

Yello- An All-In-One Instant Messaging Application

Submitted in partial fulfilment of requirements

For the degree of

Bachelor in Information Technology

by

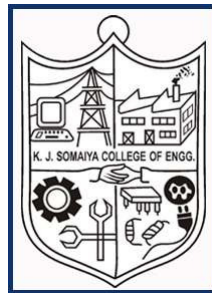
Ajay Mohandas Roll No:1214001

Rajankumar Bhimprasad Roll No:1214002

Vijay Yadav Roll No:1214050

Pradeep Bagrao Roll No:1214067

Prof. Dipti Pawade



Department of Information Technology

K.J.Somaiya College of Engineering, Mumbai-400077

(Autonomous College Affiliated to University of Mumbai)

Batch 2012-2016

University of Mumbai

Yello-An All-In-One Instant Messaging Application

Submitted in partial fulfilment of requirements

For the degree of
Bachelor in Engineering

by

Ajay Mohandas Roll No:1214001

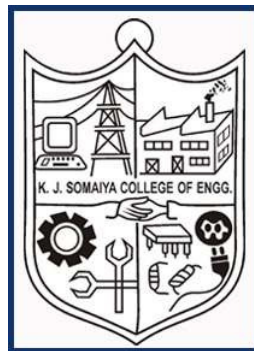
Rajankumar Bhimprasad Roll No:1214002

Vijay Yadav Roll No:1214050

Pradeep Bagrao Roll No:1214067

Under the guidance of

Prof. Dipti Pawade



Department of Computer Engineering

K.J.Somaiya College of Engineering

Vidyavihar, Mumbai-400077

Batch 2012-2016

K.J.Somaiya College of Engineering, Mumbai-400077

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the Dissertation entitled "Yello-An All-In-One Instant Messaging Application" is bona fide record of the dissertation work done by

1. Student name . Ajay Mohandas
2. Student name . Rajankumar Bhimprasad
3. Student name . Vijay Yadav
4. Student name . Pradeep Bagrao

in the year 2015-16 under the guidance of Prof. Dipti Pawade of Department of Information Technology in partial fulfillment of requirement for the Bachelors degree in Information Technology of University of Mumbai.

Guide :Prof.Dipti Pawade

Head Of Department: Prof. Sangeeta Nagpure

Principal Prof. Shubha Pandit

Date : 16-11-2015

Place : Mumbai - 400077

K.J.Somaiya College of Engineering, Mumbai-400077

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this project report entitled "Yello-An All-In-One Instant Messaging Application" is bona fide record of project work done by

1. Student name . Ajay Mohandas
2. Student name . Rajankumar Bhimprasad
3. Student name . Vijay Yadav
4. Student name . Pradeep Bagrao

This project is approved for the award of Bachelors Degree in Information Technology of University of Mumbai.

External Examiner

Internal Examiner

Date :

Place : Mumbai - 400077

Abstract

In today's world Instant Messaging applications like WhatsApp, Line, Viber have increased the social interaction among people. The purpose of such application is to keep friends and families connected no matter how far distance between them. Now we can chat with our near and dear one's easily and can see them even if they are million miles away. Some of the application gives us the provision of sending images, videos to others or call them and all this is possible by just clicking one button. This type of applications are cheaper to use and also user friendly.

Inspite of advent of such applications people are demanding for those applications that are efficient and can ease their work. Thus we have planned to create an instant messaging application similar to Whastapp but with added features. Some of the features added are built-in AppLock, offline messaging, reminder notification. The other notable features added are Integration of our application with other applications ,secure communication. Our main goal is to develop an application which is user friendly and efficient to use. By using our application user doesn't have to install any other application.

Keywords: *AppLock, Instant Messaging, Offline messaging, Secure communication, Integration of apps*

Contents

Abstract	i
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Problem Definition	1
1.2 Motivation of the thesis	1
1.3 Scope of the thesis	2
1.4 Salient contribution	2
1.5 Organization of the Thesis	2
2 Literature Survey	4
3 Software Project Management Plan	6
3.1 Introduction	6
3.1.1 Project Overview	6
3.1.2 Project Deliverables	7
3.2 Project Organization	8

3.2.1	Software Process Model	8
3.2.2	Roles and Responsibilities	9
3.2.3	Tools and Techniques	10
3.3	Project Management Plan	10
3.3.1	Tasks	10
3.3.2	Time Chart	13
4	Software Requirements Specification	15
4.1	Introduction	15
4.1.1	Product Overview	15
4.1.2	Intended Audience and Document Overview	16
4.2	Specific Requirements	16
4.2.1	User Interfaces	16
4.2.2	Software Interfaces	16
4.2.3	Communications Protocols	16
4.3	Functional Requirement	17
4.4	Non-Functional Requirement	17
4.4.1	Performance Requirements	17
4.4.2	Standards Compliance	18
4.4.3	Reliability	18
4.4.4	Availability	18
4.4.5	Maintainability	18
4.4.6	Portability	18
5	Software Design Description	19
5.1	Introduction	19

5.1.1	Design Overview	19
6	Implementation	30
6.1	Technologies to be used	30
6.1.1	Hardware Requirements	30
6.1.2	Software Requirements	30
6.1.3	Languages to be used	31
6.2	Algorihtms used	31
6.2.1	Registration and OTP verification	31
6.2.2	Add Friend List	31
6.2.3	App Lock	32
6.2.4	Chat Functionality	32
6.2.5	Offline Messaging	32
6.2.6	Integrating Web Links	33
6.2.7	Reminder Notification	33
6.2.8	Secure Messaging	34
6.3	Pseudocode	34
7	Software Test Document	36
7.1	Introduction	36
7.1.1	System Overview	36
7.1.2	Test Approach	37
7.2	Test Plan	37
7.2.1	Features not to be Tested	37
7.2.2	Testing Tools and Environment	38
7.3	Test Cases	38

8	Results and discussion	45
8.1	Different modules in application	45
8.2	Output of each modules	45
9	Conclusion and Scope for future work	52
9.1	Conclusions	52
9.2	Scope for future work	53
	References	55

List of Figures

3.1	Project Deliverables	7
3.2	Prototype Model	8
3.3	Roles and Responsibility	9
3.4	Time Line Chart	14
5.1	Architectural Diagram	20
5.2	Flow Chart	21
5.3	Use Case Diagram	22
5.4	Class Diagram	23
5.5	Sequence Diagram	24
5.6	State transition Diagram for messaging	25
5.7	State transition Diagram for reminder notification	26
5.8	State transition Diagram for registration and applock	27
5.9	State transition Diagram for Web Links	28
5.10	Activity Diagram	29
7.1	Test Case 1	39
7.2	Test Case 2	40
7.3	Test Case 3	41

7.4	Test Case 4	42
7.5	Test Case 5	43
7.6	Test Case 6	43
7.7	Test Case 7	44
7.8	Test Case 8	44
8.1	User filling registration form	46
8.2	OTP verification	46
8.3	App Lock	47
8.4	User typing message for reminder	47
8.5	Message saved in the list	48
8.6	Reminder settings	48
8.7	Before sending friend request	49
8.8	After sending friend request	49
8.9	User Interface of P2P communication	50
8.10	User Interface of Web Links	50
8.11	User Interface of Offline Messaging	51
8.12	User Interface of Secure Chat	51

List of Tables

4.1	User interface	16
-----	--------------------------	----

Chapter 1

Introduction

1.1 Problem Definition

There are ennumnber of Instant Messaging application created which provides features like calling , video chatting, sending images or videos and most of all sending messages for free as along the user has internet. But are these applications user friendly and efficient to use? Thus we have planned to create an instant messaging application similar to Whastapp but with added features. Some of the features added are built-in AppLock, offline messaging, reminder notification. The other notable features added are Integration of our application with other applications ,secure communication. Our main goal is to develop an application which is user friendly and efficient to use.

1.2 Motivation of the thesis

The objective of this application is to create a user freindly application that can really ease user's work. Even if user's net pack or internet connction is lost he/she will be able to send or receive messages. In secure chat messages sent will be in encrypted format and authorised user will be able to decrypt the messages.

1.3 Scope of the thesis

There are various modules present in this project. Firstly the user has to register and will be given OTP code for further verification. Then user will be prompted to enter default app lock password and can edit the password. Then the user will be directed to home page. If user's net connectivity goes he/she has the provision for offline messaging. The application contains a feature of setting reminder notification for the user hence user can set any important notes or any friends birthday or anniversary and application will notify him/her on that day. User has option of visiting sites like amazon, zomato via our application.

1.4 Salient contribution

The sending and receiving of messages will be in real time. In secure chat messages sent will be in encrypted format. Since our application has built in applock user doesn't have to install app-lock. With the application user can get redirected to any website like Zomato, BookMyShow if he/she wishes just by clicking on the image. User can set any important notes. User has the option of sending messages offline if net connectivity is unavailable.

1.5 Organization of the Thesis

1. Chapter 1 mainly talks about overview of the project selected and define the Motivation behind the project selected and Scope of project selected.
2. Chapter 2 provides literature survey done to understand the topic.
3. Chapter 3 provides the Software Project Management Plan (SPMP) for this project which defines the project management goals of the project and includes a description of the deliverables and deadlines.
4. Chapter 4 provides the Software Requirement Specification which contains the functional and non-functional requirements of the project. This document presents an initial description of the various functionalities and services

provided by the software. The document will also serve the basis for acceptance testing by the user.

5. Chapter 5 provides the Software Design Document which includes a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.
6. Chapter 6 provides the System Test Document. The goal of this document is to develop a test plan for the Information Management System. This document defines all the procedures and activities required to prepare for testing of the functionalities of the system. The objectives of the test plan are to define the activities to perform testing, define the test deliverables documents and to identify the various risks and contingencies involved in testing.
7. Chapter 7 provides the conclusion and scope for the future work of the project.

Chapter 2

Literature Survey

The purpose of this chapter is to cover background research and study of existing systems and references in order to improve the project scope.

- **Title: Characterizing Instant Messaging apps on smartphones**

1. Authors : Li Zhang, Chao Xu, Parth H. Pathak, and Prasant Mohapatra
2. Publication Year :2012
3. Published By :UCDAVIS Publication
4. Abstract : Proliferation of smart devices has fueled the popularity of using mobile instant messaging (IM) apps at a rapid pace. While the IM apps on smartphones have become increasingly popular, there has only been a little research on understanding the characteristics of these apps. Because most of the IM apps use proprietary protocols, it is challenging to analyze their internal operations. In this work, we present a comprehensive characterization of mobile IM apps using experiments on LTE cellular network. We decompose the operations of an IM app into multiple independent states which allows us to systematically study them. We characterize the energy and bandwidth efficiency of each of the states and provide numerous insights. Our analysis reveals that typing notification feature of the IM apps is a major contributor

to the energy consumption. We also find that the bandwidth efficiency of current IM apps are alarmingly poor compared to other applications such as email and web surfing. These, along with other findings, provided in this work can help improve the energy and network performance of IM apps. [1]

• **Title of the Paper : Secure Public Instant Messaging**

1. Authors : Mohammad Mannan, P.C. van Oorschot
2. Publication Year :2011
3. Published By :Carl-ton Paper Publication.
4. Abstract : We provide a survey on security features and threats to existing Instant Messaging (IM) networks and discuss how currently available systems fail to provide adequate security in light of existing threats. Our discussion and analysis provide a starting point from which to advance academic research in the area of secure IM systems, enabling security improvement in the longer term. [2]

Chapter 3

Software Project Management Plan

The software project management plan is a benchmark to compare with actual performance. This doc specifies: -The objectives of the project, -The project dependencies and constraints, -The project deliverables and milestones -The project timetable. -Roles and responsibilities of the members of the project, -Process and procedures to be used by the project with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather, analyze and give an in-depth insight of complete project.

3.1 Introduction

3.1.1 Project Overview

Communication is the activity of conveying meaning through a shared system of signs and semiotic rules. Communication helps in expressing ones needs, share ideas and information and share resources with each other. A few years ago, SMS messages were very popular but then came the era of smart phones and traditional SMS texts were replaced by (almost) free online messengers. Instant messaging (IM) has become a trend these days especially with the youth. A trend, pioneered by the immensely popular IM application like WhatsApp, Line, Viber. But even

after inventions of such IM Applications people still look for application that can ease their work. In this project we are developing an IM application which will overcome some of the drawbacks of other IM applications and also provide few added features. This android application is made solely for users who want a better messaging application. The expected delivery date of the project will be 31-03-2016.

3.1.2 Project Deliverables

SR. No.	Steps	Project Deliverables	Expected date
1	Information gathering and analysis	Problem definition, Scope, functional and non-functional requirements, SPMP, SRS	15-6-15
2	Concept Creation	Project Synopsis	1-8-15
3	Coding	Script and Snapshot	20-3-16
4	Testing	System test document(Executable codes, test cases and their reports)	25-3-16
5	Deployment	Working project and project report	31-3-16

Figure 3.1: Project Deliverables

3.2 Project Organization

3.2.1 Software Process Model

The process model we are using is Prototype model. The prototyping model is

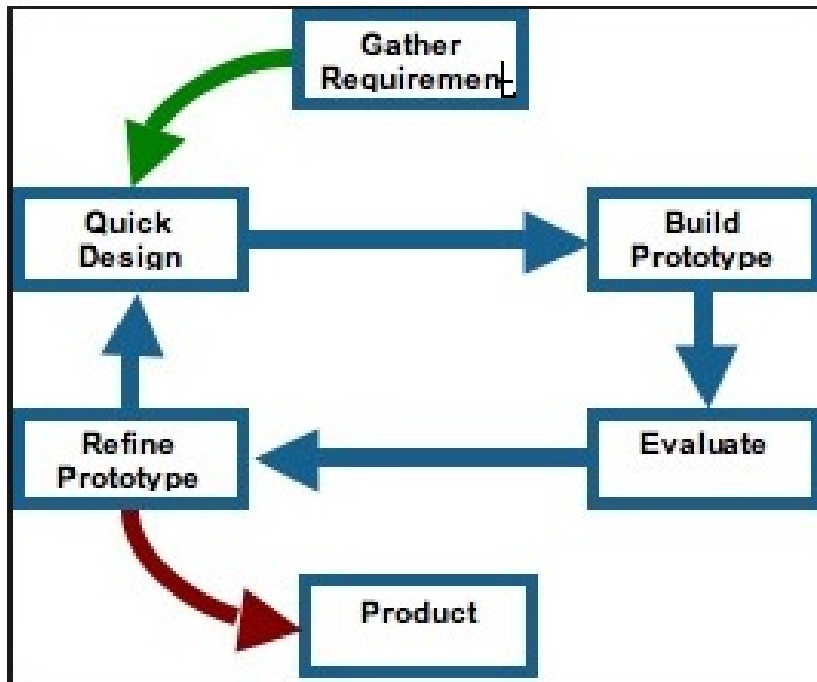


Figure 3.2: Prototype Model

applied when detailed information related to input and output requirements of the system is not available. In this model, it is assumed that all the requirements may not be known at the start of the development of the system. It is usually

used when a system does not exist or in case of a large and complex system where there is no manual process to determine the requirements. This model allows the users to interact and experiment with a working model of the system known as prototype. The prototype gives the user an actual feel of the system. [3]. The main reasons behind selecting this model are:

- This model is more flexible less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- Major requirements must be defined; however, some details can evolve with time.

3.2.2 Roles and Responsibilities

The project has 4 members Ajay Mohandas, RajanKumar Bhimprasad, Vijay Yadav, Pradeep Bagrao.

Sr. No.	Task	Ajay Mohandas	RajanKumar Bhimprasad	Vijay Yadav	Pradeep Bagrao
1	Requirement analysis	✓	✓	✓	✓
2	Planning	✓	✓	✓	✓
3	System design	✓	✓	✓	✓
4	Documentation	✓	✓	✓	✓
5	Coding	✓	✓	✓	✓
6	Testing	✓	✓	✓	✓

Figure 3.3: Roles and Responsibility

Each team member will be involved in all the activities. The different activities of our project are:

- Documentation and analysis of system.
- Identification and gathering of information needs.
- Defining requirements of resources needed for further implementation.

- Designing UI.
- Designing the technical architecture.
- Coding.
- Testing and prepare reports of defects

3.2.3 Tools and Techniques

3.2.3.1 Hardware Requirements

Android mobile devices to access internet.

A computer to build the software and test it.

1. Hard disk :400 MB
2. RAM :2 GB minimum

3.2.3.2 Software Requirements

1. JDK 7 or higher
2. Android Studio XAMPP for web server

3.2.3.3 Languages to be used

1. Java
2. XML

3.3 Project Management Plan

3.3.1 Tasks

The various tasks involved in executing the project are:

1. Concept Creation.
2. Requirement Gathering and Analysis.
3. Design
4. Coding
5. Testing

3.3.1.1 Concept Creation and Feasibility

Description

This is the first task of the project. It involves deciding upon the basic features of the project and checking the feasibility of the project.

Deliverable Milestone

Project Synopsis.

Dependencies and Constraints

Task requires the project idea to be clear and the feasibility study to be carried out.

Resource Needed

Risk during this stage is that the project might not be feasible. It is a very high impact risk which will result in the project being shelved. Hence feasibility should be carried out carefully.

3.3.1.2 Requirement Gathering and Analysis

Description

This stage of the project includes extensive research and study on other IM applications and what are their drawbacks and hence what new features can be added.

- Identify scope of the project.

- Identify requirements.
- Identify constraints.
- Identify tools and techniques required.

Deliverable Milestone

Software Requirement Specification.

Resource Needed

We will need different website references on IM applications and some IEEE papers that can give data on Secure messaging Android tutorials Offline messaging Chat Application

3.3.1.3 Design

Description

This phase includes the architectural design and GUI design of the project. The GUI should be user compatible.

Deliverable Milestone

Software Design Document.

Dependencies and Constraints

Designing phase cannot start until all the requirements are completed. The GUI is intended to be user friendly and easy to understand and meet all the requirements.

Resource Needed

The websites giving informations on GUI of other IM applications

3.3.1.4 Coding

Description

It deals with converting whatever we have done during design phase into actual mode. Here we actually write the code for each of the modules and integrate them.

Deliverable Milestone

- Source code
- Executable code.

Resource Needed

- Android studio
- XAMPP for web server
- IEEE papers
- Android tutorials

3.3.1.5 Testing

Description

It deals with testing various modules and applications of the project using black box and white box testing.

Deliverable Milestone

Executable code without errors

Dependencies and Constraints

This task requires coding phase to be completed.

3.3.2 Time Chart

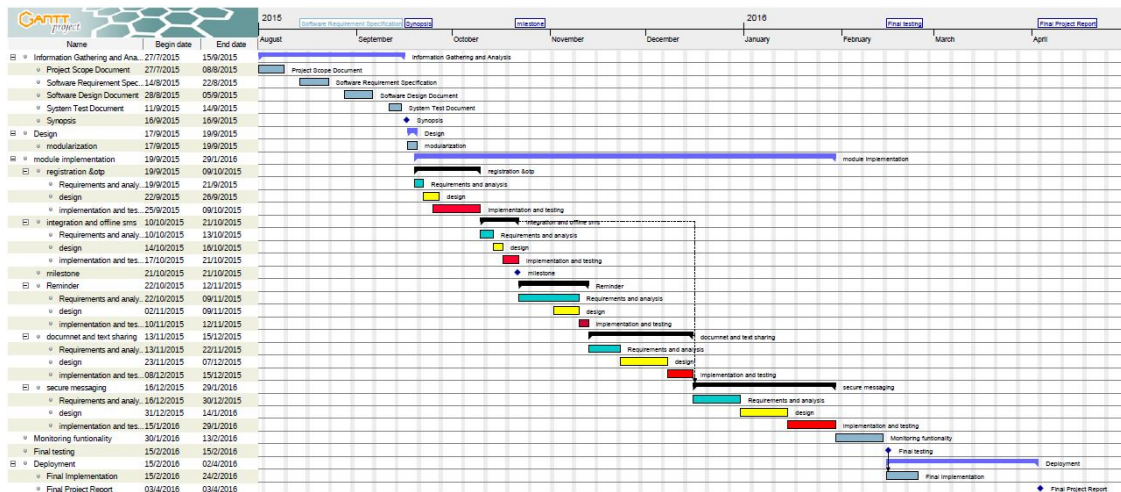


Figure 3.4: Time Line Chart

Chapter 4

Software Requirements Specification

<p>The SRS is a specification of the requirements for the software product that will be developed in project. It contains requirement specifications that answer what exactly the project is about and what are the needs to complete the project such as functional, non-functional requirements.</p>
--

4.1 Introduction

4.1.1 Product Overview

Inspite of inventions of such IM Applications like WhatsApp, Line, Viber people still look for application that can ease their work. In this project we are developing an IM application which will overcome some of the drawbacks of other IM applications and also provide few added features. The user must have an android phone and technical experience is not mandatory for using the application. The purpose of this document is to show the software requirements of the YELLO application. The functionalities and non-functionalities of this software are also described in this document.

4.1.2 Intended Audience and Document Overview

This SRS document is intended for developers and testers for reading. The document contains the functional and non-functional requirements of Yello. The developers and testers will know what all software, hardware and user interfaces are used in making the application. The SRS fully describes what the software will do and how it will be expected to perform.

4.2 Specific Requirements

4.2.1 User Interfaces

The project consists of 9 modules. Thus there will be user interface for each module to be executed.

Screen Name	Description
Registration and OTP page	UI for registration of the user and otp verification
P2P Communication	UI for peer to peer communication
App lock	UI for unlocking to application
Secure Messaging	UI for sending secure messages
Integration page	UI for various links for integration of other applications
Reminder Notification	UI for setting to-do list for the user

Table 4.1: User interface

4.2.2 Software Interfaces

To implement the project we are using the 1.0 of Android Studio. The programming languages used are XML, Java.

4.2.3 Communications Protocols

In this project we are using a combination of Client-Server and P2P protocol. A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the

network. This model of network arrangement differs from the clientserver model where communication is usually to and from a central server. P2P model is used for communication between two users. Client-Server model is used for storing the account details of a user and also authenticating the user.

4.3 Functional Requirement

1. Application will keep User records: Login/Register, maintain account
2. User will be prompted to create a lock key after registering and will always be prompted to match that pattern to use the application
3. Users can send messages in offline mode if net connectivity is unavailable.
4. User will be notified if he/she has set any reminder or get notification regarding birthday details of the person form the contact list
5. User will be redirected to the link when he/she clicks on the image
6. Users can also send encrypted messages via secure chat.

4.4 Non-Functional Requirement

Non-functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

4.4.1 Performance Requirements

Performance requirements define acceptable response times for system functionality

1. Load time for user interface screens shall take no longer than a few seconds
2. OTP verification will be done within few seconds
3. Redirection to next screen will be immediately

4.4.2 Standards Compliance

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

4.4.3 Reliability

The application will never crash, and will always be in the state to give optimal solutions.

4.4.4 Availability

The user can send messages at any particular time. The application will never abruptly terminate within the process of sending or receiving messages. Even after losing net connectivity user will be able to send or receive messages via offline mode.

4.4.5 Maintainability

The code for implementation of the project is clear and easily manageable. For future upgradation it will be easy for the developer to update the application.

4.4.6 Portability

The application will run on any mobile device that has Android as its operating system.

Chapter 5

Software Design Description

An SDD is a representation or model of the software system to be created. This provides the precise design information needed for planning, analysis, and implementation of the software system. It describes the software structure, software components, interfaces, and data necessary for the implementation phase.

5.1 Introduction

It includes the objective and scope of this document. It also contain the system architecture and the UML diagrams of the system.

5.1.1 Design Overview

The following designs of the system will give an overview of how the system will look and will give information regarding the modules and the data flow within the system.

Architectural Design

The following diagram gives a pictorial representation of our application's architecture

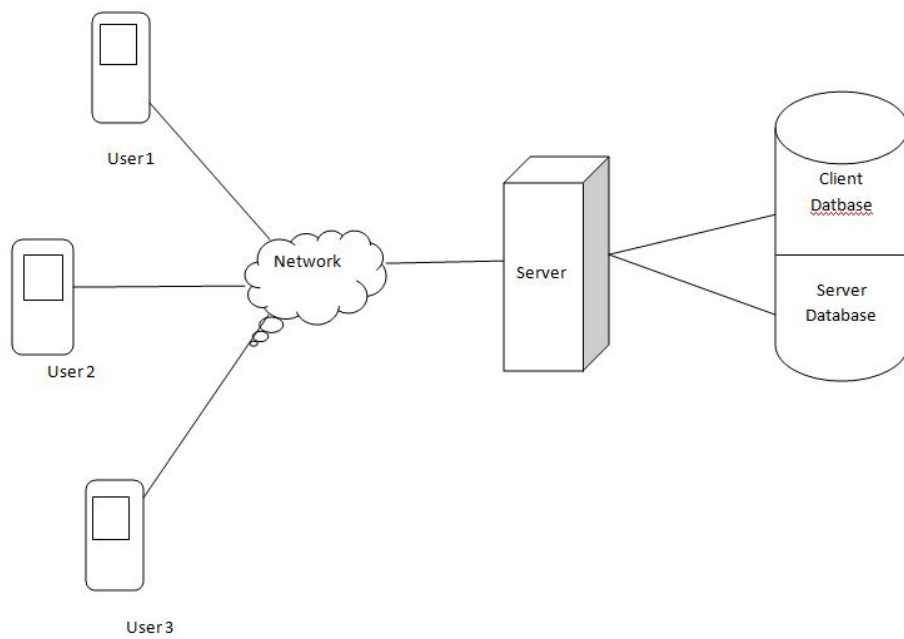


Figure 5.1: Architectural Diagram

Flow Chart

The following diagram gives a pictorial representation of our application's flowchart

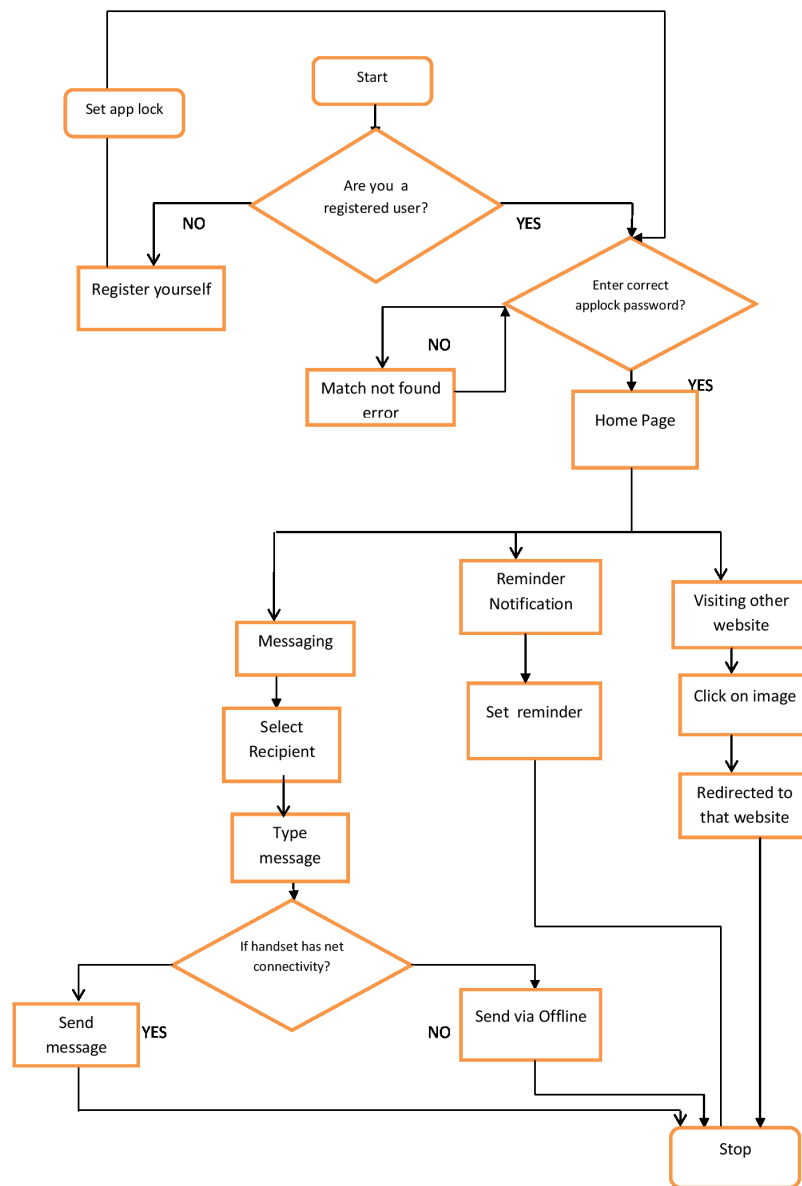


Figure 5.2: Flow Chart

Use Case Diagram

The following diagram is the use case diagram of Yello App.

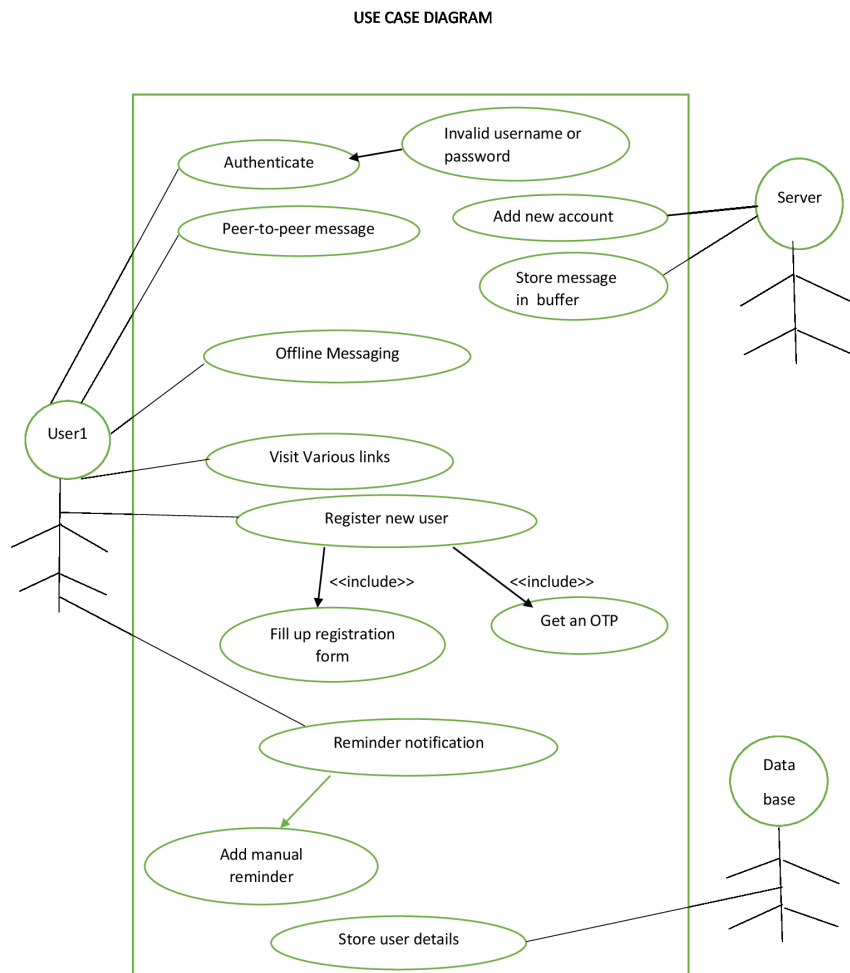


Figure 5.3: Use Case Diagram

The user , server and the database are the actors in the diagram.

Class Diagram

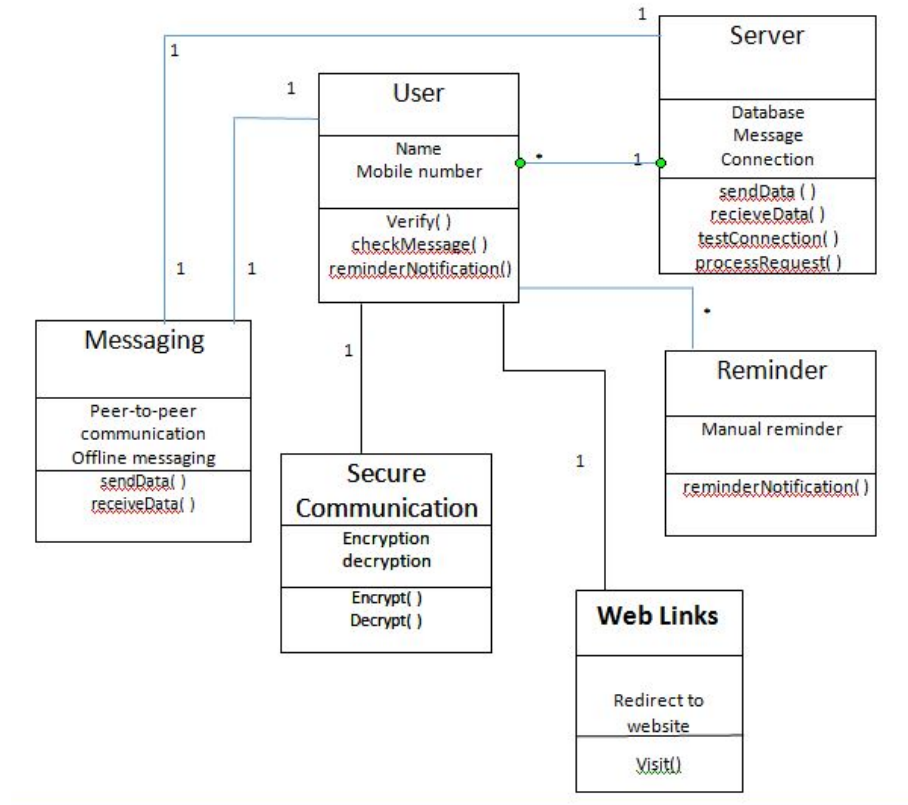


Figure 5.4: Class Diagram

This diagram is describing the structure of our system by showing its classes, their attributes and their functions.

Sequence Diagram

Sequence diagram shows how processes of our system operate with one another and in what order. It shows object interactions arranged in time sequence

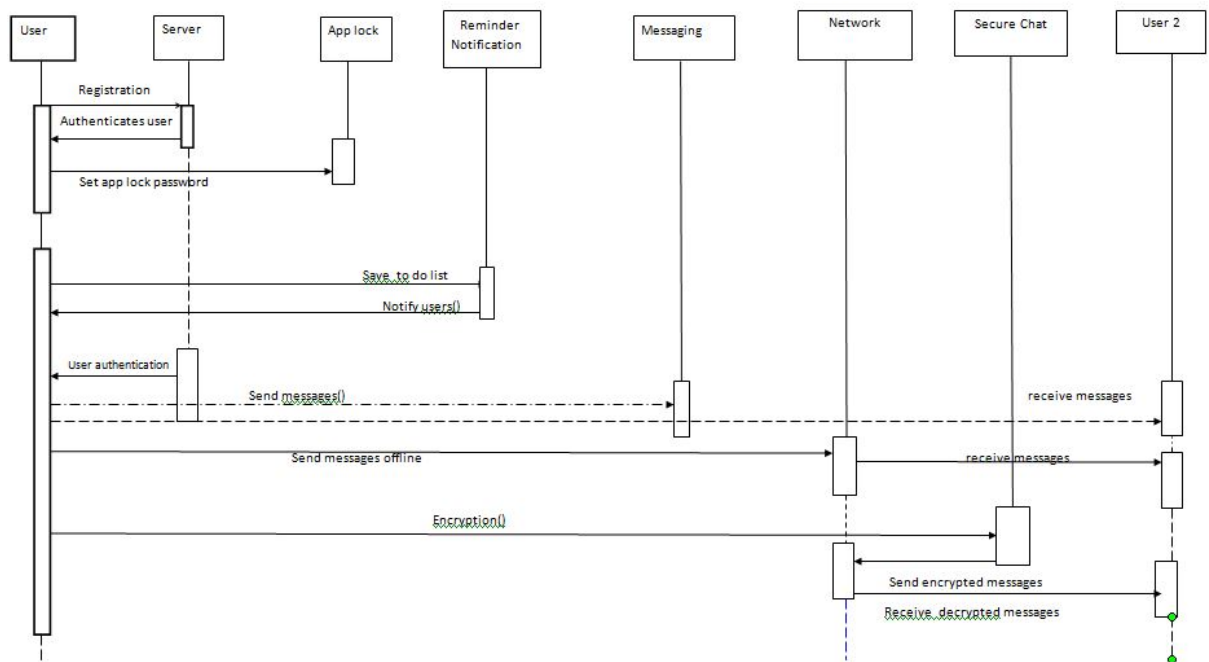


Figure 5.5: Sequence Diagram

State transition diagram

It describes different states of components in a system. The states are specific to a component/object of a system.

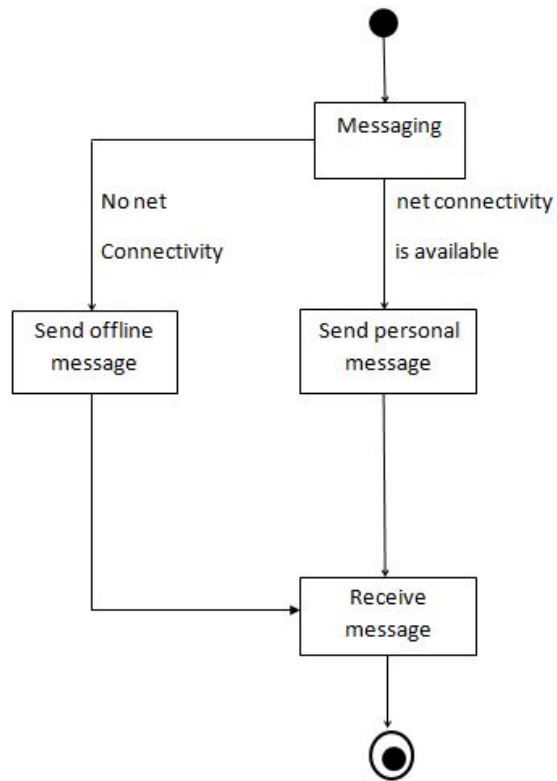


Figure 5.6: State transition Diagram for messaging

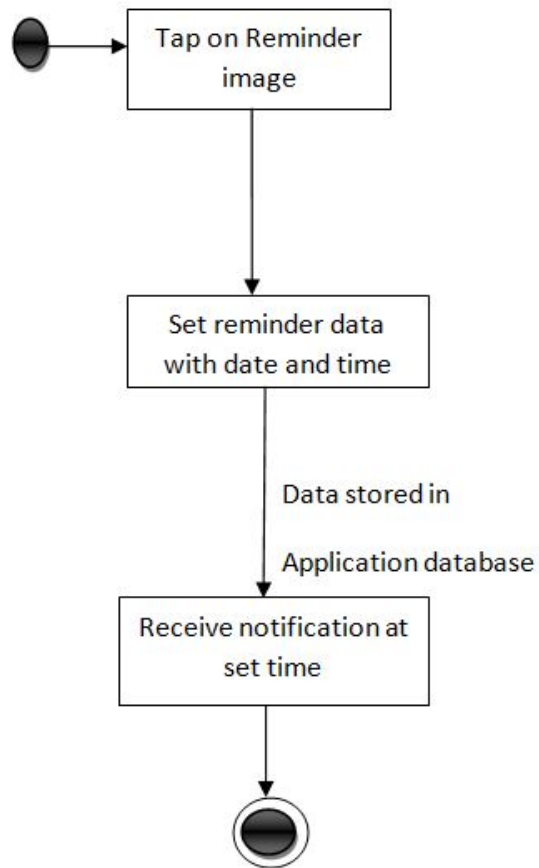


Figure 5.7: State transition Diagram for reminder notification

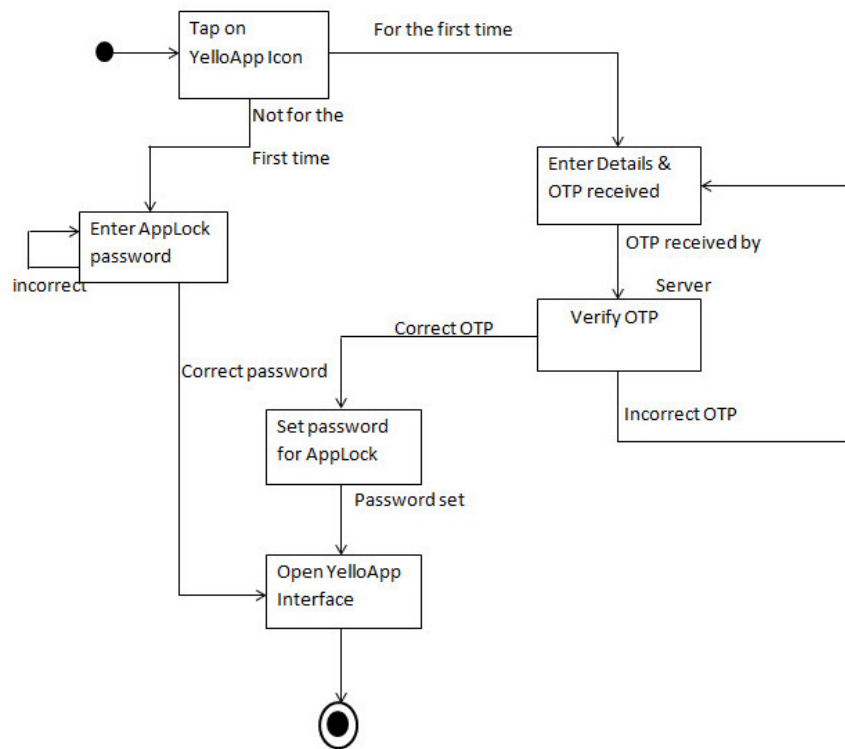


Figure 5.8: State transition Diagram for registration and applock

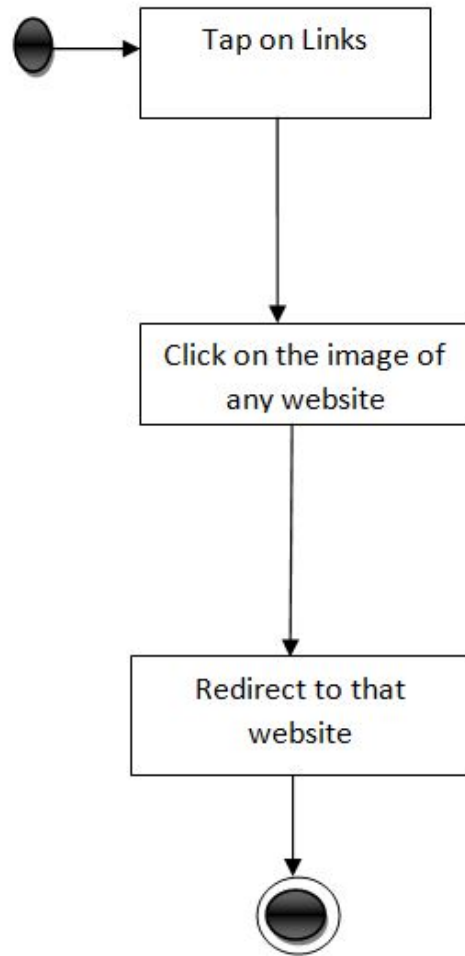


Figure 5.9: State transition Diagram for Web Links

Activity Diagram

Activity diagrams are graphical representations of workflow of stepwise activities and actions. An activity diagram is showing the overall flow of control of our system

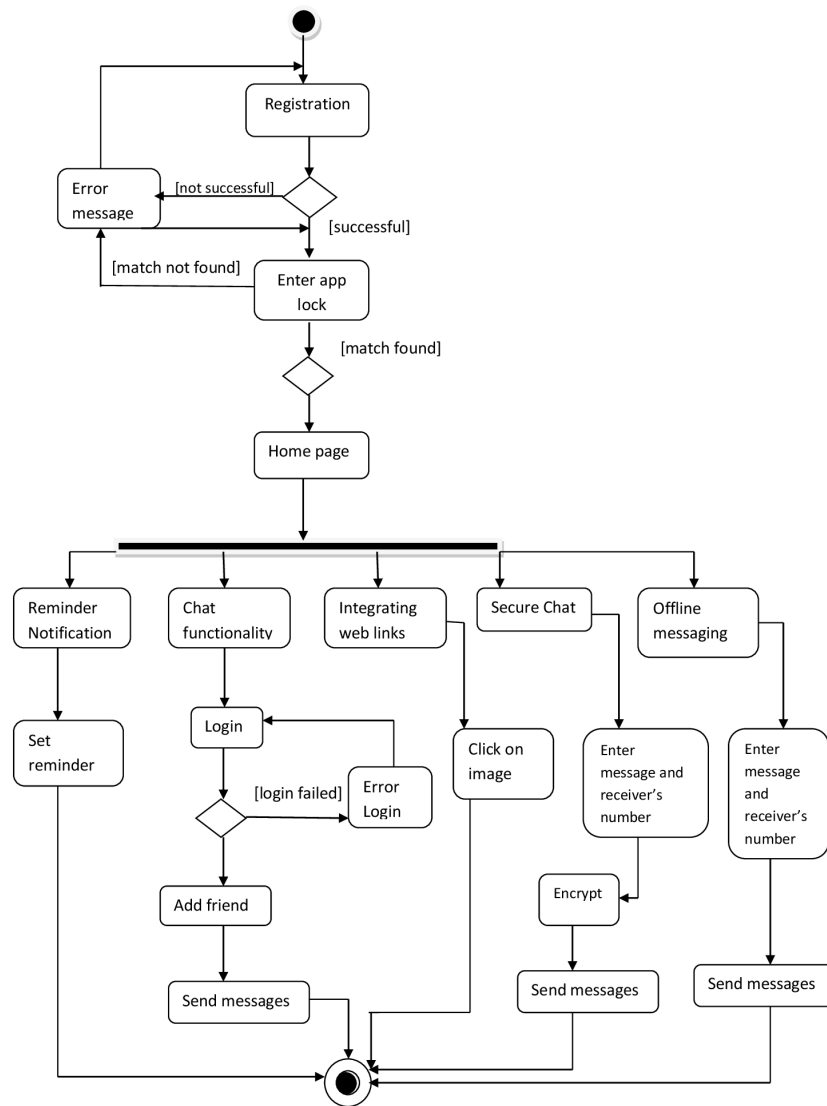


Figure 5.10: Activity Diagram

Chapter 6

Implementation

This document gives overview of implementation including technologies used to develop and run the project, also source code and pseudo code of developed system is provided.

6.1 Technologies to be used

6.1.1 Hardware Requirements

- Android mobile devices to access internet.
- A computer to build the software and test it.
 1. Hard disk:400 MB
 2. RAM :2 GB minimum

6.1.2 Software Requirements

1. JDK 7 or higher packages
2. Android Studio
3. XAMPP software

6.1.3 Languages to be used

1. Java
2. XML
3. PHP

6.2 Algorithms used

6.2.1 Registration and OTP verification

Registration is the first step before users can start using our application. In the registration form user fills details like entering their username, email-id and password. The users will get an OTP verification code, by entering the correct number they will be directed to the next page. The users details like the username and password are stored in our database along with data like status, user key, authentication time, IP address. Password is encrypted using md5 method.

1. Status key states if the user is online or offline.
2. Every user is given a key called user-key which is verified during authentication purpose.
3. IP denotes the ip address of network through which the users device is connected.
4. Authentication time is the time when the user is online

6.2.2 Add Friend List

To add friends initially a user sends a friend request to another user. For instance A sends a friend request to B. At that particular time user As user id is sent to B. At the database there is friend list table. Bs data is checked in that table. If A is already a friend of B then further operations of adding friends stops. But if A is not a friend of B then a notification message is sent to user Bs device. If B accepts the request then A is added to Bs friend list.

6.2.3 App Lock

App-lock is implemented for security purpose. Even if the mobile device goes into wrong hands he/she wont be able to view messages unless someone tries to type the correct app-lock password. In order to implement the module we have used a built-in function i.e. Shared Preference. Shared Preference allows storing data in the form of key, value pair. The value is saved by using Shared Preference Editor class. Thus whenever a user enters the number in box, if it matches with the value stored then user will be directed to the main page. User also has the provision to edit password whenever he/she wishes.

6.2.4 Chat Functionality

Before using chat functionality user has to login to chat page. In login page user has to enter username and password. At this particular time server verifies the details of the user. Users user-key which is stored in the database is verified and checked whether the user is an authorised user or not. If user is authenticated then users status key changes to online else error message will displayed to enter correct username or password. User first selects the friend he/she wishes to send message. After the message is typed user has to click the send button. If there is no problem with the network the message is delivered successfully. The receiver can view the message along with the name of the sender. In messaging module a Handler function is used which allows sending and processing message and Runnable objects associated with a thread's Message Queue. Each Handler instance is associated with a single thread and that thread's message queue. Handler helps in scheduling by using send message function which helps in sending the message to the destination. Since the messages sent is online a permission has to be set in the android manifest file:-

```
uses-permission android:name="android.permission.INTERNET"
```

6.2.5 Offline Messaging

When there is not network connectivity and user is unable to send messages, he/she has the provision to use Offline messaging via our application. In this module user

has to enter the mobile number of the receiver and the message in the text box. After the Send button is pressed the message is delivered to the receiver. On senders SMS Sent message is displayed after message is sent and SMS delivered message is delivered when message is successfully delivered to receiver. To implement this module we have to set the send sms permission in the android manifest file. Then an in-built function `sendTextMessage` is used. `sendTextMessage` contains the receivers phone number and the message to be sent as parameters. This function helps in delivering of the message to the receiver. The permission is set in the following manner:- `uses-permission android:name=android.permission.SENDSMS`

6.2.6 Integrating Web Links

If user wishes to open any applications like Book-my-show, amazon, zomato then user can do it via our application. When user clicks on any image for instance on amazon then user will be redirected to amazon website. This is done by using `setDataUri.parseurl`. In the url we can specify any websites url.

6.2.7 Reminder Notification

In reminder notification user can set reminder on any particular date, time. For instance user has set birthday reminder of any close friends or relatives and has selected the date and time, user will be notified at that set time. To implement this we have used System settings which contain various system preferences. For notification we have used the following in-built functions:-

1. `System.currentTimeMillis` This function returns the current time in milliseconds.
2. `Shared Preferences` This helps in storing data in the form of key-value pair.
3. `System.DEFAULTNOTIFICATIONURI` This helps in giving default notification sound.
4. `Dialog Interface` This function opens up a dialog box where user enters the text and after typing user click on button to store the details.

Users have the provision to change settings of reminder notification module like changing 12 hour clock to 24 hour clock or vice versa. After the message is typed users can select repeat or once as per users requirements. Repeat means user will be notified the same day every time and once means user will be notified once at that set date and time.

6.2.8 Secure Messaging

In secure chat module we have used an encryption algorithm for encrypting the messages. If user has any important messages to be sent can use the secure chat. In this when user types the message and clicks send button then message will be encrypted and sent to receiver. If user wishes to view the message then press decrypt button. The user will be prompted to enter password; if it matches then user can view the decrypted message. An in-built function `sendTextMessage` is used. `sendTextMessage` contains the receivers phone number and the message to be sent as parameters. To implement this module we have to set the send sms permission in the android manifest file. The permission is set in the following manner:- `uses-permission android:name=android.permission.SENDSMS`. The algorithm used for encryption is substitution cipher. In this if user types letter A the message sent is D i.e for every alphanumeric typed it is incremented by 3 and sent. At time of decryption the alphanumeric is decremented by the same logic to get the actual data.

6.3 Pseudocode

1. Start
2. User opens the application by clicking on the icon.
3. Registration form appears and user fills the form by filling username, password and email-id.
4. When user hits submit button, user will be directed to next page if all the fields in registration is filled correctly else error message will be displayed.

5. In next step user will be given an OTP code.
 - If user types the correct code then next page appears
 - Else error message appears
6. User then enters app lock password
7. User has the provision to edit app lock password by typing new password and submitting it
8. As user passes app lock then home page appears
9. If user clicks Reminder then
 - User types message and sets time and date
 - User click the submit button
10. If user clicks Links then
 - If user clicks on image of any website then he/she will be directed to that site
11. If user clicks Chat then user will be prompted to login. After login then user can send friend request to another user. A message will be sent to that user. After receiver accepts friend request then sender can send messages to receiver
12. If handset does not have net connectivity then user can avail offline messaging. In this user types message and receiver's mobile number and hits submit. Message will be delivered and a message will be displayed saying the message is successfully delivered else error message will be displayed.
13. If user clicks Secure chat then user types message and receiver's mobile number and hits submit. Message will be delivered in encrypted format to receiver and a message will be displayed saying the message is successfully delivered else error message will be displayed. To decrypt the message user has to click on decrypt and user has to enter app lock password to decrypt the message.

Chapter 7

Software Test Document

This chapter includes all the test scenarios such as testing approaches and type of testing used. It also shows you the functionalities which are to be tested and or not.

7.1 Introduction

7.1.1 System Overview

The aim of this phase is to test the whole project providing coverage of the system, before deploying the application. To do it has been divided according to all the features in the following different modules:

- Registration
- Reminder Notification
- P2P communication
- Group Communication
- Sending Documents
- App Lock Protection

7.1.2 Test Approach

7.2 Test Plan

The test plan is drawn up during the design stage and serves as a guide in carrying out test. The test plan includes:

1. A description of the condition under which the test will run
2. A description of the test data to be used.
3. A description of the expected result.

Description of the test data to be used. The data needed as input is:

- Source code
- User response

Description of the expected result: All the modules must run without giving any error else the application will stop abruptly and crash.

Features to be tested

1. Registration
2. Reminder Notification
3. P2P communication
4. Group Communication
5. Sending Documents
6. App Lock Protection

7.2.1 Features not to be Tested

Some areas in the project low priority modules and low risk areas that features are not tested and specified in the test plan. If the feature or the module has been used before and was considered stable then we can skip testing. The basic aim of this is reduced effort and time.

7.2.2 Testing Tools and Environment

7.2.2.1 Test Cases

A test case signification refines the test approach and identifies the feature to be covered by the particular case. It also identifies the procedures required to accomplish the testing and specifies the pass/fail criteria. It also documents the actual values used for the input along with the anticipated outputs.

7.2.2.2 Black box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. [4]

7.3 Test Cases

Project Name: Yello						
Test Case ID: Y.1.0			Test Designed by: Vijay Yadav			
Test Priority (Low/Medium/High):Medium			Test Designed date: 25-09-15			
Module Name: Registration			Test Executed by: Vijay Yadav			
Test Title: verifying the registration module			Test Execution date:26-09-2015			
Description: Checking if the form takes all valid values						
Pre-conditions: Size of username must be more than 6						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check, if the name is entered by user	Empty	“username not entered” message must be displayed	“username not entered” message is displayed	Pass	
2	Check if name entered is valid	Invalid input	“Size of username must be more than 6” message must be displayed	“Size of username must be more than 6 ” Message is displayed	Pass	
3	Check if name entered is valid	Valid input	Username is verified	Username is verified	Pass	
Post-condition: After name is verified check other details in the form						

Figure 7.1: Test Case 1

Project Name: Yello						
Test Case ID: Y.1.1				Test Designed by: Vijay Yadav		
Test Priority (Low/Medium/High):Medium				Test Designed date: 25-09-15		
Module Name: Registration				Test Executed by: Vijay Yadav		
Test Title: verifying the registration module				Test Execution date:26-09-2015		
Description: Checking if the form takes all valid values						
Pre-conditions: Email and password has to be filled						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check, if the email is entered by user	Empty	“Email not entered” message must be displayed	“Email not entered” message is displayed	Pass	
2	Check if password is entered	Empty	“Password not entered” message must be displayed	“Password not entered” message is be displayed	Pass	
3	Confirm password and click submit	Invalid input	“Password does not match” message must be displayed	“Password does not match” message is be displayed	Pass	
4	Enter Submit after all values are filled correctly	Valid input	“Registration is complete” message must be displayed	“Registration is complete” message is be displayed	Pass	
Post-condition: After registration , OTP verification is done						

Figure 7.2: Test Case 2

Project Name: Yello						
Test Case ID: Y.1.2			Test Designed by: Vijay Yadav			
Test Priority (Low/Medium/High): Medium			Test Designed date: 25-09-15			
Module Name: Registration and OTP verification			Test Executed by: Vijay Yadav			
Test Title: verifying the registration module			Test Execution date: 26-09-2015			
Description: OTP verification						
Pre-conditions: Registration must be complete						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
2	Check if number entered is valid or invalid	Invalid input	“Match not found” message must be displayed	“Match not found” is displayed	Pass	
3	Check if number entered is valid or invalid	Valid input	“Match found” must be displayed	“Match found” is displayed	Pass	
Post-condition: User is directed to app lock page						

Figure 7.3: Test Case 3

Project Name: Yello						
Test Case ID: Y.1.3				Test Designed by: Ajay Mohandas		
Test Priority (Low/Medium/High):High				Test Designed date: 25-09-15		
Module Name: Chat Functionality				Test Executed by: Pradeep Bagrao , Ajay Mohandas		
Test Title: sending and receiving messages				Test Execution date: 10-2-2016		
Description: Checking if the documents are correctly sent and recieved						
Pre-conditions: User must complete verification process and see if any friend is online						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Message is sent by user (If friend is online)	Messages	Message is sent successfully	Message reaches receiver successfully	Pass	
2	Message is sent by user (If friend is offline)	Messages	Messages will not be sent	Message is not received	Pass	
3	Receiving of messages	Messages	Message is received	Message is received by user successfully	Pass	
Post-condition: User can also use other features provided by the application						

Figure 7.4: Test Case 4

Project Name: Yello						
Test Case ID: Y.1.4				Test Designed by: Ajay Mohandas		
Test Priority (Low/Medium/High):Medium				Test Designed date: 25-09-15		
Module Name: Integration with other application				Test Executed by: Ajay Mohandas		
Test Title: verifying the integration part				Test Execution date: 28-9-2015		
Description: Checking whether the links are redirecting to desired website						
Pre-conditions: User must click on the link						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check, if user is redirected to another site	Url of any application	The page will redirect to the desired site	User is redirected successfully	Pass	
Post-condition: User will be able to view the desired site.						

Figure 7.5: Test Case 5

Project Name: Yello						
Test Case ID: Y.1.5			Test Designed by: Ajay Mohandas			
Test Priority (Low/Medium/High):Medium			Test Designed date: 25-09-15			
Module Name: Reminder Notification			Test Executed by: Rajankumar Bhimprasad			
Test Title: verifying the notification			Test Execution date: 28-09-15			
Description: Checking whether user is getting notification						
Pre-conditions: User must set a to-do list and enter the time and date						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check, if the user is notified	Insert the details	User will be notified on that set time and date	User is notified at the set time and date	Pass	
Post-condition: User will get notification as per his to-do list.						

Figure 7.6: Test Case 6

Project Name: Yello						
Test Case ID: Y.1.6			Test Designed by: Ajay Mohandas			
Test Priority (Low/Medium/High): High			Test Designed date: 1-1-2016			
Module Name: App lock			Test Executed by: Pradeep Bagrao			
Test Title: verifying the security			Test Execution date: 5-1-2016			
Description: Checking whether the user is authenticated before using the application						
Pre-conditions: User must set the app lock						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check, if the app lock is working	Insert the wrong input	“Match not found” message must be displayed	“Match not found” is displayed	Pass	
2	Check, if the app lock is working	Insert correct password	“Match found” must be displayed	“Match found” is displayed	Pass	
Post-condition: After unlocking the app, user is directed to menu page.						

Figure 7.7: Test Case 7

Project Name: Yello						
Test Case ID: Y.1.7			Test Designed by: Ajay Mohandas			
Test Priority (Low/Medium/High):Medium			Test Designed date: 10-3-2016			
Module Name: Integration of all module			Test Executed by: Vijay Yadav, Rajankumar Bhimprasad			
Test Title: verifying the Yello application			Test Execution date: 15-3-2016			
Description: Checking whether the all application is running properly						
Pre-conditions: Check all modules are working properly						
Dependencies:						
Step	Description	Input	Expected Output	Actual Output	Result(Pass/Fail)	
1	Check if each and every modules are running	Run the application	Every module is giving desired output	Each and every module is integrated and running successfully	Pass	

Figure 7.8: Test Case 8

Chapter 8

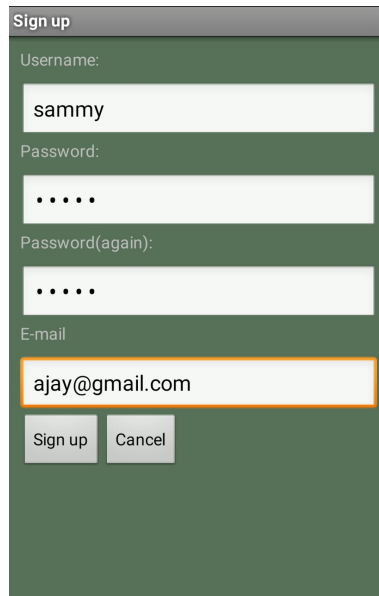
Results and discussion

This chapter covers the result obtained during testing of the developed system. It is depicted with help of system screenshots of the implementation.

8.1 Different modules in application

1. Registration and OTP verification
2. App Lock
3. Reminder Notification
4. Integrating Web Links
5. Peer-peer communication
6. Offline Messaging
7. Secure Chat

8.2 Output of each modules



A screenshot of a 'Sign up' form with a dark green background. The form has a title bar 'Sign up' in a grey box. Below it are four input fields: 'Username:' with the value 'sammy', 'Password:' with five dots, 'Password(again):' with five dots, and 'E-mail' with the value 'ajay@gmail.com'. The 'E-mail' field is highlighted with an orange border. At the bottom are two buttons: 'Sign up' and 'Cancel'.

Sign up

Username:

sammy

Password:

.....

Password(again):

.....

E-mail

ajay@gmail.com

Sign up Cancel

Figure 8.1: User filling registration form



A screenshot of an 'otp' (One-Time Password) verification form with a dark green background. The form has a title bar 'otp' in a grey box. It contains three elements: a 'Get OTP' button, an input field with the value '4690' (highlighted with an orange border), and a 'Submit' button.

otp

Get OTP

4690

Submit

Figure 8.2: OTP verification

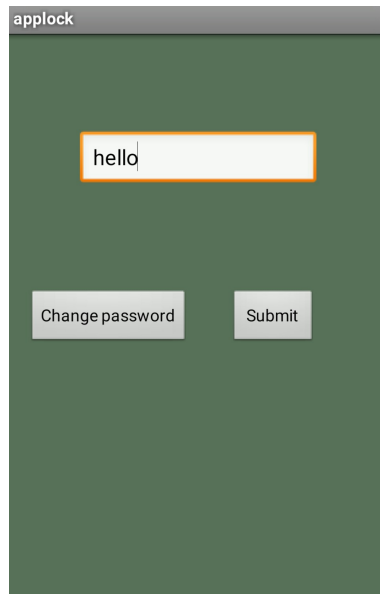


Figure 8.3: App Lock

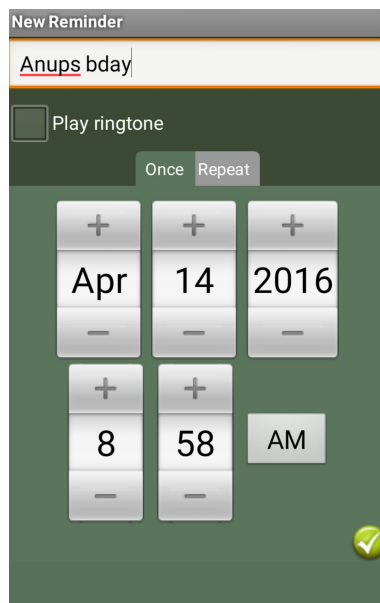


Figure 8.4: User typing message for reminder

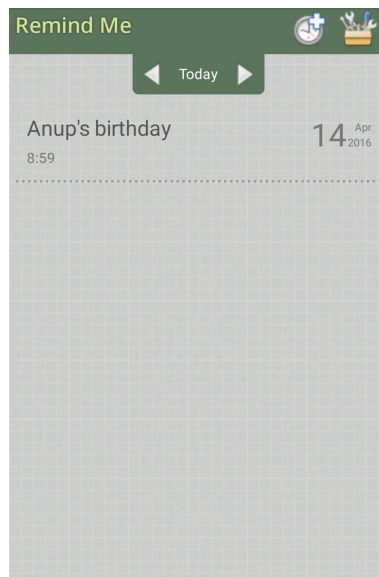


Figure 8.5: Message saved in the list

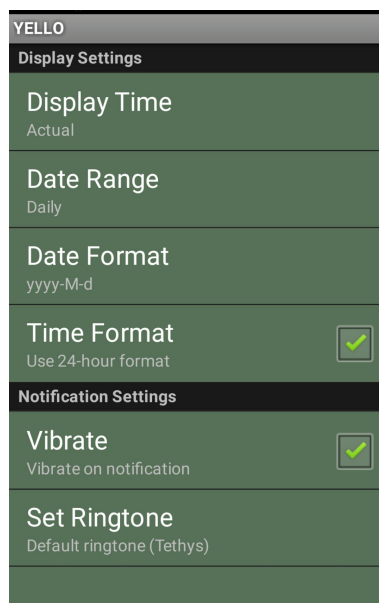


Figure 8.6: Reminder settings

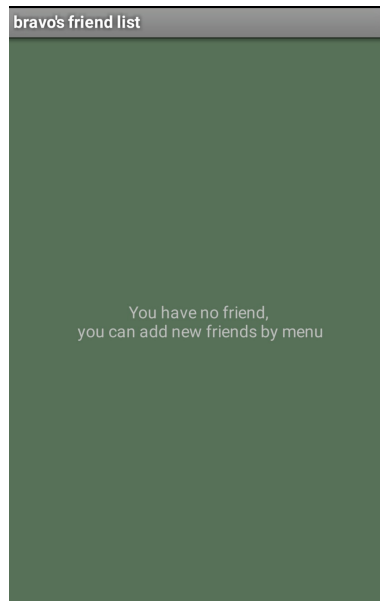


Figure 8.7: Before sending friend request



Figure 8.8: After sending friend request

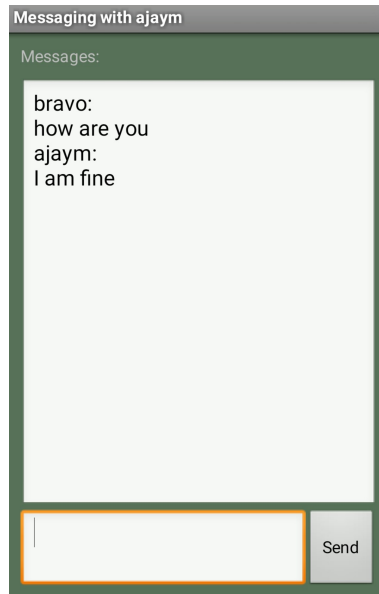
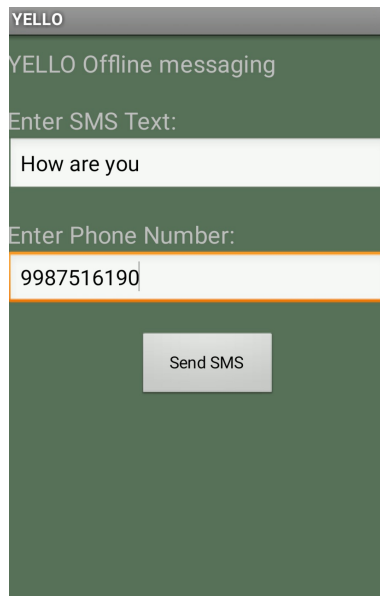


Figure 8.9: User Interface of P2P communication



Figure 8.10: User Interface of Web Links



YELLO

YELLO Offline messaging

Enter SMS Text:

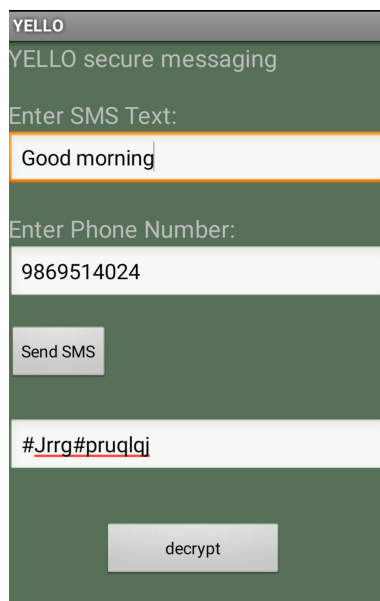
How are you

Enter Phone Number:

9987516190

Send SMS

Figure 8.11: User Interface of Offline Messaging



YELLO

YELLO secure messaging

Enter SMS Text:

Good morning

Enter Phone Number:

9869514024

Send SMS

#Jrrg#pruqlqj

decrypt

Figure 8.12: User Interface of Secure Chat

Chapter 9

Conclusion and Scope for future work

This chapter provides the conclusion driven from all the work done on the thesis. It also includes the future scope of the thesis.

9.1 Conclusions

The instant messaging applications like WhatsApp, Line , Viber has increased the level of social interaction among people. People can connect with their relatives, friends who are staying thousand miles away in real time. But as the years are passing and the level of technology is increasing the demands of people are also increasing. Our application **"YELLO"** will take the level of communication to next level. It has all the features of other instant messaging application and also some unique features are also added into it. Our application will allow users to set a to-do list and will be notified for the same. Users can also visit other website like Zomato, BookMyShow, Flipkart via our application. If net connectivity is unavailable then users can send messages in offline mode. In secure communication messages sent are in encrypted format and only authorised user's can decrypt the messages. The application created is very user friendly hence user is not required to be of any expert in technology to run our application.

9.2 Scope for future work

Currently our offline messaging and chat module is separate. In future we would integrate both and if users wants to send any message when there is no network then message will be forwarded and at the side there will be an indication that message is sent offline. Other than messaging we would also go for video calling. We will keep a limit for user to type the correct app lock number for example (not more than 5) and even then if the security code is wrong then a picture of the person will be taken without the person knowing and a message and the photo will be given to the user via mail. Stronger encryption algorithm will be used in secure chat. User's will be able to send all types of documents(word file, pdf, images, vidoes) via our application

Bibliography

- [1] Doi K. Current status and future potential of computer-aided diagnosis in medical imaging. *The British Journal of Radiology*, 2005.
- [2] <http://www.kanziuscancerresearch.org/research/types-of-cancer-being-investigated/liver-cancer?gclid=CMym8anxbcCFY9V4god7l8AFw>. accessed on june 2013.
- [3] Yufei Chen, Weidong Zhao, Qidi Wu, Zhicheng Wang, and Jinyong Hu. Liver segmentation in ct images for intervention using a graph-cut based model. In Hiroyuki Yoshida, Georgios Sakas, and MariusGeorge Linguraru, editors, *Abdominal Imaging. Computational and Clinical Applications*, volume 7029 of *Lecture Notes in Computer Science*, pages 157–164. Springer Berlin Heidelberg, 2012.
- [4] Feng xiao-Mingyue Ding Wu Qiu, Rui Wang. Research on fuzzy enhancement in the diagnosis of liver tumor from b-mode ultrasound images. *International Conference on Intelligent Computation and Bio-Medical Instrumentation*, 2011.
- [5] Vipin Chaudhary Raja S Alomari, Suryaprakash Kompalli. Segmentation of the liver from abdominal ct using markov random field model and gvf snakes. *International Conference on Complex, Intelligent and Software Intensive Systems, IEEE*, October 2008.
- [6] Z. Kato and J. Zerubia. Markov random fields in image segmentation. *Foundations and Trends in Signal Processing*, 5, 2012.

- [7] Chenyang Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, 7(3):359–369, 1998.
- [8] Monica Lupsor Mihai Socaciu Radu Badea Delia Mitrea, Sergiu Nedevschi. Experimenting various classification techniques for improving the automatic diagnosis of the malignant liver tumors, based on ultrasound images. *3rd International Congress on Image and Signal Processing (CISP2010)*, 2010.

Bibliography

- [1] Mohammad Mannan,P.C. van Oorschot *Secure Public Instant Messaging*. Carlton Paper Publication,2011.
- [2] Li Zhang, Chao Xu, Parth H. Pathak, and Prasant Mohapatra *Characterizing Instant Messaging apps on smartphones*. UCDAVIS Publication,2012
- [3] Prototype Model,
<http://tutorialspoint.com/sdlcprototype.html>
- [4] Black-Box Testing,
<https://wikipedia.org/Blackboxtesting.html>
- [5] Android Developers,
<http://developer.android.com/index.html>
- [6] GitHub,
<https://github.com/>
- [7] The NewBoston videos for Android,
<https://www.thenewboston.com/videos.php>
- [8] StackOverflow,
<http://stackoverflow.com/>
- [9] W3School,
<http://www.w3schools.com/>

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my guide Prof. Dipti Pawade for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. I also take this opportunity to express a deep sense of gratitude to Head of the department, Prof.Sangeeta Nagpure for her cordial support, valuable information and guidance, which helped me in completing this task through various stages. I am obliged to staff members of K. J. Somaiya College of Engineering, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Date:16-11-2015

Student Names

Ajay Mohandas

Rajankumar Bhimprasad

Vijay Yadav

Pradeep Bagrao