

High-Level Design & Low-Level Design

**Index**

1. Introduction ------------------------------------------------ 3

1.1 Intended audience ------------------------------------------------ 3

1.2 Project purpose ------------------------------------------------ 3

1.3 Key project objective ------------------------------------------------ 3

1.4 Project scope ------------------------------------------------ 4

2. Design overview ------------------------------------------------ 4

2.1 Design objective ------------------------------------------------ 6

2.2 Design alternative ------------------------------------------------ 6

2.3 User interface paradigms ------------------------------------------------ 6

2.4 Validations ------------------------------------------------6

-

3. System architecture ------------------------------------------------ 7

3.1 Database architecture ------------------------------------------------ 7

4. Detailed system design -------------------------------------------------9

4.1 Flowchart of main application ------------------------------------------------9

4.2 Flowchart of maintain database() ------------------------------------------------10

4.3 Flowchart of main menu ------------------------------------------------11

4.4 Flowchart of show report() ------------------------------------------------12

5. Tools Report ------------------------------------------------13

5.1 Valgrind ------------------------------------------------13

6. Testing ------------------------------------------------24

6.1 Unit Testing ------------------------------------------------24

6.2 Integration Testing -----------------------------------------------25

7. Requirements Traceability Matrix(RTM) -----------------------------------------------30

**1. Introduction**

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyse and give an in-depth insight into the complete Contacts Management application by defining the problem statement in detail. The detailed requirements of the contacts management application are provided in this document.

**1.1 Intended Audience:**

The target audience set for this project can be identified as an admin who manages multiple users and the users manage multiple contacts and automate the process of managing the contacts which includes personal and business contacts of users.

**1.2 Project Purpose:**

Contacts Management is a project that helps us understand the basic concepts of functions, file handling, and data structure. The automatic contacts management application will get users data as well as the contacts data of each user. It will create a list of personal and business contacts of all users. We can maintain a database of users and contacts by adding, updating, deleting and displaying the main menu of this contacts management application.

**1.3 Key Project Objectives:**

1. Allows the admin to add users
2. Allow the users to add their contacts
3. Updating the contacts information of a user
4. Deleting the contacts information of a user
5. Displaying personal contacts of all users
6. Displaying business contacts of all users

**1.4 Project scope:**

This project aims to create a contacts management application which takes the personal and business contacts information of multiple users such as first name, last name, mobile number, email-id in common for both type of contacts and for personal contacts, additional information like address, WhatsApp mobile number, birthdate, emergency contact and for business contacts, information like company name, designation and company website as inputs and manages the information like updating and deleting the contacts of the respective users and as well as listing the personal and business contacts information separately. The application will have an admin who has the option to add users along with the default passwords which can be changed if required by the users, such that they can login and manage their contacts later.

**2. Design Overview: -**

* **Bowling Game comprises of the following modules in maintain bowler database:**

|  |  |
| --- | --- |
| Name of the Module | Add Module |
| Handled by |  |
| Description | The bowler adds the record in the database |

|  |  |
| --- | --- |
| Name of the Module | Delete Module |
| Handled by |  |
| Description | The bowler deletes a record from database |

|  |  |
| --- | --- |
| Name of the Module | View Module |
| Handled by |  |
| Description | The bowler views the record from database |

|  |  |
| --- | --- |
| Name of the Module | Edit Module |
| Handled by |  |
| Description | The bowler edits the record in the database |

* **Bowling Game comprises of the following modules in play the game:**

|  |  |
| --- | --- |
| Name of the Module | Random Number module |
| Handled by |  |
| Description | Generating the random numbers |

|  |  |
| --- | --- |
| Name of the Module | Strike Condition Module |
| Handled by |  |
| Description | Knocking down all the pins with the first ball ends the frame |

|  |  |
| --- | --- |
| Name of the Module | Spare Condition Module |
| Handled by |  |
| Description | Knocking down all the pins with the first two balls |

|  |  |
| --- | --- |
| Name of the Module | Extra balls Module |
| Handled by |  |
| Description | A spare in the tenth frame earns one extra ball |

|  |  |
| --- | --- |
| Name of the Module | Play the game Module |
| Handled by |  |
| Description | Bowler will play the game and awarded with scores |

* **Bowling Game comprises of the following modules in show reports:**

|  |  |
| --- | --- |
| Name of the Module | Bowler datasheet Module |
| Handled by |  |
| Description | It will show the bowler datasheet from bowler database |

|  |  |
| --- | --- |
| Name of the Module | Bowling day report Module |
| Handled by |  |
| Description | It will display the bowling day report from bowling database |

## Design Objectives:

1. Add users to user database
2. Add contacts information to contacts database
3. Update the contacts information of a user
4. Delete the contacts information of a user
5. Display personal contacts of all users
6. Display business contacts of all users
   1. **Design Alternative:**

We have used standard template library (STL) to store contacts information i.e., contact\_id, first name, last name, mobile number, email\_id, address, WhatsApp mobile number, birth date, emergency contact, company name, designation and company website.

### 2.3 User Interface Paradigms:

The Contacts Management application provides an option to users to add, update, delete and display their personal and business contacts.

### Validation:

* Contact\_id should not be blank and duplication is not allowed and characters aren't allowed in the contact\_id
* We check for the validity of the contacts information; it should not contain more than the required fields

**3. SYSTEM ARCHITECTURE:**

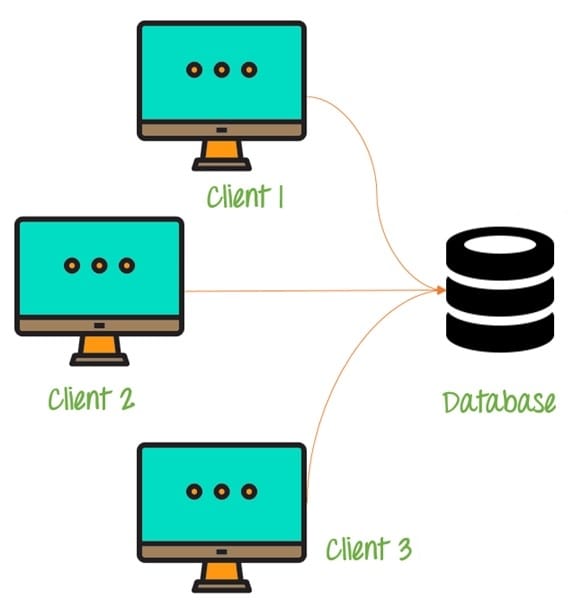
**3.1. Database Architecture**

The architecture used in this system comprises of the database architecture. It is a representation of the database management system design, wherein you can design, develop, implement and maintain the database. This architecture allows dividing the database into different components that can be independently modified, changed, replaced and altered as required for the system.

The database architecture is divided into three tiers namely,

* 1 - Tier Architecture
* 2 - Tier Architecture
* 3 - Tier Architecture

Our system is based on the Tier 1 model of the database architecture. In this type of model, the database is directly available to the user, the user can directly access the database and all of its contents. Which enables the user to directly interact and execute operations?



Some of the characteristics of Database Architecture are:

Self-Describing Nature of a Database System:

* One of the most fundamental characteristics of the database approach is that the database system contains not only the database itself but also an entire definition or description of the database structure and constraints also known as metadata of the database.

Isolation between Data, Programs and Data Abstraction:

* In a traditional file processing system, the structure of database knowledge files is embedded within the application programs, so any changes to the structure of a file may require changing all programs that access that file.

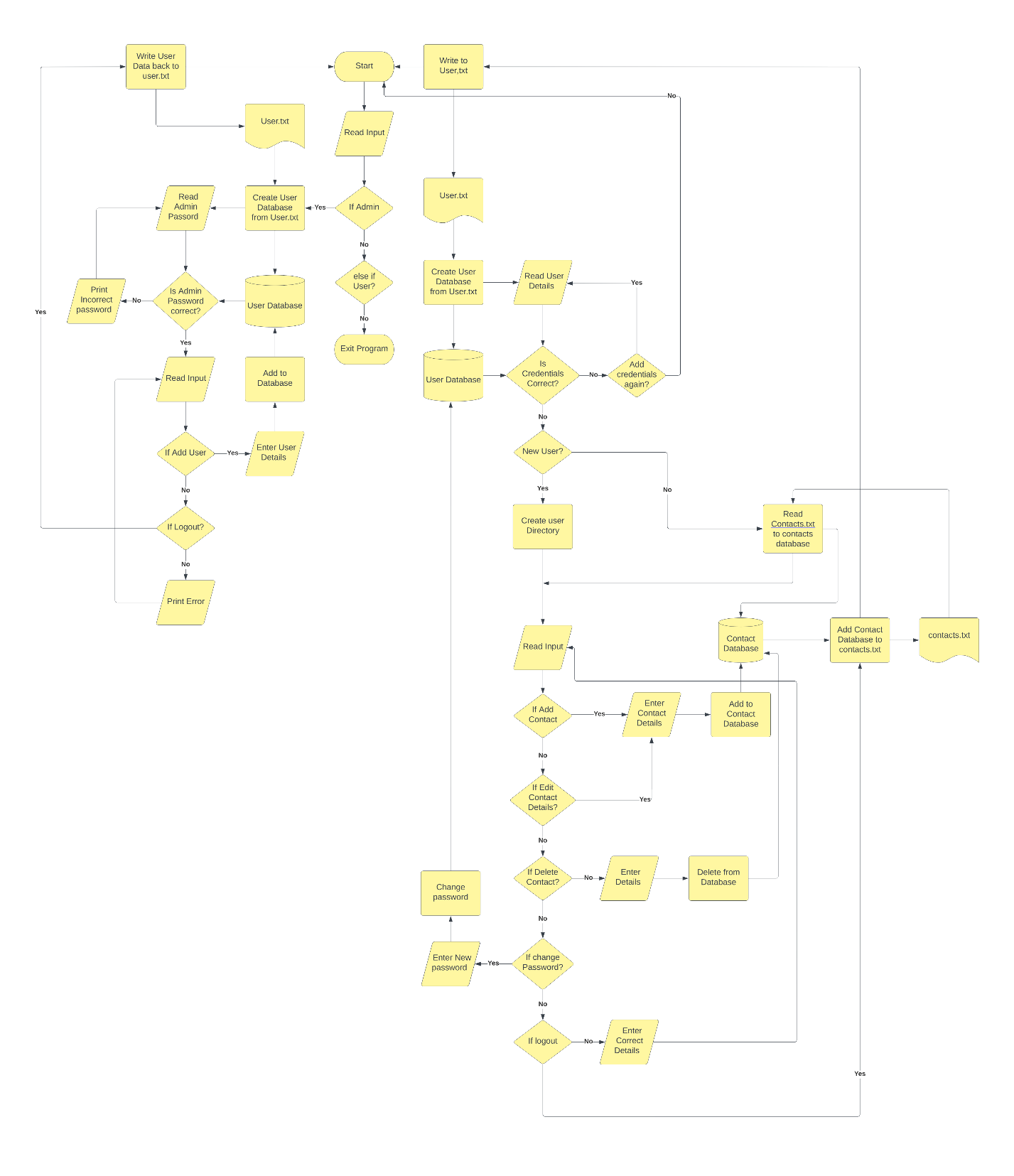
Support for Multiple Views of the Data:

* A database sometimes has many users, each of whom may require a special perspective or view of the database.

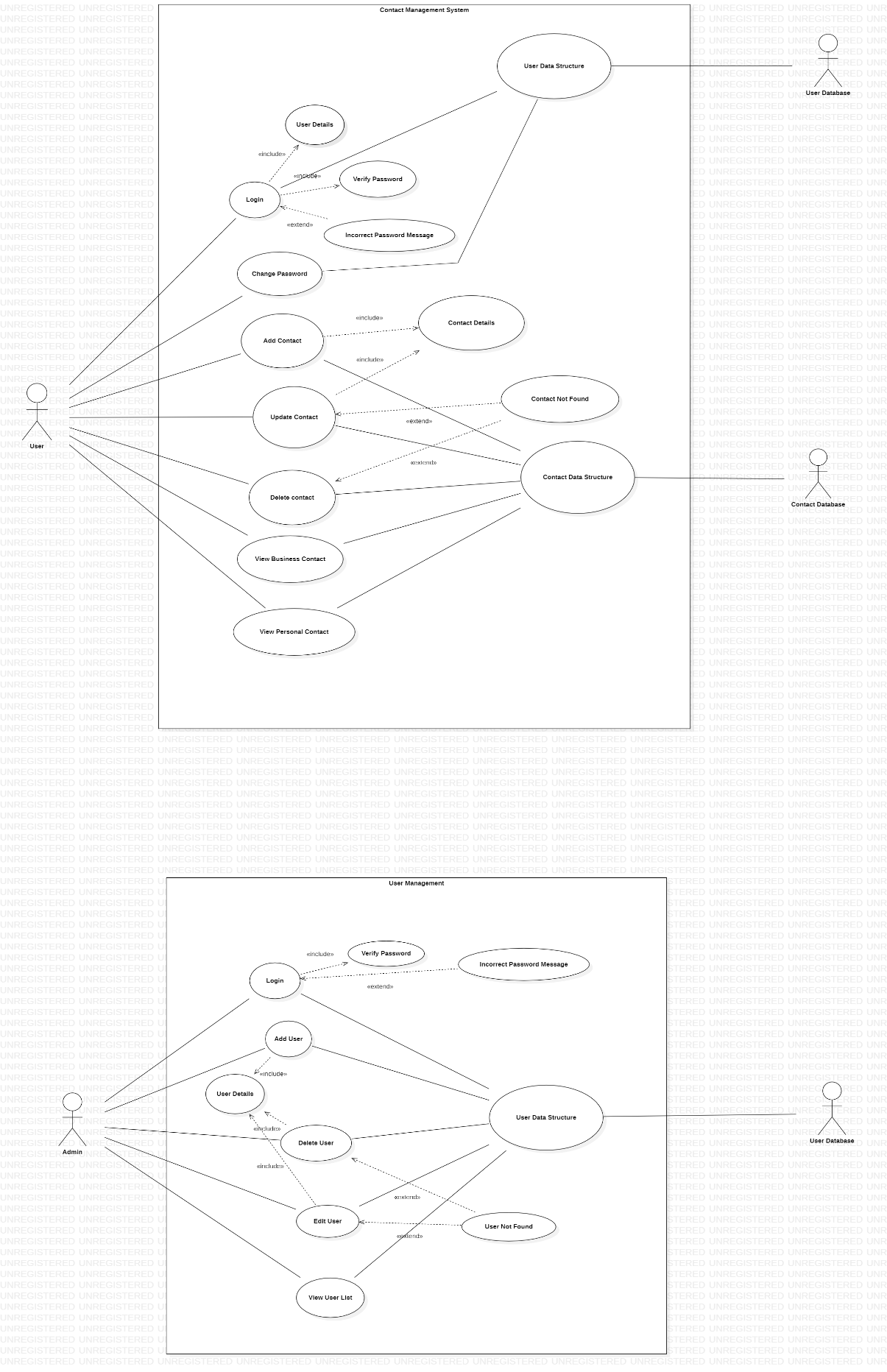
Sharing of knowledge and Multi-user Transaction Processing:

* A multi-user DBMS, as its name implies, must allow multiple users to access the database at an equivalent time or concurrently.

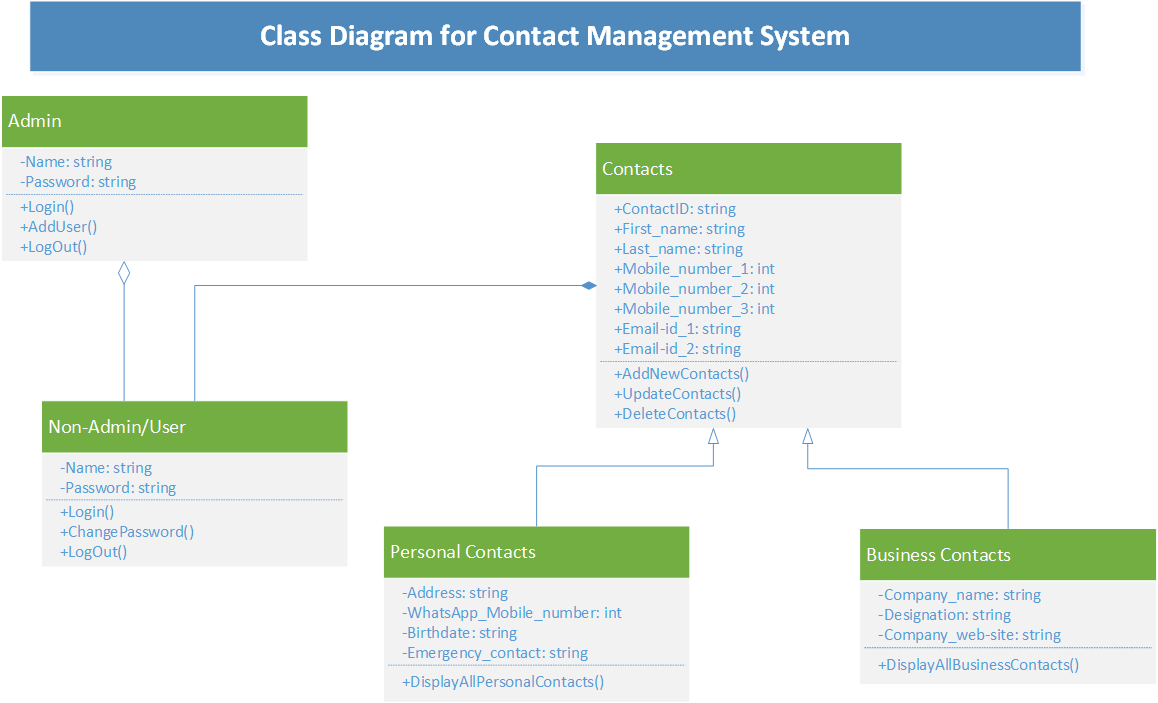
**4. DETAILED SYSTEM DESIGN:**



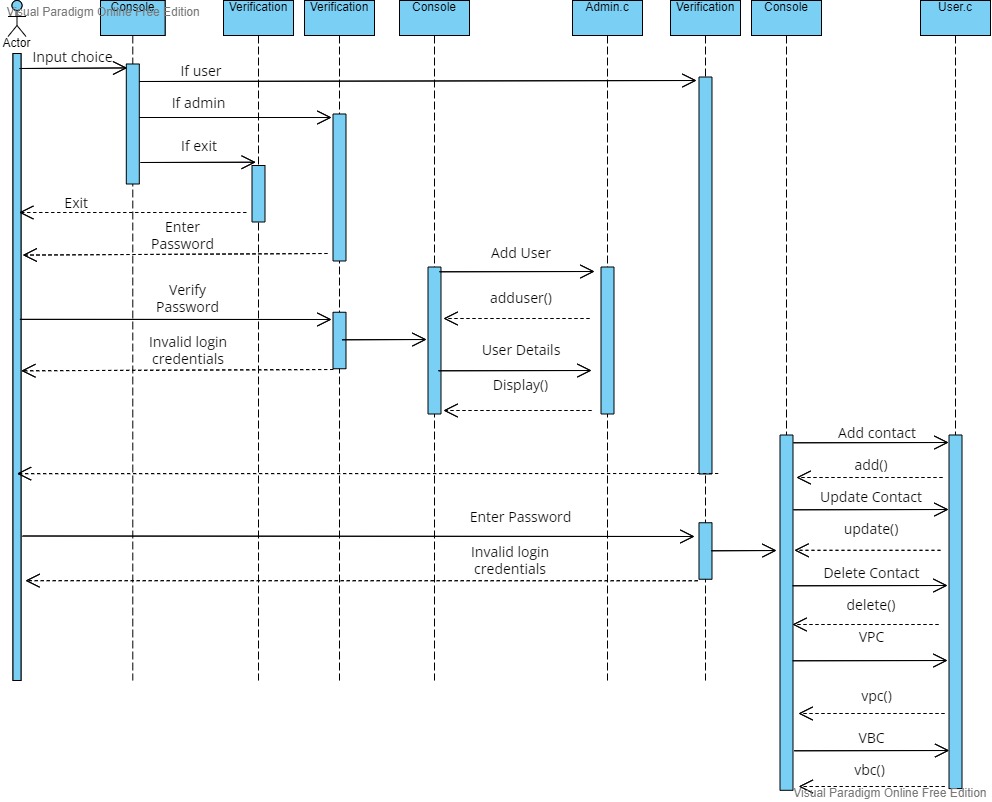
**4.1 Flow Chart of the application**



**4.2 usecase Diagram**



**4.3 Class diagram**



**4.4 Sequence Diagram**

**5. TOOLS REPORT**

**5.1 Valgrind**

**6. Testing**

**6.1 Unit Testing**

**6.2 Integration Testing**

**6.2.1 Add**

**6.2.2 Delete**

**6.2.3 Edit**

**6.2.4 View**

**6.2.5 Play the game**

**6.2.6 Bowler data sheet**

**6.2.7 Bowler day report**

**7. Requirements Traceability Matrix(RTM)**