

TRƯỜNG ĐẠI HỌC CẦN THƠ

Khoa Công Nghệ Thông Tin và Truyền Thông

-----🙞🙜🕮🙞🙜-----



**LUẬN VĂN TỐT NGHIỆP**

*Đề tài:*

**“APP QUẢN LÝ TIỀN ĐIỆN”**

-----🙞🙜 CẦN THƠ - 2017🙞🙜-----

**NHẬN XÉT CỦA GIÁO VIÊN**-----🙞🙜🕮🙞🙜-----…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….  
…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

…………………………………………………………………………………….

Cần Thơ, ngày 05 tháng 12 năm 2017

Giáo viên

**Mục Lục**

[TÓM LƯỢC 1](#_Toc498819857)

[ABSTRACT 2](#_Toc498819858)

[GIỚI THIỆU CHUNG 3](#_Toc498819859)

[I. Đặt vấn đề 3](#_Toc498819860)

[II. Mục tiêu của đề tài 3](#_Toc498819861)

[III. Các chức năng chính của ứng dụng 3](#_Toc498819862)

[IV. Nội dung nghiên cứu chính của đề tài 4](#_Toc498819863)

[1. Nghiên cứu lý thuyết: 4](#_Toc498819864)

[2. Áp dụng các nghiên cứu lý thuyết vào công việc cụ thể: 4](#_Toc498819865)

[V. Những đóng góp chính của đề tài: 4](#_Toc498819866)

[VI. Bố cục bài báo cáo luận văn 4](#_Toc498819867)

[1. Phần giới thiệu: 4](#_Toc498819868)

[2. Phần nội dung 5](#_Toc498819869)

[3. Phần kết luận 5](#_Toc498819870)

[PHẦN NỘI DUNG 6](#_Toc498819871)

[Chương 1: Cơ sở lý thuyết. 6](#_Toc498819872)

[I. Giới thiệu về Android và Android studio. 6](#_Toc498819873)

[1. Định nghĩa Android 6](#_Toc498819874)

[2. Kiến trúc hệ điều hành Android 6](#_Toc498819875)

[2.1. Linux Kernel 6](#_Toc498819876)

[2.2. Libraries 7](#_Toc498819877)

[2.3. Android Runtime 7](#_Toc498819878)

[2.4. Application Framework 8](#_Toc498819879)

[2.5. Application 8](#_Toc498819880)

[3. Android studio – công lập trình cho hệ điều hành android. 8](#_Toc498819881)

[3.1. Android studio 8](#_Toc498819882)

[3.2. Lập trình hệ điều hành Android 10](#_Toc498819883)

[3.2.1. Các thành phần trong một project Android 10](#_Toc498819884)

[3.2.1.1. AndroidManifest.xml 11](#_Toc498819885)

[3.2.1.2. file R.java 14](#_Toc498819886)

[3.2.1.3. MainActivity và activity\_main.xml 14](#_Toc498819887)

[3.2. Vòng đời của một ứng dụng Android 15](#_Toc498819888)

[3.2.1. Các loại tiến trình của Android 15](#_Toc498819889)

[3.2.2. Activity Stack 16](#_Toc498819890)

[3.2.3. Các trạng thái của Activity 17](#_Toc498819891)

[3.2.4. Vòng đời của Activity 18](#_Toc498819892)

[II. Giới thiệu về CSDL mongoDB. 19](#_Toc498819893)

[1. mongoDB là gì? 19](#_Toc498819894)

[2. Các lợi thế khi sử dụng mongoDB. 19](#_Toc498819895)

[3. Tại sao nên sử dụng mongoDB 20](#_Toc498819896)

[4. Nên sử dụng mongoDB ở đâu. 20](#_Toc498819897)

[5. Giao diện mongoDB (Studio 3T for MongoDB) 21](#_Toc498819898)

[II. Giới thiệu về Web service và kiến trúc RESTful. 21](#_Toc498819899)

[1. Web Service là gì? 21](#_Toc498819900)

[2. Đặc điểm của Web Service 21](#_Toc498819901)

[2.1. Đặc điểm 21](#_Toc498819902)

[2.2. Ưu điểm và nhược điểm 22](#_Toc498819903)

[2.2.1. Ưu điểm 22](#_Toc498819904)

[2.2.2. Nhược điểm 22](#_Toc498819905)

[2.3. Mô hình Web Service 23](#_Toc498819906)

[3. Kiến trúc RESTful 23](#_Toc498819907)

[3.1 RESUful là gì? 23](#_Toc498819908)

[3.2. Các quy tắt mà REST định nghĩa 23](#_Toc498819909)

[Chương 2: Thiết kế và xây dựng ứng dụng 24](#_Toc498819910)

[I. Kế hoạch phát triển và đảm bảo chất lượng. 24](#_Toc498819911)

[1. Mục đích 24](#_Toc498819912)

[2. Quản lý 24](#_Toc498819913)

[2.1. Công việc 24](#_Toc498819914)

[2.1.1 Các giai đoạn của chu kỳ sống của phần mềm 24](#_Toc498819915)

[2.1.2 Quy trình kiểm soát sự phát triển của phần mềm 25](#_Toc498819916)

[3. Tài liệu 25](#_Toc498819917)

[3.1. Mục đích 25](#_Toc498819918)

[3.2. Yêu cầu tài liệu tối thiểu: 25](#_Toc498819919)

[3.2.1 Mô tả yêu cầu phần mềm 25](#_Toc498819920)

[3.2.2 Mô tả thiết kế phần mềm 26](#_Toc498819921)

[3.2.3 Các kế hoạch kiểm tra và công nhận hợp lệ 26](#_Toc498819922)

[3.2.4 Báo cáo thẩm tra và công nhận hợp lệ 26](#_Toc498819923)

[3.2.5 Quản lý cấu hình phần mềm 27](#_Toc498819924)

[4. Các chuẩn, thực tiễn, qui ước và phép đo 27](#_Toc498819925)

[4.1. Mục đích 27](#_Toc498819926)

[4.2. Nội dung 27](#_Toc498819927)

[5. Xem lại phần mềm 27](#_Toc498819928)

[5.1. Mục đích 27](#_Toc498819929)

[5.2. Các yếu tố tối thiểu 27](#_Toc498819930)

[5.2.1 Xem lại các thiết kế kiến trúc 28](#_Toc498819931)

[5.2.2 Xem lại đặc tả phẩn mềm 28](#_Toc498819932)

[5.2.3 Xem lại chi tiết thiết kế 28](#_Toc498819933)

[5.2.4 Xem lại kế hoạch thẩm tra và công nhận hợp lệ 28](#_Toc498819934)

[5.2.5 Kiểm toán chức năng 28](#_Toc498819935)

[5.2.6 Kiểm toán vật lý 28](#_Toc498819936)

[5.2.7 Xem lại về quản lý 28](#_Toc498819937)

[5.2.8 Xem lại kế hoạch quản lý cấu hình 28](#_Toc498819938)

[II. Đặc tả yêu cầu ứng dụng. 28](#_Toc498819939)

[1. Mục tiêu 28](#_Toc498819940)

[2. Phạm vi sản phẩm 29](#_Toc498819941)

[3. Tài liệu tham khảo 29](#_Toc498819942)

[4. Mô tả tổng quan 29](#_Toc498819943)

[4.1. Bối cảnh sản phẩm 29](#_Toc498819944)

[4.2. Mô tả bài toán 29](#_Toc498819945)

[4.3. Chức năng của sản phẩm 30](#_Toc498819946)

[4.4. Các ràng buộc thực thi và thiết kế 30](#_Toc498819947)

[4.5. Các giả định phụ thuộc 30](#_Toc498819948)

[III. Thiết hệ thống, CSDL, thiết kế ứng dụng, mô tả hoạt động của từng chức năng. 31](#_Toc498819949)

[1. Thiết kế kiến trúc 31](#_Toc498819950)

[2.1. Mô tả dữ liệu thiết kế 33](#_Toc498819951)

[2.2 Các bảng CSDL 34](#_Toc498819952)

[2.3. Sơ đồ xử lý dữ liệu 39](#_Toc498819953)

[3. Thiết kế ứng dụng 39](#_Toc498819954)

[3.1. Sơ đồ use-case 39](#_Toc498819955)

[3.2. Xây dựng chức năng 40](#_Toc498819956)

[4. Thiết kế giao diện theo chức năng 40](#_Toc498819957)

[4.1. Đăng nhập 41](#_Toc498819958)

[4.1.1. Mô tả và mức độ ưu tiên 41](#_Toc498819959)

[4.1.2. Tác nhân và chuỗi đáp ứng. 41](#_Toc498819960)

[4.1.3. Các yêu cầu chức năng 42](#_Toc498819961)

[4.2. Xem thông tin 42](#_Toc498819962)

[4.2.1. Mô tả và mức ưu tiên 42](#_Toc498819963)

[4.2.2. Tác nhân và chuỗi đáp ứng 43](#_Toc498819964)

[4.2.3. Các ràng buộc chức năng 43](#_Toc498819965)

[4.3. Đổi mật khẩu 43](#_Toc498819966)

[4.3.1. Mô tả vầ mức độ ưu tiên 43](#_Toc498819967)

[4.3.2. Tác nhân và chuỗi đáp ứng 44](#_Toc498819968)

[4.3.3. Các ràng buộc chức năng 44](#_Toc498819969)

[4.4. Tìm kiếm thông tin 45](#_Toc498819970)

[4.4.1. Mô tả và mức độ ưu tiên 45](#_Toc498819971)

[4.4.2. Tác nhân và chuỗi đáp ứng 45](#_Toc498819972)

[4.4.3. Các ràng buộc chức năng 46](#_Toc498819973)

[4.5. Thêm người dùng 46](#_Toc498819974)

[4.5.1 Mô tả và mức độ ưu tiên 46](#_Toc498819975)

[4.5.2. Tác nhân và chuỗi đáp ứng 46](#_Toc498819976)

[4.5.3. Các ràng buộc chức năng 47](#_Toc498819977)

[4.6. Xóa Người dùng 47](#_Toc498819978)

[4.6.1. Mô tả và mức độ ưu tiên 47](#_Toc498819979)

[4.6.2. Tác nhân và chuỗi đáp ứng 48](#_Toc498819980)

[4.6.3. Các ràng buộc chức năng 48](#_Toc498819981)

[4.7. Cập nhật mật khẩu cho người dùng 48](#_Toc498819982)

[4.7.1. Mô tả và mức độ ưu tiên 48](#_Toc498819983)

[4.7.2. Tác nhân và chuỗi đáp ứng 49](#_Toc498819984)

[4.7.3. Các ràng buộc chức năng 49](#_Toc498819985)

[4.8. Cập nhật trạng thái thanh toán tiền của khách hàng. 49](#_Toc498819986)

[4.8.1. Mô tả và mức độ ưu tiên 50](#_Toc498819987)

[4.8.2 Tác nhân và chuỗi đáp ứng 50](#_Toc498819988)

[4.8.3. Các ràng buộc chức năng 50](#_Toc498819989)

[4.9. Cập nhật chỉ số của điện kế. 50](#_Toc498819990)

[4.9.1. Mô tả và mức độ ưu tiên 51](#_Toc498819991)

[4.9.2. Tác nhân và chuỗi đáp ứng 51](#_Toc498819992)

[4.9.3. Các ràng buộc chức năng 51](#_Toc498819993)

[4.10. Hiển thị danh sách khách hàng theo khu vực. 52](#_Toc498819994)

[4.10.1. Mô tả và mức độ ưu tiên 52](#_Toc498819995)

[4.10.2. Tác nhân và chuỗi đáp ứng 52](#_Toc498819996)

[4.10.3. Các ràng buộc chức năng 53](#_Toc498819997)

[4.11. Hiển thị địa chỉ khách hàng trên bản đồ. 53](#_Toc498819998)

[4.11.1. Mô tả và mức độ ưu tiên 53](#_Toc498819999)

[4.11.2. Tác nhân và chuỗi đáp ứng 53](#_Toc498820000)

[4.11.3. Các ràng buộc chức năng 54](#_Toc498820001)

[5. Yêu cầu phi chức năng 54](#_Toc498820002)

[5.1. Yêu cầu thực thi 54](#_Toc498820003)

[5.2. Yêu cầu an toàn 54](#_Toc498820004)

[5.3. Yêu cầu bảo mật 54](#_Toc498820005)

[5.4. Các đặc điểm chất lượng phần mềm 54](#_Toc498820006)

[Chương 3: Kiểm thử và đánh giá. 54](#_Toc498820007)

[I. Kiểm thử 54](#_Toc498820008)

[1. Mục tiêu 54](#_Toc498820009)

[2. Môi trường kiểm thử 54](#_Toc498820010)

[3. Kiểm thử chức năng 55](#_Toc498820011)

[4. Cách tiếp cận 55](#_Toc498820012)

[5. Tiêu chí thử thành công/thất bại 55](#_Toc498820013)

[6. Kiểm thử 55](#_Toc498820014)

[6.1. Kiểm thử thành công. 55](#_Toc498820015)

[6.2. Kiểm thử thất bại 56](#_Toc498820016)

[PHẦN KẾT LUẬN 56](#_Toc498820017)

[I. Kết quả đạt được 56](#_Toc498820018)

[II. Ưu điểm 56](#_Toc498820019)

[III. Hạn chế 57](#_Toc498820020)

[IV. Hướng phát triển 57](#_Toc498820021)

[Tài liệu tham khảo 58](#_Toc498820022)

**Danh Mục Hình**

[Hình 1. Hệ điều hành Android. 6](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820023)

[Hình 2. Android Studio 9](#_Toc498820024)

[Hình 3. Cấu trúc một android project 10](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820025)

[Hình 4. Cấu trúc file AndroidManifest.xml 13](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820026)

[Hình 5. File activity\_main.xml 15](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820027)

[Hình 6. File MainActivity.java 15](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820028)

[Hình 7. Activity Stack 16](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820029)

[Hình 8. Các trạng thái của Activity 17](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820030)

[Hình 9. Vòng đời của một Activity 18](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820031)

[Hình 10. mongoDB 19](#_Toc498820032)

[Hình 11. Giao diện Studio 3T for mongoDB 21](#_Toc498820033)

[Hình 12. Mô hình Web Service 23](#_Toc498820034)

[Hình 13. Đăng nhập 41](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820035)

[Hình 14. Xem thông tin cá nhân 42](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820036)

[Hình 15. Đổi mật khẩu cá nhân 43](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820037)

[Hình 16. Tìm kiếm khách hàng 45](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820038)

[Hình 17. Tìm kiếm nhân viên 45](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820039)

[Hình 18. Thêm khách hàng 46](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820040)

[Hình 19. Thêm nhân viên 46](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820041)

[Hình 20. Xóa người dùng 47](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820042)

[Hình 21. Xác nhận xóa người dùng 47](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820043)

[Hình 22. Cập nhật mật khẩu cho người dùng 48](#_Toc498820044)

[Hình 23. Trang thái chưa thanh toán 49](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820045)

[Hình 24. Trạng thái đã thanh toán 49](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820046)

[Hình 25. Cập nhật chỉ số điện kế 50](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820047)

[Hình 26. Hiển thị danh sách khách hàng theo khu vực 52](#_Toc498820048)

[Hình 27. Hiển thị địa chỉ khách hàng trên bản đồ 53](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820049)

**Danh Mục Mô Hình**

[Mô hình 1. Mô hình phân rã 31](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820050)

[Mô hình 2. Mô hình phân ra administator 32](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820051)

[Mô hình 3. Mô hình phân rã nhân viên 33](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820052)

[Mô hình 4. Mô hình quan hệ cơ sở dữ liệu 33](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820053)

**Danh Mục Sơ Đồ**

[Sơ đồ 1. Sơ đồ kiến trúc 31](C:\\Users\\tranm\\Documents\\Android\\Luận Văn TMTAI 2017.docx" \l "_Toc498820054)

[Sơ đồ 2. Sơ đồ xử lý dữ liệu 39](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820055)

[Sơ đồ 3. Sơ đồ use-case 39](file:///C:\Users\tranm\Documents\Android\Luận%20Văn%20TMTAI%202017.docx#_Toc498820056)

[Sơ đồ 4. Sơ đồ xử lý dữ liệu 41](#_Toc498820057)

# TÓM LƯỢC

Hiện nay, cùng với việc internet phát triển cực kì mạnh mẽ là sự ra đời của các dòng điện thoại thông minh (smart phone). Điện thoại thông minh đóng góp một phàn không nhỏ vào cuộc sống hiện nay, việc tính toán, lưu giữ thông tin, quản lý công việc trên điện thoại cũng dần được chú trọng hơn. Ngoài ra, với sự phát triển mạnh mẽ của hệ điều hành Android, việc xây dựng ứng dụng quản lý hay tính toán trên hệ điều hành này là việc làm hết sức thiết thực.

Do đó đề tài “Xây dựng ứng dụng quản lý tiền điện trên hệ điều hành Android được thực hiện nhằm mục đích tìm hiểu cách thức phát triển một ứng dụng hoàn chỉnh trên hệ điều hành Android, áp dụng những kiến thức đã học ở trường để xây dựng nên một ứng dụng hoàn chỉnh. Qua quá trình thực hiện đã đạt được kết quả:

Tìm hiểu kiếm thức về hệ điều hành di động Android

Hiểu được cách thức hoạt động của một ứng dụng, cách xây dựng một ứng dụng hoàn thiện

Tạo ra một ứng dụng từ điển mang tính thực tiễn cao, hữu ích trong cuộc sống và công việc.

Tận dụng được các kiến thức đã được học ở trường để áp dụng vào việc xây dựng một ứng dụng thực tế.

# ABSTRACT

Currenly, along with the internet is extremely powerful development is the introduction of smart phones. Smartphones contribute a small part to today's life, calculating, storing information, managing the work on the phone is also more focused. Also, with the robust development of the Android operating system, building a managed or computed application on this operating system is a real deal.

Therefore, the topic of "Building a power management application on Android operating system is done to learn how to develop a complete application on the Android operating system, applying the knowledge learned in the school to build a complete application. Through the implementation process has achieved results:

Learn about the Android mobile operating system

Understand how an application works, how to build a complete application

Create a highly practical, useful dictionary in life and work.

Take advantage of the knowledge you have learned in school to apply to building a real application.

# GIỚI THIỆU CHUNG

# I. Đặt vấn đề

Kể từ khi Apple trình làng thế hệ Iphone đầu tiên vào năm 2007, thị trường điện thoại di động thông minh đã bùng nổ và phát triển mạnh mẽ, cùng với đó, android dần vươn lên và vượt qua các hệ điều hành khác (iOS, Windows Phone, Symbia, Tizen, …) để trở thành hệ điều hành di động phổ biến nhất. Theo thống kê mới nhất, điện thoại di động thông minh sử dụng hệ điều hành android đang chiếm lĩnh hoàn toàn thị trường với 88% thị phần tính đến quý 3/2016 (nghiên cứu của công ty Strategy Analytics).

Cùng với sự phát triển mạnh mẽ của của thị trường điện thoại di động đó là sự phát triển của xu hướng lập trình trên thiết bị di động, do đó việc xây dựng ứng dụng cho điện thoại di động mà tiêu biểu là điện thoại Android trở thành một ngành công nghiệp đầy tiềm năng và hứa hẹn mang đến nhiều lợi ích. Các chuyên gia CNTT có kỹ năng phát triển ứng dụng di động là một mặt hàng "hot" hiện nay và nhu cầu đối với các lập trình viên Android thậm chí còn "hot" hơn. Các nhà tuyển dụng đang tìm thuê các lập trình viên Android nhanh hơn và nhiều hơn so với bất kỳ chuyên gia trong lĩnh vực công nghệ di động nào khác. Theo Matt Miller, Giám đốc công nghệ tại CyberCoders, "Đối với các lập trình viên di động chuyên và quen thuộc với hệ điều hành Android, không có thời điểm nào tốt hơn để cải thiện kỹ năng Android của bạn và tìm kiếm một công việc".

Hiện nay, điện thoại giúp chúng ta không ít trong công việc. Một ứng dụng quản lý thông tin giúp chung ta lưu giữ các thông tin cần thiết và rất tiện lợi, không cẩn phải dùng đến máy tính, khi đó, chúng ta có thể thoải mái di chuyển ở bất kì nơi đâu.

Một nhân viên thu phí tiền điện sẽ thuận tiện hơn rất nhiều nếu địa chỉ của các khách hàng được hiển thị lên điện thoại với sự hỗ trợ của google map. Tất cả các thông tin sẽ được hiển thị một các rõ ràng và chi tiết nhất, khi đó, việc tra cứu thông tin sẽ rất nhanh chóng không tốn quá nhiều thời gian và hiệu suất công việc cũng sẽ tăng lên đáng kể.

# II. Mục tiêu của đề tài

Đề tài “Xây dựng ứng dụng quản lý tiền điện trên hệ điều hành Android” được thực hiện với những mục tiêu sau đây:

- Tìm hiểu về hệ điều hành di động Android.

- Tìm hiểu cách xây dựng một ứng dụng cụ thể trên hệ điều hành này.

- Sử dụng các kiến thức về lập trình, cơ sở dữ liệu để áp dụng vào một công việc cụ thể.

- Tạo ra một ứng dụng mang tính thực tiễn cao. Sử dụng được trong thực tế hằng ngày.

- Củng cố lại những kiến thức đã học ở trường.

# III. Các chức năng chính của ứng dụng

- Quản lý nhân viên, quản lý khách hàng.

- Tra cứu thông của của nhân viên, khách hàng.

- Sử dụng google map để hỗ trợ hiển thị danh sách địa chỉ của khách hàng.

- Quản lý thông tin tiền điện của khách hàng.

# IV. Nội dung nghiên cứu chính của đề tài

## 1. Nghiên cứu lý thuyết:

- Tìm hiểu sơ lược về hệ điều hành Android, lịch sử ra đời và phát triển của hệ điều hành này, kiến trúc của hệ điều hành này, cách thức hoạt động của một ứng dụng trên hệ điều hành Android.

- Tìm hiểu về môi trường, ngôn ngữ lập trình và phát triển ứng dụng trên hệ điều hành Android.

- Tìm hiểu về các thành phần giao diện, cách tạo, bố trí giao diện, tương tác với giao diện.

- Tìm hiểu về Web Service – RESTful.

- Tìm hiểu về cơ sở dữ liệu mongoDB, các sử dụng cơ sở dữ liệu này và làm thế nào để sử dụng nó cho một ứng dụng Android cụ thể.

## 2. Áp dụng các nghiên cứu lý thuyết vào công việc cụ thể:

- Vẽ giao diện cho ứng dụng:

- Xử lý, tương tác với các thành phần giao diện.

- Xây dựng server với các API cần thiết.

- Thao tác truy vấn với các dữ liệu trong cơ sở dữ liệu mongoDB với server.

- Xây dựng ứng dụng theo từ chức năng cần thiết.

- Liên kết ứng dụng với server.

# V. Những đóng góp chính của đề tài:

Việc thực hiện đề tài này đã giúp bản thân em góp phần củng cố lại những kiến thức đã học, tìm hiểu thêm các kiến thức bổ ích về hệ điều hành Android cũng như làm ra được một ứng dụng từ điển hết sức cần thiết, hỗ trợ trong công việc và cuộc sống của bản thân em cũng như có thể chia sẻ để mọi người cùng sử dụng.

# VI. Bố cục bài báo cáo luận văn

Bố cục quyển luận văn bao gồm 3 phần chính: Phần giới thiệu, phần nội dung và phần kết luận.

## 1. Phần giới thiệu:

- Trình bày về vấn đề cần thiết của việc xây dựng ứng dụng trên hệ điều hành Android.

- Trình bày các mục tiêu của đề tài.

- Các chức năng chính, quan trọng của ứng dụng.

- Nội dung nghiên cứu của đề tài.

- Những đóng góp chính của đề tài.

- Bố cục của quyển luận văn.

## 2. Phần nội dung

Phần này trình bày lý thuyết về hệ điều hành Android, cách lập trình trên hệ điều hành Android, thiết kế và cài đặt giải pháp cho ứng dụng, xây dựng server và các API, triển khai ứng dụng, kiểm thử và đánh giá ứng dụng. Bao gồm 3 chương sau:

Chương 1: Cơ sở lý thuyết.

I. Giới thiệu về Android và Android studio.

II. Giới thiệu về CSDL mongoDB.

III. Giới thiệu về Web service và kiến trúc RESTful.

Chương 2: Thiết kế và xây dựng ứng dụng

I. Kế hoạch phát triển và đảm bảo chất lượng.

II. Đặc tả yêu cầu ứng dụng.

III. Thiết kế CSDL, thiết kế ứng dụng, mô tả hoạt động của từng chức năng.

Chương 3: Kiểm thử và đánh giá.

Kiểm tra lại các chức năng xem có hoạt động đúng như yêu cầu đặt ra hay không, kiểm tra xem có lỗi phát sinh trong quá trình sử dụng hay không. Ứng dụng có thể hoạt động trên một thiết bị cụ thể hay không. Đưa ra đánh giá về các chức năng.

## 3. Phần kết luận

- Kết quả đạt được.

- Hướng phát triển.

# PHẦN NỘI DUNG

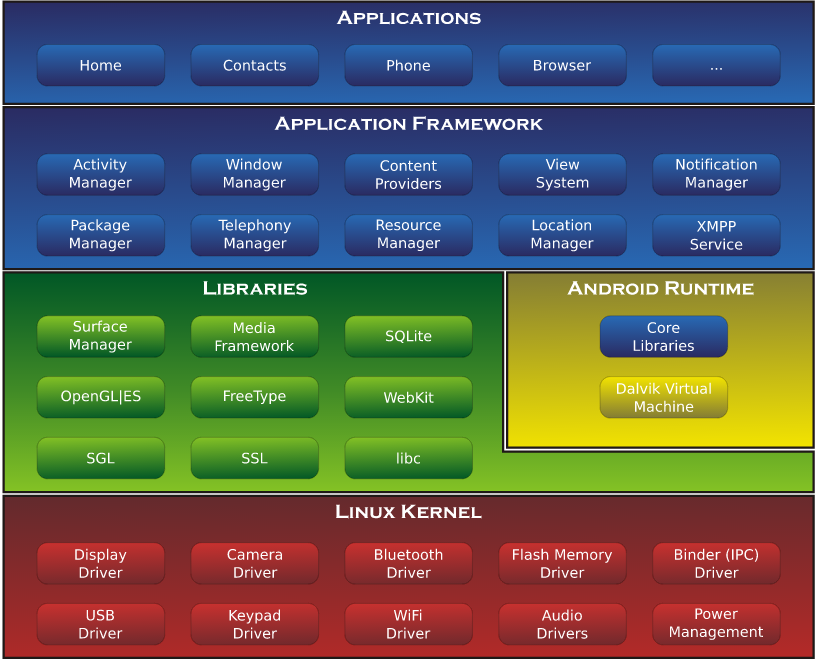
# Chương 1: Cơ sở lý thuyết.

# I. Giới thiệu về Android và Android studio.

## 1. Định nghĩa Android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

## 2. Kiến trúc hệ điều hành Android

 Hệ điều hành Android là một ngăn xếp của các thành phần ứng dụng (stack of softwave components), có thể chia thành 5 phần và 4 lớp như trong sơ đồ kiến trúc Android dưới đây.

Hình 1. Hệ điều hành Android.

### 2.1. Linux Kernel

Dưới cùng là lớp Linux - Lớp này cung cấp 1 cấp độ trừu tượng giữa phần cứng của thiết bị và các thành trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình hiển thị... Đồng thời, hạt nhân (kernel) còn xử lý tất cả các thứ mà Linux có thể làm tốt như mạng kết nối và 1 chuỗi các trình điều khiển thiết bị, giúp cho giao tiếp với các thiết bị ngoại vi dễ dàng hơn.

### 2.2. Libraries

Ở trên lớp nhân Linux là tập các thư viện bao gồm WebKit - trình duyệt Web mã nguồn mở, được biết đến như thư viện libc, cơ sở dữ liệu SQLite - hữu dụng cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, các thư viênj chơi và ghi âm audio, video, hay các thư viện SSL chiụ trách nhiệm bảo mật Internet.

Đây là các thư viện dựa trên Java phục vụ cho việc phát triển Android. Ví dụ của các thư viện này bao gồm các thư viện ứng dụng dùng để xây dựng giao diện người dùng, vẽ đồ họa hay truy cập cơ sở dữ liệu. 1 số thư viện chính của Android:

- **android.app** - Cung cấp quyền truy cập vào ứng dụng và là nền tảng của tất cả ứng dụng Android.

- **android.content** - Cung cấp quyền truy cập nội dung (content), truyền tải thông điệp giữa các ứng dụng hay các thành phần của ứng dụng.

- **android.database** - Đựoc sử dụng để truy cập dữ liệu của content provider và cơ sở dữ liệu SQLite.

- **android.opengl** - giao diện các phuơng thức Java để sử dụng OpenGL ES

- **android.os** - Cung cấp các ứng dụng với quyền truy cập vào các dịch vụ của hệ điều hành bao gồm thông điệp, các dịch vụ hệ thống và các giao tiếp nội tại.

- **android.text** - Đựoc sử dụng để hiển thị và điều chỉnh chữ trên màn hình thiết bị.

- **android.view** - Các thành phần cơ bản trong việc xây dựng giao diện người dùng của ứng dụng.

- **android.widget** - Tập các thành phần giao diện người dùng đã được xây dựng sẵn như các nút, các nhãn (label), list view, …

**- android.webkit** - Tập các lớp cho phép xây dựng khả năng duyệt web.

### 2.3. Android Runtime

Đây là phần thứ 3 của kiến trúc và nằm ở lớp thứ 2 từ dưới lên. Phần này cung cấp 1 bộ phận quan trọng là Dalvik Virtual Machine - là 1 loại Java Virtual Machine được thiết kế đặc biệt để tối ưu cho Android.

Dalvik VM sử dụng các đặc trưng của nhân Linux như quản lý bộ nhớ và đa luồng, những thứ mà đã có sẵn trong Java. Dalvik VM giúp mọ ứng dụng Android chạy trong tiến trình riêng của nó, với các thể hiện (instance) riêng của Dalvik virtual Machine.

Android Runtime cũng cung cấp 1 tập các thư viện chính giúp các nhà phát triển ứng dụng Android có thể viết ứng dụng Android bằng Java.

### 2.4. Application Framework

Lớp Android Framework cung cấp các dịch vụ cấp độ cao hơn cho các ứng dụng dưới dạng các lớp Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong ứng dụng của họ.

Android Framework bao gồm các dịch vụ chính sau:

- **Activitty Manager** - Kiểm soát tất cả khía cạnh của vòng đời ứng dụng và ngăn xếp các Activity.

- **Content Providers** - Cho phép các ứng dụng chia sẽ dữ liệu với các ứng dụng khác.

- **Resource Manager** - Cung cấp quyền truy cập vào các tài nguyên như các chuỗi, màu sắc, các layout giao diện người dùng...

- **Notifications Manager** - Cho phép các ứng dụng hiển thị cảnh báo và các thông báo cho người dùng.

- **View System** - Tập các thành phần giao diện (view) được sử dụng để tạo giao diện người dùng.

### 2.5. Application

Lớp trên cùng của kiến trúc là Application. Các ứng dụng bạn tạo ra sẽ được cài đặt trên lớp này. Ví dụ như: Danh bạ, nhắn tin, trò chơi, …

## 3. Android studio – công lập trình cho hệ điều hành android.

### 3.1. Android studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển nền tảng Android.

Nó được ra mắt vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Android Studio được phát hành miễn phí theo giấy phép Apache Licence 2.0.

Dựa trên phần mềm [IntelliJ IDEA](https://vi.wikipedia.org/w/index.php?title=IntelliJ_IDEA&action=edit&redlink=1) của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android

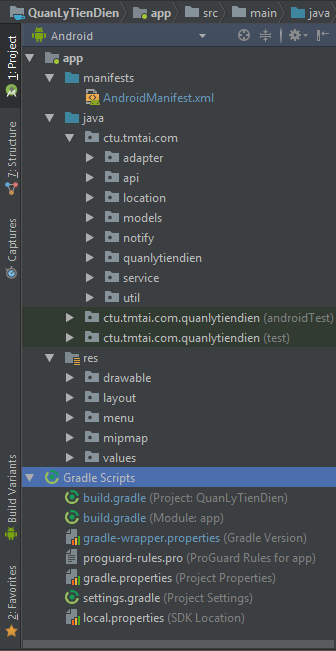
Nó hỗ trợ các hệ điều hành [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](https://vi.wikipedia.org/wiki/OS_X) và [Linux](https://vi.wikipedia.org/wiki/Linux), và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho [Android Development Tools](https://vi.wikipedia.org/w/index.php?title=Android_Development_Tools&action=edit&redlink=1) (ADT) dựa trên [Eclipse](https://vi.wikipedia.org/wiki/Eclipse).



Hình 2. Android Studio

### 3.2. Lập trình hệ điều hành Android

#### 3.2.1. Các thành phần trong một project Android

 Dưới đây là hình mô phỏng cấu trúc cây của một project Android - **Thư mục Manifiest:** Đầu tiên là thư mục manifest chứa file AndroidManifest.xml (sẽ được đề cập ở mục 2.1.1 bên dưới).

Hình 3. Cấu trúc một android project

- **Thư mục Java:** là nơi chứa các package của dự án, ta có thể tạo package ở đây và các package sẽ lưu trữ các lớp, chứa source code của ứng dụng. Như trên hình thì thư ctu.tmtai.com sẽ chứa các lớp mã nguồn của ứng dụng (MainActivity.java sẽ được nói rõ hơn trong mục 2.1.4).

- **Thư mục res** (Resources): Resources là nơi chứa mọi tài nguyên tĩnh của ứng dụng như sounds, pictures, colors… Ứng dụng muốn truy xuất tài nguyên phải thông qua lớp này bằng hàm getResources (). Thư mục res gồm các thư mục con:

- **Thư mục drawable:** Đây chính là thư mục chứa các file hình ảnh, config xml… trong dự án android. Ví dụ khi ta muốn ứng dụng sử dụng một hình ảnh nào đó là background thì ảnh đó sẽ được bỏ vào thư mục này. Hoặc một muốn điều chỉnh một nút button khi click vào màu xanh còn khi không click vào màu trắng thì chúng ta sẽ config trong file xml và lưu vào trong này.

- **Thư mục layout:** Đây chính là thư mục lưu các file xml về giao diện của các màn hình ứng dụng. Như đề cập ở trên thư mục Java ta có các package lưu các class, các class này sẽ kết nối với các file xml trong thư mục layout nào để tạo nên một màn hình có giao diện cho người dùng thao tác.

- **Thư mục mipmap:** Đây là thư mục sẽ chứa ảnh logo ứng dụng, chúng ta nên lưu logo của ứng dụng ở thư mục này (mặc dù vẫn có thể lưu ở nơi khác).

- **Thư mục values:** Thư mục này sẽ có rất nhiều file ở bên trong như sau:

- **Tập tin color.xml:** đây là file định nghĩa các mã màu trong dự án android, khi sử dụng màu nào chúng ta chỉ cần gọi tên tên mã màu đã định nghĩa trong đây.

- **Tập tin dimens.xml:** đây là file sẽ định nghĩa ra các kích thước như cỡ chữ, chiều cao, chiều rộng các view.

- **Tập tin strings.xml:** đây là file định nghĩa các đoạn văn bản trong ứng dụng Android ví dụ như chúng ta có 1 đoạn văn bản mà sử dụng đi sử dụng lại trong các màn hình khác nhau, khi chúng ta set cứng ở nhiều nơi thì khi cần chỉnh sửa thì phải tìm hết tất cả và sửa lại. Bây giờ chỉ cần định nghĩa đoạn văn bản đó trong đây và khi dùng thì gọi ra sử dụng và sau này chỉnh sữa thì chỉ cần sửa trong đây là xong, nó sẽ áp dụng cho tất cả mọi nơi.

- **Tập tin styles.xml:** đây chính là nơi định nghĩa các giao diện của các file layout trong thư mục layout đã nói phía trên. Ví dụ, khi chúng ta muốn chỉnh một nút Button chiều cao 10dp, chiều rộng 10dp, màu xanh… và chúng ta lại sử dụng kiểu thiết kế này ở 5 màn hình khác nhau. Do đó không thể mỗi màn hình lại định nghĩa lại như thế sẽ làm duplicate code (lặp lại) và sẽ không tối ưu ứng dụng được. Thay vào đó chúng ta chỉ cần định nghĩa một file giao diện như trên và ở mỗi màn hình chỉ cần gọi là xong.

Ở phía dưới có các file khác mà chúng ta cũng không cần quan tâm nhiều, riêng có 2 file build.grade là file để thiết lập các thuộc tính cho dự án android như: phiên bản SDK, Vesion ứng dụng, package, thêm thư viện ngoài.

#### 3.2.1.1. AndroidManifest.xml

\* AndroidManifest là gì?

Là file để bản kê khai trình bày những thông tin thiết yếu về ứng dụng của bạn với hệ thống Android, thông tin mà hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng. Hay nói cách khác đây là file dùng để config những thuộc tính cho ứng dụng mà khi ứng dụng khởi chạy hệ điều hành có thể hiểu được và xử lí.

\* AndroidManifest để làm gì?

- Nó đặt tên gói Java cho ứng dụng. Tên gói đóng vai trò như một mã nhận diện duy nhất cho ứng dụng.

- Nó mô tả các thành phần của ứng dụng - hoạt động, dịch vụ, hàm nhận quảng bá, và trình cung cấp nội dung mà ứng dụng được soạn bởi. Nó đặt tên các lớp triển khai từng thành phần và công bố các khả năng của chúng (ví dụ, những tin nhắn Intent mà chúng có thể xử lý). Những khai báo này cho phép hệ thống Android biết các thành phần là gì và chúng có thể được khởi chạy trong những điều kiện nào.

- Nó xác định những tiến trình nào sẽ lưu trữ các thành phần ứng dụng.

- Nó khai báo các quyền mà ứng dụng phải có để truy cập các phần được bảo vệ của API và tương tác với các ứng dụng khác.

- Nó cũng khai báo các quyền mà ứng dụng khác phải có để tương tác với các thành phần của ứng dụng.

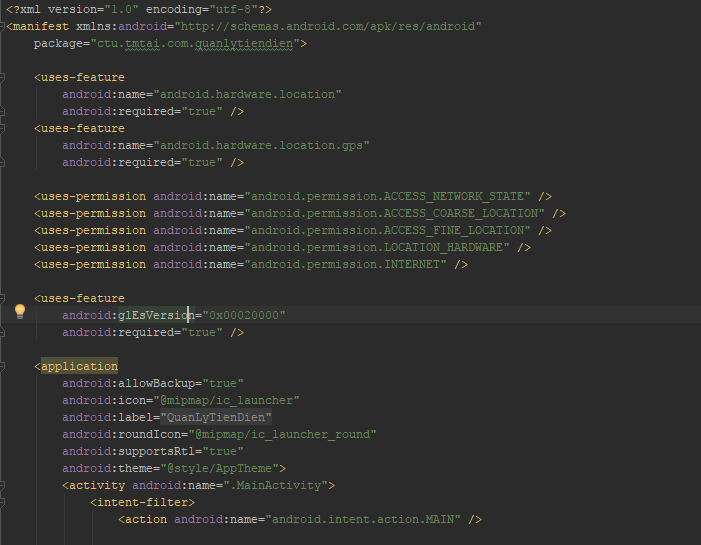
- Nó liệt kê các lớp Instrumentation cung cấp tính năng tạo hồ sơ và các thông tin khác khi ứng dụng đang chạy. Những khai báo này chỉ xuất hiện trong bản kê khai khi ứng dụng đang được phát triển và thử nghiệm; chúng bị loại bỏ trước khi ứng dụng được công bố.

- Nó khai báo mức tối thiểu của API Android mà ứng dụng yêu cầu.

- Nó liệt kê các thư viện mà ứng dụng phải được liên kết với.

Ví dụ như khi ta đặt tên cho ứng dụng, icon ứng dụng thì bạn sẽ vào file này để thiết lập. Nó có rất nhiều thứ để bạn thiết lập ở đây ví dụ như: ứng dụng muốn đọc danh bạ của điện thoại thì android bắt bạn phải xin quyền của người dùng có cho hay không và bạn sẽ thiết lập yều cầu xin quyền ở trong file AndroidManifest này.

Cấu trúc tệp AndroidManifest:

 Bao bọc cả file là thẻ:

Hình 4. Cấu trúc file AndroidManifest.xml

<manifest>

<application>

…

</application>

</manifest>

Đây là 2 thẻ root của file và chỉ xuất hiện một lần duy nhất và phải có nếu không sẽ dẫn đến phát sinh lỗi, các thẻ đóng mở khác sẽ có hoặc không tùy vào việc xây dựng ứng dụng có cần đến hay không.

Một số thẻ khác có thể được sử dụng tùy mục đích của người dùng được mô tả như sau:

- **<uses-permission />***Quyền xin được cấp***<permission />:** đây là cú pháp lệnh để bạn xin cấp một quyền gì đó khi ứng dụng tương tác với dữ liệu nào đó mà google không cho phép dùng tùy ý. Ví dụ như đề cập ở trên, khi ứng dụng muốn đọc danh bạ của một điện thoại nào đó thì nó sẽ phải xin quyền để đọc chứ không thể tự ý đọc được, cú pháp xin quyền như sau:

<uses-permission

android:name="android.permission.INTERNET"/>

Sau khi khai báo như trên ứng dụng sẽ có quyền xem danh bạ trên điện thoại, có rất nhiều quyền, muốn dùng thì phải khai báo giống như trên như: quyền kết nối internet, quyền đọc tin nhắn, quyền kiểm tra kết nối mạng, …

- **<activity>…</activity>:** đây là thẻ để khai báo các activity trong ứng dụng và các thuộc tính của activity đó, nếu như chúng ta không khai bao thì khi khởi chạy ứng dụng sẽ phát sinh lỗi.

- **<service>…</service>:** đây là một dịch vụ chạy ngầm trên ứng dụng kể cả khi ứng dụng đã tắt. Đương nhiên để sử dụng nó thì chúng ta vẫn phải khai báo trong AndroidManifest.

- **<receiver>…</receiver>:** đây là khi báo khi muốn sử dụng Broadcast Reciver, nó dùng để lắng nghe các sự kiện thay đổi của hệ thống như khi bất wifi, khi sạc pin, …

#### 3.2.1.2. file R.java

File R.java là một file được tự động sinh ra ngay sau khi tạo ứng dụng, file này được sử dụng để quản lý các thuộc tính được khai báo trong file XML của ứng dụng và các tài nguyên hình ảnh.

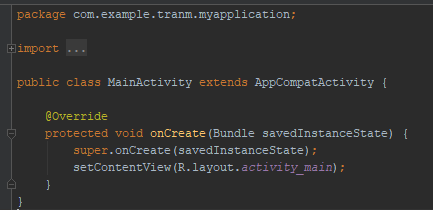
Mã nguồn của file R.java được tự động sinh ra khi có bất kì sự kiện nào xảy ra làm thay đổi các thuộc tính trong ứng dụng. Chẳng hạn khi ta kéo thả một hình ảnh vào project thì ngay lập tức thuộc tính về đường dẫn đến file đó sẽ được hình thành trong R.java, tương tự khi xóa hình ảnh thì đường dẫn trong R.java cũng mất.

Nhìn chung có thể nói chúng ta hoàn toàn không cần đụng chạm đến file này trong quá trình lập trình.

#### 3.2.1.3. MainActivity và activity\_main.xml

Khi một Ứng dụng được tạo ra thì thông thường sẽ có một Activity để khởi chạy ứng dụng. Ở đây MainActivity.java chính là class chứa toàn bộ source code, còn activity\_main.xml chính là phần giao diện (nằm trong thư mục layout). Đối với Android khi một Activity tạo ra thì thường nó đi kèm với một Layout giao diện nào đó (Tức là nó luôn được tách thành 2 phần: phần source code riêng và phần giao diện riêng).

Activity giống như là các màn hình (cửa sổ), mỗi Activity là một màn hình tương tác cụ thể nào đó.

 Ví dụ khi tạo một project với nội dung hello world thì có một file java source code **MainActivity** và file **activity\_main.xml.**

Hình 5. File activity\_main.xml

Hình 6. File MainActivity.java

### 3.2. Vòng đời của một ứng dụng Android

#### 3.2.1. Các loại tiến trình của Android

- ***Foreground process:*** là process của ứng dụng hiện thời đang được người dùng tương tác.

- ***Visible process:*** là process của ứng dụng mà activity đang hiển thị đối với người dùng (onPaused() của activity được gọi).

- ***Service process:*** là Service đang running.

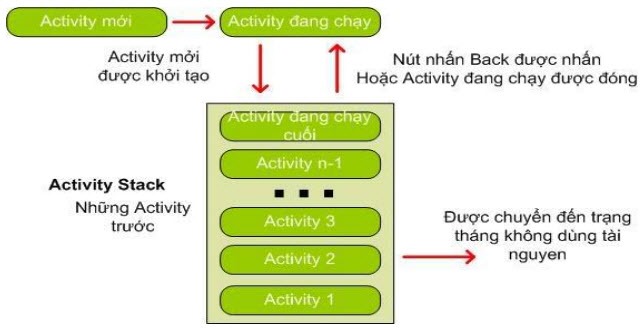
- ***Background process:*** là process của ứng dụng mà các activity của nó ko hiển thị với người dùng (onStoped() của activity được gọi).

- ***Empty process:*** process không có bất cứ 1 thành phần nào active.

Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động kill process, trước tiên là các empty process.

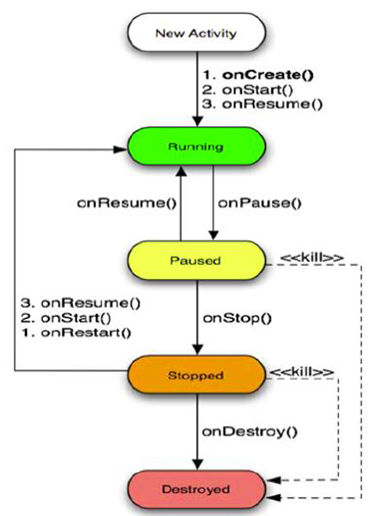
#### 3.2.2. Activity Stack

Bên trong hệ thống các activity được quản lý như một activity stack. Khi một Activity mới được start, nó được đặt ở đỉnh của stack và trở thành activity đang chạy (running activity), các activity trước sẽ tạm dừng và ở bên dưới activity mới và sẽ không thấy trong suốt quá trình activity mới tồn tại.

 Nếu người dùng nhấn nút Back thì activity kết tiếp của stack sẽ di duyển lên và trở thành active.

Hình 7. Activity Stack

#### 3.2.3. Các trạng thái của Activity

- ***active(running):*** Activity đang hiển thị trên màn hình. Activity này tập chung vào những thao tác của người dùng trên ứng dụng.

Hình 8. Các trạng thái của Activity

- ***paused:*** Activity được tạm dừng, tuy vẫn hiển thị (visible) nhưng không thể tương tác (lost focus). Có nghĩa là một activity mới ở trên nó, nhưng không bao phủ đầy màn hình. Một activity tạm dừng là còn sống, nhưng có thể bị kết thúc bởi hệ thống trong trường hợp thiếu vùng nhớ.

- ***stop:*** Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và thường bị loại bỏ khi hệ thống cần vùng nhớ cho các tác vụ khác.

- ***killed:*** Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

#### http://laptrinhandroid.edu.vn/wp-content/uploads/2016/01/29.png 3.2.4. Vòng đời của Activity

Hình 9. Vòng đời của một Activity

- **Entire lifetime:** Từ phương thức onCreate( ) cho tới onDestroy( ).

- **Visible liftetime:** Từ phương thức onStart( ) cho tới onStop( ).

- **Foreground lifetime:** Từ phương thức onResume( ) cho tới onPause( ).

Khi xây dựng Actitvity cho ứng dụng cần phải viết lại phương thức onCreate() để thực hiện quá trình khởi tạo. Các phương thức khác có cần viết lại hay không tùy vào yêu cầu lập trình.

Các hàm thực thi:

- **onCreate():** hàm này được gọi khi lớp Activity được khởi tạo, dùng để thiết lập giao diện ứng dụng và thực thi những thao tác cơ bản.

- **onStart():** hàm này được gọi khi lớp ứng dụng xuất hiện trên màn hình.

- **onResume():** hàm được gọi ngay sau OnStart hoặc khi người dùng focus ứng dụng, hàm này sẽ đưa ứng dụng lên top màn hình.

- **onPause():** hàm được gọi khi hệ thống đang focus đến 1 activity trước đó.

- **onStop():** hàm được gọi khi một activity khác được khởi động và focus.

- **onRestart():** đưọc gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().

# II. Giới thiệu về CSDL mongoDB.



Hình 10. mongoDB

## 1. mongoDB là gì?

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất.

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các document dạng JSON với schema động rất linh hoạt.

MongoDB là một cơ sở dữ liệu NoSQL hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.

## 2. Các lợi thế khi sử dụng mongoDB.

- Ít Schema hơn: mongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.

- Cấu trúc của một đối tượng là rõ ràng.

- Không có các Join phức tạp.

- Khả năng truy vấn sâu hơn. MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.

- Tuning

- MongoDB là dễ dàng để mở rộng.

- Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.

- Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

## 3. Tại sao nên sử dụng mongoDB

- Kho lưu định hướng Document: Dữ liệu được lưu trong các tài liệu kiểu JSON

- Lập chỉ mục trên bất kỳ thuộc tính nào

- Replication và tính khả dụng cao

- Tự động Shard.

- Các truy vấn đa dạng.

- Cập nhật nhanh hơn.

- Sự hỗ trợ chuyên nghiệp bởi MongoDB.

## 4. Nên sử dụng mongoDB ở đâu.

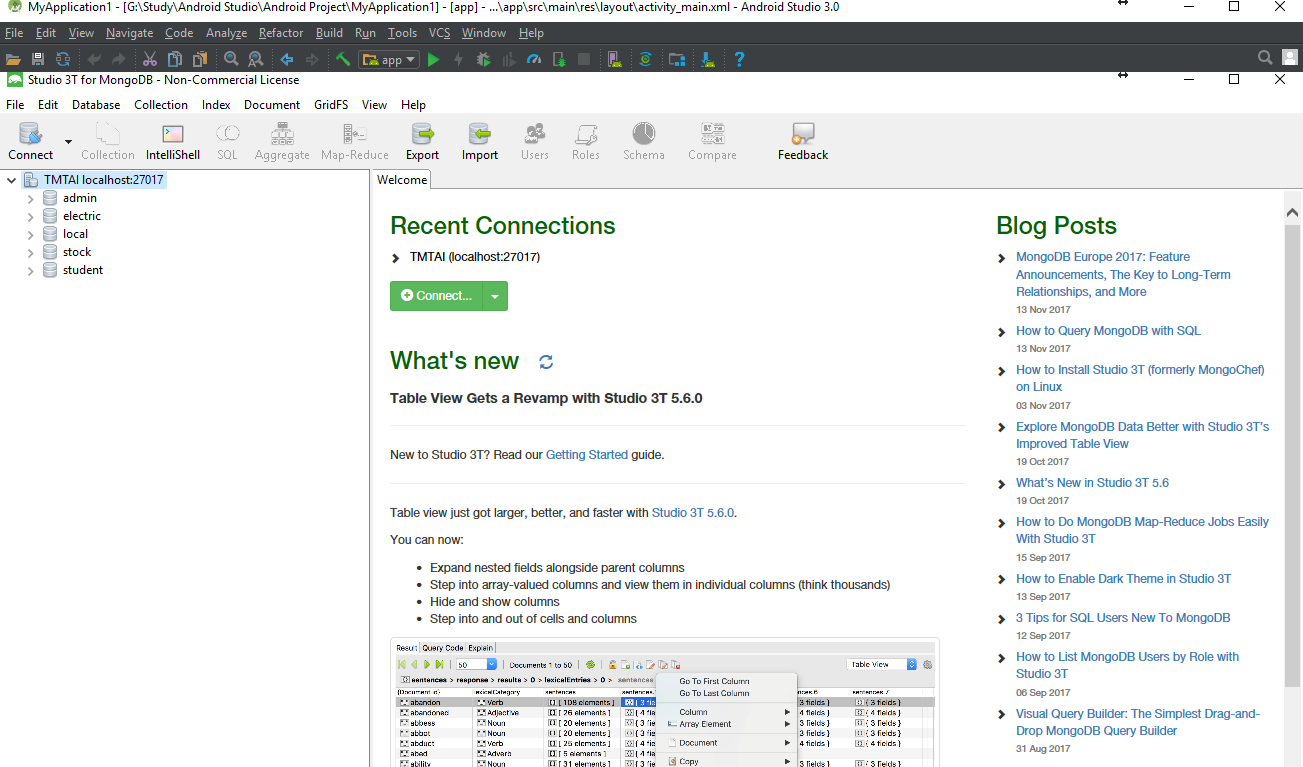
- Lượng dữ liệu lớn.

- Quản lý dữ liệu người dùng.

- Phân phối và quản lý nội dung (CM)

- Data Hub.

## 5. Giao diện mongoDB (Studio 3T for MongoDB)



Hình 11. Giao diện Studio 3T for mongoDB

# II. Giới thiệu về Web service và kiến trúc RESTful.

## 1. Web Service là gì?

- Web Service là một dịch vụ web, nó cung cấp các thông tin thô, và khó hiểu với đa số người dùng, chính vì vậy nó được sử dụng bởi các ứng dụng. Các ứng dụng này sẽ chế biến các dữ liệu thô trước khi trả về cho người dùng cuối cùng.

- Các Web Service thường cung cấp các dữ liệu thô mà nó khó hiểu đối với đa số người dùng thông thường, chúng thường được trả về dưới dạng **XML** hoặc **JSON**.

## 2. Đặc điểm của Web Service

### 2.1. Đặc điểm

- Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau. Ví dụ, đặt Web server cho ứng dụnTg trên một máy chủ chạy hệ điều hành Linux trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể chạy và xử lý bình thường mà không cần thêm yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.

- Phần lớn kĩ thuật của Dịch vụ Web được xây dựng dựa trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.

- Một Dịch vụ Web bao gồm có nhiều mô-đun và có thể công bố lên mạng Internet.

- Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra những lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp khác và cả những cá nhân thông qua mạng Internet.

- Một ứng dụng khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía server ví dụ như PHP, Oracle Application server hay Microsoft.Net…

- Ngày nay dịch vụ Web đang rất phát triển, những lĩnh vực trong cuộc sống có thể áp dụng và tích hợp dịch vụ Web là khá rộng lớn như dịch vụ chọn lọc và phân loại tin tức (hệ thống thư viện có kết nối đến web portal để tìm kiếm các thông tin cần thiết); ứng dụng cho các dịch vụ du lịch (cung cấp giá vé, thông tin về địa điểm…), các đại lý bán hàng qua mạng, thông tin thương mại như giá cả, tỷ giá hối đoái, đấu giá qua mạng…hay dịch vụ giao dịch trực tuyến (cho cả B2B và B2C) như đặt vé máy bay, thông tin thuê xe…

- Các ứng dụng có tích hợp dịch vụ Web đã không còn là xa lạ, đặc biệt trong điều kiện thương mại điện tử đang bùng nổ và phát triển không ngừng cùng với sự lớn mạnh của Internet. Bất kì một lĩnh vực nào trong cuộc sống cũng có thể tích hợp với dịch vụ Web, đây là cách thức kinh doanh và làm việc có hiệu quả bởi thời đại ngày nay là thời đại của truyền thông và trao đổi thông tin qua mạng. Do vậy, việc phát triển và tích hợp các ứng dụng với dịch vụ Web đang được quan tâm phát triển là điều hoàn toàn dễ hiểu.

### 2.2. Ưu điểm và nhược điểm

#### 2.2.1. Ưu điểm

- Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.

- Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.

- Nâng cao khả năng tái sử dụng.

- Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.

- Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.

- Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

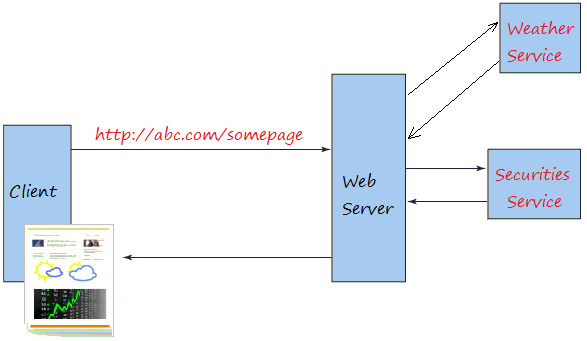
#### 2.2.2. Nhược điểm

- Những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của Dịch vụ Web, giao diện không thay đổi, có thể lỗi nếu một máy khách không được nâng cấp, thiếu các giao thức cho việc vận hành.

- Có quá nhiều chuẩn cho dịch vụ Web khiến người dùng khó nắm bắt.

- Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

### 2.3. Mô hình Web Service



Hình 12. Mô hình Web Service

## 3. Kiến trúc RESTful

#### 3.1 RESUful là gì?

REST là viết tắt của cụm từ Representational State Transfer (đôi khi còn được viết là ReST) là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính (máy tính cá nhân và máy chủ của trang web) trong việc quản lý các tài nguyên trên internet.

REST được sử dụng rất nhiều trong việc phát triển các ứng dụng Web Services sử dụng giao thức HTTP trong giao tiếp thông qua mạng internet. Các ứng dụng sử dụng kiến trúc REST này thì sẽ được gọi là ứng dụng phát triển theo kiểu RESTful.

**RESTful Web Service** là các Web Service được viết dựa trên kiến trúc REST**.**

#### 3.2. Các quy tắt mà REST định nghĩa

- **REST** định nghĩa các quy tắc kiến trúc để bạn thiết kế **Web services.**

- **REST** là một bộ quy tắc để tạo ra một ứng dụng **Web Service**, mà nó tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

- Sử dụng các phương thức HTTP một cách rõ ràng

- Phi trạng thái

- Hiển thị cấu trúc thư mục như các Urls

- Truyền tải **JavaScript Object Notation (JSON)**, **XML** hoặc cả hai.

- **REST** đặt ra một quy tắc đòi hỏi lập trình viên xác định rõ ý định của mình thông qua các phương thức của **HTTP.**

- Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức **POST**.

- Để truy xuất một tài nguyên, sử dụng **GET**.

- Để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó, sử dụng **PUT**.

- Để huỷ bỏ hoặc xoá một tài nguyên, sử dụng **DELETE**.

# Chương 2: Thiết kế và xây dựng ứng dụng

# I. Kế hoạch phát triển và đảm bảo chất lượng.

## 1. Mục đích

- Kế hoạch đảm bảo chất lượng phần mềm nhằm mục đích đảm bảo cho phần mềm phát triển theo đúng yêu cầu đã đề ra. Phần mềm được phát triển trên phạm vi và quy mô nhỏ, thời gian ngắn.

- Quản lý chất lượng phần mềm thương có 2 mục tiêu chính:

- Xây dựng chất lượng phần mềm ngay từ giai đoạn bắt đầu.

- Chất lượng phần mềm sẽ được bảo đảm xuyên suốt trong quá trình phát triển.

## 2. Quản lý

### 2.1. Công việc

#### 2.1.1 Các giai đoạn của chu kỳ sống của phần mềm

- Nghiên cứu sơ bộ (Preliminary Investigation hay còn gọi là Feasibility Study).

- Phân tích yêu cầu (Analysis):

- Thiết lập được những đặc tả yêu cầu cần cho phần mềm theo yêu cầu từ phía người sử dụng, đồng thời làm tài liệu cho thiết kế kiểm thử phần mềm.

- Thiết kế hệ thống (Design of the System):

- Thiết kế phần mềm dựa trên bản đặc tả yêu cầu của phần mềm.

- Thiết kế giao diện thân thiện, hệ thống chạy trơn tru, không phát sinh lỗi.

- Hệ thống dễ dàng cài đặt, bảo trì dễ dàng, nhanh chống.

- Cách tính năng dễ sử dụng.

- Cơ sở dữ liệu lưu với dạng file txt đính kèm.

- Xây dựng phần mềm (Softwave Construction):

- Phần mềm phải được tiền hành sau khi đã xác định được những yêu cầu của hệ thống thông qua bản kế hoạch đặc tả yêu cầu.

- Thử nghiệm hệ thống (System Testing):

- Thực hiện thử nghiệm hệ thống sau khi đã xây dựng phần mềm.

- Thực hiện, triển khai và hoàn thành phần mềm.

#### 2.1.2 Quy trình kiểm soát sự phát triển của phần mềm

Lập dự án

Viết đặc tả

Lập trình

Thiết kế

Kiểm thử

Bàn giao

## 3. Tài liệu

### 3.1. Mục đích

- Các tài liệu chi phối sự phát triển, thẩm tra và công nhận hợp lệ, sử dụng và bảo trì phần mềm:

- Tài liệu đặc tả.

- Tài liệu thiết kế.

- Tài liệu kiểm thử

- Cơ sở dữ liệu.

- Các tài liệu cần được thẩm tra xác định lại:

- Tài liệu đặc tả, phân tích yêu cầu

- Tài liệu thiết kế.

- Tài liệu kiểm thử

### 3.2. Yêu cầu tài liệu tối thiểu:

- Tài liệu đặc tả

- Tài liệu thiết kế

- Tài liệu kiểm thử

#### 3.2.1 Mô tả yêu cầu phần mềm

- Yêu cầu kỹ thuật

- Sử dụng Android Studio, mongoDB, Google Map API, Web Service, …

- Yêu cầu chức năng:

\* Admin:

- Quản lý nhân viên, khách hàng:

- Quản lý danh sách nhân viên, khách hàng.

- Thêm user, cấp lại mật khẩu cho user.

- Tìm kiếm thông tin nhân viên, khách hàng.

- Quản lý thông tin chi trả:

- Xem thông tin tổng quát về tổng các khoản thu của nhân viên.

\* Nhân viên:

- Quản lý thông tin:

- Đăng nhập.

- Xem thông tin cá nhân.

- Cập nhật thông tin cá nhân.

- Thu tiền:

- Tra cứu các hộ trong khu vực thu tiền.

- Đánh dấu các hộ đã thu tiền

- Ghi số điện mới.

#### 3.2.2 Mô tả thiết kế phần mềm

- Thiết kế bao gồm các phần chính: thiết kế hệ thống, thiết kế dữ liệu, thiết kế chức năng và những chi tiết khác của phần mềm.

#### 3.2.3 Các kế hoạch kiểm tra và công nhận hợp lệ

- Kế hoạch thẩm tra cho phần mềm bao gồm:

- Kiểm tra các chức năng của phần mềm, các thiết kế về giao diện (thân thiện, đơn giản, dễ sử dụng).

- Kiểm tra độ tin cậy và đúng đắn của phần mềm.

- Kiểm tra các tài liệu cần để phát triển phần mềm.

→ Từ đó đánh giá xem lại phần mềm có được phát triển đúng theo các yêu cầu đã đề ra hay không.

- Kế hoạch công nhận hợp lệ cho phần mềm:

- Kiểm tra cách chức năng của phần mềm có với các yêu cầu đã dặt ra, nếu đúng thì công nhận hợp lệ cho phần mềm. Ngược lại, phần mềm không đúng yêu cầu thì phải kiểm tra và chỉnh sửa lại.

#### 3.2.4 Báo cáo thẩm tra và công nhận hợp lệ

- Kết quả thẩm tra phần mềm: các tính năng cơ bản đều phát triển đúng theo kế hoạch đã đưa ra:

- Giao diện chương trình thân thiện, đơn giản, dễ sử dụng.

- Cấc tài liệu cần thiết đều được viết đúng yêu cầu đề ra.

- Kết quả công nhận hợp lệ cho phần mềm: phần mềm được phát triển đúng theo các yêu cầu đề ra.

#### 3.2.5 Quản lý cấu hình phần mềm

- Các hoạt động mà kế hoạch quản lý cấu hình nên được hoàn thành là:

- Nhận dạng cấu hình:

- Nhận dạng và đặt tên các thành phần của cấu hình.

- Xác định được các thành phần chính.

- Kiểm soát cấu hình:

- Về yêu cầu các thay đổi.

- Phê chuẩn và không phê chuẩn các thay đổi.

- Thực thi các thay đổi.

- Báo cáo tình trạng cấu hình theo mẫu.

## 4. Các chuẩn, thực tiễn, qui ước và phép đo

### 4.1. Mục đích

- Thiết lập các quy ước chung tạo sự thống nhất trong tiến trình phát triển phần mềm.

- Khi tiến hành phải thực hiện phát triển phần mềm các chuẩn thực tiễn, quy ước và phép đo phải được thông qua sau khi tiến hành các công việc khác.

### 4.2. Nội dung

- Các chuẩn tài liệu: Font chữ Times New Roman, cỡ chữ 13.

- Các chuẩn thiết kế: đơn giản, dễ sử dụng nhưng phải đáp ứng các yêu cầu đã đề ra.

- Các chuẩn lập trình: ngắn gọn, hợp lý.

- Các chuẩn chú thích tài liệu: phải đúng đắng, hợp lý.

- Các chuẩn thực tiễn và kiểm thử.

- Các phép đo quy trình và sản phẩm đảm bảo chất lượng phần mềm được chọn.

## 5. Xem lại phần mềm

### 5.1. Mục đích

- Xem lại phần mềm đã đáp ứng đủ yêu cầu chưa, nếu chưa thì phải sửa chữa lại.

- Các định nghĩa cơ bản:

- Quản lý chất lượng là một thuật nghữ chung bao gồm một số quy trình chất lượng được thực thi xuyên suốt dự án, liên quan đến quá trình lập kế hoạch chất lượng, kiểm soát chất lượng, đảm bảo chất lượng, kiểm định phần mềm.

- Kiểm thử phần mềm nhằm mục đích kiểm tra phầm mềm có phù hợp với các yêu cầu đã đề ra hay chưa, từ đó phát triển và sửa chữa cho hoàn chỉnh.

### 5.2. Các yếu tố tối thiểu

- Phần mềm phải khởi động nhanh, tốn ít tài nguyên máy.

- Dễ dàng bảo trì và cập nhật các yêu cầu mới về phần mềm.

- Thân thiện, dễ sử dụng, được người dùng chấp nhận.

#### 5.2.1 Xem lại các thiết kế kiến trúc

- Xem lại tài liệu thiết kế để đánh giá đúng về kỹ thuật của thiết kế sơ bộ phần mềm.

#### 5.2.2 Xem lại đặc tả phẩn mềm

- Xem lại tài liệu đặc tả phần mềm và kiểm tra phần mềm có được phát triển đúng chưa. Nếu chưa thì cần chỉnh sửa lại cho hợp lý và đúng với các yêu cầu đã đặt ra.

#### 5.2.3 Xem lại chi tiết thiết kế

- Xem lại các tài liệu thiết kế để xác định tính có thể chấp nhận của các thiết kế chi tiết phần mềm.

#### 5.2.4 Xem lại kế hoạch thẩm tra và công nhận hợp lệ

- Xem lại các kế hoạch thẩm tra và công nhận hợp lệ được thực hiện để đánh giá tính đúng đắng và đầy đủ các chức năng của phần mềm.

#### 5.2.5 Kiểm toán chức năng

- Kiểm tra lại tất cả các chức năng chính của phần mềm bao gồm: Thêm và thay đổi tên người dùng, lựa chọn ứng dụng, tra từ điển, ...

- Nếu việc kiểm tra thấy phần mềm chưa đáp ứng yêu cầu đã đặt ra thì phải ngưng việc hoạt động của phần mềm để khắc phục lỗi.

#### 5.2.6 Kiểm toán vật lý

- Kiểm tra lại tất cả các tài liệu quan trọng liên quan đến sự phát triển của phần mềm, sự nhất quán bên trong của phần mềm để tiến hành bàn giao phần mềm.

#### 5.2.7 Xem lại về quản lý

- Kiểm tra lại tất cả các hoạt động và các mục được xác định trong kế hoạch đảm bảo chất lượng phần mềm để có thể điều chỉnh kế hoạch cho phù hợp với sự phát triển của phần mềm.

#### 5.2.8 Xem lại kế hoạch quản lý cấu hình

- Xem lại tài liệu về quản lý cấu hình để đánh giá tính đầy đủ và đúng đắn của các phương pháp quản lý cấu hình được định nghĩa trpng quản lý cấu hình, từ đó xác định các vấn đề và tìm hướng giải quyết.

# II. Đặc tả yêu cầu ứng dụng.

## 1. Mục tiêu

- Tài liệu viết ra nhằm cung cấp thông tin dầy đủ và chi tiết về yêu cầu, tính năng và đặc điểm hệ thống.

- Giúp lập trình viên lập trình sản phẩm đúng theo yêu cầu khách hàng và có thể mở rộng phần mềm.

- Giúp kiểm thử viên dễ dàng hơn trong việc kiểm tra lỗi.

- Giúp người quản lý hiểu thêm yêu cầu khách hàng.

## 2. Phạm vi sản phẩm

- Phần mềm App Quản Lý Tiền Điện giúp người dùng có thể quản lý thông tin của người dùng, hiển thị địa chỉ của khách hang trên bản đồ google-map, lập ra chỉ số chỉ dùng điện mới cho khách hàng, tình tiền, ...

## 3. Tài liệu tham khảo

- Giáo trình Nhập Môn Công nghệ phần mềm của Ths. Phan Phương Lan.

- Báo Cáo đề tài Hệ Thống Quản Lý Điểm Rèn Luyện - PSG.TS Trần Cao Đệ.

## 4. Mô tả tổng quan

### 4.1. Bối cảnh sản phẩm

Với thời đại công nghệ thông tin như hiện nay, việc lưu giữ và trao đổi thông tin qua internet là khá phổ biến. Khi đó, mọi dữ liệu đều có thể lưu giữ trên server và được truy cập ở bất cứ nơi đâu thông qua internet.

Mặc khác, hiện nay smart phone cũng ảnh hưởng một phần không nhỏ vào cuộc sống của hầu hết mọi người.

Với tình hình đó, App quản lý tiền điện được viết ra nhằm mục đích tạo điều kiện thuận lợi cho các nhà cung cấp điện, các nhân viên bên lĩnh vực cung cấp điện thuận tiện hơn trong quá trình lưu giữ thông tin của khách hang, quản lý các chỉ số điện cũng tiện lợi hơn.

App quản lý tiền điện, với một số chức năng như: hiển thị danh sách khách hang theo từng khu vực - thành phố, hiển thị sách danh khách hàng trên bản đồ google-map, chức năng thu tiền, ghi chỉ số điện, … và được cài đặt trên điện thoại smart phone sẽ giúp người dùng thuận tiện sử dụng và quản lý thông tin mọi lúc.

### 4.2. Mô tả bài toán

Một công ty điện lực muốn lưu giữ thông tin của khách hàng trên một server, để có thể truy cập và quản lý thông qua internet. Một nhân viên thu cước phí muốn lấy thông tin các khách hàng về điện thoại và hiện thị các địa chỉ lên bản đồ để tiện cho việc di chuyển thay vì phải ghi nhớ các địa điểm. Ngoài ra, mỗi lần thay đổi thông tin từ điện thoại, thông tin lưu trữ trên server cũng được cập nhật để các quản trị viên của công ty điện lực đó dễ quản lý.

Ứng dụng “quản lý thông tin tiền điện” được tạo ra để có thể quản lý các thông tin bằng điện thoại song song với các web site trên máy tính.

Các thông tin sẽ được lưu giữ trên server, và được gửi xuống các client (điện thoại) khi truy cập vào ứng dụng. Người dùng có thể thao tác với các dữ liệu và đồng bộ lên lại server. Mặc khác, với các địa chỉ của khách hàng sẽ được hiển thị lên bản đồ (google-map), khi đó người dùng sẽ dễ dàng di chuyển đến các địa điểm. Ngoài ra, ứng dụng còn cho biết được trạng thái thanh toán tiền điện của khách hàng ở mỗi tháng.

### 4.3. Chức năng của sản phẩm

- Xác nhận, truy cập người dùng.

- Xem thông tin các nhân của người dùng.

- Hiển thị thông tin khách hàng theo khu vực – thành phố.

- Hiển thị địa chỉ của khách hàng trên google-map.

- Tính tiền điện của người sử dụng điện.

- Hiển thị trạng thái thanh toán.

### 4.4. Các ràng buộc thực thi và thiết kế

- Lập trình bằng ngôn ngữ java, xml.

- Sử dụng công cụ Android Studio, Genymotion, ...

- CSDL: mongoDB.

- Server: Tomcat.

### 4.5. Các giả định phụ thuộc

- Giả định phần mềm chạy tốt trên các thiết bị có hệ điều hành Android 4.0 trở lên.

- Tài liệu có thể tham khảo các tài liệu khác liên quan đến phần mềm đang xây dựng, có thể sử dụng lại hoặc thay đổi cho phù hợp.

# III. Thiết hệ thống, CSDL, thiết kế ứng dụng, mô tả hoạt động của từng chức năng.

## 1. Thiết kế kiến trúc

Sơ đồ 1. Sơ đồ kiến trúc

**Người dùng**

**ứng dụng**

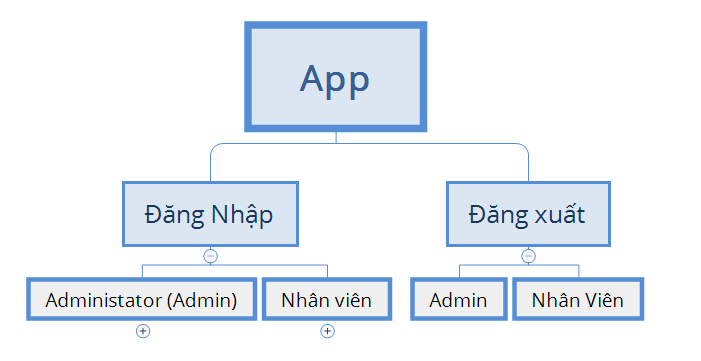
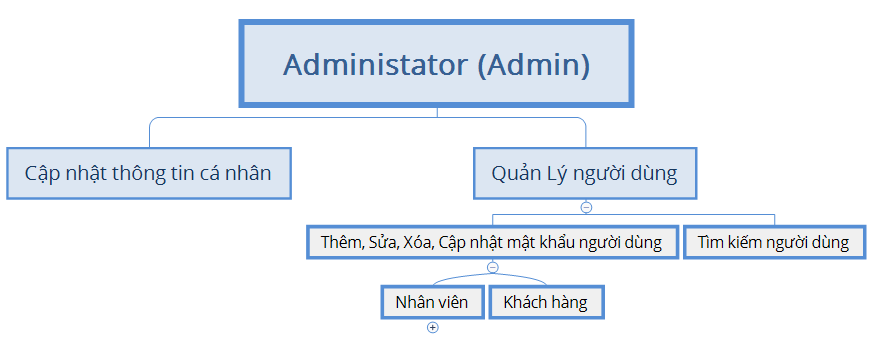
**Server**

Sử dụng

Gửi yêu cầu

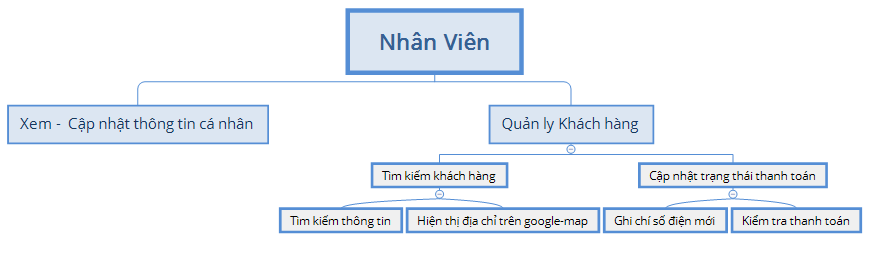
Thông tin đã xử lý

Thông tin đã xử lý

 - Mô hình phân rã

Mô hình 1. Mô hình phân rã

Mô hình 2. Mô hình phân ra administator

2. Thiết kế CSDL

Mô hình 3. Mô hình phân rã nhân viên

### C:\Users\tranm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\mohinh.png 2.1. Mô tả dữ liệu thiết kế

Mô hình 4. Mô hình quan hệ cơ sở dữ liệu

### 

### 2.2 Các bảng CSDL

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng | Diễn giải |
| 1 | USER | Bảng dữ liệu nhân viên |
| 2 | KHACH\_HANG | Bảng dữ liệu khách hàng |
| 3 | DIEN\_KE | Bảng dữ liệu điện kế của khách hàng |
| 4 | KHU\_VUC | Bảng dữ liệu các khu vực |
| 5 | TINH\_THANHPHO | Bảng dữ liệu các thành phố |
| 6 | TIME | Bảng dữ liệu thời gian (tháng/năm) |
| 7 | THANG | Bảng dữ liệu các tháng (12 tháng) |
| 8 | NAM | Bảng dữ liệu các năm |
| 9 | HOA\_DON | Bảng dữ liệu hóa đơn. |
| 10 | CHI\_TIET\_HOA\_DON | Bảng dữ liệu thông tin chi tiết của hóa đơn. |

Bảng 1. Danh sách các bảng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | code | Text |  | Not null | <pk> |  | Mã của nhân viên |
| 2 | ten | Text |  | Not null |  |  | Tên nhân viên |
| 3 | diachi | Text |  | Not null |  |  | Địa chỉ |
| 4 | ngaysinh | Date |  | Not null |  |  | Ngày sinh |
| 5 | cmnd | Text |  | Not null |  |  | CMND |
| 6 | gioitinh | Text |  | Not null |  |  | Giới tính |
| 7 | phone | Text |  | Not null |  |  | Điện thoại |
| 8 | isAdmin | Boolean |  | Not null |  |  | Quyền |
| 9 | password | Text |  | Not null |  |  | Mật khẩu đăng nhập |

Bảng 2. Bảng USER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | makh | Text |  | Not null | <pk> |  | Mã của nhân viên |
| 2 | ten | Text |  | Not null |  |  | Tên nhân viên |
| 3 | diachi | Text |  | Not null |  |  | Địa chỉ |
| 4 | ngaysinh | Date |  | Not null |  |  | Ngày sinh |
| 5 | cmnd | Text |  | Not null |  |  | CMND |
| 6 | gioitinh | Text |  | Not null |  |  | Giới tính |
| 7 | phone | Text |  | Not null |  |  | Điện thoại |
| 8 | role | Boolean |  | Not null |  |  | Quyền |
| 9 | madk | Text |  | Not null |  | <fk> | Mã điện kế |
| 10 | tenkv | Text |  | Not null |  | <fk> | Mã khu vực |

Bảng 3. Bảng KHACH\_HANG

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | madk | Text |  | Not null | <pk> |  | Mã của điện kế |
| 2 | ngaydksd | Text |  | Not null |  |  | Ngày đăng ký sử dụng |
| 3 | chisocu | Text |  | Not null |  |  | Chỉ số cũ |
| 4 | chisomoi | Date |  | Not null |  |  | Chỉ số mới |
| 5 | isthanhtoan | Boolean |  | Not null |  |  | Trạng thái thanh toán tiền |
| 6 | makh | Text |  | Not null |  | <fk> | Mã khách hàng |
| 7 | matime | Text |  | Not null |  | <fk> | Thời gian (tháng/năm) |

Bảng 4. Bảng DIEN\_KE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | makv | Text |  | Not null | <pk> |  | Mã của khu vực |
| 2 | tenkv | Text |  | Not null |  |  | Tên khu vực |
| 3 | matp | Text |  | Not null |  | <fk> | Mã thành phố |

Bảng 5. Bảng KHU\_VUC

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | matp | Text |  | Not null | <pk> |  | Mã của thành phố |
| 2 | tentp | Text |  | Not null |  |  | Tên thành phố |

Bảng 6. Bảng TINH\_THANHPHO

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | mathang | Text |  | Not null | <pk> |  | Mã của tháng |

Bảng 7. Bảng THANG

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | manam | Text |  | Not null | <pk> |  | Mã của năm |

Bảng 8. Bảng NAM

.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | matime | Text |  | Not null | <pk> |  | Mã thời gian |
| 2 | mathang | Text |  | Not null |  | <fk> | Mã tháng |
| 3 | manam | Text |  | Not null |  | <fk> | Mã năm |

Bảng 9. Bảng TIME

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | mahd | Text |  | Not null | <pk> |  | Mã của hóa đơn |
| 2 | mathue | Text |  | Not null |  |  | Mã tháng |
| 3 | madk | Text |  | Not null |  | <fk> | Mã điện kế |

Bảng 10. Bảng HOA\_DON

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu | Miền giá trị | Chấp nhận NULL | Khóa chính | Khóa ngoại | Mô tả |
| 1 | machitiet | Text |  | Not null | <pk> |  | Mã của chi tiết hóa đơn |
| 2 | tenkh | Text |  | Not null |  |  | Tên của khách hàng |
| 3 | diachi | Text |  | Not null |  |  | Địa chỉ khách hàng |
| 4 | VAT | Integer |  | Not null |  |  | Thuế VAT |
| 5 | tungay | Date |  | Not null |  |  | Tính từ ngày |
| 6 | denngay | Date |  | Not null |  |  | Tính đến ngày |
| 7 | tieuthu | Integet |  | Not null |  |  | Chỉ số tiêu thụ |
| 8 | thanhtien | Integet |  | Not null |  |  | Số tiền phải trả |
| 9 | mahoadon | Text |  | Not null |  | <fk> | Mã hóa đơn |

Bảng 11. Bảng CHI\_TIET\_HOA\_DON

### 2.3. Sơ đồ xử lý dữ liệu



DATABASE

## 3. Thiết kế ứng dụng

Sơ đồ 2. Sơ đồ xử lý dữ liệu

### C:\Users\tranm\AppData\Local\Microsoft\Windows\INetCache\Content.Word\usecase.jpg 3.1. Sơ đồ use-case

Sơ đồ 3. Sơ đồ use-case

### 3.2. Xây dựng chức năng

- Các chức năng của ứng dụng:

- Đăng nhập.

- Xem thông tin cá nhân.

- Người dùng đổi mật khẩu.

- Tìm kiếm thông tin.

- Thêm người dùng, xóa người dùng.

- Cập nhật mật khẩu cho người dùng.

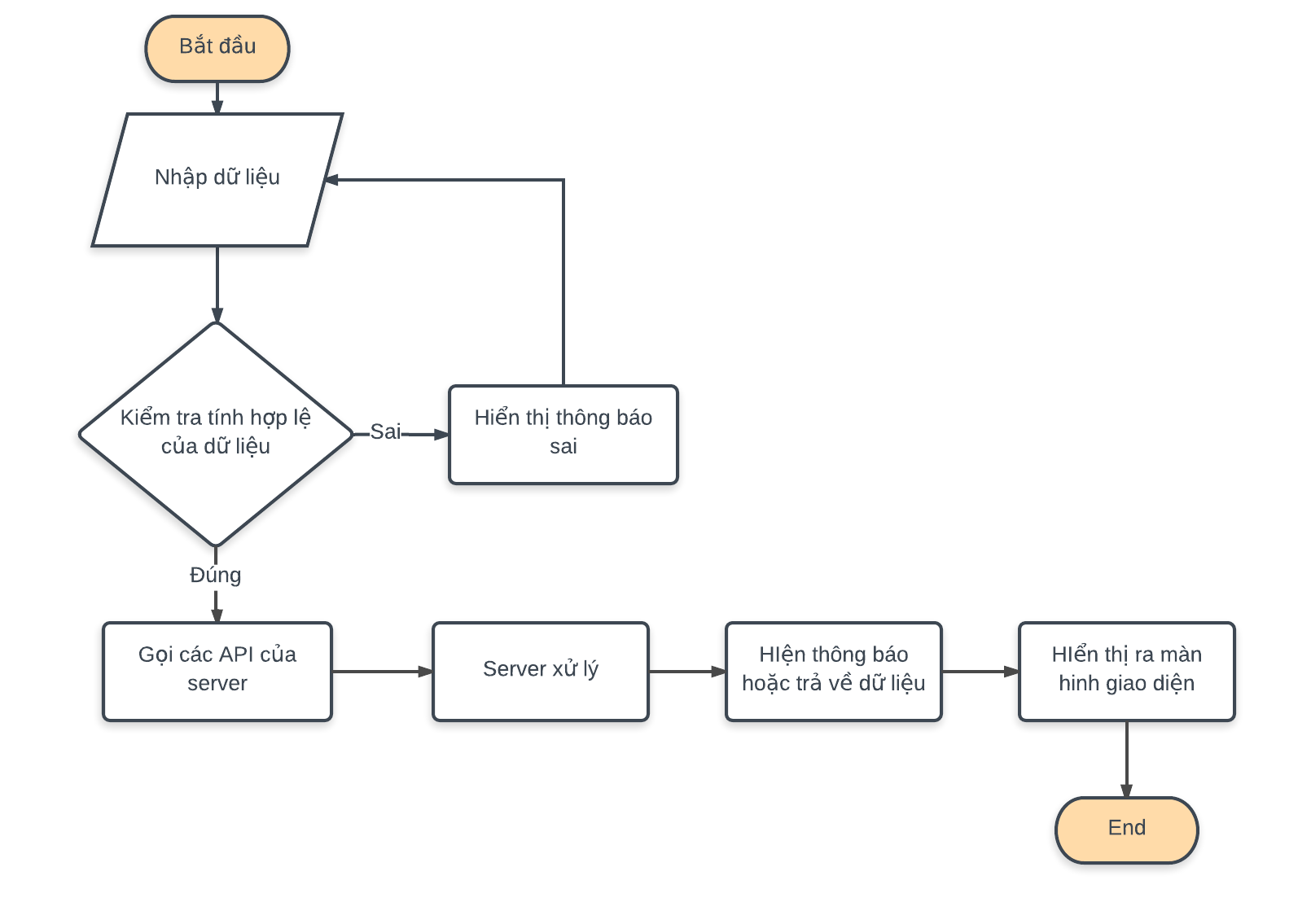
- Cập nhật trạng thái thanh toán tiền của khách hàng.

- Cập nhật chỉ số của điện kế.

- Hiển thị danh sách khách hàng theo khu vực.

- Hiển thị địa chỉ khách hàng trên bản đồ.

## 4. Thiết kế giao diện theo chức năng

\* Sơ đồ xử lý chung: 

Sơ đồ 4. Sơ đồ xử lý dữ liệu

### 4.1. Đăng nhập

Hình 13. Đăng nhập

#### 4.1.1. Mô tả và mức độ ưu tiên

- Mô tả: chức năng *Đăng nhập* cho phép người dùng đăng nhập với tài khoản của mình để vào sử dụng các chức năng với quyền của tài khoản (Admin - user).

- Mức độ ưu tiên: Cao.

#### 4.1.2. Tác nhân và chuỗi đáp ứng.

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng truy cập vào sự kiện.  2. Nhập thông tin username và password.  3. Nhấn xác nhận để đăng nhập.  4. Hệ thống kiểm tra từ trong kho CSLD.  5. Chuyển đến giao diện chính của tài khoản.  6. Kết thúc tìm kiếm. |
| Dòng thay thế | Dòng thay thế bắt đầu từ bước 4, không đúng username hoặc password:  7. Hệ thống thông báo sai username or password.  8. Đưa người dùng quay lại bước 2 của dòng sự kiện chính |
| Dòng sự kiện lỗi | 9. Khi không có kết nối internet, hệ thống sẽ không thể đăng nhập và đưa ra thông báo sai username or password. |

#### 4.1.3. Các yêu cầu chức năng

- REQ-1: Người dùng không cần xác nhận chính xác username và password, cần kết nối internet để có thể đăng nhập.

### 4.2. Xem thông tin

Hình 14. Xem thông tin cá nhân

#### 4.2.1. Mô tả và mức ưu tiên

- Mô tả: Chức năng *Xem thông tin* cho phép người dùng có thể xem lại thông tin các nhân của mình.

- Mức ưu tiên: Trung bình.

#### 4.2.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng truy cập vào sự kiện.  2. Hệ thống hiển ra thông tin cá nhân của tài khoản.  3. Kết thúc. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.2.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

### 4.3. Đổi mật khẩu

Hình 15. Đổi mật khẩu cá nhân

#### 4.3.1. Mô tả vầ mức độ ưu tiên

- Mô tả: Chức năng *Đổi mật khẩu* cho phép người dùng thay đổi mật khẩu đăng nhập của tài khoản.

- Mức ưu tiên: Trung bình.

#### 4.3.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1.Hệ thống hiển thị giao diện cập nhật mật khẩu.  2. Người dùng nhập mật khẩu cũ, mật khẩu mới và xác nhận để hoàn thành thay đổi.  3. Xác nhận để thay đổi mật khẩu.  4. Hệ thống kiểm tra mật khẩu cũ với CSDL.  5. Thông báo thay đổi mật khẩu thành công.  6. Kết thúc và trở lại giao diện ban đầu. |
| Dòng thay thế | Dòng thay thế sẽ diễn ra ở bước 3. Nếu mật khẩu cũ không khớp với CSDL or mật khẩu mới không trùng với mật khẩu xác nhận.  6. Hệ thống sẽ đưa ra thông báo sai mật khẩu. |
| Dòng sự kiện lỗi |  |

#### 4.3.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

### 4.4. Tìm kiếm thông tin

Hình 16. Tìm kiếm khách hàng

Hình 17. Tìm kiếm nhân viên

#### 4.4.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Tìm kiếm thông tin* cho phép amin, người dùng tìm kiếm thông tin của user hoặc khách hàng dựa vào một số từ khóa: tên, số điện thoại, địa chỉ, cmnd, …

- Mức ưu tiên: Trung bình.

#### 4.4.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Admin.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng nhập từ khóa cần tìm.  2. Hệ thống hiển thị ra thông tin của những người có từ khóa chính xác hoặc gần chính xác nhất. |
| Dòng thay thế | Dòng thay thế bắt đầu từ bước 2, nếu từ khóa không khớp:  3. Hệ thống trả ra kết quả trống |
| Dòng sự kiện lỗi |  |

#### 4.4.3. Các ràng buộc chức năng

- REQ-1: Người dùng với quyền là admin.

### 4.5. Thêm người dùng

Hình 18. Thêm khách hàng

Hình 19. Thêm nhân viên

#### 4.5.1 Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Thêm người dùng* cho phép admin thêm một người dùng mới / khác hàng mới vào CSDL.

- Mức ưu tiên: Trung bình.

#### 4.5.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Admin.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Admin truy cập vào chuỗi sự kiện.  2. Nhập thông tin người dùng cần thêm vào.  3. Xác nhận.  4. Hệ thống thêm người dùng vào CSDL và hiển thị thông báo thêm thành công.  5. Trờ về giao diện ban đầu. |
| Dòng thay thế | Diễn ra từ bước 3, nếu thông tin người dùng đã tồn tại:  6. Hệ thống hiển thị ra thông báo : Người dùng đã tồn tại.  7. Trở về bước 2. |
| Dòng sự kiện lỗi |  |

#### 4.5.3. Các ràng buộc chức năng

- REQ-1: Người dùng với quyền là admin.

Hình 20. Xóa người dùng

### 4.6. Xóa Người dùng

Hình 21. Xác nhận xóa người dùng

#### 4.6.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Xóa người dùng* cho phép admin xóa một tài khoản ra khỏi CSDL.

- Mức ưu tiên: Thấp.

#### 4.6.2. Tác nhân và chuỗi đáp ứng

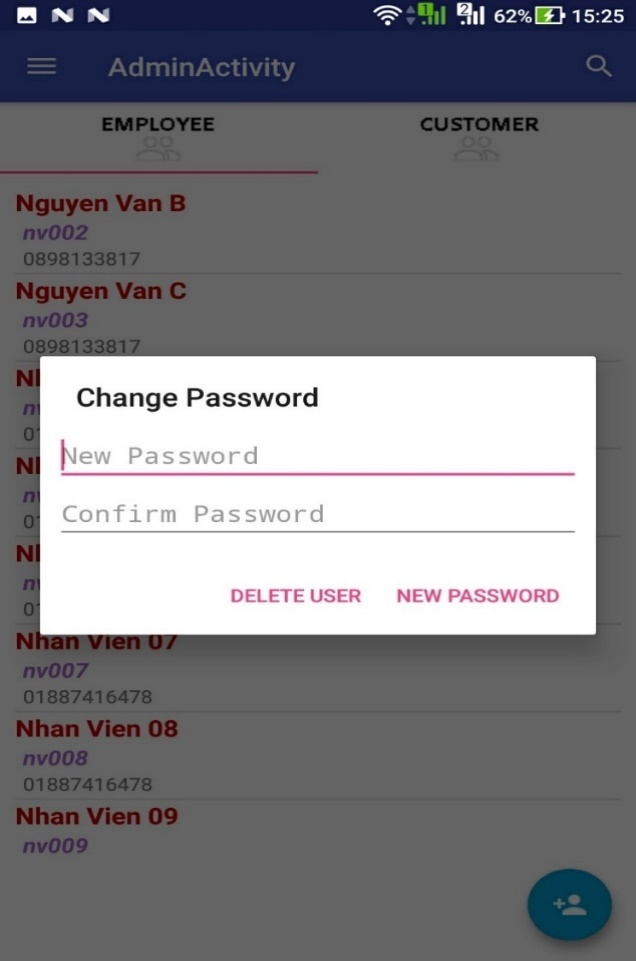
- Các tác nhân chính: Admin.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Admin truy cập vào chuỗi sự kiện.  2. Lựa chọn tài khoản cần xóa.  3. Xác nhận.  4. Hệ thống sẽ xóa tài khoản ra khỏi CSDL. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.6.3. Các ràng buộc chức năng

- REQ-1: Người dùng với quyền là admin.

### 4.7. Cập nhật mật khẩu cho người dùng



Hình 22. Cập nhật mật khẩu cho người dùng

#### 4.7.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Cập nhật mật khẩu cho người dùng* cho phép admin có thể cập nhật lại mật khẩu của người dùng.

- Mức ưu tiên: Thấp.

#### 4.7.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Admin.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Admin truy cập vào chuỗi sự kiện.  2. Lựa chọn tài khoản cần cấp lại mật khẩu.  3. Hệ thống chuyển đến giao diện cập nhật mật khẩu.  4. Nhập password (mật khẩu) mới, và xác nhận.  5. Hệ thống cập nhật lại CSDL. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.7.3. Các ràng buộc chức năng

- REQ-1: Người dùng với quyền là admin.

### 4.8. Cập nhật trạng thái thanh toán tiền của khách hàng.

Hình 23. Trang thái chưa thanh toán

Hình 24. Trạng thái đã thanh toán

#### 4.8.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *CapNhatTrangThaiKhachHang* cho phép người dùng có thể thay đổi trạng thái thanh toán của khách hàng.

- Mức ưu tiên: Cao.

#### 4.8.2 Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Truy cập vào chuỗi sự kiện.  2. Lựa chọn Khách hàng cần cập nhật.  3. Lựa chọn mục thay đổi để cập nhật trạng thái.  4. Hệ thống sẽ cập nhật lại CSDL và hiển thị thông báo.  5. Kết thúc. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.8.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

### 4.9. Cập nhật chỉ số của điện kế.

Hình 25. Cập nhật chỉ số điện kế

#### 4.9.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Cập nhật chỉ số của điện kế* ghi thêm chỉ số mới vào điện kế mà khách hàng sử dụng, để có thể cập nhật thông tin hoặc tính tiền.

- Mức ưu tiên: Cao.

#### 4.9.2. Tác nhân và chuỗi đáp ứng

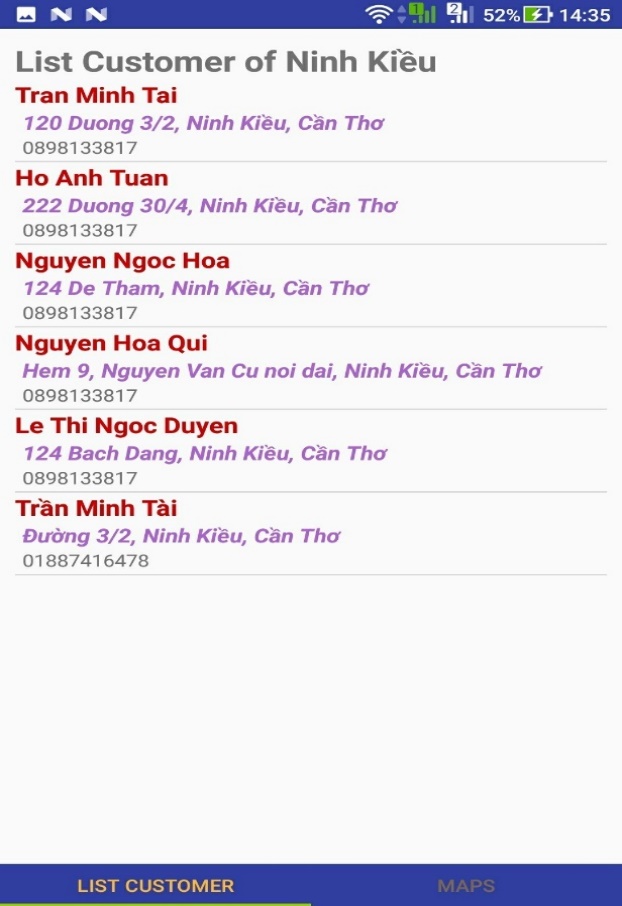
- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng truy cập vào chuỗi sự kiện.  2. Lựa chọn người dùng và điện kế để thay đổi thông tin.  3. Nhập chỉ số mới và xác nhận.  4. Hệ thống cập nhật lại CSDL.  5. Trò chơi kết thúc. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.9.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

### 4.10. Hiển thị danh sách khách hàng theo khu vực.



Hình 26. Hiển thị danh sách khách hàng theo khu vực

#### 4.10.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Hiển thị danh sách khách hàng theo khu vực* để hiển thị danh sách các khách hàng theo từng khu vực để dễ dàng quản lý.

- Mức ưu tiên: Cao.

#### 4.10.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng truy cập vào chuỗi sự kiện.  2. Lựa chọn thành phố và khu vực  3. Hệ thống truy cập CSDL và đưa ra danh sách khách hàng.  4. Kết thúc. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.10.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

### 4.11. Hiển thị địa chỉ khách hàng trên bản đồ.

Hình 27. Hiển thị địa chỉ khách hàng trên bản đồ

#### 4.11.1. Mô tả và mức độ ưu tiên

- Mô tả: Chức năng *Hiển thị địa chỉ khách hàng trên bản đồ* để hiển thị danh sách các địa chỉ của khách hàng trên bản đồ.

- Mức ưu tiên: Cao.

#### 4.11.2. Tác nhân và chuỗi đáp ứng

- Các tác nhân chính: Người dùng.

|  |  |
| --- | --- |
| Dòng sự kiện chính | 1. Người dùng truy cập vào chuỗi sự kiện.  2. Lựa chọn thành phố và khu vực.  3. Hệ thông đưa ra thông tin trên bản đồ.  4. Kết thúc. |
| Dòng thay thế |  |
| Dòng sự kiện lỗi |  |

#### 4.11.3. Các ràng buộc chức năng

- REQ-1: Người dùng cần đăng nhập.

## 5. Yêu cầu phi chức năng

### 5.1. Yêu cầu thực thi

- Thời gian load dữ liệu không quá 15 giây.

- Thời gian thực thi của các chương trình phải đảm bảo, không chiếm quá nhiều dung lượng của máy.

### 5.2. Yêu cầu an toàn

- CSLD phải đảm bảo an toàn, chính xác.

- Chương trình thực thi phải an toàn, tránh có những phần mềm độc hại cài xen vào có thể làm hư tổn phần cứng hoặc phần mềm của máy.

### 5.3. Yêu cầu bảo mật

- Khi truy xuất hoặc lưu thông tin hoặc lưu thông tin vào CSLD phải đảm bảo, cần có biện pháp an toàn dữ liệu.

### 5.4. Các đặc điểm chất lượng phần mềm

- Phần mềm có thể chạy tốt trên các thiết bị smartphone HĐH Android 4.0 trở lên hiện nay.

- Giao diện thân thiện với người dùng, màu sắc hài hòa, font chữ dễ nhìn rõ ràng dễ đọc, bố cục thông tin hợp lý, ...

- Phải chạy được trên một số thiết bị, máy ảo trên máy tính hiện nay.

- Không tốn kém quá nhiều bộ nhớ, dung lượng cứng của máy.

# Chương 3: Kiểm thử và đánh giá.

# I. Kiểm thử

## 1. Mục tiêu

- Kế hoạch kiểm thử nhầm xác định các lỗi (có xảy ra hay không) của chương trình.

- Bao gồm các hoạt dộng đảm bảo rằng phần mềm đã thực hiện đúng chức năng được đặc tả và yêu cầu của người sử dụng.

- Đảm bảo tính hoàn thiện của phần mềm trước khi bàn giao sản phẩm cho khác hàng.

- Nhằm xác minh và thẩm định các hoạt động đúng với yêu cầu đã đề ra.

## 2. Môi trường kiểm thử

- Phầm mềm: Phiên bản Android 7.1.1 (ASUS Zenphone 3 Max).

- Phần cứng:

- Tên thiết bị: ASUS Zenphone 3 Max.

- Ram 3,00 GB

- CPU 1.4Hz.

- Màn hình: độ phân giải 1920 x 1080px.

- Phiên bản GPS: WTR4905-1-60WLNSP.

3. Kiểm thử chức năng

- Các chức năng sẽ được kiểm thử:

- Đăng nhập, đăng xuất

- Tìm kiếm thông tin

- Hiển thị danh sách khách hàng theo khu vực.

- Hiển thị các địa chỉ lên bản đồ.

- Thay đổi trạng thái thanh toán.

- Các chức năng không được kiểm thử

- Thêm người dùng.

- Xoá người dùng.

- Cập nhật thông tin, cập nhật mật khẩu, cấp lại mật khẩu.

- Một số lỗi do đường truyền internet gây ra.

4. Cách tiếp cận

Kiểm thử theo cách từng chức năng, mỗi chức năng được kiểm thử với các kịch bản kiểm thử và ghi kết quả thử nghiệm.

## 5. Tiêu chí thử thành công/thất bại

- Tiêu chí thử thành công khi kết quả kiểm thử giống với kết quả mong muốn trước đó.

- Tiêu chí thử thất bại khi có kết quả thực tế của kiểm thử khác với kết quả mong muốn hoặc không có kết quả.

## 6. Kiểm thử

### 6.1. Kiểm thử thành công.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Chức năng | Test case | Môi trường | Kết quả |
| 1 | Đăng nhập | Username/password  1. admin/111111  2. nv002/123456 | Ứng dụng | Thành công |
| 2 | Đăng xuất. |  | Ứng dụng | Thành công |
| 3 | Tìm kiếm thông tin |  | Ứng dụng | Thành công |
| 4 | Hiển thị danh sách khu vực – khách hàng theo từng khu vực. | 1. Hiển thị danh sách người dùng ở Ninh Kiều – Cần Thơ  2. Hiển thị danh sách người dùng ở Phú Nhuận – TP. HCM | Ứng dụng | Thành công |
| 5 | Hiển thị các địa chỉ của khách hàng lên bản đồ google-map. | Hiển thị danh sách khách hàng ở khu vực Phú Nhuận – TP. HCM trên bản đồ | Ứng dụng | Thành công |
| 6 | Nhập chỉ số mới. |  | Ứng dụng | Thành công |
| 7 | Tình trạng thanh toán. | Thay đổi tình trạng từ: chưa thanh toán sang đã thanh toán | Ứng dụng | Thành công |
| 8 | Tính số tiền người sử dụng phải trả. | Chỉ số cũ: 500  Chỉ số mới: 520  Thành tiền: 2,986 VND | Ứng dụng | Thành công |

### 6.2. Kiểm thử thất bại

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Chức năng | Test case | Môi trường | Kết quả |
| 1 | Đăng nhập | Username/password  1. admin/1200114  2. nv002/ppoiiyhn | Ứng dụng | Thành công |

# PHẦN KẾT LUẬN

# I. Kết quả đạt được

- Hiểu được mô hình client – server, server xử lý CSDL và client hiển thị dữ liệu.

- Vận dụng ngôn ngữ JAVA vào lập trình ứng dụng một cách thực tiễn.

- Hiểu thêm về web service và vận dụng được các API.

- Bổ sung nhiều kiến thức về CSDL, JSON, XML, …

- Nâng cao khả năng phân tích, thiết kế hệ thống thông tin, khắc phục một vài điểm thiếu xót của bản thân.

- Phát huy tinh thần sáng tạo, học hỏi và giải quyết các vấn đề mới.

- Có thêm kinh nghiệm làm việc hơn với các dự án thực tế.

# II. Ưu điểm

- Việc quản lý thông qua smart-phone rất tiện lợi, có thể quản lý mọi lúc mọi nơi chỉ cần kết nối Internet.

- Ứng dụng và server riêng biệt nên đảm bảo sự an toàn của dữ liệu.

- Giao diện thân thiện với người dùng.

# III. Hạn chế

- Chức năng còn nhiều hạn chế trong thực tế.

- Việc kết nối giữa ứng dụng và server còn một số vấn đề chưa thể khắc phục nên đôi lúc ứng dụng có thể bị lỗi kết nối.

- CSDL chưa đúng với thực tế.

- Hạn chế với một số loại điện thoại cũ chưa tích hợp google-map.

- Ứng dụng chưa tối ưu hoàn toàn, một vài chức năng còn chưa hoạt động hoặc hoạt động chưa đúng.

- Hiệu suất hoạt động của ứng dụng còn chưa tốt.

# IV. Hướng phát triển

- Cải thiện hiệu suất xử lý của ứng dụng.

- Cải thiện các chức năng chưa đúng hoặc chưa hoạt động.

- Thêm chức năng thống kê, đặt lịch làm việc cho người sử dụng.

# Tài liệu tham khảo

*[1]. Giáo trình Nhập Môn Công nghệ phần mềm của Ths. Phan Phương Lan.*

*[2]. Báo Cáo đề tài Hệ Thống Quản Lý Điểm Rèn Luyện - GVHD: PSG.TS Trần Cao Đệ.*

*[3].* [*https://developer.android.com/studio/intro/index.html*](https://developer.android.com/studio/intro/index.html)*.*

*[4].* [*http://www.ninjaframework.org*](http://www.ninjaframework.org)

*[5].* [*https://docs.mongodb.com/*](https://docs.mongodb.com/)