P.S.R. ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

Approved by AICTE, New Delhi & Accredited by National Board of Accreditation (NBA)

Accredited by NAAC with "A+" Grade, Recognized under 12(B) of the UGC Act, 1956.

An ISO 9001:2015 Certified Institution

Sevalpatti (P.O), Sivakasi - 626140. Tamilnadu.

Department of Computer Science and Engineering

LABORATORY RECORD

191CS45 – SOFTWARE ENGINEERING

IV – SEMESTER

Department of Computer Science and Engineering

P.S.R. ENGINEERING COLLEGE

Sivakasi - 626140

P.S.R. ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)
Approved by AICTE, New Delhi & Accredited by National Board of Accreditation (NBA)
Accredited by NAAC with "A+" Grade, Recognized under 12(B) of the UGC Act, 1956.
An ISO 9001:2015 Certified Institution
Sevalpatti (P.O), Sivakasi - 626140. Tamilnadu.

Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

Certified that this is a Bonafide Record of work done

*by*____

Roll No ________________ *in the*____

Laboratory of this College during the academic year 2022-2023.

…………………………… ……………………………………

Staff-in-Charge *Head of the Department*

Register No:

Submitted for the Practical Examination held on………………………………………………………

………………………… ……………………………………

Internal Examiner *External Examiner*

TABLE OF CONTENTS

EXAM REGISTRATION SYSTEM

AIM:

To create an automated system to perform the Exam Registration Process

PROBLEM STATEMENT:

Exam Registration system.is used in the effective dispatch of registration form to all of the students. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, reg.no etc.,) filled by the student whose testament is verified for its genuineness by the Exam Registration System with respect to the already existing information in the database. This forms the first and foremost step in the processing of exam application. After the first round of verification done by the system, the information is in turn forwarded to the Exam Controller. The application is then processed manually based on the report given by the system. The system also provides the student the list of exam dates. The controller will be provided with fees details to display the current status of application to the student, which they can view in their online interface. After all the necessary criteria has been met, the original information is added to the database and the hall ticket is sent to the student. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. So this system uses several programming and database techniques to elucidate the work involved in this process

1. SOFTWARE REQUIREMENT SPECIFICATION:

1.1 INTRODUCTION

Exam Registration System is an interface between the Student and the Exam Controller responsible for the Issue of Hall Ticket. It aims at improving the efficiency in the Issue of Hall ticket and reduces the complexities involved in it to the maximum possible extent.

1.2 PURPOSE AND SCOPE

If the entire process of 'Issue of Hall ticket' is done in a manual manner then it would takes several days for the hall ticket to reach the student. Considering the fact that the number of students for hall ticket is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

SCOPE

- The System provides an online interface to the user where they can fill in their person details and submit the necessary documents (may be by scanning).
- The controller concerned with the issue of hall ticket can use this system to reduce his workload and process the application in a speedy manner.
- Provide a communication platform between the student and the controller.

1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

• **Exam Controller** - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entire system.

• **Student** - One who wishes to obtain the Hall Ticket

• ERS - Refers to this Examination Registration System

• **Python -** Refers to software to execute the code.

•XAMPP Server- Refers to software to manage the database.

1.4 TOOLS TO BE USED

- PYTHON IDLE
- STARUML

SRS includes two sections overall description and specific requirements **Overall Description** will describe major role of the system components and inter-connections. **Specific Requirements** will describe roles & functions of the actors. Considering the fact that the number of students for hall ticket is increasing every year, an Automated System becomes essential to meet the demand. It aims at improving the efficiency in the Issue of Hall ticket and reduces the complexities involved in it to the maximum possible extent.

2.OVERALL DESCRIPTION

The ERS acts as an interface between the 'student' and the 'exam controller'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the hall ticket.

2.1 SOFTWARE INTERFACE

• **Front End Client** – The Applicant and Administrator interface is built by using python.

• **Back En**d – XAMPP databases.

2.2 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

2.3 SYSTEM FUNCTIONS

• Secure Registration of information by the Students.

• SMS and Mail updates to the students by the controller.

• Controller can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

• Student - They are the people who desire to obtain the hall ticket and submit the information to the database.

• Exam controller - He has the certain privileges to add the registration status and to approve the issue of hall ticket. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of hall ticket.

2.4 CONSTRAINTS

• The applicants require a computer to submit their information.

• Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

• The user has to be careful while submitting the information. Much care is required.

• The Students and Exam Controller must have basic knowledge of computers and English Language.

• The student may be required to scan the documents and send.

• The student my be required to the any proof are send.

• The student has very secured to the proof was identically for carefully send.

• The Exam center has been confirmed to approved to the correct result.

<u>3.UML DIAGRAM:</u>

<u>3.1 USECASE DIAGRAM:</u>

The Exam Registration use cases in our system are:

1. Login
2. View exam details
3. Register
4. Acknowledgement
5. Fee Processing

<u>ACTORS INVOLVED:</u>

1. Student
2. System DB

<u>USE-CASE NAME: LOGIN</u>

The student enters his username and password to login and retrieve the information'.

<u>USE-CASE NAMEVIEW EXAM DETAILS</u>

The student view the details about the exam schedule which contains Date time etc...

<u>USE-CASE NAME:REGISTER</u>

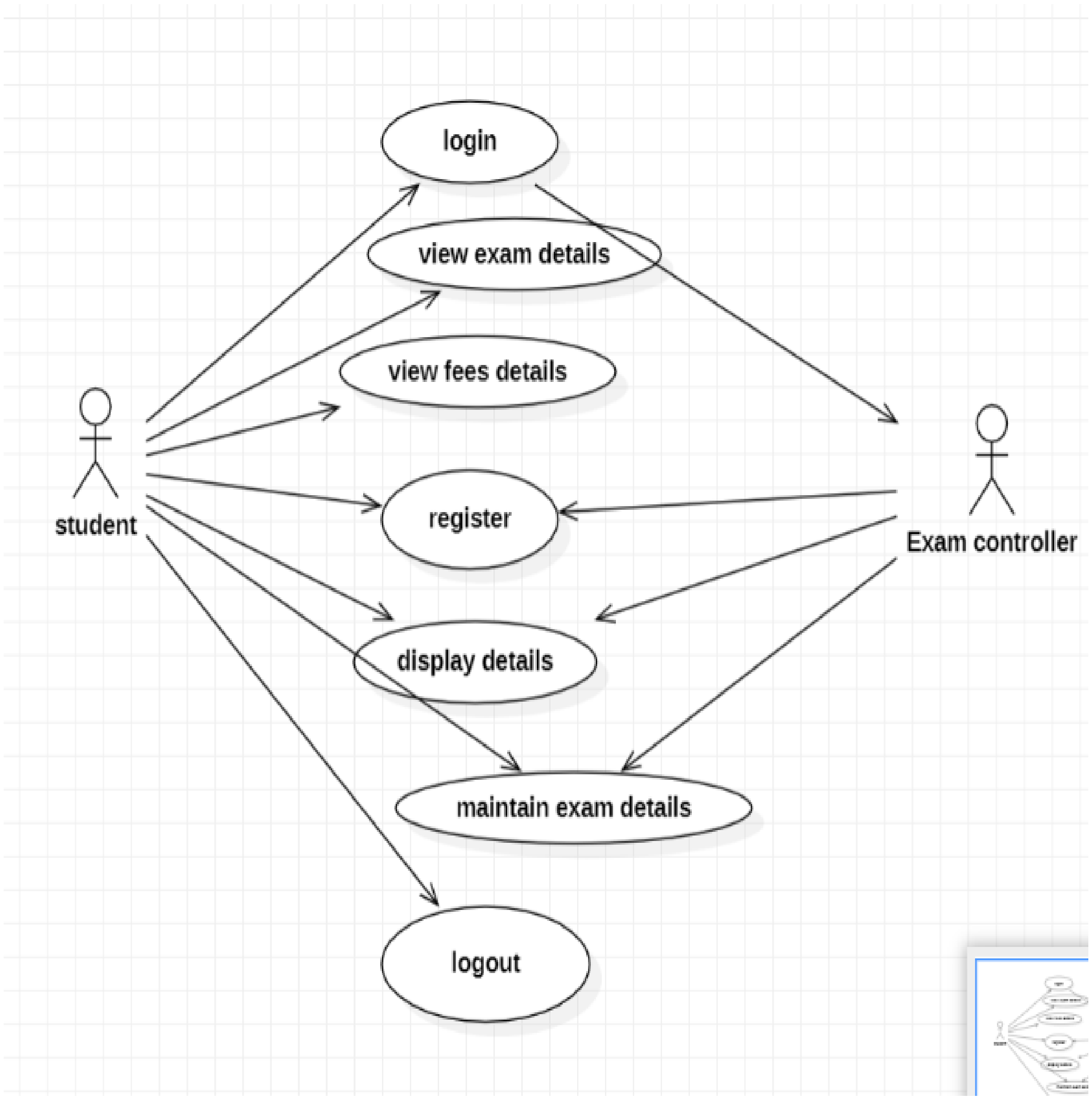
The student should notify the fee details that only the student can pay the correct amount

<u>USE CASE NAME:ACKNOWLEDGEMENT</u>

The exam fees should be paid by the student to get the hall ticket from the exam controller.

<u>USE CASE NAME FEE PROCESSING</u>

All the details should be viewed by both the student and the controller to verify whether all the entered details are correct



USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

3.2 CLASS DIAGRAM :

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects. The Exam Registration System class diagram consists of four two classes of registration system.

1. Student_details

2. Exam_details

3. Register

1) STUDENT_DETAILS

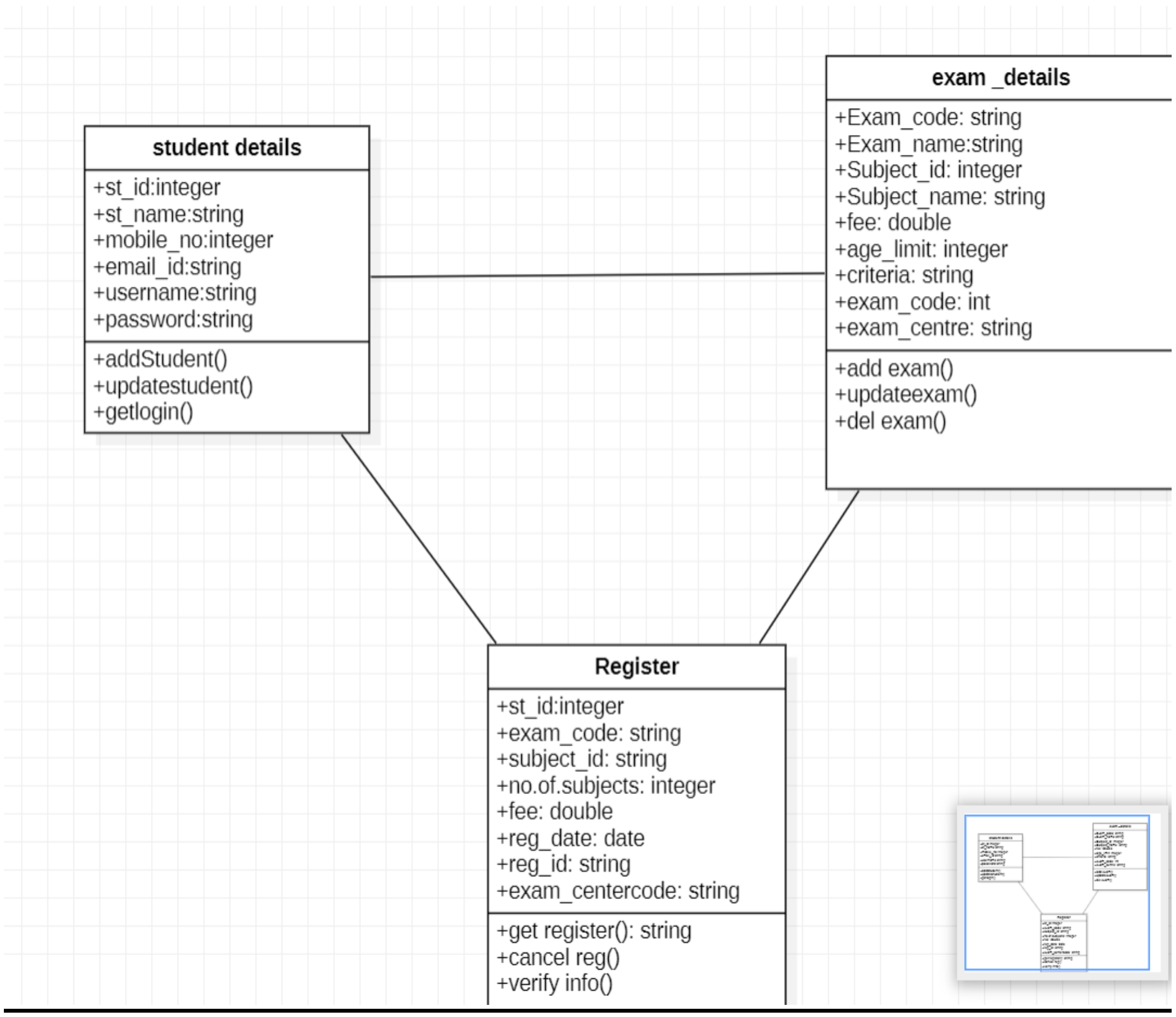
It consists of six attributes and six operations. The attributes id, password, name, age, sex, course. The operations of this class are login(), logout(), conformation(),register(), new fees details().

2) EXAM_DETAILS

It consists of four attributes and six methods. The attributes are user id, password, Exam fees, fees due. The methods are login(),logout(), fees details(), display fees(),conformation(), exam controller().

3) REGISTER

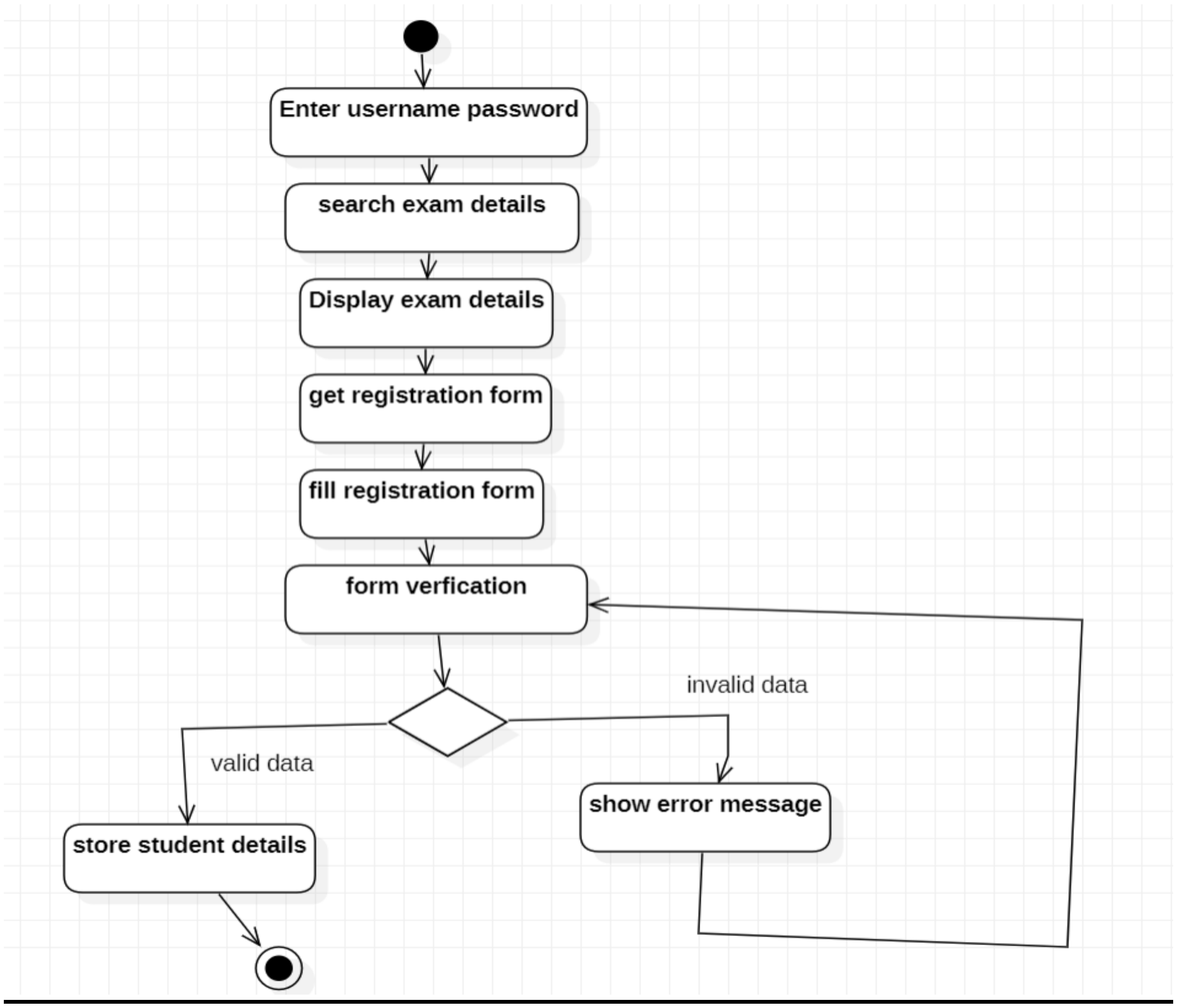
This class is used to maintain the registered student information such as, subject registered, date of registration and etc,.



CLASS DIAGRAM FOR EXAM REGISTRATION SYSTEM

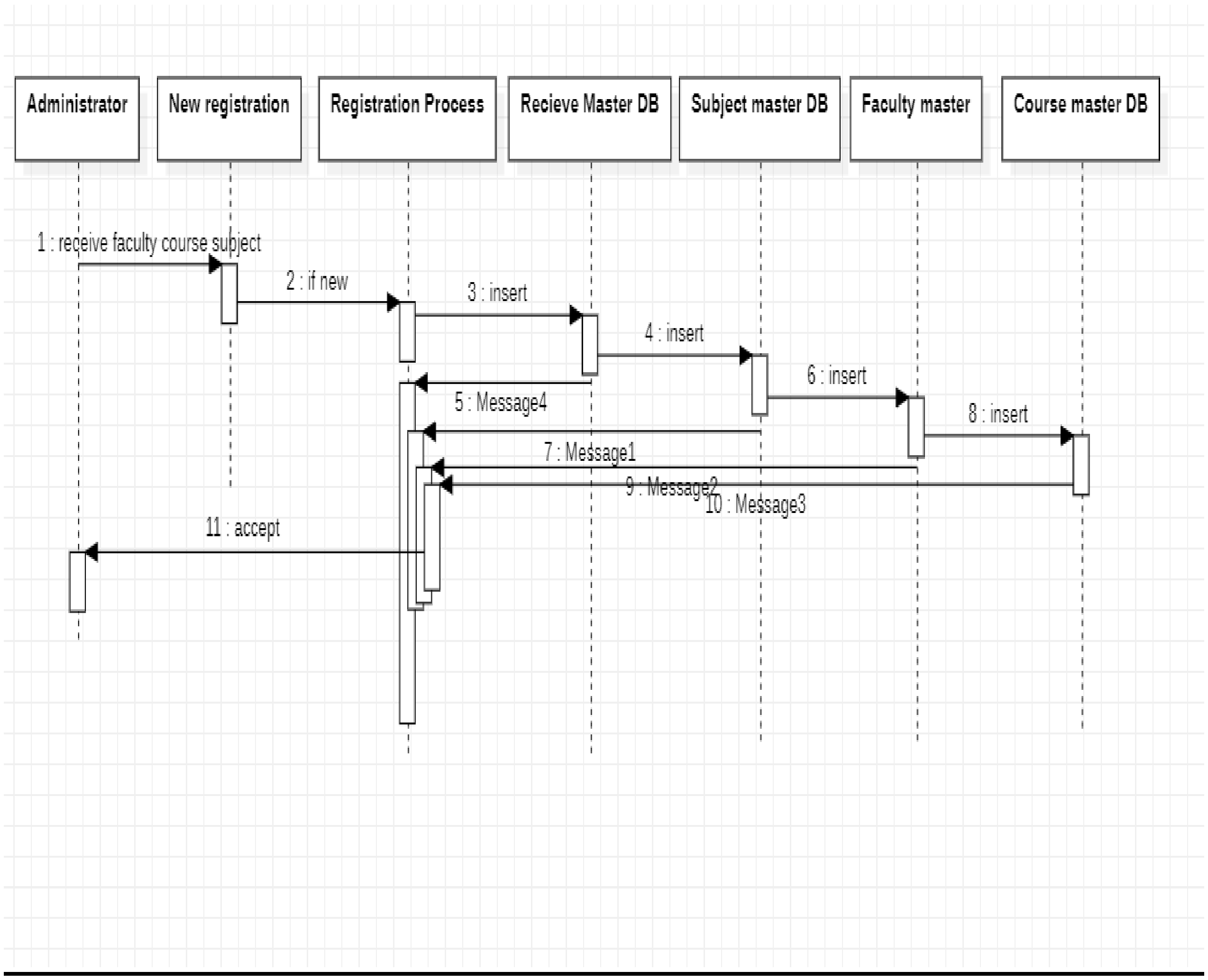
3.3 ACTIVITY DIAGRAM:

The activity diagram represents the series of activities that are occurring between the Object. Following is activity diagram which represents the Software personnel management system process.



3.4 SEQUENCE DIAGRAM:

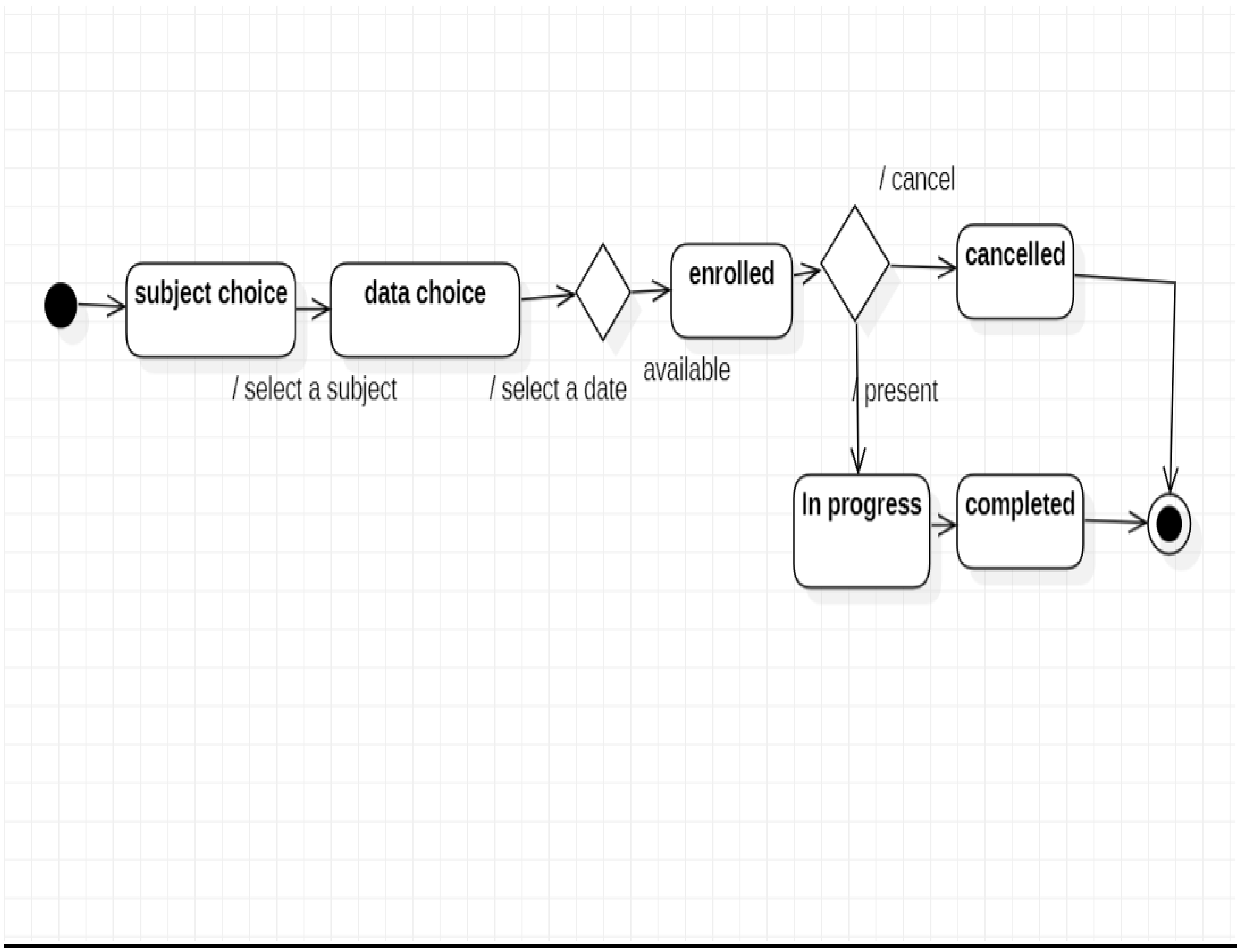
A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices. An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the “form” object is requesting an operation be performed by the “to” object. The “to” object performs the operation using a method that the class contains. It is also represented by the order in which things occur and how the objects in the system send message to one another



SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

3.5 STATECHART DIAGRAM:

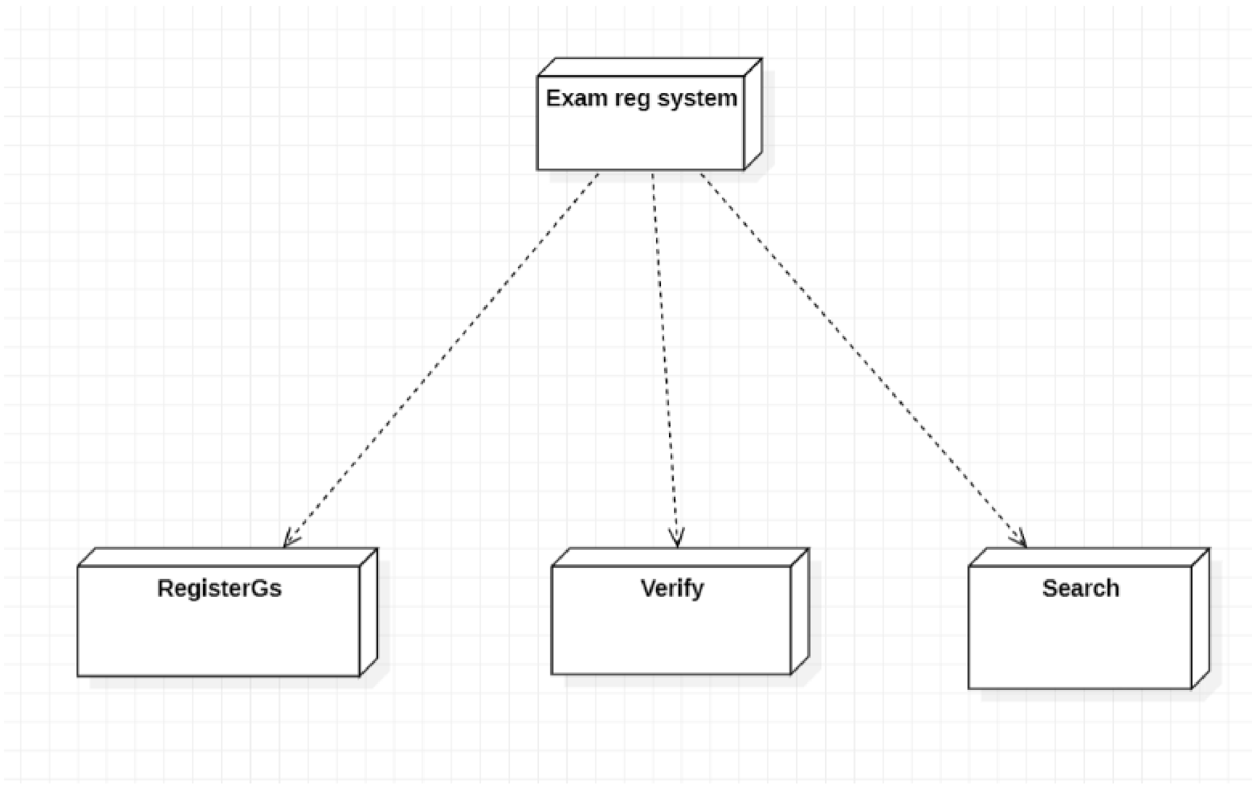
• Every object undergoes through some state and on receiving some event the state gets changed. This transition of the state can be represented by the state transition diagram.



STATECHART DIAGRAM FOR EXAM REGISTRATION SYSTEM

3.6 DEPLOYMENT DIAGRAM:

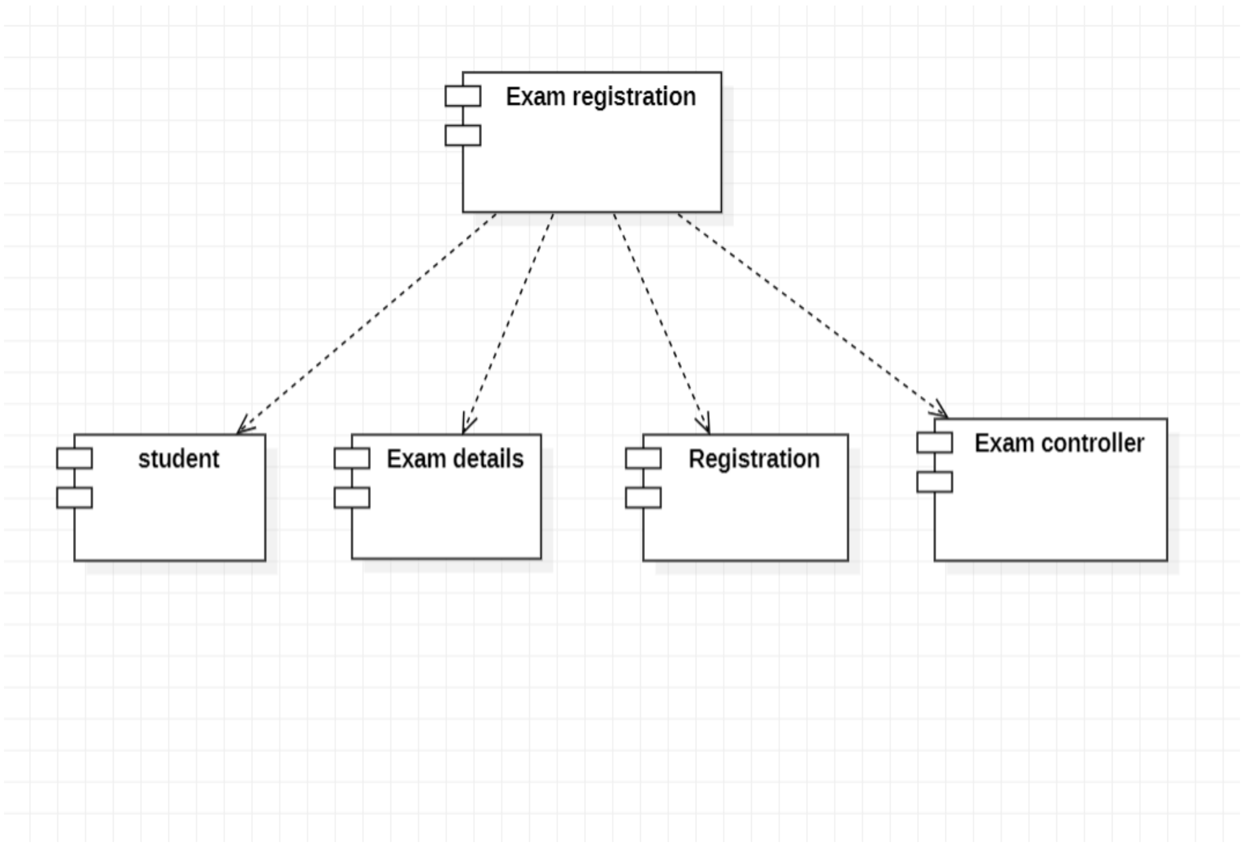
Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.



DEPLOYMENT DIAGRAM FOR EXAM REGISTRATION SYSTEM

3.7 COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.



COMPONENT DIAGRAM FOR EXAM REGISTRATION SYSTEM

4.SOURCE CODE:

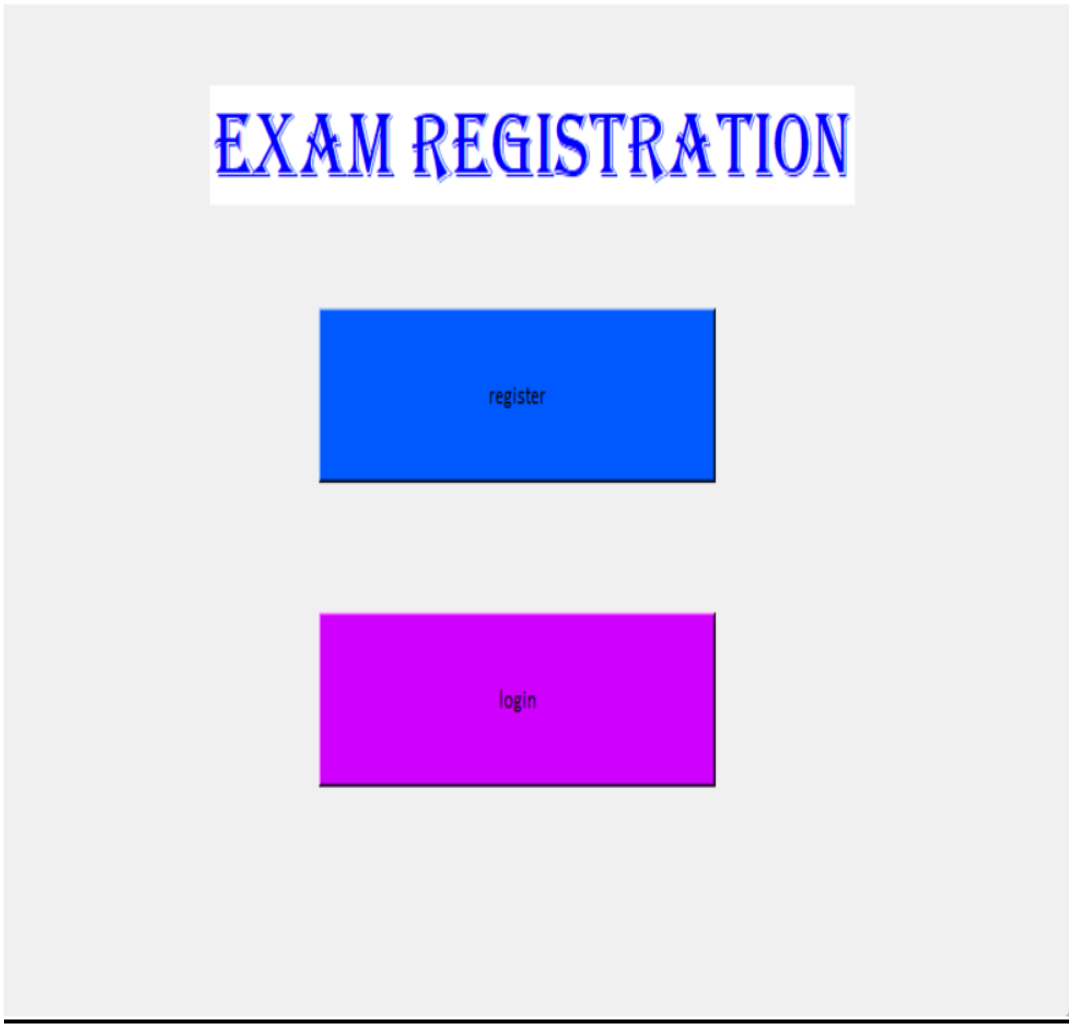
```
import tkinter

from tkinter import ttk,messagebox

import mysql.connector

from tkinter import*

root=Tk()

root.geometry("800x500")

def getvalue(student):

    s1.insert(0,select['st_id'])

    s2.insert(0,select['st_name'])

    s3.insert(0,select['dob'])

    s4.insert(0,select['qualification'])

    s5.insert(0,select['address'])

    s6.insert(0,select['mobile_no'])

    s7.insert(0,select['email_id'])

    s8.insert(0,select['username'])

    s9.insert(0,select['password'])
```

```
def validate():

    student_name= s1.get()

    email_id= s7.get()

    country= c.get()

    if (student_name=="" or email_id=="" or country== 'Select your country' or
gender == 0):

        tkinter.messagebox.showinfo('Invalid Message Alert',"Fields cannot be left
empty!")

    else:

        tkinter.messagebox.showinfo('Success Message',"Successfully registered!")

def add():

    student_id=s1.get()

    student_name=s2.get()

    dob=s3.get()

    qualification=s4.get()

    address=s5.get()

    mobile_no=s6.get()

    email_id=s7.get()
    username=s8.get()

    password=s9.get()

    mysqldb=mysql.connector.connect(host="localhost",user="root",password="
",database="exam registration")

    mycursor=mysqldb.cursor()

    try;

        sql="insert                                                    into
registration(st_id,st_name,dob,qualification,address,mobile_no,email_id,username,
password)values(%d,%s,%d,%s,%s,%d,%s,%s,%d)"


val=(student_id,student_name,dob,qualification,address,mobile_no,email_id,usern
ame,password)

        mycursor.execute(sql,val)

        mysqldb.commit()

        lastid=mycursor.lastrowid

        messagebox.showinfo("information","details inserted successfully")

        e1.focus_set()

    except:

        print(e)
        mysqldb.rollback()

        mysqldb.close()

    def show():

        mysqldb=mysql.connector.connect(host="localhost",user="root",password="
",database="exam registration")

        mycursor=mysqldb.cursor()

        mycursor.execute("select
st_id,st_name,dob,qualification,address,mobile_no,email_id,username,password
from registration")

        records=mycursor.fetchall()

        print(records)

        for
i,(st_id,st_name,dob,qualification,address,mobile_no,email_id,username,password)
in enumerate(records,start=1):

            mysqldb.close()

def register():

    root=Tk()

    root.geometry("800x500")

    root.title("exam registration")
    global s1

    global s2

    global s3

    global s4

    global s5

    global s6

    global s7

    global s8

    global s9


    Label(root,text="Exam                                             registration
form",fg="red",font=('calibri',30)).place(x=200,y=10)

    Label(root,text="student id").place(x=250,y=110)

    Label(root,text="student name").place(x=250,y=140)

    Label(root,text="dob").place(x=250,y=170)

    Label(root,text="qualification").place(x=250,y=260)

    Label(root,text="address").place(x=250,y=290)
```

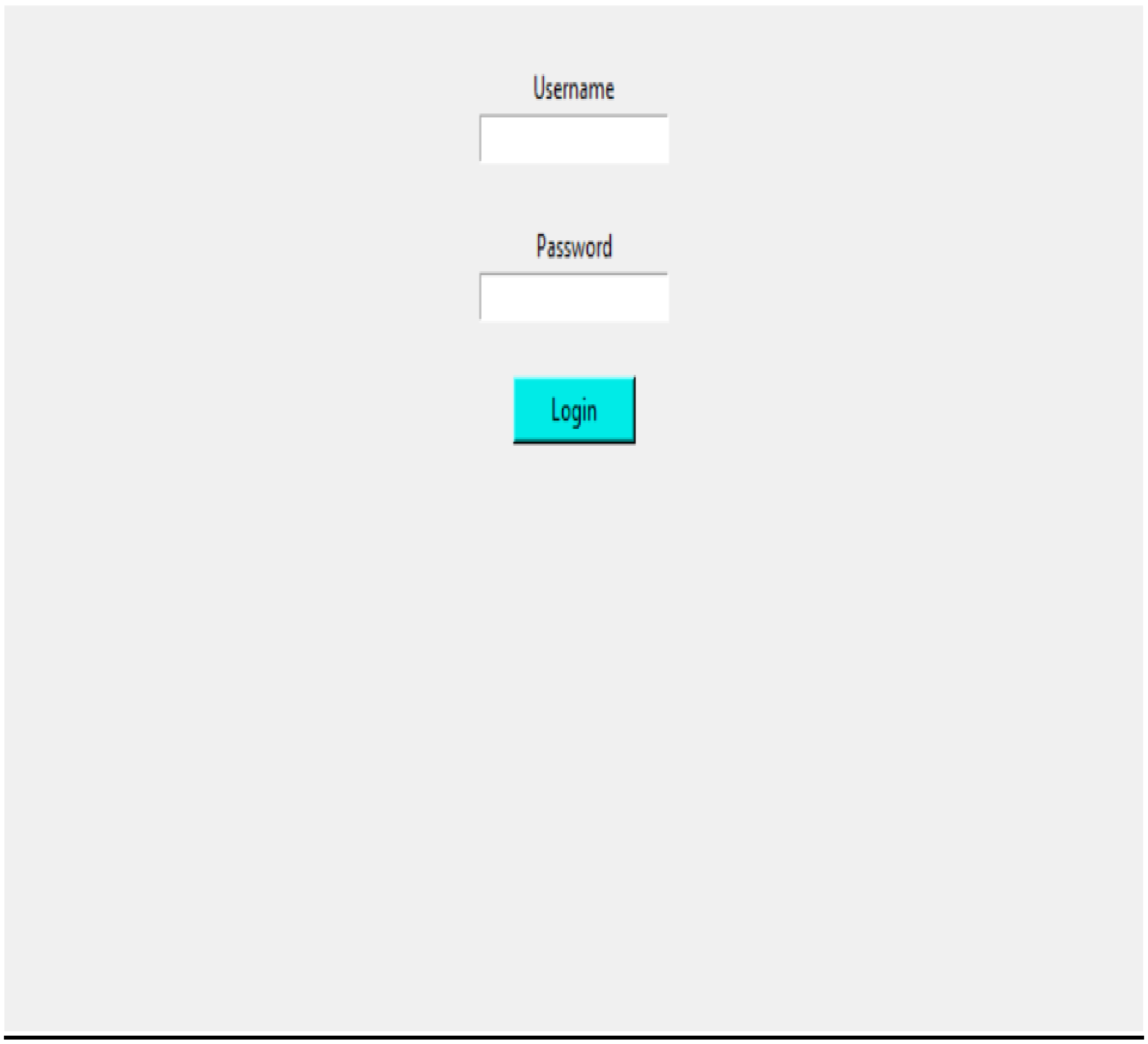
```
   Label(root,text="mobile_no").place(x=250,y=320)

   Label(root,text="email_id").place(x=250,y=350)

   Label(root,text="username").place(x=250,y=380)

   Label(root,text="password").place(x=250,y=410)

   c=StringVar()

   var=IntVar()

   label_9=Label(root,text="city",width=20,font=("bold",10))

   label_9.place(x=180,y=230)

   list1=['sivakasi','madurai','sattur','kovilpatti','trichy','chennai'];

   c.set("Select your city")

   droplist=OptionMenu(root,c,*list1)

   droplist.config(width=15)

   droplist.place(x=320,y=225)

   label_8=Label(root,text="gender",width=20,font=("bold",10))

   label_8.place(x=190,y=200)


Radiobutton(root,text="male",padx=5,variable=var,value=1).place(x=320,y=200)
Radiobutton(root,text="female",padx=5,variable=var,value=2).place(x=380,y=200
)

   s1=Entry(root)

   s1.place(x=330,y=110)

   s2=Entry(root)

   s2.place(x=330,y=140)

   s3=Entry(root)

   s3.place(x=330,y=170)

   s4=Entry(root)

   s4.place(x=330,y=260)

   s5=Entry(root)

   s5.place(x=330,y=290)

   s6=Entry(root)

   s6.place(x=330,y=320)

   s7=Entry(root)

   s7.place(x=330,y=350)
   s8=Entry(root)

   s8.place(x=330,y=380)

   s9=Entry(root)

   s9.place(x=330,y=410)

   Button(root,text="submit                                              to
verify",bg="#289166",command=validate,height=2,width=13).place(x=470,y=430
)

def login():

   root=Tk()

   root.title("Login")

   root.geometry("800x600")



   Label(root, text="").pack()

   Label(root, text="Username").pack()

   username_login_entry = Entry(root, textvariable="username")

   username_login_entry.pack()

   Label(root, text="").pack()

   Label(root, text="Password").pack()
    password__login_entry = Entry(root, textvariable="password", show= '*')

    password__login_entry.pack()

    Label(root, text="").pack()

    Button(root, text="Login", bg="#00EBE7",fg='black',width=10,
height=1).pack()

    root.mainloop()

text=Label(root,text="EXAM
REGISTRATION",foreground="blue",background="white",font=("algerian",35))

text.place(x=170,y=40)

Button(root,text="register",bg="#0059FF",command=register,height=5,width=40,f
ont=("calibri",10)).place(x=250,y=150)

Button(text="login",bg="#D000FF",height=5,command=login,width=40,font=("ca
libri",10)).place(x=250,y=300)

 root.mainloop()
```

5.OUTPUT:

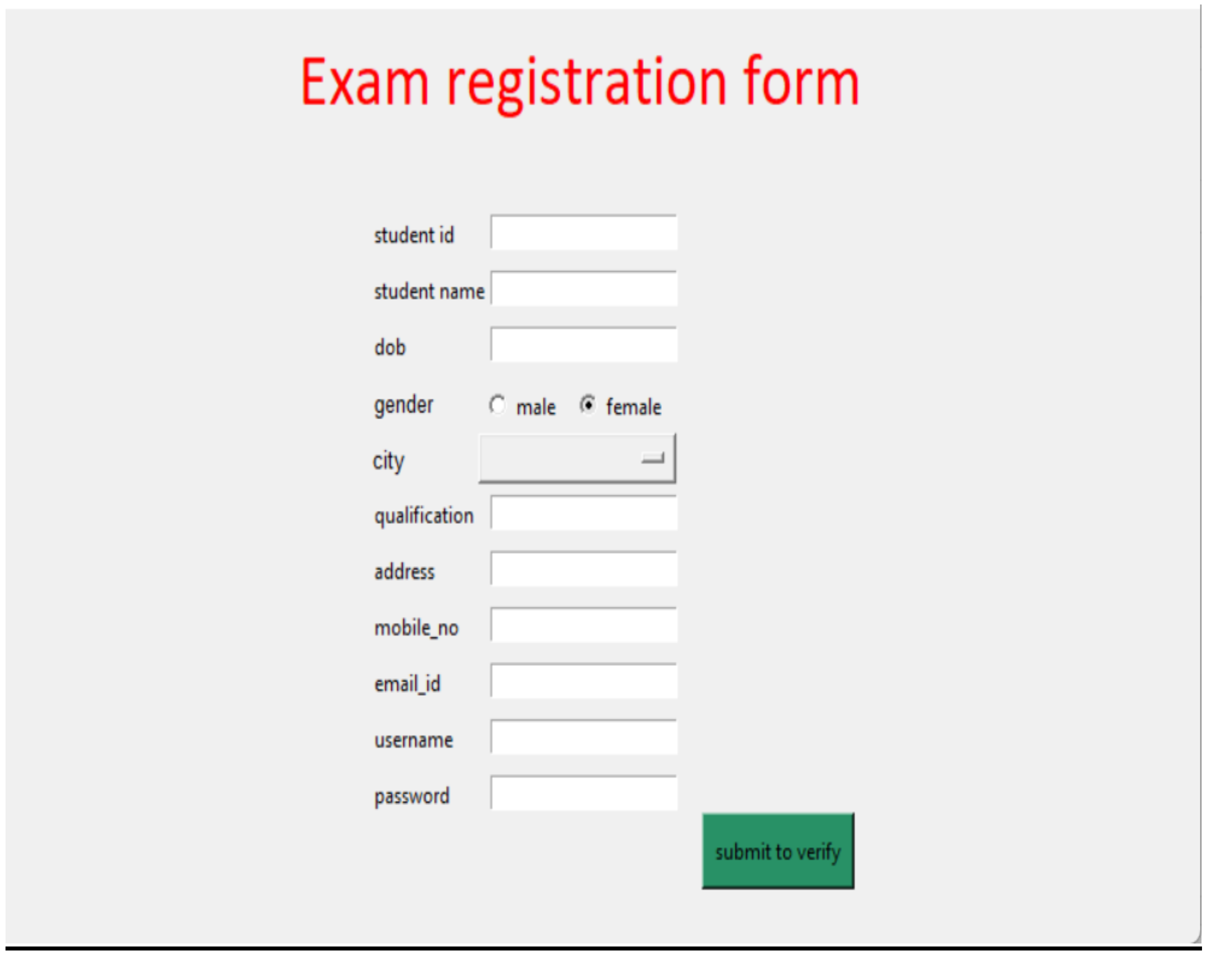
Main-page:



Login-page:



Register-page:



Exam registration form

student id	
student name	
dob	
gender	male [✓ female]
city	
qualification	
address	
mobile_no	
email_id	
username	
password	

RESULT:

Thus the mini project for Exam Registration system has been successfully executed and codes are generated.

❖