

LAB REPORT

Submitted by

Tushar Sharma [RA2011003010001]

Under the Guidance of

Dr. Jayakumar L

Associate professor, Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING



SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND

TECHNOLOGY KATTANKULATHUR – 603 203

JUNE 2022



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203**

BONAFIDE CERTIFICATE

Certified that this lab report titled **OrgBuzz- Chat Application** is the bonafide work done by Tushar Sharma (RA2011003010001) who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Dr. Jayakumar L

SEPM – Course Faculty

Associate Professor

Department of Computing Technologies

ABSTRACT

OrgBuzz is a Web Based Chat Application which is directed towards targeting small commercial networks for ease of communication for all employees registered to that particular domain. In this digital age, chat application are surging in popularity.

Our project aims to deliver a simple chat application with all the basic features, all wrapped up in a minimalist UI for ease of use. With future releases, it can even be integrated into already existing websites and be expanded beyond its current mission and vision

This is a Full Stack Project featuring front-end, back-end and middle-ware and has been hardcoded from scratch. In the following pages, we have listed all the necessary documentation pertaining to our project as well details its scope and current development cycle.

TABLE OF CONTENTS

CHAPTER NO	TI TL E	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
1	PROBLEM STATEMENT	
2	STAKEHOLDERS & PROCESS MODELS	
3	IDENTIFYING REQUIREMENTS	
4	PROJECT PLAN & EFFORT	
5	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	
7	ENTITY RELATIONSHIP DIAGRAM	
8	DATA FLOW DIAGRAM	
9	SEQUENCE & COLLABORATION DIAGRAM	
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	
11	TEST CASES	
12	MANUAL TEST CASE REPORTING	
13	IMPLEMENTATION	

CONCLUSION

REFERENCES

APPENDIX (CODE)

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1	PROBLEM STATEMENT	
2	STAKEHOLDERS & PROCESS MODELS	
3	IDENTIFYING REQUIREMENTS	
4	PROJECT PLAN & EFFORT	
5	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	
7	ENTITY RELATIONSHIP DIAGRAM	
8	DATA FLOW DIAGRAM	
9	SEQUENCE & COLLABORATION DIAGRAM	
10	DEVELOPMENT OF TESTING FRAMEWORK /USER INTERFACE	
11	TEST CASES	
12	MANUAL TEST CASE REPORTING	
13	ARCHITECTURE/DESIGN/Framework/ IMPLEMENTATION	



Department of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	To identify the Software Project, Create Business Case, Arrive at a Problem Statement
Name of the candidate	Tushar Sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the

Project Title: *OrgBuzz* (Web Based Chat Application)

Project Description:

In this digital age, chat applications are surging in popularity. OrgBuzz is a Web Based Chat Application. It is a small scale chat application centered around targeting small communication networks and delivering an appealing UI. It enables secure and open messaging to any user registered to the network. When further worked on, it can even be integrated into already existing sites.

For the implementation we will be using MongoDB for the backend and React.js for the front end. In addition to this Express and Node.js will also be used to provide a seamless robust experience. Socket.io is a JavaScript library which enables real-time communication between web clients and web servers and will also be used to implement the project.

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- This project is centered around providing a simple chat interface with an appealing UI and targets smaller communication networks.
- Users can sign in and converse with anyone who has registered without any hassle, which is in contrary to modern day chat applications where the user has to send a friend request or know some other parameter to converse with someone.
- This chat app aims to target smaller networks such as workplaces where the user can have a clean, consolidated chat app and have easy access to contact any fellow colleague.

THE HISTORY

In bullet points, describe the current situation.

- Messaging apps are surging in popularity. These past few years have brought a lot of chat apps.
- People prefer chat-based applications as they add a hint of real time interaction. A personal touch gets added to the mix.
- There are many chat applications but only few are geared towards the corporate environment. Web based chat applications can be integrated into sites to improve interaction with customers or open lines of communication can be made between members of a workplace.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- Since this project heavily revolves around Web Development, the hassles will come as skill hassles not physical limitations.
- One will have to be well acquainted with Full Stack Development to make this project a success. So lacking a skill set versed to this will be a problem.
- Also a considerable amount of time will have to be dedicated to making this project along with picking up the required skills.

APPROACH

List what is needed to complete the project.

- This is purely a code driven project involving MERN stack (MongoDB, Express, React JS, Node.js) and Socket.io .
- A deployment service such as Vercel or Heroku will also be required for hosting the Chat Application.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- To an organization, this project will bring ease of communication. Open communication lines will be drawn between colleagues so facilitate a smooth functioning work environment. The hassles of sending friend requests and knowing numbers will be not be a hindrance.
- This will also reduce the number of unproductive meets as the employees can ask any amount of doubts directly to one and other.
- Being a web application, users need not download any application but can access the chat application through a browser.

Result

Thus, the project team formed, the project is described, the business case was prepared and the problem statement was arrived.



Department of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Identification of Process Methodology and Stakeholder Description
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Team Members:

Sl No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010025	Adithya Anup
3	RA2011003010001	Tushar Sharma

Project Title:

Selection of Methodology

We will be adopting the Agile methodology for this project. The Agile Methodology is a practice which that promotes continuous iteration of testing and development throughout the lifecycle of the project. It involves constant collaboration with stakeholders and continuous improvement at each and every stage. The project will be managed by breaking it into multiple phases such as Finance, UI/UX, Frontend, Backend etc.. The team will cycle through the process of planning, executing and evaluating always.

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Owner	Supply capital or equity for the application and controls proceedings	High	High	1
Team Members	Contributing to project objectives	High	Med	2
Project Manager	Planning, directing and organizing the team members.	High	High	2
End Users	Provides feedback and utilizes platform	High	Low	3

Stakeholder Related Questions

Who is affected positively and negatively by the project?

The corporate networks would be benefitted with this project whereas the current competitors would be impacted negatively. In the end such competition will inspire growth in this particular segment and inculcate further growth.

Who has the power to make it succeed (or fail)?

The employees of the organization who would be using this chat application has the power to make it succeed along with the developers who will ensure a feature rich and relatively bug free experience.

Who makes the decisions about money?

The financial department of the company would be the ones making decisions about money. They will make budget estimates and will track the revenue stream, thereby giving the other departments the best room to work with in terms of expense.

Who are the suppliers?

The potential suppliers could be the services for hosting chat application servers such as:- Heroku and Vercel.

Who are the end users?

The end users will be the organizations who would like a platform for communication in the corporate workspace.

Who has influence over other stakeholders?

The head or the owner of the application will be the primary influence over other stakeholders as the decisions made by him will govern the direction of the application.

Who could solve potential problems with the project?

The domains could potentially solve the problems that would come under their field of expertise. By breaking down the problem we believe that any issue can be dealt with swiftly and in the best possible manner.

Who is in charge of assigning or procuring resources or facilities?

The project manager of the respective domains would be in charge of assigning and procuring resources on request from the concerned departments

Who has specialist skills which are crucial to the project?

Each individual will play an important role in the success of the project.
But as a whole, the backend developers will be the backbone of the project as they are responsible for the functioning of the chat application and for handling and linking the servers.

Result

Thus the Project Methodology was identified and the stakeholders were describe



Department Of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	System, Functional and Non-Functional Requirements of the Project
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To identify the system, functional and non-functional requirements for the project.

Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Anup

Project Title: OrgBuzz

System Requirements

- The OrgBuzz chat app is a workplace oriented chat web application to allow peer to peer communication with everyone on the network
- Anyone with a common domain name as the user can be messaged without any hindrance such as friend requests etc.
- Constraints such as not allowing large video files and high resolution images along with GIFs will be implemented as it is a work oriented network.
- The Chat Application as a whole wouldn't be taxing on the system and will just need a stable internet connection and web browser so as to prevent unnecessary file installations.

Functional Requirements

- Peer to Peer messaging lines will be established.
- A clean and elegant UI for easy navigation and use
- Messages will be end to end encrypted so as to prevent data leakage and enhance security
- In case of spamming and unlawful conduct, a report provision will be available. On reporting the user, he/she will receive a warning message stating that three reports and the user will be blocked from logging into the application by having their credentials revoked.
- In case of a server crash or any similar application failure, users will be able to access previous chats but just wont be able to send new messages.
- The web application will be responsive for all devices.

Non-Functional Requirements

- The application will be available 24x7 with the exception of downtime for maintenance and updates.
- Application downtime for updates and fixes will be informed in advance. Downtime will be from 12 a.m to 5 a.m and won't exceed it.
- Permission to use the service will be tied to a nominal fee which will be fixed in due time.
- An administrator will preside over the chat application and will be in touch with the development team in case of any issues.

Result

Thus the requirements were identified and accordingly described.



Department of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	4
Title of Experiment	Prepare Project Plan based on scope, Calculate Project effort based on resources and Job roles and responsibilities
Name of the candidate	Tushar sahrma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Prepare Project Plan based on scope, Calculate Project effort based on resources,
Find Job roles and responsibilities

Team Members:

Sl No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Anup

1. Executive Summary

OrgBuzz is a Web Based Chat Application. It is a small scale chat application centered around targeting small communication networks and delivering an appealing UI. It enables secure and open messaging to any user registered to the network. When further worked on, it can even be integrated into already existing sites.

2. Project Management Plan

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Change Management (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control

Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Identifying, Analyzing, Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]
Risk Management	Identifying, analysing, and prioritizing project risks
Procurement Management	Adhering to organization procurement process

1. Estimation

1.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the UI and Backend Research and Development Data Analytics	E1R1A1T1 (Effort-Requirement-Activity-Task)	To create a desktop-level website using MERN stack	10	Full Stack-50K Frontend-50K Backend-45K
	E1R1A1T2	Evaluation of software technology trends and incorporating them with updates and patches	4	For Research - 60K For Rolling out patches/updates – 50K
	E1R1A1T3	Administrator of Data, Database Management,	10 with shifts	Data Analyst- 100K

		Evaluating trends in data		Database Manager-80K
Customer Care	E1R1A1T4	Customer care	24 with shifts	Customer Care-12K to 15K
Application maintenance	E1R1A1T6	Advertisements, Public Relations	8-10 On requirement	60k
Marketing				

Effort (hr)	Cost (INR)
1	500

1.2. Infrastructure/Resource Cost

Infrastructure Requirement	Qty	Cost per qty	Cost per item
IR1 : Domain rights	1	2 – 5k	2 – 5k
IR2 : Server firm and other hardware	1	150k	150k
IR3 : Legal and other government documents	1	200k	200k

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support Consultant	3	2,000,000	6,000,000
License	Operating System Database Middleware IDE	10	10000	100,000
Infrastructures	Server, Storage and Network	20	20000	400,000

2. Project Team Formation

2.1. Identification Team members

Name	Role	Responsibilities
------	------	------------------

Kevin	Key Business User (Product Owner)	Provide clear business and user requirements
Adithya	Project Manager	Manage the project
Tushar	Business Analyst	Discuss and Document Requirements
Kevin	Technical Lead	Design the overall implementation using MERN
Tushar	UX Designer	Design the UI using Figma
Adithya	Frontend Developer	Develop user interface
Kevin	Backend Developer	Design, Develop and Unit Test Services/API/DB
Tushar	Tester	Performance Testing

2.2. Responsibility Assignment Matrix

RACI Matrix	Team Members			
Activity	Name (BA)	Name (Developer)	Name (Project Manager)	Key Business User
User Requirement Documentation	A	C/I	I	R

A	Accountable
R	Responsible
C	Consult
I	Inform

Result:

Thus, the Project Plan was documented successfully.



Department of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Prepare Work breakdown structure, Timeline chart, Risk identification table
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table

Team Members:

SI No	Register No	Name
1	RA2011003010018	KEVIN THOMAS KOSHY
2	RA2011003010025	ADITHYA ANUP
3	RA2011003010001	TUSHAR SHARMA

WBS With Project Schedule

Module(#)	Activity(#)	Assignee	Planned Start Date	Planned End Date	Actual Start Date	Actual End Date	Status
M1	UI/UX design	Tushar Sharma	15-03-2022	1-04-2022	16--03-2022	3-04-2022	Completed
M2	Front End	Adithya Anup	05-04-2022	20-04-2022	05-04-2022	24-04-2022	Completed
M3	Back End	Kevin Thomas Koshy	25-04-2022	20-05-2022	27-04-2022	TBD	Ongoing
M4	Project Manager	Adithya Anup	15-03-2022	30-05-2022	14-03-2022	TBD	Ongoing
M5	Testing	Tushar Sharma	23-05-2022	30-05-2022	TBD	TBD	Yet to Start

Risk Identification :

- Structured Brainstorming with Team and Stakeholders
- Checklist is a list of actions/points to be considered [Information can be used from the similar previous projects]
- Risk can be identified from
 - Assumption-Constraint analysis
 - SWOT Analysis [Strength/Weakness/Opportunity/Threat]

List (Describe) Register

Risk ID (#)	Risk Description	Impact Description
RO1	Software error	Can't redirect pages. Database Connectivity issues.
RO2	Personal Error	Color Palette Decisions and Hosting Problems
RO3	Usage may be decreased	Loss

Managing Risk

Risk ID (#)	Status [open/closed]	Risk Appetite[accept/mitigate/Transfer/Avoid	Action	Action Owner	Target Date	Remarks
RO1	open	Mitigate	upgrading	-	-	-
RO2	open	Avoid	Messaging error	-	-	-
RO3	open	Avoid	-	-	-	-

WBS

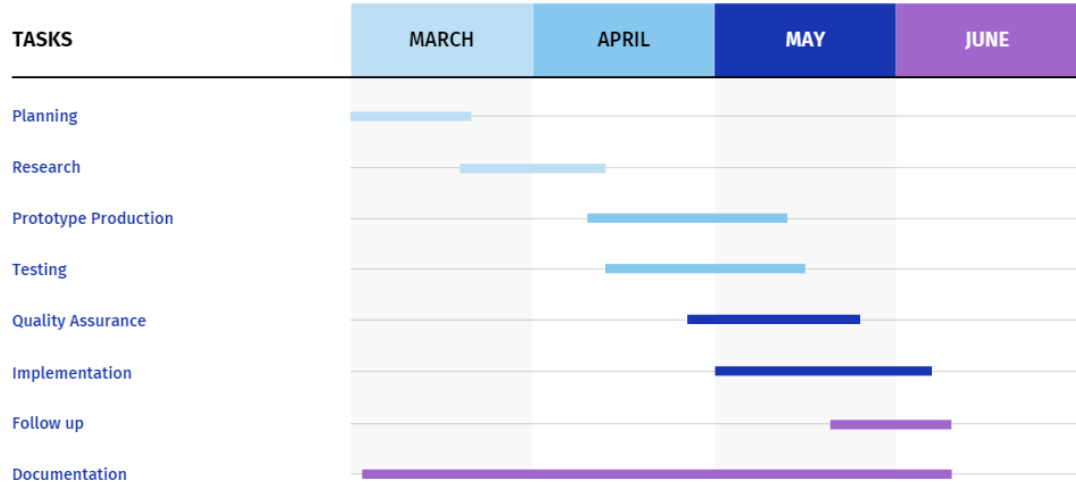
- ☐ 0.0 Chat Web Site
- ☐ 1.0 Project Management
- ☐ 2.0 Requirements Gathering
- ☐ 3.0 Analysis & Design
- ☐ 4.0 Site Software Development
 - 4.1 HTML Design and Creation
 - 4.2 Backend Software
 - 4.2.1 Database Implementation
 - 4.2.2 Middleware Development
 - 4.2.3 Security Subsystems
 - 4.2.4 Chat Interface
 - 4.2.5 Message Receive/Send
 - 4.3 Graphics and Interface
 - 4.4 Content Creation
- ☐ 5.0 Testing and Production

TIMELINE – GANTT CHART



OrgBuzz

GANTT CHART



RISK ANALYSIS – SWOT

STRENGTHS

- Expertise
- Reputation
- Cost
- Project Management
- Speed of Delivery
- Easy to Use
- Quality
- Communication
- Low risk processes
- Procurement and supplier management
- People

WEAKNESSES

- Expertise
- Reputation
- Cost
- Project Management
- Speed of Delivery
- Easy to Use
- Quality
- Communication
- Low risk processes
- Procurement and supplier management
- People

OPPORTUNITIES

- Enter new market
- Develop new product
- Reduce costs
- Add new clients
- Expand to a new geographical area
- Launch marketing campaign
- Hire new people
- Open new stores/locations
- Purchase competitor
- Increase quality of product
- Reduce risk
- Increase time of delivery
- Raise morale

THREATS

- New competitor
- Existing competitor expands / creates new product
- Increasing supplier cost
- Decreasing market demand
- Production disruption (breakdown, etc.)
- Increasing customer dissatisfaction
- Decreasing quality of work
- Acts of God
- Loss of key people to competitors
- Systems / products becoming obsolete
- Bankruptcy of supplier
- Client dissatisfaction
- Customer inability to pay

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.



Department of Networking and Communications

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	Design a System Architecture, Use Case and Class Diagram
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

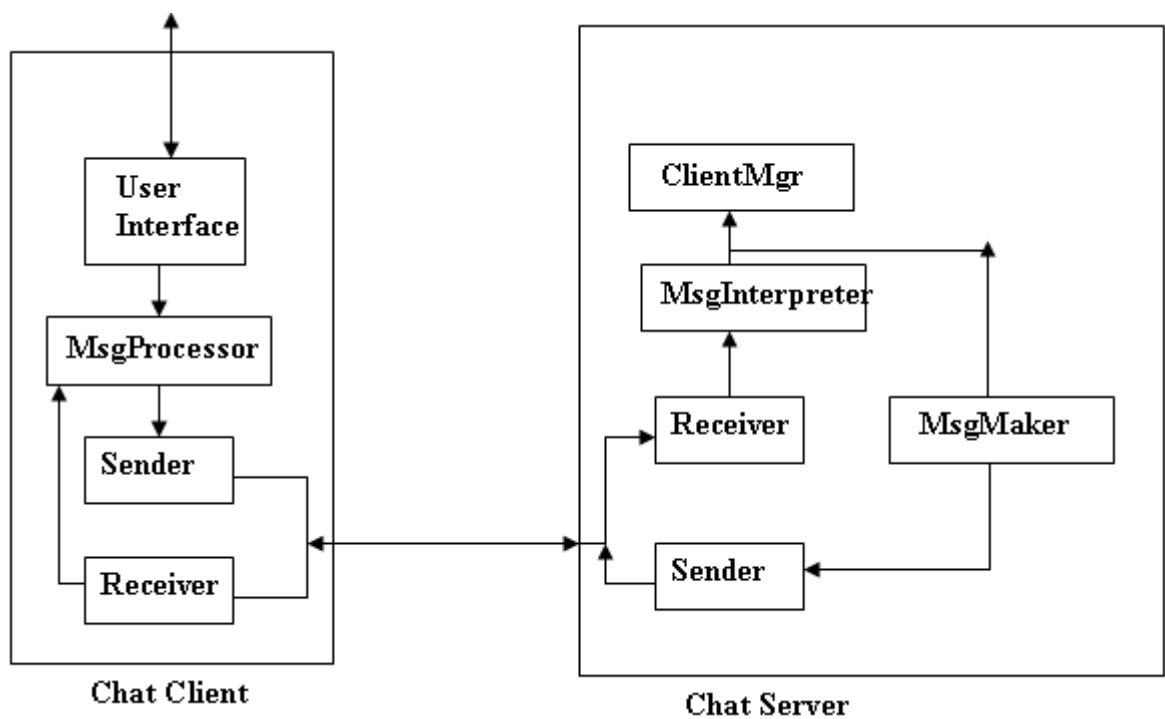
Aim

To Design a System Architecture, Use case and Class Diagram

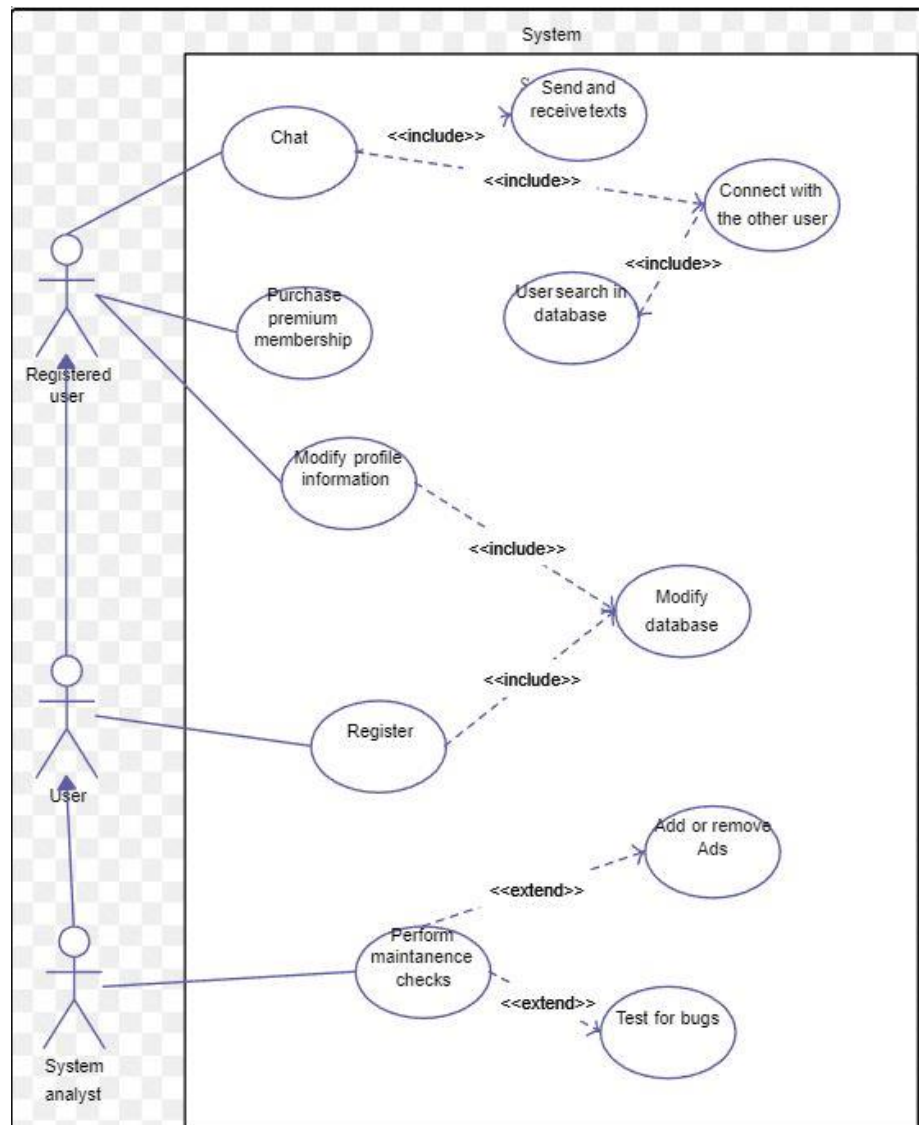
Team Members:

Sl No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Annp

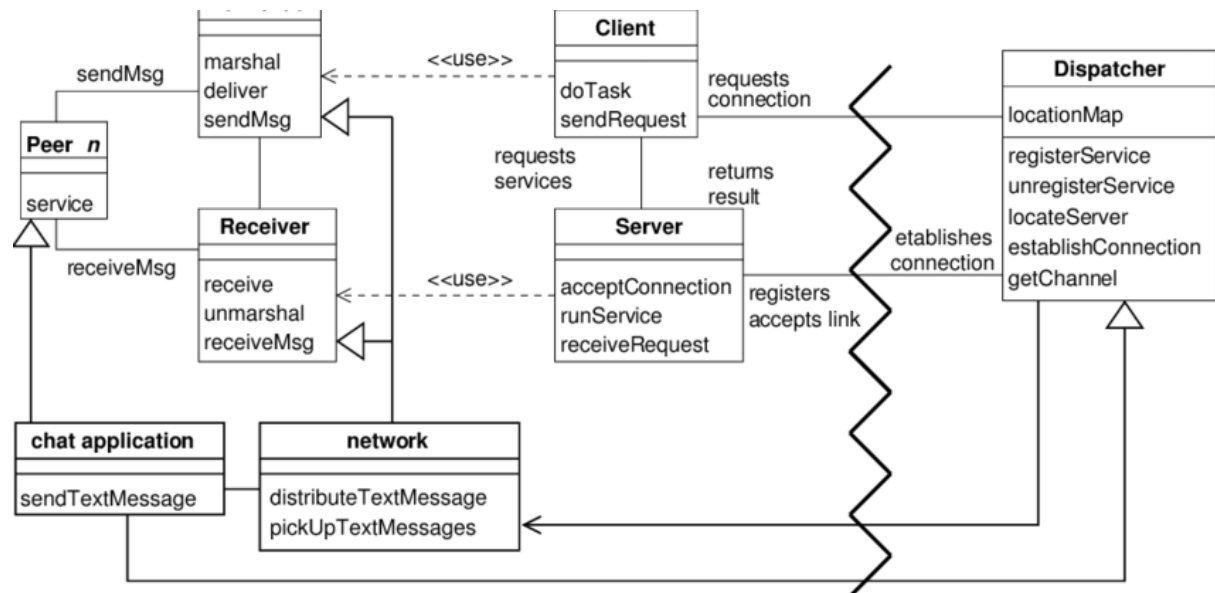
SYSTEM ARCHITECTURE



USE CASE DIAGRAM



CLASS DIAGRAM



Result:

Thus, the system architecture, use case and class diagram created successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	Design a Entity relationship diagram
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

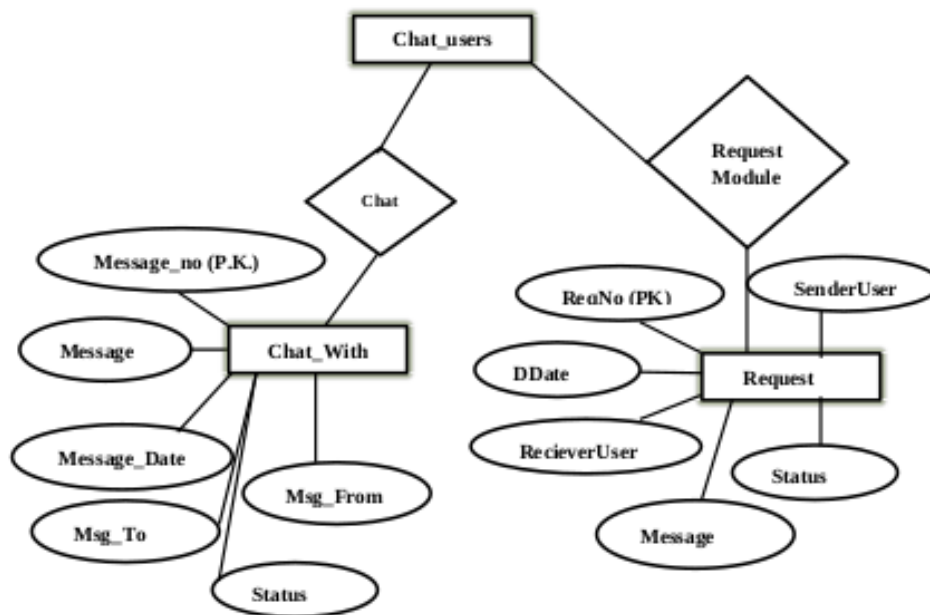
Aim

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Anup

ER Diagram of University Database



Result:

Thus, the entity relationship diagram was created successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	Develop a Data Flow Diagram (Process-Up to Level 1)
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the data flow diagram up to level 1 for OrgBuzz

Team Members:

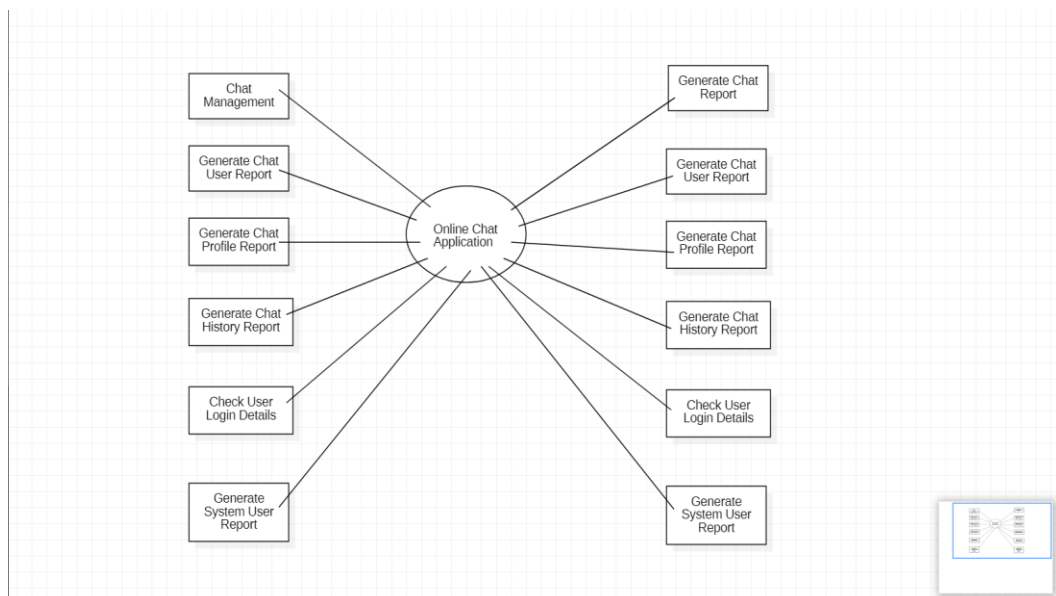
S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010025	Adithya Anup
3	RA2011003010001	Tushar Sharma

Data Flow Diagram

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles).

The data flow diagram enables us to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

Given below is the Data Flow Diagram which we have made for our chat application. It gives us knowledge about the flow of data and its basic functional layout



Result:

Thus, the data flow diagrams have been created for OrgBuzz



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	Design a Sequence and Collaboration Diagram
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

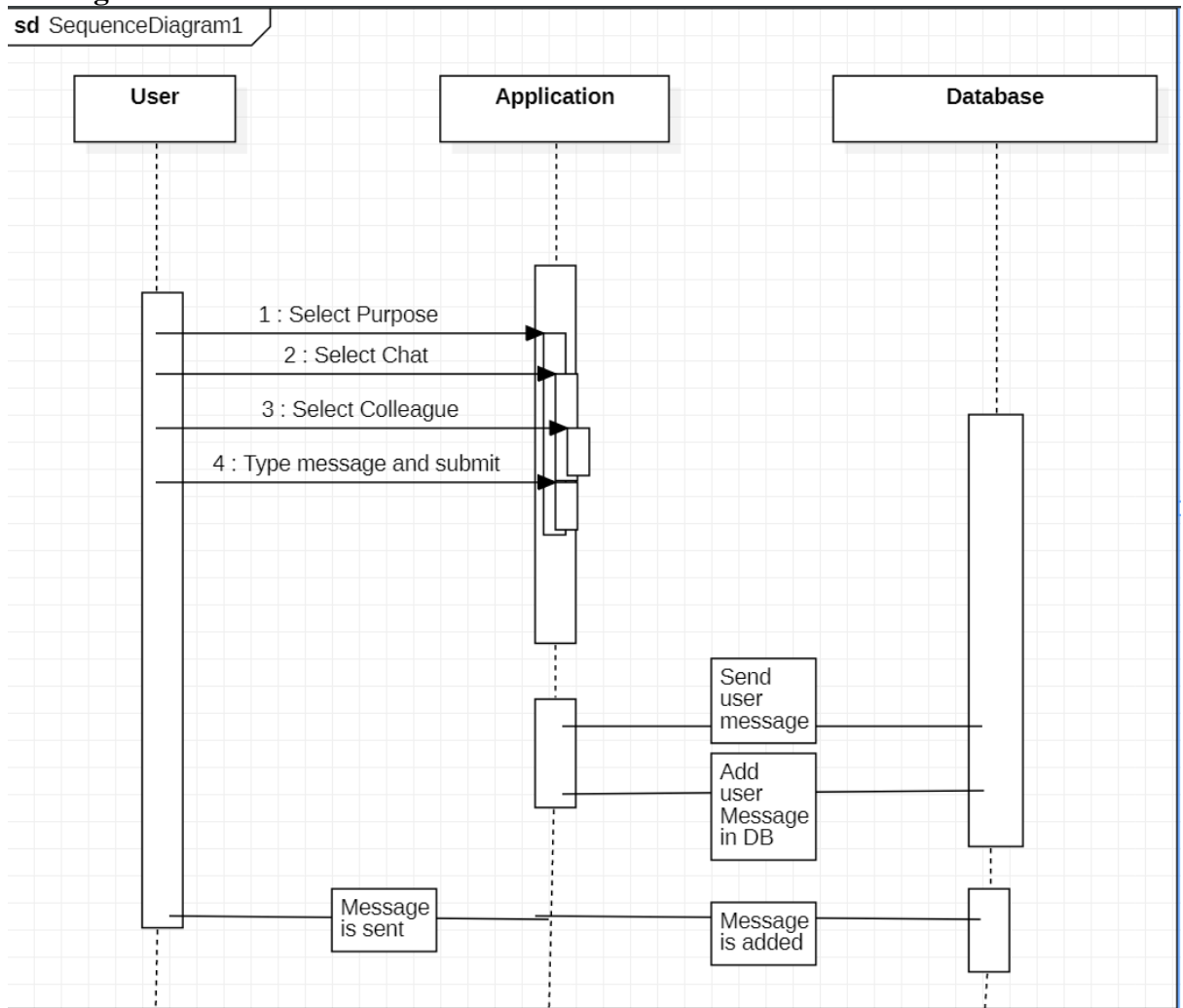
Aim

To create the sequence and collaboration diagram for the <project name>

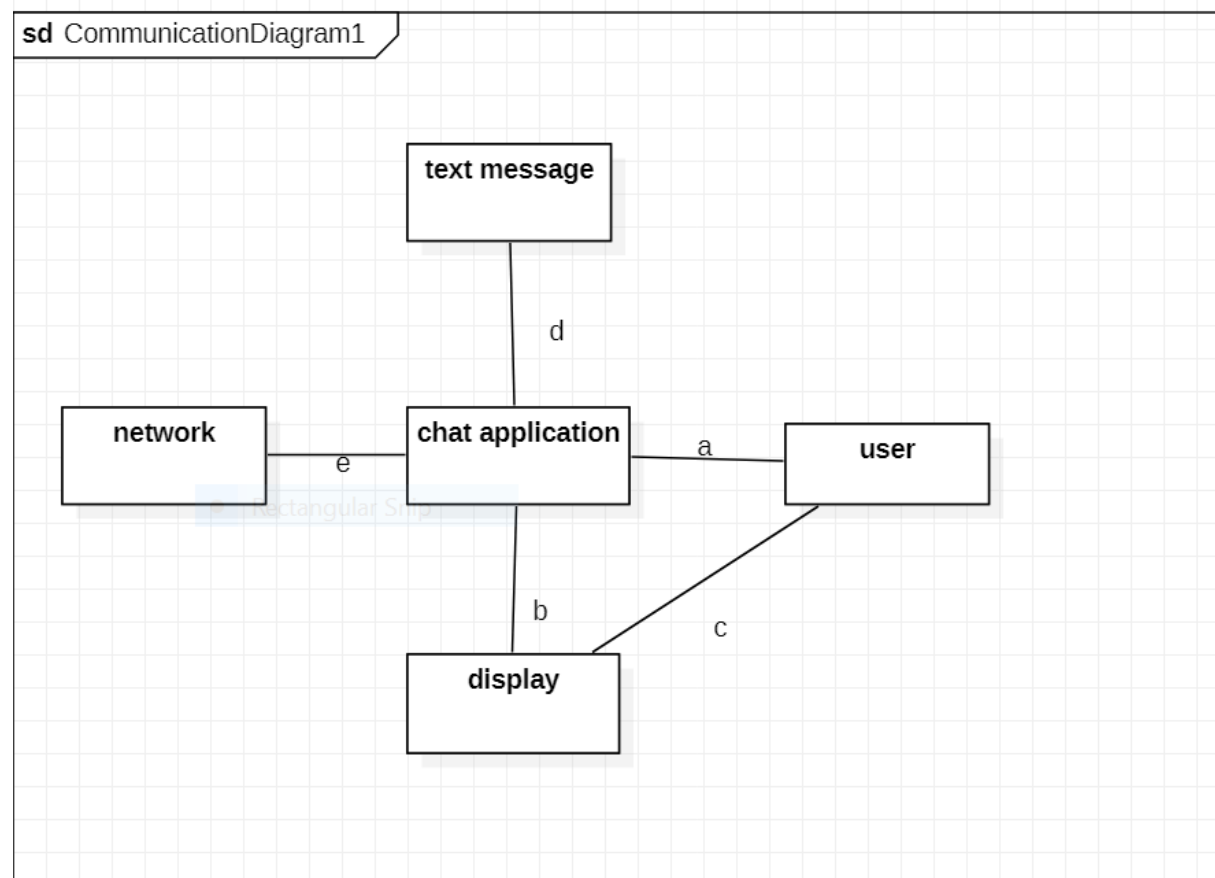
Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010025	Adithya Anup
3	RA2011003010001	Tushar Sharma

Sequence Diagram



Collaboration Diagram



Result:

Thus, the sequence and collaboration diagrams were created for the chat application.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	Develop a Testing Framework/User Interface
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the testing framework and/or user interface framework for the **OrgBuzz Chat Application**

Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Anup

Objective:

To implement the designed User Interface and test out the transitions without integrating the backend

Scope:

Successful integration of the User Interface will serve as the corner stone for setting up the back-end routes for the chat application. The page transitions will be noted and the flow can also be verified.

Approach:

The design for the User Interface was created using Figma. It will be implemented using Javascript and Express.js with the help of CSS for a bit of styling. The application has been configured primarily for PC/laptop use so the design will be adhering to that.

Test Plan

As mentioned under the approach, OrgBuzz is primarily meant for PC/Laptop use so the testing will be done only those platforms. We will test the look of the application on various devices and ensure modular functionality of the front-end. We will first be implementing the front-end using HTML and CSS to test the basic functioning and then move forward to implementing it using Javascript frameworks.

The backend will be worked up using MongoDB but the primary task at end is setting up the user routes and the flow of the site.

Result:

Thus, the testing framework/user interface framework has been created for OrgBuzz



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	Test Cases
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To develop the test cases manual for the chat application system.

Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010025	Adithya Anup
3	RA2011003010001	Tushar Sharma

Test Cases-

Test Scenario	Test Case	Expected Outcome	Actual Outcome	Status	Remarks
Check if username and password are acceptable	Enter valid username and password	User should be taken to the home page for chatting	User goes to a page where there would be several colleagues to chat with	Pass	Success
Check if minimum two users are available for chatting	If only 1 user available	Error displayed, instructing user to chat with two or more users	Error displayed	Pass	Success
Verify that a minimum of two devices should be available	If two or more devices are available	Chatting can be done successfully	User will be able to chat with other colleagues without any hindrance	Pass	Success
Seamless Message Distinction between multiple contacts	Two or more incoming messages at the same time	The incoming messages will be well organized and properly sorted	Chats are properly organized and clarity is present	Pass	Success

Result:

Thus, the test case manual has been created for chat ap



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Manual Test Case Reporting
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To prepare the manual test case report for the chat application system.

Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010025	Adithya Anup
3	RA2011003010001	Tushar Sharma

Current Status of the Test report

- Frontend and Backend have been linked and tested successfully

Present obstacles to proceed further

- Website has to be hosted

Seek help from stakeholders to remove obstacles/constraints

- Programmers/Team Members are working on getting the website hosted

Category	Progress Against Plan	Status
Functional Testing	Amber	In-Progress
Non-Functional Testing	Green	Completed

Functional	Test Case Coverage (%)	Status
Website Design	95%	Completed
Database storage	90%	Completed
Message Transfer	90%	Completed
Website Hosting	60%	In-Progress

Result:

Thus, the test case report has been created for the chat application system.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	13
Title of Experiment	Provide the details of Architecture Design/Framework/Implementation
Name of the candidate	Tushar sharma
Team Members	
Register Number	RA2011003010001
Date of Experiment	

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
Total		10	

Staff Signature with date

Aim

To provide the details of architectural design/framework/implementation

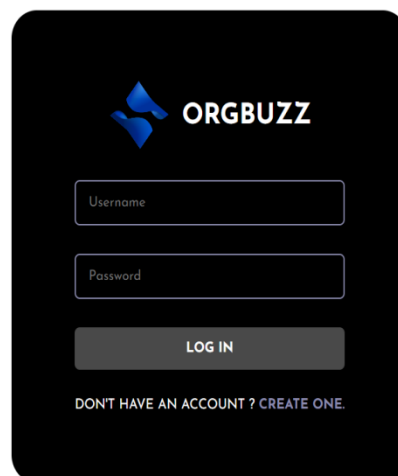
Team Members:

S No	Register No	Name
1	RA2011003010018	Kevin Thomas Koshy
2	RA2011003010001	Tushar Sharma
3	RA2011003010025	Adithya Anup

Documentation

We will be going through some few modules of our chat application “OrgBuzz” which was designed using MERN stack. Screenshots have also been attached to help better visualize the modules. At the end we have included a screenshot of most of the modules combined together and working in sync.

Login Page



```
import React, { useState, useEffect } from "react";
import axios from "axios";
import styled from "styled-components";
import { useNavigate, Link } from "react-router-dom";
import Logo from "../assets/logo.svg";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import { loginRoute } from "../utils/APIRoutes";
```



```

export default function Login() {
  const navigate = useNavigate();
  const [values, setValues] = useState({ username: "", password: "" });
  const toastOptions = {
    position: "bottom-right",
    autoClose: 8000,
    pauseOnHover: true,
    draggable: true,
    theme: "dark",
  };
  useEffect(() => {
    if (localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)) {
      navigate("/");
    }
  }, []);
  const handleChange = (event) => {
    setValues({ ...values, [event.target.name]: event.target.value });
  };
  const validateForm = () => {
    const { username, password } = values;
    if (username === "") {
      toast.error("Email and Password is required.", toastOptions);
      return false;
    } else if (password === "") {
      toast.error("Email and Password is required.", toastOptions);
      return false;
    }
    return true;
  };
  const handleSubmit = async (event) => {
    event.preventDefault();
    if (validateForm()) {
      const { username, password } = values;
      const { data } = await axios.post(loginRoute, {
        username,
        password,
      });
      if (data.status === false) {
        toast.error(data.msg, toastOptions);
      }
      if (data.status === true) {
        localStorage.setItem(
          process.env.REACT_APP_LOCALHOST_KEY,
          JSON.stringify(data.user)
        );
        navigate("/");
      }
    }
  };
  return (
    <>
      <FormContainer>
        <form action="" onSubmit={(event) => handleSubmit(event)}>
          <div className="brand">
            <img src={Logo} alt="logo" />
            <h1>OrgBuzz</h1>
          </div>
          <input
            type="text"
            placeholder="Username"
            name="username"
            onChange={(e) => handleChange(e)}
            min="3"
          />
          <input
            type="password"
            placeholder="Password"
            name="password"
            onChange={(e) => handleChange(e)}
          />
          <button type="submit">Log In</button>
          <span>
            Don't have an account ? <Link to="/register">Create One.</Link>
          </span>
        </form>
      </>
    </>
  );
}

```

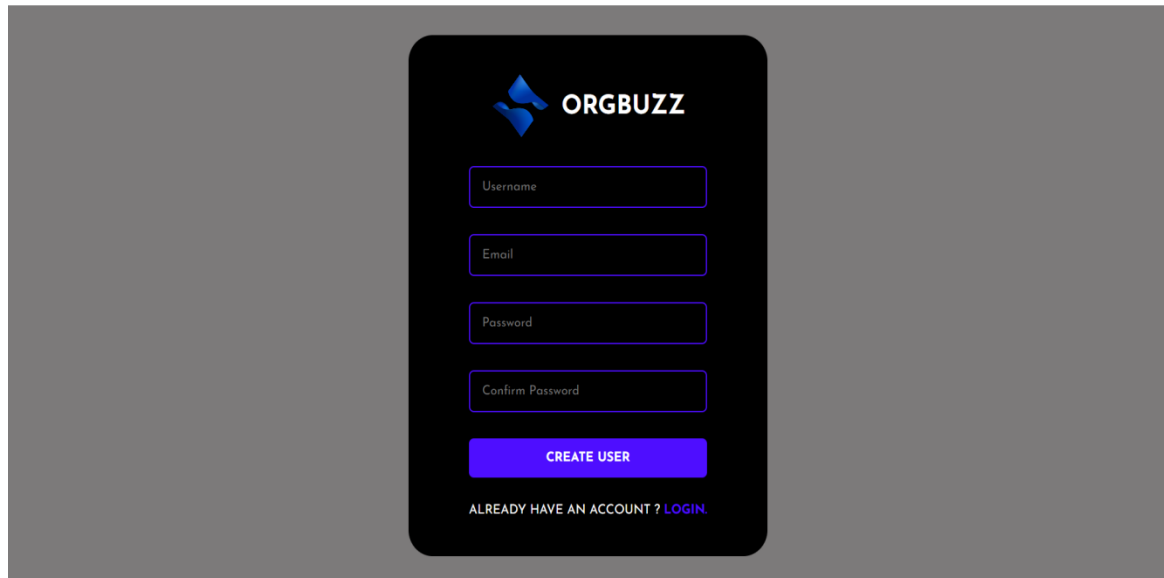
```

        </form>
      </FormContainer>
    <ToastContainer />
  </>
);
}
const FormContainer = styled.div`
  height: 100vh;
  width: 100vw;
  display: flex;
  flex-direction: column;
  justify-content: center;
  gap: 1rem;
  align-items: center;
  background-color: #FFFFFF;
  .brand {
    display: flex;
    align-items: center;
    gap: 1rem;
    justify-content: center;
    img {
      height: 5rem;
    }
    h1 {
      color: white;
      text-transform: uppercase;
    }
  }
  form {
    display: flex;
    flex-direction: column;
    gap: 2rem;
    background-color: #000000;
    border-radius: 2rem;
    padding: 5rem;
  }
  input {
    background-color: transparent;
    padding: 1rem;
    border: 0.1rem solid #8B8BAE;
    border-radius: 0.4rem;
    color: white;
    width: 100%;
    font-size: 1rem;
    &:focus {
      border: 0.1rem solid #8B8BAE;
      outline: none;
    }
  }
  button {
    background-color: #494949;
    color: white;
    padding: 1rem 2rem;
    border: none;
    font-weight: bold;
    cursor: pointer;
    border-radius: 0.4rem;
    font-size: 1rem;
    text-transform: uppercase;
    &:hover {
      background-color: #8B8BAE;
    }
  }
  span {
    color: white;
    text-transform: uppercase;
    a {
      color: #8B8BAE;
      text-decoration: none;
      font-weight: bold;
    }
  }
`;

```

We decided to go for a login page with a minimalist look and a relative simple color palette while retaining the key features required for a login page. It is welcoming and serves an entry point to our chat application.

Registration Page



```
import React, { useState, useEffect } from "react";
import axios from "axios";
import styled from "styled-components";
import { useNavigate, Link } from "react-router-dom";
import Logo from "../assets/logo.svg";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import { registerRoute } from "../utils/APIRoutes";

export default function Register() {
  const navigate = useNavigate();
  const toastOptions = {
    position: "bottom-right",
    autoClose: 8000,
    pauseOnHover: true,
    draggable: true,
    theme: "dark",
  };

  const [values, setValues] = useState({
    username: "",
    email: "",
    password: "",
    confirmPassword: "",
  });

  useEffect(() => {
    if (localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)) {
      navigate("/");
    }
  }, []);

  const handleChange = (event) => {
    setValues({ ...values, [event.target.name]: event.target.value });
  };

  const handleValidation = () => {
    const { password, confirmPassword, username, email } = values;
    if (password !== confirmPassword) {
      toast.error(
```

```

        "Password and confirm password should be same.",
        toastOptions
    );
    return false;
} else if (username.length < 3) {
    toast.error(
        "Username should be greater than 3 characters.",
        toastOptions
    );
    return false;
} else if (password.length < 8) {
    toast.error(
        "Password should be equal or greater than 8 characters.",
        toastOptions
    );
    return false;
} else if (email === "") {
    toast.error("Email is required.", toastOptions);
    return false;
}
return true;
});

const handleSubmit = async (event) => {
    event.preventDefault();
    if (handleValidation()) {
        const { email, username, password } = values;
        const { data } = await axios.post(registerRoute, {
            username,
            email,
            password,
        });
        if (data.status === false) {
            toast.error(data.msg, toastOptions);
        }
        if (data.status === true) {
            localStorage.setItem(
                process.env.REACT_APP_LOCALHOST_KEY,
                JSON.stringify(data.user)
            );
            navigate("/");
        }
    }
};

return (
    <>
        <FormContainer>
            <form action="" onSubmit={(event) => handleSubmit(event)}>
                <div className="brand">
                    <img src={Logo} alt="logo" />
                    <h1>orgbuzz</h1>
                </div>
                <input
                    type="text"
                    placeholder="Username"
                    name="username"
                    onChange={(e) => handleChange(e)}
                />
                <input
                    type="email"
                    placeholder="Email"
                    name="email"
                    onChange={(e) => handleChange(e)}
                />
                <input
                    type="password"
                    placeholder="Password"
                    name="password"
                    onChange={(e) => handleChange(e)}
                />
                <input
                    type="password"
                    placeholder="Confirm Password"
                    name="confirmPassword"
                    onChange={(e) => handleChange(e)}
                />
            </form>
        </FormContainer>
    </>
);

```

```

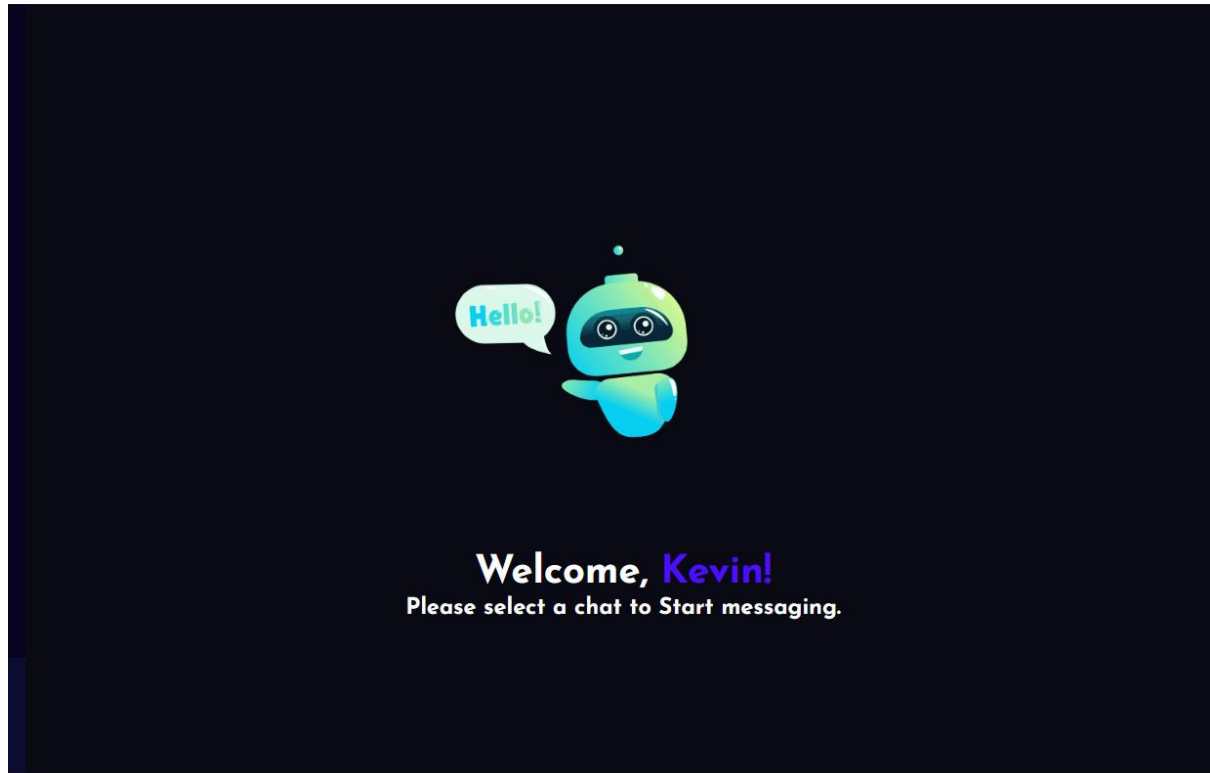
        />
        <button type="submit">Create User</button>
        <span>
            Already have an account ? <Link to="/login">Login.</Link>
        </span>
    </form>
</FormContainer>
<ToastContainer />
</>
);
}
const FormContainer = styled.div`
height: 100vh;
width: 100vw;
display: flex;
flex-direction: column;
justify-content: center;
gap: 1rem;
align-items: center;
background-color: #7C7A7A;
.brand {
    display: flex;
    align-items: center;
    gap: 1rem;
    justify-content: center;
    img {
        height: 5rem;
    }
    h1 {
        color: white;
        text-transform: uppercase;
    }
}
form {
    display: flex;
    flex-direction: column;
    gap: 2rem;
    background-color: #000000;
    border-radius: 2rem;
    padding: 3rem 5rem;
}
input {
    background-color: transparent;
    padding: 1rem;
    border: 0.1rem solid #4e0eff;
    border-radius: 0.4rem;
    color: white;
    width: 100%;
    font-size: 1rem;
    &:focus {
        border: 0.1rem solid #997af0;
        outline: none;
    }
}
button {
    background-color: #4e0eff;
    color: white;
    padding: 1rem 2rem;
    border: none;
    font-weight: bold;
    cursor: pointer;
    border-radius: 0.4rem;
    font-size: 1rem;
    text-transform: uppercase;
    &:hover {
        background-color: #4e0eff;
    }
}
span {
    color: white;
    text-transform: uppercase;
    a {
        color: #4e0eff;
        text-decoration: none;
    }
}

```

```
font-weight: bold;
}
}
;
```

As for the registration page, we have continued with a similar color palette and have given various fields to fill for registration. Stringent rules have been applied for password creation and password validation feature is also there before registering.

Welcome Page



```
import React, { useState, useEffect } from "react";
import styled from "styled-components";
import Robot from "../assets/robot.gif";
export default function Welcome() {
  const [userName, setUserName] = useState("");
  useEffect(async () => {
    setUserName(
      await JSON.parse(
        localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)
      ).username
    );
  }, []);
  return (
    <Container>
      <img src={Robot} alt="" />
      <h1>
        Welcome, <span>{userName}</span>
      </h1>
      <h3>Please select a chat to Start messaging.</h3>
    </Container>
  );
}

const Container = styled.div`
```

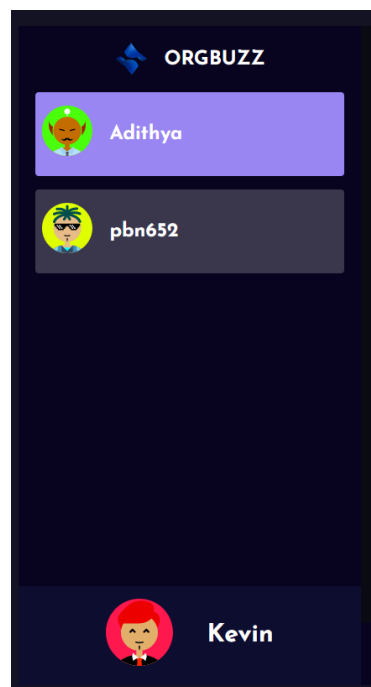
```

display: flex;
justify-content: center;
align-items: center;
color: white;
flex-direction: column;
img {
  height: 20rem;
}
span {
  color: #4e0eff;
}
};

```

On successfully logging in the user will be met with the welcome page which has been created using a blend of HTML, CSS, and JS and on selecting a particular chat this window will be replaced by the chat screen

Chats List



```

import React, { useState, useEffect } from "react";
import styled from "styled-components";
import Logo from "../assets/logo.svg";

export default function Contacts({ contacts, changeChat }) {
  const [currentUserName, setCurrentUserName] = useState(undefined);
  const [currentUserImage, setCurrentUserImage] = useState(undefined);
  const [currentSelected, setCurrentSelected] = useState(undefined);
  useEffect(async () => {
    const data = await JSON.parse(
      localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)
    );
    setCurrentUserName(data.username);
    setCurrentUserImage(data.avatarImage);
  }, []);
  const changeCurrentChat = (index, contact) => {
    setCurrentSelected(index);
    changeChat(contact);
  };
};

```

```

return (
  <>
    {currentUserImage && currentUserImage && (
      <Container>
        <div className="brand">
          <img src={Logo} alt="logo" />
          <h3>OrgBuzz</h3>
        </div>
        <div className="contacts">
          {contacts.map((contact, index) => {
            return (
              <div
                key={contact._id}
                className={`contact ${
                  index === currentSelected ? "selected" : ""
               }`}
                onClick={() => changeCurrentChat(index, contact)}
              >
                <div className="avatar">
                  <img
                    src={`data:image/svg+xml;base64,${contact.avatarImage}`}
                    alt=""
                  />
                </div>
                <div className="username">
                  <h3>{contact.username}</h3>
                </div>
              </div>
            );
          })}
        </div>
        <div className="current-user">
          <div className="avatar">
            <img
              src={`data:image/svg+xml;base64,${currentUserImage}`}
              alt="avatar"
            />
          </div>
          <div className="username">
            <h2>{currentUserName}</h2>
          </div>
        </div>
      </Container>
    )}
  </>
);
}
const Container = styled.div`
  display: grid;
  grid-template-rows: 10% 75% 15%;
  overflow: hidden;
  background-color: #080420;
  .brand {
    display: flex;
    align-items: center;
    gap: 1rem;
    justify-content: center;
    img {
      height: 2rem;
    }
    h3 {
      color: white;
      text-transform: uppercase;
    }
  }
  .contacts {
    display: flex;
    flex-direction: column;
    align-items: center;
    overflow: auto;
    gap: 0.8rem;
    &::-webkit-scrollbar {
      width: 0.2rem;
      &-thumb {

```



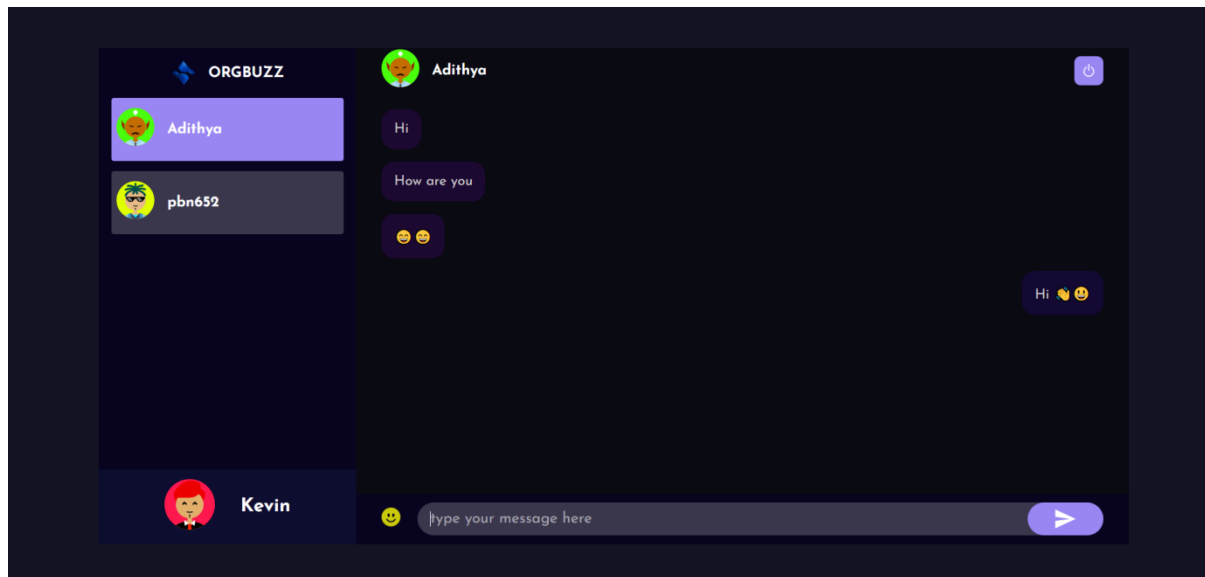
```

        background-color: #ffffff39;
        width: 0.1rem;
        border-radius: 1rem;
    }
}
.contact {
    background-color: #ffffff34;
    min-height: 5rem;
    cursor: pointer;
    width: 90%;
    border-radius: 0.2rem;
    padding: 0.4rem;
    display: flex;
    gap: 1rem;
    align-items: center;
    transition: 0.5s ease-in-out;
    .avatar {
        img {
            height: 3rem;
        }
    }
    .username {
        h3 {
            color: white;
        }
    }
}
.selected {
    background-color: #9a86f3;
}
}
.current-user {
    background-color: #0d0d30;
    display: flex;
    justify-content: center;
    align-items: center;
    gap: 2rem;
    .avatar {
        img {
            height: 4rem;
            max-inline-size: 100%;
        }
    }
    .username {
        h2 {
            color: white;
        }
    }
}
@media screen and (min-width: 720px) and (max-width: 1080px) {
    gap: 0.5rem;
    .username {
        h2 {
            font-size: 1rem;
        }
    }
}
}
;

```

Here we have our chat list, with all the available contacts who have been registered to the network. The active chat will be highlighted in lilac where as the the rest of teh chats will have a grey background. The name of the currently logged in user will be displayed at the bottom.

Chat Window



```
import React, { useState, useEffect, useRef } from "react";
import styled from "styled-components";
import ChatInput from "../ChatInput";
import Logout from "../Logout";
import { v4 as uuidv4 } from "uuid";
import axios from "axios";
import { sendMessageRoute, recieveMessageRoute } from "../utils/APIRoutes";

export default function ChatContainer({ currentChat, socket }) {
  const [messages, setMessages] = useState([]);
  const scrollRef = useRef();
  const [arrivalMessage, setArrivalMessage] = useState(null);
  useEffect(async () => {
    const data = await JSON.parse(
      localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)
    );
    const response = await axios.post(recieveMessageRoute, {
      from: data._id,
      to: currentChat._id,
    });
    setMessages(response.data);
  }, [currentChat]);
  useEffect(() => {
    const getCurrentChat = async () => {
      if (currentChat) {
        await JSON.parse(
          localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)
        )._id;
      }
    };
    getCurrentChat();
  }, [currentChat]);
  const handleSendMsg = async (msg) => {
    const data = await JSON.parse(
      localStorage.getItem(process.env.REACT_APP_LOCALHOST_KEY)
    );
    socket.current.emit("send-msg", {
      to: currentChat._id,
      from: data._id,
      msg,
    });
    await axios.post(sendMessageRoute, {
      from: data._id,
      to: currentChat._id,
      message: msg,
    });
    const msgs = [...messages];
    msgs.push({ fromSelf: true, message: msg });
    setMessages(msgs);
  };
}
```

```

useEffect(() => {
  if (socket.current) {
    socket.current.on("msg-recieve", (msg) => {
      setArrivalMessage({ fromSelf: false, message: msg });
    });
  }
}, []);
useEffect(() => {
  arrivalMessage && setMessages((prev) => [...prev, arrivalMessage]);
}, [arrivalMessage]);
useEffect(() => {
  scrollRef.current?.scrollIntoView({ behavior: "smooth" });
}, [messages]);
return (
  <Container>
    <div className="chat-header">
      <div className="user-details">
        <div className="avatar">
          <img
            src={`data:image/svg+xml;base64,${currentChat.avatarImage}`}
            alt=""
          />
        </div>
        <div className="username">
          <h3>{currentChat.username}</h3>
        </div>
      </div>
      <Logout />
    </div>
    <div className="chat-messages">
      {messages.map((message) => {
        return (
          <div ref={scrollRef} key={uuidv4()}>
            <div
              className={`message ${
                message.fromSelf ? "sended" : "recieved"
              }`}
            >
              <div className="content ">
                <p>{message.message}</p>
              </div>
            </div>
          </div>
        );
      })}
    </div>
    <ChatInput handleSendMsg={handleSendMsg} />
  </Container>
);
}
const Container = styled.div`
  display: grid;
  grid-template-rows: 10% 80% 10%;
  gap: 0.1rem;
  overflow: hidden;
  @media screen and (min-width: 720px) and (max-width: 1080px) {
    grid-template-rows: 15% 70% 15%;
  }
  .chat-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 2rem;
  }
  .user-details {
    display: flex;
    align-items: center;
    gap: 1rem;
  }
  .avatar {
    img {
      height: 3rem;
    }
  }
  .username {
    h3 {

```

