

Semester Project

(CLO-5: Develop a GUI based project for a real-world problem in a team environment)

Project Overview:

Develop a **Smart City Resource Management System** that integrates urban services like public transportation, energy grids, and emergency response units through a Java Swing GUI. This system must demonstrate advanced OOP concepts while addressing real-world urban challenges:

1. Inheritance & Abstract Classes:

- Base abstract class CityResource with common attributes (resourceID, location, status) extended by Transport Unit (buses/trains), Power Station, and Emergency Service.
- Abstract method calculateMaintenanceCost() implemented differently (e.g., Transport Unit calculates fuel costs, PowerStation uses energy output rates).

2. Composition & Aggregation:

- A CityZone class composed of ResourceHub objects (e.g., a transportation hub aggregates Bus and Train objects).
- SmartGrid class containing PowerStation arrays and Consumer objects (households/industries) with energy consumption tracking.

3. Interfaces:

- Alertable with sendEmergencyAlert() for resources like Emergency Service (police/fire) and outage-prone PowerStation.
- Reportable with generateUsageReport() for all resources (e.g., Transport Unit reports passenger counts, PowerStation shows energy output).

4. GUI (Swing):

- Interactive dashboard with:
 - Real-time maps using JPanel to display resource locations and statuses (color-coded).
 - CRUD operations for resources via forms (JTextField, JComboBox).
 - Dynamic charts (JFreeChart) showing energy/transport usage trends. (Optional for extra credit)

5. Advanced Features:

- **Generics:** CityRepository<T> to manage heterogeneous resources (transport/energy/emergency).
- **File Handling:** Save/load resource states to JSON using GSON, with exception handling for corrupted files.
- **Polymorphism:** Overridden toString() for all resources to display summaries in GUI tables.
- **Static Members:** Track city-wide metrics like totalEnergyConsumed and emergency ResponseTime.

Complexity Additions:

- **Resource Dependencies:** A PowerStation outage triggers alerts to nearby Emergency Service units.
- **Dynamic Scheduling:** TransportUnit objects adjust routes based on real-time traffic data (simulated via threads).
- **Role-Based Access:** GUI panels vary for admin (full control) vs. public (view-only).

Deliverables:

1. **Code:** Implemented OOP concepts with GUI.
2. **UML Diagrams:** Show inheritance, composition, and interface implementations.
3. **Test Cases:** Validate emergency alerts and polymorphic reports.
4. **Documentation:** Explain smart grid algorithms and thread safety measures.

Evaluation Enhancements:

- **Implementation of OOP Principles (25%):** Correct use of inheritance, polymorphism, interfaces, and composition.
- **Functional Completeness (20%):** All required functionalities are implemented and work as expected.
- **Documentation and Reporting (5%):** Well-documented code and clear project report.
- **Viva (50%):** Individual oral examination will be conducted.