

Inventory Tracking Program

By: Team Java

Jordan Hidalgo, Jorge Sanchez,
Raymond Duenas, Zackary Bair

Stan State

SWE CS 4800

10/12/2021

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

Team Member	Contribution
Jordan Hidalgo	<ul style="list-style-type: none">- Captured Class Diagram- Captured System Architecture and System Design- Assisted in Determining System Architecture- Updated System Architecture and System Design- Updated Class Diagram- Captured design of test- Assisted in the non-functional/user interface requirements
Jorge Sanchez	<ul style="list-style-type: none">- Developed sequence diagram to required detail- Assisted in Determining Algorithms and Data Structures- Updated sequence diagram- Assisted in updating class diagram- Assisted in the non-functional/user interface requirements
Zackary Bair	<ul style="list-style-type: none">- Developed initial sequence diagram- Assisted in System Architecture and System Design- Developed Identify Subsystems Graphic- Updated System Architecture and System Design- Updated Identify Subsystem Graphic- Assisted with updating class diagram
Raymond Dueñas	<ul style="list-style-type: none">- Captured Data types and Operation signature- Assisted in Determining System Architecture- Assisted updating System Architecture and System Design- Assisted in updating Class Diagram- Captured the coverage and functionalities of test

TABLE OF CONTENTS

A. Team Meeting Log	3
A.1 Discord Meeting Log	3
1. Sequence diagrams:	4
a. Add Item	4
b. Remove item	5
c. Search	6
d. Update Inventory	7
2. Class diagram and interface specification:	8
a. Class diagram:	9
b. Data types and Operation signature:	10
3. System Architecture and system design:	10
a. Architectural style:	10
b. Identify subsystems:	10
c. Mapping subsystems to hardware:	11
d. Persistent data storage:	11
e. Network protocol:	11
f. Hardware requirements:	11
4. Algorithms and Data Structures	11
a. Algorithms:	11
b. Data Structures:	12
5. User Interface Design and Implementation	12
a. Design of tests	12
1. Test coverage and their functionalities	14
2. Test coverage and functionalities:	14
3. Non-Functional/User-interface Requirements:	14
6. References:	14

A. Team Meeting Log

A.1. Discord Meeting Log:

Meeting Date: 10/12/2021

Meeting Time: 6:30 PM - 8:30PM

Attendees: Jorge, Raymond

- Created new report
- Transferred relevant information from report 1 to report 2
- Developed sequence diagram to meet report 2 requirements
- Completed Data types and operations signature

Meeting Date: 10/18/2021

Meeting Time: 6:00 PM - 7:00 PM

Attendees: Jordan, Raymond, Zackary

- Designed architecture and system design
- Identified subsystems
- Specified data storage

Meeting Date: 10/26/2021

Meeting Time: 4:00 PM - 5:20 PM

Attendees: Jordan, Jorge, Raymond, Zackary

- Added sequence diagram for log-in use case
- Refined class diagram
- Refined subsystems and system design

Meeting Date: 11/1/2021

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

Attendees: Jordan, Raymond

- User Interface Design and Implementation
- Design of test
- Test coverage and their functionalities

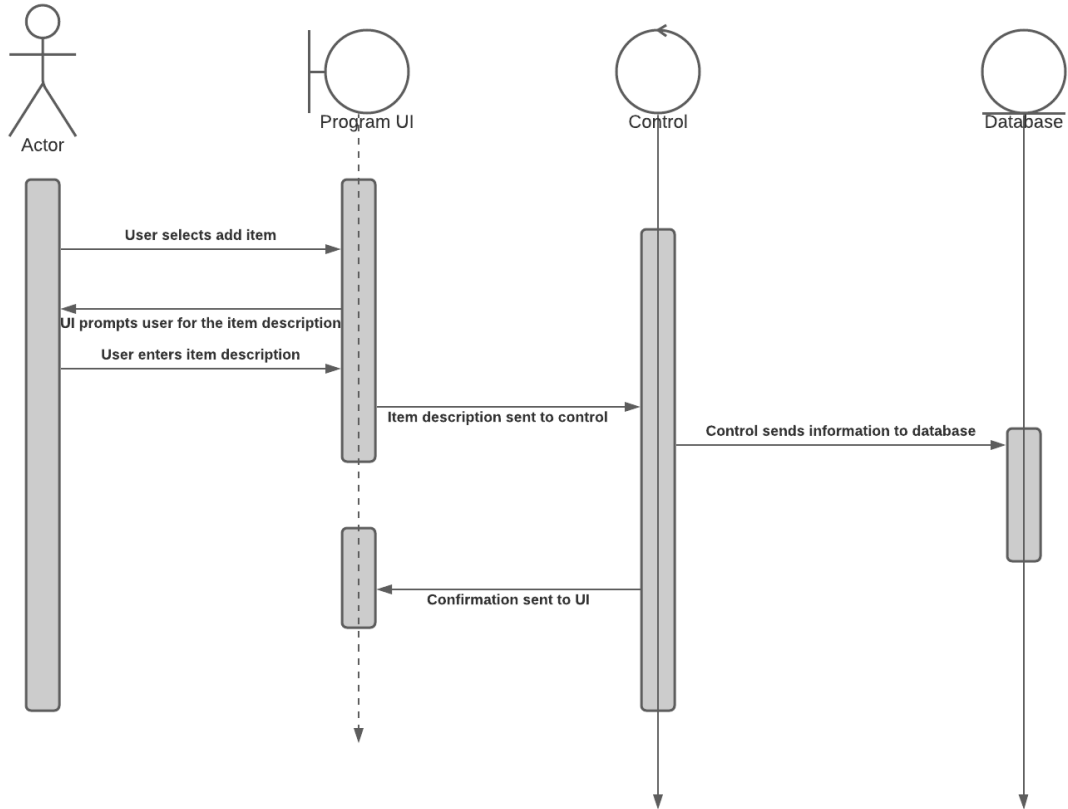
Meeting Time: 11:00 PM - 11:30 PM

Attendees: Jordan, Jorge

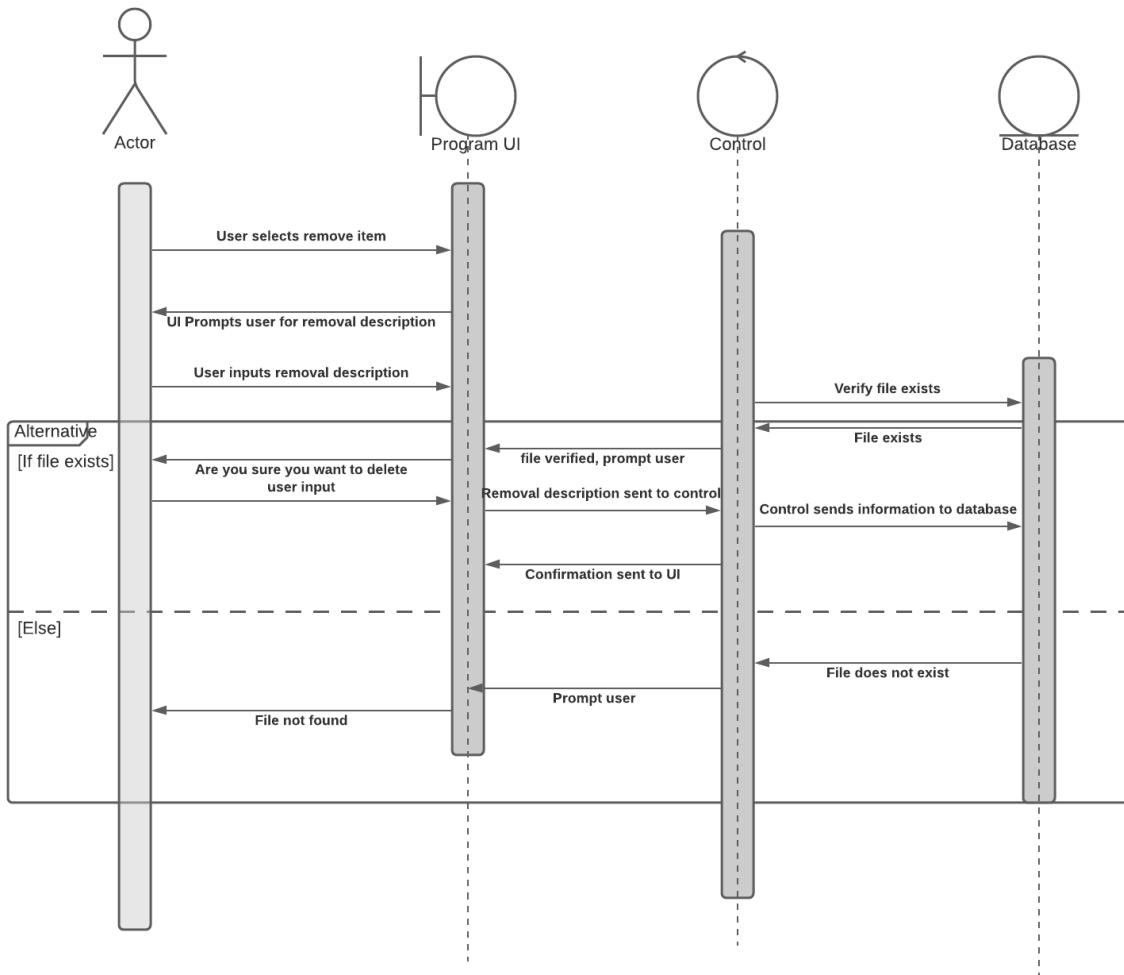
- Non-Functional/User-interface Requirements:

1. Sequence Diagrams:

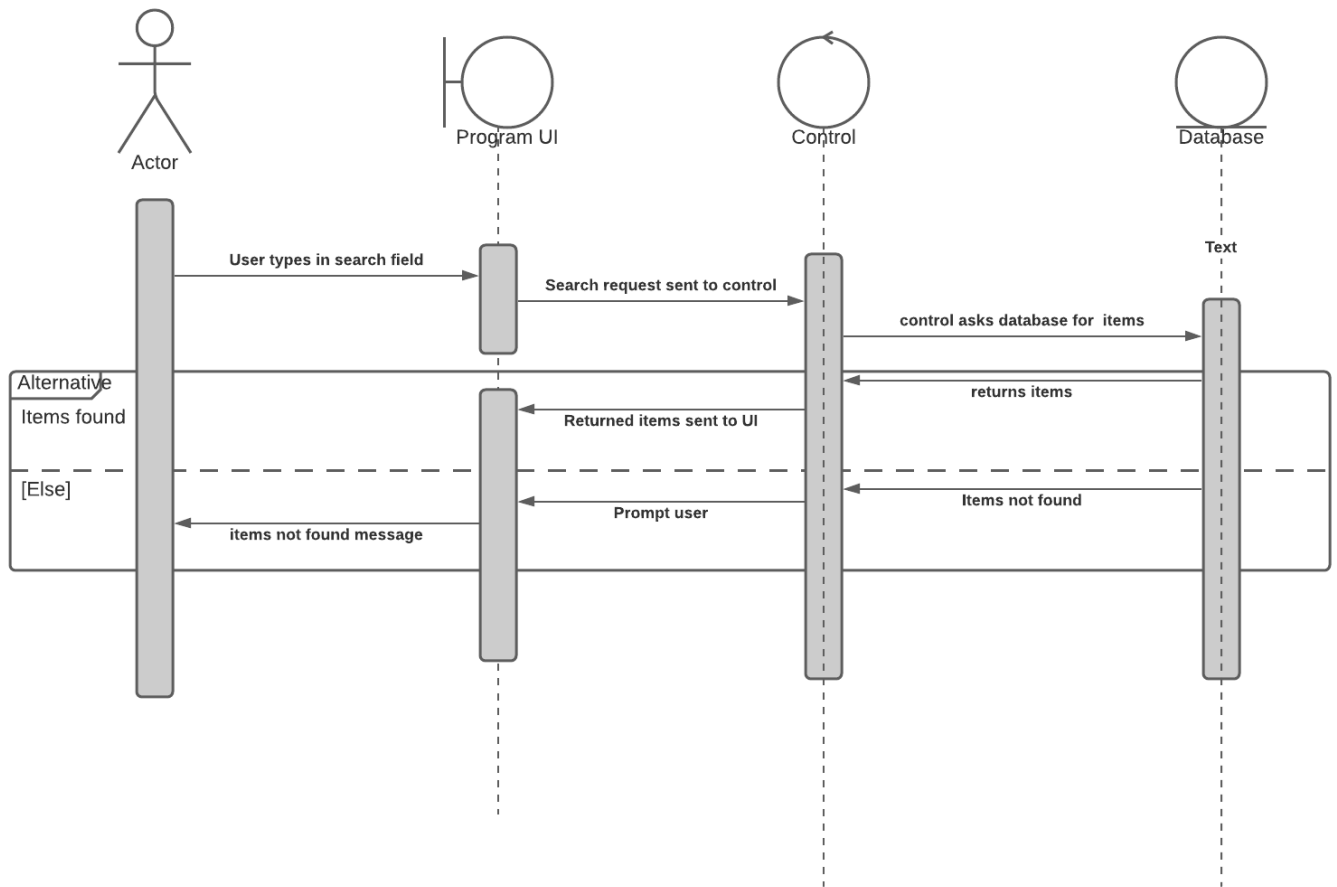
a. Add Item



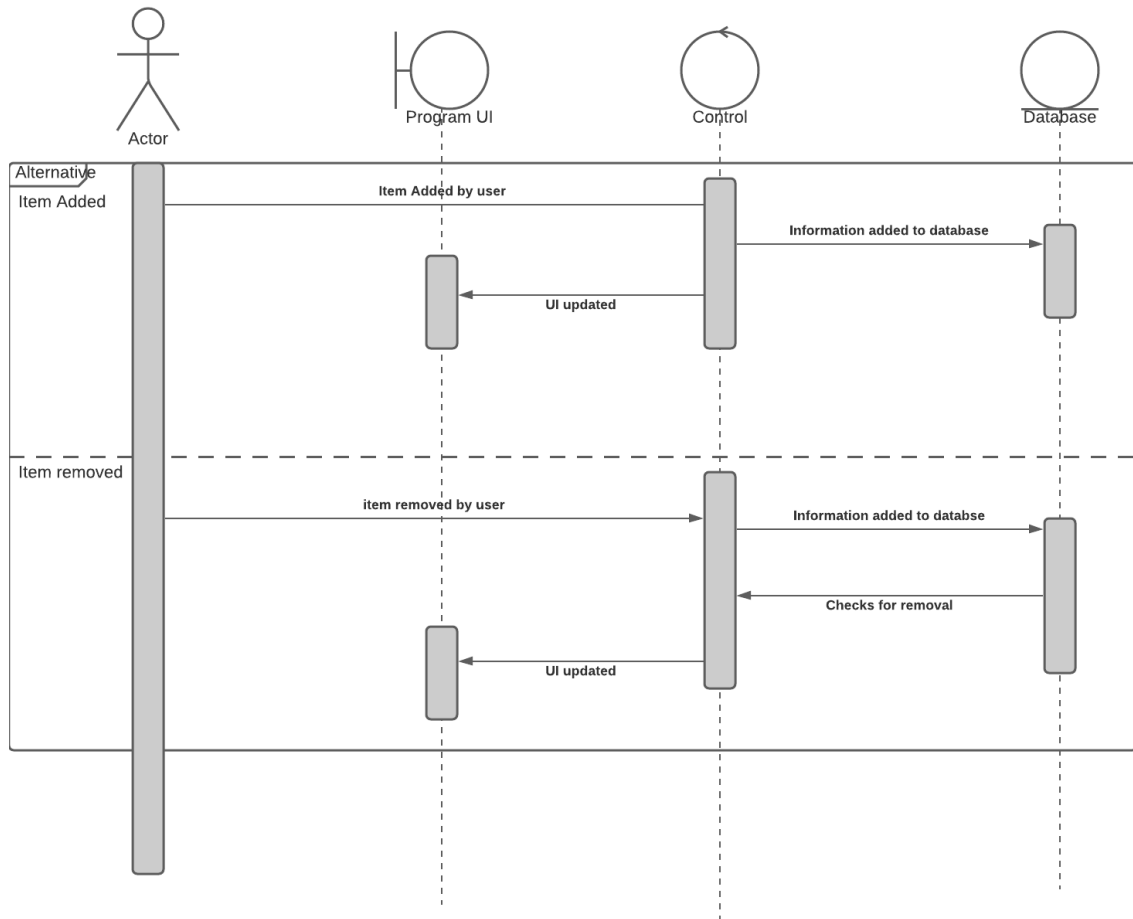
b. Remove Item



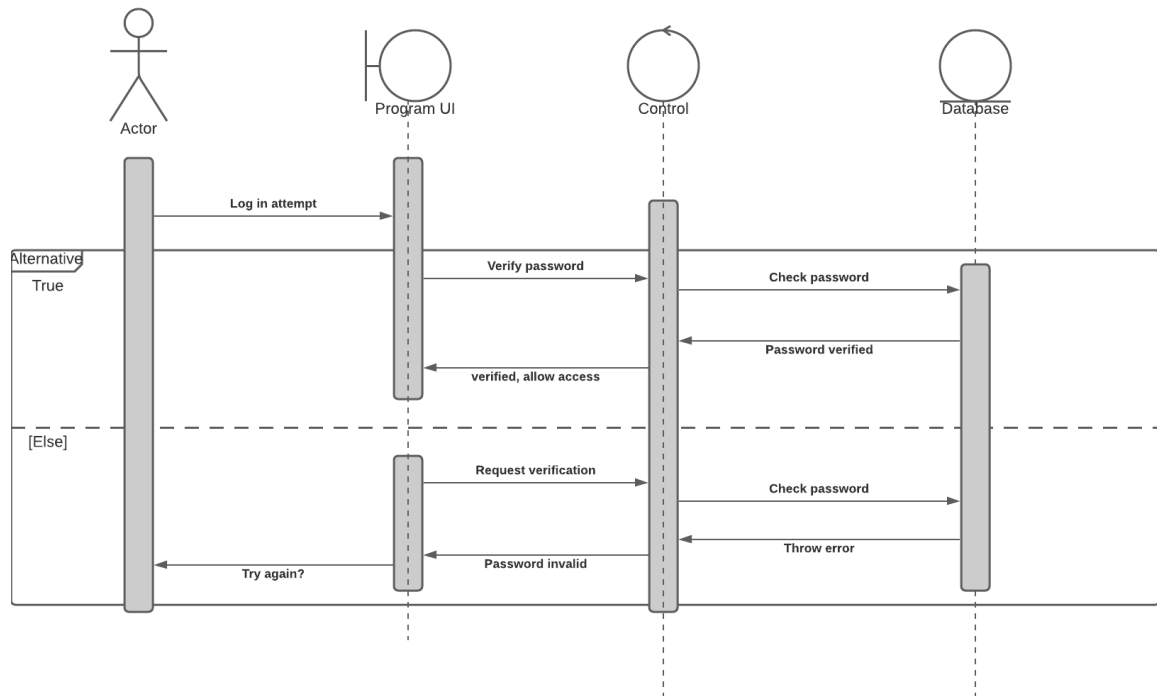
c. Search Use Case



d. Update Inventory

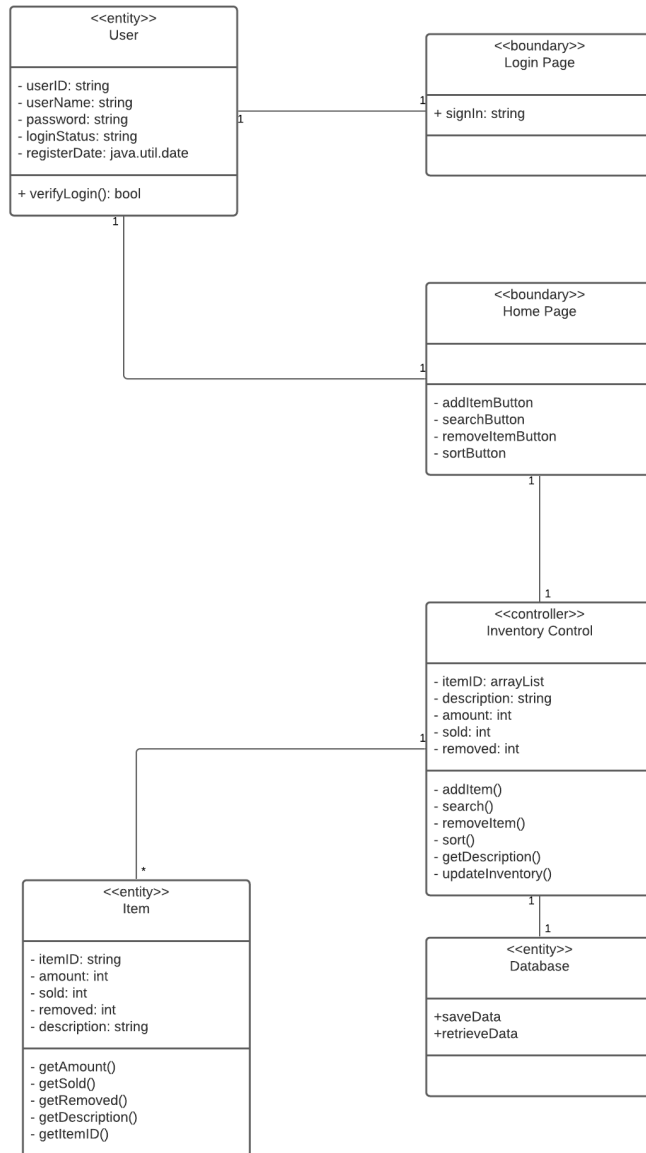


e. Log in



2. Class diagram and interface specification:

a. Class diagram:



1

b. Data types and Operation signature:

i. Inventory Control

1. Operations:

- `addItem()` - item is added to inventory stored in database
- `search()` - item is located in inventory database
- `removeItem()` - item is removed from database
- `sort()` - inventory from database is displayed in a specified order

- getDescription() -
- updateInventory() -

2. Attributes:

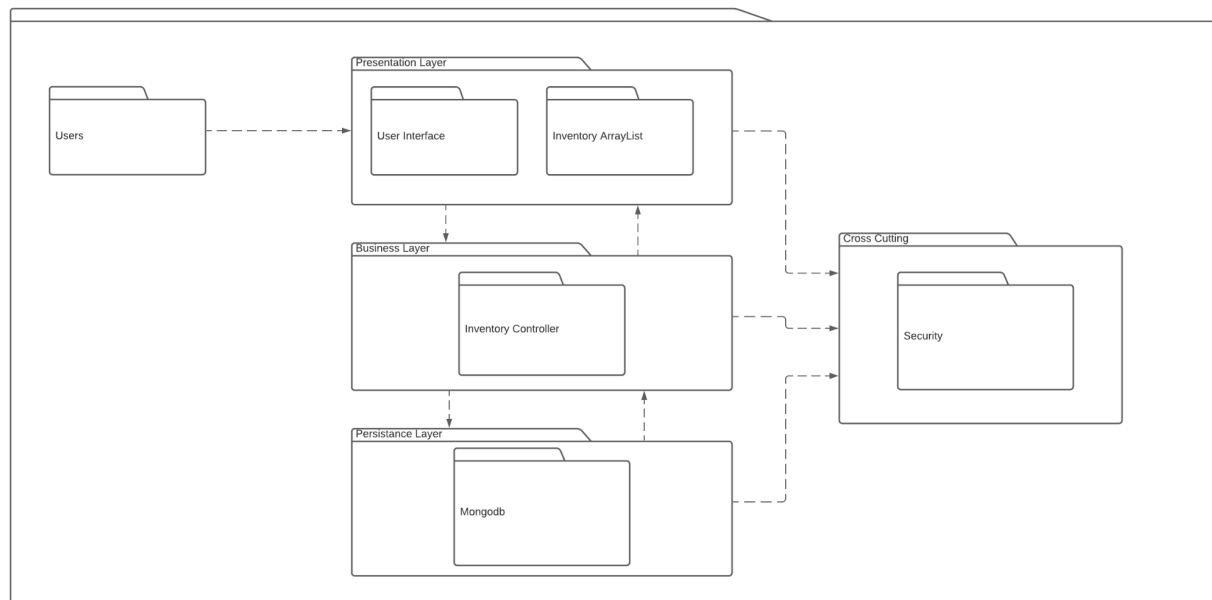
- List itemID - this will hold the list of inventory item id
- String description - this will describe the inventory item
- Int amount - this will be the amount of an item currently in the inventory
- Int sold - this will be the amount of times an item has been sold
- Int remove - this will be the quantity of an item that is being removed from the inventory

3. System Architecture and system design:

a. Architectural style:

- The architectural style that we will be using is the Layered Pattern. We will decompose our project into groups of subtasks that are contained within their own respective classes. For Sprint One, our group will be focusing on creating and connecting the database, creating the home page UI, and creating a list of items. Sprint Two will be spent focusing on creating controllers to link the UI and database, as well as any additional functionalities such as search and ranking.

b. Identify subsystems:



c. Mapping subsystems to hardware:

Our system will require multiple machines, one machine will be the local computer which our user physically interacts with. This machine will run every

subsystem besides the database. The second machine will be the server utilized by mongodb which will contain the program's database subsystem.

d. Persistent data storage:

We will store our data using an online database provided by Mongodb. Early versions of our system may store data locally; however, it is our hope to get our online database running as soon as possible. In doing so we hope to avoid relying on local storage and pull all data from the online database.

e. Network protocol:

Our system will use the Transmission Control Protocol, TCP for short. This network protocol was chosen for its reliability of data transfer. While most of the software is going to be run from the user's local machine, the TCP network protocol will allow for accurate and reliable connection to Mongodb. This is meant to ensure that the business is able to order inventory when stock is low without ordering unnecessary stock.

f. Hardware requirements:

For our system to work properly, a good network connection will be required with a desktop or laptop running Windows.

4. Algorithms and Data Structures

a. Algorithms:

We will utilize an algorithm for sorting the inventory elements in a variety of ways, comparing all elements in the database and placing them in order based on their ranking with respect to a selected criterion. Ranking will be determined by comparing the integer values for the selected characteristic of the various elements within the inventory database. We will also be utilizing algorithms for adding inventory items and removing inventory items from the database.

b. Data Structures:

The data types that we will implement in the execution of our program include list, string, int and boolean. List will be used to hold inventory items as objects containing multiple characteristics within the database. The string data type will be utilized to assign inventory item names, item descriptions, and describe the reasons for the removal of inventory items. The inventory objects of the list will also hold int values to track the amount of an item currently in stock, and the number of times an item is removed for a particular reason. Int will also be used to determine the ranking of items with respect to each other by comparing the int values for particular characteristics of objects within the list. Boolean will be used in displaying a sorted list.

Data Types: ArrayList, String, Int, Boolean

5. User Interface Design and Implementation

a. Design of tests

1. Test coverage and their functionalities

1.1

Function	1) Ability to add items to inventory
Description	- When adding an item the user should be able to press the add item button and input their data into the text box. That data should be saved into the database.
Purpose	- New items should be saved into the database.

1.2

Function	2) Ability to remove items from inventory
Description	- When removing an item the user should be able to press the remove item button and input their data into the text box. That data should be saved into the database.
Purpose	- Removed items should be removed from the database.

1.3

Function	3) Ability to search for items in inventory
Description	- Given a predetermined data set of items, the user should be able to use the search button to locate any of the items in inventory. This item and its attributes should be properly displayed in the list of items.
Purpose	- System should properly search and display the user's target item.

1.4

Function	4) Ability to state reason for item removal
Description	- When using the remove item button, user should be able to determine the reason for

	the item's removal
Purpose	<ul style="list-style-type: none"> - System should allow user to properly specify the reason for an item's removal

1.5

Function	5) Ability to rank items by popularity
Description	<ul style="list-style-type: none"> - Given a predetermined list of items and their attributes, the system should be able to rank these items by amount in inventory, amount sold, or removed when the sort button is pressed by the user.
Purpose	<ul style="list-style-type: none"> - System should properly rank items when prompted by the user

1.6

Function	6) Ability to display amount of items in inventory
Description	<ul style="list-style-type: none"> - Given a predetermined data set of items and their attributes, the program should be able to display the amount of said items properly in the list of items and item description.
Purpose	<ul style="list-style-type: none"> - System should properly display item amount

1.7

Function	7) Ability to display popularity of items in inventory
Description	<ul style="list-style-type: none"> - Given a predetermined data set of items and their attributes, the program should be able to display the popularity of an item (calculated from amount sold) properly in the list of items and item description.
Purpose	<ul style="list-style-type: none"> - System should properly display the item popularity

Function	8) Ability to access program features using username and password
Description	- Given a predetermined data username and password, the program should be able to login properly to access the inventory list, data, and program functions.
Purpose	- System should properly log the user into the system

2. Test coverage and functionalities:

Our first report outlined seven different functional requirements including: 1) Ability to add items to inventory, 2) Ability to remove items from inventory, 3) Ability to search for items in inventory, 4) Ability to state reason for item removal, 5) Ability to rank item by popularity, 6) Ability to display the amount of items in inventory, 7) Ability to display the popularity of items in inventory. We have since added an eighth function in recognition of the need for a secure login feature. 8) Ability to access program features using username and password. We have developed test cases with respect to these functions to ensure proper functionality of the program. Our tests outline to the best of our ability the most accurate expected result for each function of our program when prompted by user input.

3. Non-Functional/User-interface Requirements:

This project's goal is to develop an effective inventory management system for small business owners. The program is made with simplicity in mind and aims to minimize the effort for the user by keeping the majority of information on the home screen. The program has alerts that are color coded in order to quickly take note which items require the most attention. This program includes features such as adding items, removing items, searching for items, and sorting the inventory list by using the buttons displayed beside the inventory list. The end goal is an easy to use, simple to manage system that allows the user to properly keep track of and maintain their inventory stock.

6. References:

a. Sequence diagram design

- i. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

b. Network Protocol

- ii. <https://www.omnisecu.com/tcpip/transmission-control-protocol-tcp.php>