“Prison Surveillance Setup”

A Report submitted in partial fulfilment of the requirement for the course of

Minor Project – II

In third year – Sixth semester of

B-Tech computer science Specialization in

Banking, financial services and insurance.

Under the guidance of

Dr. Bhupesh Kumar Dewangan

Assistant Professor (Senior Scale), SOCS

Department of Informatics.

By

Name	Roll no	Sap Id	Branch
Aditya Sharma	R133218002	500069141	B-tech CSE BFSI
Rishav Singh	R133218035	500071526	B-tech CSE BFSI
Ritul Priya	R133218020	500069262	B-tech CSE BFSI

UNIVERSITY WITH A PURPOSE

SCHOOL OF COMPUTER SCIENCE

UNIVERSITY OF PETROLEUM &ENERGY STUDIES

Bidholi Campus, Energy Acres, Dehradun – 248007

January -2021

Table of contents

Minor II

PROJECT TITLE: “Prison Surveillance Setup”

ABSTRACT

We are structuring a Prison surveillance setup. This is intended to compact the current security setup and minimize the time utilization of data collection in any criminal case or general inquiry. This setup is designed to launch a single platform at the central level for prisoner surveillance. It is equipped with a login constraint and required security credentials for any data modification. There will be different user interface modes according to the authorization of login type. The required data can be obtained in various modes such as Text file, Word file, etc.

The Prison surveillance setup will be a single platform including the data of all crimes or punishment span at various states of a criminal. This consists of a login authorization which will redirect them to the interface where they want to work. Time stamping is recorded for every login in order to record data for the security analysis. This setup is consisting three interface views which are admin, jail, and inquiry. All the interface view will have different accessibility according to the user. For Security Basis only 3 login attempts are available in all types of user interface.

INTRODUCTION

Earlier, collecting data was a long-time process with too much documentation. But, as the era changes the evolution of the technology is taking place day by day and time is a lot of value to play out any action. Until now, there are prisoners set up that consists of data on a state level. We are making a single platform prison surveillance set up that is made for the Central level data where many tasks can be performed.

Firstly, we are making three interfaces in this setup which are 1.Administration 2. Police Inquiry 3. Prison Administration. At the administration level, all access is granted. In the prison administration, the addition of prisoners, attendance of prisoners, inquiry of any prisoner, etc. is provided. In the third interface which is Prison Inquiry here, only limited data will be accessed such as prisoner personal data, Punishment span, total no of crimes, criminal history, etc. There is a total of 3 login attempts provided for all the above three interfaces.

For the security layer, we have given grant access to the interfaces according to their need, if a prison administration makes an entry in the setup and if any wrong entry was made, there is no deletion access given to the prison administration he/she has to send a request letter with the proper documents to make it correct. To avoid the wastage of time during this procedure a rule constraint is added which will sort the requested order within 30 minutes.

Currently, if some crime occurs and the police catch any person or criminal, they can find out whether there is any past criminal record of that person. They can get the data instantly of that state where they are posted but, they have to wait for a long-time interval to get the record of the arrested person from any other states. So, to solve this complication we decided to make a single platform where the data of prisoners and criminal history are recorded and can be fetched easily by the officials when needed.

There will be a time stamping for every successful login and login details will be recorded for future security purpose. A feature is provided with this setup that if any of the interfaces want to get the required document in the hard copy or soft copy, they can select the option of retrieve data.

Evolution in Prison Surveillance Setup

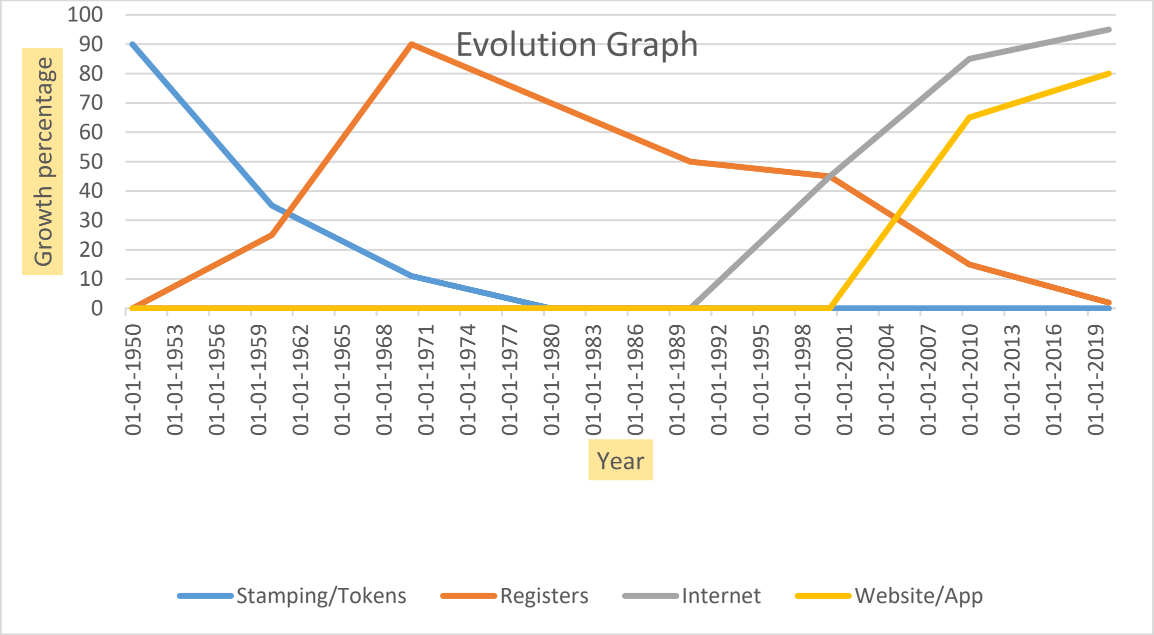


Fig.1:-Evolution in growth of prison setup

The above graph is between development in percentage and years. Basically, this graph shows that how prison management era is increasing year by year in different Sectors

Stamping/Tokens: - As the earlier practice for keeping the record of prisoners Stamping/Token system were used in the 1950's era, this practice was on its peak in 1950's but started decreasing continuously as the technology increased and was completely vanished in 1980's. This technique decreased year by year because if some of the prisoners lost their token then it was so difficult to identify their identity and keep their record track in that era.

Registers: - In 1950's practice of recording data on registers was introduced firstly and later on in 1970's it was on its peaks level and later on in 20's it started decreasing but now also it is being used in few places. It was a great move or practice to overcome the stamping/token practice but it started decreasing because the data can be easily manipulated on the registers and if the register is lost or stolen then there was no backup tool with it.

Internet: - In 1990's the new technology comes in the market that was Internet, organisations and all the sectors started moving on internet because it was very less time consuming and less man power was required to perform any activity, less chance of data stealing, In todays generation also prisons management sectors use internet for storing the data but there was a big side effect of this internet technology because the data of all the private organisation was accessible from a single platform, data stealing, hacking are some of the drawbacks of the internet technology.

Website/App: - From 20's as the Internet technology increased day by day a related technology also discovered which were websites, applications etc. it also minimizes the cybercrime rate which happens in the practice of Internet.

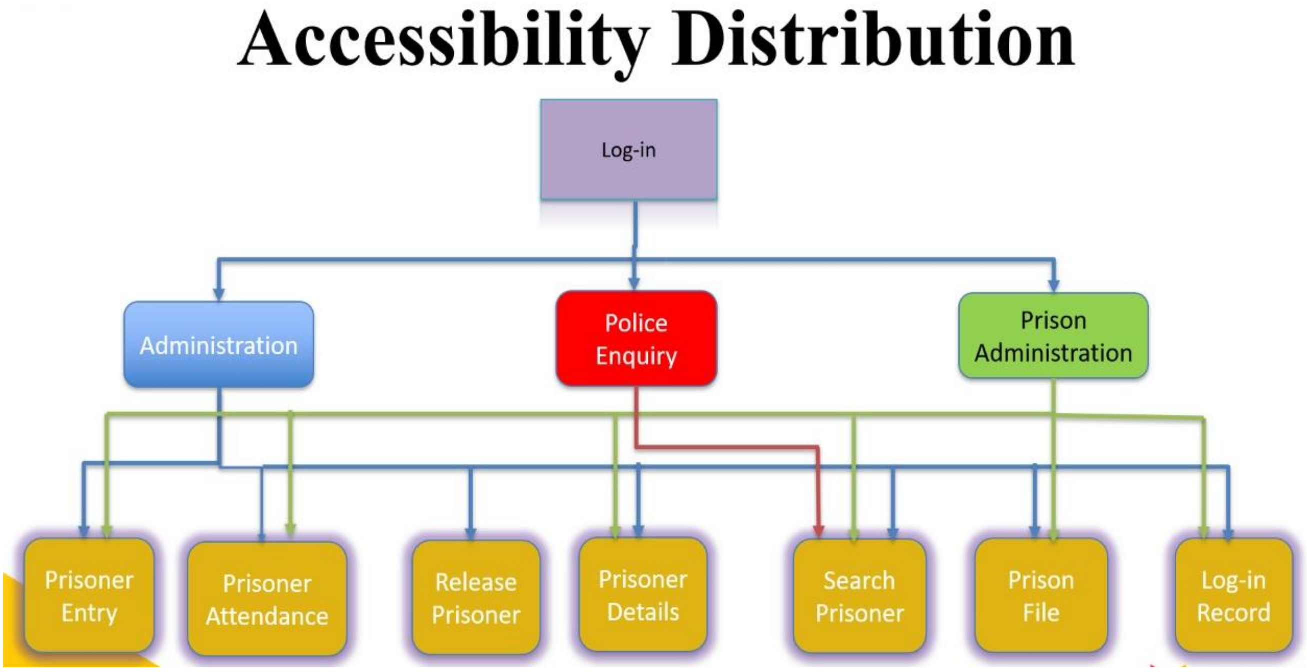


Fig .2: - Accessibility Distribution

The above figure is showing all the accessibility for the different user interfaces

The 3 Interfaces are:

1: - Administration

2: - Prison Administration

3: - Police Inquiry

Accessibility of the Interfaces:-

A. Administration.

- Prisoner Entry
- Prisoner Attendance
- Release data
- Prisoner details
- Search Prisoner
- Prison File
- Log-in Record

B. **Police Inquiry.**

  - Search Prisoner.

C. **Prison Administration.**

  - Prisoner Entry
  - Prisoner Attendance
  - Search Prisoner
  - Prisoner Details.

- **Prisoner Entry:** -This is the function where the prisoner entry will made whether it is new entry or an entry of existing criminal at different prison.
- **Prisoner Details: -**In this function we can fetch the details of the prisoners.
- **Prisoner Attendance: -**In Prisoner attendance function prison administration will maintain the attendance of the prisoner
- **Release Prisoner: -**This is the section which will be accessible only by the Administration because from this function a prisoner will be released from the particular prison.
- **Search Prisoner: -**This function will help all the three interfaces to search the particular prison for a particular state and jail.
- **Prison File:** -In this function you can get the soft copy of the prisoner details in any format.
- **Login Record: -**This is the section where all the login record for a particular time is available.

MOTIVATION

- We are building a prison surveillance setup that is made on a central level to solve the problem faced by the government bodies in the present time. This is an advanced level of currently available setups that involves many sectors of learning while making this framework.
- As per research, we have found that the time taken to gather information of any criminal or suspected person from different states is too long, by making this central level system we will help the government to reduce the time taken to gather the information and solve the case in less time interval. As we are making this project in C++ this project will help us in learning most of the concepts and applying algorithms.

- By creating this project, it will help us in learning the basic understanding of Object-Oriented Programming (OOPs). We will be learning about the software engineering part like software models, their stages, and many more. This is basically for the learning purpose which will enhance our knowledge and skills for future projects.
- Building and implementing a different algorithm for this project.
- It helps in getting wide learning on creating an advanced system or framework.
- Studying and implementing is the best way to learn and understand any concept

PROBLEM STATEMENT

As per the present scenario, retrieving any details of criminal is a time taking process and to resolve this we are building a single platform prison surveillance set up that will help in improving the portability and security of the current system and transforming the current state level setup to a single platform which will be central level accessible.

OBJECTIVE

- Creating a single platform prisoner surveillance system.
    - Creating different interfaces for different types of user.
    - Fetching details of prisoner from files.
    - Applying login credentials inside the system
        - Applying string match algorithm.
        - Recording the real-time login stamp.
    - Integration of all the interfaces in a system.

LITERATURE REVIEW

- **In the paper [1]** named **an object-oriented intelligent tourist advisor system** *by C.H.K. Tsang, Dept. of Comput. & Math., Hong Kong Tech. Coll., Hong Kong, M.H.C. Woo, C. Bloor (School of Computing and Information Systems, University of Sunderland, UK) states* that the paper describes the design and development of an expert system for a tourist information centre. The expert system was built to recommend a suitable travel schedule that satisfies user input constraints such as time period, budget and preferences. Tourist centre officers need to answer similar queries in their day-to-day work and could be replaced by an intelligent tourist advisor system. There is a lot of tourist information, such as activities and places, which can be stored as similar data structures and can be modelled into an object-oriented knowledge-based system.

- **In the paper [2]** named **File Structures: An Object-Oriented Approach with C++** by *Michael J. Folk, Bill Zoellick, Greg Riccardi states* that
  - (a) Introduce the primary designed issues that characterize file structure design.
  - (b) Introduce the notions of file structure literacy and of a conceptual toolkit for file structure design.
  - (c) Introduce file structure concepts dealing with reading, writing, field, record entries, inheritance, encapsulation etc.

- **In the paper [3]** named **Multimedia Information Management in an Object-Oriented Database System** by *Darrell Woelk, Won Kim* states that This paper describes the implementation of the Multimedia Information Manager (MIM) in the ORION object-oriented database system which is operational at MCC. We describe design objectives and implementation techniques that have satisfied the design objectives. Our design objectives include extensibility, flexibility and efficiency in supporting the capture, storage, and presentation of many types of multimedia information.

  The management of multimedia information such as images and audio is becoming an important feature of computer systems. Multimedia information can broaden the bandwidth of communication between the user and the computer system. Although the cost of the hardware required for the capture, storage, and presentation of multimedia data is decreasing every year, the software for effectively managing such information is lacking. Future database systems must provide this capability if we are to be able to share large amounts of multimedia information among many users.

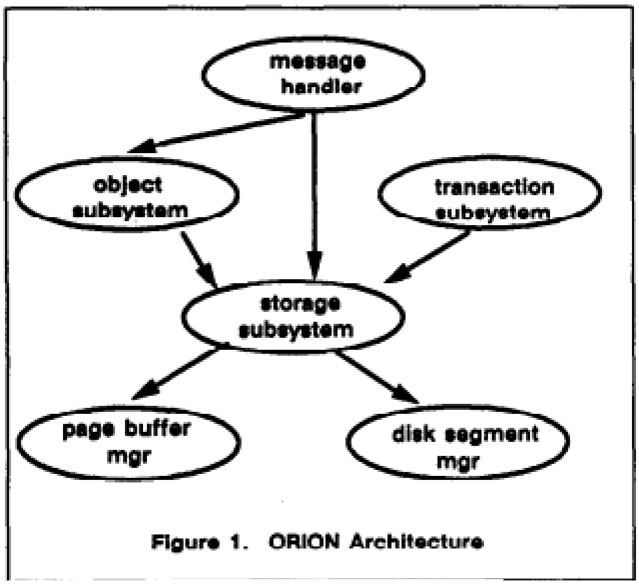


Fig .3 Orion Architecture

- In the paper[4] named **Extensible object-oriented file system** by *Bryan P. Atsatt, Earsh K. Nandkeshwar, Michael J. Seilnacht, Hemantkumar A. Thakkar, George R. Turner, Roger R. Webster* states that An object-oriented file system in an object-oriented operating system includes a file system entity class that is subclassed into a volume, directory and file subclass. These classes encapsulate standard file system properties such as name, creation date, and size, as well as standard operations such as create, open, close, and property accessors. Using object-oriented programming, the class properties and operations can easily be modified and extended. Also provided is a convenient and efficient means for searching through the entities, and collecting heterogeneous sets. Further, a category of notification classes is provided for notifying clients when an entity has changed. Still further, user authentication and protection domains are used to protect against unauthorized access. Finally, a means for working with foreign file systems running under different operating systems is provided.

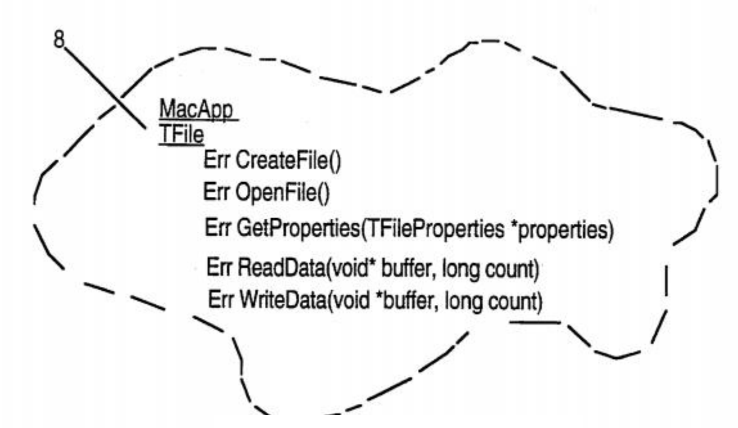

8
MacApp
TFile
Err CreateFile()
Err OpenFile()
Err GetProperties(TFileProperties *properties)
Err ReadData(void* buffer, long count)
Err WriteData(void *buffer, long count)

Fig .4 Extension file system

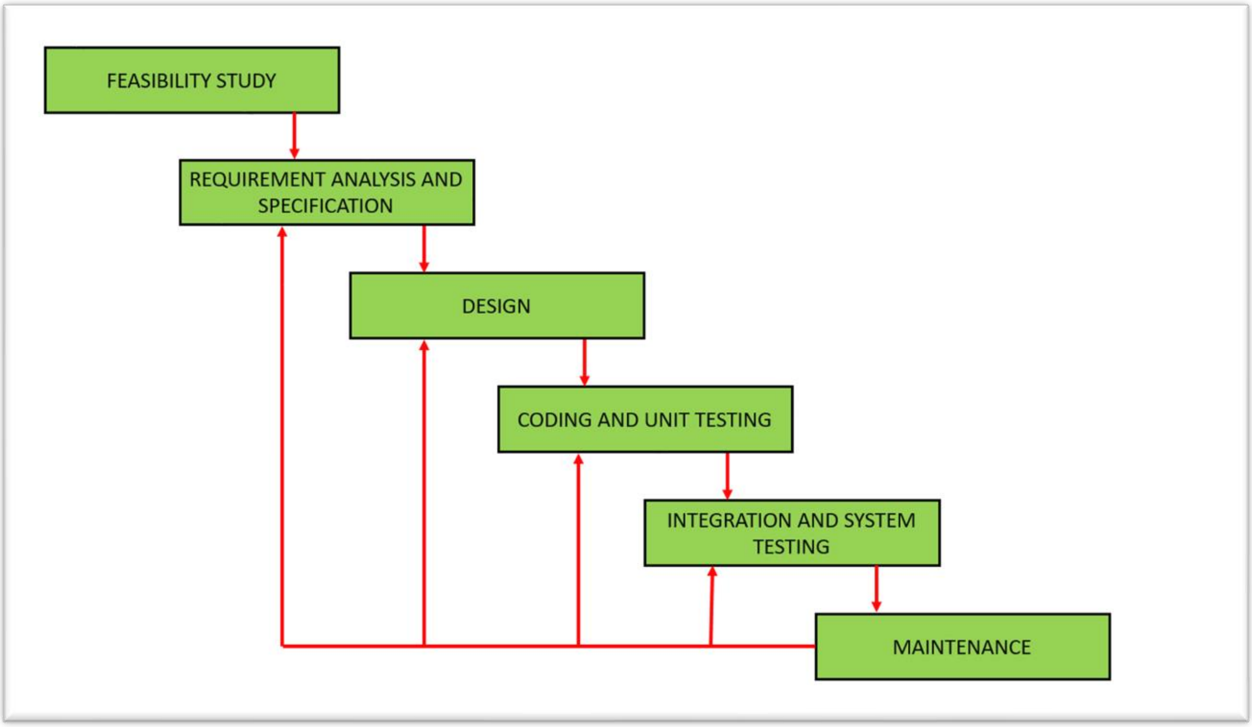
METHODOLOGY

We are using iterative waterfall model for our project. As, this model is easily understandable, and the main profit of this model is that it allows us to go back on the previous phase and change the requirements and do some modification if necessary.

As iterative model has six stages:

- Feasibility study: We have minimised the total lines of code in order to light the workability of the system and to reduce the run time and compilation time.
- Requirement analysis and specification: We have done the research part and basic requirements for this system.
- Design
- Coding and unit testing
- Integration and system testing
- Maintenance

Fig 5. Feedback paths introduced by the iterative waterfall model as shown in the figure.


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

ALGORITHM

Naive String-Matching Algorithm: -

Naive pattern searching is the simplest method among other pattern searching algorithms. It checks for all character of the main string to the pattern. This algorithm is helpful for smaller texts. It does not need any pre-processing phases. We can find substring by checking once for the string. It also does not occupy extra space to perform the operation.

The time complexity of Naive Pattern Search method is O(m*n). The m is the size of pattern and n is the size of the main string.

```
Begin
   patLen := pattern Size
   strLen := string size

   for i := 0 to (strLen - patLen), do
      for j := 0 to patLen, do
         if text[i+j] ≠ pattern[j], then
            break the loop
      done

      if j == patLen, then
         display the position i, as there pattern found
   done
End
```

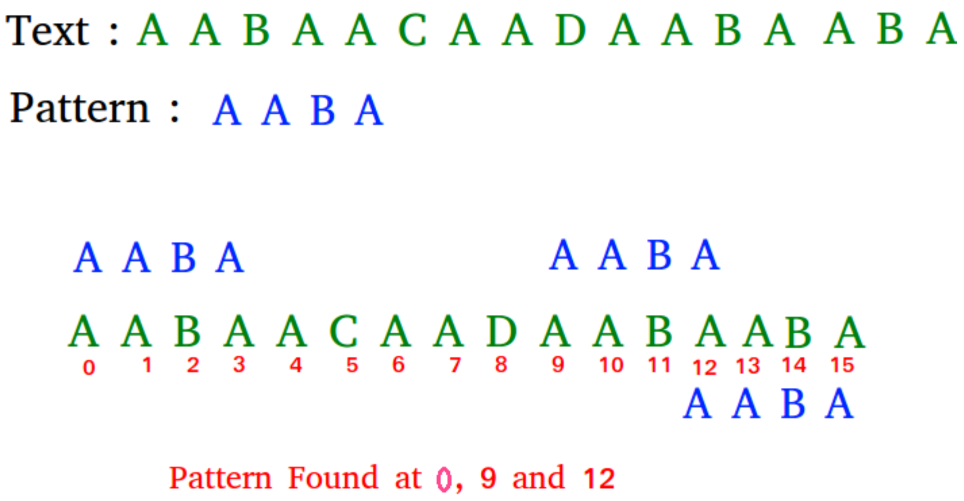
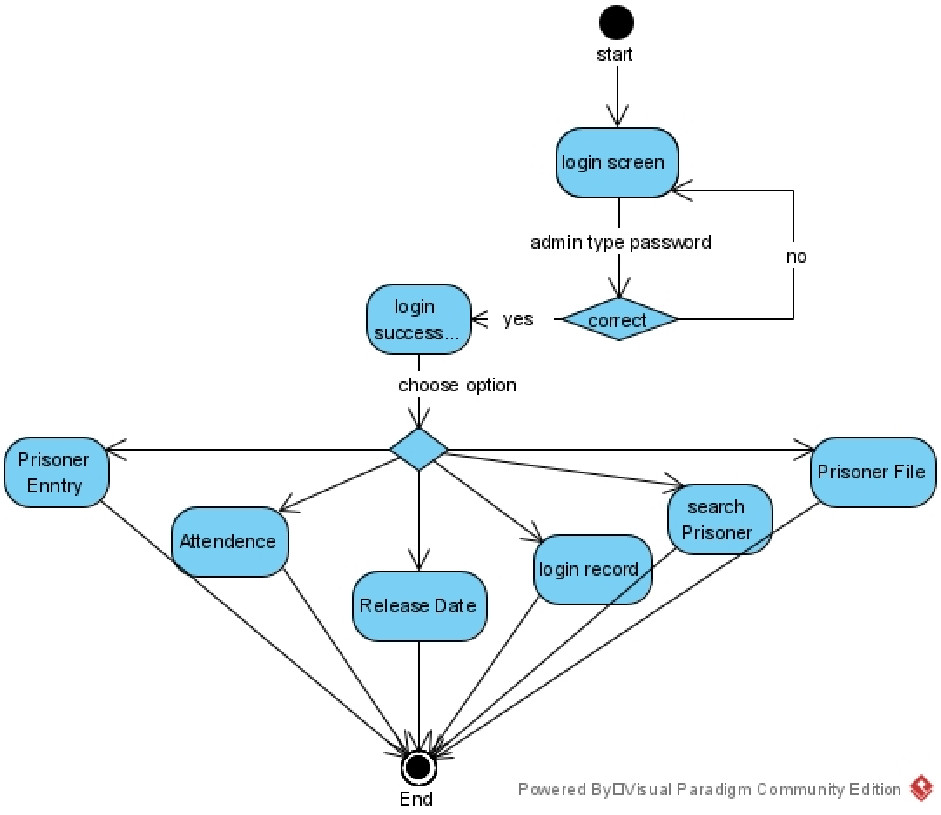

Text : A A B A A C A A D A A B A A B A
Pattern : A A B A
A A B A
A A B A
A A B A A C A A D A A B A A B A
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
A A B A
Pattern Found at 0, 9 and 12

Fig .6 The above picture shows the example of naïve Algorithm

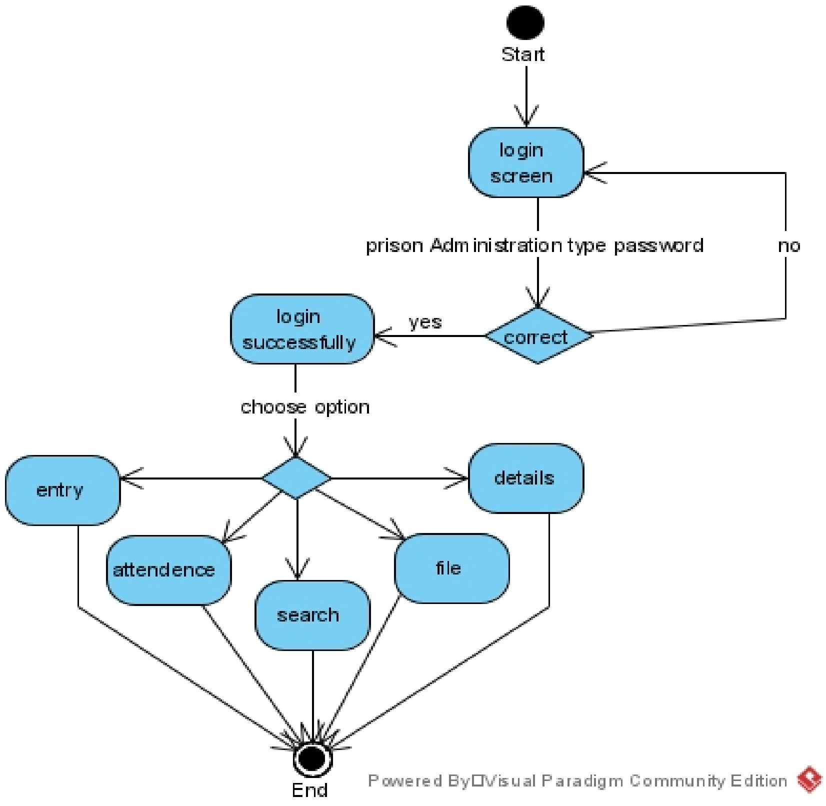
- Header Files:
    - Windows.h
    - String.h
    - fstream
    - ctime

- Functions:
    - System(“cls”) - It clears the output screen of the program.
    - Time()- It returns the current time
    - Date()- It returns the current date
    - System(“pause”)- This is a Windows-specific command, which tells the OS to run the pause program.

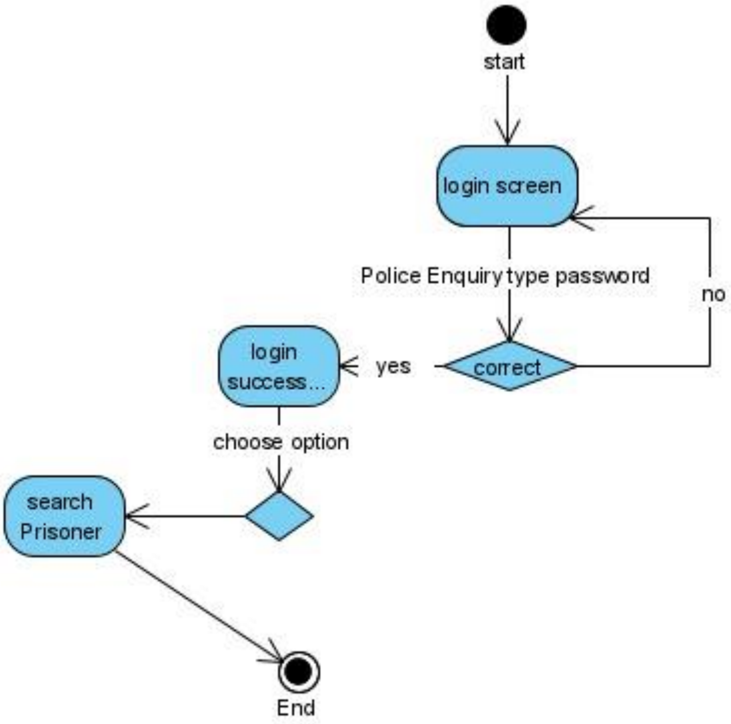
Functioning of Administration Interface



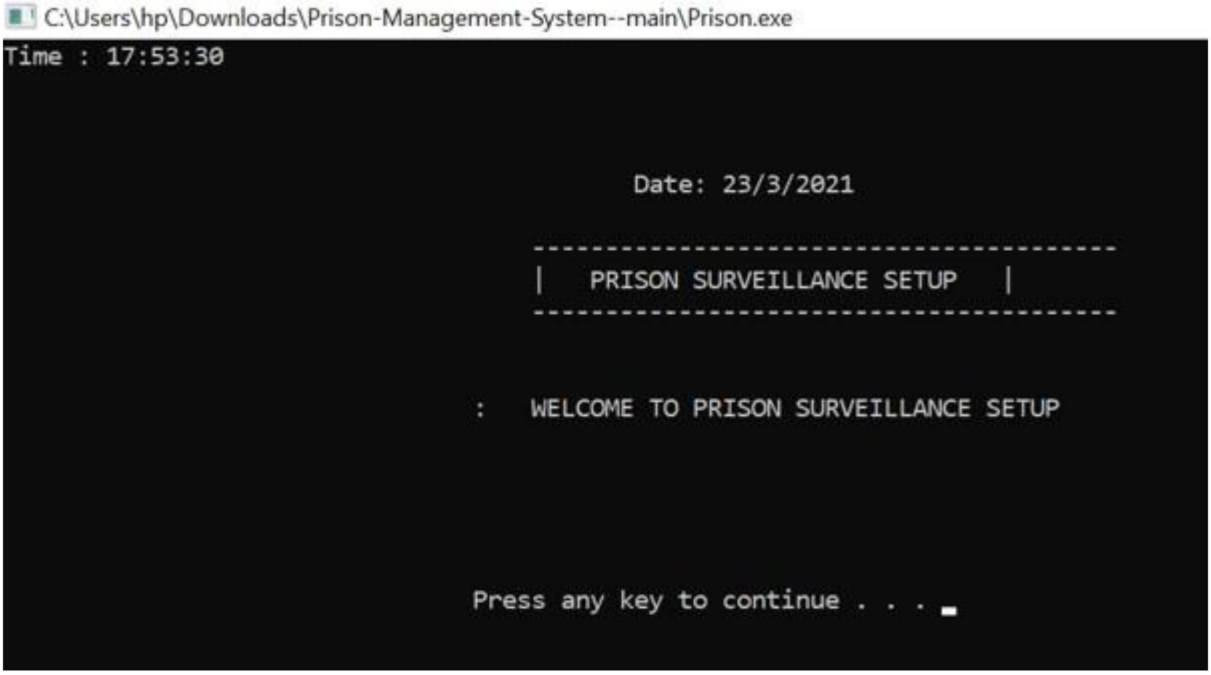
Functioning of Prison Administration Interface



Functioning of Police Inquiry Interface

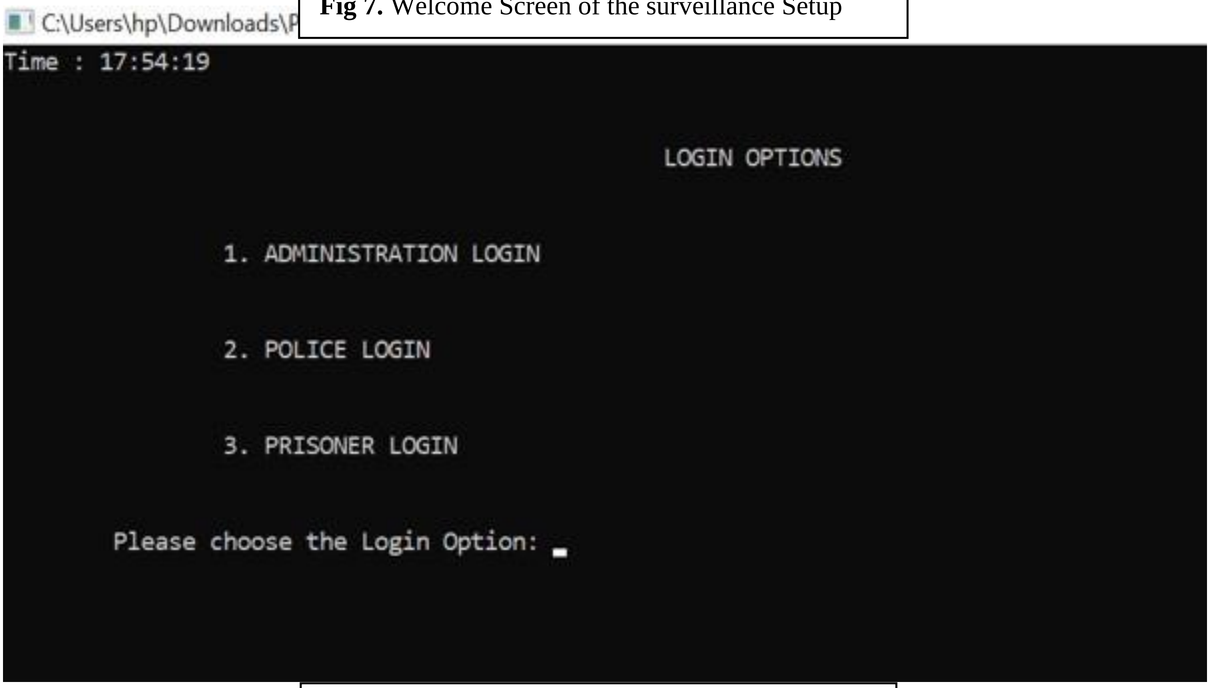


OUTPUT



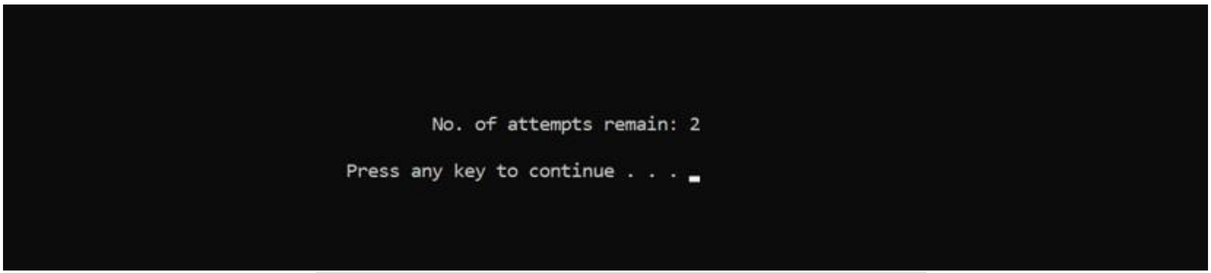
```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:53:30


                                                       Date: 23/3/2021

                                              ---------------------------------------
                                              |    PRISON SURVEILLANCE SETUP    |
                                              ---------------------------------------


                                        :   WELCOME TO PRISON SURVEILLANCE SETUP



                                        Press any key to continue . . .
```

Fig 7. Welcome Screen of the surveillance Setup



```
C:\Users\hp\Downloads\P
Time : 17:54:19

                                                        LOGIN OPTIONS


                  1. ADMINISTRATION LOGIN


                  2. POLICE LOGIN


                  3. PRISONER LOGIN


         Please choose the Login Option:
```

Fig 8. Login Page for different interfaces.



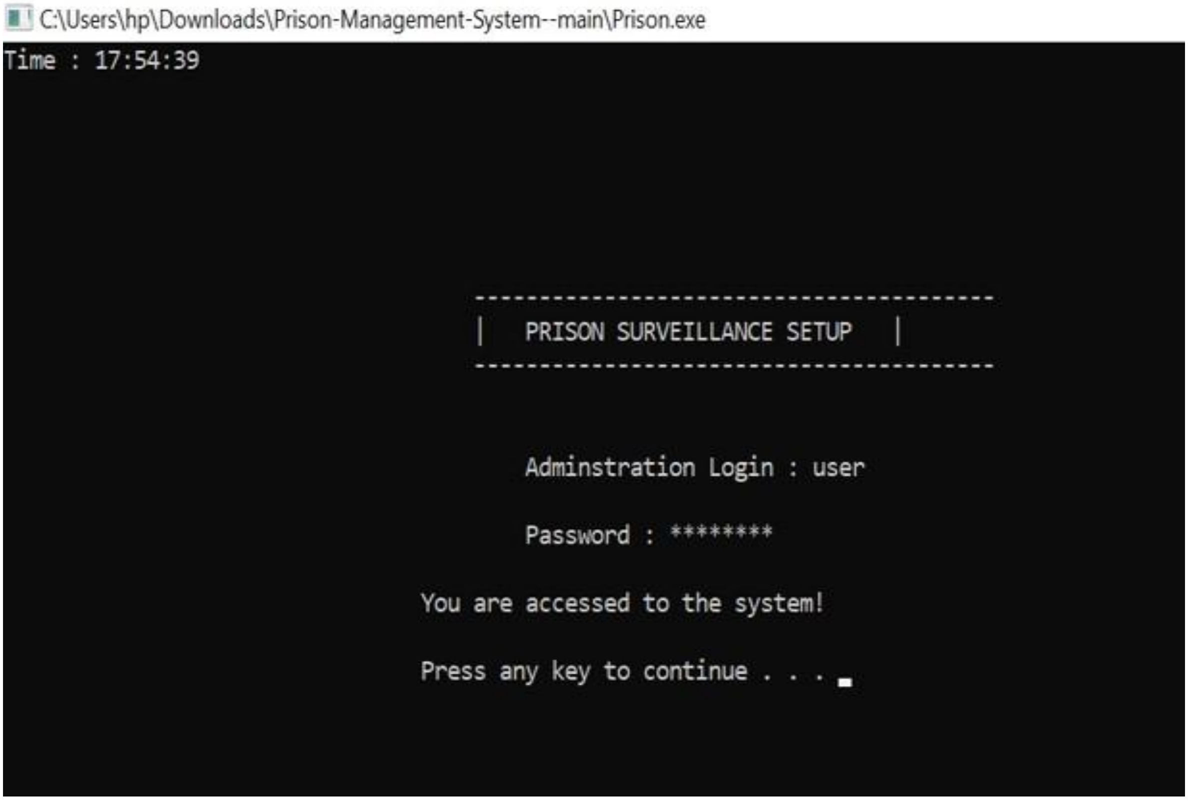
```
                                No. of attempts remain: 2

                         Press any key to continue . . .
```

Fig 9. Fault Login Attempt Output.

PSEUDOCODE OF ADMINISTRATION LOGIN

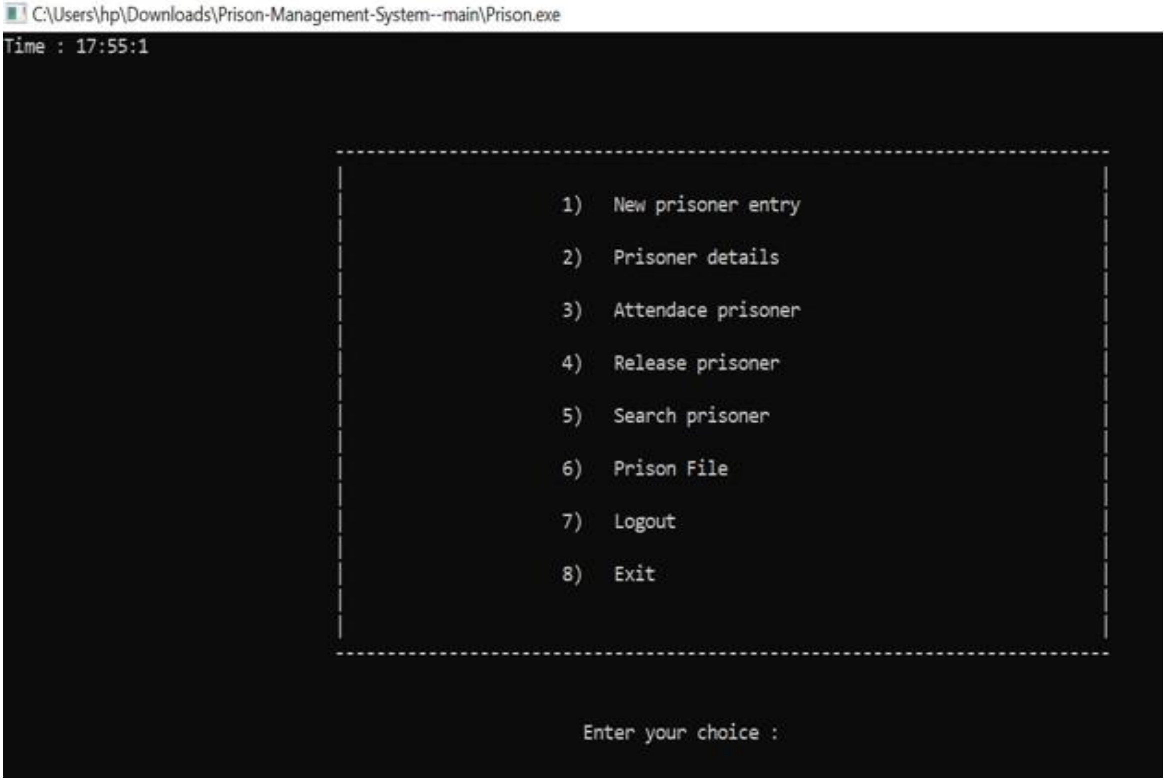
```
Step 1:- Import all the header files.

Step 2:- Declare main function

                   Int main ();

Step3 :-  Display Login options .

  Cout <<  1. Administration login

                 2. police login

                 3. prisoner login

Cin >> 1.

Step 4: - Declare Administration login function

             void Prison: Ad_Login()

Step 5: - display welcome screen

print Welcome message

Step 6: - declare variables

               Char username, password;

               Int try1;

Step 7: - Admin input user name and password
               Cin >> username
                Cin >> password
If (user name = "user" && password = "admin123")
Then {access is granted}
Else {
 no. Of attempts remains: "3- try1";}
If (try1 >= 3)
{no permission to enter the system}

Step 8: - After access to the system

Functions: 1. Set data
                 2. details
                 3. Release
                 4. Attendance
                 5. Search Prison
                 6. Prison File
                 7. Exit
                 8. Logout
Cin >> {1,2,3,4,5,6}

Step 9: - Logout
```

OUTPUT



```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:54:39

                                        ----------------------------------------
                                        |   PRISON SURVEILLANCE SETUP   |
                                        ----------------------------------------

                                            Adminstration Login : user

                                            Password : ********

                                   You are accessed to the system!

                                   Press any key to continue . . .
```

Fig 10. Administration Login Page.



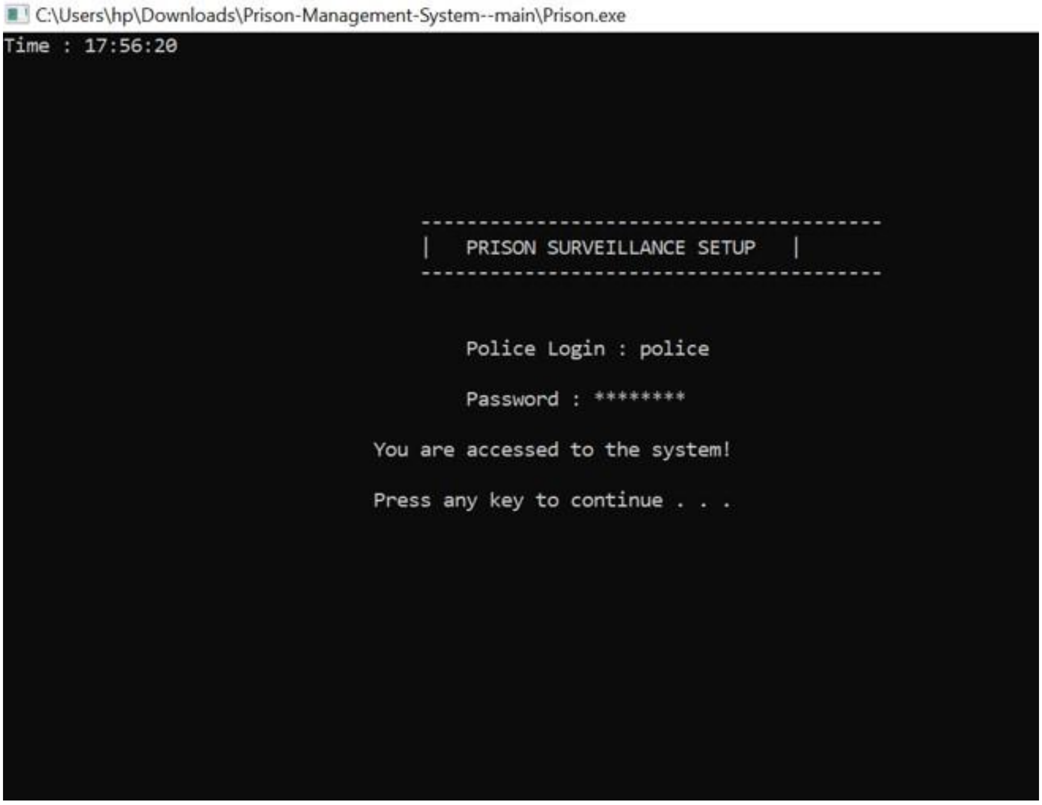
```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:55:1

                    --------------------------------------------------------------
                    |                                                            |
                    |             1)   New prisoner entry                        |
                    |                                                            |
                    |             2)   Prisoner details                          |
                    |                                                            |
                    |             3)   Attendace prisoner                        |
                    |                                                            |
                    |             4)   Release prisoner                          |
                    |                                                            |
                    |             5)   Search prisoner                           |
                    |                                                            |
                    |             6)   Prison File                               |
                    |                                                            |
                    |             7)   Logout                                    |
                    |                                                            |
                    |             8)   Exit                                      |
                    |                                                            |
                    --------------------------------------------------------------

                                   Enter your choice :
```

Fig 11. Accessibility Menu for the administration Login

PSEUDOCODE OF POLICE LOGIN

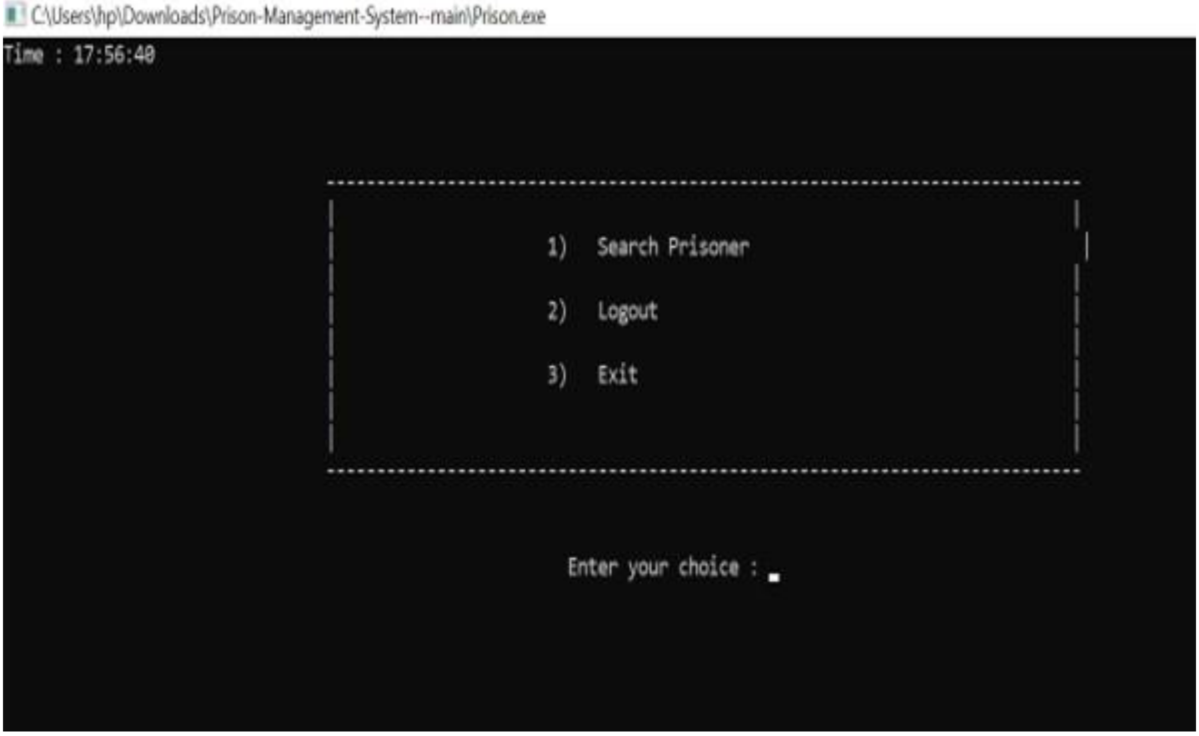
```
Step 1:- Import all the header files.
Step 2:- Declare main function
                Int main ();
Step3 :-  Display Login options .
           Cout <<  1. Administration login
                     2. police login
                      3. prisoner login
       Cin >> 2.
Step 4 :- Declare Administration login function
           void Prison::po_Login()
Step 5 :- display welcome screen
Cout<< Welcome prisoner Surveillance Setup
Step 6:- declare variables
Char username, password;
Int try1;
Step 7:- Admin input user name and password
             Cin >> username
              Cin >> password
If  ( user name = “police” && password = “ admin123)
Then { access is granted}
Else {
 no. Of attempts remain : “ 3- try1”;}
Step 8:- After access to the system
 Cout<<  1. Search Prison
              2. logout
              3. Exit
Cin >> { 1,2,3}
Step 9 :- Logout.
```

OUTPUT



```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:56:20



                                       ----------------------------------------
                                       |   PRISON SURVEILLANCE SETUP   |
                                       ----------------------------------------


                                           Police Login : police

                                           Password : ********

                                  You are accessed to the system!

                                  Press any key to continue . . .
```

Fig 12. Police Login Page



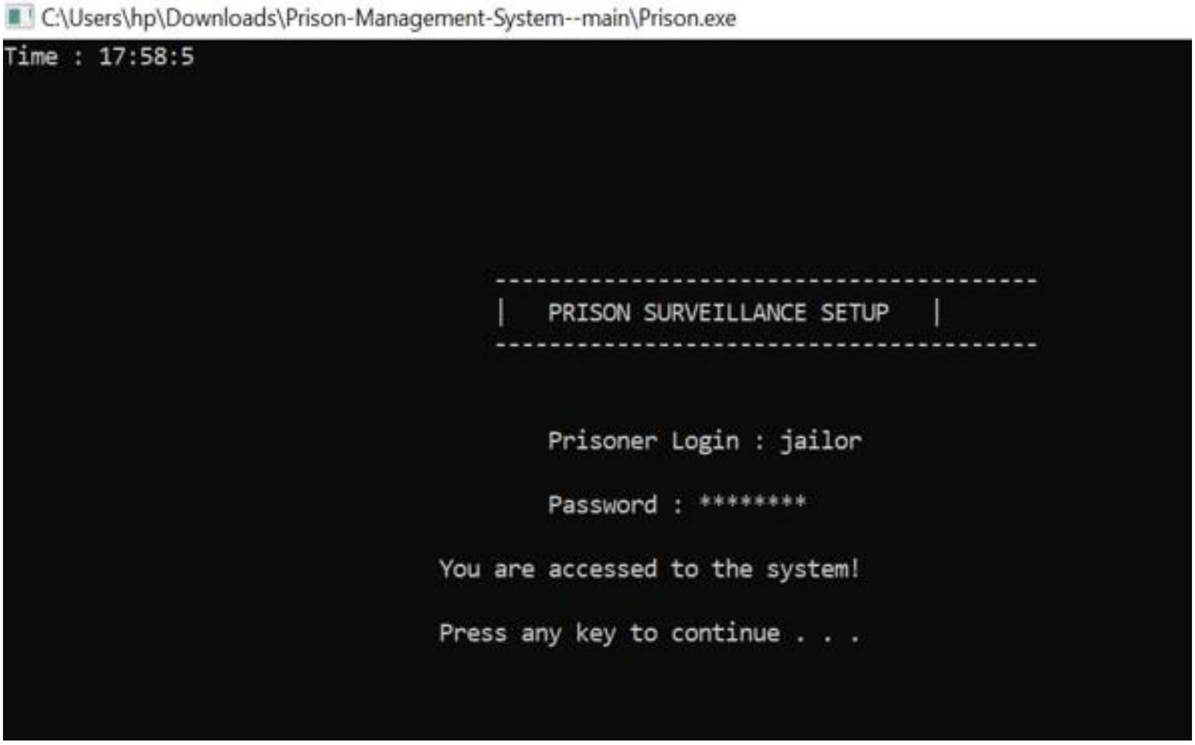
```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:56:40


                    -------------------------------------------------------------------------
                    |                                                                       |
                    |                 1)   Search Prisoner                                   |
                    |                                                                       |
                    |                 2)   Logout                                           |
                    |                                                                       |
                    |                 3)   Exit                                             |
                    |                                                                       |
                    |                                                                       |
                    -------------------------------------------------------------------------


                                    Enter your choice : _
```

Fig 13. Accessibility Menu for the administration Login

PSEUDOCODE OF JAILOR LOGIN

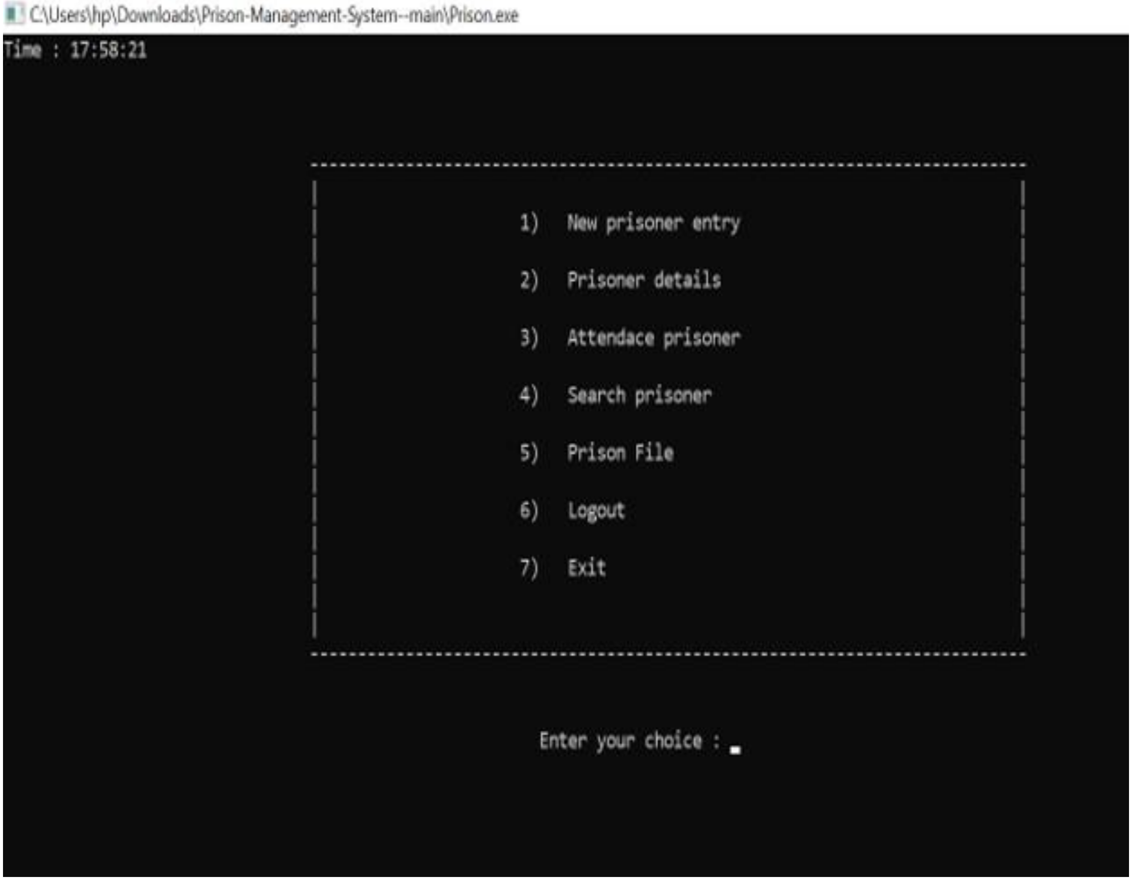
```
Step 1:- Import all the header files.

Step 2:- Declare main function

            Int main ();

Step3 :-  Display Login options .

 Cout <<  1. Administration login
          2. police login
          3. prisoner login
Cin >> 3

Step 4 :- Declare Jailor login  function

        void Prison::Pr_Login()

Step 5 :- display welcome screen

print Welcome message

Step 6:- declare variables

         Char username, password;

         Int try1;

Step 7:- Admin input user name and password
         Cin >> username
          Cin >> password
If  ( user name = "jailor" && password = " admin123")
Then { access is granted}
Else {
 no. Of attempts remain : " 3- try1";}
If  ( try1 >= 3)

{ no permission to enter the system }

Step 8:- After access to the system

Functions: 1. Set data
           2. details
            3. Attendance
           4. Search Prison
           5. Prison File
           6. Exit
           7. Logout
Cin >> { 1,2,3,4,5,6}

Step 9 :- Logout
```

OUTPUT



```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:58:5

                              ----------------------------------------
                              |   PRISON SURVEILLANCE SETUP    |
                              ----------------------------------------

                                  Prisoner Login : jailor

                                  Password : ********

                              You are accessed to the system!

                              Press any key to continue . . .
```

Fig 14. Login Page for Jailor Login



```
C:\Users\hp\Downloads\Prison-Management-System--main\Prison.exe
Time : 17:58:21

              ---------------------------------------------------------------
              |                                                             |
              |                 1)  New prisoner entry                      |
              |                                                             |
              |                 2)  Prisoner details                        |
              |                                                             |
              |                 3)  Attendace prisoner                      |
              |                                                             |
              |                 4)  Search prisoner                         |
              |                                                             |
              |                 5)  Prison File                             |
              |                                                             |
              |                 6)  Logout                                  |
              |                                                             |
              |                 7)  Exit                                    |
              |                                                             |
              ---------------------------------------------------------------


                               Enter your choice : _
```

Fig 15. Accessibility Menu for the Jailor Login

CHALLENGES FACED

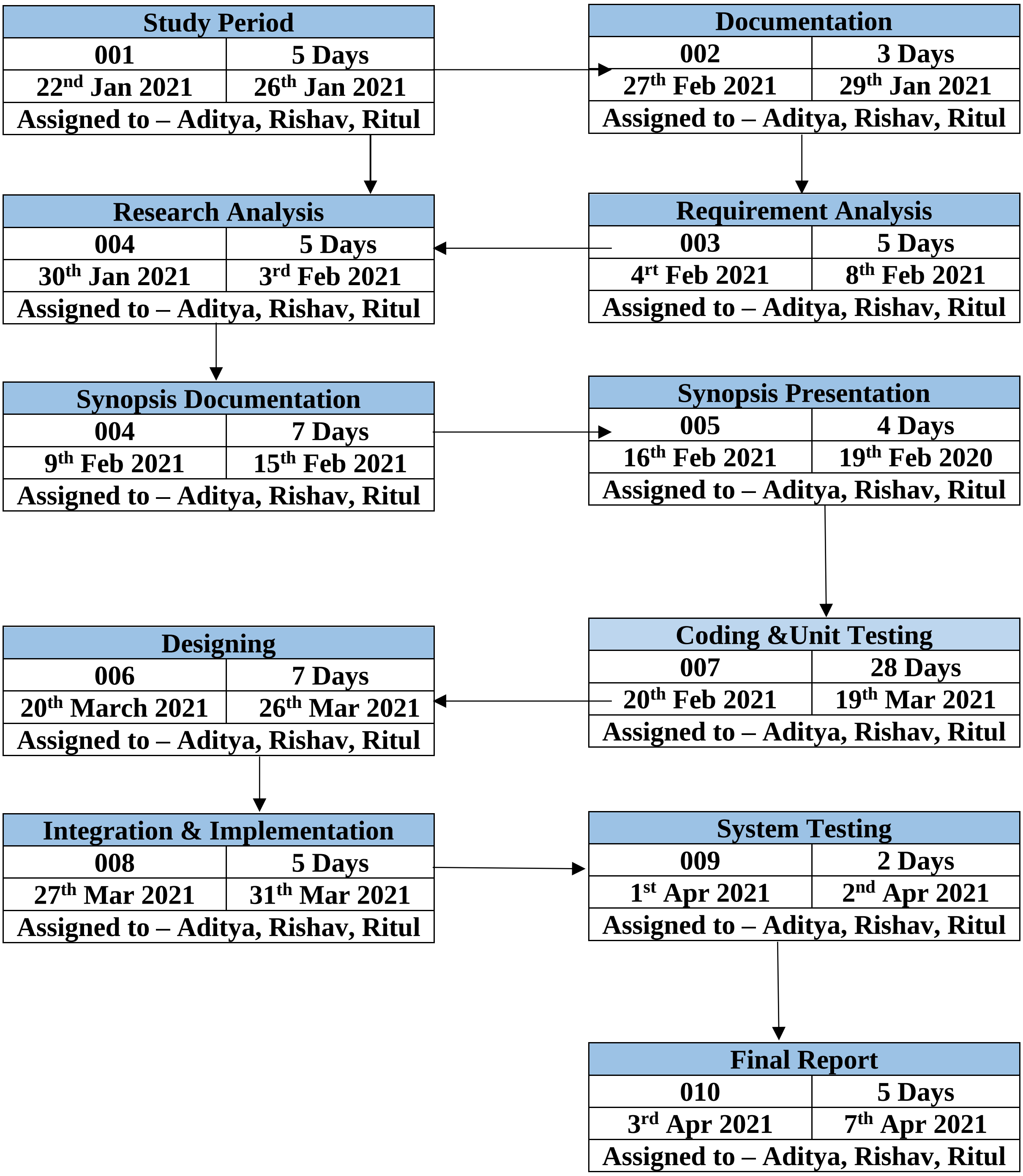
- Implementation of Naïve Algorithm.
- Implementation of real time stamping on every interface window.
- Creating and integrating all the three interfaces.
- Implementation of Secured login for every interface.
- For storing the Details so that the data won’t get lost after closing the game

FUTURE SCOPE

- In the future, this setup will be converted in Application because this will help in the portability of using this system.
- To increase the security basis, we will add a One-Time Password (OTP) for every login attempt.
- In the current scenario while interrogation of any criminal, they don't tell the real identity of themselves and in maximum cases, the proof of identity id's are also fake, so to solve this problem we will add a finger scanner facility with the system where fingerprint scanner will give approximately 95% of real data, this finger scanner device will be so optimized technique because if someone wants to get the information from desktop they can attach the scanner or someone tries to get the information from the phone/tablet they can just simply attach the scanner with the device and can fetch the data at that particular time, the time is also reduced because police individuals don't have to take the suspected person or criminal to the crime branch to fetch the details, they can get the details from anywhere.

PLAN OF WORK

Pert-chart of the schedule:



Study Period	
001	5 Days
22nd Jan 2021	26th Jan 2021
Assigned to – Aditya, Rishav, Ritul	

Documentation	
002	3 Days
27th Feb 2021	29th Jan 2021
Assigned to – Aditya, Rishav, Ritul	

Research Analysis	
004	5 Days
30th Jan 2021	3rd Feb 2021
Assigned to – Aditya, Rishav, Ritul	

Requirement Analysis	
003	5 Days
4rt Feb 2021	8th Feb 2021
Assigned to – Aditya, Rishav, Ritul	

Synopsis Documentation	
004	7 Days
9th Feb 2021	15th Feb 2021
Assigned to – Aditya, Rishav, Ritul	

Synopsis Presentation	
005	4 Days
16th Feb 2021	19th Feb 2020
Assigned to – Aditya, Rishav, Ritul	

Designing	
006	7 Days
20th March 2021	26th Mar 2021
Assigned to – Aditya, Rishav, Ritul	

Coding &Unit Testing	
007	28 Days
20th Feb 2021	19th Mar 2021
Assigned to – Aditya, Rishav, Ritul	

Integration & Implementation	
008	5 Days
27th Mar 2021	31th Mar 2021
Assigned to – Aditya, Rishav, Ritul	

System Testing	
009	2 Days
1st Apr 2021	2nd Apr 2021
Assigned to – Aditya, Rishav, Ritul	

Final Report	
010	5 Days
3rd Apr 2021	7th Apr 2021
Assigned to – Aditya, Rishav, Ritul	

SYSTEM REQUIREMENT

Software Requirements

Name of Component	Specification
Operating System	Windows 10
Front End	C++ language
IDE Required	Dev C++

Hardware Requirements

Name of Component	Specification
Processor	Intel® Core™ i5-8250U CPU @ 1.60 GHZ 1.80 GH
RAM	4 GB
Hard Disk	1 TB

REFERENCES

[1] C.H.K Tsang, M.H.C Woo, C. Bloor. "An Object Oriented intelligent tourist advisor system“ https://ieeexplore.ieee.org L.A – 12-02-2021 5:33 PM

[2] Michael J. Folk, Bill Zoellick, Greg Riccardi “An Object-Oriented Approach with C++” https://books.google.co.in/books L.A – 12-02-2021 6:07 PM

[3] Darrell Woelk, Won Kim “Multimedia Information Management in an Object-Oriented Database System” www.cloudfront.net L.A – 12-02-2021 6:45 PM

[4] Bryan P. Atsatt, Earsh K. Nankeshwar, Michael J. Seilnacht, George R. Turner “Extensible Object-Oriented File system” https://patents.google.com L.A – 13-02-2021 2:00 PM

[5] Tromp, J., Steed, A. & Wilson, J. R. (2003). Systematic usability evaluation and design issues for collaborative virtual environments. Presence, 12:3, 241-267 https://www.researchgate.net/ L.A – 13-02-2021 4:30 PM

[6]https://pdfs.semanticscholar.org L.A – 13-02-2021 4:50 PM

[7] Sayan K. Pal “Iterative Water Fall method” https://www.geeksforgeeks.org L.A – 13-02-2021 7:20 PM

[8] National Institute of correction “Prison Security” https://nicic.gov/prison-security L.A – 15-02-2021 5:30 PM

[9] Android Authority “Fingerprint Scanner working principle”www.androidauthority.com L.A – 15-02-2021 7:00PM

[10] Dr. L.P RAJU “Historical Evolution of Prison System in India” www.worldwidejournals.com L.A – 15-02-2021 7:59 PM

[11] “Overview of Prisons in India” home.rajasthan.gov.in L.A – 15-02-2021 8:30 PM