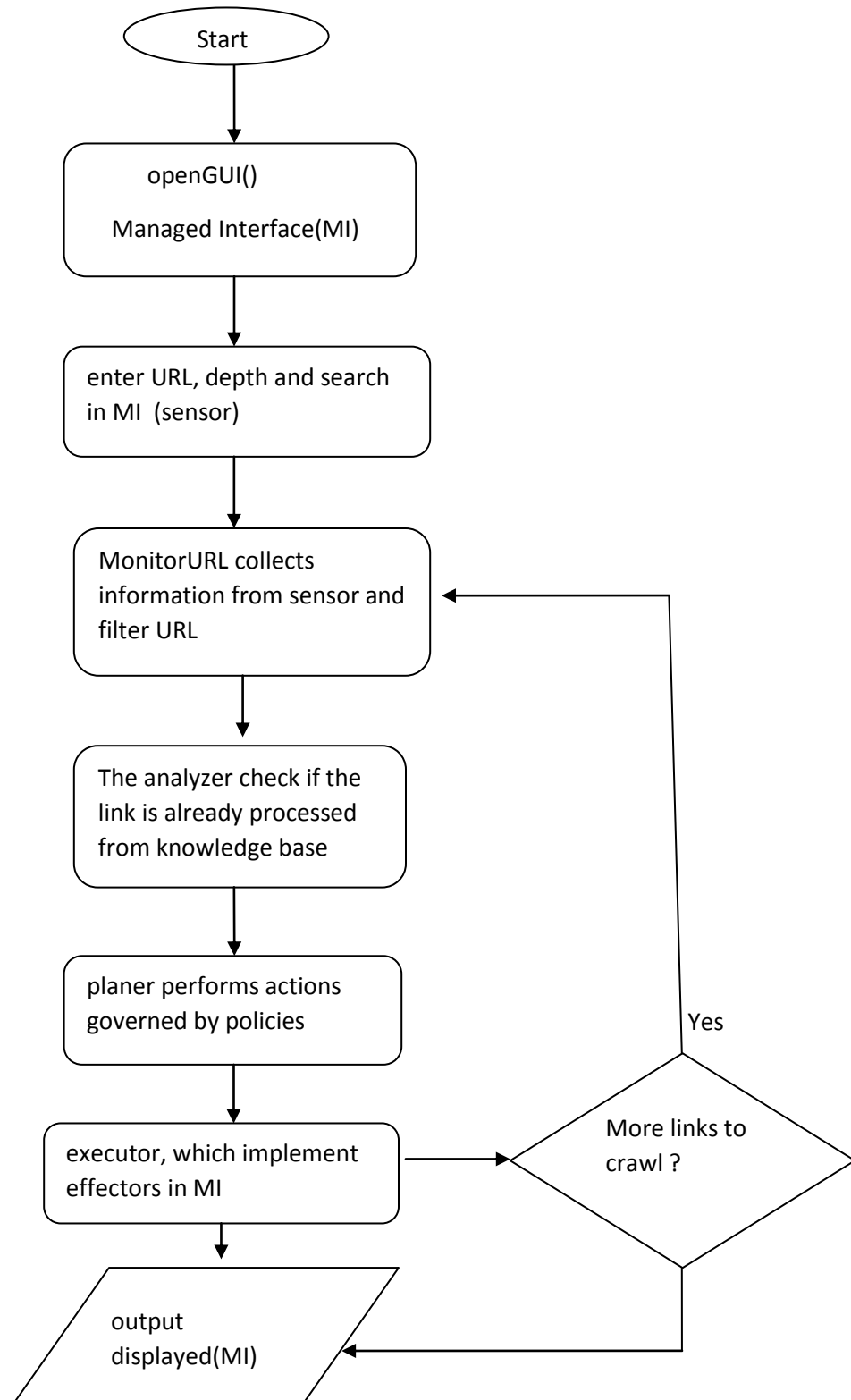
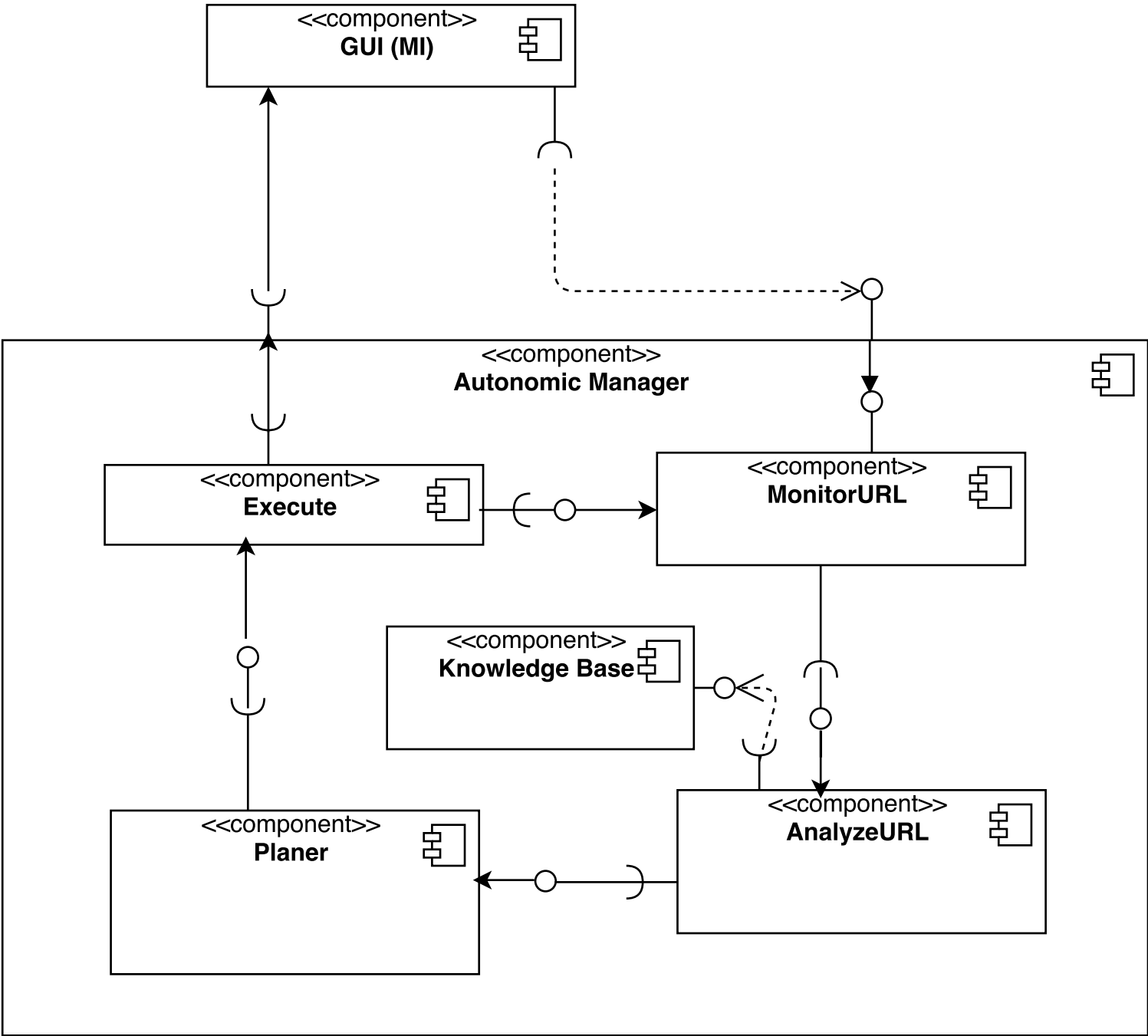
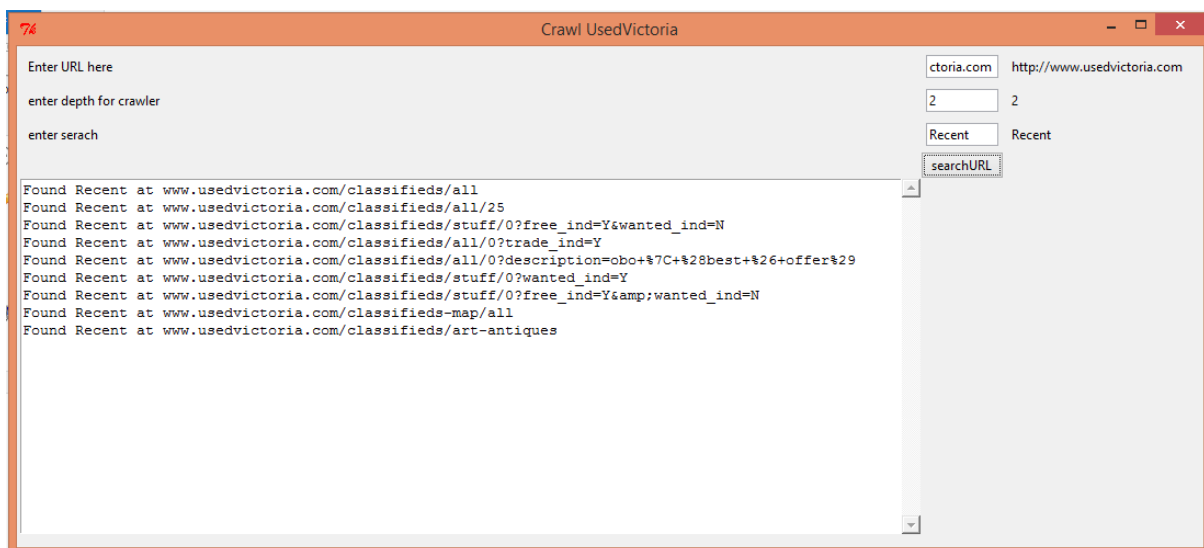
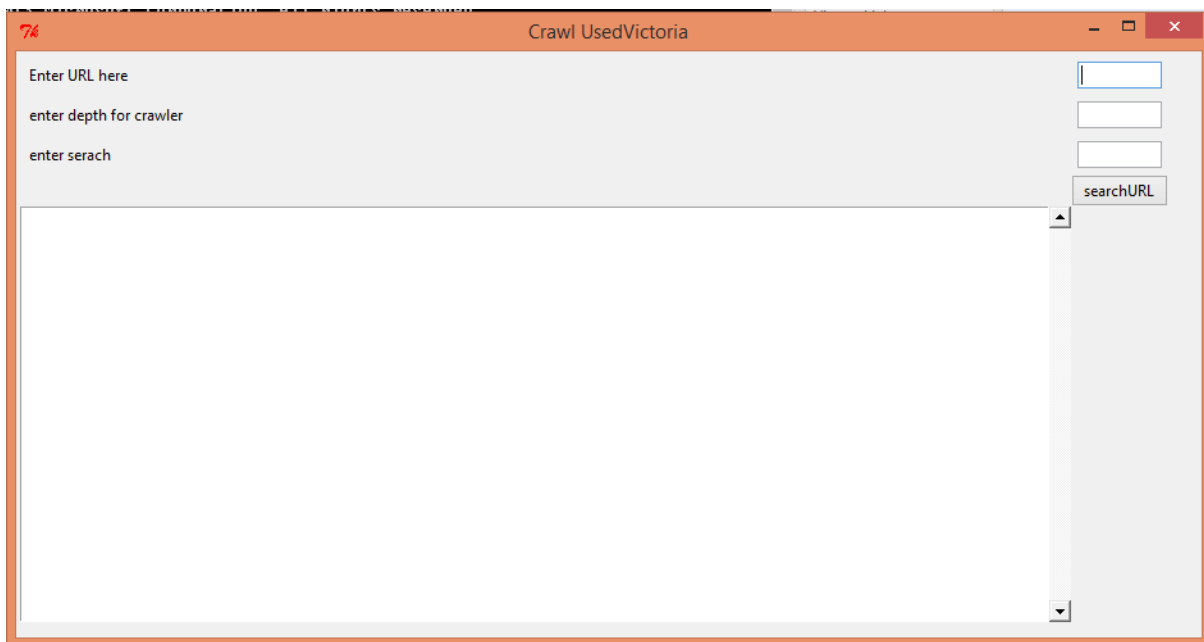


## Assignment 3 : Part 2





## Screenshots of Web crawler



Code for Crawler:

```
import httpplib
import re
import argparse
import ttk
from Tkinter import *
from ttk import *
import ScrolledText
processed = []
def open_GUI():
    global textPad, root
    root= Tk()
    root.title("Crawl UsedVictoria")

    mainframe = ttk.Frame(root, padding="3 3 12 12")
    mainframe.grid(column=0, row=0, sticky=(N, W, E, S))
    mainframe.columnconfigure(0, weight=1)
    mainframe.rowconfigure(0, weight=1)

    url = StringVar()
    depth = StringVar()
    search = StringVar()
    ttk.Label(mainframe, text="Enter URL here").grid(column=1, row=1,
sticky=(W,E))
    ttk.Label(mainframe, text="enter depth for crawler").grid(column=1,
row=2, sticky=(W))
    ttk.Label(mainframe, text="enter serach").grid(column=1, row=3,
sticky=(W))

    url_entry = ttk.Entry(mainframe, width=7, textvariable=url)
    url_entry.grid(column=2, row=1, sticky=(W,E))
    depth_entry = ttk.Entry(mainframe, width=7, textvariable=depth)
    depth_entry.grid(column=2, row=2, sticky=(W,E))
    search_entry = ttk.Entry(mainframe, width=7, textvariable=search)
    search_entry.grid(column=2, row=3, sticky=(W,E))

    ttk.Label(mainframe, textvariable=url).grid(column=5, row=1,
sticky=(W,E))
    ttk.Label(mainframe, textvariable=depth).grid(column=5, row=2,
sticky=(W))
    ttk.Label(mainframe, textvariable=search).grid(column=5, row=3,
sticky=(W))

    for child in mainframe.winfo_children():
child.grid_configure(padx=5, pady=5)

    url_entry.focus()
    depth_entry.focus()
    search_entry.focus()
    textPad = ScrolledText.ScrolledText(mainframe, width=100,
height=20)
    textPad.grid(column=1, row=8, sticky=(W))
```

```

        ttk.Button(mainframe, text="searchURL", command=lambda :
MonitorURL(url.get(),int(depth.get()),search.get())).grid(column=2,
row=5, sticky=W)

    root.mainloop()

def MonitorURL(url1,depth1,search1):
    if url1.startswith("http://"):
        analyzeURL(url1,depth1,search1)
    else:
        textPad.insert(END,"skipping"+url1+"\n")

def analyzeURL(url1,depth1,search1):

    if (not url1 in processed):
        processed.append(url1)
        url1 = url1.replace("http://", "", 1)
        host, path = url1, "/"

        urlparts = url1.split("/")
        if len(urlparts) > 1:
            host = urlparts[0]
            path = url1.replace(host, "", 1)
            planer(host,path,url1,depth1,search1)
        else:
            planer(host,path,url1,depth1,search1)

def planer(host,path,url1,depth1,search1):

    flag=0
    href = "http://"
    # make the first request
    conn = httpplib.HTTPConnection(host)
    req = conn.request("GET", path)
    res = conn.getresponse()

    # find the links in classifieds only
    contents = res.read()
    m = re.findall('href="(//classifieds+.*?)"', contents)

    # looking for the word to be searched
    if search1 in contents:
        flag=1
        execute(search1,url1,flag,href,depth1)

    #Check for following links
    for href in m:
        if href.startswith("/"):
            href = "http://" + host + href
            flag=2
            execute(search1,url1,flag,href,depth1)

def execute(search1,url1,flag,href,depth1):
    if(flag==1):
        textPad.insert(END,"Found " + search1 + " at " + url1+ "\n")

```

```
        root.update()
    # follow the links
    if (flag==2):
        if depth1:
            MonitorURL(href, depth1-1, search1)

if __name__ == "__main__":
    open_GUI()
```

CSC586A-Navpreet-Kaur-V00823334  
CSC586A-Parminder-Kaur-V00820508  
CSC586A-Simar-Arora-V00824821