**BUDHANILKANTHA TECHNICAL SCHOOL**

**Budhanilkantha, Kathmandu**

**(Affiliated to CTEVT)**

A logo with a letter and a book

Description automatically generated

**Messaging Web Application**

**“Major Project”**

**Submitted By:**

**Name: Dhirendra Kathayat**

**Roll no: 07.**

**Date:2080/09/01**

**Submitted To:**

**Department of Diploma in Computer Engineering**

***Abstract:***

The Message Web Application is introduced as a dynamic platform designed to facilitate real-time communication through text messages, file sharing, audio messaging, and video calls. With a focus on providing a seamless and efficient user experience, the application caters to a diverse audience seeking convenient and versatile communication solutions. The project employs an iterative development model, blending features from Facebook Messenger and WhatsApp, and is crafted using PHP, JavaScript, HTML, and CSS. The hardware and software infrastructure, including an ASUS ExpertBook, Intel Core i5 processor, and XAMPP as the local server, ensure robust functionality. This abstract provides a comprehensive overview of the project's purpose, scope, target audience, methodology, and the technological tools employed, setting the stage for effective communication and collaboration.

Table of Contents

[Introduction: 1](#_Toc159859764)

[Purpose: 1](#_Toc159859765)

[Scope: 1](#_Toc159859766)

[Audience: 1](#_Toc159859767)

[Literature review: 1](#_Toc159859768)

[Methodology: 1](#_Toc159859769)

[Tech and Tools: 2](#_Toc159859770)

[Hardware: 2](#_Toc159859771)

[Software: 2](#_Toc159859772)

[Development Tools: 2](#_Toc159859773)

[System Design: 2](#_Toc159859774)

[Use Case: 2](#_Toc159859775)

[ER Diagram: 4](#_Toc159859776)

[Getting Start with WeChat 4](#_Toc159859777)

[Create your Account. 5](#_Toc159859778)

[Explore the User Interface. 5](#_Toc159859779)

[Connect with Contacts: 5](#_Toc159859780)

[Start Messaging: 5](#_Toc159859781)

[Features: 5](#_Toc159859782)

[User Registration and Authentication: 5](#_Toc159859783)

[Multimedia Supports: 5](#_Toc159859784)

[Functional Requirements: 6](#_Toc159859785)

[Non-Functional Requirements: 6](#_Toc159859786)

[Architecture: 6](#_Toc159859787)

[Client-Side: 7](#_Toc159859788)

[Server-Side: 7](#_Toc159859789)

[Communication Protocols: 7](#_Toc159859790)

[Scalability and Load Balancing: 7](#_Toc159859791)

[Deployment: 7](#_Toc159859792)

[Requirements: 7](#_Toc159859793)

[Server environment: 7](#_Toc159859794)

[Hardware Requirements: 8](#_Toc159859795)

[Deployment Steps: 8](#_Toc159859796)

[Environment Configuration: 8](#_Toc159859797)

[Database Setup: 8](#_Toc159859798)

[Other configurations: 8](#_Toc159859799)

[Start Application: 8](#_Toc159859800)

[Testing: 8](#_Toc159859801)

[Documentation Updates: 8](#_Toc159859802)

[Conclusion: 8](#_Toc159859803)

[References: 9](#_Toc159859804)

# Introduction:

## Purpose:

The main purpose of this message web application is to facilitate its users with real time communication medium like text message, file sharing, audio message, audio, and video call.

## Scope:

The message web application is designed to facilitate real-time communication between users through a web-based platform. The primary goal is to provide a seamless and efficient messaging experience while promoting user engagement and collaboration.

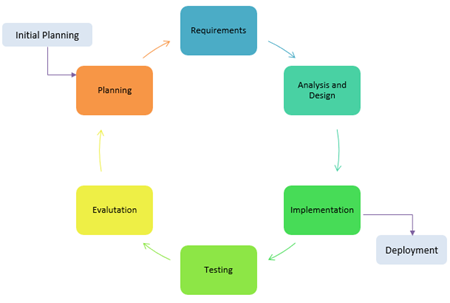
Audience:   
The audience of this message web application are general users, those who are looking for a convenient and efficient way to communicate with friends, family, and colleagues and those who are seeking a user-friendly platform for text-based and multimedia messaging.

# Literature review:

The Messaging Web Application is a simple messaging application which will run as web app on the internet. This will be hybrid application where I will add some features of Facebook Messenger and some features of WhatsApp. This web application is designed and developed to provide the efficient and easy communication between two individuals. Our web application is developed using PHP, JavaScript and iterative model is used in a development.

# Methodology:

From various methodology out in the market, we had chosen iterative model for this specific project. Iterative model is a type of model which allows users to access of earlier phases while making iterations.



# Tech and Tools:

## Hardware:

* Device: ASUS ExpertBook
* Processor: intel core i5
* RAM: 16 GB
* Storage: 128 GB SSD, 1 TB HDD

## Software:

* XAMPP as Local server
* Windows 11 Operating system
* Vs code as an IDE.

## Development Tools:

* Language: PHP, JavaScript, HTML and CSS.
* IDE: VS Code

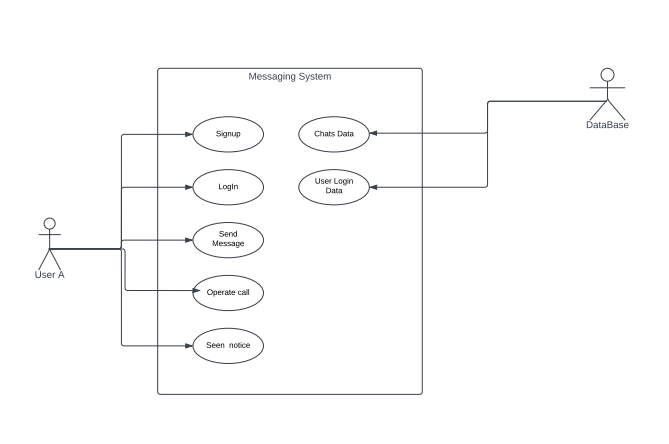
# System Design:

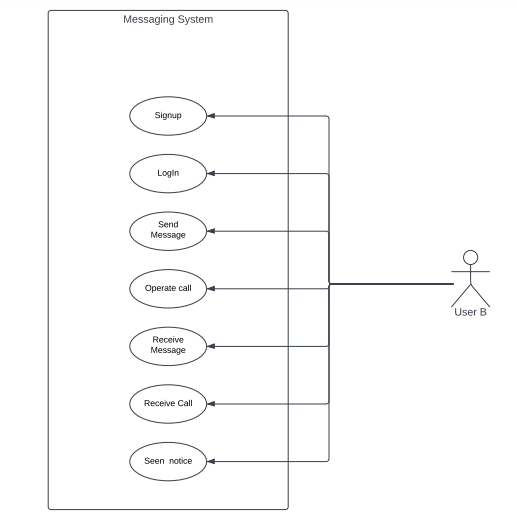
## Use Case:

Creating use case diagram requires identifying actors and their use cases. For this messaging web application use case can be presented as:

Actors:

1. User A: First actor to interact with system by sending messages, operating calls and updating status of messages and profiles.
2. Database: Can be second actor stores chat data and user login data
3. User B: It is the last actor to interact with the system. It performs message sending and receiving, operating calls, receiving calls etc.





# ER Diagram:

An ERD is used to capture the relationships between the entities and the association between them. Here is a simplified ERD of messaging application:

A screenshot of a computer screen

Description automatically generated

The ER-Diagram represents the database model of messaging application where three main entities are used, and they are:

1. User: User is the main entity of the ER-Diagram which have attributes like username, user\_id, Contact, Email, DOB, Gender, etc.
2. Messages: Message is another entity with Message\_id, from\_user, to\_user, type, message, and status etc.
3. Convo\_list: Convo\_list is another main entity with attributes Convo\_id, from\_user, to\_user and date\_created etc.

# Getting Start with WeChat

Welcome to WeChat, the platform that simplifies communication and brings people closer. This guide will help you get started with the application quickly and make the most out of its features.

## Create your Account.

To begin using WeChat, you need to create a user account.

* Visit WeChat.com.np where we will host our application.
* Sign-up: Click on ‘Signup’ button. Fill the required information, including your contact and a secure password.
* Login: The system automatically returns to the login page then log in with your contact and password.

## Explore the User Interface.

Now you are logged in, take a moment to explore the user interface:

* Dashboard: This is your home base, displaying your conversations, contacts, and notifications.
* Profile Settings: On the right corner of your dashboard navigation bar, you will see your profile and when you click on it you will be able to access your profile and logout option inside the profile option you will find the profile setting.

## Connect with Contacts:

WeChat is all about connecting with others. Here how you can add contacts or friends:

* Search for Contacts: Use the search bar to find friends, colleagues, or family members already using the platform.

## Start Messaging:

* Select the contact: Click on a contact’s name to open a chat window.
* Compose a message: Type your message in the chat box and hit ‘send’.
* Multimedia Sharing: Explore edition features like file attachments, audio message and multimedia sharing.

# Features:

WeChat, a web messaging application, typically offers a variety of features to facilitate communication and enhance user experience. Below is the list of common features that you will find in WeChat:

## User Registration and Authentication:

* User account creation with contact and email.
* Secure login mechanisms to protect user accounts.

## Multimedia Supports:

* Ability to send and receive images, videos, and other multimedia files.

# Functional Requirements:

Here are some of the functional requirements that outline the core features and capabilities that this messaging web application can offer to meet the needs of its users:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Priority | | |
| Low | Medium | High |
| 1. User |  |  |  |
| A.1. User Can login to the system |  |  |  |
| 1. User can login by Contact and password |  |  |  |
| 1. User can reset the password |  |  |  |
| 1. User can Logout from the system |  |  |  |
| 1. User can Change Password |  |  |  |
| 1. User can edit their profile |  |  |  |
| A.2. User Can Signup in the system |  |  |  |
| 1. User can sign up with credentials |  |  |  |
| 1. User can sign up with Other social media |  |  |  |
| A.3. User can communicate with each other |  |  |  |
| 1. User can communicate with message |  |  |  |
| 1. User can communicate with multimedia |  |  |  |
| 1. User can communicate with voice call |  |  |  |
| 1. User can communicate with video call |  |  |  |
| A.4. User can make friends |  |  |  |
| 1. User can send friend request |  |  |  |
| 1. User can search another user |  |  |  |

# Non-Functional Requirements:

Here are some of non-functional requirements that this system can offer given below as follows:

1. Performance: This application ca load messages and multimedia content within minimum time for optimal user experience
2. Cross-Browser Compatibility: This application will be compatible with the latest versions of major web browsers.

# Architecture:

The WeChat message web application is built on a robust and scalable architecture to ensure seamless communication and optimal performance. The architecture comprises various components working together cohesively to deliver a reliable messaging platform.

## Client-Side:

* User Interface (UI): The front-end of WeChat is designed using modern web technologies such as HTML, CSS, and JavaScript. It provides an intuitive and user-friendly interface for interacting with the application.
* web sockets: WeChat utilizes web sockets to establish real-time, bidirectional communication between the client and server, enabling instant message delivery and updates.

## Server-Side:

* Application Server: This component handles the core logic of the application, including user authentication, message processing, and communication with the database.
* Web Server: Responsible for serving the web application to users and managing HTTP requests. It interacts with the application server to fetch and update data.
* Database: WeChat employs a relational database system to store user profiles, messages, and multimedia content securely. This ensures data consistency and reliability.
* Authentication Server: Manages user registration, login, and authentication processes to ensure the security of user accounts.

## Communication Protocols:

* HTTP/HTTPS: The Hypertext Transfer Protocol is used for client-server communication, while HTTPS ensures a secure and encrypted connection, safeguarding user data during transmission.
* WebSocket: Facilitates real-time communication by establishing a persistent connection between the client and server, enabling instant message updates and notifications.

## Scalability and Load Balancing:

* Load Balancer: Distributes incoming traffic across multiple servers to ensure optimal resource utilization and prevent overloading.
* Scalable Infrastructure: The architecture is designed to scale horizontally, allowing for the addition of more servers to handle increasing user loads efficiently.

# Deployment:

## Requirements:

### Server environment:

* Operating system: Our WeChat is most compatible for Windows OS.
* Web Server: Apache (XAMPP or WAMPP).
* Database: WeChat is compatible with MYSQL database of XAMPP server.

### Hardware Requirements:

* + Processor: minimum i3 or Ryzen 3
  + RAM: minimum 4 GB
  + Storage: minimum 128 GB SSD, 256 GB HDD

## Deployment Steps:

### Environment Configuration:

Configure environment variables for sensitive information such as database, credentials, and API keys for audio and video conferencing.

### Database Setup:

Download XAMPP on a hosted server and import the database file on the server which is along with project files.

### Other configurations:

In our web application we used AgoraRTC for audio and video conferencing, so we need to update the token key. To update token register on agora site and create new permanent token and use it.

### Start Application:

Start server and start the application if it on a local server. Type localhost:8080/WeChat on the browser URL if it is hosted then use domain name on the URL bar.

### Testing:

Once the application is started once again check the functions, features are working properly or not.

### Documentation Updates:

Update any relevant documentation with production-specific details and configurations.

# Conclusion:

In summary, the Message Web Application stands as a versatile real-time communication platform designed for text messaging, file sharing, audio, and video calls. Tailored for general users seeking convenience and efficiency, it fosters engagement and collaboration. The iterative development model, inspired by Facebook Messenger and WhatsApp, utilizes PHP, JavaScript, HTML, and CSS. Supported by an ASUS ExpertBook and XAMPP, the application promises a user-friendly and technologically robust experience. This brief conclusion highlights the project's commitment to effective communication and collaboration, poised to transform digital interactions.

# References:

Meta, 2023. *Messenger.* [Online]   
Available at: https:www.messenger.com  
[Accessed 1 dec 2023].

php, 2017. *Php.* [Online]   
Available at: https://www.php.net/docs.php  
[Accessed 1 dec 2023].