Anthony Colley

Joshua Swilling

Jose Vazquez

Kennedy Stokes-Sutton

This Implementation Plan Report emphasizes the strategy for deploying the "System Sales Manager," a customer solution designed to address the complex challenge of tracking inventory and sales in a user-friendly way, while also generating reports for the business owner. The primary goal of the system is to improve business operations, reduce concerns about sales management, and allow owners to focus on the core aspects of their business.

The purpose of this implementation report is to ensure a smooth deployment process while guaranteeing the full functionality of the program. While we understand that any implementation may involve some potential disruptions, this document will serve as a guide for all team members involved, including employees and business owners, to minimize interruptions and ensure a seamless transition.

## Implementation Methods pro and cons.

In the world of IT, there are several implementation methods, such as direct cutover, pilot implementation, parallel operation, and phased implementation.

Direct Cutover

The direct cutover, also known as the "big bang" method, involves stopping the use of the old system (or starting fresh if no system exists) and setting a specific date for the full implementation, without any prior employee testing. While this methodology is less expensive than others, which will be discussed later, and offer faster transition, there are significant risks. If a bug is not identified during testing and employees encounter it, it could lead to wasted time while we try to fix it. Additionally, there may be system interruptions, and users will need to learn the new system as they go. This is why we believe this implementation method is not an option for us, as we want to maintain the highest efficiency.

Parallel Operation

In parallel operation, both the old and new systems are used simultaneously for a limited period. This allows the new system to be evaluated to ensure it meets all requirements, but it also means performing duplicate tasks. While the risk is exceptionally low, as the old system serves as a backup, and users can learn the new system gradually, we believe that duplicating tasks could lead to inefficiency and wasted time. Furthermore, some users might feel uncomfortable using two systems at the same time.

Phased Rollout

In a phased rollout, the system is implemented in stages, with each module deployed one at a time. For example, the inventory module might be implemented first, and once it is working smoothly, the next phase will be deployed. This approach allows for more control over the process, and training can be conducted in smaller, manageable steps. However, the implementation time is longer, and there will be a period where both systems coexist, which could lead to inefficiency.

Pilot Implementation

For our project, we have decided to go with the pilot implementation method. This involves selecting a small group of employees, training them on the new system, and conducting tests to ensure everything is working as expected. Once we are confident in the system's performance, we will begin deployment to the rest of the organization. The advantage of this method is that we can evaluate the system in real-world conditions. While testing typically focuses on our perspective, users may interact with the system in ways we had not anticipated, which could uncover issues we missed. Additionally, if users request features or improvements, we can address them before full deployment. The only disadvantage is that the system may not be fully implemented at 100% initially, but involving users early on can help address any gaps.

## Requirements

When we started the development of the program, we knew there were several aspects we wanted to address. The first is inventory management. We identified a useful approach for implementing this feature in the program. Since we have a specific user responsible for managing inventory, we ensured they have the proper authentication to add all items. Additionally, we track each item through its barcode, as it serves as an easy identification method for products. If an item does not have a barcode, we allow manual entry, where a unique ID can be assigned. Every time an item is added to the database, a report is generated showing the time of modification. At the same time, we create backups to ensure data integrity.

The second requirement is sales.

As one of the most important parts of the program, the sales system interacts with the entire environment. Once the inventory and database are set up, sales can be processed. Every time a sale is completed, a record is generated in the database to track the transaction. Additionally, we ensure the database is regularly updated to keep everything running smoothly.

Reporting

After addressing the inventory and sales functionalities, we focused on reporting. Every time interaction occurs in the database, we ensure a log is created. This way, we maintain accurate records and ensure the highest level of security, as the database is critical to the system's operations.

Security

As we mentioned earlier, security is a top priority for the owner. To ensure this, we implemented different management levels within the program. For example, Level 1 users are unable to edit the inventory and are only allowed to process sales. Access to the system requires a password, which is provided by the owner. In addition, if an employee is terminated, the IT department can remove their user account from the system.

## Installation Process

Installation Process

Inventory Management System

Follow the steps below to correctly set up the Inventory Management System on your local machine:

1. Download the Project Files

Download the project files from the repository. (If you are unsure how to download from GitHub, we will provide a direct link for you to download.)

1. Extract the Project Directory

After downloading, unzip the file. You will find a folder named "inventorysystemV.1."

1. Create the Virtual Environment

Execute the provided script to create a virtual environment to manage all the dependencies. You will find a folder named "requirements" listing all the necessary dependencies. Execute the first script in the folder to set everything up.

1. Install the Dependencies

Execute "bash #2" to install all required dependencies, including SQL3, which is needed to keep the database updated.

1. Start the Database

Execute "bash #3" to start the database.

1. Run the Main Application

Run the Inventory Management app to start using the system.

1. Contact IT Manager for Issues

If you encounter any issues during the setup process, contact the IT Manager for assistance.

You will find the entire implementation plan that we have been working on in this document. It outlines all the details of the inventory management system, which is built using Python and MySQL. This system will meet all the requirements set by the owner, and we guarantee full efficiency by providing functionalities to create an inventory system and keep everything updated in real-time. This document also includes all the requirements that we previously identified during the analysis phase. Additionally, we provide a detailed installation process and describe the methodology we will use to ensure the best deployment.