**Pharmacy Management System**

**Name: Shahd Rafat Gamil ID:221001378**

**Name: Hanin Fathy Ramsis ID:221001780**

**Name: Rana Sheta Gaber ID:221001279**

**Name: Moataz Hazem Nassar ID:221000418**

**D/ Hadeer Ahmed Hosny**

**TA/ Rahma Mohamed Ahmed**

**Lecture group: 5**

**1.Introduction**

The Pharmacy Management System is a Java-based application that allows users to manage a pharmacy's inventory, including adding and removing drugs, placing orders, and viewing total sales. The system includes a graphical user interface (GUI) for easy interaction.

**2.Class Descriptions**

**1.Drug Class**

The **Drug** class represents a pharmaceutical drug with attributes such as name, ID, price, category, and available quantity. There are also the setter and getter functions and ToString function.

2.**Pharmacy Class**

The **Pharmacy** class manages the inventory of drugs and handles operations like adding using addDrug Function, removing using removeDrug function, ordering drugs using placeOrder, getting the total sales of the day using getTotalSales function, searching about any drug using findDrugById function, and displaying the drugs that already available in the pharmacy using displayDrugs function.

**3.DrugNotFoundException Class**

The **DrugNotFoundException** class is a custom exception thrown when a drug with a specific ID cannot be found in the pharmacy.

4.**PharmacyManagementGUI Class**

The **PharmacyManagementGUI** class provides a graphical user interface for managing the pharmacy, allowing users to interact with the **Pharmacy** class methods through buttons and dialog boxes.

**3.Detailed Method Descriptions**

**Drug Class Methods**

* **Drug (String name, int id, double price, String category, int availableQuantity)**
  + Constructor to initialize a **Drug** object with the specified attributes.
* **String getName()**
  + Returns the name of the drug.
* **void setName(String name)**
  + Sets the name of the drug.
* **int getId()**
  + Returns the ID of the drug.
* **void setId(int id)**
  + Sets the ID of the drug.
* **double getPrice()**
  + Returns the price of the drug.
* **void setPrice(double price)**
  + Sets the price of the drug.
* **String getCategory()**
  + Returns the category of the drug.
* **void setCategory(String category)**
  + Sets the category of the drug.
* **int getAvailableQuantity()**
  + Returns the available quantity of the drug.
* **void setAvailableQuantity(int availableQuantity)**
  + Sets the available quantity of the drug.
* **String getDrugInfo()**
  + Returns a formatted string containing the drug's information.

**Pharmacy Class Methods**

* **Pharmacy (int maxCapacity)**
  + Constructor to initialize a **Pharmacy** object with a specified maximum capacity.
* **void addDrug(String name, int id, double price, String category, int availableQuantity) throws Exception**
  + Adds a new drug to the pharmacy. Throws an exception if the maximum capacity is reached.
* **void removeDrug(int id) throws DrugNotFoundException**
  + Removes a drug from the pharmacy by ID. Throws **DrugNotFoundException** if the drug is not found.
* **void placeOrder(int id, int quantity) throws DrugNotFoundException**
  + Place an order for a specified quantity of a drug by ID. Updates the total sales and the available quantity. Throws **DrugNotFoundException** if the drug is not found.
* **double getTotalSales()**
  + Returns the total sales amount.
* **void printDrugAvailability(int id) throws DrugNotFoundException**
  + Prints the availability and price of a drug by ID. Throws **DrugNotFoundException** if the drug is not found.
* **String displayDrugs()**
  + Returns a string listing all drugs in the pharmacy.
* **private Drug findDrugById(int id) throws DrugNotFoundException**
  + Finds a drug by ID and returns the **Drug** object. Throws **DrugNotFoundException** if the drug is not found.

**PharmacyManagementGUI Class Methods**

* **PharmacyManagementGUI(int capacity)**
  + Constructor to initialize the GUI with a specified pharmacy capacity.
* **void actionPerformed(ActionEvent e)**
  + Handles buttons click events to perform actions like adding, removing drugs, placing orders, displaying drugs, and showing total sales.
* **private void displayDrugs()**
  + Displays the list of drugs in the pharmacy.
* **private void getTotalSales()**
  + Shows the total sales amount.
* **public static void main (String[] args)**
  + The main method to run the GUI application. Prompts the user to enter the pharmacy capacity and then launches the GUI.

**Usage Instructions**

1. **Launching the Application:**
   * Run the **main** method in the **PharmacyManagementGUI** class.
   * Enter the pharmacy capacity when prompted.
2. **Adding a Drug:**
   * Click the "Add Drug" button.
   * Enter the drug's name, ID, price, category, and quantity in the dialog boxes.
3. **Removing a Drug:**
   * Click the "Remove Drug" button.
   * Enter the drug ID to remove.
4. **Placing an Order:**
   * Click the "Place Order" button.
   * Enter the drug ID and quantity to order.
5. **Displaying Drugs:**
   * Click the "Display Drugs" button to view the list of all drugs in the pharmacy.
6. **Getting Total Sales:**
   * Click the "Get Total Sales" button to view the total sales amount.
7. **Exiting the Program:**
   * Click the "Exit the program" button to close the application. Confirm the exit when prompted.

By following these instructions, users can efficiently manage the pharmacy's drug inventory and sales using the provided graphical user interface.

There are some screenshots of the program:

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

If the user chose Add Drug, then that will appear to him:

A screenshot of a computer screen

Description automatically generated A screenshot of a computer error

Description automatically generated

A screenshot of a computer screen

Description automatically generated A screenshot of a computer

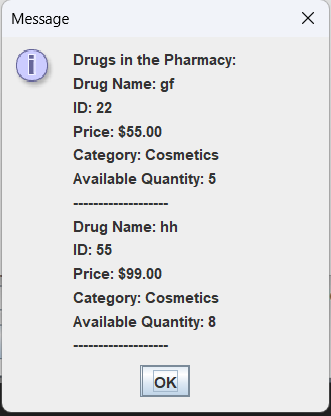
Description automatically generated

A screenshot of a computer error

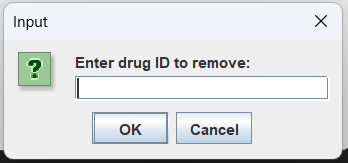
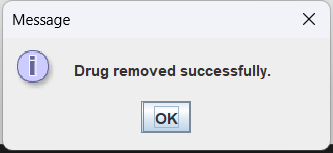
Description automatically generated A screenshot of a computer error

Description automatically generated

If the user chose Display Drugs, then that will appear to him:



If the user chose Remove Drug, then that will appear to him:

If the user chose Place Order, then that will appear to him:

A screenshot of a computer

Description automatically generated A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

If the user, chose Get Total Sales, then that will appear to him:

A screenshot of a computer error

Description automatically generated

If the user, chose Exit the Program, then that will appear to him:

A screenshot of a computer screen

Description automatically generated if the user enters Yes, that will appear:

A screenshot of a computer

Description automatically generated then it will close the program.

if the user enters No, that will return him back to the program.

That is our project.