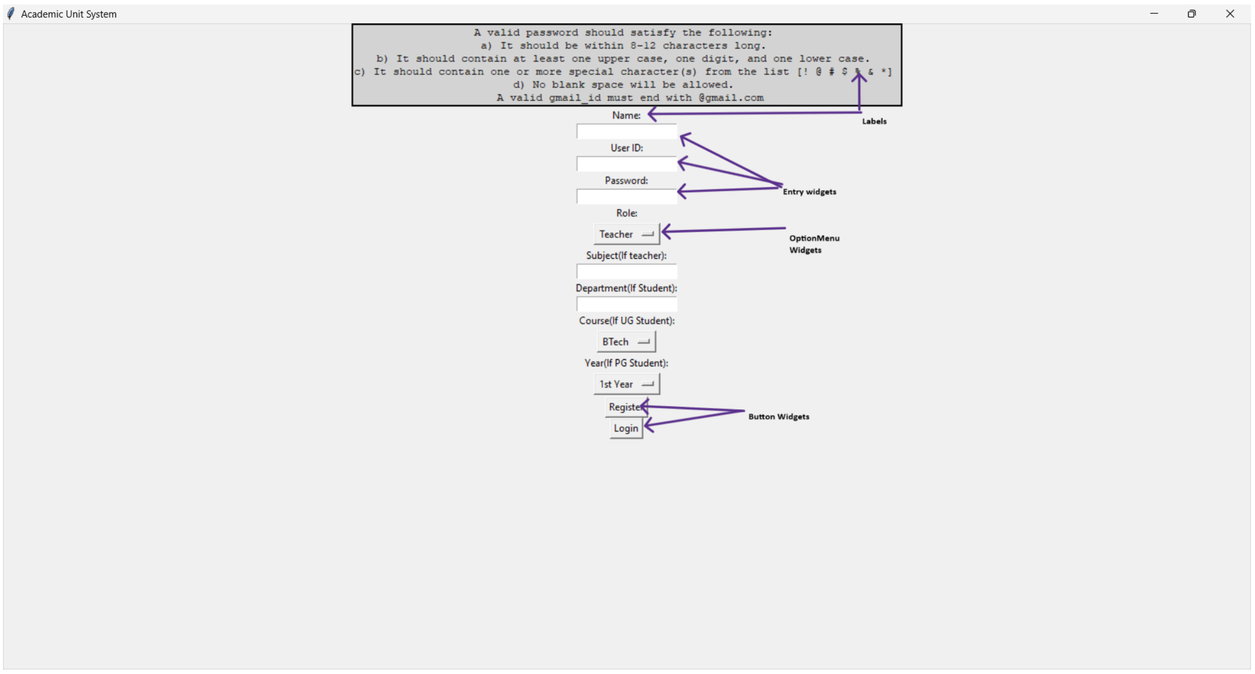
Name : Ankita Mishra

Roll Number : 22CS10010

Introduction:

The **"Academic Unit System"** is a Python-based application designed to manage user registration, authentication, and user profiles within an academic institution. The system employs object-oriented programming principles to create a flexible and scalable architecture. The user interface is implemented using the Tkinter library, providing a simple and intuitive interaction for users.



```
A valid password should satisfy the following:
a) It should be within 8-12 characters long.
b) It should contain at least one upper case, one digit, and one lower case.
c) It should contain one or more special character(s) from the list [! @ # $ % & *]
d) No blank space will be allowed.
A valid gmail_id must end with @gmail.com
```

Main Window

GUI : Graphical User Interface(GUI) is a form of user interface which allows users to interact with computers through visual indicators using items such as icons, menus, windows, etc. It has advantages over the Command Line Interface(CLI) where users interact with computers by writing commands using keyboard only and whose usage is more difficult than GUI.

Tkinter :
The tkinter package **("Tk interface")** is the standard Python interface to the Tcl/Tk GUI toolkit. It is widely used to create GUI applications.

Some definitions :

Window :
This term has different meanings in different contexts, but in general it refers to a rectangular area somewhere on the user's display screen.

Top-level window :
A window which acts as a child of the primary window. It will be decorated with the standard frame and controls for the desktop manager. It can be moved around the desktop and can usually be resized.

Widget :
The generic term for any of the building blocks that make up an application in a graphical user interface.

Core widgets:
The containers: frame, labelframe, toplevel, paned window. The buttons: button, radiobutton, checkbutton (checkbox), and menubutton.
The text widgets: label, message, text.
The entry widgets: scale, scrollbar, listbox, slider, spinbox, entry (singleline), optionmenu, text (multiline), and canvas (vector and pixel graphics).

Tkinter provides three modules that allow pop-up dialogs to be displayed:
tk.messagebox (confirmation, information, warning and error dialogs),
tk.filedialog (single file, multiple file and directory selection dialogs)
and tk.colorchooser (colour picker).

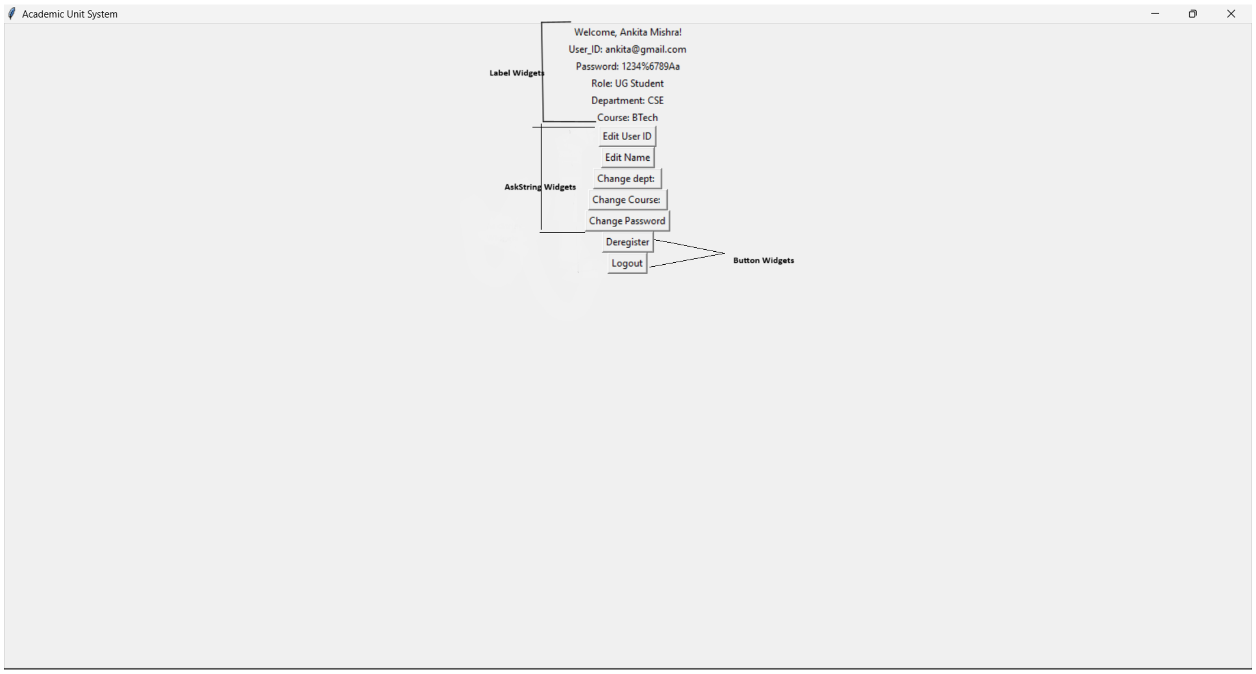
Frame :
In Tkinter, the Frame widget is the basic unit of organisation for complex layouts. A frame is a rectangular area that can contain other widgets.

Child and parent :
When any widget is created, a parent–child relationship is created. For example, if we place a text label inside a frame, the frame is the parent of the label.

Process :

There are four stages to creating a widget.

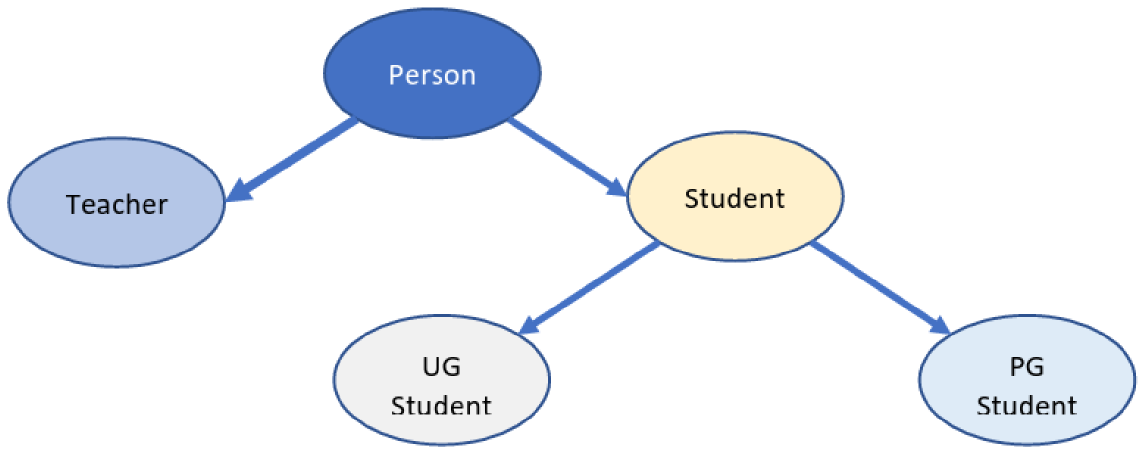
Create :
Create it within a frame.
Configure :
Change the widgets attributes.
Pack :
Packing it into position so it becomes visible. Developers also have the option to use .grid() (row=int, column=int to define rows and columns to position the widget, defaults to 0) and .place() (relx=int or decimal, rely=int or decimal, define coordinates in the frame, or window).
Bind :
Binding it to a function or event.



Welcome, Ankita Mishra!
User_ID: ankita@gmail.com
Password: 1234%6789Aa
Role: UG Student
Department: CSE
Course: BTech

Login Window

Classes:



The system follows a hierarchical structure with a base class Person, which is then extended by subclasses like Teacher, Student, UGStudent, and PGStudent. This design allows for easy extension of functionalities and the incorporation of specific attributes for different roles within the academic setting.

Person :
It is the uppermost leaf of the hierarchy. Its features contain User Id, password, Name, Role and Count.

Teacher :
'Teacher' comes under Person class. Its features contain the features of Person class and subject as additional features.

Student :
'Student' comes under Person class. Its features contain the features of Person class and dept(department) as additional features.

UG Student :
'UG Student' comes under Student class. Its features contain the features of Student class and course (BTech/Dual Degree) as additional features.

PG Student :
'PG Student' comes under Student class. Its features contain the features of Student class and year (1st year/2nd year) as additional features.

AcademicUnitSystem :

The AcademicUnitSystem class manages user data, including loading and saving user profiles using pickle. It also provides essential functionalities like user registration, authentication, and user profile updates. Password validation and user ID uniqueness are enforced during registration.

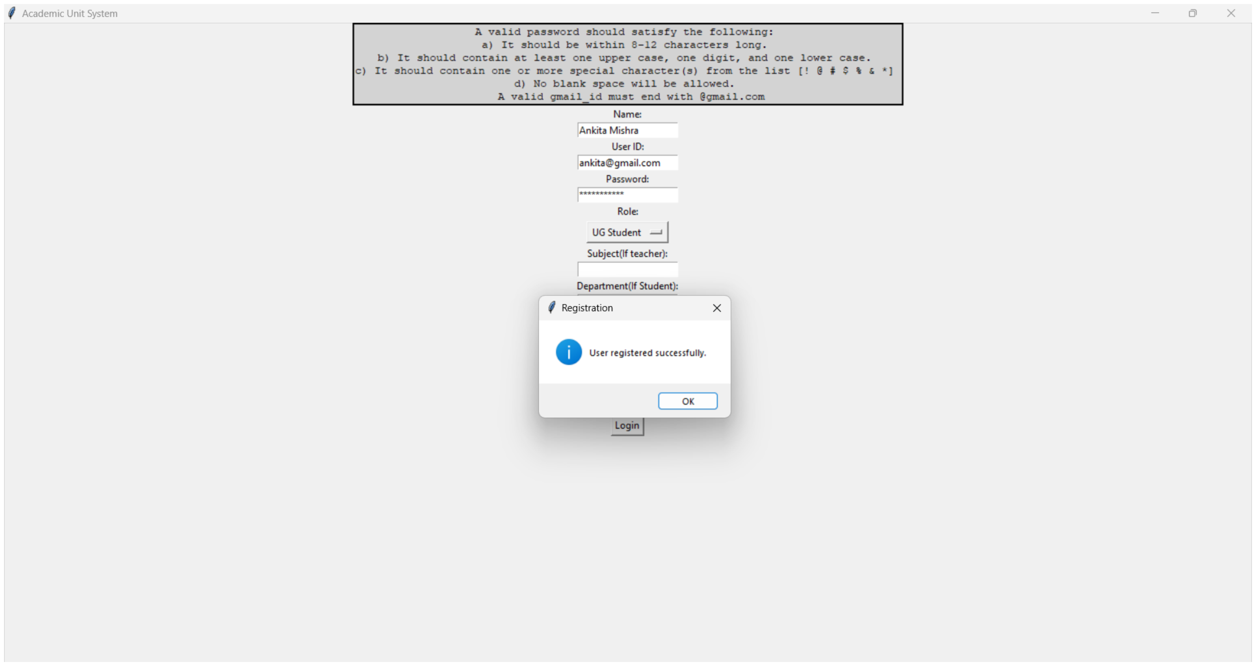
__init__(self) : Creates a list of persons currently registered and active.

load_users : Checks if filename users.pkl is present, if present, the empty list created in function init is filled with the contents of the file. If not, the list is kept empty.

save_users : Saves users after registration/Deregistration in the file users.pkl . It basically rewrites the entire content of the current list, self.users .

is_valid_password : Checks password validation logic, if passed returns True, else returns False.

register_user : If the user id has already been registered before and is active, informs the user about the same. Also checks the validity of the user id and password. If everything passes, it registers the user, otherwise cancels the process. User, if registered, is appended in the list and also in the file.



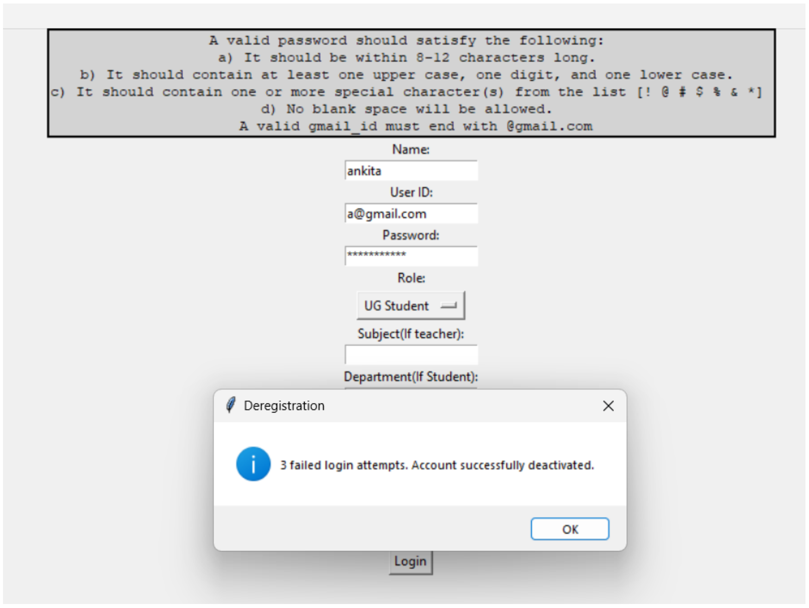
```
A valid password should satisfy the following:
a) It should be within 8-12 characters long.
b) It should contain at least one upper case, one digit, and one lower case.
c) It should contain one or more special character(s) from the list [! @ # $ % & *]
d) No blank space will be allowed.
A valid gmail_id must end with @gmail.com
```

Name: Ankita Mishra
User ID: ankita@gmail.com
Password: ***********
Role: UG Student
Subject(If teacher):
Department(If Student):

Registration

User registered successfully.

authenticate_user : Checks if the email id/user id entered matches with any user id in the list. If not, it informs the user about the same. If found, it then checks whether the password entered matches with the saved ones for that user, if not, maximum of 3 chances are given. If 3 failed attempts, the account is deactivated.



```
A valid password should satisfy the following:
a) It should be within 8-12 characters long.
b) It should contain at least one upper case, one digit, and one lower case.
c) It should contain one or more special character(s) from the list [! @ # $ % & *]
d) No blank space will be allowed.
A valid gmail_id must end with @gmail.com
```

Name: ankita

User ID: a@gmail.com

Password: ***********

Role: UG Student

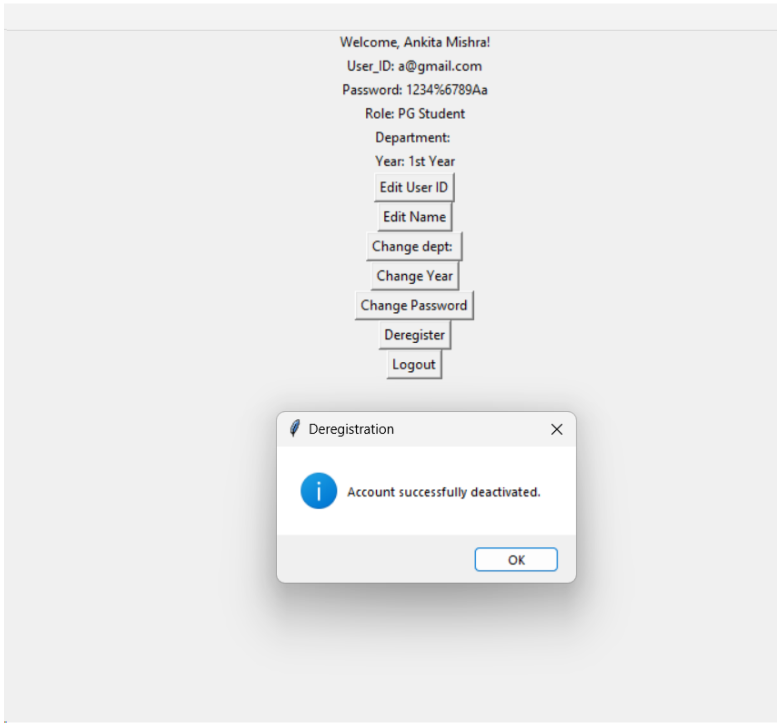
Subject(If teacher):

Department(If Student):

Deregistration

3 failed login attempts. Account successfully deactivated.

deregister_user : User is asked the password of their account.If password matches with the saved ones, account is deregistered, else the process is cancelled.



Welcome, Ankita Mishra!

User_ID: a@gmail.com

Password: 1234%6789Aa

Role: PG Student

Department:

Year: 1st Year

Deregistration

Account successfully deactivated.

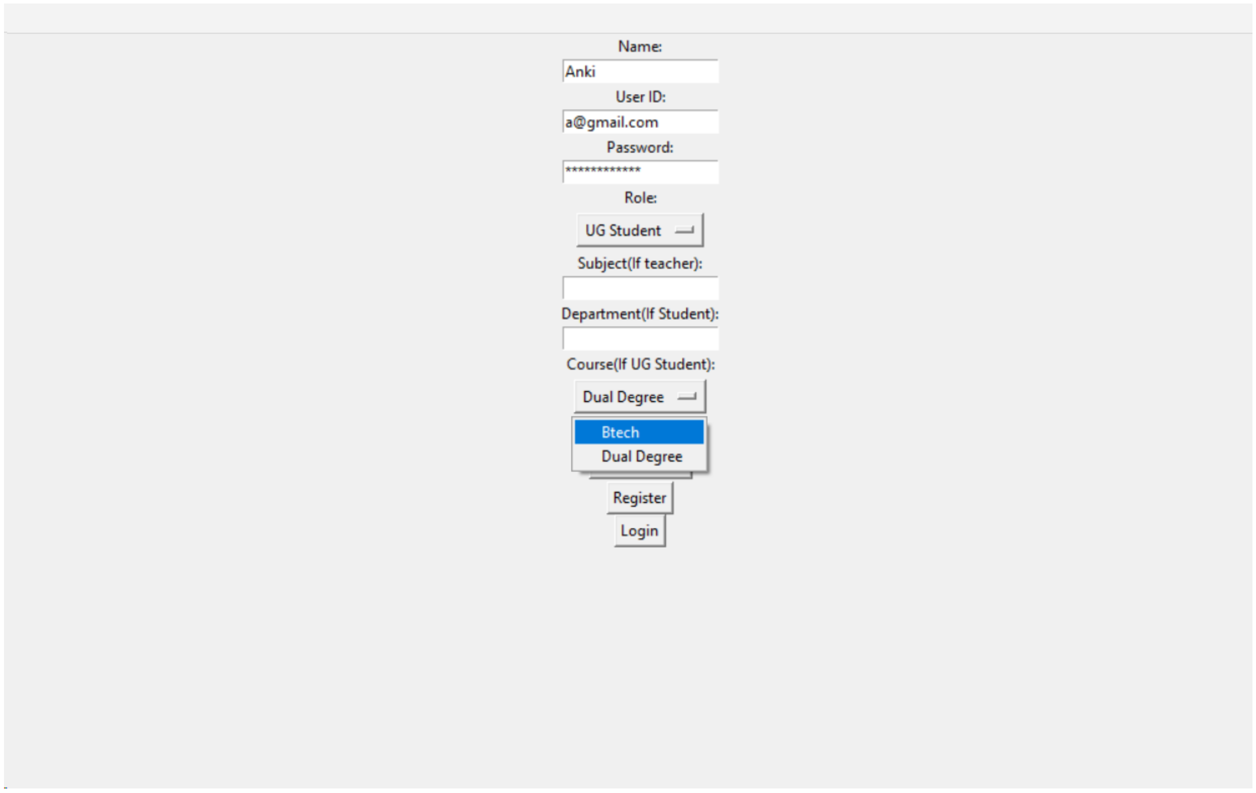
is_user_id_taken : Checks if any user with the entered user id exists in the list.

UserInterface :

The UserInterface class is responsible for creating a Tkinter-based graphical user interface, enabling users to register, login, and perform various actions related to their profiles. The interface dynamically adjusts based on the user's role, providing a seamless experience.

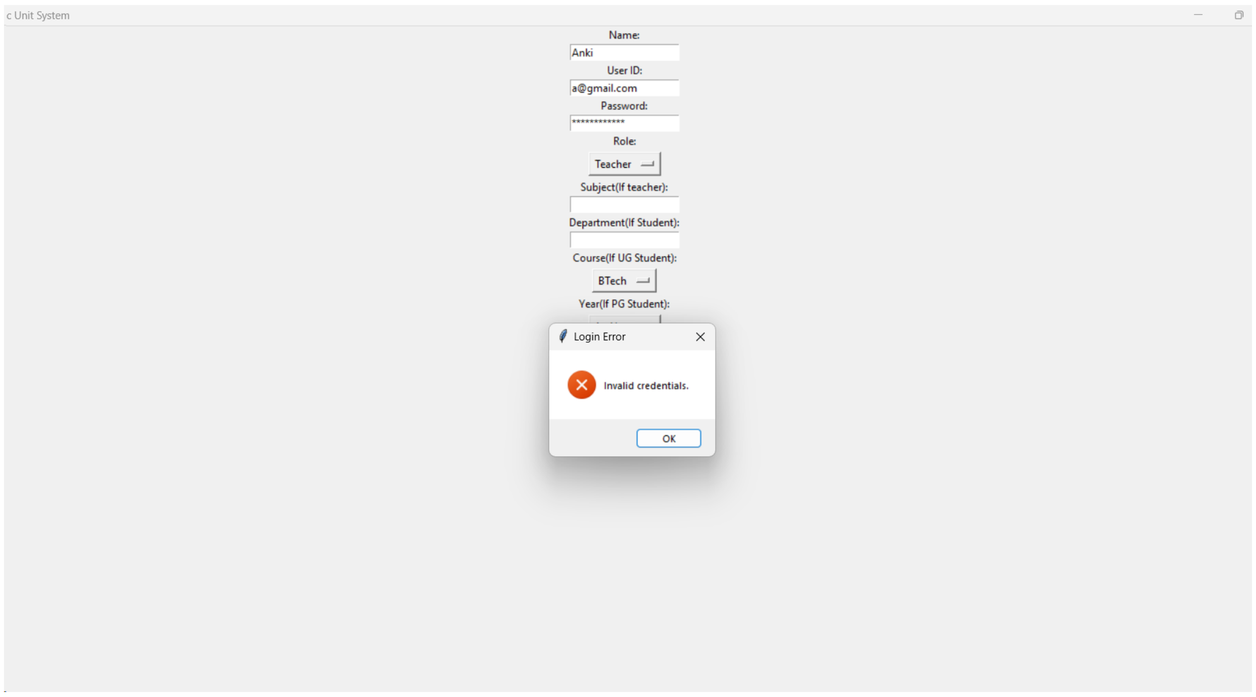
__init__ : Creates a window named 'root' text Label that has a text message informing the user the rules for user id and password. Also it has a function 'create_widgets' which basically adds more widgets to the window.

create_widgets : Uses mainly Label, Entry, StringVar, OptionMenu and Button as widgets.
1> **Label** : For directing the user the type of info to write in each entry box.
2> **Entry** : For taking input from the user.
3> **StringVar** : Used to set a feature to some initial value.(Like Teacher in role and BTech in dept as initial, can be changed by the user during the registration process.)
4> **OptionMenu** : Used to give a finite number of options as input and the user has to choose among them.
5> **Button** : Used to add buttons which directly connect with some function as called inside them.


Name:
Anki
User ID:
a@gmail.com
Password:
****
Role:
UG Student
Subject(If teacher):
Department(If Student):
Course(If UG Student):
Dual Degree

register_user : Takes all the inputs from create widget makes cases according to role of the user.
According to role, it passes relevant inputs through '**academic_unit_system.register_user**' function (defined in AcdemicUnitSystem class). If registration is successful,it informs the user about the same, otherwise it opens an error message window telling the user the process failed.

login : Takes user_id and password as input and passes through '**academic_unit_system.authenticate_user**' function (defined in AcdemicUnitSystem class). If the login is successful, it calls another function '**show_profile_options**' to show options after login, otherwise it gives the error message informing the user about the failure.


c Unit System

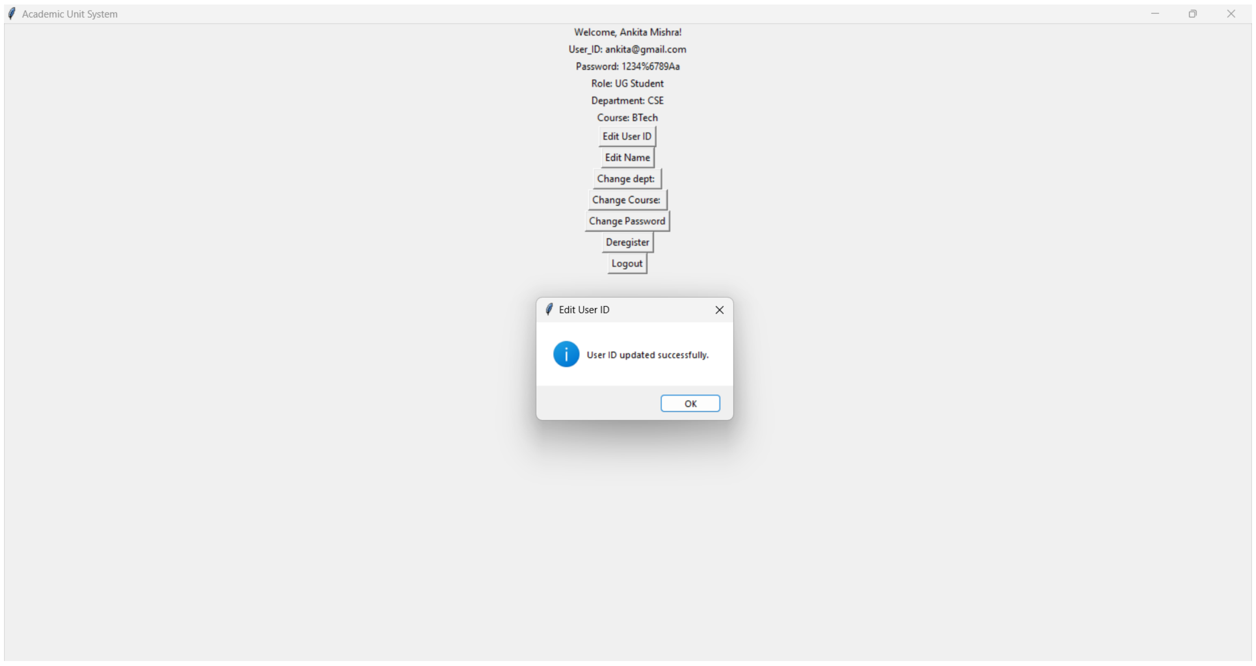
Name:
Anki
User ID:
a@gmail.com
Password:
****
Role:
Teacher
Subject(If teacher):
Department(If Student):
Course(If UG Student):
BTech
Year(If PG Student):

Login Error
Invalid credentials.
OK

show_profile_options : It destroys previous widgets and creates new widgets in the same window.

New widgets are as follows:
1> Labels : Shows Name, User ID, Password as text.
According to roles, gives related info about their roles in text.
2>Buttons : Has separate Buttons to change Name, user id, password, features specifically related to roles like subject(Teacher), Dept(Student), Course(UG Student) and Year(PG Student). Also gives buttons for deregistration and log out.
Function called for each Button as explained below.

Edit_user_id : Asks new user_id, check if the user_id is a valid email address.Also checks if that user_id is not the current user_id of any other user(user is are unique).If so, change the user id, otherwise cancel the process.


Academic Unit System

Welcome, Ankita Mishra!

User_ID: ankita@gmail.com

Password: 1234%6789Aa

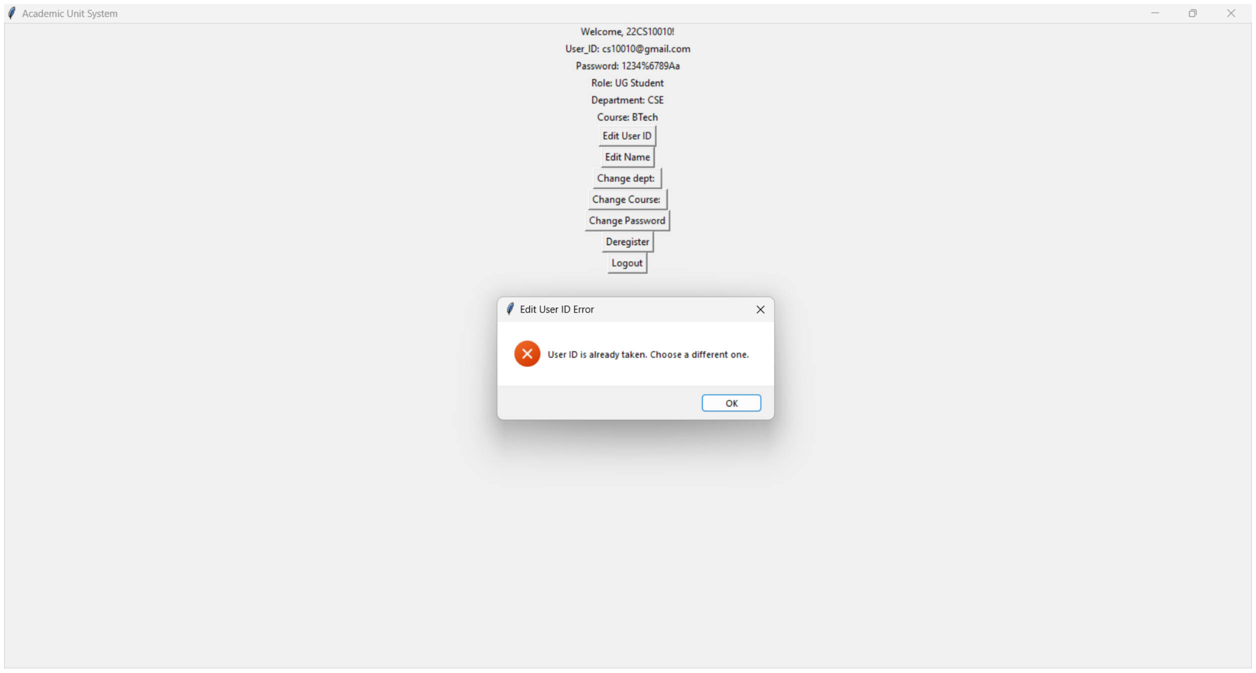
Role: UG Student

Department: CSE

Course: BTech

Edit User ID

User ID updated successfully.


Academic Unit System

Welcome, 22CS10010!

User_ID: cs10010@gmail.com

Password: 1234%6789Aa

Role: UG Student

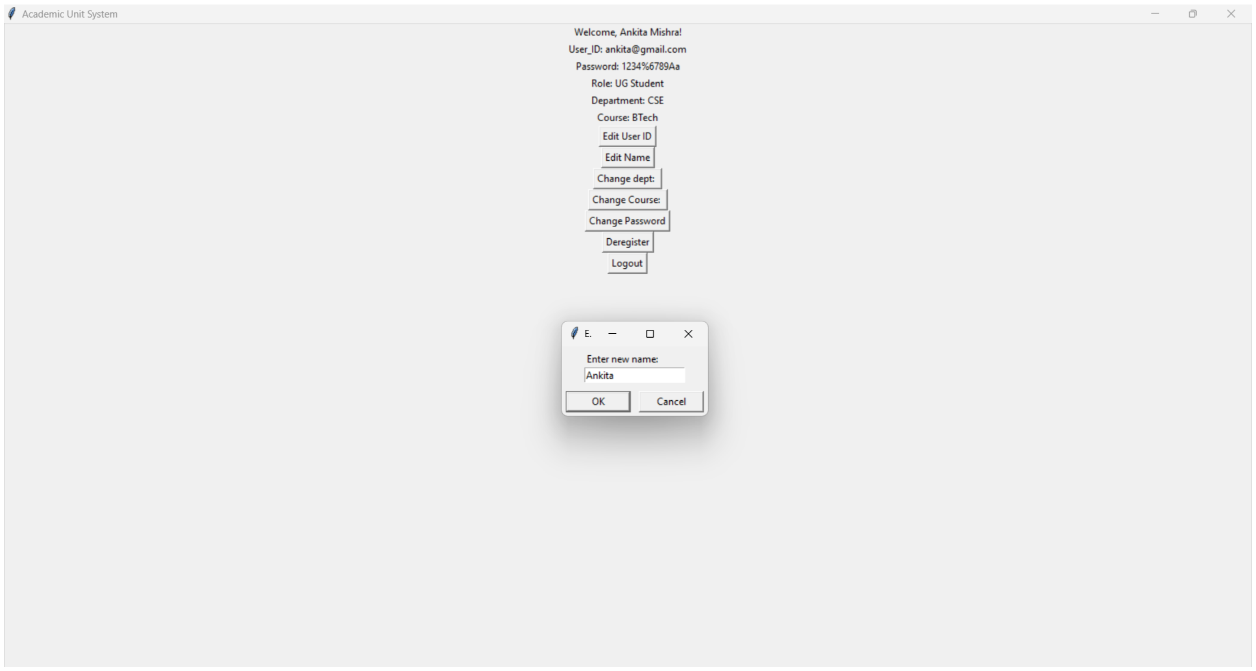
Department: CSE

Course: BTech

Edit User ID Error

User ID is already taken. Choose a different one.

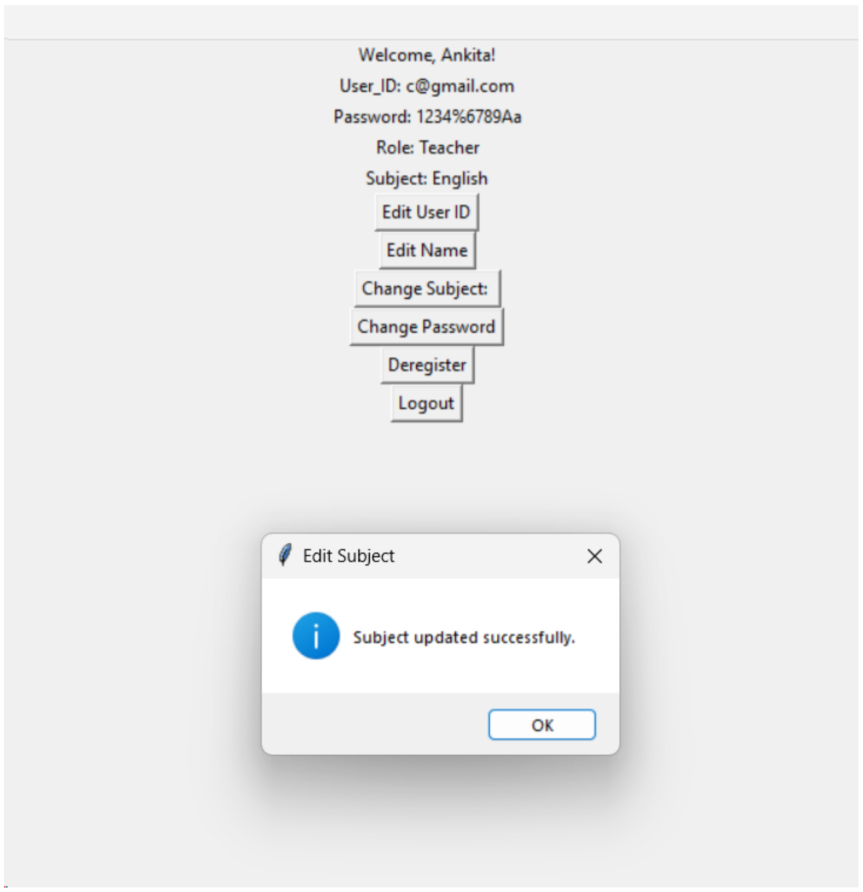
Edit_name : Asks new name, change to current name.



Welcome, Ankita Mishra!
User_ID: ankita@gmail.com
Password: 1234%6789Aa
Role: UG Student
Department: CSE
Course: BTech

Enter new name: Ankita

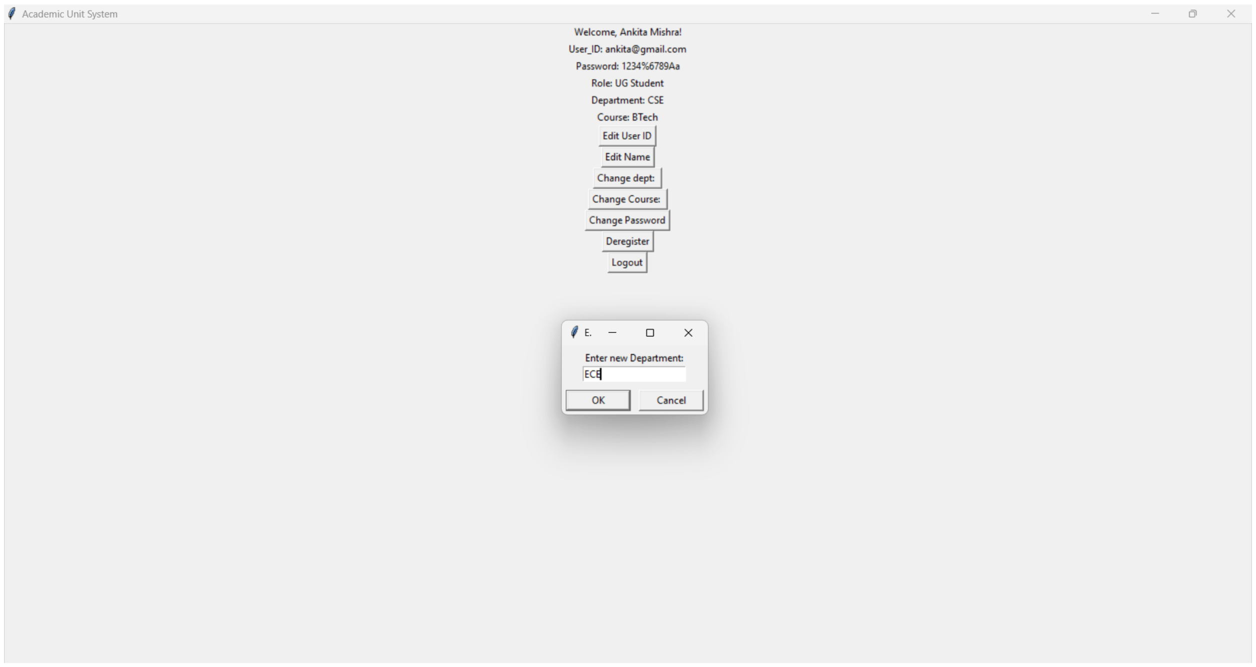
Edit_subject : Asks current subject, change to current subject.



Welcome, Ankita!
User_ID: c@gmail.com
Password: 1234%6789Aa
Role: Teacher
Subject: English

Subject updated successfully.

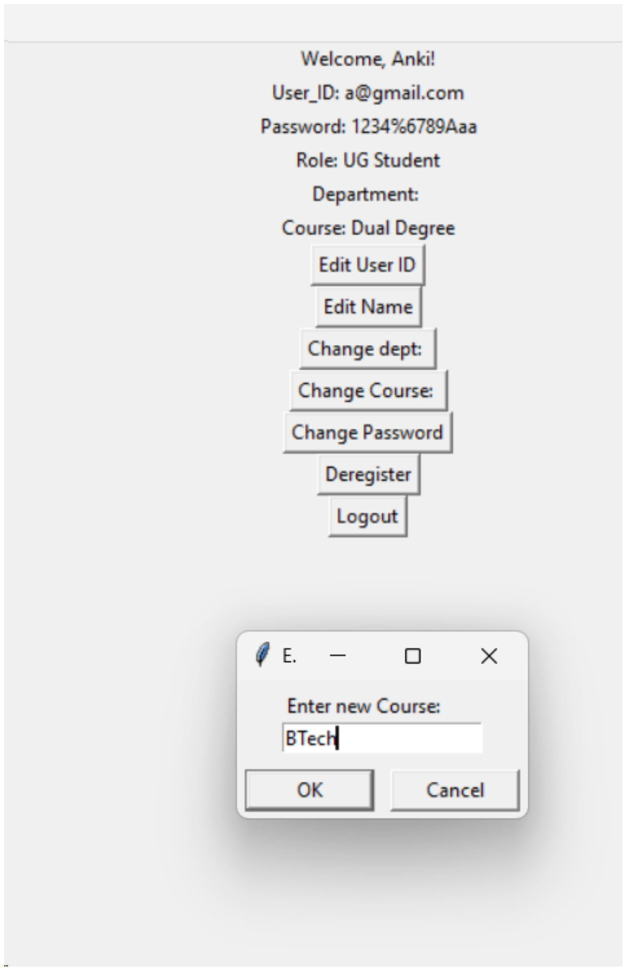
Edit_dept : Asks current department, change to current department.



Welcome, Ankita Mishra!
User_ID: ankita@gmail.com
Password: 1234%6789Aa
Role: UG Student
Department: CSE
Course: BTech

Enter new Department:
ECE

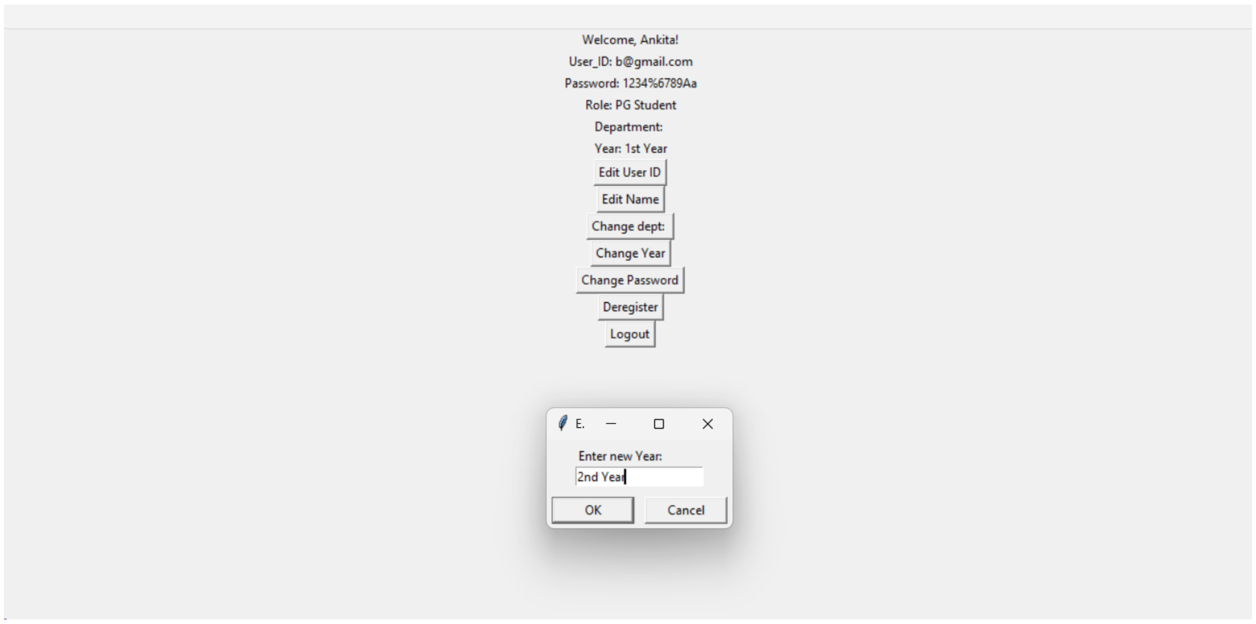
Edit_course : Asks current course, change to current course



Welcome, Anki!
User_ID: a@gmail.com
Password: 1234%6789Aaa
Role: UG Student
Department:
Course: Dual Degree

Enter new Course:
BTech

Edit_year : Asks current year, change to current year.



Welcome, Ankita!

User_ID: b@gmail.com

Password: 1234%6789Aa

Role: PG Student

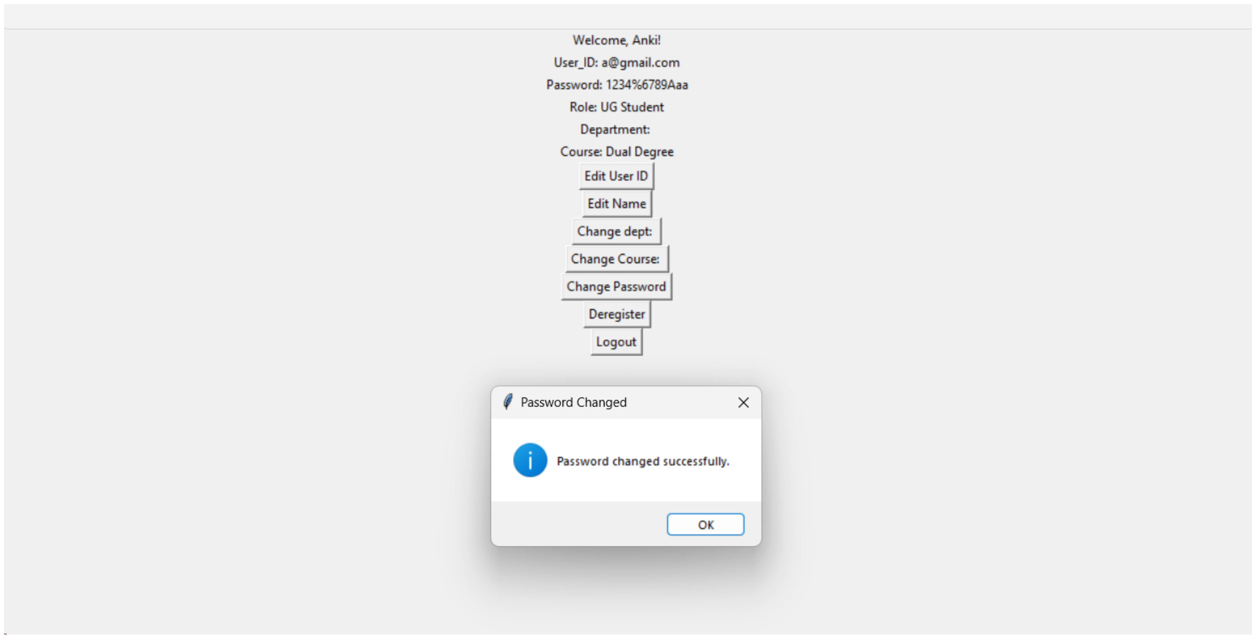
Department:

Year: 1st Year

Enter new Year:

2nd Year

Change_password : Asks new password from user. If the new password satisfies the criteria of a password, then change the password otherwise cancel the process.



Welcome, Anki!

User_ID: a@gmail.com

Password: 1234%6789Aaa

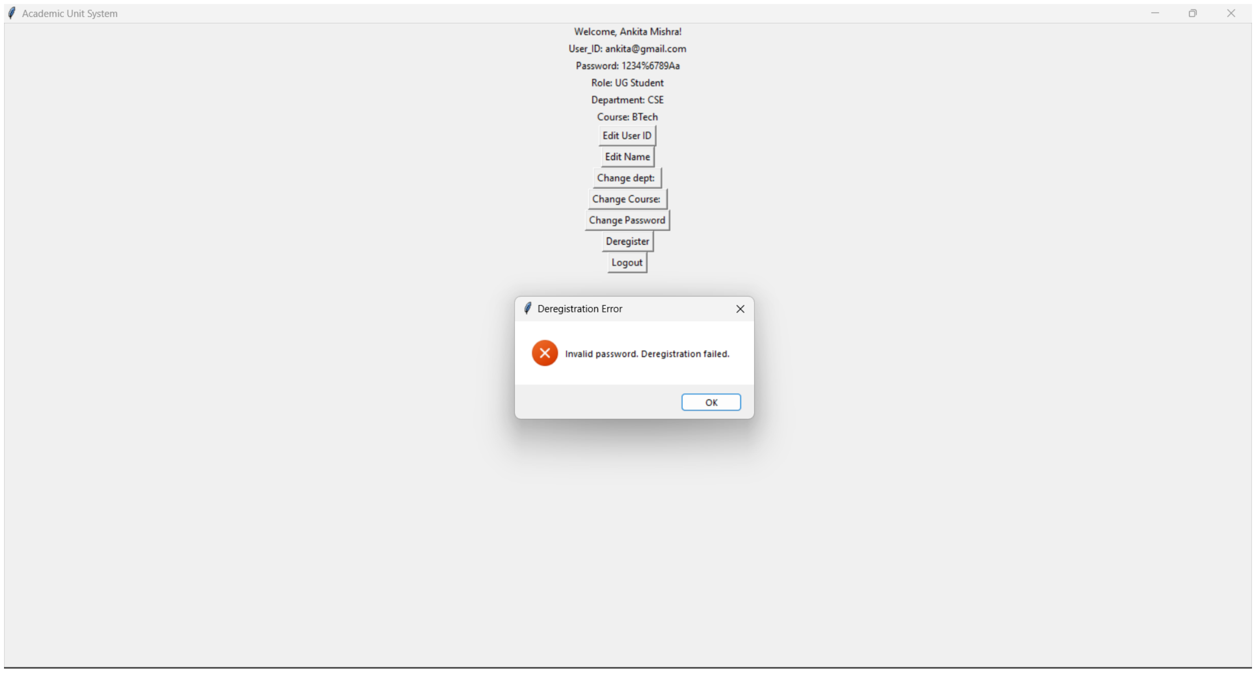
Role: UG Student

Department:

Course: Dual Degree

Password changed successfully.

Deregister_user : Asks for password to deregister user, if provided correct, deregister, otherwise cancel the process.



Welcome, Ankita Mishra!
User_ID: ankita@gmail.com
Password: 1234%6789Aa
Role: UG Student
Department: CSE
Course: BTech

Invalid password. Deregistration failed.

logout : It destroys the current widgets and returns to the main widget window.

Conclusion:

The "**Academic Unit System**" demonstrates a well-structured and extensible design using object-oriented principles. It offers a user-friendly interface for managing academic profiles and ensures the security and integrity of user data. The system's modular design allows for easy maintenance and future enhancements. Overall, it serves as a robust solution for academic institutions seeking a versatile user management system.