Acknowledgement

I would like to express my sincere thanks and gratitude to my esteemed guide, Mr. Sanjay Mate for providing me with the possibility to complete
this project with the right guidance and advice
A special gratitude to my respect Project Coordinator Mrs. Kiran Shinde, for allowing me to use the facilities available and also helping me to
coordinate my project. Furthermore, I would also like to acknowledge with much appreciation the
crucial role of faculty members and their valuable guidance which as prompted my efforts in all stages of tis project work
Finally I want to thank my parents and friends who have helped me with
their valuable suggestions and guidance and have been very helpful.

INDEX

1. Introduction

1.1 Client Information

ATSS'S INSTITUTE OF INDUSTRIAL AND COMPUTER MANAGEMENT AND RESEARCH

NIGDI, PUNE 411044

Batch 2019-2022

IICMR (Institute of Industrial and Computer Management and Research) is an AICTE, New Delhi, approved technical education institute. This Leading Institute of Computer Studies in Pune is affiliated to Savitribai Phule Pune University, recognized by the Government of Maharashtra, and Accredited by NAAC. It offers technical education, computer education, management courses, and Software Engineering courses in Pune for the students. The intake for this course is 120 students.

1.2 Abstract

The main objective of this Abhyas-Friendly is to provide students a space where they can chat with each other along with teachers in respective subjects.

Teachers can share knowledge and guidance with students in their respective channels along with any news or updates into the same application which makes things easier to manage.

Teachers and students can give points to other students as a token of thanks which will not only encourage students but also will help teachers in identifying the helpful talent.

1.3 Existing System and Need for System

In this drastic world of the Internet, there are many such students who are not socially active but do need company or some guidance while studying. Applications like chatting makes it difficult for users to track important updates and resources. The communication gaps create chaos to management. ABHYAS-FRIENDLY is an application which will allow students from all around the globe, private organisations to connect together as per the topics and also they can share resources which all can be done with live interactions from anywhere.

It has a functionality where Instructors too can connect and pass their knowledge to the students as per the needs.

Instructors and students can give points to other students as a token of thanks which will not only encourage students but also will help teachers in identifying the helpful talent.

1.4 Scope of System

Users management

• Maintain details of Teachers, Students

• **Add** - New users details can be added by admin

• **Update** - Existing users details can be updated by admin

• **View** - Admin can view existing details

• **Delete**- Existing details can be deleted

Channel Management

• Maintain details of Channels

• **Add** - New channel details can be added by admin

• **Update** - Existing channel details can be updated by admin

• **View** - Admin can view existing details

• **Delete**- Existing details can be deleted

Points Management

- Students receive points.
- Students can give points.
- Monthly one student with highest point highlights.
- Monthly points get reset.

Notification Management

- Add Notification
- Remove Notification
- Teachers can send notification channel heading
- Students can fetch notifications.

1.5 Operating Environment-Hardware and Software

Hardware configuration:

Processor Intel Pentium

RAM 4 GB

Hard disk 500 GB

Software configuration:

Operating system: Windows

Client Side Scripting:

- C#
- Unity

Server Side Scripting: Firebase & C#

Database Tool: No SQL

Testing Server: Firebase

1.6 Brief description of Technology used

C#

C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework.

C# is used to develop web apps, desktop apps, mobile apps, games and much more.

Unity

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development and is considered easy to use for beginner developers and is popular for indie game development.

Firebase

Firebase is a set of hosting services for any type of application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++)
It offers NoSQL and real-time hosting of databases, content, social authentication (Google, Facebook, Twitter and Github), and notifications, or services, such as a real-time communication server.

NoSQL

NoSQL, database technology stores information in JSON documents instead of columns and rows used by relational databases. To be clear, NoSQL stands for “not only SQL” rather than “no SQL” at all. This means a NoSQL JSON database can store and retrieve data using literally “no SQL.” Or you can combine the flexibility of JSON with the power of SQL for the best of both worlds. Consequently, NoSQL databases are built to be flexible, scalable, and capable of rapidly responding to the data management demands of modern businesses.

Operating System

Microsoft Windows, also called Windows and Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM-compatible PCs. Windows 10 is a major release of the Windows NT operating system developed by Microsoft. It is the successor to Windows 8.1, which is available for download via MSDN and TechNet, as a free upgrade for retail copies of Windows 8 and Windows 8.1 users via the Windows Store, and to Windows 7 users via Windows Update. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users

2. Proposed System

2.1 Study of similar systems

Mostly all the social media systems like

- Clubhouse
- Discord
- Twitch

2.2 Objectives of Proposed System

The main objective of this Abhyas-Friendly is to provide students a space where they can chat with each other along with teachers in respective subjects.
Teachers can share knowledge and guidance with students in their respective channels along with any news or updates into the same application which makes things easier to manage.
Teachers and students can give points to other students as a token of thanks which will not only encourage students but also will help teachers in identifying the helpful talent.

2.3 Feasibility Study

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on upcoming requirements. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analysed carefully.There are 3 parts in the feasibility study.

A. Economic Feasibility

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need for online availability of facilities. All the hardware and software costs have to be borne by the organisation. Thus the benefits of this project in the current scenario make it economically feasible. The proposed system will surely overcome the initial costs and later on running costs for the system.

B. Technical Feasibility

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology.The application is the fact that it has been developed on windows 10 platform and a high configuration of 2GB RAM on Intel Pentium Dual core processor. This is technically feasible. We studied complete functionality to be provided as described in SRS and checked if everything is possible using different kinds of frontend and backend platforms.

C. Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified and how it satisfies the requirements identified. These include such parameters as reliability, maintainability, supportability, usability, disposability, sustainability, affordability and others. Besides, proper training has to be conducted to let users know the essence of the system so that they feel comfortable about the new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doings.

2.4 Users of System

Administrator

The Administrator is the super user and has complete control over all the activities that can be performed.
The administrator can add, delete and update user details.
The administrator can also block and unblock accounts.

Students(User)

Students are the users that have complete control over all the activities related to the student portal as an authenticated user.
Students can join live rooms based on choice of interest and can have voice and video conversations.
Students can access, post blogs, resources of the rooms they joined in.

Teachers(User)

Teachers are the users that have complete control over all the activities related to the teacher portal as an authenticated user.
Teachers can join live rooms based on choice of interest and can have voice and video conversations.
Teachers can access, post blogs, resources of the rooms they joined in.

3. Analysis and Design

3.1 System Requirements

Functional Requirements:

The System must provide following functionalities -

- **Students management (User):**

  - Create username and password for students
  - Fill in students subject of interest
  - Blogs for sharing resources as per rooms.

- **Teachers management (User):**

  - Create username and password for teachers
  - Fill in subject of interest
  - Blogs for sharing resources as per rooms.

- **Login management :**

  - Allowing user to login using username and password
  - Provide functionality to reset password by selecting ‘Forgot password’.

- **Points management :**

  -Allowing users to give points as a token of thanks to students
  -Storing points for each month to display student of the month.
  -Resetting the values for updating every month.

- **Message management :**

  -Allowing users to exchange messages
  -Messages are stored and can be deleted, updated.

- **Notification management :**

  -Allowing users(teachers)to send notifications.
  -Notifications are stored and can be deleted, updated.
  -Users can check notifications.

Non Functional Requirements:

Usability-

The system provides ease of use to the system, a simple but quality user interface using better component design is developed to make it understood easily and requires less training. The system
provides a help and support menu in all interfaces to help users

Security-

System ensures secure access of confidential data (customer's details).The system uses username and passwords to prevent unauthorised access.

Performance-

The system response time for every instruction conducted by the user must not exceed more than a minimum of 10 seconds. The system should have high performance rate when executing user's
input and should be able to provide response within a short time span usually 50 second for highly complicated task and 20 to 25 seconds for less complicated task.

Availability-

The system would always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system would be available in 1 to 2 working days, so that business process is not severely affected.

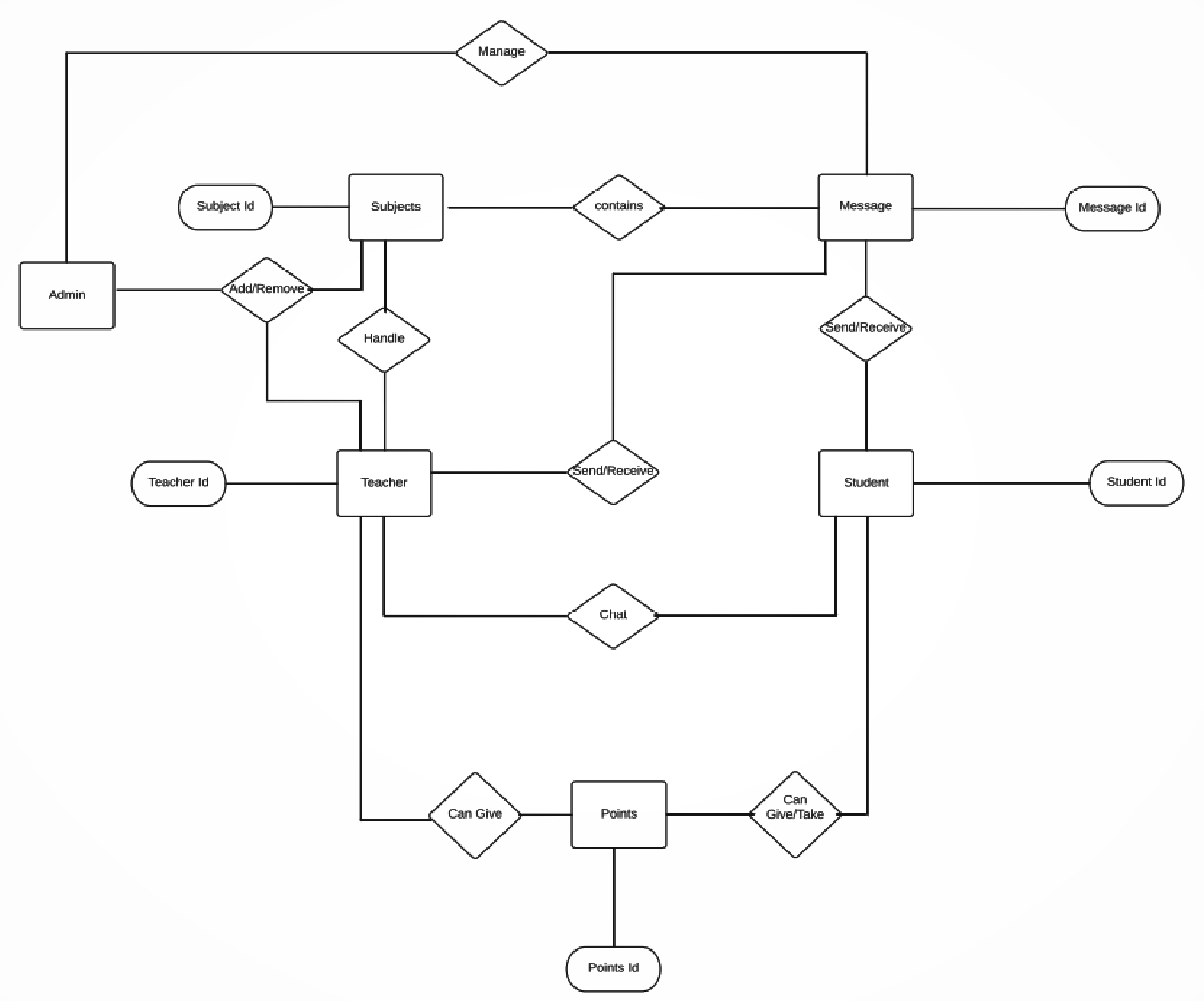
Error handling-

Error should be considerably minimised and an appropriate error message that guides the user to recover from an error should be provided. Validation of user's input is highly essential.

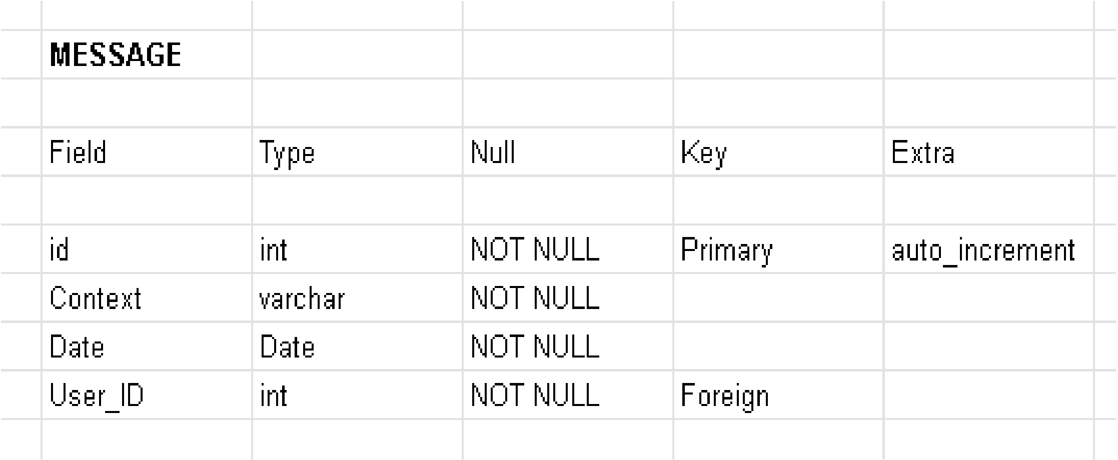
Ease of use-

Considering the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and require less training.

3.2 Entity Relationship Diagram (ERD)

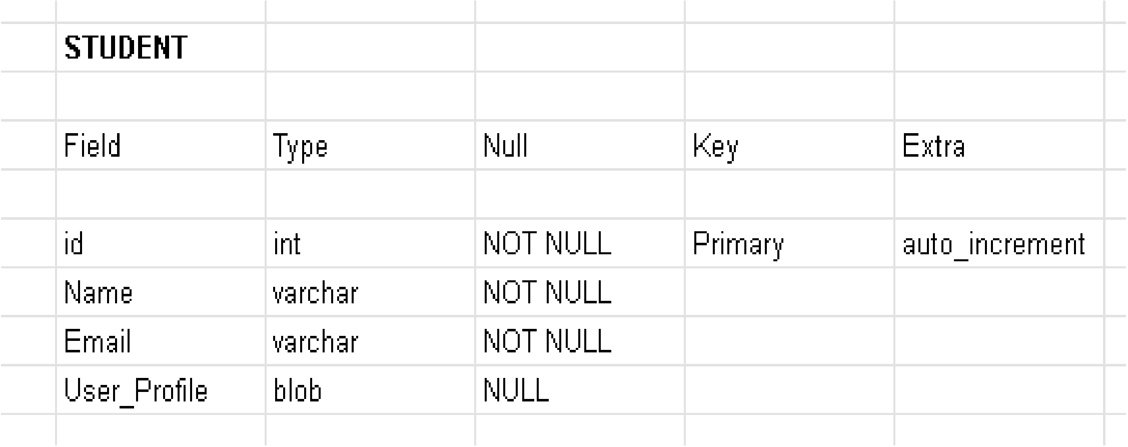


3.3 Table Structures



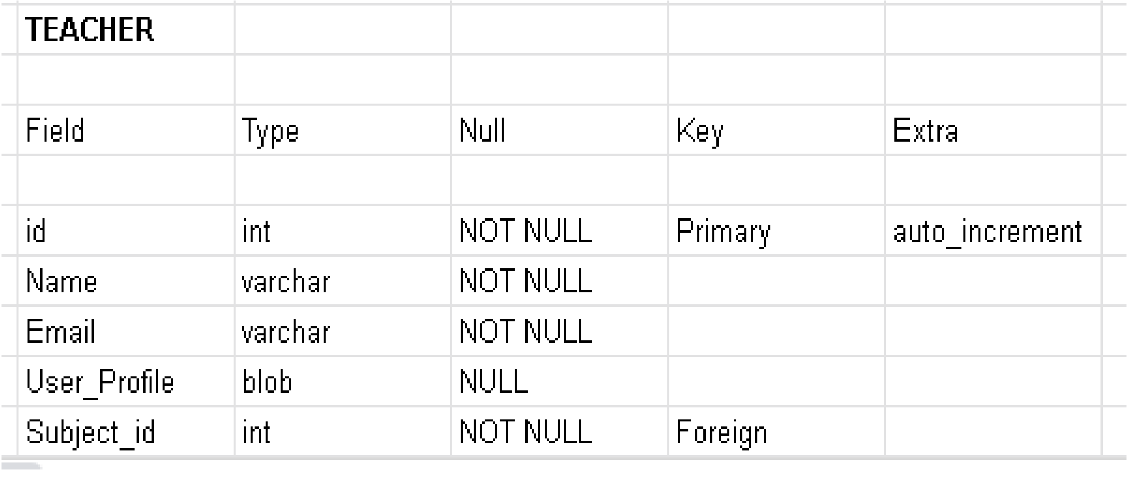
MESSAGE

Field	Type	Null	Key	Extra
id	int	NOT NULL	Primary	auto_increment
Context	varchar	NOT NULL		
Date	Date	NOT NULL		
User_ID	int	NOT NULL	Foreign	



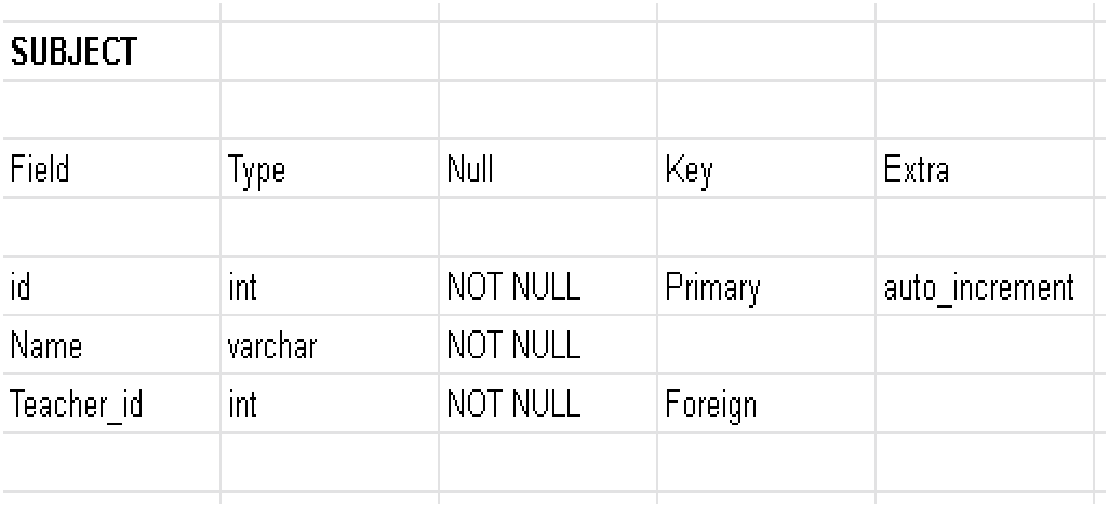
STUDENT

Field	Type	Null	Key	Extra
id	int	NOT NULL	Primary	auto_increment
Name	varchar	NOT NULL		
Email	varchar	NOT NULL		
User_Profile	blob	NULL		

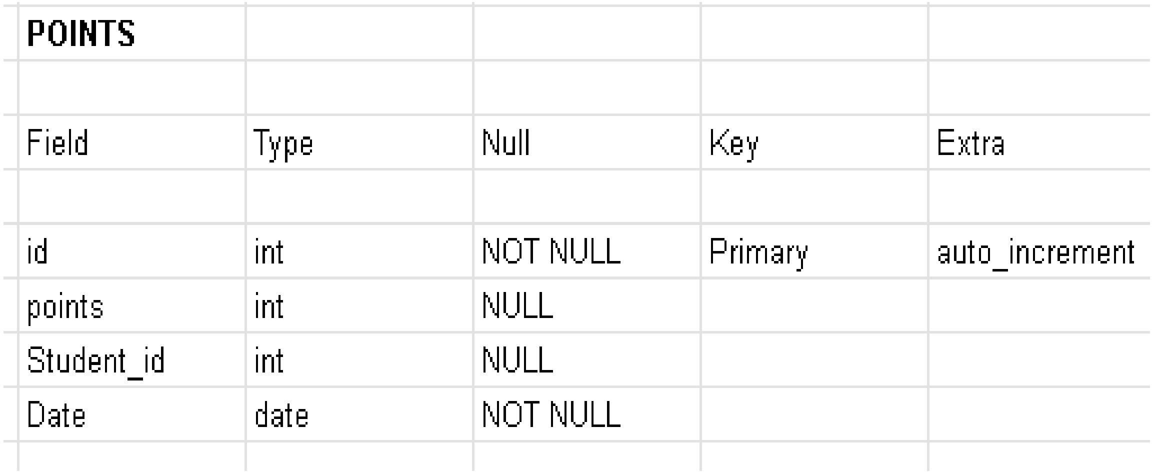


TEACHER

Field	Type	Null	Key	Extra
id	int	NOT NULL	Primary	auto_increment
Name	varchar	NOT NULL		
Email	varchar	NOT NULL		
User_Profile	blob	NULL		
Subject_id	int	NOT NULL	Foreign	

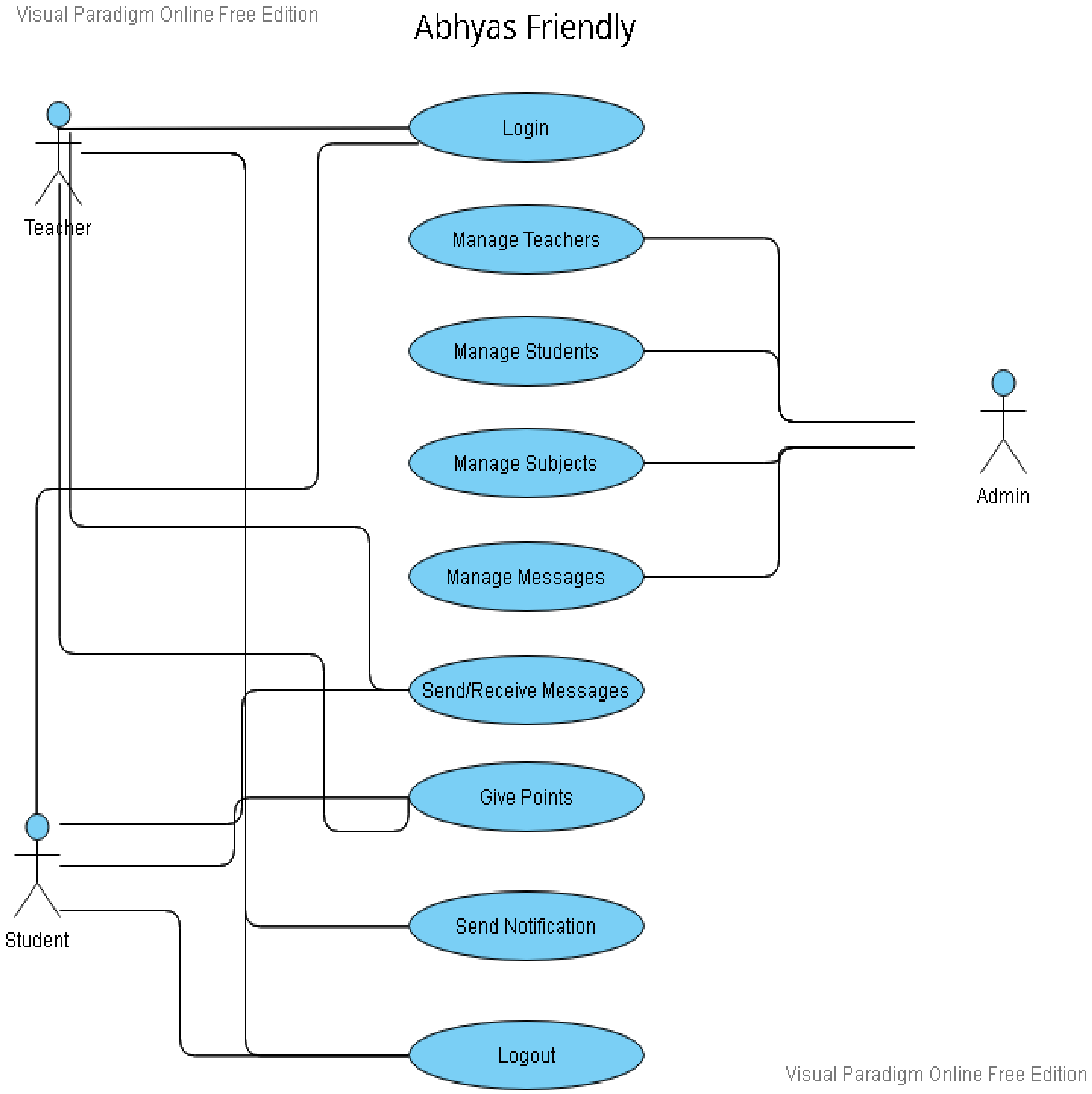


SUBJECT				
Field	Type	Null	Key	Extra
id	int	NOT NULL	Primary	auto_increment
Name	varchar	NOT NULL		
Teacher_id	int	NOT NULL	Foreign	

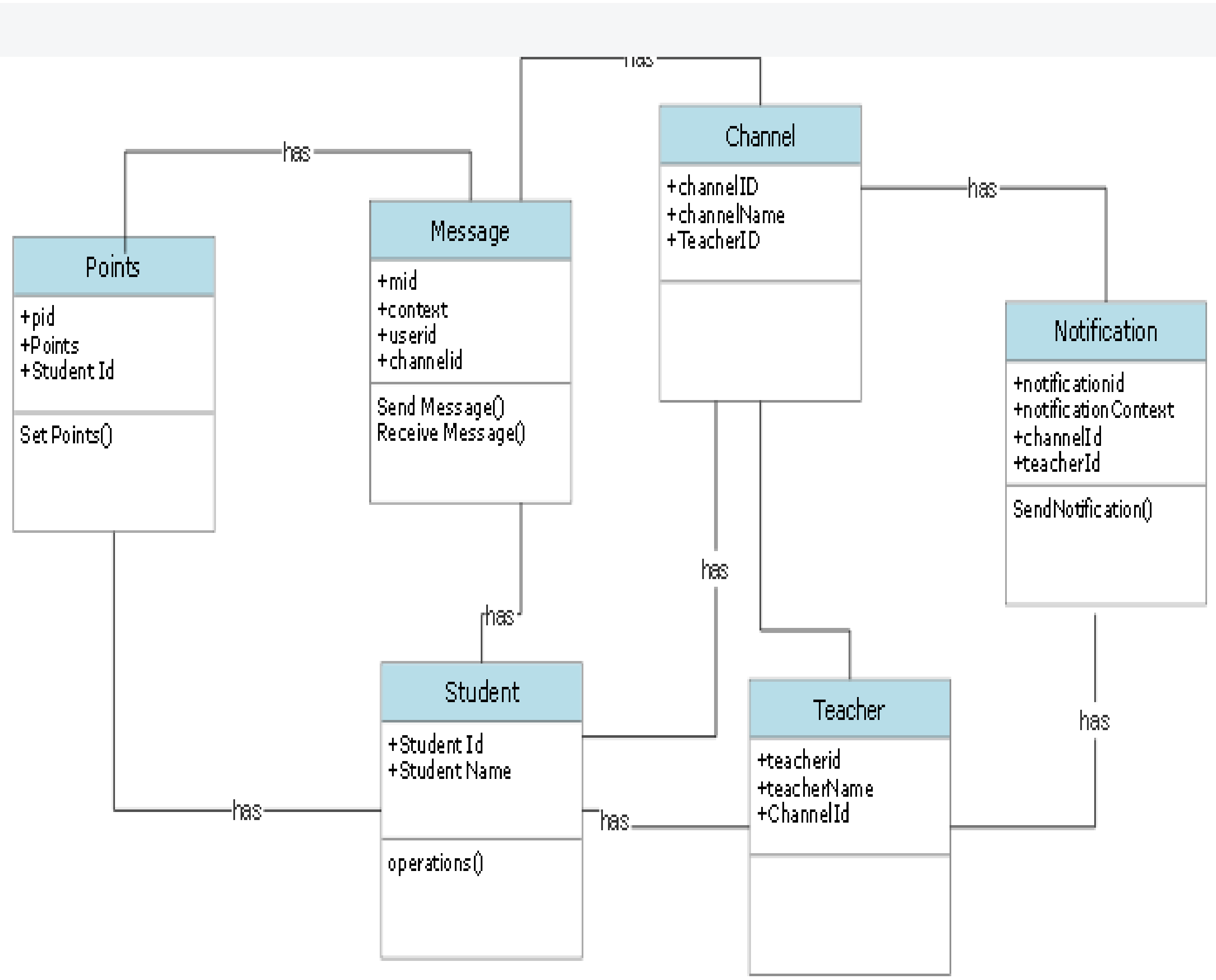


POINTS				
Field	Type	Null	Key	Extra
id	int	NOT NULL	Primary	auto_increment
points	int	NULL		
Student_id	int	NULL		
Date	date	NOT NULL		

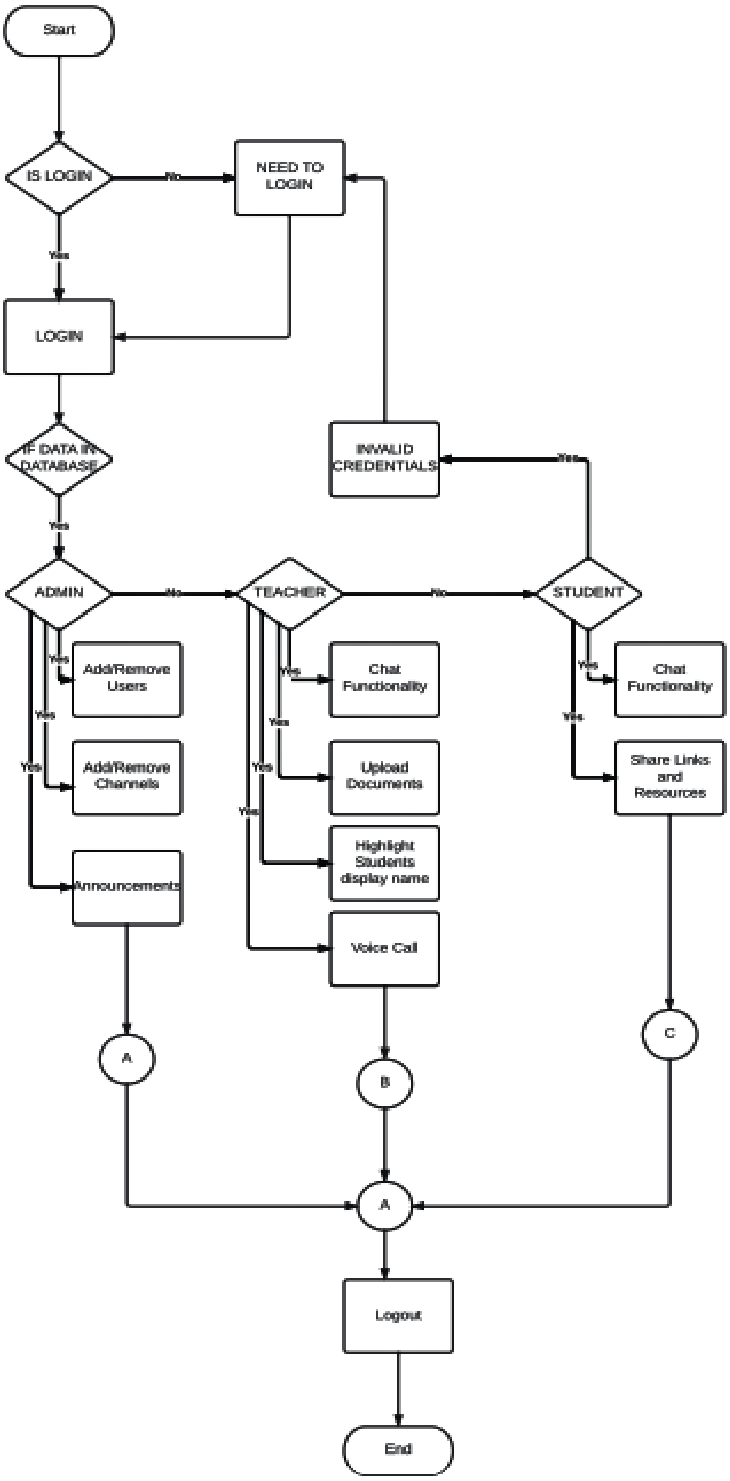
3.4 Use case Diagram



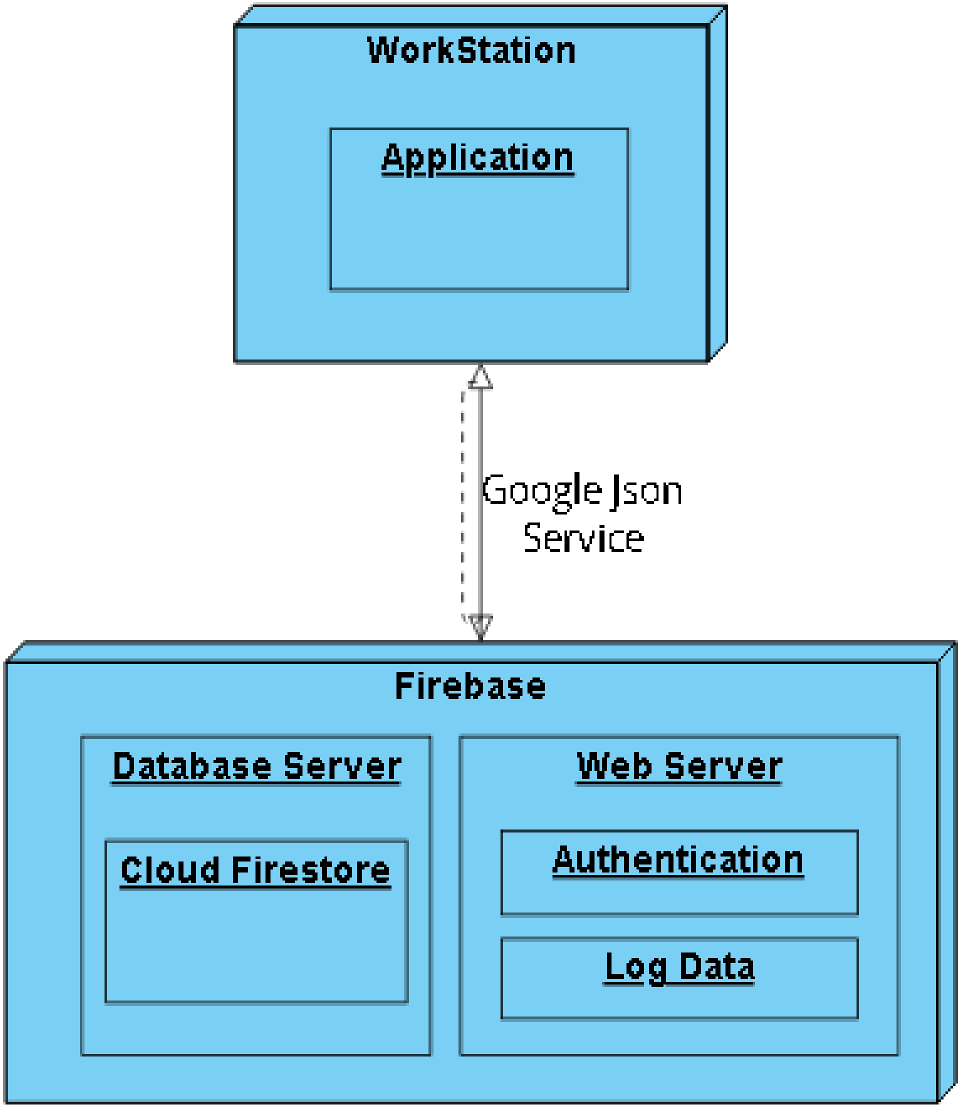
3.5 <u>Class Diagram</u>



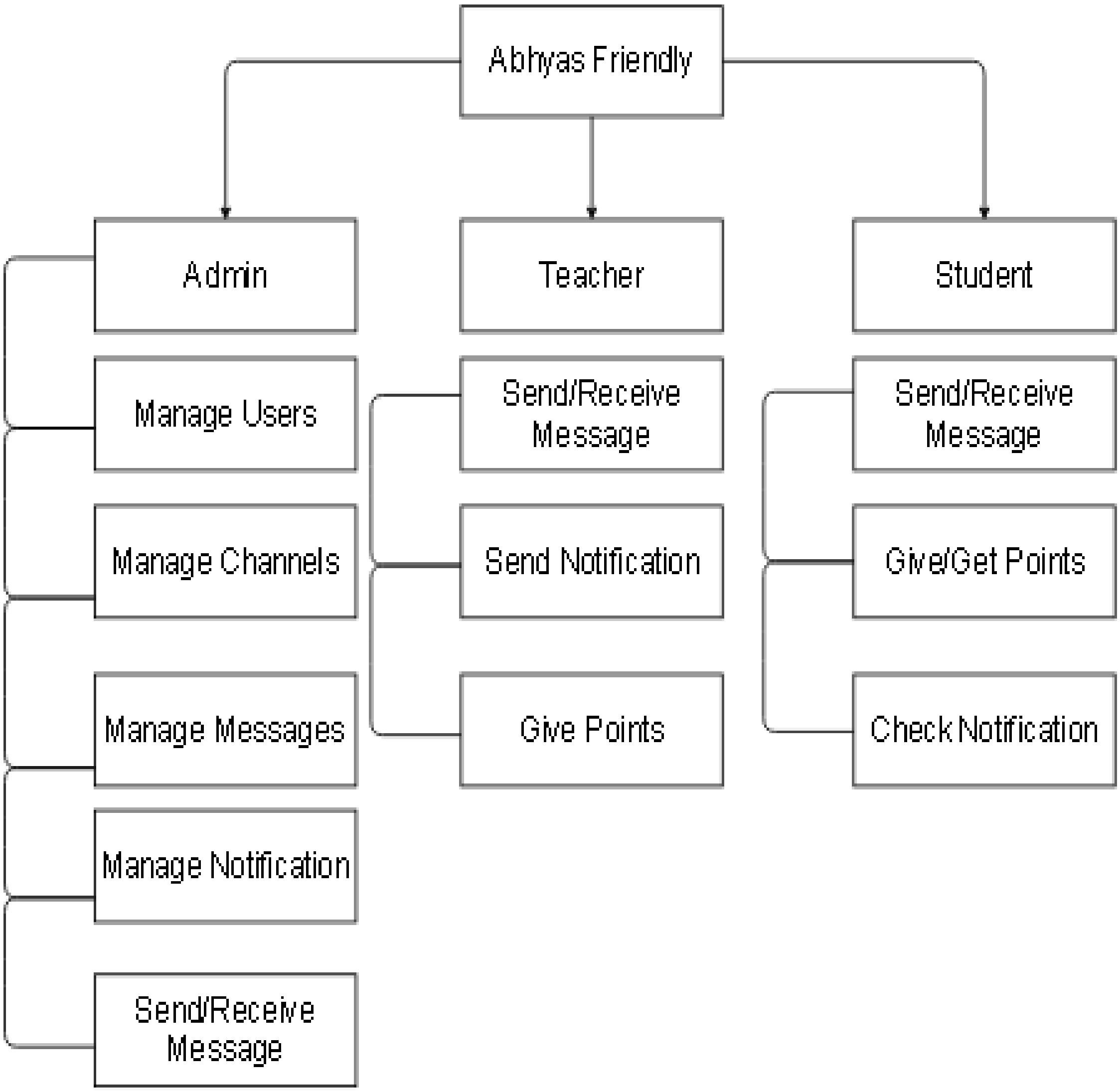
3.6 Activity Diagram



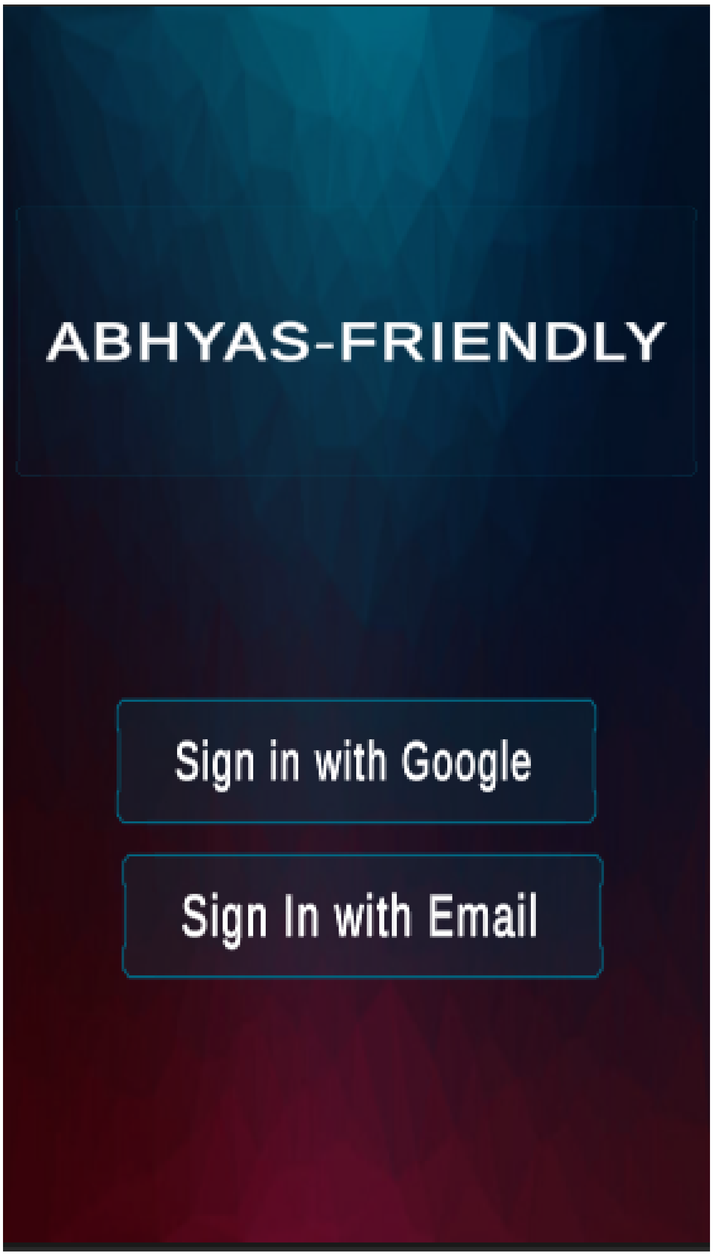
3. 7 Deployment Diagram



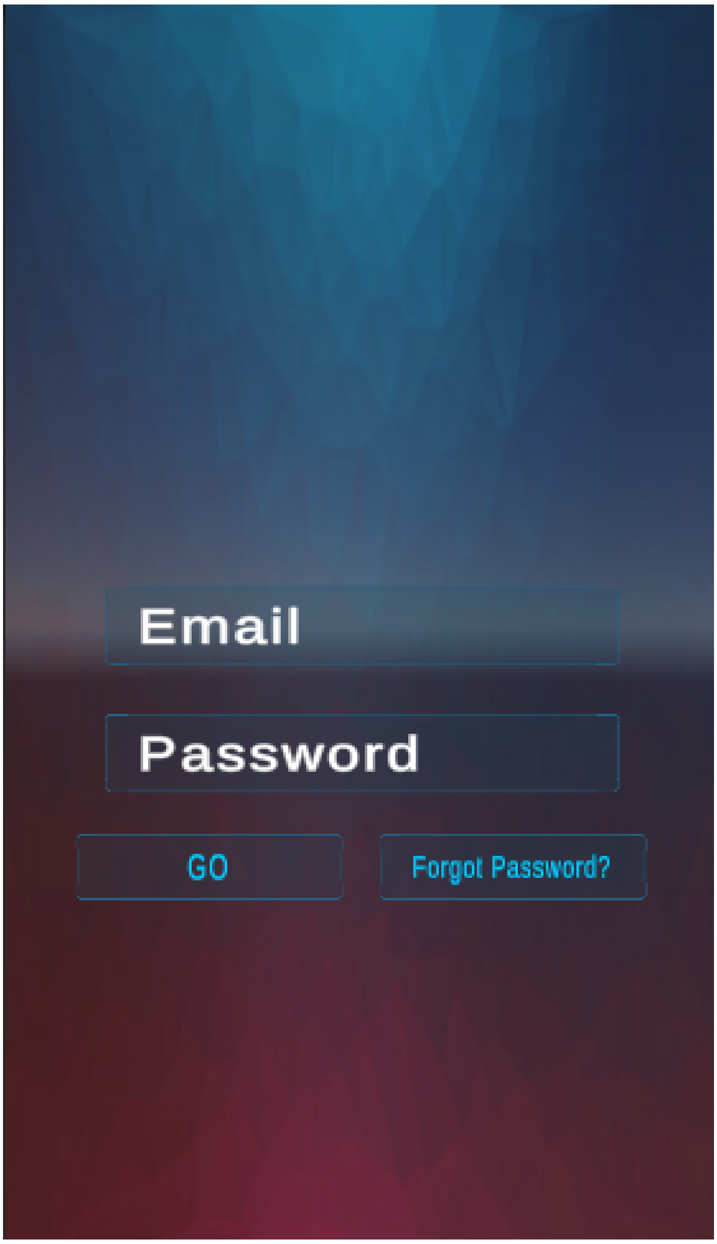
3. 8 Module Hierarchy Diagram



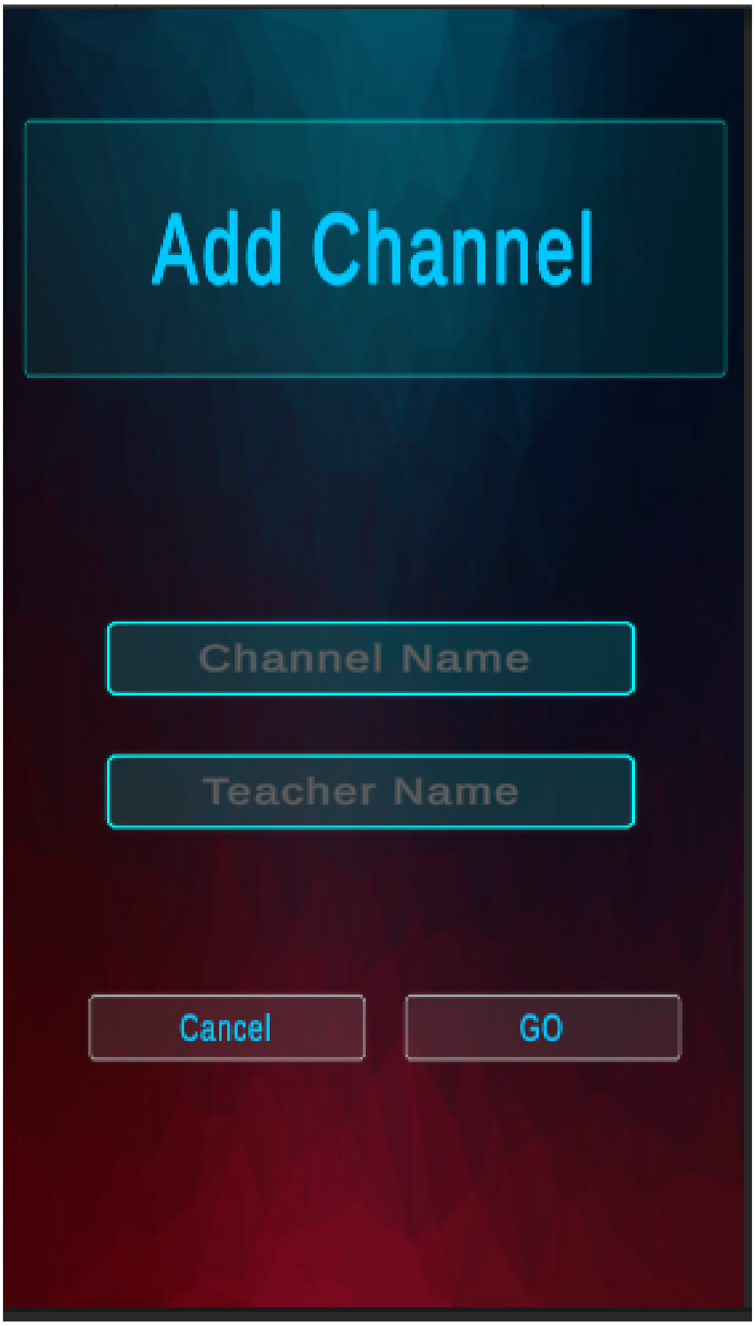
3.9 Sample Input & Output Screens

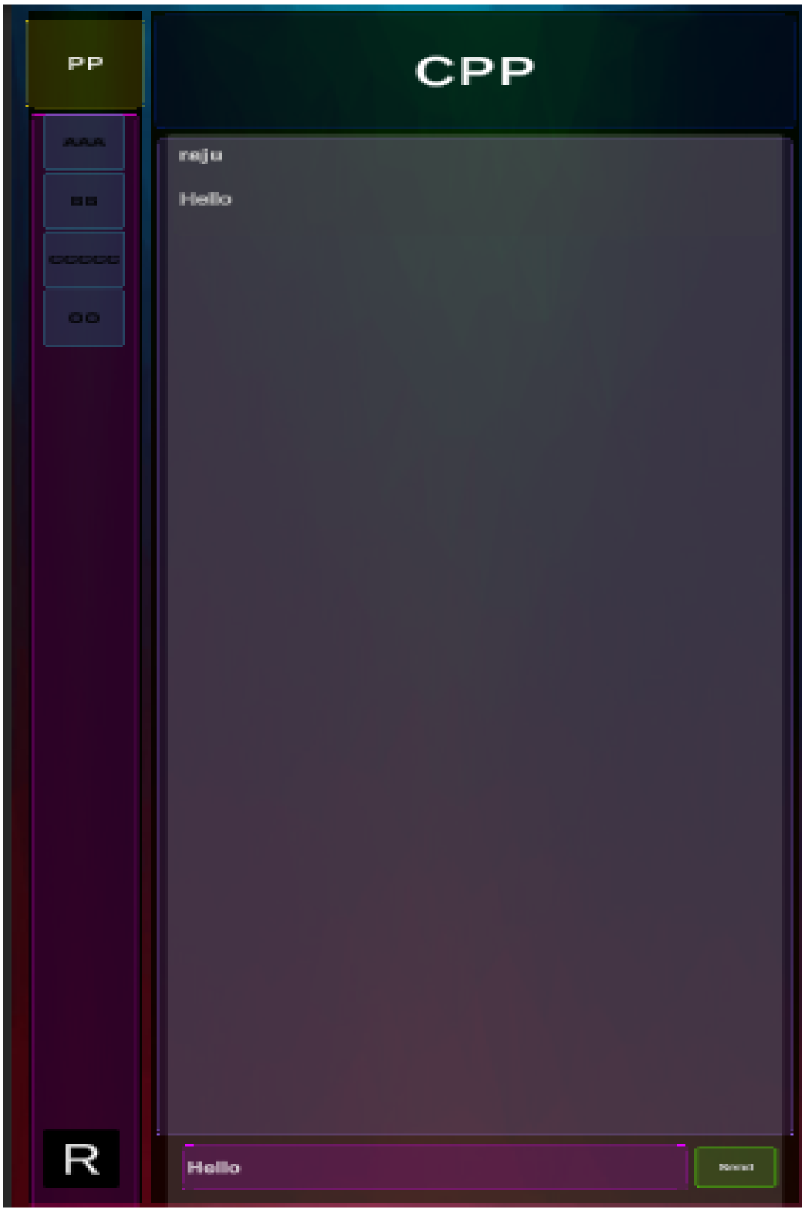


ABHYAS-FRIENDLY



Email
Password
GO
Forgot Password?

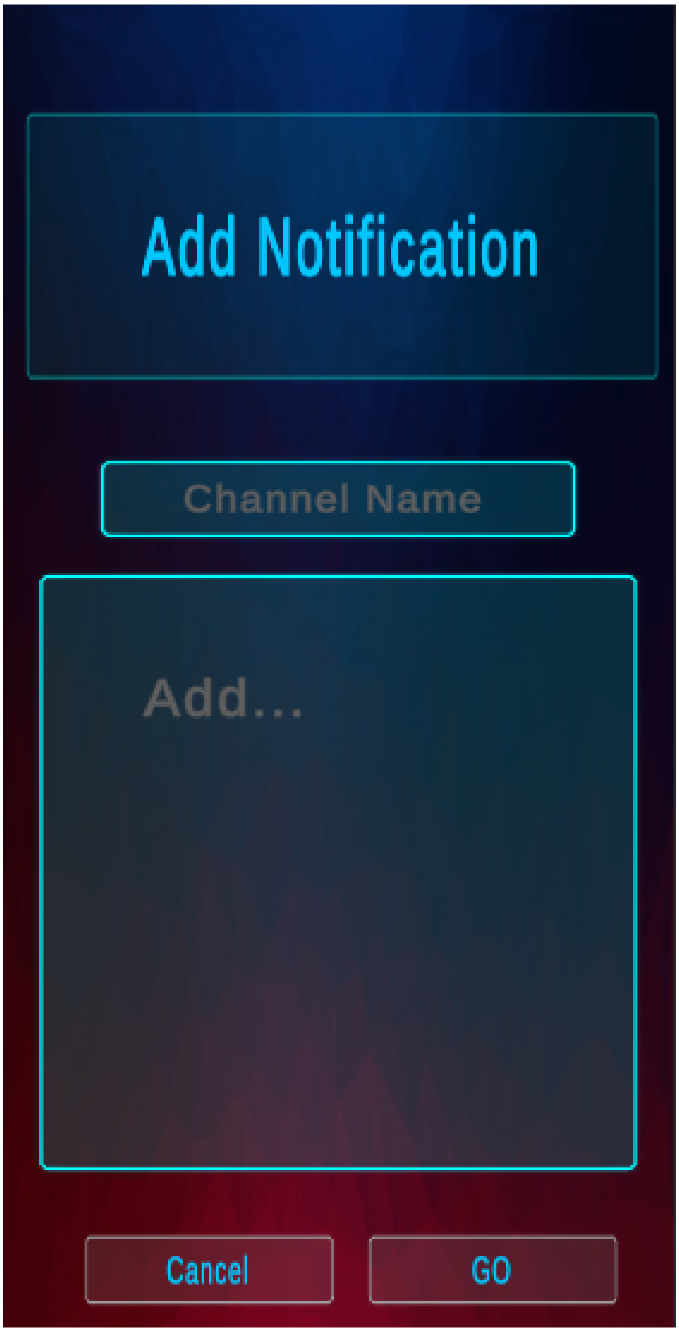

Add Channel
Channel Name
Teacher Name
Cancel
GO

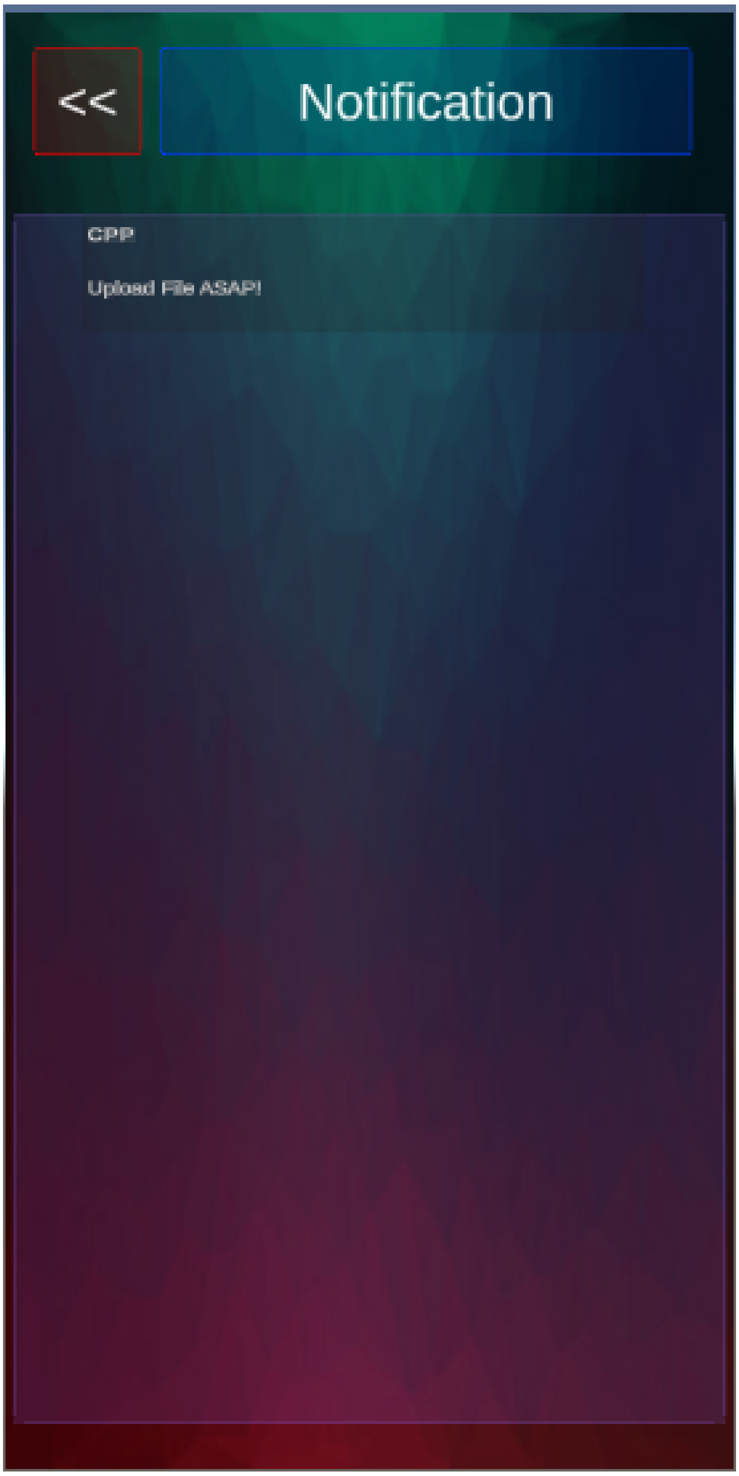


raju

Hello

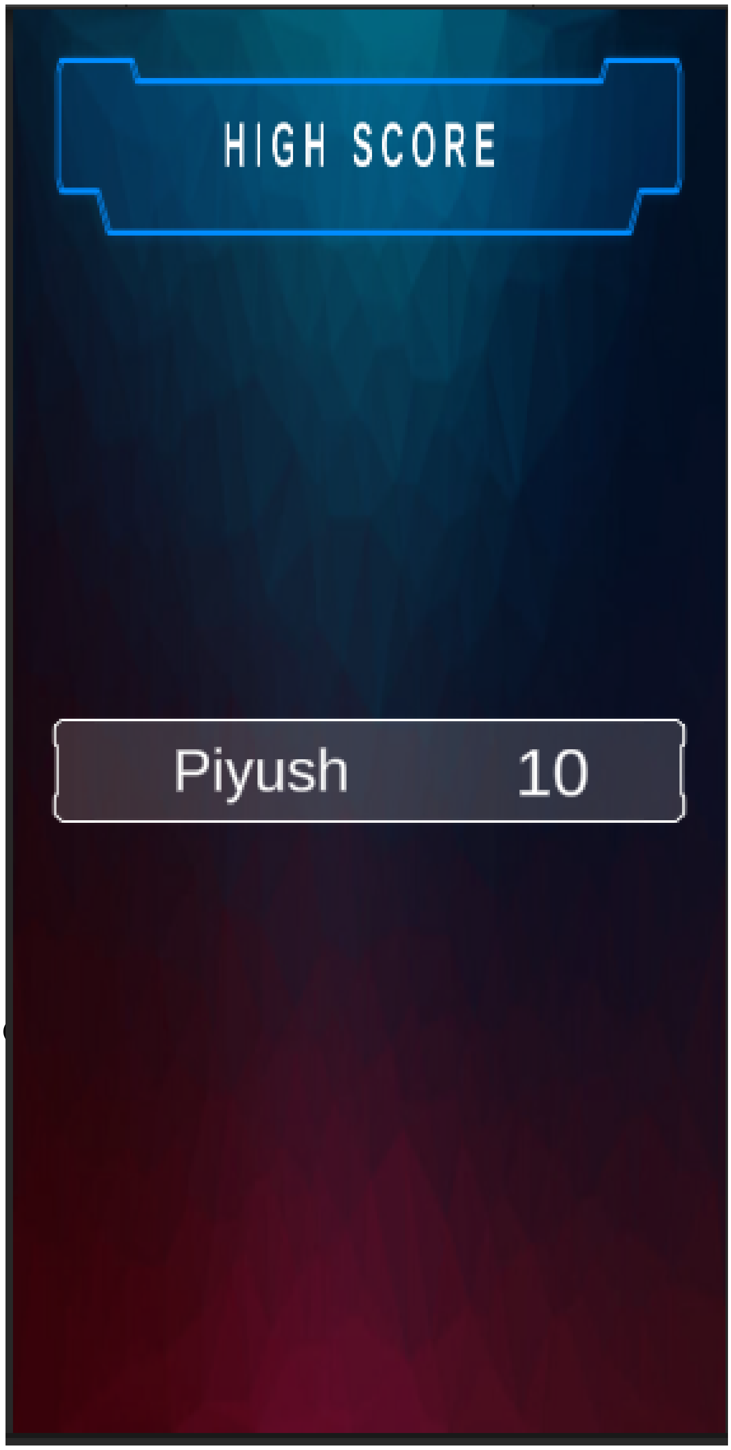
Hello


Add Notification
Channel Name
Add...
Cancel
GO



CPP

Upload File ASAP!



HIGH SCORE

Piyush 10

4 Coding

Code snippets

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using Firebase;

using Firebase.Firestore;

using UnityEngine.UI;

using TMPro;



public class FirebaseManager : MonoBehaviour

{

    FirebaseFirestore db;

    Dictionary<string, object> message = new Dictionary<string, object>();

    public TMP_InputField messsageInput;

    public GameObject messagePrefab;

    public Transform messageContainer;



    Dictionary<string, object> points = new Dictionary<string, object>();
```

```
public void CallSendMessage(string someemail)
{
   //for testing
   GLOBAL.EMAILID = someemail;


   StartCoroutine(SendMessage());
}
public void CallReferesh()
{
   StartCoroutine(ReceiveMessage());
}


IEnumerator SendMessage()
{
   print("working");
   db = FirebaseFirestore.DefaultInstance;
   message = new Dictionary<string, object>
```

```
        {
            {"ID", GLOBAL.EMAILID},
            {"Context", messsageInput.text}
        };


db.Collection("Message").Document(GLOBAL.EMAILID).SetAsync(message).
ContinueWith(task =>
            {
                if (task.IsCompleted)
                {

                }


            });


            yield return new WaitForSeconds(1f);
            StartCoroutine(InstantitateMessage(GLOBAL.EMAILID,
messsageInput.text));
```

```
    }


    IEnumerator InstantitateMessage(string email, string message)
    {
        var Imessage = Instantiate(messagePrefab, transform.position,
Quaternion.identity);
        Imessage.transform.SetParent(messageContainer, worldPositionStays:
false);

Imessage.transform.GetChild(0).GetComponent<TextMeshProUGUI>().text =
email;

Imessage.transform.GetChild(1).GetComponent<TextMeshProUGUI>().text =
message;


        yield return new WaitForSeconds(1f);
    }


    IEnumerator ReceiveMessage()
    {
        print("works");
        db = FirebaseFirestore.DefaultInstance;
        db.Collection("Message").GetSnapshotAsync().ContinueWith(task =>
        {
            QuerySnapshot snapshot = task.Result;
            foreach(var ob in snapshot)
            {
                string em = ob.GetValue<object>("ID").ToString();
                string txt = ob.GetValue<object>("Context").ToString();
                print(em+" "+ txt);
                StartCoroutine(InstantitateMessage(em, txt));
            }



        });


        yield return new WaitForSeconds(1f);
    }
}
```

5 Testing

5.1 Test Strategy

GUI Testing

GUI testing is not so focused on what the components are doing but how they communicate with each other is specified. Individual forms will be verified as per test cases.

Functional Testing

Functional tests can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended function as stated in the specification and establishing confidence that a program does what it is supposed to do. This will include unit testing for specified functionality.

Integration Testing

Testing two or more modules or functions together with the intent of finding interface defects between the modules or functions. Testing completed as a part of unit or functional testing, and sometimes, becomes its own standalone test phase. On a larger level integration testing can involve putting together a group of modules and functions with the goal of completing and verifying that the system meets the system requirements.

System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process description and flows, emphasising pre driven process links and integration points. The system as a whole will be checked for functionality.

Regression Testing

Testing with the intent of determining if bug fixes have been successful and have not created any new problem. Also, this type of testing is done to ensure that no degradation of baseline functionality has occurred. This testing will be performed to check the application for appropriate functionality after the change is made.

Unit Test Plan

Unit testing is essential for the verification of code produced during the coding phase and hence the goal is to test the internal logic of modules. Using the detailed design description as a guide, important pats are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself. All units were successfully tested.

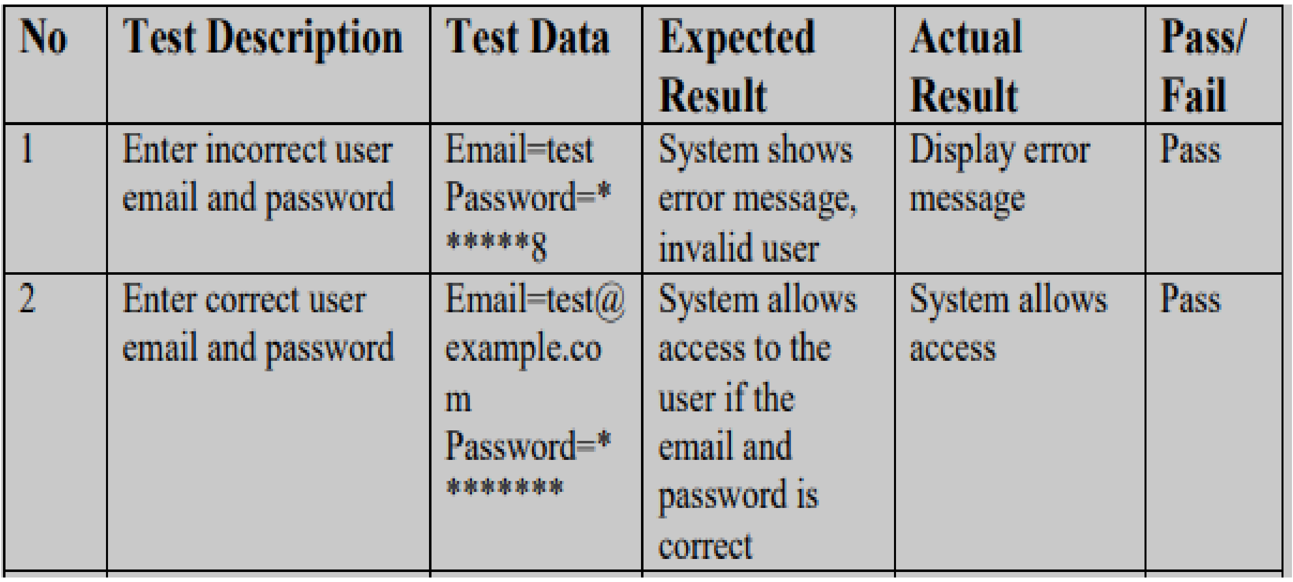
5.2 Acceptance Test Plan

It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorised entities to determine whether to accept the system or not.
User Acceptance Testing (UAT): User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose. This is also termed as End-User Testing.

Business Acceptance Testing (BAT): BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits and is quite challenging.

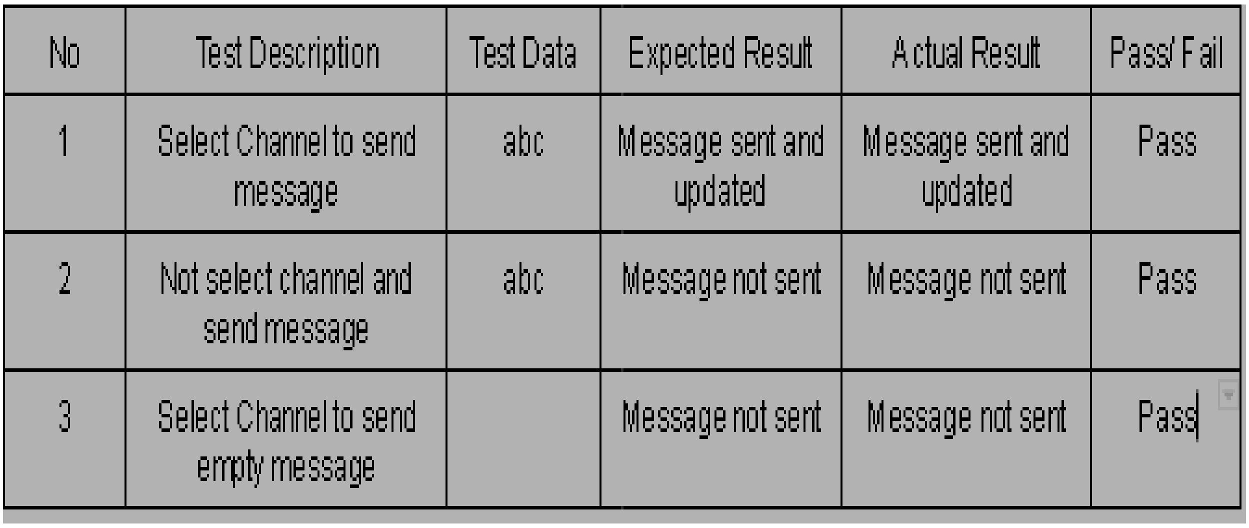
5.4 Test Cases/Test Scripts

This test case verifies the login functionality for the users



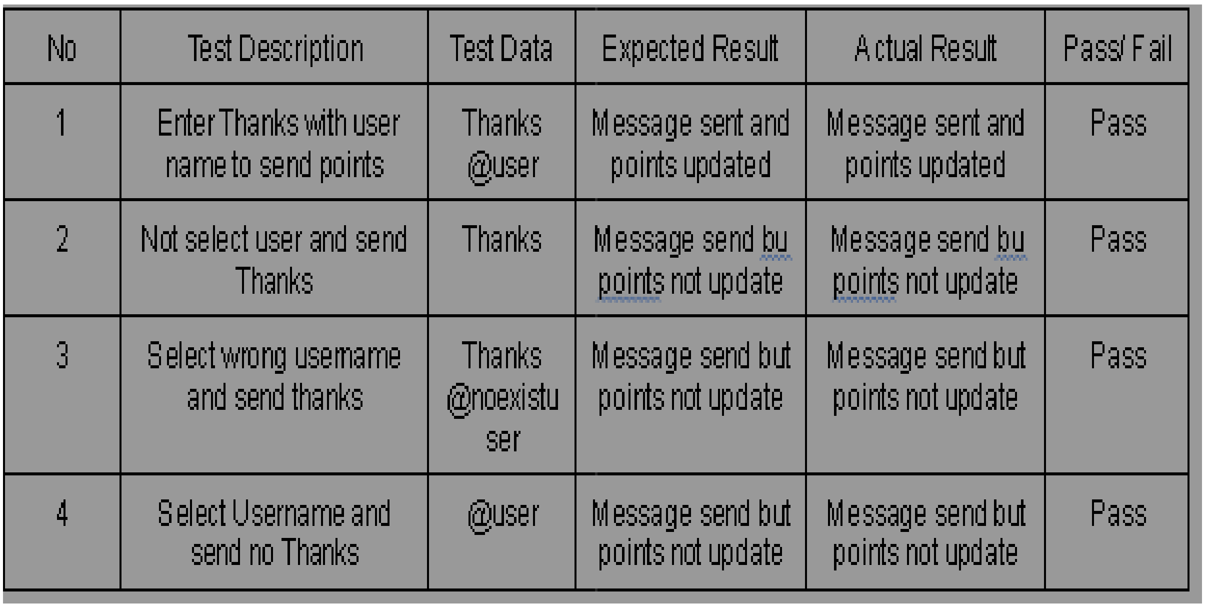
No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Enter incorrect user email and password	Email=test Password=******8	System shows error message, invalid user	Display error message	Pass
2	Enter correct user email and password	Email=test@example.com Password=********	System allows access to the user if the email and password is correct	System allows access	Pass

This test case verifies the Message module functionality for



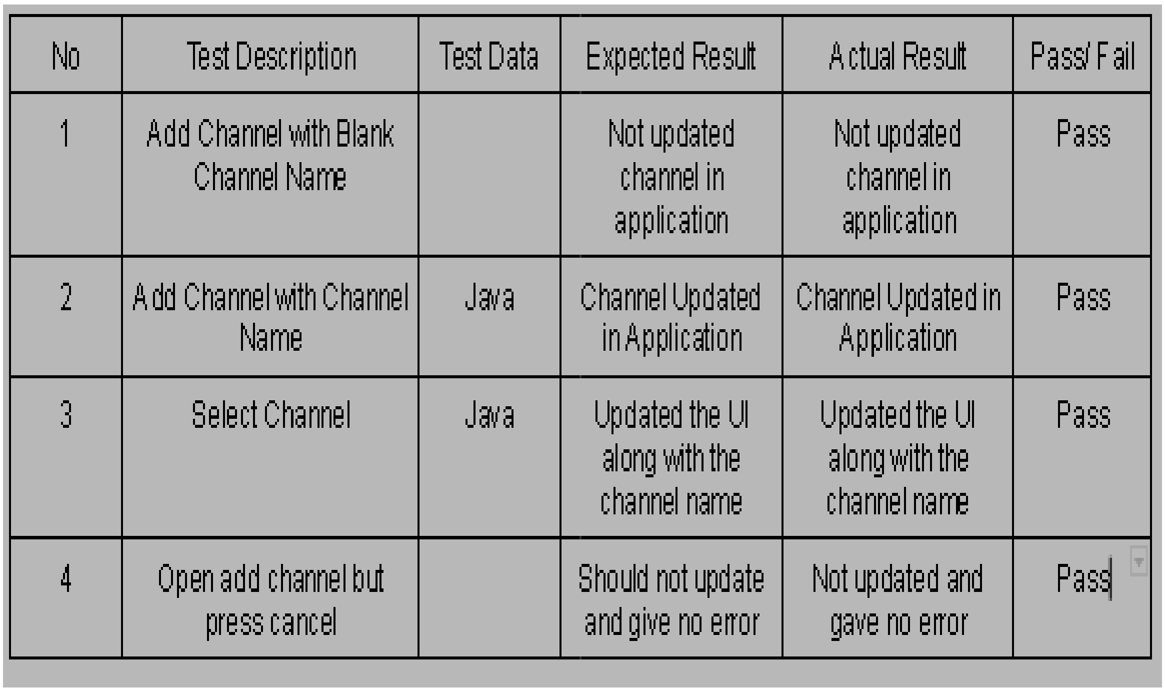
No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Select Channel to send message	abc	Message sent and updated	Message sent and updated	Pass
2	Not select channel and send message	abc	Message not sent	Message not sent	Pass
3	Select Channel to send empty message		Message not sent	Message not sent	Pass

This test case verifies the Points module functionality for



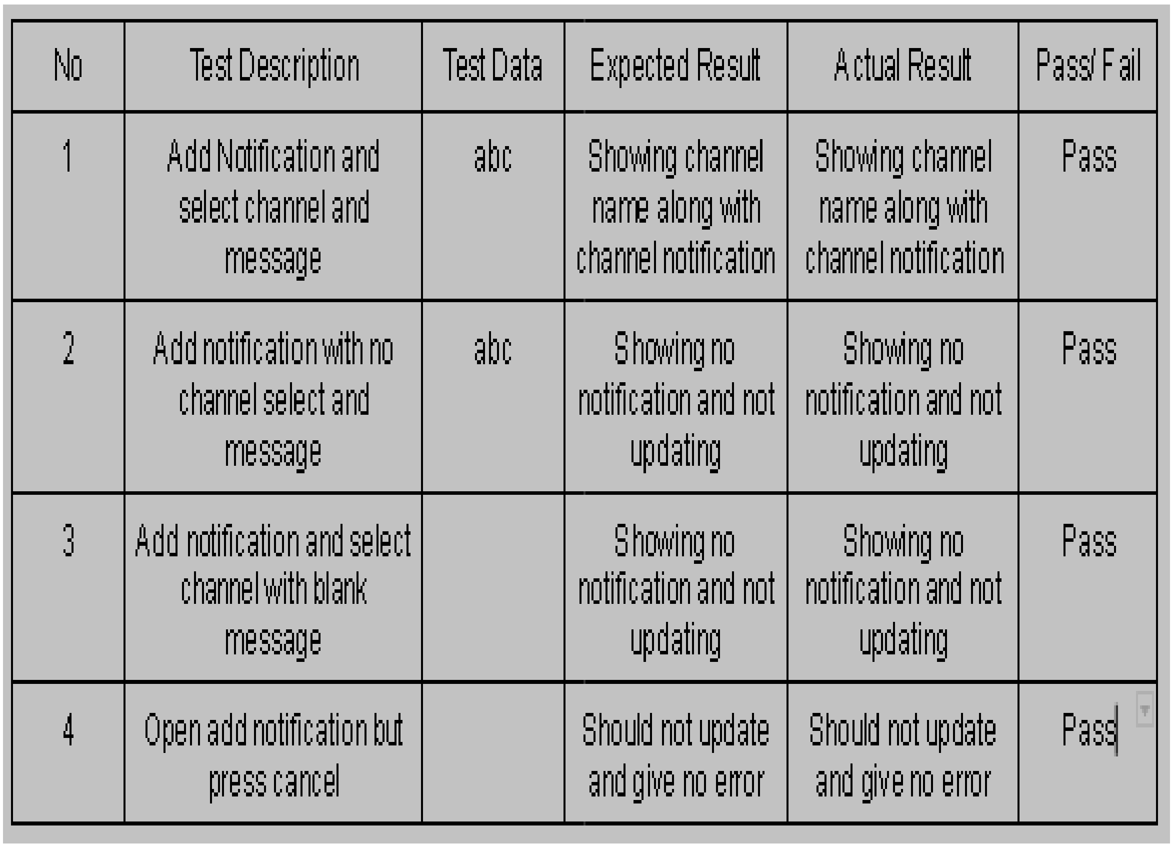
No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Enter Thanks with user name to send points	Thanks @user	Message sent and points updated	Message sent and points updated	Pass
2	Not select user and send Thanks	Thanks	Message send bu points not update	Message send bu points not update	Pass
3	Select wrong username and send thanks	Thanks @noexistuser	Message send but points not update	Message send but points not update	Pass
4	Select Username and send no Thanks	@user	Message send but points not update	Message send but points not update	Pass

This test case verifies the Channel module functionality for



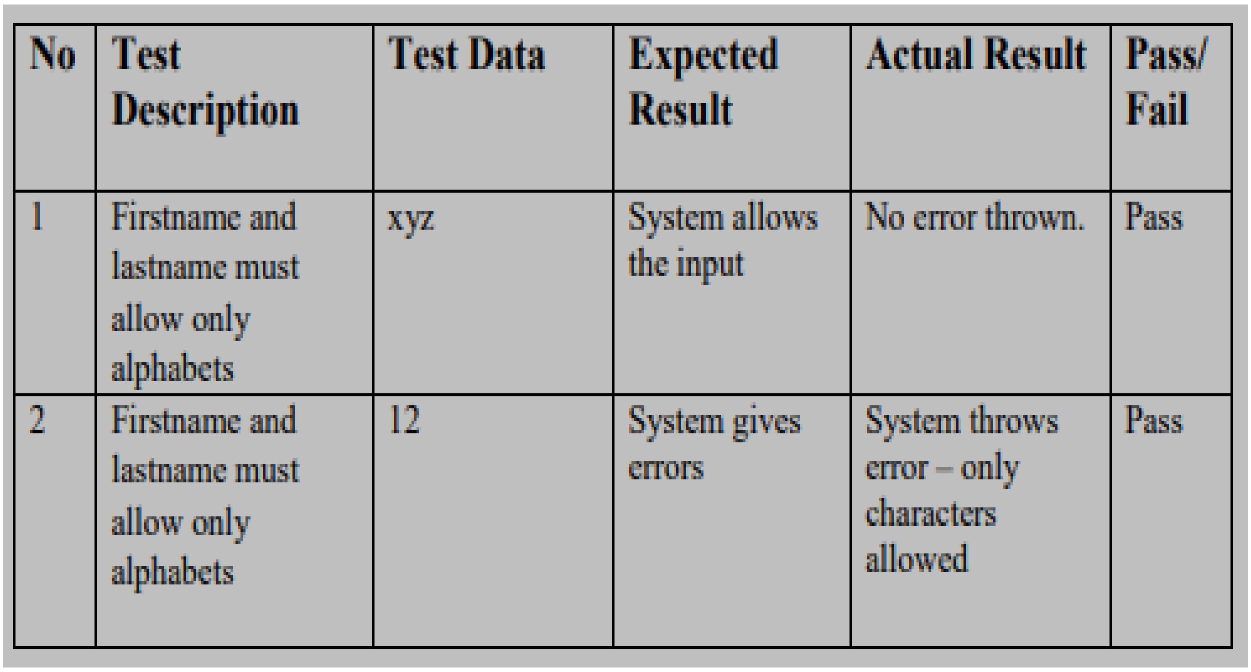
No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Add Channel with Blank Channel Name		Not updated channel in application	Not updated channel in application	Pass
2	Add Channel with Channel Name	Java	Channel Updated in Application	Channel Updated in Application	Pass
3	Select Channel	Java	Updated the UI along with the channel name	Updated the UI along with the channel name	Pass
4	Open add channel but press cancel		Should not update and give no error	Not updated and gave no error	Pass

This test case verifies the Notification module functionality for



No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Add Notification and select channel and message	abc	Showing channel name along with channel notification	Showing channel name along with channel notification	Pass
2	Add notification with no channel select and message	abc	Showing no notification and not updating	Showing no notification and not updating	Pass
3	Add notification and select channel with blank message		Showing no notification and not updating	Showing no notification and not updating	Pass
4	Open add notification but press cancel		Should not update and give no error	Should not update and give no error	Pass

This test case verifies the registration functionality for the users



No	Test Description	Test Data	Expected Result	Actual Result	Pass/ Fail
1	Firstname and lastname must allow only alphabets	xyz	System allows the input	No error thrown.	Pass
2	Firstname and lastname must allow only alphabets	12	System gives errors	System throws error – only characters allowed	Pass

5.5 Defect Report

	Project Name	**Abhyas Friendly**
	Module name / no	Channel Add
	Tester name	Piyush Panjwani
Category	**Label**	**Value**
Bug ID	ID number	#1
	Name	Package End Date -
	Reporter	Piyush Panjwani
	Submit Date	02-06-2022
Bug overview	Summary	While navigating to the main page, previous added channels are not visible.
	URL/PAGE	Main Chat
	Screenshot	-
Environment	Platform	Android
	Operating System	Android
	Application	OnePlus
Bug details	Steps to reproduce	Login and navigate to the Main Page.
	Expected result	The Previously added channel should be visible.
	Actual result	The Previously added channel is not visible.
	Description	While navigating to the main page, previous added channels are not visible.
Bug tracking	Severity	Major
	Assigned to	Piyush Panjwnai
	Priority	High
	Status	Open
	Notes	--

6 Limitations of Proposed System

1. Internet connection is required to access the system.
2. Basic knowledge of computer is required
3. Initial admin account setup must be done from the database level
4. System security is password dependent, if security of password information is not maintained and system could be in danger.
5. System requires electricity to function.
6. Firebase Cloudstore currently used is in free tier and active for a limited time period in the existing system.

7 Proposed Enhancements

1. Adding voice support for the users using the application as per channel.
2. Adding personal profile chat.
3. Adding Post feed so that other activities can be seen by other users
4. Adding notes section to get and maintain notes easily

8 Conclusion

In this drastic world of the Internet, there are many such students who are not socially active but do need company or some guidance while studying.

Applications like chatting makes it difficult for users to track important updates and resources. The communication gaps create chaos to management.

The objective of this project was to build a program for students from all around the globe, private organisations to connect together as per the topics and also they can share resources which all can be done with live interactions from anywhere.

It has a functionality where Instructors too can connect and pass their knowledge to the students as per the needs.

Instructors and students can give points to other students as a token of thanks which will not only encourage students but also will help teachers in identifying the helpful talent.

There is always a room for improvement in any software however efficient the system may be. The important thing is that the system should be flexible enough for future modifications. The system has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly.

9 Bibliography

- **Websites Google**

  https://www.google.co.in

- **Youtube**

  https://www.youtube.com

- **Stack Overflow - Where Developers Learn, Share, & Build**

  https://stackoverflow.com

10 Appendix – Cost sheet, Data sheet

Project Data Sheet

Project Name: Abhyas Friendly					Project Manager:
Project Start Date: 08 Jul 2021					Product Manager:
Clients:					Executive Sponsor: Destination
Customers					Clients Benefits
					Better customer service
					Reduce Paperwork
					Better account management
					Better data management
Project Objective Statement					Performance Attributes
The objective of this project was to build a program for students from all around the globe, private organisations to connect together as per the topics and also they can share resources which all can be done with live interactions from anywhere.					
Trade Off Matrix					Product Architecture
	Fixed	Flexible	Accept	Target	Windows 10
Scope					Unity
Schedule					C#
Response					Firebase
Stability					
Project Delay Cost Per Month:					
Exploration Factor: 8					
Business Subject / Activity					Issues and Risks
Teacher: user creation/update					
Student : user creation/update					

Channel Management : Creation/update	
Message Management : Creation/update	
Points : Add points/View history	
Major Project Milestones	
Teacher Management 15 Jul 2021	
Student Management 04 Oct 2021	
Channel Management 11 Nov 2021	
Points Management 30 Dec 2021	

11 User Manual

The following information will give more information about the processing and flow of the system

Admin module

- **Registration Module**
  - Admins can create username and password in firebase database and provide credentials to the users.
  - Email address must be unique. If the email address is already present, the system can give an error.
  - Password must be minimum 6 characters long with at least one capital, one small and one special character in it.
  - Admin can provide the details to users in the same college.

- **Add Channel**
  - Add Channel Name
  - Add Teacher Name
  - Users can access the channel after adding it.

- **Send Message**
  - Send message by selecting channel name
  - Message will update in respective channel

- **Send Points**
  - Send message by selecting channel name
  - Add Thanks and user name and send the message
  - Messages will be sent and points will update.
  - Points section will display username with high points.

- **Send Notification**
  - Send notification by selecting notification.
  - Add a message along with the channel name.
  - Notification will be sent and will be updated.
  - Users can see notifications.