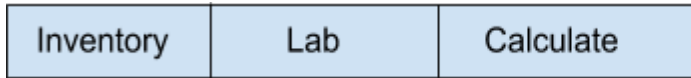


Criterion B: Design

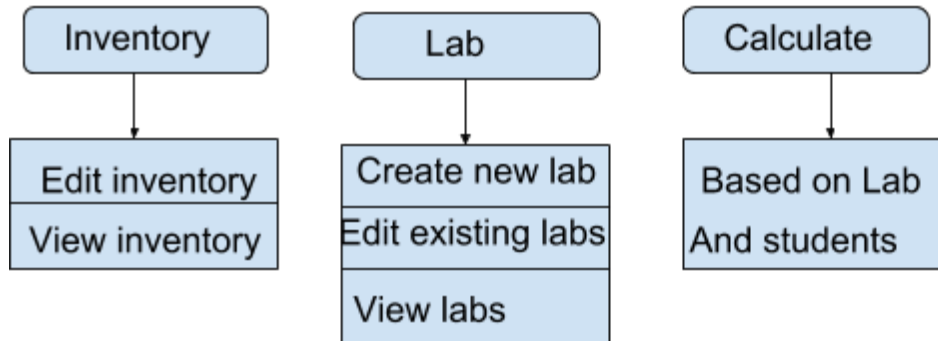
TABLE OF CONTENTS

	<u>PAGE</u>
Overall Structure	
UML Diagrams.....	2
Class Descriptions.....	3
Data Flow Diagrams.....	4
Database Structure	
Proposed Database Creation.....	8
Database Tables.....	8
GUI Representation	
Initial Sketches and Client Feedback.....	9
Menu/Start Frames.....	10
Lab/Calculate/Inventory Frames	13
Test Plan.....	17

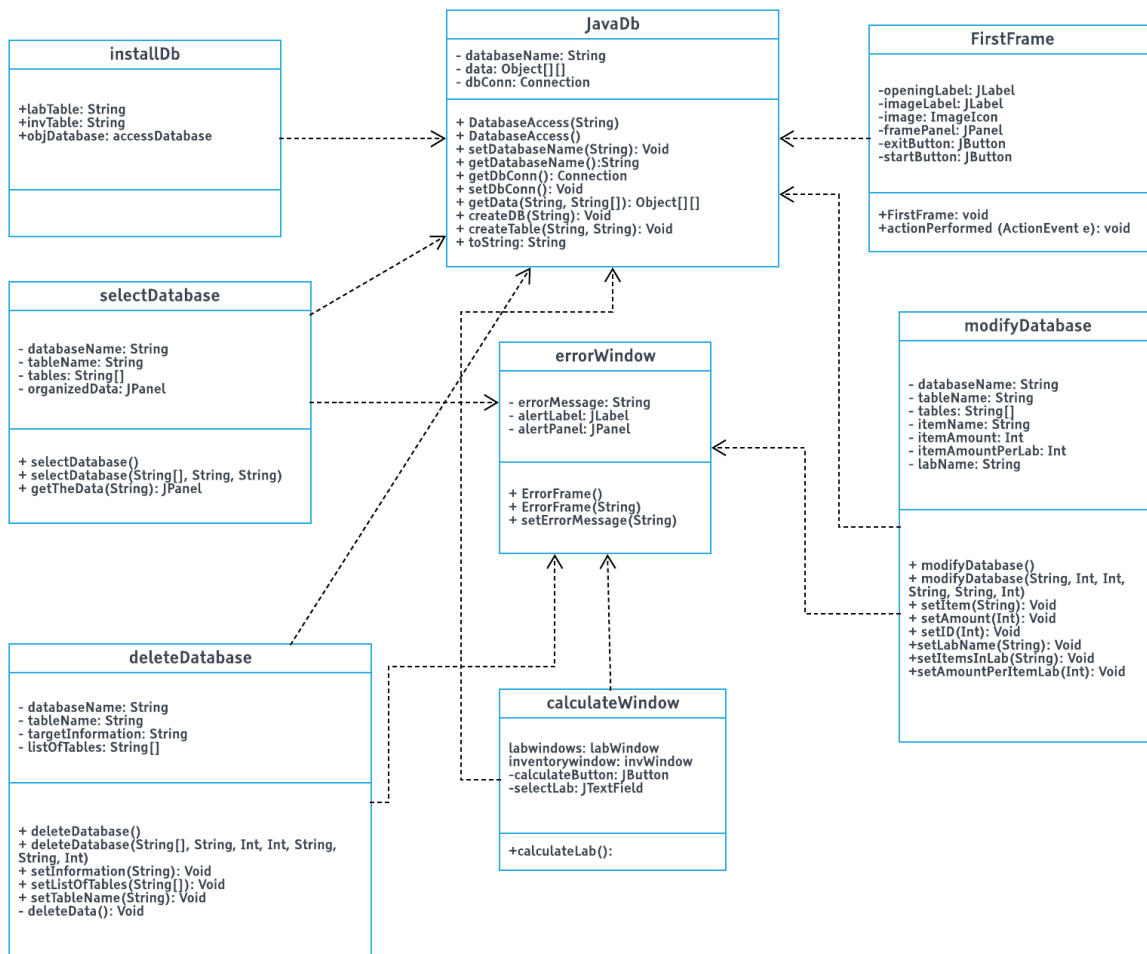
Primry functions or elements required to be kept in the Menu Bar or on the Frame:



Sub-functions for each menu:



Relationships between the classes:



Functionality of each class:

installDb - independent class that runs *once* to install the database

JavaDb - holds the information/data in the database and allows for selection, update, and deletion of data

SelectDatabase - allows for the display of data onto table through selection

deleteDatabase - allows for deletion of data

ModifyDatabase - allows for the updating of the database with new information

FirstFrame - visual elements for the user to choose to customize inventory or labs

CalculateWindow - lets user calculate if they can perform lab based on the inventory counts

ErrorWindow - A purely GUI class that notifies the user when there is an error in the table creation

Description of overall data flow:

- User presented with a home window where they can start the program or see the help menu
- When “Start” button is pressed user has options of either inventory edit, lab edit, or calculate
- When inventory is clicked user can input item type, id, and number of items
- When the lab is clicked user can input type of lab and type of items required as well as the number of those items required
- When calculate is clicked the program subtracts the amount of inventory by lab requirement and if the amount is > 0 , it *can* do the lab but if the amount is < 0 , it prompts user saying the lab requires more inventory than current levels

Data Flow Diagrams:

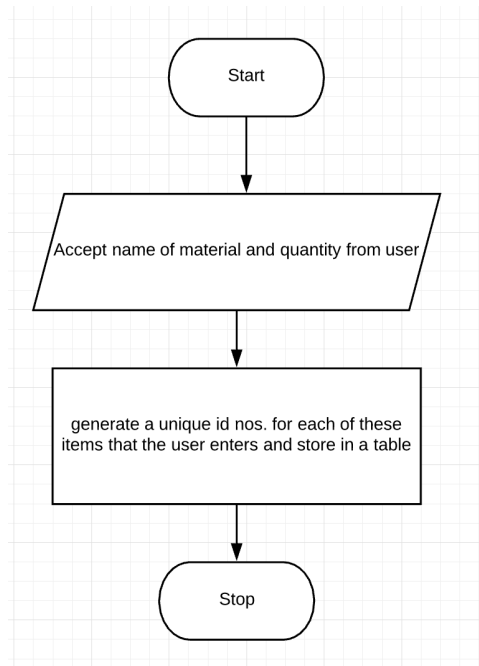


Figure 1 - Adding a new inventory item to inventory

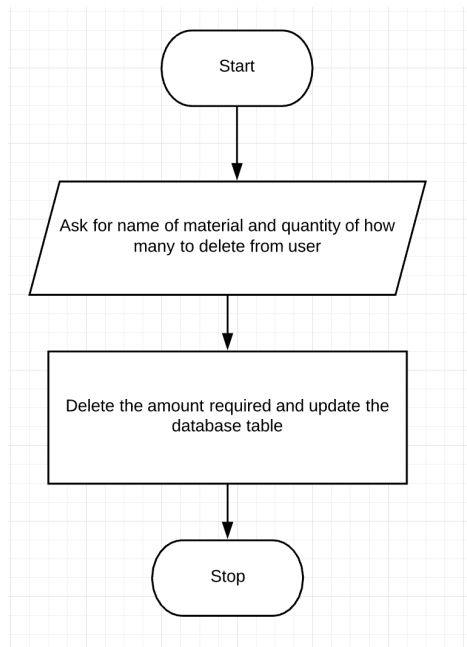


Figure 2 - Deleting an inventory item from the inventory

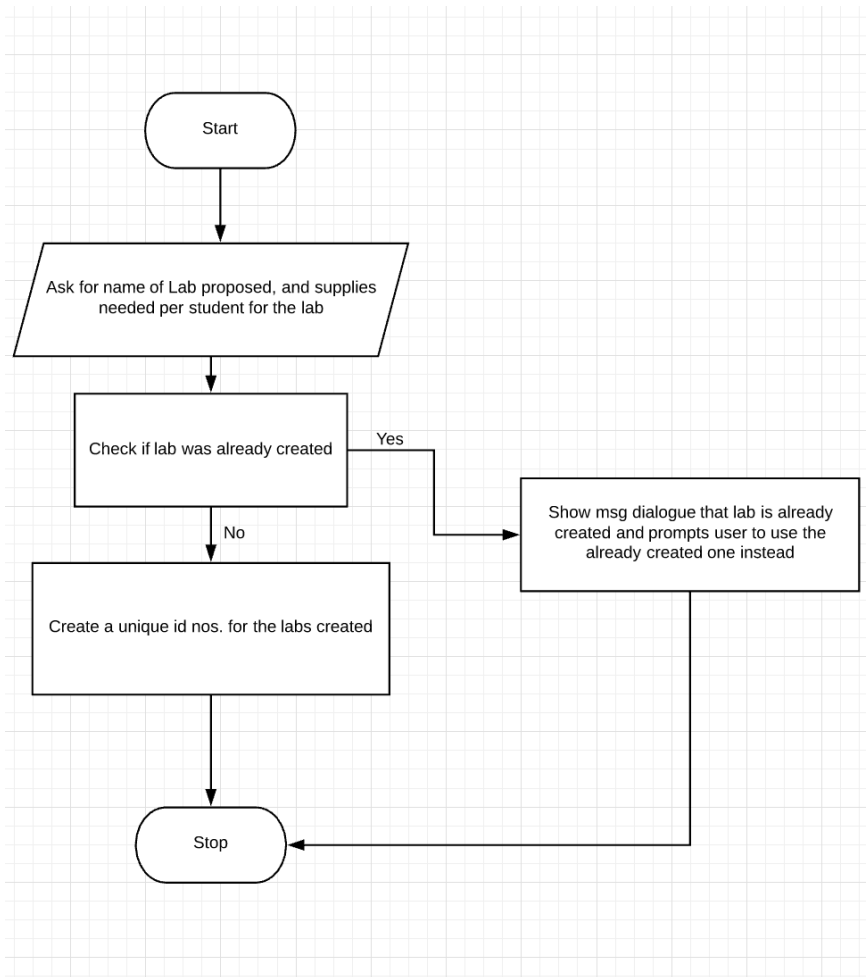


Figure 3 - Creating a new lab

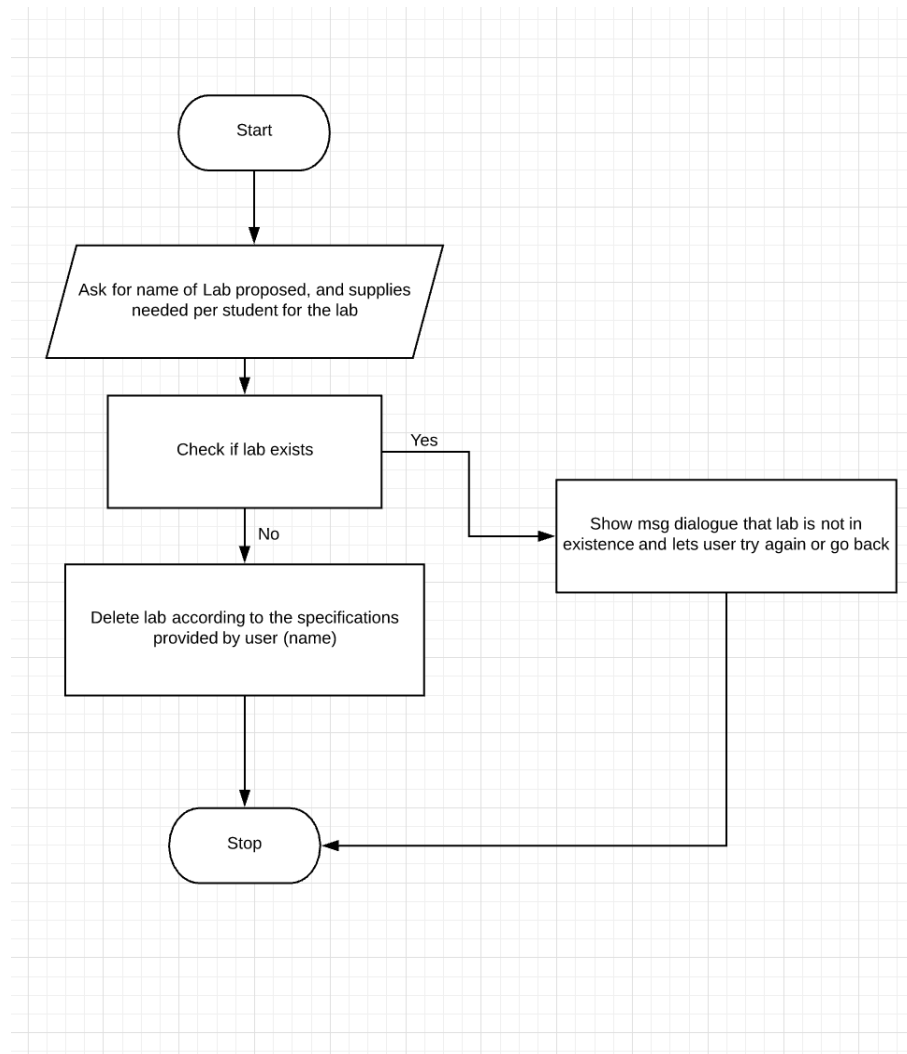


Figure 4 - Deleting an existing lab

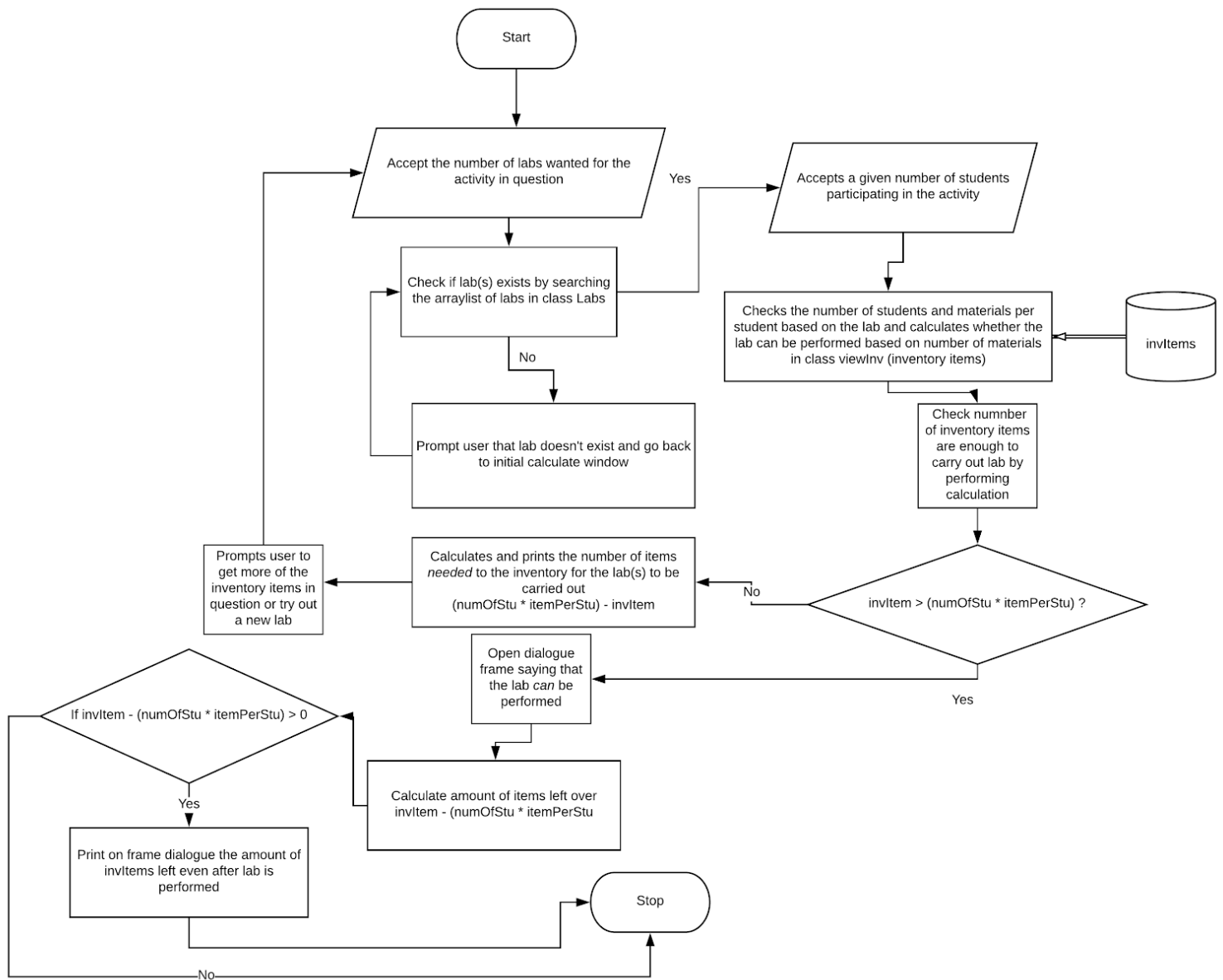


Figure 5 - Calculating whether lab can be performed based on type of lab and the number of supplies in the inventory

Databases and files created manually before writing the program

A database called LabInventory.mdb is created in the project folder where all the java classes are located, with the following tables in the third normal form.

PRIMARY KEYS IDENTIFIED BY HIGHLIGHT

Item	Amount of item

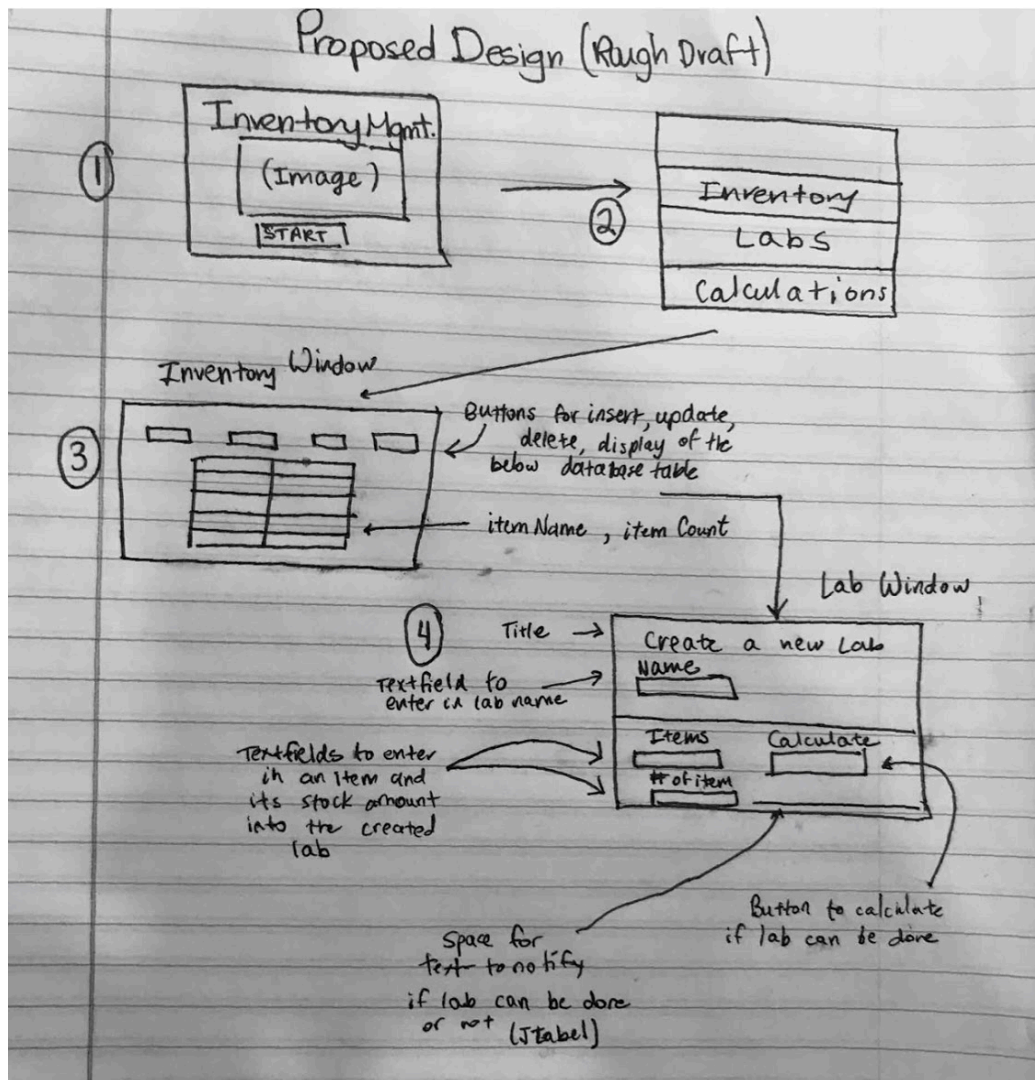
Figure 6 - This shows the inventory table in its 3NF where the primary key is 'Item' and non-key is 'Amount of Item'. Whenever the user creates a new inventory item, the name of the item and its amount is given to display on the table. Then the table below it shows the Item with its respective ID, where item is the primary key in table two.

Lab name	Items in Lab

Items in Lab	Amount of item needed in Lab

Figure 7 - This shows the labs table in its 3NF where the primary key is 'Lab name' and non-key are the item name in the lab. Then in table two, the 'items in lab' are the primary key with amount of the item required in lab column is the non-key that depends on the items in lab

Initial Concept/Design Sketches sent to {Client}¹



According to {Client}'s email where I showed her my proposed design, I need to make sure I have a "help descriptor"² to use in the application (in the second window). Additionally, I need to have space on the Labs window that can list all of the labs for {Client}

¹ As shown in Appendix B.1

² Refer to Appendix B.2 for the full email from {Client} regarding the proposed design

Design (Second Draft) of the frames/windows (Graphical User Interface)

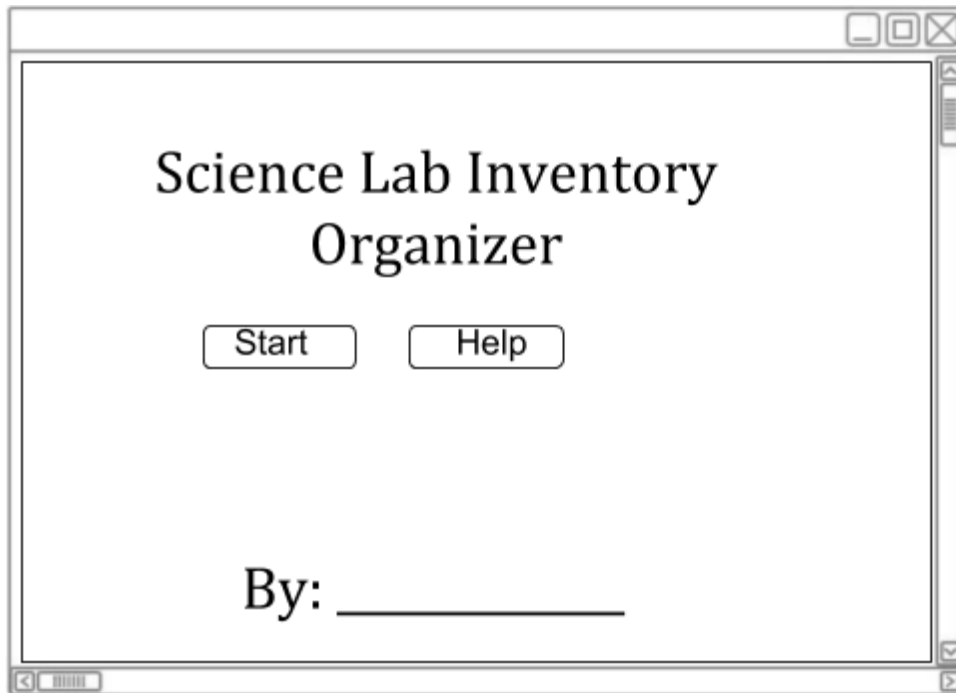


Figure 8 - Welcoming frame/window for the program with start and help options

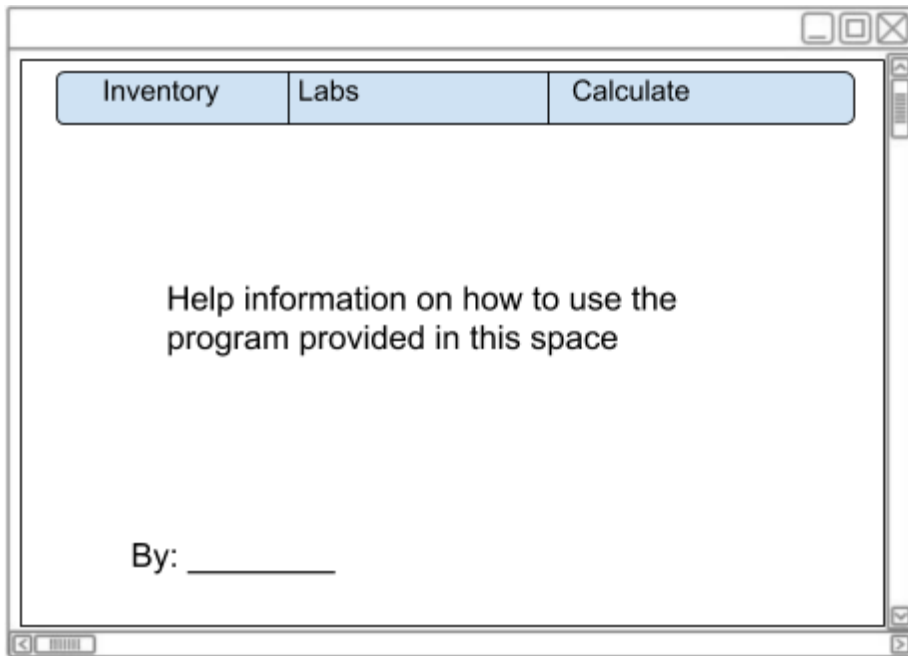
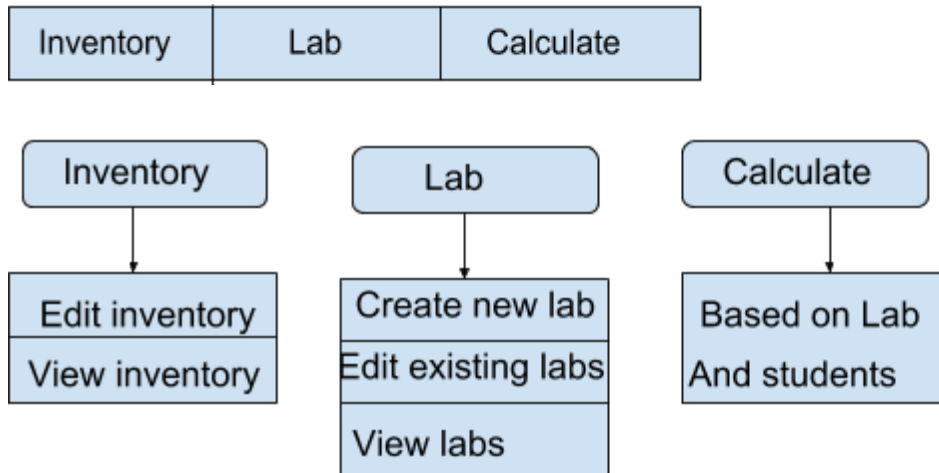
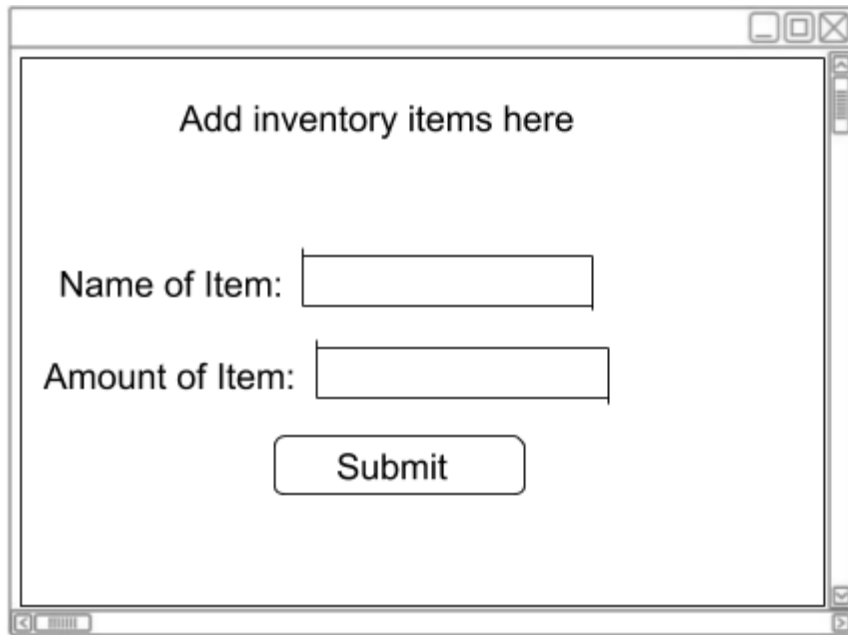


Figure 9 - Frame that appears after pressing the start button with the JMenuBar of JMenu of Inventory, Labs, and Calculate, with their own respective JMenuItems as shown previously. Frame also contains help information and credits to programmer



- When inventory is selected and then View Inventory is clicked, the frame will display the Inventory Table as shown previously with the item, amount of item in one table and then item with ID shown
- Same goes when Lab, when view labs have selected the frame, will display the Labs table as shown above where lab name and item for lab shown in one table and then the item with the amount of the item required in the lab shown in the second table



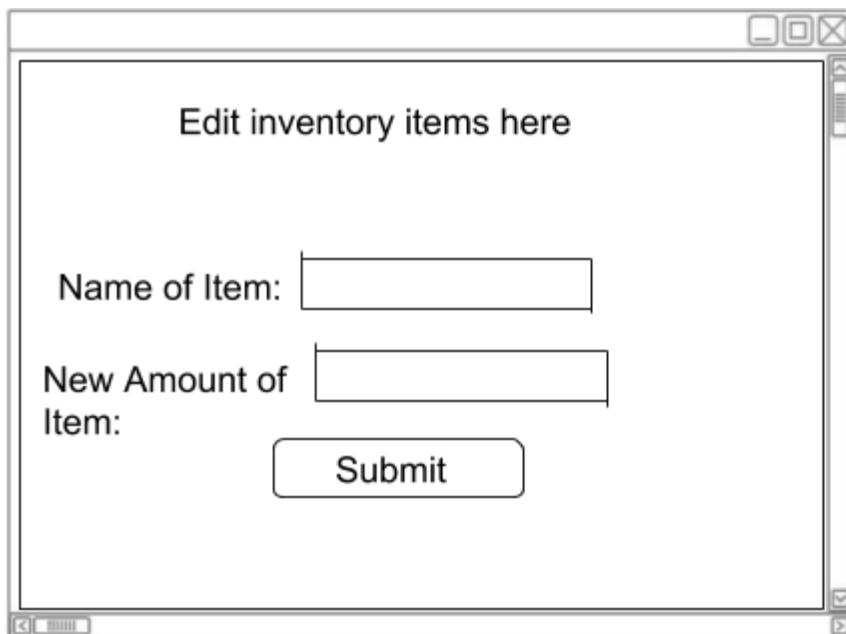
Add inventory items here

Name of Item:

Amount of Item:

Submit

Figure 10 - Frame for adding inventory items



Edit inventory items here

Name of Item:

New Amount of
Item:

Submit

Figure 11 - Frame for editing inventory items

A window titled "Add Labs here" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains three input fields and two buttons. The first row has the label "Name of Lab:" followed by a text input field and a button labeled "Add lab". The second row has the label "Name of Item:" followed by a text input field. The third row has the label "Number of Item:" followed by a text input field. Below the third input field is a button labeled "Add item".

Figure 12 - Frame for adding labs

A window titled "Edit Labs here" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains three input fields and two buttons. The first row has the label "Name of Lab:" followed by a text input field and a button labeled "Add lab". The second row has the label "Name of Item:" followed by a text input field. The third row has the label "Number of Item:" followed by a text input field. Below the third input field is a button labeled "Add item".

Figure 13 - Frame for editing labs

Calculate here

Name of Lab:

Number of Students:

Figure 14 - Frame for calculating labs with inventory

Calculate here

You have

Items left after labs

Figure 15 - Frame when user has items left over after calculating labs

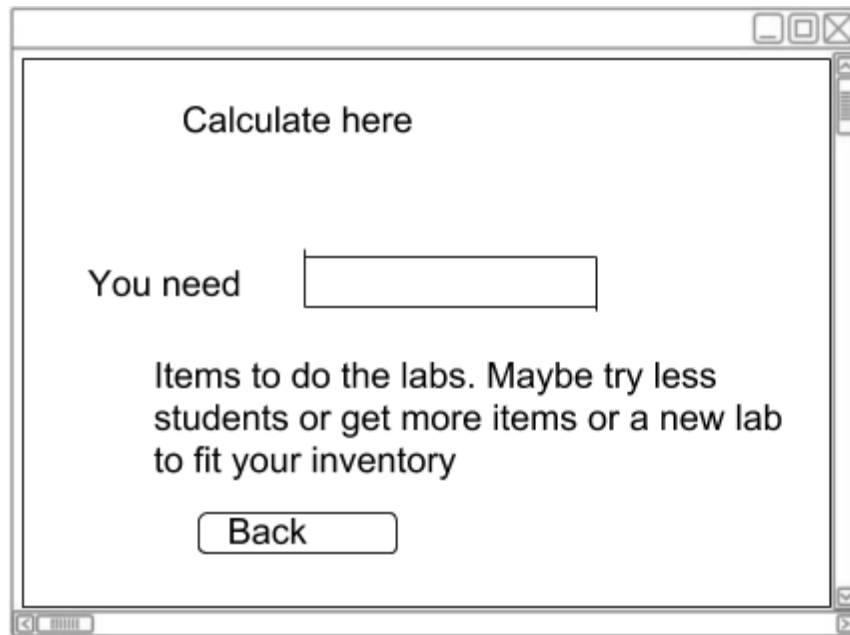


Figure 16 - Frame when user doesn't have enough items to do the lab

Test Plan

Action to be tested	Test Method
Program able to view inventory	Manually input a large number of items into the inventory. Make sure they are distinguishable with different amount numbers. Then see if they are viewable when the viewInventory function is used.
Program able to view labs	Manually input a large number of labs into the inventory. Make sure they are distinguishable with different amount numbers of items. Then see if they are viewable when the Labs class is functionable when used.
Program able to create own labs	Use the tools to create personalized labs and then view them using the Labs class to see if they are displayed in the table.
Program is able to edit the labs created	Use edit function to change lab name, items required, and amount per item required to test whether the changes are instituted.
Calculate if the inventory supplies suffices the lab requirements	Try both a lab with the sufficient inventory, one with a surplus of inventory, and one with too little inventory to see the different reactions from the program with the different inventories. The sufficient inventory should just let the user know that the lab <i>can</i> be performed, the surplus would let the user know the <i>amount</i> of extra inventory supplies, and the shortage of inventory supplies scenario would let the user know <i>how</i> many more supplies the user needs to meet the lab requirements.
Program is able to help the user when needed	Manually test out the buttons for help and see if the JLabel detailing how to use the program at each step is present
Program be able to update, delete inventory	Try changing the name or amount of items from the inventory class to see whether the changes have been registered.