



SIMPLE CONTACT MANAGEMENT SYSTEM REPORT

Higher National Diploma Project Report

The Archives

Chandrakumar Lavanuja

**Information and Communication Technology
Department of Interdisciplinary Studies
University College of Jaffna
University of Vocational Technology
Sri Lanka**

June 2025

SIMPLE CONTACT MANAGEMENT SYSTEM

A project report submitted in partial fulfilment of the requirement for the award of the Higher National Diploma for the Department of Information and Communication Technology, University College of Affna, University of Vocational Technology, Sri Lanka

By C.Lavasujan
(JF /ICT / 24/ 06)

DECLARATION

I do hereby declare that the work reported in this report was exclusively carried out by me with the help and supervision of . It describes the results of my own independent software system except where due reference has been made in the text. No part of this report has been submitted earlier or concurrently for the same or any other higher national diploma.

Date:

.....
Signed by the candidate
C.Lavasujan
(JF/ICT/24/06)



To the best of my knowledge the above particulars are correct.

.....
(Supervisor)

Date:

Acknowledgement

I would like to express my sincere gratitude to everyone who supported me throughout the development of this Simple Contact Management System project.

First and foremost, I am deeply thankful to my Lecturer Mr. Francis Clever Ron for his invaluable guidance, encouragement, and constructive feedback, which helped shape this project from concept to completion.

I would also like to thank my friends who provided helpful suggestions and support during the development process.

Finally, I extend my appreciation to all the online communities, tutorials, and documentation resources that offered technical assistance and inspiration. This project has been a great learning experience, allowing me to enhance my programming skills and deepen my understanding of software development.

This project is a small but meaningful step in my journey as a developer, and I am proud of the progress I've made.

The Archives

Abstract

This project presents the development of a **Simple Contact Management System (CMS)**, a functional desktop application engineered to efficiently store, manage, and retrieve personal and professional contact information. The primary objective was to create a reliable and **user-friendly interface** enabling core operations: adding, updating, deleting, and searching for contact details, including names, phone numbers (Tell No), email addresses, Date of Birth, and Address.

The system was developed utilizing **Visual Studio** with the **C# programming language** and relies on **Windows Forms** for the graphical user interface (GUI). Data persistence is handled by a **SQL Server Management System (SSMS)**, where a dedicated database ("lavandb") and a "Contact" table were created. The structured programming approach emphasized usability, functionality, and security.

The application includes several basic, yet critical, functionalities:

- **Secure Login Interface:** The system starts with a login form to give access only to authorized users (Admin).
- **Password Masking:** A security measure where the password input is masked with the star symbol for privacy.
- **Authentication and Validation:** If incorrect login details are entered, the user is warned with a message box and denied entry to the Dashboard.
- **Menu-Driven Dashboard:** Upon successful login, the user is directed to a Dashboard with clear options for Add Contact and View Contacts.
- **CRUD Operations:** The system supports adding new records via the "Enter Details" form and allows users to view, search, and delete existing contact details. Search functionality is specifically implemented to search by the First name.

Conclusion and Future Enhancements

The CMS project successfully demonstrates the effective use of C# and Windows Forms for building a foundational desktop application that effectively manages contact information. Testing confirmed the application's accuracy in handling contact data, making it a practical tool for individuals or small organizations. This project provides a solid basis for future development, with planned enhancements including: integrating a Cloud Database for better scalability and persistence, further GUI Enhancements for improved visual design, and implementing Backup and Restore options to manage data export and import.



Chapters



The Archives

Table of Figures

The Archives



The Archives

Chapter 1

Introduction

The Contact Management System (CMS) is a simple yet highly functional desktop application developed using C# and Windows Forms. It is strategically designed to help users, particularly individuals or small professional entities, efficiently manage their personal or professional contact information. This is achieved through a structured approach and an intuitive graphical user interface (GUI). The primary goal of this system is to provide a robust, user-friendly solution for storing, retrieving, and organizing contact details in a reliable manner.

System Access and Security

To maintain data integrity and security, the application begins with a secure Login Form. This initial authentication step ensures that only the designated administrator ("admin") can access the core functionalities of the contact management system. The user is prompted to "Enter Login Details" by providing their UserName and password into designated text fields. The form also includes Login and Clear buttons for interaction, optimizing the user experience. The implementation of this secure access point is crucial for protecting sensitive contact data from unauthorized viewing or modification.

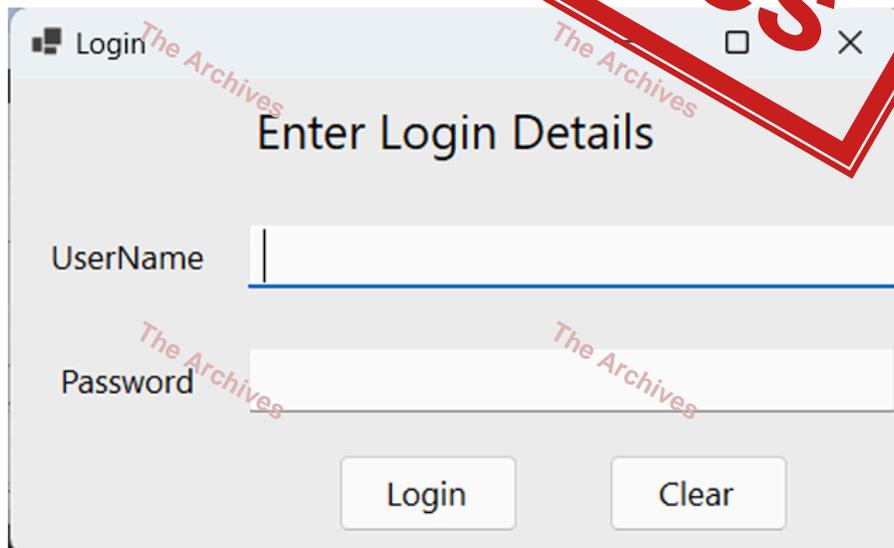


Figure 1.1: Login form with clear button.

Dashboard and Core Functionality

Upon successful authentication and login, the user is immediately directed to the main **Dashboard**. This serves as the primary navigation hub and clearly presents the two main operational options available to the administrator: **Add Contact** and **View Contacts**.

The **Add Contact** feature is designed to allow the user to input and save new contact details. This includes essential fields necessary for comprehensive record-keeping, such as the contact's name, phone number, and email address.

The **View Contacts** option provides the functionality to search for and browse through existing contacts. This feature is vital, as it enables the user to quickly locate and review any stored information, significantly improving efficiency compared to manual record systems.

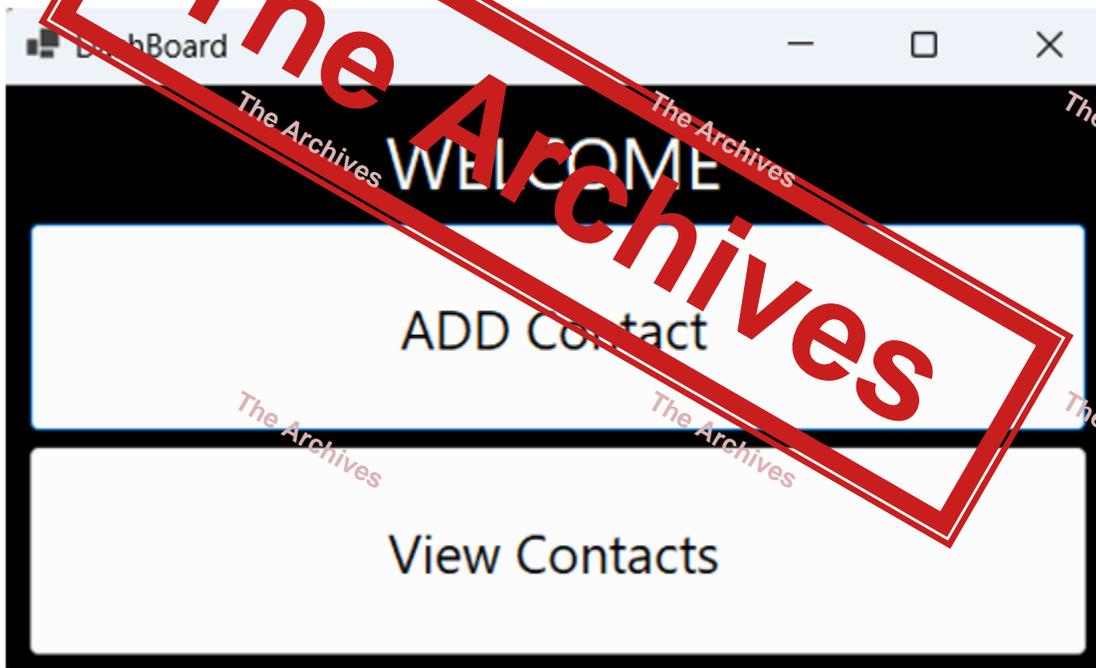


Figure 1.2: Dashboard

Adding New Contact Details

When the Add Contact option is selected from the Dashboard, the "InsertData" form, titled "Enter Details," appears. This form is the dedicated interface for the administrator to input and save comprehensive details for a new contact.

The design of the "Enter Details" form focuses on capturing all necessary contact information in a structured manner, including:

- **First Name**
- **Last Name**
- **Tel. No (Telephone Number)**
- **Email**
- **Date of Birth (DOB)**, using a date picker control
- **Address**

The Archives

The form provides an Add Contact button to save the entered data into the backend SQL database and a Logout link for secure exit.

The screenshot shows a web browser window titled "InsertData" with a form titled "Enter Details". The form contains the following fields and elements:

- First Name:
- Last Name:
- Full Name:
- Email:
- Date of Birth: (with a calendar icon)
- Address:
- Logout: [Logout](#) (in red text)
- Add Contact: (in green text)

Figure 1.3: Add contact.

Database Implementation

The persistence and secure storage of all contact data are managed using **SQL Server Management System (SSMS)**. The system utilizes a dedicated database named "**lavandb**", which contains a crucial table specifically created to store all contact records, aptly named "**Contact**".

The structure of the **dbo.Contact** table is defined by several columns, ensuring all necessary contact fields are accurately stored and managed. The columns are:

- **Fname** (varchar(100), nullable)
- **Lname** (varchar(100), nullable)
- **Telno** (varchar(50), nullable)
- **Email** (varchar(100), nullable)
- **DOB** (date, nullable)
- **Adr** (varchar(150), nullable) (representing the address)

This structured database design, as viewed in the Object Explorer of SSMS, is essential for the system's ability to efficiently retrieve, query, and manipulate contact records using SQL commands.

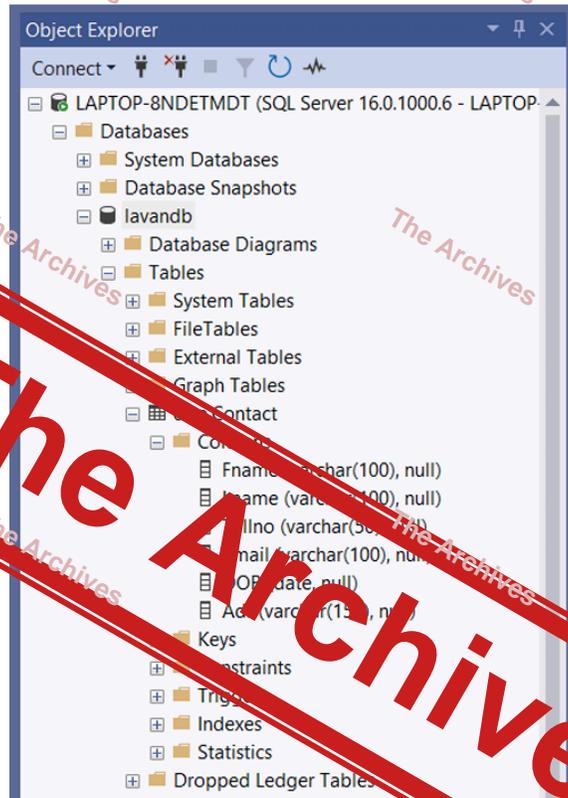


Figure 1.4: Database on SSMS

The SQL environment allows the administrator to execute commands, such as `select * from Contact;`, to view or verify the data contained within the table. This database setup underpins the entire functionality of the Contact Management System, ensuring data is stored reliably and can be retrieved quickly.

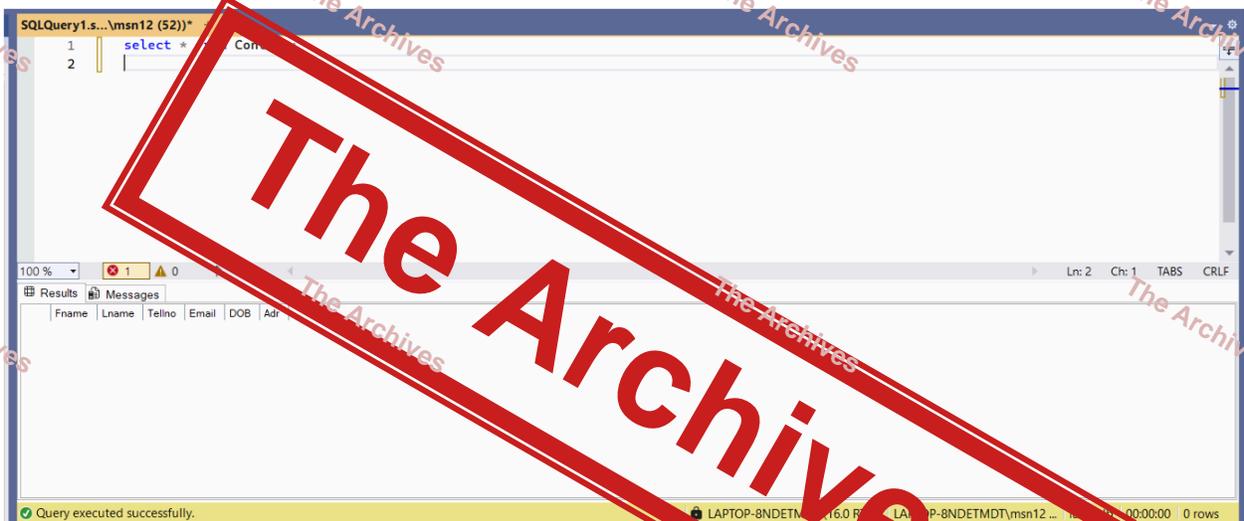


Figure 1.5: Table of the Database.

Chapter 2

Literature review

The Role of Contact Management Systems

Contact management systems have historically played a **vital role** in organizing and maintaining both personal and professional relationships. In the past, contact information was typically stored **manually**, often resulting in significant **inefficiencies**, potential **data loss**, and substantial **difficulty in retrieval**. With the progressive advancement of software development technologies, **digital contact management systems** have emerged as the standard, offering reliable tools for efficiently **storing, accessing, and managing** essential contact data.

Key Design Principles from Literature

The current literature and existing software solutions consistently highlight two major principles critical to the success of contact management applications: **user-friendly interfaces** and **secure data handling**.

1. User Experience and Interface Design

Research within the field of **human-computer interaction (HCI)** strongly suggests that the implementation of **Graphical User Interfaces (GUIs)** significantly **enhances the user experience**. Furthermore, a well-designed GUI actively works to **reduce the learning curve** for non-technical users, making the system immediately accessible and usable. The present system addresses this by utilizing **Windows Forms** to create a clear interface, including a simple **Dashboard** (Figure 1.2) and a straightforward **Add Contact** form (Figure 1.3).

2. Security and Data Protection

Security is discussed as another **critical aspect** in the literature, particularly in applications dealing with sensitive personal information. Features like **password masking** and robust **user authentication** are identified as essential requirements for protecting contact data. This project incorporates these best practices by implementing a mandatory **login system** (Figure 1.1) with **masked password input**. This ensures that only **authorized users** can successfully access and operate the application, thereby safeguarding the stored contact details.

Project Alignment and Contribution

This Simple Contact Management System project is built upon these established and recognized software design principles. It integrates a **secure login form**, a **clean dashboard interface**, and **core CRUD functionalities** such as adding and viewing contacts. By employing C# and the **SQL Server Management System** for structured data storage (Figure 1.4), the project aligns with best practices in software design and offers a practical, reliable solution for basic contact management needs.



Chapter 3

Methodology

The development of the Simple Contact Management System was executed following a **structured and iterative approach**. The core focus throughout the process was rigorously maintained on three critical aspects: **usability**, **functionality**, and **security**.

Security Implementation

A key component of the methodology involved securing access to the application and protecting sensitive data.

- **Password Masking:** In terms of security, the system implements a feature where the password input in the Login form is **encrypted or masked**. This masking uses the **star symbol** to hide the entered characters, thereby ensuring privacy and preventing shoulder surfing.

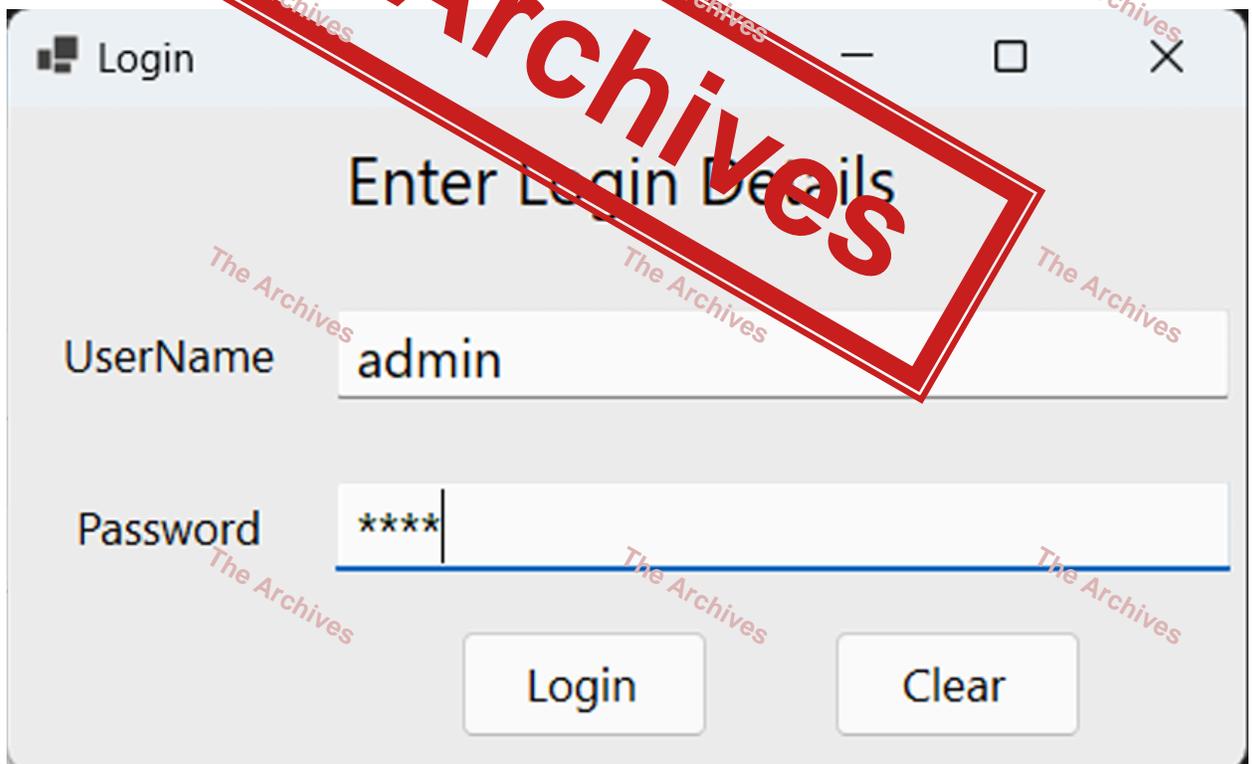


Figure 3.1: Password masking

- **Authentication and Error Handling:** The system performs validation checks during the login process. If the administrator enters an incorrect **Username or Password**, the user is immediately **warned**. This warning is delivered via a **message box** indicating "Invalid Login Details," and the system actively **prevents** unauthorized entry into the Dashboard.

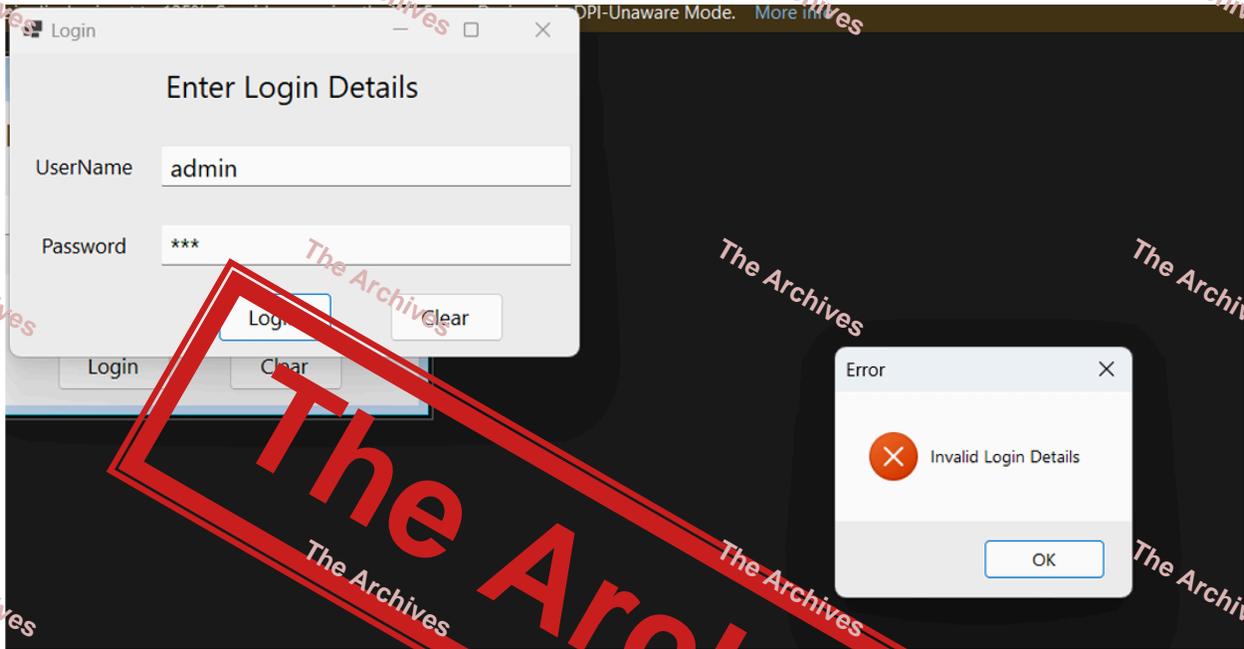


Figure 5.2: Error Message.

Contact Management Flow

The core functionality of the system is accessed after successful authentication.

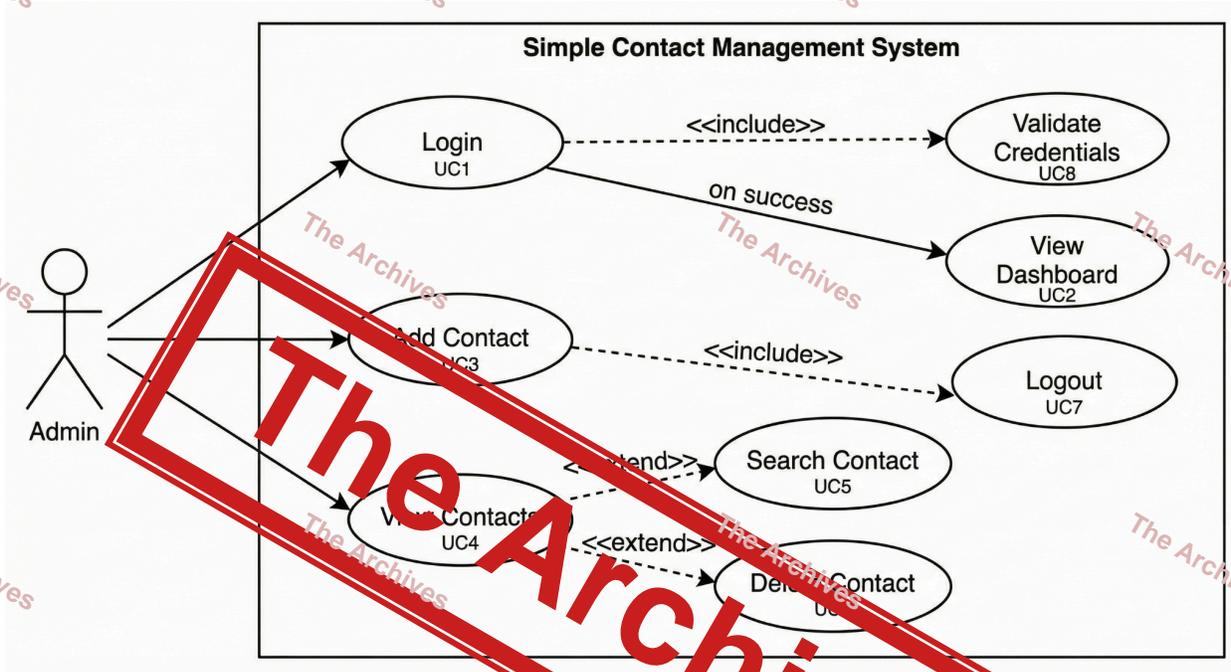
- **Dashboard Access:** After a successful login, the administrator is presented with the **Dashboard** (Figure 1.2). This interface provides the two primary options: **Add Contact** and **View Contacts**.
- **Adding Contacts:** Selecting the **Add Contact** option opens the "Enter Details" form (Figure 1.3), allowing the admin to input a person's details.
- **Viewing and Searching Contacts:** By clicking the **View Contact** option, the admin can access the interface where they can **view all contact details**. Crucially, the system includes a **search interface** that allows the admin to **search all contact details specifically using the First name**. This view also typically includes options to **Delete** contacts, which is a core part of the management functionality.



Figure 3.3: Contact Details

Chapter 4

Use case diagram for this project



This diagram is a Use Case Diagram for This System. It illustrates the available functionalities of the system and how a user interacts with them.

1. System Overview & Actor

- **System Boundary:** The large rectangle labeled "Simple Contact Management System" represents the scope of the application. Everything inside the box is a feature provided by the software.
- **Primary Actor:** The stick figure on the left, labeled **Admin**, represents the user interacting with the system. In this specific design, the Admin is the only person authorized to perform actions.

2. Core Interactions

The Admin has direct access to three main starting points in the system (indicated by the solid arrows from the Actor):

- **Login (UC1):** The entry point to the system.
- **Add Contact (UC3):** The process of creating a new contact entry.
- **View Contacts (UC4):** The process of browsing the list of existing contacts.

3. Detailed Interactions, Flows & Relationships

This diagram uses specific UML relationships (arrows with labels) to define complex behaviors. Here is how they work in this system:

- The Login Process (Includes & Sequence)

When the Admin attempts to **Login (UC1)**, two things happen based on the connections:

- **Mandatory Validation (<<include>>):** The <<include>> arrow pointing to **Validate Credentials (UC8)** means that *every time* the Login use case runs, the system *must* also run the Validate Credentials process behind the scenes to check username and password. Login cannot complete without this validation step.
- **Success Sequence:** The solid arrow labeled "on success" indicates a workflow. If the Login and Validation are successful, the system automatically moves the Admin to the **View Dashboard (UC2)** screen.

- Viewing Contacts (Extension Points)

The **View Contacts (UC4)** use case is a "base" activity that can be expanded upon by optional actions. The arrow points *from* the optional task *towards* the base task:

- **Optional Search (<<extend>>):** While viewing the contact list, the Admin has the option to **Search Contact (UC5)**. This doesn't happen automatically; it only happens if the Admin decides to use the search feature to filter results.
- **Optional Deletion (<<extend>>):** Similarly, while viewing contacts, the Admin can choose to **Delete Contact (UC6)**. This is an optional extension of the viewing process.

- The Logout Workflow (Mandatory Inclusion)

According to this specific diagram design, the **Logout (UC7)** process is tightly coupled with finishing specific tasks.

- The <<include>> arrows originating from both **Add Contact (UC3)** and **View Contacts (UC4)** point to **Logout (UC7)**.
- **Interpretation:** This implies that the system is designed such that once an Admin finishes the specific task of adding a contact or viewing the contact list, the system automatically forces a Logout action as the concluding step of that task.



Chapter 5

Conclusions and Future Works

Project Conclusion

The Simple Contact Management System (CMS), developed using the C# programming language and the Windows Forms framework, successfully fulfilled all defined requirements and primary objectives. The resulting application provides a robust and reliable platform, demonstrating proficiency in the essential functions of contact data management. The system effectively enables users to:

- Securely **log in** through an authenticated interface, upholding basic security requirements.
- Add new contacts using a dedicated "Enter Details" form, capturing comprehensive information including names, phone numbers, and addresses.
- Efficiently **view, search, and delete** existing contact records via a centralized interface, ensuring easy maintenance of the database.

The project serves as a clear and practical demonstration of the effective use of **Windows Forms** for developing intuitive desktop applications. Furthermore, it highlights the critical importance of integrating **basic data management capabilities** with strong **user authentication** early in the software design process. The CMS, therefore, provides a **solid foundation** for managing contact information in a structured and accessible manner, making it a highly practical and useful tool for **personal organization** or supporting the needs of a **small-scale professional environment**. The successful deployment and testing affirm the system's reliability and accuracy in handling transactional contact data.

Future Works

While the current system is fully operational, several significant enhancements are planned to further advance its functionality, scalability, and accessibility, ensuring its continued relevance:

1. **Cloud Database Integration:** The most impactful future enhancement involves transitioning the current local SQL Server Management System (SSMS) database to a modern **online or Cloud Database** solution. This shift is necessary to ensure greater **scalability** to handle increasing data volumes and, more importantly, to guarantee robust **data persistence** and accessibility across different devices and geographical locations. This migration would enhance the utility for users requiring remote data access.
2. **Graphics User Interface (GUI) Enhancements:** The project will benefit from focused improvements to the visual design and user experience of the Windows Forms interface. Future work will concentrate on improving the **visual design** elements and increasing the application's overall **responsiveness** to user interactions, making the system more modern and engaging.
3. **Backup and Restore Functionality:** To significantly boost the system's robustness and data safety, new features will be introduced to facilitate easy **exporting and importing** of contact data. The essential **Backup and Restore** capability will provide the administrator with a secure method for archival and data recovery, minimizing the risk of permanent data loss.
4. **Advanced Search and Filtering:** Expanding the current search functionality (which currently focuses on First Name) to include multiple fields (e.g., Email, Address, Last Name) and complex filtering options would drastically improve contact retrieval efficiency.

References

YouTube Tutorials

- C# Windows Forms: Connect to SQL Server and Perform CRUD Operations (Create Read Update and Delete):
 - **URL:** https://www.youtube.com/watch?v=T8Mq8_Y7tgY
 - *Relevance:* Directly covers connecting C# Windows Forms to SQL Server and implementing all CRUD operations, which are central to your CMS.
- 1 Hour Tutorial | Learn C# and SQL Server by Building Windows Application
 - **URL:** <https://www.youtube.com/watch?v=uJkTLHU7zI>
 - *Relevance:* A comprehensive tutorial focused on building a complete Windows Application using C# and SQL Server, fitting the scope of your project.
- Step-by-Step C# Book Management System: Part 1 - Login Form & Database
 - **URL:** <https://www.youtube.com/watch?v=OsjEGl4THTg>
 - *Relevance:* Specifically covers the creation of a **secure login form** and setting up the database in a CMS, which corresponds to Figure 1.1 and 1.4 in your report.
- C# Contact Information System Project With Source Code (using MySQL, but the concepts apply to SQL Server)
 - **URL:** <https://www.youtube.com/watch?v=agdRWqgM1L0>
 - *Relevance:* Demonstrates a full "Contacts Management System" project, including the main form, adding contacts, and editing/removing contacts.

Others

- <https://www.zendesk.com/sell/crm/contact-management-software/>
- <https://learn.microsoft.com/en-us/dotnet/csharp/>
- <https://www.scribd.com/document/379538942/Contact-Management-System>
- <https://www.jetir.org/papers/JETIR2206709.pdf>

