Textual Description of Data Structures

Project: Electronic Book Catalog

Group: 5394-RR2

We have used multiple data structures to implement different functionality of our application.

1. Add to Cart: In Add to cart functionality we are using Array List to store our book details specific to user session. The built-in methods of the Array List class allowed us to insert and delete elements from any particular position without having to move or shift any of the other elements. With these methods we also had the ability to remove items by either using the index of the element or by using the string value of the element, allowing for flexibility with minimal code.

Eg.

```
public ArrayList getAllCart()
{
    String SQL = "SELECT * FROM `cart`,`product` WHERE product_id = cart_product_id";
    int count=0;
    ArrayList resultArray = new ArrayList();
    try
    {
        statement = connection.createStatement();
        rs = statement.executeQuery(SQL);
        while(rs.next())
        {
            HashMap results = new HashMap();
            results.put("cart_id",rs.getString("cart_id"));
            results.put("cart_product_id",Integer.parseInt(rs.getString("cart_product_id")));
            results.put("cart_quantity",rs.getString("cart_quantity"));
            results.put("cart_total",rs.getString("cart_total"));
            results.put("cart_price_per_unit",rs.getString("cart_price_per_unit"));
            results.put("product_name",rs.getString("product_name"));
            results.put("product_image",rs.getString("product_image"));
            count++;
            resultArray.add(results);
        }
     }
    catch(Exception e)
    {
        System.out.println("Error is: "+ e);
    }
    return resultArray;
}
```

2. Customer Details: We are using HashMap to fetch data of our customers in O(1) time. HashMap data structures helps to reduce the complexity of code. We have mapped all our customer details respective to their customer id. Considering customerId as a key of a Map and Customer Details as a value that includes customer name, customer address, customer email, etc.

Eg.

```
public HashMap getCustomerDetails(int customer_id)
{
    HashMap results = new HashMap();
    int count=0;
    try
    {
        String SQL = "SELECT * FROM `customer` WHERE customer_id = "+customer_id ;
        statement = connection.createStatement();
        rs = statement.executeQuery(SQL);
        while(rs.next())
        {
            results.put("customer_name",rs.getString("customer_name"));
            results.put("customer_mobile",rs.getString("customer_mobile"));
            results.put("customer_email",rs.getString("customer_email"));
            results.put("customer_password",rs.getString("customer_password"));
            results.put("customer_address",rs.getString("customer_address"));
            results.put("customer_city",rs.getString("customer_city"));
            results.put("customer_state",Integer.parseInt(rs.getString("customer_state")));
            results.put("customer_pincode",rs.getString("customer_pincode"));
            results.put("customer_id",rs.getString("customer_id"));
            count++;
        }
        if(count==0)
        {
            results.put("customer_name","");
            results.put("customer_mobile","");
            results.put("customer_email","");
            results.put("customer_password","");
            results.put("customer_address","");
            results.put("customer_city","");
            results.put("customer_state",0);
            results.put("customer_pincode","");
            results.put("customer_id","");
        }
    }
```

In our project we are mainly using HashMap and Arraylist to store and fetch data from database by Java Database Connectivity.

For Searching Book functionality, we are using JDBC to connect our database MySQL to our code such that a query will run and fetch and insert data in the respective columns of table. For displaying product in our screen, we are internally using Query Language by using “LIKE” keyword as follows:

SQL = "SELECT * FROM `product`,`company`,`type` WHERE product_company_id = company_id AND product_type_id = type_id AND (product_name LIKE '%"+search_text+"%' OR product_isbn LIKE '%"+search_text+"%' OR product_professor LIKE '%"+search_text+"%' OR product_course LIKE '%"+search_text+"%' OR product_subject LIKE '%"+search_text+"%' OR product_author LIKE '%"+search_text+"%')";

```
public ArrayList searchProduct(int typeID, String search_text)
{
    int count=0;
    String SQL = "";
    if(typeID != 0)
        SQL = "SELECT * FROM `product`,`company`,`type` WHERE product_company_id = company_id AND product_type_id = type_id AND type_id = "+typeID;
    else
        SQL = "SELECT * FROM `product`,`company`,`type` WHERE product_company_id = company_id AND product_type_id = type_id AND "
                + "(product_name LIKE '%"+search_text+"%' OR product_isbn LIKE '%"+search_text+"%' OR product_professor LIKE '%"+search_text+"%'"
                        + " OR product_course LIKE '%"+search_text+"%' OR product_subject LIKE '%"+search_text+"%' "
                                + "OR product_author LIKE '%"+search_text+"%')";
    ArrayList resultArray = new ArrayList();
    try
    {
        statement = connection.createStatement();
        rs = statement.executeQuery(SQL);
        while(rs.next())
        {
```