Inventory Tracking Program

By: Team Java
Jordan Hidalgo, Jorge Sanchez,
Raymond Duenas, Zackary Bair
Stan State
SWE CS 4800
09/15/2021

Team Contributions: Jordan Hidalgo, Jorge Sanchez, Raymond Duenas, Zackary Bair

Team Member	Contribution
Jordan Hidalgo	- Captured System Requirements - Captured Solution Description - Assisted in Problem Identification - Assisted in Identifying Problem Solution - Assisted in Developing Plan of work - Assisted in Developing Timeline - Assisted in Design of Prototype
Jorge Sanchez	- Captured Problem Description - Assisted in Problem Identification - Assisted in Identifying Problem Solution - Assisted in Developing Plan of work - Assisted in Developing Timeline - Introduced Project Idea to Team - Assisted in Design of Prototype
Zackary Bair	- Ensured Plan of Work Execution - Captured Meeting minutes - Captured Project PowerPoint - Assisted in Problem Identification - Assisted in Identifying Problem Solution - Assisted in Developing Plan of work - Assisted in Developing Timeline - Assisted in Design of Prototype
Raymond Dueñas	- Captured Timeline and Sprints - Presented Project Proposal - Assisted in Problem Identification - Assisted in Identifying Problem Solution - Assisted in Developing Plan of work - Assisted in Developing Timeline - Captured Low Fidelity Prototype

TABLE OF CONTENTS

A. Team Meeting Log

    A.1. Discord Meeting Log:

Meeting Date: 09/01/2021

Meeting Time: 2:00 PM - 3:00 PM

Attendees: Jordan, Jorge, Raymond, Zackary

- First discussion: Introductions and planning
- Came up with the idea for the project
- Defined the background, problem description, and proposed solution
- Created the low fidelity prototype
- Defined the teams deadlines as well as which part is each members responsibility
- Researched Java's SWING for the purpose of creating our GUI

Meeting Date: 09/04/2021

Meeting Time: 9:00 AM - 9:15 AM

Attendees: Jordan, Jorge, Raymond

- Discussed requirements of the program

Meeting Date: 09/05/2021

Meeting Time: 10:00 AM - 11:00 AM

Attendees: Jordan, Jorge, Raymond, Zackary

- Created GitHub repository
- Finalize functionality of the program

Meeting Date: 09/07/2021

Meeting Time: 5:30 PM - 5:50 PM

Attendees: Jordan, Jorge, Raymond, Zackary

- Reviewed project presentation

Meeting Date: 09/14/2021

Meeting Time: 3:15 PM - 5:00 PM

Attendees: Jordan, Jorge, Raymond, Zackary

- Created the cover page, team contribution, team meeting log, and the first two requirements of Report 1

Report 1

1. Customer problem statement:

a. Problem statement:

Small business owners have a hard time managing the business with very little man power at their disposal. That extra man power is costly leaving the owner to shoulder the extra work on their own. This extra work can unintentionally cause areas of neglect leading to inefficient management in a business. Manual documentation of available products is hard work and time consuming but it is a necessary part of business operations leaving very little breathing room for business owners to focus on other areas of the business. Neglecting the inventory documentation will lead to owners losing touch with their customer base, which dictate the demand. Business is supply and demand and neglecting the demand for a product is bad business.

b. Proposed Solution:

The (product name) is designed to help lessen the workload by simplifying the inventory process down to the click of a button so the user has more time to focus on other areas of his business without worry. (product name) will track growing and decreasing inventory and product loss across the stores available inventory as well as maintaining a count of what products are more popular so the user can make an educated decision whether to stock more of the popular products or drop the products that are not selling well. While these tasks can be done manually by hand, (product name) makes it faster and easier so the user has more time to focus on other tasks that require attention. Good business relies on good reliable information and (product name) displays that information for the user in a moment's notice right on their monitor to help make more educated and efficient decisions for the business. (product name) is the tool that business owners can rely on to help their business flourish.

c. Timeline:

The project turn over date is Monday December, 6th. In order to successfully complete the project by the given deadline, the team will be utilizing the SCRUM Software Development Life Cycle model. Separating the workload into two six to seven week sprints, we expect to complete a fully functioning Inventory tracking system before the turn over date. The execution of the sprints will be as follows.

i. **Sprint 1:**

Spanning six to seven week sprint 1 will be dedicated to determining functionality, architecture of class communication, design of graphic user interface and writing code for the core functions of the program. The sprint will begin with focused communication with the business owner to ensure desired functionality and interface are understood before moving to implementation. Once the expectations have been communicated and are understood the development team, utilizing case diagrams and flowcharts, will determine the flow of data through the program to develop an overall structure. Once it has been determined that the program structure and owner expectations are in alignment implementation will begin. Implementation for sprint one will only go so far as to code basic user interface and core operation functionalities including input capability, class communication, data storage and manipulation. Sprint one will conclude with a presentation of a low level prototype demonstrating the development team's execution of the core functionalities that the owner requested. The presentation will serve as an opportunity to ensure the project is progressing according to the owner's vision.

ii. **Sprint 2**

Being the final sprint, sprint 2 will also span six to seven weeks. Sprint 2 will initially be dedicated to rectifying any improper executions of the owner's vision revealed during the Sprint 1 presentation. After which the primary focus of Sprint 2 will be to complete the desired auxiliary functionalities of the program, developing the final graphical user interface, and debugging any unforeseen data flow issues. Week four of Sprint 2 will include an owner vision sync meeting. The goal of the mid sprint owner vision sync is to present a high-fidelity graphical user interface allowing for any miss interpretations of owner vision to be caught and addressed before the project turn over date.

d. **Glossary of Terms:**

- **Java**: a high level, object oriented programming language for developing programs.
- **User Interface (UI)**: a series of screens, pages, visual elements and icons that allow a user to interact with a product.
- **SCRUM:** is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.
- **Software Development Life Cycle:** a process used by the software industry to design, develop and test high quality software.
- **Class**: the basic building block of an object oriented language, is a template that describes the data and behavior associated with instances of that class.

2. System requirements:

a. Functional Requirements:

REQ-X	Requirement	PW	Brief Description
REQ-1	Ability to add items to inventory	1	Users should be able to add items to their inventory.
REQ-2	Ability to remove items from inventory	3	Users should be able to remove items from their inventory.
REQ-3	Ability to search for items in inventory	2	Users should be able to search for items in their inventory.
REQ-4	Ability to state reason for item removal	5	Users should be able to determine why an item is being removed from inventory..
REQ-5	Ability to rank item by popularity	6	Program should be able to rank items by their amount sold.
REQ-6	Ability to display the amount of items in inventory	4	Program should be able to display the amount of an item in the inventory.
REQ-7	Ability to display the popularity of items in inventory	7	Program should be able to display the popularity of the items in inventory.

b. Non-functional requirements:

NF-REQ-X	Requirement	PW	Brief Description
NF-REQ-1	Simple, accessible interface	3	Interface should be simple and easy to use, with a maximum response time of 4 seconds.
NF-REQ-2	Reliable	1	Program should be able to save data in response to a crash and reboot within 10 seconds.
NF-REQ-3	Secure	2	Program should be secure with hash passwords with a quick login of 5 seconds.
NF-REQ-4	Saved Data	1	Data is saved to a local database within 3 seconds.

c. User-interface requirements

UI-REQ-X	Requirement	PW	Brief Description
UI-REQ-1	Display amount of item in inventory.	2	Program should be able to display the amount of an item in inventory.
UI-REQ-2	Ability to add, sell, search for, and remove items	1	Users should be able to add, sell, and remove items from inventory.
UI-REQ-3	View item popularity	3	Users should be able to view the popularity of a selected item in inventory.
UI-REQ-4	Sort by popularity	3	Users should be able to sort and view their inventory by popularity.

d. Prototype

Inventory Manager

Add Item to Inventory:_____

    Add Button

Search Inventory Items:_____

    Search Button

Remove Inventory Items:_____

    Remove Button

Inventory Address Capacity

	A	B	C	D	E
1					
2					
3					
4					
5					

Page after command execution

Inventory Manager

Add Item to Inventory:_____

    Add Button

Search Inventory Items:_____

    Search Button

Remove Inventory Items:_____

    Remove Button

Inventory Address Capacity

	A	B	C	D	E
1					
2					
3					
4					
5					

Location for message regarding executed command. To display

- Item Added Message
- Item Location Message
- Item Removed Message
- Unexpected Error Message

e. Constraints

The primary IDE that we will be using is Eclipse. In Eclipse we will be adding the Java package as well as a package to allow us to create a meaningful and simplistic GUI. The constraints that we will face while using Eclipse are each file of the program can have only a single team member edit the file at a time. This constraint will be solved by having each team member dedicated to an assigned file, so that no two members are trying to edit a single file at the same time. To allow for proper management of the program's files, all members will be using a common GitHub repository to upload the most recent version of each file.