NATIONAL INSTITUTE OF BUSINESS MANAGEMENT

NIBM | POWERING GREAT MINDS

Diploma in Software Engineering 24.2F

Object Oriented Programming

COURSEWORK

PREPARED BY

D.G.C Minidu	KADSE24.2F-004
W.A.S Soysa	KADSE24.2F-002
K.M.H.K Bandara	KADSE24.2F-049
Madusha Rajapaksha	KADSE24.2F-001

PREPARED FOR

Ms. Inoka Abeysinghe (course-director) &
Ms.Nilanka Singhagosha

Abstract

In this Report we will discuss about the usage of the object-oriented programming (OOP) Concepts in Java to develop a standalone general application to address a problem within the daily life Scenario. In this project we will Develop a Java Application to handle the Rice Stock management within Sri Lanka to minimize the shortage and Redundancy of rice varieties for the Customers. The Customers will be able to get the necessary information prevailing to their dire needs on the rice stock. Basic fundamentals from Java GUI and Java swing will be used to make this Project to the required standards. Some of the Significant features that this project will address to configure the Rice stock Management System are:

1.User Authentication:- This system will configure the Login and Registration functionality for the administrators and the users. CAPTCHA can also be implemented to check the logged user is a Bot or not.

2.Rice Stock Management:- Addition, Deletion, Customization can be done within the System for the 25 districts within Sri Lanka. Main Rice Stock Dealers Distribution Cycles can be viewed individually per certain district at any venue.

3.Sales and Purchase Management:- This will record the Sales and the Purchases. This will calculate the total Sales, revenue and the profit of each Rice distributor in Sri Lanka. This will also help the Government Sector to keep track on the Rice distribution Records of the main Rice Distributors in Sri Lanka.

4.Supplier and Customer management:- This will ensure the trust from the customer towards the Supplier .By Managing the records of the Suppliers and Customers a quality service can be ensured to the general Public in Sri Lanka.

5.Reports:- View daily, monthly, or yearly reports on the stock and Transactions.

Key Points

- **Introduction – providing an overview on the Significance of OOP and explanation of the Assignment Objective, Importance of the Standalone Application for Real world Problems.**

- **Problem Identification and Analysis – Identifying a Real-world problem and addressing the issues by making a standalone Application. Identification of real-world problems with objects and analyze the requirements and the functionalities for application. Selection of problem domain and its relevance.**

- Design and Implementation – Design the Structure and the architecture of the standalone application by object-oriented principles like encapsulation, inheritance and polymorphism including classes and objects. Implementation of the java Application. Clearly documenting the design decisions, hierarchies, and relationship among objects.

- Functionality and Testing - Ensuring the developed application meets the specified requirements & Functionalities. Check the correctness, robustness, and user Friendliness. Bugs handling.

- Documentation – Comprehensive report to developed Java Application, including the introduction and the concepts of OOP. Presentation and the Demonstration.

Acknowledgement

- We Acknowledge the Support & the guidance given to us by the lecturers, instructors, peers & others who have contributed to successful completion of the project.

- **Instructors & Mentors**:- Our Project won't be possible without the prior guidance & the encouragement provided to us by the instructors.

- **Peers and Collaborators**: We appreciate the collaborative spirit of our peers. Their insights, brainstorming sessions, and constructive feedback enriched our project. The exchange of ideas fostered a dynamic learning environment.

- **Friends and Family**: Last but not least, we express our gratitude to our friends and family members who patiently listened to our project updates & offered encouragement.

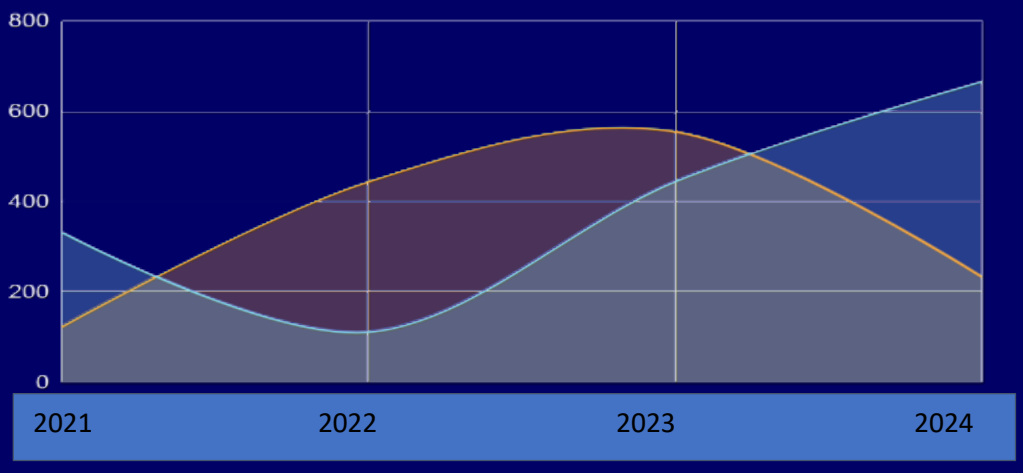
TABLE OF CONTENTS

Introduction

Rice-Stock management in Sri Lanka

Sri Lanka, often referred to as the "Granary of the East," has a long-standing tradition of rice cultivation that forms the backbone of the country's agriculture. Rice is not only the staple food for the majority of the population but also a critical component of the nation's economy. The efficient management of rice stocks is paramount to ensure food security, stabilize market prices, and enhance the livelihoods of farmers. In recent years, the challenges of unpredictable weather patterns, pest infestations, and supply chain disruptions have underscored the need for robust and agile rice stock management systems. The primary objective of this report is to design a Java program to Collect the necessary information related to the rice stocks within Sri Lanka from the rice dealers and provide the acute information of rice stocks for the general public and the government with much ease. Apart from the production of rice within the country itself Sri Lanka make dire imports from other countries as well. Some of the countries are listed below:-

- India – 63%
- Pakistan – 22%
- Thailand – 3.09%
- USA – 5.44%
- Other Asian Countries – 5.44%



Significance of Object-Oriented Programming (OOP) Concepts in Software Development

In our project to develop a sophisticated and scalable software solutions for rice stock management, Object-Oriented Programming (OOP) has emerged as a pivotal interface. OOP principles, including encapsulation, inheritance, polymorphism, and abstraction, provide a structured approach to software development that enhances modularity, maintainability, and reusability. By representing real-world entities, such as warehouses, inventory, and transactions, as objects, OOP facilitates the creation of intuitive and flexible software systems. This report explores the application of OOP concepts in developing a comprehensive rice stock management system tailored to the unique needs of Sri Lanka's agricultural sector. Object oriented programming has several advantages over the procedural programming process since OOP is faster and easier to execute, OOP provides a clear structure for the programs, it helps the Java code to be “DRY” (Don’t Repeat Yourself) and makes the code easier to maintain, modify and to debug, OOP makes it possible to create full reusable applications with less code and shorter development time period.

Importance of the Standalone Applications can be cited as follows: -

Independence from Internet Connectivity

Performance and Speed

Security and Privacy

Customization and Integration

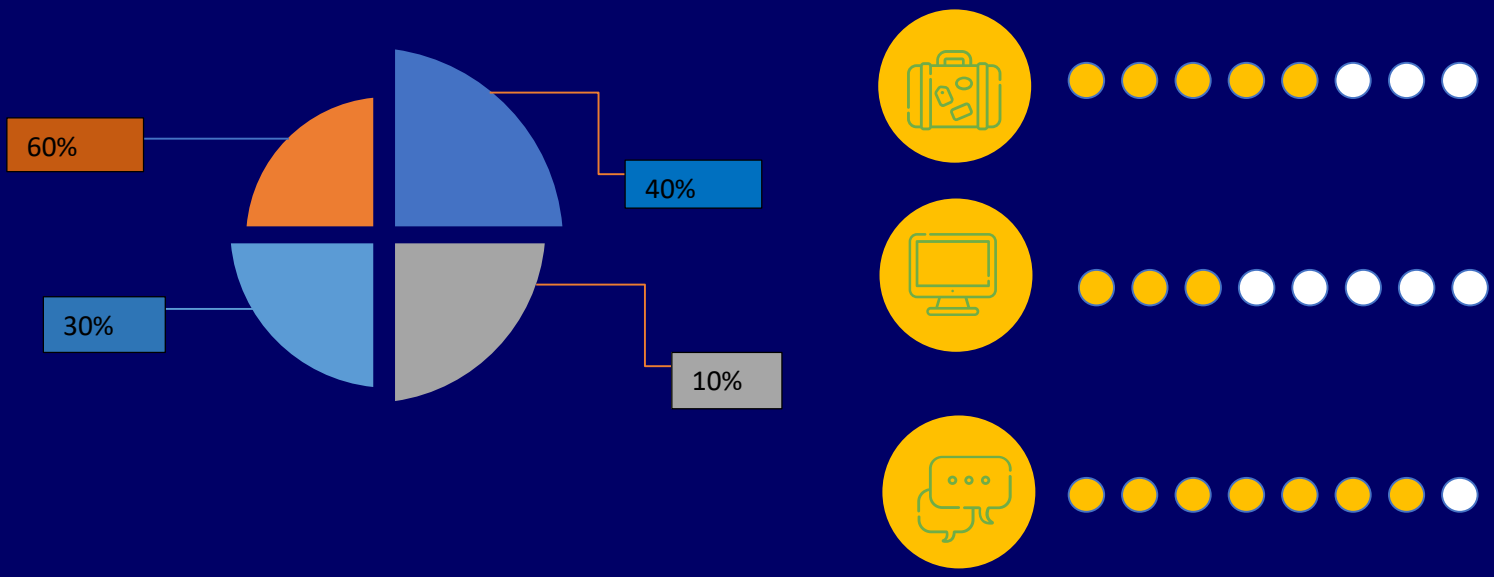
User Experience

Development Flexibility

Offline Availability

Problem Identification and Analysis

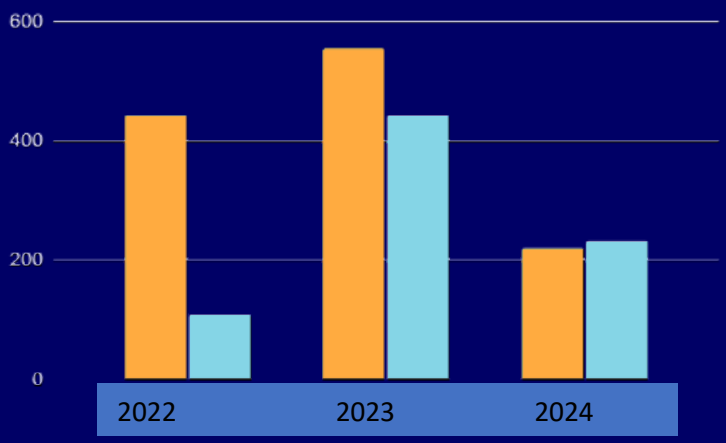
Within the Bygone days a crisis related to the redundancy of rice occurred within the market, some of the rice varieties like “Nadu”, ”Kiri Samba”, “Samba” was not for human consumption with ease. For the primary consumption over 70,000 MT of rice is needed for Sri Lanka as for the National Reports. The Consumption of rice in Sri Lanka has heighted twice the global average per capita in Sri Lanka. Still paradoxically the farmers remain poor and the market remains underdeveloped despite of the consumption. The core has issued lies in the complex and the flawed economic dynamics governing the rice industry. For addressing the above issues, a Project on **RICE STOCK MANAGEMENT IN SRI LANKA** is important.



Real World Objects associated with the Rice Stock Management in Sri Lanka

- **Rice Varieties**: Different types of rice such as Nadu, Kekulu, Samba, and Keeri Samba are commonly cultivated and consumed in Sri Lanka.

- **Paddy Fields**: The agricultural land where rice is grown, including both Maha and Yala season fields.
- **Storage Facilities**: Warehouses and storage units where harvested rice is kept before distribution.
- **Transportation Vehicles**: Trucks, tractors, and other vehicles used to transport rice from farms to storage facilities and markets.
- **Fertilizers and Pesticides**: Chemicals and substances used to enhance rice crop yields and protect them from pests and diseases.
- **Milling Equipment**: Machines used to process harvested rice into consumable grains.
- **Marketplaces**: Local markets and supermarkets where rice is sold to consumers.
- **Government Policies**: Regulations and policies related to rice production, import, and distribution.
- **Weather Conditions**: Rainfall, droughts, and floods that impact rice cultivation and yield.
- **Supply Chain Infrastructure**: The network of logistics and supply chain systems that ensure the smooth flow of rice from farms to consumers



Requirements & Functionalities expected from the Rice Stock Management

Rice Dealers within Sri Lanka who handle the Rice Stock Management in Sri Lanka

- Dudley Sirisena : - Owner of the Araliya Rice Mills, one of the biggest and the most modern rice milling in Asia, he is also a brother of former president Maithripala Sirisena. Besides rice milling, Mr. Sirisena has diversified in to the hospitality Sector with a chain of hotels under the Araliya Hotels brand located within Unawatuna and Nuwara Eliya. Notwithstanding his business success, there have been criticisms and accusations that Dudley has hoarded rice with a view to manipulate the prices within the Market. In September 2021, government authorities raided several large-scale rice mills including the Araliya Rice Mills Seizing Considerable amount of the rice stock reserves amid the concerns of Artificial Shortages and price increments.

✓ Key Attributes:-

- *Owner of the Araliya Rice mills, one of the largest rice producers in Sri Lanka.*
- *Supplies rice to wholesalers and retailers island-wide.*

- Siri Pala Gamlath : - He owns the Nipuna Rice Mills and has been in the rice milling industry since 1984. Siri Pala Gamlath is also the UPFA[United Peoples Freedom Alliance] Member of Parliament who served as a state minister for the Development of the Infrastructure facilities of Settlements and Canals in the Mahaweli Zone. Gamlath, much like Dudley Sirisena, has faced scrutiny over his involvement in the rice industry. Government raids were carried out at his warehouses among others as part of efforts to tackle alleged hoarding and ensure adequate supplies of rice reach the market.

Both men have played influential roles in the Sri Lankan rice market, shaping the modern face of the industry while finding themselves embroiled in controversy over market practices.

✓ Key Attributes:-

- ***Owner of Nipuna Rice Mills, a key player in the industry.***
- ***Has a Significant distribution Network Covering the Urban and the Rural areas.***

- Raja Pushpakumara : - New Rathna Rice Mill (Pvt) Ltd is a Raja Pushpakumara-owned company and is believed to be the biggest rice mill in Sri Lanka. Located in Polonnaruwa, the factory is home to South Asia's largest rice production line. The plant maintains a stock of 50,000 metric tons of various varieties of paddy at any time of the year. Having a milling capacity of 47 metric tons per hour, New Rathna Rice Mill produces around 400 metric tons of rice every day to maintain a continuous supply toward domestic and any export demands that may arise.

✓ Key Attributes:-

- Owner of the New Rathna Rice Mill, one of the Largest Rice Processing Companies in Sri Lanka.

- R.M. Dharmadasa : - R.M. Dharmadasa represents Lak Sahal Rice Mills, which is under the Hingurana Group of Companies Pvt. Ltd. Situated in Nalanda, Central Province, Lak Sahal Rice Mills is one of the major rice varieties produced in the country and has a good reputation for maintaining stringent quality standards in the milling process. The company provides a considerable contribution to the local rice market, catering to consumer needs and supporting the agricultural community. One contributed much to the development in the rice milling industry in Sri Lanka by making this staple available throughout the island, while the other helped in the growth of the agricultural sector.

✓ Key Attributes:-

- Owner of the Lak Sahal Rice Mills, Known for Producing High Quality Rice Varieties.

- The Requirements and the Functionalities for the Rice Stock Management will include a detailed world Map on the retailers and wholesale Distributors island-wide with the details of the respective owners, including their stock exchange.

Relevance of a Rice Stock Management

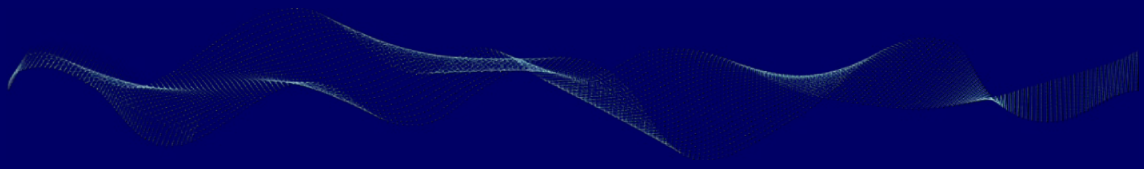
1. **Economic Impact**: Rice is a staple food in Sri Lanka, and effective stock management is crucial for ensuring food security and stabilizing market prices. Poor management can lead to shortages, price hikes, and economic instability.

2. **Agricultural Significance**: Rice cultivation is a major agricultural activity in Sri Lanka, contributing significantly to the rural economy. Efficient stock management can support farmers by ensuring fair prices and reducing post-harvest losses.

3. **Seasonal Challenges**: Sri Lanka faces seasonal variations in rice production due to its reliance on the Maha and Yala seasons. Proper stock management can help bridge the supply gap during off-season periods and mitigate the impact of climate vulnerabilities.

4. **Government Policies**: The government plays a key role in rice stock management through policies on procurement, distribution, and price control. Analyzing these policies can provide insights into their effectiveness and areas for improvement.

5. **Social Impact**: Rice shortages can have severe social consequences, affecting the livelihoods of farmers and the food security of the population. Addressing these issues through better stock management can improve the overall well-being of the community.



Economic Impact

Agricultural Significance

Social Impact

Government Policy

Design And Implementation

Main Application Class

```
Import  javax.swing.*;

Public class  RiceStockManagementSystem {

   public static void main(String[] args) {

      // Create the main application frame

      JFrame frame = new JFrame "Rice Stock Management System");

      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      frame.setSize(1000, 700);

      // Add the main panel

      MainPanel mainPanel = new MainPanel();

      frame.add(mainPanel);


      // Display the frame

      frame.setVisible(true);

   }

}
```

Main Panel Class

```
java

import javax.swing.*;

import java.awt.*;


public class MainPanel extends JPanel {

   public MainPanel() {

      setLayout(new BorderLayout());


      // Create the tabbed pane

      JTabbedPane tabbedPane = new JTabbedPane();
```

```
// Add Stock Management tab
        JPanel stockPanel = new StockPanel();
        tabbedPane.addTab("Stock Management", stockPanel);

        // Add Vendor Management tab
        JPanel vendorPanel = new VendorPanel();
        tabbedPane.addTab("Vendor Management", vendorPanel);

        add(tabbedPane, BorderLayout.CENTER);
    }
}
```

StockPanel Class

java

```
import javax.swing.*;
import java.awt.*;

public class StockPanel extends JPanel {
    public StockPanel() {
        setLayout(new BorderLayout());

        // Create the table for displaying stock data
        String[] columnNames = {"Stock ID", "Rice Type", "Quantity (kg)", "Warehouse Location", "Vendor ID"};
        Object[][] data = {}; // Initially empty
        JTable stockTable = new JTable(data, columnNames);
        JScrollPane scrollPane = new JScrollPane(stockTable);
        add(scrollPane, BorderLayout.CENTER);
        // Create the form for adding/updating stock data
        JPanel formPanel = new JPanel();
        formPanel.setLayout(new GridLayout(6, 2));
        formPanel.add(new JLabel "Stock ID:"));
        JTextField stockIdField = new JTextField();
        formPanel.add(stockIdField);
        formPanel.add(new JLabel "Rice Type:"));
```

```
        JTextField riceTypeField = new JTextField();
        formPanel.add(riceTypeField);
        formPanel.add(new JLabel("Quantity (kg):"));
        JTextField quantityField = new JTextField();
        formPanel.add(quantityField);
        formPanel.add(new JLabel("Warehouse Location:"));
        JTextField locationField = new JTextField();
        formPanel.add(locationField);
        formPanel.add(new JLabel("Vendor ID:"));
        JTextField vendorIdField = new JTextField();
        formPanel.add(vendorIdField);

    // Add buttons for actions
        JButton addButton = new JButton("Add");
        JButton updateButton = new JButton("Update");
        JButton deleteButton = new JButton("Delete");

        formPanel.add(addButton);
        formPanel.add(updateButton);
        formPanel.add(deleteButton);

        add(formPanel, BorderLayout.SOUTH);
    // Action Listeners for buttons (to be implemented)
        addButton.addActionListener(e -> {
            // Code to add vendor data
        });
        updateButton.addActionListener(e -> {
            // Code to update vendor data
        });
        deleteButton.addActionListener(e -> {
            // Code to delete vendor data
        });
    }
`
```

VendorPanel Class

java

```
import javax.swing.*;
import java.awt.*;

public class VendorPanel extends JPanel {
    public VendorPanel() {
        setLayout(new BorderLayout());

        // Create the table for displaying vendor data
        String[] columnNames = {"Vendor ID", "Vendor Name", "Contact Information", "Supplied Rice Types"};
        Object[][] data = {}; // Initially empty
        JTable vendorTable = new JTable(data, columnNames);
        JScrollPane scrollPane = new JScrollPane(vendorTable);
        add(scrollPane, BorderLayout.CENTER);

        // Create the form for adding/updating vendor data
        JPanel formPanel = new JPanel();
        formPanel.setLayout(new GridLayout(5, 2));
        formPanel.add(new JLabel("Vendor ID:"));
        JTextField vendorIdField = new JTextField();
        formPanel.add(vendorIdField);
        formPanel.add(new JLabel("Vendor Name:"));
        JTextField vendorNameField = new JTextField();
        formPanel.add(vendorNameField);
        formPanel.add(new JLabel("Contact Information:"));
        JTextField contactInfoField = new JTextField();
        formPanel.add(contactInfoField);
        formPanel.add(new JLabel("Supplied Rice Types:"));
        JTextField suppliedRiceTypesField = new JTextField();
        formPanel.add(suppliedRiceTypesField);
```

```
        // Add buttons for actions
        JButton addButton = new JButton("Add");
        JButton updateButton = new JButton("Update");
        JButton deleteButton = new JButton("Delete");

        formPanel.add(addButton);
        formPanel.add(updateButton);
        formPanel.add(deleteButton);

        add(formPanel, BorderLayout.SOUTH);

        // Action Listeners for buttons (to be implemented)
        addButton.addActionListener(e -> {
            // Code to add vendor data
        });
        updateButton.addActionListener(e -> {
            // Code to update vendor data
        });
        deleteButton.addActionListener(e -> {
            // Code to delete vendor data
        });
    }
}
```

DatabaseManager Class

java

```
import java.sql.*;

public class DatabaseManager {
    private static final String URL = "jdbc:mysql://localhost:3306/rice_stock_db";
    private static final String USER = "root";
    private static final String PASSWORD = "password";
```

```
public static Connection connect() throws SQLException {
    return DriverManager.getConnection(URL, USER, PASSWORD);
}

// Stock Management Methods
public static void addStock(int stockId, String riceType, int quantity, String location, int vendorId) throws SQLException {
    String query = "INSERT INTO stock (stock_id, rice_type, quantity, location, vendor_id) VALUES (?, ?, ?, ?, ?)";
    try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setInt(1, stockId);
        pstmt.setString(2, riceType);
        pstmt.setInt(3, quantity);
        pstmt.setString(4, location);
        pstmt.setInt(5, vendorId);
        pstmt.executeUpdate();
    }
}

public static void updateStock(int stockId, String riceType, int quantity, String location, int vendorId) throws SQLException {
    String query = "UPDATE stock SET rice_type = ?, quantity = ?, location = ?, vendor_id = ? WHERE stock_id = ?";
    try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setString(1, riceType);
        pstmt.setInt(2, quantity);
        pstmt.setString(3, location);
        pstmt.setInt(4, vendorId);
        pstmt.setInt(5, stockId);
        pstmt.executeUpdate();
    }
}

public static void deleteStock(int stockId) throws SQLException {
    String query = "DELETE FROM stock WHERE stock_id = ?";
    try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
        pstmt.setInt(1, stockId);
        pstmt.executeUpdate();
```

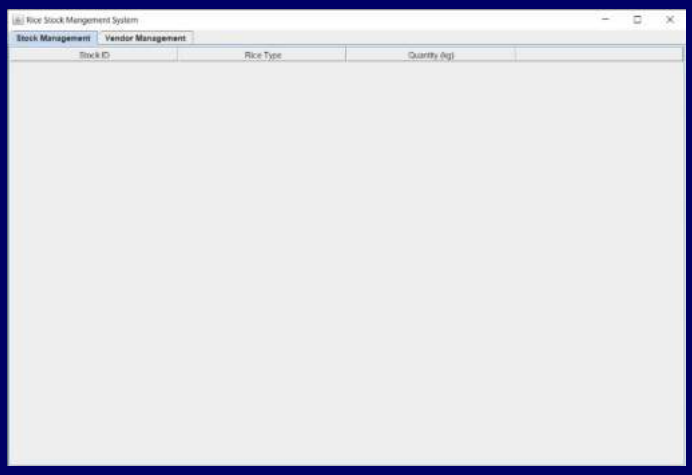
```
    public static                getAllStock() throws SQLException {
        String query = "SELECT * FROM stock";
        Connection conn = connect();
        PreparedStatement pstmt = conn.prepareStatement(query);
        return pstmt.executeQuery();
    }

    // Vendor Management Methods
    public static void addVendor(int vendorId, String vendorName, String contactInfo, String suppliedRiceTypes) throws
SQLException {
        String query = "INSERT INTO vendor (vendor_id, vendor_name, contact_info, supplied_rice_types) VALUES (?, ?, ?, ?)";
        try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setInt(1, vendorId);
            pstmt.setString(2, vendorName);
            pstmt.setString(3, contactInfo);
            pstmt.setString(4, suppliedRiceTypes);
            pstmt.executeUpdate();
        }
    }
    public static void updateVendor(int vendorId, String vendorName, String contactInfo, String suppliedRiceTypes) throws
SQLException {
        String query = "UPDATE vendor SET vendor_name = ?, contact_info = ?, supplied_rice_types = ? WHERE vendor_id = ?";
        try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setString(1, vendorName);
            pstmt.setString(2, contactInfo);
            pstmt.setString(3, suppliedRiceTypes);
            pstmt.setInt(4, vendorId);
            pstmt.executeUpdate();
        }
    }
    public static void deleteVendor(int vendorId) throws SQLException {
        String query = "DELETE FROM vendor WHERE vendor_id = ?";
        try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setInt(1, vendorId);
            pstmt.executeUpdate();
```

```
public static ResultSet getAllVendors() throws SQLException {
    String query = "SELECT * FROM vendor";
    Connection conn = connect();
    PreparedStatement pstmt = conn.prepareStatement(query);
    return pstmt.executeQuery();
  }
}
```

1. The Key Components within the Following Source code Comprises up with the Vendor Management, Stock Management and the Integration System. Within the Vendor Management process vendor ID, name, contact information, and supplied rice types are given. For the Stock Management Process stock ID, rice type, quantity, and warehouse location are given. Within the Integration procedure the vendors are linked to the rice stock supply.
2. Each class RiceStockManagementSystem, MainPanel, StockPanel, VendorPanel, DatabaseManager Encapsulates own data and methods within this project. The DatabaseManager class **encapsulates** database connection details and methods for stock and vendor management
3. The DatabaseManager class **abstract**s the database operations through methods, hiding the underlying SQL operations from the rest of the application. GUI components are abstracted into separate classes (e.g., MainPanel, StockPanel, VendorPanel), which manage their own user interface elements.
4. The StockPanel and VendorPanel classes **inherit** from JPanel, gaining all functionalities of a JPanel and allowing further customization.
5. **Polymorphism** is seen in the way the StockPanel and VendorPanel classes can be used wherever a JPanel is expected. This allows for dynamic method binding and flexible code design.

Output:-



ER Representation of the Design Implementation

1. **Entities**:
    - **Stock**
        - Stock ID (Primary Key)
        - Rice Type
        - Quantity (kg)
        - Warehouse Location
        - Vendor ID (Foreign Key)
    - **Vendor**
        - Vendor ID (Primary Key)
        - Vendor Name
        - Contact Information
        - Supplied Rice Types
2. **Relationships**:
    - One Vendor supplies multiple Stocks.
    - Each Stock is associated with one Vendor.

[Stock] 1 <--- Supplies --- N [Vendor]

Stock

=====

- Stock ID (PK)
- Rice Type
- Quantity
- Warehouse Location

Vendor

======

- Vendor ID (PK)
- Vendor Name
- Contact Information
- Supplied Rice Types

- **[Stock]** has a many-to-one relationship with **[Vendor]**, meaning a vendor can supply multiple stocks but each stock is associated with one vendor.

References

1. (Fernando, 2024)

Fernando, D., 2024. *Sri Lanka's rice dilemma.* [Online]
Available at: https://www.advocata.org/commentary-archives/2024/11/24https/wwwthemorninglk/articles/46l2aceiakss6rk9xhog#:~:text=High%20rice%20prices%2C%20shortages%20of,is%20twice%20the%20global%20average.
[Accessed 31 December 2024].

2. (Anon., 2024)

Anon., 2024. *mainlogoimage.* [Online]
Available at: https://economynext.com/sri-lanka-to-import-70000-tonnes-of-rice-to-address-shortage-minister-188960/
[Accessed 31 December 2024].

Fernando, D., 2024. *Sri Lanka's rice dilemma.* [Online]