

5.4.4 Outcome

1. **Backend Accuracy:**
    - Progressive testing confirmed the actual Python calculations replicated those of the Excel worksheet.
    - Single-input validation was the last confirmation for backend reliability before more data sets were added.
2. **User Interaction Enhancement:**
    - Dropdown menus demonstrated a functioning proof of the application that the backend workflows were integrated with user inputs.
3. **Strengthen Error Resilience:**
    - Better error handling was a preparation for the system in user testing during next iterations.
4. **GUI Development Foundation:**
    - Temporary elements confirmed that backend workflows were strong and ready for final association with a GUI.

Intermediary Test 4: Terminal-Based Option Selection Usability

- **Setup:** A placeholder dropdown system was implemented using a **text-based interface** where users would see:

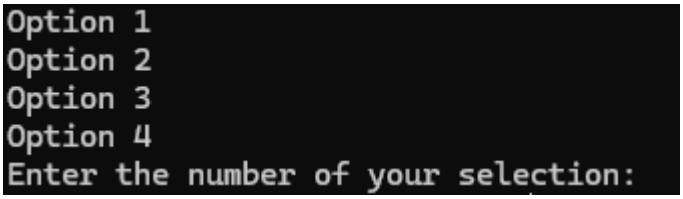


Figure 18 Terminal Placeholder Dropdown Selection

Test users were asked to select an option by entering the corresponding number.

- **Results:** Some users found the format unintuitive, hesitating before entering a number because there was no **immediate confirmation** of their selection.
- **Analysis:** The lack of visual feedback (like highlighting selected options) made it harder for users to be sure their selection was registered.
- **Steps Taken:** Added a **confirmation message** that displays the chosen option before proceeding:

Figure 19 Terminal Placeholder Dropdown Selection Response

5.5 Iteration 4: Wireframe Development

5.5.1 Focus

Iteration 4 involved designing and refining the user interface wireframes based on insights gained from earlier iterations of the backend workflow and feedback collected from users and stakeholders. These wireframes aimed to bridge the gap between functionality and user experience, focusing on usability, aesthetics, and alignment with TPC Roofing's branding.

5.5.2 Process

Building on the wireframe sketches created during the concepting phase, I incorporated the knowledge gained from developing the backend system in earlier iterations. This included understanding the Excel-based workflows, key pain points, and how the interface could complement the automated backend processes.

Design Iterations

1. **Initial Wireframe Designs**
    - **Proto 1: Dark Mode vs. Light Mode**: A light mode and dark mode version were created to test user preferences (figures 24 and 25). A short A/B test was conducted among target users, with the results indicating a preference for the dark mode. It featured a turquoise-green accent color aligned with TPC Roofing's branding.

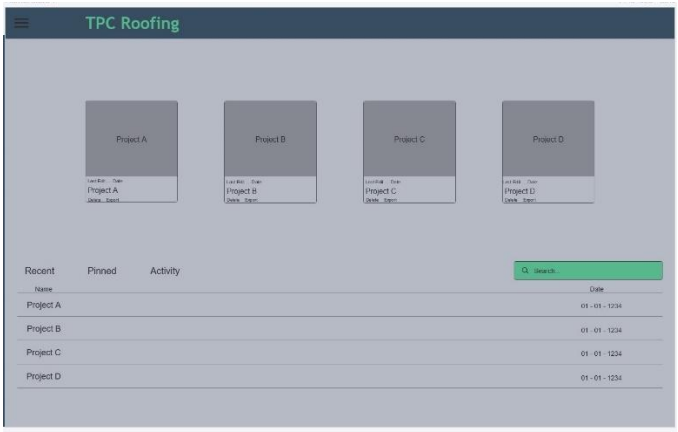


Figure 20 Light Mode Theme

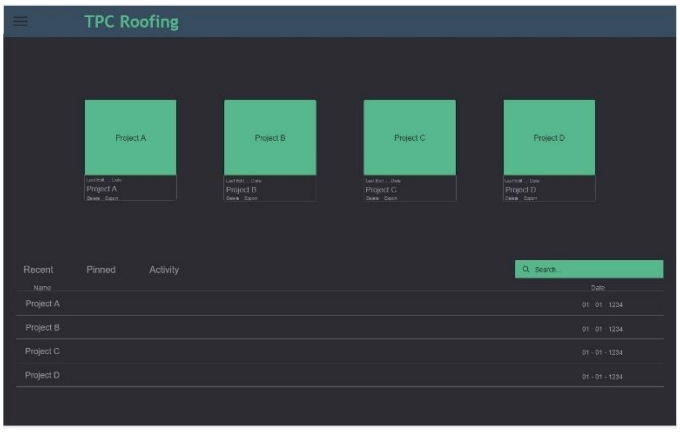


Figure 21 Dark Mode Theme

- **Proto 2: Outdated Design**: The first digital design, figure 26, was met with feedback that the color scheme appeared outdated. This version was scrapped, and subsequent iterations focused on modernizing the visual style while maintaining functionality.

Figure 22 Outdated Design Choice

2. **Proto 2: Table Refinements**

- **Alternating Rows vs. Horizontal Lines**: Multiple table styles were tested to improve readability. Users preferred alternating row colors for clarity.
- **Vertical Line Separators**: Added to separate columns clearly, improving the overall layout and usability.

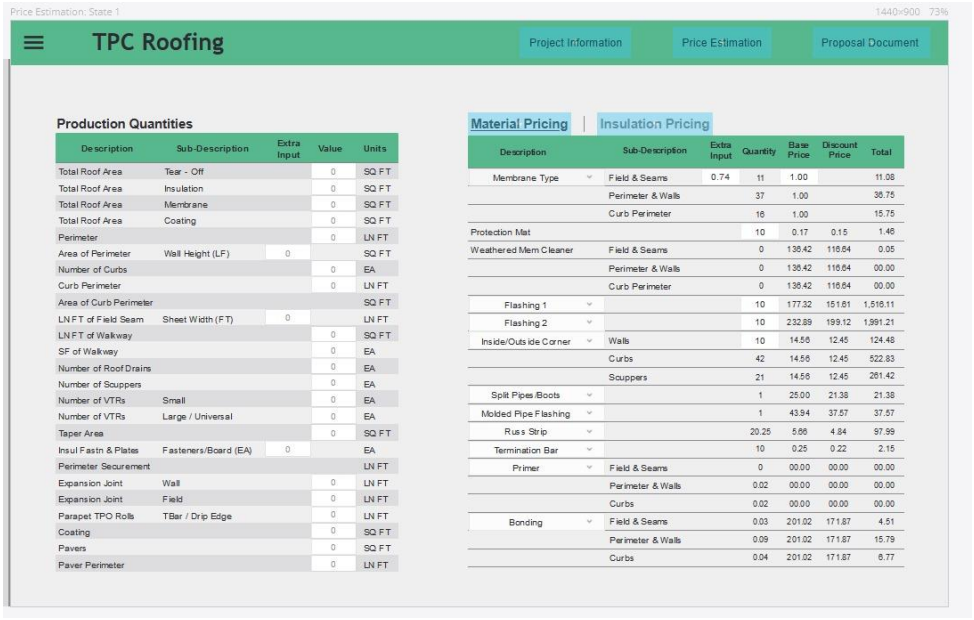


Figure 23 Alternating Rows or Horizontal Line Separation

Figure 24 No Lines or Horizontal Line Separation

- **Red Highlights for User Inputs**: Highlighting user input areas in red (figure 29) was attempted but ultimately deemed too visually jarring. This feature was replaced with more subtle cues as shown in figure 30.

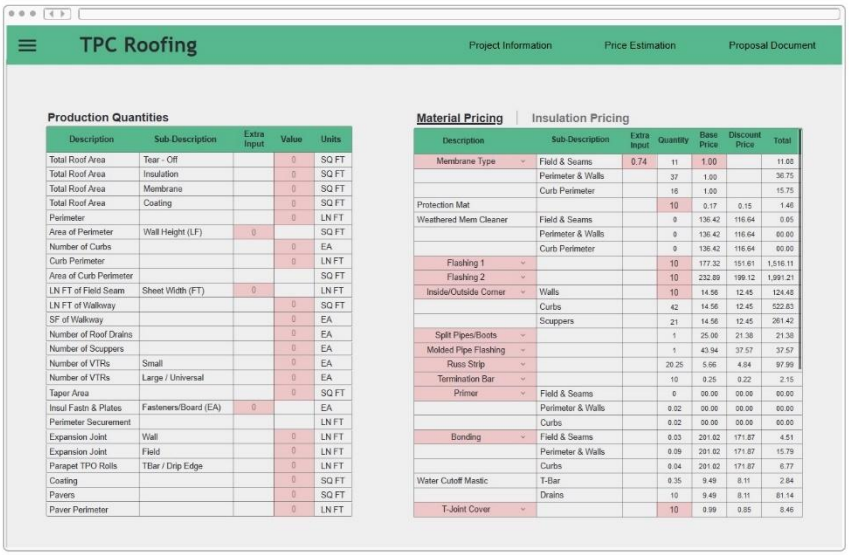


Figure 25 Red Input Highlight Indicator

Intermediary Test 5: A/B Testing on Interface Preferences

- **Setup:** Conducted an A/B test comparing light mode vs. dark mode wireframes with a sample of 23 users.
- **Results:** 16 out of 23 users preferred light mode due to better readability.
- **Analysis:** Light mode provided a more professional appearance in line with TPC branding.
- **Steps Taken:** Light mode was chosen as the default UI theme in later iterations.

For more information see A/B Testing.

5.5.3 Final Design

The final wireframe design incorporated the following features:

- A light mode interface with turquoise-green accents, reflecting TPC Roofing's branding.
- Horizontal for improved table readability with a more modern design.
- Subtle white visual cues for user input areas.
- Logical navigation elements, including sections for Production Quantities, Material Pricing, and Insulation Pricing, integrated seamlessly into the workflow.

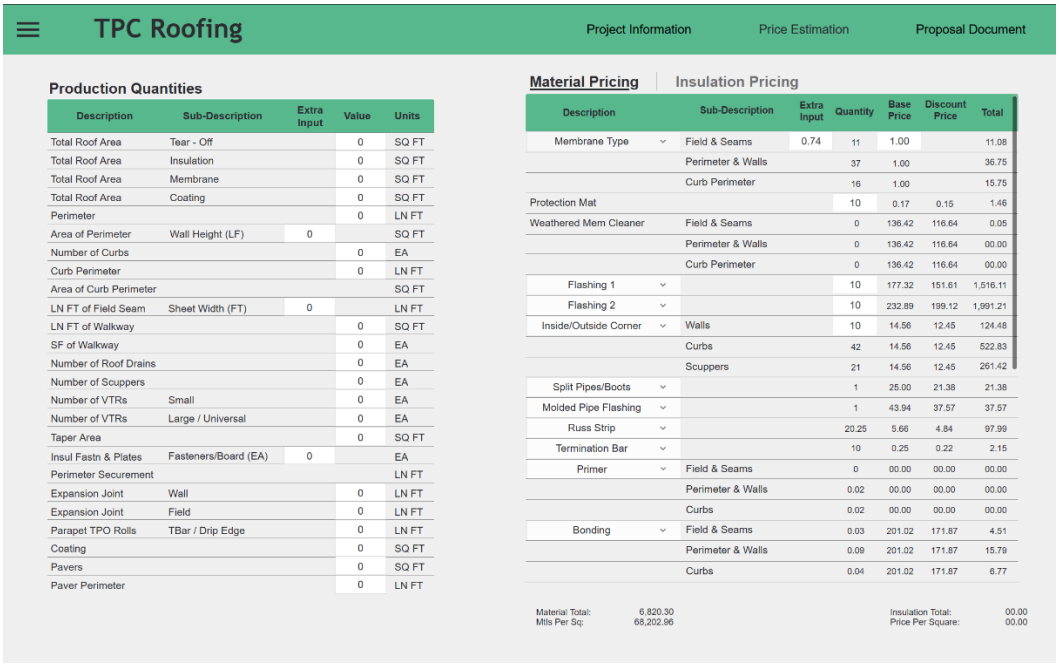


Figure 26 White Input Highlight Indicator

5.5.4 Outcome

This iteration successfully transformed backend functionality into a cohesive and visually appealing interface. The wireframe not only aligns with client expectations but also sets the stage for seamless interaction with the backend prototype.

5.6 Conclusion

The prototyping phase showcased the evolution of the project through four iterations, each building on feedback and addressing challenges. Python's flexibility and the iterative approach ensured that the prototype aligned with the client's needs while laying a strong foundation for future scalability.

6. Phase 4 – Evaluation

6.1 Set-Up

This used to determine the usability, performance, and alignment of the prototype with user expectations. My goal was to get feedback from various stakeholders and users to further improve core functionalities of the prototype and prepare for future iterations of the project, including the development of a graphical interface. The evaluation methods included are listed in the table below. Each method was chosen to address specific parts of this prototype, which would ensure a comprehensive understanding of its strengths and areas for improvement.

Method	Results
Peer Review (Turnhout et al., 2015)	Method 18
Usability Testing	Method 19
Think Aloud Method (Hanington et al., 2012)	Method 20
System Performance Validation	Method 21

Table 5 Evaluation Methods

6.2 Methods and Results

6.2.1 Peer Review

Description:

Peer reviews gather feedback from professionals and users familiar with the workflows and roofing estimation process. Participants tested the terminal-based prototype and the wireframe GUI design, testing workflow alignment, accuracy, and usability. The goal was to spot blind spots in the design and make sure that the prototype met real-world user needs. This would ensure that this project could meet the usability and scalability needs of existing and new hires at TPC, and other potential companies using this design.

Execution:

- **Participants:** Roofing estimators and professionals with data-handling experience.
- **Tasks:**
    1. **Excel Comparison** – Participants first filled out an estimate in Excel for reference.
    2. **Terminal Prototype** – They repeated the process using the coded prototype, verifying accuracy and spotting calculation discrepancies.
    3. **Wireframe Testing** – Participants reviewed the GUI wireframe for usability and navigation feedback.

Results:

- **Positive Feedback:** Improved accuracy, reduced manual effort, and better workflow efficiency, especially for large projects.
- **Challenges:** The terminal interface was unintuitive for quick edits or navigating back.
- **Suggestions:** Clearer error messages, onboarding documentation, and a streamlined process for handling unconventional materials.

Conclusion:
Peer reviews validated the prototype's accuracy and efficiency while emphasizing the need for better user navigation and onboarding support. These insights shaped the wireframe design and future development efforts.

For more test information and steps, see Method 18.

6.2.2 Usability Testing

Description:

The usability testing was conducted to test how well the prototype aligned with user expectations and supported real-world workflows. The goal was to uncover usability issues to inexperienced users, identify areas for improvement, and gauge user satisfaction.

Execution:

- **Participants:** Users with varying levels of experience in Excel and similar tools.
- **Tasks:** The test, like before, was split into three sections:

1. **Excel Comparison:** Users completed an estimate manually in Excel as a baseline.
2. **Terminal Prototype:** Participants repeated the process using the coded prototype, assessing input accuracy and error handling.
3. **GUI Wireframe Testing:** Users navigated the interface wireframe, providing feedback on design clarity and workflow intuitiveness.

- **Feedback:** After testing, participants answered questions on workflow intuitiveness, error handling, and design clarity.

Results:

- **Positive Outcomes:**
    - Automated input and material pricing calculations were significant improvements over manual Excel work.
    - Error messages were useful but needed clearer guidance for resolving mistakes.
- **Challenges:**
    - New users found the interface intimidating, especially with large data outputs.
    - Some terminology required roofing knowledge, making it difficult for non-industry users.
- **Suggestions:**
    - Simplify new material creation workflows.

- Provide detailed onboarding instructions for first-time users.

Conclusion:

Usability testing validated the prototype's efficiency and automation improvements while highlighting the need for a simplified, user-friendly interface. These insights informed plans to merge the wireframe with the terminal program into a singular, intuitive system.

For more information see Method 19.

6.2.3 Think Aloud Method

Description:

The think aloud method was conducted with a participant who regularly uses Excel but is unfamiliar with roofing estimation. This participant verbalized their thoughts while using the terminal prototype, offer insights into for usability, functionality, and clarity.

Execution:

- **Participant:** A plant biologist with extensive experience in Excel for data and analysis tasks.
- **Tasks:** The participant completed the same three-step testing process, providing real-time feedback.

Results:

- **Positive Feedback:**
  - Clear terminal prompts made data entry intuitive.
  - The participant found the automation more efficient than manual Excel input.
- **Challenges:**
  - Preferred Excel's flexibility, especially for quick edits.
  - Navigation was restrictive, lacking an easy way to go back and adjust inputs.

Conclusion:

This test provided **valuable insight into how non-roofing professionals** interact with the tool, reinforcing the need for **better navigation and flexibility** in future **GUI iterations**.

For more details see Method 20.

6.2.4 System Performance Validation

Description:

The system performance validation was a thorough review of the codebase to assess its efficiencies, error handling, scalability, and alignment with user workflows. It focused on database interactions, calculations, proper name displays, error feedback, and overall system design.

Execution:

- **Code Review**: The code was reviewed section by section, assessing database queries, calculations, formatting, and scalability for future expansions.

- **Testing Scenarios**: Validation was conducted during and after coding phases to ensure updates didn't introduce errors or disrupt previous functionality.

Results:

Strengths:

- Efficient database queries with modular functions for future adaptability.
- Dynamic production quantity structure supports different workflows.
- Modular pricing calculations allow flexibility for future enhancements.

Challenges Identified:

- Error handling needed improvement for empty/invalid database queries to prevent crashes.
- User feedback messages were unclear, leading to confusion during testing.

Key Improvements for Future Iterations:

- Strengthen error handling and validation for better stability and debugging.
- Enhance user-friendly error messages to improve clarity for both users and developers.

Conclusion:

The system's modular design supports scalability for company expansion, but improvements in error handling and input validation are needed for a better user experience. These refinements will be incorporated into future iterations.

For more details, see Method 21.

6.3 Insights and Adjustments Based on Evaluation

Testing validated that the prototype accurately replicated TPC Roofing's workflow and displayed correct data. However, key insights led to the following refinements:

1. **Improved Error Handling:** Enhanced validation checks and error messages based on usability testing feedback.
2. **Streamlined Workflow:** Simplified **frontend-backend integration**, reducing input complexity and clarifying requested inputs.
3. **Better Navigation:** Addressed usability concerns by refining navigational flow, with future improvements planned for clearer button.
4. **User-Friendly Interface:** A/B testing informed a **light mode UI** with subtle visual enhancements for clarity.

For more information see here and Method 17.

6.4 Unresolved Issues and Limitations

Though I had significant insights and improvements, some issues still remained due to the project scope and restraints:

1. **Input Validation:** Input validation was improved, but some cases (invalid data types) were not fully addressed. These will be fixed in future development.

2. **User Onboarding:** Onboarding documentation was improved with clearer design and labeling. However, some users still found the initial setup process challenging. Future iterations can include more comprehensive onboarding materials, such as tooltip pop-ups.

6.5 Future Iterations

This phase not only led to immediate refinements but also set the stage for future iterations:

1. **User Interface:** Integrating the backend with a **GUI-based frontend** will replace the terminal interface, improving usability.
2. **Enhanced Automation:** Adding **Excel sheet automation** and **proposal generation** will further reduce manual work.
3. **Scalability & Multi-User Support:** The modular design allows for **future scalability**, including **multi-user access**.
4. **Improved Error Handling & Validation:** Expanding **error handling and input validation** will ensure **greater system reliability**.

For more information see here.

6.6 Conclusion

This evaluation phase played a critical role in improving the prototype and ensuring that it met the needs of TPC's estimators. By getting feedback from various stakeholders and users, the prototype was improved over time, which addressed key pain points and laid the foundation for future development. The information I gathered in this phase will continue to guide the project as I continue to work on it so I can be sure the final product is both functional and user-friendly.

7. Conclusion

This project's design methodology clearly demonstrates the value of stakeholder input and iterative prototyping in creating a workable roofing estimation prototype. In order to solve some of the most significant issues and lay the groundwork for future advancements, the project purposefully concentrates on automating production volumes and material cost. The prototype's ability to increase workflow accuracy and efficiency was shown by the tests, which helped TPC Roofing meet their urgent requirement. This foundation is anticipated to serve as the basis for future iterations of the project, which should include features like data visualization tools, a user-friendly interface, and supplier price updates to transform decision-making and project management.

Bibliography

About SQLite. (n.d.). https://sqlite.org/about.html

AccuLynx. (2024, December 19). Best Roofing CRM Software | AccuLynx. https://acculynx.com/

Banach, Z. (2022, March 21). *Input validation errors: The root of all evil in web application security.* www.invicti.com https://www.invicti.com/blog/web-security/input-validation-errors-root-of-all-evil/

BEAM AI. (n.d.). Roofing estimating software powered by AI Takeoffs - BEAM AI. https://www.ibeam.ai/estimate/roofing-estimating-software

Best roofing software | Roofr. (n.d.-b). https://roofr.com/

Ciampaglia, S. (n.d.). Revolutionizing Construction Bids: How Automated Bidding Saves Time and Maximizes Profits. ContractComplete. https://www.contractcomplete.com/revolutionizing-construction-bids-how-automated-bidding-saves-time-and-maximizes-profits/

Codecademy. (n.d.). Learn to code - for free | Codecademy. https://www.codecademy.com/

Construction Bid Management Software: A complete guide for contractors (2025). (n.d.). https://www.projectmark.com/blog/construction-bid-management-software

CS50X 2025. (n.d.). https://cs50.harvard.edu/x

Dam, R. F., & Teo Yu Siang. (2016, November 26). *Scamper: How to Use the Best Ideation Methods. The Interaction Design Foundation; Interaction Design Foundation. https://www.interaction-design.org/literature/article/learn-how-to-use-the-best-ideation-methods-scamper?srsltid=AfmBOopMoSh9jDHHYbhcV2W2FLmPBLuXu8yNX9Z7RdBvnTWJLWlcblpF*

DB Browser for SQLite. (n.d.). DB Browser for SQLite (Version 3.12.2). https://sqlitebrowser.org

Drew, D. S., & Skitmore, R. M. (1993). "Competitive Tendering: Principles and Practices." Construction Management and Economics, 11(1), 105-112.

edX. (n.d.). CS50: Computer Science Courses and Programs from Harvard. https://www.edx.org/cs50

Fu, H., Drew, D. S., Lo, H. P., & Li, H. (2002). "The Application of a Competitive Bidding Model in Construction." Automation in Construction, 11(5), 547-559.

Gates, M. (1967). "Bidding Strategies and Probabilities." Journal of the Construction Division, ASCE, 93(1), 75-107.

iRoofing. (2024, December 4). Best Roofing software | #1 roofing app for contractors | iRoofing. https://iroofing.org/

Jobber Roofing business software. (n.d.). Jobber. https://www.getjobber.com/industries/roofing-software/

JobNimbus. (2024, October 15). JobNimbus - top roofing software for contractors. https://www.jobnimbus.com/

LEAP - #1 Sales Platform for Contractors. (2024, May 23). Leap Team - LeAP - Roofing and Remodeling software. Leap - the Complete Platform for Professional Contractors. https://leaptodigital.com/leap-team/

Learn Python - free interactive Python tutorial. (n.d.). https://www.learnpython.org/

Learn SQL | CodeCademy. (n.d.). Codecademy. https://www.codecademy.com/learn/learn-sql

Liu, H. & Kuan Tan, H. B. (2008). Testing input validation in Web applications through automated model recovery. *Journal of Systems and Software,* 81(2), 222-233. https://doi.org/10.1016/j.jss.2007.05.007

Maze. (2023, March 13). What is Rapid Prototyping? Benefits, Processes & Best Practices. Maze. https://maze.co/blog/rapid-prototyping/

McKinney, W. (2012). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. http://cds.cern.ch/record/2288466

Microsoft Corporation. (n.d.). Microsoft Excel. https://www.microsoft.com

Microsoft Corporation. (n.d.). Visual Studio Code). https://code.visualstudio.com/

Proto.io. (n.d.). Proto.io - Prototyping for all. https://proto.io/

RoofSnap. (n.d.). RoofSnap. https://roofsnap.com/

Serrat, O. (2017, May). (PDF) *The SCAMPER Technique. ResearchGate. https://www.researchgate.net/publication/318018918_The_SCAMPER_Technique*

SmartBuild Systems, 2022. "Roofing Software for Automated Bidding."

Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack Overflow. https://stackoverflow.com/

System Flowchart: Definition, application, benefits, symbols and e. (n.d.). Boardmix. https://boardmix.com/knowledge/system-flowchart/

umnov.denis. (2023, July 26). Bidding Automation: Opening Constructions Hidden Potentials. ConWize. https://conwize.io/articles/the-untold-secrets-of-bidding-automation/

Welcome to Python.org. (2024, December 19). Python.org. https://www.python.org/

Zhou, R. (2021a, December 30). Experience Prototyping - Research Methods [Group 4] - Medium. Medium. https://medium.com/research-methods-group-4/experience-prototyping-d59d69ec943d

Zhou, R. (2021b, December 30). Experience Prototyping - Research Methods [Group 4] - Medium. Medium. https://medium.com/research-methods-group-4/experience-prototyping-d59d69ec943d

Evidentiary Documents

Phase 1 - Research

The Orientation and Understanding Phase, or Research Phase, started with a series of calls with the client to understand his company and workflow, the exact notes of which can be found in the appendix, under the client interviews.

Expanded Context and Problem Statement

Contracting companies worldwide are dependent upon employees making accurate and competitive bids. In the building industry, a bid from multiple companies will be based on architectural drawings, specifications, materials suppliers and any additional instructions from the potential client. A particular business is reliant on the accuracy of the estimator to interpret this information and consistently use updated supplier information (product specificity, product availability, as well as pricing). Of course, the more experienced contractor would be better at compiling this information to generate a bid quickly and more accurately.

In fact, research shows that competitive bids are positively correlated with the experience of the contractor (Fu and Drew, 2003). Thus, the success of a company will rely on its most experienced contractors and the addition of a new employee is added or loss of an experienced contractor puts a company in jeopardy. The bidding process involves more than the contractor. Contractors must rely on the upkeep of supplier product data, changes in pricing, product availability, and product design must be regularly updated. Interestingly, I found during this study that competitive bidding is an international topic of research.

Government agencies, for instance, are reliant on the receiving accurate and fair bids to best spend limited funds on everything from new buildings, building renovations or even transportation (Liu et al., 2022). In these cases, there are teams of people on the customer side involved in analyzing bids that are received.

For all companies, the ability to quickly and accurately generate bids is important to their competitiveness. This is especially true for smaller companies like TPC Roofing. This determines their ability to stay in business as well as the ability to expand the business. Companies, of course, have contingency pricing included in a bid, for example if their potential problems with pricing fluctuations between the bid and the actual project or if the initial estimate was less accurate than expected.

TPC Roofing, like many other contracting companies worldwide, is dependent on information assembled by workers themselves. It is important to reiterate that this problem of inefficiency spans many areas where contractors must generate bids to meet customer needs. Personal observation of a home renovation business found that often several contractors report to a customer to make a bid. The contractors each spend 45 minutes to an hour on site with a potential customer and then do not provide a bid, provide a wildly over-priced bid, or return after several weeks to months and ask if a bid is still wanted.

These companies must rely on limited tools: tools that are easy enough for any employee to use and ones that are affordable. Only a few such tools are available. Commonly, these tools consist of spreadsheets and access to simple databases from suppliers to access product information. These

tools are tedious to use, making time consuming to do the manual data aggregation which delays results. These issues can get worse with growth: the addition of new employees, addition of new suppliers, addition of new products from each supplier, updating pricing, substituting similar products within a project, etc. TPC roofing employees shared these basic and important concerns about the limitations of relying only on limited access to spreadsheets or on one or two very experienced contractors. These circumstances lead to the need for an integrated system to diminish the errors in data entry and enhance productivity to allow the company to grow. The customer for this project, TPC Roofing, is representative of thousands of smaller companies that want to streamline their operation, minimize the manual burden, and create a system that is flexible, fast and accurate.

Expanded Opportunity and Goal

TPC Roofing is a company that is active in a purely competitive industry sector where the main determinant of success is speed and accuracy of estimates (Evidentiary Document, Methods 1 and 2). Their continuous use of an Excel-based manual process is a great barrier to their productivity and exposes them to the risk of mistakes, lower output, inability to quickly train additional employees, and inaccuracies. This project is intended to enhance estimation efficiency through automation, data centralization, and increase the scalability of a digital workflow.

Steps to Achieve These Goals

To address these challenges, this project focuses on **three key improvements** to TPC Roofing's estimation workflow:

1. **Automating Calculation Processes:**
    - Replace manual Excel-based material pricing calculations with automated Python functions that eliminate human error.
    - Implement real-time input validation to prevent incorrect entries before they impact final estimates.
2. **Centralizing Data for Efficiency & Accuracy:**
    - Store material pricing and production quantity data in a structured SQLite database instead of separate Excel files.
    - Develop an interface that automates supplier price updates, ensuring estimators always work with current pricing.
    - Implement a single-entry system where data is updated in one place instead of being manually transferred across multiple sheets.
3. **Improving Scalability & Training Speed:**
    - Design a modular system that can expand to support future functionalities like proposal generation and multi-user access.
    - Structure the workflow to reduce the onboarding time for new estimators by automating repetitive tasks.

- Ensure the system aligns with industry best practices for usability, making it easy to adopt and integrate with existing work processes.

The opportunities are two-fold: directly meeting the needs of TPC Roofing while simultaneously creating a model of accuracy and efficiency that will lead to increased scalability, reflecting the needs of companies across multiple industries. The smallness of TPC Roofing allows direct access to the main users, intimate knowledge of company objectives without multiple layers of management, and direct ability to test usability by those that have direct knowledge of the inefficiencies and inadequacies of the current manual spreadsheet system.

Method 1 – Client Interviews

Purpose of Meetings

1. **Meeting 1 (August):** Initial discussion and exploring project direction and understanding The client's workflow and skillsets.
2. **Meeting 2 (August):** Getting a detailed understanding of the client's workflow, supported by examples of PDFs and spreadsheets provided by the client, and identifying potential areas for automation.

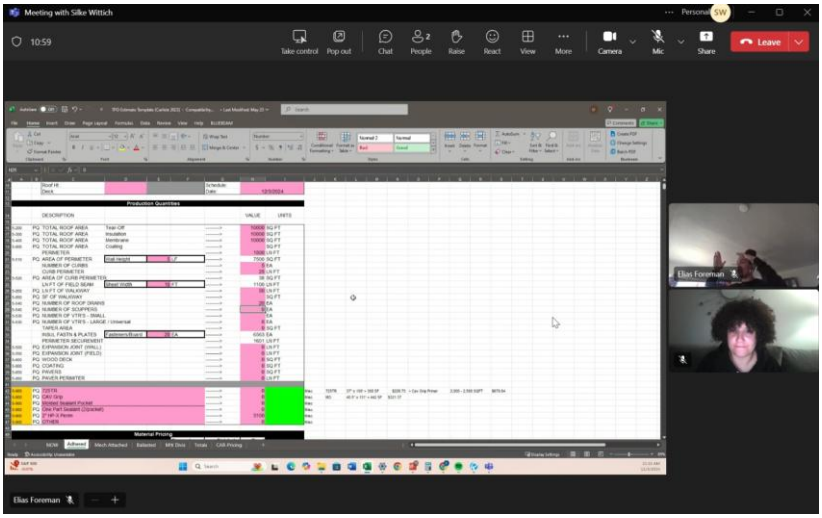


Figure 27 Client Meeting Evidence

Justification

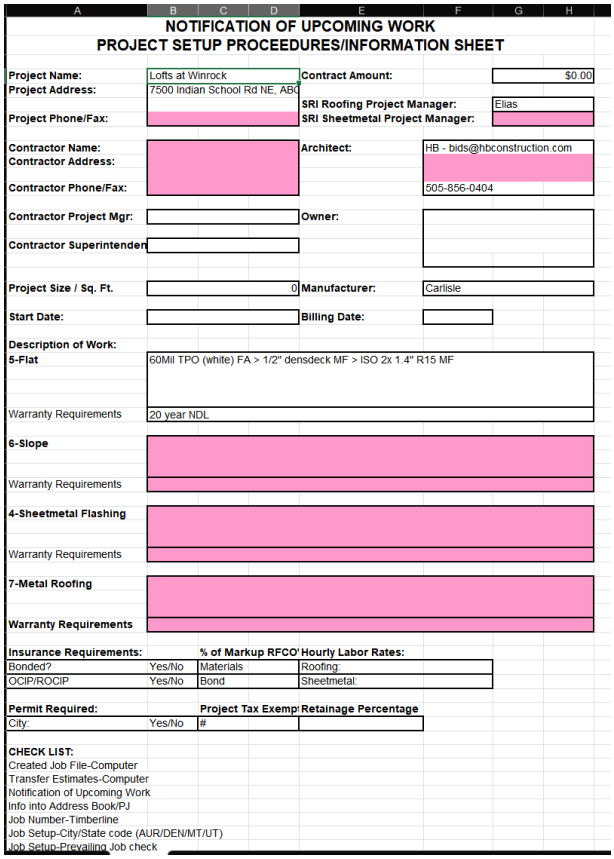
This method was chosen as a key research approach for this project because it provides direct user-centered insights into the workflow, challenges, and expectations of TPC Roofing's estimators. Since the primary goal of the project is to streamline and partially automate TPC Roofing's estimation process; understanding the current system from the end-users' perspective was important to make sure the solution aligns with their needs.

Interview 1. Understanding Workflow

Reference Appendix 1 for notes.

Interviewee: Client, Lead Estimator at TPC Roofing

1. **Blueprints:**
    - Uses a PDF reader to trace and measure relevant lines from roof plans.
    - Measurements are manually recorded since the new company lacks the PDF reader.



NOTIFICATION OF UPCOMING WORK
PROJECT SETUP PROCEEDURES/INFORMATION SHEET

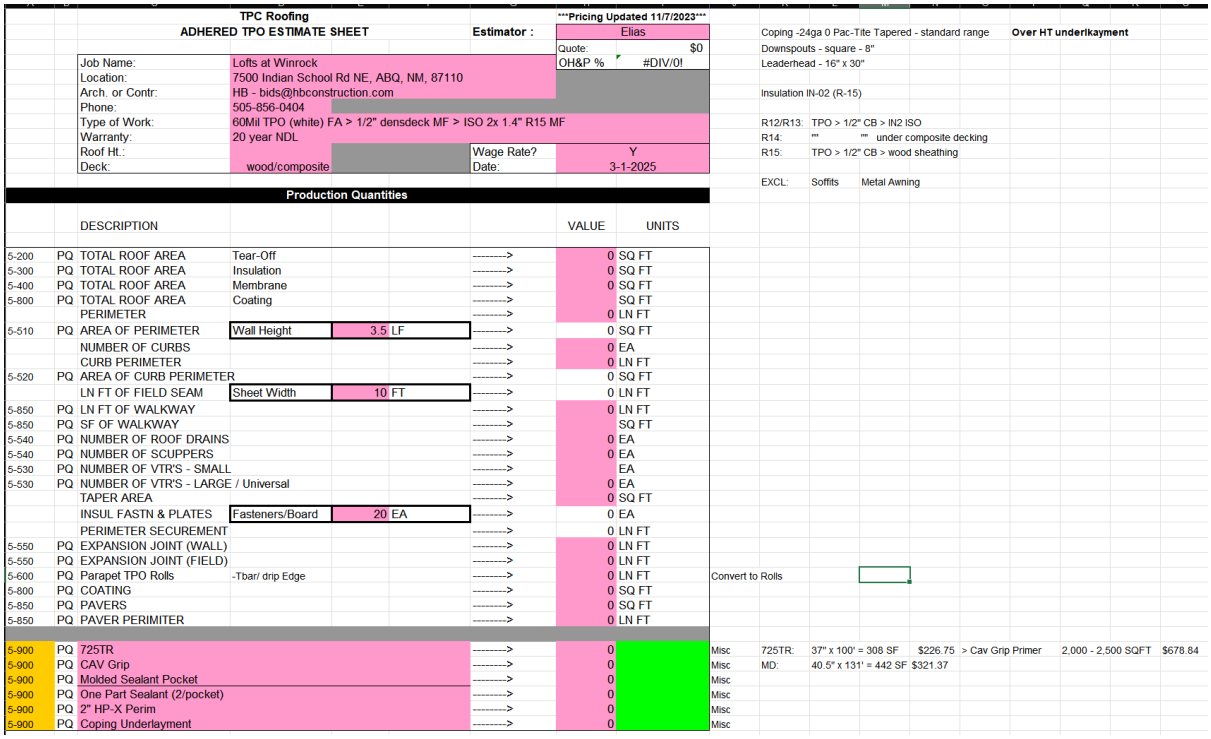
Project Name:	Lofts at Winrock	**Contract Amount:**	$0.00
Project Address:	7500 Indian School Rd NE, ABC		
		SRI Roofing Project Manager:	Elias
Project Phone/Fax:		**SRI Sheetmetal Project Manager:**	
Contractor Name:		**Architect:**	HB - bids@hbconstruction.com
Contractor Address:			
Contractor Phone/Fax:			505-856-0404
Contractor Project Mgr:		**Owner:**	
Contractor Superintenden			
Project Size / Sq. Ft.	0	**Manufacturer:**	Carlisle
Start Date:		**Billing Date:**	
Description of Work:			
5-Flat	60Mil TPO (white) FA > 1/2" densdeck MF > ISO 2x 1.4" R15 MF		
Warranty Requirements	20 year NDL		
6-Slope			
Warranty Requirements			
4-Sheetmetal Flashing			
Warranty Requirements			
7-Metal Roofing			
Warranty Requirements			

Insurance Requirements:		**% of Markup RFCO**	**Hourly Labor Rates:**
Bonded?	Yes/No	Materials	Roofing:
OCIP/ROCIP	Yes/No	Bond	Sheetmetal:

Permit Required:		**Project Tax Exemp**	**Retainage Percentage**
City:	Yes/No	#	

CHECK LIST:
Created Job File-Computer
Transfer Estimates-Computer
Notification of Upcoming Work
Info into Address Book/PJ
Job Number-Timberline
Job Setup-City/State code (AUR/DEN/MT/UT)
Job Setup-Prevailing Job check

Figure 28 Client Excel Sheet Page 1



TPC Roofing
ADHERED TPO ESTIMATE SHEET

Estimator : Elias

Pricing Updated 11/7/2023

Quote: $0
OH&P % #DIV/0!

Job Name:	Lofts at Winrock
Location:	7500 Indian School Rd NE, ABQ, NM, 87110
Arch. or Contr:	HB - bids@hbconstruction.com
Phone:	505-856-0404
Type of Work:	60Mil TPO (white) FA > 1/2" densdeck MF > ISO 2x 1.4" R15 MF
Warranty:	20 year NDL
Roof Ht.:	
Deck:	wood/composite

Wage Rate? Y
Date: 3-1-2025

Coping -24ga 0 Pac-Tite Tapered - standard range — Over HT underlkayment
Downspouts - square - 8"
Leaderhead - 16" x 30"

Insulation IN-02 (R-15)

R12/R13: TPO > 1/2" CB > IN2 ISO
R14: "" "" under composite decking
R15: TPO > 1/2" CB > wood sheathing

EXCL: Soffits Metal Awning

Production Quantities

		DESCRIPTION				VALUE	UNITS
5-200	PQ	TOTAL ROOF AREA	Tear-Off		-------->	0	SQ FT
5-300	PQ	TOTAL ROOF AREA	Insulation		-------->	0	SQ FT
5-400	PQ	TOTAL ROOF AREA	Membrane		-------->	0	SQ FT
5-800	PQ	TOTAL ROOF AREA	Coating		-------->		SQ FT
		PERIMETER			-------->	0	LN FT
5-510	PQ	AREA OF PERIMETER	Wall Height	3.5 LF	-------->	0	SQ FT
		NUMBER OF CURBS			-------->	0	EA
		CURB PERIMETER			-------->	0	LN FT
5-520	PQ	AREA OF CURB PERIMETER			-------->	0	SQ FT
		LN FT OF FIELD SEAM	Sheet Width	10 FT	-------->	0	LN FT
5-850	PQ	LN FT OF WALKWAY			-------->	0	LN FT
5-850	PQ	SF OF WALKWAY			-------->		SQ FT
5-540	PQ	NUMBER OF ROOF DRAINS			-------->	0	EA
5-540	PQ	NUMBER OF SCUPPERS			-------->	0	EA
5-530	PQ	NUMBER OF VTR'S - SMALL			-------->		EA
5-530	PQ	NUMBER OF VTR'S - LARGE / Universal			-------->	0	EA
		TAPER AREA			-------->	0	SQ FT
		INSUL FASTN & PLATES	Fasteners/Board	20 EA	-------->	0	EA
		PERIMETER SECUREMENT			-------->	0	LN FT
5-550	PQ	EXPANSION JOINT (WALL)			-------->	0	LN FT
5-550	PQ	EXPANSION JOINT (FIELD)			-------->	0	LN FT
5-600	PQ	Parapet TPO Rolls	-Tbar/ drip Edge		-------->	0	LN FT
5-800	PQ	COATING			-------->	0	SQ FT
5-850	PQ	PAVERS			-------->	0	SQ FT
5-850	PQ	PAVER PERIMITER			-------->	0	LN FT
5-900	PQ	725TR			-------->	0	
5-900	PQ	CAV Grip			-------->	0	
5-900	PQ	Molded Sealant Pocket			-------->	0	
5-900	PQ	One Part Sealant (2/pocket)			-------->	0	
5-900	PQ	2" HP-X Perim			-------->	0	
5-900	PQ	Coping Underlayment			-------->	0	

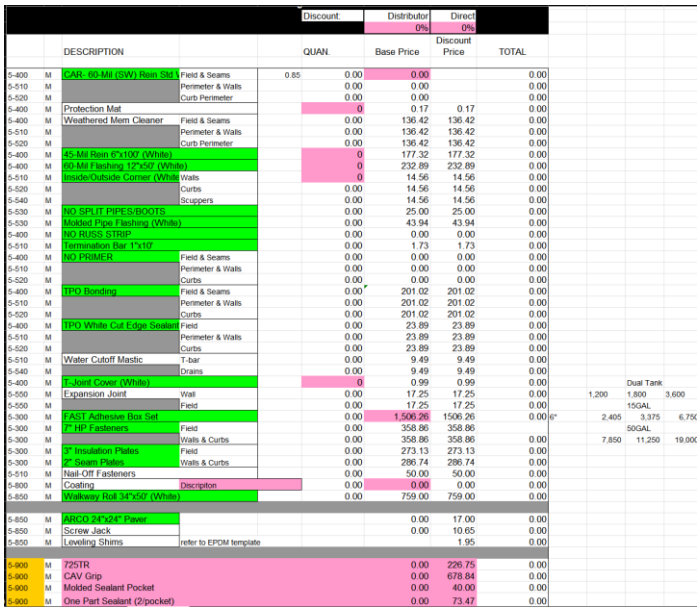
Convert to Rolls

Misc 725TR: 37" x 100' = 308 SF $226.75 > Cav Grip Primer 2,000 - 2,500 SQFT $678.84
Misc MD: 40.5" x 131' = 442 SF $321.37
Misc
Misc
Misc
Misc

Figure 29. Production Quantities Section of Original Excel Sheet

2. **Measurements to Materials:**

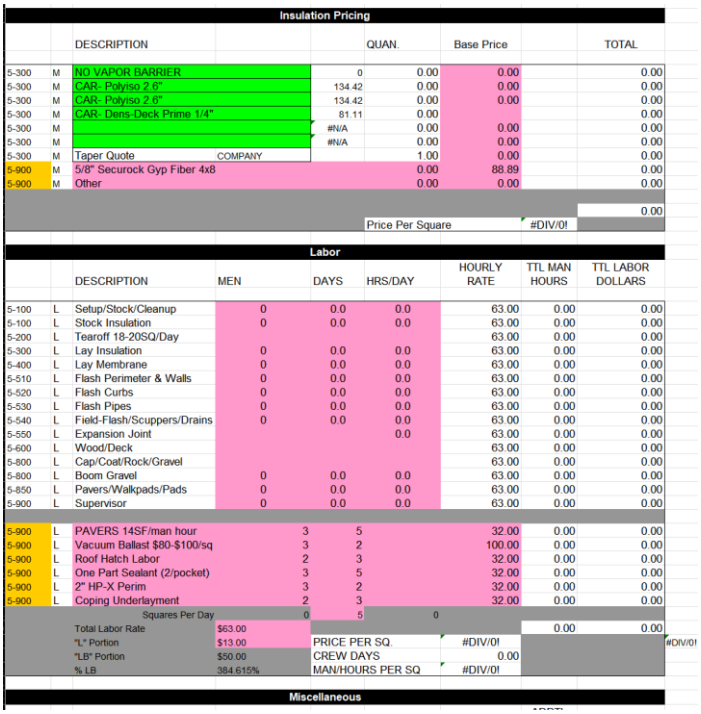
   - Measurements are inputted into Excel spreadsheets to calculate required materials (e.g., membrane, insulation, glue, screws).
   - Different tabs organize materials by roof type (TPO, wall panels, metal roof, sloped).



		DESCRIPTION		QUAN.	Discount:	Base Price	Discount Price	TOTAL
						Distributor 0%	Direct 0%	
5-400	M	CAR- 60-Mil (SW) Rein Std	Field & Seams	0.85	0.00	0.00		0.00
5-510	M		Perimeter & Walls		0.00	0.00		0.00
5-520	M		Curb Perimeter		0.00	0.00		0.00
5-400	M	Protection Mat			0	0.17	0.17	0.00
5-400	M	Weathered Mem Cleaner	Field & Seams		0.00	136.42	136.42	0.00
5-510	M		Perimeter & Walls		0.00	136.42	136.42	0.00
5-520	M		Curb Perimeter		0.00	136.42	136.42	0.00
5-400	M	45-Mil Rein 6"x100' (White)			0	177.32	177.32	0.00
5-400	M	60-Mil Flashing 12"x50' (White)			0	232.89	232.89	0.00
5-510	M	Inside/Outside Corner (White	Walls		0	14.56	14.56	0.00
5-520	M		Curbs		0.00	14.56	14.56	0.00
5-540	M		Scuppers		0.00	14.56	14.56	0.00
5-530	M	NO SPLIT PIPES/BOOTS			0.00	25.00	25.00	0.00
5-530	M	Molded Pipe Flashing (White)			0.00	43.94	43.94	0.00
5-400	M	NO RUSS STRIP			0.00	0.00	0.00	0.00
5-510	M	Termination Bar 1"x10'			0.00	1.73	1.73	0.00
5-400	M	NO PRIMER	Field & Seams		0.00	0.00	0.00	0.00
5-510	M		Perimeter & Walls		0.00	0.00	0.00	0.00
5-520	M		Curbs		0.00	0.00	0.00	0.00
5-400	M	TPO Bonding	Field & Seams		0.00	201.02	201.02	0.00
5-510	M		Perimeter & Walls		0.00	201.02	201.02	0.00
5-520	M		Curbs		0.00	201.02	201.02	0.00
5-400	M	TPO White Cut Edge Sealant	Field		0.00	23.89	23.89	0.00
5-510	M		Perimeter & Walls		0.00	23.89	23.89	0.00
5-520	M		Curbs		0.00	23.89	23.89	0.00
5-510	M	Water Cutoff Mastic	T-bar		0.00	9.49	9.49	0.00
5-540	M		Drains		0.00	9.49	9.49	0.00
5-400	M	T-Joint Cover (White)			0	0.99	0.99	0.00
5-550	M	Expansion Joint	Wall		0.00	17.25	17.25	0.00
5-550	M		Field		0.00	17.25	17.25	0.00
5-300	M	FAST Adhesive Box Set			0.00	1,506.26	1506.26	0.00
5-300	M	7" HP Fasteners	Field		0.00	358.86	358.86	
5-300	M		Walls & Curbs		0.00	358.86	358.86	0.00
5-300	M	3" Insulation Plates	Field		0.00	273.13	273.13	0.00
5-300	M	2" Seam Plates	Walls & Curbs		0.00	286.74	286.74	0.00
5-510	M	Nail-Off Fasteners			0.00	50.00	50.00	0.00
5-800	M	Coating	Discripton		0.00	0.00	0.00	0.00
5-850	M	Walkway Roll 34"x50' (White)			0.00	759.00	759.00	0.00
5-850	M	ARCO 24"x24" Paver				0.00	17.00	0.00
5-850	M	Screw Jack				0.00	10.65	0.00
5-850	M	Leveling Shims	refer to EPDM template				1.95	0.00
5-900	M	725TR				0.00	226.75	0.00
5-900	M	CAV Grip				0.00	678.84	0.00
5-900	M	Molded Sealant Pocket				0.00	40.00	0.00
5-900	M	One Part Sealant (2/pocket)				0.00	73.47	0.00

			Dual Tank	
1,200	1,800	3,600	15GAL	
6"	2,405	3,375	6,75(	
			50GAL	
	7,850	11,250	19,00(	

Figure 30. Material Pricing Section of Original Excel Sheet



Insulation Pricing

		DESCRIPTION			QUAN.	Base Price		TOTAL
5-300	M	NO VAPOR BARRIER		0	0.00	0.00		0.00
5-300	M	CAR- Polyiso 2.6"		134.42	0.00	0.00		0.00
5-300	M	CAR- Polyiso 2.6"		134.42	0.00	0.00		0.00
5-300	M	CAR- Dens-Deck Prime 1/4"		81.11	0.00			0.00
5-300	M			#N/A	0.00	0.00		0.00
5-300	M			#N/A	0.00	0.00		0.00
5-300	M	Taper Quote	COMPANY		1.00	0.00		0.00
5-900	M	5/8" Securock Gyp Fiber 4x8			0.00	88.89		0.00
5-900	M	Other			0.00	0.00		0.00
								0.00
					Price Per Square		#DIV/0!	

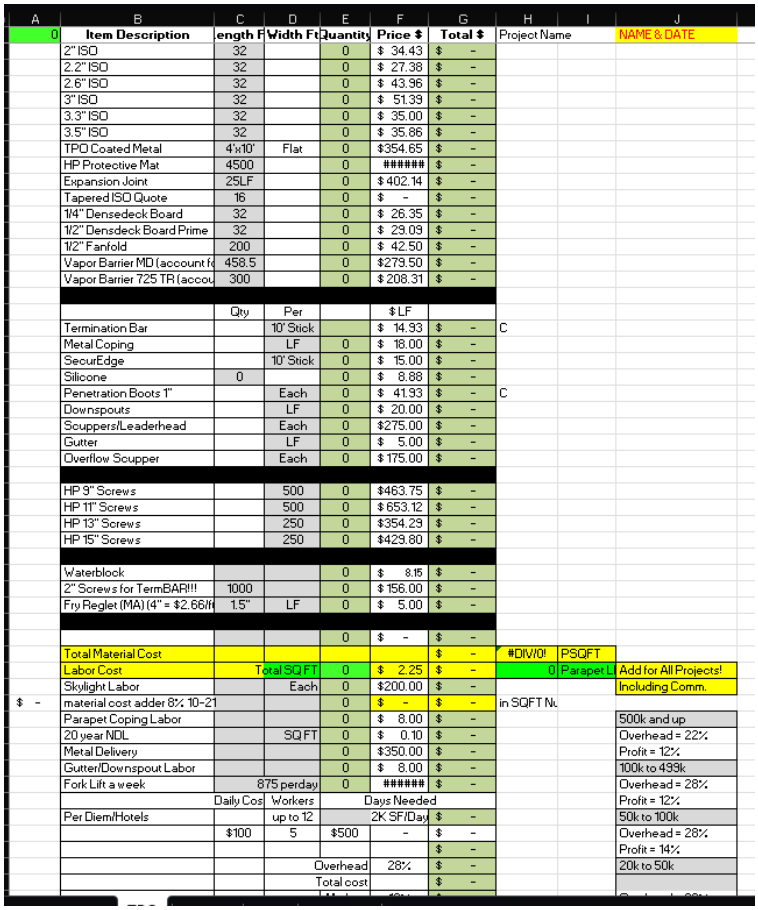
Labor

		DESCRIPTION	MEN	DAYS	HRS/DAY	HOURLY RATE	TTL MAN HOURS	TTL LABOR DOLLARS
5-100	L	Setup/Stock/Cleanup	0	0.0	0.0	63.00	0.00	0.00
5-100	L	Stock Insulation	0	0.0	0.0	63.00	0.00	0.00
5-200	L	Tearoff 18-20SQ/Day				63.00	0.00	0.00
5-300	L	Lay Insulation	0	0.0	0.0	63.00	0.00	0.00
5-400	L	Lay Membrane	0	0.0	0.0	63.00	0.00	0.00
5-510	L	Flash Perimeter & Walls	0	0.0	0.0	63.00	0.00	0.00
5-520	L	Flash Curbs	0	0.0	0.0	63.00	0.00	0.00
5-530	L	Flash Pipes	0	0.0	0.0	63.00	0.00	0.00
5-540	L	Field-Flash/Scuppers/Drains	0	0.0	0.0	63.00	0.00	0.00
5-550	L	Expansion Joint			0.0	63.00	0.00	0.00
5-600	L	Wood/Deck				63.00	0.00	0.00
5-800	L	Cap/Coat/Rock/Gravel				63.00	0.00	0.00
5-800	L	Boom Gravel	0	0.0	0.0	63.00	0.00	0.00
5-850	L	Pavers/Walkpads/Pads	0	0.0	0.0	63.00	0.00	0.00
5-900	L	Supervisor	0	0.0	0.0	63.00	0.00	0.00
5-900	L	PAVERS 14SF/man hour	3	5		32.00	0.00	0.00
5-900	L	Vacuum Ballast $80-$100/sq	3	2		100.00	0.00	0.00
5-900	L	Roof Hatch Labor	2	3		32.00	0.00	0.00
5-900	L	One Part Sealant (2/pocket)	3	5		32.00	0.00	0.00
5-900	L	2" HP-X Perim	3	2		32.00	0.00	0.00
5-900	L	Coping Underlayment	2	3		32.00	0.00	0.00
		Squares Per Day	0	5	0			
		Total Labor Rate	$63.00				0.00	0.00
		"L" Portion	$13.00	PRICE PER SQ		#DIV/0!		#DIV/0!
		"LB" Portion	$50.00	CREW DAYS		0.00		
		% LB	384.615%	MAN/HOURS PER SQ		#DIV/0!		

Miscellaneous

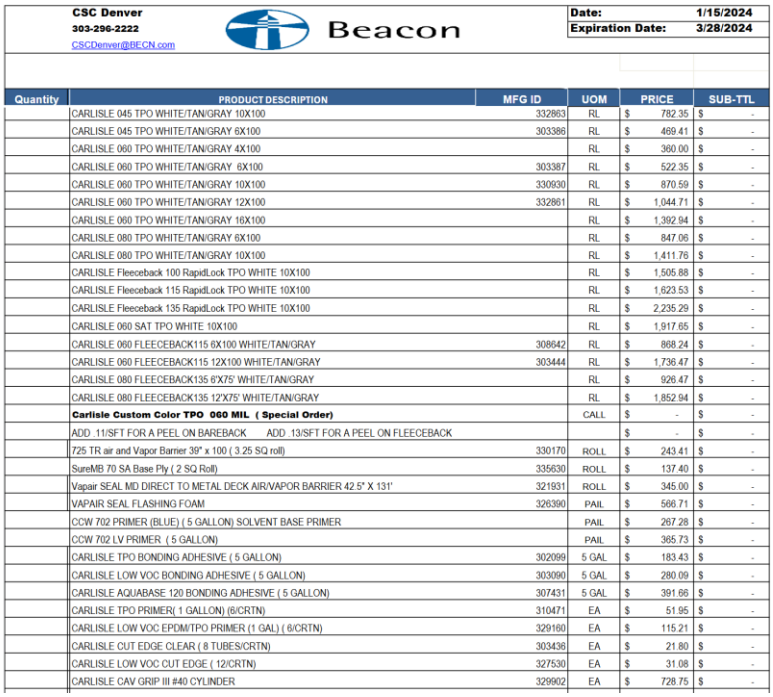
Figure 31. Insulation and Labor Pricing of Original Excel Sheet

3. **Material Quantities to Pricing:**
    - Material quantities are fed into a separate pricing sheet with multiple tabs for various material types (TPO, metal, tile, shingle).
        - There is more to the Excel sheet before going to the separate price sheet but The client wanted me to only look at the above pages for now.
    - Pricing data is manually updated, a tedious process.



A	B	C	D	E	F	G	H	I	J
0	Item Description	ength F	Width Ft	Quantity	Price $	Total $	Project Name		NAME & DATE
	2" ISO	32		0	$ 34.43	$ -			
	2.2" ISO	32		0	$ 27.38	$ -			
	2.6" ISO	32		0	$ 43.96	$ -			
	3" ISO	32		0	$ 51.39	$ -			
	3.3" ISO	32		0	$ 35.00	$ -			
	3.5" ISO	32		0	$ 35.86	$ -			
	TPO Coated Metal	4'x10'	Flat	0	$354.65	$ -			
	HP Protective Mat	4500		0	######	$ -			
	Expansion Joint	25LF		0	$ 402.14	$ -			
	Tapered ISO Quote	16		0	$ -	$ -			
	1/4" Densedeck Board	32		0	$ 26.35	$ -			
	1/2" Densdeck Board Prime	32		0	$ 29.09	$ -			
	1/2" Fanfold	200		0	$ 42.50	$ -			
	Vapor Barrier MD (account fo	458.5		0	$279.50	$ -			
	Vapor Barrier 725 TR (accou	300		0	$ 208.31	$ -			
		Qty	Per		$ LF				
	Termination Bar		10' Stick		$ 14.93	$ -	C		
	Metal Coping		LF	0	$ 18.00	$ -			
	SecurEdge		10' Stick	0	$ 15.00	$ -			
	Silicone	0		0	$ 8.88	$ -			
	Penetration Boots 1"		Each	0	$ 41.93	$ -	C		
	Downspouts		LF	0	$ 20.00	$ -			
	Scuppers/Leaderhead		Each	0	$275.00	$ -			
	Gutter		LF	0	$ 5.00	$ -			
	Overflow Scupper		Each	0	$ 175.00	$ -			
	HP 9" Screws		500	0	$463.75	$ -			
	HP 11" Screws		500	0	$ 653.12	$ -			
	HP 13" Screws		250	0	$354.29	$ -			
	HP 15" Screws		250	0	$423.80	$ -			
	Waterblock			0	$ 8.15	$ -			
	2" Screws for TermBAR!!!	1000		0	$ 156.00	$ -			
	Fry Reglet (MA) (4" = $2.66/ft	1.5"	LF	0	$ 5.00	$ -			
				0	$ -	$ -			
	Total Material Cost					$ -	#DIV/0!	PSQFT	
	Labor Cost		Total SQ FT	0	$ 2.25	$ -	0	Parapet Li	Add for All Projects!
	Skylight Labor		Each	0	$200.00	$ -			Including Comm.
$ -	material cost adder 8% 10-21			0	$ -	$ -	in SQFT Nu		
	Parapet Coping Labor			0	$ 8.00	$ -			500k and up
	20 year NDL		SQ FT	0	$ 0.10	$ -			Overhead = 22%
	Metal Delivery			0	$350.00	$ -			Profit = 12%
	Gutter/Downspout Labor			0	$ 8.00	$ -			100k to 499k
	Fork Lift a week		875 perday	0	######	$ -			Overhead = 28%
		Daily Cos	Workers		Days Needed				Profit = 12%
	Per Diem/Hotels		up to 12		2K SF/Day	$ -			50k to 100k
		$100	5	$500	-	$ -			Overhead = 28%
						$ -			Profit = 14%
				Overhead	28%	$ -			20k to 50k
				Total cost		$ -			

Figure 32. Separate Price Sheet of the Next Excel Sheet



CSC Denver
303-296-2222
CSCDenver@BECN.com

Beacon

Date: 1/15/2024
Expiration Date: 3/28/2024

Quantity	PRODUCT DESCRIPTION	MFG ID	UOM	PRICE	SUB-TTL
	CARLISLE 045 TPO WHITE/TAN/GRAY 10X100	332863	RL	$ 782.35	$ -
	CARLISLE 045 TPO WHITE/TAN/GRAY 6X100	303386	RL	$ 469.41	$ -
	CARLISLE 060 TPO WHITE/TAN/GRAY 4X100		RL	$ 360.00	$ -
	CARLISLE 060 TPO WHITE/TAN/GRAY 6X100	303387	RL	$ 522.35	$ -
	CARLISLE 060 TPO WHITE/TAN/GRAY 10X100	330930	RL	$ 870.59	$ -
	CARLISLE 060 TPO WHITE/TAN/GRAY 12X100	332861	RL	$ 1,044.71	$ -
	CARLISLE 060 TPO WHITE/TAN/GRAY 16X100		RL	$ 1,392.94	$ -
	CARLISLE 080 TPO WHITE/TAN/GRAY 6X100		RL	$ 847.06	$ -
	CARLISLE 080 TPO WHITE/TAN/GRAY 10X100		RL	$ 1,411.76	$ -
	CARLISLE Fleeceback 100 RapidLock TPO WHITE 10X100		RL	$ 1,505.88	$ -
	CARLISLE Fleeceback 115 RapidLock TPO WHITE 10X100		RL	$ 1,623.53	$ -
	CARLISLE Fleeceback 135 RapidLock TPO WHITE 10X100		RL	$ 2,235.29	$ -
	CARLISLE 060 SAT TPO WHITE 10X100		RL	$ 1,917.65	$ -
	CARLISLE 060 FLEECEBACK115 6X100 WHITE/TAN/GRAY	308642	RL	$ 868.24	$ -
	CARLISLE 060 FLEECEBACK115 12X100 WHITE/TAN/GRAY	303444	RL	$ 1,736.47	$ -
	CARLISLE 080 FLEECEBACK135 6'X75' WHITE/TAN/GRAY		RL	$ 926.47	$ -
	CARLISLE 080 FLEECEBACK135 12'X75' WHITE/TAN/GRAY		RL	$ 1,852.94	$ -
	Carlisle Custom Color TPO .060 MIL (Special Order)		CALL	$ -	$ -
	ADD .11/SFT FOR A PEEL ON BAREBACK ADD .13/SFT FOR A PEEL ON FLEECEBACK			$ -	$ -
	725 TR air and Vapor Barrier 39" x 100 (3.25 SQ roll)	330170	ROLL	$ 243.41	$ -
	SureMB 70 SA Base Ply (2 SQ Roll)	335630	ROLL	$ 137.40	$ -
	Vapair SEAL MD DIRECT TO METAL DECK AIR/VAPOR BARRIER 42.5" X 131'	321931	ROLL	$ 345.00	$ -
	VAPAIR SEAL FLASHING FOAM	326390	PAIL	$ 566.71	$ -
	CCW 702 PRIMER (BLUE) (5 GALLON) SOLVENT BASE PRIMER		PAIL	$ 267.28	$ -
	CCW 702 LV PRIMER (5 GALLON)		PAIL	$ 365.73	$ -
	CARLISLE TPO BONDING ADHESIVE (5 GALLON)	302099	5 GAL	$ 183.43	$ -
	CARLISLE LOW VOC BONDING ADHESIVE (5 GALLON)	303090	5 GAL	$ 280.09	$ -
	CARLISLE AQUABASE 120 BONDING ADHESIVE (5 GALLON)	307431	5 GAL	$ 391.66	$ -
	CARLISLE TPO PRIMER(1 GALLON) (6/CRTN)	310471	EA	$ 51.95	$ -
	CARLISLE LOW VOC EPDM/TPO PRIMER (1 GAL) (6/CRTN)	329160	EA	$ 115.21	$ -
	CARLISLE CUT EDGE CLEAR (8 TUBES/CRTN)	303436	EA	$ 21.80	$ -
	CARLISLE LOW VOC CUT EDGE (12/CRTN)	327530	EA	$ 31.08	$ -
	CARLISLE CAV GRIP III #40 CYLINDER	329902	EA	$ 728.75	$ -

Figure 33. Material Pricing List from Distributor

4. **Proposal Generation:**

   - Outputs are used to create a quote/proposal document.

- Typically includes customer details, quote expiration, project name, materials, and system descriptions.

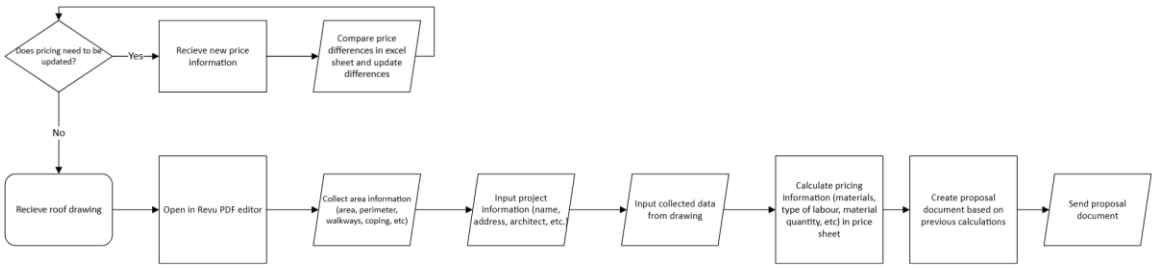


Figure 34. Workflow Flow Chart

Client Interview 2: Information and pain points

Reference Appendix 2 for notes.

Interviewee: Lead Estimator at TPC Roofing

How long does it take to estimate total?

- In total it can take a couple weeks to calculate everything: work labour, cost of materials, etc.

How long does it take to calculate all prices and materials in the current excel sheets?

- Small projects take about 8 hours, large ones about 20 hours

What are the pain points of using this format?

- "The more steps I can cut out the better"
- Biggest thing is updating prices, which comes from different excel docs from warehouses
- Pink excel sheet is only used to get number of materials, then The client uses new sheets and puts in quantities which gives numbers on paid labour and materials for TPC
- He would like to combine excel sheets

Other potential parts that could be implemented into the program?

- Word document: gather key inputs from the estimation (material, gluing or screwing, etc) and put into proposal document template.

Is there any CRM detail needed in the app?

- “If I was in contact with private customers then maybe, but I'm not in contact with a lot of customer databases.”
- Right now there are 2 estimators The client, and someone else who does the same thing as him
- Another person is all about drafting, he draws diagrams on how the roof is going together for workers in the field. He may need to look at The client’s stuff. He works after The client’s estimations and he gives the exact information

How many projects do you work on at a time?

- Rolling schedule on 3-4 projects at a time.
- Bids on 1 project a week and wins about 1 in 10 bids.

Key Challenges Identified

- **Measurement Recording:** No tool to automate blueprint measurements.
- **Spreadsheet Efficiency:** Current workflow relies on manual inputs and complex spreadsheet navigation.
- **Price Updates:** Frequent price changes require tedious manual updates.
- **Proposal Generation:** Manual data transfer into a proposal template is time-consuming.

Project Ideas and Automation Opportunities

1. **Automated Measurement Tools:**
    - Explore existing PDF readers with measuring and recording capabilities.
    - Creating a custom tool is deemed too advanced.
2. **Streamlining Spreadsheets:**
    - Enhance material calculation spreadsheets for clarity and efficiency.
    - Automate equations and inputs based on measurements.
3. **Integrated Material-Price Workflow:**
    - Develop a calculator to connect material quantities with pricing.
    - Automate supplier price updates by scanning new price documents.
4. **Proposal Automation:**
    - Use outputs from material and pricing sheets to auto-generate proposals.
5. **Unified Interface:**
    - Combine all workflows (measurements, materials, pricing, and proposals) into one app for centralized management.

- Focus on a clean, user-friendly interface with minimal automation initially.

Confidence Levels and Future Steps

- **Blueprint Tools (Low):** Research existing PDF readers and measuring tools.
- **Materials and Pricing Workflow (Medium):** Requires better understanding of roofing terms and spreadsheet processes.
- **Proposal Automation (Medium-High):** Likely achievable with structured outputs from previous steps.

Graduation Project Concepts

1. **Supplier Price Updater:**
    - Program to scan new price documents and update the main pricing sheet.
    - Include color-coded indicators for changes, new items, or unresolved items.
2. **Automated Proposal Generator:**
    - Auto-populate proposal templates using materials and prices from earlier calculations.
3. **All-in-One App:**
    - Combine all workflow steps into a centralized program.
    - Focus on organization and user-friendliness with limited automation initially.

Outstanding Questions

- Can material calculations and pricing steps be combined more efficiently?
- What specific tools or systems are used to minimize errors?

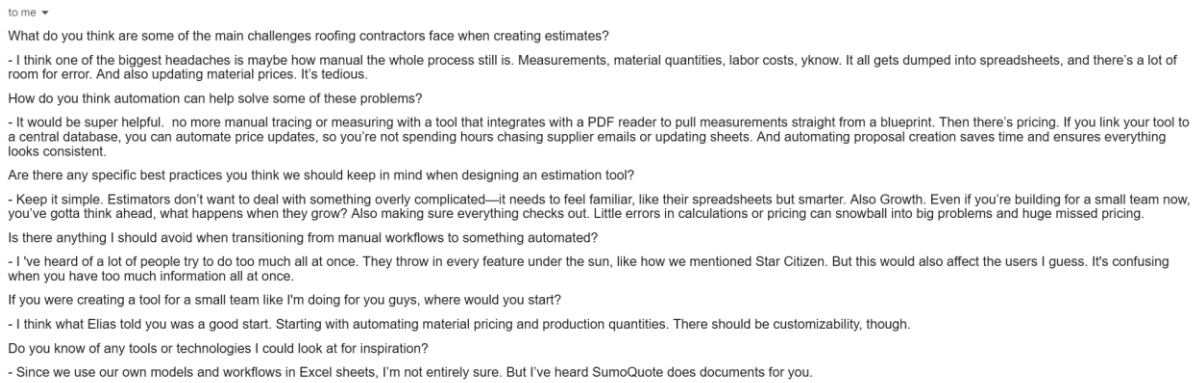
Method 2 – Expert Interview

Justification

The Expert Interview method was selected as a key research approach to complement client interviews by providing broader technical and industry insights. While the client interviews focused on understanding TPC Roofing's specific workflow and challenges, the expert interview offered valuable knowledge about best practices, industry standards, and the potential of automation in roofing estimation. This combination ensured a well-rounded understanding of the problem and informed the design of the prototype.

Due to scheduling issues and distance, this interview was held over email.

Interview



to me

What do you think are some of the main challenges roofing contractors face when creating estimates?

- I think one of the biggest headaches is maybe how manual the whole process still is. Measurements, material quantities, labor costs, yknow. It all gets dumped into spreadsheets, and there's a lot of room for error. And also updating material prices. It's tedious.

How do you think automation can help solve some of these problems?

- It would be super helpful. no more manual tracing or measuring with a tool that integrates with a PDF reader to pull measurements straight from a blueprint. Then there's pricing. If you link your tool to a central database, you can automate price updates, so you're not spending hours chasing supplier emails or updating sheets. And automating proposal creation saves time and ensures everything looks consistent.

Are there any specific best practices you think we should keep in mind when designing an estimation tool?

- Keep it simple. Estimators don't want to deal with something overly complicated—it needs to feel familiar, like their spreadsheets but smarter. Also Growth. Even if you're building for a small team now, you've gotta think ahead, what happens when they grow? Also making sure everything checks out. Little errors in calculations or pricing can snowball into big problems and huge missed pricing.

Is there anything I should avoid when transitioning from manual workflows to something automated?

- I 've heard of a lot of people try to do too much all at once. They throw in every feature under the sun, like how we mentioned Star Citizen. But this would also affect the users I guess. It's confusing when you have too much information all at once.

If you were creating a tool for a small team like I'm doing for you guys, where would you start?

- I think what Elias told you was a good start. Starting with automating material pricing and production quantities. There should be customizability, though.

Do you know of any tools or technologies I could look at for inspiration?

- Since we use our own models and workflows in Excel sheets, I'm not entirely sure. But I've heard SumoQuote does documents for you.

Figure 35 Email Interview

Interviewee: Estimator at TPC Roofing

Reference Appendix 6 for full transcript.

Can you describe common challenges roofing contractors face when creating estimations?

- One of the biggest challenges is ensuring accuracy across multiple stages of the estimation process. From taking measurements to calculating material quantities and labor costs, there's a lot of room for error, especially when using manual workflows like Excel spreadsheets. Frequent updates in material pricing also pose a challenge, as keeping pricing data current requires significant time and effort.

How can automation address these challenges?
Automation can significantly reduce errors and improve efficiency. Ex:

- **Blueprint Analysis Tools:** Automated measurement tools integrated with PDF readers can calculate dimensions directly, reducing the need for manual measurements.
- **Integrated Pricing Databases:** Linking a centralized pricing database to the estimation tool ensures all price updates are automatically reflected in calculations.
- **Proposal Generators:** By automating the proposal creation process, you can save time and ensure consistency in outputs.

These solutions can streamline workflows.

What industry best practices should be considered when designing an estimation tool?
Several best practices:

- The tool should closely align with the estimator's existing workflow to minimize the learning curve.
- **Scalability:** The design should account for future needs, such as additional users, new material types, or enhanced reporting capabilities.
- **Data Validation:** Built-in validation rules can ensure data integrity.

Are there any common mistakes when transitioning from manual workflows to automated systems?

- A common theme is overcomplicating the tool with too many features at the outset.
- Estimators often prefer simplicity.
- Another mistake is neglecting user training; even the best tools fail if users don't understand how to use them effectively.

What advice would you give for creating a roofing estimation tool tailored to a small team like TPC Roofing?

- For a small team, he recommended starting small and focusing on their most significant pain points.
- Automating material pricing and production quantities would deliver immediate value.
- Flexibility is also important—allow users to manually override automated calculations when necessary.

Are there existing tools or technologies you recommend exploring for inspiration?

- **SumoQuote:** Known for its proposal generation capabilities, good user-centric design.
- **CRM Integration Models:** While CRM may not be relevant for TPC Roofing now, examining how tools integrate with CRMs can inspire scalable features for the future.
- TPC roofing use their own models shared amongst the estimators, so insight into other tools is limited.

Key Insights from the Expert Interview

- Automation can enhance efficiency and reduce errors, especially in pricing updates and measurements.
- User-friendly design and scalability are critical for adoption and long-term success.
- Starting with core pain points, such as pricing automation, provides immediate benefits without overwhelming the team.

Method 3 – User Research: Empathy Map

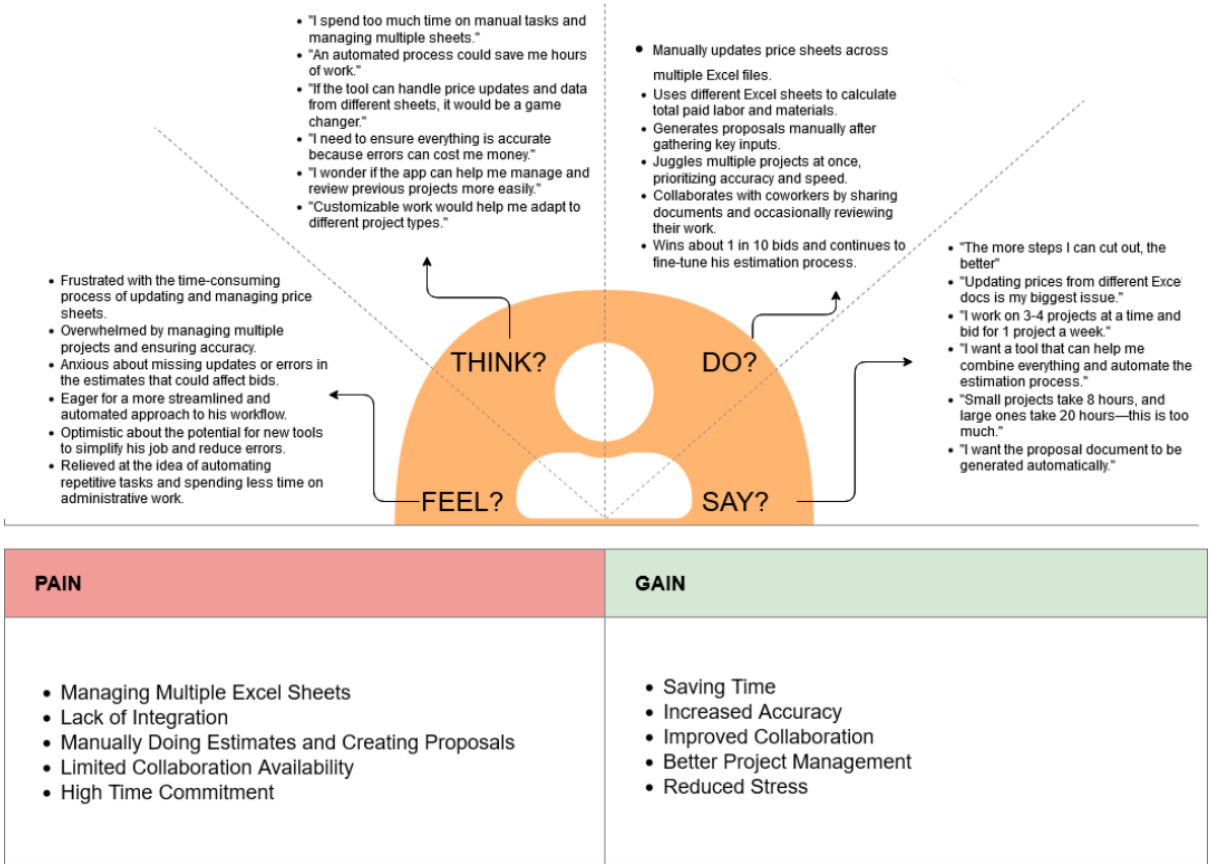


Figure 36 Empathy Map

Justification:

The empathy map captures the thoughts, feelings, actions, and pain points of TPC Roofing's estimators. It highlights key challenges, such as managing multiple Excel sheets and repetitive manual tasks, while identifying opportunities for automation and improved workflow efficiency. This user-centered tool ensured the prototype addressed real-world needs and aligned with the workflow of its end users.

Process:

The empathy map was developed using data from client interviews. Key insights were synthesized into actionable categories to inform design decisions, such as the need for automation, reduced manual effort, and error prevention.

Impact:

The empathy map influenced the prioritization of features like automated calculations and centralized data management.

Method 4 – User Research: Persona

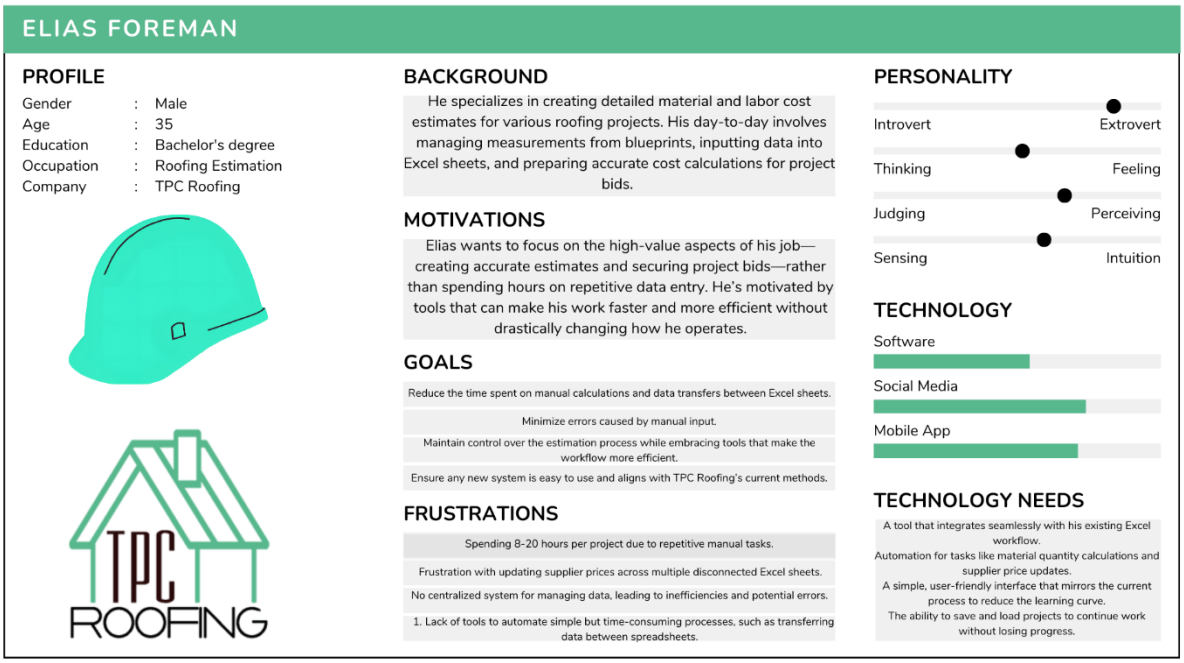


Figure 37 Persona

Justification:

The persona for the client represents the primary user of the estimation tool. It consolidates insights from client interviews and the empathy map, focusing on Elias's specific goals, pain points, and workflow. The persona guided the design of user-friendly features, such as workflow replication, error prevention, and intuitive automation.

Process:

The persona was created based on real-world data to represent the typical estimator's goals, challenges, and workflow. This ensured that design decisions remained relevant and user-focused.

Methods 5 and 6 - SWOT and Competitor Analysis

Justification

The SWOT analysis and Competitor Analysis were integral to this project as they provided a comprehensive framework for evaluating both TPC Roofing's internal strengths and weaknesses, and the external market environment. Together, these methods ensured the project was strategically positioned for success while addressing critical pain points.

SWOT Analysis for TPC Roofing

Strengths	**Weaknesses**
	1. **Limited Initial Market Presence**: As a new entrant in the roofing estimation software

1. **Customization for Specific Needs**: TPC can have the software designed specifically for its operations to meet requirements 2. **Streamlined Processes**: TPC can minimize errors and speed up estimation processing times. 3. **Integration with Existing Systems**: The potential to integrate with TPC's current Excel sheets and other tools can provide a seamless transition. 4. **User-Centric Design**: TPC can prioritize user feedback throughout the development process, ensuring the final product addresses the specific pain points and needs of its estimators and project managers.	market, TPC may struggle to compete against established players with a proven track record and loyal customer base. 2. **Resource Constraints**: Developing a new software solution requires significant resources (time, money, expertise), which is limited. 3. **Learning Curve**: Users may require training to adapt to the new software, which could temporarily slow down productivity during the transition period.
Opportunities 1. **Growing Demand for Digital Solutions**: The increasing need for automation and efficiency in the construction industry presents a significant opportunity for this program. 2. **Market Expansion**: TPC can target underserved markets or specific customer segments (e.g., smaller contractors) that may benefit from a tailored solution, creating niche opportunities. 3. **Partnerships with Suppliers**: Collaborating with material suppliers or other software companies can enhance the value of the estimation tool and provide additional functionalities.	**Threats** 1. **Intense Competition**: Established competitors like Leap and Acculynx already have a strong foothold in the market. 2. **Rapid Technological Advancements**: Keeping up with rapid changes in technology and customer expectations will require continuous investment in development and innovation. 3. **Economic Fluctuations**: Economic downturns can affect the construction industry as a whole, leading to reduced demand for roofing services and estimation software.

Table 6 SWOT Analysis chart

The SWOT analysis provides a structured evaluation of TPC's position in the roofing estimation software market, highlighting its strengths in customization, streamlined processes, and user-centric design. By identifying weaknesses such as limited market presence and resource constraints, it helps anticipate challenges during development. The analysis also pinpoints opportunities, including the growing demand for digital solutions and potential partnerships, while addressing external threats like intense competition and economic fluctuations. This comprehensive approach ensures the project is strategically aligned with market demands and positioned for success.

Alignment with Competitor Analysis

See Appendix for competitor analysis

Strengths:

- **Focused Software Development:** TPC Roofing's strength lies in its targeted approach to developing roofing estimation software without the addition of customer support and CRM features. This allows for a streamlined, efficient product designed specifically for estimating and project management.

- **User-Friendly Interface:** TPC can prioritize a user-friendly design that makes estimation processes easier for roofers, contrasting with competitors that may have more complex systems.

Weaknesses:

- **Lack of Customer Support:** TPC Roofing will not have the same level of customer support or CRM tools as many of its competitors, which may turn off customers who desire relationship management and continuing help.
- **Limited Functionality:** TPC may lose out on chances to incorporate client interactions into the process if customer relations capabilities aren't included, which could be a disadvantage for users used to extensive software.

Opportunities:

- **Niche Market Focus:** TPC can profit on the need for specialized software that specifically caters to the demands of roofing contractors by focusing just on roofing estimating and offering no customer assistance.
- **Partnerships with CRM Solutions:** To increase its attractiveness to users looking for full-service capabilities, TPC could collaborate with current CRM providers to offer integrated solutions without creating its own customer support features.
- **Growth:** The demand for roofing estimation software is confirmed by the growth trends seen across competitors, suggesting a favorable market environment for TPC to enter.

Threats:

- **Intense Competition:** Competitors like Acculynx and JobNimbus, which offer all-in-one solutions including customer support, pose a significant threat to TPC. Users may prefer these comprehensive tools despite their higher complexity and cost.

Conclusion

The TPC SWOT analysis aligns closely with the competitor analysis, helping to clarify the strategic direction TPC Roofing can take in the market. By building on its strengths and tackling its weaknesses, TPC has the potential to make the most of market opportunities while addressing the challenges posed by competitors. This alignment provides a clear focus for product development and market entry.

Unlike many competitors, TPC Roofing doesn’t work directly with private customers, which means there’s no need to include customer support or customer relations features. While this simplifies development and allows for a more targeted approach, it also highlights a gap compared to competitors that do offer these services.

Please see Appendix 3 for notes.

Method 7 - Good, Better, Best Practices for Competitor Estimation Tools

Justification

The Good, Better, Best Practices method was chosen to evaluate competitor features and guide the development of the TPC Roofing estimation tool by prioritizing essential, advanced, and innovative functionalities. This method provides a structured framework to balance immediate user needs with future scalability and innovation.

Good Practices (Basic Features)

These are foundational elements that most competitor tools offer and are essential for meeting user expectations:

Intuitive Interfaces:

- Simplified design for ease of use: seen in Roofr and RoofSnap.
- Tools like iRoofing excel in user-friendly mobile applications for on-site estimations.

Core Functionality:

- RoofSnap's aerial measurement tool and Roofr's instant estimator offer accurate and efficient processes for measuring roofs, entering data, and calculating material requirements.

Basic Workflow Integration:

- Integration with third-party systems, such as CRM tools like JobNimbus or accounting software like Acculynx. This make sure the program works with current workflows.

Better Practices (Advanced Features)

These practices provide enhanced functionality that improves efficiency and user satisfaction:

1. **Customizable Templates:**
    - Customizable workflows and proposals tailored to individual projects, such as Leap's customizable templates and JobNimbus's workflow options.
2. **Multi-Platform Accessibility:**
    - Mobile and desktop applications to support users in different environments (e.g., Acculynx's cloud-based access and iRoofing's mobile app).
3. **Enhanced Collaboration:**
    - Features that allow team members to collaborate on estimates, view edits, and communicate seamlessly (e.g., JobNimbus's project tracking and collaboration features).
4. **Pricing and Payment Management:**
    - Automated payment systems and invoice creation (e.g., Leap's payment tracking) to streamline post-estimation processes.

Best Practices (Innovative and Differentiating Features)

These are cutting-edge practices that set competitors apart in the market:

1. **Automation and AI Features:**
    - AI based tools like iRoofing's roof visualizer and RoofSnap's satellite measurement system reduce manual effort and improve accuracy.
2. **All-in-One Platforms:**
    - JobNimbus and Leap and similar that integrate project management, CRM, estimation, and invoicing into a single platform. This reduces the need for multiple tools.
3. **Scalability and Industry:**
    - Platforms tailored specifically for roofing contractors, such as Acculynx, which ensures features are aligned with industry needs.
    - Scalable solutions that grow with the business, like Leap's affordable pricing tiers.

Enhanced User Support and Customization:

- Providing extensive training and support to reduce learning curves (e.g., JobNimbus's onboarding) and enabling robust customization options for workflows.

Key Takeaways for TPC Roofing

- **Good Practices** to implement: Ensure the tool is user-friendly, focuses on core estimation features, and integrates with existing workflows like Excel.
- **Better Practices** to aim for: Introduce customization options for workflows and templates, ensure accessibility across devices, and enhance collaboration.
- **Best Practices** for future iterations: Integrate AI-based automation for advanced features like proposal generation and aerial measurements, and develop an all-in-one scalable platform tailored to roofing contractors.

Check Appendix 3 for competitor estimation tools on the market and new roofing tech trends.

Method 8 – UI/UX Literature Study

Justification

The UI/UX literature study was essential for establishing a foundation for the future interface design of TPC Roofing's estimation tool. The research made sure the interface would align with best practices, like consistency, clarity, and error prevention, while meeting the specific needs of TPC estimators. Since backend development was more important, this study provided key design principles that will guide future interface development.

Purpose of the Study:

- To gather insights on best practices for designing user-friendly and efficient interfaces.
- To ensure the interface would align with the estimators' needs and the prototype's backend structure.

Key Findings & How They Shaped Design Requirements

1. **Consistency in Interface Design**
    - **Principle:** Users benefit from a consistent layout, standardized terminology, and predictable navigation. (Dam & Teo Yu Siang, 2016)
    - **Application:** The final UI wireframe was structured with clear labeling and workflow-aligned screens.
    - **DR Impact: UR1** – The interface must align with the estimators' current Excel-based workflows to minimize the learning curve.
2. **Error Prevention & Input Validation**
    - **Principle:** Forms and input fields should prevent errors through constraints, inline validation, and clear error messages. (Liu & Kuan Tan, 2008; Banach, 2022)
    - **Application:** Error messages were iteratively improved to help estimators catch input mistakes early, reducing calculation discrepancies.
    - **DR Impact: UR2** – The system must provide clear feedback and prevent incorrect inputs to ensure accurate estimates.
3. **User Control & Efficiency**
    - **Principle:** Users should have control over the workflow, with easy navigation and undo options. (RoofSnap, 2023)
    - **Application:** The wireframe included drop-down menus for material selection, a summary screen for validation, and a back-navigation feature to correct mistakes.
    - **DR Impact: UR3** – The system must allow estimators to efficiently navigate and modify inputs as needed.
4. **Centralized Dashboards for Estimation Tools**
    - **Principle:** Roofing estimation tools benefit from dashboards that integrate multiple input fields, automated calculations, and real-time updates. (AccuLynx, 2024)
    - **Application:** The interface consolidates material selection, production quantities, and pricing updates into a single, structured layout.
    - **DR Impact: QR1** – The system should follow core UI/UX principles for clarity and ease of use.
5. **Scalability & Future Expansion**
    - **Principle:** UI design should anticipate growth, allowing for additional features like multi-user support or cloud-based collaboration. (JobNimbus, 2024)

- **Application:** The UI framework was built with modularity in mind, allowing future expansion with additional tools like supplier price updates and proposal generation.
- **DR Impact: QR4** – The design must remain flexible for future features.

How the Study Influenced the Project:

- Although the interface design has yet to be implemented, the findings provided a conceptual foundation for wireframe development.
- The backend logic (structured workflows, input validation, error handling) was influenced by usability principles to ensure future UI alignment.
- Future iterations will directly integrate these insights, ensuring estimators have a seamless transition from Excel-based workflows to the new system.

This literature study remains a valuable resource for the next phase of the project, where the interface will connect to the backend and deliver a user-friendly experience tailored to TPC Roofing's needs. See Appendix 6 for more notes.

Project Framework Version 1

Initial Project Plan

The initial project plan focused on developing an all-in-one desktop application to automate and centralize TPC Roofing's workflows. The goals included:

- Automating supplier price updates.
- Streamlining material calculations and pricing workflows.
- Creating customizable templates for proposals and invoices.
- Enhancing collaboration through centralized document management.

Initial Problem Statement

The client's workflow for roofing estimations heavily relied on manual tasks using disconnected Excel spreadsheets. Small projects took up to 8 hours, while larger projects required up to 20 hours. Major pain points included:

- Manual pricing updates from supplier data
- Inefficiency in managing data across separate sheets
- Time-consuming manual proposal creation

The proposed solution aimed to streamline these workflows through automation, integrating measurements, material calculations, pricing updates, and proposal generation into a centralized system. This approach would reduce errors, save time, and enhance collaboration, directly addressing these inefficiencies.

Tools and Technologies

Microsoft Excel:

- Integration with current workflows and data storage for cost calculations.

Application Development:

- Visual Studio Code for project prototyping.

UI/UX Design Tools:

- Microsoft Whiteboard for wireframing and prototyping the user interface.

Research Methodology

Interviews & Observations:

- Interview and shadow client and key stakeholders to gather insights into workflow challenges and needs.

Competitive Analysis:

- Research on roofing estimation software competitors (Leap, RoofSnap, etc.) to understand industry standards and gaps to be filled.

Surveys:

- Use questionnaires to gather feedback on initial design concepts from potential users within the company.

UI/UX Best Practices:

- Research and apply UI/UX principles for desktop applications to ensure usability, efficiency, and smooth integration of key features.

Key Terms and Research Questions Version 1

User-Centered Design

- What are core UX needs of estimators at TPC Roofing, and how can these be addressed through a digital interactive tool?

Automation

- How can the current estimation process be condensed and/or automated to improve accuracy and reduce time consumption?

Scalability and Adaptation

- What features would make the tool scale-able and adaptable for use as the company expands?

Data Management

- How can the estimation tool be designed to allow for easy integration of updated supplier prices and materials?

Design Requirements Version 1

Functional Requirements:

- Integration of current excel based workflows.
- Quick material pricing updates from supplier information.
- Generate estimates, proposals, and invoices.
- Automated material and labour cost calculations.

Non-functional Requirements:

- User-friendly interface following core UI principles (visibility, minimal clutter, visual cues).
- Performance and speed focused software that can handle large data sets.
- Scalability focused so that the software can handle larger and more projects and more employee activity in the future.
- Security focused to allow for future encrypted data storage and proper user access controls.

Tools and Technologies

The tools used in this project were chosen to align with the goals of automating TPC Roofing's estimation process and improving workflow efficiency. These choices enabled recreating the client's existing Excel workflows while providing scalability for future improvements.

Python

Python was chosen as the programming language because of its flexibility, simplicity, and extensive library support. Python's ability to integrate seamlessly with both Excel and SQL databases made it ideal for automating manual processes, such as pricing updates and material calculations (McKinney, 2012).

Key libraries used include:

- **pandas:** Data manipulation and handling complex Excel workflows.
- **sqlite3:** Create and manage the database for material pricing and other data.
- **openpyxl:** Reading and writing Excel files, ensuring compatibility with the client's existing tools.

SQLite

SQLite was selected as the database solution for its lightweight, file-based nature (*Learn SQL | CodeCademy*, n.d.). It allowed the project to:

- Centralize pricing and material data for easy updates and retrieval.
- Reduce reliance on multiple disconnected Excel sheets.
- Serve as a scalable solution for potential future expansions of the prototype.

VSCode

Visual Studio Code served as the development environment for writing, testing, and debugging Python code. Its extensions for Python and SQL development made the coding process easier to navigate (Microsoft Corporation, n.d.).

Microsoft Excel

Excel was used mainly as a reference tool during the development process. The project aimed to replicate and improve upon the workflows TPC Roofing currently manages through Excel, ensuring that the transition to the prototype was seamless for the estimators (Microsoft Corporation, n.d.).

Proto.io

Proto.io was used to create a wireframe for a potential user interfaces and its testing. These wireframes provided a visual representation of how the app might look and function, aiding in early-stage client feedback and testing (Proto.io, n.d.).

DB Browser for SQLite

For viewing and validating the SQLite database during development (DB Browser for SQLite, n.d.).

1. **Platform Choice:**
    - Python was chosen for its flexibility and extensive library support, including pandas for data manipulation, sqlite3 for database management, and openpyxl for Excel integration. These libraries enabled the prototype to handle TPC Roofing's complex workflows effectively (McKinney, 2012).
    - SQLite was used as the centralized database, offering a lightweight, file-based solution ideal for managing material pricing and production data (*Learn SQL | CodeCademy*, n.d.).
2. **Backend Influencing UI Design:**
    - The backend defined key inputs, calculations, and outputs that shaped the UI wireframe. This ensured that the interface would align with the system's logic and support all necessary functions for estimators.
3. **User-Centric Workflow:**
    - The prototype replicates TPC Roofing's existing Excel workflows to minimize disruption and the learning curve for estimators. By mirroring familiar processes, the design supports ease of adoption while introducing automation for repetitive tasks.
4. **Scalability:**

- The system was built with scalability in mind, allowing for future expansions like supplier price updates, proposal generation, and user management. SQLite's flexible structure supports the integration of additional features as the system evolves.

Future Iterations

The prototype lays a strong foundation by improving key aspects of TPC Roofing's workflow. Next steps will expand automation, streamline Excel sheets, and coding the user-friendly interface.

Planned Next Steps

3. **Automating Data Flow Between Excel Sheets:**
    - **Immediate Priority:** Automating the transfer of material quantities into a secondary Excel sheet, which calculates quantities and final costs, streamlining a key part of TPC Roofing's estimation process.
    - **Impact:** Eliminates manual data entry, reducing errors and saving significant time for estimators.
    - **Milestone:** This marks the prototype's first implementation of cross-sheet automation, demonstrating its potential to simplify complex workflows..
4. **Programming the User Interface:**
    - A user interface (UI) will be coded based on the wireframe designs created during the prototyping phase. This UI will visually reflect the backend's structure, enabling users to:
        - Input data (e.g., blueprint measurements, material selections).
        - View real-time calculations of production quantities and costs.
        - Save, load, and manage project data seamlessly.
    - The design of the UI will align with TPC Roofing's workflow to ensure ease of use while introducing the benefits of automation.

Future Goals

3. **Proposal Document Integration:**
    - Once the second Excel sheet workflow is functional, the next step in future iterations will be generating client-ready proposal documents directly from the system. While this feature is outside the immediate scope, the current progress lays the groundwork for this addition.
4. **Further Automation and Scalability:**

- Future versions of the prototype will focus on expanding automation capabilities, such as parsing supplier price updates and including other sections of the excel workflow that were too complex for the current project.
- integrating with cloud-based systems to enable real-time collaboration.

Phase 2 - Ideation and Conceptualizing

Method 9 - Moscow Method and Features

Must Have: • Estimation condensation/automation • Pricing info updates • View and upload relevant documents to project • Material database	**Should Have:** • View current projects • View coworkers edits and comments • Proposal template generation based on estimation
Could Have: • View past projects • Security and user permissions • Blank proposal document • Version control for estimates	**Won't Have:** • Mobile access • Security features • Notes for self • Leave notes to other coworkers/have notification page

Table 7 MoSCoW Method

The MoSCoW method was used to rank possible app features based on practicality and fit with the project's objectives. This strategy was guided by the competitive analysis and client interviews in order to solve important pain areas. The prioritized features are organized as follows:Estimation automation

- Pricing sheet information update
- View coworkers edits and comments
- Leave notes to other coworkers/have notifications page
- Estimation process condensed as much as possible
- Notes left for self
- View current projects
- View past projects
- Proposal template generation based on estimation
- Black proposal document
- View and upload relevant documents (excel, blueprints/drawings, price sheet, etc)
- Material database
- Customizable workflow templates
- Version control for estimates (check changes made over time and restore to past versions)

- Mobile access
- Security and user permissions

The resulting prioritized feature set was used to guide concept development and ensure the project stayed focused on the most critical functionalities.

Method 10 - Brainstorming

Justification

Brainstorming was utilized to investigate new concepts, making sure all viable options were evaluated before reducing the scope. This strategy promoted creative thinking and assisted in identifying a variety of solutions to the project's objectives and obstacles.

Application

In this project, brainstorming was employed to develop initial concepts for the roofing estimation tool. This session resulted in three key ideas:

- Project Management App: A collaborative tool featuring internet server connectivity to allow employees to communicate efficiently and track edits in real-time.
- Proposal Generator: Automatically generates Word document proposals based on estimation data such as materials, costs, and project kinds.
- Combined Estimation Dashboard: A consolidated platform that integrates data from several sources, including automated updates from warehouse price documentation.

By considering these ideas, the project team could identify the strengths, weaknesses, and feasibility of each concept before presenting them to the client for feedback. This helped shape the final direction of the project and ensured alignment with client expectations.

Concept 1: Project Management App

This concept explored the development of an internet-based dashboard designed to facilitate real-time collaboration among TPC Roofing estimators. The goal for this was to centralize the estimation process into a single, accessible platform, allowing multiple team members to simultaneously contribute to and track project progress while improving communication and data management.

Idea Conception

During my client interviews, Elias highlighted several difficulties related to collaboration. While estimators frequently work on overlapping projects, they struggle with disconnected workflows, relying on separate Excel spreadsheets, external documents, and email-based communication to manage bids. The lack of a shared workspace results in delays, miscommunication, and duplicated efforts. The client also mentioned previous attempts to implement collaborative tools, but these efforts failed due to usability issues, complexity, and the estimators' reliance on familiar spreadsheet-based workflows.

Proposed Features

To address these pain points, the Project Management App would integrate:

- Real-time coworker activity tracking, displaying who is working on which estimates.

- Commenting functionality, allowing team members to leave feedback directly within the system.
- A unified estimation workflow, combining calculations, material pricing, external documents, and bid proposal templates into a single platform.
- Cloud-based storage, ensuring seamless access and version control across devices.

This would allow TPC Roofing estimators to work more efficiently and collaboratively, reducing the risk of errors caused by fragmented data sources.

Challenges and Feasibility Assessment

While the idea addressed a clear problem, I quickly realized that developing a fully functional project management system was beyond the scope of this project. The technical challenges included:

- Multi-user synchronization, requiring real-time database updates and conflict resolution.
- Cloud-based infrastructure, which would introduce security and data privacy concerns.
- User adoption challenges, as estimators are accustomed to Excel-based workflows, and a fully new system might have required extensive onboarding and training.

Additionally, my current skill set and the project's timeline did not allow for the implementation of such a large-scale system. As a result, I decided to narrow the focus to a more specific aspect of the workflow—material pricing and production quantity calculations, which were identified as the most immediate pain points in my research.

This decision allowed me to create a functional, refined solution that could still streamline the estimation process while being achievable within the given constraints.

Low-Fi Sketches:

Figures 9-11 show first designs of the app's work and home pages. These drawings assisted in identifying the major pain points and prioritizing feature development.

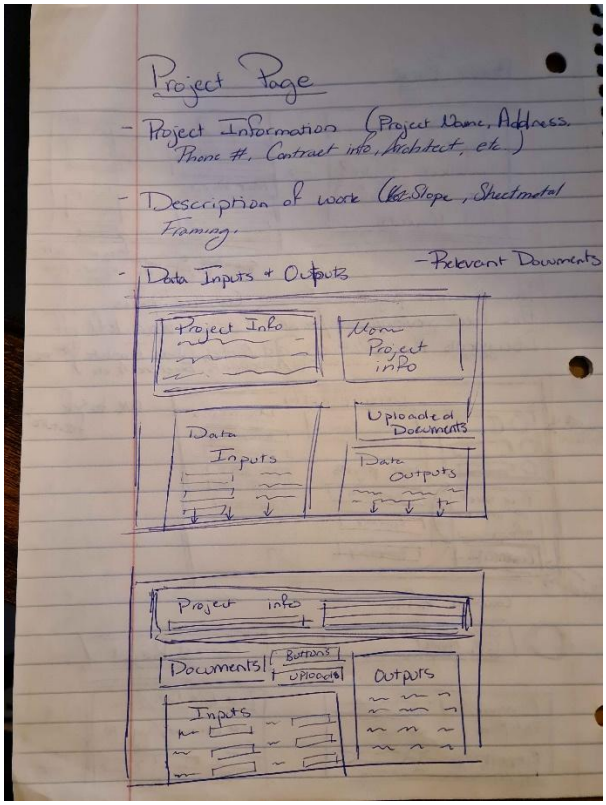


Figure 38 Interface Low-Fi Wireframe Sketch for the Work Page of Initial Project

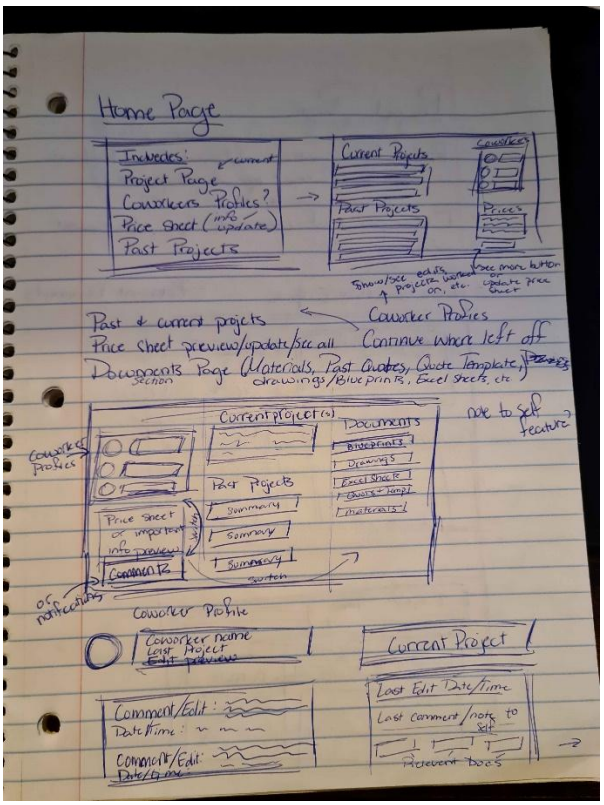


Figure 39 User Interface Low-Fi Sketch of Home Page of Initial Project

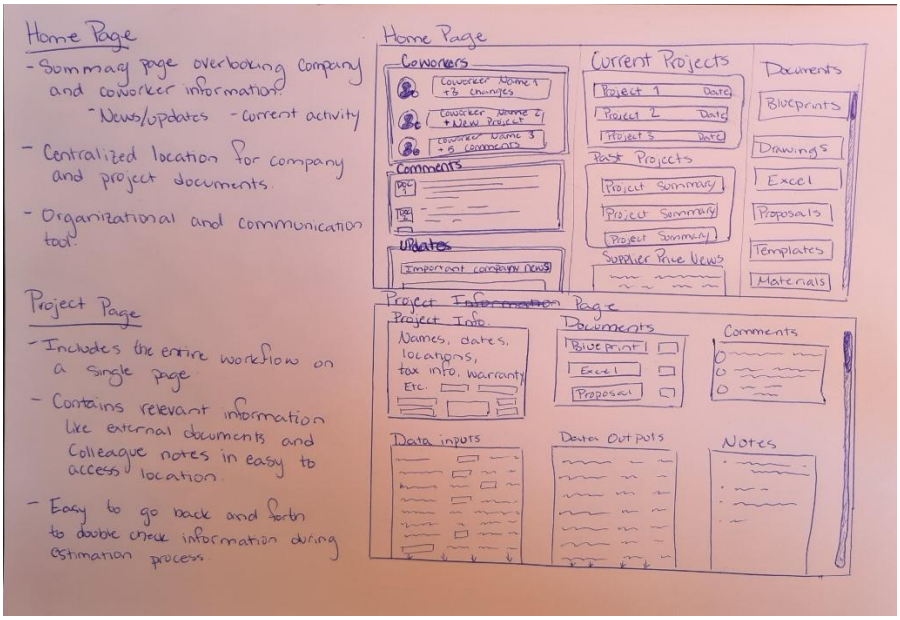


Figure 40 User Interface Low-Fi Sketch of Home Page of Initial Project

Concept 2: Automated Bid Proposal Generation

This concept explored the development of an automated bid proposal generator, a tool designed to take finalized estimation data and use it to create structured, client-ready bid proposals. Unlike the Project Management App, which aimed to improve collaboration across the entire workflow, this concept focused only on the final step—turning an approved estimate into a formatted bid proposal.

Key Difference from the Previous Concept

The Project Management App integrated estimation calculations with collaboration tools and real-time tracking, supporting multiple steps of the workflow and focused overall on accumulating all company estimation information in a single app rather than automating or improving specific sections of the workflow. It included proposal templates, but these were simply pre-formatted documents that users could manually fill out after completing their estimations.

In contrast, the Automated Bid Proposal Generator was designed to fully automate this final step, ensuring that once estimations were approved, a final bid document could be generated without requiring manual entry or formatting. This would allow the estimator to finalize and submit a bid with minimal extra effort.

Idea conception

During client interviews, estimators identified the final proposal stage as a tedious, manual process. While estimates were generated in Excel, the final step required them to:

1. Manually transfer data from spreadsheets into a proposal document.
2. Format and structure the proposal based on the client's template.
3. Double-check for calculation errors or missing values.

Not only was the previous estimation process both time-consuming and error-prone, so was this step of re-recording all of the information with a high risk of misplacing numbers, copying incorrect

values, or overlooking necessary details. Automating this step could reduce human error, improve consistency, and allow estimators to focus on the core estimating work rather than document formatting.

Proposed Features

The Automated Bid Proposal Generator would:

- Pull finalized material and labor calculations directly into a structured bid document.
- Allow users to select from multiple templates depending on client requirements.
- Automatically format data, including subtotals, tax calculations, and final bid amounts.
- Generate a finalized, export-ready PDF.

This system would ensure consistent formatting and accuracy, removing the need for estimators to manually enter values into a separate document.

Why This Concept Was Not Pursued

While this concept would have automated a critical part of the workflow, I ultimately decided not to pursue it due to two major limitations:

1. **It Came at the Very End of the Workflow**
    - This project focuses on improving accuracy and efficiency in the estimation process, but a proposal generator only improves efficiency in document formatting—not the actual estimation calculations.
    - Errors in estimation would still carry over to the bid, meaning this concept wouldn't directly address issues earlier in the workflow, such as material pricing inaccuracies or inefficient data entry.
2. **It Wouldn’t Fully Meet My Project Goals**
    - The core issues in estimation (such as updating materials, improving calculation accuracy, and reducing estimator workload) happen before the bid proposal stage.
    - While this tool would speed up proposal creation, it wouldn’t improve the accuracy or efficiency of the estimation itself—which is the actual problem my client highlighted.

Given these limitations, I decided to focus on an earlier stage of the workflow where improvements would have a more significant impact on accuracy and efficiency.

Concept 3: Combined Estimation Dashboard (Final Chosen Concept)

This concept aimed to develop a centralized estimation dashboard that integrates data from multiple Excel sheets while automating material pricing updates, production quantity calculations, and supplier price changes. Unlike the previous two concepts, which focused on workflow collaboration and bid proposal generation, this solution directly addressed the core inefficiencies in the estimation process, making it the most viable choice.

Key Differences from Other Concepts

1. **Focused on Estimation Calculations and Automation Instead of Team Collaboration and Proposal Generations.**
    - The Project Management App concept attempted to enhance team collaboration but did not fundamentally improve the estimation process itself. It relied on pre-filled templates instead of true automation.
    - The Automated Bid Proposal Generator only impacted the final stage of the workflow and did not optimize the estimation process where errors and inefficiencies occur.
2. **Directly Addresses Accuracy and Efficiency in Estimation**
    - This concept tackles errors in updating material pricing by automatically pulling and updating supplier price lists.
    - Reduces the estimator's manual workload by automating some material and production quantity calculations based on project needs and inputs.
    - Ensures that the most up-to-date pricing data is used in every estimate, preventing discrepancies caused by outdated information.

Idea Conception

During client interviews, estimators mentioned that material prices frequently change, and updating their estimation sheets was time-consuming and prone to human error. Additionally, manually calculating production quantities based on available materials added another layer of inefficiency.

Key problems identified:

- Over 500+ material prices need regular updates, leading to a high risk of using outdated costs in estimates.
- Manual quantity calculations slow down the process and introduce calculation errors.
- Estimators often need to cross-check multiple spreadsheets, making the workflow fragmented and inefficient.

This concept was developed to eliminate these issues by integrating a unified system where all estimation data is centrally managed and automatically updated.

Proposed Features

The concept would include:

- **Automated Material Price Updates:** Pulls the latest supplier pricing data, updating relevant entries in the estimate.
- **Integrated Production Calculations:** Automatically calculates material quantities based on project requirements, reducing estimator input time.
- **Excel Sheet Integration:** Instead of requiring a completely new system, the dashboard replicates the format of existing Excel sheets/their current workflow, ensuring estimators don't need to change how they work or learn anything new other than a different UI.

- **Change Tracking**: Highlights price changes and quantity updates, allowing users to review modifications before finalizing estimates.

By focusing on these pain points, this concept provides the greatest impact on the estimation process while remaining feasible within the project's timeframe.

Why This Concept Was Chosen

While the other concepts addressed specific parts of the workflow, this solution directly improves accuracy and efficiency in estimation—the core objectives of the project.

- The Project Management App was too broad and unfocused on estimation itself.
- The Bid Proposal Generator came too late in the workflow to have a meaningful impact on estimation accuracy.
- The Combined Estimation Dashboard improves the core estimating process, reducing both errors and workload for estimators.

Outcome

The brainstorming session allowed me to explore ambitious ideas while maintaining focus on the project's core objectives. It set the foundation for refining the concepts and prioritizing features that would deliver the most value within the available timeline.

Method 12 – Wireframes

Low-Fi Sketches:

Figures 9-11 show first designs of the app's work and home pages. These drawings assisted in identifying the major pain points and prioritizing feature development.

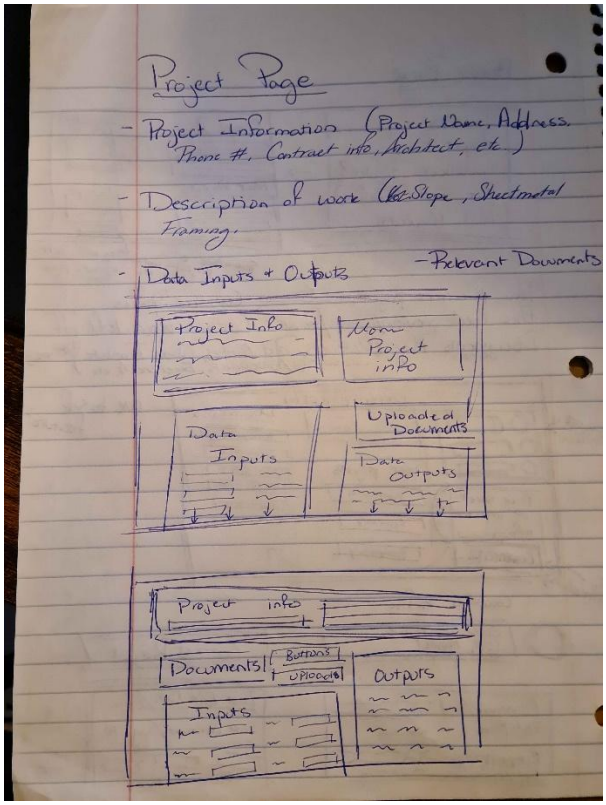


Figure 41 Interface Low-Fi Wireframe Sketch for the Work Page of Initial Project

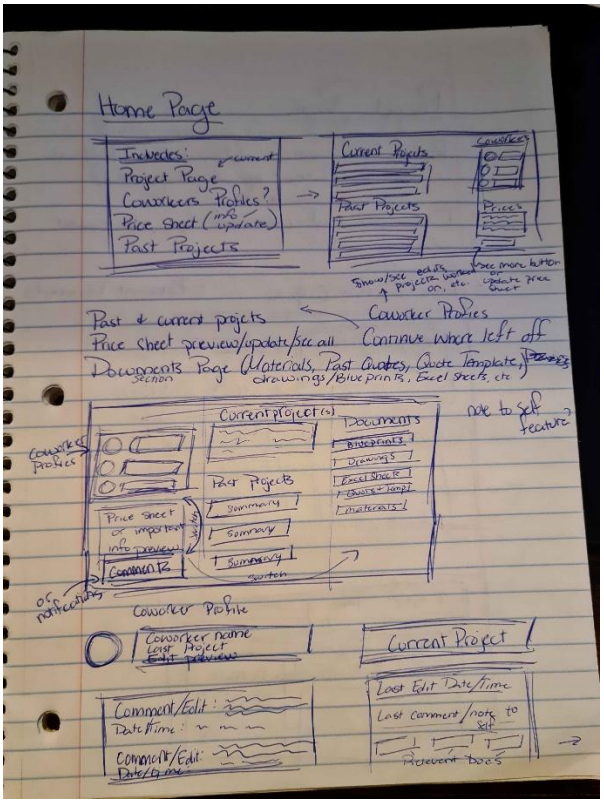


Figure 42 User Interface Low-Fi Sketch of Home Page of Initial Project

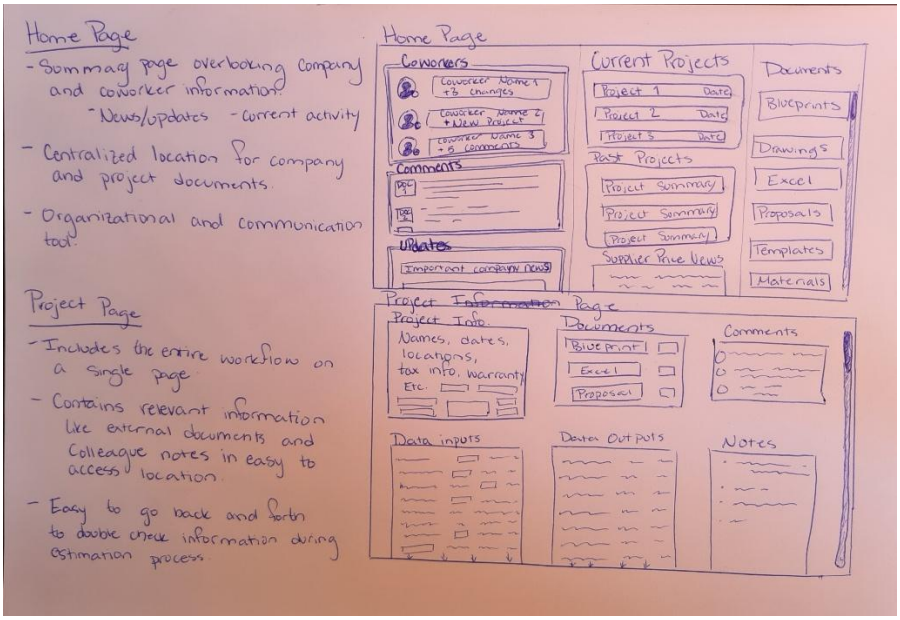


Figure 43 User Interface Low-Fi Sketch of Home Page of Initial Project

Digital Wireframes:

- A second set of wireframes integrated input and concentrated on critical workflows such as material price and manufacturing volumes (Figures 11-12).
- These wireframes acted as a guideline for future development and were approved by the customer.

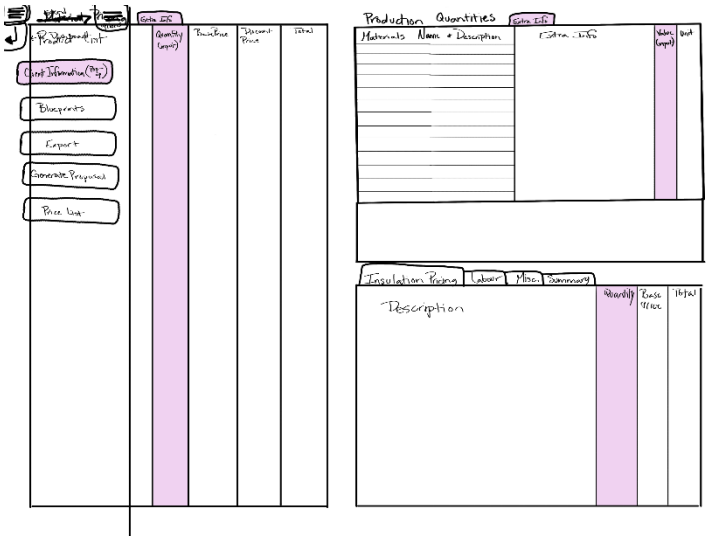


Figure 44 Digital Wireframe Sketch for Project Work Page of Project Version 2

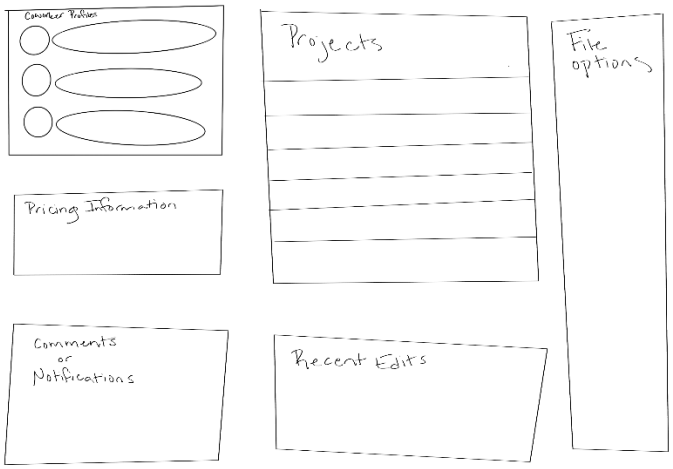


Figure 45 Digital Wireframe Sketch for Home Page of Project Version 2

After creating the second wireframes to visualize the concept, I presented them to my client for feedback and approval. The client appreciated the direction and confirmed that the proposed features addressed key challenges in their workflow. With this approval, I began the prototyping stage, focusing on implementing the broader scope of features from the initial design. However, as development progressed, it became clear that the scope was still too large to complete within the given timeframe.

During a subsequent review, I was advised to narrow the focus further to ensure a functional prototype and a clearer focus for the overall project. At this point, I decided to limit the scope to the Production Quantities and Material Pricing sections of the workflow. These areas were identified as the most critical pain points for the client and offered the highest potential for impact. This change required stopping work on the broader prototype where I was working on an automatic price sheet updater and pivoting to the current prototype in the above report, which aimed to replicate and automate key Excel sheet functionalities.

To prioritize functionality, I decided to step away from designing a full user interface and instead focused on building a robust backend in Python. The plan was to develop a simple placeholder UI that could be integrated with the backend at a later stage if time allowed. This decision helped me allocate more time to coding/learning python and refining the core automation and calculation features while keeping the interface flexible for future iterations.

Method 13 – SCAMPER Method

Justification

The SCAMPER method (Appendix 6) (Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Rearrange) was applied to refine the initial concepts generated during brainstorming. This method provided a structured approach to critically evaluate and improve the ideas, ensuring they were feasible and aligned with client requirements.

Application

The brainstorming concept refinement process was heavily influenced by feedback from my supervisor and the mid-term review (See Appendix 4). During the mid-term assessment, it became

evident that the initial concept for an all-in-one app was unrealistic given the project's time frame and complexity. So my supervisor suggested focusing on the most essential pain points discovered during the research phase, like automating pricing updates and combining material and production quantities computations into a unified system.

The brainstorming session turned to prioritizing features that were relevant to the limited focus. The SCAMPER method was very effective in re-evaluating and improving these concepts based on feedback, ensuring that the project could produce meaningful results while remaining practical within the available period. This iterative strategy kept the project adaptive and responsive to feedback from important stakeholders.

SCAMPER Method for Refinement

After selecting the concept, the SCAMPER method was applied to further refine and optimize the design. SCAMPER is a creative thinking tool that encourages the exploration of alternatives and improvements (Dam & Teo Yu Siang, 2016; Serrat, 2017). Instead of generating an entirely new concept, SCAMPER helped in making the final concept more feasible, user-friendly, and scalable. Key refinements included:

- Substituting the automated bid proposal generator with a stronger focus on automating pricing updates, addressing the most immediate user pain point.
- Combining elements from the Project Management App (real-time collaboration) with the Estimation Dashboard (data automation) to form a cohesive system.
- Adapting the all-in-one app idea to fit within existing Excel workflows, ensuring easier user adoption.
- Modifying the user interface for simplicity, aligning it with the estimator’s workflow rather than introducing a completely new system.
- Eliminating overly ambitious features (mobile access, advanced CRM tools) to keep the project within the feasible scope.
- Rearranging workflow priorities by making Production Quantities and Material Pricing the primary focus areas rather than tackling the entire estimation process at once.

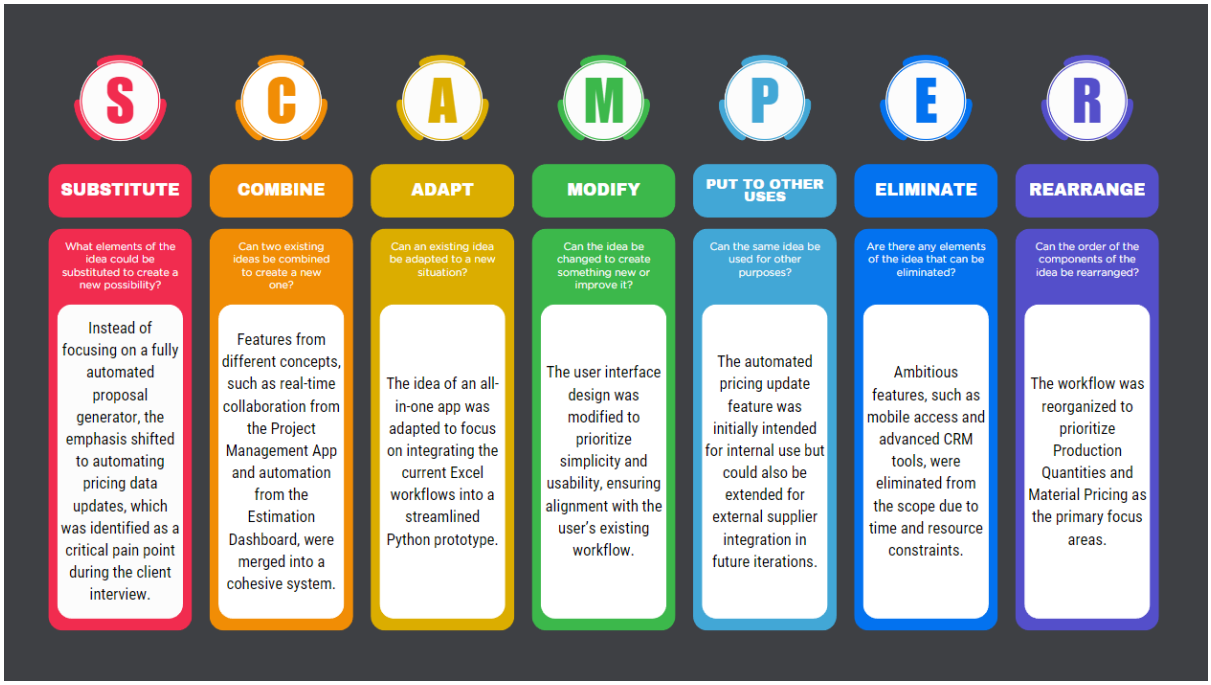


Figure 46 SCAMPER Chart

These refinements ensured that the final prototype remained technically achievable, aligned with user needs, and capable of future scalability.

Phase – 3 Prototyping

Method 14 - Rapid Prototyping

Description

Rapid prototyping is a design workflow that focuses on creating and testing small, functional components of a project to refine and validate ideas incrementally. This approach allows designers and developers to address specific challenges, test assumptions, and gather feedback without committing to a full-scale implementation (Maze, 2023).

See Appendix 8 for notes on which language to use and what file types (SQL JSON ETC).

Justification

In this project, rapid prototyping was essential due to the iterative nature of development and the need to refine processes with minimal Python experience at the time. By breaking down the project into smaller, testable components, the prototypes allowed for incremental progress, skill-building, and targeted feedback. This approach ensured that each iteration contributed to solving the client's workflow pain points while providing a foundation for future improvements.

Execution

Iteration 1: Multi-Format Automation (PDF + Excel)

1. **PDF Parsing with PDFPlumber:**
    - Small scripts were written to test the extraction of tabular data from PDFs.
    - Issues such as misaligned rows and merged cells were identified early, leading to repeated prototyping with different configurations and methods.
    - Debugging these scripts provided valuable insights but ultimately revealed the limitations of PDF parsing for this project.
2. **Excel File Handling:**
    - Parallel to PDF testing, small-scale code snippets were developed to parse Excel data using pandas and openpyxl.
    - Early tests included reading a single row or column to verify data extraction accuracy and column renaming.
    - Prototypes were adjusted iteratively to handle null values, filter unnecessary columns, and test SQL database integration with limited datasets.
3. **Database Integration:**
    - The SQL database schema was prototyped incrementally.
    - Initial tests involved inserting a single row into the database to ensure proper data handling.
    - Iterative updates added logic for checking duplicate records and updating prices only when necessary.

Iteration 2: Excel-Only Focus

1. **Single-Input Testing:**
    - After pivoting to focus exclusively on Excel files, rapid prototypes were created to handle single inputs.
    - For example, parsing a single material from the Excel file was tested to verify database queries and updates.
2. **Database Queries:**
    - Rapid prototypes were developed to retrieve data for one material at a time, ensuring that SQL queries worked as intended and returned accurate results.
    - This approach helped validate database logic before scaling up to handle full Excel files.
3. **Error Handling:**
    - Prototyping focused on testing specific error scenarios, such as uploading unsupported file types or handling missing fields in the Excel data.
    - Small code snippets were written to trigger and debug these errors, allowing for incremental improvements in the error-handling logic.
    -

Results for Iteration 3: Placeholder Interface and Backend Production

1. **Testing Individual Lines of Excel Logic:**
    - To ensure the accuracy of the Python replication of Excel workflows, single lines of Excel logic were translated into Python functions and tested independently. For example, parsing one row from the production quantities section was prototyped to validate correct calculations and data handling.
2. **Dropdown Menu Prototyping:**
    - A small-scale test was conducted to simulate the dropdown menu functionality from Excel. This included ensuring that user selections dynamically impacted the backend calculations and database queries.
3. **Input Validation and Calculation Testing:**
    - Prototypes were developed to test whether a single user input would correctly trigger the associated backend calculations, and those calculations were done correctly. These small-scale tests ensured that formulas and logic mirrored Excel's behavior, preventing errors in the final implementation.

Iteration 4: Wireframe Prototyping

- **Initial Sketches:**
    - Low-fidelity sketches were created to visualize workflows and interfaces.

    - Feedback on these sketches meant changes to create alignment with user needs before committing to detailed designs (Appendix 9).

- **Color Variants:**
    - Two wireframe versions (light mode and dark mode) were quickly created to test aesthetic preferences.
    - This process was not purely functional but allowed for rapid exploration of visual design options based on stakeholder feedback.

Results

1. **Iteration 1 Results:**
    - Identified limitations of PDF parsing, leading to a pivot to Excel-only functionality.
    - Developed a functional prototype capable of parsing Excel data and inserting it into an SQL database.
2. **Iteration 2 Results:**
    - Verified database queries and updates for individual materials through single-input prototypes.
    - Improved error handling and parsing logic for Excel files.
3. **Iteration 3 Results:**
    - Tested single lines of Excel logic to ensure accurate replication in Python.
    - Prototyped dropdown menus and inputs to validate backend interactions and calculations.
    - Incrementally validated components to align backend functionality with Excel workflows.
4. **Wireframe Results:**
    - The dark mode wireframe, featuring TPC Roofing's turquoise-green branding, was chosen based on feedback for its modern and cohesive aesthetic.

Conclusion

The creation and improvements of the project were greatly aided by the process of rapid prototype. It was able to make overall progress while continuously identifying and addressing problems early on. This approach of gradually testing smaller parts of the project confirmed that all aspects were working as intended and met the objectives of the project. In this manner, the prototypes corrected the iterations and, at the same time, laid the groundwork for what was to come.

Method 15 – System Flowchart

Description

System flowcharts are a tool used to map workflows, interactions, and changes in a system over time (*System Flowchart: Definition, Application, Benefits, Symbols and E*, n.d.). This method visually represents the steps, transitions, and dependencies within a process, helping to identify inefficiencies and refine system logic.

Justification

The TPC Roofing estimation process workflow system diagram flowcharts were the primary means to visualize the work processes. It helped clarify how the data inputs from Excel to the database updates and the data calculations took place, ensuring that the Python prototype reflected the Excel processes exactly.

Execution

Iteration 3 – Production Quantities and Material Pricing:

- Developed flowcharts to model user inputs (e.g., dropdown selections) and their impact on backend calculations.
- Showcased interactions between placeholder interface elements and database queries.

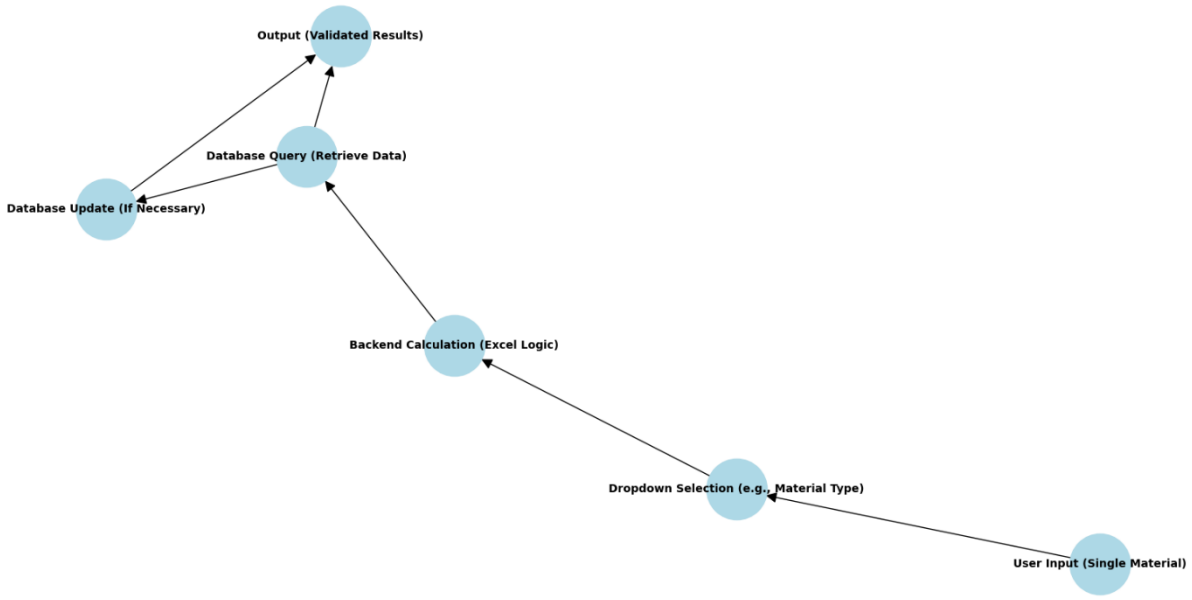


Figure 47 System Flowchart

Results

- Visualized the end-to-end workflows for each iteration, clarifying data flows and dependencies.
- Improved alignment between backend logic and user activities by mapping specific transitions and results.

Method 16 – Experience Prototyping

Description

Experience prototyping involves simulating real user interactions with a prototype to validate its functionality, usability, and alignment with user needs (Zhou, 2021b). This method helps identify issues early and refine the design based on feedback and observed workflows.

Justification

Experience prototyping was critical for ensuring that the prototype aligned with the expectations and workflows of TPC Roofing's estimators. By testing small components, such as dropdown menus, single-line calculations, and error messages, I could replicate real-world scenarios and gather valuable feedback for improvement. This iterative approach ensured that both backend functionality and user experience were well-refined before scaling up.

Execution

1. **Single Input Testing:**
    - Simulated user input of a single material to test parsing and database updates.
    - Validated whether backend calculations mirrored Excel workflows accurately.
2. **Dropdown Menu Functionality:**
    - Simulated dropdown menu interactions to test dynamic database queries and their impact on calculations.
    - Focused on ensuring that user selections influenced the correct backend processes.
3. **Error Handling Simulation:**
    - Tested scenarios where users entered invalid inputs or uploaded unsupported files.
    - Assessed clarity and usefulness of error messages, refining them based on user feedback.
4. **User Workflow Simulation:**
    - Simulated complete user workflows, including data entry, dropdown selections, and output validation.
    - Incorporated informal feedback loops to identify friction points and improve interactions.
5. **Feedback Integration:**
    - Early user testing highlighted areas for improvement, such as clearer instructions and streamlined workflows. These were addressed iteratively during prototyping.

Results

- **Improved Backend Accuracy:**
    - Single material input tests validated that Python replicated Excel calculations accurately.
- **Enhanced Usability:**

- Refinements to dropdown menus ensured seamless interactions and backend integration.

- **Clearer Error Handling:**
  - Feedback led to clearer, actionable error messages and terminal prompts.
- **User Alignment:**
  - Simulated workflows confirmed that the prototype addressed key pain points in TPC Roofing's estimation process.

Iteration 1&2 Code

```
from tkinter import Tk, Label, messagebox
from tkinterdnd2 import TkinterDnD, DND_FILES
from pathlib import Path
import pandas as pd
import sqlite3

def parse_excel(file_path):
    #read the Excel file, skipping irrelevant header rows
    data = pd.read_excel(file_path, skiprows=5)  #adjust skiprows to start reading from actual data
    print("Filtered Data read from Excel:", data.head())  #debugging, filtering

    #filter to only include the necessary columns, renaming as needed - empty/null columns were being added
    data = data.rename(columns={
        "PRODUCT DESCRIPTION": "product_description",
        "Quantity": "quantity",
        "UOM": "UOM",
        "PRICE": "price"
    })

    #select only the relevant columns
    data = data[["product_description", "quantity", "UOM", "price"]]

    #fill nulls with empty strings for non-numeric fields and 0 for price
    data = data.fillna({"product_description": "", "quantity": "", "UOM": "", "price": 0.0})

    #convert dataframe rows into a list of lists for SQL insertion
    data_rows = data.values.tolist()
    print("Parsed data rows:", data_rows)  #debugging, see what's being recorded
    return data_rows

def update_or_insert_sql(data_rows):
    #connect to the sql database
    connect = sqlite3.connect('roofing_materials.db')
    cursor = connect.cursor()

    #create table if it doesn't exist
    cursor.execute('''CREATE TABLE IF NOT EXISTS RoofingMaterials (
                        id INTEGER PRIMARY KEY AUTOINCREMENT,
                        product_description TEXT,
                        quantity TEXT,
                        UOM TEXT,
                        price REAL)''')

    for row in data_rows:
        product_description, quantity, UOM, price = row

        #check if the product already exists in the database
        cursor.execute('''SELECT id, price FROM RoofingMaterials WHERE product_description = ?''', (product_description,))
```

```
49.         existing_product = cursor.fetchone()
50.
51.         if existing_product:
52.             #if product exists, update the price if it has changed
53.             existing_price = existing_product[1]
54.             if existing_price != price:
55.                 print(f"Updating price for {product_description} from {existing_price} to
{price}")
56.                 cursor.execute('''UPDATE RoofingMaterials SET price = ? WHERE
product_description = ?''', (price, product_description))
57.         else:
58.             #if product doesn't exist, insert it
59.             print(f"Inserting new product: {product_description}")
60.             cursor.execute('''INSERT INTO RoofingMaterials (product_description, quantity,
UOM, price)
61.                               VALUES (?, ?, ?, ?)''', row)
62.
63.     #commit the changes and close the connection
64.     connect.commit()
65.     connect.close()
66.     print("Database update successful.")
67.
68. def on_drop(event):
69.     #get the dropped file path and clean it - it wouldn't find path due to spaces and
special characters in file name
70.     raw_file_path = event.data.strip().replace("{", "").replace("}", "")
71.     file_path = Path(raw_file_path)
72.
73.     #handle potential path issues
74.     if not file_path.exists():
75.         file_path = Path(" ".join(raw_file_path.split()))
76.
77.     #handle excel files
78.     if file_path.suffix.lower() == ".xlsx" and file_path.exists():
79.         #parse excel to get structured data
80.         data_rows = parse_excel(file_path)
81.         update_or_insert_sql(data_rows)  #insert or update data in sql
82.         messagebox.showinfo("File Processed", f"Content saved to roofing_materials.db")
83.     else:
84.         messagebox.showinfo("Unsupported File", "Only Excel (.xlsx) files are currently
supported.")
85.
86. #setting up the main window using tkinter
87. window = TkinterDnD.Tk()
88. window.title("Drag and Drop Files Here")
89. window.geometry("400x300")
90.
91. label = Label(window, text="Drop price sheet document here", font=("Arial", 18))
92. label.pack(expand=True, fill="both")
93.
94. #enable the window to accept dropped files
95. window.drop_target_register(DND_FILES)
96. window.dnd_bind('<<Drop>>', on_drop)
97.
98. window.mainloop()
```

Figure 16 Prototype 1 Code

Iteration 3 Main Prototype Code

main.py

The main script serves as the entry point for the application. It:

- Handles user interaction for starting a new project or loading an existing one.
- Coordinates the execution of the **Production Quantities** and **Material Pricing** workflows by interacting with their respective modules.

- Collects and saves user inputs, calculates results, and displays final outputs in the terminal.
- Manages file I/O for project data using JSON.

```
from production_quantities import ProductionQuantity
from project_io import save_to_json, load_from_json
from materials_query import fetch_materials_by_category, fetch_units
from material_pricing import MaterialPricing

#Path to the SQL database
db_file = "Price_List.db"
available_units = fetch_units(db_file)  #Fetch available units from the database

#Project initialization
choice = input("Do you want to load an existing project or start a new one? (load/new):
").strip().lower()

if choice == "new":
    project_file = input("Enter file name: ").strip()
    input_store = {}
    print("Starting a new project.")
else:
    project_file = input("Enter file name: ").strip()
    input_store = load_from_json(project_file)
    print("Loaded Inputs: ")
    for key, value in input_store.items():
        print(f'{key}: {value}')

# ================================================================================
# Production Quantities Section
# ================================================================================

#List display of measurement input area
measurements_data = [
    {"description": "Total Roof Area", "sub_description": " - Tear-Off", "is_manual": True,
"unit": "SQ FT"},
    {"description": "Total Roof Area", "sub_description": " - Insulation", "formula":
"input_store['Total Roof Area - Tear-Off']", "unit": "SQ FT"},
    {"description": "Total Roof Area", "sub_description": " - Membrane", "formula":
"input_store['Total Roof Area - Tear-Off']", "unit": "SQ FT"},
    {"description": "Total Roof Area", "sub_description": " - Coating", "is_manual": True,
"unit": "SQ FT"},
    {"description": "Perimeter", "is_manual": True, "unit": "LN FT"},
    {"description": "Area of Perimeter", "sub_description": " - Wall Height (LF)",
"extra_input": "Wall Height (LF)",
     "formula": "previous_value * input_store['Wall Height (LF)']", "unit": "SQ FT"},
    {"description": "Number of Curbs", "is_manual": True, "unit": "EA"},
    {"description": "Curb Perimeter", "is_manual": True, "unit": "LN FT"},
    {"description": "Area of Curb Perimeter", "formula": "previous_value * 1.5", "unit":
"SQ FT"},
    {"description": "LN FT of Field Seam", "sub_description": " - Sheet Width (FT)",
"extra_input": "Sheet Width (FT)",
     "formula": "input_store['Total Roof Area - Insulation'] / (input_store['Sheet Width
(FT)'] * 100) * (100+input_store['Sheet Width (FT)'])",
     "unit": "LN FT"},
    {"description": "LN FT of Walkway", "is_manual": True, "unit": "LN FT"},
    {"description": "SF of Walkway", "is_manual": True, "unit": "SQ FT"},
    {"description": "Number of Roof Drains", "is_manual": True, "unit": "EA"},
    {"description": "Number of Scuppers", "is_manual": True, "unit": "EA"},
    {"description": "Number of VTR'S", "sub_description": " - SMALL", "is_manual": True,
"unit": "EA"},
    {"description": "Number of VTR'S", "sub_description": " - LARGE / Universal",
"is_manual": True, "unit": "EA"},
    {"description": "Taper Area", "is_manual": True, "unit": "SQ FT"},
    {"description": "Insulation Fasteners & Plates", "sub_description": " - Fasteners /
Board (EA)", "extra_input": "Fasteners / Board (EA)",
```

```
 51.      "formula": "((input_store['Total Roof Area - Membrane'] / 32) * input_store['Fasteners
/ Board (EA)']) * 1.05", "unit": "EA"},
 52.     {"description": "Perimeter Securement", "formula": "(input_store['Perimeter'] +
input_store['Curb Perimeter']) * 1.05", "unit": "LN FT"},
 53.     {"description": "Expansion Joint", "sub_description": " (Wall)", "is_manual": True,
"unit": "LN FT"},
 54.     {"description": "Expansion Joint", "sub_description": " (Field)", "is_manual": True,
"unit": "LN FT"},
 55.   # FINISH FORMULA LATER WHEN MATERIAL PRICING IS FINISHED - REQUIRES INPUT FROM THERE
 56.   # {"description": "Parapet TPO Rolls", "sub_description": " -Tbar / Drip Edge  :  Convert
to Rolls", "formula": "input_store['Perimeter'] - ", "unit": "LN FT"},
 57.     {"description": "Coating", "is_manual": True, "unit": "SQ FT"},
 58.     {"description": "Pavers", "is_manual": True, "unit": "SQ FT"},
 59.     {"description": "Paver Perimeter", "is_manual": True, "unit": "LN FT"},
 60.     {"description": "725TR", "is_editable": True, "is_manual": True, "unit_dropdown": True,
"Unit": "EA"},
 61.     {"description": "CAV Grip", "is_editable": True, "is_manual": True, "unit_dropdown":
True, "Unit": "EA"},
 62.     {"description": "Molded Sealant Pocket", "is_editable": True, "is_manual": True,
"unit_dropdown": True, "Unit": "EA"},
 63.     {"description": "One Part Sealant (2/pcket)", "is_editable": True, "formula":
"input_store['Molded Sealant Pocket'] / 2", "unit_dropdown": True, "Unit": "EA"},
 64.     {"description": '2" HP-X Perim', "is_editable": True, "formula":
"(input_store['Perimeter'] * 2) * 1.05", "unit_dropdown": True, "Unit": "EA"},
 65.     {"description": "Coping Underlayment", "is_editable": True, "is_manual": True,
"unit_dropdown": True, "Unit": "EA"},
 66. ]
 67.
 68. # Create objects and collect inputs
 69. production_quantities = []
 70. previous_value = None
 71. for data in measurements_data:
 72.     pq = ProductionQuantity(
 73.         data.get('description', ''),
 74.         data.get("sub_description", ""),
 75.         data.get("is_manual", False),
 76.         data.get("extra_input", ""),
 77.         data.get("value", 0),
 78.         data.get("unit", ""),
 79.         data.get("formula", None),
 80.         data.get("is_editable", False),
 81.         data.get("unit_dropdown", False),
 82.     )
 83.     pq.set_value(input_store, previous_value, available_units)
 84.     production_quantities.append(pq)
 85.     previous_value = pq.value
 86.
 87. # Save project data to JSON file
 88. save_to_json(project_file, input_store)
 89.
 90. # Test that it works
 91. print("\nFinal Results:")
 92. print(f"{'=' * 20} Production Quantities {'=' * 20}")
 93. print("Final Input Store Values:")
 94. for key, value in input_store.items():
 95.     print(f"{key}: {value}")
 96.
 97. for pq in production_quantities:
 98.     print(f"{pq.description}{pq.sub_description}: {pq.value} {pq.unit}")
 99. print("=" * 40)
100.
101. # ============================================================================
102. # Material Pricing Section
103. # ============================================================================
104.
105. distributor_discount = float(input("Enter Distributor Discount (as a decimal, e.g., 0.10
for 10%): "))
106. direct_discount = float(input("Enter Direct Discount (as a decimal, e.g., 0.05 for 5%): "))
107. input_store["Distributor Discount"] = distributor_discount
108. input_store["Direct Discount"] = direct_discount
```

```
109.
110. # Material pricing and input
111. materials_data = [
112.     {
113.         "description": "Membrane Type Material 1",  # User selects from Membrane Type
category in SQL
114.         "category": "Membrane Type",  # Category in the SQL database
115.         "material_ids": [1, 2, 7, 8],  # Restrict to material IDs 1-4
116.         "sub_category": "Field & Seams",  # Sub-description
117.         "extra_input": True,  # Indicates user can adjust the extra input value
118.         "extra_input_formula": "selected_material[2]",  # Default: Price from SQL database
119.         "quantity_formula": "(input_store['Total Roof Area - Membrane'] + (input_store['LN
FT of Field Seam - Sheet Width (FT)'] / 2)) * 1.05",  # Formula for Quantity
120.         "is_manual_base_price": True,  # User inputs the base price
121.         "base_price_reference": None, #No reference for this material
122.         "discount_price": 0,  # Default user input
123.         "discount_price_input": True, #prompt for discount price input
124.         "total_formula": "(quantity * base_price) + discount_price"  # Formula for Total
125.     },
126.     {"description": "Membrane Type 1 Subcategory",
127.      "category": None,
128.      "sub_category": "Perimeter & Walls",
129.      "extra_input": False,
130.      "quantity_formula": "1.05 * input_store['Area of Perimeter - Wall Height (LF)']",
131.      "is_manual_base_price": False,
132.      "base_price_reference": "Membrane Type Material 1",
133.      "discount_price": 0,
134.      "discount_price_input": False,
135.      "total_formula": "(quantity * base_price) + discount_price"
136.      },
137.
138.     {"description": "Membrane Type 1 Subcategory",
139.      "category": None,
140.      "sub_category": "Curb Perimeter",
141.      "extra_input": False,
142.      "quantity_formula": "1.05 * input_store['Area of Curb Perimeter']",
143.      "is_manual_base_price": False,
144.      "base_price_reference": "Membrane Type Material 1",
145.      "discount_price": 0,
146.      "discount_price_input": False,
147.      "total_formula": "(quantity * base_price) + discount_price"
148.      },
149.
150.     {"description": "Protection Mat",
151.      "category": "Flashings",
152.      "material_ids": [100],
153.      "extra_input": False,
154.      "is_manual_quantity": True,
155.      "base_price_reference": None,
156.      "use_discount_price_formula": True,
157.      "total_formula": "(quantity * base_price) + discount_price"
158.     },
159. ]
160.
161. material_pricing_list = []
162.
163. for data in materials_data:
164.     if data["category"]:  # Only fetch materials if a category is defined
165.         print(f"Fetching materials for category '{data['category']}'...")
166.
167.         # Fetch materials from SQL database
168.         available_materials = fetch_materials_by_category(db_file, data['category'])
169.
170.         if not available_materials:
171.             print(f"No materials found for category '{data['category']}'.")
172.             continue
173.
174.         # Filter materials based on IDs
175.         if "material_ids" in data:
176.             print(f"Filtering materials with IDs {data['material_ids']}...")
```

```
177.             available_materials = [item for item in available_materials if item[0] in
data["material_ids"]]
178.
179.         if not available_materials:
180.             print(f"No materials found with IDs {data.get('material_ids')} in category
'{data['category']}'.")
181.             continue
182.
183.         # Automatically select if there's only one material, otherwise prompt the user
184.         if len(available_materials) == 1:
185.             selected_material = available_materials[0]
186.             print(f"Automatically selected material: {selected_material[1]}")
187.         else:
188.             # Display drop-down options
189.             print("Available Materials:")
190.             for idx, item in enumerate(available_materials):
191.                 print(f"{idx + 1}. {item[1]}")
192.             while True:
193.                 try:
194.                     choice = int(input("Enter the number corresponding to the material: "))
- 1
195.                     if 0 <= choice < len(available_materials):
196.                         selected_material = available_materials[choice]
197.                         break
198.                     else:
199.                         print("Invalid selection. Please choose a valid number.")
200.                 except ValueError:
201.                     print("Invalid input. Please enter a number.")
202.
203.         material = MaterialPricing(
204.             description=selected_material[1],  # Material name
205.             sub_description=data.get("sub_category", ""),  # Sub-description (from
materials_data), default no sub if no key
206.             base_price=float(selected_material[3].replace('$', '')) if
isinstance(selected_material[3], str) else float(selected_material[3]),  # Base price with $
stripped
207.             unit=selected_material[4]  # Unit
208.         )
209.
210.     else:
211.         if not material_pricing_list:
212.             print(f"Skipping {data['description']} because no previous material exists.")
213.             continue  # Skip the material
214.
215.         # For materials that do not fetch from SQL
216.         print(f"Using previously selected material for {data['description']}
{data['sub_category']}.")
217.         material = MaterialPricing(
218.             description=data["description"],  # Reuse description
219.             sub_description=data["sub_category"],  # New sub-description
220.             base_price=material_pricing_list[-1].base_price,  # Copy base price from the
last material
221.             unit=material_pricing_list[-1].unit  # Reuse the unit
222.         )
223.
224.     # Prompt for manual base price if enabled
225.     if data.get("is_manual_base_price", False):
226.         try:
227.             material.base_price = float(input(f"Enter base price for
{material.description}: "))
228.         except ValueError:
229.             print("Invalid input. Using default base price.")
230.
231.     # Prompt for discount price if enabled
232.     if data.get("discount_price_input", False):
233.         try:
234.             discount_price_input = input(f"Do you want to input a discount price for
{material.description}? (y/n): ").strip().lower()
235.             if discount_price_input == "y":
```

```
                material.discount_price = float(input(f"Enter discount price for {material.description}: "))
            elif discount_price_input == "n":
                print(f"No discount price entered. Using default value: {material.discount_price}")
            else:
                print("Invalid input. Setting discount price to default value (0).")
        except ValueError:
            print("Invalid input. Setting discount price to default value (0).")

    if data.get("is_manual_quantity", False):
        try:
            material.quantity = float(input(f"Enter Quantity for {material.description} ({material.sub_description}): "))
        except ValueError:
            print("Invalid input. Setting quantity to 0.")
            material.quantity = 0

    # Evaluate quantity formula
    if "quantity_formula" in data:
        print(f"Evaluating quantity formula for {material.description}: {data['quantity_formula']}")
        material.evaluate_and_set(
            data["quantity_formula"],
            input_store,
            {"base_price": material.base_price},
            "quantity"
        )
        print(f"Calculated quantity for {material.description}: {material.quantity}")

    # Calculate discount price if indicated
    if data.get("use_discount_price_formula", False):
        print(f"Calculating discount price for {material.description}")
        material.calculate_discount_price(
            distributor_discount=input_store["Distributor Discount"],
            direct_discount=input_store["Direct Discount"]
        )
        print(f"Calculated discount price for {material.description}: {material.discount_price}")

    # Evaluate total formula
    if "total_formula" in data:
        print(f"Evaluating total formula for {material.description}: {data['total_formula']}")
        material.evaluate_and_set(
            data["total_formula"],
            input_store,
            {"quantity": material.quantity, "base_price": material.base_price, "discount_price": material.discount_price},
            "total"
        )
        print(f"Calculated total for {material.description}: {material.total}")

    #Save material data into input_store for json
    input_store[f"{material.description} - {material.sub_description}"] = {
        "Quantity": material.quantity,
        "Base Price": material.base_price,
        "Discount Price": material.discount_price,
        "Total": material.total
    }

    # Add to list
    material_pricing_list.append(material)

#save updated input_store to json file
save_to_json(project_file, input_store)
print("Material pricing data saved successfully!")

#Display selected materials
print("\nSelected Materials:")
```

```
299. print(f"{'=' * 20} Material Pricing {'=' * 20}")
300. for item in material_pricing_list:
301.     print(f"{item.description} ({item.sub_description}) | Quantity: {item.quantity} | Base
Price: ${item.base_price} | Discount Price: ${item.discount_price} | Total: ${item.total}")
302. print("=" * 40)
303.
304. final_total = sum(material.total for material in material_pricing_list)
305. print(f"{'=' * 20} Final Total {'=' * 20}")
306. print(f"Final Total: ${round(final_total, 2)}")
307. print(f"Mtls Per Sq: ", final_total / (input_store['Total Roof Area - Membrane'] / 100))
308. print("=" * 40)
309.
```

Figure 17 main file code

materials_query.py

This module handles database interactions. It:

- Fetches materials and their attributes (e.g., price, unit) from the SQL database based on categories.
- Retrieves available unit options for user selection.
- Ensures robust error handling for database queries.

```
 1. import sqlite3
 2.
 3. def fetch_materials_by_category(db_file, category):
 4.     """
 5.     Fetch materials from the 'material_list' table based on a specific category.
 6.
 7.     Args:
 8.         db_file (str): Path to the SQLite database file.
 9.         category (str): The category to filter materials by.
10.
11.     Returns:
12.         list: A list of tuples containing material data, or an empty list on error.
13.     """
14.     materials = []  # Initialize as empty in case of failure
15.     conn = None     # Placeholder for the connection object
16.
17.
18.     try:
19.         # Connect to the database
20.         conn = sqlite3.connect(db_file)
21.         cursor = conn.cursor()
22.
23.         # Query the table with a WHERE clause
24.         query = "SELECT id, name, category, price, unit FROM material_list WHERE category =
?"
25.         cursor.execute(query, (category,))
26.
27.         # Fetch the results
28.         materials = cursor.fetchall()
29.
30.     except sqlite3.Error as e:
31.         print(f"Database error: {e}")
32.     finally:
33.         # Ensure the connection is closed
34.         if conn:
35.             conn.close()
36.
37.     return materials
38.
39. def fetch_units(db_file):
```

```
    # Fetch unit names from the material_list table under the Units catgory
    units = []
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        cursor = conn.cursor()

        # Query to fetch names where category is 'Units'
        query = "SELECT Name FROM material_list WHERE Category = 'Units'"
        cursor.execute(query)

        #Fest and format the results
        units = [row[0] for row in cursor.fetchall()]

    except sqlite3.Error as e:
        print(f"Database error while fetchingnits: {e}")
    finally:
        if conn:
            conn.close()

    return units

```

Figure 18 Code querying SQL database

project_io.py

This utility module manages project data storage and retrieval. It:

- Saves project data into a JSON file to allow persistence across sessions.
- Loads data from a JSON file if an existing project is reopened.
- Provides user-friendly feedback on save/load operations.

```
import json
import os

def save_to_json(filename, data):
    # Save input data to a JSON file
    with open(filename, "w") as file:
        json.dump(data, file, indent=4)
    print("Project saved successfully!")

def load_from_json(filename):
    # Load input data from a JSON file if it exists
    if os.path.exists(filename):
        with open(filename, "r") as file:
            print("Project loaded successfully!")
            return json.load(file)
    else:
        print(f"File '{filename}' does not exist. Starting a new project.")
        return {}

```

Figure 19 Code to Write JSON file

material_pricing.py

This module focuses on material pricing calculations. It:

- Manages material attributes like base price, quantity, and discounts.
- Provides methods to calculate discounted prices and total costs.

- Includes logic for evaluating formulas and validating user inputs for price-related operations.

```python
import sqlite3

def fetch_materials_by_category(db_file, category):
    """
    Fetch materials from the 'material_list' table based on a specific category.

    Args:
        db_file (str): Path to the SQLite database file.
        category (str): The category to filter materials by.

    Returns:
        list: A list of tuples containing material data, or an empty list on error.
    """
    materials = []  # Initialize as empty in case of failure
    conn = None     # Placeholder for the connection object


    try:
        # Connect to the database
        conn = sqlite3.connect(db_file)
        cursor = conn.cursor()

        # Query the table with a WHERE clause
        query = "SELECT id, name, category, price, unit FROM material_list WHERE category = ?"
        cursor.execute(query, (category,))

        # Fetch the results
        materials = cursor.fetchall()

    except sqlite3.Error as e:
        print(f"Database error: {e}")
    finally:
        # Ensure the connection is closed
        if conn:
            conn.close()

    return materials

def fetch_units(db_file):
    # Fetch unit names from the material_list table under the Units catgory
    units = []
    conn = None
    try:
        conn = sqlite3.connect(db_file)
        cursor = conn.cursor()

        # Query to fetch names where category is 'Units'
        query = "SELECT Name FROM material_list WHERE Category = 'Units'"
        cursor.execute(query)

        #Fest and format the results
        units = [row[0] for row in cursor.fetchall()]

    except sqlite3.Error as e:
        print(f"Database error while fetchingnits: {e}")
    finally:
        if conn:
            conn.close()

    return units
```

Figure 20 Code to Handle Material Pricing

production_quantities.py

This module calculates production-related quantities. It:

- Manages attributes for each quantity (e.g., description, formula, unit).
- Allows for manual user input or automatic calculation based on predefined formulas.
- Stores and updates the calculated values in a shared input data structure.

```
 1. class ProductionQuantity:
 2.     def __init__(self, description, sub_description, is_manual=False, extra_input="",
value=0, unit="", formula=None, is_editable=False, unit_dropdown=False):
 3.         self.description = description  # Main description of the quantity
 4.         self.sub_description = sub_description
 5.         self.is_manual = is_manual  # Determines if this requires user input
 6.         self.extra_input = extra_input  # Optional extra input
 7.         self.value = value  # Calculated or user-input value
 8.         self.unit = unit  # Measurement unit
 9.         self.formula = formula  # Row-specific formula that is optional
10.         self.is_editable = is_editable
11.         self.unit_dropdown = unit_dropdown
12.
13.     def set_value(self, input_store, previous_value=None, available_units=None):
14.
15.         # Skip if already in input_store
16.         if f"{self.description}{self.sub_description}" in input_store:
17.             self.value = input_store[f"{self.description}{self.sub_description}"]
18.             print(f"Skipping calculation for {self.description} {self.sub_description},
using stored value: {self.value}")
19.             return
20.
21.         # Handle unit dropdown if applicable
22.         if self.unit_dropdown and available_units:
23.             print(f"Available units: {', '.join(available_units)}, press Enter for no
unit.")
24.             selected_unit = input(f"Select a unit for
{self.description}{self.sub_description}: ")
25.             if selected_unit in available_units:
26.                 self.unit = selected_unit
27.                 input_store[f"{self.description}{self.sub_description} - Unit"] = self.unit
28.             else:
29.                 print("Invalid unit selection or no unit assigned.")
30.
31.         # Editable descriptions
32.         if self.is_editable:
33.             new_description = input(f"Enter description (or press Enter to keep
'{self.description}'): ").strip()
34.             if new_description:
35.                 self.description = new_description
36.
37.         # Save extra_input value if applicable
38.         if self.extra_input:
39.             try:
40.                 extra = float(input(f"Enter extra input for
{self.description}{self.sub_description} (default is {input_store.get(self.extra_input, 0)}):
"))
41.             except ValueError:
42.                 extra = input_store.get(self.extra_input, 0)  # Default to stored value if
invalid input
43.             input_store[self.extra_input] = extra
44.
45.         # Evaluate and store formula if applicable
46.         if self.formula:
```

```
            try:
                self.value = eval(self.formula, {}, {"input_store": input_store,
"previous_value": previous_value})
                input_store[f"{self.description}{self.sub_description}"] = self.value
                print(f"Set value for {self.description}{self.sub_description}:
{self.value}")
            except Exception as e:
                print(f"Error evaluating formula for
{self.description}{self.sub_description}: {e}")
                self.value = 0

        # Store manual input if no formula exists
        elif self.is_manual:
            self.value = float(input(f"Enter value for
{self.description}{self.sub_description}: "))
            input_store[f"{self.description}{self.sub_description}"] = self.value

```

Figure 21 Code to Handle Production Quantities

Method 17 - A/B Testing

Method 1: Light Mode vs. Dark Mode

Description:

An A/B test was conducted to identify the most user-friendly and visually appealing color scheme for the wireframe interface. The options included a light mode design and a dark mode design, both incorporating TPC Roofing's turquoise-green branding. The goal was to enhance usability and align the visual interface with user preferences.

Justification:

Given the critical role of aesthetics and readability in user experience, selecting a color scheme that met user expectations was essential for the wireframe development process. A visually appealing design not only enhances satisfaction but also contributes to efficient data interaction in workflows.

Execution:

Question: Do you prefer a lighter color scheme or a darker color scheme?

Audience: 23 users were surveyed, including roofing estimators, IT specialists, data analysts, and office workers who frequently interact with screen-based interfaces.

Options:

Light Mode: Featured a clean white background with turquoise-green branding accents.

Dark Mode: Used a dark background with turquoise-green branding.

Participants were shown both designs and asked to select their preferred option based on usability and aesthetics.

Results:

Winning Option: Light Mode

Votes: *Light Mode received 16 votes, while Dark Mode received 7 votes.*

Feedback: *Participants favored Light Mode for its modern and professional appearance, while Dark Mode was noted for being easier on the eyes in low-light environments.*

Results

- **Winning Option:** Light mode
- **Votes:** Light mode received 16 votes, while dark mode received 7 votes.
- The findings indicated a clear preference for the light mode design due to its perceived modern and professional appearance.

Conclusion:

Light Mode was chosen for the final design due to its broader appeal and alignment with user feedback. Its clean and professional aesthetic better suits TPC Roofing's branding and target user base.

The data chart can be seen here in Appendix 9.

Method 2: Highlight Style Preferences (Red Highlights vs. White Highlights)

Description:

An A/B test was conducted to evaluate the best method for highlighting user input fields in the wireframe interface. Two options were tested: **red highlights** and **white highlights**. The aim was to determine which approach enhanced usability without causing visual fatigue.

Justification:

Highlighting user input fields effectively is essential for guiding users during data entry while maintaining a clean interface. Testing these options ensured that the final design balanced functionality and aesthetic coherence.

Execution:

- **Question:** Do you prefer red highlights or white highlights for marking user input fields?
- **Audience:** The same group of 23 participants as the previous test, consisting of roofing estimators, IT specialists, data analysts, and other professionals.
- **Options:**
    1. **Red Highlights:** Input fields were marked in bold red for visibility.
    2. **White Highlights:** Input fields were subtly shaded white to maintain a minimalistic look.
- Participants interacted with both designs and provided their preferences based on ease of use and visual appeal.

Results:

- **Winning Option:** White Highlights
- **Feedback:** While Red Highlights were noted for their boldness and ability to draw attention, many participants found them too harsh. White Highlights were favored for their subtlety and alignment with a professional aesthetic.

Conclusion:

White Highlights were chosen for the final design due to their balanced approach, providing clear guidance for user input without being visually overwhelming. This choice ensures that the interface remains user-friendly and cohesive.

The data chart can be seen here in Appendix 9.

Phase 4 – Evaluation

Test Questions

Excel Usage

What do you find easiest about working in Excel for this task?

What do you find most frustrating or time-consuming about using Excel for this task?

Are there features in Excel you find unnecessary or redundant?

If you could change one thing about the Excel sheet, what would it be?

Are there any tasks or calculations in Excel you wish were automated or streamlined?

How could Excel better support your workflow?

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

Are all formulas working as expected when tested with various scenarios?

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

Do you understand what data is required at each step?

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

Did you encounter any moments of confusion or hesitation?

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

Do error messages guide you toward fixing the problem?

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

Are there features you expected to find but didn't?

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

Are there steps in the excel process that you think should be simplified for the interface?

Are there steps in the terminal process that you think should be simplified for the interface?

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

Were there any steps where you were unsure what to input or what the program was asking for?

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

Can you easily identify the most recently modified project?

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

Are the drop-down menus for "Roof Type" and other options easy to use?

Are there any fields, sections, or design choices that feel unnecessary or redundant?

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

Is the split between "Material Pricing" and "Insulation Pricing" clear?

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

Are there any missing pieces of information that you'd expect to see in the final proposal?

General Feedback

Was there any point where you felt lost or unsure what to do next?

Are there any features you'd like to see added to improve the user experience?

Are the labels and descriptions clear and easy to understand?

Does the interface feel too cluttered or overly simplistic in any areas?

How does this compare to using the Excel document?

Method 18: Peer Review

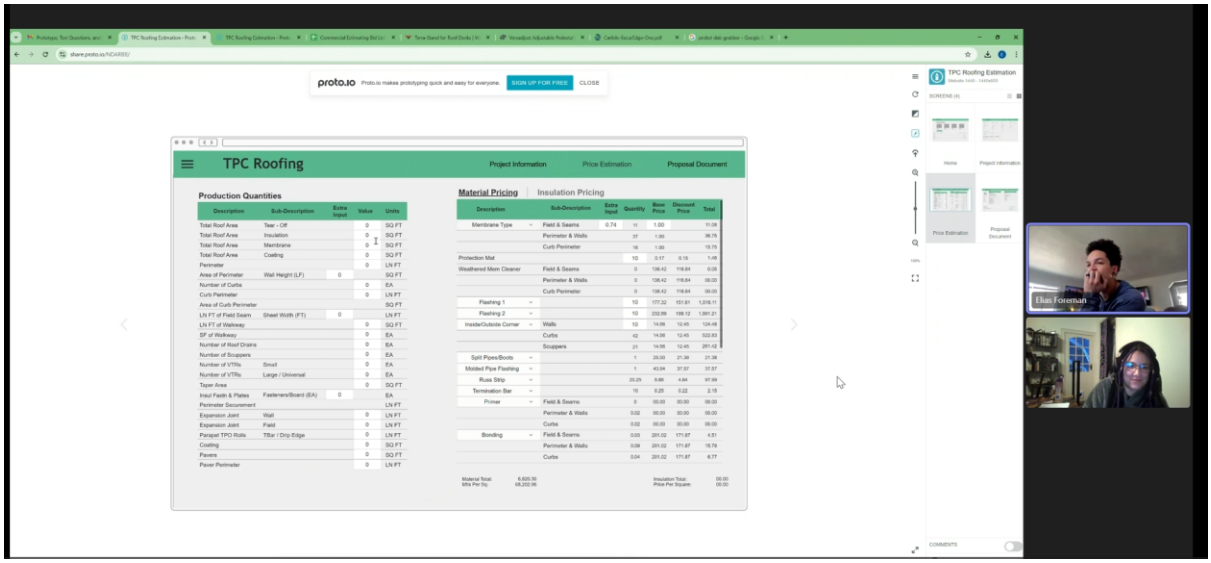


Figure 48 Peer Review Screenshot

Unfortunately, for unknown reasons, my voice was not picked up during the recording of some videos including this one, so the transcript will not be included.

Please see Appendix 11 Tester 3 and Tester 5 for Test Answers.

Description:

The peer review was conducted to gather feedback from professionals who regularly interact with Excel or data-driven workflows, as well as those familiar with the roofing industry. The goal was to identify strengths, weaknesses, and areas for improvement in the prototype and evaluate its alignment with user needs. Peer reviewers were selected for their expertise in data handling, workflow optimization, or roofing estimation processes.

Justification:

The feedback provided through peer review is critical for identifying potential blind spots in the prototype's design and ensuring that it meets the needs of real-world users. Additionally, insights from peers help refine usability, streamline processes, and align the tool with industry expectations. This method provided valuable insights into workflow alignment, usability, and accuracy of the prototype, ensuring it met real-world business needs for TPC Roofing and other potential industry users.

Execution:

- Participants: Roofing estimators, Game Design Student
- Materials: Participants were provided with access to the terminal-based code prototype and the wireframe along with a list of specific tasks to complete, such as entering data, running calculations (both to simulate the workflow of an estimator), and identifying missing or

confusing features. Each participant was asked to complete **three key testing tasks** to evaluate different aspects of the prototype:

1. **Excel Comparison Task:**
    - Participants first **completed an estimate using their standard Excel workflow**.
    - This provided a **baseline reference** for comparing manual effort and accuracy against the prototype.
2. **Terminal-Based Prototype Task:**
    - Participants entered the **same estimation data** into the **terminal-based prototype** developed for this project.
    - They were asked to **observe calculation accuracy**, ease of input, and any usability challenges.
    - If discrepancies appeared between Excel and the prototype, **participants noted calculation errors or missing features**.
3. **Wireframe Usability Testing:**
    - Participants **navigated the visual interface wireframe** and provided **design feedback**.
    - This step assessed the **potential intuitiveness of a future GUI**, particularly in comparison to Excel and the current terminal-based approach.
    - Participants **identified areas for improvement**, such as navigation flow, clarity of inputs, and overall user experience.

- Feedback was collected through written responses and verbal discussions during review sessions.

Results:

- Positive feedback was given for the code version of the Excel tasks.
- Users appreciated the accuracy of calculations but highlighted areas where the program needed clearer instructions or error messages.
- Roofing estimators noted that the tool could save time for larger projects but felt the terminal interface was less intuitive than Excel for quick edits.
- Others suggested minor changes to improve navigation and streamline data entry.

Conclusion:

The peer review provided valuable insights that guided refinements in the prototype. Feedback emphasized the importance of simplifying the user interface for quick tasks and improving instructions to aid new users. These recommendations informed subsequent iterations of the prototype and set a strong foundation for transitioning to a GUI in the future.

Method 19: Usability Testing

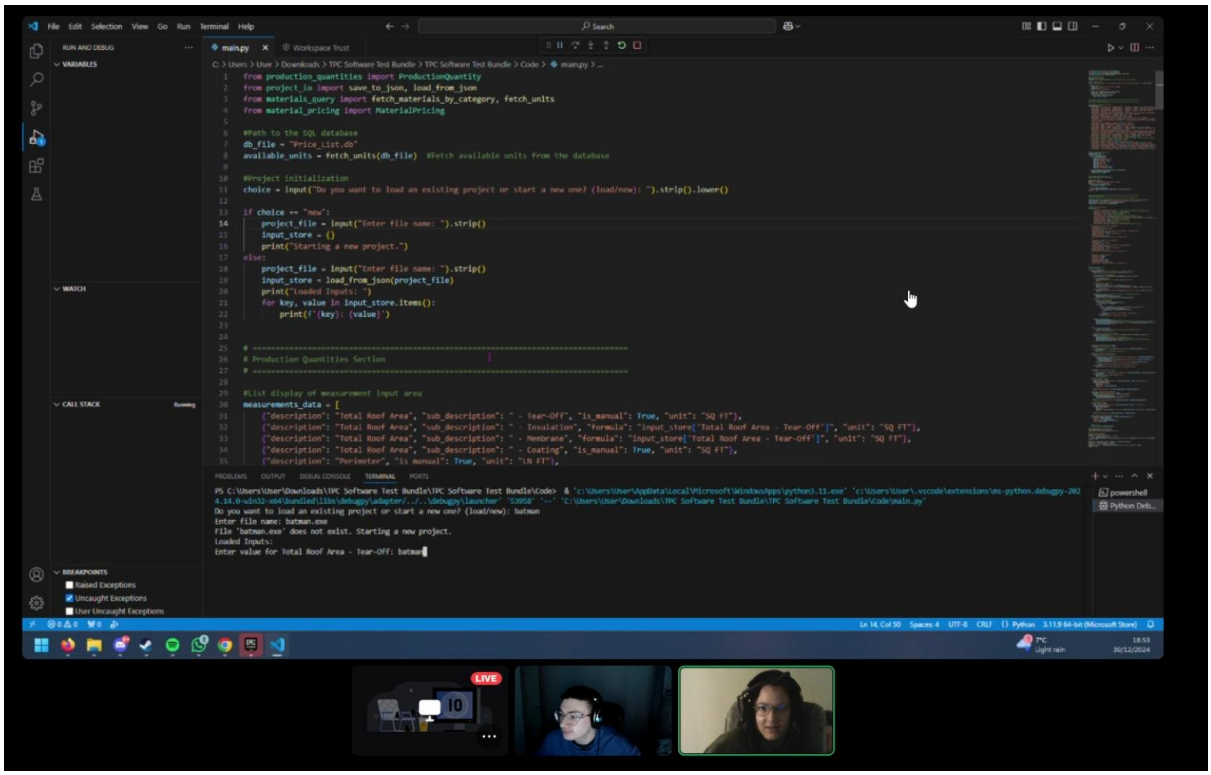


Figure 49 Usability Testing Screenshot

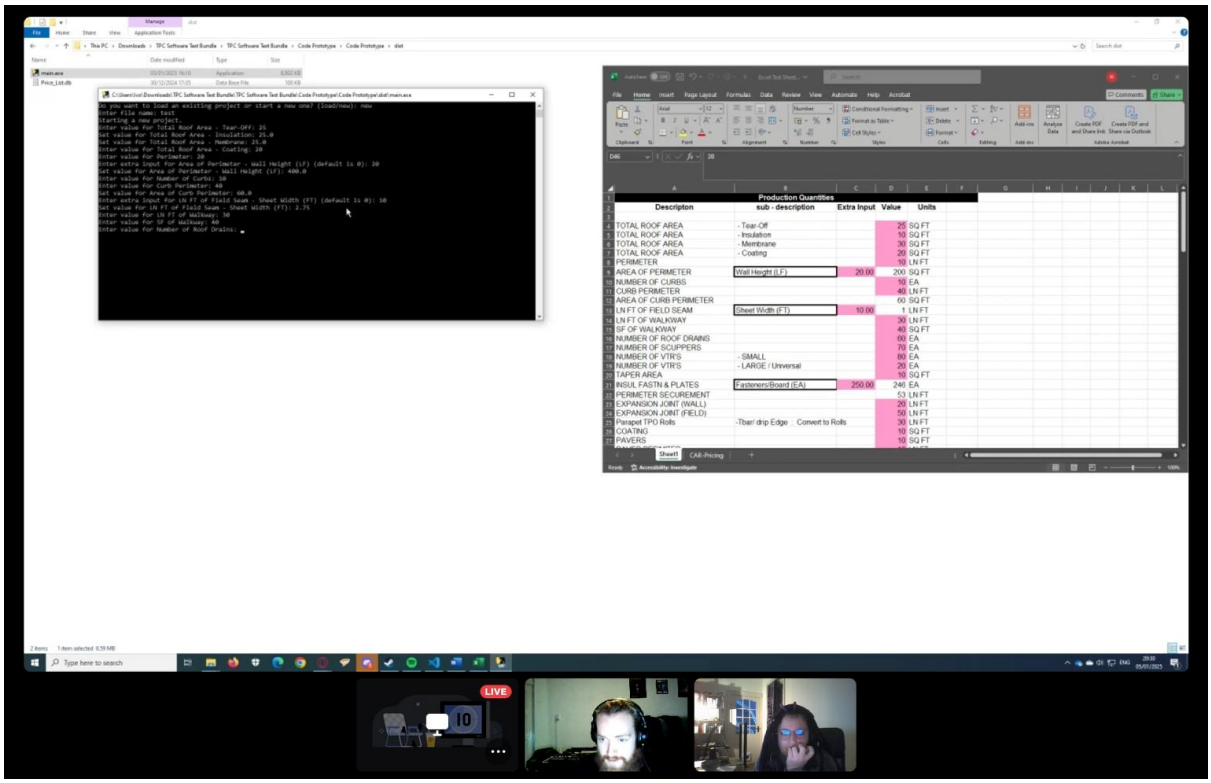


Figure 50 Usability Testing Screenshot 2

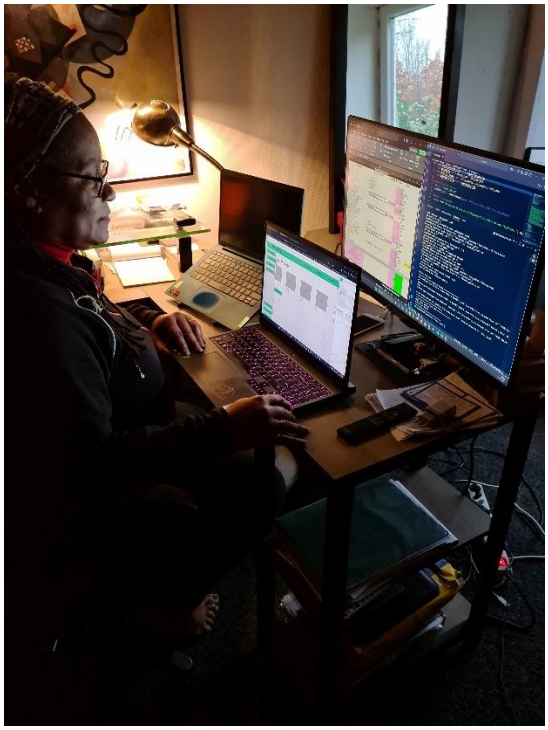


Figure 51 Usability Testing Screenshot 3

Description:

Usability testing was conducted to evaluate how well the prototype aligned with user expectations, supported workflows, and handled real-world scenarios. Participants were tasked with completing specific actions within the prototype, such as entering data, running calculations, and interpreting outputs. The testing aimed to uncover usability issues, identify areas for improvement, and gauge user satisfaction. See Appendix 11for test answers.

Justification:

Usability testing ensures that the prototype is not only functional but also user-friendly. By observing how participants interact with the tool, developers can pinpoint pain points, streamline workflows, and improve the overall user experience. This method was particularly important for validating the transition from Excel-based workflows to a terminal-based interface.

Execution:

- **Participants:** The participants were users with varying levels of experience in Excel and similar tools. Stakeholders, administrative assistants, and other users that may or may not frequently use excel and similar tools.
- **Test Environment:** Participants worked on the terminal-based prototype on their own devices, simulating real-world conditions.
- **Tasks:** The test, like before, was split into three sections:

1. Fill in data in the original Excel sheet. This was for users to gain an understanding of the flow and what would be asked of them in the terminal, and also to give them a base to go off of to compare to the terminal inputs and outputs.

2. Test the terminal program again. Here, the Excel sheet would also serve as a reference for users to better understand what the program was asking and what they had filled in, as well as for reference to compare. This task also served to record most error handling of the code and what happened during various typing mistakes.
3. Then the users would also test the GUI for design choices and navigation testing for new/unfamiliar users where I would also notes moments of confusion, hesitation, or frustration as I did in the terminal testing.

- **Observations:** I observed participants as they worked, noting moments of confusion, hesitation, or frustration.

- **Feedback Collection:** After completing the tasks, participants were asked to answer questions about their experience, focusing on the intuitiveness of the workflow, error handling, and feature coverage and design:

1. Was the workflow intuitive and logical?

2. Did the error messages guide you toward fixing problems?

3. Were there any features or steps that felt unnecessary or redundant?

Results:

- **Positive Outcomes:** The input and automation of material pricing calculations were found to be a significant improvement over manual Excel inputs since they were asked directly for the input. Error messages were generally helpful, though some users wanted clearer guidance for resolving input error and a few instances were discovered where crashes were caused because I had missed adding error messages.

- **Challenges:** The new, non-computer experienced users found the interface intimidating, especially when they were dealing with the large amounts of data output at the end that summarized everything. A few participants were confused by the terminology used in prompts, which required specific roofing knowledge.

- **Suggestions:** Testers recommended simplifying the workflow for creating new materials, as did the last test sessions, and suggested providing more detailed instructions for new users.

Conclusion:

This testing method highlighted both the strengths and weaknesses of the prototype. The feedback emphasized the importance of making this project understandable for new users like trainees or other companies with a similar design, while also maintaining its functionality for experienced estimators. These insights guided the plans for combining the wireframe into a singular program.

Method 20 – Think Aloud

Figure 52 Usability Testing Screenshot 4

Description

The Think Aloud Method was conducted with a participant who regularly uses Excel in their professional work but does not have direct experience with roofing estimation. The participant verbalized their thoughts while completing tasks in the terminal-based prototype, offering insights into usability, functionality, and clarity. See Appendix 11 for test answers.

Justification

The Think Aloud Method provides qualitative data about user interaction, highlighting areas of confusion, ease, or inefficiency. By observing the participant's real-time responses, it was possible to identify strengths and weaknesses in the tool that might not have been evident through structured testing alone.

Execution

- **Participant:** A plant biologist with extensive experience in Excel for data and analysis tasks.
- **Tasks:** The participant was asked to perform the same key tasks as previously mentioned, going through each of the 3 steps.
    - Entering data into the Production Quantities table.
    - Running calculations for material pricing and insulation pricing.
    - Reviewing outputs for accuracy.
- **Facilitation:** The participant was asked to speak their thoughts and reactions out loud throughout the session.

Results

- **Positive Feedback:** They found the terminal prompts clear and easily understood what data was required at each step. They enjoyed the efficiency of the terminal program compared to using Excel.
- **Challenges:** The user preferred Excel due to the familiarity and flexibility. But he recognized the potential of the prototype for streamlining workflows. The user, as others, pointed out the inability to navigate back and for between the questions, which felt restrictive.
- **Insights for Improvement:**
    - Adding units (e.g., %, $) directly in input fields to clarify expected values.
    - A warning or highlight system for fields requiring attention at a later stage.

Conclusion

The Think Aloud session provided valuable insights into how non-roofing professionals interact with the tool, offering a fresh perspective on usability. While the participant highlighted the efficiency and clarity of the terminal tool compared to Excel, their feedback emphasized the need for enhanced navigation and contextual guidance to further improve the user experience. These insights will be integrated into future iterations, particularly for GUI development.

Method 21: System Performance Validation

Description

System performance validation was conducted by reviewing the codebase to assess its efficiency, error handling, scalability, and alignment with user workflows. The evaluation focused on database interactions, material pricing calculations, production quantities, error handling, and overall system design. See Appendix 11 for test answers.

Justification

Given time constraints, a code review provided a reliable method to analyze the system's robustness and identify potential performance bottlenecks. This approach ensured that critical aspects like database queries, calculations, and scalability were thoroughly examined without requiring a full live test.

Execution

Code Review: The code was reviewed bit by bit to evaluate functionality and design. Key focus areas included database queries, correct calculations, proper formatting or copying of item information from previous sections and inputs, error handling, and scalability for future additions or larger datasets.

Testing Scenarios: Each testing scenario took place during, and at the end of coding sections, as well as sometimes returning to portions after I have added more. This was to ensure everything was working properly even after I had made changes, or to see that nothing accidentally affected past work.

1. **Production Quantities:**
    - The use of formulas and manual input handling was examined for dynamic adaptability to workflows.
2. **Error Handling:**
    - Exception handling mechanisms were reviewed to identify gaps and areas for improvement.
3. **Scalability:**
    - The modular code structure was evaluated for its potential to handle larger datasets or additional features.

Results

- **Database Interaction:** The queries were structured efficiently, and modular functions such as those handling data fetches supported all retrievals. However, there was a lack of detailed error handling for empty or invalid queries that I had missed, which would cause crashes.
- **Production Quantities:** From the beginning I created a dynamic structure and formulas for future adaptability to different workflows.
- **Material Pricing:** I made the methods for calculating discounts and totals modular as well as I had learned more from the Production section. Dependence on external variable like the user inputs (saved as input_store) required careful management and specific calls to fix the recurring issues of undefined errors.
- **Error Handling:** Exception and error displays were present but lacked user-friendly messages, which sometimes confused users during testing.
- **Scalability:** My modular design method supports future additions and was planned for integration with a GUI. SQL-based data management made sure that the system could handle the hundreds of materials and their data efficiently.

Conclusion

The system demonstrates strong potential in terms of performance and scalability due to its modular design and structured approach to calculations and database management. However, areas such as error handling and input validation require improvements to enhance robustness and user experience. These insights will guide further refinements and inform the development of the graphical user interface.

Insights and Adjustments Based on Evaluation

While the testing validated my main goals of ensuring that the prototype was properly replicating the workflow and displaying correct data, I got a lot of valuable insight during the evaluation phase that directly influenced the refinement of the prototype. These included the following key adjustments:

1. **Improved Error Handling:** Based on feedback from usability testing and system validation, the error handling was improved, and could still be further improved as I continue to work, for better error messages, including adding validation checks for user inputs.
2. **Streamlined Workflow:** The workflow would be simplified in the combination between the wireframe frontend and terminal backend, addressing the challenges identified during usability testing and peer reviews. Some current improvements and future improvements include minimizing the amount of required input buttons and improving the clarity of requested inputs.
3. **Better Navigation:** Feedback from the think-aloud and usability testing showed a need for better navigational methods. Some of these were already addressed in the wireframe, but clearer navigational buttons would be indicated in the future.
4. **User-Friendly Interface:** The wireframe design was improved on after a few rounds of A/B testing results as I worked on the design. There was a clear preference for a light mode interface with a subtle visual design for inputs and separation.

For more information see Method 17.

Future Iterations

This phase not only pointed out immediate refinements to be made, but also laid the foundation for future iterations. Key areas for future development include:

1. **User Interface:** Transitioning to a combine prototype with the backend and frontend GUI together in one will address the limitations of the terminal-based interface. This will provide a more intuitive and user-friendly experience. The wireframe will be the foundation and reference for the interface.
2. **Enhanced automation:** The next iteration will include the relevant sections of the second excel sheet being added to the code, adding the automation that this project aimed to achieve and cutting out extra work for the estimators. Another future iteration will also include automatic proposal generation based on collected data. These features will build on the current prototype's capabilities, reducing manual effort and improving efficiency.
3. **Scalability and Multi-User Support:** The modular design will ensure that the product can scale to support additional users and features and be adjusted to the needs of other industries. Future iterations will include multi-user support.
4. **Improved Error Handling and Input Validation:** Based on the feedback of the evaluation phase, future iterations should include more better error handling and input validation to make sure that the system can handle a wider range of user inputs.

Appendix

Appendix 1: Client Interview 1

Purpose of contact:

Discussing ideas for project direction and understanding skillsets. The client explained his workflow and the type of data he works with.

Additional information:

The client's workflow

- At his current company he gets the blueprints of clients and uses a PDF reader to record measurements by tracing the links. New company won't have the reader.
- He then puts those measurements into and excel spreadsheet to get the materials he needs: type and amount.
- In another excel sheet he gets the prices of the materials from suppliers.
- The pricing and updating is tedious.

Ideas:

1. Make a PDF reader that will scan and record measurements from the blueprints automatically (possibly also making it 3D?)
2. Automate the material and pricing process into a website or app. Additionally or alternatively automating the price updates.

Appendix 2: Client interview 2

Purpose of contact:

Understanding workflow

Additional information:

Email sent from The client providing additional information on the work process and including example of PDFs and spreadsheets.

Steps:

1. Receive blueprints and measure dimensions

- Roof plan/blueprints included w/ key
    - Which lines are relevant that need to be measured?
    - What is in/excluded in the area and/or roof measurements?
    - How are measurements best organised?

2. Translate measurements into quantities of materials to install roof. Membrane/insulation/glue/screws/metal flashings/etc.

- Done with excel sheet that needs help/overhaul
- Different tabs for different types of roof etc. - TPO/wall panels/metal roof/sloped
    - Can this be turned into equations?
    - How can this be organised better: More efficient? Clearer?
    - What if any can be automated?
    - Does The client want to keep the tabs as they are or change the organisations style?

3. Feed material quantities into price excel sheet.

- Multiple tabs: TPO/metal/tile/shingle
    - Can be turned into automated calculator. Need to find a way to connect the material calculator into this to combine two steps into one.

4. Prices fluctuate and need to be updated. This is tedious.
    - Can make this automated to read compare and update the pricing. Lists material name/prices/measurement unit
    - Probably can be combined into step 3.

5. Type a quote/proposal describing roof system and whats included in the price.

- Specific system is usually already determined by architect and described on drawings.

- Lists date, quote expiration, customer, estimator(?)
- Template: project name, quote #, address, then the materials/what needs to be done in one column and amount in the column to the right.
- Can probably be combined into step 3/4.

Workflow notes

1. Get a roof drawing
2. Revu - pdf editor allows him to measure sqft area of the roof
    a. Collect area information
    b. Green is coping
    c. Yellow walkpath
    d. Blue extra information
    e. Red perimeter
3. Go to excel sheet
    a. Insert project information
    b. Areas in pink are what are measured/areas to input data
4. Go to price sheet when all info is gathered and input materials
    a. Calculate total materials/material costs/labour costs
5. Input information into proposal document

Appendix 3: Competitor Estimation Tools Research

Abbreviation:

CRM - Customer Relationship Management

Leap

https://leaptodigital.com/leap-team/

Roofing estimating software that offers a range of features to streamline sales process and project management.

Pros:

- Customisable templates for proposals, tailored to each project
- End-to-end management tool that covers customer relationship management, task automation, project management, and etc.
- Payment system for creating invoices, tracking financials, and processing payments.
- Various tools: measurements, manufacturers, imagining technology, financing, +
- Wide range of affordable pricing packages

Cons:

- Large learning curve
- Sales capabilities may not be as strong as management features

Acculynx

https://acculynx.com/

Cloud-based roofing estimating software designed to simplify project management, estimation, and communication.

- Intuitive interface
- Mobile app
- Track projects
- Estimations
- Communication with team members and customers
- Offers integration with leading accounting and CRM software
- Singular focus is roofing industry

Pros:

- Cloud-based, accessible from anywhere
- Integration capabilities with various industry tools and software solutions

- Seamless communication with customers through automated emails and text messages

Cons:

- Higher pricing, no clear pricing info on website
- Integration capabilities come with additional costs
- More limited customization options vs competitors
- Features are better suited for desktop than mobile or tablet

JobNimbus

https://www.jobnimbus.com/

Versatile software solution that combines project management, CRM, and estimation tools in one platform

- Allows contractors to create detailed estimates
- Track projects progress
- Manage customer relationships efficiently
- Customisable workflows
- Lead organisation
- Sales optimisation, etc
- Considered one of the best CRM solutions in the industry

Pros:

- All-in-one platform for proj management, CRM, and estimation
- Customisable workflows
- Competitive pricing plans

Cons:

- Limited integration with other software solutions and competitors
- Some features may require additional training
- Less intuitive interface and more difficult navigation

RoofSnap

https://roofsnap.com/

Specialised roof estimating software designed to simplify the measurement and estimation process.

- Unique features allow users to quickly utilize aerial measurements of roofs using satellite imagery
- Eliminates need for manual measurements
- Claims to provide measurements in 30 minutes or less
- Allows you to create material orders and sign contracts with customers

Pros:

- Aerial measurement feature: quick and accurate
- Integration with CRM and proj management tools
- User-friendly interface

Cons:

- Limited customisation options for estimates
- Little to no management capabilities in customer relationships and production

Roofr

https://roofr.com/

User-friendly software that focuses on simplifying estimations and sales process. It has a robust range of features:

- Measurements
- Material ordering
- Invoicing
- Payments
- Create prof estimates and proposals in minutes
- Integration with various tools to enhance collaboration and communication

Pros:

- User friendly interface
- Manage leads and customer relationships in the app
- Customer pricing and instant estimator feature

Cons:

- Limited advanced features compared to competitors
- Good with sales but falls behind in project management
- Lack of customisation features that doesn't work as well with more complex jobs

Jobber

https://www.google.com/search?q=jobber

Comprehensive business management software for service-based industries, incl roofing contractors

- Scheduling
- Invoicing
- Quoting tools
- Makes it easy for contractors to manage projects
- Track expenses
- And communicate with customers effectively
- Known for great scheduling and getting paid but not as strong as roof estimating software

Pros:

- All-in-one platform for business operations
- More advanced customer service features than most competitors
- Great for managing schedules and dispatching your team

Cons:

- Has one of the most extensive learning curves, features can be under utilised
- Less customisation options for estimates
- Not as strong in project management beyond scheduling

iRoofing

https://iroofing.org/

Provides contractors with tools to measure, estimate and present roofing projects. Mobile app allows users to take measurements and create estimates on-site. Features are unique but limited.

- Ai-based roof-visualizer
- Great for aerial measurements and roofing logistics
- Not as strong in proj management and sales

Pros:

- Mobile app for on-site measurements and estimates

- Ai-based roof visualizer and aerial measurements
- Strong roofing logistics and instant estimate capabilities

Cons:

- Falls off in features after estimating
- Less extensive project management and customer relationship tools
- Does provide integrations, but less expansive than competitors. May not integrate as well with all systems

New roofing Technology Trends

https://www.sumoquote.com/articles/new-roofing-technology-trends

Artificial Intelligence:

Roofers can use AI to manage the daily needs of their business.

AI and what it means for roofers.

Recent advancements in roofing technology highlight several trends that can inform the development of an estimation app for TPC Roofing.

1. **Artificial Intelligence (AI):**
   AI is increasingly used by roofers to streamline daily operations. For example, AI-driven tools can optimize scheduling, manage resources, and analyze data to improve business efficiency.
2. **Smart Roofs:**
   Smart roofs incorporate sensors to detect weather changes and can perform automated tasks, like heating to melt snow. This concept of automation and environmental adaptability could inspire features in the app.
3. **Roofing Software:**
   All-in-one roofing software solutions, including CRM integrations, are becoming more popular. These tools complement business/customer relationship apps by offering features like automated pricing, templates, and streamlined workflows.
4. **Roofing Estimation Apps:**
   Apps like SumoQuote are gaining traction, offering features such as eSignatures, standardized pricing, and customizable templates. These tools provide inspiration for features tailored to TPC Roofing's needs.
5. **Supply Chain Improvements:**
   Supply chain diversification is expected to decrease, potentially causing disruptions. An app that automates supplier pricing updates and provides flexibility for sudden changes could be highly beneficial.

Roofing Estimation Software - Leap To Digital

https://leaptodigital.com/blog/roof-estimating-software/

Roofing estimation software should have the following **key features:**

- Accurate measurement tools
- Material database: pricing info
- Customisable templates: for estimates/proposals/invoices (will have TPC-specific template)
- Integration: seamless integration with other tools and systems (excel?)
- Cost calculation features
- Mobile accessibility: measure, calculate, and project manage on the go
- Reporting and analytics: provides insights into project performance, profitability, and productivity
- Training and support: offers thorough training, onboarding, resources, tutorials, and responsive customer support.

Choosing the right roof estimating software, things to consider:

- Ease of use and user interface
- Customisation options
- Integration capabilities
- Pricing and scalability
- Customer support and training

Appendix 4: Mid-Term Review Notes

Design Challenge

- Only one design challenge, I have too many
- Formulate it as a research question
- It's the main idea for the project
- DC gives structure to what I have to design

I have too much work planned out for myself.

- Focus down to automation?
  - This is too narrow, add data input or something
- Narrow Scope

Iterate on Key Terms and Research Questions

Need more expert testing in the end.

- UI testing with non-roofers/estimators

Appendix 5: SCAMPER Notes

The SCAMPER method was used to refine and optimize the final concept rather than generate entirely new ideas. By applying this technique, I was able to improve feasibility, usability, and scalability within the project constraints. Below are notes on how each SCAMPER element influenced the design:

Substitute:

- Instead of focusing on a fully automated proposal generator, the project shifted to automating pricing data updates.
- This change was made because manual pricing updates were identified as a critical user pain point in client interviews.
- Proposal generation would have addressed only the final step of the workflow, whereas pricing automation provides a more fundamental improvement.

Combine:

- Features from multiple concepts were merged:
    - Real-time collaboration from the Project Management App.
    - Automation elements from the Estimation Dashboard.
- The result was a hybrid solution that maintained automation benefits while aligning with TPC's existing workflow.

Adapt:

- The idea of an all-in-one app was adapted to work with existing Excel workflows.
- This ensured that estimators wouldn't have to learn an entirely new system—instead, the prototype integrates with tools they already use.
- Adaptation made implementation more feasible given the project's technical constraints.

Modify:

- The user interface design was adjusted to prioritize simplicity and usability.
- Rather than introducing a new interface from scratch, the design was aligned with TPC's existing workflow to make adoption easier.
- Backend changes ensured that automation worked without disrupting user habits.

Put to Other Uses:

- The automated pricing update feature was initially designed for internal use, but it could also be extended to external supplier integration in future iterations.
- This expands potential applications beyond just estimation to broader supply chain management.

Eliminate:

- Features that were too ambitious for the project's timeframe were removed:

    - Mobile access.
    - Advanced CRM tools.
    - Additional security features.
- Removing these kept the project focused on core estimation improvements while allowing future scalability.

Rearrange:

- The workflow was restructured to prioritize Production Quantities and Material Pricing as the primary focus areas.
- Instead of tackling the entire estimation process at once, the system was built to first automate pricing and calculations, then expand later.
- This made the project more manageable within the timeframe while still delivering value.

Appendix 6:User Interface Design Research

UI/UX Design Estimation Techniques

Source:
Equal Design Blog
LinkedIn UI/UX Design Advice

- **Design Methods:**
    - **Hourly-based:** Charges calculated based on the time spent designing the interface.
    - **Project-based:** Costs determined based on the scope and complexity of the project.
- **Key Steps in Estimation:**
    - Determine the **number of actions/screens** required (create a sitemap or flowchart).
    - Factor in complexity (e.g., dynamic features vs. static elements).
    - Include time for iterations, client feedback, and testing phases.

Best Practices for UI/UX Design

Source:
NNGroup – 10 Usability Heuristics

1. **Visibility of System Status:**
    - Ensure users receive feedback on what the system is doing (e.g., progress bars, notifications).
2. **Consistency and Standards:**
    - Use familiar design patterns to reduce the learning curve.

3. **User Control and Freedom:**
    - Provide easy ways to undo or redo actions.
4. **Error Prevention:**
    - Design systems to minimize errors, such as validation for inputs or warnings for irreversible actions.
5. **Flexibility and Efficiency of Use:**
    - Offer shortcuts for experienced users while keeping basic options intuitive for beginners.

UI/UX for Desktop Applications

Source:
Smashing Magazine - Desktop UI/UX Design

- **Principles for Desktop Applications:**
    1. Prioritize **key actions** based on frequency of use (e.g., place frequently used buttons prominently).
    2. Use **visual hierarchy** to guide users' attention (e.g., size, contrast, positioning).
    3. Ensure **responsiveness**: Interface should not lag when users interact.
    4. Maintain **clarity**: Avoid clutter; group related elements logically.
- **Design Techniques:**
    - Focus on **large screens** with adaptable layouts for different resolutions.
    - Use **consistent navigation** patterns across all screens.
    - Implement **keyboard shortcuts** to enhance usability for advanced users.

Designing for Roofing Estimation Apps

Source:
Riseapps - Best UX Practices for SaaS Applications

- **Targeted Features:**
    - Clear and **intuitive forms** for data input (e.g., measurement entries).
    - Use of **tooltips** and contextual help to guide users during complex processes.
    - **Drag-and-drop** file functionality (e.g., for uploading price sheets).
- **Dashboard Design:**
    - Centralized view of ongoing projects and estimates.
    - Visual representation of **data analytics** (e.g., graphs for project cost breakdowns).

- **Automation in UI:**
    - Real-time updates when data is input or changed (e.g., instant recalculation of totals).

Additional Notes

- **Importance of Prototyping:**
    - Wireframes and prototypes allow for early usability testing and stakeholder feedback.
- **Iterative Design:**
    - Emphasize continuous improvement through user feedback and testing cycles.
- **Accessibility Standards:**
    - Follow WCAG (Web Content Accessibility Guidelines) for inclusive design.

1. *Gates, M. (1967).* "Bidding Strategies and Probabilities." *Journal of the Construction Division, ASCE, 93(1), 75-107.*
    - A seminal study on bidding models and probability analysis.
2. Drew, D. S., & Skitmore, R. M. (1993). "Competitive Tendering: Principles and Practices." Construction Management and Economics, 11(1), 105-112.
    - Discusses the competitive nature of bidding and how construction firms strategize.
3. Fu, H., Drew, D. S., Lo, H. P., & Li, H. (2002). "The Application of a Competitive Bidding Model in Construction." Automation in Construction, 11(5), 547-559.
    - Examines the role of automation in improving bidding accuracy.
4. umnov.denis. (2023, July 26). Bidding Automation: Opening Constructions Hidden Potentials. ConWize. https://conwize.io/articles/the-untold-secrets-of-bidding-automation/
    - This article discusses how bidding automation software empowers firms to analyze project requirements, evaluate subcontractor bids, and determine competitive pricing with unparalleled speed and accuracy.
5. SmartBuild Systems, 2022. "Roofing Software for Automated Bidding."
    - This source highlights the benefits of using fully automated roofing software platforms for estimating and ordering, enabling contractors to take command of the bidding process and improve efficiency.
6. Ciampaglia, S. (n.d.). Revolutionizing Construction Bids: How Automated Bidding Saves Time and Maximizes Profits. ContractComplete. https://www.contractcomplete.com/revolutionizing-construction-bids-how-automated-bidding-saves-time-and-maximizes-profits/

- This article explores how automated bidding streamlines the construction bidding process, reduces errors, and maximizes profits for firms.

7. BEAM AI. (n.d.). Roofing estimating software powered by AI Takeoffs - BEAM AI. https://www.ibeam.ai/estimate/roofing-estimating-software

- This source discusses the role of AI-powered roofing estimating software in automating takeoffs, saving time, and improving bid accuracy.

8. Construction Bid Management Software: A complete guide for contractors (2025). (n.d.). https://www.projectmark.com/blog/construction-bid-management-software

- This article examines how Customer Relationship Management (CRM) systems can enhance competitive analysis in bidding, helping firms assess competitors and strengthen their proposals.

Appendix 7: Expert Interview

Interview Questions

What do you think are some of the main challenges roofing contractors face when creating estimates?

How do you think automation can help solve some of these problems?

Are there any specific best practices you think I should keep in mind when designing an estimation tool?

Is there anything I should avoid when transitioning from manual workflows to something automated?

If you were creating a tool for a small team like I'm doing for you guys, where would you start?

Do you know of any tools or technologies I could look at for inspiration?

Interview Answers

What do you think are some of the main challenges roofing contractors face when creating estimates?

- I think one of the biggest headaches is maybe how manual the whole process still is. Measurements, material quantities, labor costs, yknow. It all gets dumped into spreadsheets, and there's a lot of room for error. And also updating material prices. It's tedious.

How do you think automation can help solve some of these problems?

- It would be super helpful. no more manual tracing or measuring with a tool that integrates with a PDF reader to pull measurements straight from a blueprint. Then there's pricing. If you link your tool to a central database, you can automate price updates, so you're not spending hours chasing supplier emails or updating sheets. And automating proposal creation saves time and ensures everything looks consistent.

Are there any specific best practices you think I should keep in mind when designing an estimation tool?

- Keep it simple. Estimators don't want to deal with something overly complicated—it needs to feel familiar, like their spreadsheets but smarter. Also Growth. Even if you're building for a small team now, you've gotta think ahead, what happens when they grow? Also making sure everything checks out. Little errors in calculations or pricing can snowball into big problems and huge missed pricing.

Is there anything I should avoid when transitioning from manual workflows to something automated?

- I 've heard of a lot of people try to do too much all at once. They throw in every feature under the sun, like how we mentioned Star Citizen. But this would also affect the users I guess. It's confusing when you have too much information all at once.

If you were creating a tool for a small team like I'm doing for you guys, where would you start?

- I think what Elias told you was a good start. Starting with automating material pricing and production quantities. There should be customizability, though.

Do you know of any tools or technologies I could look at for inspiration?

- Since we use our own models and workflows in Excel sheets, I'm not entirely sure. But I've heard SumoQuote does documents for you.

Appendix 8: Code Language Comparison Notes

Python

- **Pros:**
    - Simple syntax; beginner-friendly.
    - Lots of libraries for data handling.
    - Lost of learning resources.
    - Easy prototyping and iteration.
- **Cons:**
    - Slower than compiled languages.
    - Limited for heavy computational tasks without optimization.
    - I don't know Python

JavaScript (with Node.js for backend)

- **Pros:**
    - Good for handling multiple tasks.
    - Strong ecosystem for web-based tools (e.g., Electron for desktop apps).
    - Universal language; can handle frontend and backend.
- **Cons:**
    - Harder for beginners.
    - Less suited for computational-heavy workflows.
    - I barely know javascript and never used node

Java

- **Pros:**
    - Strong performance; ideal for scalable, multi-threaded apps.
    - many libraries and tools for enterprise software.
    - Reliable for long-term maintainability.
- **Cons:**
    - steep learning curve for small projects.
    - Slower development cycle compared to Python.
    - Never used Java

C#

- **Pros:**

    - Good integration with Windows environments.
    - Strong for GUIs.
    - High performance.
    - Have most experience of all languages
- **Cons:**
    - Requires more setup and learning time.
    - Less community support for lightweight prototypes.

Appendix 9: Rapid Prototyping Notes

Notes over the course of the various rapid prototyping stages/versions.

- Tables keep merging into one cell.
- Material names split into 3 cells
- Not all of material name shows up
- Pricing cant find right column
- just use Excel for now – taking too long to figure out PDF issues
- File-drop works, but the UI doesn't confirm if it was successful - solved
- Database Errors – something with SQL table structure. – solved for updates
- Add code to check for duplicates. Inserting one line workds - solved
- Tkinter drag-and-drop label not showing right - solved
- File path with spaces is breaking the program – solved

- Clean up print statements
- Add manual input for discount price - solved
- Extra input column in Production Quantities. forgot to check if it updates totals.
- Dropdown options work – fix number of materials pulled from database later

- Total price calculation keeps throwing error. – solved. Empty qualities messed it up
- Database file doesn't show up when building exe bundle

- People don't notice drop downs in excel sheet, make clearer in wireframe
- Added color to differentiate user inputs—red was ugly. white neutral.
- Put in defaults
- navigation buttons sometimes overlap text on smaller screens
- hamburger menu glitches out when changing screens after clicking

- Highlight error cells with a soft red background when interface is made (like when direct discount is bigger than distributor discount)

Appendix 10: A/B Testing

A/B Test 1: Light Mode vs. Dark Mode

Role	Option Chosen	Comments (if provided)
Data Entry Clerk	Light Mode	"Easier to read and looks modern."
Roofing Estimator	Light Mode	"Feels more professional."
Data Scientist	Dark Mode	"Easier on the eyes in dim lighting."
Roofing Estimator	Light Mode	"Matches the company branding better."
IT Specialist	Dark Mode	"I like darker UIs for focus."
Cashier	Light Mode	
Administrative Assistant	Light Mode	"Looks cleaner."
Data Entry Clerk	Light Mode	"More readable."
Roofing Estimator	Dark Mode	
Office Worker	Light Mode	"Feels less overwhelming."
Graphic Designer	Light Mode	"Easier for long-term analysis."
Cyber Security	Light Mode	"Easier to find information."
IData Analyst	Dark Mode	"Less strain in low-light conditions."
Mobile Phone Sales	Light Mode	"Matches professional standards."
Administrative Assistant	Light Mode	
Roofing Estimator	Dark Mode	"Darker screens are less tiring."
Data Scientist	Light Mode	"Bright but not too harsh."
Plant Biologist	Light Mode	"Fits modern design trends."
IT Specialist	Dark Mode	"Preferred for night work."
Administrative Assistant	Light Mode	"Colors stand out better."
Data Entry Clerk	Light Mode	
Roofing Estimator	Light Mode	
IT Specialist	Light Mode	

A/B: Red Highlights vs. White Highlights

Role	Option Chosen	Comments (if provided)
Data Entry Clerk	White Highlights	"Less distracting and cleaner for data entry."
Roofing Estimator	Red Highlights	"Red catches the eye for important fields."
Data Scientist	White Highlights	"Subtle but effective for long hours."
Roofing Estimator	White Highlights	"Doesn't overwhelm the interface."
IT Specialist	Red Highlights	"Helps focus on input areas quickly."
Cashier	White Highlights	
Administrative Assistant	White Highlights	"More visually balanced."
Data Entry Clerk	White Highlights	"Better contrast for repetitive tasks."
Roofing Estimator	Red Highlights	
Office Worker	White Highlights	"Red feels harsh."
Graphic Designer	Red Highlights	"Looks bold and modern."
Cyber Security	White Highlights	"Keeps focus on the content."
Data Analyst	White Highlights	"Easier for continuous use."
Mobile Phone Sales	Red Highlights	"Grabs attention where needed."
Administrative Assistant	White Highlights	
Roofing Estimator	Red Highlights	"Makes input sections obvious."
Data Scientist	White Highlights	"More subtle for less strain."
Plant Biologist	White Highlights	"Matches the overall clean design."
IT Specialist	Red Highlights	"Useful for quick data entry."
Administrative Assistant	White Highlights	"Fits with a professional layout."
Data Entry Clerk	White Highlights	
Roofing Estimator	Red Highlights	
IT Specialist	White Highlights	

Appendix 11: Testing Consent Forms

All consent forms for testing can be found in a separate documents labeled Appendix – Testing Consent Forms

Appendix 12: Rigorous Testing - Usability - Think Aloud - Peer Review Test Answers

Testing of all 3 parts: excel sheet, code prototype, wireframe prototype

Tester 1 – Useability Testing

Excel Usage

What do you find easiest about working in Excel for this task?

- Nothing

What do you find most frustrating or time-consuming about using Excel for this task?

- Having to go through every fillable cell and making sure all value exist

Are there features in Excel you find unnecessary or redundant?

- Unsure

If you could change one thing about the Excel sheet, what would it be?

- Nothing, it is as straight forward as an excel can be

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- Asking for the next measurement or equiptment

How could Excel better support your workflow?

- Excel is limited in its capabilities and should be moved to a developed application where possible

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yes

Are all formulas working as expected when tested with various scenarios?

- Yes

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- Yes

Do you understand what data is required at each step?

- Yes

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- Yes

Did you encounter any moments of confusion or hesitation?

- Nope, I'm sure it would appear even more straight forward to a roofer

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- yes it asked me to reenter a valid input

Do error messages guide you toward fixing the problem?

- yes

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- Yes

Are there features you expected to find but didn’t?

- Nope

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- Data should be editable

Are there steps in the excel process that you think should be simplified for the interface?

- Easier to flow through inputs

Are there steps in the terminal process that you think should be simplified for the interface?

- Data should be able to be read on screen without scrolling back through the terminal

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- Nope

Were there any steps where you were unsure what to input or what the program was asking for?

- Nope

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- I would choose the excel as a terminal would not be user friendly and also isn't as fleshed out as something having a UI would provide

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- I think it would save time over the course of learning the program

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- I think this form works better in excel over the terminal as Excel allows you to see all data inputted already, this is fixed with the UI.

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Really clear and snappy

Can you easily identify the most recently modified project?

- **Yes**

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Yes

Are the drop-down menus for "Roof Type" and other options easy to use?

- Yes

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- None

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- Yes and you can use the tab key to move to the next inputtable text box.

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- Yes

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Yes

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yes

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Yes

Are there any missing pieces of information that you'd expect to see in the final proposal?

- None

General Feedback

Was there any point where you felt lost or unsure what to do next?

- No

Are there any features you'd like to see added to improve the user experience?

- None

Are the labels and descriptions clear and easy to understand?

- Yes

Does the interface feel too cluttered or overly simplistic in any areas?

- Nope it's fit for purpose

How does this compare to using the Excel document?

- More user friendly, less clutter, less risk of accidentaly removing formulas and creating a wrong invoice. A lot more snappier than an excel would be.

Tester 2 – Usability Testing

Excel Usage

What do you find easiest about working in Excel for this task?

- Using tab and arrows to change cells

What do you find most frustrating or time-consuming about using Excel for this task?

- I can't keep track of with row I'm in, its super spaced out

Are there features in Excel you find unnecessary or redundant?

- I don't know if it counts but its annoying if you accidentally type into the wrong cell

If you could change one thing about the Excel sheet, what would it be?

- The layout

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- I don't know what can be automated tbh

How could Excel better support your workflow?

- It would be nice to just be able to hit like tab or something and go to the next input instead of to the next cell because I kept overwriting the units and the formulas.

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yeah as far as I could tell, except when I made a mistake and had to start over

Are all formulas working as expected when tested with various scenarios?

- Yes I think so.

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- Mostly, except for the part where it asked me if I wanted to change a bunch of stuff

Do you understand what data is required at each step?

- Yeah I could figure it out

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- yes

Did you encounter any moments of confusion or hesitation?

- Yeah the part I mentioned above, it was unclear with the changing descriptions.

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- Yeah they're pretty clear

Do error messages guide you toward fixing the problem?

- Well it tells me what I did wrong but not exactly fixing the problem

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- yeah

Are there features you expected to find but didn't?

- not really, there were more than I expected

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- It's hard to read the terminal text and follow whats happening without an interface

Are there steps in the excel process that you think should be simplified for the interface?

- The drop downs, I didn't realise those were drop down choices without being told

Are there steps in the terminal process that you think should be simplified for the interface?

- The editing options

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- The whole thing feels repetitive but that's the job I guess

Were there any steps where you were unsure what to input or what the program was asking for?

- The descriptions

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- I think the terminal tool. Excel it's a bit easier to see the overall, but its hard to read. I hate looking at cells, I'd rather just be asked the question and answer one at a time.

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- It saves visual effort I think, theres no going back and forth to check the row and following it.

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- I think when the interface is implemented itll be a huge benefit.

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Yes

Can you easily identify the most recently modified project?

- Yeah the most recent few.

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Yeah I think so, assuming you know what all the extra stuff at the bottom is about for this job.

Are the drop-down menus for "Roof Type" and other options easy to use?

- Pretty straightforward.

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- I don't think so.

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- Yes

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- It seems clean but maybe they could have like an outline or something.

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Maybe there could be some vertical lines but its still pretty clear I think.

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yeah but maybe it looks a bit outdated.

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Some of the things I didn't know what it meant but if it is meant for a specific job I assume they'd know. The layout is pretty clear though.

Are there any missing pieces of information that you'd expect to see in the final proposal?

- A total price, unless I missed it.

General Feedback

Was there any point where you felt lost or unsure what to do next?

- Not really.

Are there any features you'd like to see added to improve the user experience?

- None that I can think of.

Are the labels and descriptions clear and easy to understand?

- Yes

Does the interface feel too cluttered or overly simplistic in any areas?

- There’s a lot of information in the tables that feeks a bit cluttered but I don’t know what can be done about it since you need that information.

How does this compare to using the Excel document?

- Definitely better than staring at excel cells.

Tester 3 – Peer Review

Excel Usage

What do you find easiest about working in Excel for this task?

- You can see all of the input at once.

What do you find most frustrating or time-consuming about using Excel for this task?

- Have to click on each input.

Are there features in Excel you find unnecessary or redundant?

- Not really.

If you could change one thing about the Excel sheet, what would it be?

- Maybe having not a 0 in the input field. So you don't have to remove the zero before filling it in.

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- The first three inputs should be same automatically. (Tho you already know that)

How could Excel better support your workflow?

- No idea

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yes

Are all formulas working as expected when tested with various scenarios?

- Yes

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- Pretty much except for the renaming part

Do you understand what data is required at each step?

- Yes, tho I first have to read it a bit

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- I think the flow should be: description, value, unit

Did you encounter any moments of confusion or hesitation?

- During the part where you had to input a description, a value and a unit. The order was a bit weird

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- I did not encounter this

Do error messages guide you toward fixing the problem?

- I did not encounter this

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- Yes

Are there features you expected to find but didn't?

- No

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- The unit selection

Are there steps in the excel process that you think should be simplified for the interface?

- Not really

Are there steps in the terminal process that you think should be simplified for the interface?

- The unit selection and the renaming

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- Not really

Were there any steps where you were unsure what to input or what the program was asking for?

- No

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- Terminal

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- Yes, it requires less clicking to input new fields

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- Change the order of changing description, value and unit

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Very clear

Can you easily identify the most recently modified project?

- **Yes**

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Yes

Are the drop-down menus for "Roof Type" and other options easy to use?

- Very easy to use

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- Not really

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- Yes

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- No, the buttons weren’t noticeable

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Yes

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yes

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Yes

Are there any missing pieces of information that you'd expect to see in the final proposal?

- No

General Feedback

Was there any point where you felt lost or unsure what to do next?

- No

Are there any features you'd like to see added to improve the user experience?

- Nothing particularly

Are the labels and descriptions clear and easy to understand?

- Yes

Does the interface feel too cluttered or overly simplistic in any areas?

- It was very clear

How does this compare to using the Excel document?

- I feel the input fields were better

Tester 4 – Think Aloud

Excel Usage

What do you find easiest about working in Excel for this task?

- Color fields where I need to fill out numbers/quantities

What do you find most frustrating or time-consuming about using Excel for this task?

- Easy to make mistakes and not easy to read with all the fields looking the same

Are there features in Excel you find unnecessary or redundant?

- no

If you could change one thing about the Excel sheet, what would it be?

- Make it easier to read

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- I don't have any ideas

How could Excel better support your workflow?

- I don't have any ideas

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- There was a difference, but probably because I made a typo.

Are all formulas working as expected when tested with various scenarios?

- Besides the typo issue, the rest seemed to be good

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- Yes

Do you understand what data is required at each step?

- Yes

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- Yes

Did you encounter any moments of confusion or hesitation?

- A few times

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- This was not tested

Do error messages guide you toward fixing the problem?

- Not applicable

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- yes

Are there features you expected to find but didn't?

- no

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- No idea

Are there steps in the excel process that you think should be simplified for the interface?

- no

Are there steps in the terminal process that you think should be simplified for the interface?

- no

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- no

Were there any steps where you were unsure what to input or what the program was asking for?

- no

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- Excell, probably because I am used to this

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- No.

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- I could not go back to make corrections

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Yes

Can you easily identify the most recently modified project?

- Not tested

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- yes

Are the drop-down menus for "Roof Type" and other options easy to use?

- yes

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- no

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- yes

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- no

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- yes

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- yes

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- yes

Are there any missing pieces of information that you'd expect to see in the final proposal?

- no

General Feedback

Was there any point where you felt lost or unsure what to do next?

- Not really

Are there any features you'd like to see added to improve the user experience?

- In the input fields of the UI show the unit (like % or $ signs)
- Warning/highlight option for a field that needs to be checked or filled out at a later stage.

Are the labels and descriptions clear and easy to understand?

- yes

Does the interface feel too cluttered or overly simplistic in any areas?

- no

How does this compare to using the Excel document?

- Much easier and less overwhelming on the computer screen, and no more copying-pasting from multiple excel files. 100% better!

Tester 5 – Peer Review

Excel Usage

What do you find easiest about working in Excel for this task?

- The ability to see everything in one view and make quick edits.

What do you find most frustrating or time-consuming about using Excel for this task?

- Manually updating prices and making sure formulas are correct.

Are there features in Excel you find unnecessary or redundant?

- Some formatting tools feel excessive for this task.

If you could change one thing about the Excel sheet, what would it be?

- Automate updates from supplier sheets

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- Inputting repetitive data and updating old formulas.

How could Excel better support your workflow?

- Better integration with external price sheets.

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yes, but double-checking some edge cases is needed.

Are all formulas working as expected when tested with various scenarios?

- Mostly, but there were a few calculation hiccups.

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- For the most part, yes, but a few could be worded better.

Do you understand what data is required at each step?

- Sometimes, but additional examples would help.

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- Yes, though it could be streamlined further.

Did you encounter any moments of confusion or hesitation?

- A couple, especially when inputting values in certain sections.

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- Yes, but some messages could be more specific.

Do error messages guide you toward fixing the problem?

- Usually, yes.

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- Almost, but a few minor Excel features are missing.

Are there features you expected to find but didn't?

- The ability to lock certain fields would be nice.

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- The navigation and data entry process.

Are there steps in the excel process that you think should be simplified for the interface?

- Combining material selection and quantity input into one step

Are there steps in the terminal process that you think should be simplified for the interface?

- Reducing the number of inputs required for basic tasks.

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- Yes, entering similar data repeatedly feels tedious.

Were there any steps where you were unsure what to input or what the program was asking for?

- A few, especially with the discount price section.

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- The terminal tool for automation, but Excel feels more intuitive for manual adjustments.

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- Yes, for bulk calculations; no, for quick edits or one-off changes.

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- More guidance, fewer manual inputs, and quicker navigation options.

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Yes, but a tooltip for the button would help.

Can you easily identify the most recently modified project?

- Not easily; a highlight would help.

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Mostly, though some fields need clearer examples.

Are the drop-down menus for "Roof Type" and other options easy to use?

- Yes, but more descriptions for terms would be helpful.

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- No, everything seems relevant.

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- It's a bit overwhelming. Better instructions are needed.

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- Not really; more distinct headers would help.

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Mostly, but color coding could enhance clarity.

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yes, but it could use a logo or more branding.

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Yes, but it takes a moment to find exclusions

Are there any missing pieces of information that you'd expect to see in the final proposal?

- No, everything seems included.

General Feedback

Was there any point where you felt lost or unsure what to do next?

- Yes, during the estimation section.

Are there any features you'd like to see added to improve the user experience?

- A guided setup wizard.

Are the labels and descriptions clear and easy to understand?

- Some need more detail or examples.

Does the interface feel too cluttered or overly simplistic in any areas?

- Slightly cluttered in the estimation section.

How does this compare to using the Excel document?

- It’s more efficient but less intuitive for new users.

Tester 6 – Usability Testing

Excel Usage

What do you find easiest about working in Excel for this task?

- The highlighted input cells were easiest.

What do you find most frustrating or time-consuming about using Excel for this task?

- There were too many cells to read clearly and the input cells were too far from the description cells. Additionally, the drop down boxes were not clearly accessible.

Are there features in Excel you find unnecessary or redundant?

- The blank columns of cells made the spreadsheet difficult to read. Some columns had only two or three cells to input extra data leaving another 50 blank cells which was confusing to double check inputs.

If you could change one thing about the Excel sheet, what would it be?

- I would make alternating rows a different color to better keep track of description and data entry alignment.

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- It would be easier if the perimeter areas were automatically calculated. It would also be more convenient if the tear-off, insultaion, covering layers were pre-estimated.

How could Excel better support your workflow?

- If there were parameter checks that prompted you to double check an entry that seemed to high or low.

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yes.

Are all formulas working as expected when tested with various scenarios?

- Yes.

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- The terminal prompts were clear until the section for calculating specialty products. The the requests for units/value/product renaming became tedious.

Do you understand what data is required at each step?

- Essentially yes, however, the user must be familiar with specialty products, their units and descriptions.

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

-

Did you encounter any moments of confusion or hesitation?

- Yes, when entering the specialty product data as stated above.

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- Yes, the program restated the question if the input did not meet specifications.

Do error messages guide you toward fixing the problem?

- Re-stating the input request was sufficient.

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- Yes, the tool covered all the features and was easier to work with than the Excel sheet.

Are there features you expected to find but didn't?

-

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- An easy way to introduce and coordinate specialty products and pricing.

Are there steps in the excel process that you think should be simplified for the interface?

- The interface could prompt the user to select a product option from the pull down menu. This is not clear in the Excel process.

Are there steps in the terminal process that you think should be simplified for the interface?

- No, the interface and terminal process seem like they will fit together nicely.

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- It seemed repetitive to need to name units and be offered the opportunity to rename the product.

Were there any steps where you were unsure what to input or what the program was asking for?

- No, the terminal process was much cleaner than the Excel spreadsheet.

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- I would definitely choose the terminal tool. The terminal tool is faster, easier to understand and allows faster data input.

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- As stated above, the terminal tool is faster, easier to understand and allows faster data input.
-

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- I would like to see the terminal tool prompt for corrections when values are significantly outside certain parameters.

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Yes.

Can you easily identify the most recently modified project?

- I did not try this aspect.

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Yes, the project screen was extremely clear and easy to use.

Are the drop-down menus for "Roof Type" and other options easy to use?

- Yes, after using the terminal process and Excel spreadsheet the project screen was very clear and easy to use. The drop-down menus are very clear from the project screen in comparison to the Excel sheet and terminal process.

Are there any fields, sections, or design choices that feel unnecessary or redundant?

-

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- Yes, this is easy and clear.

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- I didn't realize that these were separate clickable fields. It would be better if the fields were more noticeable buttons or there was a prompt to move to the next sections.

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Yes, this is clear.

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yes, I would like to see this kind of easy to read and understand document in many different professions.

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Yes, this was easy to read.

Are there any missing pieces of information that you'd expect to see in the final proposal?

- An expiration date of the proposal would be helpful.

General Feedback

Was there any point where you felt lost or unsure what to do next?

- Yes, with data entry on the Excel sheet, it was difficult to know what extra information was needed.

Are there any features you'd like to see added to improve the user experience?

- No, once the UI is linked, this will be a premium product. For now, the terminal process is substantially easier to work with than the Excel sheet.

Are the labels and descriptions clear and easy to understand?

- Everything is straightforward, except the specialty products. Here someone just needs to know what the specialty products are to select them and add information.

Does the interface feel too cluttered or overly simplistic in any areas?

- The UI mockup is very clear and clean in appearance.

How does this compare to using the Excel document?

- The interface looks and feels very professional. The entry points are clear and the output is clear and concise.

Tester 7 – Client Peer Review

Excel Usage (old method template)

What do you find easiest about working in Excel for this task?

- Preset formulas do all the calculations.

What do you find most frustrating or time-consuming about using Excel for this task?

- Data needs to be re-entered into multiple spreadsheets.

Are there features in Excel you find unnecessary or redundant?

- No

If you could change one thing about the Excel sheet, what would it be?

- I'd have it copy data to my proposal form.

Are there any tasks or calculations in Excel you wish were automated or streamlined?

- See above.

How could Excel better support your workflow?

- See above.

Prototype Usage

Accuracy of Functionality:

Were the outputs from the terminal tool consistent with Excel's results?

- Yes

Are all formulas working as expected when tested with various scenarios?

- All except the crashes that happened after Enter was pressed.

Clarity of Instructions:

Are the terminal prompts clear and unambiguous?

- Yes

Do you understand what data is required at each step?

- Yes

Ease of Workflow:

Is the process of completing a task (e.g., selecting materials, entering inputs, and viewing results) logical and straightforward?

- Yes

Did you encounter any moments of confusion or hesitation?

- No

Error Handling:

If you entered incorrect data, does the program provide meaningful error messages?

- Did not encounter any error messages.

Do error messages guide you toward fixing the problem?

- See above

Feature Coverage:

Do you feel all necessary features of the Excel sheet are covered in this version of the tool?

- Yes

Are there features you expected to find but didn't?

- No

Preparation for Interface Design:

What parts of the terminal workflow do you think will need the most improvement in an interface?

- Not sure

Are there steps in the excel process that you think should be simplified for the interface?

Several cells can be omitted from the process (roof area coating, SF of walkway, small VTR's, COATING, PAVERS, PAVERS PERIM)

Are there steps in the terminal process that you think should be simplified for the interface?

- No

Relevance of Speed and Understanding

Are there parts of the process that feel unnecessarily repetitive or slow?

- No

Were there any steps where you were unsure what to input or what the program was asking for?

- Not after Silke's initial explanation.

Comparing Both Approaches

If you were to choose between using Excel and the terminal tool, which would you prefer, and why?

- The terminal tool flows faster but the Excel sheet is easier to read because all the data and information is displayed in tables.

Do you feel the terminal tool saves you time or effort compared to Excel? Why or why not?

- The terminal tool flows faster.

What improvements would you like to see in the terminal tool to make it easier to use or more efficient?

- See above list of items that can be removed from the flow.

UI Design

Home Screen (Recent Projects)

Is it clear how to start a new project?

- Yes

Can you easily identify the most recently modified project?

Yes

Project Information Screen

Is it straightforward to fill in project details (e.g., Project Name, Address, Start Date)?

- Yes

Are the drop-down menus for "Roof Type" and other options easy to use?

- Yes

Are there any fields, sections, or design choices that feel unnecessary or redundant?

- SRI sheet metal PM, SRI Roofing PM, Billing date, % of markup RCFOs table, retainage %, Hourly Labor Rates table.

Price Estimation Screen

Can you easily input or edit data in the "Production Quantities" table?

- Yes

Is the split between "Material Pricing" and "Insulation Pricing" clear?

- Yes

Does the layout make it easy to distinguish between data types (e.g., Quantity, Base Price, Discount Price, Total)?

- Material pricing “extra input” column is not necessary. A subtotal field would be helpful; Example:

Membrane type – Field/Seams = 11

Perimeter & walls = 37

Curb Perimeter = 16

TOTAL = 64 material quantity / $ dollar amount

Proposal Document Screen

Is the presentation of the proposal document clear and professional?

- Yes

Can you identify all the necessary elements (e.g., customer details, exclusions, payment terms)?

- Yes

Are there any missing pieces of information that you'd expect to see in the final proposal?

- An extra field for project specific notes would be helpful

General Feedback

Was there any point where you felt lost or unsure what to do next?

- No

Are there any features you'd like to see added to improve the user experience?

- Not beyond above mentioned notes.

Are the labels and descriptions clear and easy to understand?

- Yes

Does the interface feel too cluttered or overly simplistic in any areas?

- No

How does this compare to using the Excel document?

Better!

Appendix 13: Notes taking During testing

Code:

- If you hit enter without putting g in a number during the main input sections then you get an error – add a 0 number default for no input
- The client said there are certain materials that can have other default numbers, for future use
- The Units code is caps sensitive, also ad joined abbreviations (SQFT along with SQ FT)

Wireframe:

- Someone found it considerably better visually than excel

- The Material Pricing/Insulation Pricing buttons aren't clear that theyre buttons and they look more like a label that says "material pricing | Insulation pricing"

- Multiple people missed the insulation pricing section, only one noticed immediately it could be clicked

- During the project work pages, make it more clear that the sections in the top right are buttons

- Also make it more clear which tab you're currently in

- Adding $ and . divider for material value inputs and displays
- Drop down for roof type in the project information page says "File size" instead of "Metal"
- Someone said "description" on the tables with the section names – material pricing/insulation/production quantities

System Validation

Database Interaction:

- Fetching materials works
- No error handling for invalid or empty categories—could crash if input is off.

Material Pricing

- Pricing calculations handle discounts well, and methods like calculate_total are clean for debugging.

- No checks for weird values—breaks if inputs are off.

Production Quantities

- Handles formulas and manual inputs perfect.

Error Handling

- Error messages are clear mostly

5. **Scalability**

- Complex formulas could slow things down with huge data unless optimized (or so I was told, idk how much data).

6. **Usability**

- Code structure looks ready for UI integration later.
- Prompts could be clearer. some are confusing.