Name : Vaishali Kale

Email id : kalevaishalir16@gmail.com

Assessment 1:Shopping App

```
package com.wipro.assesment1;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Scanner;

import java.util.Stack;


// ShoppingCart class

class ShoppingCart {

    private LinkedList<String> cart;


    public ShoppingCart() {

        cart = new LinkedList<>();

    }


    // Add an item to the cart

    public void addItem(String item) {

        cart.add(item);

        System.out.println(item + " added to the cart.");

    }


    // Remove an item from the cart

    public void removeItem(String item) {

        if (cart.remove(item)) {

            System.out.println(item + " removed from the cart.");

        } else {

            System.out.println(item + " not found in the cart.");
```

```
        }
    }

    // View all items in the cart
    public void viewCart() {
        if (cart.isEmpty()) {
            System.out.println("The cart is empty.");
        } else {
            System.out.println("Cart contains: " + cart);
        }
    }

    // Get the current cart (for purchase history)
    public LinkedList<String> getCart() {
        return new LinkedList<>(cart);
    }
}

// PurchaseHistory class
class PurchaseHistory {
    private Stack<LinkedList<String>> history;

    public PurchaseHistory() {
        history = new Stack<>();
    }

    // Save the current cart to purchase history
    public void saveCart(LinkedList<String> cart) {
        history.push(new LinkedList<>(cart));
        System.out.println("Cart saved to purchase history.");
    }
```

```
    // Undo the last purchase
    public LinkedList<String> undoLastPurchase() {
        if (!history.isEmpty()) {
            LinkedList<String> lastPurchase = history.pop();
            System.out.println("Last purchase undone: " + lastPurchase);
            return lastPurchase;
        } else {
            System.out.println("No purchases to undo.");
            return new LinkedList<>();
        }
    }

    // View the entire purchase history
    public void viewHistory() {
        if (history.isEmpty()) {
            System.out.println("No purchase history.");
        } else {
            System.out.println("Purchase history: " + history);
        }
    }
}


class CustomerService {
    private Queue<String> serviceRequests;

    public CustomerService() {
        serviceRequests = new LinkedList<>();
    }
```

```
    // Add a customer service request
    public void addRequest(String request) {
        serviceRequests.add(request);
        System.out.println("Customer service request added: " + request);
    }

    // Process the next customer service request
    public void processNextRequest() {
        if (!serviceRequests.isEmpty()) {
            String request = serviceRequests.poll();
            System.out.println("Processing customer service request: " + request);
        } else {
            System.out.println("No customer service requests to process.");
        }
    }

    // View pending customer service requests
    public void viewPendingRequests() {
        if (serviceRequests.isEmpty()) {
            System.out.println("No pending customer service requests.");
        } else {
            System.out.println("Pending customer service requests: " + serviceRequests);
        }
    }
}

// Main class to integrate all the above
public class ShoppingApp {
    public static void main(String[] args) {
        ShoppingCart cart = new ShoppingCart();
        PurchaseHistory history = new PurchaseHistory();
```

```
CustomerService service = new CustomerService();
Scanner scanner = new Scanner(System.in);
int choice;

do {
    System.out.println("1. Add item to cart");
    System.out.println("2. Remove item from cart");
    System.out.println("3. View cart");
    System.out.println("4. Save cart to purchase history");
    System.out.println("5. Undo last purchase");
    System.out.println("6. View purchase history");
    System.out.println("7. Add customer service request");
    System.out.println("8. Process next customer service request");
    System.out.println("9. View pending customer service requests");
    System.out.println("0. Exit");
    System.out.print("\nEnter your choice: ");
    choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {
        case 1:
            System.out.print("Enter item to add: ");
            String addItem = scanner.nextLine();
            cart.addItem(addItem);
            break;
        case 2:
            System.out.print("Enter item to remove: ");
            String removeItem = scanner.nextLine();
            cart.removeItem(removeItem);
            break;
        case 3:
```

```
            cart.viewCart();
            break;
        case 4:
            history.saveCart(cart.getCart());
            break;
        case 5:
            history.undoLastPurchase();
            break;
        case 6:
            history.viewHistory();
            break;
        case 7:
            System.out.print("Enter customer service request: ");
            String request = scanner.nextLine();
            service.addRequest(request);
            break;
        case 8:
            service.processNextRequest();
            break;
        case 9:
            service.viewPendingRequests();
            break;
        case 0:
            System.out.println("Exiting...");
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
} while (choice != 0);

scanner.close();
```

```
    }
}
```

Assessment 2: Library Management System

```
package com.wipro.assesment1;



import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

class Book implements Comparable<Book> {
	private String title;
	private String author;
	private String ISBN;

	public Book(String title, String author, String ISBN) {
		this.title = title;
		this.author = author;
		this.ISBN = ISBN;
	}

	public String getTitle() {
		return title;
	}

	public String getAuthor() {
```

```
        return author;
    }

    public String getISBN() {
        return ISBN;
    }

    @Override
    public int compareTo(Book other) {
        return this.title.compareTo(other.title);
    }

    @Override
    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " + ISBN;
    }
}

class Library {
    private List<Book> books;

    public Library() {
        books = new ArrayList<>();
    }

    // Add a book to the library
    public void addBook(Book book) {
        books.add(book);
        Collections.sort(books);
        System.out.println(book.getTitle() + " added to the library.");
    }
```

```
// Remove a book from the library
public void removeBook(String title) {
	Book toRemove = null;
	for (Book book : books) {
		if (book.getTitle().equalsIgnoreCase(title)) {
			toRemove = book;
			break;
		}
	}
	if (toRemove != null) {
		books.remove(toRemove);
		System.out.println(title + " removed from the library.");
	} else {
		System.out.println(title + " not found in the library.");
	}
}

// Display all books in the library
public void displayBooks() {
	if (books.isEmpty()) {
		System.out.println("The library is empty.");
	} else {
		for (Book book : books) {
			System.out.println(book);
		}
	}
}

// Linear search for a book by title
public Book linearSearch(String title) {
```

```
        for (Book book : books) {
            if (book.getTitle().equalsIgnoreCase(title)) {
                return book;
            }
        }
        return null;
    }

    // Binary search for a book by title
    public Book binarySearch(String title) {
        int left = 0, right = books.size() - 1;
        while (left <= right) {
            int mid = (left + right) / 2;
            Book midBook = books.get(mid);
            int cmp = midBook.getTitle().compareToIgnoreCase(title);
            if (cmp == 0) {
                return midBook;
            } else if (cmp < 0) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }
        return null;
    }
}

public class LibraryManagementSystem {
    public static void main(String[] args) {
        Library library = new Library();
        Scanner scanner = new Scanner(System.in);
```

```
int choice;

do {
        System.out.println("\nLibrary Management System");
        System.out.println("1. Add book to the library");
        System.out.println("2. Remove book from the library");
        System.out.println("3. Display all books");
        System.out.println("4. Search book by title (Linear Search)");
        System.out.println("5. Search book by title (Binary Search)");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
        case 1:
                System.out.print("Enter title: ");
                String title = scanner.nextLine();
                System.out.print("Enter author: ");
                String author = scanner.nextLine();
                System.out.print("Enter ISBN: ");
                String ISBN = scanner.nextLine();
                Book newBook = new Book(title, author, ISBN);
                library.addBook(newBook);
                break;
        case 2:
                System.out.print("Enter title of the book to remove: ");
                String titleToRemove = scanner.nextLine();
                library.removeBook(titleToRemove);
                break;
        case 3:
```

```
                library.displayBooks();
                break;
            case 4:
                System.out.print("Enter title to search (Linear Search): ");
                String titleToSearchLinear = scanner.nextLine();
                Book foundBookLinear = library.linearSearch(titleToSearchLinear);
                if (foundBookLinear != null) {
                    System.out.println("Book found: " + foundBookLinear);
                } else {
                    System.out.println("Book not found.");
                }
                break;
            case 5:
                System.out.print("Enter title to search (Binary Search): ");
                String titleToSearchBinary = scanner.nextLine();
                Book foundBookBinary = library.binarySearch(titleToSearchBinary);
                if (foundBookBinary != null) {
                    System.out.println("Book found: " + foundBookBinary);
                } else {
                    System.out.println("Book not found.");
                }
                break;
            case 0:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 0);

        scanner.close();
```

```
        }
    }
```