بسم الله الرحمن الرحيم

INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

Department of Computer Science & Engineering (CSE)

A Project proposal on

[Chat Application called “Affix”]

Submitted By:

Name:	Nurul Karim Symon	A.N.M Ahtesham Uddowla	Abdur Rahman
ID:	C183050	C191087	C191111
Signature:			

Submitted To:

Abdul Kadar Muhammad Masum, Ph.D.
Associate Professor of CSE
Department of Computer Science & Engineering (CSE)
International Islamic University Chittagong (IIUC)

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Chatting is now-a-days very useful to express our ideas as well as receive others' ideas on any topic. Chats reflect the recent trends of the society. Sometimes, it is possible to meet eminent people in chatting and have their advice. Affix is a graphical chatting application that makes chatting a pleasant experience. It has excellent features that make any user do whatever he wants while chatting.

1.2 Motivation

The motivation that works behind the project is to provide wide range opportunity to connect with others. Connectivity is the main motivation of our project. With this Chat application people will be able to chat, call, share files within few seconds.

1.3 Existing App

There might have many websites related to this topic but the project we are doing is slightly different. Here, we tried to discuss about the existing system in the world and also show the limitation and solution which have been solved in our Chat application.

❙ Telegram

Advantages of Telegram:

- This app is easy to use.
- Extremely fast messaging service.
- It has no ads, and it is completely free.
- It doesn't contain additional premium content

Limitation of Telegram:

- We cannot know the status of the contacts
- It has not multiple selections when you want to send more than one file
- The main drawback is not just showing your contacts

Slack

Advantages of Slack:

- This app is easy to use.
- Extremely fast messaging service.
- It has no ads, and it is completely free.
- Your data within the platform is always protected thanks to great security.

Limitations of Slack:

- Your conversations get deleted after 14 days
- There is minimum file storage.
- It can be a distraction. If you're not careful, you can easily get sucked into every discussion.

1.4 Objective of Affix:

The project objectives are:

- To make the chatting easy.
- To provide multiple features
- To ensure security while chatting
- To increase chat app engagement through Voice call, Video call.
- To share file
- To offer end-to-end encryption for voice calls & optional end-to-end encrypted secret chats between two online users

CHAPTER 2

METHODOLOGY

Methodology is a lot of practices. This term might be utilized to allude to rehearses which are broadly utilized over an industry or logical order, the procedures utilized in a specific research study, or the strategies used to achieve a specific task. Individuals may likewise utilize the expression "methodology" to allude to the investigation of such strategies, as opposed to the techniques themselves.
A methodology can be considered to incorporate numerous strategies, each as applied to different features of the entire extent of the philosophy. The exploration can be separated between two sections; they are subjective research and quantitative research. The Project Management Methodology empowers the conveyance association to handle these undertakings exhaustively, systematically, and in an incorporated way for adequate hazard. Along these lines, it gives benefits at the key, strategic and operational levels.

2.1 Process Model

Procedure is a movement, which works on an article and changes its state. Model is the graphical portrayal of an article. In this way, a procedure model shows the exercises of programming graphically. We need procedure model since procedure is significant than item (software). On the off chance that the procedure is great, the item likewise is great.

2.2 Recent trends in Software Process Model

1. Waterfall model
2. Incremental Model
3. Spiral Model
4. Prototyping:

    a) Throw away prototyping process model.
    b) Evolutionary prototyping process model.

2.3 A particular Process model for "Affix"

While developing the "**Affix**", we have followed Spiral model,

What is 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 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.

Phases in Spiral model-

- Requirement Analysis
- Design
- Coding (construct or build)

• Evaluation and Risk Analysis

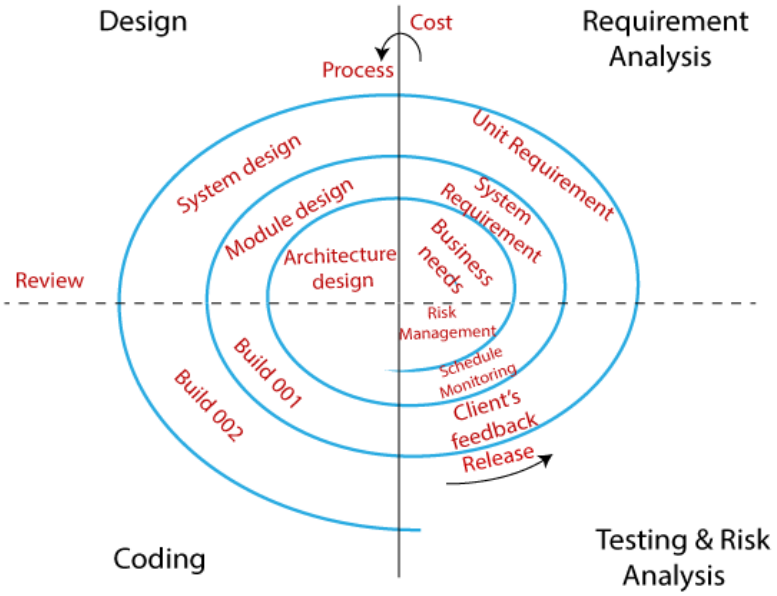

Figure 2.1: Spiral model Process

Requirements Analysis: The spiral model process starts with collecting business needs. In this, the following spirals will include the documentation of system requirements, unit requirements, and the subsystem needs. In this stage, we can easily understand the system requirements because the business analyst and the client have constant communication. And once the cycle is completed, the application will be deployed in the market.

Design: The second stage of the spiral model is designed, where we will plan the logical design, architectural design, flow charts, decision tree, and so on.

Coding (construct and design): After the compilation of the design stage, we will move to our next step, which is the coding stage. In this, we will develop the product based on the client's requirement and getting the client's feedback as well. This stage refers to the construction of the real application in every cycle.

And those spirals had an excellent clarity of the requirements, and the design details of an application are known as the build with having version numbers. After that, these builds are transferred to the client for their responses.

Testing and Risk Analysis: Once the development is completed successfully, we will test the build at the end of the first cycle and also analyze the risk of the software on the different aspects such as managing risks, detecting, and observing the technical feasibility. And after that, the client will test the application and give feedback.

We Prefer Waterfall model, because of its following advantages: -

• Flexible changes are allowed in spiral model.

• The development can be distributed into smaller parts.

• The customer can use the application at an early stage also.

• More clarity for Developers and Test engineers

• It will provide the wide use of prototypes.

Limitation of Spiral model:

• It is not suitable for the small and low-risk product because it could be costly for a smaller project.

• It is a traditional model, and thus developers only did the testing job as well.

• There is no requirement of review process and no parallel deliverables allowed in the spiral model.

• In the spiral model, management is a bit difficult; that's why it is a complex process.

• The maximum number of intermediate phases needs unnecessary paperwork.

When Spiral model should be followed:

- When there is a budget constraint and risk evaluation is important.

- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which are usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

2.3.1 Requirement Specification

A specification can be viewed as a contract between users and software developers, which defines the desired (functional/service) of the software artifacts [and other properties] of its performance, reliability etc. (non-functional) without showing how much functionality is going to be achieved.

2.3.2 Why Requirement Specification is needed?

Requirement Specifications needed because: -

1. It is more precise description of the system functionality and the constraints on its operation.

2. It can be seen as a basis for contracts between the system developer and customers.

2.4. Requirement Definition

Requirement definition provides a detailed and accurate description of the needs of the user, together with a prioritization of those needs. Here feasibility and early system description are used as input. Through the output we get the system user, statement of requirement and system scope.

2.4.1 Requirement Specification of “Affix”

Requirements There are mainly two types of requirements.

• Functional Requirement

• Non-Functional Requirements

2.4.2 Functional Requirements

This section will cover the functional requirements of the chat application.

• **Login Menu:** This functional requirement prompts the user to register for the chat application, log in, or exit the program.

• **Register function:** This aspect of the login menu will ask the user for the client's name, username, and password. The system will check if the username is available or not. It will close if the username is already taken and will go back to the main login menu.

• **Photo Upload:** The user can upload his/her photo during registration.

• **Who is Online:** This aspect will show who is online and will give the user the ability to click on a user and send a message to that user.

• **Search Option:** Users can search for other users using the search option.

• **One on One Chat:** Users will be able to communicate one on one. No one can see their chat.

• **Different Browsers:** The user can open the application with different browsers. Even opening in different browsers, users won’t get logged out.

- **Push notifications:** You don't want your users to have to open your app every few minutes to check for new messages. That would create a terrible user experience, especially on mobile. Push notifications to notify users of a new incoming message or reaction have become a must-have feature.
- **Sharing of media files:** Gone are the days of plain text messages. (Remember T9? My, how far we've come.) Gifs, emojis, audio, images and video files are the new standard. So, your app will need local media storage along with cloud storage for fast retrieval.

• **Login / Logout function:** This aspect will ask for the username and password. Errors will occur if there is a blank, the username doesn't exist, or the password doesn't match with the username. If the username and password match, you are online and able to message anyone else online. This aspect will also allow you to log out of the chat application and go back to the login menu.

• **Public Host:** We will make it in Public Host. Not Private Host.

2.4.3 Non-Functional Requirements:

These are the nonfunctional requirements of the chat application. This is basically the section that deals with the quality of the chat application rather than the functionalities of the application.

• **User Friendly:** The chat application needs to be user-friendly when using its user interface.

• **Performance and stability:** This will be a web application where all the users can access the website from anywhere in the world. And an admin team will monitor all users. So, a huge number of users can act at a time and can access the system all features at a time and not affect the system performance.

Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

Software Quality Attributes

Availability: The flight should be available on the specified date and specified time as many customers are doing advance reservations.

Correctness: The flight should reach start from correct start terminal and should reach the correct destination.

Maintainability: The administrators and flight in chargers should maintain correct schedules of flights.

Usability: The flight schedules should satisfy a maximum number of customers' needs

2.4.2 User list of Affix

There are 2 types of users in Affix. They are:

- Administrator/Admin
- User

Administrator/Admin:

Orders are stored in a database. The Administrator of their website can delete and refuse access to any user. He sees all the orders of the website and is in charge of managing the website. He makes sure that the seller and buyer fulfill their part of service before crediting the user's account. The administrator will have all the rights to create, delete and update the databases. The admin will have an interface to the product table and will have another interface to manage orders and see the status of each order. He will be able login into the membership table and manage members and also send mails to them updating them on different types of information. Information of each user is stored after an order is made. So to get the history of the past orders. The order and product database are queried to get

the details of the past order and sorted by the last order they made. After a sale process the parties involved in the process can give feedback to each other. This feedback that will be public on the website.

User:

User can use this website without any condition. But they need to registration first. After registration they can login to chat. They can also update their profile information. If user has any complain or suggestion they can contact with the admin/administrator. The user interface should be easily understood and operated by user.

2.5 Requirement Validation

Requirement validation is concerned with showing that the requirements actually define the system, which the customer wants. It has much in common with analysis as it is concerned with finding problems with the requirements. However, they are distinct processes since validation should be concerned with a complete draft of the requirements document whereas analysis involves working with incomplete requirements.

2.5.1 Requirement Validation in software process

During the processing period of software, different types of checks must be carried out on the requirements in the requirements document. These checks include-

1. Validity check: A user may think that a system is needed to perform certain functions. However, further thought and analysis may identify additional or different functions that are required. For example, in the **"Affix"**, the system has diverse users with different needs and requirements. However, the software must be arranged in such a way that the services are provided across the user community by maintaining and checking validity.

2. Consistency check: Requirements in the document should not conflict. For example, in the **"Affix"**, when we will deal with unregistered members, there should not be contradictory constraints or different descriptions of the same system function.

3. Completeness check: The requirement document should include requirements, which define all functions and constraints intended by the system user.

4. Realism check: Using knowledge of existing technology, the requirements should be checked to ensure that they could actually be implemented.

2.5.1 Requirement Validation in software process

Affix is validated. We validate our App with user requirements until it satisfies the needs. The image below represents design validation process-

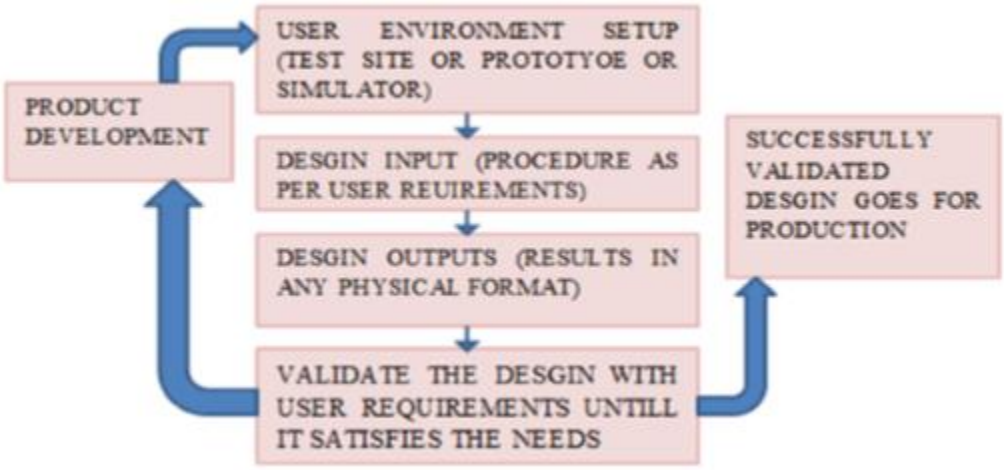

Figure 2.2: Design validation process of Affix

2.6 Service and data input

- User can give their rating based on app performance.
- Admin has the power to access the whole system.
- User can update profile.

Data input

- To Login admin and User email id and password is required.
- Admin can add, edit, delete any product, information etc.

2.7 Tools

Hardware/Software Required:

There are three types of users available in the project.

- Admin: With full access
- User: With limited access

Language:

- **Front-end:** HTML, CSS, React.JS
- **Back-end:** JavaScript, Node.JS

Database:

- MongoDB

Hardware Requirement:

For our system, minimum hardware requirements are

- PC: Intel core i3
- Processor: 64 bits
- Ram: 4GB
- Hard Disk: 500 GB
- Keyboard & Meuse

Software Requirement:

- Windows Operating System
- VS Code
- Android Studio & Any Browser

2.8 Feasibility study

Feasibility studies plan to equitably and soundly reveal the qualities and shortcomings of the current business or proposed adventure, openings and dangers as exhibited by nature, the assets required to bring through, and at last the possibilities for progress.

Economic Feasibility

It is the investigation of monetary advantages of this product. All the more usually known as cost/advantage examination, the method is to decide the advantages and reserve funds that are normal from an up-and-comer framework and contrast them and expenses. In the event that advantages exceed costs, at that point the choice is made to plan and execute the framework. A business person should precisely gauge the expense versus benefits before making a move. These sorts of applications spare our time.

Technical Feasibility

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement. The proposed system can run on any mobile OS which engineering is most interesting. Operational Feasibility 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.

Schedule Feasibility

A project will fall flat on the off chance that it takes too long to be in any way finished before it is valuable. Commonly, this implies evaluating to what extent the framework will take to create, and in the event that it tends to be finished in a given time frame utilizing a few techniques like compensation period. Calendar possibility is a proportion of how sensible the task timetable is. Our Schedule Feasibility is simply time accommodation of the task. In any case, some of the time political viciousness, terrible climate produced results to present our task due time.

2.8.1 Feasibility Study of Affix

This segment will talk about on the possibility investigation of the product interfaces plan and framework Functionalities of **Affix**. This is to recognize whether our website and the idea of the framework meet the potential clients. The attainability study will be surveyed utilizing a few parameters to check the framework prerequisites in various classifications. During the arranging stage, understanding client's necessities is significant and at times these significant angles are deliberately overlooked. Also, the significance of the client's feeling on an advanced item has been checked. The essential practicality study parameters only spotlight on estimating the effect of creating another item towards an organization's work process. The essential parameters are innovation possibility, economy achievability, operational practicality and lawful plausibility. Regardless, the assessment to survey clients' necessity towards another thought isn't featured here, despite the fact that clients assume significant jobs in deciding if an item merit purchasing or not. A decent item should be agreeable to utilize and consent to every one of the necessities expressed by the potential clients.

2.9 Gantt Chart

To complete this project, we have some roles and responsibility and if we want to complete this within due time, we should maintain the roles and responsibility.

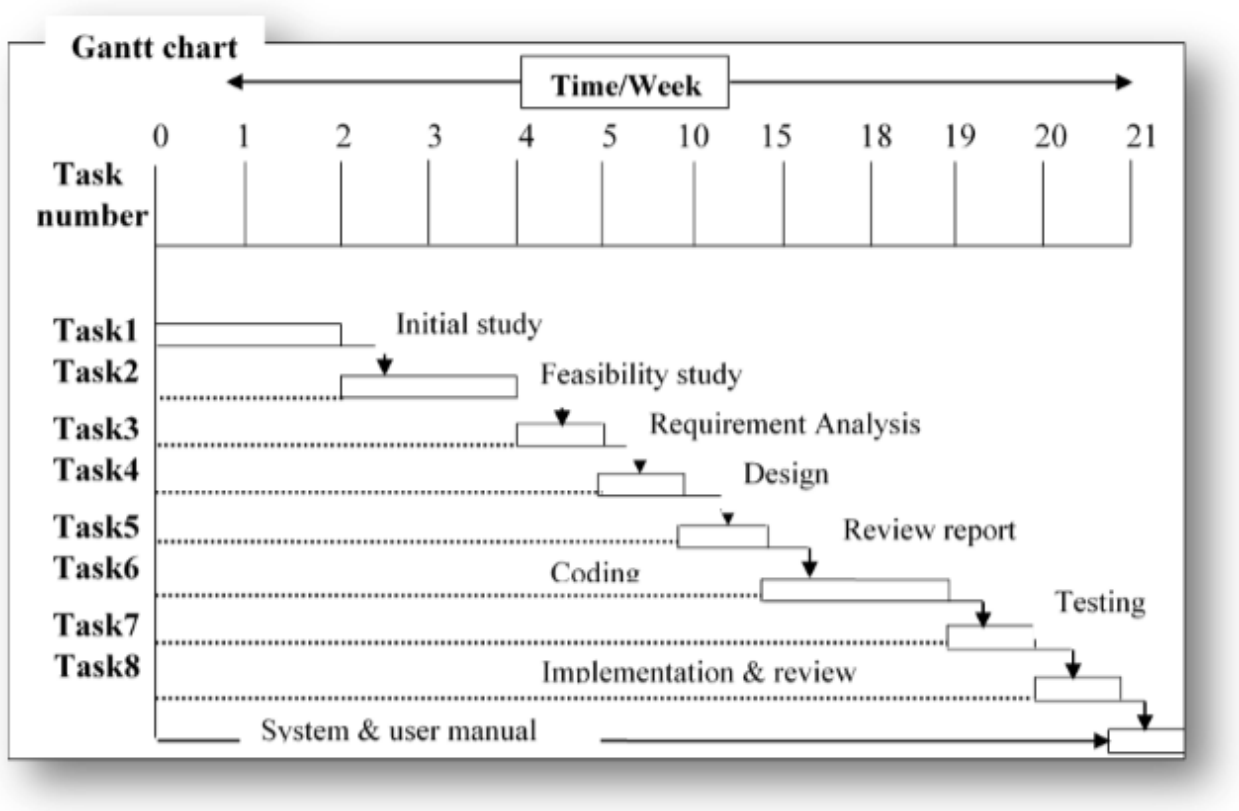

Figure 2.3: Gantt chart

2.10 Objectives/Goals of the System

- Provides the searching facilities based on various factors. Such as Chat Application, Smiles Chat, Users, Chat History
- The transactions are executed in off-line mode, hence on-line data for Chat Application, Online Chat capture and modification is not possible.
- It tracks all the information of Online Chat, Chat Profile, Users etc.
- Manage the information of Online Chat
- Shows the information and description of the Chat Application, Smiles Chat
- All the fields such as Chat Application, Smiles Chat, Chat History are validated and does not take invalid values
- It generates the report on Chat Application, Online Chat, Chat Profile
- Provide filter reports on Smiles chat, Users chat History
- You can easily export PDF for the Chat Application, Chat Profile, Users
- Application also provides excel export for Online Chat, Smiles Chat, Chat History
- You can also export the report into csv format for Chat Application, Online Chat, Chat History
- To increase efficiency of managing the Chat Application, Online Chat

- It deals with monitoring the information and transactions of Users.
- Manage the information of Chat Application
- Editing, adding and updating of Records is improved which results in proper resource management of Chat Application data.
- Manage the information of Users
- Integration of all records of Chat History.

2.11 Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Software testing can be stated as the process of validating and verifying that a software application:

1. meets the requirements that guided its design and development;

2. Works as expected, and

3. Can be implemented with the same characteristics.

The techniques for testing application are given below:

- Unit testing
- Black box testing
- Functional testing

Unit Testing

This is a procedure of testing individual code modules before they incorporated with different modules. Unit testing being tried might be a capacity, subroutine, technique or strategy. Units might be generally little gatherings of interrelated modules that are constantly executed as a gathering. The objective of unit testing is to recognize and fixed however many mistakes as would be prudent before modules are consolidated into a bigger programming unit. Mistakes become

substantially more troublesome and costly to find and fixed when numerous modules are joined. Unit testing of **Affix** are given below:

Unit Testing 1: Login as user

Testing Object: To ensure user can login into system correctly with valid user email and password.

ID	Test case	Attributes and values	Expected Result	Result
1.	Login= “email id” Password=”123”	Non register user can login	The system will not allow non-Registered user to login.	Pass
2.	Login= “email id” Password=”123”	Wrong email and password	The system throws an error Prevent from login.	Pass
3.	Login= “email id” Password=”123”	Enter correct email and password	The system can redirect to the page.	Pass

Functional Testing

Feasibility Testing is characterized as a kind of testing which confirms that each capacity of the product application works in conformance with the prerequisite determination.

This testing fundamentally includes discovery testing and it isn't worried about the source code of the application. Functional testing of **Affix** is given below:

Functional testing: Login with different user role.

Testing Object: To ensure user with different role login according to restricted System features.

ID	Test case	Attributes and values	Expected Result	Result
1.	Login as “Admin”	Login with admin Information.	Successfully Login.	Pass
2.	Login as “User”	Login with User Information.	Successfully Login.	Pass
3	Login as “User”	Login with User Information.	Successfully Login.	Pass

Black Box Testing

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

Black box testing of **Affix** is given below:

ID	Test case	Expected Result	Real Outcome	Result
1.	Login=“email id” Password=”123”	Login	Not login.	Fail
2.	Login=“email id” Password=”234”	Login	Login	Pass
3.	Login=“email id” Password=”456”	Login	Login	Pass

.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Online Chat Application 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 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 a strategy to be followed for effective system study and analysis can be taken.

Existing System of Online Chat Application:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials

3.1 Proposed System of Online Chat Application:

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data.
- Ensure data accuracies.
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

3.2 System overview of Affix

System overview of Affix refers to the whole process where data will be sent to the web server and received to database device. We have used the XML as the communication bridge between web services and web server. The overall process is illustrated in the figure below:

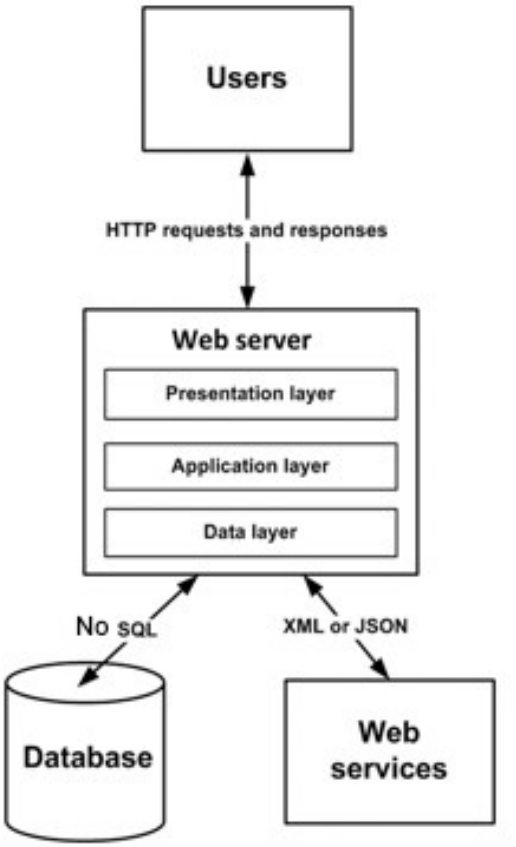

Figure 3.1: System Overview of Affix

According to the figure 3.1 in time of data sending first web server which will be generate from HTTP post request towards JavaScript. Then PHP will make a request to the web server and JavaScript receive the result from the server. The result will be converted into an XML and the XML will be given to the web Services and save into local MONGODB database. Thus, the data flow process will be occurred due to any valid request.

3.2.1 Data overview

We tried to represent all of the data in an efficient way. In **Affix**, list views are used very frequently. So, it is very important to represent list view items in a magnificent way. In web based, list views are controlled by the web server.

3.3 Security

Affix will create a secure communication between the web servers. HTTP session management is the core of web security. When web server received request from the website, then the server provide the data in XML format. All mitigations should be adopted to make sure sessions are secured. Developers should also enable/use applicable security measures.

3.4 Version support

As a web-based platform, **Affix** must be supported on most of the browsers. It must be supported by all possible and most usable browsers like Google Chrome, Mozilla Firefox and etc.

3.5 System Design

System design is the process of creating alternative solution to satisfy the study goals, evaluation the choices and the drawing up the specifications for the chosen alternative.

Objective: Transformation of the system specification into and operational system design.

Input: Project goals, user requirement priorities, specification.

Process: Transformation often with several alternatives whose costs and benefits must be qualified and compared.

Output: Logical system design, production schedule, software and hardware requirements.

3.6 A general Model of software design process

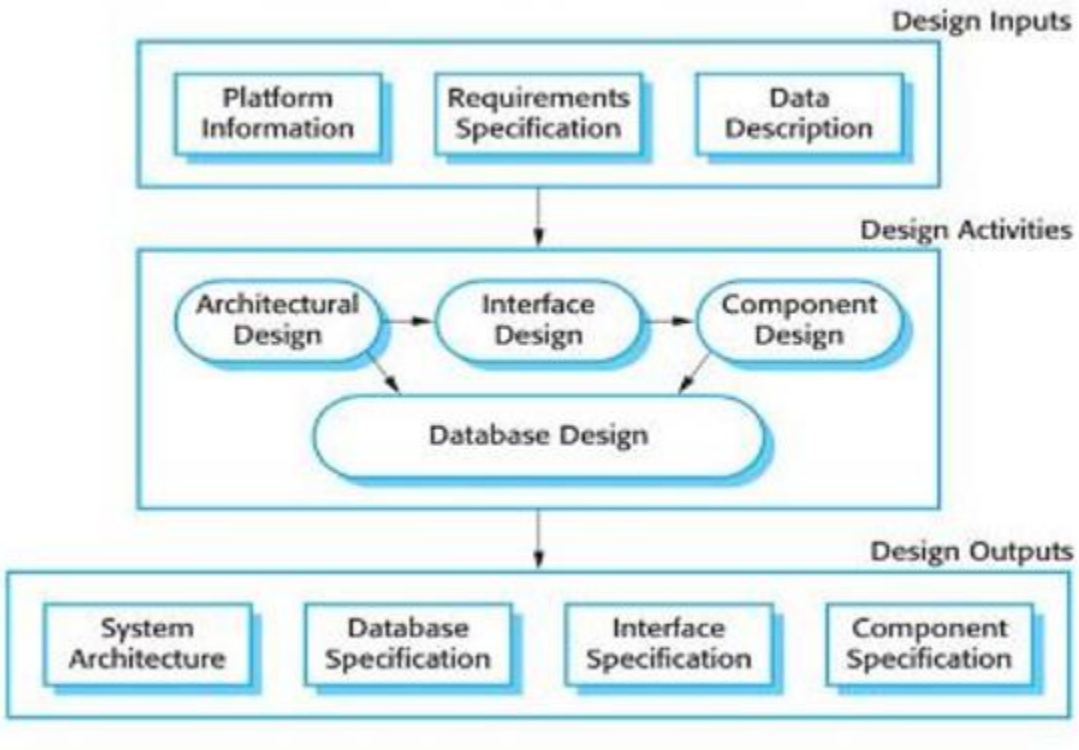

Figure 3.2: A general Model of software design process

3.7 Flow Chart

A flow chart is a graphical or symbolic representation of process. The flow chart symbols are linked together with arrows showing the process flow direction.

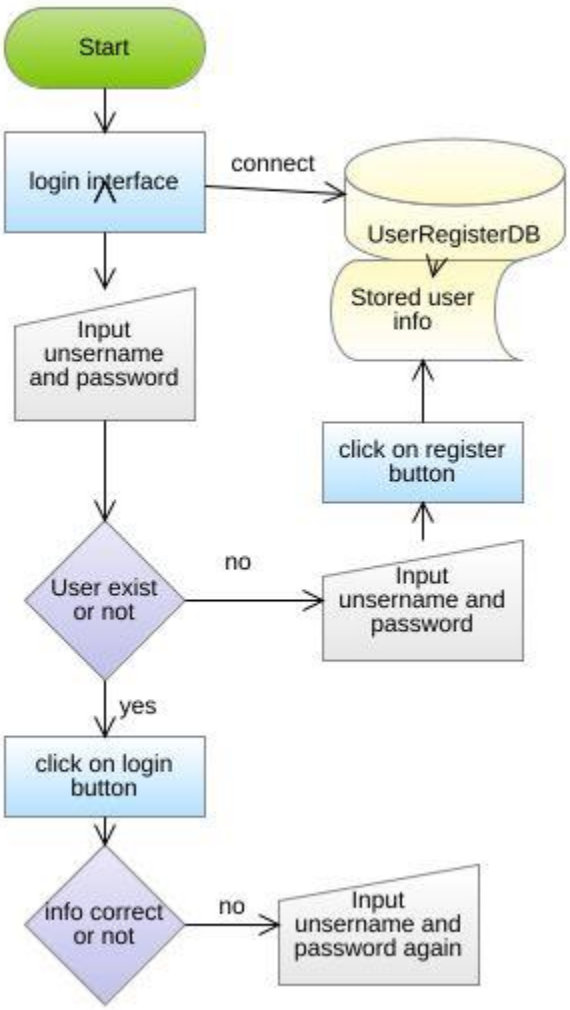

Figure 3.3 : A flowchart for Chat Application

3.8 Use case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

3.8.1 Use Case Diagram for Authentication System

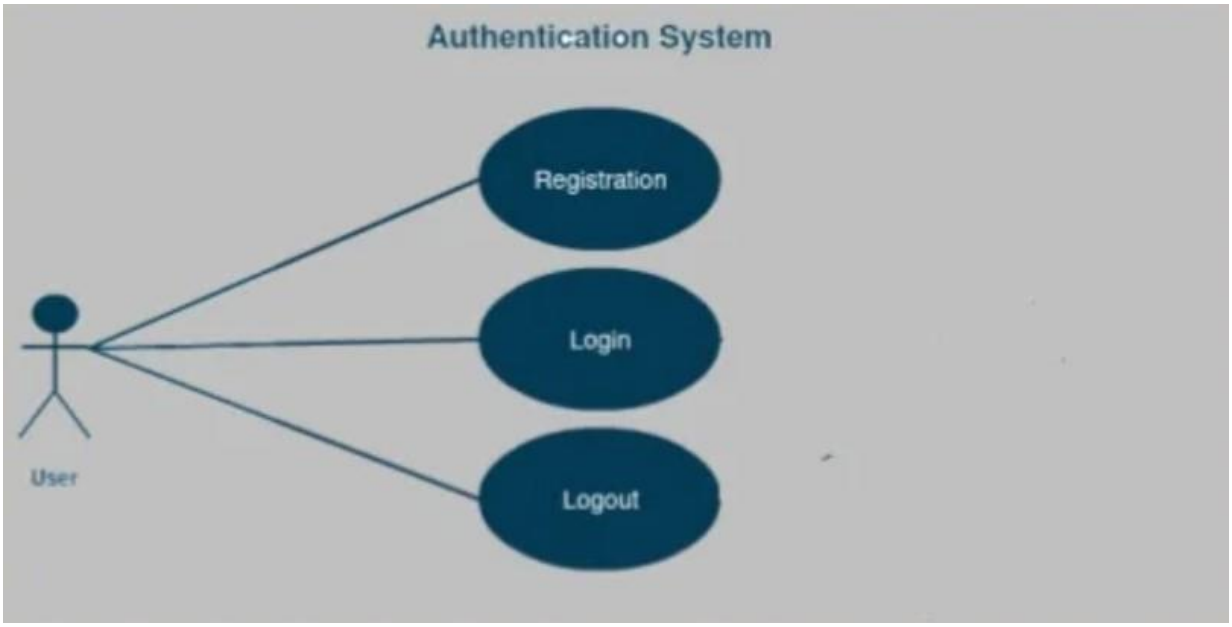

Figure 3.4: Use Case Diagram for Authentication system

3.8.2 Use Case Diagram of Contact form

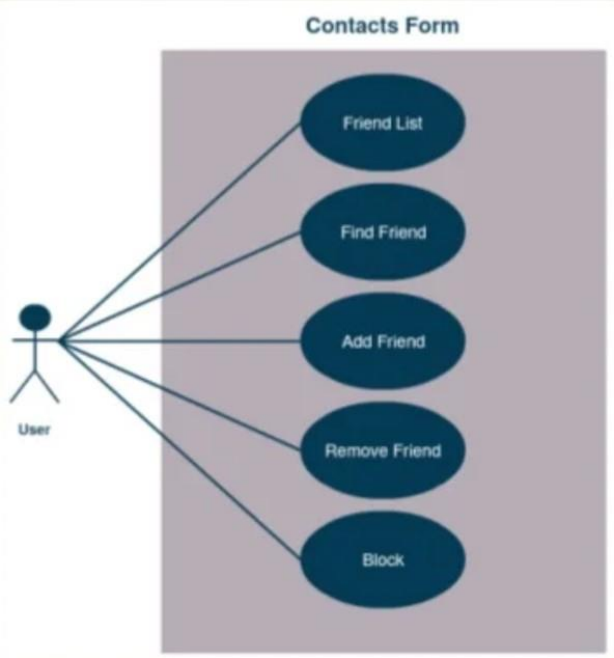

Figure 3.5: Use Case Diagram for Contact From

3.8.3 Use Case Diagram for Monitor

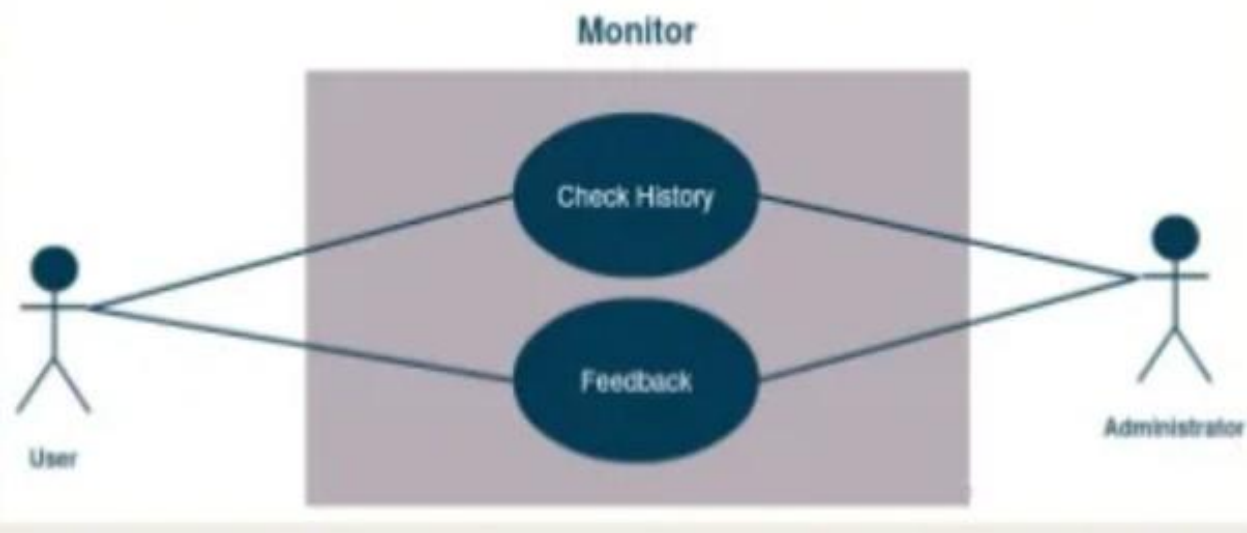

Figure 3.6: Use Case Diagram for Monitor

3.8.4 Use Case Diagram for Maintenance

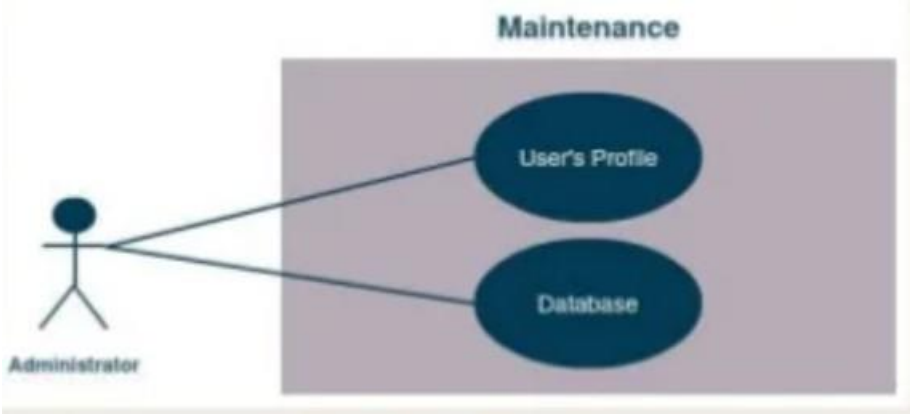

Figure 3.7: Use Case Diagram for Maintenance

3.9 Entity Relationship Diagram of Affix

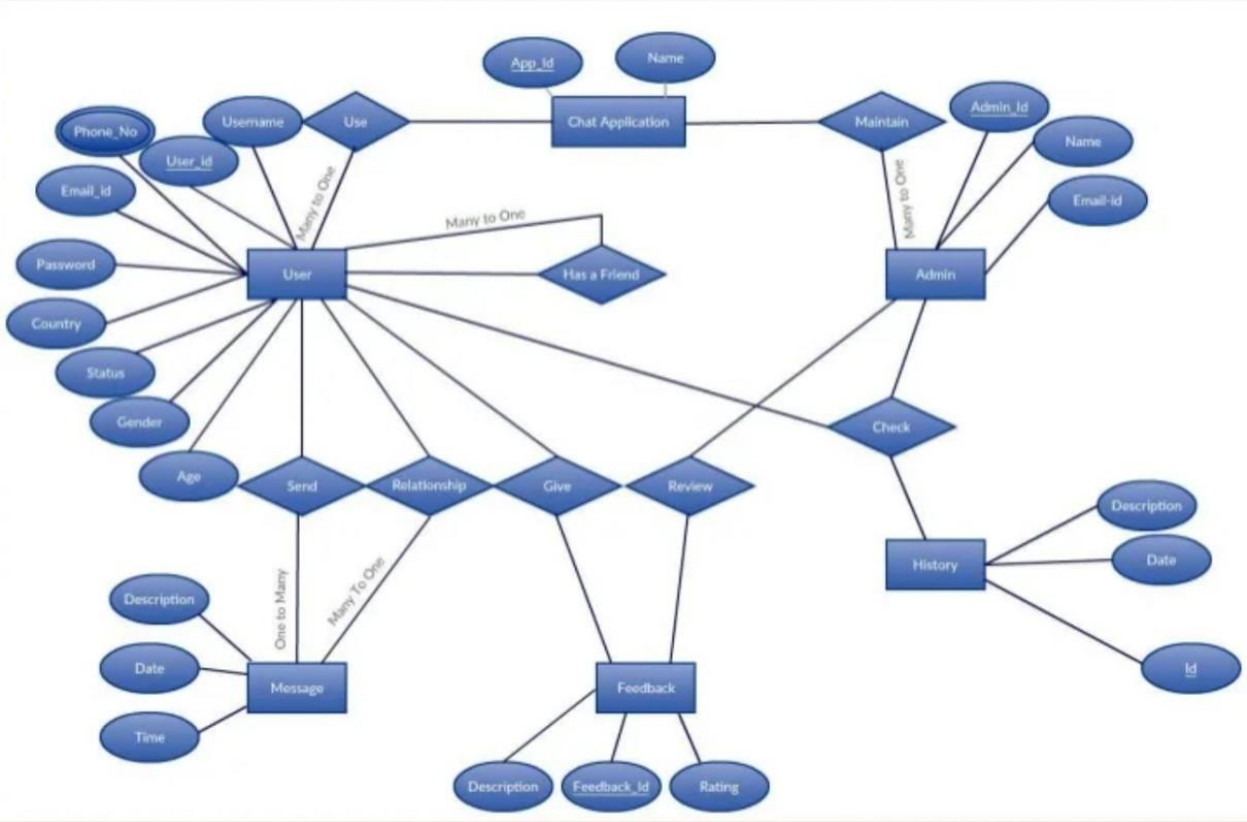

Figure 3.8: ER Diagram for Affix

3.10 Database Table of Affix

In order to fully utilize NOSQL server innovation, it is fundamental to ensure that the database is very much structured. The documents names picked to mark every one of the tables made inside the database endeavor to mirror the table's motivation and, in this way, add to well plan framework. The in-time step in structuring was to choose, as indicated by the necessities and particulars of the venture, which tables ought to be made, and what kind of data everyone should hold.

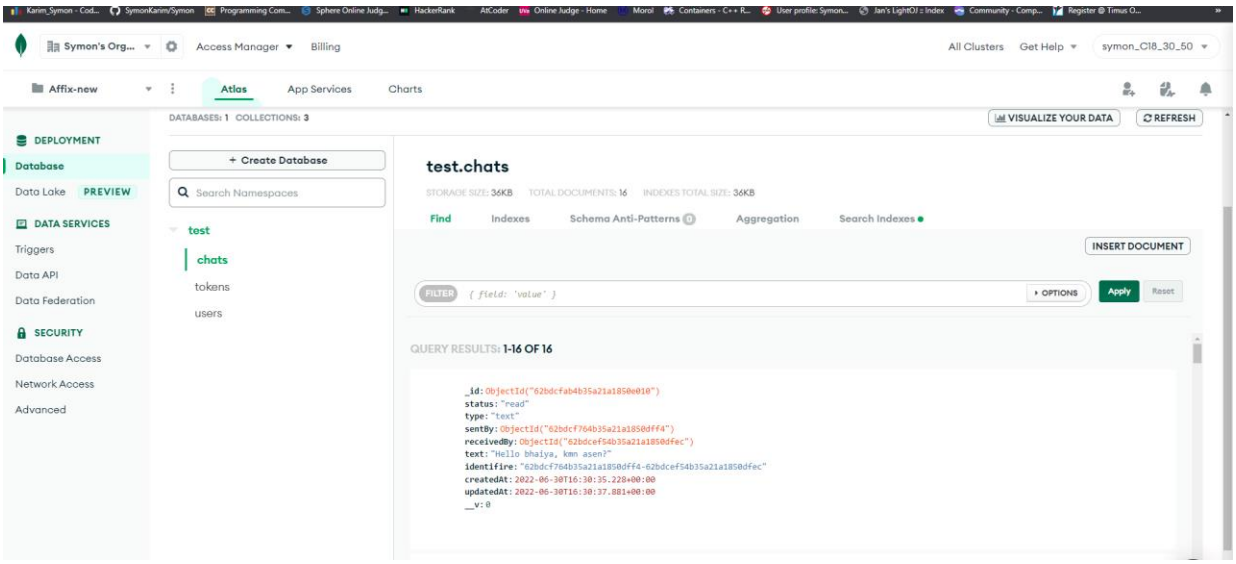

Figure 3.9: Database table of Chats

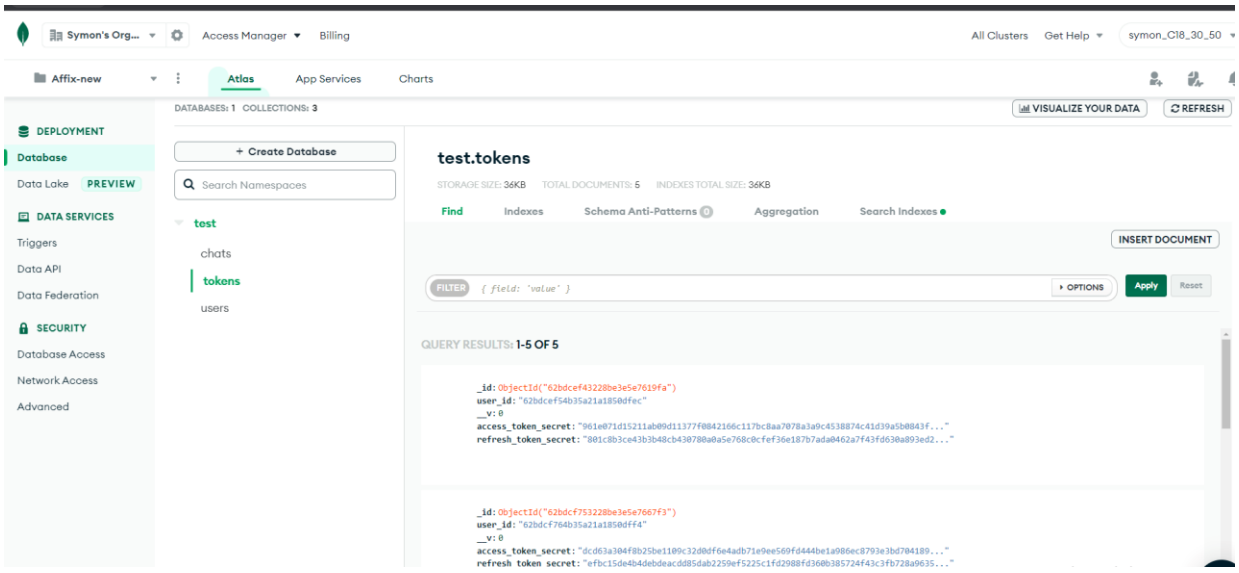

Figure 3.10: Database table of tokens

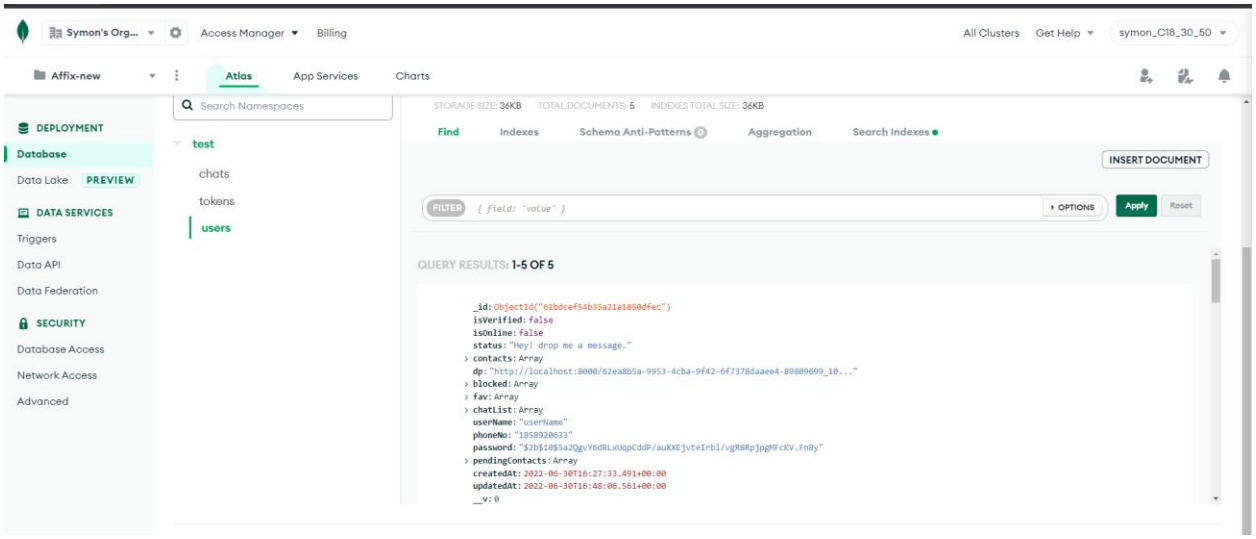

Figure 3.11: Database table of Users

3.11 Data flow Diagram

A data flow diagram (or DFD) is a graphical representation of the flow of data through an information system. It shows how information is input to and output. From the system, the sources and destinations of that information, and where that Information is stored.

3.11.1 Level 0 DFD

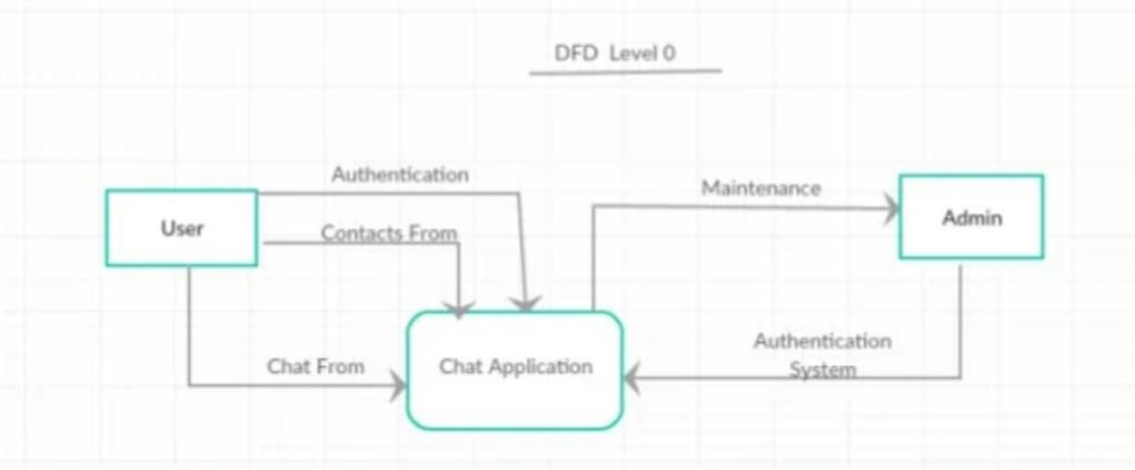

Figure 3.12: Level 0 DFD for Affix

3.11.2 Level 1 DFD for Affix

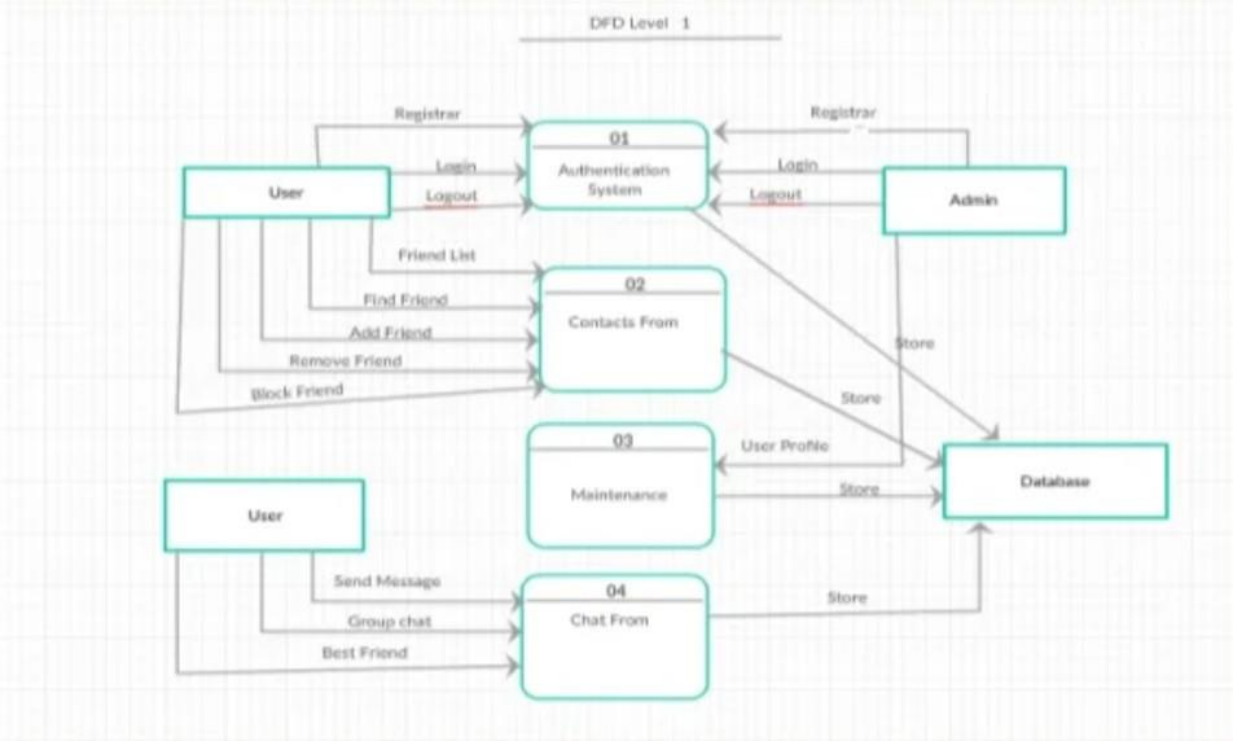

Figure 3.13: Level 1 DFD for Affix

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Overview

In this chapter, we have provided the experimental result analysis of our developed system for web. We have provided snapshots of every step that an admin, Seller and a customer can go through.

4.2 Design Model

A design model in Software Engineering is an item-based picture or pictures that speak to the utilization cases for a framework. Or on the other hand to put it another way, it is the way to depict a framework's execution and source code in a diagrammatic manner.

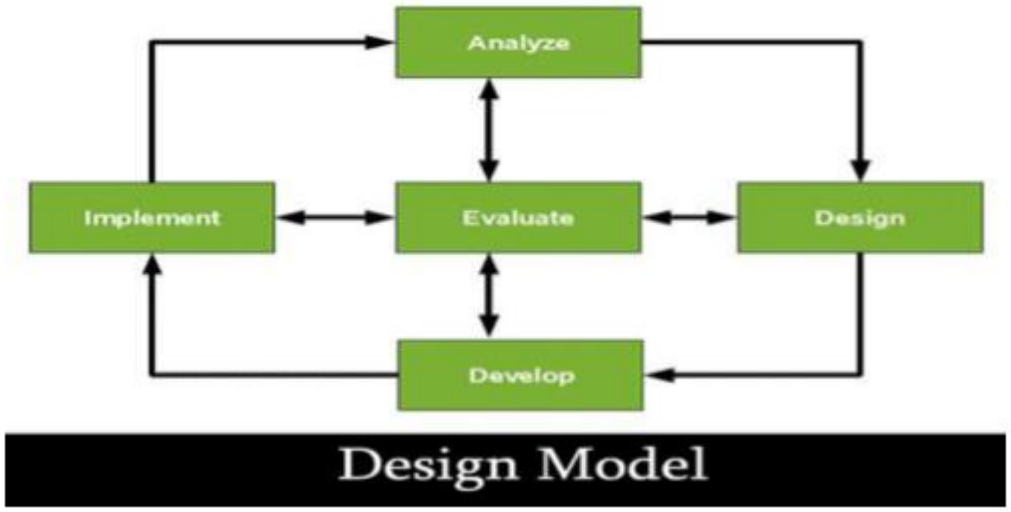

Figure 4.1 Design Model

4.3 Graphical Representation

The obtained results for both webs showed in the next sections.

4.3.1 Web Application

Affix starts with one more splash screen that exists for few seconds and then proceeds to terminal. User can easily login and navigate to the various feature of the website.

Sign Up:

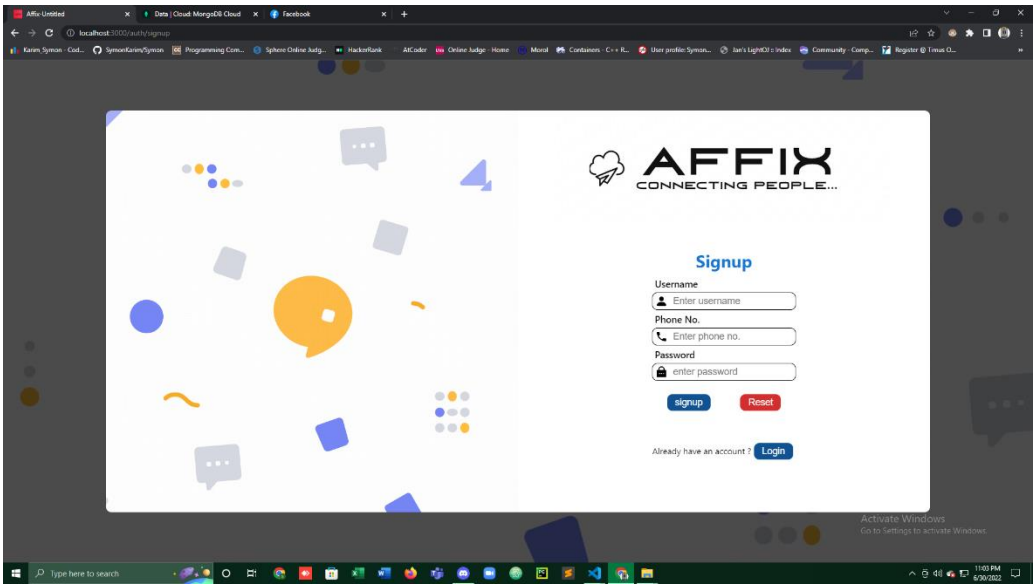

Figure 4.2: Sign Up of Affix

.

Login:

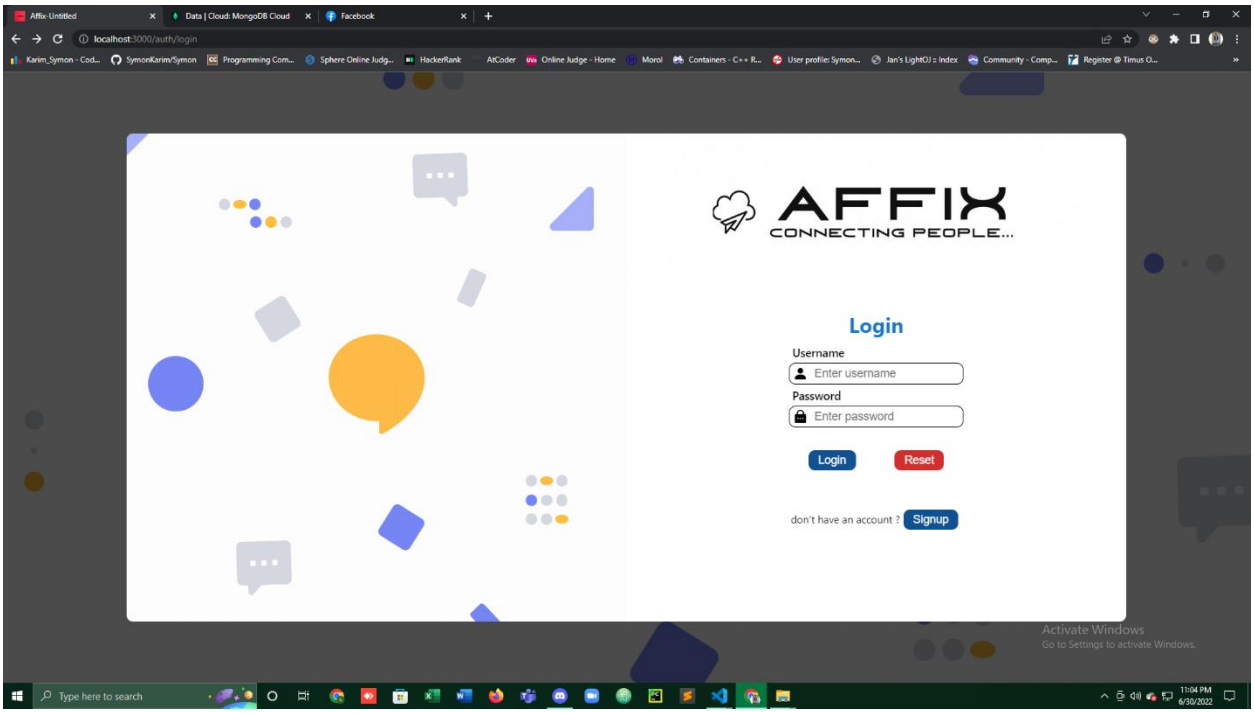

Figure 4.3: Login of Affix

Chat Body:

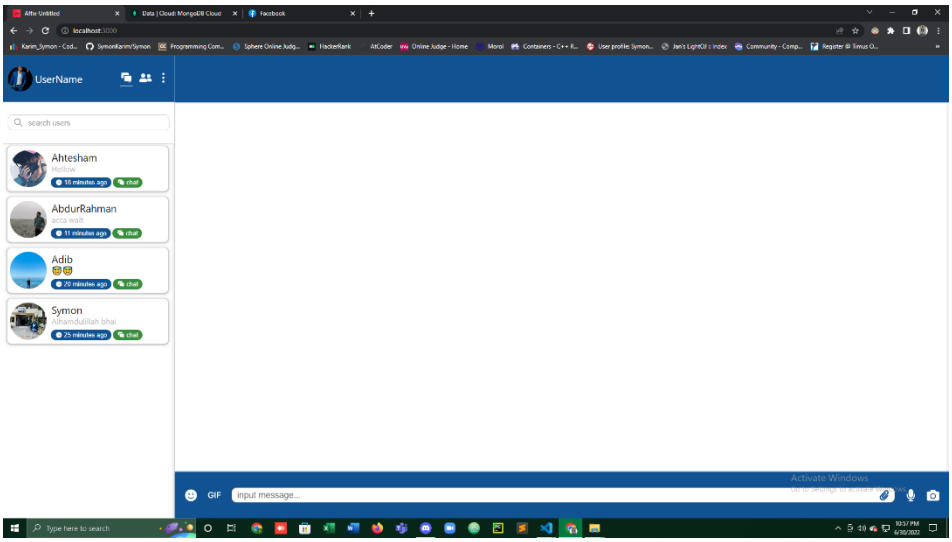

Figure 4.4: Chat Body of Affix

My Frienslist:

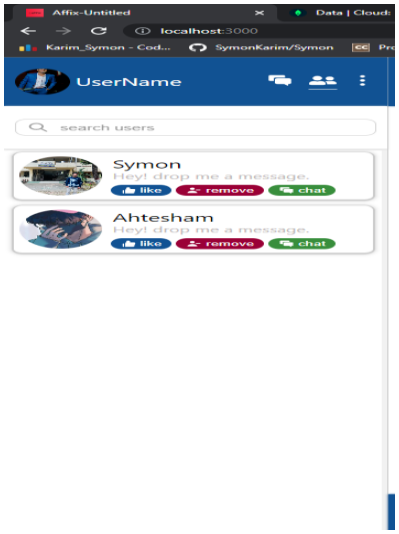

Figure 4.5: My Friend list

Profile Update:

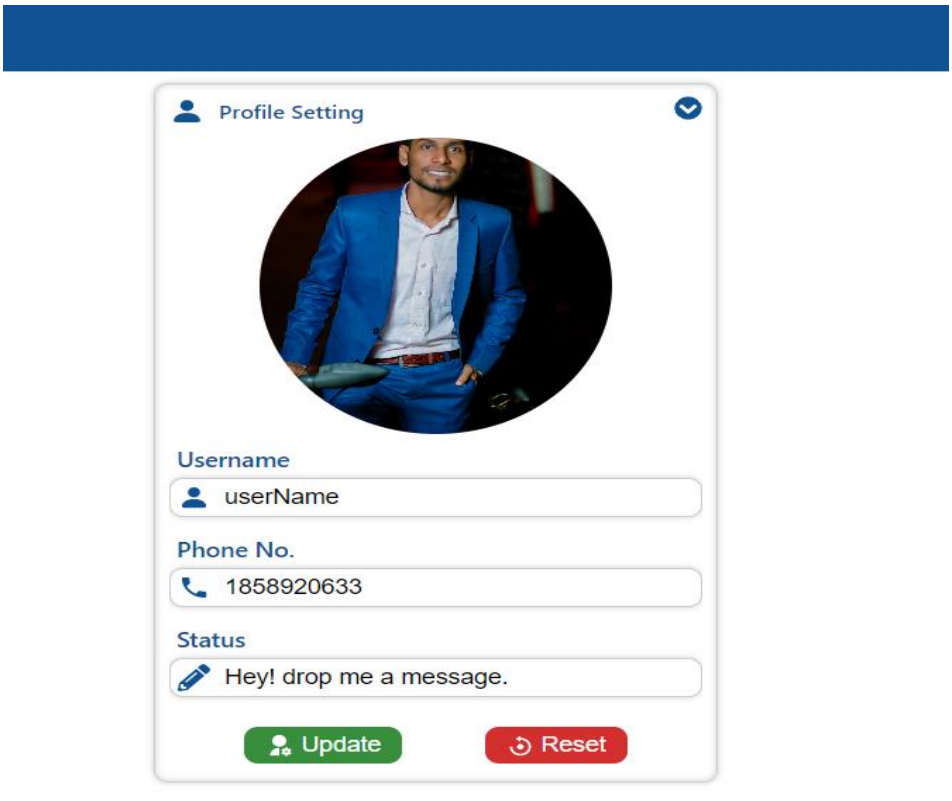

Figure 4.6: Profile Update

My Contact:

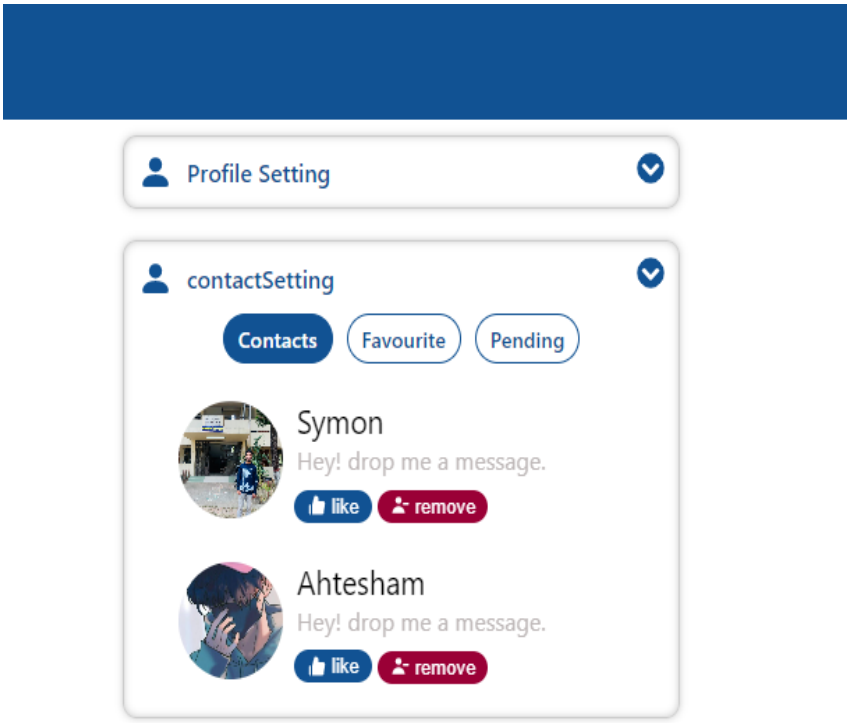

Figure 4.7: My Contact

Favourite Contact:

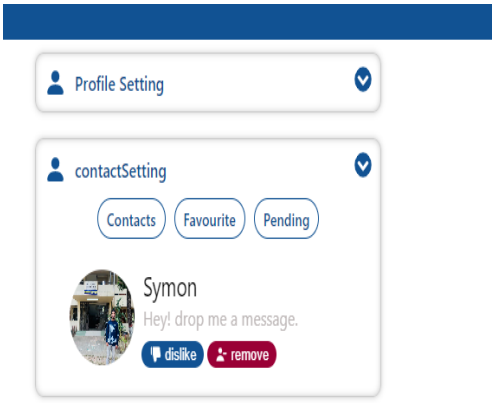

Figure 4.8: Favorite Contact

Pending Friend Request:

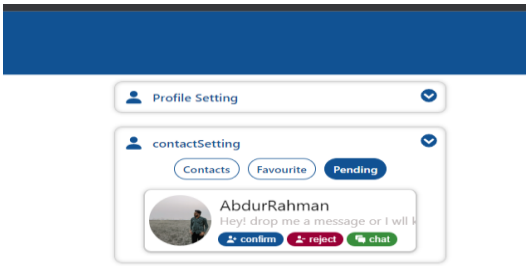

Figure 4.9: Pending Friend Request

Emoji:

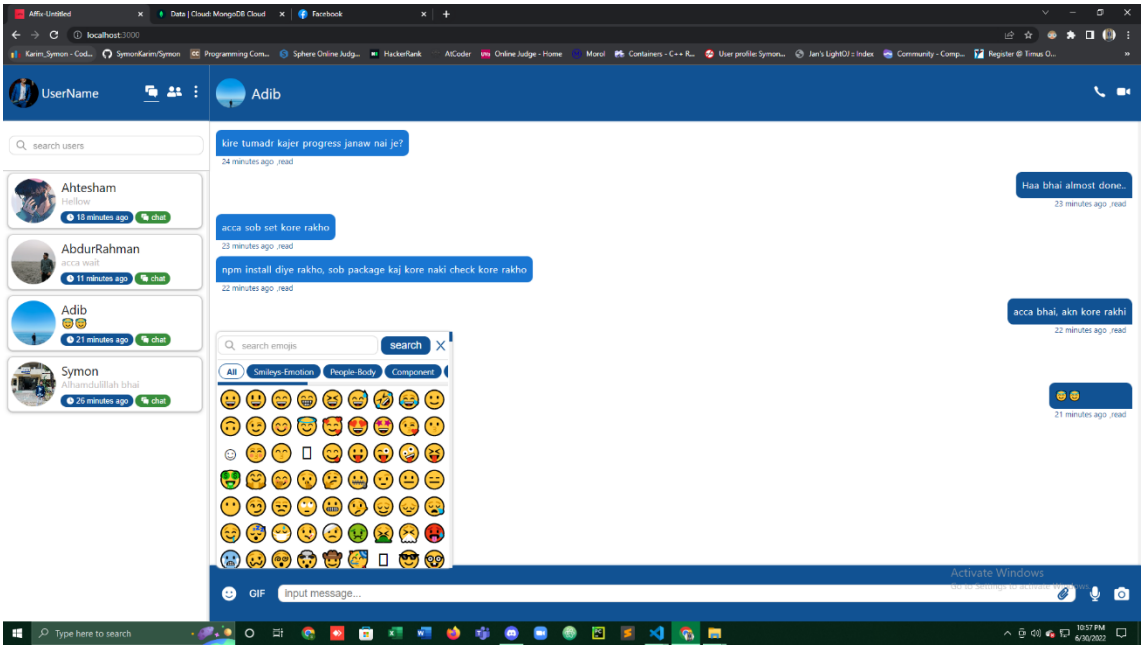

Figure 4.10: Emoji

Gif:

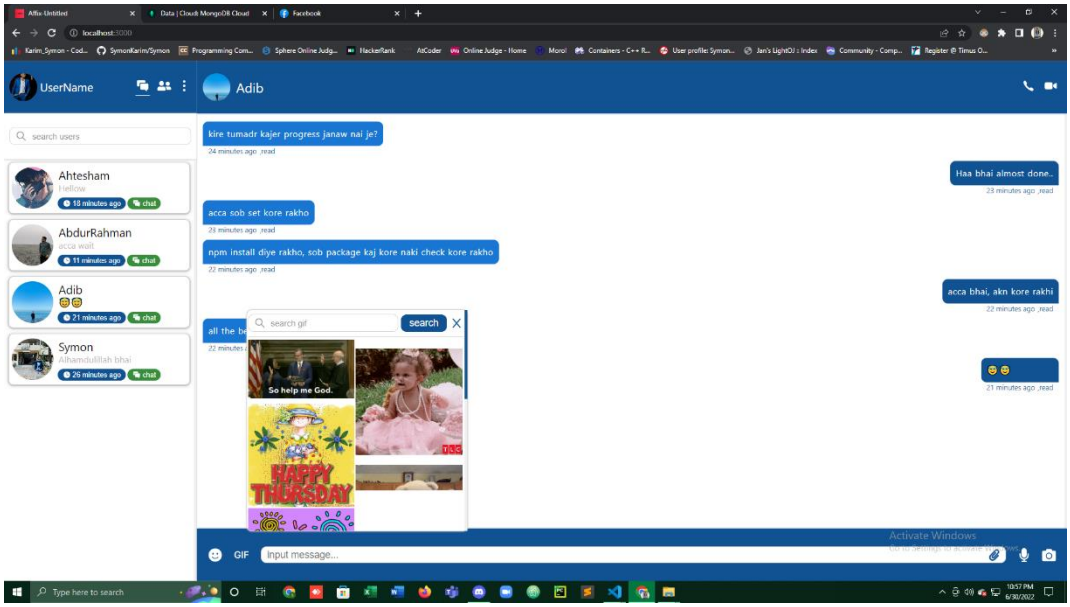

Figure 4.11: Gif

File Share :

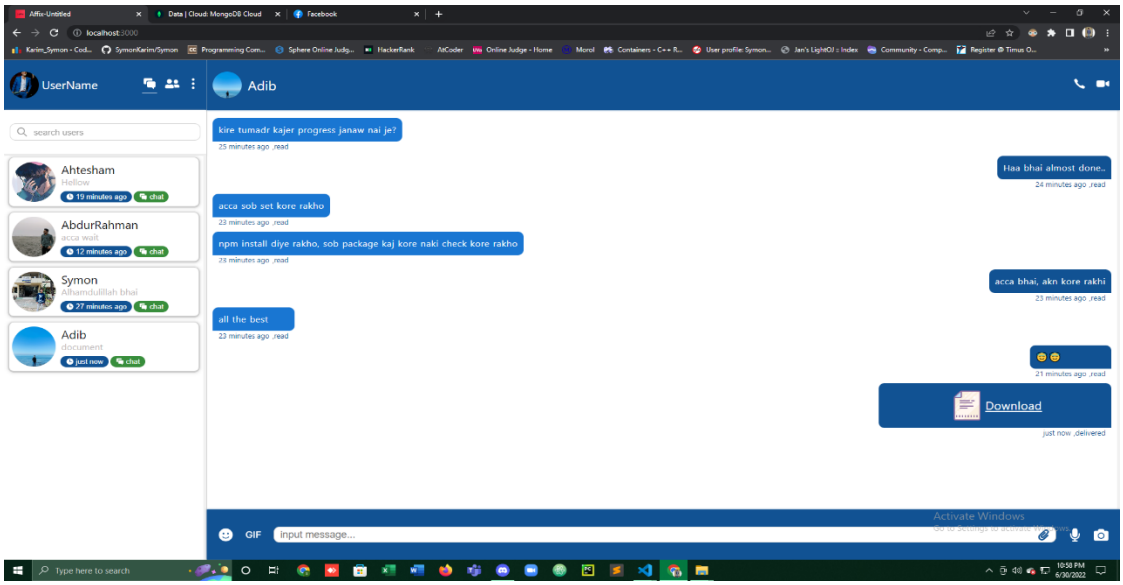

Figure 4.12: File Share

Audio Message:

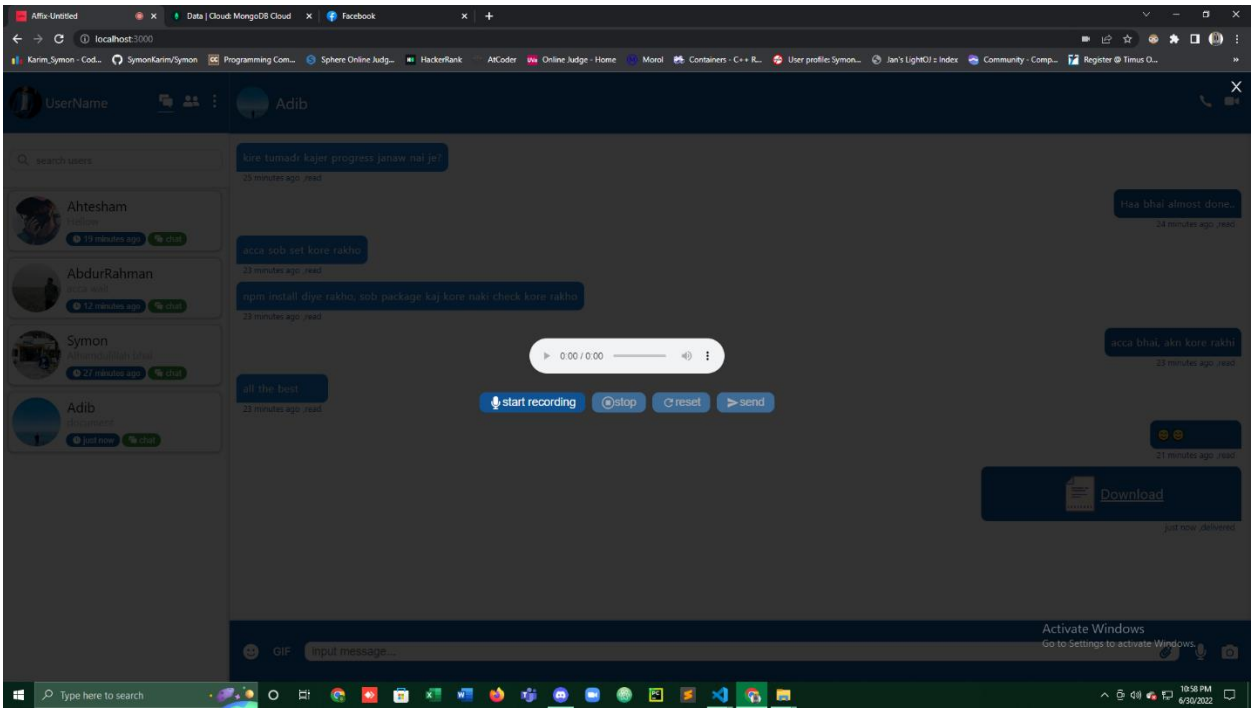

Figure 4.13: Audio Message

Video Message:

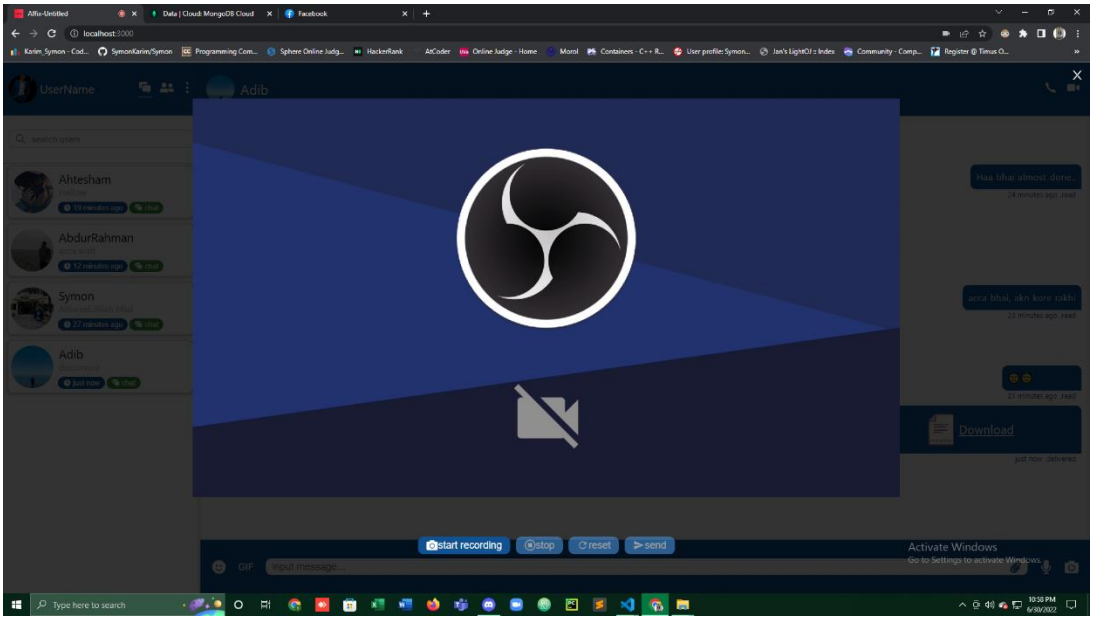

Figure 4.14: Video Message

Audio Call:

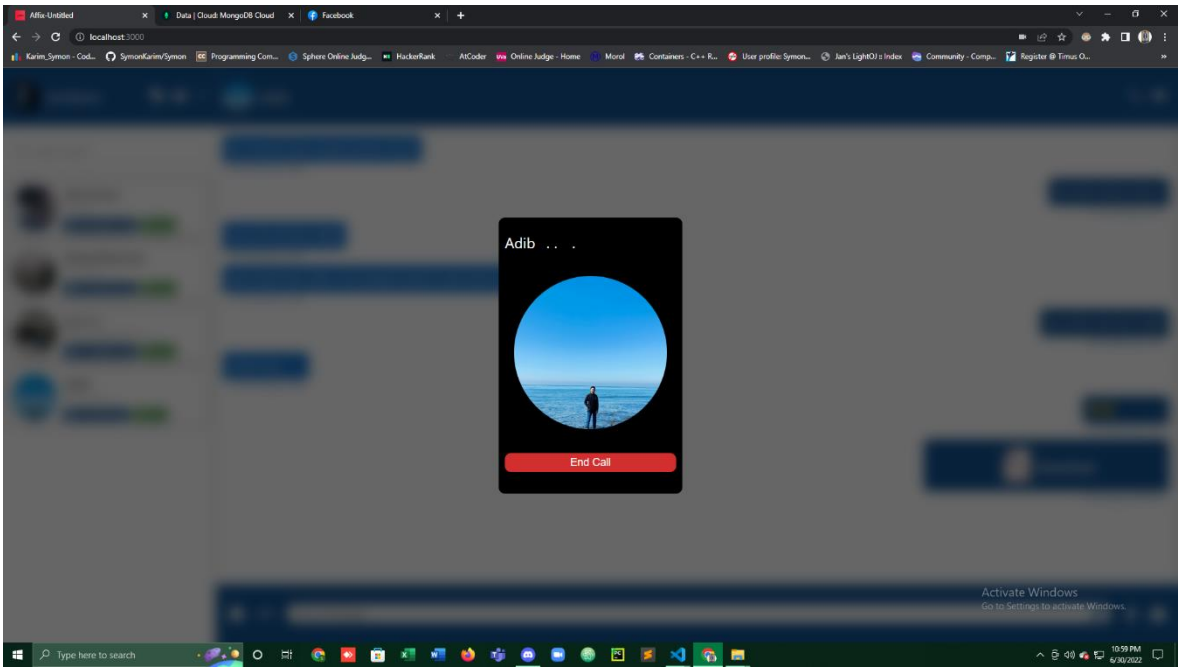

Figure 4.15: Audio Call

Video Call:

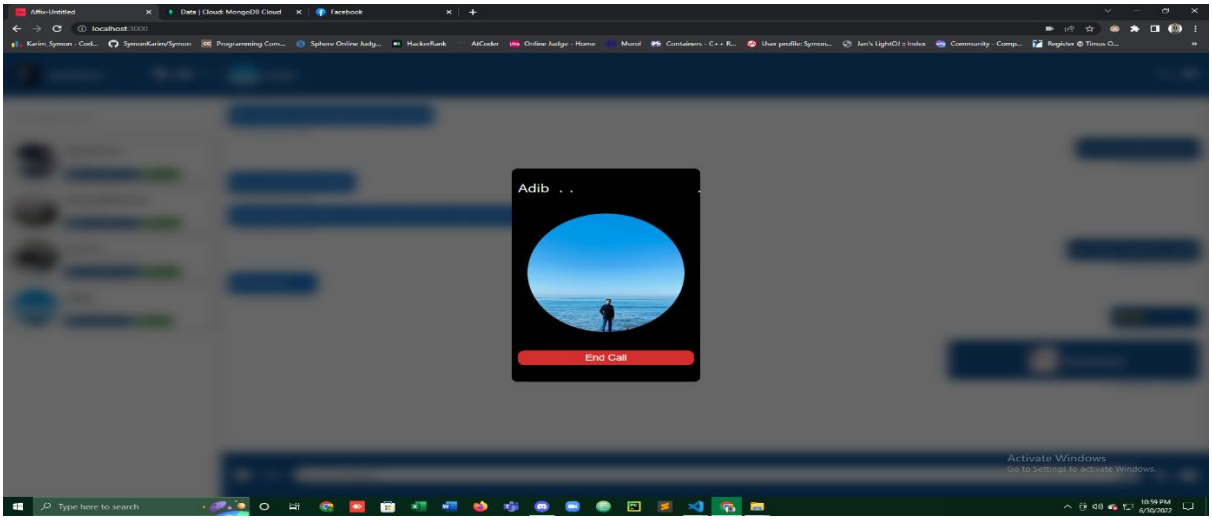

Figure 4.16: Video Call

This is How we worked and collaborated with each other:

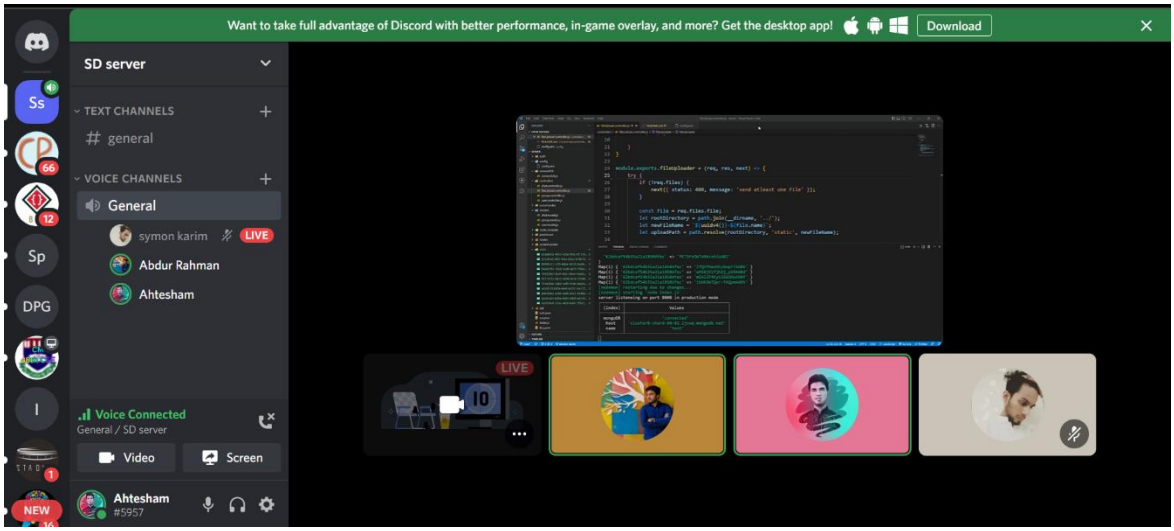

Figure 4.17: Collaboration

CHAPTER 5

CONCLUSION

5.1 Limitations

- User can't delete information.
- User can't create groups.
- Internet connection will need for use this app & update database.

5.2 Future work

- Addition of group chat/Audio/Video call feature.
- Addition of story feature.
- Addition of themes
- Access the applications on a mobile phone
- Addition of Chat-Bot
- Greater user experience
- Multilayer architecture with separation on concerns

5.3 Conclusion

My goal was to create an application where people will communicate with each other on a daily basis. The current application has fulfilled these goals. I followed the specifications strictly but enhanced some of the features when there was need for it to be done. With the goals achieved the basis of the application and this project has been achieved. Building this web application has been challenging and enriching because throughout the project I learnt a lot. There have been challenges especially when it came to the chat in Real time and while adding the file sharing feature. Careful planning made my job easier because I had to carefully think about the type of architecture, the design, the database types to use and what type of business objects to create. When this was done, I proceeded with implementation. Choosing the Node.JS framework for this project made the project to be realizable in that amount of time because the framework handled most of the heavy burden in underlying connections, security, portability and functionality. I could focus on the creation of the application and hence in this amount of time could develop a complete chat application. As I came to the end of the project, I realized that they are many enhancements that can be made on the application. Some of these ideas came from those who tested the application and some of them from me. I decided to follow the specification because there were realistic to achieve in this given amount of time. Any other enhancements to the application can be done in future development of the application.