PANDEMIC MANAGEMENT SYSTEM

UCS503 Software Engineering Project Report

End Semester Evaluation

Submitted By:

(101803574)	**Mohit Goyal**
(101803577)	**Purujit**
(101803716)	**Chirag Mahajan**
(101803718)	**Rachit Anand Srivastava**

B.E. Third Year, COE

Submitted To:
Ms. Deepali Bhagat

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
TIET, Patiala

November 2020

TABLE OF CONTENTS

PROJECT OVERVIEW

The objective of the project is to create awareness amongst the people about an ongoing pandemic/epidemic that had already occurred by creating a scalable chat group just like a subreddit with both text as well as voice channels just like discord.

a. This project contains various sub-channels for a particular pandemic and each sub-channel addresses a particular topic.
b. Each sub-channel contains many threads and each thread is nothing but a post in a sub-channel just like a post in a subreddit which also contains voice channels.
c. Participants can interact with threads i.e they can comment in these threads, share any crucial information or ask questions.
d. The moderators can conduct an AMA session where people can ask questions and then someone can answer them.

This project contains various sub-channels for a particular pandemic and each sub-channel addresses a particular topic. Each sub-channel contains many threads and each thread is nothing but a post in a sub-channel just like a post in a subreddit.
Also these sub-channels mimic the properties of a discord channel i.e. they also have voice channels along with text channels and a sub-channel may also contain bots for assistance of participants or authors.
An admin creates a channel for a particular pandemic/epidemic, then a moderator creates and manages sub-channels within this channel. An author is nothing but a participant that has created a particular thread and has full control over the thread. But the sub-channels are watched over by the moderator. Now the participants can interact with threads i.e they can comment in these threads, share any crucial information or ask questions. For eg. The moderators can conduct an AMA session where people can ask questions and then someone can answer them.
Also the threads can be deleted at any time by the autor, moderator or admin as per their wish if any of the policies are not followed for that thread. After the pandemic comes to an end only useful information can be kept in the channel and other data can be deleted.

Software Requirements Specification

for

Pandemic Management System

Version 1.0

Prepared by

Group Number: *2*

Mohit Goyal	101803574	mgoyal_be18@thapar.edu
Purujit	101803577	pgupta_be18@thapar.edu
Chirag Mahajan	101803716	cmahajan_be18@thapar.edu
Rachit Anand Srivastava	101803718	rsrivastava_be18@thapar.edu

Instructor:	*Dr. Vinod Bhalla*
Course:	Software Engineering
Lab Section:	*COE 26*
Teaching Assistant:	*Ms. Deepali Bhagat*
Date:	22-09-2020

CONTENTS

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Mohit Goyal Purujit Chirag Mahajan Rachit Anand Srivastava	Pandemic Management System	September, 2020

1 Introduction

The purpose of this document is to provide a detailed overview and specifications of the project Pandemic Management System.

1.1 Document Purpose

This document aims to provide the software and other requirements of the Pandemic Management System. It is a discussion platform which is easily accessible to the public. In this platform people will be able to discuss the different pandemic and the problems caused in the world because of them. This will help to come up with different solutions to eradicate the problems.
In addition, it explains the design, significance, implementation and objective of the project.

1.2 Product Scope

The objective of the project is to create awareness amongst the people about an ongoing pandemic/epidemic that had already occurred by creating a scalable chat group just like a subreddit with both text as well as voice channels just like discord.

e. This project contains various sub-channels for a particular pandemic and each sub-channel addresses a particular topic.
f. Each sub-channel contains many threads and each thread is nothing but a post in a sub-channel just like a post in a subreddit which also contains voice channels.
g. Participants can interact with threads i.e they can comment in these threads, share any crucial information or ask questions.
h. The moderators can conduct an AMA session where people can ask questions and then someone can answer them.

1.3 Intended Audience and Document Overview

The intended audience of this SRS consists of:

a. The Professor
b. Developers
c. Client

The remainder of this document is organized as follows: Section 2 contains a general description of the pandemic management website. Section 3 identifies the specific functional requirement of the external interfaces and performance requirements of the pandemic management website.

1.4 Definitions, Acronyms and Abbreviations

a. PMS - Pandemic Management System
b. PWA - Progressive Web Applications

1.5 Document Conventions

a. This document follows the IEEE formatting requirements.
b. Arial fonts of size 11 and 12 have been used throughout the document.
c. Document text is single spaced and maintains the 1” margins.
d. For subsection titles, we use Arial font size 14 throughout the document.
e. For the section title, we use Arial font size 18 throughout the document.

1.6 References and Acknowledgments

1.6.1 References

- https://lms.thapar.edu/moodle/mod/folder/view.php?id=4786
- https://sparxsystems.com/enterprise_architect_user_guide/14.0/guidebooks/tools_ba_use_case_diagram.html
- Ian Sommerville, “Software Engineering”
  Seventh Edition, Pearson Educations, Inc., USA, 2008,PP.43-63

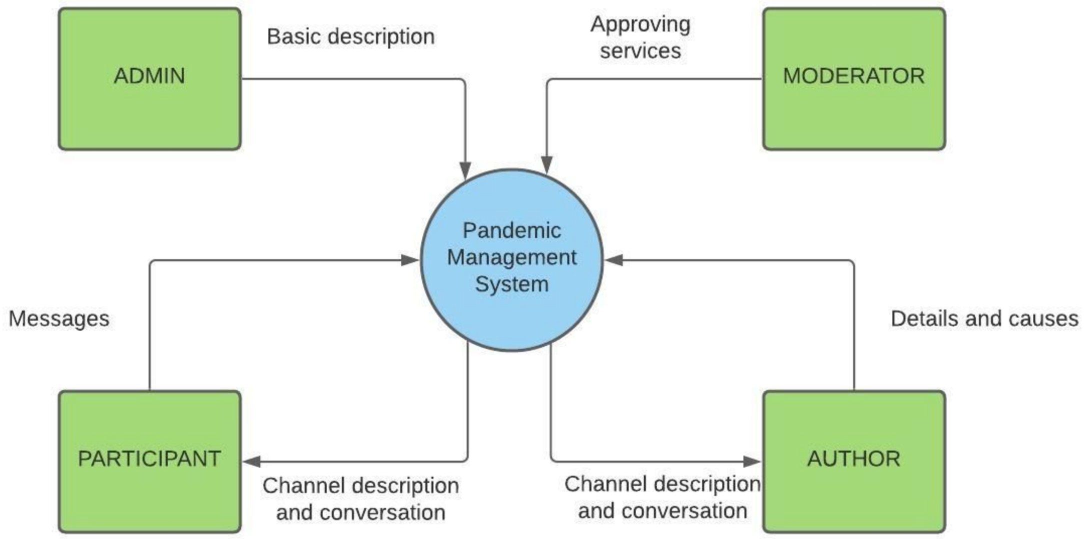
1.6.2 Acknowledgement

It is our pleasure to acknowledge that we have received a wonderful opportunity to work on a project Pandemic Management System (PMS). We would like to thank Dr. Vinod Bhalla,Assistant Professor, TIET, Patiala for the valuable suggestions and information provided towards this project. Our sincere appreciation and gratitude goes to Ms. Deepali Bhagat for her guidance, constructive comments, valuable suggestions and inspiration throughout the development of the project . Finally we would like to thank the whole faculty of Thapar Institute of Engineering and Technology for giving us this great opportunity and providing us all the resources and knowledge necessary.

2 Overall Description

2.1 Product Overview

This is a self-contained product. In other words this product is not derived from any other products but it takes some inspiration from the existing systems like Reddit and Discord.


ADMIN
Basic description
Approving services
MODERATOR
Pandemic Management System
Messages
Details and causes
PARTICIPANT
Channel description and conversation
Channel description and conversation
AUTHOR

This project contains various sub-channels for a particular pandemic and each sub-channel addresses a particular topic. Each sub-channel contains many threads and each thread is nothing but a post in a sub-channel just like a post in a subreddit.
Also these sub-channels mimic the properties of a discord channel i.e. they also have voice channels along with text channels and a sub-channel may also contain bots for assistance of participants or authors.
An admin creates a channel for a particular pandemic/epidemic, then a moderator creates and manages sub-channels within this channel. An author is nothing but a participant that has created a particular thread and has full control over the thread. But the sub-channels are watched over by the moderator. Now the participants can interact with threads i.e they can comment in these threads, share any crucial information or ask questions. For eg. The moderators can conduct an AMA session where people can ask questions and then someone can answer them.
Also the threads can be deleted at any time by the autor, moderator or admin as per their wish if any of the policies are not followed for that thread. After the pandemic comes to an end only useful information can be kept in the channel and other data can be deleted.

2.2 Product Functionality

1. User Login
2. Discussion Platform
3. Threads creation.

2.3 Design and Implementation Constraints

There may be issues with scalability when the number of users will rise which can lead to crashing of the website so the servers and database have to be upgraded.
In order to build an enterprise level application, only those tools and technology can be used which provide sufficient support for such kind of development.
Therefore, our team will be limited to using a highly mature and robust platform for development and deployment of the application. It also must include softer elements such as personnel controls, and administrative controls.

2.4 Assumptions and Dependencies

1. We are making use of several API's and in case any of the API providers shuts the service down for some reason then that will affect our site too.
2. In case the hosting service we are using stops for some reason then the site will be down
3. If something happens to the database then all of the data will be lost which can cause a lot of all.
4. Although the data sources are trusted sources, this data is not verified by our system and the moderators will handle this.
5. The views represented in the threads are personal views of the participants.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user will be interacting with an interface. The platform will firstly get loaded (depending upon whether the user is logged in or not).

- Logged In: Channel List.
- Not Logged In: Login + Register Screen

After the user log's in, A new screen opens up with:

- Navigation Screen on the left containing sub topics (or sub channels)
- The first sub channel opened by default.

After choosing a sub channel, the user will have a list of related threads to interact with. Now they can simply choose the topic of their interests and go through the discussion, and give their opinion.

3.1.2 Hardware Interfaces

For a basic MVP thought following hardware will be used:

- VM instance on GCP
- Load balancing and port forwarding using Nginx.

3.1.3 Software Interfaces

The following software technologies are used :

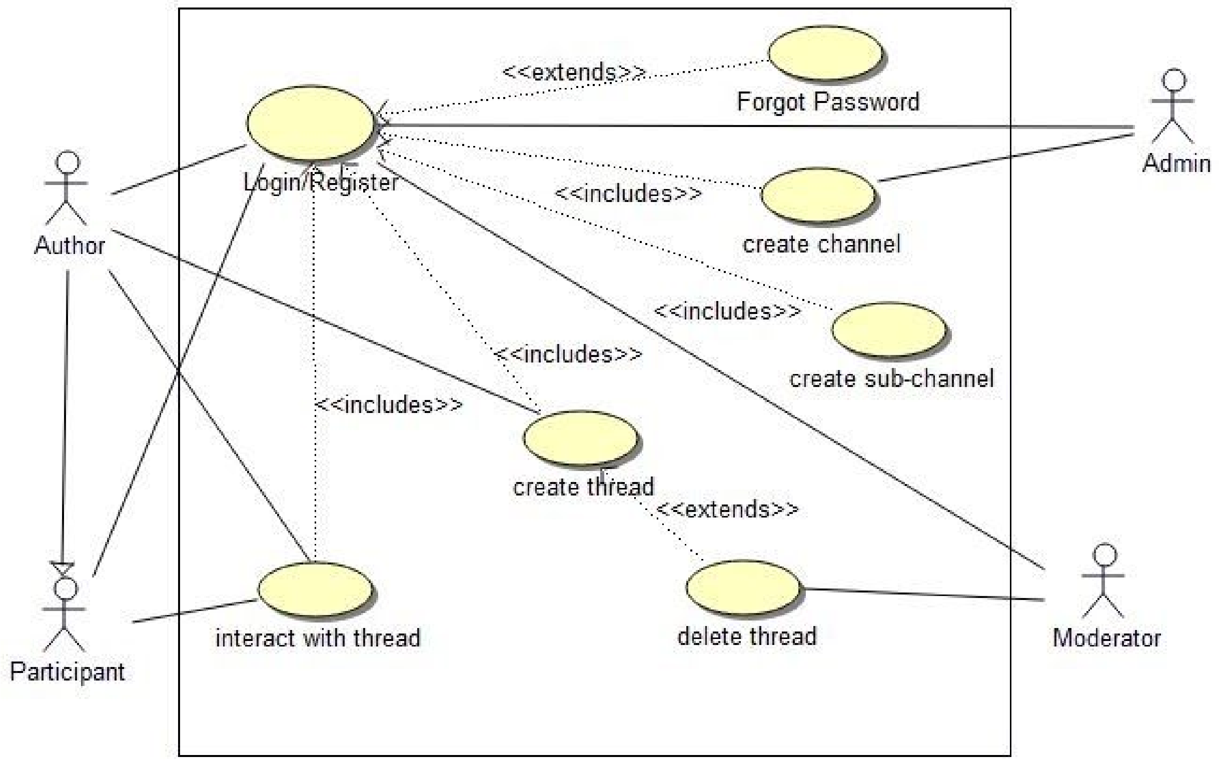
- Frontend: JavaScript, ReactJS, Html, CSS
- Backend: NodeJS, MongoDB

If backed by proper funding to scale the project, then a Kubernetes setup will be required.

3.2 Functional Requirements

1. User Login: Data is provided by the user, it is transmitted securely over https, and then encrypted using Blowfish encryption algorithm and stored in the database.
2. Discussion Platform: Users can come and look at the open discussions. Less security required here for data protection since all info is public. Care to be taken for protection attacks like XSS, CORS access etc.
3. Thread Creation: Create a new db entry of threads. Put ref to it in its parent sub-channel.

3.3 Use Case Model



3.3.1

Use Case #1 (Login/Register and U1)

Author – Purujit

Purpose - The user logins the website.

Requirements Traceability – The user must be registered on the website before.

Priority - High Priority because the user cannot use the website without logging into the system.

Post conditions - The login details will be verified at the backend and then the user will proceed in the website for creating channel/discussion as per the role of the user.

Actors – Author, Participant, Admin, Moderator

Flow of Events

1. Basic Flow -
   In case the user has a verified account be it any role, they enter the website and then interact with whatever is relevant to them.
   If the user is not registered then, they have to register themselves.

3.3.2 Use Case #2 (Forgot Password and U2)

Author - Purujit

Purpose - If a verified user has forgotten their password they can change their password after verification.

Requirements Traceability – The user must be registered on the website before.

Priority - Low priority because it's rare that someone forgets their password very often.

Post conditions - The user will be allowed to set a new password that helps them login to their account again.

Actors – Author, Participant, Admin, Moderator

Flow of Events

1. Basic Flow -
   When the user forgets their password, they use the forgot password and then after re-verification they are given the permission to change their password and then login again.

3.3.3 Use Case #3 (Create Channel and U3)

Author – Mohit Goyal

Purpose - Each pandemic will have its own channel so that the things don't get mixed up.

Requirements Traceability – The user must be an Admin.

Priority - Low Priority because there is a very low chance that there will be more than 1 pandemic at a time i.e only old pandemic channel and 1 active pandemic will have their channel.

Post conditions - A new channel is created.

Actors – Admin

Flow of Events
1. Basic Flow -

When a user is verified as an admin, they have two choices to either create a new channel or do not create a new channel.

2. Alternative Flow - An admin can choose to delete a channel as well but that is very rare.

3.3.4 Use Case #4 (Create Sub-Channel and U4)

Author – Mohit Goyal

Purpose - Create a sub-channel with a particular purpose inside a particular pandemic channel.

Requirements Traceability - The user must be a Moderator or Admin.

Priority - Medium Priority because there can be creation of multiple channels according to the situation.

Pre Conditions: A channel should be created before its creation.

Post conditions - A new sub-channel is created.

Actors – Admin, Moderator

Flow of Events

1. Basic Flow - The moderator/admin can create a new sub-channel or delete a particular sub-channel according to the need.

3.3.5 Use Case #5 (Create Thread and U5)

Author – Chirag Mahajan

Purpose - Create a thread inside a particular pandemic sub-channel and has content related to that particular sub-channel.

Requirements Traceability - The user must be a Moderator, Admin or Author.

Priority - High Priority because there will be a lot of threads because thread is nothing but an article inside the sub-channel.

Post conditions - A new thread is created(for the discussion)

Actors – Admin, Moderator, Author

Flow of Events

1. Basic Flow - The moderator/admin/author can create a new thread and moderator/admin can delete a particular thread if the thread is irrelevant/offensive or for any other reason.

3.3.6 Use Case #6 (Interact Thread and U6)

Author – Chirag Mahajan

Purpose - Interaction in the thread which has a correlation with the sub-channel with a particular purpose inside a particular pandemic subchannel.

Requirements Traceability - The user can be a participant/author/moderator/admin .

Priority - High Priority because there will be a lot of discussions.

Pre Conditions: A thread should be created before its interaction.

Post conditions - New Sub-threads are created.

Actors – Participant, Admin, Moderator, Author

Flow of Events

1. Basic Flow - Any user can interact with the particular thread and create sub-threads.

3.3.7 Use Case #7 (Delete Thread and U7)

Author – Rachit Anand Srivastava

Purpose - Delete the thread which has a correlation with the sub-channel with a particular purpose inside a particular pandemic subchannel.

Requirements Traceability - The user must be a Moderator/Admin/Author.

Priority - Medium Priority because there can be multiple thread deletions at the same time.

Pre Conditions: A thread should be created before its deletion.

Post conditions - The thread does not exist anymore.

Actors – Admin, Moderator, Author

Flow of Events

1. Basic Flow - The moderator/admin can create a new channel or delete a particular channel according to the need.

4 Other Non-functional Requirements

4.1 Performance Requirements

- If deployed on a PaaS, the per day request will be around 2k / day. No limitation on the number of users logged in.
- Technology like Code Splitting will be implemented, so not much will be required from the end user.
- A 3g internet connection will be required for smooth experience.

4.2 Safety and Security Requirements

- Personal information, like email and password will be encrypted and then stored in the database.
- Special wrapper will be used to prevent vulnerabilities such as XSS, CSRF, injections etc.
- Conversation on the channels won't be encrypted since that data is already public.
- Measures taken to limit access to the server for keeping the server itself secure (like port blocking, firewall setup, root access denied etc.).

4.3 Software Quality Attributes

4.3.1 Adaptability

The PMS is adaptable to different pandemics and meeting its requirements by having a new channel for every pandemic. In this way the information of different pandemics remains categorised and caters to the needs of future changes.
Also this website is scalable to a certain level.

4.3.2 Availability

Our website is available for all the devices or in other words the tech required in order to access our website is not required to be very advanced. Any device that has an active internet connection can be used to access our website. Further this website can work as a native application on both android and ios devices by PWA.

5 Other Requirements

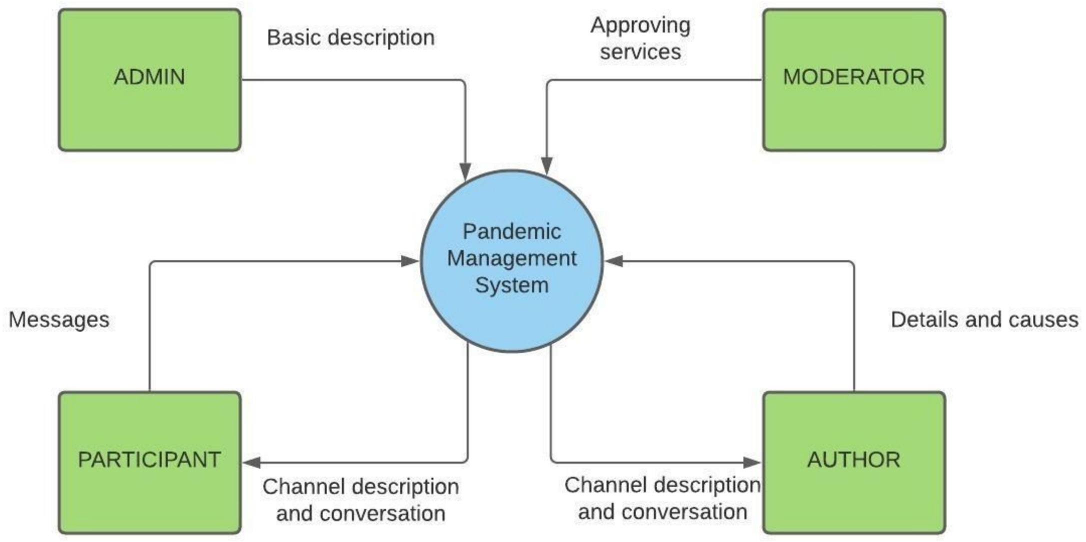
The final product will undergo rigorous testing.

- Mocha and chai frameworks will be used for backend.
- Database testing and jest for frontend.

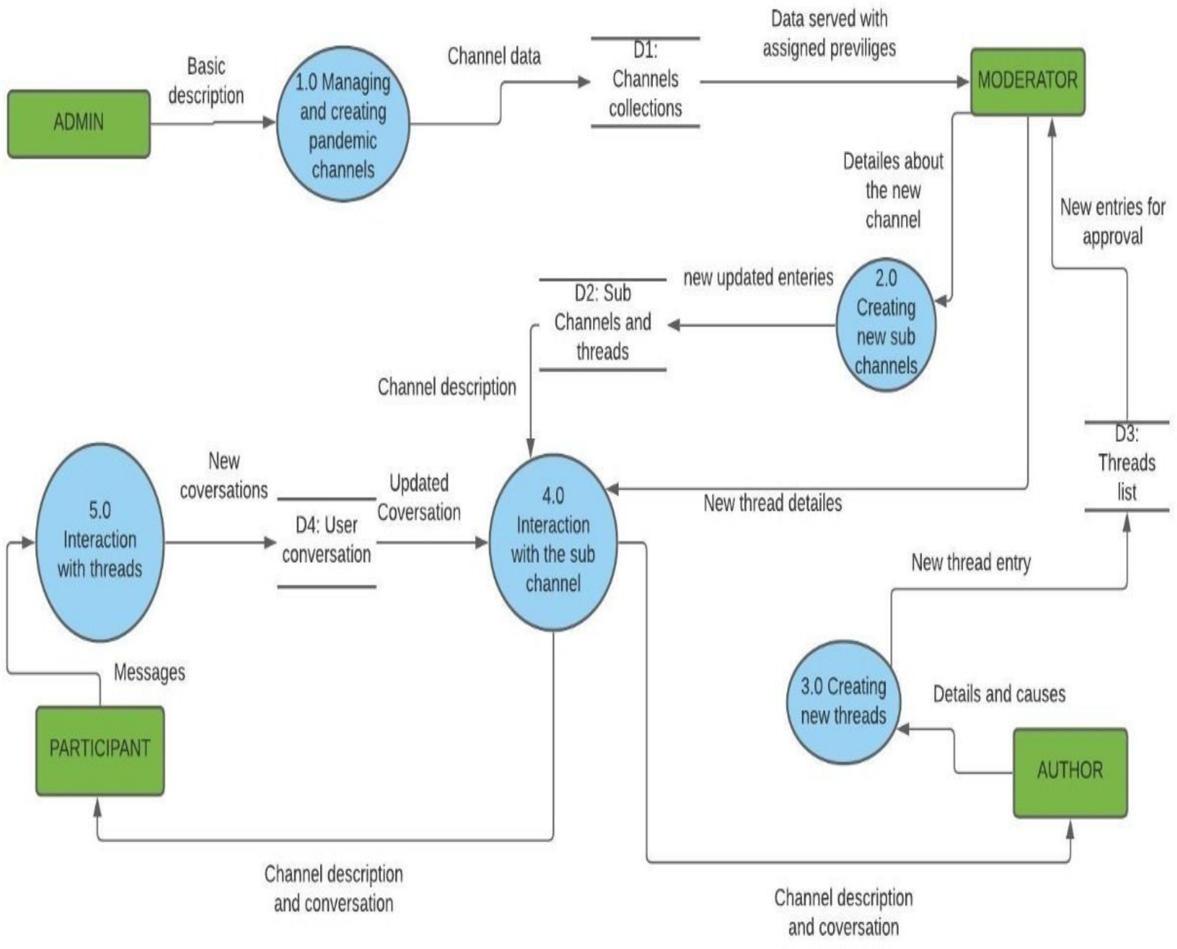
Appendix A - Group Log

S.No.	Group Activities	Members Participated	Date
1.	Online Meeting For Making Use Case Diagram	All	06/09/2020
2.	Final Editing Of Use Case Diagram	All	07/09/2020
3.	Use Case Models + Changes in Diagram	All	13/09/2020
4.	Introduction	All	17/09/2020
5.	Overall Description	All	17/09/2020
6.	Specific Requirements	All	20/09/2020
7.	Functional Requirements + Non-Functional Requirements	All	21/09/2020
8.	Editing	All	23/09/2020
9.	Combined all of the materials and put it into word document.	All	3/12/2020

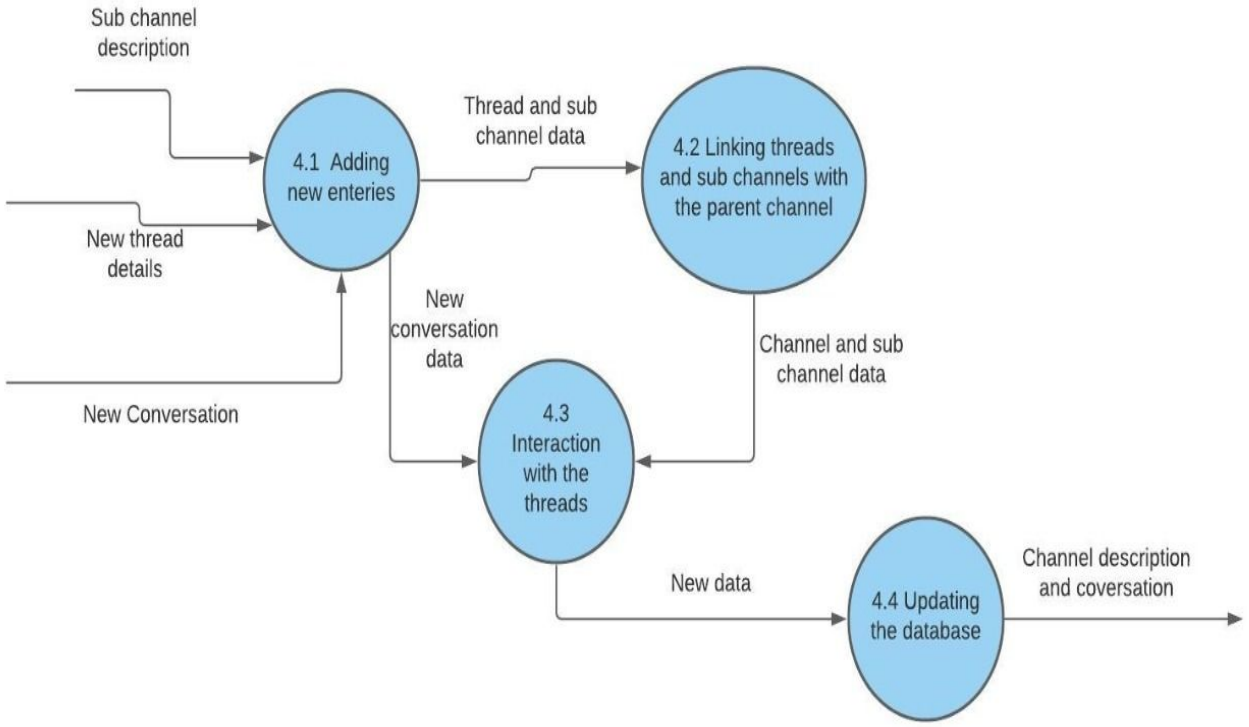
DFD LEVEL 0


ADMIN
Basic description
Approving services
MODERATOR
Pandemic Management System
Messages
Details and causes
PARTICIPANT
Channel description and conversation
Channel description and conversation
AUTHOR

DFD LEVEL 1

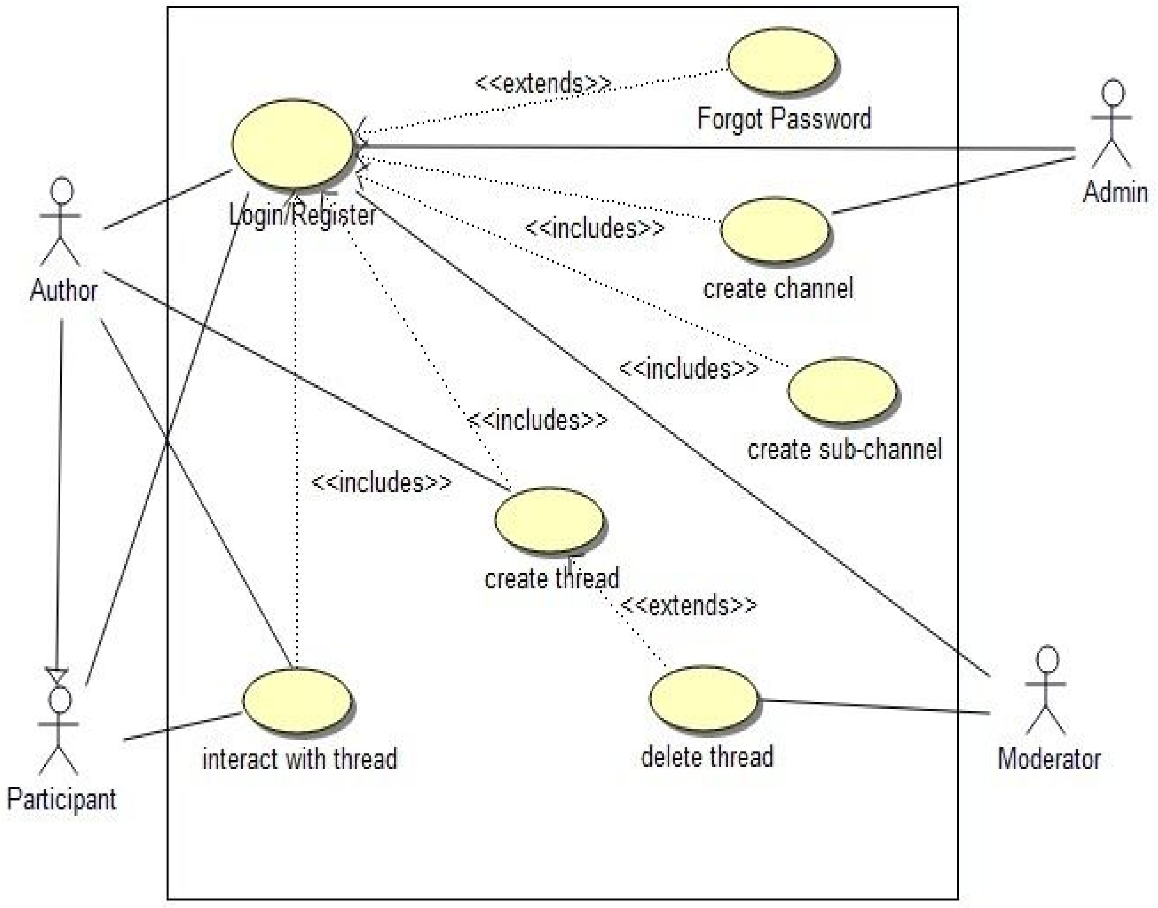

ADMIN
Basic description
1.0 Managing and creating pandemic channels
Channel data
D1: Channels collections
Data served with assigned previliges
MODERATOR
Detailes about the new channel
New entries for approval
new updated enteries
2.0 Creating new sub channels
D2: Sub Channels and threads
Channel description
D3: Threads list
5.0 Interaction with threads
New coversations
D4: User conversation
Updated Coversation
4.0 Interaction with the sub channel
New thread detailes
New thread entry
Messages
3.0 Creating new threads
Details and causes
PARTICIPANT
AUTHOR
Channel description and conversation
Channel description and coversation

DFD LEVEL 2

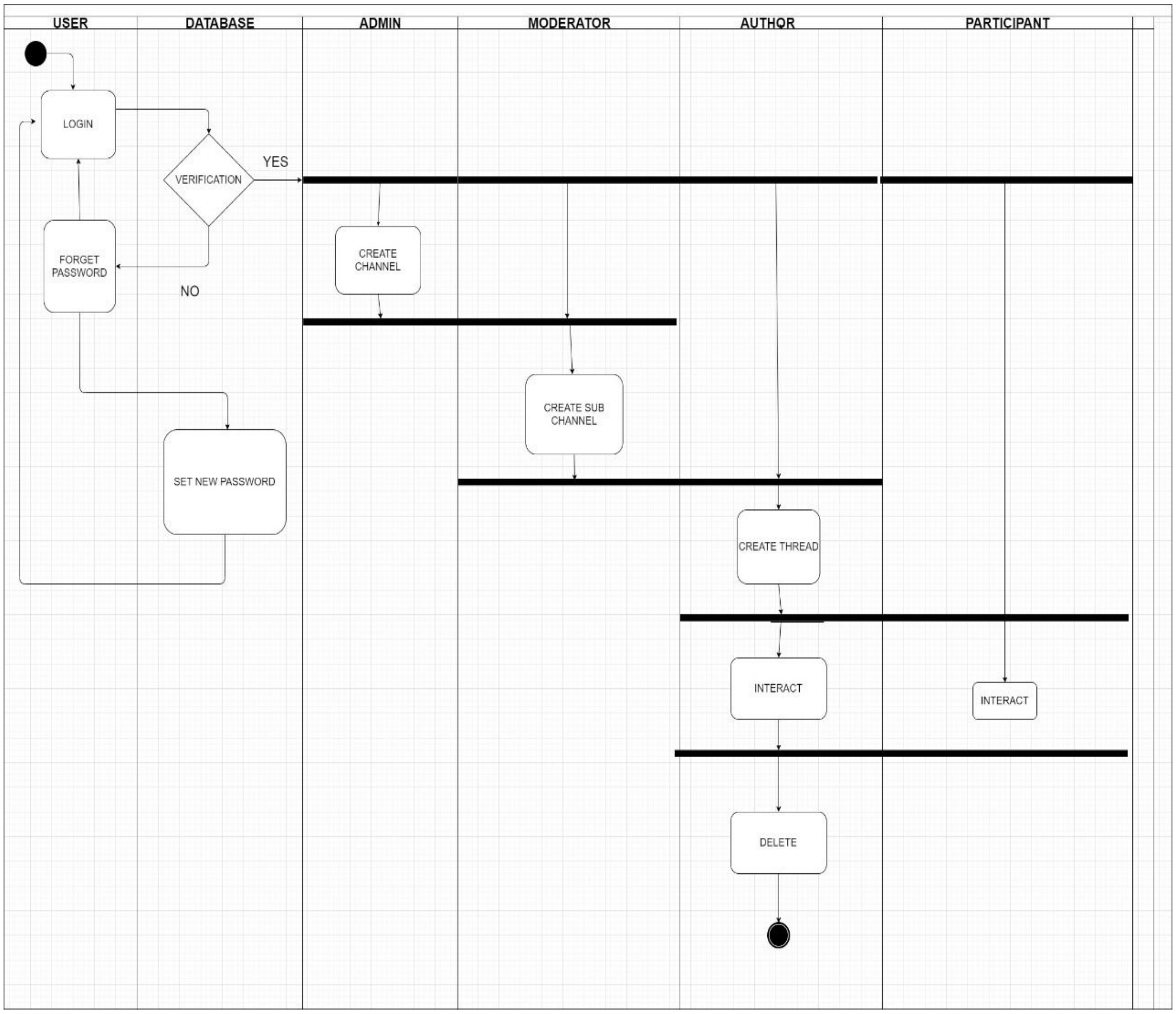

Sub channel description
New thread details
New Conversation
4.1 Adding new enteries
Thread and sub channel data
4.2 Linking threads and sub channels with the parent channel
New conversation data
Channel and sub channel data
4.3 Interaction with the threads
New data
4.4 Updating the database
Channel description and coversation

OBJECT ORIENTED ANALYSIS

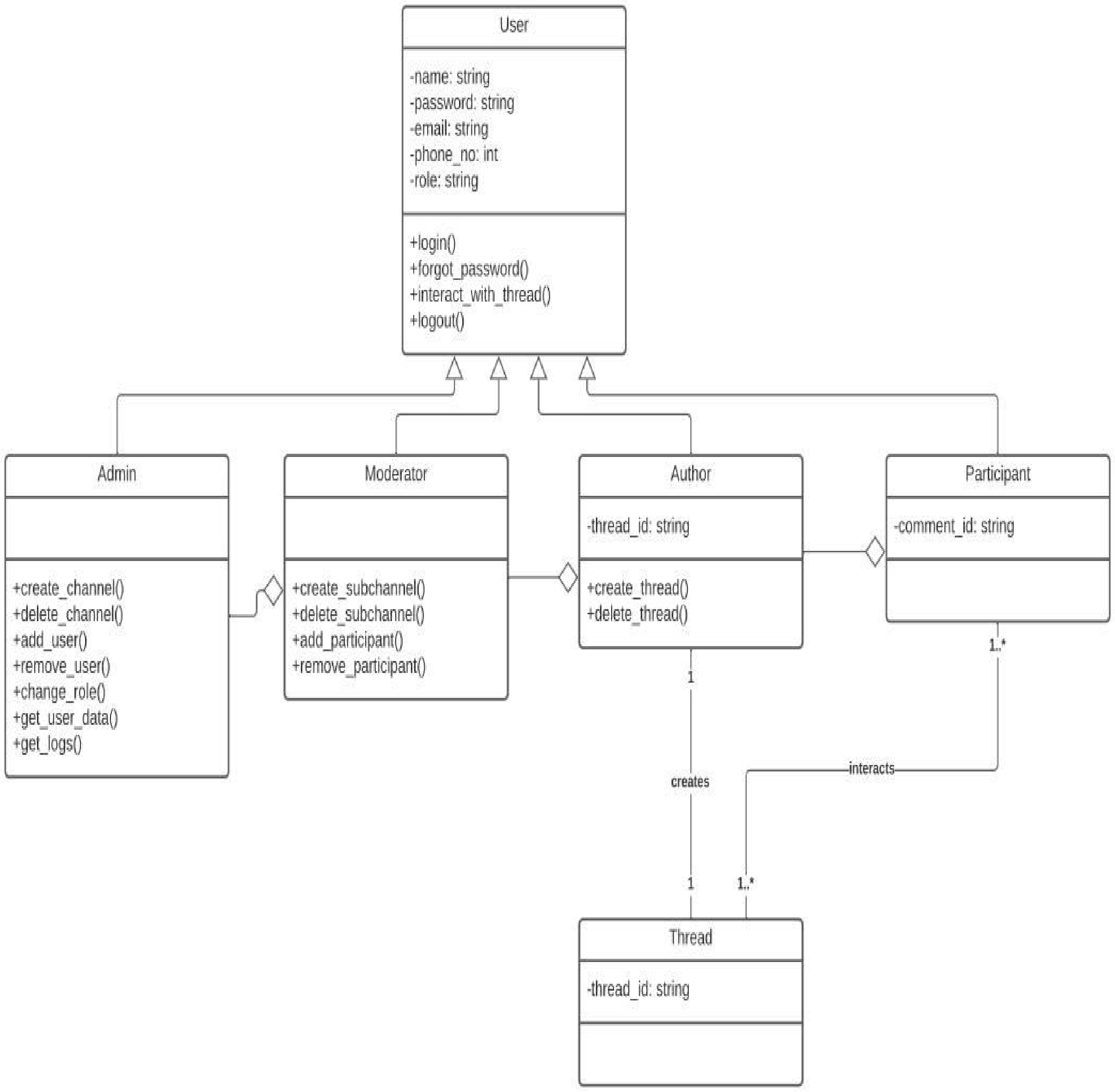
USE CASE DIAGRAM



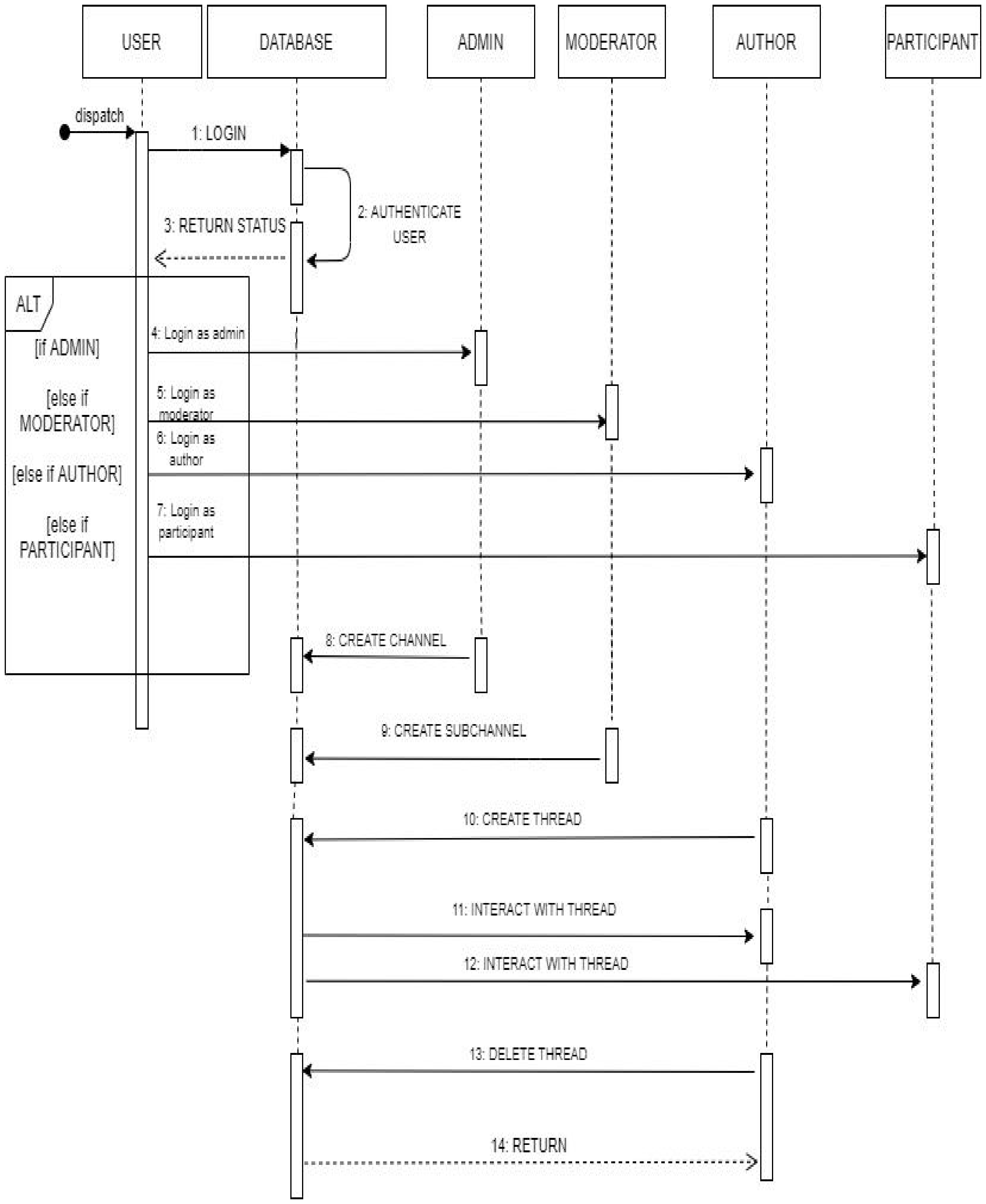
ACTIVITY DIAGRAM



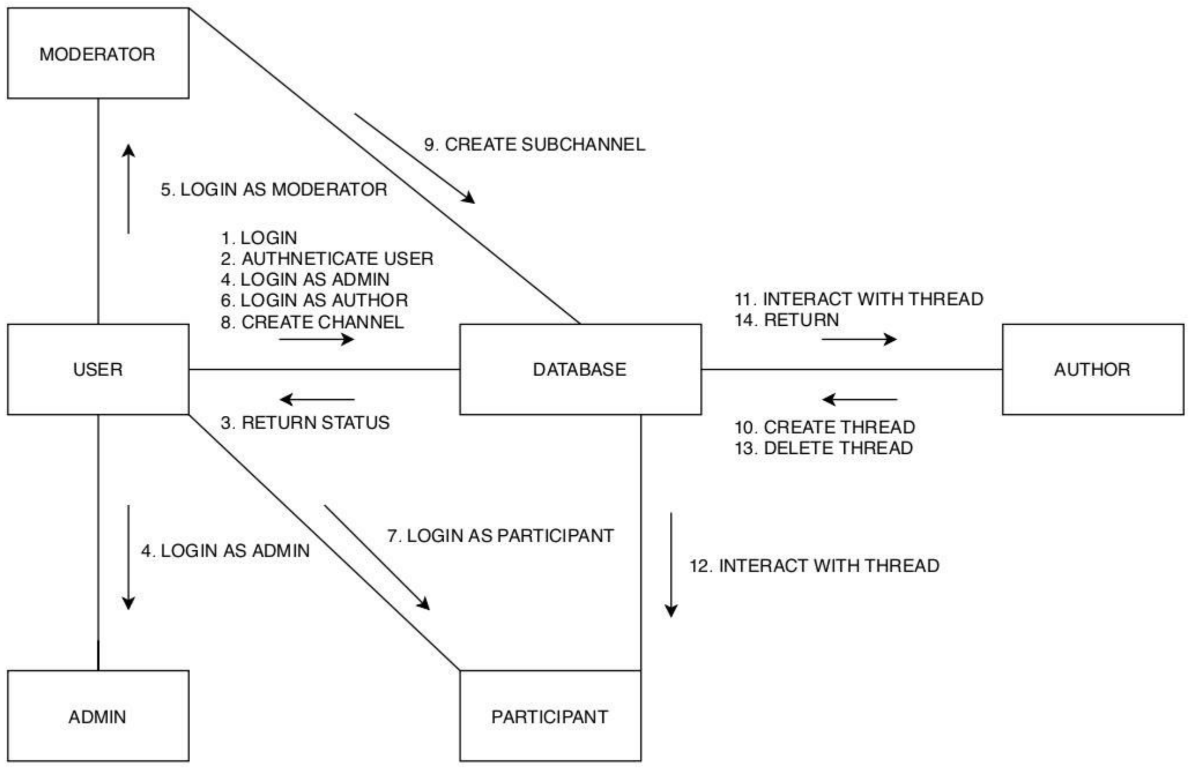
CLASS DIAGRAM



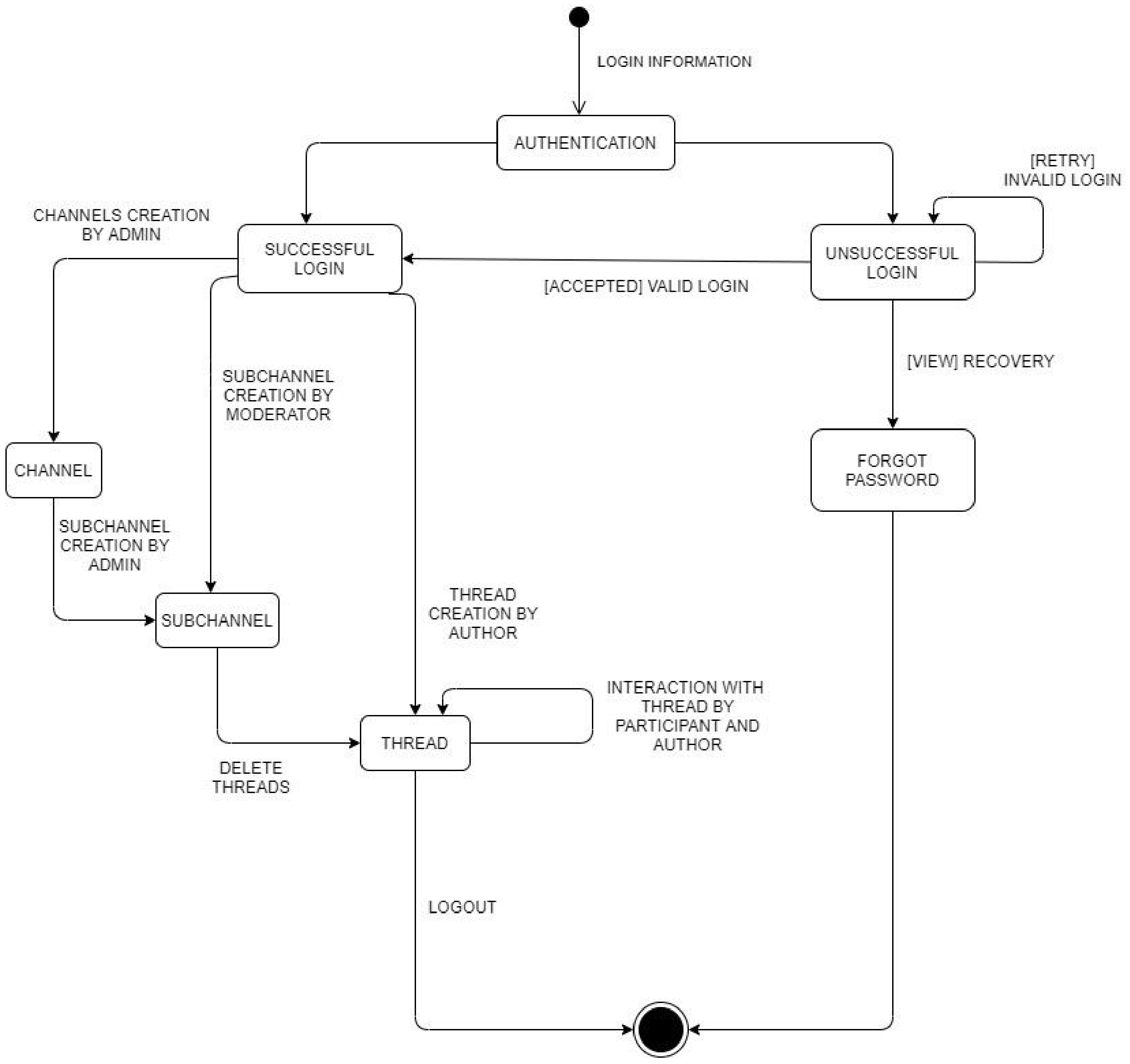
SEQUENCE DIAGRAM



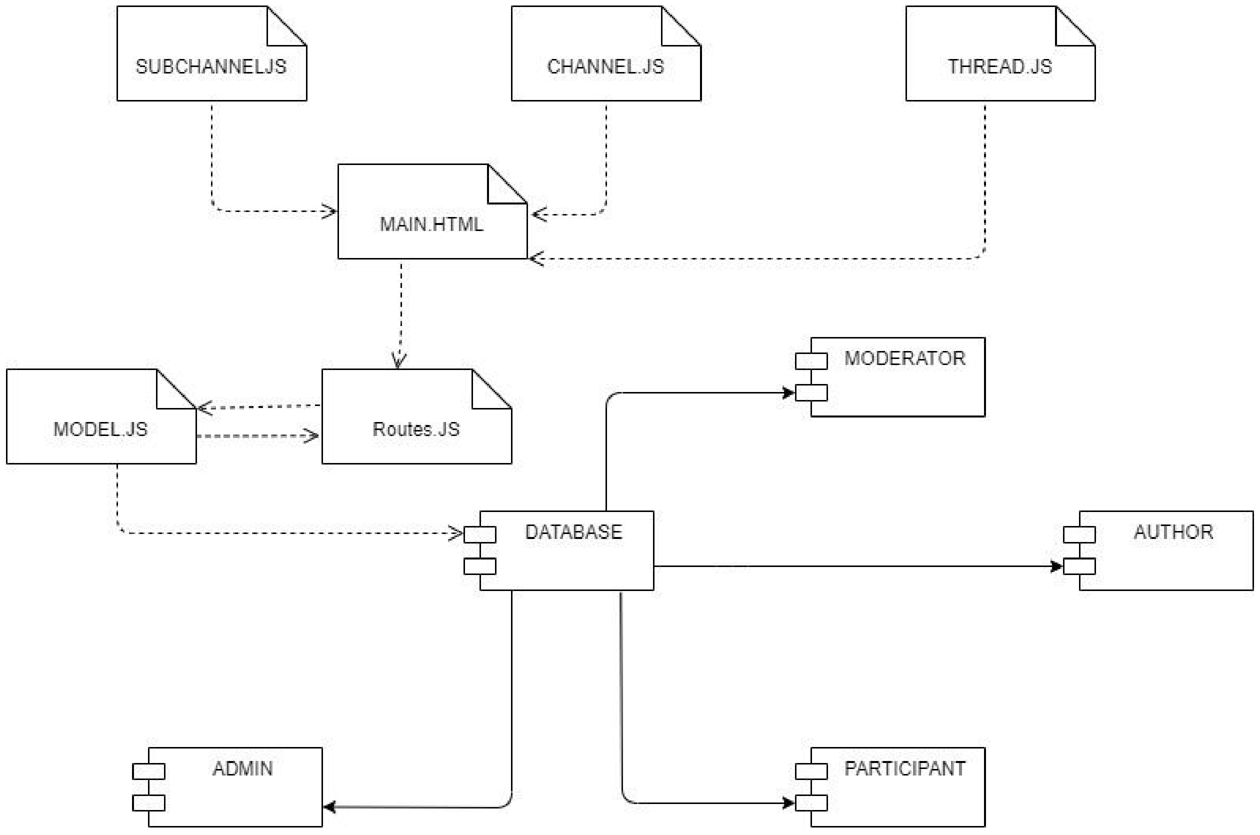
COLLABORATION DIAGRAM



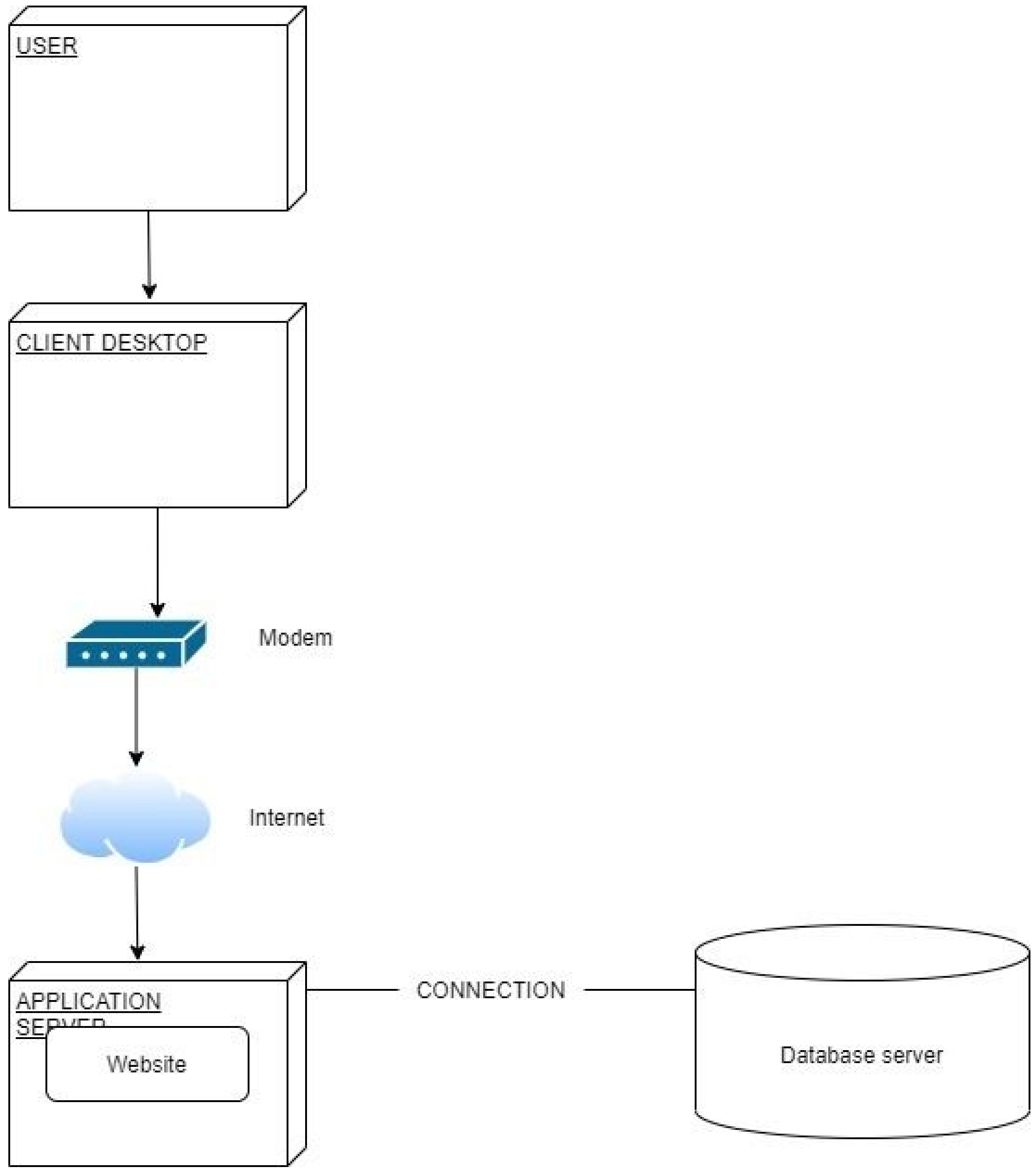
STATE CHART DIAGRAM



COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



TESTING

TEST PLAN

Test cases were run on the news application on 2 different laptops . If the website did not crash and it produced the desired result without the freezing, the test case is considered to be passed.
The testing involved the documentation of average time taken by the application to respond to an activity i.e. taking care of the time lag between query and response. It was made sure that the databases are reliable and robust against all cases. Security issues are also handled like the personal information of the user.

Scope

To test the listed functional requirements of the product along with speed, robustness and accuracy of the product.

Approach

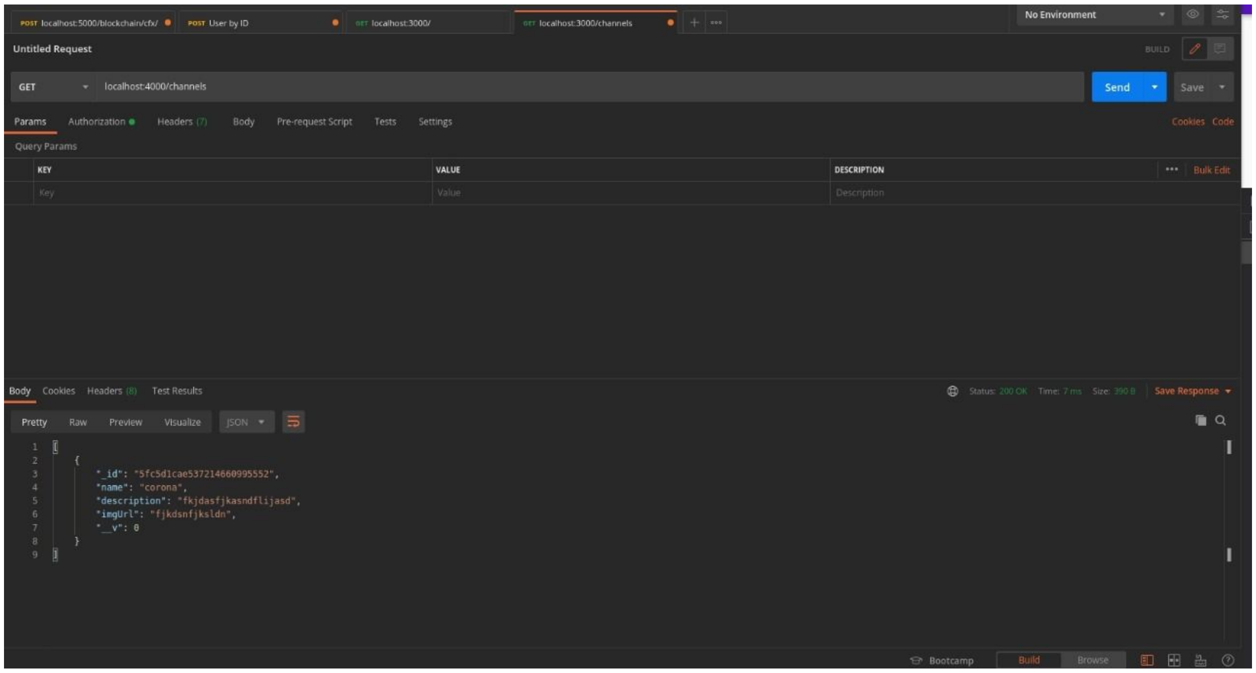
Manual testing of application on two different windows/linux/mac devices for any bugs or non-compliance with the requirements.

Item Pass Fail Criteria

If the testing produces the desired result for every activity without any freezing or taking very long time to execute, it is considered pass, otherwise fail.

TEST CASE REPORT

Test Case	Channel API
Test Case Summary	Client makes a GET request to the server.
Pre-requisites	The API must return the list of all channels.
Test-Data	
Expected Result	A list of all the channels should be returned.
Status	PASS
Executed By	Rachit
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2

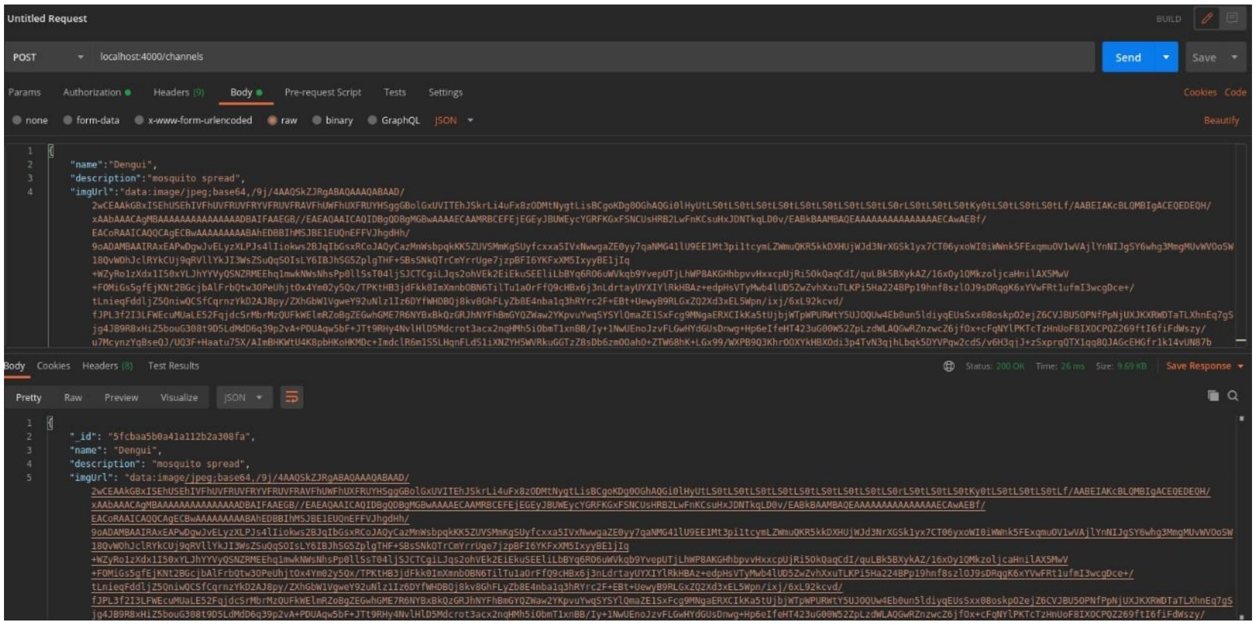


GET localhost:4000/channels

Status: 200 OK Time: 7 ms Size: 390 B

```
[
    {
        "_id": "5fc5d1cae537214660995552",
        "name": "corona",
        "description": "fkjdasfjkasndflijasd",
        "imgUrl": "fjkdsnfjksldn",
        "__v": 0
    }
]
```

Test Case	Create Channel (admin)
Test Case Summary	A new channel created.
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	User name: <value> Email: <value> Password: <value> Confirm Password: <value>
Expected Result	New Channel created in the database.
Status	PASS
Executed By	PURUJIT
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2



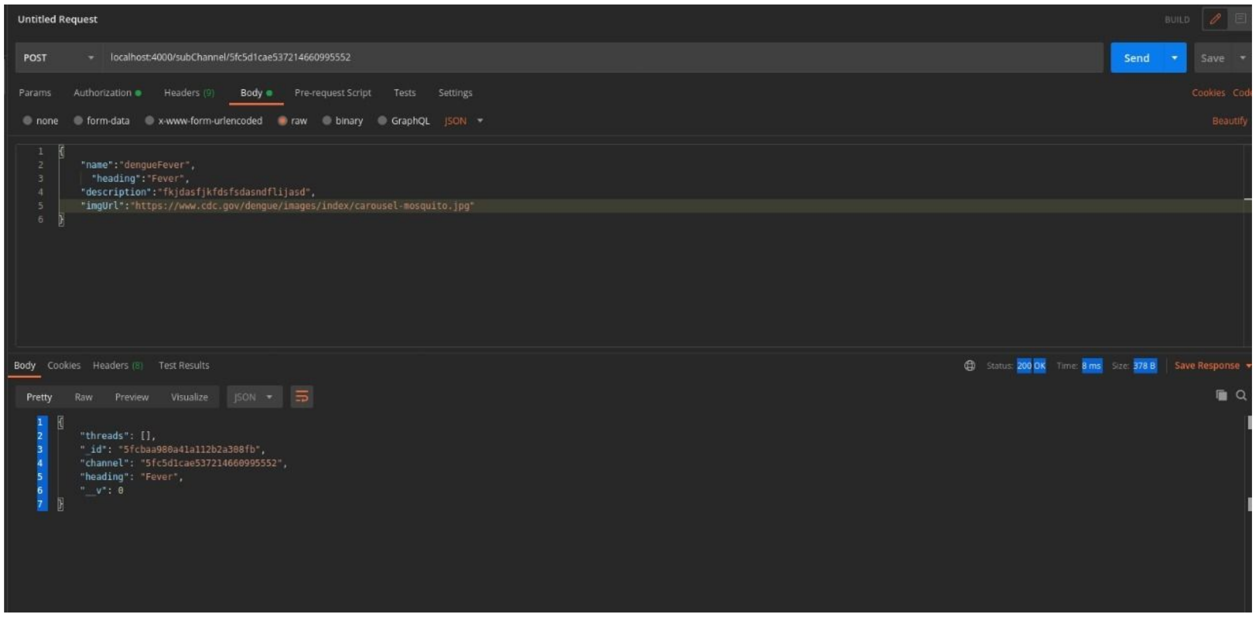
POST localhost:4000/channels

```
{
    "name":"Dengui",
    "description":"mosquito spread",
    "imgUrl":"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/
        2wCEAAkGBxISEhUSEhIVFhUVFRUVFRYVFRUVFRAVFhUWFhUXFRUYHSggGBolGxUVITEhJSkrLi4uFx8zODMtNygtLisBCgoKDg0OGhAQGi0lHyUtLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0rLS0tLS0tLS0tKy0tLS0tLS0tLS0tLf/AABEIAKcBLQMBIgACEQEDEQH/
        xAAbAAACAgMBAAAAAAAAAAAAAAAADBAIFAAEGB//EAEAQAAICAQIDBgQDBgMGBwAAAAECAAMRBCEFEjEGEyJBUWEycYGRFKGxFSNCUsHRB2LwFnKCsuHxJDNTkqLD0v/EABkBAAMBAQEAAAAAAAAAAAAAAAECAwAEBf/
        EACoRAAICAQQCAgECBwAAAAAAAAABAhEDBBIhMSJBE1EUQnEFFVJhgdHh/
        9oADAMBAAIRAxEAPwDgwJvELyzXLPJs4lIiokws2BJqIbGsxRCoJAQyCazMnWsbpqkKK5ZUVSMmKgSUyfcxxa5IVxNwwgaZE0yy7qaNMG41lU9EE1Mt3pi1tcymLZWmuQKR5kkDXHUjWJd3NrXGSk1yx7CT06yxoWI0iWWnk5FExqmuOV1wVAjlYnNIJgSY6whg3MmgMUvWVOoSW
        18QvWOhJclRYkCUj9qRVllYkJI3WsZSuQqSOIsLY6IBJhSG5ZplgTHF+SBsSNkQTrCmYrrUge7jzpBFI6YKFxXM5IxyyBE1jIq
        +WZyRo1zXdx1I50xYLJhYYVyQSNZRMEEhq1mwkNWsNhsPp0llSsT04ljSJCTCgiLJqs2ohVEk2EiEkuSEEliLbBYq6RO6uWVkqb9YvepUTjLhWP8AKGHhbpvvHxxcpUjRi5OkQaqCdI/quLBk5BXykAZ/16xOy1QMkzoljcaHnilAX5MwV
        +FOMiGs5gfEjKNt2BGcjbAlFrbQtw3OPeUhjtOx4Ym02y5Qx/TPKtHB3jdFkk0ImXmnbOBN6TilTu1aOrFfQ9cHBx6j3nLdrtayUYXIYlRkHBAz+edpHsVTyMwb4lUD5ZwZvhXxuTLKPi5Ha224BPp19hnf8szl0J9sDRqgK6xYVwFRt1ufmI3wcgDce+/
        tLnieqFddljZ5QniwQCSfCqrnzYkD2AJ8py/ZXhGbW1VgweY92uNlz1Iz6DYfWHDBQj8kv8GhFLyZb8E4nba1q3hRYrc2F+EBt+UewyB9RLGxZQ2Xd3xEL5Wpn/ixj/6xL92kcvd/
        fJPL3f2I3LFWEcuMUaLE52FqjdcSrMbrMzQUFkWElmRZoBgZEGwhGME7R6NYBxBkQzGRJhNYFhBmGYQZWaw2YKpvuYwqSYSYlQmaZE1SxFcg9MNgaERXCIkKa5tUjbjWTpWPURWtY5UJOQUw4Eb0un5ldiyqEUsSxx08oskpO2ejZ6CVJBU5OPNfPpNjUXJKXRWDTaTLXhnEq7gS
        jg4JB9R8xHiZ5bouG308t9D5LdMdD6q39p2vA+PDUAqw5bF+JTt9RHy4NvlHlD5Mdcrot3acx2nqHMh5i0bmT1xnBB/Iy+1NwUEnoJzvFLGwHYdGUsDnwg+Hp6eIfeHT423uG00W52ZpLzdWLAQGwRZnzwcZ6jfOx+cFqNYlPKTcTzHnUoF8IXOCPQZ269ftI6fiFdWszy/
        u7McynzYqBseQJ/UQ3F+Haatu75X/AImBHKWtU4K8pbHKoHKMDc+ImdclR6m1S5LHqnFLdS1iXNZYH5WVRkuGGTzZ8sDb6zmO0ahO+ZTW68hK+LGx99/WXPB9Q3KhrOOXYkHBXOdi3p4TvN3qjhLbqk5DYVPqw2cdS/v6H3qjJ+zSxprqQTX1qq8QJAGcEHGfr1k14vUN87b
```

Status: 200 OK Time: 26 ms Size: 9.69 KB

```
{
    "_id": "5fcbaa5b0a41a112b2a308fa",
    "name": "Dengui",
    "description": "mosquito spread",
    "imgUrl": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/
        2wCEAAkGBxISEhUSEhIVFhUVFRUVFRYVFRUVFRAVFhUWFhUXFRUYHSggGBolGxUVITEhJSkrLi4uFx8zODMtNygtLisBCgoKDg0OGhAQGi0lHyUtLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0rLS0tLS0tLS0tKy0tLS0tLS0tLS0tLf/AABEIAKcBLQMBIgACEQEDEQH/
        xAAbAAACAgMBAAAAAAAAAAAAAAAADBAIFAAEGB//EAEAQAAICAQIDBgQDBgMGBwAAAAECAAMRBCEFEjEGEyJBUWEycYGRFKGxFSNCUsHRB2LwFnKCsuHxJDNTkqLD0v/EABkBAAMBAQEAAAAAAAAAAAAAAAECAwAEBf/
        EACoRAAICAQQCAgECBwAAAAAAAAABAhEDBBIhMSJBE1EUQnEFFVJhgdHh/
        9oADAMBAAIRAxEAPwDgwJvELyzXLPJs4lIiokws2BJqIbGsxRCoJAQyCazMnWsbpqkKK5ZUVSMmKgSUyfcxxa5IVxNwwgaZE0yy7qaNMG41lU9EE1Mt3pi1tcymLZWmuQKR5kkDXHUjWJd3NrXGSk1yx7CT06yxoWI0iWWnk5FExqmuOV1wVAjlYnNIJgSY6whg3MmgMUvWVOoSW
        18QvWOhJclRYkCUj9qRVllYkJI3WsZSuQqSOIsLY6IBJhSG5ZplgTHF+SBsSNkQTrCmYrrUge7jzpBFI6YKFxXM5IxyyBE1jIq
        +WZyRo1zXdx1I50xYLJhYYVyQSNZRMEEhq1mwkNWsNhsPp0llSsT04ljSJCTCgiLJqs2ohVEk2EiEkuSEEliLbBYq6RO6uWVkqb9YvepUTjLhWP8AKGHhbpvvHxxcpUjRi5OkQaqCdI/quLBk5BXykAZ/16xOy1QMkzoljcaHnilAX5MwV
        +FOMiGs5gfEjKNt2BGcjbAlFrbQtw3OPeUhjtOx4Ym02y5Qx/TPKtHB3jdFkk0ImXmnbOBN6TilTu1aOrFfQ9cHBx6j3nLdrtayUYXIYlRkHBAz+edpHsVTyMwb4lUD5ZwZvhXxuTLKPi5Ha224BPp19hnf8szl0J9sDRqgK6xYVwFRt1ufmI3wcgDce+/
        tLnieqFddljZ5QniwQCSfCqrnzYkD2AJ8py/ZXhGbW1VgweY92uNlz1Iz6DYfWHDBQj8kv8GhFLyZb8E4nba1q3hRYrc2F+EBt+UewyB9RLGxZQ2Xd3xEL5Wpn/ixj/6xL92kcvd/
        fJPL3f2I3LFWEcuMUaLE52FqjdcSrMbrMzQUFkWElmRZoBgZEGwhGME7R6NYBxBkQzGRJhNYFhBmGYQZWaw2YKpvuYwqSYSYlQmaZE1SxFcg9MNgaERXCIkKa5tUjbjWTpWPURWtY5UJOQUw4Eb0un5ldiyqEUsSxx08oskpO2ejZ6CVJBU5OPNfPpNjUXJKXRWDTaTLXhnEq7gS
        jg4JB9R8xHiZ5bouG308t9D5LdMdD6q39p2vA+PDUAqw5bF+JTt9RHy4NvlHlD5Mdcrot3acx2nqHMh5i0bmT1xnBB/Iy+1NwUEnoJzvFLGwHYdGUsDnwg+Hp6eIfeHT423uG00W52ZpLzdWLAQGwRZnzwcZ6jfOx+cFqNYlPKTcTzHnUoF8IXOCPQZ269ftI6fiFdWszy/
```

Test Case	Create SubChannel
Test Case Summary	Creating Sub-Channelsusing Channel ID
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	User name: <value> Email: <value> Password: <value? Confirm Password: <value>
Expected Result	Details about the sub-channels should be received on the server and stored on the database.
Status	PASS
Executed By	RACHIT
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2



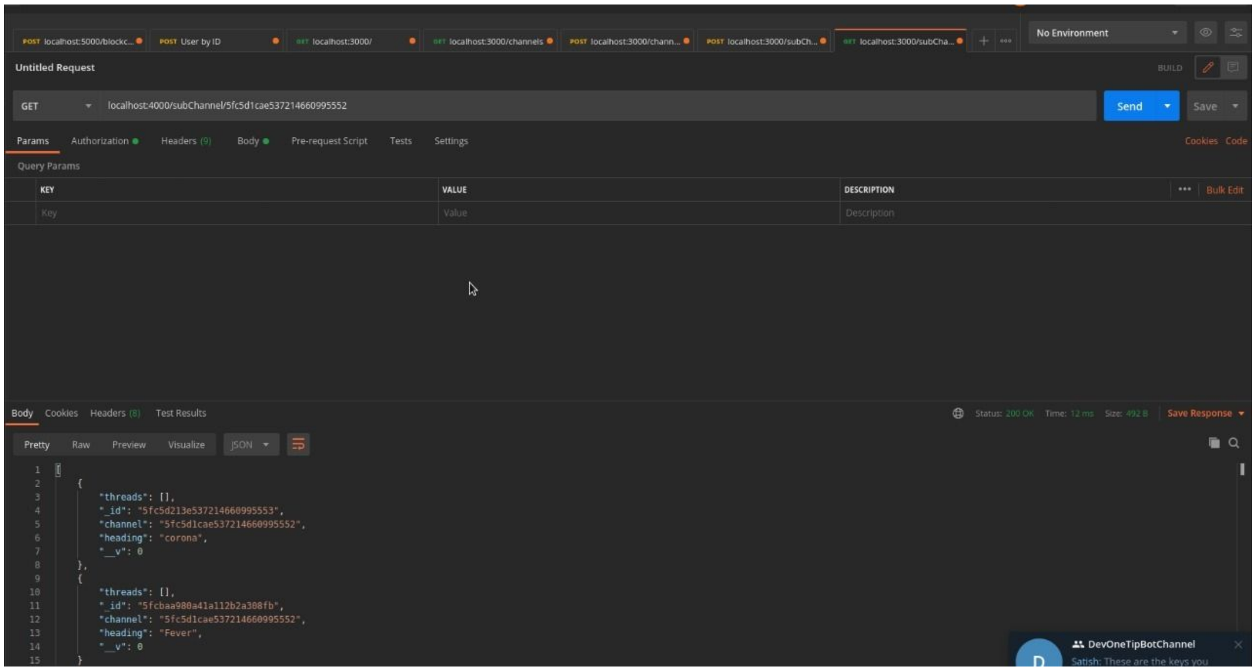
POST localhost:4000/subChannel/5fc5d1cae537214660995552

```
{
    "name":"dengueFever",
      "heading":"Fever",
    "description":"fkjdasfjkfdsfsdasndflijasd",
    "imgUrl":"https://www.cdc.gov/dengue/images/index/carousel-mosquito.jpg"
}
```

Status: 200 OK Time: 8 ms Size: 378 B

```
{
    "threads": [],
    "_id": "5fcbaa980a41a112b2a388fb",
    "channel": "5fc5d1cae537214660995552",
    "heading": "Fever",
    "__v": 0
}
```

Test Case	Get Sub Channel Details
Test Case Summary	Get the list of all sub-channels in a given Channel.
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	
Expected Result	Array of all the details of sub channels received.
Status	PASS
Executed By	PURUJIT
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2

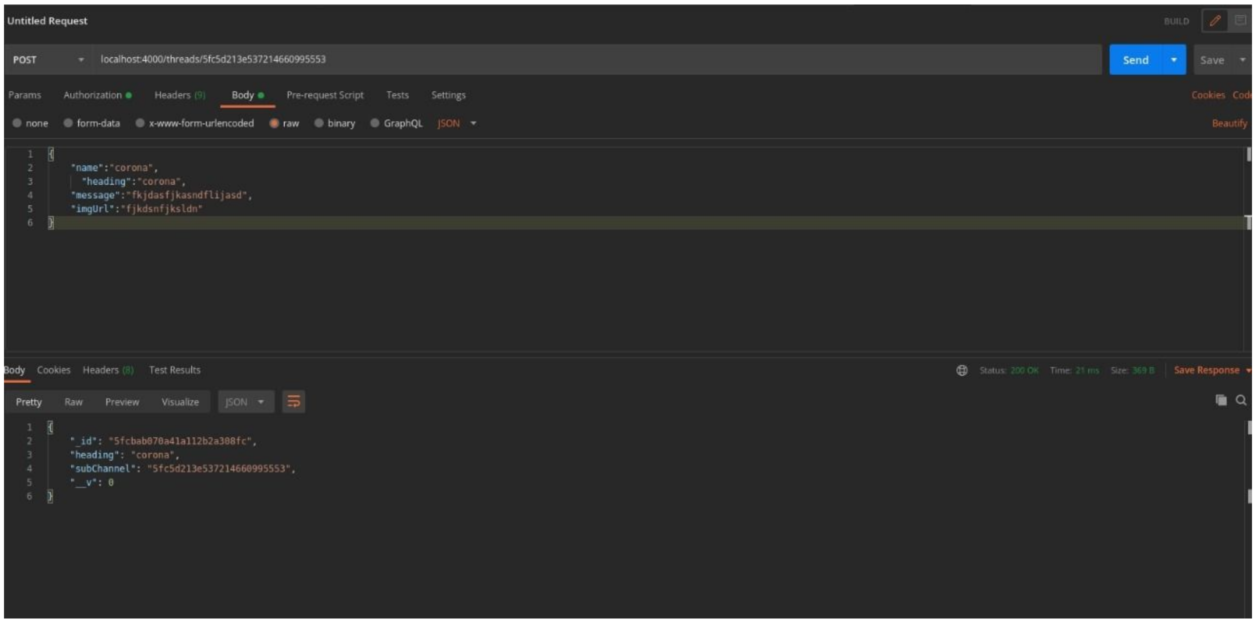


GET localhost:4000/subChannel/5fc5d1cae537214660995552

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
[
    {
        "threads": [],
        "_id": "5fc5d213e537214660995553",
        "channel": "5fc5d1cae537214660995552",
        "heading": "corona",
        "__v": 0
    },
    {
        "threads": [],
        "_id": "5fcbaa980a41a112b2a308fb",
        "channel": "5fc5d1cae537214660995552",
        "heading": "Fever",
        "__v": 0
    }
```

Test Case	Create Thread
Test Case Summary	A new thread gets created.
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	Phone Number: <value> OTP: <value>
Expected Result	New thread entry in the database with reference to sub channel.
Status	PASS
Executed By	MOHIT
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2



POST localhost:4000/threads/5fc5d213e537214660995553

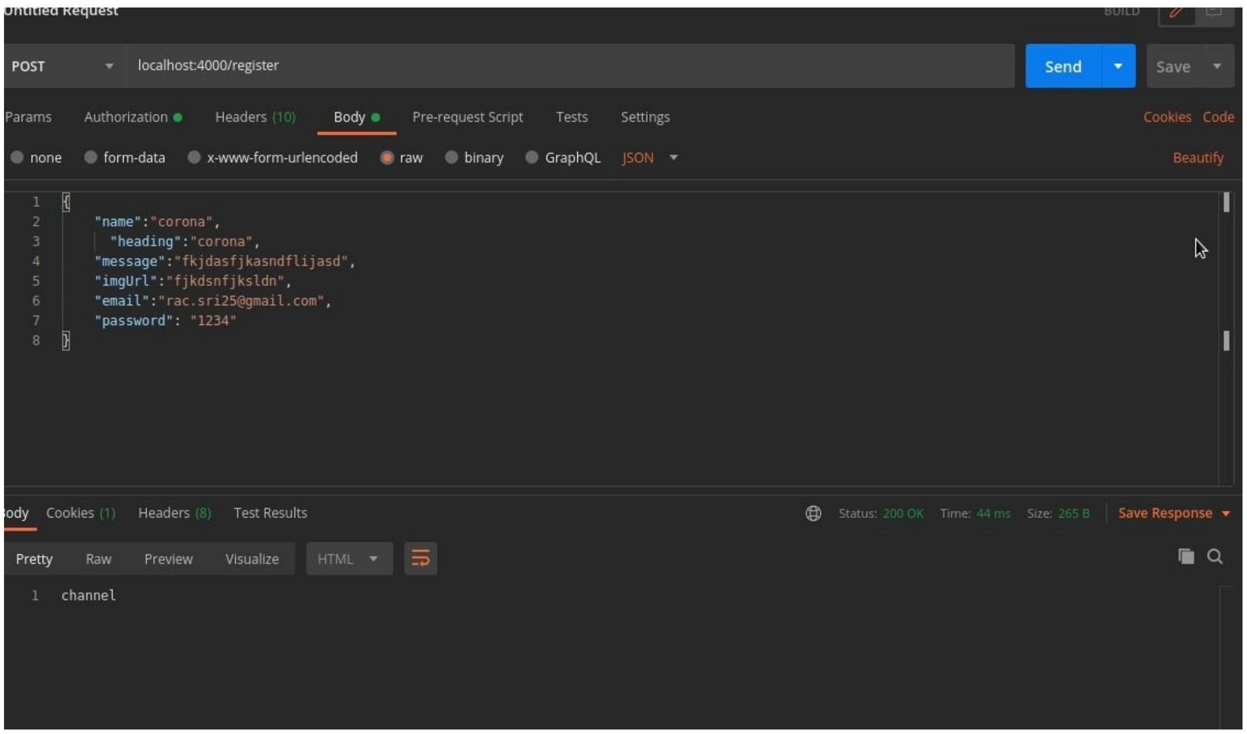
```
{
    "name":"corona",
      "heading":"corona",
    "message":"fkjdasfjkasndflijasd",
    "imgUrl":"fjkdsnfjksldn"
}
```

Status: 200 OK Time: 21 ms Size: 369 B

Body: JSON

```
{
    "_id": "5fcbab070a41a112b2a308fc",
    "heading": "corona",
    "subChannel": "5fc5d213e537214660995553",
    "__v": 0
}
```

Test Case	User Signup
Test Case Summary	User tries to enter with concurrent numbers.
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	name: <value> password:<value> email: <value>
Expected Result	User gets registered.
Status	PASS
Executed By	Chirag
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2



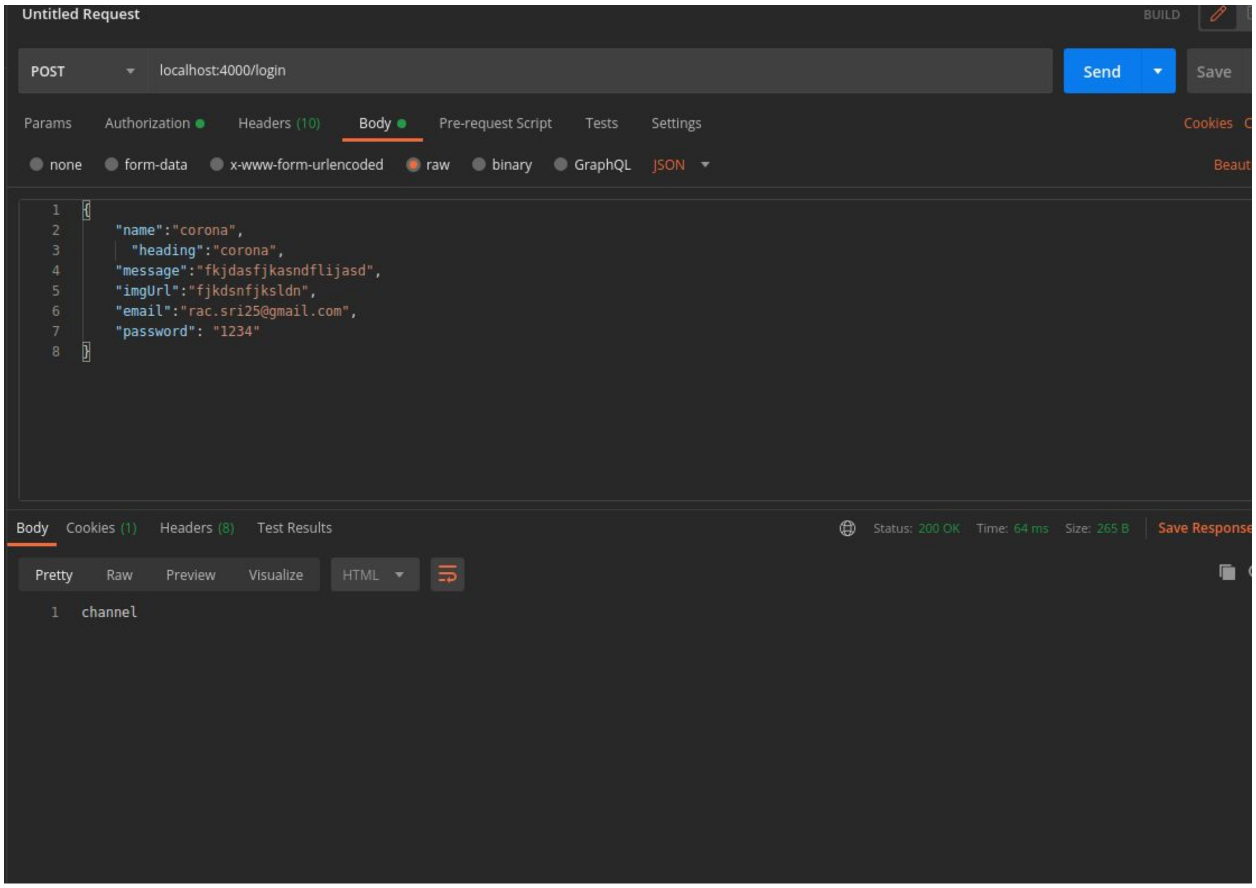
POST localhost:4000/register

```
{
    "name":"corona",
      "heading":"corona",
    "message":"fkjdasfjkasndflijasd",
    "imgUrl":"fjkdsnfjksldn",
    "email":"rac.sri25@gmail.com",
    "password": "1234"
}
```

Status: 200 OK Time: 44 ms Size: 265 B

```
channel
```

Test Case	User Login
Test Case Summary	User enters valid credentials.
Pre-requisites	The user must have connected to internet and opened the website.
Test-Data	email: <value> password: <value>
Expected Result	User logs in successfully..
Status	PASS
Executed By	Chirag
Date of Execution	4/12/2020
Test Environment	Laptop
Designed by	Group 2



POST localhost:4000/login

```
{
    "name":"corona",
      "heading":"corona",
    "message":"fkjdasfjkasndflijasd",
    "imgUrl":"fjkdsnfjksldn",
    "email":"rac.sri25@gmail.com",
    "password": "1234"
}
```

Status: 200 OK Time: 64 ms Size: 265 B

```
channel
```