

```

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
import java.util.*;
import javax.swing.*;

import lib.*;

public class Server extends JFrame {
    private ArrayList<Process> activeProcesses;
    private ModuleManager manager;

    /*
    * Default constructor for this class.
    */
    public Server()
    {
        // Initialize the module manager and load the modules
        this.manager = new ModuleManager();

        // Holder for the currently active processes
        this.activeProcesses = new ArrayList<Process>();

        this.initUI();
    } // end constructor

    /**
    * Creates the GUI for the server to interface with the user.
    */
    private void initUI() {
        JMenuBar _menuBar = new JMenuBar();

        // Server actions
        JMenu _server = new JMenu("Server");
        JMenuItem _startServer = new JMenuItem("Start Server", new ImageIcon("lib/accept.png"));
        JMenuItem _stopServer = new JMenuItem("Stop Server", new ImageIcon("lib/stop.png"));
        JMenuItem _restartServer = new JMenuItem("Restart Server", new ImageIcon("lib/arrow_refresh.png"));
        JMenuItem _exit = new JMenuItem("Exit Program");

        // Menu action listeners
        _startServer.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                startup();
            } // end method actionPerformed
        });

        _stopServer.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                shutdown();
            } // end method actionPerformed
        });

        _restartServer.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                restart();
            } // end method actionPerformed
        });

        _exit.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                shutdown();
                System.exit(0);
            } // end method actionPerformed
        });

        _server.add(_startServer);
        _server.add(_stopServer);
    }

```

```

_server.add(_restartServer);
_server.addSeparator();
_server.add(_exit);

JMenu _modules = new JMenu("Modules");
JMenuItem _manageModules = new JMenuItem("Management");
_modules.add(_manageModules);

_manageModules.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        // Add through new module addition
        manager.getManageModuleDialog();
    } // end method actionPerformed
});

JMenuItem _help = new JMenuItem("Help");
_help.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            Desktop.getDesktop().browse(new URI("https://code.google.com/p/spring-2013-se2-dijkstra
"));
        } catch (Exception e) {
            e.printStackTrace();
        } // end try/catch
    } // end method actionPerformed
});

_menuBar.add(_server);
_menuBar.add(_modules);
_menuBar.add(_help);

JPanel _layoutPanel = new JPanel(new BorderLayout());
JScrollPane scroller = new JScrollPane(ServerConsole.console);
_layoutPanel.add(scroller, BorderLayout.CENTER);

this.setJMenuBar(_menuBar);
this.add(_layoutPanel);
this.setBounds(0, 0, 600, 300);
this.setVisible(true);

// When the frame is exited, we need to shut down the services
this.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        shutdown();
        System.exit(0);
    } // end method windowClosing
});
} // end method initUI

/**
 * Starts all the processes.
 */
public void startup() {
    ArrayList<Module> modules = this.manager.getModules();

    ServerConsole.post("System startup in progress...\n");

    // Cycle through all available modules and start their process
    for(int i = 0; i < modules.size(); i++) {
        try {
            // Process to initiate
            ProcessBuilder pb = new ProcessBuilder("java", modules.get(i).getInvocation());
            // Enable logging for startup file
            File startLog = new File("logs/startup.txt");
            pb.redirectErrorStream(true);
            pb.redirectOutput(ProcessBuilder.Redirect.appendTo(startLog));

            // Start the process and add it to the list of active processes

```

```

        Process p = pb.start();
        activeProcesses.add(p);

        ServerConsole.post("Module Listening: [" + modules.get(i).getName() + "]; Port: [" +
modules.get(i).getPort() + "]\n");
    } catch (Exception e) {
        e.printStackTrace();
    } // end try/catch
} // end for loop
} // end method startup

/**
 * Shuts down all the processes.
 */
public void shutdown() {
    ServerConsole.post("System shutdown in progress...\n");
    for (int i = 0; i < activeProcesses.size(); i++)
    {
        this.activeProcesses.get(i).destroy();
        this.activeProcesses.remove(i);
    } // end for loop
} // end method shutdown

/**
 * Restarts all the processes.
 */
public void restart() {
    this.shutdown();
    this.startup();
} // end method restart

/**
 * Main entry point for this application.
 */
public static void main(String[] args) {
    new Server();
} // end function main
} // end class Server

```