Note-Nodes App

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Computer Applications

By

Prajwal Negi	204200146
Lalit Rathore	204200110
Pradeep Sharma	204200145
Ravi Sharma	204200168
Om Singh Raghav	204200140
Nitesh Deshwal	204200139
Manish Tiwary	204200115

Under the Guidance of

Mr. Puneet Sharma

(Assistant Professor)

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA

December, 2022

Department of Computer Engineering and Applications GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road,
P.O. – Chaumuhan, Mathura – 281406
U.P (India)

Declaration

We hereby declare that the work which is being presented in the B.C.A. Project **"Note-Nodes App"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Computer Applications*** and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Mr. Puneet Sharma, (Assistant Professor of Computer Engineering Department).**

The contents of this project report, in full or in parts, have not been submitted to any other institute or university for the award of any degree.

Prajwal Negi	204200146
Lalit Rathore	204200110
Pradeep Sharma	204200145
Ravi Sharma	204200168
Om Singh Raghav	204200140
Nitesh Deshwal	204200139
Manish Tiwary	204200115

Abstract

Notes making has been one of the crucial parts of the student's academics and in order to strive for better grades they have to make better notes for each individual subjects in each semester.

This although sounds simple poses a quite big problem of managing and carrying their notes notebook in each individual subject classes, sometimes they make different notebooks for different subjects and sometimes they divide the notebook in different sections.

But even after all this, there is always a possibility of notes getting lost and pages getting torn, or water spills in notebooks and several problems that are uncalled for.

Plus carrying the notebook for each individual subjects adds to the weight that the student has to carry throughout their whole semester.

Thus there is a need of an app that can solve all these issues, since we are already living in an advanced digital era. These all problems can be simplified or can be solved a certain level with the introduction of an app.

CONTENTS

Certificate

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

Project Supervisor
(Mr. Puneet Sharma)

Date: 10/12/2022

Project In charge	Program Coordinator	Head of Department
(Mr. D.P Yadav)	**(Mr. Narendra Mohan)**	**(Prof. Rohit Agrawal)**

ACKNOWLEDGEMENT

The satisfaction which accompanies the successful completion of the project, is incomplete without the mention of a few names. We take this opportunity to acknowledge the efforts of the many individuals who helped us to make this project possible.

Firstly, we would like to express our heartfelt appreciation and gratitude to our project guide **Mr. Puneet Sharma, (Assistant Professor), Computer Engineering Department.** His vision and execution aimed at creating a structure, definition and realism around the project and fostered the ideal environment for us to learn and do. This project is a result of he's teaching, encouragement and inputs in the numerous meetings he had with us, despite his busy schedule.

We would also like to extend our immense gratitude to respected Head of Department **Prof. Rohit Agrawal** who allowed us to choose the topic for our dissertation.

The experience was novel one and we would like to thank all the people, who have lent their valuable time for the completion of the report. Without their consideration it would have been difficult to complete the report.

Introduction

Present Problem Statement:

- Notes Management has been one of the quite hard tasks for the students
- Carrying the notes copy for each individual subject accounts for the bulk the students have to carry throughout the whole time in academics
- Plus sharing and management of the notes is also hard as the paper being soft gets torn and plus there is always a possibility that the notes can get lost
- Other options could be to store the images of the notes but then it gets bulky and management of the notes become very poor in the image application and sorting and organizing is nearly impossible when you have thousands of notes.

Proposed System:

- The Application provides the student to store their notes in the apps by using an interactive interface
- The Application is connected to cloud storage this, provides the feature to login and carry your notes anywhere. Multiple device login is also allowed so several users can access it
- User can create their account through their email as well.
- The Notes are sorted by their names, so Student just has to search for the name of the subject and thus they can find their respective notes, and share and edit it.
- Notes can be shared to other users by converting them to pdf and word documents which can further be shared to other users via WhatsApp, Telegram, or even Google-Drive or any platform that supports word or pdf documents.

Overview and Motivation

Overview:

The main objective of our project is to make the notes carrying and sharing process hassle free since there have many notes making app in the market already but none of them provide the facility to convert it to the pdf and word format (more editable format). Students won't have to carry individual notebooks for the each subject just to make some notes, they can carry just one notebook and write their notes on it on the go and then take a pic of it and add it to the notes app, which will be saved in the cloud and later can be edited and shared among other users.

Plus it would be much easier to organize and there won't be any problem regarding the notes being lost.

Motivation:

- Utilizing the new technology to keep and manage the notes in more verbose manner
- Saving the Excessive papers and carry load of the students books
- Providing a platform for the users to share the notes in doc or pdf form

1.1 Organization of Project Report:

PHASES	TIME DURATION
Software requirement specification	1 weeks
System design	2 weeks
Coding	8 weeks
Testing	1 weeks
Documentation	1 weeks
Implementation	1 weeks

Software Requirement Analysis

System Analysis is a detailed study of the various operations performed by a system and their relationship within and outside the system. It is a systematic technique that defines goals and objectives the goal of the development is to deliver the system in the line with the user's requirements, and analysis is this process.

System study has been conducted with the following objectives in mind: -

- Identify the client's need.
- Evaluate the system concept for feasibility.
- Perform economical and technical analysis.
- Allocate functional to hardware, software, people, database and other system elements
- Establish cost and schedule constraints.
- Both hardware and software expertise are required to successfully attain the objectives.
- To minimize the resources used and to maximize the efficiency.

2.1 Requirement Analysis

Information gathering is usually the first phase of the software development project. The purpose of this phase is to identify and document the exact requirements for the system. The user's request identifies the need for a new information system and on investigation re-defined the new problem to be based on MIS, which supports management. The objective is to determine whether the request is valid and feasible before a recommendation is made to build a new or existing manual system continue

The major steps are –

- Defining the user requirements.
- Studying the present system to verify the problem.
- Defining the performance expected by the candidate to use requirement.

2.1.1 Hardware Requirements

Processor	: Snapdragon dual core
Processor Speed	: 1Ghz
RAM	: 1 GB
Storage	: 1 GB

2.1.2 Software Requirements

Operating System	: Android 10.0 or above
Front End	: Java and XML
Back End	: Firebase and Java

2.1.3 Tools and Technology Tools:

- Windows 10 or Higher
- Android Studio
- Firebase
- Apache

Technology:

- **Java:**

Java is a popular programming language that is widely used in the development of mobile applications for the Android operating system. Java is known for its simplicity, reliability, and security, making it a great choice for building mobile apps.

One of the key advantages of using Java for Android app development is that it is an object-oriented language. This means that it allows developers to create complex, modular programs by organizing data and functionality into reusable classes and objects. This makes it easy to write, maintain, and debug code, and helps to ensure that the app is organized and efficient.

Java is also highly interoperable, meaning that it can easily integrate with other languages and frameworks. This makes it possible to use Java in conjunction with other technologies, such as C++, to create powerful and flexible mobile apps.

Another important benefit of using Java for Android app development is its portability. Java code can run on any platform that supports the Java Virtual Machine (JVM), which means that it can be used to develop apps that can run on a wide range of devices. This makes it easy to create apps that can be used on different types of smartphones, tablets, and other mobile devices.

In addition to these benefits, Java is also a powerful and versatile language. It offers a wide range of built-in features and libraries that can be used to create complex,

feature-rich mobile apps. These include support for concurrency, networking, and graphics, as well as a rich set of APIs that make it easy to access and manipulate data.

Overall, Java is an excellent choice for Android app development. Its simplicity, reliability, and portability make it a great option for building high-quality, feature-rich mobile apps that can run on a wide range of devices.

- **XML:**

XML, or Extensible Markup Language, is a standard markup language used for organizing, storing, and transmitting data. It is a popular choice for creating and sharing data between applications and across the web, and it is often used in Android app development.

In Android, XML is used to create layout files that define the user interface for an app. These files contain a series of elements and attributes that specify the structure and layout of the user interface, including the position and size of each element on the screen. For example, an XML layout file might contain a **<TextView>** element that displays a piece of text, or a **<Button>** element that allows the user to interact with the app.

XML layout files are used in Android because they provide a flexible and customizable way to create the user interface for an app. The layout is defined in a separate file, which makes it easy to modify and maintain without changing the underlying code. This also allows developers to create a consistent look and

feel across different parts of the app, and to reuse elements and layouts in multiple places.

- **Android Studio:**

Android Studio is a popular integrated development environment (IDE) for developing Android apps. It is based on the IntelliJ IDEA platform and offers a range of features to help developers create high-quality, efficient, and responsive Android apps.

Some key features of Android Studio include a visual layout editor, a code editor with intelligent code completion, and a flexible build system. It also offers a range of tools for debugging and testing, including a built-in emulator, support for third-party testing frameworks, and a rich set of performance and memory profiling tools.

One of the key advantages of Android Studio is its tight integration with the Android SDK, which makes it easy to manage the various components of the Android platform, including the Android platform itself, the Android Support Library, and the Google Play Services Library.

Android Studio also offers a range of customization options, including support for custom themes and plugins, as well as support for a range of popular version control systems. This makes it easy for developers to work collaboratively on large-scale projects, and to integrate their work with other tools and frameworks.

Overall, Android Studio is a powerful and comprehensive tool for developing Android apps, and offers a range of features to help developers create high-quality, efficient, and responsive apps quickly and easily.

- **FIGMA:**

Figma is a powerful design tool that allows users to create high-quality, professional designs for websites, apps, and other digital products. With Figma, designers can work together in real-time, making it a great option for collaborative design work. The interface is intuitive and user-friendly, so even beginners can get started quickly.

One of the key features of Figma is its online nature. Unlike other design tools that require users to download and install software, Figma can be accessed directly from a web browser. This makes it easy to work on designs from anywhere, and to share them with others.

Figma also offers a range of powerful features that make it a versatile design tool. For example, users can create complex layouts and designs using layers, groups, and frames. Figma also offers a range of design tools, including vector tools, text tools, and color tools.

Another advantage of Figma is its collaboration features. Users can share designs with others, and can see and comment on each other's work in real-time. This makes it easy to work together on a project, and to get feedback from others.

Overall, Figma is a powerful and versatile design tool that offers a range of features to help designers create professional-quality designs. With its online nature,

collaboration features, and powerful design tools, Figma is a great option for anyone looking to create high-quality designs.

- **Firebase:**

Google Firebase is a platform that provides a range of services and tools for building, improving, and growing mobile and web applications. It offers a variety of features that make it easier for developers to create high-quality, scalable, and secure apps, including a real-time database, user authentication, hosting, cloud storage, and analytics.

One of the main advantages of using Firebase is that it is a fully managed platform, which means that developers don't have to worry about managing servers or infrastructure. This allows them to focus on building their app and provides them with the scalability and reliability needed to support a large number of users.

Another key benefit of Firebase is that it integrates seamlessly with other Google products, such as Google Cloud Platform, Google Analytics, and Google Ads. This allows developers to easily integrate their app with other Google services, providing them with a more comprehensive toolset for building and growing their app.

Firebase also offers a variety of tools and services for improving app performance and user engagement. For example, it provides tools for A/B testing, user segment wielded in the following manner:

Firebase Realtime Database: This is a cloud-hosted NoSQL database that allows developers to store and synchronize data in real time. It provides offline support and

automatic scaling, making it easy to build apps that can handle a large number of users and a large amount of data.

Firebase Authentication: This is a service that provides a secure and easy-to-use authentication system for mobile and web apps. It supports a variety of authentication methods, including email and password, phone number, and social media accounts.

Firebase Hosting: This is a service that provides fast and secure hosting for web applications. It offers automatic scaling and global CDN support, making it easy to deliver content quickly and reliably to users around the world.

Overall, Google Firebase is a powerful platform that provides a range of tools and services for building, improving, and growing mobile and web applications. Its fully managed nature and integration with other Google products make it a valuable tool for developers

2.3 Analysis

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

In the process of system analysis one should first understand that, what the present system is, is how it works (i.e. Processes).After analyzing these points we become able to identify the problems in the present system. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail. After this model, from three different aspects of the system-data, function and behavior. The models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

Modules:

1. **User/Student:** The person who logs in to the app for using it for further process
2. **Server:** It's where all the data of the users will be stored

Software Design

3.1 Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical representation of the flow of data within a system. It is used to visualize how data is processed and transformed within a system, as well as the relationships between different components in the system. DFDs are often used in software engineering and systems analysis to model the flow of data within a system, and to identify potential bottlenecks and inefficiencies in the system.

A DFD consists of a series of symbols and notations that represent the various components of a system, and the flow of data between them. The main components of a DFD are:

External entities: These are the entities outside the system that interact with it, such as users, other systems, or external data sources.

Processes: These are the actions or transformations that are performed on the data as it flows through the system.

Data stores: These are the repositories where data is stored, such as databases or file systems.

Data flows: These are the paths along which data moves within the system.

DFDs can be used to model systems at different levels of abstraction, from high-level overviews that show the overall flow of data within a system, to detailed diagrams that show the precise flow of data between specific components. DFDs can also be used to model the flow of data in real-time systems, such as those used for monitoring or control.

One of the key benefits of using DFDs is that they provide a visual

representation of the flow of data within a system, making it easy to understand and communicate. This can be especially helpful when working with complex systems, or when collaborating with others on a project. Additionally, by modeling the flow of data within a system, DFDs can help identify potential problems or inefficiencies, and can provide a starting point for redesigning or improving a system.

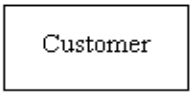
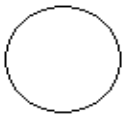
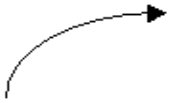
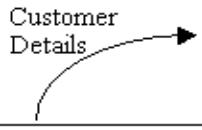
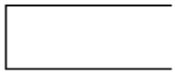
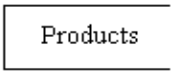
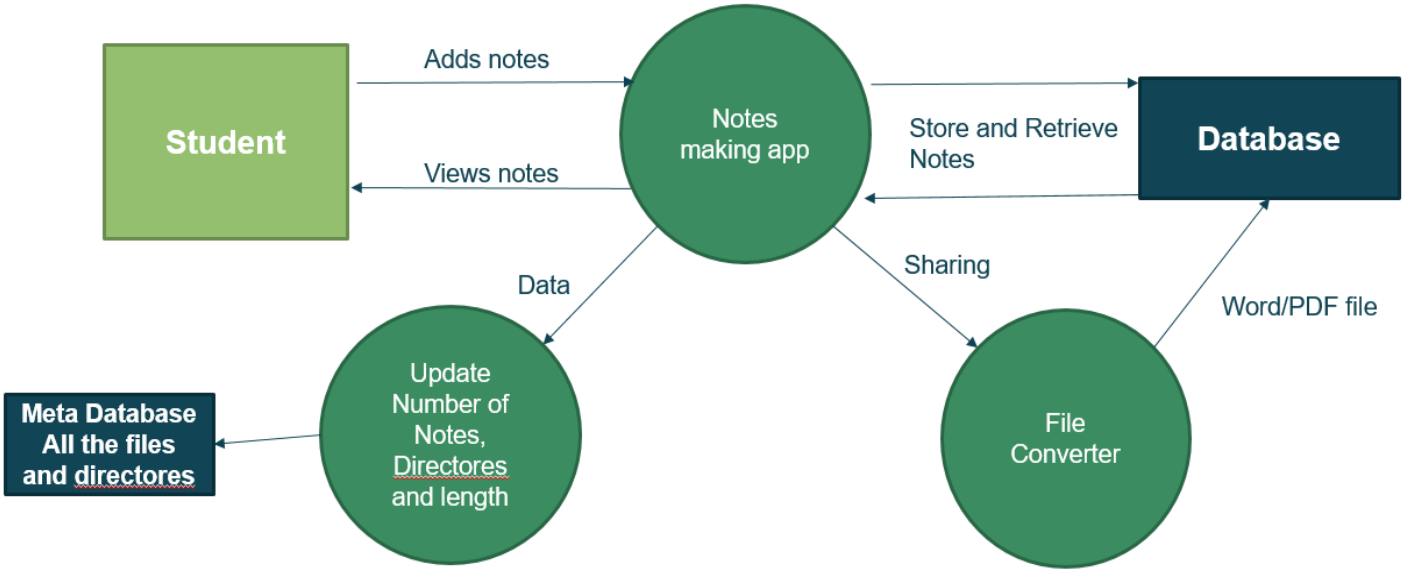
Name	Symbol	Description	Example
Entity		Used to represent people and organizations outside the system. They either input information to the system, accept output information from the system or both	
Process		These are actions that are carried out with the data that flows around the system. A process accepts input data and produces data that it passes on to another part of the DFD	
Data Flow		These represent the flow of data to or from a process	
Data Store		This is a place where data is stored either temporarily or permanently	

Fig 3.1 Data Flow Diagram Symbol

Data Flow Diagram level - 0

Data Flow Diagram Level 1



3.3 Database Design

A good database design is crucial for a high-performance application, just as an Aerodynamic body is important to a race car. If the car doesn't have smooth lines, it will produce drag and go slower. Without optimized relationships, your database won't perform as efficiently as possible. Thinking about relationships and database efficiency is part of normalization.

Beyond the issue of performance is the issue of maintenance—your database should be easy to maintain. This includes storing only a limited amount (if any) of repetitive data. If you have a lot of repetitive data and one instance of that data undergoes a change (such as a name change), that change has to be made for all occurrences of the data. To eliminate duplication and enhance your ability to maintain the data.

3.3.1 Entity Relationship Diagram (ER-Diagram)

An entity-relationship (ER) diagram is a graphical representation of the entities and relationships within a database. It is used to model the structure of a database, providing a visual representation of the relationships between different entities and their attributes.

An ER diagram typically consists of a set of rectangles, each of which represents an entity within the system. These rectangles are connected by lines, which represent the relationships between the entities. Each relationship is labeled with a verb that describes the type of relationship, such as "contains" or "has."

ER diagrams are useful for a variety of purposes, including database design and data

modeling. They can help designers and developers understand the requirements of a system and identify potential issues or areas for improvement. They can also be used to communicate the design of a database to others, such as stakeholders or team members.

There are several different types of ER diagrams, including conceptual, logical, and physical. Conceptual diagrams are high-level and abstract, providing an overview of the entities and relationships within a system. Logical diagrams, on the other hand, provide a more detailed representation of the entities and relationships within a database. Physical diagrams, meanwhile, show the specific implementation of a database, including tables and columns.

ER diagrams are typically created using specialized software, such as Microsoft Visio or Lucidchart. These tools provide a range of features and functionality for creating and editing ER diagrams, such as the ability to add and edit entities and relationships, customize the appearance of the diagram, and generate SQL code.

Overall, ER diagrams are a valuable tool for modeling and understanding the structure of a database. They provide a visual representation of the entities and relationships within a system, helping designers and developers to identify potential issues and communicate the design of a database to others.

3.3.1 Components of the ER Model

The three main components of the ER Model are **entities**, **attributes** and **relationships:-**

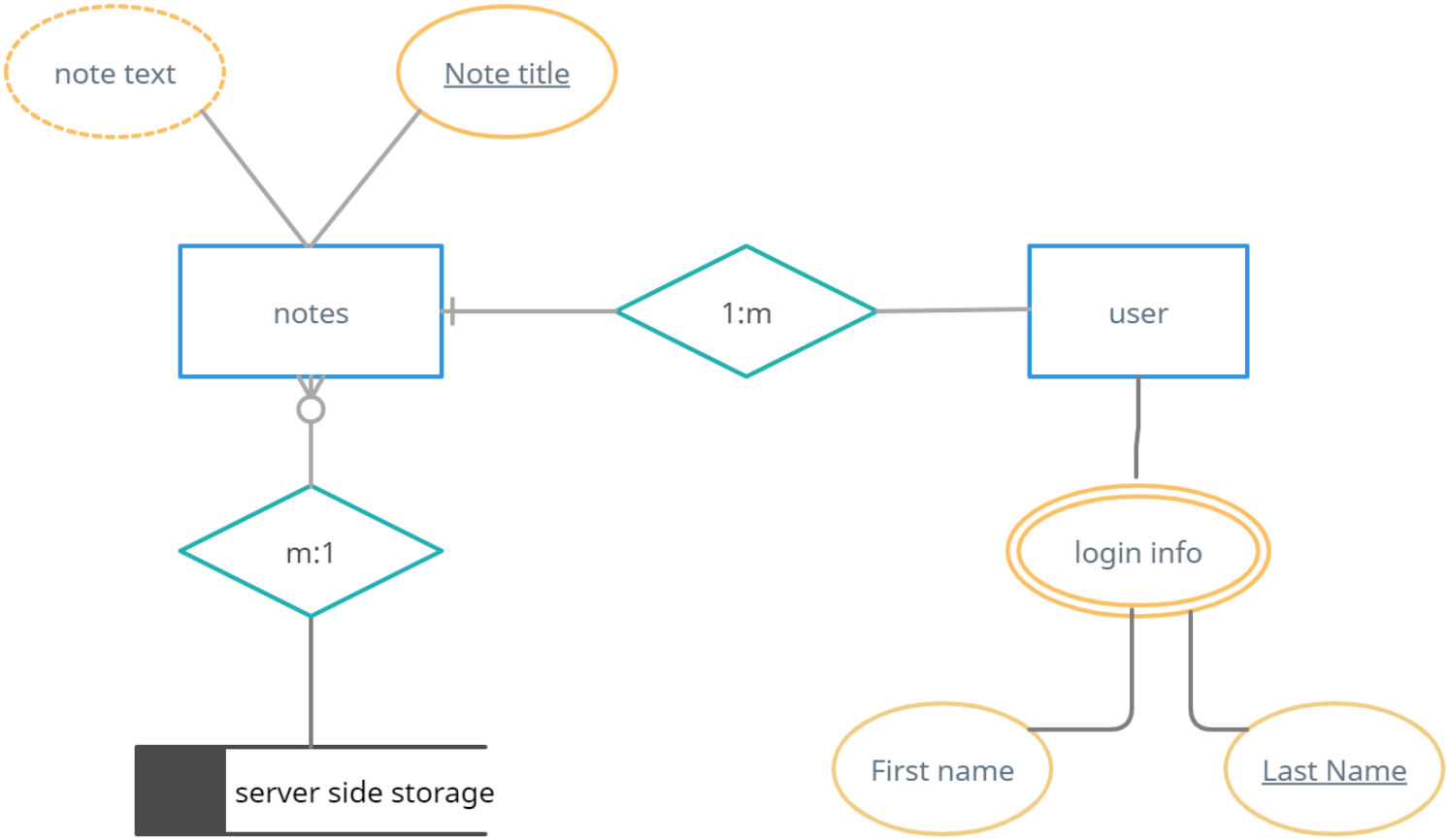
- In ERM terms, an entity is a "thing" within the organization that we want to

keep information about, such as a customer, employee or course. In other words, an entity in an ERM actually refers to a table, and rows within the table are referred to as entity occurrences. Entities are represented by rectangles containing the name of the entity. Entity names must be singular and in capital letters.

- Each entity has attributes which are the properties of each entity. Attributes will be implemented as columns in the tables. Each attribute has a domain which specifies these to possible values an attribute can have. For instance, the range of values for at telephone extension may be specific data set of integer numbers between 4000 and 4999. An attributes domain is not displayed in ER diagrams, but is recorded in the data dictionary.

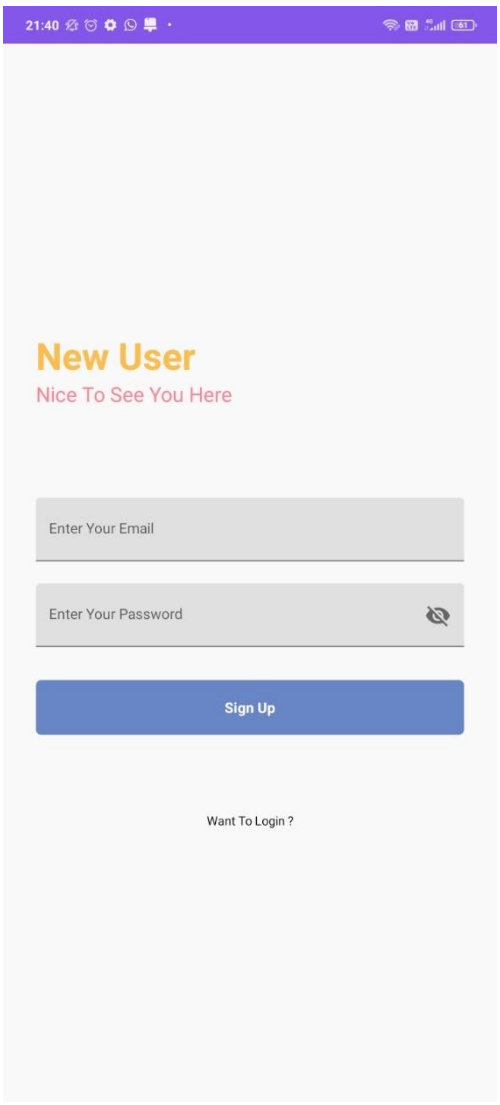
- Attributes can be of various types. A composite attribute can be subdivided into smaller parts. For example, an attribute Name can be subdivided into First Name and Last Name. Attributes that cannot be subdivided are called simple attributes. First Name and Last Name are now simple attributes. Most attributes have only a single value and as such are called single valued attributes. For example, a Teacher can have only one Last Name or a Subject can have only one Subject Code. Multivalued attributes can have more than one value. For example, a Student could have more than one Certificate or a Department may have several Extensions.
- A key attribute is an attribute that has a unique value for each entity occurrence. In other words, a key attribute is used to identify each row uniquely. For example, a Subject Code will uniquely identify each subject as not subjects can have the same Subject Code. Key attributes are represented by underlining its name.
- A relationship is the association between entities or entity occurrences.

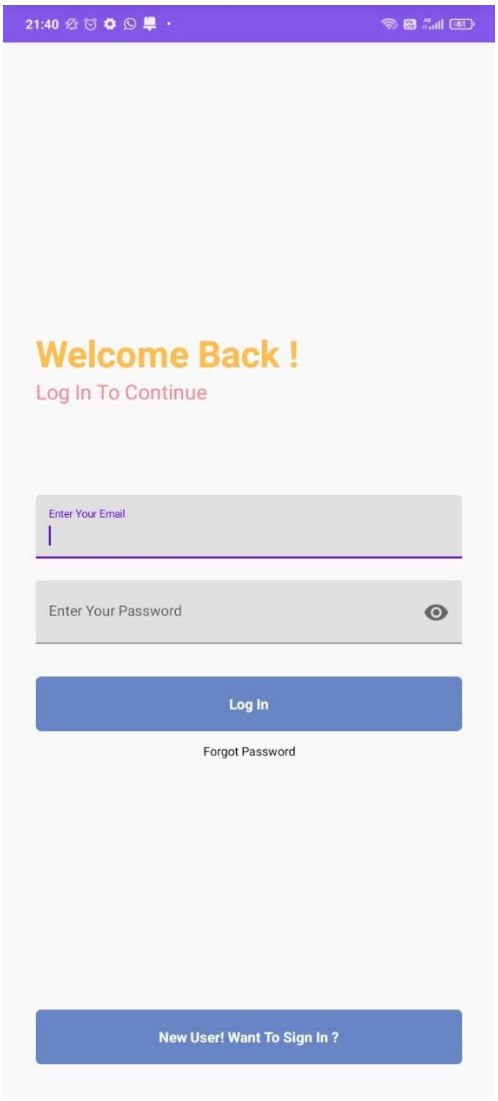
3.3.1 ER Diagram:



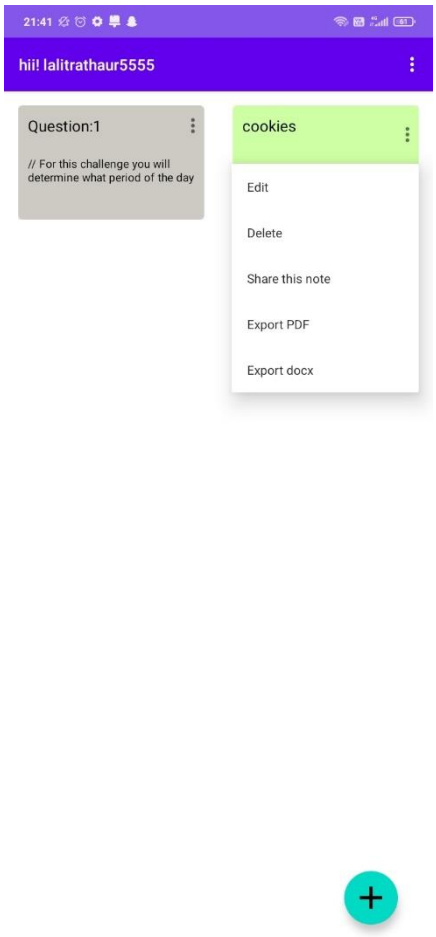
ER Diagram of the Note Nodes

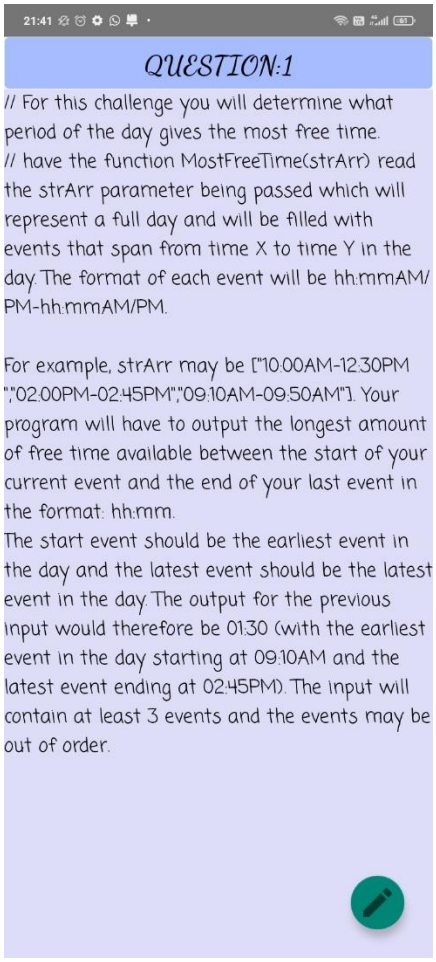
Implementation and User Interface



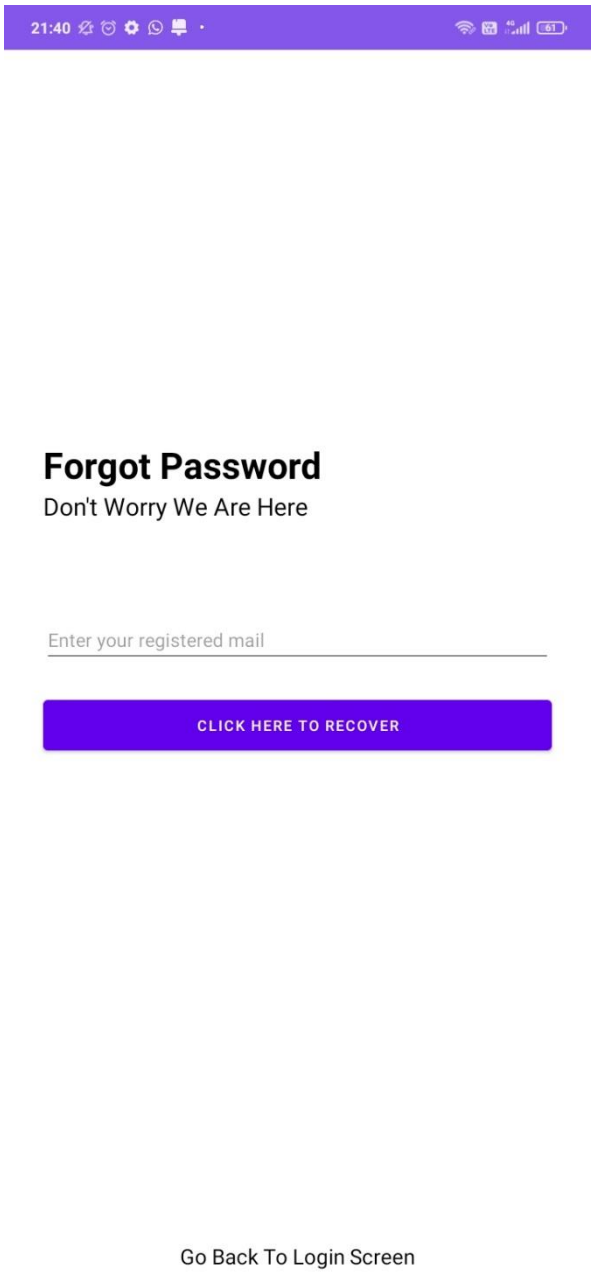


Figures showing the login page for the Existing User and New User



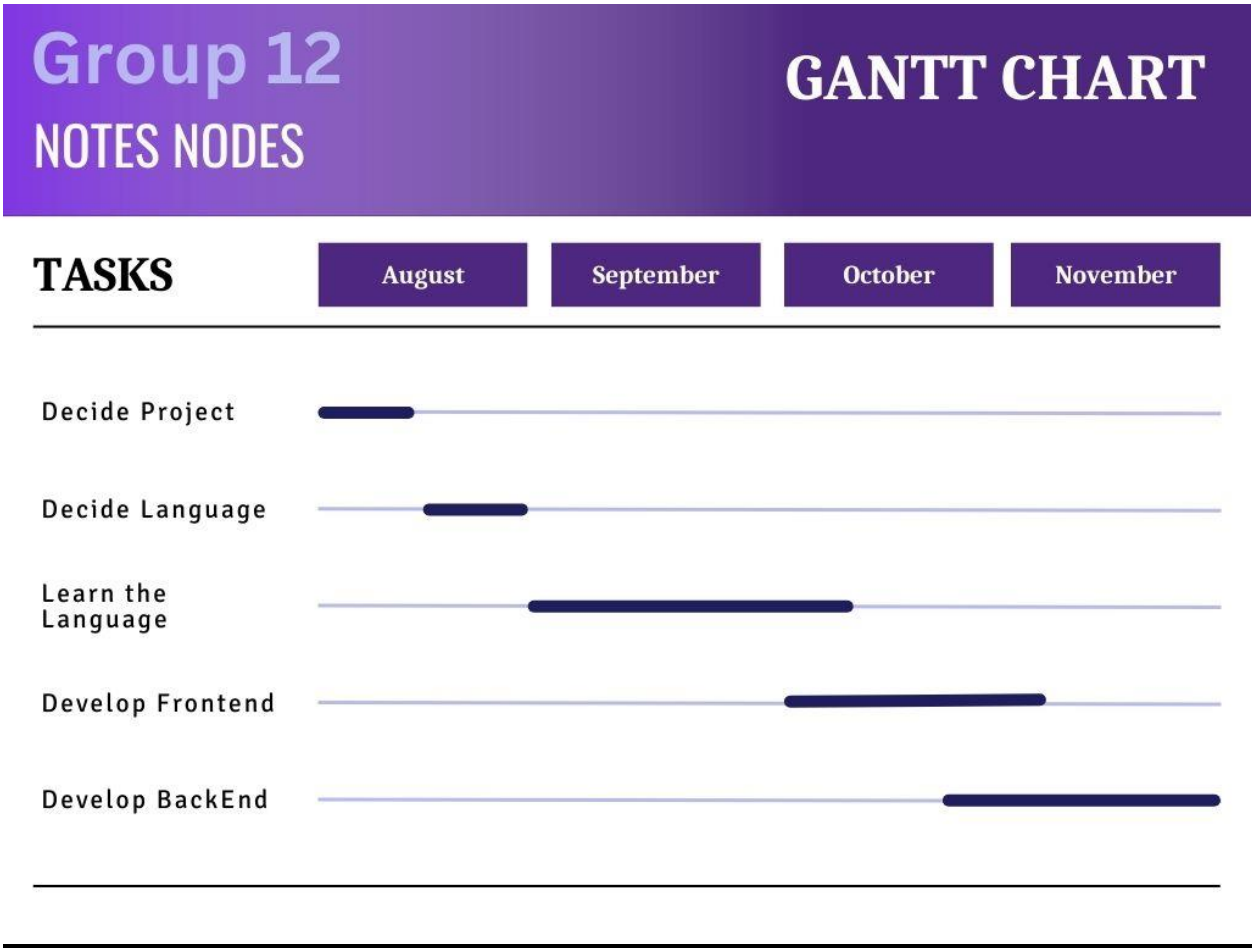


Notes user interface showing the notes and saved notes



Forget password options to recover the account

Gantt Chart


Group 12
NOTES NODES
GANTT CHART
TASKS
August
September
October
November
Decide Project
Decide Language
Learn the Language
Develop Frontend
Develop BackEnd

CONCLUSION

This application was one of the quite important app that we developed with the help of whole team,

Hoping this application if able to provide the students who are keen for their studies to make their note making capabilities much better and provide them with good platform for sharing the notes

The handy app would be able to make the studying process a bit tidy and verbose.

REFERENCES

References:

[1]. https://developer.android.com/

[2]. www.stackoverflow.com

[3]. www.javatpoint.com

[4]. Introduction to Android Development - GeeksforGeeks