# Chương 1. Khảo sát hệ thống

## Mô tả hệ thống

### Nhiệm vụ cơ bản

Xuất phát từ nhu cầu các doanh nghiệp, công ty trong tất cả các lĩnh vực, muốn nhân viên của mình có được sự tiện nghi tốt nhất, giúp cho tinh thần làm việc của nhân viên luôn đạt trạng thái tốt nhất. Đặc biệt là về nhu cầu ăn uống, nhân viên văn phòng thường có xu hướng không muốn hoặc không có thời gian để ra ngoài ăn trưa, giải khát,… Ngoài những nhân viên đã mang cơm từ nhà, những nhân viên khác có nhu cầu rất cao về việc đươc phục vụ, vận chuyển cơm trưa, đồ giải khát ngay tại tỏng công ty, có thể là đến phòng ăn chung của công ty, hoặc ngay tại chỗ ngồi của mình. Dựa trên những nhu cầu cần thiết như vậy các doanh nghiệp, công ty có quy mô vừa và lớn thường có căng – tin hoặc nhà bếp riêng để phục vụ nhân viên, nhưng do quy mô văn phòng quá rộng, toà nhà nhiều tầng, căng tin thì lại không đủ chỗ để mọi người có thể trực tiếp lên mua đồ ăn vào giờ cao điểm, giờ ăn trưa vì quá nhiều người phải xếp hàng. Do vậy công ty có nhu cầu xây dựng một ứng dụng để nhân viên có thể đặt đồ ăn, đồ uống qua đó mà không cần phải đến trực tiếp căng tin để xếp hàng. Từ nhu cầu đó, sau đây là một số nhiệm vụ cơ bản mà ứng dụng cần đáp ứng :

* Giành cho nhân viên công ty:
  + Xem danh sách sản phẩm theo ngày và theo bữa ăn
  + Tìm kiếm sản phẩm
  + Mua sản phẩm
  + Xem lịch sử đơn hàng đã đặt
  + Thêm sản phẩm yêu thích
* Giành cho căng tin:
  + Quản lý danh sách sản phẩm
  + Quản lý đơn hàng
  + Quản lý thực đơn
  + Thống kê doanh thu

1.1.2 Cơ cấu tổ chức

* Bộ phận quản lý
* Bộ phận giao hàng
* Bộ phận nhà bếp

1.1.3 Quy trình xử lý

1.1.3.1 Quy trình đặt hàng

* Quy trình tìm kiếm sản phẩm và thêm sản phẩm vào giỏ hàng
  + Người dùng ứng dụng (nhân viên công ty) có nhu cầu đặt đồ ăn sáng, ăn trưa hoặc đồ uống, đầu tiên người dùng sẽ mở ứng dụng đã được cài đặt trên điện thoại di động, sau đó tìm kiếm các sản phẩm muốn mua theo thực đơn được hiển thị sẵn theo ngày trên ứng dụng:
    - Thực đơn được sắp sếp theo ngày trong tuần từ thứ 2 đến thứ 6
    - Trong thực đơn của mỗi ngày được chia thành 3 phần:
      * Món ăn sáng
      * Món ăn trưa
      * Đồ uống
    - Món ăn được hiển thị trong từng phần, mỗi sản phẩm sẽ bao gồm:
      * Tên sản phẩm
      * Số lượng còn lại
      * Giá sản phẩm
      * Giá sản phẩm còn lại sau khi đã giảm giá (nếu có)
      * Nút thêm vào giỏ hàng
  + Người dùng ứng dụng click vào nút thêm vào giỏ hàng của sản phẩm muốn mua hoặc click vào sản phẩm để xem chi tiết sản phẩm bao gồm các thông tin chi tiết:
    - Tên đầy đủ của sản phẩm
    - Số lượng còn lại
    - Mô tả sản phẩm
    - Giá sản phẩm
    - Giá sản phẩm còn lại sau khi đã giảm giá (nếu có)
    - Nút “mua ngay”
* Quy trình đặt hàng và thanh toán đơn hàng
  + Người dùng ứng dụng bấm vào nút giỏ hàng ở trên màn hình
  + ứng dụng sẽ hiển thị danh sách các sản phẩm người dùng đã thêm vào giỏ hàng bao gồm các thông tin:
    - Tên đầy đủ của sản phẩm
    - Số lượng từng sản phẩm trong giỏ hàng
    - Mô tả sản phẩm
    - Giá sản phẩm
    - Giá sản phẩm còn lại sau khi đã giảm giá (nếu có)
  + Người dùng nhập và kiểm tra các thông tin của sản phẩm trong giỏ hàng, ngoài các thông tin sản phẩm, người dùng có thể nhập thêm và kiểm tra các thông tin:
    - Tổng số lượng các sản phẩm trong giỏ hàng
    - Tổng tiền cần thanh toán
    - Vị trí nhận đồ ăn (từ tầng 1 đến tầng 8)
    - Ghi chú thêm cho sản phẩm
  + Người dùng chọn hình thức thanh toán đơn hàng
    - Tiền mặt
    - Thẻ tín dụng/ tài khoản ngân hàng hoặc ví điện tử VNPAY
  + Người dùng bấm vào nút thanh toán
    - Nếu người dùng chưa đăng nhập: yêu cầu người dùng đăng nhập hệ thống trước khi thanh toán hoá đơn
      * Điều hướng đến trang đăng nhập
      * Người dùng đăng nhập bằng tài khoản đã đăng kí hoặc email
      * Đăng nhập xong ứng dụng điều hướng trở lại ứng dụng
    - Nếu người dùng đã đăng nhập hệ thống: chuyển tới bước tiếp theo
  + Nếu thanh toán bằng tiền mặt: đơn hàng đã được yêu cầu xác nhận, ứng dụng sẽ quay về màn hình chính
  + Nếu thanh toán bằng thẻ tín dụng / tài khoản ngân hàng hoặc ví điện tử VNPAY:
    - Ứng dụng chuyển sang màn hình chọn phương thức thanh toán:
      * Thẻ ATM và tài khoản ngân hàng
      * Thẻ thanh toán quốc tế
      * Ví điện tử
    - Người dùng chọn hình thức thanh toán “thẻ ATM và tài khoản ngân hàng”, sau đó chọn ngân hàng:
      * ứng dụng chuyển sang màn hình nhập các thông tin sau:
        + Số thẻ
        + Ngày phát hành thẻ
        + Tên chủ thẻ
      * Người dùng chọn nút xác thực: ứng dụng chuyển qua màn hình nhập mã sms otp đã được gửi về tin nhắn qua số điện thoại được liên kết với tài khoản ngân hàng tương ứng
      * Người dụng bấm nút “Xác nhận”: đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính
    - Người dùng chọn thanh toán qua “thẻ thanh toán quốc tế” và chọn 1 loại thẻ:
      * Ứng dụng chuyển qua màn nhập các thông tin sau:
        + Số thẻ
        + Ngày hết hạn
        + Số CCV/CVV
        + Họ tên khách hàng
        + Email
        + Địa chỉ
        + Tỉnh/Thành phố
        + Đất nước
      * Người dùng chọn xác thực: đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính.
    - Người dùng chọn thanh toán qua ví điện tử VNPAY:
      * Ứng dụng chuyển qua màn hình hiển thị mã QR code
      * Người dùng bật ứng dụng “Ví điện tử VNPAY” hoặc bất kì ứng dụng smart banking nào được cung cấp dịch vụ QR code từ VNPAY(ví dụ như vietcombank, Agribank, BIDV,…) để quét mã QR
      * Sau khi quét mã QR người dùng thực hiện xác thực yêu cầu trên các ứng dụng tương ứng, đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính.

1.1.3.2 Quy tình xác nhận đơn hàng từ trang quản trị của căng tin:

* Nhân viên căng tin đăng nhập vào ứng dụng web được cung cấp cho căng tin
* Nhân viên căng tin chuyển qua màn hình quản lý đơn hàng, tìm kiếm các đơn hàng chưa được xác nhận và bấm nút “xác nhận đơn hàng”. Đơn hàng đã được chuyển qua trạng thái “chờ vận chuyển”

1.1.3.3 Quy trình vận chuyển đơn hàng

* Nhân viên căng tin đăng nhập vào ứng dụng web được cung cấp cho căng tin
* Nhân viên căng tin chuyển qua màn hình quản lý đơn hàng, tìm kiếm các đơn hàng đã được xác nhận và chưa được vận chuyển và bấm nút “vận chuyển”. Đơn hàng đã được chuyển qua trạng thái “đang vận chuyển”
* Nhân viên căng tin mang đồ ăn/ đồ uống đã được làm và hoá đơn được máy tính suất ra, tới phòng ăn của từng tầng làm việc trong công ty, sau đó quay lại chọn hoàn thành đơn hàng

1.1.3.4 Quy trình thêm sản phẩm vào hệ thống

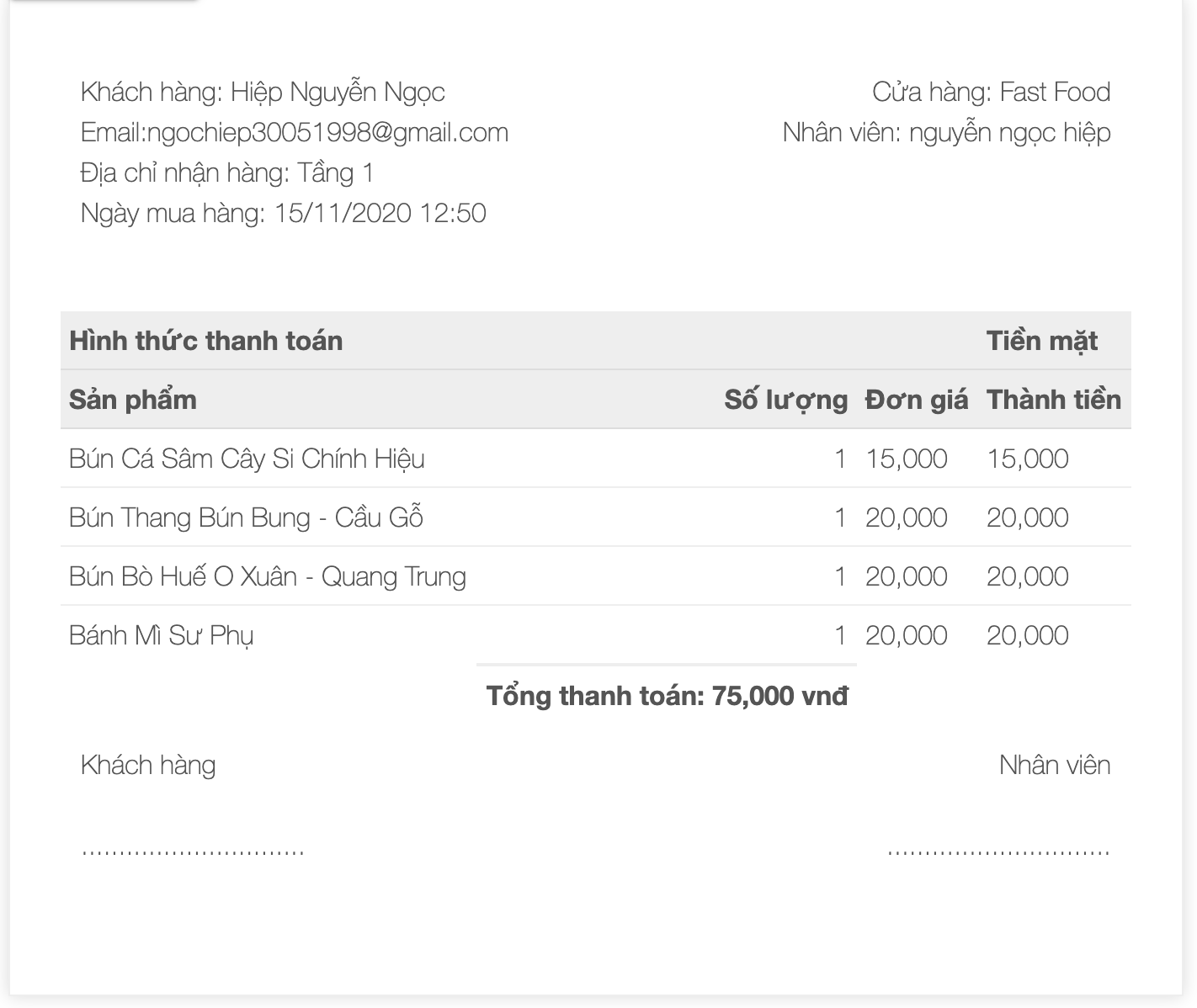
* Nhân viên căng tin đăng nhập vào hệ thống ứng dụng web được cung cấp cho căng tin
* Nhân viên căng tin chuyển qua màn hình quản lý sản phẩm hoặc bấm vào nút thêm mới sản phẩm
* Nhân viên căng tin bấm vào nút thêm sản phẩm mới sau đó hệ thống chuyển sang trang thêm mới sản phẩm
* Người dùng nhập các thông tin cần thiết của sản phẩm:
  + Tên sản phẩm
  + Loại sản phẩm
  + Đơn giá
  + Giá khuyến mại (nếu có)
  + Ảnh sản phẩm
* Người dùng bấm vào nút thêm mới: sản phẩm sẽ được thêm mới vào hệ thống

1.1.3.5 Quy trình tạo thực đơn cho từng ngày làm việc

* Nhân viên căng tin đăng nhập vào hệ thống ứng dụng web được cung cấp cho căng tin
* Nhân viên căng tin chuyển qua màn hình quản lý thực đơn
* Nhân viên căng tin chọn ngày muốn tạo thực đơn, hệ thống sẽ hiện thị những món ăn đã được thêm trước đó (nếu có)
* Nhân viên căng tin chọn bữa ăn muốn sửa:
  + Bữa sáng
  + Bữa trưa
  + Đồ uống
* Nhân viên căng tin bấm nút thêm mới, hệ thống sẽ mở popup để tìm kiếm các sản phẩm đã có trong hệ thống
* Nhân viên căng tin tìm kiếm sản phẩm, chỉnh sửa các thông tin cần thiết như:
  + Giá sản phẩm
  + Giá khuyến mại(nếu có)
  + Mô tả
  + Số lượng
  + Đơn vị tính
* Nhân viên căng tin bấm vào nút “thêm”: sản phẩm sẽ được thêm vào thực đơn tương ứng

1.1.4 Mẫu biểu

* Hoá đơn thanh toán



Hình vẽ 1.1.4

## Mô hình hóa hệ thống

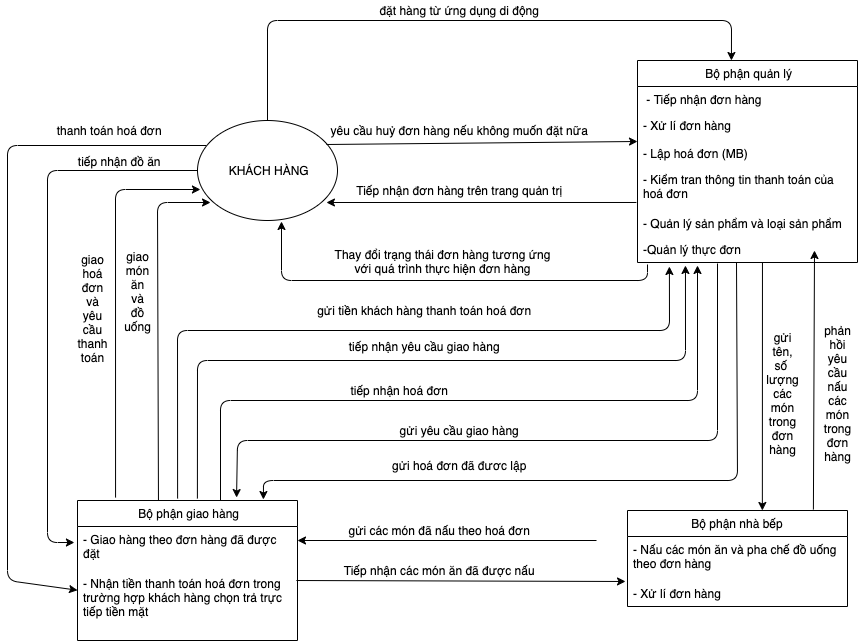
### Mô hình tiến trình nghiệp vụ

* Định nghĩa ký hiệu sử dụng
* Bộ phận trong hệ thống:

|  |
| --- |
| Tên bộ phận |
| Chức năng 1  Chức năng 2  …….. |

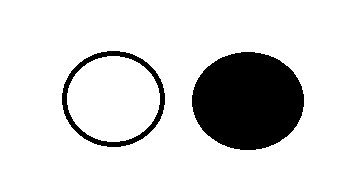
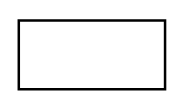
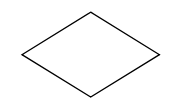
* Tác nhân tác động vào hệ thống:

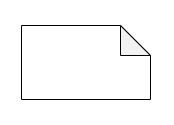
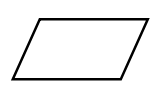
* Luồng thông tin:
* Vẽ biểu đồ tiến trình nghiệp vụ



Hình vẽ 1.2.1

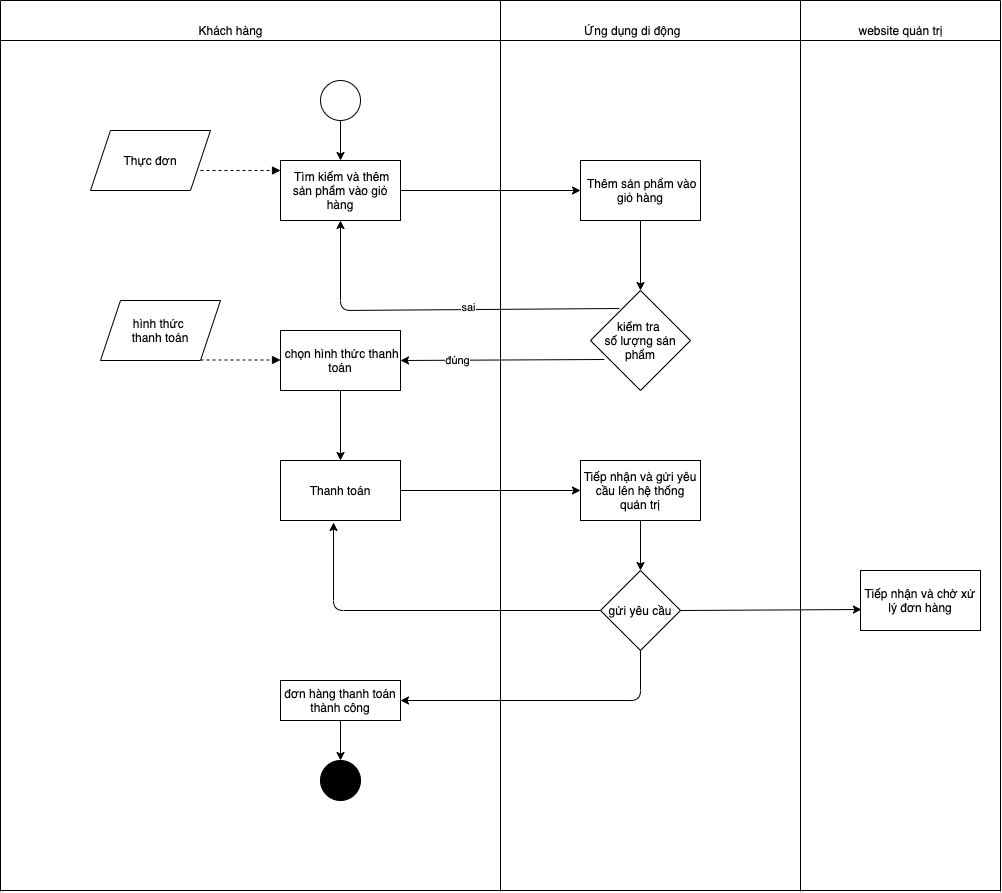
### Biểu đồ hoạt động

* Định nghĩa ký hiệu sử dụng
* Bắt đầu/kết thúc:
* Công việc:
* Điều kiện rẽ nhánh:

* Giấy tờ giao dịch:
* Kho dữ liệu:
* Luồng công việc/ luồng dữ liệu:

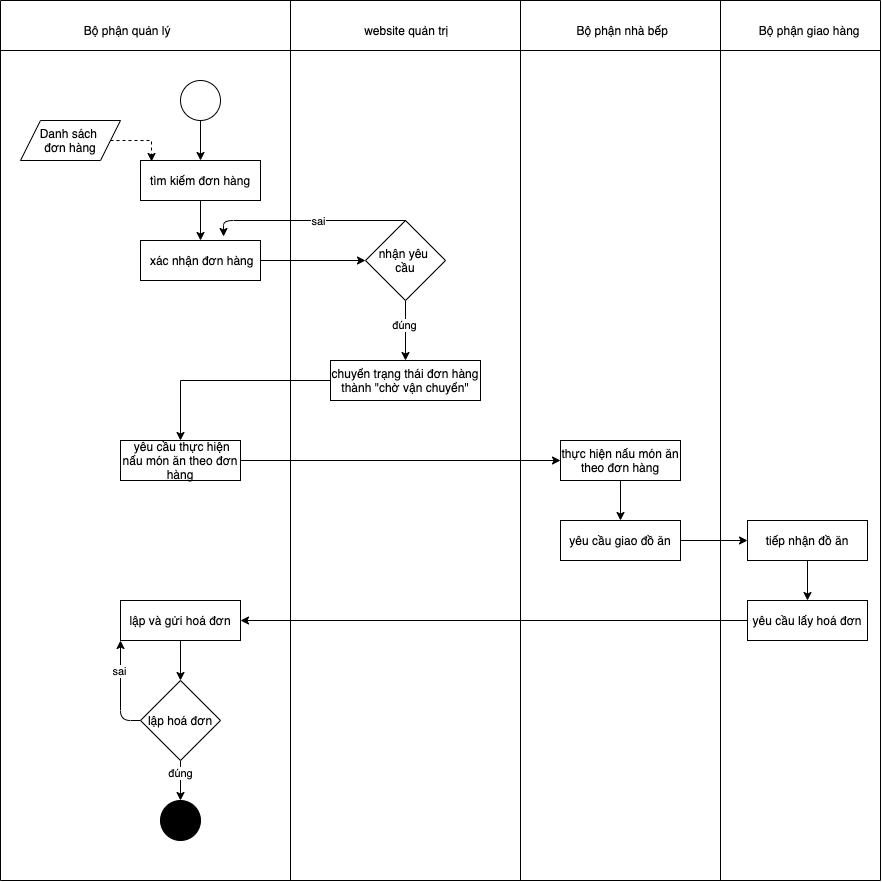


* Vẽ biểu đồ hoạt động:
  + Quy trình đặt hàng



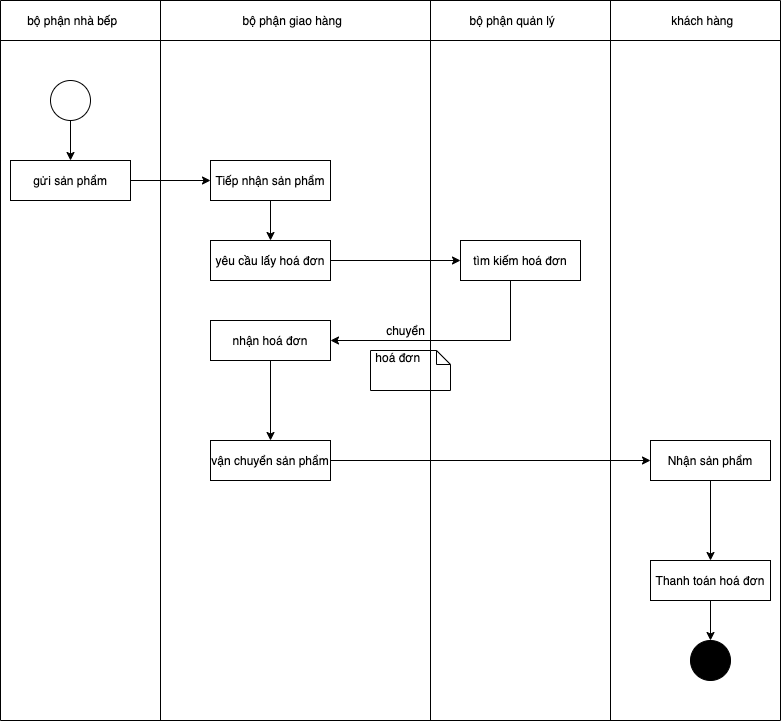
Hình vẽ 1.2.1

* + Quy trình xử lý đơn hàng



Hình vẽ 1.2.2

* + Quy trình vận chuyển đơn hàng

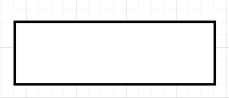


Hình vẽ 1.2.3

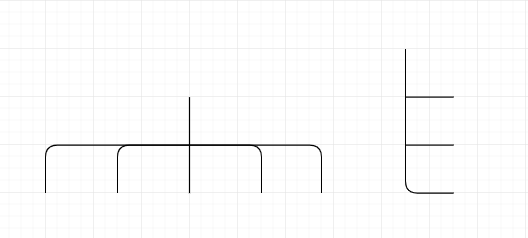
# Chương 2. Phân tích hệ thống

## Phân tích chức năng nghiệp vụ

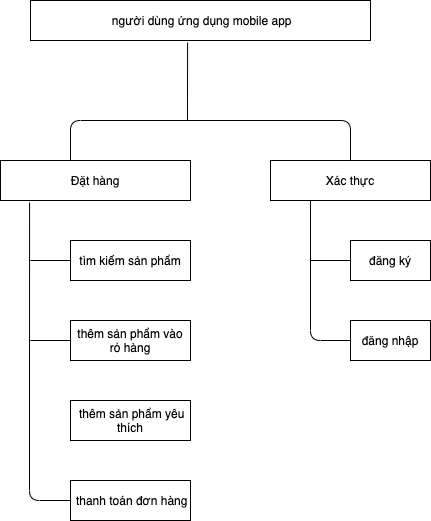
* + 1. Sơ đồ phân rã chức năng
* Ký hiệu:
  + Chức năng (công việc cần làm)



* + Quan hệ phân cấp (mỗi chức năng phân rã ra nhiều chức năng con)

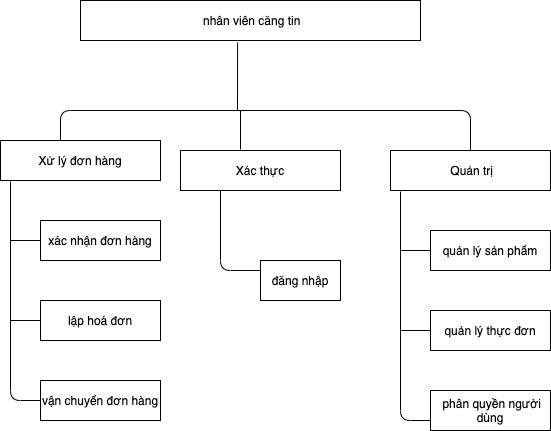


* + - 1. Sơ đồ phân rã chức năng dành cho người dùng ứng dụng mobile app



Hình vẽ 2.1.1.1

* + - 1. Sơ đồ phân rã chức năng dành cho người dùng website quản trị (nhân viên căng tin)



Hình vẽ 2.1.1.2

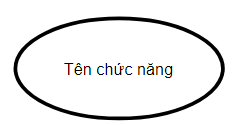
* + 1. Sơ đồ luồng dữ liệu

#### **Ký hiệu sử dụng**

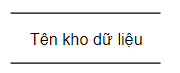
* Luồng dữ liệu: Luồng dữ thông tin vào hoặc ra khỏi tiến trình, được biểu diến bằng mũi tên có ghi thông tin di chuyển



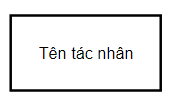
* Tiến trình: Một hoạt động liên quan đến sự biến đổi hoặc tác động của thông tin, được biểu diễn bằng hình oval, bên trong có tên của chức năng đó



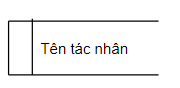
* Kho dữ liệu: Biểu diễn thông tin cần cất giữ, để một hoặc nhiều tiến trình sử dụng chúng, biểu diễn cặp đường thẳng song song chứa tên của thông tin được cất giữ



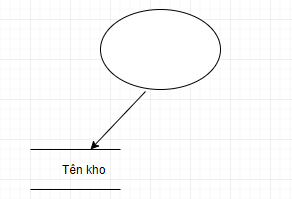
* Tác nhân bên ngoài: Là một người hoặc một nhóm người nằm ngoài hệ thống nhưng có trao đổi trực tiếp với hệ thống



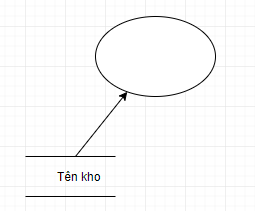
* Tác nhân bên trong: Là một tiến trình của hệ thống đang xét nhưng được trình bày ở một trang khác của biểu đồ



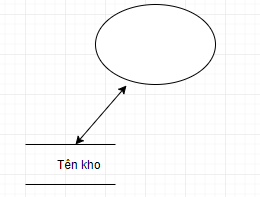
* Một số quy tắc:
  + Đưa dữ liệu vào kho dữ liệu:



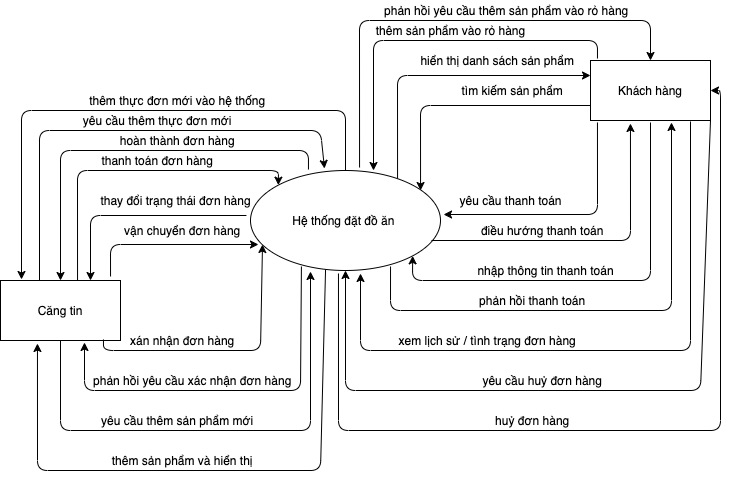
* + Lấy dữ liệu ra từ kho dữ liệu:



* + Cập nhật dữ liệu:

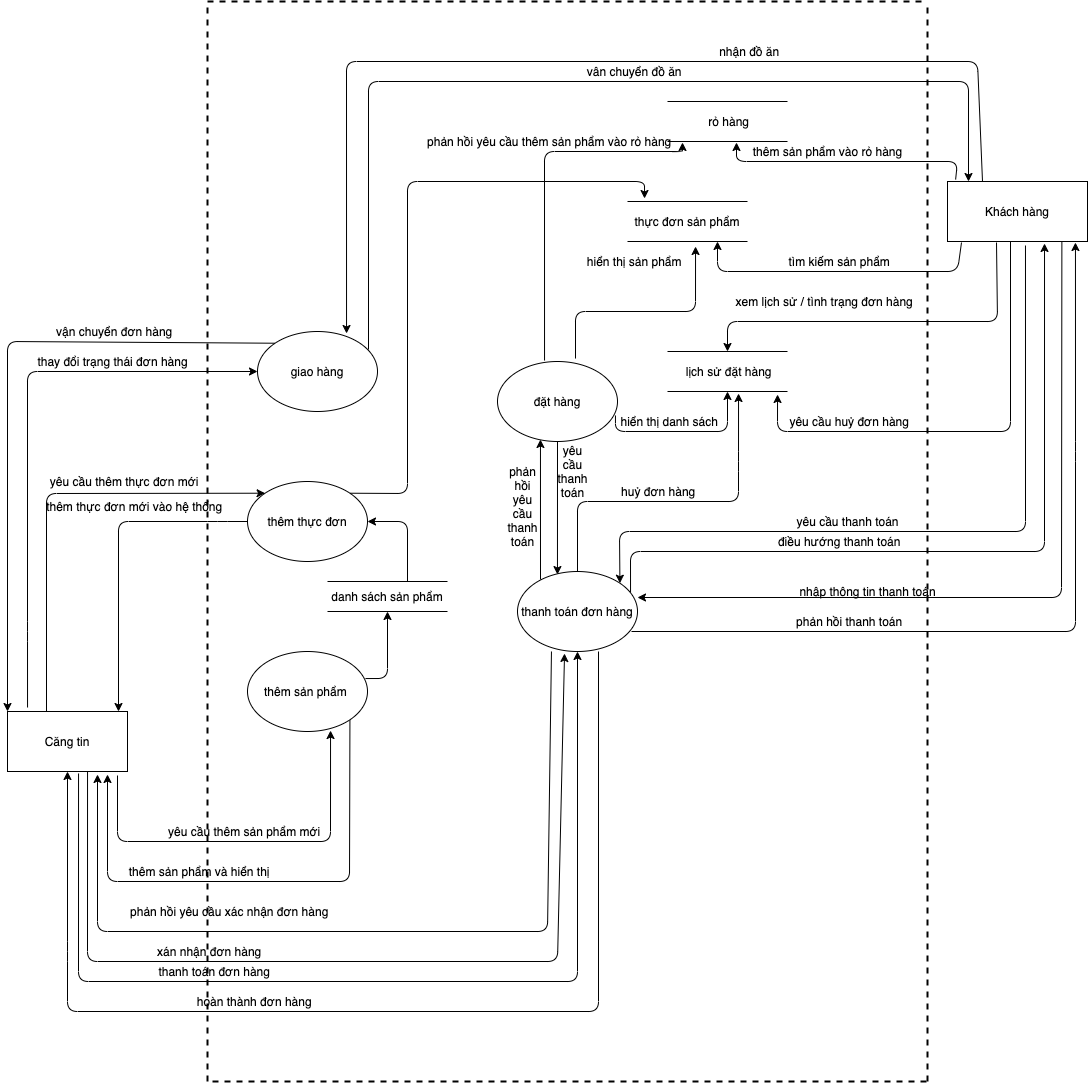


* + - 1. DFD mức khung cảnh (DFD 0)



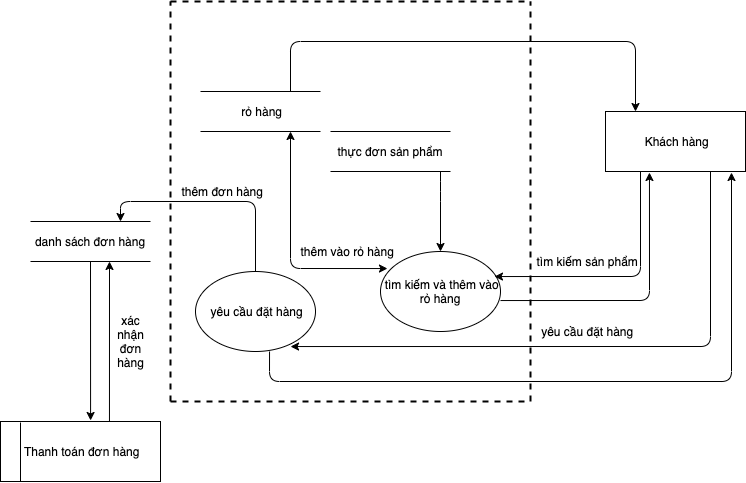
Hình vẽ 2.1.2.1

* + - 1. DFD mức đỉnh (DFD 1)



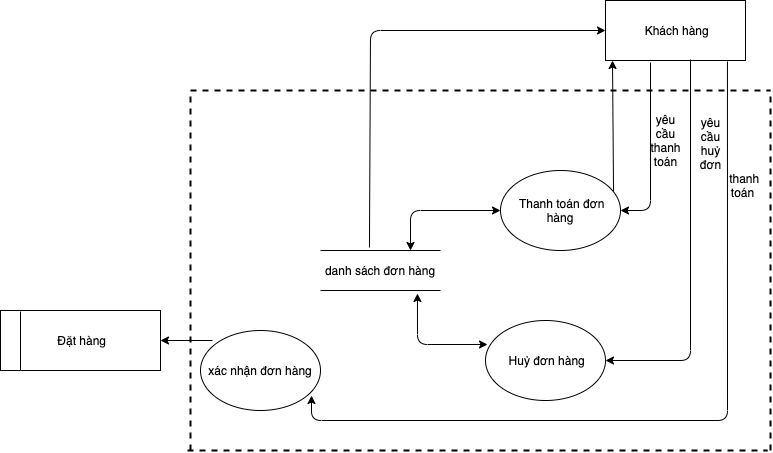
Hình vẽ 2.1.2.2

* + - 1. DFD mức dưới đỉnh(DFD 2)
* Đặt hàng



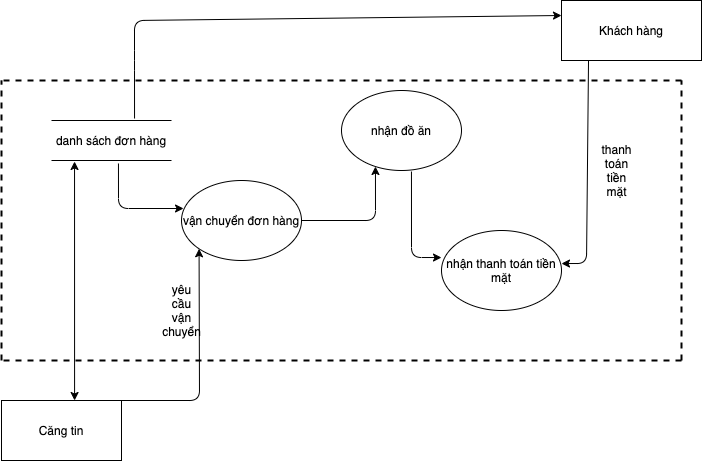
Hình vẽ 2.1.2.3.1

* Thanh toán đơn hàng



Hình vẽ 2.1.2.3.2

* Giao hàng



Hình vẽ 2.1.2.3.3

* + 1. Đặc tả tiến trình nghiệp vụ
       1. Đặt hàng
* Tìm kiếm sản phẩm
  + Đầu vào: từ khoá tên sản phẩm, loại sản phẩm, hoặc ngày trong tuần
  + Đầu ra: danh sách sản phẩm đã đươc thêm vào giỏ hàng
  + Nội dung xử lý:

Nếu: tìm thấy sản phẩm

Thì: hiển thị danh sách sản phẩm

Nếu: không tìm thấy sản phẩm

Thì: không hiển thị sản phẩm nào

* Thêm vào giỏ hàng:
  + Đầu vào: sản phẩm
  + Đầu ra: danh sách sản phẩm đã được thêm vào giỏ hàng
  + Nội dung xử lý:

Lặp:

Nếu: sản phẩm có số lượng còn lại lớn hơn 0

Thì: thêm thành công sản phẩm vào giỏ hàng

Nếu: sản phẩm đã có trong giỏ hàng

Thì: tăng số lượng sản phẩm lên 1 đơn vị tính và thông báo thêm tình công sản phẩm vào giỏ hàng

Nếu: số lượng sản phẩm bằng 0

Thì: thông báo không thể thêm sản phẩm vào giỏ hàng

* + - 1. Thanh toán đơn hàng
* Thanh toán
  + Đầu vào: loại thanh toán, thông tin giỏ hàng
  + Đầu ra: kết quả thanh toán hoá đơn
  + Nội dung xử lý:

Nếu: người dùng chưa đăng nhập

Thì: yêu cầu ngừoi dùng đăng nhập và chuyển qua màn hình đăng nhập

Nếu: người dùng đã đăng nhập

Thì: người dùng được phép thanh toán

Nếu: loại thanh toán là “thanh toán bằng tiền mặt”

Thì: đơn hàng được chuyển sang trạng thái chờ giao dịch, và có trạng thái thanh toán là “thanh toán bằng tiền mặt”

Nếu: loại thanh toán là: “thẻ tín dụng”

Thì: hệ thống mở đến màn hình cổng thanh toán trực tuyến được liên kết với ví điện tử, internet banking của VNPAY

Nếu: khởi tạo dao dịch thanh toán online thành công

Thì: người dùng chọn một trong ba hình thức thanh toán là: thẻ nội địa, thẻ quốc tế, ví điện tử VNPAY.

Lặp: người dùng nhập các thông tin cần thiết của thẻ

Nếu: người dùng nhập đúng thông tin

Thì: chuyển qua bước nhập mã OTP được gửi về số điện thoại liên kết với tài khoản ngân hàng

Nếu người dùng nhập sai thông tin

Thì: người dùng phải nhập lại thông tin chính xác

Đến khi: người dùng xác nhan OTP thành công

Thì: giao dịch thành công, hệ thống thông báo thanh toán thành công, đơn hàng được chuyển sang trạng thái “chờ xác nhận”, và trạng thái thanh toán là “đã thanh toán”, giỏ hàng được khởi tạo lại

Nếu: người dùng chọn huỷ thanh toán

Thì: hệ thống báo thanh toán thất bại, giữ nguyên giỏ hàng

* Xác nhận đơn hàng
  + Đầu vào: danh sách đơn hàng
  + Đầu ra: danh sách đơn hàng đã được chuyển trạng thái thành “đang chờ vận chuyển”
  + Nội dung xử lý:

Lặp:

Nếu: đơn hàng đang ở trạng thái chờ xác nhận

Thì: chuyển đơn hàng sang trạng thái “chờ vận chuyển”

Đến khi: hết các đơn hàng muốn xác nhận

* + - 1. Giao hàng
  + Đầu vào: hoá đơn, sản phẩm đã được chế biến
  + Đầu ra: sản phẩm được giao đến người dùng và nhận lại tiền thanh toán nếu ngừoi dùng chọn thanh toán bằng tiền mặt
  + Nội dung xử lý:

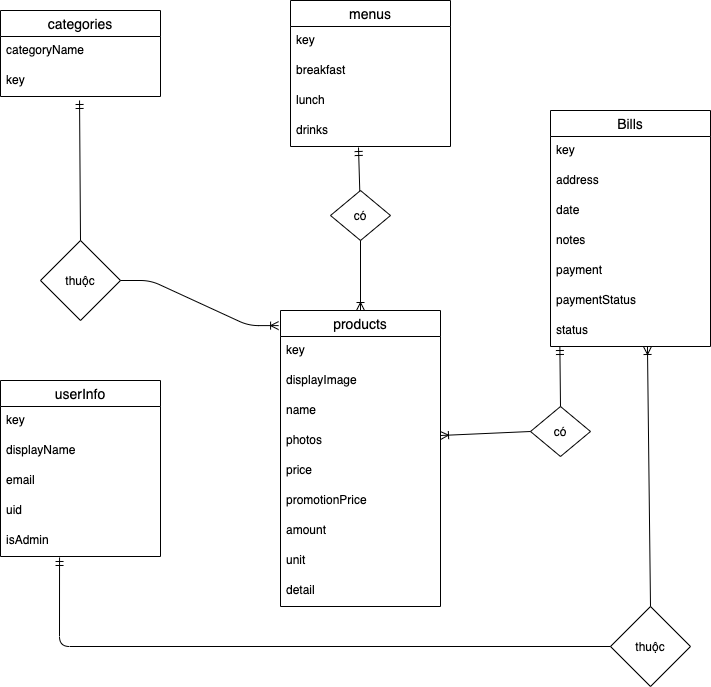
Nếu : người dùng chọn thanh toán bằng thẻ tín dụng

Thì: giao sản phẩm đến nơi nhận hàng và đổi trạng thái đơn hàng thành “hoàn thành”

Nếu: người dùng chọn thanh toán bằng tiền mặt

Thì: giao sản phẩm đến nơi nhận và nhận tiền thanh toán, sau đó đổi trạng thái đơn hàng thành “hoàn thành”.

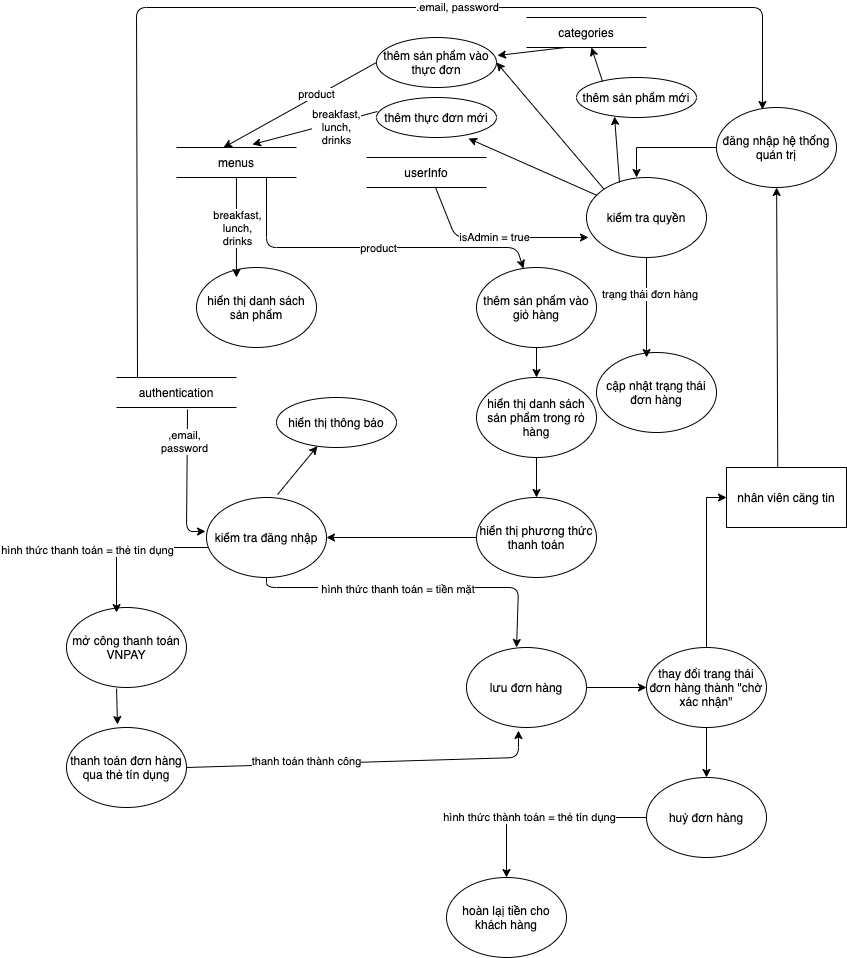
* 1. Phân tích dữ liệu nghiệp vụ
     1. Mô hình dữ liệu ban đầu



# Chương 3. Thiết kế hệ thống

## Thiết kế tổng thể

## DFD hệ thống



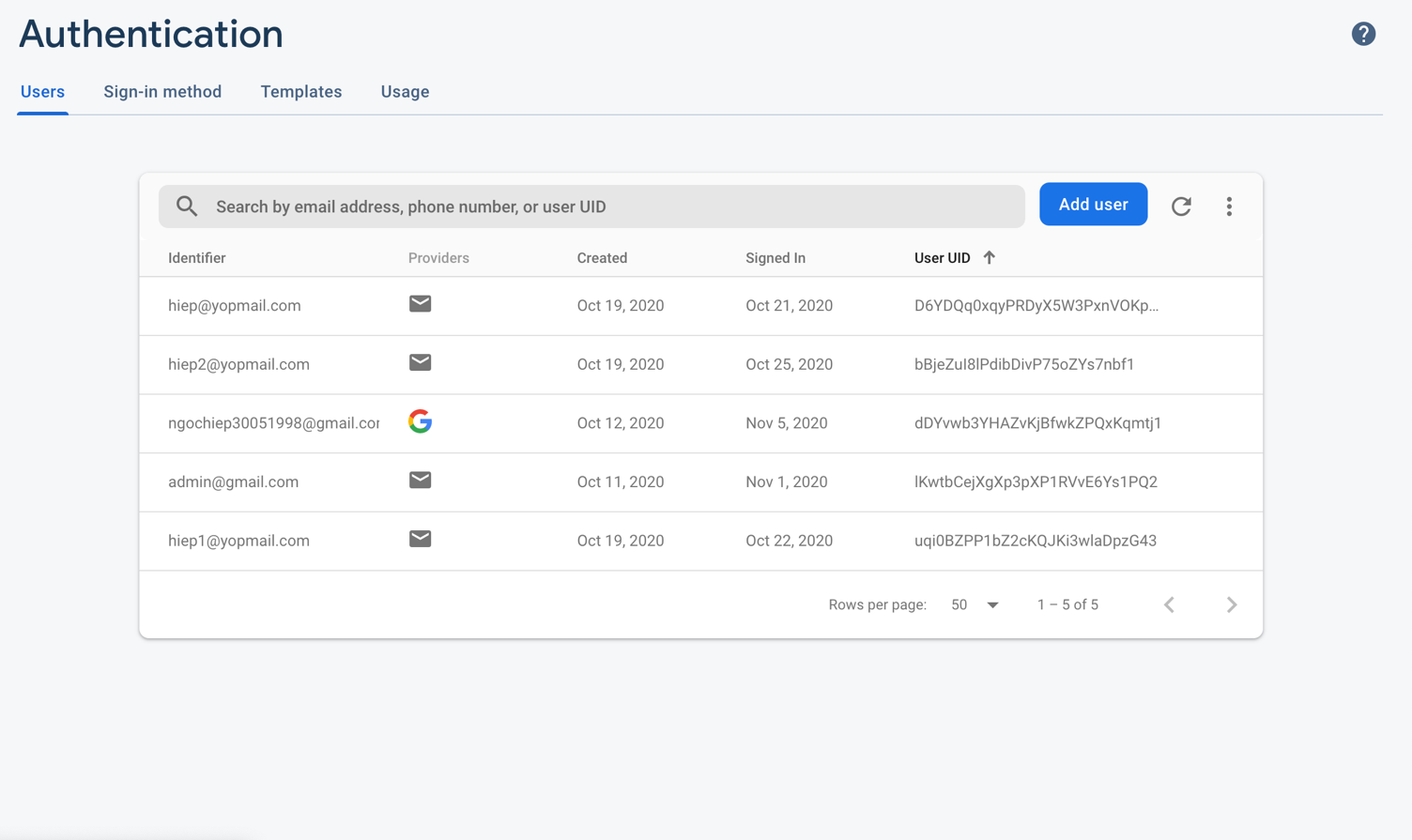
## Đặc tả tiến trình hệ thống

* + - 1. Tiến trình “hiển thị danh sách sản phẩm”
  + Đầu vào: Mã thực đơn
  + Đầu ra: Danh sách các sản phẩm thuộc bữa sáng, bữa trưa, đồ uống
  + Xử lý:
    - Người dùng chọn ngày muốn xem thực đơn ở trên thanh công cụ
    - Hệ thống lấy được mã thực đơn từ ngày người dùng chọn
    - Hệ thống lấy danh sách sản phẩm theo mã thực đơn bao gồm: số lượng, đơn giá, giảm giá, tên sản phẩm, ảnh sản phẩm. Sau đó hiển thị các sản phẩm tương ứng với 3 đầu mục bữa sáng, bữa trưa, đồ uống
      1. Tiến trình “thêm sản phẩm vào giỏ hàng”
  + Đầu vào: sản phẩm
  + Đầu ra: sản phẩm được thêm vào rỏ hàng
  + Xử lý:
* Người dùng tìm kiếm sản phẩm muốn thêm vào giỏ hàng và bấm nút “thêm vào rỏ hàng”
* Lặp : hệ thống kiểm tra xem sản phẩm người dùng chọn còn hàng không.
  + Nếu sản phẩm người dùng chọn hết hàng
  + Thì hiển thị thong báo hết hàng
  + Nếu sản phẩm còn hàng
    - * Lặp:
    - Kiểm tra xem sản phẩm đã có trong giỏ hàng chưa
    - Nếu có: tăng số lượng sản phẩm lên 1 đơn vị
    - Nếu không có: thêm sản phẩm mới vào giỏ
      1. Tiến trình “hiển thị danh sách sản phẩm trong giỏ hàng”
* Đầu vào: giỏ hàng
* Đầu ra: danh sách sản phẩm, số lượng sản phẩm, tổng giá trị đơn hàng
* Xử lý:
  + Người dùng bấm vào icon giỏ hàng trên góc màn hình
  + Hệ thống lấy danh sách sản phẩm trong giỏ hàng từ bộ nhớ tạm của điện thoại để hiển thị lên màn hình
  + Nếu danh sách sản phẩm có số lượng lớn hơn 0: hiển thị danh sách sản phẩm bảo gồm: tên sản phẩm, tên loại sản phẩm, đơn giá, giá khuyến mại, số lượng, ảnh sản phẩm và tổng số lượng, tổng giá tiền của đơn hàng
  + Nếu danh sách sản phẩm trong giỏ bằng 0: hiển thị nút quay về trang chủ
    - 1. Tiến trình “thanh toán”
* Đầu vào: danh sách sản phẩm trong giỏ hàng, loại thanh toán, nội dung thanh toán
* Đầu ra: đơn hàng được thanh toán thành công
* Xử lý:
  + Người dùng chọn hình thức thanh toán
  + Nếu hình thức thanh toán là “tiền mặt”: hệ thống lưu đơn hàng, chuyển trạng thái đơn hàng thành “chờ xác nhận”, trạng thái thanh toán của đơn hàng là “chưa thanh toán”, hệ thống xoá giỏ hàng và quay về trang chủ
  + Nếu hình thức thanh toán là “thẻ tín dụng”:
  + Nếu hình thức thanh toán là “thẻ tín dụng. :ứng dụng chuyển sang màn hình chọn phương thức thanh toán:
    - Thẻ ATM và tài khoản ngân hàng
    - Thẻ thanh toán quốc tế
    - Ví điện tử
  + Người dùng chọn hình thức thanh toán “thẻ ATM và tài khoản ngân hàng”, sau đó chọn ngân hàng:
    - ứng dụng chuyển sang màn hình nhập các thông tin sau:
      * Số thẻ
      * Ngày phát hành thẻ
      * Tên chủ thẻ
    - Người dùng chọn nút xác thực: ứng dụng chuyển qua màn hình nhập mã sms otp đã được gửi về tin nhắn qua số điện thoại được liên kết với tài khoản ngân hàng tương ứng
    - Người dụng bấm nút “Xác nhận”: đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính
  + Người dùng chọn thanh toán qua “thẻ thanh toán quốc tế” và chọn 1 loại thẻ:
    - Ứng dụng chuyển qua màn nhập các thông tin sau:
      * Số thẻ
      * Ngày hết hạn
      * Số CCV/CVV
      * Họ tên khách hàng
      * Email
      * Địa chỉ
      * Tỉnh/Thành phố
      * Đất nước
    - Người dùng chọn xác thực: đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính.
  + Người dùng chọn thanh toán qua ví điện tử VNPAY:
    - Ứng dụng chuyển qua màn hình hiển thị mã QR code
    - Người dùng bật ứng dụng “Ví điện tử VNPAY” hoặc bất kì ứng dụng smart banking nào được cung cấp dịch vụ QR code từ VNPAY(ví dụ như vietcombank, Agribank, BIDV,…) để quét mã QR
    - Sau khi quét mã QR người dùng thực hiện xác thực yêu cầu trên các ứng dụng tương ứng, đơn hàng đã được thanh toán thành công và chuyển sang trạng thái chờ xác nhận, ứng dụng sẽ trở về trang chính.
      1. Tiến trình “lưu đơn hàng”
* Đầu vào: danh sách sản phẩm trong giỏ hàng
* Đầu ra: đơn hàng được lưu vào database với trạng thái là “chờ xác nhận”
* Xử lý:
  + Hệ thống lấy thông tin được gửi lên từ phía người dùng bao gồm: danh sách sản phẩm, thông tin người dùng(số điện thoại, email,tên người dùng, mã người dùng)
  + Nếu hình thức thanh toán là “tiền mặt” thì lưu thông tin đơn hàng với trạng thái là “chờ xác nhận” và trạng thái thanh toán là “chưa thanh toán”
  + Nếu hình thức thanh toán là “thẻ tín dụng” thì lưu thông tin đơn hàng với trạng thái là “chờ xác nhận” và trạng thái thanh toán là “đã thanh toán”
    - 1. Tiến trình “Huỷ đơn hàng”
* Đầu vào: yêu cầu huỷ đơn hàng
* Đầu ra: yêu cầu được thực hiện, đơn hàng chuyển sang trạng thái “đã huỷ”
* Xử lý:
  + Nếu đơn hàng được yêu cầu huỷ do người dùng không muốn đặt hàng nữa:
    - Người dùng vào lịch sử đơn hàng
    - Bấm vào nút “huỷ đơn hàng”
    - Đơn hàng sẽ được huỷ
  + Nếu sản phẩm trong đơn hàng đã hết hoặc không đủ số lượng:
    - Nhân viên căng tin gọi điện cho khách hàng để thông báo sản phẩm đã hết hàng
    - Người dùng vào trang lịch sử đơn hàng
    - Bấm vào nút “huỷ đơn hàng”
    - Đơn hàng sẽ được huỷ
  + Nếu đơn hàng không phải ở trạng thái “chờ xác nhận”:
    - Đơn hàng sẽ không được phép huỷ
      1. Tiến trình “Hoàn lại tiền cho khách hàng”
* Đầu vào: chi tiết đơn hàng
* Đầu ra: tiền được gửi lại vào tài khoản cho khách hàng
* Xử lý:
  + Nhân viên căng tin lấy chi tiết đơn hàng đã bị huỷ
  + Nếu trạng thái thanh toán của đơn hàng là đã thanh toán
    - Nhân viên lấy thông tin đơn hàng từ hệ thống website quản trị
    - Nhân viên đăng nhập vào hệ thống <https://sandbox.vnpayment.vn/merchantv2/Home/Dashboard.htm> của VNPAY cung cấp cho đối tác để tìm giao dịch cần hoàn tiền
    - Chọn giao dịch cần hoàn tiền
    - Hệ thống sẽ chuyển sang màn hình hoàn tiền
    - Nhân viên nhập các thông tin cần thiết và bấm nút “hoàn tiền”
    - Hệ thống VNPAY sẽ gửi lại tiền đã thanh toán vào tài khoản của khách hàng
    - Hệ thống thay đổi trạng thái thanh toán của đơn hàng thành “đã hoàn tiền”
      1. Tiến trình “Kiểm tra đăng nhập”
* Đầu vào: email, password
* Đầu ra: trạng thái đăng nhập trên ứng dụng mobile app
* Xử lý:
  + Nếu người dùng chọn đăng nhập bằng tài khoản google
    - Mở trang đăng nhập bằng tài khoản google
    - Chọn tài khoản google
    - Xác thực tài khoản
  + Nếu người dùng đăng nhập bằng email, password
    - Lấy dữ liệu từ form
    - Kiểm tra xác thực với hệ thống
  + Thông báo đăng nhập thành công hoặc thất bại
    - 1. Tiến tình “thêm mới sản phẩm”
* Đầu vào: thông tin sản phẩm
* Đầu ra: sản phẩm được thêm vào hệ thống
* Xử lý:
  + Lấy thông tin từ form do người dùng nhập vào từ website quản trị
  + Thêm thêm các thông tin vừa lấy được vào bảng “categorires”
    - 1. Tiến trình “thêm mới thực đơn”
* Đầu vào: thông tin thực đơn
* Đầu ra: thực đơn mới được thêm vào hệ thống
* Xử lý:
  + Lấy thông tin thực đơn, sản phẩm được chọn vào thực đơn
  + Thêm thực đơn gồm các sản phẩm được chọn vào bảng “menus”
  1. Thiết kế kiểm soát
     1. Thiết kế nhóm người dùng
* Dựa vào cơ cấu tổ chức của hệ thống gồm có 2 nhóm người dùng là:
  + Nhóm 1: nhóm người dùng ứng dụng mobile app
  + Nhóm 2: nhóm nhân viên căng tin
* Nhóm quản trị gồm: quản lý đơn hàng, quản lý sản phẩm, quản lý thực đơn, quản lý người dùng
* Mỗi người dùng có quyền thay đổi mật khẩu và thông tin cá nhân của mình, trừ “tên đăng nhập”
  + 1. Phân hạn nhóm người dùng về tiến trình

Ký hiệu: A: Active; not A: Not Active

|  |  |  |  |
| --- | --- | --- | --- |
| Tiến trình |  | | |
| Quản trị | Người dùng |
| Tìm kiếm sản phẩm | Not A | A |
| Thêm sản phẩm vào rỏ hàng | Not A | A |
| Thanh toán đơn hàng | Not A | A |
| Đăng ký | Not A | A |
| Đăng nhập | A | A |
| Xác nhận đơn hàng | A | Not A |
| Lập hoá đơn | A | A |
| Huỷ đơn hàng | Not A | A |
| Vận chuyển đơn hàng | A | Not A |
| Quản lý sản phẩm | A | Not A |
| Quản lý thực đơn | A | Not A |
| Phân quyền người dùng | A | Not A |

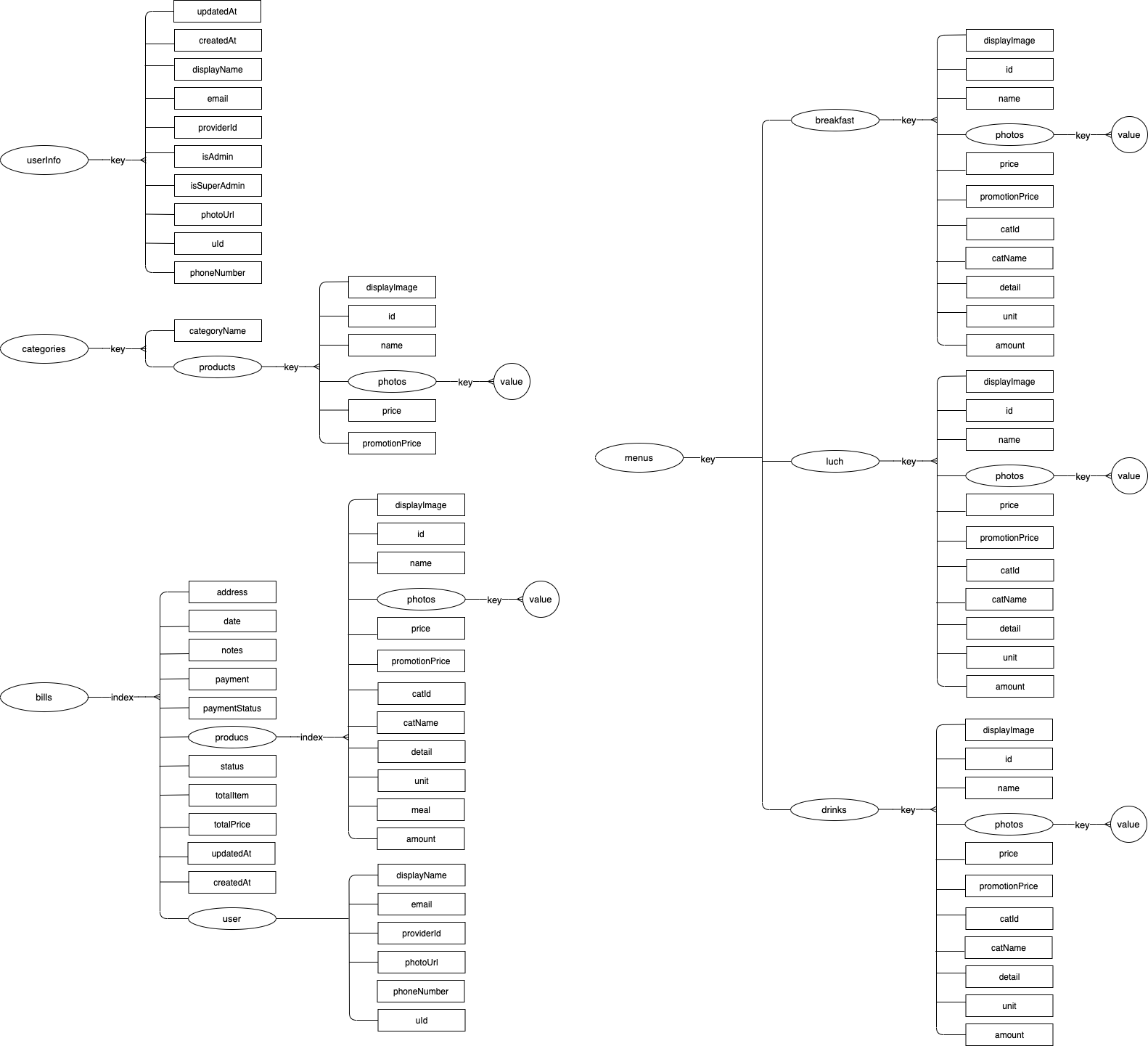
* 1. Thiết kế cơ sở dữ liệu
     1. Thiết kế dữ liệu phục vụ bảo mật
* Xác định thực thể phục vụ bảo mật
* Khi người dùng muốn sử dụng hệ thống thì phải đăng nhập vào theo email và mật khẩu được cho phép. Hệ thống sẽ kiểm tra tính hợp lệ của tên và mật khẩu người dùng và quản trị cùng với quyền của họ. Dựa vào quyền người sử dụng mà hệ thống lọc ra những module mà người đó được dùng.
* Mỗi người dùng sẽ được phân 1 nhóm quyền nhất định
  + Ví dụ: nhân viên phụ trách tính lương sẽ có quyền thực hiện, nhìn thấy các menu thêm, sửa, xóa nhưng thông tin liên quan đến việc tính lương, còn nhân viên bộ phận khác không thể nhìn thấy những menu đó.
* Sử dụng phương thức xác thực của firebase cung cấp kết hợp với node userInfo được thiết kế trong JSON Tree của firebase realtime data base
* Mô hình dữ liệu phục vụ xác thực của firebase:



* Node “userInfo”



* + 1. Mô hình dữ liệu hệ thống



* + 1. Đặc tả dữ liệu
       1. Node “userInfo”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | displayName | String | Tên hiển thị của người dùng |
| 2 | email | String | Email người dùng |
| 3 | providerId | String | Mã phương thức xác thực (google, facebook,…) |
| 4 | isAdmin | Boolean | Trường dữ liệu xác định xem người dùng có phải admin không |
| 5 | phoneNumber | String | Số điện thoại |
| 6 | isSuperAdmin | Boolean | Kiểm tra xem có phải sup |
| 7 | photoUrl | String | Đường dẫn ảnh đại diện |
| 8 | uId | String | Mã người dùng |
| 9 | createdAt | timestamp | Thời gian tạo |
| 10 | updatedAt | timestamp | Thời gian cập nhật |

* + - 1. Node “Categories”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | categoryName | String | Tên loại sản phẩm |
| 2 | products | Object | Dan sách sản phẩm nằm trong categories |

* Node “products”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | displayImage | String | Đường dẫn dến ảnh hiển thị |
| 2 | id | String | Mã sản phẩm |
| 3 | name | String | Tên sản phẩm |
| 4 | price | Number | Đơn giá sản phẩm |
| 5 | promotionPrice | Number | Giá khuyến mại của sản phẩm |
| 6 | photos | Object | Lưu trữ đường dẫn của ảnh chi tiết sản phẩm dưới dạng object :  {Key: value} |

* + - 1. Node “menus”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | breakfast | Object | Lưu danh sách sản phẩm trong thực đơn ăn sáng |
| 2 | luch | Object | Lưu danh sách sản phẩm trong thực đơn ăn trưa |
| 3 | drinks | Object | Lưu danh sách sản phẩm trong thực đơn đồ uống |

* Node con của “breakfast”, “lunch”, “drinks”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | displayImage | String | Đường dẫn dến ảnh hiển thị |
| 2 | id | String | Mã sản phẩm |
| 3 | name | String | Tên sản phẩm |
| 4 | price | Number | Đơn giá sản phẩm |
| 5 | promotionPrice | Number | Giá khuyến mại của sản phẩm |
| 6 | photos | Object | Lưu trữ đường dẫn của ảnh chi tiết sản phẩm dưới dạng object :  {Key: value} |
| 7 | amount | Number | Số lượng sản phẩm có trong thực đơn |
| 8 | catId | String | Mã loại sản phẩm |
| 9 | catName | String | Tên loại sản phẩm |

* + - 1. Node “bills”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | address | String | Địa chỉ nhận hàng, ví dụ (tầng 3) |
| 2 | date | timestamp | Thời gian đặt hàng |
| 3 | notes | String | Ghi chú của người dùng |
| 4 | payment | String | Hình thức thanh toán  - “1”: thanh toán tiền mặt  - “2”: thanh toán online |
| 5 | paymentStatus | String | Trạng thái thanh toán  - “0”: chưa thanh toán  - “1”: đã thanh toán  - “2”: đã hoàn lại tiền |
| 6 | status | string | Trạng thái đơn hàng  - “0”: chờ xác nhận  - “1”: đã xác nhận  - “2”: đang vận chuyển  - “3”: đã giao hàng  - “4”: đã huỷ |
| 7 | totalItem | Number | Tổng số lượng sản phẩm trong đơn hàng |
| 8 | totalPrice | Number | Tổng giá tiền của đơn hàng |
| 9 | user | Object | Thông tin khách hàng |
| 10 | products | array | Danh sách sản phẩm trong đơn hàng |
| 11 | vnpayTransId | String | Mã giao dịch trực tuyến thông qua cổng thanh toán vnpay |
| 12 | id | String | Mã đơn hàng |
| 13 | updatedAt | timestamp | Thời gian cập nhật đơn hàng |

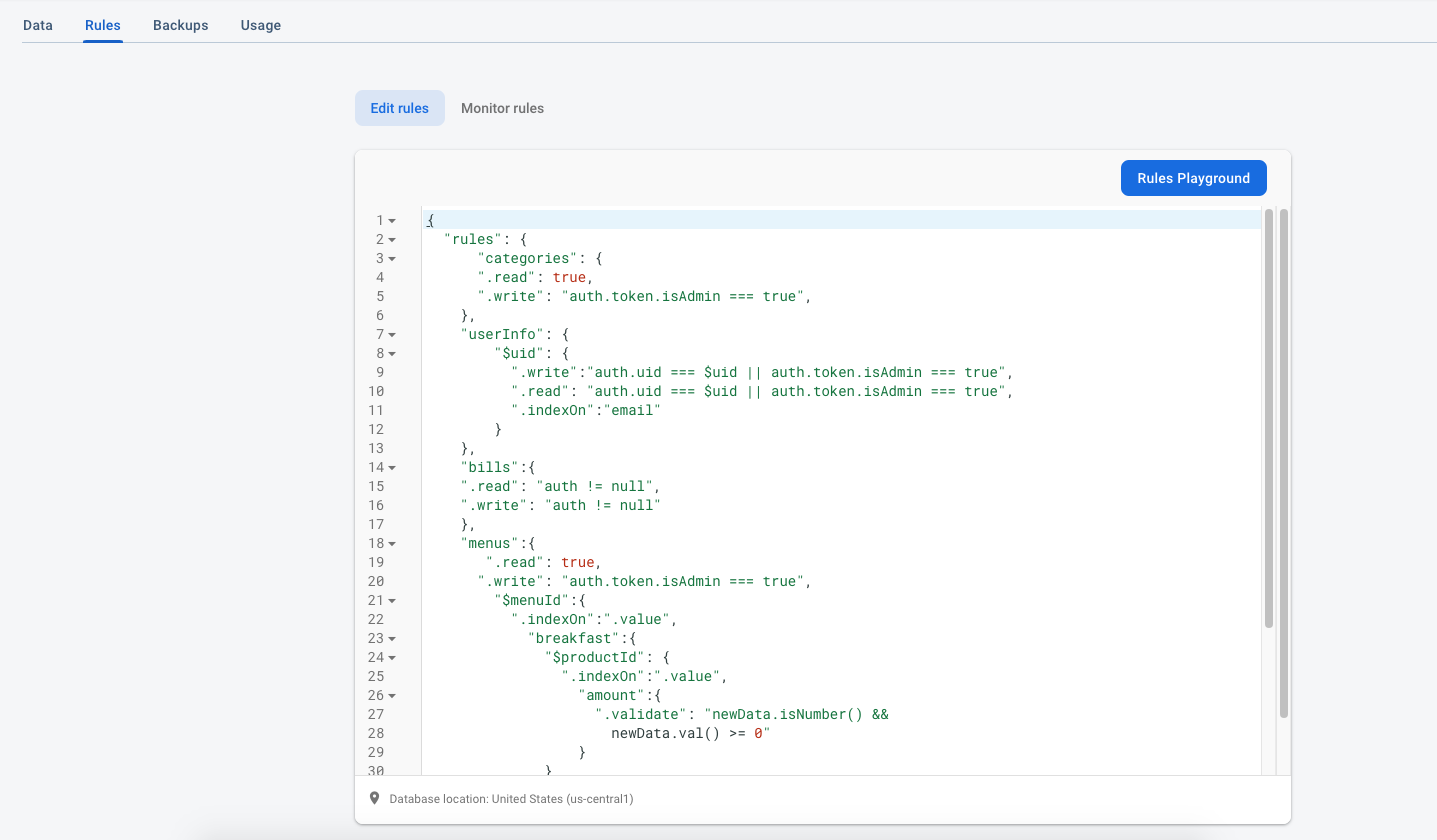
* Node “products”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | displayImage | String | Đường dẫn dến ảnh hiển thị |
| 2 | id | String | Mã sản phẩm |
| 3 | name | String | Tên sản phẩm |
| 4 | price | Number | Đơn giá sản phẩm |
| 5 | promotionPrice | Number | Giá khuyến mại của sản phẩm |
| 6 | photos | Array | Lưu trữ đường dẫn của ảnh chi tiết sản phẩm dưới dạng mảng gồm các object dạng:  {  key: string;  value: string } |
| 7 | amount | Number | Số lượng sản phẩm có trong thực đơn |
| 8 | catId | String | Mã loại sản phẩm |
| 9 | catName | String | Tên loại sản phẩm |
| 10 | Meal | String | Sản phẩm thuộc danh mục nào trong thực đơn:bữa sáng, bữa trưa, đồ uống |

* Node “user”

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | displayName | String | Tên hiển thị của người dùng |
| 2 | email | String | Email người dùng |
| 3 | providerId | String | Mã phương thức xác thực (google, facebook,…) |
| 4 | phoneNumber | String | Số điện thoại |
| 5 | photoUrl | String | Đường dẫn ảnh đại diện |
| 6 | uId | String | Mã người dùng |

* + 1. Quy tắc cho cơ sở dữ liệu
* Firebase realtime database cung cấp phương thức bảo mật dữ liệu thông qua rules đươc cài đặt và cấu hình tuỳ chỉnh



* Cấu hình rules:

{

"rules": {

"categories": {

".read": true,

".write": "auth.token.isAdmin === true",

},

"userInfo": {

"$uid": {

".write":"auth.uid === $uid || auth.token.isAdmin === true",

".read": "auth.uid === $uid || auth.token.isAdmin === true",

".indexOn":"email"

}

},

"bills":{

".read": "auth != null",

".write": "auth != null"

},

"menus":{

".read": true,

".write": "auth.token.isAdmin === true",

"$menuId":{

".indexOn":".value",

"breakfast":{

"$productId": {

".indexOn":".value",

"amount":{

".validate": "newData.isNumber() &&

newData.val() >= 0"

}

}

},

"lunch":{

"$productId": {

".indexOn":".value",

"amount":{

".validate": "newData.isNumber() &&

newData.val() >= 0"

}

}

},

"drinks":{

"$productId": {

".indexOn":".value",

"amount":{

".validate": "newData.isNumber() &&

newData.val() >= 0"

}

}

}

}

}

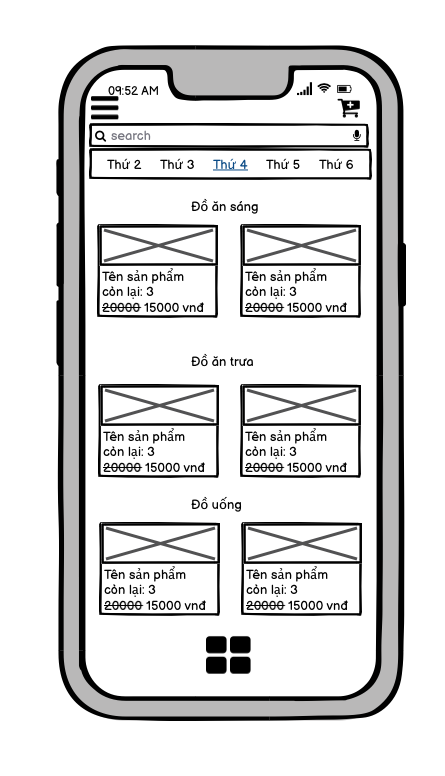
}

}

* Đặc tả cấu hình rules:

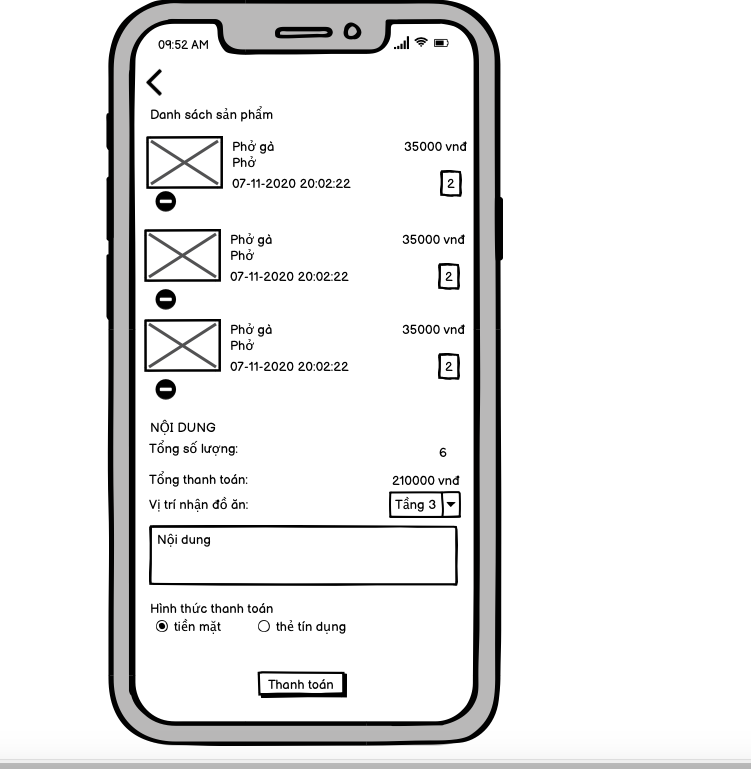
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Node** | **Super admin** | **Admin** | **Khách hàng** | **Đặc tả** |
| 1 | categories | Đọc, thêm, sửa, xoá | Đọc, thêm, sửa, xoá | Đọc |  |
| 2 | userInfo | Đọc, thêm, sửa, xoá | Đọc, thêm, sửa | Đọc, thêm, sửa, xoá | Tất cả các tác nhân muốn thao tác mới node đều phải đăng nhập hệ thống.  Super dmin được phép phân quyền cho người dùng thành admin hoặc hạ quyền từ admin thành người dùng.  Người dùng chỉ được phép đọc, ghi thông tin của chính mình |
| 3 | bills | Đọc, thêm, sửa | Đọc, thêm, sửa | Đọc, thêm, sửa | Tất cả các tác nhân muốn thao tác với node đều phải đăng nhập hệ thống |
| 4 | menus | Đọc, thêm, sửa,  Xoá | Đọc, thêm, sửa,  Xoá | Đọc | Đối với node con: “amount” điều kiện để thực hiện thao tác thêm, sửa là giá trị phải lớn hơn hoặc bằng 0 |

* 1. Thiết kế giao diện
     1. Giao diện mobile app
        1. Trang chủ



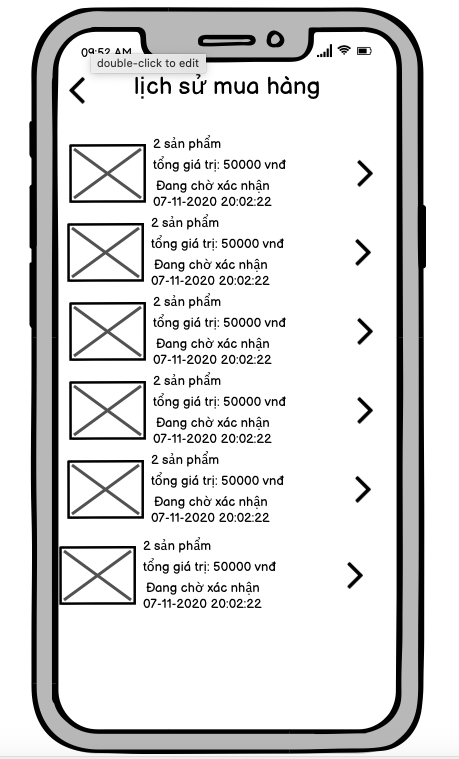
Hình vẽ 3.4.1.1

* + - 1. Giỏ hàng



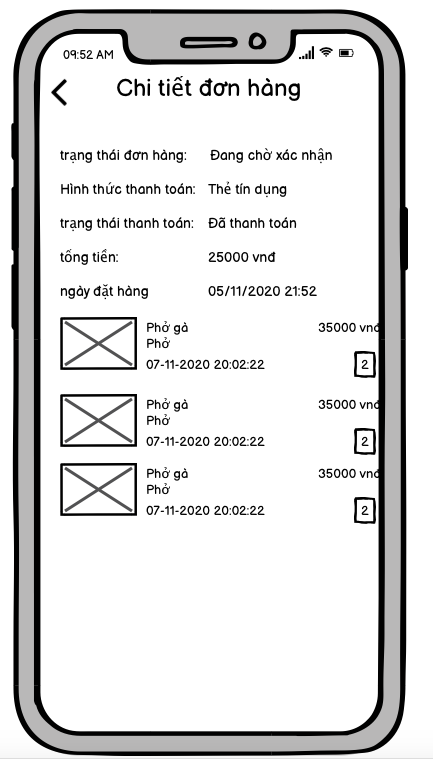
Hình vẽ 3.4.1.2

* + - 1. Lịch sử đơn hàng



Hình vẽ 3.4.1.3

* + - 1. Chi tiết lịch sử đơn hàng



Hình vẽ 3.4.1.4

# Chương 4. Xây dựng chương trình

## Mô tả hệ thống

### Ngôn ngữ sử dụng

* + - 1. Ngôn ngữ sử dụng chính: javascript
* Giới thiệu:
  + Javascipt là gì: Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

* + Một số thư viện, framework nổi tiếng hiện nay:
    - Angular: Thực tế angular 2+ được xây dựng dựa trên nền tảng typescript, nhưng về bản chất thì typescript cũng được phát triển lên từ chính javascript.
    - Nodejs
    - Reactjs
    - Vuejs
    - Jquery
    - Nestjs
      1. Ngôn ngữ sử dụng phía frontend
         1. Mobile app : framwork Ionic 4
* Trang chủ: <https://ionicframework.com/docs>
* Giới thiệu:

Sau một thời gian làm việc với nền tảng mobile, tôi muốn giới thiệu với các bạn về 1 hybrid framework để phát triển ứng dụng trên mobile

Ionic là một hybrid Framework được sử dụng để phát triển các ứng dụng di động dựa trên nền tảng công nghệ web HTML ( sự kết hợp giữa Angular và Cordova), được tạo bởi Max Lynch, Ben Sperry, và Adam Bradley vào 2013. Đây là một framework rất mạnh để viết các ứng dụng hybrid. Nó khắc phục các nhược điểm của Native app và Mobile webapp và kết hợp được sức mạnh của 2 nền tảng đó.

**\*) Ưu điểm và nhược điểm**

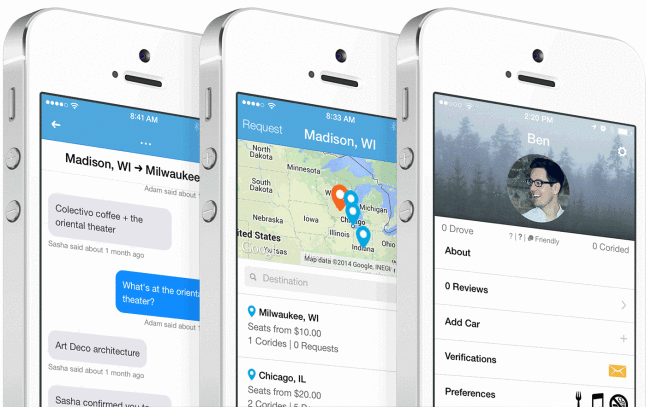
**a. Ưu điểm**

– Dễ học, thời gian phát triển nhanh, có thể sử dụng các kỹ năng từ lập trình web  
– Đa nền tảng  
– Khả năng truy cập đến các tính năng của thiết bị và hệ điều hành như bluetooth, camera,..  
– Dễ dàng thiết kế giao diện cho các thiết bị có kích cỡ khác nhau  
– Việc sử dụng AngularJS làm core giúp phần xử lý UI linh động hơn so với javasript hay thư viện Jquery.

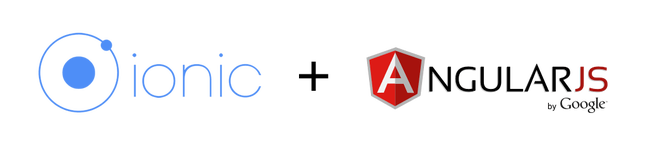
– Việc sử dụng AngularJS làm core cũng mang lại lợi thế lớn so với các framework cho ứng dụng hybrid khác.  
– Ionic cung cấp đầy đủ các thành phần trong giao diện người dùng như Pull-to-Refresh, Infinite-loader, tabs, ..

**b. Nhược điểm**

– Vẫn còn trong giai đoạn phát triển  
–  Hiệu năng vẫn chưa cao và ổn định  
– Cộng đồng phát triển ứng dụng vẫn còn chưa đông

[](https://convoicoi.files.wordpress.com/2016/01/ionic-hybrid-design.png)

* **Tính năng**
* **Pairing with AngularJS**

[](https://convoicoi.files.wordpress.com/2016/01/ionic-angular-opti.png)Ionic sử dụng AngularJS để cung cấp các cấu trúc ứng dụng, trong khi bản thân Ionic tập trung chính vào giao diện người dùng. Nói cách khác, chúng ta thấy được sự phối hợp ăn ý giữa sức mạnh của AngularJS và vẻ đẹp của Ionic UI.

Ionic cung cấp một tập các Angular directives (nghĩa là các phần tử HTML tùy biến) để làm các thành phần của nó, tạo ra sự dễ dàng để sử dụng các tiện ích gọn để viết mã HTML. Ngoài các directives, Ionic còn sử dụng và thêm vào các thành phần khác như: Angular touch recognizers, view animation logic, HTML sanitation, và asynchronous communication.

Việc xây dựng ứng dụng dựa trên AngularJS đòi hỏi mã nguồn phải có khả năng mở rộng cao để bổ sung các tính năng mới. Tuy nhiên với Ionic, người ta có thể tái sử dụng các chức năng trong ứng dụng trên các nền tảng khác nhau đồng thời vẫn có thể tùy chỉnh giao diện người dùng cho mỗi nền tảng riêng biệt. Các thành phần trong Ionic như danh sách, slide,.. chính là các directive(các thuộc tính của thẻ HTML dùng trong Angular) của AngularJS. Đó là lí do khiến cho Ionic và AngularJS kết hợp rất tốt với nhau.

Mặc dù, Ionic có thành phần giao diện sử dụng Angular, nhưng developers vẫn có thể sử dụng và hỗ trợ các framework khác như Knockout hay EmberJS.

* **Cordova**

[](https://convoicoi.files.wordpress.com/2016/01/apache_cordova1.png)Apache Cordova là một bộ khung để xây dựng ứng dụng di động sử dụng HTML, CSS và Javascript. Apache Cordova bao gồm một tập hợp các API thiết bị cho phép người lập trình di động truy cập, sử dụng các chức năng native của thiết bị như là camera hay cảm biến gia tốc bằng Javascript. Kết hợp với một bộ khung phát triển giao diện như jQuery Mobile or Dojo Mobile hoặc Ionic, cho phép ứng dụng di động có thể được phát triển chỉ dựa trên HTML, CSS và Javascript.

Khi sử dụng Cordova API, một ứng dụng có thể được xây dựng mà không phải sử dụng bất kỳ một đoạn mã native code nào. Thay vào đó, công nghệ web sẽ được sử dụng, và chúng sẽ được tổ chức trên chính ứng dụng đấy chứ không cần thông qua một server nào.

Cordova cung cấp một tập hợp các thư viện Javascript đã được chuẩn hóa để có thể sử dụng. Cordova hiện có thể sử dụng cho các nền tảng như iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada và Symbian.

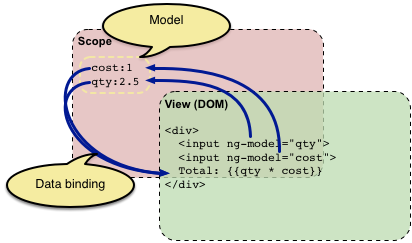
Kể từ phiên bản 3.0, bạn có thể chọn một trong hai quy trình để phát triển một ứng dụng di động.

Cross-platform (CLI) workflow: được sử dụng khi bạn muốn xây dựng ứng dụng trên càng nhiều nền tảng càng tốt, mà ít cần quan tâm đến đặc trưng của nền tảng. Trung tâm của quy trình này là CLI (Command-Line Interface), được giới thiệu từ phiên bản Cordova 3.0. CLI là một công cụ ở mức cao cho phép bạn xây dựng dự án trên nhiều nền tảng một lần duy nhất, trừu tượng hóa đi nhiều chức năng ở mức thấp. CLI cũng cung cấp một giao diện chung để sử dụng các plug-in vào ứng dụng của bạn.

Platform-centered workflow:  được sử dụng khi bạn muốn tập trung phát triển ứng dụng trên một nền tảng cụ thể và muốn can thiệp sâu vào mức thấp ví dụ như khi bạn muốn kết hợp một thành phần native đã được tùy biến vào một thành phần web của Cordova. Quy trình này phụ thuộc vào một tập các đoạn mã ở mức thấp được thiết kế riêng cho mỗi nền tảng được hỗ trợ. Vì vậy khi sử dụng cách phát triển này sẽ có thể gặp nhiều khó khăn hơn bởi vì sự thiếu hụt của những công cụ ở mức cao nhưng bạn sẽ có được sự tự do hơn trong việc phát triển ứng dụng.

* **Mô hình phát triển ứng dụng MV\* Pattern**

Angular JS được sử dụng trong Ionic là một MVC framework, nhưng theo thời gian thì nó trở nên thân thuộc hơn với khái niệm MVVM trong đó đối tượng $scope hoạt động như một ViewModel, sử dụng và thao tác trong các Controller.

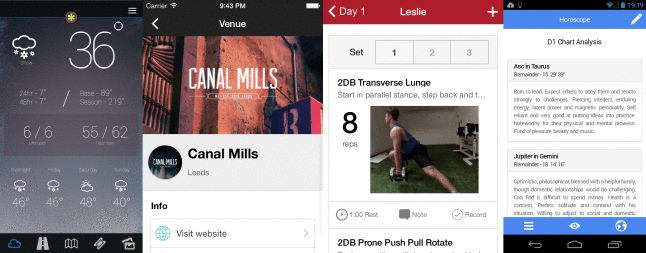
[](https://convoicoi.files.wordpress.com/2016/01/angularjs-mvvm-pattern.png)Các developer có thể linh hoạt sử dụng MVC hoặc là MVVM theo ý thích của họ, miễn là đạt được mục tiêu: Việc tách phần trình bày giao diện khỏi tầng nghiệp vụ logic thúc đẩy hiệu suất phát triển sản phẩm và bảo trì.

**Thao tác với DOM**

Về khả năng tương tác với các thành phần DOM của webView. Angular nhúng jqLite, là một phần nhỏ gọn của thư viện jQuery cho phép thao tác DOM trong các trình duyệt theo cách tương thích, và không phụ thuộc vào trình duyệt.

* **Giao diện người dùng UI**

Ionic thể hiện được tiềm năng của nó. là sự đơn giản. Giống như phong cách của Google, nó dựa trên sự tồn tại của HTML5 và CSS3 để cung cấp những trải nghiệm nhanh chóng. Chính xác là tốc độ có trong sự đơn giản của nó – không có bóng đổ hay góc tròn, chỉ đơn giản là phẳng – Như vậy, Ionic không hứa hẹn có được giao diện người dùng bóng bẩy, nhưng nó cung cấp giao diện một cách nhanh chóng và nhất quán, thậm chí là chúng ta sử dụng trên các thiết bị với khả năng thông dịch HTML5 chậm trong các ứng dụng Ionic.

[](https://convoicoi.files.wordpress.com/2016/01/ionic-ui.png)Sass là một ngôn ngữ mở rộng của CSS – Cascading Style Sheets – cho phép Ionic thêm các biến số và khả năng lồng cú pháp để mở rộng sự xuất hiện của giao diện ứng dụng. Ngoài ra, Ionic còn được đóng gói thêm các thư viện icon nguồn mở, khoảng 440 icons.

* **Giao diện dựng sẵn Widgets**

Các thành phần của Ionic cực kỳ đơn giản và mạnh mẽ. Chúng là các phần tử HTML phức hợp, được gọi là các directives, Ionic cũng cung cấp các Controller để bổ sung cho cấu hình và tương tác. Ionic cung cấp các khối xây dựng có phần đơn giản mà có thể được kết hợp để cung cấp giao diện người dùng phong phú.

[](https://convoicoi.files.wordpress.com/2016/01/ionic-ui-widgets.jpg)Ở phiên bản hiện tại v1.2.8, Ionic cung cấp một loạt các thành phần giao diện thiết kế sẵn bao gồm: form elements, header and footer bars, buttons, simple list, grid elements và các thành phần phổ biến khác nữa.

* **Performance obsessed**

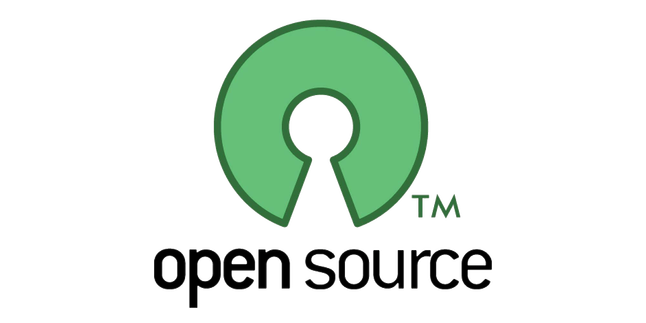
Ionic rất chú trọng đến hiệu suất, mặc dù nó ra đời chưa lâu. Ionic có một giao diện ổn định và tốc độ tốt, với các hiệu ứng chuyển động được áp dụng kỹ thuật tăng tốc phần cứng (hardware accelerating) và tối giản các thao tác với DOM. Ionic cũng không cần sử dụng đến jQuery, mặc dù người sử dụng có thể thêm vào nếu muốn. Ionic có thể tạo ra các ứng dụng phức tạp chạy trên cả iOS và Android.

Một điểm thú vị của Ionic đồng thời giúp tăng hiệu suất hoạt động của nó, đó là Ionic không cố gắng tự mình thực hiện tất cả các công việc. Để đóng gói ứng dụng, Ionic sử dụng Cordova và tận dụng cấu trúc thư mục mặc định của nó.

* **Application scripting**

Ionic không những mang đến cho chúng ta CSS và markup tùy biến cao mà còn những mẫu thiết kế (design pattern) Javascript để giúp chúng ta xây dựng những ứng dụng giống nhất với những ứng dụng native trên Android và iOS. Ionic muốn giải phóng những ứng dụng web ra khỏi thanh địa chỉ và hướng đến những tương tác giao diện mạnh mẽ như mà side menu hay view controllers.

* **Open-source**

[](https://convoicoi.files.wordpress.com/2016/01/open-source.png)Mục đích của Ionic framework không chỉ là xây dựng một bộ khung phát triển ứng dụng di động đa nền tảng mà còn muốn xây dựng một nền tảng để chia sẻ các kiến thức cho các nhà phát triển, tạo ra một cách để đưa ra những design pattern tốt nhất để xây dựng các ứng dụng di động. Vì vậy, Ionic là một framework mở hoàn toàn cho phép tất cả mọi người có thể chia sẻ những kiến thức, kinh nghiệm để xây dựng những ứng dụng di động tuyệt vời trên nền tảng HTML5.

Ionic sẽ đặt nền tảng cho ứng dụng của bạn dựa trên HTML5, CSS và Javascript. Thay vì mỗi lập trình viên phải tự tìm tòi những khuyết điểm của việc phát triển ứng dụng di động trên HTML5 thì Ionic sẽ cho bạn CSS cơ bản và tạo cho bạn 1 kiến trúc tốt để phát triển phía trên nó.

* + - * 1. Hệ thống quản trị - website admin: Framework Angular 8
* **Trang chủ: https://angular.io/**
* **Giới thiệu:**
* **Tổng quan về Angular**

Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation , http service và có các tính năng như auto-complete , navigation , toolbar , menus ,… Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

Để học được angular thì bạn cần biết các kiển thức cơ bản sau:

HTML

CSS

JavaScript

TypeScript

Document Object Model (DOM)

* **Các phiên bản của Angular**

**Angular js :** Phiên bản đầu tiền của angular là AngularJS được bắt đầu từ năm 2009 và đc ra mắt vào 20/10/2010, do lập trình viên Misko Hevery tại Google viết ra như là một dự án kiểu “viết cho vui”. Lúc đó angular js được viết theo mô hình MVC (Model-View-Controller) trong đó :

Model là thành phần trung tâm thể hiện hành vi của ứng dụng và quản lí dữ liệu.

View được tạo ra dựa trên thông tin của Model .

Controller đóng vai trò trung gian giữa Model và View và để xử lý logic .

**Angular 2 :**

Sau phiên bản angular js thì vào tháng 3 năm 2015 phiên bản bản angular 2 ra đời nhằm thay thế Angular Js với các khái niệm mới nhằm đơn giản hóa và tối ưu cho quá trình phát triển sử dụng framework này. Angular 2 thay đổi hoàn toàn so với angular js bằng việc thay Controllers và $scope ( Angular js ) bằng components và directives . Components = directives + template , tạo nên view của ứng dụng và xử lí các logic trên view. Angular 2 hoàn toàn được viết bằng Typescript. Angular 2 nhanh hơn angular js ,hỗ trợ đa nền tảng đa trình duyệt, cấu trúc cdoe được tổ chức đơn giản và dễ sử dụng hơn.

**Angular 4 :**

Ra mắt vào tháng 3/2017 đây là một phiên bản nâng cấp từ Angular 2 nên kiến trúc không thay đổi nhiều ngoài việc giảm thiểu code được tạo ra từ đó giảm kích thước tệp được đóng gói xuống 60%, đẩy nhanh quá trình phát triển ứng dụng.

**Angular 5 :**

Đã được phát hành vào ngày 1 tháng 11 năm 2017 với mục tiêu thay đổi về tốc độ và kích thước nên nó nhanh hơn và nhỏ hơn angular 4. Các tính năng mới so với angular 4:

Sử dụng HTTPClient thay vì sử dụng HTTP : bởi vì nó nhanh, an toàn và hiệu quả hơn.

Với phiên bản Angular 5 mặc định sử dụng RxJs 5.5

Multiple export aliases : Một component có thể được xuất bằng nhiều bí danh (aliases) để giảm bớt quá trình di chuyển.

Internationalized Pipes for Number, Date, and Currency: Các pipe mới được giới thiệu để tiêu chuẩn hóa tốt hơn.

Tối ưu hóa build production bằng việc sử dụng công cụ build optimizer được tích hợp sẵn vào trong CLI. Công cụ này tối ưu tree shark và loại bỏ code dư thừa.

Cải thiện tốc độ biên dịch bằng việc dùng TypeScript transforms, giờ đây khi build sẽ sử dụng lệnh “ng serve –aot”. AOT sẽ cải thiện performace khi load page và nó được dùng để deploy app lên production

**Angular 6:**

Cập nhật CLI, command line interface: thêm 1 số lệnh mới như ng-update để chuyển từ version trước sang version hiện tại; ng-add để thêm các tính năng của ứng dụng để trở thành một ứng dụng web tiến bộ.

Angular Element: Cho phép các component của Angular được triển khai dưới dạng component web, sau đó có thể được sử dụng trong bất kỳ trang HTML nào một cách dễ dàng.

Multiple Validators: cho phép nhiều Validators được áp dụng trên form builder.

Tree-shakeable providers: giúp loại bỏ mã code chết.

Sử dụng RxJS 6 với syntax thay đổi.

**Angular 7 :**

Được phát hành vào 18 tháng 10 năm 2018 với những thay đổi như :

ScrollingModule : Để scroll load dữ liệu.

Drag and Drop: Chúng ta có thể dễ dàng thêm tính năng kéo và thả vào một mục

Angular 7.0 đã cập nhật RxJS 6.3

**Angular 8 :**

Ra mắt 28 tháng 5 năm 2019 với CLI workflow improvements, Dynamic imports for lazy routes ….

**Angular 9 :**

Ra mắt mới đây 6 tháng 2 năm 2020,Angular 9 di chuyển tất cả các ứng dụng để sử dụng trình biên dịch Ivy và thời gian chạy theo mặc định. Angular đã được cập nhật để hoạt động với TypeScript 3.6 và 3.7

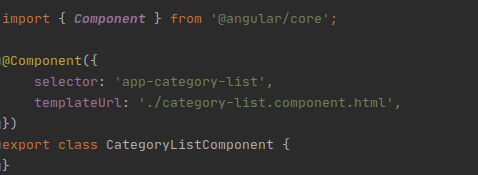
**Angular 10 :**

Ra mắt mới đây 24 tháng 10 năm 2020,Angular 10 tiếp tục nâng cấp và hoàn thiện khả năng của trình biên dịch Ivy. Angular đã được cập nhật để hoạt động với TypeScript 3.8

* **Giới thiệu về Angular**
  + **Component**

Components là một khối code trong app Angular. Nó là sự kết hợp của bộ template html và nhúng kèm code TypeScript (hoặc Javascript). Các components là độc lập với nhau và độc lập với hệ thống. Nó có thể được cài vào hoặc tháo ra khỏi hệ thống dễ dàng. Một component có thể hiểu như một control trên màn hình hiển thị, gồm giao diện html và code logic xử lý sự kiện đi kèm control đó. Một component cũng có thể to lớn như là cả 1 màn hình chứa nhiều control hoặc một nhóm nhiều màn hình. Tức là là một component cũng có thể chứa và gọi được nhiều component khác nối vào

Cấu trúc của Component:



Từ khóa [**@Component**](https://viblo.asia/u/Component) sẽ giúp định nghĩa bộ khung html cho nó. Và bên dưới là một class CategoryListComponent dùng để viết code logic. Trong định nghĩa bộ khung html, chúng ta có một số thuộc tính cần chú ý sau đây:

**selector** : Là tên được đặt để gọi một component trong code html. Ở ví dụ vừa rồi, từ khóa hello-ng-world được đặt tên cho component này. Khi cần gọi component này ra ở màn hình html cha, ta sẽ gọi bằng html tag <app-category-list></app-category-list>. Gọi như vậy thì component con sẽ được render ra component cha.\*\*

**template** : Là tự định nghĩa khung html cho component .

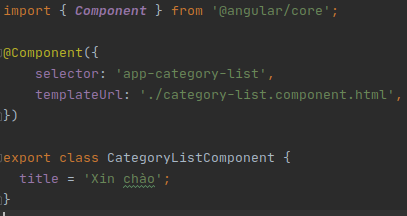
**templateUrl** : Là đường dẫn url tới file html bên ngoài để load file đó vào làm khung html cho component này.

**styles** : Là viết style css luôn vào file component này. Cách này chỉ dùng cho component đơn giản.

**styleUrls** : Là đường dẫn url đến file style css độc lập cho component này.

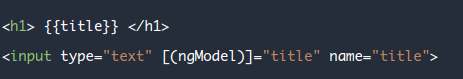
* + **Binding**

Angular có cách code Binding (kết nối giữa html và data) dữ liệu theo kiểu 2 chiều, nghĩa là html input thay đổi thì biến javascript sẽ ngay lập tức nhận được giá trị trả về và ngược lại, giá trị trong js thay đổi thì ngay lập tức màn hình html thay đổi theo.Để bind một chuỗi ra ngoài màn hình html thì rất đơn giản sử dụng 2 dấu ngoặc nhọn {{TenBien}} Ví dụ chúng ta có một Component đơn giản như sau:



https://images.viblo.asia/0959a4c9-ca65-43f0-90fd-0835db00b59d.png

* + To-way binding



Từ khóa **ngModel** lúc này không phải là thuộc tính html mà nó là từ khóa của Angular. Khi chúng ta viết **[(ngModel)]** chúng ta sẽ gắn chặt giá trị của input html với biến **title**.Dẫn đến người dùng gõ vào ô input thì thẻ h1 sẽ nhận giá trị tương ứng của **title**.

* + Xử lý sự kiện

Để gắn 1 sự kiện của control html với một hàm javascript, ta viết như sau:

https://images.viblo.asia/15862f45-85cb-4a9e-b36a-349c1060aa42.png

Khi chúng ta muốn (click) tức là muốn bind sự kiện click chuột của người dùng ở control này gọi là goBack().

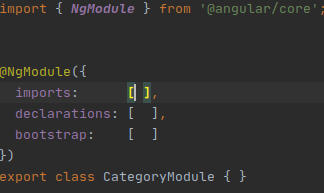
* + **ngModule**

Module là một khái niệm rộng nhất của Angular. Một module có thể bao gồm chứa các components, directives, pipes, v.v.

Module có thể được biên dịch (compile) dưới dạng ahead-of-time (AoT). Nghĩa là biên dịch ra mã thực thi để hiện ra luôn trên trình duyệt không cần vẽ vời gì từ đầu. Hãy tưởng tượng component có html và js viết riêng, khi load trang thì 2 thứ này mới nhào nặn chung để hiển thị html+data lên màn hình. AoT là thứ html+data đã nhào sẵn.

Module cũng có thể gọi module con và bắt tay được với các module khác.

Ví dụ về module chúng ta có thể bắt gặp ngay ở trong category.module.ts



Các thuộc tính của module được định nghĩa như sau:

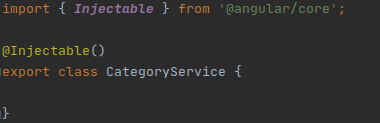
imports: Định nghĩa sự phụ thuộc (Dependency) của module này, module phụ thuộc sẽ được load trước rồi module này mới load.

declarations: Định nghĩa tất cả các component sẽ được dùng trong module này. Nếu chưa định nghĩa thì các component trong module sẽ không thể gọi nhau vì không tìm thấy nhao.

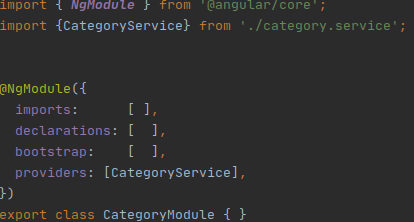
bootstrap: Mỗi ứng dụng Angular đều cần một module gốc, module này sẽ có một component gốc chứa layout gốc sẽ được render ra ở file index.html.

* + **Service**

Để tạo ra một service thì chúng ta cần import và mô tả một class với từ khóa [**@injectable**](https://viblo.asia/u/injectable) lấy từ [**@angular**](https://viblo.asia/u/angular)**/core** module.



Nếu muốn dùng chung service mà không muốn khao báo nhiều lần,component nào cũng phải tiêm nó vào.Thì lúc này có thể khai báo service ở phần **providers** của component hoặc module



* + **Router**

**Router Outlet**

Mỗi một Router sẽ có một URL để load component. Và để biết được là component sẽ render ra chỗ nào thì chúng ta viết đoạn code sau vào khung html cần chèn:

https://images.viblo.asia/2c04a65c-30c7-436f-973d-89f924ec495c.png

**Cài đặt Route cho ứng dụng Angular**

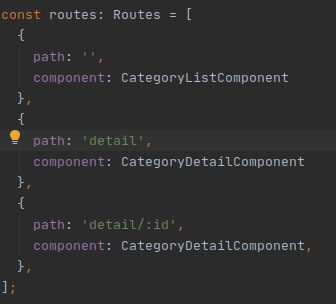
Để cài đặt toàn bộ Router cho một ứng dụng Angular thì chúng ta cần tạo ra một đối tượng JSON chứa các thuộc tính như sau:

path: Đường dẫn URL của component hiện tại.

component: Ứng với đường dẫn bên trên thì load component nào.

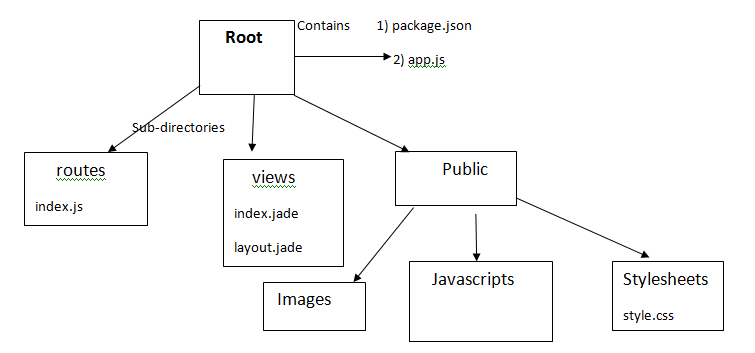
redirectTo: Chuyển hướng đến URL này nếu URL ở path không trùng. Ví dụ, khi người dùng gõ URL linh tinh, chúng ta muốn chuyển hướng và load trang Home hoặc trang báo lỗi 404 thì cần ghi rõ URL trang Home hoặc 404 vào redirectTo.

pathMatch: Cài đặt xem chế độ kiểm tra url là như thế nào. khi giá trị là full thì nghĩa là toàn bộ url từ đầu đến cuối sẽ phải chính xác như trong bộ JSON Router. Còn khi giá trị là prefix thì chuỗi đầu tiên của url (dấu sược đầu tiên) sẽ được kiểm tra. Mặc định nếu không nói gì thì prefix sẽ được chọn. Hãy xem đoạn code ví dụ về Router bên dưới:



* + - 1. Ngôn ngữ sử dụng phía backend: javascript và firebase
         1. Javascript: framework ExpressJs

**Expressjs** là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

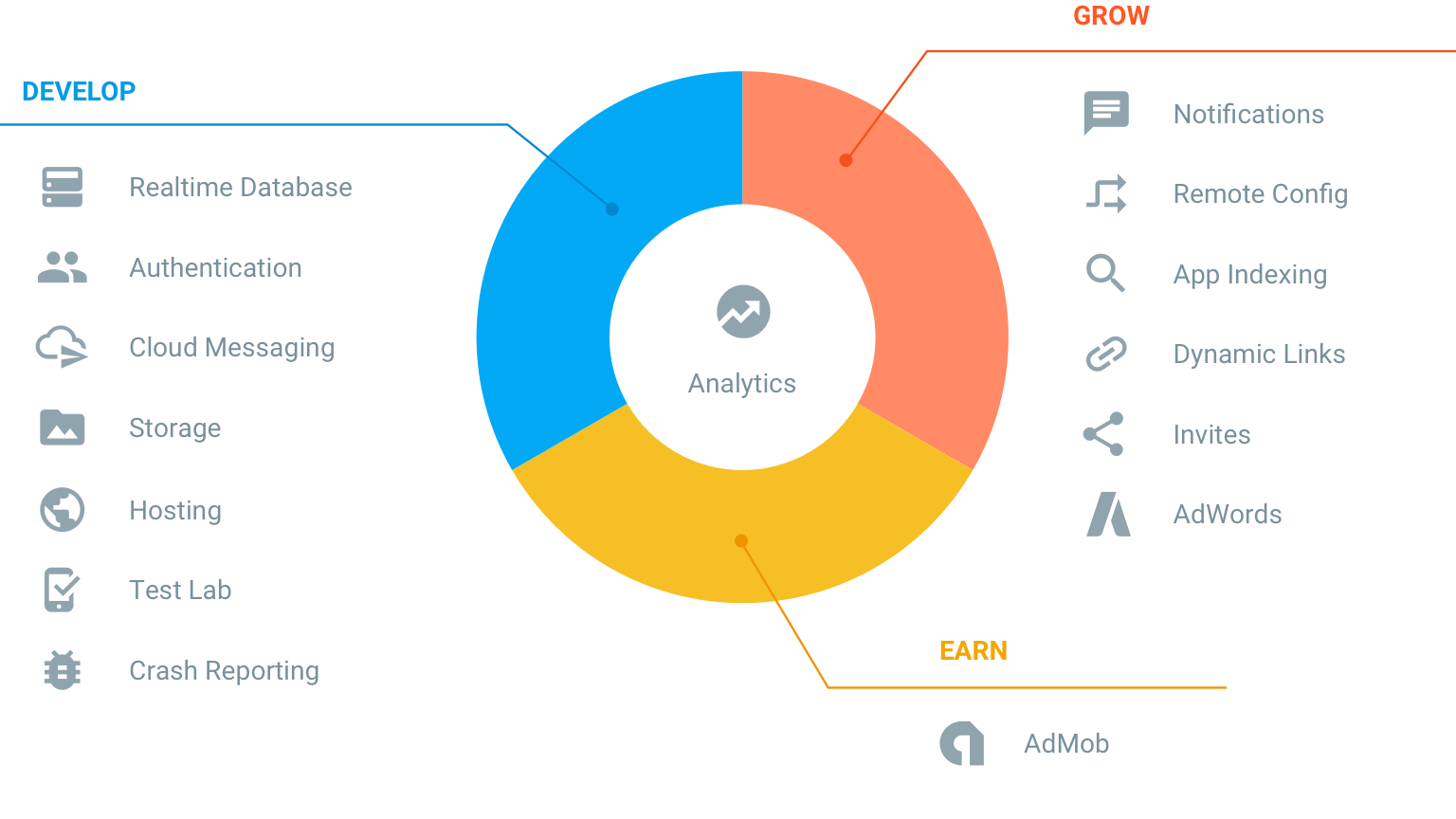


*Tổng hợp một số chức năng chính của Expressjs như sau:*

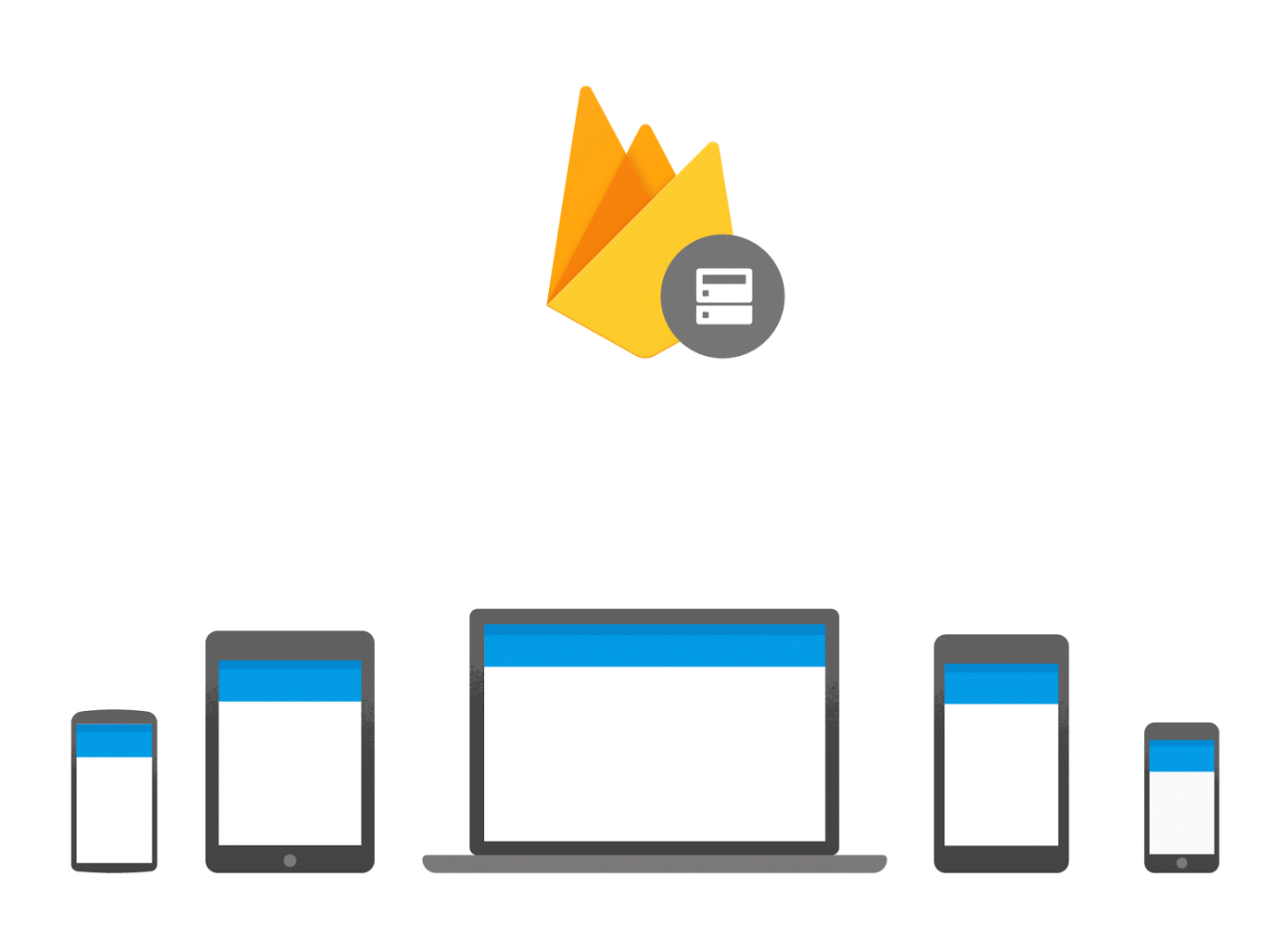
* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.
  + - * 1. Firebase
* Giới thiệu Firebase

Theo Wikipedia, thì “Firebase is a mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps“. Dịch ra là “Firebase là một nền tảng ứng dụng di động và web với các công cụ và hạ tầng được thiết kế để giúp các lập trình viên xây dựng các ứng dụng chất lượng cao“

* Các tính năng cơ bản của Firebase

  
Với Google Firebase, bạn có thể tạo ra các ứng dụng chat như Yahoo Message của ngày xưa hoặc như Facebook Messager của ngày nay trong thời gian cực ngắn như khoảng một ngày thậm chí là vài giờ bởi đơn giản là bạn chỉ cần lo phần client còn phần server và database đã có firebase lo. Firebase là sự kết hợp giữa nền tảng cloud với hệ thống máy chủ cực kì mạnh mẽ tới từ Google, để cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database. Cụ thể hơn Google Firebase cung cấp tới chúng ta những chức năng chính sau:

* Realtime Database – Cơ sở dữ liệu thời gian thực

Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.  
Ngoài ra Firebase còn cho phép bạn phân quyền một các đơn giản bằng cú pháp tương tự như javascript.  


* + Tự động tính toán quy mô ứng dụng của bạn

Khi ứng dụng của bạn muốn phát triển, bạn không cần lo lắng về việc nâng cấp máy chủ...Firebase sẽ xử lý việc tự động cho bạn. Các máy chủ của Firebase quản lý hàng triệu kết nối đồng thời và hàng tỉ lượt truy vấn mỗi tháng.

* + Các tính năng bảo mật lớp đầu

Tất cả dữ liệu được truyền qua một kết nối an toàn SSL với một chứng nhận 2048-bit. Cơ sở dữ liệu truy vấn và việc xác nhận được điều khiển tại một cấp độ chi tiết sử dụng theo một số các quy tắc mềm dẻo security rules language. Tất cả các logic bảo mật dữ liệu của bạn được tập trung ở một chỗ để dễ dàng cho việc cập nhật và kiểm thử.

* + Làm việc offline

Ứng dụng Firebase của bạn sẽ duy trì tương tác bất chấp một số các vấn đề về internet xảy ra. Trước khi bất kỳ dữ liệu được ghi đến server thì tất cả dữ liệu lập tức sẽ được viết vào một cơ sở dữ liệu Firebase ở local. Ngay khi có thể kết nối lại, client đó sẽ nhận bất kỳ thay đổi mà nó thiếu và đồng bộ hoá nó với trạng thái hiện tại server.

* Firebase Authentication – Hệ thống xác thực của Firebase

Với Firebase, bạn có thể dễ dàng xác thực người dùng từ ứng dụng của bạn trên Android, iOS và JavaScript SDKs chỉ với một vài đoạn mã. Firebase đã xây dựng chức năng cho việc xác thực người dùng với Email, Facebook, Twitter, GitHub, Google, và xác thực nạc danh. Các ứng dụng sử dụng chức năng xác thực của FireBase có thể giải quyết được vấn đề khi người dùng đăng nhập, nó sẽ tiết kiện thời gian và rất nhiều các vấn đề phức tạp về phần backend. Hơn nữa bạn có thể tích họp xác thực người dùng với các chức năng backend đã có sẵn sử dụng custom auth tokens.

* Firebase Hosting

Phát triển ứng dụng web của bạn trong thời gian ngắn với các hosting tĩnh đã được cung cấp sẵn. Tất cả các kết nối được phân phối qua SSL từ CDN trên toàn thể giới của Firebase.

Triểu khai siêu tốc: Việc triển khai sử dụng các công cụ dòng lệnh Firebase và có thể quay trở lại với phiên bản trước chỉ với một cú click chuột. Tất cả các ứng dụng sẽ có đường dẫn mặc đinh ở sau firebaseapp.com và nếu trả phí thì có thể triểu khai một tên miền tuỳ chỉnh.

SSL bởi default: Mọi ứng dụng được xử lý thông qua một kết nối an toàn, và Firebase đã cẩn thận cung cấp SSL cert cho bạn.

* 4. Cloud Messaging

Google Cloud Messaging, dịch vụ cloud-to-device push messaging nổi tiếng nhất thế giới, đang tích hợp với Firebase và đổi tên thành Firebase Cloud Messaging (FCM). FCM không có giới hạn và hoàn toàn miễn phí. FCM hỗ trợ messaging trên iOS, Android và Web, và tối ưu mạnh mẽ hướng đến bảo mật và tiếp kiệm pin. FCM có thể xử lý lượng lớn thông tin và hiện đã có thể gửi 170 tỷ tin nhắn mỗi ngày đến 2 tỷ thiết bị.  


* Firebase Storage

Firebase cũng sẽ ra mắt Firebase Storage để giúp lập trình viên upload và download file dễ dàng. Firebase Storage được Google Cloud Storage hỗ trợ, mang đến khả năng lưu trữ khổng lồ và cho phép Google Cloud project truy cập file dễ dàng. Firebase Storage client SDKs còn có logic nâng cao có thể dễ dàng xử lý điều kiện mạng kém.

* Firebase Test Lab và Crash Reporting

Chả ai muốn ứng dụng crash cả, khi đó, người dùng thì khó chịu và doanh nghiệp thì bị mất tín nhiệm. Firebase Crash Reporting là công cụ báo cáo crash có khả năng ưu tiên, hành động nhanh giúp bạn chuẩn đoán và xử lý ngay các vấn đề của ứng dụng iOS và Android. Crash Reporting còn có thể kết nối đến nhóm Audiences trong Firebase Analytics, từ đó bạn có thể biết được người dùng bị crash đang dùng thiết bị nào, khu vực địa lý nào,…

Cloud Test Lab (được giới thiệu tại Google I/O năm ngoái), giờ đây đã trở thành Firebase Test Lab for Android. Test Lap giúp bạn xác định vấn đề của ứng dụng trước khi đến tay người dùng. Test Lap cho phép bạn chạy thử ứng dụng (tự động hoặc tùy chỉnh) trên thiết bị thật do trung tâm dữ liệu Google xử lý.

### Hệ quản trị cơ sở dữ liệu: Firebase realtime database

* Firebase Realtime Database là gì?

Realtime Database một service của Firebase. Theo định nghĩa trong tài liệu của firebase thì Realtime Database là:

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Dịch nôm na thì nó có nghĩa là một cơ sở dữ liệu NoSQL lưu và đồng bộ dữ liệu trên mây. Dữ liệu được đồng bộ trên tất cả clients trong thời gian thực, và vẫn khả dụng khi ứng dụng offline.

Firebase Realtime Database là cơ sở dữ liệu lưu trữ trên mây. Dữ liệu được lưu trữ và đồng bộ hóa theo thời gian thực với mỗi client được kêt nối. Khi bạn xây dựng ứng dụng đa nền tẩng với iOS, Android, và javascript SDK, tất cả các client của bạn chia sẽ một thể hiện Realtime Database và tự động tiếp nhận các thay đổi với dữ liệu mới nhất.

Các khả năng chính của Realtime Database

Realtime:

Firebase Realtime Database sử dụng đồng bộ dữ liệu mối khi dữ liệu có thay đổi, mọi thiết bị được kết nối sẽ nhận được thay đổi trong vài mili giây.

Offline:

Khi người dùng ngoại tuyến, dữ liệu sẽ được lưu trên bộ nhớ cache của thiết bị và tự động đồng bộ khi bạn trực tuyến. Tất cả là tự động

Accessible from Client Devices

Firebase Realtime Database có thể truy cập từ một thiết bị mobile hoặc trình duyệt web. Nó không cần một ứng dụng server nào cả. Bảo mật và xác thực dữ liệu có thể thông qua các Rule bảo mật của Firebase Realtime Database, các rule được thực thi khi dữ liệu được đọc hoặc ghi.

## Giới thiệu về phần mềm