

Progress Report

- Increment 1 -

Group 14

1) Team Members

Samantha Bui

- sb21bn
- samanthabui

Ludginie Dorval

- lld22
- Ludginie

Antonio Garriga

- jaf21m
- a-garriga

William Lee

- wl23f
- wl23f

Lillian Malik

- lam22g
- lillianmalik

2) Project Title and Description

ManuFactor is a software solution designed to streamline accounting processes for manufacturing companies. It provides tools for Cost-Volume-Profit (CVP) analysis and budgeting for production. Admin users have the ability to list/add users, list/add data. Users have the ability to list/add data. Viewers have the ability to list data. The system integrates database management, role-based user access, and secure authentication. By providing a straightforward tool for handling essential accounting tasks, ManuFactor enhances efficiency and accessibility in manufacturing operations.

3) Accomplishments and overall project status during this increment

Essential features, such as automated Cost-Volume-Profit (CVP) analysis and production budgeting, have been successfully implemented. Additionally, the user role system (Admin, User, and Guest) has been set up, providing secure access control through encrypted files. Our next steps involve finalizing the integration of the front-end and back-end, as well as completing the outstanding HTML files.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

- The most significant challenge encountered was the merging of our code and the establishment of clear roles within the team. To address this, we held bi-monthly meetings to discuss our tasks and set clear objectives for the completion of each increment.
- The initial plan aimed to have the front-end and back-end merged by the end of this first increment. However, due to scheduling conflicts among team members, insufficient time was available to complete this task. Fortunately, the scope of our goals for increments 1 and 2 was relatively light,

which ensures that achieving the objectives for increment 1, alongside the tasks for increment 2, remains feasible.

- A technical issue related to the MySQL password hindered our ability to test the code as thoroughly as originally planned. This issue was resolved during our final meeting; however, resolving the problem consumed a significant portion of the meeting time, resulting in our delay in the merging of the front-end and back-end.
- The front-end logic was tested independently from the back-end logic, and the back-end logic was tested independently from the front-end logic. As of Increment I, we have not fully integrated the front-end logic with the back-end logic. However, as we progress, we will be sure to integrate efficiently, we will integrate efficiently to implement the calculations and budgeting tool. so that we can fully implement functions.

5) Team Member Contribution for this increment

*Please list each individual member and their contributions to **each of the deliverables in this increment** (be as detailed as possible). In other words, describe the contribution of each team member to:*

- a) *the **progress report**, including the sections they wrote or contributed to*
- b) *the **requirements and design document**, including the sections they wrote or contributed to*
- c) *the **implementation and testing document**, including the sections they wrote or contributed to*
- d) *the **source code** (be detailed about **which** parts of the system each team member contributed to and **how**)*
- e) *the **video or presentation***

Samantha Bui

- a) the **progress report**: Finalized description about Cost-Variable-Profit, Budgeting, RBAC, Encryption to specify admin, user, viewer functionality. Contributed to challenges by elaborating challenges of integration between front-end logic and back-end logic, specifically in testing. Contribution to next increment, prioritizing integration between back end and front end to implement calculation functions. Contributed to personal section.
- b) the **requirements and design document**: Contributed to RBAC functionality for user management. Contributed to RBAC functionality for data management.
- c) the **implementation and testing document**:
 - a. Documented the front-end logic of RBAC, explaining how user access was restricted based on role assignments due to a need for integration with the back-end logic.
 - b. Explained how RBAC functioned independently from the database in this increment, describing how role-based access was verified in the front-end before linking it to back-end data.
- d) the **source code**: Provided Python Flask app.py as well as the HTMLs for (RBAC) implementation and Encryption, implementing user role structure and securing sensitive data.
- e) the **video or presentation**: Discussed front-end intentions are for (RBAC) and Encryption as well as summarizing the challenges and next steps, particularly regarding the pending database integration and finalizing database functions for seamless interaction between front-end and back-end.

Ludginie Dorval

- a) the **progress report**: discussed my work connecting the front end with the database, explaining how I integrated the user interface to ensure smooth data retrieval.
- b) the **requirements and design document**: Provided assistance in section.
- c) the **implementation and testing document**: As part of our testing, I highlighted the challenges we encountered during this process, including properly handling encryption, verifying user existence before insertion, and addressing database connection issues.

- d) the **source code**: Initialized the MySQL database (manuDB), ensuring default users (admin, user, viewer) exist with encrypted usernames and passwords. This connects to the database, checks if each user is present, and adds them if missing.
- e) the **video or presentation**: Discussed the initialization of the database, explaining how the default users were set and ensured their credentials were securely encrypted. Also, highlighted the obstacles we faced during this process, such as handling encryption properly, verifying user existence before insertion, and resolving any database connection issues.

Antonio Garriga

- a) the **progress report**: Completed: Developed the product_cvp.py file to interact with the database and perform CVP analysis. Defined the structure of the product database table, ensuring it contains all required financial data. Initial testing of data retrieval and calculations.
In Progress: Refining the backend logic to ensure accuracy and efficiency. Validating the calculations for various financial scenarios. Improving database interactions to optimize performance.
Pending: Expanding features based on additional user requirements. Conducting full-scale testing with larger datasets. Preparing the final video or presentation.
Challenges & Solutions:
Challenge: Ensuring that all necessary data fields were included in the database for accurate calculations.
Solution: Thoroughly analyzed the CVP formula requirements and adjusted the database schema accordingly.
Challenge: Optimizing the script for performance when handling large datasets. Solution: Implemented indexing and efficient queries to improve retrieval speeds.
Next Steps:
Finalize the backend development. Conduct extensive testing to verify accuracy. Prepare documentation and presentation materials.

- b) the **requirements and design document**:

Project Background & Idea Development:

The idea for ManuFactor originated from a need to simplify financial decision-making in manufacturing. By automating CVP analysis, businesses can quickly assess profitability and make informed decisions about production levels and pricing strategies.

Database Design:

The core data is stored in the product table, which includes:

product_id (Primary Key), product_name, fixed_costs, variable_costs_per_unit, selling_price_per_unit, and target_profit (optional, for target sales calculation).

Interaction Between product_cvp.py and Database:

The script queries the product table to retrieve cost and pricing data. It calculates the break-even point, target units, and target sales (target units \times selling price per unit). The results are returned and can be displayed in a user interface or stored for further reporting.

- c) the **implementation and testing document**:

The script was written in Python using SQL queries to retrieve relevant financial data from the product table. It utilizes functions to compute break-even analysis and target sales projections. The backend logic ensures that calculations remain efficient and scalable.

Testing Process:

Initial testing with sample data verified that the script accurately computed financial values. Edge cases were considered, such as zero or negative costs, extremely high or low selling prices, and missing data fields.

Future Implementation Steps:

Continuing backend development to improve efficiency and scalability. Refining error handling and optimizing query performance. Expanding the script to accommodate additional financial metrics.

- d) the **source code**: The source code consists of `product_cvp.py`, which includes database connection logic, query execution to retrieve relevant product data, functions to compute break-even point and target sales metrics, and output formatting for easy interpretation. Future enhancements may include a web-based interface for easier user interaction and integration with other financial modules.
- e) the **video or presentation**: How the idea for ManuFactor originated. My role in handling backend development. Explanation of the product database table and its fields. How the calculations help businesses make informed financial decisions.

William Lee

- a) the **progress report**: Made a script `init_database.sql` to create a tables `user` and `product_cvp`. Storing information about the product and users where the 2 tables are associated with one another based on their foreign keys. Planning ahead, I intend to add another table `BP`
- b) the **requirements and design document**: The system is designed to store information about products and users while ensuring consistent data integrity. The use of primary and foreign keys enforces correct logic, maintaining accurate relationships between tables.
- c) the **implementation and testing document**: Designed to store information about the product and users. Data integrity was implemented by using primary and foreign key. Queries were tested through `product_cvp.py` to ensure the computation for the functions were successful.
- d) the **source code**: `init_database.sql` creates a database with 2 tables, “users” and “product_cvp”.
- e) the **video or presentation**: **Talked about my role in this project, explained the database and our plan moving forward.**

Lillian Malik

- a) the **progress report**: Primarily responsible for drafting the majority of the IT document, and provided assistance for section 5.
- b) the **requirements and design document**: Responsible for the development of the RD document, as well as the creation of the associated use-case, class, and sequence diagrams.
- c) the **implementation and testing document**: Primarily responsible for drafting the first and second sections of the IT document, and assisted with sections 3, 4, and 5.
- d) the **source code**: Due to Lillian's limited experience with SQL and databases, she did not contribute to the development of the source code. However, in future increments, Lillian will focus on contributing to the CSS and HTML components of the project.
- e) the **video or presentation**: Facilitated the recording of the presentation and provided an in-depth overview of the project.

6) Plans for the next increment

During the upcoming increment we will complete the remaining HTML files and implement the Budgeting Production feature, along with improvements for the user experience. We will also conduct thorough testing to ensure the stability of the system. On top of this, we plan to initiate discussions on the design of the User Interface through the CSS files and evaluate the inclusion of changelogs.

Along these lines, we will be prioritizing integration between the back-end logic, front-end logic, database, and HTMLs. We will do this by prioritizing connecting front-end to back-end through the database function to implement calculations. Possibly standardize MySQL authentication settings across all development environments.

7) Stakeholder Communication

Subject: Progress Update on ManuFactor – Increment 1

Dear Stakeholder,

I am writing to provide an update on the progress of ManuFactor, our web-based accounting platform.

Current Status:

We are nearing completion of Increment 1. Key features, including automated Cost-Volume-Profit (CVP) analysis and production budgeting, have been implemented. The user role system (Admin, User, and Guest) has been established, ensuring secure access control. Our remaining tasks include finalizing the front-end and back-end integration and completing the remaining HTML files. Testing will follow to ensure proper functionality.

Challenges:

The primary challenge encountered was the delayed integration of the front-end and back-end components, due to scheduling conflicts within the team. While this task was initially planned for completion by the end of Increment 1, it has been pushed to Increment 2. However, this adjustment does not impact the overall project timeline, as the scope of Increment 2 remains manageable.

Next Steps:

In the upcoming increment, we will finalize the front-end integration, complete the remaining HTML files, and add the Budgeting Production feature, as well as various user compatibility features. Comprehensive testing will also be conducted to ensure system stability.

Thank you for your continued support. Should you have any questions, please do not hesitate to reach out.

Best regards,
Group 14, ManuFactor

8) Link to video

https://youtu.be/r_vnD7plpN4