Software Requirements Specification

For

NOTES TAKING APP

(NOTER)

Version 1.0

Prepared by: Taha Mustafa(20CAB002)
Suhel Khan(20CAB005)

Aligarh Muslim University

24th October 2022

24/10/2022

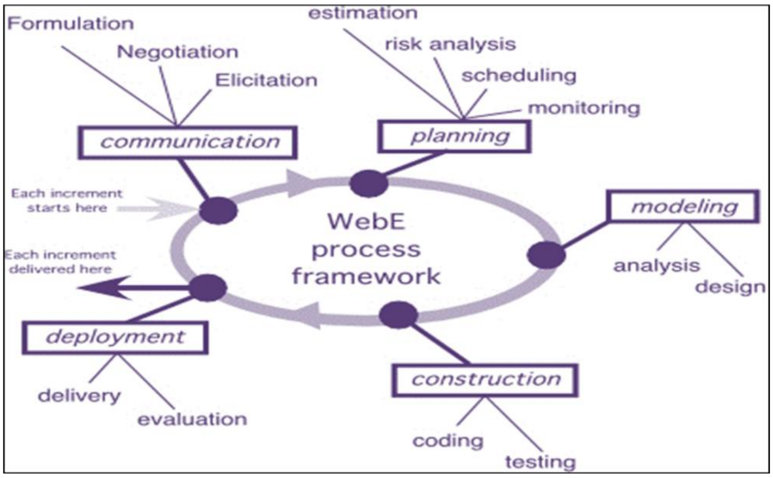
Table of Contents

1.0 Introduction

NOTER(Notes Taking APP) is a project which aims at developing an online application to promote Online Education, maintain Study Materials. Notes Making is a Web Based Application which would assist students in carrying their notes anywhere anytime. This project has login features, Student as an user can login into their own profile separately and create their notes. The Admin can login, through which the admin can monitor the whole system. The Admin after logging into his account can remove users, add users and much more. Students can login into their account to retrieve their notes and can also view other students public notes as well.

Overall this project of ours is being developed to help the students as well as other User (teachers/learners) to provide notes making and maintaining platform in the best way possible.

For building this project we will be using Agile Methodology. It is one of the most famous software development methodologies which came to prominence in the early 2000s. The agile process consists of five main activities which are as follows: - Communication, Planning, Modeling, Construction and Deployment. For writing this SRS we will be focusing on the first three activities only as those activities together make the SRS. The Communication activity deals with interaction and collaboration to find the objective and requirement needed for the creation of the project. So this part will comprise of project scope, the purpose and so on in the SRS. The Planning activity will deal with requirements which included both the functional and non-functional requirement. for the project and the most important the increment model. The Modeling activity will deal with creation of models (design models) which help the user in understanding the WebApp requirements better.



1.1 Purpose

Inspite of huge influence of technology in our daily lfe, majority of the students still create their notes manually by pen and paper,while this may work at a lower level however at a higher level managing these notes becomes a tedious task , also there is a risk of these paper notes being destroyed or lost.

To resolve this problem we have decided to construct a web based application which would help students to create and manage their notes with ease.Yes the idea of web based notes app has already been implemented and the app has already been implemented and the apps are even offering good service to it's users as such our main aim behind the construction of this app is to further increase the convenience of students by manyfold by using these apps as our base and then building upon them.

1.2 Document convention

To prepare this SRS document, we will be using Font style Times New Roman for paragraph content throughout. We will use font size 12 for writing content information and 14 for heading and subheading. We will also be using bold as well as underline style. The alignment used would be justify.

1.3 Intended audience

This document is intended for students, learners and project enthusiasts who wish to build the application or have keen aspiration in the field of web based application development. It will help the user or students using the application to understand the working behind it.

The following document is divided into five parts and three Appendixes.

We begin by **Introduction**. It would introduce the reader to the project and make them understand the purpose as well as the scope of the project. This part would also contain the references for the project. Next we have **Overall Description** that will give the reader a detailed description of the project. It would also inform about the operating environment and constraint associated with it as well as the assumption and dependencies related with it. We have **External Interface Requirements** in the third part. This part would throw light on the basic interfaces associated with the project which would include user, hardware and software interfaces. Next part is the **System Features.** This part would throw light on a detailed overview of the features that this system would be able to perform. Last but not least we have the **Non-Functional Requirement** part. This part would give information related to the performance, safety, security, and functionality about the system.

Talking about the **Appendix** it is divided into three parts. The first part **Appendix A** Would inform the reader about the glossary of the document. The second part **Appendix B** contains information regarding the Analysis model which includes Data Flow Diagrams, Entity Relationship Diagram(ER-diagram), Class diagram. It would also contain the model used to develop this project which is the **Agile Methodology**. The third part **Appendix C** would contain the "To be Determined List"

1.4 Product scope

This product will provide the users specially students the ability to manage and organize their notes in an efficient manner such that they could access them anytime they want. This application would substitute the traditional way of notes making .The "public notes" feature would also enable the students to share and view each other's note and thereby enhancing their productivity.

1.5 References

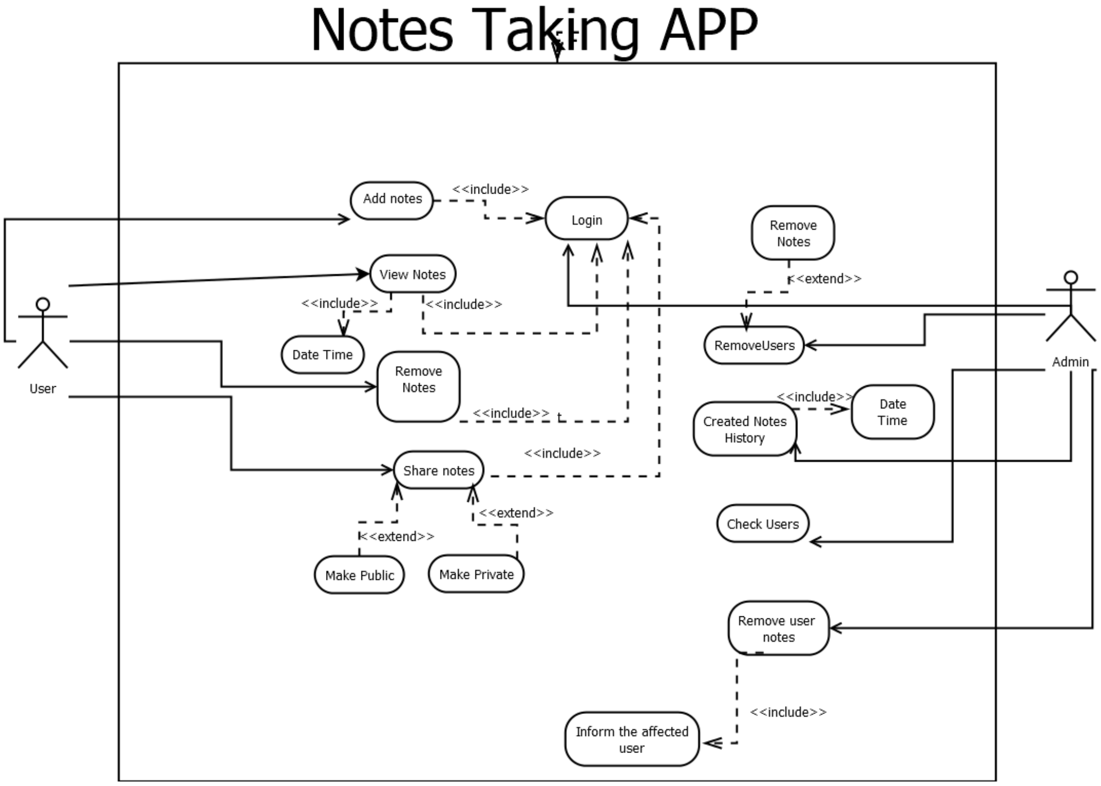
- Git Hub and various other internet resources.
- IEEE SRS template.
- Many books on web engineering.
- Web apps like Notion Evernote and Simple Note

2.0 Overall Description

2.1 Product perspective

"Noter"- A notes taking web based application is a standalone system that provides the functionality described in the Product Features section. It contains all the subsystems necessary to users for notes making, such as adding new notes , making notes public or private depending on user's preference and most importantly to manage notes history in the most efficient way. This Web-App will store all its data in an efficient manner. The following are the main features that will be included in the Notes taking web app(**Noter**):

- Cross Platform Support: It will be offer operating support for most of the known and commercial operating systems.
- User Account: This system will allow the creation of user accounts.
- Multiple Users Support: Notes making Web App will allow the creation of multiple users and support them.
- Search: Search is simply a search facility to search for various user's notes on the Notes making App.
- Access Levels: The web app would allow the users to either completely hide their notes from the external world or make them available in public.



Use Case Diagram for Noter(Web App)

2.2 Product function

The notes making Web App:

- Provide a online place to create and save notes easily .
- Provide the users to view other user's notes
- It also help user's to make modify the access levels of their notes

- It provides a user-friendly interference for the user.
- Help set up time constraints.
- It stores all data in a centralized storage system.
- Inform the user's about the latest development pushed in the system

2.3 Users classes and characteristics

This system is aimed at users who are related to academics or the user who maintain notes. They can be categorized into two profiles, each with its own responsibilities and roles.

User	Roles and Responsibilities
Student	It is the **user** who is responsible for creating and managing his notes. They are the main building block of the system.
Administrator	Responsible for controlling the system and overall development flow. They act as the supervisor for the students and are also the **admin** for the system.

Admin Roles

- View Notes History
- View Users
- Remove users
- Remove user's notes

Users Roles

- Create notes
- Delete notes
- View notes
- Publish/share notes as public or private

The user or the student has to first register himself in order to access the system. He can add ,remove, public/private his notes. He can also view the notes which are in public ledge.Each user would have a unique UserName and a password through which they could access their accounts. Each note created would have a unique id called NoteId and similarly each public note would have a PublicId.

The admin would be provided with his/her login credentials. The admin can view different users who are registered on the system and would have the ability to remove them from the system. The admin can also view notes history of all the users like the date when it was created ,the date when it was modified. The admin can only view the name of the note and the user who created it , he neither can view the content of the note nor change it. The admin would have the facility to remove a user’s note and in this case a message would be sent to the affected user.

2.4 Operating Environment

The server-side components of this system would work in a Windows 10 or newer operating system environment using XAMPP.
The user-side components of the system would be accessible through a common web browser

- Distributed database using MySQL.
- User/Server System with XAMPP.
- Operating System: Windows 10 or above.
- Frontend: HTML , CSS(Plain as well as a Framework) , JavaScript
- Backend: PHP

2.5 Design Implementations and Constraints

This document represents a study project, not an actual SRS, and lacks detailed descriptions and requirements in many areas. There are only instructions and requirements templates for creating a notes making web based application.

- The information of all users and notes must be stored in the database that is accessible by the website.
- XAMPP sever will be used as SQL engine and database.
- The system would run 24 hours a day
- User should provide correct credentials to register themselves

- The Username and password should be correct to login to the account and perform actions.

2.6 User Documentation

For setting up the server side environment a quick google search “Running Xampp on your System” would provide the user with a algorithmic approach on how to setup the environment in your local machine. As such use-manual is not needed to use the system ,the systems home page(How to use us?) itself has instructions on how to use the system.

2.7 Assumptions and dependencies

These are the following Assumptions and dependencies -

- This application will require a third-party product such as XAMPP or WAMP to run. Here we will be using XAMPP to run it.
- We will be using MySQL for storing data in database. The notes making web app will store all the operations it performs (includes management of user and notes) in the centralized data storage. We assume that the system shall be able to access and store data in any Data Base management System (DBMS).
- PHP will be used to develop this project. The notes making app will support efficient notes making. Hence notes making web app shall run on various platforms and would be able to communicate with its subsystem via internet.

3.0 External Interface Requirements

3.1 User Interfaces

User can interact with the system through Graphical User Interface(GUI). The design of the website is simple as such users can automatically surf the website without any help needed. The interface is self explanatory. Flow of the different activities is kept relativity simple .By just a mouse click a user can perform different functionalities.

3.2 Hardware Interfaces

The hardware requirements on the user side are very simple and once deployed; this website can be run on any website or used on any device by simply opening a web browser. System performance is inversely proportional to the hardware available to the user.

3.2 Software Interfaces

The notes making web based application can run on any version of Windows 10 or above. The Notes making WebApp features an easy and simple to use interface made with HTML and CSS which makes it convenient to use. The application contains an-inbuilt database for storing projects and tasks made with the help of MySQL. The database can be updated by adding new users, notes and tasks which are synchronized to the database.

3.4 Communications Interfaces :-

Users can connect or login to XAMPP from any device. You can also create an account if the user does not have one yet. Once logged into the system, users can easily start making notes and delete it, as well as share notes. Admins can also log in to manage users and the notes created by them.

4.0 System features

4.1 Account creation and log-in

4.1.1 Description and Priority

In this module the user will be able to create an account to get access to the system. This comes **high** in priority as without the user account a user would not be able to access the system.

4.1.2 Response Sequence

When the user first runs the application he/she will be taken to the how to use us page where they would be instructed on how to use the system. They then would have to login in order to access the site. In case they do not have an account they would have to first register themselves.

4.1.3 Functional Requirements

1. There are two users – admin and student.
2. Admin account is system created and he has the complete access to the application.
3. The user must create an account to get access to the application. Also the user does not have complete access to the system itself.

4.2 Notes Making

4.2.1 Description and Priority

In this module the user will be able to create his notes. This is also a **high** priority module as without it no notes can be created.

4.2.2 Response and Sequence

When the user is logged in he will be given an option to create a notes. With the click of that option the user will be redirected to a page where he can create new notes and make changes as required Once the notes is created the user will be able to save it as public/private.

4.2.3 Functional Requirements

1. Both the user and admin can perform functions but there are certain limitations to both of them.
2. The user can create notes, delete notes, and can make any desired changes available .
3. The admin can see all the users
4. The admin can also remove and delete a notes or users as she/he see fit.
5. The admin can only view Note Name along with the user’s name who created it ,he can not view the content of the notes.

5.0 Other Non-Functional Requirements

There are various non-functional requirements that apply to every web based application, including security, performance, maintenance, dependability, and so on. These are fundamental components of the overall notes making web app since they go beyond a single or small set of functions and are more necessary. These non-functional requirements can also be met by “Noter”.

5.1 Performance

The system can't slow down because the users don't have downtime to wait for it to finish an action. Every time a process is requested by the students, the system must successfully update the databases. The system's calculations must follow the guidelines specified by the administrator and shouldn't change until they are specifically altered by them.

5.2 Security

At every stage of a project's operation, “Noter” is in charge of managing enormous amounts of data generation, information sharing, storage, and analysis. It delivers strong, multi-layered security to all data transfers due to its foundation in the cloud and other cutting-edge digital technologies, safeguarding the system against information loss or misuse. “Noter” is protected from hacks because it is hosted on cloud-based servers that are situated outside of the premises. Overall, “Noter” encourages openness, safeguards privacy, protects against data theft, and provides a safe and secure environment for academic processes related to Projects.

5.3 Usability

Both administrators and Students must be able to use the system without having to read numerous instructions, therefore it must be simple to use. Both administrators and students must be able to rapidly access the system. The system must present all relevant data, information, or instructions in an understandable and straightforward manner. Users must be able to simply navigate the system's menus and understand the buttons.

5.4 Reliability

The system must continuously provide users with accurate student performance statuses. Regular confirmation of the actual levels shown in the system. Any desired action needed must be taken by admin, such as -create note history, delete users, delete notes,and many more. To prevent any unidentified entity from altering the system's data, the system must allow a password-enabled login for the user. The system must update . The system should not update the data in any database for any incomplete or unfair work. Being software as a service, Noter is highly resilient to any technology disruptions, downtime, or crashes experienced by other technology systems. It is highly secure from a data safety point of view.

5.5 Maintainability:-

Good notes making technologies are simple to keep up with. It can be upgraded quickly, and its maintenance tasks may be divided up and planned when demand is the lowest.

5.6 Supportability

The software is made to function even on systems with the most basic configuration. Even if more plugins or modules are introduced in the future, the system is flexible. To make the system more portable, the data can be exported to the administrator.

5.7 Packaging

The software must be compatible with all versions of Windows, starting with Windows 10, as well as with coming updates like Windows 11. An authentication method using the license key must be included in the software. A user's guide explaining how to use the system and the program must be included in the packing. This handbook could be provided with the software as either a booklet or a digital file.

5.8 Implementations

Microsoft Visual Studio is used in the creation of the System User Interface. Microsoft Visual Studio is used for programming. The XAMPP MySQL connection is provided to Visual Studio and is used to establish a connection between the Database and the System.

5.9 Interfacing

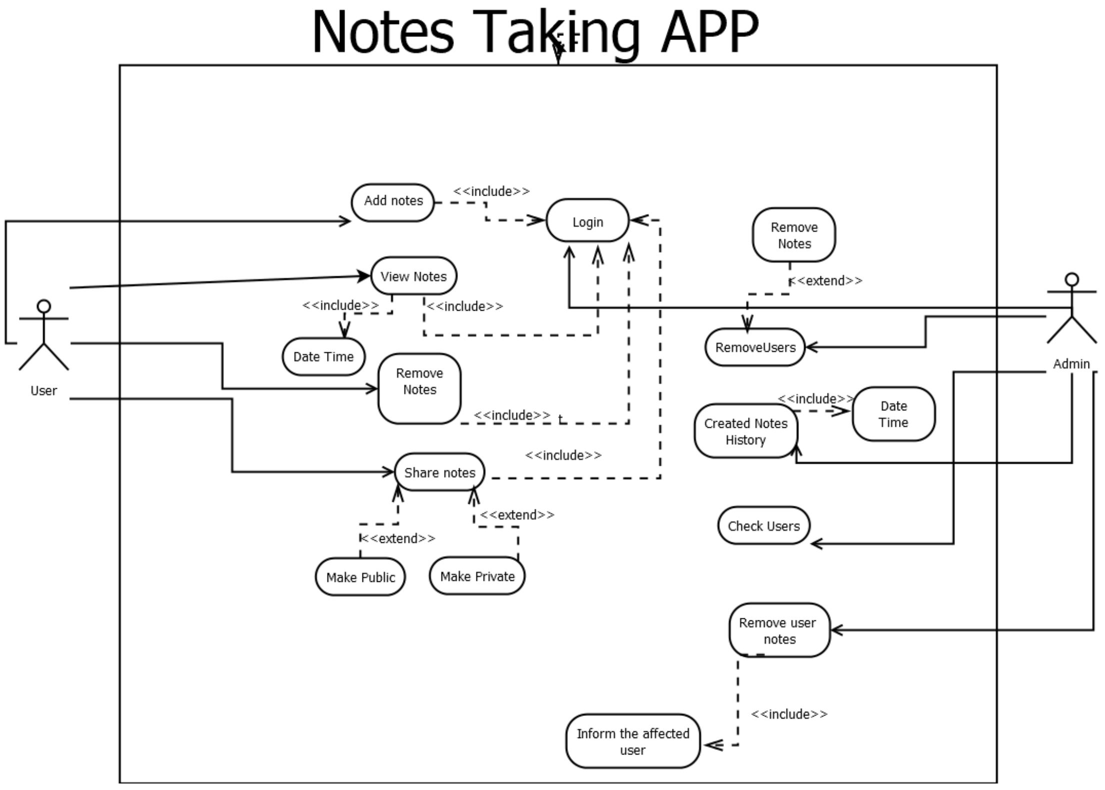
The system must provide a simple method for viewing each student's or learner's current performance. The system must be able to transparently illustrate the connections between students,notes and admin.

Appendix A:Glossary

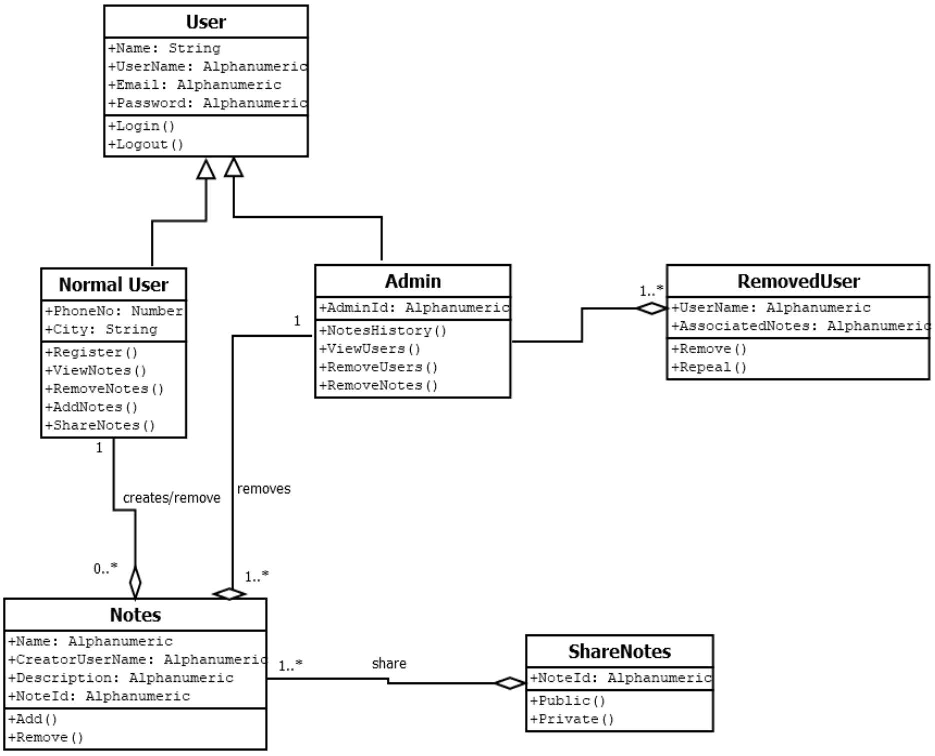
Terms	Meaning
Noter	Name of notes making WebApp or website
SRS	System Requirement Specification
Admin	Person responsible for the entire system. Also the Project Coordinator
Student	Person working on the system.
User	Term containing both Admin and student.

Appendix B:Analysis Model

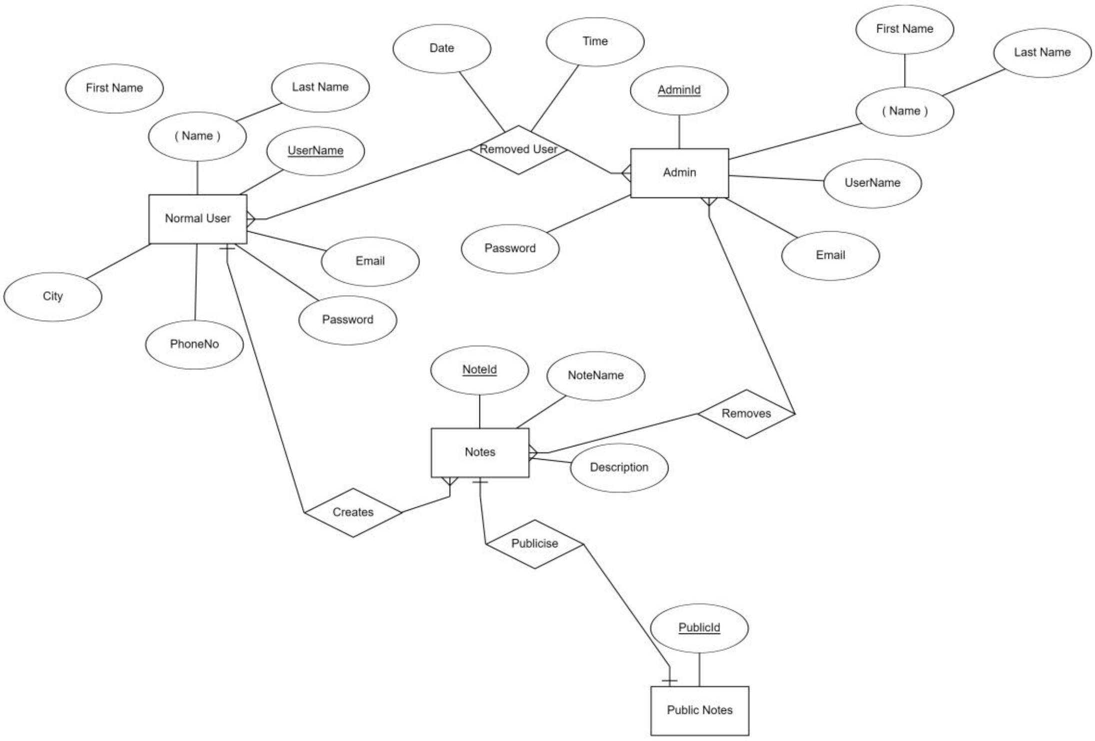
1.Use Case Diagram



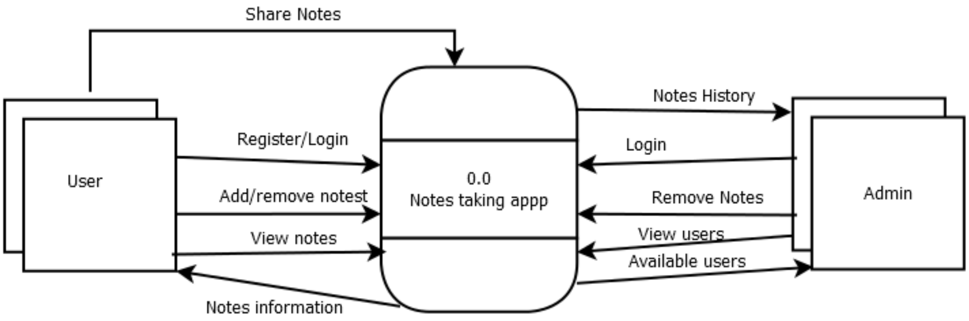
2.UML Class diagram



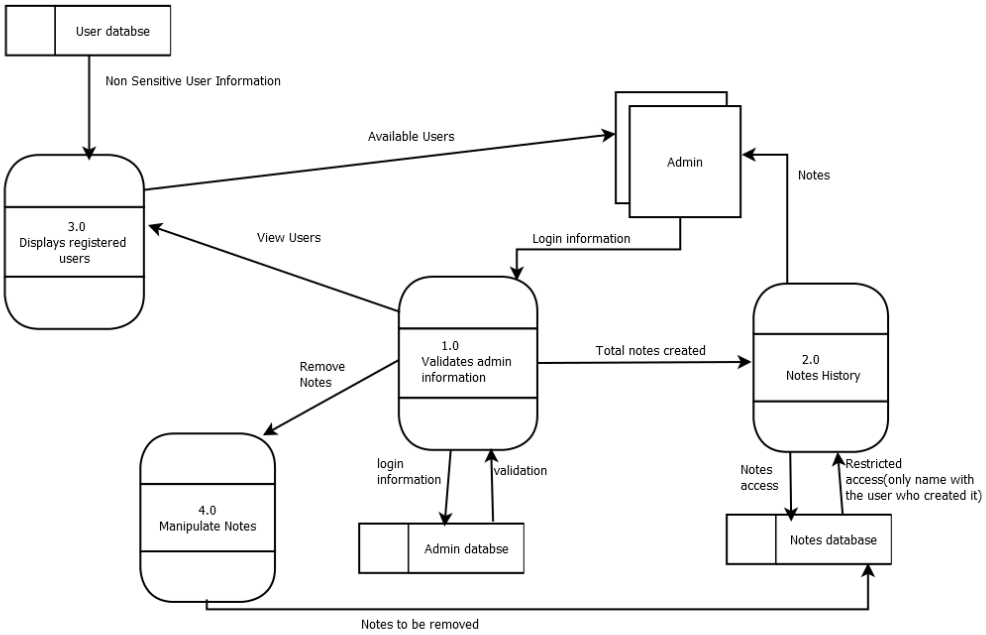
Entity Relationship Diagram



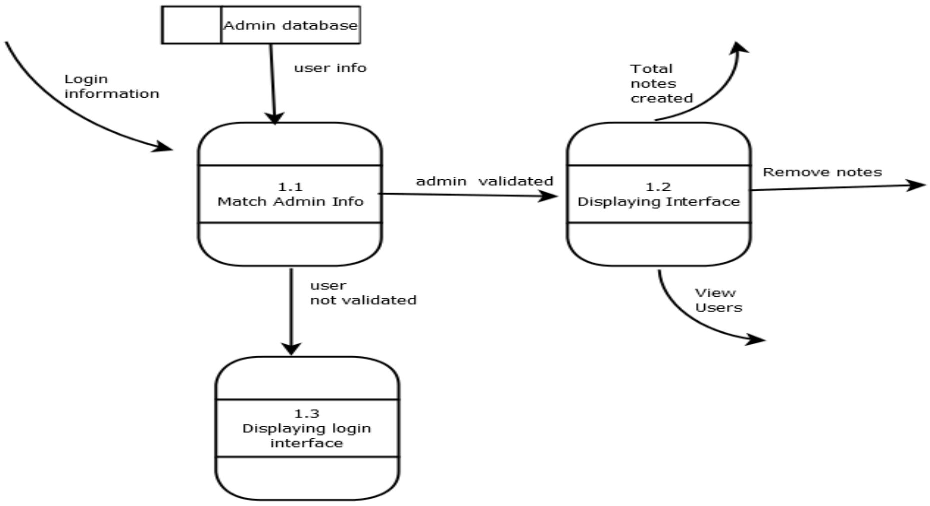
5.Context level diagram



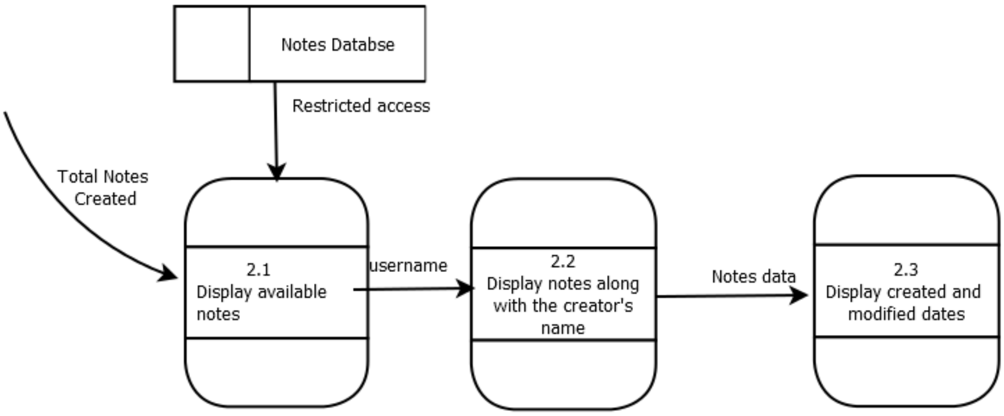
6.Admin logical data flow diagrams



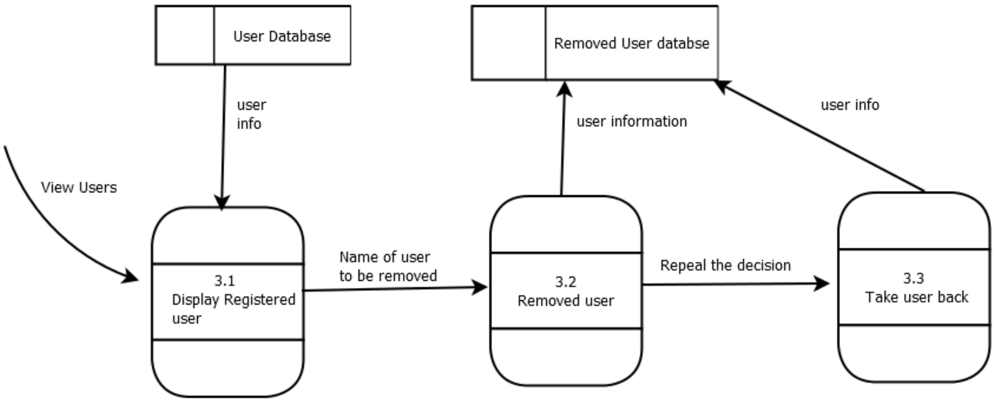
Level-1



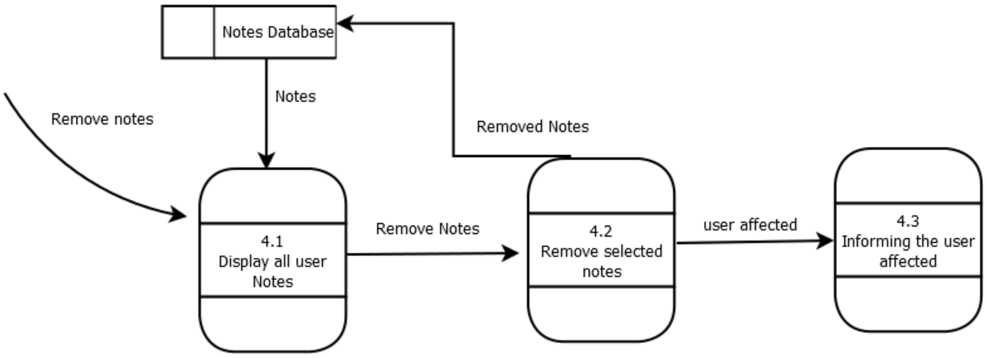
Level-2 Process-1



Level-2 Process-2

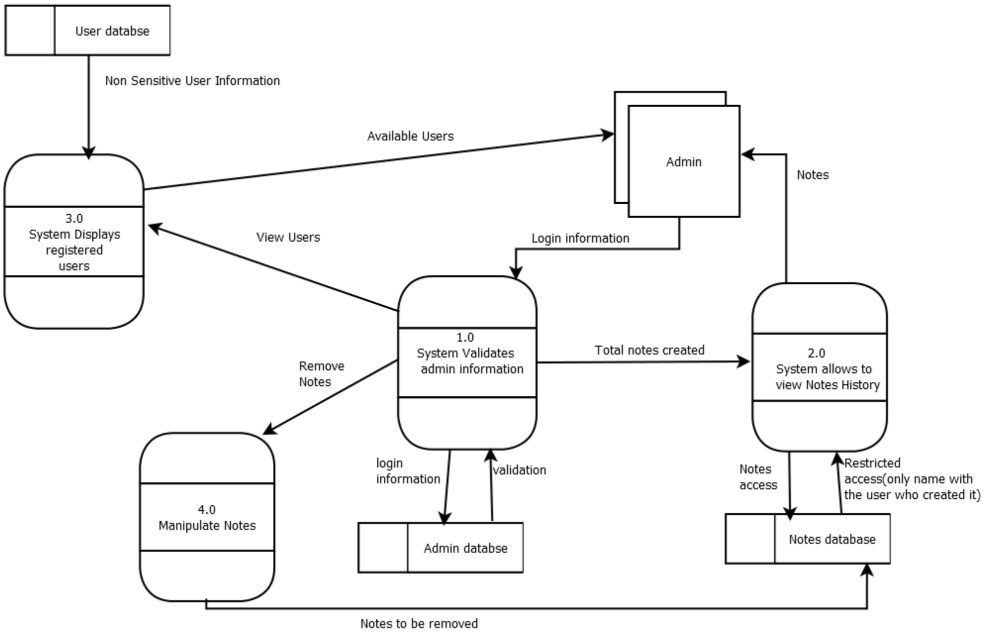


Level-2 Process-3

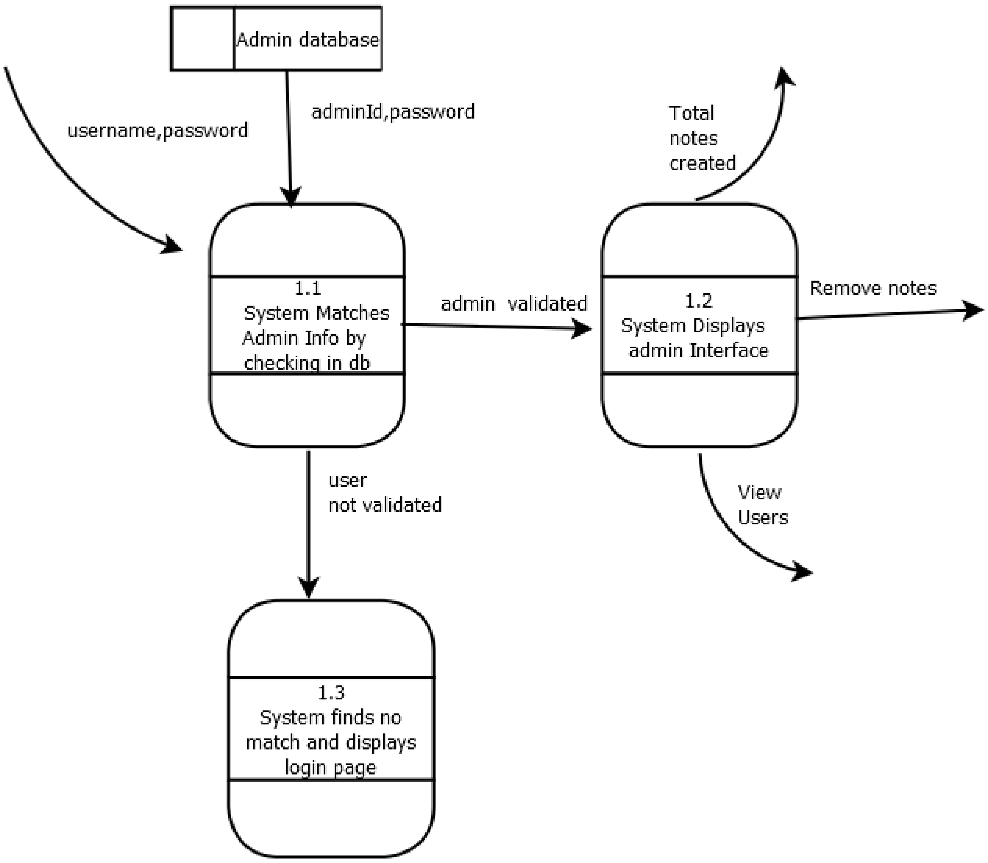


Level-2 Process-4

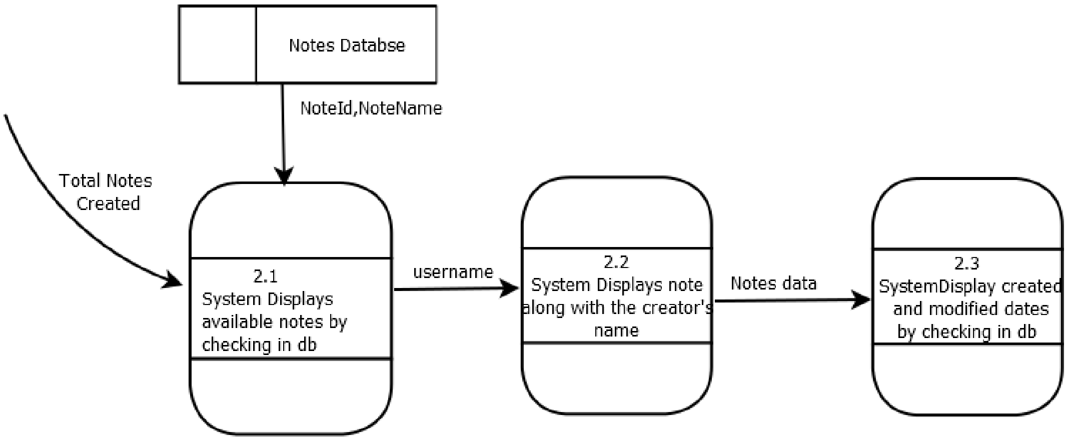
7.Admin Physical Data Flow Diagram


User databse
Non Sensitive User Information
Available Users
Admin
Notes
3.0 System Displays registered users
View Users
Login information
1.0 System Validates admin information
Total notes created
2.0 System allows to view Notes History
Remove Notes
login information
validation
Notes access
Restricted access(only name with the user who created it)
4.0 Manipulate Notes
Admin databse
Notes database
Notes to be removed

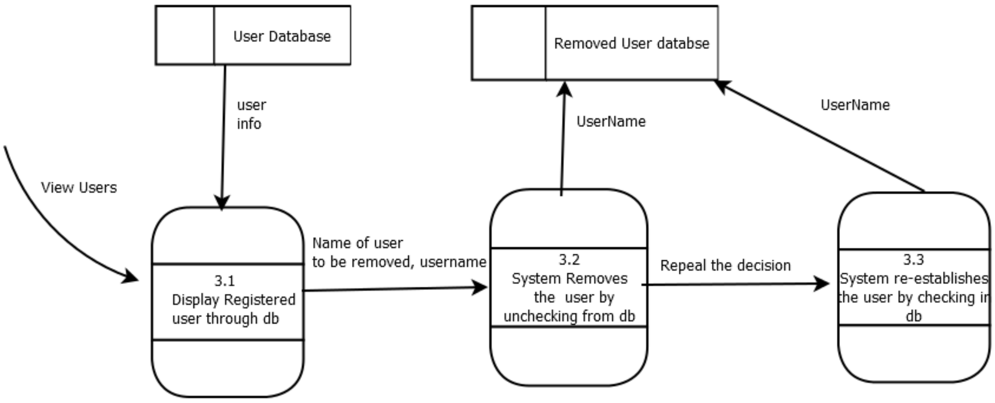
Level 1


Admin database
username,password
adminId,password
Total notes created
1.1 System Matches Admin Info by checking in db
admin validated
1.2 System Displays admin Interface
Remove notes
user not validated
View Users
1.3 System finds no match and displays login page

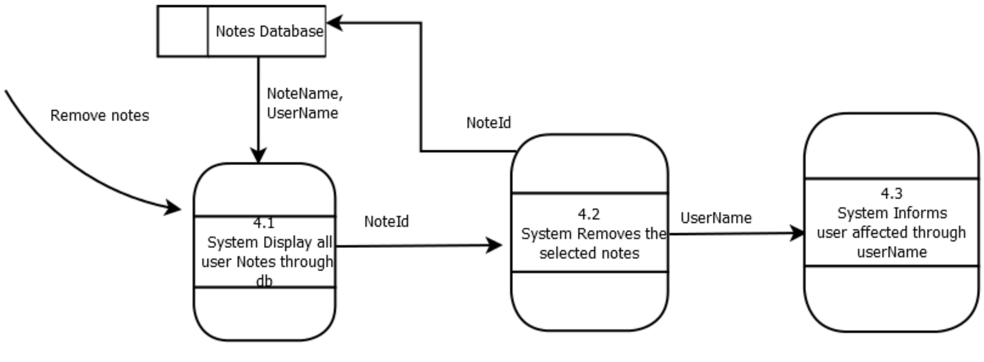
Level-2 Process-1


Notes Databse
NoteId,NoteName
Total Notes Created
2.1 System Displays available notes by checking in db
username
2.2 System Displays note along with the creator's name
Notes data
2.3 SystemDisplay created and modified dates by checking in db

Level-2 Process-2

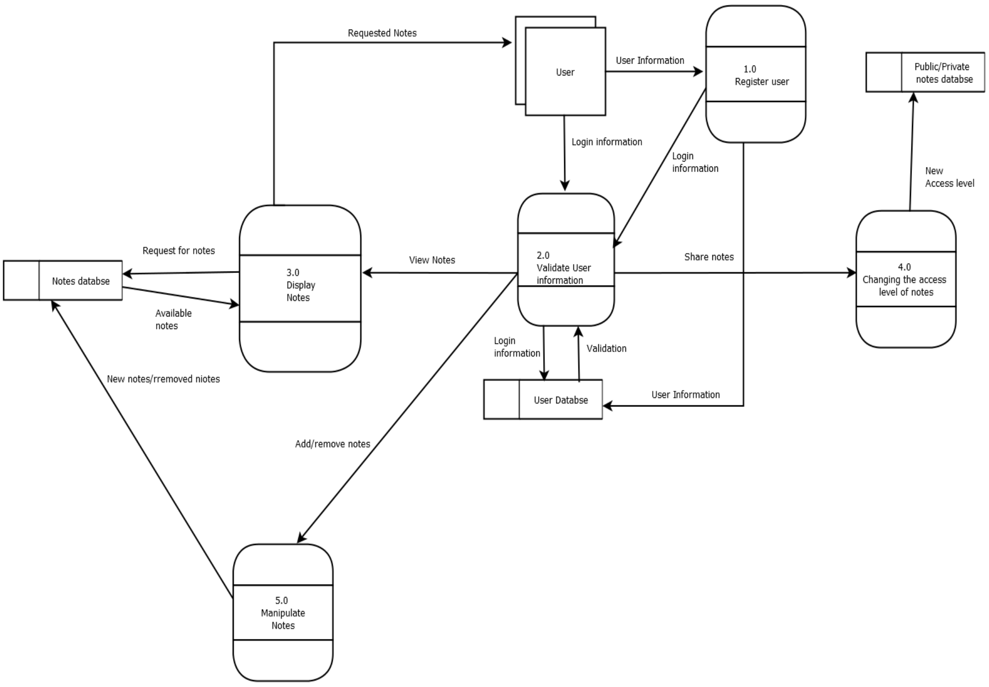

User Database
Removed User databse
user info
UserName
UserName
View Users
3.1 Display Registered user through db
Name of user to be removed, username
3.2 System Removes the user by unchecking from db
Repeal the decision
3.3 System re-establishes the user by checking in db

Level-2 Process-3

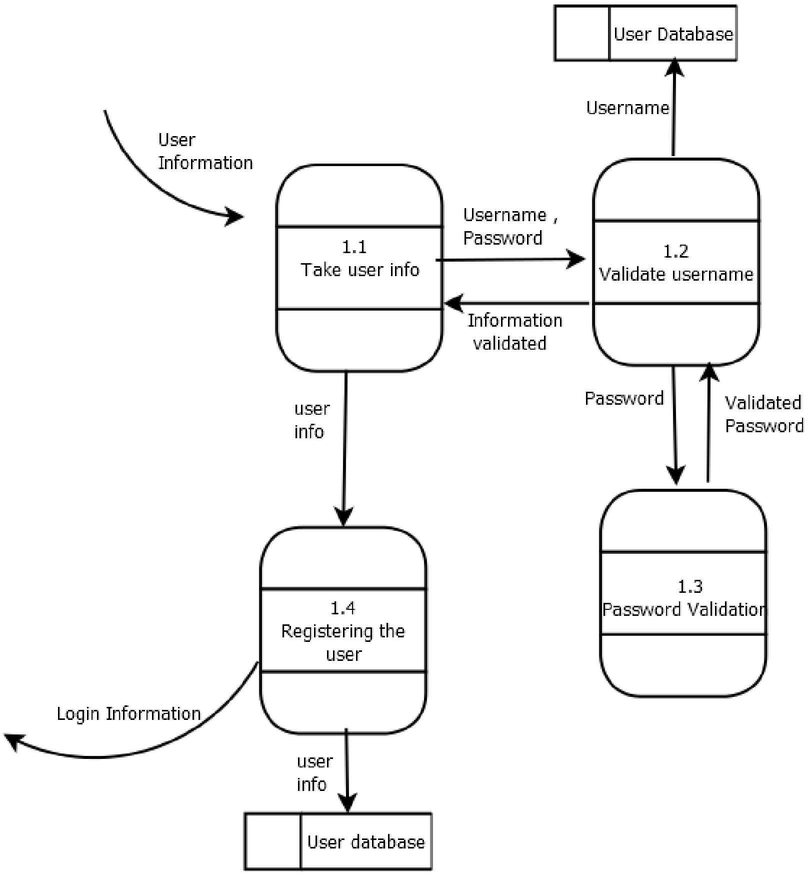

Notes Database
NoteName, UserName
Remove notes
NoteId
4.1 System Display all user Notes through db
NoteId
4.2 System Removes the selected notes
UserName
4.3 System Informs user affected through userName

Level-2 Process-4

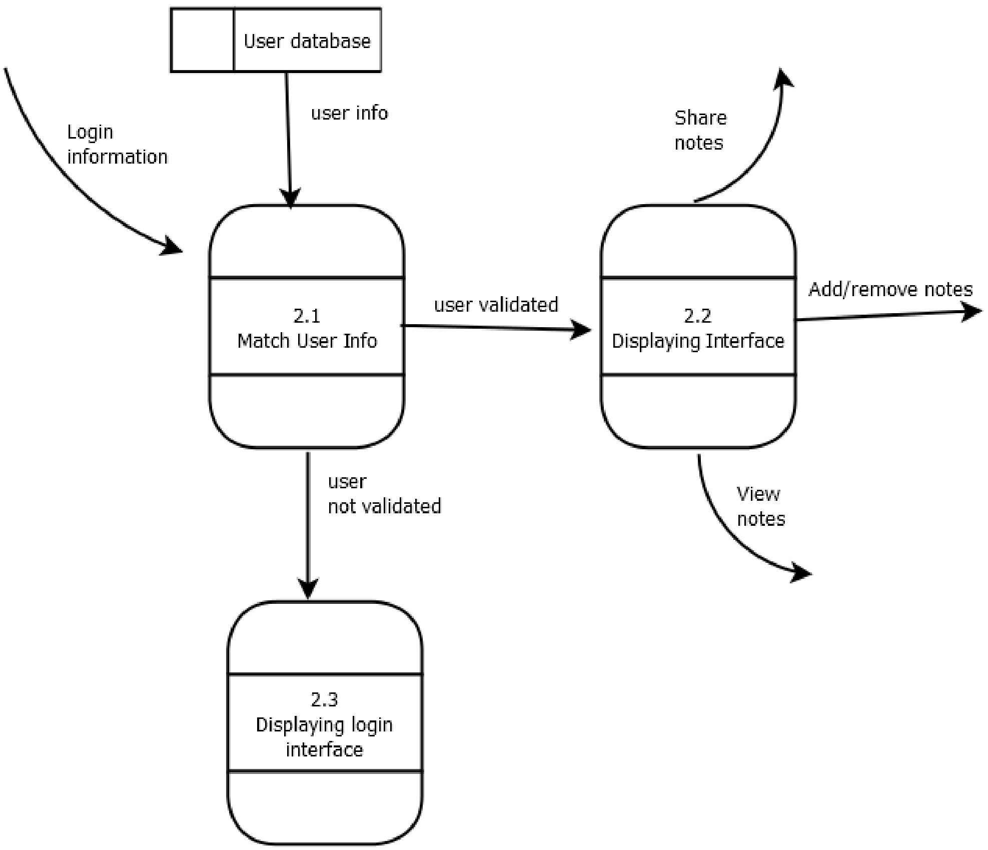
8.User logical data flow diagram



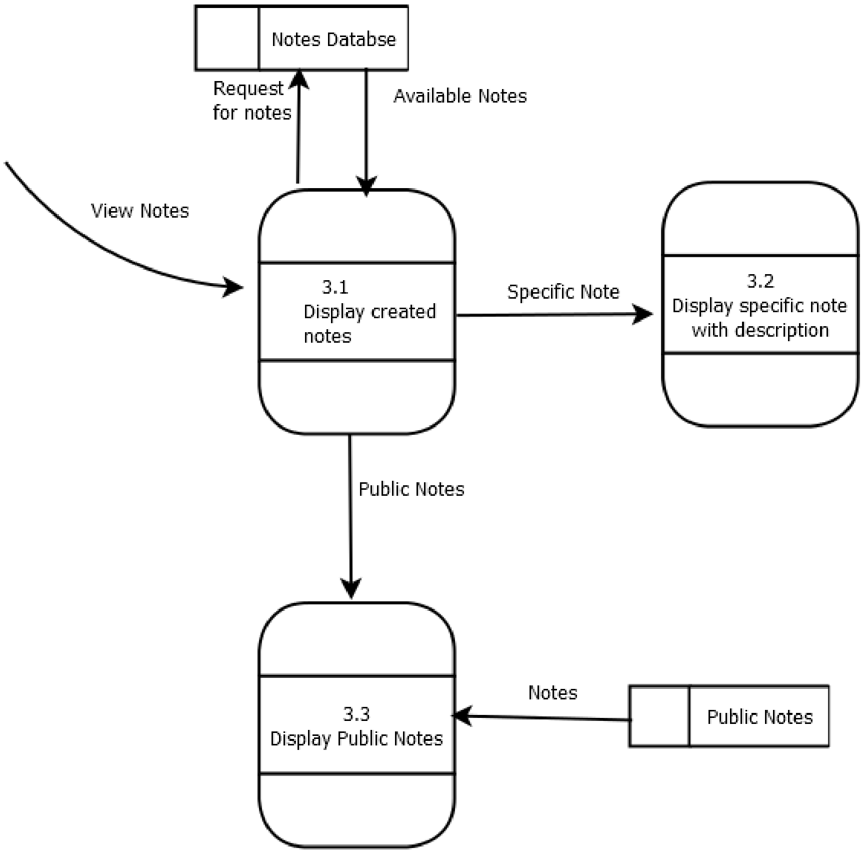
Level-1 DFD



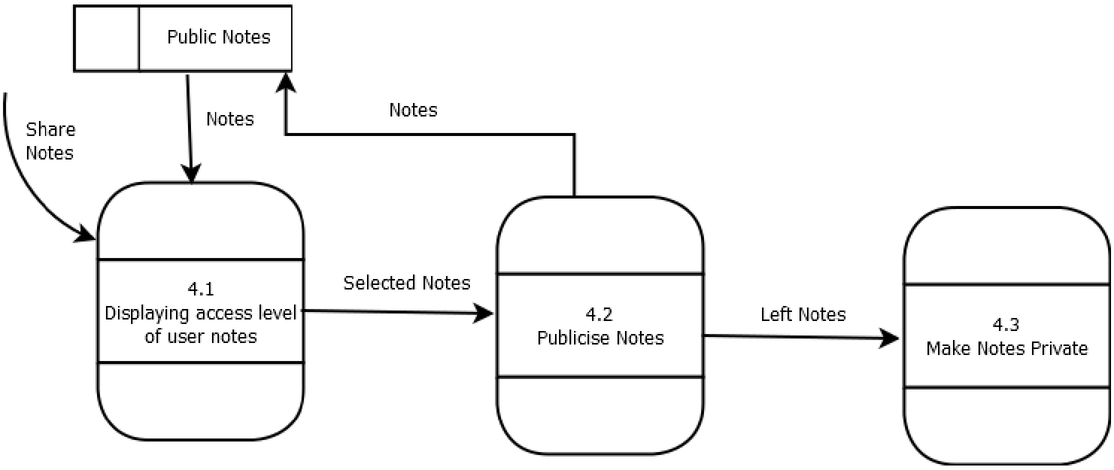
Level-2 Process-1



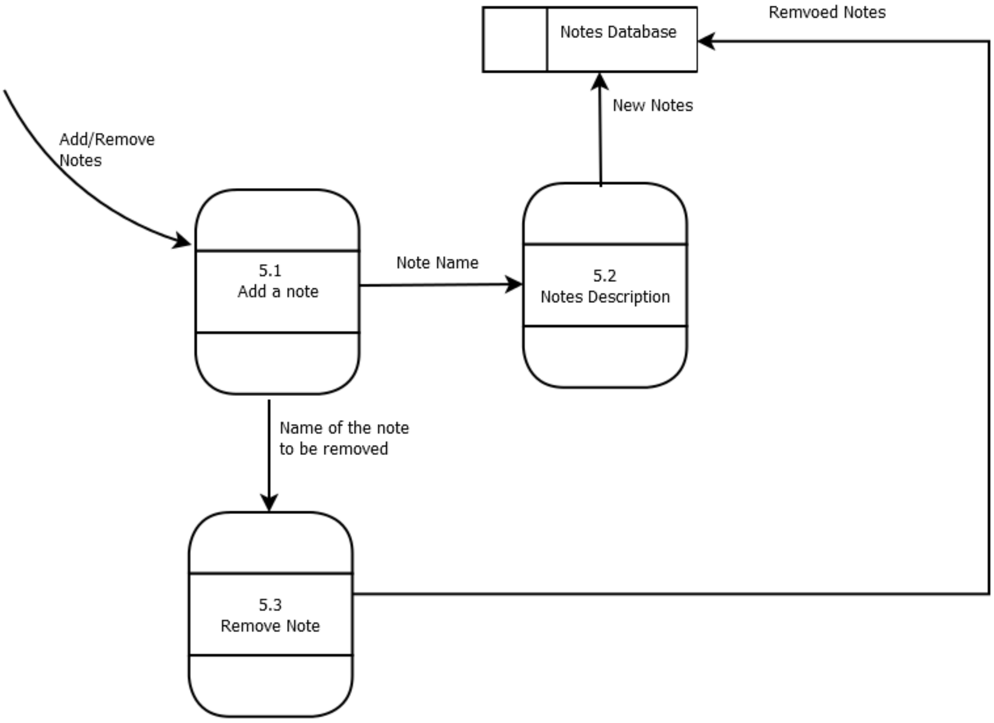
Level-2 Process-2



Level-2 Process-3

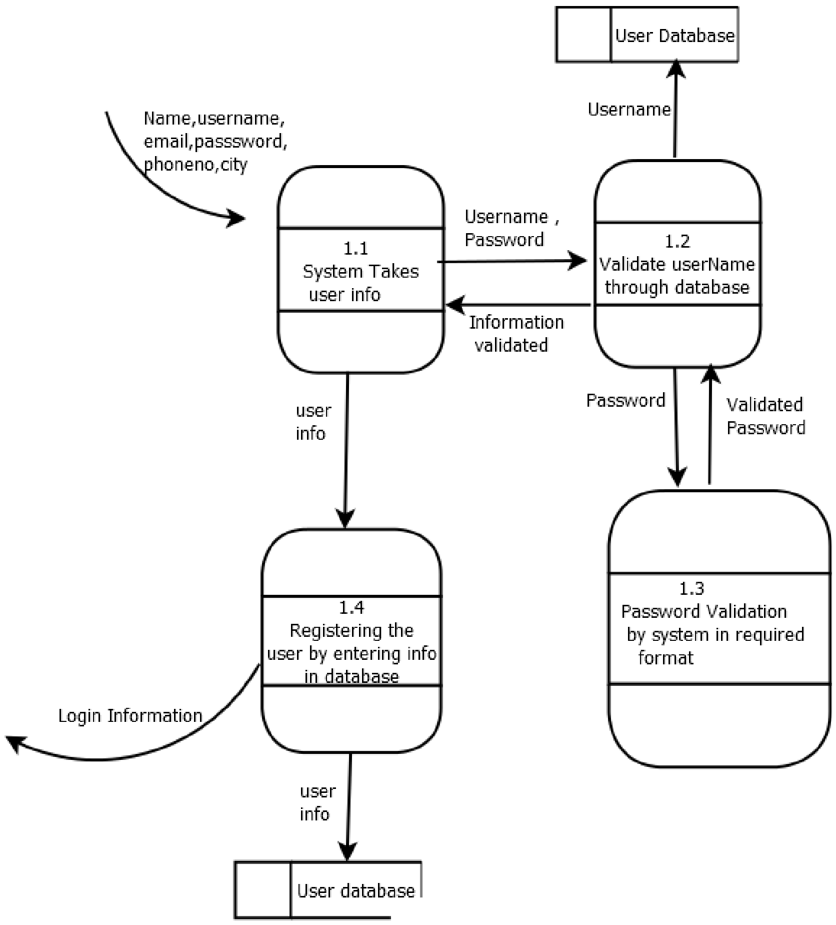


Level-2 Process-4

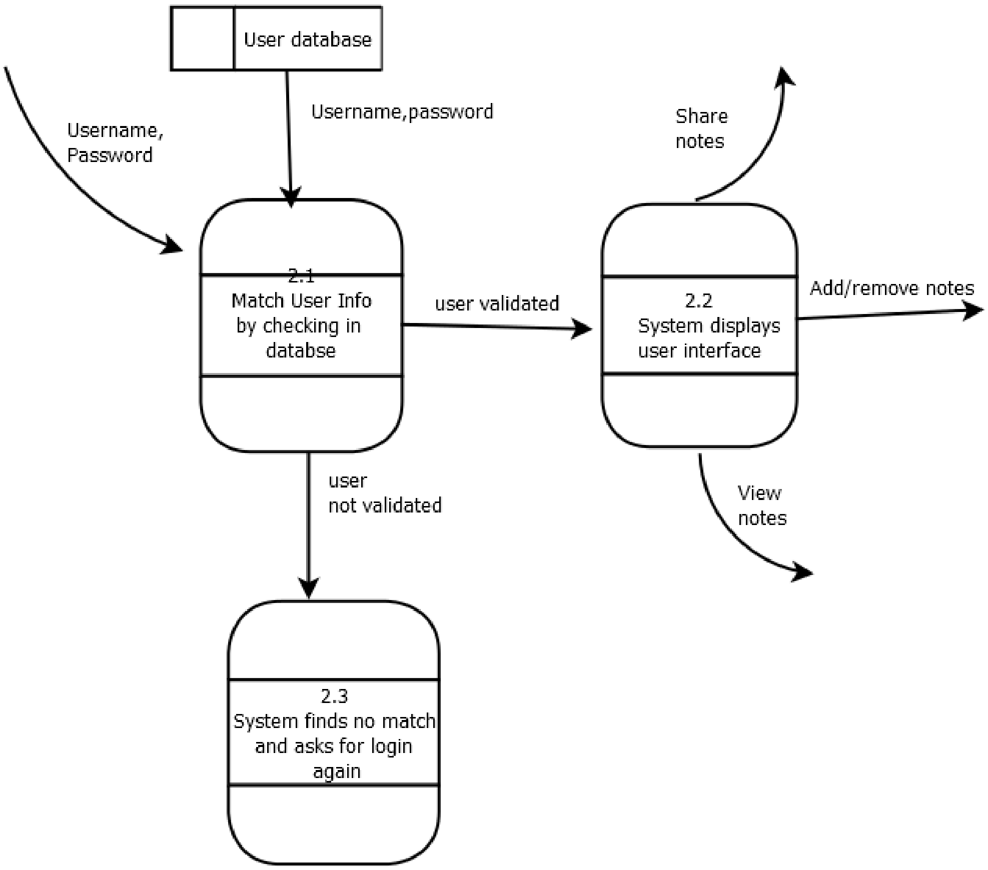


Level-2 Process-5

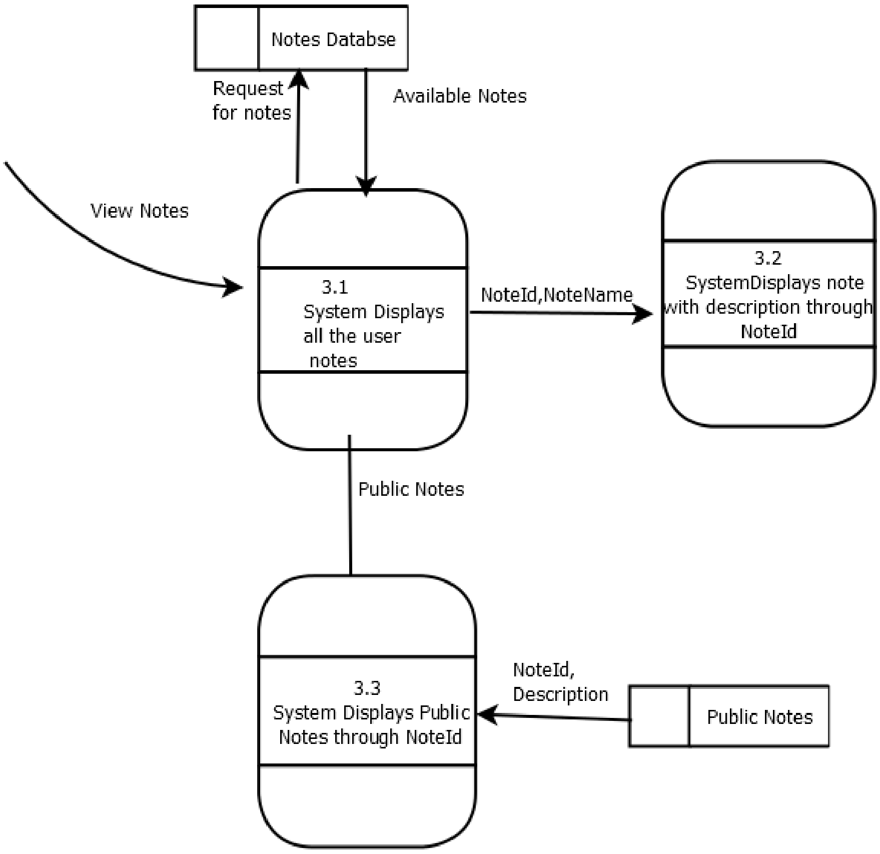
9.User physical data flow diagram



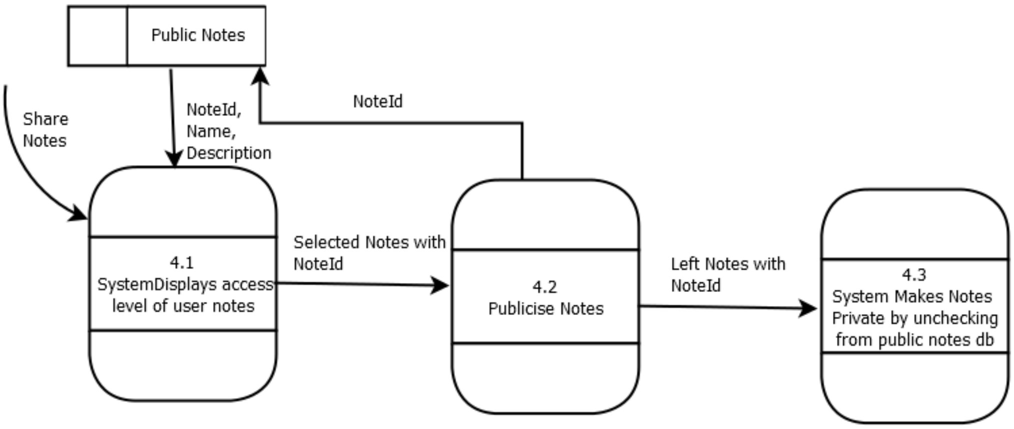
Level-2 Process-1



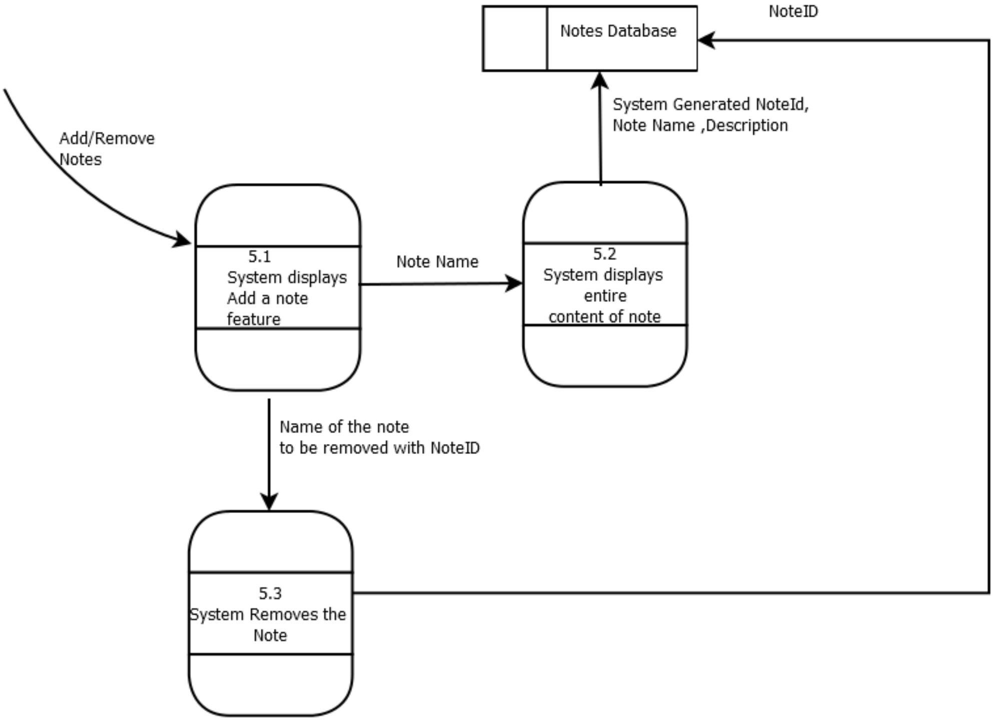
Level-2 Process-2


Notes Databse
Request for notes
Available Notes
View Notes
3.1 System Displays all the user notes
NoteId,NoteName
3.2 SystemDisplays note with description through NoteId
Public Notes
3.3 System Displays Public Notes through NoteId
NoteId, Description
Public Notes

Level-2 Process-3


Public Notes
Share Notes
NoteId, Name, Description
NoteId
4.1 SystemDisplays access level of user notes
Selected Notes with NoteId
4.2 Publicise Notes
Left Notes with NoteId
4.3 System Makes Notes Private by unchecking from public notes db

Level-2 Process-4



Level-2 Process-5

Appendix C: Agile Methodology (Increment Model)

Effort Indictor

Card No.	Card Functionality	Effort Indicator
1.	Collecting the requirements and understanding the features	4
2.	Creation of the Database.	2
3.	Creating login interface system for student and admin	1
4.	Creating signup interface system.	3
5.	Creating notes interface	1
6.	View users Interface	2
7.	View user’s Note Interface	1
8.	Creating access level interface	4
9.	Adding notes and deleting notes feature.	4
10.	Functionality for informing user’s about their notes removal	2
11.	Remove users interface	2

Incrementing Schedule

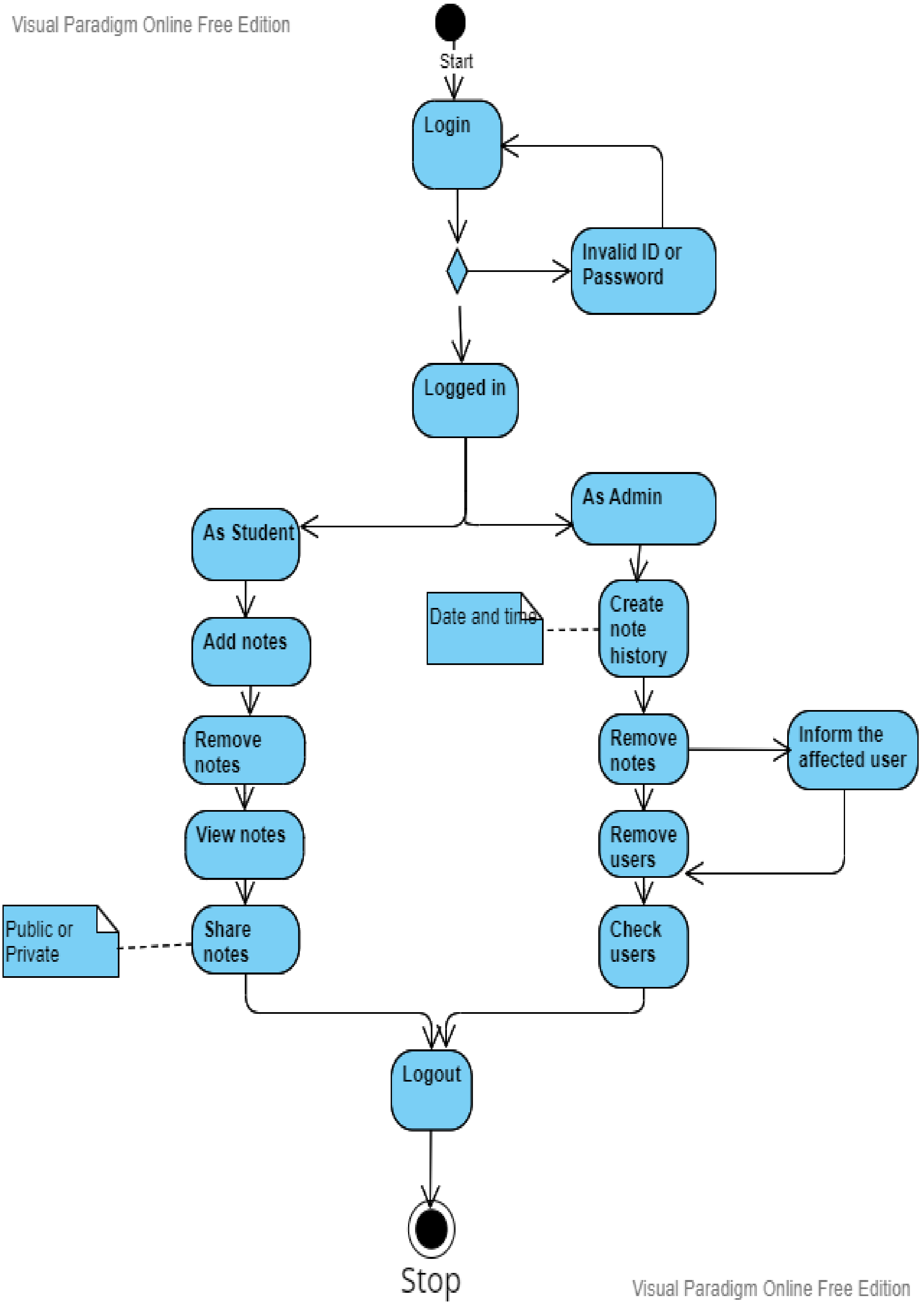
1. In the first increment we will gather the requirements and the learn about the features of the project.
2. In the second increment we will create our table and connect the keys (Primary and Foreign) with the tables properly.
3. In this increment we will be adding the login functionality both for user as well as for admin.
4. In next increment we will be adding Sign Up functionality
5. In the next increment we will create the project interface in HTML and CSS. We will create the buttons that will be required. We will also be creating the horizontal navigation bar in this increment.
6. In sixth increment we will add view users interface.
7. In the seventh increment, we will add view user's note interface, we will be also adding the left part of 6th increment.
8. In this increment we will be adding access level interface.
9. In the next increment we will add the adding and deleing notes .
10. This increment would add the functionality of letting user's know about the removal of their notes
11. In the last increment we will add the functionality of removing users by admin

Appendix D: To be determined list

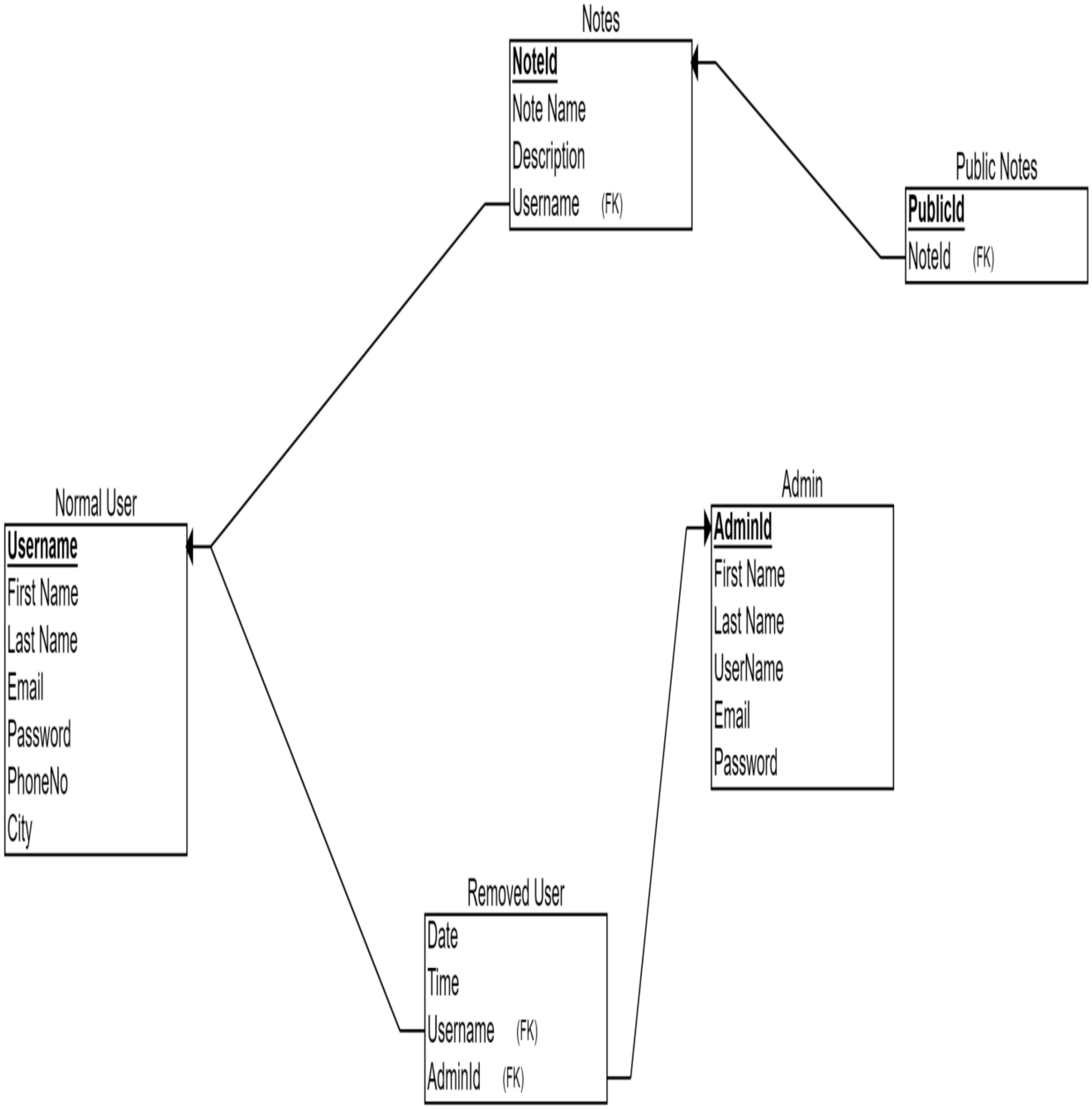
1. Verification through mobile number feature
2. Writing feedback on public notes
3. Automatic email notification regarding notes status when updated.
4. Feature of sharing notes to limited users and not to whole public

Design Model

Activity Diagram

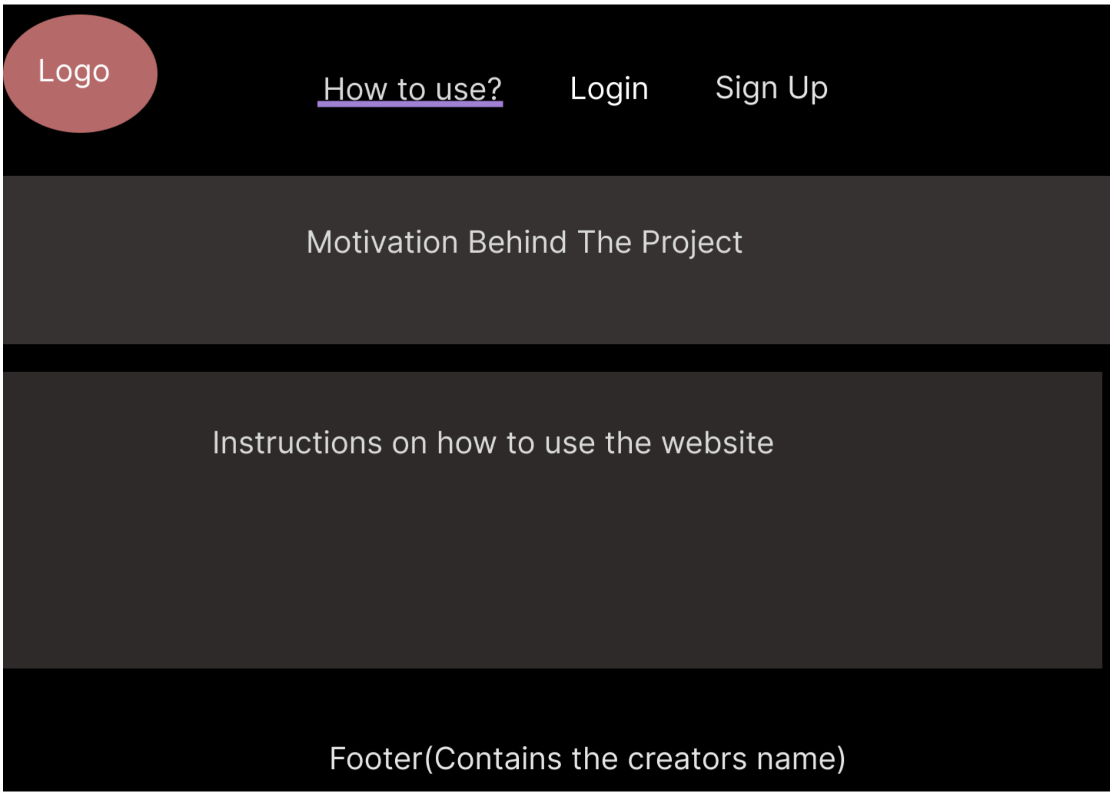


Schema Diagram

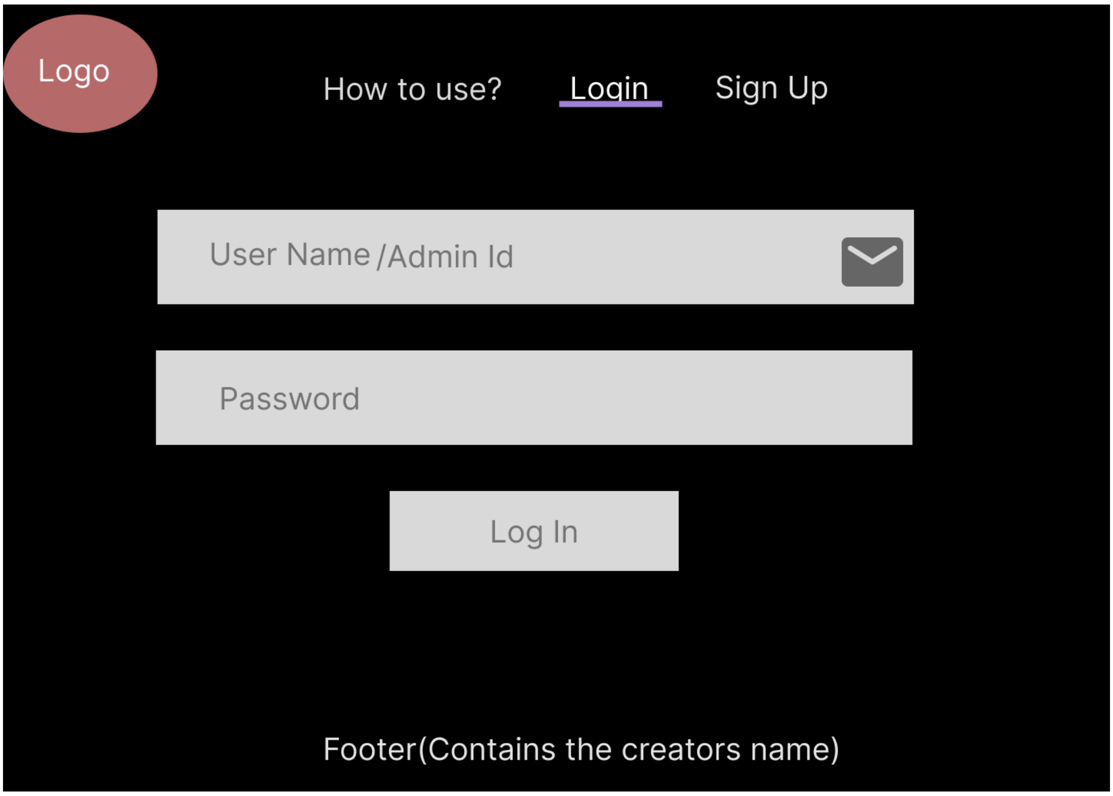


WireframeModel

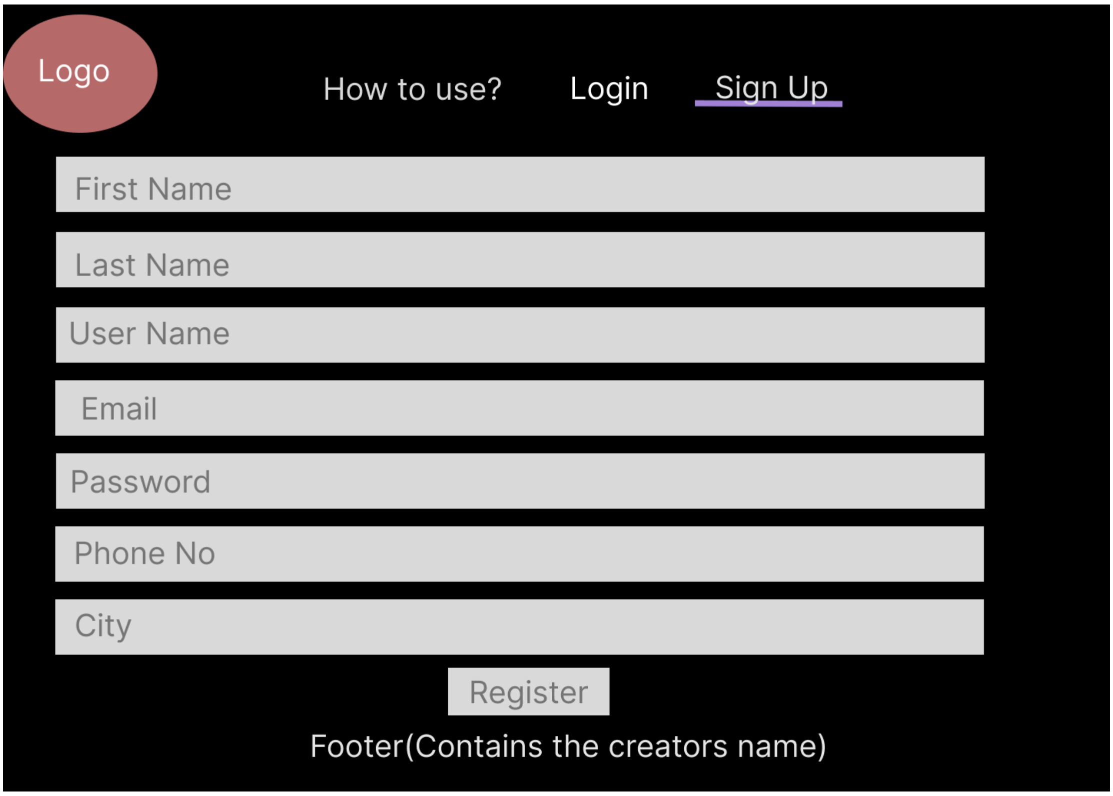
Page 1:How to use?


Logo
How to use?
Login
Sign Up
Motivation Behind The Project
Instructions on how to use the website
Footer(Contains the creators name)

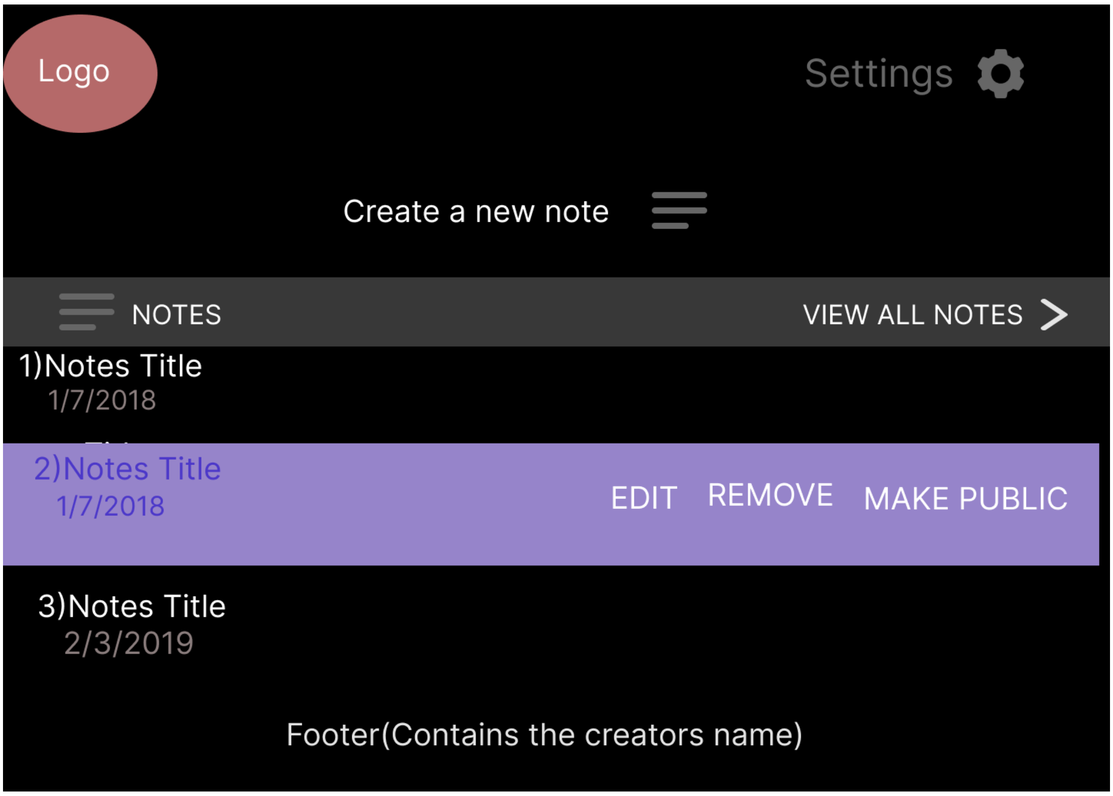
Page 2:Login



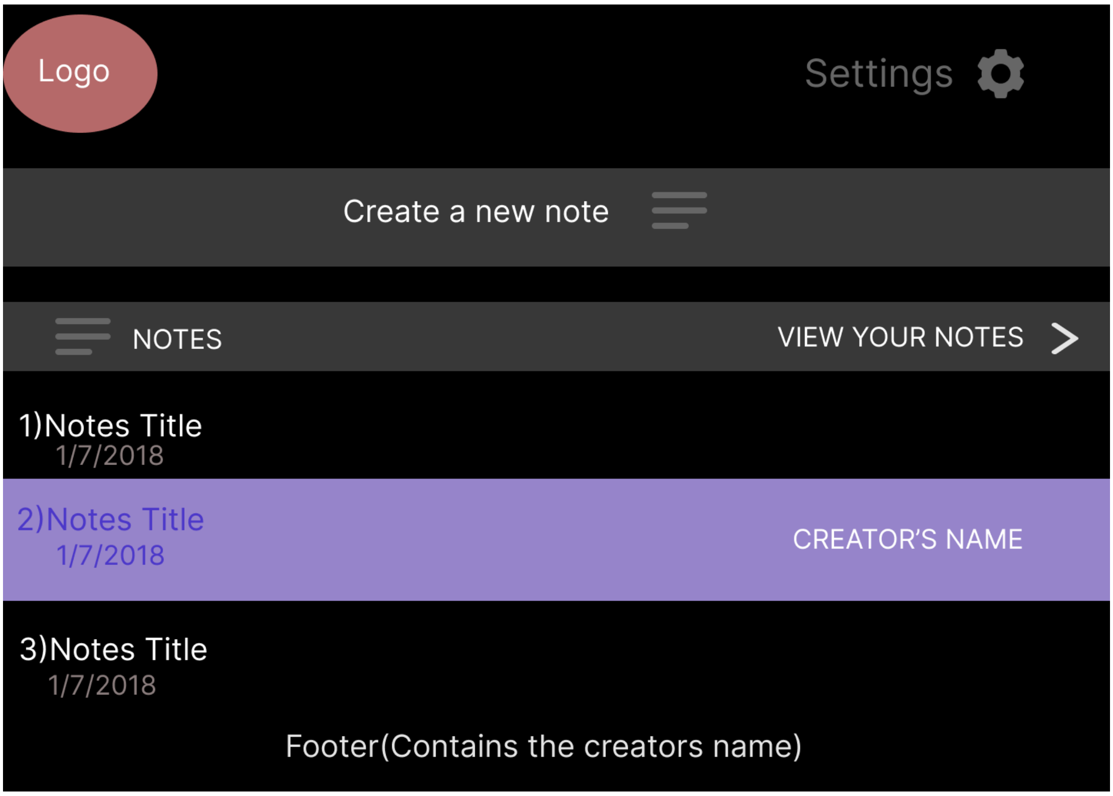
Page 3:Sign UP



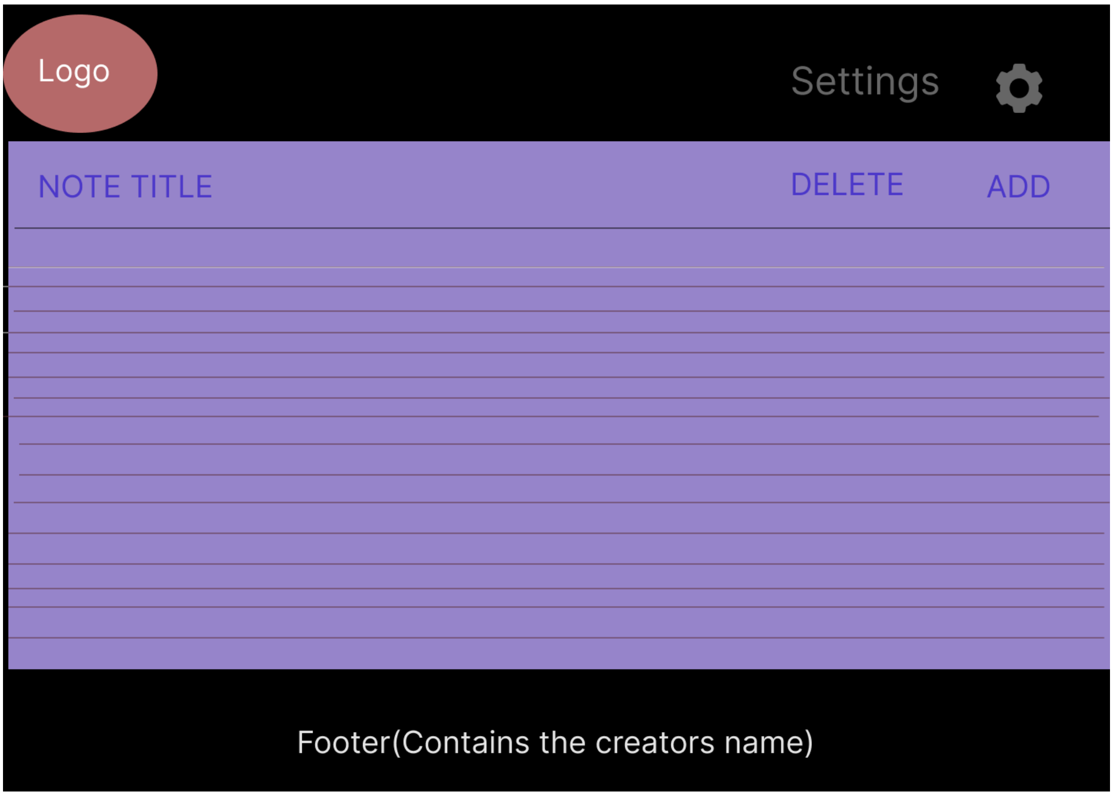
Page4: View user Notes



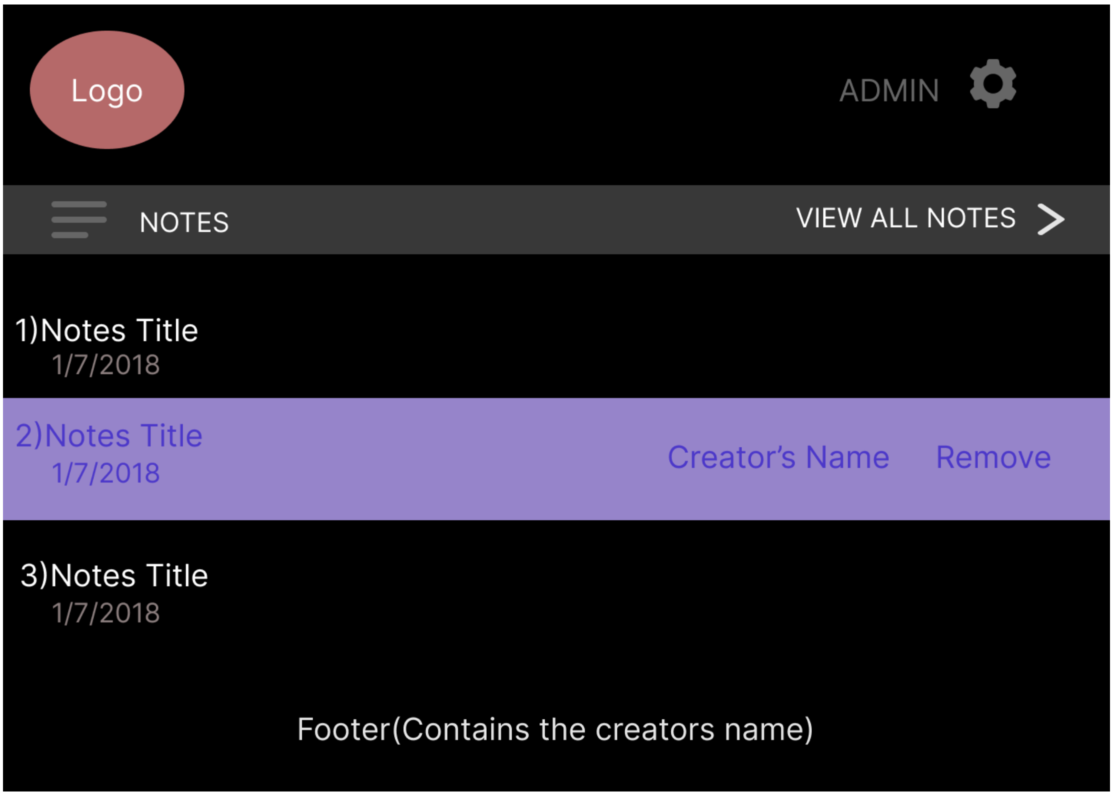
Page5:View public notes



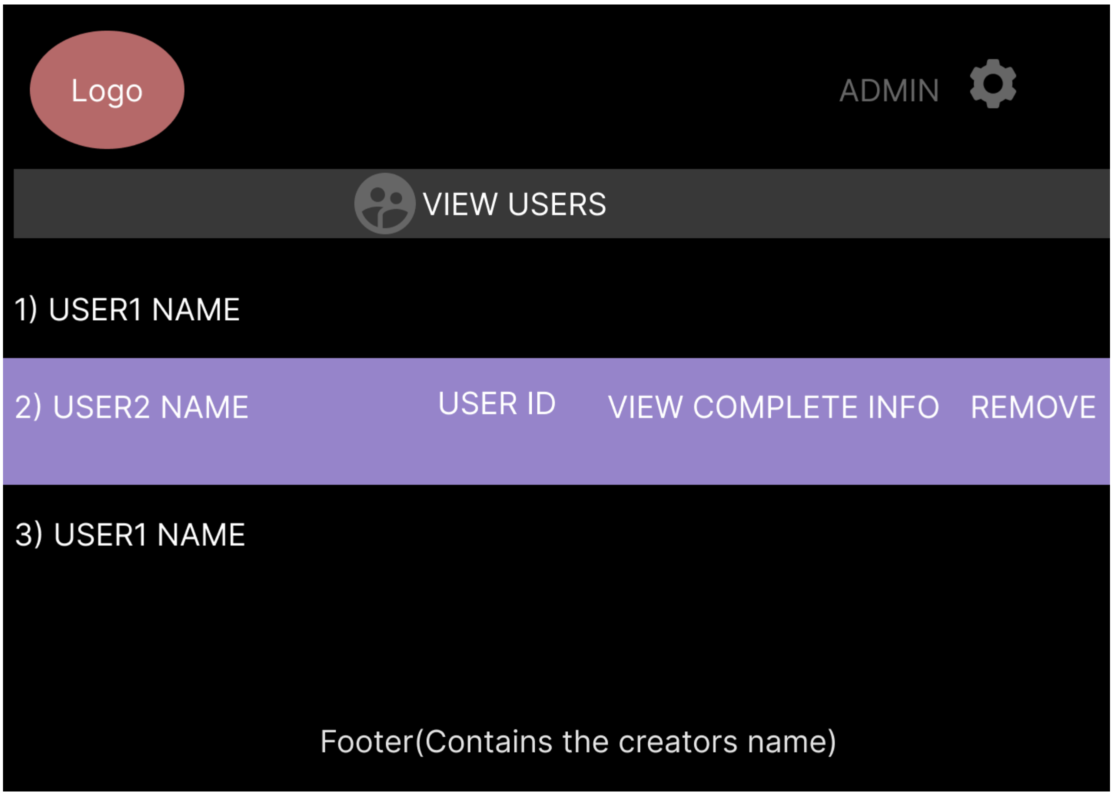
Page6:Add a note



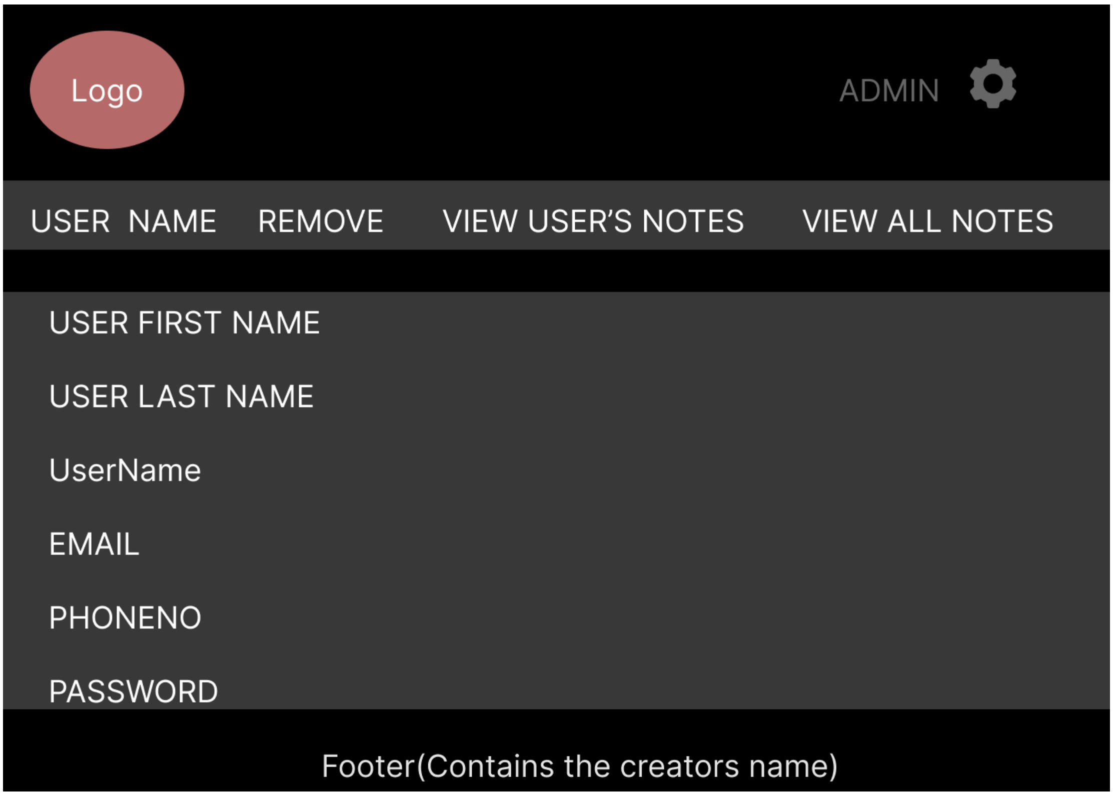
Page7: View Notes(As an admin)



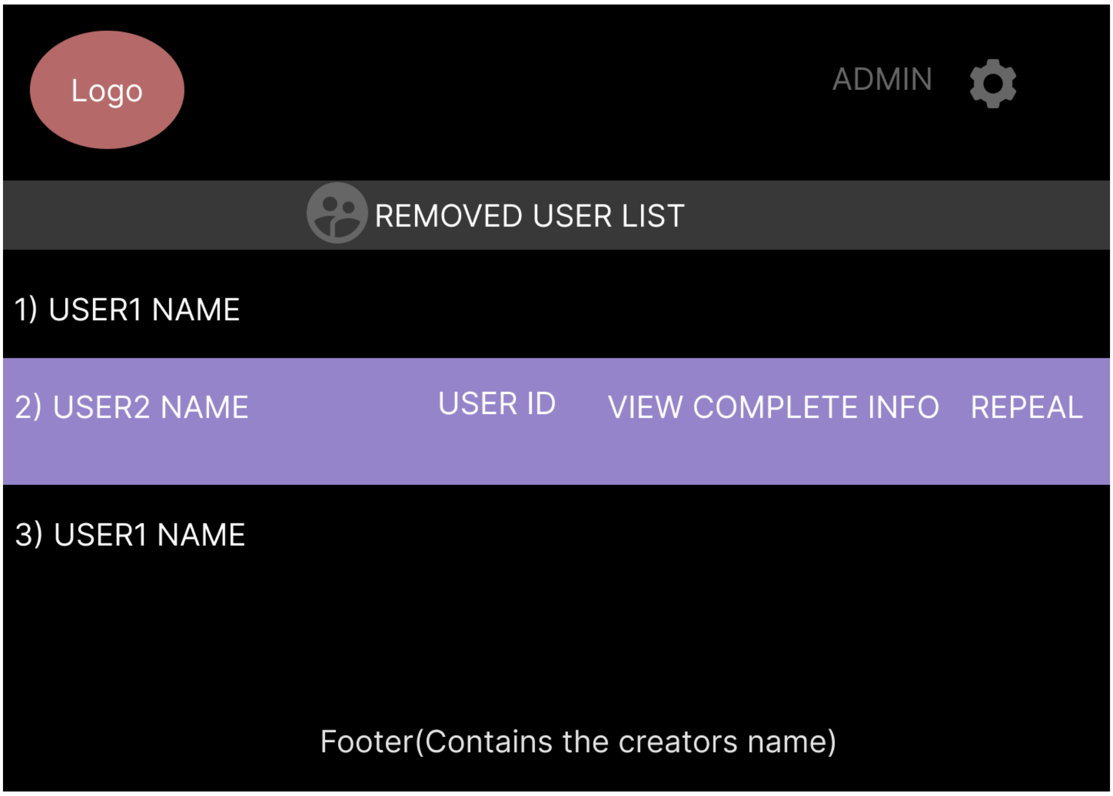
Page8:View Users



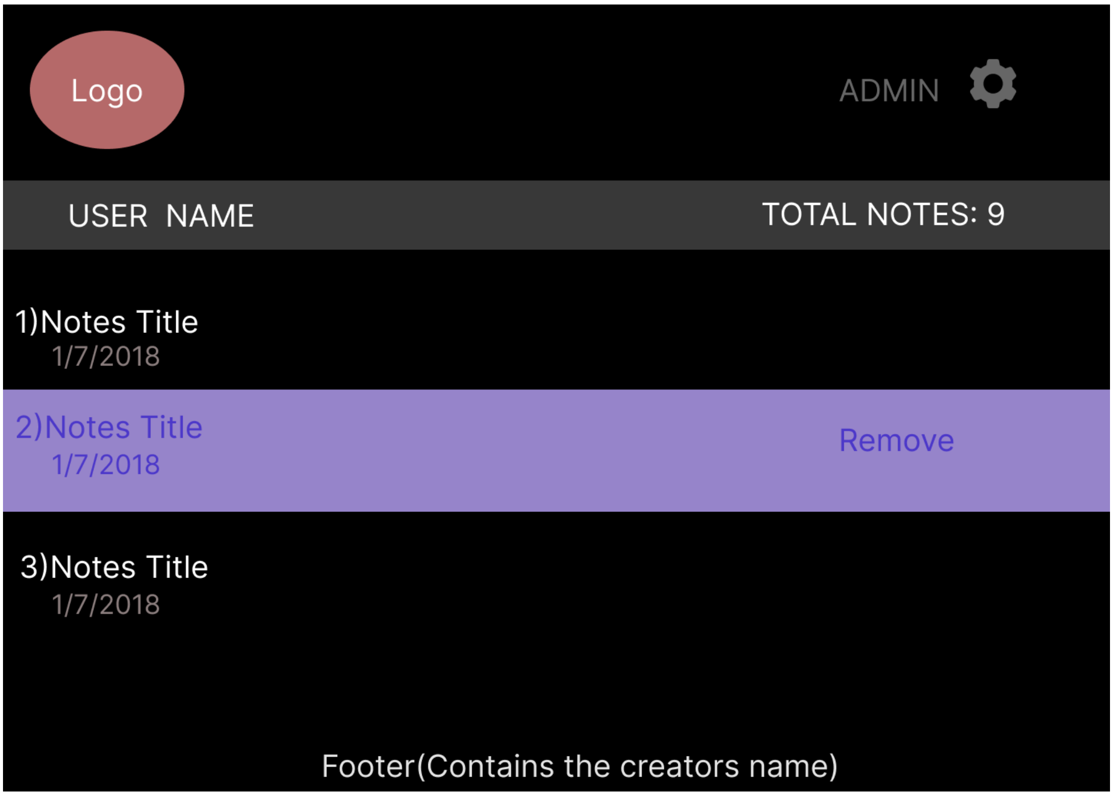
Page9:View particular user info(As an admin)



Page10:Removed user List



<u>Page11:View notes of a particular user(As an admin)</u>



Logo
ADMIN
USER NAME
TOTAL NOTES: 9
1)Notes Title
1/7/2018
2)Notes Title
1/7/2018
Remove
3)Notes Title
1/7/2018
Footer(Contains the creators name)