**LỜI CAM ĐOAN**

🙘🙛⁂🙙🙚

Tôi xin cam đoan đây là công trình nghiên cứu khoa học độc lập của riêng tôi, số liệu và kết quả nghiên cứu trong luận văn thu được do bản thân trực tiếp theo dõi thu thập với thái độ khách quan và trung thực. Các tài liệu sử dụng phân tích trong luận văn có nguồn gốc rõ ràng, đã công bố theo quy định. Các kết quả nghiên cứu trong luận văn do tôi tự tìm hiểu, phân tích một cách trung thực, khách quan và phù hợp với thực tiễn của Việt Nam. Các kết quả này chưa từng được công bố trong bất kỳ nghiên cứu nào khác.

Cần Thơ, Ngày…Tháng…Năm…

**Sinh viên thực hiện**

**LỜI CẢM ƠN**

🙘🙛⁂🙙🙚

Để hoàn thành khóa luận này, em xin tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Minh Kỳ và thầy Hồng Thanh Luận, đã tận tình hướng dẫn trong suốt quá trình viết Báo cáo tốt nghiệp.

Em chân thành cảm ơn quý thầy, cô trong khoa Công nghệ thông tin, Trường Đại học Kỹ thuật công nghệ Cần Thơ đã tận tình truyền đạt kiến thức trong những năm em học tập. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu khóa luận mà còn là hành trang quí báu để em bước vào đời một cách vững chắc và tự tin.

Cảm ơn Thầy Huỳnh Xuân Hiệp đã tạo điều kiện về không gian làm việc, thiết bị cũng như Thầy đã có nhiều hướng dẫn quý báu…

Cuối cùng em kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Trân trọng kính chào.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

**NHẬN XÉT CỦA HỘI ĐỒNG CHẤM ĐIỂM LUẬN VĂN**

**MỤC LỤC**

KÍ HIỆU VÀ VIẾT TẮT 9

DANH SÁCH HÌNH 10

DANH MỤC BẢNG 11

TÓM TẮT 12

ABSTRACT 13

TỪ KHÓA 14

CHƯƠNG 1 - TỔNG QUAN 15

1.1 Đặt vấn đề 15

1.2 Lịch sử giải quyết vấn đề 15

1.3 Phạm vi đề tài 16

1.4 Phương pháp nghiên cứu 16

1.4.1 Mục tiêu nghiên cứu 16

1.4.2 Đối tượng nghiên cứu 17

1.4.3 Phạm vi nghiên cứu 17

CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT 18

2.1 Tìm hiểu về nền tảng Android[1] 18

2.2 Tìm hiểu về GraphQL [2] 19

2.3 Tìm hiểu về Postgraphile [3][4] 21

2.4 Tìm hiểu về PostgreSQL [5] 21

2.5 Tìm hiểu về JSON Web Token [6] 22

2.6 Tìm hiểu về ReactJS [7] 22

2.7 Tìm hiểu về Apollo Client [8] 23

CHƯƠNG 3 - NỘI DUNG NGHIÊN CỨU 25

3.1 Mô tả bài toán 25

3.1.1 Bối cảnh hệ thống 25

3.1.2 Các chức năng hệ thống 25

3.1.3 Đặc điểm người dùng 26

3.1.4 Môi trường vận hành 26

3.2 Đặc tả yêu cầu 27

3.2.1 Yêu cầu chức năng 27

3.2.1.1 Quản lí trạng thái đơn hàng 27

3.2.1.2 Quản lí trạng thái biên nhận 28

3.2.1.3 Quản lí phân công xử lí đơn hàng 29

3.2.1.4 Tạo đơn hàng 30

3.2.1.5 Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu 31

3.2.1.6 Tìm kiếm và lọc quần áo theo loại có sẵn 32

3.2.1.7 Tìm kiếm đơn hàng 33

3.2.1.8 Đăng nhập hệ thống 33

3.2.1.9 Đăng xuất hệ thống 34

3.2.1.10 Đăng kí tài khoản khách hàng 34

3.2.2 Yêu cầu phi chức năng 36

3.2.3 Yêu cầu thực thi 36

3.2.4 Yêu cầu chất lượng phần mềm 36

3.2.4.1 Các quy tắc nghiệp vụ 36

3.3 Thiết kế và cài đặt 36

3.3.1 Kiến trúc hệ thống 36

3.3.2 Sơ đồ USE CASE 37

3.3.3 Sơ đồ phân rã USE CASE 38

3.3.4 Sơ đồ CDM 38

3.3.5 Sơ đồ LDM 38

3.3.6 Sơ đồ PDM 38

3.3.7 Thiết kế dữ liệu 38

3.3.8 Thiết kế theo chức năng 38

3.3.8.1 Quản lí trạng thái đơn hàng 38

3.3.8.2 Quản lí trạng thái biên nhận 38

3.3.8.3 Tạo đơn hàng 38

3.3.8.4 Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu 38

3.3.8.5 Tìm kiếm và lọc quần áo theo loại có sẵn 38

3.3.8.6 Tìm kiếm đơn hàng 38

3.3.8.7 Đăng nhập hệ thống 38

3.3.8.8 Đăng xuất hệ thống 41

3.3.8.9 Đăng kí tài khoản khách hàng 41

3.4 Kiểm thử 41

3.4.1 Giới thiệu 41

3.4.2 Chi tiết kế hoạch kiểm thử 41

3.4.3 Quản lí kiểm thử 41

3.4.4 Các trường hợp kiểm thử 41

KẾT QUẢ, THẢO LUẬN VÀ HƯỚNG PHÁT TRIỂN 42

TÀI LIỆU THAM KHẢO 43

# KÍ HIỆU VÀ VIẾT TẮT

# DANH SÁCH HÌNH

[Hình 2.1 Giao diện Android 7.0 Nougat 21](#_Toc529744453)

[Hình 2.2 Ví dụ về truy vấn dữ liệu 22](#_Toc529744454)

[Hình 2.3 Ví dụ về gọi một mutation 22](#_Toc529744455)

[Hình 3.1: Mô hình kiến trúc hệ thống 38](#_Toc529744456)

[Hình 3.2 Sơ đồ USE CASE 39](#_Toc529744457)

[Hình 3.3Giao diện đăng nhập trên điện thoại và trên web 41](C:\\Users\\vuphu\\Desktop\\luanvan\\Lu-n-v-n\\baoCao\\bao-cao-luan-van.docx" \l "_Toc529744458)

[Hình 3.4 Sơ đồ xử lí đăng nhập 42](#_Toc529744459)

# DANH MỤC BẢNG

**No table of figures entries found.**

# TÓM TẮT

# ABSTRACT

# TỪ KHÓA

# TỔNG QUAN

## Đặt vấn đề

Trong thời kì xã hội phát triển mạnh mẽ, con người nghĩ đến bản thân mình và yêu công việc nhiều hơn. Chúng ta dành thời gian nhiều hơn cho công việc, bỏ qua công việc dọn dẹp trong nhà, đặc biệt lầ chuyện giặt giũ. Đó là một vấn đề thật mệt mỏi với những người có công việc bận rộn hay cảm thấy nhàm chán với nó. Mỗi lúc như vậy, ta liền tìm ngay đến những cửa hàng dịch vụ giặt giũ. Nhưng vấn đề bất cập ở đây là trong trường hợp ta đang bận không thể đem quần áo đến tận nơi để gửi giặt là thứ nhất, thứ hai nếu chúng ta có nhiều loại quần áo và mong muốn giặt giũ với những hình thức khác nhau nhưng lại không biết cửa hàng nào có đầy đủ các hình thức mình đang cần. Bên cạnh đó, ta không chủ động được thời gian lấy quần áo nếu không được chủ của hàng cho một lịch hẹn sau khi nhận đồ giặt, quần áo của mình cũng mong muốn được chi tiết về các đặc điểm quần áo tránh trường hợp thất lạc trong quá trình sử dụng dịch vụ, cũng như chi phí bỏ ra cho một lần sử dụng dịch vụ không được minh bạch ban đầu.

Đó là vấn đề của người sử dụng dịch vụ, còn đối chủ cửa hàng một phải đối mặt với vấn đề sắp xếp các đơn hàng như thế nào để hoàn tất việc xử lí các đơn hàng một cách nhanh nhất và tiết kiệm nhất có thể. Việc xử lí bằng cách sổ sách ghi chép, hay theo thứ tự đơn hàng nào trước xử lí trước dẫn đến vấn đề những đơn hàng cần xử lí trước hạn giao trả cho khách lại phải trong tình trạng chờ đợi những đơn hàng chưa đến hạn giao trả. Cũng như việc phân loại đồ theo cách thủ công tốn thời gian.

Để giải quyết những vấn đề được nêu trên, ta cần một hệ thống mà hỗ trợ người sử dụng dịch vụ có thể chọn lựa theo yêu cầu của mình cần thiết. Và hỗ trợ đưa ra gợi ý sắp xếp lịch xử lí đơn hàng cho các máy cho chủ cửa hàng kèm với cho họ chủ động sắp xếp từng đơn hàng riêng biệt một cách thủ công. Đó là những điều mà hệ thống này mong muốn mang lại.

## Lịch sử giải quyết vấn đề

Có nhiều giải pháp đã được đặt ra để giải quyết vấn đề: Dịch vụ giặt ủi giao nhận đồ tận nơi, dịch vụ tự giặt ủi, …. Các giải pháp này đặt ra giúp cửa hàng giải quyết các vấn đề cơ bản như: Hỗ trợ khách hàng nhận đồ tận nơi nhưng quần áo của khách hàng không ghi rõ chi tiết để tránh thất lạc đồ khách, cũng như quá trình giao nhận không có biên nhận cho khách hàng kiểm tra đồ của mình. Và quần áo đã nhận về cửa hàng luôn được phân loại theo cách thủ công.

## Phạm vi đề tài

Đề tài được đặt ra với mong muốn giải quyết được vấn đề trong việc tạo đơn hàng cho khách hàng thông qua việc đặt đơn hàng thông qua ứng dụng di động. Ứng dụng hỗ trợ khách hàng chọn dịch vụ mình cần thiết và tìm kiếm những chi nhánh của hàng có hỗ trợ đầy đủ dịch vụ khách hàng đã chọn lựa.

Xây dựng một trang web quản lí thông qua đó cửa hàng có thể quản lí các đơn hàng, biên nhận bằng cách kiểm soát trạng thái của chúng. Hỗ trợ đưa ra gợi ý sắp lịch xử lí đơn hàng cho cửa hàng và phân loại tự động giúp tiết kiệm thời gian nhất có thể.

## Phương pháp nghiên cứu

### Mục tiêu nghiên cứu

Phát triển một mô hình hệ thống giặt ủi dựa trên các công nghệ phổ biến hiện nay gồm:

- Xây dựng một ứng dụng Android hỗ trợ khách hàng tạo đơn hàng và tìm được những chi nhánh giặt ủi của cửa hàng gần nhất trong phạm vi được quy định trước.

- Xây dựng một trong Web quản lí các đơn hàng của khách hàng sau khi họ chấp nhận xây dựng đơn hàng từ ứng dụng Android. Trang Web hỗ trợ nhận viên quản lí đơn hàng theo dõi được tình trạng của đơn hàng thông qua việc quản lí các dơn hàng dựa trên trạng thái của chúng. Cùng với đó, trang web cung cấp tạo đơn hàng nếu khách hàng không đặt hàng thông qua ứng dụng điện thoại.

- Để ứng dụng điện thoại và trang web liên kết với nhau thông qua một Server API trung gian làm nhiệm vụ truy xuất dữ liệu từ cơ sở dữ liệu và trả về cho Client (ứng dụng Android, trang Web).

- Áp dụng giải thuật để giải quyết được bài toán phân chia các đơn hàng vào các máy giặt sao cho thời gian xử lí các đơn hàng là nhanh nhất có thể và đúng thời gian giao trả đồ cho khách hàng.

### Đối tượng nghiên cứu

Nghiên cứu về lập trình Android nói riêng và lập trình di động nói chung. Cách liên kết ứng dụng với hệ thống API thông qua Apollo Client. Cùng kết hợp với sử dụng ReactJS để tạo nên một trang web quản lí đơn hàng.

Tìm hiểu và áp dụng GraphQL, Postgraphile vào xây dựng hệ thống API kiểu mới (một end point).

### Phạm vi nghiên cứu

Nghiên cứu các phương pháp về lập trình Android hiệu quả. Áp dụng các thư viện bổ trợ cho việc tạo ứng dụng nhanh chóng. Đối với tạo trang web bằng ReactJS, việc tạo dựng nên trang web một cách đơn giản phù hợp cho người mới bắt đầu tìm hiểu.

Nghiên cứu tạo Server GraphQL cho người mới bắt đầu kết hợp với Postgrahile, cũng như cách sử dụng cơ sở dữ liệu PostgreSQL. Việc sử dụng Postgrahile phù hợp cho người bắt đầu nghiên cứu, từng bước hiểu được cách xây dựng và viết các Mutation và Query.

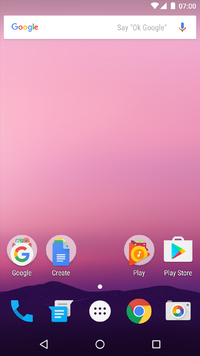
# **CƠ SỞ LÝ THUYẾT**

## Tìm hiểu về nền tảng Android[1]

**Giới thiệu:**

Android là một hệ điều hành được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng, phát triển bởi Google dựa trên nền tảng Linux. Trước đây, Android được phát triển bởi công ty liên hợp Android ( sau đó được Google mua lại vào năm 2005). Android ra mắt vào ngày 5 tháng 11 năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở (Open Handset Alliance) bao gồm 78 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. Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java.

Được xây dựng trên nền tảng mở, thư viện đa năng, mạnh mẽ, Android đã nhanh chóng được cộng đồng lập trình viên hưởng ứng mạnh mẽ. Do đó, Android có cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới (quý 3 năm 2012),với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày.



Hình 2.1 Giao diện Android 7.0 Nougat

## Tìm hiểu về GraphQL [2]

**Giới thiệu:**

GraphQL là một Graph Query Language được dành cho API. Nó được phát triển bởi Facebook và hiện tại nó được duy trì bởi rất nhiều công ty lớn, và mọi cá nhân trên khắp thế giới. GraphQL từ khi ra đời đã gần như thay thế hoàn toàn REST bởi sự hiệu quả, mạnh mẽ và linh hoạt hơn rất nhiều.

**Đặc điểm:**

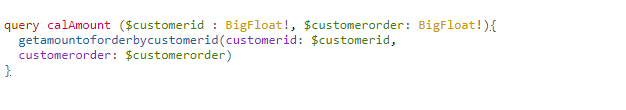
- *Thay thế cho REST:* Vấn đề mà REST đang gặp phải là nó việc phản hồi dữ liệu của REST trả về quá nhiều hoặc là quá ít. Trong cả 2 trường hợp thì hiệu suất của ứng dụng đều bị ảnh hưởng khá nhiều. Giải pháp mà GraphQL đưa ra là cho phép khai báo dữ liệu nơi mà một client có thể xác định chính xác dữ liệu mà mình cần từ một API. Đảm bảo dữ liệu đủ dùng mà không dư thừa, tăng tốc xử lí.

- *Định nghĩa cơ sở dữ liệu và kiểu dữ liệu:*

+ GraphQL có 1 hệ thống riêng dành cho nó được sử dụng để xác định schema của một api. Tất cả type được liệt kê trong một API thì được viết trong schema thì sử dụng GraphQL Schema Definition Language (SDL).

+ Schema này được dùng như là một bản giao dịch giữa client và server để xác định client có thể truy cập dữ liệu như thế nào.

*- Truy vấn dữ liệu (Query):* GraphQL sử dụng việc nạp dữ liệu khác với REST. Nó chí có duy nhất 1 single endpont và hoàn toàn phụ thuộc vào client để xác định những dữ liệu cần thiết. Vì thế client phải chỉ ra các trường cần thiết.



Hình 2.2 Ví dụ về truy vấn dữ liệu

- *Thay đổi dữ liệu (Mutations):* Trong GraphQL viêc gửi các queries được gọi là mutations. Các mutation này có 3 loại là CREATE, UPDATE và DELETE. Mutation cũng có cú pháp giống như try vấn dữ liệu (Query).



Hình 2.3 Ví dụ về gọi một mutation

*- Subscription and Realtime Updates:* Một yêu cầu quan trọng khác đối với nhiều ứng dụng đó chính là realtime, để có thể kết nối đến máy chủ để có được thông tin về các event ngay lập tức. Trong trường hợp này, GraphQL cung cấp các khái niệm gọi là subscriptions. Khi 1 client subscriptions một event, nó cũng bắt đầu và giữ các kết nối đến server. Bất cứ khi nào sự kiện đó xảy ra, server sẽ đẩy dữ liệu tương ứng đến client. Không giống như Query và Mutation, nó đi theo kiểu như “request-response-cycle”, nó sẽ subscriptions đại diện của luồng dữ liệu được gửi đến client. Subscriptions được viết bằng cách sử dụng cú pháp như Query và Mutation.

## Tìm hiểu về Postgraphile [3][4]

**Giới thiệu:**

Postgraphile được xem như thành phần để kết nối giữa GraphQL và PostgreSQL lại với nhau. Postgraphile phân tích và trả chó GraphQL những thông tin về cơ sở dữ liệu. Trong quá trinh phát triển, Postgraphile hỗ trợ kiểm tra cở sở dữ liệu thay đổi và cập nhật như thế nào sau mỗi lần gọi API bằng GraphQL.

**Đặc điểm:**

- *User and Session Management:* Postgrahile cung cấp một phương pháp quản lí Session một cách linh hoạt là JWT (JSON Web Tokens). Postgraphile chỉ cần một Sercet Key (mã bí mật) và một kiểu dữ liệu trả về, Postgrahile sẽ mã hóa nội dụng như một JWT token và đánh dấu nó.

- *Hiệu năng, kết nối hiệu quả:* Postgraphile cung cấp một hiệu năng truy xuất nhanh chóng, không gặp tình trạng N+1 query. Bên cạnh đó, nó còn hỗ trợ người sử dụng

- *Tự động tìm và tạo các quan hệ dựa trên cơ sở dữ liệu:* Postgraphile dựa trên các khóa ngoại tồn tại trong cở sở dữ liệu để sinh các liên kết khi truy xuất dữ liệu.

*- Tạo các tùy biến Query và Mutations:* Ta có dễ dàng tạo các query cũng như mutation thông qua việc tạo các function hay procedure trong cơ sở dữ liệu.

## Tìm hiểu về PostgreSQL [5]

**Giới thiệu:**

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay. PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do.

**Đặc điểm:**

- *Cung cấp đầy đủ các tính năng cần có và hỗ trợ mở rộng dễ dàng:* PostgreSQL sở hữu các bộ tính năng mạnh mẽ bao gồm kiểm soát truy cập đồng thời nhiều phiên bản (MVCC), phục hồi điểm thời gian, điều khiển truy cập hạt, không gian bảng, sao chép không đồng bộ, các giao dịch lồng nhau, sao lưu trực tuyến / nóng, một kế hoạch truy vấn / tối ưu hóa. Nó hỗ trợ bộ ký tự quốc tế, mã hóa nhiều byte, Unicode, và nó dễ dàng nhận dạng địa phương trong việc sắp xếp tìm kiếm. PostgreSQL có khả năng mở rộng cao cả về số lượng dữ liệu mà nó có thể quản lý và số lượng người dùng đồng thời có thể đáp ứng.

- *Có độ tin cậy cao và tuân thủ đủ tiêu chuẩn*: PostgreSQL là một cơ sở dữ liệu với khả năng chịu lỗi cao. Cơ sở đóng góp mã nguồn mở của nó cho phép nó xây dựng mạng hỗ trợ cộng đồng. PostgreSQL tuân thủ ACID và hỗ trợ đầy đủ các khoá ngoại, tham gia, chế độ xem, trình kích hoạt và thủ tục lưu trữ bằng nhiều ngôn ngữ khác nhau.

- *Mã nguồn mở:* Mã nguồn PostgreSQL có sẵn dưới giấy phép mã nguồn mở, cho phép bạn tự do sử dụng, sửa đổi và thực hiện nó như bạn thấy phù hợp, miễn phí. PostgreSQL không có chi phí bản quyền, giúp loại bỏ rủi ro cho việc triển khai quá mức.

## Tìm hiểu về JSON Web Token [6]

**Giới thiệu:**

JWT là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server, các thông tin trong chuỗi JWT được định dạng bằng JSON. Trong đó chuỗi Token phải có 3 phần là header, phần payload và phần signature được ngăn bằng dấu “.”.

**Đặc điểm:**

JWT có ưu điểm là mã hóa được nhiều thông tin. JWT gửi cho client mà không cần lưu phía server, nên không như một số giải pháp cũ là lưu session phía server và dùng 1 key gửi client, sau đó nhận key và kiểm tra session để xác thực và quyền.

Kịch khi sử dụng JWT thường diễn ra theo các trường hợp:

- Truy cập không xác thực sẽ báo lỗi. Yêu cầu xác thực server sẽ xác thực và mã hóa thông tin cần thiết cho lần giải mã sau, rồi trả về token (ở đây là JSON Web Token).

- Truy cập xác thực sẽ luôn kèm token trong header, hoặc phương thức POST, hoặc trên URL. Phía server sẽ giải mã JWT token nhận được và kiểm tra những yêu cầu như user, role, permission (tùy trường hợp) có trong payload của JWT.

## Tìm hiểu về ReactJS [7]

**Giới thiệu:**

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

**Đặc điểm:**

- *ReactJS cực kì hiệu quả*: ReactJS tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. ReactJS cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp ReactJS tránh những thao tác cần trên DOM mà nhiều chi phí.

*- ReactJS giúp việc viết các đoạn code JS dễ dàng hơn:* Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của ReactJS. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.

- *Nó có nhiều công cụ phát triển:* Khi bắt đầu sử dụng ReactJS, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho ReactJS. Nó giúp ta bắt lỗi code dễ dàng hơn. Sau khi ta cài đặt ứng dụng này, ta sẽ có cái nhìn trực tiếp vào virtual DOM như thể ta đang xem cây DOM thông thường.

- *Render tầng server:* Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.

*- Làm việc với vấn đề test giao diện*: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS. Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

## Tìm hiểu về Apollo Client [8]

**Giới thiệu:**

Apollo Client là một cách thức nhanh chóng khi muốn sử dụng GraphQL để tạo nên một ứng dụng người dùng. Nó giúp ta định nghĩa được những dữ liệu cần thiết và đủ để dựng nên một giao diện cho người dùng nhanh nhất có thể. Apollo Client hỗ trợ cho rất nhiều frontend platform như React, Vue.js, Angular, Android, Swift, ….

**Đặc điểm:**

- *Dễ dàng tích hợp với các frontend platform:* Nhờ việc hỗ trợ nhiều loại platform phổ biến hiện nay nên việc áp dụng Apollo Client vào việc hỗ trợ ứng dụng truy xuất dữ liệu từ API một cách dễ dàng. Ta không cần tốn quá nhiều thời gian để tìm hiểu, tích hợp, sử dụng.

- *Giúp kiểm soát và hiểu được cách ứng dụng hoạt động:* Bằng cách sử dụng cú pháp truy vấn của GraphQL nên việc sử dụng đơn giản. Nhờ đó, ta nắm bắt được lượng dữ liệu cần thiết cho mỗi bước hoạt động của ứng dụng và hiểu chính xác mỗi bước nó cần những gì tránh việc bị dư thừa dữ liệu không cần thiết. Bên cạnh đó Apollo Client được xây dựng nên bởi cộng đồng nên phù hợp đa dạng các trường hợp mà ta cần sử dụng.

# NỘI DUNG NGHIÊN CỨU

## Mô tả bài toán

### Bối cảnh hệ thống

Các hệ thống API được viết thông qua GraphQL ngày càng nhiều hơn so với RESTful API bởi sự tiện lợi. Cũng như, ta dễ dàng truy xuất thông tin mong muốn mà không phải cấu hình quá nhiều, tránh được vấn đề dư thừa dữ liệu không cần thiết khi truy xuất dữ liệu.

Người dùng hiện nay sử dụng điện thoại thường trực và Android là hiện điều hành mà có lượng người dùng đông đảo vì giá cả cũng như dễ dàng tạo và phát triển ứng dụng. Những điều đó hỗ trợ cho việc triển khai ứng dụng cho mọi người dễ dàng hơn.

ReactJS là một thư viện được Facebook hỗ trợ mạnh mẽ. Đó là một điểm mạnh giúp ta xây dựng một trang web nhanh chóng hơn, sử dụng ngay không cần quá hiểu rõ việc xây dựng một trang web theo nhưng Framework chuẩn.

### Các chức năng hệ thống

Các chức năng hệ thống cần đạt được trong đề tài đặt ra bao gồm:

- Quản lí trạng thái đơn hàng.

- Quản lí trạng thái biên nhận.

- Quản lí phân công xử lí đơn hàng.

- Tạo đơn hàng.

- Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu.

- Tìm kiếm và lọc quần áo theo loại có sẵn.

- Tìm kiếm đơn hàng.

- Đăng nhập, đăng xuất hệ thống.

- Đăng kí tài khoản khách hàng.

|  |  |  |
| --- | --- | --- |
| **STT** | **Mã chức năng** | **Tên chức năng** |
| 1 | GU\_01 | Quản lí trạng thái đơn hàng |
| 2 | GU\_02 | Quản lí trạng thái biên nhận |
| 3 | GU\_03 | Quản lí phân công xử lí đơn hàng |
| 4 | GU\_04 | Tạo đơn hàng |
| 5 | GU\_05 | Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu |
| 6 | GU\_06 | Tìm kiếm và lọc quần áo theo loại có sẵn |
| 7 | GU\_07 | Tìm kiếm đơn hàng |
| 8 | GU\_08 | Đăng nhập |
| 9 | GU\_09 | Đăng xuất |
| 10 | GU\_10 | Đăng kí tài khoản khách hàng |

Bảng 3.1 Các chức năng hệ thống

### Đặc điểm người dùng

Hệ thống bao gồm 2 nhóm người dùng chính: Nhân viên cửa hàng và khách hàng:

- *Nhân viên cửa hàng:* Để đáp ứng các khâu trong việc xử lí đơn hàng, nhận viên cửa hàng được chia làm ba loại nhận viên chính:

+ *Nhân viên quản lí đơn hàng:* Là người dùng hiện tại có nhiều quyền nhất trong việc quyết định xử lí đơn hang với mã là STAFF\_01.

+ *Nhân viên xử lí đơn hàng:* Là người có nhiệm vụ cập nhật trạng thái đơn hàng khi bắt đầu xử lí đơn hàng cũng như sau khi hoàn tất đơn hàng với mã là STAFF\_02.

+ *Nhân viên nhận và trả quần áo:* Là người có nhiệm vụ cập nhật là thông tin quần áo đã nhận (bao gồm số lượng, thời gian nhận và ngày nhận, …) và cập nhật trạng thái đơn hàng đã nhận cũng như giao trả quần áo cho khách hang với mã là STAFF\_03.

- *Khách hàng:* Là người dùng có thể đặt đơn hàng từ ứng dụng điện thoại hoặc trực tiếp từ cửa hàng.

### Môi trường vận hành

Đối với ứng dụng đặt đơn hàng chỉ hỗ trợ trên nền tảng Android với phiên bản từ 5.0 trở lên, được sử dụng bởi người dùng là *Khách hàng.*

Đối với trang web quản lí dành cho người dùng là *Nhân viên cửa hàng* sử dụng trên nền tảng web hỗ trợ truy cập thông qua các trình duyệt phổ biến hiện nay (Chrome, MS EDGE, …), do trang web được sử dụng hiển thị các thông tin chủ yếu bằng bảng dữ liệu nên việc truy cập thông qua điện thoại không được tối ưu tốt. Bên cạnh đó, yêu cầu trình duyệt phải được bật JavaScript.

Server API được viết bằng ngôn ngữ NodeJS và cơ sở dữ liệu là Postgres nên dễ dàng triển khai trên nhiều nền tảng khác nhau. Hiện tại, server được chạy toàn bộ dưới máy tính cá nhân.

## Đặc tả yêu cầu

### Yêu cầu chức năng

#### Quản lí trạng thái đơn hàng

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_01 |
| **Tên chức năng** | Quản lí trạng thái đơn hàng |
| **Đối tượng sử dụng** | Nhân viên cửa hàng (Nhân viên quản lí đơn hàng, Nhân viên xử lí đơn hàng) |
| **Tiền điều kiện** | Truy cập được trang web quản lí và đăng nhập thành công vào hệ thống. |
| **Cách xử lí** | Bước 1: Click “*Quản lí đơn hàng*” ở bên thanh menu cạnh trái và chọn trạng thái của đơn hàng. Danh mục con của quản lí đơn hàng được hiển thị như sau:   * *Nhân viên quản lí đơn hàng*: Đang chờ, đang chờ xử lí, đang xử lí, đã xử lí hoàn tất, thành công, đơn hàng bị hủy * *Nhân viên xử lí đơn hàng:* Đang xử lí, đã xử lí hoàn tất.   Bước 2: Danh sách đơn hàng được hiển thị theo dạng bảng. Ở đây người dùng có thể tìm kiếm đơn hàng dựa trên các tiêu chí là các cột của bảng.  Bước 3: Khi người dùng nhấn vào tên khách hàng để truy cập vào chi tiết đơn hàng. Ở đây, người dùng có thể xem thông tin chi tiết đơn hàng và có thể truy cập vào biên nhận của đơn hàng (nếu tồn tại). Các chức năng có thể tại trang chi tiết đơn hàng theo loại nhân viên và trạng thái đơn hàng:   * Trạng thái “*đang chờ*”: Nhân viên quản lí đơn hàng thực hiện chức năng chấp nhận, hủy đơn hàng. Nếu người dùng nhấn “*chấp nhận*” trạng thái đơn sẽ chuyển thành “*đã chấp nhận*” và tự động sinh ra một biên nhận tương ứng với đơn hàng ở trạng thái “*đang chờ nhận đồ*”. Nếu người dùng nhấn “*hủy đơn*”, đơn hàng sẽ chuyển trạng thái thành “*đã hủy*”. * Trạng thái “*đang chờ xử lí*”: Khi nhân viên xử lí đơn hàng nhấn lên nút xử lí. Trạng thái đơn hàng chuyển thành *“đang xử lí*” và người dùng được gán thành người thực hiện đơn hàng đó. * Trạng thái *“đang xử lí”:* Khi nhân viên xử lí đơn hàng nhấn lên nút hoàn tất. Trạng thái đơn hàng chuyển thành *“đã xử lí hoàn tất”.*  Và chỉ nhân viên thực hiện đơn hàng đó mới thấy được nút hoàn tất. Biên nhận của đơn hàng chuyển trạng thái thành *“đang chờ trả đồ”.* * Trạng thái *“đã xử lí hoàn tất”:* Nhân viên quản lí đơn hàng có thể nhấn lên nút tạo hóa đơn để sinh hóa đơn dựa trên biên nhận. |
| **Kết quả** | Hiển thị thông tin tất cả đơn hàng dưới dạng bảng.  Khi nhấn vào tên khách hàng hiển thị chi tiết đơn hàng. |
| **Ghi chú** |  |

#### Quản lí trạng thái biên nhận

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_02 |
| **Tên chức năng** | Quản lí trạng thái biên nhận |
| **Đối tượng sử dụng** | Nhân viên cửa hàng (Nhân viên quản lí đơn hàng, Nhân viên nhận và trả quần áo) |
| **Tiền điều kiện** | Truy cập được trang web quản lí và đăng nhập thành công vào hệ thống. |
| **Cách xử lí** | Bước 1: Click “*Quản lí biên nhận*” ở bên thanh menu cạnh trái và chọn trạng thái của biên nhận. Danh mục con của quản lí biên nhận được hiển thị như sau:   * *Nhân viên quản lí đơn hàng*: Đang chờ nhận đồ, đã nhận đồ, đang chờ giao đồ, đã giao đồ. * *Nhân viên nhận và trả quần áo:* Đang chờ nhận đồ, đã nhận đồ, đang chờ giao đồ, đã giao đồ.   Bước 2: Danh sách biên nhận được hiển thị theo dạng bảng. Ở đây người dùng có thể tìm kiếm biên nhận dựa trên các tiêu chí là các cột của bảng.  Bước 3: Khi người dùng nhấn vào tên khách hàng để truy cập vào chi tiết biên nhận. Ở đây, người dùng có thể xem thông tin chi tiết biên nhận. Các chức năng có thể tại trang chi tiết biên nhận theo loại nhân viên và trạng thái đơn hàng:   * Trạng thái “*đang chờ nhận đồ*”: Nhân viên nhận và trả quần áo nhấn “*chấp nhận*”, người dùng sẽ được gán thành người đi nhận đơn hàng đó và có nhiệm vụ cập nhật thông tin biên nhận (bao gồm số lượng đồ nhận, ngày nhận và thời gian nhận). Sau khi nhận đồ hoàn tất buộc nhấn nút *“đã nhận”* để thay đổi trạng thái biên nhận thành *“đã nhận đồ”* và đơn hàng ứng với biên nhận chuyển từ *“đã nhận”* thành *“đang chờ xử lí”.* * Trạng thái *“đang chờ giao đồ”:* Nhân viên nhận và trả quần ảo nhấn vào nút *“giao đồ”,* người dùng sẽ được gán thành người đi giao đơn hàng đó và có nhiệm vụ câp nhật thông tin biên nhận (bao gồm thời gian, ngày giao đơn hàng). Sau khi giao hoàn tất buộc nhấn nút “*đã giao*” và đơn hàng ứng với biên nhận chuyển từ “*đã xử lí hoàn tất*” thành *“thành công”.* |
| **Kết quả** | Hiển thị thông tin tất cả biên nhận dưới dạng bảng.  Khi nhấn vào tên khách hàng hiển thị chi tiết biên nhận. |
| **Ghi chú** |  |

#### Quản lí phân công xử lí đơn hàng

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_04 |
| **Tên chức năng** | Quản lí phân công xử lí đơn hàng |
| **Đối tượng sử dụng** | Nhân viên cửa hàng (Nhân viên quản lí cửa hàng) |
| **Tiền điều kiện** | Truy cập được trang web quản lí đối với nhân viên cửa hàng và đăng nhập thành công. |
| **Cách xử lí** | Phân công loại một:  Bước 1: Phân loại đơn hàng theo thứ tự loại dịch vụ trước và nhóm màu sau cùng. Sau đó, lưu thành từng túi giặt trong cơ sở dữ liệu.  Bước 2: Phân công mỗi đơn hàng được xử lí trên một máy (tương ứng tất cả túi giặt của đơn hàng sẽ cùng có một mã máy giặt).   * Ưu tiên các máy có số đơn hàng đang đợi là ít nhất. * Các đơn hàng được sắp xếp theo thứ tự tang dần dựa trên ngày và khung giờ trả đồ cho khách hàng. * Các đơn hàng cùng xử lí trên một máy sẽ được gán thứ tự xử lí.   Bước 3: Lưu kết quả vào cơ sở dữ liệu.  Phân công loại hai:  Bước 1: Tương tự bước 1 của phân công loại một.  Bước 2: |
| **Kết quả** | Hiển thị được bảng phân công bao gồm các thông tin: mã máy giặt + số thứ tự xử lí, tên khách hàng + mã số đơn hàng, mã biên nhận, trạng thái đơn hàng. |
| **Ghi chú** | Một đơn hàng có thể có một hoặc nhiều túi giặt khác nhau dựa trên phân loại. |

#### Tạo đơn hàng

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_04 |
| **Tên chức năng** | Tạo đơn hàng |
| **Đối tượng sử dụng** | Nhân viên cửa hàng (Nhân viên quản lí cửa hàng), khách hàng |
| **Tiền điều kiện** | Truy cập được trang web quản lí đối với nhân viên cửa hàng và ứng dụng điện thoại đối với khách hàng và đăng nhập thành công. |
| **Cách xử lí** | Đối với đặt đơn hàng từ trang quản lí (Nhân viên quản lí cửa hàng):  Bước 1: Nhấn vào *“tạo đơn hàng”* ở thanh danh mục  bên trái màn hình.  Bước 2: Những thông tin được mặc định sẵn: Thông tin chi nhánh, danh sách loại dịch vụ theo chi nhánh.  Bước 3: Người dùng nhập các thông tin khách hàng, địa chỉ lấy và trả đồ, ngày lấy và trả đồ cho khách, chọn khung giờ lấy và trả đồ. Nếu ngày lấy và trả đồ cùng một ngày, thì khung giờ lấy và trả đồ cách ít nhất là 1 khung giờ.  Bước 4: Nhập thông tin từng quần áo bao gồm: loại dịch vụ, loại quần áo, đơn vị tính, số lượng, màu sắc, …. Ít nhất phải tồn tại một quần áo trong đơn hàng. Nếu rỗng báo lỗi.  Bước 5: Nhấn nút *“đặt hàng”.* Đơn hàng được lưu vào cơ sở dữ liệu với trạng thái là *“nháp”.* Và chuyển sang trang xác nhận đơn hàng với thông tin chi tiết và tổng giá tiền đối với đơn hàng.  Bước 6: Nhấn nút “*đặt hàng*” một lần nữa để xác nhận đơn hàng. Đơn hàng được cập nhật với trạng thái *“đang chờ”.*  Đối với đặt đơn hàng tử ứng dụng điện thoại (khách hàng):  Bước 1: Người dùng chọn loại dịch vụ mong muốn. Kế tiếp chọn đơn vị tính là cái hay kilogram.  Bước 2: Người dùng chọn những quần áo dành cho loại dịch vụ này. Ở đây người dùng có thể dùng chức năng “*GU\_06*” để giúp thêm quần áo nhanh chóng. Khi người dùng chọn một quần áo, thông tin về số lượng là bắt buộc. Mọi thông tin đơn hàng được giữ tạm thời vào trong giỏ hàng.  Bước 3: Nếu người dùng có nhu cầu đặt thêm dịch vụ, quay lại trang chọn dịch vụ và thực lại tuần tự các bước 1, 2.  Bước 4: Người dùng truy cập vào màn hình giỏ hàng và xác nhận đặt đơn hàng. Sau đó chuyển sang màn hình thực hiện chức “*GU\_05*”. Người dùng chọn chi nhánh mong muốn.  Bước 5: Chuyển sang màn hình xác nhận đặt hàng, người dùng xem được thông tin đơn hàng của mình và phí phải trả cho đơn hàng này. Người dùng nhập ngày lấy và trả đồ cho khách, chọn khung giờ lấy và trả đồ. Nếu ngày lấy và trả đồ cùng một ngày, thì khung giờ lấy và trả đồ cách ít nhất là 1 khung giờ.  Bước 6: Người dùng nhấn *“xác nhận”* lần cuối. Đơn hàng được gửi lên server và lưu lại vào cơ sở dữ liệu với trạng thái *“đang chờ”.* Không lưu trạng thái là *“nháp”.*  Bước 7: Người dùng sẽ được chuyển sang màn hình cảm ơn cùng với mã QR Code ứng với đơn hàng. |
| **Kết quả** | Lưu đơn hàng vào cơ sở dữ liệu với trạng thái *“đang chờ”.* |
| **Ghi chú** | Toàn bộ thông tin ở chức năng tạo đơn hàng là bắt buộc. Nếu không được nhập sẽ báo lỗi.  Thông tin đơn hàng sẽ được lưu lại trong SharePreferences của ứng dụng khi chưa được người dùng đặt đơn hàng.  Mã QR Code được tạo ra bởi ID đơn hàng + ngày đặt đơn hàng + tên khách hàng. |

#### Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_05 |
| **Tên chức năng** | Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu |
| **Đối tượng sử dụng** | Khách hàng |
| **Tiền điều kiện** | Truy cập được ứng dụng điện thoại và đăng nhập thành công vào hệ thống. Người dùng đang ở bước nhập thông tin địa chỉ nơi lấy, trả đồ và chọn chi nhánh để đặt đơn hàng. |
| **Cách xử lí** | Bước 1: Dựa trên vị trí người dùng ứng dụng sẽ trả về các chi nhánh gần người dùng nhất trong phạm vi mặc định trước.  Bước 2: Kiểm tra các chi nhánh có hỗ trợ đầy đủ các dịch vụ người dùng yêu cầu hay không? Nếu có sẽ hiển thị lên màn hình.  Bước 3: Sử dụng vị trí người dùng làm vị lấy và trả đồ cho khách hàng nếu người dùng không thay đổi. |
| **Kết quả** | Hiển thị tất cả các chi nhánh trong phạm vi cho trước mà có hỗ trợ đầy đủ các dịch vụ người dùng đã chọn lên màn hình. |
| **Ghi chú** | Yêu cầu người dùng phải bật GPS và cho phép truy cập vị trí người dùng.  Khi có vị trí người dùng, hiển thị vị trí người dùng ở trung tâm màn hình. |

#### Tìm kiếm và lọc quần áo theo loại có sẵn

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_06 |
| **Tên chức năng** | Tìm kiếm và lọc quần áo theo loại có sẵn |
| **Đối tượng sử dụng** | Khách hàng |
| **Tiền điều kiện** | Truy cập được ứng dụng điện thoại và đăng nhập thành công vào hệ thống. Đang ở bước chọn quần áo thêm vào giỏ. |
| **Cách xử lí** | Bước 1: Người dùng nhấn vào *“loại quần áo”.* Và chọn một loại quần áo muốn lọc. Hoặc nhấn vào biểu tượng tìm kiếm và nhập tên quần áo tìm kiếm.  Bước 2: Ứng dụng dựa trên thông tin người dùng chọn hoặc nhập vào để lọc các quần áo và hiển thị lại cho người dùng chọn. |
| **Kết quả** | Nếu tồn tại có kết quả sẽ hiển thị theo dạng danh sách cho người dùng.  Nếu không có kết quả sẽ hiển thị rỗng. |
| **Ghi chú** | Để tìm kiếm hay lọc, người dùng bắt buộc phải chọn hoặc nhập thông tin tìm kiếm. |

#### Tìm kiếm đơn hàng

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_07 |
| **Tên chức năng** | Tìm kiếm đơn hàng |
| **Đối tượng sử dụng** | Nhân viên cửa hàng |
| **Tiền điều kiện** | Truy cập được trang web quản lí và đăng nhập thành công vào hệ thống. |
| **Cách xử lí** | Bước 1: Chọn một trong ba hình thức để tìm kiếm: Quét mã QR – Code, tên khách hàng hoặc mã đơn hàng.  Bước 2: Nhập các thông tin yêu cầu.  Bước 3: Nhấn nút *“tìm kiếm”.* |
| **Kết quả** | Hiển thị kết quả mởi khung kế bên khung tìm kiếm.  Kết quả tìm kiếm bao gồm:   * Tên khách hàng (liên kết với trang xem thông tin chi tiết đơn hàng). * Số điện thoại, email * Trạng thái đơn hàng |
| **Ghi chú** | Nếu không có thông tin nào nhập, Khi người dùng nhấn tìm kiếm, kết quả sẽ hiển thị tất cả.  Mặc định và nếu không có kết quả sẽ hiển thị *“không có kết quả nào”.* |

#### Đăng nhập hệ thống

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_08 |
| **Tên chức năng** | Đăng nhập hệ thống |
| **Đối tượng sử dụng** | Nhân viên cửa hàng, khách hàng |
| **Tiền điều kiện** | Truy cập được trang web quản lí đối với nhân viên cửa hàng và ứng dụng điện thoại đối với khách hàng. |
| **Cách xử lí** | Bước 1: Người dùng cần nhập email và mật khẩu.  Bước 2: Nhấn nút *“Đăng nhập”.*  Bước 3: Hệ thống server API kiểm trả tài khoản vừa nhập đúng hay sai. Nếu đúng trả về một chuỗi token để người dùng gửi kèm mỗi khi muốn truy xuất dữ liệu và được lưu lại tạm thời trên ứng dụng điện thoại thông qua SharePreferences và Local Storage đối với trang web. Ngược lại, thông báo lỗi. |
| **Kết quả** | Người dùng sẽ chuyển vào trang chính đối với người dùng là nhân viên cửa hàng. Đối với người dùng khách hàng chuyển vào màn hình chính của ứng dụng điện thoại. |
| **Ghi chú** | Các thông tin email và mật khẩu là yêu cầu bắt buộc.  Nếu đường truyền mạng lỗi, thì thông báo lỗi cho người dùng. |

#### Đăng xuất hệ thống

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_09 |
| **Tên chức năng** | Đăng xuất hệ thống |
| **Đối tượng sử dụng** | Nhân viên cửa hàng, khách hàng |
| **Tiền điều kiện** | Truy cập được trang web quản lí đối với nhân viên cửa hàng và ứng dụng điện thoại đối với khách hàng và đăng nhập thành công |
| **Cách xử lí** | Bước 1: Click vào Đăng xuất ở góc phải trên đối với trang web và Tài khoản -> Đăng xuất đối với ứng dụng điện thoại  Bước 2: Ứng dụng cũng như trang web sẽ xóa toàn bộ thông tin để đăng nhập và thông tin lưu tạm thời ra khỏi SharePreferences, Local Storage.  Bước 3: Tự động chuyển về trang đăng nhập. |
| **Kết quả** | Người dùng quay lại trang đăng nhập |
| **Ghi chú** | Bắt buộc mọi thông tin, dữ liệu lưu tạm thời phải được xóa sạch. |

#### Đăng kí tài khoản khách hàng

|  |  |
| --- | --- |
| **Mã yêu cầu** | GU\_10 |
| **Tên chức năng** | Đăng kí tài khoản khách hàng |
| **Đối tượng sử dụng** | Khách hàng |
| **Tiền điều kiện** | Truy cập ứng dụng điện thoại đối với khách hàng. |
| **Cách xử lí** | Bước 1: Tại màn hình đăng nhập, nhấn vào *“Tạo tài khoản mới”.*  Bước 2: Nhập các thông tin bắt buộc bao gồm: Họ và tên, email và mật khẩu mong muốn. Sau đó, nhấn vào nút “Đăng kí”.  Bước 3: Hệ thống sẽ dựa trên email người dùng đã nhập kiểm tra email đã tồn tại trong hệ thống chưa? Nếu đã tồn tại, thông báo lỗi tài khoản đã tồn tại. Nếu không, hệ thống tạo tài khoản mới cho người dùng.  Bước 4: Chuyển về trang đăng nhập. Với email và password đã được điền trước. Nếu người dùng mới nhấn *“đăng nhập”,* hệ thống thực hiên đăng nhập theo “*GU\_08*” và chuyển vào màn hình “*cập nhật thông tin*”.  Bước 5: Tại màn hình “*cập nhật thông tin*”, người dùng mới phải nhập đầy đủ các thông tin yêu cầu. Sau đó nhấn *“cập nhật”.* Người dùng mới được chuyển sang màn hình chính. |
| **Kết quả** | Người dùng truy cập vào màn hình chính và có đủ các chức năng của người dùng khách hàng |
| **Ghi chú** | Mọi thông tin yêu cầu nhập đều là bắt buộc. Nếu chưa nhập vào sẽ thông báo lỗi yêu cầu nhập. |

### Yêu cầu phi chức năng

### Yêu cầu thực thi

Giao diện đồng nhất đối với cả ứng dụng điện thoại và trang web. Sử dụng tông màu đơn giản hài hòa tạo thiện cảm khi sử dụng.

Đối với ứng dụng điện thoại, mọi dữ liệu điều được truy xuất lại từ server mỗi lần sử dụng ứng dụng.

### Yêu cầu chất lượng phần mềm

Tính đúng đắn: các chức năng của hệ thống hoạt động đúng theo yêu cầu.

Tính khả chuyển: ứng dụng dễ dàng cài đặt và chạy tốt trên mọi phiên bản từ 5.0 trở lên và nhiều loại thiết bị Android khác nhau.

Tính có thể bảo trì: mã nguồn được viết rõ ràng, dễ đọc, dễ bảo trì, cung cấp tài liệu cài đặt phần mềm.

Khả năng chịu lỗi: ứng dụng có khả năng xử lý lỗi khi gặp sự cố, đưa ra thông báo khi gặp lỗi.

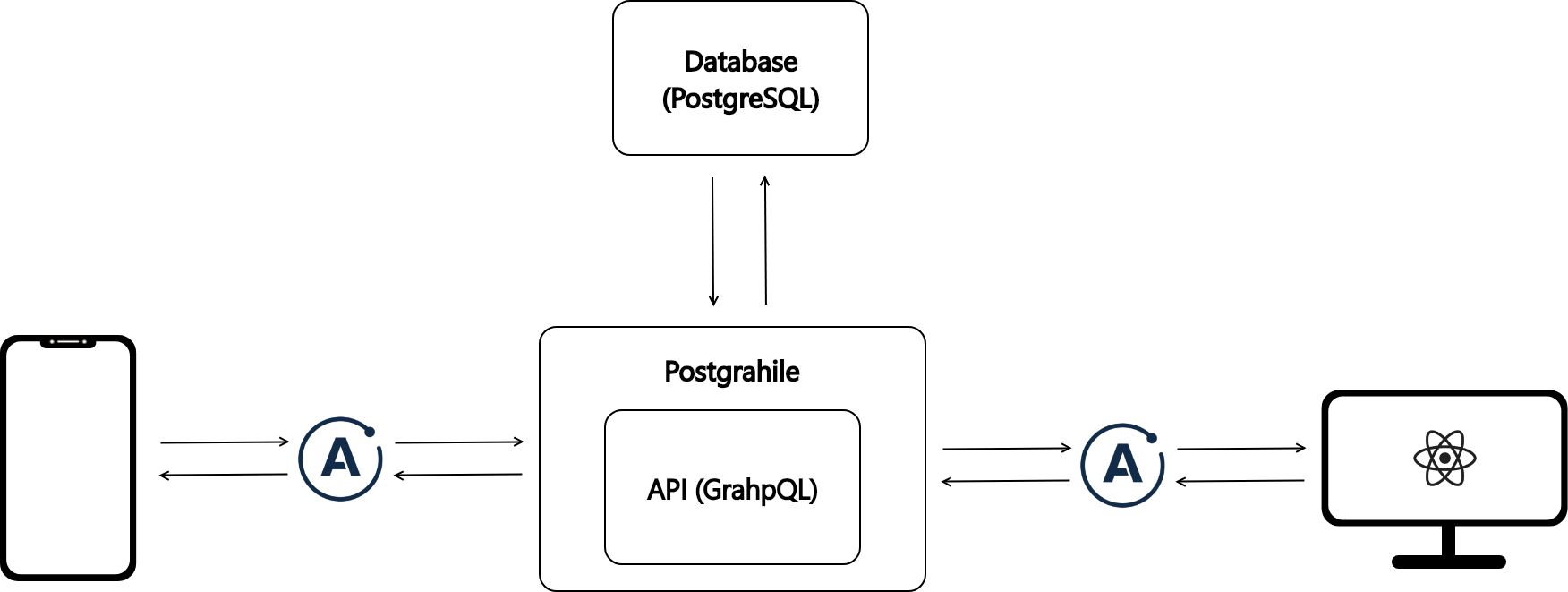
#### Các quy tắc nghiệp vụ

Trong quá trình phát triển ứng dụng, không sử dụng các phần mềm lậu để phân tích, đặc tả, thiết kế, cài đặt, kiểm thử và triển khai.

Ứng dụng không chứa các thông tin sai lệch, không vi phạm các quy định của pháp luật.

## Thiết kế và cài đặt

### Kiến trúc hệ thống



Hình 3.1: Mô hình kiến trúc hệ thống

Hệ thống được xây dựng theo mô hình Web Service APIs. Với server API được xây dựng dựa trên GraphQL và Postgrahile với nhiệm vụ chính là cung cấp các chức năng thêm, sửa, xóa dữ liệu và các chức năng xử lí khác cho client. Client thực hiện các chức năng cung cấp dữ liệu chuẩn cho server xử lí.

### Sơ đồ USE CASE



Hình 3.2 Sơ đồ USE CASE

### Sơ đồ phân rã USE CASE

### Sơ đồ CDM

### Sơ đồ LDM

### Sơ đồ PDM

### Thiết kế dữ liệu

### Thiết kế theo chức năng

#### Quản lí trạng thái đơn hàng

#### Quản lí trạng thái biên nhận

#### Tạo đơn hàng

#### Tìm kiếm chi nhánh gần nhất, có đủ các dịch vụ theo yêu cầu

#### Tìm kiếm và lọc quần áo theo loại có sẵn

#### Tìm kiếm đơn hàng

#### Đăng nhập hệ thống

##### Mục đích

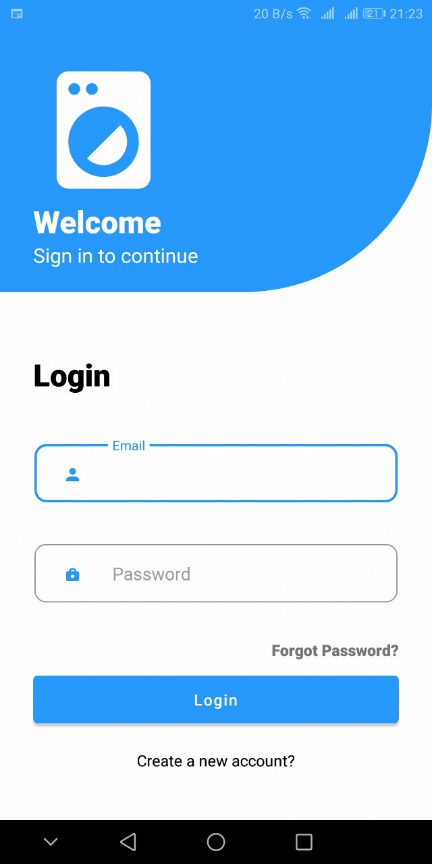
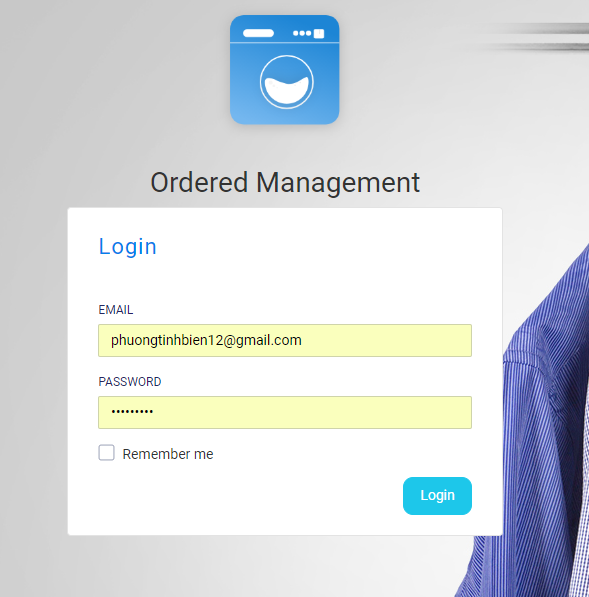
Xác thực quyền truy cập của người dùng vào các tài nguyên của hệ thống. Việc xác thực được trên phuong thức xác thực bằng token (JWT). Chuỗi xác thực bao gồm: loại tài khoản + ID người dùng + loại người dùng.

Loại tài khoản được chia làm hai loại: Tài khoản vô danh và tài khoản đã được xác thực.

Loại người dùng gồm hai loại: người dùng khách hàng (customer\_type), người dùng nhận viên (staff\_type).

##### Giao diện

Hình 3.3Giao diện đăng nhập trên điện thoại và trên web



##### Các thành phần giao diện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Loại điều khiển** | **Nội dung thực hiện** | **Giá trị mặc định** | **Lưu ý** |
| 1 | inputText | Nhập địa chỉ email |  |  |
| 2 | inputText | Nhập mật khẩu |  |  |
| 3 | button | Đăng nhập |  |  |

##### Dữ liệu sử dụng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên bảng/**  **Cấu trúc dữ liệu** | **Phương thức** | | | |
| **Thêm** | **Sửa** | **Xóa** | **Truy vấn** |
| 1 | user |  |  |  | X |
| 2 | customer |  |  |  | X |
| 3 | staff |  |  |  | X |
| 4 | staff\_type |  |  |  | X |

##### Cách xử lí



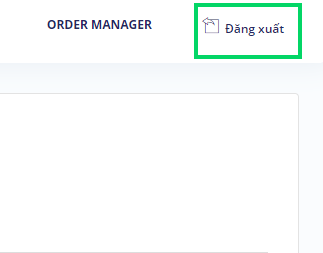
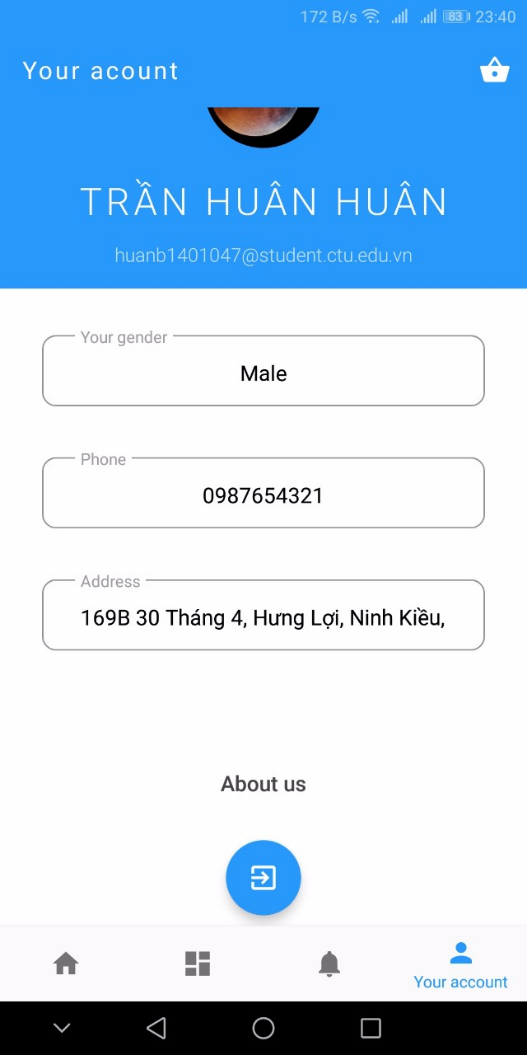
Hình 3.4 Sơ đồ xử lí đăng nhập

#### Đăng xuất hệ thống

##### Mục đích

##### Giao diện

Hình 3.5 Giao diện xử lí đăng xuất



##### Các thành phần giao diện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Loại điều khiển** | **Nội dung thực hiện** | **Giá trị mặc định** | **Lưu ý** |
| 1 | floatingButon | Đăng xuất |  |  |
| 2 | Link | Đăng xuất |  |  |

##### Dữ liệu sử dụng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên bảng/**  **Cấu trúc dữ liệu** | **Phương thức** | | | |
| **Thêm** | **Sửa** | **Xóa** | **Truy vấn** |
| 1 | Local Storage |  |  | X |  |
| 2 | Share Preferences |  |  | X |  |

##### Cách xử lí



Hình 3.6 Sơ đồ xử lí đăng xuất

#### Đăng kí tài khoản khách hàng

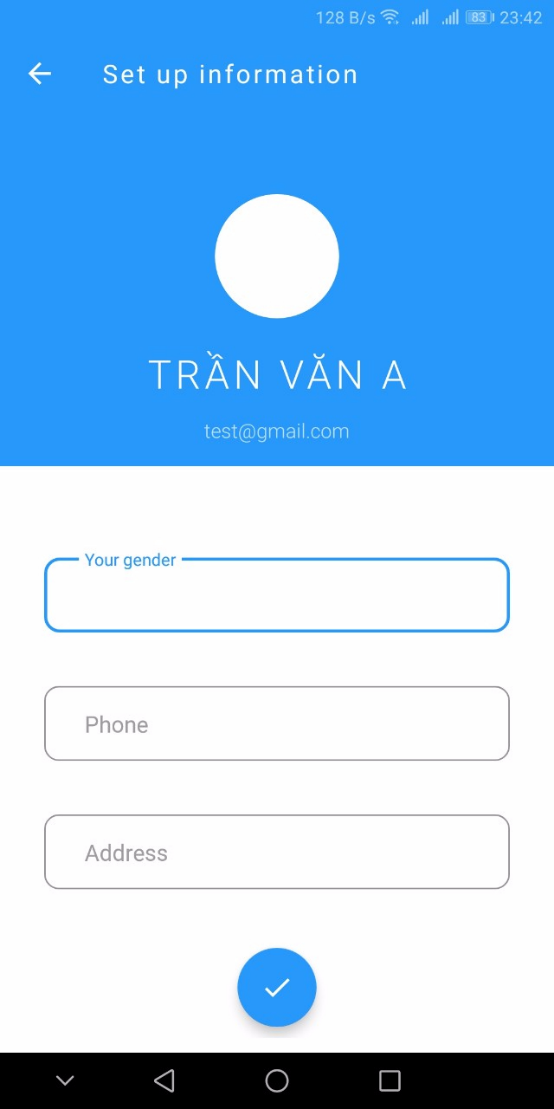
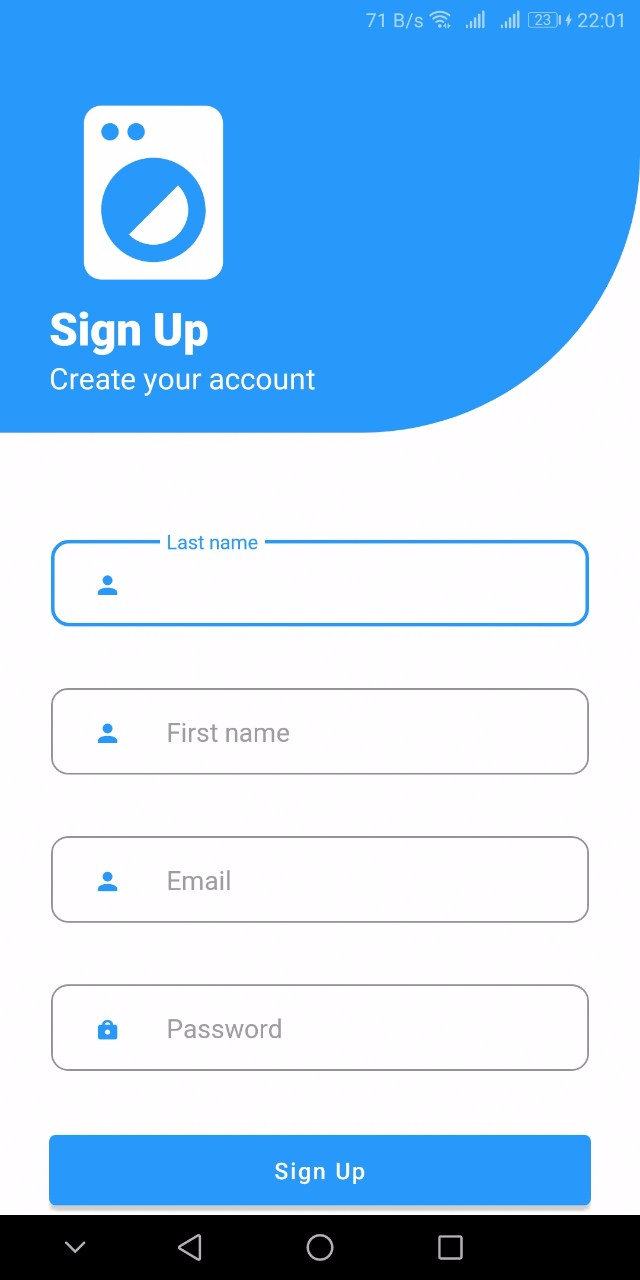
##### Mục đích

Chức năng đang kí tài khoản khách hàng được xây dựng nhằm mục đích hỗ trợ những khách hàng mới có nhu cầu muốn sử dụng dịch vụ. Việc đăng kí tài khoản hỗ trợ việc đăng kí bằng địa chỉ email. Sau khi, khách hàng đăng kí thành công buộc phải cập nhật thông tin cần thiết trước khi sử dụng dịch vụ.

##### Giao diện

##### Các thành phần giao diện

Hình 3.7 Giao diện đăng kí và cập nhật thông tin tài khoản



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Loại điều khiển** | **Nội dung thực hiện** | **Giá trị mặc định** | **Lưu ý** |
| 1 | floatingButon | Đăng xuất |  |  |
| 2 | Link | Đăng xuất |  |  |

##### Dữ liệu sử dụng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên bảng/**  **Cấu trúc dữ liệu** | **Phương thức** | | | |
| **Thêm** | **Sửa** | **Xóa** | **Truy vấn** |
| 1 | Local Storage |  |  | X |  |
| 2 | Share Preferences |  |  | X |  |

##### Cách xử lí

## Kiểm thử

### Giới thiệu

### Chi tiết kế hoạch kiểm thử

### Quản lí kiểm thử

### Các trường hợp kiểm thử

# KẾT QUẢ, THẢO LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Đạt được**

**Hạn chế**

**Hướng phát triển**

# TÀI LIỆU THAM KHẢO

1. Android
2. “Cùng tìm hiểu về GraphQL”: <https://viblo.asia/p/cung-tim-hieu-ve-graphql-07LKX4zeKV4>
3. “Automatically building and maintaining GraphQL APIs with PostgreSQLQL and Postgraphile” : [https://itnext.io/automatically-building-and-maintaining-graphql-apis-with-PostgreSQLql-and-postgraphile-c497636abd29](https://itnext.io/automatically-building-and-maintaining-graphql-apis-with-postgresql-and-postgraphile-c497636abd29)
4. “Postgraphile”: <https://www.graphile.org/postgraphile/introduction/>
5. “PostgreSQL”:
6. “JSON Web Token”:
7. “ReactJS”:<https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7>
8. “Apollo Client cho React”: <https://www.apollographql.com/docs/react/>