Scholarship Portal

SY B.Tech COMPS
Semester III (2024-25)

1601023214	Om Anand Jha
16010123215	Om Bhanushali
16010123217	Om Thanage

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Table of Contents

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Introduction

Objective:

The primary objective of the software application is to provide a user-friendly platform for students to apply for available scholarships, ensuring easy registration and review of their applications.

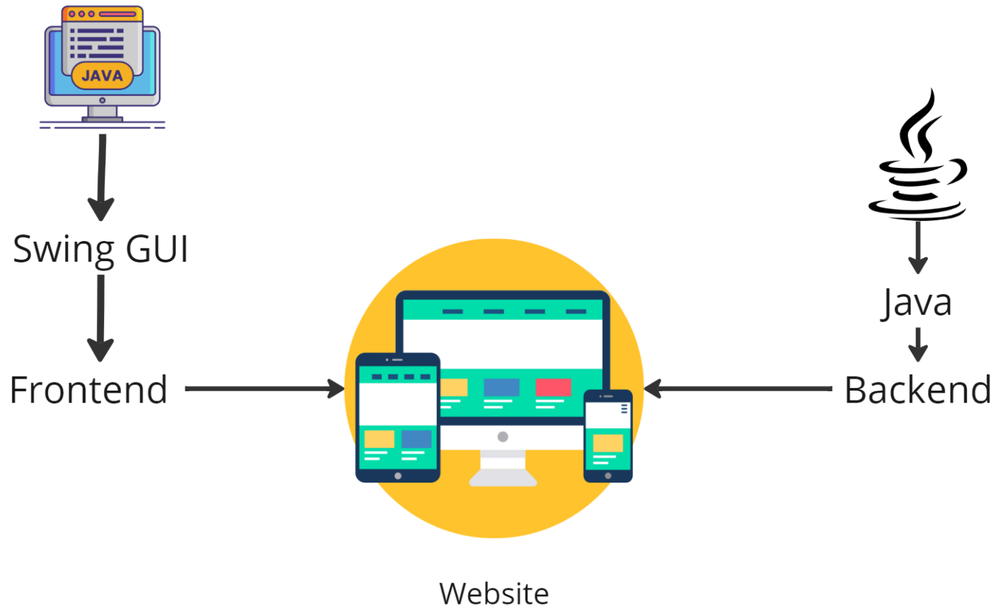
SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Introduction

Tech Stack:

- Programming Language: Java
- Programming paradigm: Object Oriented
- Libraries: Swing (GUI), Collection Framework(ArrayList, Hashmap)
- Tools: JDK



SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

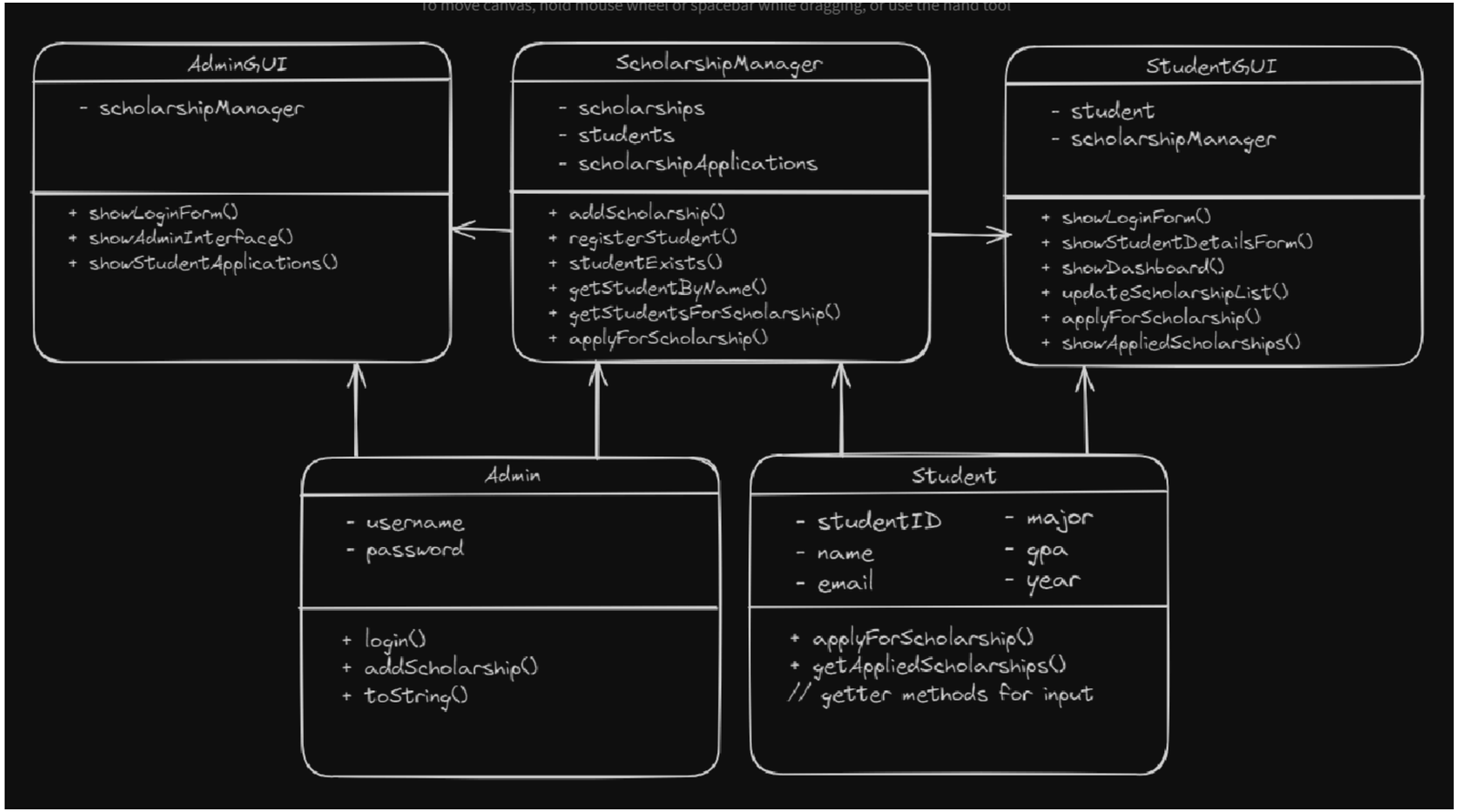
Somaiya TRUST

System Design

Problem Definition:

Many students face challenges in finding and applying for scholarships. This application simplifies the process by enabling students to register, view available scholarships based on their GPA, and apply online. It also allows them to view applied scholarships for easy reference.

Class Diagram



SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Object oriented principles used

Encapsulation:

- In ScholarshipDetail, the fields name, description, amount, and gpaRequirement are private, and public getter methods like getName(), getDescription(), getAmount(), and getGpaRequirement() are provided to access these fields.

```
public class ScholarshipDetail {
    private String name;
    private String description;
    private double amount;
    private double gpaRequirement;

    public ScholarshipDetail(String name, String description, double amount, double gpaRequirement) {
        this.name = name;
        this.description = description;
        this.amount = amount;
        this.gpaRequirement = gpaRequirement;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }

    public double getAmount() {
        return amount;
    }

    public double getGpaRequirement() {
        return gpaRequirement;
    }

    public String toString() {
        return "Scholarship Name: " + name + "\nDescription: " + description +
                "\nAmount: Rs." + amount + "\nGPA Requirement: " + gpaRequirement + "\n";
    }
}
```

Object oriented principles used

Abstraction:

•The ScholarshipManager class abstracts the details of managing scholarships and student applications. Methods like addScholarship(), getScholarships(), and applyForScholarship() provide a simplified interface for these operations.

```
    public void applyForScholarship(Student student, ScholarshipDetail scholarship) {
        if (scholarshipApplications.containsKey(scholarship)) {
            scholarshipApplications.get(scholarship).add(student);
        }
    }

    public List<Student> getStudentsForScholarship(ScholarshipDetail scholarship) {
        return scholarshipApplications.getOrDefault(scholarship, new ArrayList<>());
    }
}
```

Object oriented principles used

Inheritance:

- The AdminGUI and StudentGUI classes use instances of ScholarshipManager to manage scholarships, demonstrating composition rather than inheritance.

```java
public class AdminGUI {
    private ScholarshipManager scholarshipManager;

    public AdminGUI(ScholarshipManager scholarshipManager) {
        this.scholarshipManager = scholarshipManager;
        showLoginForm();
    }
```

```java
public class StudentGUI {
    private ScholarshipManager scholarshipManager;
    private Student student;

    public StudentGUI(ScholarshipManager scholarshipManager) {
        this.scholarshipManager = scholarshipManager;
        showLoginForm();
    }
```

Object oriented principles used

Polymorphism:

The applyForScholarship() method in the Student class calls the applyForScholarship() method in the ScholarshipManager class, demonstrating polymorphism through method calls.

```
public void applyForScholarship(ScholarshipDetail scholarship, ScholarshipManager scholarshipManager) {
    appliedScholarships.add(scholarship);
    scholarshipManager.applyForScholarship(this, scholarship);
}
```

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Thank you!