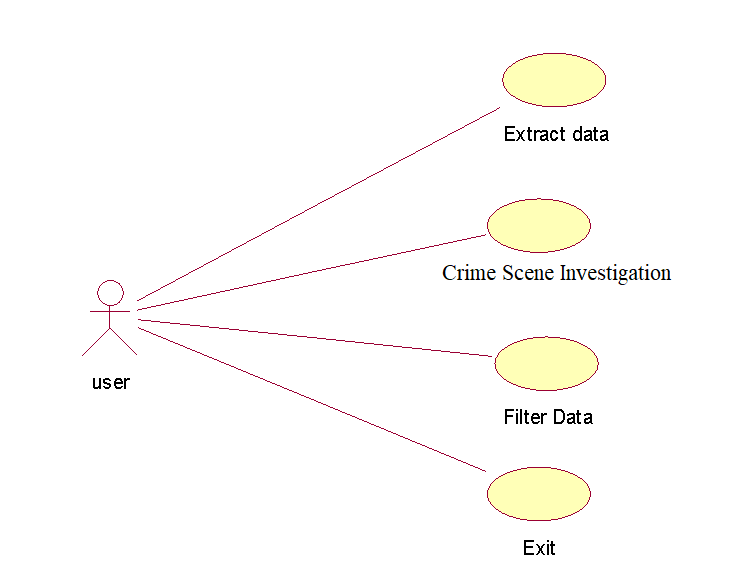


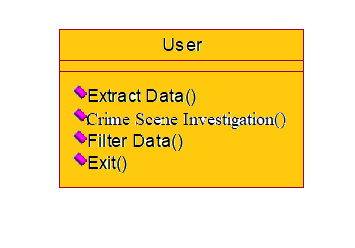
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



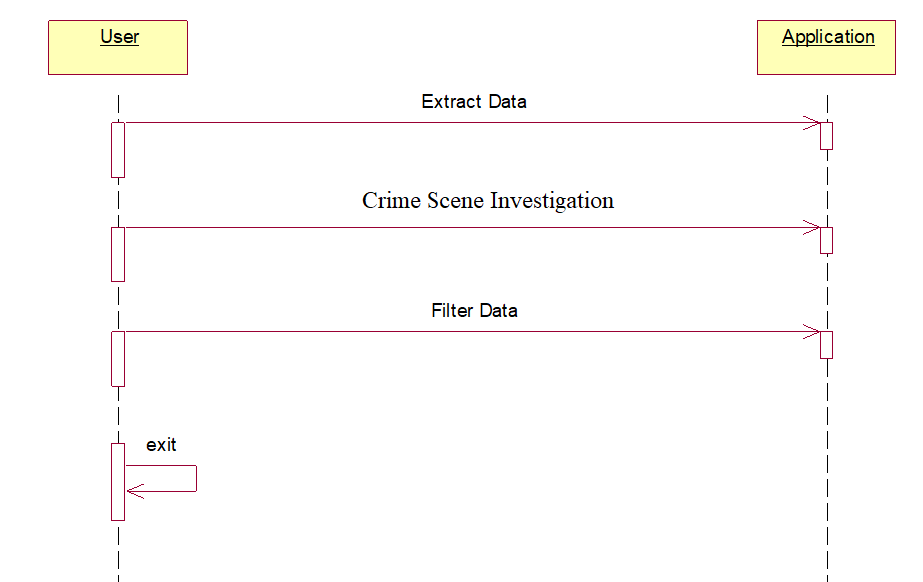
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Coding

from tkinter import messagebox

from tkinter import \*

from tkinter.filedialog import askopenfilename

from tkinter import simpledialog

import tkinter

import numpy as np

from tkinter import filedialog

from bs4 import BeautifulSoup

import datetime

import pathlib

main = tkinter.Tk()

main.title("A Digital Forensic Tool for Extracting User Activity from Mobile Devices")

main.geometry("1300x1200")

global filename

global testData

global content

def upload():

global filename

filename = filedialog.askopenfilename(initialdir = "MobileData")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,'Selected file loaded\n')

def extractData():

global content

global testData

text.delete('1.0', END)

with open(filename, 'rb') as f:

content = f.read().decode("utf-16")

f.close()

soup = BeautifulSoup(str(content), "html.parser")

testData = soup.text

text.insert(END,content)

font = ('times', 16, 'bold')

title = Label(main, text='A Digital Forensic Tool for Extracting User Activity from Mobile Devices')

title.config(bg='dark goldenrod', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

upload = Button(main, text="Upload Mobile Data", command=upload)

upload.place(x=700,y=100)

upload.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='DarkOrange1', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=700,y=150)

featureextractionButton = Button(main, text="Extract Data", command=extractData)

featureextractionButton.place(x=700,y=200)

featureextractionButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=80)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=100)

text.config(font=font1)

main.config(bg='Lavender')

main.mainloop()