Tài liệu kỹ thuật của app BKRM(Bach Khoa Retail Management)

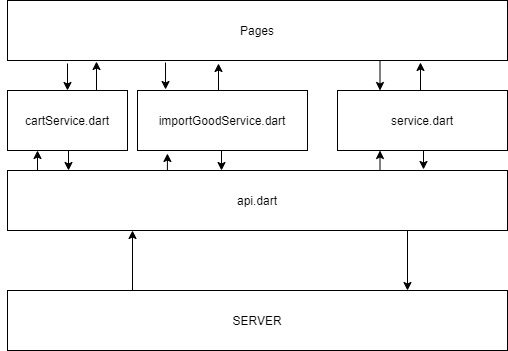
Viết bởi Dương Quang Huy

Ngày cập nhật: 25/05/2021

1. **Giới thiệu về Flutter:**

Flutter là một UI SDK mã nguồn mở được phát triển bởi Google. Vào ngày 04/12/2018 Flutter 1.0 được giới thiệu có thể tạo ra app trên 2 nền tảng Android và iOS chỉ dựa trên 1 codebase. Ngày 03/03/2021, Flutter 2.0 được ra mắt, với bản cập nhật này, Flutter được hỗ trợ chính thức để thiết kế trang web cũng như là có thêm khả năng thiết kế app cho macOS, Windows và Linux.

1. **Cấu trúc của app:**



Hình Cấu trúc chung của app

App sẽ có các lớp sau:

* api.dart : là lớp để thực hiện giao tiếp với server
* service.dart, importGoodService.dart, cartService.dart: là lớp để xử lý các dữ liệu của app, gọi api.dart để đưa dữ liệu lên server, chứa các dữ liệu cần thiết để app có thể hoạt động bình thường
* Pages: bao gồm nhiều widget nhỏ, là giao diện tương tác với người dùng
  1. **Lớp API:**

Được thiết kế theo design pattern Singleton. Lần đầu tiên API được gọi sẽ tiến hành tạo ra 1 instance API. Kể từ đó bất kể được API được tạo từ đâu, instance đã được tạo trước đó sẽ được trả về để sử dụng chứ không tạo ra các instance mới. Việc này để đảm bảo tính duy nhất và duy trì API trong suốt quá trình app hoạt động.

API sẽ thực hiện việc giao tiếp với server qua việc gọi các request POST/GET và lưu trữ thông tin để kết nối đến server

Các method trong API:

* Các method Create:
  + createEmployeeUser: tạo tài khoản nhân viên
  + createCustomer: tạo khách hàng
  + createSupplier: tạo nhà cung cấp
  + createInvoice: tạo đơn bán hàng
  + createNewProduct: tạo sản phâm
  + createCategory: tạo danh mục sản phẩm
  + createImportInvoice: tạo đơn nhập hàng
  + createRefundSheet: tạo đơn trả hàng
  + createRefundPurchaseSheet: tạo đơn hoàn lại hàng đã nhập
  + createShift: tạo ca làm việc
  + createSchedule: tạo lịch trình làm việc của nhân viên
  + createAttendance: điểm danh nhân viên làm việc trong ngày
* Các method Get:
  + getCurrentUserInfo: lấy thông tin của user đang đăng nhập
  + getItemInfoNoPagination: lấy tất cả sản phẩm của cửa hàng mà không sử dụng phân trang
  + getItemInfoPagination: lấy sản phẩm của cửa hàng theo phân trang
  + searchItem: tìm kiếm sản phẩm trong cửa hàng theo các tiêu chí
  + searchDefaultItem: tìm kiếm sản phẩm trong database lưu trữ dữ liệu của thông tin các sản phẩm cùng với barcode đã biết để tự động điền cho người dử dụng khi nhập hàng
  + getDashboardInfo: lấy thông tin của dashboard
  + getCategory: lấy danh sách các danh mục trong của hàng
  + getAllEmployees: lấy danh sách tất cả các tài khoản nhân viên
  + getAllCustomers: lấy danh sách tất cả các khách hàng của cửa hàng
  + getSuppliers: lấy danh sách nhà cung cấp
  + getPurchasedSheet: lấy danh sách các đơn nhập theo phân trang
  + getOnePurchasedSheet: lấy chi tiết đơn nhập hàng
  + getInvoice: lấy danh sách các hóa đơn bán hàng theo phân trang
  + getOneInvoice: lấy chi tiết hóa đơn bán hàng
  + getRefundSheet: lấy danh sách đơn trả hàng theo phân trang
  + getOneRefundSheet: lấy chi tiết của đơn trả hàng
  + getReturnPurchasedSheet: lấy danh sách các đơn trả hàng đã được nhập theo phân trang
  + getOneReturnPurchasedSheet: lấy chi tiết của đơn trả hàng nhập
  + getShifts: lấy danh sách các ca của cửa hàng
  + getSchedules: lấy danh sách các lịch trình làm việc của nhân viên
  + getAttendances: lấy danh sách điểm danh trong ngày
  + getAttendancesFromDateToDate: lấy chi tiết điểm danh của 1 nhân viên theo ngày
  + getReportRevenue: lấy các báo cáo thông kê về danh thu – chi phí – lợi nhuận – vốn theo thời gian
  + getReportItems: lấy thông tin thống kê về sản phẩm
  + getReportCategories: lấy thông tin về thống kê danh mục sản phẩm
  + getReportCustomers: lấy thông tin về thống kê khách hàng
  + getReportSupplier: lấy thông tin về thống kê nhà cung cấp
* Các method Edit:
  + editItems: chỉnh sửa sản phẩm
  + editCurrentUser: chỉnh sửa thông tin của người dùng hiện đang đăng nhập
  + editCustomer: chỉnh sửa thông tin khách hàng
  + editSupplier: chỉnh sửa thông tin nhà cung cấp
  + editCategory: chỉnh sửa thông tin danh mục
  + editShift: chỉnh sửa thông tin ca làm việc
  + editBranch: chỉnh sửa thông tin chi nhánh cửa hàng
  + resetPasswordForOwner: đặt lại password cho tài khoản nhân viên (chỉ có owner được sử dụng)
  + editEmployeeForOwner: chỉnh sửa thông tin của nhân viên (chỉ có owner được sử dụng)
* Các method Delete:
  + deleteItem: xóa sản phẩm
* Các method khác:
  + refreshToken: yêu cầu lấy token mới từ server khi token cũ hết hạn
  + login: thực hiện đăng nhập để lấy token và thông tin người dùng từ server
  + signUp: thực hiên đăng ký tài khoản owner mới
  + logout: thực hiện đăng xuất tài khoản hiện đang đăng nhập
  + changePassword: thay đổi mật khẩu của người dùng đang đăng nhập
  1. **Lớp Service:**
     1. Service.dart:

Là service chính chứa phần lớn các chức năng. Service này cũng được thiết kế theo Singleton để đảm bảo tính duy nhất và duy trì trong suốt thời gian app hoạt động.

Trong service này chứa các thông tin về người dùng đang đăng nhập hiện tại, các giỏ hàng, service của nhập hàng, thông tin của local db, …

Các method trong Service:

* Các method Create:
  + createEmployeeUser: tạo tài khoản nhân viên
  + createCustomer: tạo khách hàng
  + createSupplier: tạo nhà cung cấp
  + createNewProduct: tạo sản phâm
  + createCategory: tạo danh mục sản phẩm
  + createRefundSheet: tạo đơn trả hàng
  + createRefundPurchaseSheet: tạo đơn hoàn lại hàng đã nhập
  + createShift: tạo ca làm việc
  + createSchedule: tạo lịch trình làm việc của nhân viên
  + createAttendance: điểm danh nhân viên làm việc trong ngày
* Các method Get:
  + getAllItemForReserved: lấy danh sách tất cả các sản phẩm và lưu lại để phục vụ cho hoạt động offline
  + filterStoredList: lọc các sản phgẩm đã được lưu (hoạt động trong chế độ offline)
  + getStoredListItem: lấy danh sách các sản phẩm đã được lưu
  + getAllItem: lấy danh sách sản phẩm theo phân trang
  + getItems: lấy danh sách các sản phẩm theo itemId
  + searchItemInBranch: tìm sản phẩm trong cửa hàng thoe barcode hoặc theo priceId
  + searchItemInDefaultDb: tìm sản phẩm trong database lưu dữ liệu mẫu của sản phẩm
  + getDashboardInfo: lấy thông tin của dashboard
  + getSupplier: lấy danh sách các nhà cung cấp
  + getCategory: lấy danh sách các danh mục sản phẩm
  + getCustomer: lấy danh sách khách hàng
  + getEmployee: lấy danh sách tài khoản nhân viên
  + getImportInvoices: lấy danh sách đơn nhập hàng
  + getDetailPurchasedSheet: lấy chi tiết đơn nhập hàng
  + getInvoices: lấy danh sách hóa đơn bán hàng
  + getDetailInvoice: lấy chi tiết hóa đơn bán hàng
  + getRefundSheets: lấy danh sách đơn trả hàng
  + getReturnPurchasedSheet: lấy danh sách đơn trả hàng đã nhập
  + getDetailRefundSheet: lấy chi tiết đơn trả hàng
  + getDetailReturnPurchasedSheet: lấy chi tiết đơn trả hàng đã nhập
  + getShifts: lấy danh sách ca làm việc
  + getSchedules: lấy danh sách lịch trình làm việc
  + getAttendancesFromDateToDate: lấy danh sách điểm danh của nhân viên theo ngày
  + getCharts: lấy số liệu thống kê về doanh thu – lợi nhuận – chi phí – vốn theo ngày
  + getReportItems: lấy dữ liệu thống kê về sản phẩm
  + getReportCategories: lấy dữ liệu về thống kê danh mục
  + getReportCustomer: lấy dữ liệu về thống kê khách hàng
  + getReportSupplier: lấy dử liệu về thống kê nhà cung cấp
* Các method Edit:
  + editProduct: chỉnh sửa sản phẩm
  + editCurrentUser: chỉnh sửa thông tin người dùng đang đăng nhập
  + editCustomer: chỉnh sửa thông tin khách hàng
  + editSupplier: chỉnh sửa thông tin nhà cung cấp
  + editCategory: chỉnh sửa thông tin danh mục sản phẩm
  + editShift: chỉnh sửa thông tin ca làm việc
  + editBranch: chỉnh sửa thông tin chi nhánh
  + editEmployeeInfo: chỉnh sửa thông tin nhân viên
  + resetPasswordUserForOwner: thay đổi mật khẩu nhân viên (chỉ sử dụng cho owner)
* Các method delete:
  + deleteItem: xóa sản phẩm
* Các method khác:
  + networkAvailableCheck: kiểm tra xem điện thoại có kết nối wifi hoặc mạng di động không
  + requestCart: đẩy thông tin về giỏ hàng hiện tại vào stream
  + initDb: khai tạo local database
  + getLastLogInUser: lấy thông tin người dùng đăng nhập lần cuối
  + storeCurrentUserLoggedIn: lưu thông tin người dùng đăng nhập lần cuối vào local database
  + clearLastUserLoggedIn: xóa người dùng đăng nhập lần cuối
  + addDeliverNameToSupplier: thêm tên người vận chuyển vào local database
  + getDeliverNameOfSupplier: lấy tên người vận chuyền theo nhà cung cấp từ local database
  + storeDataForThisSession: lưu các dữ liệu về sản phẩm, khách hàng, các request đã được gọi vào local database
  + getDataFromLastSessionForThisSession: lấy các dữ liệu về sản phẩm, khách hàng, các request đã được gọi từ local database
  + addRequestToQueue: thêm request vào trong danh sách được lưu vào database
  + addAllRequestToQueue: thêm một list các request trong danh sách lưu trong local database
  + getQueuedRequest: lấy tất cả các request đã được lưu từ database
  + removeAllQueuedRequest: xóa tất cả các request đã được lưu
  + getLastSessionUser: lấy thông tin người dùng đăng nhập lần cuối
  + checkToChangeToOfflineMode: kiểm tra kết nối mạng để nhảy sang chế độ offline hoặc ngược lại
  + prepareDataForOffline: chuẩn bị lấy dữ liệu từ server để lưu trong local database cho chế độ offline
  + generateSortListForItem: tạo các danh sách đã được sắp xếp săn, phục vụ cho lấy sản phẩm khi offline
  + signUp: đăng ký tài khoản owner
  + logInUser: đăng nhập người dùng vào hệ thống
  + logout: dăng xuất người dùng khỏi hệ thống
  + changePassword: thay đổi mật khẩu cho người dùng hiện tại
    1. CartService.dart:

Đây là service để phục vụ cho giỏ hàng. Mỗi service này sẽ phục vụ cho một giỏ hàng. Service này lưu các thông tin về khách hàng đang phục vụ, các sản phẩm trong giỏ, các thông tin khác của giỏ hàng …

Các method trong service này:

- checkCartValid: kiểm tra tính hợp lệ của các sản phẩm trong giỏ và cập nhật thông tin sản phẩm nếu có thay đổ

- calculateAllValueInCart: tính toán các thông tin của giỏ hàng như tổng giá, tổng khuyến mãi…

- useCustomerPoint: áp dụng điểm khách hàng cho giỏ hàng

- addCartItem: thêm sản phẩm vào giỏ hàng

- getCartItem: lấy 1 sản phẩm trong giỏ

- clearCart: xóa tất cả thông tin trong giỏ

- removeCartItem: xóa 1 sản phẩm trong giỏ

- modifyCartItem: thay đổi thông tin ủa 1 sản phẩm trong giỏ

- sendInvoice: thực hiện thanh toán giỏ hàng và gửi hóa đơn cho server

* + 1. importGoodService.dart:

Đây là service để phục vụ cho việc nhập hàng. Service này chứa các thông tin như sản phẩm được nhập, thông tin nhà cung cấp…

Các method trong service này:

- updateInfo: dùng để đẩy thông tin của nhập hàng cho các stream

- addToImport: thêm một sản phẩm mới vào nhập hàng

- replaceItem: thay thế một sản phẩm trong giỏ nhập

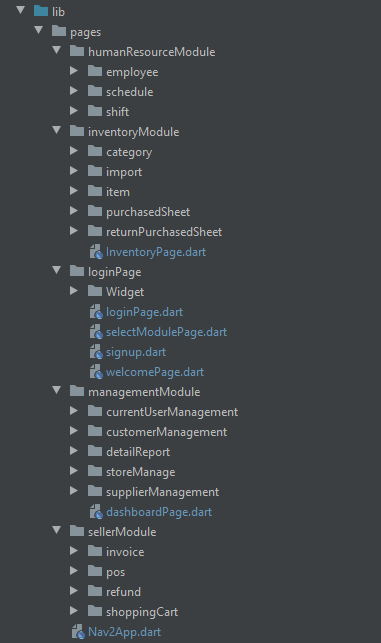
- deleteItem: xóa một sản phẩm trong giỏ nhập

- exportInvoice: tạo ra đơn nhập hàng

- sendInvoice: gửi đơn nhập hàng cho server

- clearImportService: xóa tất cả thông tin của nhập hàng

* 1. **Các trang giao diện:**



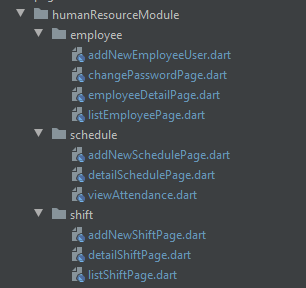
Hình Cấu trúc các trang giao diện

Cấu trúc của các trang giao diện bao gồm 4 module chính:

* humanResourceModule: chứa các trang giao diện liên quan đến nhân sự
* inventoryModule: chứa các trang giao diện liên quan đến kho hàng
* managementModule: chứa các trang liên quan đến quản lý
* sellerModule : chứa các trang liên quan đến bán hàng

Ngoài ra còn có loginPage chứa các trang liên quan đến đăng nhập, đăng ký và chọn module lúc mới vào app

* + 1. Human Resource Module :



Hình Các trang trong humanReouceModule

Trong module này có 3 thư mục con:

- employee: chứa các trang liên quan đến nhân viên như:

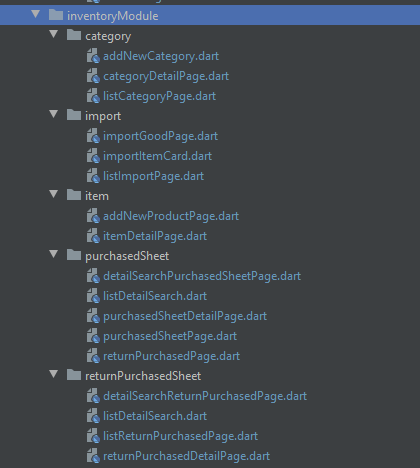
* addNewEmployeeUser: trang tạo tài khoản nhân viên mới
* changePasswordPage: trang thay đổi mật khẩu của nhân viên (chỉ dùng cho owner)
* employeeDetailPage: trang xem thông tin chi tiết của nhân viên
* listEmployeePage: trang xem danh sách các nhân viên trong cửa hàng

- schedule: chưa các trang liên quan đến lịch trình làm việc của nhân viên như:

* addNewSchedulePage: trang tạo lịch trình mới
* detailSchedulePage: trang xem chi tiết lịch trình
* viewAttendance: xem điểm danh của nhân viên

- shift: chứa các trang liên quan đến ca làm việc của cửa hàng như:

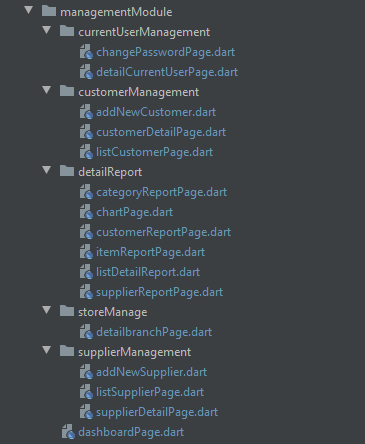
* addNewShiftPage: trang thêm ca làm việc mới
* detailShiftPage: trang xem chi tiết ca làm việc
* listShiftPage: trang xem danh sách các ca làm việc trong cửa hàng
  + 1. Inventory Module:



Hình Các trang trong inventoryModule

Trong module này chứa 5 thư mục con:

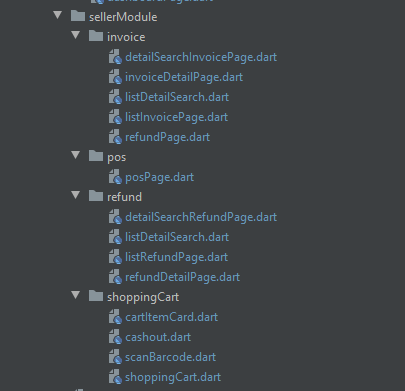
* Category: chứa các trang liên quan đến danh mục sản phẩm như:
  + addNewCategory: trang tạo mới danh mục
  + categoryDetailPage: trang xem chi tiết danh mục
  + listCategoryPage: trang xem tất cả các danh mục trong cửa hàng
* import: chứa các trang liên quan đến nhập hàng
  + importGoodPage : trang chính của giao diện nhập hàng
  + importItemCard: chứa thẻ sản phẩm trong trang nhập hàng
  + listImportPage: trang xem các sản phẩm trong kho để thêm vào danh sách nhập hàng
* item: chứa các trang liên quan đến sản phẩm như:
  + addNewProductPage: trang tạo mới sản phẩm
  + itemDetailPage: trang xem chi tiết sản phẩm
* purchasedSheet : chứa các trang liên quan đến đơn nhập hàng :
  + detailSearchPurchasedSheetPage : trang tìm kiếm chi tiết các đơn nhập
  + listDetailSearch: trang liệt kê các đơn nhập khi tìm kiếm chi tiết
  + purchasedSheetDetailPage: trang xem thông tin chi tiết đơn nhập
  + purchasedSheetPage: trang xem danh sách các đơn nhập trong cửa hàng
  + returnPurchasedPage: trang trả lại hàng đã nhập
* returnPurchasedSheet:
  + detailSearchReturnPurchasedPage: trang tìm kiếm chi tiết đơn trả hàng đã nhập
  + listDetailSearch: trang liệt kê các đơn trả hàng đã nhập khi tìm kiếm chi tiết
  + listReturnPurchasedPage: trang xem danh sách các đơn trả hàng đã nhập
  + returnPurchasedDetailPage: trang thông tin chi tiết đơn trả hàng đã nhập
    1. Management Module:



Hình Các trang trong managementModule

Trong module này chứa 5 thư mục con:

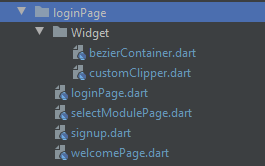
* currentUserManagement: chứa các trang liên quan đến quản lý người dùng đang đăng nhập
  + changePasswordPage: trang để thay đổi mật khẩu của người dùng đang đăng nhập
  + detailCurrentUserPage: trang xem thông tin chi tiết của người dugn2 đang đăng nhập
* customerManagement: chứa các trang liên quan đến khách hàng:
  + addNewCustomer: trang để tạo khách hàng mới
  + customerDetailPage: trang để xem thông tin chi tiết khách hàng
  + listCustomerPage: trang chứa danh scah1 các khách hàng của cửa hàng
* detailReport: chứa các trang liên quan đến thông kê chi tiết
  + categoryReportPage: trang xem thông kê về danh mcụ sản phẩm
  + chartPage: trang xem thống kê về doanh thu, chi phí, vốn, lợi nhuận
  + customerReportPage: trang xem thống kê về khách hàng
  + itemreportPage: trang xem thống kê về sản phẩm
  + listDetailReport: trang xem các loại thống kê đang có
  + supplierReportPage: trang xem thống kê về nhà cung cấp
* storeManage : chứa trang về quản lý cửa hàng
  + detailBranchPage : trang xem thông tin chi tiết của cửa hàng
* supplierManagement: chứa các trang về nhà cung cấp
  + addNewSupplier : trang thêm nhà cung cấp mới
  + listSupplierPage: trang xem danh sách các nhà cung cấp có trong cửa hàng
  + supplierDetailPage: trang xem thông tin chi tiết của nhà cung cấp
* dashboardPage : trang chứa dashboard có các thông tin cơ bản của cửa hàng
  + 1. Seller Module :



Hình 6 Các trang trong sellerModule

Trang module này chứa :

* Invoice: chứa các trang liên quan đến hóa đơn bán hàng
  + detailSearchInvoicePage: trang để tìm kiếm chi tiết hóa đơn bán hàng
  + invoiceDetailPage: trang xem thông tin chi tiết hóa đơn bán hàng
  + listDetailSearch: trang hiện danh sách các hóa đơn bán hàng khi tìm kiếm chi tiết
  + listInvoicePage: trang chứa danh sách các hóa đơn bán hàng trong cửa hàng
  + refundPage: trang để trả hàng từ hóa đơn
* pos: chứa trang hiện các sản phẩm trong kho để bán hàng
  + posPage: trang hiện các sản phẩm trong cửa hàng để có thể thêm vào giò hàng
* refund: chứa các trang liên quan đến đơn trả hảng đã bán
  + detailSearchRefundPage: trang tìm kiếm chi tiết các đơn trả hàng
  + listDetailSearch: trang chứa các danh sách đơn trả hàng sau khi tìm kiếm chi tiết
  + listRefundPage: trang chứa các đơn trả hàng của cửa hàng
  + refundDetailPage: trang xem chi tiết đơn trả hàng
* shoppingCart: chứa các trang liên quan đến giỏ hàng:
  + cartItemCard: thẻ sản phẩm của giỏ hàng
  + cashout: trang tính tiến giò hàng (đã merge với trang shoppingCart)
  + scanBarcode: trang quét liên tục mã vạch để thêm sản phẩm vào giỏ hàng
  + shoppingCart: trang giỏ hàng chính
    1. Login module:



Hình 7 Các trang trong login module

Trong module này chứa các trang liên quan đến hoạt động đăng nhập/ đăng xuất:

* loginPage: trang đăng nhập
* selectModulePage : trang hiển thị các module người dùng có thể đăng nhập
* signup : trang đăng ký
* welcomePage : trang mở đầu khi vào app (đã loại bỏ và cho app vào thằng trang đăng nhập)
* Widget : chứa các widget hổ trợ cho login page

1. **Luồng hoạt động của các chức năng:**
   1. **Các chức năng CRUD:**

Chức năng create:

* Bao gồm có: tạo khách hàng, tạo tài khoản nhân viên, tạo sản phẩm ….
* Các trang này sẽ lấy các thông tin mà người dùng nhập trên màn hình sau đó kiểm tra tính hợp lệ của dữ liệu, thông tin đấy sẽ được gửi xuống lớp service. Lớp service này sẽ kiểm tra các lại các quyền của người dùng cũng như tình trạng mạng, nếu đầy đủ thì sẽ đóng gói thành Map và đưa xuống lớp Api. Lớp Api sẽ tiến hàng gửi cho server và trả lại response từ server cho lớp service. Lớp service sẽ theo response đấy và phàn hồi lại thông tin cho các trang đã gọi.

Chức năng read:

* Bao gồm có: xem thông tin chi tiết, hiển thị danh sách …
* Các trang chi tiết thường sẽ được đưa thông tin từ các trang danh sách hoặc tự lấy thông tin này bằng cách gọi service
* Các trang hiển thị theo danh sách thì sẽ lấy thông tin từ service, tuy nhiên tùy theo việc dử liệu đấy có được phân trang không mà dữ liệu trả về sẽ có sự khách nhau

Chức năng update:

* Bao gồm có: chỉnh sửa thông tin khách hàng, nhà cung cấp, sản phẩm…
* Chức năng này thường sẽ năm chung trong trang hiển thị chi tiết, chỉ khác ở việc chỉnh sửa sẽ gọi hàm chức năng khác và có truyền dữ liệu cho service

Chức năng delete:

* Hiện tại chỉ có xóa sản phẩm
* Xóa sản phẩm sẽ gọi cùng hạm editProduct, chỉ khác ở việc truyền tham số deleted = 1
  1. **Giỏ hàng:**

Trong lớp service có chứa nhiều cartService, mỗi cartService sẽ đại diện cho một giỏ hàng đang hoạt động. Nếu service ko chứa bất kỳ giỏ hàng nào thì khi bất kỳ trang nào gọi giỏ hàng hiện tại thì service sẽ tự động tạo một giỏ hàng mới và trả cho trang đã gọi. Giỏ hàng hiện tại sẽ được chứa trong 1 biến của service khi đổi giỏ hàng thì biến này sẽ thay đổi và chỉ đến giỏ hàng muốn đổi.

Mỗi giỏ hàng sẽ chứa các thông tin đầy đủ để một giỏ hàng hoạt động như khách hàng, sản phẩm trong giỏ. Khi tương tác với giỏ hàng nào đó thì ta sẽ gọi hàm trong instance của giỏ hàng tương ứng. Khi thanh toán thì ta cũng gọi hàm thanh toán hóa đơn trong instance giỏ hàng tương ứng để gửi hóa đơn cho server

* 1. **Nhập hàng:**

Nhập hàng tương tự với giỏ hàng, chỉ khác một điểm là service chỉ chứa 1 instance của importGoodService do app không hiện thực chức năng có nhiều giỏ nhập hàng.

Khi muốn tiến hành nhập hàng thì ta cũng gọi hàm chức năng trong instance nhập hàng để gửi đơn nhập hàng đến server.

* 1. **Chức năng hoạt đông offline:**

Trong qua trình hoạt động của app, app sẽ liên tục lấy các dữ liệu về sản phẩm, khách hàng lưu trữ tạm trong các biến của lớp service để khi app mất mạng trong quá trình hoạt động, thì sẽ chuyển sang chế độ offline và lấy các dữ liệu này để hoạt động cho chức năng bán hàng và đồng thời lưu các dữ liệu đấy vào local database để chuẩn bị cho hoạt động khi không có mạng ngay từ lúc đăng nhập.