Team Members

Alaa Mohamed Ahmed Yousif – **2100443**

Maryam Hosam Eldien Moustafa Ahmed – **2101043**

Nourhan Mohamed Fawzy – **2101385**

Rana Sameh Salah - **2101463**

Task Management System

Project Introduction

The Task Management System is designed to streamline role-based task management. It integrates advanced software design patterns, a user-friendly graphical user interface (GUI), and a robust database to ensure seamless functionality and scalability.

Technologies Used:

- Programming Language: Java (GUI: Java Swing)
- Database Management: Microsoft SQL Server
- Design Patterns: Factory, Singleton, Proxy, Adapter, Builder

Design Patterns Overview

Factory Pattern
Purpose: Used to create objects without exposing the instantiation logic to the client.
Usage in Project: The RoleFactory class dynamically generates role objects based on the role type.

Classes Involved:

- Role: Interface defining role behavior.
- AdminRole: Represents administrative roles.
- UserRole: Represents general user roles.
- RoleFactory: Factory class to create role instances.

Singleton Pattern
Purpose: Ensures a class has only one instance and provides a global point of access.
Usage in Project: The DatabaseConnection class ensures a single database connection throughout the application.

Classes Involved:

- DatabaseConnection: Singleton class managing database connectivity.

Proxy Pattern
Purpose: Provides a surrogate or placeholder for another object.
Usage in Project: The RoleProxy class controls access to sensitive operations.

Classes Involved:

- Role: Interface defining role operations.
- RealRole: Class representing actual role operations.
- RoleProxy: Proxy class controlling access to RealRole.

Adapter Pattern
Purpose: Bridges the gap between incompatible interfaces.
Usage in Project: The DatabaseAdapter class integrates external systems with the current database structure.

Classes Involved:

- DatabaseAdapter: Adapter class ensuring compatibility between legacy and new systems.

Builder Pattern
Purpose: Simplifies the construction of complex objects.
Usage in Project: The GUIBuilder class constructs GUI components dynamically.

Classes Involved:

- GUIBuilder: Class for building GUI components with optional attributes.

Graphical User Interface (GUI)

The GUI is designed to provide an intuitive user experience with the following features:

- Role Table: Displays roles dynamically fetched from the database.
- Interactive Buttons: Add, update, or delete roles.
- Dialogs: Use JOptionPane for user input.

Key Components:

- RoleManagementGUI: Main class managing GUI operations.
- RoleTableModel: Class extending AbstractTableModel for table data binding.

Database Design

Database Structure
Table Name: roles

- Columns:
  - id (Primary Key, INT)
  - role_name (VARCHAR)
  - description (TEXT)

Sample SQL Queries
Insert Query:

sql

Copy code

```
INSERT INTO roles (role_name, description) VALUES ('Admin', 'Administrator role');
```

Select Query:

sql

Copy code

```sql
SELECT * FROM roles;
```

Challenges and Solutions

Challenges

- Integrating multiple design patterns.
- Ensuring consistent database connectivity.
- Designing a user-friendly interface.

Solutions

- Modular code structure and detailed documentation.
- Singleton pattern for managing database connections.
- Iterative user testing for GUI improvements.

Conclusion

The Task Management System demonstrates effective use of design patterns to achieve scalability and maintainability. The intuitive GUI and robust database structure form a strong foundation for future enhancements, including:

- User Authentication: Adding secure login functionality.
- Real-time Notifications: Informing users of updates.
- Export Options: Exporting role data to CSV or Excel.