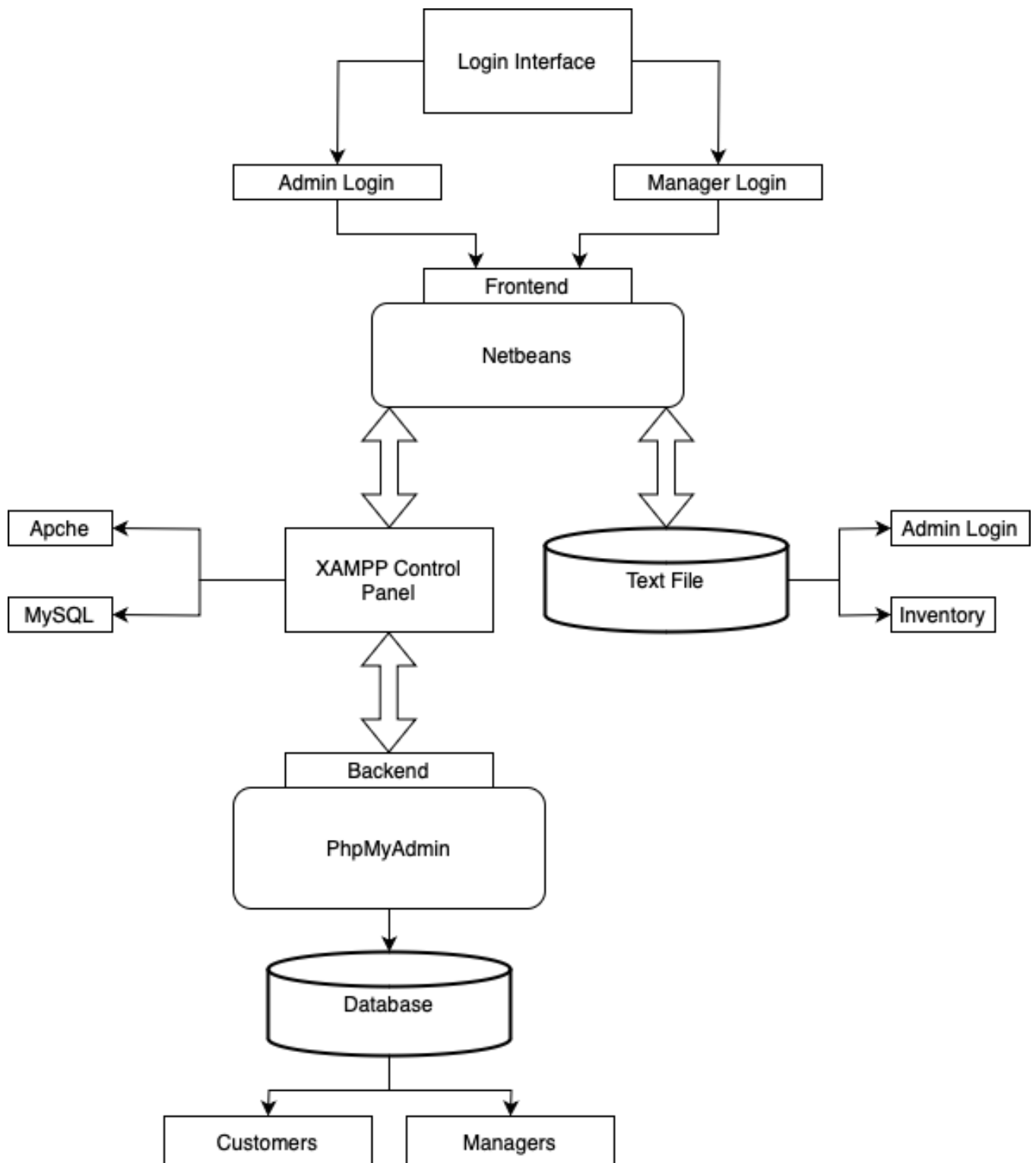


Criterion B

Table of Contents

- 1. Data Flow..... 2
- 1. Prototype Design with Algorithms 3
- 2. Database Structures 33
- 3. Application Sequencing..... 33
- 4. ERD Diagram 36
- 5. UML Diagram..... 37
- 6. Class Design 38
- 7. Test Plan 41
- 8. Updated design Idea after client Feedback 43

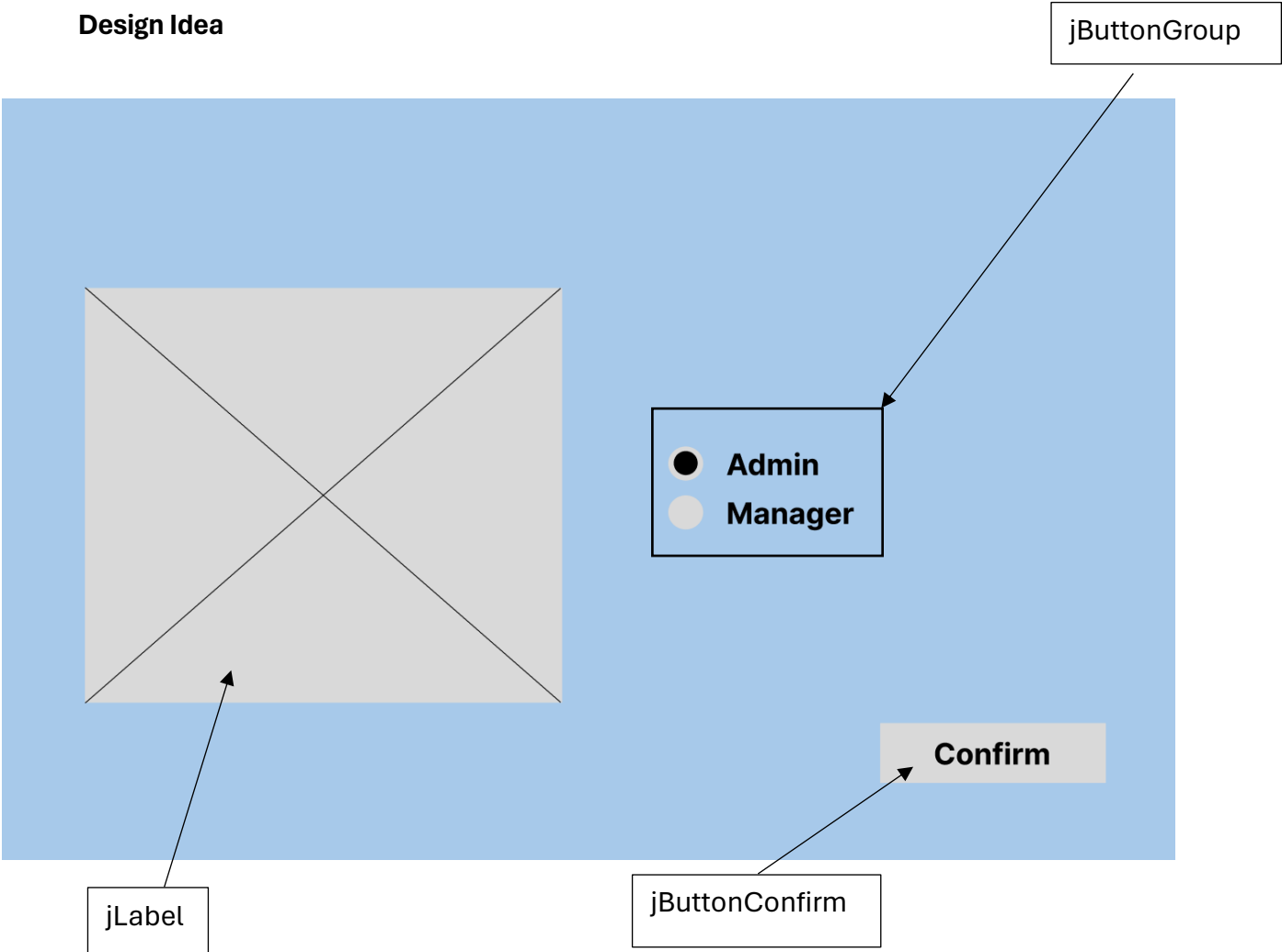
1. Data Flow



1. Prototype Design with Algorithms

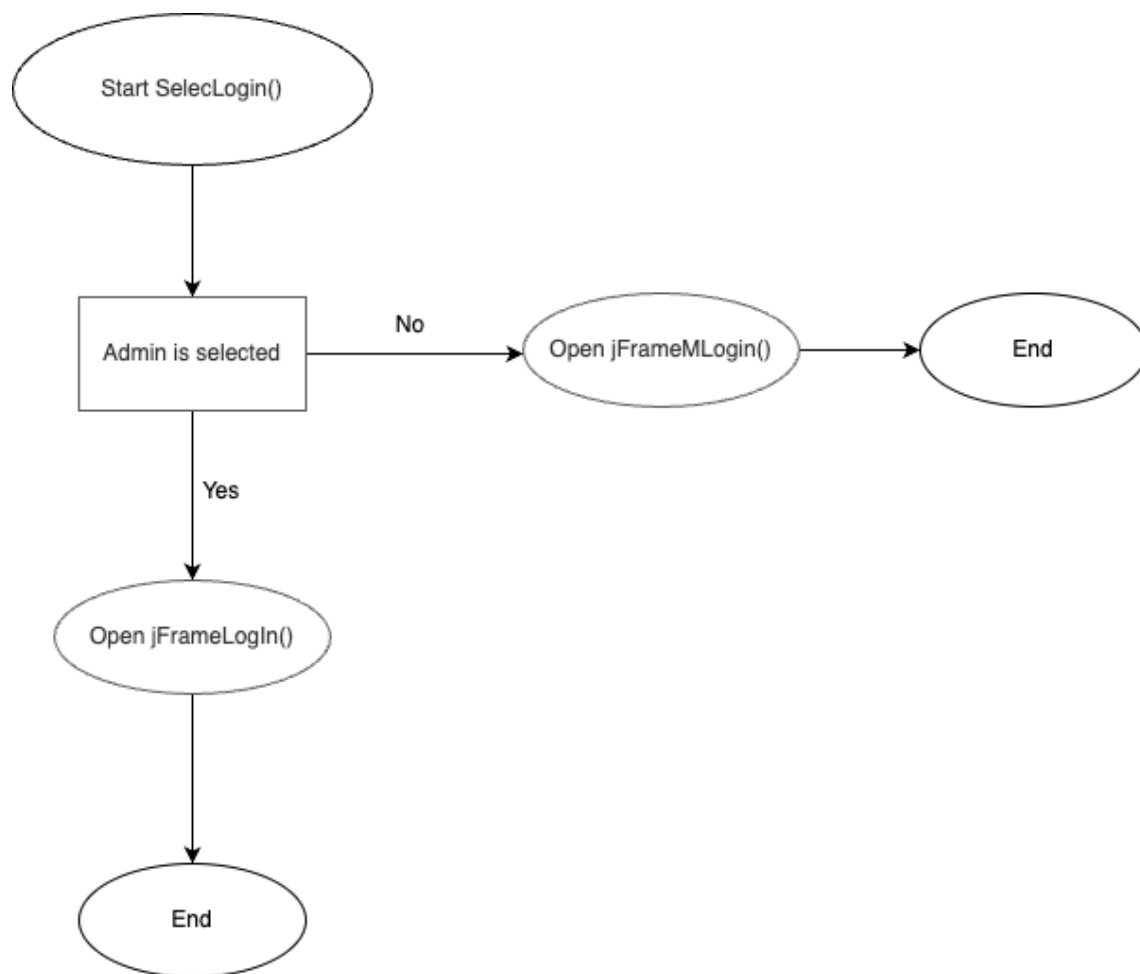
Screen 1: Selection Login

Design Idea



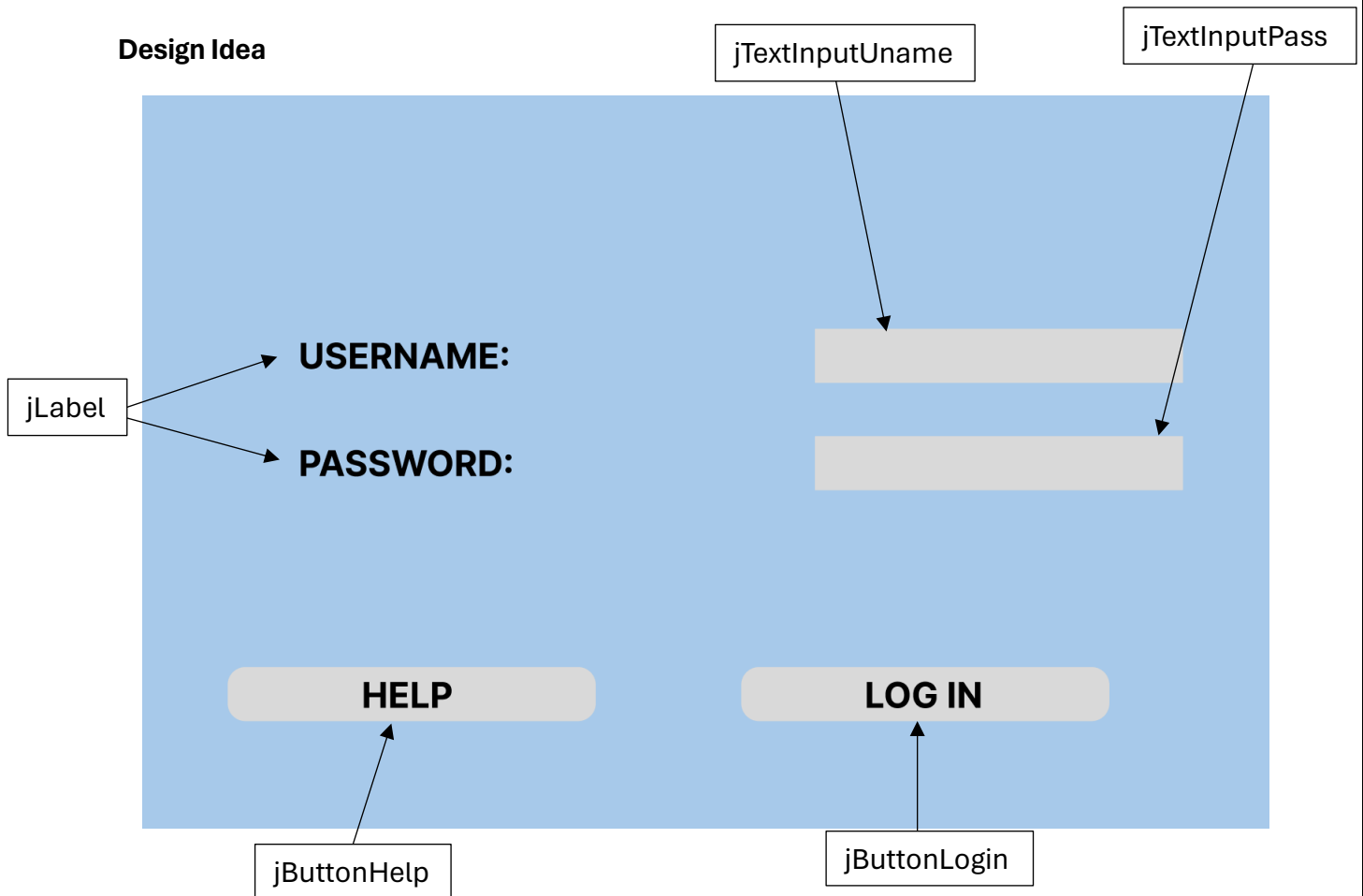
Component Name	Description	Data Access	Data Validation
jButtonConfirm	It directs the user to the selected login.	No data is accessed	No data validation is required
jButtonGroup	Allows the user to select their designation	No data is accessed	No data validation is required
jLabel	It will display an image.	No data is accessed	No data validation is required

Flow Chart

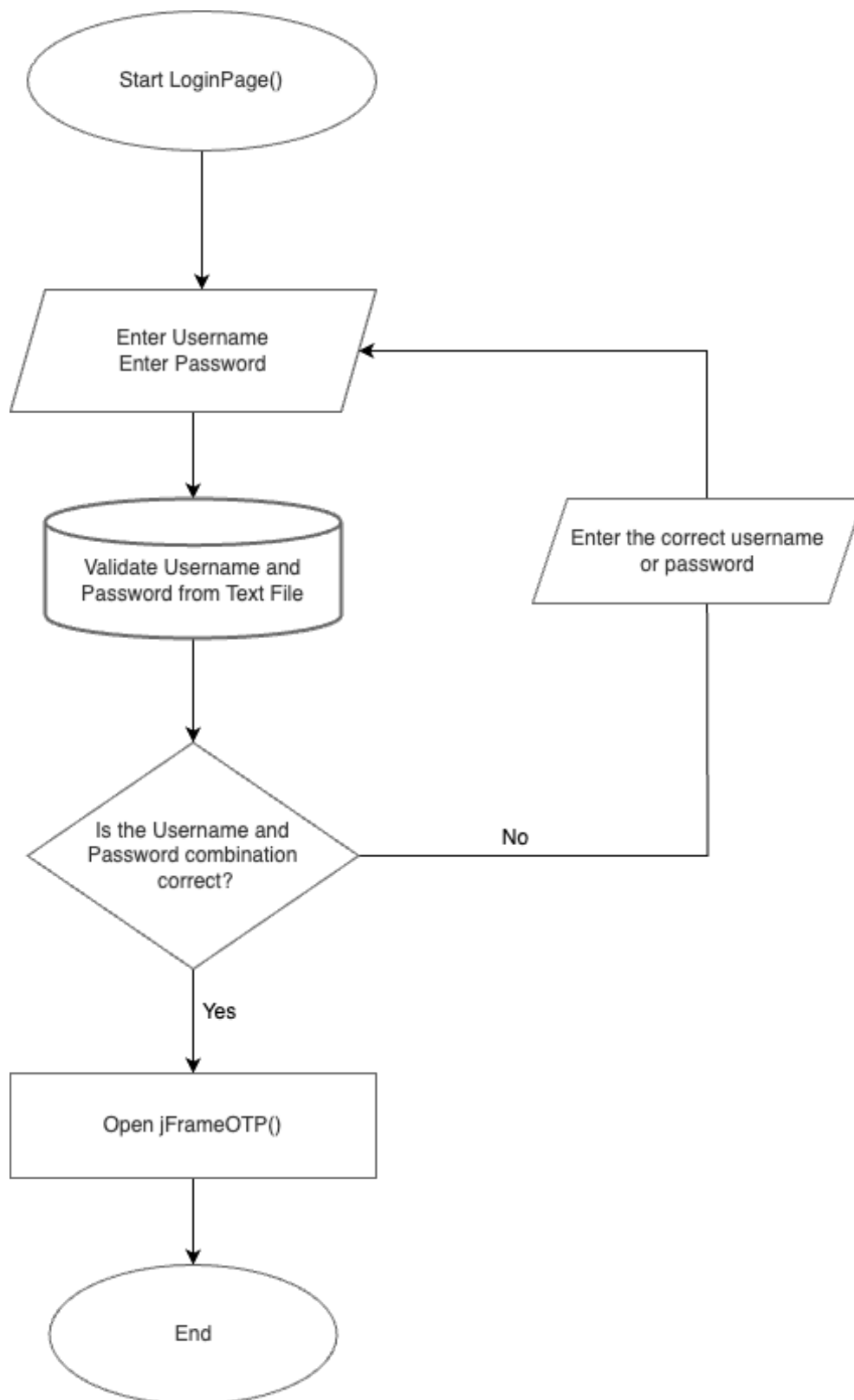


Screen 2: Admin Login

Design Idea

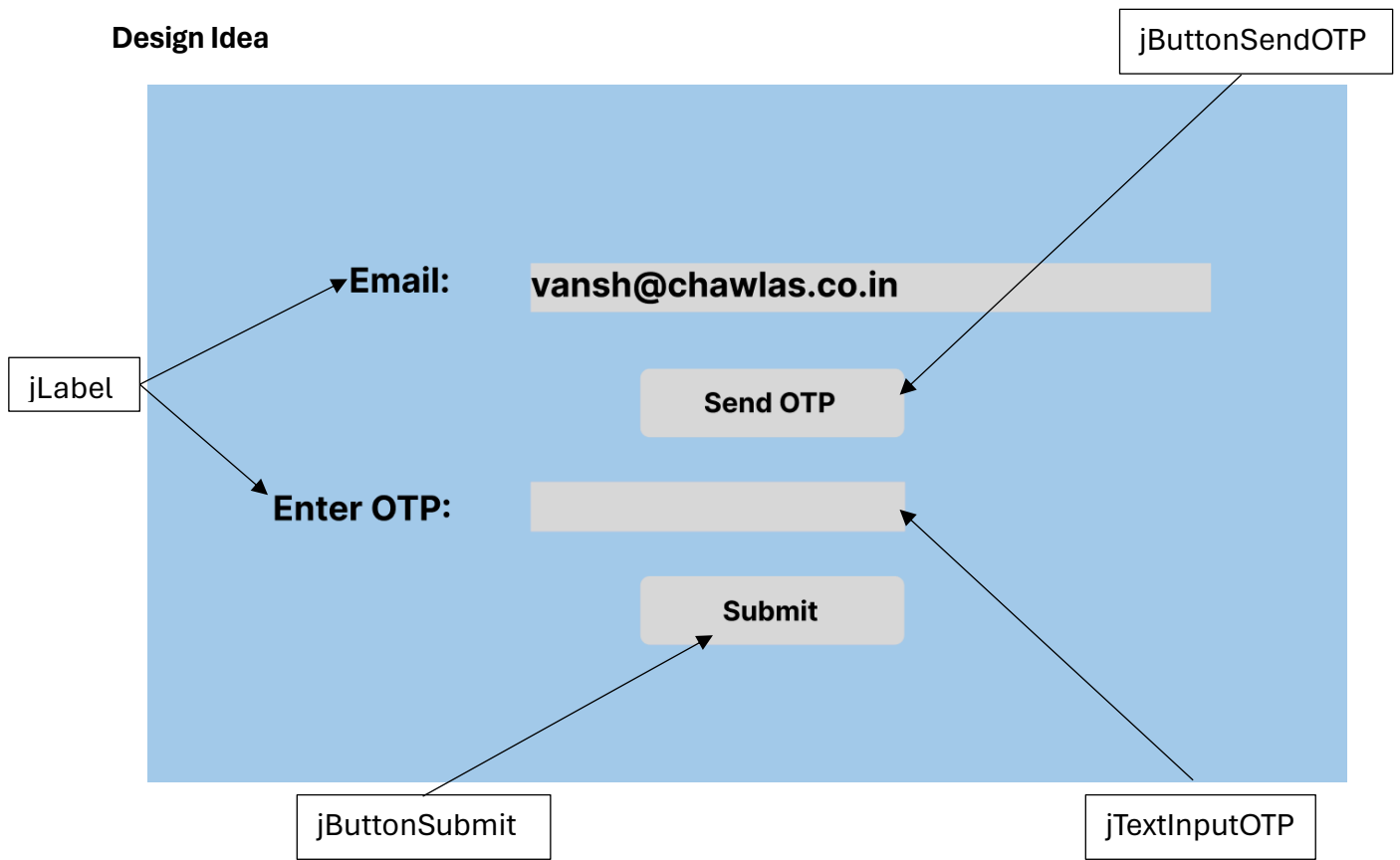


Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help pdf file	The text File help is accessed	No data validation is required
jButtonLogin	Directs the Admin into 2 Factor Authentication JFrame	The textfile UserNames.txt is accessed.	TextFieldUserName, textfield is checked for an empty string. The length of the string needs to be more than 2 characters. TextFieldPassword is checked for the string, it cannot be null.
jTextInputUname	Get the admin's username	No data is accessed	Check if Input is empty
jTextInputPass	Get the admin's password	No data is accessed	Check if Input is empty
jLabel	Display Text	No data is accessed	No data validation is required

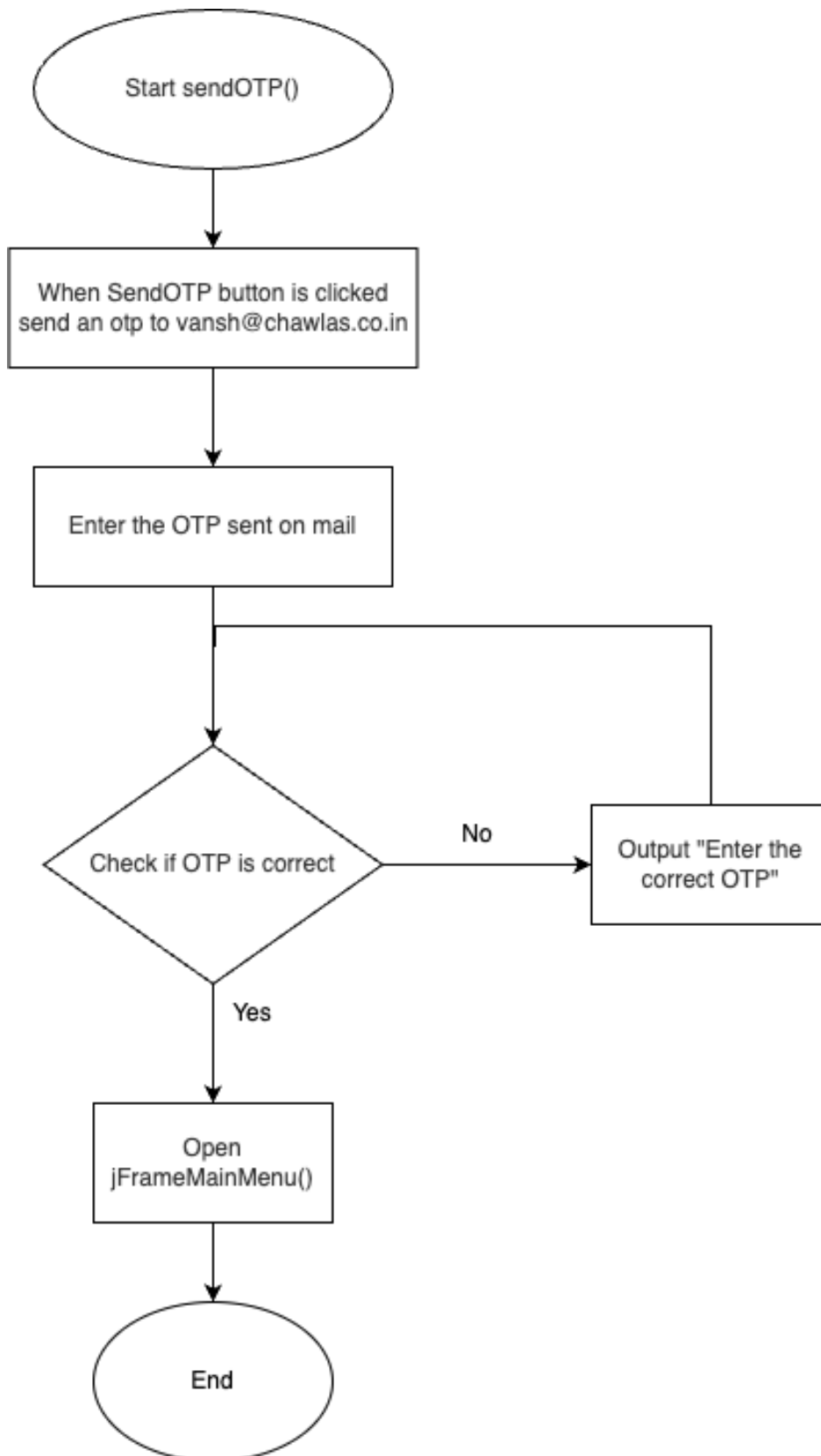
Flow Chart

Screen 3: OTP Login

Design Idea

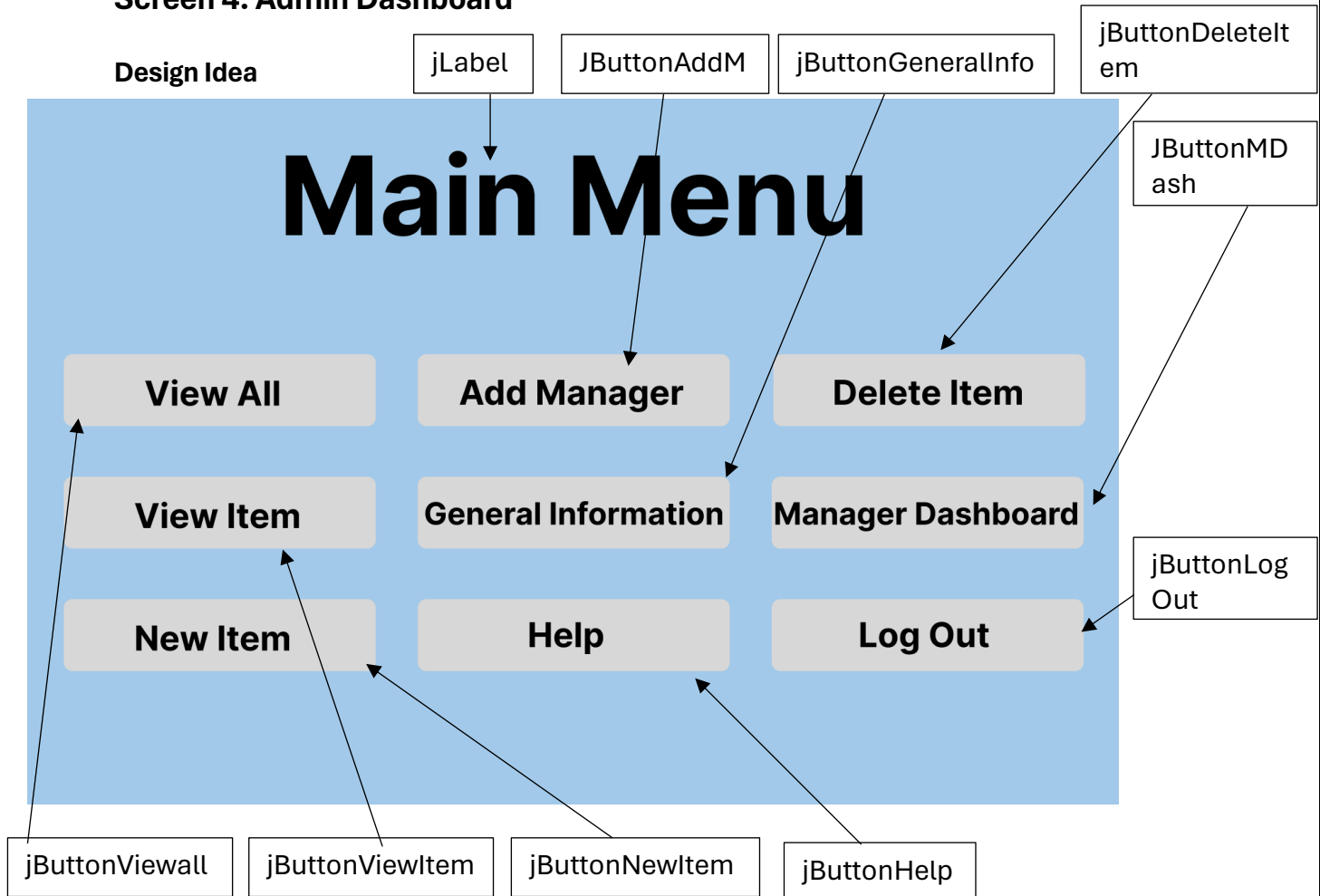


Component Name	Description	Data Access	Data Validation
jButtonSendOTP	Send's OTP to the mail address	No data is accessed	No data validation is required
jTextInputOTP	Takes user to viewitem JFrame	No data is accessed	OTP is validated with the one sent on the users mail
jButtonSubmit	Takes user to MainMenu JFrame if OTP is correct	No data is accessed	No data validation is required
jLabel	Display Text	No data is accessed	No data validation is required

Flow Chart

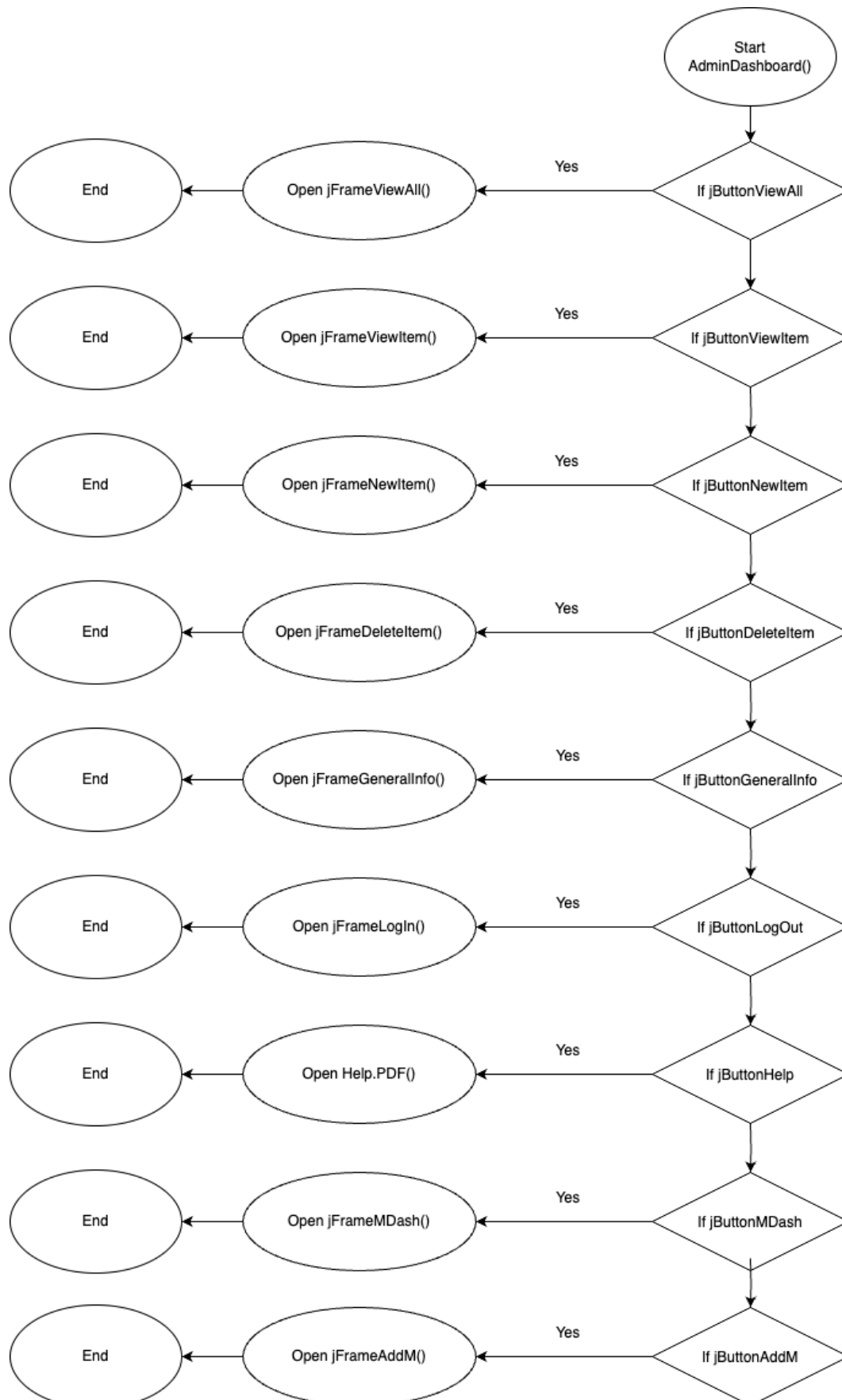
Screen 4: Admin Dashboard

Design Idea



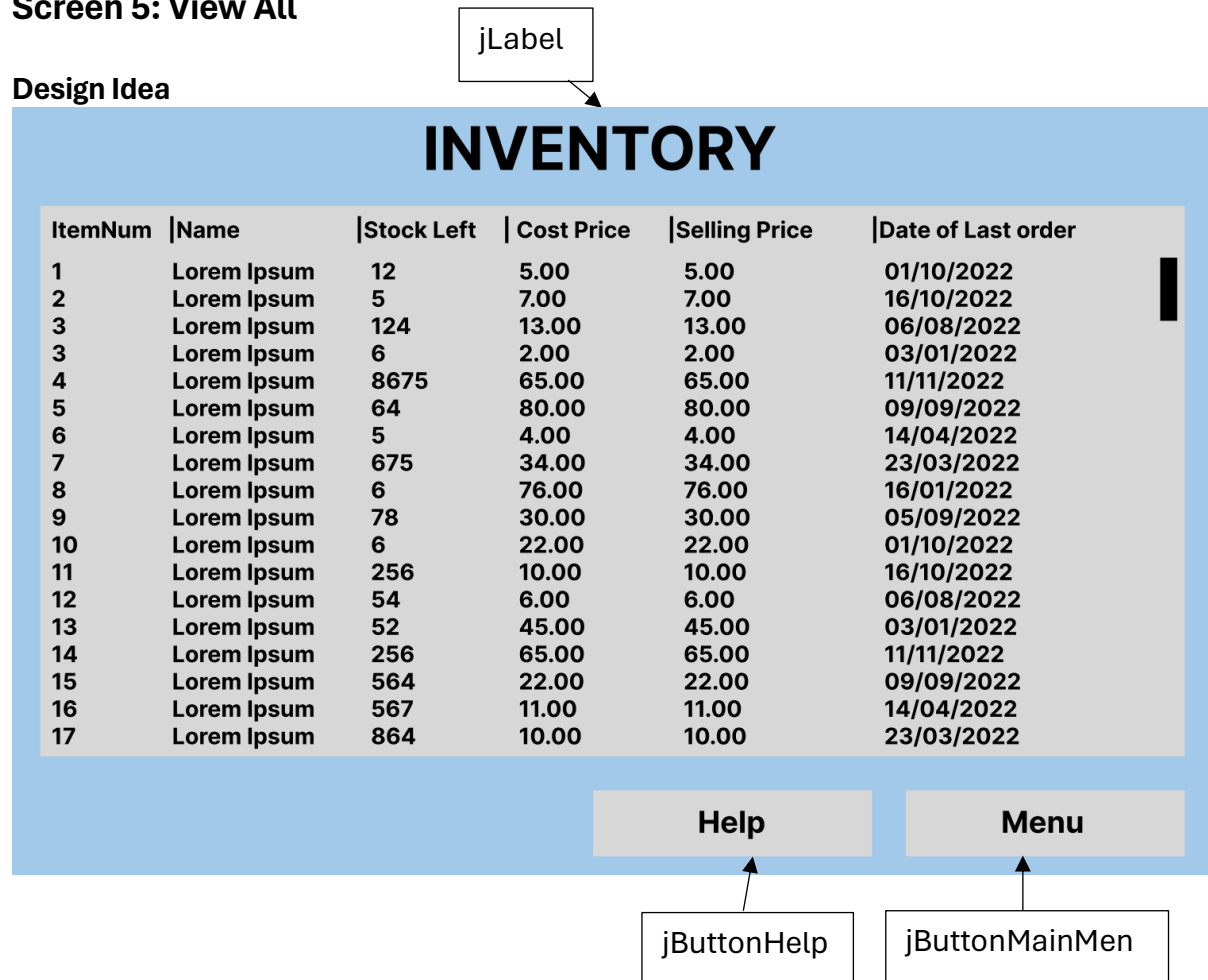
Component Name	Description	Data Access	Data Validation
<code>jButtonViewAll</code>	Takes user to viewAll JFrame	No data is accessed	No data validation is required
<code>jButtonViewItem</code>	Takes user to viewitem JFrame	No data is accessed	No data validation is required
<code>jButtonNewItem</code>	Takes user to newitem JFrame	No data is accessed	No data validation is required
<code>JButtonDeleteItem</code>	Takes user to DeleteItem JFrame	No data is accessed	No data validation is required
<code>jButtonGeneralInfo</code>	Takes user to the GeneralInfo JFrame	No data is accessed	No data validation is required
<code>jButtonLogOut</code>	Takes user to LogIn JFrame	No data is accessed	No data validation is required
<code>jButtonHelp</code>	Opens the help pdf file.	The text File help is accessed	No data validation is required
<code>JButtonMDash</code>	Takes user to MDash JFrame	No data is accessed	No data validation is required
<code>JButtonAddM</code>	Takes user to AddM JFrame	No data is accessed	No data validation is required
<code>jLabel</code>	Display Text	No data is accessed	No data validation is required

Flow Chart

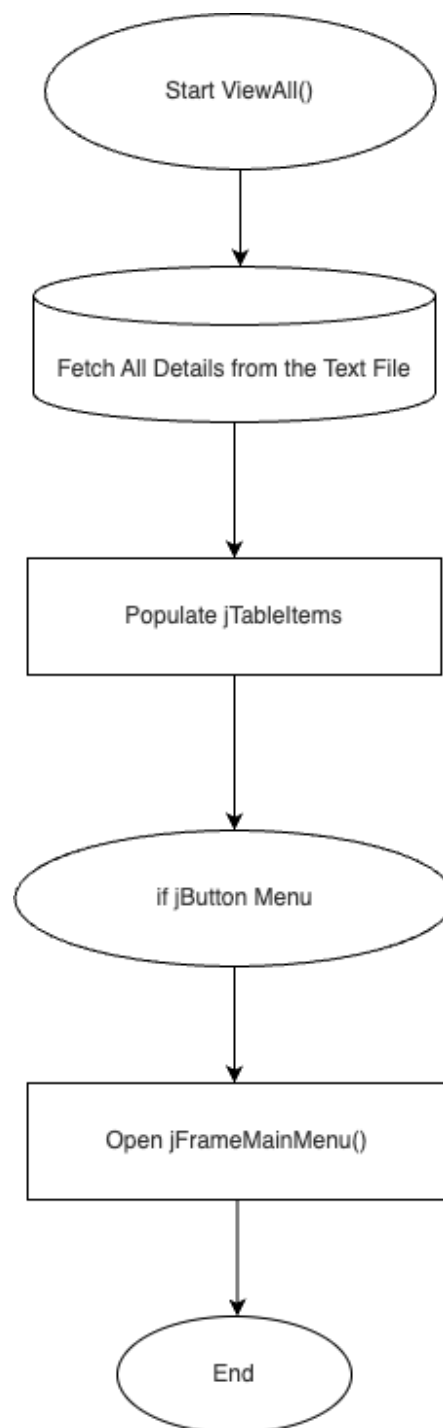


Screen 5: View All

Design Idea

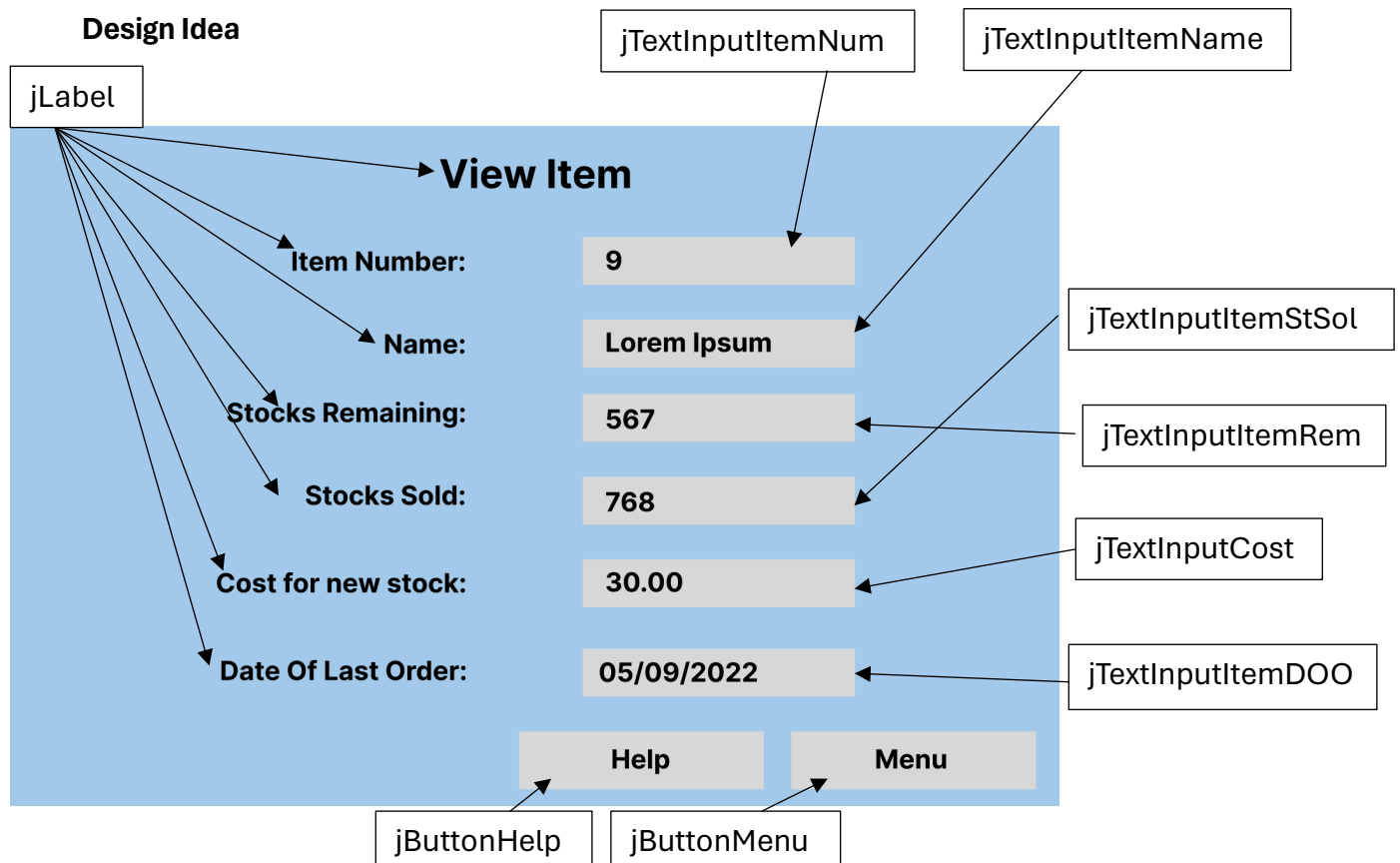


Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonMainMenu	Takes the user to the Main Menu	No data is accessed	No Data validation is required
jLabel	Display Text	No data is accessed	No data validation is required

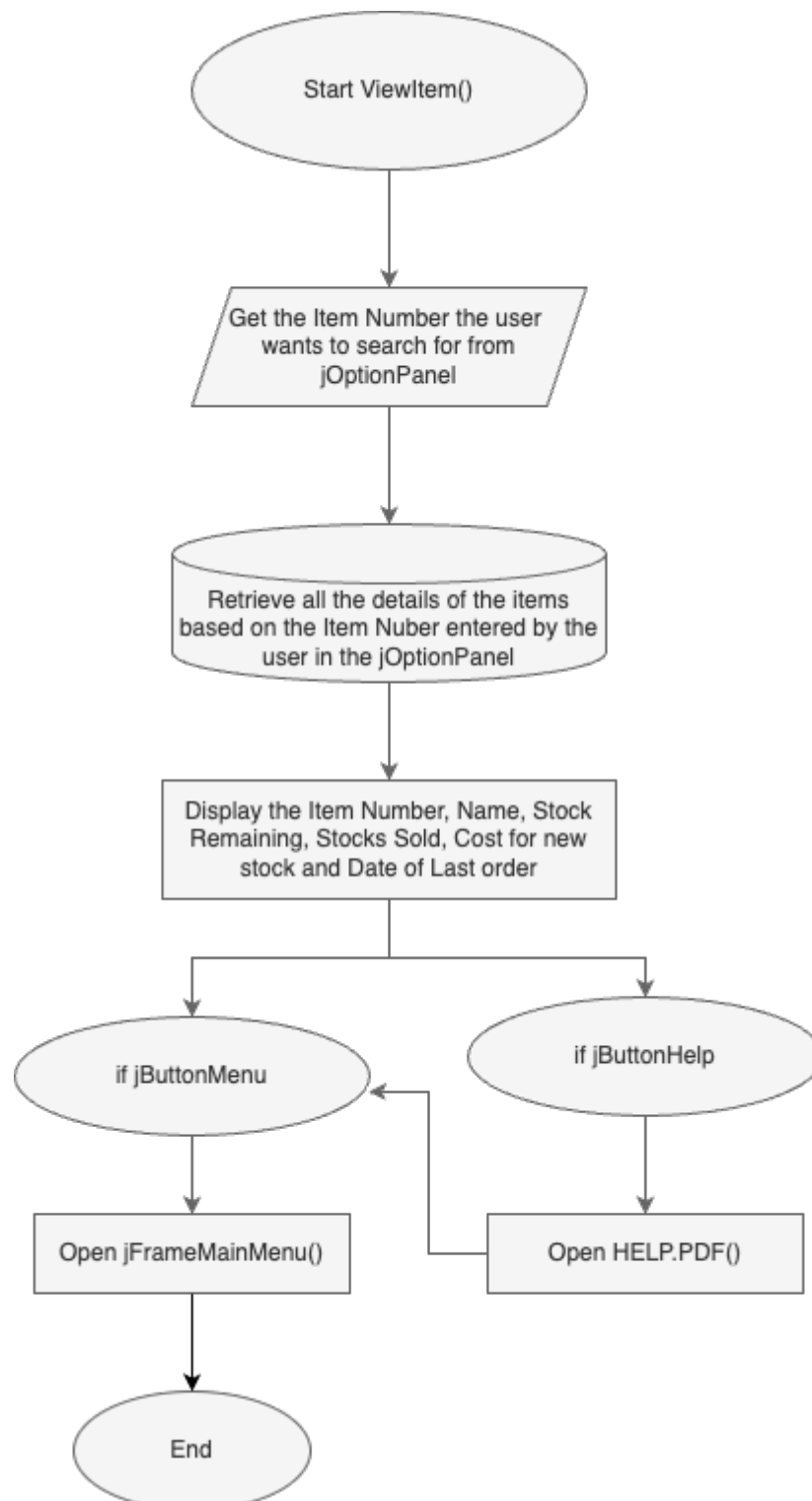
Flow Chart

Screen 6: View Item

Design Idea

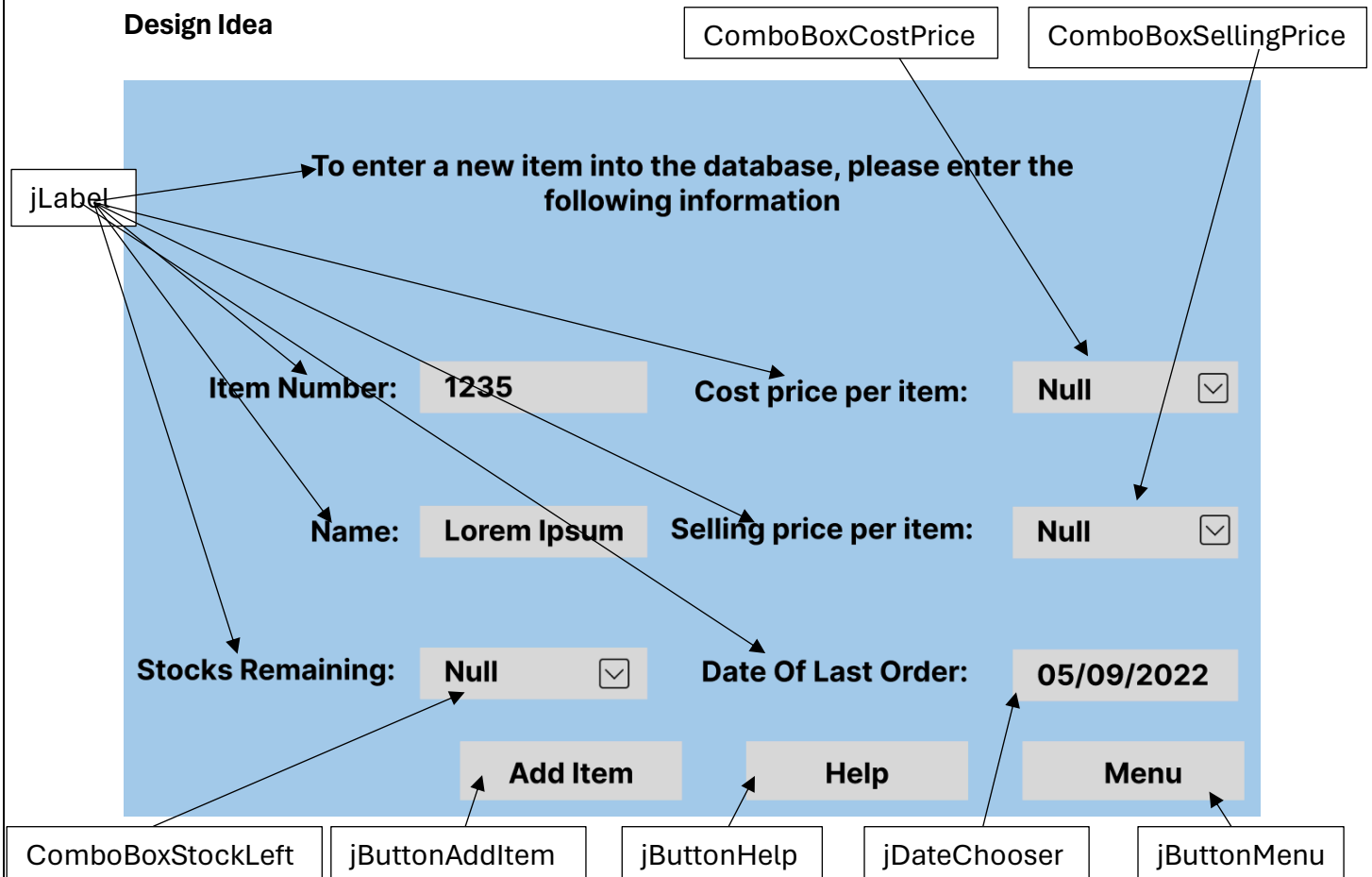


Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonMenu	Takes the user to the main menu jFrame	No data is accessed	No data validation is required
jTextInputItemNum	Generates Item Number	No data is accessed	No data validation is required
jTextInputItemName	Gets Item Name	No data is accessed	No data validation is required
jTextInputItemStSold	Gets Item Stock Sold	No data is accessed	No data validation is required
jTextInputItemRem	Gets Items Remaining	No data is accessed	No data validation is required
jTextInputCost	Gets Item Cost Price	No data is accessed	No data validation is required
jTextInputItemDOO	Gets Date of last order	No data is accessed	No data validation is required
Jlabel	Display Text	No data is accessed	No data validation is required

Flow Chart

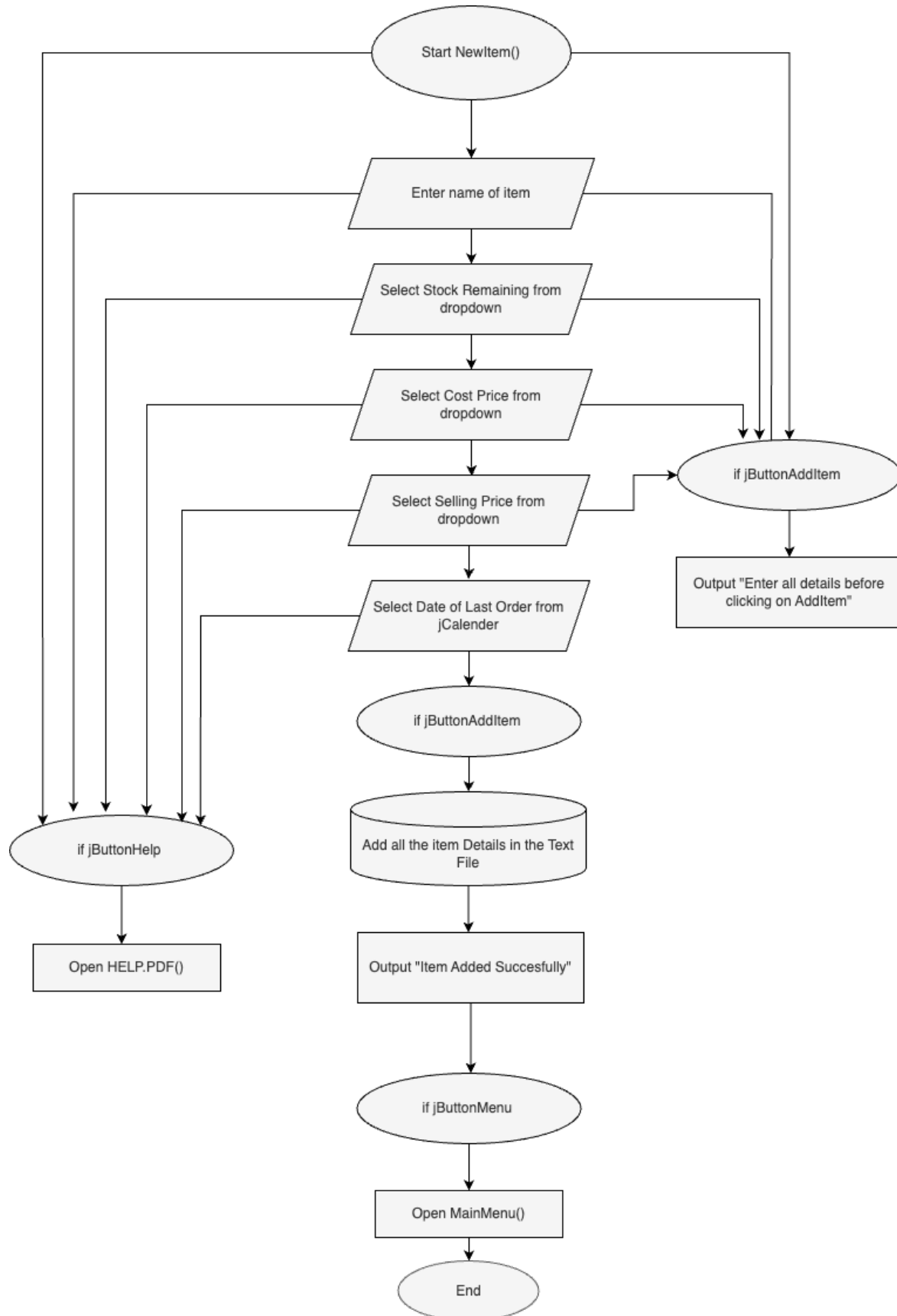
Screen 7: New Item

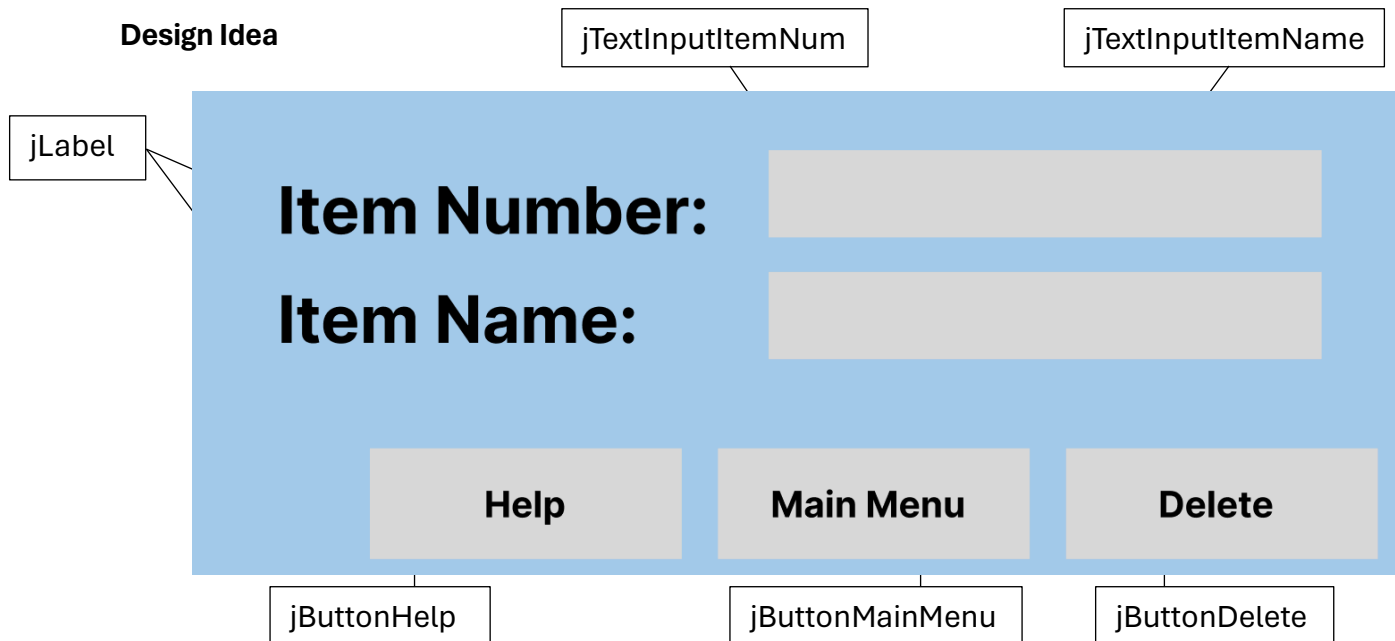
Design Idea



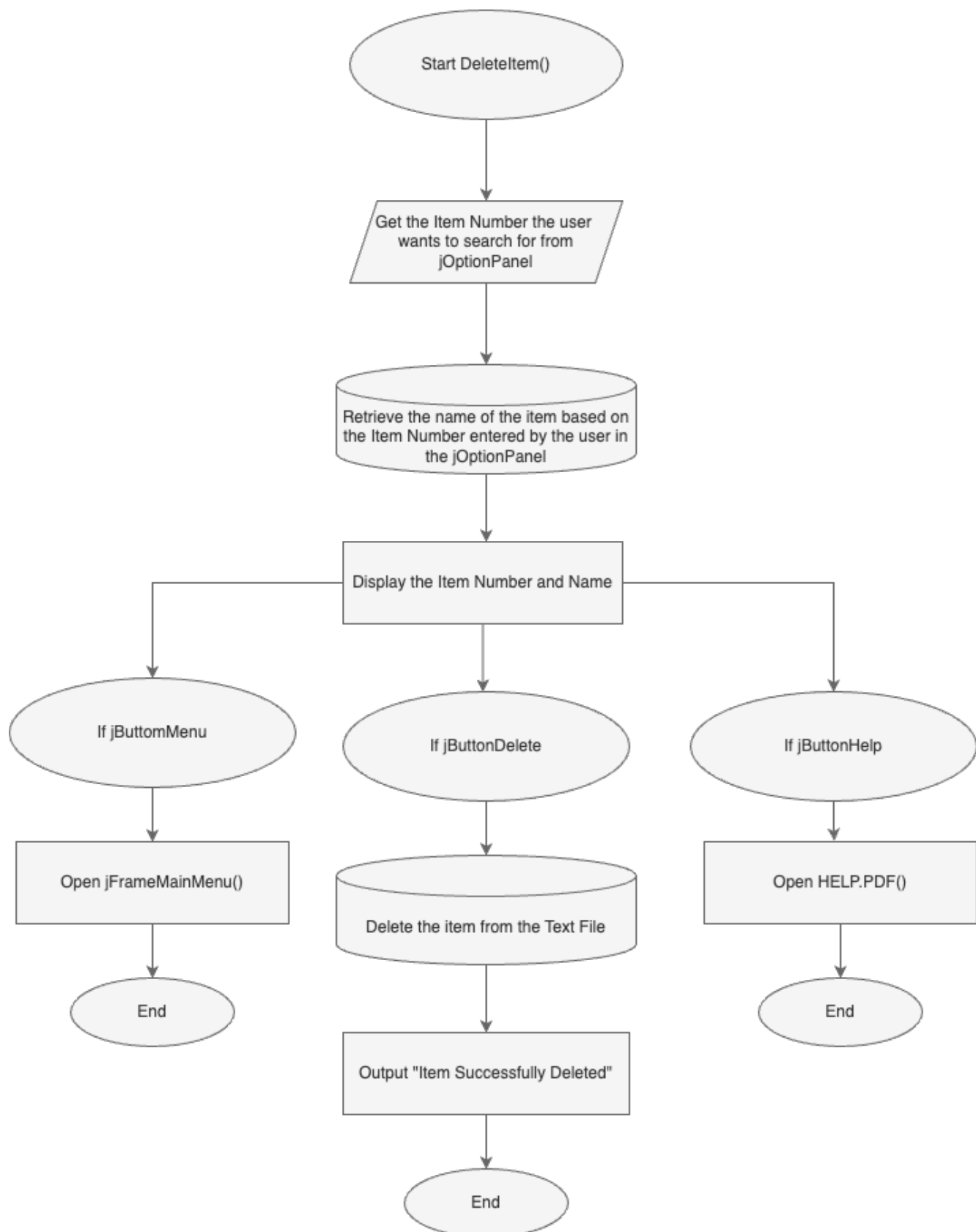
Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonAddItem	Takes the user to the main menu frame and adds the item to the text file	The text file Items is accessed. Data is stored into an object and entered into the array then the text file is edited and the item is added	TextFieldItemDescription, textfield is checked for an empty string. It is also checked that the length of the string is greater than 2
ComboBoxStockLeft			ComboBoxStockLeft is cheked for null
ComboBoxCostPrice			ComboBoxCostPrice is checked for null
ComboBoxSellingPrice			ComboBoxSellingPrice is checked for null
jDateChooser			jDateChooser is checked for an empty date
jButtonMenu	Takes user to Main Menu frame	No data is accessed	No data validation is required

Flow Chart



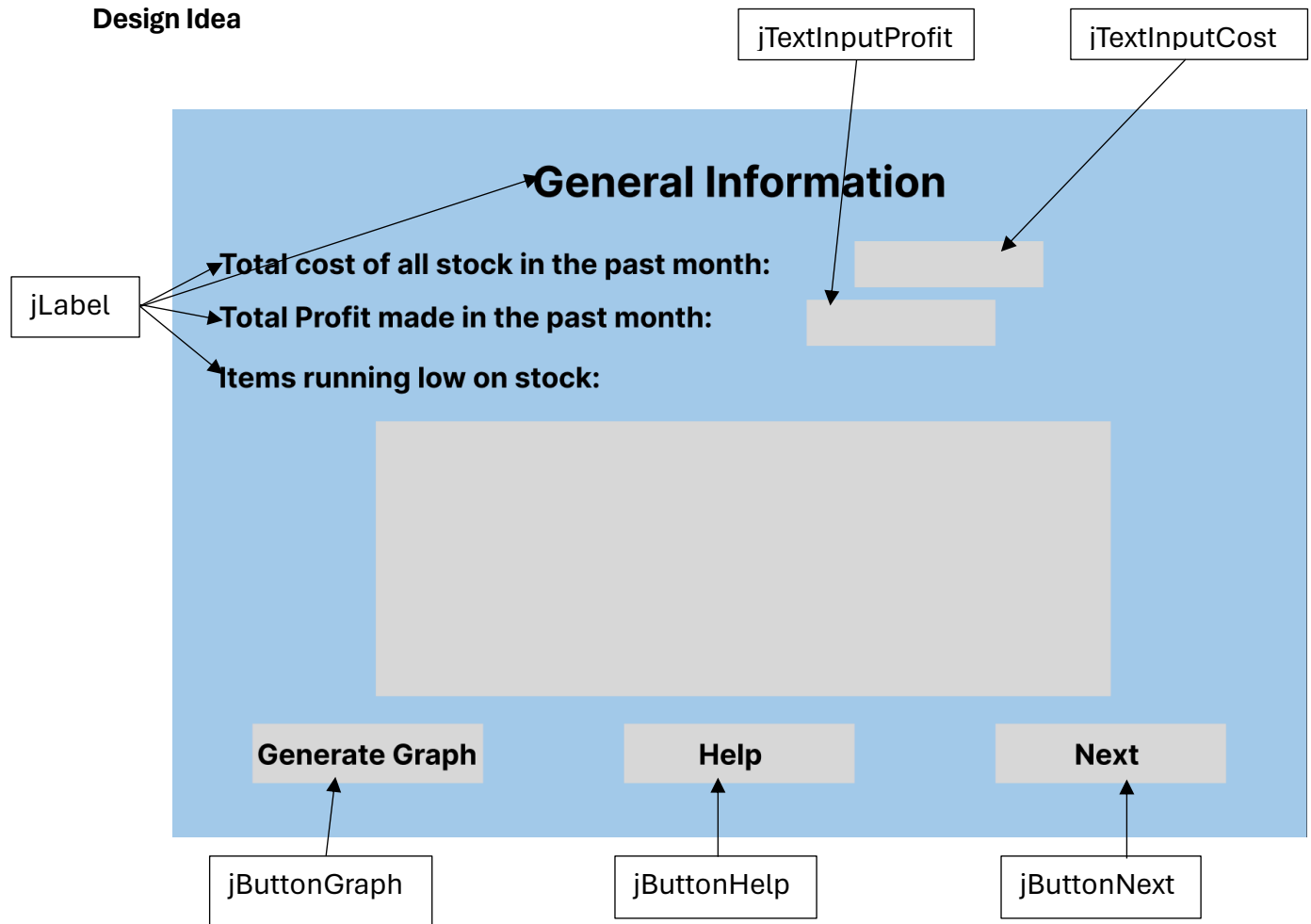
Screen 8: Delete Item**Design Idea**

Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonDelete	Deletes the item displayed form the array	The text file Items is accessed	No data validation is required
jButtonMainMenu	Opens the JFrame Main Menu	No data is accessed	No data validation is required
jTextInputItemNum	Gets input from user	No data is accessed	No data validation is required
jTextInputItemName	Shows the name of the item number	Text file is accessed	No data validation is required

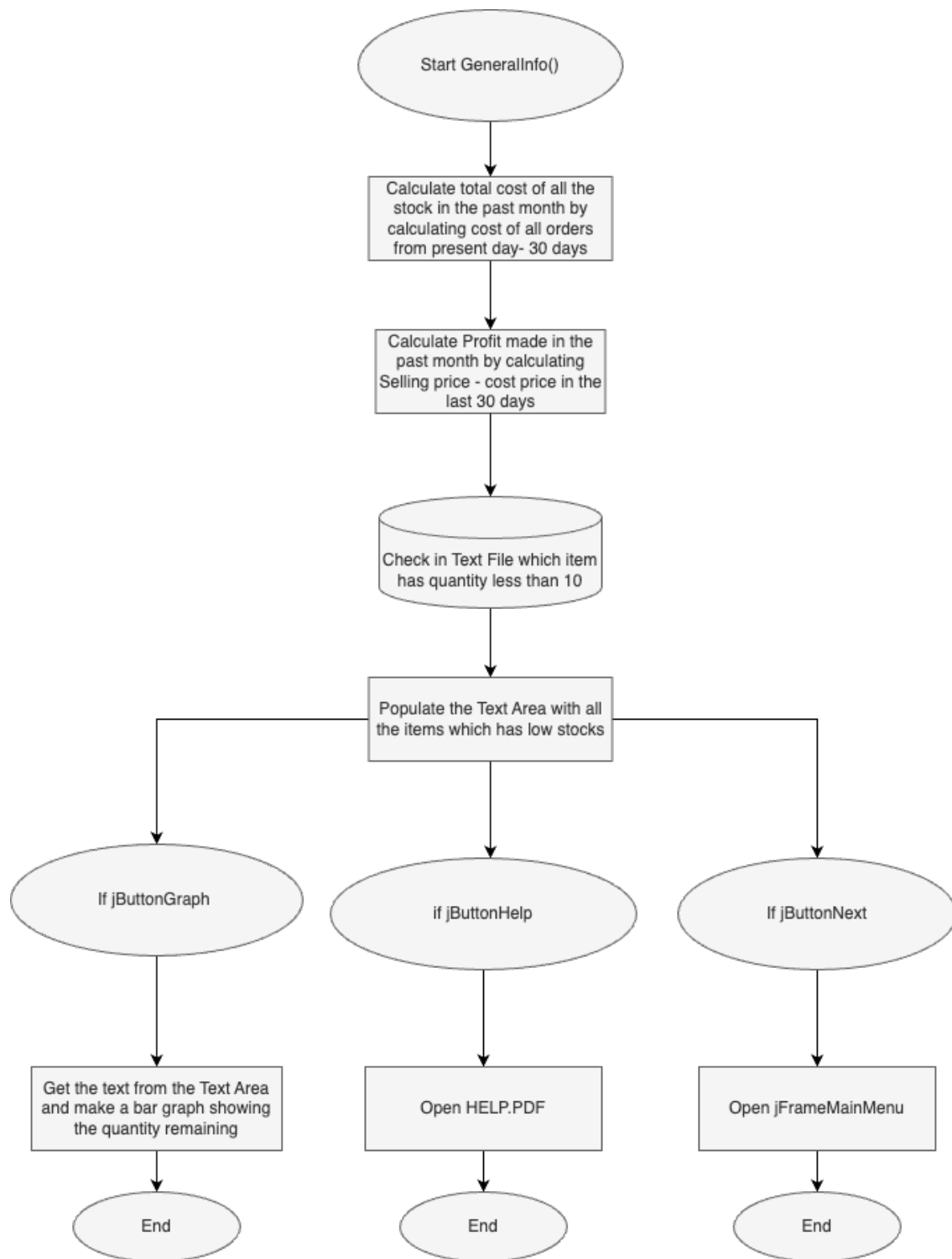
Flow Chart

Screen 9: General Information

Design Idea

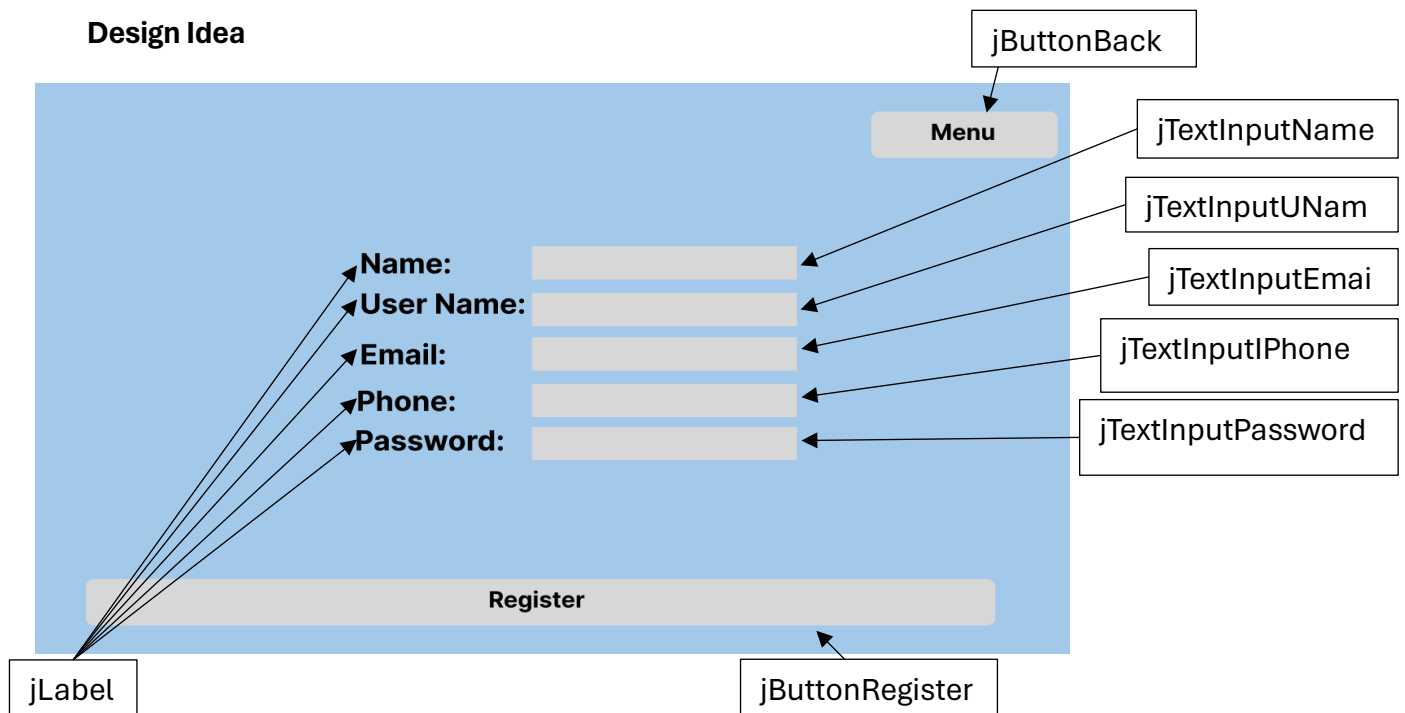


Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonNext	Takes the user to the main menu JFrame	No data is accessed	No data validation is required
jButtonGraph	Generates a graph of the low stock	The text file Items is accessed.	No data validation is required
jTextDisplay	Displays list of items which are low on stock.	No data is accessed	No data validation is required
jTextInputCost	Shows the total Cost	No data is accessed	No data validation is required
jTextInputProfit	Shows the total Profit	No data is accessed	No data validation is required
Jlabel	Display Text	No data is accessed	No data validation is required



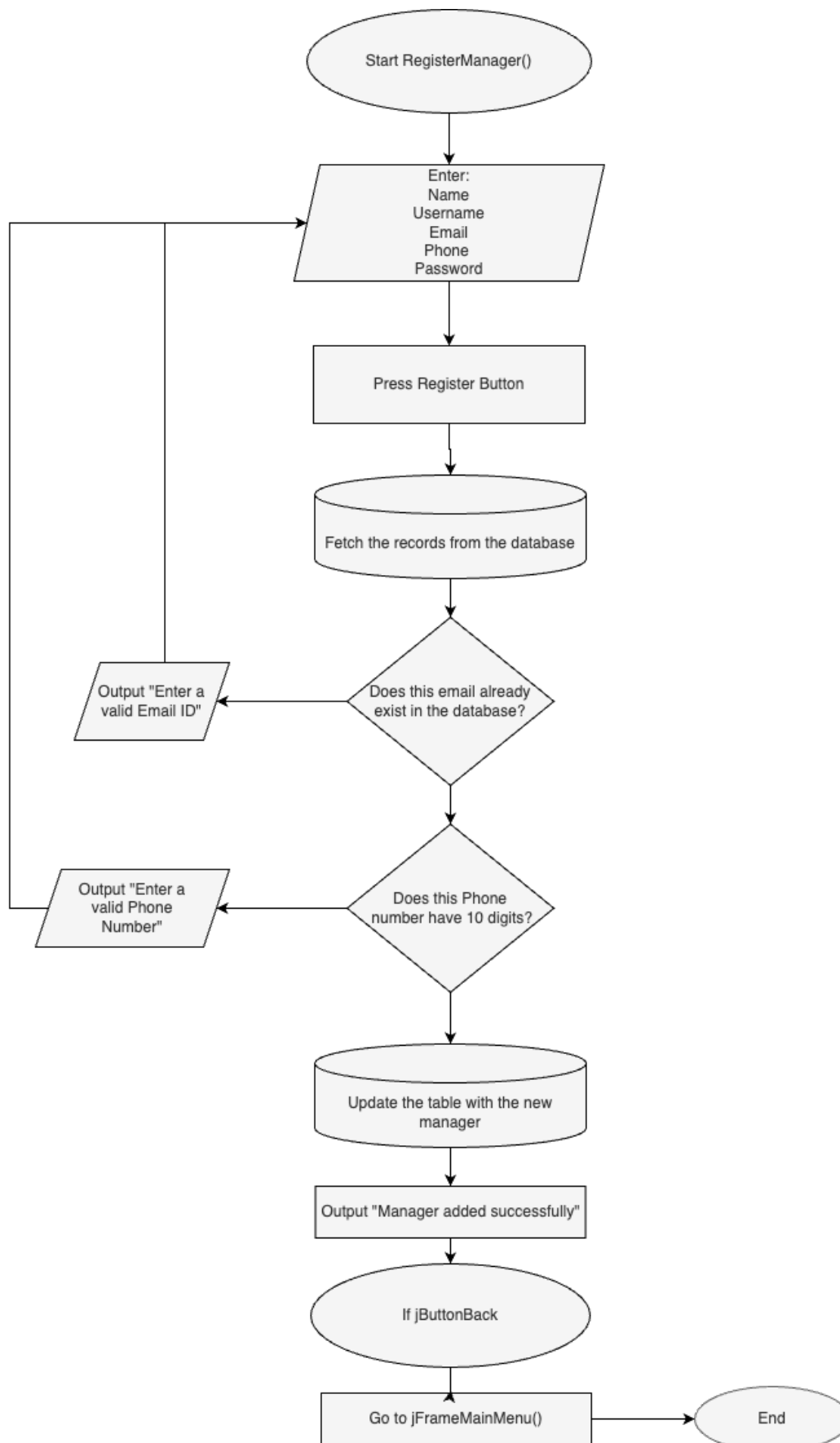
Screen 10: Manager Register

Design Idea



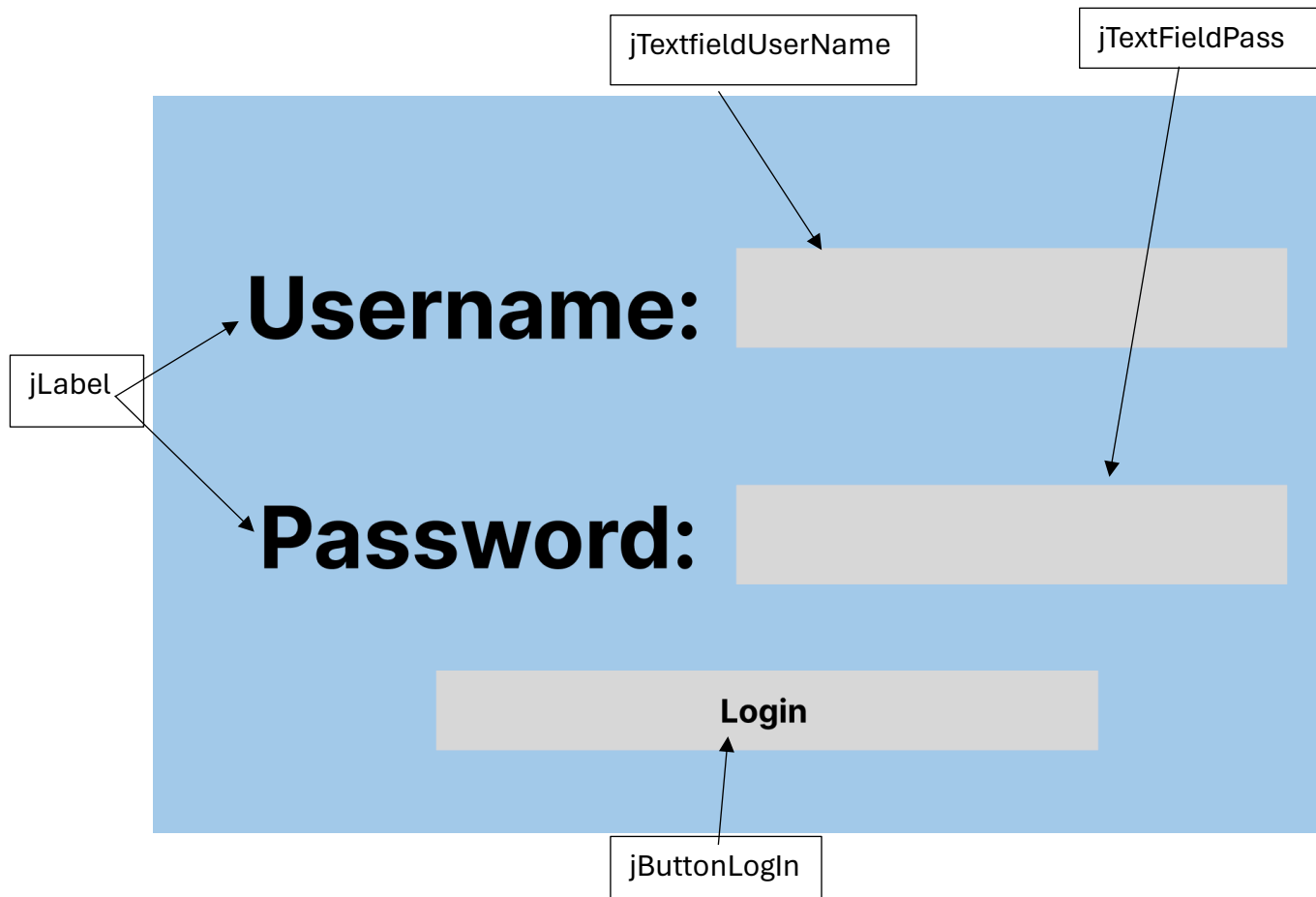
Component Name	Description	Data Access	Data Validation
jButtonBack	Takes the user back to the Main Menu	No data is accessed	No data validation is required
jButtonRegister	Data gets saved to the database	No data is accessed	No data validation is required
jTextInputName	Gets manager name	No data is accessed	No data validation is required
jTextInputUName	Gets manager user name	No data is accessed	No data validation is required
jTextInputEmail	Gets manager email	No data is accessed	No data validation is required
jTextInputIPhone	Gets manager phone number	No data is accessed	No data validation is required
jTextInputPassword	Gets manager password	No data is accessed	No data validation is required
jlabel	Display Text	No data is accessed	No data validation is required

Flow Chart

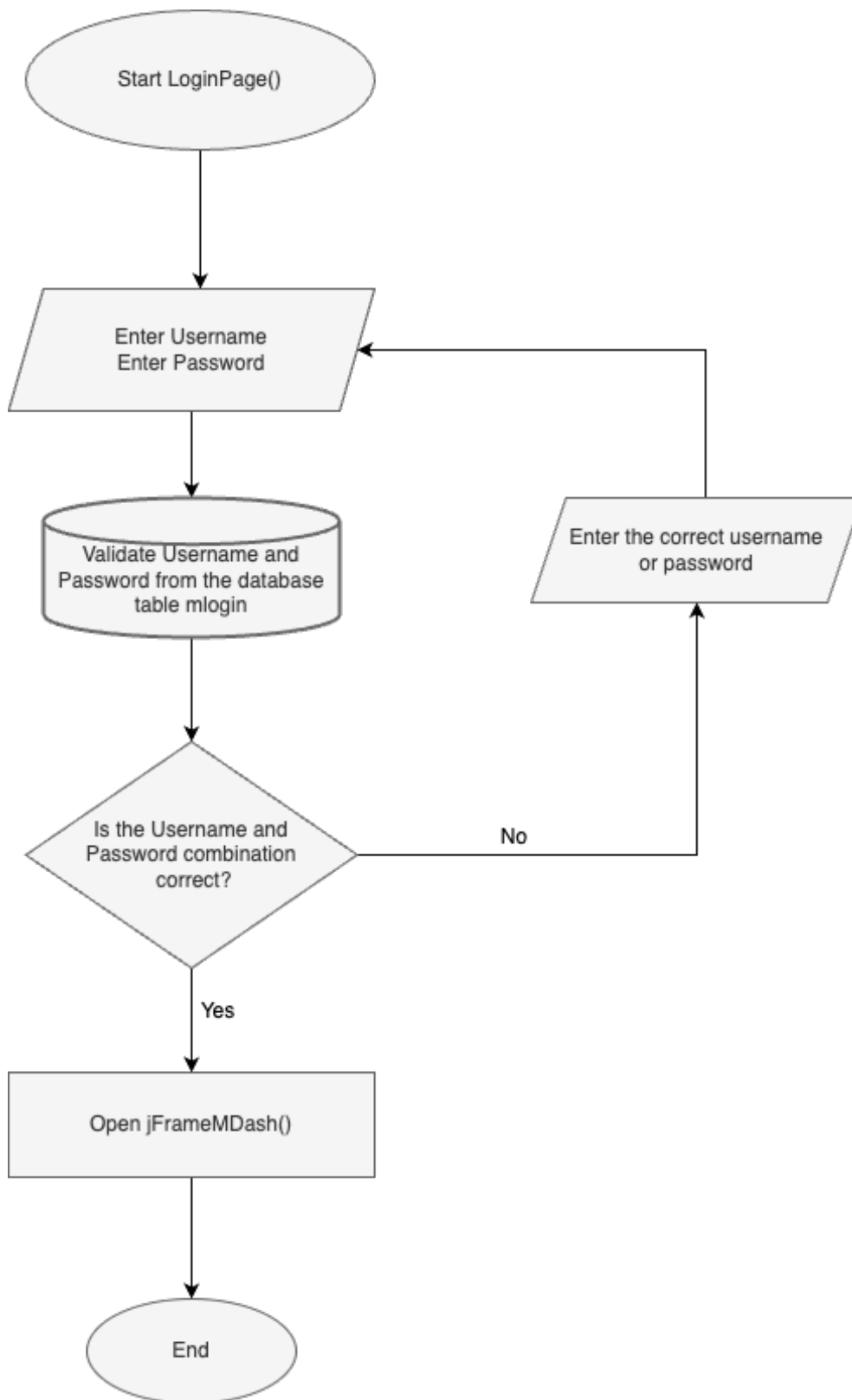


Screen 11: Manager Log In

Design Idea

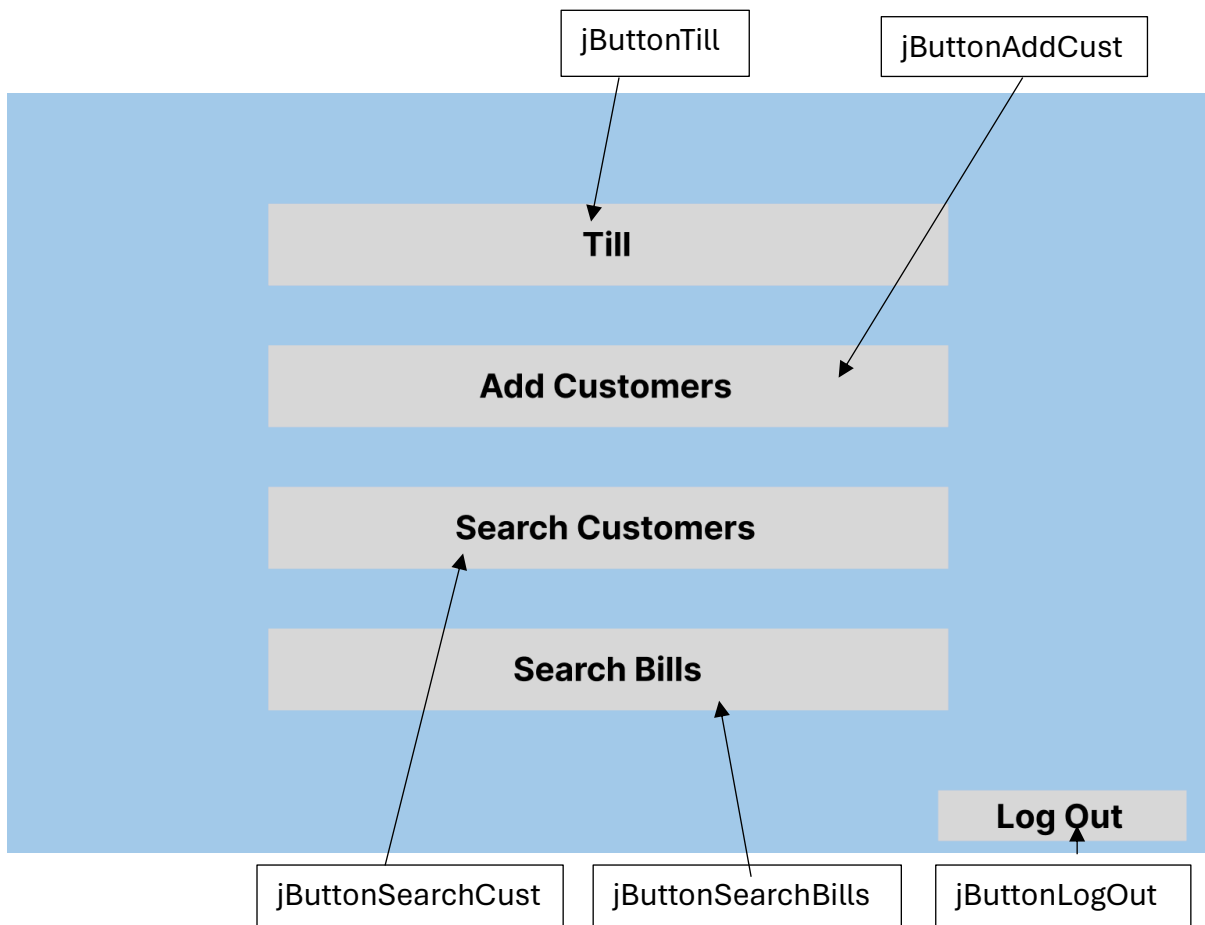


Component Name	Description	Data Access	Data Validation
jButtonLogin	Directs the manager into the manager dashboard	The Database with all the details of all managers	jTextFieldUserName, jTextFieldPass is checked for an empty string. The length of the string needs to be more than 2 characters. TextFieldPassword is checked for the string, it cannot be null.
jTextFieldUserName	Takes the username inputted by the manager	No data is accessed	No data validation is required
jTextFieldPass	Takes the password inputted by the manager	No data is accessed	No data validation is required
jlabel	Display Text	No data is accessed	No data validation is required

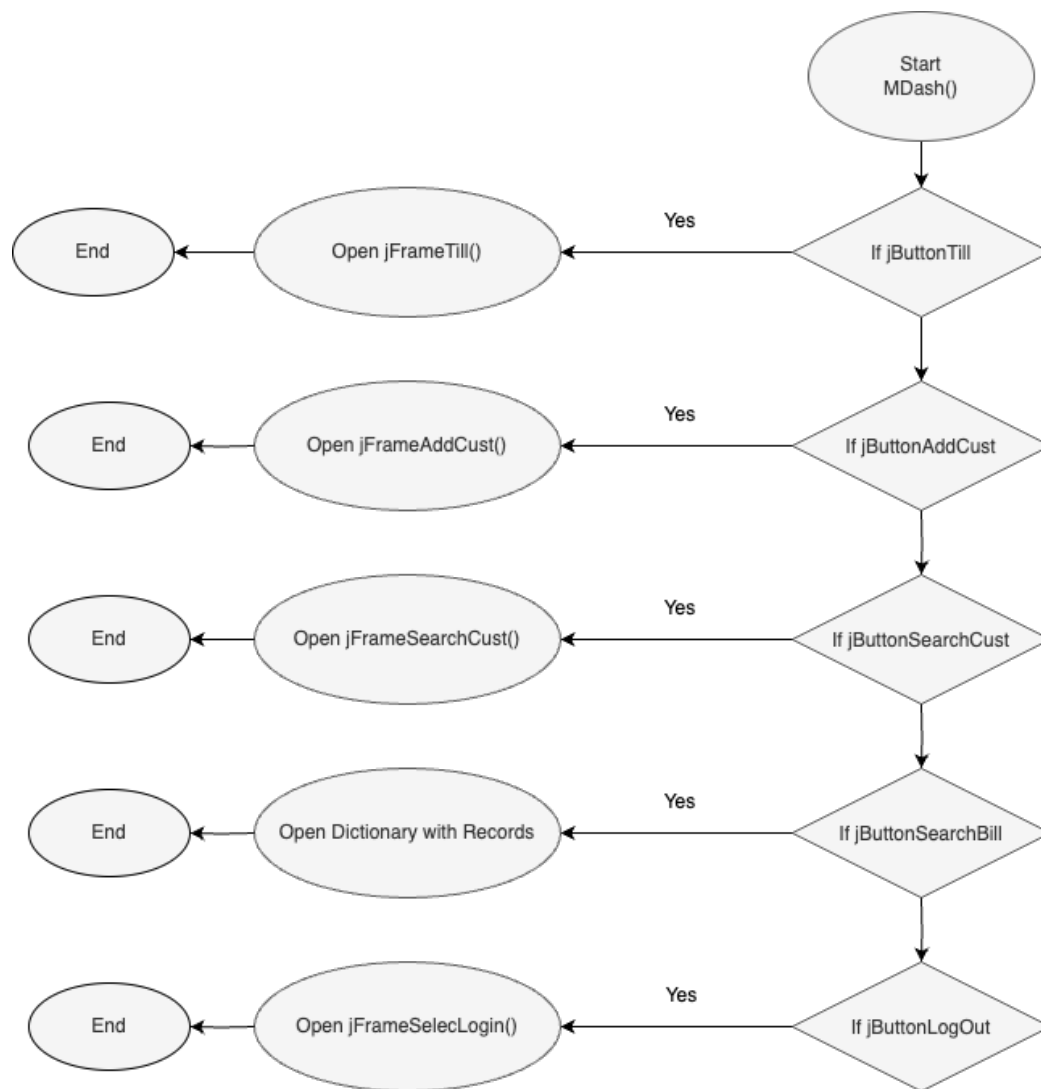
Flow Chart

Screen 12: Manager Dashboard

Design Idea

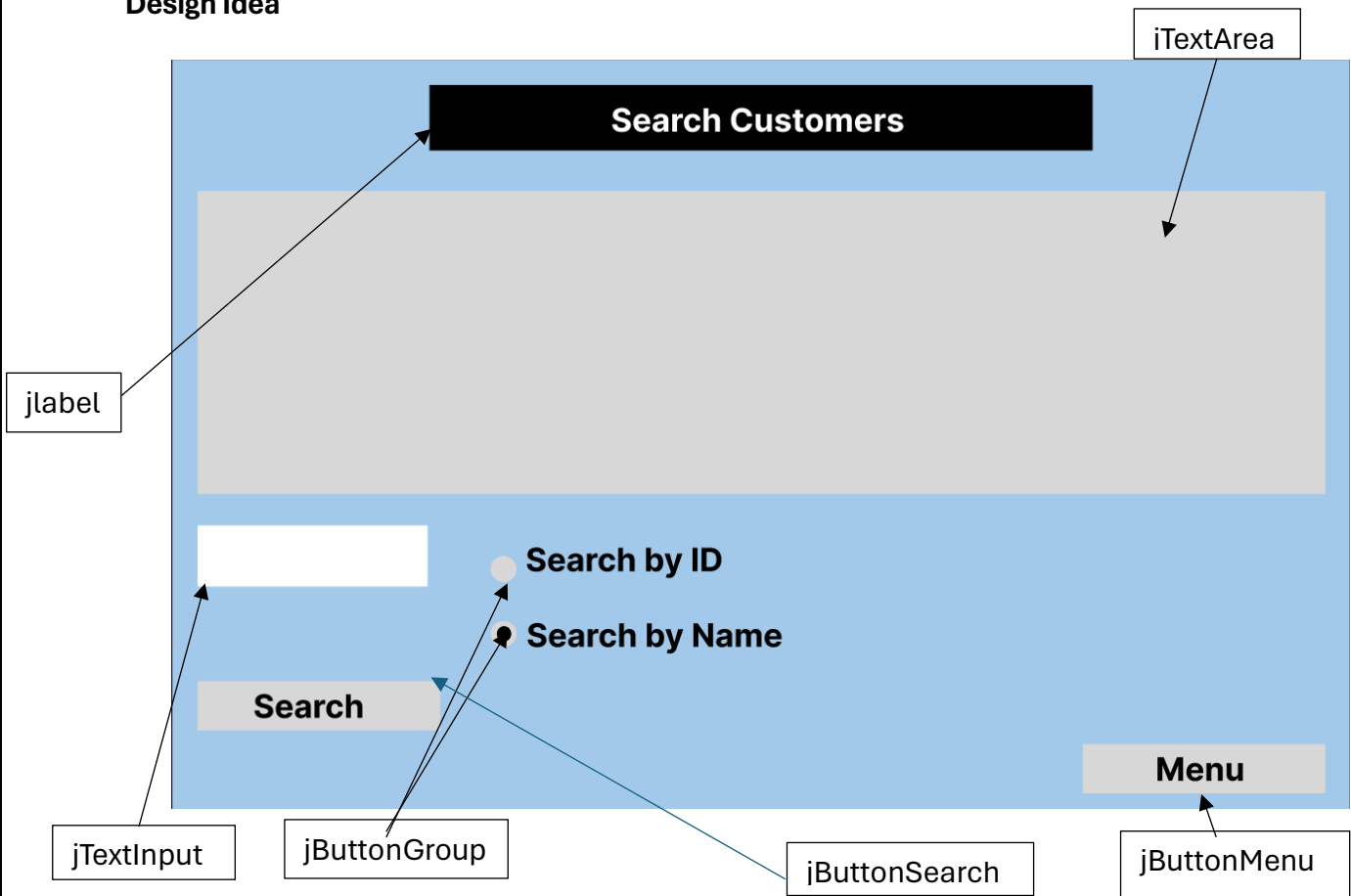


Component Name	Description	Data Access	Data Validation
jButtonTill	Takes user to Till JFrame	No data is accessed	No data validation is required
jButtonAddCust	Takes user to AddCust JFrame	No data is accessed	No data validation is required
jButtonSearchCust	Takes user to SearchCust JFrame	No data is accessed	No data validation is required
jButtonSearchBills	Takes user to Dictionary with all the bills JFrame	No data is accessed	No data validation is required
jButtonLogOut	Takes user to Login JFrame	No data is accessed	No data validation is required

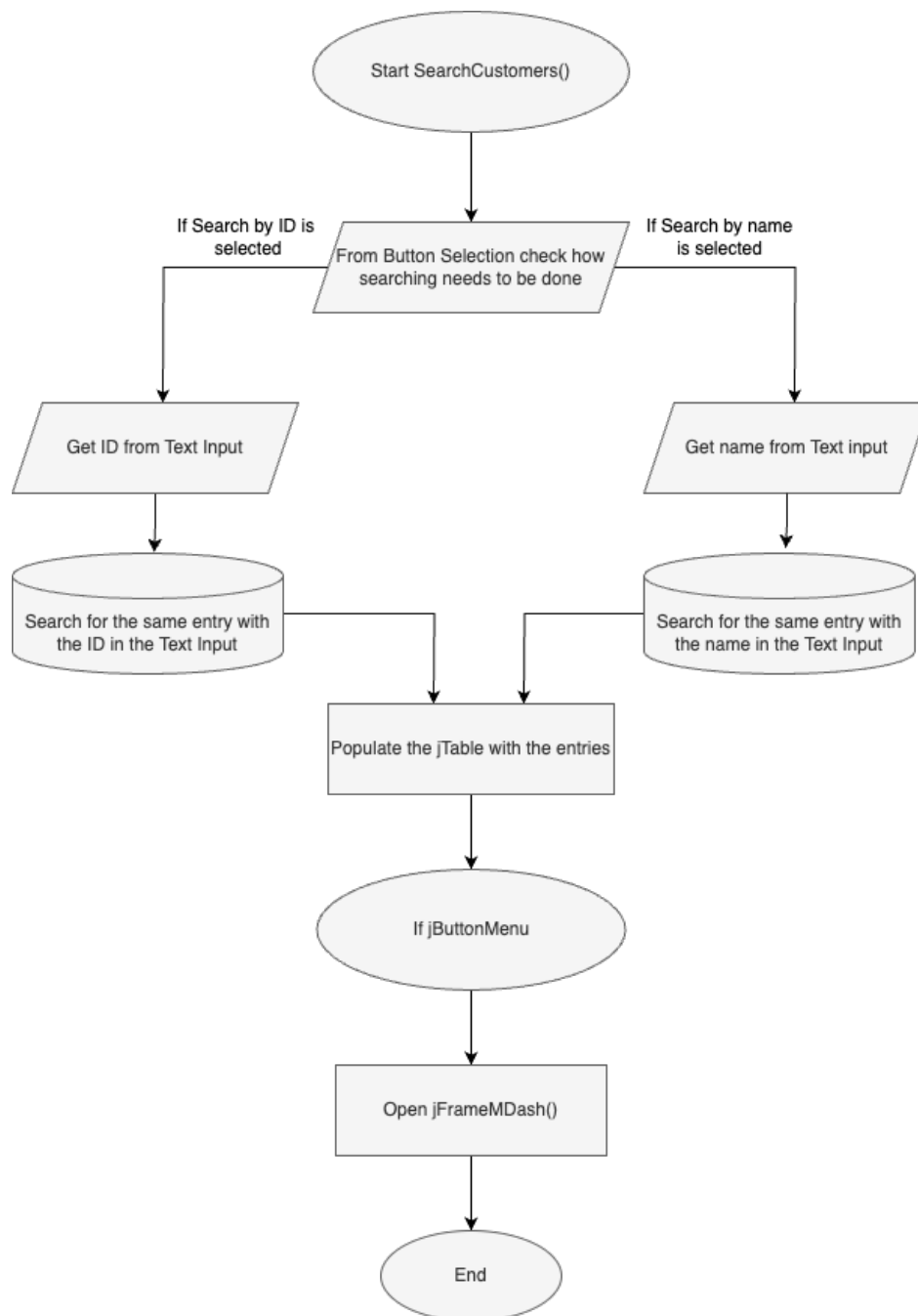
Flow Chart

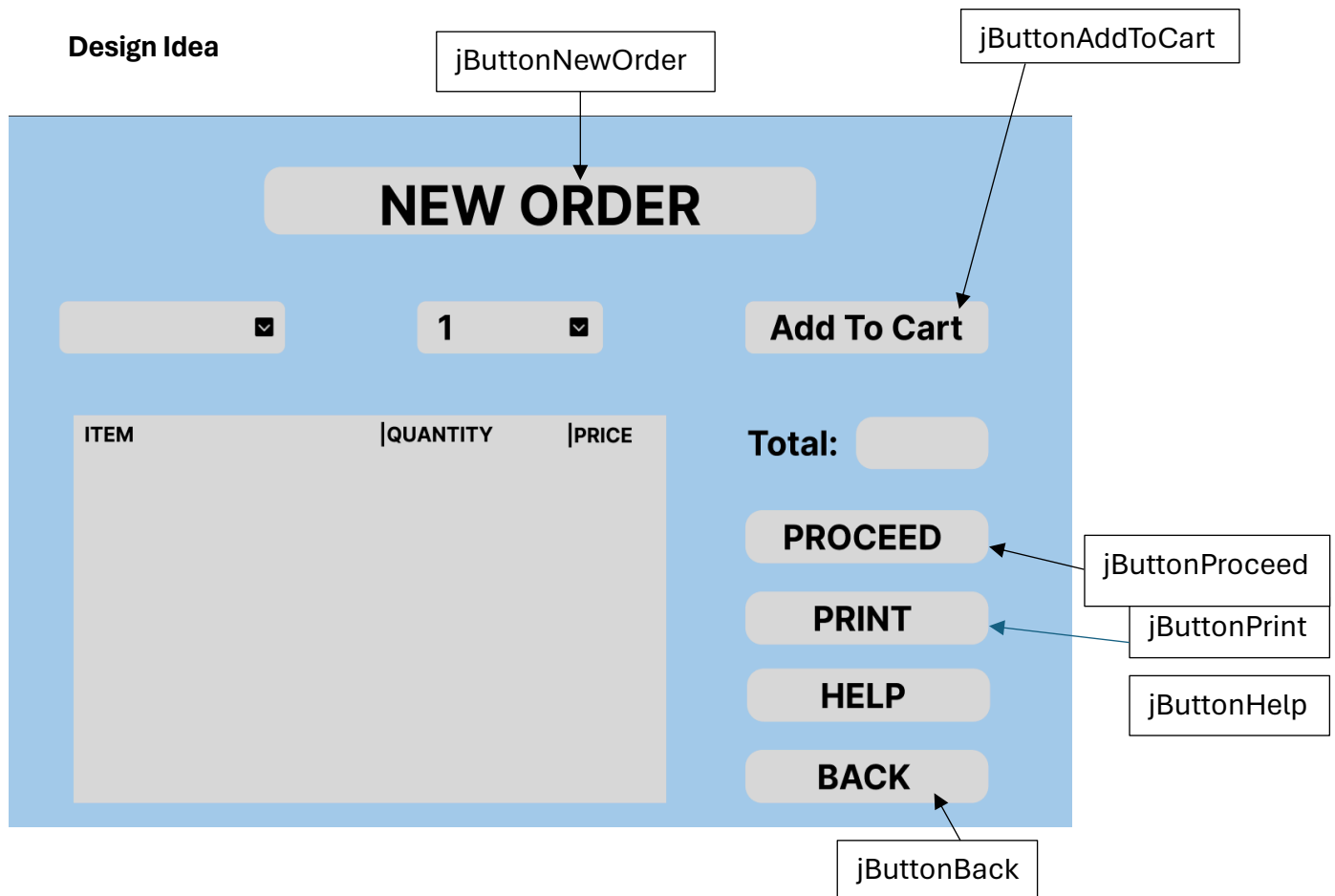
Screen 13: Search customers

Design Idea



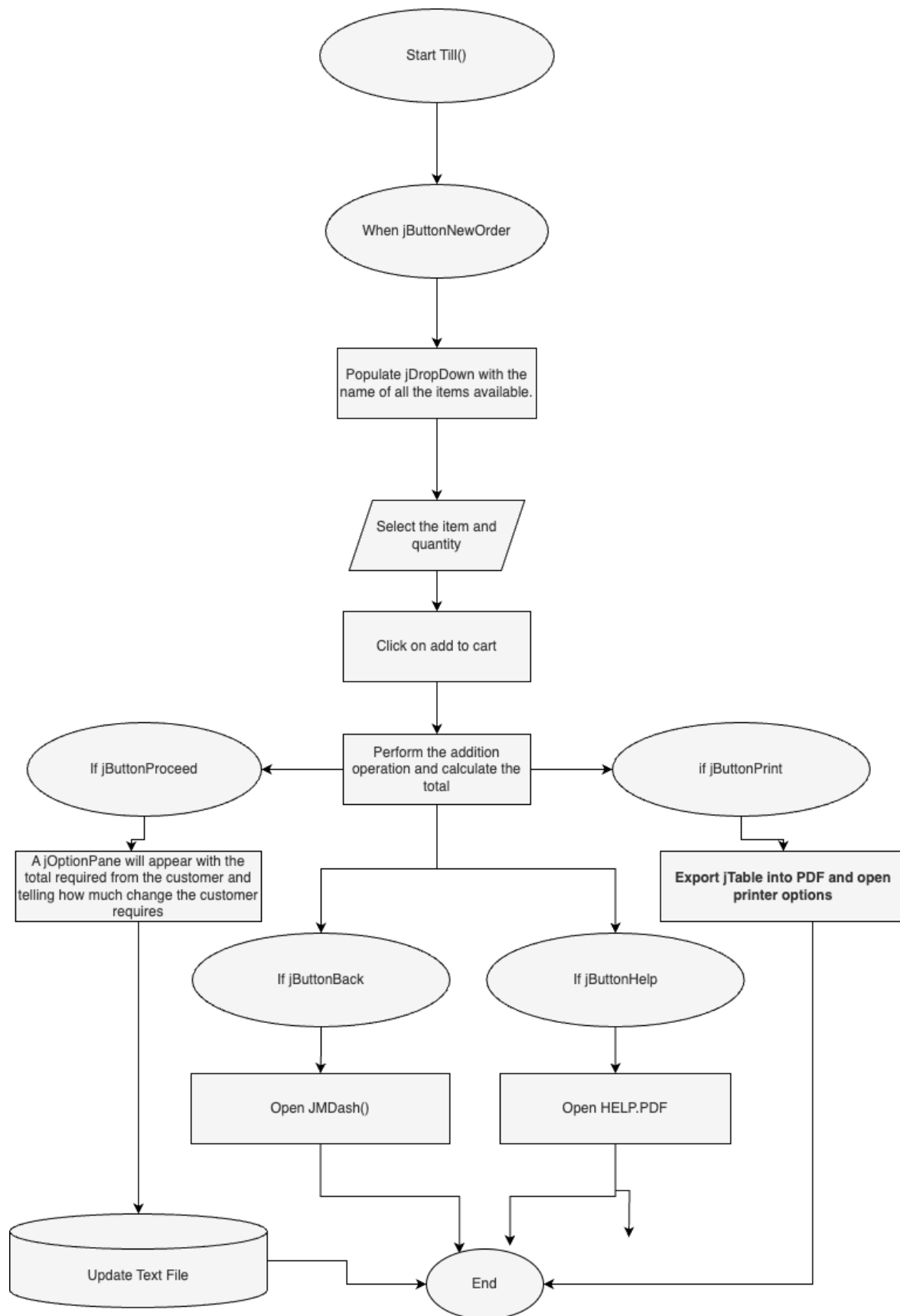
Component Name	Description	Data Access	Data Validation
jButtonSearch	Takes user to Till JFrame	No data is accessed	No data validation is required
jButtonMenu	Takes user to AddCust JFrame	No data is accessed	No data validation is required
jButtonGroup	User can select the type of search he would like to perform	No data is accessed	No data validation is required
jTextArea	It will show records	No data is accessed	No data validation is required
jlabel	Display Text	No data is accessed	No data validation is required
jTextInput	Gets the word which to search for	No data is accessed	No data validation is required

Flow Chart

Screen 14: Till**Design Idea**

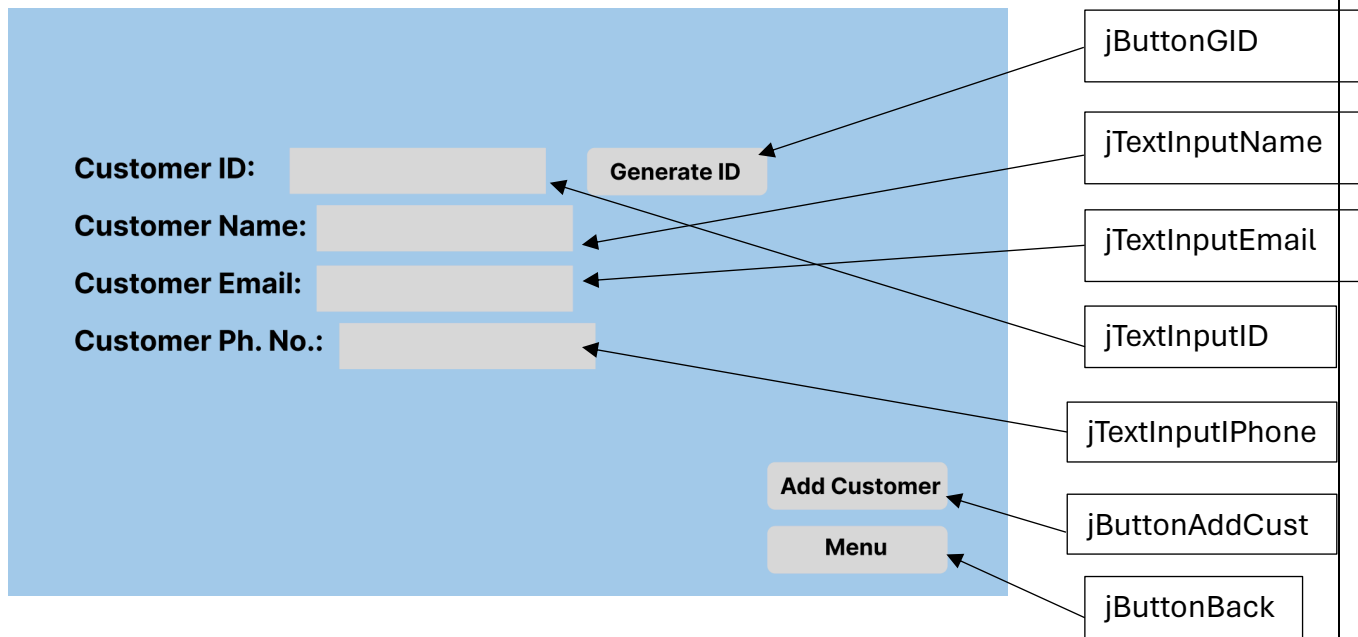
Component Name	Description	Data Access	Data Validation
jButtonHelp	Opens the help text file	The text File help is accessed	No data validation is required
jButtonBack	Takes the user to the main menu JFrame	No data is accessed	No data validation is required
jButtonPrint	Prints the jTable	No data is accessed	No data validation is required
jButtonProceed	Prompts the user to enter the amount that the customer paid	No data is accessed	No data validation is required
jButtonAddToCart	Copies the item selected and the number of units selected to the jTable	No data is accessed	No data validation is required
jButtonNewOrder	Clear all the fields in the frame	No data is accessed	No data validation is required

Flow Chart

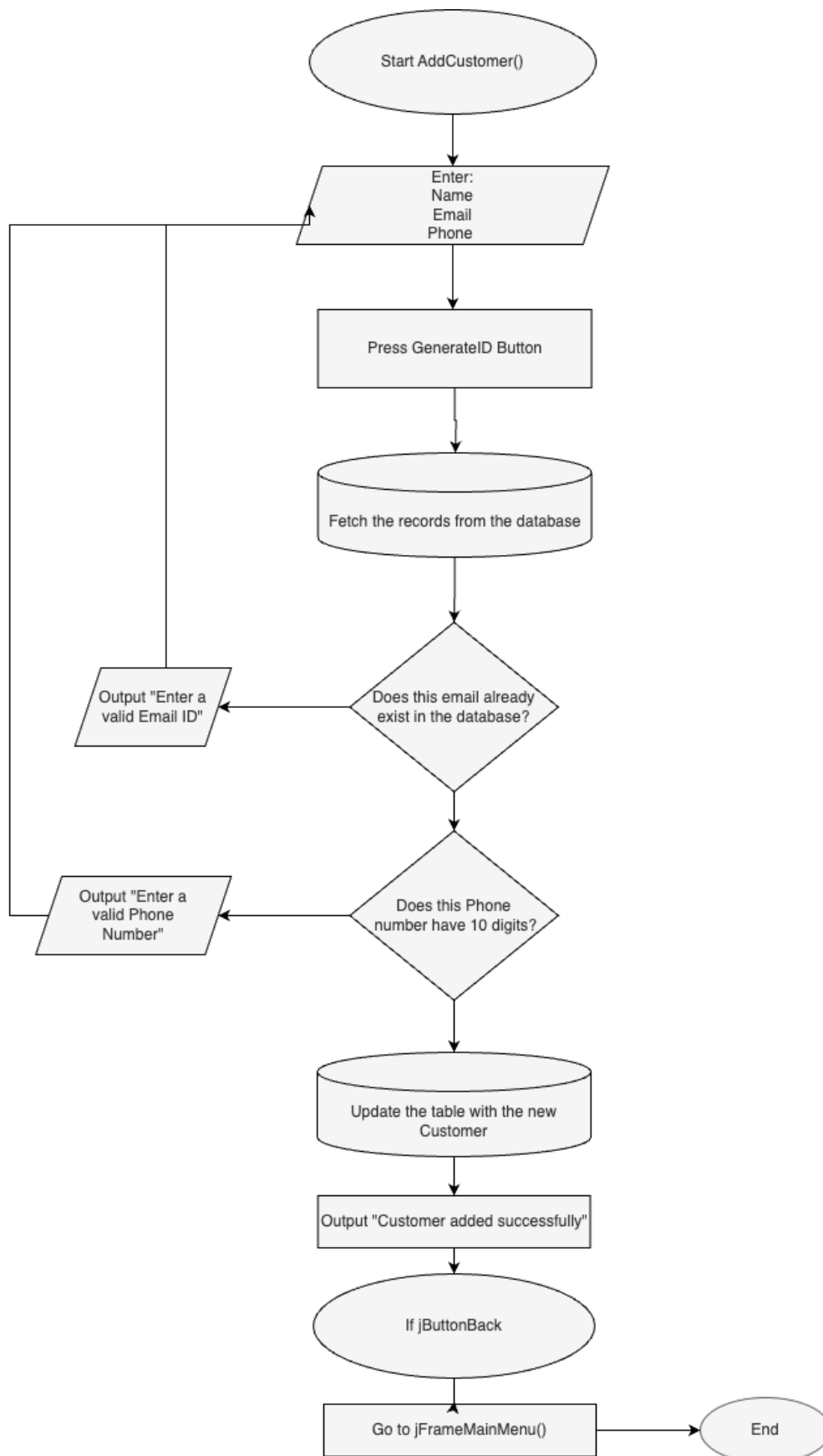


Screen 15: Customer Sign Up

Design Idea



Component Name	Description	Data Access	Data Validation
jButtonBack	Takes the user back to the Main Menu	No data is accessed	No data validation is required
jButtonAddCust	Data gets saved to the database	No data is accessed	No data validation is required
jTextInputName	Gets customers name	No data is accessed	No data validation is required
jTextInputEmail	Gets customers email	No data is accessed	No data validation is required
jTextInputID	Gets customers ID	No data is accessed	No data validation is required
jTextInputIPhone	Gets customers phone number	No data is accessed	No data validation is required
jButtonGID	Generates a unique ID for the user	No data is accessed	No data validation is required

Flow Chart

2. Database Structures

Manager Login			
Name	Type	Size	Detail
Id	Int	11	Manager ID
Name	varchar	50	Manager Name
User_Name	Varchar	50	Manager Username
Email	Varchar	50	Manager Email
Phone	Text	-	Manager Phone No.
Password	Varchar	50	Manager Password

Customer			
Name	Type	Size	Detail
Id	Int	100	Customer ID
Name	varchar	50	customer Name
Email	Varchar	50	Customer Email
Phone	Text	-	Customer Phone No.

I will use two text files with delimiters. The one file will save the usernames and passwords. The other file will save the details of the items.

The item text file will look like:

Item number # Description # Number of units left # Cost price # selling price # Units sold monthly # Date of last order

The userNames text file will look like:

User name # password

3. Application Sequencing

Admin Sequencing

1. The User enters their username and password. The Username and password is stored in variables, String Username String Password
2. The Username is searched for in the Usernames text file and, once found, the Password is validated to check if it is correct
3. If the record of the user is found in the text file, then JFrameOTP is shown OR If the Username is not found, or the Password doesn't match the password in the text file, then the user is prompted to retry
4. The OTP Gets Verified and JFrameMainMenu is displayed

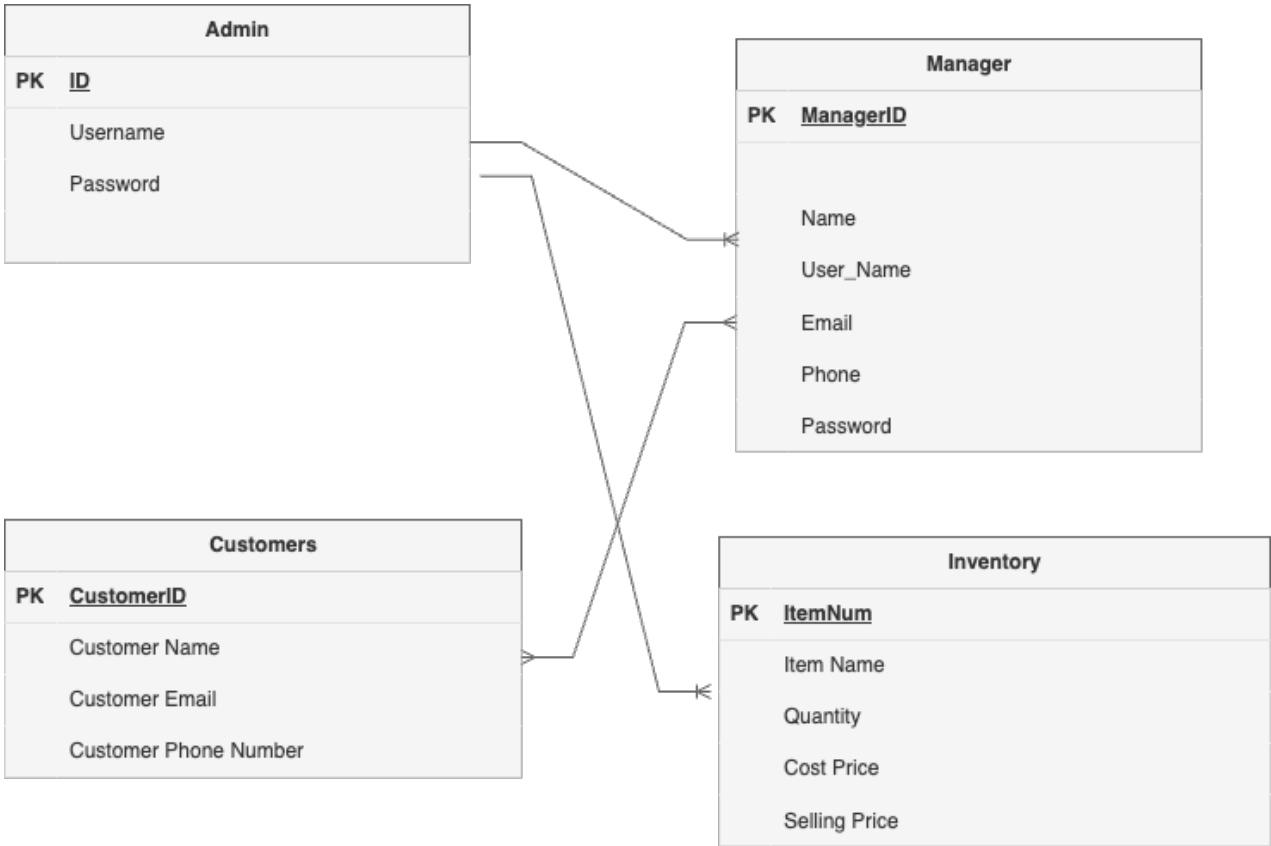
5. The user presses a button on the Main Menu
6. If the user presses the viewall JFrame:
 - a. The text file items is accessed
 - b. A table with all the items and their details is displayed
7. If the user presses the viewitem JFrame
 - a. The user is prompted to enter the item number of the item they would like to be displayed
 - b. The text file items is accessed
 - c. If this item number is not found, they are prompted to retry
 - d. If the item number is found, a JFrame is displayed with all the item details
8. If the user click the general information button
 - a. The text file Items is accessed
 - b. All the respective values to be displayed are calculated and stored in variables
 - c. Data is displayed on the JFrameGeneralInformation
9. If the user presses the newitems button,
 - a. JFrameNewItem is displayed
 - b. The user enters the relevant information and presses next
 - c. All the data selected/ entered by the user is validated
 - d. An object item is created with all the fields entered by the user
 - e. The object is used to enter the new item into the text file
10. If a user presses the deleteItem button
 - a. The user is prompted to enter an ItemNumber
 - b. The item number is searched for in the text file
 - c. If the item number is not found, the user is prompted to try again
 - d. If the item number is found, the JFrame is displayed with the item number and item description of the item
 - e. The user presses delete
 - f. The item is then deleted form the text file
11. If the user presses Help:
 - a. A text file Help is displayed
12. If the user presses log out
 - a. JFrame Log In is displayed
13. If the user presses Add Manager:
 - a. The user is able to register a new manager into the database
14. If the user presses Manager Dashboard:
 - a. The manager dashboard is displayed

Manager Sequencing

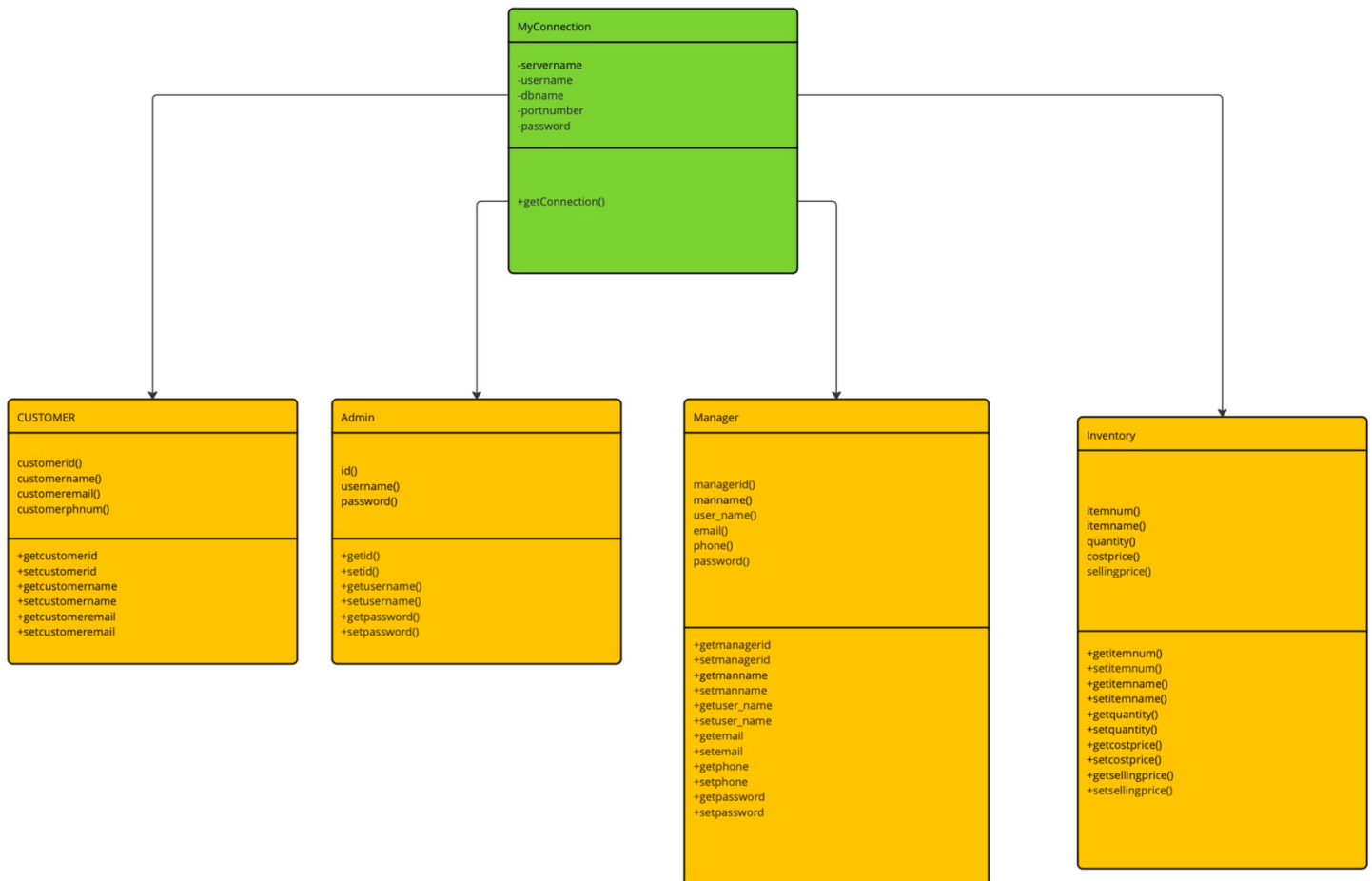
1. The User enters their username and password. The Username and password is stored checked with the credentials on phpMyAdmin to verify the manager
2. If the record of the user is found in the database, then JFrameMDash is shown OR If the Username is not found, or the Password doesn't match the password in the text file, then the user is prompted to retry
3. On JFrameTill:
 - a. The user presses new order

- b. The user selects different options from the jComboBoxes and the required quantity
 - c. The user presses Add to cart and the item
 - d. The item and the quantity is displayed in the jTable
 - e. The total is calculated
 - f. The user presses proceed
 - g. The user is prompted to enter the total amount that the customer paid
 - h. A message is displayed with the amount of change the customer requires The user presses print
 - i. The jTable is printed
 - j. The user presses print to print the jTable
 - k. The user presses Help to open the help text file
 - l. The user presses back to return to JFrameSelectLogin
4. On Add Customer:
- a. A new customer gets added into the Database created on phpMyAdmin
5. On search Customers:
- a. The database is accessed
 - b. Based on the Managers selection on Search by ID/ Search by Name, the record is searched
6. On Search Bill:
- a. The dictionary with all the customers opens and based on the customer ID, the folder gets selected and all bills are shown.

4. ERD Diagram



5. UML Diagram



6. Class Design

Item
<ul style="list-style-type: none"> ■ int ItemNumb ■ String Description ■ int StockLeft ■ Double CostPrice ■ Double SellingPrice ■ int SoldMonthly ■ String LastOrder
<ul style="list-style-type: none"> ■ Item (int itemNumber, String description, int stockLeft, Double costPrice, Double sellingPrice, int soldMonthly, String lastOrder,) <i>Parameterised Constructor which requires a parameter for all the fields</i> ■ Item () <i>A Constructor that is not parameterised</i> ■ int getItemNumber <i>A typed method that gets the item number and returns the integer</i> ■ String getDescription () <i>A typed method that gets the description of an item and returns the String</i> ■ int getStockLeft () <i>A typed method that gets the amount of stock left and returns the integer</i> ■ Double getCostPrice <i>A typed method that returns the cost price of the item as a double</i> ■ Double getSellingPrice <i>A typed method that returns the selling price of the item as a double</i> ■ int getSoldMonthly <i>A typed method that returns the number of units sold monthly as an integer</i> ■ String getLastOrder <i>A typed method that returns the date of last order as a String</i> ■ void setDescription (String desc) <i>A void method that sets the description of an item</i> ■ void setStockLeft (int StockLeft) <i>A void method that sets the amount of stock left</i> ■ void setCostPrice (double costPrice) <i>A void method that sets the cost price of an item</i> ■ void setSellingPrice (double sellingPrice) <i>A void method that sets the selling price of an item</i> ■ void setSoldMonthly (int soldMonthly) <i>A void method that sets the number of units sold monthly</i> ■ void setLastOrder (String lastOrder) <i>A void method that sets the value of the date of last order</i> ■ void EditItem (int itemNumber, String description, int stockLeft, double costPrice, double sellingPrice, int soldMonthly, String lastOrder) String lastOrder) <i>A void method that changes the value of the number of the item</i>

ItemArray

- Item [] arr
- int Count

- itemArray()
A method to create the array of objects using a text file to read data
- int getCount ()
A typed method to return the size of the array

item SearchItem (int itemNumber)

A typed method to search for an item and return that item

- boolean SearchItemNumber (int itemNumber)
A typed method to search for an item and return true if that item is found
- void InsertItem (String description, int StockLeft, double CostPrice, double SellingPrice, int SoldMonthly, String LastOrder)
A void method to insert an item into the array
- void DeleteItem (int ItemNum)
A void method to delete an item from the array
- int nextItemNumber ()
A typed method to return the next available item number
- void saveToFile ()
A void method to re-write the text file
- double totalCost()
A typed method to return the total cost of multiple items
- double totalProfit ()
A typed method to return the total profit made in the past month
- double itemprice (String name, double price)
A typed method to return the selling price of an item
- String runningLow ()
A typed method to return the list of items running low on stock
- void EditItem ()
A void method to edit details of an item
- void updateFile (String item, int amount)
A void method to update the number of units left based on the till system
- String checkOut (double owe, double paid)
A typed method to return the total change the customer requires

UserName
<ul style="list-style-type: none">■ String username■ String password
<ul style="list-style-type: none">- UserName (String username, String password) <i>A parametrised constructor to create the UserName object</i>- Boolean Check () <i>A typed method to return true if the username and password is found in the text file.</i>

7. Test Plan

Action Tested	Ways of testing
Create a username and password	<ul style="list-style-type: none"> - Enter valid username and password, verify successful creation. - Enter invalid username or password, verify appropriate error message
Check password against stored password	<ul style="list-style-type: none"> - Enter correct password, verify successful login - Enter incorrect password, verify unsuccessful login
View all items on the text file	<ul style="list-style-type: none"> - Access the "View All" option, verify all items are displayed
View a specific item	<ul style="list-style-type: none"> - Select a specific item, verify only that item is displayed
View quantity of sold item	<ul style="list-style-type: none"> - Access sold item information, verify correct quantity is displayed
View remaining stock of item	<ul style="list-style-type: none"> - Access remaining stock information, verify correct quantity is displayed
View total cost for ordering new stock	<ul style="list-style-type: none"> - Access cost information, verify correct total cost is displayed
Add a new item onto the system	<ul style="list-style-type: none"> - Add new item, verify item is successfully added
Delete an item from the text file	<ul style="list-style-type: none"> - Select item for deletion, verify item is removed from the system
View general information about stock	<ul style="list-style-type: none"> - Access general information, verify correct information such as total cost, total profit, low stock items displayed
View profits	<ul style="list-style-type: none"> - Access to viewing the monthly sales
Generate Graph of stock running low	<ul style="list-style-type: none"> - Access to viewing graph and quantity
Open Google	<ul style="list-style-type: none"> - Click on Google option, verify Google webpage is opened
Open help text file	<ul style="list-style-type: none"> - Click on help option, verify help text file is opened
Log out	<ul style="list-style-type: none"> - Click on logout option, verify user is logged out successfully
Choose multiple items	<ul style="list-style-type: none"> - Select multiple items, verify all selected items are processed correctly
View total order amount	<ul style="list-style-type: none"> - Verify total order amount is calculated accurately
Enter amount paid by customer	<ul style="list-style-type: none"> - Enter paid amount, verify entered amount is accepted and processed
View required change	<ul style="list-style-type: none"> - Verify correct change amount is calculated based on paid amount and total order amount

Print order	- Click on print option, verify order is printed successfully
Text fields for user input	- Enter data into text fields, verify entered data is accepted and processed correctly
Buttons for navigating to next JFrame	- Click on buttons, verify correct navigation to the respective JFrame
Buttons with different functions	- Click on buttons, verify each button performs its intended function
Check box for user selection	- Tick checkboxes, verify selected options are processed correctly
Combo box for selecting options	- Select options from combo box, verify selected options are processed correctly
Table for displaying list of items	- Verify all items are displayed correctly in the table
JCalendar for selecting specific date	- Select a date using JCalendar, verify selected date is accepted and processed correctly
Text Area for displaying information	- Verify text area displays information correctly

8. Updated design Idea after client Feedback

A discussion had taken place between the client and me where (Appendix 1D)¹ where the client asked me to make a few changes to the JFrameTill screen as he thought it would be a bit difficult for the admins to understand. Below is the updated design idea with the GST and the Customer ID text input.

NEW ORDER

Add To Cart

ITEM	QUANTITY	PRICE
------	----------	-------

Total:

GST **C. ID**

PROCEED
PRINT
HELP
BACK

¹ Appendix 1D (Page 5)