



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Text to Modelling Diagram Project

Project Review-3

Version: 1

Team Member Details

Reg. No	Branch	Name
19BCE0654	CSE	Anurag Chowdhury
19BCE0672	CSE	Udit Gupta
19BCE0673	CSE	Veerabhadra Mohan Khandelot

Acknowledgement

We are very grateful to acknowledge to our Prof. Swathi J N and all who have helped us in every step to complete our project. To convert our ideas well above the level of simplicity, into something concrete and meaningful.

We all team members thank whole-heartedly to Prof. Swathi J N for constantly motivating us to do better and showed complete confidence in our work. In the process, we learnt a lot other technical and non-technical things from her.

Only through her constant guidance, valuable suggestions and encouragement we were able to achieve our goal.

Executive Summary

Our Project converts a given pseudocode, which is written in text format to highly comprehensible high end UML Diagrams and Flowcharts, which are in image format. Option is given to user whether he wants to create a new text file or upload from existing files.

Overview:

When text is typed into the website, this software constructs flowchart as described in text.

The Problem:

For making flowchart, one needs to construct using complex software.

The solution:

With this software tool, anyone can make flowchart without installing any software.

Highlights:

Still most of the students install software to construct their flowchart. With this software one, do not need to waste time installing the software and get a virus-free output.

Table of Contents

1. Introduction
 - 1.1 Objective
 - 1.2 Motivation
 - 1.3 Background
2. Project Description Goals
3. Technical Specifications
4. Design Approach
 - 4.1 Design Approach
 - 4.2 Codes and Standard
 - 4.3 Constraints, Alternatives & Tradeoffs
5. Schedule, Tasks & Milestones
6. Project Demonstration
7. Results & Discussion

List of Tables

Table of Definitions, Acronyms, and Abbreviations

Table of References

Table of Shall Requirements

Table of Design Constraints

Table of User Characteristics

Abbreviations, Symbols & Notations

Abbreviations, Symbols & Notations	Description
SRS	Software Requirements Specification
WBS	Work Breakdown System
.txt	Text File Extension
.JPG	Image File Extension
.PNG	Image File Extension
Admin	Administrator (same as Developer)

Table of Definitions, Acronyms, and Abbreviations

1. Introduction

1.1 Objective

Objective is to convert text into flowchart in order to easily understand the algorithm of the text

1.2 Motivation

Motivation is that in our college and school life we have difficulties in understanding algorithm to write code in various platform. This software will help to overcome it

1.3 Background

Background is to convert text into flowchart, for it to work user doesn't need any prerequisite knowledge and user can expect the result in real time and save it for further use

2. Project Description & Goals

2.1 *Product Perspective*

The aim of the project is to create a Software tool to convert a given pseudo code to a Modelling diagram.

Our objective is to generate a flowchart in image (.PNG) format using pseudo code. The start of the pseudo code is initiated by "START" and the termination of this text file is indicated "END." The Python code converts the pseudo code into a flow chart in blocks.

The input is provided as a text file and desired output is obtained as a PNG file which. This software can be used in UML Design

2.2 *Product Functions or Functional Requirement*

The follow is a table of the requirements that the system SHALL meet. The list of requirements was produced from the initial project documentation provided by the requirements expert.

ID	Stakeholder	Requirement
1	End-User Developer	The End-User SHALL be able to login and change password.
2	End-User Developer	The End-User SHALL be able to create a text anywhere he wishes inside the directory.
3	End-User Developer	All the output image file SHALL will have a unique name.
4	End-User Developer	The End-User SHALL be able to choose a color palette of his wish for his image file.
5	End-User Developer	The End-User SHALL get the image file in the same directory where his text file is.
6	End-User Developer	The Developers SHALL be able to make the software inside any directory.

Table of Shall Requirements

2.3 *User Characteristics*

The following table identifies and describes the different users of our software. The information gathered about the different users of the system helped define what the software needs to do. Also, these users are referenced in the requirements and diagrams.

User	Description
Developer (Admin)	Developer(s) is responsible for developing the entire project starting from which process model to choose for the project to final GUI implementation of the fully functioning and developed software. Developer(s) are same as ADMIN of the project. Only they will have full access to the Project System Files.
End-Users	The End-User is the final recipient of the project. He will have to first create an account and then he can access to all the functionalities of the Software built by the developers.

Table of User Characteristics

3. Technical Specification

3.1 Use Case Diagram & State Transition Diagram

Use Case Diagram

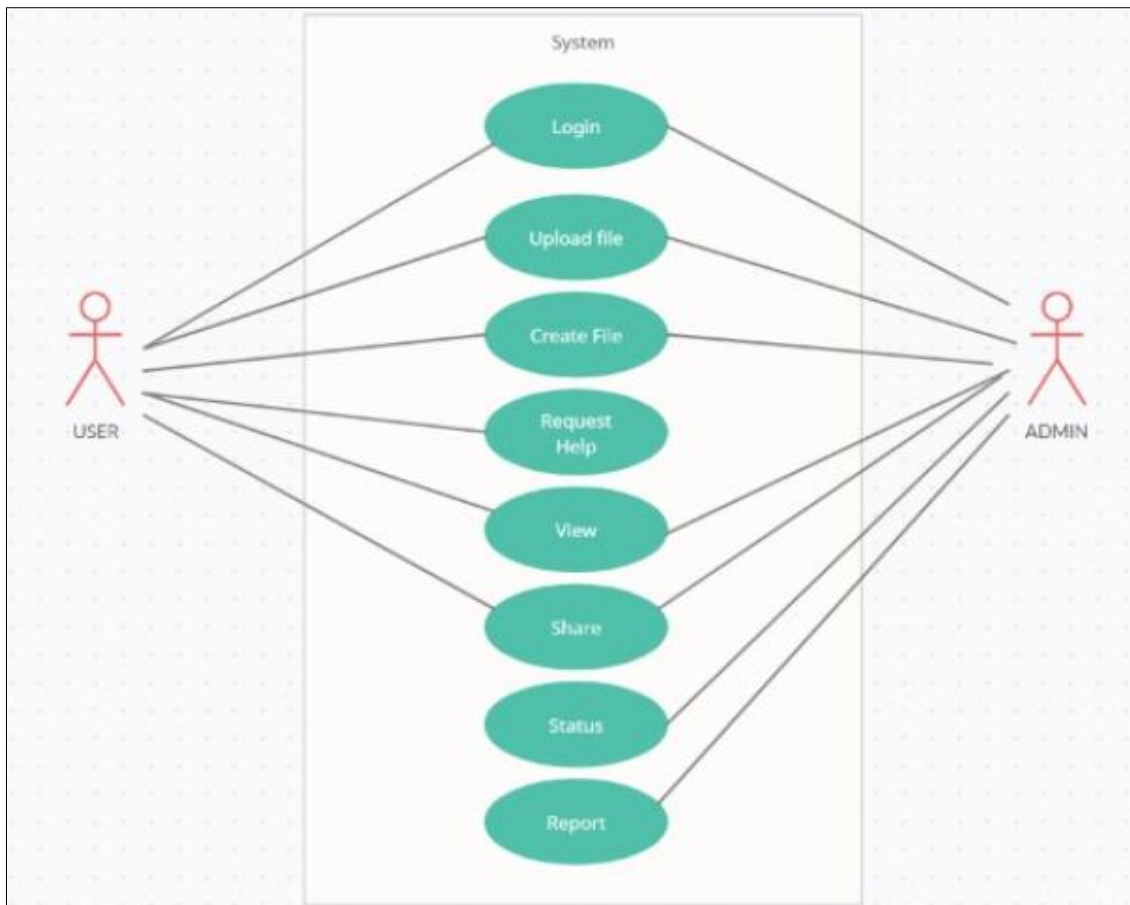


Figure 1: Use-Case Diagram
(By Creately)

State Transition Diagram

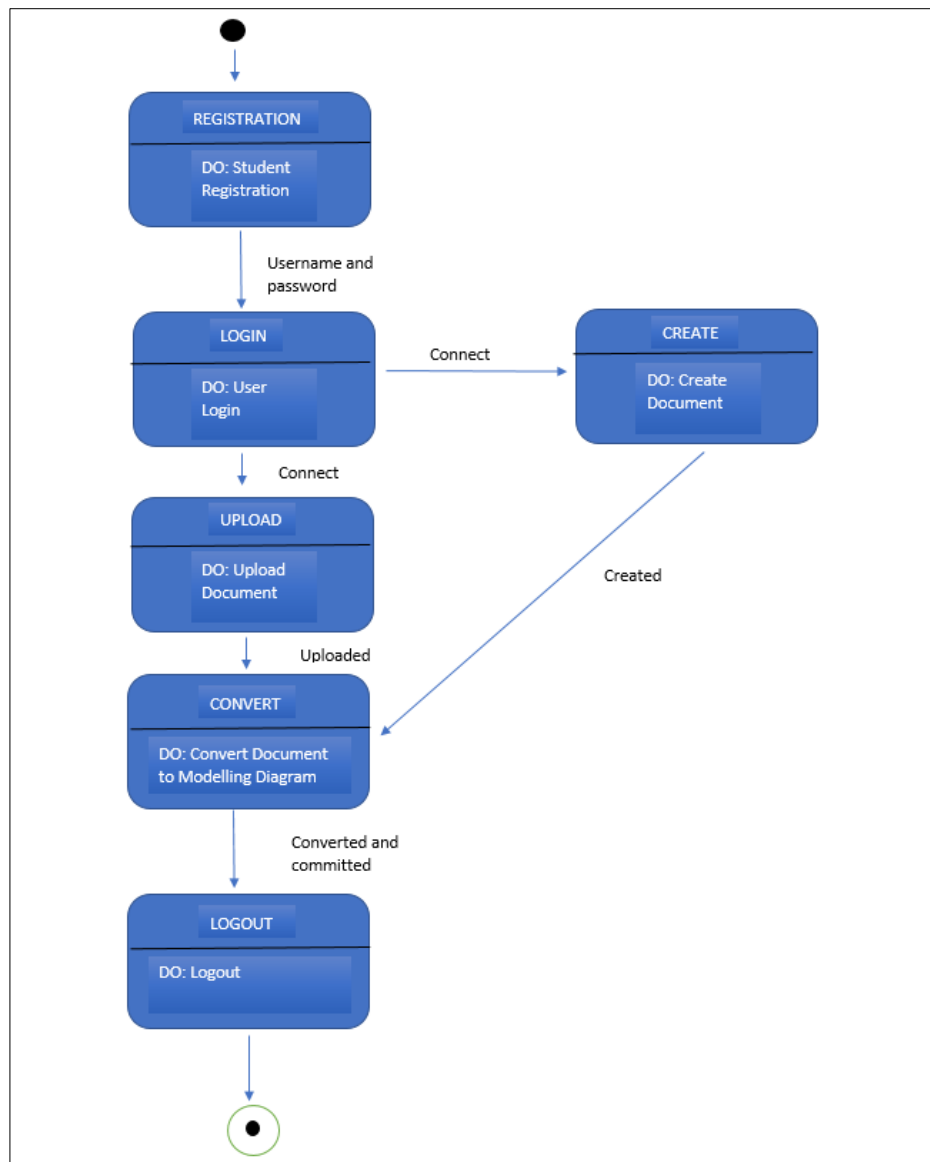


Figure 2: State Transition Diagram
(By Creately)

3.2 Process Model, Architectural Design & Control Style

Process Model

Rapid Action Development Model:

We will be using Rapid Action Development Model because:

- Due to scarcity of time, we need to choose that process model that utilizes our time in the most effective way. RAD is most helpful for quick delivery of a project.
- RAD Model emphasizes on short but effective development cycle which is most suitable when work force is very small like in our case.

Architectural Design

We choose Client Server Architecture because-

Improved Data Sharing:

Data is retained by usual business processes and manipulated on a server is available for designated users (clients) over an authorized access.

Integration of Services:

Every client is given the opportunity to access corporate information via desktop interface eliminating the necessity to log into a terminal mode or processor.

Shared Resources Amongst Different Platforms:

Application used for client-server model is built regardless of the hardware platform or technical background of the entitled software (operating system software) providing an open computing environment, enforcing users to obtain the services of clients and servers (database, application and communication services)

Data Processing Capability despite the Location:

Client-server users can directly log into a system despite of the location or technology of the processors.

Easy Maintenance:

Client-server architecture is distributed model representing dispersed responsibilities among independent computers integrated across a network. Therefore, it's easy to replace, repair, upgrade and relocate a server while client remains unaffected. This unaware change is called as Encapsulation.

Security:

Servers have better control access and resources to ensure that only authorized clients can access or manipulate data and server updates are administered effectively.

Control Style / Model

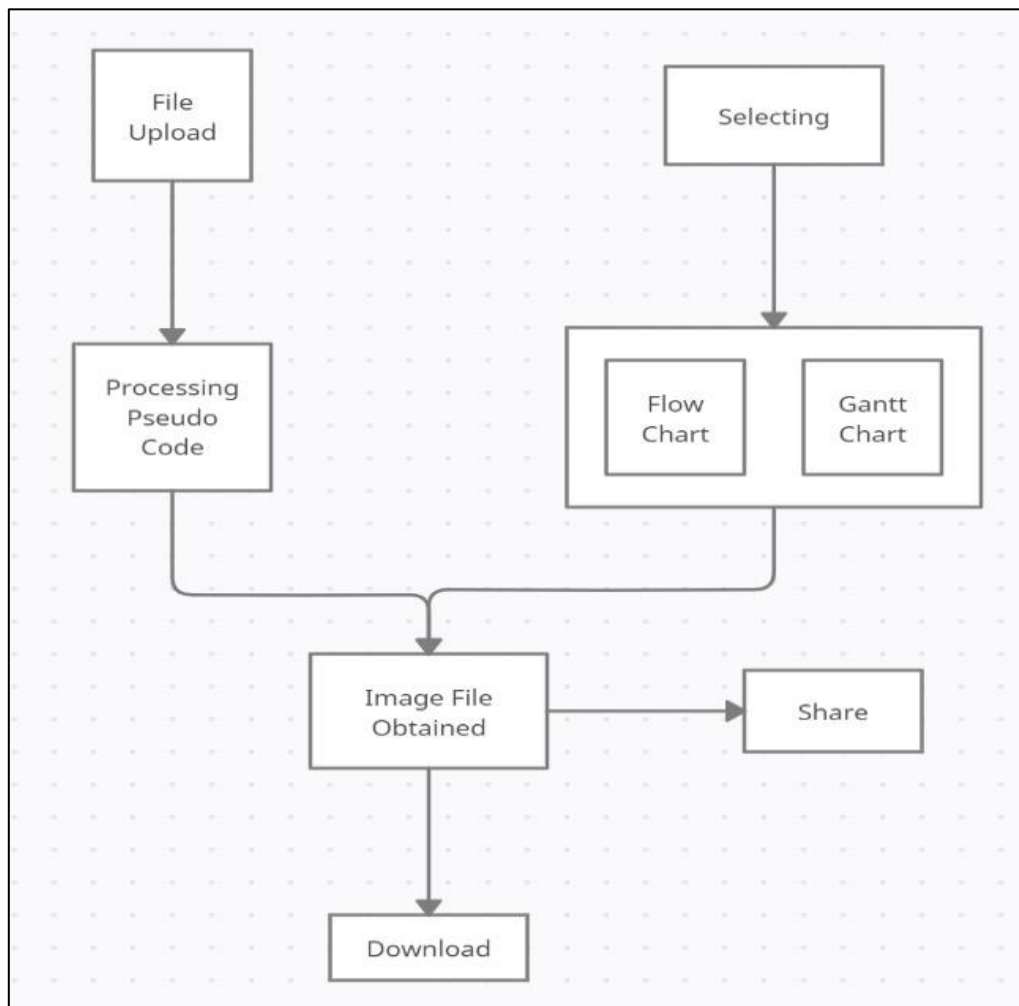


Figure 3: Control Style / Model
(By Creately)

Data Flow Diagram

The following figures represent the data flow diagrams of our software. The first data flow diagram, i.e. figure 1, is the Zero level data flow diagram or Context level data flow diagram.

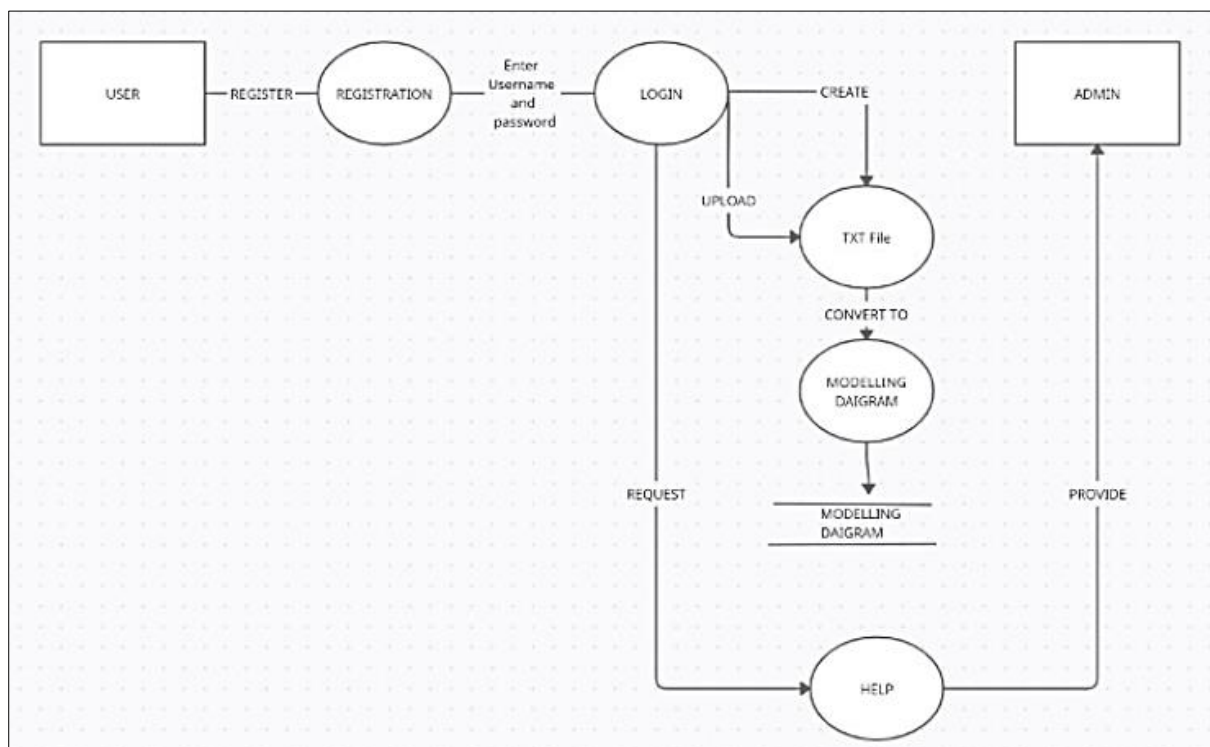
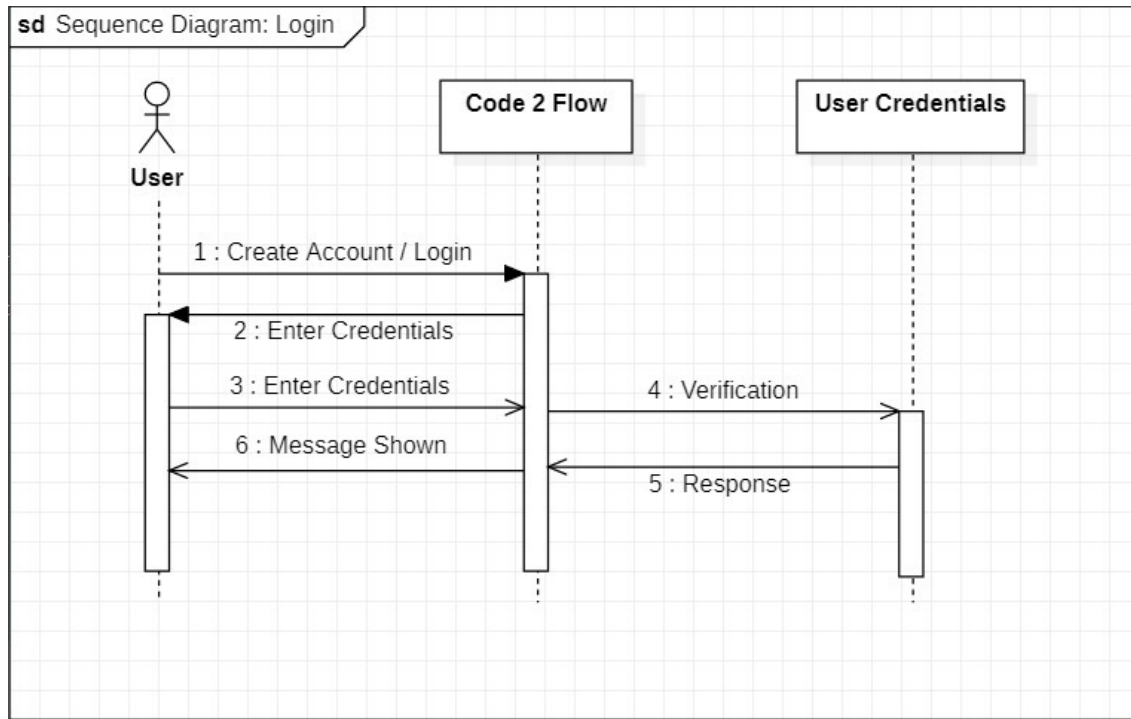


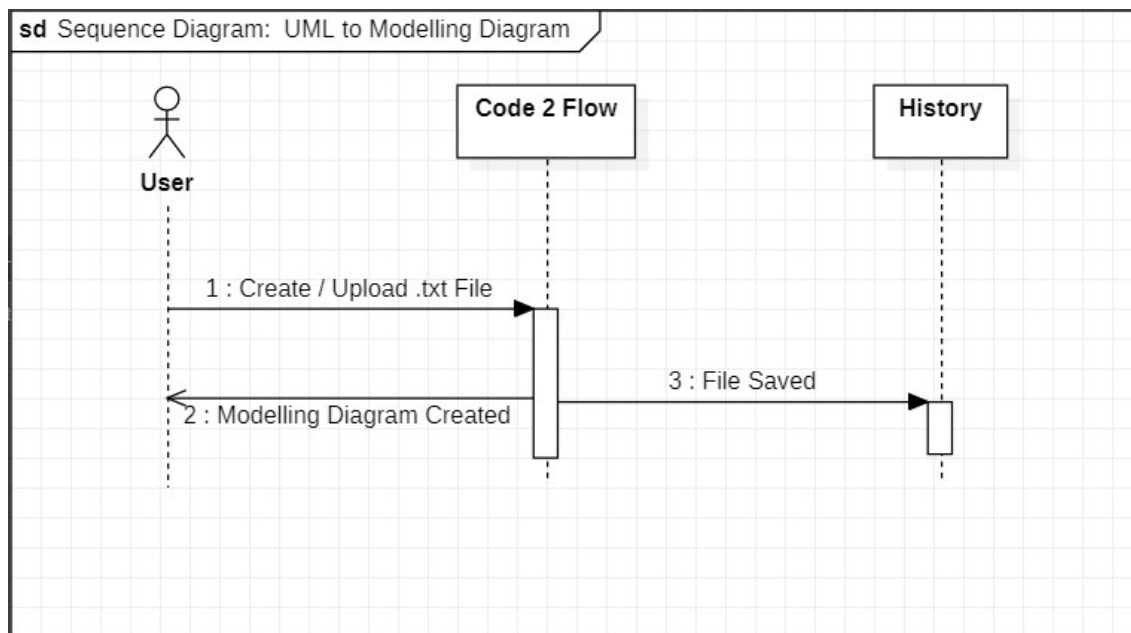
Figure 4: Data Flow Diagram
(By Creately)

3.3 Sequence, Collaboration, Class & Activity Diagram

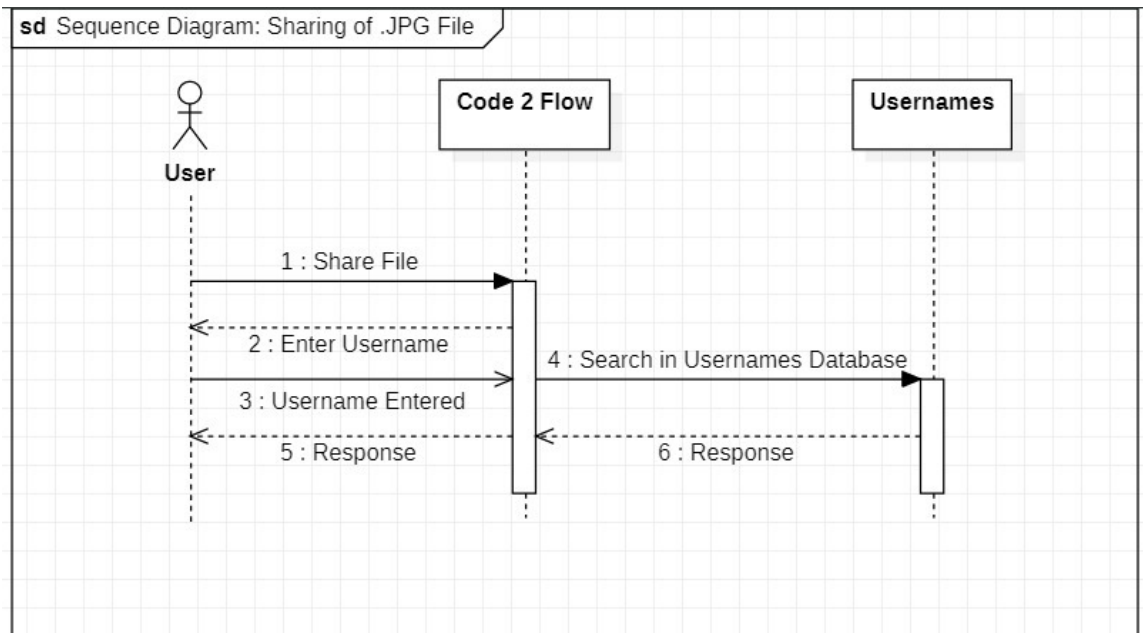
Sequence Diagram



Login: The user logs in or create an account and the details are stored in user credentials table.

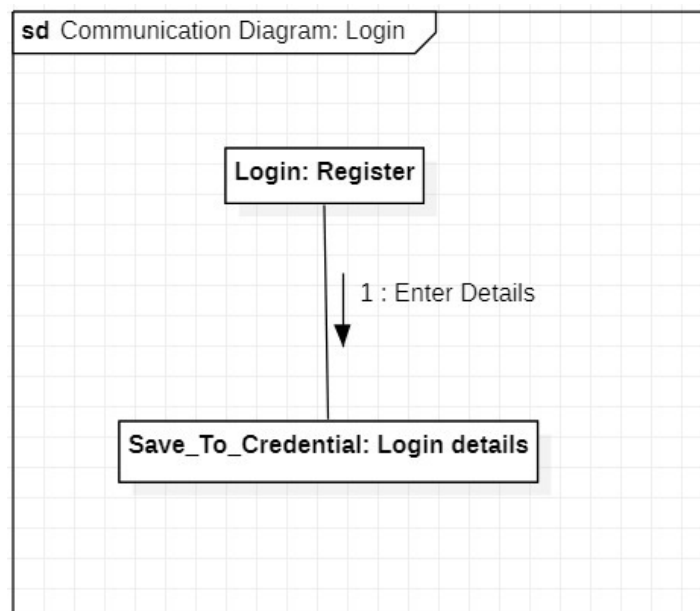


UML to modeling diagram: The user shall upload or create the text file and choose from flow chart or Gantt chart the user wants to create.

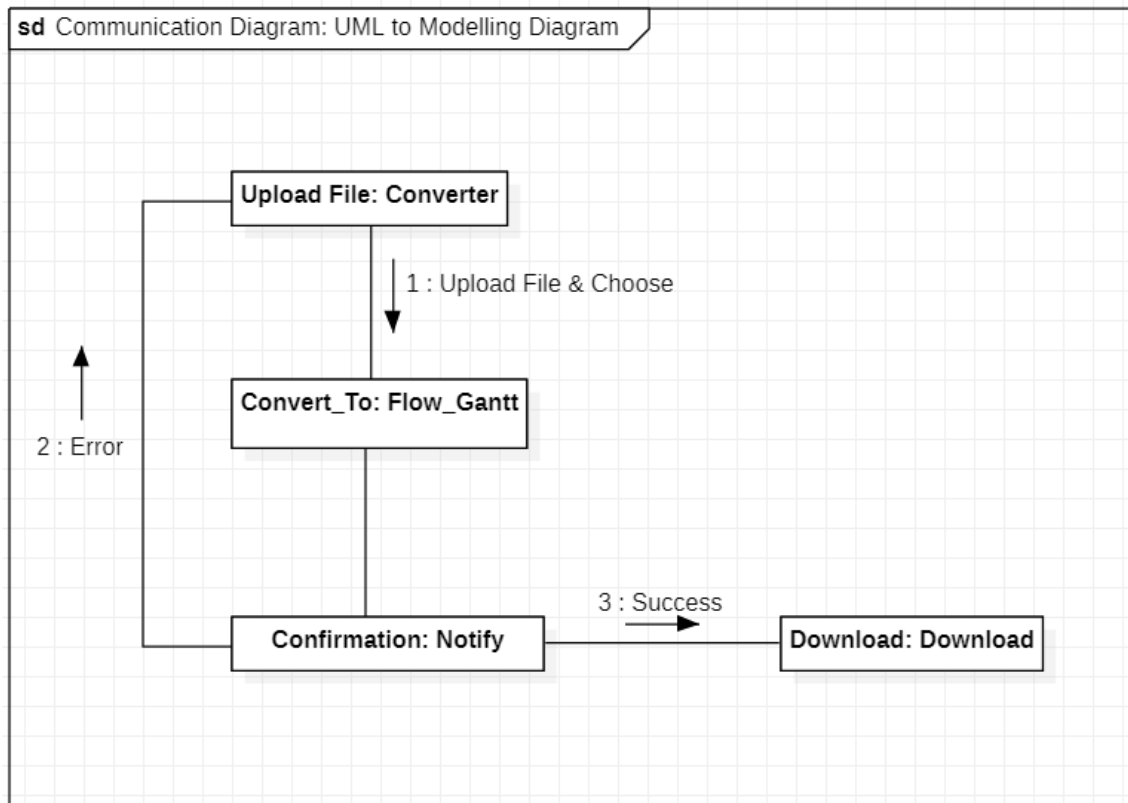


Share: If the user wants to share his jpg with others, the user need to enter the username of other user. If match found then confirmation message is send and if not found error displayed.

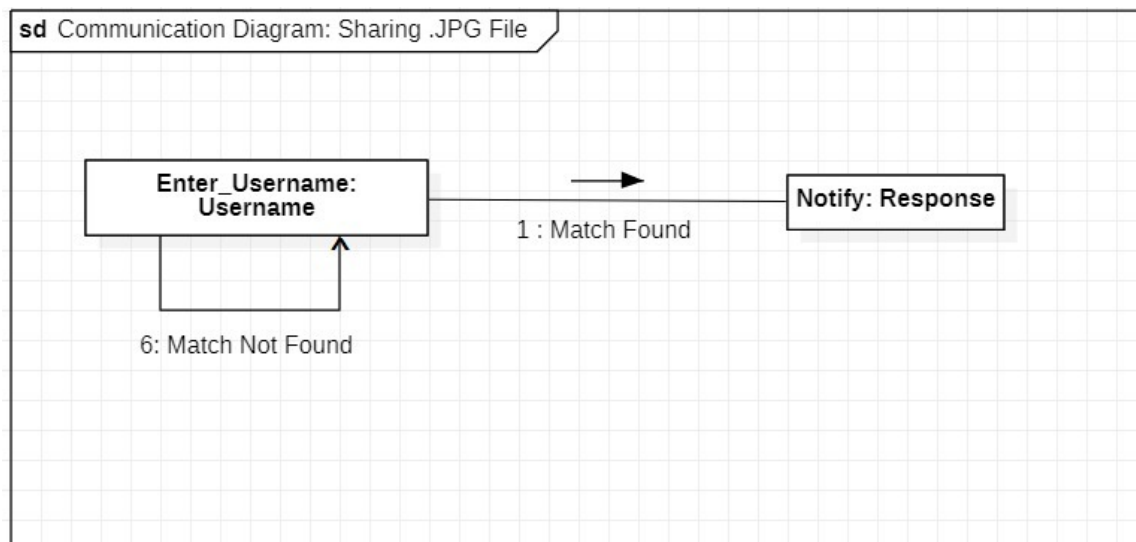
Collaboration Diagram



Login: The user logs in or create an account and the details are stored in user credentials table.



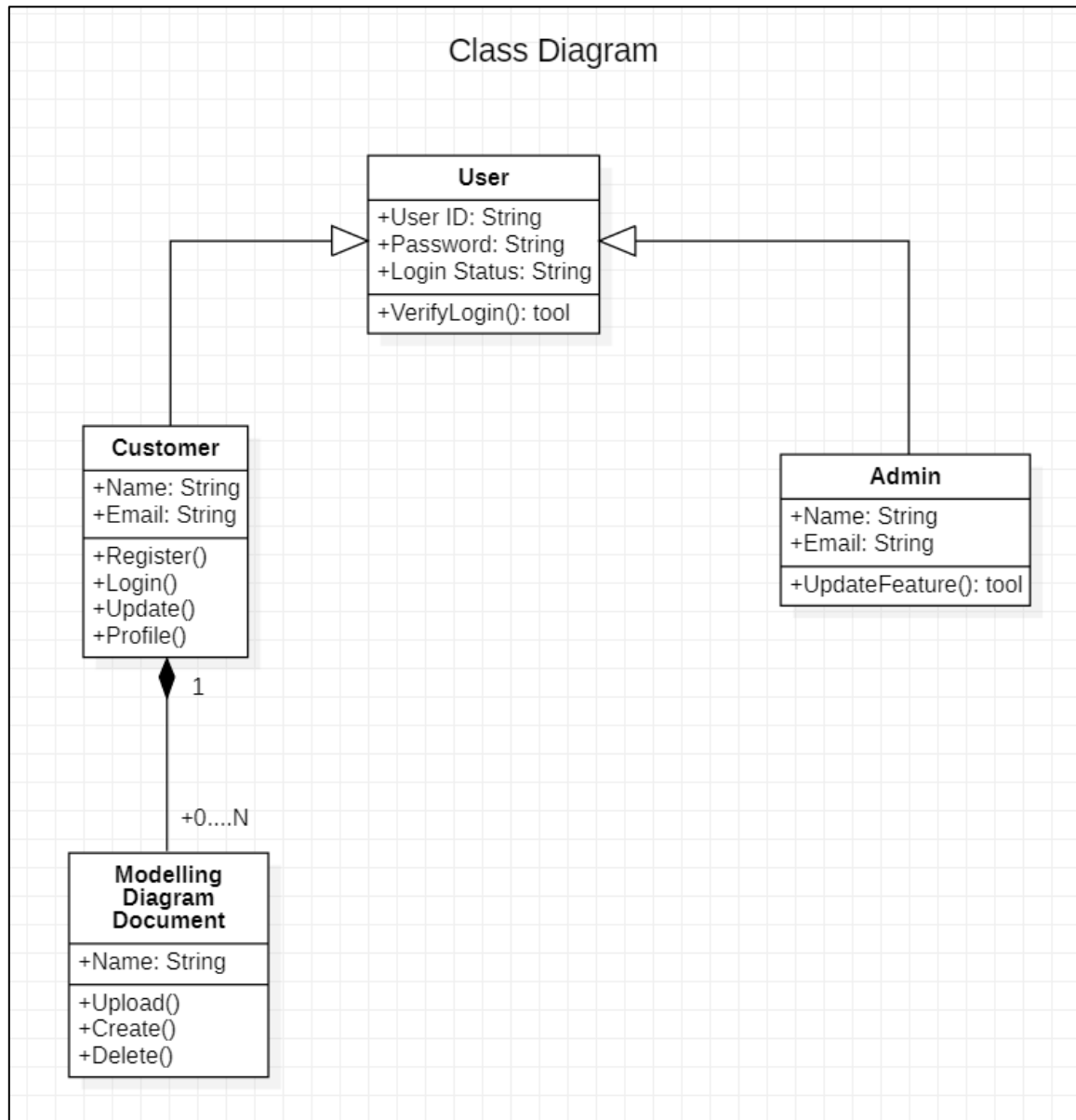
UML to modeling diagram: The user shall upload or create the text file and choose from flow chart or Gantt chart the user wants to create.



Share: If the user wants to share his jpg with others, the user need to enter the username of other user. If match found then confirmation message is send and if not found error displayed.

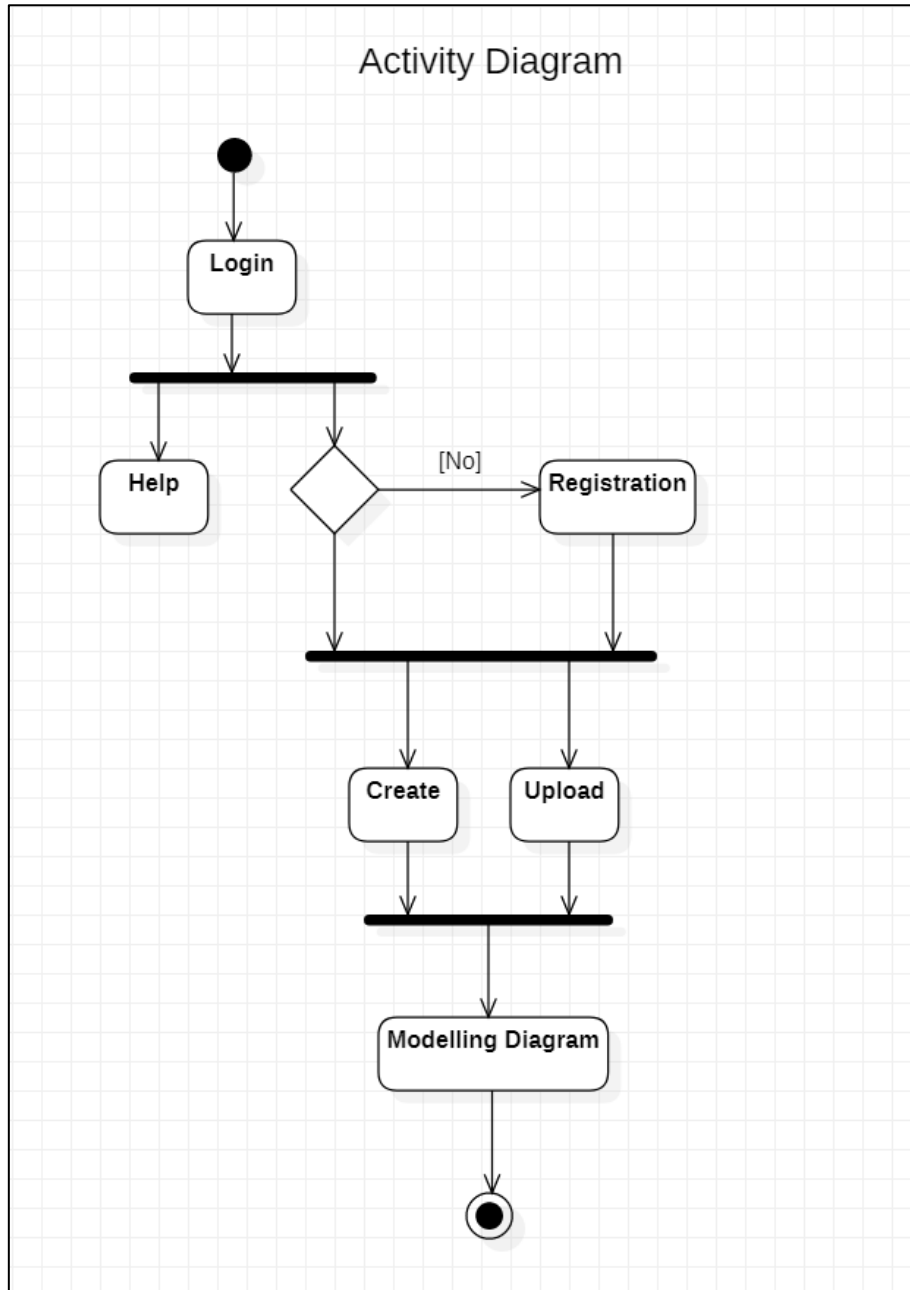
Class Diagram

This is a class diagram. We have four classes' i.e User, admin customer and modelling diagram document in which each have their attributes and methods.



Activity Diagram

This is an activity diagram. It shows the control flow of our software from start point to end point.

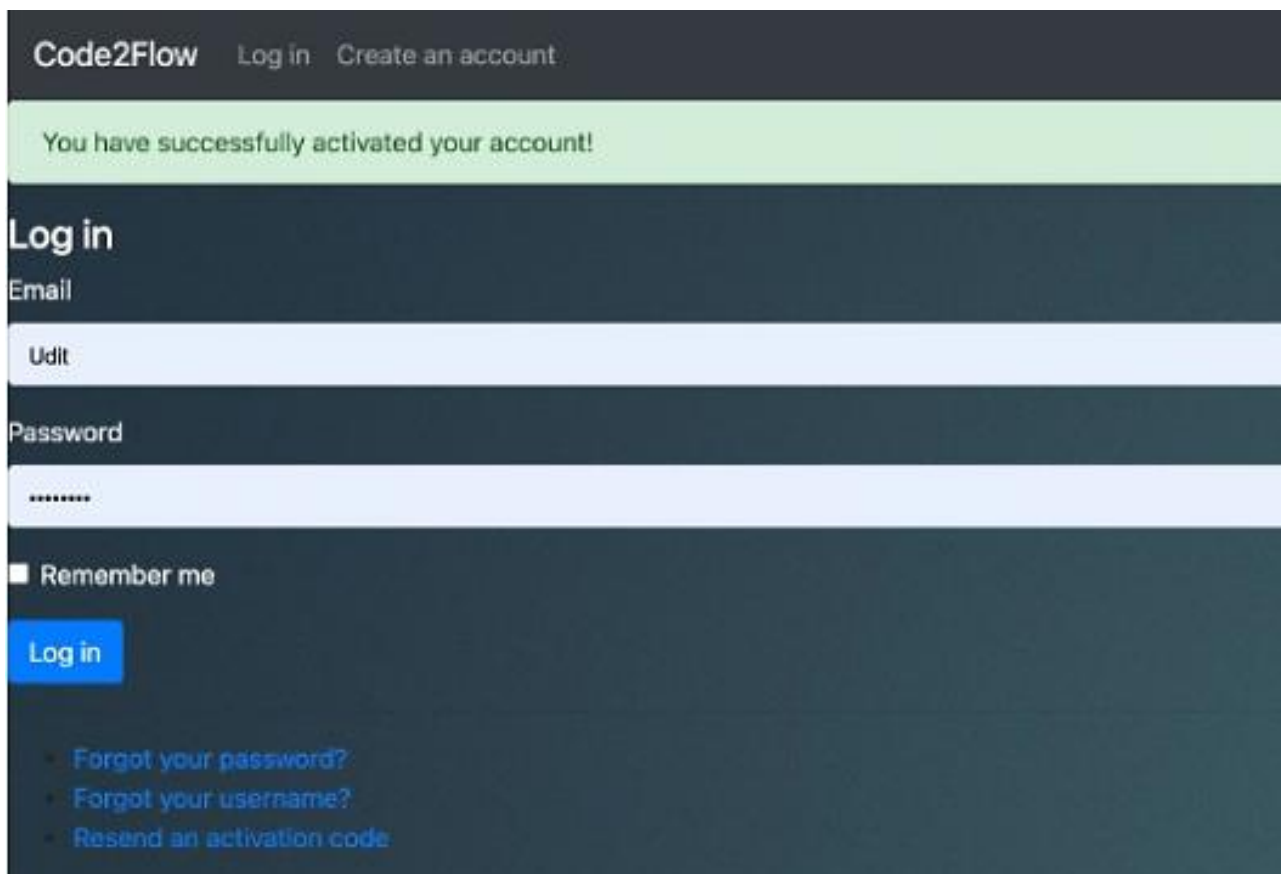


4. Design Approach & Details

4.1 Design Approach

Interface Description

Login Page



Code2Flow Log in Create an account

You have successfully activated your account!

Log in

Email

Udit

Password

☐ Remember me

Log in

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Resend an activation code](#)

Figure 1: Log-In Page

In the Figure 1, The GUI Interface of Code 2 Flow is given. This User Interface can be accessed by using the valid URL or extension. In Figure 2, the Login Module is given. The user can login in the system using his/her valid Email and Password. The user also has the alternate option to login by using his/her Google Account to sign in.

Account Creation Page

Code2Flow Log in Create an account

Create an account

Username

Udit

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

Udit

Last name

Gupta

Email

Udit

Required. Enter an existing email address.

Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation

Enter the same password as before, for verification.

Create

Figure 2: Create an Account Page

In the Figure 2, the Account Creation Module of our Interface. Anyone can become a valid user by using valid and unique e-mail address and password for the account. After entering the same the user will click on the Create Account Button and his/her account will be created.

Text File Upload & Creation Page

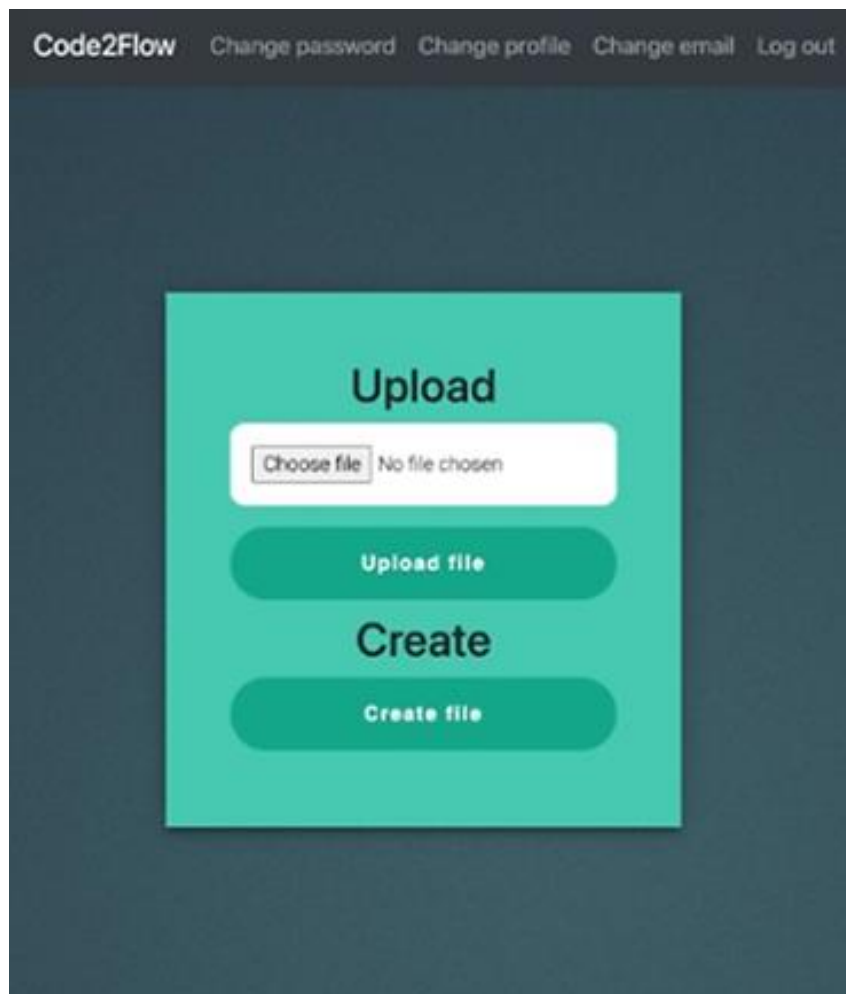
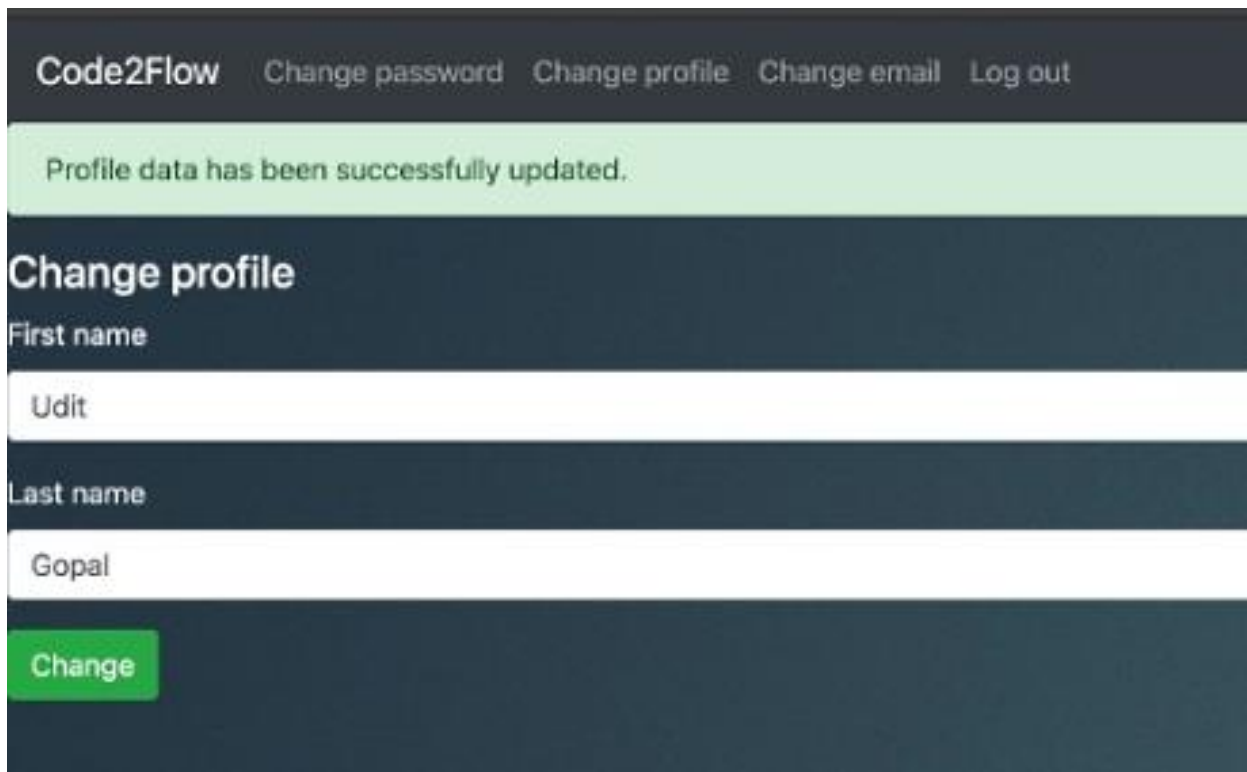


Figure 3: Upload & Create File Page

In Figure 3, we have “Upload Text File” and “Create Text File” buttons. The user can choose either from both the options. If the user already has .txt file in his/her system he can click on the “Upload Text File” button. Else the user can create a .txt file by clicking on the “Create Text File” button.

Changing Username of the user

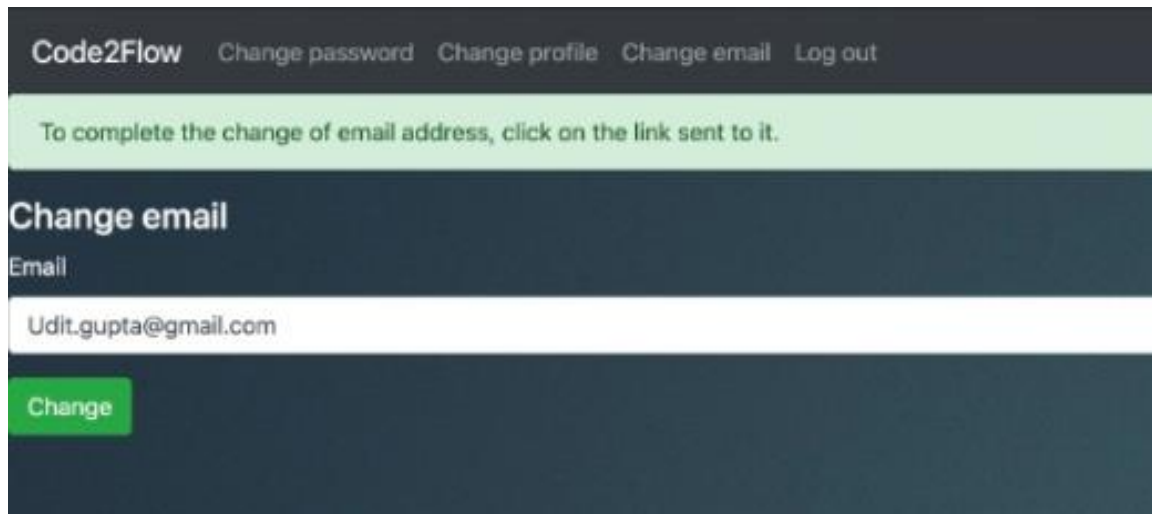


The screenshot displays the 'Change profile' interface. At the top, a dark navigation bar contains the 'Code2Flow' logo and links for 'Change password', 'Change profile', 'Change email', and 'Log out'. Below this, a light green banner shows a success message: 'Profile data has been successfully updated.' The main section, titled 'Change profile', features two input fields: 'First name' with the value 'Udit' and 'Last name' with the value 'Gopal'. A green 'Change' button is positioned at the bottom left of the form area.

Figure 4: Changing Profile Data

In Figure 4, we have the “Change Profile” page. The user can enter (change) his first name and last name as desired. After entering the data, the user clicks on the “Change” button. The message ***Profile data has been successfully updated*** confirms that the changes thus made are saved.

Changing Email of the user



Code2Flow Change password Change profile Change email Log out

To complete the change of email address, click on the link sent to it.

Change email

Email

Udit.gupta@gmail.com

Change

Figure 5: Changing Email Address

In Figure 5, we have the “Change Email” page. The user can enter (change) his email address as desired. After entering the data, the user clicks on the “Change” button. After that, the user clicks on the link sent on his email account provided by him. A subsequent message will pop on the screen denoting that the email has been changed.

Changing Password of the user

Code2Flow Change password Change profile Change email Log out

Your password was changed.

Change password

Old password

New password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation

Change

Figure 6: Changing Email Address

In Figure 6, we have the “Change Password” page. The user can enter his/her desired password such as it fulfils the requirements for setting the password. The user firstly enters the Old Password, then the New Password and enters it again for Confirmation. The user finally clicks on the “Change” button. The message: ***Your password was changed successfully*** confirms that the changes thus made have been saved.

4.2 Codes & Standards

The following table identifies and describes the different users of our software. The information gathered about the different users of the system helped define what the software needs to do. Also, these users are referenced in the requirements and diagrams.

User	Description
Developer (Admin)	Developer(s) is responsible for developing the entire project starting from which process model to choose for the project to final GUI implementation of the fully functioning and developed software. Developer(s) are same as ADMIN of the project. Only they will have full access to the Project System Files.
End-Users	The End-User is the final recipient of the project. He will have to first create an account and then he can access to all the functionalities of the Software built by the developers.

Table of User Characteristics

4.3 Constraints

The follow is a table of the design constraints that the system SHALL meet. The list of constraints was produced from the initial project documentation provided by the requirements expert.

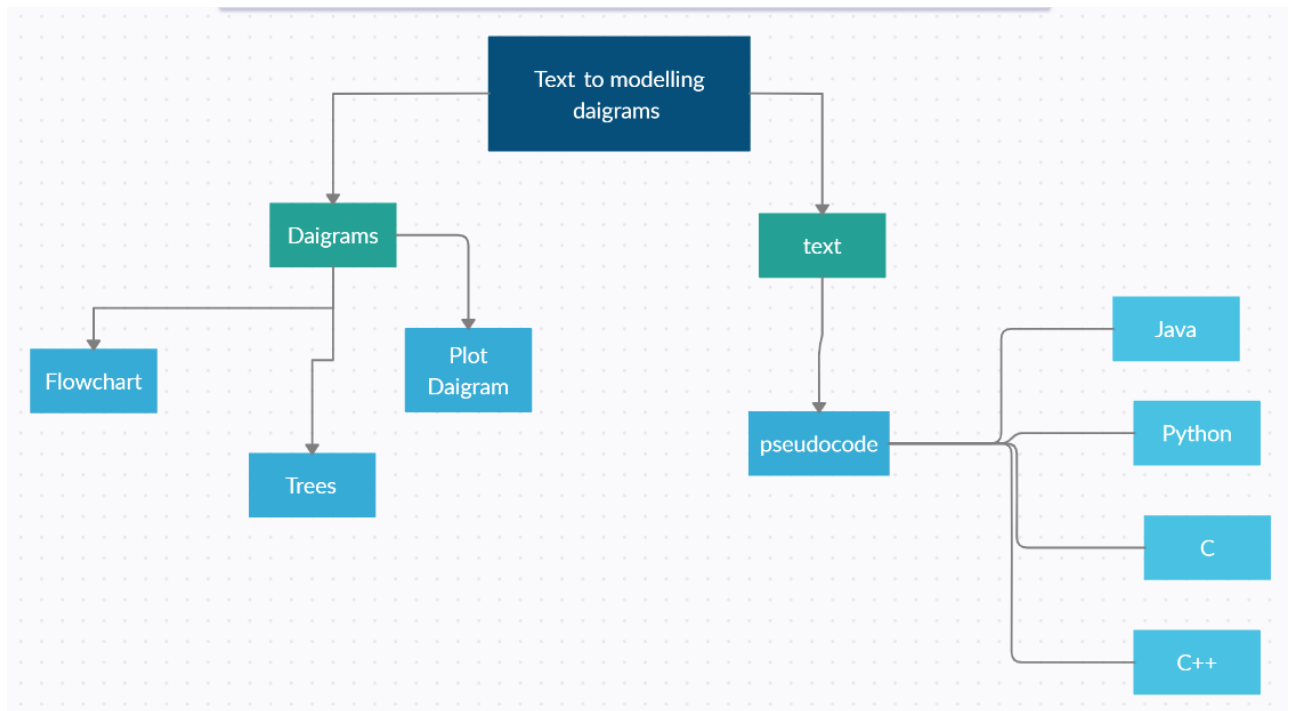
ID	Origin	<i>Shall Requirement</i>
1	End-User Developer	<i>The End-User SHALL create his text in the same directory as the software.</i>
2	End-User Developer	<i>The End-User SHALL follow a certain guidelines while writing a pseudocode.</i>

Table of Design Constraints

5. Schedule, Tasks & Milestones

Work Breakdown Structure

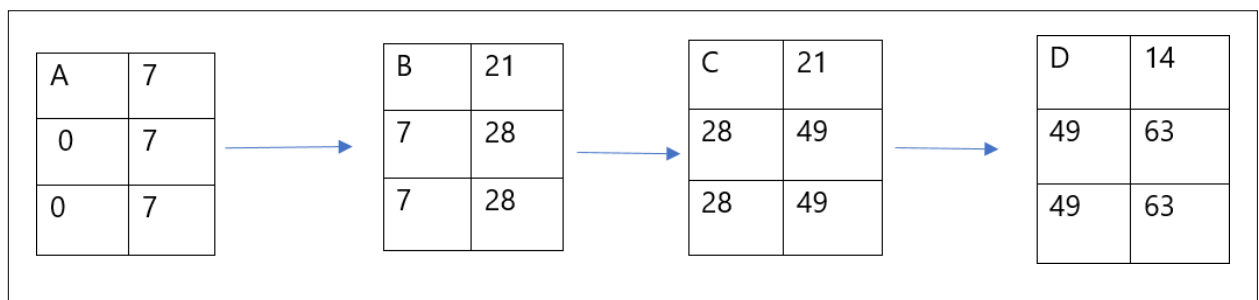
(Creately)



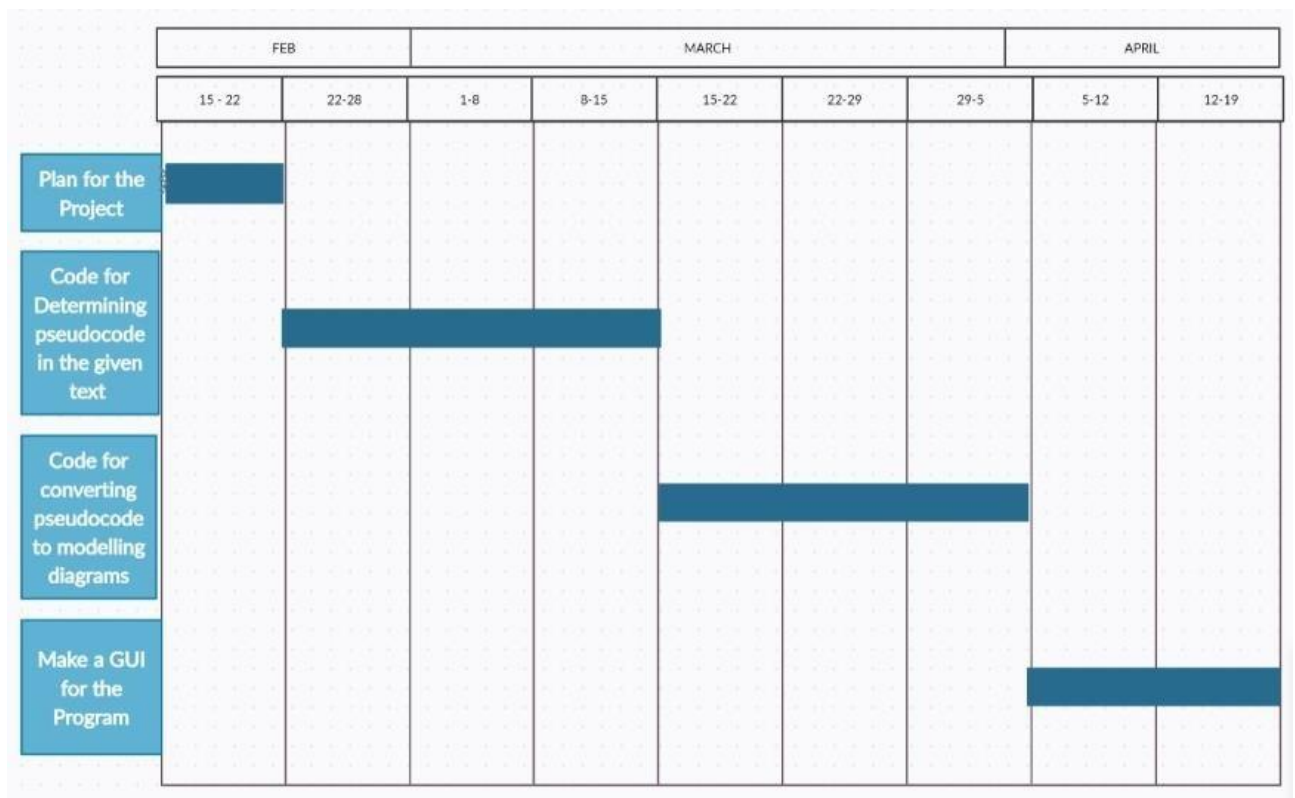
Activity Network

(SmartDraw)

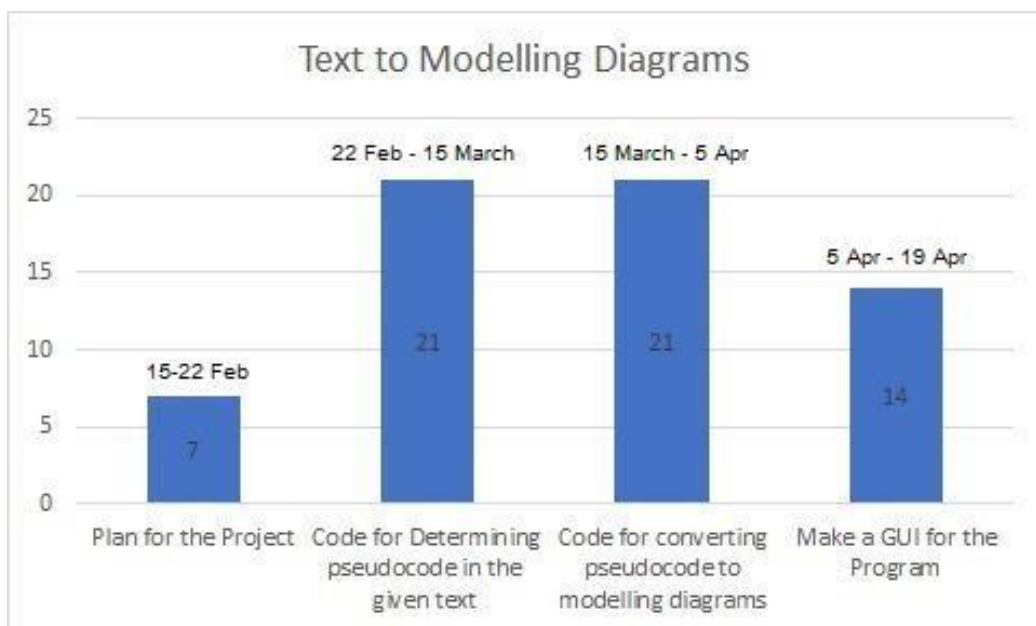
Task	Label	Predecessor	Staff required	Estimated duration
Plan for the Project	A	-	3	7 days (1 week)
Code for Determining pseudo code in the given text	B	A	3	21 days (3 week)
Code for converting pseudo code to modelling diagrams	C	A,B	3	21 days (3 week)
Make a GUI for the Program	D	A,B,C	3	14 days (2 week)



Gantt Chart (Creately)



Time line Chart (InstaGantt)



6. Project Demonstration

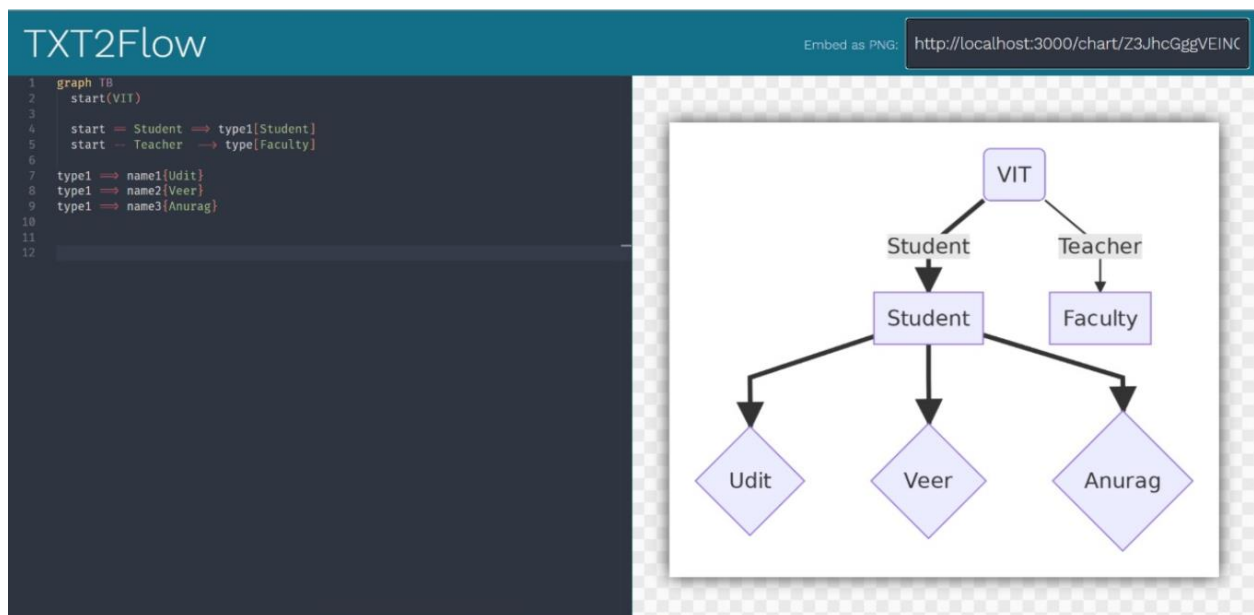


Figure 7: [L to R] Text (Pseudocode) & Image (Modelling Diagram / Flow Chart)

In Figure 7, on Left, the code is typed which is the Text (Pseudocode) part of the project. And on Right, we get the corresponding Image for the pseudocode as soon as we finish typing. On Top right side we can see feature ***Embedded as PNG***. From there we can download the image as a .PNG file if we wish to do so.

Non-Functional Requirements

Software System Attributes

Reliability

Thorough unit, milestone, and release testing will ensure reliability in the software. Comprehensive test scenarios and acceptance criteria will be established to reflect the necessary level reliability required of the software. The all delivered source code will be thoroughly tested using the established test scenarios until the acceptance criteria are satisfied by our software.

Maintainability

Our Project requires no additional or special efforts to maintain it. Till a text file created following the guidelines for writing the Pseudocode, an Image file of the respective pseudocode's diagram will be created. Since Python is a free software as all other resources are, we can theoretically ensure the maintainability of the project forever.

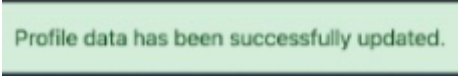
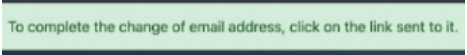
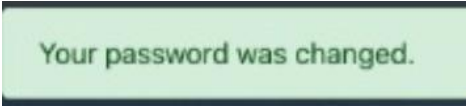
Portability

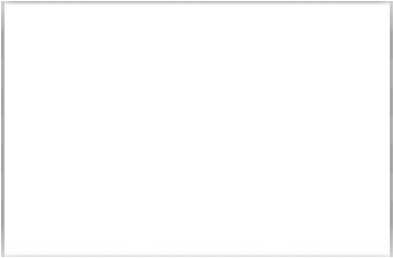
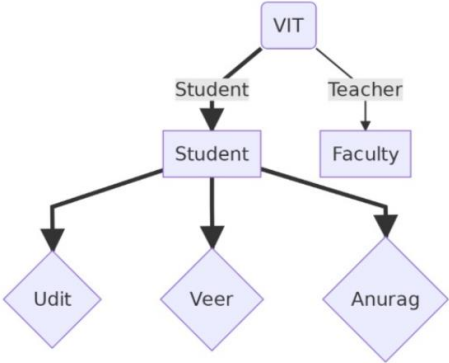
Since, both the Pseudocode and Modelling Diagram will be in simple English we can take the advantage of the language.

It is safe to say that the implementation of our software will be able to be ported to other system platforms that accept Python applications with little to no changes required. Also no significant changes are required to ensure proper execution on other Operating System platforms.

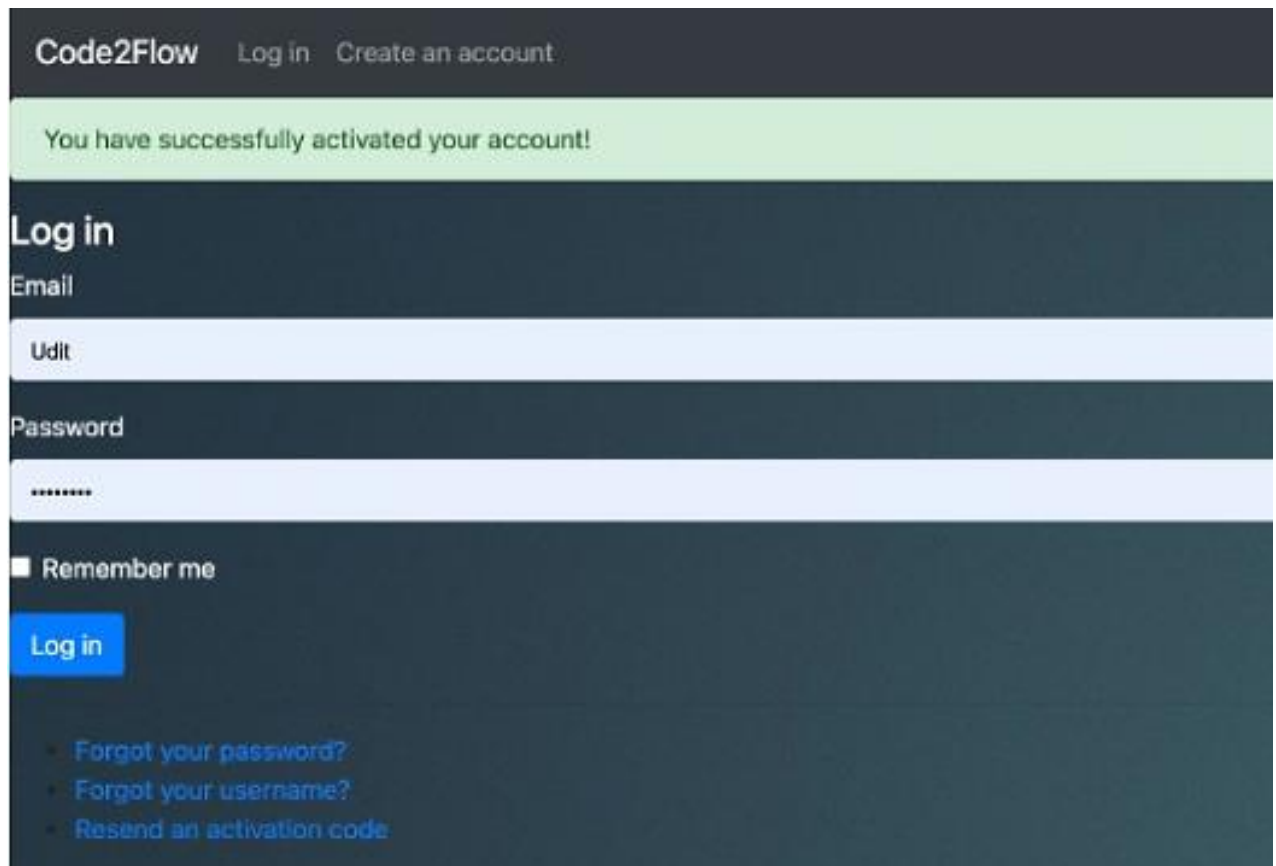
Test Cases

Test ID	Test Objective	Test Data	Expected Result (Expected Output Message)	Actual Results (Actual Output Message)* *[Screenshots are cropped]	Test (Pass /Fail)
1.	Check the validity of Email entered while creating a new account in the "Create an Account" Page	Email: Udit.gupta	Please include a '@' in the email address. '<Username >' is missing a '@'.		Pass
2.	Check whether the account is active or not i.e. the validity of Email in "Login" Page	Email: Udit.gupta@gmail	This account is not active.		Pass
3.	Check the validity of Password in "Login" Page (Provided Username i.e. E-mail is correct)	Email: Udit.gupta@gmail.com Password: <blank>	You entered an invalid password.		Pass
4.	Check whether the user's account activates or not when valid Email & Password is entered in the "Login" Page.	Email: Udit Password:	You have successfully activated your account. And "Upload & Create"		Pass

	Also, ensure that the "Upload & Create" page opens.		page shall open		
5.	Check whether the user is able to change his/her Username successfully or not	First Name: Udit Last Name: Gopal	Profile Data has been successfully updated.		Pass
6.	Check whether the user is able to change his/her E-mail successfully or not	Email: Udit.gupta@gmail.com	To complete the change of email address, click on the link sent to it.		Pass
7.	Check whether the user is able to change his/her password successfully or not	Email: Udit.gupta@gmail.com	Your Password was changed		Pass

8.	Check whether the user is able to restore his/her password successfully or not	<div>New Password: New Password Confirmation:</div>	<div>Please enter your new password twice so that we can verify you have typed it correctly.</div> <div>Your password has been set. You may go ahead and login now.</div>	<div><div>Restore password</div><div>Your password has been set. You may go ahead and log in now.</div></div>	Pass
9.	Check whether appropriate Image File is obtained or not when a Text File is uploaded (Provided user was able to successfully login)	<pre>graph TD start(VIT Email) start -- Student --> type1[Student Email] start -- Teacher --> type[Faculty Email] type1 --> name1[Udit] type1 --> name2[Veer] type1 --> name3[Anurag]</pre>	Blank Page		Pass
10.	Check whether appropriate Image File is obtained or not when a Text File is uploaded (Provided user was able to successfully login)	<pre>graph TD start(VIT) start -- Student --> type1[Student] start -- Teacher --> type[Faculty] type1 --> name1[Udit] type1 --> name2[Veer] type1 --> name3[Anurag]</pre>	Corresponding Image File		Pass

7. Result & Discussion



The screenshot displays the Code2Flow login interface. At the top, a dark header contains the site name 'Code2Flow' and links for 'Log in' and 'Create an account'. Below this, a light green banner confirms 'You have successfully activated your account!'. The main section is titled 'Log in' and includes an 'Email' label above a light blue input field containing the text 'Udit'. Below the email field is a 'Password' label above another light blue input field filled with dots. A 'Remember me' checkbox is positioned below the password field. A blue 'Log in' button is located at the bottom left of the form area. At the bottom right, there are three links: 'Forgot your password?', 'Forgot your username?', and 'Resend an activation code'.

Code2Flow Log in Create an account

You have successfully activated your account!

Log in

Email

Udit

Password

.....

☐ Remember me

Log in

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Resend an activation code](#)

[Code2Flow](#) [Log in](#) [Create an account](#)

Create an account

Username

Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

Last name

Email

Required: Enter an existing email address.

Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation

Enter the same password as before, for verification.

