TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

─────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ PHẦN MỀM

**NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG BÁN HÀNG DỰA VỊ TRÍ**

Sinh viên thực hiện : **Đặng Văn Long**

Lớp CNTT 2 - K55

Giáo viên hướng dẫn: TS.**Nguyễn Thanh Hùng**

HÀ NỘI 5-2015

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Đặng Văn Long

Điện thoại liên lạc: 0989868965 Email: dangvanlongbkhn@gmail.com

Lớp: CNTT2 - k55 Hệ đào tạo: Đại học

Đồ án tốt nghiệp được thực hiện tại:

Thời gian làm ĐATN: Từ ngày 12 / 02 / 2015 đến 30 / 05 / 2015

2. Mục đích nội dung của ĐATN

* Tìm hiểu về nền tảng xây dựng web của Google
* Xây dựng ứng dụng bán hàng dựa vị trí đa nền tảng (Android,iOS,web….)

3. Các nhiệm vụ cụ thể của ĐATN

* Tìm hiểu về framword Google Web Toolkit.
* Tìm hiểu về nền tảng Google App Engine cho Java để xây dựng ứng dụng web.
* Demo được ứng dụng bán hàng dựa vị trí trên nền tảng web và mobile.

4. Lời cam đoan của sinh viên:

Em – *Đặng Văn Long* - cam kết ĐATN là công trình nghiên cứu của bản thân em dưới sự hướng dẫn của *TS.Nguyễn Thanh Hùng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 30 tháng 5 năm 2015*  Tác giả ĐATN  *Đặng Văn Long* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng 5 năm 2015*  Giáo viên hướng dẫn  *TS.Nguyễn Thanh Hùng* |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong thời đại ngày nay, công nghệ thông tin ngày càng đóng một vai trò quan trọng trong sự phát triển của xã hội. Sự phát triển mạnh mẽ của máy tính và sự bùng nổ của các thiết bị di động đã làm ảnh hưởng to lớn đổi với mọi lĩnh vực trong cuộc sống.Trong bối cảnh nền kinh tế đang phát triển một cách mạnh mẽ thì vai trò của công nghệ thông tin là rất to lớn,nó giúp cho sự hòa nhập, giao lưu ,trao đổi..với nhau trở nên dễ dàng hơn,hiệu quả hơn.Hiểu được tầm quan trọng này,em đã quyết định lựa chọn đề tài xây dựng ứng dụng bán hàng dựa vị trí trên nền tảng web và mobile.

Để xây dựng được ứng dụng bán hàng dựa vị trí cần phải đáp ứng những yếu tố sau:

* Ứng dụng phải đáp ứng và kết nối với những người có nhu cầu mua hàng và bán hàng, một trong những yếu tố quan trọng là phải hỗ trợ ứng dụng trên tất cả các nền tảng.
* Phải đảm bảo được hiệu năng của ứng dụng : nhanh chóng, hiệu quả.

Với những yêu cầu phải giải quyết ở trên, em đã quyết định sử dụng

nền tảng của Google bào gồm: Google Web Toolkit (GWT), Google App Engine (GAE) để xây dựng ứng dụng.

Nội dung của đồ án sẽ trình bày về cách sử dụng bộ công cụ Google Web Toolkit và Google App Engine để giải quyết vấn đề và các cơ sở lý thuyết liên quan. Từ đó đưa ra mô hình thiết kế hệ thống phần mềm, chi tiết và các kết quả thu được trong quá trình cài đặt và thử nghiệm phần mềm.

Đồ án được em trình bày qua ba phần:

**Phần 1: Đặt vấn đề và định hướng giải pháp**

Đặt vấn đề, nêu lên phạm vi, mục tiêu và nhiệm vụ của đề tài.

Định hướng, phân tích và lựa chọn giải pháp để thực hiện đề tài.

Trình bày cơ sở lý thuyết về Google Web Toolkit (GWT), Google App Engine (GAE) và một số lý thuyết khác liên quan đến nội dung đồ án.

**Phần 2: Phân tích và thiết kế hệ thống phần mềm**

Chi tiết phân tích, thiết kế hệ thống.

Chi tiết thiết kế CSDL, thiết kế giao diện phần mềm.

**Phần 3: Hiệu quả trong thực tế, kiểm tra và đánh giá**

Kết quả thực hiện, cài đặt và thử nghiệm hệ thống.

**Phần 4: Kết luận và hướng phát triển**

Kết luận chung của đồ án và hướng phát triển trong tương lai.

# 

# LỜI CẢM ƠN

*Trước hết, em xin được chân thành gửi lời cảm ơn sâu sắc tới các thầy cô giáo trong trường Đại học Bách Khoa Hà Nội nói chung và các thầy cô trong Viện Công nghệ thông tin và Truyền thông, bộ môn Công nghệ phần mềm nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức và những kinh nghiệm quý báu trong suốt quá trình học tập và rèn luyện tại trường Đại học Bách Khoa Hà Nội.*

*Em xin được gửi lời cảm ơn đến TS. Nguyễn Thanh Hùng - Giảng viên bộ môn Công nghệ phần mềm, Viện Công nghệ thông tin và Truyền thông, trường Đại học Bách Khoa Hà Nội đã hết lòng giúp đỡ, tận tình hướng dẫn và chỉ dạy tận tình trong quá trình em làm đồ án tốt nghiệp.*

*Em xin gửi lời cám ơn tới Công ty an ninh mạng Bkav đã tạo điều kiện cho em có một môi trường thực tập và làm việc chuyên nghiệp trong suốt quãng thời gian 03 năm em thực tập tại đây. Xin cám ơn anh Vũ Ngọc Sơn, anh Nguyễn Công Cường cùng các bạn trong Mobile Security Team, tập thể Phòng nghiên cứu và phát triển - BRD và các anh chị cán bộ nhân viên trong Công ty An ninh mạng Bkis đã giúp đỡ em rất nhiều về mặt chuyên môn để có thể hoàn thành đồ án tốt nghiệp này.*

*Cuối cùng, em xin được gửi lời cảm ơn chân thành tới gia đình, bạn bè đã quan tâm, động viên, đóng góp ý kiến và giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành Đồ án tốt nghiệp.*

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 30 tháng 5 năm 2015*  Tác giả ĐATN  *Đặng Văn Long* |

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3

LỜI CẢM ƠN 4

DANH MỤC HÌNH ẢNH 7

DANH MỤC CÁC THUẬT NGỮ VÀ TỪ VIẾT TẮT 9

CHƯƠNG 1 - ĐẶT VẤN ĐỀ VÀ ĐINH HƯỚNG GIẢI PHÁP 10

1.1. Đặt vấn đề 10

1.2. Giới thiệu đề tài 10

1.3. Yêu cầu đặt ra đối với đề tài 11

1.4. Phân tích lựa chọn giải pháp 11

1.5. Cơ sở lý thuyết 12

1.5.1. Tìm hiểu về Google Web Toolkit 12

1.5.1.3 Hạn chế của GWT ? 14

1.5.2. Tìm hiểu về Google App Engine 21

1.5.3. GAE Datastore 24

1.5.4. GAE Channel API 24

1.5.5 Tổng quan về Google Maps API v3 26

CHƯƠNG 2 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG PHẦN MỀM 28

2.1 Kiến trúc hệ thống 28

2.2 Tổng quan về hệ thống 29

2.2.1 Mục tiêu hệ thống 29

2.1.2 Yêu cầu chung của hệ thống 29

2.2.3 Sơ đồ tổng quan của hệ thống 30

2.3. Thiết kế chương trình 31

2.3.1.Phân tích chức năng và biểu đồ UseCase của hệ thống 31

2.3.2. Biểu đồ trình tự và biểu đồ hoạt động của hệ thống 34

2.4. Thiết kế cơ sở dữ liệu 42

2.4.1. Dữ liệu thông tin tài khoản người dùng 42

2.4.2. Thông tin của khách hàng 42

2.4.4. Thông tin sản phẩm của nhà cung cấp 43

2.4.5. Thông tin ảnh 44

2.4.6 Tin nhắn 45

2.5. Thiết kế giao diện 46

2.5.1. Thiết kế giao diện màn hình đăng nhập và đăng ký 46

2.5.2. Thiết kế giao diện cho nhà cung cấp sản phẩm 47

2.5.3. Thiết kế giao diện cho khách hàng 47

2.5.4. Thiết kế giao diện hiển thị chi tiết sản phẩm 49

2.5.5. Thiết kế giao diện hiển thị map 49

CHƯƠNG 3 – TRIỂN KHAI THỬ NGHIỆM HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ 51

3.1 Cài đặt 51

3.1.1 Môi trường cài đặt 51

3.1.2 Cài đặt Google Web Toolkit (GWT) 51

3.1.3 Cài đặt Google App Engine (GAE) 52

3.2 Cài đặt giải thuật 52

3.3 Kiểm thử và đánh giá hệ thống 52

3.3.1 Kiểm thử 52

CHƯƠNG 4 - KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 55

4.1 Các kết quả đạt được 55

4.2 Hướng phát triển trong tương lai 55

TÀI LIỆU THAM KHẢO 56

# DANH MỤC HÌNH ẢNH

Hình 1 - Google Web Toolkit 12

Hình 2 - Các widgets của GWT 13

Hình 3 - Cấu trúc một ứng dụng GWT 15

Hình 4 - Tương tác không đồng bộ hóa Client-Server 19

Hình 5 - Google App Engine 21

Hình 6 - Sơ đồ tạo một kênh trên máy chủ 24

Hình 7 - Sơ đồ nhận thông tin mới 25

Hình 8 - Kiến trúc hệ thống 26

Hình 9 - Sơ đồ tổng quan 28

Hình 10 - UseCase Tổng quát 29

Hình 11 - UseCase quản lý tài khoản 30

Hình 12 - UseCase Quản lý sản phẩm 30

Hình 13 - UseCase tìm kiếm sản phẩm 31

Hình 14 - Biểu đồ trình tự chức năng đăng nhập 32

Hình 15 - Biểu đồ hoạt động chức năng đăng nhập 33

Hình 16 - Biểu đồ trình tự chức năng đăng ký 34

Hình 17 - Biểu đồ hoạt động chức năng đăng ký 34

Hình 18 - Biểu đồ trình tự chức năng quản lý sản phẩm 35

Hình 19 - Biểu đồ hoạt động chức năng quản lý sản phẩm 36

Hình 20 - Biểu đồ trình tự chức năng Tìm kiếm sản phẩm 37

Hình 21 - Biểu đồ hoạt động chức năng tìm kiếm sản phẩm 38

Hình 22 - Biểu đồ hoạt động quá trình mua hàng 39

Hình 23 - Giao diện đăng nhập và đăng ký 44

Hình 24 - Giao diện nhà cung cấp 45

Hình 25 - Giao diện khách hàng 46

Hình 26 - Giao diện xem chi tiết sản phẩm 47

Hình 27 - Giao diện hiển thị map 48

Hình 28 : Màn hình cài đặt eclipse 50

**DANH MỤC BẢNG BIỂU**

Bảng 1 - Thư mục src 18

Bảng 2 - Thư mục war 19

Bảng 3 - Thư mục test 20

Bảng 6 - Mô tả Quản lý sản phẩm 31

Bảng 7 - Mô tả Tìm kiếm sản phẩm 32

Bảng 11 - Thông tin ảnh lưu trữ 42

Bảng 12 - Tin nhắn 43

Bảng 13 - Kiểm thử giao diện 51

Bảng 14 - Tình huống kiểm thử 52

# DANH MỤC CÁC THUẬT NGỮ VÀ TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Thuật ngữ** | **Ý nghĩa** |
| GWT | Google Web Tollkit |
| GAE | Google App Engine |
| PaaS | Platform as a Service |
| GUI | Graphical user interface |
| DOM | Document Object Model |
| RPC | Remote procedure call |
| API | Application programming interface |
| NCC | Nhà cung cấp |
| SP | Sản phẩm |

# CHƯƠNG 1 - ĐẶT VẤN ĐỀ VÀ ĐINH HƯỚNG GIẢI PHÁP

Trong phần này em sẽ trình bày nội dung, phạm vi và mục tiêu của đề tài mà em thực hiện trong đồ án, định hướng, phân tích và lựa chọn giải pháp thực hiện đề tài cùng với đó là trình bày một số lý thuyết liên quan được đề cập đến trong đề tài.

## ****1.1. Đặt vấn đề****

Ở bất cứ giai đoạn này trong lịch sử, thì sự phát triển về kinh tế là luôn đóng vai trò rất quan trọng. Một triều đại muốn được vững mạnh, tồn tại lâu dài thì phải phát triển kinh tế một cách đa dạng, từ đó sẽ cải thiện được cuộc sống của nhân dân.

Từ thời xa xưa,ngoài việc phát triển kinh tế trong nước thì ông cha ta cũng đã tìm cách giao lưu, buôn bán, trao đổi hàng hóa với các nước bên ngoài bằng nhiều hình thức khác nhau như : qua đương sông, đường bộ…

Trong thời đại công nghệ thông tin phát triển như vũ bão, chỉ trong khoảng thời gian ngắn ngủi vài thập kỷ, công nghệ thông tin đã thay đổi hoàn toàn mọi mặt thế giới.Những bước nhảy vọt về công nghệ đã thúc đầy, làm động lực cho cuộc cách mạng ở hầu hết các lĩnh vực khoa học, kỹ thuật và đời sống. Ngày này, khi mà máy tính đã xâm nhập và đóng vai trò là một yếu tố quan trọng trong tất cả mọi lĩnh vực, thì những chiếc điện thoại thông minh đang phát triển rộng khắp và trở thành một phần thiết yếu không thể thiếu trong cuộc sống thường ngày. Đặc biệt hơn cả chính những sự thay đổi này đã làm cho nền kinh tế bước sang một kỉ nguyên mới.

Chỉ với một chiếc máy tính, hay một chiếc điện thoại thông minh cùng với vài thao tác cơ bản, chúng ta có thể tìm thấy được những thứ mình cần từ các trang web bán hàng trực tuyến, hay là từ những ứng dụng điện thoại trên Android,iOS…mà không cần phải đến tận cửa hàng,tiết kiệm chi phi thời gian.

Nắm được thiết yếu từ thực tế như vậy,em đã quyết định xây dựng ứng dụng bán hàng dựa vị trí để có thể kết nối được nhu cầu của mọi người,đông thời tiết kiệm tối đa chi phí đi lại.

## ****1.2. Giới thiệu đề tài****

Qua những thông tin vừa được đề cập ở phần trên, vì thế em đã quyết định lựa chọn đề tài “Xây dựng ứng dụng bán hàng dựa vị trí trên nền tảng web và mobile”, cụ thể hơn là xây dựng một phần mềm với chức năng kết nối khách hàng và nhà cung cấp các sản phẩm với nhau.Khách hàng có yêu cầu lựa chọn mặt hàng cần mua trong danh sách các mặt hàng ,cùng với những lựa chọn khác như : bán kính, khoảng giá tiền ,độ tin cậy đối với nhà cung cấp…từ đó sẽ hiển thị những nhà cung cấp thỏa mãn yêu cầu đó.Khách hàng có thể lựa chọn nhà cung cấp cố định hoặc nhà cung cấp di dộng.Nếu khách hàng chọn nhà cung cấp cố định thì sẽ hiển thị hướng dẫn chỉ đường đến, còn lựa chọn nhà cung cấp di động thì phải hiển thị được real-time vị trí của nhà cung cấp đang di chuyển cho khách hàng . Đồng thời khách hàng có thể đánh giá cho nhà cung cấp.

## 1.3. Yêu cầu đặt ra đối với đề tài

Phần mềm được xây dựng phải đạt được các yêu cầu sau:

* Khả năng kết nối và đáp ứng với nhau một cách hiệu quả và nhanh chóng, đồng thời phải tối ưu được hệ thống khi mà tài nguyên lớn.
* Phần mềm khi được xây dựng cần đảm bảo tính mềm dẻo, dễ dàng khi mở rộng và cài đặt thêm các module mới.
* Phần mềm hoạt động chính xác, ổn định, không làm ảnh hưởng đến tốc độ, chức năng của các phần mềm khác hay là hiệu năng sử dụng của thiết bị.
* Phần mềm có thể chạy được trên nền tảng web lẫn mobile.

## 1.4. Phân tích lựa chọn giải pháp

Về mặt phát triển, ứng dụng bán hàng dựa vị trí để có thể đến được tay người dùng một cách rộng rãi thì cần phải phát triển một cách lâu dài, cần một đội ngũ hỗ trợ chuyên nghiệp. Phân tích sâu hơn những yêu cầu cụ thể đối với ứng dụng thì em nhận thấy cần phải giải quyết được những vấn đề sau đây:

* Dựa trên thực tế, thì ứng dụng bán hàng dựa vị trí sẽ được người dùng sử dụng chủ yếu trên mobile (Android,iOS,Windown phone ….) đồng thời cũng hỗ trợ cho người dùng ở trên web.
* Với một dữ liệu lớn thì phải xử lý được tốt về cơ sở dữ liệu cũng là một vấn đề lớn.
* Khả năng xử lý khi có người người dùng yêu cầu cùng một lúc đồng thời hệ thống phải xử lý đáp ứng thật nhanh.
* Hiển thị được vị trí của người sử dụng trên bản đồ,đồng thời hiển thị được vị trí tương ứng khi di chuyển.

Để hoàn thành được những đáp ứng trên, tiết kiệm được chi phí thời gian, tận dụng được tối đa được những kiến thức về lập trình hướng đối tượng cộng với sự hướng dẫn, định hướng công nghệ của thầy Nguyễn Thanh Hùng thì giải pháp được đưa ra là sử dụng bộ công cụ phát triển của Google cung cấp đó là: Google Web Toolkit (GWT) và Google App Engine (GAE) điều này sẽ làm cho việc phát triển ứng dụng trở nên dễ dàng hơn.

Thật vậy, GWT giúp biên dịch từ mã Java thành mã Javascript phía máy khách. Nó hỗ trợ một tập con các gói java.lang và java.util, cùng với các API được chính GWT cung cấp. Thay vì sử dụng phân tách thông thường của các mã cơ sở phía máy khách và máy chủ, GWT cung cấp một Java API cho phép tạo các giao diện đồ họa người dùng dựa trên các thành phần, rồi biên dịch chúng để hiển thị trong trình duyệt Web của người dùng. Và kết quả cuối cùng một ứng dụng GWT sau khi biên dịch thì chỉ còn HTML, XML và Javascript. Nhờ việc biên dịch thành Javascript như vậy, GWT sẽ giúp người phát triển xây dựng được các ứng dụng chạy trên bất cứ thiết bị nào mà có hỗ trợ javascript. Mà hầu hết các thiết bị di động thông minh hiện nay đều hỗ trợ javascript. Vậy vì sao không sử dụng GWT để xây dựng một ứng dụng có thể chạy trên cả máy tính, hệ điều hành iOS, hệ điều hành Android…Do đã có chút ít kinh nghiệm về lập trình Java, người viết nhận thấy sử dụng GWT để giải quyết bài toán của mình là một lựa chọn tối ưu.

GWT sẽ đảm nhận phía client còn bộ đôi với nó GAE sẽ đảm nhận phía server. Sử dụng GAE phía server cũng là một lợi thế lớn so với cách xây dựng web truyền thống. Bởi GAE là công nghệ điện toán đám mây, nó giúp cho việc cài đặt quản lý và lưu trữ dữ liệu lớn trở nên đơn giản hơn vì tài nguyên này là của Google. Đồng thời, việc tích hợp, sử dụng các dịch vụ của Google như BigTable DataStore, Google search, images, translation… cũng trở nên dễ dàng. Với bài toán đặt ra là hỗ trợ giáo dục trực tuyến thì cơ sở dữ liệu có thể sẽ tương đối lớn, như vậy GAE sẽ hỗ trợ rất mạnh.

Nhìn vào các ưu điểm trên, thì việc sử dụng công nghệ GWT và GAE để giải quyết bài toán đặt ra là hoàn toàn hợp lý.

## 1.5. Cơ sở lý thuyết

Trong phần này sẽ trình bày các cơ sở lý thuyết liên quan đến bộ công cụ của Google là Google Web Toolkit(GWT), Google App Engine (GAE) và một số vấn đề lý thuyết khác liên quan đến nội dung của đồ án.

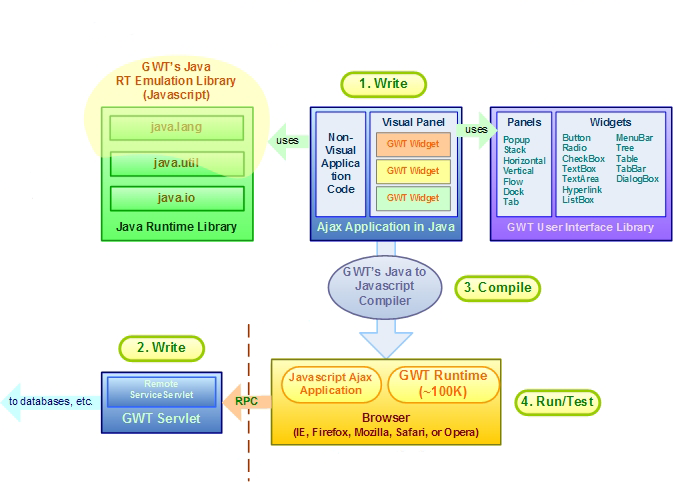
### 1.5.1. Tìm hiểu về Google Web Toolkit

#### 1.5.1.1. Giới thiệu chung về Google Web Toolkit

GWT là một framework mã nguồn mở dùng để phát triển web bằng ngôn ngữ Java được bắt đầu và quản lý bởi Google. Đã có nhiều sản phẩm của Google được xây dựng bằng framework này như Google Wave và phiên bản mới của Google Adwords.

GWT là một giải pháp khá toàn diện cho lập trình viên Java để xây dựng các ứng dụng web 2.0 mà không cần phải biết quá nhiều ngôn ngữ, đồng thời lại có thể gỡ lỗi trong IDE như khi gỡ lỗi một ứng dụng viết bằng Java thông thường, thiết lập các breakpoint, trong khi cùng lúc đó lại có thể truy cập trình duyệt vì đã được biên dịch thành Javascript - một lợi thế rất lớn vì bản thân Javascript không có chế độ debug, khiến cho rất khó khăn để xác định được vị trí lỗi trong đống code JS, ...

Trong lập trình web truyền thống, người dùng rất nhiều khi phải viết các đoạn mã nguồn khác nhau cho các trình duyệt khác nhau. Trong khi GWT sẽ làm thay việc làm việc với từng trình duyệt một cho lập trình viên bằng cách sử dụng DOM package, tức là lập trình viên sẽ chỉ tương tác với DOM package của GWT thay vì phải trực tiếp tương tác với tất cả các trình duyệt. Các lớp Widget của DOM pakage sẽ giúp cho việc xây dựng giao diện dễ dàng hơn và sẽ làm việc chính xác trên mọi trình duyệt.



Hình 1 - Google Web Toolkit

Phần tiếp theo người viết trình bày về ưu điểm của GWT mang lại để hiểu vì sao người viết lại sử dụng GWT vào đồ án của mình.

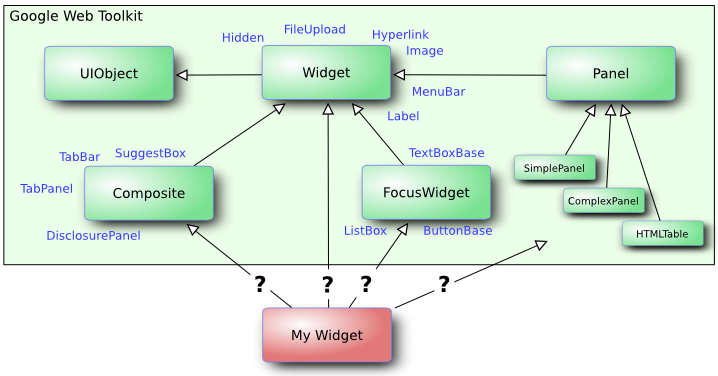
#### 1.5.1.2. Ưu điểm của GWT ?

GWT mang lại cho nhà phát triển nhiều lợi ích : Lập trình bằng java, sử dụng công nghệ Ajax, hỗ trợ nhiều gói GUIs khác nhau cho các nền tảng như Web, Android, iOS và ưu điểm nữa. Các ưu điểm ấy sẽ được trình bày cụ thể phía dưới đây.

Thứ nhất, GWT sử dụng ngôn ngữ Java để lập trình, điều này là một thế mạnh bởi vì Java là ngôn ngữ lập trình hướng đối tượng được ưa chuộng nhất hiện nay. Vì vậy sử dụng GWT để tạo ứng dụng web sẽ trở nên đơn giản như tạo ứng dụng window. Không những thế việc GWT chỉ sử dụng một ngôn ngữ lập trình duy nhất để viết ứng dụng web cho cả Client và Server, giúp cho việc lập trình trở nên dễ dàng hơn, không phải nhớ quá nhiều về mã nguồn và kỹ năng lập trình đối với từng ngôn ngữ.

Thứ hai, sử dụng công nghệ AJAX (Asynchronous Javascript and XML có nghĩa là Javascript và XML không đồng bộ) có thể đọc là "trao quyền cho javascript" và thông qua js để cung cấp một công nghệ phía client để gọi ngầm một lệnh background để phía server thực hiện và nhận thông tin trả về, update thông tin của trang nhanh mà không cần phải load lại cả trang, điều này rất hữu ích bởi việc load lại những cái không cần thiết rất mất thời gian. GWT sử dụng AJAX để trao đổi dữ giúp lập trình viên thoát khỏi ma trận các công nghệ khi viết các ứng dụng AJAX quá phức tạp khó khăn và gặp nhiều lỗi. Với GWT, bạn có thể phát triển và kiểm tra lỗi các ứng dụng AJAX bằng ngôn ngữ Java, sử dụng các công cụ phát triển Java tuỳ theo ý thích.

Thứ ba, GWT là một framework mã nguồn mở hỗ trợ nhiều gói GUIs khác nhau cho các nền tảng như Web, Android, iOS bằng cách cung cấp nhiều thư viện, đặt biệt nó cung cấp một thư viện các widget, panel giúp việc xử lý sự kiện như control Windows và thiết kế layout với các quy định chặt chẽ.

 Hình 2 - Các widgets của GWT

Cuối cùng, khi bạn triển khai ứng dụng của mình, bộ biên dịch của GWT sẽ dịch ứng dụng Java của bạn sang Javascript và HTML. Kết quả cuối cùng trả về cho client hoàn toàn là Javascript và HTML. Với các nhà phát triển web truyền thống, sẽ phải cấu hình khác nhau cho mỗi trình duyệt, nhưng với nhà phát triển GWT thì không phải cấu hình bất cứ trường hợp riêng nào. Không những thế, việc trả về HTML và Javascrip sẽ không yêu cầu cấu hình môi trường phía client đây là điểm mấu chốt để sử dụng GWT tạo ứng dụng cho cả Web và mobile.

Có nhiều ưu điểm như vậy nhưng nhiều nhà phát triển chưa dùng GWT bởi vì các hạn chế của nó.

### 1.5.1.3 Hạn chế của GWT ?

GWT mang rất nhiều ưu điểm như đã nói ở trên, nhưng tồn tại song song với những ưu điểm ấy là những hạn chế sau :

* Các mã HTML và Javascript mà GWT tạo ra khá nặng.
* Các nhà phát triển Java khi triển khai Tag Ajax chỉ mất vài phút nhưng với GWT thì mất thời gian khá lâu?????
* Không lập chỉ mục tìm kiếm: các trang web được tạo ra bởi GWT sẽ không được lập chỉ mục của công cụ tìm kiếm vì các nội dung được tạo ra tự động.
* GWT chưa quen thuộc với các nhà thiết kế web cổ điển khi họ muốn thay đổi ở html, css.

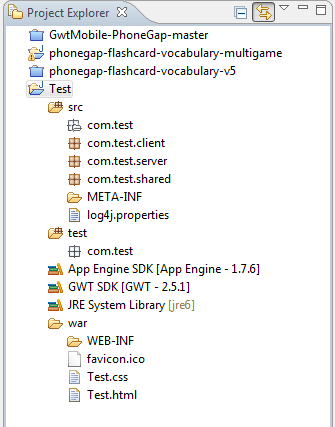
**1.5.1.4 Các thành phần của GWT**

* Bộ biên dịch Java-to-JavaScript: Dùng để dịch tất cả các mã nguồn Java của ứng dụng GWT sang code JavaScript và HTML.
* Trình duyệt dành cho Hosted Mode: Dùng để các lập trình viên chạy ứng dụng GWT bằng hosted mode.
* Thư viện JRE (Java Runtime Environment): Chỉ có 2 thư viện chuẩn của Java được sử dụng trong GWT là java.lang và java.util.

Các thư viện của GWT để thiết kế giao diện: Là các thư viện chứa các lớp được tạo sẵn (API) hoặc do người dùng tự tạo, dùng để tạo các đối tượng AJAX bằng Java trong các ứng dụng GWT. Hay còn được gọi là các widget.

**1.5.1.5 Cấu trúc của một GWT Application**

Khi một project được khởi tạo, ngoài trang chủ html, các file config nói sẽ chứa 3 thư mục con như sau: Thư mục src, thư mục war, thư mục test.



Hình 3 - Cấu trúc một ứng dụng GWT

**1.5.1.5.1 Trang chủ HTML**

Các mô đun GWT được lưu trữ trên một máy chủ web như một tập hợp các tập tin JavaScript và liên quan. Để chạy các mô đun đó, nó phải được tải từ một trang web từ phía client. Mọi trang HTML có chứa một ứng dụng GWT thông qua một thẻ SCRIPT. Trang HTML này được gọi là một trang chủ HTML. Một trang chủ HTML điển hình cho một ứng dụng được viết với GWT từ đầu có thể không bao gồm bất kỳ nội dung có thể nhìn thấy ở thẻ body.

Nhưng GWT được thiết kế để làm cho nó dễ dàng thêm chức năng GWT cho các ứng dụng web hiện tại với chỉ thay đổi nhỏ. Nó có thể cho phép các mô-đun GWT có chọn lọc chèn các widget vào địa điểm cụ thể trong một trang HTML. Để thực hiện điều này, sử dụng thuộc tính id trong thẻ HTML của bạn để xác định một định danh duy nhất là mã GWT của bạn sẽ sử dụng để đính kèm các vật dụng với phần tử HTML. Chú ý rằng các thẻ td bao gồm một thuộc tính id liên kết với chúng. Thuộc tính này có thể truy cập thông qua các lớp DOM. Bạn có thể dễ dàng đính kèm các vật dụng bằng cách sử dụng phương pháp RootPanel.get ().

Theo cách này, chức năng GWT có thể được thêm vào như là chỉ là một phần của trang đã có, và thay đổi cách bố trí ứng dụng có thể được thực hiện trong đơn giản HTML. Một trang HTML máy chủ không phải là nội dung tĩnh. Nó cũng có thể được tạo ra bởi một servlet, hoặc một trang JSP.

**1.5.1.5.2 Thư mục src**

Trong thư mục src sẽ chứa 3 gói: client, server và share:

|  |  |  |
| --- | --- | --- |
| **Gói** | **File** | **Mục đích** |
| com.test |  | Gói gốc của dự án chứa các file module XML. |
| com.test | Test.gwt.xml | Các module trong ứng dụng. Kế thừa từ com.google.gwt.user.User và thêm một lớp bắt đầu của hệ thống com.client.Test |
| com.test |  | Tài nguyên tĩnh được tải theo chương trình bằng mã GWT. Tập tin trong thư mục public sẽ được sao chép vào thư mục tương tự như các trình biên dịch ra GWT. |
| com.test | logo.gif | Một tập tin ảnh có sẵn cho các mã ứng dụng. Bạn có thể tải tập tin này lập trình sử dụng URL này: GWT.getModuleBaseURL()+"logo.gif" |
| com.test.client |  | Các tệp tin mã nguồn phía client và các thư mục con. |
| com.test.client | Test.java | Mã nguồn Java phía client dùng làm điểm bắt đầu. |
| com.test.client | TestService.java | Giao diện RPC service. |

Bảng 1 - Thư mục src

**1.5.1.5.3 Thư mục war**

Thư mục war là hình ảnh khai triển trang web, các tiêu chuẩn định dạng mở rộng war được công nhận bởi nhiều máy chủ java, Tomcat, Jetty, và chứa servlet J2EE. Nó chứa các tài nguyên:

* Nội dung tĩnh mà người phát triển cung cấp, ví dụ như trang chủ HTML.
* Đầu ra GWT đã được biên dịch.
* Các lớp java và file jar cho mã nguồn phí server.
* Một file web.xml để cấu hình ứng dụng web và các servlet.

|  |  |  |
| --- | --- | --- |
| **Thư mục** | **File** | **Mục đích** |
| Test/war/ | test.html | Trang chủ HTML tải ứng dụng Test về. |
| Test/war/ | test.css | Trang tạo kiểu tĩnh cho ứng dụng Test. |
| Test/war/Test/ |  | Thư mục chứa các module của ứng dụng Test nơi mà trình biên dịch GWT đã viết ra và các tập tin public thì đã được sao chép.  Chú ý: Theo mặc định thư mục này sẽ dài, đủ các điều tiện module  namecom.google.gwt.sample.test.Test.  Tuy nhiên, khi module XML của GWT sử dụng thì sẽ đổi tên thành Test để rút ngắn thành tên đẹp. |
| /war/Test/ | Test.nocache.js | Các " lựa chọn script " cho ứng dụng Test. Đây là script mà phải được nạp từ trang chủ HTML nạp đến module GWT vào trong trang. |
| /war/WEB-INF |  | Tất cả các mã nguồn ko public thì được đặt ở đây, hãy nhìn vào đặc tả servlet để xem chi tiết hơn. |
| /war/WEB-INF | web.xml | Cấu hình ứng dụng web của bạn và bất kỳ servlet nào. |
| /war/WEB-INF | web.xml | Cấu hình ứng dụng web của bạn và bất kỳ servlet nào. |
| /war/WEB-NF/classes |  | Các tập tin lớp Java đã biên dịch đặt ở đây để thực thi chức năng phí máy chủ. Nếu bạn sử dụng một IDE để thiết lập thư mục đầu ra thi là thư mục này. |
| /war/WEB-INF/lib |  | Các thư viện phụ thuộc và mã nguồn server của bạn cần được đặt ở đây. |
| /war/WEB-INF/lib | gwt-servlet.jar | Nếu bạn có bất kỳ servlet sử dụng GWT RPC , bạn sẽ cần phải đặt một bản sao của GWT- servlet.jar tại đây. |

Bảng 2 - Thư mục war

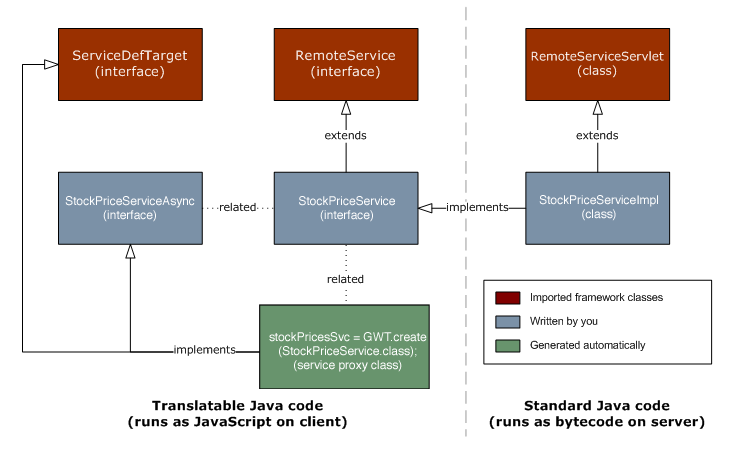
**1.5.1.5.4 Thư mục test**

Thư mục test chứa các file mã nguồn cho Junit tests.

|  |  |  |
| --- | --- | --- |
| **Gói** | **File** | **Mục đích** |
| sample.Test.client |  | Các tệp tin kiểm thử phía client và các thư mục con |
| sample.Test.client | TestTest.java | Test case cho lớp điểm vào (entry-point). |
| sample.Test.server |  | Các tệp tin kiểm thử phía server và các thư mục con. |
| sample.Test.server | SchoolCalendarServiceImplTest.java | Test case cho lớp server. |

Bảng 3 - Thư mục test

**1.5.1.6 Tương tác không đồng bộ giữa Client – Server**



Hình 4 - Tương tác không đồng bộ hóa Client-Server

Mọi ứng dụng của GWT đều chạy với mã nguồn Javascript trong trình duyệt web của người dùng cuối. Thường xuyên, các nhà phát triển muốn tạo ra các ứng dụng độc lập phía client, những ứng dụng ấy cần phải giao tiếp với máy chủ web, gửi yêu cầu và nhận được cập nhật. Mỗi lần các ứng dụng web truyển thống giao tiếp với máy chủ thì nó lấy được một trang HTML hoàn toàn mới. Ngược lại, các ứng dụng Ajax giảm tải giao diện logic của người dùng tới client và thực hiện cuộc gọi không đồng bộ tới máy chủ để chỉ gửi và nhận những dữ liệu nó cần. Điều này làm cho giao diện người dùng của ứng dụng Ajax nhiều hơn, trong khi đó lại giảm yêu cầu băng thông và tải trọng của máy chủ. GWT đưa ra 3 phương thức giao tiếp giữa client với server như sau:

* Cách thứ nhất là Making Remote Procedure Calls (GWT RPC): GWT RPC là một cơ chế để thông qua các đối tượng Java đến và đi từ một máy chủ trên tiêu chuẩn HTTP. Bạn có thể sử dụng GWT RPC đểthực hiện cuộc gọi đến Java servlets và để cho GWT chú trọng đến chi tiết mức thấp như đối tượng tuần tự.
* Cách thứ hai là nhận lại dữ liệu JSON thông qua HTTP: nếu các giao tiếp giữa ứng dụng của bạn với máy chủ mà không lưu được Java Servlet, hoặc đã sử dụng một định dạng mã JSON hay XML, bạn có thể thực hiện yêu cầu HTTP để lấy lại dữ liệu. GWT cung cấp các lớp HTTP chung để bạn có thể sử dụng để lấy các yêu cầu.
* Cách cuối cùng là thực hiện yêu cầu Cross-Site cho JSONP: nếu bạn đang xây dựng ứng dụng web lai(đa mã nguồn) mà cần sử dụng một một hoặc nhiều máy chủ web từ xa bạn phải làm việc xung quanh SOP (Chính sách cùng xuất xứ) hạn chế truy cập, bạn sẽ sử dụng JavaScript Native Interface (JSNI) để viết một thẻ script để lấy JSON với lớp đệm (JSONP).

Cho dù người phát triển sử dụng phương thức nào trong 3 cách trên thì các cuộc gọi thực hiện từ trang HTML đến server đều là không đồng bộ. Điều này có nghĩa là nó không chặn trong khi chờ phản hồi, các mã theo sau cuộc gọi thực hiện ngay lập tức. Khi cuộc gọi được thiết lập, phương pháp callback được chỉ định sẽ chạy khi cuộc gọi thực hiện.

Các cuộc gọi không đồng bộ là một nguyên tắc cốt lõi của phát triển AJAX (Asynchronous JavaScript and XML). Lợi ích của việc thực hiện cuộc gọi không đồng bộ(cho các nhà phát triển) không phải là đơn giản như các cuộc gọi đồng bộ mà trong các trải nghiệm của người dùng cuối đã được cải thiện:

* Giao diện người dùng vẫn đáp ứng: Các công cụ JavaScript trong trình duyệt web nói chung là đơn luồng, cho nên gọi một máy chủ đồng bộ làm cho trang web để "treo" cho đến khi cuộc gọi hoàn tất. Nếu mạng chậm hoặc máy chủ không đáp ứng, điều này có thể làm hỏng các trải nghiệm của người dùng cuối.
* Bạn có thể thực hiện các công việc khác trong khi chờ đợi vào một cuộc gọi máy chủ cấp phát chính thức: bạn có thể xây dựng giao diện người dùng của bạn trong khi đồng thời lấy các dữ liệu từ máy chủ để đưa vào giao diện. Điều này rút ngắn thời gian tổng thể cần cho người dùng để xem dữ liệu trên trang.
* Bạn có thể thực hiện cuộc gọi nhiều máy chủ cùng một lúc.Tuy nhiên, số lượng các cuộc gọi không bộ bị giới hạn bởi vì trình duyệt thường hạn chế số lượng kết nối mạng bên ngoài tại một thời điểm.

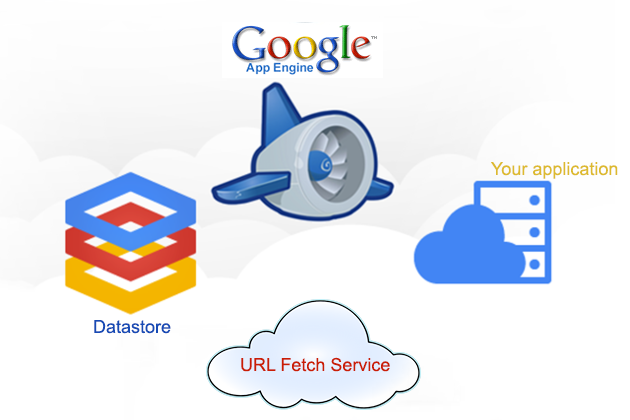
### 1.5.2. Tìm hiểu về Google App Engine

**1.5.2.1. Tổng quan về GAE**

GAE là một nền tảng cho phát triển và lưu trữ ứng dụng web trên trung tâm quản lý dữ liệu của Google. GAE là công nghệ điện toán đám mây, nó ảo hóa ứng dụng trên nhiều server. Nó tương tự các nền tảng khác như Amazon Web services hay Azure Services Platform.

Hiện tại GAE hỗ trợ hai ngôn ngữ là Python và Java. Google App Engine mang tính linh hoạt rất cao, nó cũng khá thoải mái cho các nhà phát triển có thể thử nghiệm và tiếp cận với nền tảng. Khi bắt đầu, GAE cung cấp 500 MB lưu trữ dữ liệu cho mỗi ứng dụng và bandwidth có thể quản lý xấp xỉ 5 triệu lượt truy cập/tháng.

GAE cung cấp rất nhiều cơ sở hạ tầng cho việc dễ dàng mở rộng ứng dụng, nhưng nó chỉ duy nhất chạy trong giới hạn thiết kế trong cơ sở hạ tầng của nó.



Hình 5 - Google App Engine

GAE có các đặc điểm sau :

* Tự động mở rộng và cân bằng tải
* Sử dụng datastore để lưu trữ dữ liệu, và Gql để truy vấn. DataStore không phải là cơ sở dữ liệu quan hệ.
* Việc thanh toán dựa trên số lượng đã sử dụng (gọi là quota), ứng dụng được cho một mức quota miễn phí, tức là ứng dụng được sử dụng miễn phí đến một mức nào đó,sau khi hết quota sẽ phải thanh toán. Việc thanh toán được đăng ký mức tối đa trong một tuần, và có thể thay đổi mức đăng ký cho phù hợp.
* Lưu trữ trên GAE khá uyển chuyển, nếu lưu trữ dung lượng lớn có thể sử dụng việc lưu trữ theo khối - blobstore, tuy nhiên phải kích hoạt thanh toán để có thể sử dụng dạng lưu trữ này.

#### 1.5.2.2. Ưu điểm GAE

GAE là công nghệ điện toán đám mây thể hiện rất nhiều tiện ích rõ ràng cho người phát triển, chẳng hạn như server riêng của bạn không bị hút băng thông, trong khi tốc độ truy cập của khách thì nhanh, điều này chắc chắn trang web sử dụng GAE của bạn sẽ được đánh giá cao.

Sau khi trang web của bạn được thiết lập, việc cập nhật tuyệt vời như một làn gió. Mỗi khi bạn cập nhật một tệp tin (hoặc một loạt tệp tin) bạn chỉ cần chạy câu lệnh appcfg.py để xem các thay đổi xuất hiện trên các web site của bạn.Như diễn giả Joel Spolsky nói một web site tốt là website có thể cập nhật trong một bước duy nhất, vậy GAE đã giúp thỏa mãn được yêu cầu đó.

GAE cho phép nhiều người làm việc trên một ứng dụng. Nếu bạn muốn một người bạn của bạn có thể tải lên các tệp tin vào trang web của mình, bạn không cần đưa cho anh ấy tên người dùng và mật khẩu- tất cả anh ấy cần là tài khoản Google của mình. Và tất nhiên bạn sẽ quản lý được những cập nhật ấy, bạn có thể biết ai cập nhật, cập nhật khi nào bởi App Engine cung cấp cho bạn một bản ghi tất cả các hành động của cộng tác viên.

GAE cung cấp 500MB miễn phí cho một ứng dụng lưu trữ dữ liệu, đây là một lượng khá lớn, hãy thử tính toán xem nếu bạn phải trả phí cho lượng dữ liệu này thì mỗi tháng là bao nhiêu tiền, và mỗi năm là bao nhiêu, sẽ tốn một khoản như thế nào cho cả vòng đời của phần mềm?

#### 1.5.2.3. Nhược điểm GAE

Bên cạnh những ưu điểm nêu trên thì GAE cũng có những nhược điểm khiến các nhà phát triển băn khoăn khi sử dụng. Trước hết, các nhà phát triển sẽ đặt câu hỏi Google có dám đảm bảo dịch vụ trọng mọi hình thức hay không? Google thi thoảng thay đổi thuật toán và cơ chế của mình một cách bí mật và không rõ ràng. Tất cả mọi người trong lĩnh vực tối ưu hóa công cụ cụ tìm kiếm đều biết điều này. Khi Goolgle điều chỉnh thuật toán xếp hạng của họ thì đột nhiên trang web của bạn xuất hiện xuống dưới mà thậm trí còn không lên trang đầu tiên nữa. Nếu điều đó xảy ra với GAE thì sao?

Việc lưu trữ cơ sở dữ liệu ở GAE đối với nhiều người là thuận lợi nhưng đối với những người phát triển các ứng dụng ở mức bảo mật cao thì đây không phải là một lợi thế. Mặt trái của việc xây dựng ứng dụng trên GAE là bạn sẽ phụ thuộc hoàn toàn vào các công nghệ của Google và rất khó có thể tách ra thành một ứng dụng độc lập. Yahoo hay Microsoft sẽ chẳng bao giờ mua một ứng dụng xây dựng trên nền tảng của đối thủ. Còn các nhà đầu tư cũng rất e ngại khi tài sản của công ty bạn đặt hết vào tay người khác, dù cho đó là Google.

Tuy còn một số hạn chế như vậy, nhưng để thực hiện đồ án này những nhược điểm của GAE không bị ảnh hưởng nhiều vì nó mang lại ưu điểm rất lớn. Ưu điểm cần đề bạt tới đầu tiên là việc lưu trữ cơ sở dữ liệu vào Datastore.

### 1.5.3. GAE Datastore

Từ trước tới nay việc lưu trữ cơ sở dữ liệu luôn là một bài toán đau đầu cho các nhà phát triển. Với các công ty lớn, có kinh phí đầu tư thì sẽ có riêng một máy chủ phục vụ, nhưng với những công ty nhỏ thì server sẽ phải đi thuê, và phải trả tiền hàng tháng. Từ ngày điện toán đám mây phát triển, Google cung cấp kho lưu trữ Datastore là một giải pháp khá toàn diện. Việc lưu trữ dữ liệu lớn trở nên dễ dàng hơn. Thật vậy:

GAE DataStore được dụa trên công nghệ Google Big Table

(<http://labs.google.com/papers/bigtable.html>) đó là kho lưu trữ phân tán, liên tục, được thiết kế đặc biệt để phân phối, mở rộng, phát triển. DataStore không phải là một hệ quản trị cơ sở dữ liệu quan hệ, thay vào đó, nó là nơi lưu trữ dữ liệu có giá trị, tổ chức thực thể, mỗi tổ chức thực thể được chứa trong một bộ dữ liệu có giá trị.

GAE DataStore được thiết kế sao cho thời gian yêu cầu tuyến tính với kết quả, chứ không phụ thuộc vào kích thước dữ kiệu được lưu trữ. Điều này một phần được thực hiện bằng cách DataStore xây dựng chỉ số tại thời gian ghi, và các ràng buộc hạn chế của GAE trong các loại hỗ trợ truy vấn, kết quả là cực kì hiệu quả khi đọc các bộ dữ liệu lớn.

GAE DataStore hộ trợ truy vấn và sử dụng điều khiển tương tranh một cách khả quan. Các mô hình truy vấn thì cụ thể cho App Engine làm việc: Các truy vấn được chấp nhận thì được xác định bằng cách định nghĩa và nhóm lại trên GAE DataStore.

Sau ưu điểm về việc lưu trữ ở Datastore là phải nhắc đến việc GAE cung cấp các API hữu ích, tiện dụng. GAE Channel API là một trong những API rất tiện dụng.Dưới đây em xin trình bài chi tiết về GAE Channel API

### 1.5.4. GAE Channel API

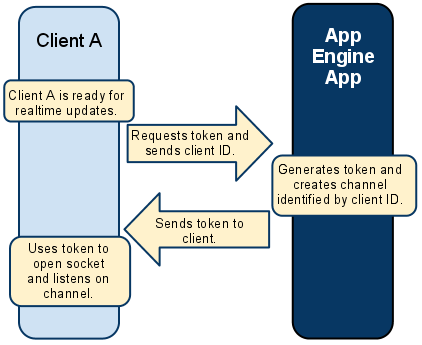
Khi nói đến việc hiển thị nhanh chóng các thông tin thay đổi từ server đến client, các lập trình viên phần mềm thường có hai chiến lược: mô hình thứ nhât là phía client gửi yêu cầu cập nhật thì server sẽ trả về các thông tin đã thay đổi hoặc mô hình thứ hai là mỗi lần server thay đổi thông tin sẽ tự động gửi về phía client.

Việc xây dựng mô hình thứ hai khá phức tạp và đòi hỏi chuyên sâu về kiến thức mạng tầng thấp, thật may măn khi Google cung cấp một thư viện mà các gói phức tạp khó hiểu thành các API dễ hiểu đó là Channel API.

Các thành phần của Channel API như sau:

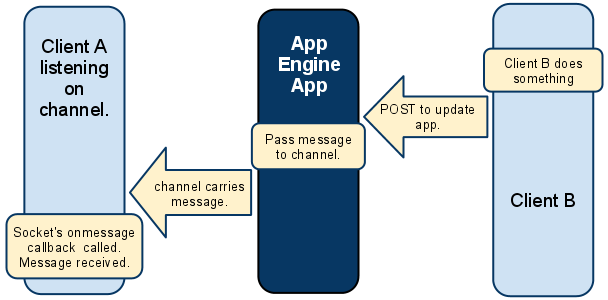
* Javasript client: Người sử dụng tương tác với một Javascipt client đã được xây dựng thành một trang web. Javascript client này phải có kết nối với một kênh duy nhất từ máy chủ, cập nhật thông tin liên quan đến các khách hàng khác, sử dụng dữ liệu và gửi tin nhắn cập nhật cho server.
* Server: có trách nhiệm tạo ra kênh duy nhất cho một Javascript client, tạo và gửi thông báo cho họ kết nối, nhận được thông báo cập nhật của khách hàng thông qua HTTP và gửi tin nhắn cập nhật cho khách hàng thông qua kênh của họ.
* Client ID: Dùng để nhận biết chính xác Javascript client trên server, Server dựa vào Client Id để gửi thông điệp cho client.
* Channel: là kênh thông tin server dùng để gửi thông điệp chi client.
* Message : Tin nhắn được gửi thông qua yêu cầu HTTP, giới hạn dung lượng một tin nhắn là 32K.
* Tokens: chịu trách nhiệm cho phép Javascript Client kết nối và lắng nghe những kênh được tạo ra cho nó.

Cách thức giao tiếp giữa Javascript Client và Server được thể hiện qua 2 sơ đồ sau:



Hình 6 - Sơ đồ tạo một kênh trên máy chủ

Sơ đồ thể hiện việc tạo ra một kênh trên máy chủ. Javascript Client gửi các yêu cầu tạo kênh và client Id cho server. Server tạo ra một kênh bằng client ID đã được gửi, và gửi cái token ấy về cho client. Client có thể sử dụng kết nối và lắng nghe kênh đó.



Hình 7 - Sơ đồ nhận thông tin mới

Sơ đồ tiếp theo này thể hiện việc Clien B gửi tin nhắn sử dụng POST đến server, server xử lý tin nhắn ấy và gửi cho Clien A qua kênh đã tạo ở trên. Client A nhận tin nhắn và sử dụng thông tin mới này.

### 1.5.5 Tổng quan về Google Maps API v3

Google Maps là một dịch vụ ứng dụng và công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google, hỗ trợ nhiều dịch vụ khác của Google nổi bật là dẫn đường. Nó cho phép thấy bản đồ đường sá, đường đi cho xe đạp, cho người đi bộ và xe hơi, và những địa điểm kinh doanh trong khu vực cũng như khắp nơi trên thế giới.

Map API là gì?

* Là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của website A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các website cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google (di chuột, room, đánh dấu trên bản đồ…)
* Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ javascripts do vậy việc sử dụng API google rất dễ dàng.
* Google Map API đã được nâng cấp lên phiên bản v3 không chỉ hỗ trợ cho các máy để bàn truyền thống mà cho cả các thiết bị di động; các ứng dụng nhanh hơn và nhiều hơn .

Các dịch vụ hoàn toàn miễn phí với việc xây dựng một ứng dụng nhỏ. Trả phí nếu đó là việc sử dụng cho mục đích kinh doanh, doanh nghiệp.

Hướng dẫn và cách sử dụng đều được Google cung cấp trên trang <https://developers.google.com/maps/>.

# CHƯƠNG 2 - PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG PHẦN MỀM

Trong phần em sẽ tiến hành mô tả tổng quan về hệ thống, đưa ra các phân tích thiết kế hệ thống, chi tiết các chức năng chính, phân tích thiết kế CSDL và thiết kế giao diện cho phần mềm.

## 2.1 Kiến trúc hệ thống



Datastore

Web Service

Web client

Hình 8 - Kiến trúc hệ thống

Kiến trúc hệ thống bao gồm :

* Datastore: với ưu điểm mạnh là đọc và truy vấn. Một ứng dụng tạo ra các thực thể, với giá trị dữ liệu được lưu trữ như thuộc tính của một thực thể. Các ứng dụng có thể thực hiện truy vấn trên các thực thể. Tất cả các truy vấn được lập chỉ mục trước cho kết quả nhanh hơn trên dữ liệu lớn. Với python, thực thể datastore được tạo từ đối tượng Python. Thuộc tính của đối tượng trở thành thuộc tính của thực thể. Datastore không sử dụng SQL để truy vấn dữ liệu mà sử dụng Gql, ngôn ngữ tương tự với SQL.
* Google App Engine : nền tảng cloud PaaS của Google sử dụng ngôn ngữ Python và các API cũng như sử dụng platform cung cấp sẵn để phát triển các ứng dụng Web, Web service …
* Web Service : Là nơi cung cấp các service cho client, được xây dựng và đặt trên cloud. Sử dụng phương thức truyền là RPC. Web service sẽ chứa các phương thức để thao tác với cơ sở dữ liệu trên datastore, đáp lời gọi từ phía client cũng như trả về kết quả.
* Web Client: Là các modul chính cuả chương trình học và thi trực tuyến trên nền GWT. Nó bao gồm các lớp Activity xử lý và lớp View hiển thị cho người dùng.

## 2.2 Tổng quan về hệ thống

### 2.2.1 Mục tiêu hệ thống

* Cho phép người dùng có thể tải và cài đặt vào nhiều thiết bị, trên các hệ điều hành khác nhau
* Cho phép đăng ký tài khoản sử dụng hệ thống
* Đối với tài khoản là nhà cung cấp thì cho phép tạo thêm sản phẩm, đồng thời quản lý sản phẩm đang cung cấp
* Đối với tài khoản là khách hàng thì đáp ứng nhanh chính xác khi khách hàng tìm kiếm sản phẩm.
* Hiển thị chỉ đường trên bản đồ, cập nhật được vị trí khi nhà cung cấp di chuyển thông báo cho khách hàng
* Hệ thống đảm bảo tính ổn định, chính xác, bảo mật và an toàn cao.
* Hệ thống có giao diện, đặc biệt là trên điện thoại mang tính thẩm mỹ và thân thiện, dễ sử dụng cho tất cả các đối tượng sử dụng điện thoại thông minh.
* Hệ thống mang tính linh hoạt cao, dễ mở rộng và đảm bảo giảm tối đa chi phí vận hành.

### 2.1.2 Yêu cầu chung của hệ thống

|  |  |
| --- | --- |
| STT | Mô tả |
| 01 | Đảm bảo chạy tốt trên điện thoại di động, máy tính bảng sử dụng hệ điều hành Android 4.0.0 trở lên |
| 02 | Đảm bảo chạy tốt trên máy tính để bàn, laptop qua mọi trình duyệt web. |
| 03 | Điện thoại phải được kết nối wifi hoặc 3G của bất kì nhà cung cấp dịch vụ nào. |
| 04 | Máy tính phải được kết nối mạng wifi, mạng ethernet hoặc mạng 3G. |
| 05 | Ngôn ngữ lập trình được sử dụng là Java, JavaScript, XML, HTML, CSS … |

***Bảng 4 - Yêu cầu chung của hệ thống***

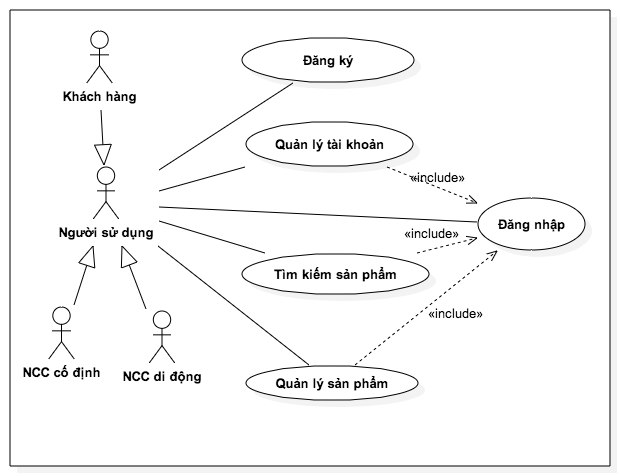
### 2.2.3 Sơ đồ tổng quan của hệ thống



Hình 9 - Sơ đồ tổng quan

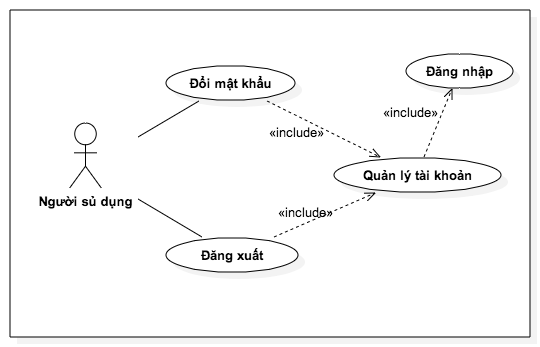
## 2.3. Thiết kế chương trình

### 2.3.1.Phân tích chức năng và biểu đồ UseCase của hệ thống



Hình 10 - UseCase Tổng quát

#### 2.3.1.1. Quản lý tài khoản

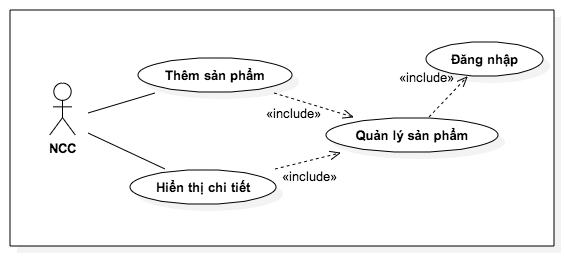


Hình 11 - UseCase quản lý tài khoản

|  |  |
| --- | --- |
| Tên UseCase | **Quản lý tài khoản** |
| Actor | Người sử dụng |
| Dòng sự kiện chính | * Người sử dụng lựa chọn thay đổi mật khẩu * Người sử dụng lựa chọn đăng xuất |
| Dòng sự kiện phụ | * Phẩn mềm đưa ra phần thiết lập mật khẩu người sử dụng phải điền đầy đủ thông tin rùi có thể ấn thay đổi hoặc không. |

**Bảng 5 - Mô tả UseCase quản lý tài khoản**

#### 2.3.1.2. Quản lý sản phẩm

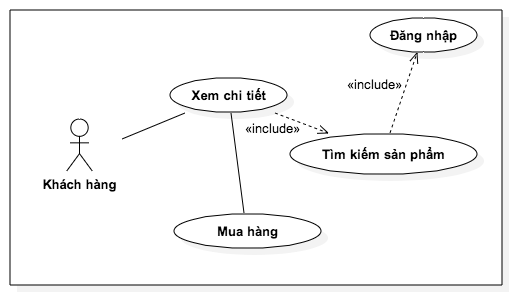


Hình 12 - UseCase Quản lý sản phẩm

|  |  |
| --- | --- |
| Tên UseCase | **Quản lý sản phẩm** |
| Actor | Người sử dụng (NCC cố định, NCC di động) |
| Dòng sự kiện chính | * Thêm sản phẩm : điền đủ các thông tin của một sản phẩm mới (tên sản phẩm, giá, số lượng….) * Hiển thị chi tiết các sản phẩm đang cung cấp |
| Dòng sự kiện phụ |  |

Bảng 6 - Mô tả Quản lý sản phẩm

#### 2.3.1.3. Tìm kiếm sản phẩm



Hình 13 - UseCase tìm kiếm sản phẩm

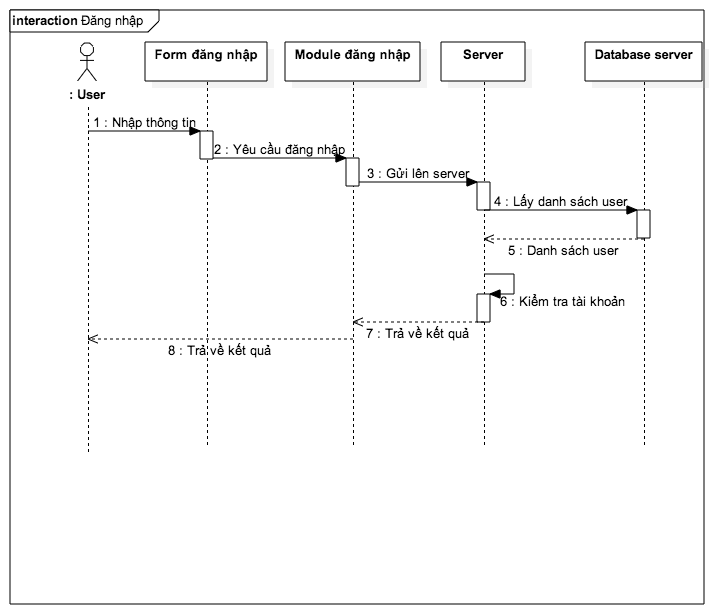
|  |  |
| --- | --- |
| Tên UseCase | **Tìm kiếm sản phẩm** |
| Actor | Người sử dụng ( Khách hàng) |
| Dòng sự kiện chính | * Người sử dụng có thể tìm kiếm sản phẩm dựa vào các điều kiện: tên sản phẩm, danh mục sản phẩm, giá tiền, khoảng cách, điểm đánh giá * Mua hàng: tìm được sản phẩm thì có thể gửi yêu cầu mua sản phẩm hiển thị vị trí của nhà cung cấp trên bản đồ |
| Dòng sự kiện phụ |  |

Bảng 7 - Mô tả Tìm kiếm sản phẩm

### 2.3.2. Biểu đồ trình tự và biểu đồ hoạt động của hệ thống

#### 2.3.2.1. Chức năng Đăng nhập

**Biểu đồ trình tự :**

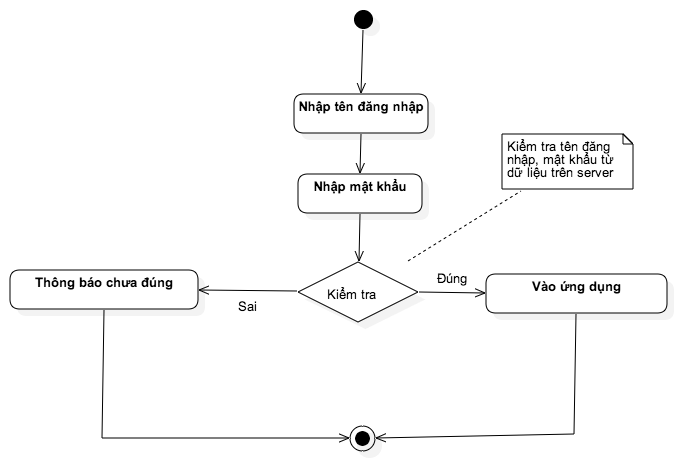


Hình 14 - Biểu đồ trình tự chức năng đăng nhập

**Mô tả hoạt động:**

* Ngưởi dùng :
* Nhập tên đăng nhập.
* Nhập mật khẩu.
* Người dùng ấn nút "Đăng nhập", khi đó server kiểm tra với các giá trị tương ứng người dùng chuyển sang giao khác nhau :
* Nếu là khách hàng chuyển sang giao diện của khách hàng
* Nếu là nhà cung cấp cố định thì chỉ sang giao diện nhà cung cấp cố định
* Nếu là nhà cung cấp di động thì chuyển sang giao diện nhà cung cấp di động

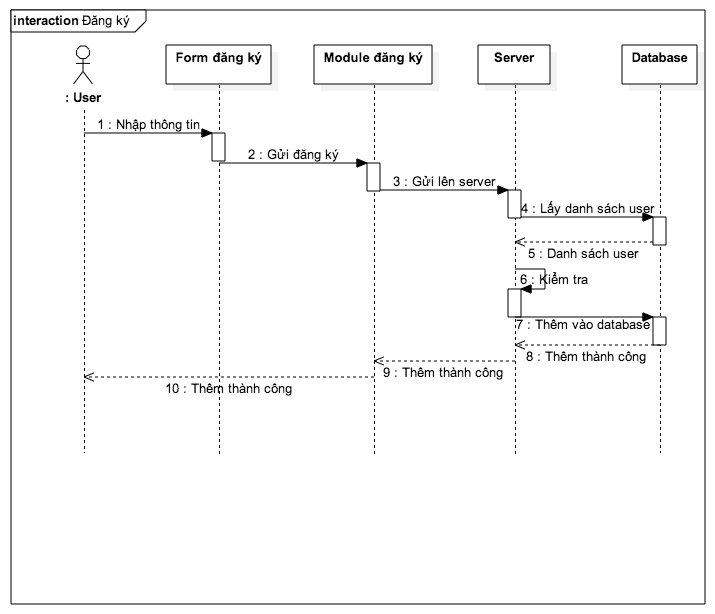
**Biểu đồ hoạt động :**



Hình 15 - Biểu đồ hoạt động chức năng đăng nhập

#### 2.3.2.2. Chức năng Đăng ký

**Biểu đồ trình tự:**

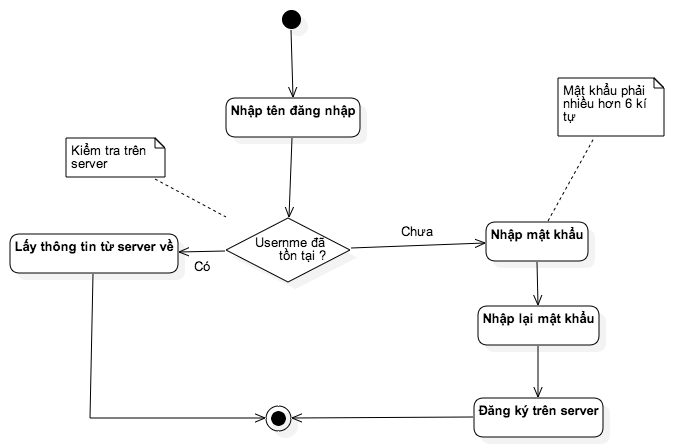


Hình 16 - Biểu đồ trình tự chức năng đăng ký

**Mô tả hoạt động:**

* Ngưởi dùng nhập các thông tin :
* Nhập tên tài khoản : tên tài khoản không chứa các kí tự đặc biệt
* Nhập mật khẩu, và nhắc lại mật khẩu : mật khẩu phải giống nhau đông thời thì phải có độ dài lớn hơn 6 kí tự.
* Chọn kiểu của tìa khoản :
* 0 – khách hàng
* 1 – nhà cung cấp cố định
* 2 – nhà cung cấp di đông
* Các thông tin đều được lưu trữ trên server.

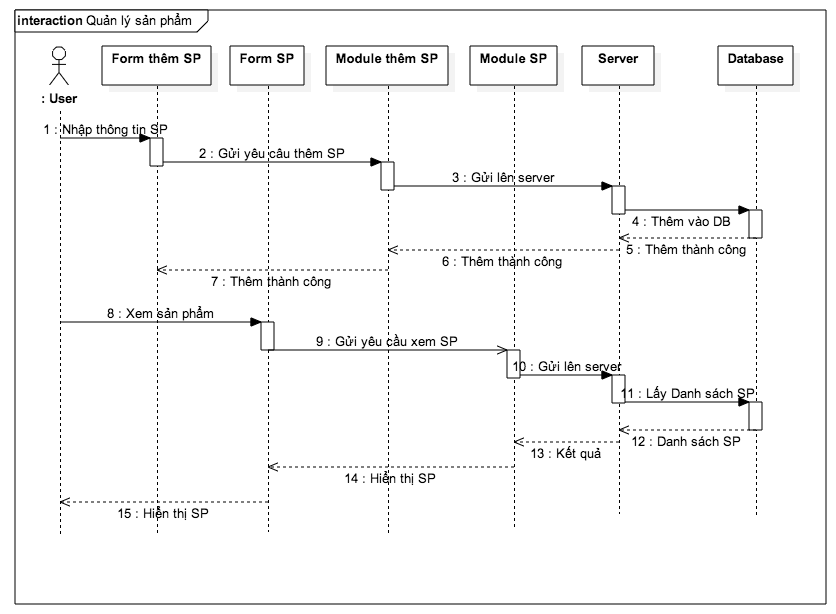
**Biểu đồ hoạt động :**



Hình 17 - Biểu đồ hoạt động chức năng đăng ký

#### 2.3.2.3. Chức năng Quản lý sản phẩm

**Biểu đồ trình tự :**

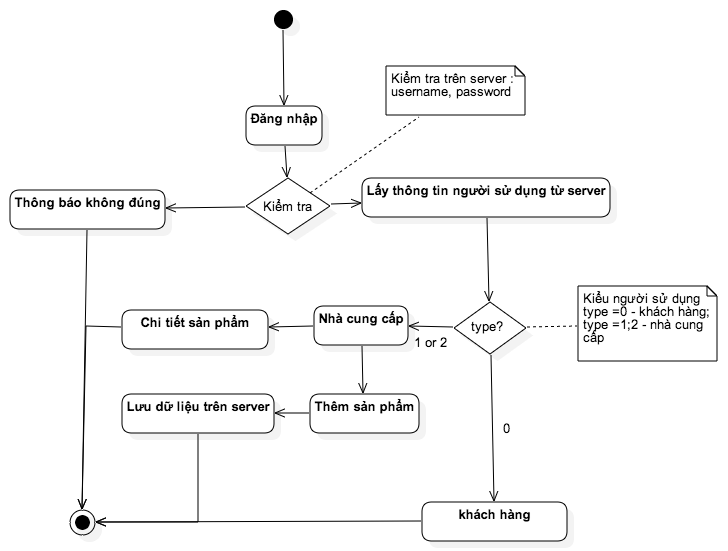


Hình 18 - Biểu đồ trình tự chức năng quản lý sản phẩm

**Mô tả hoạt động:**

* Người sử dụng là nhà cung cấp thực hiện các thao tác sau
* Thêm sản phẩm : nhập đầy đủ thông tin cho một sản phẩm mới, sau đó dữ liệu được lưu trên server.
* Hiển thị các mặt hàng đang cung cấp : tên sản phẩm, giá cả, hình ảnh, điểm đánh giá.

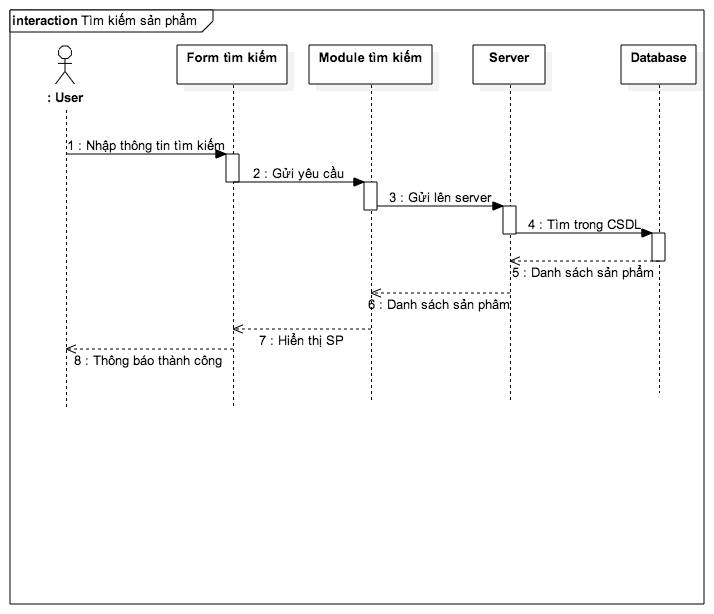
**Biểu đồ hoạt động:**



Hình 19 - Biểu đồ hoạt động chức năng quản lý sản phẩm

#### 2.3.2.4. Chức năng Tìm kiếm sản phẩm

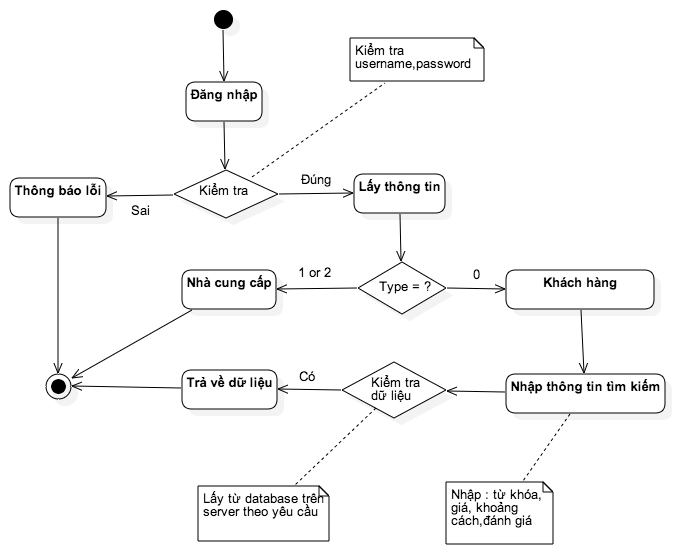
**Biểu đồ trình tự**

Hình 20 - Biểu đồ trình tự chức năng Tìm kiếm sản phẩm

**Mô tả hoạt động:**

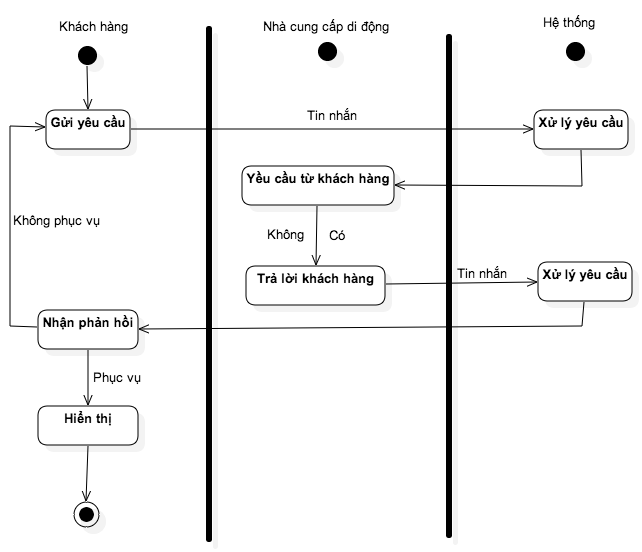
* Ngưởi dùng (khách hàng) nhập đầy đủ thông tin tìm kiếm : từ khóa, danh mục sản phẩm, giá tiền, khoảng cách…
* Server sẽ trả về dữ liệu cho khách hàng các sản phẩm phù hợp, hiển thị ra các sản phẩm của 2 nhà cung cấp cố định và di động.
* Nhà cung cấp cố định: hiển thị bản đồ chỉ đường cho khách hàng đi tới nhà cung cấp
* Nhà cung cấp di động: thực hiện chức năng mua hàng.

**Biểu đồ hoạt động :**



Hình 21 - Biểu đồ hoạt động chức năng tìm kiếm sản phẩm

#### 2.3.2.5. Biểu đồ quá trình mua hàng



Hình 22 - Biểu đồ hoạt động quá trình mua hàng

**Mô tả hoạt động:**

Quá trình mua hàng giữa khách hàng và nhà cung cấp di động, nó hoạt động như sau :

* Khách hàng tìm được sản phẩm phù hợp sau đó gửi yêu cầu muốn mua hàng đến nhà cung cấp di động.
* Nhà cung cấp nhận được yêu cầu trả lời lại khách hàng: đồng ý phục vụ hoặc không đồng ý
* Khách hàng nhận được trả lời từ nhà cung cấp :

Nếu đồng ý phục vụ : hiển thị vị trí của nhà cung cấp trên bản đồ, quan sát thấy được nhà cung cấp đang di chuyển.

Không đông ý: quay lại yêu cầu nhà cung cấp khác

* Việc trao đổi giữa khách hàng với nhà cung cấp đều phải qua server xử lý.

## 2.4. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống được lưu trữ toàn bộ trên server bao gồm : thông tin các tài khoản của người dùng, thông tin sản phẩm của các nhà cung cấp, thông tin của khách hàng, ảnh, tin nhắn. Do cơ sở dữ liệu được lưu trữ ở GAE Datastore không giống như cơ sở dữ liệu SQL nên em sẽ trình bày phần mô tả cơ sở dữ liệu bằng 2 phần: Phần thứ nhất là bảng mô tả và phần thứ hai là lớp được lưu trữ trên Datastore

### 2.4.1. Dữ liệu thông tin tài khoản người dùng

**Bảng dữ liệu :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Null | PK | Mô tả |
| UserID | String | No | Yes | Mã người dùng |
| Username | String | No |  | Tên đăng nhập |
| Password | String | No |  | Mật khẩu |
| Type | Interger | No |  | Kiểu người dùng |
| UserToken | String | No |  | Token cấp cho người dùng |
| UserLocation | String | No |  | Vị trí của người dùng |
| StateOnOff | Interger | No |  | Trạng thái đăng nhập |

***Bảng 8 - Tài khoản của người dùng***

@Entity

**public** **class** UserInfo **implements** Serializable {

@Id **private** Long id;

@Index **private** String username;

@Index **private** String password;

@Index **private** Integer type;

@Index **private** String userId;

@Index **private** String userToken;

@Index **private** String channelToken;

@Index **private** String userLocation;

@Index **private** Integer stateOnOFF = 0;// 0 -off, 1- on

**public** UserInfo() {

**super**();

}

### 2.4.2. Thông tin của khách hàng

**Bảng dữ liệu :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Null | PK | Mô tả |
| UserID | String | No | Yes | Mã khách hàng |
| Username | String | No |  | Tên đăng nhập |
| Type | Interger | No |  | Kiểu |
| Fullname | String | No |  | Họ tên khách hàng |
| Email | String | No |  | Thư điện tử |
| PhoneNumber | Interger | No |  | Số điện thoại |

***Bảng 9 - Thông tin khách hàng***

@Entity

**public** **class** Customer **implements** Serializable,IsSerializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

@Id **private** Long id;

@Index **private** String username;

@Index **private** Integer type;

@Index **private** String userid;

@Index **private** String userToken;

@Index **private** String channelToken;

@Index **private** String fullname;

@Index **private** String email;

@Index **private** **int** phoneNumber;

**public** Customer() {

**super**();

}

### 2.4.4. Thông tin sản phẩm của nhà cung cấp

**Bảng dữ liệu :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Null | PK | Mô tả |
| UserId | String | No | Yes | Mã nhà cung cấp |
| UserName | String | No |  | Tên đăng nhập |
| Type | String | No |  | Kiểu nhà cung cấp |
| Name\_provider | String | No |  | Tên nhà cung cấp |
| Name\_item | String | No |  | Tên sản phẩm |
| Catalog | String | No |  | Danh mục sản phẩm |
| Image | String | No |  | Ảnh sản phẩm |
| Description\_item | String | No |  | Mô tả |
| Number\_item | Integer | No |  | Số lượng |
| Price\_item | Integer | No |  | Giá |
| State\_item | String | No |  | Trạng thái |
| Address | String | No |  | Địa chỉ |
| Location | String | No |  | Vị trí |
| Rate\_item | Integer | No |  | Điểm đánh giá sản phẩm |
| Rate\_provider | Integer | No |  | Điểm đánh giá NCC |

***Bảng 10 - Thông tin sản phẩm nhà cung cấp***

@Entity

**public** **class** Provider **implements** Serializable,IsSerializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

@Id **private** Long id;

@Index **private** String userId;

@Index **private** String username;

@Index **private** Integer type;

@Index **private** String name\_provider;

@Index **private** String catalog\_item;

@Index **private** String name\_item;

@Index **private** String image;

@Index **private** String description\_item;

@Index **private** String number\_item;

@Index **private** Integer price\_item;

@Index **private** String state\_item;

@Index **private** String address;

@Index **private** String latlng;

@Index **private** Integer rate\_item = 0;

@Index **private** Integer rate\_provider = 0;

**public** Provider() {

**super**();

}

### 2.4.5. Thông tin ảnh

**Bảng dữ liệu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Null | PK | Mô tả |
| Key | String | No | Yes | Mã ảnh |
| Url | String | No |  | Đường dẫn của ảnh trên server |
| BlobKey | String | No |  | Key server cung cấp |

Bảng 11 - Thông tin ảnh lưu trữ

@Entity

**public** **class** Picture **implements** Serializable, IsSerializable {

@Id **private** **long** id;

@Index **private** String key;

@Index **private** String url;

@Index **private** String blobKey;

**public** Picture() {

}

**public** Picture(Long id,String key,String url, String blobKey) {

**super**();

**this**.id = id;

**this**.key = key;

**this**.url = url;

**this**.blobKey = blobKey;

}

### 2.4.6 Tin nhắn

**Bảng dữ liệu** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Null | PK | Mô tả |
| Id | Long | No | Yes | Mã |
| ClientSend | String | No |  | Người gửi |
| ClientIdSend | String | No |  | Mã người gửi |
| ClientReceive | String | No |  | Người nhận |
| ClientIdReceive | String | No |  | Mã người nhận |
| Message | String | No |  | Nội dung tin nhắn |

Bảng 12 - Tin nhắn

@Entity

**public** **class** MessageImpl **implements** Serializable, IsSerializable, Message {

@Id **private** **long** id;

@Index **private** String clientSend;

@Index **private** String clientIdSend;

@Index **private** String clientReceive;

@Index **private** String clientIdReceive;

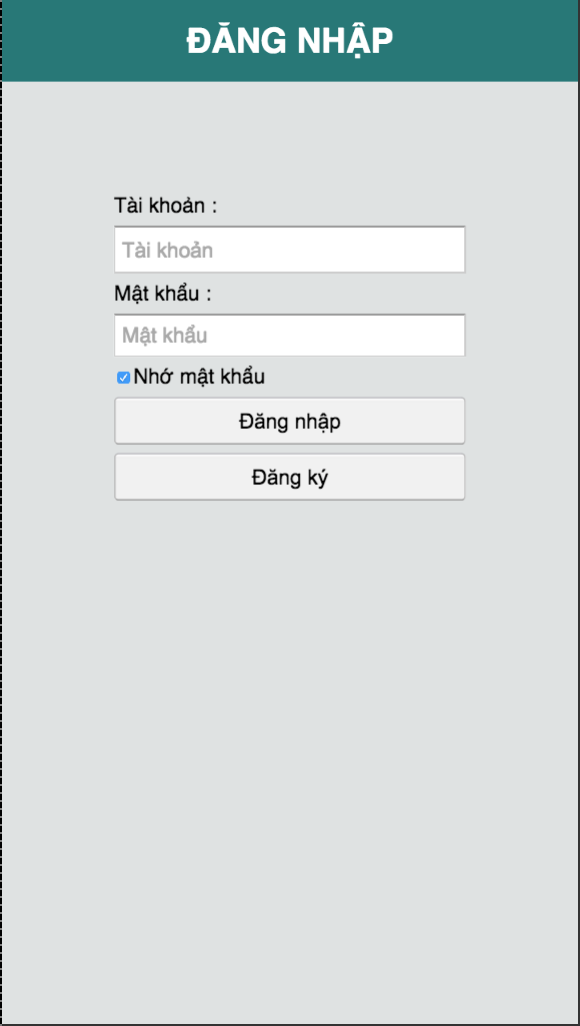
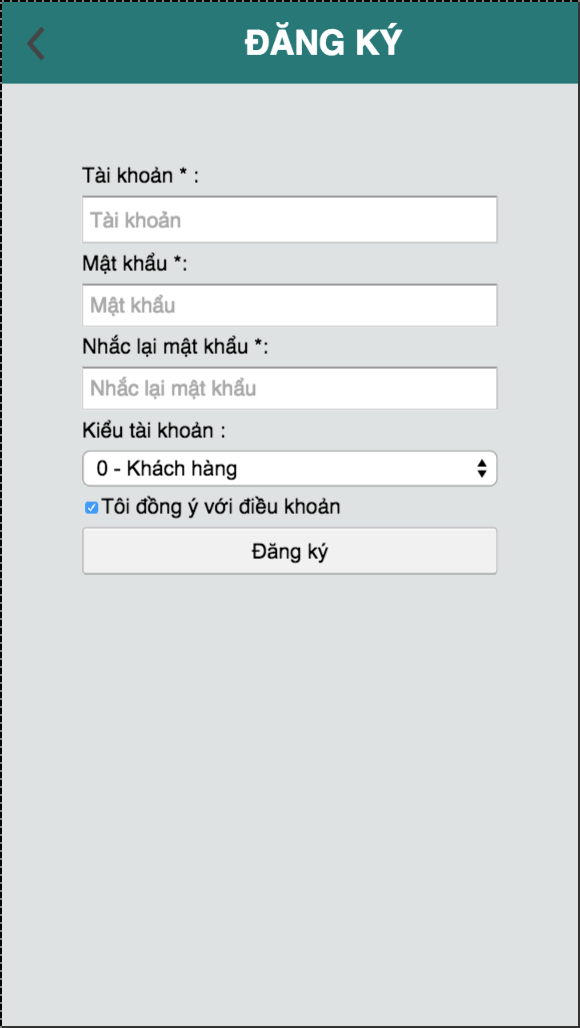
**private** String message;

**public** MessageImpl() {

}

## 2.5. Thiết kế giao diện

### 2.5.1. Thiết kế giao diện màn hình đăng nhập và đăng ký

Hình 23 - Giao diện đăng nhập và đăng ký

Mô tả:

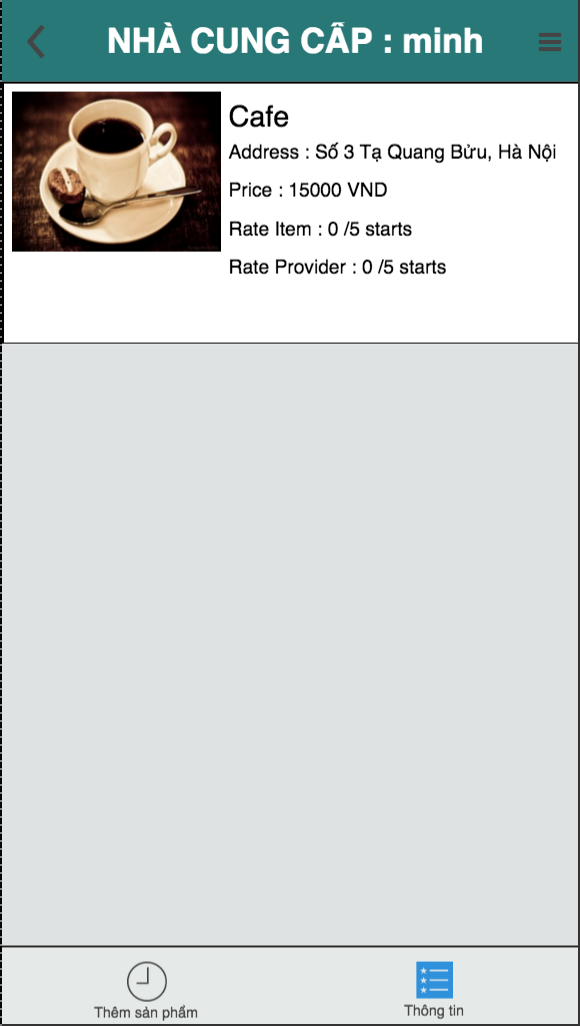
Giao diện cho phép người dùng có sự lựa chọn sau :

* Nếu đã có tài khoản và mật khẩu chọn đăng nhập.
* Khi chưa có tài khoản thì chuyển sang giao diện đăng ký

Trong phần đăng ký có các lựa chọn sau :

* Đăng ký với tài khoản là khách hàng
* Đăng ký với tài khoản là nhà cung cấp cố định
* Đăng ký với tài khoản là nhà cung cấp di động

### 2.5.2. Thiết kế giao diện cho nhà cung cấp sản phẩm

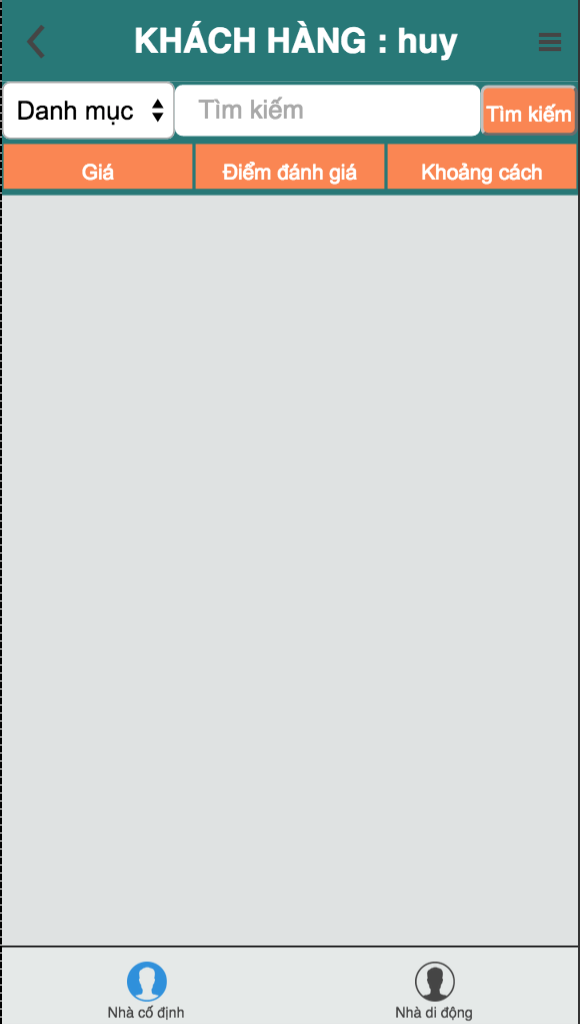
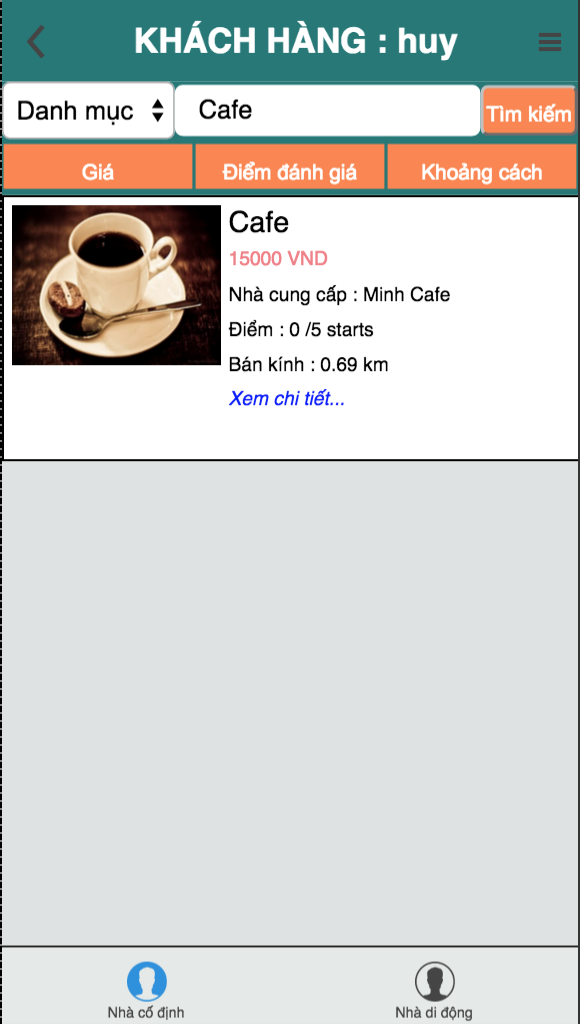
Hình 24 - Giao diện nhà cung cấp

Mô tả:

Giao diện giúp cho nhà cung cấp có thể lựa chọn :

* Thông tin : Hiển thị các sản phẩm hiện đang được cung cấp
* Thêm sản phẩm : Cho phép người dùng thêm những sản phẩm muốn cung cấp.

### 2.5.3. Thiết kế giao diện cho khách hàng

Hình 25 - Giao diện khách hàng

Mô tả:

* Giao diện giúp khách hàng có thể tìm kiếm được các sản phẩm.

Khách hàng có thể lựa chọn các tiêu chí :

* Giá : Chọn mức giá mong muốn
* Điểm đánh giá: Tìm những nhà cung cấp với điểm đánh giá mong muốn.
* Khoảng cách : Chọn khoảng cách mong muốn.

### 2.5.4. Thiết kế giao diện hiển thị chi tiết sản phẩm



Hình 26 - Giao diện xem chi tiết sản phẩm

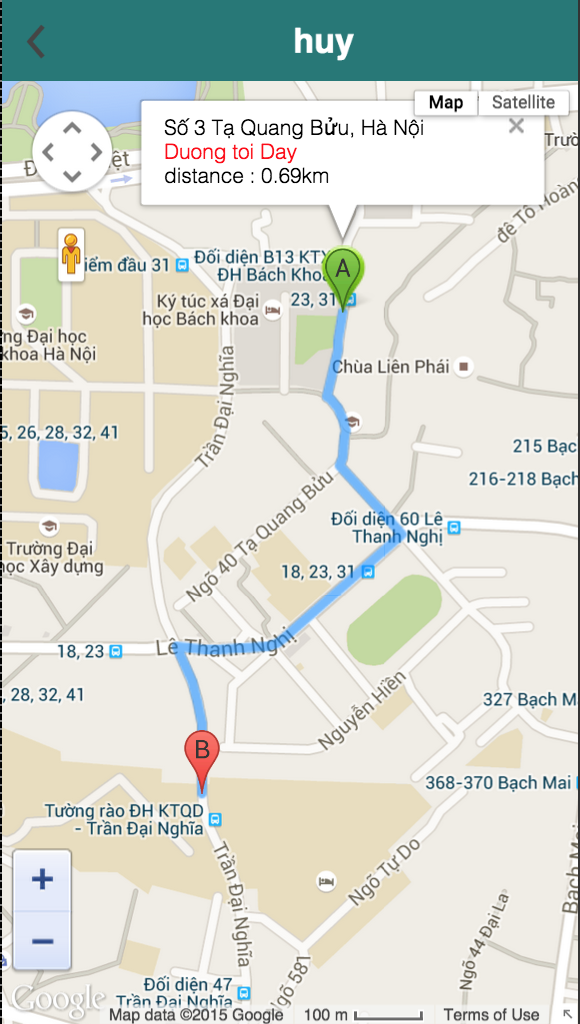
Mô tả:

Hiển thị toàn bộ thông tin của sản phẩm vừa tìm kiếm được.

Người dùng có thể lựa chọn :

* Xem vị trí và chỉ dường đến nhà cung cấp
* Đánh giá sản phẩm và nhà cung cấp

### 2.5.5. Thiết kế giao diện hiển thị map



Hình 27 - Giao diện hiển thị map

Mô tả:

Hiển thị trên bản đồ đường đi từ khách hàng đến nhà cung cấp sản phẩm

Trên đây là một số giao diện tổng quát của ứng dụng, ngoài ra còn một số giao diện nhỏ nhưng em không đề cập ở đây.

# 

# CHƯƠNG 3 – TRIỂN KHAI THỬ NGHIỆM HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ

Nội dung phần này trình bày kết quả cài đặt và thử nghiệm của ứng dụng trong thực tế, đánh giá các ưu nhược điểm của ứng dụng và kiểm thử các tính năng của ứng dụng

## 3.1 Cài đặt

### 3.1.1 Môi trường cài đặt

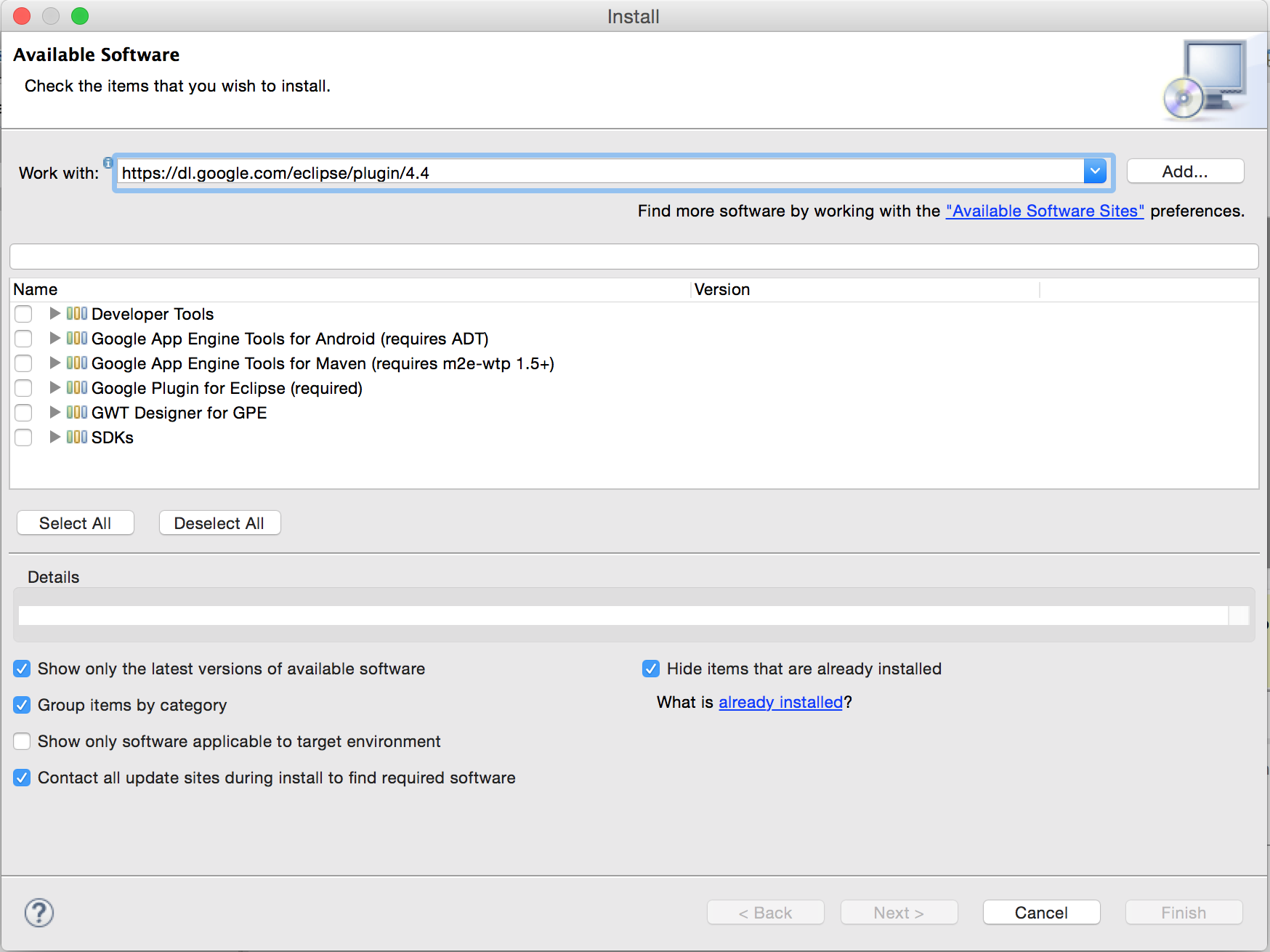
* Java SE Development Kit 7 update 25
* Công cụ lập trình: Eclipse Luna
* Google Plugin for Eclipse 4.4
* Google Web toolkit SDK 2.7.0
* App Engine SDK 1.9.17
* Cordova 5.0.0
* Phonegap 2.4.0
* Ngôn ngữ lập trình: Java
* Hệ điều hành : Mac OS X Yosemite 10.10.2

### 3.1.2 Cài đặt Google Web Toolkit (GWT)

Hướng dẫn cài đặt GWT :

* Dowload tại : <http://www.gwtproject.org/download.html>.
* Hoặc có thể download plugin cho eclipse : <https://dl.google.com/eclipse/plugin/4.4>

Eclipse/ Help/ Install New Software…



Hình 28 : Màn hình cài đặt eclipse

### 3.1.3 Cài đặt Google App Engine (GAE)

Download tại : https://cloud.google.com/appengine/downloads.

## 3.2 Cài đặt giải thuật

Trong phần này, em xin trình bày giải thuật cho cơ chế realtime khách hàng có thể nhìn thấy được vị trí của nhà cung cấp di động đang di chuyển :

* Đăng nhập => cấp ID => vào Channel
* Lấy vị trí hiện tại của khách hàng và nhà cung cấp di động
* Dùng timer để lấy vị trí của nhà cung cấp di động sau đó dùng channel gửi tin nhắn cho khách hàng
* Khách hàng nhận được tin nhắn ( vị trí hiện tại của NCC di động) sẽ update lại giao diện hiển thị.

## 3.3 Kiểm thử và đánh giá hệ thống

### 3.3.1 Kiểm thử

Kiểm thử ứng dụng là hoạt động khảo sát thực tiễn sản phẩm trong đúng môi trường chúng dự định sẽ được triển khai để có được những thông tin về chất lượng của sản phẩm. Mục đích của kiểm thử ứng dụng là tìm ra các lỗi hay khiếm khuyết của ứng dụng nhằm đảm bảo hiệu quả hoạt động tối ưu của ứng dung trong quá trình triển khai, mang lại lợi ích lớn nhất cho tất cả các bên liên quan.

#### 3.3.1.1 Kiểm thử về giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Yêu cầu test | Yêu cầu kết quả | Kết quả |
| 1 | Màn hình ban đầu là màn hình đăng nhập | Hiển thị ra màn hinh | YES |
| 2 | Đăng nhập thành công | Hiển thị giao diện tiếp theo | YES |
| 3 | Đăng nhập tài khoản nhà cung cấp | Hiển thị được giao diện quản lý sản phẩm | YES |
| 4 | Khách hàng tìm kiếm | Hiển thị ra các sản phẩm mà tìm kiếm được | YES |
| 5 | Khách hàng yêu cầu cung cấp từ nhà cung cấp cố định | Hiển thị được vị trí đường đi trên bản đồ | YES |
| 6 | Khách hàng yêu cầu từ nhà cung cấp di động | Hiển thị được trao đổi giữa khách hàng và NCC di động, hiển thị được trên bản đồ cơ chế khi NCC di chuyển | YES |
| 7 | Thiết lập tài khoản | Hiển Giao diện cho phép ngừời dùng thay đổi. | YES |
| 8 | Chi tiết sản phẩm khi được tìm kiếm | Hiển thị được chi tiết sản phẩm mà được tìm thấy | YES |
|  |  |  |  |

Bảng 13 - Kiểm thử giao diện

#### 3.3.1.2 Các tình huống kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| Tình huống | Dữ liệu | Yêu cầu kết quả | Kết quả |
| Nhập thiếu dữ liệu khi đăng ký tài khoản | Nhập vào địa chỉ username đã tồn tại trong hệ thống | Hiện thông báo cho người dung biết tài khoản đã được đăng ký. | YES |
| Đăng ký thành công tài khoản |  | Hiển thị thông báo thành công | YES |
| Thay đổi mật khẩu |  | Hiển thị thông báo thành công | YES |
| Kiểm tra email, sồ điện thoại | Nhập vào email ,số điện thoại | Hiên thị kết quả kiểm tra | NO |
| Kiểm tra nhắc lại mật khẩu | Nhập vào mật khẩu, và nhắc lại mật khẩu | Hiển thị thông báo cho người dùng biết mật khẩu không khớp nhau | NO |
|  |  |  |  |
|  |  |  |  |

Bảng 14 - Tình huống kiểm thử

# CHƯƠNG 4 - KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong phần này em sẽ tổng kết lại những gì đồ án đã đạt được và hướng phát triển tương lại cho ứng dụng bán hàng dựa vị trí mà em xây dựng.

## 4.1 Các kết quả đạt được

Đồ án tốt nghiệp đã thu được các kết quả sau:

* Tìm hiểu và nắm được các kiến thức cơ bản về các nền tảng mà Google cung cấp : Google Web Toolkit(GWT), Google App Engine(GAE), Google map API v3.
* Nắm được những kiến thức cơ bản về lập trình trên nền tảng của Google: ngôn ngữ lập trình, công cụ lập trình.
* Cài đặt, thử nghiệm hệ thống thành công.
* Hiện tại, do ứng dụng đang trong thời gian phát triển lâu dài nên trong đồ án này chỉ hoàn thành các chức năng cơ bản của ứng dụng.

## 4.2 Hướng phát triển trong tương lai

Qua đánh giá hệ thống trong phần 3.2.1 em nhận thấy sản phẩm của mình vẫn còn nhiều hạn chế vì thế trong tương lại em sẽ cố gắng hoàn thiện ứng dụng, khắc phục những nhược điểm còn tồn tại của ứng dụng, những lỗi phát sinh trong quá trình sử dụng ứng dụng

Trong tương lai, để ứng dụng ngày một hoàn thiện và có thể triển khai thực tế một cách rộng rãi, đem đến một sự trải nghiệm tốt nhất cho người dùng thì phải làm những công việc sau :

* Cải thiện lại giao diện cho ứng dụng một cách đẹp hơn, có tính thẩm mỹ cao, dễ sử dụng.
* Tối ưu lại về cách quản lý cơ sở dữ liệu.
* Nghiên cứu và phát triển thêm một số tính năng : nhà cung cấp sản phẩm thống kê tất cả các đơn đặt hàng trong tháng, thêm sản phẩm vào giỏ hàng…

# 

# TÀI LIỆU THAM KHẢO

1. Danien Guemeur, Ami Unruh, Google App Engine Java and GWT Application Development, University of Technology of Compiegne, 2010.
2. Prabhakar Chaganti, Google Web Toolkit Java Ajax Programming, University of Iowa Hygienic Laboratory, 2007.
3. Vipul Gupta, Accelerated GWT Building Enterprise Google Web Toolkit Applications, 2008.
4. David Geary with Rob Gordon, Google Web Toolkit Solutions More Cool & Useful Stuff, 2008.
5. Ryan Dewsbury,Google Web Toolkit Applications, 2007.
6. Robert Hanson, Adam Tacy, GWT in Action Easy Ajax with the Google Web Toolkit, 2007.
7. Charles Severance, Using Google App Engine Building the Web application, O’Reilly media, 2009.
8. Dan Sanderson, Progamming Google App Engine Building and run scalable Web Apps on Google’s, 2009.
9. Weslay Hales, HTML5 and Javascript Web App,Bridging the Gap bettwen the Web and the Mobile Web, 2012.
10. <https://developers.google.com/appengine>
11. <https://developers.google.com/webtool-kit>
12. <http://vi.wikipedia.org/>