

SCD-Lab

lab#9

Name: Anas-Altaf

Roll.No: 22F-3639

Codes:

Task-1:

Bug:

```
package org.logging.lab9;

class Bug {
    private String id;
    private String title;
    private String description;
    private String severity;
    private String status;

    public Bug(String id, String title, String description, String severity,
String status) {
        this.id = id;
        this.title = title;
        this.description = description;
        this.severity = severity;
        this.status = status;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getStatus() {
        return status;
    }
}
```

```

    public void setStatus(String status) {
        this.status = status;
    }

    public String getSeverity() {
        return severity;
    }

    public void setSeverity(String severity) {
        this.severity = severity;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}

```

Bugservice:

```

package org.logging.lab9;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class BugService {
    private static final Logger logger = LogManager.getLogger(BugService.class);
    private List<Bug> bugList = new ArrayList<>();

    public void logBug(String title, String description, String severity) {
        if (title.isEmpty() || description.isEmpty() || severity == null) {
            logger.warn("Invalid bug entry: Missing fields. Title: {},
Description: {}", title, description);
            throw new IllegalArgumentException("All fields must be filled.");
        }
    }
}

```

```

    }

    Bug bug = new Bug(generateBugId(), title, description, severity,
"Open");
    bugList.add(bug);
    logger.info("New bug created: [Bug ID: {}, Title: {}, Severity: {}]",
bug.getId(), title, severity);
    }

    public List<Bug> getAllBugs() {
        return bugList;
    }

    public List<Bug> searchBugs(String query) {
        logger.debug("Search initiated with query: {}", query);
        List<Bug> results = bugList.stream()
            .filter(bug ->
bug.getTitle().toLowerCase().contains(query.toLowerCase()) ||
bug.getDescription().toLowerCase().contains(query.toLowerCase()))
            .collect(Collectors.toList());

        logger.info("Search completed. {} bugs found.", results.size());
        return results;
    }

    private String generateBugId() {
        return "BUG" + (bugList.size() + 1);
    }
}

```

BugTableModel:

```

package org.logging.lab9;

import javax.swing.table.AbstractTableModel;
import java.util.ArrayList;
import java.util.List;

public class BugTableModel extends AbstractTableModel {
    private List<Bug> bugs;
    private String[] columnNames = {"Bug ID", "Title", "Description",
"Severity", "Status"};

    public BugTableModel() {
        this.bugs = new ArrayList<>();
    }

    @Override

```

```

    public int getRowCount() {
        return bugs.size();
    }

    @Override
    public int getColumnCount() {
        return columnNames.length;
    }

    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {
        Bug bug = bugs.get(rowIndex);
        switch (columnIndex) {
            case 0:
                return bug.getId();
            case 1:
                return bug.getTitle();
            case 2:
                return bug.getDescription();
            case 3:
                return bug.getSeverity();
            case 4:
                return bug.getStatus();
            default:
                return null;
        }
    }

    @Override
    public String getColumnName(int column) {
        return columnNames[column];
    }

    public void updateData(List<Bug> newBugs) {
        this.bugs = newBugs;
        fireTableDataChanged();
    }
}

```

BugTrackingApp:

```

package org.logging.lab9;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

```

```
import java.awt.event.ActionListener;

public class BugTrackingApp extends JFrame {
    private static final Logger logger =
        LogManager.getLogger(BugTrackingApp.class);

    private JTextField titleField;
    private JTextArea descriptionArea;
    private JComboBox<String> severityCombo;
    private JTable bugTable;
    private BugService bugService;

    public BugTrackingApp() {
        bugService = new BugService();
        createUI();
    }

    public static void main(String[] args) {

        BugTrackingApp app = new BugTrackingApp();
        app.setVisible(true);

    }

    private void createUI() {
        setTitle("Bug Tracking System");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        JPanel logPanel = new JPanel();
        logPanel.setLayout(new GridLayout(4, 2));

        logPanel.add(new JLabel("Title:"));
        titleField = new JTextField();
        logPanel.add(titleField);

        logPanel.add(new JLabel("Description:"));
        descriptionArea = new JTextArea(3, 20);
        logPanel.add(new JScrollPane(descriptionArea));

        logPanel.add(new JLabel("Severity:"));
        severityCombo = new JComboBox<>(new String[]{"Info", "Debug", "Warn",
"Error"});
        logPanel.add(severityCombo);

        JButton logButton = new JButton("Log Bug");
        logButton.addActionListener(new LogBugAction());
    }
}
```

```

logPanel.add(logButton);

add(logPanel, BorderLayout.NORTH);

bugTable = new JTable(new BugTableModel());
add(new JScrollPane(bugTable), BorderLayout.CENTER);

JPanel searchPanel = new JPanel();
JTextField searchField = new JTextField(15);
JButton searchButton = new JButton("Search");

searchButton.addActionListener(e -> {
    String query = searchField.getText();
    bugService.searchBugs(query);
});

searchPanel.add(new JLabel("Search:"));
searchPanel.add(searchField);
searchPanel.add(searchButton);

add(searchPanel, BorderLayout.SOUTH);
}

private class LogBugAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String title = titleField.getText();
        String description = descriptionArea.getText();
        String severity = (String) severityCombo.getSelectedItem();

        try {
            bugService.logBug(title, description, severity);

            ((BugTableModel)
bugTable.getModel()).updateData(bugService.getAllBugs());
        } catch (IllegalArgumentException ex) {
            JOptionPane.showMessageDialog(BugTrackingApp.this,
ex.getMessage(), "Input Error", JOptionPane.WARNING_MESSAGE);
        }
    }
}
}

```

Log4j2.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="ALL">

```

```

<Appenders>
  <Console name="Console" target="SYSTEM_OUT">
    <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%t] %-5level
%logger{36} - %msg%n"/>
  </Console>
  <File name="FileLogger" fileName="bug-tracking.log" append="true">
    <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%t] %-5level
%logger{36} - %msg%n"/>
  </File>
</Appenders>
<Loggers>
  <Root level="debug">
    <AppenderRef ref="Console"/>
    <AppenderRef ref="FileLogger"/>
  </Root>
</Loggers>
</Configuration>

```

Output:

GUI:

Bug Tracking System

Title: Error2

Description: A grammatical Error

Severity: Debug

Log Bug

Bug ID	Title	Description	Severity	Status
BUG1	Abc	Thei s Alphabetic...	Debug	Open
BUG2	Error2	A grammatical Er...	Debug	Open

Search: **Search**

2024-10-17T06:16:16.297253400Z main DEBUG LoggerContext[name=5a07e868, org.apache.logging.log4j.core.LoggerContext@43c1b556] started OK.

file:

1	2024-10-17 11:13:44	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- New bug created: [Bug ID: BU61, Title: Abc, Severity: Error]
2	2024-10-17 11:15:14	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- New bug created: [Bug ID: BU61, Title: Abc, Severity: Debug]
3	2024-10-17 11:15:32	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- New bug created: [Bug ID: BU62, Title: Error2, Severity: Debug]
4	2024-10-17 11:17:06	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- New bug created: [Bug ID: BU61, Title: Abcd, Severity: Error]
5	2024-10-17 11:17:13	[AWT-EventQueue-0]	DEBUG	org.logging.lab9.BugService	- Search initiated with query: Abcd
6	2024-10-17 11:17:13	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- Search completed. 1 bugs found.
7					

Console:

2024-10-17 11:17:06	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- New bug created: [Bug ID: BU61, Title: Abcd, Severity: Error]
2024-10-17 11:17:13	[AWT-EventQueue-0]	DEBUG	org.logging.lab9.BugService	- Search initiated with query: Abcd
2024-10-17 11:17:13	[AWT-EventQueue-0]	INFO	org.logging.lab9.BugService	- Search completed. 1 bugs found.