MINI PROJECT 2 REPORT

ON

StuFun Web

**Submitted by:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tushar**  **Gupta** | **Jalanki**  **Nayak** | **Kartik Agrawal** | **Yash Vardhan Gautam** | **Anubhav Nag** |  |  |  |
| **181500760** | **181500291** | **181500311** | **181500830** | **181500109** |  |  |  |

**Department of Computer Engineering & Application**

**Institute of Engineering & Technology**



GLA University

Mathura – 281406, INDIA

2021

**Project Information:**

|  |  |
| --- | --- |
| **Title of Project** | **StuFun** |
| **Technical Details** | **Hardware Requirements:**  PC of minimum 4 GB RAM and 1 TB Hard disk  **Software Requirements:**  Visual Studio Code (For designing)  Web Browser pref: Chrome (For implementing)  Android Studio (For Developing)  Figma (For Designing) |
| **Project Implementation Details** | **Fully Implemented** |
| **Project Working Period** | Start Date: 15 January 2021  End Date: 20 April 2021 |

**Summary of the Project Work:**

|  |
| --- |
| Students nowadays use WhatsApp not only as a messaging app but also as an extension of their classroom, use it not only to interact with the environment during study hours but also to share resources, connect with teachers at any given time, learn and study groups, live doubt sessions, etc. Though the current online education platform gives students the freedom to study at any time with lots of resources it lacks a sense of discipline, sociality, and status of learning. Through this project, the team hopes to build a platform that is much effective than present applications. This project is an amalgamation of an messaging platform and classroom platform, where we aspire to achieve a sense of discipline that is generally missing in chatting applications.  We have already developed an android application named ‘StuFun’ in mini project-1 that is up in the running. A web version of the application based on **MERN Stack** has been developed by the team during this mini project. This cross-platform project will not only increase the reach of the platform but will also help us learn about developing and managing cross platform software. |

**ACKNOWLEDGEMENT**

On successful completion of the project, we would take a privilege to thank the supporting manpower of our faculty members, and all our friends & colleagues with gratitude. However, we wish to make special mention of the following.

First of all, we are all thankful to our mentor **Mr. Mandeep Singh** and all the friends and mentors of the CEA department under whose able guidance we were able to complete the project up to the valuable mark. We wholeheartedly thankful to them for giving us their valuable time and attention, and for providing us a systematic way, for the completion of the project in allotted time.

**DECLARATION**

We hereby declare that the work that is being presented in the Mini Project**,** is in partial fulfillment of the requirements for Mini Project viva voce, is an authentic record of my work carried under the mentorship of **Mr. Mandeep Singh.**

|  |  |
| --- | --- |
| **Name of Candidate: Anubhav Nag**  **Roll. No.: 181500109**  **Course: B. TECH CS**  **Year: 3rd Year**  **Semester: VIth Semester**  **Signature of Candidate:** | **Name of Candidate: Jalanki Nayak**  **Roll. No.: 181500291**  **Course: B. TECH CS**  **Year: 3rd Year**  **Semester: VIth Semester**  **Signature of Candidate:** |
| **Name of Candidate: Kartik Agrawal**  **Roll. No.: 181500311**  **Course: B. TECH CS**  **Year: 3rd Year**  **Semester: VIth Semester**  **Signature of Candidate:**  **Name of Candidate: Tushar Gupta**  **Roll. No.: 181500760**  **Course: B. TECH CS**  **Year: 3rd Year**  **Semester: VIth Semester**  **Signature of Candidate:** | **Name of Candidate: Yash Vardhan Gautam**  **Roll. No.: 181500830**  **Course: B. TECH CS**  **Year: 3rd Year**  **Semester: VIth Semester**  **Signature of Candidate:** |

**ABSTRACT**

With a market cap of around $2 billion by 2021, online education is now one of the fastest-growing industries in India. The major reasons being, we are in middle of a tech revolution providing cheap internet and a large sector of the population that has high aspirations but low incomes.

Students nowadays use WhatsApp not only as a messaging app but also as an extension of their classroom, use it not only to interact with the environment during study hours but also to share resources, connect with teachers at any given time, learn and study groups, live doubt sessions, etc. Though the current online education platform gives students the freedom to study at any time with lots of resources it lacks a sense of discipline, sociality, and status of learning. Through this project, the team hopes to build a platform that is much effective than present applications. We’ve already developed a running android application and now look forward to develop its web counterpart.

In this project, we will build a Web application using MERN Stack Technology that will serve both functionalities of a chatting application and an educational application. The inspiration for the chatting application has been taken from WhatsApp Web Functionality as it being the most popular and primary daily chat application on Mobile Phone as well as Desktop. The inspiration for the educational part hasn’t been taken from a single platform but many, primarily being the Google classroom platform and the rest is the QnA section of various online educators such as the likes of Udemy, Coursera, Udacity, etc. The building blocks of the project being ReactJS, NodeJS, ExpressJS, MongoDB (or any other Database like Google Firebase), and the IDE is Visual Studio Code.

The Website has two categories of users viz. Teacher and Student. Where a Teacher can create classrooms, add students to it, discuss doubts and give an assignment, a Student can join a classroom, download various resources provided by the teacher, submit his assignments, and can also chat with his fellow students. The choice of MERN Stack as the developing environment is sole because of its wide reach and its ease of use.

**CONTENTS**

|  |  |
| --- | --- |
| Certificate | ii |
| Synopsis | iii |
| Acknowledgment | iv |
| Declaration | v |
| Abstract | vi |

1. **Introduction ...…………………………………………….……………………...…….3-8**
   1. Motivation …….……………...………………………………….…………………...3-4
   2. Objective………….…………...……………………………………………………...4-5
   3. Technologies Used….………………………………………………………………...5-8
      1. MERN Stack………………………………………………………………5
      2. MongoDB……………………………………….……….......................5-6
      3. Express.js……………………………….………………………………6-7
      4. React.js…………………………….…………………......…………….7-8
      5. Node.js……….…………………….…………………….…………….….8
      6. Visual Studio code………………………………………………………...9
2. **Software Requirement analysis ...……………………………………………………9-11**
   1. Problem Statement…………………………….……………….…...………………….9
   2. Define Module and Functionalities…………….………………...………………...10-14
      1. Application description ………….………………………………......10-12
      2. Functional Requirement……………………………………………...….13
      3. Non-Functional Requirement….………………………………………...14
3. **Software Design ...…………………………………………………...…………...….15-29**
   1. Data Flow Diagram ………………………….…………………………...…………. 15
   2. UML Diagram …..……………………………….……………………...…………16-22
      1. Class Diagram …...………………...…………………...…………….….16
      2. Activity Diagram …………………………………………………….17-21
      3. Use Case Diagram………….….……………………....……………...…22
   3. Database Design …...………………………………………...……...……………..23-29
      1. Er Diagram ………………………………….…………………………...23
      2. Table Description …..………………………….…………………….24-29
4. **Implementation and User Interface …………………………..…………………....25-42**
5. **Reference ….……………………………………………….……………………….…...43**

**INTRODUCTION**

There are numerous phone chatting apps worldwide, among those WhatsApp scores the highest rank around the globe both in android and iOS. WhatsApp was founded by Jan Koum and Brian Acton who had previously spent 20 years combined at Yahoo. WhatsApp joined Facebook in 2014 but continues to operate as a separate app with a laser focus on building a messaging service that works fast and reliably anywhere in the world. As per the record, it serves 2 billion users worldwide. WhatsApp is available in 180 countries and 60 different languages. Due to its many features, it is covering large users. In 2015 WhatsApp launched its web version which would serve as an extension to the android part. They stated:  "Our web client is simply an extension of your phone: the web browser mirrors conversations and messages from your mobile device—this means all of your messages still live on your phone". The WhatsApp user's handset must still be connected to the Internet for the browser application to function.

The development of the android app ‘StuFun’ has already shown us the path to develop a chatting model that has already met our thoughts and needs. Developing a web model based on the same platform would be the next best possible step, as a web platform along with a mobile platform would not only help access the data on different systems but would also show us the vast possible use of a cross-system accessible software.

With the recent progress of ECMAScript 2015, NoSQL databases and Node.js, as well as the increasing resources and capabilities of cloud service providers, web development has become practical and abstracted enough to enable even sophisticated full-stack web solutions to be created with significantly lesser amounts of effort and work hours than what was required by the previous popular technologies. With such advancements in the MERN Stack, developing a web app equivalent of our android application seemed the next best challenge possible.

This project is limited, both in scope and in available resources. Because of this, suitable technologies are limited to only open-source softwares. The technology stack used for the project in this study consists of MongoDB for the data storage layer, Node and Express for the service layer with Json Web Token for API authentication, React for the frontend client and CSS with Bootstrap for the client styling. The reasoning behind these choices is expounded on a per-technology basis in later chapters.

Due to the limited scope, certain aspects of the service such as data backup and real-time sync over wireless connections are not covered in the report. The thesis will only explore the 4 core technologies comprising the MERN stack, namely MongoDB, Express, React and Node, as well as certain smaller libraries immediately attached to them in the final service. The thesis includes 5 sections in total, starting with an introduction and explanation of the problem that the project constructed in the thesis seeks to solve, then conducting a technical investigation of each of the 4 core technologies including their practical application in the service, and ending with the conclusion explaining details about the final product and the creation process. The topic of the thesis was chosen both out of personal interest and because the technologies in question were already in the works in academic context, which allowed for a combination of the two projects.

* 1. **MOTIVATION**

The idea of accessing our data and digital services simultaneously across the many devices we own is a very fascinating one and has achieved great success in recent years. This allows us to take benefit of the different platforms and devices and we own without creating multiple versions of the same account.

As we all know in today’s digital world social messaging apps like WhatsApp, Telegram, Messenger, etc. are the most popular way to connect with friends, family, and businesses. These Apps allow us to do personal chats with one particular person or we can create groups where we can chat with a lot of people. Many businesses are now being run on WhatsApp as a result many official and professional work is now being done on such apps.

Nowadays such groups are also used for educational purposes where teachers create groups of their classes for announcements and can share information like class activities. But using these groups for educational purposes has some limitations. Students can be added only with their mobile numbers which are generally of their parents. Whenever a student is having some doubt regarding a topic and he/she wants to discuss it with the class or the teacher wants to provide any information or wants to do any announcement they share their information in the group and a discussion starts which most of the time goes in an informal direction.

With the emergence of digital classrooms in the educational landscape, it has become easier for teachers to arrange online teaching and learning sessions. It enables teachers to collaborate with students from anywhere and at any given point in time. All they need to do is log in to their e-learning portal and begin the interactive teaching and learning process.

Though an android application is most the feasible solution in such a case there may be cases in which the information passed is best suited for a computer environment, like a presentation or a self-explanatory videos or scientific problems and numerical. In other cases, there might be activities that are best performed when they are done on a computer environment like taking a test, or presenting a topic or coding and problem solving. Using a computer browser environment also help us see and process the various file types that aren’t very optimized for a mobile view.

Therefore, a web app based on the platform of ‘StuFun’ would not only give us a quick and lightweight model of our platform but would also increase the prowess of our other model and would give the user, freedom to educate and learn anytime, anywhere and from any device.

* 1. **OBJECTIVE**

The objective of this project is to create a web browser application based on MERN Stack within the given time limit. The android application to be created is named “**StuFun**”.

The primary application objective of this web app is to serve both as a chatting app and as a classroom app that focuses on the student and teacher side of the problem. The project must have begun by 15 January 2021 and should be completed by the end of 20 April 2021 as fully functional.

The educational objective of the project is that by the end of the project all team members would have learned about the basics of the web app development and with all the parts of MERN Stack viz: MongoDB, Express.js, React.js and Node.js.

As the app is a chat application it must have a functional chat infrastructure in two modes, viz a person-to-person mode and common group mode, the app should also be able to provide basic functionalities that everyday use apps like WhatsApp, Messenger, Hike provide.

On the part of the classroom, the objective is to create a classroom with a chat room/groups like infrastructure where students can interact with each other and their teachers on matters of their class and studies.

Another major objective of this app on the classroom side is to provide a platform for resource sharing where educational material can be provided to student’s time by time. As to make the classroom a strong tool for study and making the experience as real as possible we add another objective of creating a question panel in the classroom where it would be compulsory for the teacher to answer questions of the student.

Carrying forward this objective of making the learning environment as real as possible the team would also be adding a feature of real-time quiz where students can give academic quizzes provided by their teachers.

**1.3 TECHNOLOGY AND TECHNIQUES USED**

During this project we would be exploring the idea of technology stack in which only a fixed set of technologies is used. In a technology stack we use a set of frameworks and tools to develop a software. Great attention is paid to the combination of technologies used as they will be able to work in harmony.

Some examples are:

* MERN - (MongoDB, ExpressJS, React, NodeJS)
* MEAN - (MongoDB, ExpressJS, Angular, NodeJS)
* LAMP – (Linux, Apache, MySQL, PHP)

**We would be using MERN Stack to develop this project.**



MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

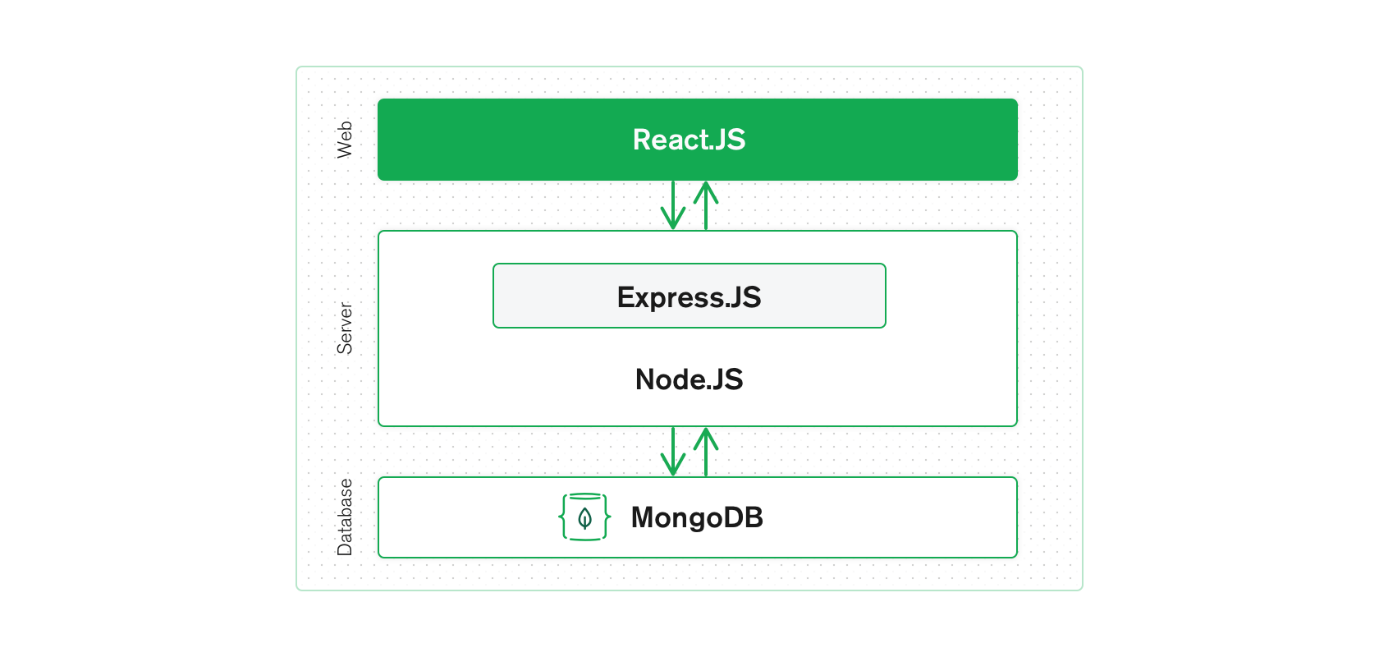
* MongoDB - document database
* Express(.js) - Node.js web framework
* React(.js) - a client-side JavaScript framework
* Node(.js) - the premier JavaScript web server

MERN is one of several variations of the MEAN stack (MongoDB Express Angular Node), where the traditional Angular.js frontend framework is replaced with React.js. Other variants include MEVN (MongoDB, Express, Vue, Node), and really any frontend JavaScript framework can work.

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

**1.3.0 MERN Stack**

The MERN architecture allows you to easily construct a 3-tier architecture (frontend, backend, database) entirely using JavaScript and JSON. The technology MERN stack used for the project in this study consists of MongoDB for the data storage layer, Node and Express for the service layer with Json Web Token for API authentication, React for the frontend client and CSS with Bootstrap for the client styling.



**1.3.1 React.js FRONTEND**

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

React’s strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you’d expect from a modern web framework: great support for forms, error handling, events, lists, and more.

**1.3.2 & 1.3.3 Express.js and Node.js SERVER TIER**

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs) or GETs or POSTs from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB’s Node.js drivers, either via callbacks for using Promises, to access and update data in your MongoDB database.

**1.3.4 DATABASE TIER**

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you’re going to want a database that’s just as easy to work with as React, Express, and Node.

In our project we have used Firebase as our Database. The Firebase Realtime Database is cloud-hosted. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

**1.3.5 Visual Studio Code IDE**

Visual Studio Code (famously known as VS Code) is a free open-source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Docker file and all of MERN stack. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

**SOFTWARE REQUIREMENT ANALYSIS**

**2.1 PROBLEM STATEMENT**

Chatting apps allow us to do personal chats with a particular person or we can create groups where we can chat with a lot of many people. Many businesses are now being run on WhatsApp as a result many official and professional work is now being done on such apps.

Education doesn’t remain aloof from such apps; nowadays such groups are also used for educational purposes where teachers create groups of their classes and can share information like class activities, pdf notes, video notes, e-books, etc.

But using these groups for educational purposes has some limitations. There are various cases like –

* Do the apps comply with GDPR guidelines and especially in the case of a student?
* Students can be added only with their mobile numbers which are generally of their parents and they quickly become a pit of unknown digits and faceless profiles.
* Whenever a student is having some doubt regarding a topic and he/she wants to discuss it with the class they share their doubts in the group and a discussion starts which most of the time goes in an informal direction.
* Students can be added only with their mobile numbers which are generally of their parents and they quickly become a pit of unknown digits and faceless profiles.
* Whenever Teacher wants to provide any information or wants to do any announcement, they post it in the group and a discussion starts which most of the time goes in informal direction.

**2.2 DEFINE MODULES AND THEIR FUNCTIONALITIES**

**2.2.1 Application Description**

The whole application consists of 2 actors i.e. Teacher and Student. Both these actors have some same and some different functionalities which differentiate the actors inside the application. These functionalities are:

**FOR TEACHER**

1. **Registration/Login -** Teachers can register themselves for 1st time with a unique email address which will be its identity inside the Application. After registering their email address, Teachers have to enter their details like Name and organization name to complete the registration process. Once the Teacher is registered, they can log in with the same email and password which was set during Registration.
2. **Chat -** This is the Chat Option where users can chat with someone just like other Chatting Application. Users can send text messages, images to other users. It consists of various other options like clear chat in which the user can clear the whole chat from their side. Users can also view the status of the messages i.e. delivered, seen.
3. **Create Classroom -** This is the create classroom option where the Teacher can create the classroom by entering the class details like the subject. Every class has a unique code which is called Class ID which will be entered by students to join the classroom. Teachers can also send the invite to students to join the classroom via email or SMS.
4. **Add Announcements -** This is the feature that keeps the students updated about class activities. The teacher can add an announcement for various activities inside the classroom. Teachers can also edit and delete the announcement as per their choice.
5. **Discussion -** This is the feature with the help of which students can resolve their queries and can interact with the teachers. In this feature, the Teacher can respond to the queries of the various students which they have asked inside the discussion panel. Teachers can also update and edit their answers inside the discussion panel.
6. **Student Management -** Inside the class, the teacher can manage the list of Students who are part of the classroom. The teacher can delete and block the students as per their choice.
7. **Delete Classroom -** The teacher can also delete the classroom. After the deletion of the classroom, everything related to that classroom like announcements made by the teacher, queries asked by students, etc. will be deleted.
8. **Manage Profile -** This functionality is the same for both students and teachers. From there student can modify their details like Name, email id, password, name of the organization, etc.

**FOR STUDENT**

1. **Registration/Login -** Students can register themselves on the application using a unique email address just like a teacher. After registering their email address, Students have to enter their details like Name, Organization name, Course, Branch to complete the registration process. Once the student has registered themselves, they can log in into the application with the email ID and password which they have entered during registration.
2. **Chat -** Chat functionality of the application is the same for both Teachers and Student. Here users can chat with someone just like other Chatting Applications. Users can send text messages, images to other users. It consists of various other options like clear chat in which the user can clear the whole chat from their side. Users can also view the status of the messages i.e. delivered and seen.
3. **Join Classroom -** Just like a teacher can create a classroom with a unique ID, Students can join any classroom by entering the unique id known as Class ID. After entering the id, the student can enter into the class and view the announcements and ask their queries in the discussion panel.
4. **View Announcements -** Just like the teacher can add the announcements inside the class, a student can view the announcements inside the class. Students can't modify or edit or update any announcement. They can only view them.
5. **Ask Query -** In this functionality, Student can ask their queries inside the Discussion Panel. The student can't answer the queries. They can only ask queries and can also view the queries asked by other students with their responses if any.
6. **View Classmates-** Students can view their classmates inside the classroom and can chat with them from there. With this feature, students can get the details of their classmates**.**
7. **Manage Profile -** This functionality is the same for both students and teachers. From there student can modify their details like Name, email id, password, name of the organization, course, branch, etc.

**2.2.2 Functional and Non-Functional Requirements**

**Category 1 users:** Personal chat

**Category 2 users:** Educational Groups (Classroom) admins and users

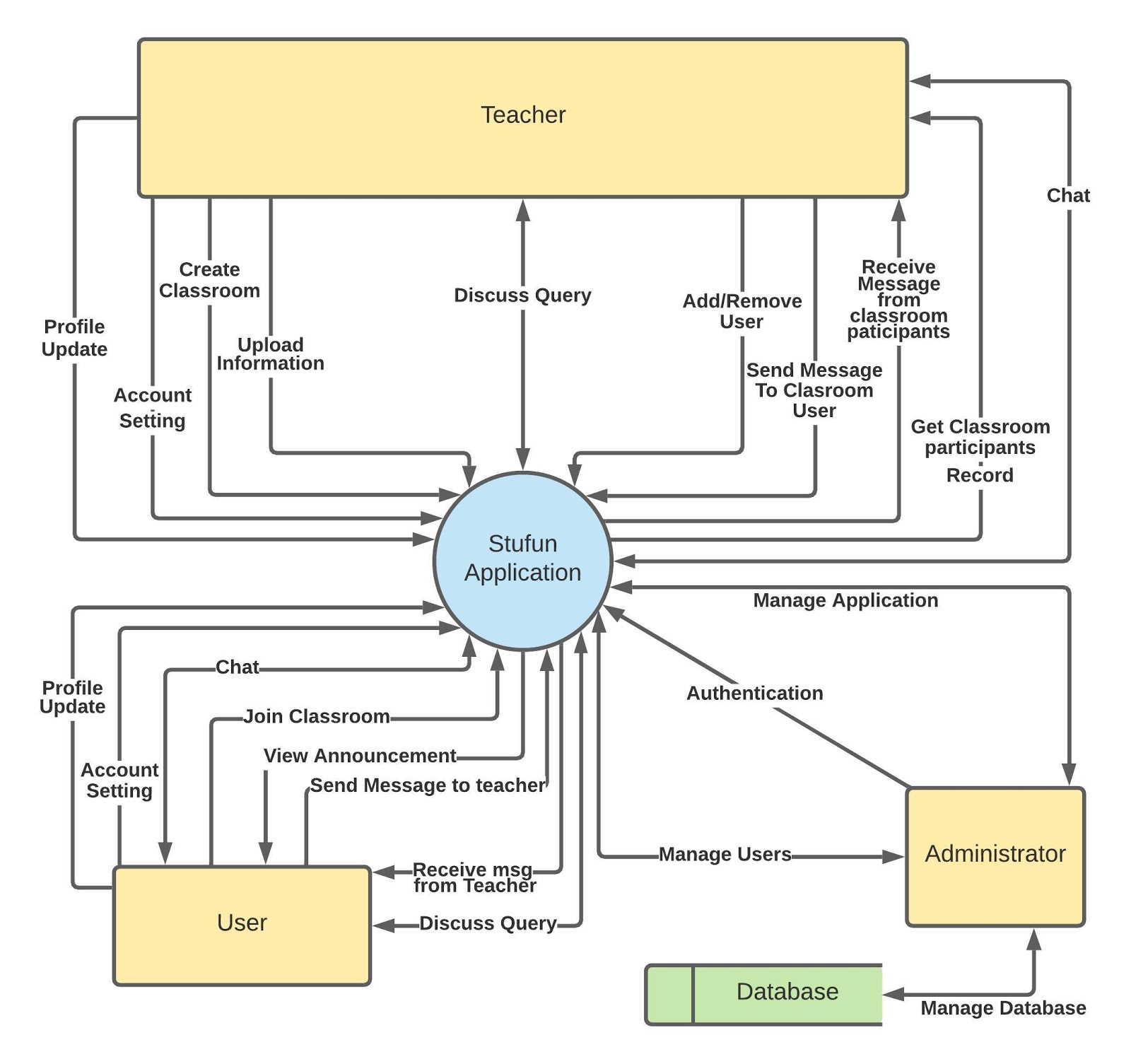
**Functional Requirements**

* + Users for category 1 must be able to register the application through a valid email id. If the user skips this step, the application should close and this email id will be a unique identifier for its account.
  + Category 1 user can send instant messages to anyone on his contact list and would be notified when the message is successfully delivered to the recipient by displaying a tick sign next to the message sent.
  + For attachments, users should be able to send images, documents, and links. Supported image formats (JPG, PNG, GIF).
  + Category 1 users must be able to get information on whether the message sent has been read by the intended recipient. If the recipient reads the message, the “Seen” mark must appear next to the message read otherwise “Delivered” mark will be shown next to msg.
  + Category 2 users will also register to the application like category 1 users.
  + An educational group (Classroom) will be created by a teacher and he will remain the sole admin.
  + To enter an educational group invites will be sent through the StuFun app or email and users will have a name redecided with their admin.
  + Category 2 users can not send instant messages to other members of the Classroom but teachers and users will be addressed in a common discussion panel to the group as a whole.
  + Teachers of a classroom will also be having functions like discussion and make announcements, as well as view classroom student details, notifications of such, will also be managed similar to the likes of schedule reminders and alert

**Non-Functional Requirements**

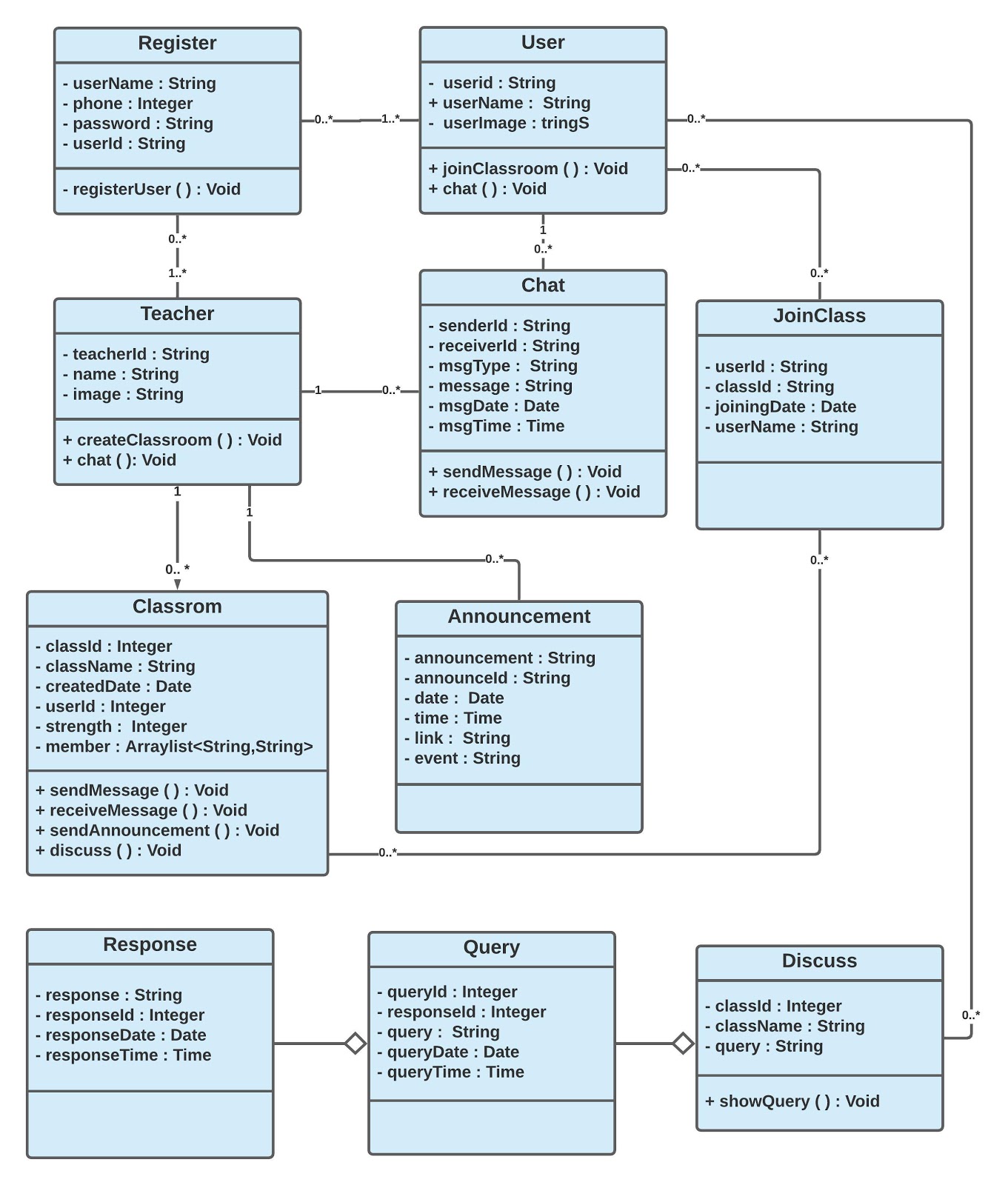
* + **Performance:** It must be able to perform in adverse conditions like; slow internet speed, low memory and RAM on the device, low battery and should provide uninterrupted connections and must have a high data transfer rate.
  + **Scalability:** StuFun should be able to provide instant messaging services to 1 thousand users at any given time and a similar service to 10 educational groups of 100 users each.
  + **Security Requirements and Privacy:** StuFun provides encryption to prevent unauthorized access to a message midway as it will use SSL standard encryption to secure data.
  + **Robustness**: In case a user’s device crashes, a backup of their chat history must be stored on remote database servers to enable recoverability, also they can have an auto backup of their data.
  + **Availability:** The StuFun internal sever has to be available 24 hours of the day.
  + **Maintainability:** Only maintainers will be allowed to connect to the internal servers.
  + **Portability and compatibility:** Users can easily logout from one device to another with their recent chats available through other resources that might require some time. Also, a large number of Android devices will be supported as the app will be light and fast.
  + **Operational Requirements:** The application must work on all mobile and tablet devices. The user interface must be consistent on all devices.

**SOFTWARE DESIGN**

**3.1 DATA FLOW DIAGRAM**

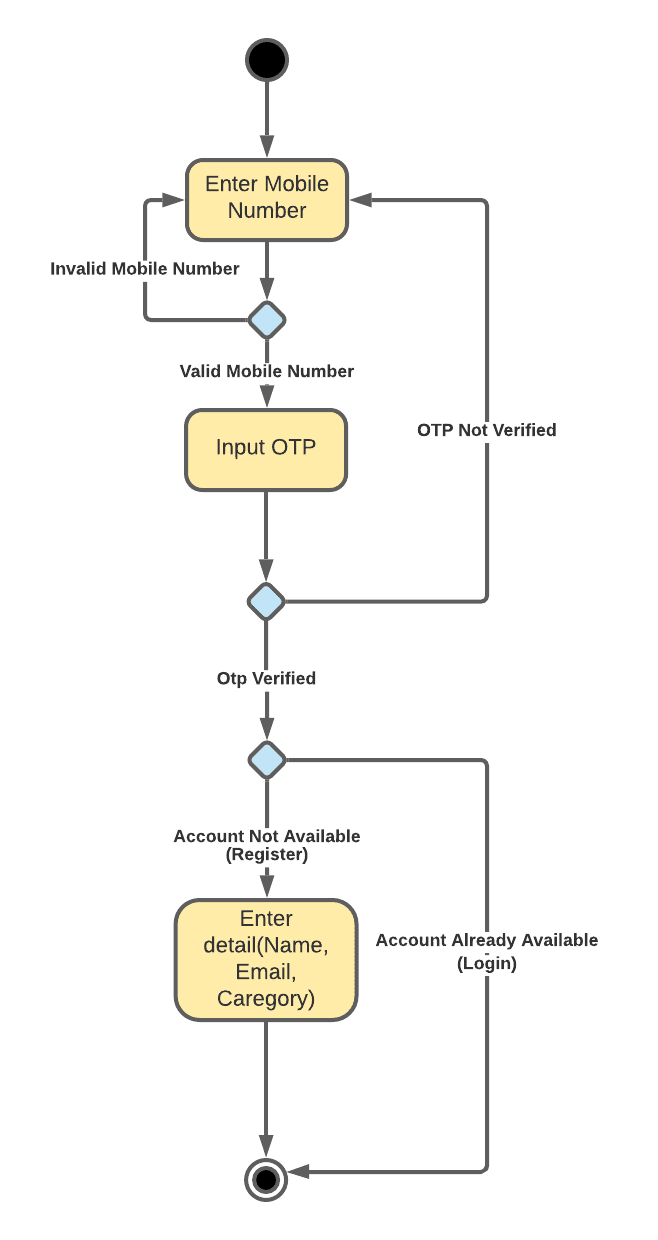
**3.2 UML DIAGRAM**

**3.2.1 CLASS DIAGRAM**

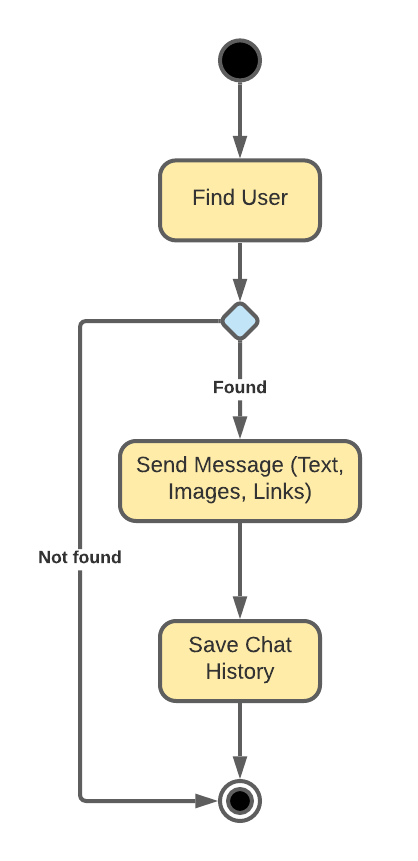


**3.2.2 ACTIVITY DIAGRAM**

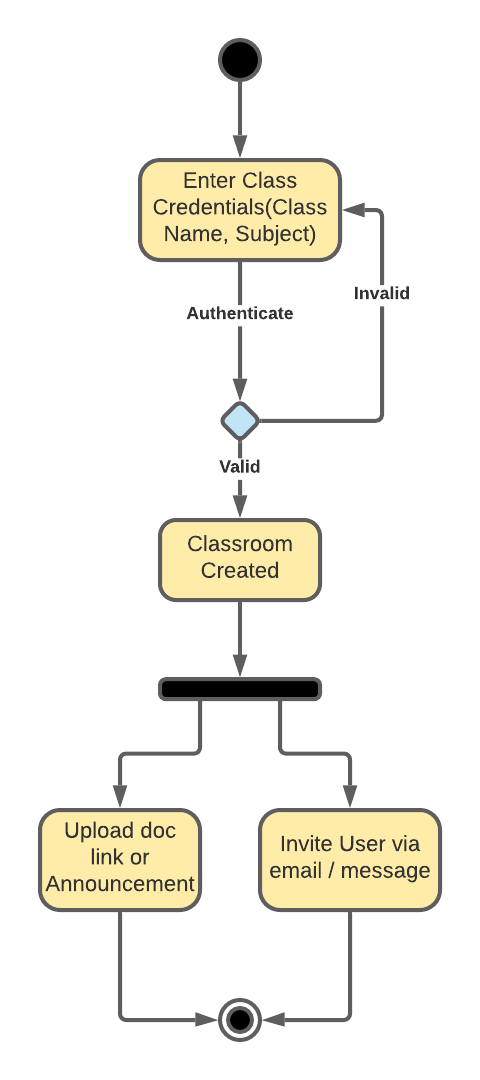
**Registration/Login**



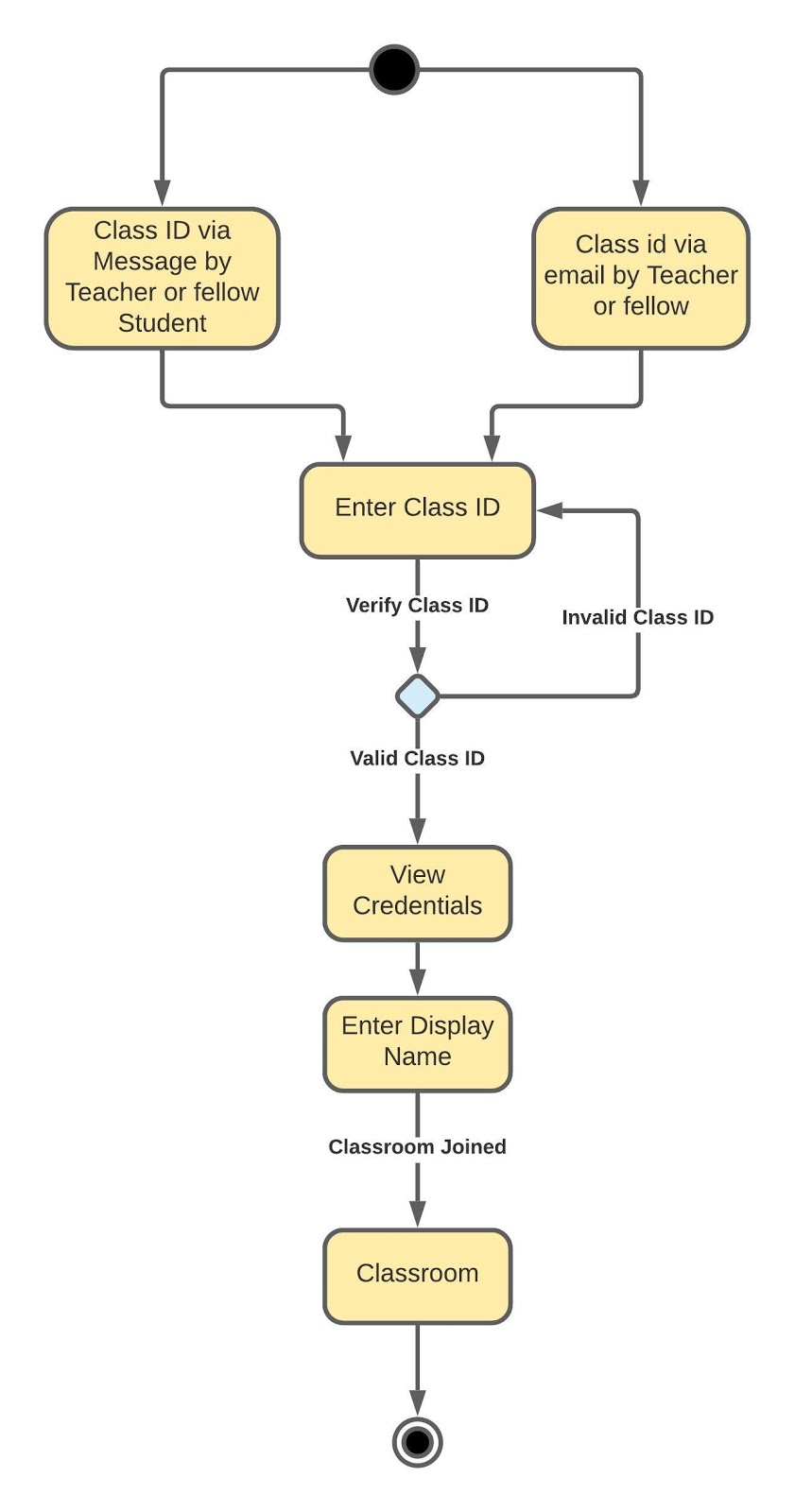
**Personal Chat**



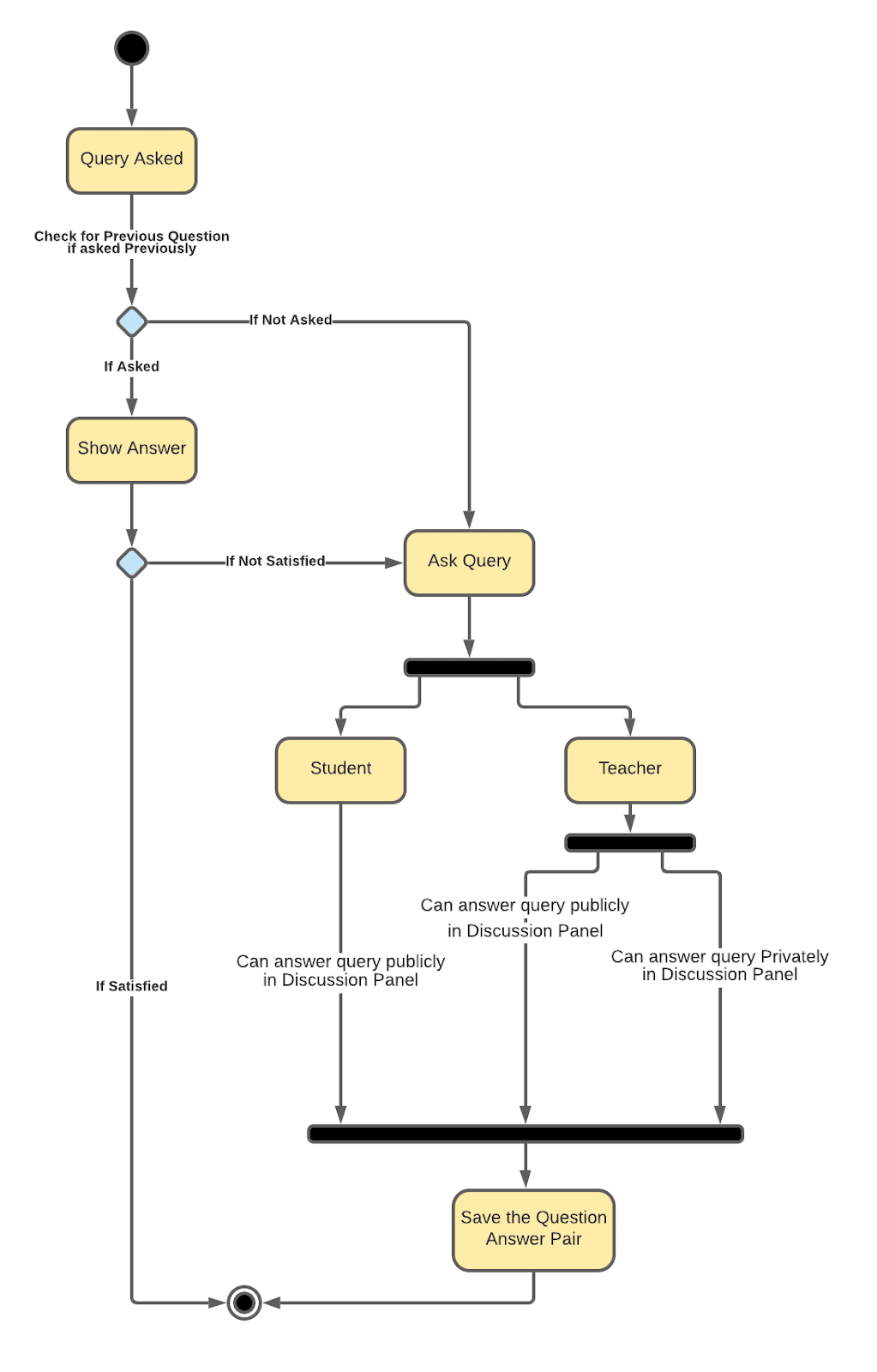
**Create Classroom**



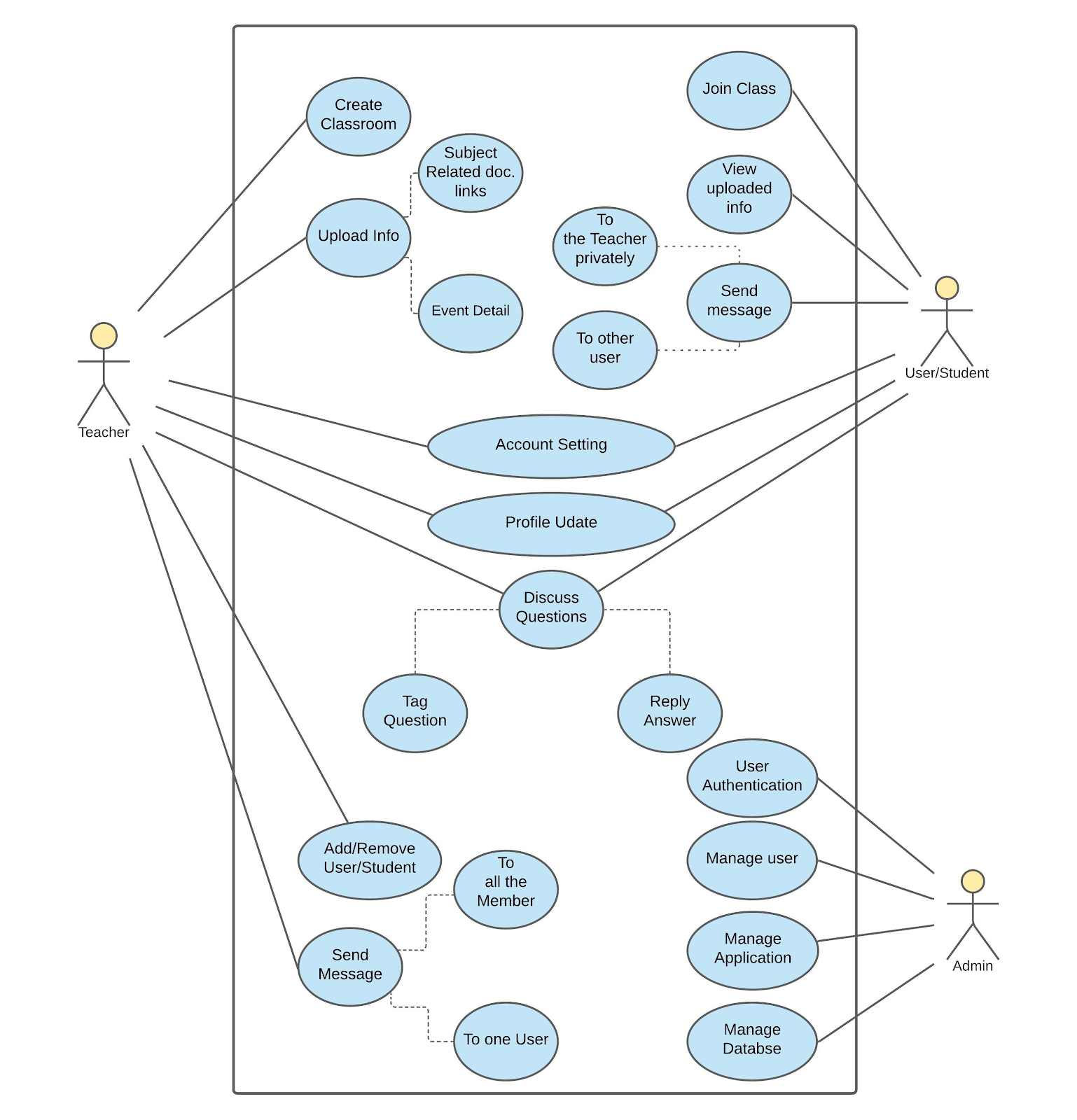
**Join Classroom**

****

**Discussion**

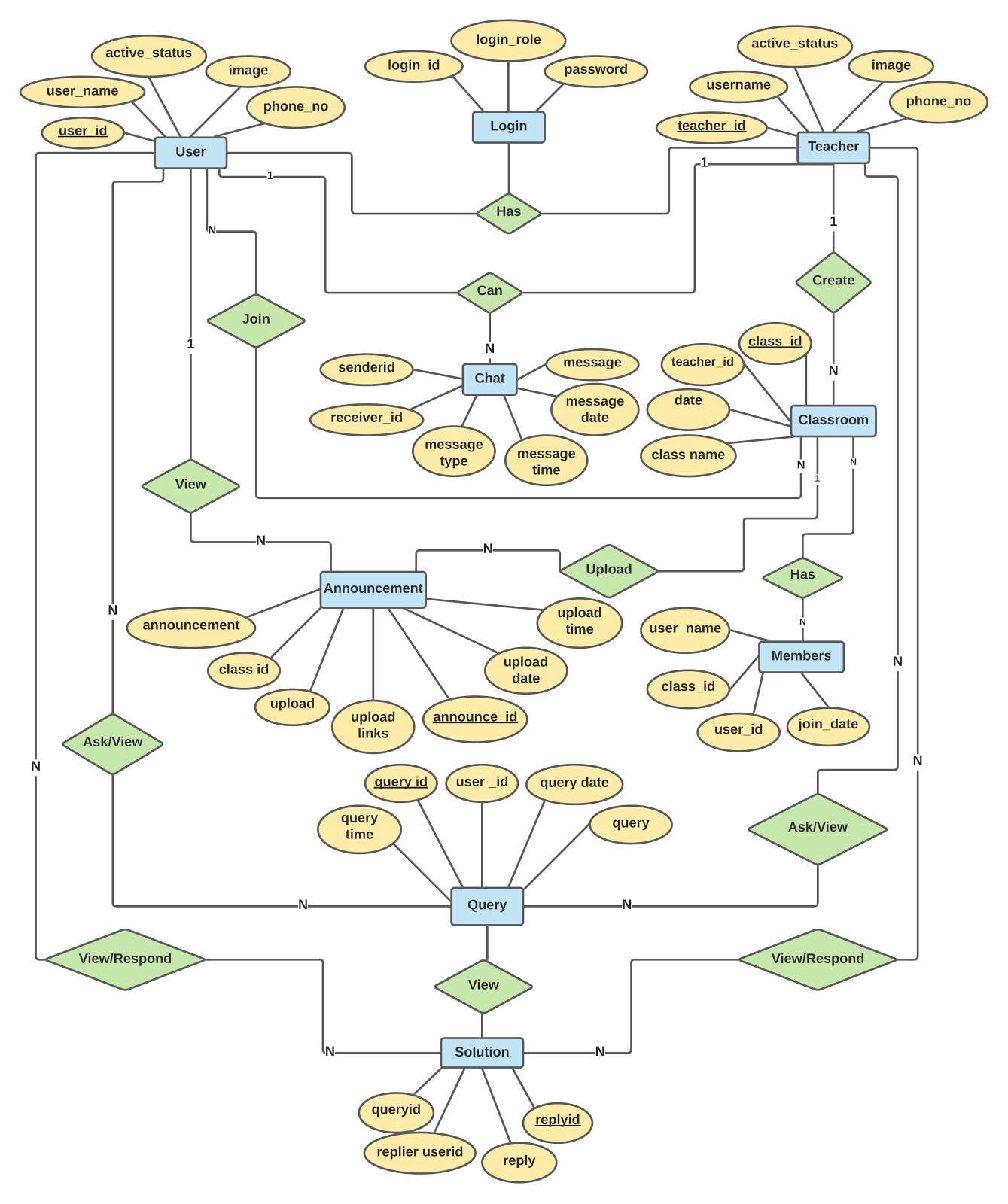
****

**3.2.3 USE CASE DIAGRAM**



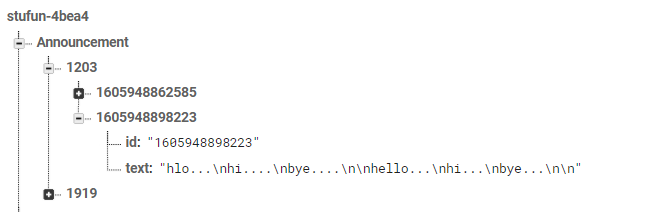
**3.3 DATABASE DESIGN**

**ER DIAGRAM**

****

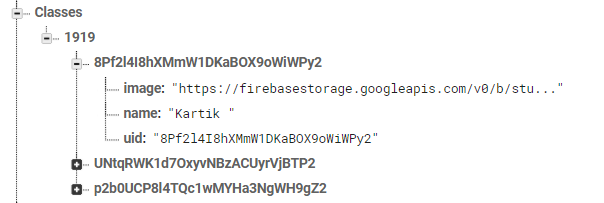
**DATABASE DESCRIPTION**

1. **Announcement**

****

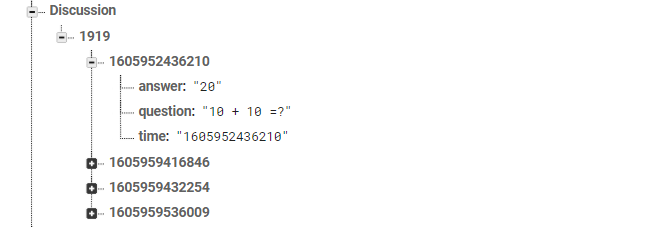
|  |  |
| --- | --- |
| Announcement table | |
| id | String |
| text | String |

1. **Classes**

****

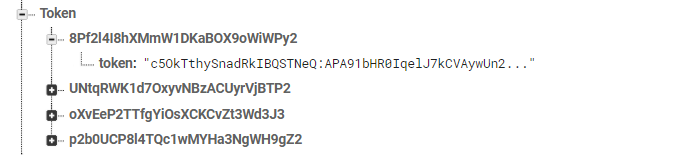
|  |  |
| --- | --- |
| Classes | |
| image | String |
| name | String |
| uid | String |

1. **Discussion**

****

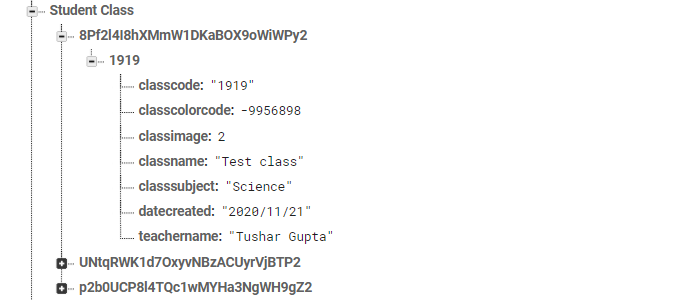
|  |  |
| --- | --- |
| Discussion | |
| answer | String |
| question | String |
| time | String |

1. **Token**

****

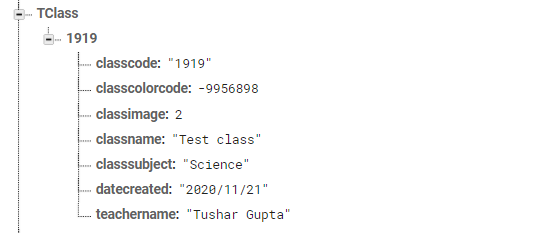
|  |  |
| --- | --- |
| Token | |
| token | String |

1. **StudentClass**

****

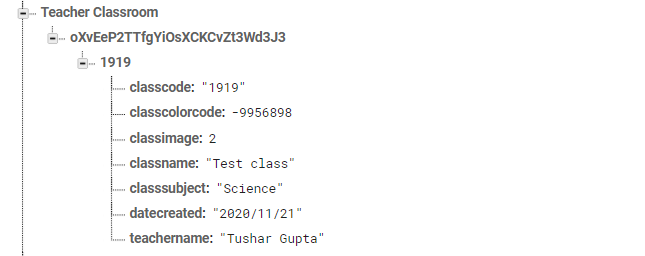
|  |  |
| --- | --- |
| Student Class | |
| classcode | String |
| classcolorcode | int |
| classimage | int |
| classname | String |
| classsubject | String |
| datecreated | String |
| teachername | String |

1. **Tclass**

****

|  |  |
| --- | --- |
| Tclass | |
| classcode | String |
| classcolorcode | int |
| classimage | int |
| classname | String |
| classsubject | String |
| datecreated | String |
| teachername | String |

1. **TeacherClass**

****

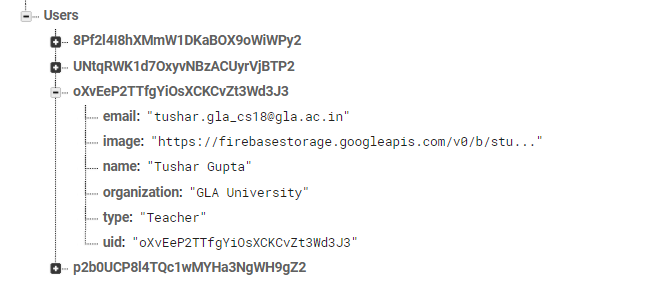
|  |  |
| --- | --- |
| Teacher Class | |
| classcode | String |
| classcolorcode | int |
| classimage | int |
| classname | String |
| classsubject | String |
| datecreated | String |
| teachername | String |

1. **UserChat**

****

|  |  |
| --- | --- |
| UserChat | |
| clear | String |
| date | String |
| image | String |
| isseen | Boolean |
| message | String |
| receiver | String |
| sender | String |
| type | String |

1. **Users**

****

|  |  |
| --- | --- |
| Users | |
| email | String |
| image | String |
| name | String |
| organization | String |
| type | String |
| uid | String |

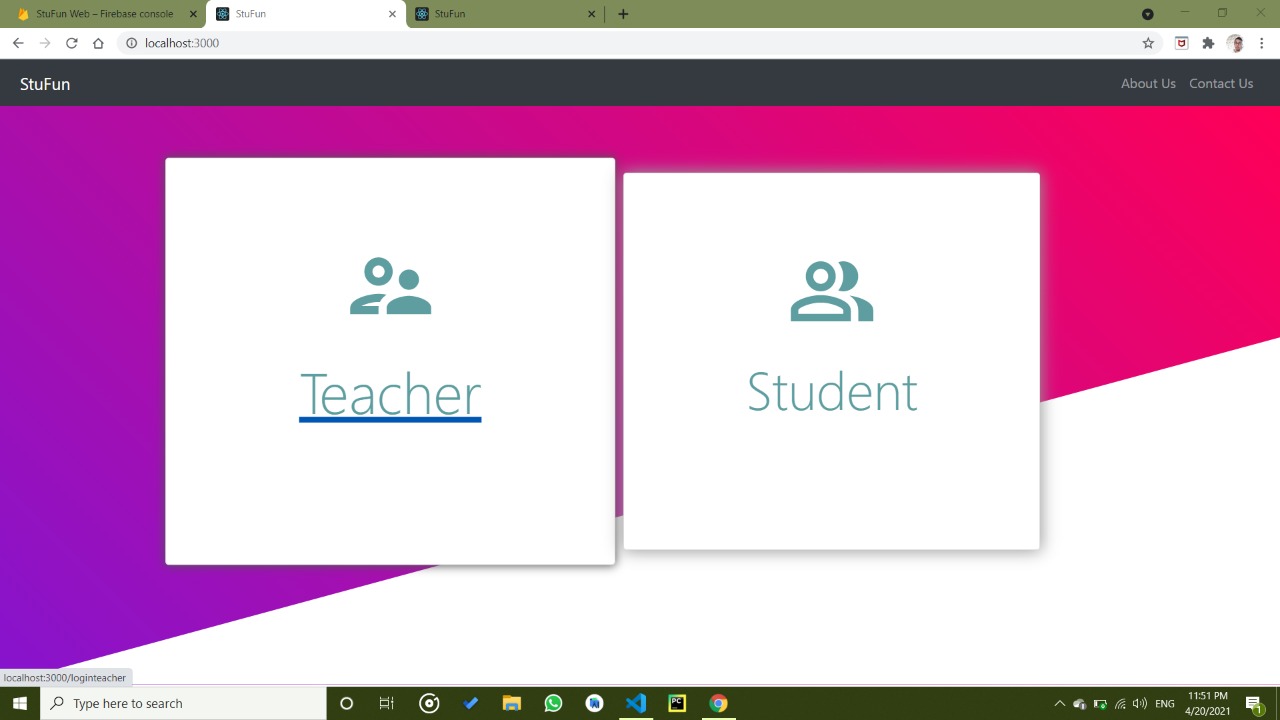
**IMPLEMENTATION AND USER INTERFACE**

**USER AS A TEACHER**

1. **Signup**

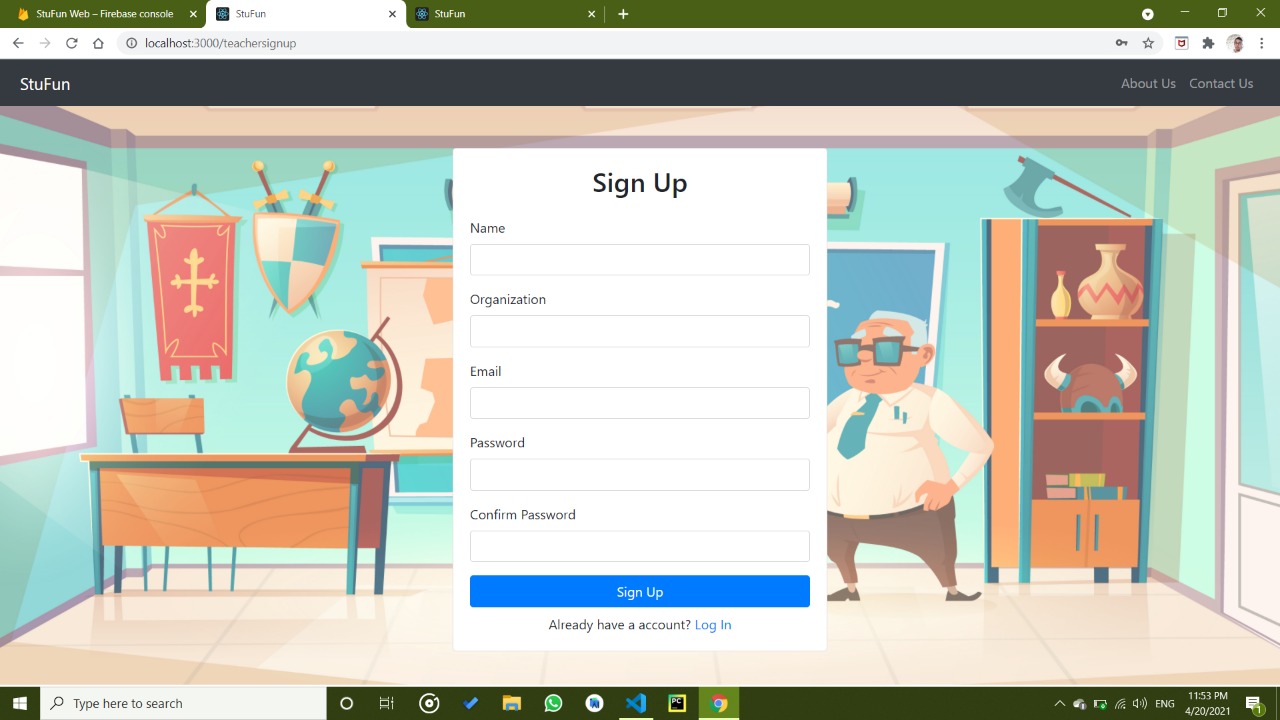
For sign up, we have use firebase authentication system. During signup user will ask for email id and password. After validation it will send a verification code to a user given id. After successful verification user refers to the profile creation activity.

After verification user has to select the account type that is Teacher account for teacher and Student account for the student.

****

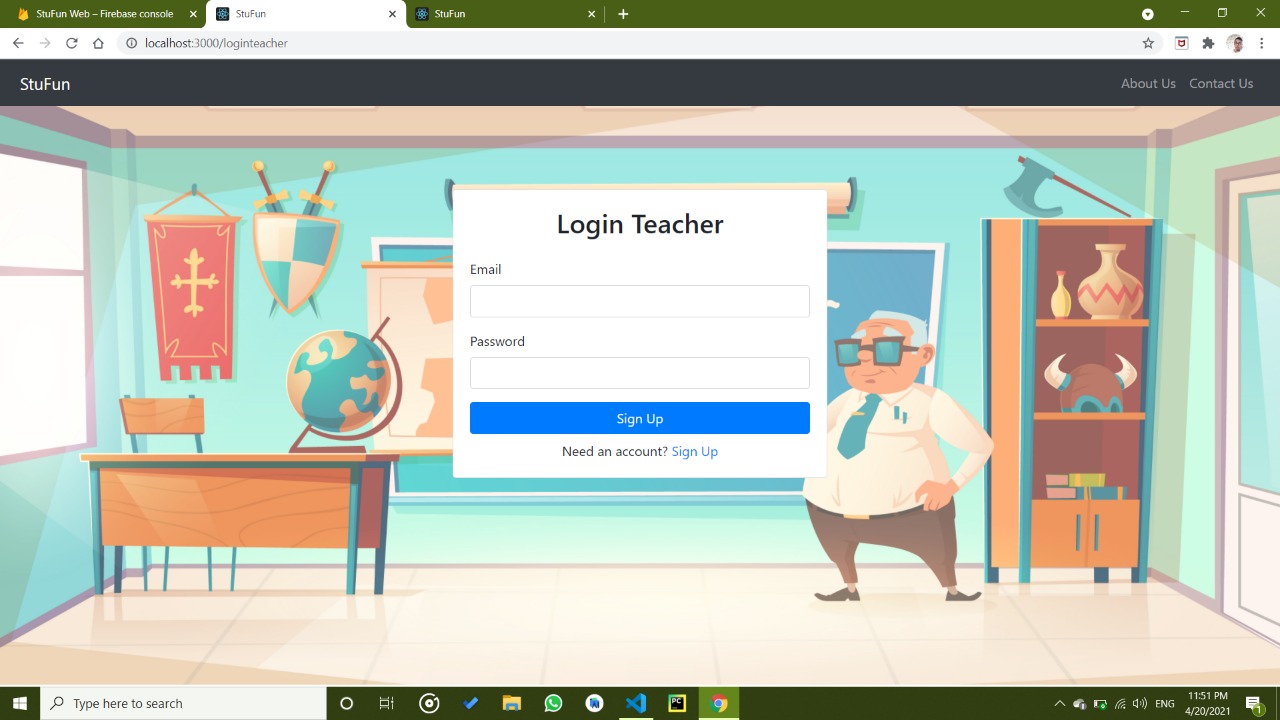
1. **Signup**

The teacher will be asked for an image, name, and Organization name. If any one of the details is not present then it will show an error.

****

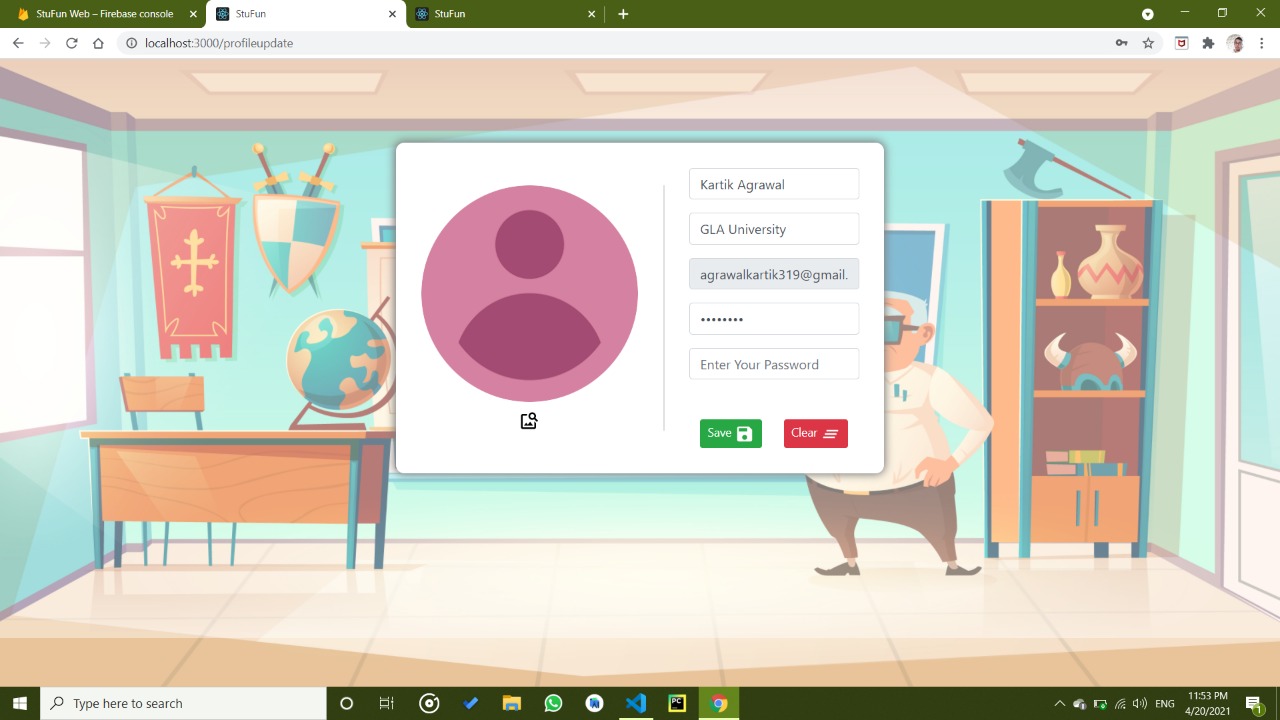
1. **Login Activity**

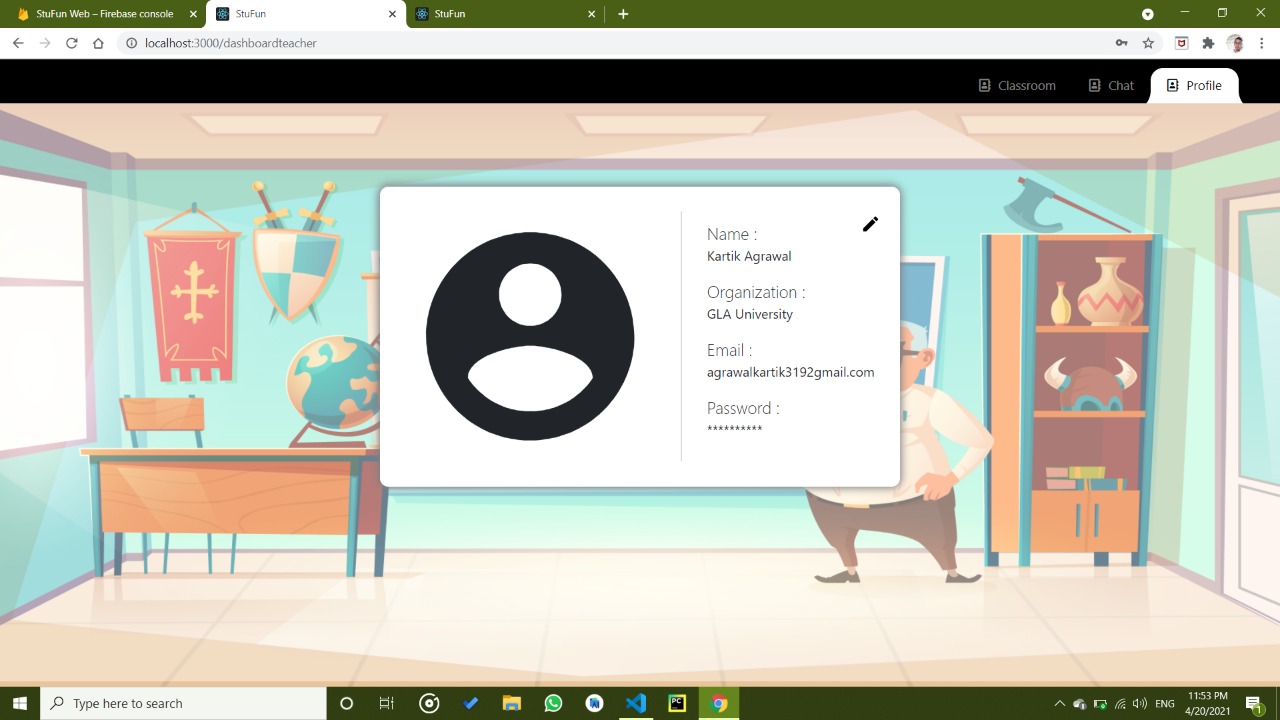
Login activity will ask for email and password. if both the things are matched it get access to the user to the application otherwise it will show a valid error message.

****

1. **Setting Activity**

Setting Activity shows the profile detail and shows the option to change the profile details. Edit Profile Activity allow user to change details such as name and organization.

****

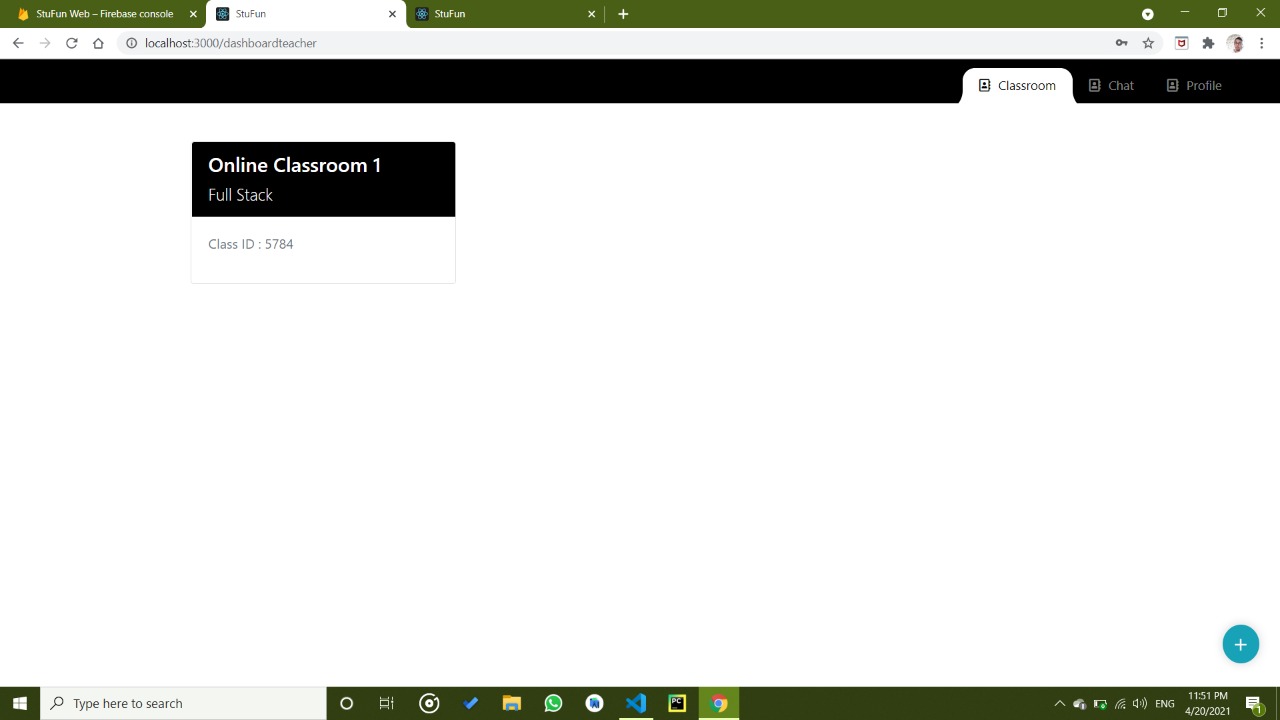
****

1. **Teacher Home Page**

On the home page, the teacher can view a list of all the class that are created by him.

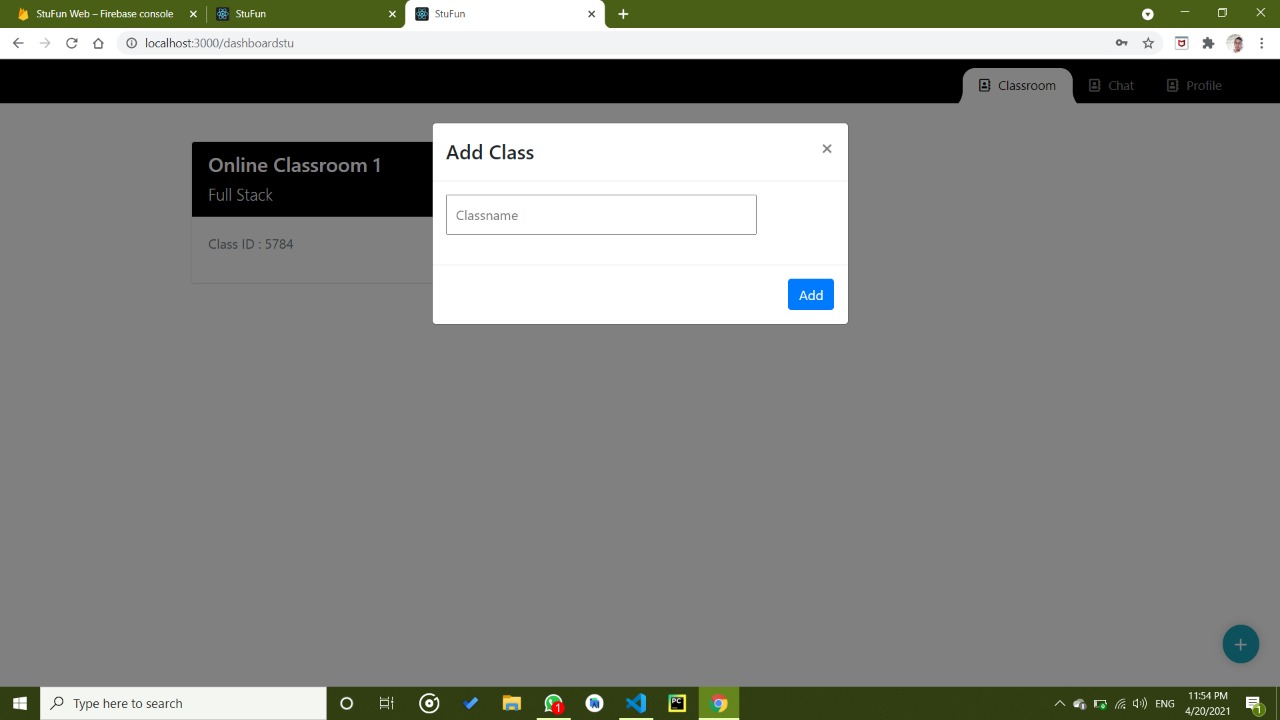
He can also create a new class by simply click the floating action button situated at the bottom-right corner.

We have termed this homepage as a dashboard simply because all the necessary operations can be reached from here.



1. **Create Classroom**

Teacher has to fill some necessary details such as Class name and Class room Subject to create a class room

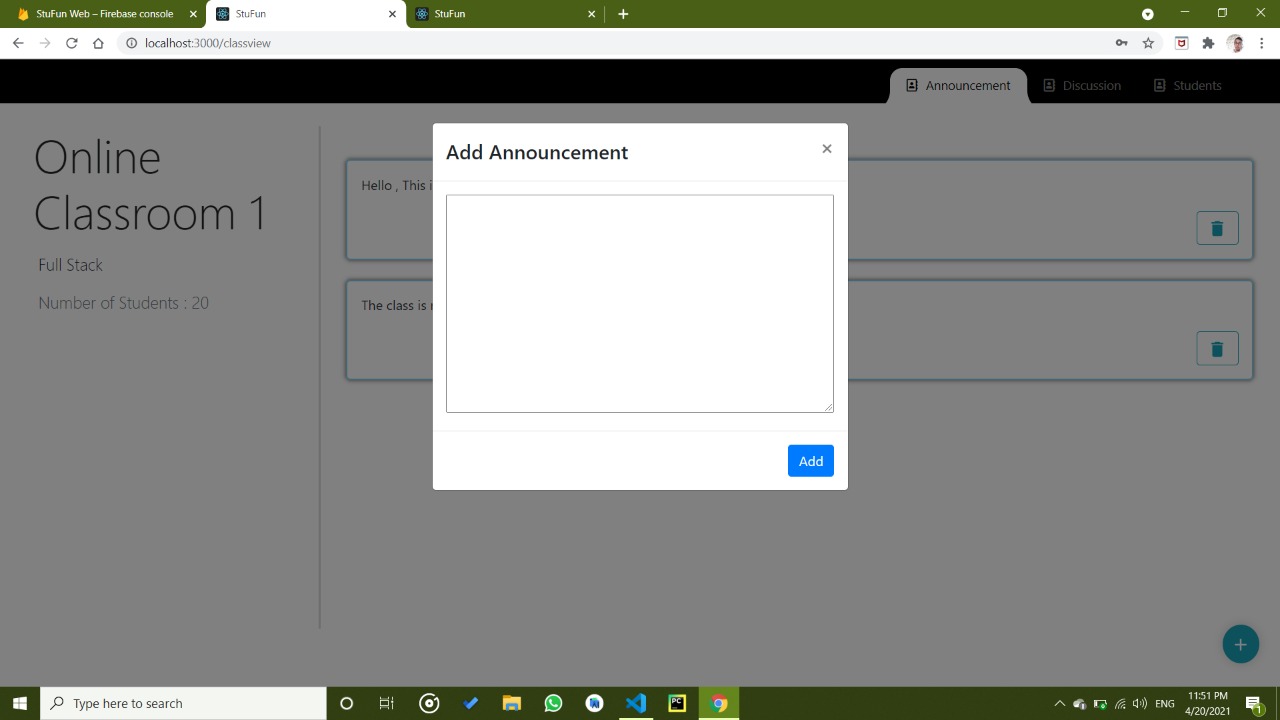


After validation, the teacher will receive a unique class code which is further used by the student to join the classroom. On the same activity, he has the option to send an invitation to join the class via email and SMS using external intent.

**7. Make Announcement**

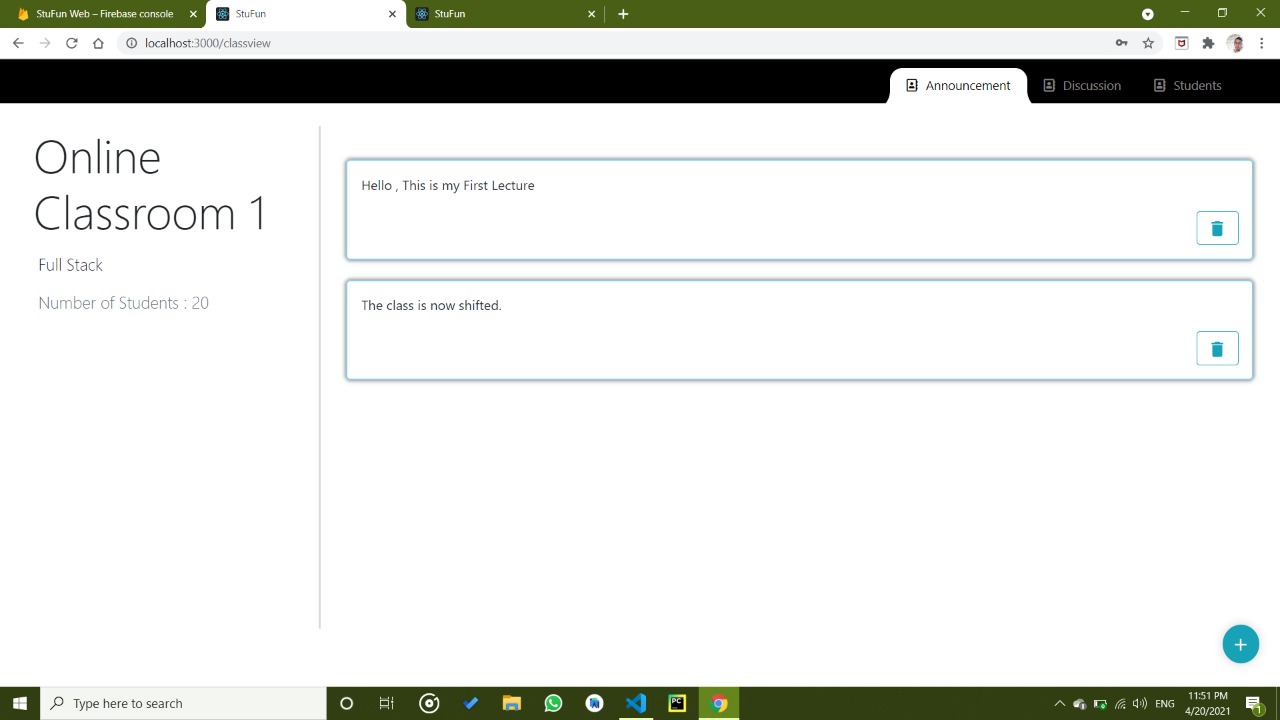
Teacher can easily add as many as announcement as he wants to add to a particular classroom.

After click add announcement the announcement gets added to the announcement recycler view of that particular classroom.

****

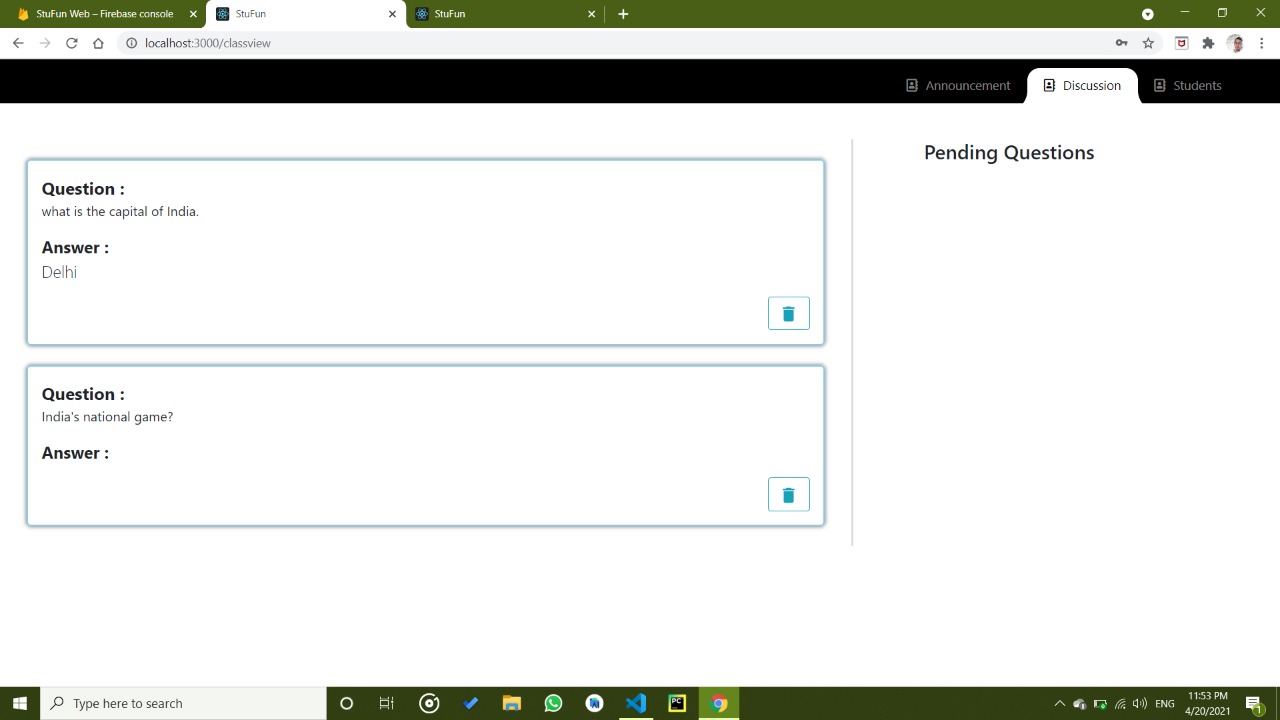
**7.1 View Announcement**

In this fragment, the teacher can view all the announcements which are made by him for the students. He can also delete an announcement by just simply right-swipe the particular announcement which is want to delete.

****

**8. Discussion Activity**

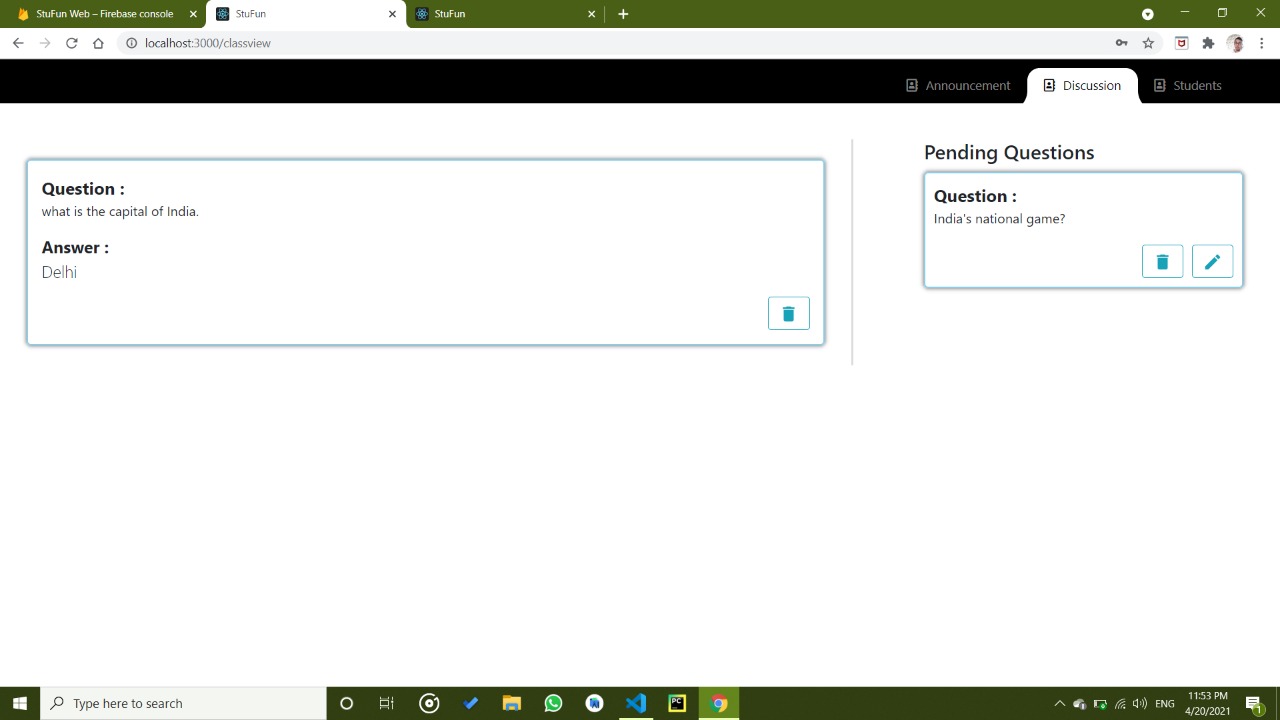
In the discussion fragment, the teacher can view all the previously asked queries with their answer as well as a pending query button.

****

**8.1 Pending Discussions**

Pending question activity contains a list of all the queries that are asked by the student and does not been answered by the teacher yet.

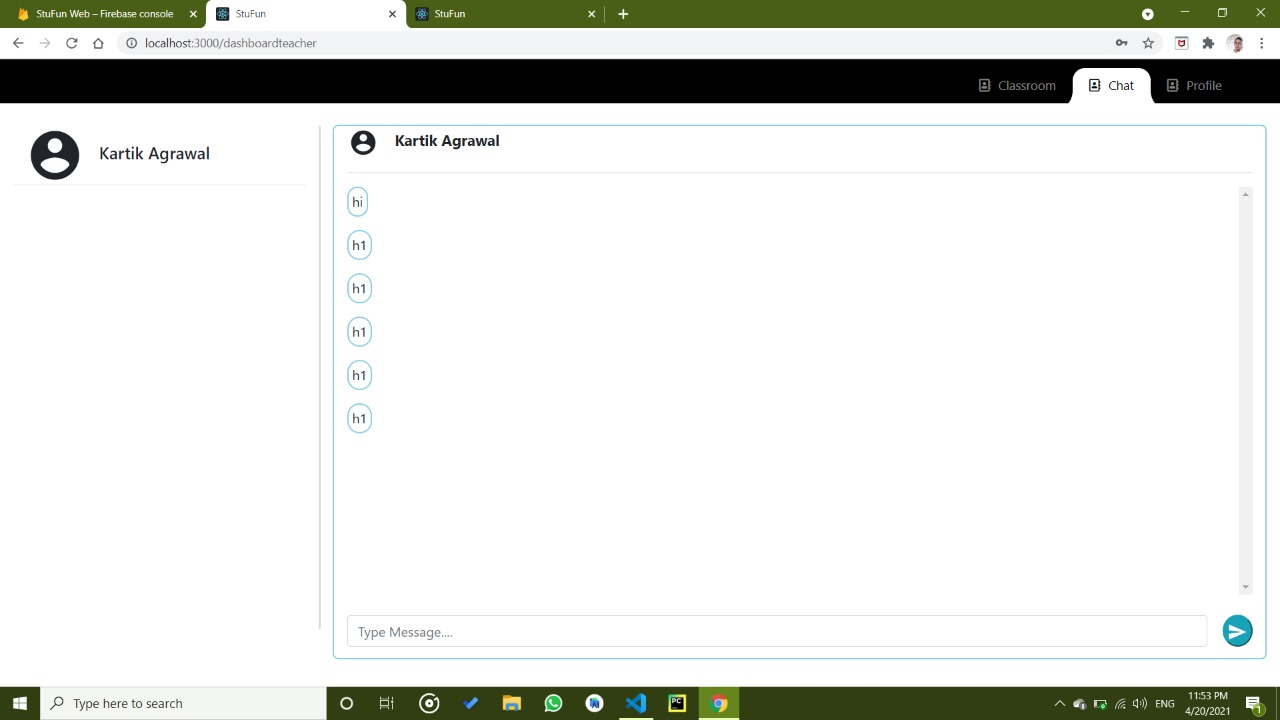
Here teacher has 2 options, either he can reply and post the query or he can delete the query.

****

**9. Chat Activity**

User (Teacher) can view a list of all the user with whom he has previously some chat.

Teacher can detail of all the student. Teacher can block the student from that particular class and also message him directly from here.



User can send a text or image message to another user.

At the bootom of the last message a “seen” mark indicates wheather a receiver has seen a message or not.

Top menu bar contains 2 option –

1. Profile – From where he can view the detail of receiver.
2. Clear chat – From where user can clear the chat.

**USER AS A STUDENT**

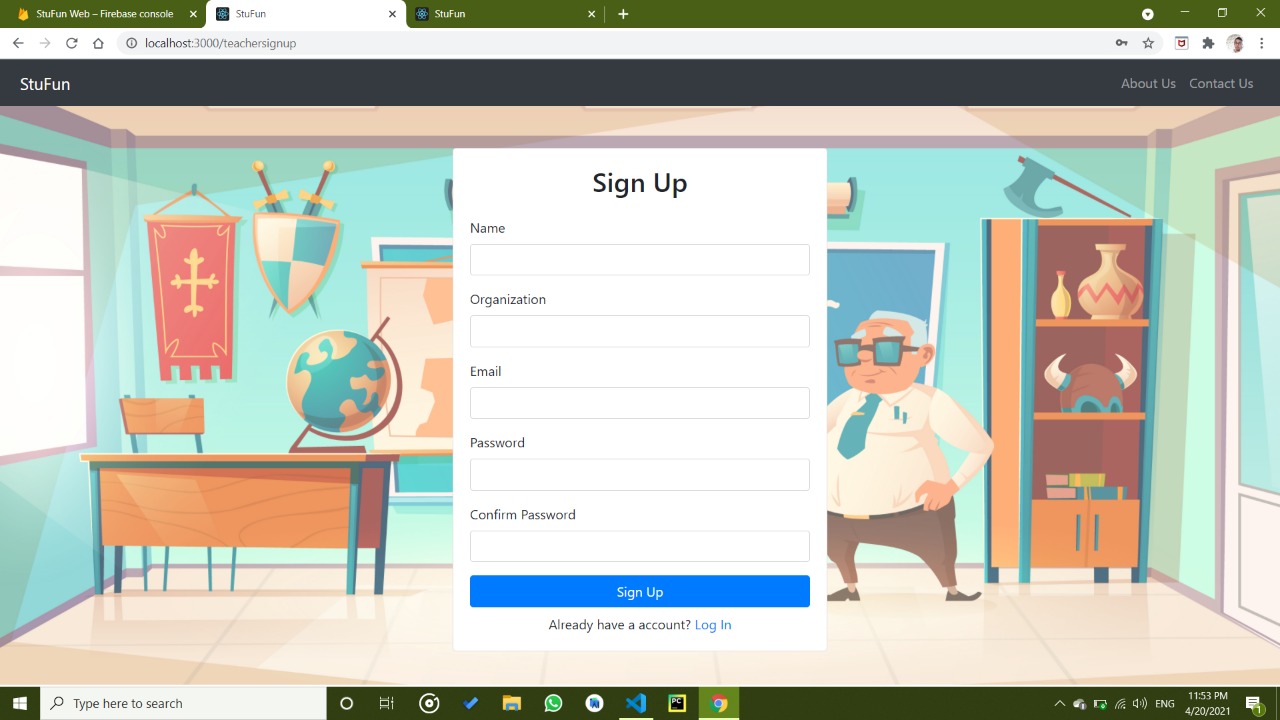
**0. Dashboard**

The dashboard is common to all users and is a navigation bar in the top right corner to access the basic three functionalities.

****

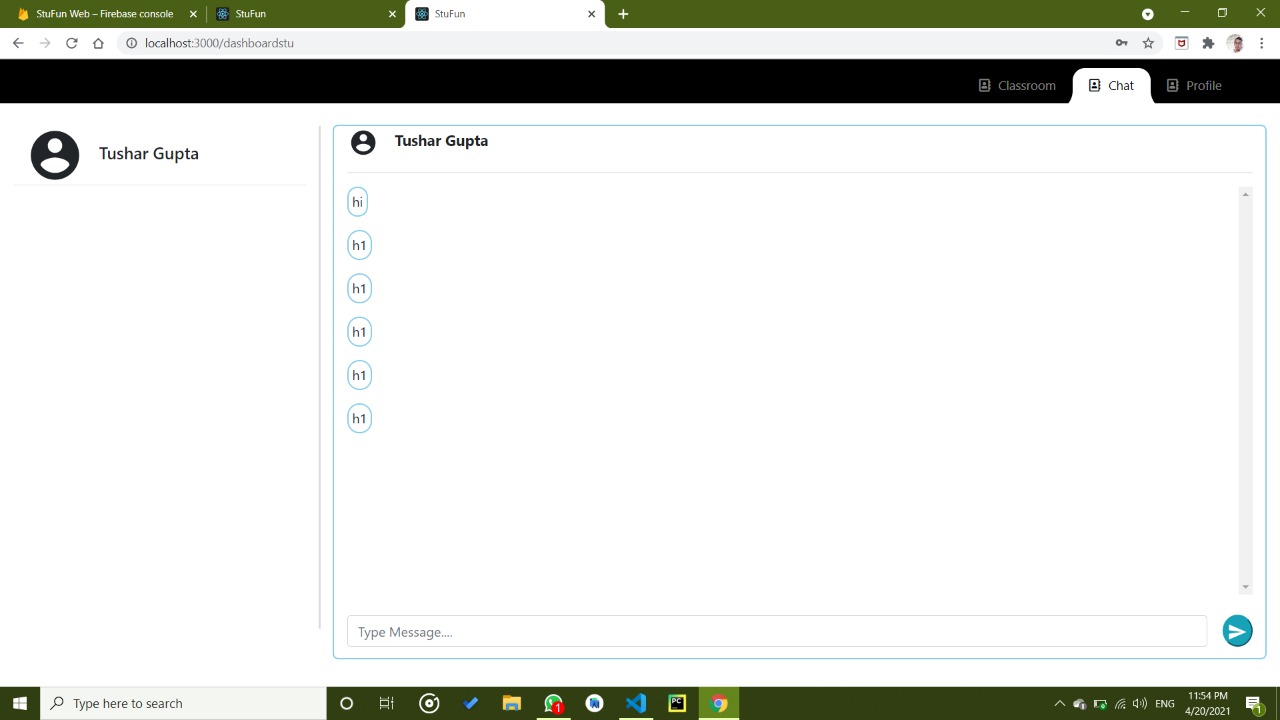
1. **Signup**

Sign up procedure for student is same as teacher except profile detail entry. During profile creation it will asked for branch and couse also along with name, image and organization.If the details are not valid then it will shows an error.

****

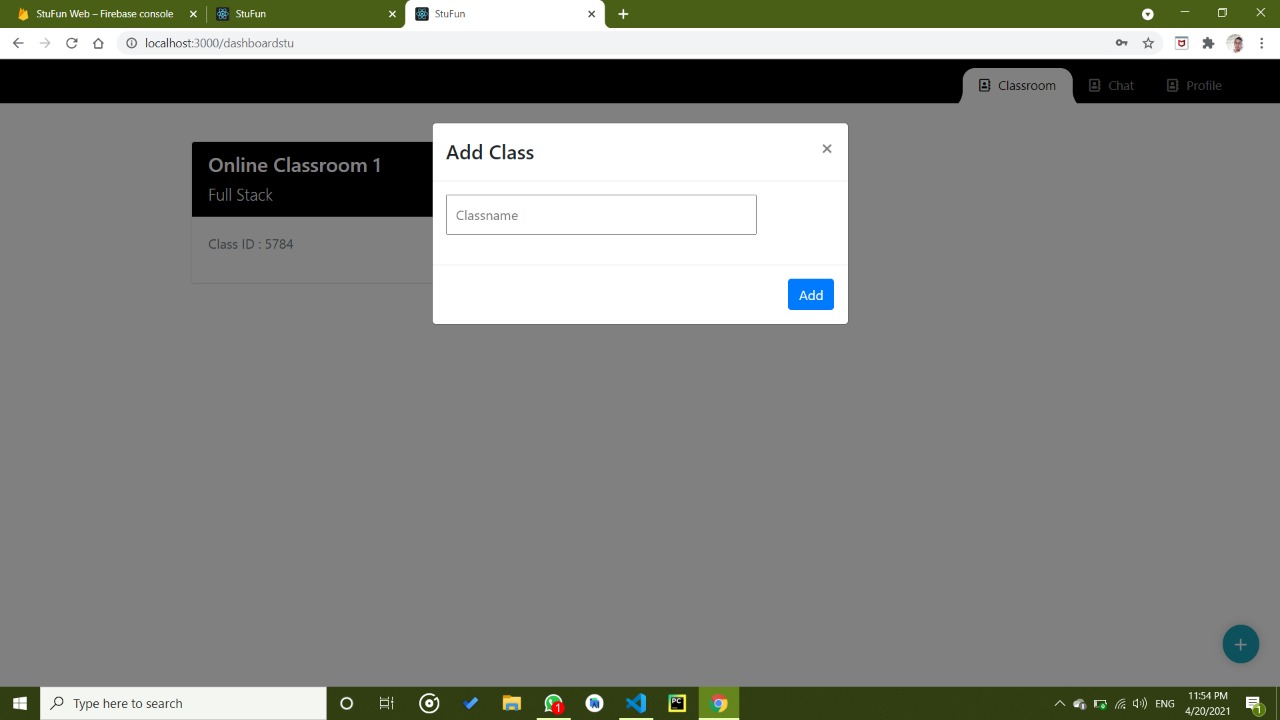
1. **Student home**

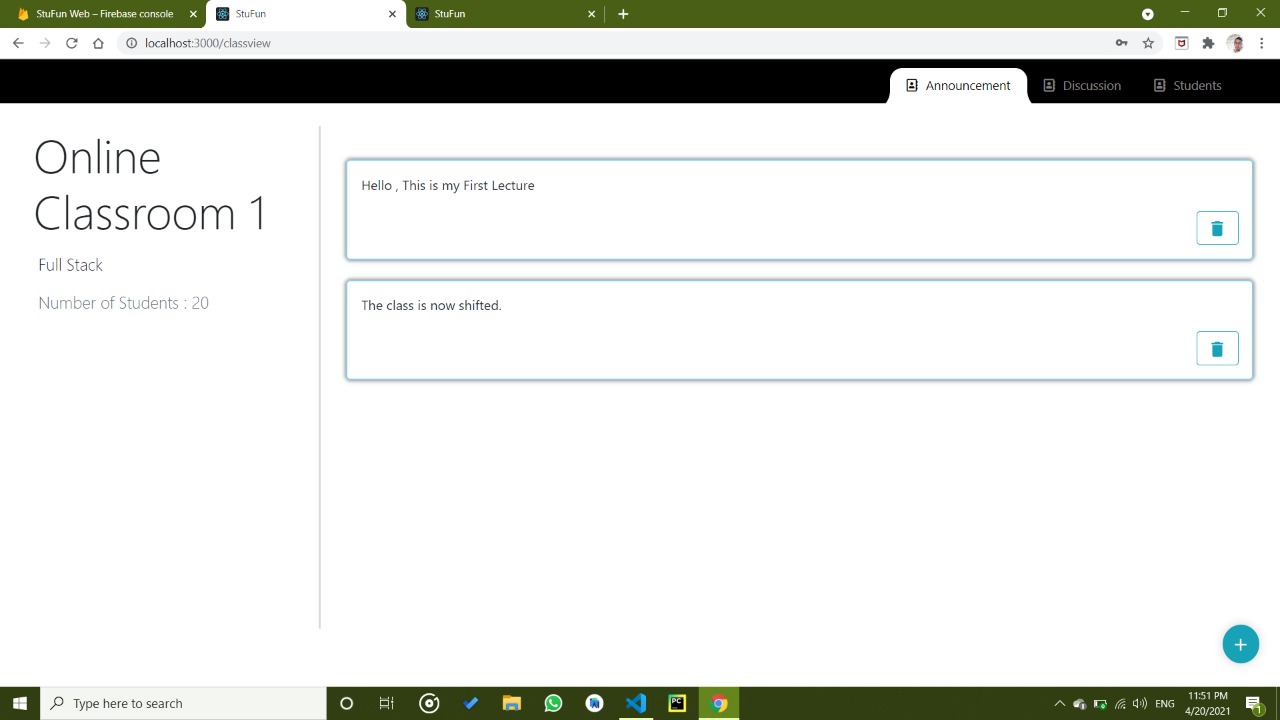
On the student home page, student can view a list of joined classroom.he can also unenroll from the particular class as well as join new class.

****

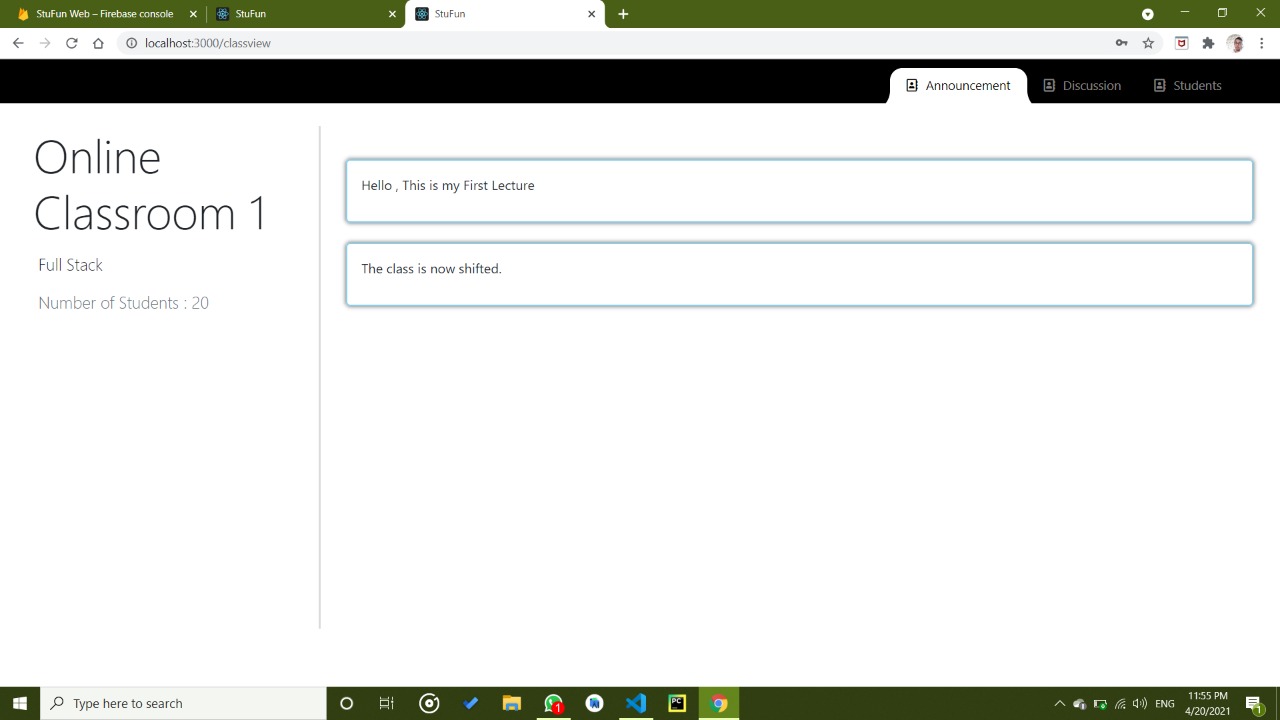
1. **Join Class**

Student can join a classroom by entering a 4 digit unique code given to him by the teacher,

****

****

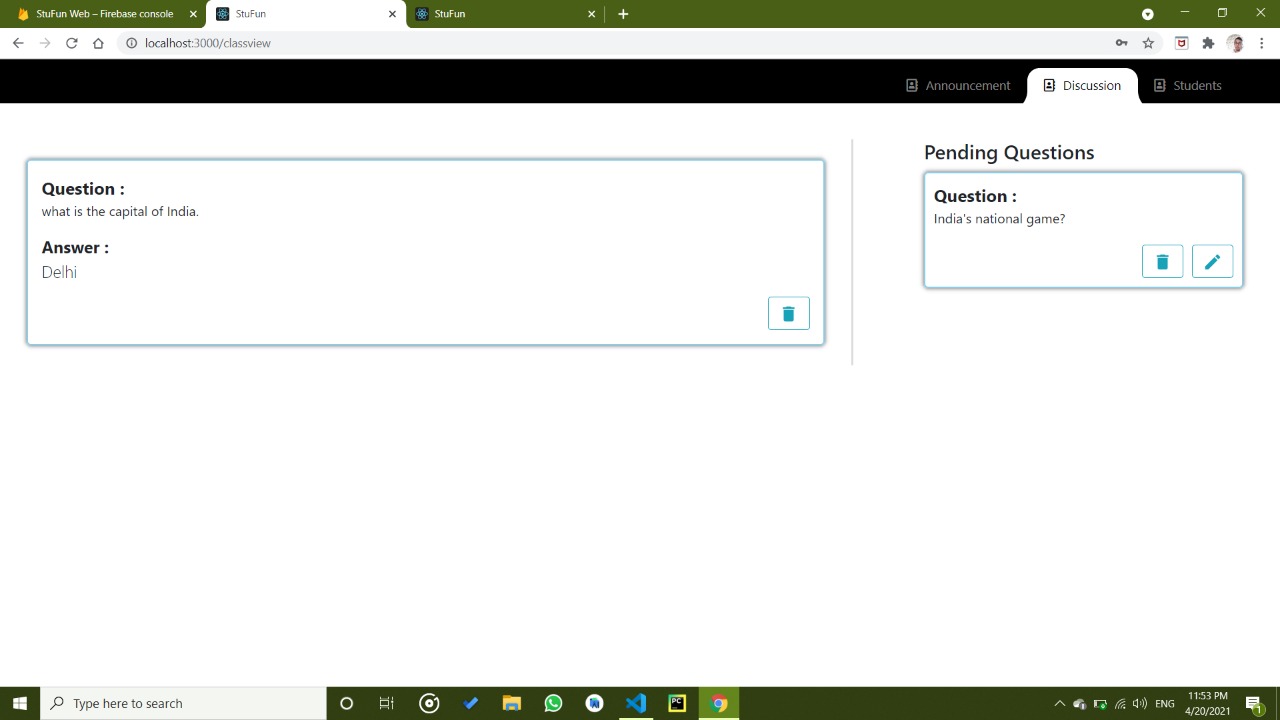
1. **Announcement Activity**

****

Student can view a list of all the announcements made by teacher which keeps him updated.

1. **Discussion Activity**

Student view a list a all the previously asked query along with its answer.

****

Student can ask a query and all his unanswered query will be stored in pending question datalist until it is replied or deleted by the user.

1. **Chat Activity**

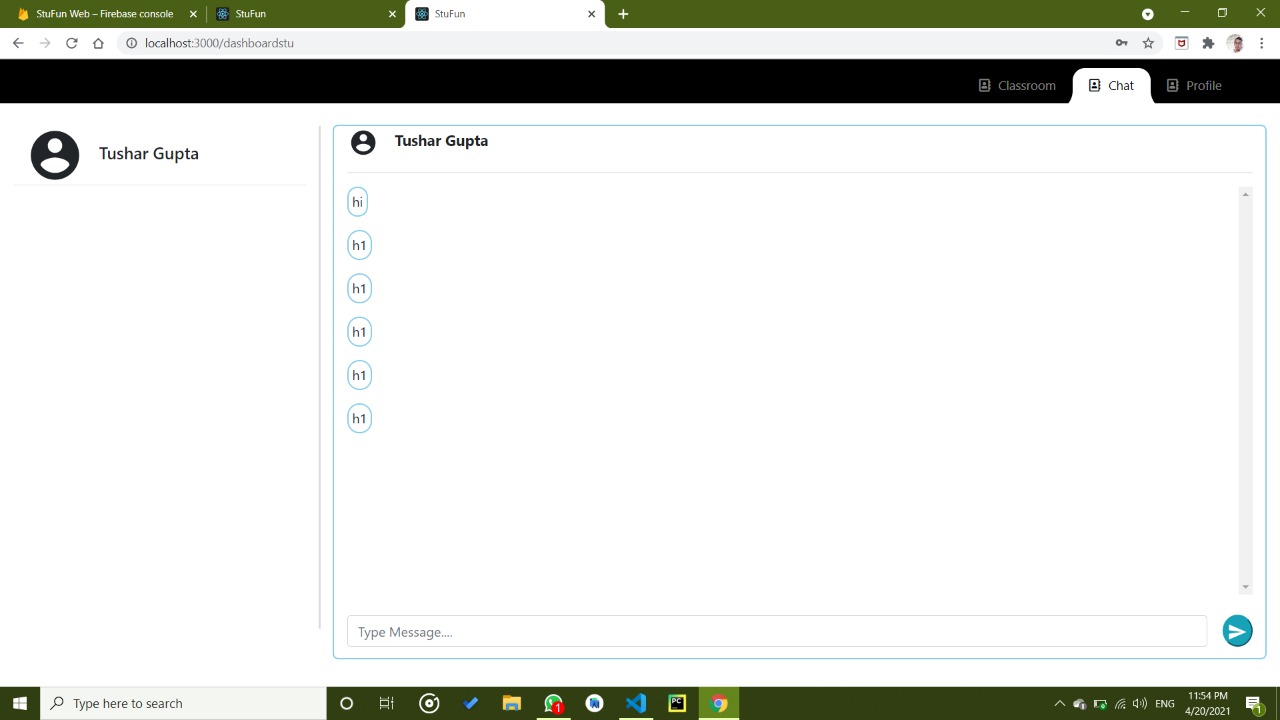
User (Student) can view a list of all the user with whom he has previously some chat.

User can send a text or image message to another user.

At the bottom of the last message a “seen” mark indicates whether a receiver has seen a message or not.

Top menu bar contains 2 option –

1. Profile – From where he can view the detail of receiver.
2. Clear chat – From where user can clear the chat.

****

# **REFERENCES**

* + - * <https://en.wikipedia.org/wiki/Android>
      * <https://en.wikipedia.org>
      * [www.coursera.in](http://www.coursera.in)
      * <https://stackoverflow.com>
      * [www.firebase.in](http://www.firebase.in)