Contact Management System

MSCS542L

Section 816

Undercover Martyn

Marist College

School of Computer Science and Mathematics

Submitted To: Dr. Reza Sadeghi

Spring 2025

Table of Contents

Table of Figures

Project Report of Contact Management System

Team Name

Undercover Martyn

Team Members

1. Reece Schenck reece.schenck1@marist.edu (Team Head)

Description of Team Members

1. Reece Schenck

Hello, I am Reece Schenck. I am a computer science and cybersecurity major and I am the manager of the club volleyball team here at Marist. As the only member of this group that makes me the team head. I chose to work alone as there is only one other student in the class.

Project Objective

PROJECT TITLE: CONTACT MANAGEMENT SYSTEM

Summary: THE CONTACT MANAGEMENT SYSTEM (CMS) RECORDS VARIED INFORMATION, SUCH AS CELL NUMBER, FAX NUMBER, HOME NUMBER, EMAIL, AND ADDRESS. THE USERS SHOULD SECURELY ADD RECORDS & UPDATE THEM. THE CMS WILL STORE THE DATA OF DIFFERENT USER TYPES IN DISTINCT SQL TABLES. THIS SYSTEM SHOULD AT LEAST SUPPORT THE FOLLOWING CAPABILITIES:

1. REQUESTING ADMIN USER AND PASSWORD FOR LOG IN (A STRING OF AT LEAST 8 CHARACTERS)

2. CHANGE THE ADMIN USER AND PASSWORD

3. ADMIN USER SHOULD BE ABLE TO ADD A USER TO CMS BY CREATING A NEW USERNAME AND PASSWORD FOR NORMAL USERS, WHO ARE NOT ABLE TO DEFINE OR REMOVE A USER

4. ADMIN USER SHOULD BE ABLE TO REMOVE A USER FROM CMS BY REMOVING HIS/HER USERNAME, PASSWORD, AND HIS/HER OTHER CORRESPONDING DATA

5. EACH USER SHOULD BE ABLE TO:

a. ADD MULTIPLE SETS OF CONTACT INFORMATION WITH THE FOLLOWING DETAILS: FIRST NAME, SURNAME, CELL PHONE, WORKPLACE PHONE, FAX NUMBER, EMAIL ADDRESS, GENDER, AND AGE

b. REMOVE A CONTACT RECORD

c. EDIT THE CONTACT RECORD'S DETAILS

d. SEARCH THROUGH CONTACTS BASED ON ONE OR SEVERAL FEATURES AND LIST THE RESULTS ON THE SCREEN. FOR INSTANCE, IT SHOULD BE ABLE TO RETURN THE CELL PHONE NUMBER OF A SPECIFIC NAME.

6. CMS SHOULD BE A USER-FRIENDLY SOFTWARE, SUCH THAT:

a. IT SHOWS A WARNING IF A USER TRIES TO INPUT CONTACT INFORMATION WITH A NAME THAT EXISTS IN THE HISTORY.

b. CMS SHOULD SHOW A WELCOME PAGE

c. CMS SHOULD SHOW A MENU OF ALL FUNCTIONS TO THE USER

d. CMS SHOULD PROVIDE THE REPORTS IN A TABULAR FORM

e. CMS SHOULD PROVIDE AN EXIT FUNCTION

Review Related Works

As seen throughout similar Contact Management Systems, they typically are not all encompassing and can be hard to use. In particular, when using HubSpot(2), Pipedrive(5), and Insightly(6), it is clear that they all specialized in some way over the other, each with their own positive and negative aspects.

HubSpot offers many positive features. Some of which include its very user friendly interface offered completely for free. It also offers automation and easy integration with other business solutions. Along with these positives, HubSpot also has some drawbacks. The main Issue is that some advanced features are locked behind the paid version. Even so, regardless of the version of HubSpot being used, there is a severe lack of customization to each individual user's unique business models.

Diving into Pipedrive, it offers similar features as HubSpot except with a more tailored financial sales focus. For the positives offered, Pipedrive is very customisable with features like notifications and reminders for activities or follow up appointments. However, similar to HubSpot, Pipedrive also has limited available advanced features on account of it being tailored specifically

for financial sales. On top of this, it is difficult to integrate Pipedrive without having to use other tools to work around its lack of compatibility.

Finally examining Insightly, it offers numerous positive features. These include a user friendly interface, easy integration, numerous comprehensive features offered such as automation and other tools. Some negatives include the pricing as it is far more expensive than other available options. Similar to HubSpot, Insightly also has very limited customizations for features such as fields and layouts.

The Merits of the Project

The Contact Management System(CMS) has several merits that make it an excellent solution for managing contact information. It allows users to store and manage comprehensive details, including cell numbers, email addresses, fax numbers, home numbers, and physical addresses, ensuring all relevant data is centralized and easily accessible. The system enforces security through mandatory admin credentials with a minimum character length of eight and provides robust access control. Admins can manage user accounts by adding or removing users and assigning credentials, while normal users are restricted to contact management tasks, ensuring data security and administrative control.

The CMS offers a range of features for users, such as the ability to add, edit, and remove detailed contact records. Its efficient search filters enable users to quickly locate specific contacts based on fields like name or phone number. The system enhances user-friendliness by providing warnings for duplicate entries, a welcoming interface, menu-driven navigation, and tabular reports for clear and professional data presentation. Additionally, it includes error handling mechanisms and a convenient exit function to streamline operations.

End users benefit from the CMS's centralized and organized data storage, which uses SQL tables for fast and reliable data retrieval. The system prioritizes security and data protection through role-based access and strict login protocols. It is highly adaptable and scalable, catering to both administrators and regular users. The system saves time with its efficient search capabilities and customizable filters while offering professional reporting features for business or personal use.

The CMS is an all-in-one solution, integrating core features that are tailored to user needs. Unlike other tools such as HubSpot, Pipedrive, or Insightly, all of which specialize in certain areas

and thus lack full customization or comprehensive functionality, the CMS balances adaptability, and functionality. It is designed with user ease in mind, featuring a welcoming interface and error handling to reduce the learning curve and maximize productivity. Furthermore, it eliminates dependency on external tools, serving as a reliable standalone solution. By addressing common pain points like limited customization and integration challenges found in competing tools, the CMS delivers a secure, user-friendly, and versatile product that effectively meets diverse end-user needs.

GitHub Repository Address

https://github.com/Reece-Schenck/MSCS542L_816_CONTACT-MANAGEMENT-SYSTEM_Undercover-Martyn

Entity Relationship Model

1. External Models

External Model 1: User and Contact Management

This model focuses on the relationship between users, their login credentials, and their contact information.

- **Entities**:
  - **User**: Represents the individuals who use the CMS.
  - **Role**: Differentiates between Admin and Normal Users.
  - **Login**: Stores the login details for each user.
  - **Contact**: Stores the contact information (name, phone numbers, etc.) of each individual.
  - **Address:** Stores the physical address of a contact
  - **Phone**: Stores different phone numbers associated with contacts.
- **Relationships**:
  - A **User** has a **Login** (1:1).
  - A **User** can have many **Contacts** (1:N).
  - A **Contact** can have many **Phones** (1:N).
  - A **Contact** can have one **Address** (1:1).
  - A **User** has one **Role** (1:1).
  - A **Contact** belongs to one **User** (N:1).

External Model 2: Communication and Subscriptions

This model adds elements for communication and subscription tracking services.

- **Entities**:
  - **Email**: Stores email addresses associated with contacts.
  - **EmergencyContact**: Represents a person to contact in case of an emergency.
  - **Message**: Represents communication between users.
  - **Subscription**: Tracks the subscriptions that users have.
- **Relationships**:
  - A **User** can have many **EmergencyContacts** (1:N).
  - A **Message** is sent between a **Sender** and a **Receiver** (M:N).
  - A **User** can have many **Subscriptions** (1:N).
  - A **Contact** can have many **Emails** (1:N).

Selections:

Selecting the Entities

Entities represent real-world objects or concepts that are relevant to the CMS. I selected the entities based on the core operations and features of the system:

- **User**: The individuals who interact with the system, such as admins or normal users.
- **Role**: A classification for users, differentiating between admin and regular users.
- **Login**: Stores the login credentials for each user.
- **Contact**: A user's stored contact information, including personal details.
- **Phone**: Each contact can have multiple phone numbers, such as work, mobile, or fax.
- **Address**: Stores the physical address of a contact.
- **Email**: Represents email addresses associated with contacts.
- **EmergencyContact**: Allows users to store emergency contacts.
- **Message**: Tracks communication between users in the system.
- **Subscription**: Represents different service plans or subscriptions that users might have.

Selecting the Attributes

For each entity, I selected relevant attributes that capture all necessary details for effective functionality in the CMS:

- **User**: Username, password, email, role, and account creation time.
- **Role**: Role name (e.g., Admin, Normal), description, permissions, and creation time.
- **Login**: Username, password, last login time.
- **Contact**: First name, last name, gender, age, and associated user ID.

- **Phone**: Phone type (work, mobile, fax), phone number, and primary flag.
- **Address**: Street, city, state, zip code, and country.
- **Email**: Email address, whether it is primary, created time, and verified status.
- **EmergencyContact**: Name, phone number, relationship to the user, and activation status.
- **Message**: Sender ID, receiver ID, message content, timestamp, and read status.
- **Subscription**: Subscription type, start date, end date, and active status.

Selecting the Relationships

The relationships define how entities are connected to one another. I chose relationships based on real-world interactions within the CMS system:

- **User** → **Login**: Each user **has** one login, and the login is connected to the user.
- **User** → **Contact**: Each user can **have** many contacts, representing the user's personal database of contacts.
- **Contact** → **Phone**: Each contact can **Contain** multiple phone numbers.
- **Contact** → **Address**: Each contact may **Contain** one address.
- **Contact** → **Email**: Each contact can **Contain** multiple emails.
- **User** → **EmergencyContact**: Each user can **have** one or more emergency contacts.
- **User** → **Subscription**: Each user can **subscribe** to a specific service or subscription plan.
- **Message** → **Sender & Receiver**: Messages are **Sent** and **Received** between users.

Selecting the Participations

I decided on partial or total participation based on the data requirements for each relationship:

- **Partial Participation**: A **Contact** may or may not have an address.
- **Total Participation**: A **Contact** must have at least one phone number.

Selecting the Cardinalities

Cardinalities define how many instances of one entity can be related to instances of another entity. I selected cardinalities based on how entities interact:

- **1:1 Cardinality**: A **User** has one **Login**.
- **1:N Cardinality**: A **User** can have multiple **Contacts**, and a **Contact** can have multiple **Phone** numbers.
- **M:N Cardinality**: A **Message** can be sent between multiple **Senders** and **Receivers** (users).

Descriptions:

Entities

1. **User**
    - Represents the individuals interacting with the CMS, either admins or regular users.
    - **Attributes**: Username, password, email, role, and account creation time.
2. **Role**
    - Differentiates users into categories like admin or regular users.
    - **Attributes**: Role name, description, permissions, and creation time.
3. **Login**
    - Stores login credentials for each user.
    - **Attributes**: Username, password, and last login time.
4. **Contact**
    - Stores detailed information about a user's contact, such as name and personal info.
    - **Attributes**: First name, last name, gender, age, and associated user ID.
5. **Phone**
    - Represents phone numbers associated with a contact (work, mobile, fax).
    - **Attributes**: Phone type (e.g., work, mobile), phone number, and primary flag.
6. **Address**
    - Stores the physical address of a contact.
    - **Attributes**: Street, city, state, zip code, and country.
7. **Email**
    - Represents email addresses associated with contacts.
    - **Attributes**: Email address, primary flag, creation date, and verified status.
8. **EmergencyContact**
    - Represents important contacts to call in case of emergency for a user.
    - **Attributes**: Name, phone number, relationship to user, and active status.
9. **Message**
    - Stores messages exchanged between users.
    - **Attributes**: Sender ID, receiver ID, message content, timestamp, and read status.
10. **Subscription**
    - Represents the subscription details for each user.
    - **Attributes**: Subscription type, start date, end date, and active status.

Relationships

1. **User** → **Login** (1:1)
    - A user **has** one login record, linking their credentials.
2. **User** → **Contact** (1:N)
    - A user can **have** many contacts, representing their contact information.
3. **Contact** → **Phone** (1:N)
    - Each contact can **contain** multiple phone numbers (work, mobile, fax).
4. **Contact** → **Address** (1:1)
    - A contact may **contain** one address.

5. **Contact** → **Email** (1:N)
    - A contact can **contain** multiple email addresses.
6. **User** → **EmergencyContact** (1:N)
    - A user can **have** multiple emergency contacts.
7. **User** → **Subscription** (1:N)
    - A user can **subscribe** to multiple services.
8. **Message** → **Sender & Receiver** (M:N)
    - Messages are **Sent** and **Received** between users.

Participations

1. **Partial Participation**:
    - **Contact** → **Address**: A contact may or may not have an address.
2. **Total Participation**:
    - **Contact** → **Phone**: A contact must have at least one phone number.

Cardinalities

1. **1:1 Cardinality**:
    - **User** → **Login**: A user has one login record.
2. **1:N Cardinality**:
    - **User** → **Contact**: A user can have multiple contacts.
    - **Contact** → **Phone**: A contact can have multiple phone numbers.
3. **M:N Cardinality**:
    - **Message** → **Sender & Receiver**: A message can be sent between multiple senders and receivers.

Advanced Features:

- **Composite Attribute**: Group Street, City, State, ZipCode, Country into a composite attribute for **Address**.
- **Multivalued Attribute**: Represent multiple **Phone** numbers for **Contacts**.
- **Derived Attribute**: Include **Age** for **Contact** based on DateOfBirth.

2. Conceptual Model

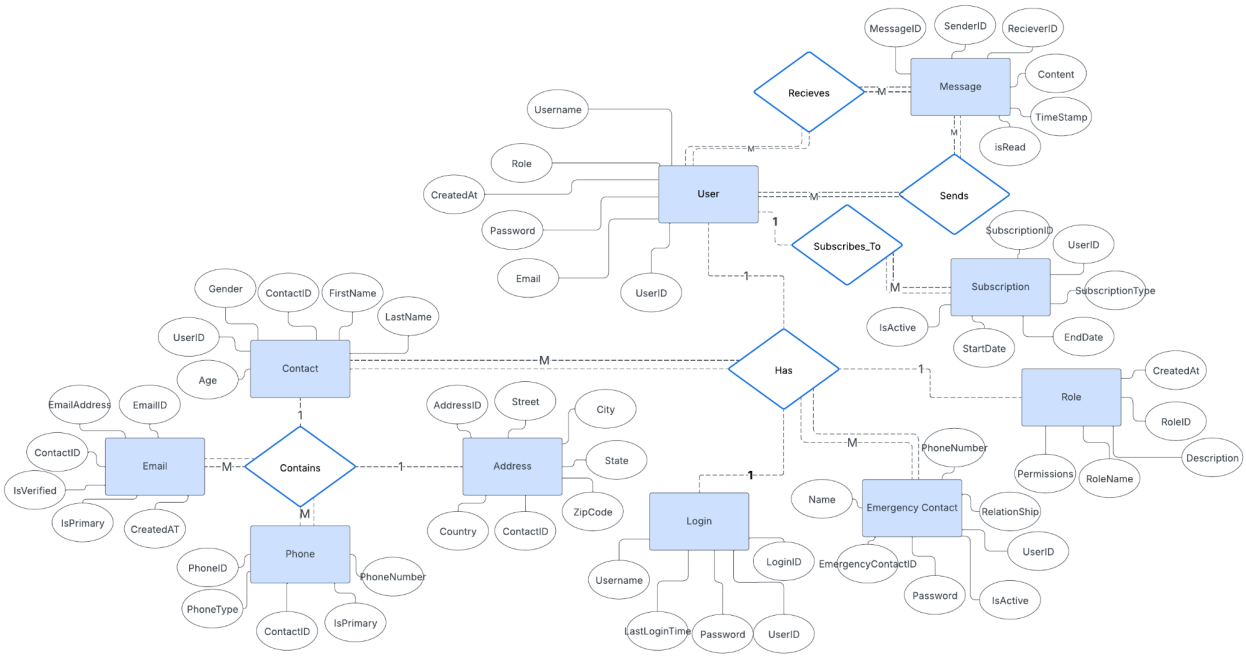
Entities and Attributes

1. **User**
    - Attributes: `UserID` (PK), `Username`, `Password`, `Email`, `Role`, `CreatedAt`
2. **Role**
    - Attributes: `RoleID` (PK), `RoleName` (Admin, Normal), `Description`, `Permissions`, `CreatedAt`
3. **Login**

    - Attributes: `LoginID` (PK), `Username`, `Password`, `UserID` (FK), `LastLoginTime`
4. **Contact**
    - Attributes: `ContactID` (PK), `FirstName`, `LastName`, `Gender`, `Age`, `UserID` (FK)
5. **Phone**
    - Attributes: `PhoneID` (PK), `PhoneType` (Work, Mobile, Fax), `PhoneNumber`, `ContactID` (FK), `IsPrimary`
6. **Address**
    - Attributes: `AddressID` (PK), `Street`, `City`, `State`, `ZipCode`, `Country`, `ContactID` (FK)
7. **Email**
    - Attributes: `EmailID` (PK), `EmailAddress`, `ContactID` (FK), `IsPrimary`, `CreatedAt`, `IsVerified`
8. **EmergencyContact**
    - Attributes: `EmergencyContactID` (PK), `Name`, `PhoneNumber`, `Relationship`, `UserID` (FK), `IsActive`
9. **Message**
    - Attributes: `MessageID` (PK), `SenderID` (FK), `ReceiverID` (FK), `Content`, `Timestamp`, `IsRead`
10. **Subscription**

- Attributes: `SubscriptionID` (PK), `UserID` (FK), `SubscriptionType`, `StartDate`, `EndDate`, `IsActive`

ER Diagram:



Enhanced Entity Relationship Model

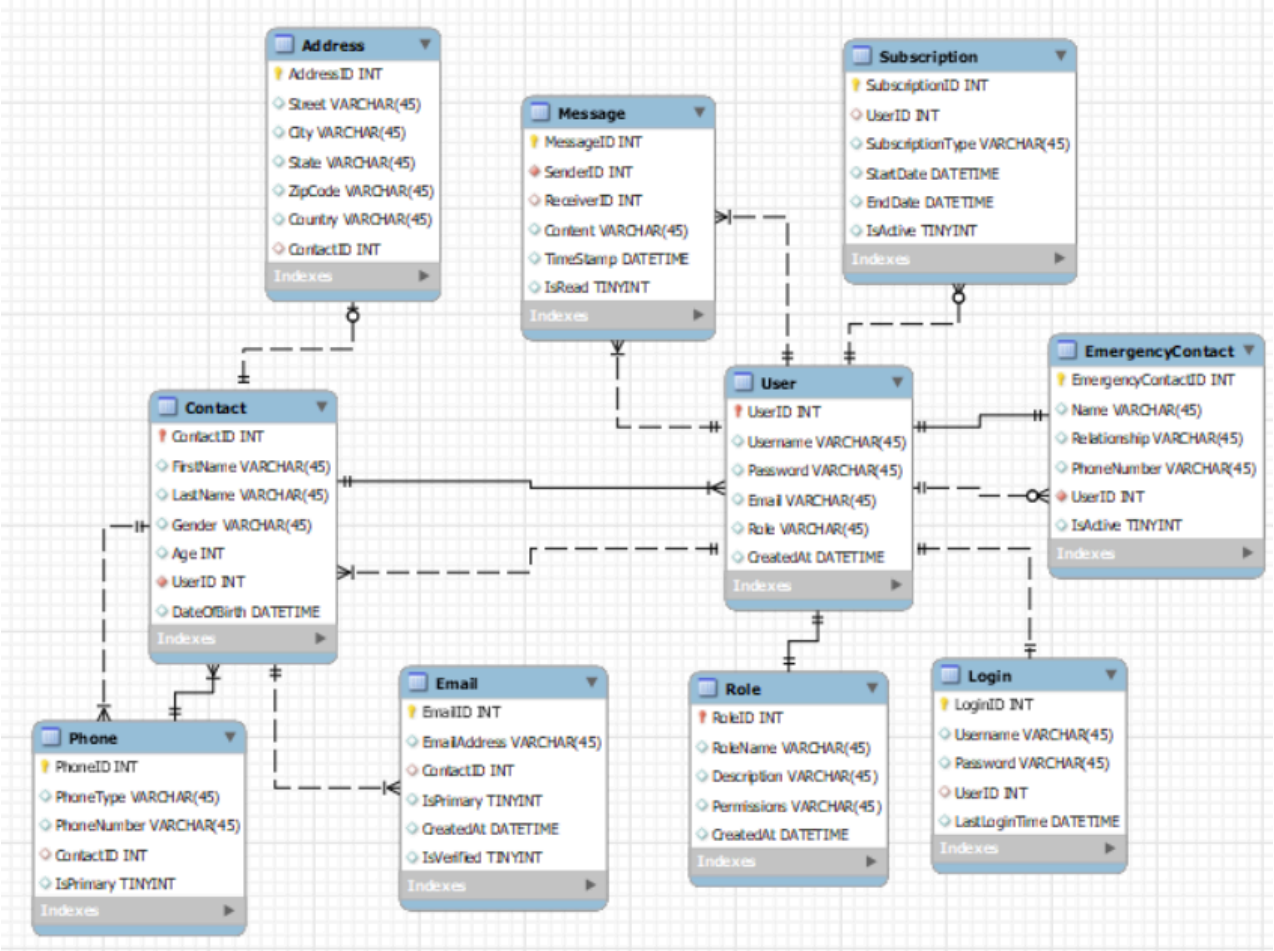
Each entity, such as **User**, **Contact**, and **Phone**, was designed with at least five attributes, including a primary key(PK) to ensure unique identification of records. For example, the **UserID** in the **User** entity and **ContactID** in the **Contact** entity serve as primary keys. Foreign keys(FK) were introduced to establish relationships between entities, such as the **UserID** in the **Contact** entity, which references the **UserID** in the **User** entity.

The relationships in the EER model were defined using 1:1, 1:N, and M:N cardinalities. A 1:1 relationship, such as between **User** and **Login**, ensures that each user has a single login. A 1:N relationship, such as between **User** and **Contact**, allows each user to have multiple contacts. Meanwhile, an M:N relationship, such as between **Message** and **User**(sender and receiver), enables many-to-many interactions where a user can send and receive multiple messages. Total participation was applied where every instance of an entity must participate in a relationship, such as every **Contact** requiring at least one **Phone**. Partial participation was used where relationships are optional, such as not all

Contacts having an **Address**.

The model also incorporates advanced features, including multivalued, composite, and derived attributes. For example, the **PhoneType** attribute in the **Phone** entity supports multiple values like work, mobile, and fax numbers. The **Address** entity was designed as a composite attribute, breaking down into **Street**, **City**, **State**, **ZipCode**, and **Country**. Additionally, a derived attribute, such as **Age** in the **Contact** entity, was calculated based on the **DateOfBirth** attribute.

EER Diagram:



References

(1) HubSpot. "Top Contact Management Software for 2023." *Blog HubSpot*, 2023, https://blog.hubspot.com/sales/contact-management-software.

(2) HubSpot. "Contact Management CRM | Free Tool for Tracking Customer Information." *HubSpot*, 2023, https://www.hubspot.com/products/crm/contact-management.

(3) Capterra. "Best Contact Management Software 2023." *Capterra.com*, 2023, https://www.capterra.com/contact-management-software/.

(4) TechRadar. "Best Contact Management Software in 2023." *TechRadar*, 2023, https://www.techradar.com/best/best-contact-management-software.

(5) Pipedrive. "Contact Management Software for Sales Teams." *Pipedrive.com*, 2023, https://www.pipedrive.com/en/roles/crm-for-contact-managers.

(6) Insightly. "CRM Pricing Plans." *Insightly.com*, 2023, https://www.insightly.com/pricing-plans/.