CPIT-251 | Group Project

[Internship Accessibility and Communication Enhancement]

Group members:

Ahmed Abdulrahman	
Abdulaziz	
Nawaf	
Fahad	

Supervised by Prof. Mohammed AL-Haddad

Table Of Contents

Abstract

The "Internship Accessibility and Communication Enhancement" project aims to address the challenges faced by interns in accessing internship opportunities and effectively communicating within professional environments. The project emphasizes the importance of leveraging technology, enhancing communication strategies, and fostering partnerships between educational institutions and industry professionals to create a more equitable and supportive internship experience. Through a combination of research, case studies, and stakeholder collaboration, this project provides a comprehensive framework for improving internship accessibility and communication, ultimately contributing to better career preparedness and success for students.

Introduction

Internships play a crucial role in bridging the gap between academic learning and professional experience, providing students with valuable opportunities to apply theoretical knowledge in real-world settings. However, not all students have equal access to these opportunities due to various barriers such as geographical constraints, socioeconomic disparities, and communication challenges. These obstacles can limit the ability of talented individuals to secure internships, thereby hindering their professional growth and career development.

This project aims to explore and address the challenges that students face in accessing internships and effectively communicating within professional environments. By analyzing the current landscape and identifying key areas of improvement, the project seeks to propose practical solutions that can make internships more accessible and inclusive for a diverse range of students. The focus will be on leveraging technology, improving communication channels, and fostering collaborations between educational institutions and industry partners to create a more equitable and supportive environment for all interns.

Through comprehensive research, stakeholder engagement, and the development of strategic recommendations, this project aspires to enhance the overall internship experience, ensuring that all students can gain the skills and experience necessary for successful careers.

Problem:

The current internship landscape presents significant hurdles for both students and companies. One of the primary challenges is the lack of accessibility to internship opportunities. Often, students struggle to discover available internships, and even when they are aware of openings, obtaining detailed information about the company and the specific role can be challenging and opaque. Additionally, establishing connections between companies and universities for recommendations from administrators and professors poses a common difficulty. This lack of streamlined communication can lead to missed opportunities for students who possess the necessary skills and knowledge for internships, as companies may overlook them due to incomplete information or insufficient recommendations.

Considering these challenges, there is a compelling opportunity to develop a solution that addresses these pain points. By providing a centralized platform that offers comprehensive information about internship opportunities, facilitates communication between universities and companies, and offers insights into the requirements for specific roles, this system can serve as a valuable resource for both students and companies.

This system aims to bridge the gap between students and companies, empowering students to pursue their dream roles with confidence and enabling companies to identify and recruit the most suitable candidates for their internship programs. Through effective system analysis and design, we can create a solution that revolutionizes the internship experience, fostering meaningful connections and mutually beneficial partnerships.

Proposed Solution:

To address the challenges identified in the internship landscape, we propose the development of a comprehensive internship-finding system. This system will offer a user-friendly platform designed to meet the needs of both students and companies, facilitating efficient internship discovery, communication, and recruitment processes.

1- ***Centralized Internship Database:***
Implement a centralized database of internship opportunities to ensure easy accessibility for students. Include detailed information about each internship, such as company background, job description, requirements, and application deadlines.

2- ***User-friendly Interface:***
Design an intuitive user interface that allows students to search and filter internship listings based on criteria such as location, industry, duration, and required skills. Present relevant information about each internship in a clear and concise manner, enabling students to make informed decisions.

3- ***Communication Tools:***
Integrate communication features that enable seamless interaction between students and companies.
Provide messaging capabilities to facilitate inquiries, application submissions, and follow-up discussions.

4- ***Mobile Compatibility:***
Ensure that the system is compatible with mobile devices, allowing students and companies to access internship listings and communication features on the go.

Approach Used & Why:

In our project, we will adopt Agile, as it offers numerous advantages that align perfectly with our objectives, including:

Flexibility in Requirements: Internship programs often require continuous adjustments based on feedback from interns, mentors, and company stakeholders. Agile's ability to adapt to evolving needs allows for changes to be incorporated without disrupting the entire project plan. This is especially useful when accessibility and communication needs vary across different stakeholders.

Collaboration and Communication: A key tenet of Agile is open and continuous communication within the team. For your project, frequent feedback loops between the internship coordinators, HR departments, and the interns themselves can ensure that communication enhancements are on track and aligned with the actual needs of the stakeholders.

Quick Wins and Feedback Cycles: Instead of waiting for the end of the project to assess success, Agile allows for ongoing feedback after each sprint or iteration. This ensures that any enhancements to accessibility or communication are tested in real-world scenarios early on, allowing for immediate course corrections where needed.

Focus on User Experience: Agile places a strong emphasis on delivering value to the user, which aligns perfectly with your goal of enhancing internship accessibility and communication. Each sprint can be focused on solving specific user experience problems, ensuring a holistic improvement over time.

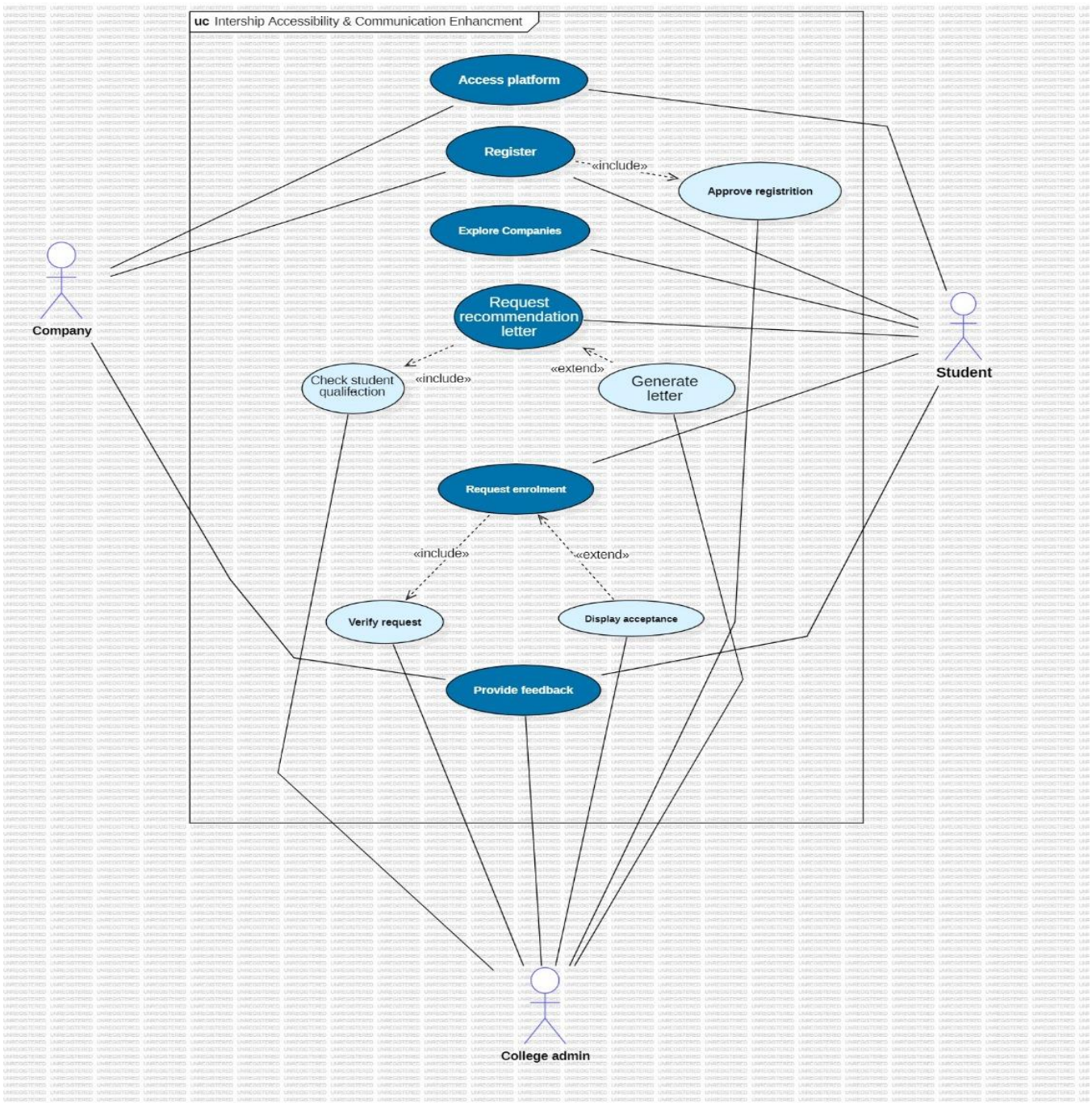
Use case diagram:

- **Actors:**
  - Company
  - Student
  - College admin

- **Use Cases:**
  - Access platform.
  - Register.
  - Explore companies.
  - Request recommendation letter.
  - Request enrollment.
  - Provide feedback.
  - Approve registration.
  - Generate letter.
  - Check student qualification.
  - Verify request.
  - Display acceptance.

- **Relationships:**
  - **Include** approve registration in register:
    After registration, approve and validate register.
  - Request recommendation letter **include** in check student qualification:
    To request recommendation letter, student qualifications must be checked.
  - Request enrollment **include** verify request.
  - **Extend** from request recommendation letter to generate letter.
  - **Extend** from request enrollment to display acceptance.

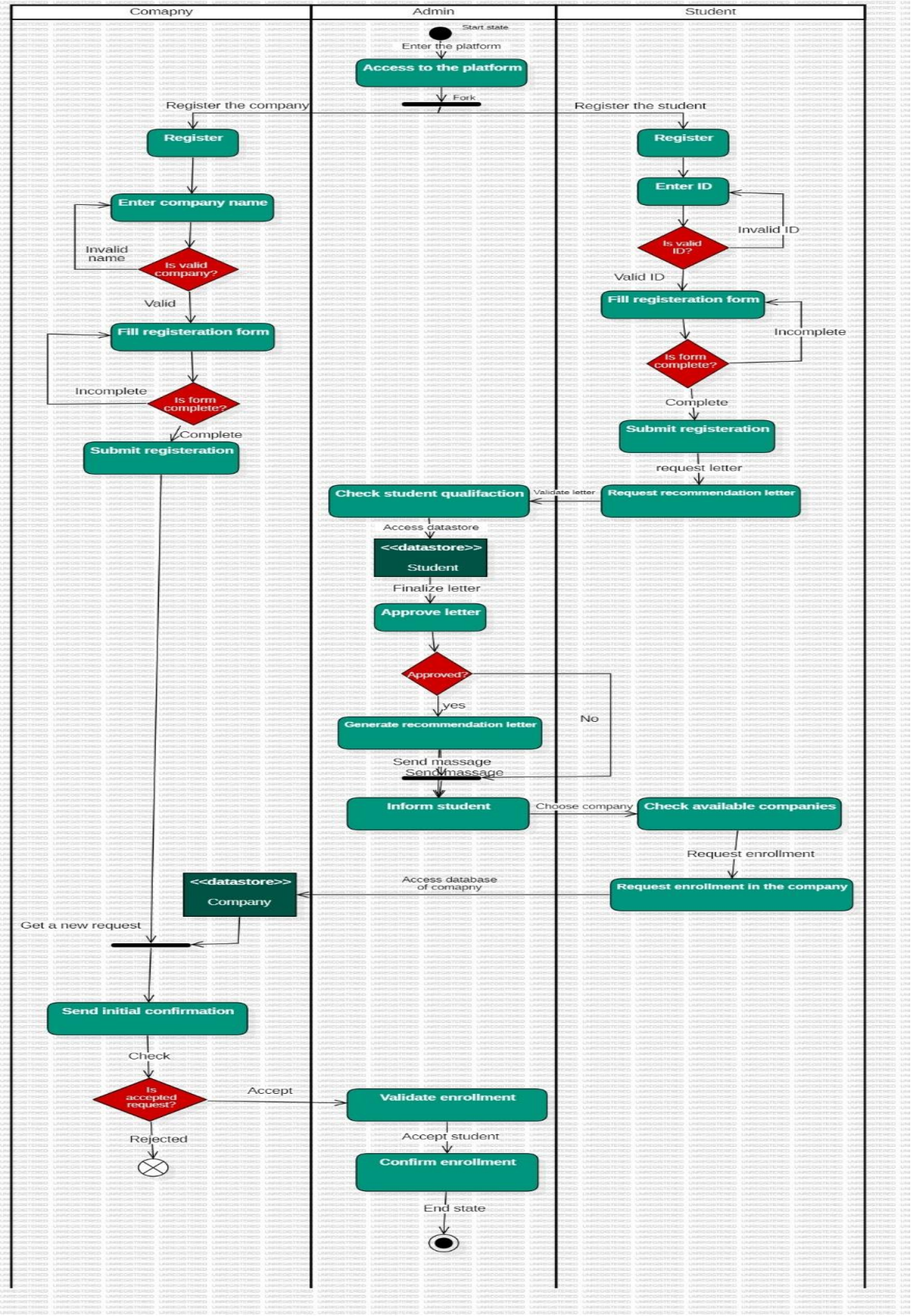
Use Case Diagram



Activity diagram:

- **Swimlanes:**
  - **Admin: -**
    The activity diagram is start from admin. Admin check student qualification and then generate letter. Moreover, he is confirm enrollment.
  - **Student: -**
    Student register in the platform, and then fill registration form then request recommendation letter.
  - **Company: -**
    Company register in the platform, and then fill registration form then send initial confirmation to students.

- **Description:**
  After both the company and the student log in to the platform, they must fill out the required fields. Then the student can request a letter of recommendation and it will be approved by the admin. After that, the student explores the companies available on the site and chooses the company he wishes to apply to. After applying to a specific company, he must wait until he receives a final response from the company.


Comapny
Admin
Student
Start state
Enter the platform
Access to the platform
Fork
Register the company
Register the student
Register
Register
Enter company name
Invalid name
Is valid company?
Valid
Fill registeration form
Incomplete
Is form complete?
Complete
Submit registeration
Enter ID
Invalid ID
Is valid ID?
Valid ID
Fill registeration form
Incomplete
Is form complete?
Complete
Submit registeration
request letter
Check student qualifaction
Validate letter
Request recommendation letter
Access datastore
<<datastore>>
Student
Finalize letter
Approve letter
Approved?
yes
No
Generate recommendation letter
Send massage
Send massage
Inform student
Choose company
Check available companies
Request enrollment
Request enrollment in the company
Access database of comapny
<<datastore>>
Company
Get a new request
Send initial confirmation
Check
Is accepted request?
Accept
Rejected
Validate enrollment
Accept student
Confirm enrollment
End state

Class diagram

- **Class & Attribute and method:**

  - **Student: -**
    Attributes: dept, CV, GBA
    Method: UploadCV(File CV), setDept(String dept), entireGBA(float gba), retrieve GBA( ), getDept(), retrieveCV( )

  - **User: -**
    Attributes: Password, ID, Name
    Method: setName(String Name), setID(int ID), setPassword(String password), getName( ), getPassword(), getID( )

  - **Company: -**
    Attributes: startUp
    Method: isStartUp(Boolean start), getStartUp()

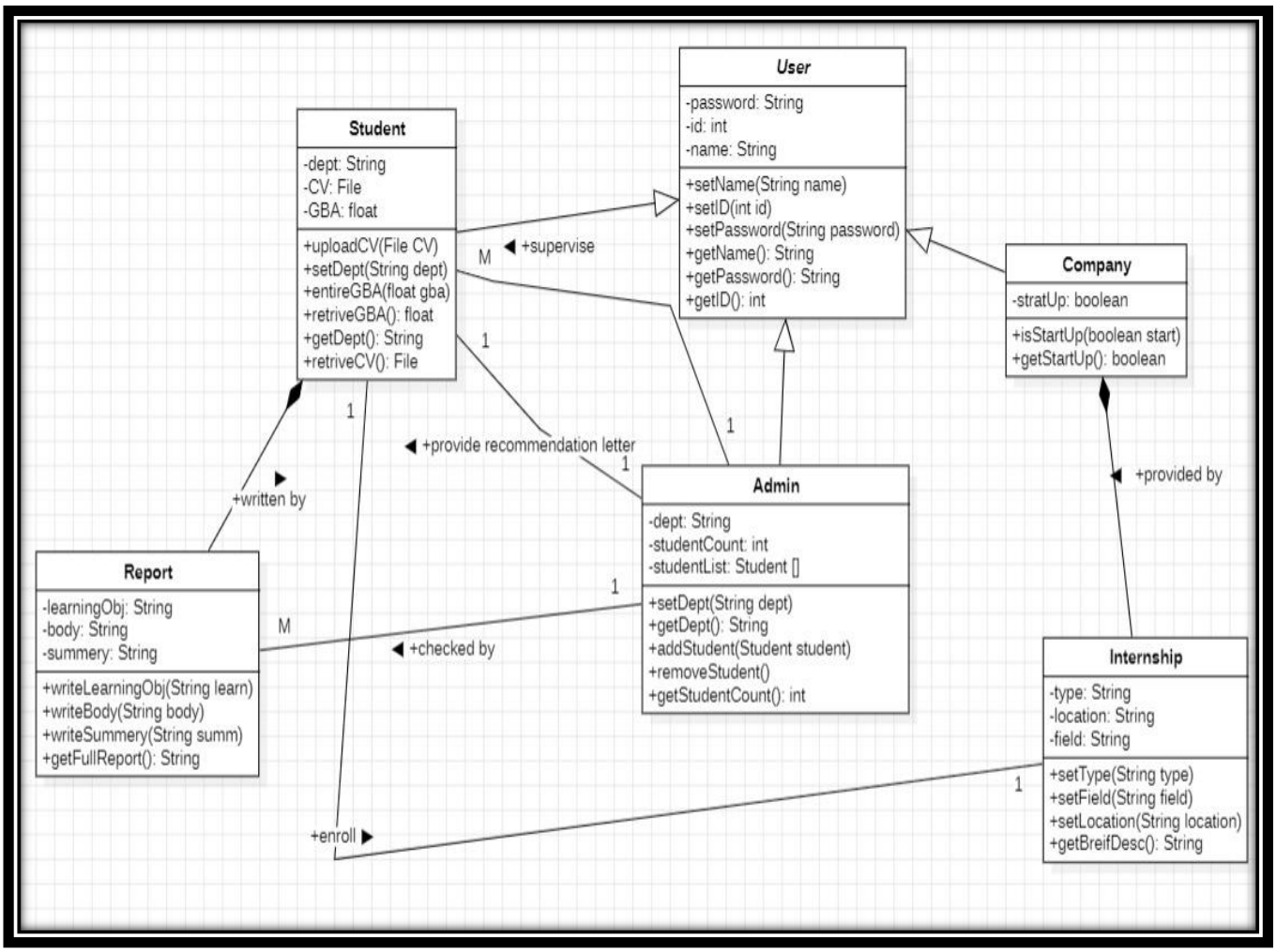
  - **Internship: -**
    Attributes: type, location, field
    Method: setType(String type), setField(String field), setLocation(String location), getBreifDesc()

  - **Admin: -**
    Attributes: dept, studentCount, studentList
    Method: setDept(String dept), getDept(), addStudent(Student student), removeStudent( ), getStudentCount()

  - **Report: -**
    Attributes: learningObj, body, summary
    Method: writeLearningObj(String learn), writeBody(String body), writeSummary(String summ), getFullReport()

- **Class Relationships:**
  - **Admin** inherit from user.
  - **Company** inherit from user.
  - **Student** inherit from user.
  - An **Internship** is provided by **Company** (Composition).
  - A **report** is written by **student.** Composition).
  - **Student** enroll in an **internship**.

- **Class Diagram:**

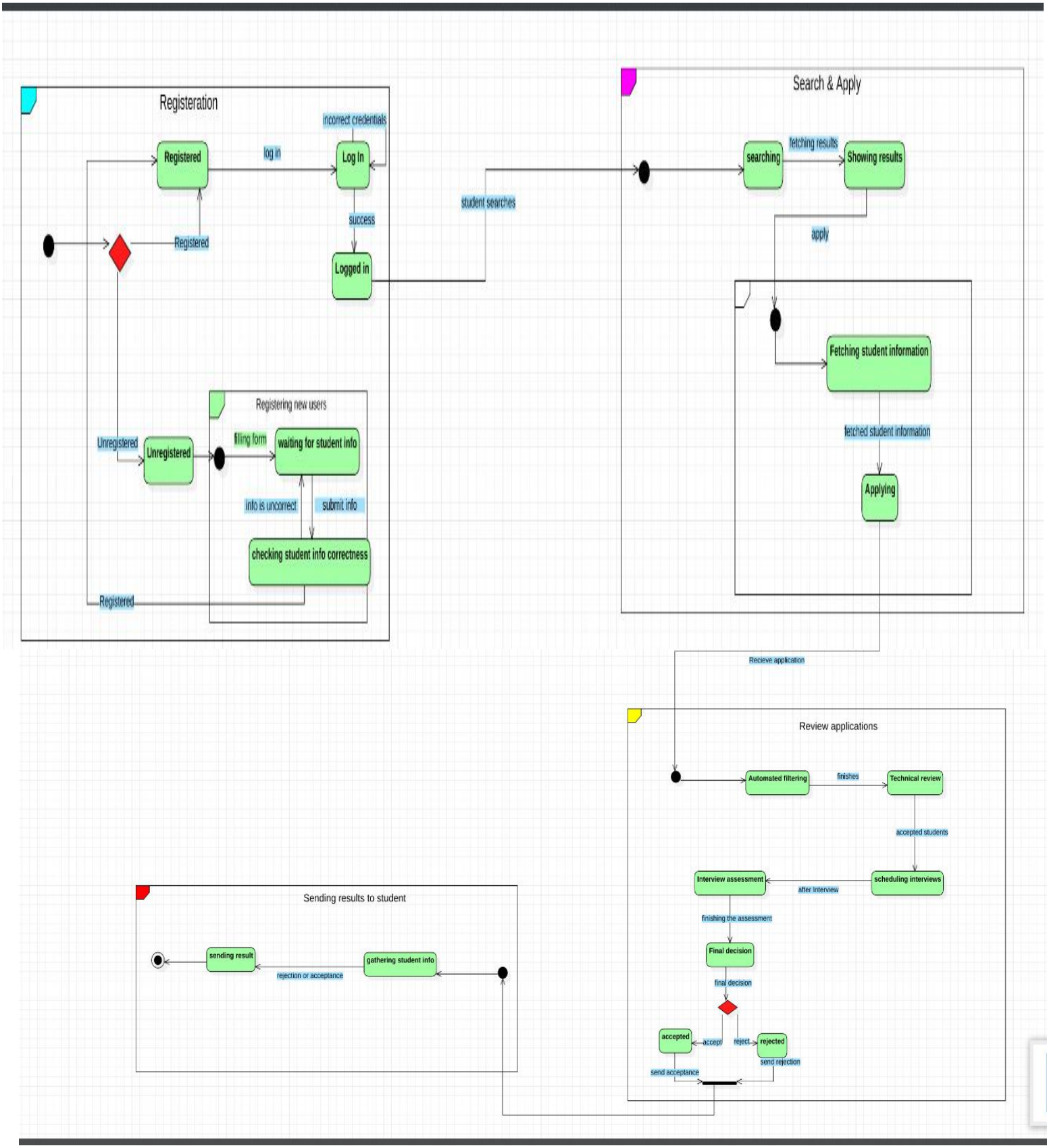


State diagram:

- **Possible states:**
  - **Registered: -**
    register important information when logging in for the first time.
  - **Log in: -**
    Enter the username and password to log in correctly.
  - **Logged in: -**
    The entry was made incorrectly.
  - **Unregistered: -**
    Important information was not registered when you first logged in.
  - **Waiting for student info: -**
    Few moments to fetch data from database.
  - **Checking student info correctness: -**
    Verify the accuracy of the personal information entered.
  - **Searching: -**
    Search for a company.
  - **Showing results: -**
    Show search results.
  - **Fetching student information: -**
    Few moments to fetch data from database.
  - **Applying: -**
    Accept request.
  - **Automated filtering: -**
    Filter based on something specific.
  - **Technical review: -**
    Comprehensive technical review.
  - **Interview assessment: -**
    Evaluating the person after the interview.

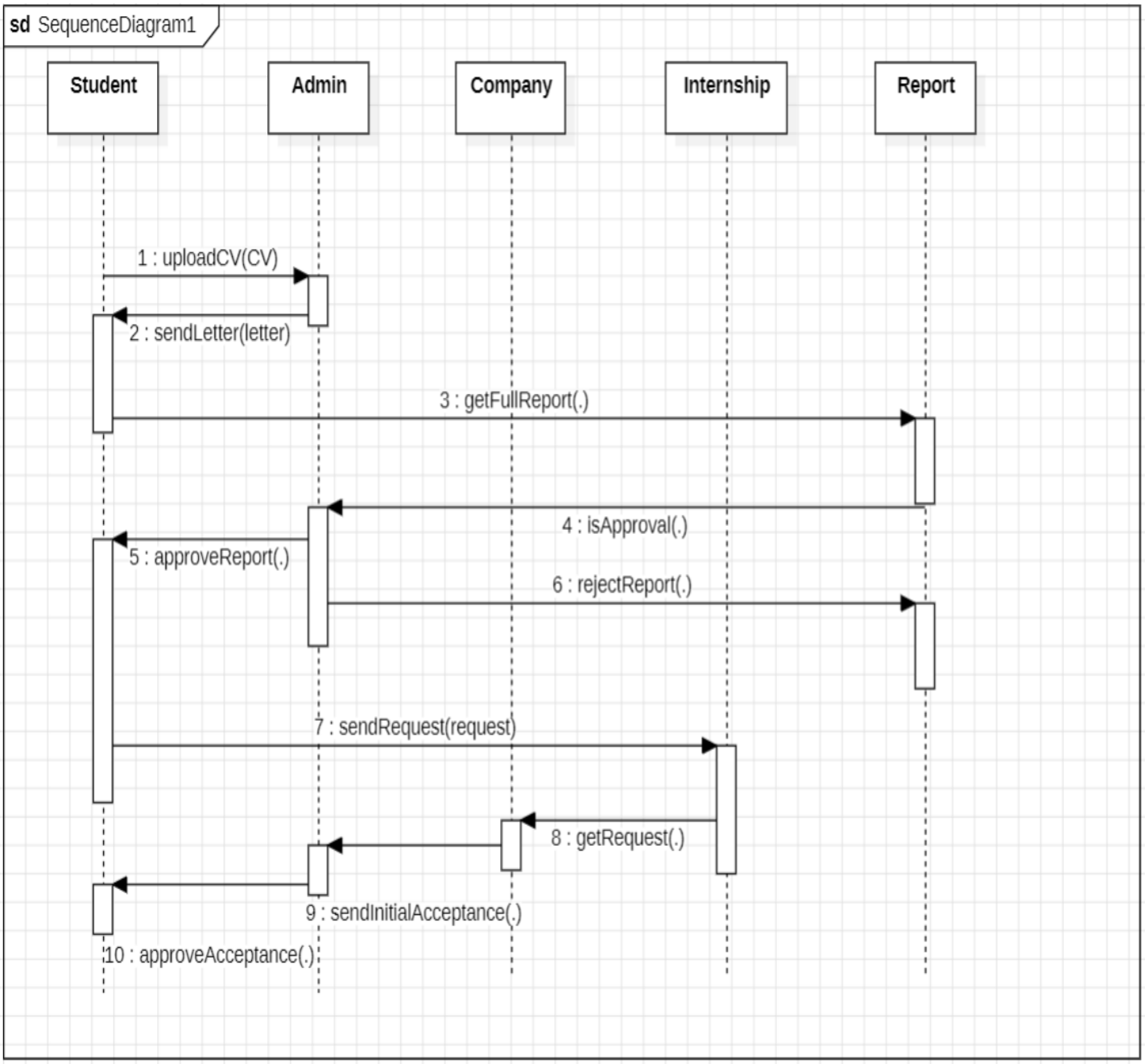
- **Final decision: -**
  Accept or reject person.
- **Sending result: -**
  Send final decision to a person.

- **Possible events:**
  - **Log in: -**
    Occur when logging in for the first time.
  - **Success: -**
    Occur when the user enter username and password correctly.
  - **Filling form: -**
    Occur when student click on register.
  - **Submit info: -**
    Occur after register the student in platform.
  - **Student search: -**
    Occur when student search in companies.
  - **Fetching results: -**
    Occur when student search in companies.
  - **Receive application: -**
    Occur after applying student information.
  - **Finishing the assessment: -**
    Occur when starting evaluate the person.
  - **Send acceptance: -**
    Occur when the student is get acceptance after interview.
  - **Send rejection: -**
    Occur when the student is get rejection after interview.


Registeration
Registered
log in
Log In
incorrect credentials
success
Logged in
Registered
Registering new users
Unregistered
Unregistered
filling form
waiting for student info
info is uncorrect
submit info
checking student info correctness
Registered
student searches
Search & Apply
searching
fetching results
Showing results
apply
Fetching student information
fetched student information
Applying
Recieve application
Review applications
Automated filtering
finishes
Technical review
accepted students
Interview assessment
after Interview
scheduling interviews
finishing the assessment
Final decision
final decision
accepted
accept
reject
rejected
send rejection
send acceptance
Sending results to student
sending result
rejection or acceptance
gathering student info

Sequence diagram:

- **Massages & methods:**
  - **uploadCV(CV): -**
    When **student** upload it's CV to **admin**.
  - **sendLetter(letter): -**
    Admin give **student** letter.
  - **getFullReport(): -**
    Get the full report from **report.**
    isApproval(): -
    Check whether the report is approved or not.
  - **approveReport(): -**
    The **admin** approves the report.
  - **rejectReport(): -**
    The **admin** rejects the report.
  - **sendRequest(request): -**
  - **Student** sends request to **internship** in order to enroll.
  - **getRequest(): -**
    The transaction goes to the **company**.
  - **sendInitialAcceptance(): -**
    The **company** send initial acceptance to **admin**.
  - **approveAcceptance(): -**
    The admin approve the enrollment.


sd SequenceDiagram1
Student
Admin
Company
Internship
Report
1 : uploadCV(CV)
2 : sendLetter(letter)
3 : getFullReport(.)
4 : isApproval(.)
5 : approveReport(.)
6 : rejectReport(.)
7 : sendRequest(request)
8 : getRequest(.)
9 : sendInitialAcceptance(.)
10 : approveAcceptance(.)

ER diagram:

- **Entities:**
  - **Admin:**

    Attributes: AdminID, Role, Department, Name.

    Responsible for overseeing the enrollment and management of internships for students.
  - **Student:**

    Attributes: StudentID, Name, Email, Phone, Major, GPA.

    Represents students who apply for and participate in internships.
  - **Internship:**

    Attributes: InternshipID, Type, Location, Details, Duration, CompanyID.

    Represents internship opportunities provided by companies.
  - **Company:**

    Attributes: CompanyID, Name, Location, Details.
    Represents the companies offering internships to students.
  - **Enrollment:**

    Attributes: EnrollmentID, StudentID, InternshipsID, AdminID, EnrollmentDate, Status.
    Represents the enrollment status of students in internships, overseen by an admin.
  - **Application:**

    Attributes: ApplicationID, StudentID, InternshipsID, AdminID, ApplicationDate, Status.
    Represents applications submitted by students for internships.

- **Relationships:**
  - **Oversees:**

    Between: Admin & Enrollment

    Description: Indicates that each enrollment is overseen by an admin who is responsiple for managing student's internship.

  - **Enrollsln:**

    Between: Student & Enrollment
    Description: Shows that students are enrolled in internships once their applications are approved, linking students to their respective internship.

  - **HasEnrollment:**

    Between: Internship & Enrollment
    Description: Connects internships with their enrollments, indicating which students are actively participating in each internship.

  - **Offers:**

    Between: Company & Internship
    Description: Illustrates that companies provide internship opportunities for students, associating each internship with the offering company.

  - **AppliesFor:**

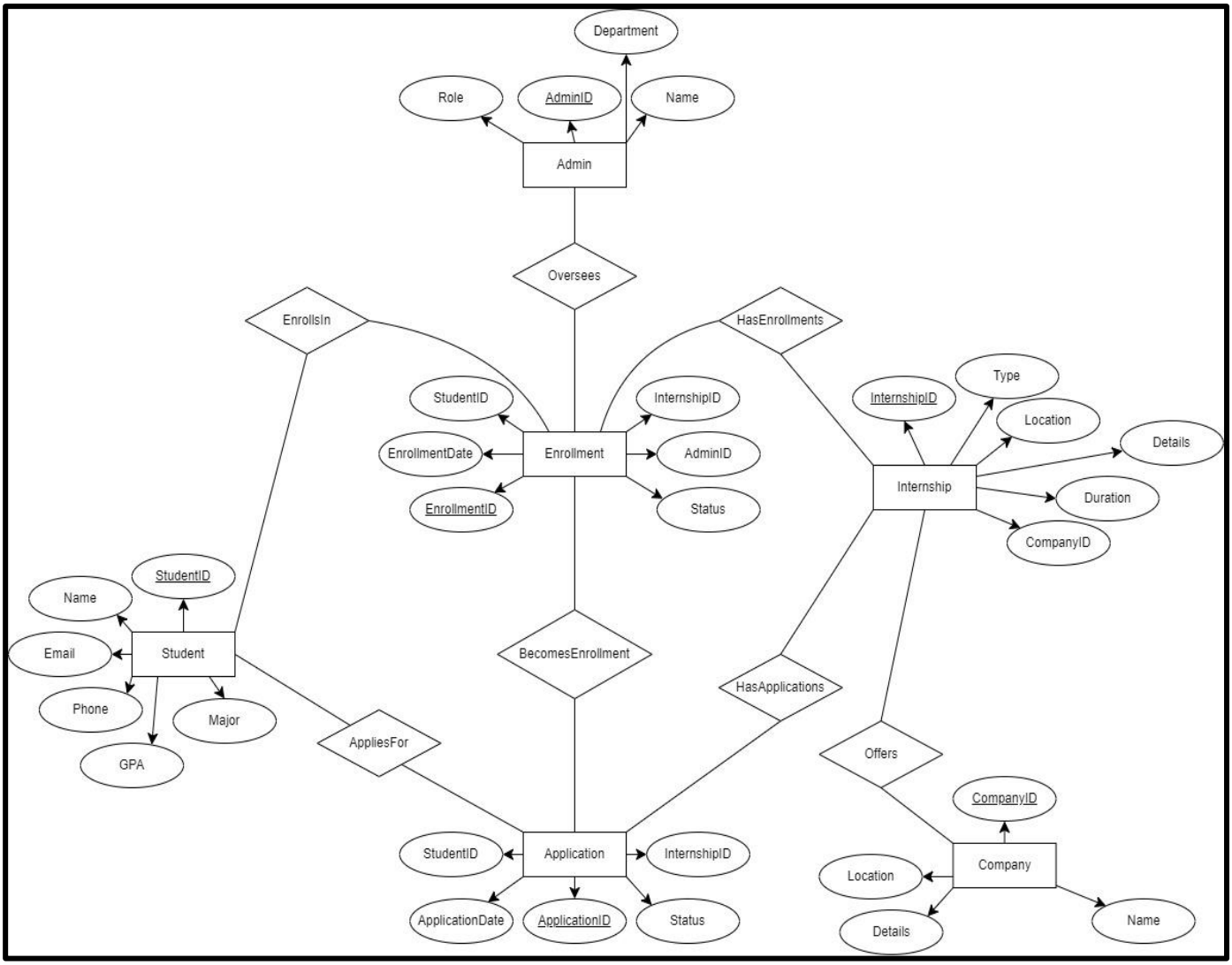
    Between: Student & Application
    Description: Represents the act of students applying for available internships, capturing the application process and its status.

  - **HasApplication:**

    Between: Internship & Application
    Description: Indicates that each internship has multiple applications from different students, tracking the demand and selection process for each internship.

  - **BecomesEnrollment:**

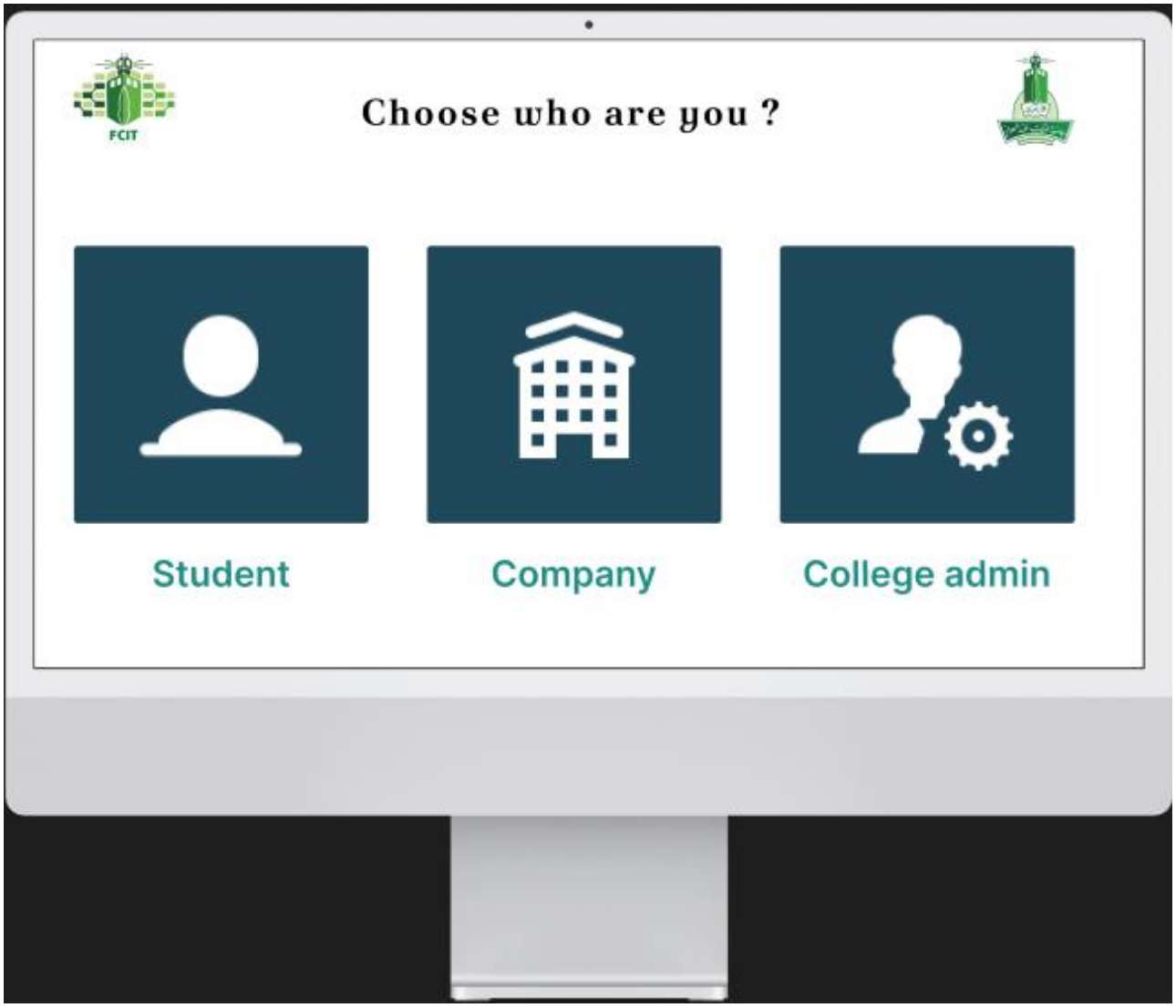
    Between: Application & Enrollment
    Description: Represents the transition of an application to an enrollment when a student is accepted for an internship, making the application official as an enrollment.


Department
Role
AdminID
Name
Admin
Oversees
EnrollsIn
HasEnrollments
StudentID
InternshipID
EnrollmentDate
Enrollment
AdminID
EnrollmentID
Status
InternshipID
Type
Location
Details
Internship
Duration
CompanyID
StudentID
Name
Email
Student
Phone
Major
GPA
BecomesEnrollment
HasApplications
AppliesFor
Offers
CompanyID
StudentID
Application
InternshipID
Location
Company
ApplicationDate
ApplicationID
Status
Details
Name

Interfaces

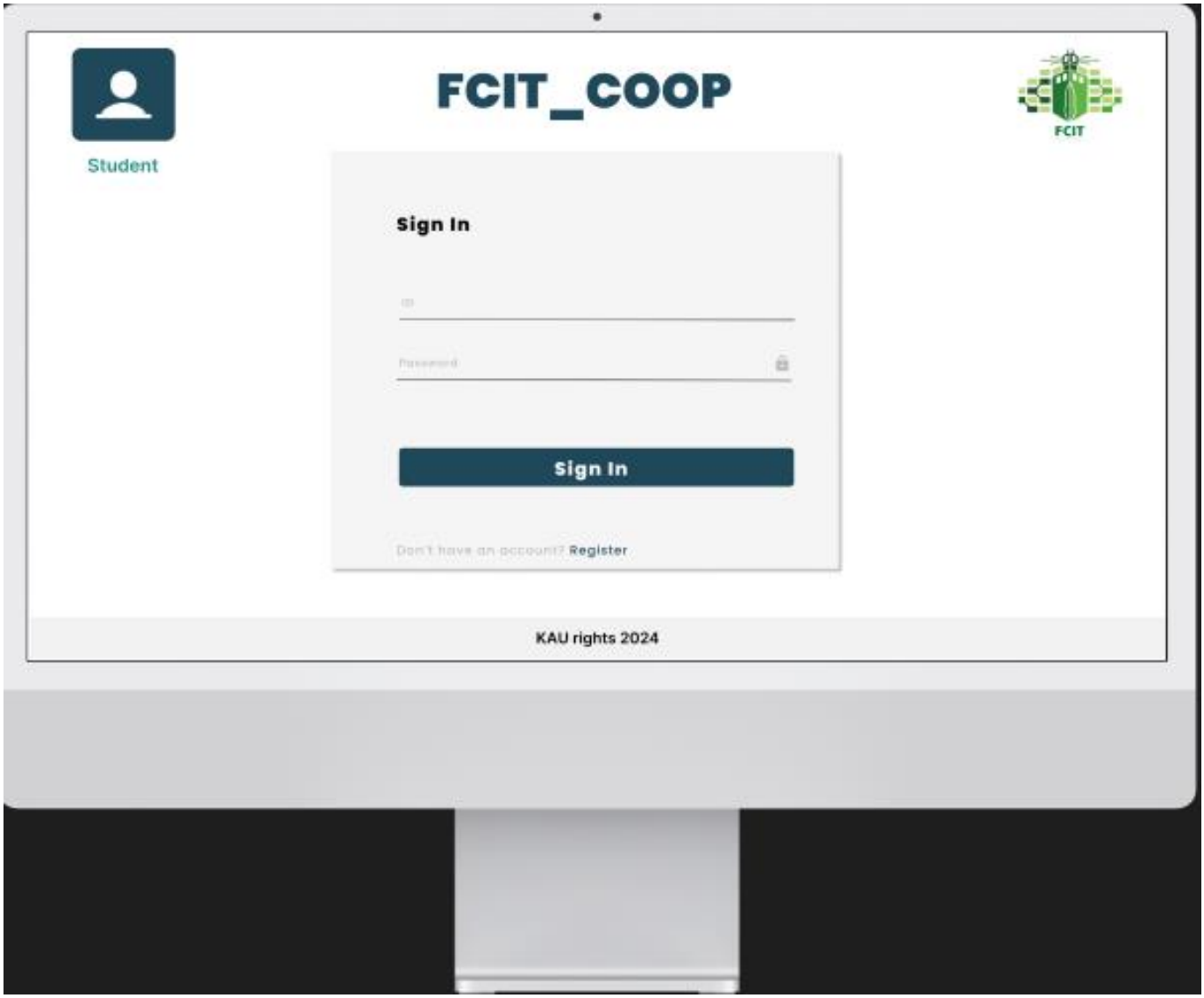
Splash Screen

Home Page



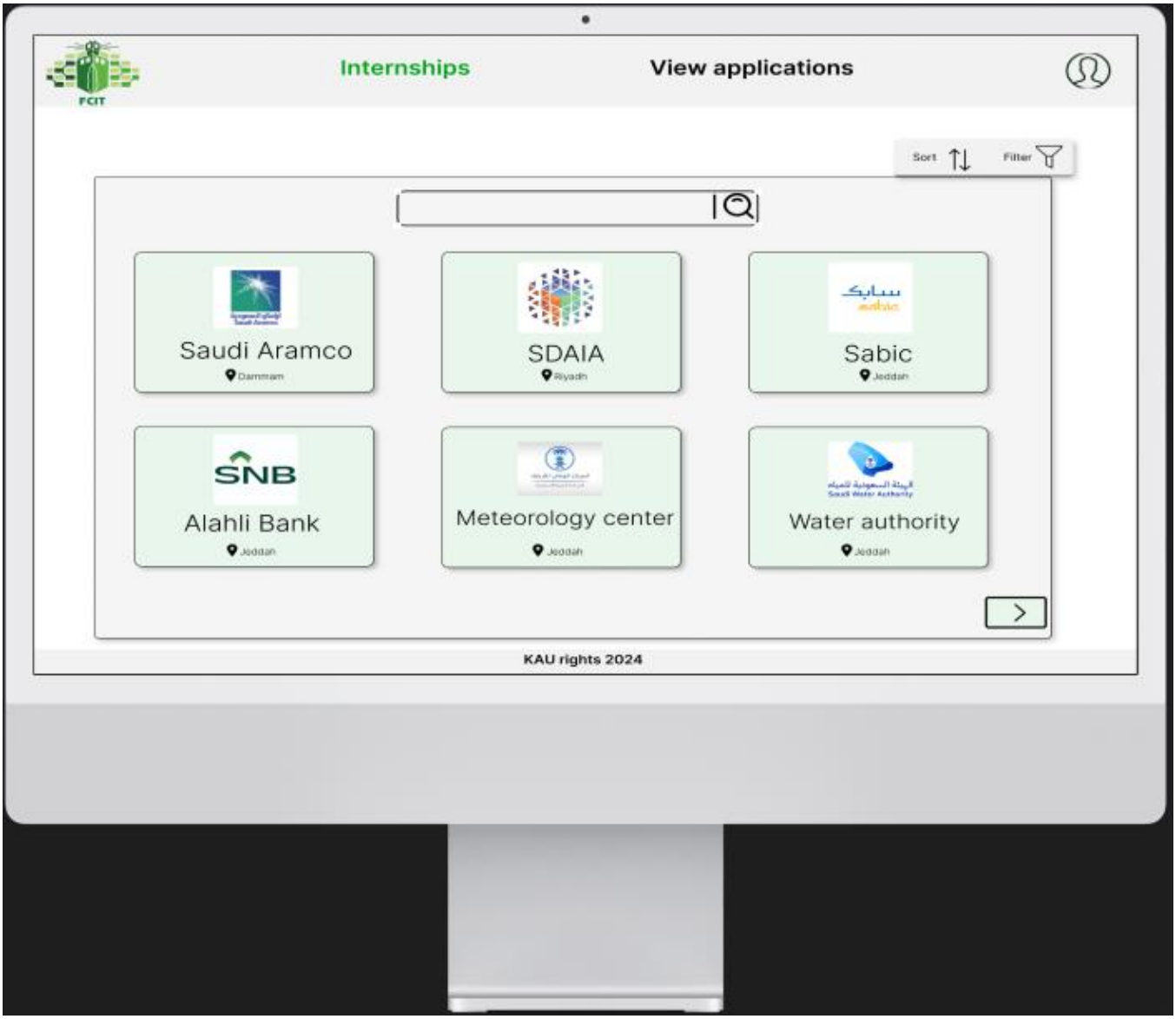
Student interfaces

Student: sign in

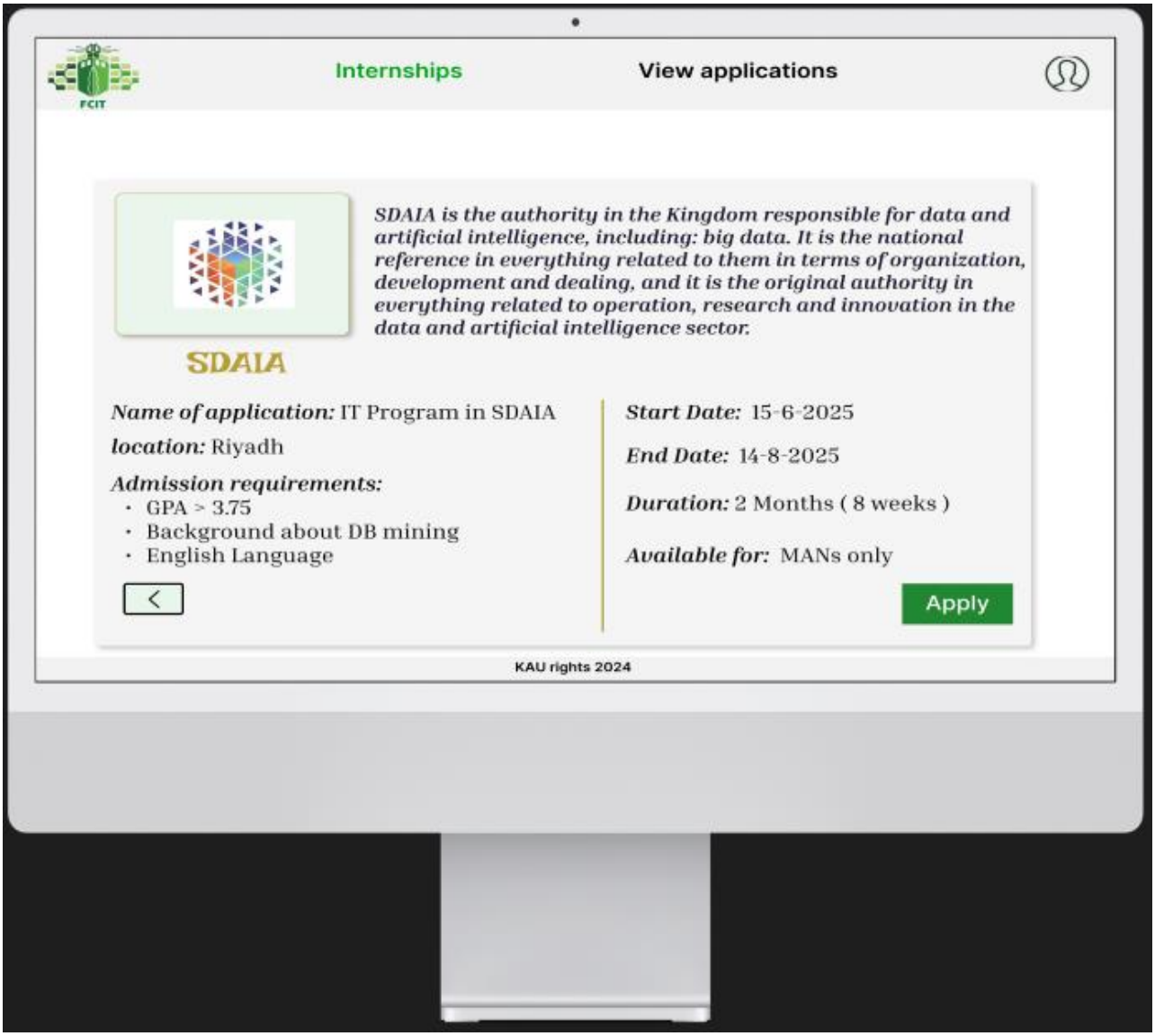


Student: register

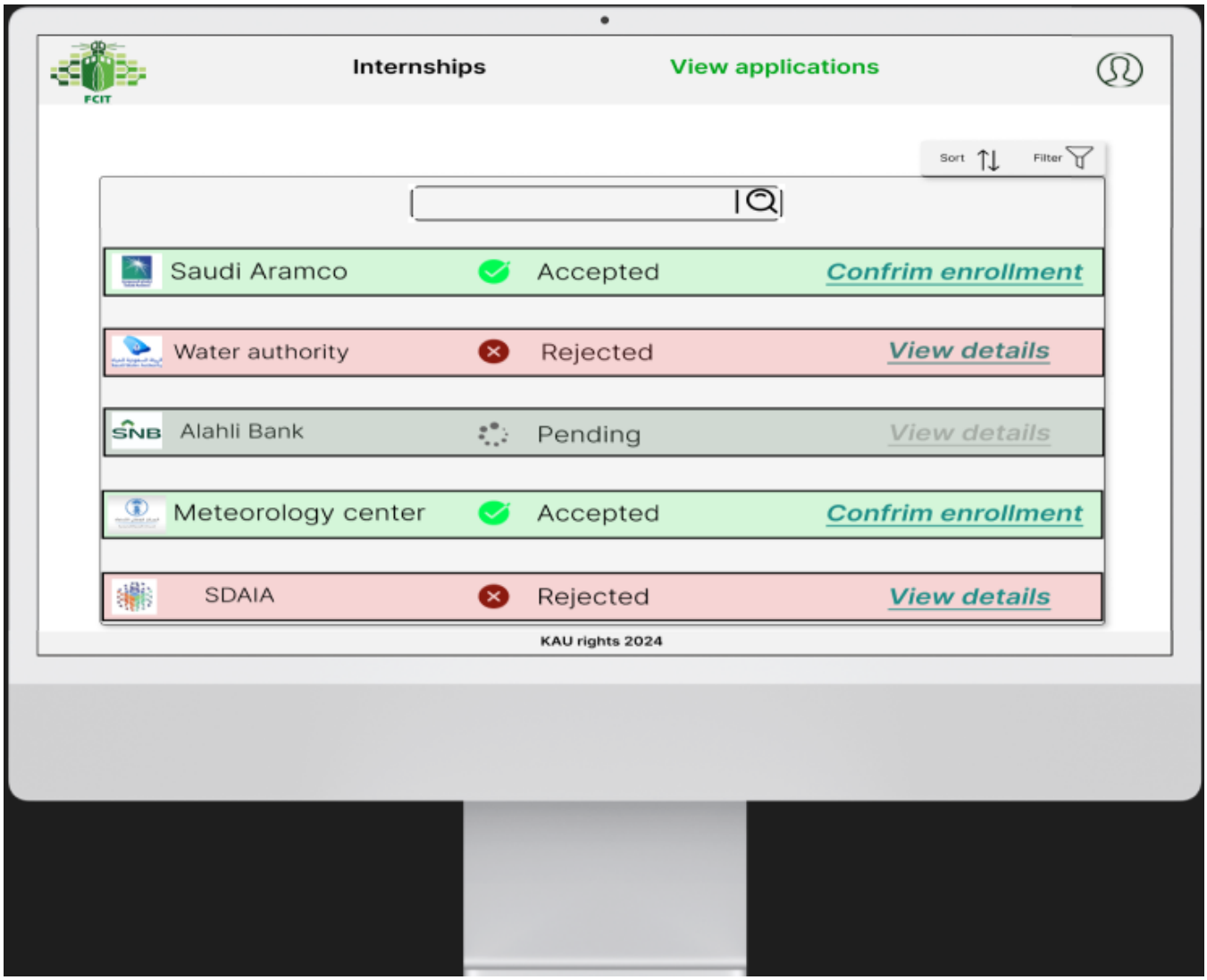
Student: Internships



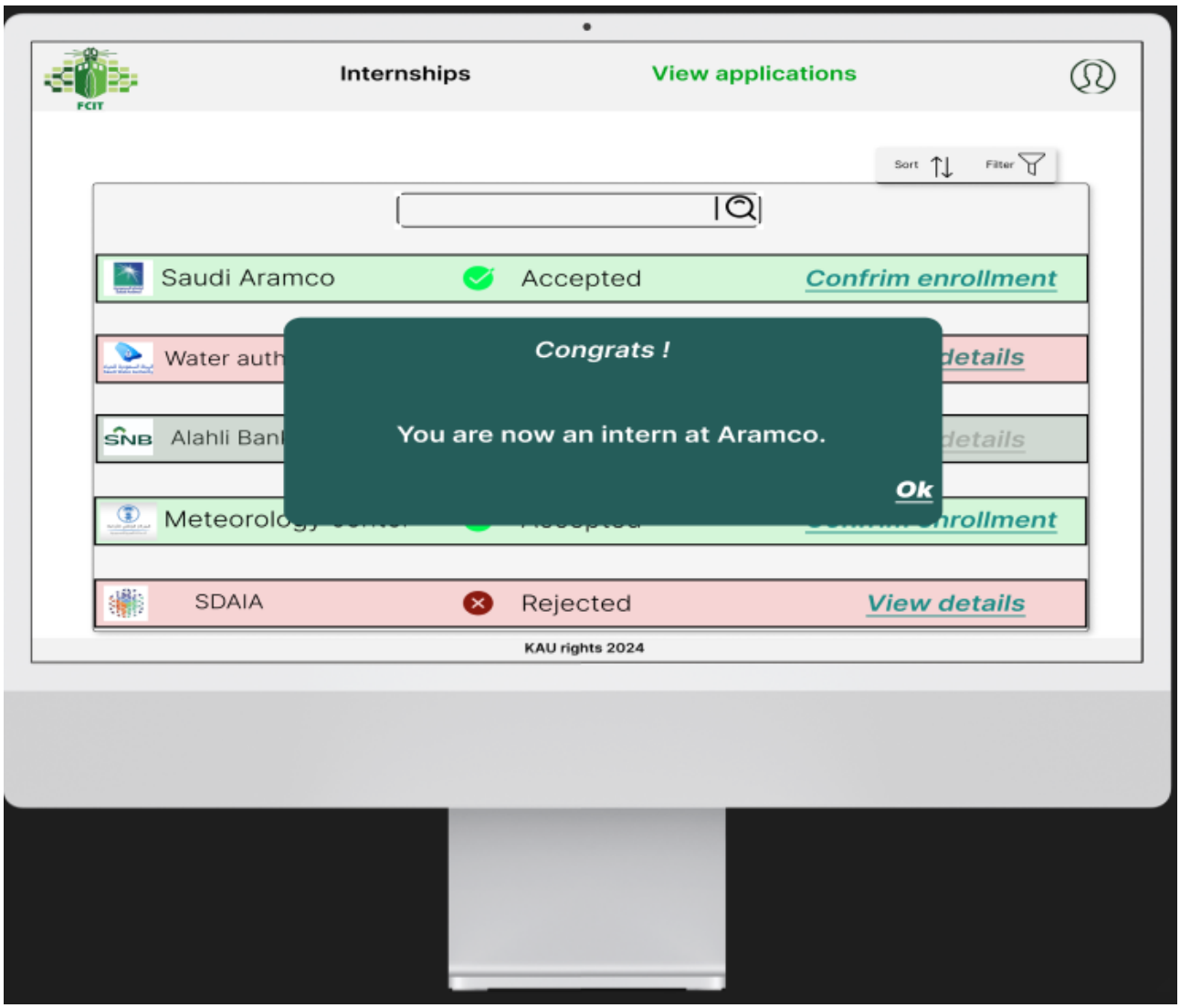
Student: Internships (2)



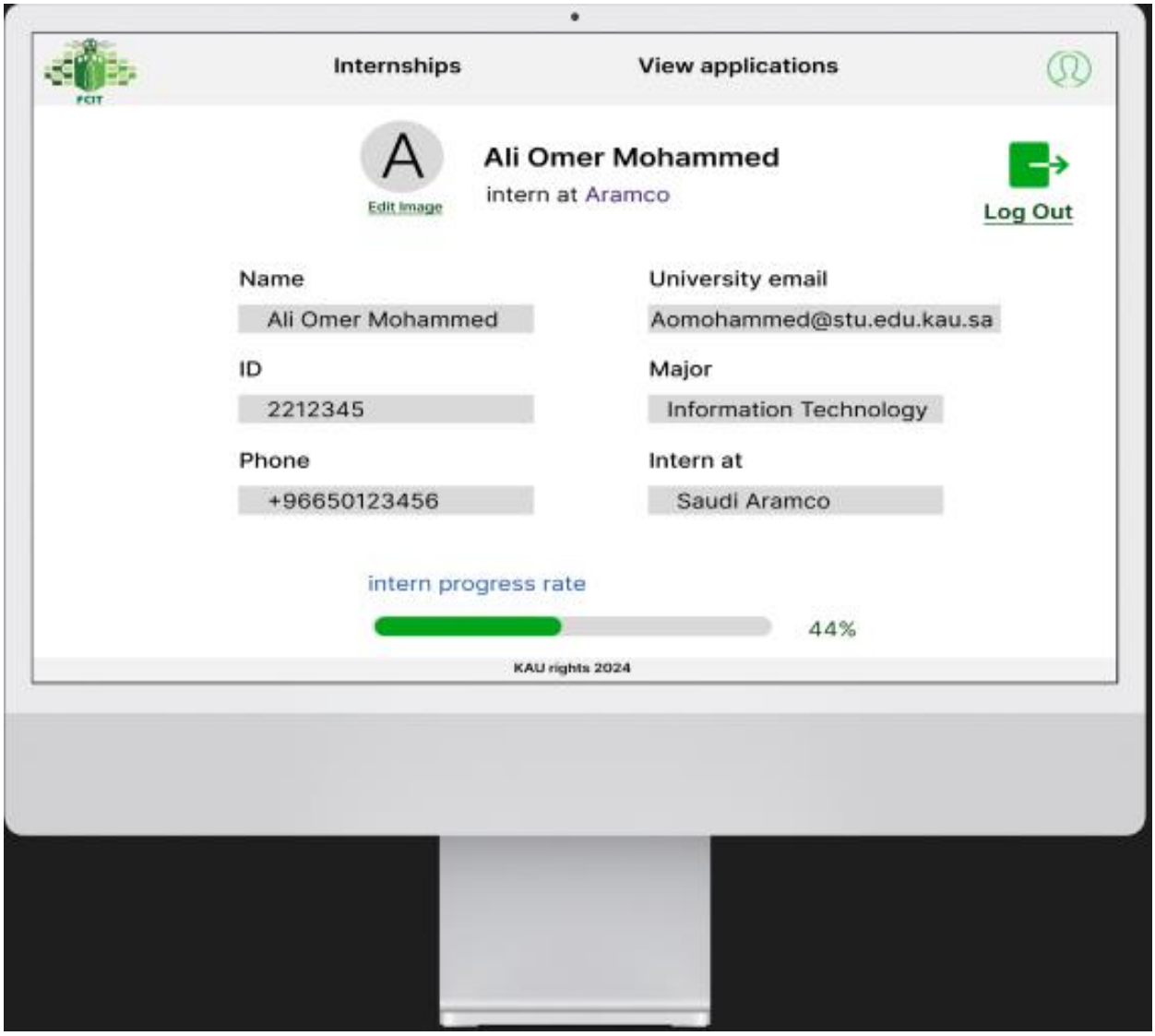
Student: View application



Student: View application (2)

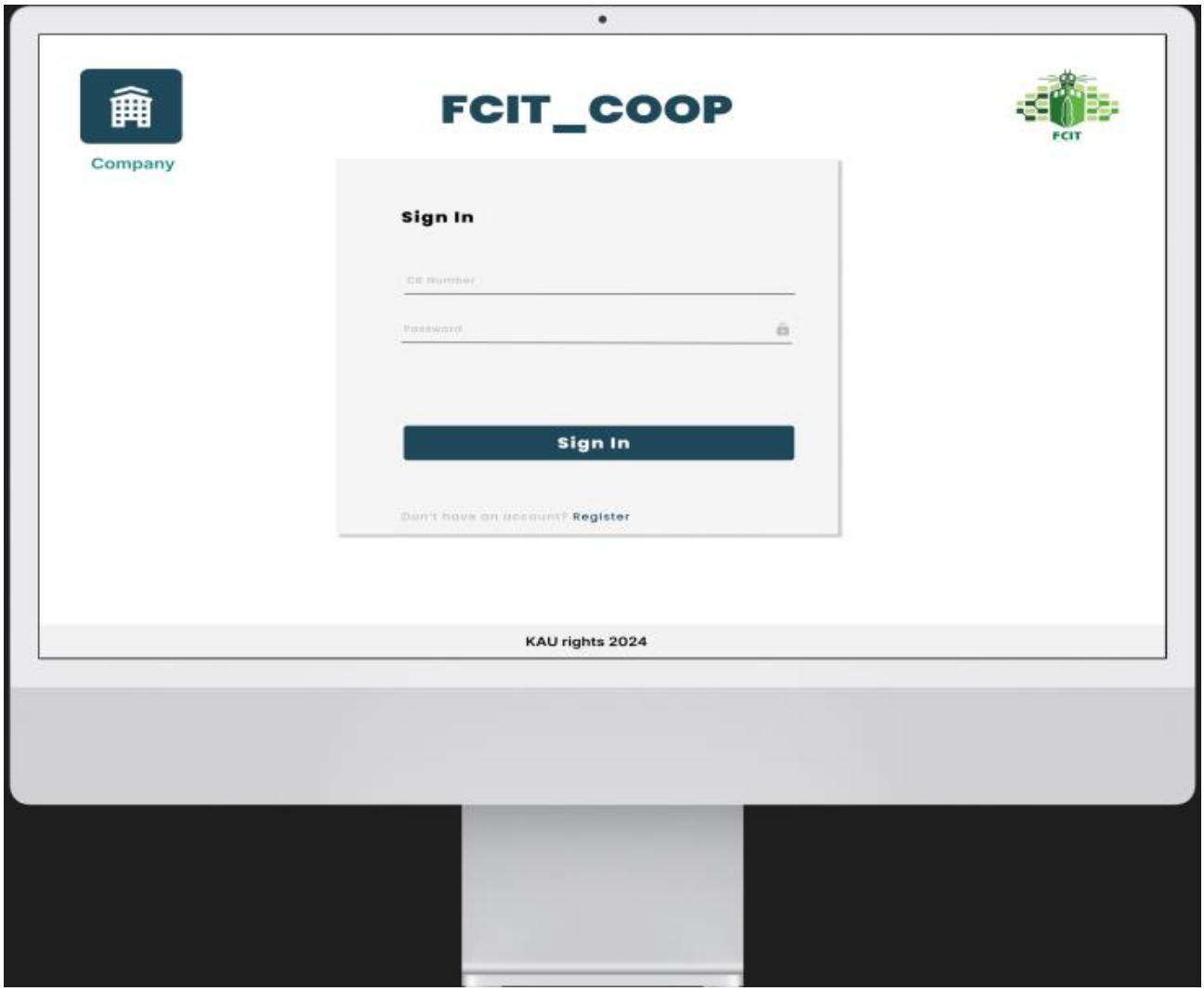


Student: Personal info

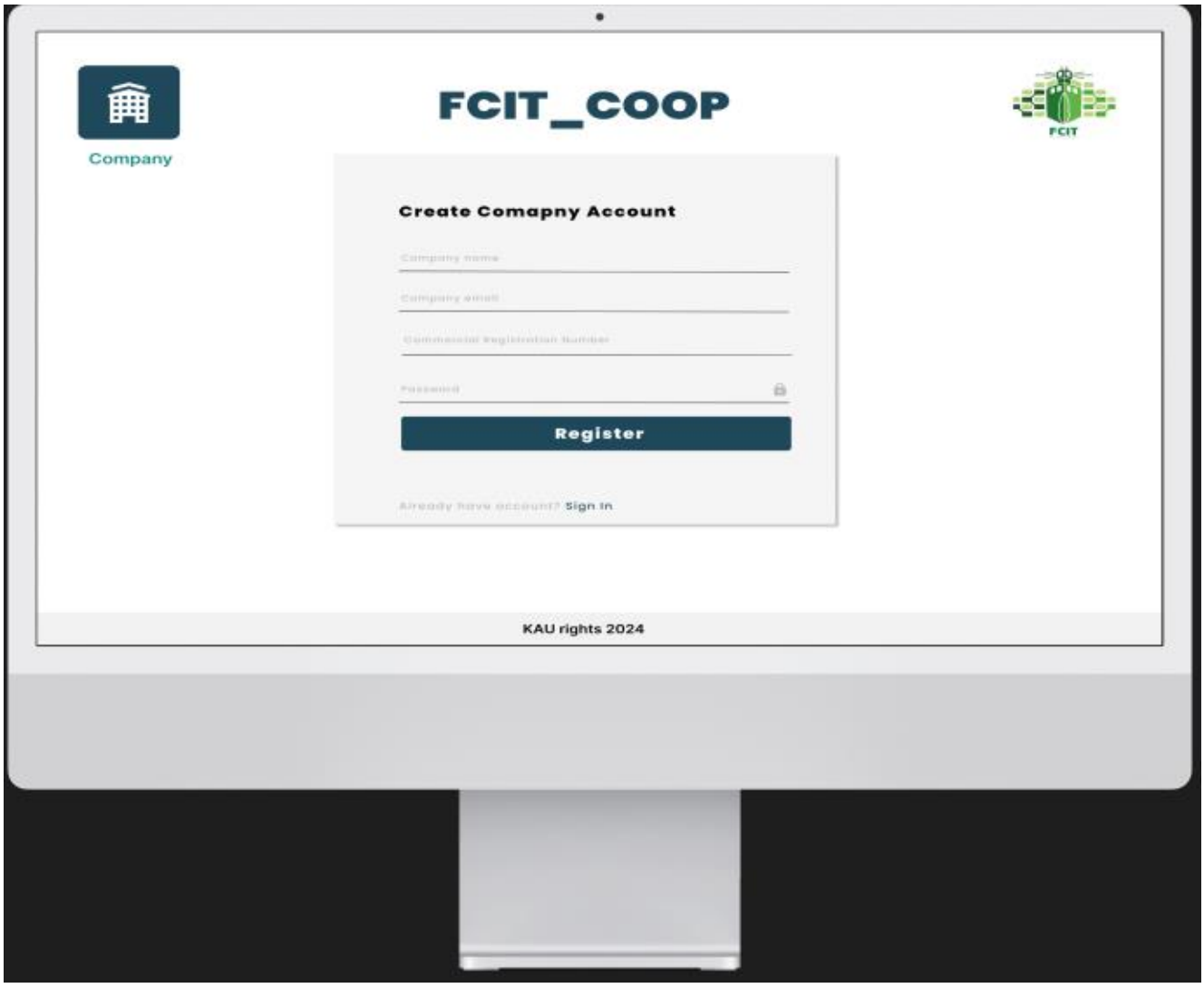


Company interfaces

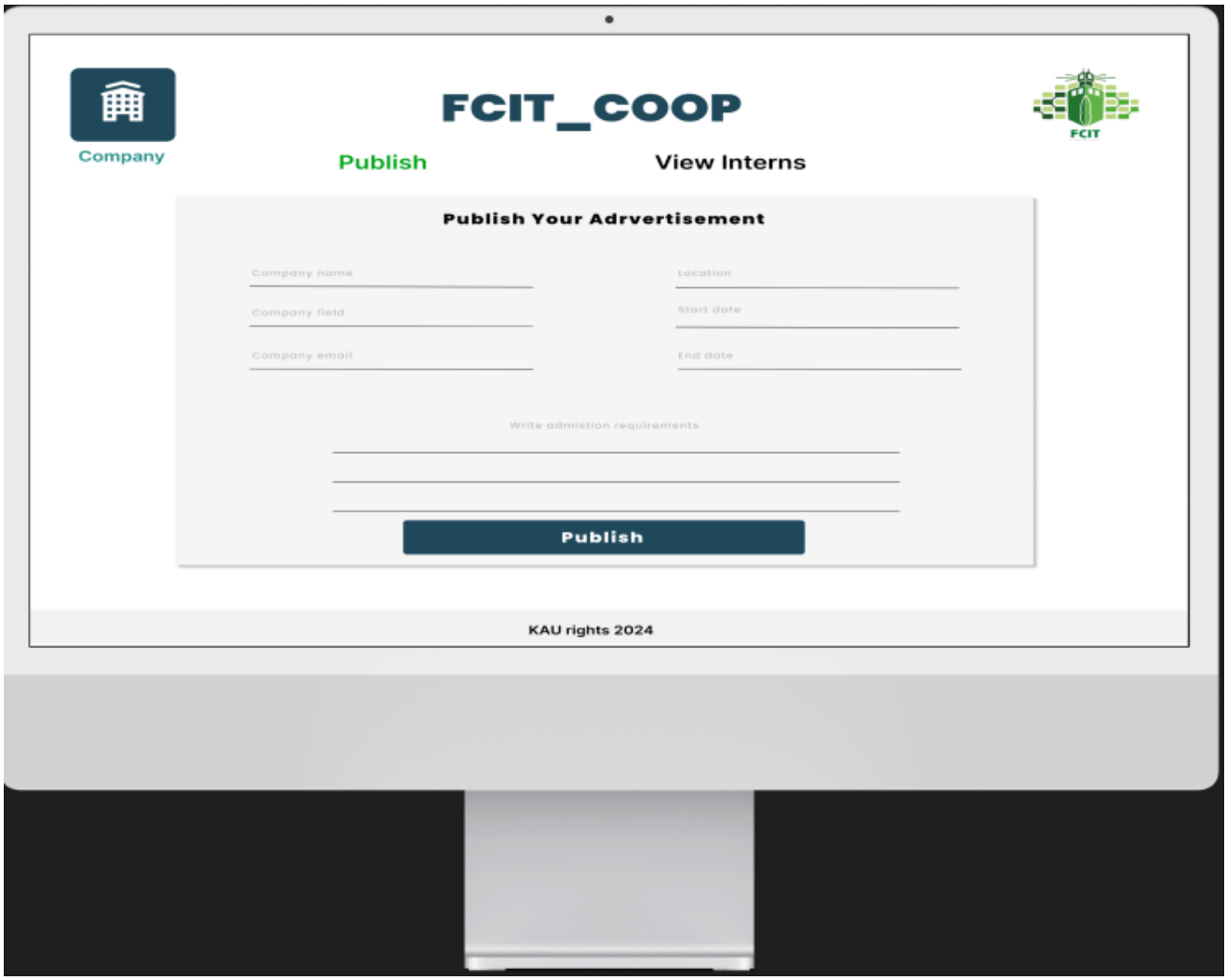
Company: sign in



Company: registration



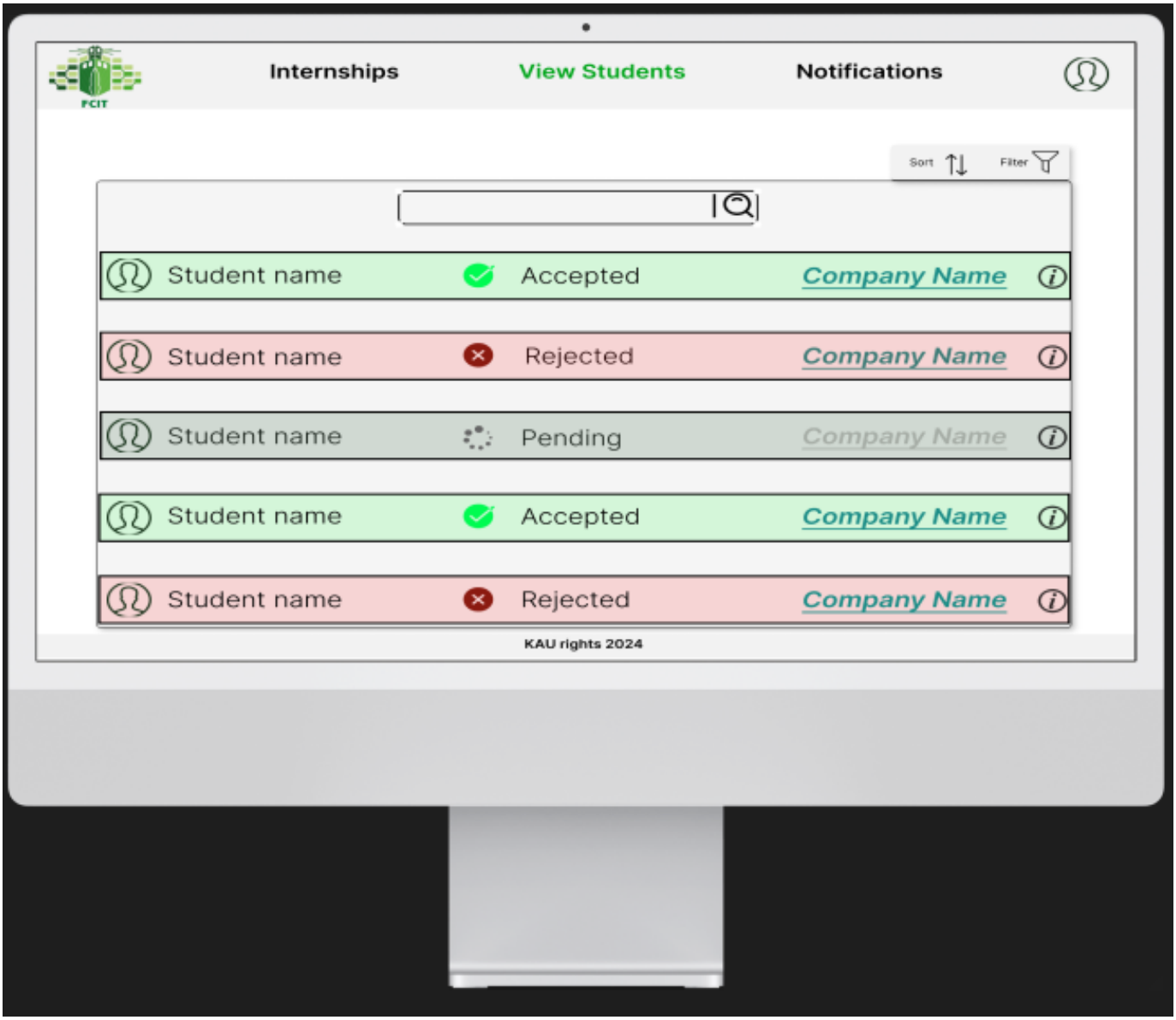
Company: publish



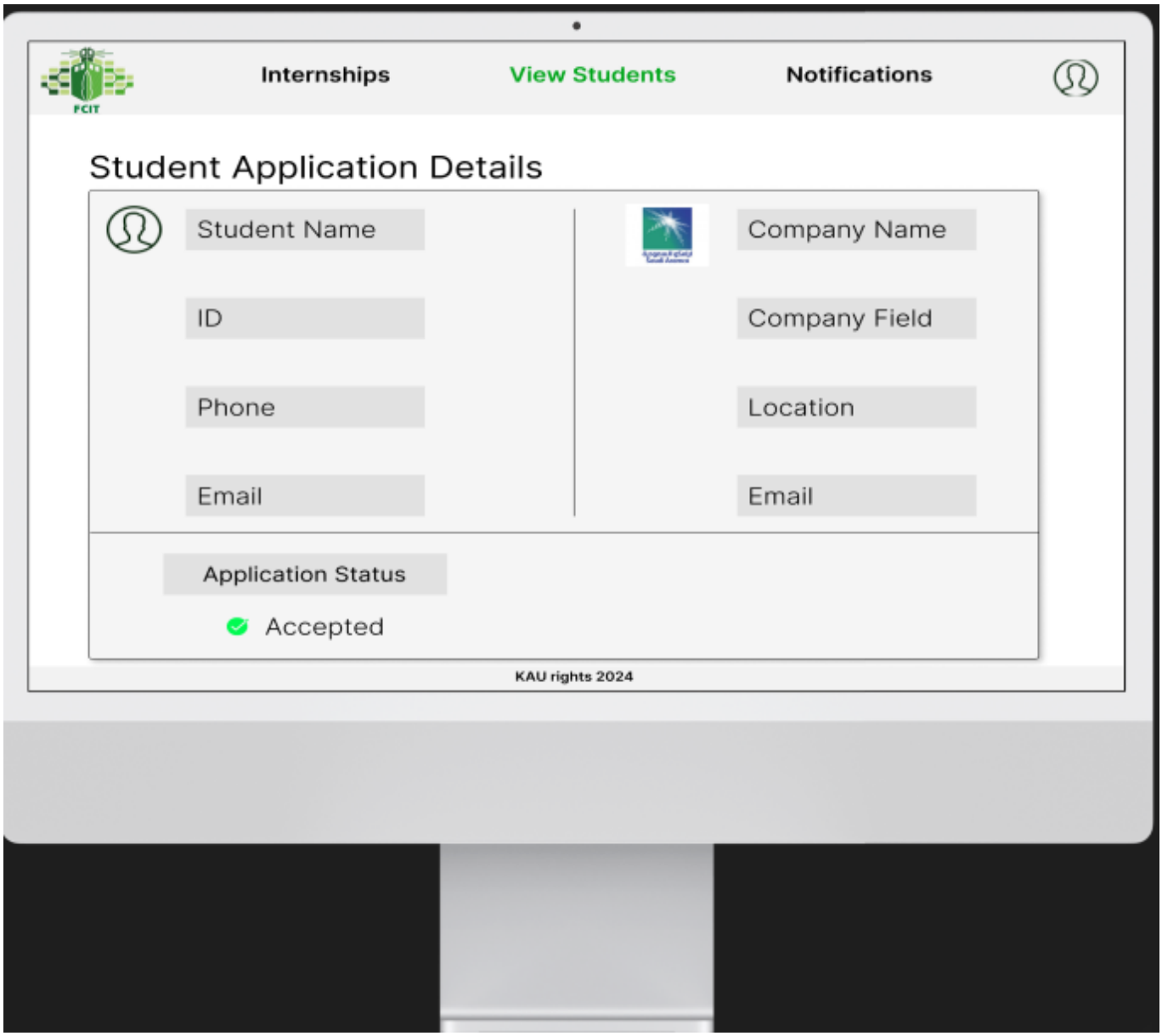
Company: View interns

Admin interfaces

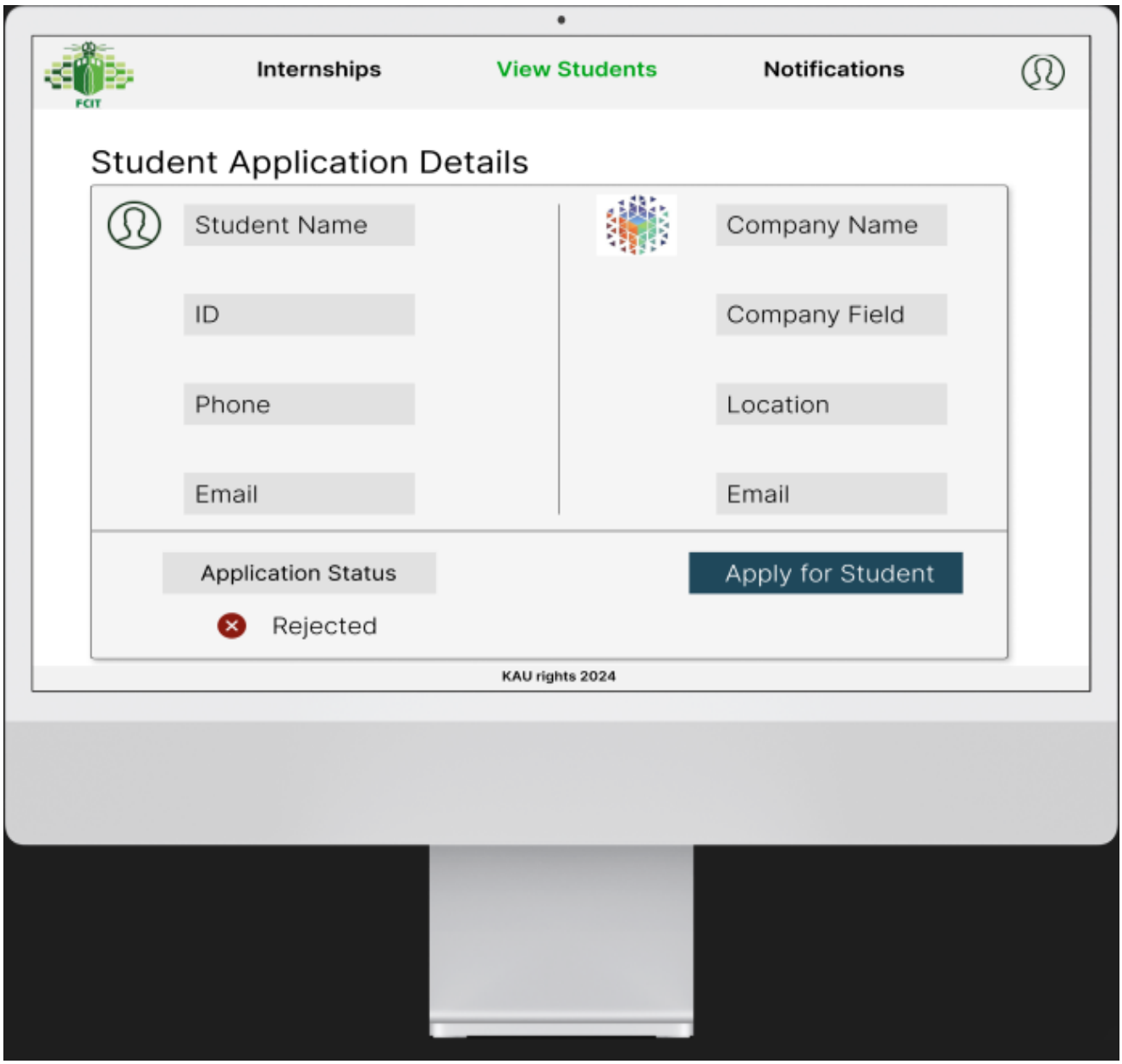
Admin: view students



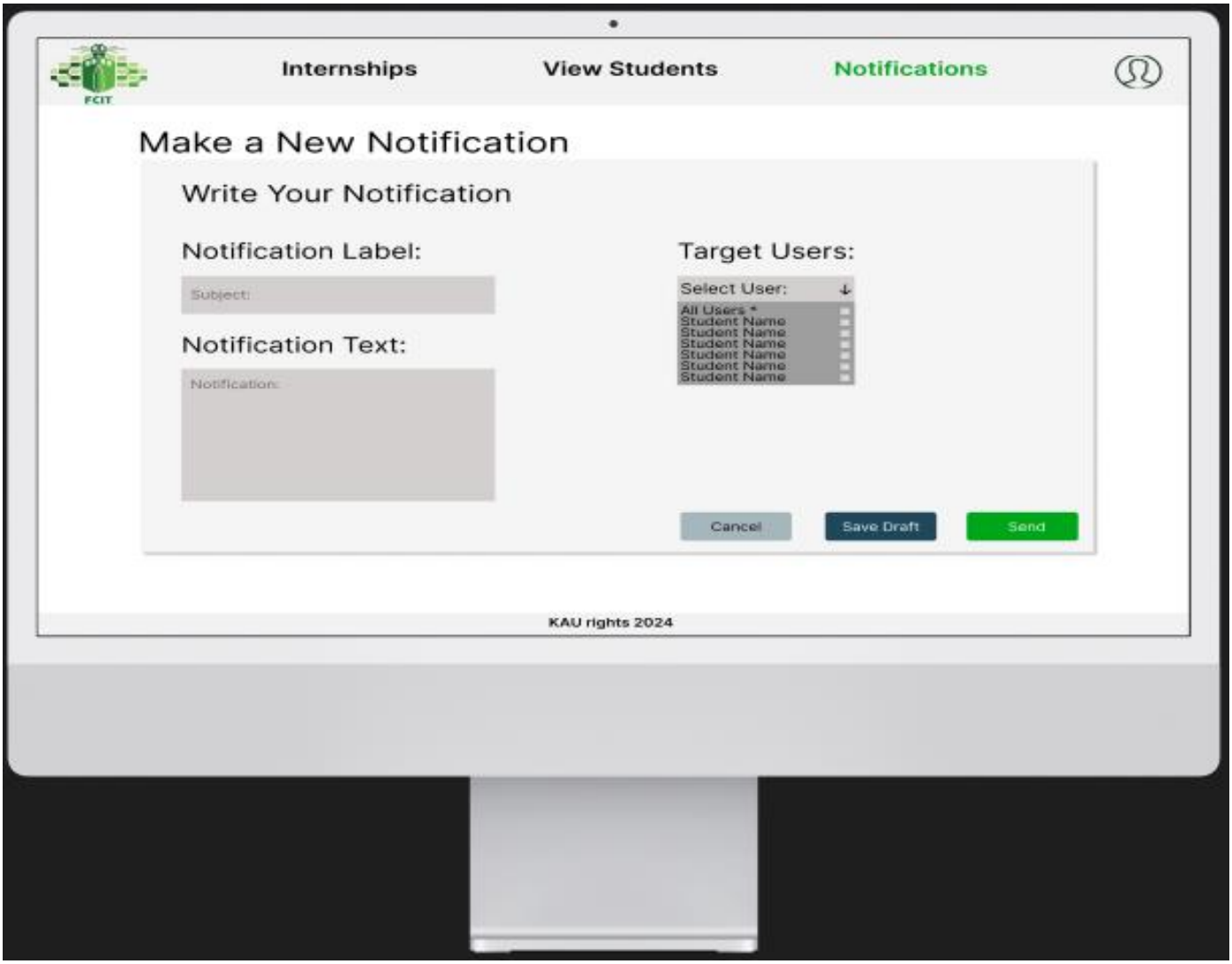
Admin: view accepted students



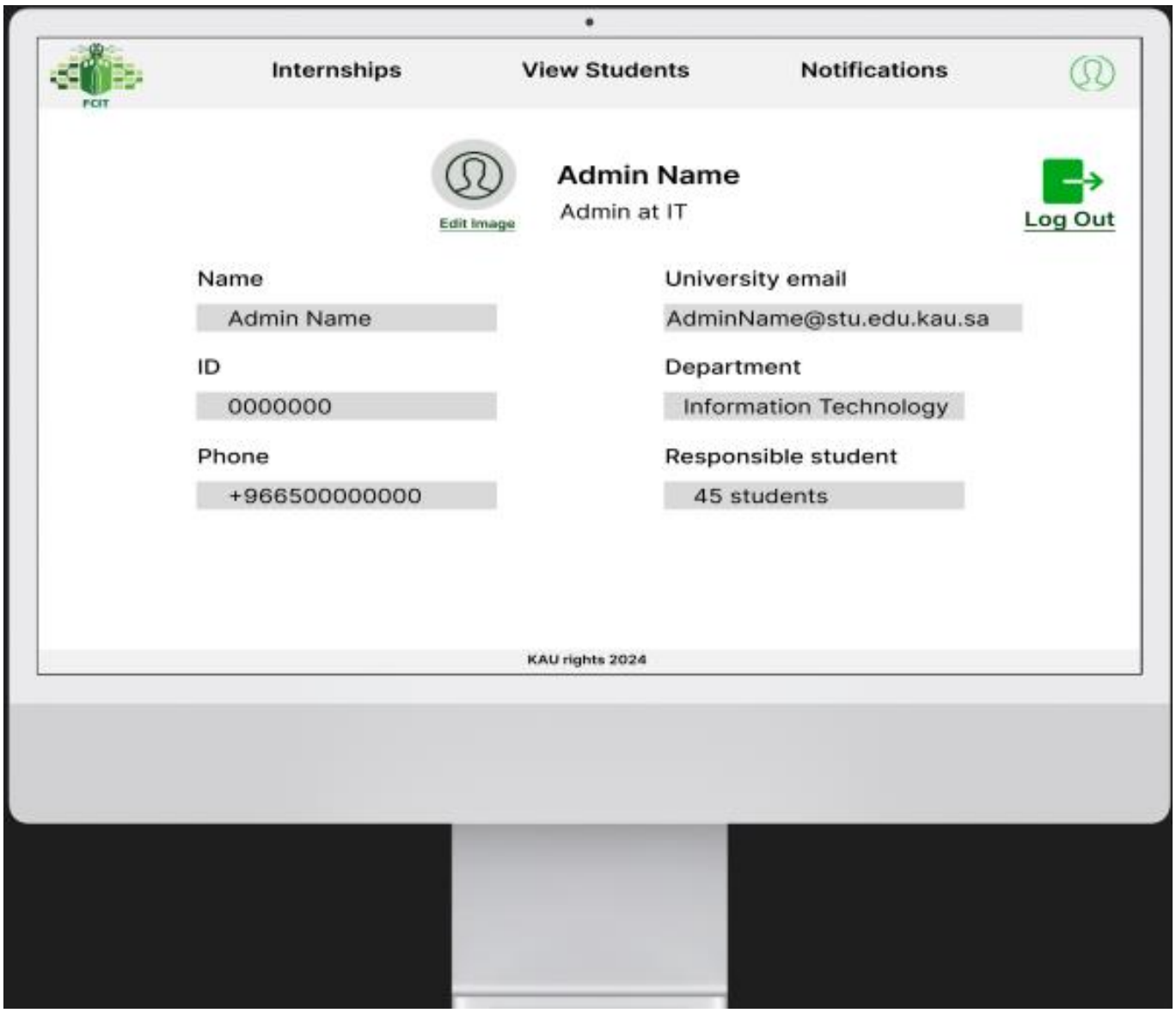
Admin: view rejected students



Admin: Notification

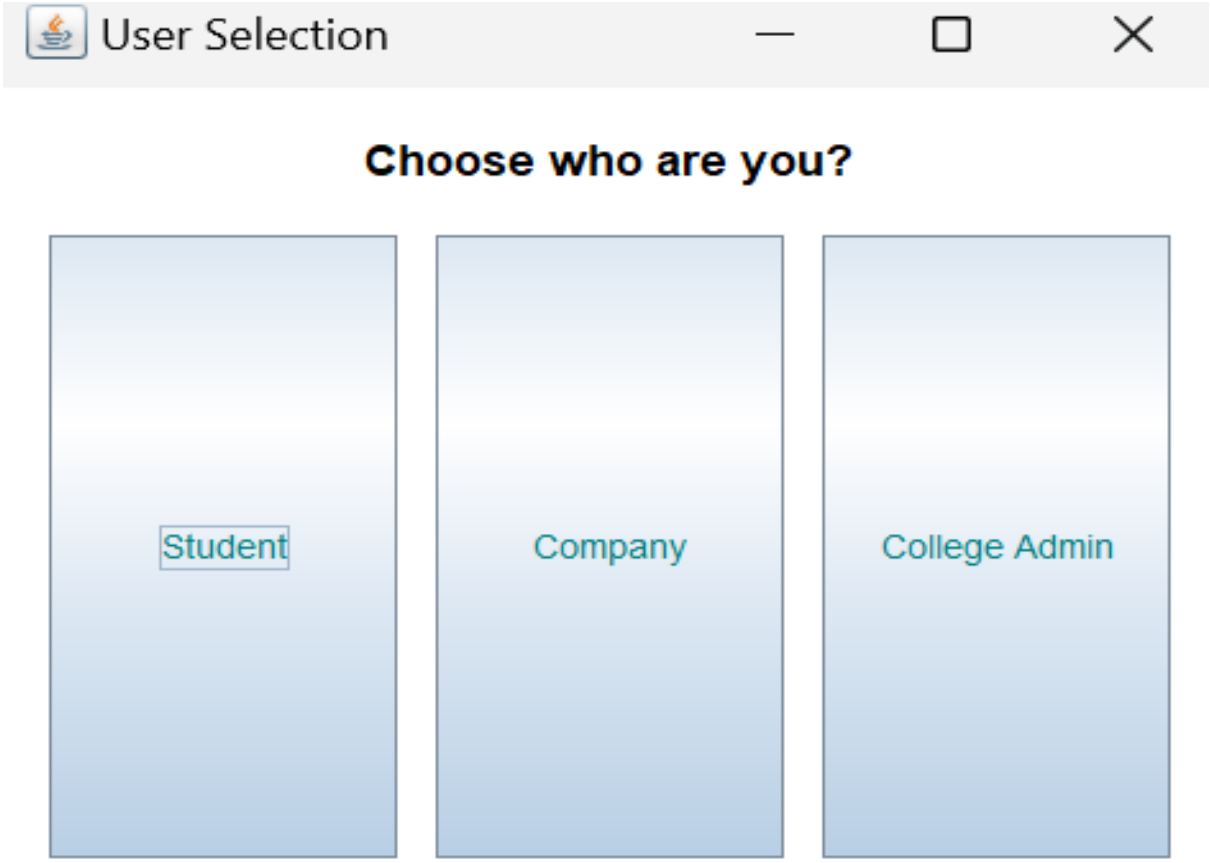


Admin: Personal info



Home page code

```
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SelectionScreen extends JFrame {

    public SelectionScreen() {
        setTitle("User Selection");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null); // Center the window
        initializeComponents();
    }

    private void initializeComponents() {
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y_AXIS));
        mainPanel.setBorder(new EmptyBorder(20, 20, 20, 20));
        mainPanel.setBackground(Color.WHITE);

        // Title label
        JLabel titleLabel = new JLabel("Choose who are you?", SwingConstants.CENTER);
        titleLabel.setFont(new Font("Arial", Font.BOLD, 18));
        titleLabel.setForeground(Color.BLACK);
        titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
        mainPanel.add(titleLabel);
        mainPanel.add(Box.createVerticalStrut(20)); // Add spacing
    // Panel for the user type buttons
    JPanel buttonPanel = new JPanel(new GridLayout(1, 3, 15, 15));
    buttonPanel.setBackground(Color.WHITE);

    // Create and add the buttons with icons
    JButton studentButton = createButton("Student", "student_icon.png");
    JButton companyButton = createButton("Company", "company_icon.png");
    JButton adminButton = createButton("College Admin", "admin_icon.png");

    buttonPanel.add(studentButton);
    buttonPanel.add(companyButton);
    buttonPanel.add(adminButton);

    // Add button panel to main panel
    mainPanel.add(buttonPanel);

    add(mainPanel); // Add mainPanel to the frame's content pane
}

private JButton createButton(String text, String iconPath) {
    JButton button = new JButton(text);
    button.setFont(new Font("Arial", Font.PLAIN, 14));
    button.setForeground(new Color(0, 128, 128));
    button.setHorizontalTextPosition(SwingConstants.CENTER);
    button.setVerticalTextPosition(SwingConstants.BOTTOM);

    // button.setIcon(new ImageIcon(getClass().getResource(iconPath)));
        // Set up the button action to print the selection
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "You selected: " + text);
            }
        });

        return button;
    }

    public static void main(String[] args) {
        SelectionScreen frame = new SelectionScreen();
        frame.setVisible(true);
    }
}
```


User Selection
Choose who are you?
Student
Company
College Admin

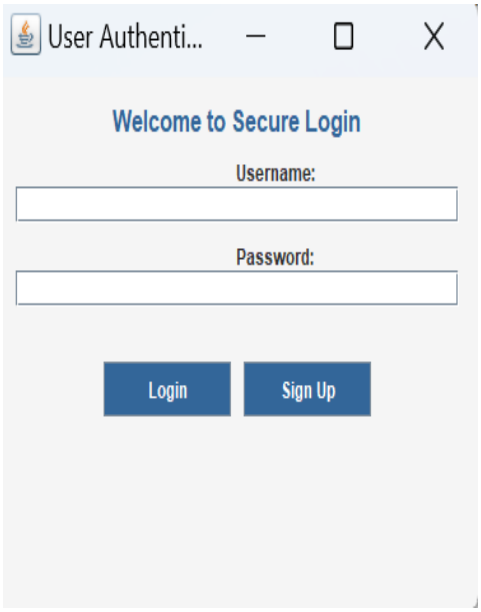
Sign up & login & store in DB code

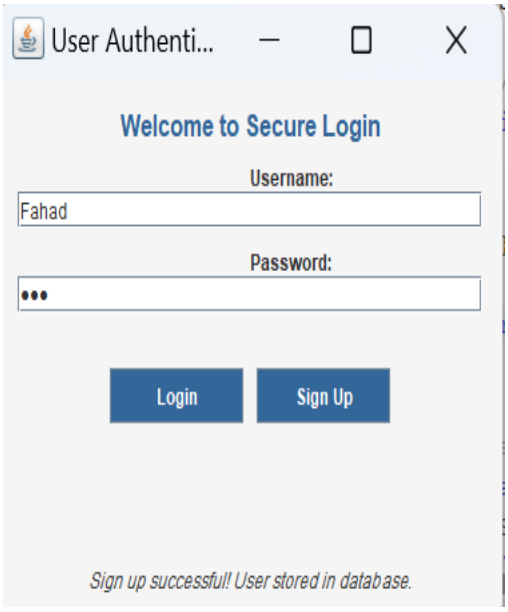
```
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashMap;

public class SimpleUI extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JLabel statusLabel;

    // HashMap to simulate the database for storing user data
    private HashMap<String, String> simulatedDatabase = new HashMap<>();

    public SimpleUI() {
        setTitle("User Authentication");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);  // Center the window
        initializeComponents();
    }

    // Method to initialize UI components with simplified layout
    private void initializeComponents() {
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.Y_AXIS));
        mainPanel.setBorder(new EmptyBorder(15, 15, 15, 15));
        mainPanel.setBackground(new Color(245, 245, 245));
```

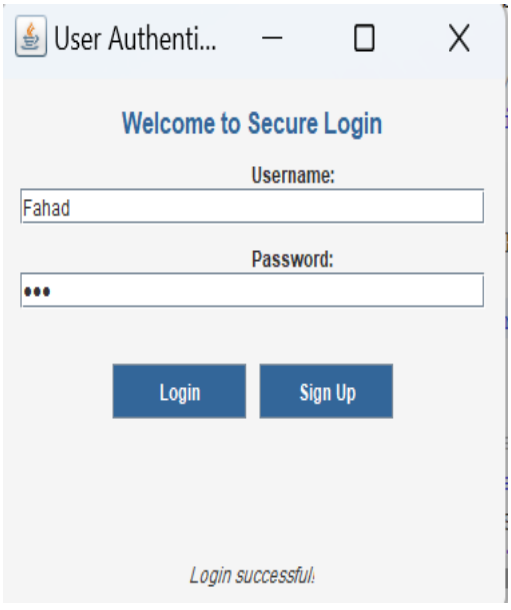
```
// Title label with custom font
JLabel titleLabel = new JLabel("Welcome to Secure Login", SwingConstants.CENTER);
titleLabel.setFont(new Font("Arial", Font.BOLD, 16));
titleLabel.setForeground(new Color(51, 102, 153));
titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
mainPanel.add(titleLabel);
mainPanel.add(Box.createVerticalStrut(10));  // Add spacing

// Username label and field
JLabel usernameLabel = new JLabel("Username:");
usernameLabel.setAlignmentX(Component.LEFT_ALIGNMENT);
mainPanel.add(usernameLabel);

usernameField = new JTextField();
usernameField.setMaximumSize(new Dimension(Integer.MAX_VALUE, usernameField.getPreferredSize().height));
mainPanel.add(usernameField);
mainPanel.add(Box.createVerticalStrut(10));  // Add spacing

// Password label and field
JLabel passwordLabel = new JLabel("Password:");
passwordLabel.setAlignmentX(Component.LEFT_ALIGNMENT);
mainPanel.add(passwordLabel);

passwordField = new JPasswordField();
passwordField.setMaximumSize(new Dimension(Integer.MAX_VALUE, passwordField.getPreferredSize().height));
mainPanel.add(passwordField);
mainPanel.add(Box.createVerticalStrut(20));  // Add spacing
```

```
// Button panel with Login and Sign-Up buttons
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
buttonPanel.setOpaque(false);

JButton loginButton = new JButton("Login");
styleButton(loginButton);
loginButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        login();
    }
});

JButton signUpButton = new JButton("Sign Up");
styleButton(signUpButton);
signUpButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        signUp();
    }
});

buttonPanel.add(loginButton);
buttonPanel.add(signUpButton);
mainPanel.add(buttonPanel);
mainPanel.add(Box.createVerticalStrut(10));  // Add spacing
    // Status label to display messages
    statusLabel = new JLabel(" ", SwingConstants.CENTER);
    statusLabel.setFont(new Font("Arial", Font.ITALIC, 12));
    statusLabel.setForeground(Color.DARK_GRAY);
    statusLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    mainPanel.add(statusLabel);

    add(mainPanel); // Add mainPanel to the frame's content pane
}

// Utility method to style buttons
private void styleButton(JButton button) {
    button.setBackground(new Color(51, 102, 153));
    button.setForeground(Color.WHITE);
    button.setFocusPainted(false);
    button.setFont(new Font("Arial", Font.BOLD, 12));
    button.setPreferredSize(new Dimension(100, 30));
}

// Method to get user input and validate
private String[] getUserInput() {
    String username = usernameField.getText().trim();
    String password = new String(passwordField.getPassword()).trim();  // Convert password to String

    // Validate input: Ensure both fields are filled
    if (username.isEmpty() || password.isEmpty()) {
        statusLabel.setText("Username and Password cannot be empty.");
        return null;
    }
    return new String[]{username, password};
}

// Method to handle login functionality
private void login() {
    String[] userDataInput = getUserInput();
    if (userDataInput == null) return;  // Exit if validation fails

    String username = userDataInput[0];
    String password = userDataInput[1];

    // Check if user exists and password matches in simulated "database"
    if (simulatedDatabase.containsKey(username) && simulatedDatabase.get(username).equals(password)) {
        statusLabel.setText("Login successful!");
    } else if (simulatedDatabase.containsKey(username)) {
        statusLabel.setText("Incorrect password.");
    } else {
        statusLabel.setText("User not found. Please sign up.");
    }
}

// Method to handle sign up functionality
private void signUp() {
    String[] userDataInput = getUserInput();
    if (userDataInput == null) return;  // Exit if validation fails

    String username = userDataInput[0];
    String password = userDataInput[1];
```

```
        // Check if user already exists in simulated "database"
        if (simulatedDatabase.containsKey(username)) {
            statusLabel.setText("User already exists. Please log in.");
        } else {
            // Store the new user in the simulated "database"
            sendToDatabase(username, password);  // Call the simulated database method
        }
    }

    // Simulated method to "send" data to a database
    private void sendToDatabase(String username, String password) {
        // "Store" the data in the HashMap
        simulatedDatabase.put(username, password);

        // Display a success message
        statusLabel.setText("Sign up successful! User stored in database.");

        // Simulate database operation feedback (optional)
        System.out.println("Simulated Database: Stored user -> Username: " + username + ", Password: " + password);
    }

    public static void main(String[] args) {
        SimpleUI frame = new SimpleUI();
        frame.setVisible(true);
    }
}
```



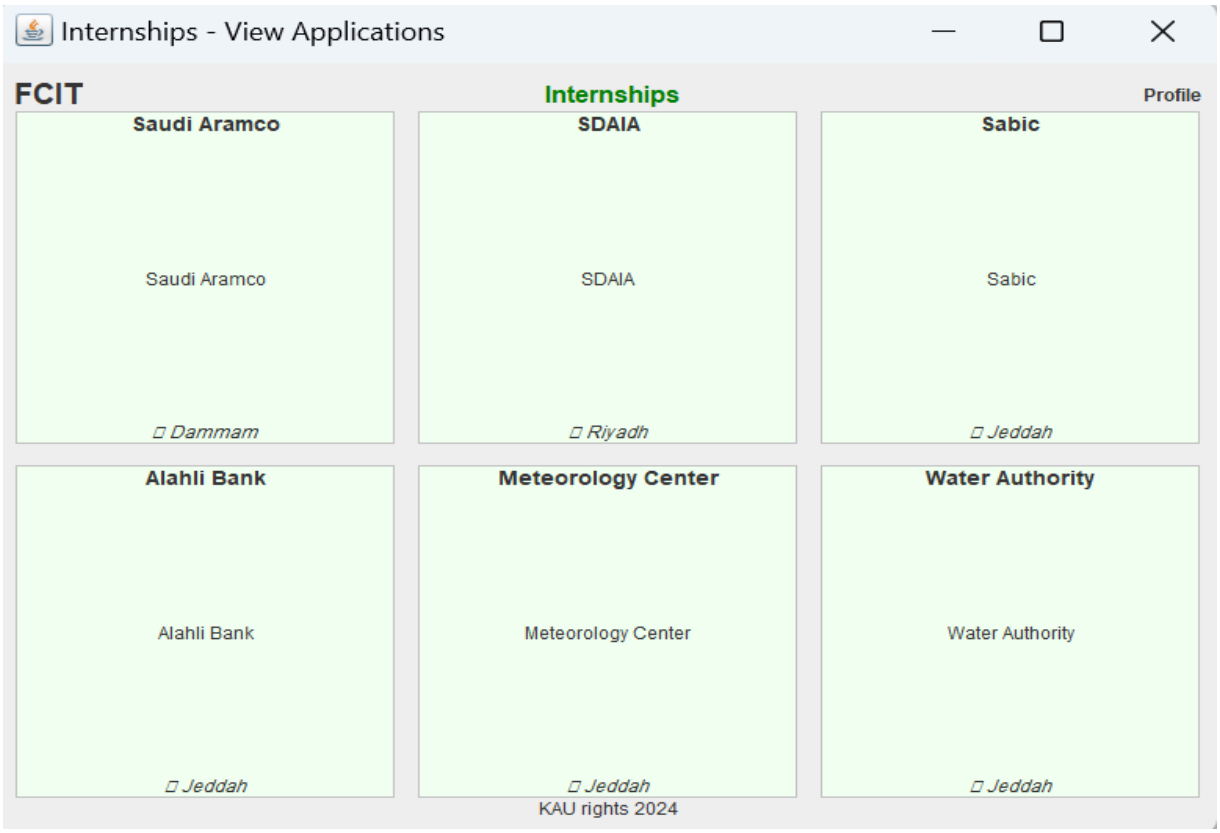




View Application Code

```
public class InternshipApplicationInterface extends JFrame {

    public InternshipApplicationInterface() {
        setTitle("Internships - View Applications");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null); // Center the window
        initializeComponents();
    }

    private void initializeComponents() {
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BorderLayout());
        mainPanel.setBorder(new EmptyBorder(10, 10, 10, 10));

        // Top bar with logo, title, and profile icon
        JPanel topBar = new JPanel(new BorderLayout());
        JLabel logoLabel = new JLabel("FCIT"); // Placeholder for logo, replace with an actual icon if available
        logoLabel.setFont(new Font("Arial", Font.BOLD, 20));
        topBar.add(logoLabel, BorderLayout.WEST);

        JLabel titleLabel = new JLabel("Internships", SwingConstants.CENTER);
        titleLabel.setFont(new Font("Arial", Font.BOLD, 16));
        titleLabel.setForeground(new Color(0, 128, 0));
        topBar.add(titleLabel, BorderLayout.CENTER);

        JLabel profileIcon = new JLabel("Profile"); // Placeholder for profile icon, replace with an actual icon
        profileIcon.setHorizontalAlignment(SwingConstants.RIGHT);
        topBar.add(profileIcon, BorderLayout.EAST);

        mainPanel.add(topBar, BorderLayout.NORTH);

        // Search and filter panel
        JPanel searchPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT, 10, 10));
        JTextField searchField = new JTextField(20);
        JButton searchButton = new JButton("Search");
        JButton sortButton = new JButton("Sort");
        JButton filterButton = new JButton("Filter");

        searchPanel.add(searchField);
        searchPanel.add(searchButton);
        searchPanel.add(sortButton);
        searchPanel.add(filterButton);

        mainPanel.add(searchPanel, BorderLayout.CENTER);

        // Company grid panel
        JPanel companyGrid = new JPanel(new GridLayout(2, 3, 15, 15));

        // Add sample companies
        companyGrid.add(createCompanyTile("Saudi Aramco", "Dammam"));
        companyGrid.add(createCompanyTile("SDAIA", "Riyadh"));
        companyGrid.add(createCompanyTile("Sabic", "Jeddah"));
        companyGrid.add(createCompanyTile("Alahli Bank", "Jeddah"));
        companyGrid.add(createCompanyTile("Meteorology Center", "Jeddah"));
        companyGrid.add(createCompanyTile("Water Authority", "Jeddah"));

        // Wrap the grid in a scroll pane if needed
        JScrollPane scrollPane = new JScrollPane(companyGrid);
        scrollPane.setBorder(null);
        mainPanel.add(scrollPane, BorderLayout.CENTER);

        // Footer
        JLabel footerLabel = new JLabel("KAU rights 2024", SwingConstants.CENTER);
        footerLabel.setFont(new Font("Arial", Font.PLAIN, 12));
        mainPanel.add(footerLabel, BorderLayout.SOUTH);

        add(mainPanel);
    }

    // Method to create a company tile
    private JPanel createCompanyTile(String name, String location) {
        JPanel companyTile = new JPanel();
        companyTile.setLayout(new BorderLayout());
        companyTile.setBorder(BorderFactory.createLineBorder(Color.LIGHT_GRAY));
        companyTile.setBackground(new Color(240, 255, 240));

        // Placeholder for company logo
        JLabel logoLabel = new JLabel(name, SwingConstants.CENTER);
        logoLabel.setFont(new Font("Arial", Font.PLAIN, 12));
        companyTile.add(logoLabel, BorderLayout.CENTER);

        // Company name and location
        JLabel nameLabel = new JLabel(name, SwingConstants.CENTER);
        nameLabel.setFont(new Font("Arial", Font.BOLD, 14));
        companyTile.add(nameLabel, BorderLayout.NORTH);

        JLabel locationLabel = new JLabel("📍 " + location, SwingConstants.CENTER);
```

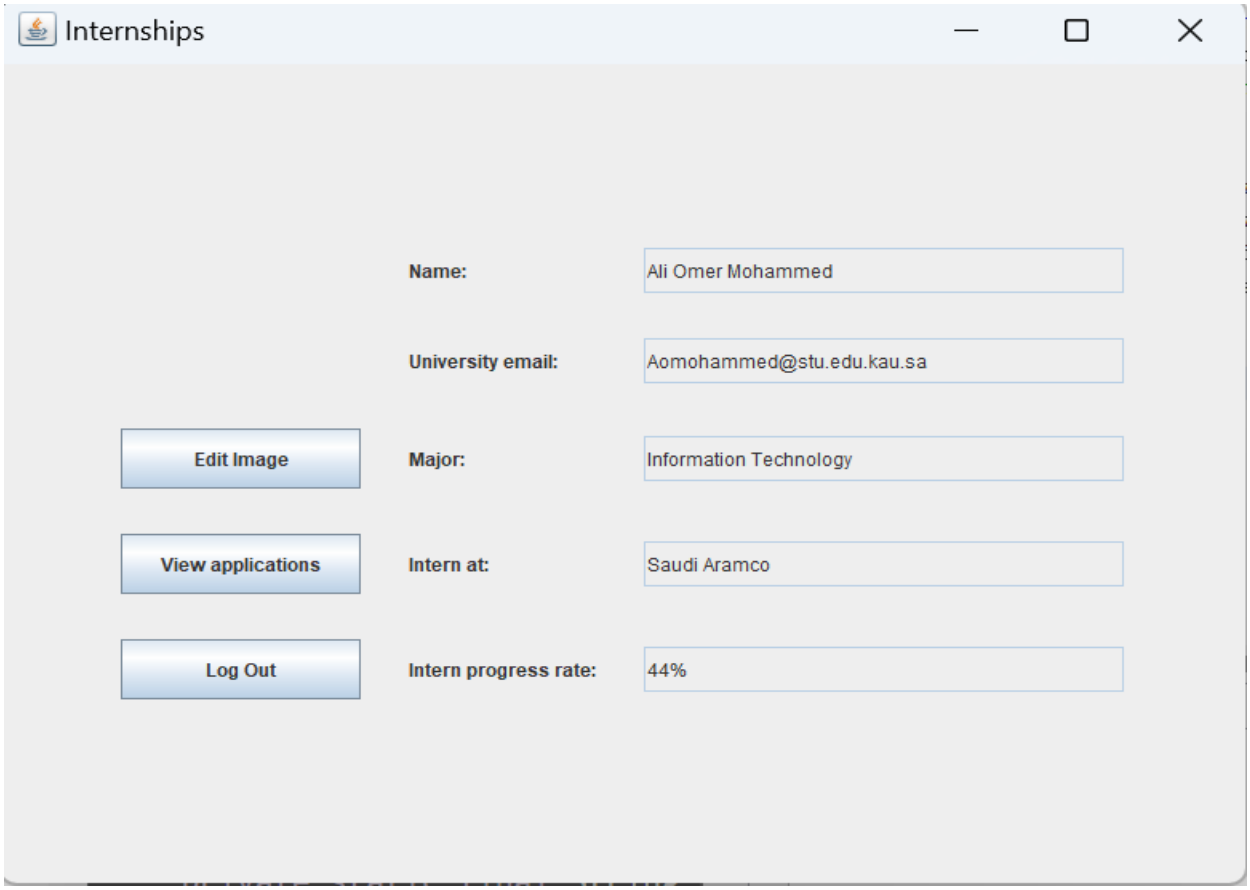
```
    JLabel locationLabel = new JLabel("📍 " + location, SwingConstants.CENTER);
    locationLabel.setFont(new Font("Arial", Font.ITALIC, 12));
    companyTile.add(locationLabel, BorderLayout.SOUTH);

    return companyTile;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        InternshipApplicationInterface frame = new InternshipApplicationInterface();
        frame.setVisible(true);
    });
}
```



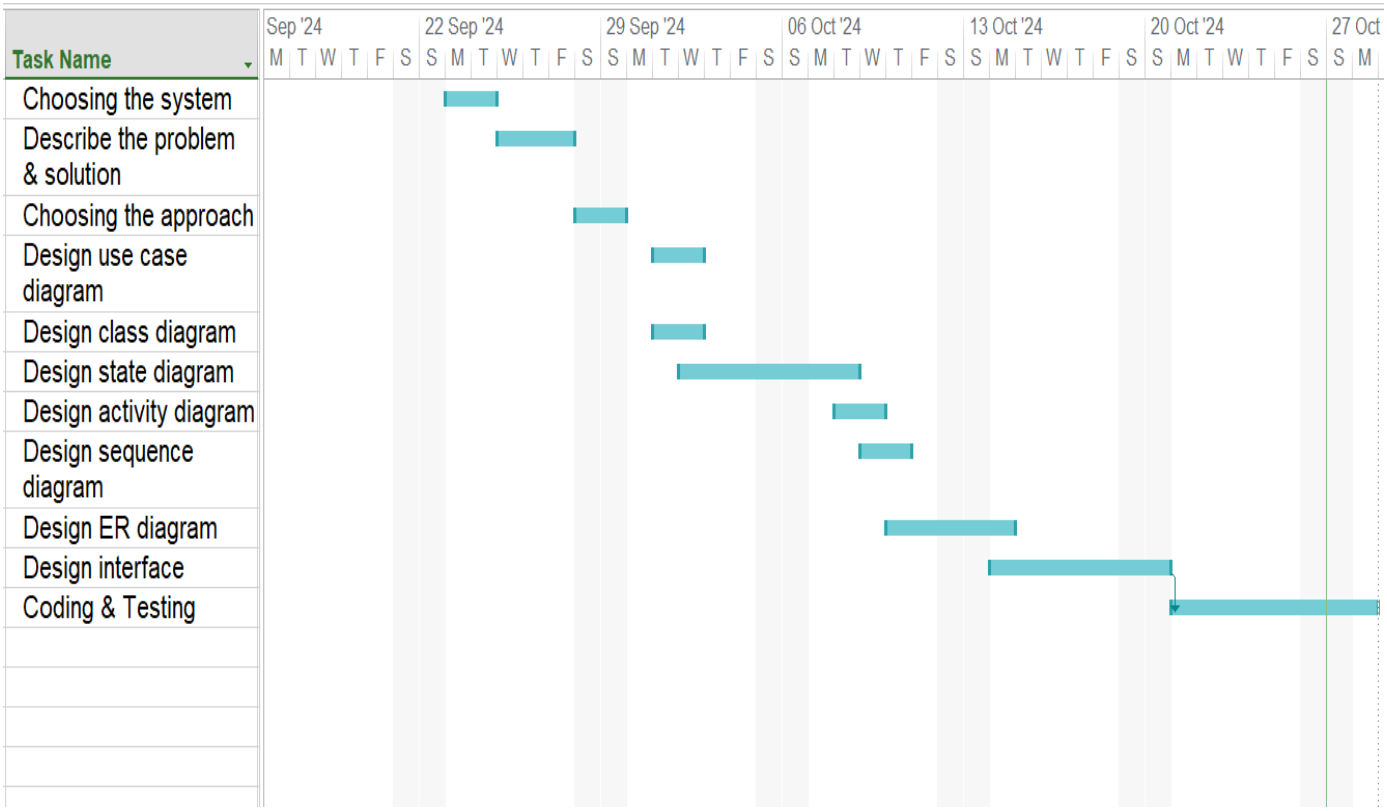
Personal Info Code

```
public class InternshipApplicationUI extends JFrame {
    private static final String TITLE = "Internships";
    private static final int FRAME_WIDTH = 800;
    private static final int FRAME_HEIGHT = 600;

    private JLabel profileImageLabel, nameLabel, emailLabel, majorLabel, internAtLabel, progressLabel;
    private JTextField nameField, emailField, majorField, internAtField, progressField;
    private JButton editImageButton, viewApplicationsButton, logoutButton;

    public InternshipApplicationUI() {
        setTitle(TITLE);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
        setLocationRelativeTo(null);
        setLayout(new GridBagLayout());

        initComponents();
    }

    private void initComponents() {
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(15, 15, 15, 15);
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridheight = 2;
        gbc.anchor = GridBagConstraints.CENTER;

        // Profile image
        ImageIcon profileImage = new ImageIcon("profile_image.png");
        profileImageLabel = new JLabel(profileImage);
        add(profileImageLabel, gbc);

        // Name
        gbc.gridx = 1;
        gbc.gridy = 0;
        gbc.gridheight = 1;
        gbc.anchor = GridBagConstraints.WEST;
        nameLabel = new JLabel("Name:");
        add(nameLabel, gbc);

        gbc.gridx = 2;
        nameField = new JTextField("Ali Omer Mohammed");
        nameField.setEditable(false);
        nameField.setPreferredSize(new Dimension(300, 30));
        add(nameField, gbc);

        // University email
        gbc.gridx = 1;
        gbc.gridy = 1;
        emailLabel = new JLabel("University email:");
        add(emailLabel, gbc);

        gbc.gridx = 2;
        emailField = new JTextField("Aomohammed@stu.edu.kau.sa");
        emailField.setEditable(false);
        emailField.setPreferredSize(new Dimension(300, 30));
        add(emailField, gbc);
```

```
    // Major
    gbc.gridx = 1;
    gbc.gridy = 2;
    majorLabel = new JLabel("Major:");
    add(majorLabel, gbc);

    gbc.gridx = 2;
    majorField = new JTextField("Information Technology");
    majorField.setEditable(false);
    majorField.setPreferredSize(new Dimension(300, 30));
    add(majorField, gbc);

    // Intern at
    gbc.gridx = 1;
    gbc.gridy = 3;
    internAtLabel = new JLabel("Intern at:");
    add(internAtLabel, gbc);

    gbc.gridx = 2;
    internAtField = new JTextField("Saudi Aramco");
    internAtField.setEditable(false);
    internAtField.setPreferredSize(new Dimension(300, 30));
    add(internAtField, gbc);

    // Intern progress rate
    gbc.gridx = 1;
    gbc.gridy = 4;
    progressLabel = new JLabel("Intern progress rate:");
    add(progressLabel, gbc);
    progressField = new JTextField("44%");
    progressField.setEditable(false);
    progressField.setPreferredSize(new Dimension(300, 30));
    add(progressField, gbc);

    // Buttons
    gbc.gridx = 0;
    gbc.gridy = 2;
    gbc.gridwidth = 1;
    editImageButton = new JButton("Edit Image");
    editImageButton.setPreferredSize(new Dimension(150, 40));
    add(editImageButton, gbc);

    gbc.gridy = 3;
    viewApplicationsButton = new JButton("View applications");
    viewApplicationsButton.setPreferredSize(new Dimension(150, 40));
    add(viewApplicationsButton, gbc);

    gbc.gridy = 4;
    logoutButton = new JButton("Log Out");
    logoutButton.setPreferredSize(new Dimension(150, 40));
    add(logoutButton, gbc);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        InternshipApplicationUI frame = new InternshipApplicationUI();
        frame.setVisible(true);
    });
}
```


Internships
Name:
Ali Omer Mohammed
University email:
Aomohammed@stu.edu.kau.sa
Edit Image
Major:
Information Technology
View applications
Intern at:
Saudi Aramco
Log Out
Intern progress rate:
44%

Gantt Chart


Task Name
Choosing the system
Describe the problem & solution
Choosing the approach
Design use case diagram
Design class diagram
Design state diagram
Design activity diagram
Design sequence diagram
Design ER diagram
Design interface
Coding & Testing
Sep '24
22 Sep '24
29 Sep '24
06 Oct '24
13 Oct '24
20 Oct '24
27 Oct
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M

Conclusion

This project addresses the critical issue of limited accessibility and poor communication in the internship landscape. The proposed solution aims to create a centralized platform that streamlines the internship discovery process, facilitates seamless communication, and fosters stronger connections between students, educational institutions, and companies.

The user-friendly interface, comprehensive internship database, and integrated communication tools empower students to easily access relevant opportunities, enhancing their career preparedness. The Agile approach ensures the system remains flexible and adaptable to evolving stakeholder needs, enabling quick wins and timely course corrections.

By bridging the gap between students and companies, this project has the potential to create a more inclusive and equitable internship ecosystem, leading to better career outcomes for students and more successful talent acquisition for companies.

Group Members	Roles
Ahmed Abdulrahman Alhuthyfi	**Describe the problem & solution, design use case, State, ER diagrams, UI/UX interface.**
Abdulaziz Mohammed Shaheen	**Describe the problem & solution, design class, activity, ER diagrams, UI/UX interface.**
Nawaf Othman Alghamdi	**Choosing the approach, design use case, activity diagrams, prepare PowerPoint slides.**
Fahad Abdullah Alhawas	**Design state, use case diagrams, Coding & Testing, Database implementation.**

References:

[1] "What are some of the challenges of doing an internship?," Typeset, [Online]. https://typeset.io/questions/what-are-some-of-the-challenges-of-doing-an-internship-4dr33cf22m.

[2] "Common challenges of interns and grads," Atlassian, [Online]. https://www.atlassian.com/company/careers/resources/career-growth/common-challenges-of-interns-and-grads.