SILIGURI INSTITUTE OF TECHNOLOGY

**PROJ- CS881**

**WECONNECT (Android Chat Application)**

##### BY

##### CSE\_PROJ\_2023\_04

|  |  |
| --- | --- |
| **Name of Students** | **Roll NO.** |
| 1. **RITIKA MUKHERJEE** | **11900119080** |
| 1. **AMISHA SINGH** | **11900119093** |
| 1. **ARNAB SAHA** | **11900120103** |
| 1. **SUBHANKAR SAHA** | **11900120092** |

**Under the Guidance**

**Of**

**Prof. KRITTIBAS PARAI**

Submitted to the Department of **Computer Science & Engineering** in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in **Computer Science & Engineering.**

**Year of Submission: 2023**



**Siliguri Institute of Technology**

**P.O. SUKNA, SILIGURI, DIST. DARJEELING, PIN:734009**

**Tel: (0353)2778002/04, Fax: (0353) 2778003**

**DECLARATION**

This is to certify that Report entitled **“*WECONNECT (Android Chat Application)*”** which is submitted by me in partial fulfillment of the requirement for the award of degree B.Tech. in **Computer Science Engineering** at **Siliguri Institute of Technology** under **Maulana Abul Kalam Azad University of Technology**, West Bengal. We took the help of other materials in our dissertation which have been properly acknowledged. This report has not been submitted to any other Institute for the award of any other degree.

Date:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Name of the Student | Roll No | Signature |
| 1 | Ritika Mukherjee | 11900119080 |  |
| 2 | Amisha Singh | 11900119093 |  |
| 3 | Arnab Saha | 11900120103 |  |
| 4 | SubhankarSaha | 11900120092 |  |

**CERTIFICATE**

This is to certify that the project report entitled “***WECONNECT (Android Chat Application)*”**

submitted to **Department of Computer Science & Engineering of Siliguri Institute of Technology** in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** during the academic year **2022-23,** is a bonafide record of the project work carried out by them under my guidance and supervision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Group Number: CSE\_PROJ\_2023\_04** | | | |
| **SN** | **Name of the students** | **Registration No** | **Roll No** |
| 1. | Ritika Mukherjee | 035362 | 11900119080 |
| 2. | Amisha Singh | 031023 | 11900119093 |
| 3. | Arnab Saha | 201190100120002 | 11900120103 |
| 4. | SubhankarSaha | 201190100120013 | 11900120092 |

**---------------------------------------**

**Signature of Project Guide**

**Name of the Guide:**

**------------------------------------------**

**Signature of the HOD**

**Department of Computer Science & Engineering**

### Acknowledgement

The acknowledgement page depicts the gratitude, respect and thankfulness of the student towards the people who helped him in pursuing the project successfully and ensured successful completion and implementation of the project. In this page, the author expresses his gratitude and concern by using praising and thanksgiving words.

First and foremost, we would like to thank our Project Mentor Professor Kritibas Parai , who guided us in doing these projects. He provided us with invaluable advice and helped us in difficult periods. His motivation and help contributed tremendously to the successful completion of the project.

At last but not in least, we would like to thank everyone who helped and motivated us to work on this project.

Signature of all the group members with date

1.

2.

3.

4.

***Table of Contents***

[**ARTICLE II. 1.INTRODUCTION 7**](#_1fob9te)

[**ARTICLE III. 2. SYSTEM ANALYSIS 8**](#_3znysh7)

[**Section 3.01 Identification of Need 8**](#_2et92p0)

[**Section 3.02 Preliminary Investigation 8**](#_tyjcwt)

[**Section 3.03 Feasibility Study 9**](#_3dy6vkm)

[**Section 3.04 Project Planning 9**](#_1t3h5sf)

[**Section 3.05 Project Scheduling 9**](#_4d34og8)

[**Section 3.06 Software requirement specifications (SRS) 10**](#_2s8eyo1)

1. [Functional Requirements 10](#_17dp8vu)
2. [Non-Functional Requirements 10](#_3rdcrjn)

[**Section 3.07 Data model, Control Flow diagrams & State Diagrams: 10**](#_26in1rg)

[**Section 3.08 Software Engineering Paradigm applied 11**](#_lnxbz9)

[**Section 3.09 Data model, Control Flow diagrams, State Diagrams/Sequence diagrams, ERD’s/ ClassDiagrams/CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon yourproject requirements: 14**](#_35nkun2)

[**ARTICLE IV. SYSTEM DESIGN 18**](#_1ksv4uv)

[**ARTICLE V. CODING: 21**](#_44sinio)

[**ARTICLE VI. TESTING 40**](#_2jxsxqh)

[**ARTICLE VIII. COST ESTIMATION OF THE PROJECT: 41**](#_z337ya)

[**ARTICLE IX. REPORTS: 41**](#_3j2qqm3)

[**ARTICLE X. PERT CHART, GANTT CHART: 41**](#_3j2qqm3)

[**ARTICLE XI. CONCLUSION AND RECOMMENDATIONS: 41**](#_3j2qqm3)

[**ARTICLE XII. REFERENCING AND APPENDICES: 32**](#_1y810tw)5

**Abstract:**

Time and technology improvements have necessitated faster information dissemination. Unified, customized, and intelligent information applications will become more crucial in the future. Their professional as well as personal lives. Two-way communication devices, pagers, personal organizers, screen phones, and other similar devices are examples of such gadgets. “WECONNECT (Android Chat Application)” is an application helps the students to easily access the requirements that he/she want to access. It increases our knowledge and it can be implemented in the future. This application is open for 24\*7 and when questions are asked answers will come. Anyone from our college and from anywhere can ask any questions regarding technology, science, mathematics, or prospects of it etc. This helps our generation to gather knowledge and experience and build a strong network between college peers as well. Students also can get access of their seniors working in various companies. So, they can ask for referral through this app. Thus, this app will make an impact on students’ placements also. This application is very efficient to use specially for growth-oriented people. An android-based college campus was designed to transmit the information. Updates will be provided by the application admins time to time.

**Project Deliverable's: An android chat application**

It is designed and implemented using simple and open source in demand language tools

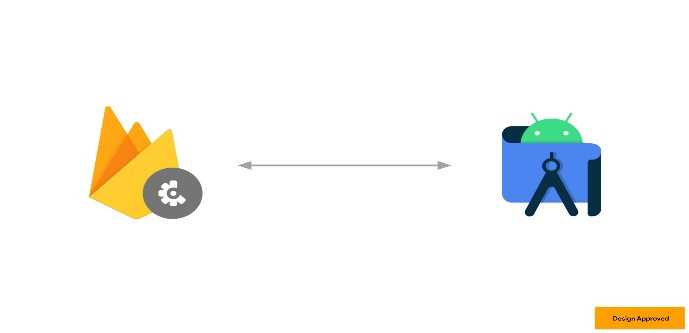
Index Terms – Android studio, Mobile application, College campus, Two-way communication, WECONNECT, Chat Application.

***Article II.***

***1.INTRODUCTION:***

**“*WECONNECT (Android Chat Application)*”** is an internet based instant messaging application which provides the user to communicate with other users in a fast and convenient way. Both the devices must have an active internet connection for the communication.

As a part of day-to-day life, academic or professional life users need to send and receive files. Using this application user can communicate with any user all over the world In this application we are using Google firebase as the backend to store the data of the application such as messages, pictures, files and more. User has to register or sign-in through their respective mail-id and can use the services. When the user sign-in to the application, user can search for another user where the communication is need to be done. User can create their profile according to which other users will be able to identify each other. This application is designed of android mobile phone users. User can respond to the messages received by just typing the reply message and press the send button.User can give their grievance and it will directly be sent to admin and strict actions should be taken.



***Materials Used:***

**Android Studio:** Built on JetBrains' IntelliJ IDEA software and designed exclusively for Android development, Android Studio is the official integrated development environment (IDE) for Google's Android operating system. In 2020, it will be offered for download or as a subscription-based service on Windows, macOS, and Linux-based operating systems. It takes the position of the Eclipse Android Development Tools (E-ADT) as the primary IDE for developing native Android apps. Android Studio is a unified development environment that allows you to create apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. You can break down your project into functional components that you can design, test, and debug independently using structured code modules.

**Devices:** A laptop, often known as a laptop computer or notebook computer, is a small, portable computer featuring a screen and an alphanumeric keypad. Although 2-in-1 PCs with a detachable keyboard are sometimes marketed as laptops or as having a laptop mode, laptops traditionally feature a clamshell form factor with the screen mounted on the inside of the higher lid and the keyboard mounted on the inside of the lower lid. Laptops are suited for mobile use because they can be folded shut for travelling. Its name comes from the fact that it was designed to be used on a person's lap in the first place. Laptops are now employed in a wide range of situations, including work, school, gaming, web browsing, personal multimedia, and ordinary home computer use.

**Gradle build tool**: Gradle is a software development tool that is noted for its versatility i.e., prebuilt in android studio. The construction of applications is automated using a build automation tool. Compiling, linking, and packaging the code are all part of the construction process. With the help of build automation tools, the process becomes more uniform. It's well-known for its ability to create automation in Java, Scala, Android, C/C++, and Groovy. Over XML, the program supports the groovy-based Domain Specific Language. Gradle is a tool that lets you create, test, and distribute applications across several platforms. The program is widely used to create software and major projects. Gradle combines Ant and Maven's advantages while minimizing their disadvantages.

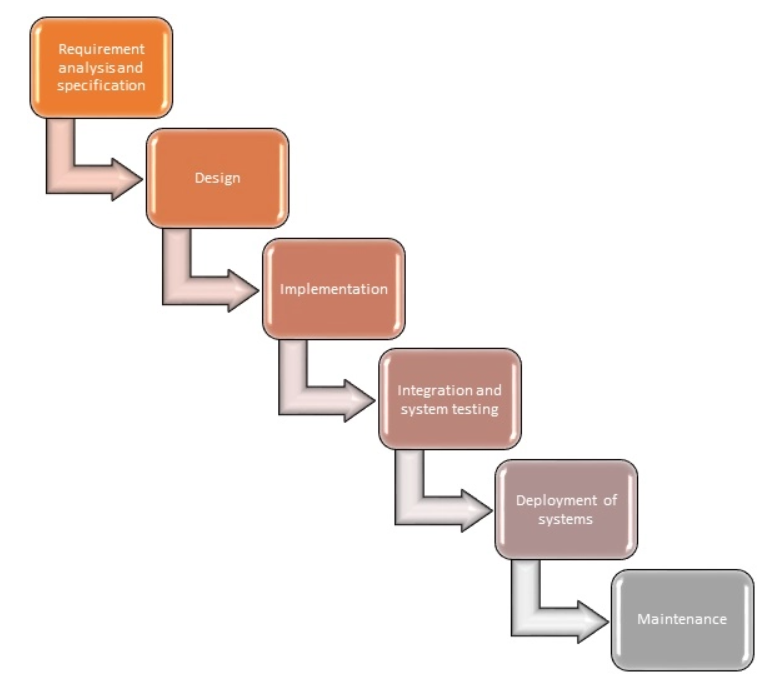
**Java:** Java is a general-purpose, object-oriented programming language based on classes that is supposed to have fewer implementation requirements. It is a computer platform for application development. As a result, Java is quick, secure, and dependable. It is widely used in laptops, data centers, game consoles, scientific supercomputers, cell phones, and other devices to construct Java applications. The Java Platform is a set of technologies that help programmers create and execute Java programming applications more effectively. It includes an execution engine, a compiler, and a library collection. It is a collection of computer software and specifications. It is a collection of software and specs for computers. Sun Microsystems developed the Java platform, which was eventually acquired by Oracle Corporation.

**Firebase:** Backend-as-a-Service provider Firebase (Baas). It offers a variety of tools and services to assist developers to create high-quality apps, expanding their user base, and making money. It is based on Google's technology. Firebase is a NoSQL database application that saves information in JSON-like documents. It also provides user authentication via different modes.**[4]**

*Article III.*

*2. System Analysis:*

### System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Online Chat Application to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing, and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.



**Identification of Need:**

##### This project is to create a chat application with a server and users to enable the users to chat with each other’s. To develop an instant messaging solution to enable users to seamlessly communicate with each other. The project should be very easy to use enabling even a novice person to use it.

**Section 3.03 Feasibility Study**

After doing the project Online Chat Application, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the

* Economic Feasibility future upcoming requirements.

This is a very important aspect to be considered while developing a project. We

decided the technology based on minimum possible cost factor.

• All hardware and software cost must be borne by the organization.

• Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later

running cost for system.

* Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

* Operational Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

##### Section 3.04 Project Planning

Our projects went through some steps and proper planning. First, we tried to understand the problem statement and checked the necessary toolkits and concepts behind it. After consuming the proper knowledge, we started developing the model and concluded.

##### Section 3.05 Project Scheduling

##### This document provides a scalable scheduling tool and associated schedule development, analysis, and monitoring methods to prepare, monitor, and report project schedules. Our Project is not that complex so we will not use very complex scheduling method.An elementary Gantt chart or Timeline chart for the development plan is given below. The plan explains the tasks versus the time (in weeks) they will take to complete.

##### Screenshot_2023-04-10-02-18-46-31_40deb401b9ffe8e1df2f1cc5ba480b12.jpg

##### [1]

##### Wi's are weeks of the months, for i=1, 2, 3, 4

##### Section 3.06 Software requirement specifications (SRS)

After the extensive analysis of the problems in the system, we are familiarized with the requirements that the current system needs. The requirement that the system needs is categorized into the functional and non- functional requirements. These requirements are listed below.

**Functional Requirements:**

This section will cover the functional requirements of the chat application.

User Registration

User must be able to register for the application through a valid phone number.On installing the application, user must be prompt be register their phone no if the user skips this step application will closed. The users phone number will be unique identifier of his/her account on chat app.

Users The Application should detect all the contact from the user phone book if any of the contacts have user account with chat application, those contacts must automatically added to the user contact list on chat application. If any of the contact have not yet register on chat app, then invitation option will be sent. Asking them to join the chat application along with the link to the

chat application. Communication

User should be able to send instant message to any contact on his/her chat application contact list. User should notify when message is delivered and display tick when message is

seen.

User should be able to create group of contacts. User should be able to broadcast messages to

these groups. Unread Message

User must be able to get information on whether the message has been read by intended recipient. if recipient reads the message.2 ticks must appear next to message read.

**Non-Functional Requirements**

These are the nonfunctional requirements of the chat application. This is basically the section that deals with the quality of the chat application rather than the functionalities of the application. User Friendly. The chat application needs to be user friendly, when using its user interface.GUI by using GUI's, it should make the application more user friendly and better to use instead of a command line. Buttons will be used.

Privacy

Messages shared between users should be encrypted to maintain privacy. Robustness

In case user device crashes, a backup of their chat history must be stored on remote database servers to enable recoverability.**[1]**

##### Section 3.07 Software Engineering Paradigm applied:

In order to implement achat application, we must consider several factors, and to evaluate and compare different approaches and techniques during all the software engineering process. So, to implement achat application we have first started to analyze several approaches both systems and tools in order to identify strong points and weaknesses. Then we have made a comparative analysis of technological specifications to choose one to base our system upon in order to standardize all the resources in our platform. Finally, we have also done an analysis of some key features of metadata tools confronting the object metadata tool we have developed on ‘WeConnect’ with some meta-data tools. We aim to give a perspective of the methodologies used for analyzing chat systems, since there are several aspects to consider.

In this type of analysis, we considered Technical evaluate chat platforms/systems: -

**Table 1.** Technical aspects to empirically evaluate Students’ chat system

|  |  |
| --- | --- |
| **Tools/Features** | **Relevance** |
| **Technical Aspects** | Considers some technical aspects that should be considered regarding the platform’s flexibility |
| Interoperability/integration | Interoperability of data and integration with other systems. |
| Standards and specifications compliance | The standards and specifications that the platform supports. |
| Extensibility | If it is possible to add new components to the platform. |
| Adaptation and Personalization | Takes care of issues regarding user personalization, adaptation, and customization |
| Interface Customization and Personalization | Possibility to customize and personalize the interface regarding the user’s taste. |
| Choose Interface | How the interface should look to be easily adaptive to anyone |
| Administrative | Takes care of issues regarding the management of the platform |
| Database Access mechanisms | Mechanisms to retrieve information from databases. |
| Produce reports | Produce statistical reports about the use of the platform. |
| Administrative workflows quality & functionalities | Mechanisms and functionalities to accelerate administrative workflows in order to get better and faster responses |
| Tracking users | Track user actions to check if they are in the right way. |
| Resources Management | Takes care of issues regarding the management of the resources. |
| File upload/download mechanisms | Mechanisms to import and export resources. |
| Communication | Takes care of the communications tools provided by the platform. |
| Grievance box | Availability of forums so students and teachers can trade experiences and discuss themes. |
| Chat | Provides a synchronous tool for students to trade experiences and discuss several topics. |
| Email& Password | Emails and passwords should be provided by each user in order to user authentication |
| Inquiries | Allow teachers to make inquiries about certain relevant matters. |
| User Detail | Complete user detail should be taken during sign up provided in the user profile in order to know about the person one is chatting with |

Analyzing several chat-applications we found that most of the systems have good administrative and communication tools, compliance with standards, high implementation level and good user interface. On the other hand, we noticed that they have problems regarding modularity management. Most of the chat application is aiming to provide all in one facility that confuses users and from that point user requires specific app for specific task. Also, the professional ones are highly chargeable in terms of bulk use and of course in terms of costs. On table 3 we resume some strong points and weaknesses that we have found.**[4]**

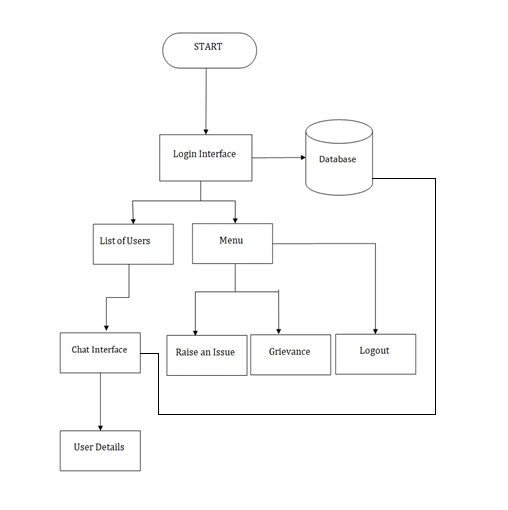
**Table 3.** Strong points and weaknesses of chat applications

|  |  |
| --- | --- |
| **Strong Points** | **Weaknesses** |
| File transfer | Confusing because of all-in-one functionality |
| Video chat, Audio chat | Not focused for any specific community |
| Money transfer | No option for report against any issue |
| Group chat | Highly chargeable for business purpose uses |

### Section 3.08 Data model, Control Flow diagrams, State Diagrams/Sequence diagrams, ERD’s/ Class Diagrams/CRC Models/Collaboration Diagrams/Use-case Diagrams/Activity Diagrams depending upon project requirements:[3]

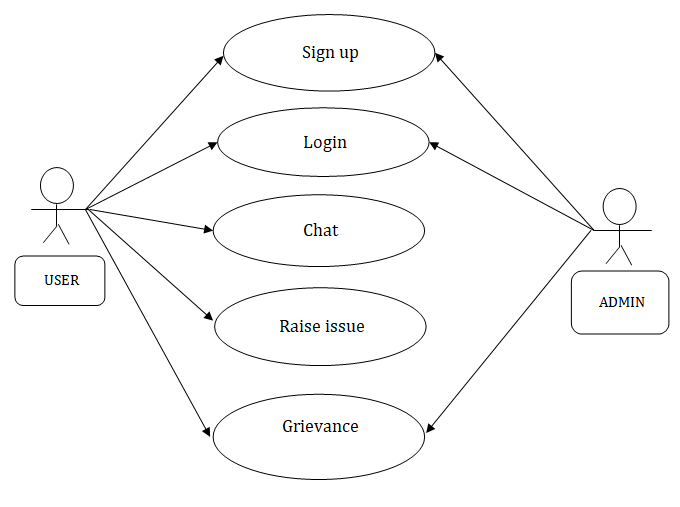
There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry. One of the major tasks here is to understand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer. This project makes a novel attempt to address the issue of program analysis and generation of diagrams, which can depict the structure of a program in a better way. Today, UML is being considered as an industrial standard for software engineering design process. It essential provides several diagramming tools that can express different aspects/ characteristics of program such as: -

##### Flow Diagram:



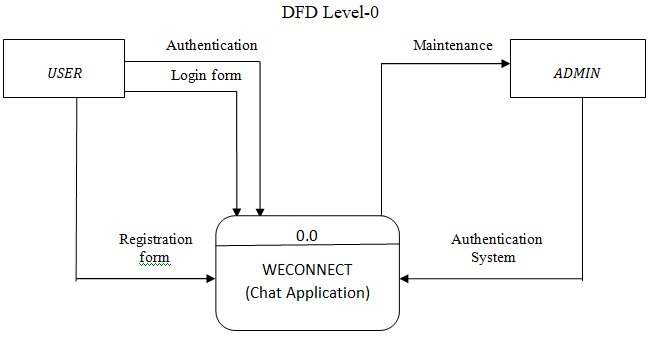
**Use Case Diagram**

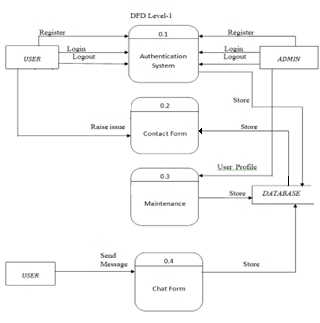
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.



**Data Flow Diagram:**

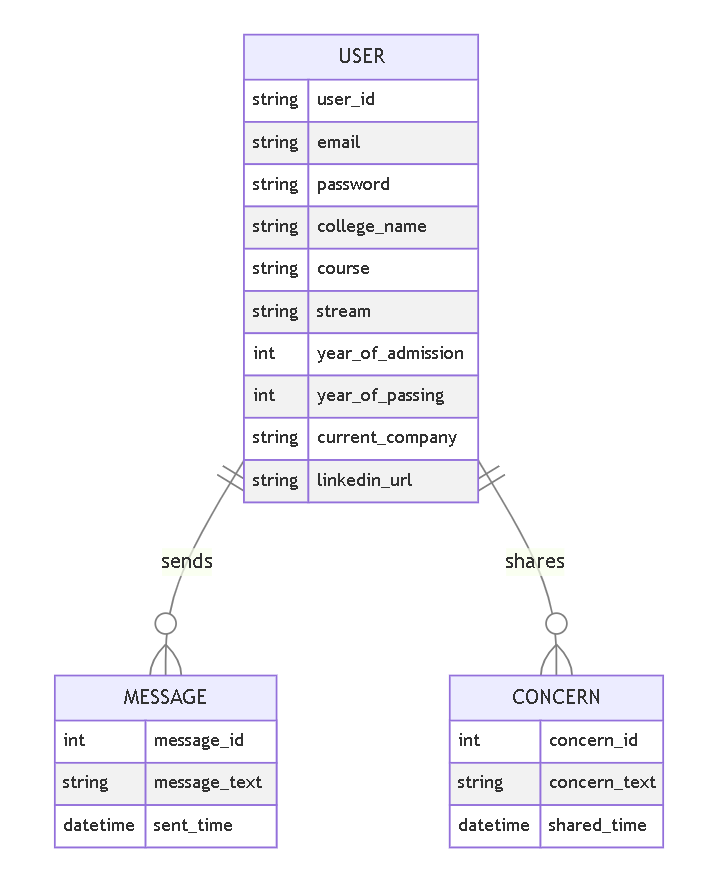
A data flow diagram (DFD) is a graphical representation of the &quot; flow&quot; of data through an information system.





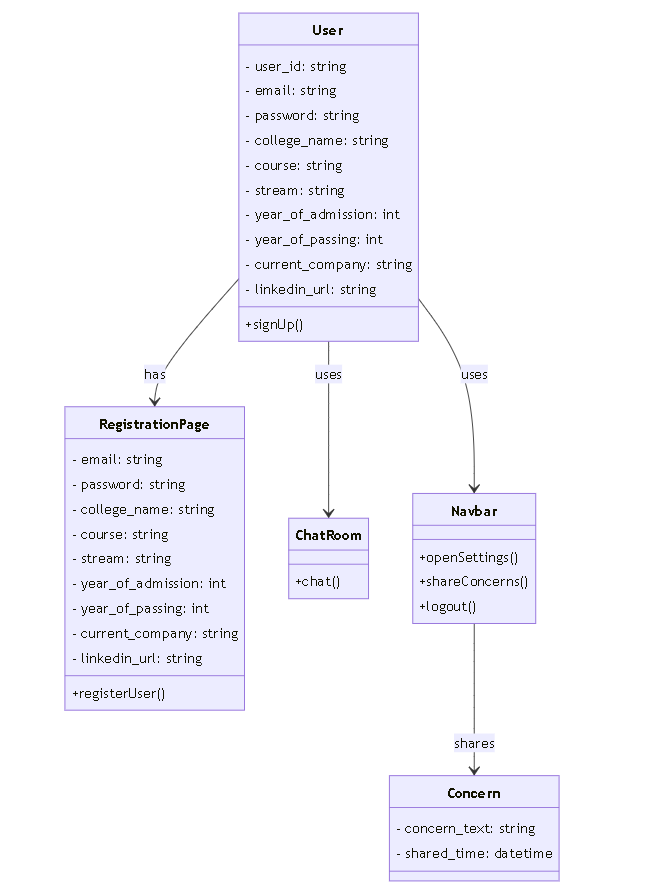
**ER Diagrams**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. A type of diagram used in data modeling for relational data bases.



**Class diagram**

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.



***Article IV.***

***System Design:***

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients’ requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. Primary Design Phase:

In this phase, the system is designed at block level. The blocks are created based on analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design Phase:

In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

* Design various blocks for overall system processes.
* Design smaller, compact, and workable modules in each block.
* Design various database structures.
* Specify details of programs to achieve desired functionality.
* Design the form of inputs, and outputs of the system.
* Perform documentation of the design.
* System reviews***.***

**Section 4.01**

**Modularization details:**

Modularization is the process of breaking down a complex software system into smaller, more manageable modules or components. In the case of a chat application like WeConnect, the following modules can be identified:

1. **Sign-up Module**: This module handles the user registration process. It collects user information such as name, email address, password, and other details required for creating a user account.

2. **Sign-in Module**: This module allows the user to sign in to their account using their email address and password.

3. **User List Module**: This module displays a list of all existing users from a specific college. It retrieves user information from the database and displays it in a user-friendly format.

4. **Chat Module**: This module allows users to send messages to each other in real-time. It uses WebSocket technology to establish a persistent connection between the client and the server.

5. **User Profile Module**: This module displays the user's profile information such as name, email address, profile picture, and other details. It allows the user to update their profile information.

6. **Grievance Module**: This module allows users to report any issues or grievances they have with the application or other users. It collects user feedback and sends it to the administrator for review.

7. **User Academic/Professional Detail Module**: This module allows users to view other user's academic details such as their department, year of study, and other information related to their academic professional career.

Each module should be designed to be independent of the others, so changes to one module do not affect the functionality of other modules. This also helps to make the application more modular and easier to maintain over time.

**Data integrity:**

User:

| **Attribute** | **Data Type** | **Description** |
| --- | --- | --- |
| UserID (PK) | Integer | Unique identifier for the user |
| Username | String | User's username |
| Email | String | User's email address |
| Password | String | User's password |
| CollegeName | String | Name of the college the user is affiliated with |
| Course | String | User's course of study |
| Stream | String | User's stream of study |
| YearOfAdmission | Integer | Year of admission to the college |
| YearOfPassing | Integer | Year of passing out from the college |
| CurrentCompany | String | User's current company (if applicable) |
| LinkedInURL | String | User's LinkedIn profile URL |

Message:

| **Attribute** | **Data Type** | **Description** |
| --- | --- | --- |
| MessageID | Integer | Unique identifier for the message |
| Content | String | Content of the message |
| Timestamp | DateTime | Timestamp indicating when the message was sent |

Concern++

Relationships:

One-to-Many relationship between User and Message:

User can send multiple Messages.

Message is sent by one User.

Many-to-One relationship between Message and User:

Message can be received by multiple Users.

User can receive multiple Messages.

One-to-Many relationship between Chat and Message:

Chat can have multiple Messages.

Message belongs to one Chat.

One-to-Many relationship between User and Contact:

User can have multiple Contacts.

Contact belongs to one User.

Note: The data model provided above represents the entities, attributes, and relationships involved in the WeConnect chat application. It can serve as a foundation for designing the database schema and implementing the necessary data storage for the application.

Data integrity is an important consideration when developing mobile applications that use cloud-based data storage systems such as Firebase. Firebase is a cloud-based platform that provides several services, including Realtime Database, Cloud Fire store, Authentication, and Storage, among others. Android Studio provides built-in support for Firebase, which makes it easy to integrate Firebase services into Android application. Here we are using Firebase authentication and Firebase real time database and Firebase storage among all the services.

Firebase provides several features that ensure data integrity, such as:

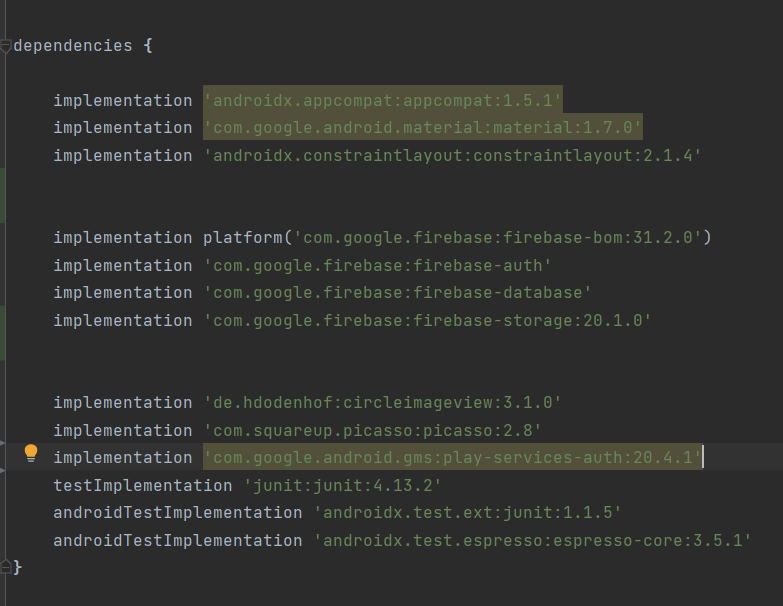
1. Authentication: Firebase provides secure authentication options that can be easily integrated into your Android application. This allows users to sign in securely and protects against unauthorized access to data.

2. Realtime Database: The Firebase Realtime Database stores data in JSON format and provides real-time synchronization and automatic conflict resolution. This ensures that data is consistent across all devices and maintains data integrity.

3. Cloud Firestore: Cloud Firestore is a NoSQL document database that provides powerful querying capabilities and real-time synchronization. It automatically handles data synchronization and conflict resolution, which helps ensure data integrity.

To ensure data integrity, it is important to design your database schema carefully and implement proper security measures. This includes defining appropriate read and write permissions, implementing proper validation checks, and using encryption to protect sensitive data. By following these best practices, you can ensure that your Android application with Firebase integration maintains data integrity.[2]

**Example:**



**[3]**

**Section 5.02**

**Database design:**

We might have come across the word "**Database**" quite often. This term carries a high emphasis on its arms. More often, it is not just related to the developer's perspective but is quite often used with non-tech groups or communities. Technically, a database is more of a storage term used to denote the relationship with different forms of data that are coagulated in a single place. Thus, we can define a database as an organized collection of data, generally stored and accessed electronically through computer systems. This article is highly centric to the database design and its association with citable terms and methodologies was commonly taken into account. We'll be discussing those terms concerning database design to understand the bits and pieces. Let's talk about it straight away.

Google Firebase is a mobile platform for helping to develop app, and increase your user base. It's made up of features that can be mixed and matched to fit the needs of the user. Most of the features on Firebase are free to use. I will be using Firebase for app sign in options and authentication and to maintain a user database.**[2]**

**Firebase Database**

Not only is the database being used to store user information but it is also being used to store user text messages and send them back to the application for all users to see. Below is an example of the message and user structure as seen on Firebase and further below that is a screenshot of the same conversation within the app. **[2]**

**Firebase Authentication**

Firebase authentication uses SDK's and ready-made UI libraries to authenticate users to my app. The authentication process supports password authentication and integrates with popular federated identity providers such as Facebook, Twitter, for this app we will only use Google or email and password authentication.**[2]**



The important consideration that can be taken into account while emphasizing the importance of database design can be explained in terms of the following points given below.

Database designs provide the blueprints of how the data is going to be stored in a system. A proper design of a database highly affects the overall performance of any application.

The designing principles defined for a database give a clear idea of the behavior of any application and how the requests are processed.

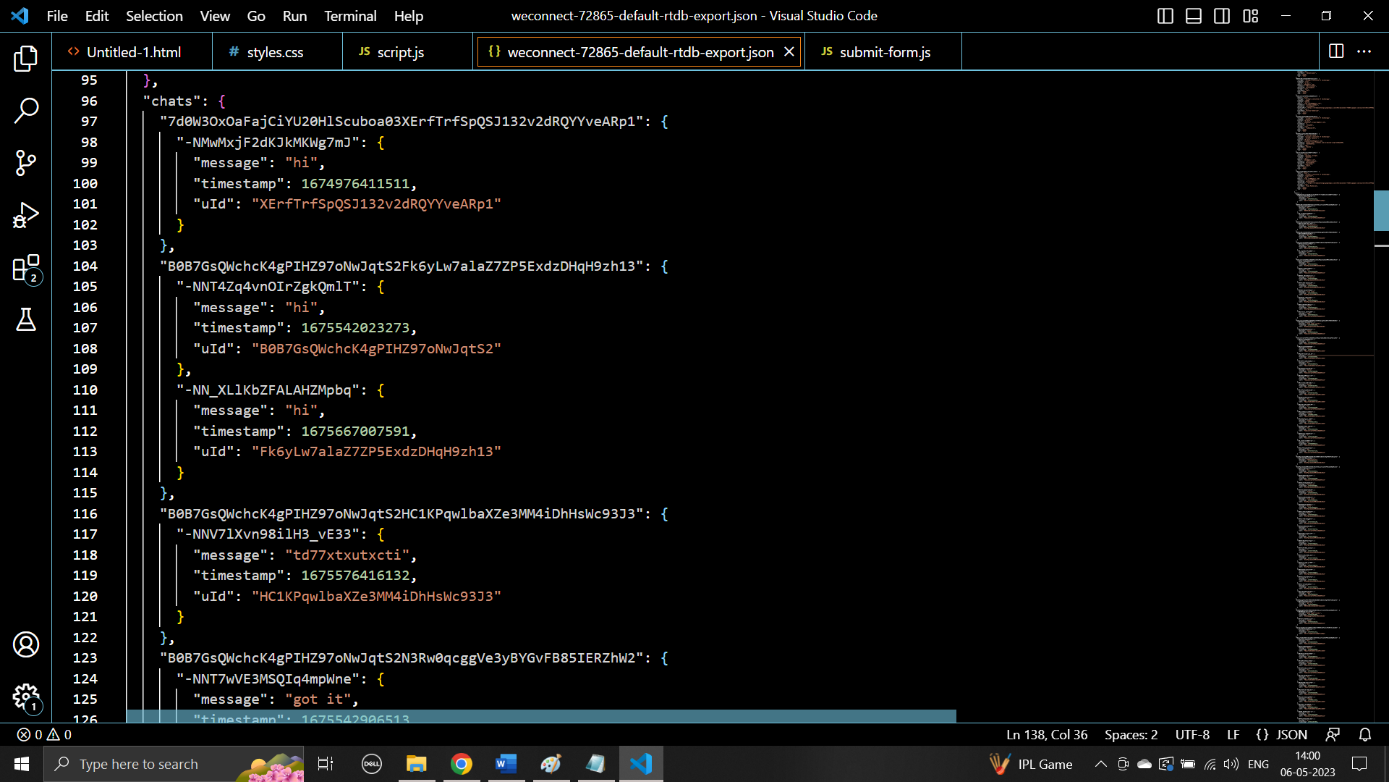
Another instance to emphasize the database design is that a proper database design meets all the requirements of users.

Data servers and push notification system is implemented using firebase. Firebase is one of the platforms which provides a real-time database and cloud services which allows the developer to make these applications with ease.

Lastly, the processing time of an application is greatly reduced if the constraints of designing a highly efficient database are properly implemented.

Here the database is created in a nosql json like format. Here two realtime table like structurs are created to support the system. One is **user** and another is **chat**.

Each entry in the user table has an autocreated unique id (userid) as a **primary key**. Each entry also consists the user detail such as user name, password, accademic details and professional details .**[2]**



On the other hand, chat table is using the user id as a foreign key so that the sender of the chat

Can be identified. Each chat stores as an entry in the chat table and has a chat id which is a concatenation of sender’s user id and receiver’s user id. Each entry also consists timestamp and message.**[2]**

**Section 5.03**

##### User Interface Design

User Interface Design is concerned with the dialogue between a user and the

computer. It is concerned with everything from starting the system or logging into the

system to the eventually presentation of desired inputs and outputs. The overall flow of

screens and messages are called a dialogue.

The following steps are various guidelines for User Interface Design:

1. The system user should always be aware of what to do next.

2. The screen should be formatted so that various types of information, instructions

and messages always appear in the same general display area.

3. Message, instructions or information should be displayed long enough to allow

the system user to read them.

4. Use display attributes sparingly.

5. Default values for fields and answers to be entered by the user should be

specified.

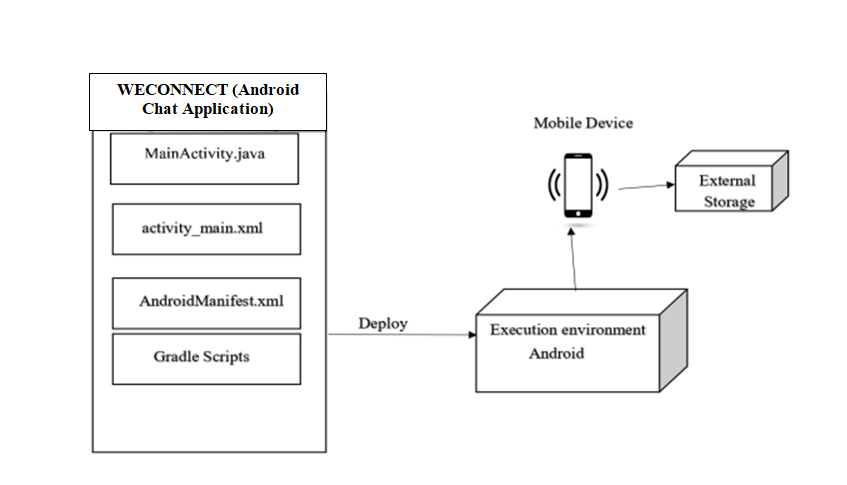
6. A user should not be allowed to proceed without correcting an error.

7. The system user should never get an operating system message or fatal error.

**Coding**

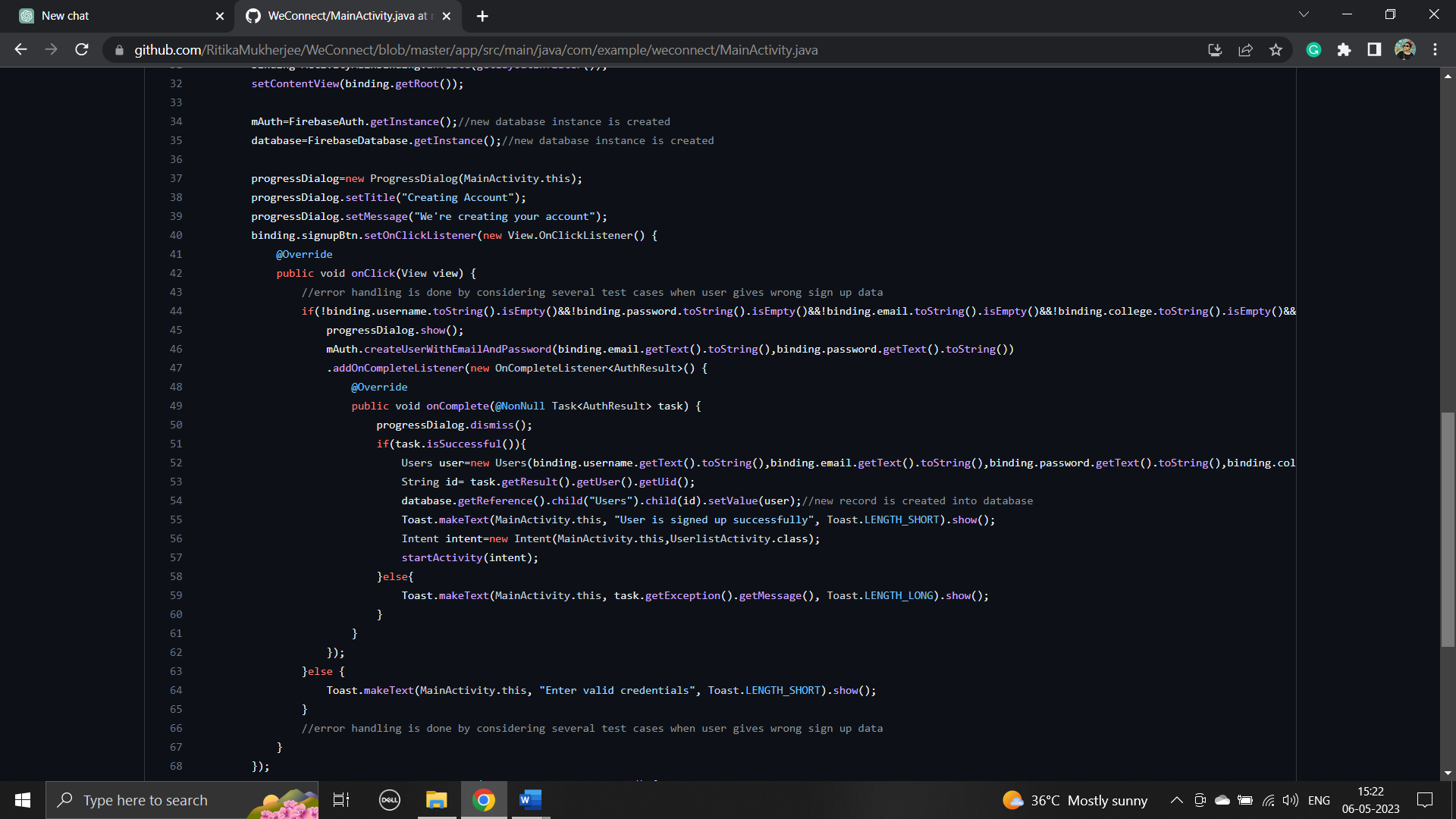
**Implementation**:

For the creation of the we connect app, the requirements consist of a laptop or a computer device that has a fast processor. The RAM must also be higher. The software that needed to be installed in Android Studio. Based on IntelliJ IDEA, Android Studio is the official Integrated Development Environment (IDE) for Android app development. Next, the project template is selected after clicking the Empty Activity and the next button is selected. Then the configuration of the project is done. The name field is changed to College Connect Development. The project’s package name is selected "com.college.weconnect". The project is saved in the desired location with the save option. Java or Kotlin has to be selected in the Language Drop–down menu. Java is selected for this project. The lower version of Android is selected to support the Minimum SDK field. The library android.support.libraries are selected for the legacy library support. The other options are the same as they are. Then Finish is clicked for the creation of the project window. After the processing time, the main window of the Android Studio will appear. The main files that have to be written are MainActivity, this is the most important activity. It's your app's starting point. The system launches an instance of this Activity and loads its layout when you develop and run your app. The next file is activity\_main.xml, this XML file specifies the user interface layout for the activity (UI). AndroidManifest.xml is the next file that is configured, the manifest file defines each of the app's components and identifies the app's basic characteristics. The Gradle scripts have to be configured, there are two files with this name: "Project: College Connect Development" for the project and "Module: College Connect Development.app" for the app module. Each module has its build.gradle file, but there is only one module in this project. To manage how the Gradle plugin creates the project, use the build.gradle file for each module. See Configure in build for further information on this file. **[1]**

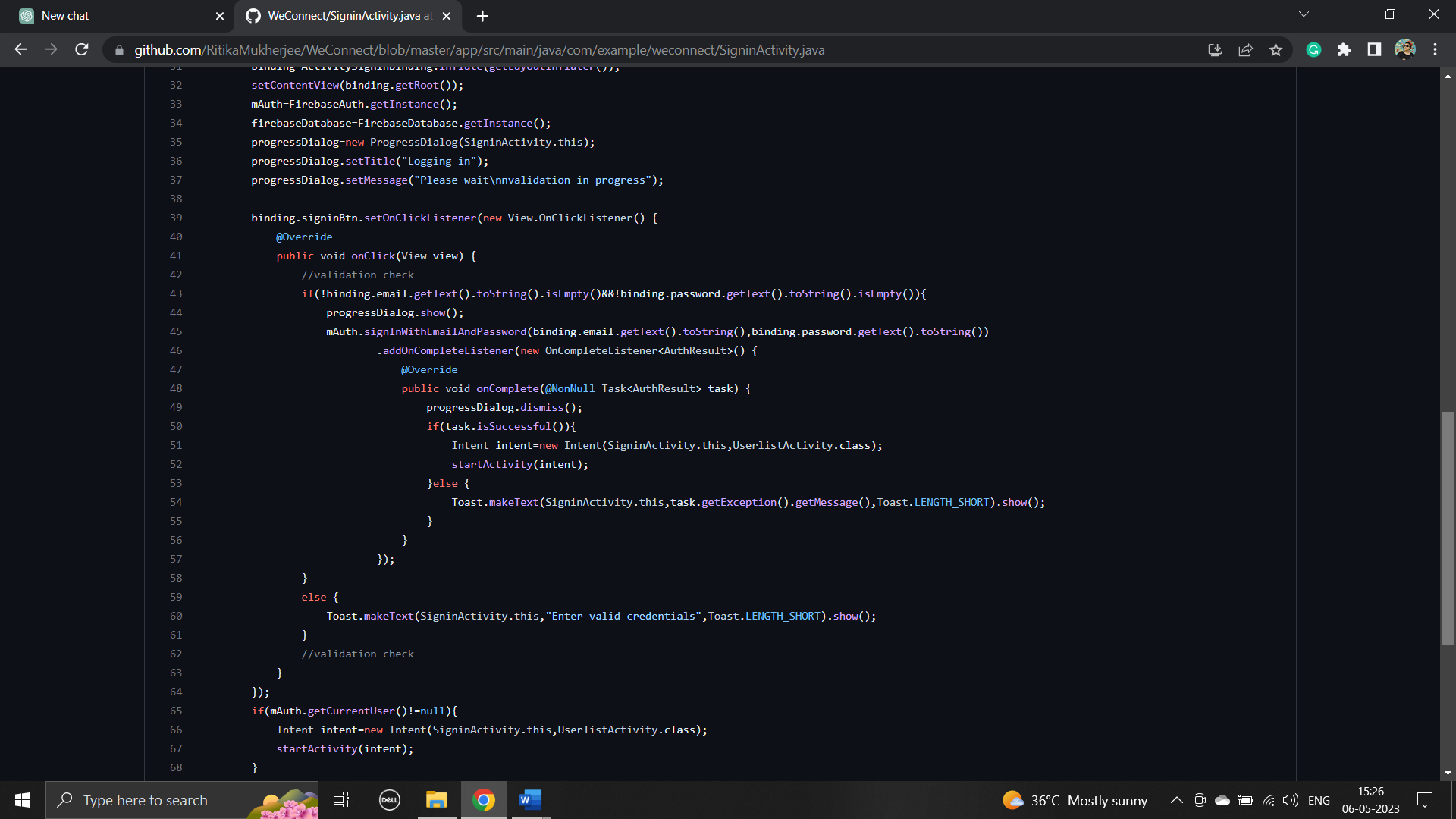
**[5]**

Block diagram of the implementation of WECONNECT (A chat application).

These are few snaps of important code snippets:

****

**Sign up code[1]**

****

**Sign in code [1]**

**Article VI.**

**Testing:**

**TEST CASES:**

In any endeavor, testing is always important before the actual implementation. SDLC is no different, in fact testing in SDLC is so important these days that developers have to work in testing longer that most stages in SDLC.

Testing is an essential part of the software development process to ensure the quality and reliability of the application. In the case of WeConnect, various testing techniques and strategies can be employed, such as unit testing, integration testing, system testing, and acceptance testing. Test case designs should be created to cover different scenarios and functionalities of the chat application. Test reports should document the results of the tests, including any issues or defects found and their resolutions.

For Testing this site we use selenium for UI Testing and Automation testing .

**Debugging and Code Improvement:** Debugging involves identifying and fixing errors or defects in the code. It is an iterative process that aims to improve the code's functionality and performance. Code improvement focuses on enhancing the code structure, readability, and efficiency. Techniques such as code reviews, code refactoring, and performance optimization is used to improve the code quality and maintainability of the WeConnect chat application.

**Section 6.02 System Security measures:**

Implementing security measures is crucial to protect the WeConnect chat application and its users' data. This includes implementing authentication mechanisms to ensure only authorized users can access the application. User profiles and access rights should be created to manage user permissions and privileges. Additionally, encryption techniques can be used to secure data transmission and storage, and measures should be taken to prevent common security vulnerabilities such as cross-site scripting (XSS) and SQL injection.

**Database/Data Security:**

To ensure database security, measures such as data encryption, access controls, and regular backups should be implemented. User authentication and authorization mechanisms should be enforced to restrict unauthorized access to sensitive data. Proper database design and implementation of security protocols help protect user data from unauthorized access and maintain data integrity.

**Creation of User Profiles and Access Rights:**

User profiles and access rights allow for proper management of user permissions and privileges within the WeConnect chat application. User profiles can store information about users, such as their username, email, and profile picture. Access rights define what actions and functionalities a user can perform within the application. For example, administrators may have additional privileges compared to regular users. Proper implementation of user profiles and access rights ensures secure and controlled access to different features of the chat application.

***Section 6.03 Cost Estimation of the Project:***

Cost estimation of the project:

Cost estimation involves determining the resources, time, and financial investment required to develop and maintain the WeConnect chat application. The COCOMO (COnstructive COst MOdel) model is a widely used cost estimation technique in software engineering. It considers factors such as project size, complexity, and development team experience to estimate the effort and cost involved. By utilizing the COCOMO model or similar estimation techniques, the project team can plan and allocate resources effectively, helping to manage the project's budget and timeline.

To calculate the cost of our project using the COCOMO model, we need to estimate the effort based on the lines of code and adjust it using the project duration and the number of people involved. The COCOMO model provides three different modes: Organic, Semi-Detached, and Embedded. For simplicity, let's assume an Organic project mode, which is suitable for smaller projects with experienced developers.

Based on the given lines of code range (2000-4000), let's take the average as 3000 lines of code.

Effort = a \* (KLOC)^(b)

a = 2.4 (constant based on project mode)

b = 1.05 (constant based on project mode)

KLOC = 3000 (thousands of lines of code)

Effort = 2.4 \* (3)^1.05

Effort = 8.269 person-months

Now, let's adjust the effort based on the project duration and the number of people involved.

Adjusted Effort = Effort / (Duration in Months)

Adjusted Effort = 8.269 / 4

Adjusted Effort = 2.067 person-months

Finally, let's calculate the cost using the adjusted effort and an average cost per person-month. For the purpose of estimation, let's assume an average cost of $5,000 per person-month.

Cost = Adjusted Effort \* Cost per person-month

Cost = 2.067 \* $5,000

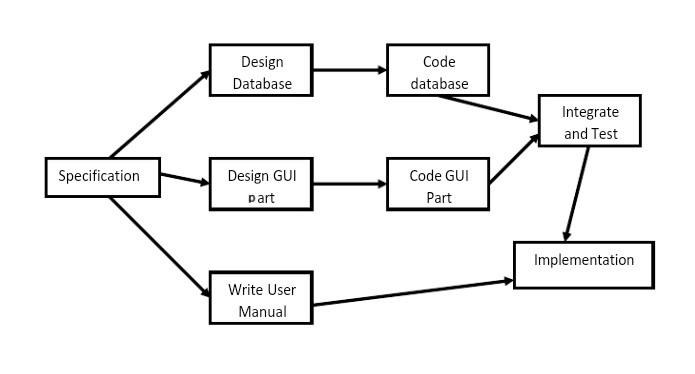
Cost = $10,335

Please note that this is a rough estimate based on the COCOMO model and the given information. The actual cost may vary depending on various factors such as project complexity, team experience, and other cost drivers.

**PERT Chart, Gantt Chart**

PERT CHART (Program Evaluation Review Technique)

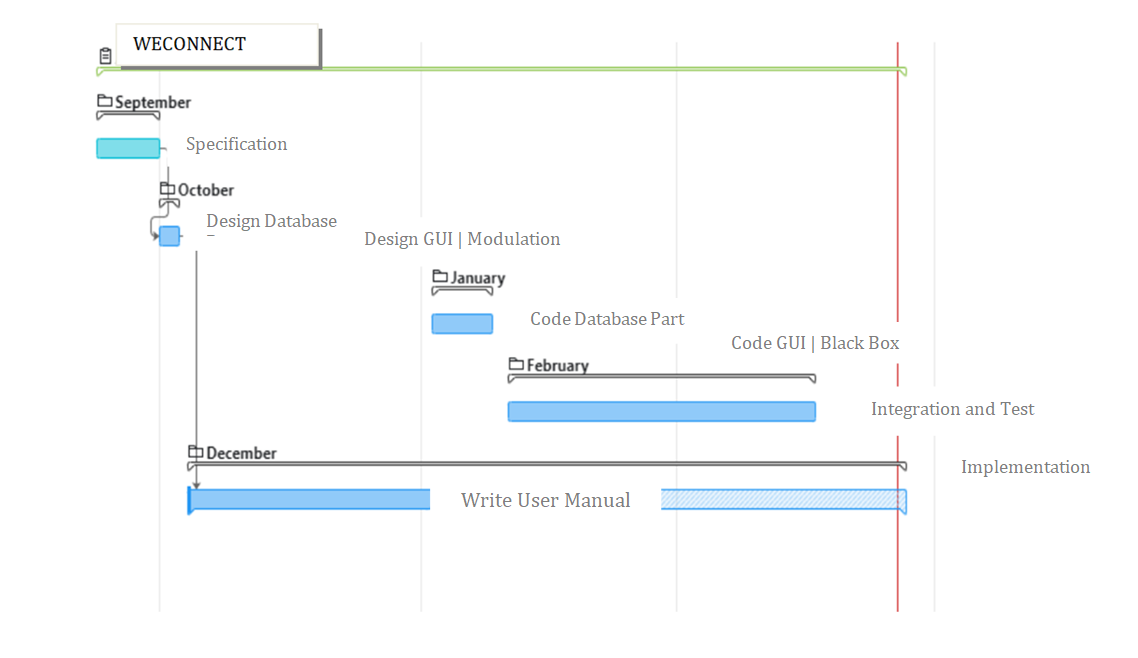
PERT chart is organized for events, activities, or tasks. It is a scheduling device that shows graphically the order of the tasks to be performed. It enables the calculation of the critical path. The time and cost associated along a path is calculated and the path requires the greatest amount of elapsed time in critical path.



PERT Chart representation

GANTT CHART

It is also known as Bar chart is used exclusively for scheduling purpose. It is a project controlling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is a bar chart with each bar representing activity. The bars are drawn against a time line. The length of time planned for the activity. The Gantt chart in the figure shows the gray parts is slack time that is the latest by which a task has been finished.



GANTT Chart representation

***REPORTS***

##### User manual- Processes:-

##### If the user wants to use the weconnect, we must download the application using the apk store, install and register it by providing login information. The login information includes user name and password. For the new user, the user must sign up by providing Full Name, Email Id and Password. Once, he registers, the registered information is stored in the server and can be validated, checking for the valid credentials for the next time he logins with the application. After using the application, he can logout as shown in below settings.

|  |  |
| --- | --- |
| welcome.png | sign.jpg |
| Weconnect logo | Sign in Features  Open chat app then click on sign in button. Then submit valid info to access your account. |
| registeration.jpg | 1st.jpg |
| Registration form for new users  Open chat app and click on sign up. Then window will pop up. Then you submit valid infos to successfully register. | Chats page, connections |
| IMG-20230327-WA0016.jpg | profile.jpg |
| Private chatting | profile |
| setting.jpg | concerns.jpg |
| settings | Concerns / grievances |

***Article VIII.***

***Conclusion and Recommendations:***

In conclusion, WECONNECT an Android Chat Applications is made up of all the most recent technologies in the way of providing a reliable system. It gives instant messaging with the user directly but virtually. So that the doubts he/she is having can be solved as soon as possible. The main objective of the project is to develop a Secure Chat Application. We have taken a wide range of literature review in order to achieve all the tasks, where we came to know about some of the products that are existing in the market. We made detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in our application. In the process of research, we came to know about the latest technologies and different algorithms. This chat application will also have authentication, integrity, and confidentiality to provide secure communication over the network. The idea is only a modest attempt to meet the educational system's requirements. There has also been a variety of user-friendly coding added. This application proves that it satisfies all the criteria. Manpower is reduced as a result of this initiative. It provides up-to-date information to instructors and students whenever they need it. As a result, a college administration system based on Android, is ideal. The required information will be available to all administrators, authorities, teachers, students, and guardians.

There is always a room for improvements in any apps. Right now, we are just dealing with text communication. There are several android apps which serve similar purpose as this project, but these apps were rather difficult to use and provide confusing interfaces. A positive first impression is essential in human relationship as well as in human computer interaction. This project hopes to develop a chat service Android app with high quality user interface.

In future we may be extended to include features such as:

1. File Transfer

2. Voice Message

3. Video Message

4. Audio Call

5. Video Call

6. Group Call

***Article IX.***

***Referencing and Appendices:***

**Video References**

**[1] Implementing sign in/signup and chat functionality**

[**https://www.youtube.com/watch?v=lJfMvLTKnwI**](https://www.youtube.com/watch?v=lJfMvLTKnwI)

**[2] Firebase Tutorial**

[**https://www.youtube.com/watch?v=DIN07bJ0pFM**](https://www.youtube.com/watch?v=DIN07bJ0pFM)

**Documentations**

**[3]Steps to Setup Firebase with Android Studio**

[**https://firebase.google.com/docs/android/setup**](https://firebase.google.com/docs/android/setup)

**[4]Guide to Write Project Report**

[**https://clickup.com/blog/project-documentation/**](https://clickup.com/blog/project-documentation/)

**Book References**

**[5]Android Application Development** by **tutorials point**

[**https://www.tutorialspoint.com/android/android\_tutorial.pdf**](https://www.tutorialspoint.com/android/android_tutorial.pdf)