TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

Icon

Description automatically generated

ĐỒ ÁN

THIẾT KẾ KIẾN TRÚC PHẦN MỀM

WEB NHẮN TIN

Nhóm: 5

Lớp : 66PM

Sinh viên thực hiện: Vũ Văn Thao - 0204566

Phạm Văn Đăng - 0186566

HÀ NỘI, 27/05/2025.

MỤC LỤC

[**Chương 1. Giới thiệu đề tài** 3](#_Toc199153568)

[**1.1.Lý do chọn đề tài** 3](#_Toc199153569)

[**1.2. Mục tiêu của đồ án** 3](#_Toc199153570)

[**1.3. Phạm vi và giới hạn** 3](#_Toc199153571)

[**1.4. Phương pháp thực hiện** 3](#_Toc199153572)

[**Chương 2. Phân tích hệ thống** 4](#_Toc199153573)

[**2.1. Mô tả bài toán** 4](#_Toc199153574)

[**2.2. Yêu cầu chức năng** 4](#_Toc199153575)

[**2.3. Yêu cầu phi chức năng** 4](#_Toc199153576)

[**2.4. Use Case Diagram tổng quát** 5](#_Toc199153577)

[**2.5. Mô tả các Use Case chính** 5](#_Toc199153578)

[**2.6. Biểu đồ hoạt động (Các chức năng chính)** 6](#_Toc199153579)

[**Chương 3. Thiết kế hệ thống** 6](#_Toc199153580)

[**3.1. Lựa chọn mô hình kiến trúc (Monolithic, Microservices, MVC, Layered, Clean Architecture, ...)** 6](#_Toc199153581)

[**3.2. Sơ đồ kiến trúc tổng thể** 7](#_Toc199153582)

[**3.3. Biểu đồ tuần tự (*các chức năng chính*)** 7](#_Toc199153583)

[**3.4. Thiết kế lớp (Class Diagram)** 11](#_Toc199153584)

[**3.5. Thiết kế giao diện người dùng (Mockup / Wireframe)** 11](#_Toc199153585)

[**Chương 4. Cài đặt và triển khai hệ thống** 13](#_Toc199153586)

[**4.1. Môi trường triển khai** 13](#_Toc199153587)

[**4.2. Cài đặt hệ thống** 13](#_Toc199153588)

[4.2.1 Yêu cầu hệ thống 13](#_Toc199153589)

[4.2.2 Các bước cài đặt chương trình 13](#_Toc199153590)

[**Chương 5. Kết quả và đánh giá** 14](#_Toc199153591)

[**5.1. Kết quả thử nghiệm hệ thống** 14](#_Toc199153592)

[**5.2. Đánh giá hiệu quả hệ thống** 14](#_Toc199153593)

[**Tài liệu tham khảo** 15](#_Toc199153594)

[**Phụ lục (nếu có)** 15](#_Toc199153595)

# **Chương 1. Giới thiệu đề tài**

* 1. **Lý do chọn đề tài**

Trong bối cảnh xã hội hiện đại, việc liên lạc và trao đổi thông tin giữa con người ngày càng phụ thuộc vào các nền tảng trực tuyến. Các ứng dụng nhắn tin truyền thống như Messenger, Zalo, Telegram tuy phổ biến nhưng thường phức tạp và thiếu tính cá nhân hóa. Đối với sinh viên ngành công nghệ thông tin, việc xây dựng một website nhắn tin giúp rèn luyện kỹ năng lập trình thực tế, đồng thời hiểu sâu hơn về các công nghệ web hiện đại.  
Đề tài “Xây dựng website nhắn tin sử dụng **React.js và Firebase**” được lựa chọn nhằm ứng dụng kiến thức đã học, kết hợp với công nghệ thời gian thực và lưu trữ đám mây để tạo ra một nền tảng nhắn tin đơn giản, hiệu quả và dễ mở rộng.

## **1.2. Mục tiêu của đồ án**

- Thiết kế và xây dựng một website nhắn tin theo thời gian thực sử dụng **React.js** cho giao diện người dùng và **Firebase** cho backend.

- Hỗ trợ các tính năng: đăng ký, đăng nhập, thêm bạn bè, gửi/nhận tin nhắn (văn bản và hình ảnh), hiển thị trạng thái “đã xem”, thời gian gửi.

- Triển khai chức năng chặn người dùng, chỉ hiển thị nút “bỏ chặn” nếu người đó đã bị chặn.

- Giao diện trực quan, dễ sử dụng trên trình duyệt web

## **1.3. Phạm vi và giới hạn**

**Phạm vi**:

- Người dùng có thể đăng ký, đăng nhập bằng email và mật khẩu.

- Gửi và nhận tin nhắn dạng văn bản hoặc hình ảnh theo thời gian thực bằng Firestore.

- Quản lý danh sách bạn bè, chức năng chặn người dùng, theo dõi trạng thái đã xem tin nhắn.

- Tương thích với các trình duyệt hiện đại (Chrome, Firefox, Edge).

**Giới hạn**:

- Không có phiên bản ứng dụng di động (mobile app).

- Firebase Authentication chỉ hỗ trợ đăng nhập cơ bản, chưa tích hợp Google/Facebook login.

- Chưa áp dụng mã hóa đầu cuối (end-to-end encryption).

- Hạn chế về số lượng người dùng trong giai đoạn thử nghiệm.

## **1.4. Phương pháp thực hiện**

**-** Thu thập và phân tích yêu cầu: Xác định các tính năng thiết yếu của một hệ thống nhắn tin.

**-** Thiết kế hệ thống: Sử dụng sơ đồ use case, mô hình dữ liệu Firebase, thiết kế giao diện bằng React.

**-** Xây dựng hệ thống:

* + Frontend: React.js, sử dụng useState, useEffect, và các hook tùy chỉnh.
  + Backend: Firebase Authentication, Cloud Firestore, Firebase Storage.

**-** Kiểm thử: Thử nghiệm chức năng và hiệu năng trên nhiều trình duyệt.

**-** Triển khai và đánh giá: Đăng tải web lên Firebase Hosting hoặc nền tảng tương tự, đánh giá mức độ hoàn thiện của sản phẩm.

# **Chương 2. Phân tích hệ thống**

## **2.1. Mô tả bài toán**

- Hiện nay, nhu cầu nhắn tin và trao đổi thông tin trực tuyến giữa người dùng là vô cùng phổ biến

- Các nền tảng có sẵn như Messenger, Zalo, WhatsApp tuy đầy đủ tính năng nhưng lại không phù hợp cho các hệ thống tùy biến hoặc học tập

- Bài toán đặt ra là xây dựng một website nhắn tin theo thời gian thực cho phép người dùng có thể đăng ký, đăng nhập, kết bạn, gửi tin nhắn (văn bản, hình ảnh), và theo dõi trạng thái tin nhắn

- Hệ thống cần đảm bảo đồng bộ dữ liệu, hoạt động ổn định trên trình duyệt và có thể dễ dàng mở rộng thêm tính năng

- Firebase sẽ được sử dụng làm backend để xử lý thời gian thực và lưu trữ dữ liệu, React.js được dùng để xây dựng giao diện người dùng

## **2.2. Yêu cầu chức năng**

- Người dùng có thể đăng ký tài khoản mới

- Người dùng có thể đăng nhập và đăng xuất khỏi hệ thống

- Người dùng có thể tìm kiếm

- Người dùng có thể nhắn tin văn bản theo thời gian thực

- Hệ thống hỗ trợ gửi hình ảnh trong cuộc trò chuyện

- Hiển thị thời gian gửi tin nhắn và trạng thái “đã xem”

- Người dùng có thể chặn bạn bè và không thể gửi tin nhắn đến người bị chặn

- Chỉ có thể bỏ chặn thông qua nút “Bỏ chặn”, không thể bỏ chặn bằng hành động khác

- Danh sách trò chuyện hiển thị theo thứ tự tin nhắn mới nhất

## **2.3. Yêu cầu phi chức năng**

- Giao diện đơn giản, dễ sử dụng, tương thích với các trình duyệt phổ biến

- Hệ thống phản hồi nhanh và cập nhật dữ liệu theo thời gian thực (real-time)

- Đảm bảo độ ổn định và toàn vẹn dữ liệu khi có nhiều người dùng truy cập đồng thời

- Bảo mật thông tin người dùng, sử dụng Firebase Authentication để kiểm soát truy cập

- Cấu trúc mã nguồn rõ ràng, dễ bảo trì và mở rộng về sau

- Tương thích với cả máy tính thông qua trình duyệt

## **2.4. Use Case Diagram tổng quát**

A diagram of a company

AI-generated content may be incorrect.

## **2.5. Mô tả các Use Case chính**

**Use case: đăng nhập**

- Tên: đăng nhập

- Tác nhân: người dùng (user)

- Mục tiêu: cho phép người dùng truy cập vào hệ thống bằng tài khoản đã đăng ký

- Mô tả: người dùng nhập email và mật khẩu để xác thực. nếu thông tin hợp lệ, hệ thống cho phép đăng nhập và chuyển đến giao diện chính

- Quan hệ mở rộng: có thể mở rộng với “đăng ký” nếu người dùng chưa có tài khoản

**Use case: đăng ký**

- Tên: đăng ký

- Tác nhân: người dùng

- Mục tiêu: tạo mới một tài khoản người dùng

- Mô tả: người dùng nhập thông tin như email, mật khẩu. hệ thống kiểm tra hợp lệ và tạo tài khoản trên firebase authentication

**Use case: quản lý liên hệ**

- Tên: quản lý liên hệ

- Tác nhân: người dùng

- Mục tiêu: quản lý danh sách bạn bè và mối quan hệ với người dùng khác

- Mô tả: bao gồm tìm kiếm người dùng, gửi lời mời kết bạn, chặn người dùng và bỏ chặn người dùng. hệ thống cập nhật trạng thái kết bạn và trạng thái chặn trong cơ sở dữ liệu

**Use case: quản lý tệp tin và phương tiện**

- Tên: quản lý tệp tin và phương tiện

- Tác nhân: người dùng

- Mục tiêu: hỗ trợ người dùng gửi và nhận tệp tin (ảnh) trong cuộc trò chuyện

- Mô tả: người dùng có thể tải lên hình ảnh từ thiết bị cá nhân, lưu trữ trên firebase storage. người nhận có thể xem và tải xuống ảnh

- Quan hệ bao hàm: gồm các hành động “tải lên hình ảnh” và “tải xuống tệp tin/hình ảnh”

## **2.6. Biểu đồ hoạt động (Các chức năng chính)**

# **Chương 3. Thiết kế hệ thống**

## **3.1. Lựa chọn mô hình kiến trúc**

- Hệ thống sử dụng kiến trúc **Component-based Architecture** phù hợp với framework React.js

- Mỗi chức năng giao diện (gửi tin nhắn, danh sách trò chuyện, tìm kiếm bạn bè...) được xây dựng thành một component riêng biệt

- Backend sử dụng Firebase – kiến trúc không máy chủ (serverless) nên không cần mô hình Monolithic hay Microservices

- Firebase Firestore hoạt động theo kiến trúc realtime, cung cấp luồng dữ liệu đồng bộ cho frontend

**Lý do lựa chọn Component Architecture**

- Tăng khả năng tái sử dụng code

- Dễ bảo trì và mở rộng khi chia nhỏ thành các khối độc lập

- Phù hợp với triết lý của React (tất cả là component)

- Hỗ trợ tốt cho quản lý trạng thái thông qua Context hoặc Redux

## **3.2. Sơ đồ kiến trúc tổng thể**

**A diagram of a firebase storage system

AI-generated content may be incorrect.**

## **3.3. Biểu đồ tuần tự (*các chức năng chính*)**

- Biểu đồ tuần tự đăng nhậpA diagram of a diagram

AI-generated content may be incorrect.

- Biểu đồ tuần tự đăng ký

A diagram of a company

AI-generated content may be incorrect.

- Biểu đồ tuần tự người dùng

A diagram of a chatbot

AI-generated content may be incorrect.

- Biểu đồ tuần tự tìm kiếm tin nhắn người dùng

A diagram of a chat

AI-generated content may be incorrect.

- Biểu đồ tuần tự nhắn tin

A diagram of a diagram

AI-generated content may be incorrect.

- Biểu đồ tuần tự gửi ảnh

A diagram of a diagram of a chatbot

AI-generated content may be incorrect.

- Biểu đồ tuần tự chặn người dùng

A diagram of a diagram

AI-generated content may be incorrect.

## **3.4. Thiết kế lớp (Class Diagram)**

A diagram of a computer code

AI-generated content may be incorrect.

## **3.5. Thiết kế giao diện người dùng (Mockup / Wireframe)**

- Đăng nhập và đăng ký

A screenshot of a computer

AI-generated content may be incorrect.

- Trang chủ

A screenshot of a chat

AI-generated content may be incorrect.

- Tìm kiếm tin nhắn

A blurry image of a blue and brown background

AI-generated content may be incorrect.

# **Chương 4. Cài đặt và triển khai hệ thống**

## **4.1. Môi trường triển khai**

- Trình duyệt: Google Chrome, Firefox, Edge (phiên bản mới nhất)

- Hệ điều hành: Windows 10+, macOS hoặc Linux

- Công nghệ frontend: React.js (phiên bản 18 trở lên)

- Backend: Firebase (Authentication, Firestore, Storage)

- Trình quản lý gói: npm hoặc yarn

- Công cụ lập trình: Visual Studio Code

## **4.2. Cài đặt hệ thống**

* + 1. Yêu cầu hệ thống

- CPU: Intel Core i3 trở lên

- RAM: Tối thiểu 4GB

- Ổ cứng: Tối thiểu 500MB trống để lưu mã nguồn và cài đặt gói

**Phần mềm:**

- Node.js: phiên bản 16 trở lên

- npm: phiên bản tương ứng với Node.js

- Trình duyệt hỗ trợ ES6+

- Tài khoản Google để cấu hình Firebase

* + 1. Các bước cài đặt chương trình

**Bước 1: Cài đặt môi trường phát triển**

- Cài đặt Node.js từ <https://nodejs.org>

- Cài đặt Visual Studio Code (VSCode)

- Tạo project React:  
A white rectangular object with a black border

AI-generated content may be incorrect.

Bước 2: Cài đặt các thư viện cần thiết

A close-up of a computer screen

AI-generated content may be incorrect.

Bước 3: Cấu hình Firebase

- Truy cập https://console.firebase.google.com

- Tạo project mới

- Kích hoạt Firebase Authentication (email/password)

- Tạo Firestore Database và Firebase Storage

- Thêm config Firebase vào file firebase.js trong source code:

A screenshot of a computer

AI-generated content may be incorrect.

Bước 4: Khởi chạy ứng dụng



# **Chương 5. Kết quả và đánh giá**

## **5.1. Kết quả thử nghiệm hệ thống**

Sau khi hoàn tất quá trình xây dựng và triển khai, hệ thống đã được thử nghiệm với các chức năng chính như sau:

- Đăng ký, đăng nhập người dùng bằng Firebase Authentication hoạt động ổn định

- Gửi và nhận tin nhắn văn bản theo thời gian thực hoạt động chính xác với độ trễ thấp

- Gửi hình ảnh và hiển thị ảnh trong khung chat được thực hiện mượt mà nhờ Firebase Storage

- Tình trạng tin nhắn “đã xem” và thời gian gửi được cập nhật đúng theo từng người dùng

- Danh sách bạn bè được cập nhật theo thời gian thực

- Tính năng chặn và bỏ chặn người dùng hoạt động đúng: khi bị chặn, người kia không thể nhắn tin, và chỉ có thể bỏ chặn bằng nút thao tác

- Giao diện hiển thị tốt trên cả máy tính và điện thoại thông qua trình duyệt

Hệ thống đã được chạy thử với 10 tài khoản người dùng, mô phỏng quá trình nhắn tin đồng thời. Các phản hồi được xử lý tốt, không phát hiện lỗi nghiêm trọng.

## **5.2. Đánh giá hiệu quả hệ thống**

**Ưu điểm**

- Giao diện đơn giản, dễ sử dụng, thân thiện với người dùng

- Cập nhật tin nhắn và trạng thái theo thời gian thực giúp người dùng tương tác mượt -mà

- Firebase hỗ trợ tốt về bảo mật, hiệu năng và mở rộng dữ liệu

- Cấu trúc component giúp mã nguồn dễ bảo trì và phát triển thêm tính năng

- Khả năng triển khai dễ dàng nhờ Firebase Hosting

**Hạn chế**

- Chưa có tính năng gọi thoại, gọi video

- Chưa hỗ trợ thông báo đẩy (push notification) khi có tin nhắn mới

- Không có mã hóa đầu cuối, dữ liệu tin nhắn có thể đọc được từ Firestore

**Đề xuất phát triển trong tương lai**

- Tích hợp chức năng thông báo đẩy

- Bổ sung mã hóa end-to-end để tăng tính bảo mật

- Tối ưu hóa giao diện mobile và thêm chế độ tối (dark mode)

- Xây dựng phiên bản ứng dụng di động riêng biệt (React Native)

# **Tài liệu tham khảo**

* Ứng dụng nhắn tin Messenger