**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**TRẦN NGUYỄN AN**

**PHÁT TRIỂN RESTFUL API CHO ỨNG DỤNG CẢNH BÁO KẸT XE SỬ DỤNG .NET CORE**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2023BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**TRẦN NGUYỄN AN**

**PHÁT TRIỂN RESTFUL API CHO ỨNG DỤNG CẢNH BÁO KẸT XE SỬ DỤNG .NET CORE**

**Mã số sinh viên: 2051010004**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: Tiến sĩ TRƯƠNG HOÀNG VINH**

**TP. HỒ CHÍ MINH, 2023**

LỜI CẢM ƠN

Để hoàn thiện đồ án này, em đã nhận được sự giúp đỡ từ các thầy cô trong khoa Công Nghệ Thông Tin của trường Đại học Mở Thành phố Hồ Chí Minh. Nhất là sự giúp đỡ của thầy Trương Hoàng Vinh và thầy Dương Hữu Thành đã tích cực phản hồi nh**ữ**ng câu hỏi về đề tài đồ án của em. Qua đó đã giúp em hoàn thành tốt được đề tài của chính mình.

Tuy còn nhiều thiếu sót về kiến thức chuyên ngành nhưng em cũng đã hoàn thành được các yêu cầu cơ bản của đề tài. Kính mong thầy cô trong khoa đánh giá và nhận xét tích cực về bài làm của em. Những nhận xét sẽ giúp em nhận ra được những thiếu sót, từ đó em có thể cải thiện được nhưng kiến thức đã thiếu sót trong quá trình làm đồ án.

Em xin chân thành cảm ơn những thầy cô đã giúp đỡ và nhận xét về đề tài của em.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

Đề tài “Phát triển RESTful API cho ứng dụng cảnh bảo kẹt xe” phát triển một phần mềm giúp người đi được nhận được cảnh báo về các đoạn đường đang bị kẹt xe. Sử dụng API của Google Map kết hợp với phương thức getCurrentPosition() để biết được người dùng đang ở vị trí nào và cảnh báo cho người đến sau qua thông báo của điện thoại. Sử dụng SQL Server để quản lý và truy xuất Database, sẽ tìm cách đưa Database triển khai online. Dùng ngôn ngữ C# tạo RESTful API để thực hiện các thao tác truy xuất dữ liệu từ SQL Server. Dùng ngôn ngữ Java trong lập trình ứng dụng di động.

Khi sử dụng người đi đường cần có tài khoản để có thể đăng lên các cảnh báo. Sẽ kiểm tra từ nhiều tài khoản tránh trường hợp có người muốn lợi dụng ngầm gây rồi cho người đi đường khác. Dựa trên tọa độ từ người đăng đầu tiên sẽ kiểm tra xung quanh xem có người khác đăng hay không. Sau đó sẽ dựa trên tọa độ đó để cảnh bảo cho những người đang ở gần.

ABSTRACT

**The topic "Developing a RESTful API for vehicle breakdown alert application using ASP.NET CORE" involves the development of software aimed at providing real-time traffic congestion alerts to commuters. It utilizes the Google Maps API in conjunction with the getCurrentPosition() method to determine the user's current location and issues traffic alerts to those following behind via smartphone notifications. SQL Server is used for database management and retrieval, with plans to deploy the database online. A RESTful API is created using the C# programming language to facilitate data retrieval operations from SQL Server. Java is employed for mobile application development.**

**Users are required to have accounts to post alerts and prevent misuse by other users. Multiple accounts are checked to avoid situations where individuals may attempt to exploit the system to create false alerts. Based on the coordinates provided by the first user, the system checks the surrounding area for additional alerts from other users. Subsequently, it issues alerts to those who are in close proximity.**

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc148402067)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc148402068)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc148402069)

[ABSTRACT 4](#_Toc148402070)

[DANH MỤC TỪ VIẾT TẮT 9](#_Toc148402071)

[DANH MỤC HÌNH VẼ 10](#_Toc148402072)

[DANH MỤC BẢNG 11](#_Toc148402073)

[MỞ ĐẦU 12](#_Toc148402074)

[Chương 1. GIỚI THIỆU ĐỀ TÀI 13](#_Toc148402075)

[1.1. Giới thiệu 13](#_Toc148402076)

[1.2. Mục tiêu đề tài 13](#_Toc148402077)

[1.2.1. Thiết kế giao diện người dùng (UI/UX) 13](#_Toc148402078)

[1.2.2. Phát triển tính năng cơ bản 14](#_Toc148402079)

[1.2.3. Tích hợp cơ sở dữ liệu SQL Server 14](#_Toc148402080)

[1.2.4. Phát triển tính năng đăng nhận cảnh báo 14](#_Toc148402081)

[1.2.5. Bảo mật và quản lý tài khoản 14](#_Toc148402082)

[1.3. Lý do chọn đề tài 14](#_Toc148402083)

[1.3.1. Tính thực tiễn 14](#_Toc148402084)

[1.3.2. Tính ứng dụng rộng rãi 14](#_Toc148402085)

[1.3.3. Khả năng ứng dụng thực tế 14](#_Toc148402086)

[1.3.4. Khám phá kiến thức mới 15](#_Toc148402087)

[1.3.5. Tính đột phá 15](#_Toc148402088)

[Chương 2. CƠ SỞ LÝ THUYẾT 16](#_Toc148402089)

[2.1. Quản lý dữ liệu 16](#_Toc148402090)

[2.1.1. Thu thập dữ liệu 16](#_Toc148402091)

[2.1.2. Lưu trữ dữ liệu 16](#_Toc148402092)

[2.1.3. Xử lý dữ liệu 16](#_Toc148402093)

[2.1.4. Bảo mật dữ liệu 16](#_Toc148402094)

[2.1.5. Quản lý tài khoản người dung 16](#_Toc148402095)

[2.1.6. Quản lý dữ liệu trực tuyến 16](#_Toc148402096)

[2.2. Xây Dựng RESTful API (C#) 17](#_Toc148402097)

[2.2.1. RESTful API là gì? 17](#_Toc148402098)

[2.2.2. Mục tiêu của RESTful API 17](#_Toc148402099)

[2.2.3. Ngôn ngữ lập trình C# 17](#_Toc148402100)

[2.2.4. Các phương thức HTTP 17](#_Toc148402101)

[2.2.5. Định dạng dữ liệu 17](#_Toc148402102)

[2.2.6. Xác thực và quản lý phiên làm việc 18](#_Toc148402103)

[2.2.7. Quản lý lỗi và xử lý ngoại lệ 18](#_Toc148402104)

[2.3. An Toàn và Quản Lý Tài Khoản 18](#_Toc148402105)

[2.3.1. Bảo mật dữ liệu 18](#_Toc148402106)

[2.3.2. Xác thực người dùng 18](#_Toc148402107)

[2.3.3. Bảo vệ khỏi tấn công 18](#_Toc148402108)

[2.3.4. Đăng ký và đăng nhập 18](#_Toc148402109)

[2.3.5. Quên mật khẩu 18](#_Toc148402110)

[2.3.6. Quản lý thông tin cá nhân 19](#_Toc148402111)

[2.3.7. Quản lý quyền truy cập 19](#_Toc148402112)

[2.3.8. Bảo mật mật khẩu 19](#_Toc148402113)

[2.4. JWT 19](#_Toc148402114)

[2.4.1. JWT là gì? 19](#_Toc148402115)

[2.4.2. JWT hoạt động như thế nào? 19](#_Toc148402116)

[2.4.3. Tại sao lại cần đến JWT? 20](#_Toc148402117)

[2.5. Swagger 20](#_Toc148402118)

[2.5.1. Swagger là gì? 20](#_Toc148402119)

[2.5.2. Swagger dùng như thế nào? 20](#_Toc148402120)

[2.5.3. Tại sao lại cần đến Swagger 21](#_Toc148402121)

[2.6. Bcrypt 21](#_Toc148402122)

[2.6.1. Bcrypt là gì? 21](#_Toc148402123)

[2.6.2. Bcrypt hoạt động như thế nào? 21](#_Toc148402124)

[2.6.3. Tại sao lại cần đến Bcrypt? 21](#_Toc148402125)

[2.7. AspNetCore.Authorization 22](#_Toc148402126)

[2.7.1. AspNetCore.Authorization là gì? 22](#_Toc148402127)

[2.7.2. AspNetCore.Authorization hoạt động như thế nào? 22](#_Toc148402128)

[2.7.3. Tại sao lại cần đến AspNetCore.Authorization? 22](#_Toc148402129)

[2.8. RestSharp 22](#_Toc148402130)

[2.8.1. RestSharp là gì? 22](#_Toc148402131)

[2.8.2. RestSharp hoạt động như thế nào? 23](#_Toc148402132)

[2.8.3. Tại sao lại cần đến RestSharp? 23](#_Toc148402133)

[2.9. Mailgun 23](#_Toc148402134)

[2.9.1. Mailgun là gì? 23](#_Toc148402135)

[2.9.2. Mailgun hoạt động như thế nào? 23](#_Toc148402136)

[2.9.3. Tại sao lại chọn Mailgun? 23](#_Toc148402137)

[Chương 3. PHÁT TRIỂN HỆ THỐNG 25](#_Toc148402138)

[3.1. Giới thiệu 25](#_Toc148402139)

[3.2. Kiến trúc hệ thống 25](#_Toc148402140)

[3.3. Phân tích thiết kế 25](#_Toc148402141)

[3.3.1. Sequence Diagram 25](#_Toc148402142)

[3.3.2. Thiết kế cơ sở dữ liệu 36](#_Toc148402143)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 42](#_Toc148402144)

[4.1. Kết luận 42](#_Toc148402145)

[4.1.1. Kiến thức tích lũy 42](#_Toc148402146)

[4.1.2. Ưu điểm 42](#_Toc148402147)

[4.1.3. Khuyết điểm 42](#_Toc148402148)

[4.2. Hướng phát triển 43](#_Toc148402149)

[TÀI LIỆU THAM KHẢO 44](#_Toc148402150)

[PHỤ LỤC 46](#_Toc148402151)

DANH MỤC TỪ VIẾT TẮT

API – Application Programming Interface

JSON – JavaScript Object Notation

BCrypt - Blowfish Crypt

REST – Representational Stat Transfer

DANH MỤC HÌNH VẼ

[Hình 3–1: Hình ảnh mô hình kiến trúc hệ thống 25](#_Toc148397436)

[Hình 3–2: Hình ảnh sơ đồ tuần tự chức năng đăng kí 26](#_Toc148397437)

[Hình 3–3: Hình ảnh sơ đồ tuần tự chức năng đăng nhập 27](#_Toc148397438)

[Hình 3–4: Hình ảnh sơ đồ tuần tự chức năng quên mật khẩu 28](#_Toc148397439)

[Hình 3–5: Hình ảnh sơ đồ tuần tự chức năng đổi mật khẩu 30](#_Toc148397440)

[Hình 3–6: Hình ảnh sơ đồ tuần tự chức năng thay đổi thông tin 31](#_Toc148397441)

[Hình 3–7: Hình ảnh sơ đồ tuần tự chức năng thêm cảnh báo 32](#_Toc148397442)

[Hình 3–8: Hình ảnh sơ đồ tuần tự chức năng kích hoạt cảnh báo 34](#_Toc148397443)

[Hình 3–9: Hình ảnh sơ đồ tuần tự chức năng danh sách cảnh báo được kích hoạt 35](#_Toc148397444)

[Hình 3–10: Hình ảnh sơ đồ tuần tự chức năng lấy danh sách cảnh báo 35](#_Toc148397445)

[Hình 3–11: Hình ảnh sơ đồ tuần tự chức năng đổi quyền hạn 36](#_Toc148397446)

[Hình 3–12: Hình ảnh sơ đồ ERD 37](#_Toc148397447)

[Hình 3–13: Hình ảnh sơ đồ cơ sở dữ liệu 38](#_Toc148397448)

DANH MỤC BẢNG

[Bảng 3.1: Bảng mô tả các trường dữ liệu thuộc bảng User 39](#_Toc148397449)

[Bảng 3.2: Bảng mô tả các trường dữ liệu thuộc bảng Role 40](#_Toc148397450)

[Bảng 3.3: Bảng mô tả các trường dữ liệu thuộc bảng Warning 40](#_Toc148397451)

[Bảng 3.4: Bảng mô tả các trường dữ liệu thuộc bảng Address 41](#_Toc148397452)

[Bảng 3.5: Bảng mô tả các trường dữ liệu thuộc bảng Coordinates 41](#_Toc148397453)

[Bảng 3.6: Bảng mô tả các trường dữ liệu thuộc bảng Province 42](#_Toc148397454)

[Bảng 3.7: Bảng mô tả các trường dữ liệu thuộc bảng District 42](#_Toc148397455)

MỞ ĐẦU

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu

Hiện nay trên địa bàn thành phố Hồ Chí Minh, lượng xe cá nhân ngày càng tăng nhưng hạ tầng không theo kịp, dẫn đến việc nhiều địa điểm xảy ra ùn tắc kéo dài. Thông tin từ Sở Giao thông Vận tải thành phố Hồ Chí Minh cho biết vào ngày 6 tháng 1địa bàn thành phố đã nâng tổng số địa điểm thường xuyên xảy ra kẹt xe lên tổng số 24 điểm [1].

Vấn đề kẹt xe trở thành một thách thức lớn đối với cuộc sống hàng ngày của mọi người. Việc bị kẹt xe không chỉ gây ra sự bất tiện mà còn tác động tiêu cực đến môi trường, thời gian và tình trạng tâm lý của người tham gia giao thông [2]. Vì vậy, để giải quyết vấn đề này, chúng tôi đã nghĩ đến việc phát triển một ứng dụng Android có chức năng cảnh báo các địa điểm kẹt xe gần khu vực của người sử dụng.

Một ứng dụng giúp người đi đường nhận cảnh báo về tình trạng kẹt xe trên các đoạn đường gần khu vực người sử dụng. Ứng dụng sử dụng API của Google Maps và các phương thức để xác định vị trí hiện tại của người dùng và cảnh báo họ về tình trạng giao thông qua thông báo trên điện thoại.

Để sử dụng ứng dụng này, người đi đường cần phải tạo một tài khoản để có thể đăng thông tin về tình trạng giao thông. Điều này giúp kiểm soát và tránh việc có người sử dụng ứng dụng với mục đích gây rối cho người đi đường khác. Ứng dụng sẽ gửi các cảnh báo được người dùng thêm về ban quản trị. Quản trị viên của ứng dụng sẽ nhận được thông tin khi có cảnh báo mới và kiểm tra thông tin qua hình ảnh và các thông tin đi kèm của cảnh báo. Sau đó sẽ quyết định cảnh báo có hoạt động hay không.

## Mục tiêu đề tài

### Thiết kế giao diện người dùng (UI/UX)

Cần tạo ra một giao diện thân thiện với người dùng và dễ sử dụng. Giúp mang đến trãi nghiệm tốt cho người dung, đảm bảo người dùng không cảm thấy khó chịu khi sử dụng ứng dụng.

### Phát triển tính năng cơ bản

Tạo ra các tính năng cơ bản như xác định vị trí hiện tại của người dùng, hiển thị thông tin về tình trạng giao thông, và cảnh báo kẹt xe. Đảm bảo tính năng hoạt động một cách chính xác và ổn định.

### Tích hợp cơ sở dữ liệu SQL Server

Tạo kết nối với cơ sở dữ liệu SQL Server để lưu trữ và quản lý thông tin về tình trạng giao thông. Bao gồm cả việc lưu trữ thông tin về tài khoản người dùng và thông tin đăng kẹt xe.

### Phát triển tính năng đăng nhận cảnh báo

Phát triển tính năng cho phép người dùng đăng thông tin về kẹt xe và nhận cảnh báo từ những người khác.

### Bảo mật và quản lý tài khoản

Bảo mật dữ liệu người dùng và quản lý tài khoản người dùng một cách an toàn.

## Lý do chọn đề tài

### Tính thực tiễn

Vấn đề kẹt xe là một thách thức phổ biến và đáng lo ngại trong cuộc sống đô thị ngày nay. Với tốc độ gia tăng của số lượng phương tiện và mật độ dân số tăng cao, ứng dụng như này có tiềm năng giúp cải thiện tình trạng giao thông.

### Tính ứng dụng rộng rãi

Ứng dụng cảnh báo kẹt xe có tiềm năng được sử dụng rộng rãi bởi mọi người tham gia giao thông, từ người lái xe hàng ngày đến những người cần di chuyển trong thành phố.

### Khả năng ứng dụng thực tế

Có thể trở thành một dự án thực tế và hữu ích cho cộng đồng. Nó có tiềm năng để được triển khai và sử dụng hàng ngày.

### Khám phá kiến thức mới

Việc phát triển một ứng dụng Android đòi hỏi sự tìm hiểu và áp dụng nhiều kiến thức về lập trình, giao diện người dùng, và quản lý dữ liệu, giúp mở rộng kiến thức và kỹ năng của người thực hiện.

### Tính đột phá

Nếu thành công, đề tài này có tiềm năng trở thành một ứng dụng có tính đột phá trong việc giải quyết vấn đề kẹt xe và cải thiện giao thông đô thị.

# CƠ SỞ LÝ THUYẾT

## Quản lý dữ liệu

Quản lý dữ liệu là quá trình tổ chức, lưu trữ, và quản lý thông tin trong một hệ thống hoặc tổ chức. Điều này bao gồm việc thu thập, lưu trữ, sắp xếp, xử lý, bảo mật và cung cấp dữ liệu một cách hiệu quả và an toàn.

### Thu thập dữ liệu

Quản lý dữ liệu bắt đầu bằng việc thu thập thông tin về tình trạng giao thông và kẹt xe được thực hiện thông qua dữ liệu từ người dùng ứng dụng.

### Lưu trữ dữ liệu

Dữ liệu thu thập cần được lưu trữ sao cho dễ dàng truy cập và sử dụng. Cơ sở dữ liệu có thể được sử dụng để lưu trữ thông tin về tình trạng giao thông, các điểm đánh dấu vị trí, và các thông tin khác liên quan đến đề tài.

### Xử lý dữ liệu

Dữ liệu thu thập thường cần được xử lý để trích xuất thông tin hữu ích và tạo ra báo cáo. Phân tích dữ liệu để xác định các điểm kẹt xe, tính toán thời gian ước tính để đi qua các tuyến đường, và hiển thị thông tin trên bản đồ.

### Bảo mật dữ liệu

Vì dữ liệu liên quan đến vị trí và giao thông có tính riêng tư và quan trọng, việc bảo mật dữ liệu là rất quan trọng.

### Quản lý tài khoản người dung

Thu thập và lưu trữ thông tin đăng nhập và quản lý quyền truy cập của người dùng.

### Quản lý dữ liệu trực tuyến

Quản lý và duy trì cơ sở dữ liệu trực tuyến, đảm bảo tính khả dụng và độ tin cậy của dữ liệu.

## Xây Dựng RESTful API (C#)

Việc xây dựng một RESTful API (Application Programming Interface) bằng ngôn ngữ C# là một phần quan trọng của quá trình phát triển ứng dụng.

### RESTful API là gì?

REST là viết tắt của "Representational State Transfer," là một kiến trúc dùng để thiết kế các dịch vụ web (web services) dựa trên giao thức HTTP. RESTful API là một loại API tuân theo các nguyên tắc và quy tắc của kiến trúc REST.

### Mục tiêu của RESTful API

Cung cấp một cách tiếp cận cho ứng dụng Android để truy xuất và tương tác với dữ liệu về tình trạng giao thông và thông tin kẹt xe. Cho phép người dùng đăng thông tin cảnh báo về kẹt xe thông qua giao diện API. Cung cấp các chức năng quản lý tài khoản và quyền truy cập cho người dùng ứng dụng di động.

### Ngôn ngữ lập trình C#

C# là một ngôn ngữ lập trình phát triển bởi Microsoft, được sử dụng phổ biến trong việc phát triển ứng dụng Windows và dịch vụ web. C# là một ngôn ngữ đa năng và mạnh mẽ, phù hợp cho việc xây dựng RESTful API.

### Các phương thức HTTP

RESTful API sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để thực hiện các thao tác truy vấn và tương tác với dữ liệu.

* GET: Sử dụng để truy xuất thông tin từ API.
* POST: Sử dụng để tạo mới dữ liệu hoặc thực hiện các thao tác khác.
* PUT: Sử dụng để cập nhật thông tin hiện có.
* DELETE: Sử dụng để xóa thông tin.

### Định dạng dữ liệu

RESTful API trả về dữ liệu dưới dạng JSON (JavaScript Object Notation) để dễ dàng truyền tải và phân tích bởi ứng dụng di động.

### Xác thực và quản lý phiên làm việc

Để bảo mật API, cần xác thực người dùng và quản lý phiên làm việc (session) của họ bao gồm việc sử dụng mã thông báo (token) để đảm bảo tính riêng tư và bảo mật.

### Quản lý lỗi và xử lý ngoại lệ

RESTful API cần có khả năng xử lý lỗi và trả về các mã trạng thái HTTP thích hợp để báo cáo kết quả của các yêu cầu API, ví dụ: mã 200 OK, mã 404 Not Found, mã 500 Internal Server Error, v.v.

## An Toàn và Quản Lý Tài Khoản

### Bảo mật dữ liệu

Dữ liệu về tình trạng giao thông và thông tin kẹt xe là quan trọng và có giá trị, vì vậy cần phải được bảo mật.

### Xác thực người dùng

Để đảm bảo tính riêng tư và an toàn, ứng dụng cần phải xác thực người dùng khi họ đăng nhập. Sử dụng một cơ chế xác thực như tên người dùng và mật khẩu hoặc mã thông báo (token) dựa trên phiên làm việc.

### Bảo vệ khỏi tấn công

Cần phải có các biện pháp bảo vệ khỏi các loại tấn công như tấn công SQL Injection, tấn công Cross-Site Scripting (XSS), và tấn công Cross-Site Request Forgery (CSRF). Điều này đòi hỏi việc kiểm tra và xử lý đầu vào người dùng cẩn thận.

### Đăng ký và đăng nhập

Quản lý tài khoản bao gồm việc cho phép người dùng đăng ký tài khoản mới và đăng nhập vào tài khoản hiện có.

### Quên mật khẩu

Cần cung cấp một cách cho người dùng khôi phục mật khẩu nếu họ quên mật khẩu của mình. Thực hiện thông qua việc gửi mật khẩu mới được tạo thông qua một phương thức và gửi đến email đã đăng ký của người dùng.

### Quản lý thông tin cá nhân

Người dùng cần có khả năng quản lý thông tin cá nhân của họ, bao gồm việc cập nhật thông tin liên hệ và thay đổi mật khẩu.

### Quản lý quyền truy cập

Ứng dụng cần có khả năng quản lý quyền truy cập của người dùng. Xác định những tính năng và dữ liệu mà mỗi người dùng có quyền truy cập.

### Bảo mật mật khẩu

Mật khẩu của người dùng cần được bảo mật bằng cách lưu trữ chúng trong dạng được mã hóa và sử dụng các biện pháp bảo vệ như đặt điều kiện về độ dài và phức tạp của mật khẩu.

## JWT

### JWT là gì?

JWT là viết tắt của Json Web Token là một tiêu chuẩn được định nghĩa trong RFC 7519 [3]. JWT được sử dụng để truyền tải thông tin giữa người dùng và máy chủ một cách đáng tin cậy và bảo mật thông qua một chuỗi JSON. Thông tin được thể hiện trong JWT được bảo mật rất cao nhờ vào chữ ký số được tạo ra bằng mã khóa công khai hoặc riêng tư.

Một JWT sẽ được chia ra làm ba phần :HEADER,PAYLOAD và VERIFY SIGNATURE. Trong đó HEADER là phần chưa thuật toán mã hóa ví dụ là HS256,HS512,… PAYLOAD là phần chứa các dữ liệu được cấu hình trước trong cài đặt. VERIFY SIGNATURE là chữ ký xác minh của token tạo ra [4].

### JWT hoạt động như thế nào?

Khi một client đăng nhập vào hệ thống thì lúc này hệ thống sẽ gửi thông tin đến server và server sẽ kiểm tra thông tin mà client gửi lên server xem có hợp lệ hay không. Khi hợp lệ thì server sẽ tạo ra token để đưa cho client mà mỗi lần client muốn gửi một yêu cầu gì đến server thì sẽ gửi kèm token đã được cho để server xác minh và thực hiện yêu cầu của client. Việc làm này đảm bảo tính an toàn khi xác thực giữa quá trình giao tiếp giữa client và server [5].

### Tại sao lại cần đến JWT?

**Bảo mật:** JWT giúp bảo vệ thông tin người dùng và quyền truy cập cá nhân. Nếu token bị đăng cắp thì nó chỉ có hiệu lực một thời gian ngắn giúp hạn chế sự lạm dụng.

**Đăng nhập một lần (Single Sign-on):** cho phép người dùng chỉ đăng nhập một lần và có thể truy cập các chức năng của server mà không cần đăng nhập lại.

**Truyền đạt và trao đổi thông tin:** JWT đảm bảo tính toàn vẹn của dữ liệu và xác minh danh tính người dùng khi gửi thông tin, giúp người dùng nhận biết được nguồn gốc thông tin và đảm bảo tính chính xác.

**An toàn và phổ biến:** JWT sử dụng các thuật toán mã hóa, dễ sử dụng và được hỗ trợ rộng rã trong nhiều ngôn ngữ lập trình [5].

## Swagger

### Swagger là gì?

Là một bộ công cụ mã nguồn mở để tạo ra OpenAPI specfications giúp trong việc thiết kế, tạo tài liệu và sử dụng REST APIs trở nên dễ hơn [6].

### Swagger dùng như thế nào?

Swagger được dùng để xác định api mà muốn thiết kế trong đó bao gồm các việc xác định endpoints, phương thức HTTP(GET,POST,PUT,DELETE).

Swagger được dùng để xây dựng, soạn thảo OpenAPI specification giúp mô tả API, cung cấp mô tả chi tiết về cách sử dụng API, chứa thông tin về các endpoints, tham số, phương thức, cấu trúc dữ liệu, các yêu cầu bảo mật, cách gửi các yêu cầu, cách xử lý lỗi, cách xác thực dưới dạng trang web hoặc tệp tài liệu

Cung cấp giao diện trực quan để kiểm tra API mà không cần sử dụng các công cụ ngoài như. Có thể gửi yêu cầu HTTP và xem kết quả trả về để kiểm tra được tính đúng đắn của API [7].

### Tại sao lại cần đến Swagger

Tiết kiện thời gian: Swagger cho phép tạo OpenAPI specification một cách tự động từ mã nguồn, giúp tiết kiệm thời gian và công sức trong việc viết tài liệu.

Kiểm tra trực tiếp: Không cần một công cụ ngoài nào để kiểm tra vì Swagger đã cung cấp sẵn một giao diện để kiểm tra.

Mã hóa tự động: có tính năng mã hóa tự động, xác thực cao giúp bảo vệ người dùng khỏi tấn công và các mối đe dọa bảo mật.  
Hỗ trợ đa ngôn ngữ: Hỗ trợ nhiều ngôn ngữ lập trình, cho phép tạo mã cho các ứng dụng và máy chủ bằng ngôn ngữ lập trình mong muốn.

Single Sign-On(Đăng nhập một lần): Cho phép đăng nhập một lần mà không cần phải đăng nhập lại cho phép truy cập nhiều hệ thống khác nhau [7].

## Bcrypt

### Bcrypt là gì?

Bcrypt là một chức năng mã hóa mật khẩu được thiết kế bởi Niels Provos và David Mazières dựa trên các thuật toán mã hóa blowfish giúp ngăn chặn được các cuộc tấn công Brute-force hoặc Dictionary attack [8].

### Bcrypt hoạt động như thế nào?

Bcrypt sẽ tạo ra Salt – một chuỗi ngẫu nhiện dể kết hợp với mật khẩu trước khi tiến hành băm ra. Sau khi tạo Salt xong thì bắt đầu tới giai đoạn là Rounds – đây là số lần lặp để quá trình băm mật khẩu trở nên chậm hơn và tăng tính bảo mật hơn thông thường giá trị mặc định của Rounds là 10 tức là 2^10 hay 1024 vòng lặp để băm mật khẩu. Sau đó là tiền hành băm mật khẩu dựa trên mật khẩu đã nhập vào, Salt đã tạo ra ở trên và số vòng băm sẽ tiến hàm băm mật khẩu dựa trên thuật toán Blowfish [8].

### Tại sao lại cần đến Bcrypt?

Dễ dàng chống các cuộc tấn công Brute-force và tấn công Dictionary attack. Có tăng cấp bảo mật lên tức là tăng số vòng lặp lên để tăng tính bảo mật hơn so với bình thường mà không cần thay đổi cơ sở dữ liệu mật khẩu có. Mã hóa mạnh mẽ vì khi mỗi lần chạy sẽ tạo ra chuỗi Salt ngẫu nhiên cho mỗi lần băm và sử dụng thuật toán Blowfish mạnh mẽ giúp cho việc mật khẩu được bảo mật một cách an toàn nhất. Có khả năng mở rộng và có thể tích hợp dễ dàng trong nhiều dự án khác nhau [8].

## AspNetCore.Authorization

### AspNetCore.Authorization là gì?

Là một phần của ASP.NET Core là một framework phát triển ứng dụng của Microsoft. Cung cấp các tính năng liên quan đến việc xác thực và kiểm tra quyền [9].

### AspNetCore.Authorization hoạt động như thế nào?

Trước khi kiểm tra quyền truy cập hệ thống thì AspNetCore.Authorization sẽ xác minh danh tính người dùng sau khi xác minh thành công thì sẽ tới xác minh quyền truy cập hệ thông thống qua middleware đã được cài đặt trước để được sử dụng những chức năng chỉ dành cho quyền đó còn ngược lại nếu không thì sẽ trả về phản hồi là không có quyền đó [9].

### Tại sao lại cần đến AspNetCore.Authorization?

Mang tính bảo mật giúp kiểm soát và quản lý quyền truy cập vào các chức năng của ứng dụng. Có thể kiểm tra vai trò của người dùng để cấp quyền truy cập vào các chức năng đã được cài đặt từ trước. Điều này rất hữu ích trong việc quản lý các chức năng và tài nguyên của hệ thông. Có thể quản lý các chức năng, các hành động truy cập vào tài nguyên của hệ thống giúp xác định được ai đang thực hiện các chức năng gì. Dễ dàng tích hợp đối với cái dự án sử dụng ASP.NET CORE dễ dàng tích hợp vào để sử dụng tính năng bảo mật [9].

## RestSharp

### RestSharp là gì?

Là một thư viện C# được sử dụng để tạo và thực hiện các yêu cầu HTTP đối với các dịch vụ web dựa trên kiến trúc REST cung cấp một cách thuận tiện để tạo các yêu cầu HTTP và xử lý phản hồi từ các dịch vụ RESTful [10].

### RestSharp hoạt động như thế nào?

Khởi tạo RestClient để để đại diện cho endpoint cả dịch vụ web RESTful, nơi mà muốn gửi yêu cầu HTTP. Sau khi khởi tạo xong thì bắt đầu tạo RestRequest để xác định yêu cầu HTTP cụ thể muốn gửi đến dịch vụ web, Trong RestRequest có thể thiết lập các phương thức HTTP bình thường (GET, POST, PUT, DELETE).Sau khi cài đặt xong thì bắt đầu thực hiện yêu cầu sử dụng RestClient và RestRequest để gửi yêu cầu tới web dịch vụ. Lúc này sẽ nhận được phản hồi từ dịch vụ web thì bắt đầu xử lý [10].

### Tại sao lại cần đến RestSharp?

Dễ dàng sử dụng RestSharp cung cấp một giao diện đơn giản và dễ dàng sử dụng cho việc tạo và thực hiện các yêu cầu HTTP. Tích hợp dễ dàng đối với các ứng dụng, dự án .NET và tích hợp RESTful API để dễ dàng gửi các yêu cầu theo kiến trúc REST. RestSharp hỗ trợ xác thực khác nhau như Basic Authentication, Oauth và các phương thức xác thực tùy chỉnh khác [10].

## Mailgun

### Mailgun là gì?

Mailgun là một dịch vụ cung cấp các giải pháp email và thông báo dựa trên API cho các ứng dụng và các trang web. Dịch vụ này được sử dụng để gửi, nhận và quản lý email và thông báo qua API [11].

### Mailgun hoạt động như thế nào?

Mailgun là một dịch vụ api nên cho phép ta gửi mail, nhập mail qua API việc này khiến công việc gửi mail nhanh chóng hơn và có thể dễ dàng quản lý email đến từ người dùng hoặc dịch vụ khác. Cung cấp tính năng xác thực để đảm bảo email không bị xác minh sai và bị gắn mác như email spam, email rác. Theo dõi và thông kê số số email đã gửi, kiểm tra tình trạng email, thông kế việc email đã được mở, email bị hủy đăng ký và nhiều thông tin khác [11].

### Tại sao lại chọn Mailgun?

Trong vô vàn các dịch vụ gửi email thì mailgun dễ dàng sử dụng, tùy chỉnh mạnh mẽ theo ý muốn mình, quản lý được email hệ thống đã gửi, tích hợp sẵn theo dõi và thông kê số lượng email một cách chi tiết và có thể tích hợp một cách dễ dàng vào nhiều ứng dụng. Dịch vụ và giá cả rất phù hợp từ miễn phí cho tới các dự án nhỏ đến lớn [11].

# PHÁT TRIỂN HỆ THỐNG

## Giới thiệu

Hệ thống cảnh báo kẹt xe được xây dựng dựa trên việc lưu trữ dữ liệu trong SQL Server kết hợp với ASP.Net Core API để làm máy chủ và ứng dụng di động sử dụng các thư viện, công cụ và dịch vụ có sẵn để được xem như máy con trong mô hình mạng máy tính client-server.

Máy chủ còn tích hợp JWT, Bcrypt, dịch vụ Sendmail để giúp hệ thống hoàn thiện hơn về mặt bảo mật.

Về phần máy con tích hợp bộ công cụ Google Maps API, cloudinary và các thư viện hỗ trợ khác để có thể hoàn thiện hơn về cảm giác trải nghiệm của người dùng.

## Kiến trúc hệ thống

A diagram of a computer system

Description automatically generated

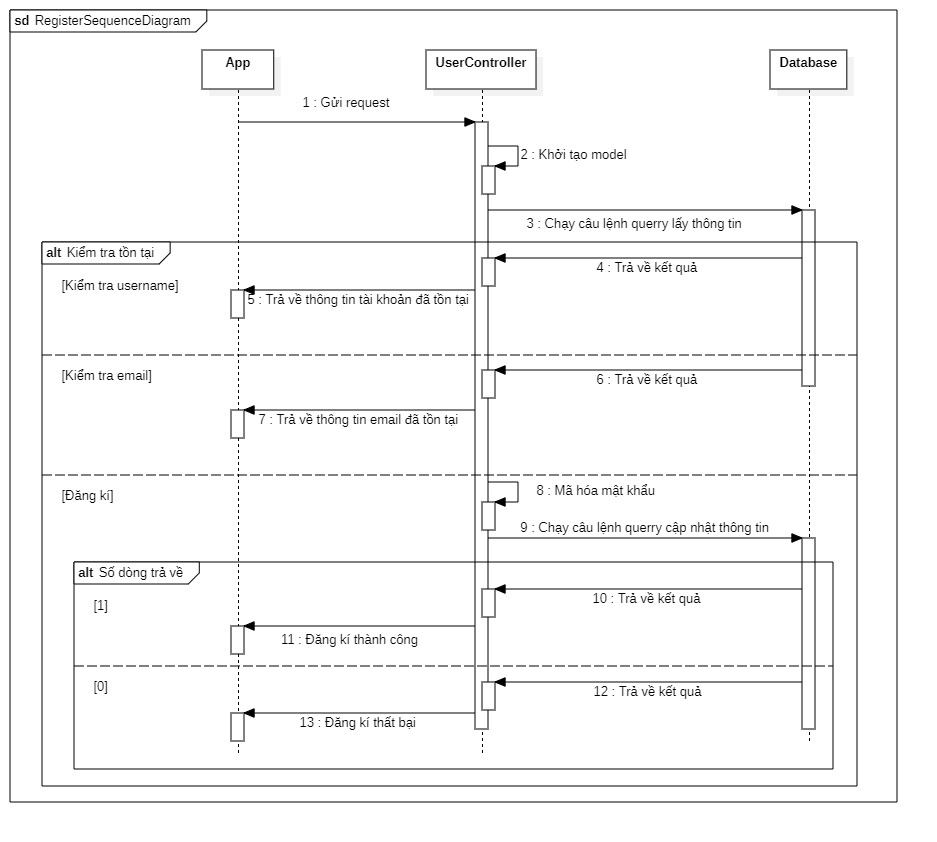
Hình –: Hình ảnh mô hình kiến trúc hệ thống

## Phân tích thiết kế

### Sequence Diagram

#### Mô tả sơ đồ tuần tự chức năng đăng kí

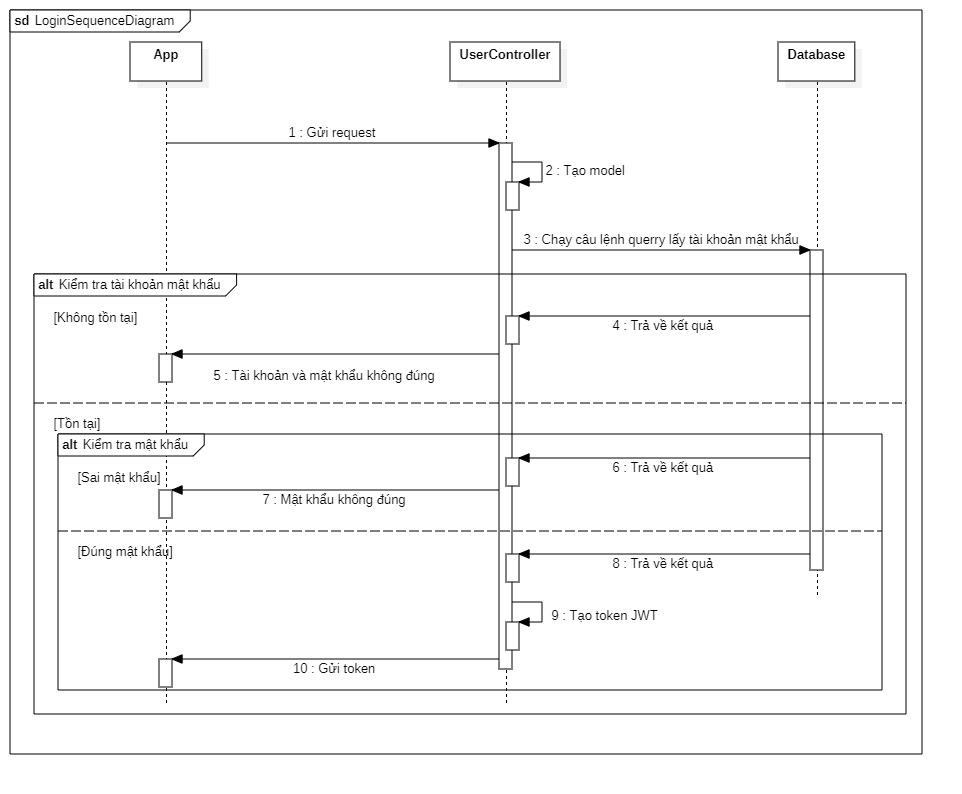
* Ứng dụng gửi yêu cầu đăng kí kèm thông tin đăng kí
* Khởi tạo model sau đó lấy thông tin đăng kí đưa vào model
* Chạy câu lệnh querry để thông tin username và email trong database
  + Trường hợp có kết quả: gửi thông báo cho ứng dụng username hoặc email đã tồn tại
  + Trường hợp không có kết quả thì bắt đầu mã hóa mật khẩu nhập lưu thông tin đăng kí vào database
    - Trường hợp thành công: Gửi thông báo thành công về cho ứng dụng
    - Trường hợp không thành công: Gửi thông báo không thành công về cho ứng dụng



Hình –: Hình ảnh sơ đồ tuần tự chức năng đăng kí

#### Mô tả sơ đồ tuần tự đăng nhập

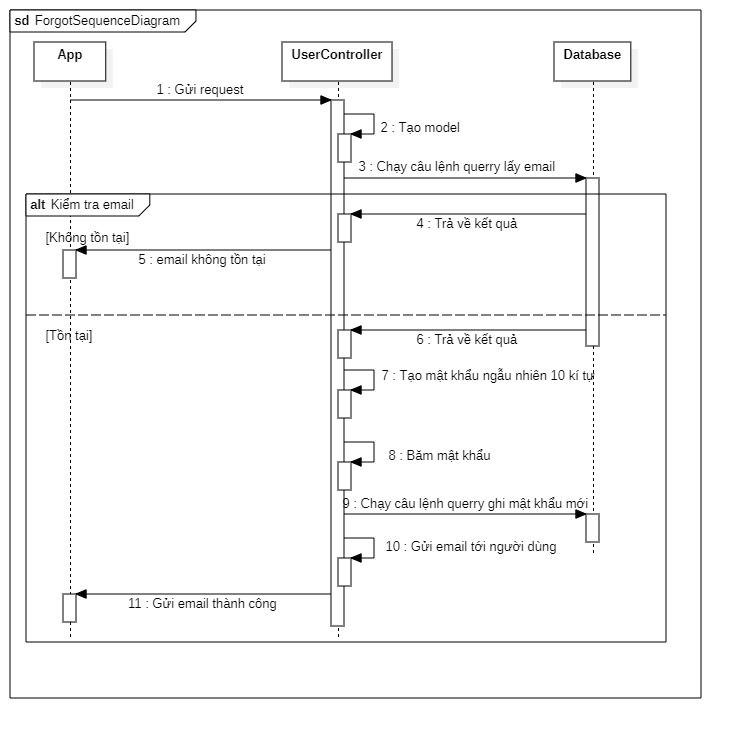
* Ứng dụng gửi yêu cầu đăng nhập kèm thông tin
* Khởi tạo model đăng nhập
* Chạy câu lệnh querry lấy thông tin tài khoản
  + Trường hợp không tồn tại: Gửi thông báo tài khoản và mật khẩu không đúng
  + Trường hợp tồn tại:
    - Trường hợp mật khẩu sai: sẽ gửi thông báo sai mật khẩu về cho ứng dụng
    - Trường hợp mật khẩu đúng: tạo token và gửi token cho ứng dụng



Hình –: Hình ảnh sơ đồ tuần tự chức năng đăng nhập

#### Mô tả tuần tự sơ đồ quên mật khẩu

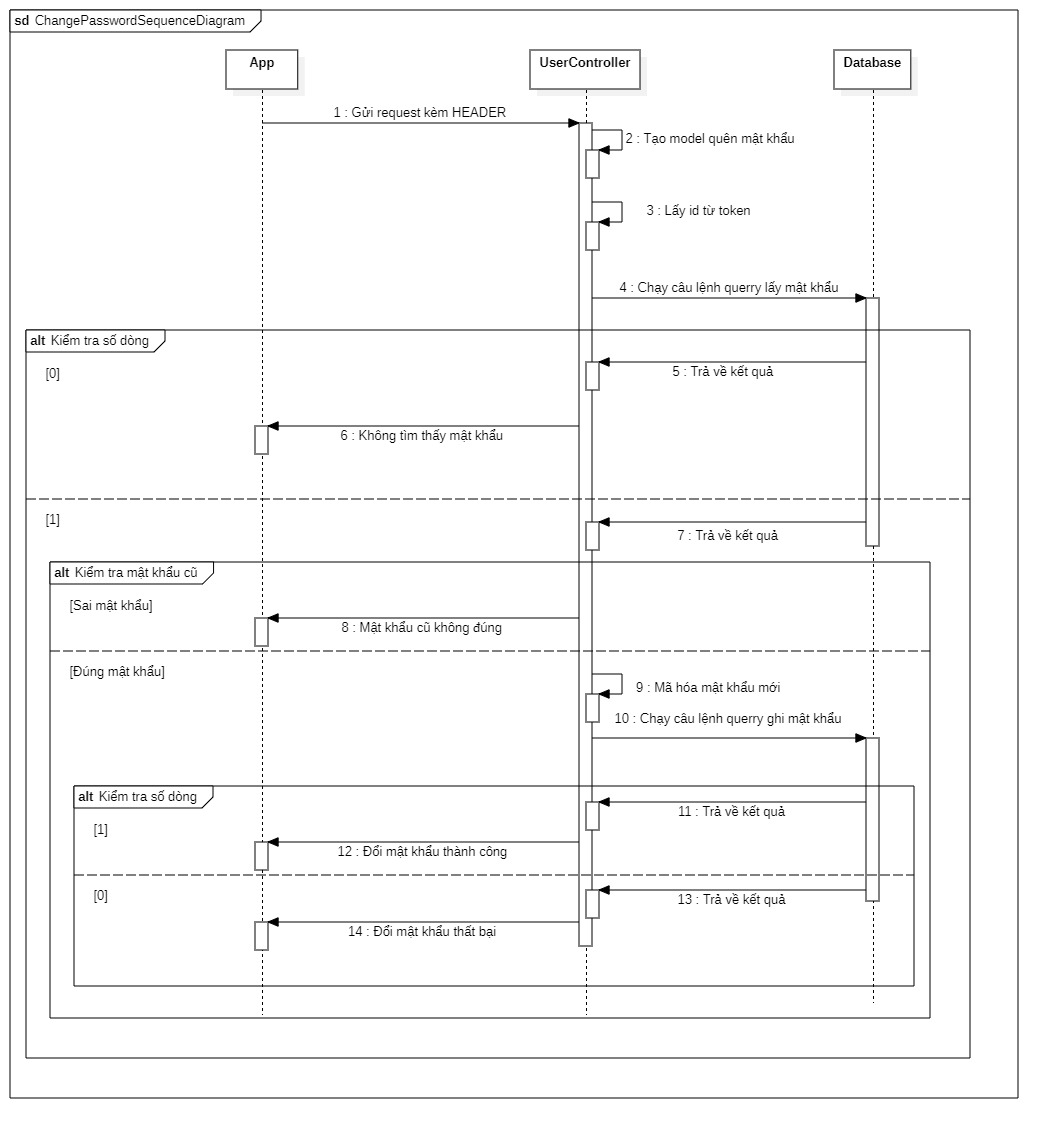
* Ứng dụng gửi yêu cầu quên mật khẩu kèm thông tin
* Tạo model quên mật khẩu
* Chạy câu lệnh querry lấy email từ database
  + Trường hợp email không tồn tại: gửi thông báo email không tồn tại về ứng dụng
  + Trường hợp email tồn tại:
    - Tạo ngẫu nhiên mật khẩu 10 kí tự
    - Băm khẩu đã tạo ở trên
    - Chạy câu lệnh querry ghi mật khẩu mới vào database
    - Gửi email tới người dùng
    - Thông báo tới ứng dụng gửi email thành công



Hình –: Hình ảnh sơ đồ tuần tự chức năng quên mật khẩu

#### Mô tả sơ đồ tuần tự đổi mật khẩu

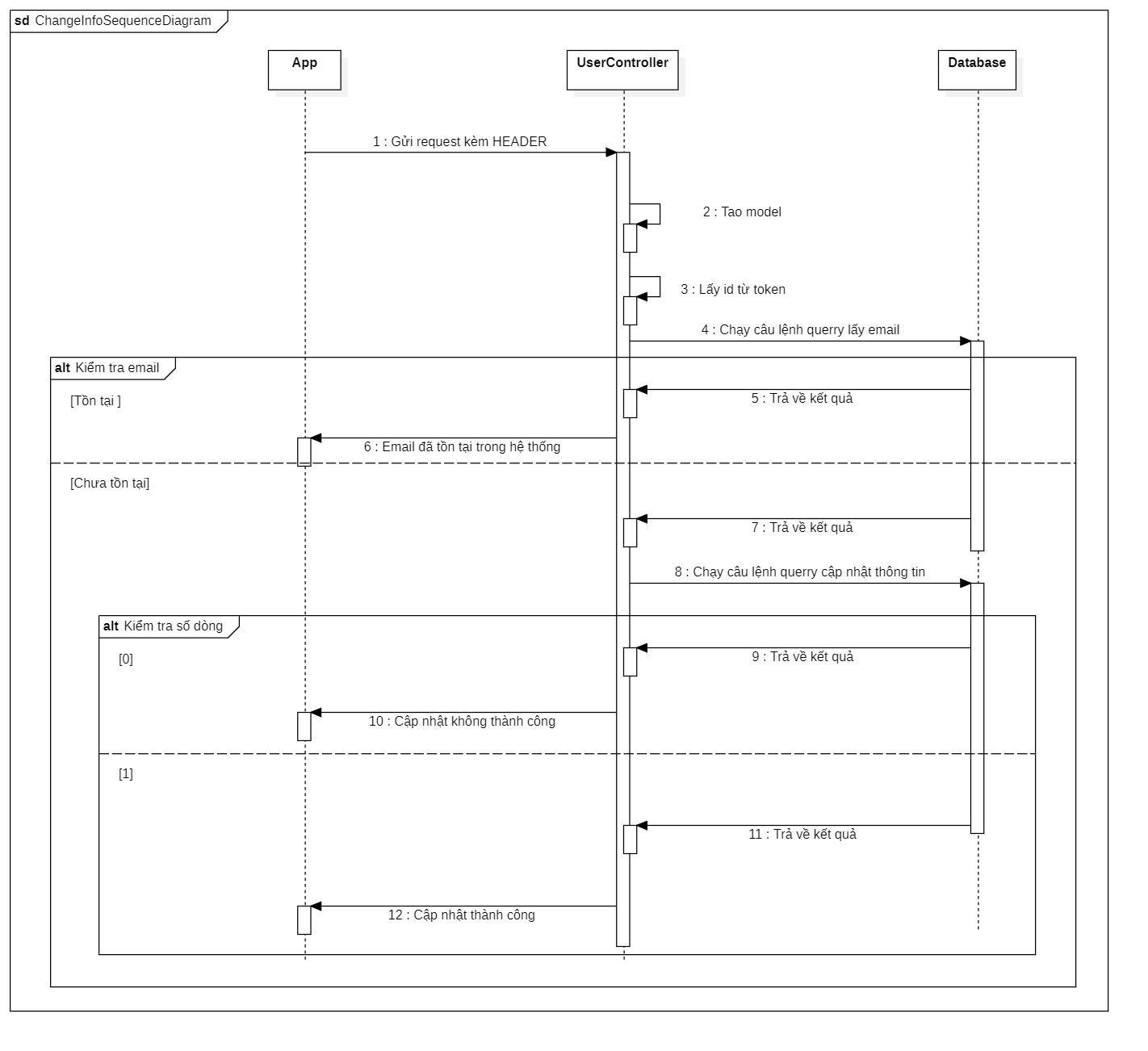
* Ứng dụng gửi yêu cầu đổi mật khẩu kèm thông tin
* Tạo model đổi mật khẩu
* Chạy câu lệnh querry lấy mật khẩu từ database
  + Trường hợp email không tồn tại: gửi thông báo mật khẩu không tồn tại về ứng dụng
  + Trường hợp mật khẩu tồn tại:
    - Trường hợp mật khẩu sai: gửi thông báo sai mật khẩu tới ứng dụng
    - Trường hợp mật khẩu đúng:
      * Mã hóa mật khẩu mới
      * Chạy câu truy vấn ghi mật khẩu mới vào database
        + Trường hợp có thay đổi: Gửi thông báo đổi mật khẩu thành công về ứng dụng
        + Trường hợp không thay đổi: Gửi thông báo đổi mật khẩu không thành công về ứng dụng



Hình –: Hình ảnh sơ đồ tuần tự chức năng đổi mật khẩu

#### Mô tả sơ đồ tuần tự đổi thông tin

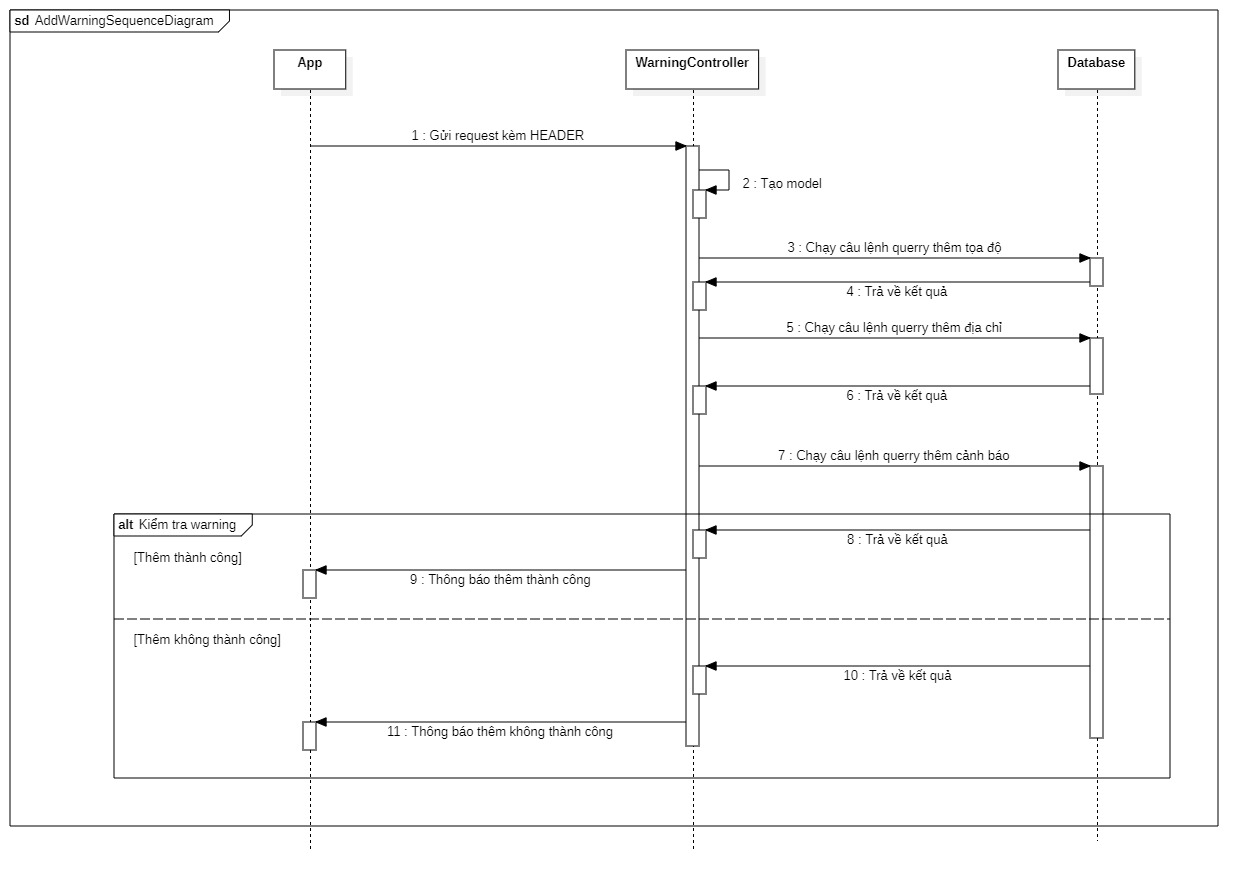
* Ứng dụng gửi yêu cầu đổi thông tin người dùng kèm thông tin
* Tạo model đổi thông tin
* Chạy câu lệnh querry lấy email từ database
  + Trường hợp email tồn tại: gửi thông báo email tồn tại về ứng dụng
  + Trường hợp email không tồn tại:
    - Trường hợp cập nhật thành công: gửi thông báo cho ứng dụng đã cập nhật thành công
    - Trường hợp cập nhật không thành công: gửi thông báo cho ứng dụng đã cập nhật không thành công



Hình –: Hình ảnh sơ đồ tuần tự chức năng thay đổi thông tin

#### Mô tả sơ đồ tuần tự thêm cảnh báo

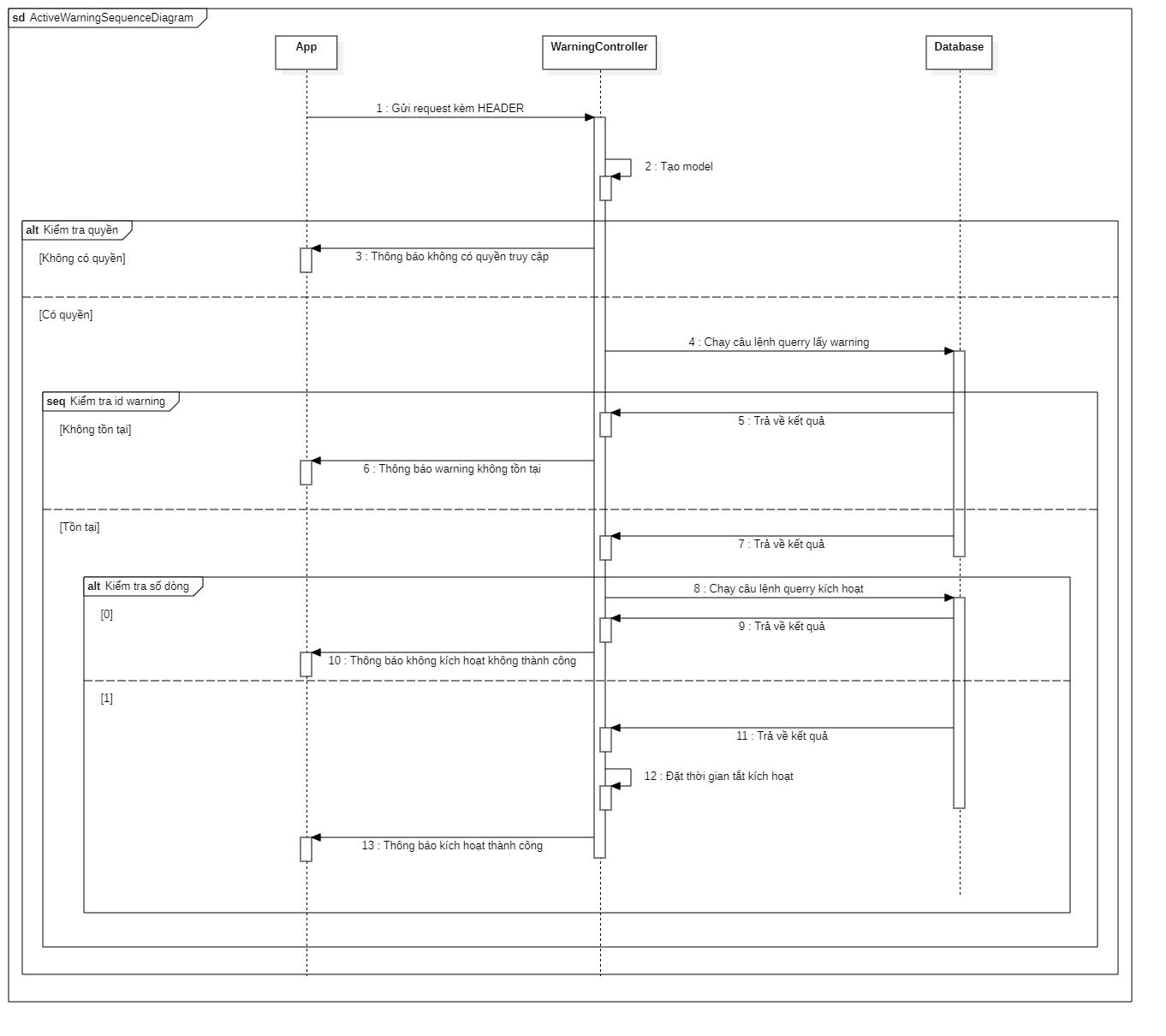
* Ứng dụng gửi yêu cầu thêm cảnh bảo kèm thông tin
* Tạo model thêm cảnh bảo
* Chạy câu lệnh querry thêm tọa độ
* Chạy câu lệnh querry thêm địa chỉ
* Chạy câu lệnh thêm cảnh báo
  + Trường hợp thêm thành công: gửi thông báo về ứng dụng thêm cảnh báo thành công
  + Trường hợp thêm không thành công: gửi thông báo về ứng dụng thêm không thành công



Hình –: Hình ảnh sơ đồ tuần tự chức năng thêm cảnh báo

#### Mô tả sơ đồ tuần tự kích hoạt cảnh báo

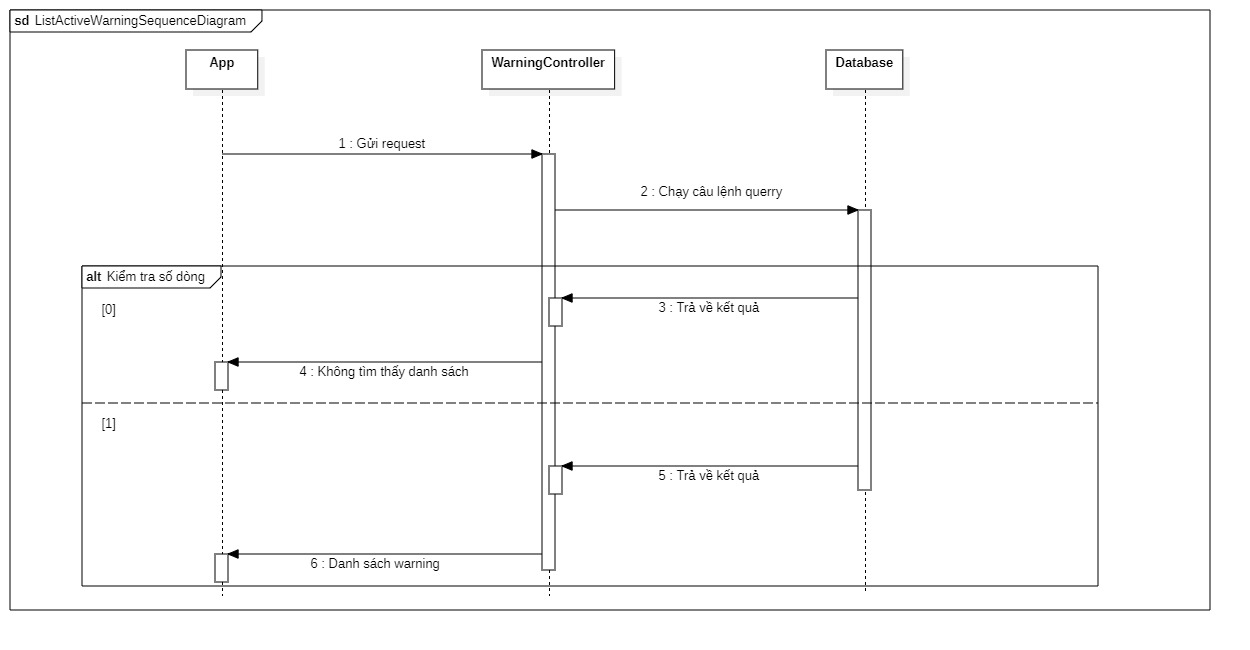
* Ứng dụng gửi yêu cầu kích hoạt thông báo kèm thông tin
* Tạo model kích hoạt thông báo
* Trường hợp không có quyền truy cập : Đưa ra thông báo cho ứng dụng không có quyền truy cập
* Trường hợp có quyền truy cập:
  + Chạy câu lệnh querry lấy id cảnh báo từ database
  + Trường hợp id cảnh báo không tồn tại: Đưa ra thông báo cho ứng dụng cảnh báo không tồn tại
  + Trường hợp id tồn tại:
    - Chạy câu lệnh querry kích hoạt cảnh báo đến database
      * Trường hợp kích hoạt không thành công: Đưa ra thông báo cho ứng dụng cảnh báo kích hoạt không thành công
      * Trường hợp kích hoạt thành công:
        + Đặt thời gian để tắt kích hoạt
        + Đưa ra thông báo cho ứng dụng kích hoạt cảnh báo thành công



Hình –: Hình ảnh sơ đồ tuần tự chức năng kích hoạt cảnh báo

#### Mô tả sơ đồ tuần tự chức năng đưa ra danh sách cảnh báo được kích hoạt

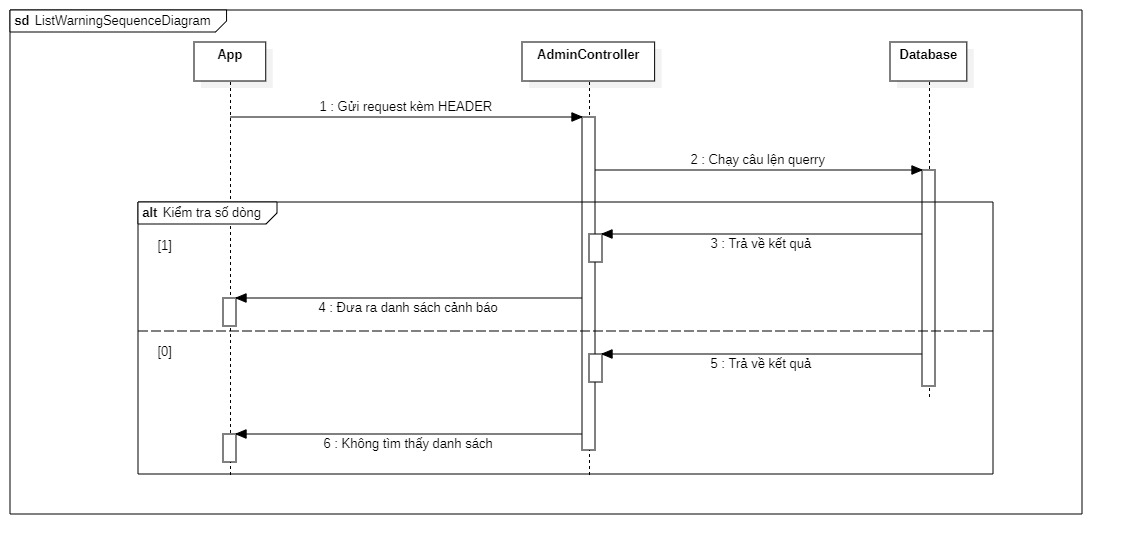
* Ứng dụng gửi yêu cầu lấy danh sách cảnh báo đã được kích hoạt
* Chạy câu lệnh querry lấy danh sách từ database
  + Trường hợp không tìm thấy danh sách: gửi thông báo về ứng dụng không tìm thấy danh sách
  + Trường hợp tìm thấy danh sách: gửi danh sách cảnh báo đã được kích hoạt



Hình –: Hình ảnh sơ đồ tuần tự chức năng danh sách cảnh báo được kích hoạt

#### Mô tả sơ đồ tuần tự danh sách cảnh báo

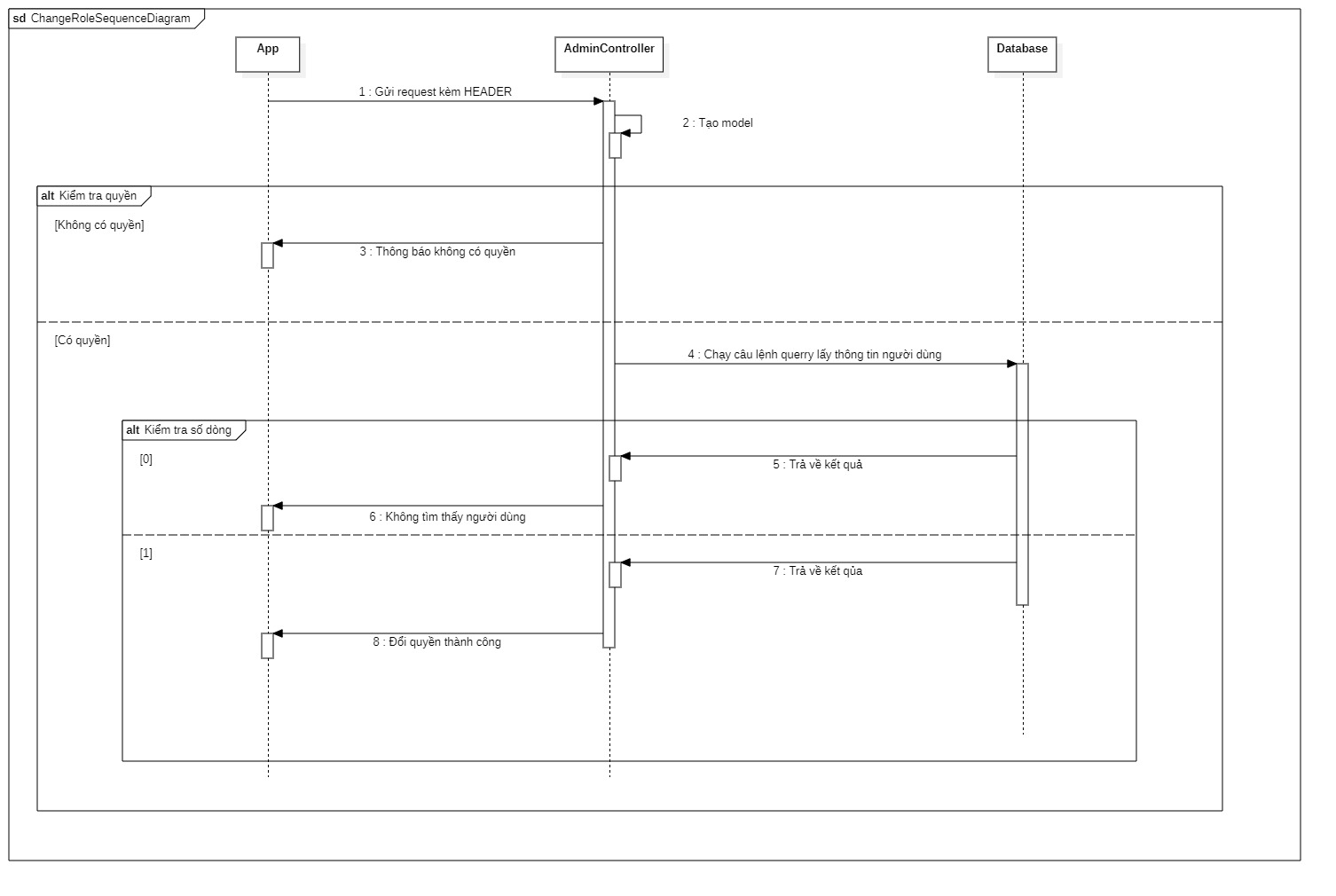
* Ứng dụng gửi yêu cầu lấy danh sách cảnh báo kèm thông tin
* Chạy câu lệnh querry lấy danh sách từ database
  + Trường hợp danh sách tồn tại: gửi danh sách cảnh báo về ứng dụng
  + Trường hợp danh sách không tồn tại: gửi thông báo về ứng dụng không tìm thấy danh sách



Hình –: Hình ảnh sơ đồ tuần tự chức năng lấy danh sách cảnh báo

#### Mô tả sơ đồ tuần tự đổi quyền hạn

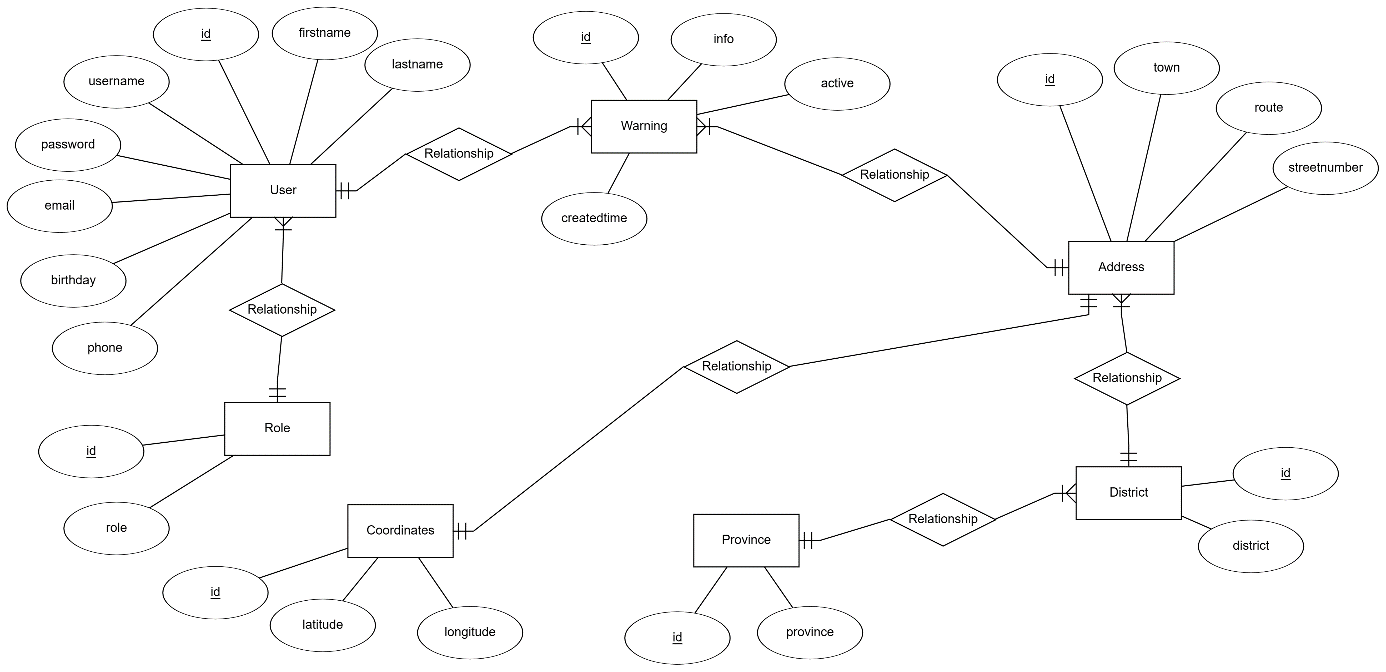
* Ứng dụng gửi yêu cầu đổi quyền hạn kèm thông tin
* Tạo model đổi quyền
* Trường hợp không có quyền truy cập : Đưa ra thông báo cho ứng dụng không có quyền truy cập
* Trường hợp có quyền truy cập:
  + Chạy câu lệnh querry lấy thông tin người dùng
    - Trường hợp không tìm thấy người dùng: Đưa ra thông báo cho ứng dụng không tìm thấy người dùng
    - Trường hợp tìm thấy người dùng: Đưa ra thông báo cho ứng dụng đổi quyền hạn thành công



Hình –: Hình ảnh sơ đồ tuần tự chức năng đổi quyền hạn

### Thiết kế cơ sở dữ liệu

#### Sơ đồ ERD



Hình –: Hình ảnh sơ đồ ERD

#### Mô hình quan hệ RD

User (id, firstname, lastname, username, password, email, phone, birthday, #roleid)

Warning (id, info, createdtime, active, #userid, #addressid)

Address (id, town, route, streetnumber, #districtid, #coordinatesid)

District (id, district, #provinceid)

Province (id, province)

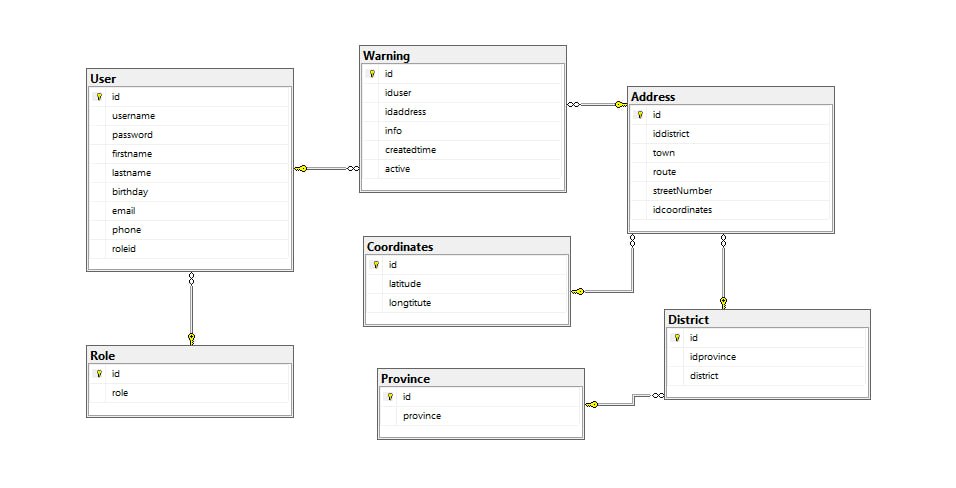
Coordinates (id, latitude, longitude)

Role (id, role)

#### Phân tích mối quan hệ ràng buộc

* Mối quan hệ 1-n từ bảng **User** đến bảng **Role**, đem khóa chính của bảng **Role** làm khóa phụ trong bảng **User** để nhận biết trường hợp một người dùng chỉ có một quyền hạn duy nhất và một quyền hạn có thể có nhiều dùng.
* Mối quan hệ 1-n từ bảng **User** đến bảng **Warning**, đem khóa chính của bảng **User** làm khóa phụ trong bảng **Warning** để nhận biết trường hợp một cảnh báo chỉ được đăng bởi một người dùng nhưng một người dùng có thể đăng được nhiều cảnh báo.
* Mối quan hệ 1-n từ bảng **Warning** đến bảng **Address**, đem khóa chính của bảng **Address** làm khóa phụ của bảng **Warning** để biết được rằng một cảnh báo thì có thể có nhiều địa chỉ nhưng một địa chỉ chỉ có một cảnh báo.
* Mối quan hệ 1-1 từ bảng **Coordinates** đến bảng **Address**, đem khóa chính của bảng **Coordinates** thành khóa phụ của bảng **Address** để biết được rằng một địa chỉ chỉ có duy nhất một tọa độ và ngược lại.
* Mối quan hệ 1-n từ bảng **Address** đến bảng **District**, đem khóa chính bảng **District** làm khóa phụ bảng **Address** để biết được rằng một quận thì có nhiều địa chỉ nhưng một địa chỉ thì chỉ có một quận.
* Mối quan hệ 1-n từ bảng **District** đến bảng **Province**, đem khóa chính của bảng **Province** làm khóa ngoại của bảng bảng **District** để biết được rằng một tỉnh thì có nhiều quận nhưng một quận thì chỉ thuộc trong một tỉnh.

#### Sơ đồ cơ sở dữ liệu



Hình –: Hình ảnh sơ đồ cơ sở dữ liệu

#### Các bảng trong cơ sở dữ liệu

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của người dùng. |
| Username | Nvarchar (Not Null) | Lưu trữ tên đăng nhập của người dùng. |
| Password | Nvarchar (Not Null) | Lưu trữ mật khẩu sau khi đã đường mã hóa của người dùng. |
| Firstname | Nvarchar (Not Null) | Lưu trữ tên người dùng. |
| Lastname | Nvarchar (Not Null) | Lưu trữ họ và tên lót của người dùng. |
| Birthday | Date (Null) | Lưu trữ ngày sinh của người dùng. |
| Email | Nvarchar (Not Null) | Lưu trữ email cá nhân của người dùng, phục vụ cho việc cấp lại mật khẩu mới khi người dùng quên mật khẩu. |
| Phone | Numberic (Not Null) | Lưu trữ số điện thoại của người dùng, phục vụ cho mục tiêu phát triển các tính năng mới. |
| RoleId | Int (Not Null) | Lưu trữ id của trường dữ liệu trong bảng Role, phục vụ cho việc phát triển các tính năng mới liên quan đến phân quyền người dùng. |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng User

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của phân quyền. |
| Role | Nvarchar (Not Null) | Lưu trữ tên phân quyền |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng Role

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của cảnh báo. |
| Iduser | Int (Not Null) | Trường dữ liệu được liên kết với bảng User dùng để lưu trữ việc người dùng nào đã thêm cảnh báo. |
| Idaddress | Int (Not Null) | Lưu trữ địa chỉ của cảnh báo. |
| Info | Nvarchar (Not Null) | Lưu trữ đường dẩn của hình ảnh minh chứng của cảnh báo. |
| Createdtime | Datetime (Not Null) | Lưu trữ ngày giờ khi cảnh báo đường thêm. |
| Active | Bit (Not Null) | Lưu trữ trạng thái của cảnh báo có được kích hoạt hay không. |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng Warning

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của địa chỉ. |
| Iddistrict | Int (Not Null) | Trường dữ liệu được liên kết với bảng Province dùng để lưu trữ việc địa chỉ thuộc quận huyện của tỉnh. |
| Town | Nvarchar (Not Null) | Lưu trữ thông tin xã, phường, thị trấn của địa chỉ. |
| Route | Nvarchar (Not Null) | Lưu trữ thông tin tên đường của địa chỉ. |
| Streetnumber | Nvarchar (Not Null) | Lưu trữ số nhà của địa chỉ, kể cả dấu “/” của các ngỏ ngách. |
| Idcoordinates | Int (Not Null) | Trường dữ liệu được liên kết với bảng Coordinates dùng để lưu trữ tọa độ của địa chỉ. |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng Address

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của tọa độ. |
| Latitude | Decimal (Not Null) | Lưu trữ vĩ độ của tọa độ. |
| Longitude | Decimal (Not Null) | Lưu trữ kinh độ của tọa độ |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng Coordinates

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của tỉnh. |
| Province | Nvarchar (Not Null) | Lưu trữ thông tin của 63 tỉnh thành Việt Nam. |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng Province

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Công dụng |
| Id | Int (Not Null) | Trường dữ liệu tự động tăng khi thêm một hàng vào cơ sở dữ liệu. Lưu trữ thông tin id của quận huyện. |
| Idprovince | Int (Not Null) | Trường dữ liệu được liên kết với bảng Province dùng để lưu trữ việc quận huyện thuộc địa phận tỉnh. |
| District | Nvarchar (Not Null) | Lưu trữ 612 quận huyện thuộc các tỉnh thành của Việt Nam theo Tổng cục Thống kê 2019 |

Bảng .: Bảng mô tả các trường dữ liệu thuộc bảng District

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Xây dựng được hệ thống hoàn thiện kết hợp giữa ứng dụng di động và RESTful API để hiện thị các thông tin cảnh báo trong thời gian thực. Hoàn thiện các chức năng để người dùng tương tác với hệ thống một cách thân thiện chức năng người dùng có trải nghiệm tuyệt với khi sử dụng ứng dụng. RESTful API hoàn thiện được các yêu cầu cơ bản để lưu trữ dữ liệu cùng thông tin của người dùng. Bảo mật được thông cá nhân người sử dụng.

### Kiến thức tích lũy

* Học được thêm về JWT
* Học được thêm về mã hóa mật khẩu Bcrypt
* Học được thêm về cách Swagger hoạt động như thế nào
* Học được thêm về cách phân quyền hoạt động như thế nào trong dự án ASP.NET CORE
* Học được thêm về cách đưa một cơ sở dữ liệu lên online để dễ dàng tương tác

### Ưu điểm

* Áp dụng được kiến thức đã học vào đồ án
* Bảo vệ được được thông tin và dữ liệu truyền đi thông qua JWT
* Mã hóa được mật khẩu người dùng
* Tăng thêm kỹ năng sử dụng phần mềm SSMS
* Phân quyền được các chức năng cần thiết
* Thêm được các kỹ năng sử dụng một số dịch vụ gửi email từ bên cung cấp
* Làm quen được thêm các chức năng của Swagger

### Khuyết điểm

* Chưa làm được giao diện quản lý dành cho admin
* Chưa tự động tắt cảnh báo theo thời gian thực khi đã hết
* Chưa phát triển được trang web riêng để thực hiện các chức năng API phải dựa trên giao diện có sẵn của Swagger
* Chưa làm đúng chuẩn CRUD của HTTP
* Cơ sở dữ liệu còn sơ sài chưa đầy đủ

## Hướng phát triển

* Phát triển được giao diện để thực hiện các chức năng API dễ dàng hơn
* Phát triển được hệ thống tự động bật tắt cảnh bảo theo thời gian thực mà không cần phải đặt thời gian sẵn trong hệ thống
* Kết hợp được thêm các cơ sở dữ liệu khác để cải thiện tốc độ truy xuất dữ liệu

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Gia Minh, "VNEXPRESS," FPT, 7 1 2023. [Online]. Available: https://vnexpress.net/tp-hcm-phat-sinh-6-diem-un-tac-4556971.html. [Accessed 9 10 2023]. |
| [2] | MinhTriND, "TinhTe," 29 7 2017. [Online]. Available: https://tinhte.vn/thread/ban-co-biet-ket-xe-la-gi.2714027/. [Accessed 2 9 2023]. |
| [3] | Jones, et al, "Datatracker," 5 2015. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7519. [Accessed 2 9 2023]. |
| [4] | Khoa Bui , "viblo," 29 5 2023. [Online]. Available: https://viblo.asia/p/cung-minh-tim-hieu-ve-json-web-token-jwt-Rk74axvAVeO. [Accessed 2 9 2023]. |
| [5] | ITNavi, "ITNavi," 2 11 2021. [Online]. Available: https://itnavi.com.vn/blog/jwt-la-gi-nhung-vai-tro-va-thanh-phan-chinh-cua-jwt. [Accessed 2 9 2023]. |
| [6] | GPCoder, "GPCoder," 29 7 2019. [Online]. Available: https://gpcoder.com/5967-gioi-thieu-swagger-cong-cu-document-cho-restfull-apis/. [Accessed 2 9 2023]. |
| [7] | "Open AI Hub," 22 6 2022. [Online]. Available: https://blog.openapihub.com/en-us/openapi-swagger-automate-your-api-development-process/. [Accessed 2 9 2023]. |
| [8] | "stackjava," 16 3 2018. [Online]. Available: https://stackjava.com/demo/bcrypt-la-gi-code-vi-du-bcrypt-bang-java-jbcrypt.html. [Accessed 2 9 2023]. |
| [9] | Microsoft, "Microsoft," Microsoft, 6 4 2022. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/security/authorization/introduction?view=aspnetcore-7.0. [Accessed 2 9 2023]. |
| [10] | Thảo Meo, "vb.net," 22 4 2022. [Online]. Available: https://laptrinhvb.net/bai-viet/devexpress/---Csharp----Gioi-thieu-thu-vien-RestSharp-dung-goi-API-trong-Dotnet/aaac2b3962ff98e6.html. [Accessed 2 9 2023]. |
| [11] | TheNando, "Mailgun," 21 12 2022. [Online]. Available: https://documentation.mailgun.com/en/latest/quickstart.html. [Accessed 2 9 2023]. |

PHỤ LỤC