Nile University

Fall 2025

(CSCI217) Advanced Computer Programming and Concept Project

Instructor: Dr. Sameh Abd El Rahman

TA: Rahma Mohamed

Date: 28/12/2024

Marwan Mohamed Zaki 211001365

Omar Samir Mohamed 211000372

Mahmoud Reda Abderauf 211000489

Abdullah Giath Elkhateeb 211001730

Detailed Report on Pharmacy Management System

Project Overview

The Pharmacy Management System is a Java-based application designed to facilitate the management of drugs in a pharmacy. It includes functionalities such as adding, removing, and ordering drugs, managing inventory, tracking sales, and providing a graphical user interface (GUI) for user interaction. The project is implemented using four main classes: Drug, Pharmacy, Order, and PharmacyGUI.

Table of Contents

Classes Overview

Drug Class

The Drug class represents a drug entity with attributes and methods to manage its properties.

Attributes:

- **private int quantity**: Stores the quantity of the drug in stock.
- **private String id**: A unique identifier for the drug.
- **private String name**: The name of the drug.
- **private String category**: The category of the drug (e.g., Cosmetics, Prescription).
- **private double price**: The price of the drug.

Methods:

- **public Drug(String id, int quantity, String name, String category, double price)**: Constructor to initialize a drug.
- **public int getQuantity()**: Retrieves the quantity of the drug.
- **public void setQuantity(int quantity)**: Updates the quantity of the drug.
- **public String getId()**: Retrieves the ID of the drug.
- **public void setId(String id)**: Updates the ID of the drug.
- **public String getName()**: Retrieves the name of the drug.
- **public void setName(String name)**: Updates the name of the drug.
- **public String getCategory()**: Retrieves the category of the drug.
- **public void setCategory(String category)**: Updates the category of the drug.
- **public double getPrice()**: Retrieves the price of the drug.
- **public void setPrice(double price)**: Updates the price of the drug.

Pharmacy Class

The Pharmacy class manages a collection of drugs and their capacity.

Attributes:

- **private int capacity**: Maximum number of drugs the pharmacy can store.
- **private ArrayList<Drug> drugs**: A list of Drug objects.

Methods:

- **public Pharmacy(int capacity)**: Constructor to initialize the pharmacy.
- **public int getCapacity()**: Retrieves the pharmacy's capacity.
- **public void setCapacity(int capacity)**: Updates the capacity.
- **public ArrayList<Drug> getDrugs()**: Returns the list of drugs.
- **public boolean addDrug(Drug drug)**: Adds a drug to the pharmacy if there is available capacity.
- **public boolean removeDrug(String id)**: Removes a drug by its ID.
- **public Drug dispDrug(String id)**: Retrieves a drug by its ID.

Order Class

The Order class handles drug orders and tracks total sales.

Attributes:

- **private static double totalsales**: A static variable to track the total sales revenue.

Methods:

- **public boolean addOrder(Drug drug, int quantity)**: Places an order for a specified quantity of a drug and updates sales.
- **public double getTotalSales()**: Retrieves the total sales.
- **public void resetTotalSales()**: Resets the total sales to zero.

PharmacyGUI Class

The PharmacyGUI class provides a graphical interface for managing the pharmacy.

Attributes:

- **private JFrame frame**: The main frame of the GUI.
- **private JPanel addPanel**: Panel for adding drugs.
- **private JPanel inventoryPanel**: Panel for displaying inventory.
- **private JPanel orderPanel**: Panel for managing orders.
- **private JTextField idField**: Text field for entering drug ID.
- **private JTextField nameField**: Text field for entering drug name.
- **private JTextField categoryField**: Text field for entering drug category.
- **private JTextField priceField**: Text field for entering drug price.
- **private JTextField quantityField**: Text field for entering drug quantity.
- **private JTextArea displayArea**: Text area for displaying inventory.
- **private JComboBox<String> colorComboBox**: Dropdown for selecting background color.
- **private JButton addDrugButton**: Button for adding a drug.

- **private JButton displayDrugsButton**: Button for displaying all drugs.
- **private JButton orderButton**: Button for placing an order.
- **private JButton removeDrugButton**: Button for removing a drug.
- **private JButton viewSalesButton**: Button for viewing total sales.
- **private JButton resetSalesButton**: Button for resetting sales data.
- **private JButton changeColorButton**: Button for changing the background color.
- **private JTextField capacityField**: Text field for setting pharmacy capacity.
- **private JButton setCapacityButton**: Button for updating pharmacy capacity.
- **private Pharmacy pharmacy**: An instance of the Pharmacy class to manage drug data.
- **private Color defaultBackgroundColor**: Stores the default background color for theme management.
- **private static final String realpassword**: Stores the authentication password.

Methods:

- **public PharmacyGUI()**: Constructor that initializes the GUI and handles authentication.
- **private void initialize()**: Sets up the GUI components and layout.
- **private void addDrug()**: Handles adding new drugs to the pharmacy.
- **private void displayDrugs()**: Displays all drugs in the inventory.
- **private void placeOrder()**: Places an order for a drug.
- **private void removeDrug()**: Removes a drug from the inventory.
- **private void viewTotalSales()**: Displays total sales.
- **private void resetSales()**: Resets sales data.
- **private void changeBackgroundColor()**: Changes the background color of the GUI.
- **private void setCapacity()**: Sets the pharmacy capacity.
- **public static void playSound(String soundFile)**: Plays a sound effect for user feedback.
- **public static void main(String[] args)**: Entry point of the program.

Error Handling

The application includes robust error handling mechanisms:

1. **Authentication**: Limits password attempts to 3 and terminates the program on failure.

2. **Input Validation**:
    - Ensures valid numeric input for prices, quantities, and capacities.
    - Handles invalid inputs gracefully using try-catch blocks.
    - Null Input Validations.
3. **Capacity Checks**: Prevents adding drugs beyond the pharmacy's capacity.
4. **Null Checks**: Verifies the existence of drugs before performing operations.
5. **User Feedback**: Displays appropriate error messages using JOptionPane.

Instructions for Using the System

1. **Entering Password**: Enter a value in the JOptionPane and press Ok to continue to the next step.

2. **Setting Initial Capacity**: Enter a numeric value in the JOptionPane and press Enter to set the Pharmacy's capacity.

3. **Display Drugs**: Clicking “Display All Drugs” to show the drugs you have in the storage.

4. **Order a Drug**: enter the drug's ID and the quantity you want, Click 'Order Drug". If the drug and quantity are available order is accepted.

5. **Remove a Drug**: enter the drug's ID, Click 'Remove Drug". If the drug is available removal is accepted.

6. **Reset Sales**: Click "Reset Sales" to reset the total sales of the day.

7. **View Total Sales**: Click "View All Sales" to display all Sales in the Pharmacy.

8. **Updating Capacity**: Enter a numeric value for the new pharmacy capacity, Click "Reset" to update the pharmacy’s capacity.

9. **BackGround Color Changer**: Choose the color you want from the ComboBox, Click "Apply" to change the color of the background.

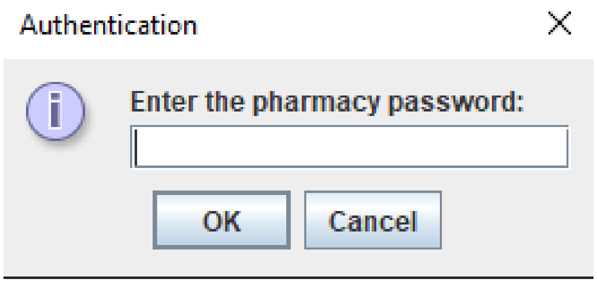
GUI Design and Features

The GUI is designed using Java Swing and includes the following features:

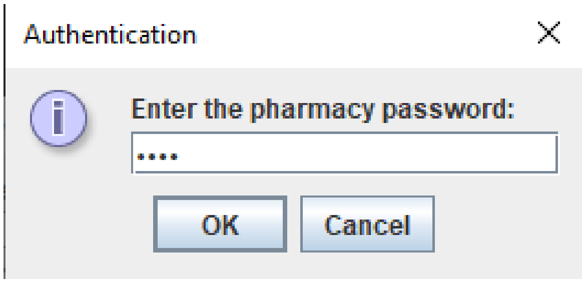
- **Add Panel**: Inputs for adding new drugs.
- **Inventory Panel**: Displays the list of all drugs.
- **Order and Removal Panel**: Manages orders, removals, and sales.
- **Customization**: Allows changing background colors and updating pharmacy capacity.
- **Authentication**: Password-protected access.

Photos to Include

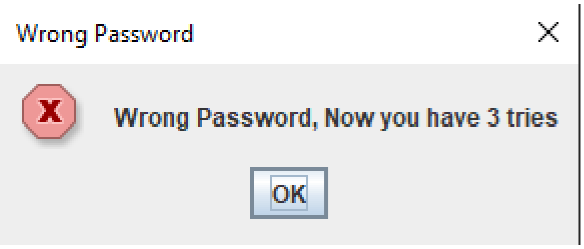
1. **Authentication and Initial Capacity Dialog**:

- 

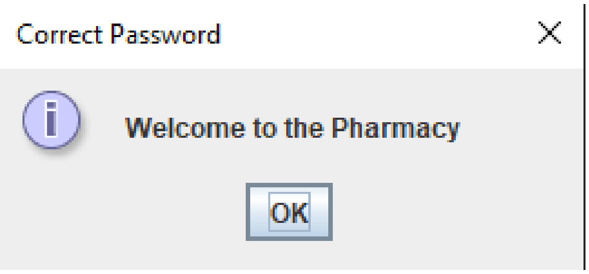
Enter the pharmacy password:

- 

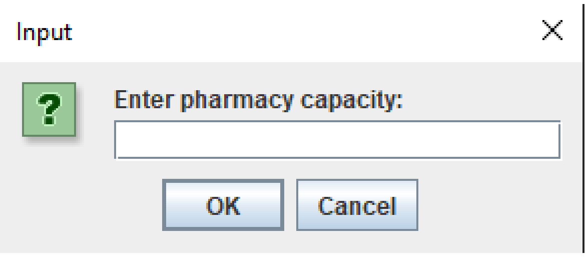
Enter the pharmacy password:
••••

- 

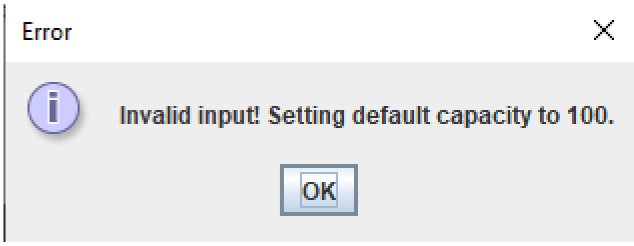
Wrong Password, Now you have 3 tries

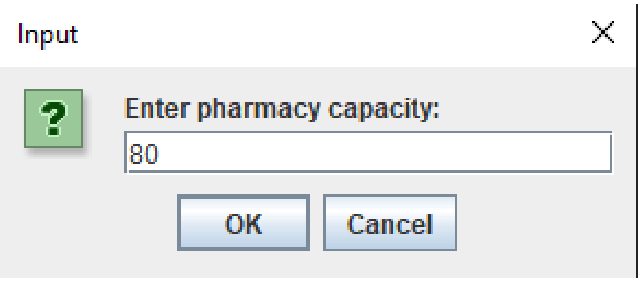
- 

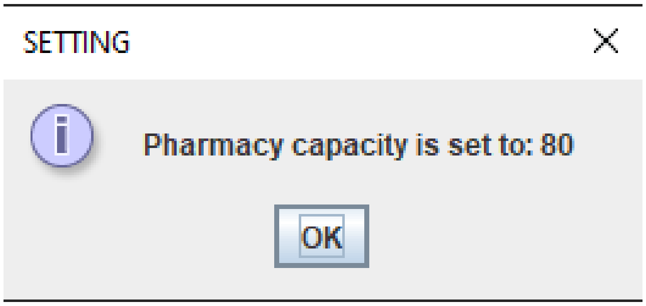
Welcome to the Pharmacy

- 

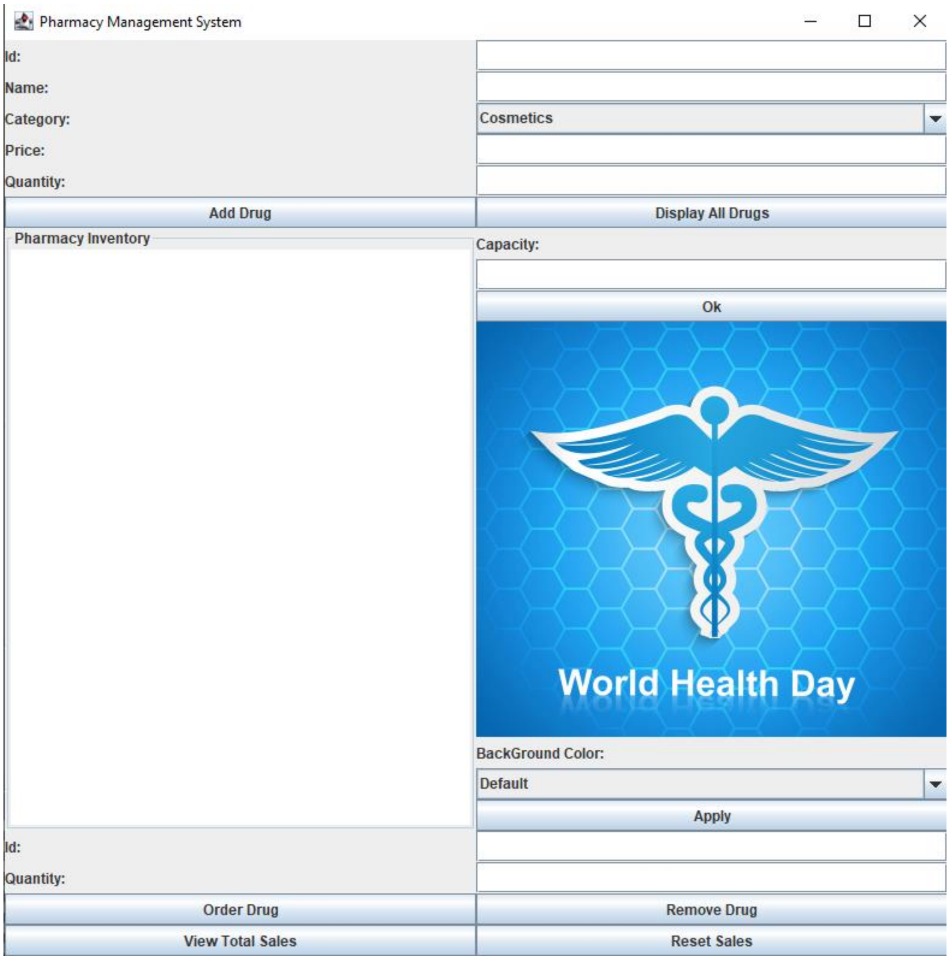
Enter pharmacy capacity:

- 
  Error
  Invalid input! Setting default capacity to 100.

- 
  Input
  Enter pharmacy capacity:
  80

- 
  SETTING
  Pharmacy capacity is set to: 80

2. **Pharmacy Frame**:

- 

Id:

Name:

Category: Cosmetics

Price:

Quantity:

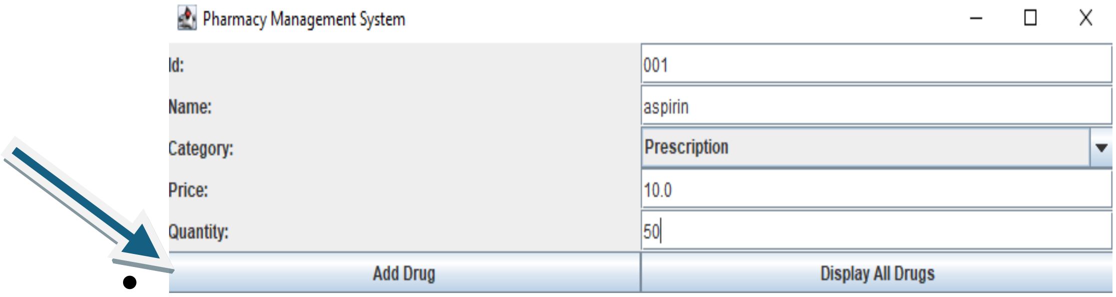
Capacity:

BackGround Color: Default

Id:

Quantity:

3. **Add Drug Panel**:

- 

Id: 001

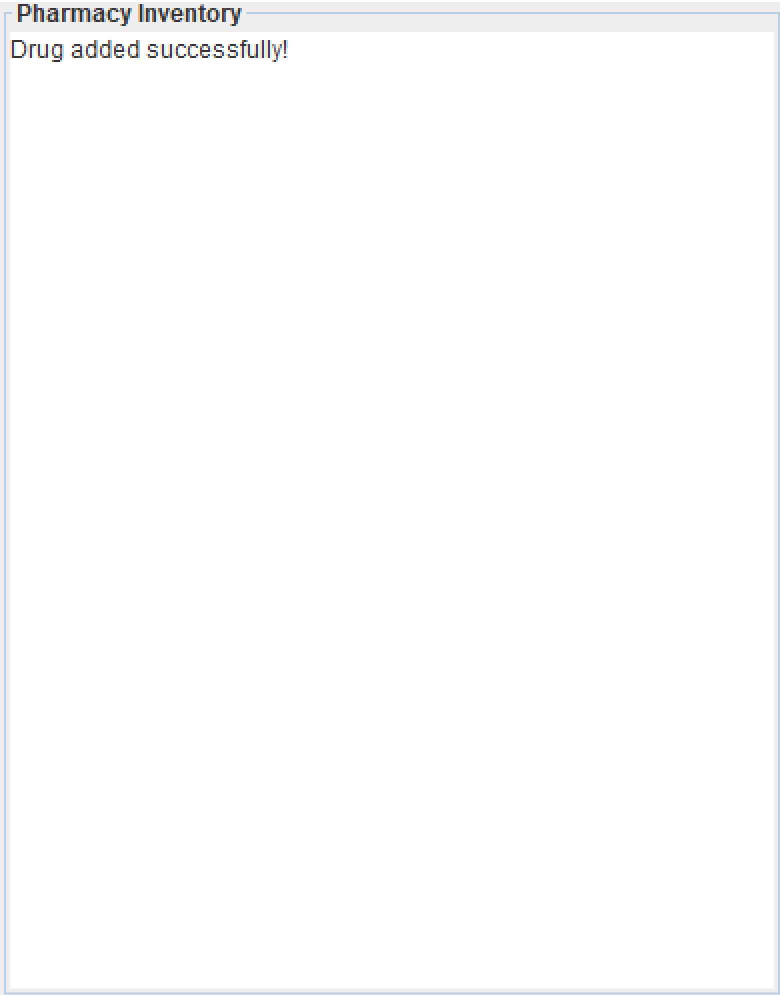
Name: aspirin

Category: Prescription

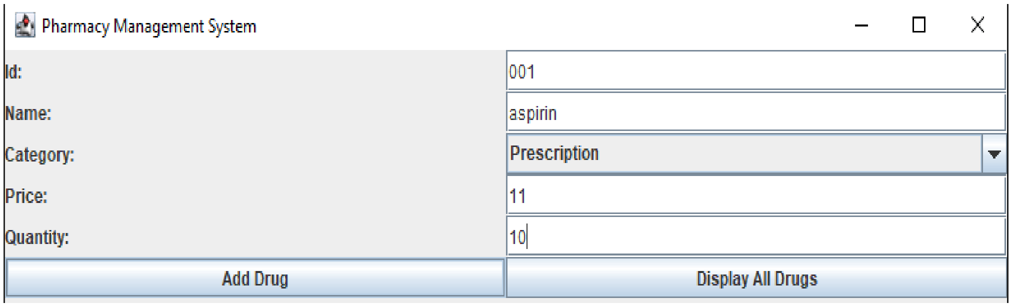
Price: 10.0

Quantity: 50

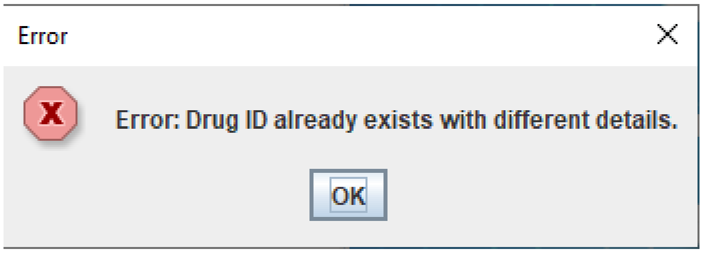
-

- 

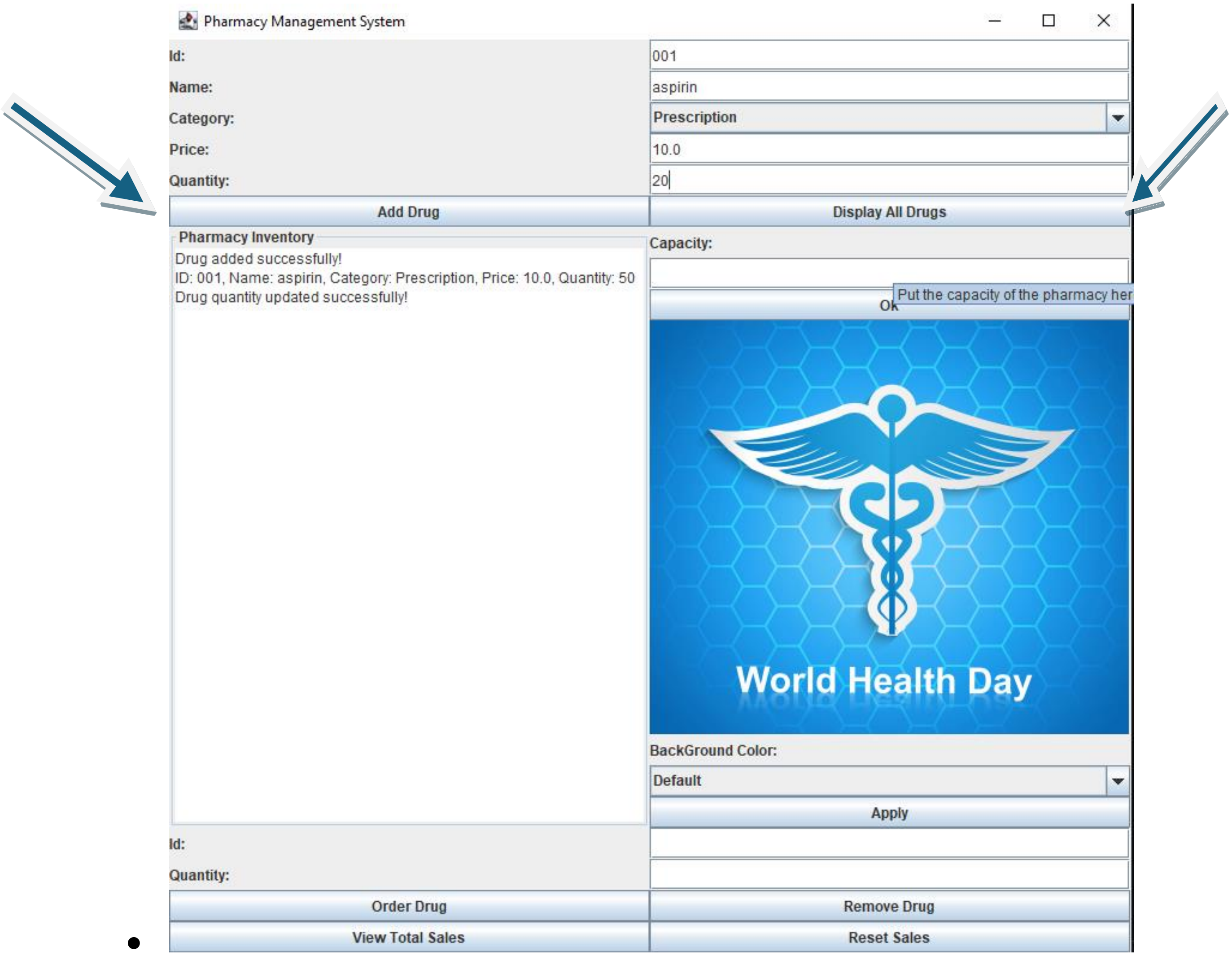
  Drug added successfully!

- 

Id:	001
Name:	aspirin
Category:	Prescription
Price:	11
Quantity:	10

- 

  Error: Drug ID already exists with different details.



Pharmacy Management System

Id: 001

Name: aspirin

Category: Prescription

Price: 10.0

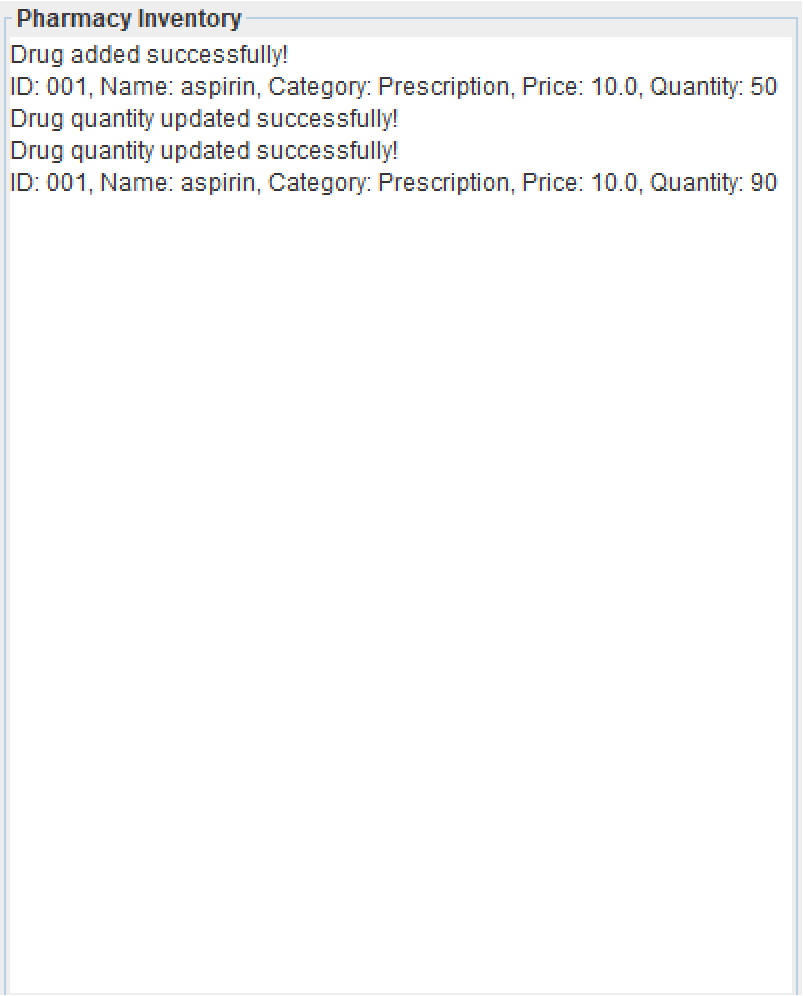
Quantity: 20

Pharmacy Inventory

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
```

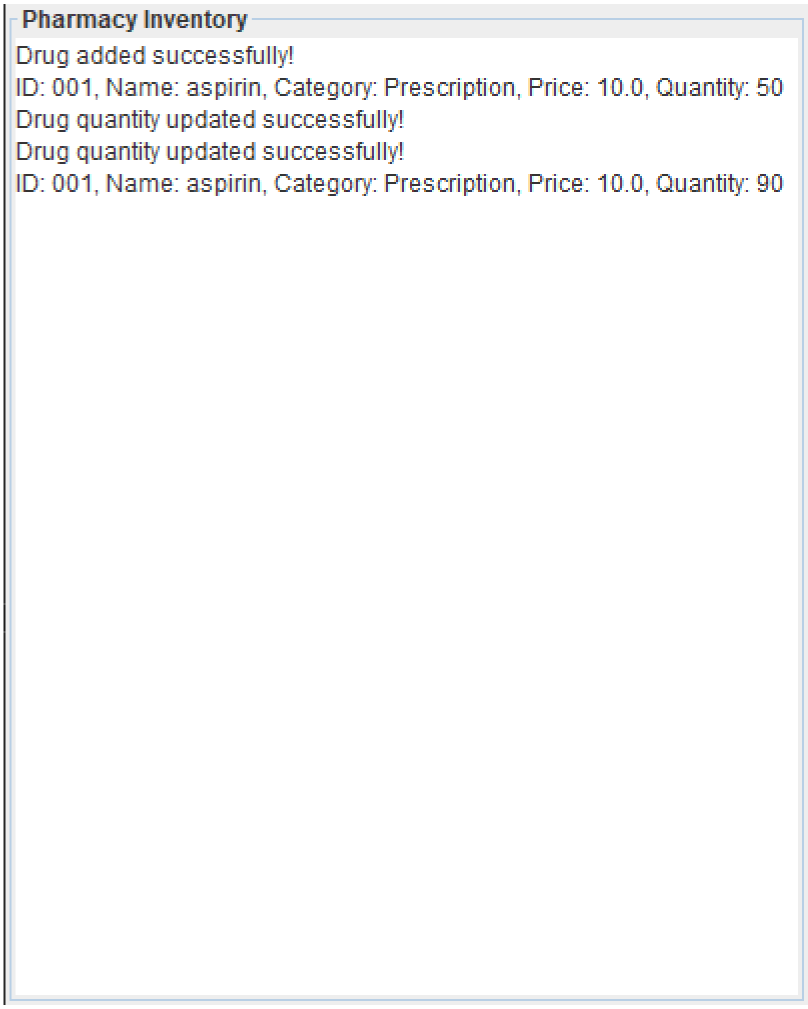
BackGround Color: Default

Put the capacity of the pharmacy her

- 

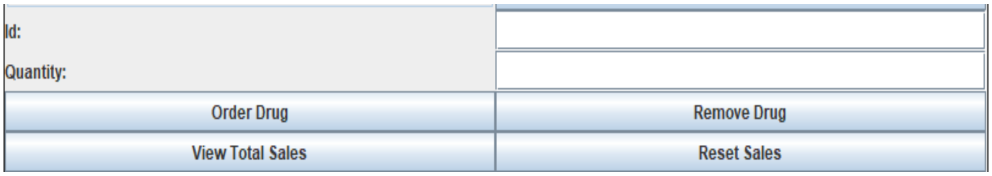
```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
```

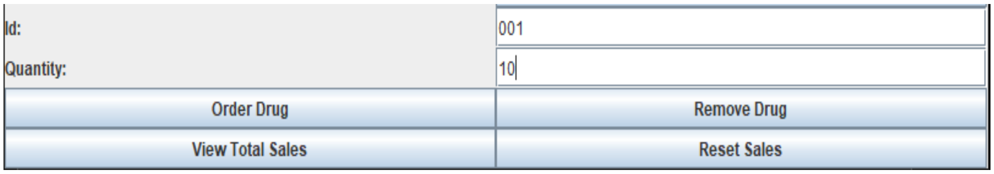
4. **Inventory Display**:

- 

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
```

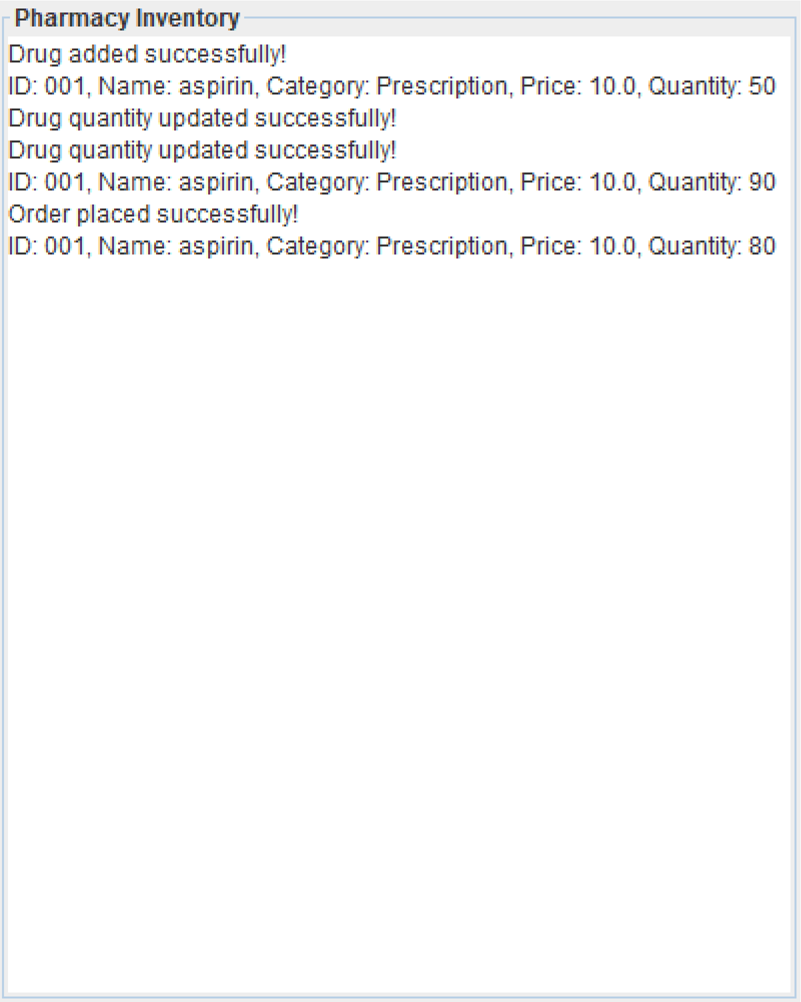
5. **Order and Removal Management**:

- 

- 

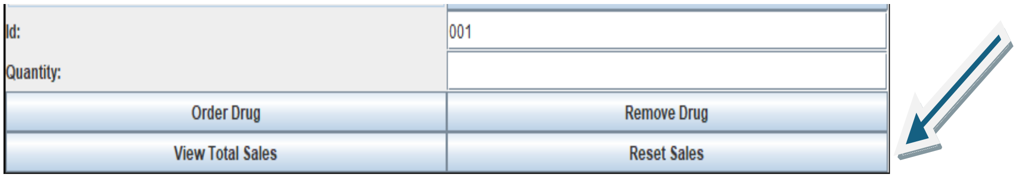
Id: 001

Quantity: 10

- 

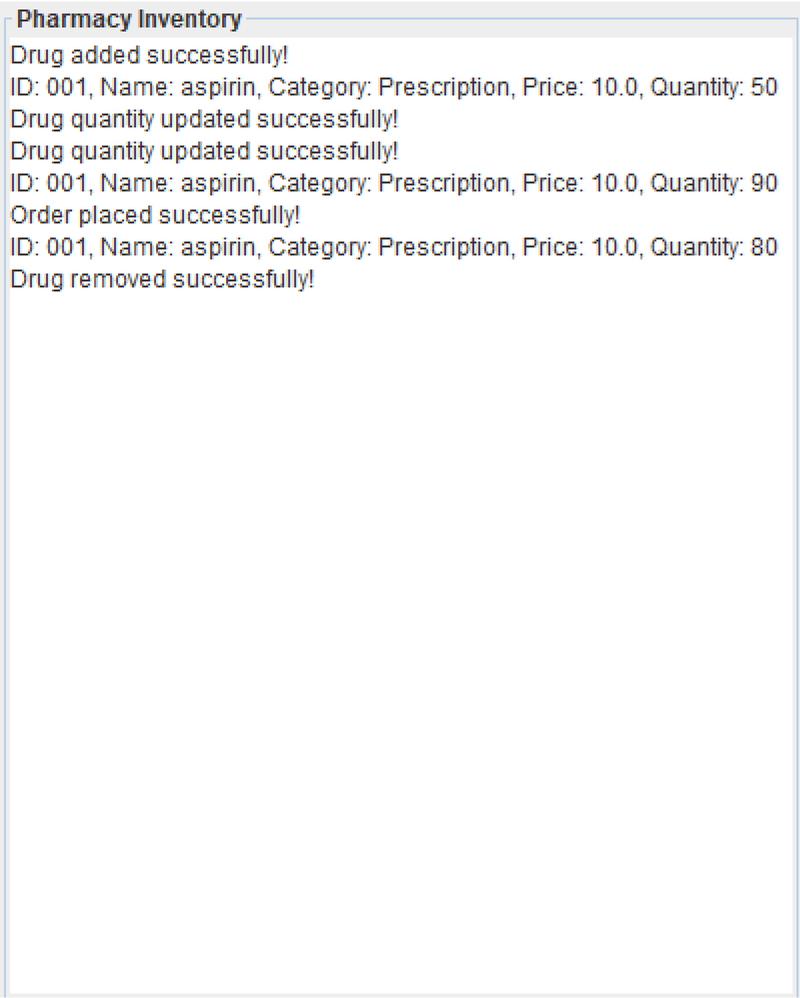
Pharmacy Inventory

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
Order placed successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 80
```

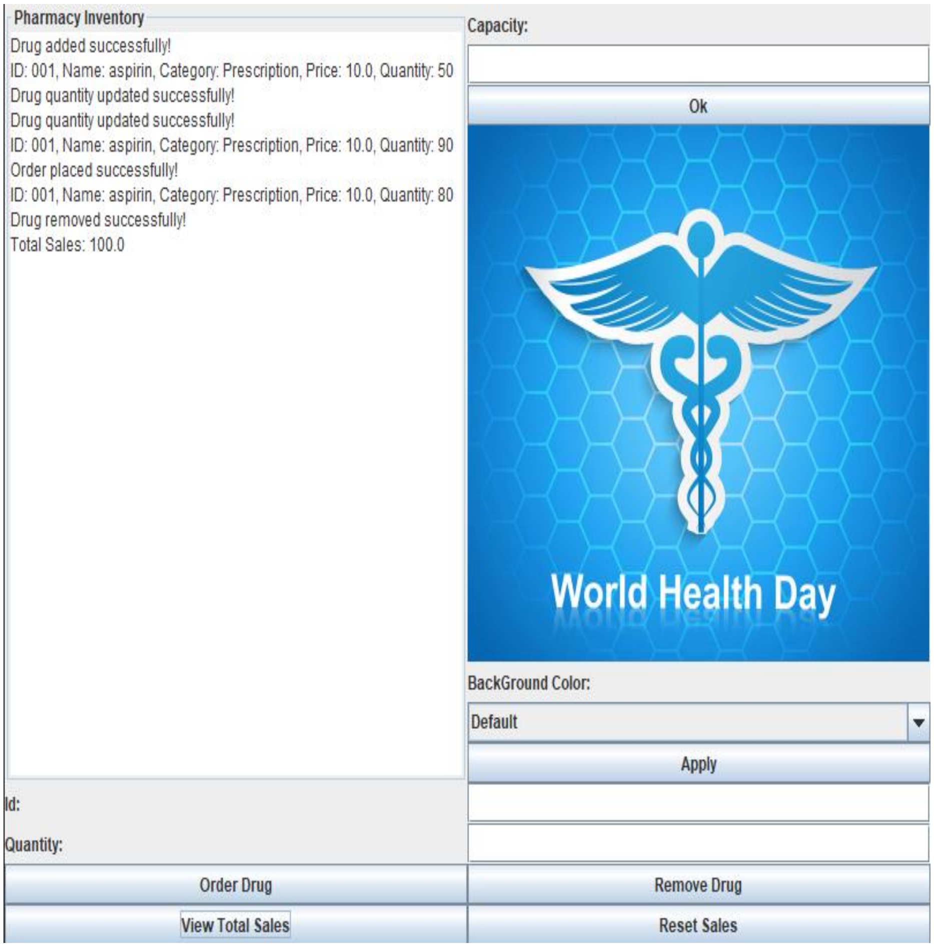
- 

Id: 001

Quantity:

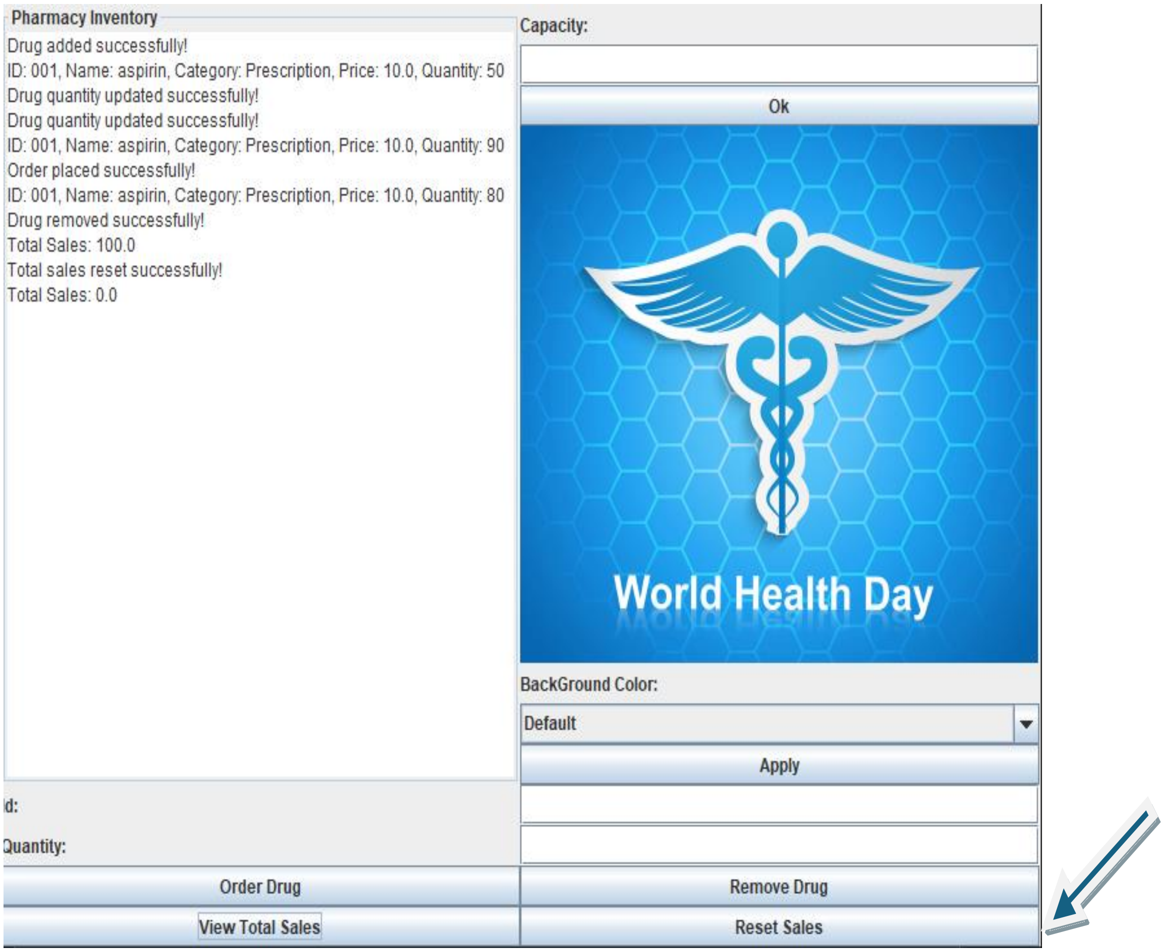
- 

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
Order placed successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 80
Drug removed successfully!
```



```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
Order placed successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 80
Drug removed successfully!
Total Sales: 100.0
```

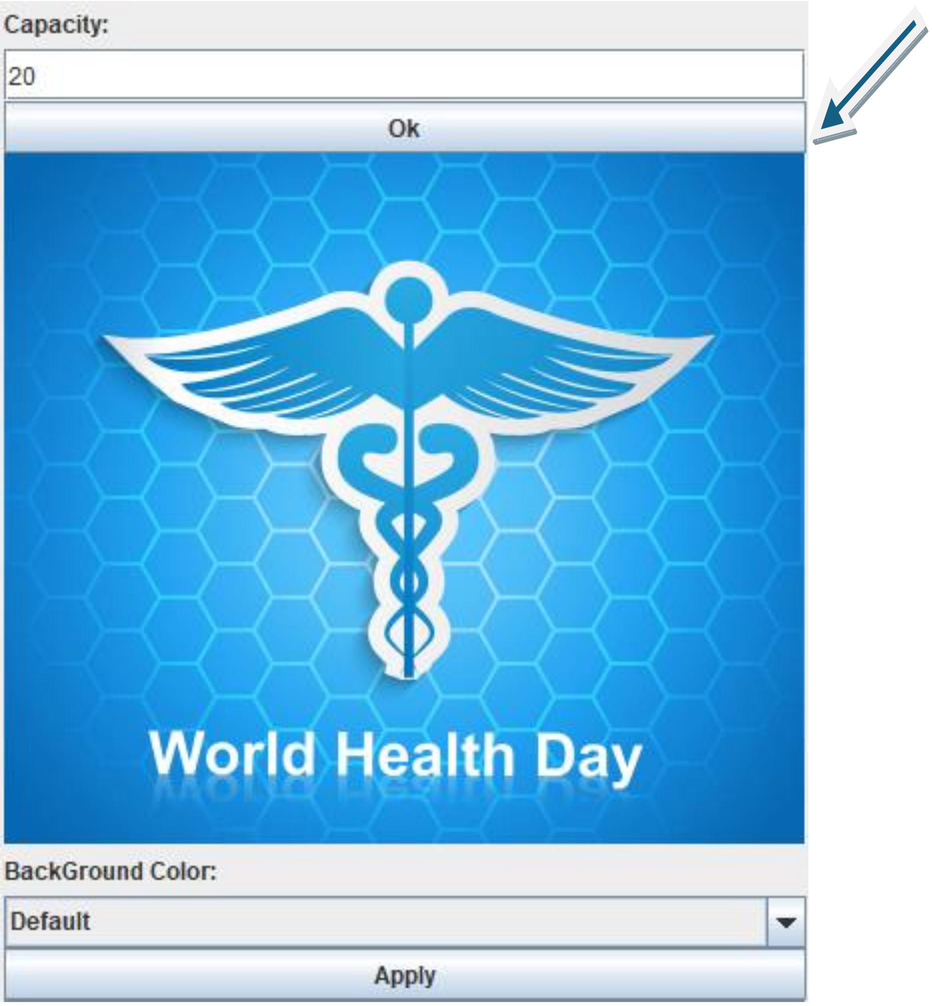
BackGround Color: Default

- 

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
Order placed successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 80
Drug removed successfully!
Total Sales: 100.0
Total sales reset successfully!
Total Sales: 0.0
```

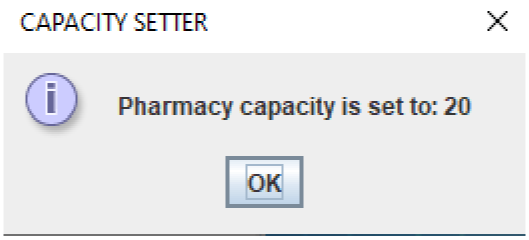
BackGround Color: Default

6. **Theme Customization and Capacity Updater**:

- 

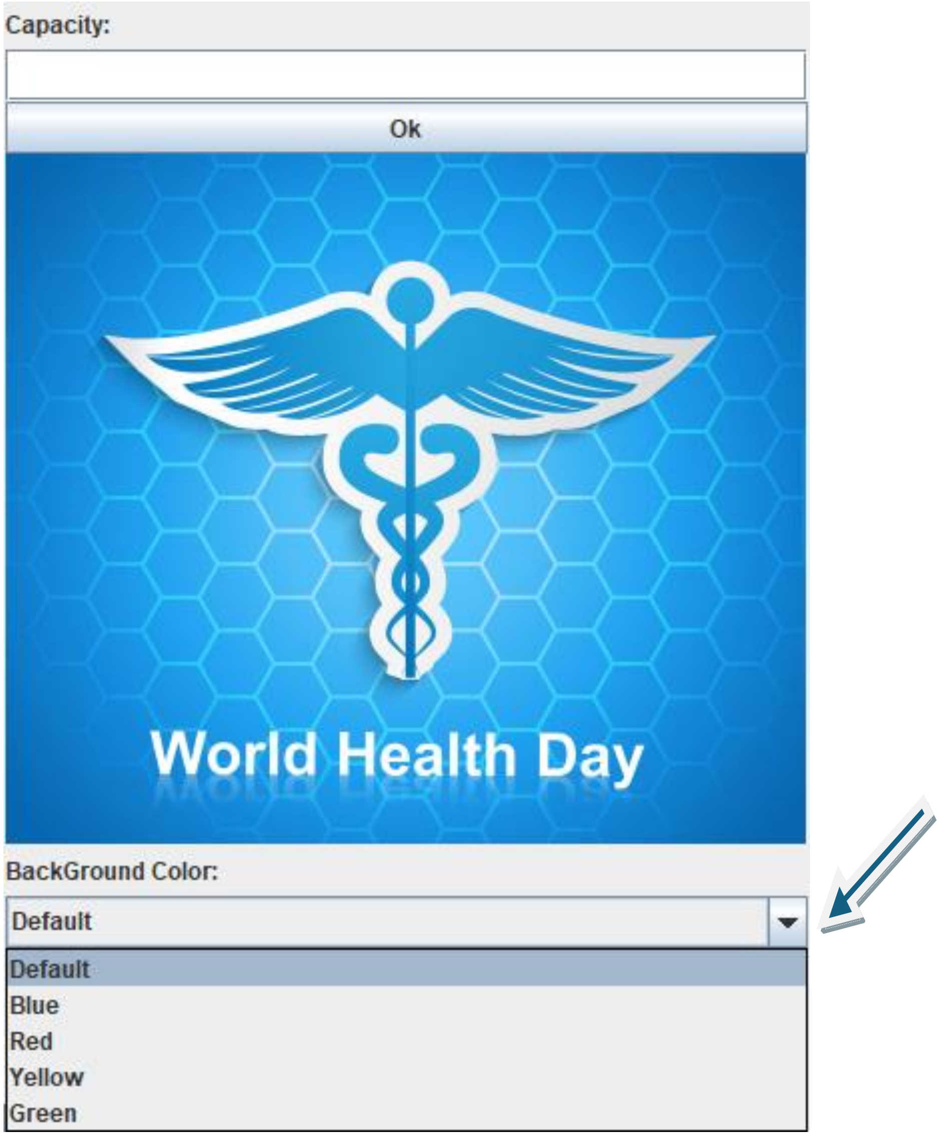
  Capacity: 20

  BackGround Color: Default

- 

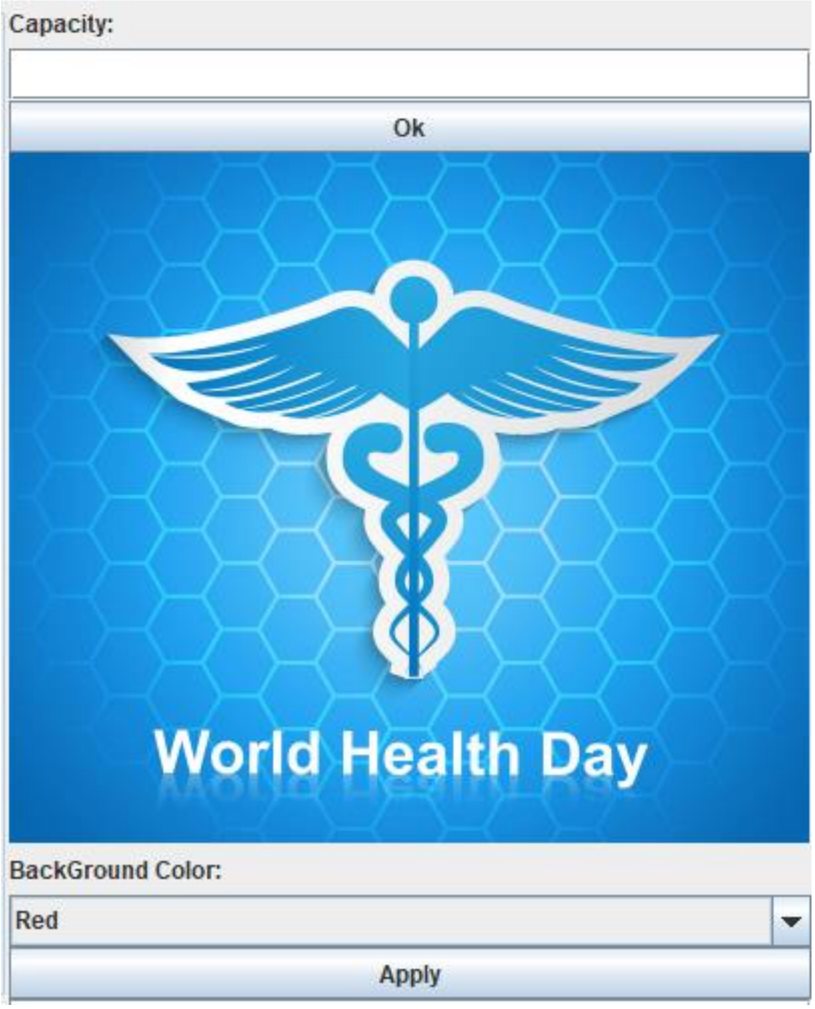
  CAPACITY SETTER

  Pharmacy capacity is set to: 20

- 

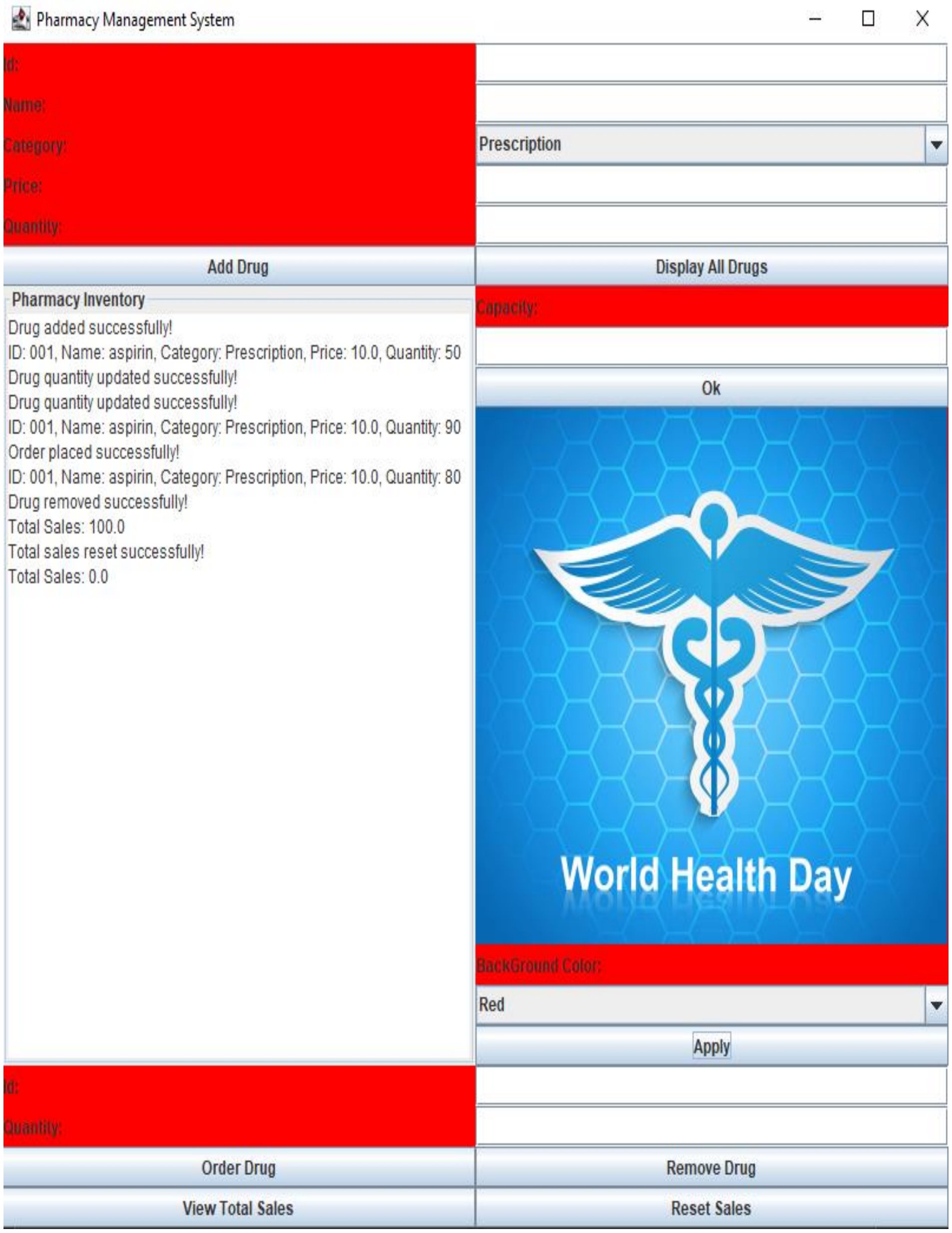
Capacity:

BackGround Color: Default

- 

Capacity:

BackGround Color: Red

- 

Id:

Name:

Category: Prescription

Price:

Quantity:

```
Drug added successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50
Drug quantity updated successfully!
Drug quantity updated successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 90
Order placed successfully!
ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 80
Drug removed successfully!
Total Sales: 100.0
Total sales reset successfully!
Total Sales: 0.0
```

BackGround Color: Red

Id:

Quantity:

Sample Outputs

1. **Adding a Drug**:
   - Input: ID: 001, Name: aspirin, Category: Prescription, Price: 10.0, Quantity: 50.
   - Output: "Drug added successfully!"
2. **Order Placement**:
   - Input: ID: 001, Quantity: 10.
   - Output: "Order placed successfully!"
3. **Error Handling**:
   - Invalid Input: Displays an error dialog for incorrect data entries.

Conclusion

The Pharmacy Management System effectively streamlines pharmacy operations by integrating inventory management, sales tracking, and user-friendly GUI features. It demonstrates robust error handling and provides opportunities for customization.

Future Suggestions

1. **Database Integration**: Enhance data persistence by integrating a database to store drug inventory and sales data.
2. **Multi-User Support**: Implement user roles and permissions for pharmacists, managers, and administrators.
3. **Barcode Scanning**: Add support for barcode scanners to streamline drug identification and sales.
4. **Analytics Dashboard**: Include visual analytics to monitor sales trends and inventory status.
5. **Cloud Backup**: Enable automatic cloud-based backups for improved data security and recovery.