Plagiarism Scan Report

Summary	
Report Genrated Date	24 Apr, 2018
Plagiarism Status	**100% Unique**
Total Words	996
Total Characters	5997
Any Ignore Url Used	

Content Checked For Plagiarism:

1. INTRODUCTION

Chat Application is medium of communication between users that utilizes computer programs that allow for two-way conversations between users in real time environment. Typically, the users of android smartphone using this app will be client and the data is passed and receive through Firebase Database will act as server if we consider client-server model. Once they are friends to each other, they can communicate with one another by typing messages in a network connected environment. The user can also see all of the messages entered by his friends. To be friends with each other one of the user should send friend request to another users and that user to accept it. Users can login, register to their account and they can also update their status.

1.1. Purpose of Study

In this project, we explore the critical role chat apps play in the distribution of digital journalism today and in the future. Mobile traffic due to its portability and simplicity of news apps, chatting apps present a good opportunity for customer development and engagement.
Communication has emerged as the new trend of social on mobile devices, and the sheer size of audiences on the social chat apps is too big to consider. These type of apps also present chance to classify differences mobile traffic sources and to minimize vulnerability should WhatsApp or other platforms decrease traffic for fresh settled tools similar to them. While new comers generally indicated optimist and excitement for their work on communicating apps, nearly all targeted out that as an industry we are still in an early, exploratory phase. Most major chatting apps expended the last few years achieving their user experience, only lately turning their attention to media-owner corporations. By that in mind, we need readers to not only learn from the case trainings obtainable, but also to initiate experiments of their own to find the right plan for any reporting team.

1.2. Problem Statement

• This project is to create the chat application with a server which will enable the users to chat with each other
• To build instant messaging solution to allow users to flawlessly connect with each other.
• To make this project easy to navigate from one activity to another activity.
• To develop timing information like when user was online.
• To ensure the security that the message only delivered to the target person.
• Encryption service for each message.
• Offline storing message for loading faster.
• Compressing the image for performance issue.

2. Hardware and software requirment

Hardware System Configuration:

Processor - Intel Core i3/i5/i7
Speed - 1.6 GHz
RAM - 4GB (min)
Hard Disk - 100 GB (min)

Software System Configuration:

Operating System - Windows / Ubuntu/Mac
Programming Language - Java, XML

Compiler for windows OS – Android Studio

Client requirement:

Any android mobile device with any version

3. Project Design

3.1. Architecture:

Fig 1 Architecture Diagram

This Messenger application would be able on a Server-Client building inside network. The Client– server design is service which allows task divides between the suppliers of source s, then send request to clients. The Server Side would be a continuously running service listening to the different Clients asking its services. The chat application would be connected on every chatting client. A Database of users would be maintained by the Server. When a client login to the application, the Server authenticates the user of the client android device. Once the user is authenticated the mail of the client is registered to the Server and it sends the list of connected user friends and other applicable information to the Client. When the user wishes to chat to some other user, his message along to name and time will send to receiver. Once are friends to each other, they can communicate with one another by typing messages in a network connected environment. The user can also see all of the messages entered by his friends. To be friends with each other one of the user should send friend request to another users and that user to accept it. Users can login, register to their account and they can also update their status.

Fig 2. Firebase as a data warehouse and process tool

3.2. Algorithm

Step 1: The app run welcome page and then go to start page.

Step 2: in start page it will display login or register buttons.

Step 3: If login selected then displays main page it will display 3 fragments(friend requests, messages, users) and menu and jump to step 5 else go to step 4.

Step 4: If register button is clicked then it will store all information to database and goes to step 3.

Step 5: In menu bar
-if “Logout” is selected it will erase the login information and goes to step 2.
-if “All users” is selected then it will display all users
-if “Account setting” Is selected then it will goes to setting page.
In Fragments
-If “Friends fragment” slides it will display the friends request and sent users
-If “Message fragment” is slides it will display the message
-If “Friends fragments” is selected it will show all online and offline friends.
if any user list is selected it will go to step 6

Step 6: In Friends profile.one user can send request, cancel request, accept request and decline request.

3.3. Data model

The Firebase Realtime Database is a cloud-based data storage and processing database. Information is put in storage as JSON and synchronized in real-time to every associated client. Once you figure cross-platform apps with our Smart Phone, and JavaScript SDKs, all of our clients share Realtime Database instance then spontaneously receive apprises with the latest data.

Data structure of this project
users
{
"Friend_Requests" : {
"sender_id" : {
"receiver_id" : {
"request_type" : "sent"
}
},
},
"Friends" : {
"user_id" : {
"friend_id" : {
"date" : "22-March-2018"
},
},
"Messages" : {
"user_id" : {
"receiver_id" : {
"-L8InfN6tb4hM_JZeFYa" : {
"message" : "jgnnqakacoaucplc0 "