Online Chatting Application

Jhalak Mittal*, ArushiGarg [x], Shivani Sharma[$]

[*,x](Department of IT, ABES Institute of Technology, Ghaziabad (U.P.), India)

Abstract: *Communication through web is turning out to be absolutely necessary nowadays. An online communication permits the clients to speak with others in a quick and advantageous manner. Thinking about this, the online communication application must be capable off the writings or pictures or some other documents in a quicker manner with least deferral or with no postponement. Firebase is one of the stages which gives a constant database and cloud administrations which permits the designer to make these applications effortlessly. Texting can be considered as a stage tomaintain communication. Android gives better stage to create different applications for texting contrasted with different stages, for example, iOS. The fundamental goal of this paper is to introduce a product application for the starting of an ongoing communication between administrators/clients. The framework created on android will empower the clients to speak with other clients through instant messages with the assistance of web. The framework requires both the gadget to be associated by means of web. This application depends on Android with the backend gave by google Firebase.*

Keywords*: Real-time database, Cloud Services, Android, Instant Messaging, Google Firebase.*

I. INTRODUCTION

In the real world the communication plays a very vital role. People have been communicating with each other through various applications or mediums. In the beginning people communicated with each other using letters or other sources, as these mediums could take much time to deliver the content. Cell phones are another medium of communication but the drawback /disadvantage is for any limited or small message which need to be passed to another user then phone call is not aperfect way. The developers/engineers then looked to implement a text-based communication which would allow an in instant communication service. In 1984, the concept of SMS was developed in the Franco German GSM cooperation by FriedhelmHillebrand and Bernard Ghillebaert. The limitation of SMS was the limited size i.e., 128 bytes [1] [2], after the accent of smartphones from 10yearsmany messaging applications have been developed. Some are Bluetooth based and some were internet based such as WhatsApp [3], WeChat [4] and others. Android is an operating system for mobiles which was developed by google. This operating system permits the applications to be utilized on mobiles. As it was developed by google, android users can create mobile applications and can be sold through android application stores such as play store. Firebase is a NoSQL database which make use of sockets which allows the users to store and retrieve the data from the database [5]. An Android version should be greater than 2.3, android studio 1.5 or higher version, and android studio project are the prerequisites to connect the firebase to an android application. Firebase provides a various kind of services such as: Firebase Authentication [6]: Firebase Authentication is useful to both developers and the users. Developing and maintaining sign-in set-up may be a bit difficult and time taking. Firebase provides an easy API [6] for sign in. It also provides the data backup using real time databases. Firebase cloud: For storing the data such as video, text, pictures building the infrastructure would be difficult and expensive for a new developer so the firebase provides the platform of cloud storage [7]. Real time database: It is a cloud facilitated NoSQL database. Aside from the authentication, cloud service and real time databases firebase also provides a service for crash reporting Crash Reporting: when some unexpected crashes occur in any applications it may be difficult to conclude why the application crashed. Firebase provides crash reporting service to deal with these crashes. This paper is concerned of a software application for the establishment of a real time communication services between operators/users. Chat application many-to-many type of communication system where the users will able to exchange the messages among themselves [8]. User can create the chatroom according to the requirement or can also join to the existing chatroom.

1.1WhatsApp

WhatsApp is one of the most popular Messaging Application, recently enabled end-to-end encryption for its 1 billion users across all platforms.WhatsApp uses part of a security protocol developed by Open Whisper System, so provides a security-verification code that can share with a contact to ensure that the conversation is encrypted [7]. It is difficult to trust in WhatsApp application completely because the application is not open source, making it difficult to verify the functioning process and match them with the work of the encryption protocol which was announced.

II. RELATED WORK

The greater part of the web open messaging, picture or document sharing applications are using logic through which the substance that has shared freely will be gathered under one reference or name. Instagram or Facebook which has a moment picture or video sharing feature uses this kind of mechanism. The algorithm for this logic is to check whether the content posted by the client contains an extraordinary character hash image

(number sign or pound sign) toward its start. The hash image is considered as a key and checks whether the name is now existing or not. In the event that truly, at that point the substance posted by the client is shared and appears at different clients when pertinent pursuit demands starts. In the event that the name is new, at that point the room is made in the database and the rest of the instrument is same as referenced.

III. OBJECTIVE

- The feature of authorization allows the user to use their account anywhere anytime with the use of any mobiles phone.
- User needs to send chat request before sending any messages to any user. When user accept their request then only they send the messages otherwise not.
- User has to login their application with their email id and password as there is no need of OTP.

IV. PROPOSED SYSTEM

4.1 Secure Mobile Chat Requirements

In this section, we propose a set of requirements to make secure chat application:

req1: Password stored on the chat server should be encrypted. req2: Providing secure session or TLS.Communicationis with the right person and no man in the middle can read the messages. req3: Messages must be encrypted to maintain security and privacy in this application

req4: Local storage must be protected by encryption.

req5: Messages are not stored on the chat server but stored on the user's device.

req6: It is not allowed to exchange messages if they are not friends.

4.2 Proposed Architecture

The proposed engineering is intended to be Client-Server visit application. In customer side, when a client sets up the application, the client either chooses registration or sign in. In server side, the visit server comprises of clients' server and a message server. Client's server that deals with client's certifications. Message server handles messages between clients by utilizing Firebase Cloud Messaging (FCM).If the beneficiary is disconnected, the messages will be put away incidentally on the FCM line for a particular timeframe, and when beneficiary becomes online these messages are sent to him at that point erased from the line. The generic architecture of proposed chatsappeared in fig1.

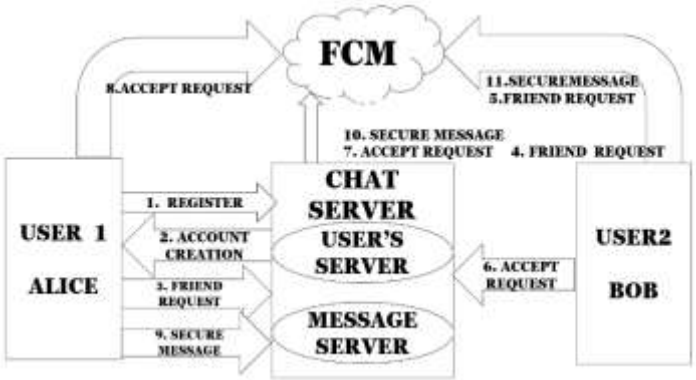

Fig 1: GENERIC ARCHITECTURE OF PROPOSED SYSTEM

4.3 Registration an account

Each account has just a single device and it is recognized by device id. Furthermore, Email and username are one of a kind. Name, email and password are required to enlist new account.Aftertyping the registration information,the password is encoded by utilizing XSalsa20 calculation[15]then the client accreditations are sent to the server. After check, the server produces one kind of a identifier that goes about as the client ID. From that point forward, the affirmation message is gotten for fruitful enlistment to the customer application and the customer data is stored in local storage.The Application produces a lot of keys:

(a) Key for encrypting password.

(b) An open key pair for calculating meeting key.

(c) Symmetric storage key for encrypting/decrypting local storage contains contact list and chathistory.

4.4 Login

Email and password are required for client validation as appeared in fig2.After typing the verification data, the secret phrase is encoded then the client qualifications are sent to the server. The server browses if the email and password are valid. After approval, JSON Web Token (JWT)[16] is made and sends to the client to store it. At

the point when a client makes a request at the later time, JWT is passed with the requests. The server confirms of the JWT, in the event that it is legitimate, the request is processed.

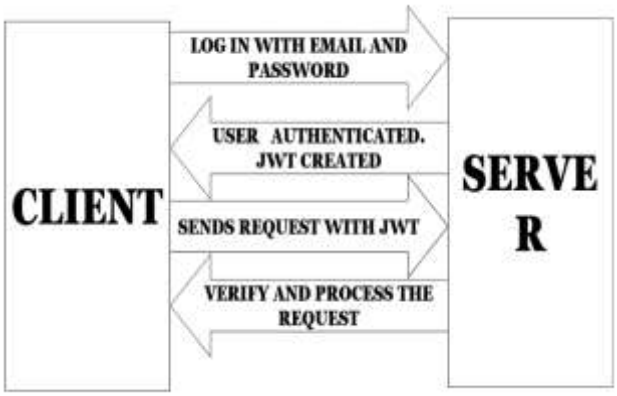

Fig 2: LOGIN PROCESS

4.5 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) is a service that encourages informing between mobile applications and server applications. It's based on Google Play Services that supports cross-stage (iOS, Android and Web). It is a free assistance that permits sending lightweight messages from the server to the devices at whatever point there is new information available[17]. This spares a great deal of client's battery by abstaining from requesting to the server for new messages.Towards the start of running the application just because gets the accompanying: The application interfaces with FCM server and registers itself. (2) When fruitful registration FCM gives enlistment token to the device and token particularly recognizes every device. (3) The application sends the registration token to the server to store it in MongoDB database.

The above steps are appeared in Fig. 3. Firebase Cloud Messaging. At the point when the server sends a pop-up message, it sends a request to FCM sending the push message alongside the registration token. FCM distinguishes the objective device by utilizing registration token at that point begins to push information.

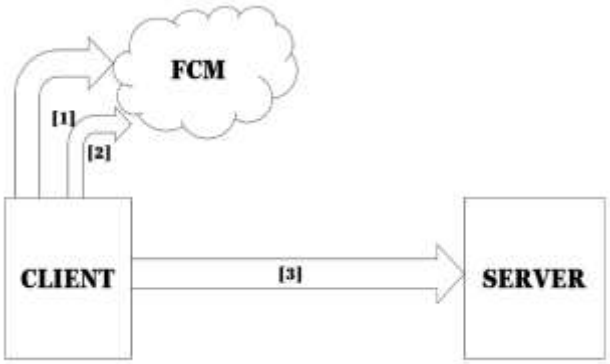

Fig 3: FIRE BASED CLOUD MESSAGING

4.6 Session key Setup

To add clients to contact list either by username or by email address. For sending a request to a companion on the assumption that the principal client knows the username or email of the second client due to the username and email are unique for every client and the subsequent client ought to have already registered in the server. Probably, the primary client is called Alice and the second is called Bob. When the send demand, Bob name is composed by Alice and her open key is brought from the local storage then the request is sent to the server. At the point when a request is gotten, it shows up as a Notification(Fig. 4). In the event that the companionship demand is acknowledged by Bob, his private key is brought with Alice's open key to figure the meeting key by utilizing Elliptic Curve Diffie-Hellman (ECDH) over the bend Curve25519 [18] and hashes the outcome with HSalsa20 [15] then the meeting key is put away in IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.11, November 2017 111 neighbourhood stockpiling (Fig. 5). At last, the acknowledgment is sent with his open key to the server to be conveyed to Alice. Endless supply of the acknowledgment of the solicitation, similar strides on the above are taken. The meeting key is determined by utilizing Alice private key and Bob open key then it is put away in the local storage for sometimes in the future. The meeting key is the equivalent for the two gatherings and this is the quality of the Elliptic Curve Diffie-Hellman (ECDH) and along these lines it is hard to attack by the man-in-the-centre.

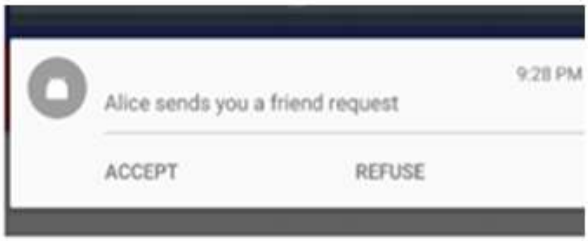

Fig 4: Friend-request notification.

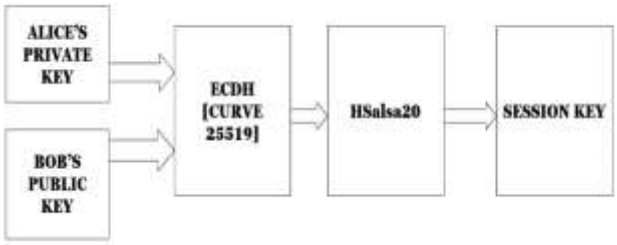

Fig.5: SESSION KEY SETUP

4.7 Exchanging Messages

At the point when a message is composed, the application encrypts the message utilizing XSalsa20 encryption calculation to encode the message body and Poly1305 to process a Message Authentication Code (MAC) [19]. Each message has its own different key and nonce which brings better security for each single message in such finding one of the keys can't decode past messages. Subsequent to encrypt the message, it is encoded again utilizing the beneficiary's meeting key then it is sent to the server (Fig. 6). After the message is gotten from FCM, the MAC of the encrypt message is determined and contrasts it and got MAC to confirm the honesty of the message. In the event that the outcomes are not the equivalent, it is dismissed and doesn't show to the client else it is decrypt by the sender meeting key. Next, the message body is confirmed in similar strides above. Presently the key and nonce to decode the message are known. The message is then decrypt and put away in the local storage and showed to the beneficiary. in the event that the beneficiary uses the application it will be shown in the visit window in Fig. 6 that is a procedure to encrypt a message.

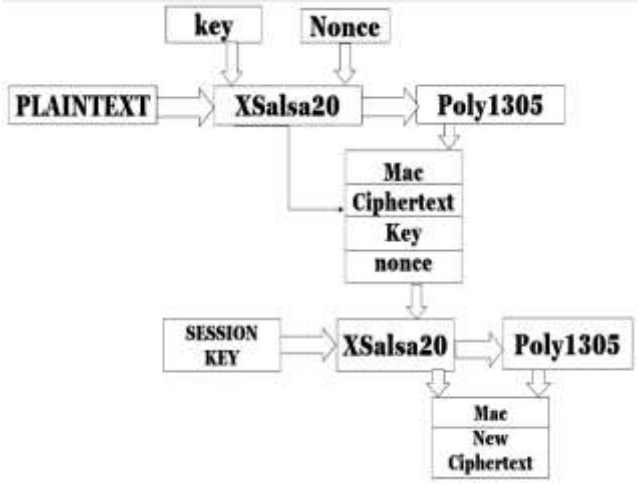

Fig. 6 PROCEDURE TO ENCRYPT A MESSAGE

4.8 Local Storage

The information is put away locally in the application by utilizing Realm database. Domain is a lightweight portable database that supports cross-stage. It's anything but difficult to utilize and quick. More, it has loads of present highlights, for example, JavaScript Object Notation (JSON) support, a familiar API, information change notices and encryption support [20]. Encoded information is shielded from unapproved get to and is available just if have been a correct encryption key. Domain utilizes AES-256+SHA2 calculation and 64-byte key for scrambling stockpiling [21]. To plan Realm stockpiling goes through a few stages that are: Step 1: The application checks whether the lock screen is available or not. In the event that it exists, the accompanying advances are finished. Stage 2: Generate Realm Key that is utilized for encoding stockpiling. Stage 3: Generate key from Keystore. Stage 4: Realm key is scrambled with the key created in stage 3 by utilizing AES in CBC

mode. Stage 5: Save the scrambled key in shared inclinations in private mode with the goal that different applications can't get to this information index. Three documents are put away in the nearby stockpiling. UserInfomation document that stores all data relating to the client. While Friends document stores all data relating to the companions. At last, Messages document stores all data relating to messages.

4.9 Server-Side Implementation

Server-side has depended on Node JS[22] and MongoDB database[23]. Hub JS is quick, fit for taking care of an enormous number of synchronous associations with high throughput, which is comparable to high versatility. MongoDB gives TLS that makes a safe association (Fig. 7). To play out a customer demand goes through a few stages that are:

Stage 1: Initially, must run the MongoDB association at that point run the Node JS from Command Prompt. At this stage, the server is prepared to get the customer's solicitation.

Stage 2: When the customer sends a solicitation, the server gets the HTTP demand in JSON group. The solicitation at that point parsed. Stage 3: The HTTP demand is contrasted and the base way on the off chance that it is coordinated, it is given to Express system. Stage 4: The Express gets the HTTP solicitation and courses it to the particular endpoint that coordinated it. If there should be an occurrence of not coordinated with any of the courses will show mistake in Command Prompt. Else, it will be sent to the controller which handles the necessary capacity. Stage 5: Make a solicitation to MongoDB database by mongoose for preparing capacity. Stage 6: When the information is brought from MongoDB database and the necessary tasks are done, Node JS gets the reaction at that point sends to the customer(fig8). implementation of client request.

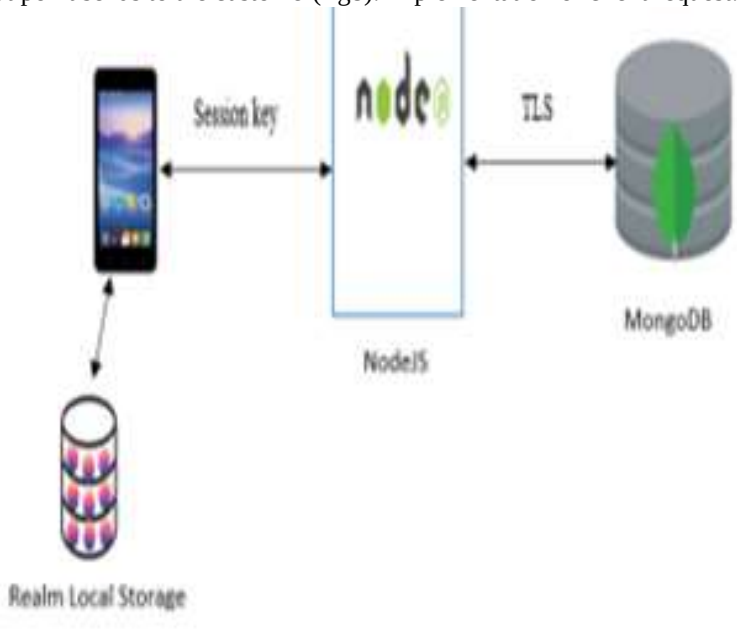

Fig 7. THE SPECIFIC ARCHITECTURE OF PROPOSED CHAT

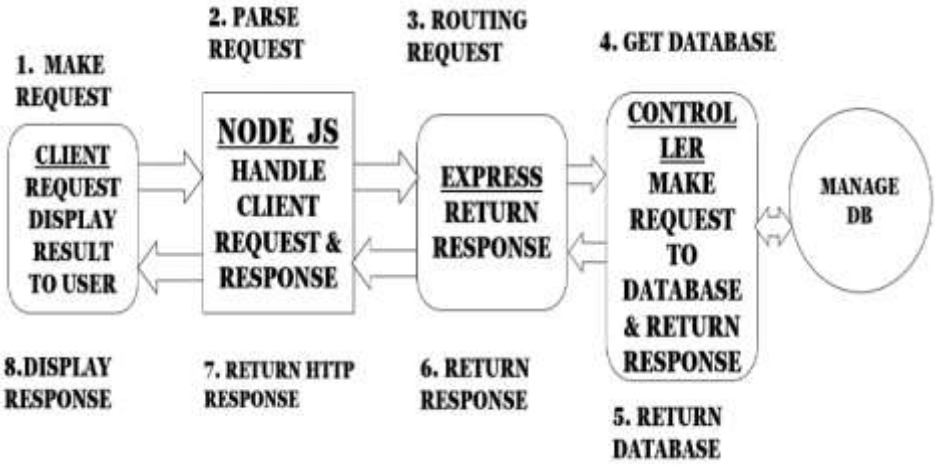

Fig 8. IMPLEMENTATION OF A CLIENT REQUEST

V. IMPLEMENTATION AND RESULTS

1. Registration
   1.1 Enter your mail id.
   1.2 Choose password.
2. Login to Your Account
   2.1 Enter your mail id.
   2.2 Enter your password.
   2.3 If valid user move step 3.
   2.4 if not a valid user go-to step1.
3. Find Friends to Start Chatting
   3.1 Send chat request.
4. Requests
   4.1 If requests come from known user then accept.
   4.2 Else reject.
5. Contacts
   5.1 Easily check your friends in contact section.
6. Account Settings
   6.1 Set display pic.
   6.2 Set status.
7. Chatting
   7.1 Easily chat with friends and check their messages too.
8. Groups
   8.1 Create a group.
   8.2 Check the group joined by you.
8.3 Chat in groups to enjoy chatting with more than one people at a same time.
9. Logout
   9.1 logout the app.

The steps appeared in below fig9.Chat room work flow chart.

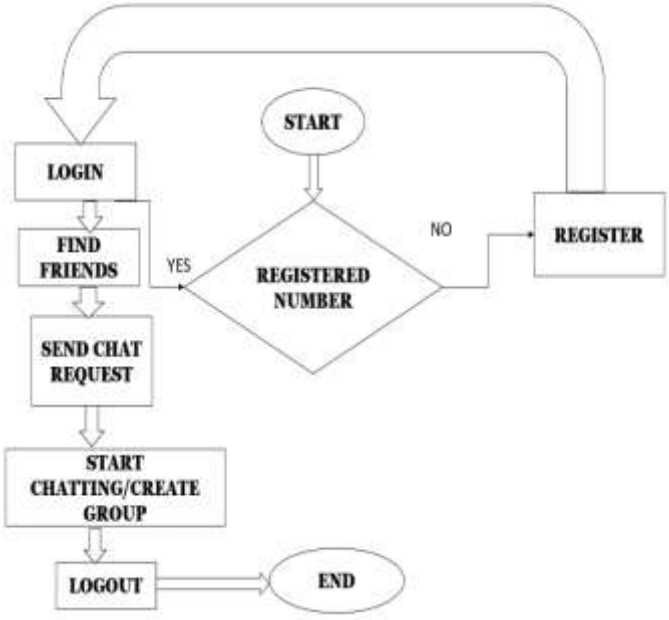

Fig 9. Chat room work flow chart

RESULT

The last framework will result as a real time application which gives the clients to impart to one another with an ease. The application will have a login page through which the client can register and login themselves.Clients can send and get instant messages.

Here are some differences to notice between our application and WhatsApp :-

The feature of authorization allows the user to use their account anywhere anytime with the use of any mobiles phone.
User needs to send chat request before sending any messages to any user. When user accept their request then only they send the messages otherwise not.
User has to login their application with their email id and password as there is no need of OTP.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we presented a detail for saving the security and protection of the talk application. We portrayed a lot of prerequisites for making secure talk and actualize it by utilizing present day strategies and lightweight for giving velocity and great assurance to its customers. XSalsa20 calculation perfect for cell phones in view of its high security, superior and keeps up battery life. Customers can be sure that no one can peruse their messages, regardless of whether the cell phone arrives at wrong hands can't enter to the application and can't get to the information put away locally.
There is in every case some spot for improvements in any product application, anyway great and proficient the application might be. At this moment, we are managing just the texting between the friends. In future the application may add more features to incorporate a few highlights, for example,
1. Voice informing.
2. Gathering calling
3. Live spilling
4. Messages auto erase after a given time.
5. Customized message tunes.
Also, an informing application highlight which permits the client to make talk room while in discussion with another client by simply sending the chatroom name with the hash
Symbol at the beginning.

VII. REFERENCES

1. Anon., 2015. Development of a Health Care Assistant App for the Seniors. International Journal of Applied Science and Engineering, pp. 3-5.
2. Jianye Liu; Jiankun Yu, Research on Development of Android Applications, 4th International Conference on Intelligent Networks and Intelligent Systems, 15 December 2011
3. AbhinavKathuria et al, Challenges in Android Application Development: A Case Study, Vol.4 Issue.5, May- 2015, pg. 294-299
4. Li Ma et al, Research and Development of Mobile Application for Android Platform, International Journal of Multimedia and Ubiquitous Engineering 9(4):187-198 • April 2014
5. Nikhil M. Dongre, Nikhil M. Dongre, Journal of Computer Engineering (IOSR-JCE), Volume 19, Issue 2, Ver. I (Mar.-Apr. 2017), PP 65-77
6. Javed Ahmad Shaheen et al, Android OS with its Architecture and Android Application with Dalvik Virtual Machine Review, International Journal of Multimedia and Ubiquitous Engineering Vol. 12, No. 7 (2017), pp. 19-30
7. SajidNabi Khan, IkhlaqUlFirdous, Review on Android App Security, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 4, April 2017
8. LazarelaLazareska, KireJakimoski et al, Analysis of the Advantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa, American Journal of Software Engineering and Applications 2017; 6(5): 116-120
9. Bin Peng et al, The Android Application Development College Challenge, 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, 18 October 2012
10. Shao Guo-Hong, Application Development Research Based on Android Platform,2014 7th International Conference on Intelligent Computation Technology and Automation, 08 January 2015
11. S Karthick, Android security issues and solutions, 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 13 July 2017
12. PravinAuti, SangamMahale, VikramZanjad, MadhuriDangat, n.d. An Android Based Global Chat Application. 4(1), pp. 1-2.
13. PravinAuti, SangamMahale, VikramZanjad, MadhuriDangat, n.d. An Android Based Global Chat Application. 4(1).
14. S, A. K., n.d. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase. s.l.: s.n.
15 D. J. Bernstein, "Extending the Salsa20 nonce," no.Mc 152, pp. 1–14, 2011.
16 M. B. Jones, "The Emerging JSON-Based Identity Protocol Suite," 2011.
17 "Firebase Cloud Messaging | Firebase."[Online]. Available: https://firebase.google.com/docs/cloud-messaging/.
18 D. J. Bernstein, "Curve25519 : new Diffie-Hellman speed records," vol. 25519, 2006.
19 D. J. Bernstein., "Poly1305." [Online]. Available: https://en.wikipedia.org/wiki/Poly1305.
20 "Realm: Create reactive mobile apps in a fraction of the time." [Online]. Available: https://realm.io/.
21"Realm Swift 2.10.2."[Online]. Available: https://realm.io/docs/swift/latest/.
22 "Node.js."[Online]. Available: https://nodejs.org/en/.
23 "NoSQL Databases Explained | MongoDB." [Online]. Available: https://www.mongodb.com/nosql-explained.
View