

---

# Helpline Operator System

## Project Report

2021-7-09

Made By:

- 1) Shoray Singhal (LCI2020037)
- 2) Ayusman Samal (LCS2020024)
- 3) Udbhav Misra (LCI2020019)
- 4) Aman Garg (LCI2020031)

Submitted to:

**Saurabh Srivastava**

# Introduction

---

This report begins with a brief introduction to this project and the different stages of the design and the development of the **HELPLINE SYSTEM**. Before a detailed discussion, first of all, we would like to give an overall idea about this project.

## TITLE OF THE PROJECT

### **“HELPLINE OPERATOR SYSTEM”**

In our **Helpline Operator System** data is processed and the results are used for running the essential services as efficiently as possible. This system maintains the list of available Essential Service Units and distributes them according to the level of urgency. It also maintains the data provided by a user who is typically a helpline operator and is receiving an emergency call. We also have tried to include some level of security in our project by password protecting the user's data and storing the password in the form of SHA-512 hash.

## **Why Computerized Helpline System?**

Using this program, we will store the information more accurately, efficiently, and securely rather than the manually calculated and manual records information.

The information can be extracted or edited at the ease of the user, without any hassle unlike manually kept records.

A good helpline system will process input data faster and reduce clerical time while making certain:

- That output is correct.
- The System is secure and reliable.

## **Main Goal of The Project**

---

The main goal of the project is to obtain complete and correct information, to avoid confusion and data mismanagement.

The process is done as:

- The attendee logs into the system and attends the call.
- Keeps track of the number of Essential Service personnel available.
- Produces regular reports giving the total information required as management needs to know details like – the department/s

needed, number of units needed, location of the person in distress, and number of people on site.

- Change the password in case the old one is Compromised.

## Drawbacks with Existing System

---

- Time Consuming.
- Less Data Security.
- More paperwork has to be done.
- No report generation.
- Low efficiency and redundancy

## Advantages of the Proposed System

---

- Less time-consuming.
- Less paperwork.
- Reports Generation can be done easily.
- Queries can be done easily.
- Easy Redundancy.
- It is fully user-friendly.
- Faster Access to information.
- Log data can be easily manipulated using NumPy or Pandas.

## Why using C++

---

Developments in software technology continue to be dynamic. New tools and techniques are announced in quick succession. This has forced the software engineers and industry to continuously look for new approaches to software design and development. They are becoming more and more critical given the increasing complexity of software systems and the highly competitive nature of the industry.

These rapid advances appear to have created a situation of crisis within the industry. The following issues need to be addressed to face this crisis:

- How to represent real-life entities of problems in system design?
- How to design systems with open interfaces?
- How to ensure reusability and extensibility of modules
- How to develop modules that are tolerant to any changes in the future?
- How to improve software productivity and decrease software cost?
- How to improve the quality of service?
- How to manage time schedules?
- How to industrialize the software development process?

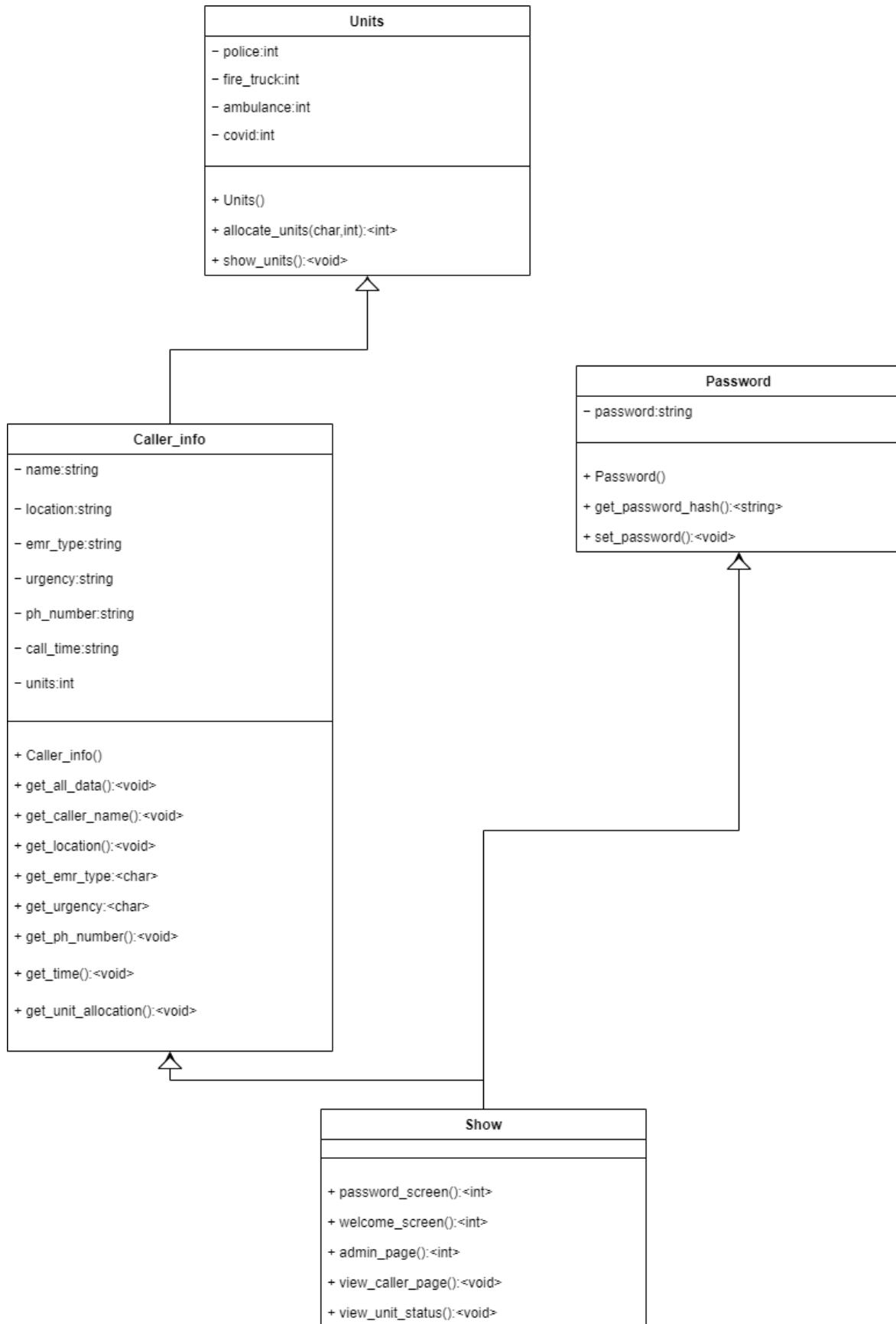
## **Benefits of OOP**

Object-oriented Programming contributes to the solution of many problems associated with the development quality of software products. The new technology promises greater programmer productivity, a better quality of software, and lesser maintenance cost. The principal advantages are:

- Through inheritance, we can eliminate redundant code and extend the use of existing classes.
  - We can build programs from the standard working modules that communicate with one another rather than having to start writing the code from scratch. This leads to saving development time and higher productivity.
  - The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program
  - It is possible to have map objects in the problem domain to those in the program.
  - It is easy to partition the work into a project based on objects.
  - Object-oriented systems can be easily upgraded from small to large systems.
  - Message passing techniques for communication between objects make the interface descriptions with external systems much simpler.
- Software complexity can be easily managed.

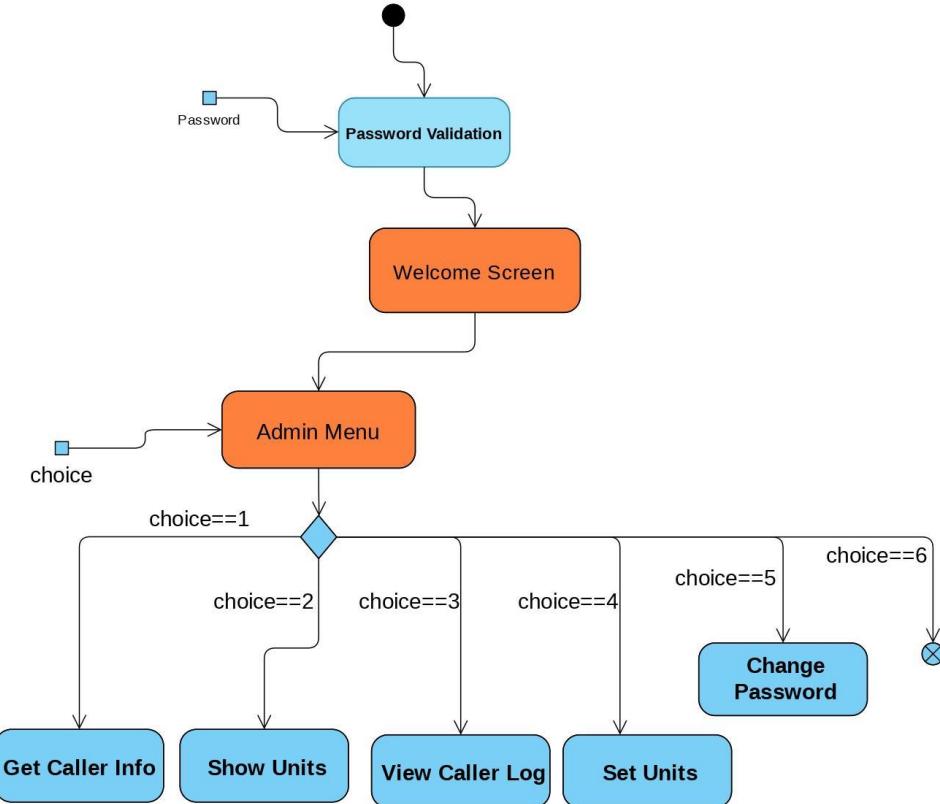
# System Analysis:

## Class Diagram

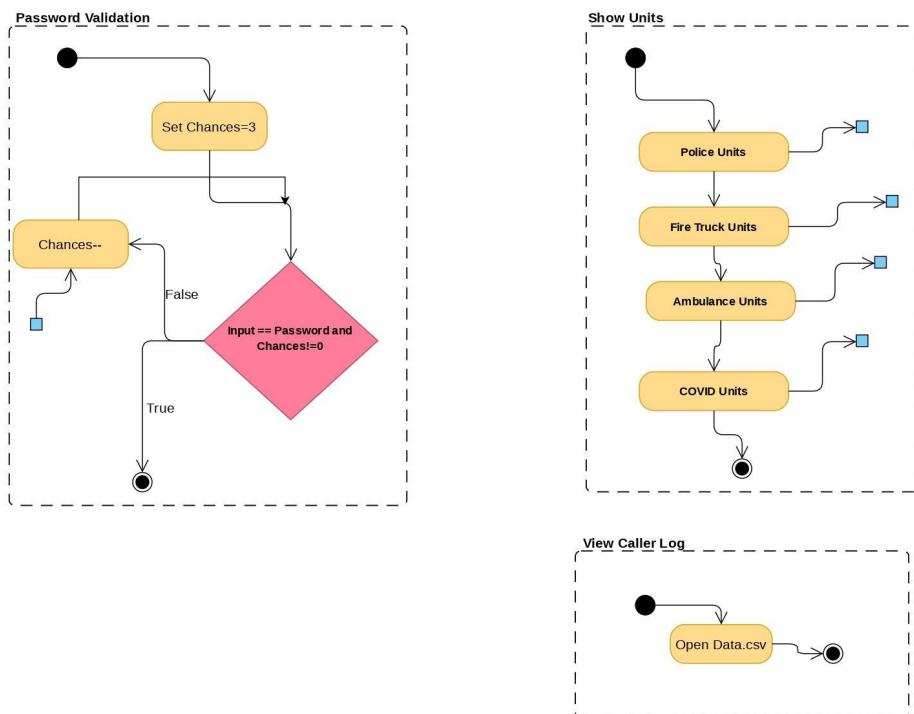


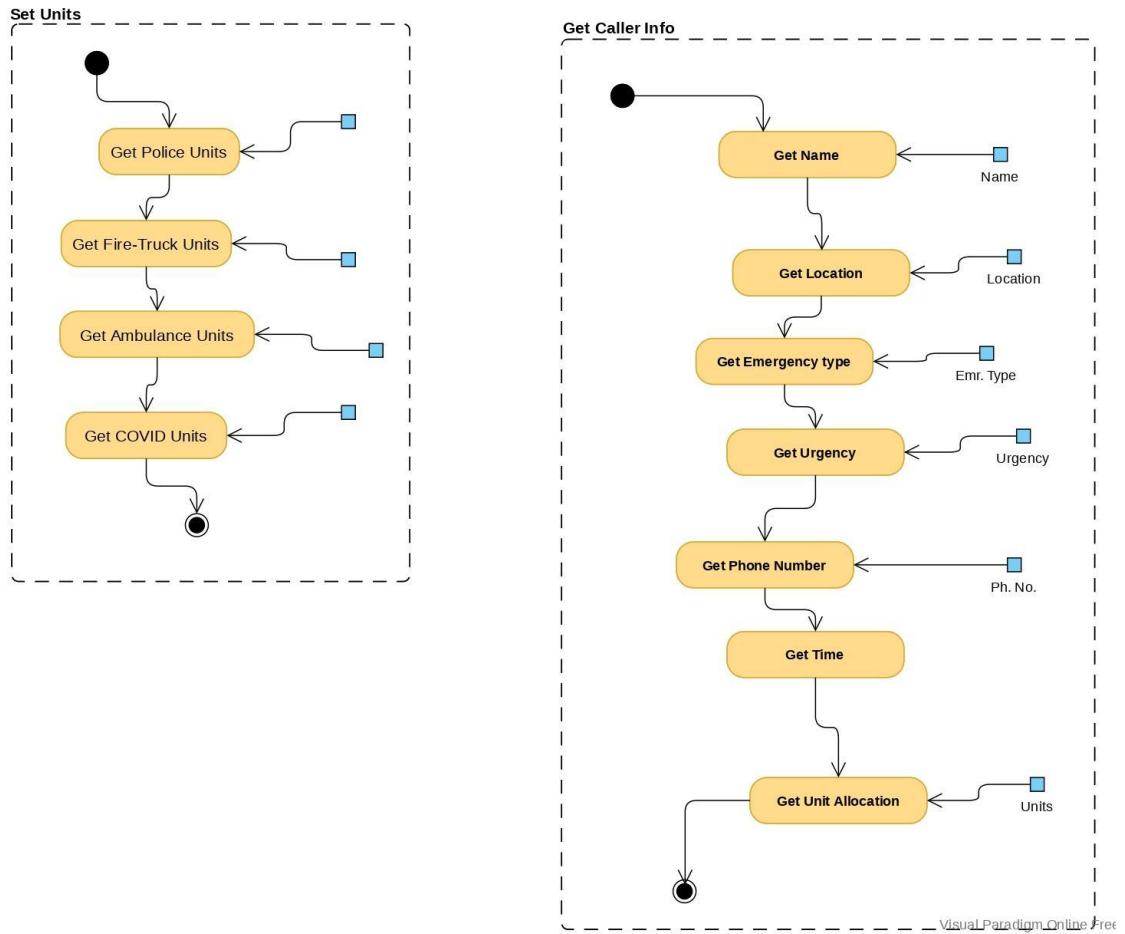
# Activity Diagrams

Overall Functionality:

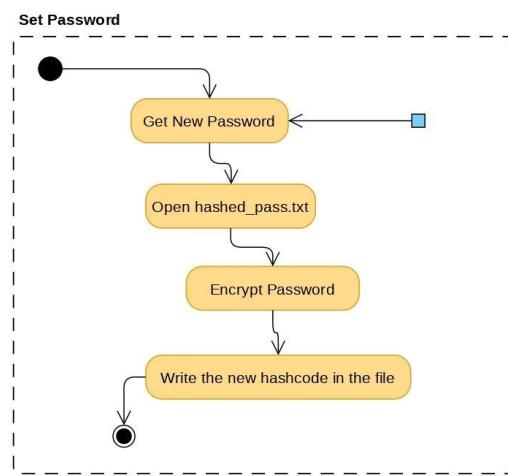


Activities:

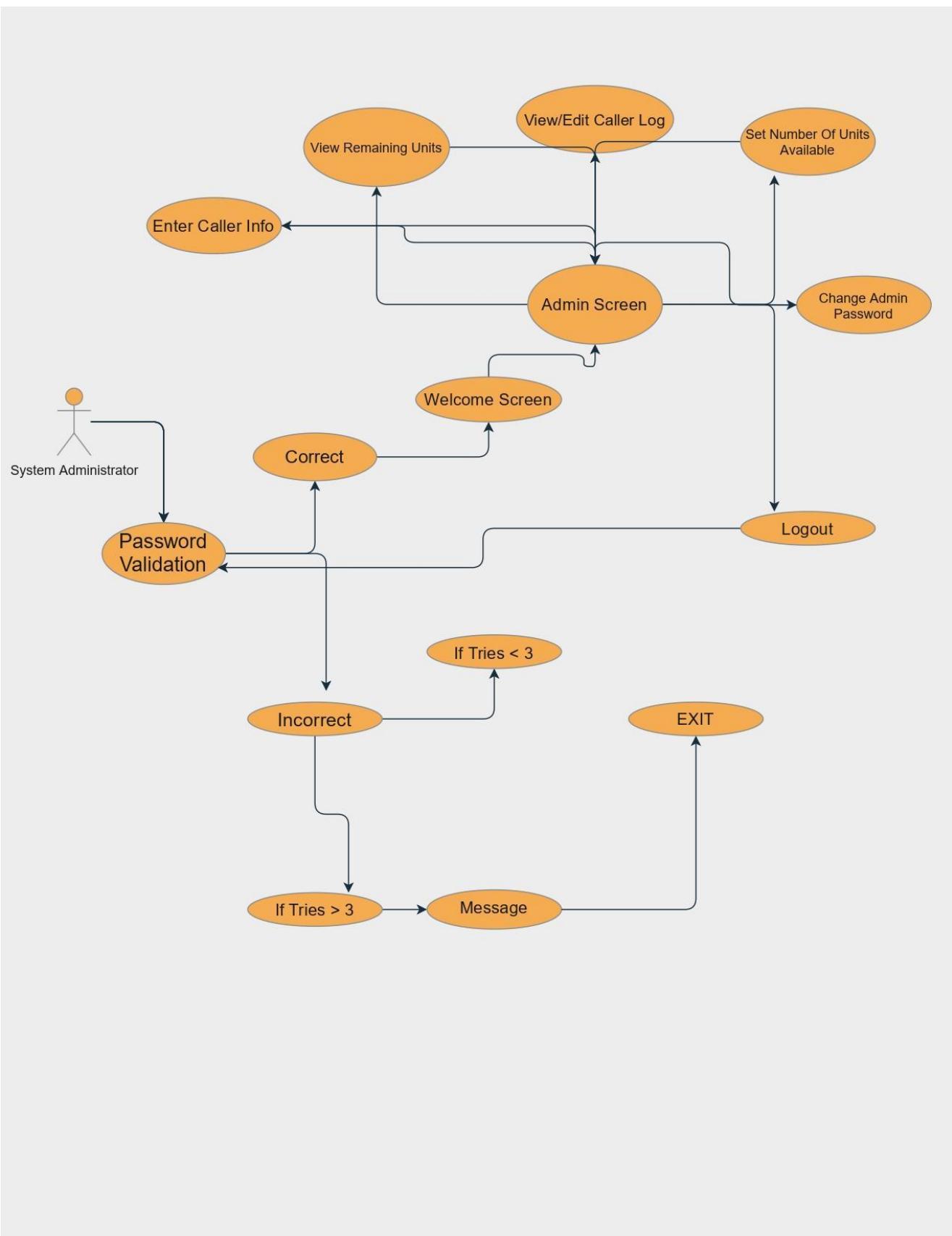




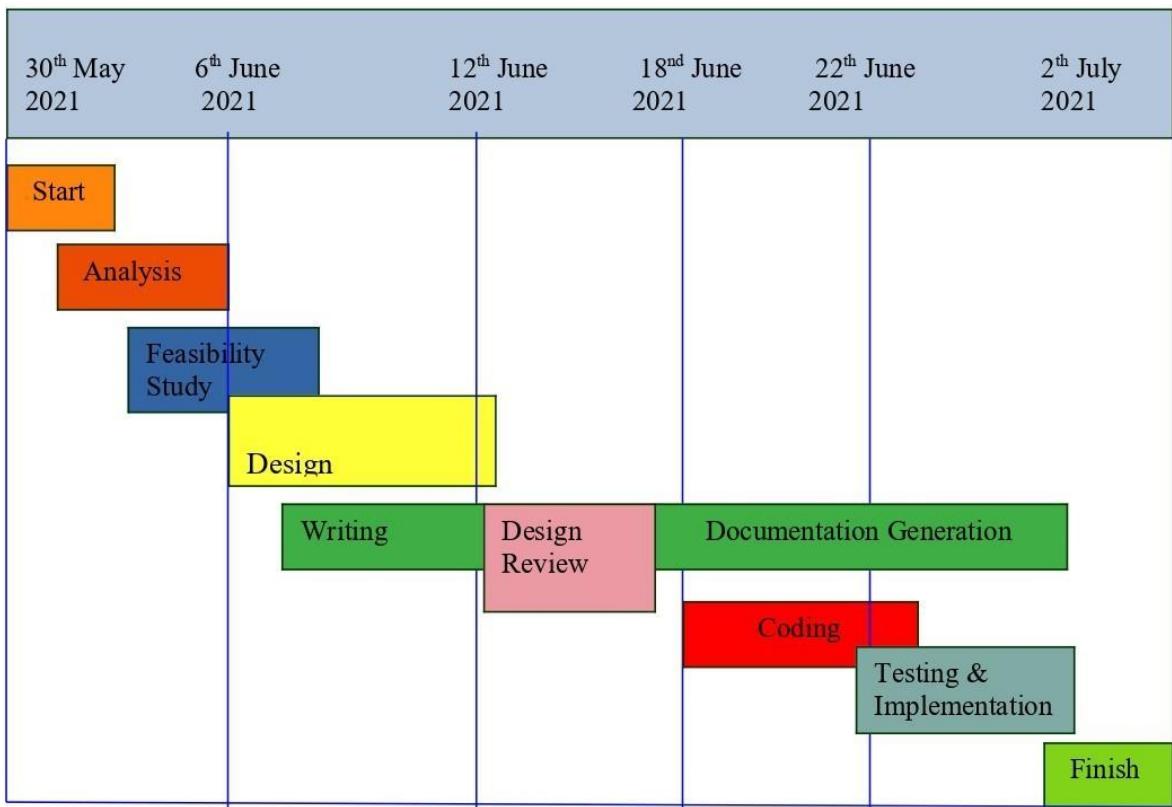
Visual Paradigm Online Free



# Use Case Diagram



# Gantt Diagram



## System Design

Our objectives with this system are as follows :

1. Making it easy for government operators to file an emergency.
2. To make sure that the necessary help reaches the people who need it as fast as possible.
3. Even though we have the technology, a lot of reports are still made on paper. This software will eliminate the use of paper.
4. To make it easier to find reports as it is digitally stored.
5. Less space usage.
6. To make it less likely that the reports get lost or destroyed from an accident.

Major steps in system design:

1. Specification of various emergencies.
2. Specification of the severity level of emergency.
3. Specification of system output.
4. Development of system flowcharts.
5. Implementing the system.

# Program Structure

We are developing an emergency helpline system to provide a computerized facility to the helpline operator and store the records digitally.

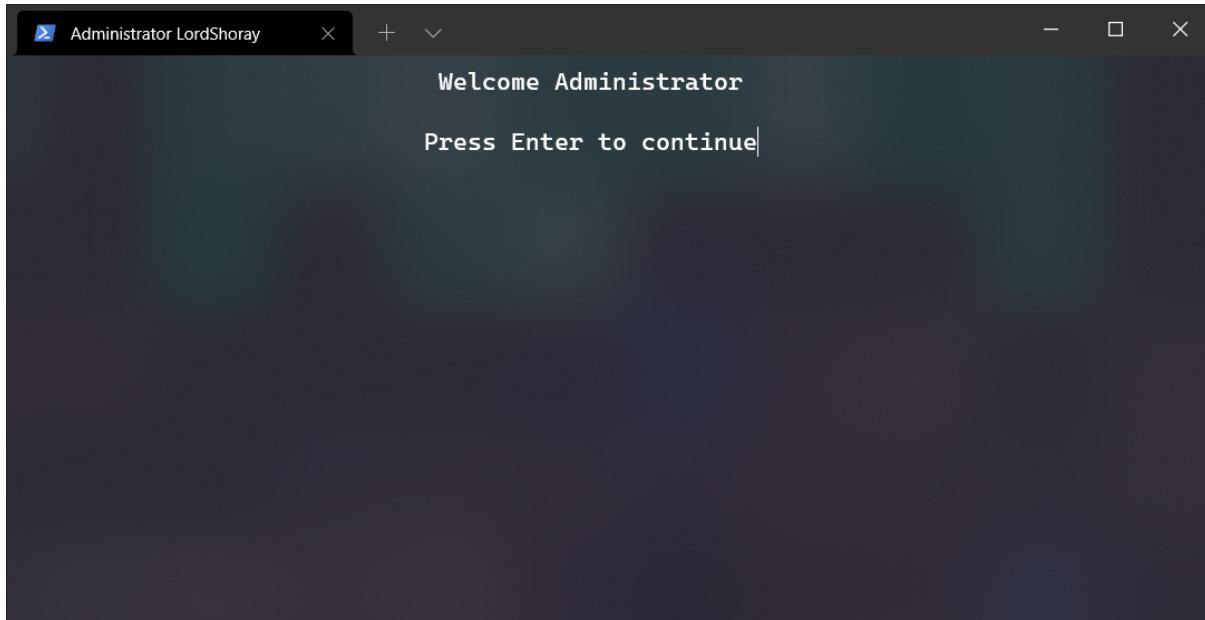
## Naming Conventions

Objects	Naming Conventions
Header Files	sha512.h, caller_info.h, show.h, csvfile.h, password.h, units.h, iostream.h, iomanip.h, stdlib.h, ctime.h, string.h, fstream.h, sstream.h, cstring.h,
Classes	Caller_info, Password, Show, Units
Functions	Centerstring(); Caller_info(); get_all_data(); get_caller_name(); get_location(); get_emr_type(); get_urgency(); get_ph_number(); get_time(); get_unit_allocation(); runner(); main(); Password(); get_password_hash(); set_password(); password_screen(); welcome_screen(); admin_page(); view_caller_page(); view_unit_status() sha512(string); Units(); set_units(); allocate_units(char, int); show_units()
Data files	hashed_pass.txt, Data.csv
Program variables	t, name, location, emr_type, urgency, ph_number, call_time, units, password, chances, choice, police, fire_truck, ambulance, covid
Program files	Helpline.cpp, sha512.cpp, caller_info.cpp, show.cpp, password.cpp, units.cpp

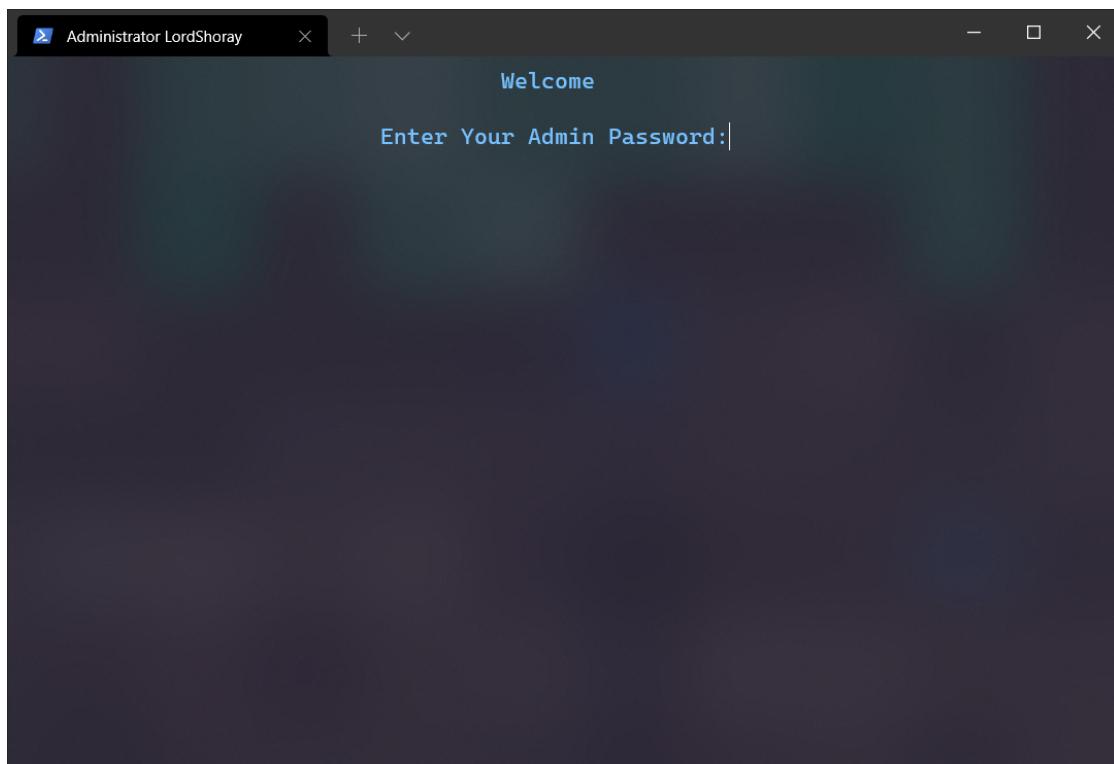
**In this system, the passwords are hashed and stored in `hashed_pass.txt` to ensure proper security**

## Working of the System

Upon execution of the program, the program asks for the administrator password and is given 3 chances.

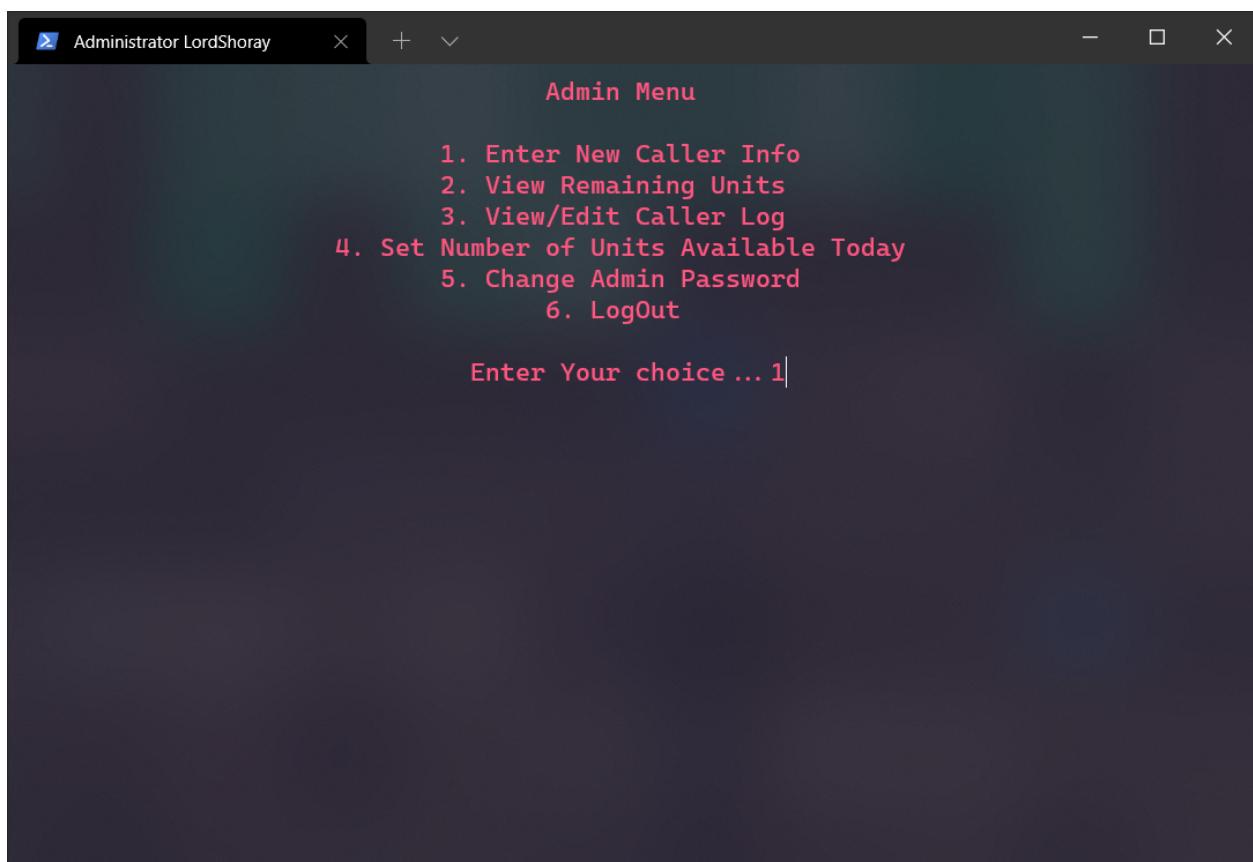


After entering a valid password, the following welcome screen appears

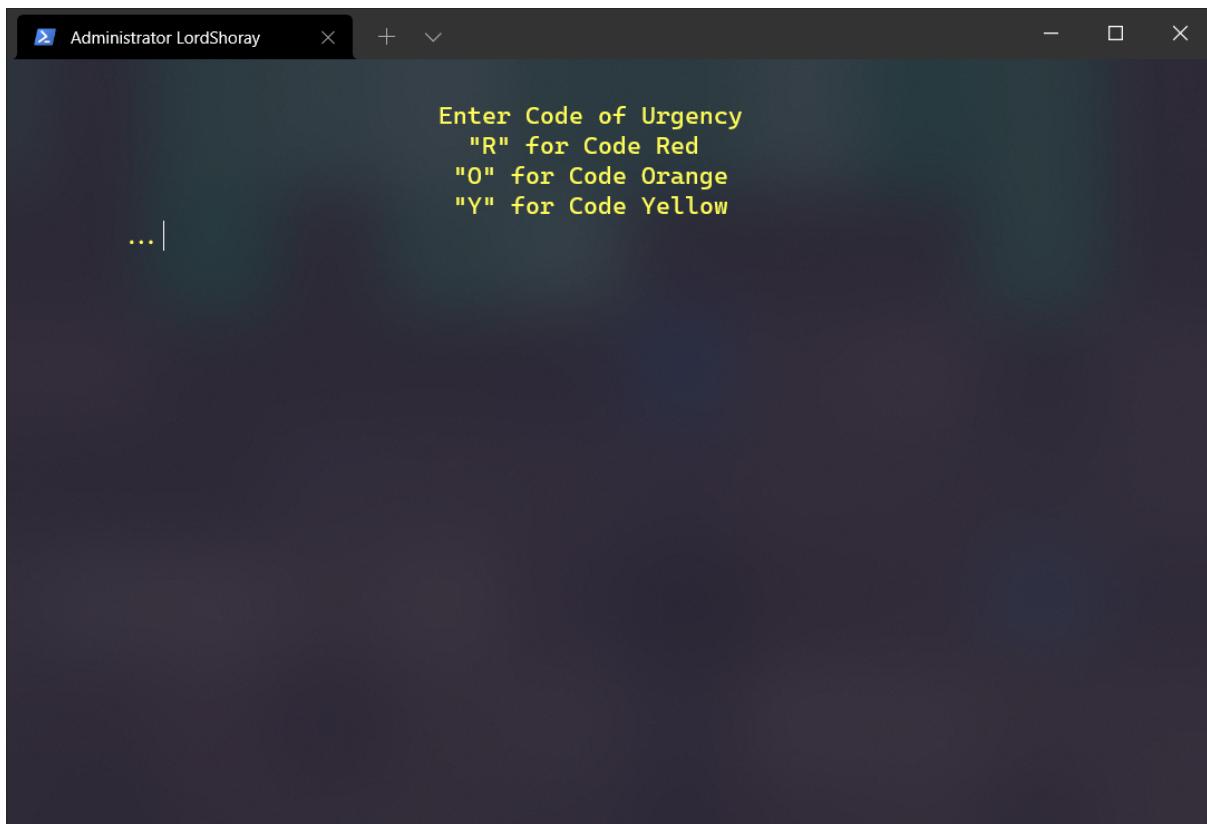
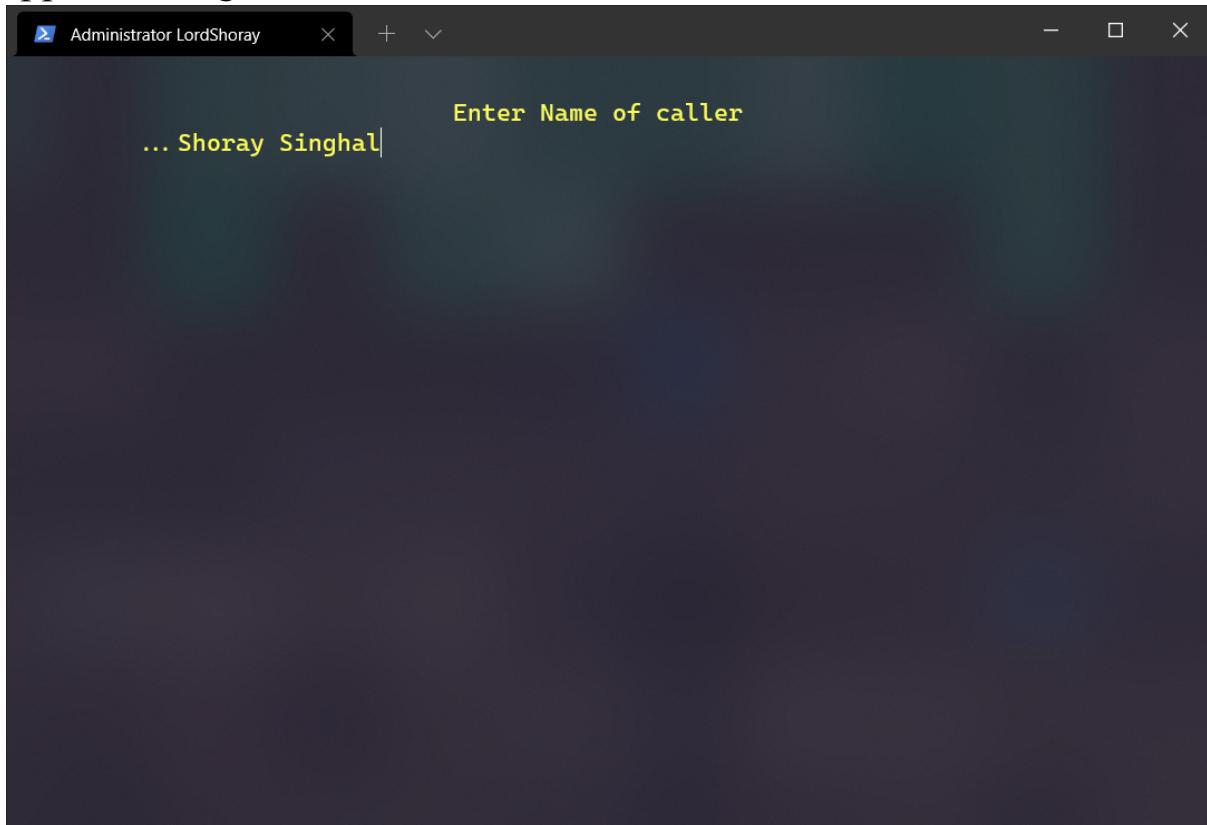


Then the following options are displayed:

- 1. Enter new caller info:** To enter new caller information.
- 2. View Emergency Unit Status:** To view the current status of the remaining emergency units.
- 3. View Previous Callers:** To view a log of previous callers.
- 4. Set Units Available:** To Set the number of each unit type available that day.
- 5. Change Admin Password:** To change the administrator password.
- 6. Logout:** To log out of the program.



On selection of **Option 1**, several screens of the following type appear asking for the caller data:



On selection of **Option 2**, the following table appears:

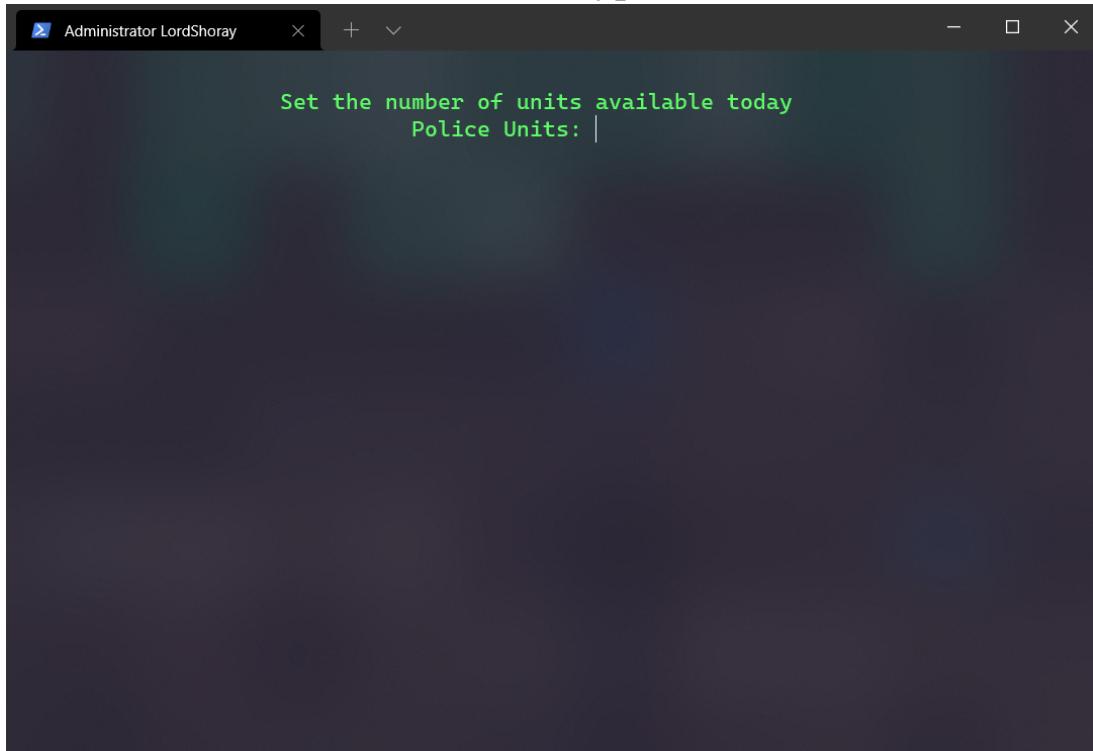
Sl. No:	Type	Units
1	Police	99
2	Fire	15
3	Medical	50
4	COVID	25

Press Enter to go to admin menu ... |

On selection of **Option 3**, a following appears and a CSV file is opened in a new window:

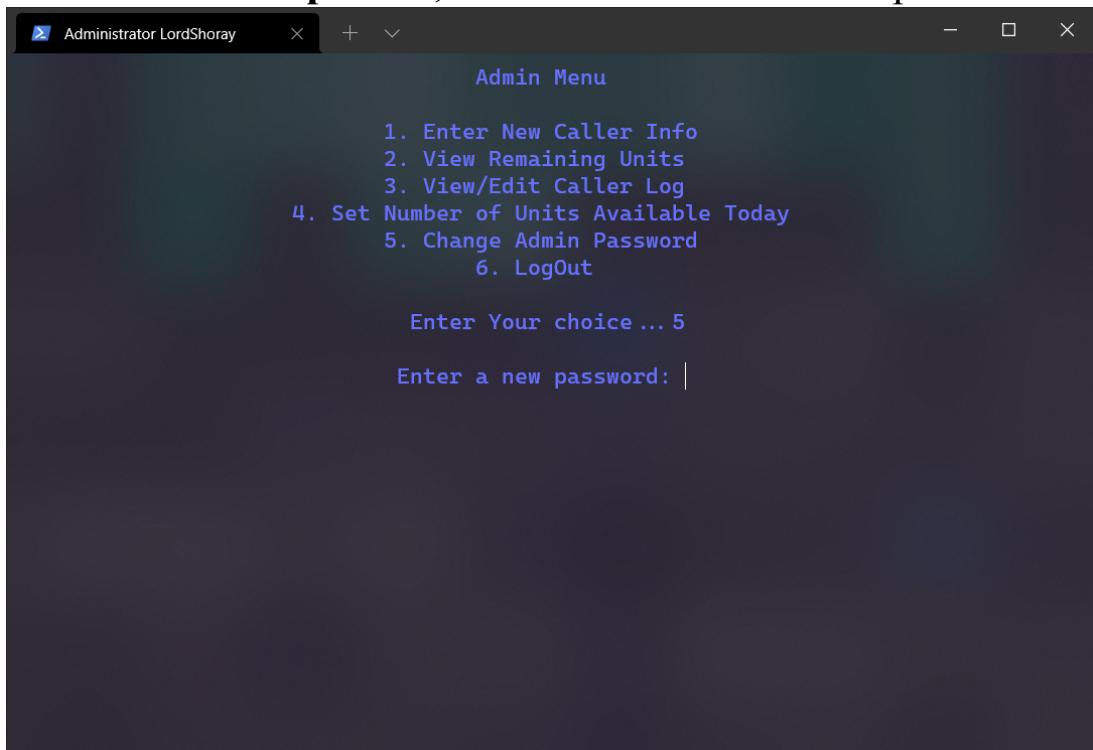
Caller Information		
Log file is opening ...		
Log file closed ...		
Press Enter to go to admin menu ...		

On selection of **Option 4**, the following appears asking for the number of units available of each type:



```
Administrator LordShoray      -  □  ×
Set the number of units available today
Police Units: |
```

On selection of **Option 5**, the screen asks for a new password:



```
Administrator LordShoray      -  □  ×
Admin Menu
1. Enter New Caller Info
2. View Remaining Units
3. View/Edit Caller Log
4. Set Number of Units Available Today
5. Change Admin Password
6. LogOut
Enter Your choice ... 5
Enter a new password: |
```

When **Option 6** is selected the program logs out and takes us back to the Password Screen.

The program can be **EXITED** at any time by using ***CTRL+C***

## Source Code

[Here](#)

saved on GitHub

## Important Functions and Their Purposes

The list of all user-defined functions (UDFs) and their purpose is listed as follows:

1. **void centerstring(string s, int width):** This function takes in two parameters, a string, and an integer, and outputs the string centered according to the width(integer) provided.
2. **Caller\_info(), Password(), Units():** Constructors
3. **void get\_all\_data():** This function calls other get functions to initiate data input.
4. **void get\_caller\_name():** This function takes the name of the caller as input and stores it in the variable name.
5. **void get\_location():** This function takes the location of the caller and stores it in the variable location.
6. **void get\_emr\_type():** This function takes the keyword corresponding to the emergency type (P for police, F for fire, M for medical, C for COVID) and stores it in the variable emr\_type.
7. **void get\_urgency():** This function takes the keyword corresponding to the emergency level(R for red, O for orange, Y for yellow) and stores it in the variable urgency.
8. **void get\_ph\_number():** This function takes the caller's valid phone number and stores it in the variable ph\_number.

- 9. void get\_time():** This function stores the system time when the data is being filled.
- 10. string get\_password\_hash():** This function opens the file containing the password hash and returns the hash as a string.
- 11. void set\_password():** This function sets the new password by overwriting the old password present in the file with the new password hash.
- 12. int password\_screen():** This function displays the password screen, compares the input password hash with the stored password hash, and responds appropriately.
- 13. int welcome\_screen():** This function displays the welcome screen.
- 14. int admin\_page():** This function displays the administrator menu after the password has been verified.
- 15. void view\_caller\_page():** This function opens the log file.
- 16. void view\_unit\_status():** This function displays the unit status table.
- 17. void set\_units():** This function takes in input from the user about the number of units available in that case and then stores the data appropriately.
- 18. int allocate\_units(char,int):** This function takes in the type of units and the amount to be deducted from the available units and returns a response appropriately.
- 19. show\_units():** This function displays a table of the remaining units.
- 20. string sha512(string input):** This function takes in a string and returns its corresponding sha512 hash.
- 21. void runner():** This function executes the choice logic in the system.
- 22. int main():** This function is the main function of the system and is executed automatically when the system starts and return 0 upon completion.

## Training of people involved with the system

The people who are directly involved with this system must be trained with the following topics:

1. How to turn the program on and off.
2. Familiarity with all the possible options of the program.
3. Familiarity with the administrator password and how to change it.
4. Aware of all the possible malfunctions of the program and how to troubleshoot it.
5. Training with the keyboard so that the operator can use the program quickly and efficiently.

## Conversion Methods

For the conversion from the old pen and paper system of keeping helpline records, the following needs to be done:

1. Converting physical records into digital format.
2. Training the helpline operators to use the helpline program using the training methods mentioned above.

## Testing

The techniques that have been used for testing this system include the following:

**1. Path Testing:**

Each possible path from input to output is traversed once.

**2. Function Testing:**

Each function is tested at least once.

**3. Special Value Testing:** Testing so that no special character causes any errors.

## **System Security Measures**

With this system, the security goals were as follows :

1. To ensure that unauthorized access cannot be gained to the helpline software.
2. To ensure that passwords remain confidential.
3. To ensure that people with insufficient privilege cannot change the password.

With the above goals in mind, the following steps have been taken :

1. The operator panel is password protected.
2. All the passwords are hashed so that clear text passwords cannot be seen.
3. The SHA-512 algorithm is used so that passwords cannot be cracked.
4. The operator can change the password at any given time.