

**PROJECT REPORT**  
**ON**  
**TEACHER FEEDBACK FORM**

**SUBMITTED BY**

**Ms. SUVARNA D.POTE**

**SUBMITTED TO**

**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**  
**IN PARTIAL FULFILLMENT OF DEGREE**  
**MASTER OF COMPUTER APPLICATION (SEM-II)**

**UNDER THE GUIDANCE OF**

**Dr. SHVETI CHANDAN**

**Through,**



**Sadhu Vaswani Institute of Management**  
**Studies for Girls, Koregaon**  
**Park, Pune-411001**  
**2024-25**



## **DECLARATION BY STUDENT**

**To.**  
**The Director.**  
**SVIMS, Koregoan Park, Pune.**

I undersigned hereby declare that this project titled, **"Student Feedback Form"** written and submitted by me to **SPPU, Pune**, in partial fulfillment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATION (MCA-1)** under the guidance of **Dr. Shveti Chandan**, is my original work. I further declare that to the best of my knowledge and belief, this project has not been submitted to this or any other University or Institution for the award of any Degree.

**Place: Pune**  
**Date :22/04/2025**

**YOUR NAME & SIGN**  
**(Suvarna D.Pote)**



## ACKNOWLEDGMENT

I extend my sincere gratitude to **Dr. B.H.Nanwani, Dr. Neeta Raskar** and **Dr. Shveti Chandan** for allowing me to carry out the study and for their constant encouragement, valuable suggestions and guidance during the research work.

I extend my special thanks to **Dr.Shveti Chandan** for their kind co-operation and inspiration.

I extend my special gratitude to my dearest family members and friends who encouraged and motivated me to complete the project report.

**Place:** Pune  
**Date :**22/04/2025

**YOUR NAME & SIGN**  
**(Suvarna D.Pote)**



# Index

CHAPTER	DETAILS	PAGE NO
<b>1</b>	<b>CHAPTER 1: INTRODUCTION</b> 1.1 Client /Organization Profile 1.2 Need for System 1.3 Scope and Feasibility of Student feedback form 1.4 Operating Environment HW/SW Requirement 1.5 Teqnology used	<b>1</b>
<b>2</b>	<b>CHAPTER 2: PROPOSED SYSTEM</b> 2.4 Proposed System 2.5 Objectives of System 2.6 User Requirements	<b>3</b>
<b>3</b>	<b>CHAPTER 3: ANALYSIS &amp; DESIGN</b> 3.4 DFD 3.5 Table specifications(Database) 3.6 ERD 3.7 Object Diagram 3.8 Class Diagram 3.9 Use Case Diagrams 3.10 Web Site Map Diagram.	<b>4</b>

<b>4</b>	<b>CHAPTER 4:</b> <b>USER MANUAL</b> 4.4 User Interface Design (Screens etc.) 4.5 Limitations 4.6 Future enhancement	<b>9</b>
<b>5</b>	<b>BIBLIOGRAPHY</b>	<b>10</b>
<b>6</b>	<b>ANNEXURE:</b> Sample program code	<b>11</b>

# **CHAPTER 1: INTRODUCTION**

## **1.1 Client / Organization Profile**

This project is developed as part of an academic mini project for [Your College/Institute Name], under the Computer Science/IT department. The purpose of the project is to provide a digital and efficient solution to collect feedback from students regarding their courses, faculty, and facilities.

## **1.2 Need for System**

Manual feedback collection is time-consuming, prone to errors, and lacks proper analytics. A digital feedback form addresses these issues by ensuring accuracy, saving time, and providing structured data for analysis.

## **1.3 Scope and Feasibility of Feedback Form for Students**

This system is designed to gather feedback from students through a user-friendly interface. It is feasible for small to mid-sized educational institutions. The project scope includes data collection, basic validation, and storing responses in a local file or database.

## **1.4 Operating Environment (Hardware/Software Requirements)**

### **Hardware:**

**Processor:** Intel i7

**RAM:** 4GB

**Storage:** 100 MB

### **Software:**

**Operating System:** Windows

JDK

IntelliJ IDEA

Java Swing for GUI

File System or SQLite (optional) for data storage

### **1.5 Technology Used**

Programming Language: Java

GUI Framework: Java Swing

Development Tool: IntelliJ IDEA

Data Handling: File I/O

Version Control: Git



## **CHAPTER 2: PROPOSED SYSTEM**

### **2.4 Proposed System**

The proposed system is a Java-based application where students fill a feedback form regarding subjects, teaching quality, and suggestions. The data is saved securely for further analysis.

### **2.5 Objectives of the System**

Provide a user-friendly interface for students

Collect accurate and structured feedback

Save feedback in a persistent format (text file/database)

Minimize manual work for faculty and admin

### **2.6 User Requirements**

Basic computer usage skills

Java Runtime Environment installed

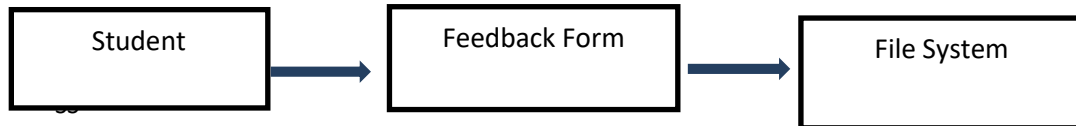
Permissions to read/write files on system

## CHAPTER 3: ANALYSIS & DESIGN

### 3.4 DFD (Data Flow Diagram)

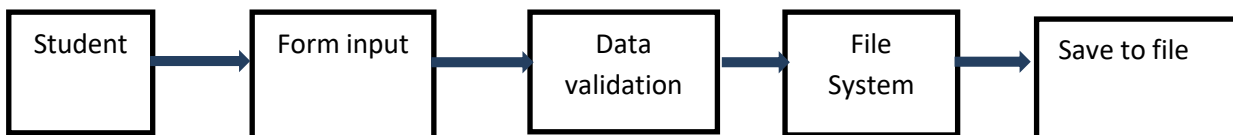
Level 0:

Student → Feedback Form → File System



Level 1:

Student → Form Inputs → Data Validation → Save to File



### 3.5 Table Specifications (Database)

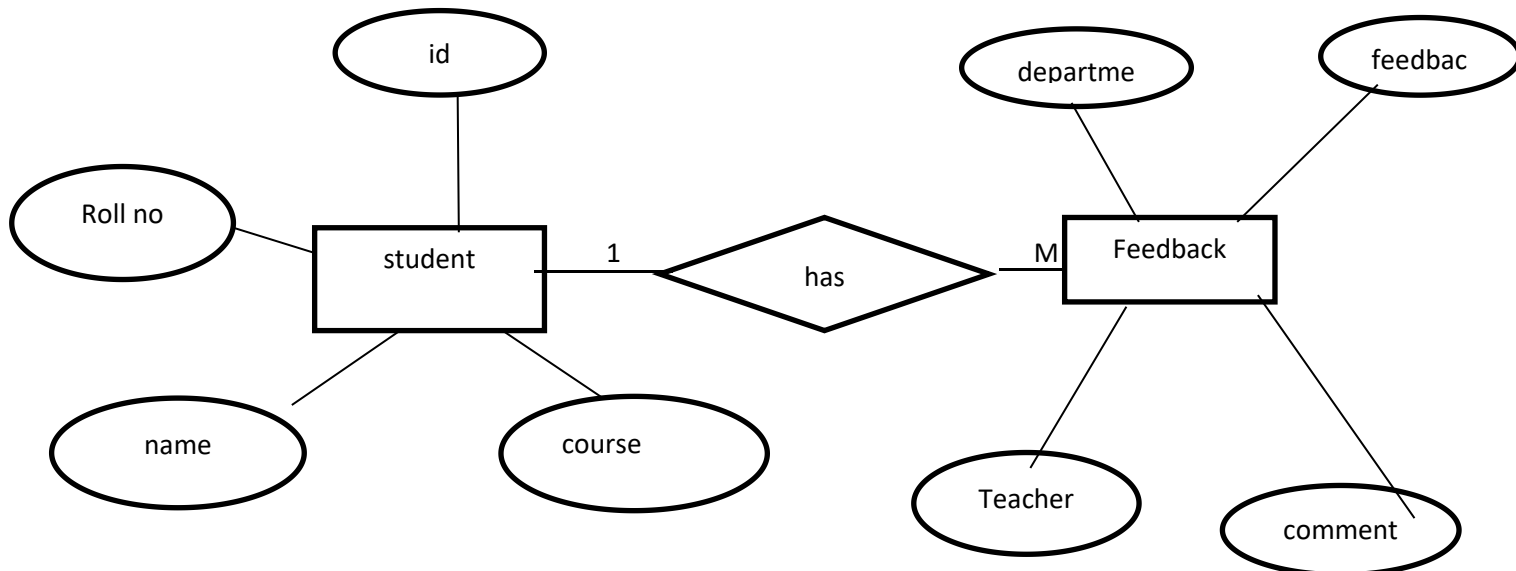
Field Name	Data Type
Name	TEXT
Student Name	TEXT
Roll Number	UNIQUE STUDENT ID
Branch	TEXT
Department	TEXT
Subject	TEXT
Suggestion	TEXT

### 3.6 ERD (Entity Relationship Diagram)

**Entities:** Student, Feedback

**Relationships:**

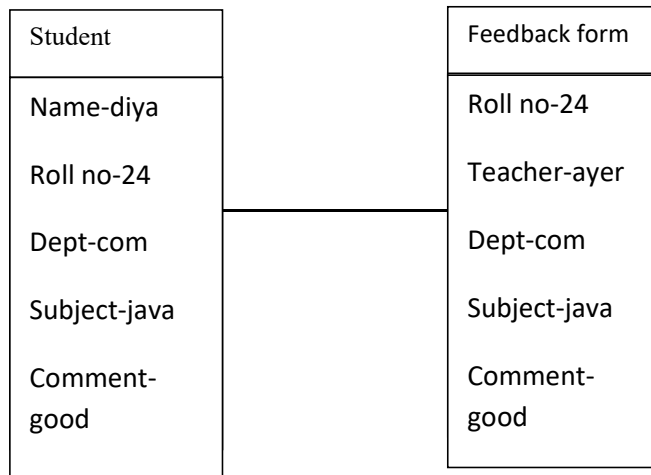
One Student can submit multiple Feedback entries



### 3.7 Object Diagram

#### Objects:

StudentForm, Feedback



### 3.8 Class Diagram

#### Classes:

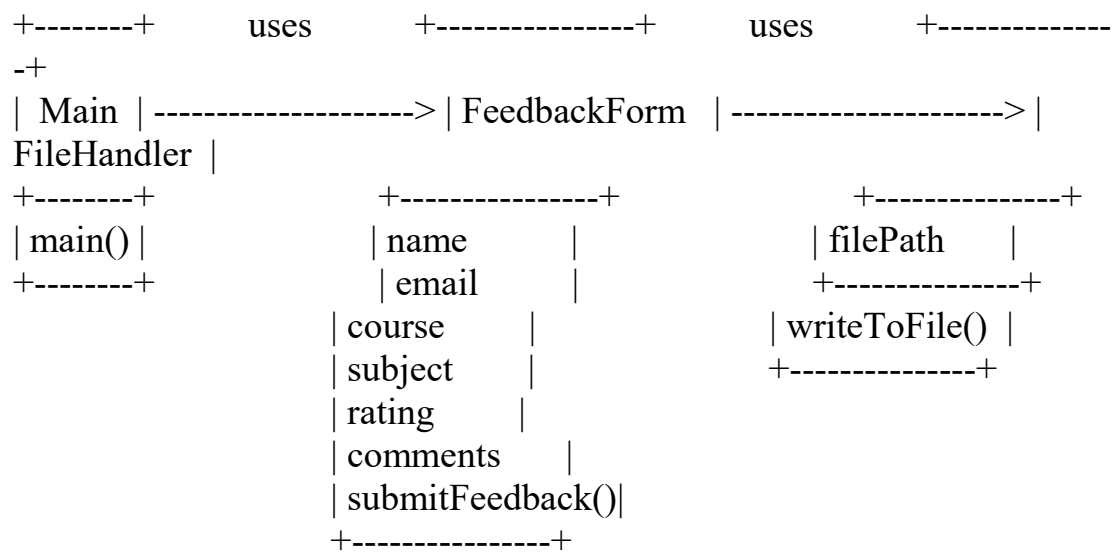
FeedbackForm

Main

FileHandler

#### Relationships:

Main invokes FeedbackForm, which uses FileHandler to save data



### 3.9 Use Case Diagrams

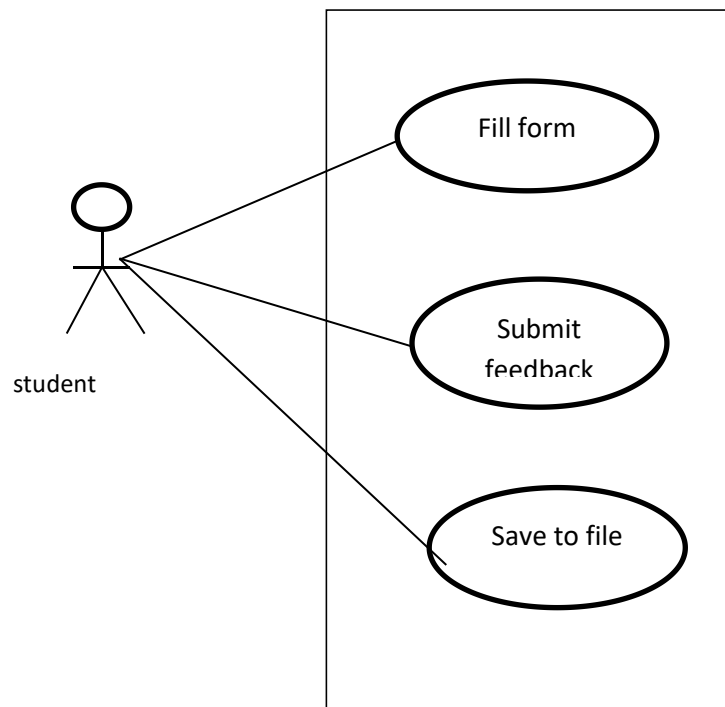
Actors: Student

Use Cases:

Fill Form

Submit Feedback

Save to File



### 3.10 Website Map Diagram

Not applicable (since this is a desktop application). Alternatively, provide a GUI flow:

Welcome → Fill Form → Submit → Confirmation

## **CHAPTER 4: USER MANUAL**

### **4.4 User Interface Design (Screens etc.)**

The UI contains:

Text Fields: Name, Roll Number

Dropdowns: Branch, Subject

Radio Buttons: Teaching quality (1 to 5)

Text Area: Suggestions

Buttons: Submit, Clear

### **4.5 Limitations**

No backend database used (file-based storage)

No admin login or analytics view

Not suitable for large-scale deployments

### **4.6 Future Enhancements**

Add admin dashboard to view/export reports

Implement database support (MySQL or SQLite)

Email confirmation or notifications

Enhance UI with JavaFX or web-based version

## **BIBLIOGRAPHY**

Java: The Complete Reference by Herbert Schildt

Oracle Java Documentation

Java Swing Tutorials – Oracle & GeeksforGeeks

IntelliJ IDEA Documentation

GitHub and Stack Overflow for code examples



## ANNEXURE: Sample Program Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
```

```
class StudentFeedbackForm {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new FeedbackForm());
    }
}
```

```
class FeedbackForm extends JFrame implements ActionListener {
    JLabel title, nameLabel, rollLabel, deptLabel, yearLabel, facultyLabel,
    feedbackLabel;
    JTextField nameField, rollField;
    JComboBox<String> deptBox, yearBox, facultyBox;
    JTextArea feedbackArea;
    JButton submitButton;
```

```
FeedbackForm() {
    setTitle("Student Feedback Form");
    setSize(500, 600);
    setLayout(new GridLayout(9, 2, 5, 5));
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
title = new JLabel("Feedback Form", SwingConstants.CENTER);
title.setFont(new Font("Arial", Font.BOLD, 20));
```

```
nameLabel = new JLabel(" Name:");
nameField = new JTextField();
```

```
rollLabel = new JLabel(" Roll No:");
rollField = new JTextField();
```

```
deptLabel = new JLabel(" Department:");
String[] departments = {"Computer Science", "IT", "Electronics", "Mechanical"};
deptBox = new JComboBox<>(departments);
```

```
yearLabel = new JLabel("Year:");
String[] years = {"1st", "2nd", "3rd"};
yearBox = new JComboBox<>(years);
```

```
facultyLabel = new JLabel("Faculty Name:");
String[] faculties = {"Prof. Sharma", "Prof. Mehta", "Prof. Iyer"};
facultyBox = new JComboBox<>(faculties);
```

```
feedbackLabel = new JLabel("Feedback:");
feedbackArea = new JTextArea();
feedbackArea.setRows(4);
feedbackArea.setLineWrap(true);
feedbackArea.setWrapStyleWord(true);
```

```
submitButton = new JButton("Submit");
submitButton.addActionListener(this);
```

```
add(title); add(new JLabel());
add(nameLabel); add(nameField);
add(rollLabel); add(rollField);
add(deptLabel); add(deptBox);
add(yearLabel); add(yearBox);
add(facultyLabel); add(facultyBox);
add(feedbackLabel); add(new JScrollPane(feedbackArea));
add(new JLabel()); add(submitButton);
```

```
setVisible(true);
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    if (e.getSource() == submitButton) {
        String name = nameField.getText();
        String roll = rollField.getText();
        String dept = (String) deptBox.getSelectedItem();
        String year = (String) yearBox.getSelectedItem();
        String faculty = (String) facultyBox.getSelectedItem();
        String feedback = feedbackArea.getText();
```

```
        try (FileWriter fw = new FileWriter("feedback.txt", true);
```

```
BufferedWriter bw = new BufferedWriter(fw);  
PrintWriter out = new PrintWriter(bw) {
```

```
    out.println("Name: " + name);  
    out.println("Roll No: " + roll);  
    out.println("Department: " + dept);  
    out.println("Year: " + year);  
    out.println("Faculty Name: " + faculty);  
    out.println("Feedback: " + feedback);  
    out.println("-----");
```

```
OptionPane.showMessageDialog(this, "Feedback submitted successfully!");
```

```
    nameField.setText("");  
    rollField.setText("");  
    feedbackArea.setText("");
```

```
    } catch (IOException ioException) {  
        JOptionPane.showMessageDialog(this, "Error saving feedback.");  
    }  
}  
}  
}
```

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.io.*;
```

```
class StudentFeedbackForm {  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(() -> new FeedbackForm());  
    }  
}
```

```
class FeedbackForm extends JFrame implements ActionListener {  
    JLabel title, nameLabel, rollLabel, deptLabel, yearLabel, facultyLabel,  
    feedbackLabel;  
    JTextField nameField, rollField;
```

```
JComboBox<String> deptBox, yearBox, facultyBox;  
JTextArea feedbackArea;  
JButton submitButton;
```

```
FeedbackForm() {  
    setTitle("Student Feedback Form");  
    setSize(500, 600);  
    setLayout(new GridLayout(9, 2, 5, 5));  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    title = new JLabel("Feedback Form", SwingConstants.CENTER);  
    title.setFont(new Font("Arial", Font.BOLD, 20));
```

```
    nameLabel = new JLabel(" Name:");  
    nameField = new JTextField();
```

```
    rollLabel = new JLabel(" Roll No:");  
    rollField = new JTextField();
```

```
    deptLabel = new JLabel(" Department:");  
    String[] departments = {"Computer Science", "IT", "Electronics", "Mechanical"};  
    deptBox = new JComboBox<>(departments);
```

```
    yearLabel = new JLabel(" Year:");  
    String[] years = {"1st", "2nd", "3rd"};  
    yearBox = new JComboBox<>(years);
```

```
    facultyLabel = new JLabel("Faculty Name:");  
    String[] faculties = {"Prof. Sharma", "Prof. Mehta", "Prof. Iyer"};  
    facultyBox = new JComboBox<>(faculties);
```

```
    feedbackLabel = new JLabel("Feedback:");  
    feedbackArea = new JTextArea();  
    feedbackArea.setRows(4);  
    feedbackArea.setLineWrap(true);  
    feedbackArea.setWrapStyleWord(true);
```

```
    submitButton = new JButton("Submit");  
    submitButton.addActionListener(this);
```

```
    add(title); add(new JLabel());
```

```

add(nameLabel); add(nameField);
add(rollLabel); add(rollField);
add(deptLabel); add(deptBox);
add(yearLabel); add(yearBox);
add(facultyLabel); add(facultyBox);
add(feedbackLabel); add(new JScrollPane(feedbackArea));
add(new JLabel()); add(submitButton);

```

```

setVisible(true);

```

```

@Override

```

```

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == submitButton) {
        String name = nameField.getText();
        String roll = rollField.getText();
        String dept = (String) deptBox.getSelectedItem();
        String year = (String) yearBox.getSelectedItem();
        String faculty = (String) facultyBox.getSelectedItem();
        String feedback = feedbackArea.getText();
    }
}

```

```

try (FileWriter fw = new FileWriter("feedback.txt", true);
     BufferedWriter bw = new BufferedWriter(fw);
     PrintWriter out = new PrintWriter(bw)) {

```

```

    out.println("Name: " + name);
    out.println("Roll No: " + roll);
    out.println("Department: " + dept);
    out.println("Year: " + year);
    out.println("Faculty Name: " + faculty);
    out.println("Feedback: " + feedback);
    out.println("-----");
}

```

```

JOptionPane.showMessageDialog(this, "Feedback submitted successfully!");

```

```

nameField.setText("");
rollField.setText("");
feedbackArea.setText("");

```

```

} catch (IOException ioException) {
    JOptionPane.showMessageDialog(this, "Error saving feedback.");
}

```

}

}

}

}

Student Name :

Student Roll No. :

Subject Selected : Soft Skill

OverAll Feedback : 1.0

1. Teacher is prepared for class : Strongly Disagree
2. Teacher knows His/Her Subject : Strongly Disagree
3. Teacher is Organised and neat : Strongly Disagree
4. Teacher Plans class time and Assignments : Strongly Disagree
5. Teacher is flexible : Strongly Disagree
6. Teacher is clear about assignments and test : Strongly Disagree
7. Teacher allows you to be active in the classroom : Strongly Disagree
8. Teacher manages the time well : Strongly Disagree
9. Teacher returns homework in a timely manner : Strongly Disagree
10. Teacher grades Fairly : Strongly Disagree
11. I have learned a lot from the teacher about this subject : Strongly Disagree
12. Teacher gives good feedback on projects : Strongly Disagree
13. Teacher is creative in developing lesson plans : Strongly Disagree
14. Teacher encourages students to speak up : Strongly Disagree
15. Teacher follows through on what he/she says : Strongly Disagree
16. Teacher listens student's point of view : Strongly Disagree
17. Teacher respects the opinions and decision of students : Strongly Disagree
18. Teacher accepts responsibility for his/her own mistakes : Strongly Disagree
19. Teacher is willing to learn from students : Strongly Disagree
20. Teacher is sensitive to the needs of students : Strongly Disagree
21. Teacher's words and actions match : Strongly Disagree
22. Teacher is fun to be with : Strongly Disagree
23. Teacher likes and respects students : Strongly Disagree
24. Teacher helps when you ask for help : Strongly Disagree
25. Teacher is consistent and fair in discipline : Strongly Disagree
25. Teacher is consistent and fair in discipline : 26. I trust this teacher : Strongly Disagree
27. Teacher tries to model student's expectations : Strongly Disagree
28. Teacher is firm in discipline without being too strict : Strongly Disagree

Name: sonali

Roll No: 11

Department: Electronics

Year: 2nd

Faculty Name: Prof. Mehta

Feedback: she is the best teacher

-----

Name: arun

Roll No: 11

Department: IT

Year: 3rd

Faculty Name: Prof. Iyer

Feedback: he is best teacher

-----





