**WAKEUP**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**PELLIVILLA UMADEVI (1602-20-737-056)**

**RAMNENI THIRUPAHI (1602-20-737-031)**

**MD.ASADUDDIN (1602-20-737-024)**



**Department of Information Technology Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University and Approved by AICTE) Ibrahimbagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University and Approved by AICTE) Hyderabad-500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

We, **PELLIVILLA UMADEVI, THIRUPATHI RAMNENI ,** and **,ASADUUDIN**

bearing hall ticket numbers, **1602-20-737-056, 1602-20-737-031** and **1602-20-737-024**, hereby declare that the project report entitled **WAKEUP** is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**PELLIVILLA UMADEVI**

**1602-20-737-056**

**RAMNENI THIRUPATHI**

**1602-20-737-031**

**MD.ASADUDDIN**

**1602-20-737-024**

(Faculty In-Charge) (Head,Dept of IT)

**ACKNOWLEDGMENT**

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement.

We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators <Name#1> and

<Name#2>, for guiding us in understanding the process of project development & giving us timely suggestions at every phase.

We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

**ABSTRACT**

The main objective of the project is to wakeup the users in a by giving task to do. This project is useful for students and many people for getting up early in the morning. Users can use this alarm when they decided to wake up with intentionally and definately. Here the user has the chance to do select the music he want to play.

The whole process is going by importing so many library and by using threads. The framework which was used in this project was PYTHON.

**TABLE OF CONTENTS**

**1.Introduction**

a. Information------- 6

b. Features-----------7

**2.Technology**

a. Software Requirements---------8

b. Hardware Requirements---------8

**3.Proposed Work**

a. Design----------9

b. Implementation----------13

c. Testing------------21

**4.Results-------------23**

**5.Outcomes apart from curriculum-----26**

**6.Conclusion and future work------27**

**7. References**

**INTRODUCTION**

**Information**

An **alarm clock** is designed to alert an individual or group of individuals at a specified time. The primary function of these clocks is to awaken people from their night's sleep or short naps; they are sometimes used for other reminders as well. Most use sound then vibration. Usually to turn off the sound, a button or handle on the clock is pressed; most clocks automatically turn off the alarm if left unattended long enough but in this alarm the user must solve the question and should answer correctly then only it turn off otherwise it rings continuously.

Most of the people want to wakeup early but due to laziness and tiredness they turn off the alarm by just pressing one button and go back to the nap. But by using this alarm the user must solve the question to turn off the sound. To solve the question the user must have an active mind by which the mood to sleep will be change.

**b. Motivation**

We get these idea from us. Daily we put the alarm and turn it off and work which want to do or planned to do the early morning was get postponed continuously then we got an idea if we our mind get activated while we wake up it make the sleeping mood off.

And it is only possible when the we make some questions to solve in the alarm to turn off the alarm. As we get idea we tried to implement in python as your mini project. And we think it’s a new idea and make us to learn how the time and alarm works with python.

And our main moto or goal is to wake up the user.

**c. Features**

Our objective is to create a simple alarm which wake up the user by solving questions and wake up the user.

●In this, User has a choice to select the ringtone they want to play as sound.

●After that the user has to set the alarm time when he want to wake.

●When its time is equal to alarm set time then the sound will be on.

●At the same time the question is get displayed on GUI.

●If the user won’t solve correctly or they even attempt the question the question will display and sound will go on continuously.

●By solving question correctly then the question will disapper GUI.

**Technology**

**a. Hardware Requirements**

The hardware components used in order to create the wakeup(alarm) are as follows

* Computer with 2GB RAM
* Ringtone downloads
* Internet
* External drives for backup

**b. Software Requirements**

* We use pychram in order to develop the code for the game.
* We use our downloaded sounds to set as ringtone.
* Operating System. We have chosen Windows operating system for its best support and user friendliness.

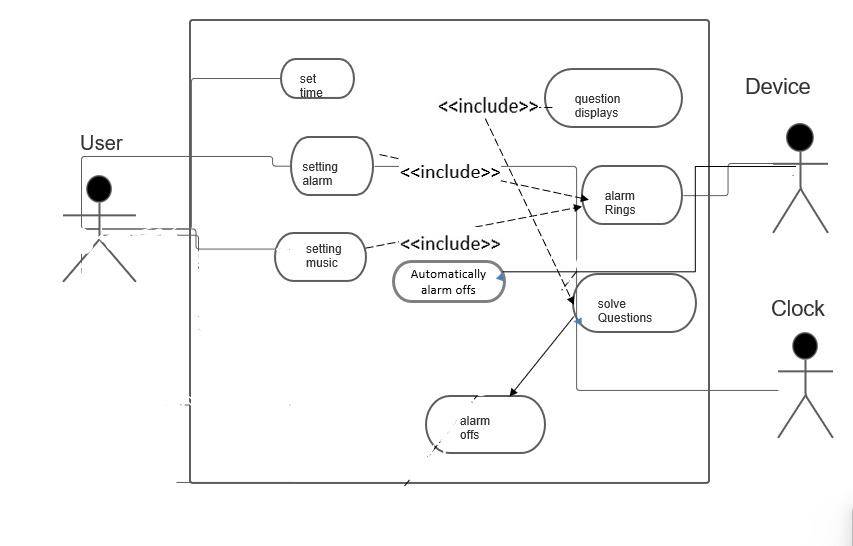
**PROPOSED WORK**

**a. Design**

**Use Case Diagram**

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



USECASE: To set the music.

|  |  |
| --- | --- |
| **User** | **System** |
|  | 1.There was button set ringtone |
| 2.Clicking on that the it display the menu we can set and play any music |  |
|  | 3.Then music will be set to that ringtone |

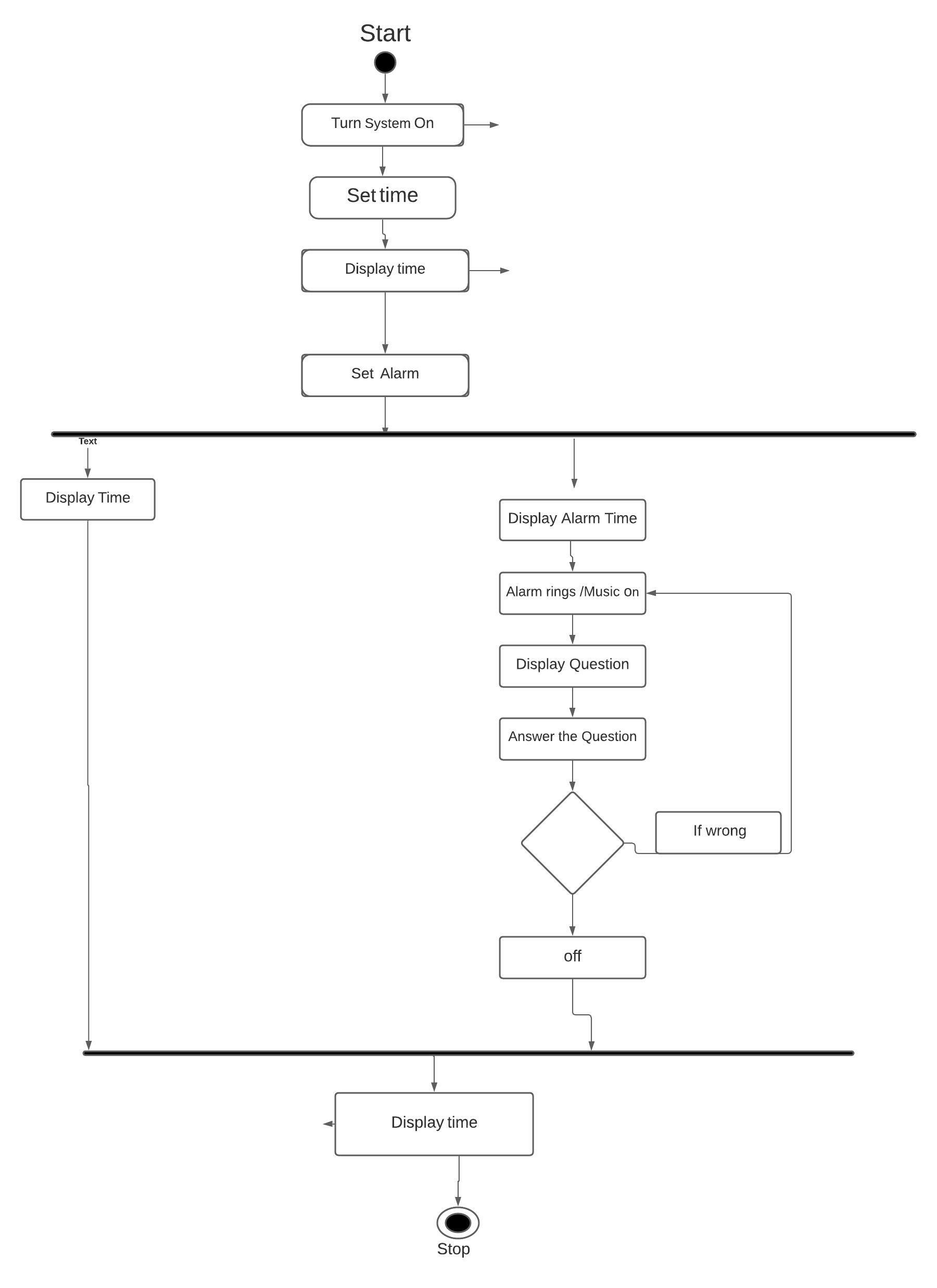
USECASE: To set alarm time

|  |  |
| --- | --- |
| **User** | **System** |
|  | 1.Digital alarm displayed on the screen. |
| 2.Set the alarm. When the user want to Wakeup. |  |
|  | 3.Then it displays that “alarm set”. |
|  |  |

USECASE: Display the question.

|  |  |
| --- | --- |
| **User** | **System** |
|  | 1.When alarm time is equal to current time display question |
| 2.Solves the question |  |
|  | If correct question disappears.  Else again the same question displays. |

ACTIVITY DIAGRAM



**MODULE WISE CODE**

**from tkinter import \***

**import datetime**

**import time**

**from threading import \***

**import random**

**import playsound as ps**

**from time import strftime**

**from time import sleep**

**import threading**

**def setmusic():  
 global root  
 root = Tk()  
  
 x = '1'  
 l1 = Label(root, text="Ringtone set to r" + x, font=("Helvetica 15 bold"))  
 global e1  
 e1 = Entry(root, text="Ringtone 1", width=100, border=20)  
 title=Label(root, text="select te music", font=("Helvetica 15 bold")).pack()  
 m = Frame(root)  
 m.pack()  
  
  
 Label(m, text="Ringtone1", font=("Helvetica 15 ")).grid(row=0,column=0)  
 Label(m, text="Ringtone2", font=("Helvetica 15 ")).grid(row=1,column=0)  
 Label(m, text="Ringtone3", font=("Helvetica 15 ")).grid(row=2,column=0)  
 Label(m, text="Ringtone4", font=("Helvetica 15 ")).grid(row=3,column=0)  
 Button(m, text="set", font=("Helvetica 15"), command=setmusic1).grid(row=0,column=1)  
 Button(m, text="set", font=("Helvetica 15"), command=setmusic2).grid(row=1,column=1)  
  
 Button(m, text="set", font=("Helvetica 15"), command=setmusic3).grid(row=2,column=1)  
  
 Button(m, text="set", font=("Helvetica 15"), command=setmusic4).grid(row=3,column=1)  
 Button(m, text="play", font=("Helvetica 15"), command=playmusic1).grid(row=0,column=2)  
 Button(m, text="play", font=("Helvetica 15"), command=playmusic2).grid(row=1,column=2)  
  
 Button(m, text="play", font=("Helvetica 15"), command=playmusic3).grid(row=2,column=2)  
  
 Button(m, text="play", font=("Helvetica 15"), command=playmusic4).grid(row=3,column=2)  
 #l1.pack()  
  
   
def setmusic1():  
 global ringtone  
 ringtone="C:\\Users\\uma36\\Downloads\\alarm.mp3"  
 Label(root, text="Ringtone set to r1", font=("Helvetica 15 bold")).pack()  
 sleep(5)  
 root.destroy()  
def setmusic2():  
 global ringtone  
 ringtone= "C:\\Users\\uma36\\Downloads\\alarm\_clock.mp3"  
 Label(root, text="Ringtone set to r2", font=("Helvetica 15 bold")).pack()  
 sleep(5)  
 root.destroy()  
def setmusic3():  
 global ringtone  
 ringtone="C:\\Users\\uma36\\Downloads\\alarm\_alarm.mp3"  
  
 Label(root, text="Ringtone set to r3", font=("Helvetica 15 bold")).pack()  
 sleep(5)  
 root.destroy()  
def setmusic4():  
 global ringtone  
 ringtone="C:\\Users\\uma36\\Downloads\\wake\_up.mp3"  
 Label(root, text="Ringtone set to r4", font=("Helvetica 15 bold")).pack()  
 sleep(5)  
 root.destroy()  
def playmusic1():  
 ps.playsound("C:\\Users\\uma36\\Downloads\\alarm.mp3")  
def playmusic2():  
 ps.playsound("C:\\Users\\uma36\\Downloads\\alarm\_clock.mp3")  
def playmusic3():  
 ps.playsound("C:\\Users\\uma36\\Downloads\\alarm\_alarm.mp3")  
def playmusic4():  
 ps.playsound("C:\\Users\\uma36\\Downloads\\wake\_up.mp3")  
   
def Question():  
 value=random.randint(1,10)  
 if(value==1):  
 question="569+23 "  
 ans='592'  
 o1='500'  
 o2='578'  
 o3=ans  
 o4='595'  
  
 if (value == 2):  
 question = "Which country are the Giza Pyramids in?"  
 ans = ' Egypt.'  
 o1 = 'Uk'  
 o2 = 'India'  
 o4 = ans  
 o3 = 'Italy'  
  
 if (value == 3):  
 question = " Which is the longest river on the earth?"  
 ans = 'Nile'  
 o3 = 'Ganga'  
 o2 = 'Krishna'  
 o1 = ans  
 o4 = 'yammuna'  
  
 if(value==4):  
 question = "7+45\*78 "  
  
  
 ans='3517'  
 o3 = '500'  
 o1 = '4000'  
 o2 = ans  
 o4 = '5400'  
  
 if (value == 5):  
 question = "Who invented Computer "  
  
 ans = ' Charles Babbage'  
 o3 = 'marks'  
 o1 = 'James Bond'  
 o2 = ans  
 o4 = 'michel'  
  
 if(value==6):  
 question="456+562+369"  
 ans='1387'  
 o3 = '2387'  
 o2 = '1587'  
 o1 = ans  
 o4 = '5405'  
  
 if(value==7):  
 question="145\*3"  
 ans='435'  
 o3 = '450'  
 o2 = '425'  
 o1 = ans  
 o4 = '540'  
 if(value==8):  
 question="3547\*2"  
 ans='7094'  
 o3 = '7504'  
 o1 = '7004'  
 o2 = ans  
 o4 = '5400'  
 if(value==9):  
 question="1458+589"  
 ans='2047'  
 o3 = '2509'  
 o2 = '3200'  
 o4 = ans  
 o1 = '5407'  
 if(value==10):  
 question="1235+965"  
 ans='2200'  
 o3 = '2500'  
 o2 = '2600'  
 o1 = ans  
 o4 = '3500'  
  
 return question, ans, o1, o2, o3, o4  
  
class Sound(Thread):  
 global answered  
 def run(self):  
 while True:  
 ps.playsound(ringtone)  
  
 if answered:  
 break  
 def stop(self):  
 return False  
class Alarm(Thread):  
 def run(self):  
 hi = Label(myWindow, text="alarm set", font=("Helvetica 15 bold")).pack()  
 while True:  
 global set\_alarm\_time  
 set\_alarm\_time=f"{hour.get()}:{minute.get()}:{second.get()}"  
 #time.sleep(1)  
 current\_time=datetime.datetime.now().strftime("%H:%M:%S")  
  
 strtime1=str(set\_alarm\_time)  
 hour1=strtime1[:2]  
 minute1=strtime1[3:5]  
 #print(minute1)  
  
 strtime2=str(current\_time)  
 hour2=strtime2[:2]  
 minute2=strtime2[3:5]  
 #print(minute2)  
 currentm=int(minute1)  
 global alarmtimem  
 global alarmtimeh  
 global ringtone  
  
 alarmtimem=int(minute2)  
 currenttimeh=int(hour1)  
 alarmtimeh=int(hour2)  
 if current\_time==set\_alarm\_time:  
 print("Time to wake up")  
  
  
 q.pack(pady=10)  
  
  
 q1.pack(pady=10)  
  
 oframe.pack()  
 Button(oframe, text=o1, font=("Helvetica 15"),command=setans1).pack(pady=20,side=LEFT)  
 Button(oframe, text=o2, font=("Helvetica 15"),command=setans2).pack(pady=20,side=LEFT)  
  
 Button(oframe, text=o3, font=("Helvetica 15"),command=setans3).pack(pady=20,side=LEFT)  
  
 Button(oframe, text=o4, font=("Helvetica 15"),command=setans4).pack(pady=20,side=LEFT)  
  
 t.start()  
 sleep(2)  
 c.start()  
  
class Check(Thread):  
 def run(self):  
 global answered  
 global set\_alarm\_time  
 while(answered==False):  
 time = set\_alarm\_time  
 ctime = datetime.datetime.now()-datetime.timedelta(minutes=1)  
 ctimes=ctime.strftime("%H:%M:%S")  
  
 print((myanswer, ans))  
 if myanswer == ans or time == ctimes:  
  
  
 answered=True  
 t.join()  
 oframe.destroy()  
 q.destroy()  
 q1.destroy()  
 a.join()  
  
  
def time():  
   
 str=strftime("%H : %M : %S %p")  
   
 lbl.config(text=str)  
   
 lbl.after(1000,time)  
  
def setans1():  
 global myanswer  
 myanswer=o1  
 print(myanswer)  
def setans2():  
 global myanswer  
 myanswer=o2  
 print(myanswer)  
def setans3():  
 global myanswer  
 myanswer=o3  
 print(myanswer)  
def setans4():  
 global myanswer  
 myanswer=o4  
 print(myanswer)  
answered=False  
myanswer=""  
question,ans,o1,o2,o3,o4=Question()  
t = Sound()  
c= Check()  
myWindow=Tk()  
  
myWindow.title("Digital Clock")  
  
lbl=Label(myWindow,font=("Algerian",50),background="black",foreground="cyan")  
lbl.pack()  
Button(myWindow,text="Set Ringtone",font=("Helvetica 15"),command=setmusic).pack(pady=20)  
q=Label(myWindow, text="Question", font=("Helvetica 20 bold"), fg="blue")  
q1=Label(myWindow, text=question, font=("Helvetica 12 "), fg="blue")  
oframe = Frame(myWindow)  
#calling time function  
time()  
Label(myWindow,text="Alarm Clock",font=("Helvetica 20 bold"),fg="blue").pack(pady=10)  
  
Label(myWindow,text="Set Time",font=("Helvetica 15 bold")).pack()  
ringtone="C:\\Users\\chand\\Desktop\\ringtone4.mp3"  
frame = Frame(myWindow)  
frame.pack()  
  
hour = StringVar(myWindow)  
hours = ('00', '01', '02', '03', '04', '05', '06', '07',  
 '08', '09', '10', '11', '12', '13', '14', '15',  
 '16', '17', '18', '19', '20', '21', '22', '23', '24'  
 )  
hour.set(hours[0])  
  
hrs = OptionMenu(frame, hour, \*hours)  
hrs.pack(side=LEFT)  
  
minute = StringVar(myWindow)  
minutes = ('00', '01', '02', '03', '04', '05', '06', '07',  
 '08', '09', '10', '11', '12', '13', '14', '15',  
 '16', '17', '18', '19', '20', '21', '22', '23',  
 '24', '25', '26', '27', '28', '29', '30', '31',  
 '32', '33', '34', '35', '36', '37', '38', '39',  
 '40', '41', '42', '43', '44', '45', '46', '47',  
 '48', '49', '50', '51', '52', '53', '54', '55',  
 '56', '57', '58', '59', '60')  
minute.set(minutes[0])  
  
mins = OptionMenu(frame, minute, \*minutes)  
mins.pack(side=LEFT)  
  
second = StringVar(myWindow)  
seconds = ('00', '01', '02', '03', '04', '05', '06', '07',  
 '08', '09', '10', '11', '12', '13', '14', '15',  
 '16', '17', '18', '19', '20', '21', '22', '23',  
 '24', '25', '26', '27', '28', '29', '30', '31',  
 '32', '33', '34', '35', '36', '37', '38', '39',  
 '40', '41', '42', '43', '44', '45', '46', '47',  
 '48', '49', '50', '51', '52', '53', '54', '55',  
 '56', '57', '58', '59', '60')  
second.set(seconds[0])  
  
secs = OptionMenu(frame, second, \*seconds)  
secs.pack(side=LEFT)  
a=Alarm()  
Button(myWindow,text="Set Alarm",font=("Helvetica 15"),command=a.start).pack(pady=20)  
  
mainloop()**

**c. TESTING**

**Testcase-1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC01 | | **Use Case ID:** UC01 | |
| **Test Case Title: Set Ringtone** | |  | |
| **Test Case Description:**  Verify whether user can apply any ringtone and gets the question and solve it and there was digital clock at the top. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.open the alarm using any editor and run.  2.Set ringtone  3.Set the alarm time. | The user gets access to the select ringtone if he follows the procedure mentioned and gets an GUI interface of alarm. | | The user sees the question and alarm gets off when solve correctly. |

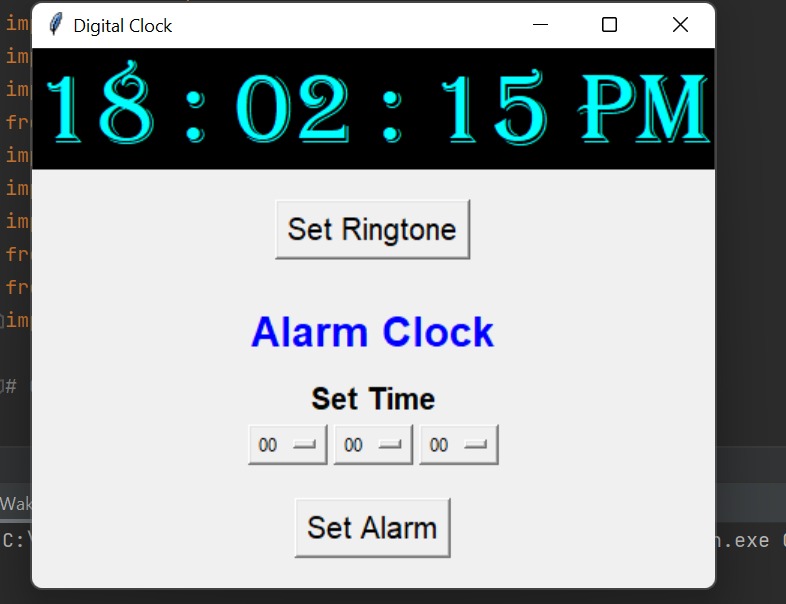
**Testcase-2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC02 | | **Use Case ID:** UC02 | |
| **Test Case Title: Set alarm time** | |  | |
| **Test Case Description:**  Verify whether user can apply any ringtone and gets the question and solve it and there was digital clock at the top | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.open the alarm using any editor and run.  2.Set ringtone  3.Set the alarm time. | The user gets access to the select ringtone if he follows the procedure mentioned and gets an GUI interface of alarm. | | The user sees the question and alarm gets off when solve correctly. |

**RESULT:**

**OUTPUT(Screen Shots)**

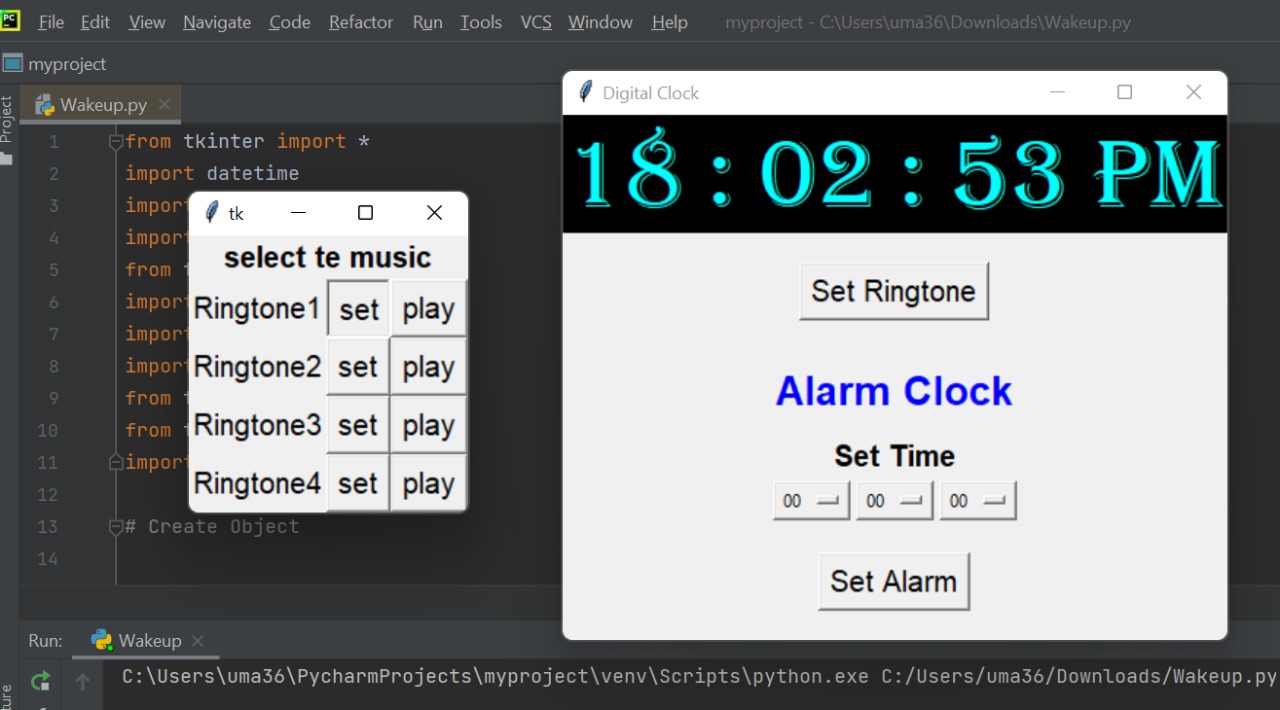
**When we run the code.**



**When we click on set ringtone**

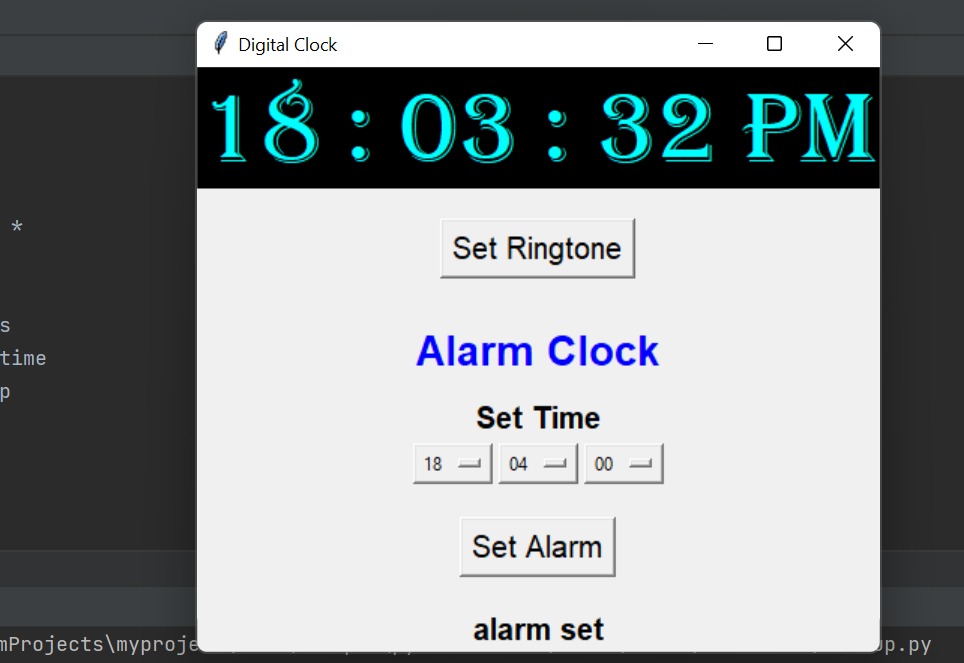
**By clicking on set it set the alarm**

**By clicking on play it paly the sound**



**After that we need to set the alarm time**

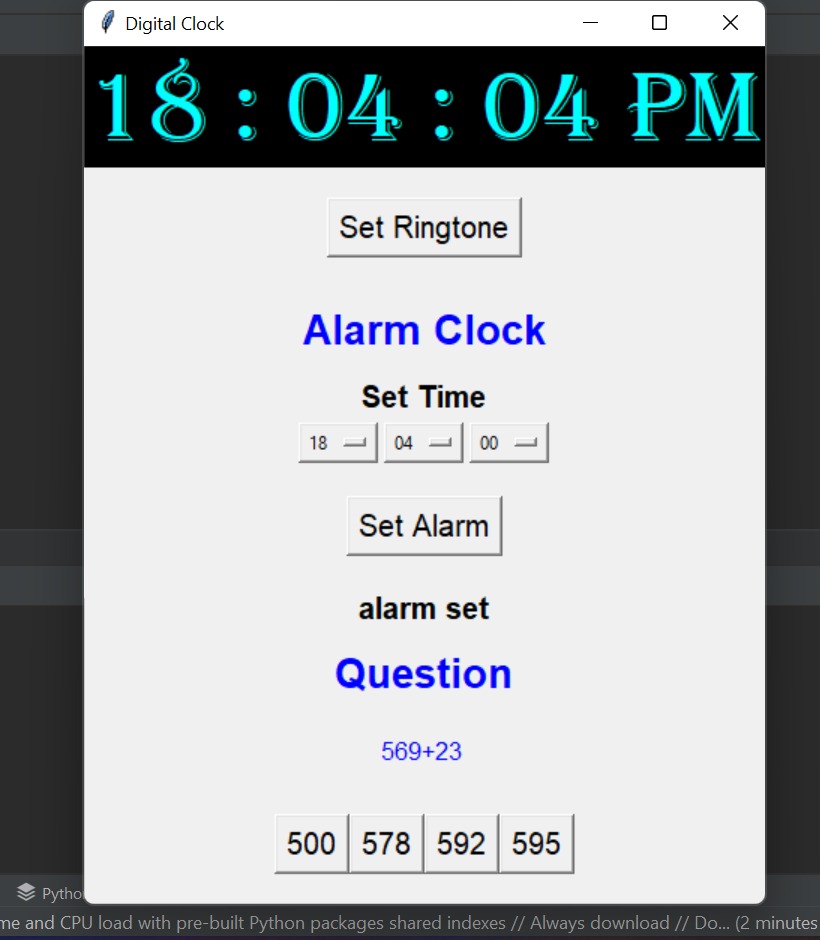
**Then we get the conformation that “alarm set”**



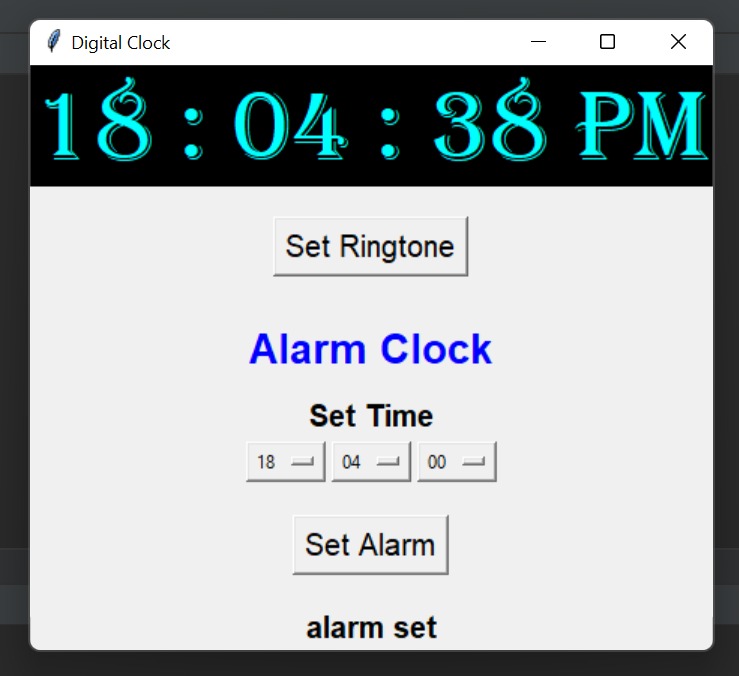
**When alarm time is done then the sound will be on.**

**At the same time question will display.**

**It will be continuous to ring until user enter the correct answer.**



**After solving the questions correctly the alarm went off.**



**Outcomes from the project apart from curriculum**

● I have got to know how to use different kinds of modules.

●I have discovered different kinds of libraries and modules and got to know their implementation and working.

●I have got to know about playsound, turtle most importantly tkinter.

●I have learnt how to build GUI using tkinter module.

●I have got to know how to handle events in GUI and about the components in them.

**CONCLUSION AND FUTURE WORK:**

In this wakeup project we implemented all the functions by using threads. This alarm as display question and turn off after question was solved. I hope this was very useful project. We learnt so many things.

F**uture work**

● We want to make the user to set the question type. Up to now user will get random questions.

● Improve Graphical Representation and make it more interesting.

● Introduce new features by asking questions by displaying pictures for kids where kids are users.

● And the questions and music will be set according to the users age in default.

The project is successfully implemented with all the features in wakeup is as per the requirements.

**REFERENCES:**

* **Youtube channel:** **https://www.youtube.com/watch?v=7pvc7NUQJ4w**
* **Geek for geeks**