# Contents

## Introduction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .2

## Project Organization. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .2

## Risk Analysis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

Schedule Risk

Technical Risk

Resource Availability

Scope Risk

Presentation Quality

Risk Identification

## Hardware and Software Requirements. . . . . . . . . . . . . . . . . . . . . . . . . 6

## Project Breakdown. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .7

## Project Planned Schedule. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .8

## Monitoring and reporting. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .10

## Appendix. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .11

# 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, establishing some dates to ensure everything is on track. Most activities are to be completed within one week, while the more complex tasks will take about two to three weeks. The target date for completing all coding is December 4. After the project is finished, the documentation and training materials should be completed within a week. We are allowing for some extra time—up to one week—for the final delivery date, just in case of any complications. However, we believe we can meet on these dates.

# 

# Monitoring and reporting mechanisms

As we know, as a team it is important to maintain communication among all team members in case, we encounter any discrepancies in the schedule or issues integrating something into the code. We will stay connected through Discord, holding weekly meetings, but we can send messages at any time and aim to be available 24/7. To avoid missing any activities, we are following the schedule on Trello. To ensure we are working with the most up-to-date version of the project, we are submitting our code to GitHub.

Each new addition should be added as a comment in the code and documented. Once the coding is complete, we should finalize the documents and create the training document.

# Appendix

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

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Estimated time to complete (Weeks) | 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 |  |  |