

# **Paigaam**

Submitted for the partial fulfilment of the Degree  
for  
Bachelor of Technology  
(Information Technology)



**Submitted By:**

Gurnoor Singh

146023

1411254

**Submitted To:**

Sidharath Jain

Training Co-ordinator

IT Department

---

Department of Information Technology

Guru Nanak Dev Engineering College

Ludhiana 141006

# Company Certificate



**TESTING & CONSULTANCY CELL**  
Providing Technical Services Since 1979  
**GURU NANAK DEV ENGINEERING COLLEGE**  
An Autonomous College u/s 2(f) and 12(B) of UGC Act 1956  
NBA Accredited Programmes under Tier - I ( Washington Accord )  
"A" Grade NAAC Accredited, Punjab Govt. Aided Status, ISO: 9001 : 2008 Certified  
Gill Park, Gill Road, Ludhiana - 141006 Lat 30° 51' 29" N Lon 75° 51' 39"E  
Ph.: 0161-2491193, 5064509 Website : <http://tcc.gnec.ac.in> E-mail : [tcc@gnec.ac.in](mailto:tcc@gnec.ac.in)

No. GNDEC/TCC/SMT/ 10197

Dated: 11.6.17

## TO WHOM IT MAY CONCERN

I here by certify that Mr. Gurnoor Singh University Roll No. 1411254 of Guru Nanak Dev Engineering College, Ludhiana has undergone Industrial & Software Training from June 16, 2017 to August 08, 2017 at our organisation to fulfill the requirements for the award of degree of B.Tech. (IT). He worked under different projects during the training under our supervision and found him sincere and hard working.

(Dr. H. S. Rai)  
Dean Testing & Consultancy

Figure 1:

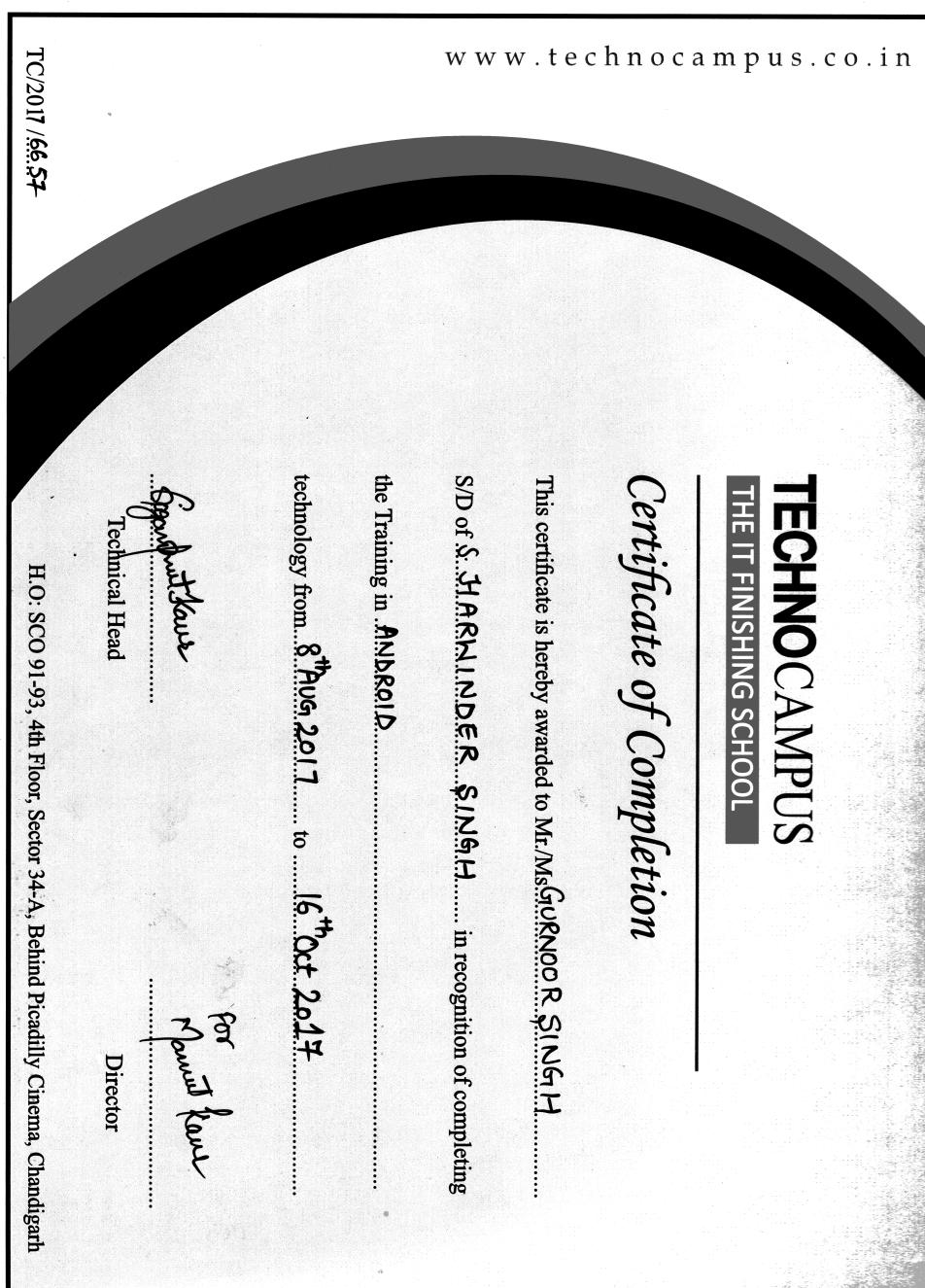


Figure 2:

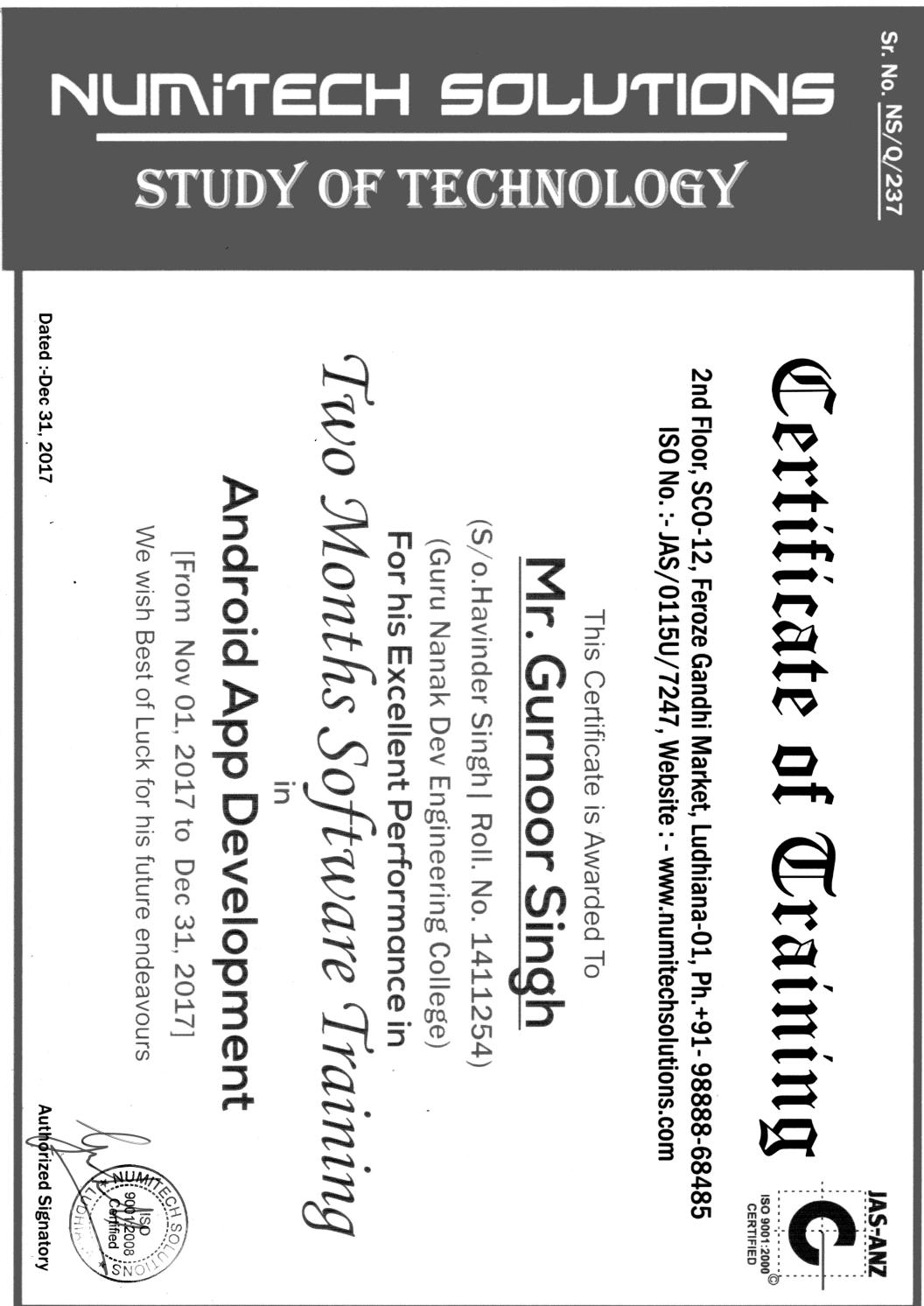


Figure 3:

# **Abstract**

A messaging app is a big undertaking. Making it needs the collaboration of many technologies and computing disciplines. This report represents the background study, technology used development, implementation and testing of a messaging app called Paigaam. Paigaam is a multiplatform app which includes an android app and a webapp. It features chatting through XMPP(Extensible Messaging and Presence Protocol) and the chat is end-to-end encrypted. Users can make their own groups and send images to each other.

The unique thing about this app is that it also features a chatbot called Paigam-Assistant that can respond to multiple queries from students and teachers. The students can ask frequently asked questions such as attendance details in subjects, fees paid status, enrolled courses, teachers details, student details, average percentage in semesters to the bot and get responses.

The App is the one stop solution for all communication needs for the students and faculty members inside and out of the college campus. It is a full-fledged messaging App offering a complete, rich and innovative feature set. Users don't have to suffer through any enraging registration. Log in using your college ID and password and explore the college centric social network.

Here are some of the App's most lavishing features:

- Chat using multiple accounts from a single copy of the App.
- Encrypt your messages over the medium.
- Automatic conversation backup on the server, sparing your precious phone storage.
- Groups for conference chats.
- Share files and documents of any type such as pdf, docx, images etc.
- Save your contacts on your phone and / or with your account and import them to any phone via login.
- Video chat through the webapp.

## **Acknowledgement**

I, student of Guru Nanak Dev Engineering College, Ludhiana, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

The author is highly grateful to Dr. Sehijpal Singh Principal, Guru Nanak Dev Engineering College, Ludhiana for providing her with the opportunity to carry out her Six month Training at Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana.

The author would like to whole heartedly thank Dr. H.S. Rai Dean, Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana who is a vast sea of knowledge and without whose constant and never ending support and motivation, it would never have been possible to complete the project and other assignments so efficiently and effectively.

Finally, I would thanks to all whoever have contributed in this report work

Gurnoor Singh

# Definitions, Acronyms and Abbreviations

## Definitions

Instant Message: An electronic message sent in real time via the Internet and therefore immediately available for display on the recipient's screen.

Chatbot: A chatbot is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test.

## Abbreviations

- XMPP: Extensible Messaging and Presence Protocol
- GNU: GNU's not UNIX
- XML: Extensible Markup Language
- OTR: Off-the-Record
- JDK: Java Development Kit
- SDK: Software Development Kit
- NDK: Native Development Kit
- API: Application Program Interface
- SSL: Secure Sockets Layer
- IDE: Integrated Development Environment
- OMEO: Multi-End Message and Object Encryption
- OpenPGP: Pretty Good Privacy
- SSH: Secure SHell
- QR-Code: Quick Response Code

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction to Organization . . . . .	1
1.1.1	Testing and Consultancy Cell . . . . .	1
1.1.2	Techno Campus . . . . .	2
1.1.3	Numitech Solutions . . . . .	3
1.2	Introduction to Project . . . . .	3
1.2.1	Overview . . . . .	3
1.3	The Existing System . . . . .	4
1.4	User Requirement Analysis . . . . .	4
1.5	Objective of Project . . . . .	5
<b>2</b>	<b>Requirement Analysis and System Specification</b>	<b>6</b>
2.1	Feasibility Study . . . . .	6
2.1.1	Tyes of Feasibility Study . . . . .	7
2.1.1.1	Technical Feasibility . . . . .	7
2.1.1.2	Economic Feasibility . . . . .	8
2.1.1.3	Behavioral Feasibility . . . . .	9
2.2	Software Requirement Analysis . . . . .	11
2.3	Software Requirement Specification . . . . .	12
2.3.1	Functional Requirements . . . . .	12
2.3.1.1	Students . . . . .	12
2.3.1.2	Teachers . . . . .	12
2.3.2	Non-Functional Requirements . . . . .	12
2.3.2.1	Databases . . . . .	12
2.3.2.2	Platform . . . . .	13
2.3.2.3	Web Browser . . . . .	13
2.4	SDLC model to be used . . . . .	13
2.4.1	Spiral Model . . . . .	13
<b>3</b>	<b>Project Design</b>	<b>15</b>
3.1	Product Perspective . . . . .	15
3.2	Product Functions . . . . .	15
3.3	User Characteristics . . . . .	16
3.4	Constraints . . . . .	16
3.5	Data Flow Diagram . . . . .	16
3.6	Flowchart . . . . .	17

3.7	Table Structure . . . . .	18
3.7.1	Paigaam . . . . .	18
3.7.2	Chatbot(Paigaam-Assistant) . . . . .	20
3.8	ER Diagrams . . . . .	22
3.8.1	Assumptions and Dependencies . . . . .	23
3.8.2	Specific Requirements . . . . .	23
3.8.2.1	Software Requirements . . . . .	23
3.8.2.2	Hardware Requirements . . . . .	23
<b>4</b>	<b>Development and Implementation</b>	<b>24</b>
4.1	Development and Implementation . . . . .	24
4.1.1	Introduction to Languages . . . . .	24
4.1.1.1	Java . . . . .	24
4.1.2	Python . . . . .	24
4.1.3	Javascript . . . . .	24
4.1.4	SQLite . . . . .	25
4.2	Any other Supporting Languages or tools . . . . .	26
4.2.1	Android . . . . .	26
4.2.2	Android Studio . . . . .	26
4.2.3	Strophe.js . . . . .	26
4.2.4	BOSH (protocol) . . . . .	27
4.2.5	Ejabberd . . . . .	27
4.2.6	Errbot . . . . .	27
4.2.7	SleekXMPP . . . . .	28
4.3	Ubuntu: An open source OS . . . . .	28
4.4	Introduction To Doxygen . . . . .	29
4.4.1	Installation of Doxygen . . . . .	30
4.5	Introduction to LATEX . . . . .	30
4.5.1	Typesetting . . . . .	31
4.6	Introduction to Github . . . . .	31
4.6.1	What is Git? . . . . .	33
4.6.2	Installation of Git . . . . .	34
4.6.3	Various Git Commands . . . . .	34
4.6.3.1	Create Repositories . . . . .	35
4.6.3.2	Make Changes . . . . .	35
4.6.3.3	Group Changes . . . . .	35
4.6.3.4	Synchronize Changes . . . . .	36
4.7	Working with Server . . . . .	36
4.8	Implementation of project . . . . .	38

4.8.1	Architecture of project . . . . .	38
4.8.2	Working of the Paigaam-Assistant(Chatbot) . . . . .	39
4.9	Test Cases . . . . .	39
4.9.1	Paigaam App . . . . .	39
4.9.2	Paigaam Assistant(Chatbot) . . . . .	43
4.9.3	Paigaam Web Client . . . . .	44
<b>5</b>	<b>Conclusion and Future Scope</b>	<b>48</b>
5.1	Conclusion . . . . .	48
5.2	Future Scope . . . . .	48
	<b>BIBLIOGRAPHY</b>	<b>48</b>

---

## LIST OF FIGURES

1	.....	I
2	.....	II
3	.....	III
1.1	Guru Nanak Dev Engineering College .....	1
1.2	Techno Campus Logo .....	2
1.3	Numitech Solutions Logo .....	3
1.4	Paigaam Logo .....	4
2.1	Spiral Model .....	14
3.1	Level 0 DFD .....	17
3.2	Level 1 DFD .....	17
3.3	Flow Chart .....	18
3.4	Registered Users .....	18
3.5	Users in a Group .....	19
3.6	Pending Messages .....	19
3.7	Privacy Settings defined By User .....	20
3.8	Created Groups .....	20
3.9	Last Seen Information .....	20
3.10	Student Details .....	21
3.11	Course Details .....	21
3.12	Paigaam ER Diagram .....	22
3.13	Paigaam ER Diagram .....	23
4.1	Java Logo .....	24
4.2	Python Logo .....	24
4.3	Javascript logo .....	25
4.4	SQLite Logo .....	25
4.5	Android Logo .....	26
4.6	Android Studio Logo .....	26
4.7	Ejabberd Logo .....	27
4.8	Errbot Logo .....	27
4.9	Ubuntu .....	28
4.10	Doxygen logo .....	29
4.11	Donald Knuth, Inventor Of T <sub>E</sub> X typesetting system .....	30
4.12	Output of the avove program .....	32
4.13	Github Logo .....	32

4.14 Git Logo . . . . .	34
4.15 Server Communication . . . . .	36
4.16 Paigaam Architecture . . . . .	38
4.17 Paigaam-Assistant Working . . . . .	39
4.18 Login Test ID and Password . . . . .	40
4.19 Internet Connectivity Test . . . . .	41
4.20 Sending Message Test . . . . .	41
4.21 Sending Encrypted text/file Test . . . . .	42
4.22 Messaging in Group Test . . . . .	43
4.23 Testing Paigaam Assistant as a Student . . . . .	44
4.24 Testing Paigaam Assistant as a Teacher . . . . .	45
4.25 Paigaam Login Test . . . . .	45
4.26 Paigaam Login Success Test . . . . .	46
4.27 Video Chat Test . . . . .	47
4.28 Chat Accepting Video Chat Test . . . . .	47
4.29 Video chat Accepted Test . . . . .	47

### 1.1 Introduction to Organization

#### 1.1.1 Testing and Consultancy Cell



Figure 1.1: Guru Nanak Dev Engineering College

I attended starting 2 months of my training at TCC i.e. Testing and Consultancy Cell, GNDEC Ludhiana under the guidance of Dr. H.S. Rai Dean Testing and Consultancy Cell from 6 th June 2017 to 8 th August 2017.

Testing and Consultancy Cell was established in the year 1979 with a basic aim to produce quality service for technical problems at reasonable and affordable rates as a service to society in general and Engineering fraternity.

Consultancy Services are being rendered by various Departments of the College to the industry, State Government Departments and Entrepreneurs and are extended in the form of expert advice in design, testing of materials & equipment, technical surveys, technical audit, calibration of instruments, preparation of technical feasibility reports etc. This consultancy cell of the college has given a new dimension to the development programmers of the College. Consultancy projects of over Rs. one crore are completed by the Consultancy cell during financial year 2009-10.

The main goal of this institute is:

- To build and promote teams of experts in the upcoming specialisations.
- To promote quality research and undertake research projects keeping in view their relevance to needs and requirements of technology in local industry.
- To achieve total financial independence.
- To start online transfer of knowledge in appropriate technology by means of establishing multipurpose resource centres.

Ours is a pioneer institute providing Consultancy Services in the States of Punjab, Haryana, Himachal, J&K and Rajasthan. Various Major Clients of the Consultancy Cell are as under:

- Northern Railway, Govt. of India
- Indian Oil Corporation Ltd.
- Larson & Turbo.
- Multi National Companies like AFCON & PAULINGS.
- Punjab Water Supply & Sewage Board

### **1.1.2 Techno Campus**



Figure 1.2: Techno Campus Logo

I attended my next two months of training from 8 th Aug 2017 to 10 Oct 2017 at Techno Campus, Ludhiana. Techno Campus is a respected learning solutions provider. We are dedicated to creating success stories for our customers. Our unique integrated learning solution is a proven approach and can be customised to your individual training needs, be it in the areas of Technical Training, Desktop Applications Training, Professional Development Training.



Figure 1.3: Numitech Solutions Logo

### 1.1.3 Numitech Solutions

The remaining two months of training was attended at Numitech Solutions from 1 st Nov 2017 to 31 st Dec 2017. Numitech Solutions (ISO 9001:2008 Certified) is a prime name in the field of Website Development and Designing Services / 6 Months and 6 Weeks Industrial Training / Engineering College and School Projects for all Streams. Established Since 2010, In business hub (Feroze Gandhi Market) center of Ludhiana, we keep accelerating our company at exponential rate to achieve a good reputation in Market. We Provide Training and Projects to More than 2000 Students each Year.

Our Key Focus is on providing a Best Quality Service, High end Solutions for Industry and a Great Learning Experience with total practical approach for students to relate their skills with industrial benchmarks. We provide Tech Solutions for Lots of Technologies Like Website Design and Development, Software Development, Embedded, Wireless, Mechanical, Automation, PLC, VLSI, MATLAB, Linux, Java, PHP, Solar Projects, Android OS, Robotics etc.

Were a Team of Highly skilled, zealous, enthusiastic, devoted and Experienced people with strong work ethics who believe in doing what we love and giving full freedom for creativity for each project We believe Our success lies in Our Clients and Students success. So, we take each client business and each student Project / Training Very seriously, because fortune of our company is in future of our client.

## 1.2 Introduction to Project

### 1.2.1 Overview

This Service is the one stop solution for all communication needs for the students and staff/faculty members inside and outside of the college campus. It has an android based client that communicates with the server using XMPP protocol (Extensible Messaging and Presence Protocol). XMPP servers are used worldwide for communication, chatting and messaging over internet.

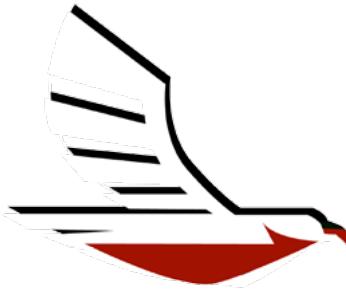


Figure 1.4: Paigaam Logo

A chatbot (also known as a talkbot, chatterbot, Bot, IM bot, interactive agent, or Artificial Conversational Entity) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some chatterbots use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.

The app features a chatbot called Paigaam Assistant which is developed to answer the following queries from students or faculty of the college:

### 1.3 The Existing System

There is no dedicated existing system for this purpose of college centric social network. Although students have their own Facebook and WhatsApp groups for communication with their friends and get knowledge about the activities going on in college.

The existing system has no chatbot to answer student or teacher queries, neither on the website nor in any app

### 1.4 User Requirement Analysis

Although existing system doing their jobs as expected but in some situations, they fail.

1. In case of WhatsApp, some students are not comfortable in providing their phone numbers to whole class. Reason may be anything but that they start

missing out the updates that their friends provide via these groups.

2. Inconsistency of users. This may rear but not everyone has account signed up for all services that others used for communications. Users are distributed over a huge number and variety of communications services available.
3. Complexity. Many applications are complex and that's why user did not use them at their maximum potentials.
4. Most of these services provide a limiting support for document sharing.
5. Users need to go to individual and ask them for their ID/Phone numbers to add them to the groups.
6. Most of the services takes a huge amount of data storage on users devices and did not provide any backup protection.
7. Now days encryption is important but not all are able to provide encryption.
8. Chatbots have changed the way users interact with the computers.

Users need a dedicated service that they trust, rely on and did not need to register individually to use their features. All that needs taken into account and we are tried to provide all such facilities in this single service and is now ready to use by the users

## **1.5 Objective of Project**

It is a full-fledged messaging Application offering a complete, rich and innovative feature set

- To provide users a platform where they can simply log in using their college ID and password and explore the college social network.
- Users can ask frequently asked questions such as attendance, percentage etc. with the help of a chatbot which can save their time and as well as facultys time.
- Teachers can directly share assignments to students and can keep in touch with the students even after the college hours.
- Users can create groups, send/receive documents, images, videos

# CHAPTER 2

## REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

### 2.1 Feasibility Study

This review is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Carrying out a feasibility study involves information assessment, information collection and report writing. The information assessment phase identifies the information that is required to answer the three questions set out above. Once the information has been identified, you should question information sources to discover the answers to these questions. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

A feasibility study is designed to provide an overview of the primary issues related to a business idea. The purpose is to identify any make or break issues that would prevent your business from being successful in the marketplace.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software.

Objectives of feasibility study are listed below:

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

## 2.1.1 Tyes of Feasibility Study

Various types of feasibility that are commonly considered include technical feasibility, economic feasibility, and behavioural feasibility.

### 2.1.1.1 Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. In large engineering projects consulting agencies that have large staffs of engineers and technicians conduct technical studies dealing with the projects. In individual agricultural projects financed by local agricultural credit corporations, the technical staff composed of specialized agricultural engineers, irrigation and construction engineers, and other technicians are responsible for conducting such feasibility studies.

The Technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system. This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. A Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members
- Determines whether the relevant technology is stable and established
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

It is possible to provide a college social network and there are living examples of the same. Many private colleges and universities have their own network from where students can talk to other students and staff.

#### **2.1.1.2 Economic Feasibility**

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

Economic feasibility is the cost and logistical outlook for a business project or endeavor. Prior to embarking on a new venture, most businesses conduct an economic feasibility study, which is a study that analyzes data to determine whether the cost of the prospective new venture will ultimately be profitable to the company. Economic feasibility is sometimes determined within an organization, while other times companies hire an external company that specializes in conducting economic feasibility studies for them.

The purpose of business in a capitalist society is to turn a profit, or to earn positive income. While some ideas seem excellent when they are first presented, they are not always economically feasible. That is, that they are not always profitable or even possible within a company's budget. Since companies often determine their budget's several months in advance, it is necessary to know how much of the budget needs to be set aside for future projects. Economic feasibility helps companies determine what that dollar amount is before a project is ultimately approved. This allows companies to carefully manage their money

to insure the most profitable projects are undertaken. Economic feasibility also helps companies determine whether or not revisions to a project that at first seems unfeasible will make it feasible.

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

With the help of some authorities, the project is affordable. This project is on time investment and very small effort is required to maintain the service up and running. Since the software used for developing this software is open source, the project is highly economical.

#### **2.1.1.3 Behavioral Feasibility**

Behavioral feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. It is a

measure of how well the solution of problems or a specific alternative solution will work in the organization. It is also measure of how people feel about the system. If the system is not easy to operate, than operational process would be difficult. The operator of the system should be given proper training. The system should be made such that the user can interface the system without any problem.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?
- The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

## 2.2 Software Requirement Analysis

Software requirement analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users

and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## **2.3 Software Requirement Specification**

### **2.3.1 Functional Requirements**

#### **2.3.1.1 Students**

1. Login/Logout
2. Join and Create Groups
3. Private Chat with Authenticated Users
4. Send/Receive Files/Photos
5. Join Departmental Groups
6. Update Status such as online/offline/away
7. Ask queries to the chatbot

#### **2.3.1.2 Teachers**

1. Login/Logout
2. Create Groups according to classes
3. Join and create new users
4. Private Chat with authenticated Users
5. Send/Receive Files
6. Update Details
7. Ask queries to the chatbot

### **2.3.2 Non-Functional Requirements**

#### **2.3.2.1 Databases**

A database management system that is available free of cost in the public domain should be used.

### **2.3.2.2 Platform**

It should be software platform independent.

### **2.3.2.3 Web Browser**

We should be able to software the product from any browser.

## **2.4 SDLC model to be used**

There are five phases in this model and the first phase is the planning stage. The planning stage determines the objectives of the project and whether the project should be given the green light to proceed. This is where the proposal submission comes into picture.

After obtaining the approval, the next phase is analysis. Gathering and analysing the system and user requirements is essential for entry to the design step. With the user requirements gathering completed, there is a need to prepare the resources for the project. Be it software or hardware components, careful consideration and selection is to be taken care at this stage. The decision on the appropriate resources to be used is further elaborated under the subsections below. The next step is to design the system and database structure.

Results from the analysis and preparation that were concluded from the previous stage are put into action. With the user requirements in mind, the flow of the system is planned and the user interface is designed to suit their easy navigation needs. In addition, the number of tables, attributes, primary and unique keys of the database is listed. After completing the design, actual coding begins. Database is created and codes are written. Some of the codes required amendments and improvement to it so these are being developed at this fourth stage of the waterfall model. With the development completed, testing will begin.

The codes and database are tested to ensure the results obtained are as intended. More time is spent on both development and testing stages because it is inevitable to have errors and issues and buffer time is allocated for troubleshooting. This system will use spiral model.

### **2.4.1 Spiral Model**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development

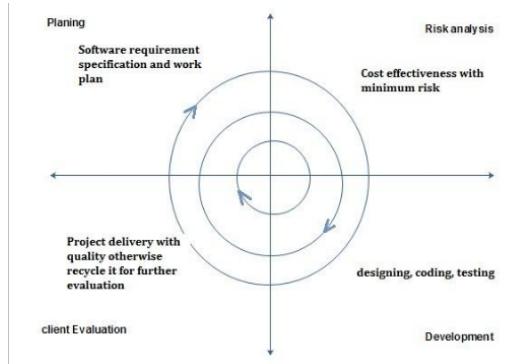


Figure 2.1: Spiral Model

model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral. It is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.

### 3.1 Product Perspective

The product is an open source, community driven service which is licensed under the GNU general Public License. Product Aim is to provide an app offering a complete, rich and innovative feature set. User don't have to suffer through any enraging registration. They can login using their college ID and password and start using the service.

This service mainly is an implementation of client-server model. Where clients are: -

- Android application dedicated to college centric social network.
- Web client enable non-android users to interact and enjoy the services.

And server is Ejabberd.

Ejabberd is an XMPP application server, written mainly in the Erlang programming language. It can run under several Unix-like operating systems such as Mac OS X, GNU/Linux, FreeBSD, NetBSD, OpenBSD and OpenSolaris. Extensible Messaging and Presence Protocol (XMPP) is a communications protocol for message- oriented middleware based on XML (Extensible Markup Language). It enables the near-real-time exchange of structured yet extensible data between any two or more network entities. WhatsApp user the same server and protocol to provide their services.

### 3.2 Product Functions

The system performs the following functions. The functions depend on the users level and permission package, as explained in the user characteristics. Features that this project going to provide:

- Login with user ID or roll number.
- Automatic conversation back up on the server, sparing your precious phone storage.
- Save your contacts on your phone and / or with your account and port them to any phone via login.
- Chat using multiple accounts from a single copy of the App.

- End-to-end encryption with OMEMO.
- Send and receive images as well as other kind of files.
- Indication when your contact has read your message.
- Pre-configured classification into groups for conference chats.

### **3.3 User Characteristics**

Users of this project are the staff and students of the college. We can classify the users in the following categories: -

- Students: Users who use the application on daily basis.
- Teachers: They can use this app to interact with the students and ask queries to the chatbot.
- College Authorities: Authorities can use the application to provide updates to students about the various activities going in the college.

Also, the project is open source so there will be developers that contributes to the project in the future. They can be students or teachers associated to develop the application.

### **3.4 Constraints**

The software has the following constraints:

- Bad internet connectivity.
- Server Traffic or any type of DDOS (Denial of Service) attack.
- Flaw in database design
- The project completed in the duration of 2 months.

### **3.5 Data Flow Diagram**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination

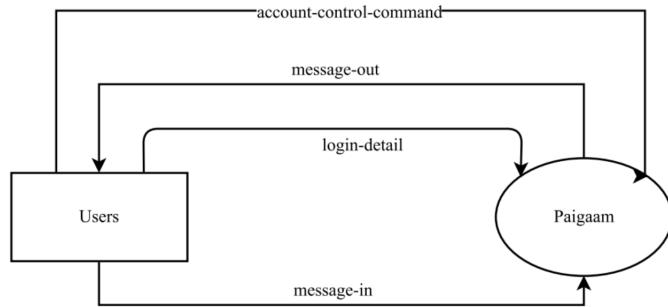


Figure 3.1: Level 0 DFD

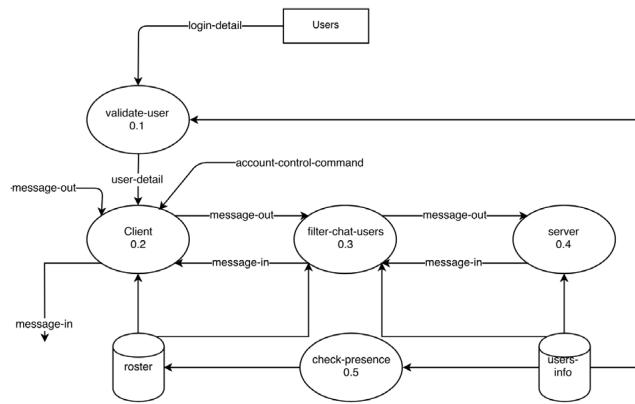


Figure 3.2: Level 1 DFD

### 3.6 Flowchart

A flowchart is a type of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. Today, flowcharts are used for a variety of purposes in manufacturing, architecture, engineering, business, technology, education, science, medicine, government, administration and many other disciplines. This allows anyone to view the flowchart and logically follow the process from beginning to end and following are flowchart of DoS showing flow of control and Data in the software:-

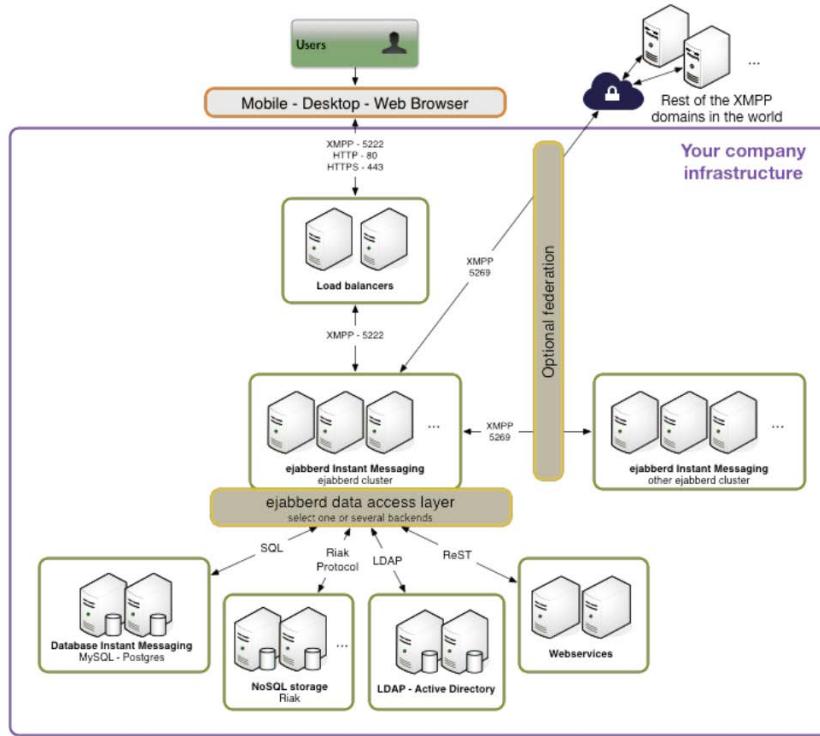


Figure 3.3: Flow Chart

## 3.7 Table Structure

### 3.7.1 Paigaam

Ejabberd SQL database schema is design in order to maintain all the record about the users and messages.

#### Table users

Contains the information required to authenticate users.

Field	Type	Usage
<code>username</code>	string	User
<code>password</code>	string	User password, can be hashed
<code>created_at</code>	timestamp	When the user account was created

Figure 3.4: Registered Users

#### Table roster users

This is a quite complex table, used as a store for a quite complex protocol that is the one defined to manage rosters (groups) and subscriptions. In the common case of two users adding each other as contacts, entries in the roster

table follows a series of steps as they move from a subscription request to the final approval and bi-directional subscription being established. This process can be initiated either by the user, or by the (possible remote) peer.

Field	Type	Usage
username	String	User
jid	String	Contact jid
nick	String	Contact nickname
subscription	Char	'B'=both   'T'=To   'F'=From   'N'=none
ask	char	'S'=subscribe   'U'=unsubscribe   B='both'   'O'=out   'T'=in   'N'=none
askmessage	string	Message to be displayed on the subscription request
server	char	'N' for normal users contacts
subscribe	string	
type	string	"item"
created_at	timestamp	Creation date of this roster entry

Figure 3.5: Users in a Group

### Table spool

Messages sent to users that are offline are stored in this table. Do not confuse this with general message archiving: messages are only temporarily stored in this table, removed as soon as the target user is back online and the pending messages delivered to it.

Field	Type	Usage
username	string	User
xml	blob	Raw packet
seq	integer	Unique, auto increment sequence number.
created_at	timestamp	When the message was stored

Figure 3.6: Pending Messages

### Table privacy\_list\_data

The table is used to store privacy rules.

### Table muc\_room

It is used to store persistent rooms, that is, rooms that must be automatically started with the server.

Field	Type	Usage
username	string	User
xml	blob	Raw packet
seq	integer	Unique, auto increment sequence number.
created_at	timestamp	When the message was stored

Figure 3.7: Privacy Settings defined By User

Field	Type	Usage
name	string	Room name
host	string	Hostname of the conference component
opts	string	Room options, encoded as erlang terms
created_at	timestamp	Creation date

Figure 3.8: Created Groups

### Table last

This table is used to store the last time the user was seen online.

Field	Type	Usage
username	string	User
seconds	string	Timestamp for the last time the user was seen online
state	string	Why user got disconnected. Usually is empty

Figure 3.9: Last Seen Information

### 3.7.2 Chatbot(Paigaam-Assistant)

#### Table student\_details

This table is used to store student details.

Field	Type	Usage
student_roll_no	integer	Student Roll Number
first_name	string	Student First Name
last_name	string	Student Last Name
fee_status	boolean	Student Fee Status
advisor_id	integer	Student's advisor id
semester	integer	Student Current Semester

Figure 3.10: Student Details

### Table Teacher Details

This table stores the details of teachers..

Field	Type	Usage
course_id	integer	Course ID
course_name	string	Name of the Course
t_id	integer	Teacher's ID
course_semester	string	Teacher Phone Number

Figure 3.11: Course Details

## 3.8 ER Diagrams

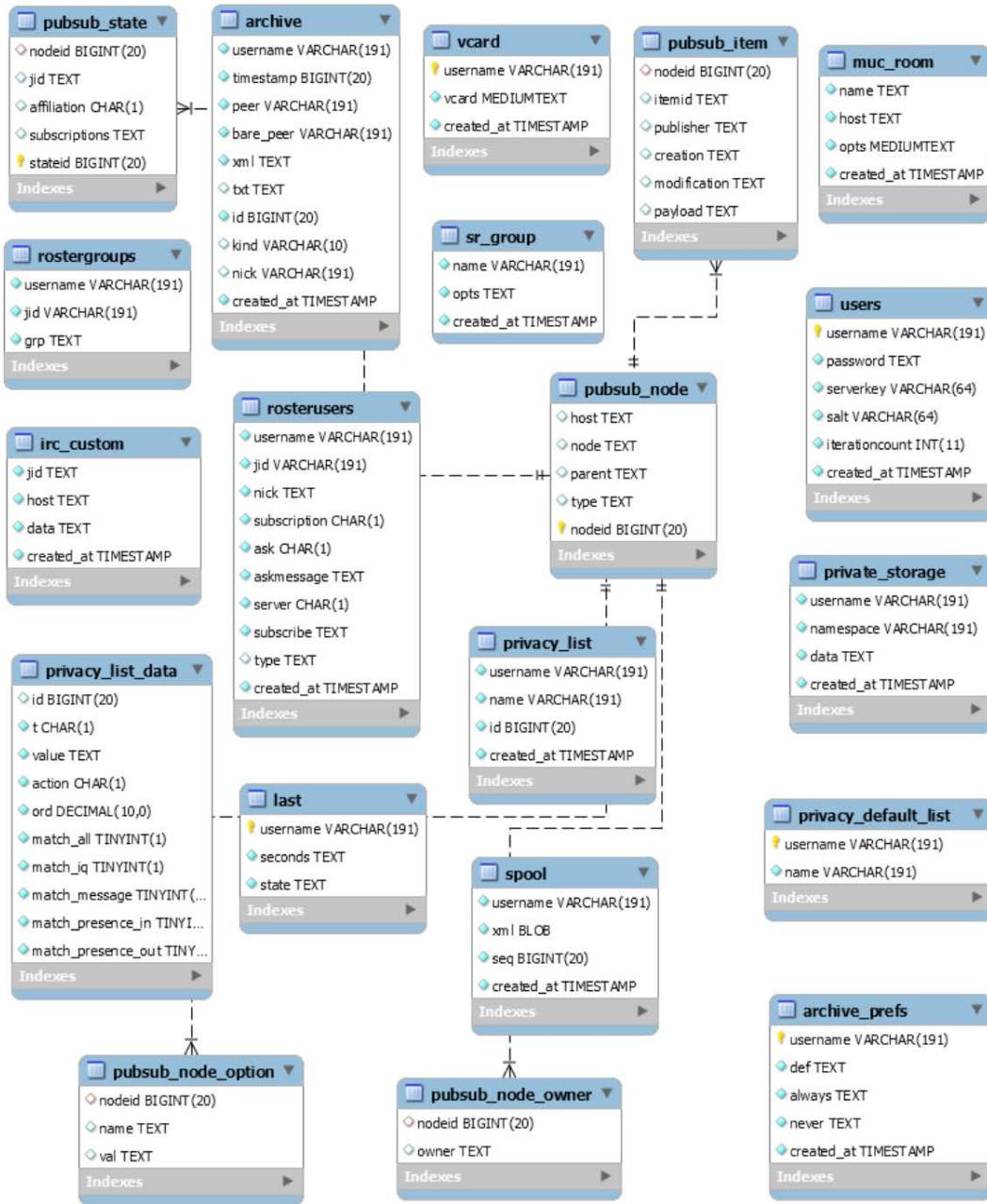


Figure 3.12: Paigaam ER Diagram

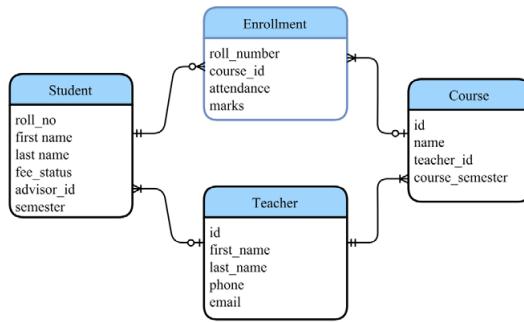


Figure 3.13: Paigaam ER Diagram

### 3.8.1 Assumptions and Dependencies

This system is provisioned to be built on the Android Platform which is highly flexible. Decision regarding which database to use should be taken considering the fact that data being exchanged or stored is large, and the appropriate data management system will yield efficient performance. The software has the following dependencies:

- The system is highly dependent on internet connectivity and required permissions to send/receive the messages to/from the server.
- Android operating system should be updated to at least 4.0 version.

### 3.8.2 Specific Requirements

#### 3.8.2.1 Software Requirements

For developing the application, the following are the Software Requirements:

- Operating System: Windows 10, Android 4.0 KitKat, Linux
- Languages: HTML, CSS, JavaScript & Python, Java.
- Database: SQLite.
- Tools: Web Browser: Chrome, Firefox, Sleek XMPP, Errbot.

#### 3.8.2.2 Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Intel core i3.
- RAM: 4GB.

# CHAPTER 4

## DEVELOPMENT AND IMPLEMENTATION

### 4.1 Development and Implementation

#### 4.1.1 Introduction to Languages

##### 4.1.1.1 Java



Figure 4.1: Java Logo

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform. The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

##### 4.1.2 Python



Figure 4.2: Python Logo

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

##### 4.1.3 Javascript

JavaScript (/dvskrpt/) is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language



Figure 4.3: Javascript logo

specification. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

#### 4.1.4 SQLite



Figure 4.4: SQLite Logo

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

## 4.2 Any other Supporting Languages or tools

### 4.2.1 Android



Figure 4.5: Android Logo

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input.

### 4.2.2 Android Studio



Figure 4.6: Android Studio Logo

Android Studio is the official integrated development environment (IDE) for Android platform development. Android Studio provides the fastest tools for building apps on every type of Android device.

### 4.2.3 Strophe.js

Strophe.js is an XMPP library for JavaScript. Its primary purpose is to enable web-based, real-time XMPP applications that run in any browser. This library uses either Bidirectional-streams Over Synchronous HTTP (BOSH) to emulate a persistent, stateful, two-way connection to an XMPP server or alternatively Web Sockets.

#### 4.2.4 BOSH (protocol)

Bidirectional-streams Over Synchronous HTTP (BOSH) is a transport protocol that emulates a bidirectional stream between two entities (such as a client and a server) by using multiple synchronous HTTP request/response pairs without requiring the use of polling or asynchronous chunking.

#### 4.2.5 Ejabberd



Figure 4.7: Ejabberd Logo

ejabberd is an XMPP application server, written mainly in the Erlang programming language. It can run under several Unix-like operating systems such as Mac OS X, GNU/Linux, FreeBSD, NetBSD, OpenBSD and OpenSolaris. Additionally, ejabberd can run under Microsoft Windows. The name ejabberd stands for Erlang Jabber Daemon (Jabber being a former name for XMPP) and is written in lowercase only, as is common for daemon software.

#### 4.2.6 Errbot



Figure 4.8: Errbot Logo

Errbot is a chatbot, a daemon that connects to your favourite chat service and brings your tools into the conversation. The goal of the project is to make it easy for you to write your own plugins, so you can make it do whatever you want: a deployment, retrieving some information online, trigger a tool via an API, troll a co-worker. Errbot is being used in a lot of different contexts: chatops (tools for devops), online gaming chatrooms like EVE, video streaming chatrooms like livecoding.tv, home security, etc.

#### 4.2.7 SleekXMPP

SleekXMPP is an elegant Python library for XMPP (aka Jabber, Google Talk, etc). SleekXMPP is an MIT licensed XMPP library for Python 2.6/3.1+, and is featured in examples in XMPP: The Definitive Guide by Kevin Smith, Remko Tronon, and Peter Saint-Andre. If youve arrived here from reading the Definitive Guide, please see the notes on updating the examples to the latest version of SleekXMPP.

### 4.3 Ubuntu: An open source OS



Figure 4.9: Ubuntu

During my training, I also got familiar with a great and open source Operating System, Ubuntu. Firstly, it was quite difficult for a regular MS Windows user to port to Ubuntu. I did all of my project work using this vast operating system. Ubuntu is a Debian-based Linux operating system, with Unity as its default desktop environment. It is based on free software and named after the Southern African philosophy of ubuntu (literally, "human-ness"), which often is translated as "humanity towards others" or "the belief in a universal bond of sharing that connects all humanity".

Ubuntu's goal is to be secure "out-of-the box". By default user's programs run with low privileges and cannot corrupt the operating system or other user's files. For increased security, the sudo tool is used to assign temporary privileges for performing administrative tasks, which allows the root account to remain locked and helps prevent inexperienced users from inadvertently making catastrophic system changes or opening security holes.

Doxygen is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus rela-



Figure 4.10: Doxygen logo

tively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

Doxygen supports multiple programming languages, especially C++, C, C#, Objective-C, Java, Python, IDL, VHDL, Fortran and PHP.[2] Doxygen is free software, released under the terms of the GNU General Public License.

## 4.4 Introduction To Doxygen

- Requires very little overhead from the writer of the documentation. Plain text will do, Markdown is support, and for more fancy or structured output HTML tags and/or some of doxygen's special commands can be used.
- Cross platform: Works on Windows and many Unix flavors (including Linux and Mac OS X).
- Comes with a GUI frontend (Doxywizard) to ease editing the options and run doxygen. The GUI is available on Windows, Linux, and Mac OS X.
- Automatically generates class and collaboration diagrams in HTML (as clickable image maps) and L<sup>A</sup>T<sub>E</sub>X (as Encapsulated PostScript images).
- Allows grouping of entities in modules and creating a hierarchy of modules.
- Doxygen can generate a layout which you can use and edit to change the layout of each page.
- Can cope with large projects easily.

#### 4.4.1 Installation of Doxygen

Doxygen can be installed using following commands:

```
$ git clone https://github.com/doxygen/doxygen.git  
$ cd doxygen  
$ ./configure  
$ make
```

### 4.5 Introduction to L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X, I had never heard about this term before doing this project, but when I came to know about it's features, found it excellent. L<sup>A</sup>T<sub>E</sub>X is a document markup language and document preparation system for the T<sub>E</sub>X typesetting program. Within the typesetting system, its name is styled as L<sup>A</sup>T<sub>E</sub>X.

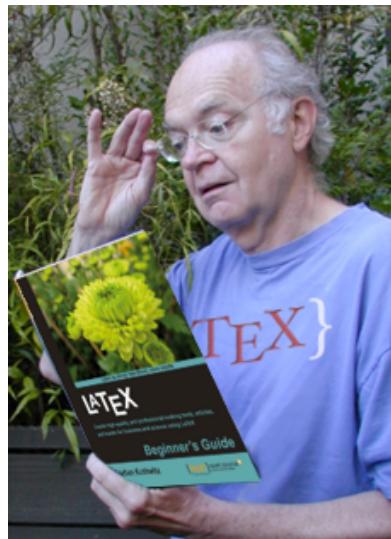


Figure 4.11: Donald Knuth, Inventor Of T<sub>E</sub>X typesetting system

Within the typesetting system, its name is styled as L<sup>A</sup>T<sub>E</sub>X. The term L<sup>A</sup>T<sub>E</sub>X refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in L<sup>A</sup>T<sub>E</sub>X, a .tex file must be created using some form of text editor. While most text editors can be used to create a L<sup>A</sup>T<sub>E</sub>X document, a number of editors have been created specifically for working with L<sup>A</sup>T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats

to PDF, L<sup>A</sup>T<sub>E</sub>X is used because of the high quality of typesetting achievable by T<sub>E</sub>X. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.

L<sup>A</sup>T<sub>E</sub>X is intended to provide a high-level language that accesses the power of T<sub>E</sub>X. L<sup>A</sup>T<sub>E</sub>X essentially comprises a collection of T<sub>E</sub>X macros and a program to process L<sup>A</sup>T<sub>E</sub>X documents. Because the T<sub>E</sub>X formatting commands are very low-level, it is usually much simpler for end-users to use L<sup>A</sup>T<sub>E</sub>X.

#### 4.5.1 Typesetting

In preparing a L<sup>A</sup>T<sub>E</sub>X document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the L<sup>A</sup>T<sub>E</sub>X system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
\maketitle
\LaTeX{} is a document preparation system
for the \TeX{} typesetting program.
\end{document}
```

### 4.6 Introduction to Github

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.



Figure 4.12: Output of the avove program



Figure 4.13: Github Logo

GitHub offers both paid plans for private repo handle everything from small to very large projects with speed and efficiency. ositories, and free accounts, which are usually used to host open source software projects. As of 2014, GitHub reports having over 3.4 million users, making it the largest code host in the world.

GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub account in order to qualify for a job.

The Git feature that really makes it stand apart from nearly every other Source Code Management (SCM) out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of

those lines of development takes seconds.

This means that you can do things like:

- Frictionless Context Switching.

Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.

- Role-Based Code lines.

Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.

- Feature Based Work flow.

Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.

- Disposable Experimentation.

Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

#### 4.6.1 What is Git?

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.



Figure 4.14: Git Logo

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free and open source software distributed under the terms of the GNU General Public License version 2 to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

#### 4.6.2 Installation of Git

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

```
$ sudo apt-get update
```

```
$ sudo apt-get install git
```

This will install the git on your pc or laptop.

#### 4.6.3 Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line

instructions are:-

#### 4.6.3.1 Create Repositories

Start a new repository or obtain from an exiting URL

**\$ git init [ project-name ]**

Creates a new local repository with the specified name

**\$ git clone [url ]**

Downloads a project and its entire version history

#### 4.6.3.2 Make Changes

Review edits and craft a commit transaction

**\$ git status**

Lists all new or modified files to be committed

**\$ git diff**

Shows file differences not yet staged

**\$ git add [file ]**

Snapshots the file in preparation for versioning

**\$ git commit -m "[descriptive message ]"**

Records file snapshots permanently in version history

#### 4.6.3.3 Group Changes

Name a series of commits and combine completed efforts

**\$ git branch**

Lists all local branches in the current repository

**\$ git branch [branch-name ]**

Creates a new branch

**\$ git checkout [branch-name ]**

Switches to the specified branch and updates the working directory

**\$ git branch -d [branch-name ]**

Deletes the specified branch

#### 4.6.3.4 Synchronize Changes

Register a repository bookmark and exchange version history

**\$ git fetch [bookmark ]**

Downloads all history from the repository bookmark

**\$ git merge [bookmark /[branch]]**

Combines bookmarks branch into current local branch

**\$ git push [alias [branch]]**

Uploads all local branch commits to GitHub

**\$ git pull**

Downloads bookmark history and incorporates changes

## 4.7 Working with Server

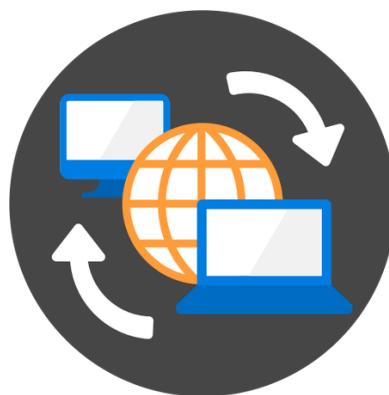


Figure 4.15: Server Communication

I had done the whole project on ubuntu server and had also learnt about making your system a server.

What is a Remote Server?

In simple words its nothing much but a Computer that is not attached to a users keyboard but over which he or she has some degree of control (like can see data of that computer, can retrieve or send data etc.)

For going deep you need to know about ssh (Secure Shell).

I had done it using Mosh. There are few terms related to this :

- SSH: It is a Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer.
- MOSH: It is a software tool used to connect from a client computer to a server over the Internet, to run a remote terminal.
- Tmux: tmux is basically a terminal multiplexer. It is used so that within one terminal window we can open multiple windows and split-views.
- OpenSSH: It is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling or transferring files between computers. Traditional tools used to accomplish this is telnet which is not much secure.

In Unix, you can use SCP (the scp command) to securely copy files and directories between remote hosts without starting an FTP session or logging into the remote systems explicitly.

The scp command uses SSH to transfer data, so it requires a password.

Some of the useful commands in this for checking errors or for other purposes are:

- ll: This command is used to list the detail information of files and folder of a current directory.
- tail -f error.log: This is used for checking errors.
- sudo apt-get install openssh-server
- sudo vim /etc/ssh/sshdconfig  
(To edit this as per your preferences. But first take a backup of this file for later default configurations if needed.)
- sudo restart ssh  
(To check your ssh daemon is running or not.)
- ps -A — grep sshd  
(This command should produce a line like this:  
some-number ? 00:00:00 sshd)

- ssh user@hostip  
(To enter into a remote server from some other system. )

## 4.8 Implementation of project

### 4.8.1 Architecture of project

The following is the architecture of the Paigaam app. It contains the following modules:

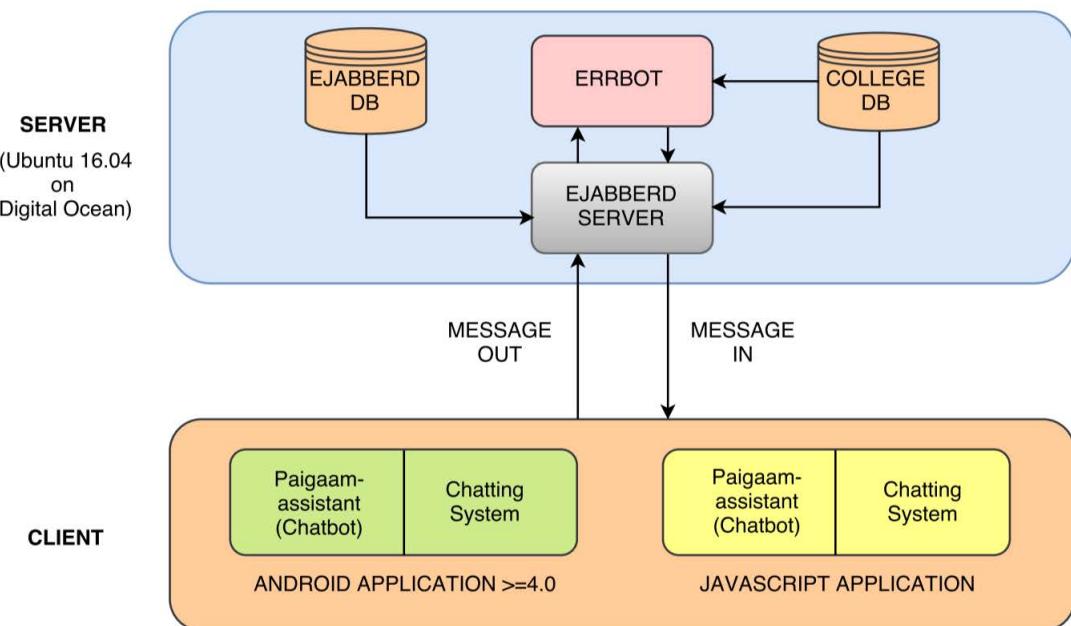


Figure 4.16: Paigaam Architecture

**Server** The server is including the following modules:

- Ejabberd Server: Ejabberd is an XMPP server. It manages the user chats and user accounts and is connected to ejabberd database.
- Errbot: This is the brain behind the Paigaam assistant and is used to manage bot requests. It is connected to a SQLite database.

**Client** The client includes the following modules:

- Android App(Paigaam): The app has chatting features and a chatbot to answer user queries.
- JavaScript Web App: This is the web based client of Paigaam.

#### 4.8.2 Working of the Paigaam-Assistant(Chatbot)

The chatbot working is demonstrated in the following figure:

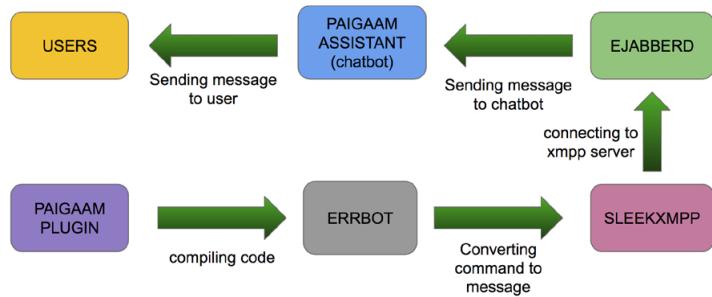


Figure 4.17: Paigaam-Assistant Working

- **Errbot** The Errbot runs the python scripts and convert messages in XMPP format with the help of SleekXMPP.
- **SleekXMPP** SleekXMPP acts as an interface between Errbot and ejabberd and helps Errbot to send/receive messages to/from the ejabberd XMPP server.
- **Ejabberd** The Ejabberd XMPP server is responsible for the messaging service of the app.
- **Paigaam Assistant** The Paigaam assistant(chatbot) sends and gets messages from the ejabberd server and shows them to the users.
- **Users** The users interact with the chatbot to ask queries and get the answers.

#### 4.9 Test Cases

The test cases include the testing of the Paigaam app, the Paigaam assistant(Chatbot) and as well as the web client of Paigaam.

##### 4.9.1 Paigaam App

###### 1. Login

**Description:** A non-registered user should not be able to successfully login into the app.

**Precondition:** The user must not be registered with an email address

and password.

**Test Steps:**

- Open the app.
- Click on Options and then Manage Accounts.
- Click on Add account icon.
- Enter the wrong registered roll id and password and click on next.

**Expected Result:** A message is displayed saying unauthorized password and/or server not found.

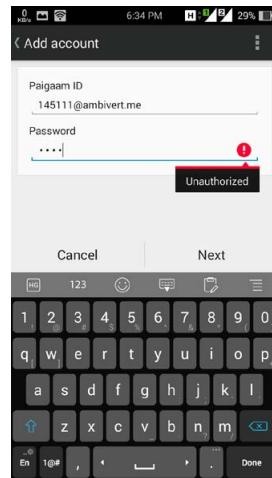


Figure 4.18: Login Test ID and Password

2. Internet Connectivity

**Description:** The user should be shown a message if the internet is not active.

**Precondition:** Internet is not connected.

**Test Steps:**

- Open the app.
- Click on Options and then Manage Accounts.

**Expected Result:** A message is displayed saying No Connectivity under the account name.

3. Sending message to a contact

**Description:** A registered user should be able to send a message to a registered contact.

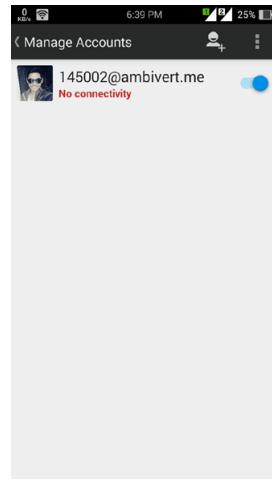


Figure 4.19: Internet Connectivity Test

**Precondition:** The user must already be registered with an email address and password and should not be connected to the internet  
**Test Steps:**

- Open the app.
- Click on the name of the contact to send a message
- Type the message in the field.
- Click on the send button

**Expected Result:** A message state 'waiting' is shown below the sent message until the internet is connected.

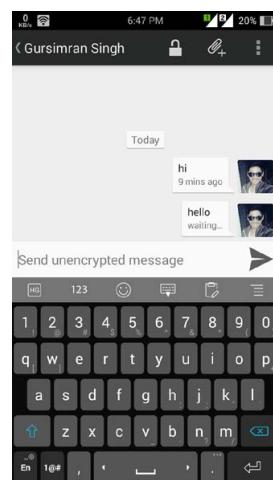


Figure 4.20: Sending Message Test

4. Sending an encrypted message/file to a contact using OMEMO (Multi-End Message and Object Encryption).

**Description:** A registered user should be able to send an encrypted message to a contact.

**Precondition:** The user must already be registered with an email address and password and internet is connected and OMEMO encryption is selected.

**Test Steps:**

- Open the app.
- Click on the name of the contact to send a message.
- Select encryption method to OMEMO.
- Select the file to send.
- Click on the send button

**Expected Result:** The file/image is sent successfully with an encryption lock sign below the message.



- Click on the name of the group to send a message.
- Click on the send button

**Expected Result:** The message is sent successfully..



Figure 4.22: Messaging in Group Test

#### 4.9.2 Paigaam Assistant(Chatbot)

- Sending queries to the chatbot as a student and getting the required results.

**Description:** A registered student should be able to send queries to the chatbot and get the appropriate results.

**Precondition:** The user must already be registered with an email address and password and internet is connected.

**Test Steps:**

- Open the app and click on Paigaam-Assistant from the contact list.
- Type the below query and press send button:
  1. Hi
  2. My Fee Status
  3. My attendance
  4. My marks in 7th semester
  5. Teacher details

**Expected Result:** The chatbot should reply with the correct message corresponding to the query.

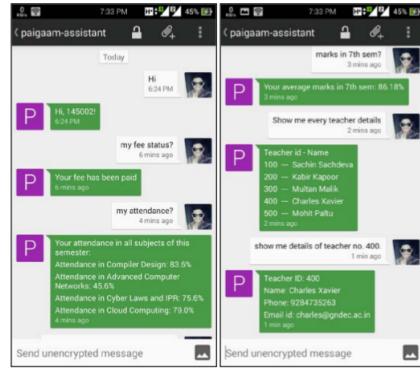


Figure 4.23: Testing Paigaam Assistant as a Student

- Sending queries to the chatbot as a teacher and getting the required results.
- Description:** A registered teacher should be able to send queries to the chatbot and get the appropriate results.

**Precondition:** The teacher must already be registered with an email address and password and internet is connected.

#### Test Steps:

- Open the app and click on Paigaam-Assistant from the contact list.
- Type the below query and press send button:
  1. Hi
  2. My courses
  3. List students under course number 200
- Click on the send button

**Expected Result:** The chatbot should reply with the correct message corresponding to the query.

### 4.9.3 Paigaam Web Client

- Login and messaging
- Description:** A registered user should be able to login into the app and send messages to contacts.

**Precondition:** Internet should be connected and modern browser such as Chrome/Firefox should be used.

#### Test Steps:

- Open a web browser and open the URL <https://ambivert.me/jsxc/example/>
- Enter the registered ID and password
- Click on Login Button

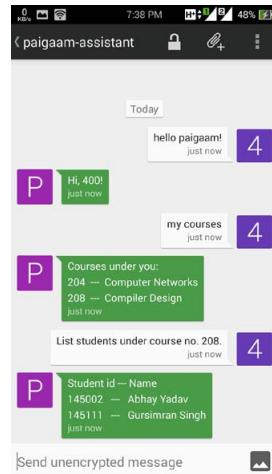


Figure 4.24: Testing Paigaam Assistant as a Teacher

- Once logged in, click on a contact name
- Send message.

**Expected Result:** The user is successfully logged in and the main interface of the app is shown.

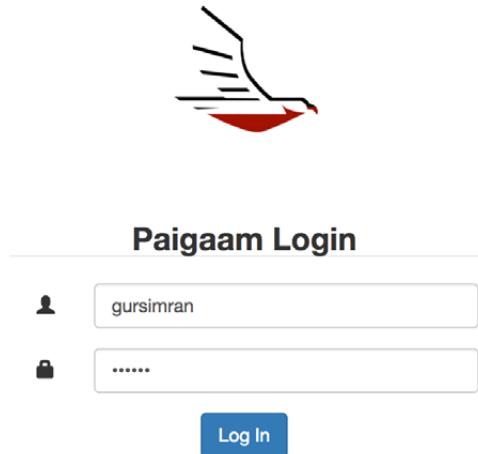


Figure 4.25: Paigaam Login Test

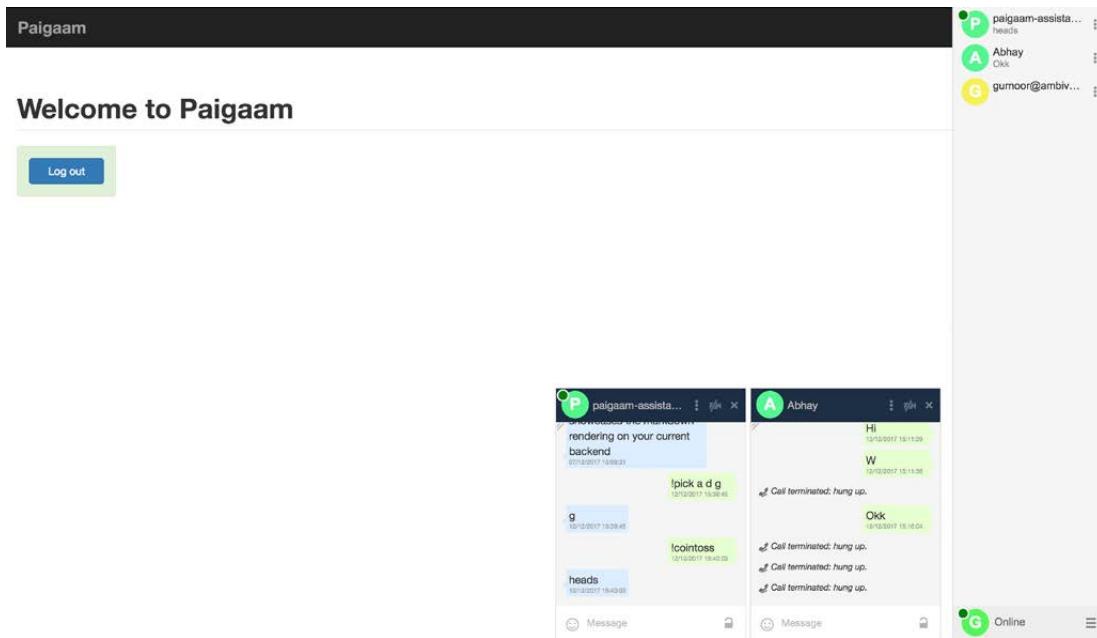


Figure 4.26: Paigaam Login Success Test

- Video Chat

**Description:** A registered user should be able to Video chat with a contact.

**Precondition:** Internet should be connected and modern browser such as Chrome/Firefox should be used.

**Test Steps:**

- Open a web browser and open the URL <https://ambivert.me/jsxc/example/>
- Enter the registered ID and password
- Click on Login Button
- Once logged in, click on a contact name
- Click on a contact name and then on the video chat button

**Expected Result:** The video chat request is sent to the contact and video streaming is possible..

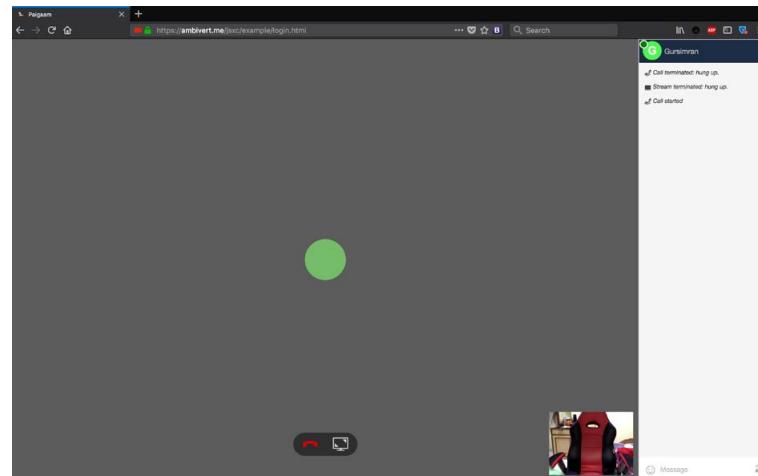


Figure 4.27: Video Chat Test

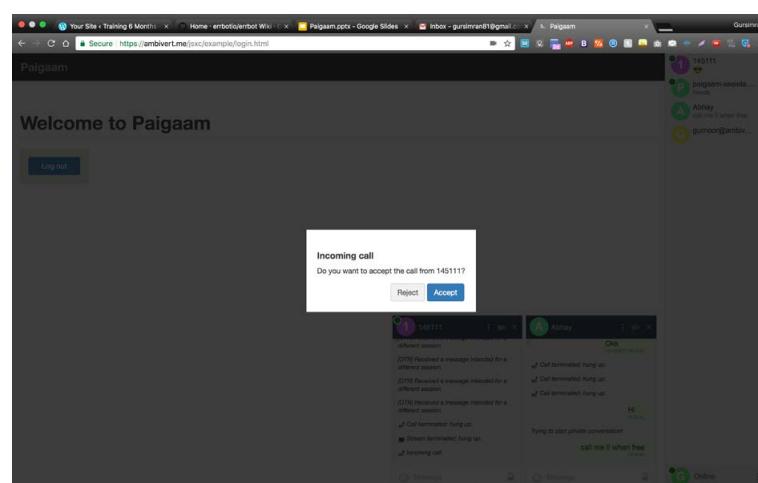


Figure 4.28: Chat Accepting Video Chat Test

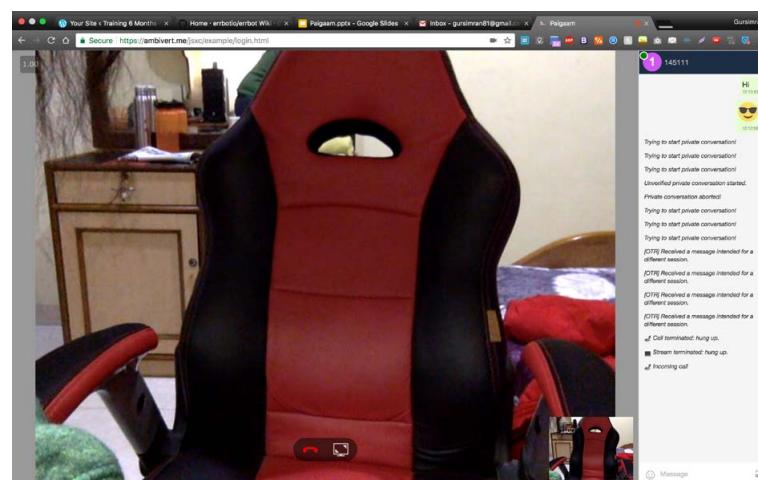


Figure 4.29: Video chat Accepted Test

## CONCLUSION AND FUTURE SCOPE

### 5.1 Conclusion

To summarise in computing, a web application or web app is a clientserver software application in which the client runs in a web browser. In this project a web application and an Android app is developed which runs on the client-server modal. This application can be useful in colleges and similar environments where there is a need for constant communication between teachers and students.

### 5.2 Future Scope

The development of this college centric social network is not over yet. New feature can be added and there is a lot that can be still possible in this project. As this project is connected to open source community it has a lot of scope for future improvements and additions as other individuals can also contribute in it and add additional functionality.

The following are some of the features which can be added into the project:

1. Calling both Voice and Video.
2. Inbuilt Document Viewer to view shared document.
3. Real-time Document editing integration that work like google docs and help user to share.
4. Music Player integrated in the application that remove the need to have a third-party music.
5. Voice Recognition to command the chatbot.
6. The chatbot can be made smart and can work on machine learning models in order to learn from the user queries.

## BIBLIOGRAPHY

- [1] Klaus, JSXC, Dec 2017. [Online]. Available: <https://www.jsxc.org>.
- [2] N. Fritz, sleekxmpp, 2017. [Online]. Available: <http://sleekxmpp.com/>.
- [3] Errbot Documentation, 2016. [Online]. Available: <http://errbot.io/en/latest/index.html#user-guide>.
- [4] XMPP ORG, 2017. [Online]. Available: <https://xmpp.org/>.
- [5] Ejabberd Documentation, 2017. [Online]. Available: <https://docs.ejabberd.im/>.
- [6] Conversatoins, 2016. [Online]. Available: <https://conversations.im/>.
- [7] P. S.-A. a. R. T. Kevin Smith, XMPP: The Definitive Guide, O'Reilly Media, Inc, 2009.