**Course Project Report**

**Title of Project:** Security System for Medium Engineering Company

**Group Members:**

Mangesh Shinde.

Vishal Shinde.

Siddhant Shirode.

Devashish Shriwas.

**Introduction:**

**Project Objectives:**

* Safety is paramount. The aim of system management is to demonstrate the importance of an automated system.
* The aim of the project is to simplify the process of maintaining data of both visitors and employees.
* Collect all relevant information from the employee, such as name, address, salary, and phone number.
* To build a database containing all the information entered by the person.
* To minimize inefficiencies and human error.

**Tools Used:**

* SQLite
* Microsoft Visual Studio
* Qt Creator 4.14.2(Community)
* DB Browser
* SQLite Studio

**Detail information:**

We used SQLite as a database engine. Our project is done in three stages that we will discuss in detail in the topic ‘help and literature’. In early two stages we used “Microsoft Visual Studio” as our coding environment, as its provides a rich source of packages related to databases. “Qt Creator 4.14.2” in the third stage for making GUI. “DB Browser” and “SQLite Studio” are used in our project for opening a .db extension file for our satisfaction.

**What Is a DB File?**

The .DB [file extension](https://www.lifewire.com/what-is-a-file-extension-2625879) is often used by a program to indicate that the [file](https://www.lifewire.com/what-is-a-file-2625878) is storing information in some kind of structured database format.

For example, mobile phones might use them to store encrypted application data, contacts, text messages, or other information.

Other programs might use DB files for plugins that extend the functions of the program, or for keeping information in tables or some other structured format for chat logs, history lists, or session data.

Some files with the DB extension might not be database files at all, like the Windows Thumbnail Cache format used by *Thumbs. dB* files. Windows uses these files to show thumbnails of a folder's images before you open them.

**About SQLite**

SQLite is an in-process library that implements a [self-contained](https://www.sqlite.org/selfcontained.html), [serverless](https://www.sqlite.org/serverless.html), [zero-configuration](https://www.sqlite.org/zeroconf.html), [transactional](https://www.sqlite.org/transactional.html) SQL database engine. The code for SQLite is in the [public domain](https://www.sqlite.org/copyright.html) and is thus free for use for any purpose, commercial or private. SQLite is the [most widely deployed](https://www.sqlite.org/mostdeployed.html) database in the world with more applications than we can count, including several [high-profile projects.](https://www.sqlite.org/famous.html)

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database [file format](https://www.sqlite.org/fileformat2.html) is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between [big-endian](http://en.wikipedia.org/wiki/Endianness) and [little-endian](http://en.wikipedia.org/wiki/Endianness) architectures. These features make SQLite a popular choice as an [Application File Format](https://www.sqlite.org/appfileformat.html). SQLite database files are a [recommended storage format](https://www.sqlite.org/locrsf.html) by the US Library of Congress. Think of SQLite not as a replacement for [Oracle](http://www.oracle.com/database/index.html) but as a replacement for [fopen()](http://man.he.net/man3/fopen)

SQLite is a compact library. With all features enabled, the [library size](https://www.sqlite.org/footprint.html) can be less than 600KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be [faster than direct filesystem I/O](https://www.sqlite.org/fasterthanfs.html).

SQLite is [very carefully tested](https://www.sqlite.org/testing.html) prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves [100% branch test coverage](https://www.sqlite.org/testing.html" \l "coverage). SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are [ACID](http://en.wikipedia.org/wiki/ACID) even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides [bugs list](http://www.sqlite.org/src/rptview?rn=1) and minute-by-minute [chronologies](http://www.sqlite.org/src/timeline) of code changes.

The SQLite code base is supported by an [international team](https://www.sqlite.org/crew.html) of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the [published interface spec](https://www.sqlite.org/c3ref/intro.html), [SQL syntax](https://www.sqlite.org/lang.html), and database [file format](https://www.sqlite.org/fileformat2.html). The source code is free to anybody who wants it, but [professional support](https://www.sqlite.org/prosupport.html) is also available.

**Architectural Difference – SQLite vs MySQL**

* SQLite is an open-source project available in the public domain.
* MySQL is an open-source project which is owned by Oracle.
* SQLite is a server-less database and is self-contained. This is also referred to as an embedded database which means the DB engine runs as a part of the app.
* On the other hand, MySQL requires a server to run. MySQL will require a client and server architecture to interact over a network.

**Qt framework:**

Qt, one of the earlier frameworks, is now 21 years old. Previously owned by Nokia, the current owner is the Finland-based Qt Company. The platform is licensed as free/open source, and available in India Mobile, Professional and Enterprise versions starting at $79 per month.

This older framework has reinvented itself with its latest incarnation, Qt 5, which is used in everything from mobile apps to [automobiles](https://www.qt.io/qt-automotive-suite/) and [medical devices](https://www.qt.io/qt-in-medical/). One of its big strengths is the windowing system, which lets you create user interfaces, and includes advanced features such as displaying charts, data visualizations and maps from third-party providers. Qt Version 5 now handles touch as well as mouse and keyboard handling, and you can add virtual keyboards on X11 and Windows. On KDE Linux, Qt is the native GUI library.

Qt is programmed in C++, where the programming language is extended by the preprocessor MOC (meta-object compiler) with features such as the elementary signal and slot mechanism (enables event-controlled communication between program objects). To do this, the pre-processor generates C++ conforming to the standard from the Qt source code, even before compiling, which means Qt applications with current C++ compilers such as GCC, ICC, MinGW or MSVC can be translated. Newer versions of the framework also offer access to the Qt’s own markup language QML, which ensures simplifications, particularly with GUI development. Aside from these internal language solutions, there are various connectivity’s supplied by third party providers for other program languages such as Python, Ruby, Go, Java, and PHP.

With qmake (standard solution) and Qbs (QtBuild Suite) the framework holds two of its own build-systems, where other systems such as CMake can also be used. Users also have their own integrated development environment, Qt Creator, which has a code editor among other things, and enables quick access to selected components.

**Advantages:**

* Qt has its own designer and designing a good GUI with the GUI designer available with Qt designer is simple with the help of spacers, buttons, html editor etc.,
* It is a complete framework you can do almost anything, designing a Good GUI, using Bluetooth, and displaying images within a few lines of code.
* Cross platformed runs on Windows,Linux,Mac,Android,etc.,

**What is QML?**

QML is a user interface specification and programming language. It allows developers and designers alike to highly create performant, fluidly animated and visually appealing applications. QML offers a highly readable, declarative, JSON-like syntax with support for imperative JavaScript expressions combined with dynamic property bindings.

**Literature and Help**

Security Management System (SMS) is defined as an efficient method to managing sensitive company information so that it remains secure. The security management system is a very broad area that generally includes everything from the supervision of security guards at malls and museums to the installation of high-tech security management systems that are generally made to protect an organization’s data. Read on to learn more about this field and get examples of the types of security management in place today.

**Feature of Security Management System:**

Security management relates to the physical safety of buildings, people and products.

Security management is the identification of the organization’s assets.

Generally, Security Management System is provided to any enterprises used for security management and procedures as information classification, risk assessment, and risk analysis to identify threats, categorize assets, and rate.

**Importance of security management:**

There are some important of security management which is generally provided to any organization and which are given below:

**Intellectual Property:**

There are principal reasons that organizations formalize an innovation management program is to gain a competitive edge on the competition. Although if the initial ideation phases are open to everyone, a lot of work goes into developing and refining those ideas and that refinement is often the difference between an incremental idea and a transformative one and the companies don’t protect those later stage refinement activities, then they could lose the competitive edge they gain by instituting an innovation management program in the first place.

**Data Integrity:**

Security Management systems have confidence in lots of data to help prioritize and validate initiatives and generally we could be talking about votes and comments on ideas, ROI data, and beyond. If security management systems aren’t secure, this data could be stripped or tampered with. It will simply make an idea or project appear more popular or more valuable if the system can be gamed.

**Personally, Identifiable Information:**

All who participate in a security management program shares at least their personal information in order to log on to the system and where privacy is everything – security management systems are provided to protect all their users as a matter of course.

**System Interconnectivity:**

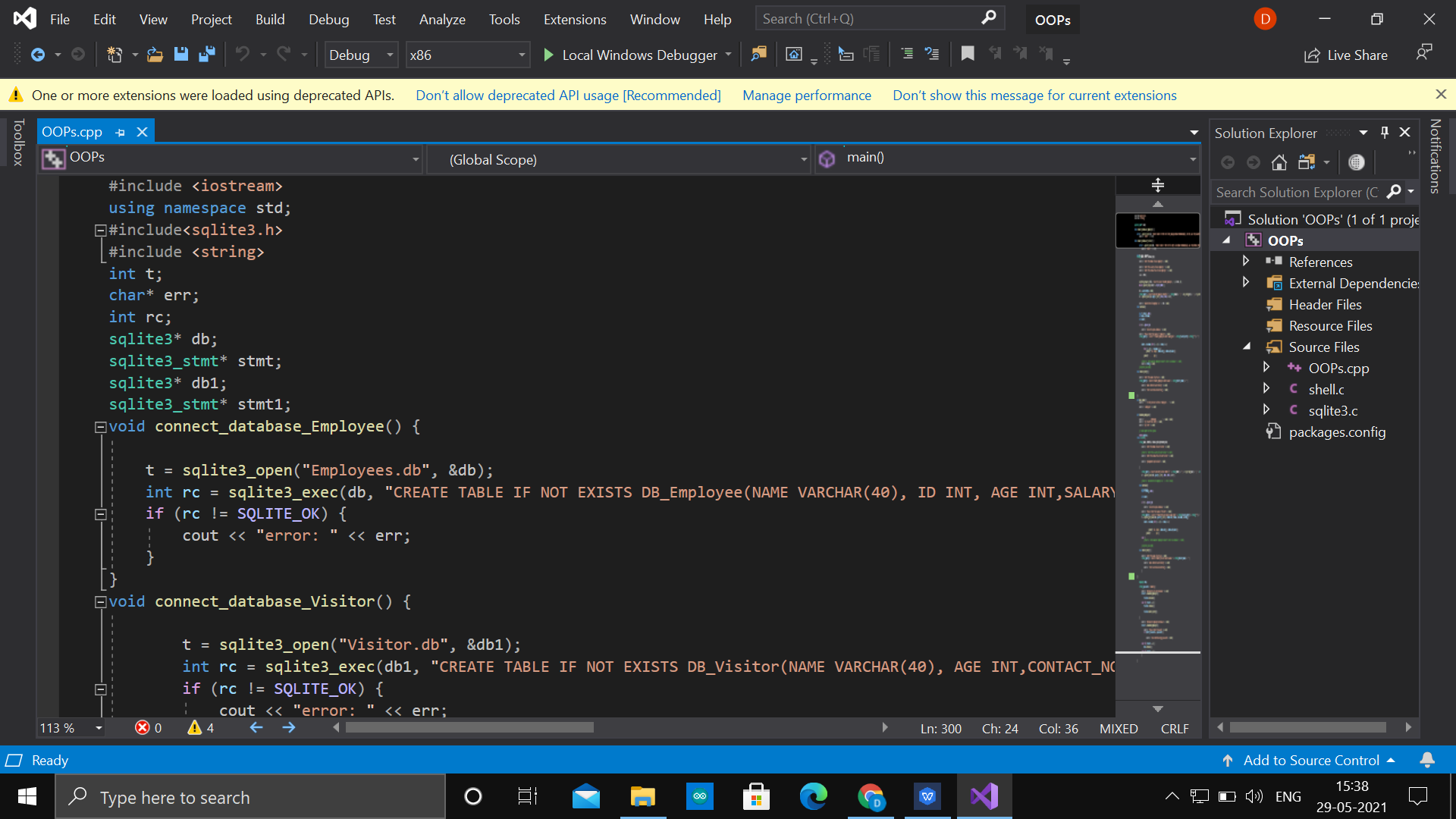
Generally, security management software interacts with a variety of other systems like project management, social software, and beyond, etc. Frailness in one system can lead to frailness in others, which is why any security management system has to be equal to the systems with which it interacts.

**Table 1: Information Security Management System Domains**

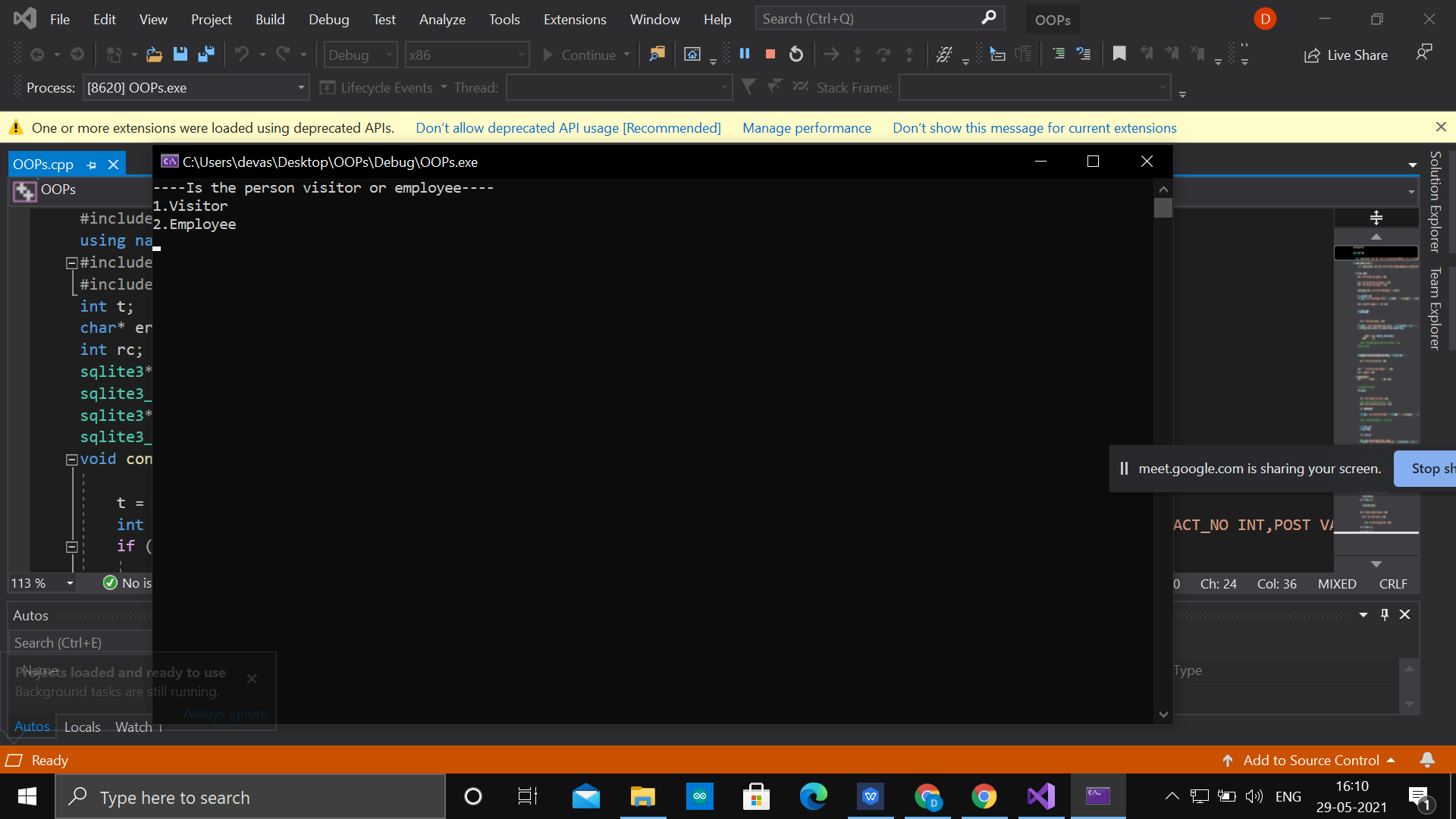
|  |  |
| --- | --- |
| **Domains** | **Objective** |
| Security policy | Provide management direction and support for information security |
| Organizational Security | Manage information security within the organization. |
| Asset Management | Achieve and maintain appropriate protection of organizational assets |
| Human Resources | Details any personnel issues like training, responsibilities, and how employees responded to security incidents |
| Physical and Environmental Security | prevent unauthorized physical access |
| Communications and Operations Management | Ensure the correct and secure operation of information processing facilities. |
| Access Control | To control access to information. |
| Information System Acquisition, Development & Maintenance | Ensure that security is an integral part of information systems. |
| Information Security Incident Management | Ensure information security events and weaknesses associated with information systems are communicated in a manner allowing timely corrective action to be taken |
| Business Continuity Management | Maintenance of essential business activities during adverse conditions, from coping with major disasters to minor, local issues |
| Compliance | Avoid breaches of any security requirements. |

**Previous Code:**

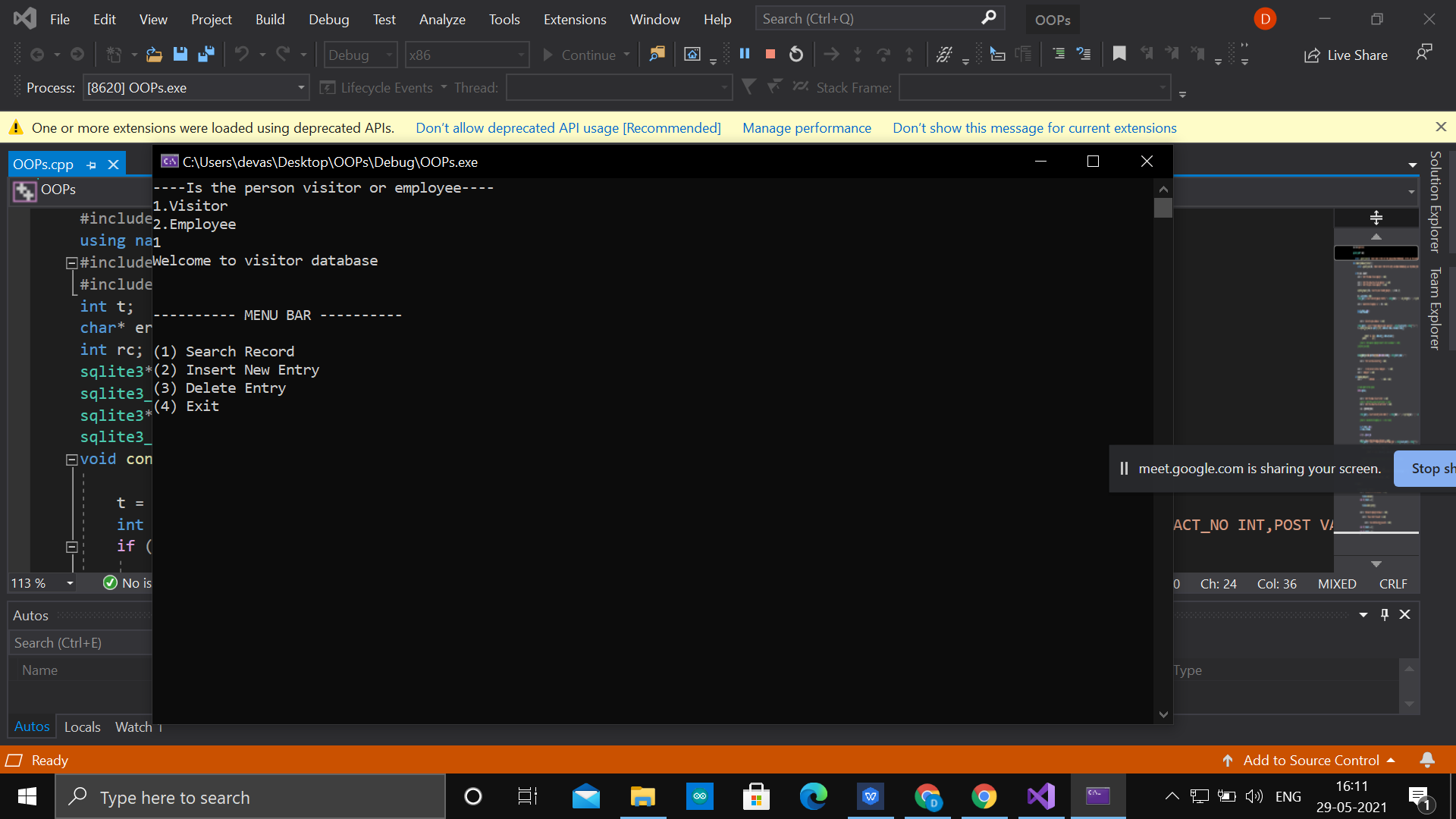
So basically, this is our first approach from which we started our course project. We decided to do our course project in c++. So, we chose Microsoft Visual Studio as our coding environment as it is easy to include an SQLite environment. So, after that we chose DB Browser and SQLite studio for opening the .db file. Some of our group members were facing problems while dealing with DB Browser, for this reason we installed SQLite for a backup plan. So, in our first approach, we decided to do coding without GUI. So, we did code only for the database. But the limitation of this type of project is that all outputs were visible in a single screen which is run in c++ environment. So, the disadvantage of this type of project is that the run window was crowded with output. This created a problem for users while analyzing results and black window with white words was not looking good to work with.

\

First Approach



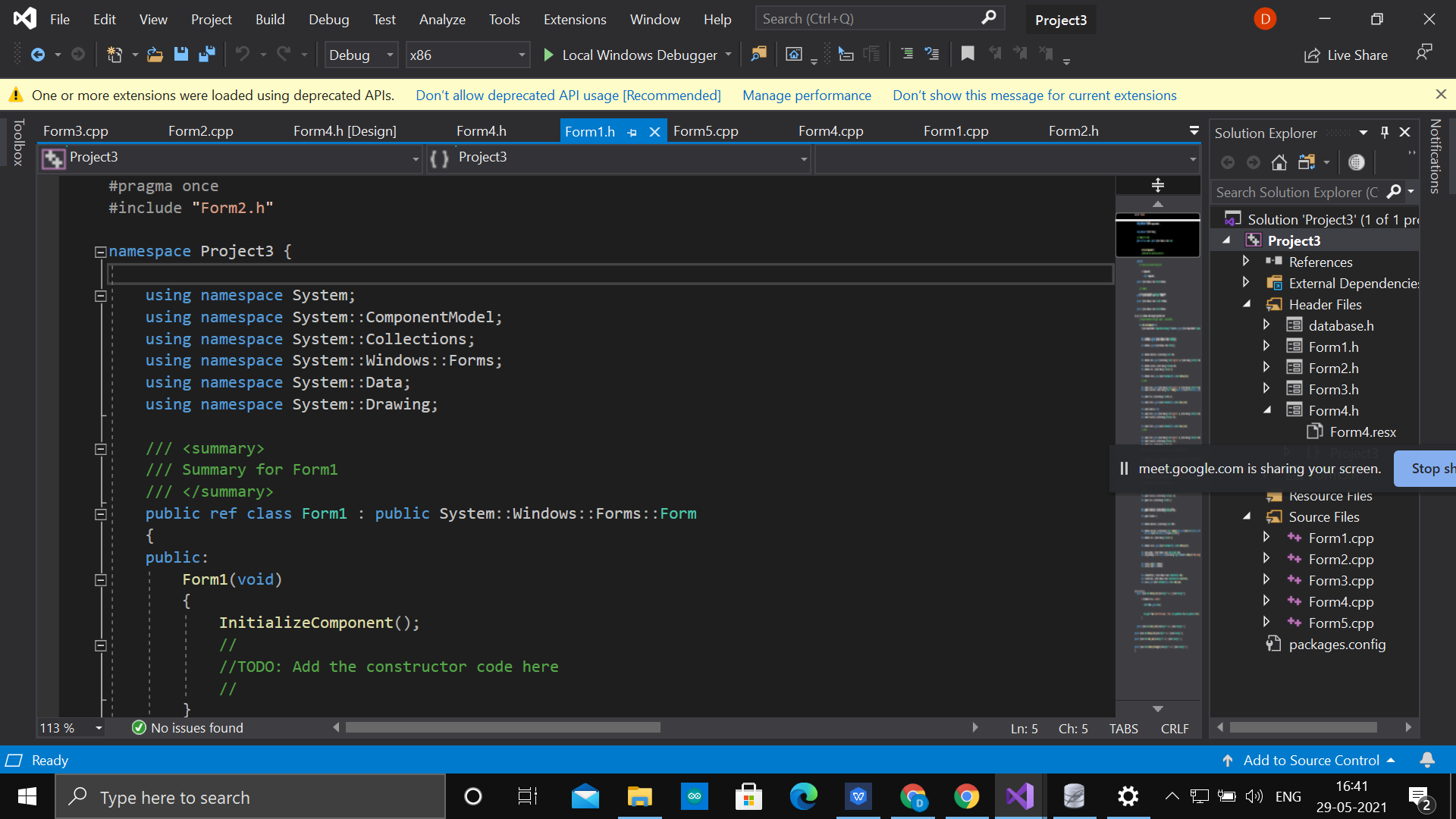
Run window



Visitor Window

**Second Approach:**

So, to cope with the disadvantages that we faced in our first approach ,we decided to make GUI for our project. So, the question arose in our mind was how to proceed? After searching , we came to know that there is one way to do this .NET framework. Then after going through some videos on YouTube, we came to know that .NET supports C++ in CLR form. Then we started working in .NET Framework .Then after doing 2-3 forms we faced difficulty with database connection. Then we came to know that .NET framework actually created for C#, and it did not work properly with C++ CLR window. Then we started searching for another environment for creating GUI in C++.



Second Approach(.NET Framework)

**Third(Final) Approach:**

So, the after cope with problems in .NET Framework, we started searching for another environment for creating GUI. After searching ,we came to know that Qt framework is good for working in C++ and database connection with the C++ is easy in Qt framework as it is provided inbuilt SQL package. So, we decided to work in Qt framework. First, we install Qt creator 4.14.2(Community) and started watching tutorial on YouTube and reading material related to Qt framework. So first we decided to final strategy how to proceed. Then we decided first to cope with the coding part related to databases and then GUI.

**Help:**

The help or references that we have used for our first approach are:

[www.sqlite.org](http://www.sqlite.org)

SQLite YouTube tutorials and QT YouTube tutorials:

<https://doc.qt.io/>

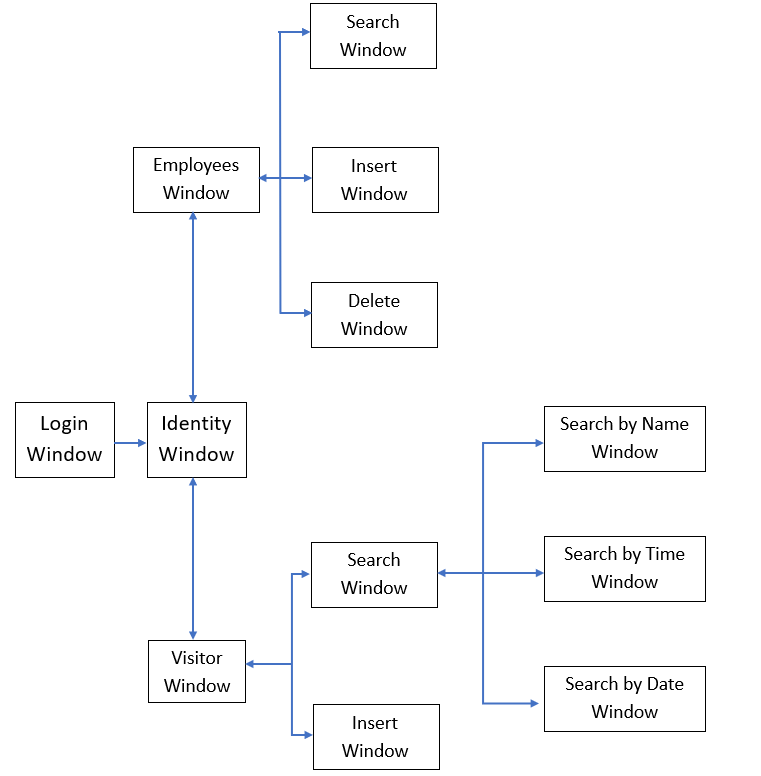
We first searched all the tools that we will be requiring for our project. Then we learnt the basics of all these tools by watching the YouTube tutorials and referring to the official documentation. And thus started the project. And if we faced any errors then we referred to the online documentation and thus solved the error.

Planed Work for this Project.

|  |  |
| --- | --- |
| Before Mid-Sem | After Mid-Sem |
| Introduction about the various course projects by the Professor. | Program related to class visitors. |
| Searching and collecting information for the course project | Creating Database of visitors |
| Working with the basics of SQLite and DBM | Finalize the project |
| Created Program related to class employees. | Preparation for report and ppt for final review |
| Created Database of employees. | Final Review Of Course Project |

**Design Aspect**

**Flow chart:**

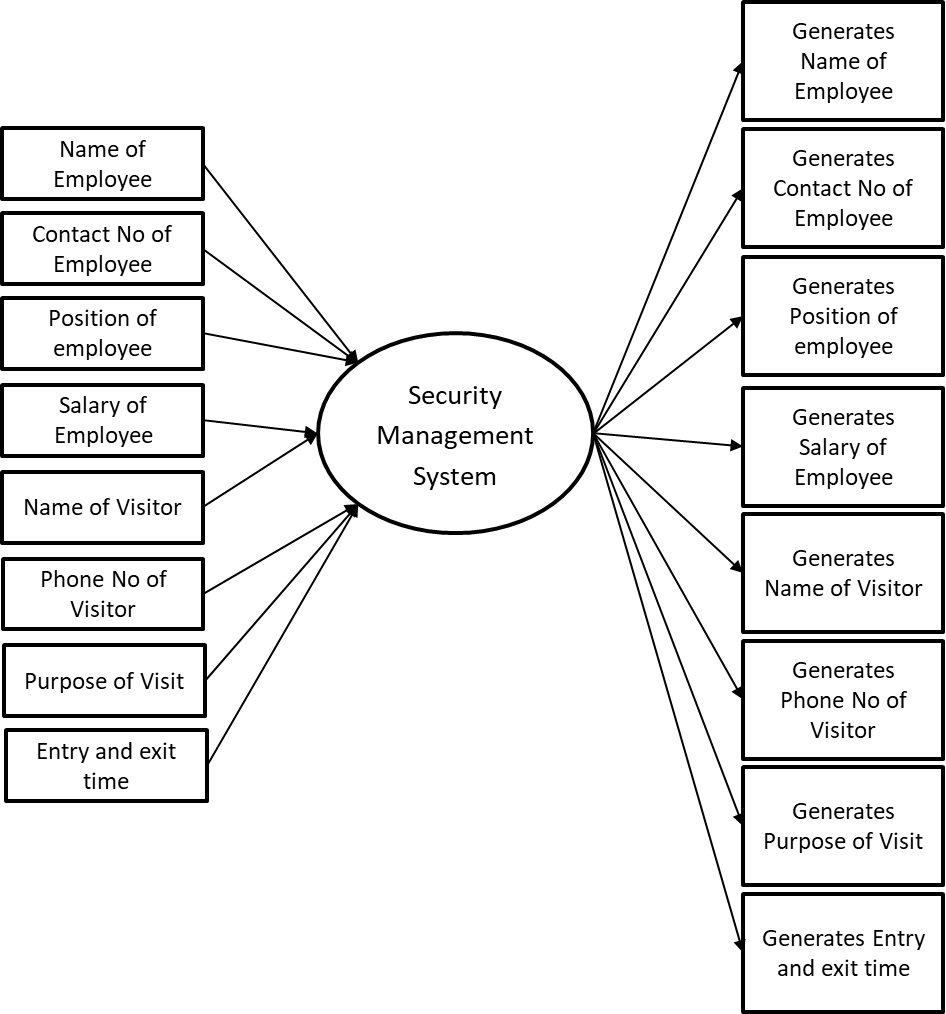


As we can see in the above flowchart, the first window is the home page or the login window. The next is the Identity window. From the identity window there are two options to open. The first is the Employee window and the second is the visitor window. From the Employee window we have three options that are search window, insert window, and delete window. From visitor we have two option that are search and insert and, in the search, further there are three options, search by name, search by time and search by date.

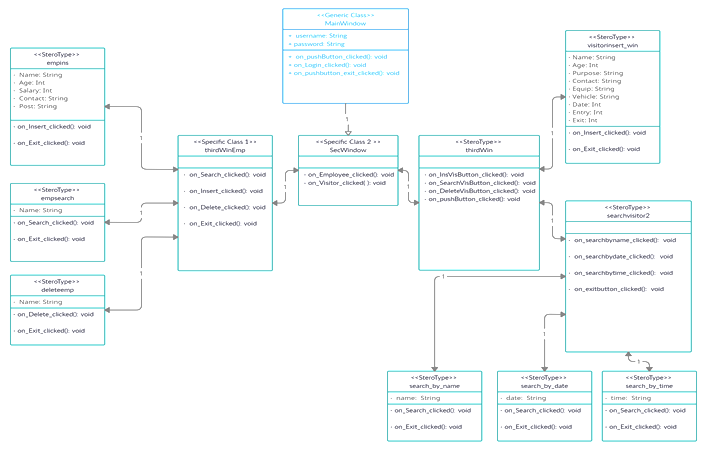
Data Flow Diagram (DFD):



**Level 2 DFD**

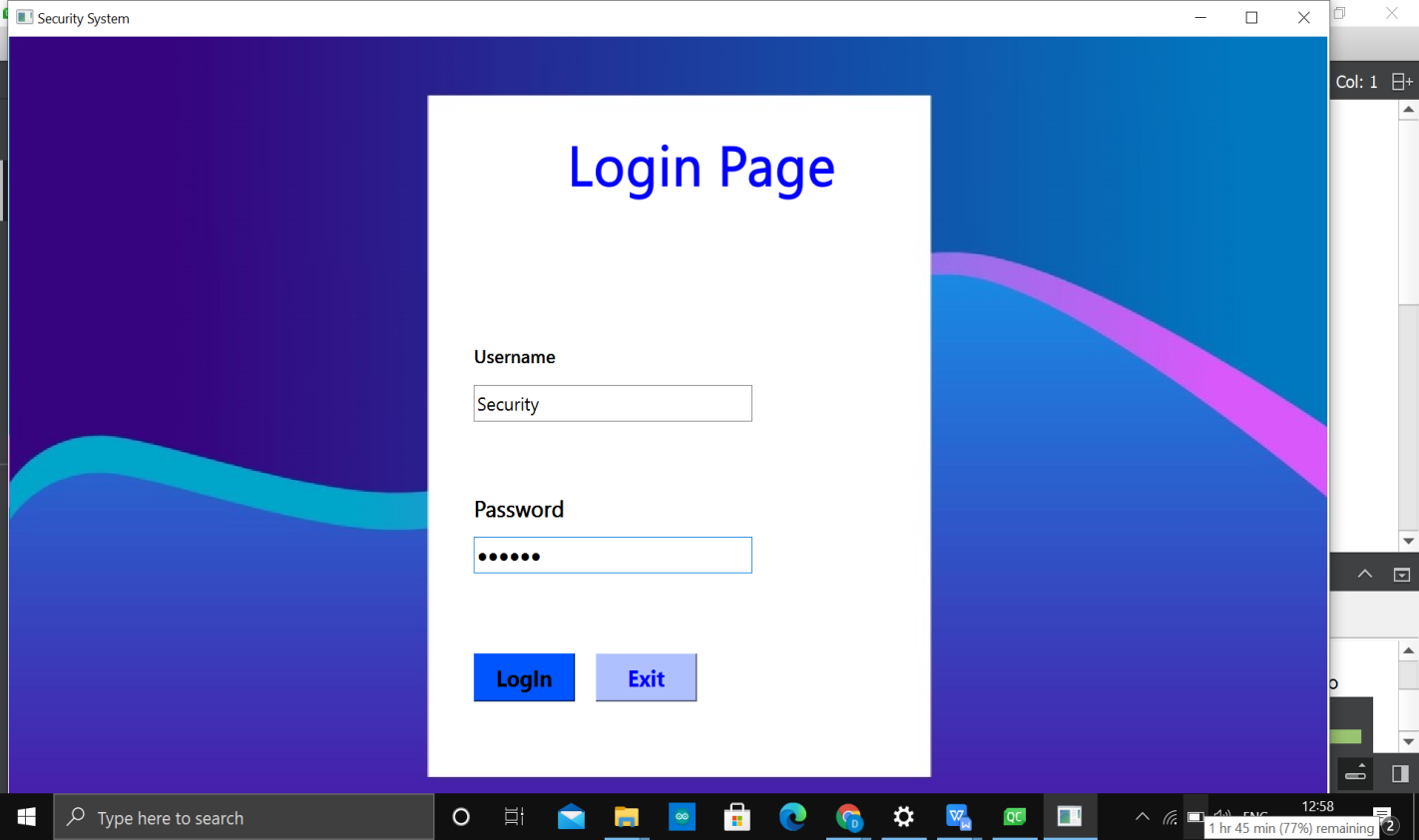


UML Diagram:



The above diagram is the UML diagram. UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. So, in the above UML diagram all the classes, variables, modifiers, methods, etc are mentioned. Through UML diagrams, it becomes very easy to understand the basic structure of the program and the code can be understood easily.

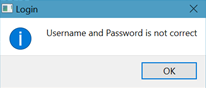
**Features and Explanation of the Application**

**Login Page:**

The Main Menu

**Explanation:**

Initially, when the program starts the home window which is shown in the above image appears. The project is done in Qt which provides a drag and drop feature to change the design of the interface. So in the home page first the label login page was set and then the username and password labels were added. Then the text boxes which contain the username and password entry were added. Then the buttons of login and exit were added. Then the CPP code was done to check whether the username and password matches with the default username and password. After entering the correct credentials, if the user clicks on the login button then the next window gets opened or else if the user clicks on the exit button then the application closes. If the credentials entered are wrong or missing then the message is displayed that the credentials entered are incorrect.

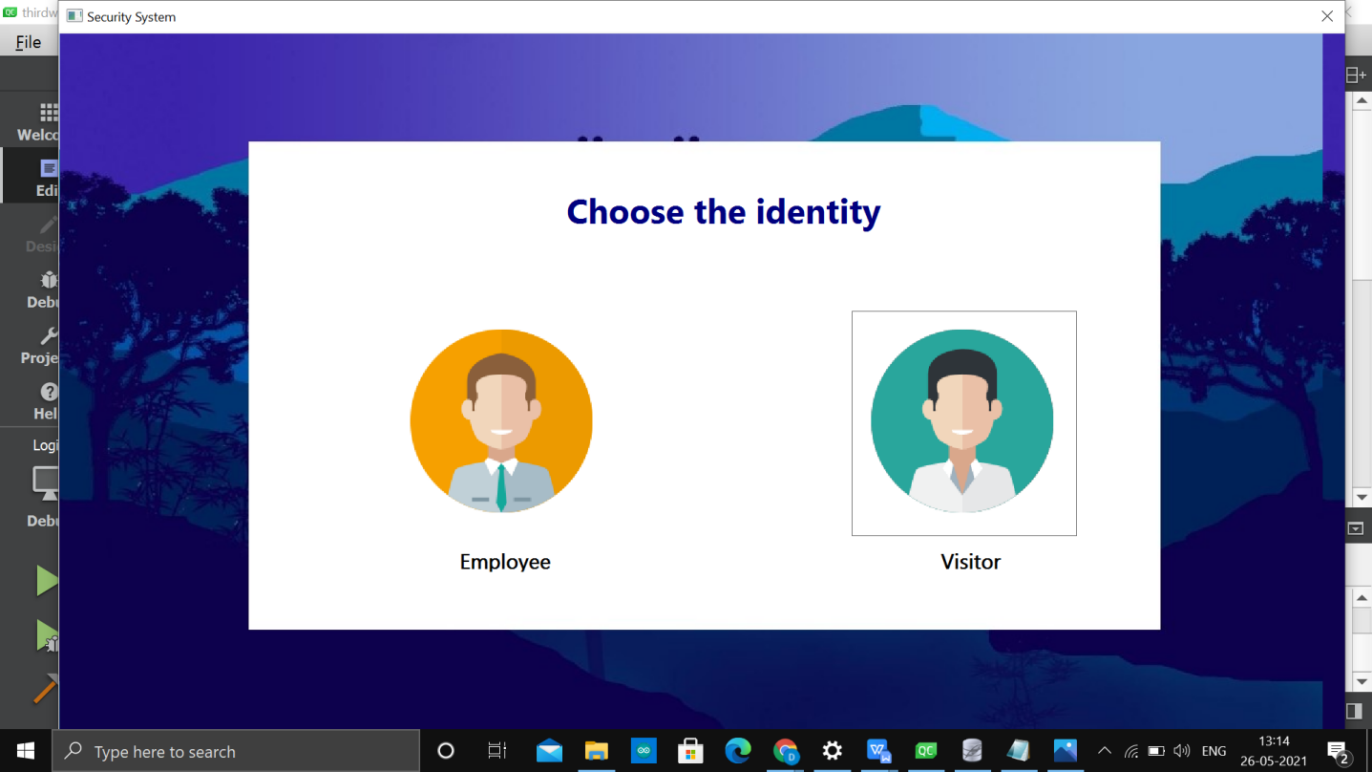


Error Message Box

**Explanation:**

If we entered any wrong credentials then the above message is displayed and if the credentials are missing even then the above message is displayed.

**Identity page:**

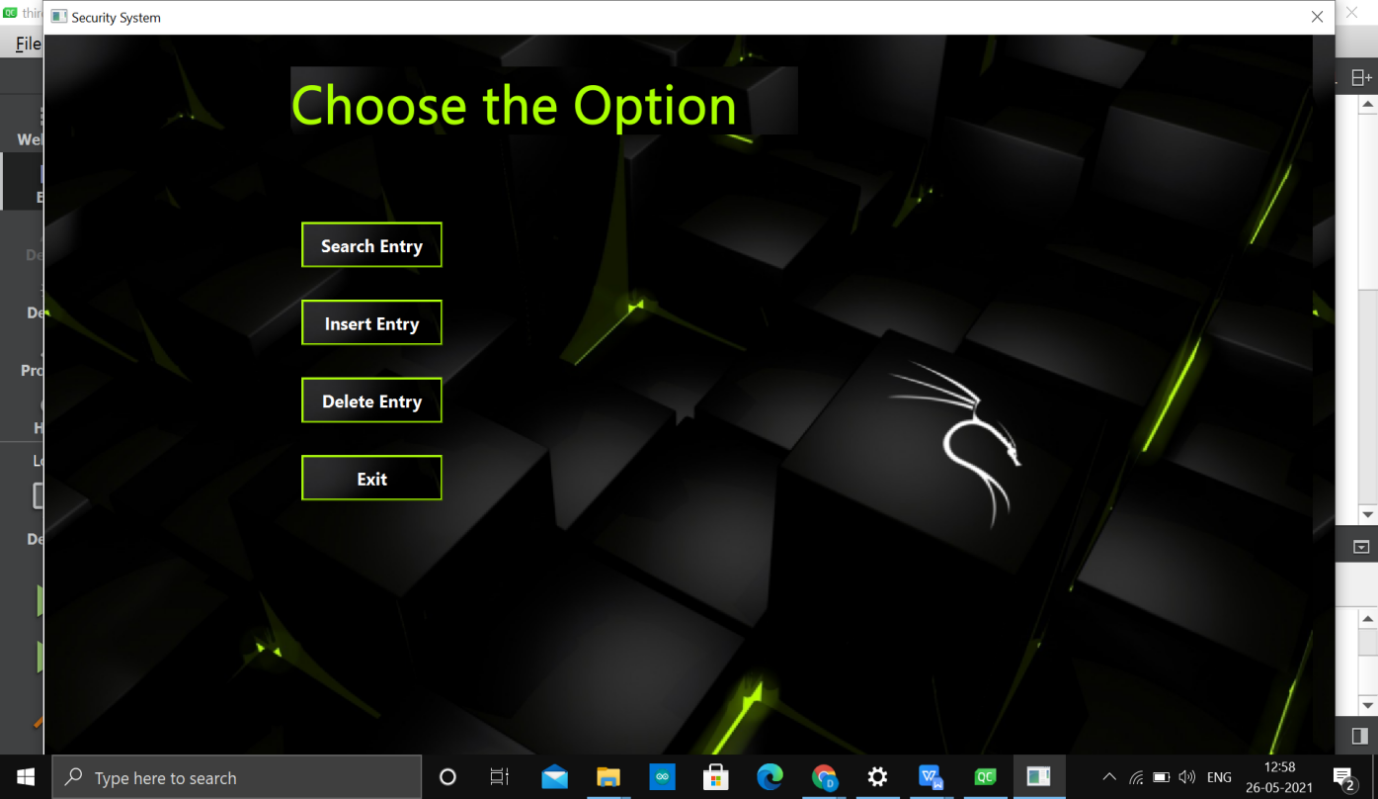


Choosing window

**Explanation:**

The second page which gets opened after clicking on the login button is the Identity window. The identity window consists of the Employee and the visitor button, also a label is added at the top. whenever the user clicks on the employee button the employee window gets opened and whenever the user clicks on the visitor button the visitor window gets opened. The Identity window looks like what is in the image above.

**Employee Window:**

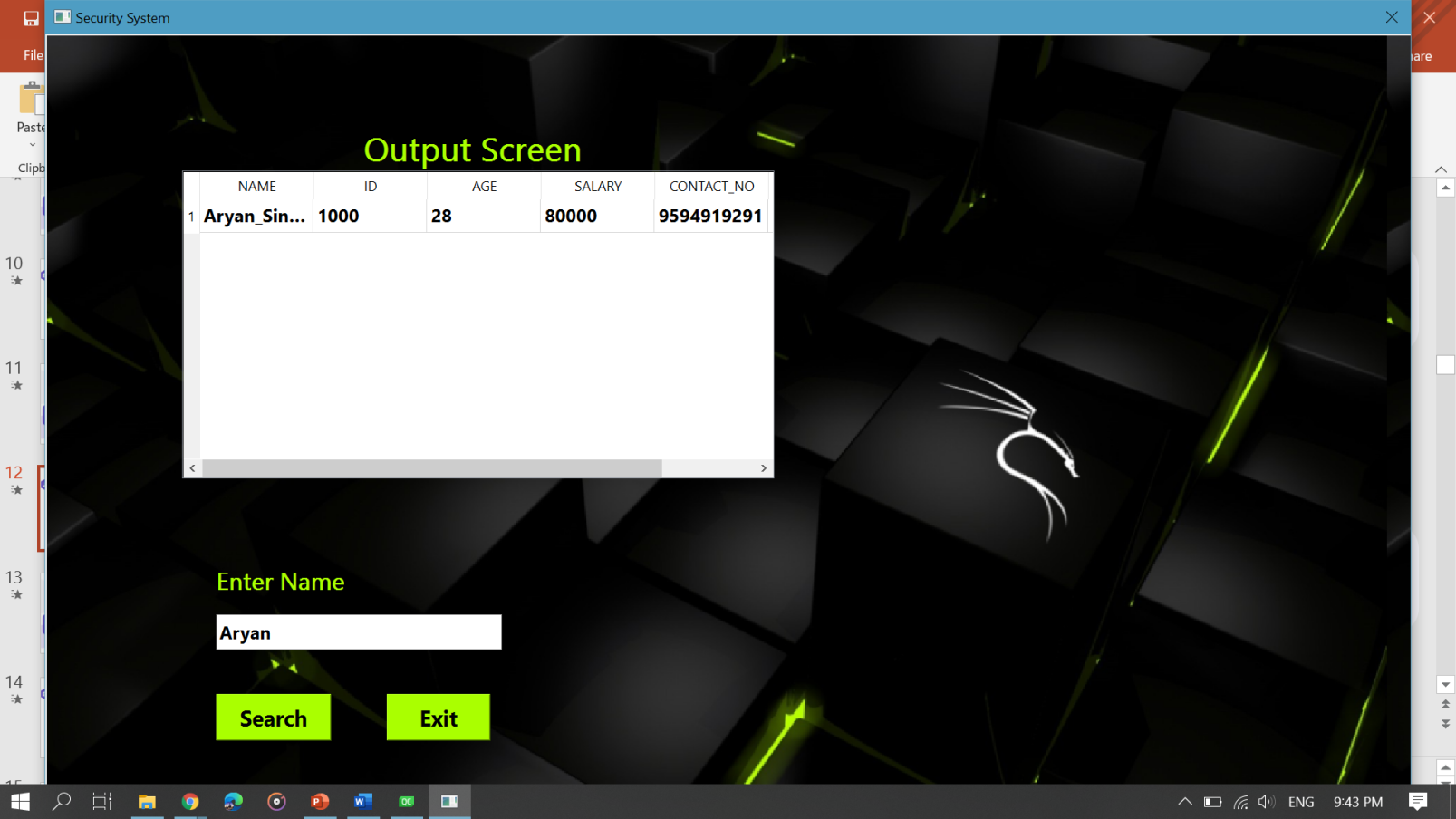


Employee window

**Explanation:**

After clicking on the employee button the window which gets opened is the Employee window which consists of the four buttons as shown in the image above. The first button is the search entry which is used to search whether the particular employee exists or not in the database. The next is the insert entry button which is used to insert a new entry in the database. The next is the delete entry button which is used to delete the entry from the database. And the last is the exit button which opens the previous window.

**Employee Search Window:**

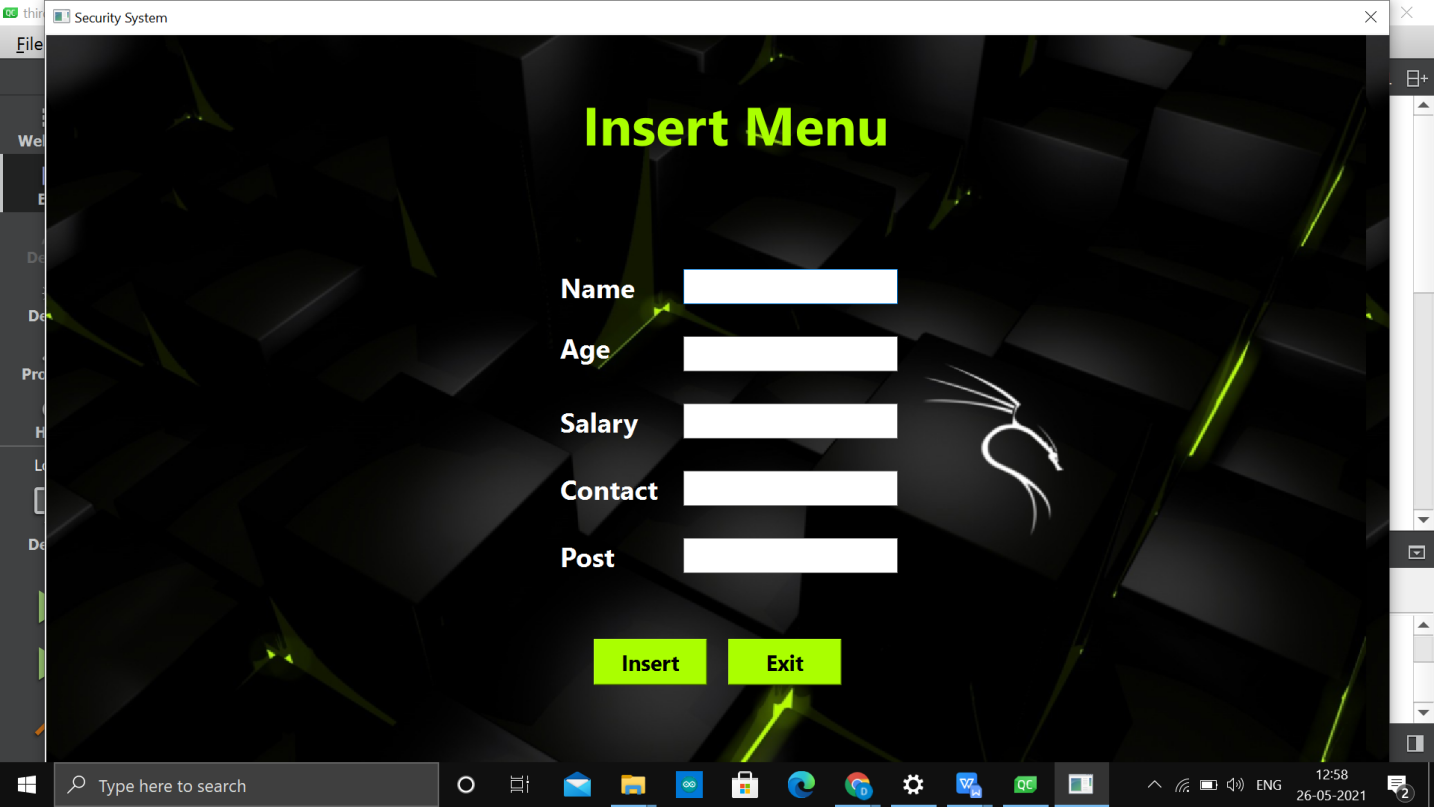


Search window(Employee)

**Explanation:**

After clicking on the search button the window shown in the above window gets opened which consists of an output screen label, the output panel, enter name label, entry, and the search and exit buttons. If the user wants to get the total database of a person then the name of that person is entered in the entry and the search button is clicked. after clicking on the search button the information of the person is shown in the panel. As shown in the above image. And whenever the user clicks the exit button then the previous window gets opened. The SQLite is the tool used for storing the database.

**Employee Insert Window:**

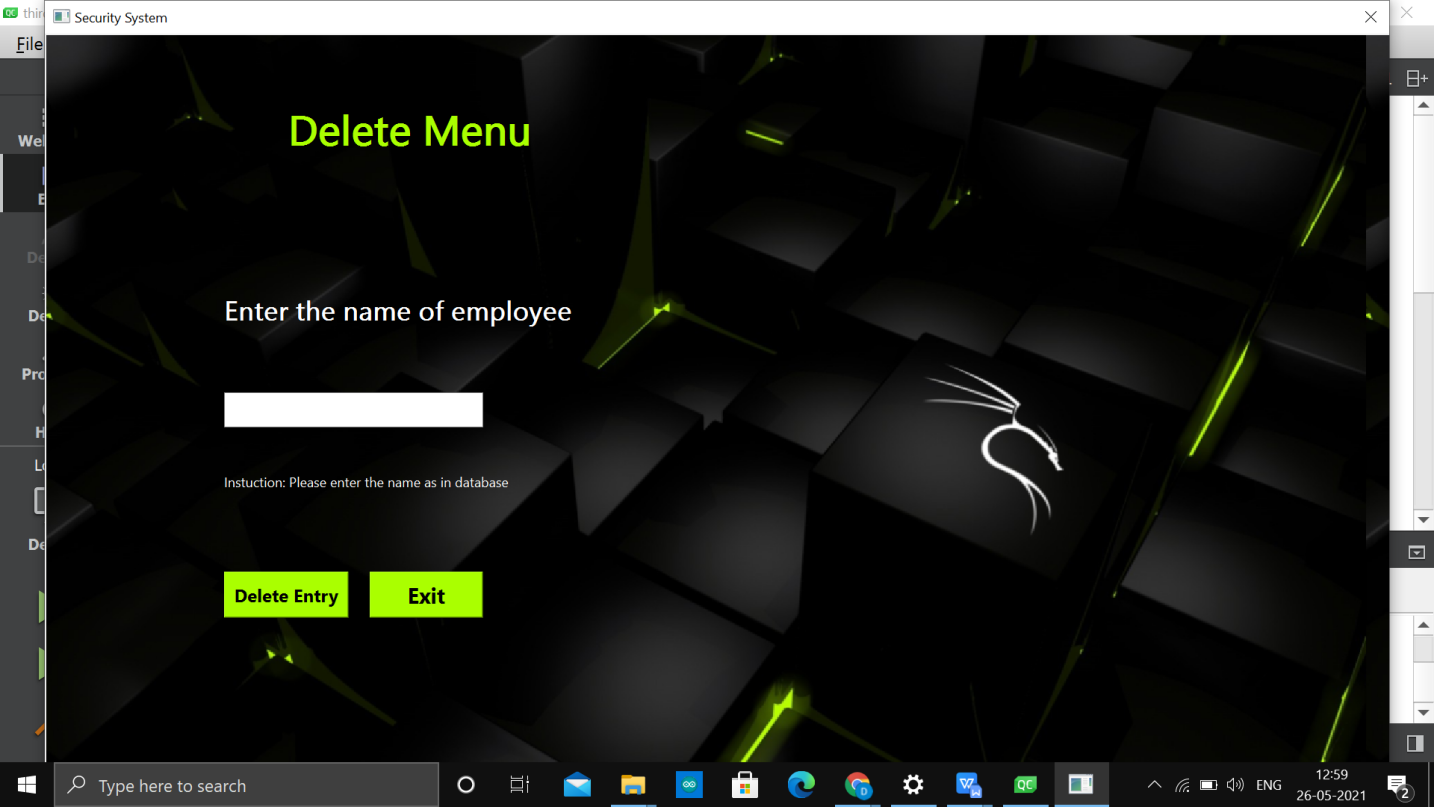


Insert window(Employee)

**Explanation:**

Whenever the user clicks on the insert button the above window gets opened. This window gives us the option to add a new employee database in the database. The person’s name, age, salary, contact number, post is entered in the entry boxes and when the insert button is clicked then the information of the employee is stored in the database. If the exit button is clicked then the previous window gets opened.

**Employee delete Window:**

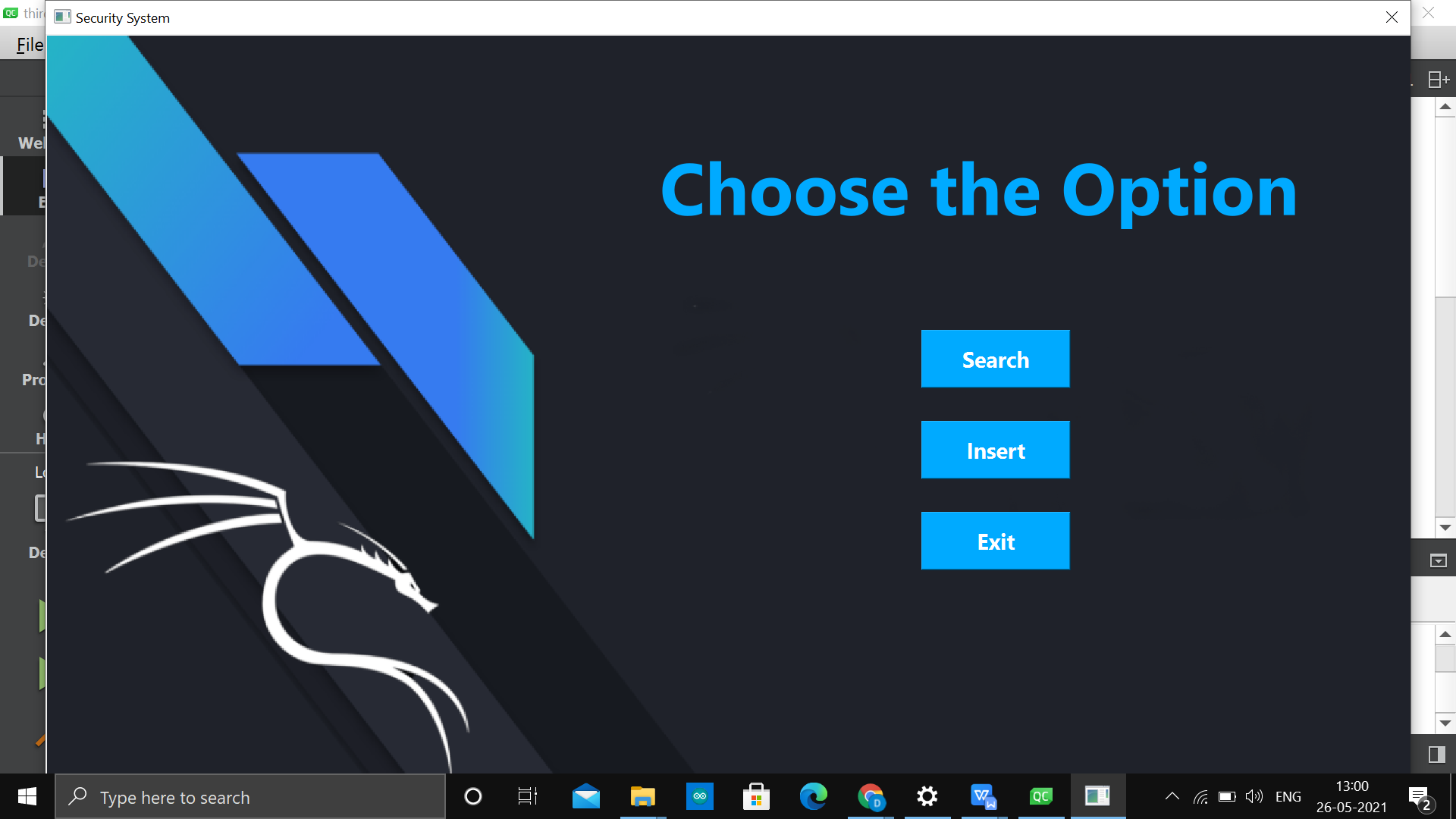


Delete window(Employee)

**Explanation:**

And if the user clicks on the delete button then the window shown in the above image gets opened. Hence, to delete the data of an employee from the database above the window is useful. The employee whose data is to be deleted is entered in the entry and then the delete entry button is clicked. After clicking the delete button the entry of the employee gets deleted. And clicking on the exit button the previous window gets opened.

**Visitor Window:**

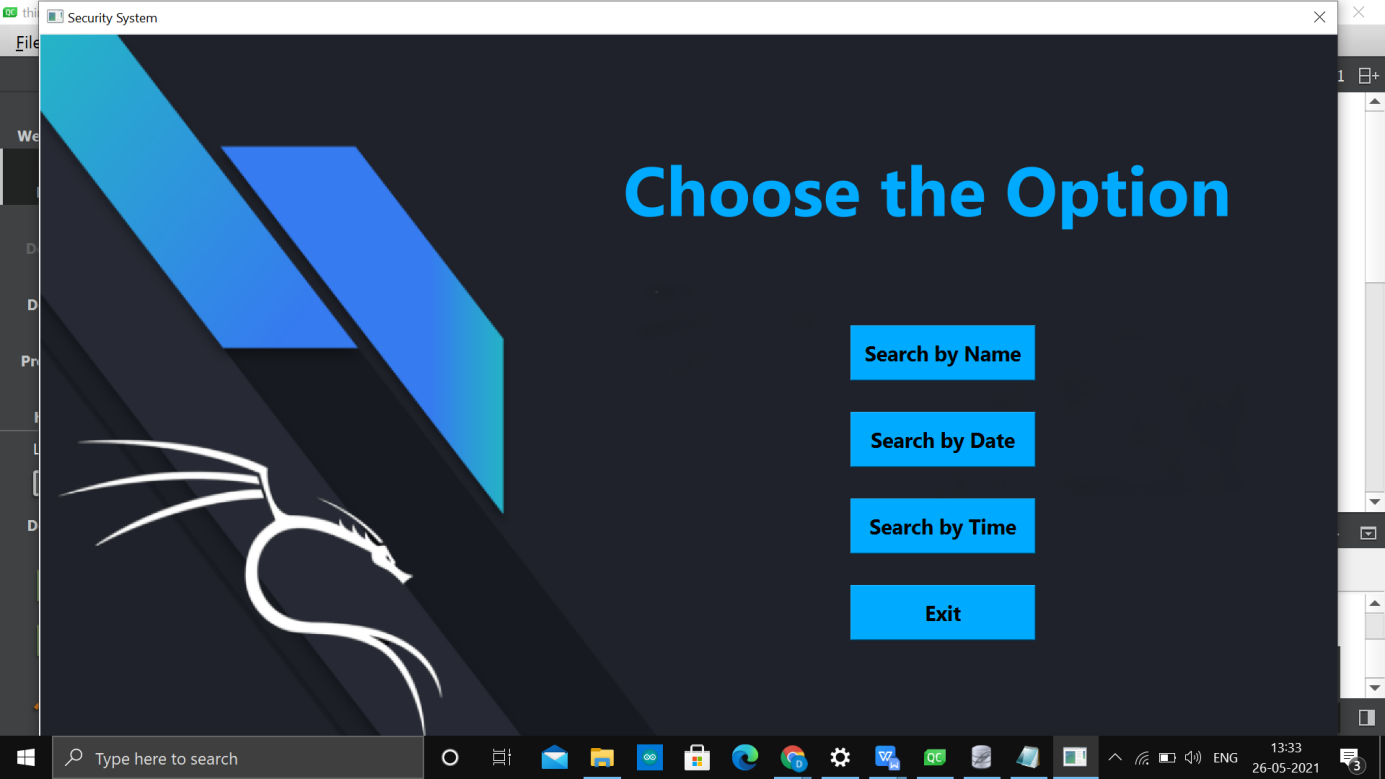


Visitor window

**Explanation:**

The window shown in the image above is the image of the visitor window. The window consists of three buttons: the first one is the search, the second is the insert button and the last is the exit button. The search button is used to search the visitor in the particular month on a particular time or by the name. The insert button is used to add the information about the new visitor in the database. The exit button is used to open the previous window of the Identity.

**Visitor Search Window:**

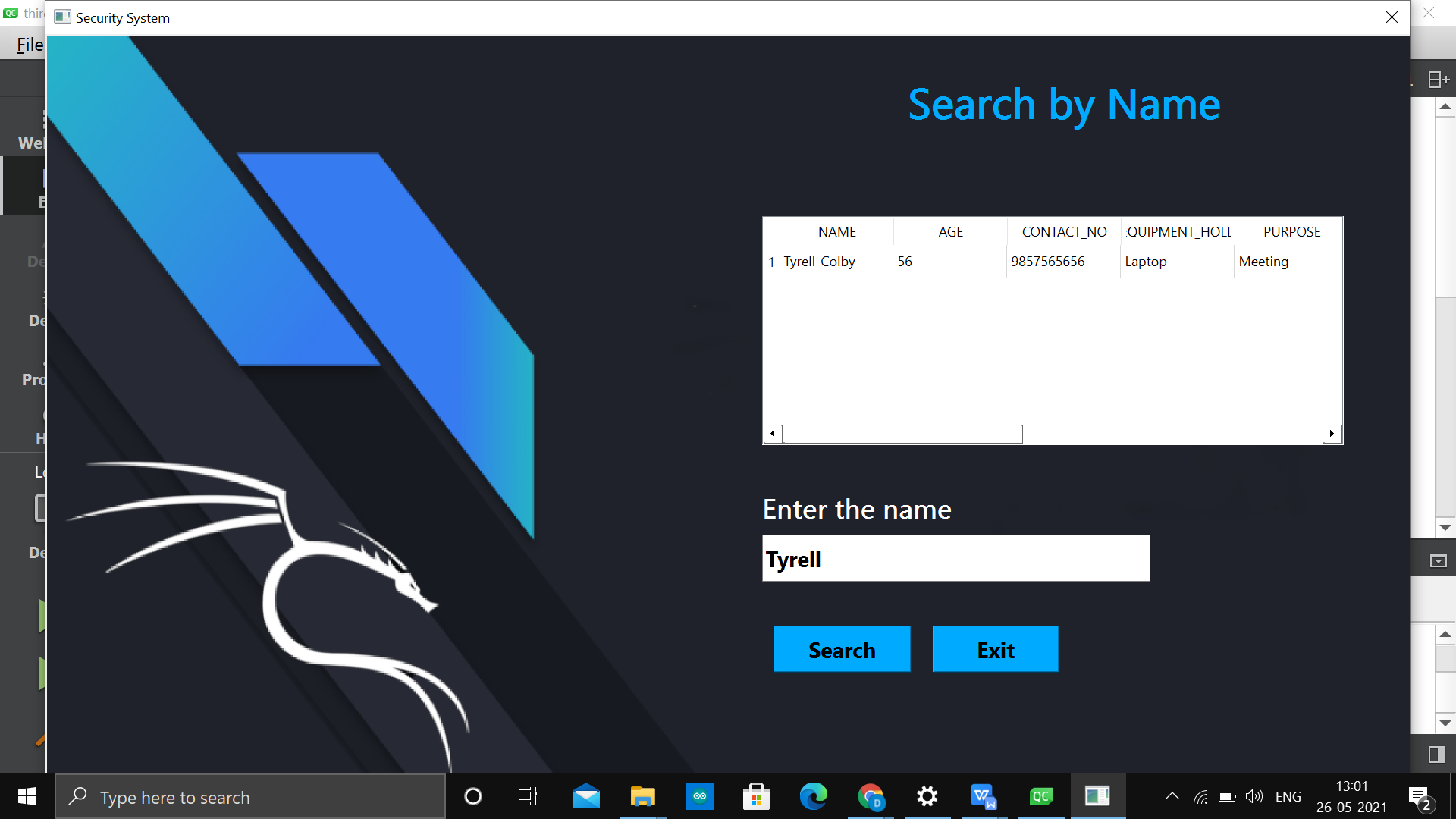


Search window (Visitor)

**Explanation:**

When the user clicks on the search button of the previous window the window shown in the above image gets open. This window contains 4 buttons the first one is search by name the second is search by date third is the search by time and forth is exit button. If you want to search the visitor by name search by name button is clicked. And if you want to search the visitor by time search by time button is clicked. If you want to search the visitor in a particular update interval then search by date is clicked. If the user clicks on the exit button then the previous window gets opened.

**Search by Name Window:**

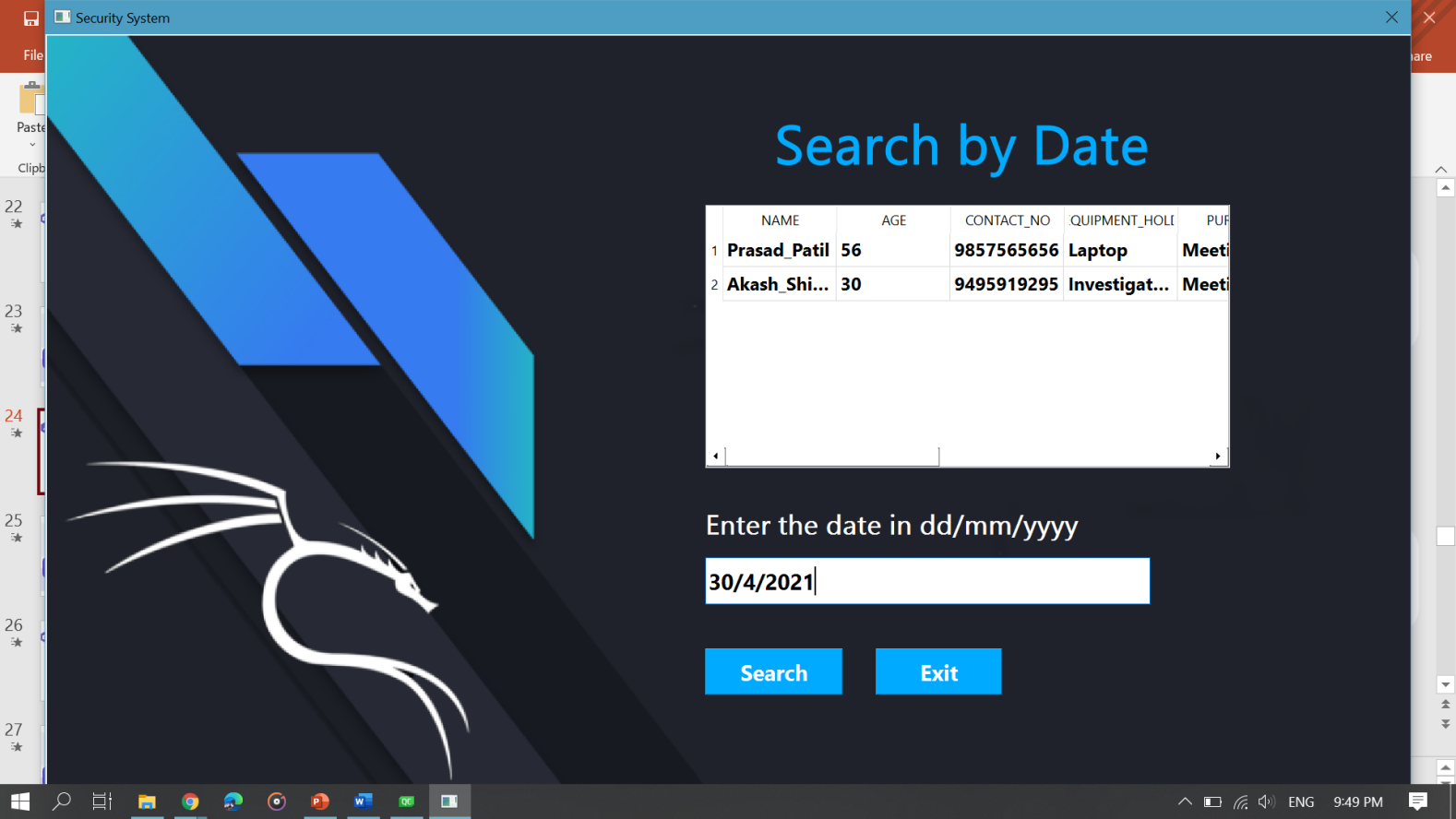


Search by name window (Visitor)

**Explanation:**

After clicking the button named ”Search by name” the above window will open .In this window there are four widgets named “Search button”, “Exit button”, “Line-edit(Entry)” and “TableView” .After clicking Search button the information follow by typing name in Entry, the information regarding that specific visitor will be displayed in TableView.

**Search by Date Window:**

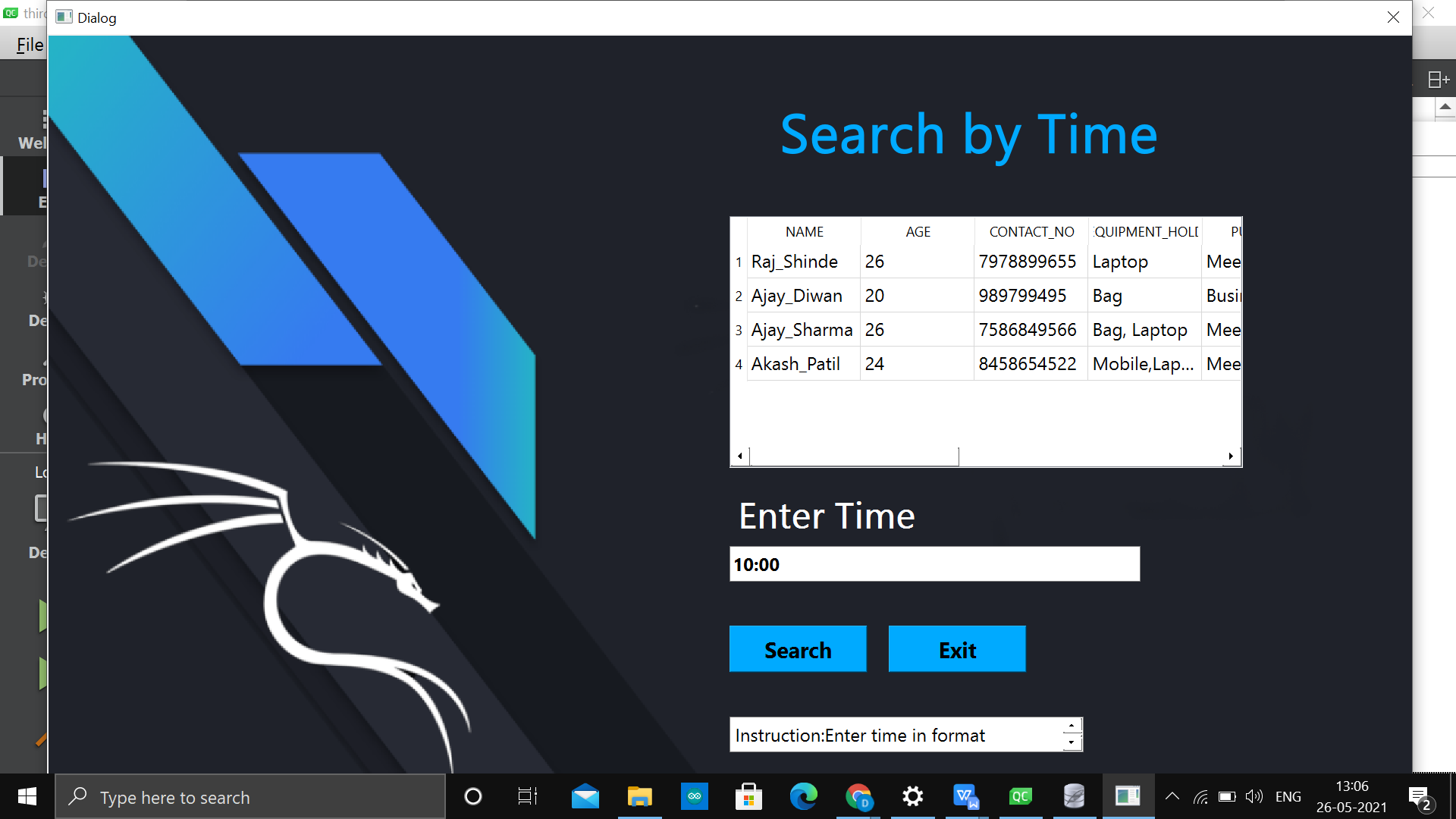


Search by date window (Visitor)

**Explanation:**

The window shown above is the search by date window. When the user enters the date in the entry and clicks on the search button, the date is displayed in the panel as shown in the image. The no of visitors visited on that particular date are shown in the panel.

**Search by date Window:**

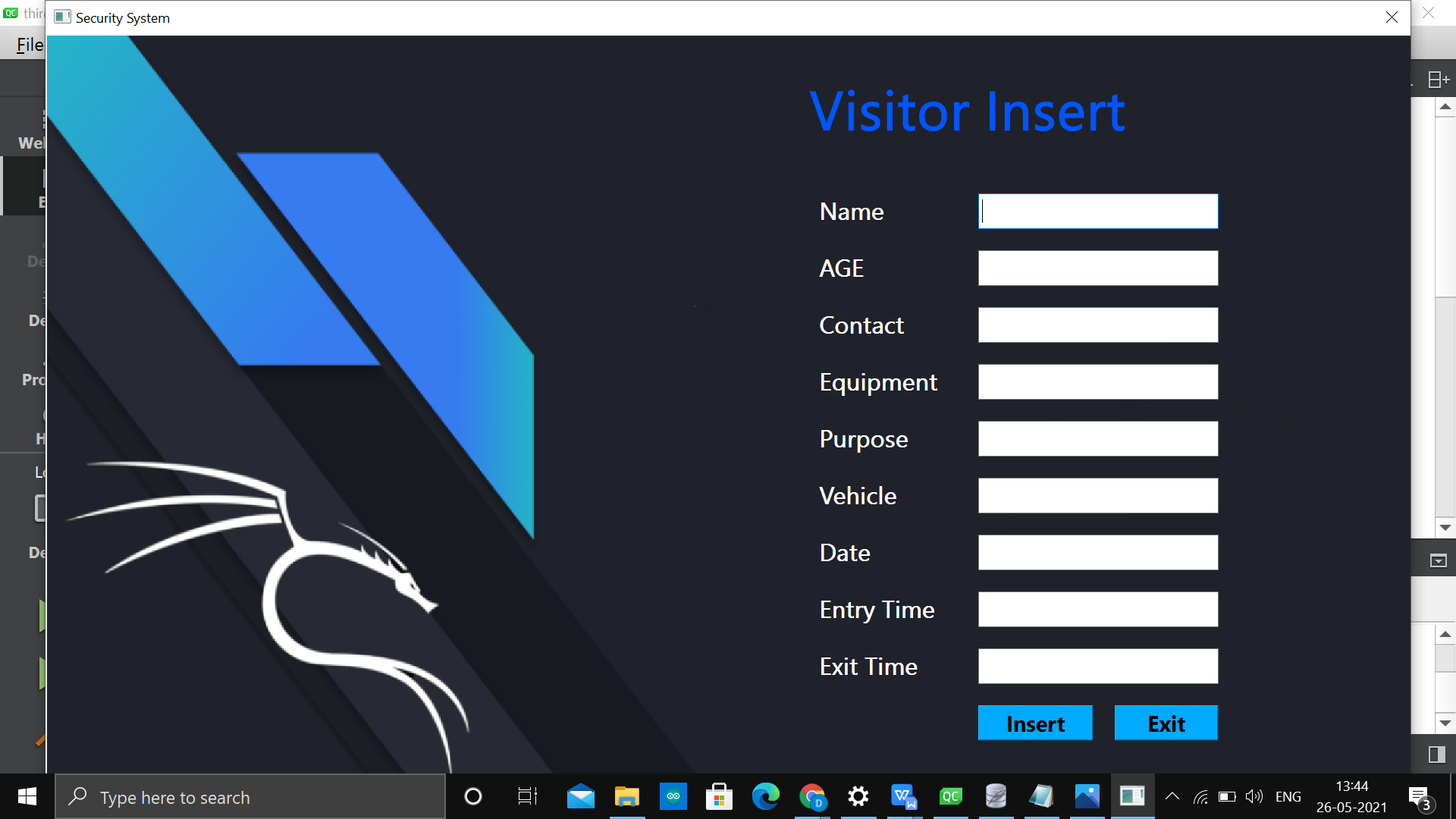


Search by time window (Visitor)

**Explanation:**

The next is the search by time window. In this window when the time interval is entered in the entry then the visitors visited to the company between that time interval are displayed in the panel. When the user clicks on the exit button the previous window gets opened.

**Visitor Insert Window:**



Insert window (Visitor)

**Explanation:**

The last window is the visitor insert window. If a new visitor approaches the company then this window is opened. The window consists of the name, age, contact, equipment, purpose, vehicle, date, entry time and exit time entries. When all the entries are entered and the insert button is clicked then that data of the visitor is saved to the database. When the exit button is clicked then the previous window gets opened.

**Future scope:**

* For the project, a graphical user interface can be developed.
* This system is designed for small engineering firms; however, we will develop a robust system with more functionality and information to manage large data sets for large engineering firms.
* Sensor integration with software
* Creating executable file which is platform independent.

**Conclusion:**

* This project would provide a better security management system for every engineering company that requires all details about their visitors as well as all of the company's employees.
* All data can be safely processed and managed.

**References:**

[www.sqlite.org](http://www.sqlite.org)

SQLite YouTube tutorials

QT YouTube tutorials

<https://doc.qt.io/>