Android Chat Application Development using AWS

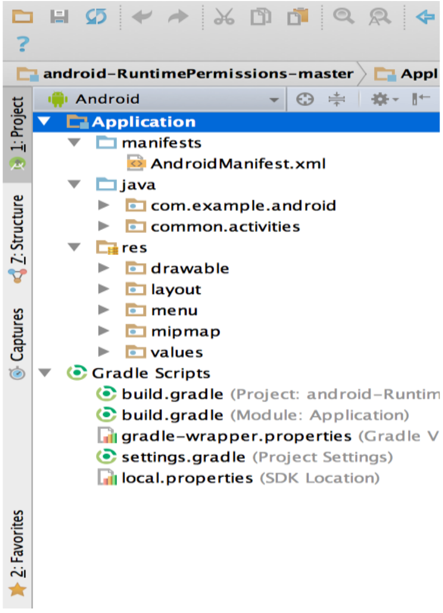
Sujatha Kumari B A, Chiranth N L, Pooja P

Abstract: ***Online chatting is a kind of communication that is done over the Internet, which uses an online chat application to it. The online chatting offers the users to experience real-time conversation with the people who are located geographically in a different location that is it offers real-time transmission of messages from one person to other. Chat messages are usually short so that the other participant responds to them quickly. Thereby, it makes the users of the online chatting app, feel the chat like a real-time conversation. The online chat application allows the user to address the single message to an individual receiver or multiple receivers that is by using the point to point communication and multicast communication techniques. All these parameters must be considered while designing a chat application along with these parameters the other important parameter that should be considered is the security. There is a growing need for providing security for the mobile chat applications, with a large user number it is very important for a chap application to provide all security for the data they store and transmit for better resistance to cyber-attacks. So to provide the security all the messages that are being transmitted and the data stored in the database must be encrypted. For the storage of the user's details and their data, the AWS (Amazon Web service) is used. AWS is a cloud platform that provides several IT infrastructure like database, SQL, virtualization and many more. It allows storing data from the online application and maintains all the user's detail which helps in for the registration and login of the user to the online chat application.***

Keywords: cyber-attacks; AWS; cloud; SQL; virtualization.

I. INTRODUCTION

As we go a few decades behind it was not so easy for people to interact who are geographically in different places. Even though there was a lot of instant messaging applications, but that didn't provide a feeling of real-time conversation to the people. And also the instant messaging applications were not designed for the mobile phones and tablets that made the users use this application only through their desktops. That made the invention of the chat application that allows the user to send message to another person through mobiles, tablets and even through their desktops. This chat application makes the user feel the conversation as real time. The design of the Android chat application is done using the Android studio. It is an Integrated Development Environment (IDE) for the development of Android applications. The Android studio has some modules to build the applications as the user needs, some of the modules are app module, Library module, and Google App Engine modules. All the build files are made visible to the user at the top level under Gradle Scripts. Each application module that is created in Android studio contains the three folders one is manifests which contains the AndroidManifest.xml file in it and the other one is java module which contains the java source code files. The last one is res module which contains the non-code resources like XML layouts, bitmap and UI strings. The structure of Android Application built in Android studio is as shown in Figure 1.1 [1].



```
android-RuntimePermissions-master 〉 Appl
Android
▼ Application
    ▼ manifests
          AndroidManifest.xml
    ▼ java
        ▶ com.example.android
        ▶ common.activities
    ▼ res
        ▶ drawable
        ▶ layout
        ▶ menu
        ▶ mipmap
        ▶ values
▼ Gradle Scripts
      build.gradle (Project: android-Runtim
      build.gradle (Module: Application)
      gradle-wrapper.properties (Gradle V
      settings.gradle (Project Settings)
      local.properties (SDK Location)
```

Figure 1.1 Structure of an Application

Revised Manuscript Received on July 22, 2019.

Sujatha Kumari B A, Jss Science and Technological University, Mysore, Karnataka

Chiranth N L, Jss Science and Technological University, Mysore, Karnataka

Pooja P, Jss Science and Technological University, Mysore, Karnataka

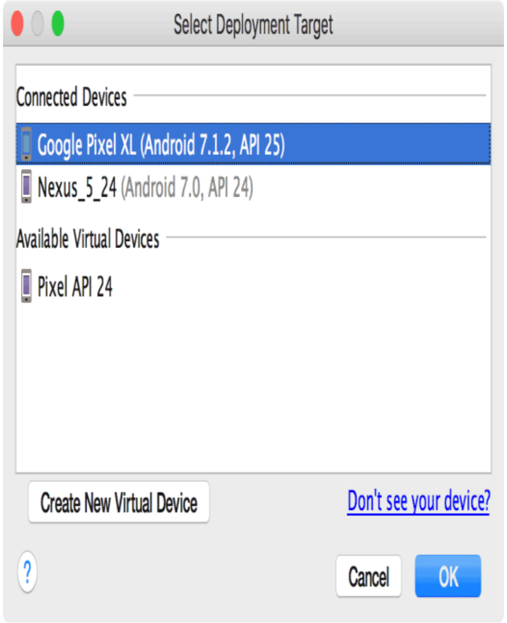

Select Deployment Target
Connected Devices
Google Pixel XL (Android 7.1.2, API 25)
Nexus_5_24 (Android 7.0, API 24)
Available Virtual Devices
Pixel API 24
Create New Virtual Device
Don't see your device?
?
Cancel
OK

Figure 1.2. Selecting a Mobile platform

Android Studio has tools for every development stage of the application, the user only needs to do the coding for the application. Several features are being provided by the Android studio for the users to help to have a more productive platform when coding. Some of the features are code completion, custom code- completion templates, quick fixes of bugs from lint, creates application layouts automatically as the coding is done, helps in creating images for all screen densities. Android Studio allows the application built to be deployed in the Android mobile phones and also the application can be deployed to the Android emulator. As soon as the application is installed, the instant run allows the user to push the code as an application without a new APK creation. The deployment of an android application to mobile phones is shown in Figure 1.2.

The Android studio also helps to link to the externals resources through coding. The database for the application built in the android studio can be an external resource such as a cloud environment. There are several cloud platforms like Amazon Web Services, Google Cloud Platform, Microsoft Azure, etc., all these platforms can be linked and used in the applications built in android studio.

The Amazon web services, also allow the testing of the Android app in a cloud environment as the testing of the application is critically important. Obtaining the physical devices for testing of the applications built is itself a major logistical challenge. At this situation cloud testing helps the user to deploy their applications in the cloud, and test the application as same as the real environment by using the virtual platforms present in the AWS cloud, wherein both the manual and automated tests of the applications can be done to ensure the quality and efficient working of the application. The AWS also provides the database and storage that is needed by the application built, to use all the resources the application must be linked with the cloud environment through environment through coding, so that the data that need to be stored automatically stores in the cloud environment without any user interface. And the AWS cloud ensures that all the data that is being stored by the application is maintained in a secure way so that no data of the application is under threats of attacks. Making use of all the features of both the Android Studio and the AWS cloud platform an efficient and secure chat application is designed [2].

II. IMPLEMENTATION

The chat application is developed using the android studio platform as mentioned in the introduction, the advantages are likely to be faster deployment of fresh builds, more accurate programming, faster programming and testing, inclusive app development and better app indexing, these some of the advantages of Android Studio. The code which is developed consists of four section or four activity they are starter application, main activity, chat activity, and user list activity. An activity is a process to which certain set of action take places, likely to be set of instruction will be given to that activity to perform.

The Starter application is the activity where it describes the whole processes of the app as shown in Figure 2.1, this activity should be executed at the start of the application or on launching the app. The main activity is the activity where it gets the information required from the user in the chat app, the main activity will be asking for the user to log in or signup for the app, based on the user selection, the activity redirects to the specified operation to take place if the user selects the signup, the application will request the username and password to be given for the user account so that later the user can use his account with his username and password in the background the data will be stored in the registered server, but if the user selects the login option user will be requested to produce the correct username and password that was given by him at the time signup, in turn, the password and username will be checked with database stored in the background. If it fails to match, throws an exception message to the user that it is invalid. If it is valid it will be changed to the user list activity which shows the list of the users who are using the app are displayed so that the user can select of his own wish to chat with the other user that he likes and later the chat activity comes into picture where the user will have options to send the messages and emoji's to the other user.

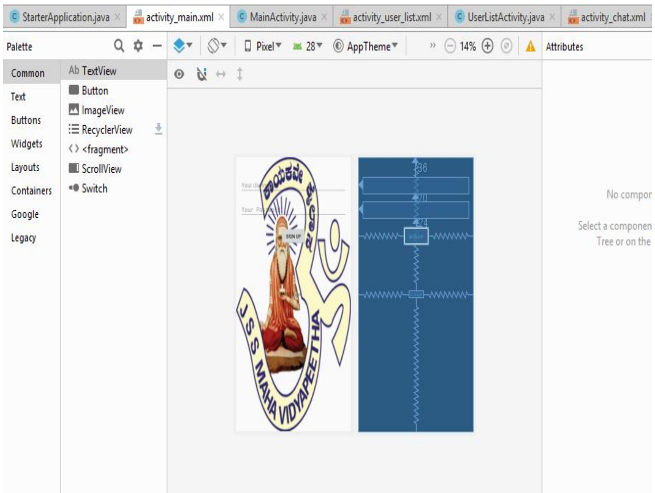
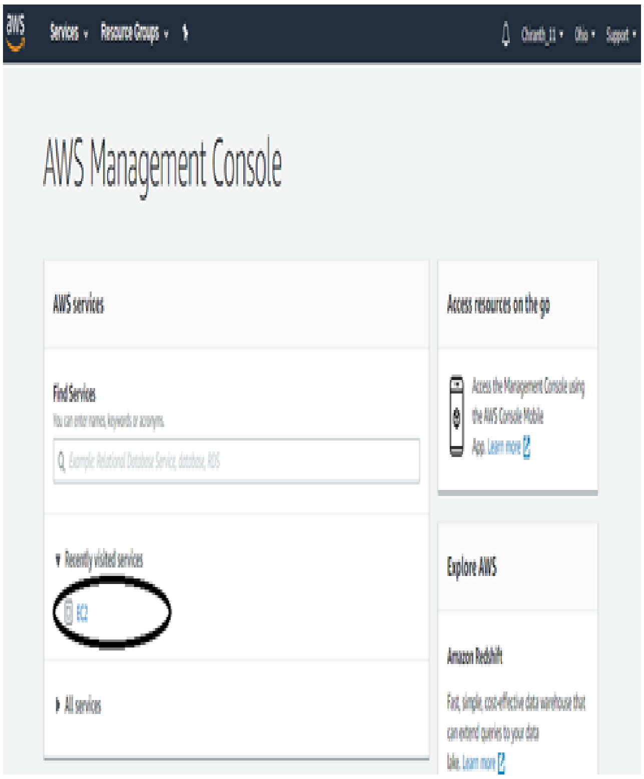


Figure 2.1. Starter Activity (activity_main.xml in android studio) Layout Image

The messages which are exchanged between the two users are secured because the messages that are shared between them should be secure enough such that nonusers or hackers won't read the messages of the users since messaging app is where they share the private messages. The usage of SHA-256 or SHA-512 algorithms to get security for the messages is done. The usage of Parse server instances to establish the server which is freely available in the AWS framework under EC (Elastic Compute) which helps to launch the server with username and random password for the server login. And based on the user OS platform if he using Ubuntu user can just open network terminal to get connected to the server but the situation gets changed in case of windows OS user, for this the user needs to use putty software. In below each step of creating the server and running, it is shown for creating the server for the application.

Step 1: Selection of the EC2 component in AWS frame as shown in the AWS console portal. The EC2
(Elastic compute cloud) that provides a virtual environment for the user to have all the enabled features for having a server to maintain all the details that are generated by the chat application, the selection step is shown in Figure 2.2.



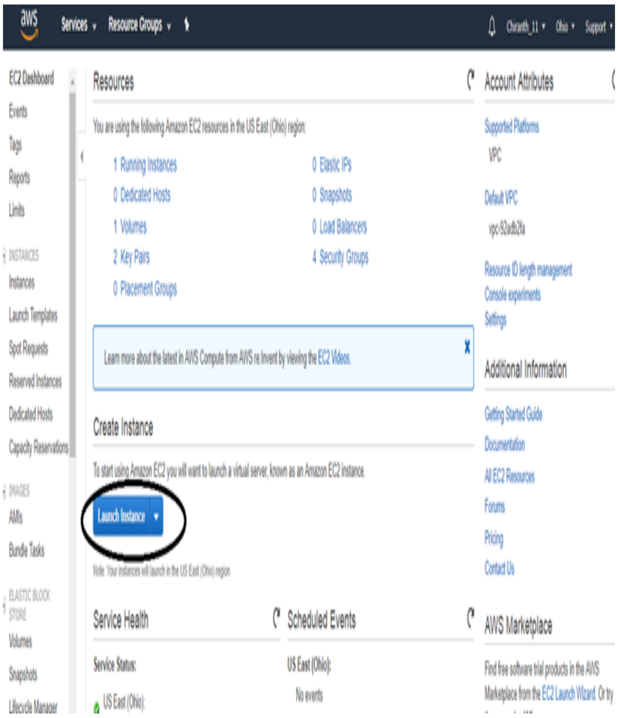
You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

Access the Management Console using the AWS Console Mobile App. Learn more

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

Figure 2.2 Selecting of EC2 component in AWS



Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

1 Running Instances
0 Dedicated Hosts
1 Volumes
2 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
4 Security Groups

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (Ohio) region

Account Attributes

Supported Platforms: VPC

Default VPC: vpc-92adb2fa

Additional Information

Service Health

Service Status:

US East (Ohio):

Scheduled Events

US East (Ohio):

No events

AWS Marketplace

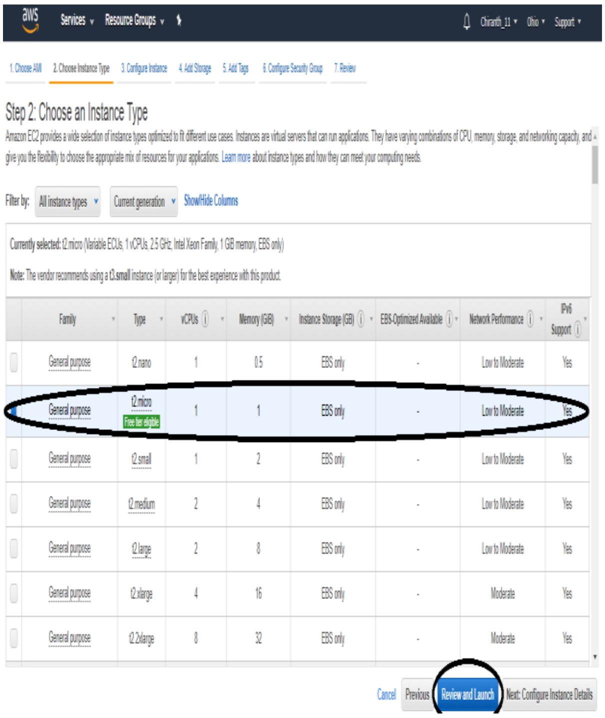
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try

Figure 2.3 Launching an instance in AWS

Step 2: Launching an instance in the EC console, to use the EC2 instance need to be created and that need to be launched, Figure 2.3 shows an image of creating an instance and launching an instance in AWS

Step 3: In this step, the selection of Amazon machine image should be done, wherein we need to select a server to maintain all the data's that is being generated by the chat application. The Parse server is selected as it provides a free service for storing all the data generated by the application. Figure 2.4 shows the selection of Parse servers in AWS Amazon machine image.

Step 4: A paired key will be established when the Parse server is selected, as shown in Figure 2.5 where the key will be used in the application coding part, to get connected between the front end and back end so that the application data will be automatically stored in the Parse server. Finally, the server is launched.



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

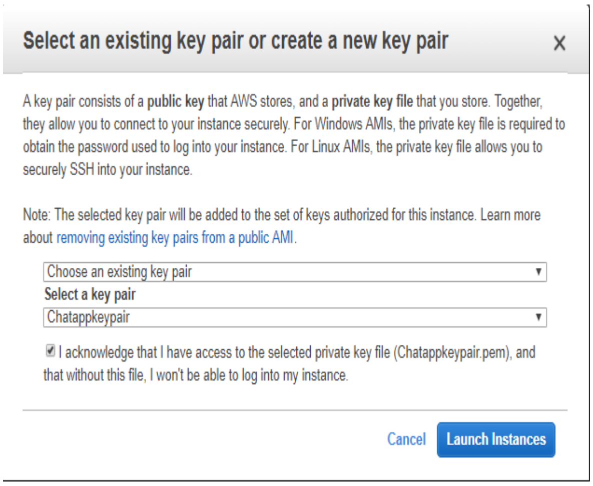
Filter by: All instance types · Current generation

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Note: The vendor recommends using a **t3.small** instance (or larger) for the best experience with this product.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Figure 2.4 Selection of Parse server in AWS



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair

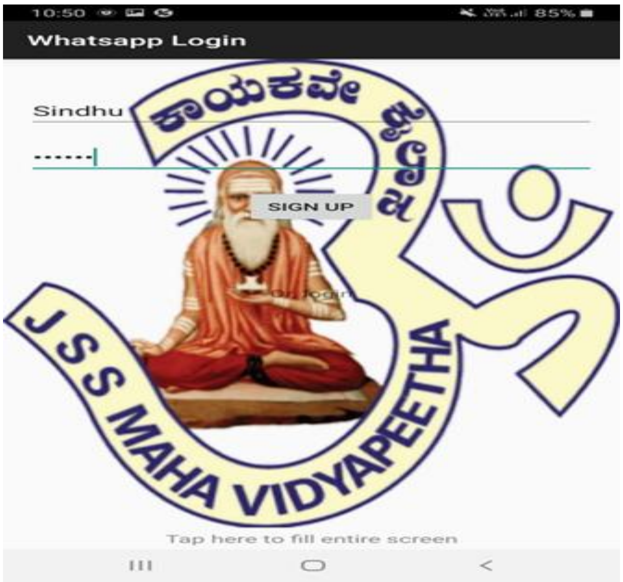
Chatappkeypair

☑ I acknowledge that I have access to the selected private key file (Chatappkeypair.pem), and that without this file, I won't be able to log into my instance.

Figure 2.5 Key pair generation from Parse server

III. RESULTS

The android chat application implementation results are shown in the below figures, Figure 3.1 shows the signup and the login page of the Android chat application. When the unregistered user needs to use the chat application, first the user needs to register by providing his user name and password, the password is encrypted and stored in the AWS. When the registered user wants to use the chat application, as soon as he enters the username and password, the password will be checked whether it is matching with the password stored in the parse sever or not. Figure 3.2 shows the user list that is the list of the registered users, by using that list the user can initiate the conversation through messages. Finally, conversation messages are shown in Figure 3.3.

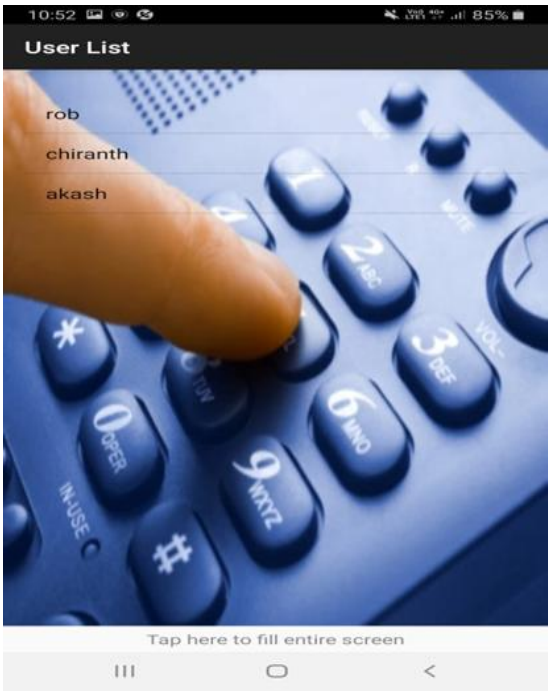


Whatsapp Login

Sindhu

••••••

Figure 3.1 Login page of the Chat Application



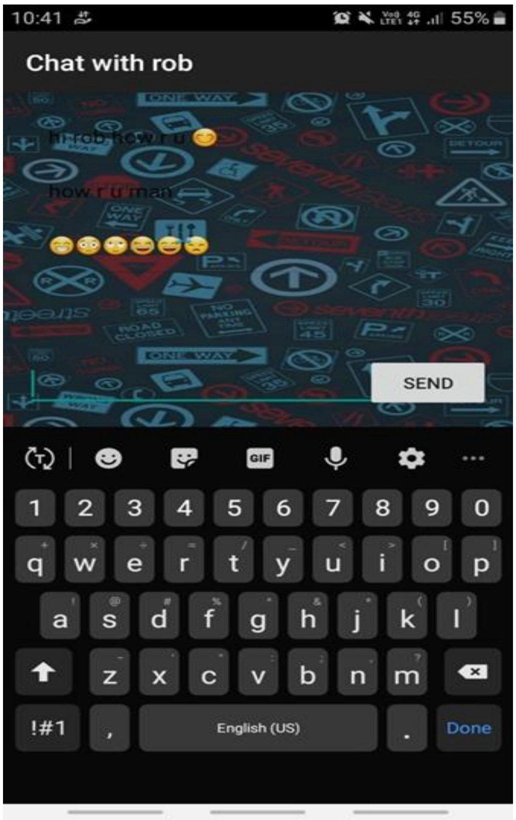
User List

rob

chiranth

akash

Figure 3.2 User List in Chat applications



Chat with rob

hi rob how r u 😊

how r u man

😁😀😁😂😅😒

Figure 3.3 Messages in the application

IV. CONCLUSION

The chat application is one of the most using and trending applications of the current world. The designing of such an application helps the user to know each and every point of the chat application they use also the faults and the loopholes that are present in the application will also be known. Designing of the chat application using AWS is the most advantageous feature as the AWS cloud service provides all the infrastructure that is needed to create an application. Along with that it also helps in encrypting the data and storing it in the storage of the cloud. So that the user's details and the data the user transmits to others are more secure from several cyber-attacks.

REFERENCES

[1]Android studio developer link, https://developer.android.com/studio/features

[2] Amazon web service providers, https://aws.amazon.com/about-aws/

[3] L.T. Duffy, DoD collaboration and chat systems: Current status and way ahead, in: Proceedings of the International Conference on Semantic Computing, IEEE Computer Society, 2008, pp. 573–576.

[4] Paczkowski, J.: WhatsApp: Bigger Than Twitter. Online. All Things D (April 2013), http://allthingsd.com/20130416/whatsapp-bigger-than-twitte/

[5]] E.N. Forsyth, C.H. Martell, Lexical and discourse analysis of online chat dialog, in: Proceedings of the International Conference on Semantic Computing, IEEE Computer Society, 2007, pp. 19–26.

[6] reenwald, G.: English NSA Collecting Phone Record of Millions of Verizon Customers Daily. Online. The Guardian (June 2013), http://www.theguardian.com/ world/2013/jun/06/nsa-phone-records-verizon-court-order

[7]Purnomosidi, B. (2013). Pengembangan Aplikasi Cloud Computing Menggunakan Node.js.

[8] Rahmadini, A. (2013). Pembangunan Sistem Informasi Pengelolaan Inventaris Barang Divisi Pustekin Berbasis Web. Bandung: Politeknik Telkom.

[9] Sidik, B. (2011). JavaScipt. Bandung: Informatika.

[10] Kiessling, Manuel. 2012. The Node Beginner Book. lulu.com, United Stated.

[11] Mardan, Azat. 2012. Practical Node.js: Building Real-World Scalable Web Apps. Appress.

[12] Teixeira, Pedro. 2012. Professional Node.js: Building Javascript Based Scalable Software Kindle Edition. Wrox.