Project Plan: Software Solution

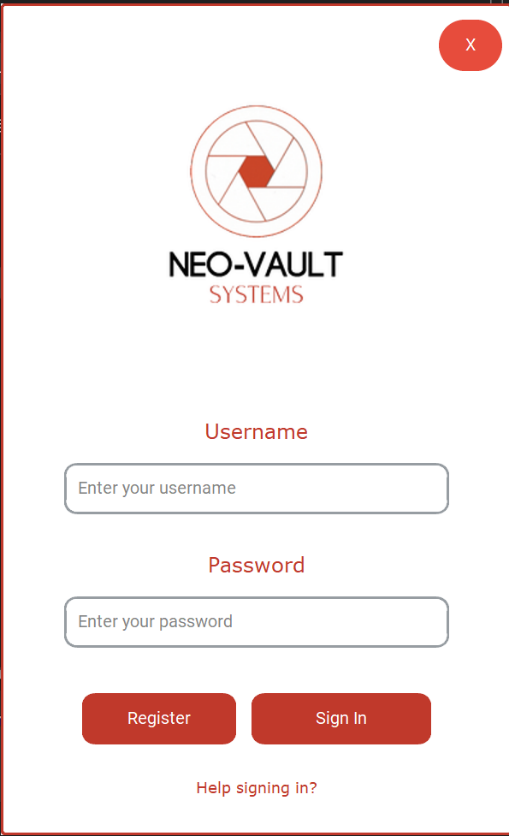
Team 4:

Anthony Colley

Joshua Swilling

Jose Vazquez

Kennedy Stokes-Sutton



Contents

[Introduction 3](#_Toc185157044)

[Project Organization 3](#_Toc185157045)

[Risk Analysis 4](#_Toc185157046)

[Schedule Risk 4](#_Toc185157047)

[Technical Risk 4](#_Toc185157048)

[Resource Availability 4](#_Toc185157049)

[Scope Risk 4](#_Toc185157050)

[Presentation Quality 4](#_Toc185157051)

[Risk Identification 5](#_Toc185157052)

[Hardware and Software Requirements 7](#_Toc185157053)

[Project Breakdown 7](#_Toc185157054)

[Project Schedule 8](#_Toc185157055)

[Monitoring and reporting mechanisms 9](#_Toc185157056)

[Planning and Design 9](#_Toc185157057)

[Implementing Inventory 9](#_Toc185157058)

[Implementing Orders 9](#_Toc185157059)

[Records 9](#_Toc185157060)

[Login 10](#_Toc185157061)

[Appendix 11](#_Toc185157062)

[Deployment 13](#_Toc185157063)

# Introduction

In this project, we introduce a user-friendly software solution designed to simplify inventory management for small businesses. Our software provides an intuitive system for tracking inventory and managing sales, eliminating many of the complications found in other programs. Inventory is the root of any business, knowing what is in stock and identifying items that are running low are essential tasks. With our software, business owners can easily monitor stock levels, track transactions, and control access by granting employees specific permissions for sales and inventory management. The software also streamlines the process of adding new products by using barcode scanning. Our goal is to make daily operations easier for both customers and employees, enhancing efficiency and accuracy.

As we certainly know, storing all information in a database can raise concerns about security. However, we assure that we prioritized reliability and security when managing the inventory. Storing all inventory information in the database as the most secure and efficient option.

# Project Organization

|  |  |  |
| --- | --- | --- |
| Roles |  |  |
| Designer/Organizer | Jose Vazquez | Ensures an easy design for customers and makes sure that the team can easily implement it into the program. Working on the database. |
| Organizer/Programmer-Backend | Anthony Colley | Ensuring that all assignments are completed on time according to our schedule and writing the program code. |
| Programmer-Frontend | Joshua Swilling | Write the program code, specifically the frontend, ensuring that the design meets the customer's requirements |
| Tester | Kennedy Stokes-Sutton | Conducting all analyses and testing everything to ensure that the program works at peak performance without bugs, guaranteeing customer satisfaction |

# Risk Analysis

There are several risks that we consider just to prevent those happening or if in any circumstance it happens, we will know how to manage.

## Schedule Risk

To minimize scheduling issues, the team develop a clear timeline with individual responsibilities and milestones. Weekly check-ins will help keep everyone aligned with the schedule. If any delay occurs, adjustments will be made to prioritize the essential features to be the first one deployed.

## Technical Risk

For any technical risks, the team will ensure backups are regularly saved to the cloud (GitHub) to prevent data loss. Each member will keep the local copies of the project files to mitigate risks related to cloud or local software failures. If any mistake by any team member is submitted to the cloud, we also meet before making changes to the program and are going to make a copy of the project for versioning.

## Resource Availability

All the team members’ roles and responsibilities are established early on. If any team member cannot contribute for whatever reason, task will be relocated temporarily to ensure the project remains on schedule. Tasks are going to be tracked in Trello app, and with the organizers among the team.

## Scope Risk

The Project scope is clearly defined at the start of the project, and any proposed changes or features that we want to add must be discussed and approved by the team to avoid any deviation from the planning scope or schedule.

## Presentation Quality

While coding is a big priority, the team will allocate the necessary time toward the end of the project to focus on all the presentation elements and documentation necessary to make the customer happy, ensuring the final product meets the quality standards without rushing.

# Risk Identification

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Description | Risk Priority | Risk Plan |
| Excessive Coding Requirements | Incomplete features could impact the functionality of the project. | HIGH | * Keep tracking every feature added, and if there is any issue facing it while implementing let the team know to look for the better solution. * Designers are looking for a new way of integrating the features. |
| Debugging Limitations | Incomplete debugging in the code could lead to a possibles bugs once the project is released | HIGH | * Making sure that every line of code is debugging to find if it is working as intended. |
| Not enough testing | Incomplete testing or missing some normal use during testing could lead to project failure or bad information entering in the database. | HIGH | * Exhausting testing to make sure the program works as intended. * Implement data validation for each user input field to ensure that the data entered matches the required type. For example, if a field expects a number, the system should display an error if the input is not a number. |
| The project has some issues with the security | Incomplete testing in the security of the program could lead to big issues in maintaining all the data. | HIGH | * Making sure that the user with the required permissions, such as "Inventory management," could edit the database, and the user with the lower permissions, such as "Sales Employer," could see the database but cannot edit the database. |
| Data loss for a hardware failure. | Any hardware issues can happen, and it could lead to potential data loss. | MEDIUM | * Making sure that every time that any transaction, item add, item deleted, and so on, a copy of the databases is done. * Ensuring those copies containing the last updated database. |
| Human error in data entry | Employees put the wrong item, or wrong price in an item. | MEDIUM | * Request data validation each time something is done. |
| The product does not meet the regulatory requirements | We do not meet the regulatory requirements | MEDIUM | * Request the necessary audits before we deploy the program. |
| Fraudulent transactions | A customer made a fraudulent transaction | MEDIUM | * Keep tracking of each transaction to identify the possible fraudulent transaction. |
| Poor Interface Design | The design is quite difficult to understand. | MEDIUM | * Providing enough documentation and training documentation to avoid any possible misunderstanding with the program, the team is also going to make sure that each button is intuitive and is an easy understanding for all kinds of employees. |
| Limiting Access | Limiting access is necessary to making sure that just the right people can do any big modifications | LOW | * Define in the documentation, which is every permission and how to give it to the employees. * If somebody makes any modification else without permission, they will be able to see which user did it, time, and date. |
| Not enough Training | Limiting training | LOW | * Ensure that the training book is clear and simple enough |
| Product becomes obsolete | The project becomes obsolete with the time | LOW | * Keeping maintaining the program once it is deployed to avoid any obsolescence. |

# Hardware and Software Requirements

Making sure of the best performance and reliability of the program, we have the next requirement:

Hardware

* Processor: Minimum dual-core processor, recommended quad-core.
* RAM: at least 4 GB; recommended 8 GB
* Storage: at least 128 GB SDD
* Display: Minimum 1200 x 720 resolution.
* Input Devices: Keyboard, mouse, and barcode reading.

Software

* OS: Windows 10 or later.

# Project Breakdown

For the project breakdown, we have defined the following activities to ensure better organization and meet all requirements. We believe this is the best way to keep everything updated and to allow any team member to access the features early if needed. Our primary focus should be on creating the database. Once the database is well-defined, the designers, coders, and testers can start working together to deploy everything as quickly as possible.

# Project Schedule

The team defined the schedule based on the project breakdown, setting key milestones to ensure everything stays on track. Most tasks were completed within the planned one-week time, while more complex tasks took two to three weeks as anticipated. The coding phase was successfully completed by the target date of December 4. Following the completion of the project, we managed to finalize the documentation and training materials within the planned one-week window. Additionally, we allocated some buffer time up to one week—for the final delivery, to account for any unforeseen challenges. Thanks to the team's dedication and effective planning, we are confident that we have met all deadlines and are on schedule for successful delivery.

A screenshot of a computer

Description automatically generated

# Monitoring and reporting mechanisms

The team, at the start of the project, defined a possible end date for the project solution based on the project breakdown, following the project schedule as we had planned. Once the project was completed, we began finalizing the documents and conducting pilot testing to verify that everything was working as intended. If the customer requires additional features, we could add them. However, our main goal was to deploy the first version with the essential tasks functioning, while ensuring it met the team's expectations.

# Planning and Design

During the first week, starting October 5, 2024, we established roles for everyone on the team and began planning the project. As we had been instructed, the main goal was to create a solution for tracking the inventory system and updating it in real time, using a database in the backend. The designer and organizer worked to create the best plan to achieve the desired results.

The planning was crucial and was completed without any inconvenience, thanks to the collaboration of all team members. We established that every week, each team member would have specific tasks, and it would be essential to have at least one meeting to report on progress.

We also identified that one of the most important things to address first was creating the database. Once the schema of the database was created, we could begin working on the inventory system. After some discussion, we decided to proceed with the database scheme presented in the plan, as we believed it was the best option.

# Implementing Inventory

Once the database was created, the developers began working on inventory implementation. The goal was to ensure that the inventory could be updated at any time and that it was well-organized. Employees should also be able to easily search for item information. It took approximately 11 days to complete the inventory, with constant communication between the coders to ensure alignment and agreement on the approach. Additionally, it was necessary to explain why having several tables in the database was important for optimal resource usage.

# Implementing Orders

Once the inventory system was completed and everything was functioning as expected, we moved on to the orders section. Initially, the plan was to create a sales section, but the customer requested an order management system instead. As a result, we adjusted the development to focus on creating an order manager, while also ensuring that employees could check inventory levels. It was crucial that orders would not include items that were out of stock.

Once the order management system was implemented correctly, we monitored the entire process to ensure the database was communicating properly with the orders module.

# Records

We also recognized the importance of having an organized inventory, so we added a feature to generate reports about sales. The admin would also be able to track which orders were about to be shipped and who had created each order. We did not want to lose any information, so we agreed with the developers and organizer to save all movements in the database. If the admin wanted to generate a report on orders, they would be able to download a file containing the date, item, quantity, and price.

# Login

Security was one of the highest priorities for our program. Since we wanted the system to be a useful tool, it was essential to maintain the highest security standards. Therefore, the login system was created to prevent unauthorized access, as the information could be sensitive.

Once all the features had been implemented, we began testing. The tester checked every aspect of the system that a normal user might interact with, and the testing file included descriptions of what had been evaluated and why. As expected, we found some bugs, but the good communication between testers and coders helped resolve these issues. We were able to release the first version of the system solution on Saturday, December 14, 2024.

# Appendix

We have the next activities that we are going to meet with the team.

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Estimated weeks to complete | Dependencies |
| 1.- Plaining project. | Planning the project is the first thing to do | 1 | None |
| 2.- Meeting to create the roles | In the meeting everyone gets a role about what they are doing to do in the project. | 1 | 1 |
| 3.- Doing the first activities to start working on the project. | The first activities such as risks assignments, project plan, and so on, are necessary to scope the project. | 1 | 1,2 |
| 4.- Create the database for the inventory | The first thing that we must address is the databases management need to create the data bases with all the features that the project will have | 1 | None |
| 5.-Implemeting the tables | Checking if the tables of the databases meet the highest standard for the team, and if it is well designed to prevent any duplication of the information | 1 | 4 |
| 5.1.- Testing the database | Testing if everything works and intended | 1 | 4,5 |
| 6.- Coding: Creating the inventory management | Start coding the inventory | 1 | None |
| 6.1.-Coding: Connecting SQL and Python. | Once the inventory is done, we can start the connection between the database and the python program. | 1 | 4,5,5.1,6 |
| 6.2.- Testing | Verify if the database and the python program do not have any issues, and all the information is correct. | 1 | 4,5,5.1,6,6.1 |
| 6.3.-Coding: Creating the record | Creating the record for each time somebody interacts with the database | 1 | 4,5,5.1,6,6.1,6.3 |
| 7.-Coding: Sales System | Coding the sales system | 2 | 6.3 |
| 7.1.-Coding: Connecting Inventory with Sales | Connecting the inventory part with the sales. | 1 | 5,6 |
| 7.2.- Testing | Testing that both systems are well connected. | 1 | 5,6,7 |
| 8.- Coding: Record Transaction (History) | Creating the record system1 | 1 | None |
| 9.-Coding: Connecting record transaction with the table(Transactions) | Connecting the record system with the table “Transactions” | 1 | 4,5.1 |
| 9.1.- Testing | Evaluating the records | 1 | 9 |
| 10.- Coding: Login Window | Create a login window every time that you start the program requires a login system | 1 | None |
| 10.1 Testing | Security Testing | 1 | 4,10 |
| 11.- General Testing | Evaluating all the functions of the program together | 1 | 10.1 |
| 12.- Reporting the test to the team | If there is any issue, then the coders will be able to identify and fix for a prior release. | 2 | 11 |
| 13.- Documentation | Once that everything is meeting the team standard, we should be able to finish the documentation of the project | 1 | 12 |
| 14.- Training Book | Creating the training books for all the users | 1 | None |
| 15.- Release V1.0 | Release the first version of the program | - | 14 |
| 16.- Maintenance | Keep working with the next features as a customer requirement |  |  |

# Deployment

As we are ready to release the pilot test and the first version of our program, we can confidently say that we have met all the team's expectations. As our front-end developer explained, it was crucial to make the user interface (GUI) easy to read and navigate. However, this does not mean that just because the GUI looks simply, it lacks functionality. Once all the goals of the project, such as orders, inventory, login, and other key features, were met, the GUI was fully developed to ensure the best possible user experience.

We also considered that some employees may find it challenging to adapt to working on a computer or may be reluctant to use a computer system at all. To address this, we invested significant effort into creating an intuitive solution, minimizing any potential delays during training and deployment. Our goal is to ensure that the transition is smooth and that everyone can easily adopt the new system.