1. INTRODUCTION

In this era of technological advancement, there are numerous messaging apps that users can choose from. Messaging apps are used for chatting purpose. They make communication easy using instant messaging and voice calls. The messaging apps can be installed on a smartphone regardless of the smartphone operating system. All these applications provide various features to ensure security, integrity, and consistency. All these apps let the user send messages and the messages can be of any type like text, images, audios, documents, live videos & audios too. Some of the chat applications of this era are WhatsApp, Facebook, Snapchat, Skype, IMO, Telegram Messenger

1.1 Existing System

Earlier there was no mode of online communication between users. In big or small organizations communication between users posed a challenge. Let consider a few Most used chat applications and their Limitations

1.1.1 WhatsApp

- Group chat for up to 256 people Only
- No Poll System to Know Opinions of Group members
- No Real-Time Language Translation

1.1.2 FaceBook

- It's controlled by Facebook. Privacy issues have been raised in the past and to be fair Facebook was, and still is about monitoring its user behaviour and data collection
- No Poll System to Know Opinions of Page Followers

1.1.3 Telegram

- Allow Malicious & Phishing URLs to open in Device
- Allow to Sending Abusive and Inappropriate Messages
- No Real-Time Language Translation
- Poll System with Limited Options

Drawbacks of existing chat applications:

- No Language Translation
- No SPAM URL detection
- Chat app allows every message whether it is abusive or not
- No polling system

1.2 Proposed System

This Chat application will be like any regular chat app, but with more features like below

- It Restricts Abusive Messages Using Sentiment Analysis
- Real-Time Language Translation
- Polls with Live Results
- Text to Speech
- Text Recognition in Images
- URL & Email Detection in Images
- Tagging Each Message as IMPORTANT, CONFIDENTIAL, PERSONAL
- Drawing Pad

2. REQUIREMENT ANALYSIS

2.1 Introduction

Software requirement is a functional or non-functional need to be implemented in the system. Functional means providing a particular service to the user. For example, in context to this chat application, the functional requirement will be when the user sends a message it must be able to send to the receiver at any cost.

Software requirement can also be a non-functional, it can be a performance requirement. For example, a non-functional requirement is where user able to read old messages in chat app without internet. software requirement is a Functional or Non-functional need that has to be implemented into the system. Software requirement is usually expressed as a statement.

2.2Hardware and Software requirements

Hardware requirements for talkie are

- Processor: Minimum 1GHz. Recommended 2GHz or more.
- Memory (RAM): Minimum 3GB.
- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)

Software requirements for talkie are

- Editors
  - VS code editor
  - Android Studio
- Backend
  - Node.js
- Database
  - MongoDB
- Testers
  - Socket.io Tester
  - POSTMAN

2.3 Software Requirements Specification

The following section provides an overview of the derived Software Requirements Specification (SRS) for the subject talkie. The document is presented and its' intended audience outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant application will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

2.3.1 Vision

It's The Time for the next evolution in Internet Communication. Bringing together different elements of the AI (e.g. conversational AI, machine learning, and deep learning), we have a unique vision for our messenger.

Although they've been a lot of Messenger apps, we will add our futuristic vision to come up with some new ideas on what we think a Messenger should be, and with whom (for now) they should be interacting with.

2.3.2 Objective

- **Communication:** To develop an instant messaging solution to enable users to seamlessly communicate with each other
- **User-friendliness:** The project should be very easy to use enabling even a novice person to use it.
- **Restriction of Abusive messages**: chat app can understand the polarity of the message using sentiment analysis by which it restricts abusive messages
- **Translation of Messages**: translates message text from one language to other languages. It supports 74 languages

2.3.3 Scope

- Chat Application is going to be a communication app, it will be able to communicate between two android devices using point to point socket communication.

- The limitation of the Chat is it does not support the restriction of other language abusive messages. To overcome this limitation we are concurrently working on developing better package than can do sentiment analysis on any language.

- Companies would like to have a communication software wherein they can communicate instantly within their organization.

- The fact that the app restricts the user to open any URL in chat and make the device more secure from malicious URLs. User can open URLs after running URL classifier.

Exclusions:

- Users cannot access the data in the database directly.

- User who is not admin cannot add or remove members.

Assumptions:

- The database is considered here.

- Database that consists of information about the registered users, chats, messages, groups, group messages.

- The system has to authenticate a user, determine whether he is valid or not.

2.3.4 System Functions

User

- Log in to the app.
- Send a message to contacts having talkie account.
- Translate message from one language to another
- Delete message
- Tag a message
- Update the tag
- Create a Group

Admin

- Add group members
- Delete Group members

2.3.5 Detailed Software Requirements

Actor Name	Administrator
Actor Id	TC01
Description	Handles all admin related tasks throughout the application
Main Activities	Uses the system to setup initial data, define access control
Frequency of Use	Medium
Work Environment Location	App
Number of Users	Any Number

Table 2.1 Use case- Admin

Actor Name	User
Actor Id	TC02
Description	User of the Chat App.
Main Activities	User will communicate with other users through messages, files, polls in both personal and group chat
Frequency of Use	High
Work Environment Location	App
Number of Users	Any Number

Table 2.2 Use case- User

2.3.6 Functional Capabilities

- User can send messages, files, polls
- User can create a group
- User can Translate the message from one language to other
- User can tag the message as IMPORTANT, PERSONAL, CONFIDENTIAL
- User can delete a message
- Admin can add or remove members

2.3.6.1 Business Rules / Validation

Only the Admin can update members.

2.3.7 Security Requirements

2.3.7.1 User Management and Authentication

- The telephone number of the user should be the identification for the user.
- If The telephone number Used by another new user his details will be replaced with new users detail

2.3.7.2 User roles and access controls:

- The system has to authenticate a user, determine whether he is an old user or a new user.
- Only authorized personnel with a token will be allowed to access the Chat App
- A user who creates the group is default set as admin
- One group can have one admin
- User with Admin Rights Can remove or add members in the group

2.3.8 Other Non-functional requirements

- The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. (Performance)
- Flexible service-based architecture will be highly desirable for future extension (Supportability)
- The system should be available 24 X 7. (Reliability/ availability) Downtime should be less than 1%. (Reliability/availability)

Exclusions:
User cannot chat directly with any user. User can chat with other User only if both users saved their Contact number.

Assumptions
Database is considered here.

- Database that consists of information about the registered users, messages, groups.
- Only the group admin can update users in a group.

3. SYSTEM DESIGN

3.1 Introduction

The purpose of the design phase is to plan a solution for the problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how they interact with each other to produce the result. The goal of the design process is to produce a model or representation of a system can be used later to build that system. The produced model is called the design of the system.

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It provides sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture**.**

3.2 UML Diagram

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modelling and documenting software. It's one of the most popular business process modelling techniques.It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we can better understand possible flaws or errors in software or business processes.

UML has been used as a general-purpose modelling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modelling workflows as well as a wider range of features to improve readability and efficacy.

3.2.1 Use case diagram for entire system

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

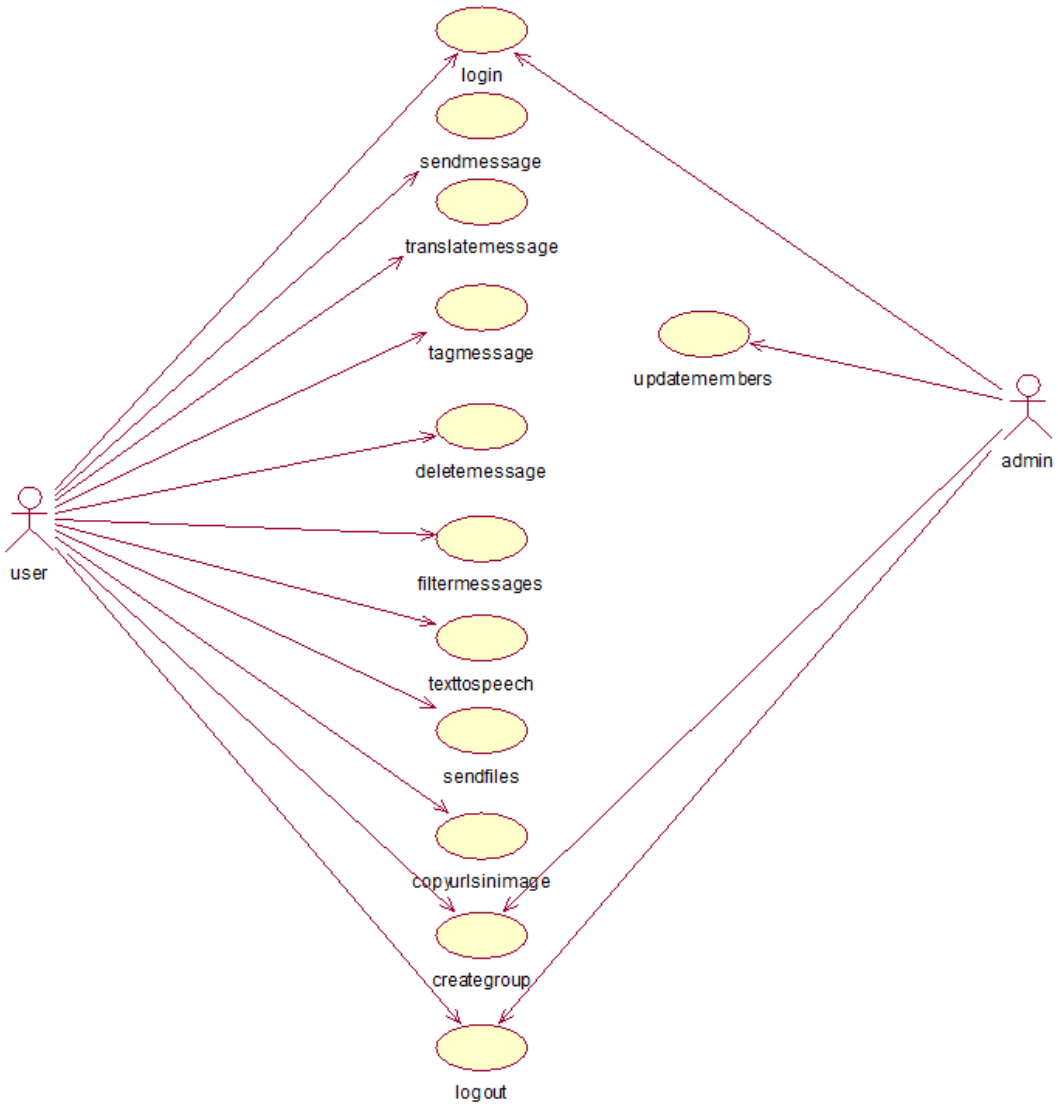


Fig 3.1 Use case diagram for Entire System

The above Use case diagram represents the functionality of entire system. The system contains two actors user and admin, where user have to login into account, send message, translate message, tag message, delete message, filter message, text to speech, send files, copy urls in image, create group, logout. In the similar way the admin can login into the account and can update the group members.

3.2.2 Use case for admin

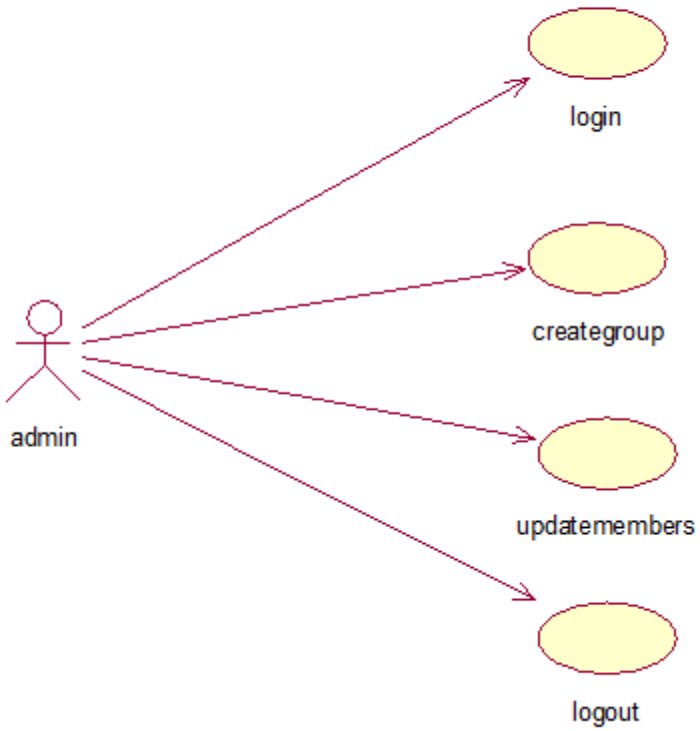


Fig 3.2 Use case diagram for admin

The Use case diagram for admin describes the functionalities of the admin.The admin can perform similar operations like user but he can update members of the group like he can add members and delete members if required.

3.2.3 Use case for user

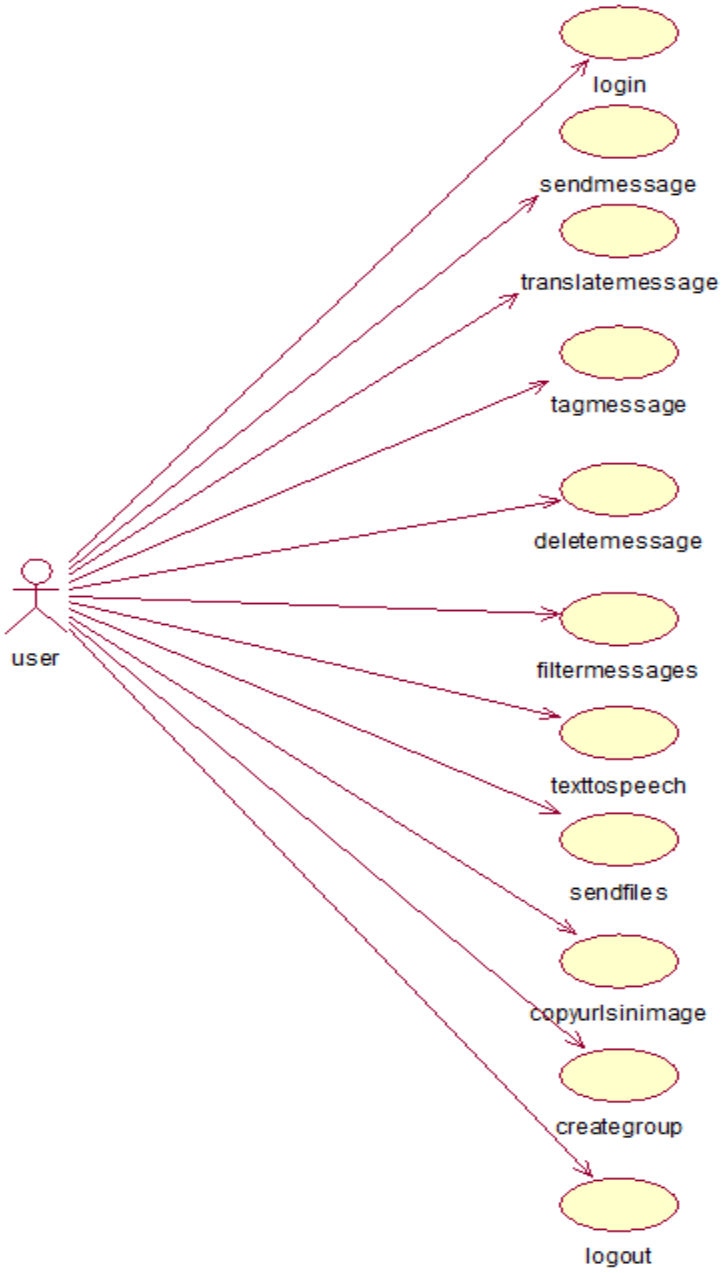


Fig 3.3 Use case diagram for user

The above Use case diagram describes about the functionality of the user. Once the user logins use will be redirected to chat page. The user can send message, translate message, tag message, delete message, filter message, text to speech, send files, copy urls in image, create group, logout.

Use case

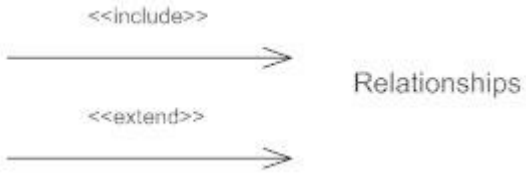
Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

Relationships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labelled either "uses" or "extends."



3.2.4 Sequence Diagram

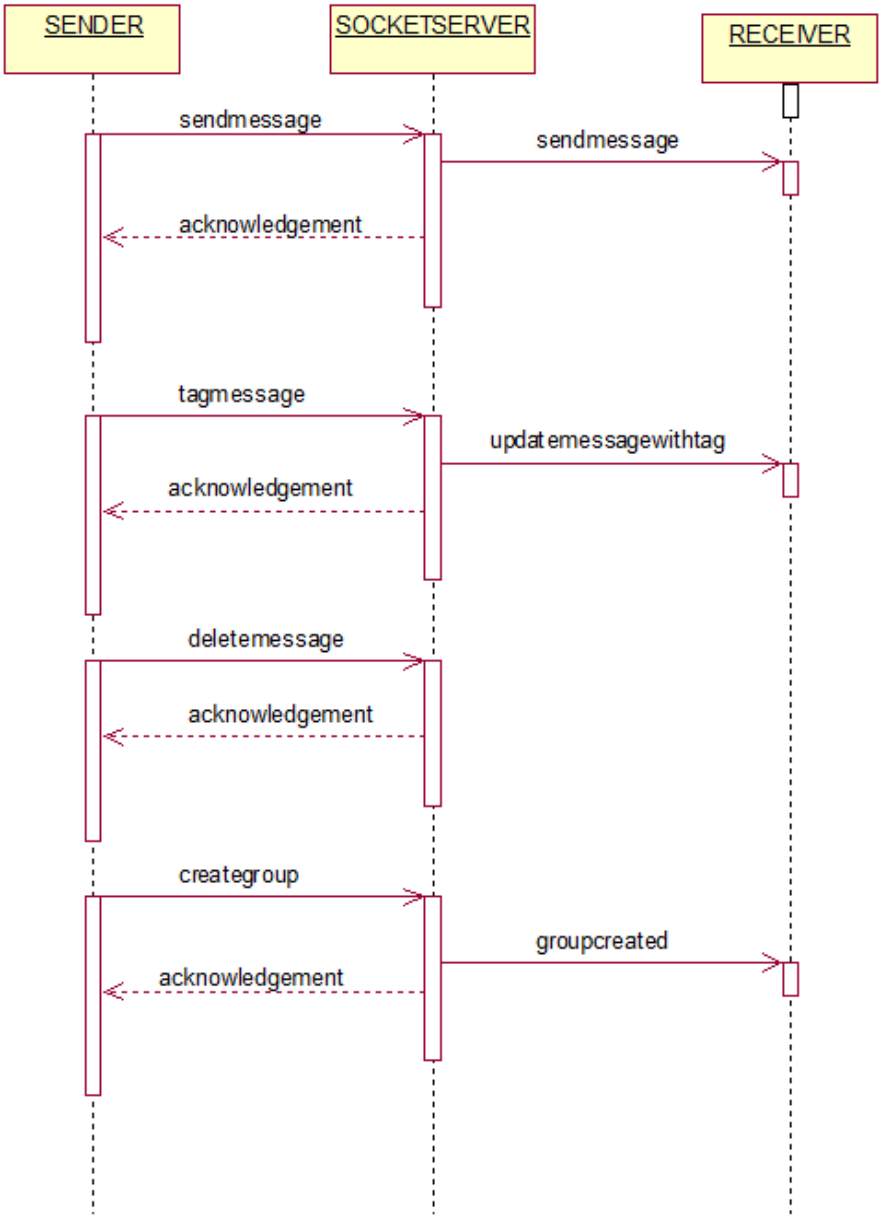


Fig 3.4 Sequence diagram

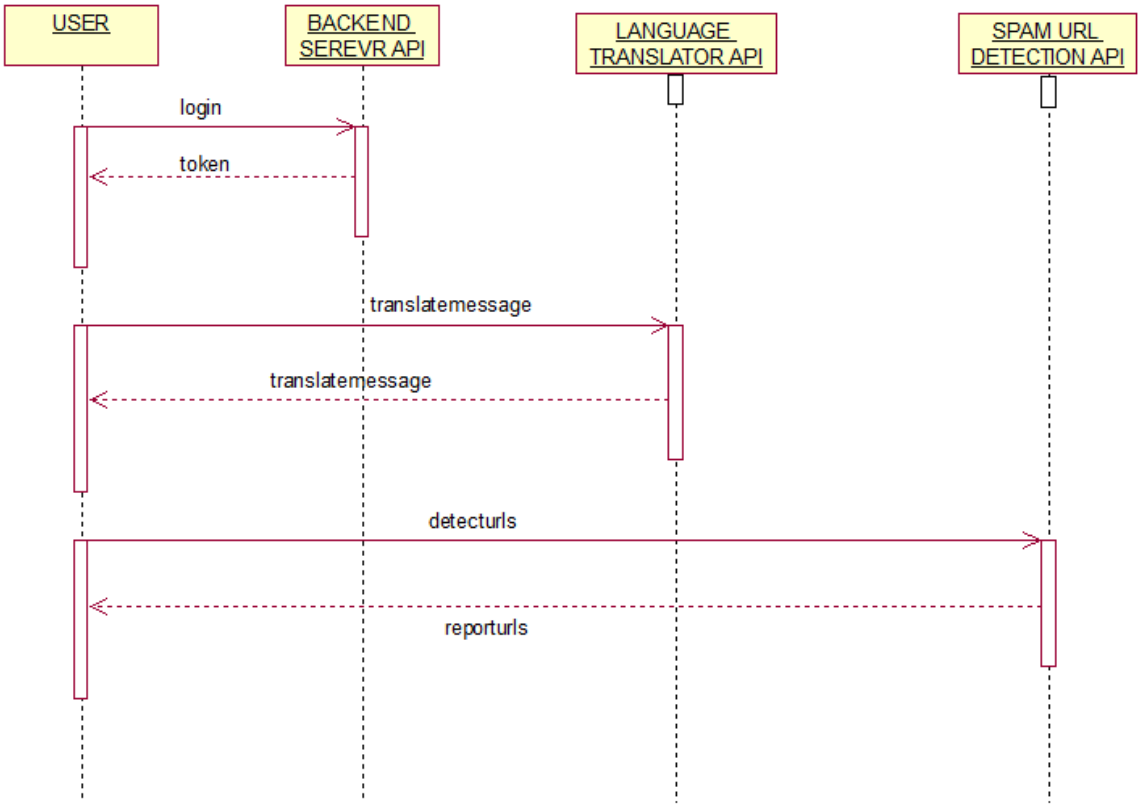


Fig 3.5 Sequence diagram

Sequence Diagram

As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active only when needed or when another object wants to communicate with them. All communication is represented in a chronological manner. A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

3.2.5 Class Diagram

A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations.

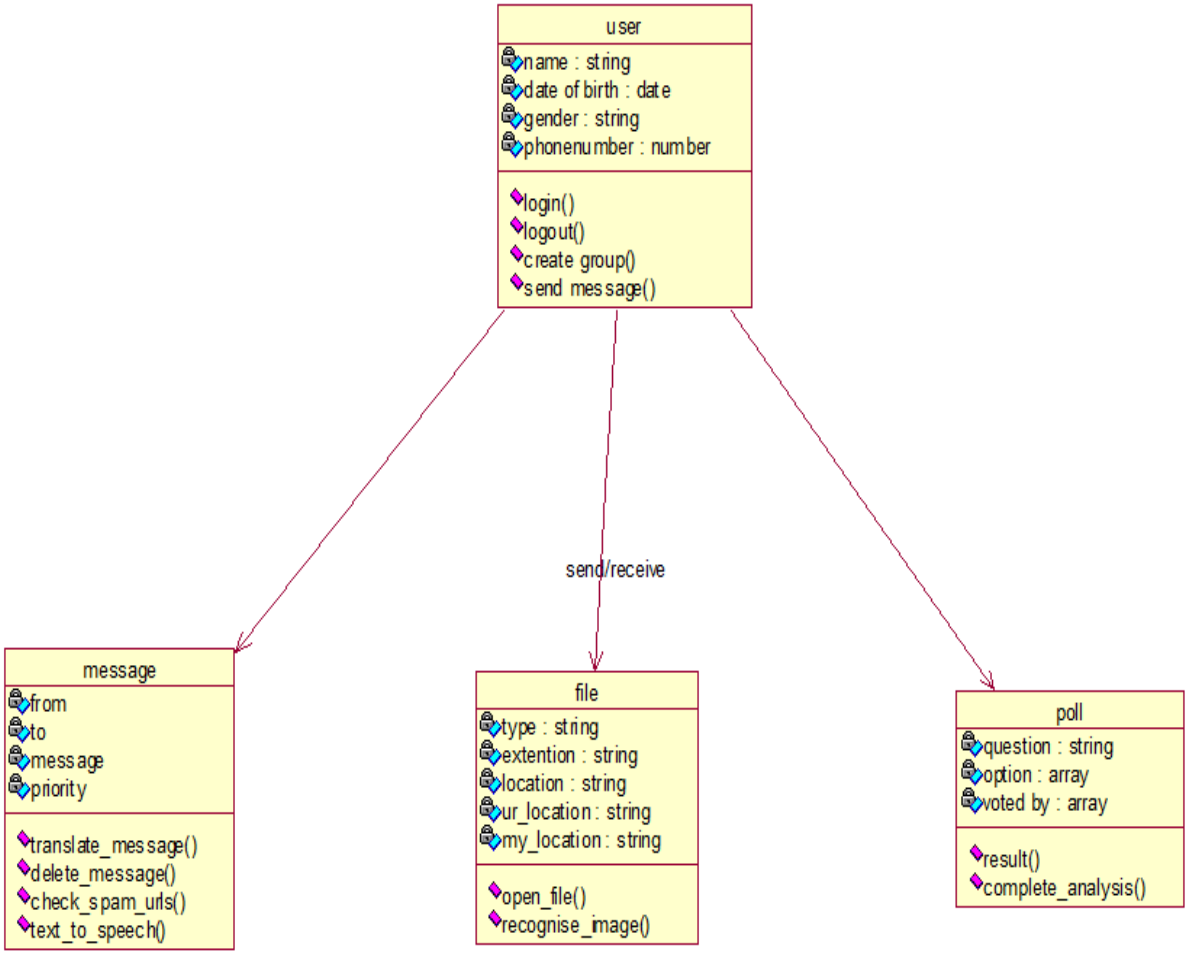
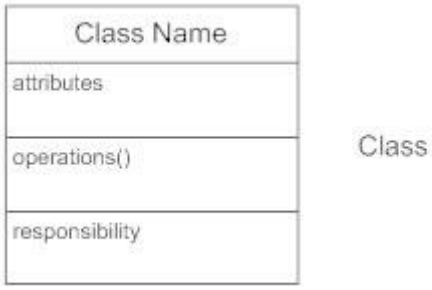


Fig 3.6 Class diagram

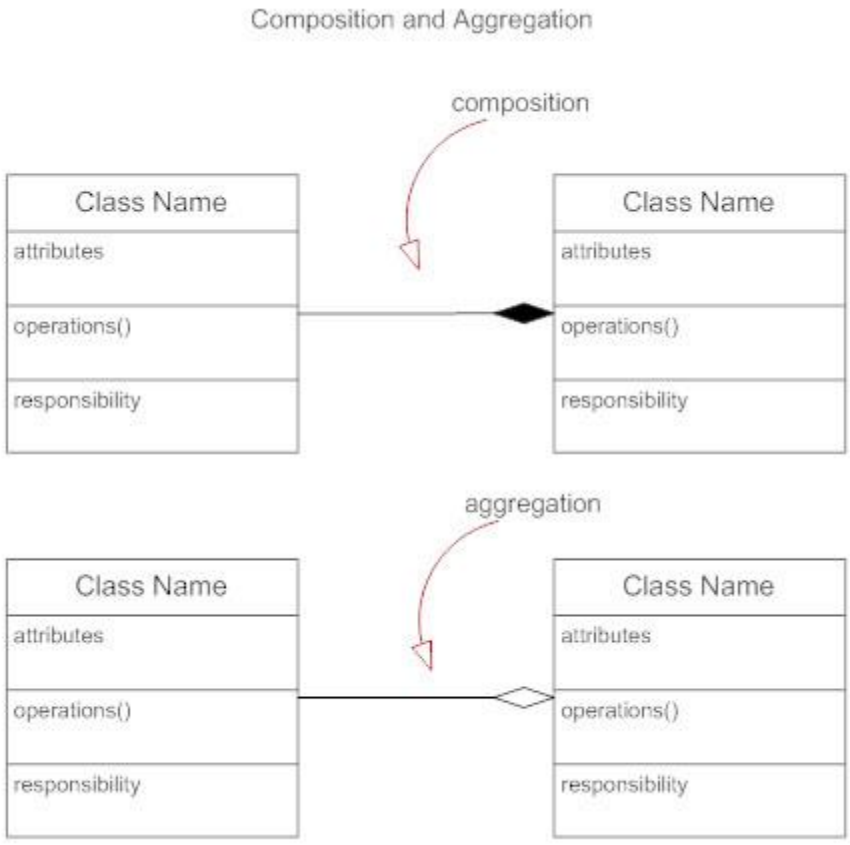
Classes

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.



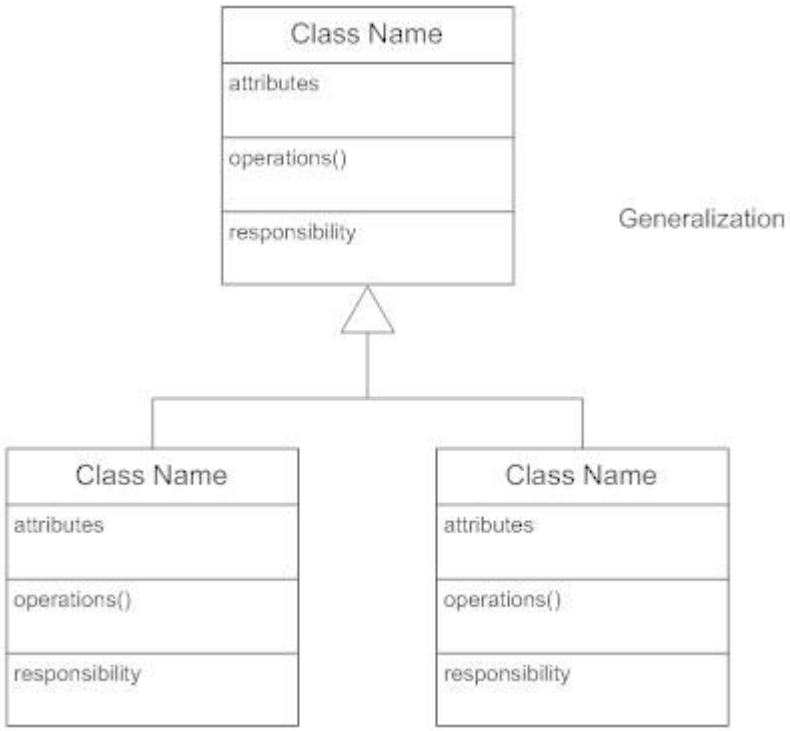
Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other.



Generalization

Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another.



4. SYSTEM IMPLEMENTATION

4.1 Introduction

Implementation simply means carrying out the activities described in your work plan. Executing a project in the water and sanitation sector is a very complex mission, as it requires the coordination of a wide range of activities, the overseeing of a team, the management of budget, the communication to the public, among other issues. Independent of whether it is a social project to raise the awareness and promote hygiene or it is a construction project for service delivery, there is a certain process that has to be followed. The following lines will give you an introduction into the implementation of projects in sustainable sanitation and water management, and highlights key aspects that have to be taken into account for a successful implementation

4.2 Selected Software

4.2.1 Technologies:

1. Ionic
2. Apache Cordova
3. Node.js
4. Flask

4.2.2 Libraries or Dependencies

1. Pandas
2. Scikit-learn
3. Socket.io
4. Sentiment
5. OCR

Technologies:

1.Ionic:

Ionic Framework is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies (HTML, CSS, and JavaScript).It is focused on the frontend user experience, or UI interaction of an app (controls, interactions, gestures, animations). It's easy to learn, and integrates nicely with other libraries or frameworks, such as Angular, or can be used standalone without a frontend framework using a simple script include.

2.Node.js:

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web

page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

3.Apache Cordova:

Apache Cordova (formerly Phone Gap) is a mobile application development framework originally created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as Phone Gap, and later released an open source version of the software called Apache Cordova. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone. It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device.

4.Flask :

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Libraries or Dependencies

1.Packages used in Node.js

Sentiment

Sentiment is a Node.js module that uses the **AFINN-165** wordlist and **Emoji Sentiment Ranking** to perform **sentiment analysis** on arbitrary blocks of input text. Sentiment provides several things:

- The ability to append and overwrite word / value pairs from the AFINN wordlist
- The ability to easily define custom strategies for negation, emphasis, etc. on a per-language basis

Socket.io

Socket.IO enables real-time bidirectional event-based communication. It consists of:

- a Node.js server
- a Javascript client library for the browser (or a Node.js client)

2.Packages used in Ionic

This Package defines a global textocr object, which provides an method that accepts image uri or base64 inputs. If some text was detected in the image, this text will be returned as a string.

If text if Found Following like Object is returned

{

"foundText" : true,

"blocks": {

"blocktext": ["Home]

}

}

Following Object will be returned if No Text is found

{

"foundText" : false

}

3.Packages used in Python

Pandas: Pandas Data Frame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas Data Frame consists of three principal components, the **data, rows,** and **columns**. In the real world, a Pandas Data Frame will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, and Excel file. Pandas Data Frame can be created from the lists, dictionary, and from a list of dictionary etc.

Scikit-learn: Scikit-learn is probably the most useful library for machine learning in python. It is on Numpy, Scipy and matplot lib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimenisionality reduction.

Implementation of Sentiment analysis using Node.js library:

To achieve Sentiment analysis, following process is considered.

AFINN is a list of words rated for valence with an integer between -5(negative) to +5(positive). Sentiment analysis is performed by cross-checking the string tokens (words, emojis) with the AFINN list and getting their respective scores.
The comparative score is simply: **sum of each token / number of tokens**.
So for example let's take the following:
I think this is a good time to do bad things
That string results in the following:

```
{
  score: 0,
  comparative: 0,
  calculation: [ { bad: -3 }, { good: 3 } ],
  tokens: [
    'i',      'think',
    'this',   'is',
    'a',      'good',
    'time',   'to',
    'do',     'bad',
    'things'
  ],
  words: [ 'bad', 'good' ],
  positive: [ 'good' ],
  negative: [ 'bad' ]
}
```

Returned Objects

- **Score**: Score computed by adding the sentiment values of recognized words.
- **Comparative**: Comparative score of the input string.
- **Calculation**: An array of words that have a negative or positive valence with their respective AFINN score.
- **Token**: All the tokens like words or emojis found in the input string.
- **Words**: List of words from input string that were found in AFINN list.
- **Positive**: List of positive words in input string that were found in AFINN list.
- **Negative**: List of negative words in input string that were found in AFINN list.

In this case, good has a value of 3, bad has a value of -3, and the remaining tokens are neutral with a value of 0. Because the string has 14 tokens the resulting comparative score looks like: (3 + -3) / 14 = 0,

This approach leaves you with a mid-point of 0 and the upper and lower bounds are constrained to +5 and -5 respectively

Implementation of Naïve Bayes using SKlearn library:

For working of Naive Bayes algorithm in our project as of the work-flow shown in above section we have implemented some classed and functions, some of them are predefined which are provided by the **Sklearn** library and some of them are user defined.

First we need to import MultinomialNB from **Sklearn**

from sklearn.naive_bayes import MultinomialNB

Function used to generate model which is saved into variable clf :

clf = MultinomialNB()

Then this model will be trained with Training dataset:

clf.fit(X_train,y_train)

Following Command to Saved Model as file so that we don't need to train again and again

joblib.dump(clf, 'model.pkl')

Retrieving model from Pickle file

clf = joblib.load('model.pkl')

Predicting Url Class

```
Inputs=['www.google.com','www.fake-bank.com']
vect = cv.transform(inputs).toarray()
x = str(clf.predict(vect))
print(x)
```

[0 1]

1 represents SPAM class

0 represents HAM class

Implementation of Polling using Array with JSON Objects:

Declaring array with JSON Object as follows

```
options = [
  {
    name : '',
    score : 0,
    votedby : []
  }
];
```

When User want to add another option another JSON Object is pushed into the array

Attributes in Each JSON Object and their description

- name attribute data type is String denoting option name
- score attributes data type is Integer denoting option name
- votedby attributes data type is Array denoting Voters Id's who voted for this option

Whenever a new User Voted for Option his ID will be pushed into the array

Implementation of Message Restriction and Sentiment Analysis of Message using Sentiment Package:

Consider Following Example

```
JS sentiment.js > ...
1   var Sentiment = require('sentiment');
2   var sentiment = new Sentiment();
3   var result = sentiment.analyze('damn shit');
4   console.dir(result);
```

Result

```
{
  score: -6,
  comparative: -3,
  calculation: [ { shit: -4 }, { damn: -2 } ],
  tokens: [ 'damn', 'shit' ],
  words: [ 'shit', 'damn' ],
  positive: [],
  negative: [ 'shit', 'damn' ]
}
```

Performing Sentiment Analysis and assigning Score to the Message.

If Score if less than -5, It seems to Abusive Message. The message will be not be sent to the receiver and a Warning message will be displayed at sender Side Since he needs to understand his abusive message will not be received by the receiver.

If Message I having polarity greater than 0 Message Seems to be positive and Displayed with Green Chat Bubble

If Message I having polarity less than 0 and greater than -5 Message Seems to be negative and Displayed with Red Chat Bubble.

If Message I having polarity equal to 0. Message Seems to be neutral and Displayed with Blue Bubble.

4.3. Sample Code

4.3.1. Backend API Code

```
app.post('/login', function (req, res) {
  User.findOne({ phoneNumber: req.body.phoneNumber }, (err, user) => {
    if (user) {
      User.updateOne({ phoneNumber: { $eq: req.body.phoneNumber } }, req.body, (err, data) =>
 {
        if (data) {
          return res.status(201).json({ token: createToken(req.body) });
        }
        if (err) {
          return res.status(400).json({ msg: err });
        }
      });
    }
    else {
      let newuser = User(req.body);
      newuser.save((err, user) => {
        if (err) {
          return res.status(400).json({ msg: err });
        }
        return res.status(201).json({ token: createToken(user) });
      });
    }
    if (err) {
      return res.status(400).json({ msg: err });
    }
  });
})
```

4.3.2 Socket API

```
    socket.on("send-message", message => {

    var x = sentiment.analyze(message.message);
    if (x.score < -5) {    message.isBan = true; }
    message.score = x.score;
    message.createdAt = new Date();
    let newMessage = Message(message);
    newMessage.save(function (err, data) {
      if (data) {
        channel = data.from + "-" + data.to;
        io.emit(channel, data);
      }
    })  });
```

4.3.3 Translator API

```
 let options = {
   method: 'POST',
   baseUrl: endpoint,
   url: 'translate',
   qs: {
    'api-version': '3.0',
    'to': req.body.languages
   },
   headers: {
    'Ocp-Apim-Subscription-Key': subscriptionKey,
    'Content-type': 'application/json',
    'X-ClientTraceId': uuidv4().toString()
   },
   body: [{
        'text': req.body.text
   }],
   json: true,
};
return request(options);
})
```

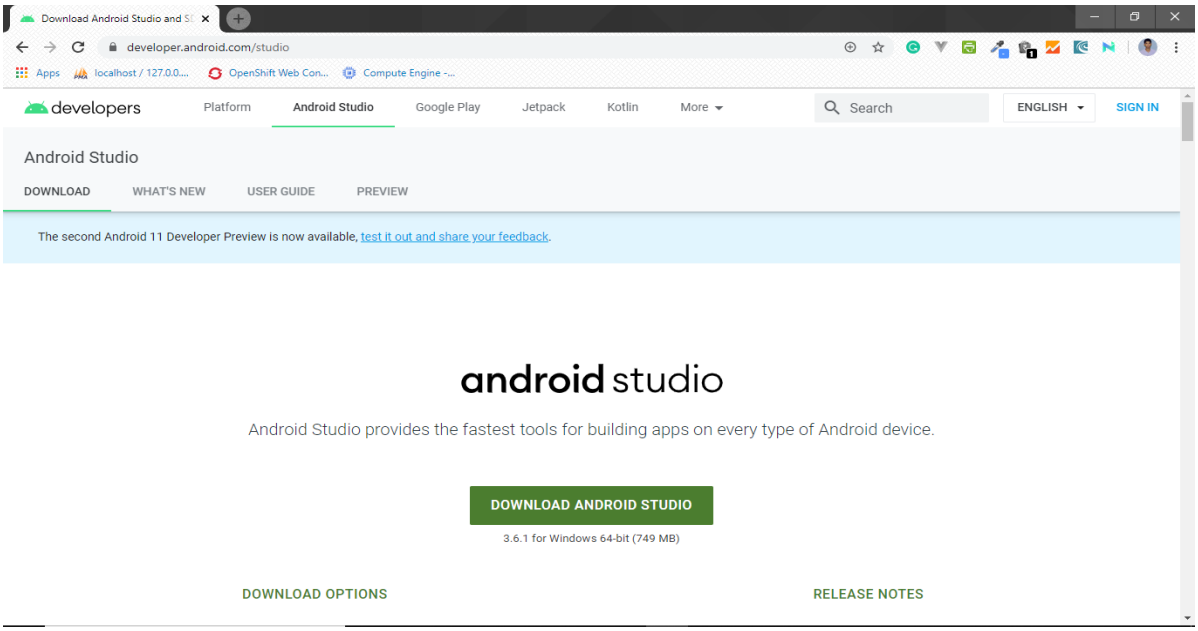
4.3.4 SPAM URL Detection

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
df = pd.read_csv('dataset.csv',encoding = "ISO-8859-1")
X = df['url']
y = df['label']
cv = CountVectorizer()
X = cv.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
clf = MultinomialNB()
clf.fit(X_train,y_train)
vect = cv.transform(inputs).toarray()
clf.predict(vect)
```

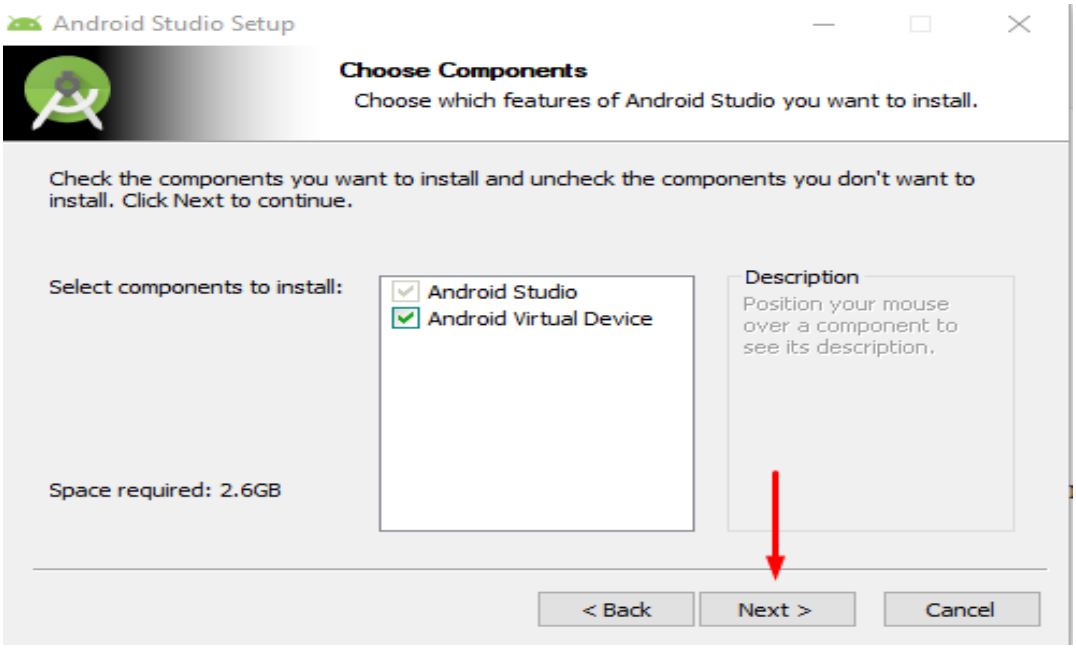
4.4 INSTALLATION STEPS

4.4.1 Installing ANDROID STUDIO:

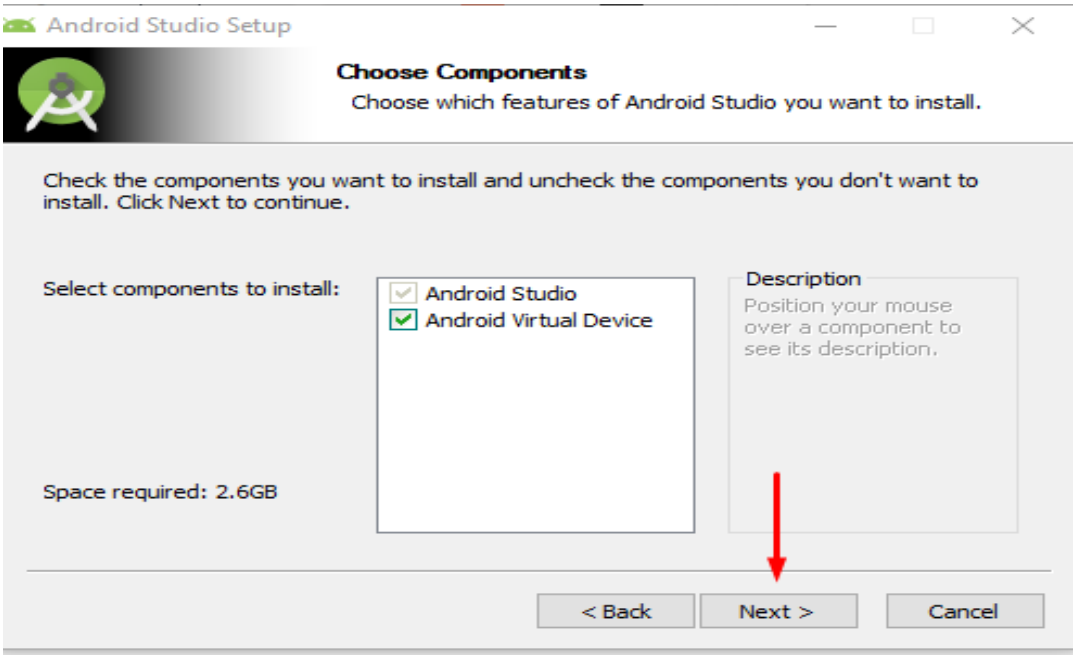
1.Download the binary file (.exe file) of Android Studio for Windows 10 on its official site which is looking like below.



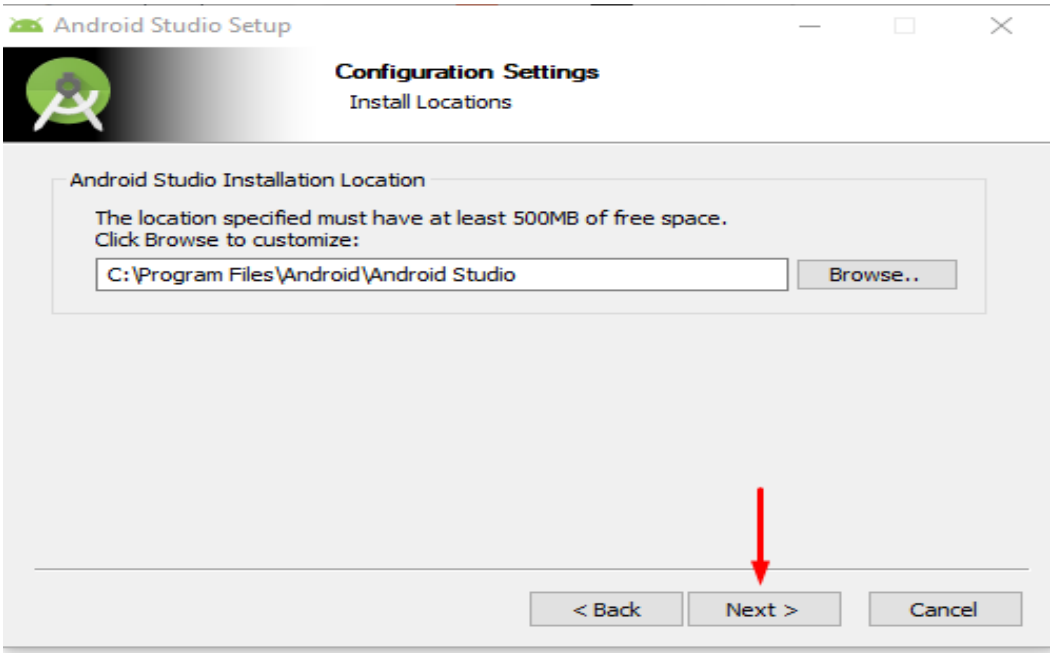
2.Before installing, make sure your system has at least 3GB of RAM (8GB is more recommended) and 2GB of free disk space (4GB is more recommended). Once you have downloaded the binary file of Android Studio for Windows, double-click it to start installing. Click the Next button to continue.



3. On the next step, you will be asked what components you want to install. You will need all of the Android Studio components so just leave default to install all components and click the Next button to continue.

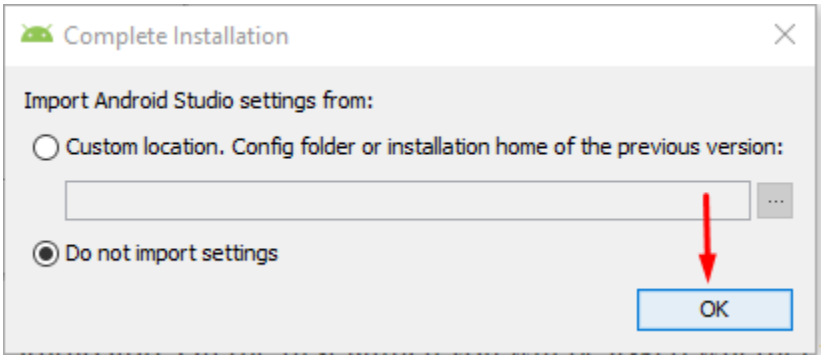


4. Next, select the installation location of Android Studio and click the Next button. The installer will be copying the necessary files to your computer.

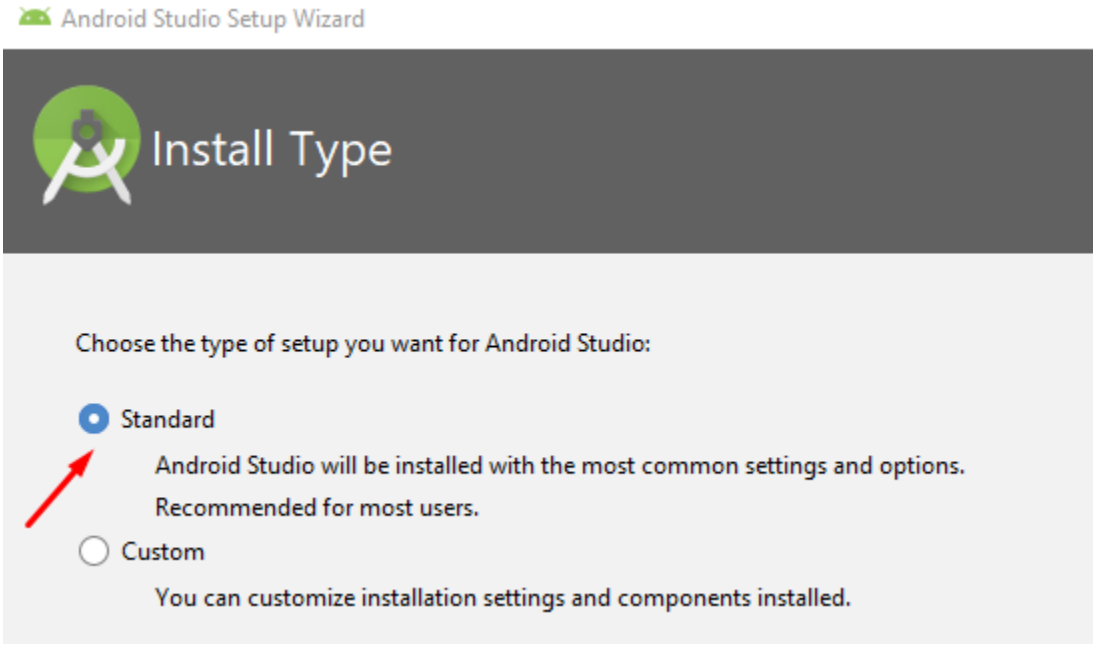


5. Set up Android Studio on Windows 10

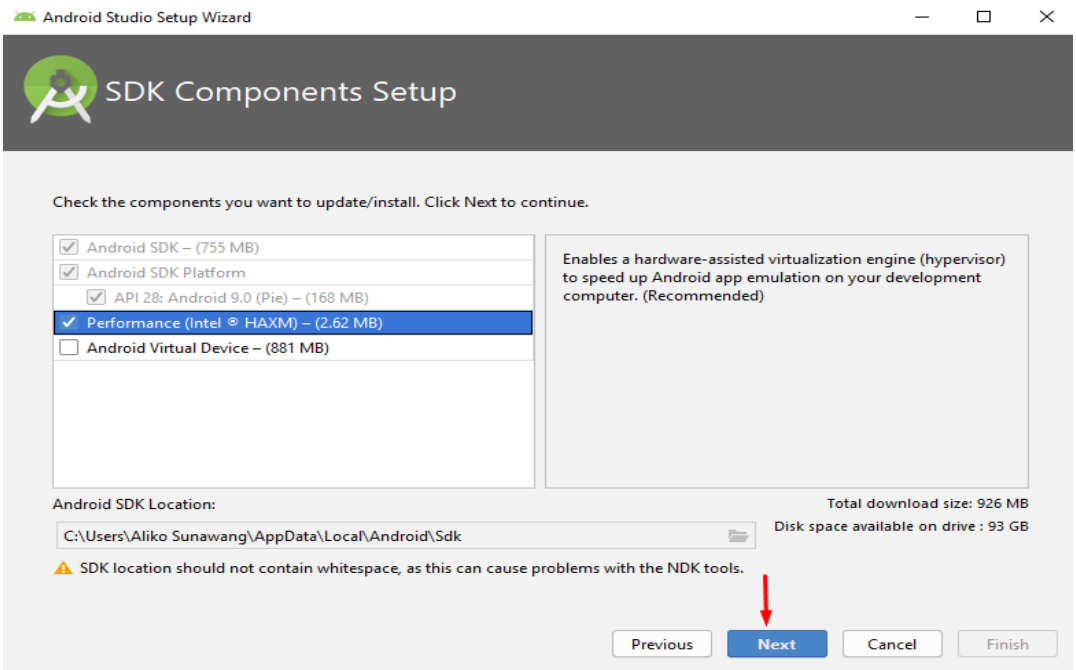
6.Once the installation process finished, launch the Android Studio app. On the first launch you will be asked whether you want to import the settings. Select the Do not import settings option and click the OK button to continue.



7. On the setup wizard, select the Standard option

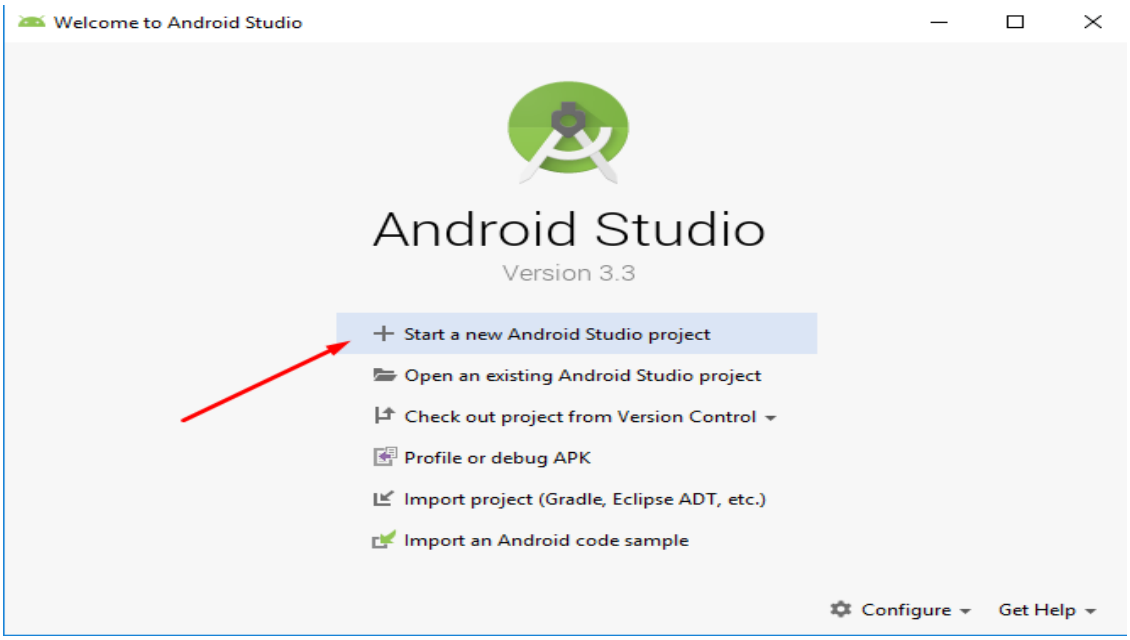


8. On the next step, leave it default and click the Next button.



9. Android Studio will be downloading the necessary packages before you can use it. Wait until the process is done. This part can take a while so be patient. Don't forget to click the Finish button once the download process is done.

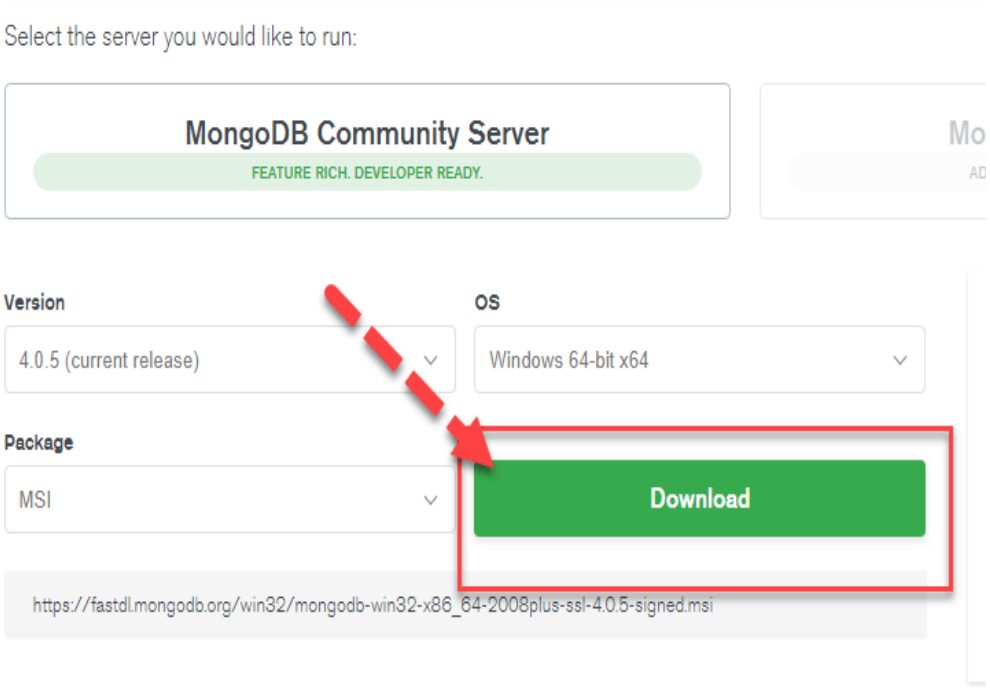
10. After Successful Installation below window will be appeared



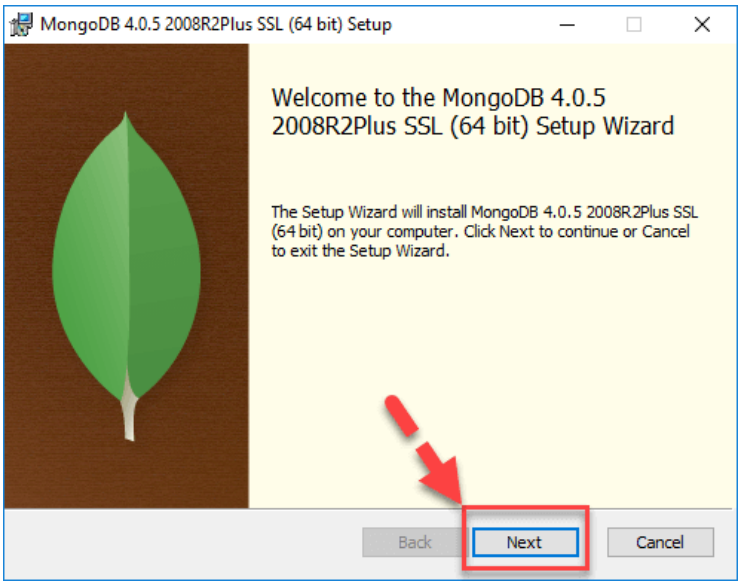
4.4.2 Installing MONGODB:

The following steps can be used to install MongoDB on Windows 10

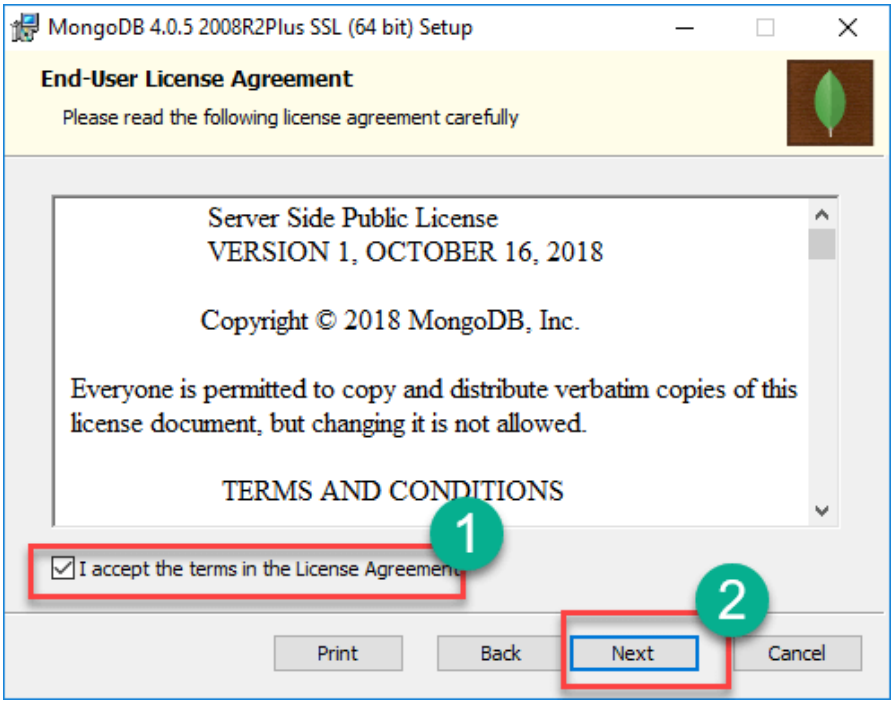
1. Go to link and Download MongoDB Community Server. We will install the 64-bit version for Windows



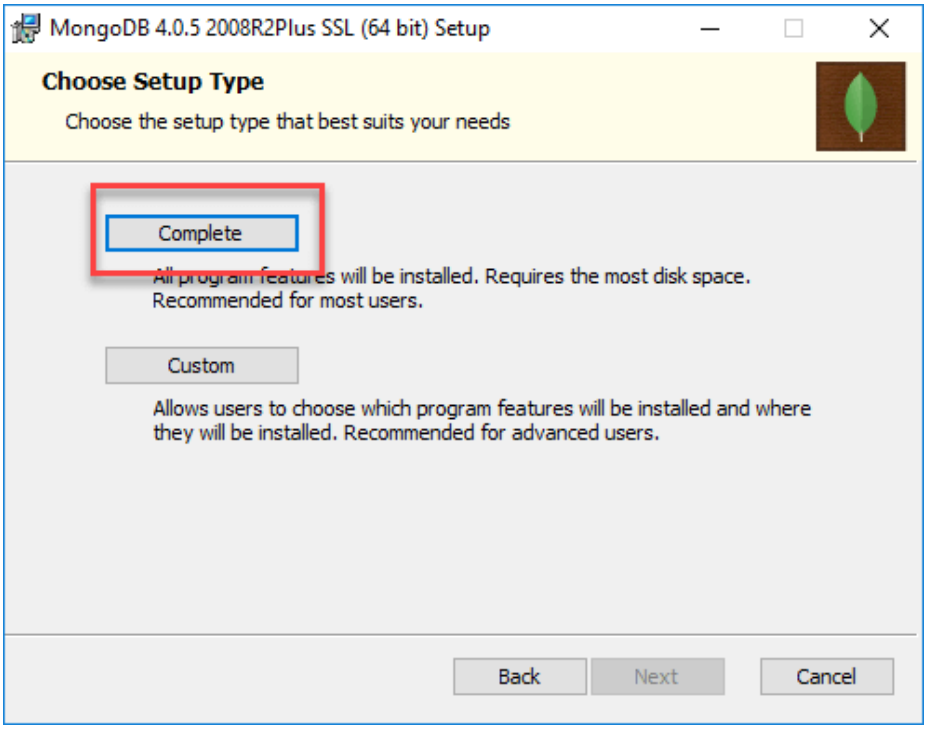
2. Once download is complete open the msi file. Click Next in the start up screen



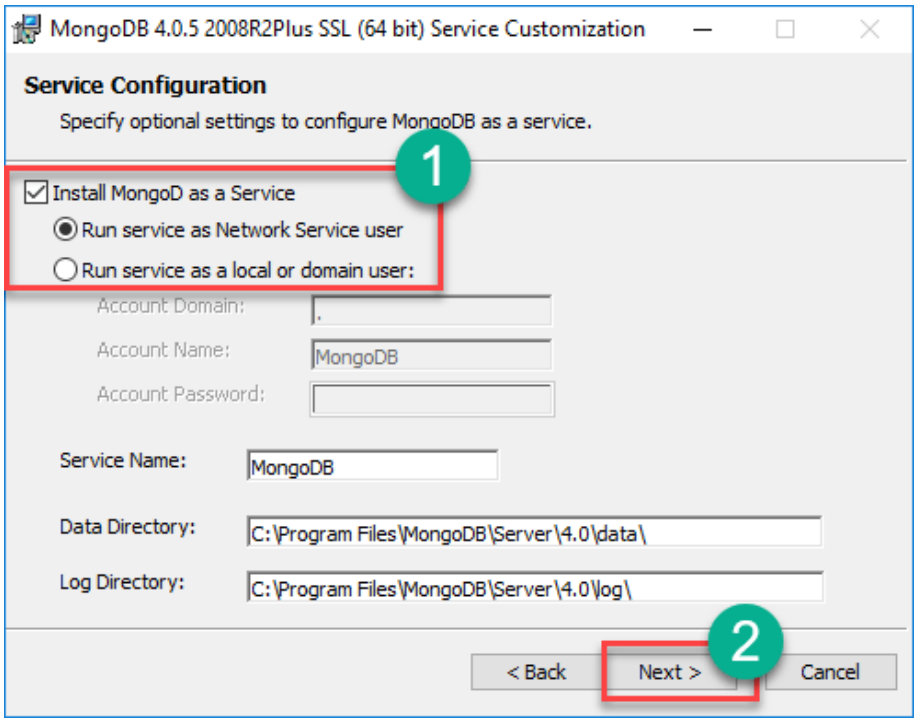
3.Accept the End-User License Agreement and Click Next



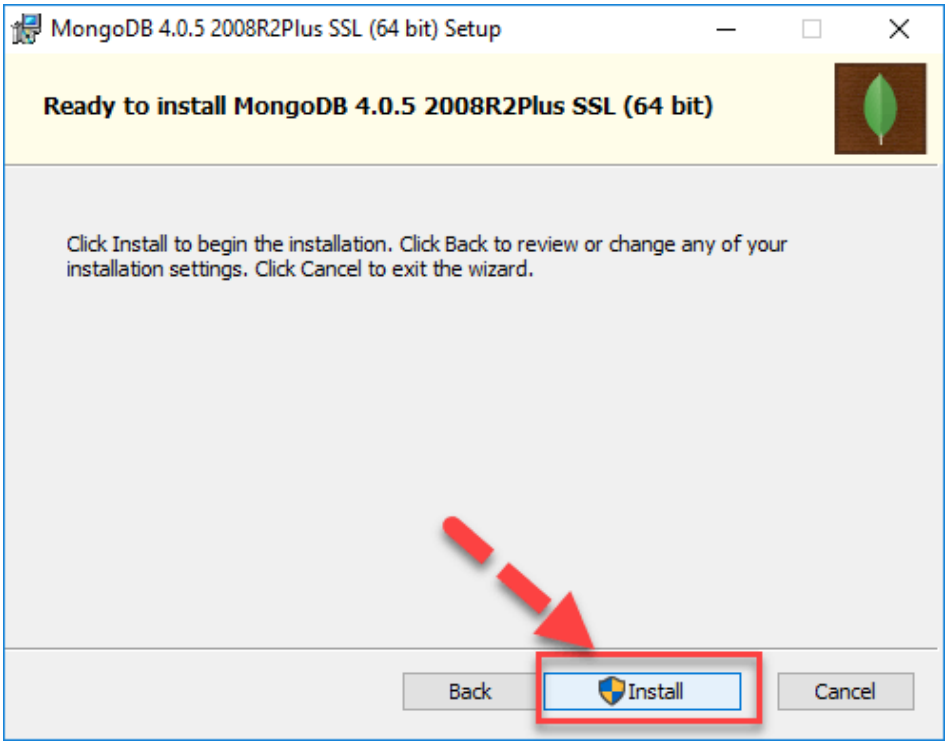
4. Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.



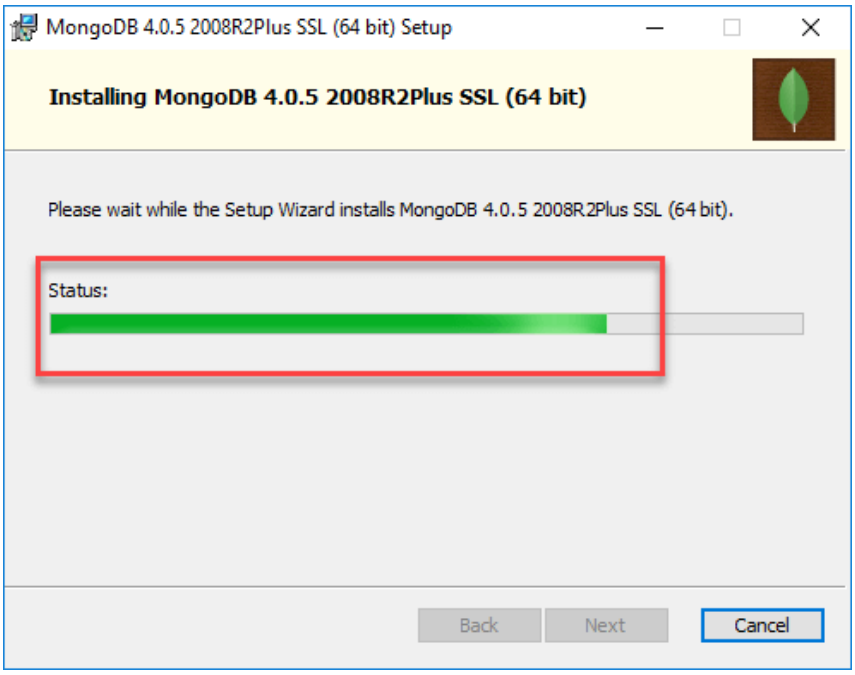
5. Select “Run service as Network Service user”. Make a note of the data directory, we’ll need this later. Then Click Next



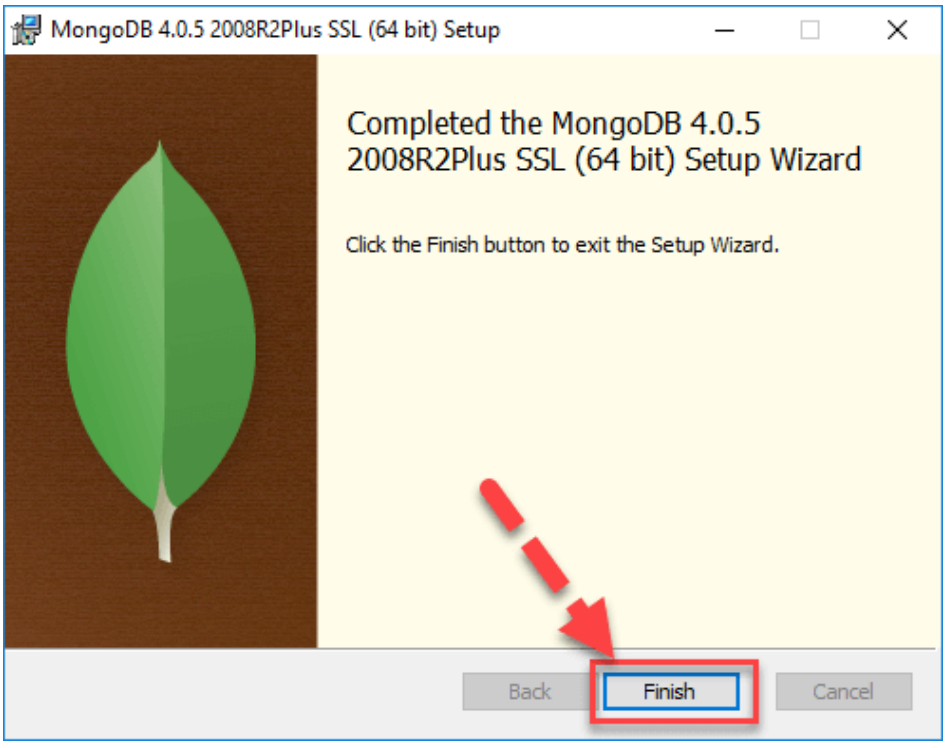
6. Click on the Install button to start the installation.



7. Installation begins. Click next once completed



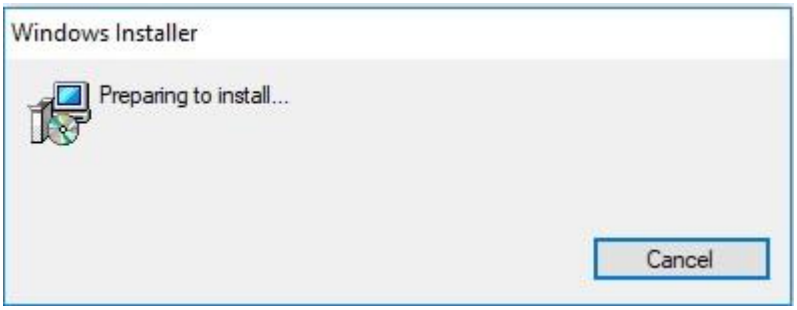
8. Click on the Finish button to complete the installation



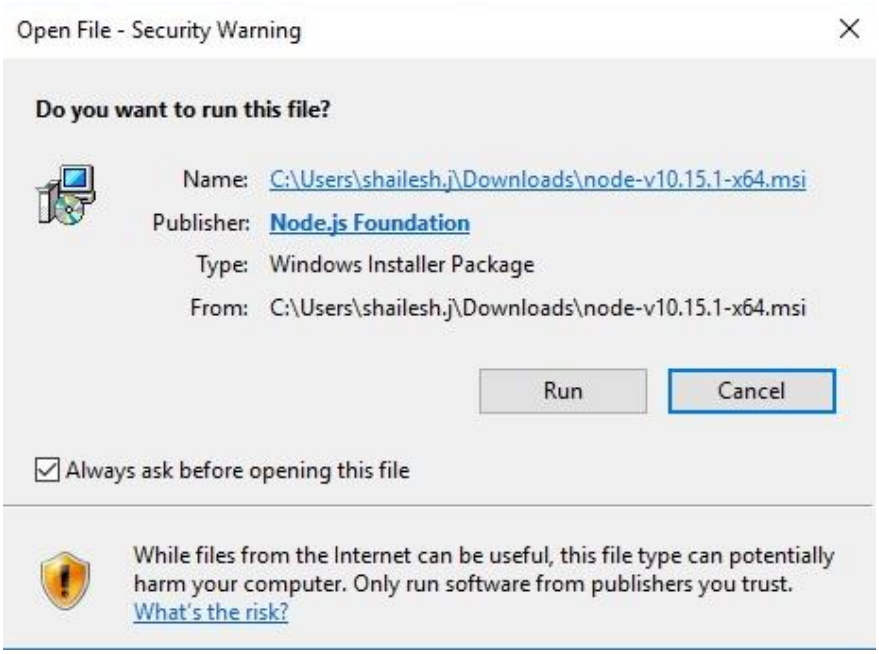
4.4.3 Installing NODEJS:

1. Download the installer file from the download page. Its will be of the file name, something like, node-v10.15.1-x64.msi, where x64 means that the installer's target platform is 64 bit machines and v10.15.1 is the version that will be installed.

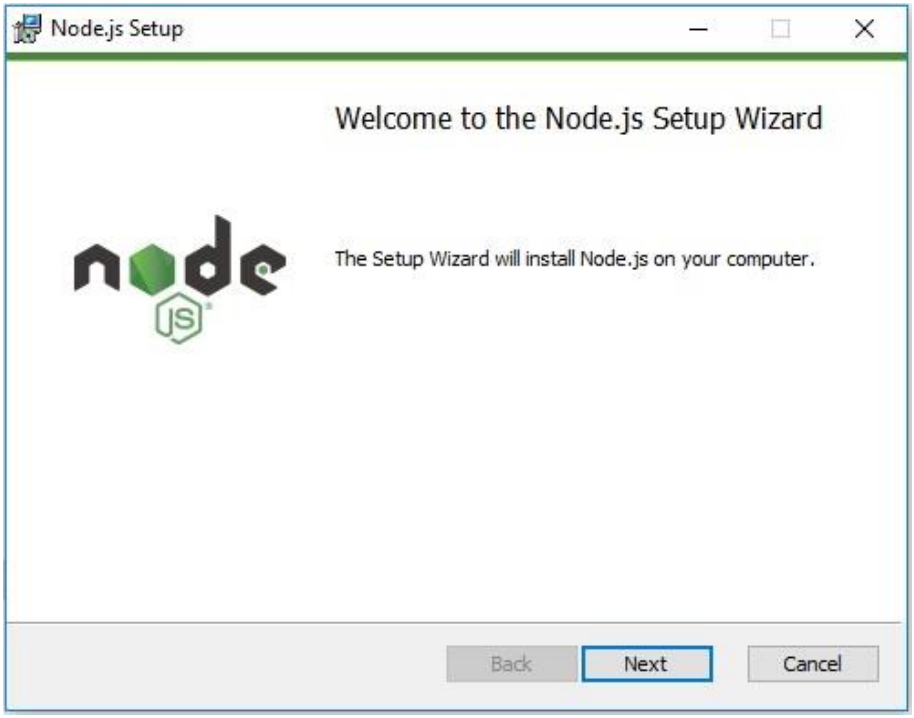
2. Run the installer by double clicking on it. Accept the UAC warning if you see it, which would be something like, "Would you like the installer to make the changes to your system". Click on yes to continue with the installation process.



Click on run to continue. This warning is the standard windows security warning for files downloaded over the internet.

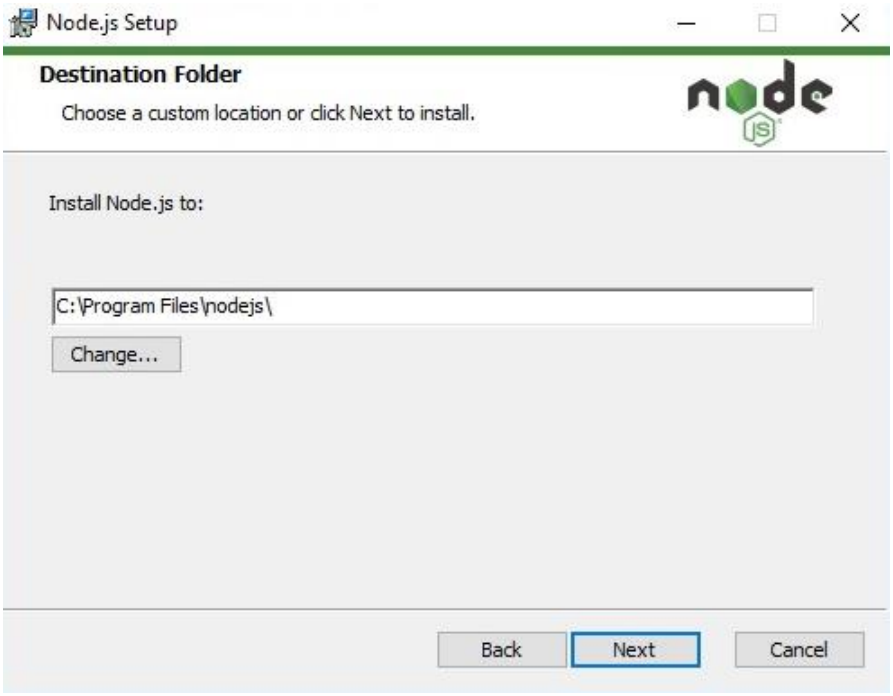


3.Click on next to continue

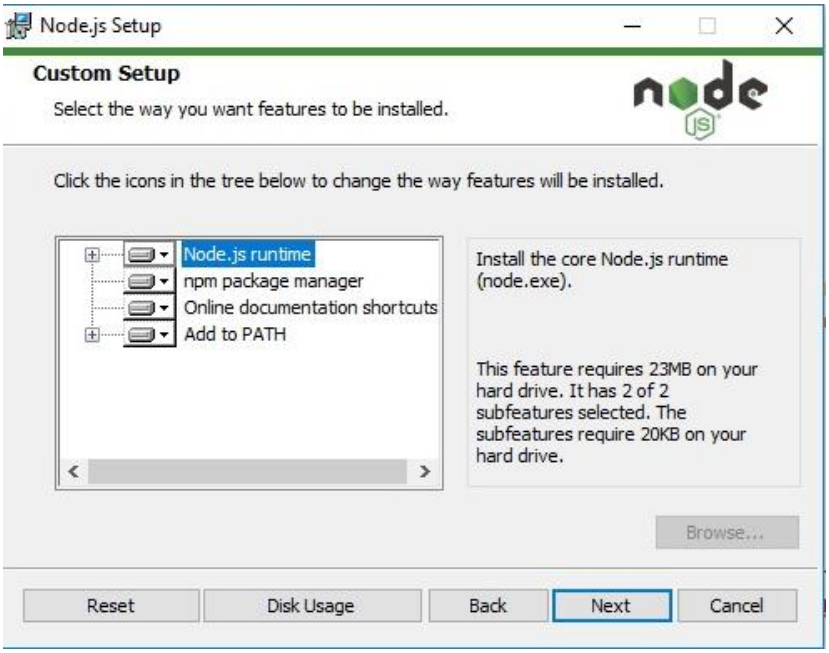


4. Accept the end user licence agreement by clicking on the check box and clicking on next

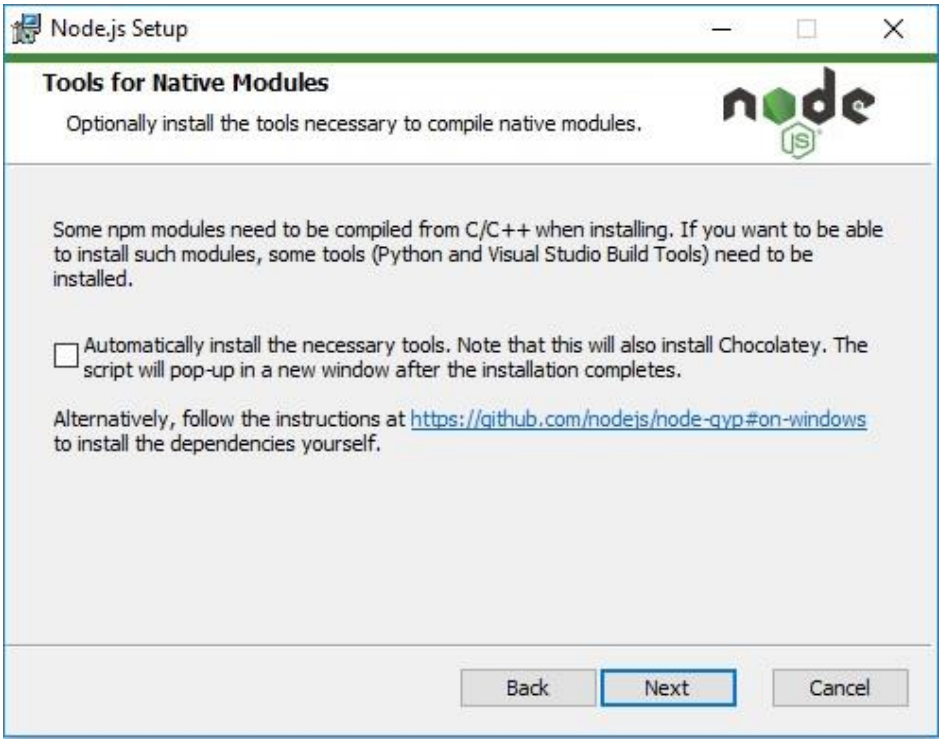
5. You will have to select the destination folder, I normally accept the default. Click on next



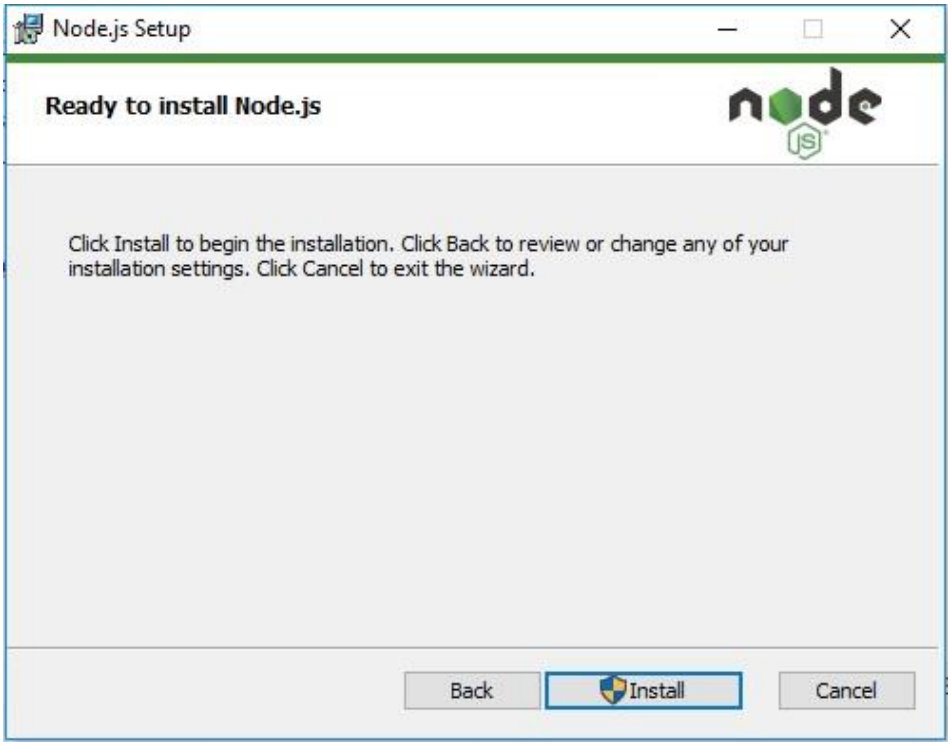
6. Here you will have the option to change the features that will be be default. Accept the defaults by clicking next. I normally accept the defaults.



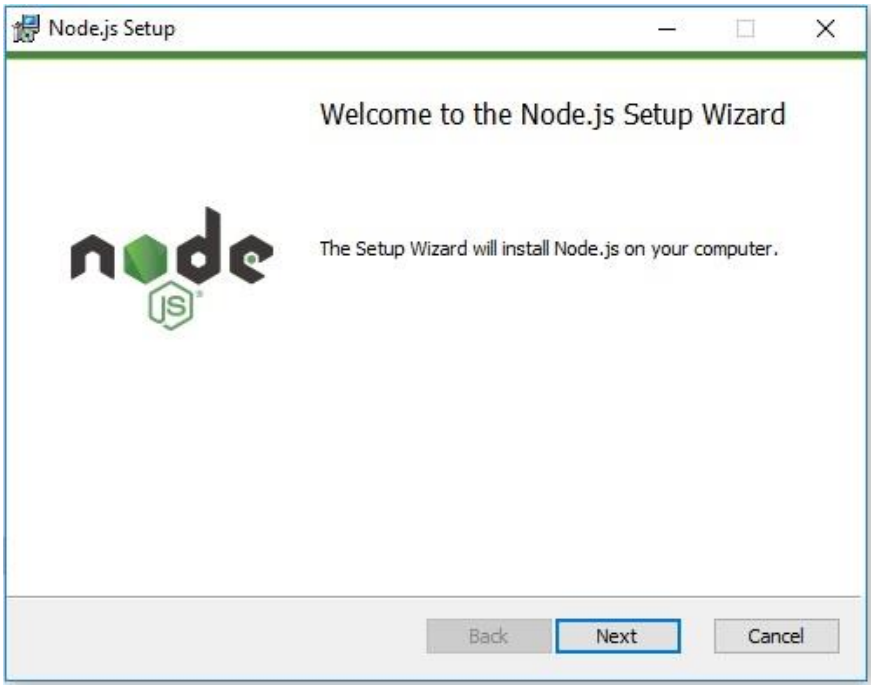
You will be asked if you want to install Tools for Native Modules. I normally leave it as it is Unchecked.



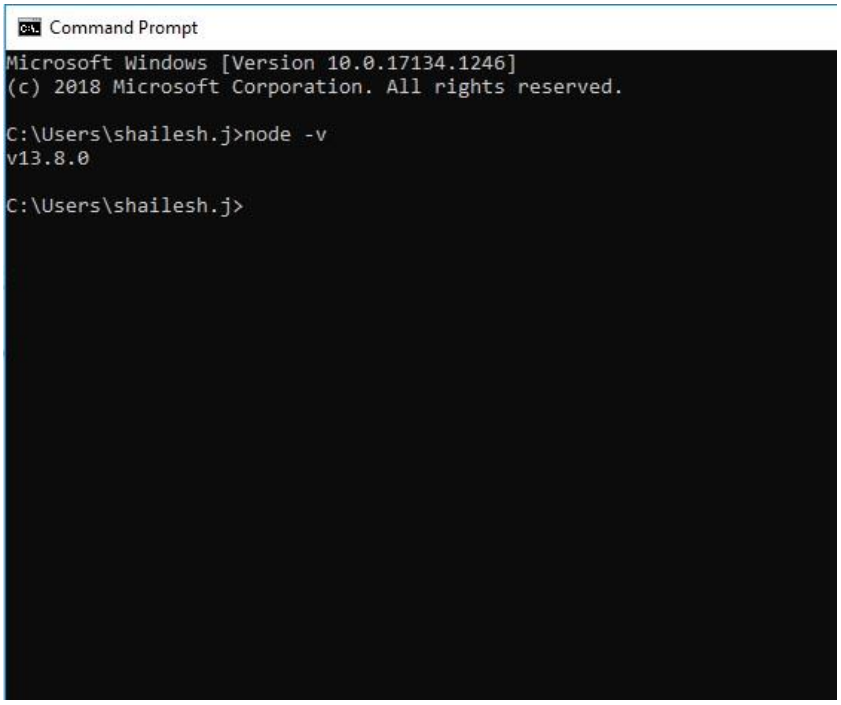
7. Now you will see ready to install window. Click install to proceed.



At the end you will see installation complete message. Click finish to complete the process, This will have Node.js installed on your system and ready to use.

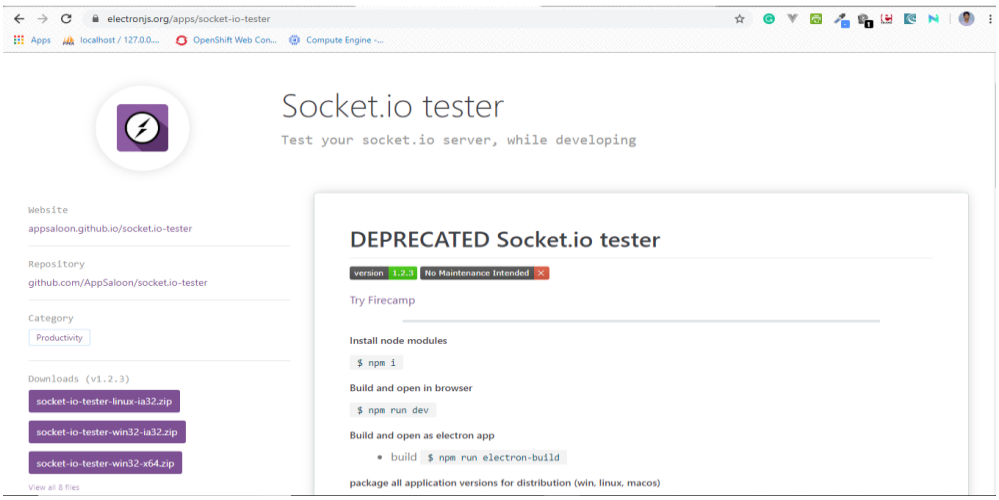


8. Open terminal and use the command node -v to check the version of installed nodejs.



4.4.4 Installing SOCKET TESTER:

1.go to "https://www.electronjs.org/apps/socket-io-tester" official site which is looking like below



2. Download respective Zip file according to OS

3. Unzip the File

4. A file with below icon will be there

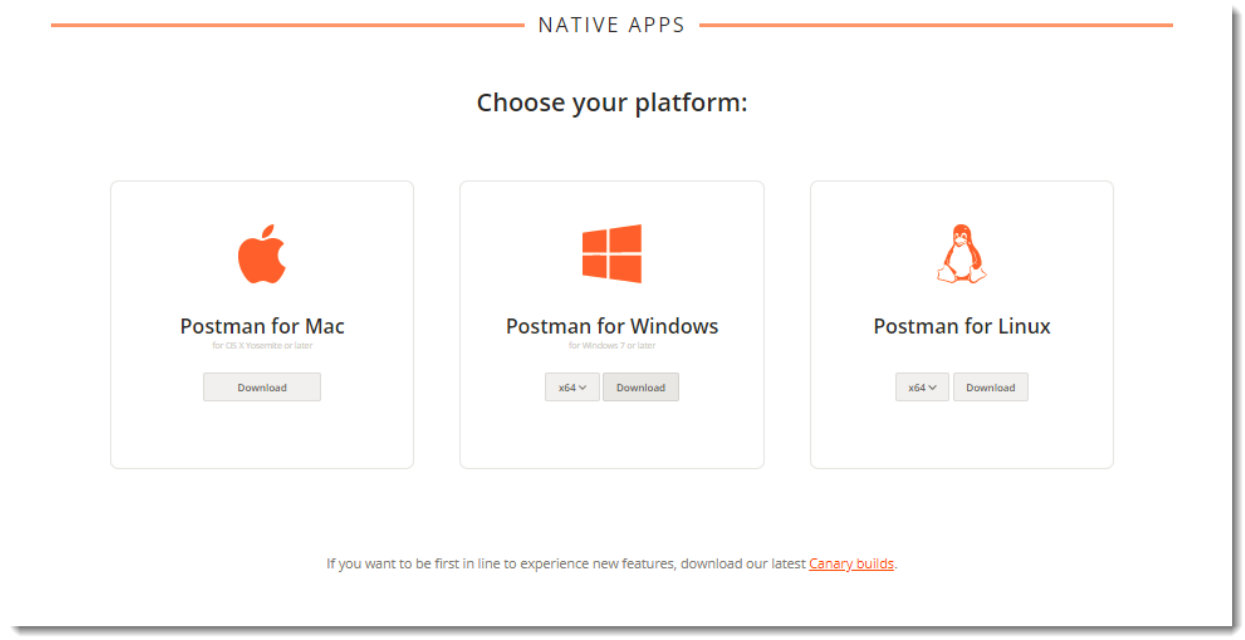
Double click on the file

5. Following window will be opened

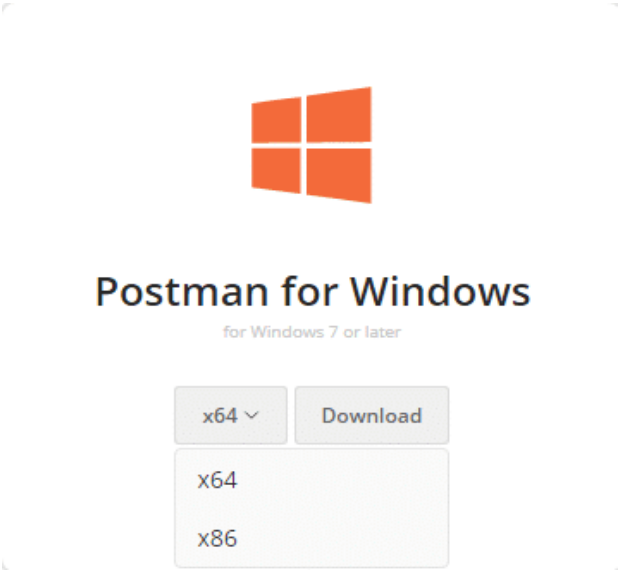

socket.io tester
socket-io-tester Edit Window
New tab
http://www.example.com/socket.io
namespace (optional)
Send Message
Event name
Message arguments
1
Add
Remove
None
String
Clear
Send message
Listen for events
Event name
Pick a color
Add event
Delete all messages

4.4.5 Installing POSTMAN:

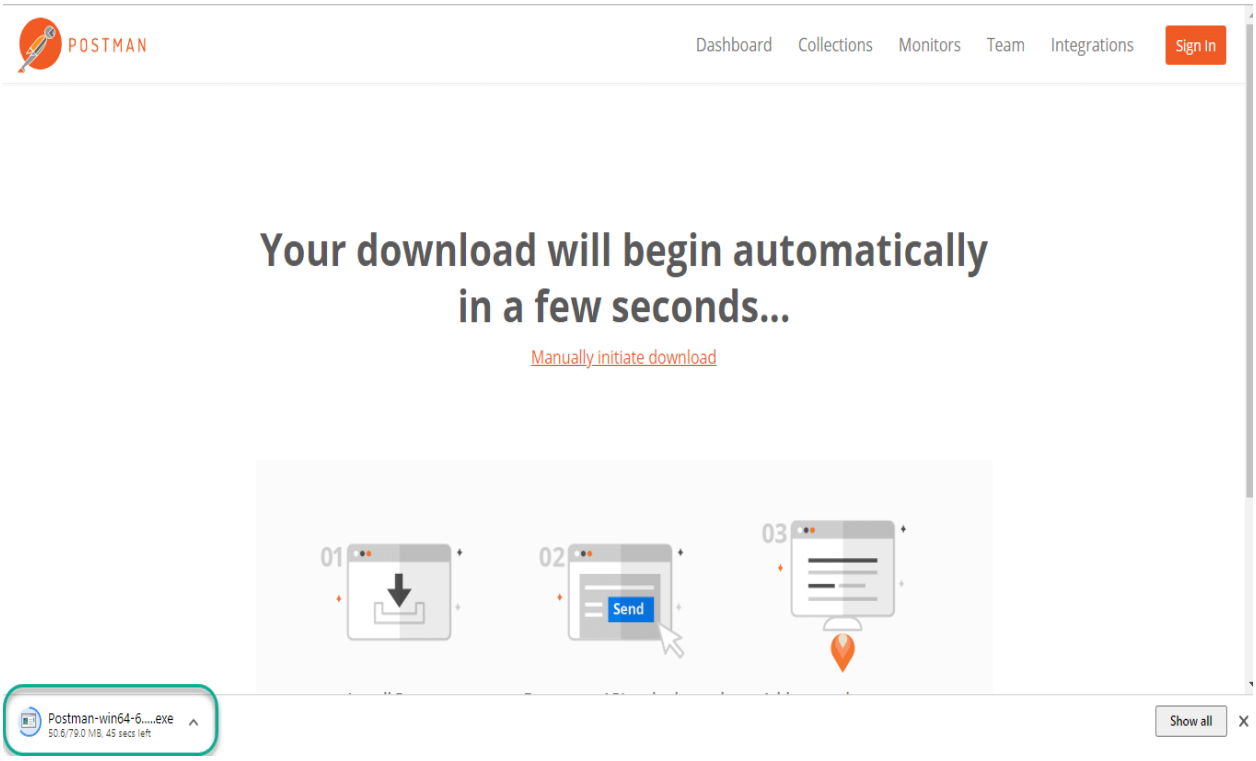
1.To install Postman, go to this linkhttps://www.getpostman.com/appsand click Download for Mac / Windows / Linux depending on your platform. Install Post Man Home Page.



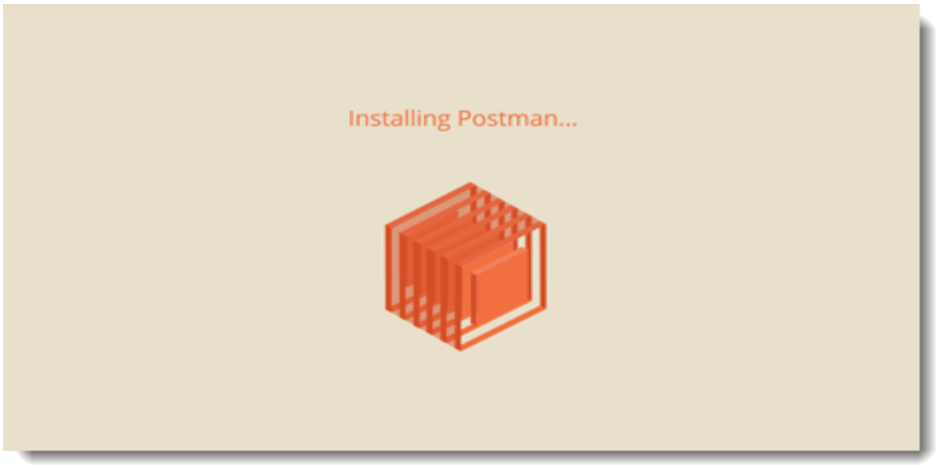
2. Choose the operating system on which you want to install Post Man and click on "Download" button.



3.Once you download the exe file, you will be need to install the application. Since I am using Chrome browser, the downloaded exe will appear at the bottom left of the browser.

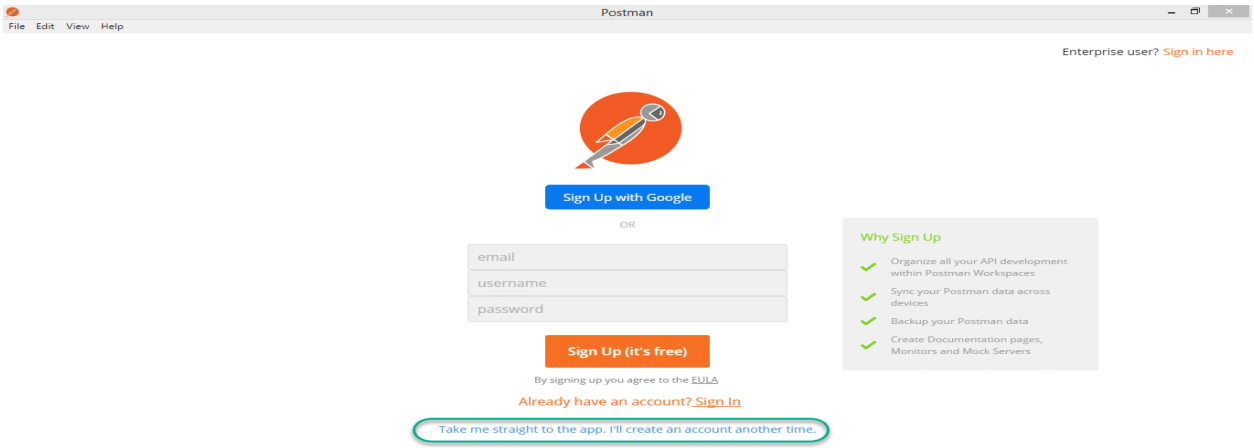


4. Click on the exe file to install it on the system. First it will install the POSTMAN application.

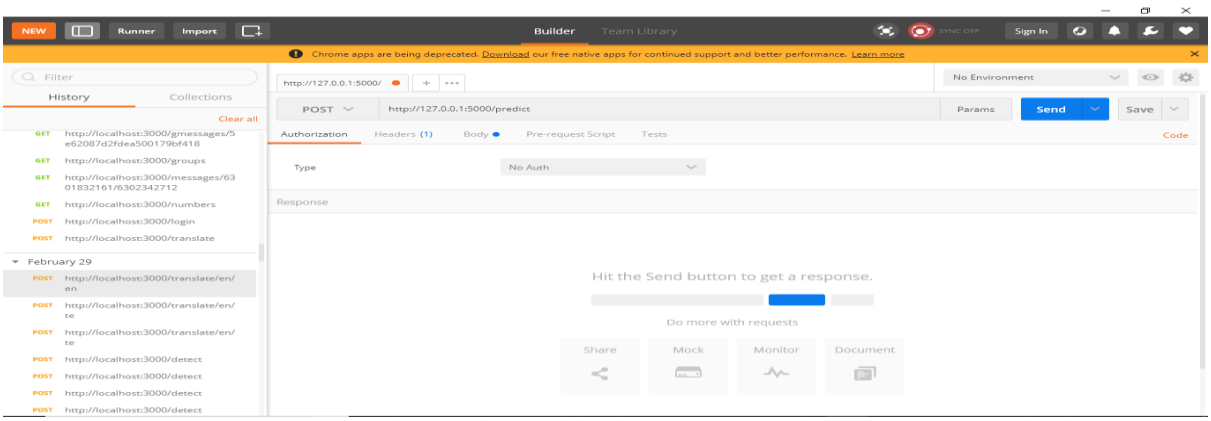


5. There is no further steps for installing. After completion, it will automatically start opening the Post Man tool.

6. Once you have the application window up, click on Take straightto the app. create anaccount another time as highlighted. Alternatively, you can sign up with google but it does not matter at present.



7. After we will get following window



5. TESTING

5.1. Introduction to Testing

Testing is a process of verifying and validating if the developed computer software is correct, complete and has the quality which is acceptable. That means, it is checking if a software system meets specifications and that it fulfills its intended purpose. Hence, it's a universally accepted (and most debated topic) that testing computer software can never be completely established.

VERIFICATION means are we building the product right. Does our software actually achieve its goals without any bugs or gaps?

VALIDATION means are we building the right product. Is our product actually is what we should have built? Does it actually meet the high-level requirements?

Testing Levels:

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Unit Testing:

Unit Testing is done to check whether the individual modules of the source code are working properly. i.e testing each and every unit of the application separately by the developer in the developer's environment. It is AKA Module Testing or Component Testing

Integration Testing:

Integration Testing is the process of testing the connectivity or data transfer between a couple of unit tested modules. It is AKA I&T Testing or String Testing. It is subdivided into Top-Down Approach, Bottom-Up Approach and Sandwich Approach (Combination of Top Down and Bottom Up).

System Testing (end to end testing):

It's a black box testing. Testing the fully integrated application this is also called as end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the users experiences with the application.

Acceptance Testing:

To obtain customer sign–off so that software can be delivered and payments received. Types of acceptance testing are alpha, beta & gamma testing.

Why Test Database?

Data Mapping

In software systems, data often travels back and forth from the UI (user interface) to the backend DB and vice versa. So these are some aspects to watch for: Check whether the fields in the UI/frontend forms are mapped consistently with the corresponding fields in the DB table. Typically, this mapping information is defined in the requirements documents.

Whenever a certain action is performed in the front end of an application, a corresponding CRUD (Create, Retrieve, Update and Delete) action gets invoked at the back end. A tester will have to check if the right action is invoked and whether the invoked action in itself is successful or not.

ACID properties validation

Atomicity, Consistency, Isolation, and Durability. Every transaction a DB performs has to adhere to these four properties.

Atomicity: means that a transaction either fails or passes. This means that even if a single part of the transaction fails- it means that the entire transaction has failed. Usually, this is called the "all-or-nothing" rule.

Consistency: A transaction will always result in a valid state of the DB

Isolation: If there are multiple transactions and they are executed all at once, the result/state of the DB should be the same as if they were executed one after the other.

Durability: Once a transaction is done and committed, no external factors like power loss or crash should be able to change it.

Data integrity: Data integrity is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire lifecycle and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data

For any of the CRUD operations, the updated and most recent values/status of shared data should appear on all the forms and screens. The value should not be updated on one screen and display an older value on another one. When the application is under execution, the end user mainly utilizes the 'CRUD' operations facilitated by the DB Tool.

C: Create – When user 'Save' any new transaction, 'Create' operation is performed.

R: Retrieve – When user 'Search' or 'View' any saved transaction, 'Retrieve' operation

U: Update – When user 'Edit' or 'Modify' an existing record, the 'Update' operation performed.

D: Delete – When a user 'Remove' any record from the system, 'Delete' operation of DB is performed.

Any database operation performed by the end user is always one of the above four. So devise your DB test cases in a way to include checking the data in all the places it appears to see if it is consistently the same.

Business rule conformity. More complexity in databases means more complicated components like relational constraints, triggers, stored procedures, etc. So testers will have to come up with appropriate SQL queries in order to validate these complex objects.

What to Test (Database Testing Checklist)

1) Transactions

2) Database Schemas

3) Triggers

4) Stored Procedures

5) Field constraints

Database Schemas

A database schema is nothing more than a formal definition of how the data is going to be organized inside a DB.

To test it:

- Identify the requirements based on which the database operates. Sample requirements:
- Primary keys to be created before any other fields are created.
- Foreign keys should be completely indexed for easy retrieval and search.
- Field names starting or ending with certain characters.
- Fields with a constraint that certain values can or cannot be inserted.
- Use one of the following methods according to the relevance:
- SQL Query DESC<table name> to validate the schema.
- Regular expressions for validating the names of the individual fields and their val.
- Tools like Schema Crawler.

5.2. Test cases

Test case Id	Input	Description	Expected Result
VS_RS01	No Name No number	Both name and number not specified	Please fill all mandatory fields
VS_RS02	Valid name No number	name is specified without number	Please fill all mandatory fields
VS_RS03	Valid name No number	number is specified without name	Please fill all mandatory fields
VS_RS04	Valid name Valid number	Both name and number are specified	We enter into chats tab

Table 5.1 Login page

Testcase id	Input	Description	Expected Result
VS_RS01	Contain contacts	After Successful login. Contacts with Talkie Account will be displayed	It displays the list of contacts
VS_RS02	No contacts	None of the Contacts in device have Talkie Account	No contacts in your mobile have Talkie accounts
VS_RS03	Create group	Group is created by group name and members added to it	Group created successfully refresh if not updated

Table 5.2 Home page

Testcase id	Input	Description	Expected Result
VS_RS01	Group creation with no name	Group is created without specifying name	Group creation failed
VS_RS02	Choosing file	If we choose file upload	Displays files
VS_RS03	Choosing file Termination	If we did not select the file	File Upload cancelled
VS_RS04	Online	When we switch from Offline to Online	You are now Online
VS_RS05	Offline	When we switch from Online to Offline	You are now Offline

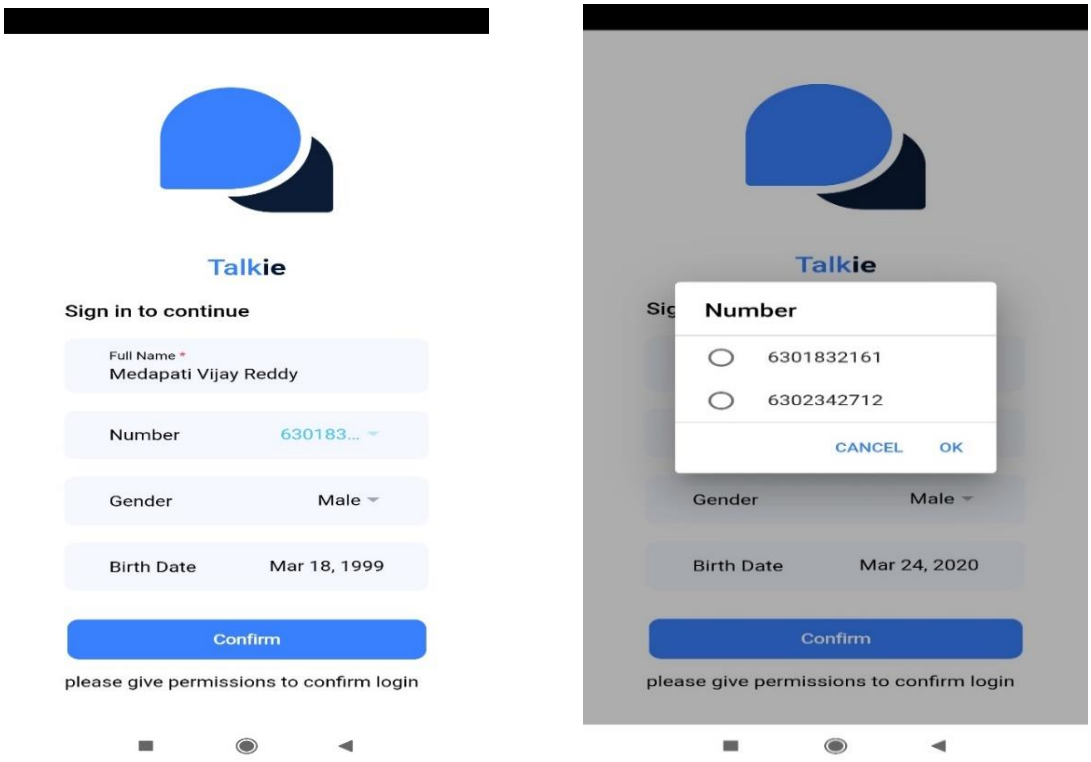
VS_RS06	Choose file	If we select image or video It shows Preview after Uploading	File uploaded Message and Preview
VS_RS07	one or more languages Selected for Translation	message can be translated to one or more languages at a time	It translates the message into options selected
VS_RS08	No language Selected for Translation	If we does not select language	Please select at least one language
VS_RS09	text speak mode on	Tapping on a text message	It read the message with voice
VS_RS10	Text speak mode off	Tapping on a text message	No voice will be encountered
VS_RS11	Text Recognition on image	Tap on image and Select recognize text Option	It displays the url's and text in image
VS_RS12	Text Recognition on Text Message	Tap on image and Select recognize text Option	You can recognize text in image files only
VS_RS13	Applying filter URLS Option on text message with No URLs	Tap on message and select filter urls Option	No urls found
VS_RS14	Applying filter URLS Option on text message with www.google.com	Tap on message and select filter urls Option	www.google.com is ham
VS_RS15	Applying filter URLS Option on text message with www.amazee.com	Tap on message and select filter urls Option	www.amazee.com is spam

VS_RS16	Send message as damn hell	If we send bad words Message will not reach to the receiver	It shows spam symbol and message cannot be sent
VS_RS17	Tagging Others message	Tap on message and select Tag Option	Alert you can tag messages sent by you only
VS_RS18	Send message as how are u	Tag as important , confidential and personal	Change colours at receiver end
VS_RS19	Posting Poll with no question and options	Posting poll with No options	Alert message describing question with atleast 2 options need to post the poll
VS_RS20	Posting Poll with question and 1 option	Posting poll with single option	Alert message describing question with atleast 2 options need to post the poll
VS_RS21	Neutral message	If we receive message as hi, hello having polarity as zero	It displays in blue colour
VR_RS22	Positive message	If we receive message as amazing one having polarity > 5	It displays in green colour
VR_RS23	Negative message	If we receive message as you are bad boy , irresponsible, hell having polarity < 5	It displays in orange colour

Table 5.3 Chat page

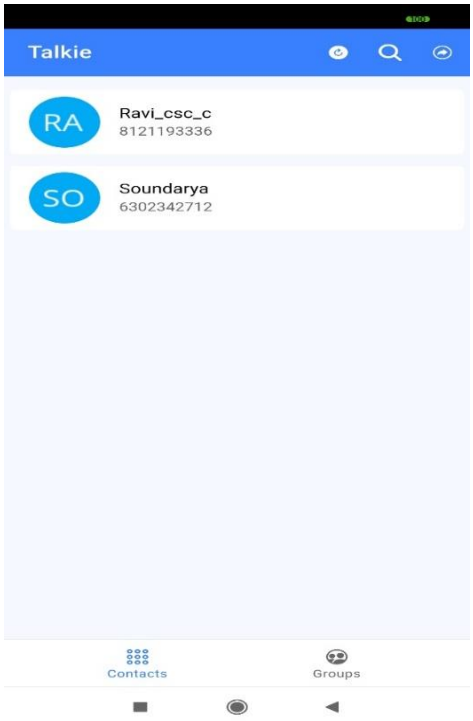
6.SCREENS & REPORTS

6.1. Screenshots:



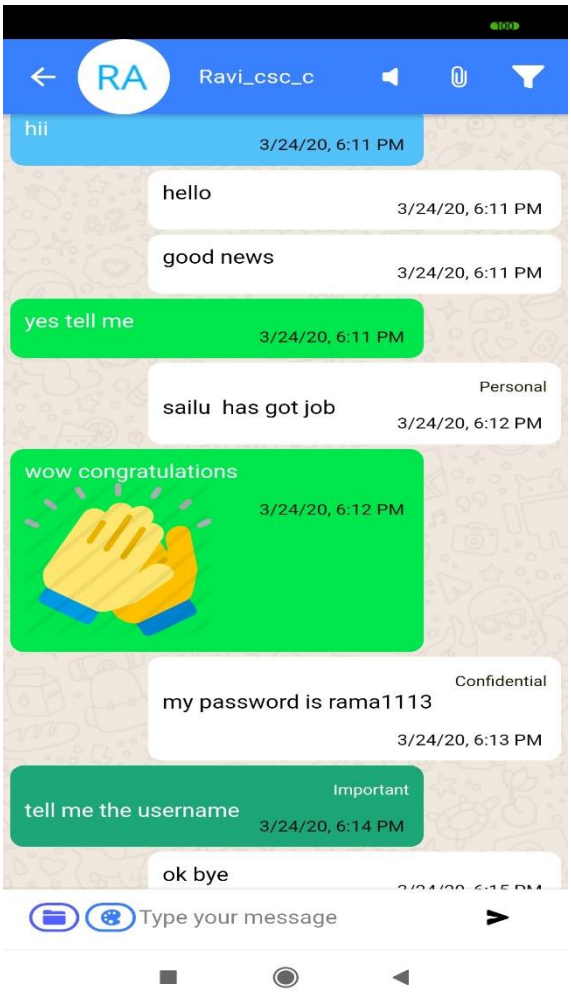
Screen 6.1 Login Page

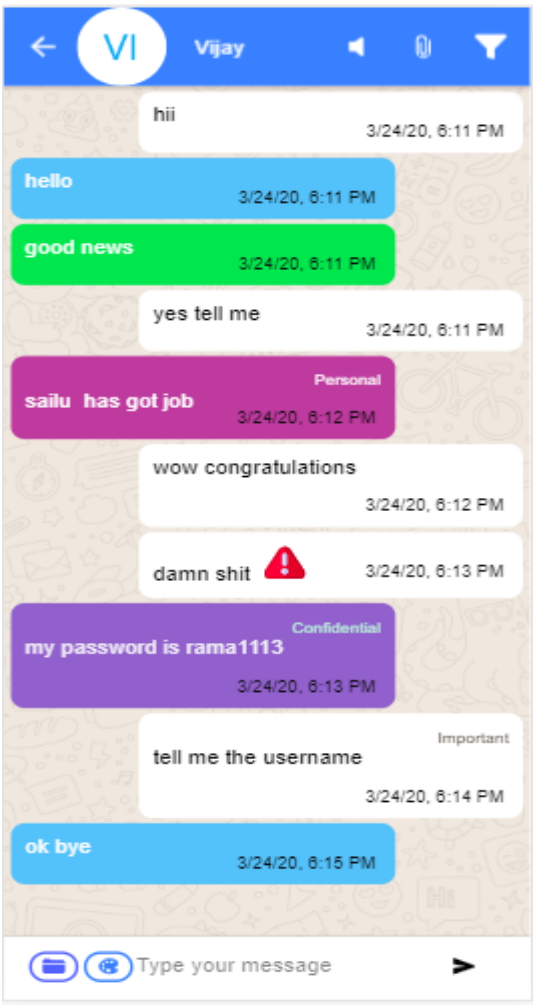
User need to fill details & select One Sim in the Device with which he/she want to login



Screen 6.2 Contacts Page

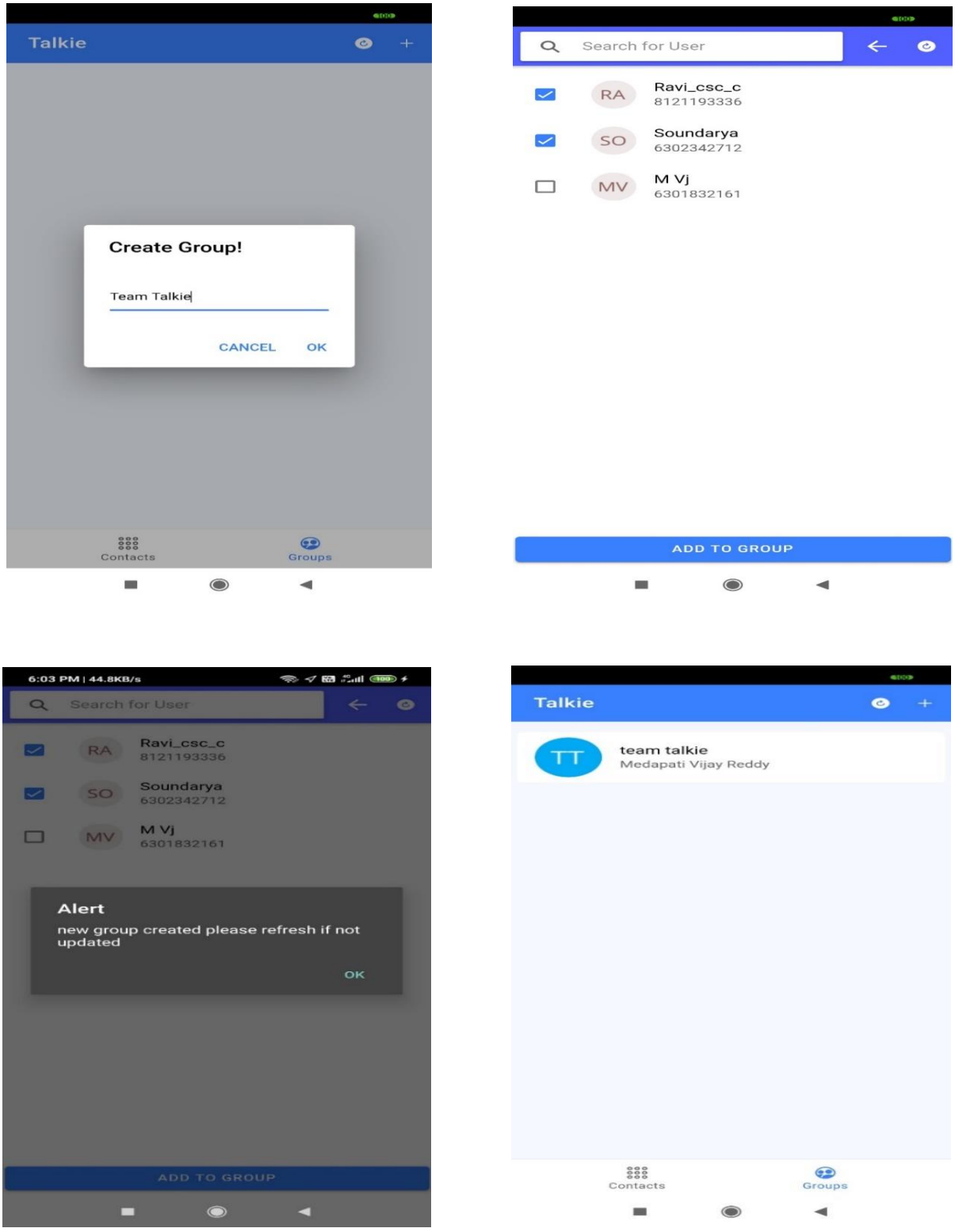
Contacts In Device having Talkie Account Already will be Displayed After Successful Login





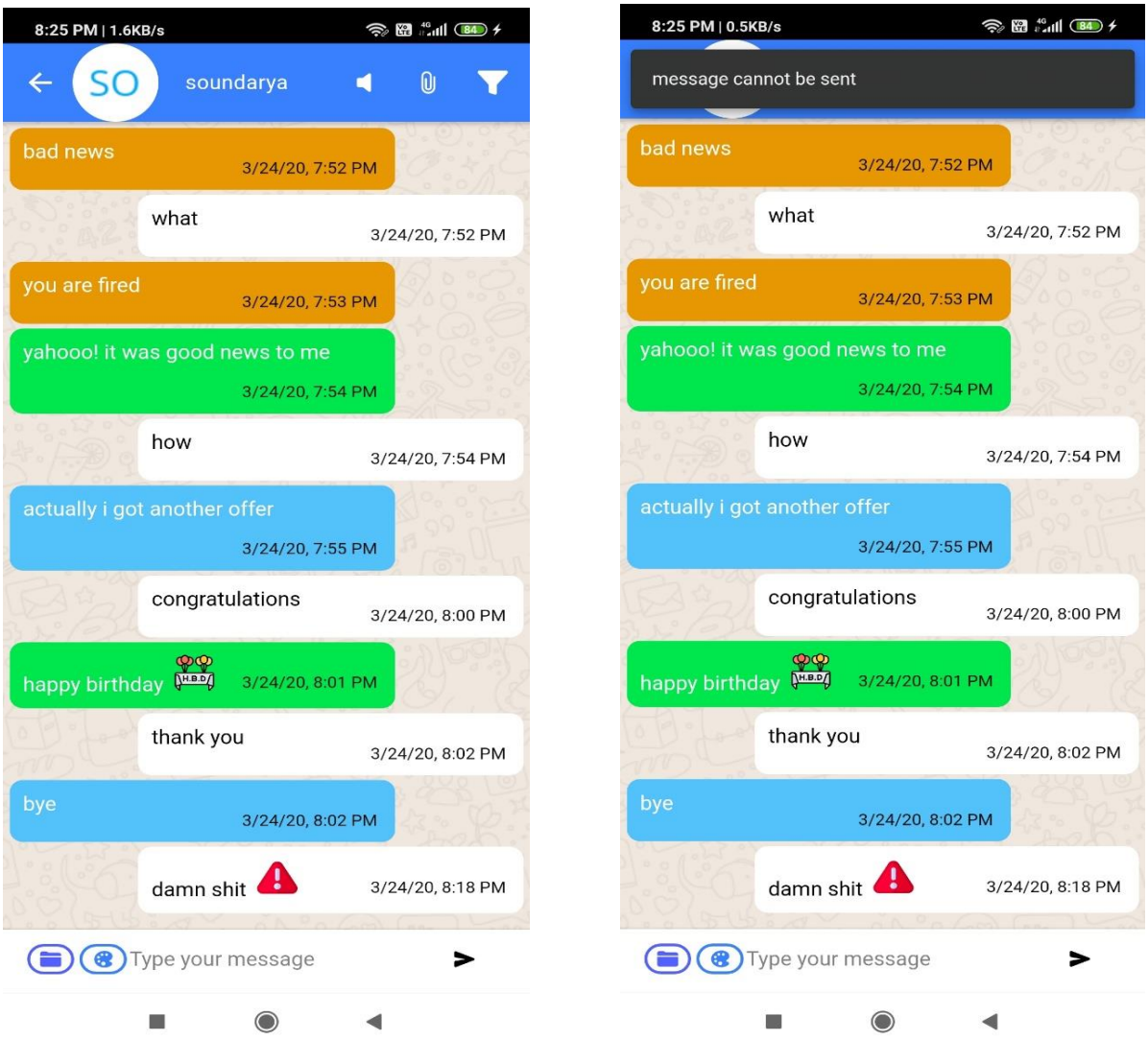
Screen 6.3 Chat Interface

Above Screen describes how Sender and Receiver Chat Interface is displayed



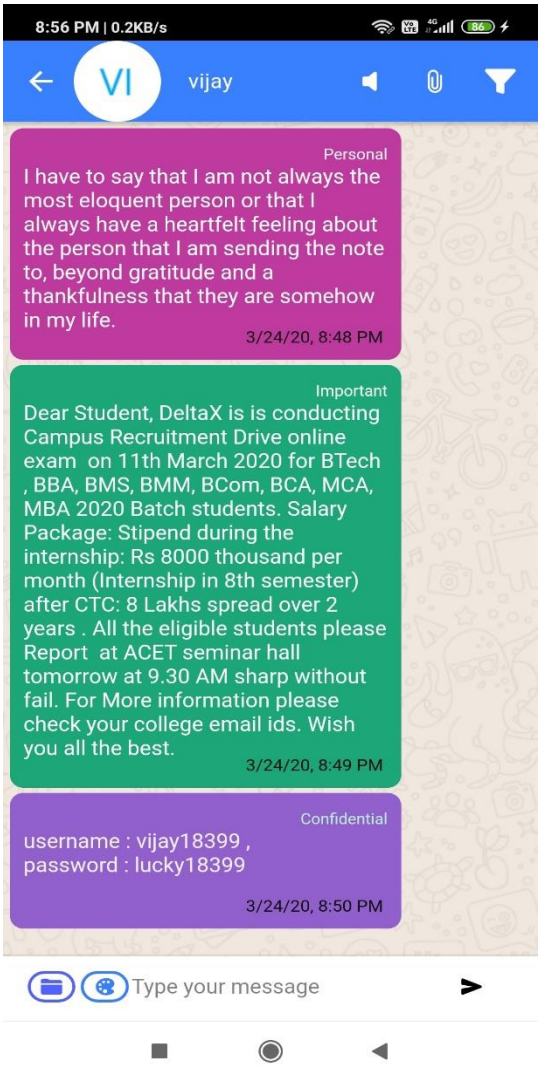
Screen 6.4 Group Creation

Above Screens describes how User can Create a group and add members to It at the time of Group Creation



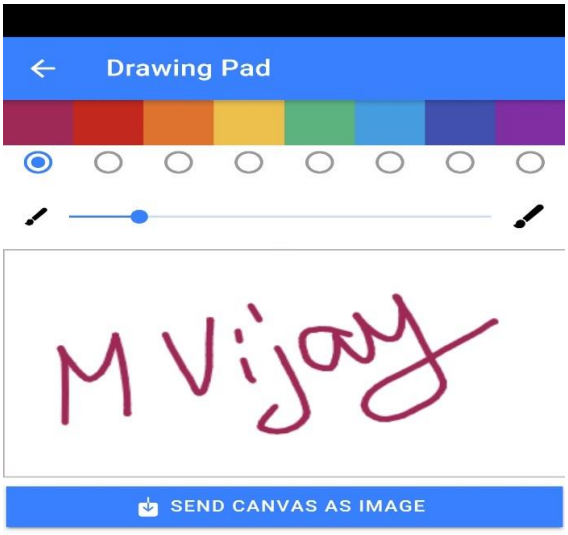
Screen 6.5 Abusive message restriction

Above Screens describes how Restricted message are Shown to Sender



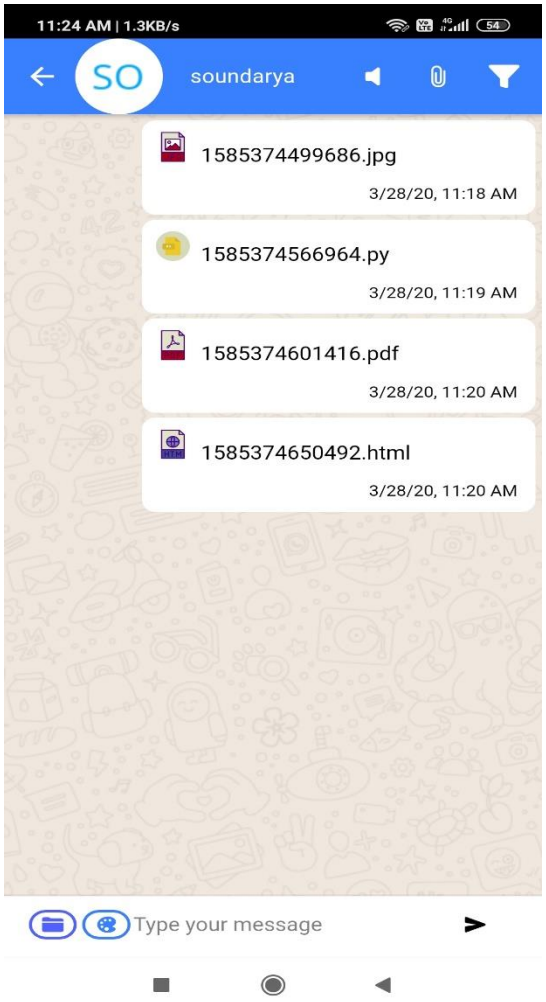
Screen 6.6 Message Tagging

Above Screen describes how Tagged Messages are Displayed in terms of colour Personal, Important, Confidential is displayed as purple, green, violet Colored bubble respectively



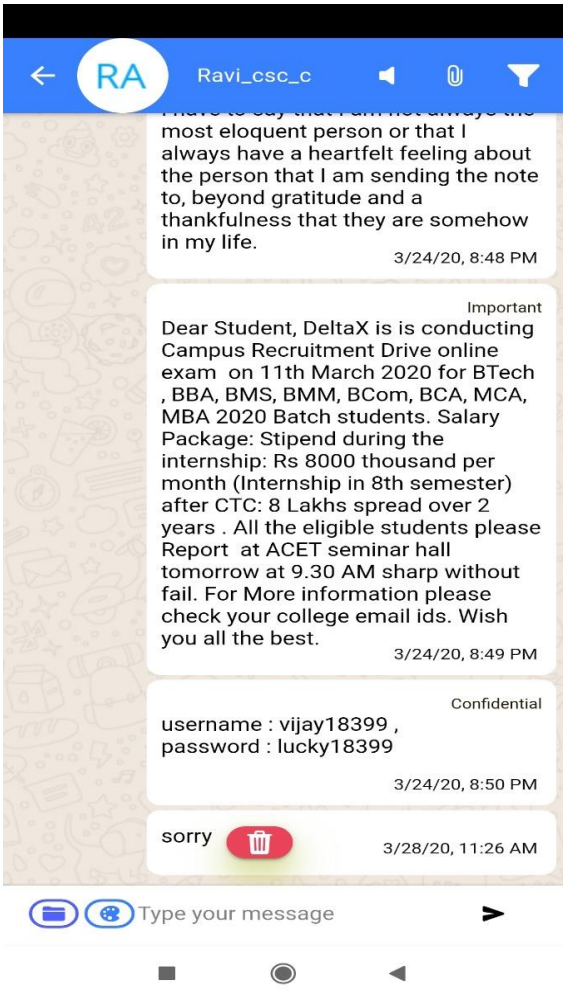
Screen 6.7 Drawing Pad

Above Screen is Drawing Pad having multiple coloured and different sized brushes, User can draw Canvas and send them as a PNG image to Others



Screen 6.8 File Type Indication

Above Screen describes User can Send files of any type and those are Displayed with their respective extension icons, So User can understand what file he/she received

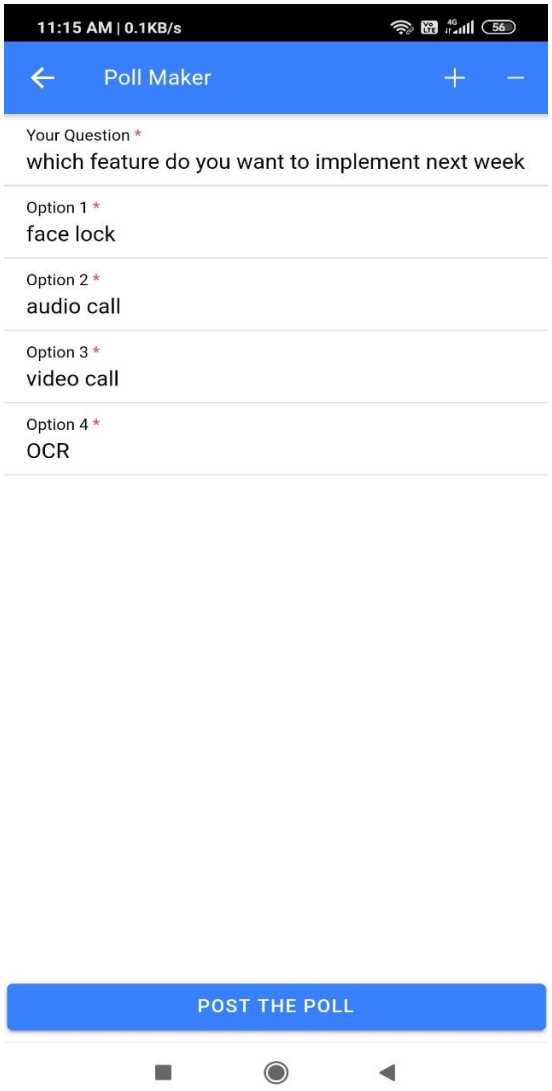


Screen 6.9 Delete for Every One [icon displayed for 10 minutes]

In Above Screen, Delete icon displayed on chat bubble is used delete message for everyone which is displayed for 10 minutes which counts from message sending time

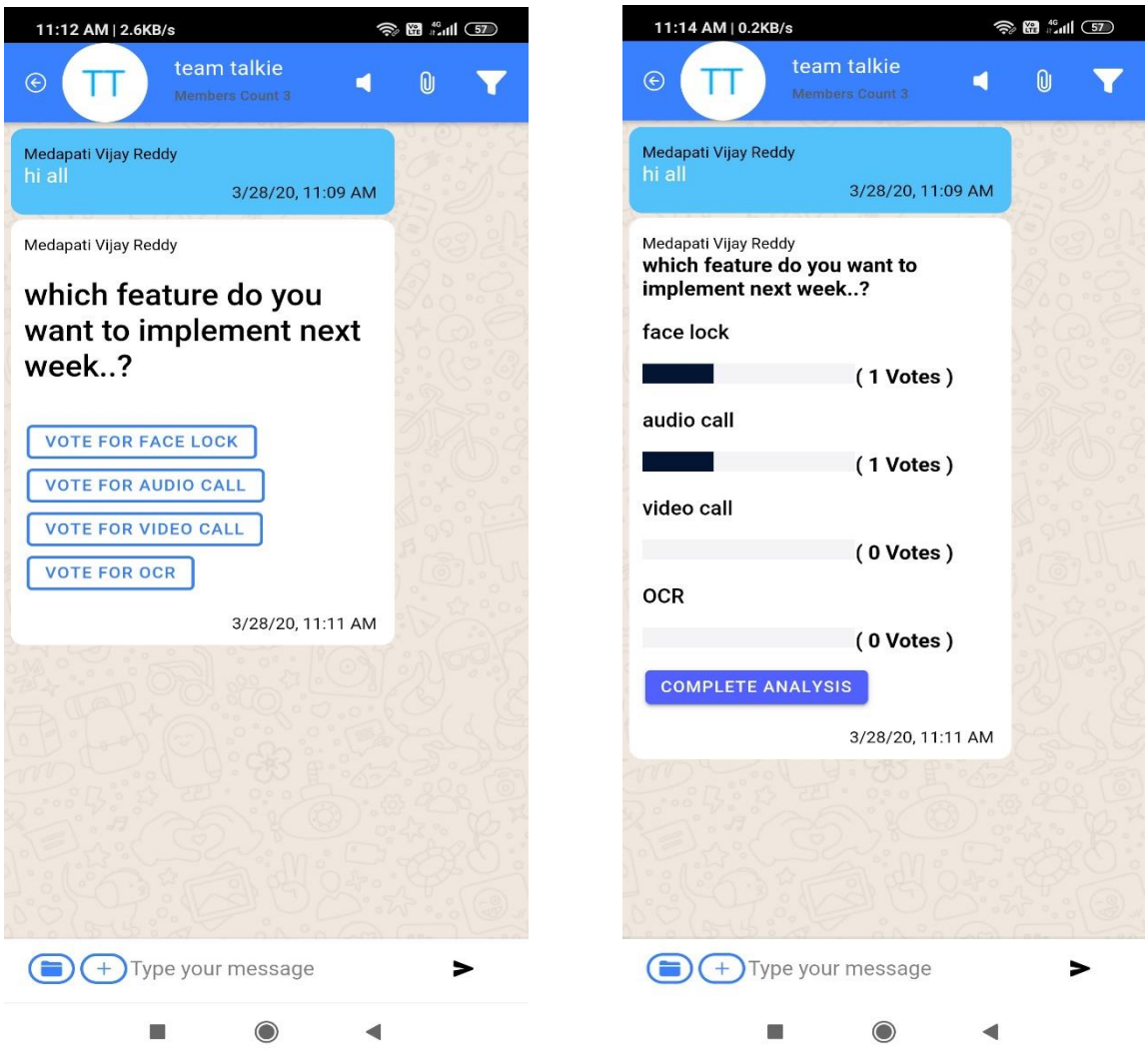
Screen 6.10 Capture Media

Above options in Screen are displayed when attachment icon clicked which are used to capture image, audio, video and send them to others



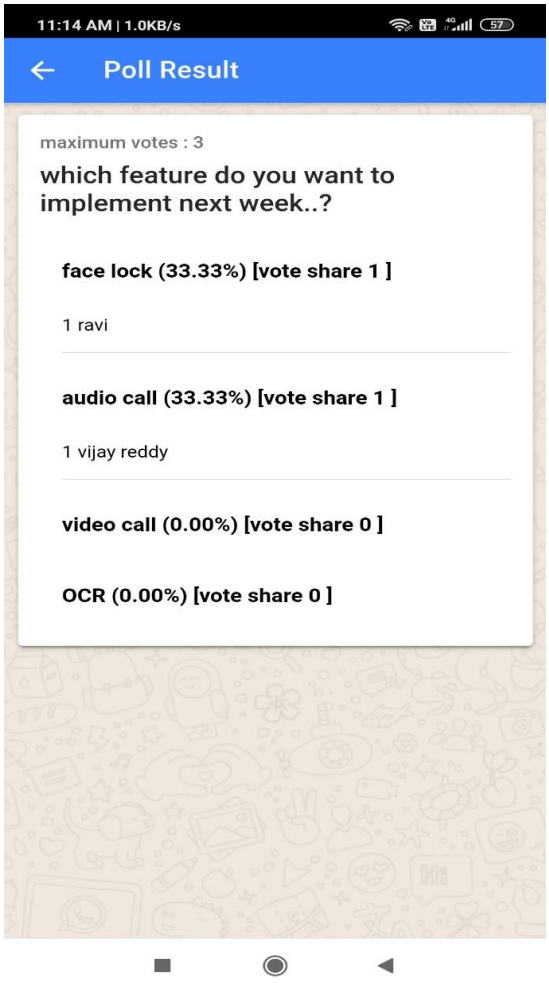
Screen 6.11 Poll maker

In Above Screen, Poll Maker used for posting a question and multiple options in the group



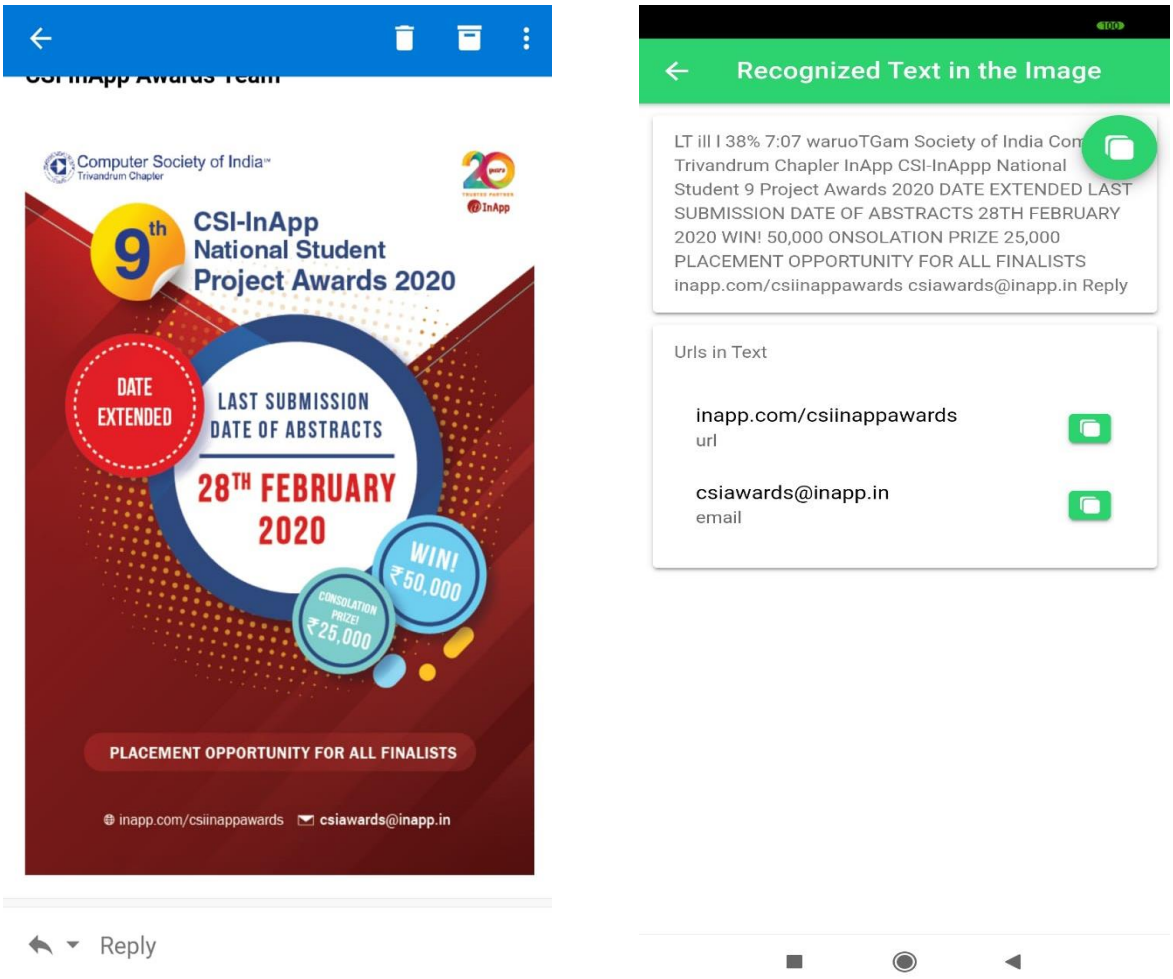
Screen 6.12 Poll & Result

In Above Screen shows how Poll is displayed if a user didn't vote in the poll and if he voted in a poll.If the user voted he/she can see the result..



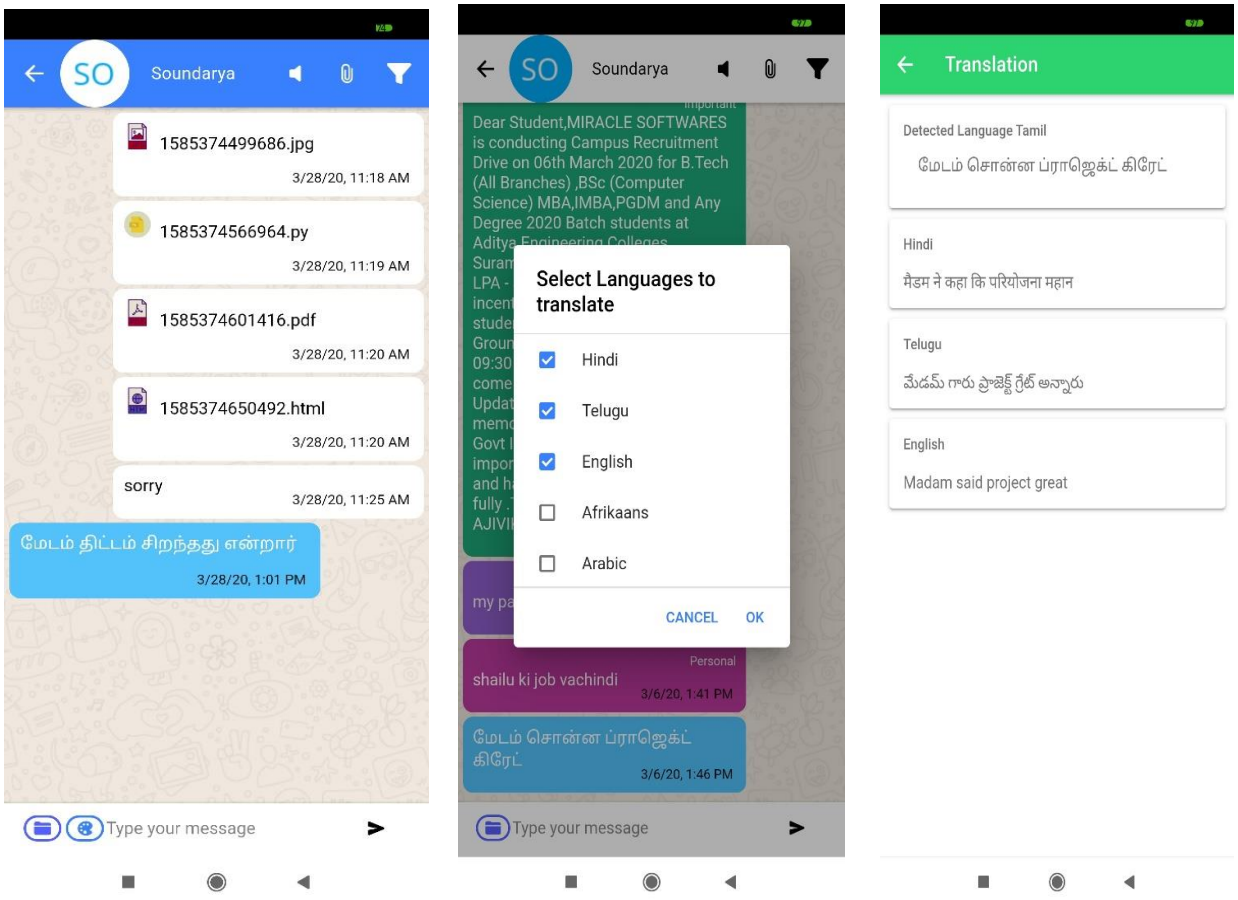
Screen 6.13 Complete Poll Report

In Above Screen Complete Poll Result is displayed which include various things like total eligible voters, vote share for each option and who voted for each option



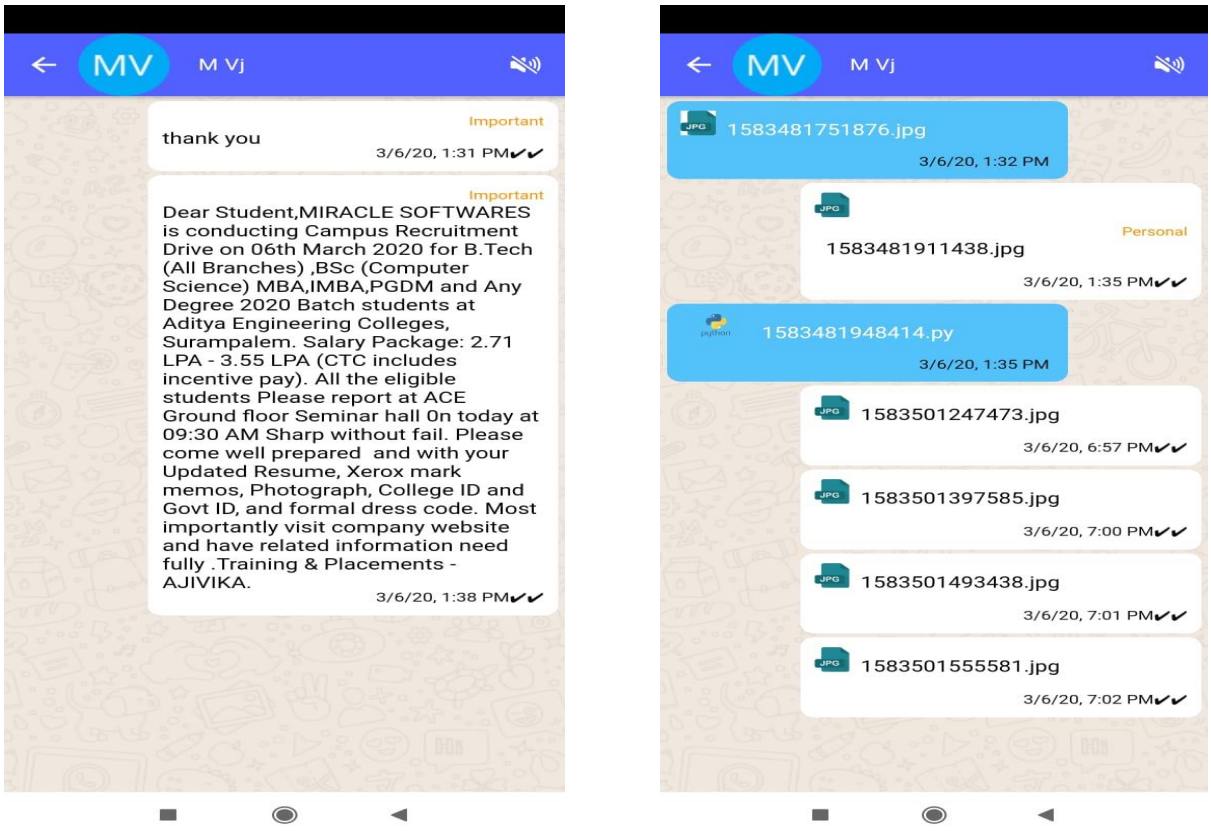
Screen 6.14 OCR Result Page

In Above Screen How OCR Result is Shown, It Detects all text data in the image and detects all URLs in text too.



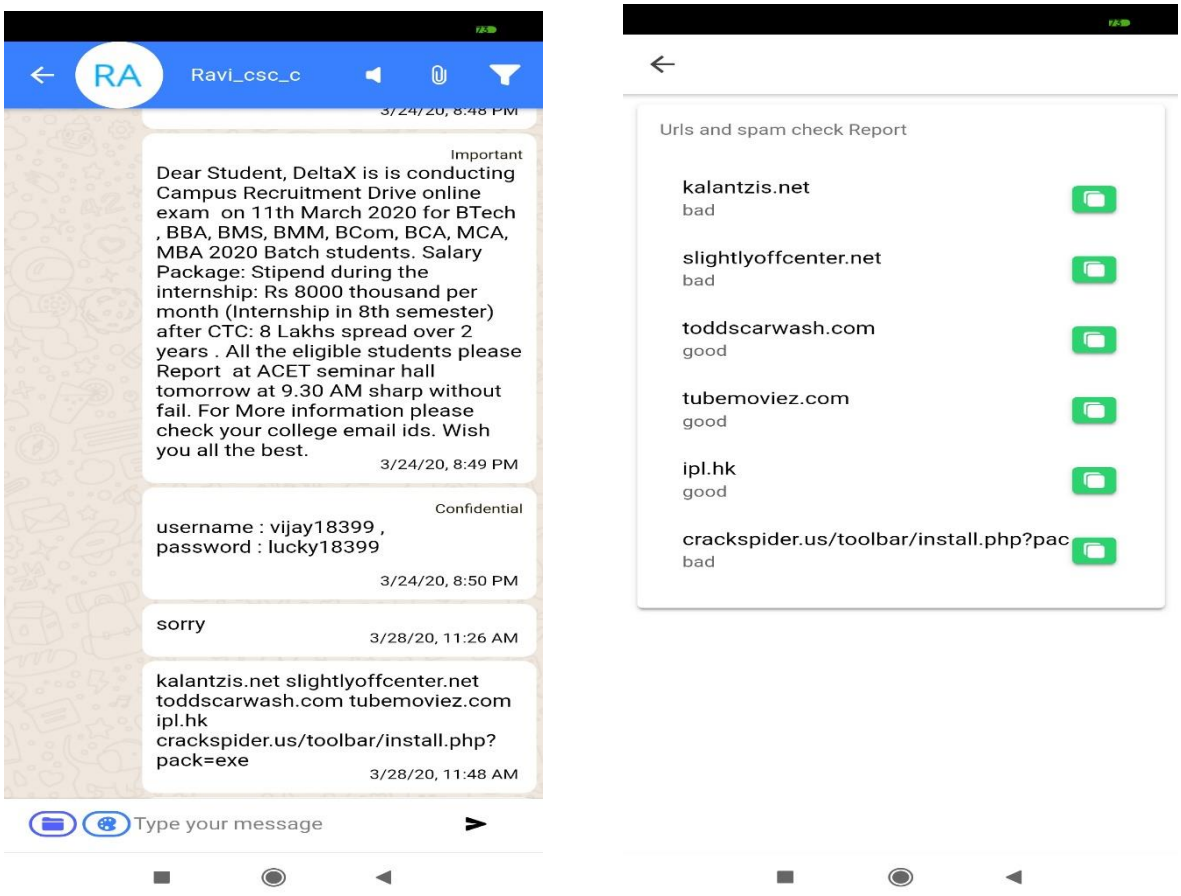
Screen 6.15 Translation

In Above Screen How Translation Result is Shown, It gives the option to which language message text need to be translated, After user selection result will be shown on another page



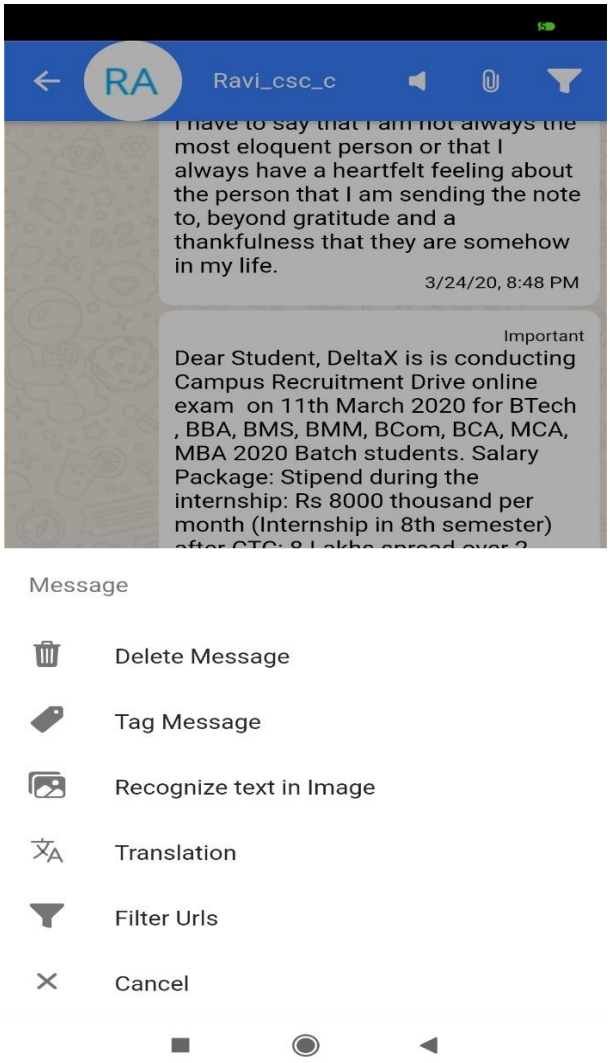
Screen 6.16 Message filter

In Above Screen How Message Filter Result is Shown, It gives the option to which Type of the message needs to be filtered whether he/she wants Media, Important, Personal, Confidential Messages, After user, selection result will be shown in other Screen



Screen 6.17 Spam Report

In Above Screen How User can perform URL classification, It Detects all URLs in the message text and Classifies them to good or bad shows then in other Screen



Screen 6.18 Message Options

In Above Screen We have All Options that can be performed on a text message

6.2 Reports

Test case Id	Input	Description	Expected Result	Reports
VS_RS01	No Name No number	Both name and number not specified	Please fill all mandatory fields	Pass
VS_RS02	Valid name No number	name is specified without number	Please fill all mandatory fields	Pass
VS_RS03	No name Valid number	number is specified without name	Please fill all mandatory fields	Pass
VS_RS04	Valid name Valid number	Both name and number are specified	We enter into chats tab	Pass

Table 6.1 Login page

Test case id	Input	Description	Expected Result	Reports
VS_RS01	Contain contacts	After Successful login. Contacts with Talkie Account will be displayed	It displays the list of contacts	Pass
VS_RS02	No contacts	None of the Contacts in device have Talkie Account	No contacts in your mobile have Talkie accounts	Pass
VS_RS03	Create group	Group is created by giving group name and adding members to it	Group created successfully refresh if not updated	Pass

Table 6.2 Home Page

Testcase id	Input	Description	Expected Result	Reports
VS_RS01	Group creation with no name	Group creation initiated without specifying name	Group creation failed	Pass
VS_RS02	Choosing file	If we choose file upload	Displays files	Pass
VS_RS03	Choosing file Termination	If we did not select the file	File Upload cancelled	Pass
VS_RS04	Online	When we switch from Offline to Online	You are now Online	Pass
VS_RS05	Offline	When we switch from Online to Offline	You are now Offline	Pass
VS_RS06	Choose file	If we select image or video It shows Preview after Uploading	File uploaded Message and Preview	Pass
VS_RS07	one or more languages Selected for Translation	message can be translated to one or more languages at a time	It translates the message to the selected languages	Pass
VS_RS08	No language Selected for Translation	If we does not select language	Please select atleast one language	Pass
VS_RS09	Text to speech mode on	Tapping on a text message	It read the message with voice	Pass

VS_RS10	Text to speech mode off	Tapping on a text message	No voice will be encountered	Pass
VS_RS11	Text Recognition on image	Tap on image and Select recognize text Option	It displays the url's and text in image	Pass
VS_RS12	Text Recognition on Text Message	Tap on image and Select recognize text Option	You can recognize text in image files only	Pass
VS_RS13	Applying filter URLS Option on text message with No URLs	Tap on message and select filter urls Option	No urls found	Pass
VS_RS14	Applying filter URLS Option on text message with www.google.com	Tap on message and select filter urls Option	www.google.com is ham	Pass
VS_RS15	Applying filter URLS Option on text message with www.amazee.com	Tap on message and select filter urls Option	www.amazee.com is spam	Pass
VS_RS16	Send message as damn hell	If we send bad words Message will not reach to the receiver	It shows spam symbol and message cannot be sent	Pass
VS_RS17	Tagging Others message	Tap on message and select Tag Option	Alert you can tag messages sent by you only	Pass
VS_RS18	Send message as how are u	Tag as important , confidential and personal	Change colours at receiver end	Pass

VS_RS19	Posting Poll with no question and options	Posting poll with No options	Alert message describing question with atleast 2 options need to post the poll	Pass
VS_RS20	Posting Poll with question and 1 option	Posting poll with single option	Alert message describing question with atleast 2 options need to post the poll	Pass
VS_RS21	Neutral message	If we receive message as hi, hello having polarity as zero	It displays in blue colour	Pass
VR_RS22	Positive message	If we receive message as amazing one having polarity > 5	It displays in green colour	Pass
VR_RS23	Negative message	If we receive message as you are bad boy , irresponsible, hell having polarity < 5	It displays in orange colour	Pass

Table 6.3 Chat page

7. CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

Chatting is a very common used application among users. General users use instant messaging services to communicate with other individual users. In our project, we have provided with many enhanced features for a chat application. The features like sentiment analysis, language translator, spam detection, message tagging, text recognition in image plays a major role.

The Chat App protects user and device from various things like Abusive messages, Phishing Attacks and Malicious URLs.Each message received can be translated to more than 60 languages using 3rd party Service namely Microsoft Azure Cognitive service. The chat application is so aimed that the people could have a better experience of chatting. It has the potential to attract more and more users to interact and connect.

Scope

- Sentiment Analysis With Multiple Language Support
- Instant URL Classification
- Fingerprint Lock

8. BIBLIOGRAPHY

References:

[1] Chatting Application with Real Time Translation Nikhil Chaudhari 1, SushmaShinkar 2, Priyanka Pagare Bachelor of Engineering, Computer Engineering Department, Sanghavi college of Engineering , Maharashtra, India

[2] Sentiment Analysis of Chat Application Swanand Joshi, Amey Ruikar Department of Information and Technology Pune Institute of Computer and Technology Pune, India

Websites referred:

- https://devdactic.com/ionic-realtime-socket-io/
- https://ionicframework.com/
- https://docs.mongodb.com/