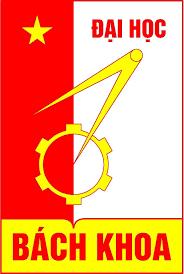
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

====o0o====



**Báo cáo tổng kết GR1**

**Đề tài:**

**Tìm hiểu về UI/UX và phát triển ứng dụng áp dụng các nguyên lý thiết kế UI/UX**

**Giảng viên hướng dẫn : TS. Vũ Thị Hương Giang**

**Sinh viên thực hiện : Đặng Như Thủy - 20144373**

**Lớp : Việt Nhật AS K59**

# 

# Mục Lục

[**Mục Lục**](#_cegz5kyx0yty) **1**

[**LỜI CẢM ƠN**](#_lhg6sjeyagjs) **2**

[**Khảo sát xí nghiệp**](#_xiz1ga4mu84) **3**

[**Mô tả sản phẩm**](#_bu9wz5ghw9cr) **5**

[Đặc điểm của sản phẩm](#_18a9cd3imhlq) 5

[Mô tả khái quát](#_jn3jh01w9a3d) 5

[Phần mềm dành cho khách hàng:](#_kgq83y20nksu) 5

[Phần mềm dành cho nhân viên](#_y6bx496k4zav) 6

[Đặc tả một số Usercase chính](#_kne4wcejlyhn) 7

[So sánh với các sản phẩm tương tự](#_b5afu9cu2d5m) 12

[**Bài toán liên quan và giải pháp**](#_ryimklptljz3) **17**

[Bài toán tối ưu: quãng đường và công việc cho nhân viên](#_2p73t9xyb1ui) 17

[Các tham số đầu vào](#_hdteheeitjw6) 17

[Mô tả bài toán](#_ocluduol4ncl) 18

[Thuật toán tham lam](#_ha1jcz7so2y4) 18

[Thuật toán di truyền (tham khảo)](#_dlbr68rgojmv) 22

[**Kiến trúc hệ thống và yêu cầu công nghệ**](#_hbkgw8kwtgyn) **24**

[Kiến trúc hệ thống](#_hh9f27vzd2ib) 24

[Yêu cầu công nghệ](#_mnx0n91n6qui) 27

[**Tài liệu tham khảo**](#_f6pgwco3lvv0) **27**

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến TS. Vũ Thị Hương Giang – giảng viên Trường Đại Học Bách Khoa Hà Nội, đã tận tình hướng dẫn , giúp đỡ em trong suốt quá trình nghiên cứu tốt nghiệp 1.

Em chân thành cảm ơn quý Thầy, Cô trong khoa Công Nghệ Thông Tin Và Truyền Thông – Trường Đại Học Bách Khoa Hà Nội 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 này mà còn là hành trang quí báu để em bước vào đời một cách vững chắc và tự tin.

Trong giai đoạn đầu, việc tìm hiểu và khảo sát vẫn còn nhiều thiếu sót, em rất mong sự chỉ bảo thêm của cô và sẽ cố gắng cải thiện trong các kì sau.

**Nội dung**

# Khảo sát xí nghiệp

1. Quy trình giặt là Sapy
   1. Loại hình đăng kí hiện có:

* Trực tiếp mang tới và đăng kí
* Đăng kí qua trang web : giatlasapy.com (lượng khách đăng kí chủ yếu là các khách sạn và nhà hàng)
  1. Quy trình
* Nhận đồ giặt là:
  + Trực tiếp: khách hàng mang đồ đến tận nơi đăng kí và nhận đồ
  + Qua trang web giatlasapy.com: thường là các khách sạn và nhà hàng, sẽ có xe tới nhận đồ và trả đồ.
  + Nhận và giặt là tận nơi: Thông qua số hotline, nhân viên thông qua địa chỉ và số điện thoại tới nhận đồ.
* Đánh mã số:
  + Nếu đồ giặt tính theo chiếc, nhân viên dựa trên bảng mã của loại quần áo sẽ nhập mã đó, có thể điền thêm các thông tin liên quan như số lượng, màu, các vết hỏng hóc... vào trong máy. Nếu đồ giặt tính theo cân sẽ điền thêm thông tin cân nặng. Máy tính sẽ tự động in ra 2 phiếu gồm mã số tương ứng cho đồ. Một phiếu cho túi đồ, một phiếu cho khách nhận hàng.
  + Hình thức mã số được đánh: VD 18A28/30001 trong đó 18 là hai chữ số cuối của năm, A là tháng 1 (tương ứng tháng 2 là B, tháng 3 là C ,...) 28 là ngày, 30001 là số thứ tự của đồ giặt thứ 1
* Chuyển tới xưởng giặt: nhân viên dựa trên các mã đã được đánh trên đồ sẽ xác nhận qua máy tính và lần lượt cho vào máy giặt.
* Chuyển tới xưởng sấy khô: Sau quá trình giặt xong các nhân viên sẽ xác nhận mã số đồ đã giặt qua máy tính và chuyển tới xưởng sấy.
* Chuyển tới kho chứa: Sau khi sấy xong, nhân viên chuyển các đồ đã sấy tới kho chứa đồ và xác nhận qua máy tính đồ đã sấy xong.
* Chuyển đồ về kho chứa:
  + Kho được chia theo các gian: mỗi gian tương ứng với mỗi tháng.
  + Các đồ tương ứng giặt của tháng nào sẽ được đưa vào nguyên của tháng đó.
  + Các đồ khi giặt và sấy xong đưa vào gian trong kho sẽ không theo thứ tự mà sắp xếp ngẫu nhiên.
  + Chú ý:
* Khi số lượng quần áo lên khá nhiều, đồ sẽ được gom dồn vào các tháng trước đó (từ tháng 1). Vì vậy buộc các nhân viên phải nhớ loạt đồ đã giặt nằm ở gian nào. Từ đó tìm kiếm trong gian vị trí đồ.
* Lượng quần áo khi quá hạn lấy sẽ được chuyển sang một kho chứa dự phòng khác và sắp xếp đúng theo thứ tự về gian hàng.
* Trả đồ cho khách:
  + Với đồ khách sạn và nhà hàng: có xe ô tô vận chuyển
  + Với đồ nhận giặt là tại nhà: có xe vận chuyển trả tận nơi
  + Với đồ giặt là trực tiếp:
* Khách hàng mang phiếu đã nhận khi đăng kí tới.
* Nhân viên sử dụng máy đọc mã vạch đọc dãy mã trên phiếu, thông tin đồ giặt trong phiếu hiện lên trên máy tính. Nhân viên dựa trên thông tin sẽ xác định nơi đồ đang lưu giữ (Chưa giặt, Đang giặt, Đang Sấy, Trong kho hàng, trong kho tồn).
* Thông tin khách sẽ được lưu lại, mã số phiếu sẽ được xóa.

1. Thông tin xí nghiệp yêu cầu từ khách hàng:

* Tên
* Địa chỉ
* Số điện thoại
* Ngày/giờ giao nhận đồ
* Thông tin về đồ giặt: loại đồ giặt, màu sắc, cân hay chiếc, đặc điểm hiện tại.

1. Yêu cầu chính về phần mềm từ xí nghiệp

* Phần mềm cho khách hàng: sử dụng để có thông tin liên lạc, đặt đồ trước, nhận được thông báo đã đặt thành công, cập nhật trạng thái đang giặt hay đã trả theo thời gian thực,...
* Phần mềm cho xí nghiệp: Nhận hàng đã đặt gồm thông tin khách hàng, thông tin đồ giặt, thông báo đã nhận hay từ chối.

# Mô tả sản phẩm

## Đặc điểm của sản phẩm

### Mô tả khái quát

#### Phần mềm dành cho khách hàng:

* Khi người dùng mở ứng dụng lên, không yêu cầu đăng nhập hay đăng kí tài khoản. Yêu cầu người dùng lựa chọn khu vực đang sống, trong menu lựa chọn chỉ có thể chọn những khu vực xí nghiệp có cửa tiệm đặt tại đó đang kinh doanh. Hiện tại chỉ có Hà Nội.
* Người dùng vào màn hình chính, có menu dạng các tab: Trang chủ, giá cả, lịch sử, thông tin cá nhân.
* Tại tab Trang chủ:
* Sẽ có các lựa chọn về hình thức như giặt khô là hơi, chỉ là, chỉ sấy,... Mỗi hình thức sẽ có các loại đồ như vải len, bò, vét, đồ sơ sinh, đồ bếp núc, đồ phòng ngủ, gấu bông, rèm cửa, giầy dép,... ngoài ra có thêm mục giặt theo cân và đồ giặt do người dùng tự nhập.
* Bên cạnh các mục là giá cả tham khảo cùng nút tăng/giảm số lượng, tổng tiền ước lượng sẽ được ghi bên dưới cùng.
* Khi người dùng nhấn chọn đặt lịch, màn hình đặt lịch sẽ hiện lên.
* Tại màn hình đặt lịch người dùng cần điền thêm các thông tin: Tên, số điện thoại, mặc định sẽ chọn địa chỉ trong phần thông tin cá nhân nhưng nếu người dùng muốn nhập địa chỉ khác có thể nhập lại địa chỉ bằng chữ (bên dưới phần này sẽ có một bản đồ, sau khi người dùng nhập sẽ hiển thị), lựa chọn loại hình đặt gồm: một là người dùng tự mang đồ tới, sau khi xong sẽ tự mang về; hai là nhận trả đồ tận nhà.
* Với loại người dùng tự mang và lấy chỉ cần lựa chọn ngày/giờ mang đồ. Với loại nhận trả tại nhà, người dùng cần lựa chọn ngày/giờ ngày giao và nhận.
* Tiếp theo người dùng chọn loại hình thanh toán: thanh toán khi nhận đồ, bằng thẻ.
* Thông tin đồ đã đặt, số lượng, số tiền tham khảo được hiển thị.
* Cuối cùng chọn chấp nhận để đặt hoặc hủy. Màn hình sẽ chuyển sang tab lịch sử của mục người dùng vừa đặt.
* Tại tab Giá cả
* Có thể tìm kiếm bằng thanh tìm kiếm trên cùng.
* Có thể xem danh sách các hình thức giặt và các loại đồ có thể giặt cùng giá cả tham khảo.
* Tại tab lịch sử
* Danh sách lịch sử đã đặt lịch của người dùng được ghi lại.. Người dùng click chọn một mục để xem chi tiết.
* Chi tiết gồm thông tin trạng thái của đồ là các biểu tượng, khi đến giai đoạn đó biểu tượng tương ứng sẽ sáng lên và tích màu xanh. Với trường hợp nhận trả tận nhà: nhân viên đang tới nhận, đã nhận, đã giặt xong, nhân viên đang tới trả, đã trả; với trường hợp người dùng tự mang tới và tự đến lấy: đã nhận, đã giặt xong, đã trả.
* Mỗi khi chuyển trạng thái sẽ đưa thông báo ra thanh trạng thái hệ thống của điện thoại.
* Ngoài ra gồm các thông tin của hóa đơn về ngày/giờ giao nhận, các đồ giặt, số lượng, giá cả tham khảo(chỉ tiết sẽ được điều chỉnh khi tới giai đoạn nhận đồ), thông tin của nhân viên sẽ tới trao nhận đồ (với trường hợp giao nhận tận nơi).
* Đánh giá, comment.
* Chức năng mong muốn: gọi điện và nhắn tin cho nhân viên đang mang tới.
* Tại tab thông tin cá nhân
* Người dùng có thể thay đổi thông tin cá nhân: tên, số điện thoại, địa chỉ mặc định(gồm một bản đồ hiển thị), thêm thẻ thanh toán.

#### Phần mềm dành cho nhân viên

* Khi mở ứng dụng lên, nhân viên cần đăng nhập tài khoản của mình.
* Tại màn hình chính gồm các tab: Chấp nhận đơn đặt hàng, đang tới nhận đồ, đã nhận, đang giặt, đang tới trả đồ, đã giao. Mỗi tab sẽ gồm danh sách các đơn hàng đang ở trạng thái đó.
* Tại mỗi tab(giai đoạn, trạng thái) hóa đơn nào được xử lý xong, nhân viên sẽ chọn dấu tích và tự động hóa đơn sẽ chuyển sang trạng thái kế tiếp hoặc dấu X để quay lại nếu nhấn nhầm.
* Trong tab chấp nhận đơn hàng, khi nhân viên chọn chi tiết một đơn hàng, thông tin của đơn hàng sẽ hiển thị gồm: tên, số điện thoại, địa chỉ của khách hàng, hình thức đặt(tự mang hay nhận trả tận nhà), hình thức giặt, các loại đồ giặt, số lượng, ngày/giờ nhận và trả, hình thức thanh toán.
* Trong tab đang tới nhận đồ và đang tới trả đồ nhân viên sẽ có một bản đồ. Hệ thống sẽ dựa vào địa chỉ trên các hóa đơn hiện có của trạng thái để tính toán ĐƯỜNG ĐI NGẮN NHẤT. Mỗi lần có hóa đơn được cập nhật vào đường đi sẽ được tính toán lại và hiển thị lên bản đồ.
* Trường hợp khách hàng tự mang đồ tới: hóa đơn đó sẽ bỏ qua giai đoạn nhân viên tới nhận và nhân viên tới trả.
* Sau khi đồ đã được trả cho khách hoặc đồ đã hủy nhận, mọi hóa đơn sẽ được đưa vào trong lịch sử.
* Nhân viên có thể thay đổi thông tin cá nhân.

### Đặc tả một số Usercase chính

* Đặt lịch theo hình thức nhận trả tận nơi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC001 | | **Tên use case** | Đặt lịch theo hình thức nhận trả tận nơi |
| **Tác nhân** | Khách hàng | | | |
| **Tiền điều kiện** | Đặt lịch thành công | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Khách hàng | Lựa chọn hình thức giặt, đồ muốn giặt là và số lượng từng đồ hoặc có thể ghi cụ thể trong phần do người dùng tự ghi. | |
| 2. | Khách hàng | Chọn đặt lịch | |
| 3. | Hệ thống | Chuyển sang màn hình thông tin đơn hàng | |
| 4. | Khách hàng | Nhập các thông tin cần thiết: tên, số điện thoại, địa chỉ nhận trả, ngày/giờ nhận trả, lựa chọn hình thức NHẬN TRẢ TẬN NƠI, hình thức thanh toán | |
| 5. | Hệ thống | Kiểm tra các trường cần yêu cầu điền đã đầy đủ chưa? | |
| 6. | Khách hàng | Chấp nhận hoặc hủy việc đặt lịch bằng cách click nút Chấp nhận hoặc hủy | |
| 7. | Hệ thống | Chuyển sang màn hình đặt lịch của đơn hàng vừa đặt. | |
| **Luồng sự kiện thay thế** | 5a. | Hệ thống | Thông báo lỗi: cần nhập đủ thông tin cần thiết. | |
| **Hậu điều kiện** | Không có | | | |

* Đặt lịch theo cách mang đồ đến tiệm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC002 | | **Tên use case** | Đặt lịch theo hình thức khách hàng tự mang đồ tới |
| **Tác nhân** | Khách hàng | | | |
| **Tiền điều kiện** | Đặt lịch thành công | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Khách hàng | Lựa chọn hình thức giặt, đồ muốn giặt là và số lượng từng đồ hoặc có thể ghi cụ thể trong phần do người dùng tự ghi. | |
| 2. | Khách hàng | Chọn đặt lịch | |
| 3. | Hệ thống | Chuyển sang màn hình thông tin đơn hàng | |
| 4. | Khách hàng | Nhập các thông tin cần thiết: tên, số điện thoại, địa chỉ nhận trả, ngày/giờ nhận trả, lựa chọn hình thức TỰ MANG ĐỒ TỚI, hình thức thanh toán | |
| 5. | Hệ thống | Kiểm tra các trường cần yêu cầu điền đã đầy đủ chưa? | |
| 6. | Khách hàng | Chấp nhận hoặc hủy việc đặt lịch bằng cách click nút Chấp nhận hoặc hủy | |
| 7. | Hệ thống | Chuyển sang màn hình đặt lịch của đơn hàng vừa đặt. | |
| **Luồng sự kiện thay thế** | 5a. | Hệ thống | Thông báo lỗi: cần nhập đủ thông tin cần thiết. | |
| **Hậu điều kiện** | Không có | | | |

#### 

* Hủy đặt lịch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC003 | | **Tên use case** | Hủy đặt lịch |
| **Tác nhân** | Khách hàng | | | |
| **Tiền điều kiện** | Hủy đặt lịch thành công | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Khách hàng | Vào đơn hàng muốn hủy | |
| 2. | Hệ thống | Kiểm tra đơn hàng đã ở trạng thái NHẬN hay chưa? | |
| 3. | Khách hàng | Chọn hủy đặt lịch | |
| 4. | Hệ thống | Thông báo tới nhân viên và | |
| 5. | Hệ thống | Đơn hàng sẽ được xóa khỏi danh sách đang chờ được chấp nhận | |
| 6. | Hệ thống | Thông báo cho khách hàng đã hủy đặt lịch thành công | |
| **Luồng sự kiện thay thế** | 2a. | Hệ thống | Khi đã ở ngay tại hoặc sau trạng thái Nhận: nút hủy bỏ sẽ bị tối đi. | |
| **Hậu điều kiện** | Không có | | | |

* Xem thông tin trạng thái đồ giặt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC004 | | **Tên use case** | Xem thông tin trạng thái đồ giặt |
| **Tác nhân** | Khách hàng | | | |
| **Tiền điều kiện** | Hiển thị đúng trạng thái hiện tại của đồ giặt | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Hệ thống | Khi trạng thái của đồ thay đổi, thiết bị có bật mạng. Hệ thống của ứng dụng sẽ hiển thị thông báo trên thanh trạng thái hệ thống của thiết bị. | |
| 2. | Khách hàng | Lựa chọn đơn hàng đã đặt trong mục lịch sử. | |
| 3. | Hệ thống | Hiển thị toàn bộ thông tin của hóa đơn | |
| 4. | Hệ thống | Cập nhật trạng thái của đơn hàng hiện tại và thay đổi hiệu ứng trên thanh trạng thái. | |
| **Luồng sự kiện thay thế** |  |  |  | |
| **Hậu điều kiện** | Không có | | | |

* Chấp nhận đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC005 | | **Tên use case** | Chấp nhận đơn hàng |
| **Tác nhân** | Nhân viên | | | |
| **Tiền điều kiện** | Chấp nhận đơn hàng của khách | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Hệ thống | Cập nhật các đơn hàng liên tục và hiển thị lên màn hình chính của nhân viên. | |
| 2. | Nhân viên | Chọn xem đơn hàng tại tab Đơn hàng đang chờ. | |
| 3. | Hệ thống | Hiển thị chi tiết thông tin đơn hàng khách đã đặt. | |
| 4. | Nhân viên | Chọn chấp nhận đơn hàng. | |
| 5. | Hệ thống | Chuyển đơn hàng sang trạng thái kế tiếp: với nhân trả tận nơi sẽ chuyển sang trạng thái Nhân viên đang tới nhận đồ; với loại khách hàng tự mang đồ sẽ chuyển sang trạng thái khách đang mang đồ tới. | |
| 6. | Hệ thống | Cập nhật lại trạng thái trên cơ sở dữ liệu | |
| 7. | Hệ thống | Cập nhật lại trạng thái của đồ trên tài khoản của khách hàng. | |
| **Luồng sự kiện thay thế** |  |  |  | |
| **Hậu điều kiện** | Không có | | | |

* Từ chối đơn hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC006 | | **Tên use case** | Từ chối đơn hàng |
| **Tác nhân** | Nhân viên | | | |
| **Tiền điều kiện** | Từ chối đơn hàng | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Hệ thống | Cập nhật các đơn hàng liên tục và hiển thị lên màn hình chính của nhân viên. | |
| 2. | Nhân viên | Chọn xem đơn hàng tại tab Đơn hàng đang chờ. | |
| 3. | Hệ thống | Hiển thị chi tiết thông tin đơn hàng khách đã đặt. | |
| 4. | Nhân viên | Chọn từ chối đơn hàng. | |
| 5. | Hệ thống | Cập nhật trạng thái bị hủy | |
| 6. | Hệ thống | Chuyển đơn hàng sang lịch sử | |
| 7. | Hệ thống | Cập nhật lại trạng thái trên cơ sở dữ liệu | |
| 8. | Hệ thống | Thông báo cho khách hàng đơn hàng đã bị từ chối. | |
| **Luồng sự kiện thay thế** |  |  |  | |
| **Hậu điều kiện** | Không có | | | |

* Thay đổi trạng thái đơn hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã use case** | UC007 | | **Tên use case** | Thay đổi trạng thái đơn hàng |
| **Tác nhân** | Nhân viên | | | |
| **Tiền điều kiện** | Thay đổi trạng thái đơn hàng và cập nhật trạng thái trên tài khoản của khách hàng. | | | |
| **Luồng sự kiện chính**  **(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | |
| 1. | Hệ thống | Cập nhật các đơn hàng liên tục và hiển thị lên màn hình chính của nhân viên. | |
| 2. | Nhân viên | Chọn xem đơn hàng tại bất kì tab nào cần chuyển đổi trạng thái. | |
| 3. | Hệ thống | Hiển thị chi tiết thông tin đơn hàng | |
| 4. | Nhân viên | Chọn nút tích xanh bên phải để chuyển đơn hàng sang trạng thái tiếp theo hoặc nút X bên trái để chuyển về trạng thái trước đó | |
| 5. | Hệ thống | Cập nhật lại trạng thái đơn hàng. | |
| 6. | Hệ thống | Cập nhật lại trạng thái trên cơ sở dữ liệu | |
| 7. | Hệ thống | Thông báo cho khách hàng trạng thái của đơn hàng. | |
| **Luồng sự kiện thay thế** | 5.a | Hệ thống | Nếu chuyển sang trạng thái Nhân viên đang tới nhận hoặc trả hệ thống sẽ tính toán đường đi Ngắn nhất cho nhân viên di chuyển. | |
| **Hậu điều kiện** | Không có | | | |

### So sánh với các sản phẩm tương tự

* **Laundrapp**:
* Cần xác định mã Poscode để xem khu vực có chi nhánh để hỗ trợ dịch vụ giặt là hay không.
* Có thể Order tại mọi màn hình.
* Đồ cần giặt sẽ phân chia theo từng loại: loại hay giặt nhất, giặt theo cân, đồ ngủ, comle, đồ chăn ga gối đệm, quần, váy, áo khoác, khăn và cà là vạt, đồ nhà bếp, đồ nhà tắm.
* Có chức năng tìm kiếm các loại quần áo.
* Có thể tăng/giảm số lượng đồ cần giặt bằng cách nhấn giữ Item đồng thời kéo sang phải là tăng, kéo sang trái là giảm.
* Khi Order, có đăng kí ngày/giờ nhận và ngày/giờ trả.
* Khi Order, xác định vị trí hiện tại hoặc nhập vị trí sẽ giao nhận đồ (có bản đồ hiển thị vị trí).
* Người dùng có thể xem vị trí của nhân viên đang tới qua một bản đồ.
* Thông báo các trạng thái tiến độ : Đặt -> Lấy -> Đang xử lý -> Trả.
* Thông tin hóa đơn gồm: loại đồ giặt, giá từng loại, số lượng và thành tiền.
* Có thể nhập mã voucher để giảm giá.
* **Ahlijasa** (một sản phẩm startup của Indonesia):
* Khi vào cần đăng kí tài khoản google hoặc facebook.
* Màn hình chính có duy nhất nut Order.
* Khi chọn Order sẽ cần xác định vị trí để nhận trả đồ.
* Hệ thống sẽ kiểm tra xem khu vực có hỗ trợ dịch vụ giặt là hay không.Nếu khu vực hỗ trợ dịch vụ, người dùng có thể chọn loại đồ muốn giặt, số lượng, có thể nhập voucher nếu có.
* Có chức năng tìm kiếm các loại quần áo.
* Sau đó nhập ngày/giờ giao và nhận.
* Cần nhập đầy đủ thông tin tên, số điện thoại để liên lạc nếu người dùng chưa nhập.
* Hiển thị thông tin của hóa đơn, người dùng kiểm tra lại thông tin và Confirm.
* Sau khi đã Order, người dùng có thể xem hóa đơn của mình. Trong hóa đơn sẽ gồm các thông tin: về tiến trình(Đang xử lý, đang đến nhận đồ, đang giặt, đang giao trả hàng, hòan tất), Bản đồ gồm thông tin vị trí của mình và nhân viên đang tới, đồ mình đã đặt, ngày giờ nhận và trả, thông tin liên hệ.
* Có thể gọi điện hoặc nhắn tin cho nhân viên đang tới.
* Tại màn hình chính sẽ có phần thông báo về tiến độ hoặc quảng cáo.
* Có thể chat với nhân viên qua ứng dụng.
* Không thể hủy khi đã Order xong.
* Người dùng có thể thay đổi thông tin cá nhân.
* **Getlaundry**(chỉ áp dụng ở Dubai):
* Không cần đăng nhập khi vào.
* Có chức năng chat với nhân viên hỗ trợ.
* Tại màn hình chính, chọn vị trí của mình. Hệ thống sẽ xác định khu vực đó có hỗ trợ dịch vụ giặt là hay không.
* Một dãy các tiệm giặt sẽ hiển thị. Dựa vào sao đánh giá, lời quảng cáo ngắn, khách hàng có thể lựa chọn một HÃNG ưa thích.
* Khi người dùng đã chọn, tùy vào từng hãng sẽ có những kiểu giặt là khác nhau như Giặt, Sấy, Vắt và từng loại quần áo. Người dùng tăng giảm số lượng tùy ý nhưng ít nhất tổng giá thành phải trên một mức độ nhất định mới được đặt.
* Sau khi đã chọn xong loại quần áo và số lượng, người dùng cần điền đầy đủ địa chỉ của nhà hoặc nơi làm việc theo định dạng mẫu, chọn ngày/giờ sẽ giao nhận đồ. Khách hàng có thể kiểm tra lại loại đồ và số lượng đã đặt.
* Có chức năng tìm kiếm các loại quần áo.
* Cuối cùng để có thể Checkout, người dùng cần đăng nhập.
* Khi đăng kí tài khoản, hệ thống sẽ gửi mã đến số điện thoại để xác thực.
* Có thể thanh toán bằng một số loại thẻ.
* Khi giao hàng, hệ thống sẽ thông báo tới tài khoản.
* Người dùng có thể thay đổi thông tin cá nhân.
* **Laundry Waves**(chỉ áp dụng tại Ấn độ)
* Khi vào sẽ có các lựa chọn: Gọi điện trực tiếp, Order qua ứng dụng
* Nếu gọi trực tiếp sẽ chuyển sang chức năng gọi điện của điện thoại đó cùng với số điện thoại của tiệm.
* Nếu Order qua ứng dụng cần đăng kí và đăng nhập. Khi đăng kí cần các thông tin: tên, email, sđt, mã vùng, mã xác nhận.
* Khi đã đăng nhập, người dùng lực chọn hình thức giặt đồ như giặt sấy, là, là hơi,.. Mỗi loại sẽ có lựa chọn cho đồ của nam, nữ, trẻ em, đồ chơi, gia đình.
* Khi Checkout, cần nhập đầy đủ địa chỉ sẽ giao nhận đồ, hình thức thanh toán(qua thẻ, khi giao hàng, khi nhận hàng)
* Người dùng kiểm tra lại hóa đơn và đặt lịch.
* Người dùng có thể thay đổi thông tin cá nhân.
* Người dùng có thể xem lại lịch sử đã đặt.
* **Laundryheap**:
* Khi vào có thể đăng nhập hoặc không cần.
* Nếu không đăng nhập sẽ cần lựa chọn quốc gia và nhập email, người dùng cần xác thực qua email đã đăng kí.
* Người dùng có thể xem thông tin về giá cả của các loại đồ giặt.
* Tại màn hình chính sẽ có nút để Order. Khi người dùng click vào đó, sẽ hiển thị các hình thức giặt là như giặt khô, là hơi, sấy... mỗi hình thức sẽ có các nút checkitem để chọn.
* Tiếp theo người dùng cần xác định địa chỉ chính xác của mình để giao, nhận đồ. Có thể ở nhà, cơ quan, khách sạn,... và các chú ý cần thiết.
* Tiếp theo người dùng cần chọn ngày/giờ giao và nhận đồ.
* Người dùng nhập tên và số điện thoại để liên lạc.
* Do chỉ cho thanh toán bằng thẻ nên người dùng bắt buộc phải nhập mã thẻ thanh toán.
* Người dùng có thể thay đổi thông tin cá nhân.
* **Piing**(chỉ áp dụng tại Singapore):
* Khi vào cần đăng kí hoặc đăng nhập tài khoản. Người dùng nhập tên, email, mã poscode để xác định khu vực có hỗ trợ dịch vụ hay không, một địa chỉ cố định để giao nhận hàng. Người dùng có thể thêm địa chỉ giao nhận hàng. Cần nhập thông tin thẻ thanh toán.
* Tại màn hình chính sẽ có nút Ping giữa màn hình. Hệ thống sẽ tìm nhân viên gần nhất để đến lấy đồ. Trong một khoảng thời gian là 300 giây nếu khách hàng không nhận gọi sẽ tự hủy hoặc người dùng lựa chọn gọi sau để chuyển tới phần đặt lịch.
* Trong phần đặt lịch, người dùng lựa chọn loại hình giặt, ngày giờ giao đồ, chọn loại đồ, chọn hình thức thanh toán (chỉ bằng thẻ), chọn ngày giờ nhận đồ.
* Tại màn hình Order sẽ có các trạng thái. Đang đặt, nhận đồ, đang giặt và giao hàng.
* Có thể liên lạc trực tiếp với nhân viên đang đến lấy đồ.
* Có thể cập nhật hoặc hủy order.
* Thêm các chú ý khi giặt, lấy và giao đồ
* Người dùng có thể xem giá cả các loại hình thức giặt và đồ giặt.
* Người dùng có thể thay đổi thông tin cá nhân.
* **Washmen**(chỉ áp dụng ở Dubai):
* Khi vào sẽ cần đăng kí số điện thoại, hệ thống gửi mã xác thực, sau đó tiền tên.
* Tại màn hình chính: Có hai hình thức đặt: cách bình thường và cách giao nhận nhanh.
* Sau khi chọn một trong hai cách, cần nhập địa chỉ cụ thể để giao nhận đồ, lựa chọn ngày/giờ giao nhận đồ với hai loại giao tận tay hoặc để đồ ở cửa. Chọn loại đồ để giặt là. Chọn thanh toán và điền các thông tin (chỉ bằng thẻ).
* Trên màn hình chính sẽ hiển thị trạng thái gồm: Đã đặt lịch -> Nhân viên đang trên đường đến lấy -> Đã nhận -> Đang giặt -> nhân viên đang trên đường đến trả -> Hoàn thành.
* Có thể gọi điện hoặc nhắn tin cho nhân viên.
* Trong menu có thể xem giá cả của từng loại đồ giặt.
* **CleanCloud**:
* Khi vào cần đăng kí hoặc đăng nhập. Các thông tin bao gồm: tên, địa chỉ, số điện thoại, postcode, email.
* Vào màn hình chính: Lựa chọn ngày/giờ giao nhận đồ, địa chỉ tự động ánh xạ địa chỉ đã nhập, lựa chọn hình thức giặt(chỉ là, sấy sạch, giặt là, ...) Mỗi hình thức sẽ có các loại đồ kèm giá cả, người dùng lựa chọn và tăng/giảm số lượng. Cuối cùng Submt Order.
* Người dùng chỉ có thể thanh toán bằng hình thức thông qua thẻ.
* Không có thông báo về trạng thái đồ.
* **Justclean**(chỉ áp dụng ở VỊnh Ba Tư):
* Khi vào chưa cần đăng nhập, có thể lựa chọn 2 ngôn ngữ(anh và tiếng mẹ đẻ), chọn khu vực sống.
* Tại màn hình chính: Chọn hãng của tiệm muốn đặt(dựa vào sao đánh giá và comment để người dùng lựa chọn). Mỗi tiệm sẽ có thể tra thông tin các loại đồ có thể đặt kèm giá cả, ngoài ra có 2 loại order là order nhanh và order dạng nhập đồ, mỗi loại Order lại có 2 hình thức dịch vụ nhanh và bình thường(khác biệt về giá cả). Khi chọn mỗi loại sẽ chọn ngày/giờ giao nhận đồ. Khi đồng ý sẽ yêu cầu đăng kí hoặc đăng nhập tài khoản.
* Khách hàng có thể theo dõi vị trí của nhân viên đang tới trong một bản đồ, có thể liên lạc hoặc gửi tin nhắn đến họ.
* Khách hàng có thể đánh giá tiệm giặt.
* Thanh toán bằng thẻ.
* **Laundry Locker**:
* Hình thức kinh doanh: xí nghiệp có những khu là nơi chứa các ngăn đồ với khóa mật mã điện tử. Khi người dùng sử dụng ứng dụng có thể chọn khu mang đồ tới, chọn một ngăn tủ còn trống, hệ thống sẽ gửi mật mã để mở, người dùng mang đồ tới ngăn đó cho đồ vào và khóa lại mật mã đã nhận. Nhân viên của xí nghiệp đi thu đồ sẽ nhập mật mã và lấy đồ về xí nghiệp giặt. Khi nhận lại đồ có hai cách: một là người dùng quay lại ngăn chứa đồ và nhập mã để lấy đồ, hai là giao tận nơi.
* Khi vào cần đăng kí hoặc đăng nhập. Khi đăng kí cần các thông tin: tên, email, số điện thoại. Sau đó cần xác định địa chỉ để giao nhận đồ. Lựa chọn hình thức thanh toán(thẻ, bitcoin)
* Người dùng chọn loại hình muốn giặt, nhân viên sẽ tự sắp xếp và ghi lại đồ đạc có trong ngăn. Khi đồ được giao trả lại ngăn, sẽ có email hoặc thư sms gửi đến, người dùng quay lại nơi đã để đồ để lấy.
* Người dùng được giao đồ tận nhà, vào phần giao tận nơi, lựa chọn ngày/giờ nhận, địa chỉ cụ thể.
* Khi giặt đồ xong sẽ có tin nhắn thông báo về số điện thoại đã đăng kí.

|  |  |  |
| --- | --- | --- |
|  | Phần mềm thiết kế | Phần mềm tương tự |
| Quy mô phần mềm | Hệ thống nhỏ | Hệ thống lớn. |
| Phạm vi áp dụng | Là ứng dụng đầu tiên áp dụng cho xí nghiệp Sapy. | Áp dụng tại một khu vực nhất định của các nước. |
| Công việc | Các chức năng khác biệt:   * Có các hình thức: nhận trả tận nơi, tự mang đồ tới. * Nhân viên có thể cập nhật lại lộ trình tới nhận đồ. | Các chức năng khác biệt:   * Xác thực khi đăng kí. * Thanh toán bằng thẻ. * Chat trên ứng dụng. |

# Bài toán liên quan và giải pháp

## **Bài toán tối ưu: quãng đường và công việc cho nhân viên**

*Tìm lộ trình đường đi ngắn nhất và lập lịch công việc cho Nhân viên thu gom hoặc trả đồ.*

### **Các tham số đầu vào**

* **Quãng đường** do Google map cung cấp. Các quãng đường giữa hai điểm đã được google map tối ưu, sử dụng các quãng đường cùng các thông số khác để tính toán lộ trình tối ưu cho nhân viên.
* **Số lượng đồ** mà nhân viên có thể chở: tính theo cần, chiếc. Mỗi nhân viên chỉ chở được lượng đồ nhất định. Khi được phân công đi thu gom hoặc trả đồ, dựa trên lượng đồ tối đa đó để phân chia công việc hợp lý.
* **Lượng tiền ước lượng** tính theo hóa đơn của khách hàng: Các hóa đơn có giá trị càng cao sẽ càng được ưu tiên hơn.
* **Số lượng nhân viên** của xí nghiệp: ô tô và xe máy. Lượng nhân viên thu gom của xí nghiệp là có hạn, các nhân viên sẽ được chia đều tới các khu vực.
* **Lượng thời gian** làm việc còn lại của nhân viên: Nếu tổng thời gian ước lượng cần làm vượt quá thời gian tối đa còn lại của nhân viên trong ngày để biết thời gian tối đa nhân viên còn làm việc trong ngày. Cần phân công việc cho nhân viên tới các khu vực gần nhất nhưng thu được lợi nhuận cao nhất cho xí nghiệp.

### **Mô tả bài toán**

Nhân viên đi thu gom đồ giặt bằng ô tô hoặc xe máy, mỗi nhân viên muốn đi tới n địa điểm T1, T2, ... Tn đã phân công trên bản đồ. Xuất phát từ một địa điểm nào đó, nhân viên phải đi qua tất cả các địa điểm đã giao để thu gom hoặc trả đồ, mỗi địa điểm đi tới một lần. Tìm lộ trình cho nhân viên sao cho chi phí (thời gian, phí xe cộ) là nhỏ nhất.

### **Thuật toán tham lam**

Phương pháp tham lam là kỹ thuật thiết kế thường được dùng để giải các bài toán tối ưu. Phương pháp được tiến hành trong nhiều bước. Tại mỗi bước, theo một chọn lựa nào đó ( xác định bằng một hàm chọn), sẽ tìm một lời giải tối ưu cho bài toán nhỏ tương ứng. Lời giải của bài toán được bổ sung dần từng bước từ lời giải của các bài toán con.Các lời giải theo phương pháp tham lam thường chỉ là chấp nhận được theo điều kiện nào đó, chưa chắc là tối ưu.

Cho trước một tập A gồm n đối tượng là các địa điểm nhân viên cần tới, ta cần phải chọn một tập con S của A. Với một tập con S được chọn ra thỏa mãn các yêu cầu: {tận dụng thời gian tối đa của nhân viên, lộ trình ngắn nhất, gom hoặc trả lượng đồ nhiều, thu lợi nhuận cao}, ta gọi là một nghiệm chấp nhận được. Cụ thể, xét một hàm mục tiêu : Với lượng nhân viên và phương tiện có hạn, đưa ra được lộ trình ngắn nhất trong khoảng thời gian làm việc tối đa của nhân viên để có thể thu hoặc trả được đồ nhiều nhất cho khách hàng gắn mỗi nghiệm chấp nhận được với một giá trị là lợi nhuận của xí nghiệp. Nghiệm tối ưu là nghiệm chấp nhận được mà tại đó hàm mục tiêu đạt giá trị lớn nhất tức thu được lợi nhuận cao nhất.

Đặc điểm chung của thuật toán tham lam

* Lời giải cần tìm có thể mô tả như là bộ gồm hữu hạn các thành phần thoả mãn điều kiện đã đặt ra, ta phải giải quyết bài toán một cách tối ưu -> hàm mục tiêu
* Để xây dựng lời giải ta có một tập các ứng cử viên là các địa điểm cần đi qua.
* Xuất phát từ lời giải rỗng, thực hiện việc xây dựng lời giải từng bước, mỗi bước sẽ lựa chọn địa điểm trong tập ứng cử viên để bổ xung vào lộ trình hiện tại.
* Xây dựng được một hàm nhận biết được tính chấp nhận được của lộ trình hiện có -> Hàm Solution(S) -> Kiểm tra thoả mãn các yêu cầu.
* Một hàm Select(C) cho phép tại mỗi bước của thuật toán lựa chọn địa điểm có triển vọng nhất để bổ xung vào lộ trình hiện có -> dựa trên căn cứ vào ảnh hưởng của nó vào hàm mục tiêu(lợi nhuận của xí nghiệp).
* Xây dựng hàm nhận biết tính chấp nhận được của địa điểm được lựa chọn, để có thể quyết định bổ xung địa điểm được lựa chọn bởi hàm Select vào lộ trình -> Feasible(S ẩn x).

Sơ đồ thuật toán

\* input A[1..n]

\* output S //lộ trình;

greedy (A,n)

S = 0;

while ( A <> 0)

{

x= Chọn(A); A = A-{x};

if( S U {x} chấp nhận được )

S = S U {x};

Return S;

}

1. **Phân tích, thiết kế thuật toán**:

Xét đồ thị có các đỉnh là các địa điểm trên bản đồ, khoảng cách giữa hai đỉnh là khoảng cách giữa các địa điểm trên bản đồ cùng và thời gian di chuyển, trọng số của các đỉnh là lợi thuận ước lượng tính theo đơn đặt hàng của khách và lượng đồ của khách. Ở đây có hai ràng buộc quan trọng: lượng đồ nhân viên có thể chở, thời gian nhân viên còn phải làm việc.

Input C= (Cij)

output TOUR // Hành trình tối ưu,

Mô tả :

COST;//Chi phí tương ứng

TOUR := 0; COST := 0; v := u; // Khởi tạo

Mọi k := 1 -> n ://Đi qua các địa điểm đã tính toán

// Chọn địa điểm gần kề

- Chọn <v, w> là đoạn nối 2 địa điểm có chi phí(đường đi và thời gian) nhỏ nhất tính từ địa điểm v hiện tại đến các địa điểm chưa qua.

- TOUR := TOUR + <v, w>; //Cập nhật lộ trình

- COST := COST + Cvw ; //Cập nhật chi phí(thời gian, đường đi, lợi nhật)

// Chuyến đi hoàn thành TOUR := TOUR + <v, u>; COST := COST + Cvw

1. **Độ phức tạp thuật toán**

Thao tác chọn đỉnh thích hợp trong n đỉnh được tổ chức bằng một vòng lặp để duyệt. Nên chi phí cho thuật toán xác định bởi 2 vòng lặp lồng nhau, nên T(n) € O (n2).

1. **Cài đặt thuật toán**

B1: Sắp xếp các đỉnh là các địa điểm cần thu/trả đồ theo số tiền ước lượng trên hóa đơn ko giảm.

B2: Tính đơn giá trung bình của đơn hàng trên từng quãng đường bằng Tiền ước lượng của đơn hàng chia cho số km từ vị trí hiện tại tới địa điểm đó(không tính các địa điểm đã đi qua).

B3: Lựa chọn địa điểm thỏa mãn:

* Có đơn giá lớn nhất.
* Tổng lượng đồ thu nhỏ hơn hoặc bằng lượng đồ lớn nhất mà nhân viên có thể chở.
* Tổng thời gian ước lượng đi theo lộ trình nhỏ hơn hoặc bằng thời gian của nhân viên còn có thể làm trong ngày.

B4: Quay lại thực hiện bước 2.

Ta có ma trận a là mảng hai chiều với mỗi a[i][j] (i,j <= n là số địa điểm cần đi qua) là trọng số hay đơn giá trung bình tính bằng số tiền ước lượng của đơn hàng chia cho khoảng cách từ địa điểm i đến địa điểm j.

int GTS (matrix a, int n, int TOUR[max], int Ddau)

{

int v, //đỉnh đang xét

int k, //duyệt qua n đỉnh để chọn

int w; //đỉnh được chọn trong mỗi bước

int mini; //Đơn giá trung bình của đơn hàng

int COST; //Tổng đơn giá trung bình

int daxet[max]; //đánh dấu các đỉnh đã xét.

for(k = 1; k <= n; k++)

daxet[k] = 0; //chưa đỉnh nào được xét

COST = 0; //ban đầu COST == 0

int i; // biến đếm đến n đỉnh thì dừng

v = Ddau; //Chọn đỉnh xuất phát là 1

i = 1;

TOUR[i] = v; //đưa v vào chu trình

daxet[v] = 1; //đỉnh v đã được xét, đánh dấu là 1

while(i < n)

{

mini = 0;//VC

for (k = 1; k <= n; k++)

if(!daxet[k])

if(mini < a[v][k])

{

mini = a[v][k];

w = k;

}

v = w;

i++;

TOUR[i] = v;

daxet[v] = 1;

COST += mini;

}

COST += a[v][Ddau];

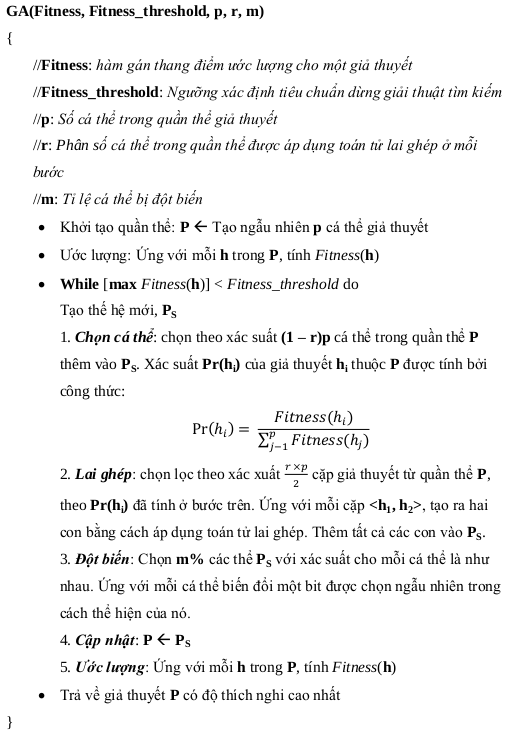
return COST;

}

### **Thuật toán di truyền (tham khảo)**

Bài toán dành cho GAs là tìm kiếm trên không gian các giả thuyết ứng cử  
để xác định giả thuyết tốt nhất. Trong GAs “giả thuyết tốt nhất” được định nghĩa như là một giả thuyết tối ưu hóa một đại lượng số được định nghĩa trước cho bài toán sắp tới, được gọi là độ thích nghi của giả thuyết. Ví dụ, nếu tác vụ học hỏi là bài toán xấp xỉ một hàm chưa biết cho tập mẫu huấn luyện gồm dữ liệu đầu vào và dữ liệu đầu ra, thì độ thích nghi có thể được định nghĩa như là độ chính xác của giả thuyết trên dữ liệu huấn luyện này. Nếu tác vụ là học chiến lược chơi cờ, độ thích nghi có thể là số ván thắng của chiến lược này khi đấu với các chiến lược khác trong quần thể hiện tại.  
Mặc dù các thuật giải di truyền được thực hiện thay đổi theo bài toán cụ thể, nhưng chúng chia sẻ chung cấu trúc tiêu biểu sau: Thuật giải hoạt động bằng cách cập nhật liên tục tập giả thuyết – được gọi là quần thể. Ở mỗi lần lặp, tất cả các cá thể trong quần thể được ước lượng tương ứng với hàm thích nghi. Rồi quần thể mới được tạo ra bằng cách lựa chọn có xác suất các cá thể thích nghi tốt nhất từ quần thể hiện tại. Một số trong những cá thể được chọn được đưa nguyên vẹn vào quần thể kế tiếp. Những cá thể khác được dùng làm cơ sở để tạo ra các cá thể con bằng cách áp dụng các tác động di truyền: lai ghép và đột biến.

1. **Các bước giải thuật cơ bản**

****

*Tìm đường đi ngắn nhất cho khách hàng từ vị trí hiện tại đến tiệm giặt Sapy.*

Giải pháp: sử dụng API của GOOGLE MAP

Các tiệm của xí nghiệp sẽ được hiển thị trên một google map. Dựa vào tọa độ của các tiệm đó và tọa độ do người dùng xác định, sử dụng api của google map để đưa ra 1-3 đường đi Ngắn nhất tới tiệm Gần nhất. Nếu người dùng nhấn chọn một tiệm khác trên bản đồ, hệ thống sẽ đưa ra 1-3 đường đi Ngắn nhất tới tiệm đó.

# Kiến trúc hệ thống và yêu cầu công nghệ

## Kiến trúc hệ thống

Sử dụng mô hình Client - Server.

**Server** sử dụng dịch vụ của Google Firebase.

* Các yêu cầu cần thiết
* Lưu trữ dữ liệu chung trên một server yêu cầu đơn giản hóa các thao tác với dữ liệu. Tài khoản của khách hàng và nhân viên cần lưu trữ, thông tin về đồ giặt và trạng thái cũng cần lưu lại.
* Bảo mật về dữ liệu. Các dữ liệu về người dùng và hóa đơn là quan trọng.
* Dữ liệu được đồng bộ hóa trên các server trên tất cả các client trong thời gian thực. Thông báo trạng thái cho khách về trạng thái của đồ giặt.
* Xác thực người dùng qua Email, google, facebook. Khi người dùng đăng kí tài khoản, nếu người dùng đăng kí bằng google hoặc facebook sẽ cần xác thực: gửi tin nhắn đến mail, hoặc số điện thoại.
* Hỗ trợ gợi ý cài đặt ứng dụng bằng việc chia sẻ với bạn bè.
* Có thể quảng cáo ứng dụng cho xí nghiệp.
* Test ứng dụng trước khi tới tay người dùng.
* Các giải pháp dựa vào Google Firebase
* **Cloud Messaging** là một giải pháp tin nhắn đa nền tảng đáng tin cậy miễn phí.Mỗi tin nhắn dung lượng đến 4KB trong ứng dụng client.
* **Firebase Authentication** Hầu hết các ứng dụng cần xác thực quyền. Giúp ứng dụng lưu dữ liệu an toàn sử dụng trong các đám mây.
* **Firebase Realtime Database** cơ sở dữ liệu đám mây NoSQL đồng bộ hóa. Dữ liệu được đồng bộ hóa trên tất cả các client trong thời gian thực, và luôn có sẵn khi ứng dụng offline.
* **Firebase Storage** được xây dựng cho các nhà phát triển ứng dụng, để lưu trữ và phục vụ nội dung do người dùng tạo ra, chẳng hạn như hình ảnh hoặc video.
* **Firebase Test Lab** cung cấp các thiết bị vật lý và ảo cho phép chạy thử nghiệm mô phỏng môi trường sử dụng thực tế
* **Firebase Crash Reporting** Thông tin toàn diện và hành động để giúp chẩn đoán và sửa chữa các vấn đề trong ứng dụng.
* **Notifications** cho phép gửi và duy trì tin nhắn của người dùng. Sử dụng là miễn phí và không giới hạn.
* **Remote Config** Cập nhật ứng dụng không cần triển khai một phiên bản mới. Cài đặt nhanh chóng dễ dàng.
* **App Indexing** Trợ giúp người dùng khám phá và tái tham gia với các ứng dụng của bạn bằng cách hiển thị có liên quan nội dung trong ứng dụng trong kết quả tìm kiếm của Google.
* **Invites** giới thiệu ứng dụng và chia sẻ.
* **AdWords** Tự động liên kết AdWords với người dùng mà bạn xác định trong căn cứ Firebase Analytics. Cải thiện nhắm mục tiêu quảng cáo và tối ưu hóa hiệu suất chiến dịch của bạn.

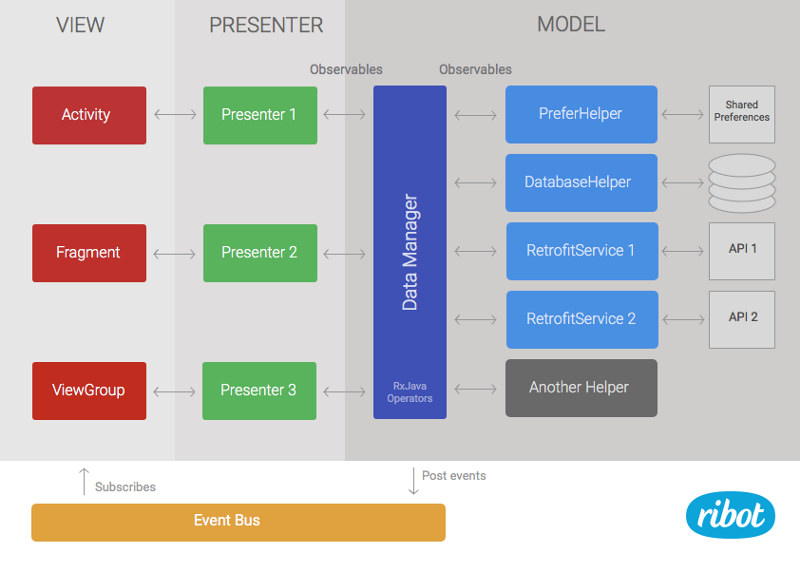
**Client:** lập trình bằng ngôn ngữ javascript, sử dụng Framework React native

* Các yêu cầu cần thiết
* Phát triển trên hệ điều hành: android và iOS.
* Cần phát triển một hệ thống nhanh chóng.
* Giới thiệu về React native
* Một framework cho phép các lập trình viên xây dựng các ứng dụng native mà chỉ sử dụng ngôn ngữ lập trình javascript.
* React native cho phép xây dựng các ứng dụng trên android và ios chỉ với một ngôn ngữ thống nhất là javascript. Không như các framework hybrid khác (viết một lần triển khai nhiều nơi).
* Hỗ trợ Reload ứng dụng trong khi lập trình, không cần mất thời gian build lại cả hệ thống.
* Có các thư viện hỗ trợ thiết kế UI
  + **React Native Animatable:** việc thêm nhanh các hình động và chuyển tiếp đơn giản vào ứng dụng React Native.
  + **React Native Push Notification:**Thư viện này hỗ trợ rất tốt cho các thông báo đẩy cục bộ. Nó có các tính năng như thông báo lịch trình, nhắc lại thông báo dựa trên ngày, tuần, giờ... mà không có trong các thư viện khác.
  + **React Native FCM:**Thư viện này cũng có hỗ trợ cho các thông báo địa phương với sự lập kế hoạch và hỗ trợ định kỳ.
  + **React Native Progress:**Hiển thị tiến độ tải hoặc bất kỳ hành động nào khác là phần quan trọng trong một ứng dụng
  + **React Native Swiper:**React Native swiper rất hữu ích cho việc triển khai giới thiệu ứng dụng, hình ảnh băng chuyền và hình ảnh.
  + **React Native Share:**Chia sẻ thành phần với tùy chỉnh giao diện người dùng
  + **React Native Image Picker:**hỗ trợ lựa chọn từ bộ sưu tập và ảnh chụp từ máy ảnh
  + **React Native Vector Icons:**chứa các thành phần Icon.
  + ...

1. Kiến trúc phần mềm

Sử dụng kiến trúc: MVP

* Giới thiệu về MVP



* Model tổ chức business logic của ứng dụng. Nó kiểm soát dữ liệu có thể được tạo ra, lưu trữ, và sửa đổi.
* View là một giao diện thụ động hiển thị dữ liệu và các tuyến đường hành động dùng cho Presenter.
* Presenter lấy dữ liệu từ Model và cho thấy nó trong các View. Nó cũng xử lý hành động người dùng mong muốn được nó bằng các View.
* Áp dụng vào ứng dụng:
* View là giao diện sử dụng của khách hàng và nhân viên
  + Hiển thị các màn hình chính: trang chủ, giá cả, lịch sử, thông tin cá nhân.
  + Bản đồ, các item, đường đi,... trên bản đồ.
  + Các chức năng đặt lịch, trạng thái.
* Model là dữ liệu được lưu trữ trên server gồm dữ liệu của người dùng như thông tin cá nhân, thông tin đồ giặt.
* Presenter
  + Các hàm tính toán thuật toán
  + Các hàm sử dụng api của google map
  + Các hàm xử lý việc đặt và nhận lịch
  + Các hàm xử lý việc thao tác với google firebase để gửi lấy dữ liệu, cập nhật trạng thái,...

## Yêu cầu công nghệ

* Server: Google Firebase
* Client: sử dụng ngôn ngữ javascript, framework React Native
* Hệ điều hành: Ubuntu hoặc windown 10.
* Công cụ lập trình: VS Code hoặc Sublime text3.
* Bộ giả lập android: Genymotion.

# Tài liệu tham khảo

1. Thuật toán tham lam [*http://voer.edu.vn/c/thuat-toan-tham-lam/018b828c/b658e813*](http://voer.edu.vn/c/thuat-toan-tham-lam/018b828c/b658e813)
2. Thuật toán di truyền: *Đồ án tốt nghiệp " Giải thuật di truyền với bài toán người du lịch" của sinh viên trường Đại Học Bách Khoa Hà Nội năm 2013.*
3. *Dont\_Make\_Me\_Think\_A\_Common\_Sense\_Approach\_to\_Web\_Usability\_2nd\_Ed\_2005*
4. *Design Of Everyday Things*
5. *Uxpin\_mobile\_book\_of\_trends\_2014*
6. *Uxpin\_the\_guide\_to\_wireframing*