

User Stories:

Check Course List (main developer: Hongru Wang, Matthew)

- Course associated information including course number, course title, course grades and term taken.
- Priority: Medium
- Cost: 3-5 days

Check Course Prerequisite (main developer: Hongru Wang, Matthew, Dave)

- The prerequisites of a specific EECS courses are provided by the system.
- Priority: Medium
- Cost: 3-5 days

Academic Enrollment Record Database (main developer: Hongru Wang, Matthew)

- Create a user academic database to store the academic records of the student's courses.
- Priority- MEDIUM
- Cost: 3 days

Personal information Database (main developer: Hongru Wang, Matthew)

- Create a user database for managing the students' information including their login information, academic records, and student personal information.
- Priority: HIGH
- Cost: 5-7 days

Register user interface for student (main developer: Wu)

- Create a register user interface for students and admins to register to be a user of our system.
- Priority: HIGH
- Cost: 3 days

Login user interface for student (main developer: Wu)

- Create a login user interface for students and admins to login to access our system.
- Priority: HIGH
- Cost: 3 days

admin information database(main developer: Hongru Wang)

- Create a admin database for managing the school members' information including their login information, academic records, and admin's personal information.
- Priority: HIGH
- Cost: 5-7 days

Allow academic advisors to edit course taken(main developer: Hongru Wang, Matthew)

- add/drop course function provided for admin
- Priority:Meduim
- Cost: 2 days

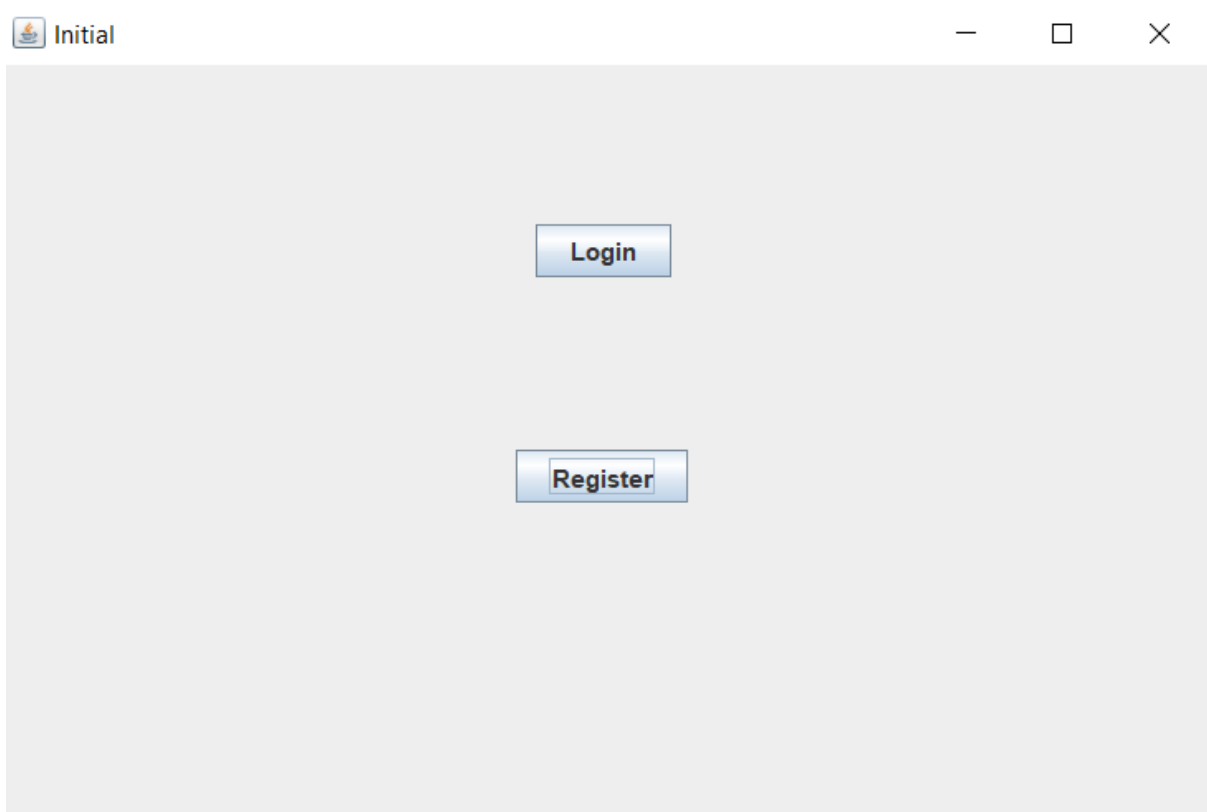
EECS department are able to edit course taken(main developer: Hongru Wang, Matthew)

- Advisor can add students to the course waitlist queue
- Priority: HIGH
- Cost: 3 days

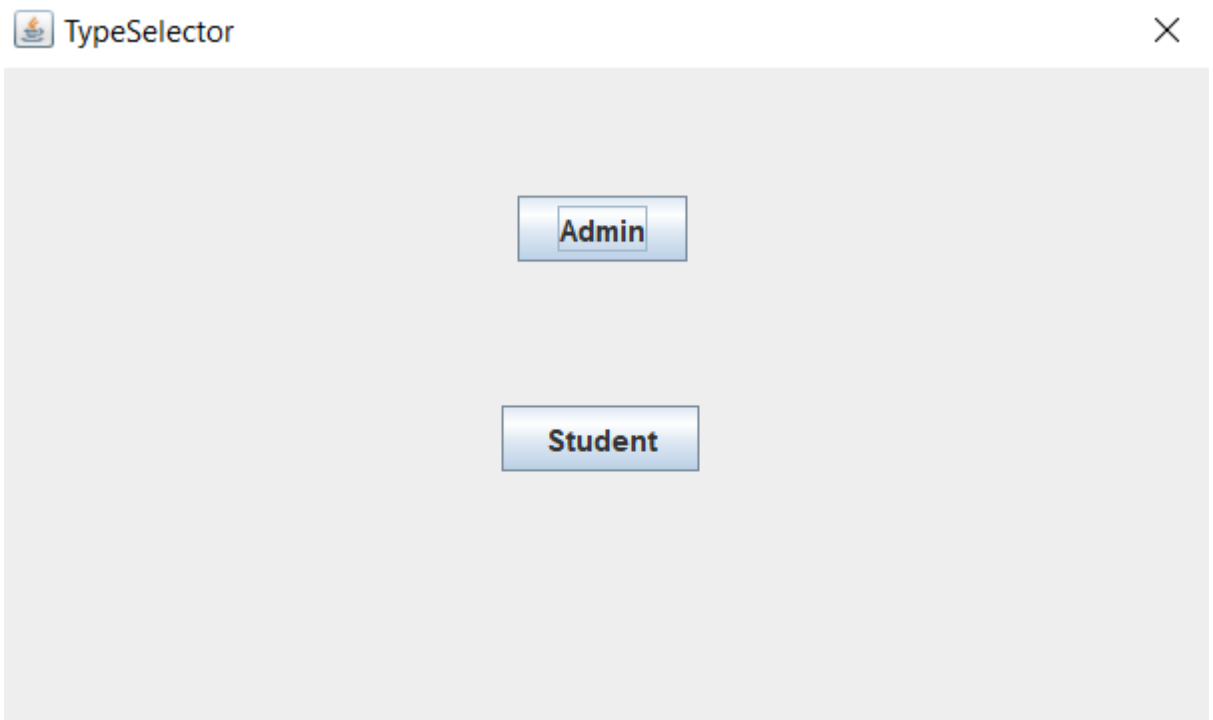
End-2-End test on Student Register User Interface:

Case 1: normal scenario

1. After running the application, the entry page UI is shown, click on the register button

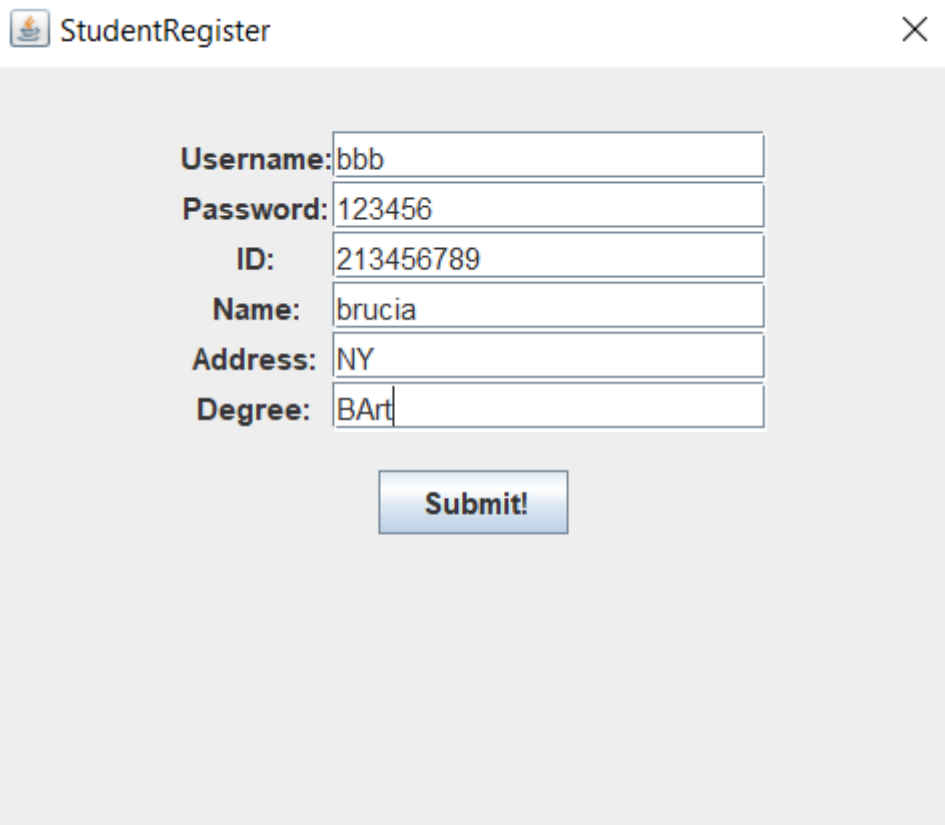


2. After click on the register button, then click on the student button



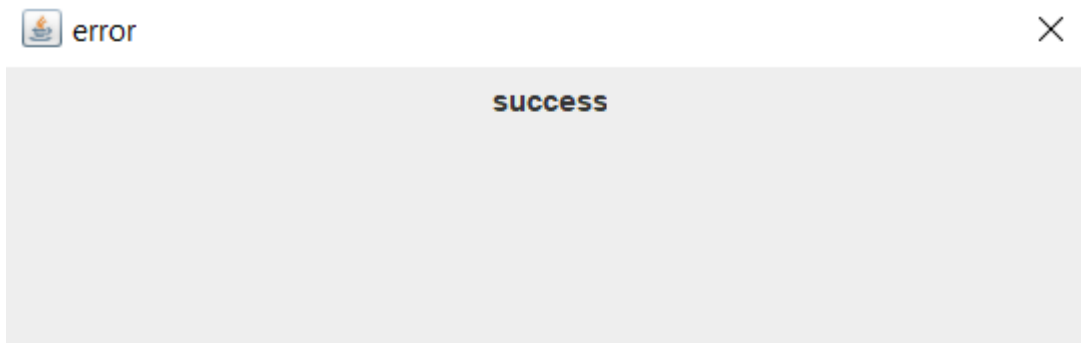
A screenshot of a Java Swing window titled "TypeSelector". The window has a standard title bar with a close button (X) in the top right corner. The main content area is light gray and contains two blue rectangular buttons with white text. The top button is labeled "Admin" and the bottom button is labeled "Student".

3. Input the new student information (normal case)



A screenshot of a Java Swing window titled "StudentRegister". The window has a standard title bar with a close button (X) in the top right corner. The main content area is light gray and contains a registration form. The form consists of six labels followed by text input fields: "Username:" with "bbb", "Password:" with "123456", "ID:" with "213456789", "Name:" with "brucia", "Address:" with "NY", and "Degree:" with "BArt". Below the form is a blue rectangular button with white text labeled "Submit!".

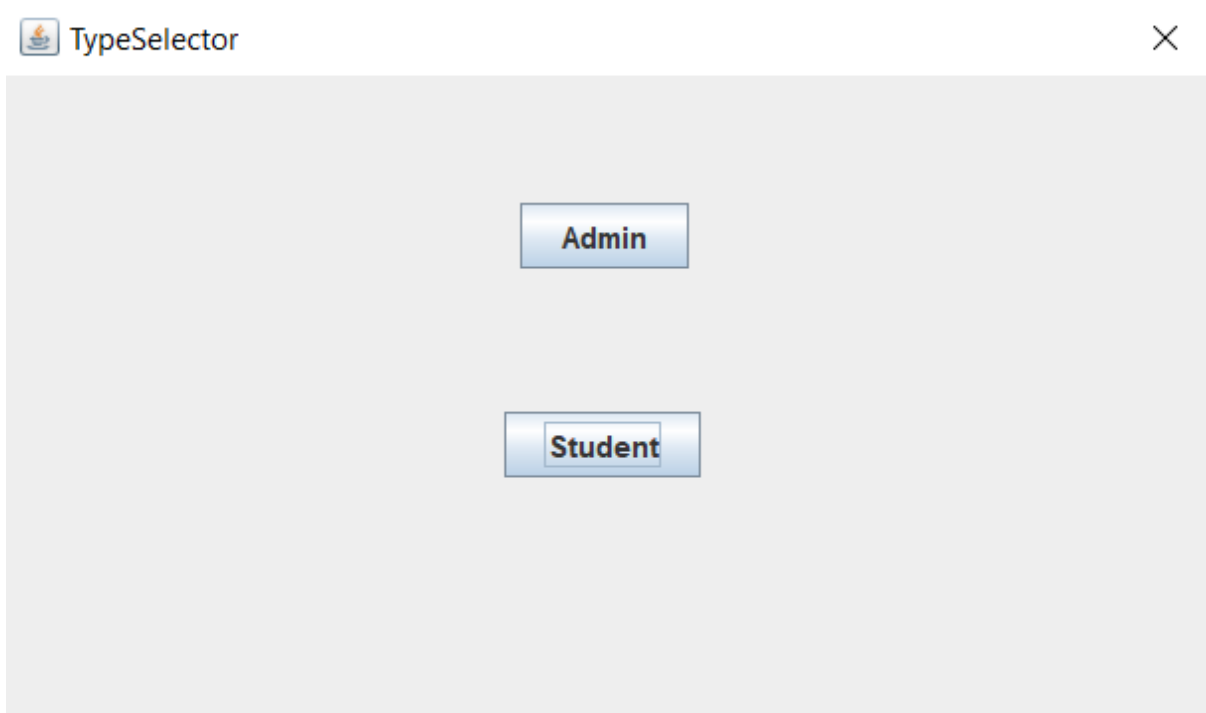
5.It shows success page



6.The record is added to the database

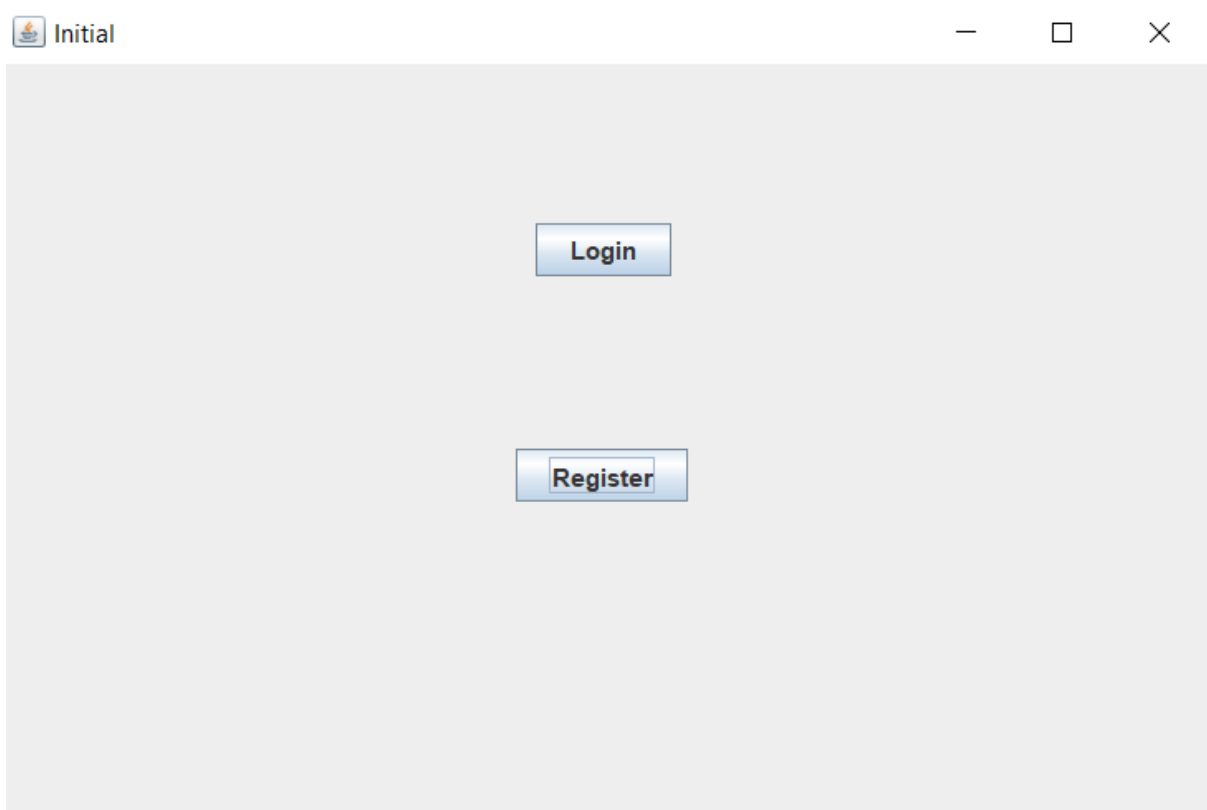
213456789	brucia	bbb	123456	NY	BArt
-----------	--------	-----	--------	----	------

7. Once you click on the X button, the program will return to the type selector page

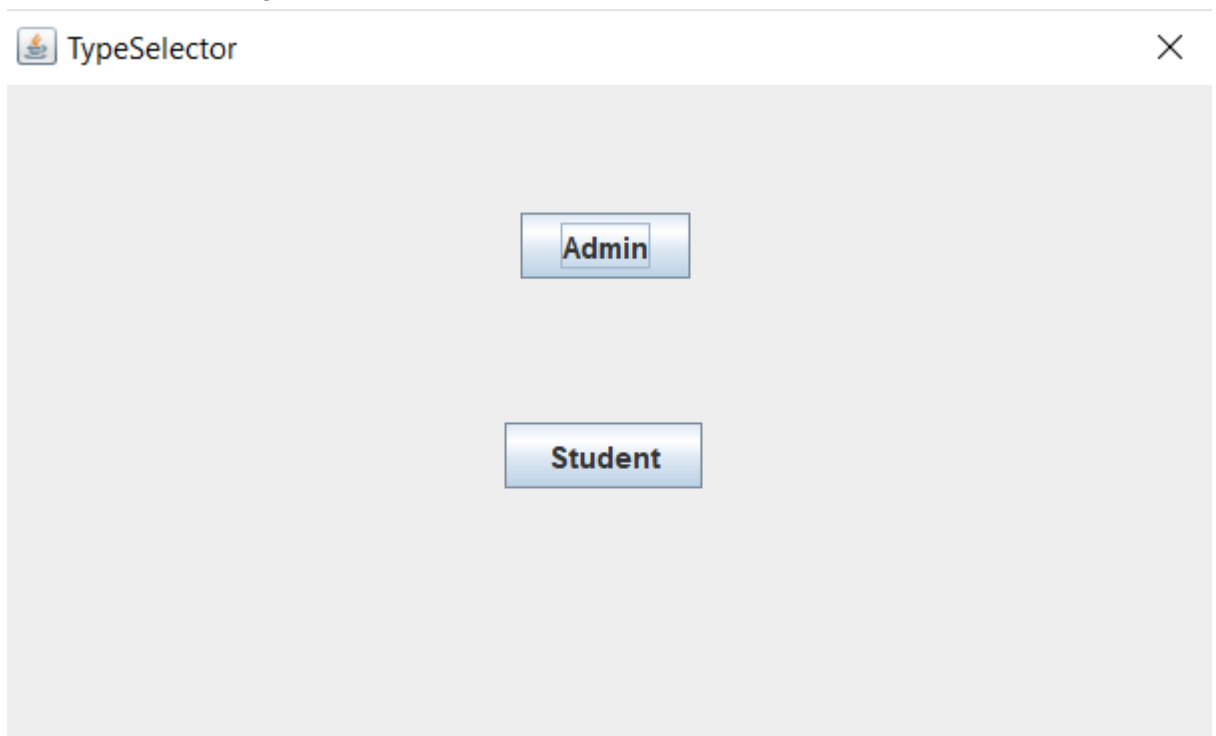


Case 2: inputting a duplicate ID

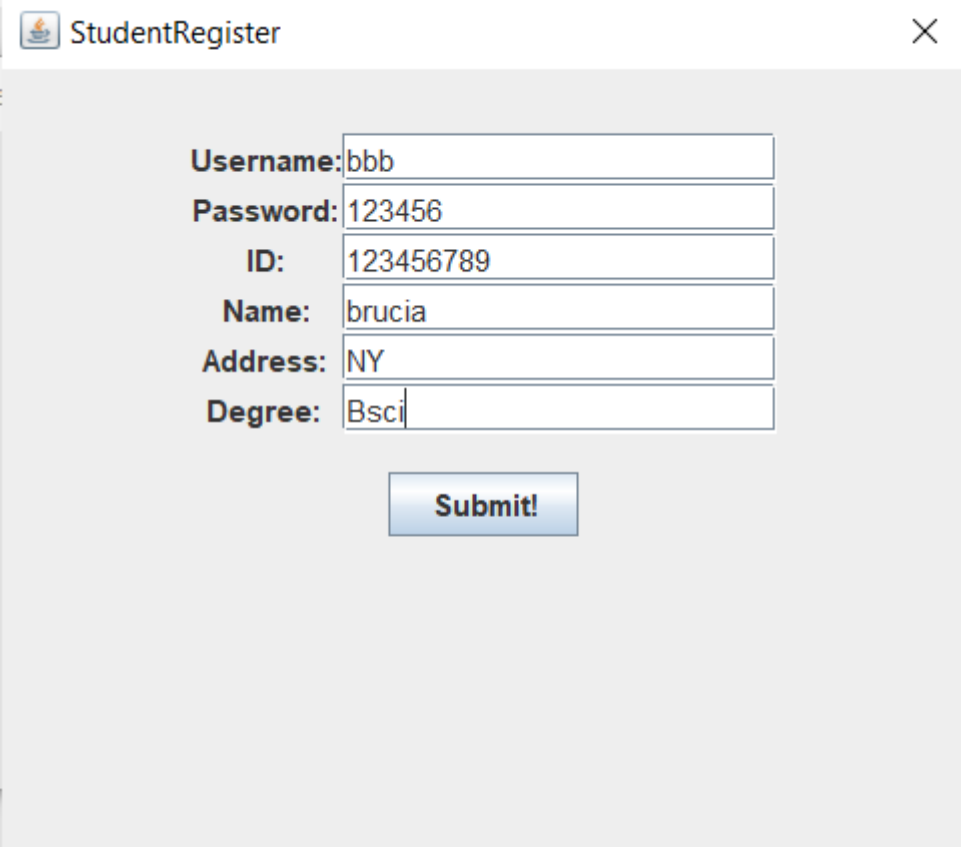
Step 1: After running the application, the entry page UI is shown, click on the register button



Step 2: After click on the register button, then click on the student button



Step 3: Input the new student information(with duplicate ID)

A window titled "StudentRegister" with a close button (X) in the top right corner. It contains a form with the following fields: Username: bbb, Password: 123456, ID: 123456789, Name: brucia, Address: NY, and Degree: Bsci. A "Submit!" button is located below the form fields.

StudentRegister

Username: bbb

Password: 123456

ID: 123456789

Name: brucia

Address: NY


Degree: Bsci

Submit!

Step 4: Since the ID 123456789 already exist in the database:

id	name	account	password	address	degree
bruce	bruce	123456	ny	123456789	Bsci
223456789	NULL	123456789	bbbbaaswj	North York	BS
218041103	matthew	145214321	mattpat1	Toronto	software engineering
218041106	joe	171456998	joeman	Toronto	software engineering
123456789	bruce	bruce	123456	ny	Bsci
123456	bruce	bruce1	123456	ny	sci
NULL	NULL	NULL	NULL	NULL	NULL

Step 5: It shows fail page

A window titled "error" with a close button (X) in the top right corner. The main content area is gray and displays the word "fail" in the center.

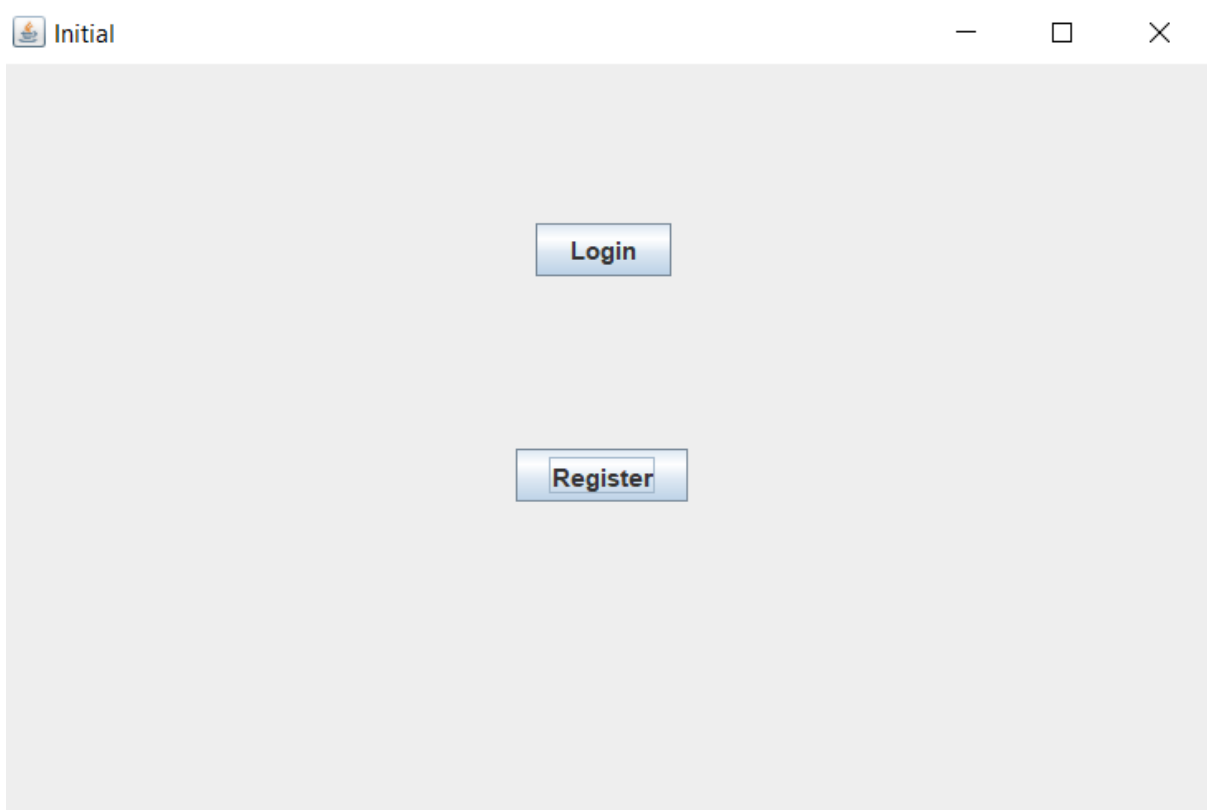
error

fail

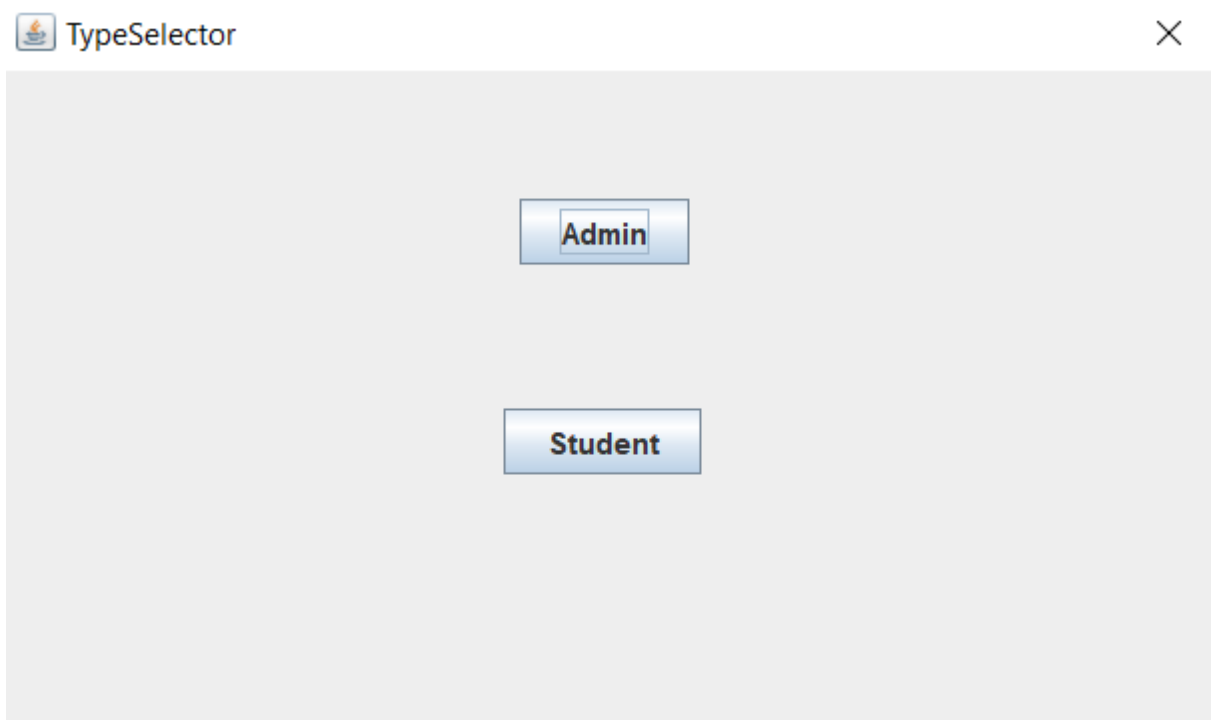
Step 6: Once you click on the X button, the program will stop running

Case 3: inputting a duplicate account and password

Step 1: After running the application, the entry page UI is shown, click on the register button



Step 2: After click on the register button, then click on the student button



Step 3: Input the new student information (with duplicate account and password)



StudentRegister



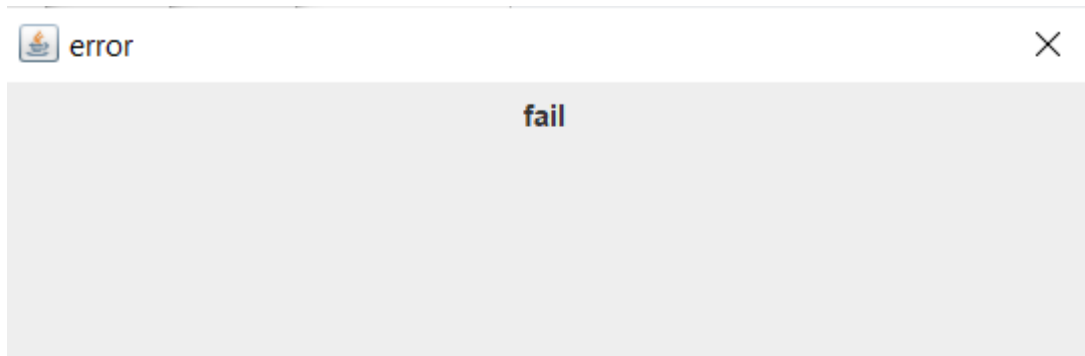
Username: bbb
Password: 123456
ID: 21345
Name: brucia
Address: NY
Degree: BArt

Submit!

Step 4: Since the username bbb and password 123456 already exist in the database:

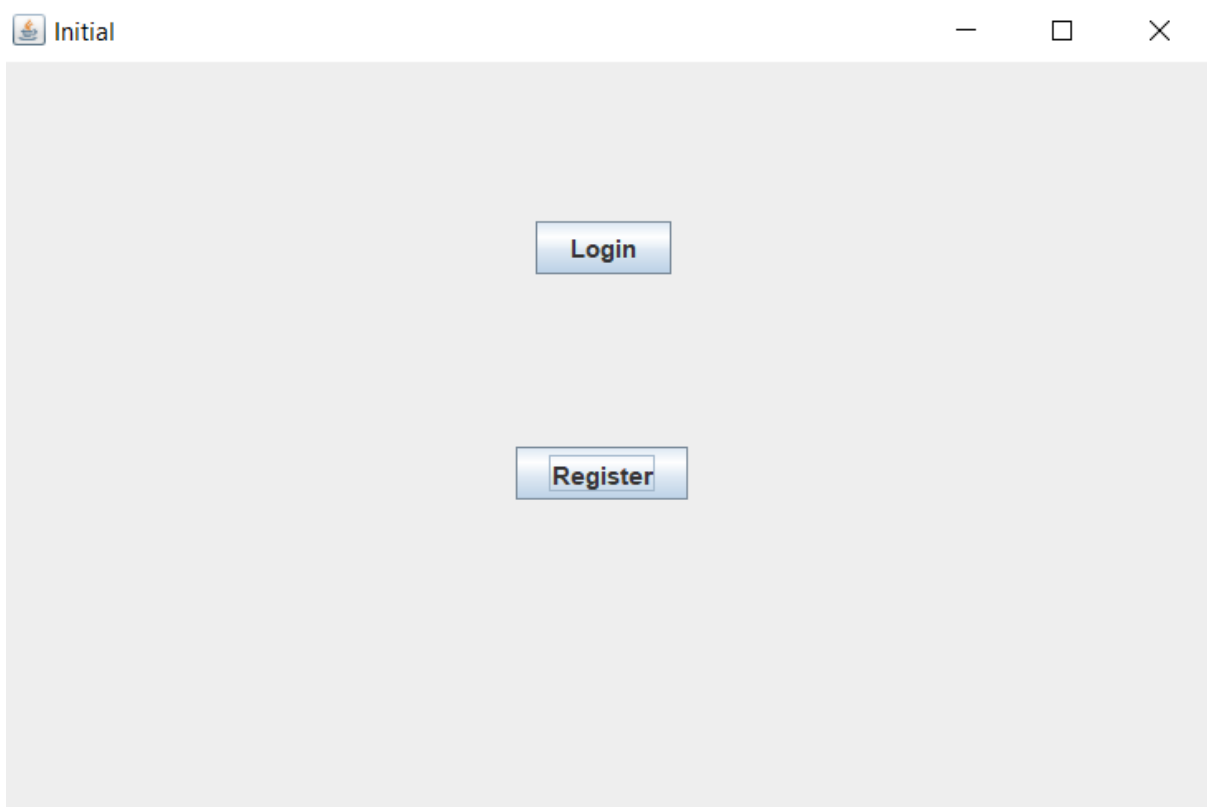
id	name	account	password	address	degree
bruce	bruce	123456	ny	123456789	Bsci
223456789	NULL	123456789	bbbbaaswj	North York	BS
218041103	matthew	145214321	mattpat1	Toronto	software engineering
218041106	joe	171456998	joeman	Toronto	software engineering
123	brucia	bbb	123	NY	BEng
213456789	brucia	bbb	123456	NY	BArt
213	brucia	bbb1	123	NY	BArt
123456789	bruce	bruce	123456	ny	Bsci
123456	bruce	bruce1	123456	ny	sci
NULL	NULL	NULL	NULL	NULL	NULL

Step 5: It shows fail page

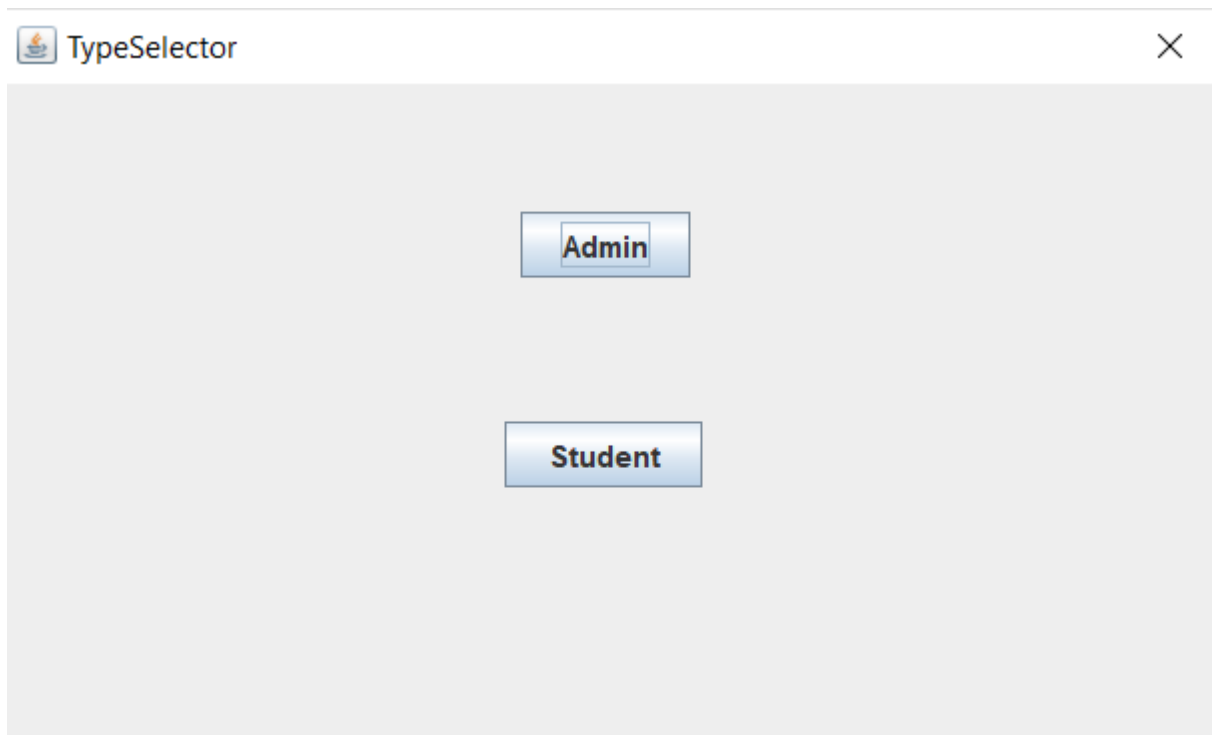


Case 4: inputting a duplicate account but different password

Step 1: After running the application, the entry page UI is shown, click on the register button



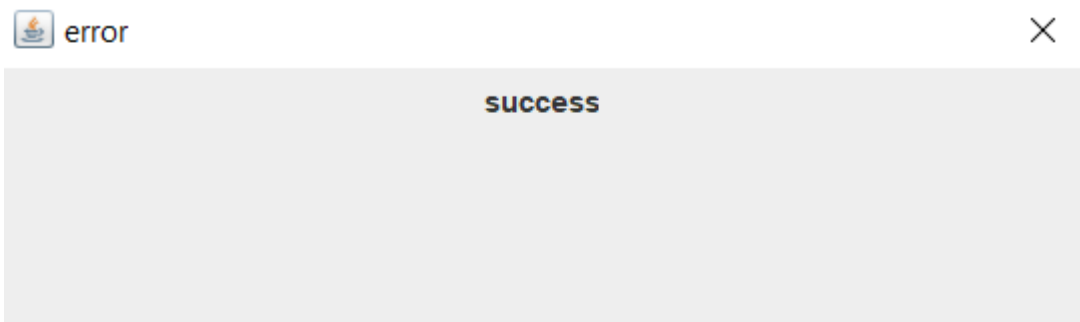
Step 2: After click on the register button, then click on the student button



Step 3: Input the new student information (with duplicate account but different password)

A screenshot of a window titled "StudentRegister" with a close button (X) in the top right corner. The window has a light gray background. It contains several input fields for student information, each with a label to its left. The fields are: "Username:" with the value "bbb", "Password:" with the value "123", "ID:" with the value "123", "Name:" with the value "brucia", "Address:" with the value "NY", and "Degree:" with the value "BEng". Below these fields is a blue button with white text labeled "Submit!".

Step 4: It shows success page

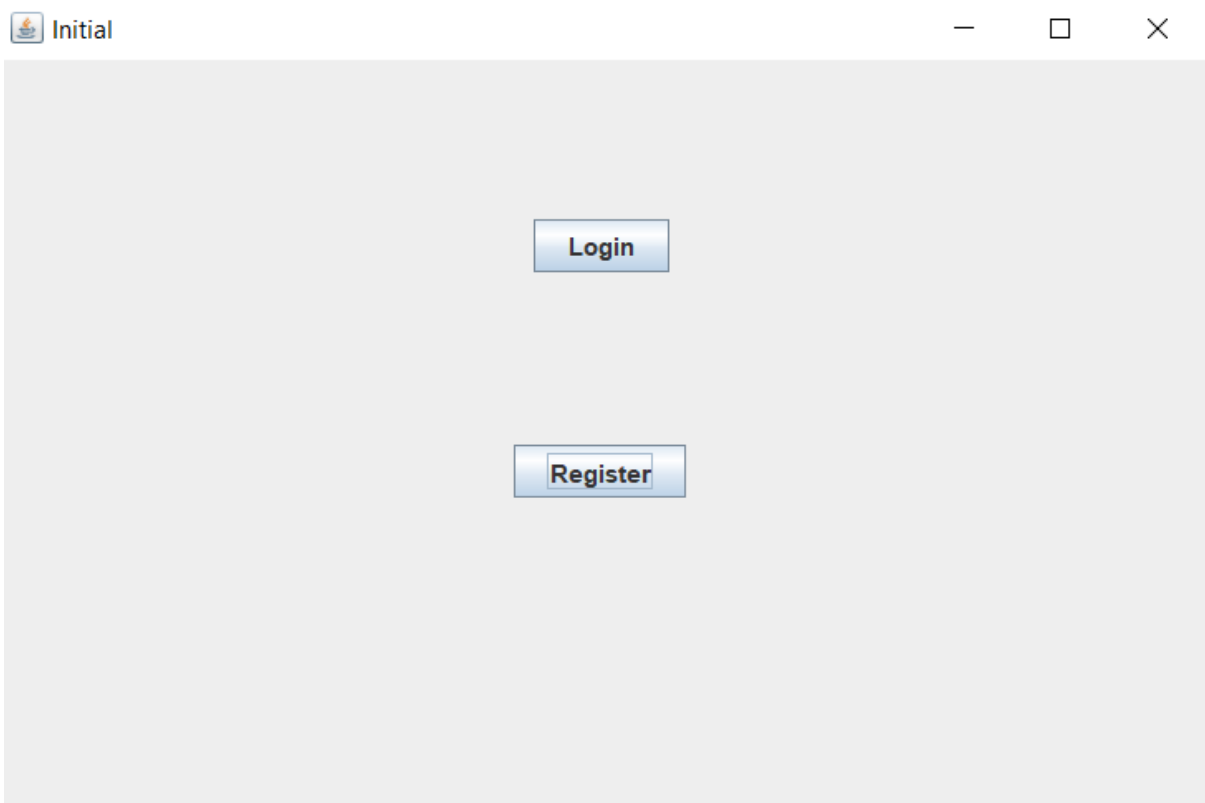


Step 5: The record is added to the database

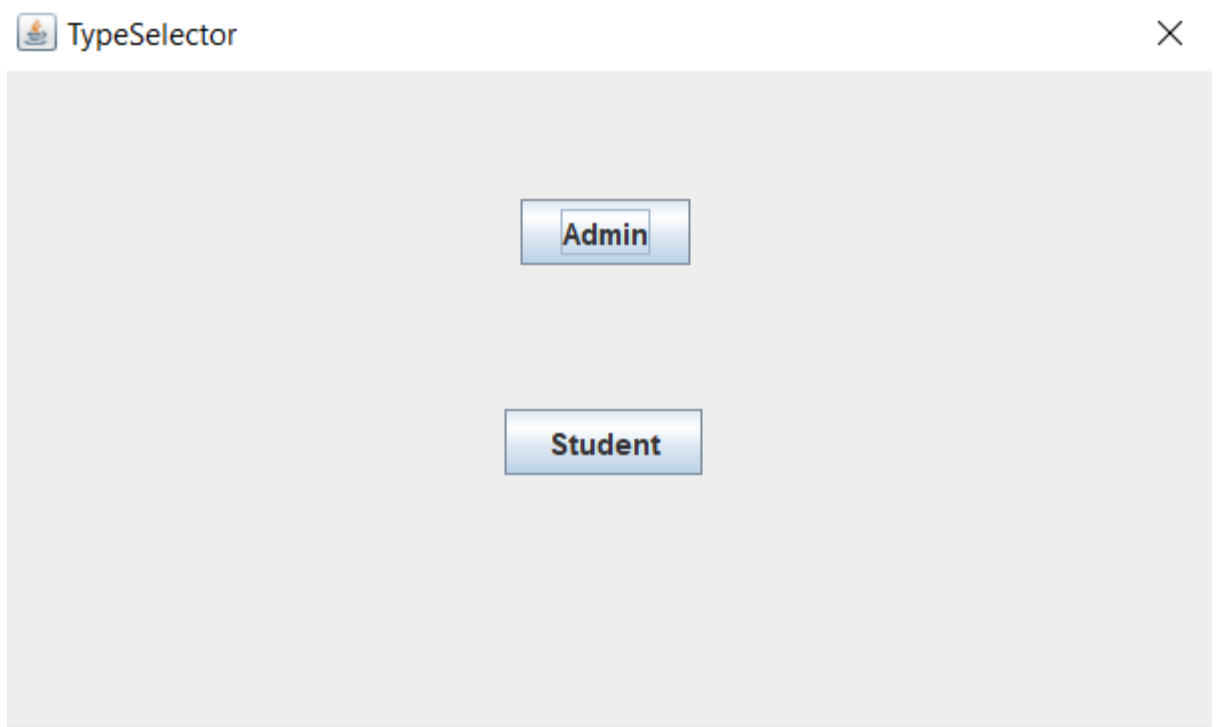
123	brucia	bbb	123	NY	BEng
-----	--------	-----	-----	----	------

Case 5: inputting a duplicate password but different account

Step 1: After running the application, the entry page UI is shown, click on the register button

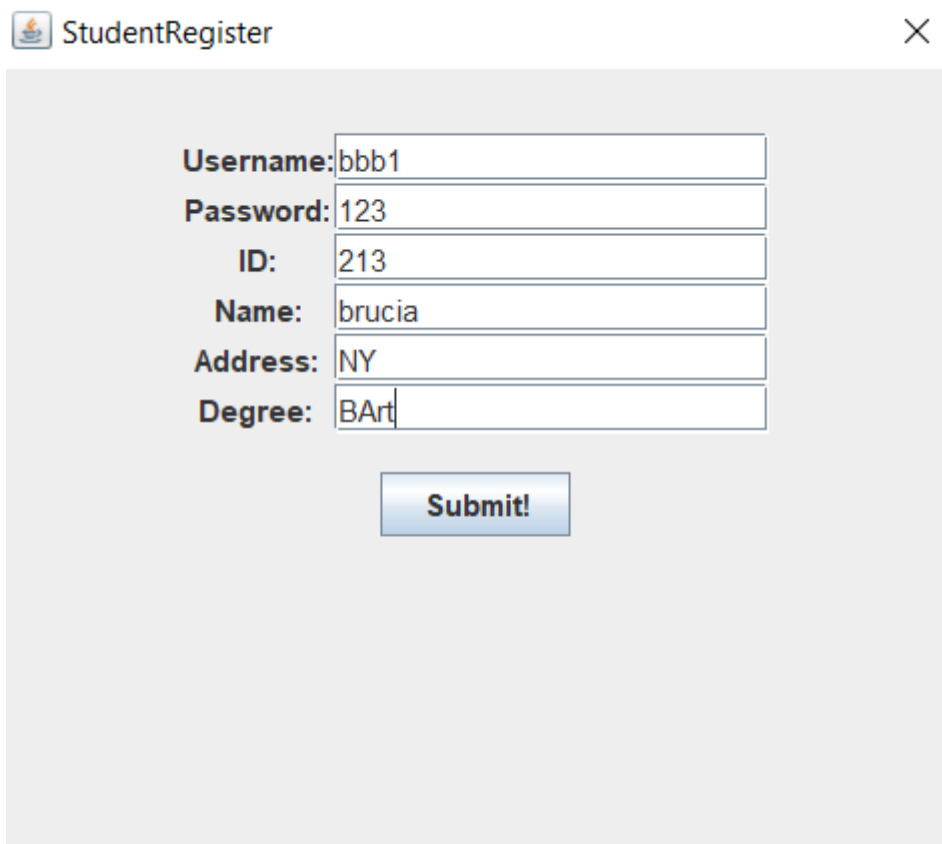


Step 2: After click on the register button, then click on the student button



A window titled "TypeSelector" with a close button (X) in the top right corner. The window has a light gray background. In the center, there are two blue buttons with white text. The top button is labeled "Admin" and the bottom button is labeled "Student".

Step 3: Input the new student information (with different account and same password)



A window titled "StudentRegister" with a close button (X) in the top right corner. The window has a light gray background. It contains a form with the following fields and values:

Username:	bbb1
Password:	123
ID:	213
Name:	brucia
Address:	NY
Degree:	BArt

Below the form is a blue button with white text labeled "Submit!".

Step 4: It shows success page



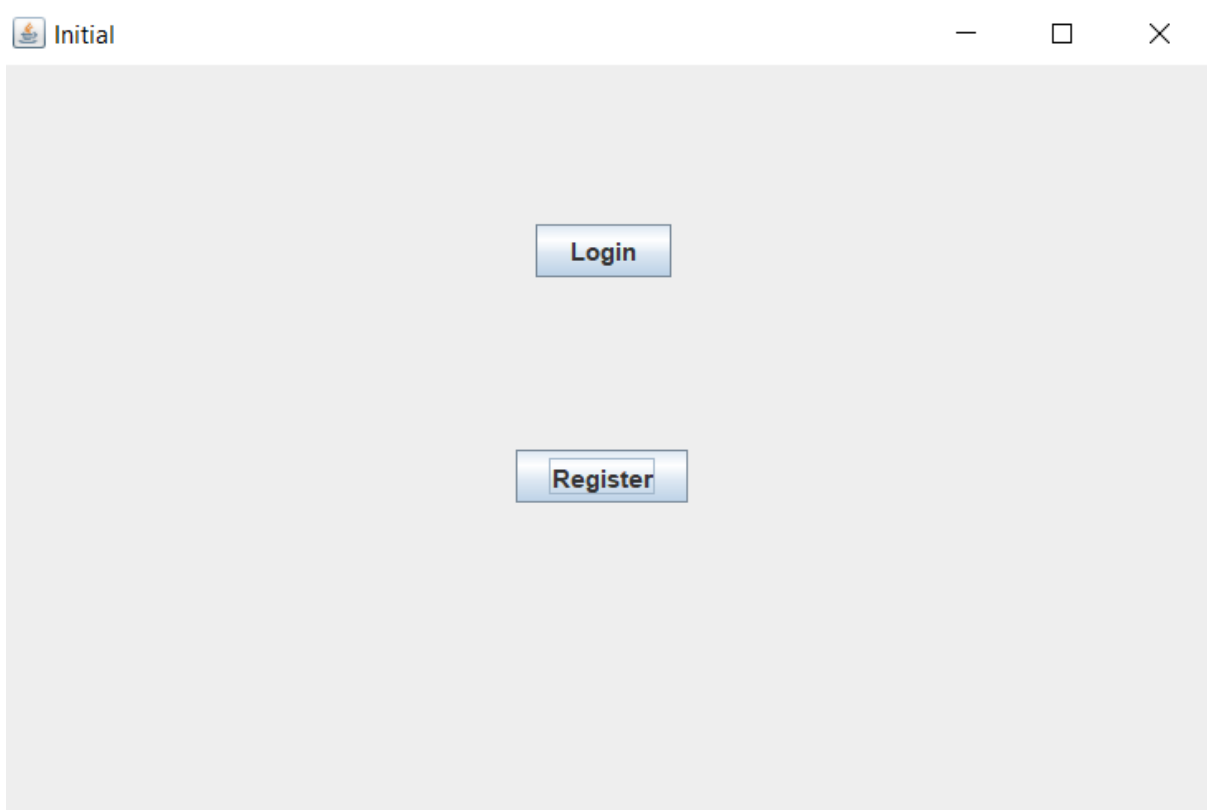
Step 5: The record is added to the database

213	brucia	bbb1	123	NY	BArt
-----	--------	------	-----	----	------

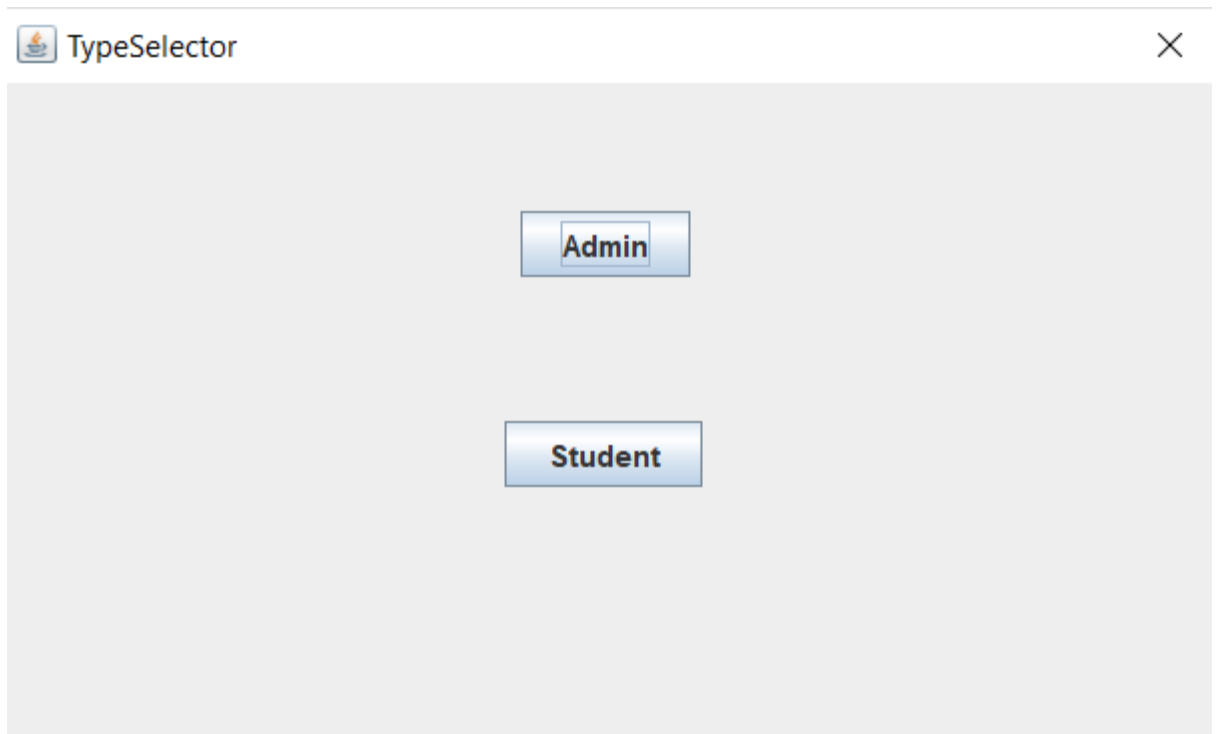
End-2-End test on Student Login User Interface:

Case 1: normal scenario

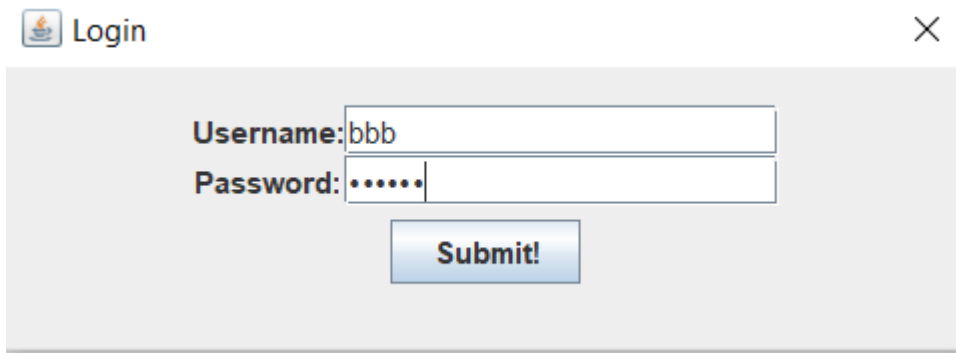
Step 1: After running the application, the entry page UI is shown, click on the login button



Step 2: After click on the register button, then click on the student button

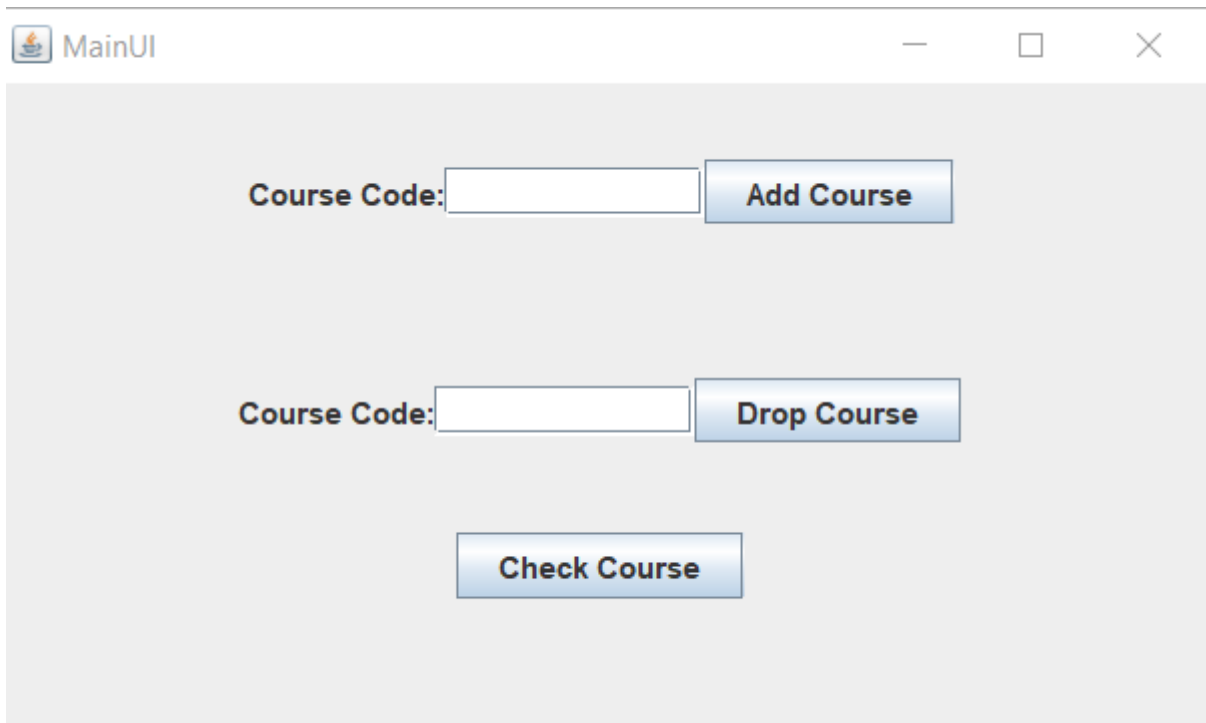


Step 3: Input the student login information (with correct account and correct password)



A screenshot of a 'Login' window. The window has a title bar with a small icon and the text 'Login', and a close button (X) in the top right corner. The main area is light gray and contains two text input fields. The first field is labeled 'Username:' and contains the text 'bbb'. The second field is labeled 'Password:' and contains six dots. Below the password field is a blue button with the text 'Submit!'.

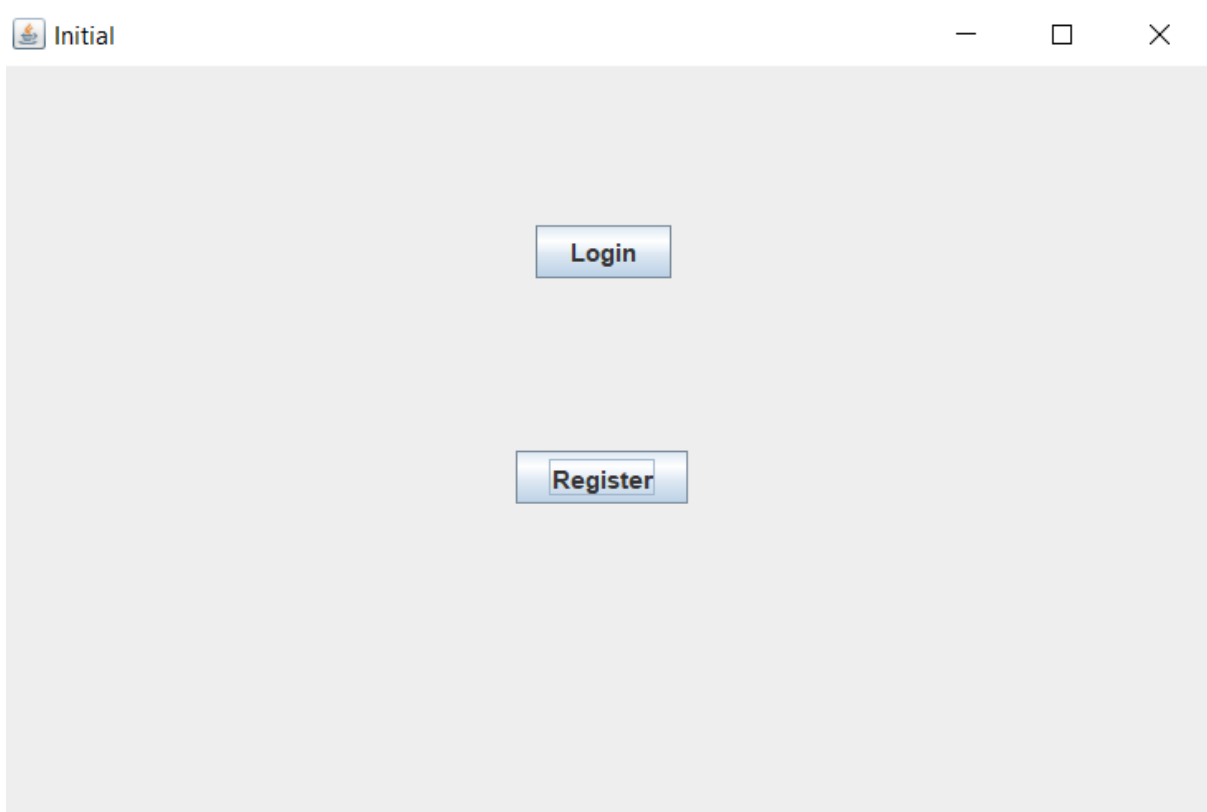
Step 4: main user interface will be shown



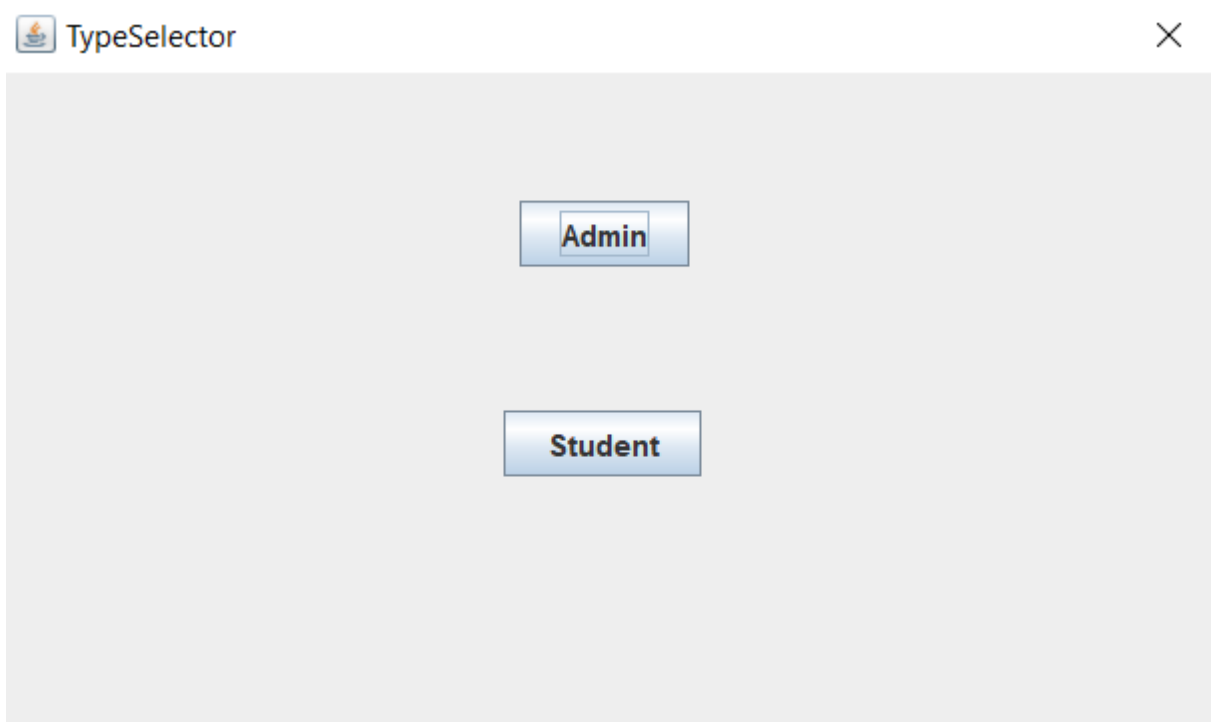
A screenshot of a 'MainUI' window. The window has a title bar with a small icon and the text 'MainUI', and standard window controls (minimize, maximize, close) in the top right corner. The main area is light gray and contains three sets of controls. The first set consists of a text input field labeled 'Course Code:' followed by a blue button labeled 'Add Course'. The second set consists of a text input field labeled 'Course Code:' followed by a blue button labeled 'Drop Course'. The third set is a single blue button labeled 'Check Course' centered below the other two.

Case 2: account that does not exist

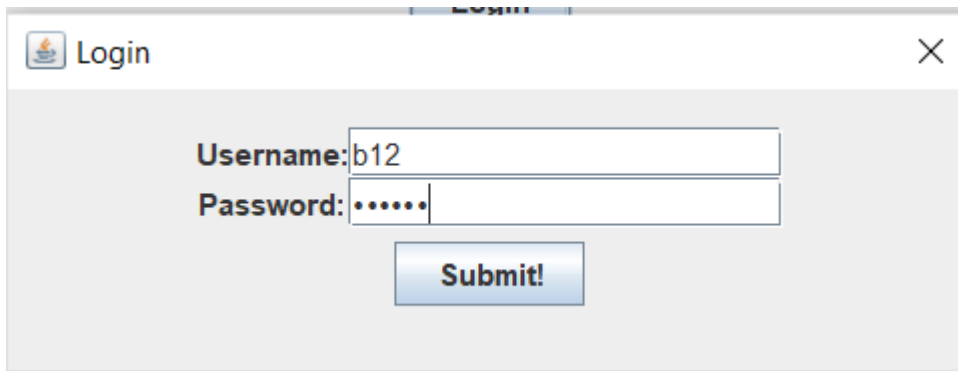
Step 1: After running the application, the entry page UI is shown, click on the login button



Step 2: After click on the register button, then click on the student button



Step 3: Input the student login information (with account does not exist but correct password)

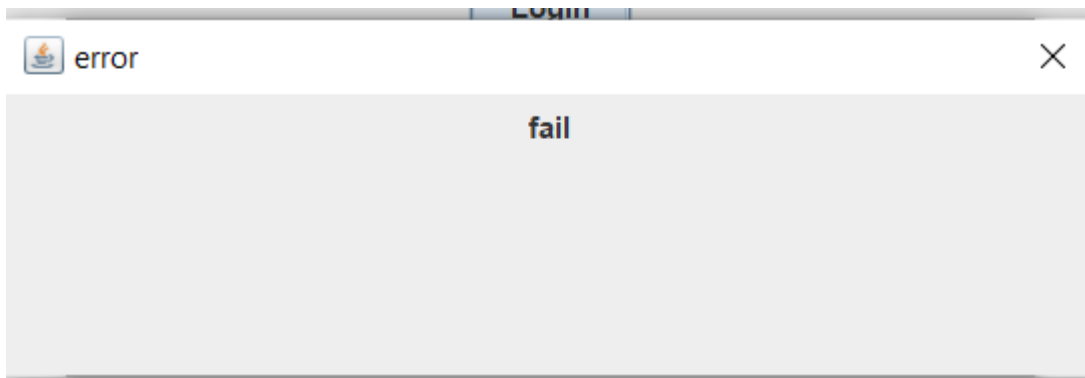
A login dialog box titled "Login" with a close button (X) in the top right corner. It contains two input fields: "Username:" with the text "b12" and "Password:" with masked characters ".....". Below the fields is a blue button labeled "Submit!".

Username: b12

Password:

Submit!

Step 4: a error page will be shown

An error dialog box titled "error" with a close button (X) in the top right corner. The main content area displays the word "fail" in a bold, black font.

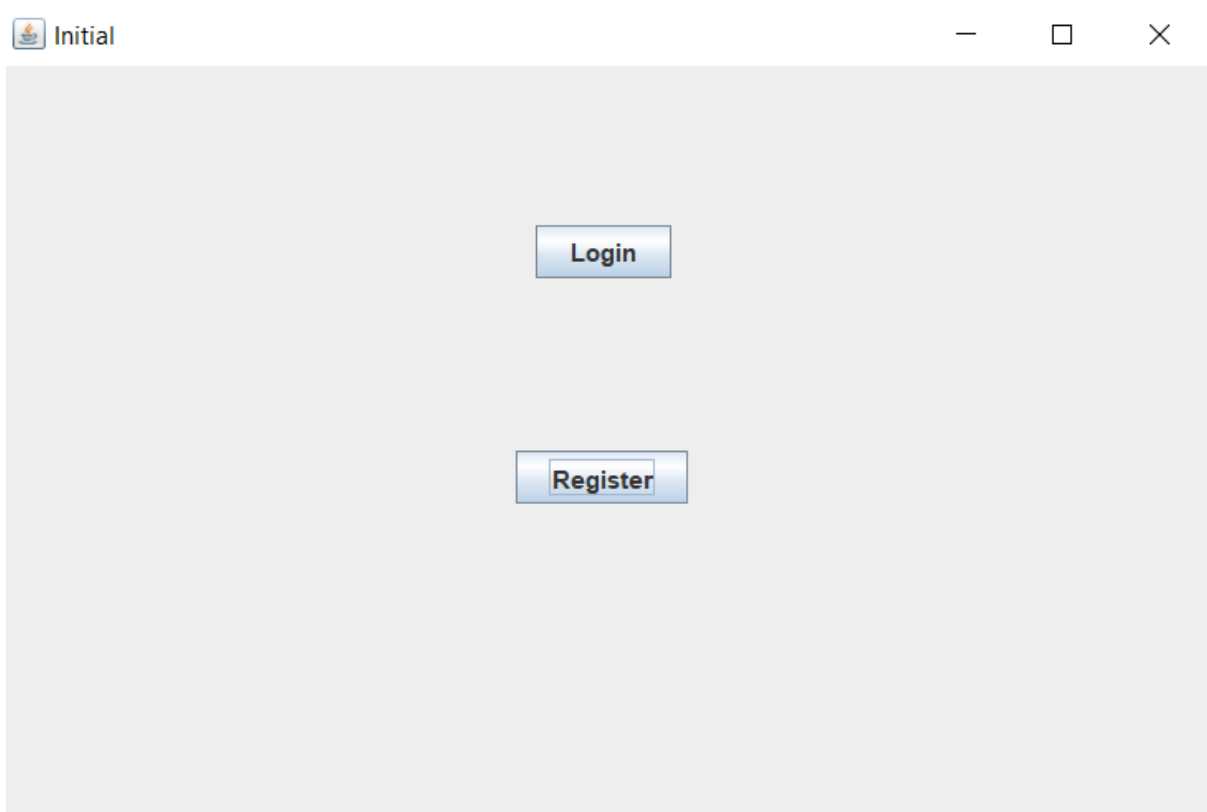
fail

Step 5: since the database does not have the login information, so the login failed

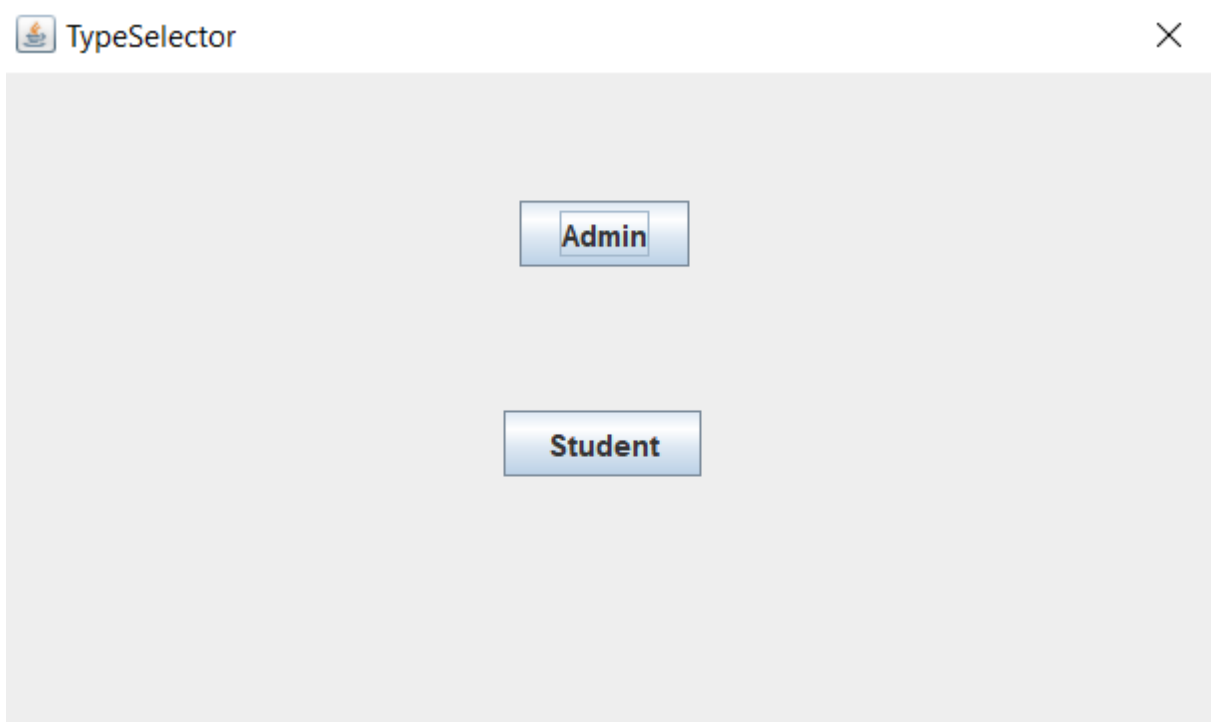
	id	name	account	password	address	degree
▶	bruce	bruce	123456	ny	123456789	Bsci
	223456789	NULL	123456789	bbbbaaswj	North York	BS
	218041103	matthew	145214321	mattpat1	Toronto	software engineering
	218041106	joe	171456998	joeman	Toronto	software engineering
	123	brucia	bbb	123	NY	BEng
	213456789	brucia	bbb	123456	NY	BArt
	213	brucia	bbb1	123	NY	BArt
	123456789	bruce	bruce	123456	ny	Bsci
	123456	bruce	bruce1	123456	ny	sci
*	NULL	NULL	NULL	NULL	NULL	NULL

Case 3: correct account but wrong password (234)

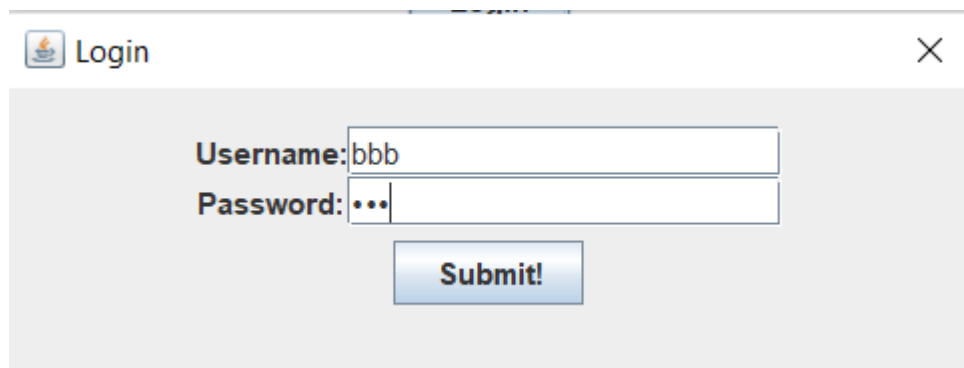
Step 1: After running the application, the entry page UI is shown, click on the login button



Step 2: After click on the register button, then click on the student button



Step 3: Input the student login information (with account does not exist but correct password



A login dialog box titled "Login" with a close button (X) in the top right corner. It contains two input fields: "Username:" with the text "bbb" and "Password:" with three dots indicating a masked password. Below the fields is a blue "Submit!" button.

Step 4: a error page will be shown



An error dialog box titled "error" with a close button (X) in the top right corner. The main content area is gray and displays the word "fail" in bold black text.

Step 5: since the database does not have the login information, so the login failed

	id	name	account	password	address	degree
▶	bruce	bruce	123456	ny	123456789	Bsci
	223456789	NULL	123456789	bbbbaaswj	North York	BS
	218041103	matthew	145214321	mattpat1	Toronto	software engineering
	218041106	joe	171456998	joeman	Toronto	software engineering
	123	brucia	bbb	123	NY	BEng
	213456789	brucia	bbb	123456	NY	BArt
	213	brucia	bbb1	123	NY	BArt
	123456789	bruce	bruce	123456	ny	Bsci
	123456	bruce	bruce1	123456	ny	sci
•	NULL	NULL	NULL	NULL	NULL	NULL

2. Code review:

1.code smells

Dispensables - No Comments

For the student register and student login code part, there is no comments in the code, make the code hard to understand.

(1) Snapshot of student register code part

```
JLabel Degree = new JLabel("Degree:");
GridBagConstraints c12 = new GridBagConstraints();
GridBagConstraintsSetter(c12,0,6,0);

JLabel fill1 = new JLabel(" ");
GridBagConstraints cf1 = new GridBagConstraints();
GridBagConstraintsSetter(cf1,0,0,0);

JButton submit = new JButton("Submit!");

submit.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e){
        Student s = new Student(NameBox.getText(),DegreeBox.getText(),AddressBox.getText());
        if(!s.getStatus()) {
            Error error = new Error(frame,"fail");
            error.setVisible(true);
            System.exit(0);
        }
        Error error = new Error(frame,"success");
        error.setVisible(true);
        sr.setVisible(false);
        // System.out.println(UsernameBox.getText()+" "+PasswordBox.getText());
    }
});
```

(2) Snapshot of student login code part

```
submit.addActionListener(new ActionListener(){
    @Override
    public void actionPerformed(ActionEvent e){
        if(UsernameBox.getText().length()==9)
        {
            if(db.login(UsernameBox.getText(),new String>PasswordBox.getPassword())
                frame.setVisible(false);
                lg.setVisible(false);
                MainUI.getInstance(UsernameBox.getText());
            }else {
                Error error = new Error(frame,"fail");
                error.setVisible(true);
                System.exit(0);
            }
        }else if(UsernameBox.getText().length()==10) {
            if(db.adminLogin(UsernameBox.getText(),new String>PasswordBox.getPassw
                frame.setVisible(false);
                lg.setVisible(false);
                MainUIAdmin.getInstance();
            }else {
                Error error = new Error(frame,"fail");
                error.setVisible(true);
                System.exit(0);
            }
        }
    }
}
```