

CS 487 Final Report

University Portal

Tanya Prabhakar and [Elijah Wendel](#)

June 22, 2022

SECTION 1 - Initial Design

The initial design consisted of an application for the university that handles the health and safety, job search opportunities and managing time or assignments. The program will provide information and details about the health and safety for students, staff and faculty of the institution. This consists of improving student wellness through giving location, working hours, basic information about health, and numbers to call in case of queries or emergencies. Another feature is job searching, which can help students jump-start their career and provide resources for searching for jobs. This will show details of career fairs on and off campus with a time, location and companies that will be present. In addition, it will serve as a connection to communicate with companies that work closely with the institutions or are searching for new college employees. The application will also help with student's assignment and time management by providing study resources, ways to destress, exam study tips, and time management tips. In addition, it will provide the users with access to their course, in which professors can post content, assignments that are due, and discussion boards. The students can interact with the content and submit their assignments through this portal.

SECTION 2 - Updates

Section 2.1 Requirements:

Functional Requirements:

Functional Requirements	Original Idea	Final Result
Purpose	The purpose of the system is to assist users in a university with health and safety, job search opportunities, and assignment and time management issues.	This requirement was satisfied in that the user has a basic functionality of the major features, but not the full functionality of the app.
Health and Safety	Health and safety should have resources for users about safety incidents that happen on-campus, and general health and wellbeing tips.	Health and safety has phone numbers and information about medical and public safety help.
Job search	The job search should have resources about career fairs, and other job opportunities for students.	Job search lists companies with their name, if they are hiring and their salary. Does not have an option to apply.
View assignment and time	The system should have	Did not satisfy the requirement.

Functional Requirements	Original Idea	Final Result
management	information for student users regarding assignment information, which would be similar to Blackboard. The system should also have time management resources for students.	
Error Handling	The application should be able to handle errors, such as users entering a wrong password. Other errors might be if the system crashes, the system should send a dialog message to the user explaining the crash.	Error handling like incorrect typing of passwords is met.
Submitting assignments	Students can use the application to submit homework assignments, and online quizzes and tests. Administrators will also have access to do so in order to help with any errors.	Did not satisfy the requirement. Can only view courses.
Publishing information	Some staff can publish health and safety information, or job opportunities. Professors can publish content and assignments. Admins have the capability to publish anything and everything they want.	Did not satisfy the requirement.

Non-functional Requirements:

Attributes	Original Idea	Final Results
Security	Password protected accounts, and need access to view the content. Grades and assignments can only be changed by the professors. Information in the health and safety, and job search should only be changed by permitted staff.	Successfully secures the application from users with wrong or without accounts. Grades cannot be changed by anyone unless access to the database which is also password protected. Information for health and safety and job search cannot be changed.

Attributes	Original Idea	Final Results
Capacity	Should at least have the capacity of 20 thousand users.	Did not test for this capacity.
Compatibility	Compatible on smartphone, tablet or laptop.	Did not test, but should work for java supported systems.
Usability for Students	Students should be able to use the application to assist them with job opportunities, health & safety, and managing assignments and time.	Students are able to use it, but cannot submit assignments.
Usability for Staff	The Staff should be able to use the application to submit public safety incidents, and view the job opportunities and health & safety features.	The staff are not able to submit public safety incidents, they are able to view the job opportunities and health and safety tips.
Usability for Professors	The Professors should be able to use the application to post assignments and grade assignments. They should be able to use the job opportunities, health & safety, and managing assignments and time features.	Professors cannot post assignments, change grades but they can view students, courses, job search and health and safety information.

Section 2.2. Test plan/cases:

Test Plan	Initial Idea	Final Result
Ease of access	Program shall be straightforward	The program was easy for test users to use because of the easy to use menu and layout
Exception handling	User errors should be handled and avoid breaking down the system Errors from hardware should be handled, such as components... Errors from software such as design and code.	We had different errors such as sometimes the code was not working, and it took some time to make sure each user had the correct permissions for the app. For example, the student should not have permission to enter the professors tab.

Test Plan	Initial Idea	Final Result
Users log in screen	Users log in screen should have a username and password box for the users to enter their username and password. The password has to match the username. Username must be unique.	User was able to login successfully. This test case worked.
Users can log out	User log out screen should have a screen that says they have been logged out of the system.	User logs out and loses access to the application.
Users home page	The application should present three options to click on, health and safety, job search, and courses.	Home page displays four options instead of three, courses for students and professors, job search and health and safety.
Health and safety feature	Have two pages, one for health and another for safety. Each with information and numbers to call.	The health and safety page displayed different health tips and safety tips. The staff can't submit a public safety report. The administrator can't post a safety incident.
Job search feature	View job opportunities in a list which has the company name, positions available, and a button to apply, which allows the user to submit personal information and their CV.	The Job search page displayed a list of companies that users can apply to. Users can't apply to actual jobs.
Courses feature	The course feature would have the courses listed for each semester, along with the professor name, the course prerequisites, and the date/time of the course.	Two different panels, for students, show the courses with class name, grade and professor. For the professor it shows students in class and courses he/she is teaching.
System Administrator	The system administrator has logged in and is at the system administrator page. They have features to view activity of users, their personal information and credentials. The system shows the results of the transaction.	System administrators can log in and view all four features but cannot edit any of the features since that has not been implemented.

Test Plan	Initial Idea	Final Result
Graduate Student	The graduate student has logged in, and is at the graduate student home page. The graduate student registers for the classes they want for the semester. The graduate student submits their homework assignment. The system will show the results of the transactions.	The graduate student uses the student tab to view their current classes. The student can also use the health and safety tabs to view information about health and safety tips. Students can't register for classes.
Undergraduate Student	The undergraduate student has logged in, and is at the undergraduate student home page. The undergraduate student searches for job opportunities in the job search feature. The undergraduate student submits their online quiz. The system will show the results of the transaction.	The undergraduate student uses the student tab to view their current classes. The student can also use the health and safety tabs to view information about health and safety tips. Students can't register for classes.
Professor	The professor user has logged in and is at the professor home page. The professor posts an assignment. The professor grades an assignment. The system will show the results of the transaction.	Professor has access to courses for professors, job search and health and safety info. Does not have access to courses for students. Can't post assignments, discussions, or change grades.
Staff	The staff user has logged in and is at the staff home page. The staff files a public safety incident. The system will show the results of the transaction.	Staff can log in and only view job search and health and safety info, but cannot modify them.

Section 2.3 Design:

The health and public safety layout also changed from the design reports. Initially we thought it would be better to have the health and public safety reports on different pages in the app. We instead decided to make the health and public safety pages together in 1 page in the app. We thought this would actually be better for the user. If they are both included on the same page, it would allow the user to quickly read the public safety and health tips at the same time. If they were on different pages, the user

might only go to the health page without going to the public safety page.

We originally planned to have the Main Menu and the Sign Out buttons both on the right side upper corner. We thought it would be easier to have both buttons in the same location. We then decided to change the Main Menu button to the left side upper corner and keep the Sign Out button in the right side upper corner because we thought if both buttons are in the same corner, a user may accidentally press the Sign Out button instead of the Main Menu button or vice versa. Putting the buttons in the separate corners eliminates the problem of a user making a quick mistake of hitting the wrong button.

SECTION 3 - Test Results

Test results captured in evaluating your prototype

When we were evaluating the prototype, we noticed that we had to change the font sizes. Originally we had the font size too small. We decided to make it a bigger font so that the user would be able to more easily see the text.

After evaluating the prototype, we also realized we needed to change some of the text and background colors. We originally thought we should make the colors in the university's official school colors black and red. After the evaluation we decided to make the colors of the app more neutral and standardized colors of gray and light blue. This would be easier on the user's eyes.

Initially we had the main categories in the main menu listed vertically. We decided to change the layout of the main categories to horizontal. The horizontal view would be easier if the user is using the application on a desktop or laptop. Since this is a university portal filled with academic resources, we assumed most users would be accessing the application from a desktop or laptop instead of a cell phone, and so the horizontal view would be better to use.

The prototype was not able to show a fully functional application of the system designed, however it left a lot of room for scalability as buttons can be added easily to allow users to then modify text boxes and data entries such as more company listings, assignments and possibly in the future to register for courses.

SECTION 4 - Prototype Artifacts

SECTION 4.1 Source Code

Deny access to non professors:

```
JButton btnNewButton_1_3 = new JButton("Professors");
btnNewButton_1_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521/ORCL", "HR", "oracle");
            Statement st = conn.createStatement();
            BufferedReader reader = new BufferedReader(new FileReader("Admin.ser"));
            String customerID = reader.readLine();
            reader.close();
            ResultSet rs = st.executeQuery("select * from Professor");
            if(rs.next() && rs.getString("EMAIL").compareTo(customerID)==0) {
                new CoursesProf().setVisible(true);
                dispose();
            }
            else {
                JOptionPane.showMessageDialog(null, "Access Denied. You must be a professor to access the portal.");
            }
            st.close();
            conn.close();
        }
        catch (Exception r) {
            r.printStackTrace();
        }
    }
});
```

This code denies access to users who are not professors. This code uses JDBC which sets a connection between Java and the database server. It is similar to the login of the application however it does not take in user input and compare them to verify that the user has access. Instead, after the user logs in, the program stores the user's email which is a unique identifier and compares that with the list of professors. If the email is in the table then it is a professor and is granted access. An improvement could be to improve the security of this feature and to allow access to Admin's so they can enter this feature.

Display job search list:

```
String[][] data = new String[12][4];

try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521/ORCL", "HR", "oracle");
    PreparedStatement pst = conn.prepareStatement("select * from JOBS");
    ResultSet rs = pst.executeQuery();
    Statement st = conn.createStatement();
    int i = 0;
    while(rs.next()) {
        data[i][0] = rs.getString("ID");
        data[i][1] = rs.getString("COMPANY");
        data[i][2] = rs.getString("HIRING");
        data[i][3] = rs.getString("PAY");
        i++;
    }
    st.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
}

String[] columnNames = {"ID", "Company", "Hiring", "Pay"};

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(70, 131, 644, 238);
getContentPane().add(scrollPane);
table = new JTable(data, columnNames);
scrollPane.setViewportView(table);
table.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
    }
});
}
```

The code also uses JDBC to pull information from the database server and then assigns each attribute to a location on an array called “data”. Then each element of the array is displayed on a table that is created, which then uses a while loop to gather all of the information that the database sends with the query: select * from JOBS. It creates the table with a fixed size of 12x4 entries meaning 4 columns and 12 rows, with columns listed as “Id”, “Company”, “Hiring”, and “Pay”.

SQL code:

```
CREATE TABLE Students (
    ID int,
    Name Varchar(50),
    Email Varchar(50),
    Sex Varchar(8),
    PhoneNumber Int,
    Password Varchar(24),
    PRIMARY KEY (ID)
);
```

This code is an example of creating the Students table which has an id, name, email, sex, phonenumber and password associated with it. A possible improvement is more personal information and connecting this table with the courses table so that each student has a list of courses they are taking at the moment. This can be done using a foreign key of a list of ID's of the courses the student is in.

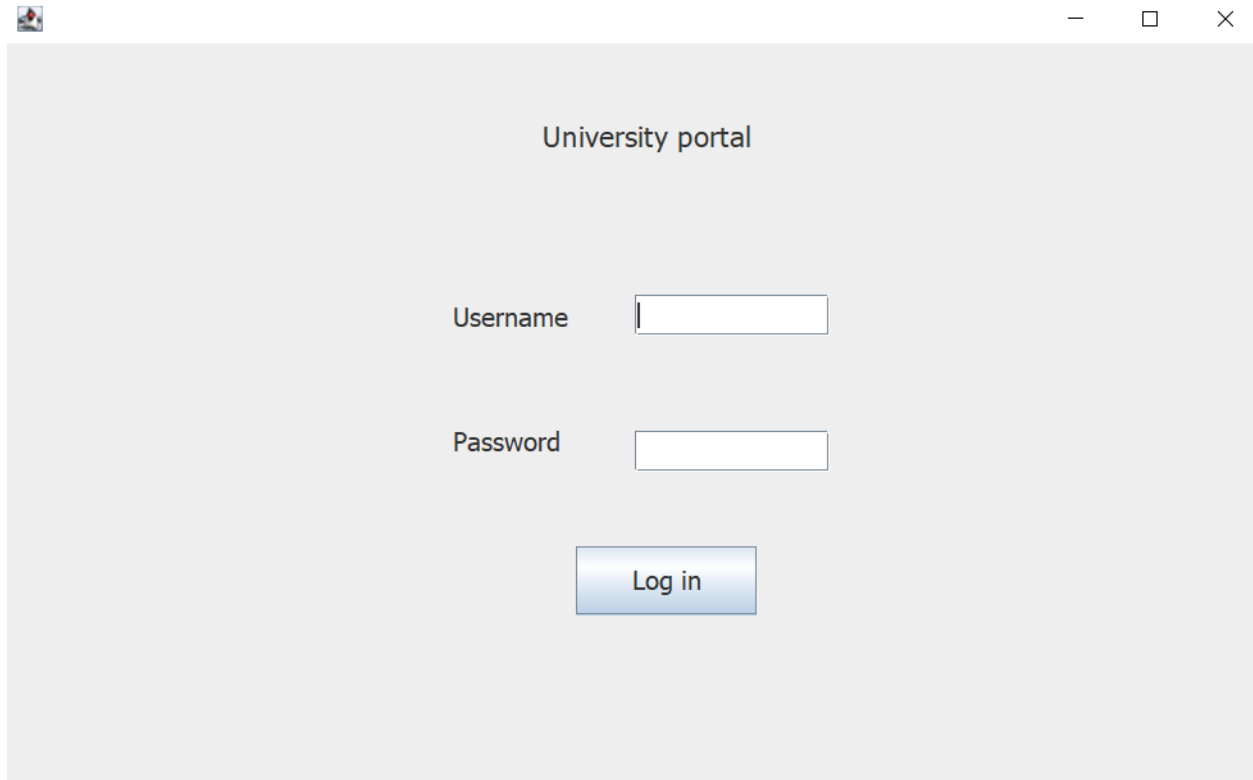
Display health and safety information:

```
JLabel lblNewLabel_2_1_1_1_1 = new JLabel();
lblNewLabel_2_1_1_1_1.setText("<html><hl align='center'>Contact Public Safety\r\n"
    + "PHONE\r\n"
    + "1.312.808.6363 (Emergency)\r\n"
    + "1.312.808.6300 (Non-Emergency)\r\n"
    + "EMAIL\r\n"
    + "publicsafety@iit.edu\r\n"
    + "DEPARTMENT TYPE\r\n"
    + "Administrative\r\n"
    + "OFFICE LOCATION\r\n"
    + "Public Safety\r\n"
    + "Tech Central\r\n"
    + "3424 South State Street, Suite 115\r\n"
    + "Chicago, IL 60616</hl>");
lblNewLabel_2_1_1_1_1.setBounds(435, 308, 308, 102);
contentPane.add(lblNewLabel_2_1_1_1_1);
```

This code is in the health and safety feature which has information stored in the project and displays it through a JLabel when the user clicks this feature. It uses html to display the information in an organized way. Possible improvements is to use an API to IIT's website and gather real time information and also make the page prettier and more organized. In addition, using a hyperlink to click on a phone number or email so the user could contact that person would also be in the future development.

SECTION 4.2 UI Screens and Documentation

Log in screen:

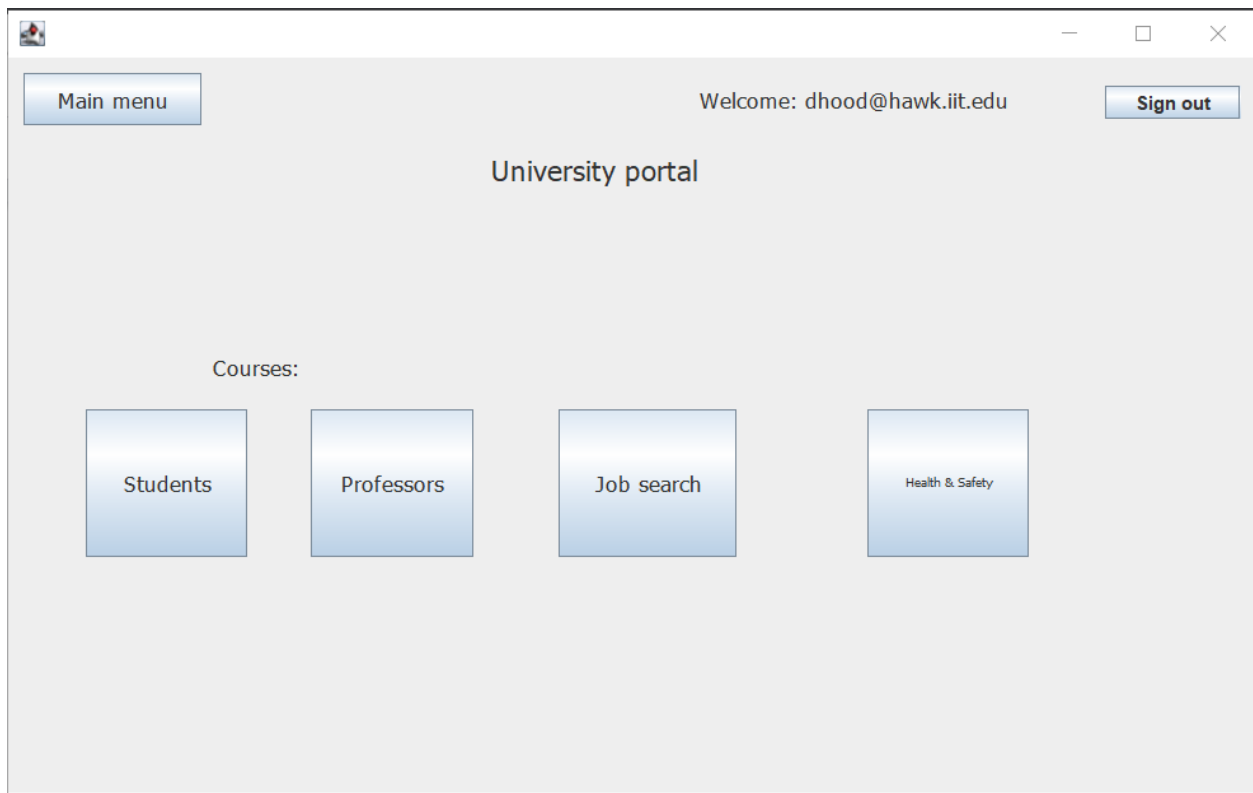


A screenshot of a web browser window titled "University portal". The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main content area is light gray and contains the following elements:

- The text "University portal" centered at the top.
- A label "Username" followed by a text input field.
- A label "Password" followed by a text input field.
- A blue "Log in" button centered below the input fields.

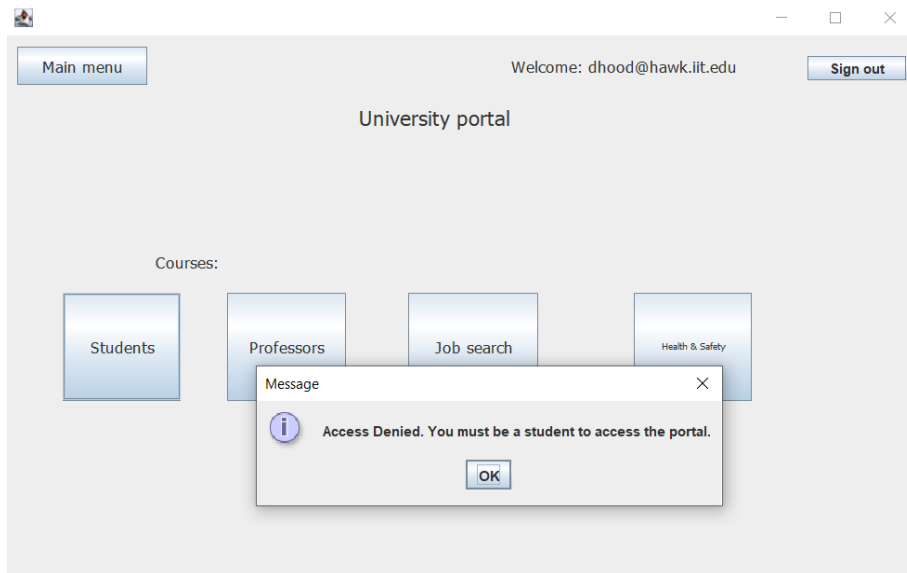
Users type their username and password and if they match the database then they are granted access. If not, a pop up prompt shows up telling them the information is incorrect and to try again.

Main menu:

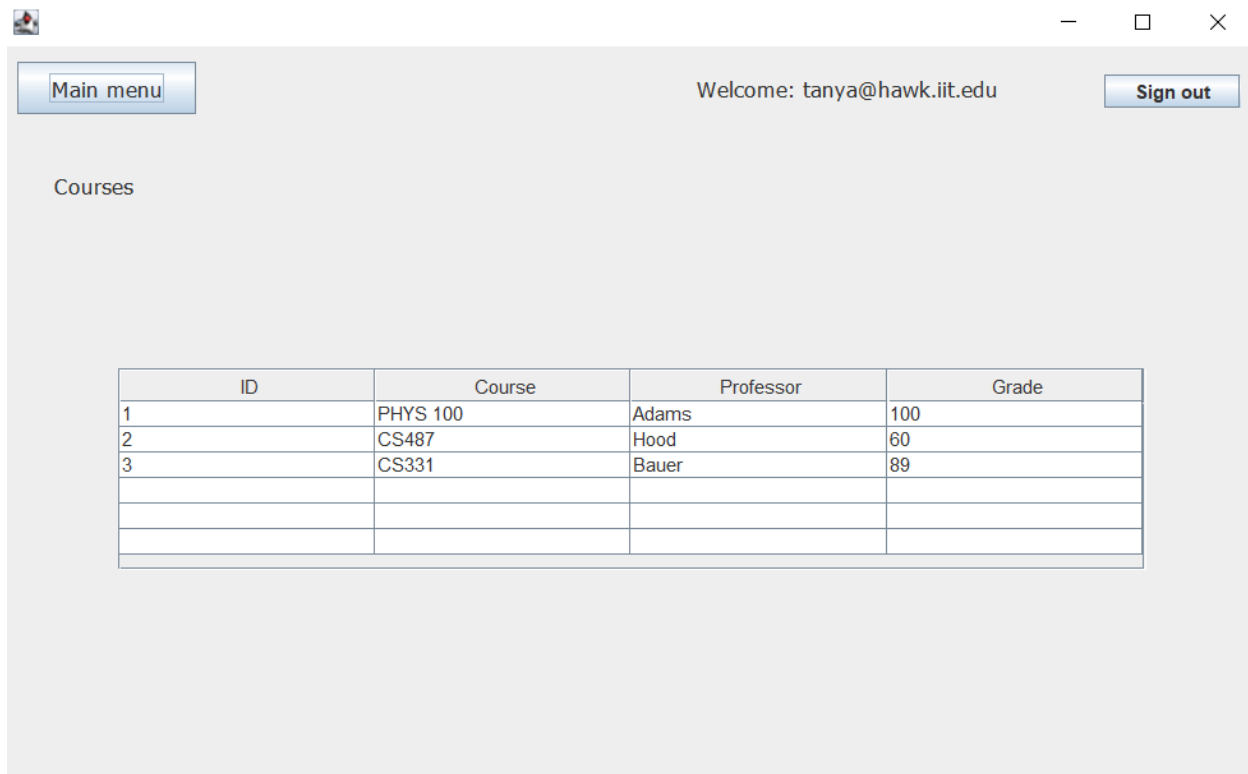


The main menu screen lists the main categories of the application: courses, job search, and health and safety. The Courses section has 2 subsections: one for students and professors. The students only have access to the student's tab, and the professors only have access to the professor's tab. A possible improvement is a calendar with the students classes and assignment dates and the ability to register for classes.

Courses:

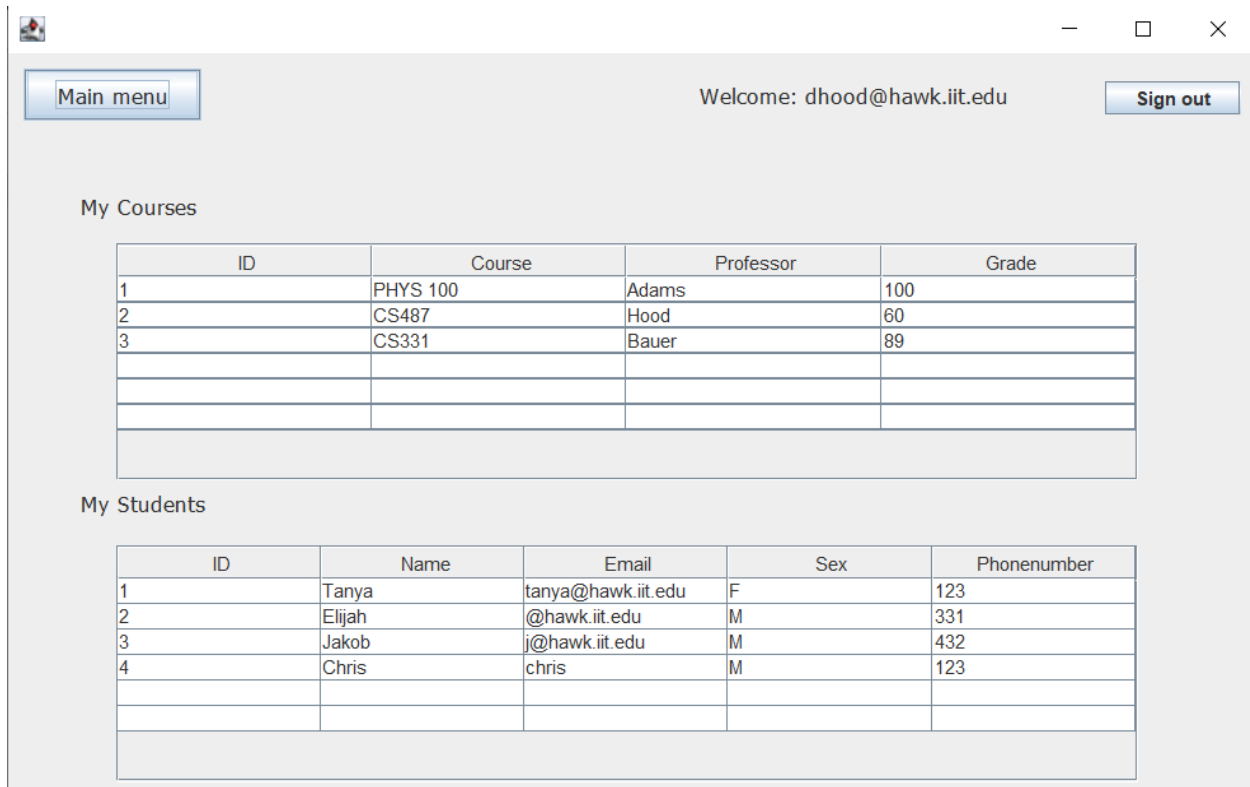


The above UI screen demonstrates a professor trying to access the student's portal, and the professor user will receive an error message stating that they must be a student to access the portal.



This is the students portal. The student portal will list the current courses the student is enrolled in. A scalability of this screen would include making it so that when the user clicks on a course, it takes the

user to a page with a more detailed description of the course, the building location, and time the course meets.



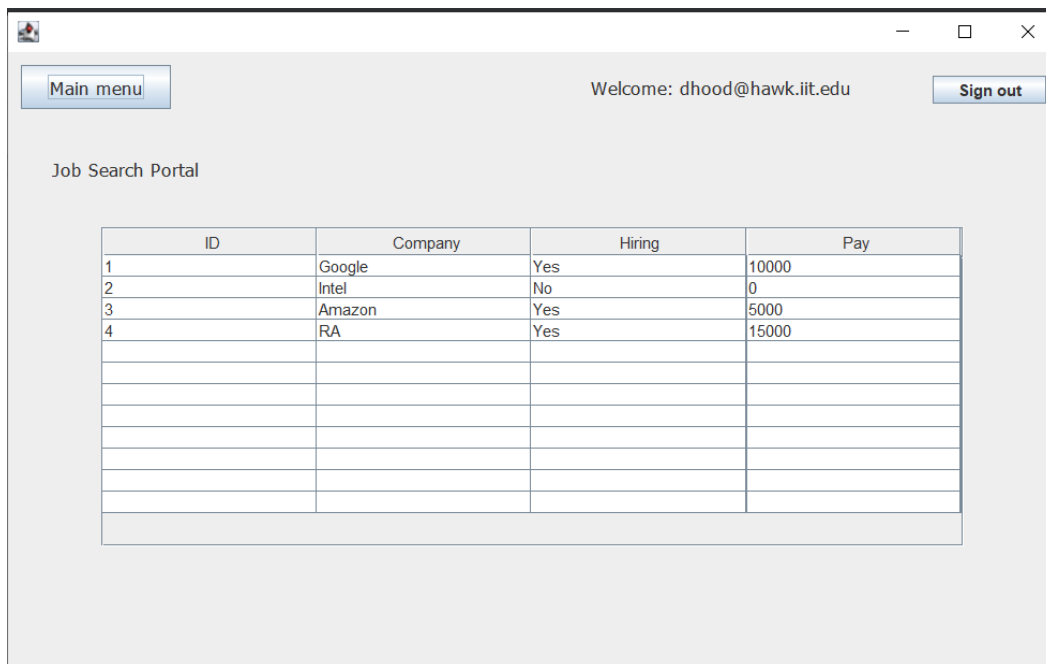
The screenshot shows a web application interface for a professor's portal. At the top left is a 'Main menu' button. At the top right, it says 'Welcome: dhood@hawk.iit.edu' next to a 'Sign out' button. Below the welcome message is a section titled 'My Courses' which contains a table with columns: ID, Course, Professor, and Grade. The table lists three courses: PHYS 100 (Adams, Grade 100), CS487 (Hood, Grade 60), and CS331 (Bauer, Grade 89). Below this is a section titled 'My Students' which contains a table with columns: ID, Name, Email, Sex, and Phonenumber. The table lists four students: Tanya (tanya@hawk.iit.edu, F, 123), Elijah (@hawk.iit.edu, M, 331), Jakob (j@hawk.iit.edu, M, 432), and Chris (chris, M, 123).

ID	Course	Professor	Grade
1	PHYS 100	Adams	100
2	CS487	Hood	60
3	CS331	Bauer	89

ID	Name	Email	Sex	Phonenumber
1	Tanya	tanya@hawk.iit.edu	F	123
2	Elijah	@hawk.iit.edu	M	331
3	Jakob	j@hawk.iit.edu	M	432
4	Chris	chris	M	123

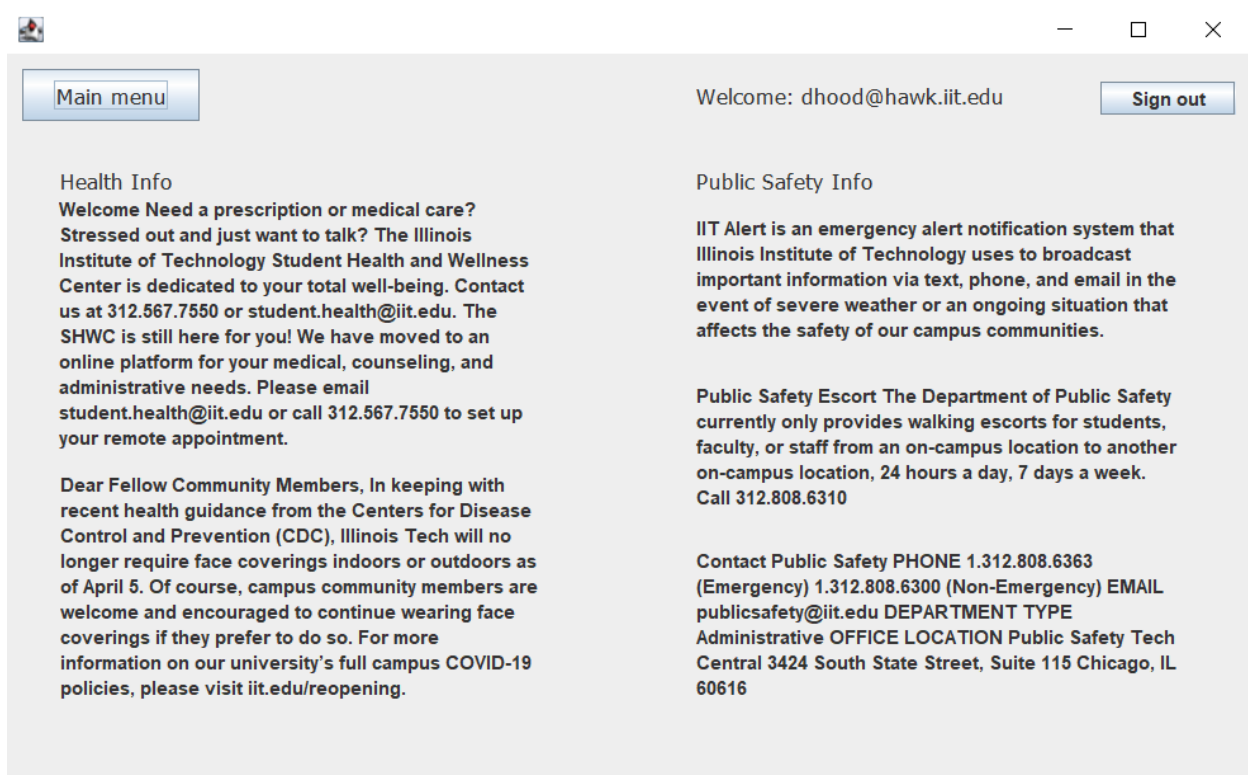
This is the professor's portal. The professor's portal has all the students in the class listed, along with the student ID, name, email, sex, and phone number. It also lists the courses the professor is teaching for the semester. A scalability for the feature is having an option for the professor to post an assignment, and submit a student's grade.

Job search:



This is the job search portal. It has the possible jobs that users may want to apply to. All users have access to the job search page. It lists the job ID, company, hiring, and pay. A scalability would be to have an option for the user to apply to the job.

Health and safety information page:



This is the health and safety page. It lists general health tips and public safety information about the university. A scalability feature would be to have any user submit a public safety report incident.

SECTION 4.3 Demonstration of Prototype

Two mp4 videos are included in the submission which first discusses the requirements, test plan and shows how the prototype works. The second explains the platforms and languages the prototype was developed on and some examples of code for certain features.