

PHONEBOOK MANAGEMENT SYSTEM SPECIFICATION

I.INTRODUCTION:

- The **Contact Management System** acts as a powerful assistant, keeping all your essential connections right at your fingertips. More than just a simple phonebook, it is a tool designed to help you manage your digital life more effectively. With just a few simple steps, you can create, update, or organize your contact list, ensuring that every piece of information you need is always ready, anytime and anywhere.

1. Functional Requirement:

- The system must provide essential features to enable users to interact with and manage contact information effectively. The basic functional requirements include:

1.1 Add New Contact:

- Allows users to create new contact records by entering detailed information such as Full Name, Phone Number, Email, Address, and Company. The system shall validate data integrity (e.g., phone number format) before saving it to the database.

1.2 Edit Contact:

- Allows users to modify or update the information of an existing contact in the list. Once the changes are saved, the system will update to the latest data and display the notification: "Update successful."

1.3 Delete Contact:

- Allows users to remove contacts that are no longer needed from the list. To ensure data safety, the system shall require a confirmation prompt before performing a permanent deletion.

1.4 Search Contacts:

- Provides a smart filter that allows users to quickly search based on keywords such as Name or Phone Number. The system will retrieve and display matching results instantly. If no corresponding data is found, the system will respond with the message: "No matching results found."

1.5 Display Contact List:

- Provides an intuitive interface for users to view their entire contact list, sorted alphabetically. Each contact will be displayed with its primary identification information. If the user has no contacts, the system will display an empty state: "The list is currently empty."

1.6 Admin:

- Provides dedicated control tools for administrators to ensure stable system operation. The Admin has the authority to manage user accounts (such as locking or unlocking accounts).

2. Non-Functional Requirement:

- In addition to the core features, the system must meet the following operational and quality standards:

2.1 Security:

- Protect personal data with secure login and strict access control between Users and Admins.

2.2 Usability:

- Simple, intuitive UI with clear system notifications and feedback.

2.3 Compatibility:

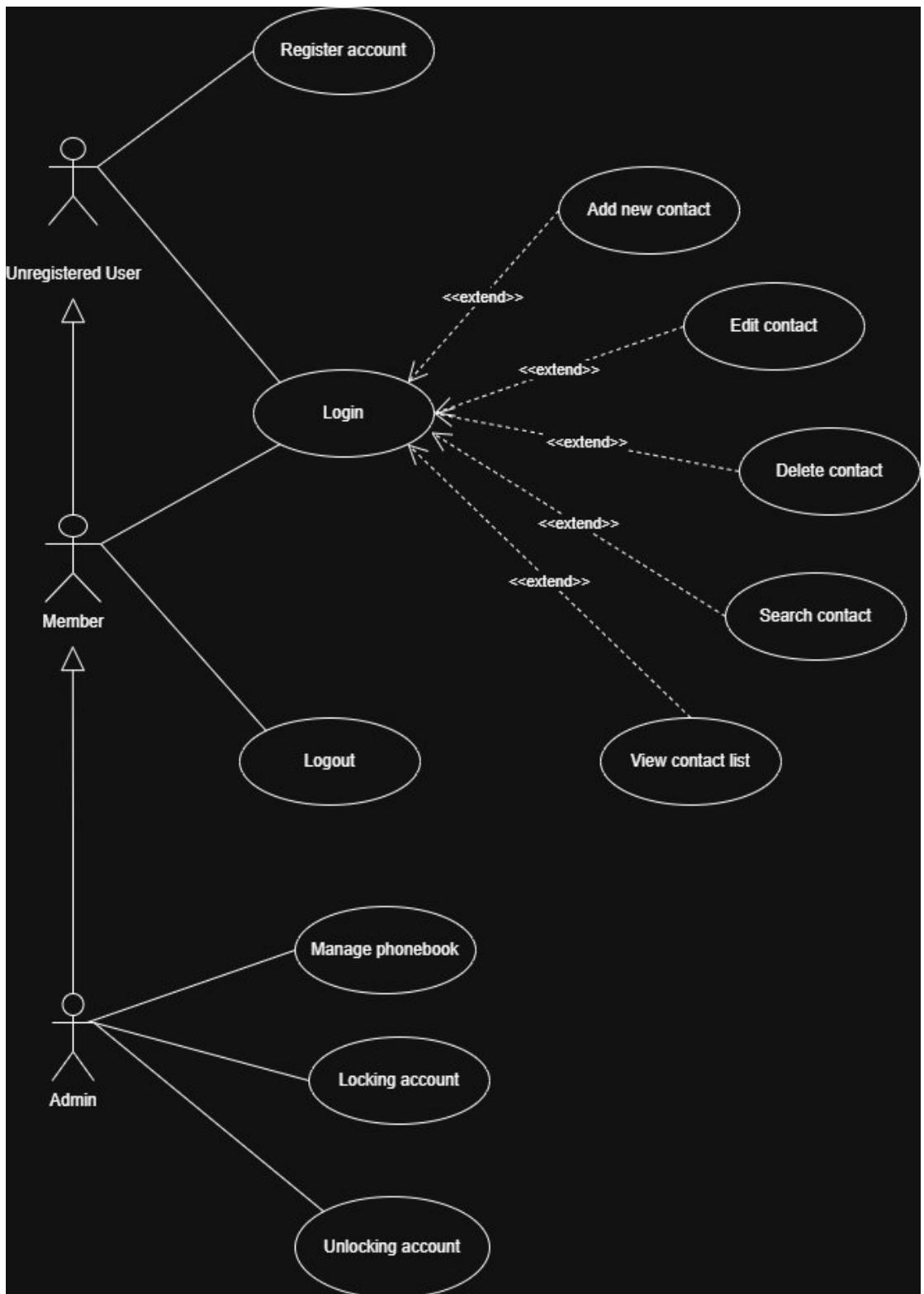
- Works smoothly across all major web browsers and devices (Responsive design).

2.4 Design:

- Programming language: c++

II.USE CASE DIAGRAM

1. Overview diagram:



1.1 List of actors:

STT	Use case	Meaning
1	Unregistered user	guests, users who have not logged into the system or have registered a new account
2	Member	Members, users can use the following functions after logging into the system
3	Admin	Manages all user accounts and contacts.

1.2 list of use-case:

STT	Use case	Meaning
1	Register account	Register a new account
2	Login	Log in to the system
3	Add new contact	Add contact to address book
4	Edit contact	Edit or update contacts
5	Delete contact	Delete one or more contacts
6	Search contact	Find a contact by name/number
7	View contact list	Display all contacts
8	Log out	Log out of the system
9	Manage phonebook	Contact list management
10	Locking account	Open an account for the member
11	Unlocking account	Account overdue lock

2.Flow of event:

2.1 Register account: This use case allows customers to register an account to access the system.

Actor: Unregistered user

2.1.1 Main event line:

1. User selects "**Register account**".

2. System displays the registration form. User enters required details (Name, Email, Password).

3. User submits the information.

4. System validates the data and checks for existing accounts.

5. - **Success:** Account is created, and a "**Registration Successful**" message appears.

- **Failure:** System displays an error message (e.g., "Email already exists").

2.2 Login: Users need to use an account and password to log into the system

Actor: Unregistered user

2.2.1 Main event line:

1. The user selects the "**Login**" function on the interface.

2. The system displays the login form. The user enters their Email and Password.

3. The system validates the provided credentials against the stored data in the database.

4. If valid: The system grants access, switches the account status to "User", and redirects to the contact management dashboard.

5. If invalid: The system denies access and displays an error message: "Incorrect email or password. Please try again."

2.3 Add new contact: The user will add new contacts to their list.

Actor: Member

2.3.1 Main envent line:

1. User selects the "**Add new contact**" option.

2. The system displays a form to enter contact information (Name, Phone number, Email, Address, etc.).

3. User enters the data and clicks the "Save" button.

4. The system validates the information and adds the contact to the list.

5. The system displays a success message: "Contact added successfully!".

2.4 Edit contact: The user can edit or update their contacts

Actor: Member

2.4.1 Main event line:

1. User selects the specific contact they wish to edit from the list.
2. The system displays the current contact details in an editable form.
3. User modifies the necessary information and clicks the "**Save**" button.
4. The system updates the records and displays a success message: "**Update successful!**"

2.5 Delete contact: User can remove their contact.

Actor: User

2.5.1 Main event line:

1. User selects the contact to delete.
2. The system displays a confirmation warning.
3. User confirms the deletion.
4. The system removes the contact from the list.
5. The system updates the records and displays a success message: "**Delete successful!**"

2.6 Search contact: User can find the contact they want to see

Actor: User

2.6.1 Main event line:

1. User enters a search keyword (name, phone number, etc.).
2. The system filters the contact list.
3. Displays the results matching the keyword.

2.7 View contact list: User can see all their contact in the form of a list

Actor: User

2.7.1 Main event line:

1. User selects the "View contact list" option.

2. The system displays the user's entire contact list.
3. User can select to view details of each contact.

2.8 Log out: Member logged out of the system.

Actor:User

2.8.1 Main event line:

1. User selects "Log out."
2. The system confirms the logout.
3. The login session ends → returns to Guest state.

2.9 Manage phonebook: The administrator can manage the contact list.

Actor: Admin

2.9.1 Main event line:

1. Admin logs into the system.
2. Selects the "Manage phonebook" function.
3. The system displays the list of users.

2.10 Locking account: The administrator can open an account for members.

Actor: Admin

2.10.1 Main event line

1. Admin selects a user account
2. Admin chooses Lock Account
3. System updates account status to Locked
4. System displays lock confirmation

2.11 Unlocking account: The administrator can lock accounts that have expired due to inactivity.

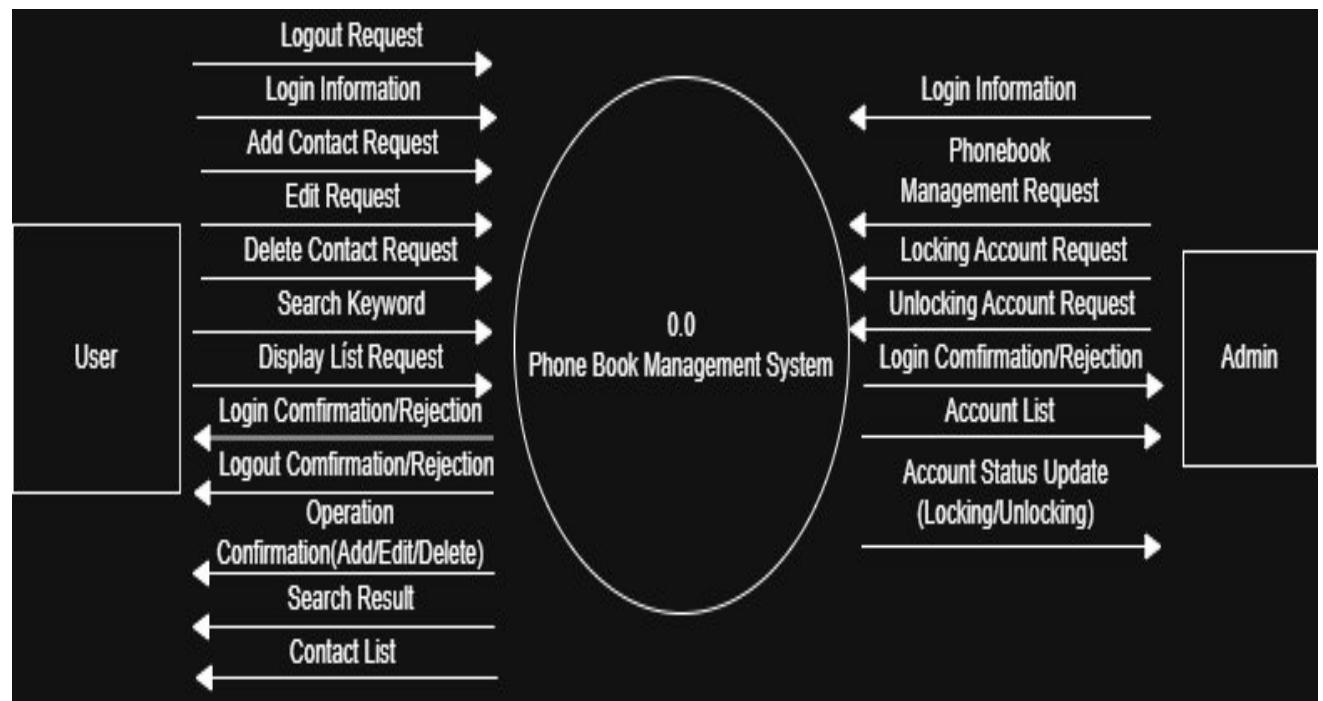
Actor: Admin

2.11 Main event line

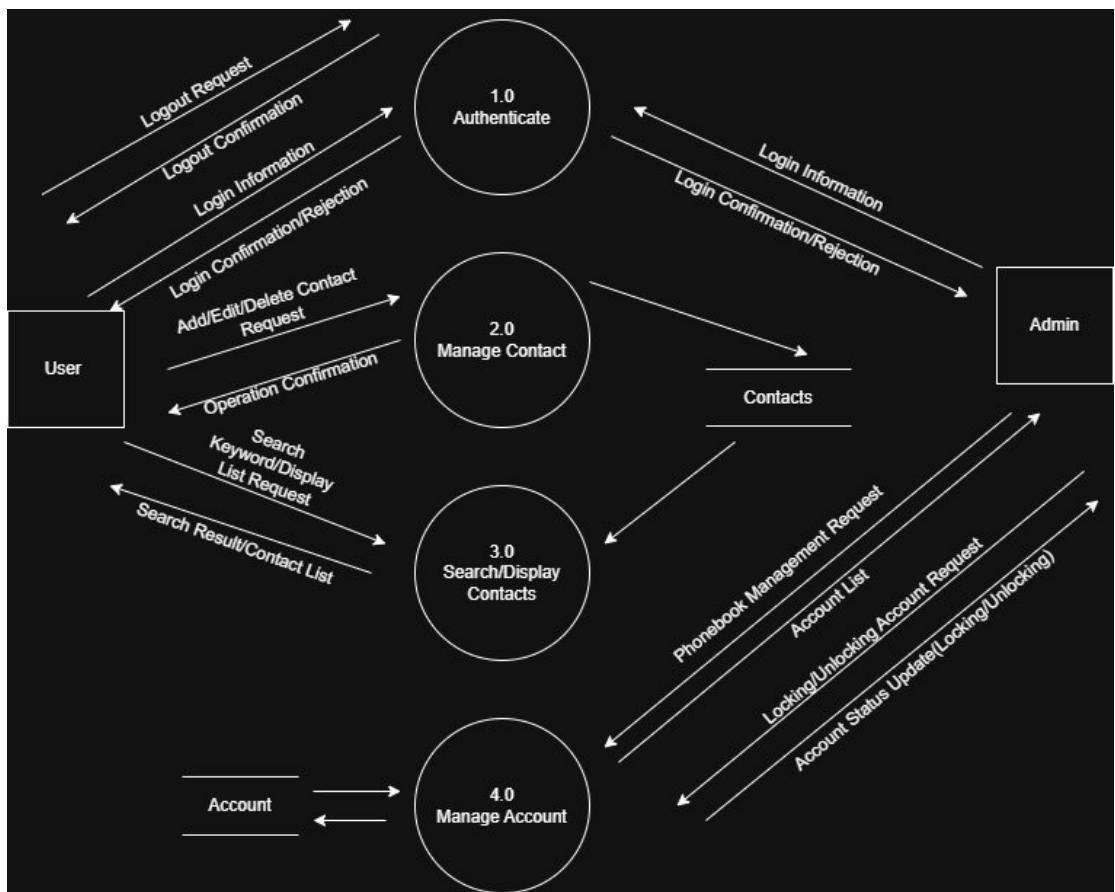
1. Admin selects a locked account
2. Admin chooses Unlock Account
3. System updates account status to Active
4. System displays unlock confirmation

III. DFD DIAGRAM

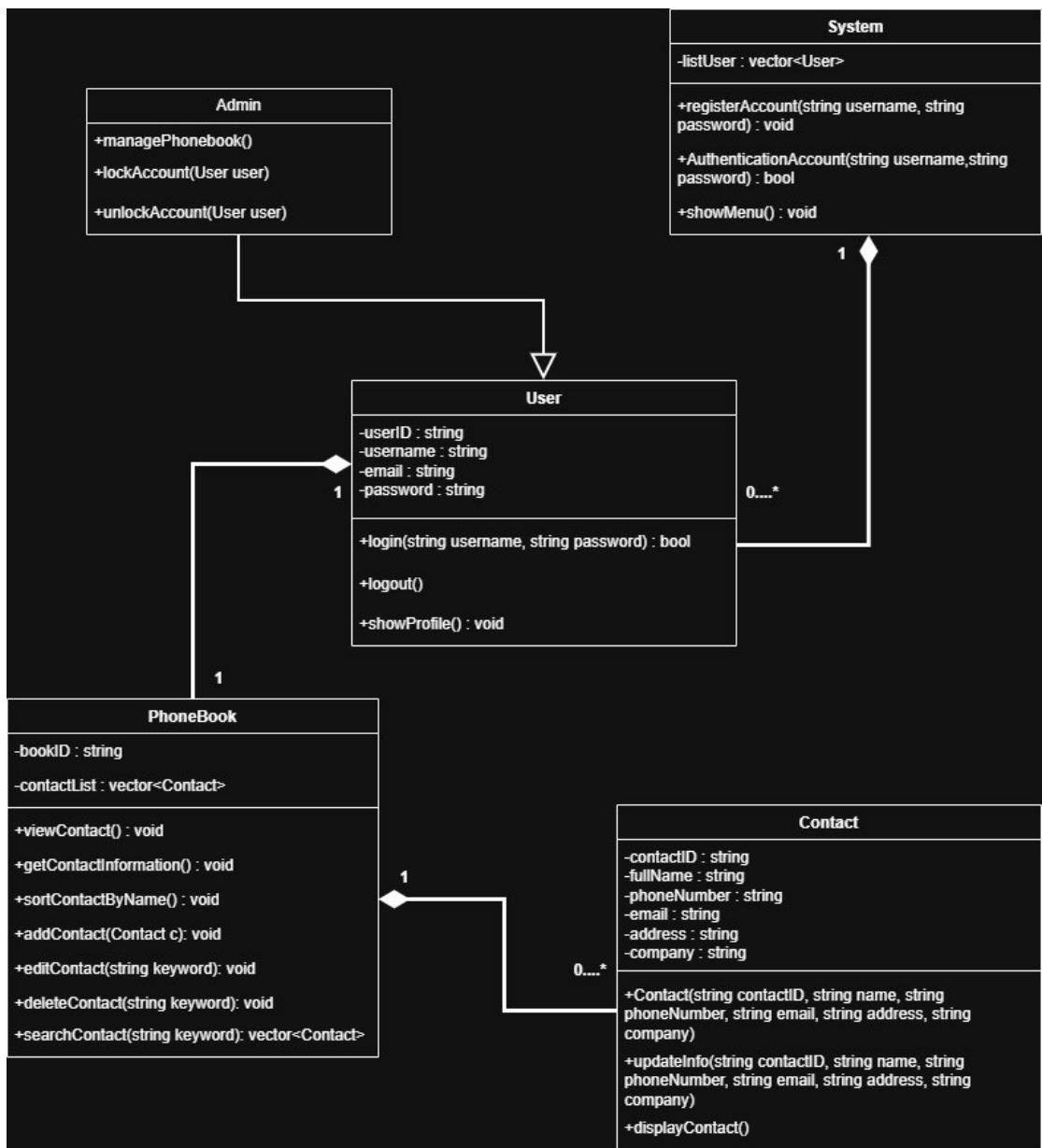
1. IV0



2.IV1



IV. CLASS DIAGRAM



V.DATA MODEL

