 **ĐẠI HỌC HUẾ**

**TRƯỜNG ĐẠI HỌC KHOA HỌC**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**KHÓA LUẬN**

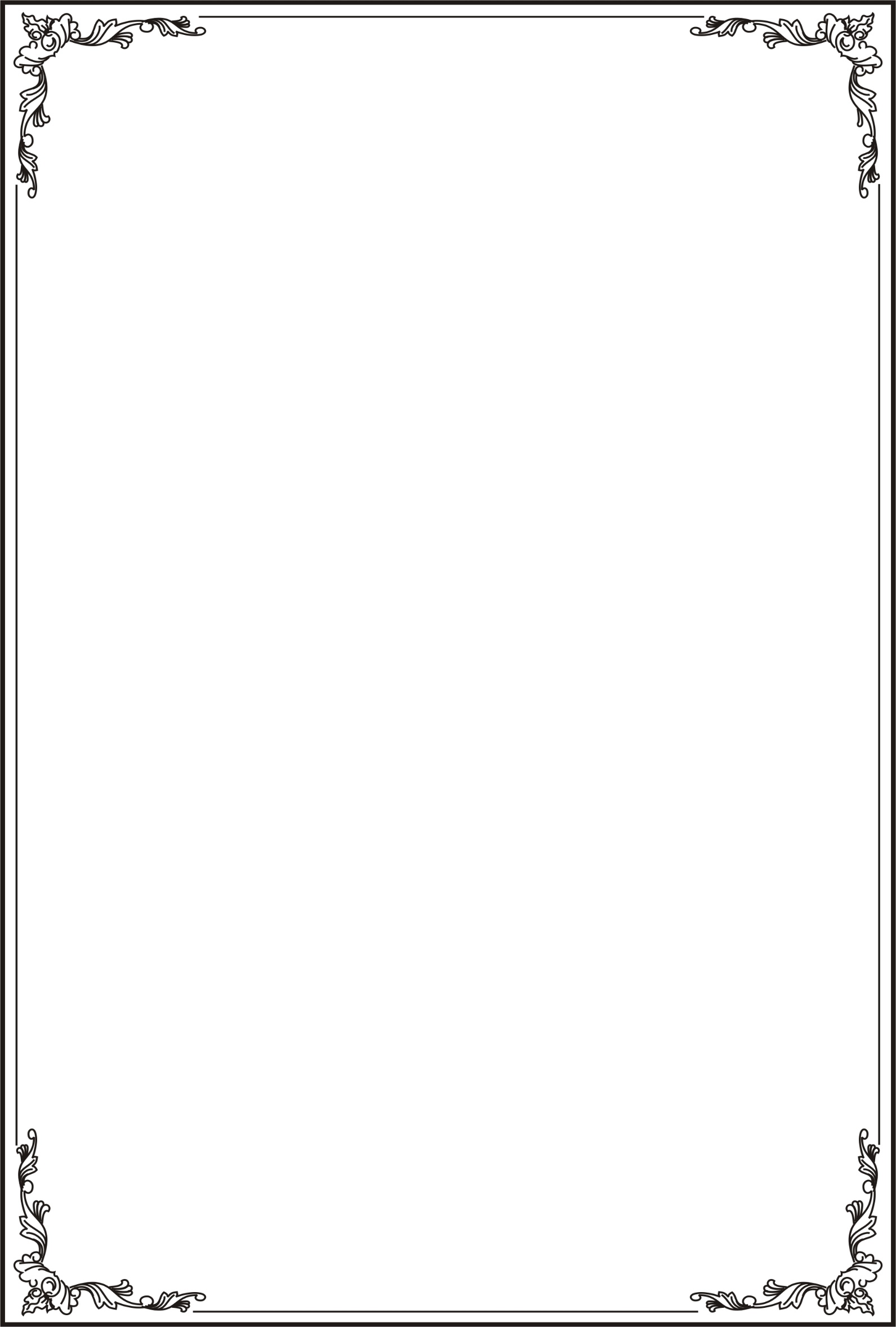
**TỐT NGHIỆP ĐẠI HỌC**

**TÌM HIỂU SPRING FRAMEWORK VÀ MONGODB ĐỂ XÂY DỰNG WEBSITE BÁN BẢO HIỂM TRỰC TUYẾN**

Sinh viên thực hiện: **NGUYỄN ĐĂNG TỈNH**

Lớp: TINK38D – Hệ chính quy

Huế 05/2018

 **ĐẠI HỌC HUẾ**

**TRƯỜNG ĐẠI HỌC KHOA HỌC**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

**KHÓA LUẬN**

**TỐT NGHIỆP ĐẠI HỌC**

**TÌM HIỂU SPRING FRAMEWORK VÀ MONGODB ĐỂ XÂY DỰNG WEBSITE BÁN BẢO HIỂM TRỰC TUYẾN**

Sinh viên thực hiện: **NGUYỄN ĐĂNG TỈNH**

Lớp: TINK38D – Hệ chính quy

Giáo viên hướng dẫn: TS. NGUYỄN HOÀNG HÀ

Huế 05/2018

**MỤC LỤC**

**LỜI CẢM ƠN** …..………………….…...……………………………………….1

**LỜI GIỚI THIỆU** ………………….…...……………………………………….2

**CHƯƠNG 1: TỔNG QUAN** ……………………………………………………4

1.1 GIỚI THIỆU VỀ INTERNET VÀ MỘT SỐ KHÁI NIỆM ..........….……...4

1.1.1 Giới thiệu về Internet ……………………………………............……...4

1.1.2 Web Server ………...……………………………………............……...4

1.1.3 Web client ………….……………………………………............……...4

1.1.4 Trình duyệt Web - Web Browser ……………………………….……...4

1.1.5 HTTP (HyperText Transfer Protocol) .………………………………....5

1.1.6 URL (Uniform Resouce Locator) ……………………………….……...5

1.1.7 HTML (HyperText Markup Language) …………………….......……...5

1.2 TỔNG QUAN VỀ LẬP TRÌNH WEB …….………………………………5

1.2.1 Web tĩnh và lập trình Client - side ……………………………………...5

1.2.1.1 Web tĩnh ……………….…………………………………………...5

1.2.1.2 Đặc điểm khi xử lý tại client ……………….…….………………...6

1.2.2 Giới thiệu về Internet ……………………………………............……...6

1.2.2.1 Web động ……………….……...…………………………………...6

1.2.2.2 Đặc điểm khi xử lý tại server ..……………………………………...6

1.2.2.3 Khi triển khai một ứng dụng Web …..……………………………...7

1.3 NGÔN NGỮ LẬP TRÌNH JAVA …….…………………..…….….……...7

1.3.1 Tính trừu tượng (Abstraction) …..……………………………………...7

1.3.2 Tính đa hình (Polymorphism) …..……………………………………...7

1.3.3 Tính kế thừa (Inheritance) ……....……………………………………...7

1.3.4 Tính đóng gói (Encapsulation) …..……………………………………..8

1.3.5 Một số đặc tính khác ……………..……………………………………..8

**CHƯƠNG 2: TÌM HIỂU SPRING FRAMEWORK VÀ MONGODB** ...……9

2.1 SPRING FRAMEWORK ……………..…………………………….……...9

2.1.1 Sự ra đời của Spring ………..………………….…..………….………..9

2.1.2 EJB và Spring ………………………………….…..………….………10

2.1.3 Spring Core ……………………………………………………………11

2.1.4 Spring Bean …………………………………………………………...11

2.1.5 Dependency Injection (DI) .…………………………………………...12

2.1.6 Spring Context ……………………………...………….……………...12

2.1.7 Spring Expression Language (SpEL) ……………………………...….13

2.1.8 Các dự án trong Spring Framework ……………………………......…13

2.1.8.1 Spring MVC …………………………………..……………...…...13

2.1.8.2 Spring Data …………………………..………..……………..……13

2.1.8.3 Spring Security ………………………………..……………..……13

2.1.8.4 Spring Boot ..…………………………………..……………...…...13

2.1.8.5 Spring Batch …………………………………..……………..……13

2.1.8.6 Spring Intergration …………………………………………...……14

2.1.8.7 Spring XD ……………………………………..……………...…...14

2.1.8.8 Spring Social ……………………………...…..……………...…...14

2.2 MONGODB …………………………..…………………………….…….14

2.2.1 Giới thiệu về hệ cơ sở dữ liệu NoSQL ………..………………………14

2.2.1.1 Khái niệm NoSQL ………………………...…..……………...…...14

2.2.1.2 Ưu nhược điểm của CSDL NoSQL ..……...…..……………...…...15

2.2.2 Hệ cơ sở dữ liệu MongoDB ………….………..………………………16

2.2.2.1 Giới thiệu về MongoDB ...………………...…..……………...…...17

2.2.2.2 Công cụ RoboMongo làm việc với MongoDB .……………...…...18

2.2.2.3 MongoDB CRUD …….....………………...…..……………...…...18

2.2.2.3.1 Giới thiệu MongoDB CRUD ……………...……………...…..18

2.2.2.3.2 Các khái niệm trong MongoDB CRUD ……………...……….19

2.2.2.4 Data Model ……..…….....………………...…..……………...…...21

2.2.2.4.1 Giới thiệu Data Model ……..……………...……………..........21

2.2.2.4.2 Các khái niệm cơ bản về Data Model ……..……………......…21

2.2.2.5 Một số đặc trưng của MongoDB …..……...…..……………...…...24

2.2.2.6 Truy vấn trong MongoDB ………....……...…..……………...…...25

2.2.2.6.1 Các toán tử so sánh ….……..……………...……………......…25

2.2.2.6.2 Ví dụ so sánh giữa SQL và MongoDB .…...……………...…...26

2.2.2.7 Ưu điểm và nhược điểm của MongoDB …..…..……………...…...27

**CHƯƠNG 3: XÂY DỰNG WEBSITE BÁN BẢO HIỂM TRỰC TUYẾN** ..28

3.1 ĐẶC TẢ YÊU CẦU ….…………………………………………….……..28

3.1.1 Mô tả bài toán ……..…...……………………………………….…….28

3.1.1.1 Thông tin bài toán ...…….………....……...…..……………...……28

3.1.1.2 Cơ chế hoạt động ……….………....……...…..……………...……28

3.1.2 Bài toán đặt ra ……….....……………………………………….…….29

3.1.2.1 Xem thông tin bảo hiểm ……..…....……...…..……………...……29

3.1.2.2 Mua bảo hiểm ………………..…....……...…..……………...……30

3.1.2.3 Thanh toán ….………………..…....……...…..……………...……31

3.1.2.4 Các chức năng khác …...……..…....……...…..……………...……31

3.2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ..………………………..…….32

3.2.1 Biểu đồ Use Case ……....………………....…………………….…….32

3.2.2 Biểu đồ lớp …..…….......……………………………………….…….33

3.2.2.1 Tổng quát các lớp ……….………....……...…..……………...…...33

3.2.3 Biểu đồ hoạt động …………...………………………………….…….35

3.2.3.1 Đăng ký tài khoản ……….………....……...…..……………...…...35

3.2.3.2 Tìm lại mật khẩu ..……….………....……...…..……………...…...36

3.2.3.3 Đăng nhập ……………….………....……...…..……………...…...37

3.2.3.4 Cập nhật thông tin cá nhân ...……....……...…..……………...…...38

3.2.3.5 Mua bảo hiểm ……………...……....……...…..……………...…...39

3.2.3.6 Xem danh sách bảo hiểm đã mua .....……...…..……………...…...40

3.2.3.7 Khai báo hồ sơ …………......……....……...…..……………...…...40

3.2.3.8 Xem danh sách hồ sơ khai báo ….....……...…..……………...…...41

3.2.3.9 Thanh toán ………………....……....……...…..……………...…...41

3.2.3.10 Giám định hồ sơ ……....……..……...…..……………...………..42

3.2.4 Biểu đồ tuần tự ……..…..…...………………………………….…….43

3.2.4.1 Đăng ký tài khoản ……….………....……...…..……………...…...43

3.2.4.2 Tìm lại mật khẩu ..……….………....……...…..……………...…...44

3.2.4.3 Đăng nhập ……………….………....……...…..……………...…...45

3.2.4.4 Cập nhật thông tin cá nhân ...……....……...…..……………...…...46

3.2.4.5 Mua bảo hiểm ……………...……....……...…..……………...…...47

3.2.4.6 Xem danh sách bảo hiểm đã mua .....……...…..……………...…...47

3.2.4.7 Khai báo hồ sơ …………......……....……...…..……………...…...48

3.2.4.8 Xem danh sách hồ sơ khai báo ….....………....…..……………….48

3.2.4.9 Thanh toán ………………....……....……...…..……………...…...49

3.2.4.10 Giám định hồ sơ ……....……....……...………………...………..50

3.2.5 Thiết kế cơ sở dữ liệu ………....……………………………….…….51

3.2.5.1 Các đối tượng trong bảng …...…..………………….………...…...51

3.3 CÀI ĐẶT CHƯƠNG TRÌNH ………...………………………..…….……..55

3.3.1 Giao diện phía khách hàng ...……....……...…..……………......…...55

3.3.2 Giao diện phía quản trị ....…………....……...…..……………...…...62

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** ................................………………67

**TÀI LIỆU THAM KHẢO** ……………………………………………………..68

**LỜI CẢM ƠN**

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Em xin gửi lời cảm ơn chân thành đến TS. Nguyễn Hoàng Hà, giảng viên Bộ môn Công nghệ công mềm - Trường Đại học Khoa học Huế người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm khoá luận.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong trường Đại học Khoa học Huế nói chung, các thầy cô trong Bộ môn Công nghệ phần mềm nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành khoá luận tốt nghiệp.

Em xin chân thành cảm ơn!

Huế, 05/2018

Si**n**h viên

Nguyễn Đăng Tỉnh

**LỜI GIỚI THIỆU**

Ngày nay sự phát triển mạnh mẽ của tin học làm cho máy tính trở thành phương tiện không thể thiếu trong mọi lĩnh vự đời sống. Hơn lúc nào hết các tính năng của máy tính được khai thác một cách triệt để. Nếu như vài ba năm trước máy tính ở nước ta máy tính được sử dụng chủ yếu như một công cụ để soạn thảo văn bản thông thường, hoặc các công việc lập trình, quản lý phức tạp, xử lý dự liệu biểu bảng, thương mại, khoa học,… thì giờ đây, cùng với sự vươn xa của mạng Internet trên lãnh thổ Việt Nam máy tính còn là phương tiện bạn có thể ngồi trên bàn làm việc cá nhân tại gia đình mà trao đổi thông tin liên lạc đi khắp toàn cầu.

Nền tin học càng phát triển thì con người càng có nhiều những phương hướng mới, công cụ mới để xử lý thông tin và nắm bắt được nhiều thông tin hơn, tin học được ứng dụng trong mọi ngành nghề, mọi lĩnh vực của đời sống xã hội. Việc áp dụng Tin học vào trong quản lý, sản xuất, kinh doanh du lịch là một xu hướng tất yếu.

Kết quả của việc áp dụng tin học trong quản lý là việc hình thành các hệ thống thông tin quản lý nhằm phục vụ chon nhu cầu xử lý dữ liệu và cung cấp thông tin cho các chủ sở hữu hệ thống đó, ở nước ta hiện nay trong những năm gần đây các ứng dụng tin học vào quản lý đang ngày một nhiều hơn, đa dạng hơn.

Hệ thống thông tin được đề cập đến trong đồ án này là hệ thống hỗ trợ hoạt động quản lý công ty bán bảo hiểm (Bảo hiểm Bình An). Nếu việc quản lý công ty bán bảo hiểm đều làm bằng tay thì đó là một công việc rất khó khan và phức tạp, không hiệu quả, tốn nhiều thời gian vào việc phân tích và xử lý dữ liệu mà kết quả đưa ra lại có độ chính xác không cao trong quá trình tính toán và thống kê. Nhất là số người tham gia của mỗi đơn vị là rất đông và hàng tháng nhân viên các bộ phận tốn rất nhiều thời gian đề cập nhật và thống kê…

Đề giải quyết công việc khó khan này một cách nhanh chóng và thuận lợi thì tin học đã cung cấp cho ta các phương pháp và công cụ để xây dựng chương trình ứng dụng gần như tự động hóa được các công việc của nhân viên.

**CHƯƠNG 1**

**TỔNG QUAN**

* 1. **GIỚI THIỆU VỀ INTERNET VÀ MỘT SỐ KHÁI NIỆM**

**1.1.1 Giới thiệu về Internet**

**-** Là tập hợp các máy tính nối mạng trên thế giới nhằm trao đổi thông tin với nhau thông qua giao thức TCP/IP (Transmission Control Protocol/Internet Protocol)

- Mang lại rất nhiều tiện ích hữu dụng cho người sử dụng, nổi bật nhất là dịch vụ thư điện tử (email), trò chuyện trực tuyến (chat) và truy tìm dữ liệu (search engine)

- Dịch vụ được dùng rất phổ biến trên Internet là World Wide Web (WWW, thường được gọi tắt là Web)

**1.1.2 Web Server**

**-** Máy tính lưu trữ Website

- Trả kết quả về cho Web Client khi nhận được yêu cầu

**1.1.3 Web Client**

**-** Máy tính dùng để truy cập các trang web

- Có khả năng yêu cầu và nhận kết quả từ Web Server

**1.1.4 Trình duyệt Web – Web Browser**

**-** Phần mềm dùng để xem các tài liệu hoặc tìm kiếm các tài nguyên trên World Wide Web

- Một số trình duyệt thông dụng hiện nay:

* Chrome của Google
* Internet Explorer (IE) có sẵn trong Microsoft Windows của Microsoft
* Mozilla Firefox của Tập đoàn Mozilla
* Netscape Navigator của Netscape
* Opera của Opera Software

**1.1.5 HTTP (HyperText Transfer Protocol)**

**-** Là giao thức chuyển giao siêu văn bản trên web

- Giao thức này là tập hợp các qui định dùng để trao đổi các tài liệu (văn bản, hình ảnh, âm thanh, video, các tập tin đa truyền thông,…) giữa Web server và trình duyệt Web

**1.1.6 URL (Uniform Resource Locator)**

**-** Đường dẫn chỉ tới một trang web cụ thể trên Internet.

- Cú pháp đầy đủ: scheme://<host> [:port] [<path>[ ?<querystring> ] ]

* scheme: lọai dịch vụ Internet
* host: địa chỉ máy chủ chứa tài nguyên
* port: cổng dịch vụ trên máy chủ
* path: đường dẫn và tên của tập tin tài nguyên trên máy chủ
* querystring: các tham số được gửi kèm theo http

**1.1.7 HTML (HyperText Markup Language)**

**-** Ngôn ngữ dùng để xây dựng các trang web

- Gồm các tag giúp web browser biết cách định dạng thông tin hiển thị

* 1. **TỔNG QUAN VỀ LẬP TRÌNH WEB**

**1.2.1 Web tĩnh và lập trình Client - side**

**1.2.1.1 Web tĩnh**

**-** Là trang web được trình bày dưới dạng văn bản. Nội dung trong trang được tạo ra lúc thiết kế và không được thay đổi khi có người dùng truy cập

**-** Được viết bằng ngôn ngữ HTML và không có kết nối với cơ sở dữ liệu (CSDL)

- Khi có nhu cầu xử lý trên trang web tĩnh, ta dùng ngôn ngữ Client-side script => Lập trình Client-side

- Quá trình xử lý chỉ xảy ra tại Web client

**1.2.1.2 Đặc điểm khi xử lý tại client**

**-** Nhanh, không mất thời gian để thể hiện kết quả lên trang web

- Chỉ có thể thực hiện những xử lý đơn giản và độc lập

- Không cần phải cài đặt phần mềm tại web server

- Trình duyệt phải hỗ trợ ngôn ngữ script mà trang HTML đang sử dụng

- Rất hữu ích cho DHTM

**1.2.2 Web tĩnh và lập trình Server - side**

**1.2.2.1 Web động**

**-** Là trang web có nội dung được lấy từ CSDL

- Dùng ngôn ngữ Server-side script để xử lý

- Quá trình xử lý xảy ra tại Web server => Lập trình Server-side

**1.2.2.2 Đặc điểm khi xử lý tại server**

**-** Chậm, mất thời gian để thể hiện kết quả lên trang web

- Có khả năng thực hiện những xử lý phức tạp và truy cập cơ sở dữ liệu

- Trang web linh động, phục vụ đa yêu cầu của người dung

- Phải cài đặt phần mềm tại Web server

**1.2.2.3 Khi triển khai một ứng dụng Web**

**-** Cần nhận biết bản chất của mỗi xử lý là nên thực hiện ở Client hay Server

- Một website có thể chứa các trang web tĩnh và động

* 1. **NGÔN NGỮ LẬP TRÌNH JAVA**

**-** Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995 (Java 1.0 [J2SE]). Tính đến thời điểm này (tháng 2/2015) phiên bản mới nhất của Java Standard Edition (JSE) là 8. Với ưu thế về đa nền tảng (multi platform) Java càng lúc càng được ứng dụng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác...

**-** Java là ngôn ngữ lập trình hướng đối tượng nên nó cũng có 4 đặc điểm chung của các ngôn ngữ hướng đối tượng

**1.3.1 Tính trừu tượng (Abstraction)**

**-** Là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.

**1.3.2 Tính đa hình (Polymorphism)**

**-** Cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.

**1.3.3 Tính kế thừa (Inheritance)**

**-** Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

**1.3.4 Tính đóng gói (Encpsulation)**

**-** Là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

**1.3.5 Một số đặc tính khác**

* **Độc lập nền (Write Once, Run Anywhere)**: Không giống như nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã byte code chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.
* **Đơn giản**: Học Java thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++.
* **Bảo mật**: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)...
* **Đa luồng**: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được xử dụng rất nhiều trong lập trình game.
* **Hiệu suất cao: N**hờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.
* **Linh hoạt**: Java được xem là linh hoạt hơn C/C ++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

**CHƯƠNG 2**

**TÌM HIỂU SPRING FRAMEWORK VÀ MONGODB**

* 1. **SPRING FRAMEWROK**

**2.1.1 Sự ra đời của Spring**

**-** Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề “J2EE Development without EJB”. Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB.

- Ngày nay, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

- Để ngăn chặn sự phức tạp trong phát triển các ứng dụng, Spring Framework thường dựa trên các quan điểm như sau:

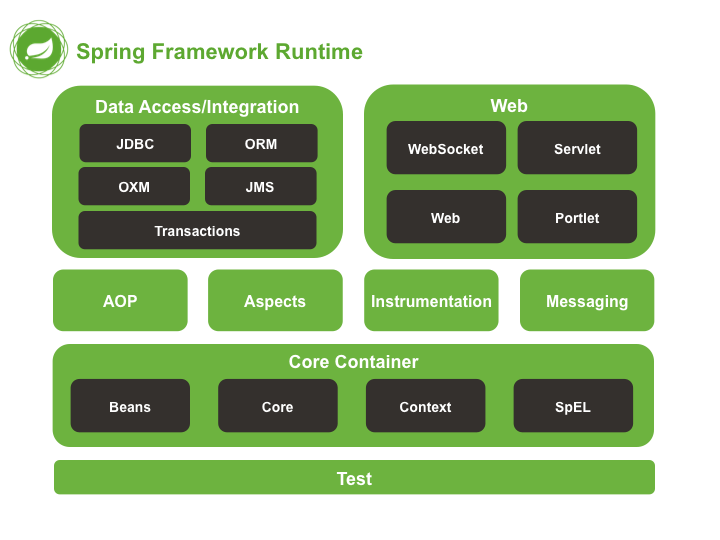
* Đơn giản hóa công việc phát triển thông qua việc sử dụng các đối tượng Java đơn giản hay còn được gọi là POJO (Plain Old Java Object)
* Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection và viết các interface
* Tiếp cận lập trình khai báo bằng cách sử dụng các quy tắc (convention) và các khía cạnh (aspect) chung
* Giảm thiểu các mã nghi thức và soạn sẵn (boilerplate) thông qua việc sử dụng các khuôn mẫu (template) và các khía cạnh

**2.1.2 EJB và Spring**

**-** Có thể nói rằng Spring Framework ra đời là nhờ sự thành công của **EJB**. EJB (Enterprise JavaBeans) là một kiến trúc phát triển dành cho các ứng dụng doanh nghiệp đòi hỏi tính mạnh mẽ và khả năng mở rộng cao. EJB là một thành phần quan trọng của nền tảng Java EE. EJB được tạo ra bởi hai gã khổng lồ công nghệ là IBM và Sun vào những năm cuối thập niên 90. Dưới sự phát triển của hai tập đoàn này, EJB đã trở nên nặng nề hơn. Tiêu chuẩn EJB cũng nhanh chóng được thông qua. Sau tất cả, EJB được ca ngợi là giải pháp tốt nhất để xây dựng các ứng dụng phân tán dành cho doanh nghiệp.

- Nhưng rồi những chỉ trích về chi tiết kỹ thuật của EJB đã tăng nhanh còn hơn sự nổi tiếng của nó. EJB là một framework lớn với độ phức tạp cao. Các nhà phát triền khi làm việc với EJB đã nhận ra rằng nó không hoạt động như những gì nó đã đề ra. EJB chỉ tuyệt vời trên bàn giấy, nhưng ra thực tế đó lại là một chuyện khác.

- EJB vẫn được tiếp tục phát triển, dễ sử dụng hơn nhiều so với buổi ban đầu. Nhưng những chỉ trích vẫn còn đó, nỗi thất vọng trong cộng đồng Java ngày càng tăng lên, đòi hỏi cần có một giải pháp thay thế.



*Hình 2.1* Spring Framework Runtime.

**2.1.3 Spring Core**

**-** Spring Core chính là thành phần trung tâm, cốt lõi của Spring Framework. Đây chính là nền tảng để xây dựng nên các thành phần khác.

- Lược đồ dưới đây mô tả mối quan hệ giữa Spring Core với các thành phần khác trong Spring Framework:

**2.1.4 Spring Bean**

**-** Spring Bean là trung tâm của Spring Core và là trái tim của một ứng dụng Spring. Trái ngược với EJB, Spring Framework được thiết kế từ lõi bằng cách sử dụng các POJO hay các Spring Bean. Spring Bean có thể được hiểu là các đối tượng Java đơn giản. Điều này khá tương đồng với nguyên lý thiết kế "Đơn trách nhiệm" (single responsibility) của Robert C Martin trong lập trình hướng đối tượng.

**2.1.5 Dependency Injection (DI)**

**-** Dependency Injection (có thể dịch tiêm các thành phần phụ thuộc) là một sức mạnh nổi bật của Spring Framework. Trong các phiên bản gốc của EJB người ta không thấy sự xuất hiện của Dependency Injection, nhưng có lẽ do nhận thấy sự thành công từ Spring khi sử dụng Dependency Injection, các phiên bản EJB 3.x đã cung cấp các annotation để thực hiện inject các dependency. Dependency Injection là một mẫu thiết kế phần mềm mà các đối tượng phụ thuộc sẽ được inject vào một lớp nào đó. Dependency Injection là một implementation cụ thể của khái niệm Inversion of Control (đảo ngược điều khiển).

- Ta sẽ có một ví dụ đơn giản về Dependency Injection như sau: Bạn có một web controller có nhiệm vụ lưu thông tin gửi từ form người dùng. Theo nguyên lý Đơn trách nhiệm, bạn không muốn lớp controller tương tác với cơ sở dữ liệu. Thay vào đó, bạn sẽ sử dụng một lớp service để làm công việc này. Như vậy, controller của bạn sẽ chỉ phải xử lý dữ liệu của form (get form data, validate data, ...) rồi gọi một phưong thức của lớp service được inject để lưu dữ liệu. Controller không cần phải quan tâm hay lo lắng về kết nối cơ sở dữ liệu, pooling hay bảng nào sẽ được update. Cũng như service không cần phải biết request có những thông tin gì.

**2.1.6 Spring Context**

**-** Spring Context mang mọi thứ lại với nhau. Spring Context kế thừa các tính năng của Spring Bean và bổ sung các hỗ trợ cho internationalization (ví dụ như các resource bundle), event propagation, resource loading... Ngoài ra, Spring Context cũng hỗ trợ các tính năng của Java EE như EJB, JMX và truy cập từ xa cơ bản. Interface **ApplicationContext** là tiêu điểm của Spring Context.

**2.1.7 Spring Expression Language (SpEL)**

**-** Spring Expression Language là một ngôn ngữ ngắn gọn giúp cho việc cấu hình Spring Framework trở nên linh hoạt hơn.

**2.1.8 Các dự án trong Spring Framework**

**-** Trên thực tế Spring Framework là một tập hợp của nhiều dự án con. Spring Core như chúng ta đã tìm hiểu là nền tảng của các dự án trong Spring Framework. Một một dự án sẽ đảm nhận một chức năng riêng trong việc xây dựng các ứng dụng doanh nghiệp. Chúng ta sẽ cùng điểm qua các dự án lớn trong Spring.

**2.1.8.1 Spring MVC**

**-** Spring MVC được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web. Đây là một dự án chúng ta không thể bỏ qua khi xây dựng các ứng dụng Java web.

**2.1.8.2 Spring Data**

**-** Cung cấp một cách tiếp cận đúng đắn để truy cập dữ liệu từ cơ sở dữ liệu quan hệ, phi quan hệ, map-reduce và thậm chí còn hơn thế nữa.

**2.1.8.3 Spring Security**

**-** Cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng của bạn.

**2.1.8.4 Spring Boot**

**-** Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng.

**2.1.8.5 Spring Batch**

**-** Giúp chúng ta dễ dàng tạo các lịch trình (scheduling) và tiến trình (processing) cho các công việc xử lý theo mẻ (batch job).

**2.1.8.6 Spring Intergration**

**-** Spring Integration là một implementation của Enterprise Integration Patterns (EIP). Dự án này thiết kế một kiến trúc hướng thông điệp hỗ trợ việc tích hợp các hệ thống bên ngoài.

**2.1.8.7 Spring XD**

**-** Mục tiêu của dự án này là đơn giản hóa công việc phát triển các ứng dụng Big Data.

**2.1.8.8 Spring Social**

**-** Dự án này sẽ kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, Linkedin...

* 1. **MONGODB**

2.2.1 Giới thiệu về hệ cơ sở dữ liệu NoSQL

2.2.1.1 Khái niệm NoSQL

- Với hầu hết các thời kỳ web, Hệ quản trị cơ sở dữ liệu quan hệ dựa trên SQL đã thống trị hầu hết các hệ Quản trị Cơ sở dữ liệu. Tuy nhiên, thời gian gần đây, một cách tiếp cận mới đã bắt đầu biết đến là NoSQL, tạo ra sự thay thế cho các hệ quản trị cơ sở dữ liệu truyền thống.

- NoSQL còn có nghĩa là Non-Relational - không ràng buộc. Tuy nhiên, thuật ngữ đó ít phổ dụng hơn và ngày nay người ta thường dịch NoSQL thành Not Only SQL - Không chỉ SQLS. NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm.

- Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các cơ sở dữ liệu quan hệ nguồn mở nhỏ nhưng không sử dụng SQL cho truy vấn.

- Hệ CSDL này có thể lưu trữ, xử lý từ lượng rất nhỏ đến hàng petabytes dữ liệu với khả năng chịu tải, chịu lỗi cao nhưng chỉ đòi hỏi về tài nguyên phần cứng thấp.

- NoSQL thích hợp cho các ứng dụng cực lớn (dịch vụ tìm kiếm, mạng xã hội…) và các ứng dụng nhỏ.

|  |  |  |
| --- | --- | --- |
| **Tính năng** | **CSDL quan hệ** | **NoSQL** |
| Hiệu suất | Kém hơn  Truy vấn SQL  Có quan hệ giữa các bảng | Cực cao  Bỏ qua truy vấn SQL  Bỏ qua ràng buộc quan hệ |
| Hiệu suất đọc-ghi | Kém do thiết kế để đảm bảo vào/ra liên tục của dữ liệu. | Tối ưu về tốc độ đọc ghi dữ liệu. |
| Phần cứng | Đòi hỏi về phần cứng cao. | Đòi hỏi phần cứng thấp hơn. |

*Bảng 2.1* So sánh giữa SQL và NoSQL.

* + - 1. Ưu nhược điểm của CSDL NoSQL
* Ưu điểm của NoSQL:

- Là mã nguồn mở: không phải mất chi phí và có xu hướng tin cậy, an ninh và nhanh hơn để triển khai so với các hệ quản trị cơ sở dữ liệu độc quyền.

- Linh hoạt trong việc mở rộng và phát triển: về phía công ty quản lý thì giúp dễ dàng mở rộng máy chủ khi dữ liệu ngày càng lớn hoặc lượng truy cập, tải dữ liệu quá lớn, thay vì thuê một máy chủ lớn hơn để thế máy chủ trước thì công ty chỉ cẩn thuê thêm một máy chủ khác. Về phía người lập trình thì giúp dễ dàng thêm Collection(Bảng) hoặc cặp dữ liệu (Cột). Dễ dàng trong việc thống kê, truy vấn nhanh.

- Áp dụng được công nghệ điện toán đám mây: dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu cảu các ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các kết nối mạnh. NoSQL lưu trữ dữ liệu thường mở rộng theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn.

- Được các hãng lớn như Amazon, Facebook và Google sử dụng. Làm nên tảng, cơ sở cho các công ty lớn áp dụng cũng như có sự tin cậy cao về hệ quản trị NoSQL.

* Nhược điểm của NoSQL:

- Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhấtt từ nhà cung cấp RMBMS(vì được phát triển trước NoSQL một thời gian khó dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được hỗ trợ tốt nhất.

- Chưa được sử dụng rộng rãi vì nó vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu được xây dựng từ lâu, nên việc chuyển đổi cũng là vấn đề khó với nhiều doanh nghiệp và chưa được hỗ trợ tốt về chức năng cũng như sự ổn định.

- Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có tool giao diện tương tác cũng như các phương thức tốt nhất. Dẫn đến hạn chế về tri thức nghiệp vụ.

- Chia sẽ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng riêng của mình. Sự thiếu hụt các tiêu chuẩn có nghĩa là nó không có khả năng để chuyển một cách đơn giản từ một nhà cung cấp này sang nhà cung cấp khác.

2.2.2 Hệ cơ sở dữ liệu MongoDB

- Trong những gương mặt góp phần làm suy tàn SQL thì MongoDB nổi lên là một CSDL đáng tin cậy và dễ dùng nhất. Mongo viết bằng C++. Nó thích hợp cho các ứng dụng tầm trung trở lên. Nếu tỉ lệ lượng dữ liệu ghi vào CSDL của ứng dụng lớn hơn lượng đọc thì đây càng là lựa chọn hợp lý.

- MongoDB là một CSDL có khả năng mở rộng, hiệu suất cao, mã nguồn mở và hướng văn bản.

2.2.2.1 Giới thiệu về MongoDB

- MongoDB là một mã nguồn mở, dùng để thiết kế cơ sở dữ liệu để dễ dàng phát triển và mở rộng, cung cấp hiệu suất cao, tính sẵn sàng cao

- Cấu trúc của Mongodb bao gồm các cặp “thuộc tính” – “giá trị”, MongoDb document tương tự như đối tượng JSOn. Giá trị của thuộc tính bao gồm các thuộc tính giá trị cụ thể, document, mảng, mảng các đối tượng.

Các tính năng chính:

* Hiệu suất cao: hỗ trợ nhúng dữ liệu dạng mô hình dữ liệu giúp giảm thiểu hoạt động của server, truy vấn dữ liệu sử dụng chỉ mục giúp tối ưu tốc độ truy vấn.
* Dễ dàng tăng tính mở rộng: chế độ sharding tự động ( tự động phân cùng dữ liệu trên máy chủ, đồng bộ hóa dữ liệu tốt.

- MongoDb chạy trên hầu hết các nền tảng và hỗ trợ kiến trúc 64-bit để sử dụng sản xuất và cả hai 64-bit và 32-bit để thử nghiệm.

|  |  |  |
| --- | --- | --- |
|  | **MongoDB** | **MongoDB Enterprise** |
| Amazon Linux | Hỗ trợ | Hỗ trợ |
| Debian 7.1 | Hỗ trợ | Hỗ trợ |
| RedHat / CentOS 6.2+ | Hỗ trợ | Hỗ trợ |
| SUSE 11 | Hỗ trợ | Hỗ trợ |
| Ubuntu LTS12.04 | Hỗ trợ | Hỗ trợ |
| Ubuntu LTS14.04 | Hỗ trợ | Hỗ trợ |
| Windows Server 2012 | Hỗ trợ | Hỗ trợ |
| Mac OSX 10.6+ | Hỗ trợ |  |
| RedHat / CentOS 5.5+ | Hỗ trợ |  |
| RedHat / CentOS 5.7+ | Hỗ trợ | Hỗ trợ |
| Windows Server 2008 R2 | Hỗ trợ | Hỗ trợ |

Bảng 2.2. Các hệ điều hành hỗ trợ.

2.2.2.2 Công cụ RoboMongo làm việc với MongoDB

- RoboMongo là một công cụ trực quan giúp bạn quản lí Database MongoDB, mó là một phần mềm mã nguồn mở miễn phí, hỗ trợ cả 3 loại hệ điều hành: Linux, Windows, Mac OS.

2.2.2.3 MongoDB CRUD

2.2.2.3.1 Giới thiệu MongoDB CRUD

- CRUD là viết tắt của Creat( Tạo), Retrieve(Lấy), Update(Cập nhật), và Delete(Xóa). MongoDB lưu trữ dữ liệu trong các document, với các cặp thuôc tính: giá trị như Json.

{

“MaNhanVien”:NV2, Thuộc tính:Giá trị

“Quyen”:Nhanvien, Thuộc tính:Giá trị

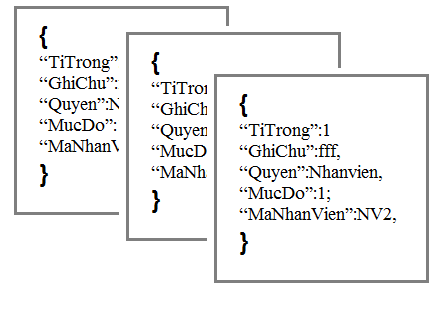
“MucDo”:1; Thuộc tính:Giá trị

“GhiChu”:fff, Thuộc tính:Giá trị

“TiTrong”:1 Thuộc tính:Giá trị

}

MongoDB lưu trữ tất cả các document trong Collection. Collection là một nhóm các doucument giống nhau về chỉ số các thành phần.



* + - * 1. Các khái niệm trong MongoDB CRUD

- Database là gì ?

* Database là một ô chứa dữ liệu ở mức vật lí, mỗi database sẽ có nhiều Collection và được thiết lập để lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDb thường có thể tạo nhiều cơ sở dữ liệu.

- Collection (Bộ sưu tập) là gì?

* Collection trong MongoDb là nhóm các tài liệu (document) nó tương đương với một bảng(table) trong cơ sở dữ liệu thông thường nên mỗi collection sẽ chỉ thuộc về một database duy nhất. Tuy nhiên nó có sự khác biệt đó là nó không có sự ràng buộc Relationship như các hệ quản trị khác nên việc truy xuất dữ liệu rất nhanh, chính vì thế mỗi collection có thể chức nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định.

- Document (Văn bản) là gì?

* Document trong MongoDb có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp “key-value” nên có tính năng động rất lớn. Ta có thể hiểu nó như các record dữ liệu trong MySQL, tuy nhiên nó có sự khác biệt là các cặp “key-value” có không thể giống nhau ở mỗi document.
* MongoDB có thể lưu trữ nhiều CSDL độc lập, mỗi CSDL này có các bộ sưu tập và điều khoản riêng của mình
* MongoDB đi kèm với một trình tiện ích JavaScript đơn giản nhưng mạnh mẽ, nó hữu ích trong quản trị và thao tác dữ liệu.
* Mỗi văn bản có một khóa đặc biệt, đó là “\_id”, nó là duy nhất trong bộ sưu tập của văn bản.

| **SQL** | **MongoDB** |
| --- | --- |
| Table (Bảng) | Collection ( Bộ sưu tập) |
| Row (Hàng) | Document (Văn bản) |
| Column (Cột) | Field (Một cặp key - value) |
| Primary key(Khóa chính) | Primary key (Mặc định là \_id do chính Mongo tạo) |

Bảng 2.3. So sánh giữ SQL và MongoDB.

2.2.2.4 Data Model

2.2.2.4.1 Giới thiệu Data Model

- Hệ quản trị cơ sở dữ liệu MongoDB là một mô hình dữ liệu linh hoạt. Không giống như các cơ sở dữ liệu SQL, phải xác định và triển khai sơ đồ của một bảng trước khi chèn dữ liệu, Collection trong MongoDB rất linh hoạt không nhất thiết phải cho biết những thuộc tính có trong Collection khi nào cần bổ sung sau vẫn được. Sự linh hoạt này tạo điều kiện cho việc hướng tới một cách lưu trữ như một thực thể hay một đối tượng. Mỗi document có phù hợp với các trường dữ liệu của Collection chứa nó, ngay cả khi thay đổi gâng như toàn bộ dữ liệu. Tuy nhiên, trong thực tế, các Document trong một Collection chung một cấu trúc.

- Khi thiết kế cơ sở dữ liệu cần chú ý về cân bằng nhu cầu của các ứng dụng (tức là có nên tạo nhiều Collection lồng nhau), các đặc tính hiệu suất khi thực hiện truy vấn, các mô hình dữ liệu. Khi thiết kế các mô hình dữ liệu, luôn luôn xem xét việc sử dụng các dữ liệu (ví dụ như truy vấn,cập nhật, và xử lý các dữ liệu) cũng như các cấu trúc vốn có của bản thân dữ liệu

2.2.2.4.2 Các khái niệm cơ bản về Data Model

a) Thiết kế Data Model

- Data Model nhúng

* Với MongoDB, bạn có thể nhúng các dữ liệu liên quan vào một Structure hoặc Document, hay còn gọi là mô hình “demormalized”.

Ví dụ:

Tài khoản

{

“MaNhanVien”:NV1,

“TenNhanVien”:aa,

“Quyen”:Q1,

“TrangThai”:nghi phep,

“TaiKhoan”:

{

“TenTaiKhoan”: tk1,

“MatKhau”:\*\*\*,

“MatKhauCap1”: 123

},

“GhiChu” : aaaa,

“SoLuongBanBe”: 5

}

* Mô hình dữ liệu nhúng cho phép các ứng dụng lưu trữ phần dữ liệu liên quan của thông tin trong Document của các cơ sở dữ liệu.
* Sử dụng nhúng mô hình dữ liệu khi: Cơ sở dữ liệu có mối quan hệ giữa các thực thể.
* Nói chung, nhúng cung cấp hiệu suất tốt hơn cho hoạt động đọc, cũng như khả năng yêu cầu và lấy dữ liệu có liên quan trong một hoạt đọng cơ sơ dữ liệu duy nhất.

- Chuẩn hóa Data Model

* Tạo liên kết giữa các “id” của các Document với nhau

Ví dụ:

Quyền

{

“id”:<ObjectId>,

“MoTa”: aaa,

“MucDo”: 1,

“LoaiQuyen”: Nhanvien

}

Tài Khoản

{ “id”: NV1,

“TenNhanVien”: nguyen,

“Quyen\_id”: <ObjectId>,

“TrangThai”: nghi phep,

“TaiKhoan”:

{

“TenTaiKhoan”: tk1,

“MatKhau”:\*\*\*,

“MatKhauCap1”: 123

},

“GhiChu” : aaaa,

“SoLuongBanBe”: 5

}

b) Operational Factors and Data Models (Các hoạt động và mô hình dữ liệu)

**-** Mô hình hóa dữ liệu ứngd ụng cho MongoDB phụ thuộc vào dữ liệu, cũng như các đặc điểm của MongoDB.

* Document Growth
* Indexes

2.2.2.5 Một số đặc trưng của MongoDB

|  |  |
| --- | --- |
| **Thuộc tính** | **Đặc trưng** |
| Lưu trữ hướng văn bản | Văn bản theo phong cách JSON với những lược đồ động đơn giản |
| Hỗ trợ chỉ mục đầy đủ | Chỉ mục trên bất kỳ các thuộc tính |
| Tính sao lặp và tính sẵn sàng cao | Mở rộng |
| Auto-sharding | Mở rộng theo chiều ngang mà không ảnh hưởng đến chức năng |
| Truy vấn | Đa dạng, truy vấn dựa trên văn bản |
| GirdFS | Lưu trữ với bất kỳ kích cỡ nào mà không làm phức tạp ngăn xếp |
| Hỗ trợ thương mại | Hỗ trợ doanh nghiệp, đào tạo, tư vấn |

Bảng 2.4.Thuộc tính và đặc trưng của MongoDB.

Các lưu ý khi chọn MongoDB:

- Nếu ứng dụng có tính chất Insert cao, bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.

- Ứng dụng ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dụng. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì sẽ bỏ qua phần đó nên sẽ an toàn.

- Ứng dụng bạn có nhiều dữ liệu. Bởi vì MongoDB có khả năng tìm kiếm thông tin liên quan cũng khá nhanh.

- Máy chủ không có hệ quản trị CSDL, trường hợp này thường bạn sẽ sử dụng SQL LIFE hoặc MongoDB.

2.2.2.6 Truy vấn trong MongoDB

- Một trong những tính năng tốt nhất của MongoDB là hỗ trợ truy vấn động. Hệ thống hỗ trợ truy vấn động không yêu cầu bất cứ chỉ mục nào để tìm dữ liệu. Người dùng có thể tìm dữ liệu với việc sử dụng bất kỳ tiêu chuẩn nào. Với CSDL quan hệ, truy vấn động là chuẩn hóa.

- MongoDB hỗ trợ một số các đối tượng truy vấn để lấy dữ liệu. Ví dụ, giả sử chúng ta muốn sử dụng trình MongoDB để trả về mọi văn bản trong bộ sưu tập users. Truy vấn sẽ được viết như sau:

db.users.find({})

- Trong trường hợp này, điều kiện của chúng ta là trống, nó phù hợp với mọi văn bản trong bộ sưu tập.

Ví dụ: db.users.find({'ten’: a})

Ở đây, lựa chọn của chúng ta là tất cả các văn bản mà thuộc tính *ten* là *a.*

2.2.2.6.1 Các toán tử so sánh

- Ta có bảng các toán tử sau:

|  |  |
| --- | --- |
| $gt | ứng với các giá trị lớn hơn giá trị điều kiện trong truy vấn |
| $gte | ứng với các giá trị lớn hơn hoặc bằng giá trị điều kiện trong truy vấn |
| $in | ứng với bất kỳ giá trị nào nằm trong một tập cho trước |
| $lt | ứng với các giá trị nhỏ hơn giá trị điều kiện trong truy vấn |
| $lte | ứng với các giá trị nhỏ hơn hoặc bằng giá trị điều kiện trong truy vấn |
| $ne | ứng với tất cả các giá trị khác với giá trị điều kiện trong truy vấn |
| $nin | ứng với bất kỳ giá trị nào không nằm trong một tập hợp cho trước |

2.2.2.6.2 Ví dụ so sánh giữa SQL và MongoDB

- Để nắm rõ hơn các lệnh khi chuyển từ cơ sở dữ liệu SQL sang MongoDB, cần phải tham khảo các lệnh sau đây:

|  |  |
| --- | --- |
| **SQL** | **MONGODB** |
| INSERT INTO ten VALUES(3,5) | db.ten.insert({a:3,b:5}) |
| SELECT a,b FROM ten | db.ten.find({},{a:1,b:1}) |
| SELECT \* FROM ten | db.ten.find() |
| SELECT \* FROM ten WHERE tuoi=22 | db.ten.find({tuoi:22}) |
| SELECT a,b FROM ten WHERE tuoi=22 | db.ten.find({tuoi:22},{a:1,b:1}) |
| SELECT \* FROM ten WHERE tuoi=22 ORDER BY ho | db.ten.find(({tuoi:22}),sort({ho:1}) |
| SELECT \* FROM ten WHERE tuoi>22 | db.ten.find({tuoi:{$gt:22}}) |
| SELECT \* FROM ten WHERE tuoi!=22 | db.ten.find({tuoi:{$ne:22}}) |
| SELECT \* FROM ten WHERE ho LIKE “%nguyen%” | db.ten.find({ho:/nguyen/}) |
| SELECT \* FROM ten WHERE ho LIKE “nguyen%” | db.ten.find({ho:/^nguyen/}) |
| SELECT \* FROM ten WHERE tuoi>22 AND tuoi<=40 | db.ten.find({‘tuoi’:{$gt:22,$lte:40}}) |
| SELECT \* FROM ten WHERE a=1 and b=’q’ | db.ten.find({a:1,b:’q’}) |
| SELECT \* FROM ten WHERE a=1 or b=2 | db.ten.find({$or :[{a:1},{b:2}]}) |
| SELECT COUNT(\*y) FROM ten | db.ten.count() |
| SELECT COUNT(\*y) FROM ten where tuoi>22 | db.ten.find({tuoi:{‘$gt’:22}}),count() |
| UPDATE ten SET a=1 WHERE b=’q’ | db.ten.update({b:’q’},{$set:{a:1}},false,true) |
| DELETE FROM ten WHERE z=”abc” | db.ten.remove({z:’abc’}) |

* + - 1. Ưu điểm và nhược điểm của MongoDB
* Ưu điểm

- Khả năng lưu trữ không giới hạn tùy thuộc vào ổ cứng nhưng vẫn đảm bảo hiệu suất cao.

- Khả năng mở rộng theo chiều ngang.

- Lưu trữ theo key - value giúp đẩy nhanh tốc độc truy xuất dữ liệu.

- Tính đáp ứng cao.

* Nhược điểm:

- Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao

- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

**CHƯƠNG 3**

**XÂY DỰNG WEBSITE BÁN BẢO HIỂM TRỰC TUYẾN**

* 1. **ĐẶC TẢ YÊU CẦU**

**3.1.1 Mô tả bài toán**

**3.1.1.1 Thông tin bài toán**

**-** Đây là kênh bán hàng mới trong chiến lược đa dạng hóa sản phẩm bảo hiểm, kênh phân phối nhằm hướng đến đối tượng khách hàng cá nhân của Bình An tiếp theo sự thành công của kênh bán hàng qua máy rút tiền tự động và đặc biệt là sự thành công của kênh bán hàng liên kết qua ngân hàng.  
 - Ngoài chức năng mua bán bảo hiểm trực tuyến, Website còn là địa chỉ để khách hàng tra cứu nhanh các thông tin cần thiết khi mua bảo hiểm như: thông tin chi tiết về điều khoản điều kiện bảo hiểm, mức phí, thông tin về bồi thường, câu hỏi và trả lời về các thắc mắc khách hàng thường gặp phải khi mua bảo hiểm.

**-** Với website bán hàng trực tuyến của Bình An, khách hàng có thể lựa chọn các sản phẩm bảo hiểm phù hợp nhu cầu và thực hiện thanh toán trực tuyến thông qua nhiều hình thức như thẻ thanh toán quốc tế Visa, Master, thẻ ngân hàng nội địa, ví điện tử...

**3.1.1.2 Cơ chế hoạt động**

**Hoạt động chính của Website như sau:**

- Khách hàng có thể truy cập vào Website để xem thông tin chi tiết về điều khoản điều kiện bảo hiểm, mức phí, thông tin về bồi thường.

- Để mua loại bảo hiểm nào đó khách hàng phải đăng ký một tài khoản cá nhân. Sau khi đăng ký thành công khách hàng có thể dùng tài khoản đó để đăng nhập vào hệ thống và thực hiện các chức năng au (Hệ thống có thiết lập thời gian time out cho tài khoản đăng nhập nhằm bảo mật thông tin cho khách hàng tốt hơn):

* Khách hàng có thể đổi lại mật khẩu khi thấy cần thiết nhằm bảo vệ tốt tài khoản của mình hơn
* Khách hàng có thể cập nhật thông tin cá nhân, thông tin tài khoản của mình.
* Khách hàng có thể xem thông tin chi tiết hoặc mua các gói bảo hiểm tương ứng thích hợp với từng loại xe của mình đang sở hữu và thời hạn muốn mua.
* Khách hàng có thể xem thông tin danh sách các gói bảo hiểm của mình đã mua (Số tiền, trạng thái thanh toán, ...).
* Sau khi mua bảo hiểm, nếu khách hàng gặp sự cố nào đó đối với gói bảo hiểm của mình đã mua thì có thể truy cập vào Website để khai báo hồ sơ và sau đó các nhân công ty sẽ đến hiện trường và giám định.
* Khách hàng có thể xem danh sách hồ sơ đã khai báo đối với gói bảo hiểm đã mua (Trạng thái xác nhận đền bù, Số tiền đền bù, …).
* Khách hàng có thể chọn phương thức thanh toán: Thanh toán khi mua bảo hiểm hoặc thanh toán bằng tài khoản ngân hàng.

**3.1.2 Bài toán đặt ra**

Xây dựng được một Website có thể quản lý được các hoạt động mua bán bảo hiểm như: Bán bảo hiểm, Xem thông tin danh sách bảo hiểm, Khai báo hồ sơ, Xem thông tin danh sách hồ sơ khai báo, thanh toán, …

**3.1.2.1 Xem thông tin bảo hiểm**

**-** Đăng ký tài khoản: Khách hàng chưa phải là thành viên thì phải đăng ký một tài khoản để đăng nhập vào Website. Khách hàng phải nhập đầy đủ các thông tin bắt buộc như Email, họ tên, địa chỉ,… để hoàn thành chức năng đăng ký.

**-** Kiểm tra tài khoản: Khi đăng ký tài khoản hệ thống sẽ kiểm tra email của tài khoản có tồn tại không, nếu có thì xác nhận khách hàng đăng ký thành công, nếu chưa thì khách hàng phải chọn email khác.

**-** Tìm lại mật khẩu: Khi không nhớ mật khẩu của mình khách hàng có thể vào trang tìm mật khẩu rồi điền thông tin gồm email, xác nhận captcha sau đó hệ thống sẽ kiểm tra và gửi một mật khẩu mới về email tài khoản đó nếu thông tin chính xác.

**3.1.2.2 Mua bảo hiểm**

**-** Đăng nhập: Khách hàng đã có tài hoản có thể đăng nhập để sử dụng các chức năng của một thành viên Website. Khách hàng phải điền đúng email và mật khẩu đã được đăng ký mới có thể đăng nhập thành công. Hệ thống có sử dụng thời gian để time out tài khoản thành viên của khách hàng.

- Cập nhật thông tin cá nhân: Khách hàng có thể cập nhật thông tin cá nhân của mình, đổi mật khẩu.

- Mua bảo hiểm: Sau khi đăng nhập thành công khách hàng có thể mua các gói bảo hiểm phù hợp và điền thông tin vào phiếu mua bảo hiểm: Biển số xe, số chỗ ngồi, thời hạn,…

- Xem danh sách bảo hiểm: Khách hàng có thể xem danh sách các gói bảo hiểm đã mua ở trang Danh sách bảo hiểm.

- Khai báo hồ sơ: Khi khách hàng có sự cố đối với gói bảo hiểm đã mua thì khách hàng có thể truy cập vào Website để khai báo hồ sơ, sau đó bên bộ phận quản lý sẽ cho nhân viên đến giám định hiện trường và xác nhận hồ sơ khai báo có đúng sự thật không và xác nhận được đền bù không.

- Xem danh sách hồ sơ khai báo: Khách hàng có thể xem thông tin danh sách các hồ sơ đã khai báo để biết được trạng thái được đền bù và số tiền đền bù từ gói bảo hiểm đã mua.

**3.1.2.3 Thanh toán**

- Thanh toán: Khi mua xong bảo hiểm, khách hàng có thể thanh toán bằng hai phương thức là thanh toán khi mua bảo hiểm hoặc tài khoản ngân hàng, nếu chọn thanh toán bằng tài khoản ngân hàng thì khách hàng có sử dụng Webserver để xác nhận mã OTP được gửi qua email của khách hàng trong quá trình chuyển tiền.

**3.1.2.4 Các chức năng khác**

**-** Hiển thị các thông tin liên quan đến bảo hiểm: tin khuyến mãi, tin giới thiệu,…

**-** Song ngữ (Việt – Anh, mặc định: tiếng việt).

**-** Hỗ trợ tư vấn trực tuyến.

**-** Gửi mail xác nhận khi mua bảo hiểm,thông báo gói bảo hiểm sắp hết hạn.

**-** Trang quản trị Website (admin page):

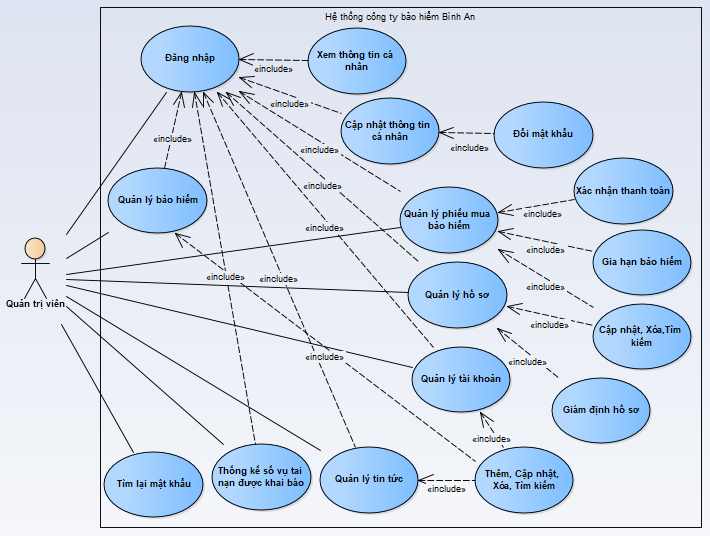
* Quản lý bảo hiểm
* Quản lý phiếu mua bảo hiểm
* Quản lý hồ sơ
* Quản lý tài khoản
* Quản lý tin tức
  1. **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**3.2.1 Biểu đồ Use Case**

Từ các yêu cầu về chức năng của hệ thống, ta có thể mô hình hóa các chức năng của hệ thống bởi Use case tổng quát sau:



*Hình 3.1* Biểu đồ Usecase phía khách hàng.



*Hình 3.2* Biểu đồ Usecase phía quản trị.

**3.2.2 Biểu đồ lớp**

**3.2.2.1 Tổng qua các lớp**

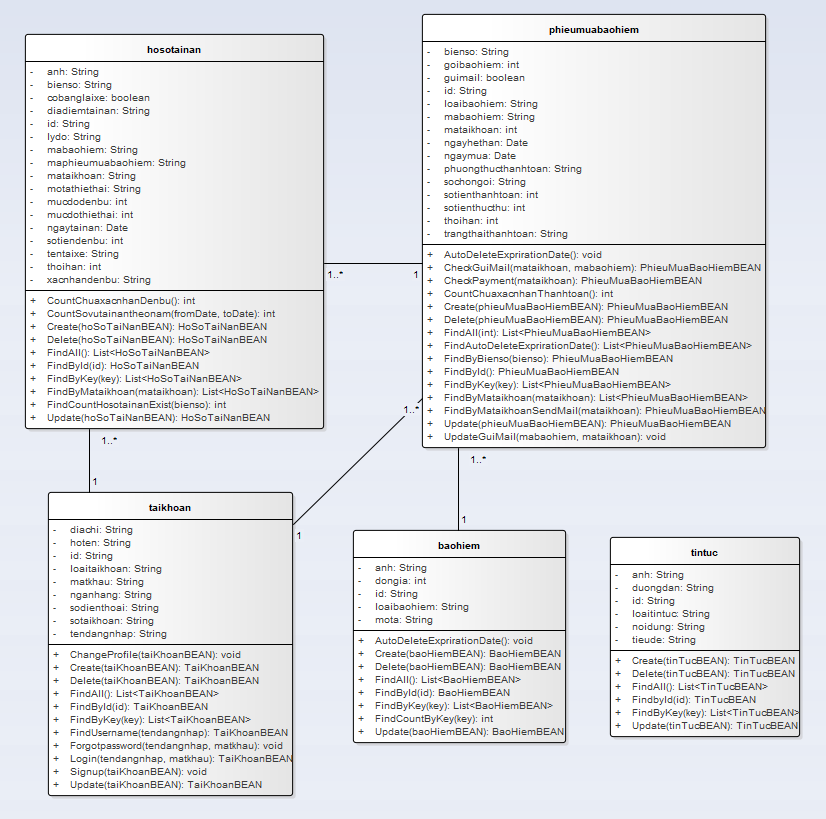
**-** Lớp bảo hiểm: Là lớp chứa thông tin các loại bảo hiểm mà khách hàng cần mua.

**-** Lớp phiếu mua bảo hiểm: Là lớp chứa thông tin khi khách hàng mua khi bảo hiểm, bao gồm các thông tin như: biển số xe, gói bảo hiểm, thời hạn, ngày mua, …

**-** Lớp hồ sơ tai nạn: Là lớp chứa thông tin hồ sơ tai nạn khi khách hàng khai báo đối với gói bảo hiểm đã mua, bao gồm các thông tin như: biển số xe, địa điểm tai nạn, ngày tai nạn, thiệt hại, …

**-** Lớp tài khoản: Là lớp chứa thông tin của khách hàng và quản trị viên.

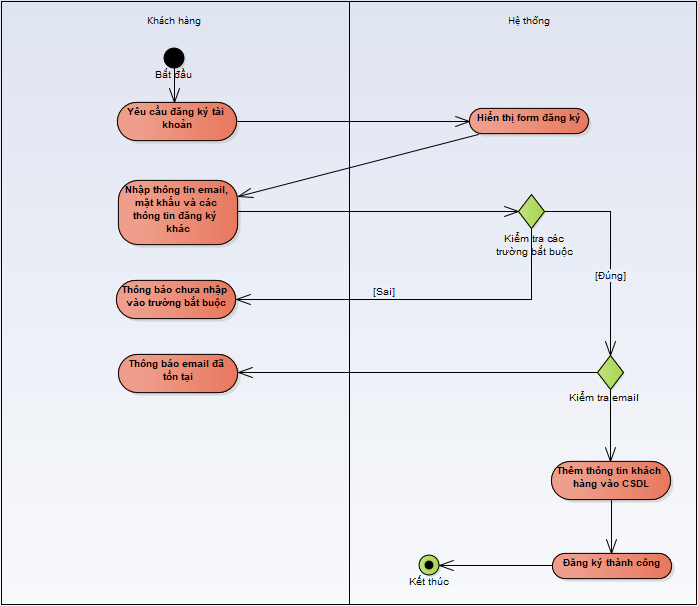
**-** Lớp tin tức: Là lớp chứa thông tin các tin tức liên quan đến bảo hiểm như: tin giảm giá, tin giới thiệu, hỗ trợ, …



*Hình 3.3* Biểu đồ lớp.

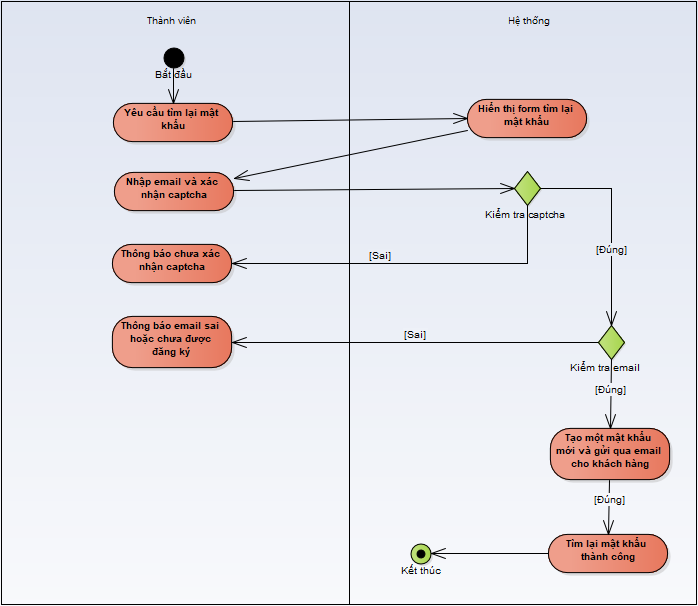
**3.2.3 Biểu đồ hoạt động**

**3.2.3.1 Đăng ký tài khoản**



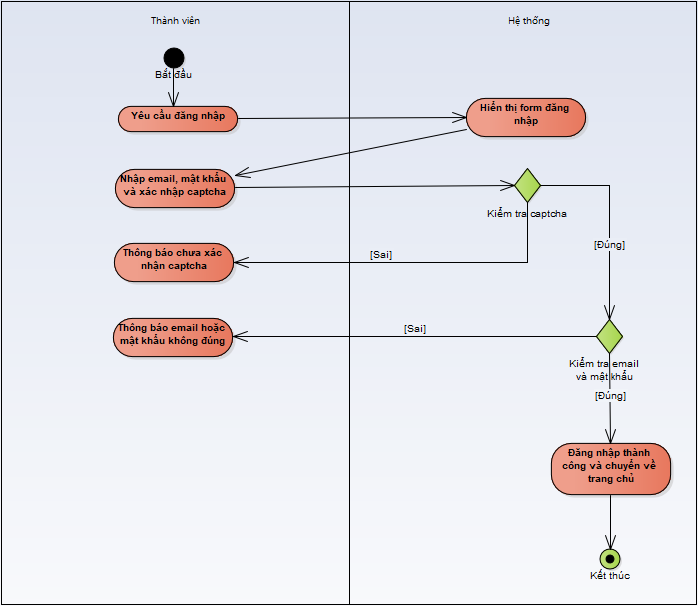
*Hình 3.4* Biểu đồ hoạt động chức năng đăng ký tài khoản.

**3.2.3.2 Tìm lại mật khẩu**



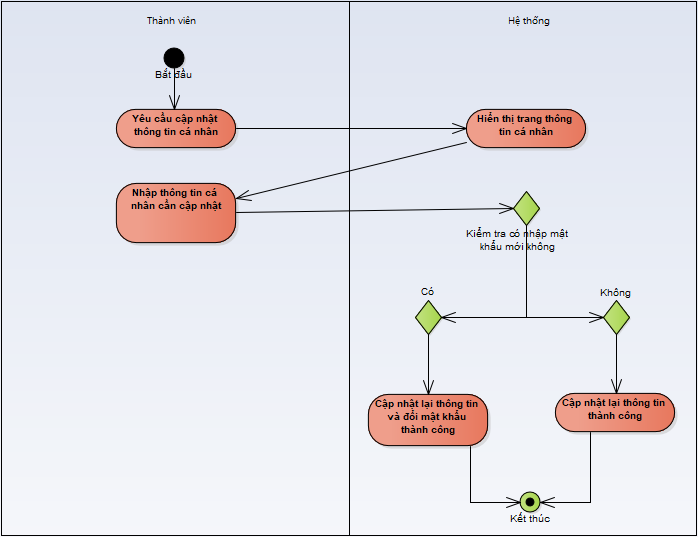
*Hình 3.5* Biểu đồ hoạt động chức năng tìm lại mật khẩu.

**3.2.3.3 Đăng nhập**



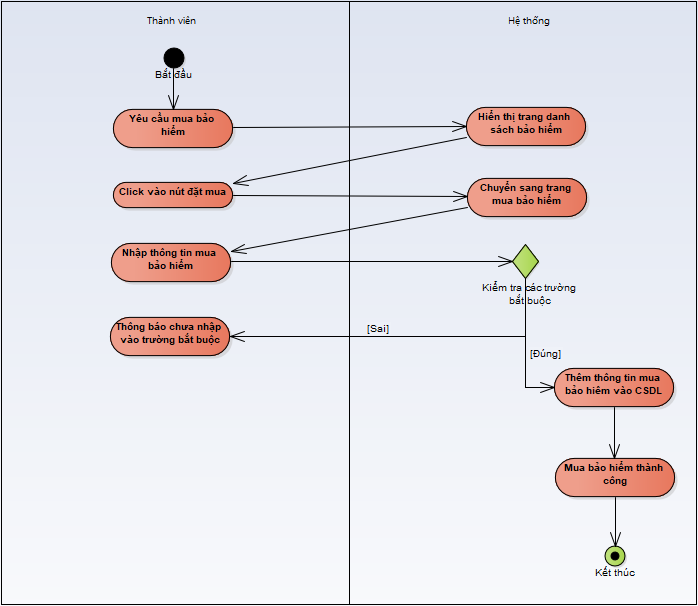
*Hình 3.6* Biểu đồ hoạt động chức năng đăng nhập.

**3.2.3.4 Cập nhật thông tin cá nhân**



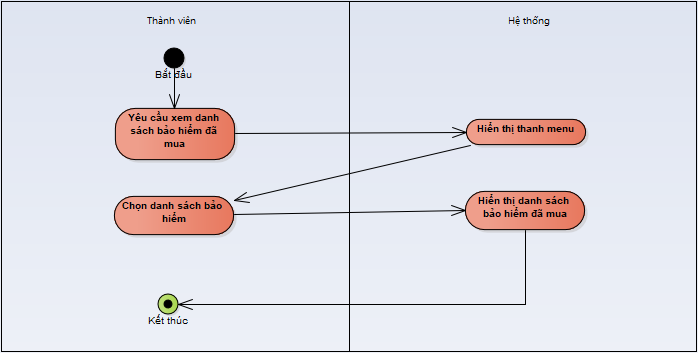
*Hình 3.7* Biểu đồ hoạt động chức năng cập nhật thông tin cá nhân.

**3.2.3.5 Mua bảo hiểm**



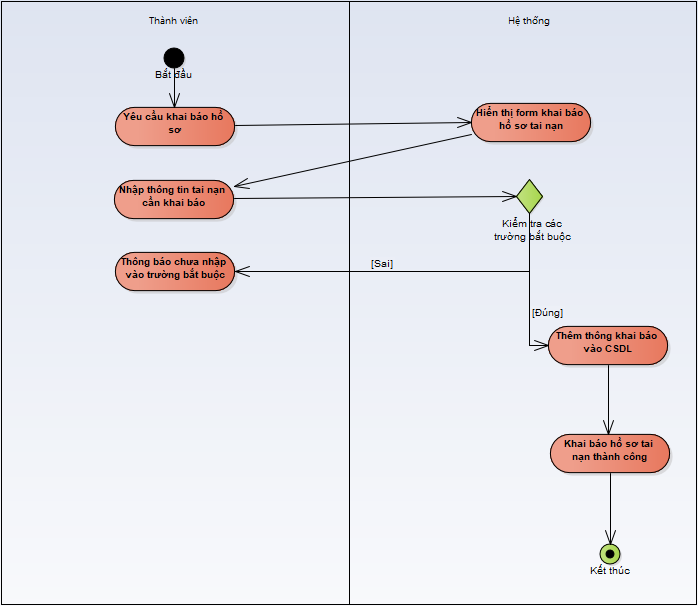
*Hình 3.8* Biểu đồ hoạt động chức năng mua bảo hiểm.

**3.2.3.6 Xem danh sách bảo hiểm đã mua**



*Hình 3.9* Biểu đồ hoạt động chức năng xem danh sách bảo hiểm đã mua.

**3.2.3.7 Khai báo hồ sơ**



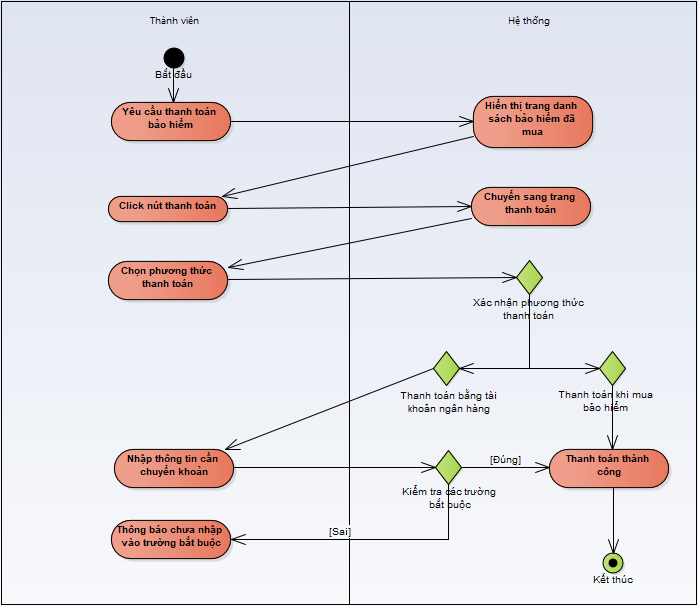
*Hình 3.10* Biểu đồ hoạt động chức năng khai báo hồ sơ.

**3.2.3.8 Xem danh sách hồ sơ khai báo**



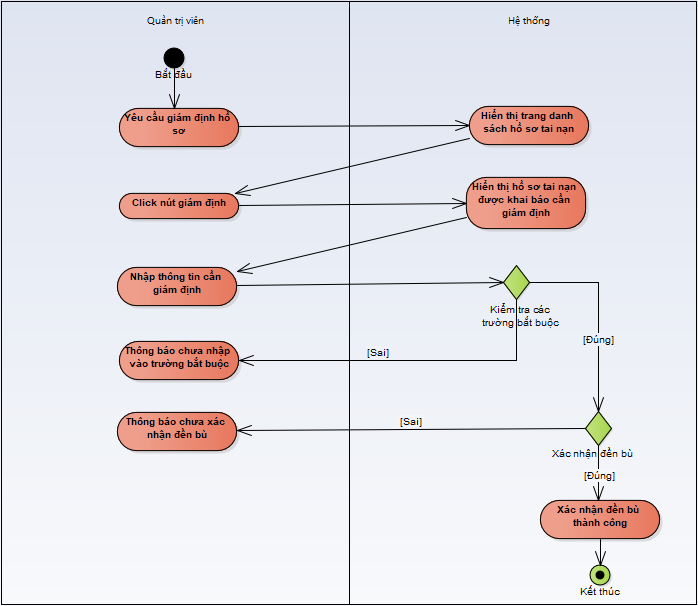
*Hình 3.11* Biểu đồ hoạt động chức năng xem danh sách hồ sơ khai báo.

**3.2.3.9 Thanh toán**



*Hình 3.12* Biểu đồ hoạt động chức năng thanh toán.

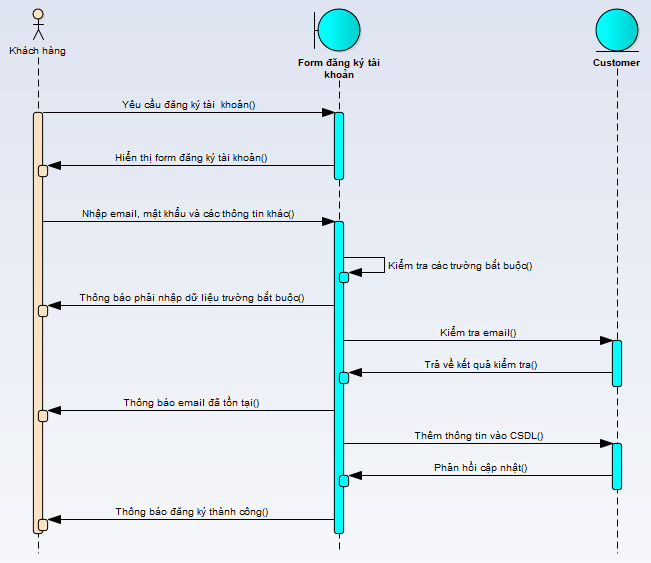
**3.2.3.10 Giám định hồ sơ**



*Hình 3.13* Biểu đồ hoạt động chức năng giám định hồ sơ.

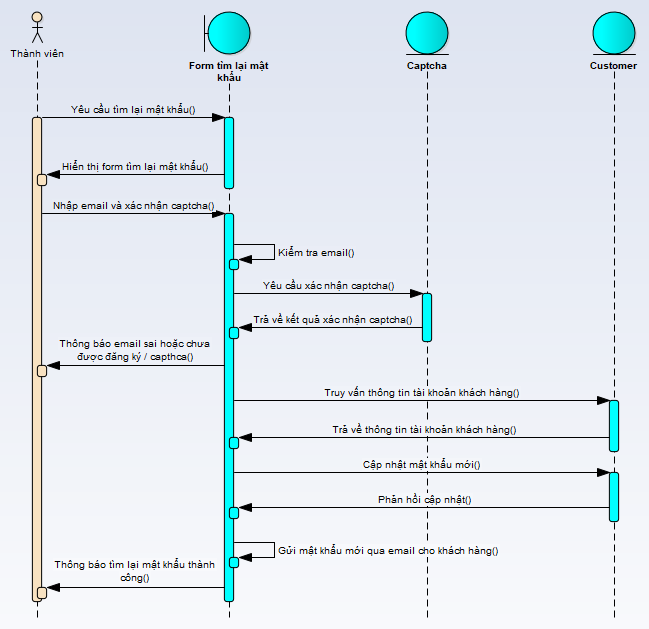
**3.2.4 Biểu đồ tuần tự**

**3.2.4.1 Đăng ký tài khoản**



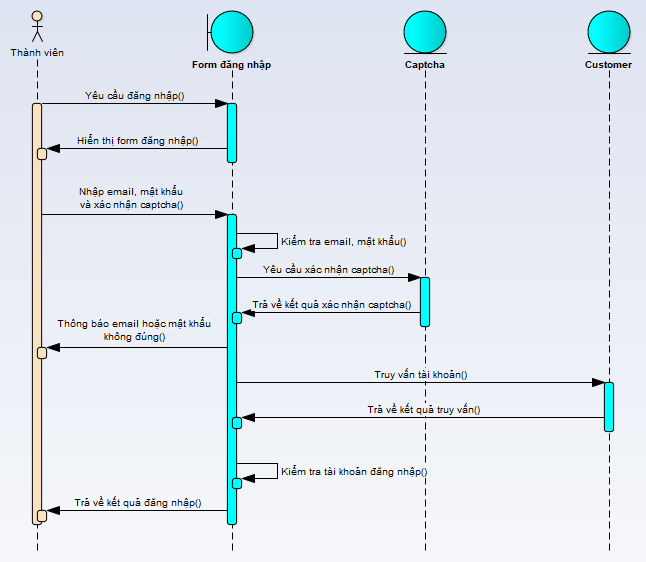
*Hình 3.14* Biểu đồ tuần tự chức năng đăng ký tài khoản.

**3.2.4.2 Tìm lại mật khẩu**



*Hình 3.15* Biểu đồ tuần tự chức năng tìm lại mật khẩu.

**3.2.4.3 Đăng nhập**



*Hình 3.16* Biểu đồ tuần tự chức năng đăng nhập.

**3.2.4.4 Cập nhật thông tin cá nhân**



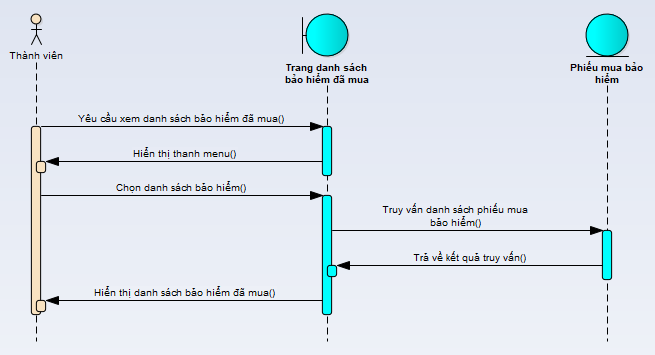
*Hình 3.17* Biểu đồ tuần tự chức năng cập nhật thông tin cá nhân.

**3.2.4.5 Mua bảo hiểm**



*Hình 3.18* Biểu đồ tuần tự chức năng mua bảo hiểm.

**3.2.4.6 Xem danh sách bảo hiểm đã mua**



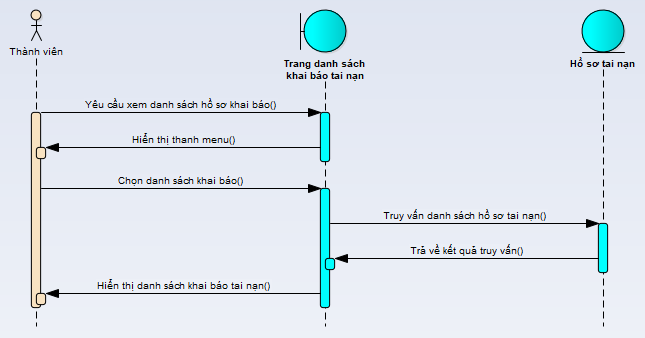
*Hình 3.19* Biểu đồ tuần tự chức năng xem danh sách bảo hiểm đã mua.

**3.2.4.7 Khai báo hồ sơ**



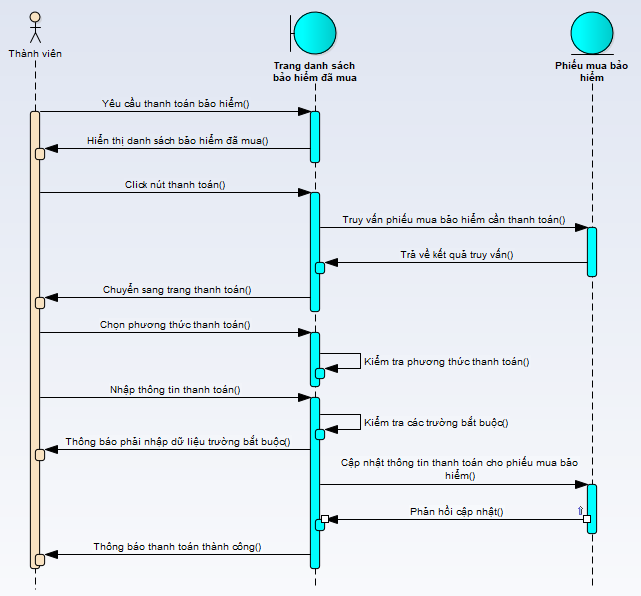
*Hình 3.20* Biểu đồ tuần tự chức năng khai báo hồ sơ.

**3.2.4.8 Xem danh sách hồ sơ khai báo**



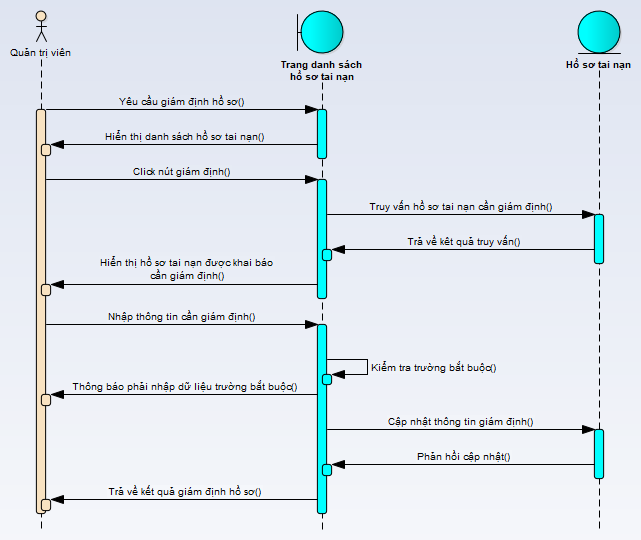
*Hình 3.21* Biểu đồ tuần tự chức năng xem danh sách hồ sơ khai báo.

**3.2.4.9 Thanh toán**



*Hình 3.22* Biểu đồ tuần tự chức năng thanh toán.

**3.2.4.10 Giám định hồ sơ**



*Hình 3.23* Biểu đồ tuần tự chức năng giám định hồ sơ.

**3.2.5 Thiết kế cơ sở dữ liệu**

**3.2.5.1 Các đối tượng trong bảng**

**-** Bảo hiểm (baohiem):

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| id | String | Mã bảo hiểm, khóa chính của bảng |
| loaibaohiem | String | Loại bảo hiểm |
| dongia | Integer | Đơn giá |
| anh | String | Ảnh |
| mota | String | Mô tả bảo hiểm |

**-** Phiếu mua bảo hiểm (phieumuabaohiem):

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| id | String | Mã phiếu mua bảo hiểm, khóa chính của bảng |
| bienso | String | Biển số xe |
| sochongoi | String | Số chỗ ngồi |
| ngaymua | Timestamp | Ngày mua |
| ngayhethan | Timestamp | Ngày hết hạn |
| thoihan | Integer | Thời hạn |
| goibaohiem | Integer | Gói bảo hiểm |
| sotienthucthu | Integer | Số tiền thực thu |
| phuongthucthanhtoan | String | Phương thức thanh toán |
| sotienthanhtoan | Integer | Số tiền thanh toán |
| trangthaithanhtoan | String | Trạng thái thanh toán |
| guimail | Boolean | Trạng thái gửi mail |
| mabaohiem | String | Mã bảo hiểm |
| loaibaohiem | String | Loại bảo hiểm |
| mataikhoan | String | Mã tài khoản |

**-** Hồ sơ tai nạn (hosotainan):

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| id | String | Mã hồ sơ tai nạn, khóa chính của bảng |
| bienso | String | Biển số xe |
| ngaytainan | Timestamp | Ngày tai nạn |
| tentaixe | String | Tên tài xế |
| cobanglaixe | Boolean | Trạng thái có bằng lái xe |
| diadiemtainan | String | Địa điểm tai nạn |
| motathiethai | String | Mô tả thiệt hại |
| anh | String | Ảnh |
| mucdothiethai | Integer | Mức độ thiệt hại |
| mucdodenbu | Integer | Mức độ đền bù |
| thoihan | Integer | Thời hạn |
| sotiendenbu | Integer | Số tiền đền bù |
| lydo | String | Lý do không đền bù |
| xacnhandenbu | String | Xác nhận đền bù |
| mabaohiem | String | Mã bảo hiểm |
| mataikhoan | String | Mã tài khoản |
| maphieumuabaohiem | String | Mã phiếu mua bảo hiểm |

**-** Tài khoản (taikhoan):

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| id | String | Mã tài khoản, khóa chính của bảng |
| tendangnhap | String | Tên đăng nhập (email) |
| matkhau | String | Mật khẩu |
| loaitaikhoan | String | Loại tài khoản |
| hoten | String | Họ tên |
| diachi | String | Địa chỉ |
| sodienthoai | String | Số điện thoại |
| nganhang | String | Tên ngân hàng |
| sotaikhoan | String | Số tài khoản ngân hàng |

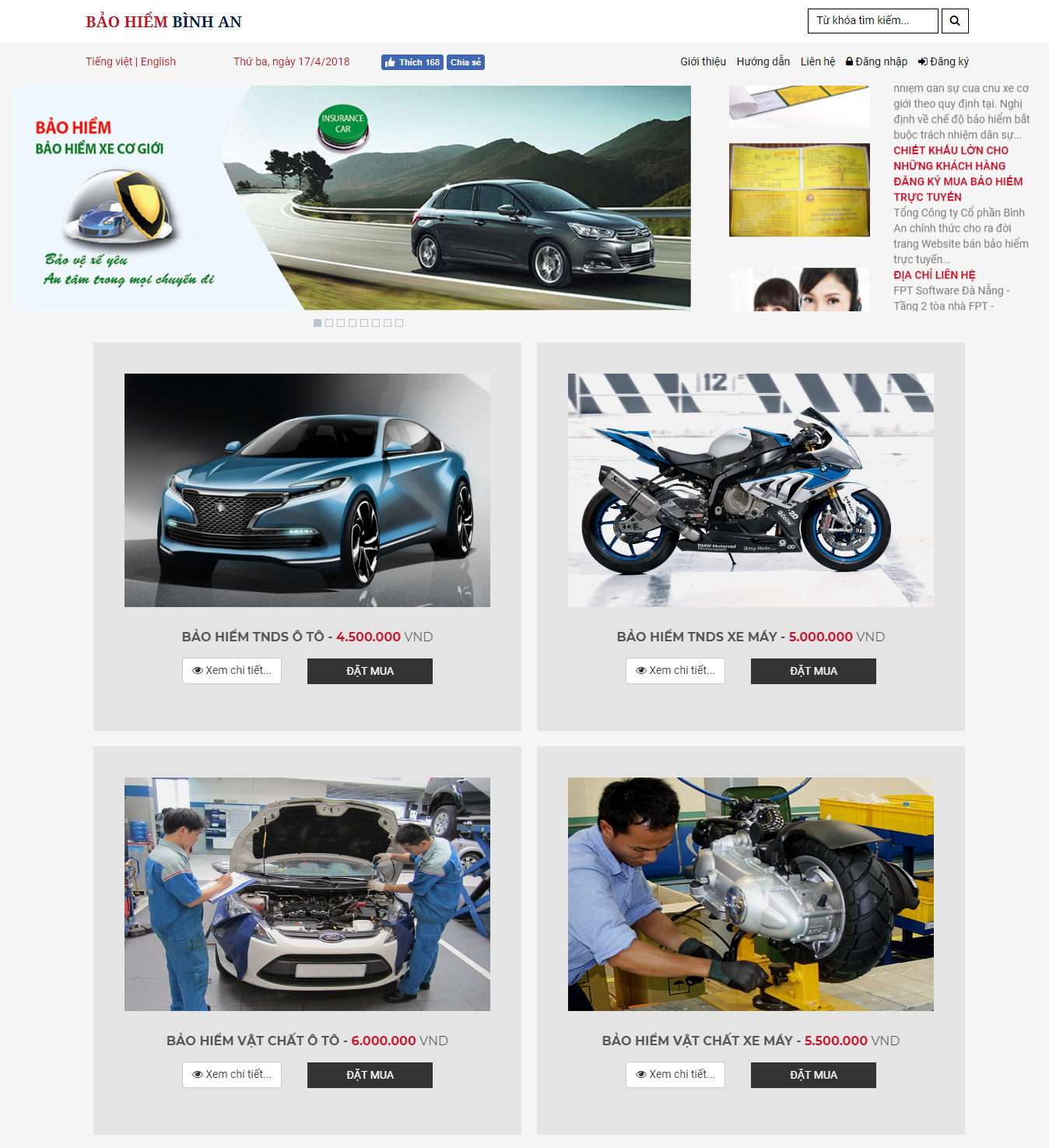
**-** Tin tức (tintuc):

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| id | String | Mã tin tức, khóa chính của bảng |
| loaitintuc | String | Loại tin tức |
| tieude | String | Tiêu đề |
| noidung | String | Nội dung |
| anh | String | Ảnh |
| duongdan | String | Đường dẫn liên kết |

* 1. **CÀI ĐẶT CHƯƠNG TRÌNH**

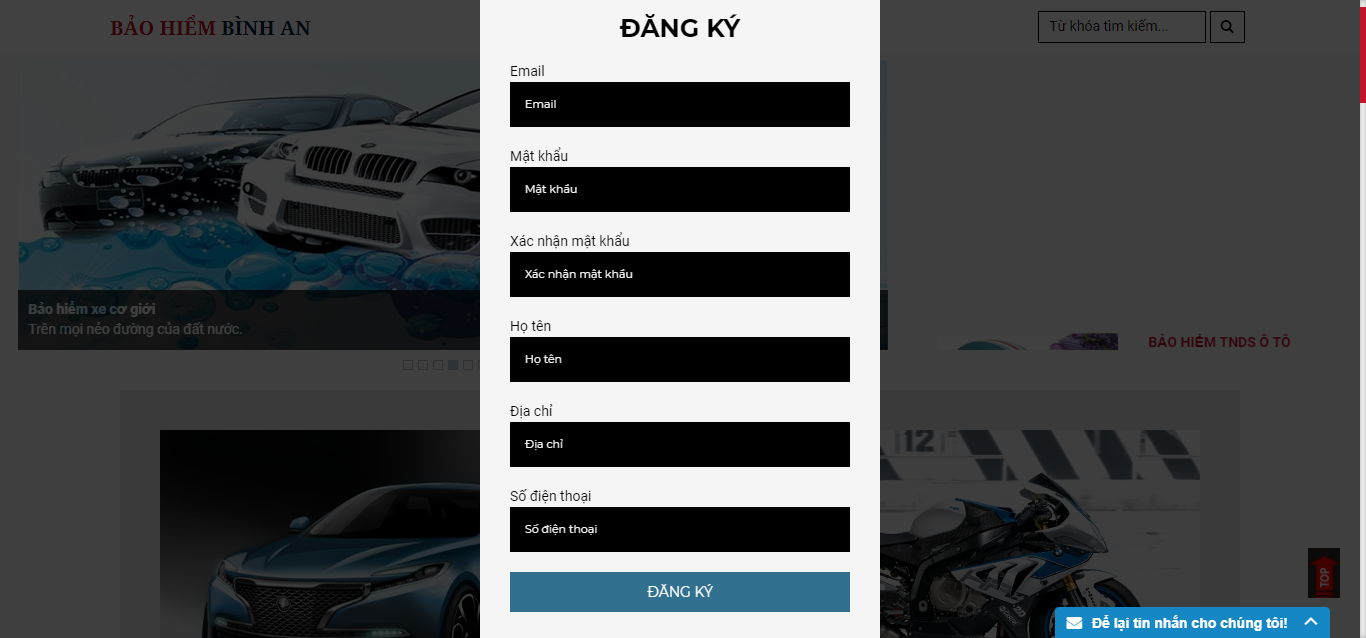
**3.3.1 Giao diện phía khách hàng**

**-** Trang chủ (Chưa đăng nhập)



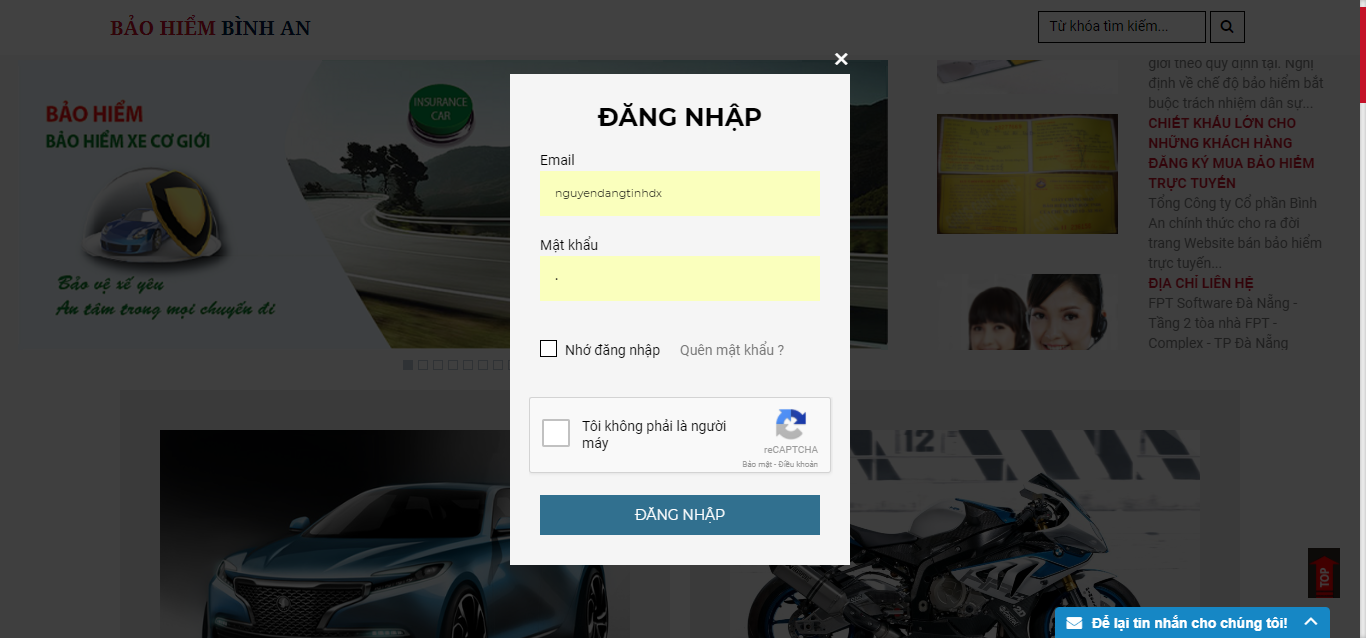
*Hình 3.24* Giao diện trang chủ khi chưa đăng nhập.

**-** Đăng ký



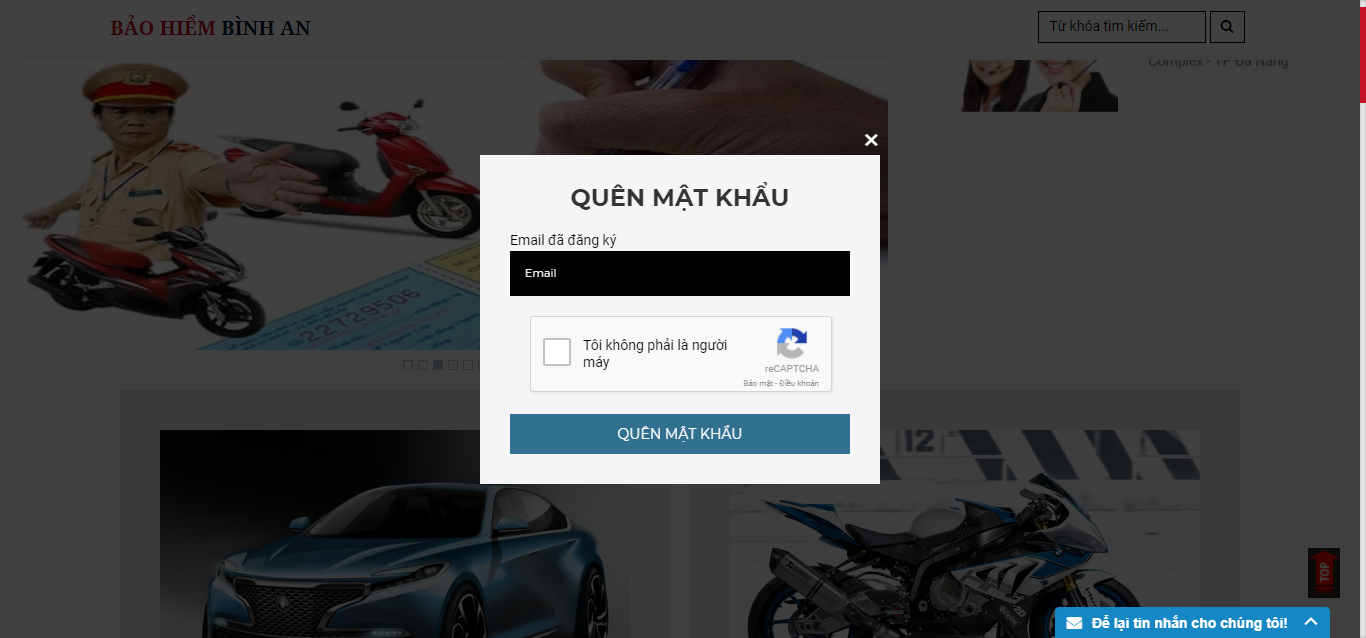
*Hình 3.25* Giao diện trang đăng ký.

**-** Đăng nhập



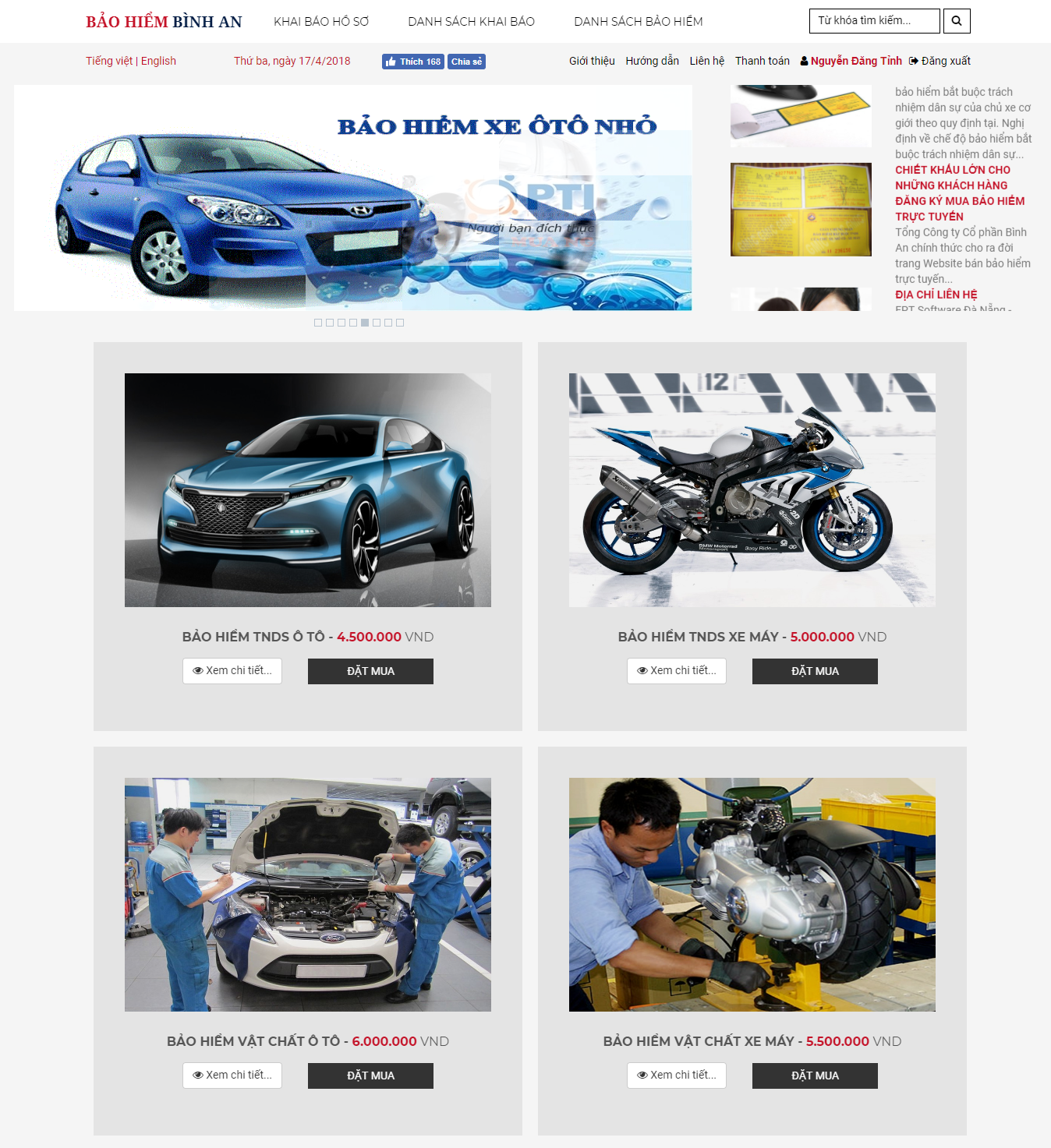
*Hình 3.26* Giao diện trang đăng nhập.

**-** Tìm lại mật khẩu



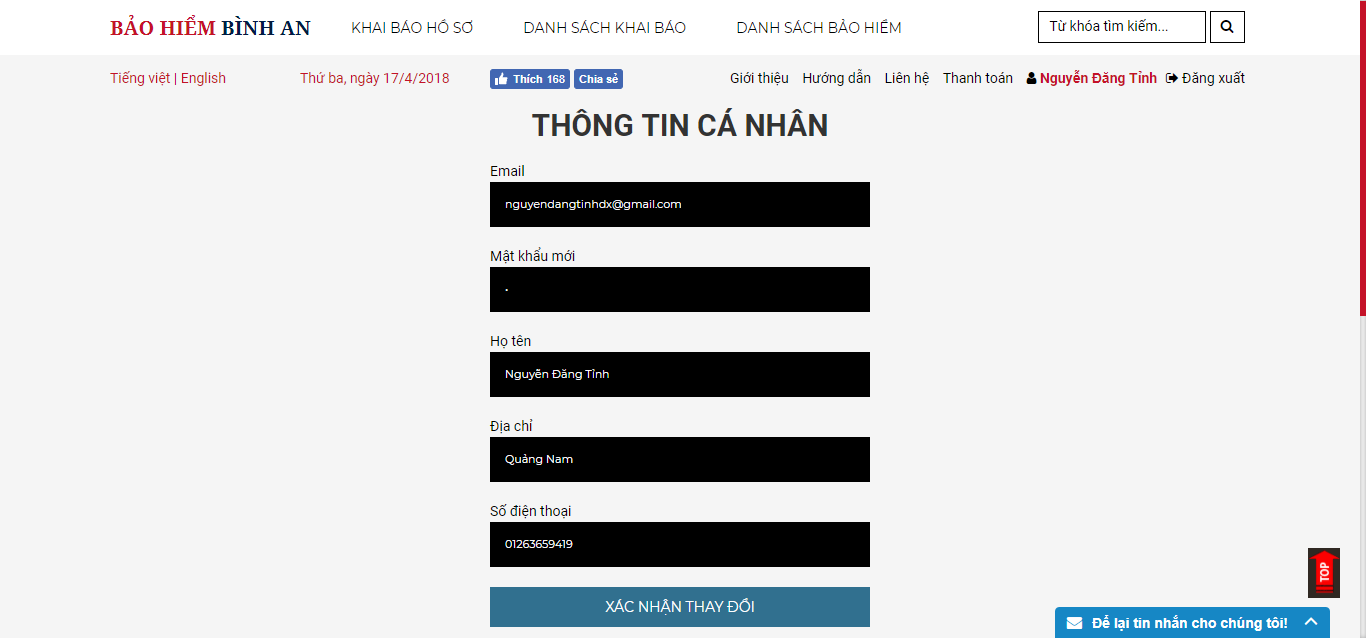
*Hình 3.27* Giao diện trang tìm lại mật khẩu.

**-** Trang chủ (Đã đăng nhập)



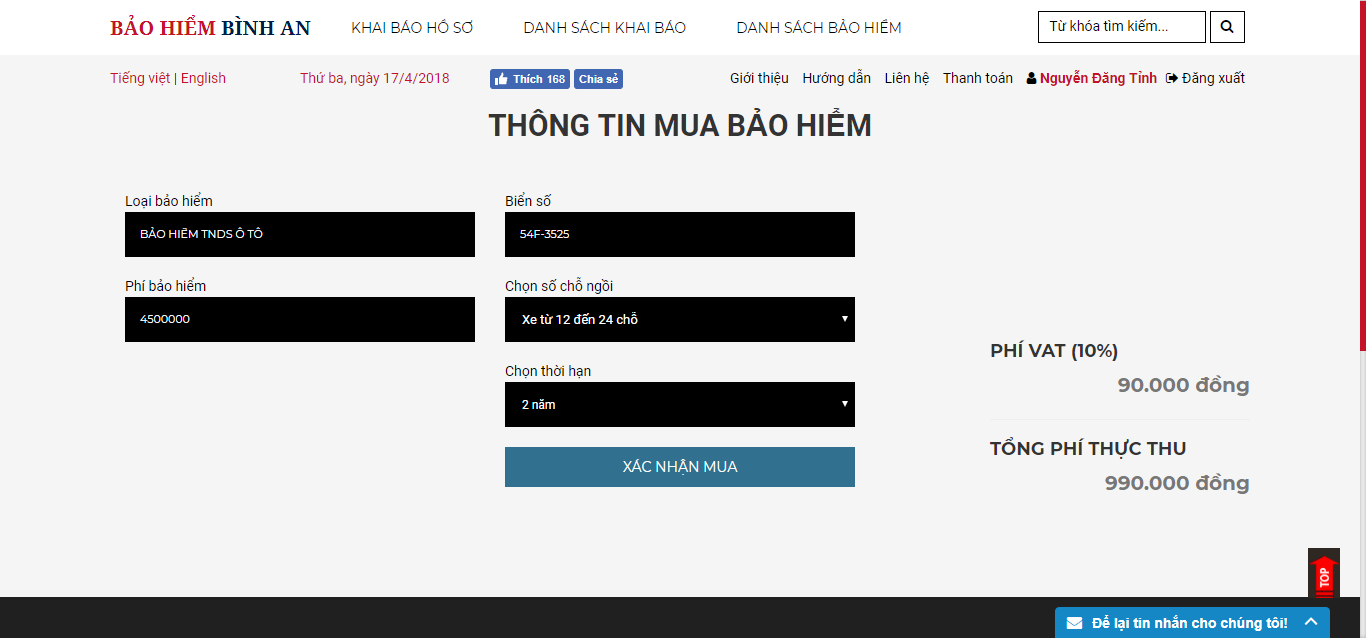
*Hình 3.28* Giao diện trang chủ khi đã đăng nhập.

**-** Thông tin cá nhân



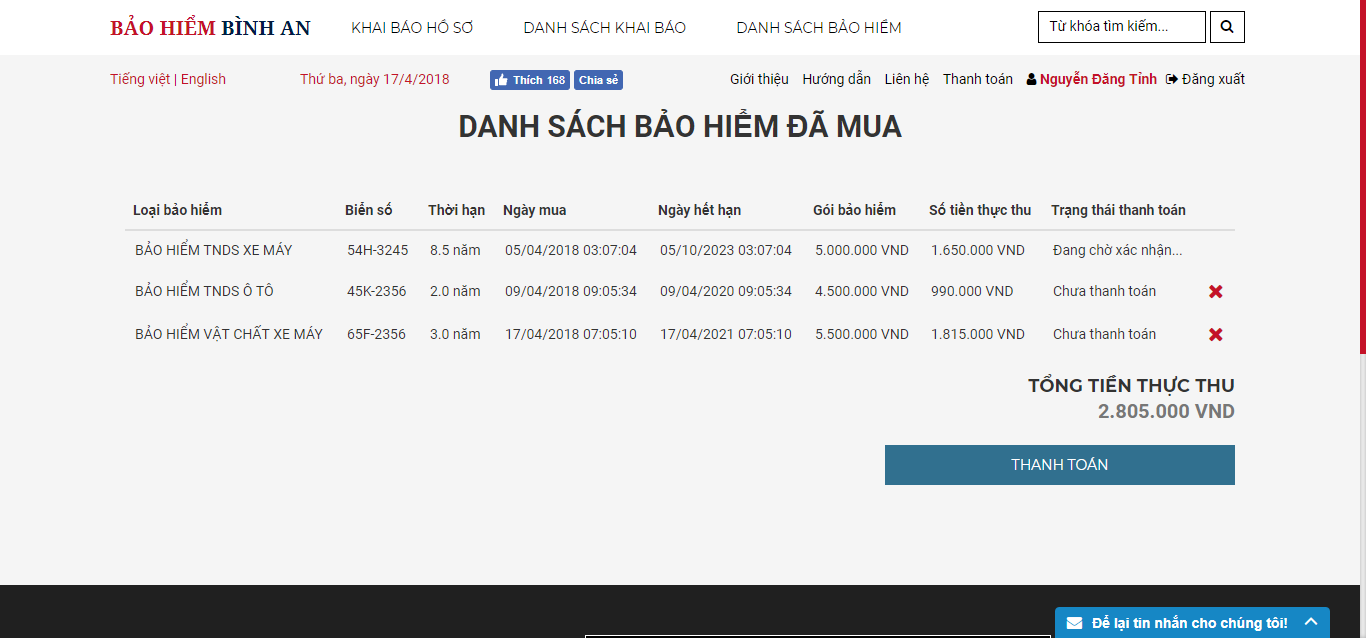
*Hình 3.29* Giao diện trang thông tin cá nhân.

**-** Mua bảo hiểm



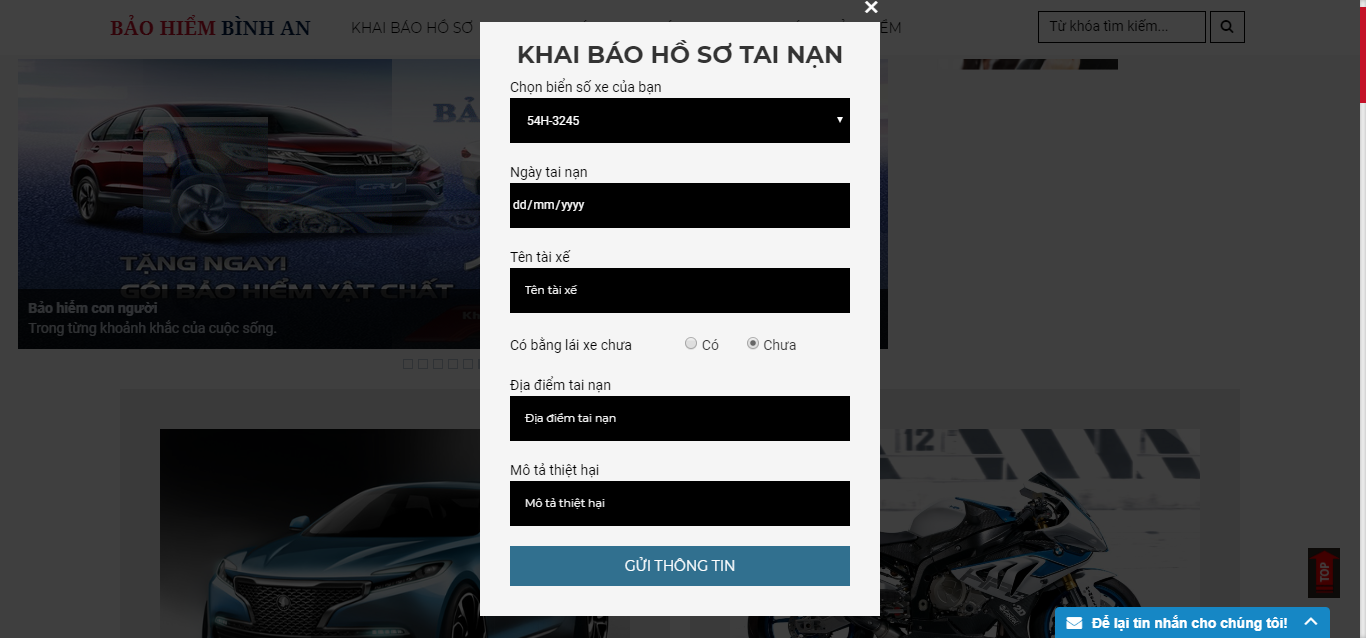
*Hình 3.30* Giao diện trang mua bảo hiểm.

**-** Xem bảo hiểm đã mua



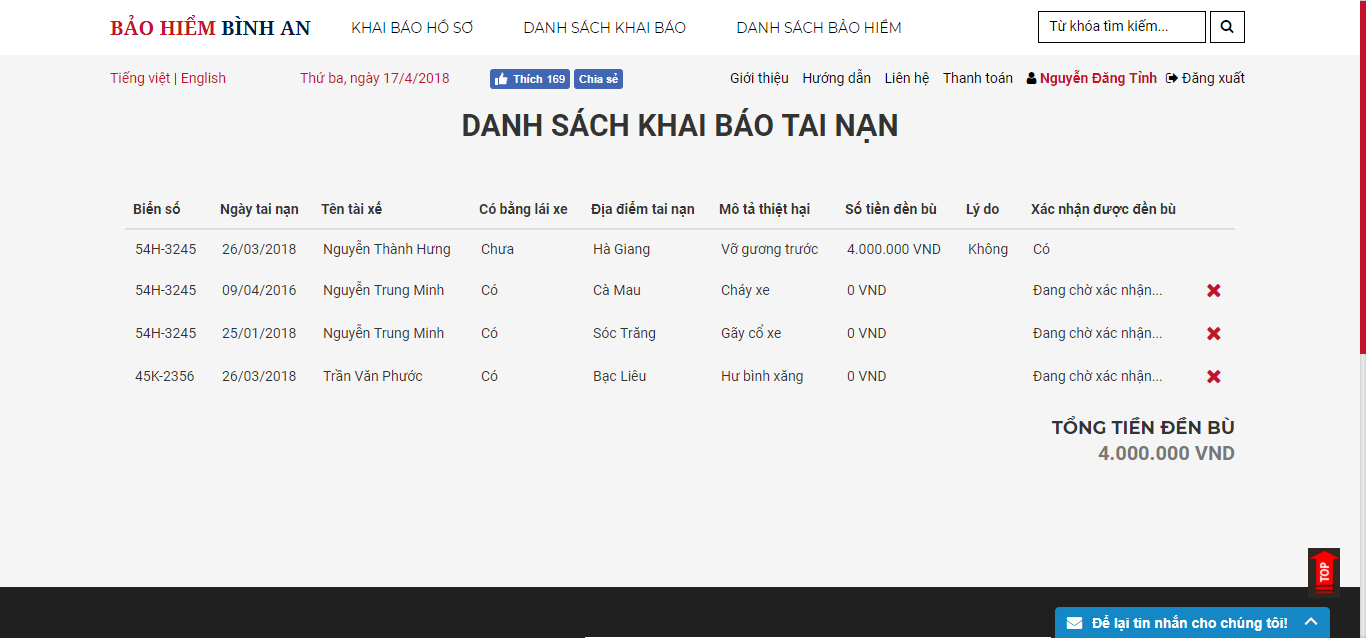
*Hình 3.31* Giao diện trang danh sách bảo hiểm đã mua.

**-** Khai báo hồ sơ tai nạn



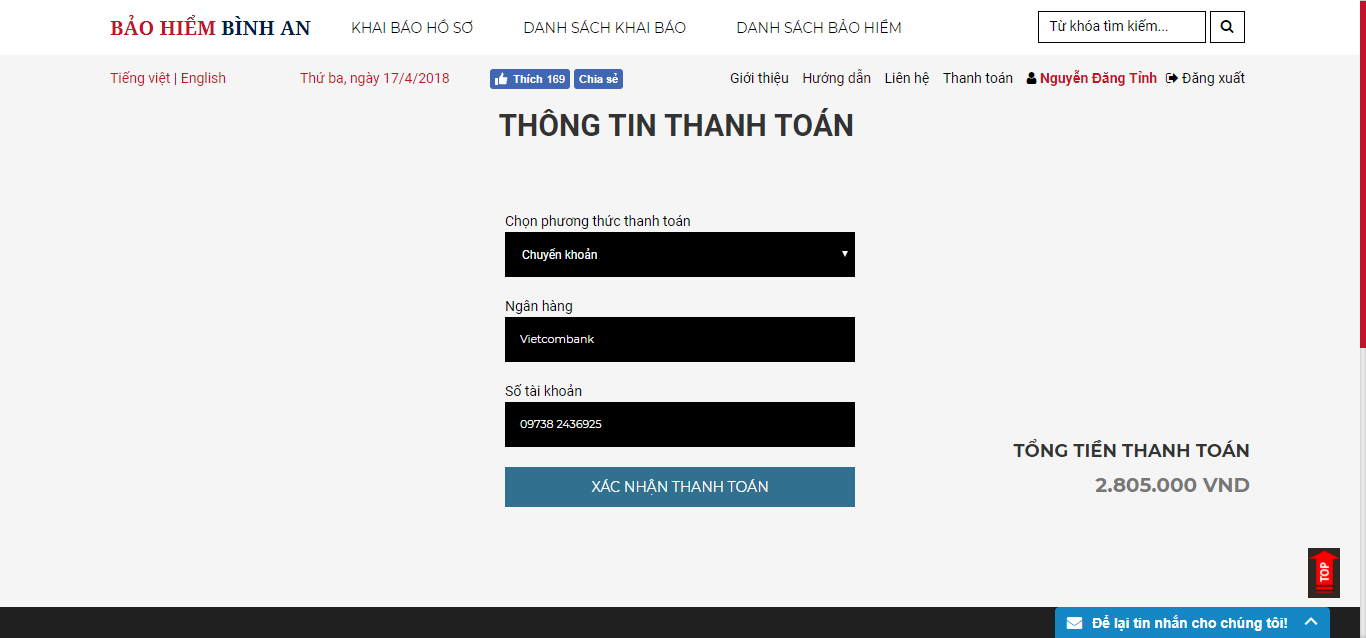
*Hình 3.32* Giao diện trang khai báo hồ sơ tai nạn.

**-** Xem khai báo tai nạn



*Hình 3.33* Giao diện trang danh sách khai báo tai nạn.

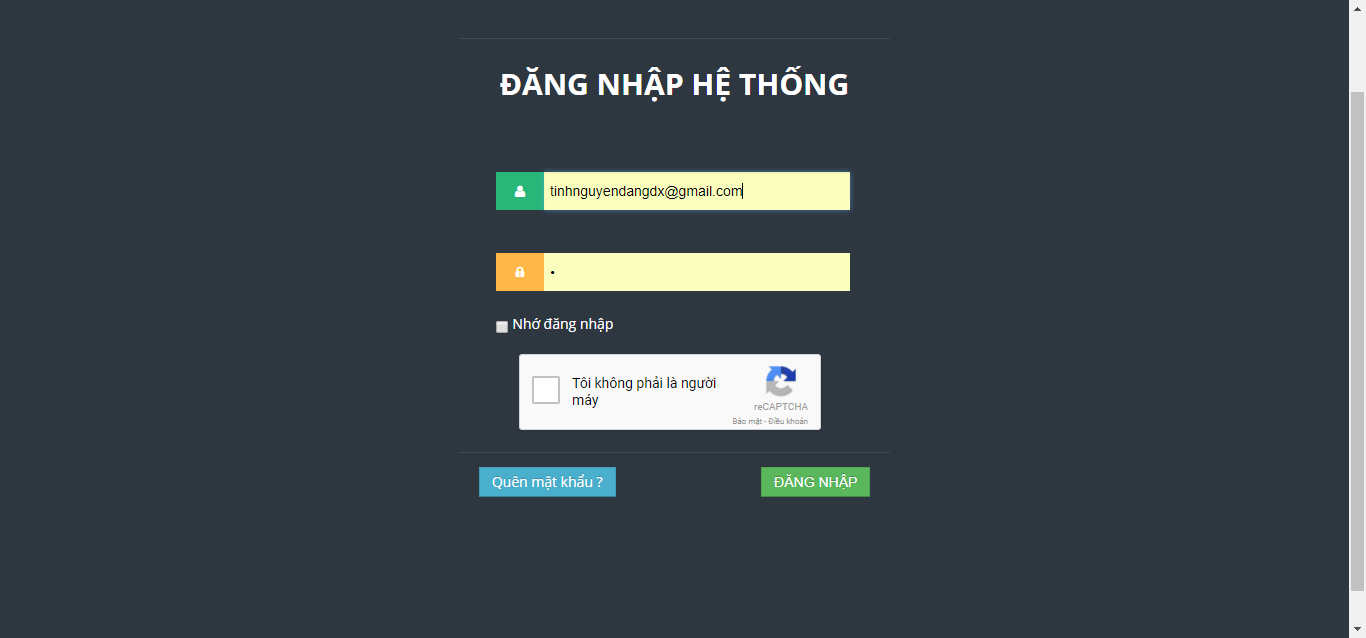
**-** Thanh toán



*Hình 3.34* Giao diện trang thanh toán.

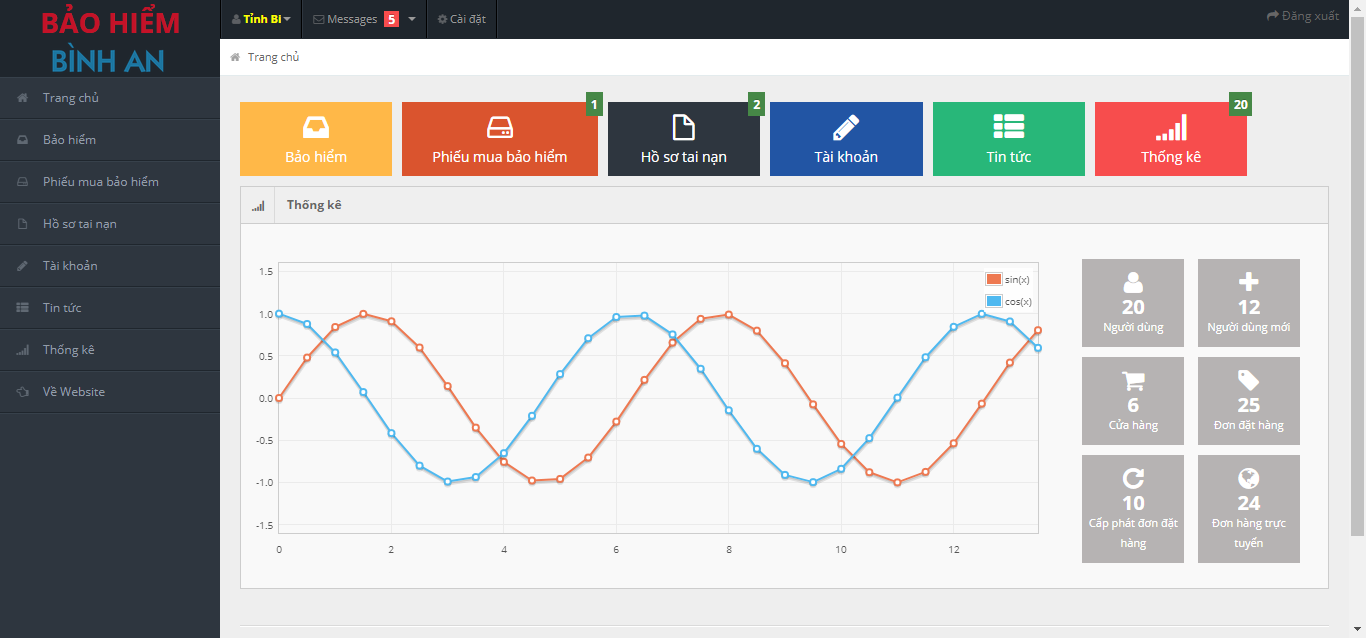
**3.3.2 Giao diện phía quản trị**

**-** Đăng nhập



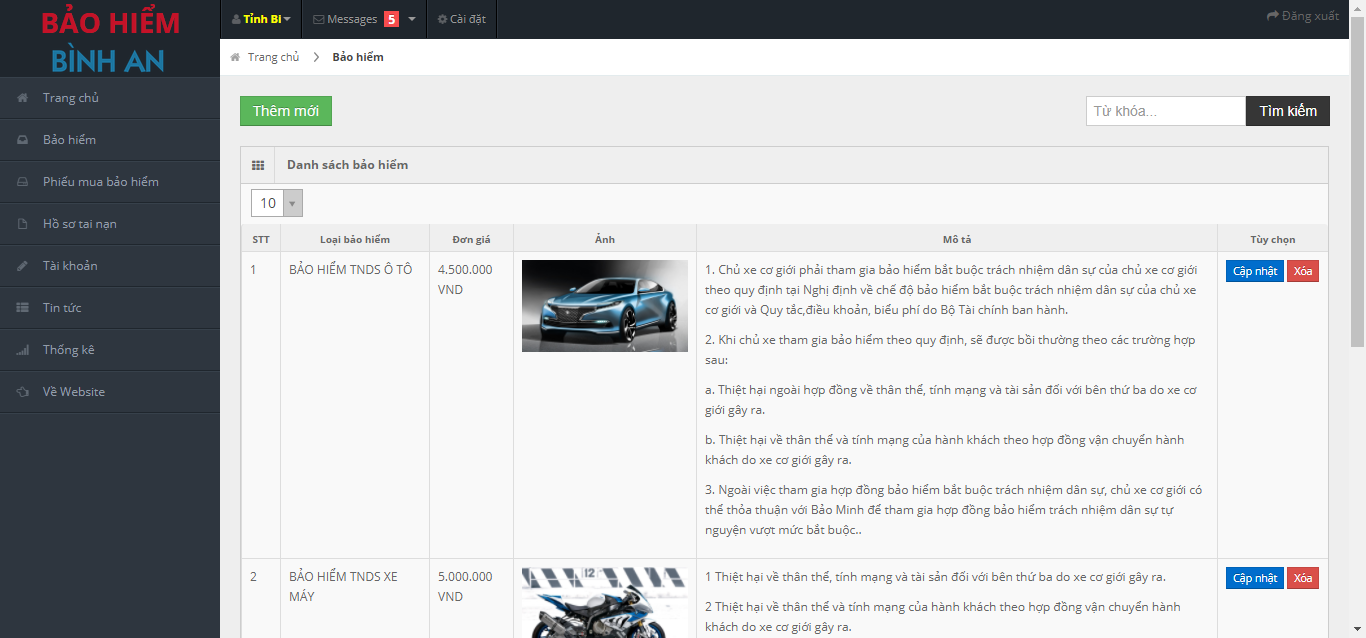
*Hình 3.35* Giao diện trang đăng nhập.

**-** Trang chủ



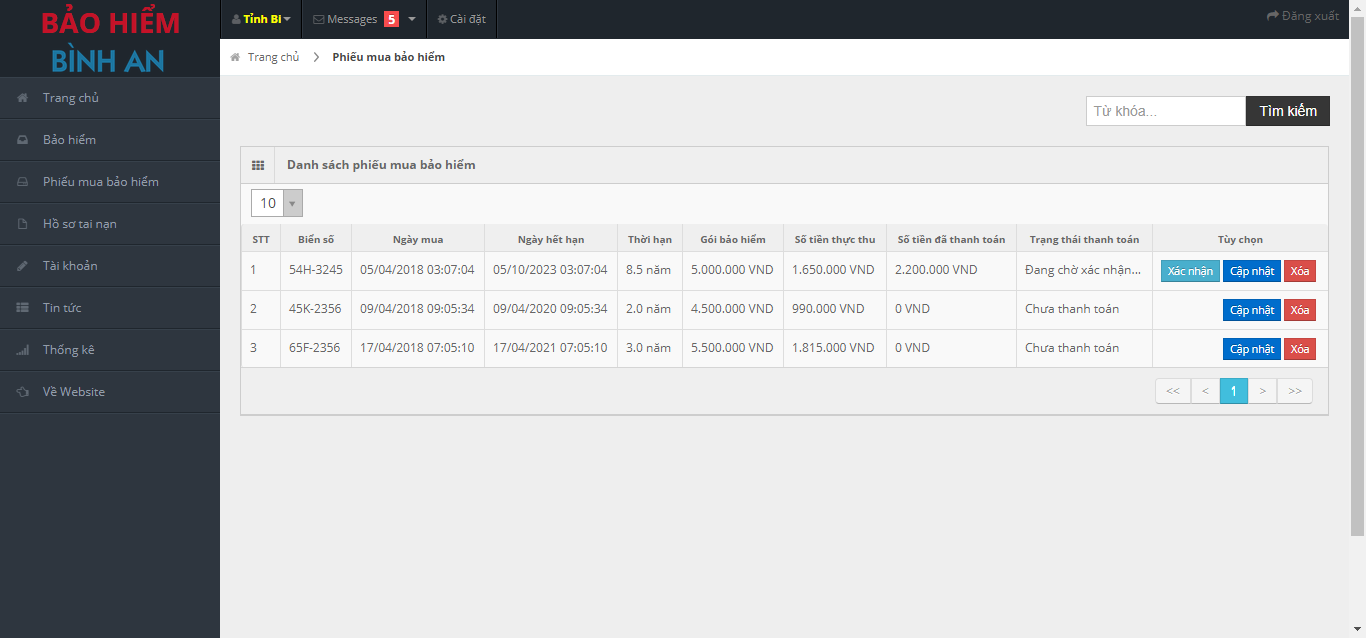
*Hình 3.36* Giao diện trang chủ.

**-** Quản lý bảo hiểm



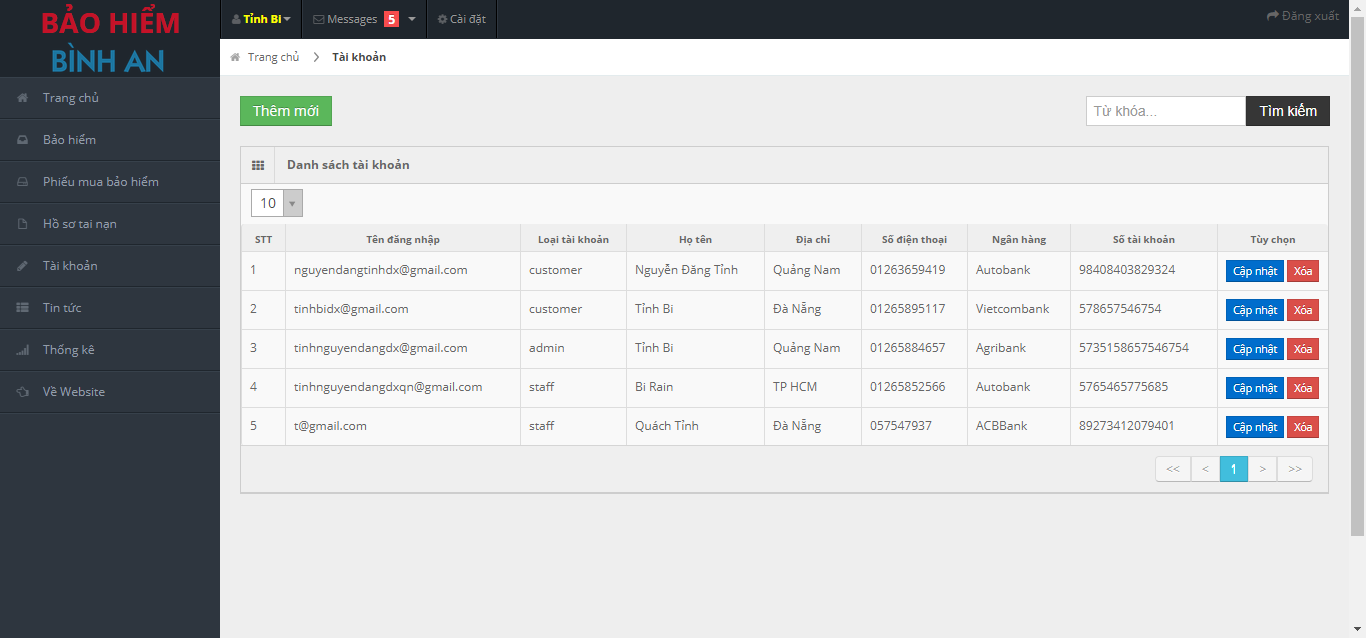
*Hình 3.37* Giao diện trang quản lý bảo hiểm.

**-** Quản lý phiếu mua bảo hiểm



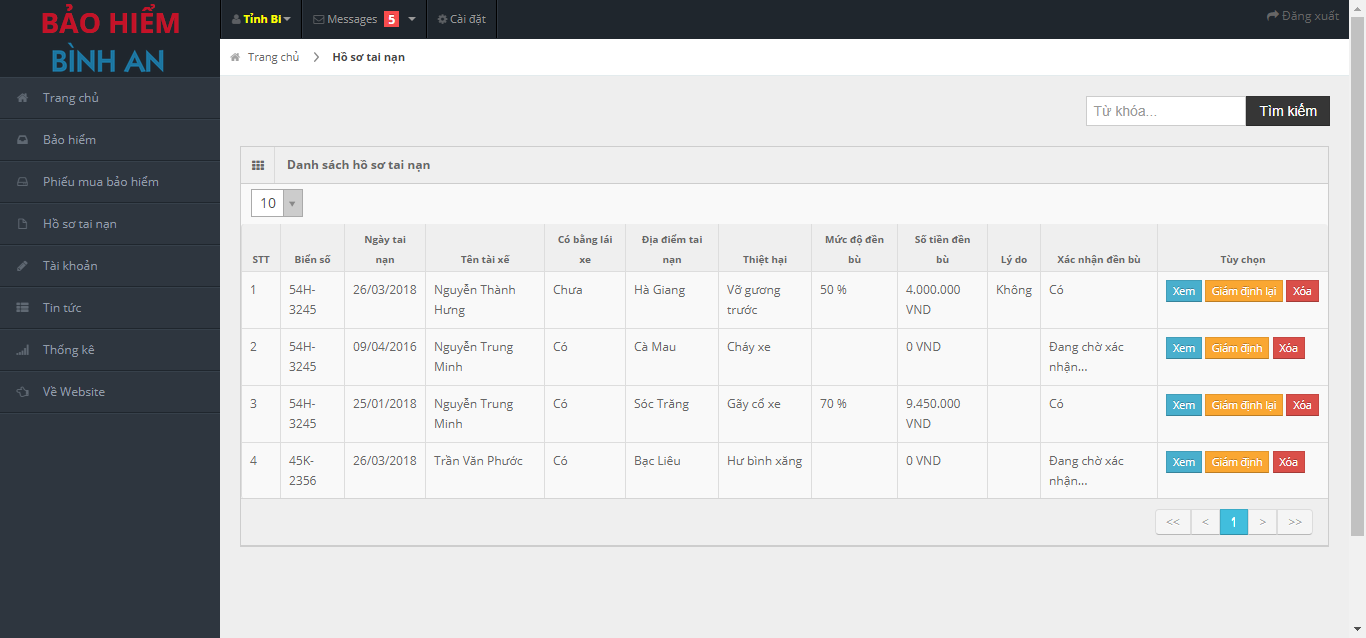
*Hình 3.38* Giao diện trang quản lý phiếu mua bảo hiểm.

**-** Quản lý tài khoản



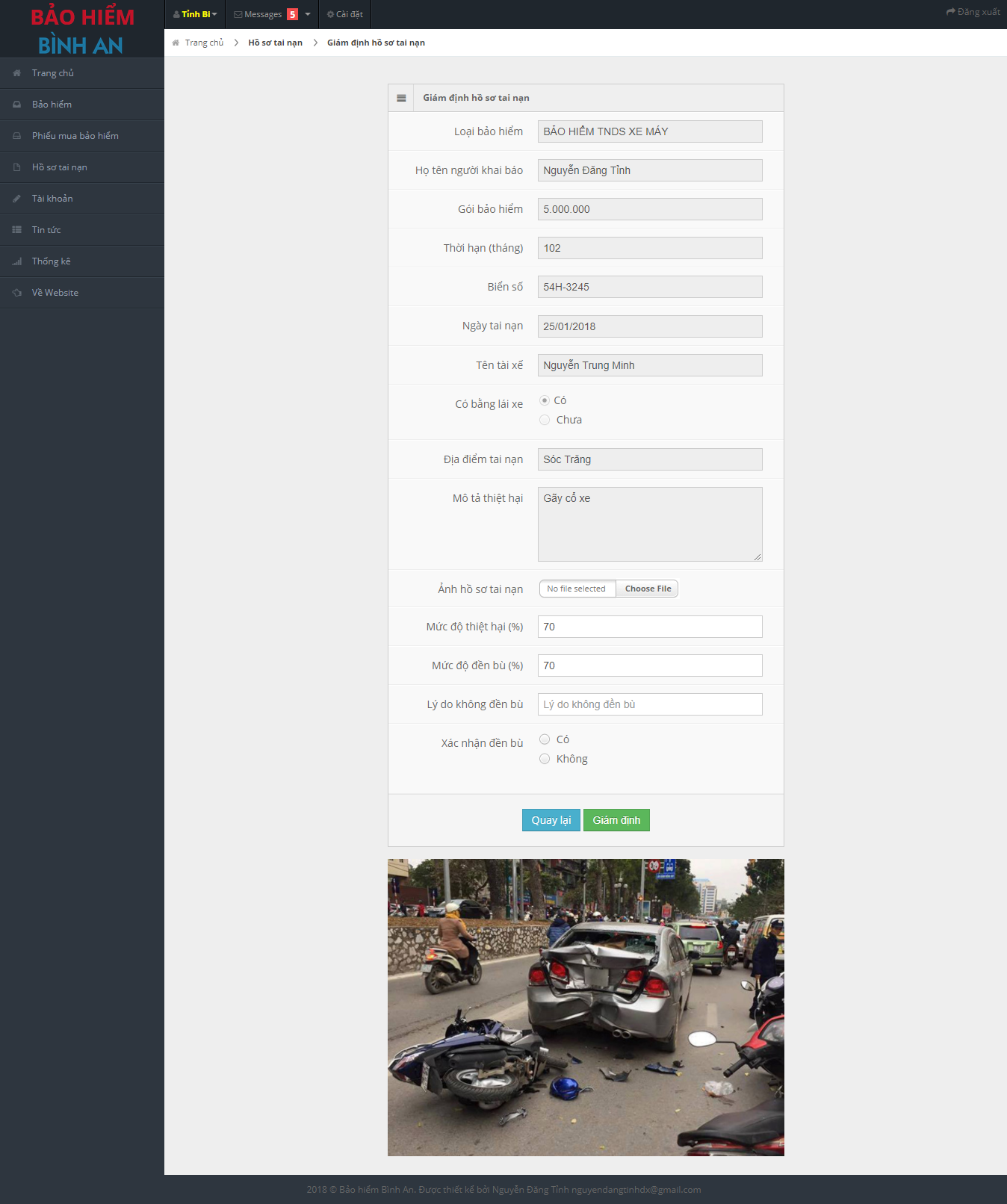
*Hình 3.38* Giao diện trang quản lý tài khoản.

**-** Quản lý hồ sơ tai nạn



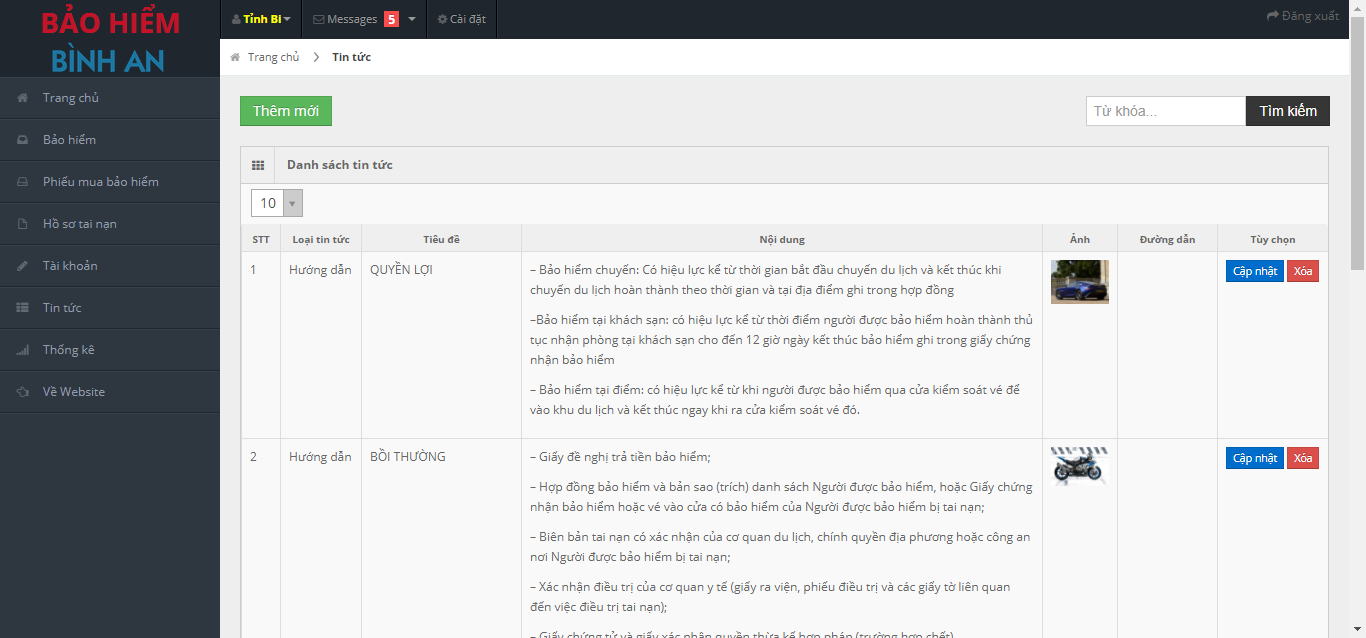
*Hình 3.39* Giao diện trang quản lý hồ sơ tai nạn.

**-** Giám định hồ sơ tai nạn



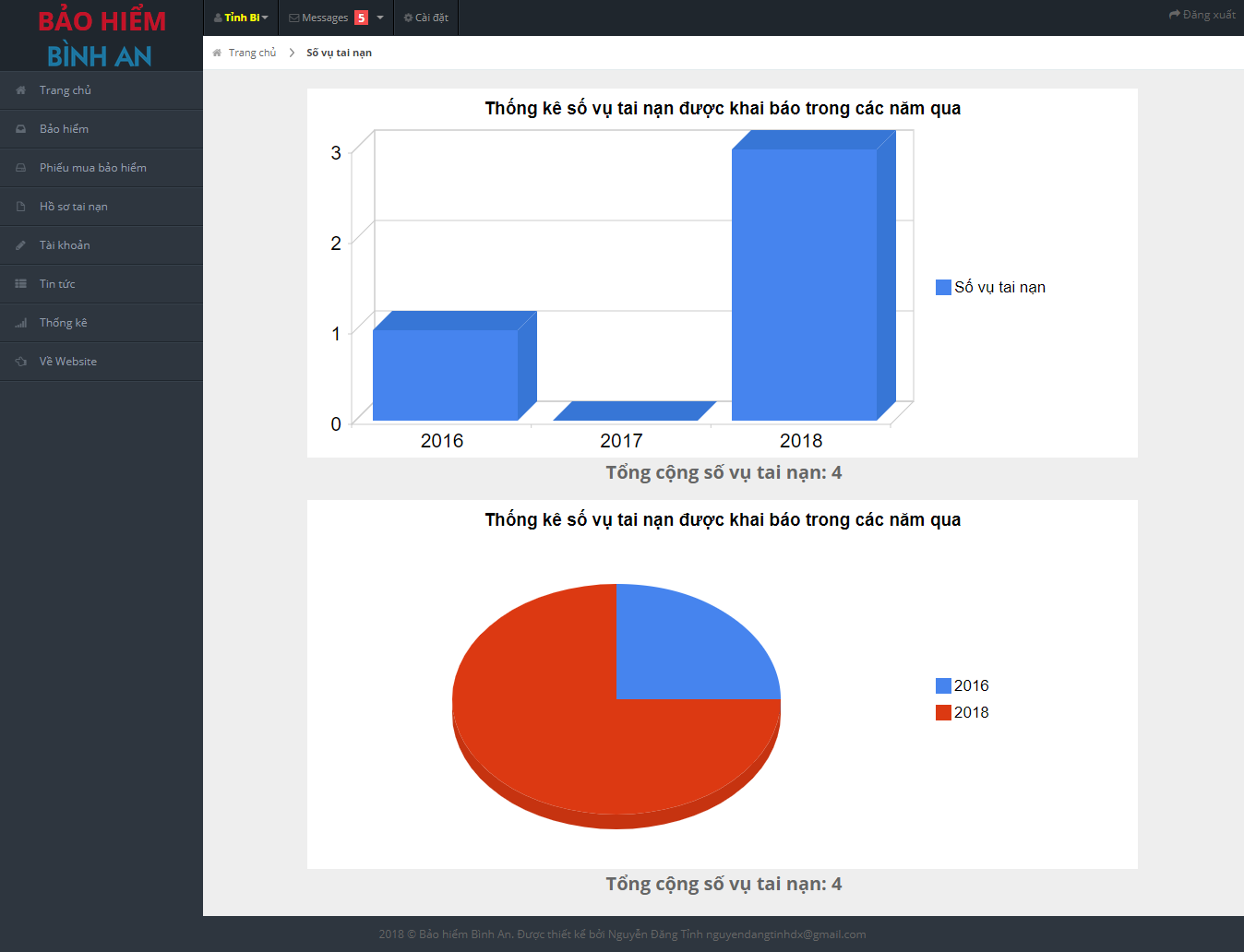
*Hình 3.40* Giao diện trang giám định hồ sơ tai nạn.

**-** Quản lý tin tức



*Hình 3.41* Giao diện trang quản lý tin tức.

**-** Thống kê số vụ tai nạn



*Hình 3.42* Giao diện trang thống kê số vụ tai nạn.

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* **Kết quả đạt được:**
* Nắm được khái quát về Framwork Spring cũng như điểm mạnh và điểm yếu của nó.
* Nắm được kiến thức cơ bản về cơ sở dữ liệu MongoDB như: Cú pháp, câu lệnh truy vấn, …
* Hiểu rõ cách thức hoạt động của một dự án làm bằng Framework Spring cũng như nắm được mô hình của nó. Biết cách cài đặt và sử dụng một số gói thư viện của Spring để phục vụ cho việc xây dựng Website.
* Xây dựng thành công một Website được viết trên nền tảng Framework Spring - MongoDB và ứng dụng tốt yêu cầu chức năng của Website.
* **Hướng phát triển:**
* Đề tài hiện tại chỉ dừng lại với tính chất cơ bản chưa nêu ra được nhiều vấn đề lớn của Spring – MongoDB. Vì vậy tiếp theo cần phát triển đề tài theo hướng một cách cụ thể hơn, chuyên sâu hơn khi đó mới đáp ứng được nhu cầu của các người dùng trong cộng đồng Việt Nam trong điều kiện đang khang hiếm tài liệu Tiếng Việt.
* Đối với Website Bảo hiểm Bình An, hiện tại Website chỉ bao gồm các chức năng cơ bản vì thế cần hợp tác với các chuyên viên bảo hiểm để xây dựng Website ngày càng phong phú hơn, mạnh mẽ hơn và đặc biệt được tăng cường bảo mật hơn.

**TÀI LIỆU THAM KHẢO**

[1] Nguyễn Văn Ba, Phát triển hệ thống hướng đối tượng với UML và C++, Nhà xuất bản Đại học Quốc gia Hà Nội, 2005.

[2] Nguyễn Mậu Hân, Giáo trình Phân tích và thiết kế hệ thống thông tin, 2004.

[3] <https://vi.wikipedia.org/wiki/Spring_Framework>

[4] <https://itphutran.com/tai-lieu-hoc-lap-trinh-spring-framework>

[5] <https://docs.mongodb.com>