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

SOFTWARE ENGINEERING FINAL PROJECT REPORT

COURSE CODE:CSE3001

SLOT-D2

TOPIC: TOOL TO SUPPORT TIME COMPEXITY OF PROGRAMS

PREPARED BY:-

1. ADITYA NARAYAN(19BCE2172)
2. KISHAN KUMAR SHARMA(19BCE2569)

COMPLETED UNDER THE GUIDANCE OF:

DR.SWATHI J.N

ACKNOWLEDGEMENT

Our team would like to express our special thanks of gratitude towards our Software and Engineering faculty Dr.Swathi J.N for providing us the golden oppurtunity of working on this project :Tool to support time complexity of Programms.

Moerover she has always been a constant guide and a supportive figure in the development,processing and fruitful execution of this project.She has always been a helping hand in case of any ambiguity of glitch regarding the working of project.

Through this project we have atttained innumerable knowledge in the field of software engineering and its methodology and also gained skills highly relevant in this sector.

EXECUTIVE SUMMARY

This project deals with the development of a tool in form of a software meant to compatible in any standard application.

The software would basically compute the time complexity of any program passed in through as an input by the user.An additional feature of determining the case of time-complexity has also been introduced.

It would be an aid to the coders/developers who intend to determine their code's complexity and would hence enhance it accordingly.

A time span of 3 months has been spent over the planning,designing and execution of this software.

Free source UML tool Lucid Chart has been used for designing the specifications.Pencil(a free source tool) has been used for the development of the mock GUI model.

Python has been used for the purpose of implemetatiion of code,and XCODE has been used for the development of software.

It is a.py type of software.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1.IDE-Integrated Development Environment

2.PY-Python Extension

3.HTTP-Hyper Text Transmission Protocol

4.www-world wide web

5.etc-Et Cetera

6.GUI-Graphic User Interface

7.HTTPS-Secure Hypertext Transfer Protocol

1. INTRODUCTION

1.1 OBJECTIVE

The sole objective of this project is to develop a software as a tool for computation of the time complexity of the given piece of code generated by the user and then inform the user about the analytical case of the complexity computed, such as worst case,best case complexity and hence allowing the stakeholder to analyse the code in a better way.

1.2 MOTIVATION

In the process of software development,magnamous chunks of code are written resulting in strenous load on the computational software.Hence in such cases the codes with economical time complexities would be more prefferable.Henceforth the complexity of the code involved comes into larger play.If computed to a degree of accuracy the coders can therefore be aware of the efficiency of code and enhance their algorithm accordingly..

2. BACKGROUND

During development of a software,the length of codes written are generally magnamous in size,hence problems related to their under efficiency occur frequently.Hence it becomes a matter of paramount importance that the complexity of the code be meticulously computed and the effciency of the code can be improved henceforth so that it results in less strenous on the compiler and computation device.

2.PROJECT DESCRIPTION AND GOALS

2.1 GOALS:

1.Computation of time complexity of given code.

2.Analyse the case of the given code and recommend it back to user.

2.2DESCRIPTION:

The software would alow the user to register himself and henceforth he can login himself in the system.The code should fall under the category of the prescribed languages by the system.The user is advised to enter the code with syntactical clarity.Moreover it solely depends upon the descretion of the user whether he wants to utilise the recommendation option provided.The result would be computed in cyclometric fashion.The project would take approximatly 3 months in its completion.

3.TECHNICAL SPECIFICATIONS

Softwares used for design development

1.LUCID CHART:An open online source tool for creating uml diagrams.

2.STAR UML: An open source tool for creating and designing uml diagrams.

3.PENCIL:An open source tool for designing mock interfaces.

4.CREATELY:-A free open source tool for designing flow chart diagramms.

Tools used for code compilation and software development

1.PYTHON:A programming language for code compilation.

2.XCODE:An apple based software development tool

3.GITHUB:A provider for software development.

4.TKINTER:-A python binding for GUI development kit.

4. DESIGN, APPROACH AND DETAILS

4.1 DESIGN APPROACH/MATERIALS AND METHODS

SOFTWARE MODEL ADOPTED :

Waterfall model has been adopted as this is a simple layered project ,the requirements are well understood and clear. -In this project the parallel running of processes won't be feasible hence the above mentioned model is better to operate with.

MODULES DESCRIPTION:

The time complexity tool has been decomposed into the following modules.

• Configuration Module: This module collects data from the user to be used for establishing communication with the user.

• Authentication Module: This module authenticates the data given by the user.

• Language selector Module: This module will help the user to select any one programming language in which the time complexity of the code has to be calculated.

• Code section Module: This module will collect the code from the user whose time complexity has to be calculated.

• Result and Recommendation Module: This module will show the time complexity of program and recommend the best suitable program for the user

The various design related model diagrams have been shown below.

1. DATA FLOW DIAGRAM

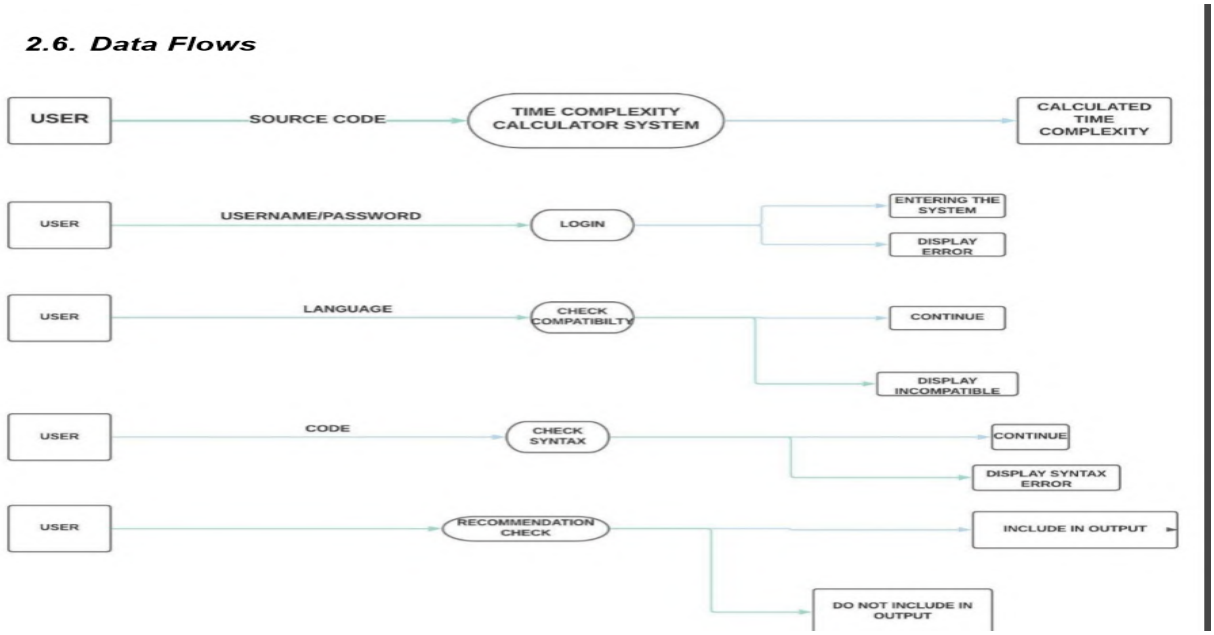


Fig:4.1.1

2. STATE TRANSITION DIAGRAM

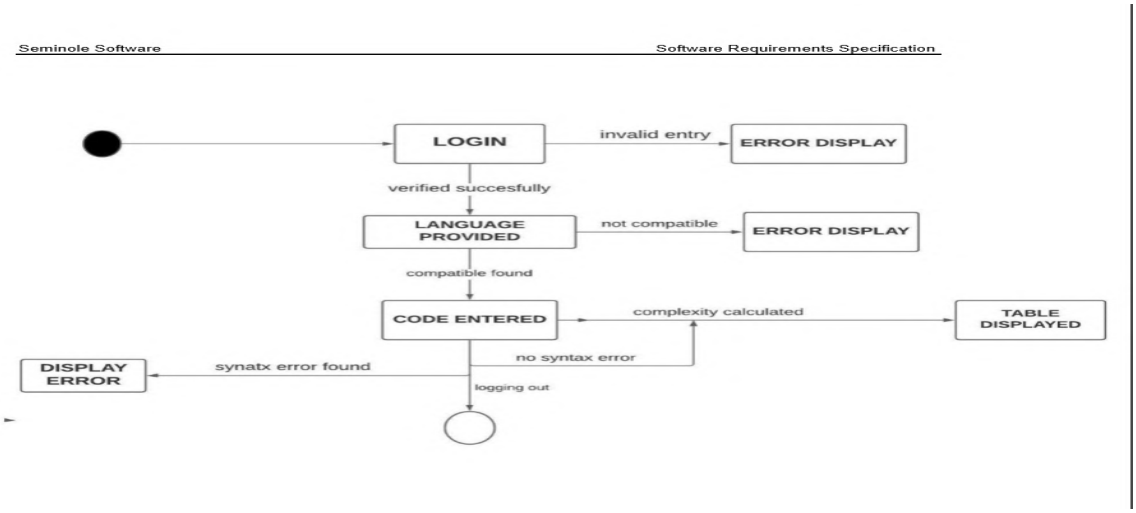


Fig 4.1.2

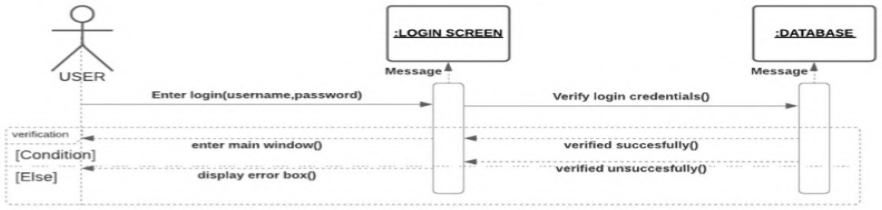
3.USE CASE DIAGRAM

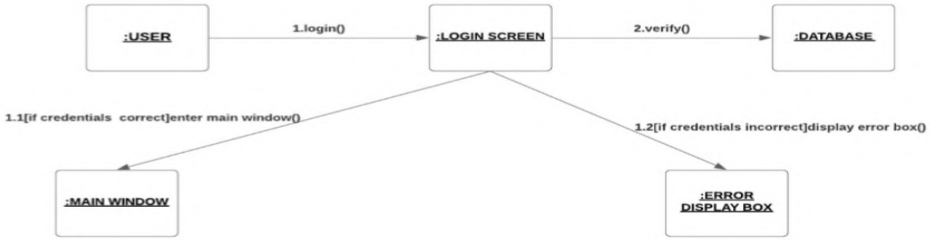
Fig:4.1.3

4.SEQUENCE AND COLLABORATION DIAGRAM

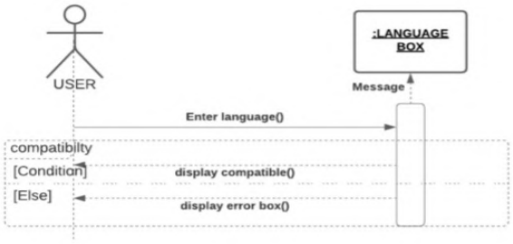
Sequence and Collaboration Diagram:

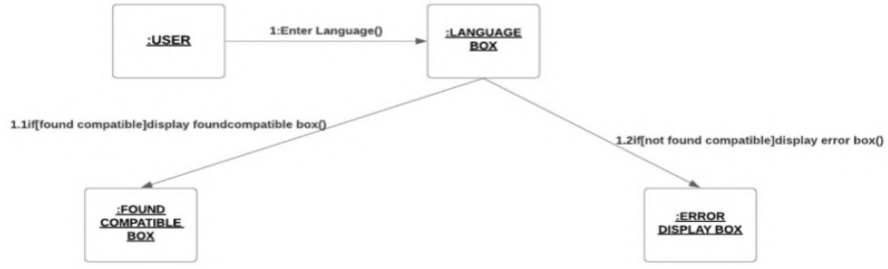
1. Login system:



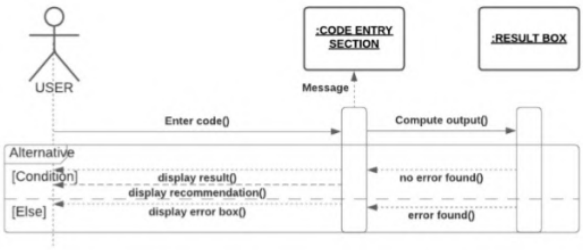


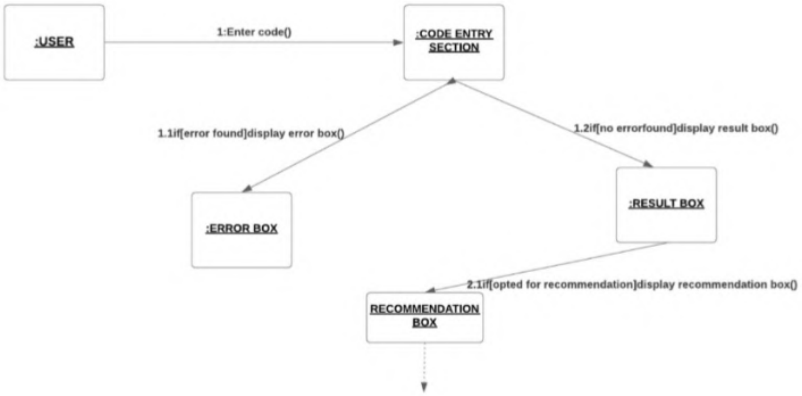
2. Language selector:



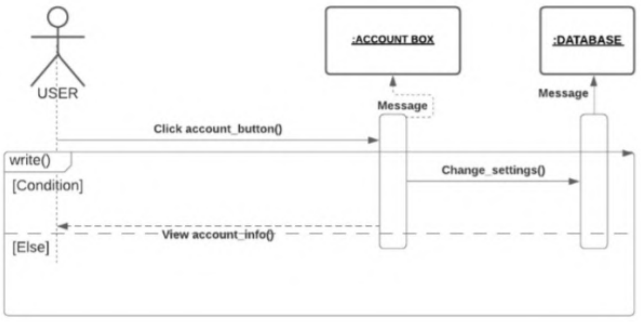


3. Code entry section:





4. Account info:



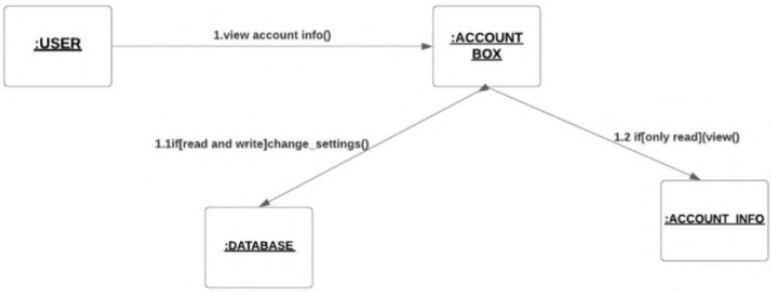


Figure 4.1.4

5.CLASS DIAGRAM

Class Diagram:

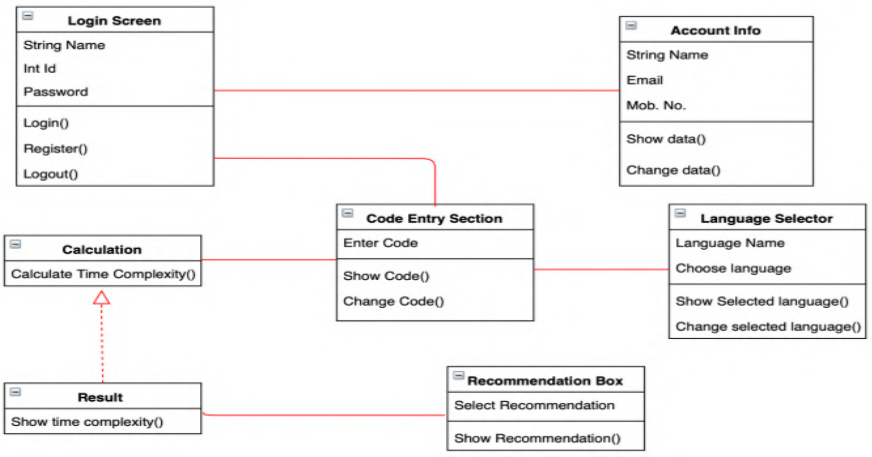
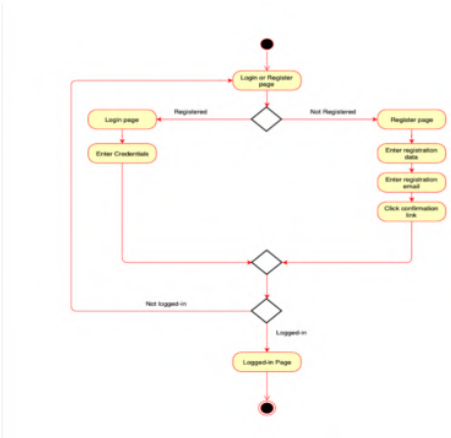


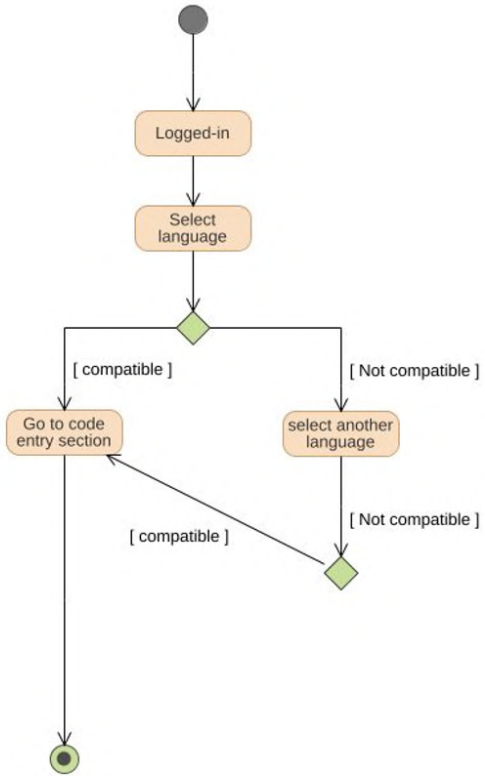
Fig:4.1.5

ACITVITY DIAGRAM

Activity diagram:

1. Login system:




Logged-in
Select language
[compatible]
[Not compatible]
Go to code entry section
select another language
[Not compatible]
[compatible]

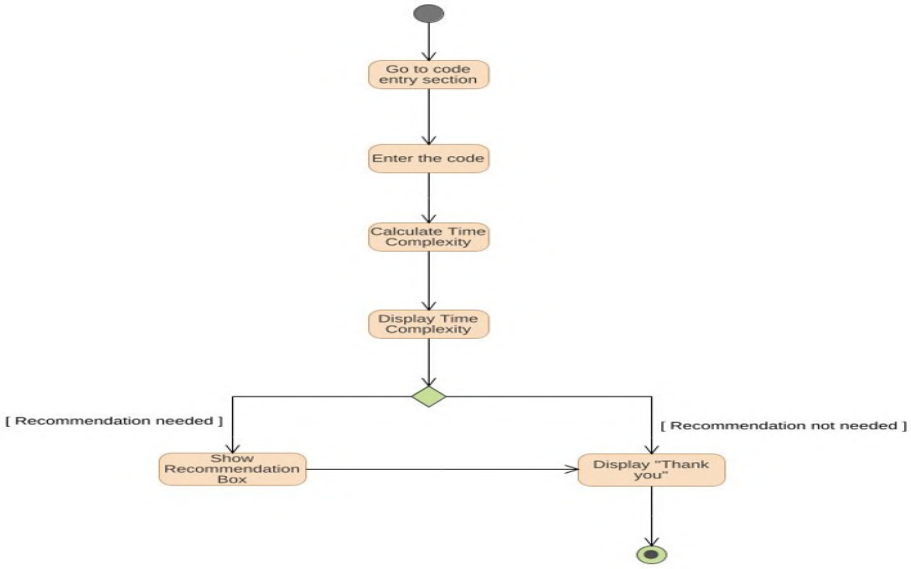

Go to code entry section
Enter the code
Calculate Time Complexity
Display Time Complexity
[Recommendation needed]
[Recommendation not needed]
Show Recommendation Box
Display "Thank you"

Fig 4.1.6

ARCHITECTURE MODEL

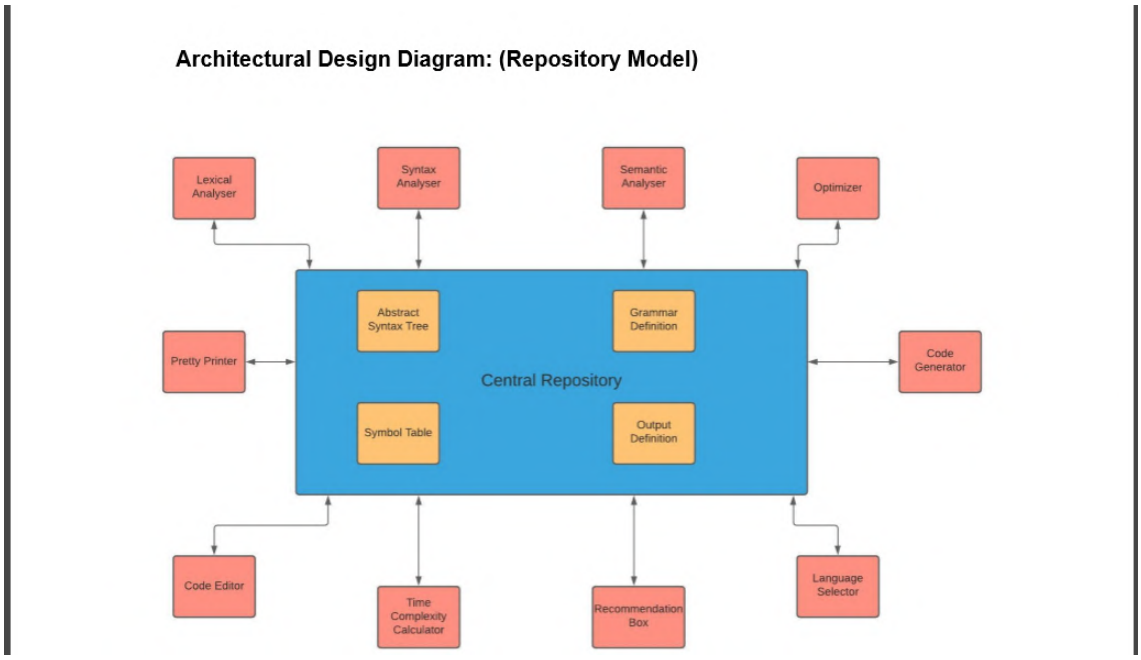


Fig:4.1.7

CONTROL MODEL SELECTION:

Call return control model has been selected as the control model for our system as our system deal with the sole task of time complexity calculation,(it being the centralised subsystem) all the others subsystems are sequentially dependent on it. Elaborately as we speak, the login process, the checking of compatibility, and the checking of syntax ,they all are done in sequential manner eventually leading to the computation of time complexity.

INTERFACE MOCK DESIGN DIAGRAM

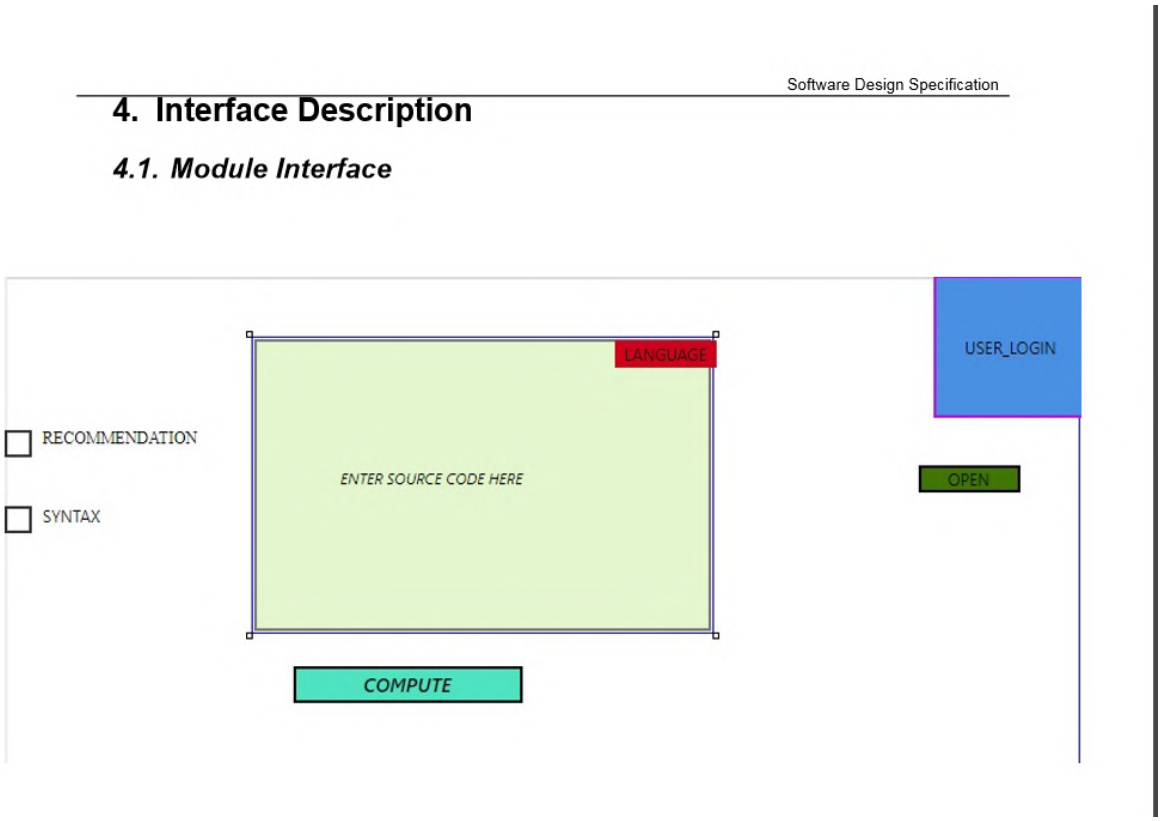


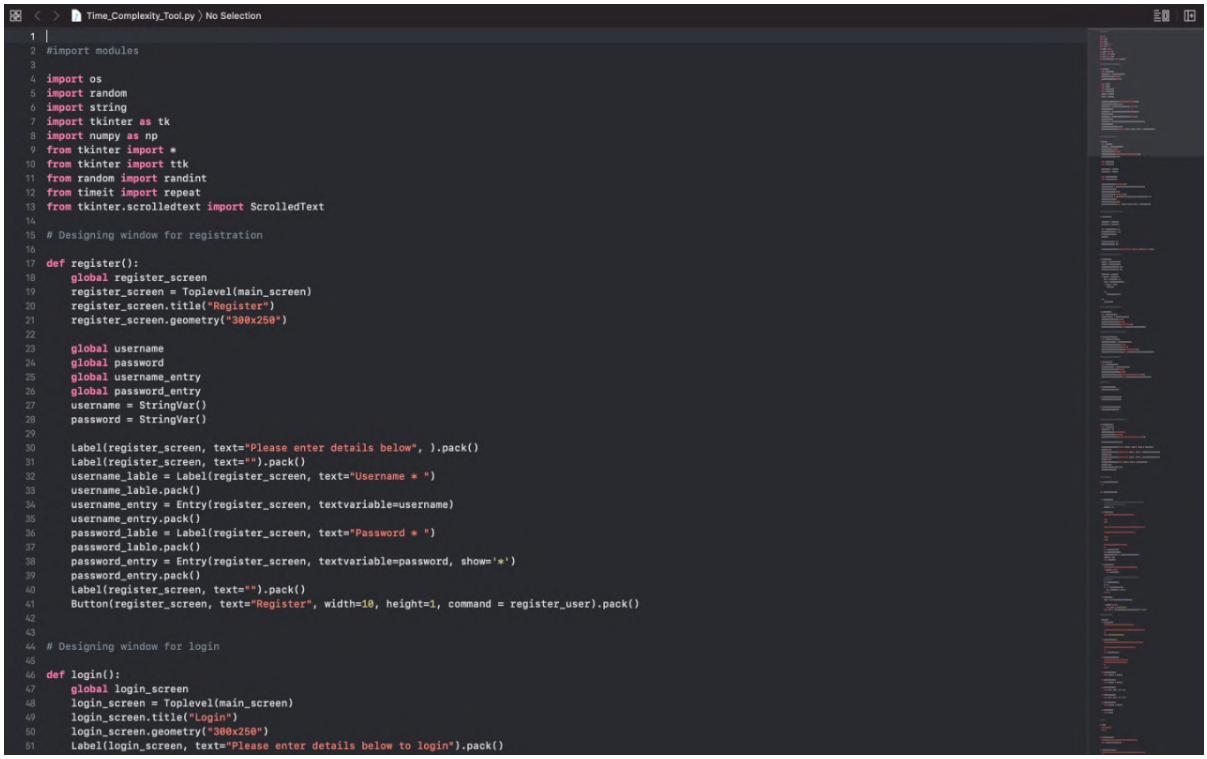
Fig:4.1.8

4.2 CODES AND STANDARDS

The code has been written and compiled using Python language.

Version of Python language being used:

Python IDE has been used for the back-end development of the code.



```
Time_Complexity_Tool.py 〉No Selection

#import modules

import os
import random
import string
import tkinter as tk
import numpy as np
from tkinter import *
from tkinter import ttk
from random import randint
from timeit import repeat
from tkinter.scrolledtext import ScrolledText

# Designing window for registration

def register():
    global register_screen
    register_screen = Toplevel(main_screen)
    register_screen.title("Register")
    register_screen.geometry("300x250")

    global username
    global password
    global username_entry
    global password_entry
    username = StringVar()
    password = StringVar()

    Label(register_screen, text="Please enter details below", ).pack()
    Label(register_screen, text="").pack()
    username_lable = Label(register_screen, text="Username * ")
    username_lable.pack()
    username_entry = Entry(register_screen, textvariable=username)
    username_entry.pack()
    password_lable = Label(register_screen, text="Password * ")
    password_lable.pack()
    password_entry = Entry(register_screen, textvariable=password, show='*')
    password_entry.pack()
    Label(register_screen, text="").pack()
    Button(register_screen, text="Register", width=10, height=1, command = register_user).pack()


# Designing window for login

def login():
    global login_screen
    login_screen = Toplevel(main_screen)
    login_screen.title("Login")
    login_screen.geometry("300x250")
    Label(login_screen, text="Please enter details below to login").pack()
```

Fig:4.2.1

Due to the magnamous size of the code,the github link of the code has been provided below.

https://github.com/kishan1468/Time_complexity_Analyser_SE/tree/main

4.3 CONSTRAINTS ,ALTERNATIVES AND TRADEOFFS

DESIGN 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.

Table of Design Constraints

ID	Origin	*Shall Requirement*
1	*Administrator*	*The system shall be cost effective so that many people can use it.*
2	*Programmer*	The system shall be reliable.
3	*Programmer*	The system shall be portable.
4	*Administrator*	The system shall be ethical.
5	*Administrator*	The system shall follow the law.
6	*Administrator*	The system shall take care of privacy of the user.
7	*Administrator*	The system shall be safe for the user.
8	*Programmer*	The system shall be user friendly.
9	*Programmer*	The system shall be robust.
10	*Programmer*	The system shall be very fast.
11	*Programmer*	The system shall be efficient.

TABLE 4.3.1

PRODUCT CONSTRAINTS

i. Can’t detect syntactically erroneous code.

ii. Only one user supported at a time.

iii. Limited to a fixed set of languages.

iv. Can only be launched by Python Ide launcher.

ALTERNATIVES

Well there is no alternative associated with the functioning of the software.

TRADEOFFS

JAVA IDE and Visual Studios were the former choice for the development of code and back end .Although Python IDE and XCODE proved to be more fruitful and user friendly.Hence they were used for the development process.

5. SCHEDULE TASKS AND MILESTONES

The project nearly took roundabout 4 months for its completion.Taking the scheduled date into account tht project has been satisfactorily completed.

Here are figures depicting the duration cycle of the project.

GANTT CHART

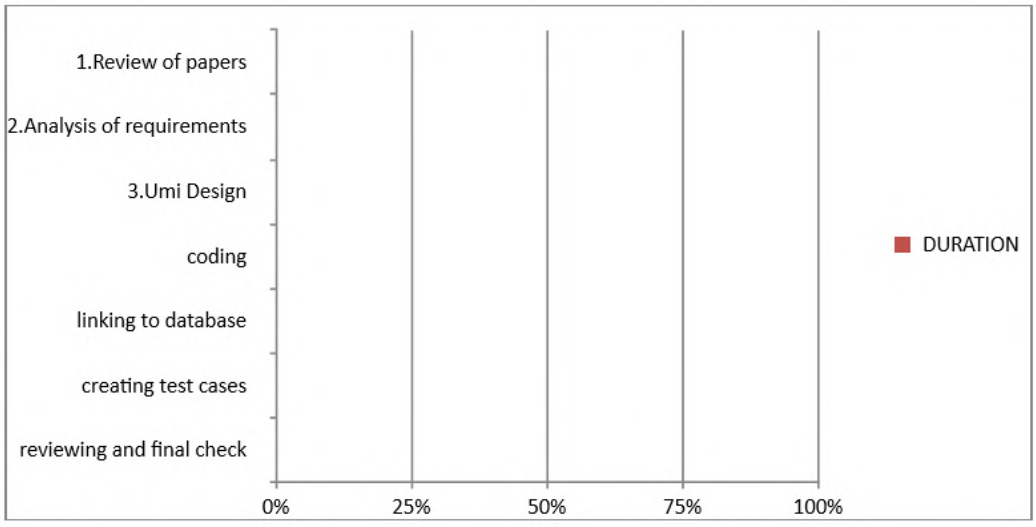
GANTT CHART :-

Sl.No	Task	Feb	Mar			Apr			May
		01-30	1-10	10-20	20-30	1-10	10-20	20-29	1-20
1.	Project title selection and analysis								
2.	Architecture and paper review								
3.	Developing the tool								
4.	Implementation								
5.	Collecting the Data								
6.	Data Analysis and testing								
7.	Demonstration and Paper draft								

Tool Used: **Pages**

TABLE 5.1

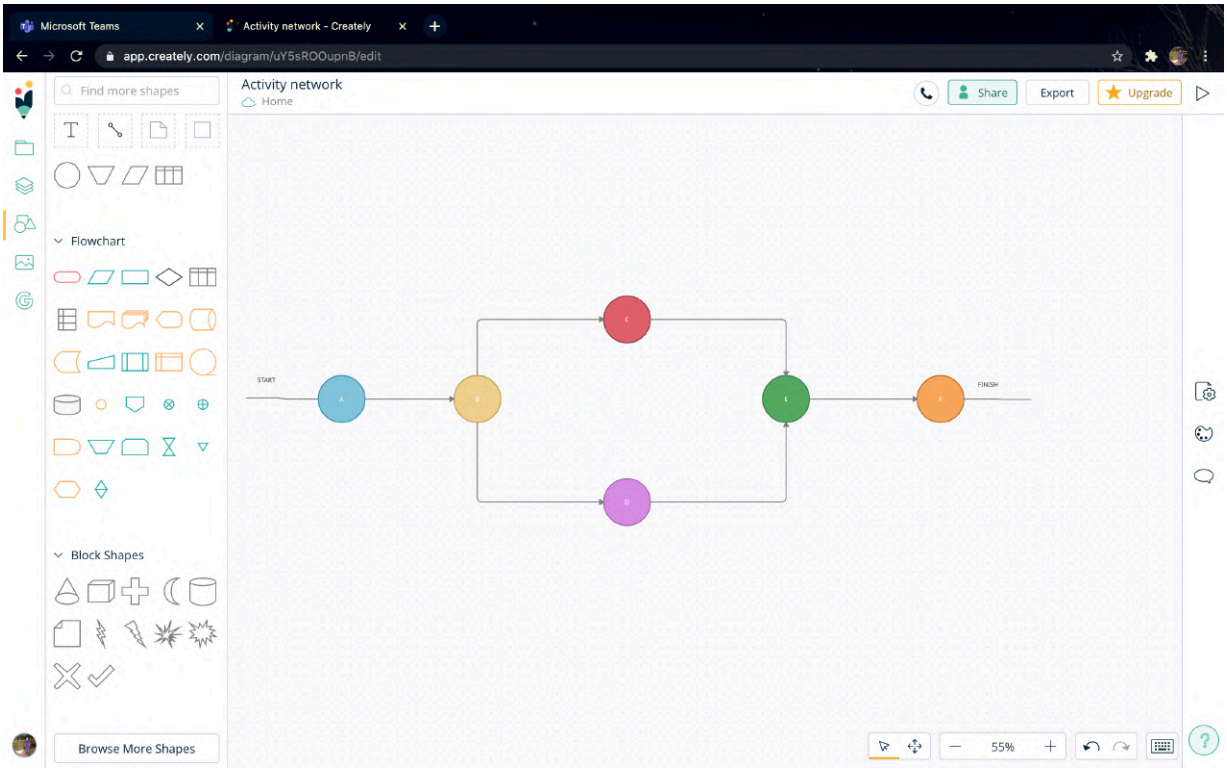
TIMELINE CHART



-

TABLE 5.2

ACTIVITY NETWORK DIAGRAM



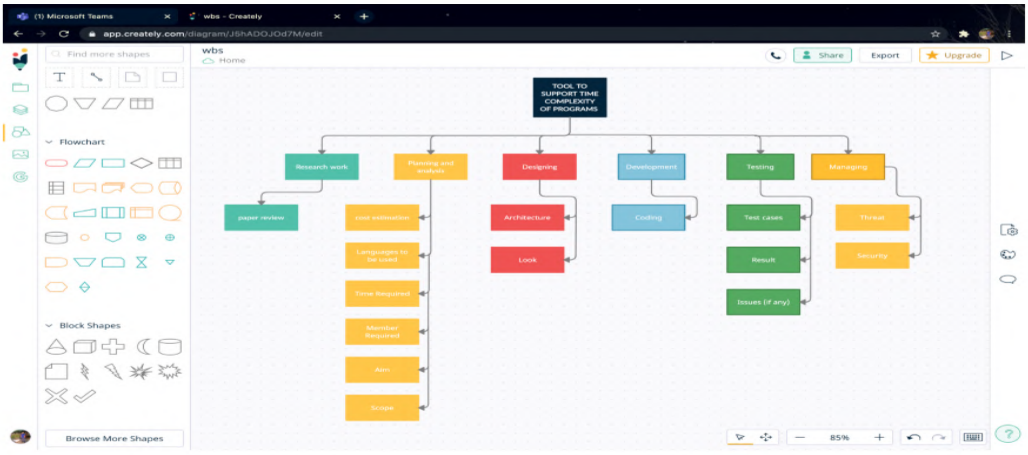
TASK	LABEL	PREDECESSOR	MEMBER REQUIRED	ESTIMATED DURATION
Planning and analysis	A		2	15 days
Designing	B	A	1	7 days
Coding	C	A,B	1	14 days
Adding Data	D	A,B	1	5 days
Test cases	E	A,B,C,D	1	7 days
Demonstration	F	E	2	3 days

TABLE 5.3

TASKS

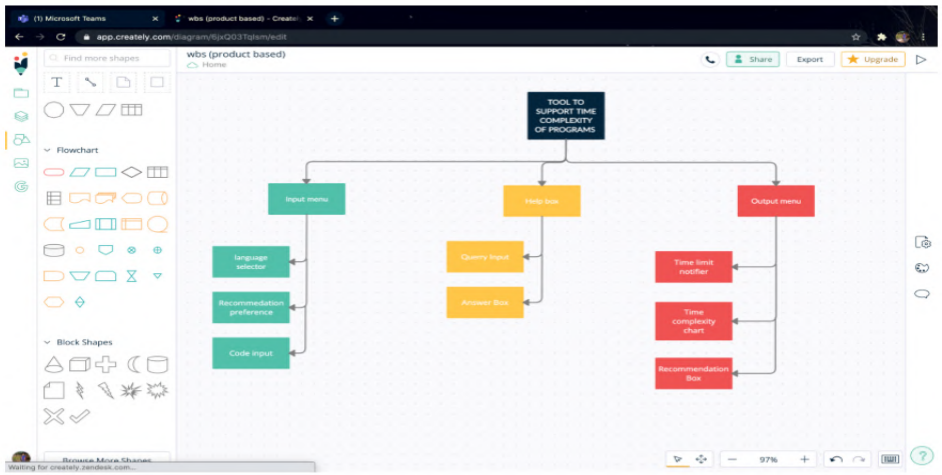
The tasks for both the process structure and product structure have been potrayed through work break-down structures.

Work Breakdown Structure: (process model)



Tool Used: **creately**

Work Breakdown Structure: (Product model)



Tool Used: **creately**

Fig:5.1

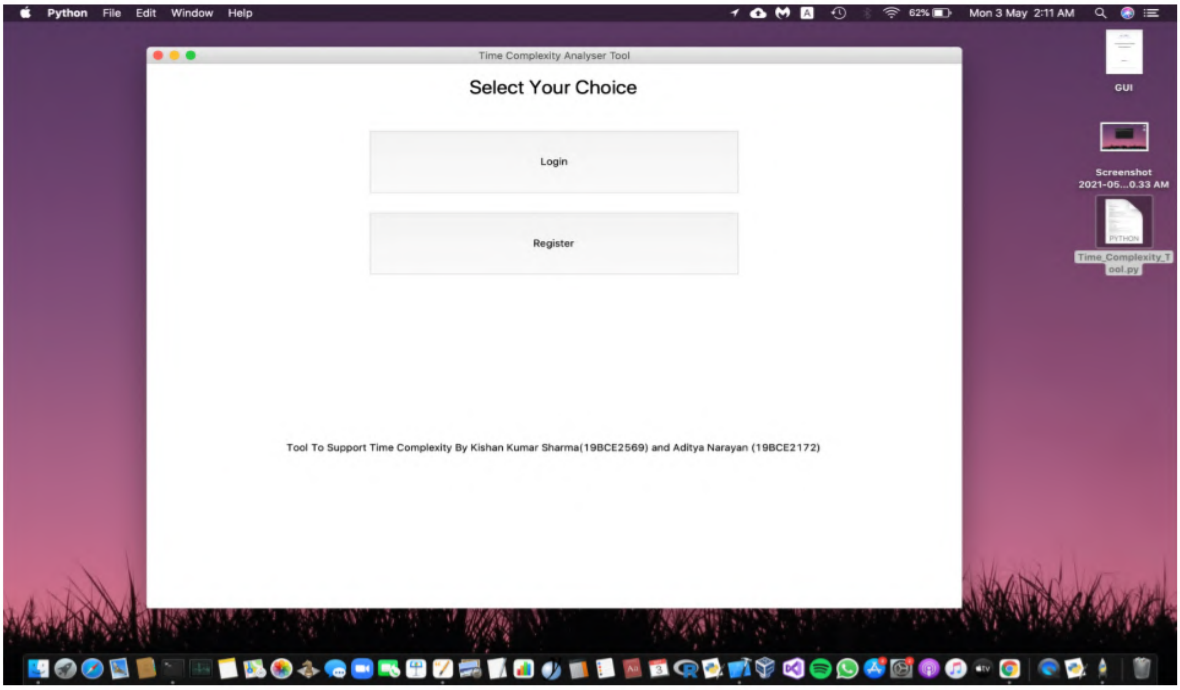
6 . PROJECT DEMONSTRATION

The code has been written in Python Ide. Xcode and Github have been used for the development of the project.It has nearly took 3 weeks for the development of the software and code compilation.

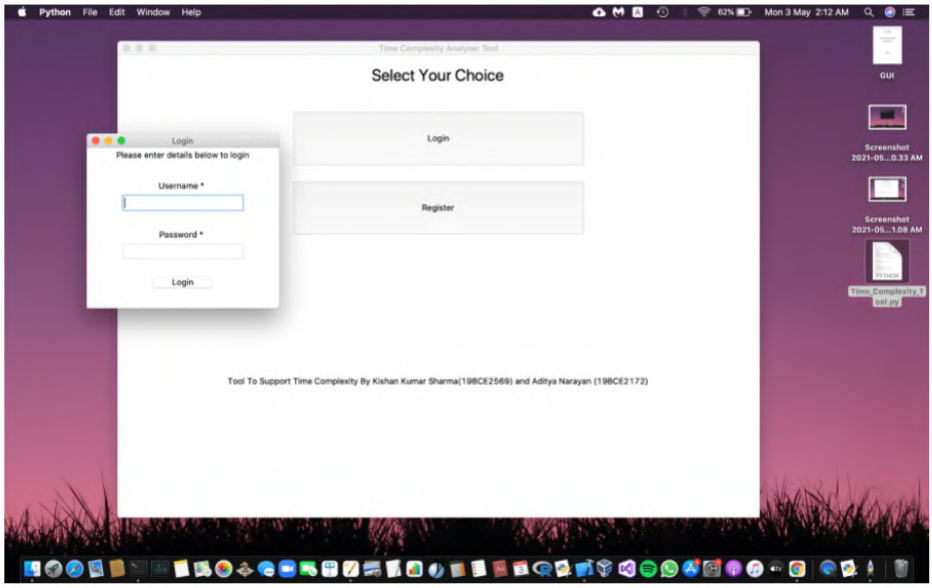
The demonstration has been made through these screenshots taken during the implementation of the software.

1. Calling the python file from terminal.

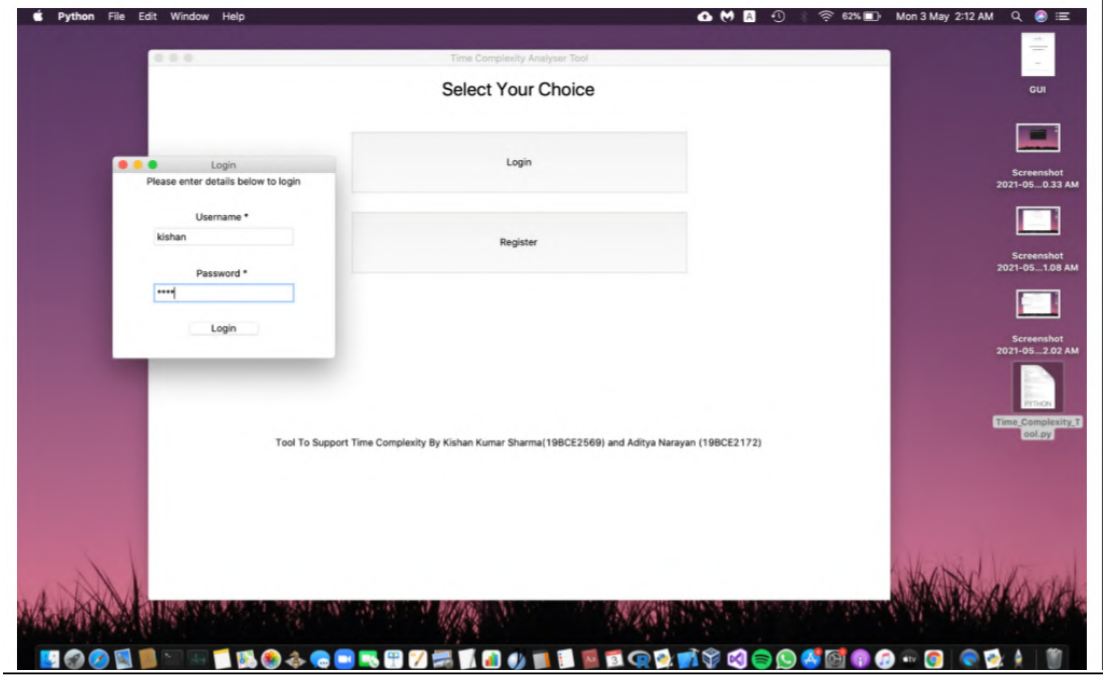
2. After successfully calling the main python file, we get into the login/ register page where user has the option to login or register as per his requirement. Here we have two buttons:
- Login- if user already has an account.
- Register- if user don't have an account but wanted to make one.



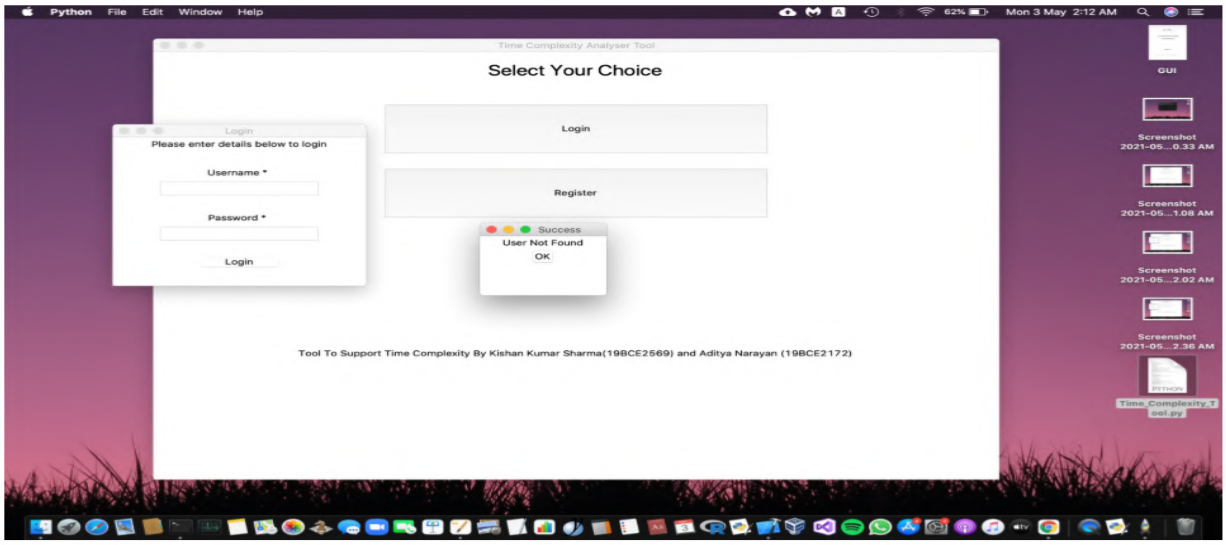
3. After clicking on login button we get to the Login page where the software ask the user to enter the login credentials to login into the system.



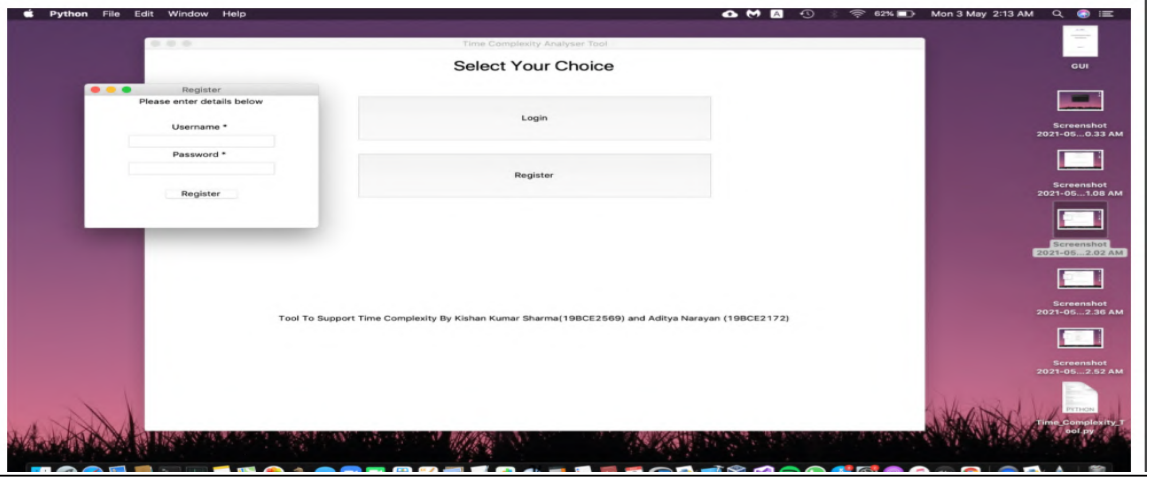
4. After entering the credentials the software will verify if the credentials are correct or not. If correct then it allows the user to login and displays “login success” otherwise ask user to register and display “user not found”.



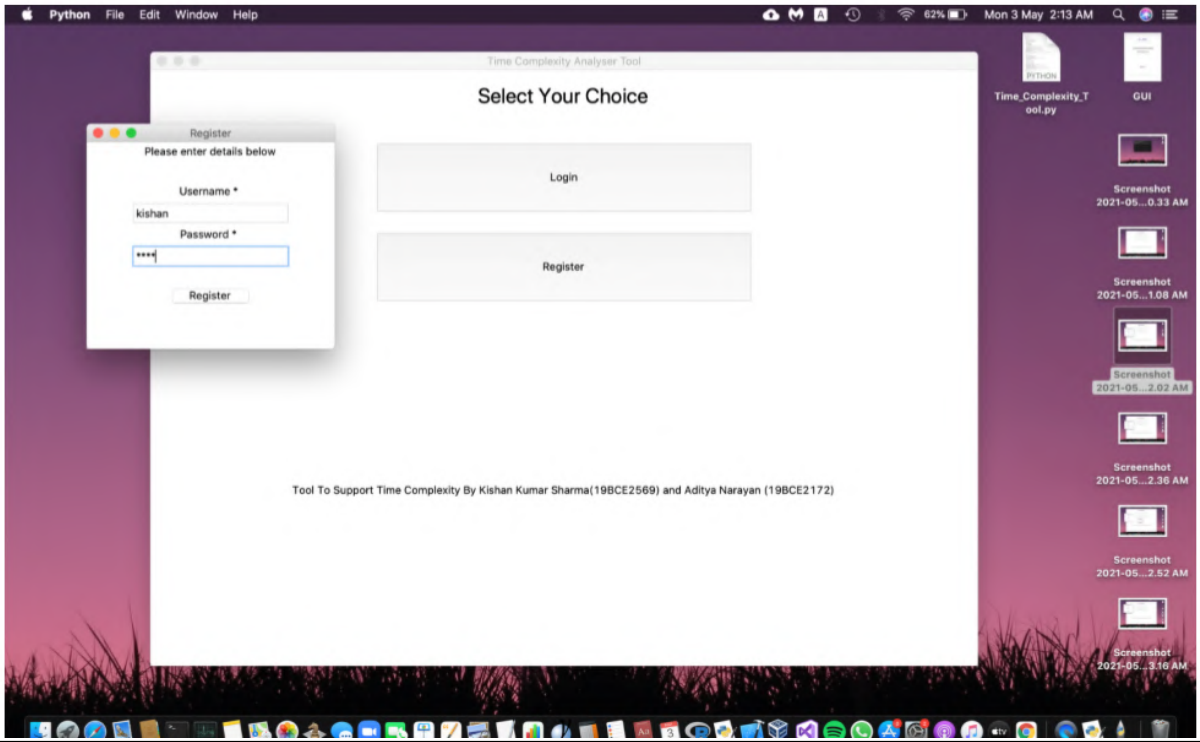
5. Like expected, it displays “user not found” as we don’t have an account. So we need to register and make an account to login into the system.



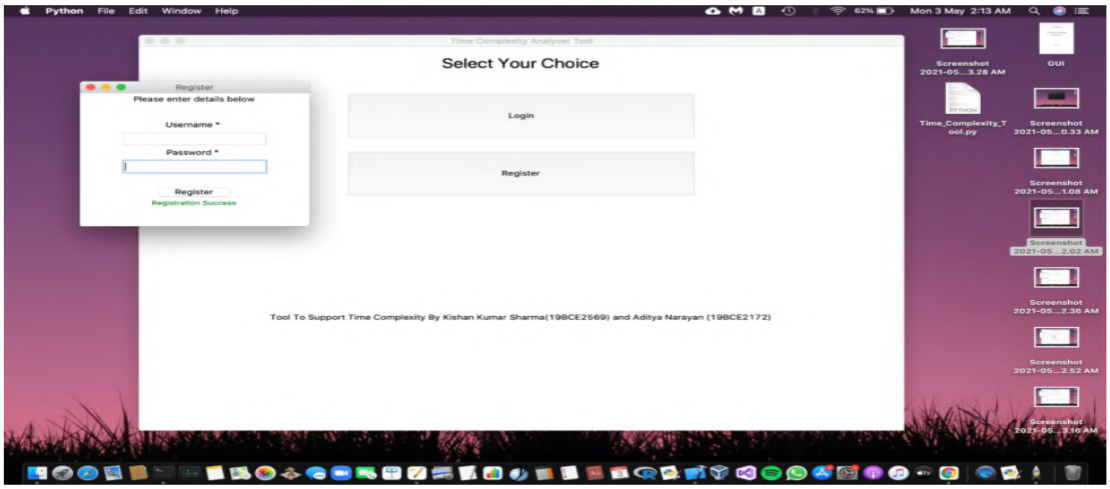
6. After clicking register button, we get to the Register page where we need to enter our username and password to register an account.



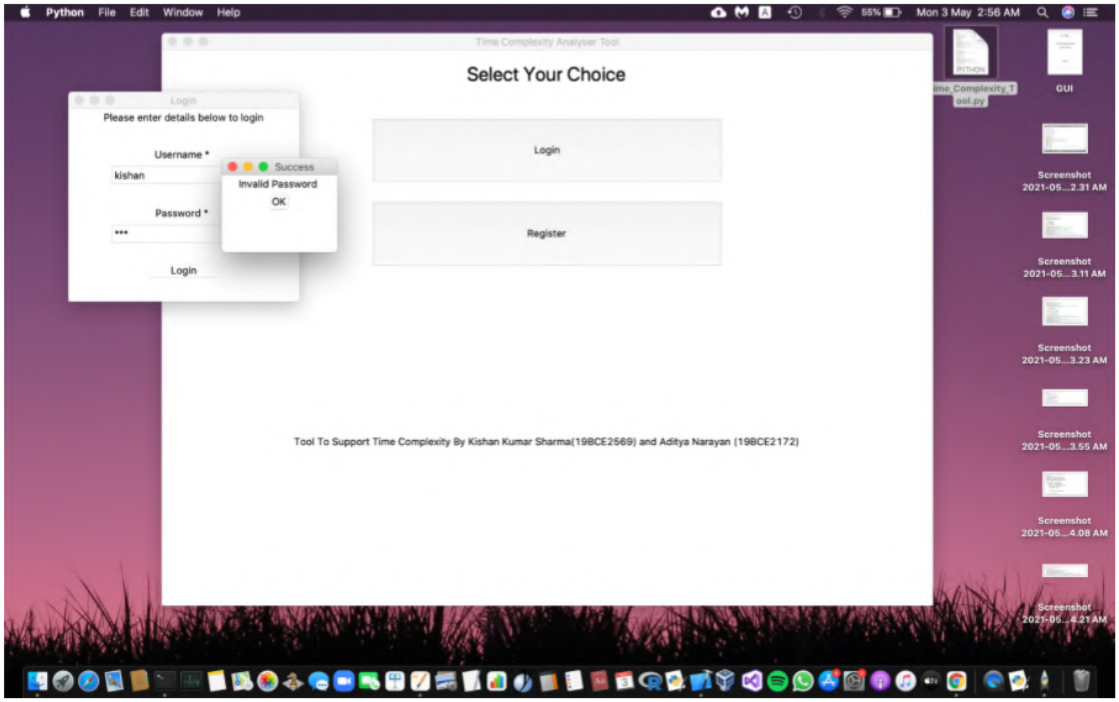
7. Here I have entered my username as “kishan” and my password as “1234” to register an account.



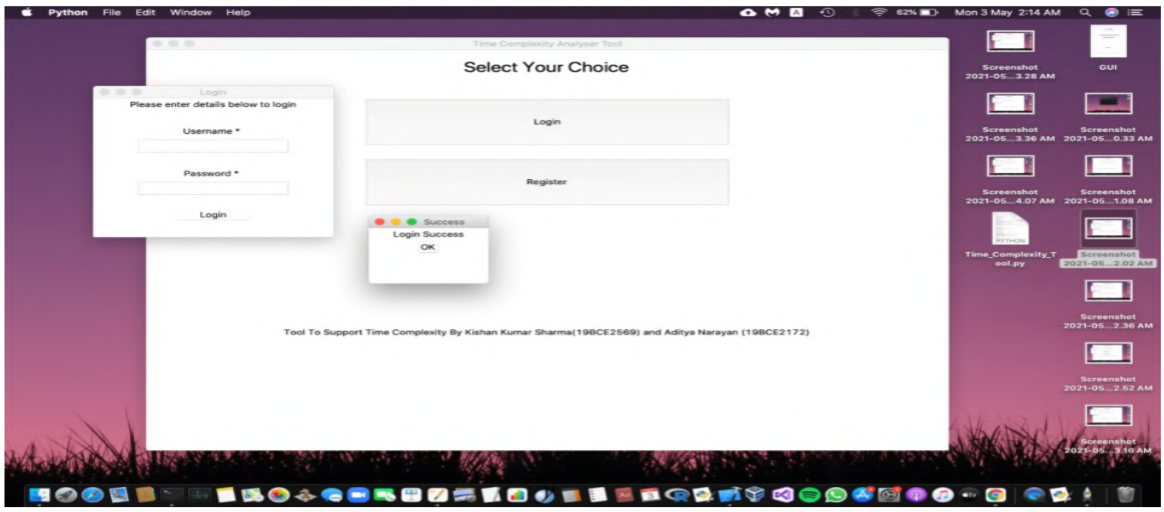
8. As expected, the register page is displaying that " Registration Success". Now we will be using the same username and password to login into the system.



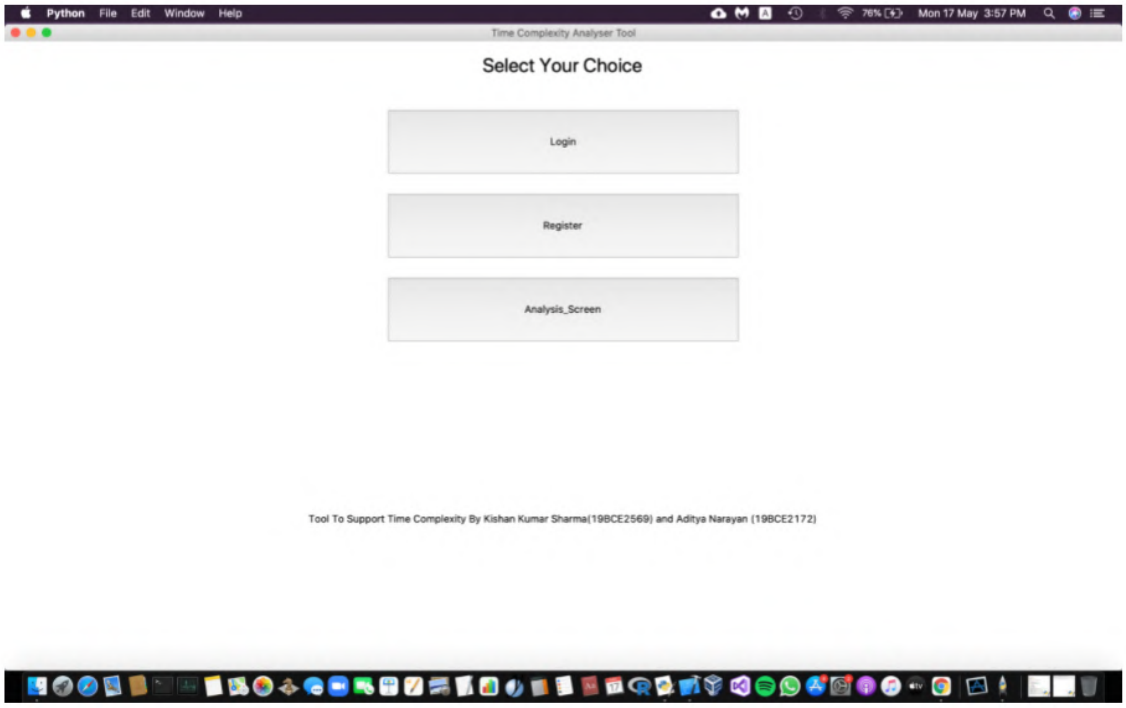
9. Now again after clicking login button, we get to the login page where I have enter my username as "kishan" and my password as "123" which is incorrect password as my password is "1234". So as expected the system is showing "invalid password".



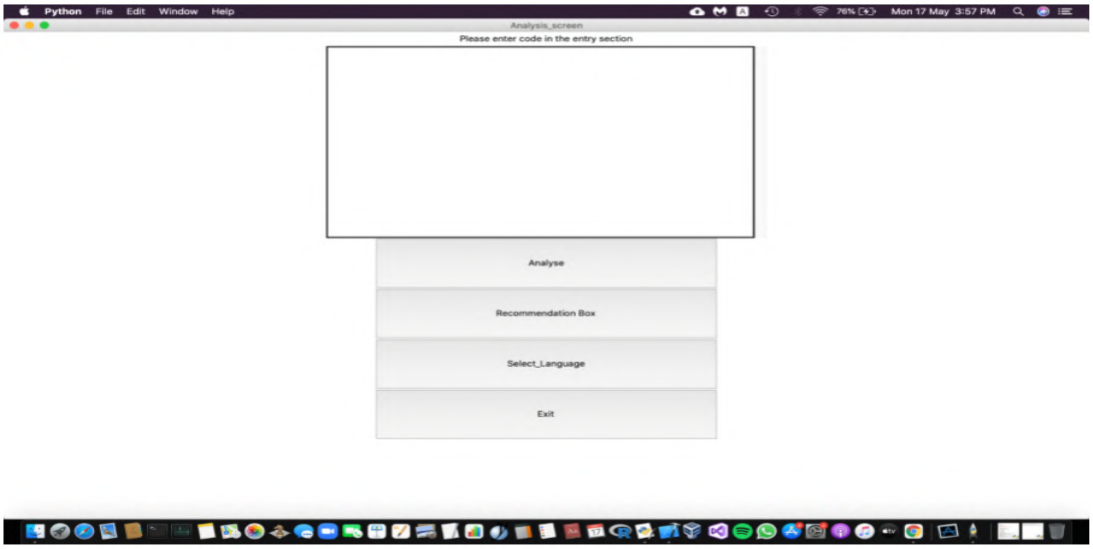
11. As expected, after entering the correct username and password, the system is showing “Login Success”. So now we have successfully entered the system.



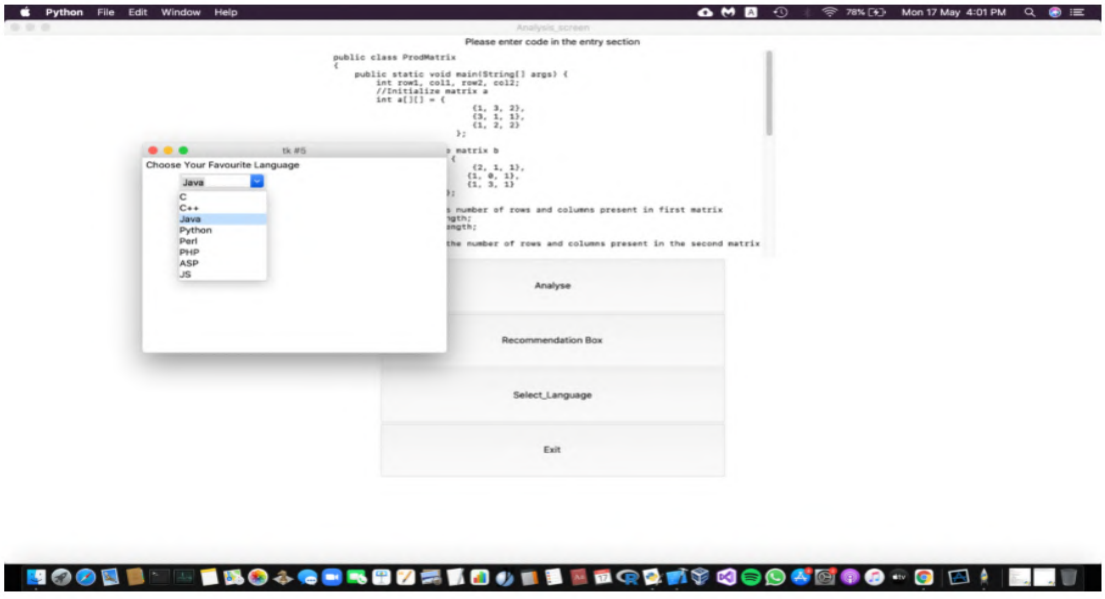
12. Now we can click on analysis_screen button to go to the code analysis page.



13. After clicking on code analysis, as expected the system has moved to the analysis screen. In analysis screen there are 4 buttons given such as analyse button to analyse the complexity of code, recommendation box for recommendation, select language to select the programming language, and exit button to exit the software. Also a code entry section box is designed to take the input from user.

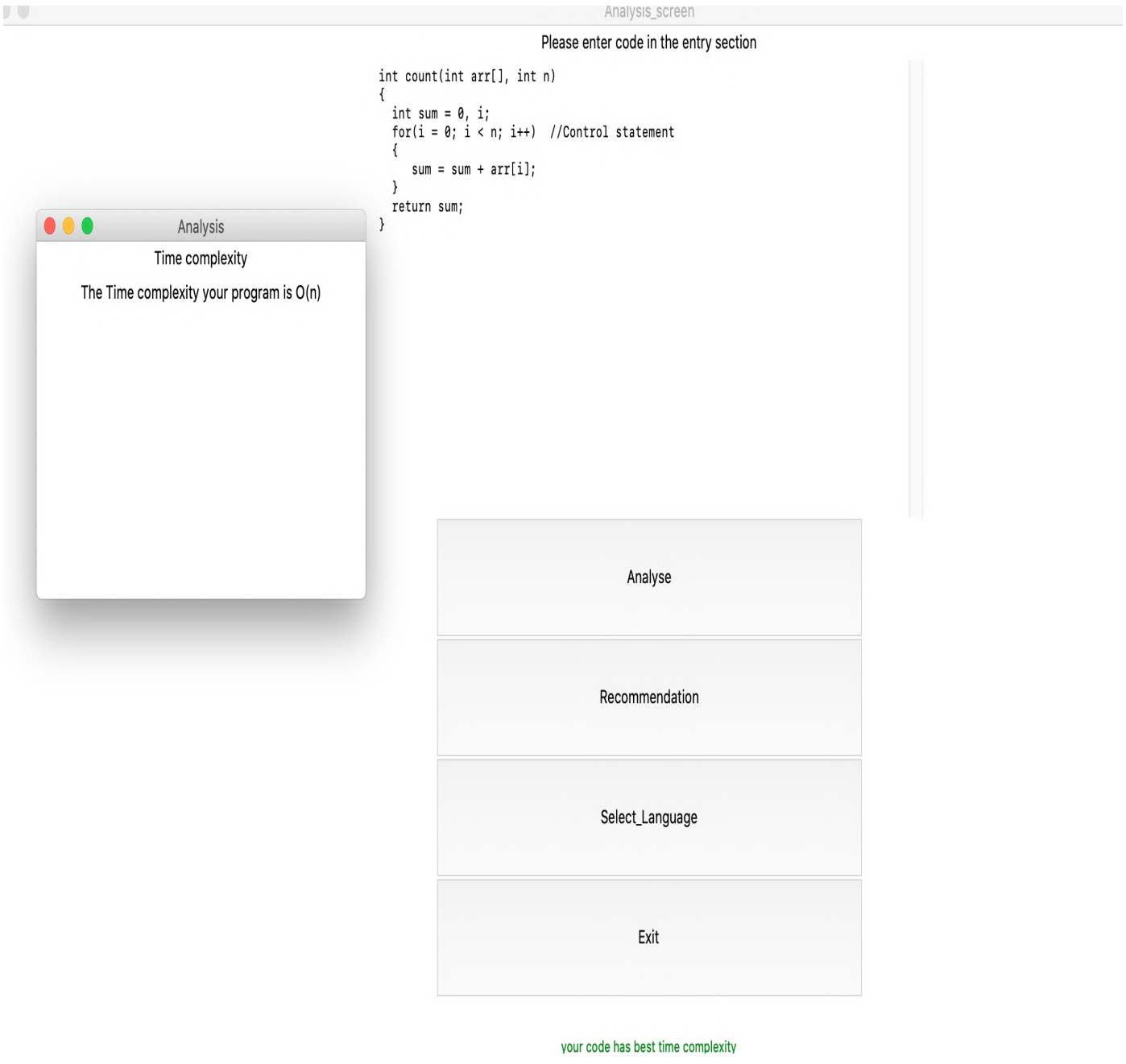


14. As expected we can choose our language from menu box and write or copy paste the codes in the code entry section.



15.The code is passed into the code entry section and the analyse button is clicked.

The result is displayed alongside .



16.Another set of code is passed by the user and the anlyse button is clicked along with recommendation button.The time complexity is displayed alongside,with the case analysis of code diplayed right below

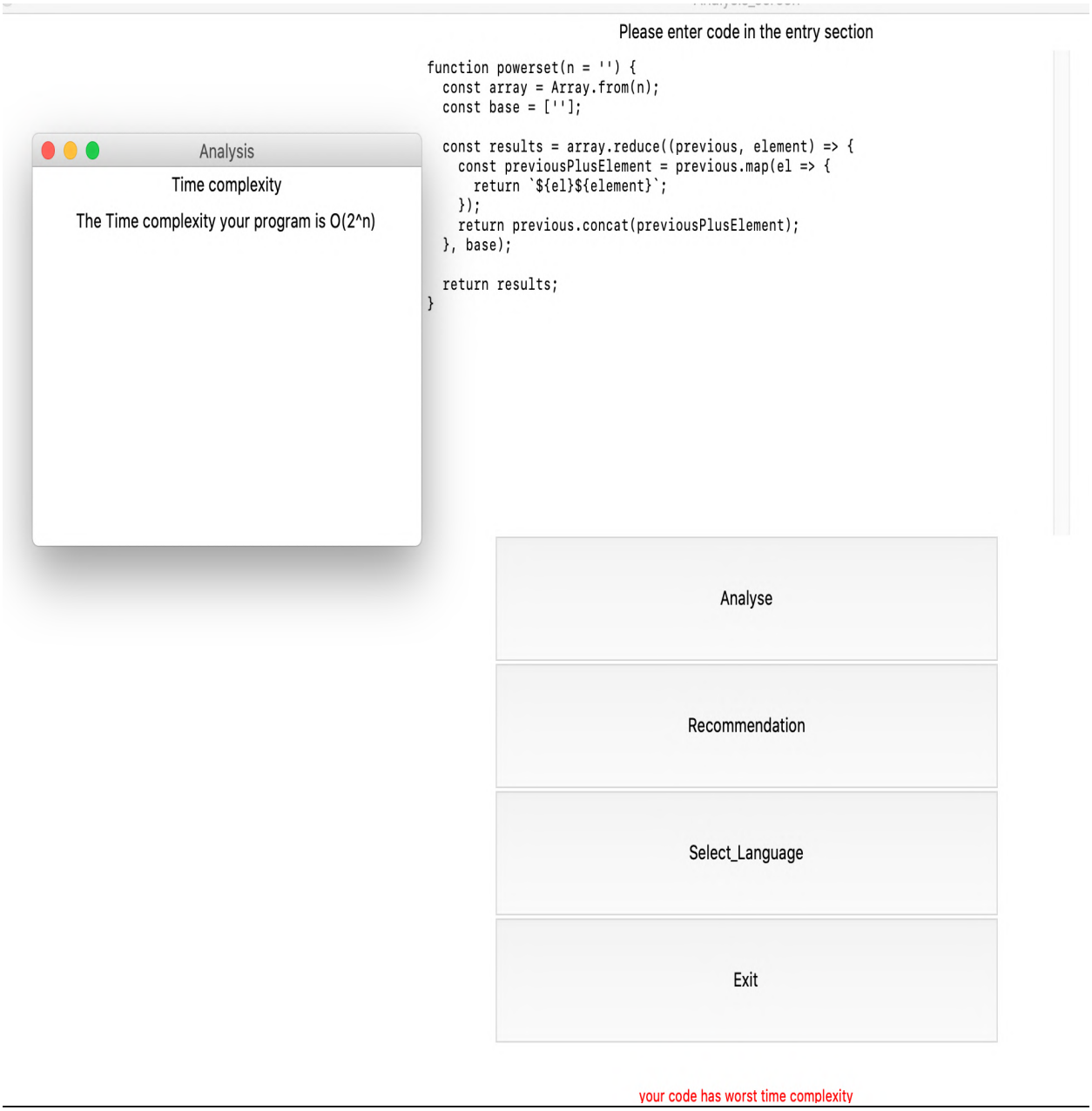


Fig 6.1

7. RESULTS AND DISCUSSION/COST ANALYSIS

After the event of final execution the software was passed through a number of experimental test cases.The results have been displayed below in tabular form.

1. App Launch Testing:

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1	Does the app launch	App called from terminal	App will be launched on main screen	App has been launched successfully	Pass
2	Does login button working properly	"Login" button clicked	Login page will be launched	Login page has been launched successfully	Pass
3	Does Register button work properly	"Register" button clicked	Register page will be launched	Register page has been launched successfully	Pass
4	Does Analysis screen button working properly	"Analysis button" selected	Analysis screen will be launched	Analysis screen has been launched successfully	Pass
5	Does Analysis button working properly	int count(int arr[], int n) { int sum = 0, i; for(i = 0; i < n; i++) // Control statement { sum = sum + arr[i]; } return sum; }	O(n) should be displayed successfully	O(n) should be displayed successfully	Pass
6	Does Recommendation button working properly	Recommendation button selected	Shows the recommendation section succesfully	Shows the recommendation successfully	Pass
7	Does select language button working properly	"C" language selected	Language selected succesfully	Language selected successfully	Pass

8	Does exit button working properly	"Exit" button clicked	Exits the app succesfully	Exits the app successfully	Pass
9	Is user able to copy/paste or write the code in entry section	Class Aditya Public static void main()	Code displayed/ copied on screen succesfully	Code displayed / copied on screen succesfully	Pass
10	Does the app has correct label and texts	Recommendation Label checked	Label Found Correct	Label Found Correct	Pass

TABLE 7.1

2. Register module testing:

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1	Can user enter his username	ram	Username should be entered	Username has been entered	Pass
2	Can user enter his password	12345	Password should be entered	Password has been entered	Pass
3	Is register button working properly	Ram 12345	"Registration Succesfull"	"Registration Succesfull"	Pass

3. Login module testing:

Test Case ID	Test Objective	Test Data	Expected Results	Actual Results	Test Pass/Fail
1	Can user enter his username	ram	Username entered	Username entered	Pass
2	Can user enter his password	12345	Password Entered	Password Entered	Pass
3	Is login button working properly	Ram 12345	Login Succesfull	Login Succesfull	Pass

COST ANALYSIS

The monetary expense regarding the development of this software has been negligible.

The development process required installation of few softwares such as XCODE,PYTHON IDE, TKINTER etc.

The software development took strenous human workforce of two members regarding its planning, development and execution.

The total time taken in the whole process was roundabout four months.

CONCLUSORY OBSERVATIONS

The software posses numerous scope of improvement and upgradations.For instance the present form of interface seems a bit less vivid.Moreover the platform can allow multi users to work simultaneously at a single instance of time.

During the execution process observed glitches when manamous junks of code with highly complex iterations where passed.In such cases the system found it difficult to fully analyse the recommendational section of the code.

THANK YOU!