**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

**NGUYỄN TẤN ĐÔ – NGUYỄN TRƯƠNG TRUNG TÍN**

**HỆ THỐNG ĐIỂM THƯỞNG TRÊN THIẾT BỊ DI ĐỘNG DÙNG ANDROID**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**TP. HCM, 2015**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

**NGUYỄN TẤN ĐÔ – 1112074**

**NGUYỄN TRƯƠNG TRUNG TÍN – 1112330**

**HỆ THỐNG ĐIỂM THƯỞNG TRÊN THIẾT BỊ DI ĐỘNG DÙNG ANDROID**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. ĐINH BÁ TIẾN**

**KHÓA 2011 - 2015**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Tp. Hồ Chí Minh, ngày … tháng … năm 2015

Giáo viên hướng dẫn

[Kí và ghi rõ họ tên]

**TS. ĐINH BÁ TIẾN**

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Khóa luận đáp ứng yêu cầu tốt nghiệp của cử nhân Công nghệ thông tin.

Tp. Hồ Chí Minh, ngày … tháng … năm 2015

Giáo viên phản biện

[Kí và ghi rõ họ tên]

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy – Tiến sĩ Đinh Bá Tiến, người trực tiếp hướng dẫn, giúp đỡ chúng em về phương pháp, kiến thức, và luôn tạo điều kiện tốt nhất cho chúng em hoàn thành tốt khóa luận của mình.

Chúng em xin trân trọng cảm ơn các Thầy Cô trong khoa Công Nghệ Thông Tin trường Đại học Khoa Học Tự Nhiên đã tận tình giảng dạy truyền đạt các kiến thức quý báu cho chúng em trong suốt bốn năm qua. Những kiến thức được học trên giảng đường sẽ là hành trang quý giá giúp chúng em vững bước trong tương lai.

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại học Khoa Học Tự Nhiên, Đại học Quốc Gia Tp. Hồ Chí Minh đã quan tâm và tạo những điều kiện thuận lợi cho chúng em trong quá trình học tập và thực hiện tốt đề tài tốt nghiệp.

Do trình độ nghiên cứu và thời gian có hạn, chúng em đã cố gắng hết sức nhưng chắc chắn không tránh khỏi những thiếu sót và hạn chế. Rất mong nhận được sự góp ý và chỉ dẫn của quý Thầy Cô.

Cuối cùng, chúng em xin trân trọng cảm ơn và chúc sức khỏe quý Thầy Cô!

Tp. Hồ Chí Minh, ngày … tháng … năm 2015

Sinh viên

Nguyễn Tấn Đô – Nguyễn Trương Trung Tín

Khoa Công Nghệ Thông Tin

Bộ môn Công Nghệ Phần Mềm

ĐỀ CƯƠNG CHI TIẾT

|  |  |
| --- | --- |
| **Tên đề tài:** Hệ thống điểm thưởng trên thiết bị di động dùng Android. | |
| **Giáo viên hướng dẫn:** TS. Đinh Bá Tiến | |
| **Thời gian thực hiện:** 01/12/2014 đến 30/6/2015. | |
| **Sinh viên thực hiện:**  Nguyễn Tấn Đô - 1112074  Nguyễn Trương Trung Tín - 1112330 | |
| **Loại đề tài:** Tìm hiểu công nghệ và xây dựng ứng dụng | |
| **Nội dung đề tài :**  Phân tích, tìm hiểu những mô hình và giải pháp tích lũy và sử dụng điểm sau khi mua hàng cho khách hàng đã và đang được hoạt động ở các hệ thống cửa hàng, siêu thị, để tìm ra được điểm mạnh, điểm yếu và hạn chế của mỗi giải pháp, từ đó đề xuất và lựa chọn giải pháp phù hợp nhất, có tính khả thi cao nhất đối với hệ thống tích lũy điểm cho mỗi hệ thống cửa hàng, siêu thị tại Việt Nam. | |
| **Kế hoạch thực hiện:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Tuần 1** | **Tuần 2** | **Tuần 3** | **Tuần 4** | | **Tháng 12** | - Thảo luận và thống nhất ý tưởng. | - Thảo luận và thống nhất ý tưởng. | - Thảo luận và thống nhất ý tưởng. | - Thảo luận và thống nhất ý tưởng. | | **Tháng 01** | – Nghiên cứu đặc điểm của điện toán đám mây  – Nghiên cứu về lưu trữ trên  đám mây | - Tìm hiểu về Amazon Web Webservice.  - Thống nhất bussiness case cho ứng dụng. | - Tìm hiểu cách phát triển ứng dụng android.  - Thảo luận use-case ứng dụng.  - Cài đặt hoàn thiện chức năng bất kỳ trên Amazon Web Webservice | - Thiết kế dữ liệu và chọn giải pháp lưu trữ dữ liệu trên Amazon Web Webservice.  - Hoàn thiện chức năng đơn giản: Đăng nhập | | **Tháng 02** | - Hoàn thiện tài liệu usecase  - Thiết kế UI  - Thiết kế sơ đồ kiến trúc cho hệ thống | - Hoàn thiện tài liệu usecase - Thiết kế UI  - Thiết kế sơ đồ kiến trúc cho hệ thống | - Hoàn thiện tài liệu usecase  - Thiết kế UI  - Thiết kế sơ đồ kiến trúc cho hệ thống | - Hoàn thiện tài liệu Đặc tả yêu cầu  - Hoàn thiện tài liệu Thiết kế phần mềm  - Hoàn thiện tài liệu Phát biểu công việc | | **Tháng 03** | - Hoàn thiện tài liệu Đặc tả yêu cầu  - Hoàn thiện tài liệu Thiết kế phần mềm  - Hoàn thiện tài liệu Phát biểu công việc | - Tổng hợp nội dung nghiên cứu lý thuyết và báo cáo vào khóa luận. | – Hoàn thiện chương 2 khóa luận  – Xem xét lại chương 3 và 4 theo yêu cầu khóa luận  – Tổng hợp nội dung nghiên cứu ứng dụng báo cáo vào khóa luận | - Hoàn thiện lại chương 3 và 4, trọng tâm vào di chuyển các ứng dụng lên đám mây và những ví dụ thực hành | | **Tháng 04** | - Hoàn thiện khóa luận các chương  - Tìm hiểu cách xây dựng khung ứng dụng | – Hoàn thiện khóa luận các chương  – Thảo luận về giao diện ứng dụng  – Viết khung chương trình | Hoàn thiện chức năng chương trình | Hoàn thiện chức năng chương trình | | **Tháng 05** | Hoàn thiện chức năng chương trình | Hoàn thiện chức năng chương trình | Hoàn thiện chức năng chương trình | Hoàn thiện chức năng chương trình | | **Tháng 06** | Hoàn thiện chức năng chương trình | – Báo cáo ứng dụng vào khóa luận  – Hoàn thiện lại khóa luận | – Làm video demo  – Slide thuyết trình  – Các tài liệu cần nộp | – Xem lại kiến thức  – Tập thuyết trình  – Nộp khóa luận | | |
| **Xác nhận của giáo viên hướng dẫn** | **Ngày … tháng … năm 2015**  Sinh viên thực hiện  Nguyễn Tấn Đô – Nguyễn Trương Trung Tín |

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc393392266)

[ĐỀ CƯƠNG CHI TIẾT 2](#_Toc393392267)

[MỤC LỤC 5](#_Toc393392268)

[DANH MỤC HÌNH 9](#_Toc393392269)

[DANH MỤC BẢNG 12](#_Toc393392270)

[TÓM TẮT KHÓA LUẬN 13](#_Toc393392271)

[CHƯƠNG 1. GIỚI THIỆU 14](#_Toc393392272)

[1.1. Giới thiệu đề tài. 14](#_Toc393392273)

[1.1.1. Dịch vụ hỗ trợ khách hàng của các doanh nghiệp công ty: 14](#_Toc393392274)

[1.1.2. Hệ thống hỗ trợ khách hàng zendesk 15](#_Toc393392275)

[1.1.3. Giới thiệu về ứng dụng nhiều người thuê. 16](#_Toc393392276)

[1.1.4. Khung ứng dụng SmartHelpDesk trong môi trường đám mây. 16](#_Toc393392277)

[1.2. Giải pháp cho vấn đề. 17](#_Toc393392278)

[1.2.1. Nội dung của ứng dụng: 18](#_Toc393392279)

[1.2.2. Công nghệ được lựa chọn: 22](#_Toc393392280)

[1.3. Mục tiêu luận văn 23](#_Toc393392281)

[1.4. Phạm vi đề tài. 23](#_Toc393392282)

[CHƯƠNG 2. LÝ THUYẾT NỀN TẢNG 24](#_Toc393392283)

[2.1. Điện toán đám mây 24](#_Toc393392285)

[2.1.1. Tổng quan về điện toán đám mây: 24](#_Toc393392286)

[2.1.2. So sánh ứng dụng truyền thống với ứng dụng đám mây: 26](#_Toc393392287)

[2.1.3. Chuyển đổi sang điện toán đám mây 31](#_Toc393392288)

[2.1.4. Khuyết điểm của điện toán đám mây: 37](#_Toc393392289)

[2.2. Ứng dụng nhiều người thuê 41](#_Toc393392290)

[2.2.1. Phân biệt giữa ứng dụng một người thuê và nhiều người thuê. 41](#_Toc393392291)

[2.2.2. Phát triển ứng dụng nhiều người thuê trong môi trường đám mây. 44](#_Toc393392292)

[2.3. Khung ứng dụng. 48](#_Toc393392293)

[2.3.1. Khung ứng dụng là gì. 48](#_Toc393392294)

[2.3.2. Tại sao phải sử dụng khung ứng dụng. 49](#_Toc393392295)

[2.3.3. Các khái niệm, kỹ thuật khi xây dựng một khung ứng dụng. 52](#_Toc393392296)

[2.4. Giải pháp 63](#_Toc393392297)

[2.4.1. Tổng quan về Amazon Web Service 63](#_Toc393392298)

[2.4.2. Cơ sở dữ liệu NoSQL - Amazon DynamoDB 80](#_Toc393392299)

[2.4.3. Công cụ phát triển 92](#_Toc393392300)

[CHƯƠNG 3. CÀI ĐẶT. 95](#_Toc393392301)

[3.1. Các chức năng của ứng dụng sẽ cài đặt. 95](#_Toc393392303)

[3.1.1. Module 1: Nhóm chức năng quản lý: 95](#_Toc393392304)

[3.1.2. Module 2: Nhóm chức năng tìm kiếm: 96](#_Toc393392305)

[3.1.3. Module 3: Nhóm chức năng thống kê – hiển thị: 96](#_Toc393392306)

[3.1.4. Module 4: Nhóm chức năng hệ thống: 97](#_Toc393392307)

[3.2. Sơ đồ usecase 97](#_Toc393392308)

[3.2.1. Danh sách actor: 101](#_Toc393392309)

[3.2.2. Danh sách usecase: 102](#_Toc393392310)

[3.3. Cài đặt khung ứng dụng SmartHelpDesk 105](#_Toc393392311)

[3.3.1. Đặc tả usecase khung ứng dụng: 105](#_Toc393392312)

[3.3.2. Cài đặt khung giao diện và ngôn ngữ cho ứng dụng: 106](#_Toc393392313)

[3.3.3. Hướng dẫn sử dụng khung ứng dụng SmartHelpDesk: 109](#_Toc393392314)

[3.4. Thiết kế dữ liệu Amazon DynamoDB 133](#_Toc393392315)

[3.4.1. Sơ đồ dữ liệu: 133](#_Toc393392316)

[3.4.2. Danh sách các bảng dữ liệu: 135](#_Toc393392317)

[3.5. Kiến trúc hệ thống. 137](#_Toc393392318)

[3.5.1. Kiến trúc theo mô hình 3 layer: 137](#_Toc393392319)

[3.5.2. Cây phân rã hệ thống: 138](#_Toc393392320)

[3.6. Giao diện. 139](#_Toc393392321)

[3.6.1. Danh sách các màn hình chính: 140](#_Toc393392322)

[3.6.2. Một số giao diện chính: 140](#_Toc393392323)

[CHƯƠNG 4. TỔNG KẾT ĐÁNH GIÁ 149](#_Toc393392324)

[4.1. Kiến thức đạt được 149](#_Toc393392326)

[4.2. Kết quả ứng dụng 149](#_Toc393392327)

[4.2.1. Môi trường phát triển ứng dụng: 149](#_Toc393392328)

[4.2.2. Những chức năng đã hoàn thành: 150](#_Toc393392329)

[4.2.3. Những chức năng chưa hoàn thành: 150](#_Toc393392330)

[4.2.4. Đánh giá: 151](#_Toc393392331)

[4.3. Những thuận lợi và khó khăn khi thực hiện đề tài này 151](#_Toc393392332)

[4.3.1. Thuận lợi: 151](#_Toc393392333)

[4.3.2. Khó khăn: 152](#_Toc393392334)

[4.4. Phương pháp nghiên cứu và phát triển đề tài trong tương lai 152](#_Toc393392335)

[4.4.1. Công nghệ: 152](#_Toc393392336)

[4.4.2. Ứng dụng: 152](#_Toc393392337)

[4.5. Lời kết 152](#_Toc393392338)

[TÀI LIỆU THAM KHẢO 154](#_Toc393392339)

[PHỤC LỤC 157](#_Toc393392340)

[Phụ lục 1. Lược đồ tuần tự của ứng dụng 157](#_Toc393392341)

[***1.1.*** ***Lược đồ tuần tự chức năng Tạo mới thông điệp:*** 157](#_Toc393392342)

[***1.2.*** ***Lược đồ tuần tự chức năng trả lời thông điệp:*** 162](#_Toc393392343)

[***1.3.*** ***Lược đồ tuần tự chức năng sửa trạng thái thông điệp:*** 163](#_Toc393392344)

[***1.4.*** ***Lược đồ tuần tự chức năng cập nhật vai trò người dùng:*** 163](#_Toc393392345)

[Phụ lục 2. Tạo chứng nhận định danh truy cập AWS bằng Amazon Cognito 164](#_Toc393392346)

[Phụ lục 3. Các chức năng ứng dụng sẽ cài đặt 165](#_Toc393392347)

[***3.1.*** ***Module 1: Nhóm chức năng quản lý:*** 165](#_Toc393392350)

[***3.2.*** ***Module 2: Nhóm chức năng tìm kiếm:*** 167](#_Toc393392351)

[***3.3.*** ***Module 3: Nhóm chức năng thống kê – hiển thị:*** 167](#_Toc393392352)

[***3.4.*** ***Module 4: Nhóm chức năng hệ thống:*** 168](#_Toc393392353)

DANH MỤC HÌNH

[*Hình 2-1. Mô hình tổng quan về điện toán đám mây 25*](#_Toc393392170)

[*Hình 2-2. Vòng lặp bảy bước di chuyển vào đám mây 34*](#_Toc393392171)

[*Hình 2-3. Mô tả kiến trúc logic một người thuê, nhiều thể hiện 42*](#_Toc393392172)

[*Hình 2-4. Mô tả kiến trúc logic nhiều người thuê, một thể hiện 43*](#_Toc393392173)

[*Hình 2-5. Tổng quan về mối quan hệ giữa ứng dụng và một khung ứng dụng 49*](#_Toc393392174)

[*Hình 2-6. So sánh giữa khung ứng dụng và lớp thư viện 52*](#_Toc393392175)

[*Hình 2-7. Mô tả nhiều tầng trong một ứng dụng 53*](#_Toc393392176)

[*Hình 2-8. Minh họa điểm phổ biến 56*](#_Toc393392177)

[*Hình 2-9. Những điểm nóng 57*](#_Toc393392178)

[*Hình 2-10. Khung hộp trắng 59*](#_Toc393392179)

[*Hình 2-11. Khung hộp đen 60*](#_Toc393392180)

[*Hình 2-12. Khung hộp xám 62*](#_Toc393392181)

[*Hình 2-11. Giao diện AWS Management Console 93*](#_Toc393392182)

[*Hình 3-1. Sơ đồ usecase tổng quát 98*](#_Toc393392183)

[*Hình 3-2. Sơ đồ usecase cho khách hàng 99*](#_Toc393392184)

[*Hình 3-3. Sơ đồ usecase cho quản trị viên hệ thống 99*](#_Toc393392185)

[*Hình 3-4. Sơ đồ usecase cho nhân viên hỗ trợ 100*](#_Toc393392186)

[*Hình 3-5. Sơ đồ usecase cho quản trị viên công ty 101*](#_Toc393392187)

[*Hình 3-6. Sơ đồ usecase cho lập trình viên công ty 101*](#_Toc393392188)

[*Hình 3-7. Sơ đồ lớp khung ngôn ngữ (Language Plugin) 106*](#_Toc393392189)

[*Hình 3-8. Sơ đồ tuần tự khung ngôn ngữ (Language Plugin) 107*](#_Toc393392190)

[*Hình 3-9. Sơ đồ lớp khung giao diện (Theme Plugin) 108*](#_Toc393392191)

[*Hình 3-10. Sơ đồ tuần tự khung giao diện (Theme Plugin) 109*](#_Toc393392192)

[*Hình 3-11. Bước 3 của thao tác import project vào eclipse 111*](#_Toc393392193)

[*Hình 3-12. Bước 4 của thao tác import project vào eclipse 112*](#_Toc393392194)

[*Hình 3-13. Bước 5 của thao tác import project vào eclipse 113*](#_Toc393392195)

[*Hình 3-14. Bước 6 của thao tác import project vào eclipse 114*](#_Toc393392196)

[*Hình 3-16. Màn hình kết quả 118*](#_Toc393392197)

[*Hình 3-17. Vị trí danh sách các màn hình và đặc tả cho từng màn hình 119*](#_Toc393392198)

[*Hình 3-18. Mô tả lớp ActivityLayoutId 119*](#_Toc393392199)

[*Hình 3-19. Mô tả Interface IThemePlugin 120*](#_Toc393392200)

[*Hình 3-20. Mô tả lớp ThemePlugin 120*](#_Toc393392201)

[*Hình 3-21. Mô tả cách thay đổi giao diện chính của màn hình Bookmark 128*](#_Toc393392202)

[*Hình 3-22. Mô tả thao tác thay đổi menu màn hình Views 129*](#_Toc393392203)

[*Hình 3-23. list\_item\_views\_detail\_header.xml first\_plugin\_list\_item\_views\_detail\_header.xml 130*](#_Toc393392204)

[*Hình 3-24. Màn hình kết quả 133*](#_Toc393392205)

[*Hình 3-25. Sơ đồ cơ sở dữ liệu 134*](#_Toc393392206)

[*Hình 3-26. Kiến trúc 3 layer 138*](#_Toc393392207)

[*Hình 3-27. Cây phân rã hệ thống 139*](#_Toc393392208)

[*Hình 3-28. Màn hình đăng nhập 141*](#_Toc393392209)

[*Hình 3-29. Màn hình đăng ký 142*](#_Toc393392210)

[*Hình 3-30. Màn hình chính 143*](#_Toc393392211)

[*Hình 3-31. Màn hình tạo mới phiếu thắc mắc. 145*](#_Toc393392212)

[*Hình 3-32. Màn hình danh sách phiếu thắc mắc. 146*](#_Toc393392213)

[*Hình 3-33. Màn hình nội dung phiếu thắc mắc. 147*](#_Toc393392214)

[*Hình 3-34. Màn hình nhập trả lời phiếu thắc mắc. 148*](#_Toc393392215)

[*Hình phụ lục 1-1. Lược đồ tuần tự chức năng tạo mới thông điệp 157*](#_Toc393392216)

[*Hình phụ lục 1-1-1. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 1) 158*](#_Toc393392217)

[*Hình phụ lục 1-1-2. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 2) 159*](#_Toc393392218)

[*Hình phụ lục 1-1-3. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 3) 160*](#_Toc393392219)

[*Hình phụ lục 1-1-4. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 4) 161*](#_Toc393392220)

[*Hình phụ lục 1-2. Lược đồ tuần tự chức năng trả lời thông điệp 162*](#_Toc393392221)

[*Hình phụ lục 1-3. Lược đồ tuần tự chức năng sửa trạng thái thông điệp 163*](#_Toc393392222)

[*Hình phụ lục 1-4. Lược đồ tuần tự chức năng cập nhật vai trò người dùng 164*](#_Toc393392223)

DANH MỤC BẢNG

[*Bảng 2-1. Chi tiết bảy bước di chuyển vào đám mây 33*](#_Toc393392247)

[*Bảng 2-2. Các API hỗ trợ bởi dịch vụ AWS 66*](#_Toc393392248)

[*Bảng 2-3. Hoạt động của những tài nguyên S3 với phương thức HTTP 72*](#_Toc393392249)

[*Bảng 2-4. Những ví dụ URI tài nguyên S3. 73*](#_Toc393392250)

[*Bảng 2-5. Những loại thực thể EC2 75*](#_Toc393392251)

[*Bảng 3-1. Nhóm chức năng quản lý 96*](#_Toc393392252)

[*Bảng 3-2. Nhóm chức năng tìm kiếm 96*](#_Toc393392253)

[*Bảng 3-3. Nhóm chức năng thống kê – hiển thị 97*](#_Toc393392254)

[*Bảng 3-4. Nhóm chức năng hệ thống 97*](#_Toc393392255)

[*Bảng 3-5. Bảng danh sách Actor 102*](#_Toc393392256)

[*Bảng 3-6. Bảng danh sách usecase 105*](#_Toc393392257)

[*Bảng 3-7. Đặc tả usecase khung ứng dụng 106*](#_Toc393392258)

[*Bảng 3-8. Danh sách các layout màn hình 126*](#_Toc393392259)

[*Bảng 3-9. Danh sách bảng cơ sở dữ liệu 137*](#_Toc393392260)

[*Bảng 3-10. Danh sách các màn hình 140*](#_Toc393392261)

[*Bảng phụ lục 3-1. Nhóm chức năng quản lý 167*](#_Toc393392262)

[*Bảng phụ lục 3-2. Nhóm chức năng tìm kiếm 167*](#_Toc393392263)

[*Bảng phụ lục 3-3. Nhóm chức năng thống kê – hiển thị 168*](#_Toc393392264)

[*Bảng phụ lục 3-4. Nhóm chức năng hệ thống 169*](#_Toc393392265)

TÓM TẮT KHÓA LUẬN

Nội dung luận văn trình bày về các kiến thức mà nhóm thực hiện luận văn đã tìm hiểu, các công cụ, kĩ thuật đã sử dụng để có thể hoàn thành luận văn này và xây dựng ứng dụng minh họa. Bao gồm:

* *Chương 1 – Giới thiệu:* trình bày vấn đề mà luận văn giải quyết, những khái niệm liên quan khi xây dựng một khung ứng dụng trên nền tảng đám mây*.*
* *Chương 2 – Lý thuyết nền tảng:* trình bày cái nhìn tổng quan về công nghệ điện toán đám mây, cái nhìn mô hình tổng quan, và những điểm đặc trưng. Tập trung vào phân biệt giữa ứng dụng điện toán đám mây và ứng dụng truyền thống, các bước chuyển đổi ứng dụng truyền thống từ môi trường vật lý cố định đến môi trường đám mây ảo hóa, và những khuyết điểm của điện toán đám mây còn gặp phải. Trình bày về giải pháp công nghệ Amazon Web Service, dịch vụ nền tảng do Amazon cung cấp và được nhóm lựa chọn để xây dựng ứng dụng. Bao gồm tổng quan về Amazon Web Service và vấn đề lưu trữ dữ liệu thông qua cơ sở dữ liệu NoSQL DynamonDB và giới thiệu các công cụ phát triển mà nhóm đã sử dụng.
* *Chương 3 – Cài đặt:* trình bày về các chức năng sẽ được cài đặt, cách thiết kế và giải pháp lưu trữ dữ liệu, thiết kế kiến trúc và một số màn hình giao diện của ứng dụng sau khi cài đặt.
* *Chương 4 – Tổng kết, đánh giá:* trình bày về các kết quả mà nhóm đạt được sau khi hoàn thành khóa luận và những định hướng phát triển trong tương lai.
* *Phụ lục:* trình bày lượt đồ tuần tự một vài chức năng chính của hệ thống, hướng dẫn tạo máy ảo xác thực token cho các request từ client (mobile…), hướng dẫn tạo plugin trong ứng dụng SmartHelpDesk.

CHƯƠNG 1. GIỚI THIỆU

* 1. Giới thiệu đề tài.
     1. Dịch vụ hỗ trợ khách hàng của các doanh nghiệp công ty:

Trong các hoạt động kinh doanh, dịch vụ khách hàng thường được nhắc đến mà theo đó người tiêu dùng thường đòi hỏi chất lượng dịch vụ khách hàng từ nhà sản xuất hoặc cung cấp dịch vụ. Để tồn tại được trong nền kinh tế đang cạnh tranh gay gắt như hiện nay thì dù công ty bạn đã có thâm niên hoạt động trong lĩnh vực chuyên môn của mình, bạn vẫn phải dành cho dịch vụ khách hàng với vị trí ưu tiên hàng đầu. Rất nhiều doanh nghiệp chi hàng triệu đô la cho việc quảng cáo để thu hút khách hàng nhưng rồi lại khiến họ bỏ đi chỉ vì không có dịch vụ chăm sóc khách hàng, hoặc vì chất lượng phục vụ quá tệ. Nếu muốn giữ chân khách hàng mà quảng cáo đã mang lại cho bạn, điều tối quan trọng là bạn phải cung cấp một dịch vụ khách hàng hoàn hảo.

Dịch vụ khách hàng không phải là chi phí mà là một món đầu tư có khả năng sinh lợi cao. Dịch vụ khách hàng hoàn hảo sẽ từng bước tập hợp được một lượng không nhỏ khách hàng trung thành, từ đó tạo ra lợi nhuận, bởi vì khi các khách hàng hài lòng, họ không chỉ mua nhiều hơn mà còn mua thường xuyên hơn. Theo một nghiên cứu do Hiệp hội Quản trị Hoa Kỳ tiến hành, các khách hàng trung thành tạo ra tới 65% doanh thu của một công ty.

Cơ sở khách hàng thường xuyên của bạn càng lớn, bạn sẽ càng tốn ít tiền cho quảng cáo và tiếp thị. Riêng về quảng cáo, có thể khẳng định rằng không một phương thức quảng cáo nào hiệu quả hơn “tiếng lành đồn xa”. Các khách hàng hài lòng về dịch vụ của bạn sẽ kể với những người khác về dịch vụ tuyệt vời mà bạn cung cấp, điều đó có giá trị và đáng tin cậy hơn bất cứ thứ gì bạn đăng hoặc phát triển các phương tiện thông tin đại chúng.

* + 1. Hệ thống hỗ trợ khách hàng zendesk

Cách giải quyết vấn đề về “dịch vụ khách hàng” của Zendesk là làm sao đề khách hàng có thể nhanh chóng giải quyết được thắc mắc của mình về công ty qua hệ thống Zendesk. Cách tiếp cận, giải quyết vấn đề của Zendesk là hệ thống cho phép khách hàng có thể tham gia vào quá trình tạo thắc mắc, và nhận được những phản hồi liên tục từ phía công ty. Zendesk cung cấp cách trao đổi dễ nhất và nhanh nhất giữa khách hàng và công ty đó là khách hàng có thể tạo thắc mắc bằng cách gửi mail qua địa chỉ email đã hổ trợ, hoặc tạo một thắc mắc qua trang mạng (website) hoặc điện thoại. Những thắc mắc sẽ được các nhân viên hổ trợ trả lời nhanh chóng và ngay lập tức. Từ đó việc giao tiếp giữa khách hàng trở nên đơn giản và thuận tiện. Khách hàng có thể cập nhật liên tục những thắc mắc qua hòm thư hoặc thông qua trang mạng hoặc thông qua điện thoại, điều này giúp khách hàng và công ty có thể trao đổi với nhanh mọi nơi mọi lúc và ở bất kì nơi đâu.

* Các công ty nào hiện đang sử dụng hệ thống Zendesk.

Hiện nay có hơn 40.000 công ty lớn và nhỏ đã và đang sử dụng hệ thống Zendesk, ví dụ như “vodafone”, “GILT”, “shopify”, “groupon”, “GOV.UK”. Zendesk hiện đang phục vụ cho hơn 300.000.000 người trên thế giới.

* Hệ thống Zendesk.

Zendesk là hệ thống cung cấp dịch vụ khách hàng dựa trên nền tảng đám mây được công ty Zendesk một công ty phát triền phần mềm có trụ sở tại San Francisco, California xây dựng. Hệ thống Zendesk bao gồm các phiếu thắc mắc (ticket), các mức độ tự phục vụ, các tính năng hổ trợ cho khách hàng.

Zendesk được viết trên nền tảng là Ruby on Rails và đáng chú ý ở đây là nó có thể tích hợp vào các “hệ thống quản lý nội dung”, các công cụ quản lý khách hàng và các ứng dụng web. Từ tháng 3/2013 Zendesk đã tiến hành đưa hệ thống lên nền tảng di động, hiện tại Zendesk đang hổ trợ cho các nền tảng sau web, android, ios, windownphone.

* + 1. Giới thiệu về ứng dụng nhiều người thuê.

Ứng dụng nhiều người thuê là ứng dụng mà sử dụng kiến trúc với một khởi tạo của ứng dụng có thể phục vụ cho nhiều người thuê. Người thuê có thể tùy chỉnh lại một một số điểm của ứng dụng như giao diện, một số tùy chỉnh về quy trình làm việc. Có thể mở rộng dữ liệu cho mỗi khách hàng nếu họ cần. Mỗi khách hàng sẽ có quyền truy cập nguồn tài nguyên dữ liệu của riêng mình, độc lập với những người thuê khác.

Khi lập trình một ứng dụng nhiều người thuê, nhà phát triển có thể tiết kiệm được chi phí hơn là một ứng dụng dành cho một người thuê. Bởi lẽ một ứng dụng thường phải trả một số tiền nhất định về việc thuê dung lượng bộ nhớ, xử lý. Ứng dụng nhiều người thuê sẽ giảm tải đáng kể những chi phí này. Hơn nữa như những chi phí như bản quyền bên dưới như hệ điều hành và hệ quản trị cơ sở dữ liệu bạn chỉ cần mua một lần cho những bản quyền này. Ngoài ra việc phát triển hệ thống ứng dụng nhiều người thuê là cực kì phức tạp vì dữ liệu của tất cả khách hàng điều được lưu trữ ở một cơ sở dữ liệu, và mỗi khách hàng sẽ có những yêu cầu về dữ liệu khác nhau và cần phải đảm bảo viêc truy cập dữ liệu giữa các khách hàng không được xảy ra, đảm bảo an toàn, bảo mật cho cơ sở dữ liệu. Tuy nhiên điều này lại thuận lợi không nhỏ là từ tập dữ liệu lớn này chúng ta có thể khai thác những dữ liệu này và tìm kiếm xu hướng của các công ty… thay vì phải tìm kiếm, thu thập dữ liệu từ những cơ sở dữ liệu khác nhau.

Ứng dụng nhiều người thuê là một trong những tính năng quan trọng của lĩnh vực điện toán đám mây.

* + 1. Xây dựng khung ứng dụng SmartHelpDesk trên trong môi trường đám mây.

Giới thiệu về ứng dụng SmartHelpDesk:

Qua tìm hiểu và nghiên cứu về hệ thống Zendesk, đây là một hệ thống lớn và được xây dựng trên nhiều nền tảng, và có rất nhiều chức năng cực kì phức tạp. Ứng dụng SmartHelpDesk của nhóm xây dựng là một ứng dụng mô phỏng một số chức năng chính và quan trọng của hệ thống Zendesk, được xây dựng trên nền tảng di động Android, cung cấp các tính năng hổ trợ cho khách hàng như có thể tạo thắc mắc, câu hỏi một các dễ dàng, và nhận được phản hồi từ bộ phận hổ trợ của công ty một cách nhanh chóng và dễ dàng nhất qua điện thoại di động. Ứng dụng SmartHelpDesk cũng giúp các nhân viên trong hệ thống có thể làm việc với nhau dễ dàng hơn.

Ứng dụng giúp xây dụng một hệ thống hổ trợ những tính năng tốt nhất, hiệu quả cao, giúp công ty có thể tiết kiệm chi phí, công sức để hổ trợ khách hàng tốt một cách tốt nhất. Hơn nữa ứng dụng có thể phục vụ cho nhiều công ty cùng lúc giúp nhà phát triển có thể tiết kiệm phi phí một cách hiệu quả nhất.

Khung ứng dụng Smart Help Desk trong môi trường đám mây:

Việc hổ trợ đa ngôn ngữ và thay đổi giao diện cho ứng dụng là một trong những yêu cầu không thể thiếu của một phần mềm, ứng dụng SmartHelpDesk cũng vậy. Ngoài việc đảm bảo các chức năng của hệ thống đã được xây dựng (sử dụng nền tảng đám mây Amazon Web Service), SmartHelpDesk còn cung cấp một khung ứng dụng (framework) hổ trợ các lập trình viên khác có thể bổ sung các giao diện theo tùy biến của mình, bổ sung các ngôn ngữ mới cho ứng dụng. Việc cung cấp các hàm, thư viện giúp các lập trình viên khác không cần phải hiểu mã nguồn đã được viết sẵn mà chỉ cần tập trung định nghĩa các giao diện mới cho ứng dụng, các ngôn ngữ mới cho ứng dụng. Từ đó có thể nhanh chóng thay đổi giao diện cho ứng dụng, đưa ra các giao diện mới cho ứng dụng.

* 1. Giải pháp cho vấn đề.

Phần này trình bày phương pháp chung để giải quyết vấn đề được đưa ra ở mục, bao gồm xác định nội dung của ứng dụng sẽ xây dựng và việc lựa chọn công nghệ điện toán đám mây để xây dựng ứng dụng.

* + 1. Nội dung của ứng dụng:

Giới thiệu tổng quát các vai trò người dùng trong hệ thống:

*Khách hàng (Customer):* Khi khách hàng có thắc mắc cần sự trợ giúp, SmartHelpDesk sẽ hổ trợ khách hàng tạo nhanh một phiếu thắc mắc thông qua email hoặc thông qua giao diện trên điện thoại. Từ đó khách hàng có thể trao đổi và nhận sự hổ trợ từ công ty, các phản hồi sẽ liên tục được cập nhật có thể xem qua hòm thư hoặc qua điện thoại.

*Nhân viên hỗ trợ (Agent):* SmartHelpDesk sẽ cung cấp các công cụ giúp chăm sóc khách hàng một cách tự động, các nhân viên hổ trợ sẽ được phân quyền giúp cho những người làm việc trong công ty có thể trao đổi, giám sát công việc để tăng hiệu quả việc trợ giúp khách hàng. Hệ thống cũng sẽ sắp xếp hợp lý để quản lý những nội dung hỗ trợ cho khách hàng.

*Quản trị viên công ty (Admin):* SmartHelpDesk hỗ trợ những quyền lợi đặc biệt giúp cho Quản trị viên của công ty có thể quản lý được những nhân viên hỗ trợ của mình và cũng có thể hỗ trợ được những vấn đề mà người dùng mắc phải.

*Quản trị viên của hệ thống SmartHelpDesk:*Những người quản trị viên này có nhiệm vụ quản lý những công ty được tạo ra trong SmartHelpDesk.

Mối quan hệ - vai trò của từng người dùng trong hệ thống.

* + - * + **Người quản trị viên của công ty (administrator).**

SmartHelpDesk xây dựng cung cấp cho người quản trị của công ty những quyền để có thể quản lý những bộ phận trong công ty mình cũng như quản lý việc hỗ trợ cho khách hàng.

Quản trị viên có thể quản lý những nhân viên hỗ trợ của công ty, họ có thể thêm nhân viên hỗ trợ, xóa nhân viên hỗ trợ, hoặc sửa thông tin cá nhân của nhân viên hỗ trợ nào đó. Bên cạnh đó quản trị viên cũng có thể nâng cấp một nhân viên hỗ trợ lên thành quản trị viên nếu như họ thực hiện tốt công việc, hoặc có thể giảm cấp họ xuống thành khác hàng nếu như họ không còn là nhân viên hỗ trợ của công ty nữa.Với vai trò là quản trị, quản trị viên có thể chỉ định nhân viên hỗ trợ nào đó trả lời thắc mắc những chủ đề do khách hàng yêu cầu cần được giải đáp.

SmartHelpDesk cung cấp cho quản trị viên một tính năng rất hay đó là quản trị viên có thể tạo ra được những tổ chức (Organization), mỗi công ty được tạo ra trên SmartHelpDesk thì có thể có nhiều tổ chức trong đó. Quản trị viên chính là người trực tiếp quản lý những tổ chức này. Các công việc như thêm, xóa và sửa thông tin cá nhân cho những tổ chức là những thao tác cần thiết để hiện thực những đặc quyền trên. Những người dùng trong công ty sẽ được phân loại và đưa vào tổ chức tương ứng theo cách mà quản trị viên thiết lập. Ví dụ: Tổ chức A gồm những người dùng (khách hàng, quản trị viên, nhân viên hổ trợ) có những tên miền của email như “gmail.com yahoo.com” sẽ được phân loại vào tổ chức A. Việc chia nhỏ thành thành những tổ chức nhỏ hơn, thuận tiện cho việc giải quyết các vấn đề nếu khách hàng gặp phải, những câu hỏi của họ sẽ đến đúng bộ phận hơn, và họ sẽ nhận được những câu trả lời nhanh hơn.

Ngoài ra, quản trị viên có thể quản lý cả những khách hàng, họ có thể thêm người dùng, xóa người dùng và sửa thông tin cá nhân người dùng. Và có một chức năng đặc biệt là họ có thể nâng cấp khách hàng thành nhân viên hỗ trợ, hoặc quản trị viên nếu như khách hàng đó chính là nhân viên của công ty và có những góp ý tích cực cho sự phát triển của công ty.

Bên cạnh tổ chức, thì quản trị viên còn đảm nhận nhiệm vụ quản lý những nhóm - những bộ phận hỗ trợ khách hàng (Group). Những bộ phận này sẽ bao gồm những nhân viên hỗ trợ, sẽ có thể có rất nhiều nhóm trong công ty, số lượng nhóm sẽ không có giới hạn. Như những công ty thực tế ở ngoài, quản trị viên sẽ đặt những nhân viên hỗ trợ vào đúng với bộ phận tương ứng của họ, việc này cũng sẽ giúp phân chia công ty ra thành nhiều khâu làm tăng hiệu quả của việc giải đáp thắc mắc.

Những tổ chức trong công ty có thể được liên kết đến một và chỉ một bộ phận hỗ trợ, và một nhóm lại có thể liên kết với nhiều tổ chức khác nhau. Khi khách hàng gửi yêu cầu đến tổ chức đó thì yêu cầu đó sẽ được liên kết và gửi ngay đến nhóm đã được tương tác bởi quản trị viên.

Quản trị viên cũng có thể quản lý, tư vấn và thao tác trên những phiếu thắc mắc nhưng hầu như nhiệm vụ chính này thuộc về những nhân viên hỗ trợ của công ty (agent).

* + - * + **Nhân viên hổ trợ của công ty (agent).**

SmartHelpDesk xây dựng những quyền của nhân viên hỗ trợ được quy định bởi những người quản trị viên. Nhân viên hỗ trợ được tạo ra với nhiệm vụ là những người giải đáp những vấn đề của khách hàng gặp phải.

Nhân viên hỗ trợ cũng có thể quản lý được khách hàng, họ có thể thêm khách hàng, xóa khách hàng, sửa thông tin cá nhân của khách hàng nhưng họ không thể nâng cấp khách hàng lên thành nhân viên hỗ trợ được.

Những người quản lý này có thể thao tác trên những phiếu thắc mắc hay còn gọi là những phiếu thắc mắc cần được giải đáp của khách hàng, họ sẽ tư vấn, trả lời những phiếu thắc mắc có sẵn, những phiếu thắc mắc được quản trị viên tạo ra, những phiếu thắc mắc được gửi đến từ tổ chức.... Họ cũng có thể thay đổi những trạng thái của phiếu thắc mắc, ví dụ như phiếu thắc mắc này mới tạo, phiếu thắc mắc này đã được giải quyết, phiếu thắc mắc này đang chờ xử lý ... Nhìn chung công việc của nhân viên hỗ trợ chính là trợ giúp, hỗ trợ những yêu cầu từ khách hàng

* + - * + **Khách hàng (customer).**

Đây là đối tượng chính mà SmartHelpDesk hướng tới, những công cụ, những chức năng được làm ra để tối ưu việc chăm sóc đối tượng này.

Khách hàng thường có những vấn đề thắc mắc, những câu hỏi cần lời giải, thì SmartHelpDesk chính là cầu nối giữa khách hàng và những người quản trị, quản lý của công ty. Khi họ có thắc mắc gì đó, họ có thể gửi mail trực tiếp đến bộ phận hỗ trợ của công ty, và mail này sẽ chuyển thành những phiếu thắc mắc trong giao diện quản lý của những người hỗ trợ. Những người này sẽ xem và tư vấn lại cho khách hàng.

Ngoài ra, khi người quản trị viên hoặc nhân viên hổ trợ tạo ra một tài khoản khách hàng nào đó vào trong hệ thống công ty thì khách hàng đó có thể đăng nhập vào công ty và gửi những thắc mắc của mình lên trong chính giao diện đó (không cần phải qua mail).

Khách hàng chỉ có thể quản lý được phiếu thắc mắc của mình như là tạo một phiếu thắc mắc, tìm kiếm phiếu thắc mắc hay lọc phiếu thắc mắc nào đó trong danh sách phiếu thắc mắc của chính họ. Ngoài ra, họ cũng có thể quản lý thông tin cá nhân của họ, nhưng không thể thay đổi địa chỉ email (đây là thông tin đăng nhập được những người hỗ trợ gắn cho họ khi kết nối vào hệ thống công ty).

* + - * + **Quản trị viên của hệ thống SmartHelpDesk.**

Những người quản trị viên này có nhiệm vụ quản lý những công ty được tạo ra trong SmartHelpDesk.

Họ có quyền thêm công ty, xóa công ty, chỉnh sửa nội dung công ty.... Mỗi công ty khi đăng ký vào hệ thống sẽ chọn gói cước sử dụng của mình: gói miễn phí, gói tính theo tháng.... Người quản trị viên này sẽ có nhiệm vụ quản lý gói cước đó, họ có thể nâng cấp gói cước sử dụng của công ty nếu như công ty trả tiền hoặc có thể chuyển gói cước của công ty thành miễn phí nếu họ không đóng tiền. Mỗi mức độ của gói cước sẽ khác nhau về bộ nhớ lưu trữ cũng như những chức năng có trong hệ thống.

* + 1. Công nghệ được lựa chọn:

Nền tảng dạng dịch vụ (Platform as a Service – PaaS) giúp tóm gọn sự phức tạp của phần mềm trung gian và cơ sở hạ tầng, cho phép nhà phát triển viết mã nguồn nhanh hơn bằng cách sử dụng các nền tảng (framework) và dịch vụ (service) mà họ cần. Đối với nhà khai thác, nền tảng dạng dịch vụ làm giảm chi phí hoạt động từ việc quản lý và mở rộng kiến trúc ứng dụng. Ngoài ra, nền tảng dạng dịch vụ còn giảm chi phí đầu tư ban đầu như chi phí phần cứng và bản quyền phần mềm.

Với nền tảng dạng dịch vụ, xây dựng các ứng dụng trên một cơ sở hạ tầng ảo hóa cho phép nhà phát triển có thể tải và chạy một dự án trong vòng vài phút và hoàn thành nó trong vài ngày thay vì phải mất vài tuần đến vài tháng. Điều này phù hợp với yêu cầu của khách hàng doanh nghiệp: “nhanh hơn, nhanh hơn, nhanh hơn nữa”.

Từ những vẫn đề đã đặt ra nhóm đã lựa chọn việc xây dựng ứng dụng dựa trên dịch vụ nền tảng của amazon cung cấp. Với việc lưu trữ dữ liệu nhóm chọn Amazon DynamoDB để lưu trữ dữ liệu. DynamoDB là một cơ sở dữ liệu NoSQL đáp ứng cho việc phát triển xây dựng một ứng dụng nhiều người thuê với yêu cầu dữ liệu lớn, hiệu suất cao, khả năng mở rộng liền mạch. DynamoDB cho phép khách hàng để giảm tải gánh nặng hành chính hoạt động và mở rộng quy mô cơ sở dữ liệu phân phối cho AWS, chúng ta không phải lo lắng về trích lập dự phòng phần cứng, cài đặt và cấu hình, nhân rộng, phần mềm vá, hoặc cụm rộng.

Chúng em xử dụng “Native App Android” để hiện thực hóa các chức năng của ứng dụng, cũng như xây dựng khung ứng dụng cho phép lập trình viên khác có thể bổ sung và chỉnh sửa giao diện và ngôn ngữ cho ứng dụng.

* 1. Mục tiêu luận văn

Sau đây là một số mục tiêu đặt ra khi thực hiện đề tài:

* Hiểu rõ về công nghệ điện toán đám mây, có cái nhìn mô hình tổng quan và các đặc trưng của điện toán đám mây.
* Hiểu được sự khác biệt giữa ứng dụng điện toán đám mây với ứng dụng truyền thống. Nắm được một số cách chuyển đổi ứng dụng doanh nghiệp từ một môi trường vật lý cố định đến một môi trường đám mây ảo hóa.
* Những khuyết điểm của điện toán đám mây còn gặp phải.
* Phát triển và triển khai một khung ứng dụng trên một dịch vụ điện toán đám mây (SmartHelpDesk) cho phép lập trình viên khác có thể bổ sung và chỉnh sửa giao diện và ngôn ngữ cho ứng dụng.
* Nâng cao kỹ năng tự nghiên cứu, khả năng đọc tài liệu tiếng Anh chuyên ngành công nghệ thông tin để tìm hiểu một công nghệ mới.
* Cải thiện kĩ năng làm việc nhóm, làm việc theo qui trình, khả năng trình bày và giải quyết vấn đề.
  1. Phạm vi đề tài.

Số lượng nhà cung cấp dịch vụ đám mây khá phong phú, và mỗi nhà dịch vụ đều có những hỗ trợ dịch vụ và công nghệ phát triển khác nhau, trong phạm vi của luận văn, chỉ đi sâu vào dịch vụ đám mây của Amazon, còn các nhà cung cấp dịch vụ đám mây khác như Microsoft, Google... thì chỉ ở mức khảo sát đánh giá.

Hệ thống SmartHelpDesk là một hệ thống rất lớn có nhiều chức năng phức tạp. Việc xây xậy một hệ thống như vậy là một việc rất khó khăn, cần nhiều thời gian và công sức. Hơn nữa ứng dụng xây dựng trên nền tảng di động nên chúng em chỉ tập trung vào những chức năng chính và xây dựng khung ứng dụng cho ứng dụng.

CHƯƠNG 2. LÝ THUYẾT NỀN TẢNG

1. 1. Điện toán đám mây
      1. Tổng quan về điện toán đám mây:

Định nghĩa:

Hiện có rất nhiều định nghĩa về khái niệm ‘điện toán đám mây’:

*“Điện toán đám mây là một hệ thống tính toán song song và phân tán bao gồm một tập hợp các liên-kết-nối và các máy tính tự động phục vụ như một hoặc nhiều tài nguyên tính toán thống nhất dựa trên các thỏa thuận cấp độ dịch vụ SLA (Service-level agreements) được lập ra giữa các nhà cung cấp dịch vụ và người tiêu dùng.”*

Giáo sư Rajkumar Buyya (đồng Chủ tịch của IEEE Task Force về Cluster Computing)

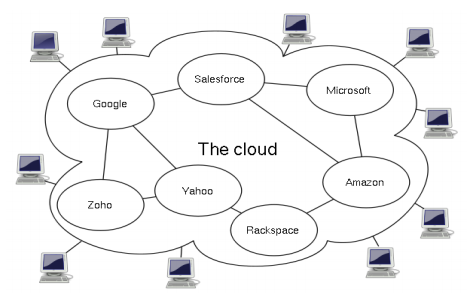
*“Điện toán đám mây là một mô hình cho phép truy cập mạng thuận tiện, theo nhu cầu đến một kho tài nguyên điện toán dùng chung, có thể định cấu hình (ví dụ như mạng, máy chủ, lưu trữ, ứng dụng), có thể được cung cấp và thu hồi một cách nhanh chóng với yêu cầu tối thiểu về quản lý hoặc can thiệp của nhà cung cấp dịch vụ.”*

Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST)

Tuy nhiên, ta có thể rút ra các đặc điểm mà một ‘đám mây’ nên có như: trả phí theo mức độ sử dụng, tính linh động cao và nguồn tài nguyên được xem như là vô tận, có giao diện tự phục vụ, nguồn tài nguyên là ảo hóa. Ngoài việc tính toán và lưu trữ, điện toán đám mây còn cung cấp các dịch vụ phần mềm, bao gồm các giao diện lập trình ứng dụng API và các công cụ phát triển, tất cả vì một mục tiêu tối thượng là làm sao cho người dùng chạy hệ thống Công nghệ thông tin của họ mỗi ngày trên ‘mây’ [1].

Mô hình tổng quan:

Theo định nghĩa thì điện toán đám mây là một mô hình mà các nguồn tài nguyên được chia sẻ. Tất cả các nguồn điện toán khổng lồ như phần mềm, dịch vụ sẽ nằm trên các máy chủ ảo trên internet thay vì trong máy tính cá nhân và văn phòng để mọi người ở bất cứ nơi đâu cũng có thể kết nối và sử dụng khi cần.



Hình 2-1. Mô hình tổng quan về điện toán đám mây

*(Nguồn: [1])*

Các điểm đặc trưng:

Đây là những đặc trưng của mô hình điện toán đám mây, các mô hình dịch vụ không có đủ các đặc điểm này thì chưa được coi là mô hình điện toán đám mây (theo định nghĩa của NIST): [2]

* **Tự phục vụ theo nhu cầu**: Người tiêu dùng có thể tự quy định, tự cung cấp năng lực tính toán.
* **Truy cập qua mạng**: các tài nguyên dùng chung có thể được truy cập qua mạng LAN hoặc mạng Internet, tùy vào cách mà tài nguyên đó được triển khai.
* **Kho tài nguyên dùng chung**: Tất cả những tài nguyên tính toán của nhà cung cấp dịch vụ sẽ được ảo hóa thành một kho tài nguyên dùng chung phục vụ cho nhiều người tiêu dùng theo mô hình đa người thuê (multi-tenant model).
* **Khả năng co dãn nhanh chóng**: Năng lực tính toán có thể được cấp quyền rất nhanh và có khả năng đàn hồi (elastically), đôi khi được thực hiện tự động. Điều này nhằm mục đích để năng lực tính toán có thể nhanh chóng co dãn, theo hướng mở rộng thêm hoặc giảm bớt.
* **Đo lường dịch vụ**: Những hệ thống đám mây sẽ tự động điều khiển và tối ưu hóa nguồn tài nguyên sử dụng bằng khả năng đo lường chính xác của mỗi loại dịch vụ.
  + 1. So sánh ứng dụng truyền thống với ứng dụng đám mây:

Phần này chúng em tập trung vào việc so sánh và phân biệt ứng dụng truyền thống với ứng dụng xây dựng trên đám mây công cộng thông qua sử dụng dịch vụ nền tảng. Chúng em sẽ không so sánh ứng dụng truyền thống với toàn bộ các mô hình đám mây khác.

Về định nghĩa:

Ứng dụng truyền thống là ứng dụng được triển khai trên máy chủ riêng của doanh nghiệp hoặc triển khai trên máy chủ được thuê của nhà cung cấp dịch vụ máy chủ. Trong ba thập kỷ qua, mô hình doanh nghiệp tự quản lý máy chủ của họ được ưa chuộng và tin dùng. Mô hình này cho phép doanh nghiệp toàn quyền kiểm soát cơ sở hạ tầng của họ. Doanh nghiệp có thể nhìn thấy, chạm vào máy chủ, vì chúng được đặt trong các trung tâm dữ liệu của doanh nghiệp. Ứng dụng của doanh nghiệp được triển khai, bảo trì và quản lý bởi đội ngũ quản trị viên của doanh nghiệp. [3]

Ngược lại, ứng dụng được triển khai trên đám mây hay trên máy chủ thuê của nhà cung cấp dịch vụ thì cơ sở hạ tầng không thuộc quyền quản lý của doanh nghiệp. Họ thuê quyền sử dụng từ nhà cung cấp và trả phí. Điểm khác nhau cơ bản giữa hai mô hình này chính là ở những đặc điểm của đám mây. Ứng dụng triển khai trên đám mây sẽ không trả phí theo gói cước như ứng dụng thuê máy chủ, nó trả phí dựa trên những tài nguyên mà ứng dụng đám mây đã sử dụng. Khi cần thiết, ứng dụng đám mây có thể co giãn năng lực xử lý và lưu trữ để tối ưu cho ứng dụng. Ngược lại ứng dụng được triển khai trên máy chủ thuê của nhà cung cấp thì chỉ được sử dụng tối đa tài nguyên mà gói thuê bao quy định. [3]

Về cách xây dựng ứng dụng:

Về cơ bản việc xây dựng ứng dụng trên ba môi trường này đều giống nhau. Lập trình viên xác định yêu cầu, danh mục chức năng cần thực hiện. Tùy theo hướng phát triển, mô hình phát triển và công nghệ phát triển mà lập trình viên sẽ thực hiện lập kế hoạch, thiết kế, viết mã nguồn và kiểm thử.

Đám mây là một môi trường không có trạng thái (Stateless), mỗi phiên làm việc (session) của người dùng có thể sử dụng một hoặc nhiều máy chủ đám mây khác nhau, vì thế lưu giữ trạng thái phiên làm việc của người dùng trên bộ nhớ của máy chủ theo cách truyền thống là không khả thi. Nhà thiết kế cần lưu ý việc này để có biện pháp lưu trữ phiên làm việc tại nơi khác. [4]

Khả năng tự co giãn năng lực xử lý của đám mây là một ưu điểm vượt trội của ứng dụng trên đám mây so với ứng dụng truyền thống. Xây dựng ứng dụng dựa trên đám mây mà không cần tự động mở rộng quy mô thì tương tự như việc xây dựng ứng dụng và thuê máy chủ để triển khai. Nhiều nhà cung cấp dịch vụ đám mây như Amazon's EC2 hay Windows Azure không cung cấp tính năng tự động mở rộng quy mô một cách mặc định. Thay vào đó, họ dựa vào các công cụ và dịch vụ khác nhau để cung cấp tính năng mở rộng quy mô một cách năng động. Vì vậy thiết kế và xây dựng ứng dụng trên đám mây cũng cần phải xây dựng chiến lược tự động mở rộng quy mô xử lý cho phù hợp. [5]

Về cách triển khai ứng dụng:

Để triển khai ứng dụng lên máy chủ riêng của doanh nghiệp, cần phải cài đặt lên máy chủ những nền tảng (framework) mà ứng dụng cần để chạy. Tạo cơ sở dữ liệu nếu có, triển khai mã nguồn đã biên dịch của ứng dụng và các tài nguyên mà ứng dụng cần. Cấu hình máy chủ, NAT port, thiết lập tường lửa, chính sách bảo mật cho ứng dụng được dùng công cộng, hoặc cấu hình cho ứng dụng sử dụng nội bộ.

Triển khai ứng dụng lên máy chủ thuê hay lên đám mây tương tự như trên máy chủ riêng ở công đoạn tạo cơ sở dữ liệu nếu có, triển khai mã nguồn đã biên dịch của ứng dụng và các tài nguyên mà ứng dụng cần. Tuy nhiên, có một chút khác biệt so với triển khai ứng dụng lên máy chủ riêng. Một số nhà cung cấp dịch vụ thuê máy chủ hoặc đám mây không cho phép cài đặt thêm các nền tảng khác ngoài các nền tảng hỗ trợ sẵn, khi đó ta phải lựa chọn nhà cung cấp nào hỗ trợ nền tảng mà ứng dụng cần để chạy. Những cấu hình về an ninh của máy chủ (tường lửa, chính sách bảo mật) thường do nhà cung cấp thực hiện và tuân theo chính sách của nhà cung cấp dịch vụ.

Đối với đám mây, cần phải cấu hình ứng dụng: số lượng bản sao của ứng dụng sẽ chạy, vùng địa lý mà ứng dụng sẽ chạy, cấu hình những dịch vụ mà ứng dụng sử dụng (ví dụ dịch vụ phân phối nội dung - CDN trên Windows Azure).

Về cách quản lý ứng dụng:

Đối với máy chủ riêng, doanh nghiệp toàn quyền quản lý tất cả mọi thứ của máy chủ như không gian lưu trữ máy chủ, điện, làm mát, phần cứng, sao lưu, phần mềm, hệ điều hành, an ninh, kế hoạch khắc phục các sự cố, bảo trì, nâng cấp, cập nhật ứng dụng hoặc phần cứng.

Đối với máy chủ thuê hay đám mây, việc quản lý máy chủ sẽ do nhà cung cấp dịch vụ phụ trách. Việc quản lý của khách hàng với hệ điều hành, tường lửa, chính sách bảo mật tùy thuộc vào sự hỗ trợ (các thuê bao máy chủ cao cấp thường hỗ trợ nhiều chính sách hơn các thuê bao giá rẻ) hoặc cung cấp như các dịch vụ (ví dụ nền tảng Windows Azure hỗ trợ người dùng triển khai máy ảo Windows Server 2008 R2 để tùy chỉnh hệ điều hành cho ứng dụng, cung cấp dịch vụ kết nối ứng dụng - Windows Azure AppFabric để tăng chính sách bảo mật cho ứng dụng).

Khách hàng toàn quyền quản lý dữ liệu và ứng dụng của mình (sao lưu, phục hồi dữ liệu, xem ứng dụng đã sử dụng bao nhiêu tài nguyên băng thông, lưu trữ, xử lý v.v) trên cả ba môi trường triển khai (máy chủ riêng, máy chủ thuê, đám mây).

Đối với ứng dụng triển khai trên máy chủ riêng hay trên máy chủ thuê nếu doanh nghiệp muốn cập nhật, nâng cấp mở rộng ứng dụng, thì họ cần phải tạm ngưng ứng dụng, sau đó tái triển khai ứng dụng mới (nếu muốn cập nhật) hay cần ráp thêm phần cứng (mở rộng quy mô máy chủ riêng) hoặc liên hệ nhà cung cấp để mua gói thuê bao mới (mở rộng quy mô máy chủ thuê). Điều này khiến cho ứng dụng không sẵn sàng để phục vụ người dùng. Trong khi đó trên đám mây ứng dụng có thể tự động mở rộng quy mô và khi cập nhật ứng dụng thì nó không cần phải tạm ngưng mà vẫn có thể phục vụ người dùng với phiên bản trước đó.

Ưu và khuyết điểm của ứng dụng đám mây và ứng dụng truyền thống

Đối với ứng dụng triển khai trên máy chủ riêng, ưu điểm quan trọng nhất là doanh nghiệp toàn quyền quản lý cơ sở hạ tầng để phục vụ nhu cầu của doanh nghiệp. Toàn quyền áp dụng mọi phương pháp bảo mật cũng như mọi chính sách mà bản thân doanh nghiệp tin tưởng được. Những ứng dụng sử dụng nội bộ với dữ liệu quan trọng sẽ nằm trong tầm kiểm soát của doanh nghiệp và không phơi bày ra bên ngoài từ đó an toàn và bảo mật hơn. Việc tự quản lý hạ tầng của mình cũng khiến cho doanh nghiệp không phụ thuộc vào nhà cung cấp dịch vụ, tăng tính độc lập của doanh nghiệp.

Ngược lại, khuyết điểm của của ứng dụng triển khai trên máy chủ thuê hay trên đám mây chính là phụ thuộc vào nhà cung cấp dịch vụ, doanh nghiệp không thể toàn quyền kiểm soát hạ tầng ứng dụng. Phụ thuộc vào mạng lưới toàn cầu. Nếu không có mạng lưới toàn cầu thì ứng dụng có thể tê liệt, dữ liệu quan trọng khi lưu trữ và truyền qua mạng dễ dàng bị xâm phạm.

Ứng dụng triển khai trên máy chủ riêng rõ ràng chủ động hơn về mặt quản lý tuy nhiên điều đó cũng khiến nó cần chi phí cao trong việc đầu tư máy chủ, bản quyền phần mềm, duy trì, bảo trì, quản lý máy chủ, thậm chí phải cần thêm một đội ngũ nhân viên kỹ thuật để quản trị hệ thống. Điều này lại là ưu điểm của máy chủ thuê và đám mây, việc xây dựng hệ thống với máy chủ thuê và đám mây Tiết kiệm chi phí đầu tư ban đầu cho hệ thống, nhân lực. Không phải lo lắng về việc duy trì, bảo trì và quản trị hệ thống từ đó tiết kiệm chi phí kỹ thuật, tập trung vào công việc kinh doanh.

Đối với máy chủ riêng hay máy chủ thuê, trong trường hợp cần mở rộng hoặc cập nhật ứng dụng thì có thể phải tạm ngừng ứng dụng. Doanh nghiệp có thể không sử dụng hết toàn bộ tài nguyên của mình dẫn đến lãng phí. Nếu quá nhiều người truy cập sẽ khiến hệ thống bị quá tải. Ngược lại, ứng dụng trên đám mây có tính sẵn sàng cao. Có thể tự động tăng giảm quy mô một cách nhanh chóng, dễ dàng xây dựng ứng dụng phân tán mang lại tốc độ tối ưu nếu nó được sử dụng bởi nhiều người dùng ở khắp nơi trên thế giới, khi nâng cấp không cần phải ngừng ứng dụng, dễ dàng để xem thống kê tài nguyên mà ứng dụng sử dụng và chỉ phải trả tiền cho những gì mà ứng dụng tiêu thụ.

* + 1. Chuyển đổi sang điện toán đám mây

*Phần này nhắm đến các kiến trúc sư* điện *toán đám mây đang chuẩn bị để di chuyển một ứng dụng doanh nghiệp từ một môi trường vật lý cố định đến một môi trường đám mây ảo hóa.*

Chuyển đổi ứng dụng vào đám mây:

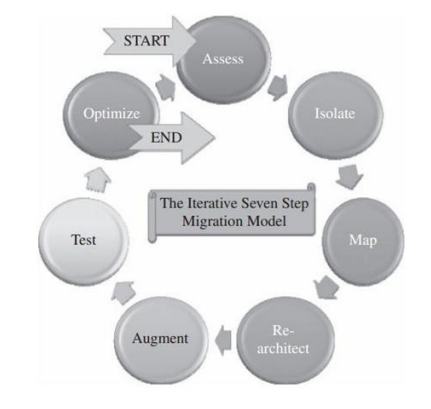
Di chuyển một ứng dụng vào các đám mây có thể xảy ra ở một trong nhiều cách: Hoặc là ứng dụng sạch sẽ và độc lập, do đó, nó chạy như nó vẫn chạy, hoặc một số mức độ của mã cần phải được sửa đổi và thích nghi. Tóm lại, việc di chuyển có thể xảy ra tại một trong năm cấp độ của ứng dụng, mã, thiết kế, kiến trúc và sử dụng.Sau đây là mô hình tham khảo bảy bước để di chuyển một ứng dụng vào trong đám mây [6].

*Mô hình bảy bước di chuyển vào đám mây:*

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Tên bước** | **Chi tiết** |
| Bước 1 | **Đánh giá** | * Lợi ích kinh tế từ việc tận dụng các tính năng của điện toán đám mây * Chi phí di cư * Chi phí tái diễn * Các phân khúc dữ liệu cơ sở dữ liệu * Di cư cơ sở dữ liệu * Di cư chức năng * Hỗ trợ các yêu cầu phi chức năng |
| Bước 2 | **Cô lập** | * Môi trường thực thi theo thời gian thực * Giấy phép * Thư viện phụ thuộc * Các ứng dụng phụ thuộc * Độ trễ nút cổ chai * Hiệu suất nút cổ chai * Kiến trúc phụ thuộc |
| Bước 3 | **Ánh xạ** | * Ánh xạ thông điệp: theo thứ tự xuôi & ngược * Môi trường ánh xạ * Thư viện ánh xạ & thời gian thực thi xấp xỉ |
| Bước 4 | **Tái kiến trúc** | * Ước tính chức năng bị mất bằng cách sử dụng đám mây thực thi theo thời gian thực hỗ trợ giao diện lập trình ứng dụng API * Use-case mới * Phân tích * Thiết kế |
| Bước 5 | **Củng cố - Tăng cường** | * Khai thác thêm các tính năng điện toán đám mây * Tìm kiếm sự tăng cường chi phí thấp * Tự động mở rộng * Lưu trữ * Băng thông * Bảo mật |
| Bước 6 | **Kiểm thử** | * Cải thiện các trường hợp kiểm thử và tự động hóa thử nghiệm * Chạy bằng chứng khái niệm * Kiểm thử chiến lược di cư * Kiểm thử các trường hợp củng cố đám mây * Kiểm thử vể tái sản xuất |
| Bước 7 | **Tối ưu hóa** | * Tối ưu hóa, lặp và lặp * Tận dụng hầu hết các tính năng của đám mây cho việc di cư * Tối ưu hóa phù hợp với tiêu chuẩn và quản trị * Cung cấp sự di cư có hiệu quả đầu tư ROI tốt nhất * Xây dựng lộ trình để tận dụng các tính năng điện toán đám mây mới |

Bảng 2-1. Chi tiết bảy bước di chuyển vào đám mây

Mô hình bảy bước di chuyển vào đám mây là một phần của những nỗ lực từ sự hiểu biết và tận dụng các dịch vụ điện toán đám mây trong bối cảnh doanh nghiệp [6]. Bảng 2.1. thể hiện các bước trong mô hình di chuyển vào các đám mây, trong khi hình 2.4 là sự lặp đi lặp lại quá trình di chuyển bảy bước vào đám mây.



Hình 2-2. Vòng lặp bảy bước di chuyển vào đám mây

*(Nguồn: [6])*

* Đánh giá sự di chuyển vào đám mây bao gồm: đánh giá để hiểu các vấn đề liên quan ở cấp độ ứng dụng hoặc mã, thiết kế, kiến trúc, hoặc mức độ sử dụng. Bước đầu tiên của quá trình lặp đi lặp lại của mô hình bảy bước di chuyển về cơ bản là ở cấp độ đánh giá.
* Bước tiếp theo là cô lập tất cả hệ thống phụ thuộc vào môi trường của các thành phần ứng dụng doanh nghiệp trong trung tâm dữ liệu tập trung. Sau khi cô lập hoàn tất, quay về tạo ra các cấu trúc định hướng giữa những gì có thể sẽ vẫn còn bị giữ lại tại trung tâm dữ liệu cục bộ và những gì đi vào đám mây. Có thể sẽ phải tái kiến trúc, tái thiết kế, và tái cài đặt trên đám mây một phần đáng kể của ứng dụng. Do sự chuyển đổi này, một số chức năng có thể bị mất.
* Trong bước tiếp theo của quá trình, tận dụng các tính năng nội tại của dịch vụ điện toán đám mây để tăng cường ứng dụng doanh nghiệp, xác nhận và kiểm tra các hình thức mới của ứng dụng doanh nghiệp với một bộ phần mềm kiểm thử mở rộng bao gồm: kiểm thử các thành phần của các ứng dụng doanh nghiệp trên đám mây. Những kết quả kiểm thử có thể là tích cực hay hỗn hợp.
* Trong trường hợp kết quả kiểm thử hỗn hợp, lặp lại và tối ưu hóa cho phù hợp. Sau một số lần lặp lại tối ưu hóa như vậy, di chuyển được coi là thành công. Việc thực hành chỉ ra rằng tốt nhất là lặp thông qua mô hình bảy bước để tối ưu hóa và đảm bảo rằng việc di chuyển vào đám mây mạnh mẽ và toàn diện.

Rủi ro di chuyển và giảm nhẹ:

Thách thức lớn nhất đối với bất kỳ dự án di chuyển vào đám mây nào là những rủi ro chuyển đổi có được xác định và giảm nhẹ có hiệu quả hay không. Trong mô hình bảy bước di chuyển vào đám mây, bước kiểm thử và xác thực bao gồm nhiều nỗ lực để xác định các rủi ro di chuyển trọng điểm. Trong bước tối ưu hóa, chúng ta giải quyết bằng các cách tiếp cận khác nhau để giảm thiểu những rủi ro đã xác định. Nguy cơ làm cho việc di chuyển thất bại gồm hai loại chính: **rủi ro chuyển đổi tổng quát và những rủi ro liên quan đến bảo mật** [6].

Những **vấn đề chuyển đổi** phải giải quyết bao gồm:

* Giám sát hiệu suất và điều chỉnh ở một mức độ xác định mà từ đó có thể tạo ra tất cả các mức độ lệch lạc.
* Duy trì hoạt động liên tục đồng thời khắc phục sai sót trong môi trường dịch vụ điện toán đám mây.
* Tuân thủ các tiêu chuẩn và các vấn đề quản trị, địa chỉ IP và các vấn đề cấp giấy phép.
* Tuân thủ chất lượng dịch vụ QoS, các thông số của thỏa thuận mức độ dịch vụ SLA tương ứng với cam kết, quyền sở hữu, chuyển nhượng, và lưu trữ dữ liệu trong ứng dụng.
* Tính di động và các vấn đề về khả năng tương tác khác.

Trên **mặt trận bảo mật** cũng có rất nhiều các vấn đề, gồm:

* Bảo mật ở các cấp độ khác nhau của ứng dụng doanh nghiệp áp dụng trên các đám mây.
* Các vấn đề về độ tin cậy và sự riêng tư.
* Tuân thủ một số quy định pháp luật, bao gồm bắt buộc đăng nhập cũng như giữ lại quyền thanh tra tất cả các dấu vết ở cấp chi tiết (nếu có).
* Chưa có giải pháp vững mạnh ngăn chặn các tác động của việc rò rỉ dữ liệu trong các môi trường điện toán đám mây từ các nhà cung cấp.
* Cuối cùng là vấn đề quản lý danh tính/nhận dạng nhất quán.

Chuyển đổi từ đám mây này sang đám mây khác:

Việc chuyển đổi ứng dụng từ đám mây này sang đám mây khác tuy không phải là chưa có ai nghĩ đến nhưng vẫn còn rất mới mẻ và chưa có nhiều thực nghiệm. Tuy rằng trên lý thuyết là có thể thực hiện được nhưng không có nghĩa là sẽ dễ dàng. Để chuyển các ứng dụng từ một nền tảng đám mây này qua nền tảng khác (chẳng hạn từ Amazon sang GoGrid) và từ các trung tâm dữ liệu cục bộ lên đám mây cần phải có các công cụ, hay nhà cung cấp dịch vụ chuyên cho việc này. Hơn nữa, các nhà cung cấp điện toán đám mây cũng cần phải thích ứng với các công nghệ chuẩn căn bản để đảm bảo tính tương thích vận hành, phù hợp với các nhóm ngành khác nhau.

Các vấn đề liên quan đến chuyển đổi ứng dụng từ đám mây này sang đám mây khác có thể được nhận thức rõ trong nội dung của các chương trước, như là:

* Vấn đề về các tiêu chuẩn và nền tảng căn bản: sự không thống nhất về công nghệ (điện toán đám mây hiện chỉ là một khái niệm, không có định nghĩa chính xác về mặt kỹ thuật), sự không tương thích về nền tảng làm mất tính di động của các ứng dụng trên nền tảng đám mây. Để giải quyết vấn đề này cần sử dụng đến các tiêu chuẩn căn bản trong khi các tiêu chuẩn dành cho điện toán đám mây vẫn chưa được phát triển hoàn thiện.
* Vấn đề bản quyền và sự cạnh tranh: những nhà cung cấp dịch vụ điện toán đám mây thành công đều sử dụng những công nghệ, cách thức khác nhau để đem đến cho người sử dụng đám mây trải nghiệm tốt nhất đồng thời tìm cách buộc họ vào nền tảng của mình. Chưa kể đến những phương thức định giá, quản lý người dùng và bảo mật… của mỗi nhà cung cấp dịch vụ lại khác nhau.

Phiên bản hiện tại của Open Cloud Manifesto [7] hỗ trợ sự tương thích vận hành cho dữ liệu và ứng dụng, trong khi Open Cloud Consortium [8] lại khuyến khích các khung vận hành mở dẫn tới việc các đám mây được vận hành cùng nhau bởi các chủ thể khác nhau. Mục tiêu nhắm tới là có thể chuyển các ứng dụng từ đám mây này sang đám mây khác mà không cần phải viết lại chúng.

* + 1. Khuyết điểm của điện toán đám mây:

*Điện toán đám mây vẫn đang trong giai đoạn phát triển, ngoài những lợi ích đã trình bày ở các phần trên, nó cũng có những khuyết điểm mà người tiêu dùng và các nhà phân tích lo ngại. Phần này sẽ trình bày về các khuyến điểm của điện toán đám mây cách khắc phục các khuyết điểm đó ở thời điểm hiện tại.*

Tính bảo mật:

Hầu hết các doanh nghiệp đều lo ngại về vấn đề bảo mật khi sử dụng điện toán đám mây. Họ lo lắng về các lỗ hỏng có thể bị tất công, họ không biết được khi nào các thông tin các tài nguyên công nghệ thông tin của họ nằm ngoài tường lửa của họ. Giải pháp cho vấn đề này nằm ở các nhà cung cấp dịch vụ điện toán đám mây, họ phải có những chuẩn bảo mật rõ ràng.

Đối với các trung tâm dữ liệu truyền thống, cách tiếp về vấn đề bảo mật thông thường là vành đai tường lửa (perimeter firewall), vùng tự do truy cập (demilitarized zones), phân đoạn mạng (network segmentation), hệ thống phát hiện và chống xâm nhập (intrusion detection and prevention systems), và các công cụ quản lý mạng (network monitoring).

Các yêu cầu bảo mật dành cho các nhà cung cấp dịch vụ điện toán đám mây cũng bao gồm những công nghệ và công cụ giống như các trung tâm dữ liệu truyền thống, bao gồm những ứng dụng có vành đai tường lửa cao. Tuy nhiên, phân đoạn vật lý (physical segmentation) và vấn đề bảo mật dựa trên phần cứng không thể áp dụng cho các cuộc tấn công giữa các máy ảo trên cùng một máy chủ. Các máy chủ điện toán đám mây sử dụng cùng hệ điều hành, ứng dụng của doanh nghiệp và các ứng dụng web được đóng gói thành các máy ảo trên các máy chủ vật lý. Vì vậy, một kẻ tấn công có thể khai thác từ xa các điểm yếu của hệ thống hay ứng dụng đó. Ngoài ra, việc nhiều máy áo đựa chứa trên cùng máy chủ vật lý làm tăng khả năng tấn công bề mặt (attack surface) và rủi ro các máy ảo tấn công lẫn nhau. Hệ thống phát hiện và chống xâm nhập cần có thể phát hiện được các hoạt động nguy hại ở mức độ máy ảo bất chấp vị trí của máy ảo trong môi trường đám mây ảo. [9]

Tóm lại, môi trường ảo triển khai các cơ chế bảo mật trên các máy ảo bao gồm: tường lửa, hệ thống phát hiện và chống xâm nhập, hệ thống giám sát (integrity monitoring), kiểm tra đăng nhập (log inspection) sẽ làm cho các máy ảo an toàn và sẵn sàng để triển khai.

Hiệu năng:

Các vấn đề chính về hiệu năng thường gặp phải đối với các ứng dụng với giao tác định hướng (transaction-oriented) hoặc ứng dụng có dữ liệu tập trung (data-intensive). Ngoài ra, người dùng ở càng xa nhà cung cấp dịch vụ đám mây cũng có thể sẽ thấy rõ vấn đề trễ mạng (delay) khi sử dụng dịch vụ. [9]

Khả năng kiểm soát:

Một số bộ phận của công nghệ thông tin sẽ quan tâm tới vấn đề này vì nhà cung cấp dịch vụ điện toán đám mây chỉ có một cách kiểm soát trên toàn bộ các nền tảng mà không thiết kế các nền tảng khác nhau cho các công ty khác nhau với các quy trình nghiệp vụ của riêng họ. [9]

Chi phí băng thông:

Với điện toán đám mây, các công ty có thể tiết kiệm tiền vào phần cứng và phần mềm nhưng chi phí băng thông có thể họ phải trả cao hơn. Chi phí băng thông có thể nhỏ hơn đối với các ứng dụng có dữ liệu không tập trung nhưng sẽ đáng kể đối với các ứng dụng có dữ liệu tập trung. [9]

Độ tin cậy:

Điện toán đám mây vẫn chưa đem lại sự tin cậy 24/24 đối với người sử dụng. Vẫn có trường hợp dịch vụ điện toán đám mây bị ngưng hoạt động trong vài giờ. Trong tương lai, ta có thể mong đợi có nhiều nhà cung cấp dịch vụ điện toán đám mây hơn, các dịch vụ phong phú hơn, các tiêu chuẩn rõ ràng và hoạt động tốt nhất. [9]

Chuyển đổi giữa các nền tảng:

Đối với dịch vụ nền tảng (PaaS), các nhà cung cấp dịch vụ tạo ra các môi trường phát triển trực tuyến giúp các lập trình viên nhanh chóng phát triển và kiểm thử các ứng dụng web cho khách hàng và ứng dụng này chạy trên nền tảng của họ. Mỗi nhà cung cấp dịch vụ PaaS có xu hướng chỉ tập trung xung quanh một trong các công nghệphát triển phổ biến hiện nay dựa trên Java, .Net, hoặc Ruby on Rails.

Khi sử dụng dịch vụ PaaS, các lập trình viên phải sử dụng cùng một tiêu chuẩn và công nghệ phù hợp với nhà cung cấp dịch vụ đó. Nếu đã lựa chọn và phát triển thì việc chuyển đổi ứng dụng từ nền tảng đã lựa chọn sang một nền tảng đám mây khác sẽ gặp nhiều khó khăn. Đây cũng là một trong những khuyết điểm cần phải quan tâm khi xây dựng ứng dụng trên nền tảng đám mây.

Với nền tảng Google App Engine(GAE), GAE hỗ trợ môi trường Java để chạy ứng dụng, lập trình viên sử dụng môi trường phát triển Eclipse có cài plugin Google để phát triển và triển khai ứng dụng (cũng có thể triển khai bằng dòng lệnh). Ngoài ra, lập trình viên sẽ phải sử dụng phương pháp truy xuất dữ liệu mà GAE hỗ trợ (JDO, JPA). Do đó, khi muốn chuyển đổi ứng dụng sang một dịch vụ PaaS có hỗ trợ môi trường Java runtime khác như CloudBee hay VMForce sẽ gặp phải nhiều khó khăn, ít nhất phải phát triển lại các thành phần truy xuất dữ liệu vì mỗi nhà cung cấp chạy trên một hạ tầng của riêng mình.

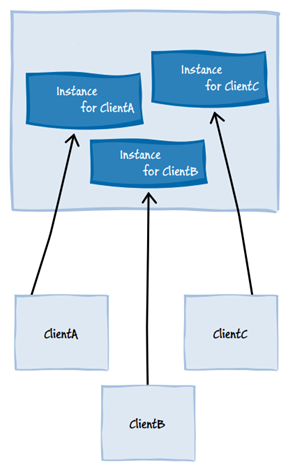
Window Azure là một nền tảng hỗ trợ nền tảng .NET, ta có thể phát triển các ứng dụng web ASP.NET và triển khai trên đó. Môi trường phát triển cũng được hỗ trợ rất tốt cho lập trình viên với bộ công cụ Window Azure Tools dành cho Visual Studio, nhờ đó lập trình viên có thể xây dựng, debug, đưa ứng dụng lên nền tảng Azure ngay trong Visual Studio chỉ với vào thao tác đơn giản. Window Azure cũng cấp cơ sở dữ liệu SQL Azure được xây dựng trên Microsoft SQL Server. Công nghệ này cung cấp môi trường SQL Server trên đám mây, bổ sung index, view, store procedure, trigger, … SQL Azure có thể truy xuất bằng ADO.NET và một số giao thức truy xuất dữ liệu của Window khác. Do đó, khi muốn chuyển đổi một ứng dụng web ASP.NET với cơ sở dữ liệu SQL Server sang ứng dụng chạy trên nền tảng Window Azure, ta phải chuyển đối cơ sở dữ liệu SQL Server sang SQL Azure. Tuy nhiên, nếu ta không đưa ứng dụng này lên nền tảng đám mây Window Azure mà đưa lên một dịch vụ PaaS cũng hỗ trợ nền tảng .NET là AppHarbor, ta có thể sử dụng ngay cơ sở dữ liệu SQL Server mà không cần chuyển đổi vì AppHarbor hỗ trợ cơ sở dữ liệu Microsoft SQL Server.

Đối với một công nghệ phát triển ứng dụng web cũng được ưa thích là Ruby on Rails, cũng có một số nhà cung cấp dịch vụ PaaS hỗ trợ nền tảng này như Heroku và Engine Yard. Khi muốn chuyển đổi nền tảng qua lại giữa hai dịch vụ PaaS trên, việc chuyển đổi sẽ dễ dàng hơn hai ví dụ ở trên vì cả Heroku và Engine Yard đều sử dụng cơ sở dữ liệu MySQL. Ngoài ra ta có thể tìm được công cụ hỗ trợ chuyển đổi giữa hai nền tảng này (EngineYard-Migrate).

Như vậy, khi sử dụng dịch vụ nền tảng đám mây, ta cần phải lựa chọn kỹ công nghệ phát triển ứng dụng và nhà cung cấp dịch vụ được cho là phù hợp về chi phí, môi trường phát triển mà nhà cung cấp đó hỗ trợ. Khi muốn chuyển đổi nền tảng, ta phải lựa chọn dịch nền tảng phù hợp để chuyển đổi, tránh rủi ro phải xây dựng lại ứng dụng từ đầu.

* 1. Ứng dụng nhiều người thuê
     1. Phân biệt giữa ứng dụng một người thuê và nhiều người thuê.

Khi lựa chọn nhà cung cấp máy chủ đám mây dịch vụ phần mềm (SaaS), vấn đề lựa chọn kiến trúc nhiều người thuê hay một người thuê thường được nhắc đến đầu tiên. Trong khi phần lớn các nhà đầu tư cung cấp dịch vụ phần mềm (SaaS) dựa vào kiến trúc nhiều người thuê, đó là lựa chọn tốt nhất để hiểu sự khác nhau giữa hai lựa chọn trên để đảm bảo rằng bạn đang lựa chọn đúng cho việc kinh doanh của bạn. Hiện tại và cả trong tương lai, các nhu cầu chức năng, phân khúc trong công nghiệp, và yêu cầu bảo mật chỉ là một vài yếu tố để quyết định sự lựa chọn đúng dịch vụ. Kiến trúc nền tảng dịch vụ một người thuê.



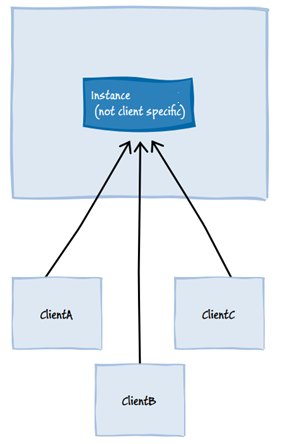
Hình 2-3. Mô tả kiến trúc logic một người thuê, nhiều thể hiện (single tennat, multi-instance)

*(Nguồn: Microsoft)*

Kiền trúc dịch vụ phần mềm (SaaS) một người thuê gồm nhiều thể hiện (multi – instance) là kiến trúc có mỗi thể hiện riêng biệt của ứng dụng phần mềm và cơ sở hạ tầng hỗ trợ được sử dụng bởi mỗi khách hàng, hoặc người thuê. Kiến trúc một người thuê chủ yếu được sử dụng bởi các công ty, doanh nghiệp muốn quản lý tập trung, cơ sở hạ tầng đặt tại nơi làm việc, hoặc nhu cầu của họ về mức độ bảo mật cao hơn. Với một người thuê, mỗi công ty có một cơ sở dữ liệu riêng biệt, một hệ thống riêng biệt hoặc chúng được đặt trên một máy chủ riêng lẻ hoặc tách biệt bằng cách dùng toàn quyền kiểm soát bảo mật để tạo ra một mạng lưới máy chủ ảo.

Trong môi trường dịch vụ phần mềm (SaaS) một người thuê, mỗi người thuê sẽ mua riêng một phần mềm có thể tùy chỉnh để đáp ứng nhu cầu mà họ cần. Các nhà cung cấp máy chủ đám mây sẽ đáp ứng nền tảng cho ứng dụng cuối, người thuê được cung cấp quyền đặc biệt để đảm bảo cấu hình. Ví dụ, người dùng có thể đưa vào thêm một số tính năng như thêm vào các tiểu chuẩn riêng biệt và kết nối các dữ liệu khác nhau vào ứng dụng.

Kiến trúc nền tảng dịch vụ nhiều người thuê



Hình 2-4. Mô tả kiến trúc logic nhiều người thuê, một thể hiện (multi tennat, single-instance)

*(Nguồn: Microsoft)*

Trong kiến trúc nhiều người thuê, còn được gọi là thể hiện duy nhất, dữ liệu từ nhiều công ty sẽ cùng lưu trữ trên một máy chủ, thông thường tách biệt nhau bằng một phân vùng để ngăn chặn dữ liệu từ công ty này sang công ty khác.

Nhiều người thuê là một kiến trúc với chỉ một thể hiện của phần mềm phục vụ cho nhiều khách hàng.

Mỗi khách hàng được gọi là 1 người thuê (tennant). Mỗi người thuê có thể tùy chỉnh một vài phần của ứng dụng, như màu sắc của giao diện hoặc các quy tắc kinh doanh, nhưng họ không thể tùy chỉnh mã nguồn ứng dụng.

Ứng dụng nhiều người thuê có thể giúp tiết kiệm chi phí, bởi vì chi phí phát triển và duy trì phần mềm dựa trên sự chia sẽ. Điều này là trái ngược với ứng dụng một người thuê, kiến trúc này cho mỗi khách hàng họ muốn sở hữu riêng thể hiện phần mềm của chính họ và có thể truy cập vào mã nguồn.

Với kiến trúc nhiều người thuê, nhà cung cấp chỉ cần cập nhật một lần, và tất các người thuê sẽ được cập nhật đồng thời. Với kiến trúc một người thuê, nhà cung cấp phải trực tiếp thay đổi từng thể hiện của phần mềm để cập nhật ứng dụng.

Trong điện toán đám mây, ý nghĩa của kiến trúc nhiều người thuê có thể mở rộng bởi mô hình dịch vụ hưởng lợi ích từ ảo hóa và điều khiển truy cập.

Nhà cung cấp một phần mềm như một dịch vụ (SaaS) có thể chạy một thể hiện của ứng dụng đó với cùng một thể hiện dữ liệu và cung cấp quyền truy cập web đến nhiều khách hàng. Như một kịch bản, mỗi dữ liệu người thuê là cô lập và duy trì vô hình với các người thuê khác.

* + 1. Phát triển ứng dụng nhiều người thuê trong môi trường đám mây.

Triển khai một ứng dụng nhiều nhiều người thuê trên điện toán đám mây.

*Phần này trình bày một số yêu cầu và mục đích chung của ứng dụng nhiều người thuê. Một số ứng dụng có thể không phù hợp trong một số tình huống cụ thể và tầm quan trọng của yêu cầu và mục tiêu khác nhau trong mỗi tình huống. Ví dụ không phải tất cả ứng dụng nhiều người thuê yêu cầu cùng một mức độ tùy biến của người thuê hoặc đối mặt với những hạn chế pháp lý như nhau.*

Xem xét các yêu cầu và múc đích đới với một ứng dụng nhiều người thuê từ quan điểm của người thuê và của nhà cung cấp.

*Khía cạnh người thuê:* nhiều người thuê chia sẻ việc sử dụng ứng dụng nhiều người thuê. Nhiều người thuê khác nhau sẽ có mục đích và yêu cầu khác nhau. Người thuê không muốn biết nhà cung cấp cài đặt như thế nào nhưng mong đợi ứng dụng thực thi như thể người thuê sử dụng duy nhất nó. Dưới dây là danh sách mục tiêu và yêu cầu quan trọng từ một khía cạnh người thuê:

* + - * Sự cô lập, tách ly (Isolation): cá nhân người thuê không muốn những hoạt động của người thuê khác ảnh hưởng đến mình, không thê truy xuất đến dữ liệu của mình.
      * Tính sẵn sàng cao (Availability): ứng dụng có khả năng thực thi liên tục khi gặp sự cố về lưu trữ dữ liệu, và khả năng chịu tải liên tục khi số lượng người truy cập và sử dụng lớn.
* Tính mở rộng (Scalability): người thuê dùng chung 1 ứng dụng, mỗi người thuê mong muốn ứng dụng có khả năng mở rộng theo yêu cầu của người thuê
  + - * Chi phí (Cost): Chi phi thấp hơn việc sử dụng ứng dụng một người thuê, bởi vì nhiều người thuê có khả năng chia sẽ tài nguyên. Người thuê cần nghiên cứu mô hình tính phí để có thể uớc tính chi phí mà mình có thể sử dụng.
      * Tùy chỉnh (Customizability): có thể tùy chỉnh thêm hoặc bỏ các đặc tính, thay đổi màu sắc, logo, hoặc thêm các đoạn mã script.

*Khía cạnh nhà cung cấp:* dưới dây là danh sách các mục đích và yêu cầu quan trọng:

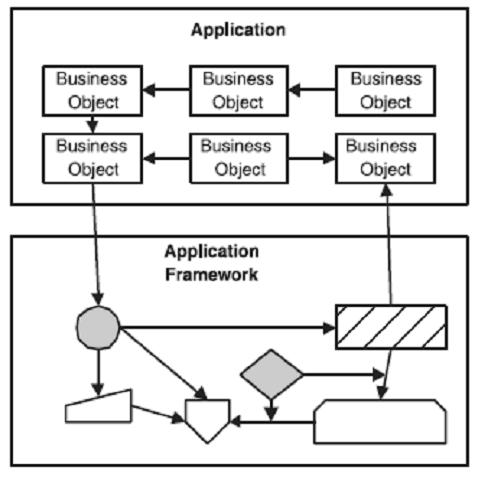
* + - * Đáp ứng những mong đợi của người thuê: Một nhà cung cấp có thể đề nghị một tập quy tắc chung để xác định làm thế nào ứng dụng đáp ứng mọng đợi người thuê.
      * Lợi nhuận (Profitability): nhà cung cấp muốn có một tỉ lệ thích hợp về lợi nhuận dựa trên vốn đầu tư đã thực hiện trong việc phát triển và cung cấp dịch vụ. Doanh thu từ các ứng dụng phải đủ để bao gồm cả vốn và chi phí hoạt động của ứng dụng.
      * Thanh toán (Billing): cách để tính tiền, yêu cầu ứng dụng giám sát việc sử dụng tài nguyên, nếu nhà cung cấp không muốn sử dụng một số tiền cố định mà chi một tỷ lệ trả tiền nào đó (như trả tiền hàng tháng).
      * Nhiều cấp độ dịch vụ: mỗi dịch vụ sẽ có chức năng, giới hạn sử dụng khác nhau. Ví dụ như cấp độ cơ bản, cấp độ doanh nghiệp.
      * Cung cấp (Provisioning): có khả năng cung cấp cho một người thuê mới, khi sử dụng ứng dụng nhiều người thuê với số lượng người thuê mới cần tự động cung cấp các dịch vụ cho người thuê mới.
      * Khả năng duy trì (Maintainability): khả năng duy trì, có khả năng cập nhật và thực hiện duy trì khi người thuê sử dụng nó.
      * Quản lý (Monitoring): giám sát ứng dụng mọi lúc để xác định vấn đề và giải quyết vấn đề.
      * Tự động (Automation): tự động thêm người thuê, tự động mở rộng ứng dụng tự động bằng cách thêm hoặc bỏ các tài nguyên khi người thuê yêu cầu.

Chọn lựa kiến trúc dữ liệu nhiều người thuê.

*Trình bày về những nhân tố quan trọng để xem xét khi thiết kế dữ liệu cho ứng dụng nhiều người dùng*

* + - * **Lưu trữ dữ liệu trong ứng dụng**
* Ứng dụng nhiều người dùng là cần sự phân chia dữ liệu cho từng người và thêm vào đó là bảo đảm được sự mở rộng để hỗ trợ mỗi người.
  + - * + Nhiều nhân tố khi lựa chọn kiểu của bộ nhớ: giá tiền, tính mở rộng…
      * **Tính sẵn sang của bộ nhớ**
        + Phụ thuộc 2 nhân tố: kĩ thuật lưu trữ không bị lỗi với bất kì yêu cầu nào và hành vi của việc kết nối giữa ứng dụng và các ràng buộc thậm chí là ép buộc truy cập.
        + Kiểu của dữ liệu phải có sẵn, và số lượng tối đa nhất định. Cẩn thận thiết kế cách lưu trữ để sử dụng chiến thuật phân chia tránh đụng chạm vào các giới hạn này.
        + SLA (Service Level Agreement) bảo đảm tính sẵn sàn là 99.9% nhưng thực tế sẽ bị nghẽn lại khi ứng dụng tải cơ sở dữ liệu trên máy chủ hoặc trên các cơ sở dữ liệu riêng.
        + Nhân tố thứ hai là hiệu suất giữa ứng dụng và dữ liệu, có thể sử dụng kĩ thuật nhóm những dữ liệu giống nhau để đặt những tài nguyên cùng nằm trên một dữ liệu trung tâm.
        + Trong những ứng dụng và dữ liệu ở nhiều địa lý khác nhau thì người ta tạo mộ bộ dữ liệu sau đó sao chép ra và lưu vào bộ nhớ đệm và chuyển đến vùng gần với ứng dụng.
      * **Kiến trúc dữ liệu nhiều người dùng:**
        + Phải có chế độ riêng biệt với những người dùng khác nhau và phải hỗ trợ tính mở rộng và tùy chỉnh nơi chứa dữ liệu.
      * **Phân chia dữ liệu**
        + Nên nhận biết các lỗi tiết lộ thông tin trong ứng dụng nhiều người dùng.Tuy nhiên việc thiết kế theo lý thuyết để tách riêng các cơ sở dữ liệu của những người thuê chỉ cung cấp một mức độ nhất định.
        + Kiểu thiết kế lúc này sẽ cho phép sử dụng các kĩ thuật kiểm soát cơ sở dữ liệu, một kế hoạch phân chia dữ liệu hoặc là kết hợp với những cái khác.Trong ứng dụng nhiều người dùng, thì một người chỉ được truy cập vào cơ sở dữ liệu của họ.
  1. Khung ứng dụng.
     1. Khung ứng dụng là gì.

Hãy bắt đầu bằng định nghĩa một khung là gì? Theo từ điển văn hóa của Mỹ: “Khung là một cấu trúc để hổ trợ hoặc đóng gói một thứ gì đó khác, đặc biệt cung cấp khung sườn làm cơ sở cho một thứ gì đó có cấu trúc”. Cấu trúc đóng vai trò quan trọng trong việc phát triển ứng dụng. Một ứng dụng khá phức tạp có thể phải chuyển đổi và thay đổi các thành phần bên trong nó mà không ai có thể theo dõi các mối quan hệ giữa chúng. Cấu trúc giúp chúng ta tổ chức những thành phần thay đổi này vào một vài thành phần giúp chúng ta có thể dễ dàng quản lý và chuyển đổi. Một khung ứng dụng cung cấp cho lập trình viên một cấu trúc và một khuôn mẫu làm cơ sở để xây dựng ứng dụng của họ. Một khung thường bao gồm các lớp trừu tượng, các lớp cụ thể và định nghĩa trước các giao tác giữa các lớp bên trong khung. Lập trình viên có thể xây dựng ứng dụng trên khung này để tái sử dụng mã nguồn và những thiết kế được cung cấp bởi bộ khung.



Hình 2-5. Tổng quan về mối quan hệ giữa ứng dụng và một khung ứng dụng

*(Nguồn: [10])*

Khái niệm về khung ứng dụng không phải là mới, các loại khung đã xuất hiện cách đây một vài thập kỷ. MVC là khung cho việc lập trình giao diện người dùng được thiết kế theo mẫu “Observer Design Pattern”. Ngoài ra còn có một số khung lập trình giao diện khác như MacApp và MFC để hổ trợ phát triển ứng dụng trên Macintosh và Windowns… [10].

* + 1. Tại sao phải sử dụng khung ứng dụng.

Ở đây có năm lợi ích chính của việc sử dụng khung ứng dụng [11].

* *Tính mô đun*

Việc phần chia ứng dụng thành những thành phần có cấu trúc (mô đun) cho phép nhà phát triển có thể chia khung ứng dụng theo từng phần. Những nhà phát triển muốn sử dụng một thành phần trong khung ứng dụng thì những thành phần này được bảo vệ để tránh những thay đổi làm ảnh hưởng đến những thành phần khác trong khung ứng dụng. Điều này làm tăng hiệu suất và giảm thời gian chỉnh sủa mã nguồn mà bị ảnh hưởng bởi những lập trình viên khác. Bằng việc chia nhỏ thành các thành phần chúng ta có thể tối đa hóa hiệu suất bằng cách chỉ định một lập trình viên vào một mô đun cụ thể. Ví dụ như trong lập trình ứng dụng web, những lập trình viên thành thạo lập trình ở tầng giao diện có thể chỉ định vào phần giao diện của ứng dụng. Tương tự với những lập trình viên xử lý nghiệp vụ có thể chỉ định vào phần lập trình xử lý nghiệp vụ của ứng dụng. Hơn nữa các lập trình viên có thể sử dụng qua lại các mô đun có liên quan với nhau khi phát triển.

* *Tính tái sử dụng*

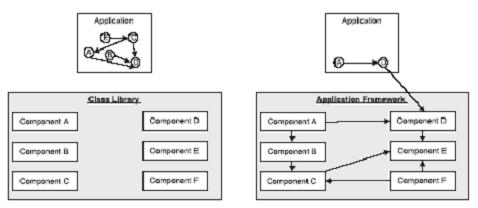
Tính tái sử dụng mã nguồn là một trong những tính năng quan trọng và là mục tiêu chính của ứng dụng. Một khung ứng dụng cung cấp tính tái sử dụng cho ứng dụng bằng cách chia sẽ những lớp và mã nguồn. Ứng dụng thông thường bao gồm nhiều tác vụ giống như ngoài đời sống thực tế. Tuy nhiên những lập trình khác trong nhóm lại có những cài đặt riêng của chính họ điều này làm trùng lắp mẵ nguồn, mà còn khó khăn trong việc bảo trì khi có sự cố xảy ra, một khi có sự thay đổi của những tác vụ chúng ta buộc phải thay đổi lại nhiều nơi trên ứng dụng để đảm bảo tính toàn vẹn của nó. Hơn nữa mỗi lập trình viên lại có cách tiếp cận vấn đề khác nhau trong khi cài đặt điều này dẫn đến chúng ta sẽ có một thiết kế phần mềm kém và không thể lường trước được điều gì sẽ xảy ra. Với một khung ứng dụng chúng ta có thể quản lý những sự thay đổi và có thể sử dụng các thành phần của khung một cách dễ dàng từ tầng ứng dụng. Hơn nữa chúng ta có thể kiểm tra thiết kế của phần mềm thông qua những thành phần bên trong một cách dễ dàng.

* *Tính mở rộng*

Tính mở rộng là khả năng mở rộng thêm những tính năng cho khung ứng dụng hiện tại và thay đổi những thành phần cho phù hợp với những yêu cầu nghiệp vụ. Tính mở rộng là một trong những tính năng quan trọng khi xây dựng khung ứng dụng, mỗi ứng dụng doanh nghiệp thường có những yêu cầu nghiệp vụ riêng, kiến trúc và cách cài đặt khác nhau, và chúng ta không thể đưa vào cùng trong một khung chung được. Nhưng cách thiết kế một khung ứng dụng thì để chúng ta có khả năng tùy chỉnh lại, và như thế các ứng dụng doanh nghiệp khác nhau có thể sử dụng tính năng chung của khung ứng dụng, các nhà phát triển có thể tự do điều chỉnh lại ứng dụng của họ cho phù hợp với những yêu cầu nghiệp vụ riêng. Tuy nhiên khi tạo một khung ứng dụng tính mở rộng cần phải xác định được ngữ cảnh và các giả định của ứng dụng mà bạn đang phát triển

* *Tính đơn giản*

Tính đơn giản liên quan đến cách mà một khung ứng dụng đóng gói nhiều sự kiểm soát các luồng xử lý và giấu chúng khi lập trình viên sử dụng. Việc đóng gói của khung ứng dụng khác với lớp thư viện (class library) rằng một lớp thư việc chỉ bao gồm một tập các thành phần mà lập trình viên có thể sử dụng, tuy nhiên họ phải hiểu mối quan hệ giữa các thành phần và viết luồng xử lý để các thành phần đó làm việc với nhau. Còn với khung ứng dụng việc đóng gói điều khiển các luồng xử lý được thực hiện sẵn rồi, các lập trình viên không cần phải viết mã nguồn để điểu khiển các thành phần giao tác với nhau. Khi một nhà phát triển bắt đầu sử dụng bộ khung này họ có thể đạt hiệu quả cao khi sử dụng bởi vì họ không cần phải biết quá nhiều về các thành phần trong khung ứng dụng làm việc với nhau như thế nào.



Hình 2-6. So sánh giữa khung ứng dụng và lớp thư viện

*(Nguồn: [11])*

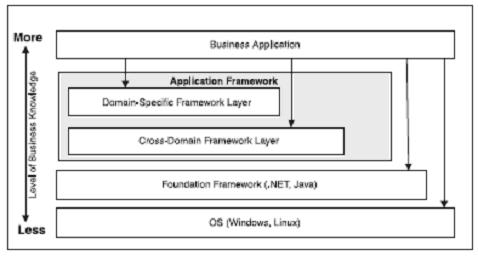
* *Tính bảo trì*

Tính bảo trì là cung cấp khả năng thay đổi một cách hiệu quả khi thay đổi những yêu cầu nghiệp vụ. Các thành phần của khung ứng dụng thường được chia sẻ qua nhiều ứng dụng hoặc nhiều vùng trong một ứng dụng. Việc chỉ có một bản duy nhất của bộ khung giúp ứng dụng trở nên dễ bảo trì bởi vì bạn chỉ cần thay đổi một lần khi mà yêu cầu thay đổi. Một khung ứng dụng cũng có nhiều tầng, mỗi tầng được thiết kế để phục một nghiệp vụ nào đó, những tầng này dễ thay đổi khi yêu cầu nghiệp vụ thay đổi, khi những thay đổi này xảy ra chỉ thành phần tại tầng nơi mà giả định nghiệp vụ bị sai thì cần được sửa đổi và kiểm tra lại. Ở tẩng dưới cùng của những thành phần khung ứng dụng là tầng mà không giả định nghiệp vụ. Chúng cũng là thành phần chung nhất trong bộ khung. Vì vậy việc tiêm các mức độ hiểu biết về nghiệp vụ vào những tầng của khung ứng dụng giúp chúng ta có thể giảm những ảnh hưởng từ việc thay đổi những quy luật và những yêu cầu của ứng dụng. Điều này cũng giúp chúng ta giảm chi phí bảo trì, chúng ta chỉ chỉnh sửa những đoạn mã nguồn mà bị ảnh hưởng khi nghiệp vụ thay đổi.

* + 1. Các khái niệm, kỹ thuật khi xây dựng một khung ứng dụng.

Một số khái niệm cơ bản:

Một ứng dụng xây dựng trên một bộ khung gồm có hai tầng: tầng ứng dụng (the application layer), tầng khung làm việc (the framework layer). Tầng khung làm việc có thể bao gồm nhiều thành phần và có thể được nhóm lại vào “khung chỉ rõ lĩnh vực” (domain-specific framework) và “khung đa lĩnh vực” (cross-domain framework) [11].



Hình 2-7. Mô tả nhiều tầng trong một ứng dụng

*(Nguồn: [11])*

*Ứng dụng nghiệp vụ (The Business Application)*

Ứng dụng nghiệp vụ đại diện cho các tùy chỉnh ứng dụng của nhà phát triển. Nó cài đặt chi tiết những kiến thức nghiệp vụ cho các ứng dụng cụ thể theo sự phân tích nghiệp vụ sẵn có. Nếu quy trình nghiệp vụ và quy luật thay đổi những thay đổi hầu như chỉ xảy ra ở đây và nó sẽ bị cô lập lại.

*Khung ứng dụng (The Application Framework)*

Khung ứng dụng được xem như một nửa của ứng dụng, nơi mà kiến trúc được phát triển như một nền tảng cho các nhà phát triển sử dụng để xây dựng những ứng dụng nghiệp vụ của họ. Khung ứng dụng có thể được chia thành hai phần “khung chỉ rõ lĩnh vực” và “khung đa lĩnh vực”.

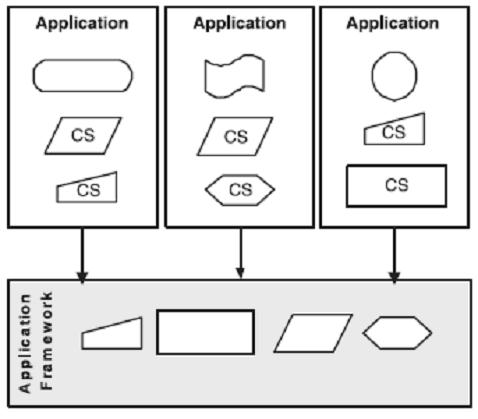
* **Khung chỉ rõ lĩnh vực** bao gồm những thành phần chuyên dụng mà nhắm đến mục tiêu là một lĩnh vực nghiệp vụ cụ thể. Nếu so sánh với tầng ứng dụng nghiệp vụ thì tầng khung chỉ rõ lĩnh vực chỉ cần cài đặt những kiến thức cơ bản cho tất cả các ứng dụng của một nghiệp vụ. Trong khi đó ở tầng ứng dụng nghiệp vụ bạn phải cung cấp những kiến thức về nghiệp vụ và xử lý logic đối với những nghiệp vụ cụ thể. Ở tầng ứng dụng nghiệp vụ bạn có thể thiết kế và thực hiện các ứng dụng thực tế thì ở khung chỉ rõ lĩnh vực được thiết kế bởi những chuyên gia có hiểu biết sâu về những nghiệp vụ cụ thể và biết cách đóng gói và trừu tượng hóa những lĩnh vực nghiệp vụ để thích hợp với những nhà phát triển xây dựng những ứng dụng nghiệp vụ cụ thể.
* **Khung đa lĩnh vực** bao gồm những chủ đề phổ biến trong số lĩnh vực nghiệp vụ khác nhau của những ứng dụng, mà chúng ta có thể đóng gói thành một khung đa lĩnh vực. Ví dụ chúng ta có thể sử dụng vài cách để quản lý thông tin cấu hình được sử dụng bởi những phần khác nhau của ứng dụng. Trong môi trường ứng dụng java phân tán một ứng dụng thường trao đổi với ứng dụng khác thông qua những dịch vụ thông báo, dịch vụ này là có sẵn và được sử dụng để chuyển đổi thông tin giữa các ứng dụng khác nhau. Như bạn vừa thấy chúng ta có thể xác định những chủ đề phổ biến trong số các loại khác nhau của ứng dụng và phát triển dịch vụ và các thành phần để phục vụ những yêu cầu phổ biến này. Những người phát triển khung đa lĩnh vực là những người mà đã phát triển có kinh nghiệm phát triển nhiều ứng dụng và có một sự hiểu biết tốt về thiết kế phần mềm cũng như việc hiểu biết về các tính năng mà phổ biến cho nhiều ứng dụng. Họ không cần phải biết nghiệp vụ cụ thể, họ chỉ cần có kiến thức tốt về kĩ năng lập trình hướng đối tượng để có thể xây dựng một khung để cho các lập trình viên các có thể dễ dàng cắm thêm tùy chỉnh về xử lý nghiệp vụ để giải quyết những vấn đề đặc biệt. Khung đa lĩnh vực không chứa những kiến thức về nghiệp vụ cụ thể, nó có thể được coi là chung cho hầu hết các ứng dụng. Như vậy nó không bị ảnh hưởng khi những yêu cầu hay luật lệ nghiệp vụ thay đổi. Tuy nhiên tầng này lại ảnh hưởng khi nhìn nhận một chủ đề thông dụng mới mà phát sinh trong khi phát triển ứng dụng, các chủ đề như thế sẽ được cài đặt trong tầng này.

Kỹ thuật để phát triển một khung ứng dụng

Dưới đây là một số các kĩ thuật thông dụng và những cách tiếp cận giúp chúng ta có thể phát triển một khung ứng dụng.

* *Những điểm phổ biến (Common Spots):*

Những điểm phổ biến đại diện cho những nơi trong ứng dụng nghiệp vụ mà các chủ đề phổ biến được lặp đi lặp lại. Nếu một phần nào đó của ứng dụng lặp đi lặp lại mà không có sự thay đổi, chúng ta có thể tách những điểm phổ biến này ra và đóng gói lại vào khung, điều này giúp chúng ta tránh được việc trùng lắp mã nguồn và tái sử dụng mã nguồn.



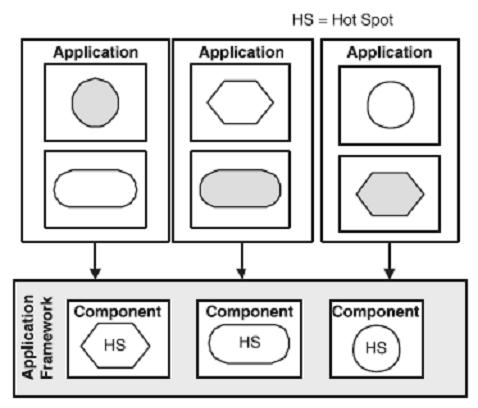
Hình 2-8. Minh họa điểm phổ biến

*(Nguồn: [11])*

Trong hình trên khung ứng dụng chứa những thành phần mà cung cấp những cài đặt cho những điểm phổ biến mà được tìm thấy ở ứng dụng. Khi nhà phát triển sử dụng chúng, chúng ta có thể tham chiếu đến những cài đặt cho những điểm phổ biến này trong thành phần của khung thay vì tự phát triển chúng. Mỗi ứng dụng nghiệp vụ là duy nhất và ở đây có nhiều điểm mà mỗi ứng dụng có những thay đổi đáng kể nhưng vẫn được xem là những điểm phổ biến thì điều này nhà phát triển cần phải tính đến những thay đổi khi phát triển khung và đảm bảo cho các nhà phát triển khác có thể tận dụng lợi thế của khung ứng dụng.

* *Những điểm nóng (Hot Spots):*

Những điểm nóng là một phần của tính linh động trong khung ứng dụng. Là nơi mà chúng ta giữ chổ trong khung ứng dụng nơi mà những tùy chỉnh xảy ra. Ở đây chúng ta không phải cài đặt gì trong khung cả, mà sau đó nó sẽ được cài đặt khi xây dựng một ứng dụng nghiệp vụ trên khung ứng dụng này.



Hình 2-9. Những điểm nóng

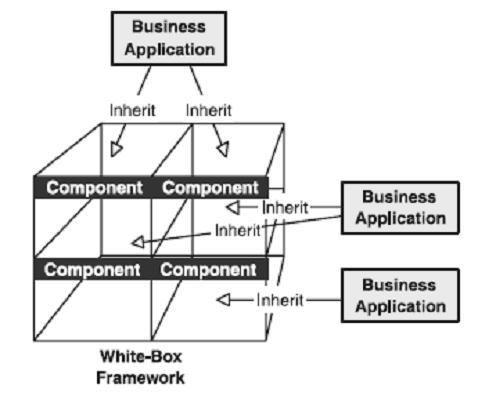
*(Nguồn: [11])*

Những thành phần trong tầng khung ứng dụng bao gồm những điểm nóng. Mỗi ứng dụng có lẽ sử dụng một số thành phần của khung thông qua ứng dụng. Khi đó nếu ứng dụng sử dụng những thành phần mà chứa những điểm nóng thì cần phải cài đặt lại những điểm nóng này để sử dụng khung ứng dụng. Như hình ở trên bạn có thể thấy những cách cài đặt khác nhau (hình màu sắc khác nhau) giữa những ứng dụng. Việc xác định một điểm nóng là một việc không hề dễ dàng, để làm được điều này bạn phải hiểu các lĩnh vực nghiệp vụ và hiểu vị trí có khả năng tùy chỉnh lại trong ứng dụng. Việc có qua nhiều các điểm nóng trong khung dẫn đến việc viết lại mã nguồn quá nhiều, ngược lại việc có quá ít các điểm nóng làm cho các lập trình viên khó thích nghi với khung khi mà những tùy chỉnh mà họ muốn khó thực hiện.

Dưới đây là hai cách tiếp cận mà cho phép những điểm nóng xuất hiện trong khung ứng dụng.

* Cách tiếp cận kế thừa: được hiện thực bởi hai khái niệm hướng đối tượng là “hook method” và “template method”.
* Cách tiếp cận thành phần: Cách tiếp cận kế thừa là một cách đơn giản để cho phép những điểm nóng xuất hiện trong khung ứng dụng, nhưng bởi vì lập trình viên phải biết những dữ liệu và phương thức có nào có sẵn trong lớp cha, cũng như là quan hệ qua lại giữa chúng để cài đặt những lớp trừu tượng. Họ thường cần phải hiểu biết chi tiết về khung ứng dụng để sử dụng chúng. Điều này có thể dẫn đến việc không thể điều khiển truy cập và việc hiệu chỉnh những trạng thái của lớp con mà vượt ra ngoài khuôn khổi thiết kế của lớp cha. Cách mà để nhà phát triển giảm sự hiểu biết chi tiết về khung ứng dụng là định nghĩa các điểm nóng như là một tập các giao diện (interface) là cách để kết nối đến khung ứng dụng. Nhà phát triển có thể tạo các thành phần bằng cách cài đặt những giao diện sau đó thêm những tùy chỉnh vào khung ứng dụng.
* *Khung hộp trắng (White-Box frameworks)*

Khung hộp trắng là một khung mà bao gồm các lớp trừu tượng, để thích ứng với bộ khung này các nhà phát triển phải tạo những lớp cụ thể (concrete classes) để kế thừa những lớp trừu tượng này. Khung hộp trắng dựa trên cách tiếp cận kế thừa để cho phép những điểm nóng xuất hiện trên khung ứng dụng.



Hình 2-10. Khung hộp trắng

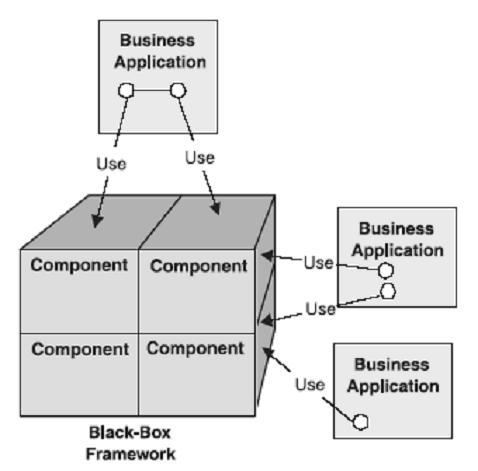
*(Nguồn: [11])*

Khi phát triển loại khung này bạn cần phải giả định dạng luồng xử lý nghiệp vụ liên quan đến mỗi thành phần trong khung thông qua phương thức mẫu (template method). Khi một nhà phát triển bắt đầu sử dụng có chỉ viết lại (override) những phương thức trong ở lớp kế thừa và họ không cần phải biết gì về toàn bộ quá trình xử lý và cách những phương thức trừu tượng này hoạt động bên trong khung.

Điểm hạn chế của loại khung này là tính linh hoạt. Khi xác định luồng xử lý xảy ra bên trong một thành phần thông qua phương thức mẫu điều này làm mã nguồn cứng nhắc (hard-coded), điều này làm đối lập với tính chất linh hoạt khi một yêu cầu nghiệp vụ thay đổi đòi hỏi phải thay đổi những luồng xử lý này. Hơn nữa loại khung này thường yêu cầu nhà phát triển phải biết nhiều chi tiết về các thành phần và họ cần phải tham chiều tới những phương thức trừu tượng và các biến trong khi cài đặt mã nguồn. Điều này buộc họ phải hiểu các thành phần chi tiết trong khung như là một điều tiên quyết.

* *Khung hộp đen (Black-box frameworks)*

Khung hộp đen bao gồm những lớp cụ thể và sẵn sàng để sử dụng và các nhà phát triển có thể mở rộng những thành phần của khung này để đạt được sự tùy chỉnh trong khung hộp đen này. Khi sử dụng loại khung này họ phải kết hợp những thành phần lại để đạt được kết quả họ mong muốn. Khung hộp đen có thể chứa nhiều điểm phổ biến và nó cũng sử dụng cách tiếp cận thành phần (the composition approach) để cho phép tính năng điểm nóng.



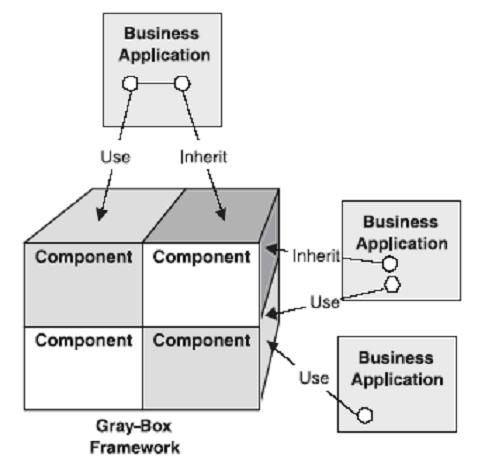
Hình 2-11. Khung hộp đen

*(Nguồn: [11])*

Khi sử dụng loại khung này chúng ta đạt được tính linh hoạt hơn là khung loại trắng. Nhà phát triển có thể chọn những thành phần khác nhau để đạt được những yêu cầu cụ thể. Việc thích nghi với những thành phần là thông qua những định nghĩa về giao diện như những phương thức và thuộc tính. Nhưng khi cài đặt nhà thì phát triển phải cài đặt những luồng xử lý và logic để liên kết những thành phần lại với nhau và phải biết các thành phần liên kết với nhau như thế nào.

* *Khung hộp xám (Gray-Box Frameworks)*

Khung hộp xám dùng thường kết hợp cả hai cách tiếp cận kế thừa và tiếp cận thành phần. Các ứng dụng thực tế bên ngoài thường sử dụng khung hộp xám. Việc lựa chọn các sử dụng các thành phần theo cách tiếp cận kế thừa hay tiếp cận thành phần được quyết định dựa vào các thành phần được hiện thực như thế nào và cách mà lập trình viên sử dụng nó như thế nào trong các ứng dụng nghiệp vụ của họ.



Hình 2-12. Khung hộp xám

*(Nguồn: [11])*

Khi bạn thu thập nhiều tính năng từ nhiều thành phần thì chúng ta phải biết kết hợp các thành phần lại. Nhưng trong cách tiếp cận kế thừa, lớp được kế thừa thường chứa phần lớn các tính năng, yêu cầu của nó rồi, điều này làm giảm số lượng những đối tượng cần tạo và truy cập để có được cùng kết quả như việc sử dụng cách tiếp cận thành phần.

Theo cách tiếp cận thành phần, nhà phát triển làm việc với các thành phần của khung một cách tách rời, điều này làm cho ứng dụng có thể thích ứng với những thay đổi linh hoạt và dễ mở rộng hơn. Tuy nhiên khi ứng đụng đã được triển khai họ sẽ đối mặt với việc di chuyển các phần dẫn đến việc khó bảo trì, một yêu cầu nào đó thay đổi có lẽ sẽ ảnh hưởng đến một loại các thành phần của khung. Còn cách tiếp cách kế thừa có ít linh hoạt hơn nhưng bù lại có ít phải di chuyển các thành phần hơn, khi yêu cầu nghiệp vụ thay đổi thì chúng ta chỉ thay đổi ở một vài lớp trong cây kế thừa. Vì vậy việc kết hợp tính linh hoạt và tạo một vài thành phần thô để các nhà phát triển làm việc với một vài thành phần để đạt được kết quả họ muốn, sau đó khung ứng dụng sẽ trở nên dễ sử dụng và việc phát triển trở nên hiệu quả hơn.

* *Mẫu thiết kế (Design Patterm)*

Việc sử dụng mẫu thiết kế giúp nhà lập trình khung ứng dụng giải quyết các vấn đề xảy ra một cách phổ biến khi xây dựng phần mềm như làm thế nào để cải thiện việc xử lý khi thay đổi xảy ra và việc phải tùy chỉnh lại ứng dụng. Một số mẫu thiết kế được giới thiệu trong quyển sách [12].

* 1. Giải pháp
     1. Tổng quan về Amazon Web Service

Giới thiệu:

Amazon Web Service (AWS) là một bộ các dịch vụ tính toán từ xa (remote computing services) được xây dựng trên nền tảng điện toán đám mây thông qua môi trường mạng (internet) bởi Amazon.com [13]. Amazon Web Service (AWS) là tên của một tập hợp các dịch vụ mạng (web service) được xây dựng bởi Amazon cho phép nhà phát triển thứ ba có thể truy cập và xây dựng ứng dụng trên nền tảng công nghệ của công ty. Tập hợp các dịch vụ này bao gồm một số dịch vụ cơ sở hạ tầng có thể làm tăng hoặc thay thế các dịch vụ cơ sở hạ tầng truyền thống được yêu cầu thông qua ứng dụng mạng (web). Những dịch vụ cơ sở hạ tầng này cung cấp lưu trữ, tính toán, hệ thống thông điệp, hệ thống thanh toán, và cơ sở dữ liệu từ đó bạn có thể truy cập với một tài khoản amazon.com và thẻ credit card.

Sử dụng AWS, bạn có thể trưng dụng sức mạnh tính toán, lưu trữ, và các dịch vụ khác trong vài phút và có sự linh hoạt để lựa chọn nền tảng phát triển hoặc mô hình lập trình có ý nghĩa nhất cho những vấn đề đang cố gắng để giải quyết. Bạn chỉ phải trả cho những gì bạn sử dụng, không có chi phí trả trước hoặc cam kết dài hạn, điều này làm cho AWS trở thành một cách hiệu quả để cung cấp các ứng dụng.

Hiện tại AWS cung cấp các thành phần chính từ đó bạn có thể xây dụng ứng dụng một cách linh hoạt như là: Amazon Simple Storage Service (S3), Amazon Elastic Compute Cloud (EC2), Amazon Simple Queue Service (SQS), Flexible Payments Service (FPS), DynamoDB… Thông tin các thành phần chính này chúng em sẽ trình bày ở phần sau [14].

AWS dễ dàng phân biệt với các nhà cung cấp khác với các đặc điểm nổi bật [20]:

* **Linh hoạt:** AWS cho phép tổ chức sử dụng các mô hình lập trình, hệ điều hành, cơ sở dữ liệu, và kiến trúc mà họ đã quen thuộc. Ngoài ra, sự linh hoạt này giúp các tổ chức và kết hợp các kiến trúc để phục vụ cho nhu cầu kinh doanh đa dạng của họ.
* **Chi phí hiệu quả**: Với AWS, tổ chức chỉ trả tiền cho những gì họ sử dụng, mà không cần phí trả trước hoặc cam kết dài hạn.
* **Khả năng mở rộng và mềm dẻo**: Các tổ chức có thể nhanh chóng thêm và trừ tài nguyên AWS cho các ứng dụng của họ để đáp ứng nhu cầu của khách hàng và quản lý chi phí.
* **An toàn**: AWS xây dựng các dịch vụ phù hợp với thực tiễn an ninh tốt nhất, cung cấp các tính năng bảo mật thích hợp trong các dịch vụ và các văn hướng dẫn sử dụng các tính năng.
* **Kinh nghiệm**: Khi sử dụng AWS, tổ chức có thể tận dụng hơn mười lăm năm kinh nghiệm của Amazon trong việc cung cấp cơ sở hạ tầng toàn cầu với quy mô lớn đáng tin cậy và an toàn.

Tương tác với Amazon Web Service:

Để tương tác với AWS, thư viện giao tiếp phải tạo thông điệp yêu cầu HTTP để mô tả hành động thực thi, và nó phải cung cấp dữ liệu sẽ thực thi. Thư viện sẽ gửi thông điệp yêu cầu để chỉ định dịch vụ, chờ trả lời, xác định yêu cầu có thực hiện thành công hay không, và gửi trả lời ngược về khách (client) cho những xử lý tiếp theo [15].

* + - * 1. REST-Based APIs:

Các dịch vụ của amazon S3, EC2, SQS, FPS, và SimpleDB được xây dựng thông qua hai giao diện lập trình ứng dựng (APIs) dựa trên các chức năng chuẩn của giao thức HTTP là: REST và giao diện truy vấn (query interfaces). Mỗi API dịch vụ định nghĩa cấu trúc và nội dung của thông điệp yêu cầu dạng HTTP.

* Lời yêu cầu và trả lời HTTP (HTTP requests and responses): Lời yêu cầu HTTP (HTTP request) bao gồm 4 thành phần chính là URI, phương thức HTTP (HTTP method), mở đầu lời yêu cầu (request headers) và phần thân của lời yêu cầu (request body). Lời phản hồi HTTP (HTTP responses) tương tự với cấu trúc của lời yêu cầu HTTP. Lời phản hồi bao gồm thông tin trạng thái chỉ định lời yêu cầu có thực hiện thành công hay không và dữ liệu kết quả trả về.
* API REST và truy vấn (query): 3 loại APIs được cung cấp bởi dịch vụ hạ tầng AWS: REST, Query, và SOAP. Với mỗi loại cung cấp những thế mạnh khác nhau của việc tương tác với dịch vụ.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dịch vụ** | **REST API** | **Query API** | **SOAP API** |
| S3 | Có | Không | Có |
| EC2 | Không | Có | Có |
| SQS | Có | Có | Có |
| FPS | Không | Có | Có |
| SimpleDB | Không | Có | Có |

Bảng 2-2. Các API hỗ trợ bởi dịch vụ AWS

* Tài liệu XML: AWS dùng tài liệu XML để mô tả thông tin có cấu trúc để gửi và nhận trong một dịch vụ.
  + - * 1. Chứng thực người dùng:

AWS đề nghị thông điệp lời yêu cầu API phải có chữ ký điện tử được ký bởi chủ tài khoản AWS. Dịch vụ này dùng chữ kỹ này để nhận dạng người gửi và để đảm bảo rằng lời yêu cầu (request) không bị thay đổi trên đường đi. Chữ ký xác nhận được gắn vào lời yêu cầu và là một phần quan trọng trong quá trình trao đổi thông tin khi sử dụng dịch vụ AWS.

Với mỗi tài khoản người dùng AWS có quan hệ với bộ giấy chứng nhận và bạn dùng nó để ký vào thông điệp lời yêu cầu REST hoặc Query.Những giấy chứng nhận này được gọi là định danh khóa truy cập AWS (AWS Access Key Identifiers), được tạo ra bởi một cặp giá trị bao gồm một mã khóa truy cập (Access Key ID) và một khóa truy cập bảo mật (Secret Access Key). Mã khóa truy cập định nghĩa chủ tài khoản là người thực hiện lời yêu cầu, khóa truy cập bảo mật được dùng để tính toán một chữ ký điện tử cho những lời yêu cầu.

Để ký cho các lời yêu cầu API REST hoặc API Query, bạn phải phát sinh một khóa là Hash Message Authentication Code (HMAC) xác thực những lời yêu cầu. Điều này có nghĩa là mỗi thông điệp lời yêu cầu (request message) được tóm tắt vào giá trị hash rồi được ký bởi giấy chứng nhận khóa truy cập bảo mật quan hệ với tài khoản AWS. Kết quả của tính toán HMAC là chữ ký điện tử.

Sau đây là 3 bước yêu cầu ký trên một thông điệp lời yêu cầu AWS:

1. Phát sinh chuỗi mô tả lời yêu cầu chính là tóm tắt thông điệp lời yêu cầu.
2. Ký trên chuỗi này dùng một chữ ký số HMAC với khóa bảo mật của bạn.
3. Gắn chữ ký vào thông điệp lời yêu cầu trước khi gửi nó đến Amazon.

*Mô tả lời yêu cầu (request descriptions):*

Bản mô tả lời yêu cầu là bản tóm tắt nội dung thông điệp lời yêu cầu AWS. Khi một dịch vụ Amazon nhận một thông điệp lời yêu cầu đã được ký, nó phát sinh một bản tóm tắt thông điệp và tính toán giá trị chữ ký cần thiết để chứng mình rằng lời yêu cầu này bắt nguồn từ một tài khoản người dùng trong dịch vụ. Nếu giá trị chữ ký được tính toán bởi dịch vụ không khớp với giá trị chữ ký được cung cấp trong lời yêu cầu thì lời yêu cầu được xem là không hợp lệ và bị trừ chối.

Quy tắc việc tạo mô tả lời yêu cầu là giá trị chuỗi thay đổi phụ thuộc vào giao diện AWS mà bạn dùng. Trong một vài trường hợp thì quy tắc này sẽ phụ thuộc vào dịch vụ mà bạn gửi thông điệp đến.

*Mô tả yêu cầu chữ ký (*signing request Descriptions*):*

Bước tiếp theo trong việc chứng thực thông điệp yêu cầu AWS của bạn là ký trên chuỗi mô tả lời yêu cầu dùng HMAC với thuật toán SHA-1(thuật toán băm an toàn). Trong cùng quá trình thì thường phát sinh giá trị chữ ký số cho cả hai REST và Query.

Sau đây là các bước bạn phải làm để phát sinh một chữ ký lời yêu cầu (request signature) từ một chuỗi mô tả lời yêu cầu:

* Tạo đối tượng thuật toán chữ ký dùng thuật toán SHA-1.
* Chỉ dẫn thuật toán dùng giấy chứng nhận khóa bảo mật AWS là khóa bảo mật cho thuật toán.
* Tính toán chữ ký HMAC của chuỗi mô tả lời yêu cầu. Chuỗi mô tả phải được mã hóa định dạng Unicode UTF-8.
* Mã hóa dữ liệu chữ ký nhị phân đến chuỗi văn bản Base-64.

*Các thuộc tính dịch vụ thông thường:*

Để tạo lời yêu cầu và tương tác được chứng thực với AWS, triển khai dịch vụ khách (service client implementations) phải biết giấy chứng nhận tài khoản AWS của mình. Sau đây là năm thuộc tính khóa mà người cài đặt dịch vụ khách phải biết để làm việc:

* Khóa truy cập AWS (AWS Access Key).
* Khóa bảo mật AWS (AWS Secret Key).
* HTTPS an toàn: Giao thức HTTPS an toàn nên được dùng thay thế giao thức HTTP khi liên lạc với AWS.
* Có thể sửa lỗi: Có thông tin nhật ký chi tiết của việc giao tiếp dịch vụ, hỗ trợ cho việc sửa lỗi.
* Time offset: Một giá trị kiểu nguyên chỉ định thời gian khác nhau giữa dịch vụ và hệ thống cục bộ của bạn. Offset này sẽ được dùng khi phát sinh mô tả lời yêu cầu, vì vậy hệ thống cục bộ với thời gian không đúng sẽ vẫn có thể tương tác với AWS.
  + - * 1. Thực thi lời yêu cầu AWS

Để thực thi thao tác API thông qua giao diện REST hoặc Query của AWS, triển khai dịch vụ khách (service client implementations) sẽ thực hiện các bước sau:

* Tạo URI chỉ định tài nguyên dịch vụ yêu cầu sẽ thực hiện.
* Tạo một thông điệp lời yêu cầu bao gồm tất cả thông tin dịch vụ sẽ cần thực hiện.
* Phát sinh chuỗi mô tả lời yêu cầu chính là tóm tắt thông điệp lời yêu cầu.
* Chứng thực lời yêu cầu bởi việc phát sinh chữ ký từ chuỗi mô tả lời yêu cầu và gắn chúng vào lời yêu cầu.
* Chuyển giao thông điệp yêu cầu HTTP tới dịch vụ và nhận phản hồi.
* Phân tích phản hồi được gửi bởi dịch vụ và có được thông tin kết quả chỉ định của những thao tác đó.

Dịch vụ lưu trữ đơn giản (Simple Storage Service S3):

Dịch vụ lưu trữ đơn giản (S3) cung cấp không giới hạn không gian lưu trữ online cho bất kì loại tập tin hay loại dữ liệu nào. Thông tin lưu trữ trên S3 được truy cập từ bất kỳ đâu khi có mạng Internet và được duy trì trong hệ thống chắc chắn cao [16].

* + - * 1. Tổng quan:

Mô hình dữ liệu của S3 rất đơn giản, bao gồm chỉ có hai loại tài nguyên lưu trữ: đối tượng và thùng chứa (buckets). Đối tượng lưu trữ dữ liệu và siêu dữ liệu (metadata), và thùng chứa là vật chứa có thể lưu giữ không giới hạn đối tượng. Tính đơn giản này của hệ thống giúp hệ thống có thể linh động và dễ dàng lắp ghép phù hợp với những mục đích khác nhau, tuy nhiên nó cũng có nghĩa là nếu bạn cần thực hiện một công việc phức tạp, bạn phải tạo một chương trình thông minh để bổ sung các tính năng còn thiếu trong mô hình dữ liệu này.

Ngoài ra lưu trữ dữ liệu S3 cung cấp cơ cấu điều khiển truy cập cho phép bạn lưu trữ thông tin của bạn ở chế độ riêng tư hoặc chia sẻ công khai và có thể truy cập với bất kỳ ai trên Internet. Bộ cài đặt điều khiển truy cập được cấu hình bằng cách dùng danh sách các luật để mô tả. Bộ cài đặt điều khiển truy cập có thể được áp dụng đến cả hai tài nguyên thùng chứa và đối tượng.

Tài nguyên trong S3 được đồng nhất bởi chuẩn Universal Resource Identifiers (URIs). Điều này có nghĩa là đối tượng truy cập công cộng có thể được tải từ URI, dạng là http://s3.amazonaws.com/bucket-name/object-name. S3 cũng cho phép tài nguyên được truy cập dùng tên miền truy cập khác. Chức năng này cho phép bạn công khai đường dẫn đến tài nguyên của bạn dựa trên tên miền, dạng như là http://www.mysite.com/object-name.

* + - * 1. Kiến trúc:

S3 được xây dựng trên kiến trúc phân tán bên trong Amazon. Dữ liệu của bạn được lưu trữ một cách phân tán qua nhiều máy chủ vật lý. Nếu bạn muốn, bạn có thể chỉ định vị trí địa lý nơi mà thùng chứa (bucket) của bạn sẽ được lưu trữ. Chiến lược lưu trữ này cung cấp những lợi ích to lớn như là đáng tin cậy, dễ dàng thay đổi,… tuy nhiên nó cũng có những trở ngại mà bạn phải cân nhắc khi xây dựng ứng dụng sử dụng S3.

* Đối tượng S3 không thể thao tác thủ công giống như tập tin thông thường: Không giống như tập tin trên đĩa, một đối tượng không thể thay đổi tên hay thay đổi vị trí trong S3. Nó cũng không thể thay đổi một phần nhỏ của một dữ liệu của đối tượng. Nếu bạn cần thay đổi đối tượng, chỉ có cách là bạn tải lên một đối tượng đã được thay đổi và thay thế đối tượng muốn thay đổi.
* Thời gian lan truyền: bởi vì dữ liệu S3 của bạn được phân tán trên nhiều máy chủ và nhiều trung tâm dữ liệu, nó có thể tốn thời gian thay đổi để phân tán dữ liệu đến những vị trí đó.
* Những lời yêu cầu S3 thỉnh thoảng gây lỗi: S3 sẽ thỉnh thoảng thất bại trong quá trình xử lý yêu cầu. Ứng dụng khách sử dụng S3 phải được thiết kế để xử lý những lỗi này.
* Địa chỉ IP của S3 có thể thay đổi theo thời gian: để đảm bảo S3 duy trì tính sẵn sàng cao, địa chỉ IP quan hệ với tên DNS S3, dạng như s3.amazonaws.com, có thể được cập nhật theo thời gian. Chương trình khách S3 hoạt động trong khoảng thời gian dài nên phân tích tên DNS dịch vụ định kì để nhận được địa chỉ IP mới nhất.
  + - * 1. Tương tác với S3:
* *REST API*

API của S3 được xây dựng thông qua hai giao diện: REST và SOAP. Trong cuốn luận văn này chúng em chỉ đề cập đến giao thức REST. API REST cho dịch vụ S3 dùng năm phương thức HTTP để thực thi thao tác API: GET, HEAD, PUT, DELETE, và POST. Điều này có nghĩa là mỗi phương thức có những thế mạnh, phụ thuộc vào loại tài nguyên S3 để thao tác: đối tượng, thùng chưa, danh sách điều khiển truy cập (Access Control List ACL), hoặc bản thân dịch vụ S3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tài nguyên** | **GET** | **HEAD** | **PUT** | **DELETE** | **POST** |
| Dịch vụ S3 | Lấy danh sách thùng chứa (buckets) | - | - | - | - |
| Thùng chứa | Lấy danh sách đối tượng của thùng chứa | - | Tạo thùng chứa | Xóa thùng chứa | - |
| Đối tượng | Lấy dữ liệu và siêu dữ liệu (metadata) của đối tượng | Lấy siêu dữ liệu của đối tượng | Tạo hoặc thay thế đối tượng | Xóa đối tượng | Tạo hoặc thay thế đối tượng |
| ACL (cho một thùng chưa hoặc đối tượng) | Lấy cài đặt ACL | - | Cập nhật cài đặt ACL mới | - | - |

Bảng 2-3. Hoạt động của những tài nguyên S3 với phương thức HTTP

Bảng 1.2 liệt kê một vài thao tác bạn có thể thực hiện trên tài nguyên S3 dùng các loại phương thức khác nhau.

* *Xây dựng URI trong S3:*

Các tài nguyên trong dịch vụ S3 được định danh bằng URI bao gồm ba thành phần:

* Vị trí dịch vụ S3 và giao thức giao tiếp (HTTP hoặc HTTPS).
* Lời yêu cầu tài nguyên S3 phải có đích, như là một thùng chứa hoặc một đối tượng.
* Tham số lời yêu cầu là tùy chọn được dùng để định danh tài nguyên đặc biệt, ví dụ là bộ cài đặt điều khiển truy cập quan hệ với một tài nguyên.

Để thực thi một hành động trên tài nguyên S3, bạn phải tạo một chuỗi URI định danh tài nguyên.

|  |  |
| --- | --- |
| **Tài nguyên** | **Ví dụ URI** |
| Dịch vụ S3 | http://s3.amazonaws.com/ |
| Thùng chưa | http://s3.amazonaws.com/bucket\_name |
| Đối tượng | http://s3.amazonaws.com/bucket\_name/objectkey |
| ACL | http://s3.amazonaws.com/bucket\_name/objectkey?acl |
| Đối tượng thông qua tên miền con S3 | http://bucketname.s3.amazonaws.com/objectkey |
| Đối tượng thông qua tên máy chủ ảo | http://www.mydomain.com/objectkey |

Bảng 2-4. Những ví dụ URI tài nguyên S3.

Điện toán đám mây co dãn (Elastic Compute Cloud EC2):

Elastic Compute Cloud (EC2) cung cấp một môi trường cho việc thực thi máy chủ ảo. Bạn có thể quản lý mỗi máy chủ ảo giống như máy vật lý, cài đặt phần mềm bạn cần và cấu hình nó làm việc theo cách bạn muốn. Dịch vụ này cho phép bạn tạo một “hồ máy chủ” có thể thay đổi kích thước (resizable pool of servers) cho việc xử lý tính toán công việc. Bạn có thể bắt đằu với nhiều máy chủ ảo cần thiết để thực thi nhiệm vụ, tăng hoặc giảm số lượng máy chủ, và ngừng tất cả chúng khi nhiệm vụ hoàn thành [17].

* + - * 1. Tổng quan EC2:

Dịch vụ EC2 bao gồm ba thành phần chính thực thể, môi trường, hình ảnh máy Amazon.

* Thực thể (Instances): Thực thể EC2 là máy ảo chạy trong môi trường EC2 và thực hiện nhiệm vụ tính toán.
* Môi trường: Thực thể chạy trong môi trường EC2, cái mà cung cấp điều khiển truy cập có thể cấu hình, dữ liệu có ngữ cảnh, và thông tin khác cái mà thực thể (instance) cần để làm việc.
* Hình ảnh máy Amazon (Amazon Machine Images): AMI là những tập tin lưu giữ ảnh chụp (snapshot) của thực thể EC2 vào một thời điểm nào đó trong thời gian, có bao gồm cả những phần mềm, cấu hình, dữ liệu. Những hình ảnh này phục vụ khởi động ổ đĩa cho thực thể.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tài nguyên** | **Nhỏ** | **Lớn** | **Rất lớn** |
| Nền tảng | 32-bit x86 | 64-bit x86 | 64-bit x86 |
| Tỉ lệ CPU | 1 ECU (1 virtual core) | 4 ECUs (2 virtual cores of 2 ECUs each) | 8 ECUs (4 virtual cores of 2 ECUs each) |
| Bộ nhớ (RAM) | 1.7 GB | 7.5 GB | 15 GB |
| Lưu trữ | 150 GB | 840 GB (two 420 GB partitions) | 1680 GB (four 420 GB partitions) |
| Lưu trữ (phân vùng gốc) | 10 GB | 10 GB | 10 GB |
| Hiệu suất nhập/xuất | Trung bình | Cao | Cao |
| Tên loại thực thể | m1.small | m1.large | m1.xlarge |

Bảng 2-5. Những loại thực thể EC2

Bảng 1.3 liệt kê ba loại thực thể EC2 được cung cấp.Tùy theo nhu cầu mà bạn có thể chọn loại thực thể phù hợp.

* + - * 1. Cặp khóa (keypairs):

EC2 cho phép lưu trữ cặp khóa công khai hoặc riêng tư quan hệ với tài khoản AWS trong môi trường EC2. Dịch vụ này sẽ phát sinh một cặp khóa dựa vào yêu cầu và sẽ cung cấp cho bạn khóa riêng tư chỉ có thể tải được một lần duy nhất. Dịch vụ này lưu giữ bên trong bản sao chép của khóa công cộng (public key) tương ứng với khóa riêng tư (private key). Khi bạn khởi tạo một thực thể trong dịch vụ, bạn có thể yêu cầu môi trường cung cấp khóa công cộng từ một trong số cặp khóa của bạn cho thực thể này như là dữ liệu theo ngữ cảnh. Dùng cặp khóa này để đảm bảo thực thể chỉ có thể truy cập được bởi người có khóa riêng tư.

Công việc khó khăn của việc tạo một cặp khóa được thực hiện bởi thao tác CreateKeyPair. Thao tác này hướng dẫn EC2 phát sinh cặp khóa công khai- riêng tư 2,048-bit dùng thuật toán mã hóa RSA và chỉ định tên cho cặp khóa này. Thành phần khóa công khai được lưu trữ trong môi trường EC2 của bạn và có thể được tham chiếu đến bằng dùng tên đã được chỉnh định cho cặp khóa. Khóa riêng được cung cấp cho bạn trong tin nhắn trả lời của hành động. Trách nhiệm của bạn là lưu trữ khóa riêng tư, giữ cho nó an toàn, và cung cấp nó khi bạn đăng nhập vào thực thể (instance) của bạn.

Thao tác DeleteKeyPair mục đích xóa cặp khóa với tên chỉ định trong tài khoản EC2 của bạn.

Dịch vụ hàng đợi đơn giản (Simple Queue Service SQS):

Dịch vụ hàng đợi đơn giản (Simple Queue Service SQS) cung cấp nơi lữu trữ chắc chắn và chuyển giao thông điệp giữa bất kì khách (client) hoặc máy tính thông qua Internet. Nó cho phép người gửi và người nhận thông điệp tương tác với nhau mà không cần phải giao tiếp trực tiếp và không cần yêu cầu mỗi bên phải sẵn sàng hoặc phải được kết nối mạng [18].

SQS kết hợp những thế mạnh của kiến trúc thông điệp được quy ước-mềm dẻo-giao tiếp chịu lỗi-với một cơ sở hạ tầng phân tán, linh hoạt và đáng tin cậy.

SQS được tạo bởi hai tài nguyên chính là: thông điệp và hàng đợi. Thông điệp là một mẫu dữ liệu theo ngữ cảnh có kích thước lên tới 256 KB có thể được gửi đến SQS và lưu trữ trong dịch vụ cho đến khi nó được chuyển giao tới một hoặc nhiều người nhận. Hàng đợi, những thông điệp được lưu trữ trong những hàng đợi. Hàng đợi phân phát những thông điệp liên quan với nhau tới một nhóm và cung cấp những tùy chọn cấu hình cho việc chuyển giao thông điệp và điều khiển truy cập.

SQS được cài đặt dưới dạng một hệ thống phân tán, sao chép mỗi thông điệp được lưu trữ ở nhiều máy chủ vật lý và có thể qua nhiều trung tâm dữ liệu. Chiến lược này cung cấp những lợi ích bên cạnh đó nó cũng mang lại những bất lợi cần tránh khi thiết kế ứng dụng dựa trên SQS.

* Thông điệp được nhận có thể không phải là kết quả trả về chính xác.
* Thông điệp có thể không được chuyển giao nhanh chóng.
* Thông điệp có thể chuyển giao bị hư hỏng.
* Thông điệp có thể tái chuyển giao gây ra trùng lắp thông điệp.
  + - * 1. Hàng đợi (queue):
* Tạo hàng đợi: Thao tác CreateQueue tạo hàng đợi mới với tên cung cấp. Tên hàng đợi phải là duy nhất trong tài khoản SQS của bạn. Hoạt động trả về một thông điệp phản hồi bao gồm giá trị URL định danh duy nhất cho hàng đợi vừa tạo.
* Liệt kê danh sách hàng đợi: Thao tác ListQueues trả về danh sách các hàng đợi trong tài khoản SQS của bạn. Danh sách này có thể đươc giới hạn để chỉ bao gồm những hàng đợi với tên bắt đầu là tiền tố chỉ định.
* Xóa hàng đợi: Thao tác DeleteQueue xóa một hàng đợi trong tài khoản SQS của bạn. Một cách an toàn, SQS sẽ không cho phép bạn xóa một hàng đợi có bao gồm thông điệp; tuy nhiên, bạn có thể viết lại chức năng an toàn này và cho phép dịch vụ xóa những hàng đợi có thông điệp.
  + - * 1. Thông điệp (message):

Một thông điệp được định danh duy nhất bởi một giá trị ID được phát sinh bởi dịch vụ. Một định danh duy nhất của thông điệp bao gồm URL của hàng đợi và ID thông điệp do đó sẽ là định danh duy nhất cho thông điệp trong SQS. Thông điệp có kích thước tối đa là 256 KB. Nội dung thông điệp SQS phải là ký tự mã Unicode và theo chuẩn XML, được định nghĩa trong XML 1.0. Nếu bạn cần phải truyền tải thông tin nhị phân, hoặc bạn có nghi ngờ về việc liệu tin nhắn của bạn sẽ phù hợp với những hạn chế bộ ký tự, bạn nên mã hóa dữ liệu thông điệp của bạn bằng cách sử dụng Base 64.

* Gửi thông điệp: Thao tác SendMessage tạo một thông điệp mới trong hàng đợi SQS. SQS Query API cho phép thông điệp gửi được dưới dạng HTTP GET hoặc POST. Kích thước cho thông điệp gửi dùng GET là chỉ 8 KB.
* Nhận thông điệp: Thao tác ReceiveMessage cho phép một khách SQS (SQS client) nhận thông điệp được gửi tới một hàng đợi chỉ định. Thao tác này cho phép bạn chỉ định giới hạn tối đa số thông điệp mà bạn muốn nhận; tuy nhiên, SQS không đảm bảo ràng sẽ cung cấp số lượng thông điệp đúng yêu cầu.
* Xóa thông điệp: Thao tác DeleteMessage cho phép xóa một thông điệp chỉ định trong hàng đợi. Thông điệp sẽ bị xóa ngay lập tức, bất kể trạng thái hoặc nó đã được chuyển giao đến bến kì khách SQS nào hay chưa (SQS client).
* Xem thông điệp: Thao tác PeekMessage cho phép bạn có thể xem nội dung thông điệp bất chấp trạng thái của nó. Bạn chỉ có thể thực hiện thao tác này khi bạn biết định danh của thông điệp đó.

Dịch vụ thanh toán linh động (Flexible Payments Service FPS):

* + - * 1. Tổng quan:

Dịch vụ thanh toán linh động (Flexible Payments Service FPS) chuyển giao tiền giữa cá nhân hoặc công ty có tài khoản thanh toán Amazon với nhau. Tiền có thể được thanh toán từ một khoảng dư thanh toán Amazon, một thẻ tín dụng hoặc tài khoản ngân hàng và được nạp vào tài khoản thanh toán Amazon của người nhận, với chi phí rẻ hơn bất kỳ khoản phí giao dịch nào. Dịch này này cho phép nhà phát triển ứng dụng mạng xây dựng hệ thống của mình cho việc nhận thanh toán từ khách hàng hoặc tạo thị trường ứng dụng là trung gian giao dịch giữa các bên thứ ba [19].

Dịch vụ FPS cho phép thực hiện giao dịch trong ba bước:

1. Thanh toán được định nghĩa bằng việc dùng những quy tắt để mô tả điều kiện phải được đáp ứng khi giao dịch. Những quy định này được sử dụng để kiểm soát các loại giao dịch được phép, số tiền sẽ được thanh toán, thanh toán cho ai, và các điều kiện khác.
2. Tất cả những người tham gia trong một giao dịch đồng ý với các quy định thanh toán và phê duyệt giao dịch.
3. FPS kiểm tra các điều kiện quy tắc thanh toán được đáp ứng sau đó thực hiện các giao dịch chuyển tiền.
   * + - 1. Tương tác với FPS:

Để cho phép các nhà phát triển để tạo ra các ứng dụng FPS mà không sợ tác dụng phụ tai hại từ lỗi hoặc sự hiểu lầm, Amazon cung cấp hai môi trường FPS khác nhau: sản xuất (production) và sandbox.

* Môi trường sandbox: Là một hệ thống FPS đầy đủ chức năng để phát triển và thử nghiệm mà không cần chuyển tiền hoặc phải chịu phí giao dịch thực tế. Sandbox làm cho nó có thể kiểm tra tất cả các khía cạnh của FPS, từ các hoạt động API và bên thứ ba. Khi một ứng dụng đã được kiểm tra kỹ lưỡng trong môi trường sandbox thì có thể chuyển sang môi trường sản xuất để thực thi thực tế.
* Môi trường sản xuất: Giao dịch thực hiện trong môi trường này sẽ chuyển tiền giữa các tài khoản thanh toán Amazon, rút tiền trên thẻ tín dụng hoặc tài khoản ngân hàng, phải chịu phí giao dịch, và cũng có thể chậm trễ. Để thực hiện thử nghiệm các giao dịch trong môi trường này, bạn sẽ cần khách hàng, đồng nghiệp là những người sẵn sàng giúp bạn thực hiện giao dịch với tiền của họ.

API dịch vụ mạng FPS được xây dựng thông qua hai giao diện giao tiếp API: Query và SOAP.

Cơ sở dữ liệu DynamoDB:

Chúng em sẽ trình bày chi tiết phần sau.

*Bạn có thể tham khảo ở [34].*

* + 1. Cơ sở dữ liệu NoSQL - Amazon DynamoDB

Tổng quan:

Amazon DynamoDB là một dịch vụ đầy đủ và nhanh chóng về quản lý cơ sở dữ liệu NoSQL, làm cho việc quản lý cở sở dữ liệu (CSDL) trở nên đơn giản với chi phí hiệu quả để lưu trữ và truy vấn bất kỳ về số lượng dữ liệu, và phục vụ bất kỳ mức độ về yêu cầu thông lượng. Tất cả các dữ liệu được lưu trữ trên ổ đĩa solid state (SSD), và được nhân rộng trên 3 khu vực (availability zone) để có thể sẵn sàng và có độ bền cao. Với DynamoDB, bạn có thể giảm tải gánh nặng quản lý hoạt động và mở rộng quy mô của một cụm cơ sở dữ liệu phân tán cao có sẵn, trong khi phải trả một mức giá thấp cho những gì bạn sử dụng.

Amazon DynamoDB được thiết kế để giải quyết các vấn đề cốt lõi của quản lý cơ sở dữ liệu, hiệu suất, khả năng mở rộng, và độ tin cậy. Nhà phát triển có tạo ra một bảng cơ sở dữ liệu có thể lưu trữ, lấy bất kỳ số lượng dữ liệu, và phục vụ bất kỳ mức độ yêu cầu lưu lượng.

Amazon DynamoDB cho phép khách hàng có thể giảm tải gánh nặng quản trị của việc xử lý và mở rộng quy mô một cụm cơ sở dữ liệu phân tán có sẵn trong khi chỉ phải trả một giá biến thấp cho các nguồn lực mà họ tiêu thụ [21].

Điểm nổi bật dịch vụ:

Amazon DynamoDB là một dịch vụ cơ sở dữ liệu NoSQL quản lý đầy đủ, cung cấp hiệu suất nhanh và dự đoán với khả năng mở rộng liền mạch. Với một vài cú nhấp chuột trong giao diện quản lý AWS, khách hàng có thể tạo ra một bảng cơ sở dữ liệu DynamoDB mới, mở rộng quy mô lên hoặc xuống theo công suất yêu cầu của họ đối với bảng mà không làm suy giảm tuổi thọ hay hiệu suất. DynamoDB cho phép khách hàng có thể giảm tải gánh nặng hoạt động hành chính và co dãn quy mô cơ sở dữ liệu phân phối đến AWS, để họ không phải lo lắng về lập dự phòng phần cứng, cài đặt, cấu hình, nhân rộng, phần mềm vá, hoặc co dãn khối [22].

Sau đây là một số tính năng DynamoDB chính:

* **Khả năng mở rộng -** DynamoDB được thiết kế thông liền mạch và co dãn lưu trữ.
  + *Được cung cấp thông lượng*
  + *Tự động co dãn lưu trữ***:** Không có giới hạn về số lượng dữ liệu bạn có thể lưu trữ trong một bảng DynamoDB, và các dịch vụ tự động phân bổ lưu trữ nhiều hơn, như bạn lưu trữ nhiều dữ liệu hơn bằng cách sử dụng DynamoDB viết API.
* **Nhanh chóng, dự đoán hiệu suất -** Trung bình thời gian trễ dịch vụ cho DynamoDB thường là một con số mili giây. Dịch vụ chạy trên các ổ đĩa trạng thái rắn, và được xây dựng để duy trì ổn định, độ trễ nhanh chóng ở bất kỳ quy mô.
* **Dễ dàng quản lý -** DynamoDB là một dịch vụ quản lý đầy đủ - bạn chỉ cần tạo một bảng cơ sở dữ liệu và để cho các dịch vụ xử lý phần còn lại. Bạn không cần phải lo lắng về phần cứng hoặc phần mềm cung cấp, cài đặt và cấu hình, phần mềm vá, hoạt động, cụm cơ sở dữ liệu phân phối đáng tin cậy.
* **Được xây dựng trong khả năng chịu lỗi -** DynamoDB đã được xây dựng trong khả năng chịu lỗi, tự động và đồng bộ sao chép dữ liệu của bạn trên nhiều khu sẵn có trong một khu vực để có thể sẵn sàng và để giúp bảo vệ dữ liệu của bạn.
* **Linh hoạt -** DynamoDB không có một sơ đồ cố định. Thay vào đó, mỗi mục dữ liệu có thể có một số khác nhau về các thuộc tính. Các loại dữ liệu (chuỗi, số, nhị phân, và bộ) tạo thêm sự phong phú cho các mô hình dữ liệu.
* **Hiệu quả chỉ mục -** Tất cả các mục trong một bảng DynamoDB được xác định bởi một khóa chính, cho phép bạn truy cập vào các dữ liệu một cách nhanh chóng và hiệu quả. Bạn cũng có thể xác định chỉ số thứ cấp (secondary index) vào các thuộc tính không khóa, và truy vấn dữ liệu của bạn bằng cách sử dụng khóa luân phiên (alternate key).
* **Tính nhất quán mạnh mẽ -** Không giống như nhiều cơ sở dữ liệu không quan hệ, DynamoDB làm cho phát triển dễ dàng hơn bằng cách cho phép bạn sử dụng thống nhất mạnh mẽ về đọc để đảm bảo bạn luôn luôn đọc các giá trị mới nhất. DynamoDB hỗ trợ nhiều loại dữ liệu cơ sở (số, chuỗi, nhị phân, và các thuộc tính đa giá trị).
* **Chi phí hiệu quả-** DynamoDB được thiết kế cực kỳ hiệu quả chi phí cho khối lượng công việc của bất kỳ quy mô. Bạn có thể bắt đầu với một lớp miễn phí cho phép hơn 40 triệu hoạt động cơ sở dữ liệu mỗi tháng, và trả giá theo giờ thấp và chỉ cho các tài nguyên bạn tiêu thụ trên giới hạn đó.
* **An toàn-** DynamoDB là an toàn và sử dụng phương pháp mã hóa để xác thực người dùng và ngăn chặn truy cập dữ liệu trái phép.
* **Tích hợp giám sát-** DynamoDB hiển thị số liệu hoạt động chính cho bảng trong Management Console AWS. Dịch vụ này cũng tích hợp đồng hồ đám mây (cloudwatch) để bạn có thể nhìn thấy yêu cầu thông lượng và độ trễ cho mỗi bảng DynamoDB, và dễ dàng theo dõi tiêu thụ tài nguyên của bạn.

Mô hình dữ liệu:

* + - * 1. Các khái niệm mô hình dữ liệu – bảng, mục và thuộc tính:

Các khái niệm mô hình dữ liệu DynamoDB bao gồm các bảng, các mục và các thuộc tính.

Trong Amazon DynamoDB, một cơ sở dữ liệu là một tập hợp các bảng. Một bảng là một bộ sưu tập các mục (item) và mỗi mục là một tập hợp các thuộc tính (attribute).

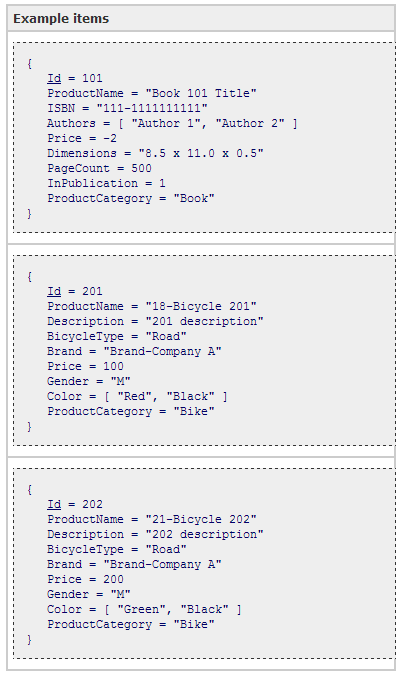
Trong một cơ sở dữ liệu quan hệ, một bảng có một giản đồ được xác định trước như tên bảng, khóa chính, danh sách các tên cột và kiểu dữ liệu của họ. Tất cả các dòng dữ liệu lưu trữ trong bảng phải có bộ các cột. DynamoDB là một cơ sở dữ liệu NoSQL: Ngoại trừ các khóa chính cần thiết, một bảng DynamoDB là lược đồ nhỏ (schema-less). Kích thước mục là tổng độ dài của tên thuộc tính và giá trị (nhị phân và độ dài UTF-8) của nó.

Mỗi thuộc tính trong một mục là một cặp khóa-giá trị (key-value). Một thuộc tính có thể là giá trị đơn hoặc là đa giá trị.

Ví dụ, một mục cuốn sách có thể có thuộc tính tiêu đề và tác giả. Mỗi cuốn sách có một tên nhưng có thể có nhiều tác giả. Thuộc tính đa giá trị là một tập hợp; giá trị nhân bản không được phép.

Ví dụ, hãy xem xét lưu trữ một danh mục các sản phẩm trong DynamoDB. Bạn có thể tạo ra một bảng ProductCatalog với thuộc tính Id là khóa chính của nó.



Bạn có thể lưu trữ các loại mặt hàng sản phẩm trong bảng. Bảng sau đây cho thấy các mặt hàng mẫu.

Trong ví dụ, bảng ProductCatalog có một mục sách, hai mục xe đạp. Mục 101 là sách với nhiều thuộc tính bao gồm đa giá trị tác giả. Mục 201, 202 là xe đạp, những mục này có đa giá trị màu sắc. Các giá trị thuộc tính được hiển thị theo cú pháp JSON

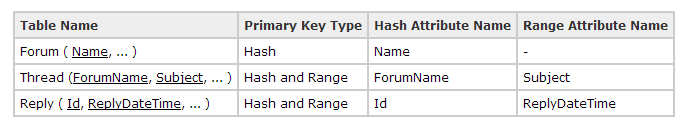
DynamoDB không cho phép giá trị thuộc tính chuỗi rỗng hoặc null.

* + - * 1. Khóa chính

Khi tạo bảng, ngoài tên bảng thì bạn phải chỉ định khóa chính của bảng. DynamonDB hỗ trợ 2 loại khóa chính

* Khóa chính loại băm (hash): Trong trường hợp này khóa chính có 1 thuộc tính, thuộc tính băm. DynamonDB xây dựng chỉ mục băm trên thuộc tính khóa chính. Ví dụ trên thì thuộc tính băm cho bảng ProductCatalog là ID.
* Khóa chính loại miền và băm (hash and range): Trong trường hợp này khóa chính có 2 thuộc tính. Thuộc tính thứ nhất là thuộc tính băm, thuộc tính thứ hai là thuộc tính miền. DynamoDB xây dựng chỉ mục băm trên thuộc tính khóa chính băm và chỉ mục miền sắp xếp trên thuộc tính khóa miền.

Ví dụ, Amazon Webservice duy trì một vài forum, mỗi forum có nhiều chủ đề thảo luận, và mỗi chủ đề có nhiều trả lời.



* + - * 1. Chỉ mục thứ cấp (secondary index)

Khi bạn tạo bảng với khóa miền và băm, bạn có thể tùy chọn định nghĩa một hoặc nhiều chỉ mục thứ cấp trên bảng. Một chỉ mục thứ cấp cho phép bạn truy vấn dữ liệu trên bảng dùng khóa thay phiên. Thêm vào đó bạn có thể truy vấn dựa vào khóa chính.

Ví dụ trên bảng Reply, bạn có thể truy vấn dữ liệu dựa vào ID (băm), và ReplyDateTime (băm và miền). Bạn có thuộc tính trên bảng là PostedBy với ID User của người đăng trả lời. Với chỉ mục thứ cấp trên PostedBy, bạn có thể truy vấn dữ liệu dựa vào ID và PostedBy.

DynamonDB hỗ trợ bạn 2 loại chỉ mục thứ cấp:

* Chỉ mục thứ cấp cục bộ: Chỉ mục có cùng khóa băm trong bảng nhưng khác khóa miền
* Chỉ mục thứ cấp toàn cục: chỉ mục với khóa băm và miền có thể khác trong bảng.

Bạn có thể định nghĩa 5 chỉ mục thứ cấp cục bộ và 5 chỉ mục thứ cấp toàn cục trong một bảng.

* + - * 1. Các loại dữ liệu DynamoDB

Amazon DynamoDB hỗ trợ các loại dữ liệu:

* Loại dữ liệu cơ sở: số, chuỗi và nhị phân.
* Loại đa giá trị: bộ số, bộ chuỗi, và bộ nhị phân.
* Chuỗi: Chuỗi dạng Unicode với mã hóa UTF8. Chúng không có giới hạn kích thước chuỗi. Chú ý rằng chiều dài chuỗi phải lớn hơn 0.

Ví dụ:



* Số: Số có thể số âm hoặc dương. Một số có thể lên tới 38 con số sau phần thập phân, số có thể nằm trong khoảng 10-128 đến 10+126. Số đã được chuyển đổi sang dạng chuỗi khi gửi lên DynamonDB, tuy nhiên DynamonDB xử lý chúng vẫn theo loại số với những công thức toán học.

Ví dụ:



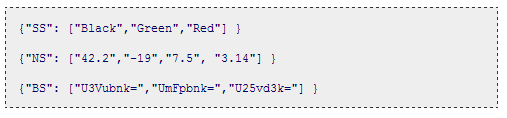
* Nhị phân: Loại nhị phân có thể lưu trữ file nhị phân, ví dụ: dữ liệu nén, dữ liệu mã hóa hoặc là ảnh… Loại này cũng không có giới hạn về kích thước giá trị nhị phân khi bạn chỉ định nó và chiều dài của thuộc tính phải lớn hơn 0.

Ví dụ sau là thuộc tính nhị phân, dùng mã hóa Base64.



* Bộ chuỗi, số, nhị phân

Ví dụ:



*Các bạn có thể tham khảo tại [23].*

Thao tác hỗ trợ trên DynamonDB:

* + - * 1. Thao tác với bảng:

DynamoDB cung cấp các hoạt động để tạo, cập nhật và xóa bảng. Sau khi bảng được tạo, bạn có thể sử dụng thao tác cập nhật bảng để tăng hoặc giảm thông lượng cung cấp cho một bảng. DynamoDB cũng hỗ trợ một hoạt động để lấy thông tin bảng (thao tác mô tả bảng (DescribeTable)) bao gồm cả tình trạng hiện tại của bảng, khóa chính khi bảng đã được tạo ra. Các thao tác lấy danh sách bảng (ListTables) cho phép bạn có được một danh sách các bảng trong tài khoản của bạn trong khu vực của các thiết bị đầu cuối bạn đang sử dụng để giao tiếp với DynamoDB.

* + - * 1. Thao tác với mục:

Thao tác mục cho phép bạn thêm, cập nhật và xóa các mục từ một bảng. Các thao tác cập nhật mục (UpdateItem) cho phép bạn cập nhật giá trị thuộc tính có sẵn, thêm các thuộc tính mới, và xóa thuộc tính hiện có từ một mục. Bạn cũng có thể thực hiện cập nhật có điều kiện. DynamoDB cung cấp một hoạt động để lấy một mục duy nhất (GetItem) hoặc nhiều mục (BatchGetItem). Bạn có thể sử dụng BatchGetItem hoạt động để lấy hàng từ nhiều bảng.

* + - * 1. Truy vấn và quét:

Các thao tác truy vấn cho phép bạn truy vấn một bảng bằng cách sử dụng thuộc tính băm và một bộ lọc nhiều tùy chọn. Nếu bảng có chỉ mục thứ cấp bạn vẫn có thể truy vấn bằng khóa. Truy vấn là cách hiệu quả nhất để lấy các mục từ một bảng hoặc từ chỉ mục thứ cấp. Bạn có thể sử dụng biểu thức điều kiện trong cả truy vấn và quét để điều khiển kết quả trả về.

* + - * 1. Đọc dữ liệu và nhất quán:
        2. Cập nhật điều kiện và truy cập đồng thời kiểm soát:

Trong một môi trường đa người dùng, điều quan trọng là để đảm bảo cập nhật dữ liệu được thực hiện bởi một khách hàng không ghi đè cập nhật được thực hiện bởi khách hàng khác. Vấn đề "mất cập nhật" là một vấn đề cơ sở dữ liệu cổ điển. Giả sử hai khách hàng đọc cùng một mục. Cả hai khách hàng có được một bản sao của mục đó từ DynamoDB. Khách hàng 1 gửi một yêu cầu để cập nhật các mục. Khách hàng 2 không biết về bất kỳ bản cập nhật. Sau đó, khách hàng 2 gửi yêu cầu riêng của mình để cập nhật các mục, ghi đè lên bản cập nhật được thực hiện bởi khách hàng 1. Do đó, bản cập nhật được thực hiện bởi khách hàng 1 bị mất.

DynamoDB hỗ trợ "ghi có điều kiện" tính năng cho phép bạn chỉ định một điều kiện khi cập nhật một mục. DynamoDB viết mục chỉ khi điều kiện quy định được đáp ứng; nếu không nó trả về một lỗi. Trong ví dụ "mất cập nhật", khách hàng 2 có thể thêm một điều kiện để xác minh giá trị mục trên phía máy chủ cũng giống như các bản sao mục trên phía máy khách. Nếu mục trên máy chủ được cập nhật, khách hàng 2 có thể lựa chọn để có được một bản cập nhật trước khi áp dụng bản cập nhật riêng của mình.

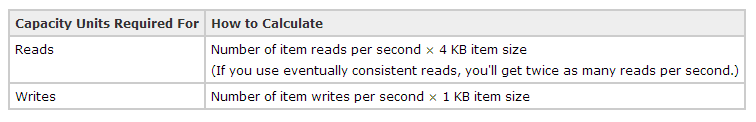
DynamoDB cũng hỗ trợ một "bộ đếm nguyên tử" tính năng mà bạn có thể gửi một yêu cầu để thêm hoặc bớt từ một giá trị thuộc tính đã tồn mà không gây trở ngại với một yêu cầu ghi khác cùng thực hiện đồng thời. Ví dụ, một ứng dụng web có thể muốn duy trì số lần truy cập của mỗi người truy cập vào trang web của mình. Trong trường hợp này, khách hàng chỉ muốn tăng giá trị bất kể những gì giá trị trước đó đã. DynamoDB hoạt động ghi hỗ trợ tăng hoặc giảm giá trị thuộc tính hiện có [24].

Thông lượng được cung cấp trên Amazon DynamonDB:

Khi bạn tạo ra hoặc cập nhật một bảng, bạn chỉ định dung tích thông lương bao nhiêu bạn muốn dành cho việc đọc và viết. DynamoDB sẽ dự trữ tài nguyên máy cần thiết để đáp ứng nhu cầu thông lượng, mà vẫn đảm bảo phù hợp, hiệu suất có độ trễ thấp.

Bảng dưới đây giải thích làm thế nào để tính toán thông lượng cung cấp mà bạn cần.

|  |  |
| --- | --- |
| **Tiêu chí** | **Cách tính** |
| Đọc | Số các mục đọc trên giây x 4 KB kích thước mục. |
| Ghi | Số các mục ghi trên giây x 1 KB kích thước mục. |



Nếu việc đọc hoặc viết các request ứng dụng của bạn vượt quá thông lượng được cung cấp cho một bảng, thì sau đó những request đó có thể được tăng cường. Bạn có thể sử dụng Management Console AWS để theo dõi thông lượng cung cấp, thông lượng thực tế và có thể thay đổi khả năng được cung cấp của bạn với dự đoán của những thay đổi lưu lượng truy cập.

* + - * 1. Đơn vị dung tích đọc:

Nếu kích thước các mục của bạn nhỏ hơn 4KB, mỗi đơn vị dung tích đọc sẽ đưa cho bạn một đọc nhất quán trên giây hoặc hai đọc nhất quán cuối cùng trên giây. Bạn không thể nhóm đa mục trong một thao tác đọc đơn mặc dù các mục có kích thước là 4KB hoặc nhỏ hơn. Ví dụ, nếu các mục của bạn là 3KB và bạn muốn đọc 80 mục trên giây từ bảng, thì bạn cần cung cấp thông lượng 80 (đọc trên giây) x 4KB (được làm tròn tới biên 4KB) = 80 đơn vị dung tích đọc.

Nếu kích thước các mục của bạn lớn hơn 4KB, bạn cần làm tròn kích thước các mục tới biên 4KB kế tiếp. Ví dụ, nếu các mục của bạn là 6KB và bạn muốn đọc 100 mỗi giây, bạn cần cung cấp thông lượng 100 (đọc trên giây) x 2 (6KB / 4KB = 1.5, làm tròn đến 2) = 200 đơn vị dung tích đọc.

Bạn có thể dùng thao tác truy vấn và quét trong DynamonDB để nhận nhiều mục liên tục nhau từ bảng trong một request đơn.

* + - * 1. Đơn vị dung tích ghi:

Nếu các mục của bạn kích thước nhỏ hơn 1KB thì mỗi đơn vị dung tích ghi sẽ đưa cho bạn 1 ghi mỗi giây. Bạn không thể nhóm đa mục trong cùng một thao tác ghi đơn mặc dù những mục đó là 1KB hoặc nhỏ hơn. Cho ví dụ, nếu những mục của bạn là 512bytes và bạn muốn ghi 100 mục trên giây đến bảng của bạn thì bạn sẽ cần cung cấp 100 đơn vị ghi.

Nếu các mục của bạn kích thước lớn hơn 1KB, bạn sẽ cần làm tròn kích thước mục đến biên 1KB kế tiếp. Ví dụ, nếu các mục của bạn là 1.5KB và bạn muốn làm 10 ghi trên giây, thì bạn sẽ cần cung cấp thông lượng 10 (viết trên giây) x 2 (1,5KB được làm tròn đến 2KB) = 20 đơn vị dung tích ghi.

*Các bạn có thể tham khảo tại [25].*

Truy cập DynamoDB:

Amazon DynamoDB là một dịch vụ web có sử dụng HTTP và HTTPS như một giao thức vận chuyển và javascript object notation (JSON) như là một định dạng tin nhắn tuần tự. Mã ứng dụng của bạn có thể có những yêu cầu trực tiếp đến dịch vụ web DynamoDB API. Thay vì làm việc yêu cầu các API DynamoDB trực tiếp từ ứng dụng của bạn, chúng tôi khuyên bạn sử dụng AWS phát triển phần mềm Kits (SDK) bao gồm các thư viện quan tâm yêu cầu chứng thực, tuần tự, và quản lý kết nối.

SDK AWS cung cấp các API cấp thấp phù hợp chặt chẽ với DynamoDB API cơ bản. Để đơn giản hóa hơn nữa việc phát triển ứng dụng, SDK cũng cung cấp các API bổ sung sau đây:

* Java và. NET SDK cung cấp các API với các mức độ trừu tượng cao hơn. Các interface cấp cao hơn cho phép bạn xác định các mối quan hệ giữa các đối tượng trong chương trình của bạn và các bảng cơ sở dữ liệu lưu trữ dữ liệu các đối tượng. Sau khi xác định bảng mapping này, bạn gọi phương thức đối tượng đơn giản. Điều này cho phép bạn lập trình hướng đối tượng chứ không phải là lập trình hướng cơ sở dữ liệu.
* NET SDK. Cung cấp các lớp helper mà bao bọc một số chức năng ở mức độ thấp API để tiếp tục đơn giản hóa code của bạn.

DynamoDB cũng cung cấp một giao diện quản lý cho phép bạn làm việc với các bảng và các mục. Bạn có thể tạo, cập nhật và xóa bảng mà không cần viết bất kỳ dòng code nào. Bạn có thể xem tất cả các mục hiện có trong một bảng hoặc sử dụng một truy vấn để lọc các mục trong bảng. Bạn có thể thêm các mục mới hoặc xóa các mục. Bạn cũng có thể sử dụng giao diện điều khiển quản lý để giám sát hiệu suất của bảng của bạn. Sử dụng số liệu đồ hồ đám mây (cloudwatch) trong giao diện điều khiển, bạn có thể theo dõi bảng thông lượng và số liệu hiệu suất khác.

* + - * 1. Khu vực và thiết bị đầu cuối cho DynamoDB:

Theo mặc định các SDK AWS và giao diện điều khiển cho DynamonDB tham chiếu đến khu vực US-West (Oregon). AWS cung cấp danh sách các khu vực phù hợp cho các vị trí trên thế giới.

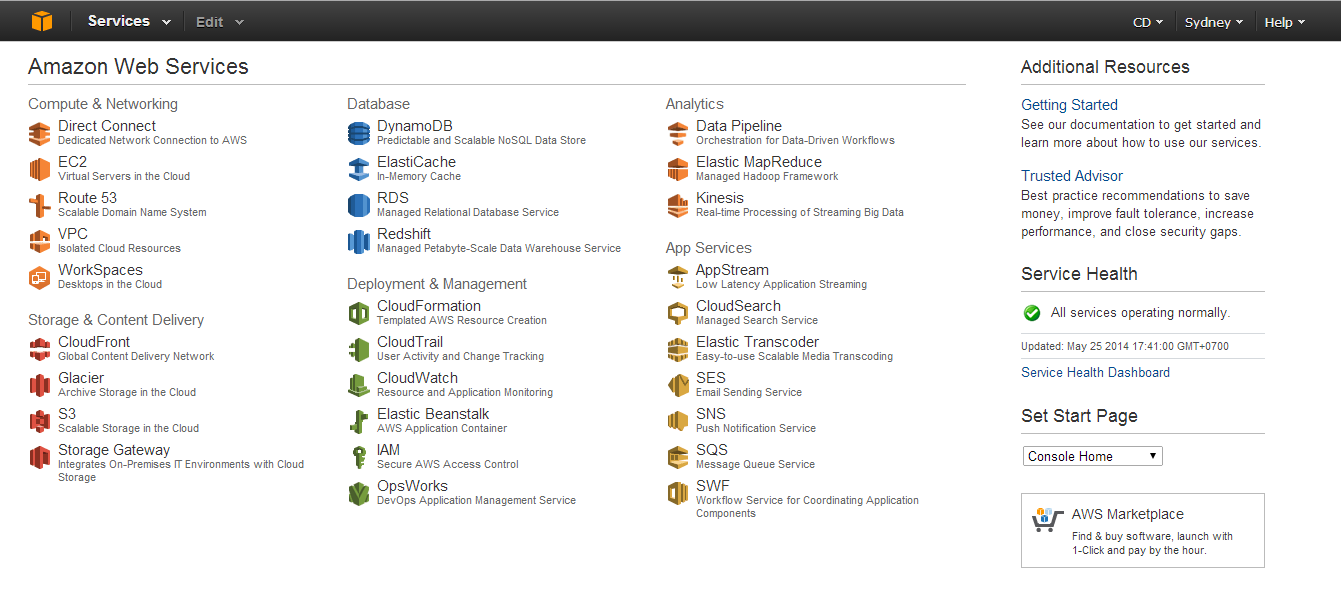
*Các bạn có thể tham khảo tại [26].*

* + 1. Công cụ phát triển

*Phần này trình bày về không gian lưu trữ, công cụ, plugin, thư viện mà đề tài này sử dụng để xây dựng ứng dụng.*

AWS Management Console:

AWS Management Console là trung tâm quản lý các dịch vụ của Amazon. Từ giao diện này, ta có sử dụng các dịch vụ của amazon đã liệt kê.



Hình 2-11. Giao diện AWS Management Console

*(Nguồn: Amazon)*

Eclipse IDE và AWS Toolkit:

Eclipse là một IDE mã nguồn mở được viết bằng java. Nó sử dụng cơ chế plugin để hỗ trợ cho việc phát triển sản phẩm trên các ngôn ngữ và framework khác nhau. Và đặc biệt hỗ trợ mạnh mẽ cho phát triển sản phẩm trên nền tảng java, android.

Sử dụng SVN để quản lý version code.

AWS Toolkit cho eclipse là plugin nguồn mở cho eclipse IDE, nó làm dễ dàng hơn cho nhà phá triển ứng dụng để phát triển, debug, và deploy ứng dụng. Một vài tính năng của AWS Toolkit là:

* + - * 1. AWS Explorer:

AWS Explorer có thể giúp bạn tương tác với nhiều dịch vụ AWS từ bên trong eclipse. Hỗ trợ dữ vụ dữ liệu bao gồm: Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB, Amazon Simple Notification Service (Amazon SNS), and Amazon Simple Queue Service (Amazon SQS). AWS Explorer cũng cung cấp truy cập đến quản lý Amazon EC2 và deployment đến AWS Elastic Beanstalk.

AWS Explorer hỗ trợ đa tài khoản AWS, bạn có thể dễ dàng thay đổi hiển thị tài nguyên trong AWS Explorer từ một tài khoản đến tài khoản khác.

* + - * 1. Amazon Elastic Compute Cloud (Amazon EC2):

Từ AWS Explorer, bạn có thể xem Amazon Machine Images (AMIs), tạo EC2 từ AMIs, kết nối đến máy ảo dùng SSH…

* + - * 1. AWS DynamoDB:

Với Toolkit, bạn có thể tạo và chỉnh sửa thuộc tính trong bảng DynamoDB và chạy thao tác quét trên bảng.

* + - * 1. AWS Elastic Beanstalk:

AWS Elastic Beanstalk làm nó một cách dễ dàng cho bạn để phát triển ứng dụng Java đến AWS.

* + - * 1. AWS SDK for Java integration:

AWS Toolkit cho Eclipse cài đặt phiên bản mới nhất của AWS SDK cho Java. Từ Eclipse bạn có thể dễ dàng chỉnh sửa, build và run bất kì sample nào.

CHƯƠNG 3. CÀI ĐẶT.

*Nội dung chương này tập trung đi sâu vào phần cài đặt chức năng chính cho ứng dụng, thiết kế giao diện, cách thiết kế và giải pháp lưu trữ dữ liệu trên DynamoDB, thiết kế kiến trúc của hệ thống với kiến trúc 3 lớp (3 Layer) và một số thành phần liên quan đến hệ thống. Chi tiết phần cài đặt bổ sung xem phụ lục phần 3.*

1. 1. Các chức năng của ứng dụng cài đặt.
      1. Module 1: Nhóm chức năng quản lý:

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Đăng ký tạo công ty | Quản trị viên ứng dụng Smart Help Desk có thể tạo ra 1 công ty mới, công ty này sẽ được cấp một tên miền hoạt động trong hệ thống. |
| Đăng ký người dùng | Người dùng có thể tạo ra một công ty mới, người dùng sẽ được cung cấp một tên miền cho công ty. |
| Phân quyền người dùng | Quản trị viên công ty có thể phân quyền được vai trò của người sử dụng, có thể cho họ là khách hàng, là nhân viên hổ trợ hoặc là quản trị viên công ty. |
| Thay đổi thông tin người dùng | Quản trị viên công ty sử dụng chức năng này để cập nhật những thay đổi trong thông tin cá nhân của người dùng. |
| Tạo phiếu thắc mắc | Chức năng này để tạo phiếu thắc mắc nào đó cần trao đổi. |
| Thay đổi thông tin phiếu thắc mắc | Quản trị viên công ty và nhân viên hổ trợ sử dụng chức năng này để thay đổi cập nhật thông tin của phiếu thắc mắc. |
| Tạo nhóm | Quản trị viên công ty thực hiện chức năng này để tạo một nhóm phục vụ cho việc trả lời các câu hỏi của khách hàng. |
| Tạo tổ chức | Quản trị viên công ty thực hiện chức năng này để tạo ra tổ chức nào đó trong công ty. |

Bảng 3-1. Nhóm chức năng quản lý

* + 1. Module 2: Nhóm chức năng tìm kiếm:

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Tìm kiếm. | Thực hiện tìm kiếm những phiếu thắc mắc theo tên, nội dung, email, tên người tạo có liên quan tới phiếu thắc mắc mà người dùng cung cấp, sau đó sẽ hiển thị kết quả lên màn hình cho người dùng. |

Bảng 3-2. Nhóm chức năng tìm kiếm

* + 1. Module 3: Nhóm chức năng thống kê – hiển thị:

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Xem thông tin phiếu thắc mắc | Người dùng chọn chức năng này để xem nội dung của phiếu thắc mắc trong quyền của mình. |

Bảng 3-3. Nhóm chức năng thống kê – hiển thị

* + 1. Module 4: Nhóm chức năng hệ thống:

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Đăng nhập | Người dùng chọn chức năng đăng nhập vào hệ thống. Tùy vào vai trò của từng người dùng sẽ tương ứng với các giao diện với các chức năng phù hợp với từng người. |
| Trả lời phiếu thắc mắc | Người dùng có thể nhập nôi dung trả lời cho phiếu thắc mắc. |

Bảng 3-4. Nhóm chức năng hệ thống

* 1. Sơ đồ usecase



Hình 3-1. Sơ đồ usecase tổng quát



Hình 3-2. Sơ đồ usecase cho khách hàng



Hình 3-3. Sơ đồ usecase cho quản trị viên hệ thống



Hình 3-4. Sơ đồ usecase cho nhân viên hỗ trợ



Hình 3-5. Sơ đồ usecase cho quản trị viên công ty



Hình 3-6. Sơ đồ usecase cho lập trình viên công ty

* + 1. Danh sách actor:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên actor** | **Ý nghĩa/Ghi chú** |
| **1** | Khách hàng | Là khách hàng của công ty, là thành phần quan trọng trong hệ thống. Là người tạo phiếu thắc mắc. |
| **2** | Nhân viên hỗ trợ | Là những nhân viên hổ trợ của công ty, được phân bổ vào các bộ phận khác nhau (nhóm), đúng với nghiệp vụ của mình để giải đáp thắc mắc cho khách hàng. |
| **3** | Quản trị viên công ty | Là những người quản lý của công ty, quản lý các nhân viên hổ trợ, khách hàng, và các phiếu thắc mắc, … |
| **4** | Quản trị viên hệ thống | Quản lý tất cả các chức năng của hệ thống SmartHelpDesk. |
| **5** | Lập trình viên | Nhà phát triển ứng dụng android. |

Bảng 3-5. Bảng danh sách Actor

* + 1. Danh sách usecase:

|  |  |  |
| --- | --- | --- |
| **Mã** | **Tên use case** | **Ý nghĩa/Ghi chú** |
| **UC1** | Đăng nhập |  |
| **UC2** | Đăng ký |  |
| **UC3** | Cập nhật thông tin tài khoản |  |
| **UC4** | Tìm kiếm phiếu thắc mắc |  |
| **UC5** | Tạo và nhận phiếu thắc mắc thông qua email |  |
| **UC6** | Xem chi tiết phiếu thắc mắc |  |
| **UC7** | Trả lời phiếu thắc mắc |  |
| **UC8** | Đánh dấu phiếu thắc mắc |  |
| **UC9** | Bỏ đánh dấu phiếu thắc mắc |  |
| **UC10** | Tạo mới phiếu thắc mắc |  |
| **UC11** | Xem phiếu thắc mắc gần đây |  |
| **UC12** | Xem phiếu thắc mắc đã đánh dấu |  |
| **UC13** | Tạo mới khách hàng | Tạo mới khách hàng |
| **UC14** | Chọn loại phiếu thắc mắc để xem |  |
| **UC15** | Sửa phiếu thắc mắc | Cập nhật thông tin phiếu thắc mắc: ủy thác, chuyển tiếp, tên chủ đề,… |
| **UC16** | Thiết lập cấu hình |  |
| **UC17** | Quản lý nhóm | Hiển thị danh sách nhóm |
| **UC18** | Thêm nhóm |  |
| **UC19** | Sửa nhóm | Thay đổi tên, cập nhật thành viên trong nhóm |
| **UC20** | Thêm nhân viên hỗ trợ vào nhóm | Thêm nhân viên hỗ trợ vào nhóm |
| **UC21** | Xóa nhóm |  |
| **UC22** | Quản lý người dùng | Hiển thị danh sách người dùng (nhân viên hỗ trợ, khác hàng, quản trị viên công ty) |
| **UC23** | Xóa người dùng |  |
| **UC24** | Sửa thông tin người dùng |  |
| **UC25** | Quản lý phiếu thắc mắc | Hiển thị danh sách các loại phiếu thắc mắc (đã giải quyết, chưa chỉ định, đang chờ phản hồi từ khách hàng, …) |
| **UC26** | Chọn loại phiếu thắc mắc để quản lý | Hiển thị danh sách phiếu thắc mắc thuộc loại tương ứng mà người dùng chọn xem. |
| **UC27** | Đánh dấu phiếu thắc mắc dạng spam | Xóa phiếu thắc mắc đó đồng thời khóa tài khoản tạo phiếu thắc mắc đó |
| **UC28** | Xóa phiếu thắc mắc. |  |
| **UC29** | Tạo mới trình cắm ngôn ngữ | Nhà phát triển ứng dụng có thể cài đặt để tạo thêm ngôn ngữ trong hệ thống |
| **UC30** | Tạo mới trình cắm giao diện | Nhà phát triển ứng dụng có thể cài đặt để tạo thêm giao diện trong hệ thống |
| **UC31** | Quản lý tài khoản | Chức năng người quản trị viên hệ thống Smart Help Desk quản lý tài khoản đăng ký vào hệ thống. |
| **UC32** | Xem thông tin tài khoản | Quản trị viên hệ thống Smart Help Desk xem thông tin chi tiết của tài khoản đã đăng ký. |
| **UC33** | Sửa thông tin tài khoản | Quản trị viên hệ thống Smart Help Desk sửa thông tin tài khoản. |
| **UC34** | Thêm tài khoản | Quản trị viên hệ thống Smart Help Desk thêm tài khoản. |
| **UC35** | Xóa tài khoản | Quản trị viên hệ thống Smart Help Desk xóa tài khoản. |
| **UC36** | Thay đổi cấp độ tài khoản | Quản trị viên hệ thống Smart Help Desk nâng cấp hoặc giảm cấp tài khoản. |

Bảng 3-6. Bảng danh sách usecase

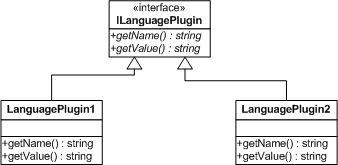
* 1. Cài đặt khung ứng dụng SmartHelpDesk
     1. Đặc tả usecase khung ứng dụng:

|  |  |
| --- | --- |
| **Tóm Tắt:** | Chức năng này giúp cho các lập trình viên khác có thể phát triển tiếp tục phần mềm bằng cách mở rộng chức năng bổ sung thêm ngôn ngữ và giao diện của ứng dụng. |
| **Độ Ưu Tiên:** | Cần thiết |
| **Mức Độ Sử Dụng:** | Thường thường |
| **Người Sử Dụng:** | Lập trình viên khác |
| **Điều Kiện Tiên Quyết:** | Người dùng có mã nguồn của ứng dụng. |
| **Dòng Sự Kiện Chính:** | 1. Người dùng mở file ReadMe.pdf để xem chuẩn của lớp mà mình sẽ thiết kế. 2. Thực hiện các bước theo hướng dẫn trong file ReadMe.pdf 3. Build và chạy lại ứng dụng. 4. Hệ thống cập nhật thêm ngôn ngữ và giao diện. |
| **Dòng Sự Kiện Phụ:** |  |
| **Ghi Chú** |  |
| **Giao Diện** |  |
| **Kiểm Thử** |  |

Bảng 3-7. Đặc tả usecase khung ứng dụng

* + 1. Cài đặt khung giao diện và ngôn ngữ cho ứng dụng:

Cài đặt khung ngôn ngữ:



Hình 3-7. Sơ đồ lớp khung ngôn ngữ (Language Plugin)

*Giải thích:*

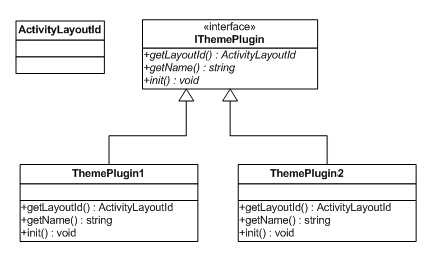
* ILanguagePlugin: mô tả chuẩn (interface) hổ trợ các phương thức để lấy tên và giá trị trình cắm (plugin) muốn bổ sung vào.
* LanguagePlugin (LanguagePlugin1, LanguagePlugin2) cài đặt các phương thức của chuẩn IlanguagePlugin.



Hình 3-8. Sơ đồ tuần tự khung ngôn ngữ (Language Plugin)

Mô tả quy trình thay đổi ngôn ngữ theo yêu cầu của người dùng. Khi người dùng yêu cầu thay đổi ngôn ngữ theo tên loại trình cắm tương ứng, hệ thống sẽ tìm kiếm loại ngôn ngữ hiển thị tương ứng và thay đổi ngôn ngữ của ứng dụng.

Cài đặt khung giao diện:



Hình 3-9. Sơ đồ lớp khung giao diện (Theme Plugin)

*Giải thích:*

* ActivityLayoutId: lớp chứa các thông tin cần thiết về mã tài nguyên (resource id).
* IThemePlugin: mô tả chuẩn (interface) hổ trợ các phương thức cần thiết để xây dựng khung giao diện.
* ThemePlugin (ThemePlugin1, ThemePlugin2) cài đặt các phương thức của chuẩn IThemePlugin.



Hình 3-10. Sơ đồ tuần tự khung giao diện (Theme Plugin)

Mô tả quy trình thay đổi giao diện theo yêu cầu của người dùng. Khi người dùng yêu cầu hiển thị giao diện theo tên loại trình cắm tương ứng, hệ thống sẽ tìm kiếm loại giao diện hiển thị tương ứng và thay đổi giao diện của ứng dụng.

* + 1. Hướng dẫn sử dụng khung ứng dụng SmartHelpDesk:

Tạo trình cắm (plugin) ngôn ngữ

* + - * 1. Đặc tả
* Interface IlanguagePlugin:

**public** **interface** ILanguagePlugin {

String getName();

String getValue();

}

* Phương thức getName(): trả về tên hiển thị của trình cắm.
* Phương thức getValue(): trả về giá trị định danh của ngôn ngữ tương ứng. Ngôn ngữ được định danh bởi 2 ký tự theo chuẩn mã ngôn ngữ ISO 639-1, một số ít khác theo chuẩn mã khu vực ISO 3166-1-alpha-2. Ví dụ: English: en, Việt Nam: vi…. [28]*.*
* Class LanguagePluginManager:

**public** **class** LanguagePluginManager {

**public** ArrayList<ILanguagePlugin> getLanguagePlugins();

**private** **void** init() ;

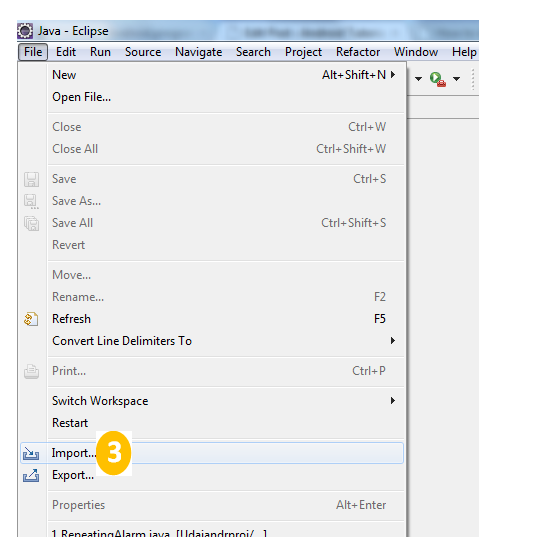
}

* Phương thức getLanguagePlugins(): trả về danh sách các trình cắm có trong ứng dụng.
* Phương thức init(): Khởi tạo danh sách các trình cắm. Nhà phát triển nếu thêm trình cắm ngôn ngữ thì phải thêm dòng mã nguồn vào hàm này. Xem chi tiết trong mã nguồn phần ghi chú.
  + - * 1. Các bước thực hiện
* Chuẩn bị: Trước khi băt đầu, chúng ta cần phải có android-sdk hỗ trợ API 17, công cụ lập trình Eclipse cho android, “source code” của ứng dụng smart help desk.
* Import project Smart Help Desk vào Eclipse:

Bước 1: Chuẩn bị project.

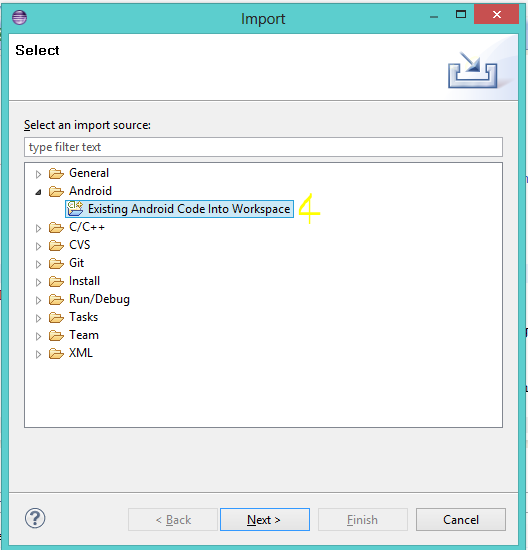
Bước 2: Giải nén project (nếu project đã được nén).

Bước 3: Import project đã được giải nén vào Eclipse: Chọn File >> Import



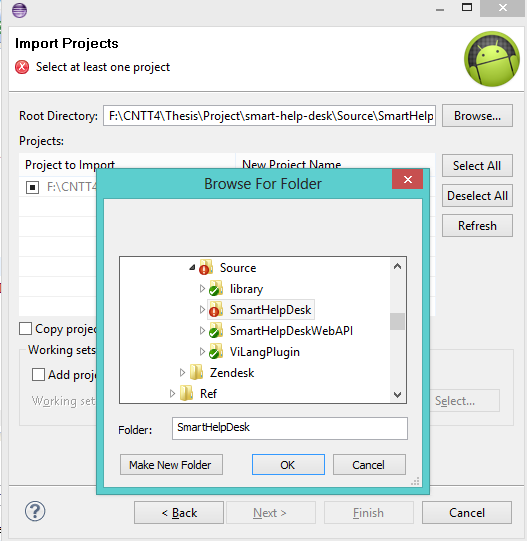
Hình 3-11. Bước 3 của thao tác import project vào eclipse

Bước 4: Import project đã được giải nén vào Eclipse: Select Existing Projects into Workplace and click next.



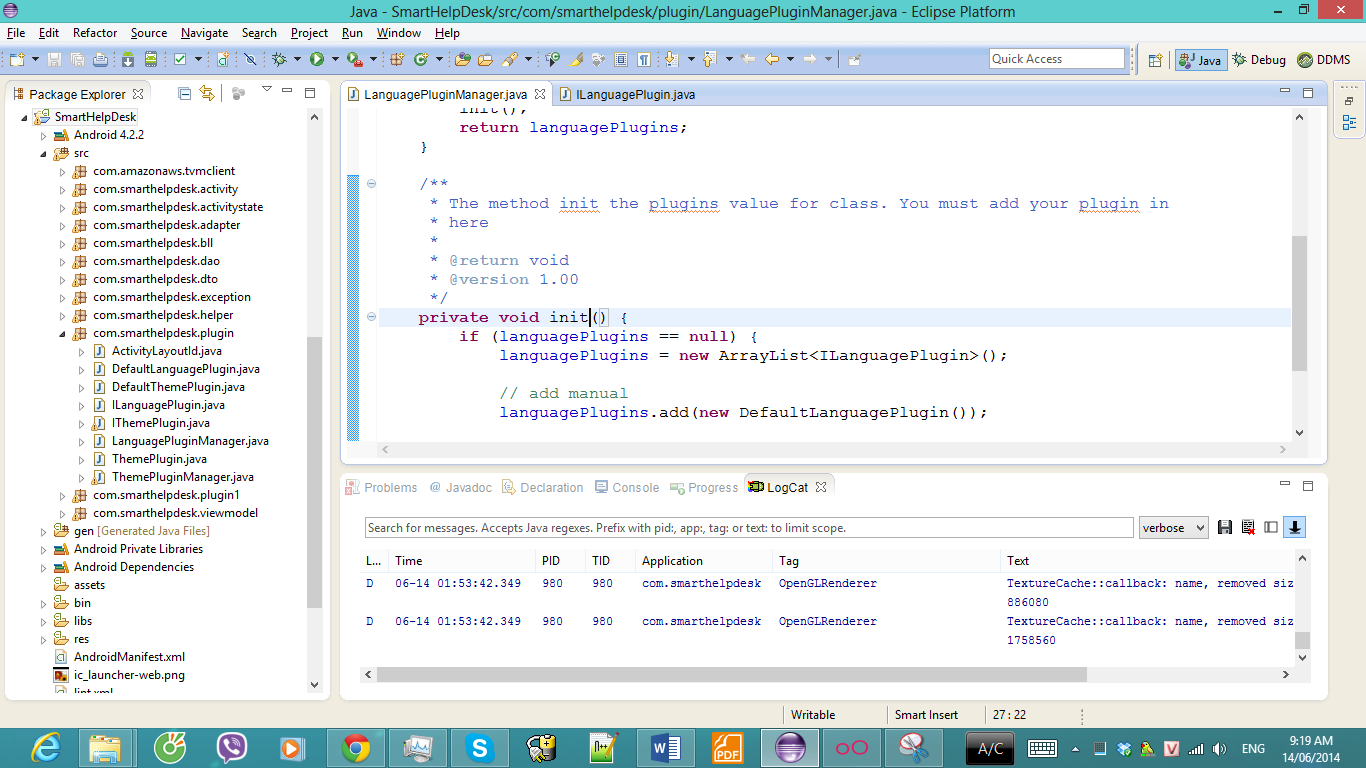
Hình 3-12. Bước 4 của thao tác import project vào eclipse

Bước 5: Import project đã được giải nén vào Eclipse: Chọn Browser để chọn project đã được giải nén, click OK.



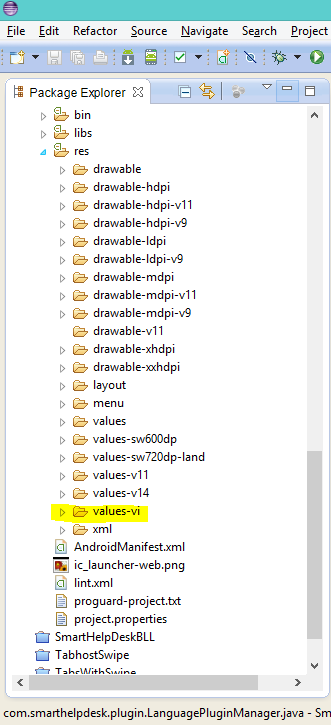
Hình 3-13. Bước 5 của thao tác import project vào eclipse

Bước 6: Click Finish, bạn sẽ thấy project được import hiển thị bên dưới



Hình 3-14. Bước 6 của thao tác import project vào eclipse

* Bắt đầu tạo trình cắm ngôn ngữ
* Tạo thư mục values-xx trong trong thư mục res của project. (xx là mã ký hiệu ngôn ngữ cần tạo, xem lại mục 1.1.1).



*Hình 3-15. Vị trí thư mục values-xx trong project*

* Sao chép tất cả các tập tin string.xml và strings\_xxx.xml trong thư mục values và dán vào thư mục vừa tạo (values-xx).
* Định nghĩa lại tất cả nội dung trong các tập tin “string” vừa copy, thay đổi nội dung phần nội dung sang ngôn ngữ tương ứng.

Ví dụ:

Nội dung tập tin string ban đầu:

<string name=*"hello\_world"*>Hello world!</string>

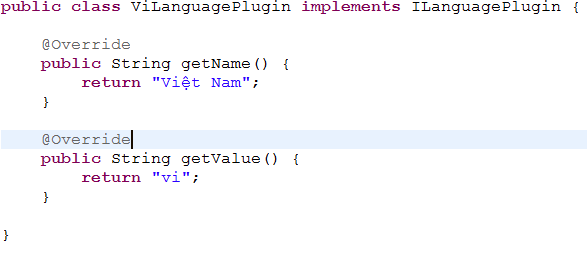
Thay đổi nội dung tương ứng thành ngôn ngữ Việt Nam:

<string name=*"hello\_world"*>Xin chào thế giới!</string>

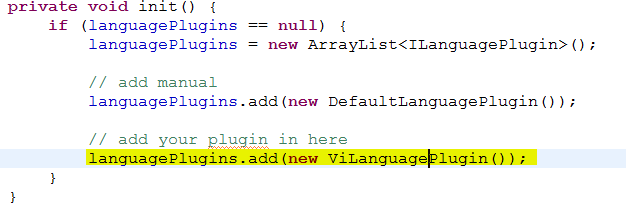
* Tạo một lớp để tạo mới trình cắm ngôn ngữ, lớp này phải được cài đặt từ interface IlanguagePlugin trong package mà bạn có thể tạo mới và cài đặt các hàm tương ứng theo hướng dẫn mục 1.1.1 (chú ý phương thức getName() nên trả về là giá trị duy nhất trong các trình cắm để tránh bị nhầm lẫn khi sử dụng).
* Sau khi tạo class và cài đặt các phương thức, vào tập tin LanguagePluginManager.java trong package com.smarhelpdesk.plugin, rồi vào phương thức init() rồi thêm lớp trình cắm ngôn ngữ bạn vừa tạo.

***Ví dụ:***

* Bạn tạo mới lớp cho ngôn ngữ tiếng việt có tên là ViLanguagePlugin trong package là com.smarthelpesk.plugin1



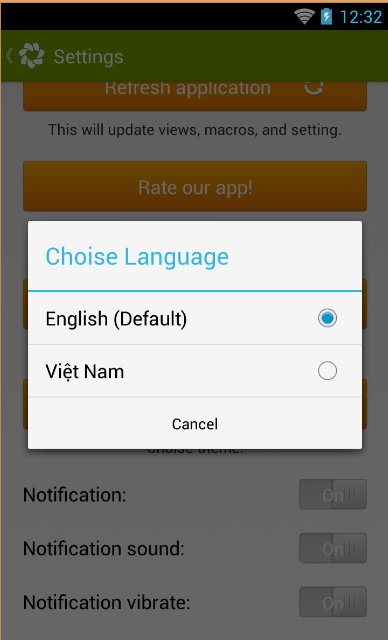
* Sau đó bạn thêm nội dung vào phương thức init trong lớp LanguagePluginManager như sau:



* Buid và sử dụng plugin

Sau thực hiện xong các bước bên trên, tiến hành “build” và chạy ứng dụng.

* Kết quả:
* Khi ứng dụng chạy lên, đăng nhập vào ứng dụng (đăng ký tài khoản nếu chưa có thông qua ứng dụng), vào phần cài đặt.



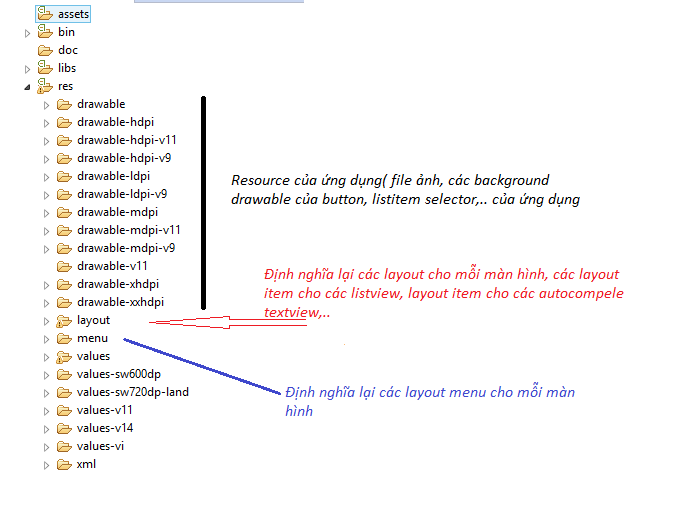
Hình 3-16. Màn hình kết quả

* Sau đó chọn nút Language, sẽ hiển thị danh sách các trình cắm ngôn ngữ đã tạo (Default là của hệ thống). Và chọn vào ngôn ngữ tương ứng thì ứng dụng sẽ thay đổi sang ngôn ngữ bạn vừa chọn.

Tạo trình cắm giao diện

* + - * 1. Tông quan về trình cắm giao diện

Để thay đổi giao diện cho ứng dụng, ta chỉ cần định nghĩa lại các tập tin xml trong thư mục layout của ứng dụng. Khi định nghĩa lại giao diện cho ứng dụng ta chỉ có thể bố trí lại các thành phần trong layout, chỉnh sửa màu sắc, thay đổi hình ảnh. Chúng ta không thể thay đổi các control trong ứng dụng, các quy tắc về đặt tên id cho các control cũng không được chỉnh sửa lại.

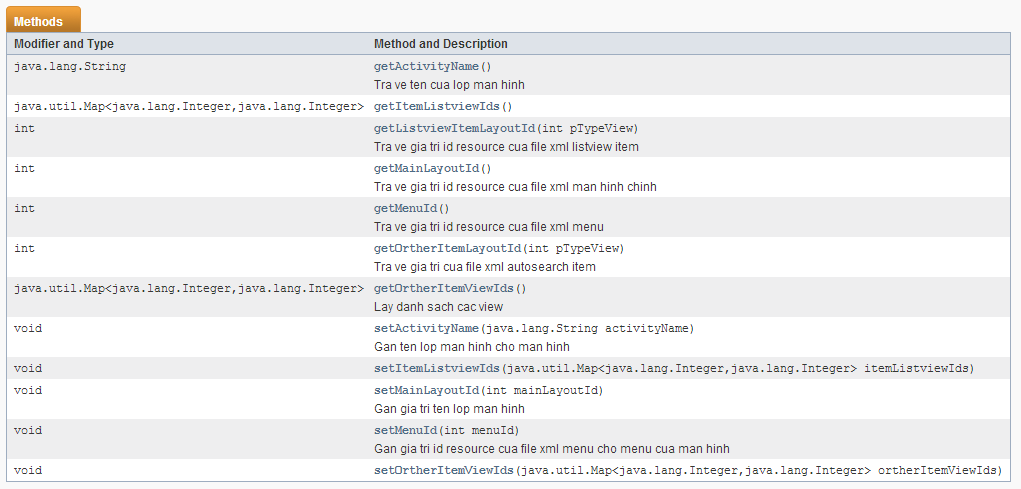


Hình 3-17. Vị trí danh sách các màn hình và đặc tả cho từng màn hình

* + - * 1. Đặc tả

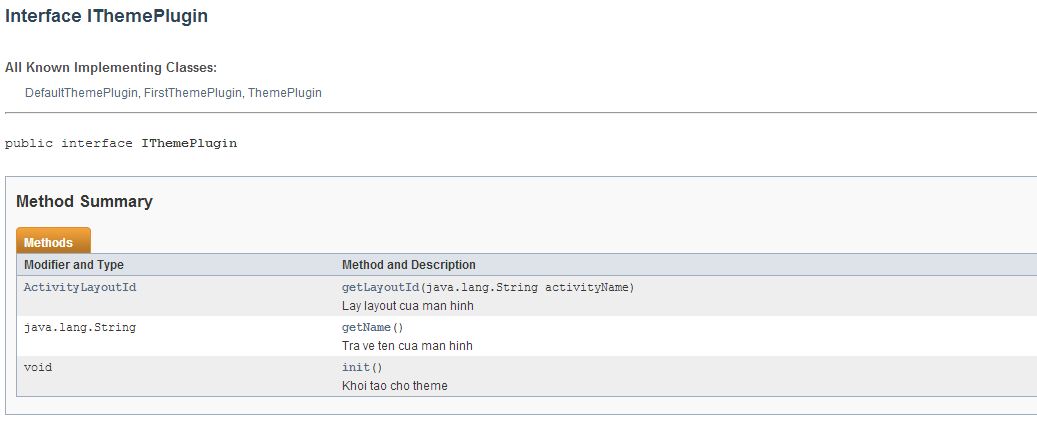
***Lớp ActivityLayoutId:***

* ***“****Mô tả các id thành phần của một giao diện gồm: mainLayoutId – id resource của tập tin giao diện màn hình, menuId – id resource của tập tin giao diện menu, itemListviewIds – danh sách id resource của các tập tin giao diện mô tả cho listview, ortherItemViewIds – danh sách id resource của tập tin giao diện các view khác như autocomplte textview, popup menu”.*
* ***Các phương thức***



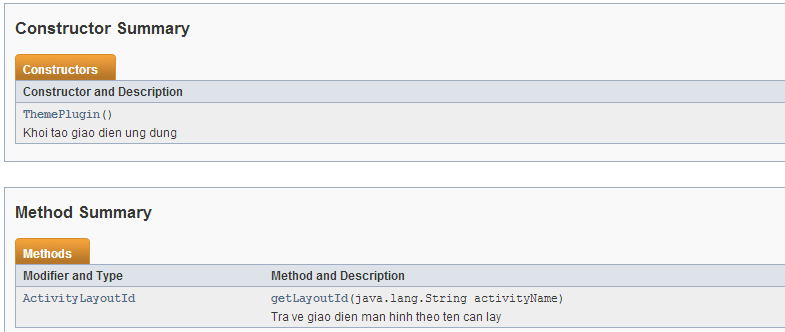
Hình 3-18. Mô tả lớp ActivityLayoutId

***Interface IthemePlugin:***

******

Hình 3-19. Mô tả Interface IThemePlugin

***Class ThemePlugin:***

******

Hình 3-20. Mô tả lớp ThemePlugin

***Danh sách các layout màn hình, và các thành phần khi định nghĩa lại cho mỗi giao diện***

|  |  |  |
| --- | --- | --- |
| STT | Tên màn hình | Mô tả các thành phần cho một giao diên màn hình |
|  | AddToDashboardActivity  (Màn hình thêm phím tắt đến màn hình chính) | **File xml layout main**: activity\_add\_to\_dashboard.xml  **Danh sách file xml layout item listview**:   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_add\_dboard.xml |
|  | BookmarkActivity  (Màn hình đánh dấu) | **File xml layout main**:  activity\_bookmark.xml  **File xml layout menu**:  activity\_views\_activity.xml  **Danh sách file xml layout item listview**:   * ViewItemType.VIEW\_ITEM\_HEADER   list\_item\_views\_detail\_header.xml   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_views\_detail\_content |
|  | DefaultActivity  (Màn hình mặc định) | **File xml layout main**:  acitivity\_default.xml |
|  | DescriptionTicketActivity  (Màn hình nhập mô tả phiếu thắc mắc) | **File xml layout main**: activity\_edit\_description\_ticket.xml  **File xml layout menu**:  activity\_description\_ticket.xml |
|  | EditSubjectTicketActivity  (Màn hình nhập thông tin phiếu thắc mắc) | **File xml layout main**: activity\_edit\_subject\_ticket.xml  **File xml layout menu**:  activity\_description\_ticket.xml |
|  | InformationActivity  (Màn hình thông tin tài khoản của người dùng) | **File xml layout main**: activity\_information.xml |
|  | LoadActivity  (Màn hình tải ứng dụng) | **File xml layout main**: acitivity\_load.xml |
|  | LoginActivity  (Màn hình đăng nhập) | **File xml layout main**: activity\_login.xml |
|  | MainActivity  (Màn hình chính) | **File xml layout main**: activity\_dashboard.xml  **File xml layout menu**: activity\_main\_actions.xml |
|  | ManageGroupActivity  (Màn hình quản lý nhóm) | **File xml layout main**: activity\_manage\_group.xml  **File xml layout menu**: activity\_manage\_groups.xml  **File xml layout item listview**:   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_manage\_groups.xml |
|  | ManageUserActivity  (Màn hình quản lý người dùng) | **File xml layout main**: activity\_manage\_user.xml  **File xml layout menu**: activity\_manage\_users.xml  **File xml layout item listview:**   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_manage\_users.xml |
|  | NewTicketActivity  (Màn hình tạo mới phiếu thắc mắc) | **File xml layout main**: activity\_new\_ticket.xml  **File xml layout menu**: activity\_new\_ticket.xml  **File xml layout item listview**:   * ViewItemType.CONTENT   List\_item\_newticket.xml   * ViewItemType.NEWTICKET\_HEADER   Newticket\_section\_header.xml   * ViewItemType.NEWTICKET\_PROPERTIES\_HEADER   list\_item\_header\_properties.xml |
|  | NewTicketRequesterActivity  (Màn hình nhập người yêu cầu khi tạo phiếu thắc mắc) | **File xml layout main**: activity\_new\_ticket\_requester.xml  **File xml layout item view cho control tìm kiếm người yêu cầu:**  autocomplete\_people\_item.xml |
|  | NewTicketTagsActivity  (Màn hình nhập thẻ đánh dấu khi tạo phiếu thắc mắc) | **File xml layout main**: activity\_new\_ticket\_tag.xml  **File xml layout menu**: activity\_new\_ticket\_search.xml |
|  | RecentActivity  (Màn hình xem các phiếu thắc mắc gần đây) | **File xml layout main**: activity\_recent.xml  **File xml layout menu**: activity\_views\_activity.xml  **Danh sách file xml layout item listview:**   * ViewItemType.VIEW\_ITEM\_HEADER   list\_item\_views\_detail\_header.xml   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_views\_detail\_content |
|  | RegisterActivity  (Màn hình đăng ký) | **File xml layout main**: activity\_register.xml |
|  | SearchResultsActivity  (Màn hình kết quả tìm kiếm) | **File xml layout main**: activity\_search\_results.xml |
|  | SetPassWordActivity  (Màn hình cập nhật mật khẩu) | **File xml layout main**:  SetPassWordActivity.xml |
|  | SettingActivity  (Màn hình cài đặt) | **File xml layout main**:  SettingActivity.xml |
|  | ViewDetailActivity  (Màn hình xem chi tiết phiếu thắc mắc) | **File xml layout main**: activity\_view\_detail.xml  **File xml layout menu**: activity\_views\_activity.xml  **Danh sách file xml layout item listview:**   * ViewItemType.VIEW\_ITEM\_HEADER   list\_item\_views\_detail\_header.xml   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_views\_detail\_content |
|  | ViewsActivity  (Màn hình thể hiện cách xem) | **File xml layout main**: activity\_views.xml  **File xml layout menu**: activity\_views\_activity.xml  **Danh sách file xml layout item listview:**   * ViewItemType.VIEW\_ITEM\_HEADER   list\_item\_views\_detail\_header.xml   * ViewItemType.VIEW\_ITEM\_CONTENT   list\_item\_views\_detail\_content |

Bảng 3-8. Danh sách các layout màn hình

* + - * 1. Các bước thực hiện
* Chuẩn bị: Trước khi băt đầu, chúng ta cần phải có android-sdk hỗ trợ API 17, công cụ lập trình Eclipse cho android, source code của ứng dụng smart help desk.
* Import project SmartHelpDesk vào Eclipse: Xem phần import project SmartHelpDesk ở mục
* Bắt đầu tạo trình cắm giao diện:

Xác đinh những màn hình nào muốn thay đổi giao diện (xem bảng mô tả các thành phần của giao diên), để xác định các thành phần cần thay đổi trong một màn hình.

* Mainlayout: Trong thư mục layout, sao chép và tạo mới các màn hình giao diện muốn thay đổi (nhớ đổi tên các tập tin layout này theo một quy tắc cho dễ nhìn ví dụ: first\_plugin\_activity\_load.xml. Tiến hành thay đổi giao diện cho từng màn hình.)
* Menu layout: Trong thư menu, sao chép và tạo các giao diện cho menu muốn thay đổi (nhớ đổi tên các layout menu này theo một quy tắc cho dễ nhìn ví dụ: first\_plugin\_activity\_views.xml. Tiến hành thay đổi giao diện cho từng menu của từng màn hình.)
* Layout item cho listview: Trong folder layout, sao chép và tạo mới các layout item cho listview muốn thay đổi (nhớ đổi tên các layout này theo một quy tắc cho dễ nhìn ví dụ: first\_plugin\_list\_item\_newticket.xml. Tiến hành thay đổi giao diện cho từng màn hình).

***Ví dụ cụ thể khi thay đổi màn hình Bookmark.***

***Bước 1:*** *Xác định các thành phần của màn hình:*

File xml layout main: activity\_bookmark.xml

File xml layout menu: activity\_views\_activity.xml

Danh sách file xml layout item listview:

• ViewItemType.VIEW\_ITEM\_HEADER

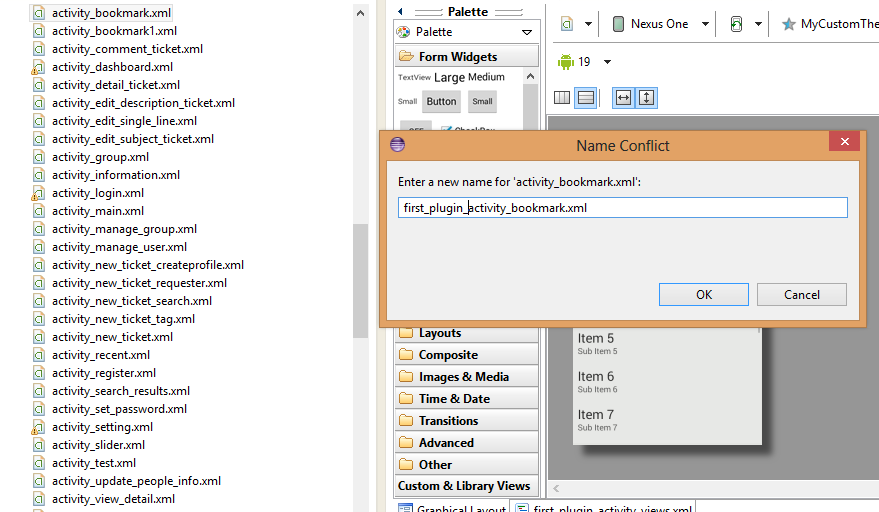
list\_item\_views\_detail\_header.xml

• ViewItemType.VIEW\_ITEM\_CONTENT

list\_item\_views\_detail\_content.xml

***Bước 2:*** *Tạo các file layout cho các thành phần của màn hình:*

* Sao chép tập tin activity\_bookmark.xml cần thay đổi giao diện và dán vào thư mục layout, đổi tên thành first\_plugin\_activity\_bookmark.xml

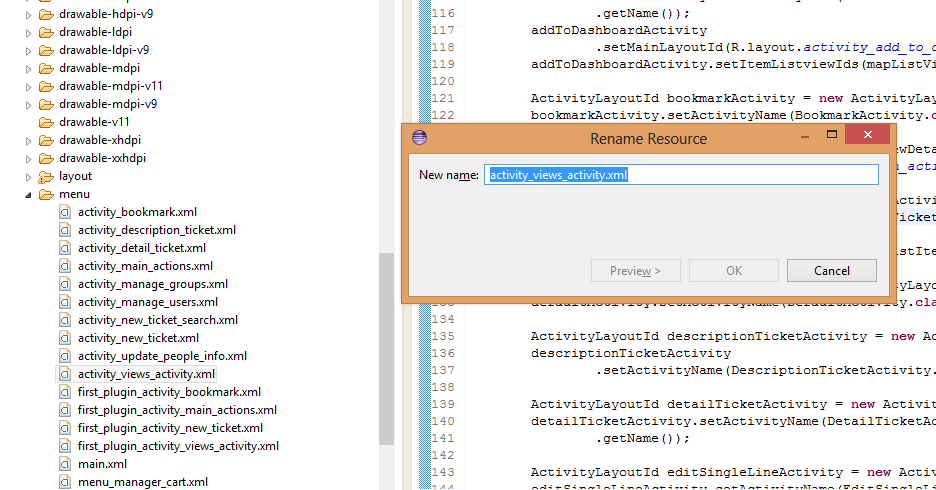


Hình 3-21. Mô tả cách thay đổi giao diện chính của màn hình Bookmark

Tiến hành thay đổi lại các giao diện cho first\_plugin\_activity\_bookmark.xml như hình ảnh, màu sắc, cách bố trí.

* Nếu muốn thay đổi menu cho màn hình.

Sao chép tập tin activity\_views\_activity.xml trong thư mục menu và dán vào thư mục này, đổi tên thành first\_plugin\_activity\_views\_activity.xml

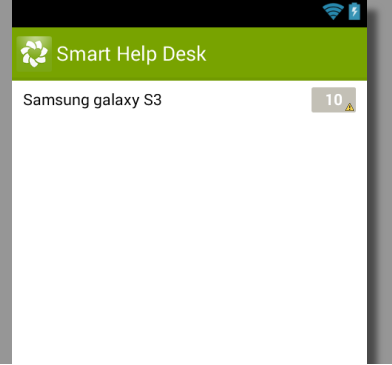
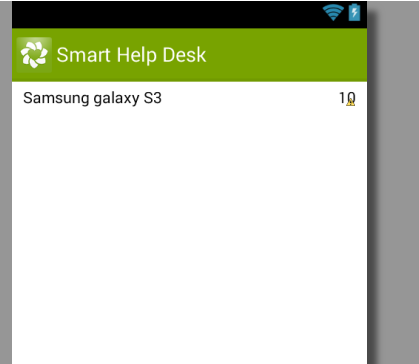


Hình 3-22. Mô tả thao tác thay đổi menu màn hình Views

Tiến hành thay đổi lại các giao diện cho first\_plugin\_activity\_views\_activity.xml trong thư mục menu như hình ảnh, tiêu đề, cách xuất hiện…

Sao chép tập tin list\_item\_views\_detail\_header.xml và tập tin list\_item\_views\_detail\_content.xml trong thư mục layout và dán vào thư mục này, đổi tên thanh first\_plugin\_list\_items\_views\_detail\_header.xml và first\_plugin\_list\_item\_views\_detail\_content.xml

Tiến hành thay đổi lại giao diện, các bố trí các control trong những item view layou này.



Hình 3-23. list\_item\_views\_detail\_header.xml first\_plugin\_list\_item\_views\_detail\_header.xml

Tương tự tiến hành thay đổi cho first\_plugin\_list\_item\_views\_detail\_header.xml

***Bước 3:*** Tiến hành tạo lớp và đưa vào hệ thống.

* Tạo class FirstThemePlugin extends ThemePlugin implements IthemePlugin. Ở đây chúng ta phải override lại 2 phương thức là:

@Override

**public** String getName() {

**return** "First theme";

}

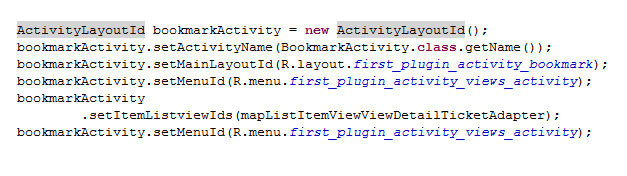
Trả về tên của plugin Theme mới tạo

@Override

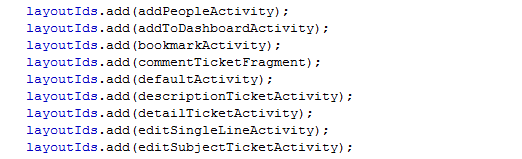
public void init() {

// Thêm những màn hình muốn chỉnh sửa vào đây

// Sau khi xác định các thành phần của một màn hình tiến hành tạo một lớp ActivityLayoutId bookmarkActivity = new ActivityLayoutId(); và đưa các thành phần muốn chỉnh sửa vào giao diện. Lưu ý nếu muốn giữ nguyên thành phần nào (dùng lại thành phần mặc định) thì ta sẽ không gán giá trị cho thành phần đó.

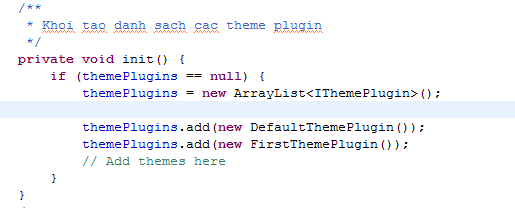


// Đưa những màn hình muốn thay đổi giao diện vào hệ thống.



}

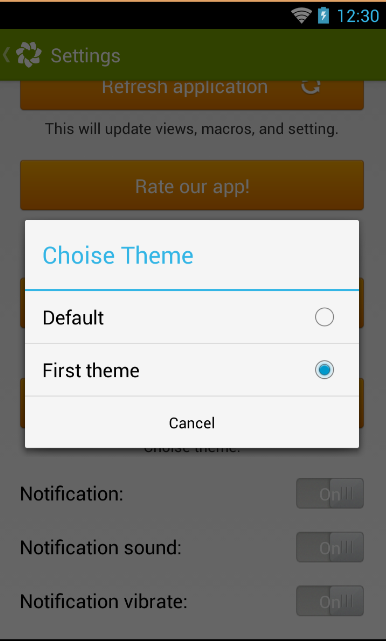
* Tiến hành đưa FirstThemePlugin vào hệ thống, trong class ThemePluginManager của com.smarthelpdesk.plugin. Đưa thêm FirstThemePlugin vừa tạo vào.



* Build và sử dụng trình cắm giao diện

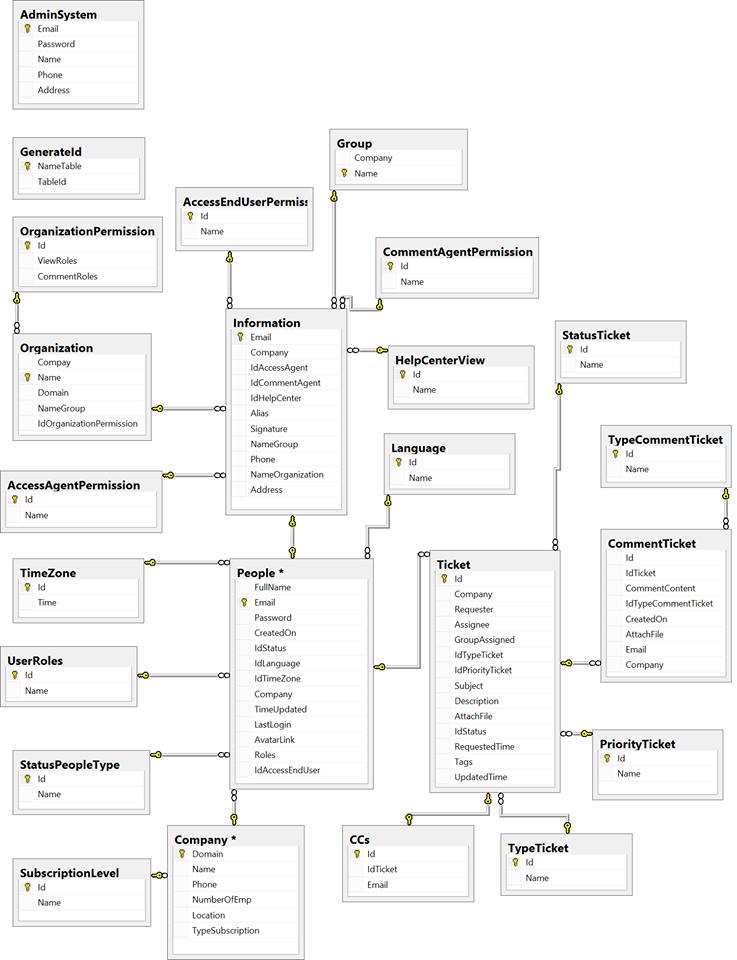
Sau thực hiện xong các bước bên trên, tiến hành build và run ứng dụng.

* Kết quả
* Khi ứng dụng chạy lên, đăng nhập vào ứng dụng (đăng ký tài khoản nếu chưa có thông qua ứng dụng), vào phần setting.



Hình 3-24. Màn hình kết quả

* Sau đó chọn vào nút Theme, sẽ hiển thị danh sách các trình cắm giao diện. Và chọn vào giao diện tương ưng thì ứng dụng sẽ thay đổi sang giao diện bạn vừa chọn.
  1. Thiết kế dữ liệu Amazon DynamoDB
     1. Sơ đồ dữ liệu:



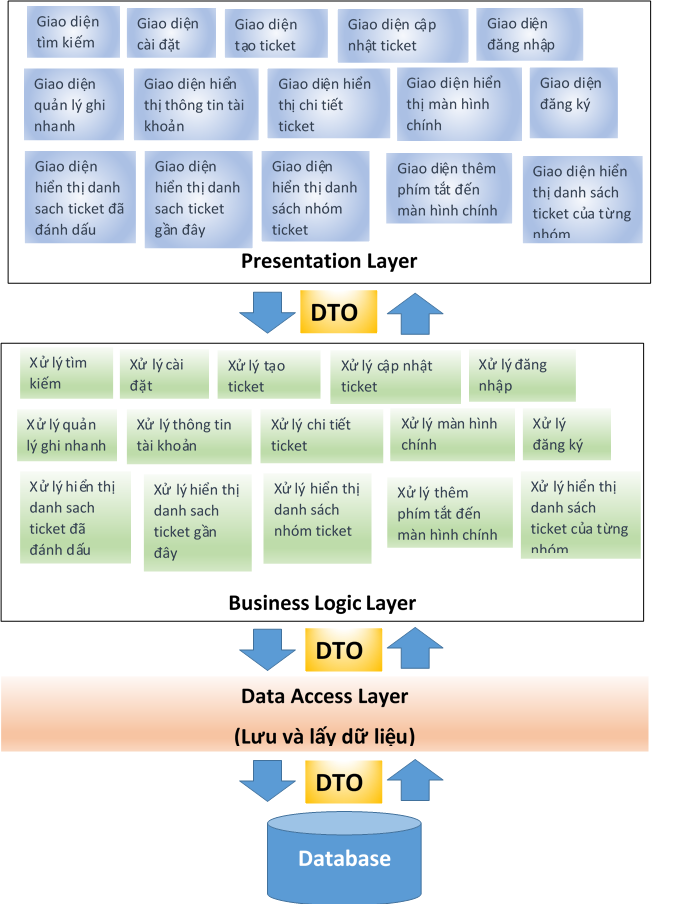
Hình 3-25. Sơ đồ cơ sở dữ liệu

* + 1. Danh sách các bảng dữ liệu:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Bảng** | **Ý nghĩa** |
| 1 | AccessAgentPermission | Lưu trữ những quyền truy cập của người hỗ trợ khách hàng. |
| 2 | AccessEndUserPermission | Lưu trữ những quyền truy cập của khách hàng. |
| 3 | AdminSystem | Lưu trữ thông tin của những quản trị viên của hệ thống Smart Help Desk. |
| 4 | CCs | Lưu trữ thông tin khi một phiếu thắc mắc được chuyển tiếp bởi 1 chủ đề nào đó. |
| 5 | CommentAgentPermission | Lưu trữ những quyền bình luận của người hỗ trợ khách hàng. |
| 6 | CommentTicket | Lưu trữ những bình luận trong phiếu thắc mắc. |
| 7 | Company | Lưu trữ thông tin của những công ty đăng ký vào hệ thống. |
| 8 | Group | Lưu trữ thông tin những nhóm hỗ trợ khách hàng trong công ty. |
| 9 | HelpCenterView | Lưu trữ những quyền mà người hỗ trợ khách hàng có thể thao tác khi đăng nhập vào hệ thống. |
| 10 | Language | Lưu trữ những ngôn ngữ |
| 11 | Organization | Lưu trữ những tổ chức của công ty |
| 12 | OrganizationPermission | Lưu trữ những quyền truy cập của những người dùng trong tổ chức. |
| 13 | People | Lưu trữ thông tin của người dùng. |
| 14 | PriorityTicket | Lưu trữ những độ ưu tiên của 1 chủ đề |
| 15 | UserRoles | Lưu trữ những cấp độ của người dùng trong công ty. |
| 16 | StatusPeopleType | Lưu trữ những trạng thái về tài khoản của người dùng trong công ty. |
| 17 | StatusTicket | Lưu trữ những trạng thái của phiếu thắc mắc. |
| 18 | SubscriptionLevel | Lưu trữ những cấp độ gói cước sử dụng của công ty. |
| 19 | Ticket | Lưu trữ những yêu cầu hỏi đáp của khách hàng gửi đến cho bộ phận hỗ trợ. |
| 20 | TimeZone | Lưu trữ những múi giờ. |
| 21 | TypeCommentTicket | Lưu trữ những loại của bình luận trong chủ đề. |
| 22 | TypeTicket | Lưu trữ những loại của phiếu thắc mắc. |
| 23 | Information | Lưu thông tin riêng cho người hỗ trợ và quản trị viên công ty. |
| 24 | GeneratedId | Lưu mã phát sinh kế tiếp cho một số đối tượng |

Bảng 3-9. Danh sách bảng cơ sở dữ liệu

* 1. Kiến trúc hệ thống.
     1. Kiến trúc theo mô hình 3 layer:



Hình 3-26. Kiến trúc 3 layer

* + 1. Cây phân rã hệ thống:



Hình 3-27. Cây phân rã hệ thống

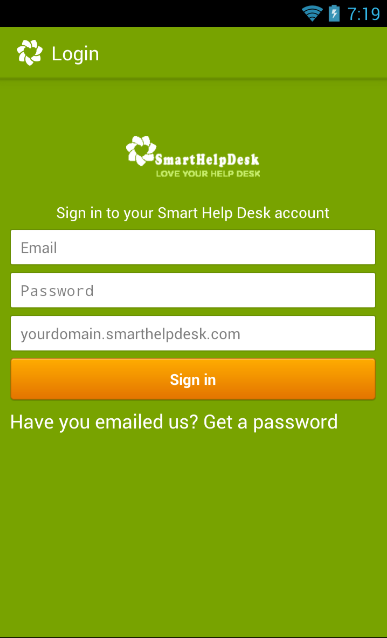
* 1. Giao diện.
     1. Danh sách các màn hình chính:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên màn hình** | **Ý nghĩa/Ghi chú** |
|  | Màn hình đăng nhập | Đăng nhập vào hệ thống |
|  | Màn hình đăng ký | Đăng kí tài khoản cho một công ty. |
|  | Màn hình chính | Hiển thị thông tin các chức năng chính của ứng dụng, từ màn hình này có người dùng có thể chọn nhanh chóng các chức năng của ứng dụng. |
|  | Màn hình tạo mới phiếu thắc mắc | Tạo mới một phiếu thắc mắc. |
|  | Màn hình danh sách phiếu thắc mắc theo nhóm | Hiển thị danh sách phiếu thắc mắc |
|  | Màn hình nội dung phiếu thắc mắc | Hiển thị nội dung phiếu thắc mắc bao gồm: nội dung trao đổi, thắc mắc của khách hàng, thời gian, tác giả... |
|  | Màn hình nhập trả lời phiếu thắc mắc | Người dùng nhập thông tin trả lời, có thể thể đính kèm file. |

Bảng 3-10. Danh sách các màn hình

* + 1. Một số giao diện chính:

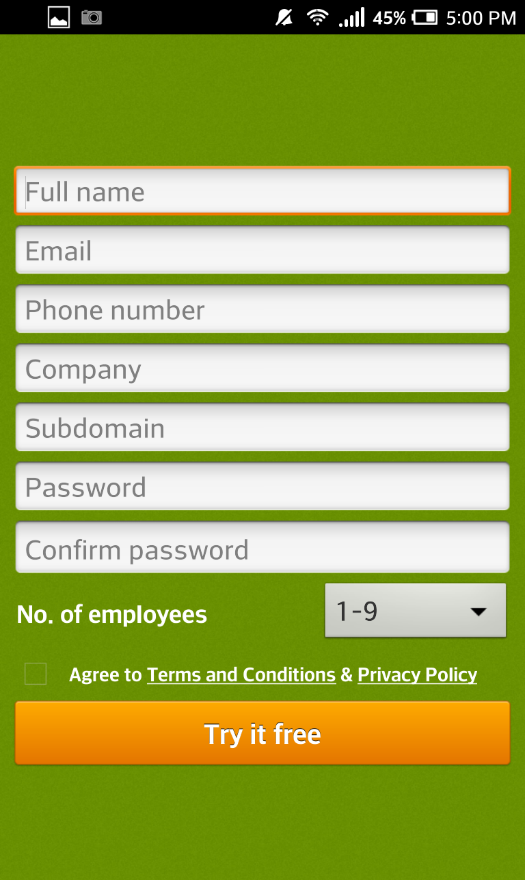
Màn hình đăng nhập:



Hình 3-28. Màn hình đăng nhập

Người dùng nhập tên email và mật khẩu và tên miền của hệ của công ty mình sau đó chọn đăng nhập để đăng nhập vào hệ thống. Hệ thống sẽ kiểm tra thông tin và hiển thị chi tiết lỗi.

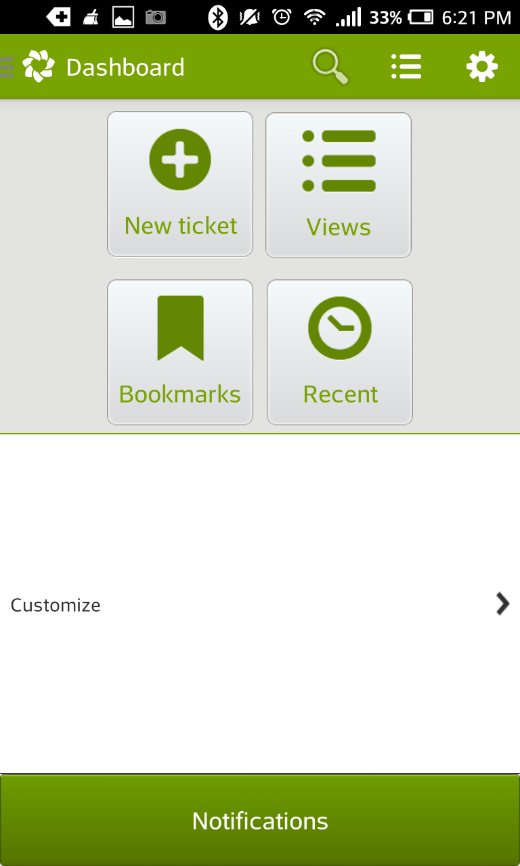
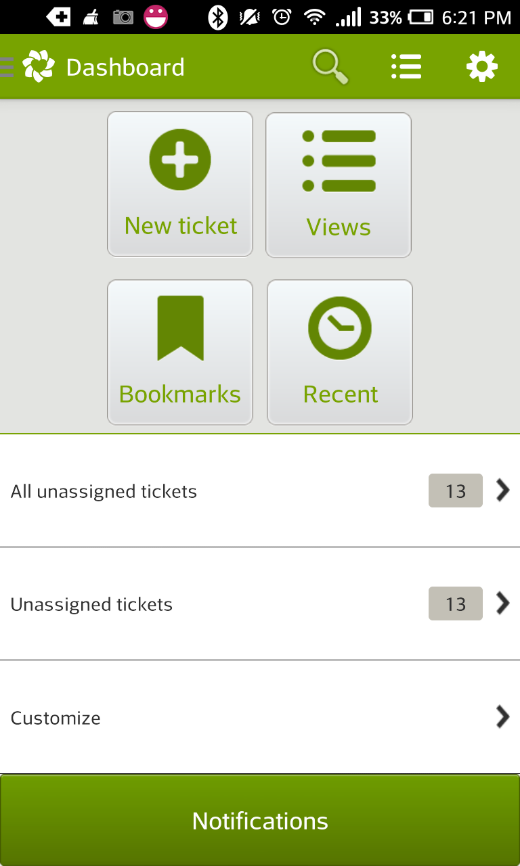
Màn hình đăng ký:

**

Hình 3-29. Màn hình đăng ký

Người dùng nhập họ tên, email, điện thoại, tên công ty, tên miền con, mật khẩu, xác nhận mật khẩu, chọn số lượng nhân viên, đồng ý với điều khoản và các chính sách. Người dùng nhấn “Try it free” để đăng kí tài khoản SmartHelpDesk. Phần mềm sẽ thông báo lỗi nếu người dùng nhập sai các thông tin đăng kí.

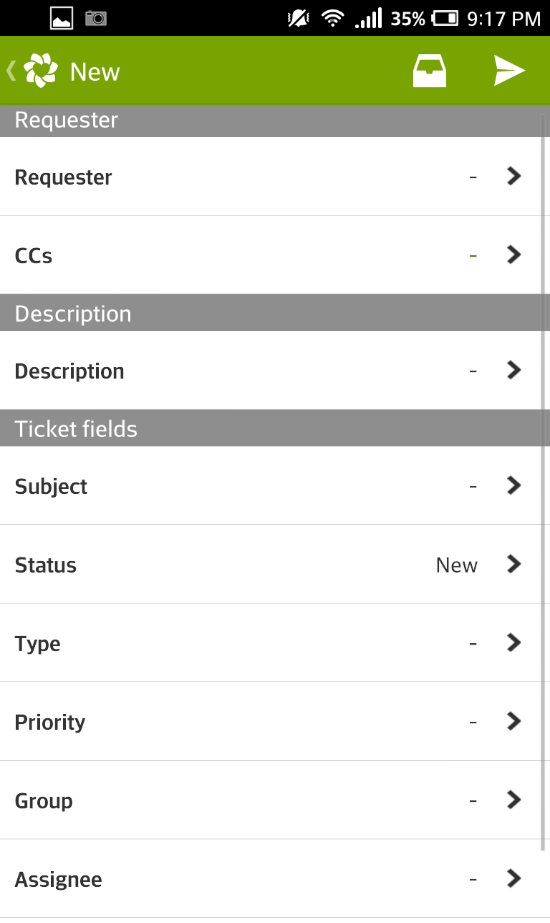
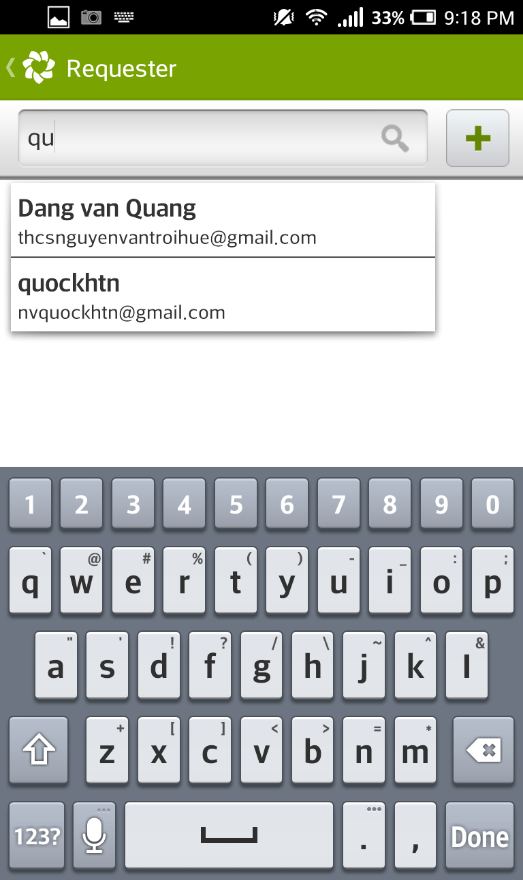
Màn hình chính:

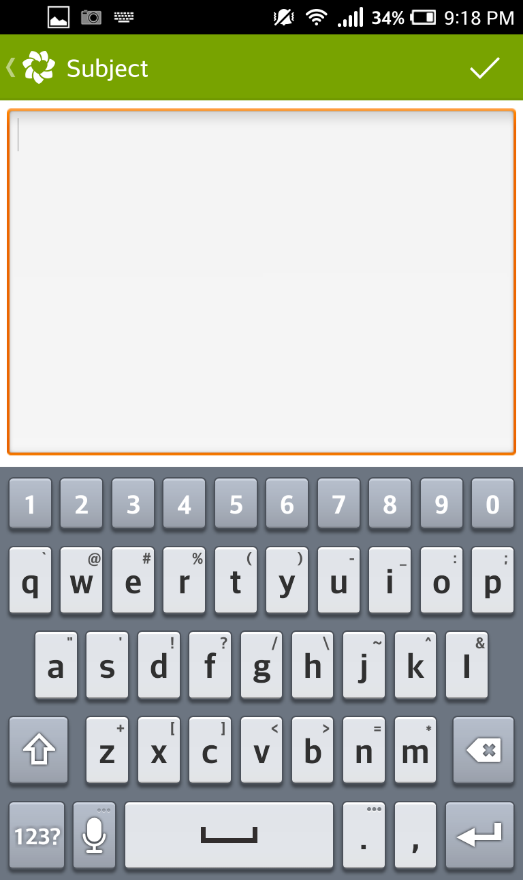
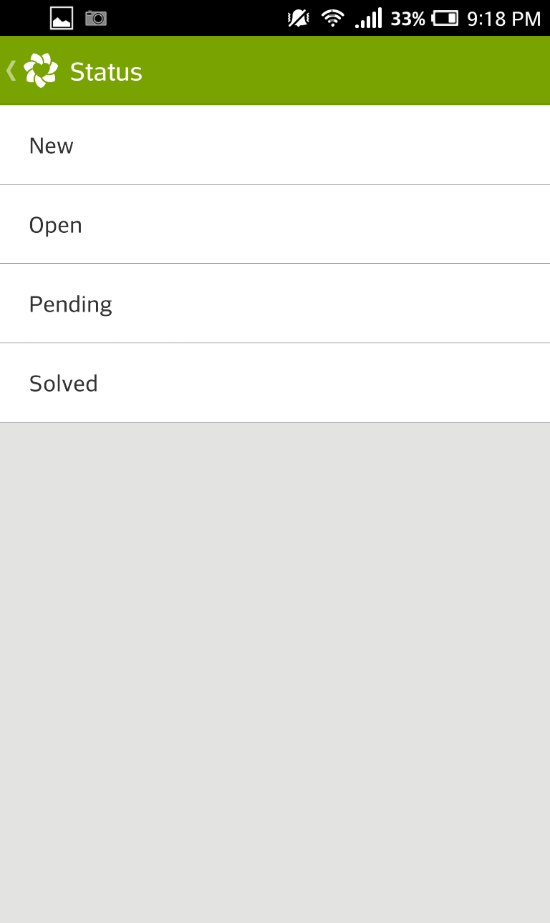
* *

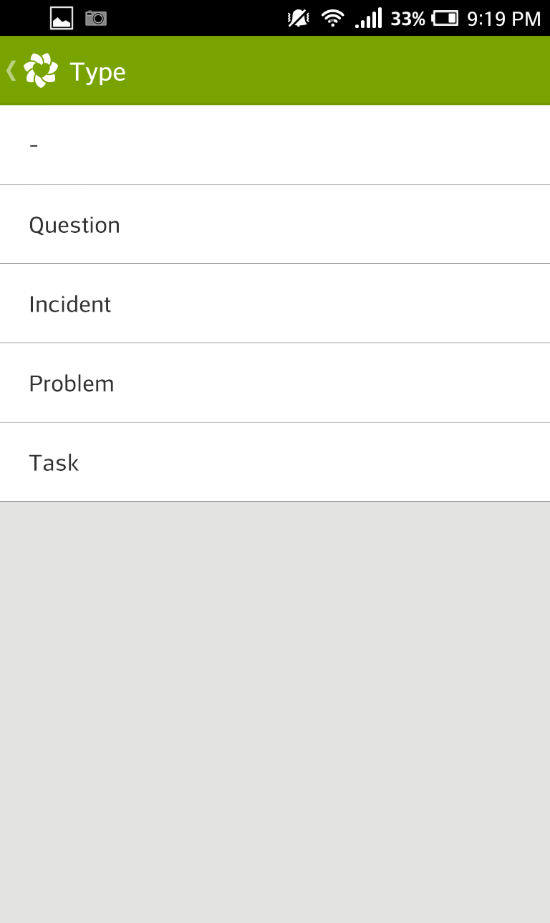
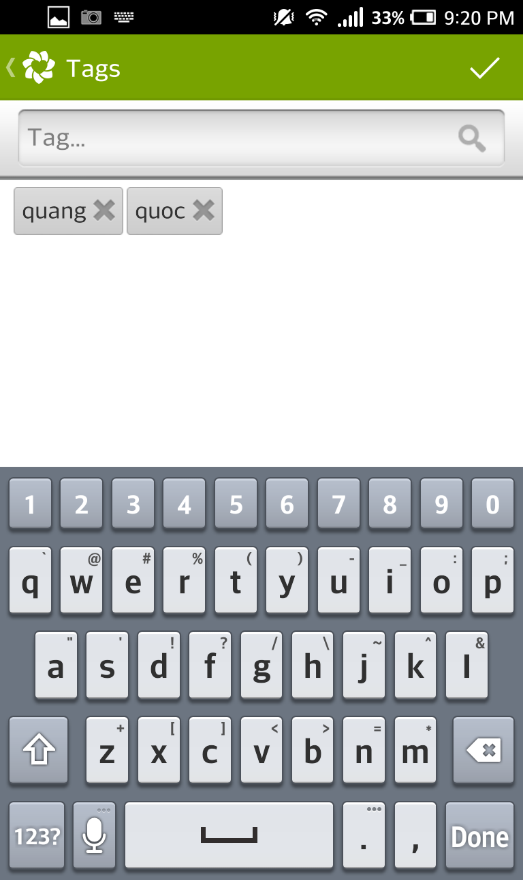
Hình 3-30. Màn hình chính

Sau khi đăng nhập thành công, người dùng sẽ vào màn hình chính, ở màn hình chính sẽ hiên các chức năng, thanh menu, các phím dùng nhanh, tùy theo phân quyền mà màn hình sẽ hiện các chức năng tương ứng.

Màn hình tạo mới phiếu thắc mắc:

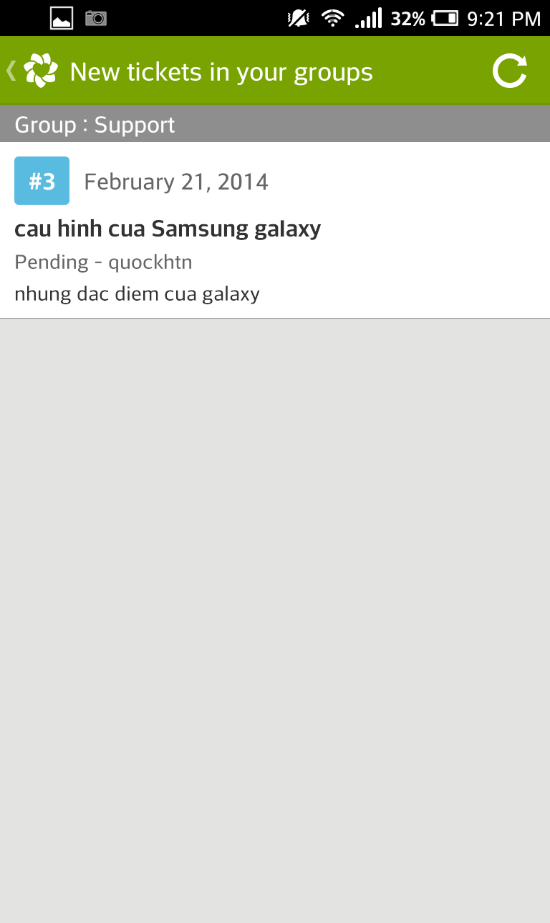
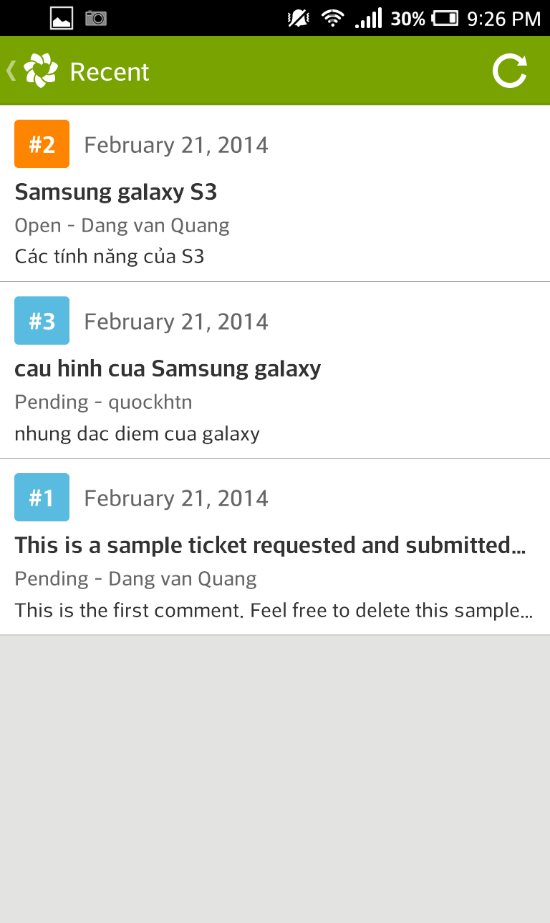
 

Hình 3-31. Màn hình tạo mới phiếu thắc mắc.

Màn hình hiện thị các thông tin cho người dùng có thể tạo mới phiếu thắc mắc: Người gửi yêu cầu, chuyển tiếp tới ai, nội dung, chủ đề phiếu thắc mắc, trạng thái, loại, độ ưu tiên, thuộc nhóm nào, những từ được đánh dấu kèm theo. Người dùng có thể sử dụng những từ viết nhanh hỗ trợ nhập nội dung phiếu thắc mắc bằng chọn và áp dụng từ viết nhanh. Để lưu phiếu thắc mắc lại người dùng chọn vào gửi.

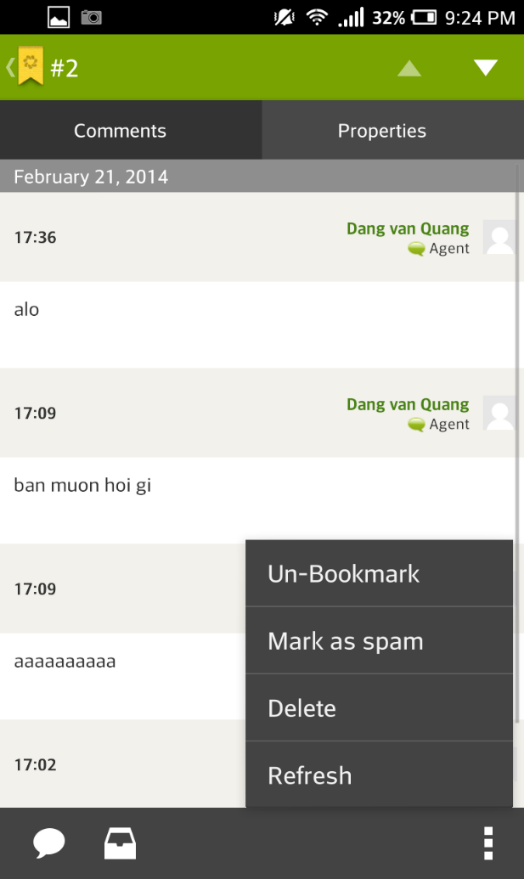
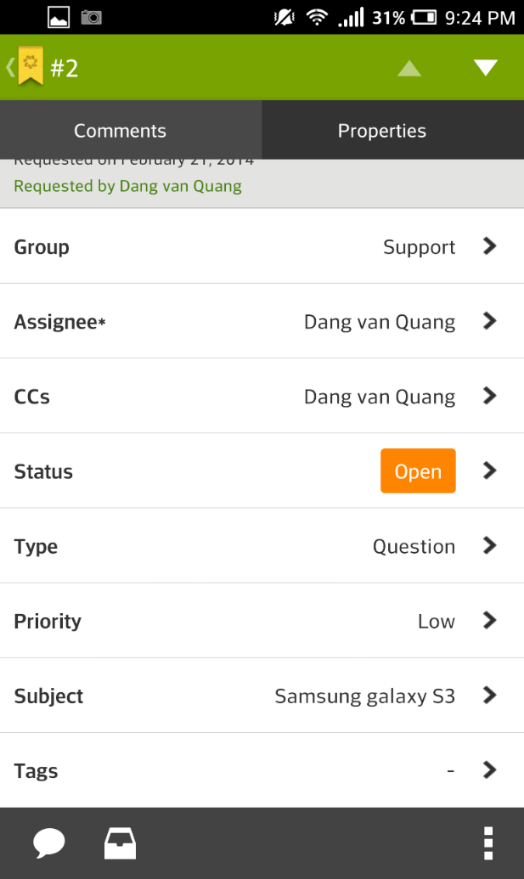
Màn hình danh sách phiếu thắc mắc theo nhóm:

Hình 3-32. Màn hình danh sách phiếu thắc mắc.

Hiển thị danh sách các phiếu thắc mắc trong nhóm được chọn bao gồm những thông tin tóm tắt: số thứ tự phiếu thắc mắc, thời gian, chủ đề, nội dung, tình trạng. Có nút cập nhật danh sách phiếu thắc mắc. Khi người người click vào phiếu thắc mắc nào thì sẽ chuyển màn hình sang chi tiết phiếu thắc mắc đó.

Màn hình nội dung phiếu thắc mắc:

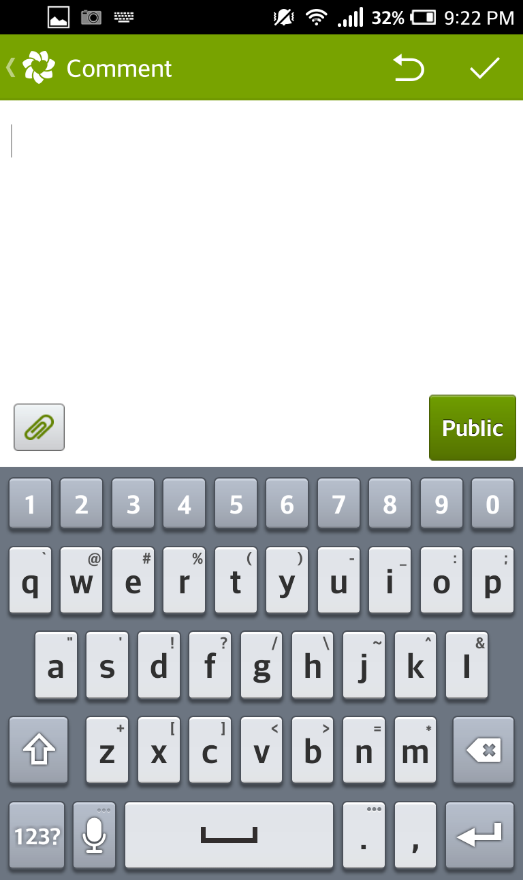
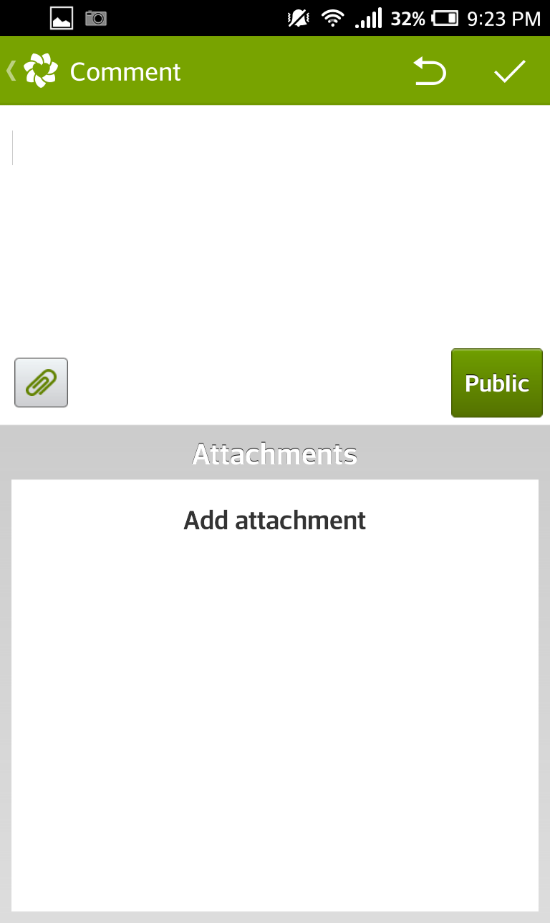
 

Hình 3-33. Màn hình nội dung phiếu thắc mắc.

Màn hình hiển thị nội dung chi tiết phiếu thắc mắc, có thể đánh dấu – gỡ đánh dấu, xóa, cập nhật:

1. Bên phần bình luận: Thời gian, tác giả, nội dung trả lời, trả lời.
2. Bên phần thuộc tính: Hiển thị thông tin khác của phiếu thắc mắc nhóm, người chỉ địn, những người chuyển tiếp… có thể cập nhật những thông tin: nhân viên hộ trợ, những người chuyển tiếp, nhóm, …

Màn hình nhập trả lời phiếu thắc mắc:

Hình 3-34. Màn hình nhập trả lời phiếu thắc mắc.

Màn hình này giúp người dùng nhập thông tin nội dung trả lời, có thể trở về, đính kèm ảnh, có thể chọn chế độ cho mọi người cùng xem (public) hoặc riêng tư (private) phiếu thắc mắc và nhấn dấu tic để gửi trả lời.

CHƯƠNG 4. TỔNG KẾT ĐÁNH GIÁ

1. 1. Kiến thức đạt được

* Hiểu rõ về công nghệ điện toán đám mây, có cái nhìn mô hình tổng quan và các đặc trưng của điện toán đám mây.
* Hiểu được sự khác biệt giữa ứng dụng điện toán đám mây với ứng dụng truyền thống. Nắm được một số cách chuyển đổi ứng dụng doanh nghiệp từ một môi trường vật lý cố định đến một môi trường đám mây ảo hóa.
* Những khuyết điểm của điện toán đám mây còn gặp phải và cần khắc phục trong tương lai.
* Nắm được dịch vụ Amazon Web Service: kiến trúc, dịch vụ, nền tảng công nghệ và những thế mạnh tận dụng được. Xây dựng được một ứng dụng android sử dụng DynamoDB (NoSQL), tương tác với cơ sở dữ liệu thông qua AWS SDK.
* Vận dụng được kiến thức lập trình android, mô hình “3 layer” vào lập trình ứng dụng; vận dụng những kiến thức tìm hiểu được về dịch vụ lưu trữ trên đám mây của Amazon Web Service để lựa chọn dịch vụ lưu trữ thích hợp và linh hoạt trong xử lý.
* Kinh nghiệm làm việc nhóm, lên kế hoạch, giải quyết khó khăn, khả năng nghiên cứu học hỏi công nghệ mới, khả năng diễn đạt những gì mình tìm hiểu thành luận văn, học được kỹ năng viết khóa luận tốt nghiệp.
  1. Kết quả ứng dụng
     1. Môi trường phát triển ứng dụng:
* Hệ điều hành: Android 3.0 trở lên.
* Hệ quản trị cơ sở dữ liệu: DynamoDB.
* Công cụ phân tích thiết kế: Rational Rose 2007, Microsoft SQL.
* Công cụ xây dựng ứng dụng: Eclipse IDE for Android.
* Các thư viện đã dùng: Android SDK, AWS SDK.
* Đưa ứng dụng lên Google Play Store
  + 1. Những chức năng đã hoàn thành:

Dưới đây là danh sách các chức năng chúng em đã xây dựng khung cho ứng dụng SmartHelpDesk trên nên tảng di động.

* Đăng nhập sau đó phân loại người dùng với các chức năng tùy theo từng vai trò của mỗi người.
* Đăng kí tài khoản mới cho công ty.
* Tìm kiếm phiếu thắc mắc.
* Xem thông tin chi tiết phiếu thắc mắc.
* Trả lời phiếu thắc mắc.
* Đánh dấu phiếu thắc mắc.
* Bỏ đánh dấu phiếu thắc mắc.
* Tạo mới phiếu thắc mắc.
* Xem phiếu thắc mắc gần đây.
* Xem phiếu thắc mắc đã đánh dấu.
* Tạo mới khách hàng.
* Sửa phiếu thắc mắc.
* Thiết lập cấu hình.
* Quản lý phiếu thắc mắc tùy theo vai trò tùy theo các vai trò của người dùng, sẽ chỉnh sửa được các thông tin đúng với phạm vi mà họ quản lý.
* Quản lý nhóm thêm nhóm, sửa nhóm, xóa nhóm, thêm nhân viên vao nhóm
* Quản lý người dùng thêm người dùng, xóa, chỉnh sửa thông tin người dùng, tùy theo các vai trò của người dùng, sẽ chỉnh sửa được các thông tin đúng với phạm vi mà họ quản lý.
  + 1. Những chức năng chưa hoàn thành:
* Chức năng quản lý các công ty dành cho quản trị viên của hệ thống SmartHelpDesk.
* Chức năng quản lý tổ chức, liên kết tổ chức đến nhóm.
* Chức năng tự nhận thông báo khi có phiếu thắc mắc mới, khi nhận được phản hồi từ khách hàng hoặc nhân viên hổ trợ.
* Chức năng đính kèm tập tin cho phiếu thắc mắc.
* Chức năng thay đổi ảnh đại diện của người dùng.
  + 1. Đánh giá:

Ứng dụng mà chúng em xây dựng cơ bản đã đạt được những yêu cầu đề ra khi xây dựng một khung ứng dụng trên nền tảng đám mây. Và xây dựng được hầu hết các chức năng mà nhóm đề ra (80%).

Việc sử dụng kỹ thuật “Khung hộp xám” trong thiết kế một khung ứng dụng giúp việc mở rộng ứng dụng trở nên dễ dàng và linh hoạt hơn. Để bổ sung thêm các ngôn ngữ giao diện mới cho ứng dụng nhà phát triển chỉ cần cài đặt lại một số hàm để tùy chỉnh lại theo yêu cầu của mình và không cần phải hiểu biết mã nguồn của ứng dụng đã cài đặt như thế nào, mà nhà phát triển chỉ cần hiểu cần phải làm gì để xây dựng ứng dụng cho riêng mình dựa trên nền tảng khung ứng dụng có sẵn mà nhóm cung cấp. Tuy nhiên, điểm hạn chế lớn lớn nhất là các nhà phát triển ứng dụng vẫn có thể chỉnh sửa trực tiếp lên mã nguồn gốc của khung ứng dụng, dẫn đến có thể gây ra một số lỗi không mong muốn. Vì thế trong tương lai chúng em sẽ cải thiện khuyết điểm này của khung ứng dụng.

* 1. Những thuận lợi và khó khăn khi thực hiện đề tài này
     1. Thuận lợi:
* Đã có kinh nghiệm về lập trình android.
* Khả năng tìm hiểu công nghệ mới nhanh.
* Có kiến thức kỹ thuật, công nghệ tốt được học ở trường.
  + 1. Khó khăn:
* Việc đăng ký tài khoản Amazon Web Service khá khó khăn, tài khoản sử dụng nhiều tài nguyên vượt giới hạn miễn phí dẫn đến tài khoản bị khóa và phải tạo nhiêu tài khoản để thực hiện đề tài. Và thời gian đăng ký thành công tài khoản amazon khá lâu ảnh hưởng đến tiến độ thực hiện luận văn.
* Chưa có nhiều kinh nghiệm về điện toán đám mây.
* Không biết cách thức, quy trình phản hồi từ khách hàng và doanh nghiệp trong thực tế.
* Kiến thức về nền tảng đám mây của Amazon khá rộng lớn nhưng thời gian tìm hiểu và viết ứng dụng không nhiều.
  1. Phương pháp nghiên cứu và phát triển đề tài trong tương lai
     1. Công nghệ:
* Nghiên cứu và sử dụng những dịch vụ điện toán đám mây của Amazon, nhằm phục vụ cho việc mở rộng ứng dụng ở tầm quy mô lớn hơn.
* Nghiên cứu chi tiết hơn về các nhà cung cấp dịch vụ lớn khác. Như Microsoft, Google Engine... để nâng cao kinh nghiệm sử dụng dịch vụ điện toán đám mây.
* Nghiên cứu thêm các công nghệ DynamoDB hỗ trợ nhằm tăng tốc độ truy xuất dữ liệu.
  + 1. Ứng dụng:

Qua những hạn chế khi phát triển ứng dụng, nhóm có những ý tưởng nhằm phát triển ứng dụng trong tương lai:

* Tích hợp thời gian thực cho ứng dụng.
* Hoàn thiện hết các chức năng của ứng dụng.
  1. Lời kết

Vì khoảng thời gian tương đối ngắn, khả năng còn hạn hẹp, chúng em chỉ mới nghiên cứu được một phần nhỏ về nền tảng Điện toán đám mây nói chung, Amazon Web Service nói riêng. Nó là một lĩnh vực không còn quá mới, xong gần đây được giới công nghệ đánh giá khá cao, và đầy hứa hẹn trong tương lại không xa. Với những kiến thức và kĩ năng được rèn luyện trong suốt thời gian học tại trường và đặc biệt hơn là khoảng thời gian ngắn ngủi của khóa luận đã giúp chúng em học hỏi được nhiều điều. Và hy vọng nó là sẽ giúp chúng em có được tư trang thuận lợi trên con đường phát triển sự nghiệp tương lai.

TÀI LIỆU THAM KHẢO

[1] Rajkumar Buyya, James Broberg, and Andrzej Gościński. “Introduction to Cloud Computing” in *Cloud Computing: Principles and Paradigms*. William Voorsluys, James Broberg, Rajikumar Buyya, John Wiley & Sons, 2011, pp. 3-41.

[2] NIST, Cloud Computing Program. Internet: http://www.nist.gov/itl/cloud/, [June. 20, 2014].

[3] Office365 Canada. (2011) On-premise, Hosted, or Microsoft Cloud? Internet: http://blogs.technet.com/b/office365\_canada/archive/2011/03/29/on-premisehosted-or-microsoft-cloud-part-one.aspx/, [June. 20, 2014].

[4] Brian Prince. 5 Ways to Use Microsoft Azure to Ease Cloud Migration. Internet: http://www.codeguru.com/csharp/article.php/c18091/5-Ways-to-UseMicrosoft-Azure-to-Ease-Cloud-Migration.htm/, [June. 20, 2014].

[5] Luis Rodero-Merino, Rajkumar Buyya Luis M. Vaquero, "Dynamically Scaling Applications in the Cloud," *ACM-Association for Computing Machinery*, 2011.

[6] Rajkumar Buyya, James Broberg, and Andrzej Gościński. “Migrating into a Cloud” in *Cloud Computing: Principles and Paradigms*. T. S. Mohan, John Wiley & Sons, 2011, pp. 43-56.

[7] Open Cloud Manifesto. Internet:http://www.opencloudmanifesto.org/, [Apr. 14, 2012].

[8] Open Cloud Consortium. Internet:http://opencloudconsortium.org/, [Apr. 14, 2012].

[9] Borko Furht and Armando J.Escalante, *Handbook of Cloud Computing* , Kindle Edition , 29-09-2010.

[10] Xin Chen. “Introduction to Application Frameworks” in *Developing Application Frameworks in .Net Apress*, 2004, pp. 1-10.

[11] Xin Chen. “Dissection of an Application Framework” in *Developing Application Frameworks in .Net Apress*, 2004, pp. 11-38.

[12] *Design Patterns: Elements of Reusable Object-Oriented Software*, by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, the “Gang Of Four” (GOF)”.

[13] AWS Architecture Center. Internet: http://aws.amazon.com/architecture/, [June. 20, 2014].

[14] James Murty. “Infrastructure in the Cloud” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[15] James Murty. “Interacting with Amazon Web Services” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[16] James Murty. “S3 Simple Storage Service” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[17] James Murty. “EC2 Elastic Compute Cloud” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[18] James Murty. “SQS Simple Queue Service” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[19] James Murty. “FPS Flexible Payments Service” in *Programming Amazon Web Service*. Publisher O’Reilly, 2008.

[20] Amazon Web Services. Internet: http://en.wikipedia.org/wiki/Amazon\_Web\_Services/, [June. 20, 2014].

[21] What is Amazon DynamoDB? Internet: docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html/, [June. 20, 2014].

[22] Service Highlights. Internet: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Features.html/, [June. 20, 2014].

[23] DynamoDB Data Model. Internet: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DataModel.html/, [June. 20, 2014].

[24] Supported Operations in DynamoDB. Internet: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/APISummary.html/, [June. 20, 2014].

[25] Provisioned Throughput in Amazon DynamoDB. Internet: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/ProvisionedThroughputIntro.html/, [June. 20, 2014].

[26] Accessing DynamoDB. Internet: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Endpoints.html/, [June. 20, 2014].

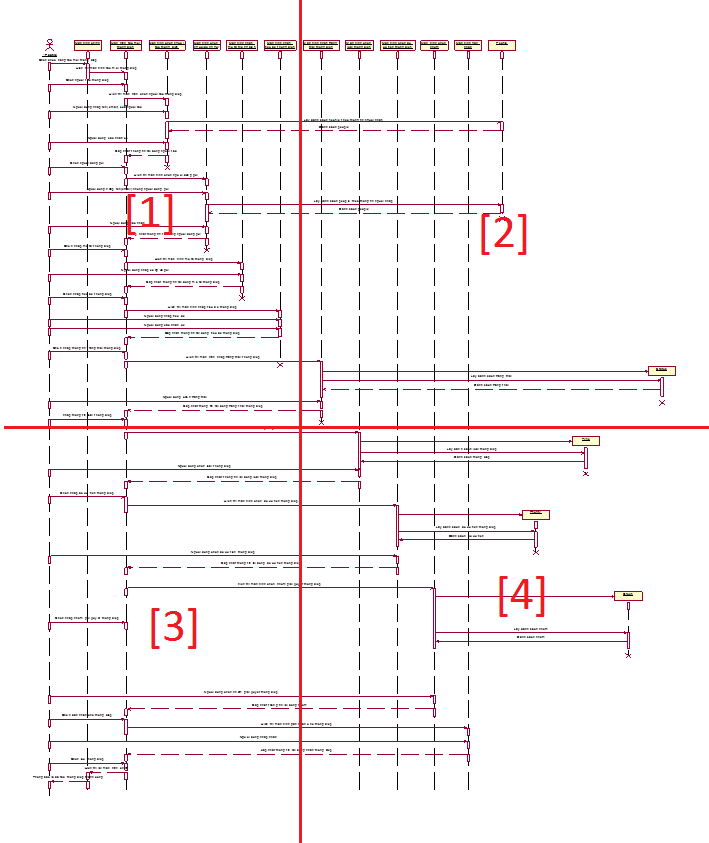
[27] Token Vending Machine for Anonymous Registration. Internet: http://aws.amazon.com/code/8872061742402990/, [June. 20, 2014].

[28] Codes for the Representation of Names of Languages. Internet: http://www.loc.gov/standards/iso639-2/php/code\_list.php/, [June. 20, 2014].

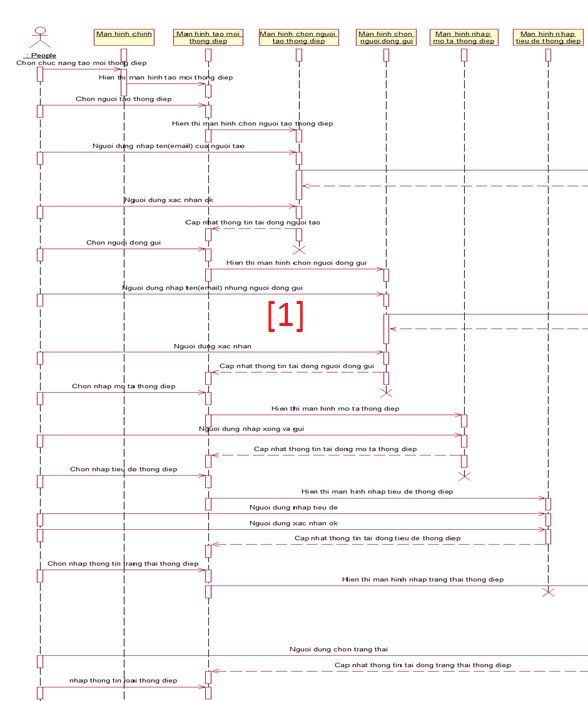
PHỤC LỤC

## **Phụ lục 1. Lược đồ tuần tự của ứng dụng**

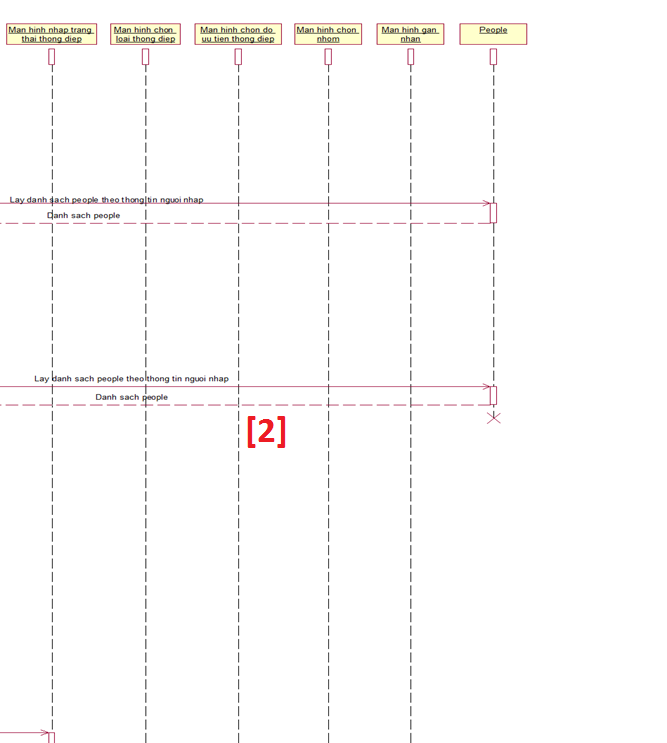
* 1. ***Lược đồ tuần tự chức năng Tạo mới thông điệp:***



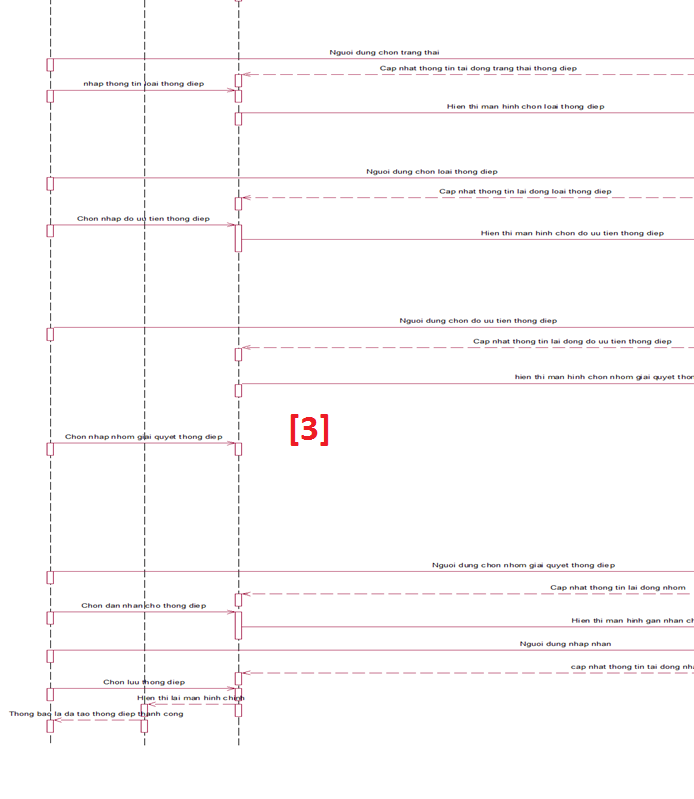
Hình phụ lục 1-1. Lược đồ tuần tự chức năng tạo mới thông điệp



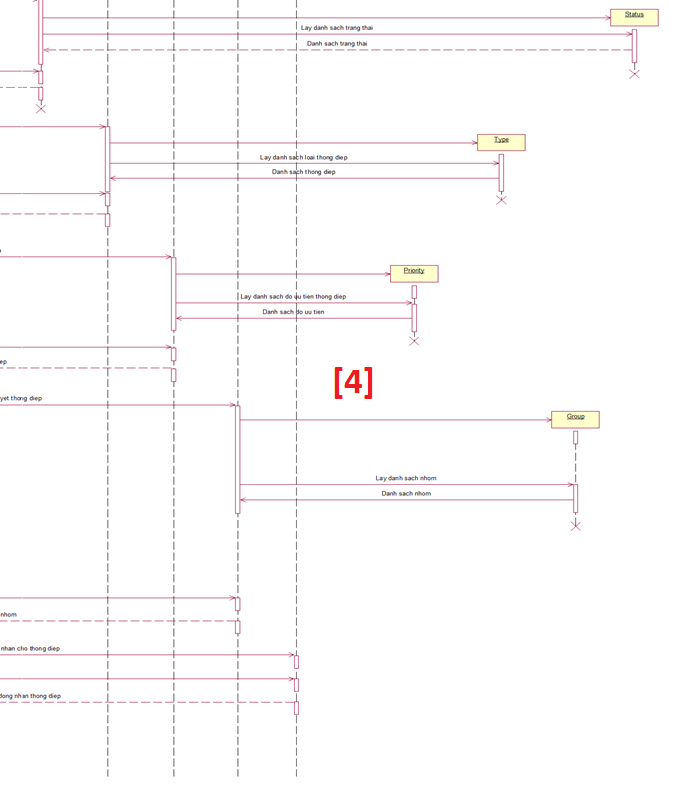
Hình phụ lục 1-1-1. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 1)



Hình phụ lục 1-1-2. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 2)



Hình phụ lục 1-1-3. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 3)



Hình phụ lục 1-1-4. Lược đồ tuần tự chức năng tạo mới thông điệp (phần 4)

* 1. ***Lược đồ tuần tự chức năng trả lời thông điệp:***



Hình phụ lục 1-2. Lược đồ tuần tự chức năng trả lời thông điệp

* 1. ***Lược đồ tuần tự chức năng sửa trạng thái thông điệp:***



Hình phụ lục 1-3. Lược đồ tuần tự chức năng sửa trạng thái thông điệp

* 1. ***Lược đồ tuần tự chức năng cập nhật vai trò người dùng:***



Hình phụ lục 1-4. Lược đồ tuần tự chức năng cập nhật vai trò người dùng

## **Phụ lục 2. Tạo chứng nhận định danh truy cập AWS bằng Amazon Cognito**

Amazon Cognito là một dịch vụ mạng mà quy trình chứng thực người dùng di động đến AWS một cách đơn giản. Amazon Cognito làm việc phối hợp với AWS Identity and Access Management (IAM) và AWS Security Token Service (AWS STS) để định danh duy nhất một người dùng và cung cấp cho họ một định danh phù hợp xuyên suốt vòng đời một ứng dụng. Dùng Amazon Cognito, bạn có thể tạo định danh truy cập duy nhất cho người dùng và chứng thực họ để có thể truy cập một cách bảo mật đến tài nguyên AWS.

Trước khi bạn tích hợp ứng dụng của bạn với Amazon Cognito, bạn cần tạo một hồ chứa định danh (identity pool) nơi mà lưu trữ đặc tả thông tin định danh người dùng đến tài khoản của bạn.

Để tạo một hồ chứa định danh cho những người dùng của bạn, bạn login vào màn hình chính Amazon Cognito của AWS. Thuật sĩ tạo hồ chứa định danh sẽ hướng dẫn cho bạn cách một hồ chứa định danh mới.

Khi một hồ chứa định danh mới được tạo, bạn sẽ có AWS account ID, Amazon Cognito identity pool ID, và ARNs để có thể khởi tạo Amazon Cognito client.

## **Phụ lục 3. Các chức năng ứng dụng cài đặt**

2. 1. ***Module 1: Nhóm chức năng quản lý:***

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Đăng ký tạo công ty | Quản trị viên ứng dụng Smart Help Desk có thể tạo ra 1 công ty mới, công ty này sẽ được cấp một tên miền hoạt động trong hệ thống. |
| Đăng ký người dùng | Người dùng có thể tạo ra một công ty mới, người dùng sẽ được cung cấp một tên miền cho công ty. |
| Phân quyền người dùng | Quản trị viên công ty có thể phân quyền được vai trò của người sử dụng, có thể cho họ là khách hàng, là nhân viên hổ trợ hoặc là quản trị viên công ty. |
| Thay đổi thông tin người dùng | Quản trị viên công ty sử dụng chức năng này để cập nhật những thay đổi trong thông tin cá nhân của người dùng. |
| Xóa người dùng | Quản trị viên công ty sử dụng chức năng này để xóa người dùng nào đó ra khỏi công ty. |
| Tạo phiếu thắc mắc | Chức năng này để tạo phiếu thắc mắc nào đó cần trao đổi. |
| Thay đổi thông tin phiếu thắc mắc | Quản trị viên công ty và nhân viên hổ trợ sử dụng chức năng này để thay đổi cập nhật thông tin của phiếu thắc mắc. |
| Xóa phiếu thắc mắc | Quản trị viên công ty và nhân viên hổ trợ sử dụng chức năng này để xóa một phiếu thắc mắc nào đó. |
| Tạo nhóm | Quản trị viên công ty thực hiện chức năng này để tạo một nhóm phục vụ cho việc trả lời các câu hỏi của khách hàng. |
| Xử lý thông tin nhóm | Quản trị viên công ty sử dụng chức năng này để thay đổi thông tin của những nhóm hỗ trợ khách hàng. |
| Xóa nhóm | Quản trị viên công ty thực hiện chức năng này để xóa nhóm hỗ trợ khách hàng nào đó. |
| Tạo tổ chức | Quản trị viên công ty thực hiện chức năng này để tạo ra tổ chức nào đó trong công ty. |
| Cập nhật thông tin tổ chức | Quản trị viên công ty thực hiện chức năng này để thay đổi thông tin của những tổ chức trong công ty. |
| Xóa tổ chức | Quản trị viên công ty thực hiện chức năng này để xóa tổ chức ra khỏi công ty. |

Bảng phụ lục 3-1. Nhóm chức năng quản lý

* 1. ***Module 2: Nhóm chức năng tìm kiếm:***

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Tìm kiếm. | Thực hiện tìm kiếm những phiếu thắc mắc theo tên, nội dung, email, tên người tạo có liên quan tới phiếu thắc mắc mà người dùng cung cấp, sau đó sẽ hiển thị kết quả lên màn hình cho người dùng. |

Bảng phụ lục 3-2. Nhóm chức năng tìm kiếm

* 1. ***Module 3: Nhóm chức năng thống kê – hiển thị:***

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Xem danh sách người dùng | Quản trị viên công ty sử dụng chức năng này để xem toàn bộ danh sách người dùng. |
| Xem danh sách phiếu thắc mắc | Người dùng thực hiện chức năng này để xem toàn bộ phiếu thắc mắc phù hợp với phân quyền của họ. Như khách hàng thì chỉ được xem phiếu thắc mắc của mình, Quản trị viên công ty có thể xem toàn bộ phiếu thắc mắc của cả nhân viên hổ trợ và khách hàng. |
| Xem danh sách tổ chức | Quản trị viên công ty sử dụng chức năng này để xem toàn bộ danh sách các tổ chức. |
| Xem thông tin cá nhân | Người dùng chọn chức năng này để xem thông tin cá nhân của mình. Ngoài ra quản trị viên công ty cũng sử dụng để xem thông tin của những người dùng trong công ty. |
| Xem thông tin phiếu thắc mắc | Người dùng chọn chức năng này để xem nội dung của phiếu thắc mắc trong quyền của mình. |
| Xem thông tin nhóm | Người dùng thuộc bộ phận công ty thực hiện chức năng này để xem thông tin cá nhân của những nhóm hỗ trợ trong công ty. |
| Xem thông tin tổ chức | Người dùng thuộc bộ phận công ty thực hiện chức năng này để xem thông tin riêng của những tổ chức trong công ty. |

Bảng phụ lục 3-3. Nhóm chức năng thống kê – hiển thị

* 1. ***Module 4: Nhóm chức năng hệ thống:***

|  |  |
| --- | --- |
| Thành phần | Diễn giải |
| Gửi yêu cầu | Người dùng là khách hàng sẽ thực hiện chức năng này để gửi một yêu cầu đến nhân viên hổ trợ hoặc quản trị viên công ty để được trợ giúp. |
| Đăng nhập | Người dùng chọn chức năng đăng nhập vào hệ thống. Tùy vào vai trò của từng người dùng sẽ tương ứng với các giao diện với các chức năng phù hợp với từng người. |
| Trả lời phiếu thắc mắc | Người dùng có thể nhập nôi dung trả lời cho phiếu thắc mắc. |

Bảng phụ lục 3-4. Nhóm chức năng hệ thống