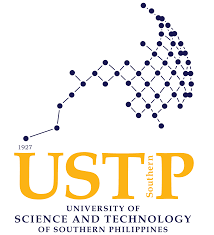
**ONE PERFORMANCE INNOVATIVE TASK**

**“Offline Visitor Log Management System : Enhancing Security and Efficiency at USTP Oroquieta”**

****

In Partial Fulfillment of the Requirements in

Department of Information Technology

University of Science and Technology of Southern Philippines

Oroquieta City Campus

**Lovern Jane B. Dela Cruz**

**Gretsin Indus**

**Aj Gomez**

**Ruth Nalagon**

May 2025

**PROBLEM ANALYSIS REPORT**

**Introduction**

At University of Science and Technology of Southern Philippines (USTP) Oroquieta Campus, the current process of visitor logging is manual which is they relying only on traditional pen and paper logbooks at the guard house. Obviously, this method is time-consuming, and messy. It is also prone to missing list, and the handwriting of a person is not clear resulting in hard to organize data, and it is difficult to access the previous records for the previous visitors. Security and data tracking are quietly being affected. Through the proposed USTP Visitor Log Management System, visitors can easily input their personal information in the system so that the list of visitors in USTP Oroquieta Campus can be kept over the years in safety and enhanced the security without relying on internet connection.

**Community Problem Description**

The current way of recording the information of visitors is slow, manual, and unsafe. It causes problems in keeping track of people, responding to emergencies quickly, and storing records properly.

A. Outdated Manual Logging. Currently, visitors information Is recorded using paper-based logbooks, which are hard to manage, prone to damage, loss and data inconsistency. Retrieving past records requires manual searching which is time-consuming.

B. Lack of Real-time Monitoring. There is no system to monitor the visitor flow in real-time. Securities cannot respond quickly when there is emergencies occur. There is no easy way to access the records immediately, especially when they need an information about the past visitors.

C. Unable Record Management. The data for visitors is stored only in logbooks without backups meaning when the logbooks is full every data is stored in archived with no digital copy. This results that there is a limited access for every data and there is no way to analyze the copies of every visitors entering the campus over time. That is why it leads to weakens the security protocol.

**Relevance and Impact Assessment**

Developing an Offline Visitor Log Management System brings many benefits to this school at USTP Oroquieta, especially improving security and safety, accuracy, and also comfort.

A.Improved Campus Security. Digital visitor log system enhances security by providing accurate and timely information about those people who enter the campus who are not students, staff, or instructors. It also helps in identifying suspicious entries or patterns and easier to access for emergency purposes.

B.Data Accuracy and Accessibility. Storing visitors data, the Visitor Log Management System reduces errors and allows for instant search and reporting. Data can be accessed by authorized personnel anytime, making it more useful and easier to access for emergency purposes.

C.Faster Visitor Process. This system reduces time spent at entry points, they don’t need to write but to encode their name, purpose, and other information needed, leading to smoother and faster visitor processing.

**Conclusion and Recommendations**

Enhancing visitor logging at USTP-Oroquieta Campus is not just about convenience; it’s about safety, security, and modernization. The problems of outdated manual processes, inefficient data handling, and lack of communication channels can be addressed by developing a Python-based Visitor Management System. This system will allow for automated check-ins, accurate visitor information, easy data retrieval, and improved response during emergencies.

Through adopting this tech-based solution, USTP can not only improve its visitor management practices but also set an example of how digital tools can support day-to-day campus operations effectively.

**Recommendations:**

1.Develop a Python-based visitor log management system for USTP-Oroquieta Campus. This group will create a computer program using Python language that will record all visitor information. This system will replace the paper logbook and make recording faster, safer, and easier to manage.

2. Include features like login, contact info,purpose of visit, and timestamp logging.The system will allow the visitors to enter their name, phone number, address, purpose of visiting, and the exact time and date they arrive . These details will help keep complete and clear records of everyone who enter the campus.

3. Implement a search function to allow quick filtering of past visitors and also to retrieve the data. Add a search tool so that staff can quickly find visitor information by typing their name, address, contact number, date or reason. This will save as time, especially during emergencies or if someone needs to be traced.

4. Train front desk/security personnel to operate the system efficiently. This group will teach the people using this system like guards and receptionist of how to use it properly. Training will help them to avoid mistakes and make sure everything is safe and convenience.

5. Promote the system across departments as part of USTP’s digital transformation initiatives. This group will introduce our system to enhance the security and efficiency at USTP Oroquieta to make a sustainable campus when it comes to security.

**TECHNOLOGICAL SOLUTION PROPOSAL**

**(3 to 6 pages)**

“**OFFLINE VISITOR LOG MANAGEMENT SYSTEM: ENHANCING SECURITY AND EFFICIENCY AT USTP OROQUIETA”**

**l.Overview of the Proposed Solution**

The Offline Visitor Log Management System is a simple computer program made with Python. It is a console application, which means it runs in a text-only screen or Text User Interface and does not need a mouse or graphics. This system is designed to replace the old manual logbook used at the University of Science and Technology of Southern Philippines (USTP) Oroquieta Campus.

Its goal is to make the process of recording visitor information faster, safer, and more accurate. It works offline, which means it does not need internet. It helps campus guards or staff keep track of all visitors using a digital method.

**Main Features of the System:**

• **Visitor Log-in:**

A user can enter their full name, contact number, purpose for visit, and address. The time and date of the visit are added automatically by the system. This makes sure the record is always correct and consistent.

• **Search Visitor by Name**:

It allows the user to search for a visitor using their full or partial name. This helps find visitor records quickly.

• **View All Visitors:**

This option shows a list of all people who visited, along with their full details, including the auto-recorded date and time.

• **Exit the Program:**

This allows the user to close the system properly when they are done.

Our system which is Offline Visitor Log Management System is to help school guards or staff record and track visitors in a safe and organized way without needing the internet. It allows users to input visitor details like name, contact number, purpose of visiting, and the automated time in and date. The system saves the records in a file so they can be checked later if needed. This helps improve security, make visitor tracking easier, and avoid using paper logbooks.

Even though our system is simple and offline, it is very useful and secure for managing visitor records. Actually, this system gives people in the community at USTP-Oroquieta Campus peace of mind, knowing that visitors who enterd in the campus are properly recorded without internet and it saves time and paper logbooks , which is good in our environment.

**ll. Algorithms and Data Structures Used**

Although the Offline Visitor Log Management System is simpler than a full stack application, it effectively used core programming languages which is Python and data structures to manage visitor data and enhancing real-world security especially in USTP Oroquieta.

**A.Core Features and Algorithms**

**1. Visitor Login**

1.Visitors enter their Full Name (validated for completeness), Contact Number (must be exactly 11 digits), Purpose of Visit, and Address.

2. Our system adds the current time and date automatically.

3. Visitors information is stored as a dictionary in an in-memory list.

**2. Data Storage and Retrieval**

Visitor information is saved in a text file called visitor\_log.txt. Each visitor’s details are stored on one line, separated by the “|” symbol. When the program starts, it reads this file and breaks each line into parts to make a visitor dictionary. All visitor dictionaries are saved in a list called visitor\_log while the program runs.

**3. Data Structures Used**:

List: Holds all visitor dictionaries in the program’s memory.

Dictionary: Each visitor is stored as a dictionary with these keys: name, contact\_number, purpose, time\_in, date, and address.

**4. Search Function**

The program looks through the list of visitors one by one to find names that include the search word. It ignores uppercase or lowercase letters when searching.

**B. Key Data Structures**

**1.Visitor Entry (Dictionary)**

Each visitor is stored in memory using a dictionary with specific keys:

**Format:**

Visitor = {

“name”: name,

“contact\_number”: contact\_number,

“purpose”: purpose,

“time\_in”: time\_in,

“date”: date,

“address”: address

}

**2.Visitor Log (List of Dictionaries)**

The main in-memory database is a list that contains all visitor entries.

**Format:**

Visitor\_log = []

Visitor\_log.append(visitor)

**3.Visitor Record in File (String Format)**

Each visitor is saved in visitor\_log.txt as a formatted string:

**Format:**

Def save\_visitor\_to\_file(visitor):

With open(“visitor\_log.txt”, “a”, encoding=”utf-8”) as file:

Line = f”{visitor[‘name’]} | {visitor[‘contact\_number’]} | {visitor[‘purpose’]} | {visitor[‘time\_in’]} | {visitor[‘date’]} | {visitor[‘address’]}\n”

File.write(line)

**lll. Minimum Viable Product (MVP) prototype of the Offline Visitor Log Management System written in Python.**

Core functionalities:

•Visitor Log In

• Search Visitors Name

• View all visitors

• Exit

import os

from datetime import datetime

visitor\_log = []

class bcolors:

    HEADER = '\033[95m'

    PINK = '\033[95m'

    CYAN = '\033[96m'

    GREEN = '\033[92m'

    OKBLUE = '\033[94m'

    YELLOW = '\033[93m'

    FAIL = '\033[91m'

    ENDC = '\033[0m'

    BOLD = '\033[1m'

UNDERLINE = '\033[4m'

def clear\_screen():

    os.system('cls' if os.name == 'nt' else 'clear')

def header\_art():

    print(bcolors.CYAN + bcolors.BOLD)

    print("\t\t\t\t\t\t\t  ,--. ,--. ,---. ,--------.,------. ")

    print("\t\t\t\t\t\t\t  |  | |  |'   .-''--.  .--'|  .--. '  ")

    print("\t\t\t\t\t\t\t  |  | |  |`.  `-.   |  |   |  '--' | ")

    print("\t\t\t\t\t\t\t  '  '-'  '.-'    |  |  |   |  | --' ")

    print("\t\t\t\t\t\t\t   `-----' `-----'   `--'   `--'" + bcolors.ENDC)

def show\_menu():

    print("\n" + " " \* 52 + bcolors.PINK + "=" \* 46 + bcolors.ENDC)

    print(" " \*52 + bcolors.YELLOW + "== WELCOME TO VISITOR LOG MANAGEMENT SYSTEM ==" + bcolors.ENDC)

    print(" " \* 52 + bcolors.PINK + "=" \* 46 + bcolors.ENDC)

    print(" " \* 63 + bcolors.GREEN + "1. Log In Visitor" + bcolors.ENDC)

    print(" " \* 63 + bcolors.GREEN + "2. Search Visitor by Name" + bcolors.ENDC)

    print(" " \* 63 + bcolors.GREEN + "3. View All Visitors" + bcolors.ENDC)

    print(" " \* 63 + bcolors.FAIL + "4. Exit" + bcolors.ENDC)

def get\_input(prompt, field\_name, validate\_func=None):

    while True:

        value = input(" " \* 28 + prompt).strip()

        if value == "":

            print(" " \* 28 + bcolors.FAIL + f"Please enter your {field\_name}." + bcolors.ENDC)

        elif validate\_func and not validate\_func(value):

            continue

        else:

            return value

def is\_valid\_name(name):

    parts = name.split()

    if len(parts) < 2:

        print(" " \* 50 + bcolors.FAIL + "Please enter your full name (first and last name)." + bcolors.ENDC)

        return False

    return True

def capitalize\_name(name):

    return ' '.join(word.capitalize() for word in name.split())

def is\_valid\_contact(number):

    if not number.isdigit():

        print(" " \* 63 + bcolors.FAIL + "Please enter a valid contact number (numbers only)." + bcolors.ENDC)

        return False

    elif len(number) != 11:

        print(" " \* 50 + bcolors.FAIL + "Contact number must be exactly 11 digits." + bcolors.ENDC)

        return False

    return True

def is\_valid\_address(address):

    if len(address) < 10:

        print(" " \* 50 + bcolors.FAIL + "Please enter a complete address (at least 10 characters)." + bcolors.ENDC)

        return False

    return True

def save\_visitor\_to\_file(visitor):

    with open("visitor\_log.txt", "a", encoding="utf-8") as file:

        line = f"{visitor['name']} | {visitor['contact\_number']} | {visitor['purpose']} |  {visitor['time\_in']} | {visitor['date']} | {visitor['address']}\n"

        file.write(line)

def load\_visitors\_from\_file():

    if os.path.exists("visitor\_log.txt"):

        with open("visitor\_log.txt", "r", encoding="utf-8") as file:

            for line in file:

                parts = line.strip().split("|")

                if len(parts) == 6:

                    visitor\_log.append({

                        "name": parts[0],

                        "contact\_number": parts[1],

                        "purpose": parts[2],

                        "time\_in": parts[3],

                        "date": parts[4],

                        "address": parts[5]

                    })

def log\_in\_visitor():

    print("\n" + " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    print(" " \* 63 + bcolors.YELLOW + "=== VISITOR LOG IN ===" + bcolors.ENDC)

    print( " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    name\_raw = get\_input(" " \* 22 + bcolors.BOLD +"Enter full name: ","full name", is\_valid\_name)

    name = capitalize\_name(name\_raw)

    contact\_number = get\_input(" " \* 22 + bcolors.BOLD +"Enter contact number: ", "contact number", is\_valid\_contact)

    purpose = get\_input(" " \* 22 + bcolors.BOLD +"Enter purpose of visit: ", "purpose")

    address = get\_input(" " \* 22 + bcolors.BOLD +"Enter address: ", "address", is\_valid\_address)

    now = datetime.now()

    time\_in = now.strftime("%H:%M")

    date = now.strftime("%Y-%m-%d")

    print(" " \* 50 + bcolors.BOLD + f"Time In: {time\_in} | Date: {date}" + bcolors.ENDC)

    visitor = {

        "name": name,

        "contact\_number": contact\_number,

        "purpose": purpose,

        "time\_in": time\_in,

        "date": date,

        "address": address

    }

    visitor\_log.append(visitor)

    save\_visitor\_to\_file(visitor)

    print(" " \* 50 + bcolors.YELLOW + f"Visitor '{name}' logged successfully." + bcolors.ENDC)

    input(" " \* 50 + "Press Enter to continue...")

ef search\_visitor():

    print("\n" + " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    print(" " \* 63 + bcolors.YELLOW + "=== SEARCH VISITOR ===" + bcolors.ENDC)

    print("\n" + " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    search\_name = input(" " \* 50 + "Enter name to search: ").strip().lower()

    found = False

    for visitor in visitor\_log:

        if search\_name in visitor["name"].strip().lower():

            print(" " \* 50 + bcolors.GREEN + f"Name: {visitor['name']}"+ bcolors.ENDC)

            print(" " \* 50 + bcolors.GREEN + f"Contact\_number: {visitor['contact\_number']}"+ bcolors.ENDC)

            print(" " \* 50 + bcolors.GREEN + f"Purpose: {visitor['purpose']}"+ bcolors.ENDC)

            print(" " \* 50 + bcolors.GREEN + f"Time\_in: {visitor['time\_in']}"+ bcolors.ENDC)

            print(" " \* 50 + bcolors.GREEN + f"Date: {visitor['date']}"+ bcolors.ENDC)

            print(" " \* 50 + bcolors.GREEN + f"Address: {visitor['address']}"+ bcolors.ENDC)

            print(" " \* 50 + "-" \* 40)

            found = True

    if not found:

            print(" " \* 50 + bcolors.FAIL + "No visitor found with that name." + bcolors.ENDC)

    input(" " \*50 + "Press Enter to Continue..")

def view\_all\_visitors():

    clear\_screen()

    print("\n" + " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    print(" " \* 63 + bcolors.YELLOW + "=== VIEW ALL VISITORS ===" + bcolors.ENDC)

    print("\n" + " " \* 50 + bcolors.PINK + "=" \* 48 + bcolors.ENDC)

    if not visitor\_log:

        print(" " \* 63 + bcolors.FAIL + bcolors.FAIL + "No visitors logged yet." + bcolors.ENDC)

    else:

        print(" " \* 63 + bcolors.GREEN + " === VISITOR LISTS ===")

        for i, visitor in enumerate(visitor\_log, 1):

            print(" " \* 10 + bcolors.YELLOW +

                  f"{i}. Name: {visitor['name']} | Contact\_Number: {visitor['contact\_number']} | "

                  f"Purpose: {visitor['purpose']} | Time In: {visitor['time\_in']} | "

                  f"Date: {visitor['date']} | Address: {visitor['address']}" + bcolors.ENDC)

    input(" " \* 50 + "Press Enter to continue...")

load\_visitors\_from\_file()

while True:

    clear\_screen()

    header\_art()

    show\_menu()

    choice = input("\n" + " " \* 53 + bcolors.BOLD + "Choose an option (1-4): " + bcolors.ENDC)

    if choice == '1':

        clear\_screen()

        log\_in\_visitor()

    elif choice == '2':

        clear\_screen()

        search\_visitor()

    elif choice == '3':

        clear\_screen()

        view\_all\_visitors()

    elif choice == '4':

        print(" " \* 63 + bcolors.YELLOW+ "Thank You! Come Again!" + bcolors.ENDC)

        print(" " \* 63 + bcolors.YELLOW + "EXITING THE SYSTEM" + bcolors.ENDC)

        break

    else:

        print(" " \* 53 + bcolors.FAIL + "Invalid choice. Please try again." + bcolors.ENDC)

        input(" " \* 53 + bcolors.FAIL + "Press Enter to continue..." + bcolors.ENDC)

**IV. Implementation Feasibility**

### **A.** Technical Feasibility

### Programming Language: This group used Python Language 3.x to make a system.

### **Environment:** It works on Windows, macOS, and Linux as long as Python is installed.

### **Libraries Used:** No need for extra libraries and it can use built-in ones like datetime for time and os for clearing the screen.

### **Design Simplicity:** The program runs in the console with styled text(like ASCII headers and colored text). It works offline and doesn’t need the internet or a database. And it is only TExt-User Interface.

### 

### 

### 

1. **Education and Operational Feasibility**

* **Target Developers:** First to second-year BSIT students learning Python and data structures.
* **User-Friendliness:** Clear menu driven interface with color prompts and validations for full name, contact number, and address input.
* **Training Support:** No advance knowledge required to operate the system. A basic user guide or help command can be included.
* **Integration:** Although stand alone, future version can be enhance into a GUI app or web based system on the same logic.

1. **Limitations**

* No real-time updates or multi-user access .
* Manual Input all visitor details(including date and time, which are automatic) are manually entered by the user.
* Only shows simple text results, no charts or visuals.

# IV. Conclusion

The Offline Visitor Log Management System is designed to help our school security. This visitor log-in is to ensure that our visitor information is secured. Instead of manually write information into a logbook, visitors can simply type their details into the system through the console interface. This streamlines the logging process, reduces paper use, making it more efficient and organized. This approach offers a faster, neater, and more secure way of handling visitor records without needing internet or expensive software. This is a big help for our school, which is the University of Science and Technology of Southern Philippines(USTP).It is ideal for academic use, student research, and prototyping, while remaining aligned with the university’s commitment to digital innovation and efficient resource management.

### STS IMPACT ASSESSMENT REPORT

**Introduction**

Today, many schools are starting to use digital tools to make their daily tasks easier, faster, and more eco-friendly. One task that is often forgotten is the logging of visitors. Most schools still use paper logbooks, which can easily get messy, lost, or damaged, and it’s easy to make mistakes when writing by hand. At the University of Science and Technology of Southern Philippines (USTP) Oroquieta Campus, keeping visitor records safe and accurate is important for safety, event planning, and working with the community. To help with this, our team made an **Offline Visitor Log Management System.** This is a simple and easy-to-use program made with Python that helps record visitor information on a computer—no internet or expensive software needed.

### Societal, Ethical, and Environmental Implications

When creating and using a system in school, it’s important to think about its effect on society, ethics, and the environment. Even small systems like our **Offline Visitor Log Management System** can make a positive impact when used responsibly.

**I. Societal Implications**

·**1. **Better Access to Visitor Records.** Our system** helps school staff and security teams quickly check visitor logs. This means there is no need to flip through messy logbooks, making the process more organized and accessible for everyone.

**2.**Stronger Campus Safety and Awareness**.** Since the system records visitor names, contact numbers,pupose of visiting, and time and date is automatically generated.It helps keep the school community safe. Students and teachers feel more secure knowing there’s a reliable record of who enters the school.

**3.**Support for Learning and Research****  
Since it’s built using Python and basic data structures, our system is great for IT students who are still learning programming. It serves as a real example of how code can solve community problems.

**II. Ethical Implications**

**1.Data Privacy and Security.** The system stores visitor information in a text file and works fully offline. This limits the risk of online data connection. It’s important, though, to make sure only trusted people can access the saved files.

**2. Fair Use and No Discrimination.** The system treats all visitors equally. Everyone’s information is recorded the same way, without bias. It promotes fairness in how guests are logged and viewed.

**3. Responsibility of Use.** Users are responsible for entering correct details. Having a system that keeps logs makes people more careful and honest, which builds a responsible environment for both staff and visitors.

**III. Environmental Implications**

**1. Less Paper Use.** Since everything is saved digitally, there’s no need to use paper logbooks. This helps reduce waste and supports the university’s goal of being more eco-friendly.

**2. Lower Resource Consumption.** Because the system is simple and runs offlline doesn’t need servers, constant internet, or large devices—just a basic computer with Python installed. This saves electricity and equipment

3. **Supports Sustainable Practices.** By using tech instead of paper and pens, the school makes small but meaningful steps toward sustainability. Even basic systems can help the school become more environment-friendly.

### Justification of the Technological Solution’s Benefits

**The Offline Visitor Log Management System** is more than just a regular school project. It shows how a simple tool can support digital change in schools. Some of its main benefits are:

1. **Faster visitor logging than writing in a notebook.** The system makes the process of recording visitors quicker than using a paper logbook, saving time for both staff and guests.

2. **Fewer mistakes, since inputs are checked and saved clearly.** It helps reduce errors because it checks if the information is complete and saves it in a clear, organized format.

3. **Better organization with digital records that are easy to search.** All visitor data is stored digitally, making it easier to find and review records whenever needed.

4. **Good for beginner programmers who want to build real-life tools.** The project is simple but realistic, making it a great way for new programmers to apply their skills to something useful.

**5. Safer data handling with offline access and local storage.** Since it works without the internet and saves data locally, there’s less risk of hacking or data leaks.

**6. Supports green practices by reducing paper use.** It helps protect the environment by removing the need for paper logbooks and reducing physical waste.

**Conclusion:**

From an STS point of view, the **Offline Visitor Log Management System** helps improve safety, responsibility, and environmental care. It may be a small project, but it shows how technology can solve real problems in schools. It also supports USTP Oroquieta’s mission to use digital tools for learning, management, and community service in a smart and sustainable way. The Offline Visitor Log Management System is a helpful and practical tool that improves how visitor information is recorded at USTP Oroquieta. It makes logging faster, reduces errors, and keeps records organized and easy to find. Because it works offline and saves data locally, it also helps protect visitor privacy and data security. This system supports environmental goals by cutting down on paper use and is a great example of how simple technology can make school operations better. Overall, it shows how digital tools can support a safer, more efficient, and greener campus.