System Overview

The Network Management System is an object-oriented application designed to manage network devices and their configurations. It consists of several classes, each with a specific purpose and responsibility.

System Workflow

**User starts the application:**

Main initializes Network Manager and loads configurations.

**User selects an operation:**

Main routes the operation to Network Manager.

**Operation Handling:**

Device operations use Router Device or Switch Device.

Connections managed by Connected Devices.

VLANs handled by Different Types.

**Feedback:**

Timer Delay provides visual or timing effects.

**Persistence:**

Results are saved to files and logged for future use.

Main Class (Main.java)

Purpose:

Entry point and user interface for the system.

Key Features:

Infinite Loop Menu System:

Displays a menu to interact with the user and execute operations.

Nine Main Operations:

Add Device

Delete Device

Configure Device

Add Connection

Check Connectivity

View Devices

View Configuration

Save Changes

Exit

**Uses Timer Delay for UI Feedback.**

Workflow:

**Initialization:**

Loads existing configuration using Network Manager.

**Menu Handling:**

Displays the menu and handles user input.

**Operation Routing:**

Routes user-selected operations to corresponding methods in Network Manager.

Network Manager Class (NetworkManager.java)

Purpose:

Core management of network devices and connections.

Key Components:

**Connected Devices:** Tracks all device connections.

**Devices List:** Maintains a list of all network devices.

Main Operations:

**Device Management:**

Add, delete, and configure devices.

**Connection Management:**

Handle creation and validation of connections.

**File Operations:**

Save and load configurations.

**Configuration Viewing:**

View details of devices and their configurations.

**Features:**

Logging system for all operations.

Input validation.

File persistence for data storage.

Network Component Class (NetworkComponent.java)

Purpose:

Abstract base class for network devices.

Attributes:

**Component ID:** Unique identifier for the device.

**Component Name:** Name of the device.

**Component Description:** Optional description.

Features:

Two constructors (with and without description).

Complete getter and setter methods.

Equality comparison implementation.

Router Device Class (RouterDevice.java)

Purpose:

Router-specific implementation.

Key Features:

**Port Management System:**

Includes port addition, configuration, and validation.

**Device Reconfiguration:**

Allows updates to router configurations.

**Port List Management:**

Tracks and manages port information.

**Main Operations:**

Add/configure ports.

Reconfigure the router.

Manage port lists.

Switch Device Class (SwitchDevice.java)

Purpose:

Switch-specific implementation.

Key Features:

**VLAN Management:**

Supports up to 5 VLANs and their configurations.

**Connected Devices Tracking:**

Monitors devices connected to the switch.

Main Operations:

Create and manage VLANs.

Track connected devices.

Validate configurations.

Connected Devices Class (ConnectedDevices.java)

Purpose:

Manages connections between devices.

Features:

Bidirectional connection tracking.

Connection validation.

Equality comparison for connections.

Main Operations:

Create connections.

Manage references to connected devices.

Different Types Class (DifferentTypes.java)

Purpose:

Generic type management for network components.

Attributes:

**ID:** Component identifier.

**Name:** Component name.

**Number Of Devices Connected:** Number of connections.

Features:

Full getter and setter methods.

Equality comparison implementation.

Used primarily for VLAN management.

Timer Delay Class (TimerDelay.java)

Purpose:

Manages UI timing and feedback.

Features:

**1 second delay:** introduceDelay1Secs().

**3 second delay:** introduceDelay3Secs().

**ASCII Art Display:** hehehe().

Use Cases:

UI timing effects.

Loading animations.

Delayed error messages.