



**Bản ghi chép sửa đổi và xác nhận**

**Ghi chép thay đổi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tác giả** | **Phiên bản** | **Các thay đổi liên quan** | **Ngày** |
| Võ Đăng Thuận | 0.0.1 | Hoàn thành | 22/11/2023 |
| Phùng Trần Đăng Khôi | 0.0.2 | Hoàn thành | 23/11/2023 |

**Người kiểm tra**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Doanh nghiệp** | **Phiên bản** | **Chức vụ** | **Ngày** |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

Nội dung

[1.](#_heading=h.gjdgxs) Giới thiệu 5

[1.1.](#_heading=h.30j0zll) Mục đích 5

[1.2.](#_heading=h.1fob9te) Phạm vi 5

[1.3.](#_heading=h.tyjcwt) Đối tượng liên quan và tổ chức tài liệu 5

[Tài liệu liên quan 6](#_heading=h.3dy6vkm)

[2.](#_heading=h.3j2qqm3) Yêu cầu chức năng 7

[2.1.](#_heading=h.4d34og8) Use Case Description 7

[UC1: Đăng ký (cho nhân viên) 7](#_heading=h.1y810tw)

[UC2: Đăng ký (cho khách hàng) 9](#_heading=h.49x2ik5)

[UC3: Đăng nhập 11](#_heading=h.2p2csry)

[UC4: Quên mật khẩu 13](#_heading=h.4f1mdlm)

[UC5: Tuyển dụng 15](#_heading=h.2u6wntf)

[UC6: Tính lương 19](#_heading=h.nmf14n)

[UC7: Sa thải 21](#_heading=h.37m2jsg)

[UC8: Nhắn tin giữa các phòng ban 24](#_heading=h.1mrcu09)

[UC9: Nhắn tin giữa khách hàng và người bán 26](#_heading=h.46r0co2)

[UC10: CRUD đơn vị vận chuyển 30](#_heading=h.2lwamvv)

[UC11: CRUD Thông tin người bán 35](#_heading=h.111kx3o)

[UC12: Xem thông tin người bán 39](#_heading=h.3l18frh)

[UC13: CRUD Quảng cáo 40](#_heading=h.206ipza)

[UC14: Xem quảng cáo (khách hàng) 46](#_heading=h.4k668n3)

[UC15: CRUD nghiệp vụ kế toán 48](#_heading=h.2zbgiuw)

[UC16: CRUD định khoản 52](#_heading=h.1egqt2p)

[UC17: Lập bảng cân đối tài khoản 58](#_heading=h.3ygebqi)

[UC18: Lập bảng cân đối kế toán 61](#_heading=h.2dlolyb)

[UC19: Lập báo cáo kết quả hoạt động kinh doanh 63](#_heading=h.3cqmetx)

[UC20: CRUD Ngành hàng 66](#_heading=h.4bvk7pj)

[UC21: CRUD Loại hàng 71](#_heading=h.2r0uhxc)

[UC22: CRUD Thuộc tính khác biệt nội sản phẩm 76](#_heading=h.1664s55)

[UC23: CRUD Hàng hóa 81](#_heading=h.3q5sasy)

[UC24: Đánh giá 87](#_heading=h.25b2l0r)

[UC25: Thêm/xóa Giỏ hàng 89](#_heading=h.kgcv8k)

[UC26: Thêm/xóa wish list 94](#_heading=h.34g0dwd)

[UC27: CRUD đơn vị thanh toán 97](#_heading=h.1jlao46)

[UC28: CRUD Đơn vị thanh toán (khách hàng) 102](#_heading=h.43ky6rz)

[UC29: CRUD thông tin cá nhân (Khách hàng) 105](#_heading=h.2iq8gzs)

[UC30: CRUD địa chỉ giao hàng (khách hàng) 109](#_heading=h.xvir7l)

[UC31: Tính điểm khách hàng 116](#_heading=h.3hv69ve)

[UC32: Thêm/Xóa khỏi blacklist 118](#_heading=h.1x0gk37)

[UC33: Xác nhận đơn hàng 121](#_heading=h.4h042r0)

[UC34: Xác nhận đơn hàng (Giao bên vận chuyển) 124](#_heading=h.2w5ecyt)

[UC35: Hủy đơn hàng (bên bán) 127](#_heading=h.1baon6m)

[UC36: Hủy đơn hàng (bên mua) 130](#_heading=h.3vac5uf)

[UC37: Tạo đơn hàng (Khách hàng) 133](#_heading=h.2afmg28)

[UC38: CRUD khuyến mãi 139](#_heading=h.pkwqa1)

[UC39: CRUD voucher 145](#_heading=h.39kk8xu)

[2.2.](#_heading=h.1opuj5n) Danh sách mô tả 151

[a.](#_heading=h.35nkun2) Mô tả hiển thị (view): 151

[4.](#_heading=h.44sinio) Yêu cầu phi chức năng 151

[User Access and Security 151](#_heading=h.2jxsxqh)

[Performance Requirements 152](#_heading=h.4i7ojhp)

[Yêu cầu triển khai (implementation) 153](#_heading=h.2xcytpi)

[5.](#_heading=h.1ci93xb) Các yêu cầu khác 154

[Archive Function 154](#_heading=h.3whwml4)

[Security Audit Function 154](#_heading=h.2bn6wsx)

[6.](#_heading=h.qsh70q) Các yêu cầu hệ thống 154

[SharePoint Sites 154](#_heading=h.3as4poj)

[SharePoint Lists 154](#_heading=h.1pxezwc)

[SharePoint Permission Levels 155](#_heading=h.147n2zr)

[SharePoint Groups 155](#_heading=h.3o7alnk)

[SharePoint Web Parts 155](#_heading=h.23ckvvd)

[Custom Pages 156](#_heading=h.48pi1tg)

[Scheduled Agents 156](#_heading=h.32hioqz)

[Technical Concern 156](#_heading=h.1hmsyys)

[7.](#_heading=h.41mghml) Phụ lục 156

[Glossary 156](#_heading=h.2grqrue)

[Messages 157](#_heading=h.3fwokq0)

[Issues List 159](#_heading=h.28h4qwu)

# Giới thiệu

# Mục đích

Tài liệu mô tả chi tiết và thiết kế yêu cầu phần mềm (SRS document) này chứa các yêu cầu để tiến hành xây dựng phần mềm Cung cấp dịch vụ mua bán trực tuyến trên nền tảng di động (Android) và các thiết kế chi tiết cho ứng dụng.

Tài liệu này cùng với các tài liệu khác như tài liệu mô tả chi tiết, tài liệu danh sách chức năng bao gồm các yêu cầu hoàn chỉnh để tiến hành xây dựng ứng dụng.

Cuối cùng, tài liệu này sẽ xác định về mặt kỹ thuật, tức cách ứng dụng sẽ hoạt động. Từ đó các lập trình viên, kỹ sư phần mềm có thể căn cứ vào tài liệu này để tiến hành lập kế hoạch phát triển, phân công nhiệm vụ và triển khai xây dựng ứng dụng.

# Phạm vi

Tài liệu này được chuẩn bị để sử dụng cho phần mềm Cung cấp dịch vụ mua bán trực tuyến, được xây dựng để sử dụng trên thiết bị di động, nằm trong dự án xây dựng phần mềm Cung cấp dịch vụ mua bán trực tuyến (PSS).

# Đối tượng liên quan và tổ chức tài liệu

Tài liệu này liên quan đến các đối tượng sau:

* Đội ngũ phát triển: Chịu trách nhiệm phát triển chi tiết thiết kế, triển khai và thực hiện unit testing, kiểm tra tích hợp (integration testing) và kiểm tra hệ thống cho ứng dụng đã xây dựng
* Đội ngũ xây dựng tài liệu: Chịu trách nhiệm viết hướng dẫn sử dụng cho ứng dụng.
* Đội ngũ kiểm thử chấp nhận người dùng (UAT): Chịu trách nhiệm tiến hành, chỉ đạo các phiên kiểm tra chấp thuận của người dùng (user’s acceptance) đối với người dùng cuối.

Bên dưới là các phần của tài liệu:

* **1. Giới thiệu**: Phần này mô tả giới thiệu tổng quan về tài liệu này.
* **2. Các yêu cầu chức năng**: Phần này mô tả chi tiết các yêu cầu chức năng.
* **3. Các yêu cầu phi chức năng:** Phần này mô tả các yêu cầu phi chức năng của ứng dụng như là các yêu cầu về giao diện, hiệu suất ứng dụng, bảo mật ứng dụng.
* **4. Các yêu cầu khác:** Phần này mô tả các yêu cầu khácnhư lưu trữ hoặc chức năng kiểm tra các bảo mật.
* **5.** **SharePoint Application Design:** This section describes the design of SharePoint application.
* **6. Phụ lục**: Phần này của tài liệu mô tả thêm các yêu cầu khác cho ứng dụng và các thông tin hỗ trợ cho tài liệu này, mô tả chi tiết các từ được viết tắt sử dụng trong tài liệu này.

# Tài liệu liên quan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Title** | **Version** | **File Name / Link** | **Description** |
| 1 | ***FunctionList*** | 0.0.1 |  |  |
| 2 | ***Công báo 279 + 280 - Thông tư 200*** | 0.0.1 |  |  |
| 3 | ***Công báo 281 + 282 - Thông tư 200*** | 0.0.1 |  |  |
| 4 | ***Công báo 283 + 284 - Thông tư 200*** | 0.0.1 |  |  |
| 5 | ***Công báo 285 + 286 - Thông tư 200*** | 0.0.1 |  |  |
| 6 | ***Công báo 287 + 288 - Thông tư 200*** | 0.0.1 |  |  |
| 7 | ***Công báo 289 + 290 - Thông tư 200*** | 0.0.1 |  |  |
| 8 | ***BRD\_PSS*** | 0.0.1 |  |  |
| 9 | ***MẪU BẢNG CÂN ĐỐI TÀI KHOẢN THEO THÔNG TƯ 200*** | 0.0.1 |  |  |
| 10 | ***TÍNH BẢNG CÂN ĐỐI KẾ TOÁN*** | 0.0.1 |  |  |
| 11 | ***TÍNH BÁO CÁO KẾT QUẢ HOẠT ĐỘNG KINH DOANH*** | 0.0.1 |  |  |
| 12 | ***View Description*** | 0.0.1 |  |  |
| 13 | ***List Description*** | 0.0.1 |  |  |
| 14 | ***UI\_Wireframe*** | 0.0.1 |  |  |

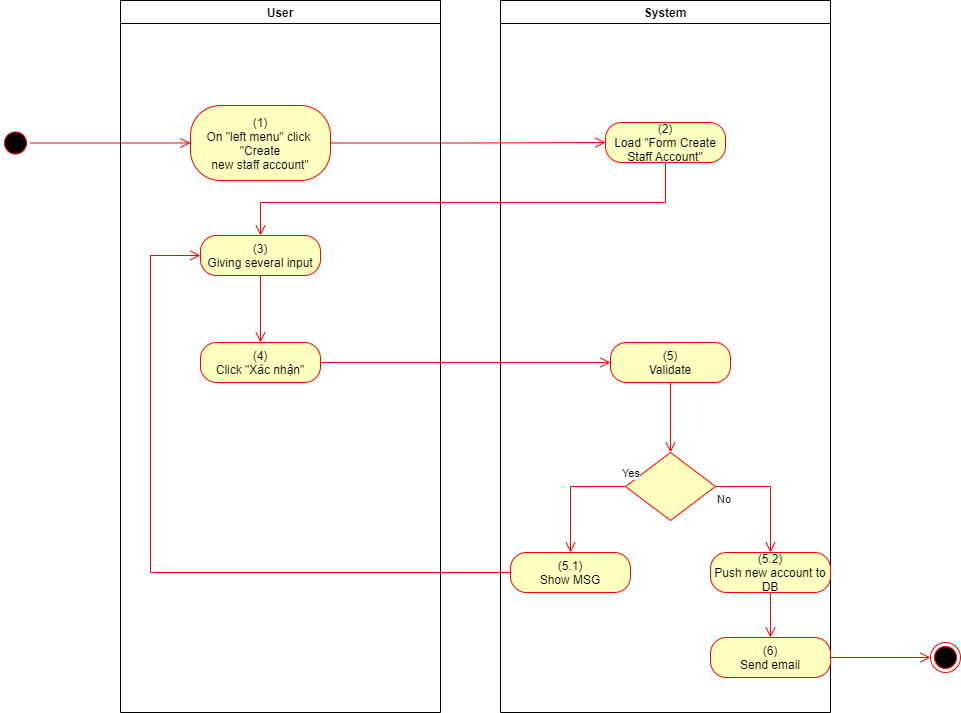
# Yêu cầu chức năng

# Use Case Description

### UC1: Đăng ký (cho nhân viên)

|  |  |
| --- | --- |
| **Name** | **Đăng ký (cho nhân viên)** |
| **Description** | Admin của Công ty được PSS cung cấp dịch vụ sẽ tạo tài khoản cho nhân viên Công ty. |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi BGD click vào button “Create new staff account” ở màn hình “Account Management” |
| **Pre-condition** | * User phải là “BGD” hoặc thuộc ”Phòng nhân sự” |
| **Post-condition** | * Tài khoản staff mới được tạo và được gửi vào hộp thư email lúc đăng ký. |

#### Activities Flow



*Figure 1: Sign up (satff) Activities Flow*

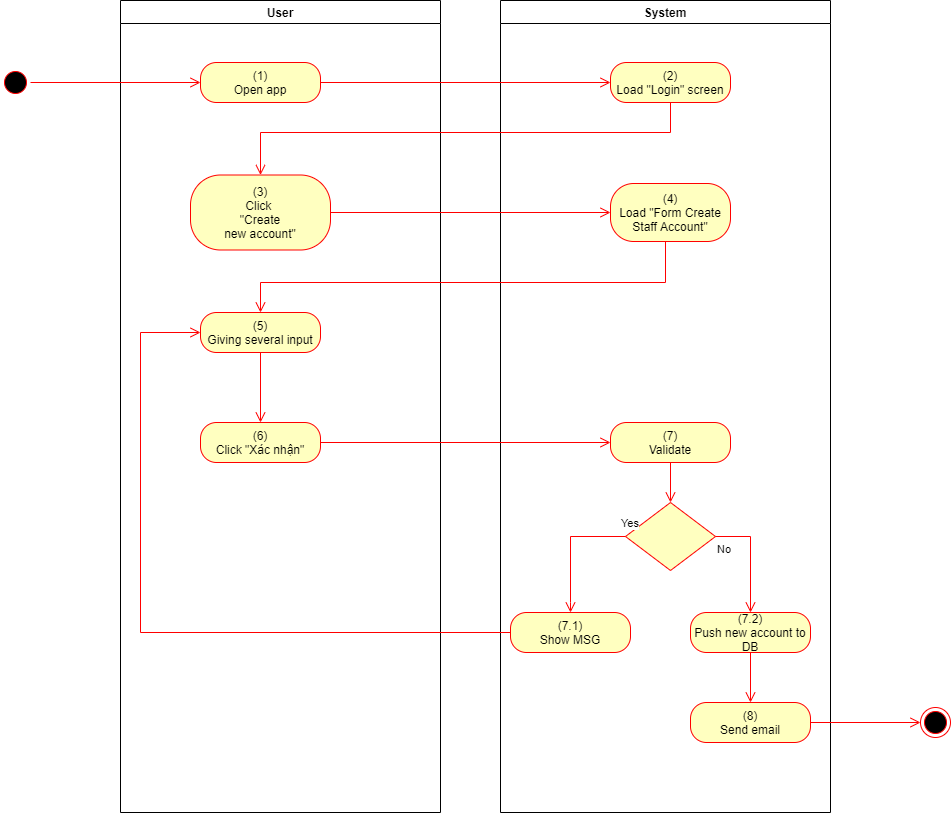
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình chính dành cho nhân viên (màn hình “Main Screen”). * Khi ấn vào left menu item “Đăng ký tài khoản mới (cho nhân viên)”[chỉ hiển thị cho tài khoản mark là “BGD” hoặc “Phòng Nhân sự”] Hệ thống tải form “Form Create new staff account” [trong sheet Screen 1 – Account Management]. |
| *(2)* | *BR2* | **Submitting Rules:**  Khi người dùng đã nhập tất cả thông tin vào các ô TextBox trong “Form create new staff account” khi click vào button “Xác nhận” thì hệ thống sẽ đưa thông tin account lên database (nếu vượt qua tất cả các kiểm tra định dạng thông tin). Còn nếu nhấn vào “Thoát” thì dù đã nhập hết đúng thông tin hay chưa nhập xong hay nhập có sai thì đều sẽ không được lưu mà thoát khỏi màn hình.   * If (Click “Xác nhận”) * If (AllValidate == true) => pushToDataBase(NewAccount), sendEmail(email) * Else => gửi thông báo MSG0x về lỗi sai như trong mô tả “Validate rule” * If (Click “Thoát”) * Form “Form Create new staff account” đóng.   **Email được gửi sẽ có định dạng sau:**   |  |  | | --- | --- | | From | SystemEmail | | To | [email] | | Cc | N/A | | Subject | Your PSS Account | | Body | Your account is: [email] Your password is: [password] | |
| *(3)* | *BR3* | **Validate Rules:**   * Kiểm tra định dạng Tên nhân viên: * If (NameOf == “” ) => MSG01 * If (NameOf.contains([number] || NameOf.contains([SpecialCharacter])) => MSG02 * Kiểm tra định dạng Số điện thoại: * If (PhoneNumberOf == “”) => MSG01 * If (PhoneNumberOf.contains([nonNumber]) => MSG03 * If (PhoneNumberOf.length > 10 || PhoneNumber < 10 || !PhoneNumber.startWith([validNumber])) => MSG4 * Kiểm tra định dạng CMND/CCCD * If (residentID == “”) => MSG01 * If (residentID.contains([nonNumber]) => MSG03 * If (residentID.length > 10 || PhoneNumber < 10) => MSG4 * Kiểm tra định dạng email * If (email == “”) => MSG01 * If (email.checkRightEmailFormat()) => MSG04 * If (email.isExistInCompany()) => MSG05 * Kiểm tra định dạng Password * If (Password == “”) => MSG01 * If (Password.contains(“ ”)) => MSG4 * Kiểm tra định dạng xác nhận Password * If (PasswordAgain != password) => MSG06 |

### UC2: Đăng ký (cho khách hàng)

|  |  |
| --- | --- |
| **Name** | **Đăng ký (cho khách hàng)** |
| **Description** | Khách hàng có thể tự tạo tài khoản riêng của mình đối với một cửa hàng |
| **Actor** | 1. User 2. None-user 3. System |
| **Trigger** | * Khi Khách hàng click vào button “Create new account” ở màn hình “Login” |
| **Pre-condition** | None |
| **Post-condition** | * Tài khoản mới được tạo và được gửi vào hộp thư email lúc khách hàng đăng ký. |

#### Activities Flow



*Figure 2: Sign up (customer) Activities Flow*

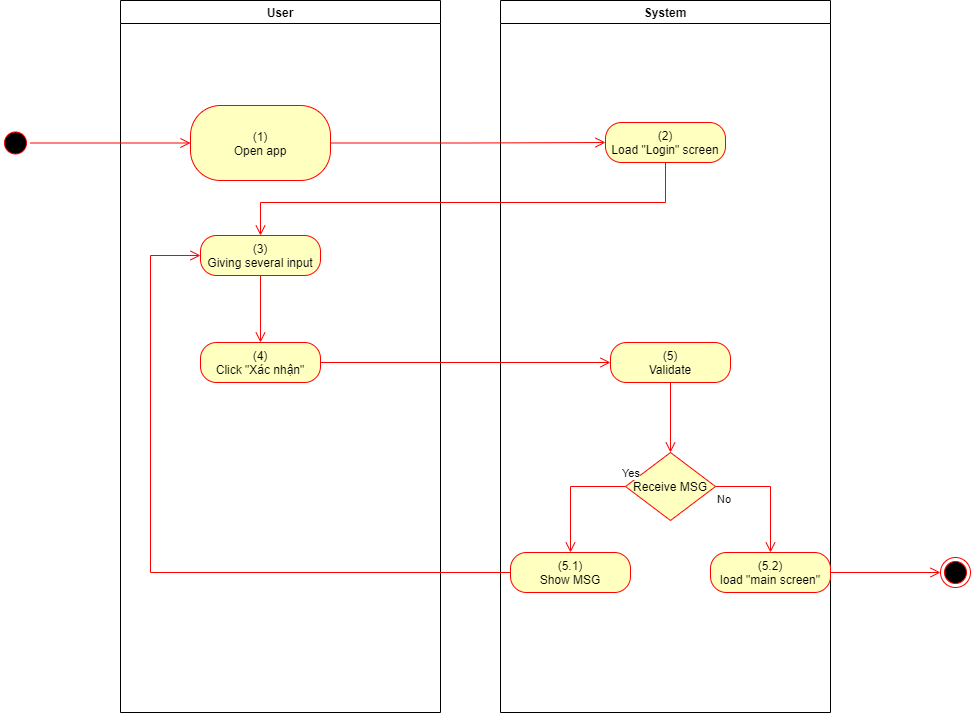
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Login”. * Người dùng click vào chọn “Sign up”. Xuất hiện màn hình “Sign up”. [Được mô tả trong List Description file, Screen 1 – Account Management] |
| *(2)* | *BR2* | **Submitting Rules:**  Khi người dùng đã nhập tất cả thông tin vào các ô TextBox trong “Create new account screen” khi click vào button “Xác nhận” thì hệ thống sẽ đưa thông tin account lên database (nếu vượt qua tất cả các kiểm tra định dạng thông tin). Còn nếu nhấn vào “Quay lại” thì dù đã nhập hết đúng thông tin hay chưa nhập xong hay nhập có sai thì đều sẽ không được lưu mà thoát khỏi màn hình.   * If (Click “Xác nhận”) * If (AllValidate == true) => pushToDataBase(NewAccount) * Else => gửi thông báo MSG0x về lỗi sai như trong mô tả “Validate rule” * If (Click “Quay lại”) * Quay lại màn hình “Login”.   **Email được gửi sẽ có định dạng sau:**   |  |  | | --- | --- | | From | SystemEmail | | To | [email] | | Cc | N/A | | Subject | Your PSS Account | | Body | Your account is: [email] Your password is: [password] | |
| *(3)* | *BR3* | **Validate Rules:**   * Kiểm tra định dạng Tên khách hàng: * If (NameOf == “” ) => MSG01 * If (NameOf.contains([number] || NameOf.contains([SpecialCharacter])) => MSG02 * Kiểm tra định dạng Số điện thoại: * If (PhoneNumberOf == “”) => MSG01 * If (PhoneNumberOf.contains([nonNumber]) => MSG03 * If (PhoneNumberOf.length > 10 || PhoneNumber < 10 || !PhoneNumber.startWith([validNumber])) => MSG4 * Kiểm tra định dạng CMND/CCCD * If (residentID == “”) => MSG01 * If (residentID.contains([nonNumber]) => MSG03 * If (residentID.length > 10 || PhoneNumber < 10) => MSG4 * Kiểm tra định dạng email * If (email == “”) => MSG01 * If (email.checkRightEmailFormat()) => MSG04 * If (email.isExist) => MSG05 * Kiểm tra định dạng Password * If (Password == “”) => MSG01 * If (Password.contains(“ ”)) => MSG4 * Kiểm tra định dạng xác nhận Password * If (PasswordAgain != password) => MSG06 |

### UC3: Đăng nhập

|  |  |
| --- | --- |
| **Name** | **Đăng nhập** |
| **Description** | Chức năng cho phép người dùng đăng nhập vào hệ thống |
| **Actor** | 1. User 2. Non-user 3. System |
| **Trigger** | * Khi người dùng mở ứng dụng |
| **Pre-condition** | None |
| **Post-condition** | * Vào màn hình “Main screen”. |

#### Activities Flow



*Figure 3: Login Activities Flow*

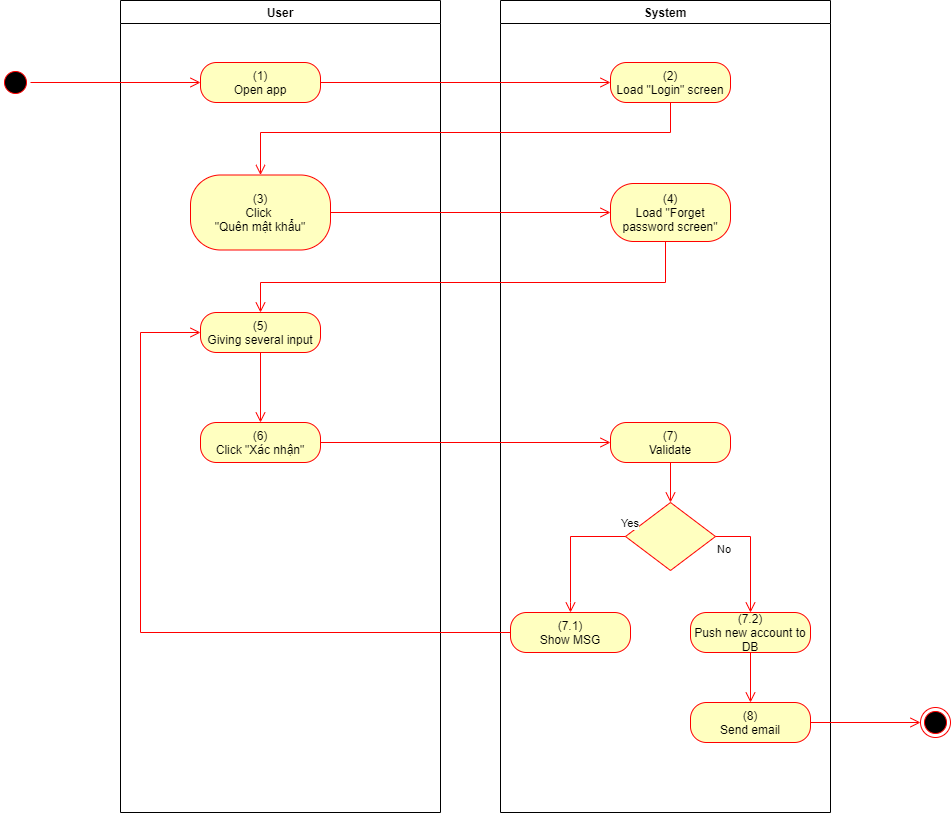
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Login”. |
| *(2)* | *BR2* | **Submitting Rules:**  Khi người dùng đã nhập tất cả thông tin vào các ô TextBox trong “Create new account screen” khi click vào button “Xác nhận” thì hệ thống sẽ đưa thông tin account lên database (nếu vượt qua tất cả các kiểm tra định dạng thông tin). Còn nếu nhấn vào “Quay lại” thì dù đã nhập hết đúng thông tin hay chưa nhập xong hay nhập có sai thì đều sẽ không được lưu mà thoát khỏi màn hình.   * If (Click “Đăng nhập”) * If (AllValidate == true) => mở màn hình “Main Screen” * Else => gửi thông báo MSG0x về lỗi sai như trong mô tả “Validate rule” * If (Click “Đăng ký”) * Mở màn hình “Sign Up”. * If (Click “Quên mật khẩu”) * Mở màn hình “Forgot Password” |
| *(3)* | *BR3* | **Validate Rules:**   * Kiểm tra định dạng email * If (email == “”) => MSG01 * If (email.checkRightEmailFormat()) => MSG04 * Kiểm tra định dạng Password * If (Password == “”) => MSG01 * If (Password.contains(“ ”)) => MSG4 * Kiểm tra khi để Authentication: * If (!email.isExistInComapany() || Password != db.Password) => MSG07 |

### UC4: Quên mật khẩu

|  |  |
| --- | --- |
| **Name** | **Quên mật khẩu** |
| **Description** | Chức năng cho phép người dùng đổi mật khẩu khi đã quên. |
| **Actor** | 1. User 2. Non-user 3. System |
| **Trigger** | * Khi người dùng ấn chọn nút “Quên mật khẩu” trên màn hình “Login” |
| **Pre-condition** | None |
| **Post-condition** | * Mật khẩu của email quên mật khẩu được thay đổi. |

#### Activities Flow



*Figure 4: Forget password Activities Flow*

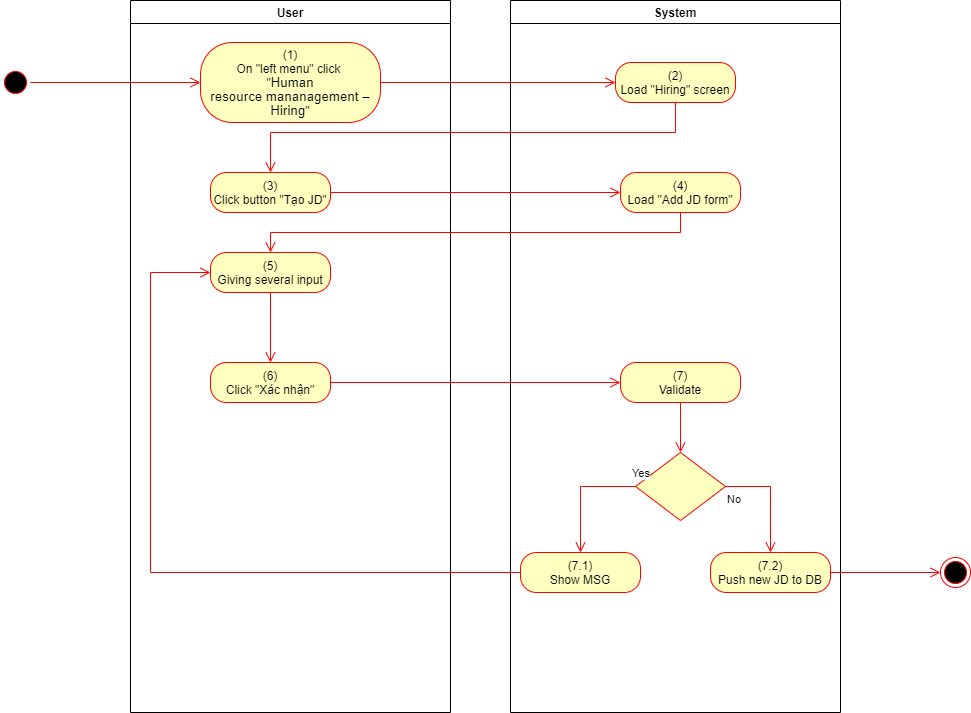
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Forgot Password” khi người dùng nhấn chọn button “Quên mật khẩu” ở màn hình “Login”. |
| *(2)* | *BR2* | **Submitting Rules:**  Người dùng nhập email quên mật khẩu vào ô Email để hệ thống gửi mã code xác thực vào đó. (Trong lúc này tất cả textbox, button khác (ngoại trừ button “gửi Code”) đều vô hiệu hóa)   * tbCode.isEnable = false * tbNewPassword.isEnable = false * btnConfirm.isEnable = false (button “Xác nhận”)   Khi người dùng đã ấn chọn button “gửi Code”, có thể điền và ấn tất cả phần còn lại (hủy vô hiệu hóa), đồng thời hệ thống gửi mã xác thực lên Email đã nhập:   * If (Click “gửi Code” && validate(email))) * tbCode.isEnable = true * tbNewPassword.isEnable = true * btnConfirm.isEnable = true * code = Random() [6 chữ số hóa thường bất kỳ] => StoreAndSend(code, email).   Người dùng tiến hành nhập code và mật khẩu mới và ấn chọn “Xác nhận” để đổi mật khẩu.   * If (Click “Xác Nhận”) * If (tbCode.text == code && validate(newPassword)) => Đổi mật khẩu thành công. * If (!validate(newPassword)) => nhận MSG0x * If (tbCode.text !=code) => nhận MSG08   **Email được gửi sẽ có định dạng sau:**   |  |  | | --- | --- | | From | SystemEmail | | To | [Email] | | Cc | N/A | | Subject | Verify code to changing password for PSS application | | Body | This email is send from PSS app, if you weren’t sending any request, please ignore this email.  Your change password code is: [code] | |
| *(3)* | *BR3* | **Validate Rules:**   * Kiểm tra định dạng email * If (email == “”) => MSG01 * If (email.checkRightEmailFormat()) => MSG04 * Kiểm tra định dạng Password * If (Password == “”) => MSG01 * If (Password.contains(“ ”)) => MSG4 |

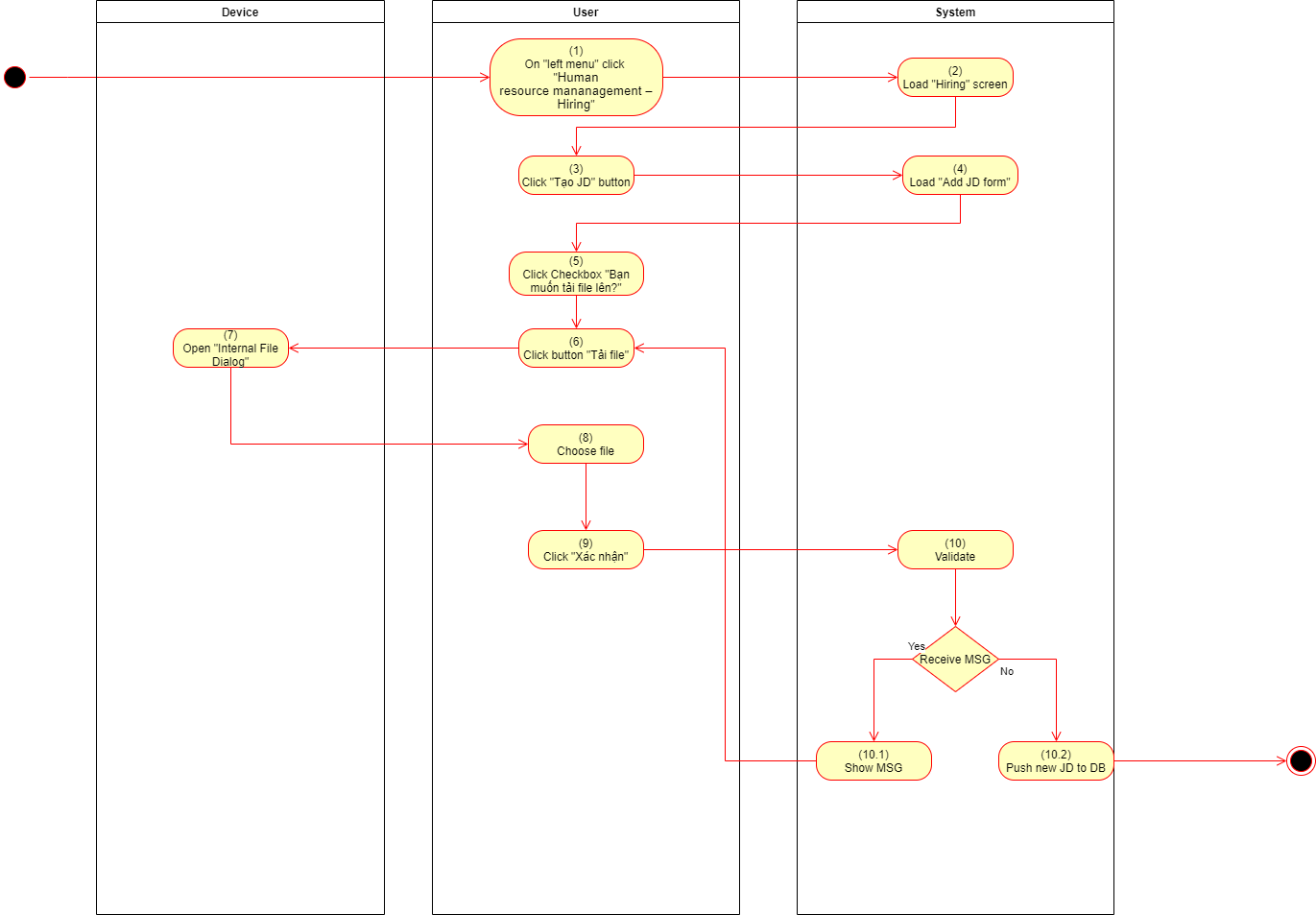
### UC5: Tuyển dụng

|  |  |
| --- | --- |
| **Name** | **Tuyển dụng** |
| **Description** | Chức năng cho phép tạo các thông báo tuyển dụng, hỗ trợ xuất file, in JD hỗ trợ việc tuyển dụng. |
| **Actor** | 1. User 2. System 3. Device |
| **Trigger** | * Khi nhân viên phòng nhân sự hoặc ban giám đốc ấn chọn vào “left menu” mục “Human resource management – Hiring” |
| **Pre-condition** | Đăng nhập bằng tài khoản của “BGD” hoặc phòng nhân sự. |
| **Post-condition** | * Bản JD theo mong muốn của người tạo. * Có thể có quảng cáo tuyển dụng trên app người mua. |

#### Activities Flow



*Figure 5: Hiring Activities Flow*

**

*Figure 6: Hiring (upload) Activities Flow*

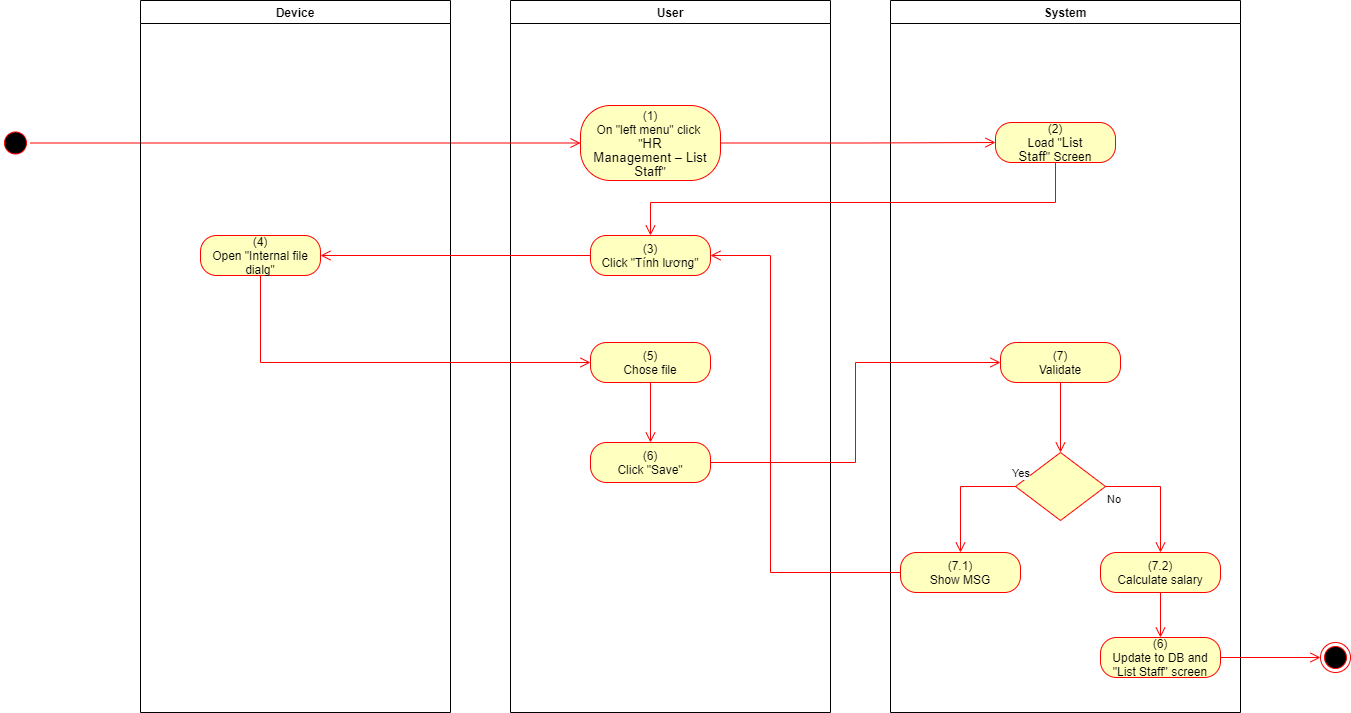
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Hiring” khi nhấn chọn vào “Left Menu” mục “Human resource mananagement – Hiring” . * Khi người dùng ấn chọn vào button “Tạo JD” trong màn hình “Hiring Screen”, “Add JD form” sẽ được tải lên. |
| *(2)* | *BR2* | **Adding JD rules:**  Người dùng tiến hành nhập tên JD.   * If (nameJD == “”) => MS01.   Người dùng chọn Checkbox (để isUpload == true) nếu muốn thực hiện tải lên file JD docs đã soạn sẵn (nếu không ấn chọn thì button “Tải lên” sẽ không thể ấn chọn => btnUpload.isEnable = false) .   1. **Thực hiện không tải lên JD có sẵn.**  * Không click chọn Checkbox (isUpload == false). * Tiến hành nhập theo **template JD** có sẵn, được chia làm 5 phần sau đây :  1. Tên công ty. => tbNameCompany.isEnable = true 2. Nơi làm việc => tbLocation.isEnable = true 3. Tên công việc. => tbPosition.isEnable = true 4. Yêu cầu công việc => tbJobRequirement.isEnable = true 5. Mô tả công việc. => tbJobDesscription.isEnable = true  |  | | --- | | Tên công ty: [companyName] | | Nơi làm việc: [Location] | | Tên công việc: [position] | | Yêu cầu công việc: [jobRequirement] | | Mô tả công việc: [jobDisCription] |  * If (nameJD != “” && companyName != “” && location != “” && position != “” && jobRequirement != “” && jobDescription != “”) && (startDateJD != null && endDateJD != Null && startDateJD < endDate && startDate > DateTime.Now()) * If (Click “Tạo JD mới”) => thực thi đưa lên database và cập nhật lại list: * pushDataToDB(new JD(ToDocs(companyName, location, position, jobDescription, jobRequirement), startDateJD, endDateJD)) * Đưa data lên database bằng cách tạo 1 JD, gồm 1 document (fileword) gồm các thông tin như trên **Template JD**. Ngày bắt đầu quảng cáo và ngày kết thúc quảng cáo. * Else: * If (companyName == “” || location == “” || position == “” || jobRequirement == “” || jobDescription == “” || startDateJD == NULL || endDateJD == NULL)) => nhận MSG01 * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDateJD < endDate) => Nhận thông báo MSG10.  1. **Thực hiện tải file JD (words) có sẵn từ máy**.  * Click chọn Checkbox (isUpload == true) => btnUpload.isEnable = true. (tbNameCompany.isEnable = false, tbLocation.isEnable = false, tbPosition.isEnable = false, tbJobRequirement.isEnable = false , tbJobDesscription.isEnable = false) * Click chọn “tải JD từ máy” => openInternalFileDialog() [window cung cấp].   => Chọn file và ấn chọn ok.  => Nếu không chọn ok mà cancel thì không file nào được lưu (if(Click “Cancel”) => Back())   * If (nameJD != “” && [docsFile] != null && startDateJD != null && endDateJD != Null && startDateJD < endDate && startDate > DateTime.Now()) => Click “Tạo JD mới” => thực thi đưa lên database và cập nhật lại list: * pushDataToDB(new JD([docsFile]), startDateJD, endDateJD)) * Đưa data lên database bằng cách tạo 1 JD, gồm 1 document (fileword) gồm các thông tin như trên **Template JD**. Ngày bắt đầu quảng cáo và ngày kết thúc quảng cáo. * Else: * If ( nameJD || startDateJD == NULL || endDateJD == NULL)) => nhận MSG01 * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDateJD < endDate) => Nhận thông báo MSG10.  1. **Hỗ trợ đăng lên quảng cáo:**  * Mỗi 1h sẽ cập nhật lại danh sách gồm các JD còn hiệu lực (DateTime.Now() < endDateJD) sẽ đưa lên quảng cáo (view của khách hàng)   Each ([timeStamp]){   * listAdvertise = [Empty array] * Check Foreach (JD jd in listJD) => if (jd.endatẹJD > DateTime.Now()) => listAdvertise.AddToAdvertise(jd) * pushToAdverties(listAdvertise)   } |

### UC6: Tính lương

|  |  |
| --- | --- |
| **Name** | **Tính lương** |
| **Description** | Chức năng cho phép người dùng đăng nhập vào hệ thống |
| **Actor** | 1. Hệ thống (tự động) 2. User |
| **Trigger** | * Khi người dùng ấn chọn tính lương. * Hoặc khi đến 00:00:00 ngày đấu tháng. |
| **Pre-condition** | * Đăng nhập với tư cách BGD hoặc phòng nhân sự. |
| **Post-condition** | * Một file excel lương được tính. |

#### Activities Flow



*Figure 7: Calculate Salary Activities Flow*

**

*Figure 8: Calculate Salary Activities Flow (Automatic)*

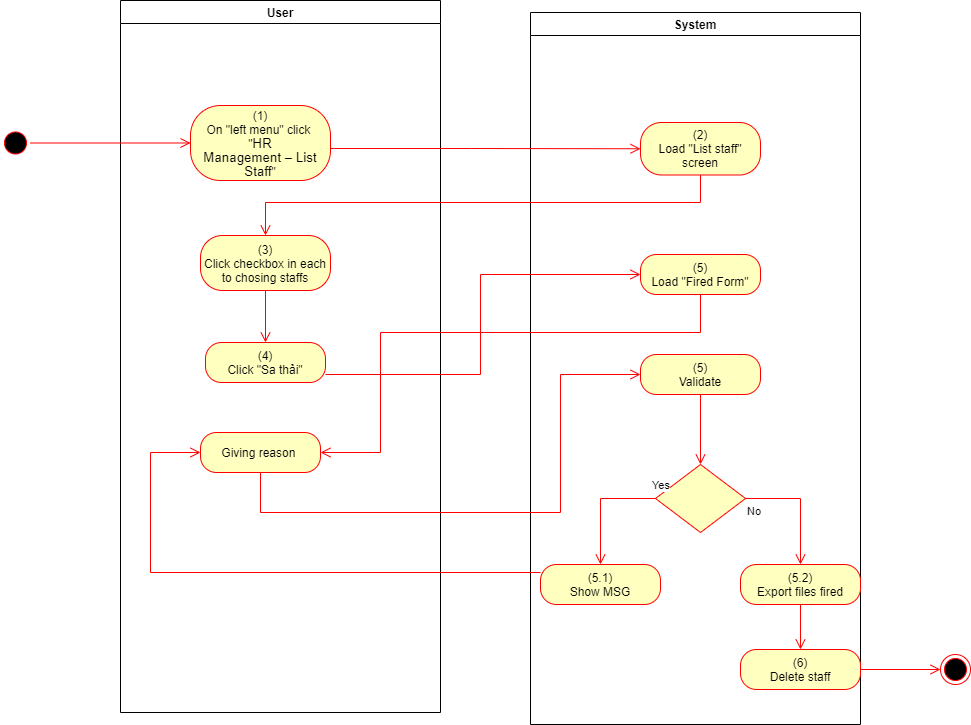
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Để hệ thống tải màn hình “List Staff Screen” chọn mục “HR Management – List Staff” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**   1. **Trường hợp hệ thống tự tính lương vào 23:59:59 ngày cuối tháng.**  * When(DateTime.Now() == DateTime(DateTimeFormater(“31/xx/2xxx 23:59:59”))) => Mặc định tất cả đi làm đầy đủ [absentDate == 0] * Foreach(Staff staff in listSatff) * staff.Salary = staff.Salary \* staff.position.fromEnumToInt() \* standardSalary[default = 0.75] * Chuyển từ position sang enum: * “BGD” = > 4 * “Nhân viên” => 1 * “Phó phòng” => 2 * “Trưởng phòng” => 3  1. **Trường hợp ấn chọn button “Tính lương”.**  * If(Click “Tính lương”) => openInternalFileDialog() [window cung cấp].   => Chọn file và ấn chọn “Save”.  => Nếu không chọn ok mà cancel thì không file nào được lưu (if(Click “Cancel”) => Back())   * Sau khi ấn chọn “Save” trong Dialog hệ thống sẽ tự động tính lương bằng cách scan **File excel tính lương**. * If(!Validate(excelFile)) => MSG11 * If(Validate(excelFile) && !fitCurrentSystemListStaff(excelFile)) => MSG12 * If (Validate(excelFile) && fitCurrentSystemListStaff(excelFile)) => Tiên hành tính lương: * Foreach(Staff staff in listSatff) => staff.Salary = staff.Salary \* staff.position.fromEnumToInt() \* standardSalary[default = 0.75] – ([absentDate]/30) \* (standardSalary)  1. Định dạng **File excel tính lương:**   Luật tính vắng: (ngày làm 8 tiếng)  If ((EndTime.hour – StartTime.hour) < 8) => [absent]= [absent] +1   |  |  |  |  |  | | --- | --- | --- | --- | --- | | STT | Id nhân viên | Yyyy-mm-dd | | … | | StartTime | EndTime |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

### UC7: Sa thải

|  |  |
| --- | --- |
| **Name** | **Sa thải** |
| **Description** | Hỗ trợ chọn lựa sa thải nhân viên, hỗ trợ sa thải, tạo biên bản sa thải |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng chọn các người cần sa thải và ấn chọn button “Sa thải”. |
| **Pre-condition** | * Đăng nhập với tư cách BGD hoặc phòng nhân sự. |
| **Post-condition** | * Một biên bản sa thải. |

#### Activities Flow



*Figure 9: Fired Activities Flow*

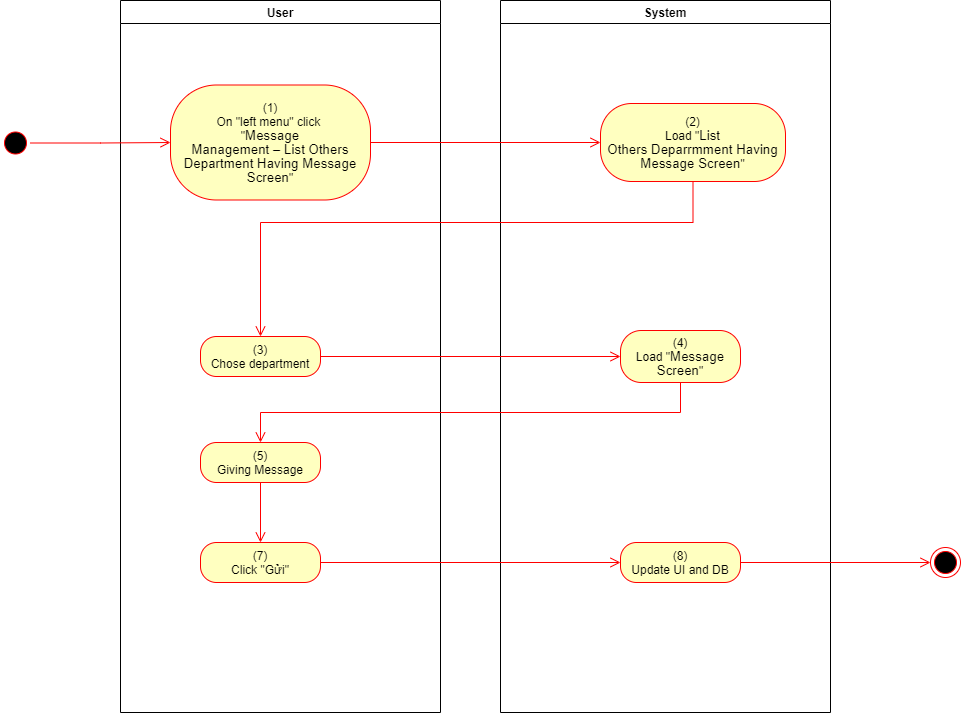
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Để hệ thống tải màn hình “List Staff Screen” chọn mục “HR Management – List Staff” ở “left menu”. * Khi ấn chọn nhiều hoặc một nhân viên rồi ấn “Sa thải” sẽ hiện ra “Fired Form”. |
| *(2)* | *BR2* | **Business Rules:**   * Người dùng ấn chọn vào checkbox trong listStaff để chọn các nhân viên sẽ bị sa thải * Click(Checkbox) isChose=!isChose. * Nếu người dùng chọn nhiều hay chọn một nhân viên thì chỉ có duy nhất một “Fired Form” hiển thị theo đó: * staffId = “\*” * staffName = “\*” * makeDecisionName = currentUser.getName() (lấy từ tài khoản người đang đăng nhập) * dateMakeDecision = DateTime.Now() * Tiến hành nhập lý do sa thải: * If (reason = “”) => nhận MSG01 khi ấn chọn “Xác nhận” * Ấn chọn button “Xác nhận” để in/nhận biên bản sa thải (lúc này nhân viên sẽ không còn trong list). * Foreach(Staff staff in listStaff) => if (staff.isChose) => listStaff.Remove(staff) (Xóa tất cả staff đã chọn) => ExportFileFired(staff.Id, staff.name, makeDecisionName, dateMakeDecision, reason). * Điều này có nghĩa tất cả các nhân viên chọn trong cũng 1 lần sẽ nhận chung 1 lý do sa thải. * Nếu Ấn chọn button “Thoát” thì sẽ không có gì xảy ra, và quay lại màn hình “List Staff” với vẫn còn các Staff đã chọn trước đó nhưng chưa thực hiện sa thải. * If(Click “Thoát”) backToListStaff()   **Định dạng biên bản sa thải:**   |  | | --- | | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM | | Độc lập – Tự do -Hạnh phúc | | **BIÊN BẢN SA THẢI** | | **Quyết định sa thải nhân viên mã id: [staff.id] Họ và tên [staff.Name]** | | **Chiếu theo quyết định của: [makeDecisionName]** | |  | | **Lý do sa thải: [reason]** | | **Quyết định thực hiện từ ngày: [dateMakeDecision]** | | **Ký tên** | |

### UC8: Nhắn tin giữa các phòng ban

|  |  |
| --- | --- |
| **Name** | **Nhắn tin giữa các phòng ban** |
| **Description** | Hỗ trợ thực hiện trao đổi tin nhắn giữa các phòng ban trong công ty |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở vào mục “Message Management - List Others Department Having Message Screen” ở “left menu” |
| **Pre-condition** | * Là tài khoản của trưởng phòng hoặc phó phòng. |
| **Post-condition** | * Nhắn tin qua lại giữa các phòng ban |

#### Activities Flow



*Figure 10: Department’s message Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Ở “left menu” người dùng ấn chọn “Message Management – List Others Department Having Message Screen” để tiến hành tải lên “List Others Department Having Message Screen”. Sau đó chọn phòng ban để mở “Message Screen” với phòng ban đó. |
| *(2)* | *BR2* | **Business Rules:**  Khi người dùng đã ở “List Other Department Having Message Screen” sẽ hiển thị list các departme kèm tin nhắn mới nhất:   * listDepartmentAndMess = makingListOfDepartmentAndMess(GetFromDb(“Select \* from Department), GetFromDb(Select tie[numberOfDepartment] from Message Where [Each Department newest message]) * Có thể thực hiện lọc danh sách gồm các phòng ban có tình trạng tin nhắn theo: * Chưa đọc => newestMessage.Status = “UNREAD” * Đã đọc => newestMessage.Sattus= “READ” * Tất cả [default] [“UNREAD” or “READ”] * If (department.listMessage.ContainsStatus([filterStatus])) => displayList.add (department) (chỉ thêm phòng ban vào danh sách hiển thị khi có status như yêu cầu)   newestMessage = GetFromDb(Select \* from Message Where [Each Department newest message])  Khi người dùng ấn chọn vào một Item của list, màn hình “Message Screen” đối với Item đó sẽ hiển thị, bào gồm list tin nhắn, tên, avatar của phòng và nếu có tin nhắn chưa đọc sẽ được chuyển thành đã đọc:   * nameOfPerson = departmentAndMess.getName() [từ selected listDepartmentAndMess item] * imageOfPerson = departmentAndMess.getImage() * listMessage = listFromDB(“”Select \* from Message where senderID = [departmentAndMess.ID]” ||receiverID=[departmentAndMess.ID]) * if(havingUnreadMessage) => foreach (Message message in listMessage) => message.status = “READ”   Khi người dùng nhập vào ô tbMessage (“ô nhập tin nhắn”) và ấn chọn button “Gửi” thì hệ thống sẽ kiểm tra và gửi tin nhắn:   * If (tbMesage.text = “”) => MSG01 * If (tbMessage.text != “”) => message = tbMessage.text => Send(message, departmentAndMess.getID(), DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”]   Khi người dùng ấn chọn button “Chọn hình ảnh”/”Chọn video” hệ thống sẽ mở nơi lưu trữa hình ảnh trong máy để tiến hành chọn:   * If (Click “Chọn hình ảnh”) => openInternalSavingImageDialog() * if(Click “Save” && image == NULL ) => MSG13 * If (Click”Save” && image != NULL) => Send(image, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FORMAT => “yyyy-mm-dd hh:mm:ss”] * If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen” * If (Click “Chọn video”) => openInternalSavingVideoDialog() * if(Click “Save” && image == NULL ) => MSG14 * If (Click”Save” && video != NULL) => Send(video, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FORMAT => “yyyy-mm-dd hh:mm:ss”] * If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen” |

### UC9: Nhắn tin giữa khách hàng và người bán

|  |  |
| --- | --- |
| **Name** | **Nhắn tin giữa khách hàng và người bán** |
| **Description** | Hỗ trợ trao đổi tin nhắn giữa khách hàng và nơi bán (phòng ban hoặc đơn vị được chỉ định xử lý các nghi vấn, kiến nghị, yêu cầu từ khách hàng) |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở vào mục “Message Management - List Customer Having Message Screen” ở “left menu” * Nếu là khách hàng: Chọn mục “Message” ở “left menu” |
| **Pre-condition** | * Là tài khoản của BGD hoặc Phòng kinh doanh. * Là tài khoản khách hàng ([Customer perspective]) |
| **Post-condition** | * Nhắn tin qua lại giữa khách hàng và bên bán |

#### Activities Flow



*Figure 11: Seller perspective message with customer Activities Flow*

**

*Figure 12: Customer perspective message with seller Activities Flow*

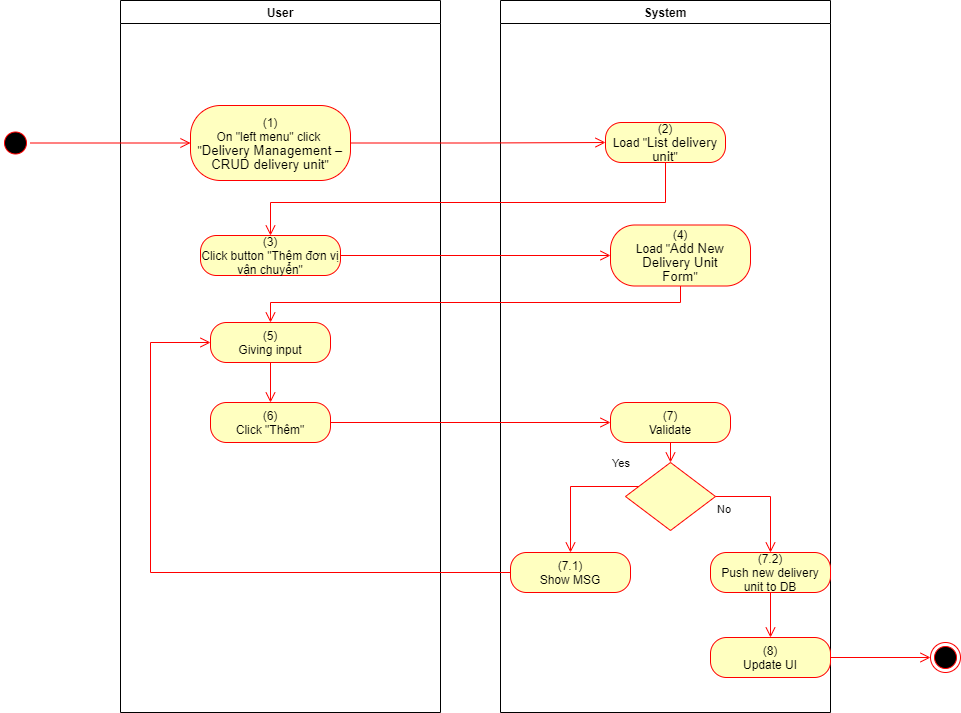
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Ở “left menu” người dùng ấn chọn “Message Management – List Customer Having Message Screen” để tiến hành tải lên “List Others Customer Having Message Screen”. Sau đó chọn khách hàng để mở “Message Screen” với người đó. * Nếu là khách hàng: Người dùng ấn chọn “Message” ở “left menu” để tiến hành mở trực tiếp “Message Screen”. |
| *(2)* | *BR2* | **Business Rules:**   1. **Góc nhìn từ người bán:**   Khi người dùng đã ở “List Other Customer Having Message Screen” sẽ hiển thị list các customer kèm tin nhắn mới nhất:   * listCustomerAndMess = makingListOfCustomerAndMess(GetFromDb(“Select \* from Customer where [Message.count != 0]), GetFromDb(Select tie[numberOfCustomer] from Message Where [Each Department newest message]) * Có thể thực hiện lọc danh sách gồm các khách hàng có tình trạng tin nhắn theo: * Chưa đọc => newestMessage.Status = “UNREAD” * Đã đọc => newestMessage.Sattus= “READ” * Tất cả [default] [“UNREAD” or “READ”] * If (customer.listMessage.ContainsStatus([filterStatus])) => displayList.add (customer) (chỉ thêm khách hàng vào danh sách hiển thị khi có status như yêu cầu)   newestMessage = GetFromDb(Select \* from Message Where [Each Department newest message])  Khi người dùng ấn chọn vào một Item của list, màn hình “Message Screen” đối với Item đó sẽ hiển thị, bào gồm list tin nhắn, tên, avatar của phòng và nếu có tin nhắn chưa đọc sẽ được chuyển thành đã đọc:   * nameOfPerson = customerAndMess.getName() [từ selected listDepartmentAndMess item] * imageOfPerson = customerAndMess.getImage() * listMessage = listFromDB(“”Select \* from Message where senderID = [customerAndMess.ID]” || receiverID = [customerAndMess.ID]) * if(havingUnreadMessage) => foreach (Message message in listMessage) => message.status = “READ”   Khi người dùng nhập vào ô tbMessage (“ô nhập tin nhắn”) và ấn chọn button “Gửi” thì hệ thống sẽ kiểm tra và gửi tin nhắn:   * If (tbMesage.text = “”) => MSG01 * If (tbMessage.text != “”) => message = tbMessage.text => Send(message, departmentAndMess.getID(), DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”]   Khi người dùng ấn chọn button “Chọn hình ảnh”/”Chọn video” hệ thống sẽ mở nơi lưu trữ hình ảnh trong máy để tiến hành chọn:   * If (Click “Chọn hình ảnh”) => openInternalSavingImageDialog() * if(Click “Save” && image == NULL ) => MSG13 * If (Click”Save” && image != NULL) => Send(image, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”] * If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen” * If (Click “Chọn video”) => openInternalSavingVideoDialog() * if(Click “Save” && image == NULL ) => MSG14 * If (Click”Save” && video != NULL) => Send(video, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FORMAT => “yyyy-mm-dd hh:mm:ss”]   If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen”   1. **Góc nhìn từ khách hàng:**   Khi người dùng ấn chọn vào mục “Message” bên “left menu” thì sẽ tiến hành mở màn hình “Message Screen” để đối thoại với người bán [nhiều tài khoản người bán sẽ chỉ giao tiếp với 1 người mua thông qua 1 “Message screen”]   * nameOfPerson = “PSS Customer Service” * imageOfPerson = [PSS LOGO] * listMessage = listFromDB(“”Select \* from Message where senderID = [customerAndMess.ID]” || receiverID = [customerAndMess.ID]) * if(havingUnreadMessage) => foreach (Message message in listMessage) => message.status = “READ”   Khi người dùng nhập vào ô tbMessage (“ô nhập tin nhắn”) và ấn chọn button “Gửi” thì hệ thống sẽ kiểm tra và gửi tin nhắn:   * If (tbMesage.text = “”) => MSG01 * If (tbMessage.text != “”) => message = tbMessage.text => Send(message, departmentAndMess.getID(), DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”]   Khi người dùng ấn chọn button “Chọn hình ảnh”/”Chọn video” hệ thống sẽ mở nơi lưu trữ hình ảnh trong máy để tiến hành chọn:   * If (Click “Chọn hình ảnh”) => openInternalSavingImageDialog() * if(Click “Save” && image == NULL ) => MSG13 * If (Click”Save” && image != NULL) => Send(image, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”] * If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen” * If (Click “Chọn video”) => openInternalSavingVideoDialog() * if(Click “Save” && image == NULL ) => MSG14 * If (Click”Save” && video != NULL) => Send(video, departmentAndMess.getID(),DateTime.Now().SHORT\_FORMAT).[SHORT\_FỎMAT => “yyyy-mm-dd hh:mm:ss”]   If (Click “Cancel”) => Back() => tức không có gì được gửi hay thực hiện mà chỉ back lại màn hình “Message Screen” |

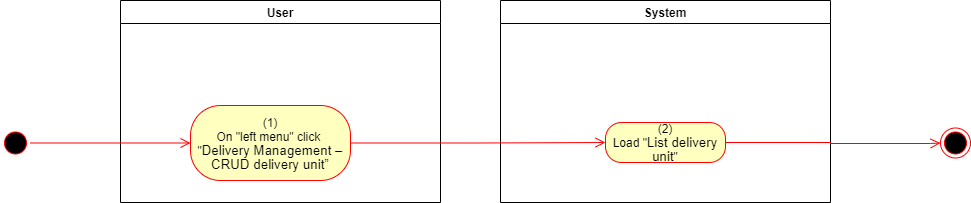
### UC10: CRUD đơn vị vận chuyển

|  |  |
| --- | --- |
| **Name** | **CRUD đơn vị vận chuyển** |
| **Description** | Chức năng cho phép người dùng đăng nhập vào hệ thống |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dừng ấn chọn mục “Delivery Management – CRUD delivery unit” trong “left menu” |
| **Pre-condition** | Đăng nhập với tài khoản của BGD |
| **Post-condition** | * Thực thi CRUD được đơn vị vận chuyển theo thao tác người dùng. |

#### Activities Flow



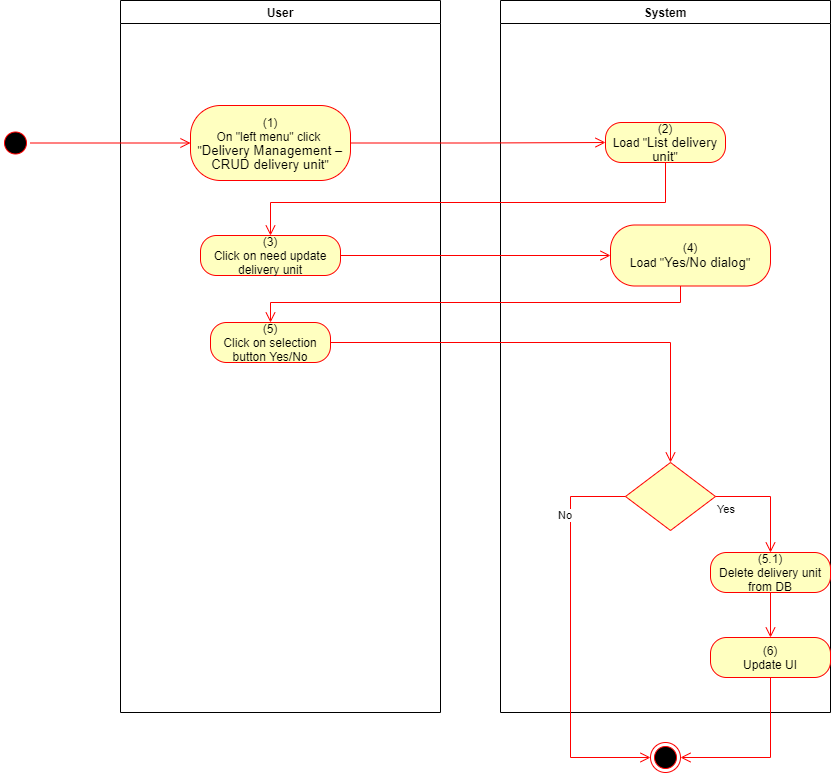
*Figure 13: Create Delivery Unit Activities Flow*

**

*Figure 14: Read Delivery Unit Activities Flow*

**

*Figure 15: Update Delivery Unit Activities Flow*

**

*Figure 16: Delete Delivery Unit Activities Flow*

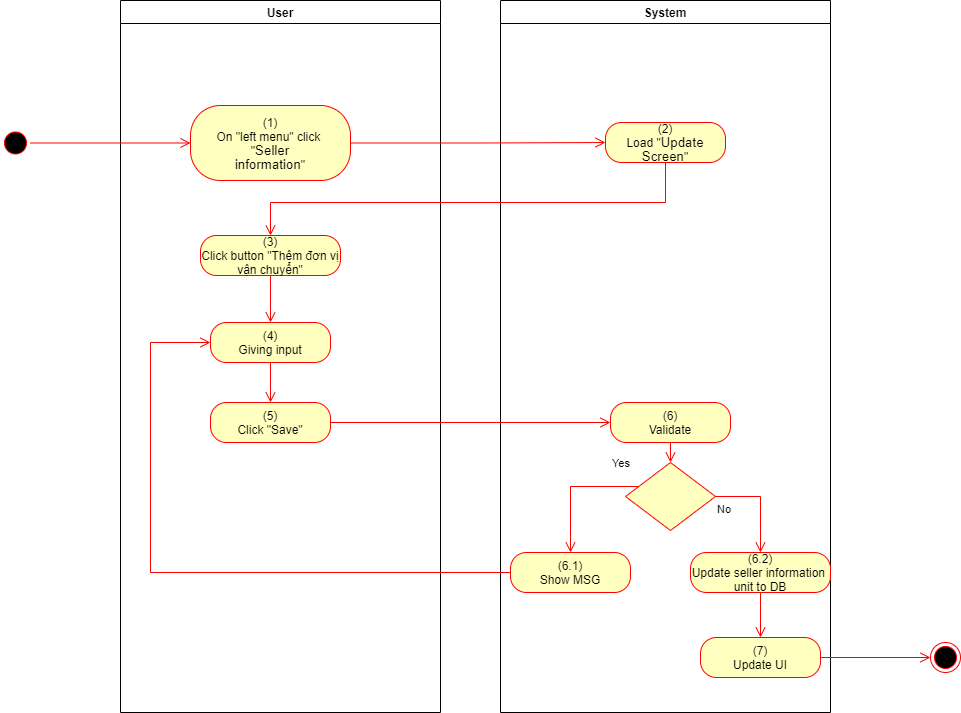
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Khi người dùng ấn chọn vào mục “Delivery Management – CRUD delivery unit” màn hình “List delivery unit” sẽ được tải lên. |
| *(2)* | *BR2* | **Business Rules:**   1. **Read:**   Khi người dùng ấn chọn vào mục “Delivery Management – CRUD delivery unit” màn hình sẽ hiển thị màn hình “List delivery unit” bao gồm nhiều deliveryItem.  deliveryItem sẽ chứa toàn bộ thông tin về một đơn vị vận chuyển, nhưng màn hình “List delivery unit” sẽ hiển thị nhiều deliveryItem với các thông tin cơ bản như trong mô tả ở **View Description** :   * Khi thực hiện tải màn hình “List delivery unit” sẽ thực hiện tải list từ database về: * listDeliveryUnit = loadListFromDb("Select \* from DeliveryUnit”).  1. **Create:**   Để thực hiện việt tạo thêm đơn vị vận chuyển, thực hiện ấn vào button “Thêm đơn vị” để hệ thống thực hiện mở “Add New Delivery Unit Form”. Tiến hành nhập các thông tin mà form yêu cầu, trong đó, các API sẽ được nhập theo một trong 2 định dạng sau:   |  |  |  |  | | --- | --- | --- | --- | | 1 | https:// | [Domain name].[top level domain] | /[endParh] | | 2 | Localhost:: | [portNumber] | /[endPath] | | Note: [endPath] nếu cần truyền pathvariable thì thực hiện như lúc khai báo code API => /{variableName} | | | |  * Việc kiểm tra format sẽ được hàm checkFormatAPI(String api) thực hiện. (tự xây dựng)   Khi nhấn chọn button “Thêm” thì hệ thống sẽ thực hiện kiểm tra định dạng các thông tin và gửi các MSG tương ứng hoặc đưa đơn vị vận chuyển mới lên Database:   * If (Click “Thêm”) => ValidateInput() * ValidateInput: * If (nameDelivery == “” || account == “” || password = “” || orderInfoAPI == “” || priceAPI == “”|| authenAPI == “”) => nhận MSG01. * If (account.contains(“ ”)|| password.contains(“ ”)) => nhận MSG04. * If(checkFormatAPI(orderInfoAPI) || checkFormatAPI(priceAPI) || checkFormatAPI(authenAPI)) =>Nhận MSG04 * If (ValidateInput()) => SendToDB(new DeliveryItem(nameDelivery, account, password, orderInfoAPI, priceAPI, auhtenAPi, updateDate=DateTime.Now())). * Đóng gói dữ liệu và gửi dữ liệu lên database.   Nếu người dúng ấn chọn button “Thoát” thì dù thông tin có sai hay không thì hệ thộng sẽ không lưu lại mà quay lại màn hình “List Delivery Unit”:   * If(Click “Thoát”) => Back().  1. **Update:**   Để thực hiện việc thay đổi, cập nhật một đơn vị vận chuyển, thực hiện ấn vào button “Chỉnh sửa” trên mỗi item của listDeliveryUnit trong màn hình “List Delivery Unit” để thực hiện chỉnh sửa cho item đó. Khi ấn chọn button “Chỉnh sửa” hệ thống sẽ hiển thị “Update Delivery Unit Form”. Người dùng tiến hành thực hiện các chỉnh sửa (Form có chứa sẵn các thông tin của item đã chọn).   * nameDelivery = deliveryItem.nameDelivery * account = deliveryItem.account * pasword = deliveryItem.pasword * orderInfoAPI = deliveryItem.orderInfoAPI * priceAPI = deliveryItem.priceAPI * authenAPI = deliveryItem.authenAPI * Thực hiện gán để các thông tin hiển thị trên Form.   Click chọn button “Lưu” để kiểm tra định dạng và tải dữ liệu mới lên database.   * If (Click “Lưu”) => ValidateInput() * ValidateInput: * If (nameDelivery == “” || account == “” || password = “” || orderInfoAPI == “” || priceAPI == “”|| authenAPI == “”) => nhận MSG01. * If (account.contains(“ ”)|| password.contains(“ ”)) => nhận MSG04. * If(checkFormatAPI(orderInfoAPI) || checkFormatAPI(priceAPI) || checkFormatAPI(authenAPI)) =>Nhận MSG04 * If (ValidateInput()) => updateToDB(deliveryItem.ID ,nameDelivery, account, password, orderInfoAPI, priceAPI, auhtenAPi, updateDate=DateTime.Now)). * Đóng gói dữ liệu và gửi dữ liệu lên database.   Nếu người dúng ấn chọn button “Thoát” thì dù thông tin có sai hay không thì hệ thộng sẽ không lưu lại mà quay lại màn hình “List Delivery Unit”:   * If(Click “Thoát”) => Back().  1. **Delete:**   Để thực hiện việc xóa một đơn vị vận chuyển, thực hiện ấn vào button “Xóa” trên mỗi item của listDeliveryUnit trong màn hình “List Delivery Unit” để thực hiện xóa item đó. Khi ấn chọn item “Xóa” hệ thống sẽ mở hộp thoại Yes/No để thực hiện chọn lựa có chắc chắn xóa hay không.   * If (Click “Yes”) => listDeliveryUnit.deleteInListAndDB(deliveryItem.ID) => updateUI(). * Khi click button “Yes” sẽ thực hiện xóa item ra khỏi list và database sau đó cập nhật lại giao diện. * If (Click “No”) => Back() * Nếu người dùng ấn chọn button “No” thì sẽ thực hiện tắt Yes/No dialog và quay lại màn hình “List Delivery Unit”. |

### UC11: CRUD Thông tin người bán

|  |  |
| --- | --- |
| **Name** | **CRUD Thông tin người bán** |
| **Description** | Hỗ trợ thực hiện CRUD đối với thông tin người bán. |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng chọn mục “Seller information” trong “left menu” |
| **Pre-condition** | Đăng nhập với tư cách BGD hoặc thuộc “phòng kinh doanh” |
| **Post-condition** | * CRUD thông tin người bán như thao tác người dùng |

#### Activities Flow



*Figure 17: CRUD Seller information Activities Flow*

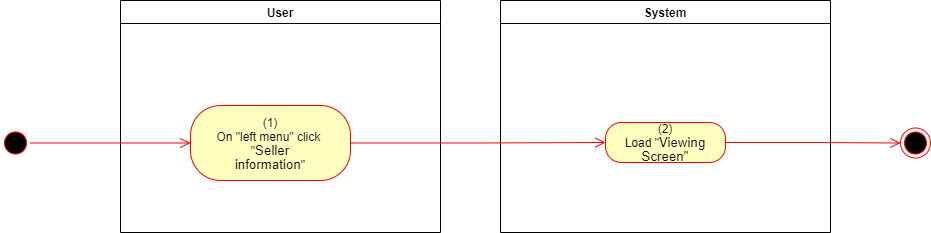
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Khi người dùng chọn mục “Seller information” trong “left menu” hệ thống sẽ hiển thị màn hình “Update Screen”. |
| *(2)* | *BR2* | **Business Rules:**  Đối với việc tạo lập thông tin người bán việc xem và thay đổi dành cho bên bán sẽ thực hiện chung một màn hình “Update Screen”, đối với lần đầu vào ứng dụng (hoặc khi chưa có thông tin), màn hình sẽ hiển thị tất cả các trường đều trống.   * When (sellerInfomation == NULL) * nameCompany = sellerInfomation.nameCompany = “” * listAddress = sellerInfomation.listAddress = NULL * updateDate = sellerInfomation.updateDate = NULL * mainImage = sellerInfomation.mainImage = NULL * video = sellerInfomation.video = NULL * listProductImage =sellerInfomation.listProductImage = NULL * Else (sellerInformation !=NULL) * nameCompany = sellerInfomation.nameCompany * listAddress = sellerInfomation.listAddress * listAddress = sellerInfomation.listAddress * updateDate = sellerInfomation.updateDate * mainImage = sellerInfomation.mainImage * video = sellerInfomation.video * listProductImage =sellerInfomation.listProductImage   Việc nhập tất cả các thành phần là không bắt buộc để lưu là không bắt buộc.   * If (Click “Lưu”) => If (nameCompany = “” ||listAddress == NULL ||updateDate = NULL ||mainImage == NULL ||video == NULL || listProductImage == NULL) => không có thông báo.   Khi cần tạo hoặc thay đổi ảnh nền ta thực hiện ấn chọn vào “Chọn ảnh nền” (Nơi hiển thị ảnh nền trong màn hình “Update Screen”) sau đó chọn ảnh từ Dialog hiện lên và ấn chọn Save để tiến hành lưu hoặc chọn “Back” để thoát và không có gì thay đổi.   * If (Click “mainImage”) => openInternalImageSavingDialog() * If (Click “Save”) => sellerInformation.setMainImageAndPushToDB([selectedImage]) * Lưu hình ảnh đã chọn (selectedImage) lên database và đưa ảnh nền đã chọn lên màn hình “Update Screen”. * If (Click “Back”) = > Back() * Quay về màn hình “Update Screen”   Khi cần thêm hoặc thay đổi ảnh sản phẩm nổi bật ta thực hiện ấn chọn vào “Danh sách sản phẩm đặc trung” (Nơi hiển thị ảnh sản phẩm đặc trưng trong màn hình “Update Screen”) sau đó chọn các ảnh từ Dialog hiện lên và ấn chọn Save để tiến hành lưu hoặc chọn “Back” để thoát và không có gì thay đổi.   * If (Click “listProductImage”) => openInternalImageSavingDialog() * If (Click “Save”) => sellerInformation.addListProductImageAndPushToDB([selectedImageList]) * Lưu hình các ảnh vừa chọn (selectedImageList) lên database và đưa các ảnh đã chọn lên màn hình “Update Screen”. * If (Click “Back”) = > Back() * Quay về màn hình “Update Screen”   Khi cần tạo hoặc thay đổi video ta thực hiện ấn chọn vào “Chọn video” (Nơi hiển thị video trong màn hình “Update Screen”) sau đó chọn video từ Dialog hiện lên và ấn chọn Save để tiến hành lưu hoặc chọn “Back” để thoát và không có gì thay đổi.   * If (Click “video”) => openInternalVideoSavingDialog() * If (Click “Save”) => sellerInformation.setVideoAndPushToDB([selectedVideo]) * Lưu video đã chọn (selectedVideo) lên database và đưa video đã chọn lên màn hình “Update Screen”. * If (Click “Back”) = > Back() * Quay về màn hình “Update Screen”   Khi cần tạo hoặc thêm hoặc xóa địa chỉ, ở đây là một list các địa chỉ (dành cho các cửa hàng nhiều chi nhánh), thực hiện nhập/chọn cho các thông tin ở các ô “Tỉnh”, “Huyện”, “Xã”, “Đường”. Khi tất cả đã nhập/chọn đủ thì hệ thống tự update UI thêm 4 trường trên để tiếp tục nhập (Khi xóa 4 trường đều trống và hiện có 1 hoặc nhiều địa chỉ đầy đủ ở trên thì hệ thống sẽ tự động xóa đi 4 mục này).   * listDistrict = GetAllDistrictFromDB("Select \* from District”) * dbDistrict.isEnable = false (Drop box district) * dbWard.isEnable = false (Drop box ward) * tbLane.isEnable = false (text box Lane) * Ban đầu sẽ tự động tải các tỉnh, các DropBox của “Huyện” và “Xã” và textbox “Đường” sẽ bị “vô hiệu hóa” * If(Chose “Tỉnh”) => dbDistrict.isEnable = true => getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “huyện” và tải tất cả các huyện theo ID của tỉnh đã chọn. * If(Chose “Huyện”) => db.Ward.isEnable = true => getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “Xã” và tải tất cả các huyện theo ID của tỉnh đã chọn.   Khi tất cả đã nhập/chọn đủ thì hệ thống tự update UI thêm 4 trường trên để tiếp tục nhập (Khi xóa 4 trường đều trống và hiện có 1 hoặc nhiều địa chỉ đầy đủ ở trên thì hệ thống sẽ tự động xóa đi 4 mục này).   * If AllUIForInputHaving(dbDistrict.selectedItem != NULL && dbProvince.selectedItem != NULL && dbWard.selectedItem != NULL && tbLane.text != “”) addUIUnitForInput() * Khi tất cả ô đã được nhập chọn, hệ thống thực hiện thêm 1 Cụm UI nhập địa chỉ * If (UIForInputHaving(dbDistrict.selectedItem == NULL && dbProvince.selectedItem == NULL && dbWard.selectedItem == NULL && tbLane.text == “”) && HavingUIUnitFullFill() && HavingUIforInput) => deleteUIForInput() * Khi có 1 cum UI nhập địa chỉ trống (không có thông tin nào được chọn hoặc nhập) và có 1 hoặc nhiều địa chỉ đã được nhập và có UI để nhập rồi thì thực hiện xóa cụm UI rỗng đó.   Khi thay đổi chọn một “tỉnh”, “huyện”, “xã” bất kỳ, thì các cấp bậc dưới sẽ được “vô hiệu hóa” và xóa thông tin tương ứng, thay đổi lại danh sách lựa chọn tương ứng.   * If (change “Tỉnh”) * dbDistrict.isEnable = true, dbDistrict.SelectedItem = NULL,dbDistrict.ListDistrict=getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * dbWard.isEnable = false, dbWard.selectedItem = NULL, dbWard.listWard = [Empty list]. * tbLane.text = “”, tbLane.isEnable = fasle. * If (change “Huyện”) * dbWard.isEnable = true, dbWard.SelectedItem = NULL,dbWard.ListDistrict=getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * tbLane.text = “”, tbLane.isEnable = false. * If (change “Xã”) * tbLane.text = “”, tbLane.isEnable = false   Khi ấn chọn nút “Chỉnh sửa” thì hệ thống sẽ kiểm tra định dạng các thông tin và đưa các thông tin lên cơ sở dữ liệu.   * If (dbProvince.selectedItem != NULL) * If (dbDistrict.selectedItem == NULL || dbWard.selectedItem == NULL || tbLane.text = “”) => nhận MSG01. * If (dbDistrict.selectedItem != NULL) * If (dbWard.selectedItem == NULL || tbLane.text = “”) => nhận MSG01. * If (dbWard.selectedItem != NULL) * If (tbLane.text = “”) => nhận MSG01. |

### UC12: Xem thông tin người bán

|  |  |
| --- | --- |
| **Name** | **Xem thông tin người bán** |
| **Description** | Chức năng cho phép người dùng đăng nhập vào hệ thống |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng chọn mục “Seller information” trong “left menu” |
| **Pre-condition** | Đăng nhập với tư cách khách hàng. |
| **Post-condition** | * Vào màn hình “Viewing Screen” của “Seller information”. |

#### Activities Flow



*Figure 18:View Seller information Activities Flow*

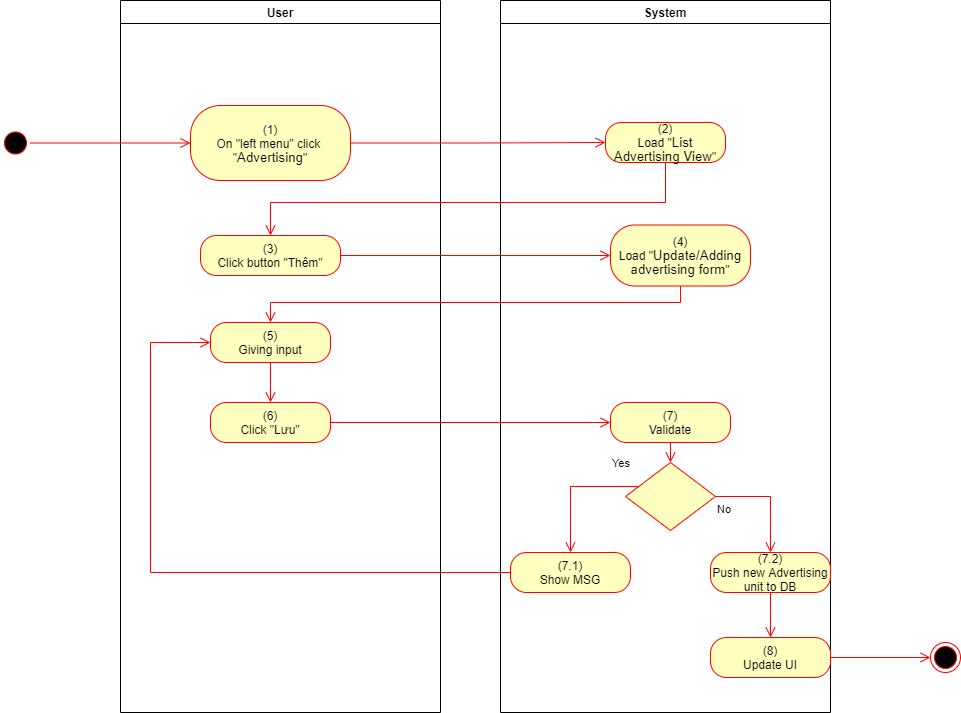
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Viewing Screen” khi người dùng thực hiện chọn mục “Seller information” trong “left Menu”. |
| *(2)* | *BR2* | **Business Rules:**  Khi người dùng ấn chọn vào mục “Seller information” trong “left menu” hệ thống thực hiện tải các thông tin của người bán lên từ cơ sở dữ liệu theo Format sau:   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  |  | [mainImage] | |  |  | | [nameComapy]  [AllCaps] | | | | | | [Video] | | | | | | [List Product Image] | | | | | | [List location]  [Format] = [lane]+ “, ” +[ward] + “, ” + [district] + “, ” + [province] + “.” | | | | | |  |  |  |  | |  * sellerInformation = getFromDB("select \* from sellerInformation”). * mainImage = sellerInformation.mainImage. * nameCopany = sellerInformation.nameCompany * video = sellerInfomation.video * listProductImage = sellerInformation.listProductImage |

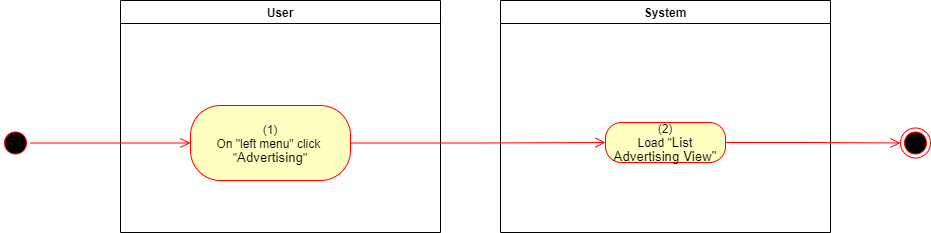
### UC13: CRUD Quảng cáo

|  |  |
| --- | --- |
| **Name** | **CRUD Quảng cáo** |
| **Description** | Hỗ trợ thực hiện CRUD đối với quảng cáo của người bán |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng chọn mục “Advertising” trong “left menu” |
| **Pre-condition** | Đăng nhập với tư cách “BGD” hoặc “Phòng kinh doanh” |
| **Post-condition** | * CRUD được Quảng cáo theo thao tác người bán |

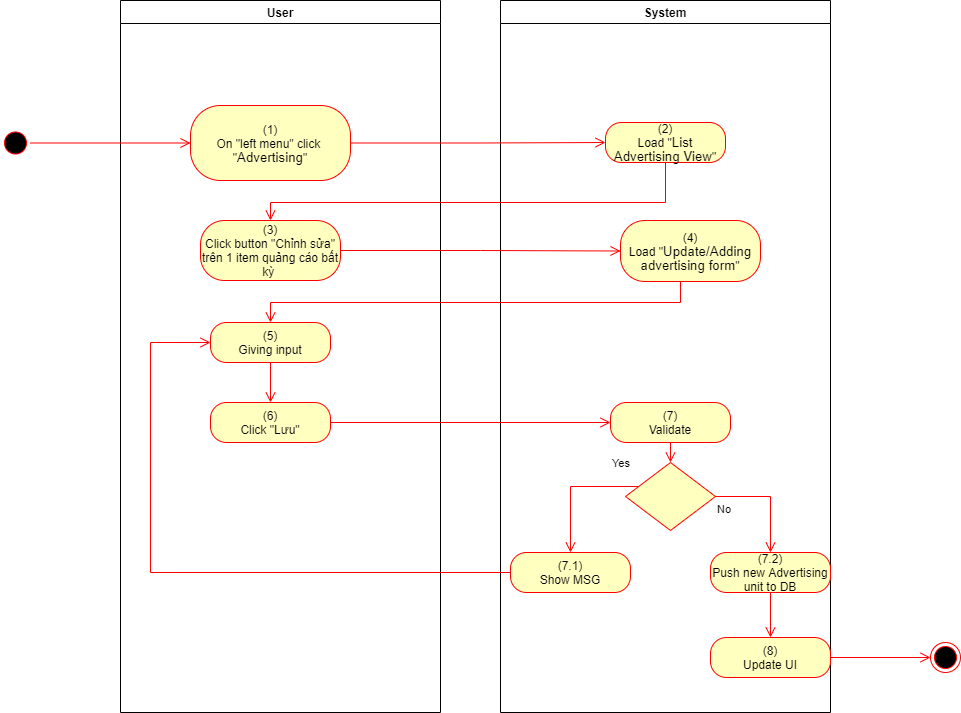
#### Activities Flow



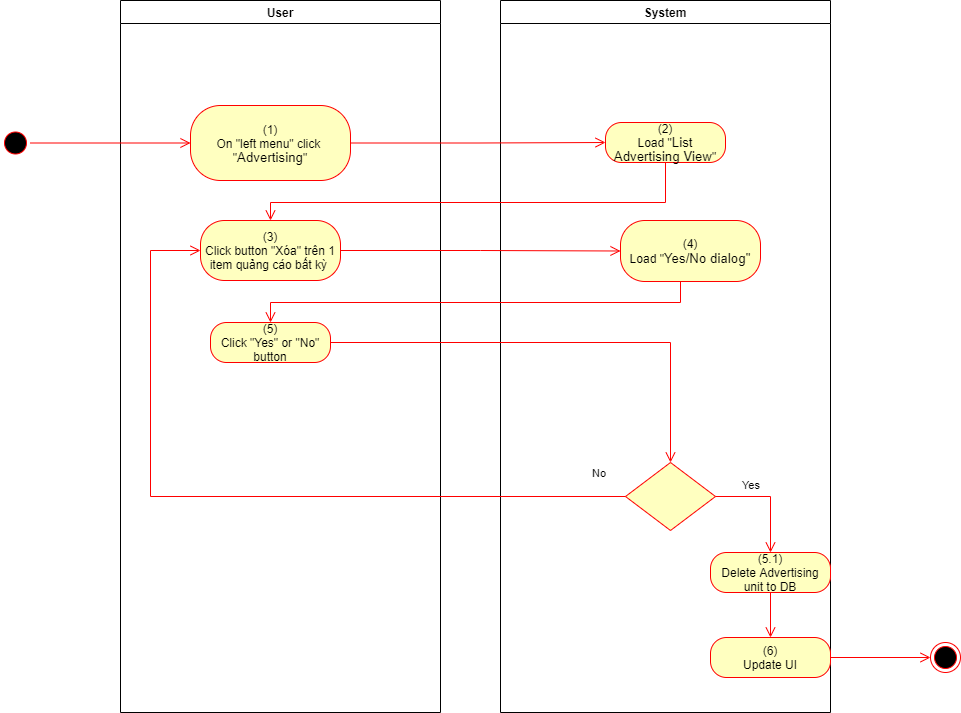
*Figure 19: Create Advertising Activities Flow*

**

*Figure 20: Read Advertising Activities Flow*

******

*Figure 21: Update Advertising Activities Flow*

******

*Figure 22: Delete Advertising Activities Flow*

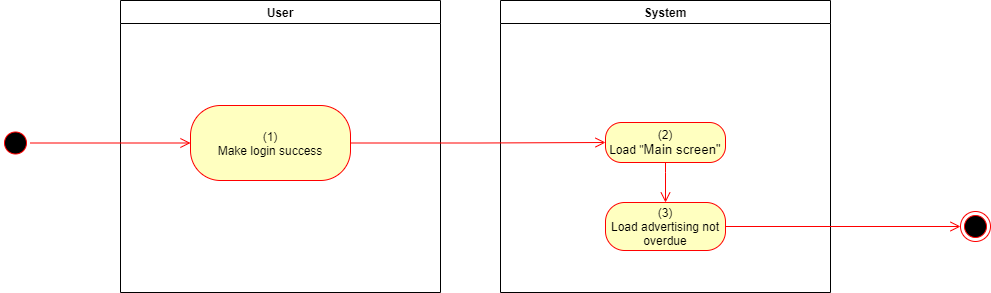
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Advertising View” khi người dùng thực hiện ấn chọn mục “Advertising” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Để có thể tiến hành xem các quảng cáo đã được tạo sẵn, tiến hành ấn chọn vào mục “Advertising” ở “left menu”, lúc này hệ thống sẽ mở màn hình “List Advertising View” và tải listAdvertising (danh sách các quảng cáo trên cơ sở dữ liệu) và hiển thị listDisplay (danh sách quảng cáo sẽ hiển thị):   * If (open “List AdvertisingView”) => listAdvertising = getListFromDB(“Select \* from advertising”) * dbFilter.selectedStatus = “ALL” => listDisplay = listAdvertising. * Khi mở “List Advertising View” thì tiến hành tải tất cả quảng cáo từ database về máy. Khi đó dbFilter (tức drop box dùng để chọn các kiểu lọc) sẽ cài đặt mặc định là “ALL” và danh sách hiển thị sẽ bằng toàn bộ danh sách có trên database.   Để có thể lọc danh sách các quảng cáo, ta tiến hành chọn các lựa chọn có trong Drop Box thanh lọc. Khi đó listDisplay sẽ được thay đổi lại để hiển thị các quảng cáo có tình trạng như lựa chọn đã chọn trong Drop Box.   * Các lựa chọn trong Drop Box (dbFilter.statusList): “ALL”, “CURRENT”, “OVERDUE”, “FUTURE”. * Each [change dbFilter.selectedStatus] = > listDisplay = [empty list] * If (dbFilter.selectedStatus == “ALL”) => listDisplay = listAdvertising. * If (dbFilter.selectedStatus == “CURRENT”) * Foreach(Ads ads in listAdvertising) => if(ads.startDate < DateTime.Now() && ads.endDate > DateTime.Now()) => listDisplay.add(ads) * If (dbFilter.selectedStatus == “OVERDUE”) * Foreach(Ads ads in listAdvertising) => if(ads.endDate < DateTime.Now()) => listDisplay.add(ads) * If (dbFilter.selectedStatus == “FUTURE”) * Foreach(Ads ads in listAdvertising) => if(ads.startDate > DateTime.Now()) => listDisplay.add(ads)  1. **Create**   Để có thể tạo thêm quảng cáo, ta thực hiện nhấn chọn button “Thêm” ở màn hình “List Advertising View”, khi đó hệ thống sẽ thực hiện load màn hình “Update/Adding advertising form”, khi đó ta sẽ thực hiện nhập các thông tin cần thiết vào, và ấn chọn “Lưu” để thêm quảng cáo vào cơ sở dữ liệu:   * If (Click “Thêm”) => open(“Update/Adding advertising form”) * tbNameAds.text = “” (nơi nhập tên quảng cáo (nameAds)) * tbdescriptionAds.text = “” (nơi nhập mô tả (descriptionAds)) * tbProductID.text = “” (nơi nhập ld product để có thể dẫn đến nơi mua (productID)) * imageAds = NULL (nơi chọn ảnh Quảng cáo) * videoAds = NULL (nơi chọn video Quảng cáo) * startDate = “” (nơi chọn ngày bắt đầu quảng cáo) * endDate = “” (nơi chọn ngày kết thúc quảng cáo) * Khi click button “thêm” thì sẽ mở màn hình “Update/Adding advertising form” và set các giá trị đều là rỗng để người dùng tiến hành nhập vào hệ thống.   Để tiến hành gửi quảng cáo lên database ta tiến hành ấn chọn button “Lưu” để hệ thống kiểm tra định dạng các thông tin và gửi lên database:   * If (Click “Lưu”) * If (tbNameAds.text == “” || tbdescriptionAds.text == “” || tbProductID.text == “” || (imageAds == NULL && videoAds = NULL)|| startDate == “” || endDate == “”) => nhận MSG01 * If (!isExist(tbProductID.text)) => nhận MSG15 * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDate < endDate) => Nhận thông báo MSG10. * Else => newAds = new Advertising(tbNameAds.text, tbdescriptionAds.text, tbProductID, imageAds, videoAds, startDate, endDate, [updateTime = DateTime.Now()]) => pushDataToDB(newAds)) * listDisplay. Add (newAds) * Trong khi kiểm tra các trường có rỗng hay không thì lưu ý: Video và Image có thể có 1 trong 2 hoặc cả 2, nhưng cả 2 không được phép cùng rỗng. * Sau khi kiểm tra tất cả đều ổn (không nhận được bất cứ MSG nào) thì hệ thống sẽ đưa tất cả thông tin vào class Advertising để đưa lên Database và thêm vào listDisplay để cập nhật UI. * If (Click “Thoát”) => Back() * Nếu click “No” thì hệ thống sẽ quay lại màn hình “List Advertising View” mà không có bất cứ sự thay đổi nào dù có sai sót hay không.  1. **Update**   Để có thể chỉnh sửa quảng cáo, ta thực hiện nhấn chọn button “chỉnh sửa” ở trên 1 Item quảng cáo bất kỳ (selectedAds) trong màn hình “List Advertising View”, khi đó hệ thống sẽ thực hiện load màn hình “Update/Adding advertising form”, khi đó ta sẽ cập nhập các thông tin cần thiết vào, và ấn chọn “Lưu” để chỉnh sửa quảng cáo trên cơ sở dữ liệu và cập nhật UI:   * If (Click “Thêm”) => open(“Update/Adding advertising form”) * tbNameAds.text = selectedAds.nameAds * tbdescriptionAds.text = selectedAds.descriptionAds * tbProductID.text = selectedAds.productID * imageAds = selectedAds.imageID * videoAds = selectedAds.videoAds * startDate = selectedAds.startDate * endDate = selectedAds.endDate * Khi click button “chỉnh sửa” thì sẽ mở màn hình “Update/Adding advertising form” và set các giá trị là các giá trị của item được chọn.   Để tiến hành gửi quảng cáo lên database ta tiến hành ấn chọn button “Lưu” để hệ thống kiểm tra định dạng các thông tin và gửi lên database:   * If (Click “Lưu”) * If (tbNameAds.text == “” || tbdescriptionAds.text == “” || tbProductID.text == “” || (imageAds == NULL && videoAds = NULL)|| startDate == “” || endDate == “”) => nhận MSG01 * If (!isExist(tbProductID.text)) => nhận MSG15 * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDate < endDate) => Nhận thông báo MSG10. * Else => => updatedAds = new Advertising(tbNameAds.text, tbdescriptionAds.text, tbProductID, imageAds, videoAds, startDate, endDate, [updateTime = DateTime.Now()]) => pushDataToDB(updatedAds). * listDisplay.Update(updatedAds) * Trong khi kiểm tra các trường có rỗng hay không thì lưu ý: Video và Image có thể có 1 trong 2 hoặc cả 2, nhưng cả 2 không được phép cùng rỗng. * Sau khi kiểm tra tất cả đều ổn (không nhận được bất cứ MSG nào) thì hệ thống sẽ đưa tất cả thông tin vào class Advertising để đưa lên Database và cập nhật lại dữ liệu ở listDisplay để cập nhật lại UI. * If (Click “Thoát”) => Back() * Nếu click “No” thì hệ thống sẽ quay lại màn hình “List Advertising View” mà không có bất cứ sự thay đổi nào dù có sai sót hay không.  1. **Delete**   Để có thể xóa quảng cáo, ta thực hiện nhấn chọn button “xóa” ở trên 1 Item quảng cáo bất kỳ (selectedAds) trong màn hình “List Advertising View”, khi đó hệ thống sẽ thực hiện load Yes/No dialog, và ấn chọn “Yes” để xóa quảng cáo trên cơ sở dữ liệu và cập nhật UI:   * If (Click “xóa”) * If (Click “Yes”) => deleteOnDB(selectedAds) => listDisplay.delete(selectedAds) * If (Click “No”) => Back() * Nếu chọn Button “xóa” thì Yes/No dialog sẽ hiện ra, và nếu click chọn “Yes” thì hệ thống sẽ tiến hành xóa đi item đã chọn ở cả UI và database. * Nếu click “No” thì hệ thống sẽ quay lại màn hình “List Advertising View” mà không có bất cứ sự thay đổi nào. |

### UC14: Xem quảng cáo (khách hàng)

|  |  |
| --- | --- |
| **Name** | **Xem quảng cáo (khách hàng)** |
| **Description** | Chức năng cho phép người dùng đăng nhập vào hệ thống |
| **Actor** | 1. User 2. No – account user |
| **Trigger** | * Khi người dùng mở ứng dụng (vào Main screen) |
| **Pre-condition** | None |
| **Post-condition** | * Xem được quảng cáo. |

#### Activities Flow



*Figure 23: View Advertising Activities Flow*

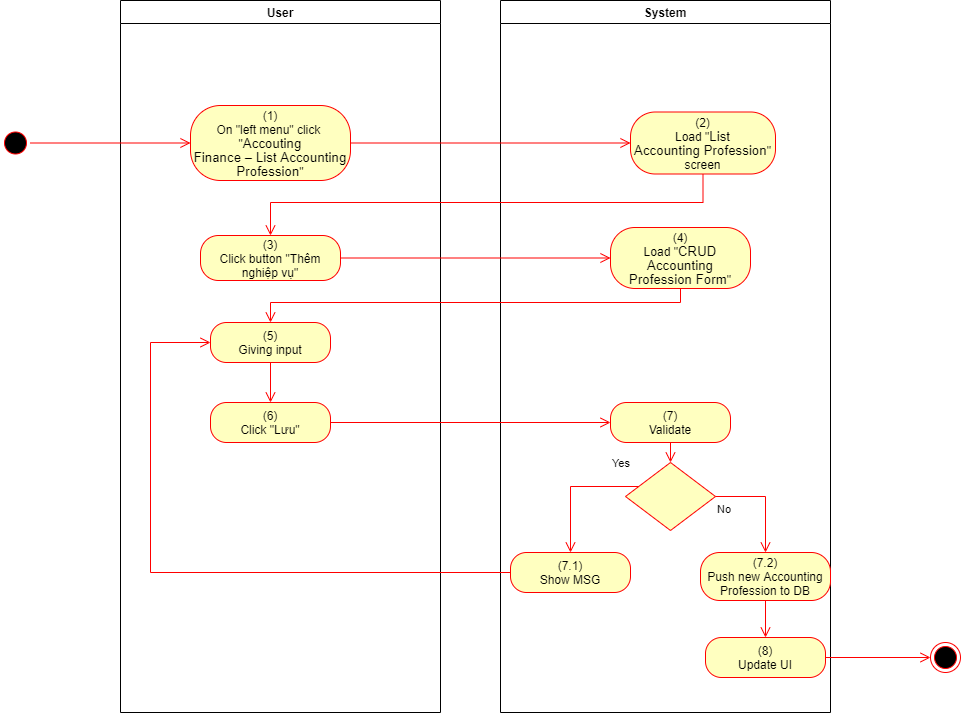
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải các Quảng cáo còn trong thời gian sử dụng. |
| *(2)* | *BR2* | **Business Rules:**  Khi người dùng mở được đến “Main screen”, trên cùng sẽ là các quảng cáo hiển thị như sau:   * [listDisplay = loadFromDB(“Select \* from Advertising where Advertising.status = “CURRENT”)] * Nếu có cả Video và Hình ảnh (videoAds và imageAds)  |  |  | | --- | --- | | [videoAds] | [imageAds] | | O O O  [số lượng “O” = listDisplay.size]  (đây là nơi thể hiện và nơi ấn để đi đến quảng cáo bất kỳ) | |  * Nếu chỉ có Video hoặc Hình ảnh (videoAds hoặc imageAds):  |  | | --- | | [videoAds/imageAds] | | O O O  [số lượng “O” = listDisplay.size]  (đây là nơi thể hiện và nơi ấn để đi đến quảng cáo bất kỳ) |   Khi người dùng ấn chọn vào quảng cáo, hệ thống sẽ chuyển sang “Product Detail View” của sản phẩm quảng cáo đang quảng cáo:   * If (Click “AdvertiseAds”) => moveToDetailProduct(selectedAdvertiseAds.productID) |

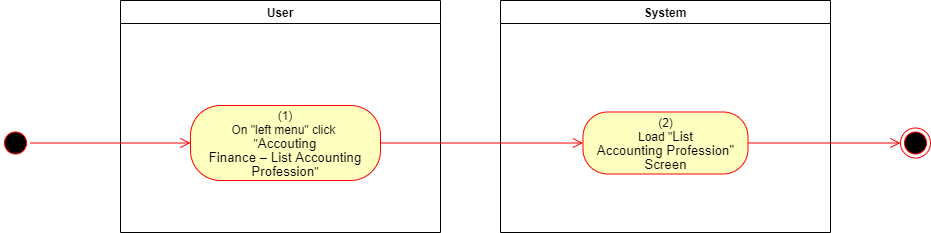
### UC15: CRUD nghiệp vụ kế toán

|  |  |
| --- | --- |
| **Name** | **CRUD nghiệp vụ kế toán** |
| **Description** | Chức năng cho phép CRUD các nghiệp vụ kế toán |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng chọn mục “Accounting Finance – CRUD Accounting Profession” trong “left menu”. |
| **Pre-condition** | Đăng nhập với tư cách BGD hoặc Phòng kế toán. |
| **Post-condition** | * CRUD nghiệp vụ kế toán như thao tác người dùng. |

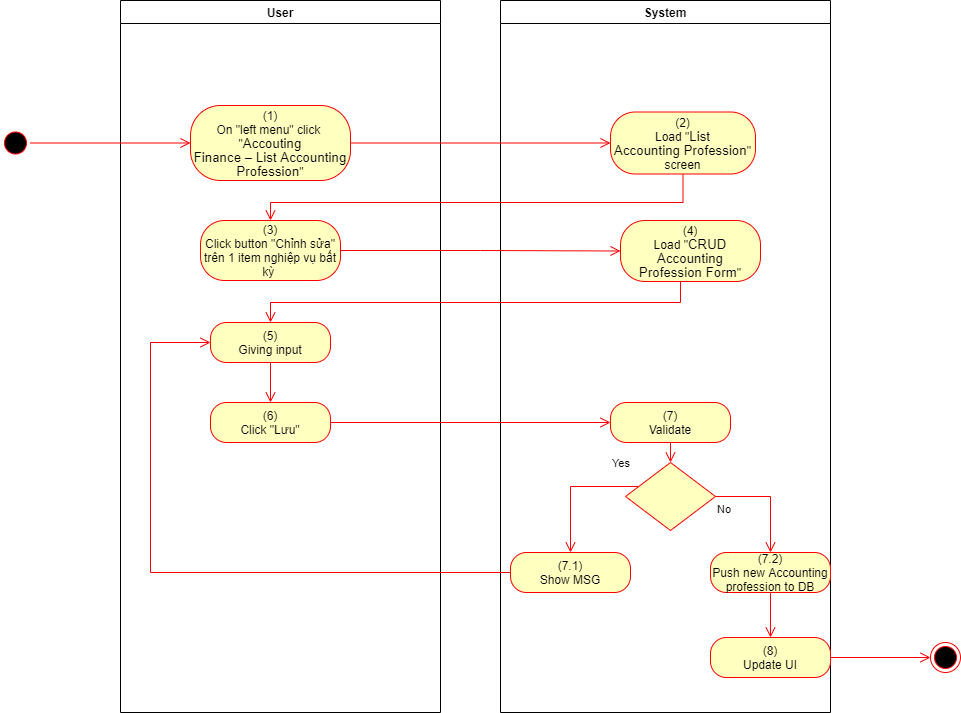
#### Activities Flow



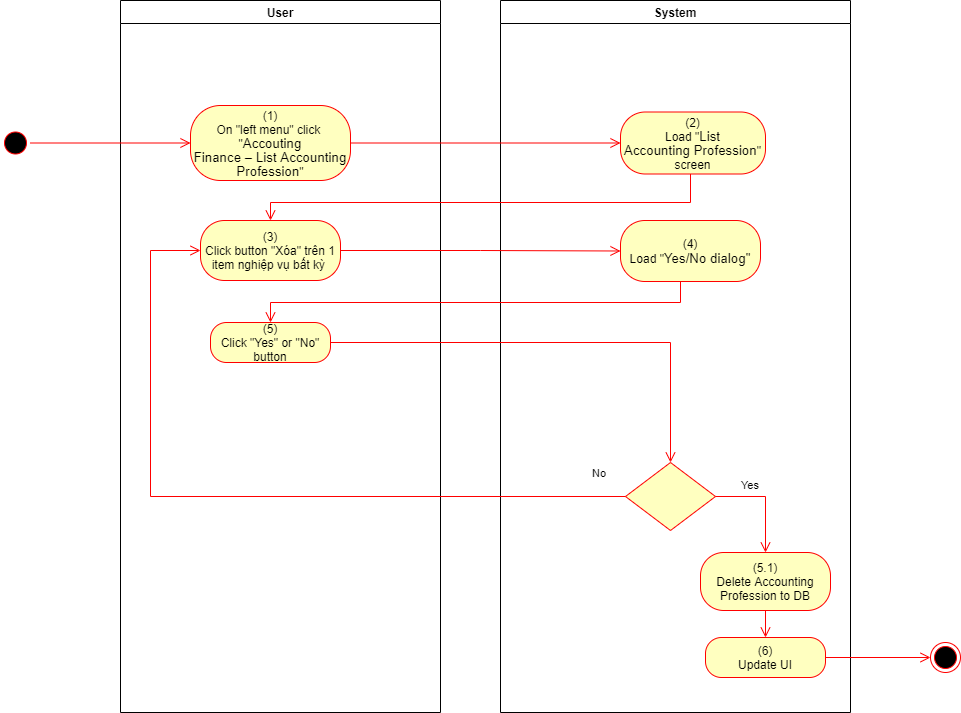
*Figure 24: Create Accounting Profession Activities Flow*

**

*Figure 25: Read Accounting Profession Activities Flow*

**

*Figure 26: Update Accounting Profession Activities Flow*

**

*Figure 27: Delete Accouting Profession Activities Flow*

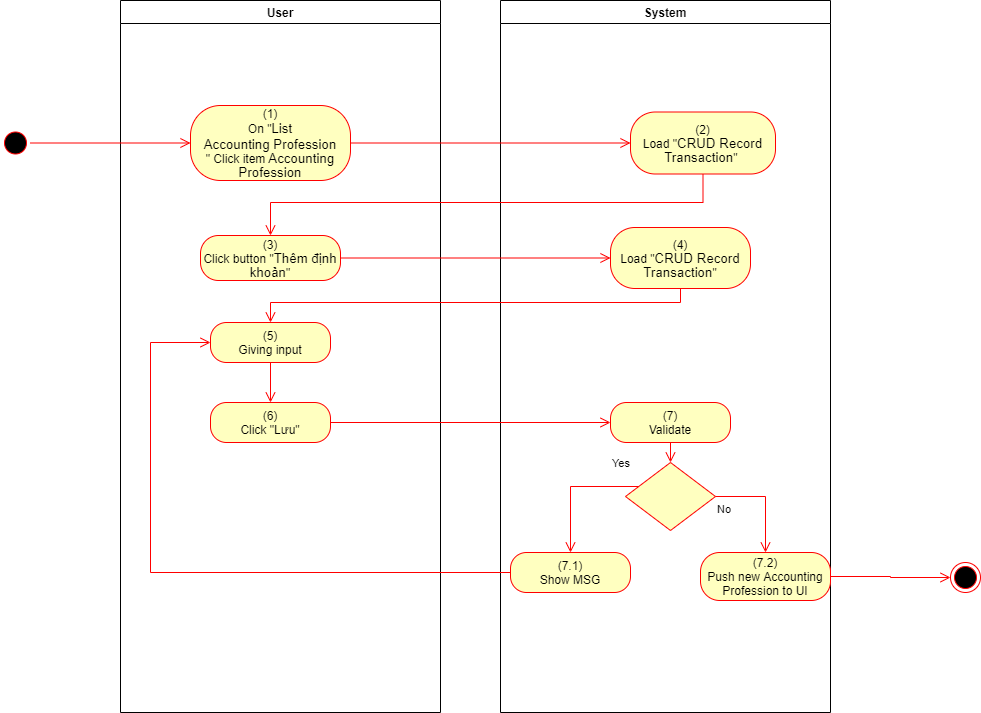
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Accounting Profession” khi tiến hành ấn chọn mục “Accounting Finance – List Accounting Profession” trong “left menu”. |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Để có thể xem các nghiệp vụ kế toán ta tiến hành ấn chọn mục “Accounting Finance – List Accounting Profession” trong “left menu”. Hệ thống sẽ tải dữ liệu trên database về listDislay để đưa lên màn hình.   * listDisplay = listFormDB(“Select \* from Accounting\_Proffession”)  1. **Create**   Để tiến hành tạo một nghiệp vụ kế toán mới ta tiến hành ấn chọn button “Thêm nghiệp vụ” ở màn hình “List Accounting Profession”, hệ thống sẽ hiển thị “CRUD Accounting Profession Form”, để tiến hành nhập các thông tin cần thiết để tạo nghiệp vụ kế toán:   * If (Click “Thêm nghiệp vụ”) => openForm(“CRUD Accounting Profession Form”) * nameProff = “” (tên nghiệp vụ kế toán) * createProffTime = NULL (ngày chỉ định tạo nghiệp vụ) * listRecord = NULL (danh sách định khoản của nghiệp vụ) * Lúc ấn chọn button “Thêm nghiệp vụ” hệ thống tiến hành mở “CRUD Accounting Profession Form” với tất cả thông tin cần nhập đều trống, do là Create nên các trường id, update time sẽ bị unable.   Để tiến hành thêm định khoản cho các nghiệp vụ kế toán ta nhấn chọn button “Thêm định khoản” để mở “CRUD Record Transaction” bắt đầu **UC17.**   * If (Click “Thêm định khoản”) => openForm(“CRUD Record Transaction”). => start **UC16**   Để tiến hành cập nhật định khoản vừa tạo cho nghiệp vụ kế toán ta nhấn chọn button “chỉnh sửa” của một Item định khoản bất kỳ để lại mở “CRUD Record Transaction” và bắt đầu **UC16.**   * If (Click “chỉnh sửa”) => openForm(“CRUD Record Transaction”). => start **UC16**   Để gửi nghiệp vụ kế toán mới lên database ta tiến hành ấn chọn nút “Lưu” hệ thống kiểm tra định dạng các thông tin nhập và gửi lên trên database:   * If (Click “Lưu”) * If(nameProff == “” || createProffTime == “”) => nhận MSG01. * Else => newProff = new (nameProff, createProffTime, listRecord, DateTime.Now())=> SaveToDB(newProff) * **[Điều kiện này ở UC16 mục có \*]** * listDisplay.add(newProff) * Khi nhấn chọn button “lưu” hệ thống sẽ kiểm tra thông tin như trên (listRecord có thể nhập sau), nếu không phải gửi bất kỳ MSG nào hệ thống sẽ đưa dữ liệu lên database và cập nhật lại UI. * If (Click “Thoát”) => Back() * Khi nhấn chọn button “thoát” dù cho thông tin có sai hay không đều sẽ không lưu lại mà quay về màn hình “List Accounting Profession”.  1. **Update**   Để tiến hành cập nhật một nghiệp vụ kế toán ta tiến hành ấn chọn button “Chỉnh sửa” ở một Item nghiệp vụ kế toán bất kỳ (selectedProff) ở màn hình “List Accounting Profession”, hệ thống sẽ hiển thị “CRUD Accounting Profession Form”, để tiến hành nhập các thông tin cần thiết để tạo nghiệp vụ kế toán:   * If (Click “Chỉnh sửa”) => openForm(“CRUD Accounting Profession Form”) * nameProff = selectedProff.nameProff * createProffTime = selectedProff. createProffTime * listRecord = selectedProff.listRecord * idProff = selectedProff.id * updateDate = selectedProff.updateDate * Lúc ấn chọn button “Chỉnh sửa” hệ thống tiến hành mở “CRUD Accounting Profession Form” với tất cả thông tin của selectedProff.   Để tiến hành tạo thêm định khoản cho các nghiệp vụ kế toán ta nhấn chọn button “thêm định khoản” để mở “CRUD Record Transaction” bắt đầu **UC17.**   * If (Click “Thêm định khoản”) => openForm(“CRUD Record Transaction”). => start **UC16**   Để tiến hành cập nhật định khoản cho các nghiệp vụ kế toán ta nhấn chọn button “chỉnh sửa” của một Item định khoản bất kỳ để mở “CRUD Record Transaction” bắt đầu **UC16.**   * If (Click “chỉnh sửa”) => openForm(“CRUD Record Transaction”). => start **UC16**   Để cập nhật nghiệp vụ kế toán lên database ta tiến hành ấn chọn nút “Lưu” hệ thống kiểm tra định dạng các thông tin nhập và gửi lên trên database:   * If (Click “Lưu”) * If(nameProff == “” || createProffTime == “”) => nhận MSG01. * Else => newProff = new (nameProff, createProffTime, listRecord, DateTime.Now())=> updateToDB(newProff) * **[Điều kiện này ở UC16 mục có \*]** * listDisplay.update(newProff) * Khi nhấn chọn button “lưu” hệ thống sẽ kiểm tra thông tin như trên (listRecord có thể nhập sau), nếu không phải gửi bất kỳ MSG nào hệ thống sẽ đưa dữ liệu lên database và cập nhật lại UI. * If (Click “Thoát”) => Back() * Khi nhấn chọn button “thoát” dù cho thông tin có sai hay không đều sẽ không lưu lại mà quay về màn hình “List Accounting Profession”.  1. **Delete**   Để tiến hành xóa một nghiệp vụ kế toán ta tiến hành ấn chọn button “Xóa” ở một Item nghiệp vụ kế toán bất kỳ (selectedProff) ở màn hình “List Accounting Profession”, hệ thống sẽ hiển thị một Yes/No dialog để tiến hành chọn lựa việc có chắc chắn xóa hay không:   * If (Click “xóa”) * If (Click “Yes”) => deleteOnDB(selectedProff) => listDisplay.delete(selectedProff) * If (Click “No”) => Back() |

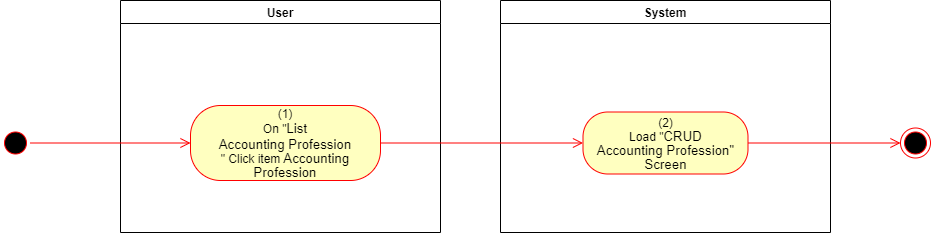
### UC16: CRUD định khoản

|  |  |
| --- | --- |
| **Name** | **CRUD định khoản** |
| **Description** | Chức năng cho phép CRUD các định khoản trong nghiệp vụ kế toán |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người đang ở “CRUD Accounting Profession” sau đó chọn vào/tạo một nghiệp vụ kế toán bất kỳ, ấn vào “thêm”/”chỉnh sửa”/”xóa” để mở “CRUD Record Transaction” |
| **Pre-condition** | Đăng nhập với tư cách BGD hoặc Phòng kế toán. |
| **Post-condition** | * CRUD định khoản kế toán như thao tác người dùng. |

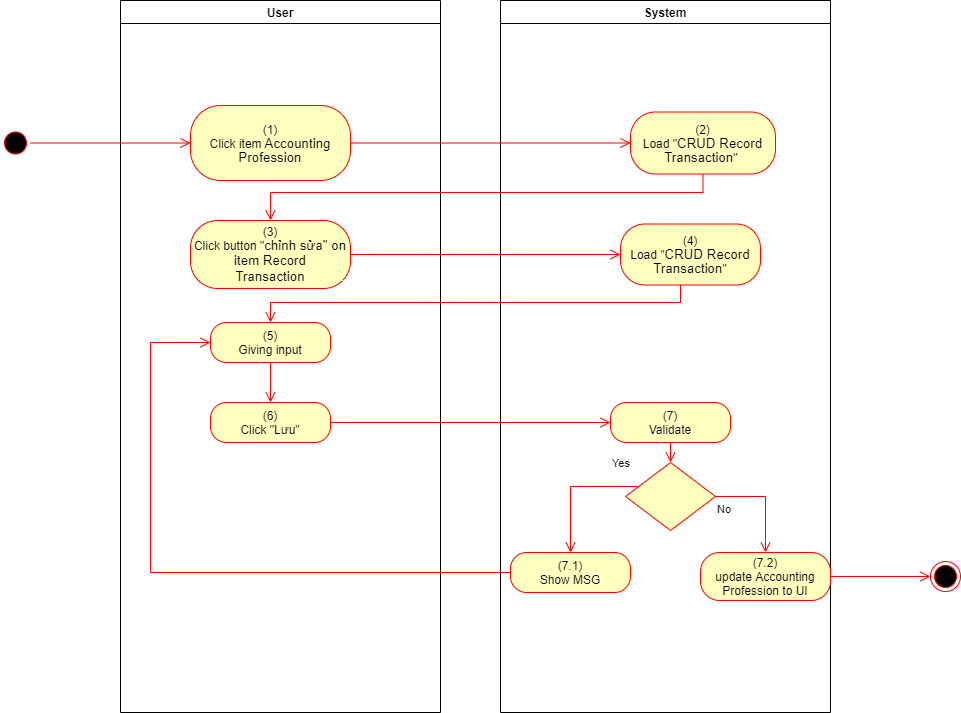
#### Activities Flow



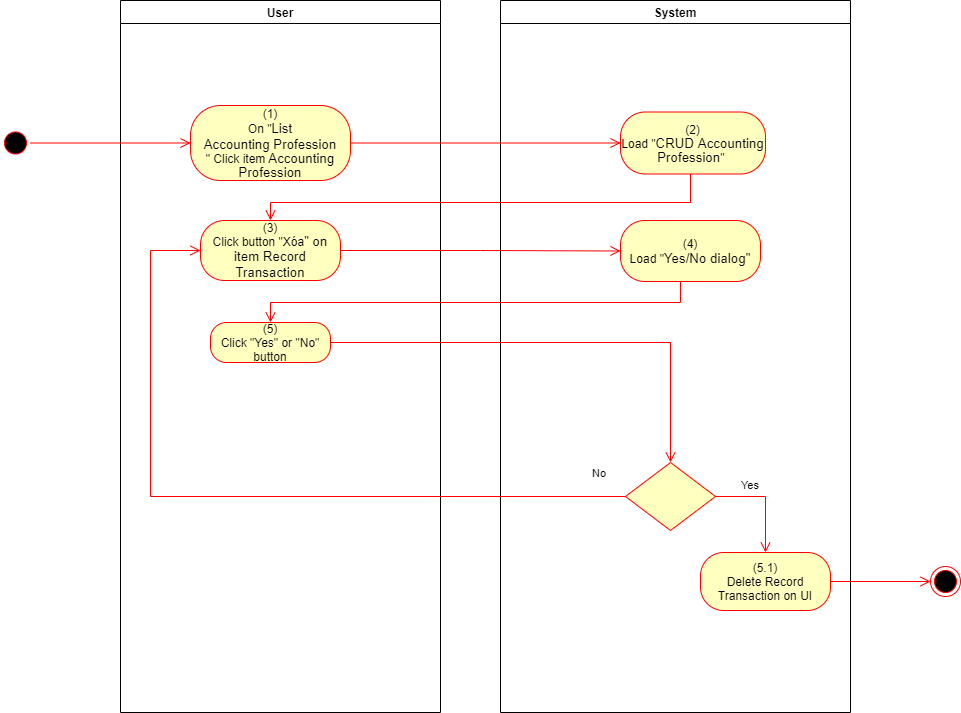
*Figure 28:Create Record Transaction Activities Flow*

**

*Figure 29:Read Record Transaction Activities Flow*

**

*Figure 30:Update Record Transaction Activities Flow*

**

*Figure 31:Delete Record Transaction Activities Flow*

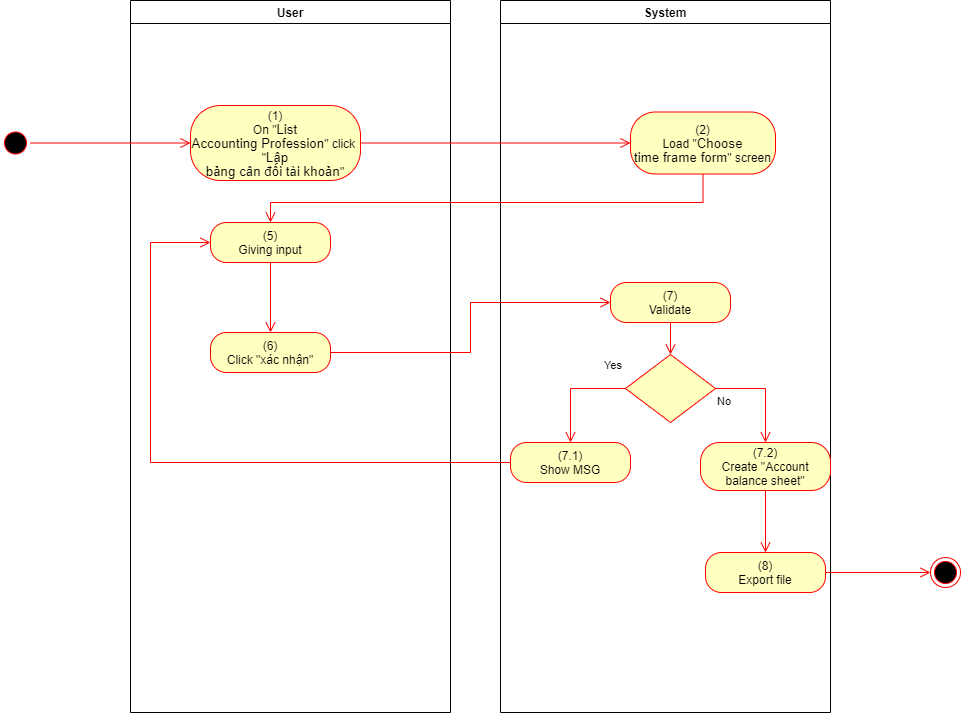
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Xem như hiện tại đang ở một màn hình “CRUD Accounting Profession”, ta ấn chọn vào vào “thêm”/”chỉnh sửa”/”xóa” để mở “CRUD Record Transaction” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Để xem các định khoản kế toán ta tiến hành ấn chọn ”Chỉnh sửa” ở một Item nghiệp vụ kế toán bất kỳ (selectedProff) ở màn hình “List Accounting Profession”, hệ thống sẽ hiển thị “CRUD Accounting Profession Form”, nơi có chứa các định khoản của một nghiệp vụ kế toán:   * listDisplayRecord = listFormDB(“Select \* from Accounting\_Proffession”)[Position].listRecord   Hoặc lúc tạo thêm nghiệp vụ kế toán (lúc ấn chọn “Thêm nghiệp vụ” ở màn hình “List Accounting Profession”, hệ thống hiển thị “CRUD Accounting Profession” ở dạng thêm), thì lúc đã tạo một số định khoản cũng sẽ hiển thị các định khoản đó.   * listDisplayRecord = [InputForRecordList] * các định khoản vừa tạo chưa đưa lên Database.  1. **Create:**   Tại form “CRUD Accounting Profession” dù bất kỳ đang “thêm” hay “chỉnh sửa”, để thực hiện tạo định khoản kế toán ta ấn chọn vào button “Thêm định khoản” để mở form “CRUD Record Transaction” để thực hiện nhập các thông tin cần thiết để tạo định khoản:   * If (Click “Thêm định khoản”) => open(“CRUD Record Transaction”) * dbAccount.data = loadListFromDB(“Select \* from accounting\_acount”) (dbAccount là drop box chứa các tài khoản kế toán) * dbAccount.selectedAccount = NULL * listCredit = NULL (danh sách số phát sinh bên có) * listDebit = NULL (danh sách số phát sinh bên nợ) * tbFinal = “” (số kết chuyển cuối kỳ, tự động tính khi thay đổi 2 listCredit và listDebit) * Khi click button “Thêm định khoản” hệ thống tiến hành mở “CRUD Record Transaction” và đặt các giá trị trên là rỗng.   Để tiến hành đưa định khoản lên database ta tiến hành ấn chọn button “Lưu”, để hệ thống kiểm tra định dạng các thông tin và đưa lên cơ sở dữ liệu.   * If (Click “Lưu”) * If (dbAccount.selectedAccount == NULL || (listCredit == NULL && listDebit == NULL)) => nhận MSG01. * Foreach(Long item in listCredit) => if(item.contain([NONE\_NUMBER\_CHARACTER])) => nhận MSG03. * Foreach(Long item in listDedit) => if(item.contain([NONE\_NUMBER\_CHARACTER])) => nhận MSG03. * Else => newRecord = new Record(dbAccount.selectedAccount, listCredit, listDebit, tbFinal, ) => listDisplayRecord.add(newRecord) * Khi ấn nút “lưu” hệ thống sẽ tự động kiểm tra định dạng các thông tin đã nhập, nếu không có bất kỳ MSG nào thì sẽ đưa record vào trong list hiển thị. (chưa đưa lên database, chỉ đưa lên database khi ấn chọn button “lưu” ở form “CRUD Accounting Profession” (update/create Nghiệp vụ kế toán)). * Note: listCredit và listCredit có thể có 1 trong 2 NULL nhưng không thể cả 2 đều NULL   Việc tính toán “kết chuyển” (tức final) sẽ được tính như sau:   * If ( account.startWith(“1” or “2”) => listDebit.Sum() – listCredit.Sum(). * Đối với loại tài khoản bắt đầu bằng 1 hoặc 2 thì sẽ lấy tổng bên Nợ trừ tổng bên Có. * If (account.StartWith(“3” or “4”)) => listCredit.Sum() – listDecit.Sum(). * Đối với loại tài khoản bắt đầu bằng 3 hoặc 4 thì sẽ lấy tổng bên Có trừ tổng bên Nợ. * If (account.startWith(“5” or “6” or “8” or “7” or “9”)) => final.isEnable = true.   **\* Khi ấn “lưu” ở “CRUD Record Transaction” phải đảm bảo phương trình kế toán, tài sản bằng nguồn vốn (đảm bảo phương pháp ghi sổ kép).**   * **sumAllCredit = 0; foreach(record item in listRecord) => foreach (Long item in record.listCredit) => sumAllCredit += item** * **sumAllDebit = 0; foreach(record item in listRecord) => foreach (Long item in record.listDedit) => sumAllDebit += item** * **if (sumAllCredit != sumAllDebit) => nhận MSG 16.** * **Thực hiện tình tổng tất cả số phát sinh bên Có và bên Nợ của tất cả tài khoản định khoản. Nếu không bằng nhau có nghĩ là nghiệp vụ đang định khoản sai.**  1. **Update:**   Tại form “CRUD Accounting Profession” dù bất kỳ đang “thêm” hay “chỉnh sửa”, để thực hiện tạo định khoản kế toán ta ấn chọn vào button “chỉnh sửa” ở Item định khoản bất kỳ (selectedRecord) để mở form “CRUD Record Transaction” để thực hiện nhập các thông tin cần thiết để tạo định khoản (Nếu đang thực hiện “thêm” ở “CRUD Accounting Profession” thì việc sau khi đã tạo 1 định khoản thì cũng có thể thực hiện “chỉnh sửa” định khoản đó):   * If (Click “Chỉnh sửa”) => open(“CRUD Record Transaction”) * dbAccount.data = loadListFromDB(“Select \* from accounting\_acount”) * dbAccount.selectedAccount = selectedRecord.selectedAccount * listCredit = selectedRecord.listCredit * listDebit = selectedRecord.listCredit * tbFinal = selectedRecord.final * Khi click button “Chỉnh sửa” trên một item định khoản bất kỳ hệ thống tiến hành mở “CRUD Record Transaction” và đặt các giá trị trên theo item định khoản đó.   Để tiến hành lưu lại thay đổi của định khoản lên database ta tiến hành ấn chọn button “Lưu”, để hệ thống kiểm tra định dạng các thông tin và đưa lên cơ sở dữ liệu.   * If (Click “Lưu”) * If (dbAccount.selectedAccount == NULL || (listCredit == NULL && listDebit == NULL)) => nhận MSG01. * Foreach(Long item in listCredit) => if(item.contain([NONE\_NUMBER\_CHARACTER])) => nhận MSG03. * Foreach(Long item in listDedit) => if(item.contain([NONE\_NUMBER\_CHARACTER])) => nhận MSG03. * Else => updateRecord = new Record(dbAccount.selectedAccount, listCredit, listDebit, tbFinal, ) => listDisplayRecord.update(newRecord) * Khi ấn nút “lưu” hệ thống sẽ tự động kiểm tra định dạng các thông tin đã nhập, nếu không có bất kỳ MSG nào thì sẽ thay đổi record trong list hiển thị. (chưa đưa lên database, chỉ đưa lên database khi ấn chọn button “lưu” ở form “CRUD Accounting Profession” (update/create Nghiệp vụ kế toán)). * Note: listCredit và listCredit có thể có 1 trong 2 NULL nhưng không thể cả 2 đều NULL   Việc tính toán “kết chuyển” (tức final) sẽ được tính như sau:   * If ( account.startWith(“1” or “2”) => listDebit.Sum() – listCredit.Sum(). * Đối với loại tài khoản bắt đầu bằng 1 hoặc 2 thì sẽ lấy tổng bên Nợ trừ tổng bên Có. * If (account.StartWith(“3” or “4”)) => listCredit.Sum() – listDecit.Sum(). * Đối với loại tài khoản bắt đầu bằng 3 hoặc 4 thì sẽ lấy tổng bên Có trừ tổng bên Nợ. * If (account.startWith(“5” or “6” or “8” or “7” or “9”)) => final.isEnable = true.  1. **Delete:**   Để tiến hành xóa một định khoản kế toán ta tiến hành ấn chọn button “Xóa” ở một Item định khoản kế toán bất kỳ (selectedRecord) ở màn hình “CRUD Accounting Profession”, hệ thống sẽ hiển thị một Yes/No dialog để tiến hành chọn lựa việc có chắc chắn xóa hay không:   * If (Click “xóa”) * If (Click “Yes”) => listDisplay.delete(selectedProff) * Khi click button “xóa” chỉ cập nhật UI, chưa đưa lên database, chỉ đưa lên database khi ấn chọn button “lưu” ở form “CRUD Accounting Profession” (update/create Nghiệp vụ kế toán) * If (Click “No”) => Back() * Không làm gì mà quay lại form “CRUD Accounting Profession” |

### UC17: Lập bảng cân đối tài khoản

|  |  |
| --- | --- |
| **Name** | **Lập bảng cân đối tài khoản** |
| **Description** | Chức năng cho phép tạo bảng cân đối tài khoản dựa trên các nghiệp vụ đã định khoản |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng ấn chọn button “Lập bảng cân đối tài khoản” trên màn hình “List Accounting Profession” |
| **Pre-condition** | Đăng nhập với tài khoản của BGD hoặc Phòng kế toán. |
| **Post-condition** | * Tạo lập được file bảng cân đối tài khoản (word/excel) |

#### Activities Flow



*Figure 32: Create Account balance sheet Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải form “Choose time frame form” khi người dùng ấn chọn button “Lập bảng cân đối tài khoản” trên màn hình “List Accounting Profession”. |
| *(2)* | *BR2* | **Business Rules:**  Để tiến hành lập bảng cân đối tài khoản, người dùng ấn chọn button “Lập bảng cân đối tài khoản” trên màn hình “List Accounting Profession”, khi đó màn hình “Choose time frame form” sẽ hiển thị, và người dùng tiến hành chọn ngày:   * If (Click “Lập bảng cân đối tài khoản”) => open(“Choose time frame form”) * startDate = NULL (nơi chọn ngày bắt đầu) * endDate = NULL (nơi chọn ngày kết thúc)   Để tiến hành gửi ngày bắt đầu và kết thúc để tạo bảng cân đối tài khoản, ta thực hiện ấn chọn button “Xác nhận” ở form “Choose time frame form”.   * If (Click “Xác nhận”) * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDate < endDate) => Nhận thông báo MSG10. * Else createAccountBalanceSheet(startDate, endDate) [hàm tạo bảng cân đối tài khoản]   Thực hiện tạo bảng cân đối tài khoản như sau:   * Thực hiện duyệt tất cả nghiệp vụ kế toán trong khoản thời gian đã chọn.(listProffForUsing) * Foreach (Proff proff in listProff) => if (proff.createProffDate > startDate && proff.createProffDate < endDate) => listProffForUsing.add(proff). * Thực hiện kiểm tra số lượng số tài khoản riêng rẽ, và tính tổng số SPS bên có và SPS bên nợ và tổng “Kết chuyển” (final mỗi loại). * listRecordForUsing [Mỗi item sẽ gồm, account,tổng SPS bên có, tổng SPS bên nợ, tổng Final] * Foreach (Proff proff in listProff) => Foreach (Record record in proff ) * if(listRecordForUsing.contains(record.account)) => [cộng SPS Có/Nợ của record vào trong item trùng account (item.account == record.account) của listRecordForUsing] * else listRecordForUsing.add(record)   Theo đó bảng cân đối tài khoản sẽ trình bày như sau:  \*Số dư đầu kỳ sẽ do người dùng nhập (sau khi xuất file)   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Mã TK | Tên TK | Số dư đầu kỳ | | Số phát sinh trong kỳ | | Số dư cuối kỳ | | | Nợ | Có | Nợ | Có | Nợ | Có | | [Account] | [Account.name] | x | x | [sumDebit] | [sumCredit] | x | x | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |   \*Số dư cuối kỳ sẽ lấy:  - Đối với tài khoản bắt đầu bằng “1” “2” => [tổng nợ đầu kỳ] + [tổng nợ trong kỳ] – [Tổng có trong kỳ] => số sư âm/dương ghi vào [Nợ cuối kỳ]  - Đối với tài khoản bắt đầu bằng “3” “4” => [tổng có đầu kỳ] + [tổng có trong kỳ] – [Tổng nợ trong kỳ] => số sư âm/dương ghi vào [Có cuối kỳ] |

### UC18: Lập bảng cân đối kế toán

|  |  |
| --- | --- |
| **Name** | **Lập bảng cân đối kế toán** |
| **Description** | Chức năng cho phép tạo bảng cân đối kế toán dựa trên bảng cân đối tài khoản |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng ấn chọn button “Lập bảng cân đối kế toán” trên màn hình “List Accounting Profession” |
| **Pre-condition** | Đăng nhập với tài khoản của BGD hoặc Phòng kế toán. |
| **Post-condition** | * Tạo lập được file bảng cân đối kế toán (word/excel) |

#### Activities Flow



*Figure 33:Create Accounting balance sheet Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải form “Choose time frame form” khi người dùng ấn chọn button “Lập bảng cân đối kế toán” trên màn hình “List Accounting Profession”. |
| *(2)* | *BR2* | **Business Rules:**  Để tiến hành lập bảng cân đối kế toán, người dùng ấn chọn button “Lập bảng cân đối kế toán” trên màn hình “List Accounting Profession”, khi đó màn hình “Choose time frame form” sẽ hiển thị, và người dùng tiến hành chọn ngày:   * If (Click “Lập bảng cân đối kế toán”) => open(“Choose time frame form”) * startDate = NULL (nơi chọn ngày bắt đầu) * endDate = NULL (nơi chọn ngày kết thúc)   Để tiến hành gửi ngày bắt đầu và kết thúc để tạo bảng cân đối kế toán, ta thực hiện ấn chọn button “Xác nhận” ở form “Choose time frame form”.   * If (Click “Xác nhận”) * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDate < endDate) => Nhận thông báo MSG10. * Else createAccountingBalanceSheet(startDate, endDate) [hàm tạo bảng cân đối kế toán]   Thực hiện tạo bảng cân đối kế toán như sau:   * Thực hiện duyệt tất cả nghiệp vụ kế toán trong khoản thời gian đã chọn.(listProffForUsing) * Foreach (Proff proff in listProff) => if (proff.createProffDate > startDate && proff.createProffDate < endDate) => listProffForUsing.add(proff). * Thực hiện kiểm tra số lượng số tài khoản riêng rẽ, và tính tổng số SPS bên có và SPS bên nợ và tổng “Kết chuyển” (final mỗi loại). * listRecordForUsing [Mỗi item sẽ gồm, account,tổng SPS bên có, tổng SPS bên nợ, tổng Final] * Foreach (Proff proff in listProff) => Foreach (Record record in proff ) * if(listRecordForUsing.contains(record.account)) => [cộng SPS Có/Nợ của record vào trong item trùng account (item.account == record.account) của listRecordForUsing] * else listRecordForUsing.add(record)   \*Số dư cuối kỳ sẽ lấy:  - Đối với tài khoản bắt đầu bằng “1” “2” => [tổng nợ đầu kỳ] + [tổng nợ trong kỳ] – [Tổng có trong kỳ] => số sư âm/dương ghi vào [Nợ cuối kỳ]  - Đối với tài khoản bắt đầu bằng “3” “4” => [tổng có đầu kỳ] + [tổng có trong kỳ] – [Tổng nợ trong kỳ] => số sư âm/dương ghi vào [Có cuối kỳ]  - Việc ghi và template sẽ áp dụng như file sau đây:     * Chú ý: * N1xx || N2xx => ghi số dương * C1xx || C2xx => ghi số âm * C3xx || C4xx => ghi số dương. * N3xx || N4xx => ghi số âm. * MSxxx => số cuối kỳ của mã số trong Bảng cân đối kế toán đó. |

### UC19: Lập báo cáo kết quả hoạt động kinh doanh

|  |  |
| --- | --- |
| **Name** | **Lập báo cáo kết quả hoạt động kinh doanh** |
| **Description** | Chức năng cho phép tạo lập báo cáo kết quả hoạt động kinh doanh |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng ấn chọn button “Lập báo cáo kết quả hoạt động kinh doanh” trên màn hình “List Accounting Profession” |
| **Pre-condition** | Đăng nhập với tài khoản của BGD hoặc Phòng kế toán. |
| **Post-condition** | * Tạo lập được file báo cáo kết quả hoạt động kinh doanh (word/excel) |

#### Activities Flow



*Figure 34: Create Business Result Report Activities Flow*

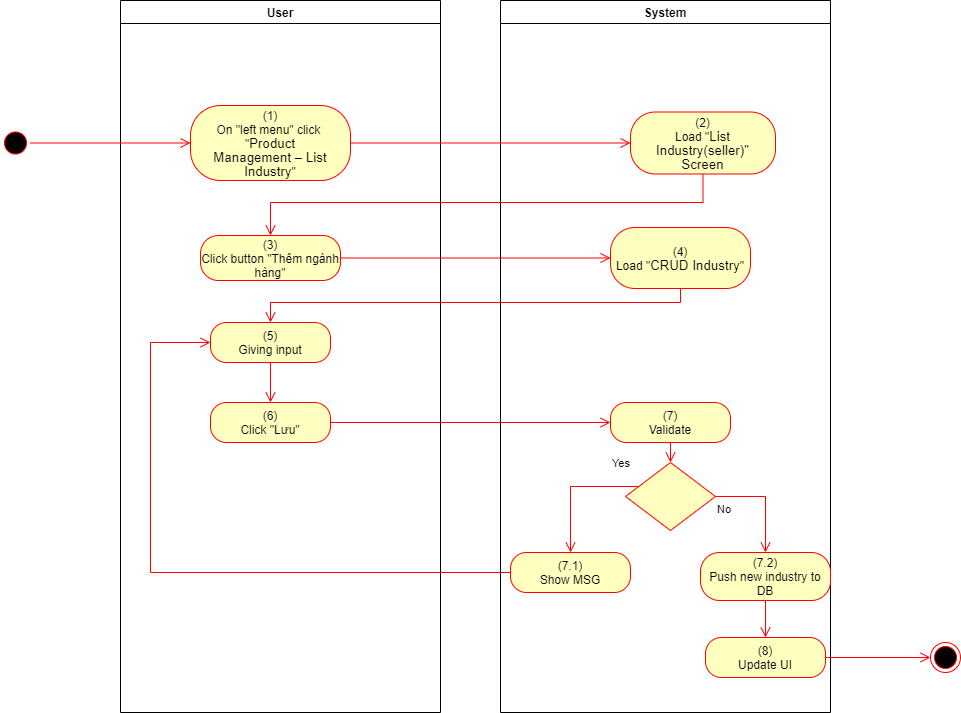
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải form “Choose time frame form” khi người dùng ấn chọn button “Lập báo cáo kết quả hoạt động kinh doanh” trên màn hình “List Accounting Profession”. |
| *(2)* | *BR2* | **Business Rules:**  Để tiến hành lập báo cáo kết quả hoạt động kinh doanh, người dùng ấn chọn button “Lập báo cáo kết quả hoạt động kinh doanh” trên màn hình “List Accounting Profession”, khi đó màn hình “Choose time frame form” sẽ hiển thị, và người dùng tiến hành chọn ngày:   * If (Click “Lập báo cáo kết quả hoạt động kinh doanh => open(“Choose time frame form”) * startDate = NULL (nơi chọn ngày bắt đầu) * endDate = NULL (nơi chọn ngày kết thúc)   Để tiến hành gửi ngày bắt đầu và kết thúc để tạo báo cáo kết quả hoạt động kinh doanh, ta thực hiện ấn chọn button “Xác nhận” ở form “Choose time frame form”.   * If (Click “Xác nhận”) * If (startDate > DateTime.Now()) => Nhận thông báo MSG09. * If (startDate < endDate) => Nhận thông báo MSG10. * Else createBusinessResultReport(startDate, endDate) [hàm tạo báo cáo kết quả hoạt động kinh doanh]   Thực hiện tạo báo cáo kết quả hoạt động kinh doanh như sau:   * Thực hiện duyệt tất cả nghiệp vụ kế toán trong khoản thời gian đã chọn.(listProffForUsing) * Foreach (Proff proff in listProff) => if (proff.createProffDate > startDate && proff.createProffDate < endDate) => listProffForUsing.add(proff). * Thực hiện kiểm tra số lượng số tài khoản riêng rẽ, và tính tổng số SPS bên có và SPS bên nợ và tổng “Kết chuyển” (final mỗi loại). * listRecordForUsing [Mỗi item sẽ gồm, account,tổng SPS bên có, tổng SPS bên nợ, tổng Final] * Foreach (Proff proff in listProff) => Foreach (Record record in proff ) * if(listRecordForUsing.contains(record.account)) => [cộng SPS Có/Nợ của record vào trong item trùng account (item.account == record.account) của listRecordForUsing] * else listRecordForUsing.add(record)   \*Số dư cuối kỳ sẽ lấy:  - Đối với tài khoản bắt đầu bằng “5” “7” => Số dư cuối kỳ = [ tổng bên có] – [trừ bên Nợ]  - Đối với tài khoản bắt đầu bằng “6” “8” => Số dư cuối kỳ = [ tổng bên có] – [tổng bên Nợ]  - Đối với tài khoản bắt đầu bằng “9” => ko có số dư => chỉ có [tổng có] – [tổng nợ”  - Việc ghi và template sẽ áp dụng như file sau đây:     * Chú ý: * N5xx || N7xx => ghi số dương * C5xx || C7xx => ghi số âm * C6xx || C8xx => ghi số dương. * N6xx || N8xx => ghi số âm. * Các ô trống sử dụng công thức bên dưới “tên chỉ tiêu” tương ứng. |

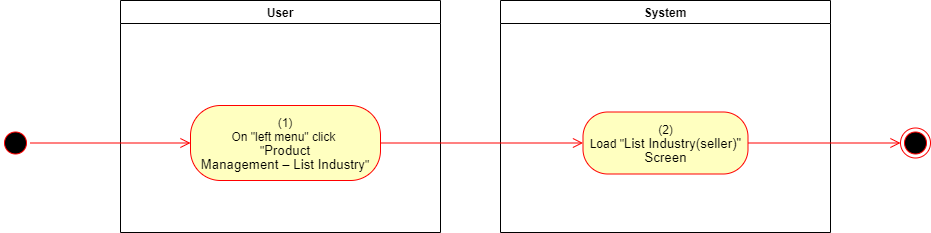
### UC20: CRUD Ngành hàng

|  |  |
| --- | --- |
| **Name** | **CRUD ngành hàng** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc ngành hàng |
| **Actor** | 1. User System |
| **Trigger** | * Khi người dùng mở mục “Product Management – List Industry” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD ngành hàng theo yêu cầu |

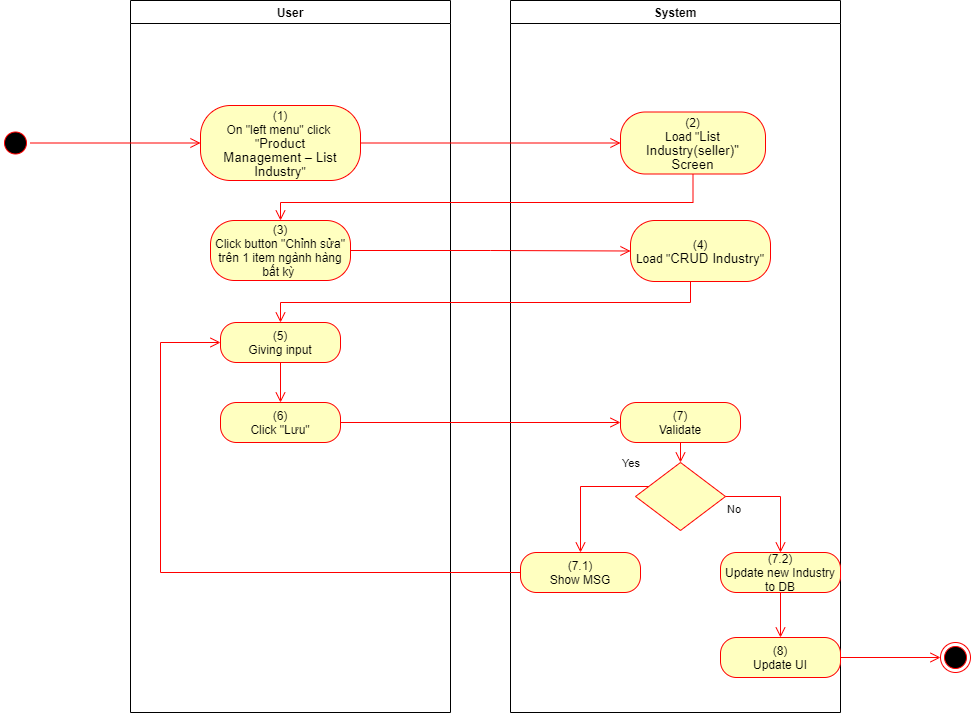
#### Activities Flow



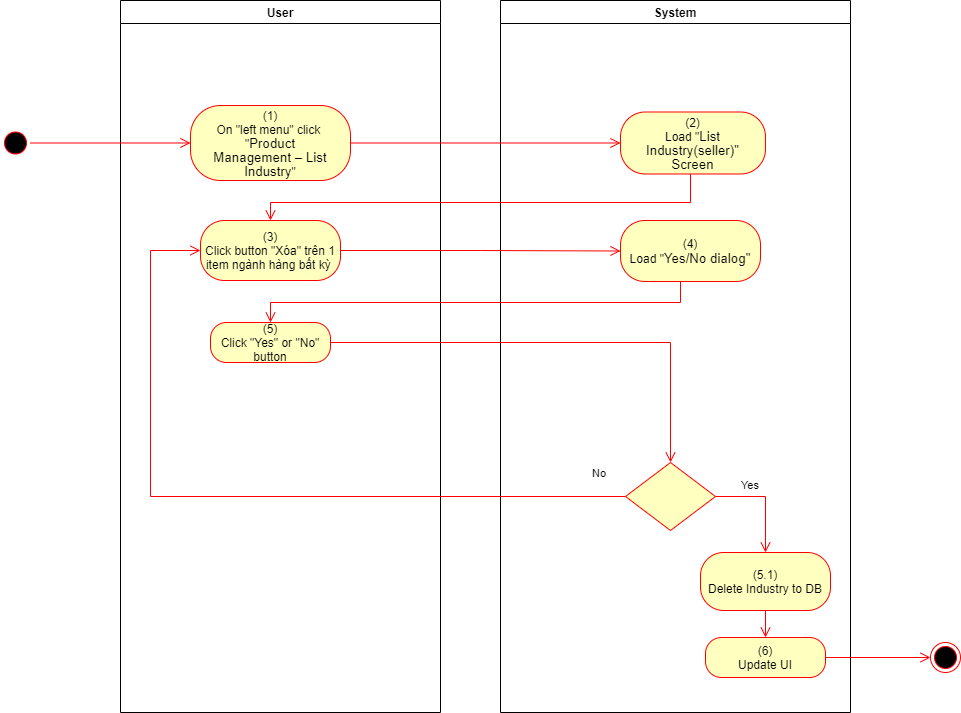
*Figure 35:Create Industry Activities Flow*

**

*Figure 36:Read Industry Activities Flow*

**

*Figure 37:Update Industry Activities Flow*

**

*Figure 38:Delete Industry Activities Flow*

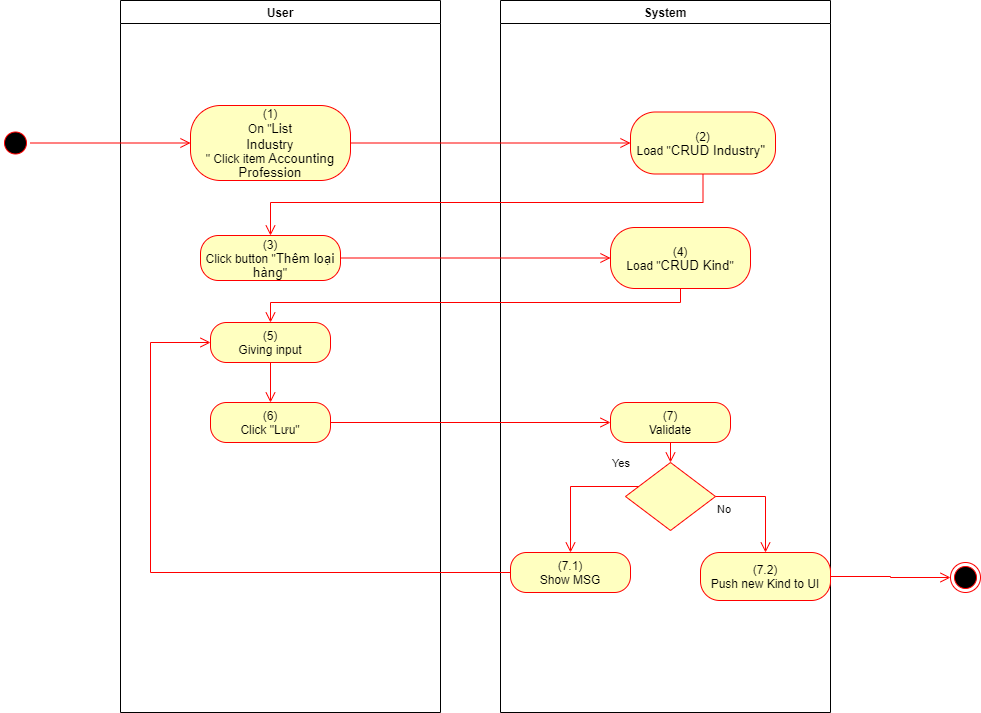
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Industry (Seller)” khi người dùng chọn mục “Product Management – List Industry” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Product management – List Industry”, hệ thống thực hiện tải màn hình “List Industry(seller)” bao gồm các ngành hàng đã được tạo (listIndus):   * If (Click “Product management – List Industry”) => open(“List Industry(seller)”) * listIndus = getListtFromDB(“Selec \* from Industry”)  1. **Create**   Để tiến hành thêm ngành hàng, ta thực hiện ấn chọn button “Thêm ngành hàng” trên màn hình “List Industry(Seller)”, lúc này form “CRUD Industry” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm ngành hàng”) => open(“CRUD Industry”) * nameIndus = “” * listKind = NULL * image = NULL * Khi click button “Thêm ngành hàng” tiến hành mở “CRUD Industry” và set các thông số được nhập là rỗng.   Để tiến hành thêm Loại hàng vào Ngành hàng => **UC22**  Để tiến hành lưu ngành hàng (lưu kèm loại hàng đã thêm haowcj chỉnh sửa) ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameIndus == “” || image == NULL) => nhận MSG01. * Else => newIndustry = new Industry(nameIndus, image, listKind) => pushToDB(newIndustry), listDisplay.add(newIndustry) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Industry”  1. **Update**   Để tiến hành cập nhật ngành hàng, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item ngành hàng bất kỳ (SelectedIndus) trên màn hình “List Industry(Seller)”, lúc này form “CRUD Industry” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Industry”) * nameIndus = selectedIndus.nameIndus * listKind = selectedIndus.listKind * image = selectedIndus.image * Khi click button “Chỉnh sửa” tiến hành mở “CRUD Industry” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành thêm/chỉnh sửa Loại hàng vào Ngành hàng => **UC22**  Để tiến hành lưu ngành hàng (lưu kèm loại hàng đã thêm/chỉnh sửa) ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameIndus == “” || image == NULL) => nhận MSG01. * Else => updateIndustry = new Industry(nameIndus, image, listKind) => pushToDB(updateIndustry), listDisplay.update(updateIndustry) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay).  1. **Delete**   Để tiến hành xóa ngành hàng, ta thực hiện ấn chọn button “Xóa” của một Item ngành hàng bất kỳ (SelectedIndus) trên màn hình “List Industry(Seller)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * If (“Select count\* from Product where Product.idIndustry == selectedIndus.id” > 0) => nhận MSG17. * Else => listDisplay.delete(selectedIndus) => deleteOnDB(selectedIndus) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database nếu hiện tại không có sản phẩm nào thuộc ngành hàng đó. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Industry” |

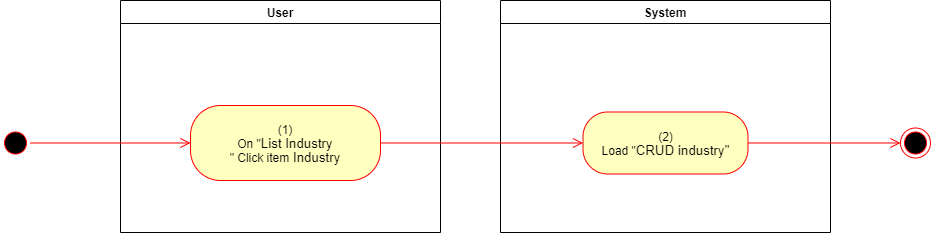
### UC21: CRUD Loại hàng

|  |  |
| --- | --- |
| **Name** | **CRUD loại hàng** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc loại hàng |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng ấn chọn vào một titem loại hàng bất kỳ trong “CRUD Industry” hoặc lúc chọn nút “Thêm loại hàng” trong “CRUD Industry” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD loại hàng theo yêu cầu |

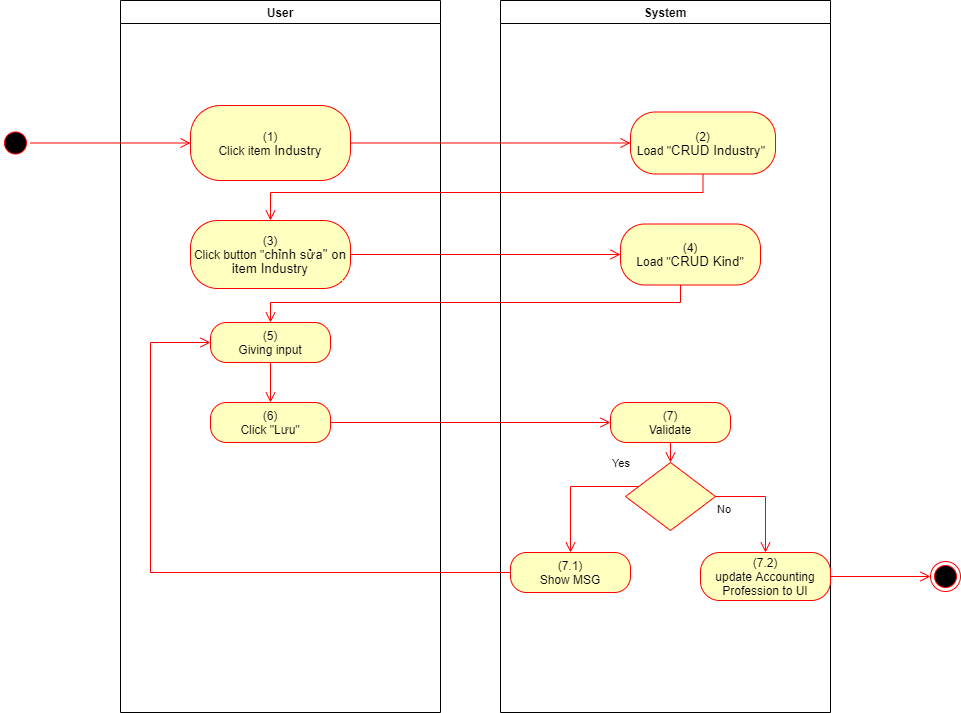
#### Activities Flow



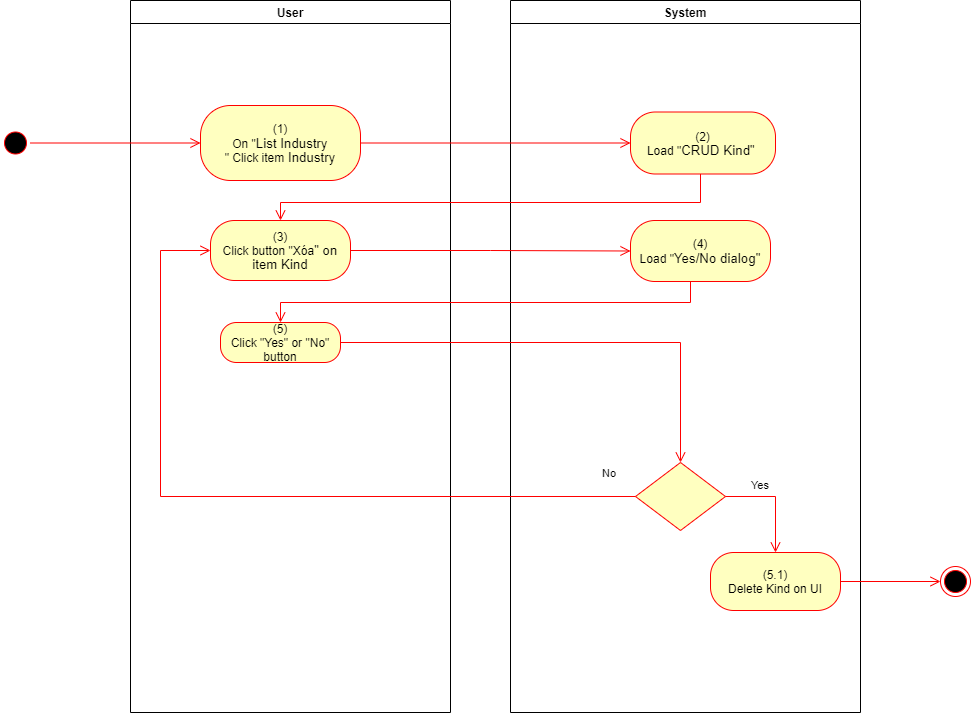
*Figure 39: Create Kind Activities Flow*

**

*Figure 40: Read Kind Activities Flow*

**

*Figure 41: Create Kind Activities Flow*

**

*Figure 42: Create Kind Activities Flow*

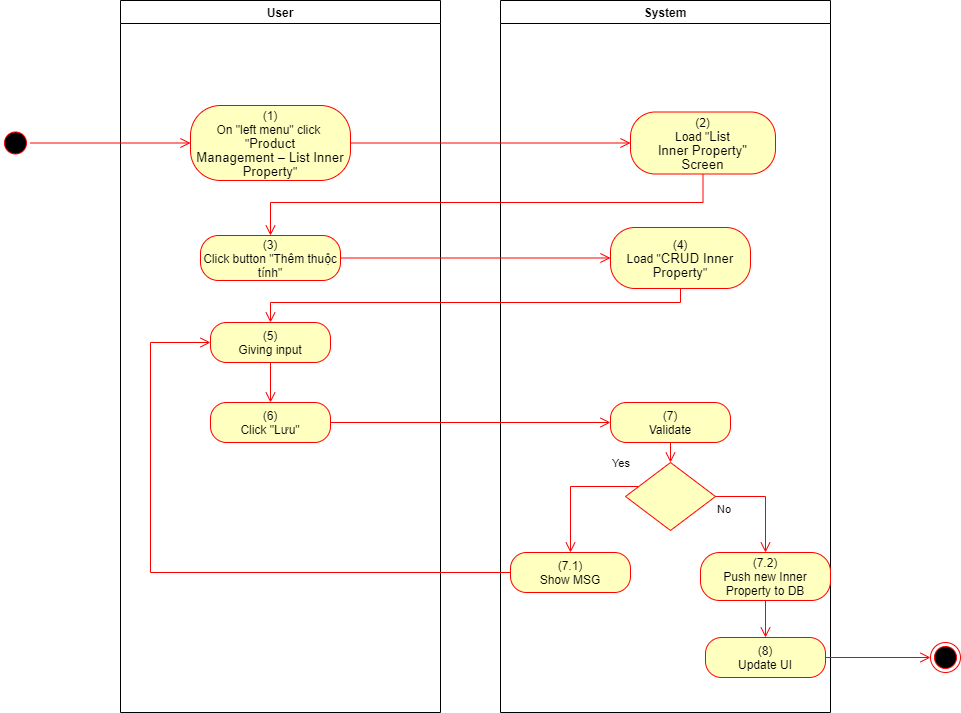
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “CRUD kind ” khi người dùng chọn vào nút “chỉnh sửa” của item bất kỳ trong “CRUD Industry” hoặc lúc chọn button “thêm ngành hàng” trong “CRUD industry” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn button “Chỉnh sửa” của một Item Ngành hàng bất kỳ (selectedIndustry) trong “List Industry”, hệ thống thực hiện tải màn hình “CRUD industry” trong đó có thông tin ngành hàng cừa chọn và danh sách loại hàng (listKind):   * If (Click “Chỉnh sửa”) => open(“CRUD Industry“) * listKind = getListFromDB(“Select \* From Kind where Kind.industryID = selectedIndustry.id”) * Khi tiến hành chỉnh sửa ngành hàng thì cũng sẽ tải danh sách các loại hàng tương ứng lên.  1. **Create**   Để tiến hành thêm ngành hàng, ta thực hiện ấn chọn button “Thêm ngành hàng” trên màn hình “CRUD industry”, lúc này form “CRUD Kind” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm loại hàng”) => open(“CRUD Kind”) * nameKind = “” * imageKind = NULL * Khi click button “Thêm loại hàng” tiến hành mở “CRUD Kind” và set các thông số được nhập là rỗng.   Để tiến hành lưu loại hàng hàng ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và cập nhật UI.   * If (Click “Lưu”) * If (nameKind == “” || imageKind == NULL) => nhận MSG01. * Else => newKind = new Kind(nameKind, image) => listDisplay.add(newKind) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên UI (listDisplay). (Chỉ cập nhật trên database khi ấn “lưu” ở màn hình “CRUD Industry”) * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “CRUD Industry”  1. **Update**   Để tiến hành cập nhật ngành hàng, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item loại hàng bất kỳ (SelectedKind) trên màn hình “CRUD Industry”, lúc này form “CRUD Kind” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Kind”) * nameKind = selectedKind.nameKind * imageKind = selectedKind.imageKind * Khi click buttun “Chỉnh sửa” tiến hành mở “CRUD Kind” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành lưu loại hàng ta thực hiện nhân chọn button “Lưu” trong “CRUD Kind” để tiến hành kiểm tra định dạng thông tin và cập nhật UI.   * If (Click “Lưu”) * If (nameKind == “” || imageKind == NULL) => nhận MSG01. * Else => updateKind = new Kind(nameKind, imageKind) => listDisplay.update(updateKind) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại UI (listDisplay). (Chỉ cập nhật trên database khi ấn “lưu” ở màn hình “CRUD Industry”) * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “CRUD Industry”  1. **Delete**   Để tiến hành xóa ngành hàng, ta thực hiện ấn chọn button “Xóa” của một Item loại hàng bất kỳ (SelectedKind) trên màn hình “CRUD industry”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * If (“Select count\* from Product where Product.idKind = selectedKind.id” >0) => nhận MSG18. * Else => listDisplay.delete(selectedKind) => deleteOnDB(selectedKind) * Khi click button “xóa”sẽ thực hiện xóa ở UI và chỉ xóa cả ở database nếu hiện tại không có sản phẩm nào thuộc ngành hàng đó và ấn chọn “Lưu” ở màn hình “CRUD Industry”. * If (Click “No”) => Back() * Không làm gì mà quay lại form “CRUD Industry” |

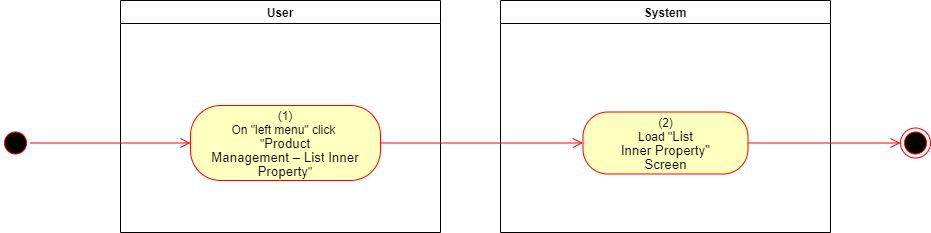
### UC22: CRUD Thuộc tính khác biệt nội sản phẩm

|  |  |
| --- | --- |
| **Name** | **CRUD Thuộc tính khác biệt nội sản phẩm** |
| **Description** | Hỗ trợ thực hiện các thào tác thêm, sửa, xóa, đọc thuộc tính khác biệt nội sản phẩm |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “Product Management – List Inner Property” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD thuộc tính khác biệt nội sản phẩm theo yêu cầu |

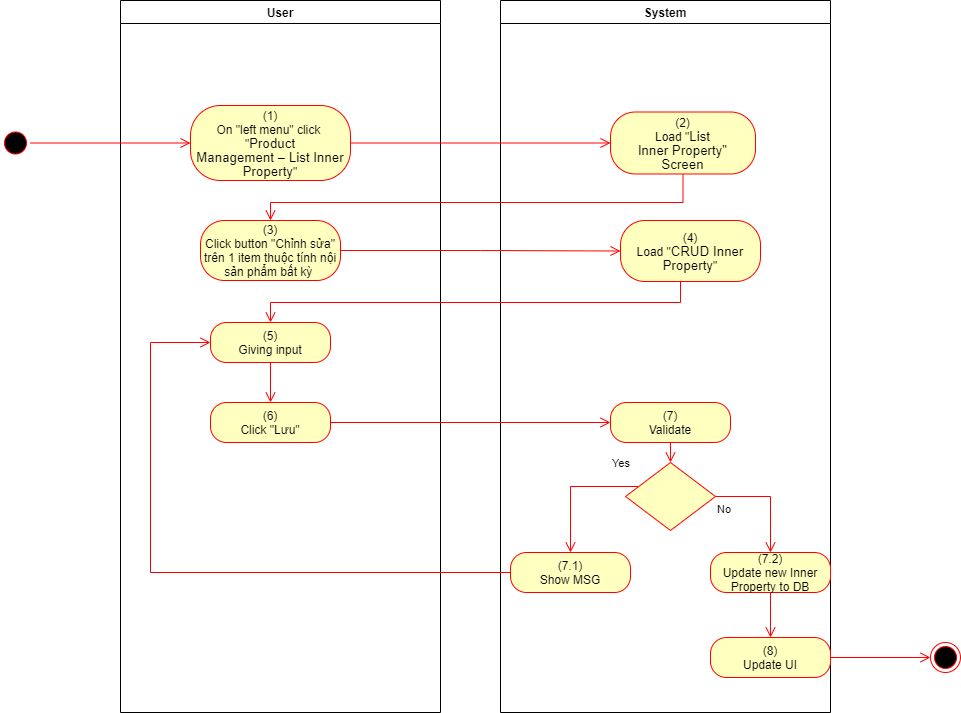
#### Activities Flow



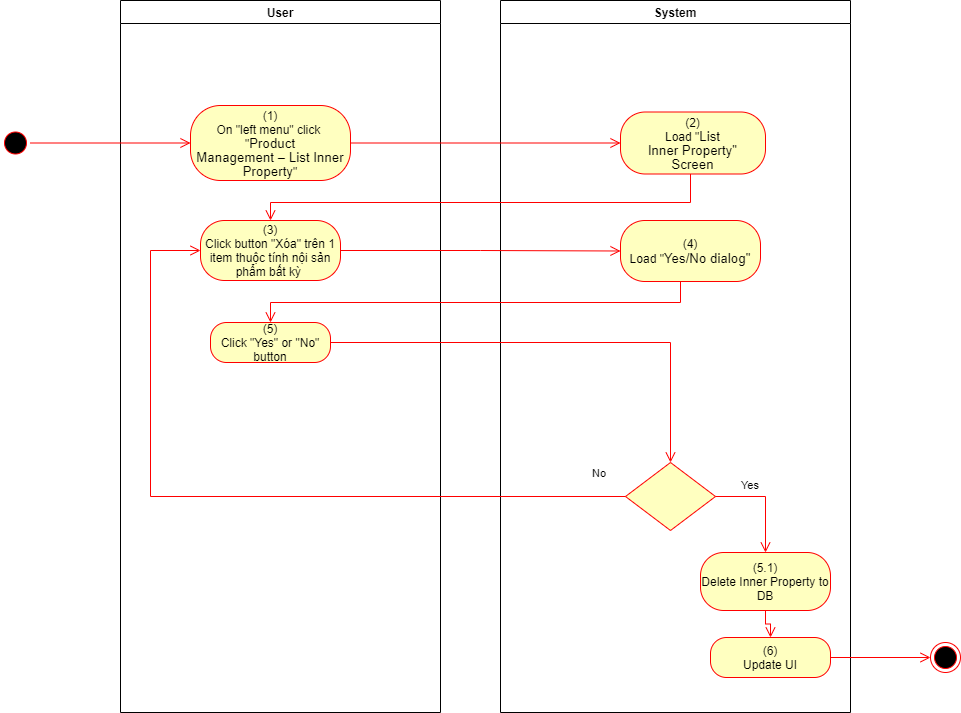
*Figure 43: Create Inner Property Activities Flow*

**

*Figure 44: Read Inner Property Activities Flow*

**

*Figure 45: Update Inner Property Activities Flow*

**

*Figure 46: Delete Inner Property Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Inner Property” khi người dùng chọn mục “Product Management – List Inner Property” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Product management – List Inner Property”, hệ thống thực hiện tải màn hình “List Inner Property” bao gồm các thuộc tính khác biệt nội sản phẩm đã được tạo (listInnerProp):   * If (Click “Product management – List Inner Property”) => open(“List Inner Property”) * listInnerProp = getListtFromDB(“Selec \* from Inner\_Property”)  1. **Create**   Để tiến hành thêm thuộc tính khác biệt nội sản phẩm, ta thực hiện ấn chọn button “Thêm thuộc tính” trên màn hình “List Inner Property”, lúc này form “CRUD Inner Property” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm thuộc tính”) => open(“CRUD Inner Property”) * nameProp = “” * isColor = False * listValue = NULL * Khi click button “Thêm thuộc tính” tiến hành mở “CRUD Inner Property” và set các thông số được nhập là rỗng (isColor là false).   Để tiến hành lưu thuộc tính khác biệt nội sản phẩm ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameProp == “”) => nhận MSG01. * foreach(String item in listValue) => if (!rightFormat(item)) => MSG30 * If (isColor) => foreach(String item in listValue) => if (!item.rightFormat(“#xxxxxx”)) => MSG19 * Else => newInnerProp = new InnerProperty(nameProp, isColor, listValue) => pushToDB(newInnerProp), listDisplay.add(newInnerProp) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Nếu chọn isColor tức thuộc tính là màu sắc thì định dạng của String value phải là “#xxxxxx” mỗi x là 1 chữ số (ko có ký tự đặc biệt) * Các phần tử trong listValue (Danh sách giá trị) sẽ có cú pháp.  |  |  |  | | --- | --- | --- | | [Value] | /// | [Giá tiền gia tăng] |  * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Inner Property”  1. **Update**   Để tiến hành cập nhật thuộc tính khác biệt nội sản phẩm, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item thuộc tính khác biệt nội sản phẩm bất kỳ (SelectedInnẻProp) trên màn hình “List Inner Property”, lúc này form “CRUD Inner Property” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Inner Property”) * nameProp = selectedInnerProp.nameProp * listValue = selectedInnerProp.listValue * isColor = selectedInnerProp.isColor * Khi click button “Chỉnh sửa” tiến hành mở “CRUD Inner Property” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành lưu thuộc tính khác biệt nội sản phẩm ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameProp == “” ) => nhận MSG01. * foreach(String item in listValue) => if (!rightFormat(item)) => MSG30 * If (isColor) => foreach(String item in listValue) => if (!item.rightFormat(“#xxxxxx”)) => MSG19 * Else => updateInnerProp = new InnerProperty(nameProp, isColor, listValue) => pushToDB(updateInnerProp), listDisplay.update(updateInnerProp) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay).  1. **Delete**   Để tiến hành xóa thuộc tính khác biệt nội sản phẩm, ta thực hiện ấn chọn button “Xóa” của một Item thuộc tính khác biệt nội sản phẩm bất kỳ (SelectedInnerProp) trên màn hình “List Inner Property”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * If (“Select count\* from Inner\_Property where Inner\_Property == selectedInnerProp.id” >0) => nhận MSG20. * Else => listDisplay.delete(selectedInnerProp) => deleteOnDB(selectedInnerProp) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database nếu hiện tại không có sản phẩm nào thuộc ngành hàng đó. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Inner Property” |

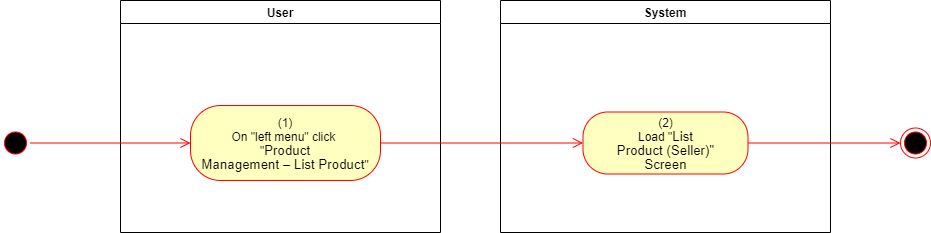
### UC23: CRUD Hàng hóa

|  |  |
| --- | --- |
| **Name** | **CRUD hàng hóa** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc hàng hóa |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “Product Management – List Product (Seller)” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD hàng hóa/ sản phẩm theo yêu cầu |

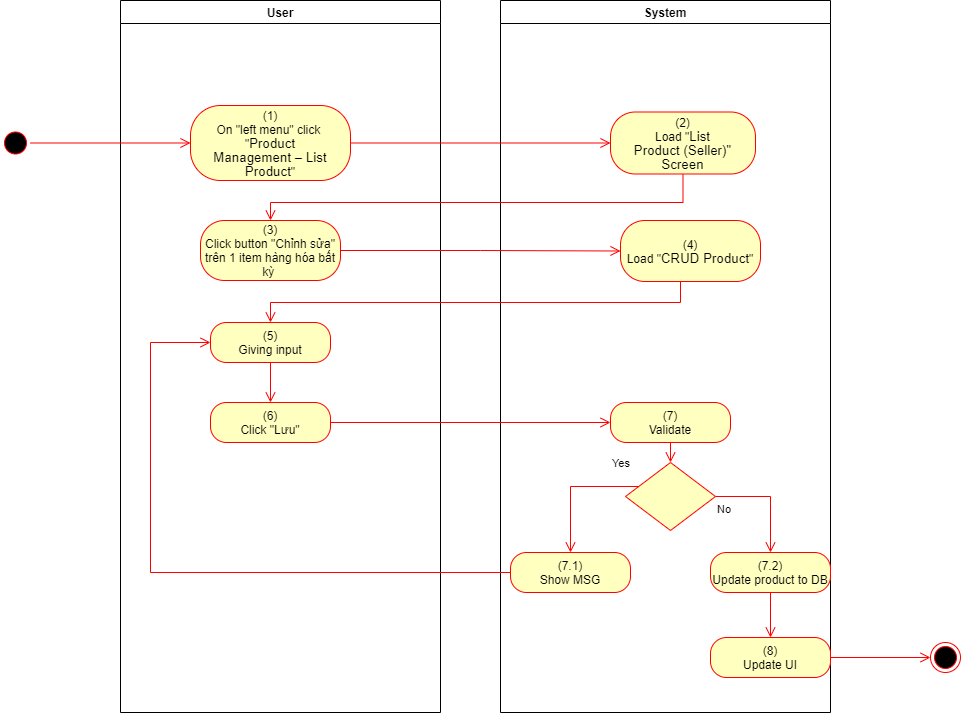
#### Activities Flow



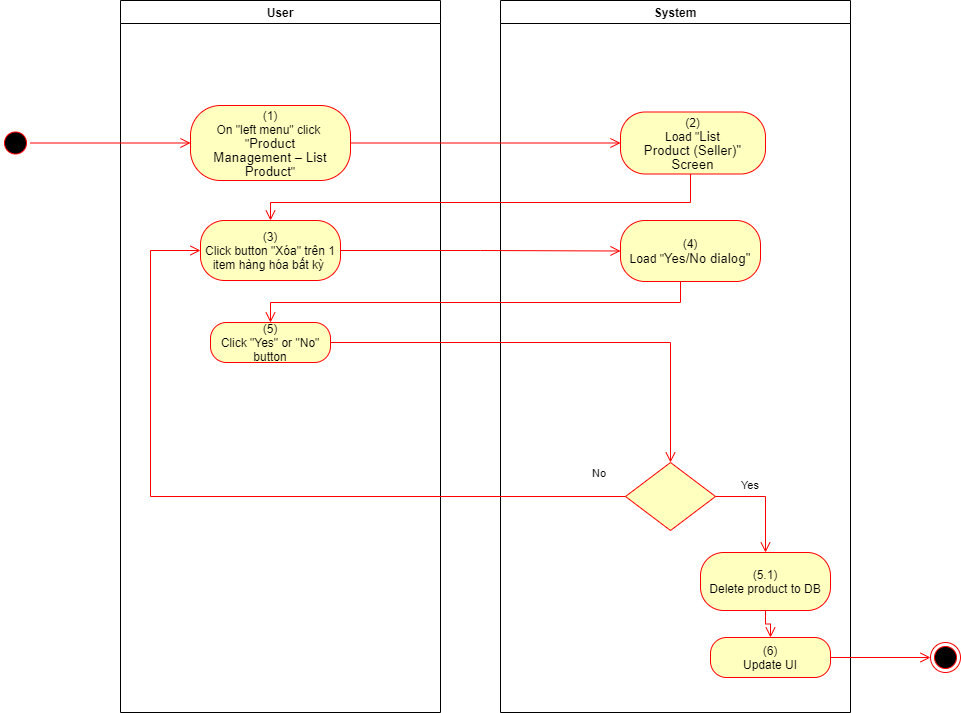
*Figure 47: Create Product Activities Flow*

**

*Figure 48: Read Product Activities Flow*

**

*Figure 49: Update Product Activities Flow*

**

*Figure 50: Delete Product Activities Flow*

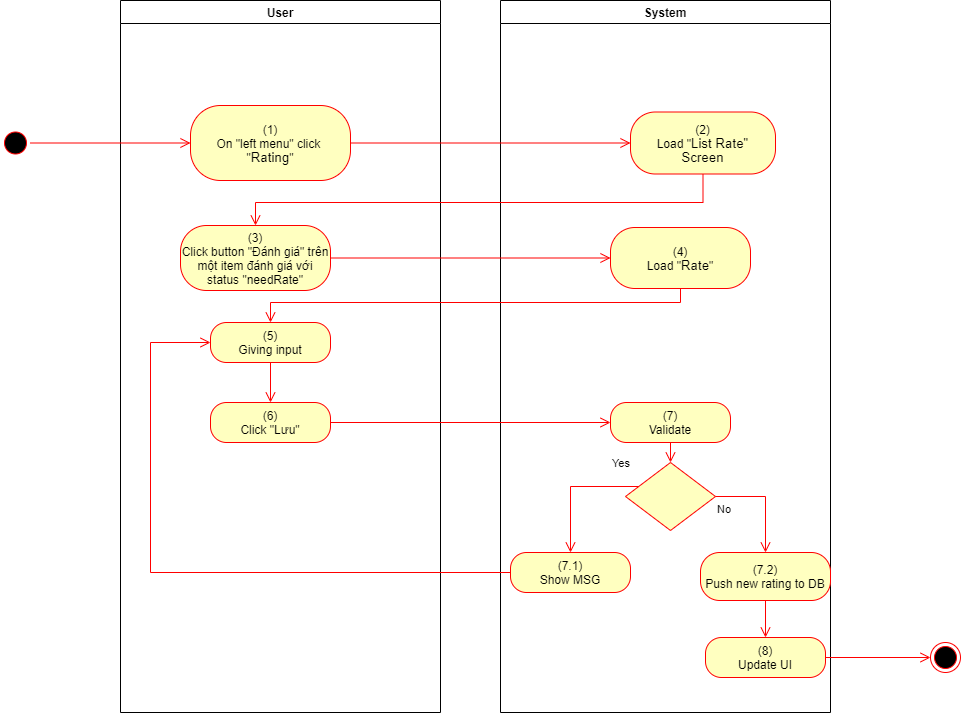
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Product (Seller)” khi người dùng chọn mục “Product Management – List Product” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Product management – List Product (Seller)”, hệ thống thực hiện tải màn hình “List Product(seller)” bao gồm các ngành hàng đã được tạo (listProduct):   * If (Click “Product management – List Product”) => open(“List Product(seller)”) * listProdcut = getListtFromDB(“Selec \* from Product”) * listDisplay = listProduct   Khi người dùng muốn tìm kiếm sản phẩm theo tên hoặc id, thực hiện nhập vào thanh tìm kiếm để tiến hành tìm kiếm.   * OnSearchChange() => keyword = search.text * If (keyword.startWith(“#”)) listDisplay = getListFromDb (“select \* from product where product.id.contains(keyword)”) * Else listDisplay = getListFromDb (“select \* from product where product.name.contains(keyword)) * Khi mà thanh search có sự thay đổi, tiến hành gán keyword bằng với text trong thanh search, nếu keyword bắt đầu bằng “#” thì search theo Id, không thì sẽ search theo tên sản phẩm.   Khi người dùng muốn thực hiện xếp hạng sản phẩm, thực hiện chọn các option trong thanh xếp hạng:   * OnSelectionChange() * if ( selection == “rate”) => SortDescendingByRate(listDisplay) * if ( selection == “Lượt mua”) => SortDescendingByBuyRate (listDisplay) * if ( selection == “Giá”) => SortAscendingByPrice(listDisplay) * xếp hạng theo: "rate", "Lượt mua", "giá" (lúc chưa bấm thì sẽ lấy theo thứ tự của database, khi bấm thì sẽ theo thứ tự như sau:   + Rate: Giảm dần  + Lượt mua: Giảm dần.  + Giá: Tăng dần)   1. **Create**   Để tiến hành thêm hàng hóa, ta thực hiện ấn chọn button “Thêm hàng hóa” trên màn hình “List Product(Seller)”, lúc này form “CRUD Product” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm hàng hóa”) => open(“CRUD Product”) * nameProduct = “” * price = “” * listVideo = NULL * listImage = NULL * idIndustry = “” * idKind = “” * listInnerPropId = NULL * description = “” * Khi click button “Thêm hàng hóa” tiến hành mở “CRUD Product” và set các thông số được nhập là rỗng.   Để tiến hành lưu hàng hóa ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameProduct == “” || (listImage == NULL && listVideo == NULL) || price == “” || idIndustry == “” || idKind == “”) => nhận MSG01. * Else => newProduct = new Product(idProduct, nameProduct, price, listVideo, listImage, updateDate, idIndustry, idKind, listInnerPropId, description) => pushToDB(newProduct), listDisplay.add(newProduct) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Để có thể chọn listInnerPropId, ấn chọn button “Thêm thuộc tính” trong “CRUD Product” để mở bảng chọn “List choose Inner Property” => Ấn chọn “xác nhận” trong “List choose Inner Property” để tiến thêm thuộc tính vào UI (chưa vào DB) => nếu ấn chọn “Back” => thoát mà không có gì được lưu. * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Product (seller)”  1. **Update**   Để tiến hành cập nhật hàng hóa, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item hàng hóa bất kỳ (SelectedProduct) trên màn hình “List Product(Seller)”, lúc này form “CRUD Product” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Product”) * idProduct = selectedProduct.idProduct * nameProduct = selectedProduct.nameProduct * price = selectedProduct.price * listVideo = selectedProduct.listVideo * listImage = selectedProdcut.listImage * idIndustry = selectedProdcut.idIndustry * idKind = selectedProdcut.idKind * listInnerPropId = selectedProduct.listInnerPropId * description = selectedProduct.description. * Khi click button “Chỉnh sửa” tiến hành mở “CRUD Product(Seller)” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành lưu hàng hóa ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameProduct == “” || (listImage == NULL && listVideo == NULL) || price == “” || idIndustry == “” || idKind == “”) => nhận MSG01. * Else => updateProduct = new Product(nameIndus, image, listKind) => pushToDB(updateProduct), listDisplay.update(updateProduct) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Để có thể chọn listInnerPropId, ấn chọn button “Thêm thuộc tính” trong “CRUD Product” để mở bảng chọn “List choose Inner Property” => Ấn chọn “xác nhận” trong “List choose Inner Property” để tiến thêm thuộc tính vào UI (chưa vào DB) => nếu ấn chọn “Back” => thoát mà không có gì được lưu.  1. **Delete**   Để tiến hành xóa ngành hàng, ta thực hiện ấn chọn button “Xóa” của một Item hàng hóa bất kỳ (SelectedProdcut) trên màn hình “List Product(Seller)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * listDisplay.delete(selectedProduct) => deleteOnDB(selectedProduct) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Product (Seller)” |

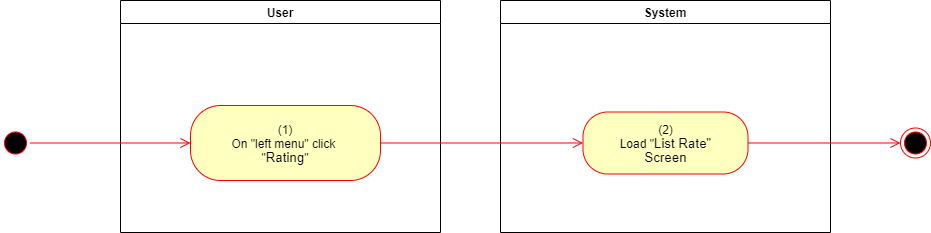
### UC24: Đánh giá

|  |  |
| --- | --- |
| **Name** | **Đánh giá** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc đánh giá |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “Product Management – List rate” |
| **Pre-condition** | Đăng nhập với tư cách khách hàng. |
| **Post-condition** | * CRUD đánh giá theo thao tác người dùng |

#### Activities Flow



*Figure 51: Create Rating Activities Flow*

**

*Figure 52: Read Rating Activities Flow*

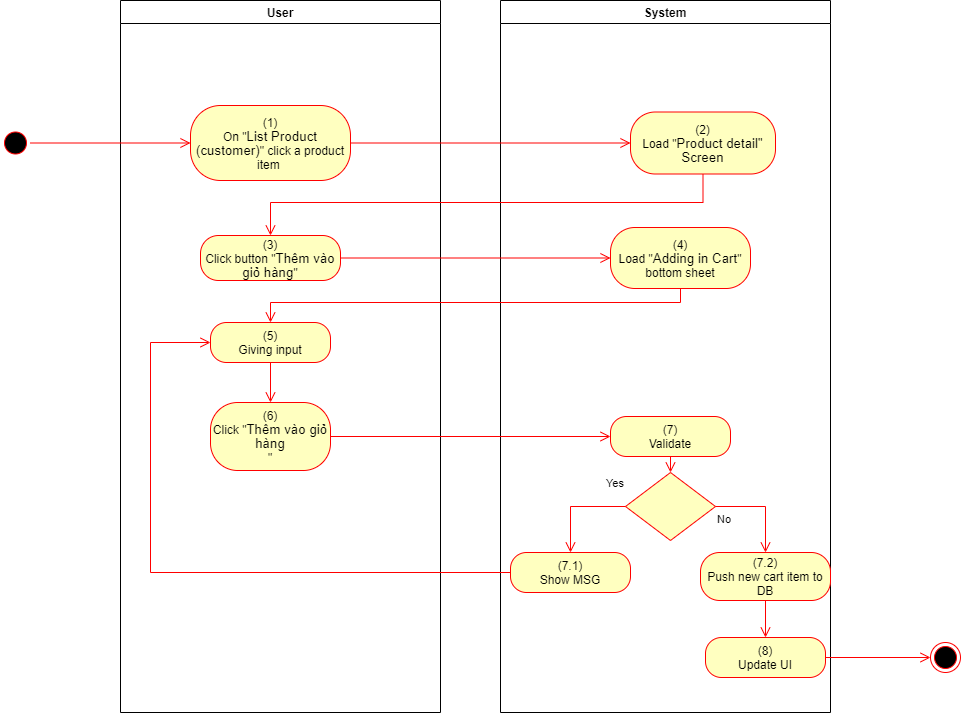
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Rate” khi người dùng chọn mục “Rating” (mục chỉ có ở khách hàng) |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Rating”, hệ thống thực hiện tải màn hình “List rate)” bao gồm các đánh giá hàng hóa đã mua thành công (cả đã và chưa đánh giá) đã được tạo (listProduct):   * If (Click “Rating”) => open(“List Rate(”) * listRate = getListtFromDB(“Selec product from Order, Product (inner join) where Order.status = “SUCCESS” ”) * listDisplay = listRate   Khi người dùng muốn thực hiện lọc các đánh giá, thực hiện chọn các option trong thanh lọc:   * OnSelectionChange() * if ( selection == “needRate”) => listDisplay = getAllNeedRate(listRate) * if ( selection == “Rated”) => getAllRated(listRate) * lọc các status trực tiếp từ listRate tải về trước đó.  1. **Create**   Để tiến hành thêm hàng hóa, ta thực hiện ấn chọn button “Đánh giá” trên một Item đánh giá bất kỳ (selectedItem) ở màn hình “List Rate)”, lúc này form “Rate” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết (Chỉ có thể đánh giá với những sản phẩm chưa đnahs giá bao giờ)   * If (Click “Đánh giá”) * If (selectedItem.isRates = false) => open(“Rate”) * idProduct = selectedItem.idProduct * nameProduct = selectedItemt.nameProduct * price = selectedItem.price * amount = selectedItem.amount * listImage = NULL * listVideo = NULL * rate = NULL * rateText= “” * Khi click button “Thêm hàng hóa” tiến hành mở “CRUD Product” và set các thông số được nhập là rỗng.   Để tiến hành lưu hàng hóa ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (rate== NULL || rateText = “”)> nhận MSG01. * Else => newRate = new Rate(idProduct, nameProduct, price,listImage, listVideo, rate, rateText) => pushToDB(newRate), listDisplay.update(newRate) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Rate”   **\*Khi cần chọn danh sách ảnh ta ấn vào listImage sau đó chọn các ảnh từ Dialog hiện lên và ấn chọn Save để tiến hành lưu hoặc chọn “Back” để thoát và không có gì thay đổi.**   * **If (Click “listImage”) => openInternalImageSavingDialog()** * **If (Click “Save”) => listImage.addListImage ([selectedImageList])** * **Lưu hình các ảnh vừa chọn (selectedImageList) lên database và đưa các ảnh đã chọn lên màn hình “Rate”.** * **If (Click “Back”) = > Back()** * **Quay về màn hình “Rate”**   **Khi cần chọn danh sách video ta ấn vào listVideo sau đó chọn các ảnh từ Dialog hiện lên và ấn chọn Save để tiến hành lưu hoặc chọn “Back” để thoát và không có gì thay đổi.**   * **If (Click “video”) => openInternalVideoSavingDialog()** * **If (Click “Save”) => listVideo.setListVideo ([selectedListVideo])** * **Lưu video đã chọn (selectedVideo) lên database và đưa video đã chọn lên màn hình “Rate”.** * **If (Click “Back”) = > Back()** * **Quay về màn hình “Rate”**   **\*: Áp dụng cho các Use case cần tải ảnh và video (cả list và chọn đơn)** |

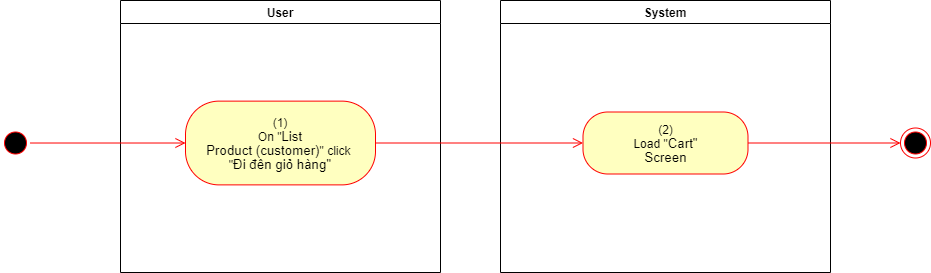
### UC25: Thêm/xóa Giỏ hàng

|  |  |
| --- | --- |
| **Name** | **Thêm/xóa giỏ hàng** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, xóa, xem giỏ hàng hiện tại |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “List Product (Customer)” sau đó ấn chọn một item sản phẩm bất kỳ để mở “Product detail” * Hoặc khi người dùng chọn button “Đi đến giỏ hàng ” trong màn hình “List Product (customer)” hoặc “Product detail” |
| **Pre-condition** | Đăng nhập với tư cách khách hàng. |
| **Post-condition** | * Thêm/xóa theo thao tác người dùng |

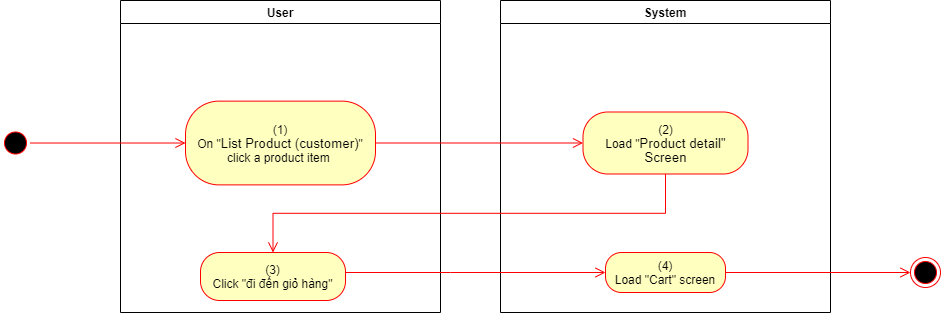
#### Activities Flow



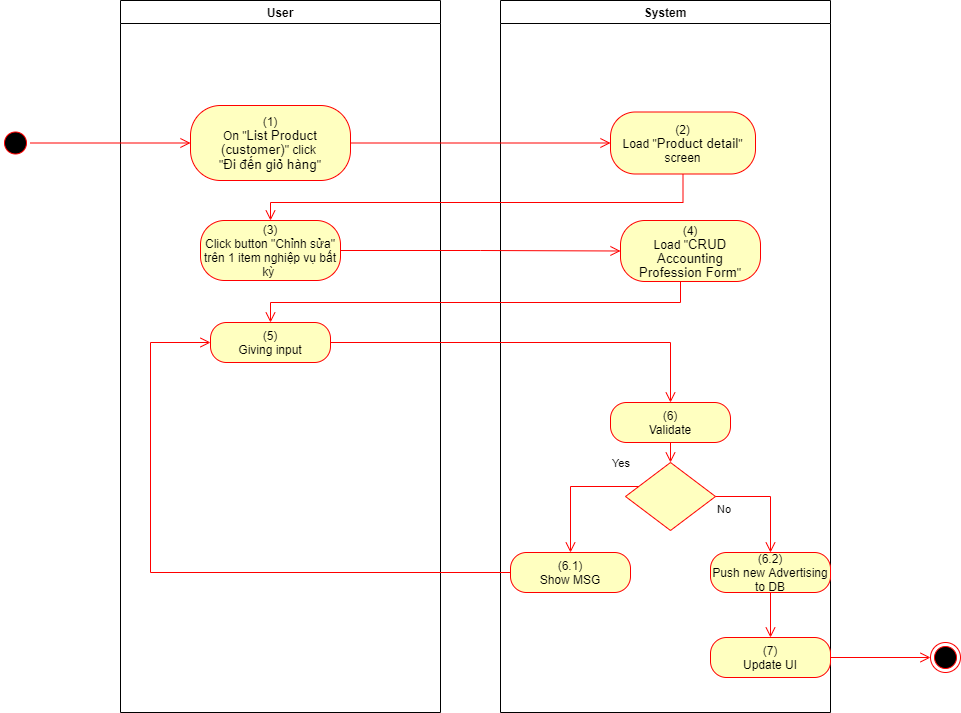
*Figure 53: Add to cart Activities Flow*

**

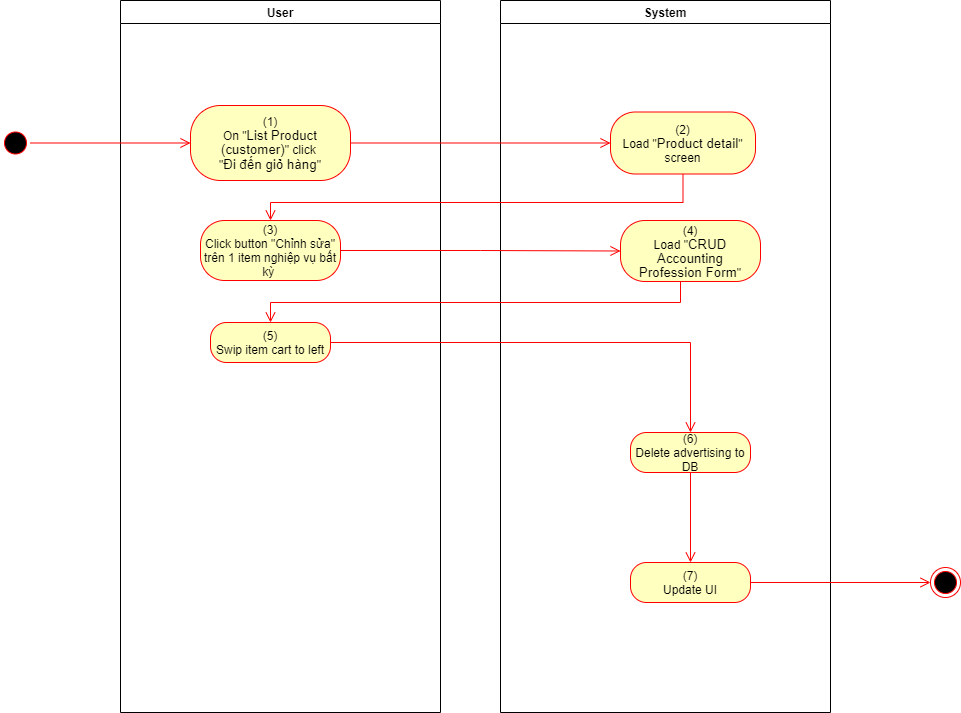
*Figure 54: Read cart Activities Flow (first way)*

**

*Figure 55: Read cart Activities Flow (second way)*

**

*Figure 56: Update to cart Activities Flow*

**

*Figure 57: Delete from cart Activities Flow*

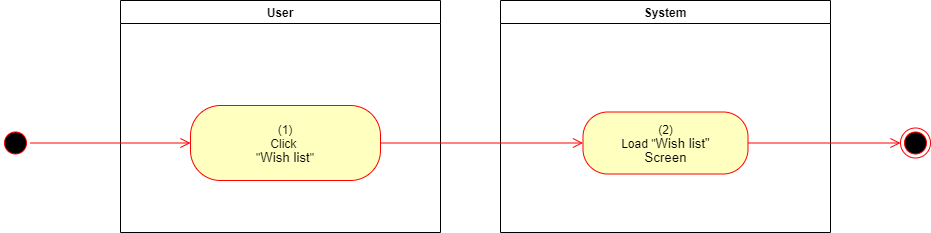
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Product detail (customer)” khi người dùng chọn một item sản phẩm bất kỳ trong “List Product (customer)”. * Hệ thống tải màn hình “Cart” khi ấn vào button “đi đến giỏ hàng” trong “List Product (Customer)” hoặc “Product detail” |
| *(2)* | *BR2* | **Business Rules:**   1. **Xem giỏ hàng**   Để tiến hành xem giỏ hàng, thực hiện một trong các hành động sau đây:   * Ấn chọn button “Đi đến giỏ hàng” ở màn hình “List Product (customer)” để mở xem giỏ hàng. * Ấn chọn một item bất kỳ trong màn hình “List Product (customer)”, lúc này màn hình “Product detail” sẽ xuất hiện, ấn chọn button “Đi đến giỏ hàng” để mở giỏ hàng.  1. **Thêm vào giỏ hàng**   Để tiến hành thêm hàng hóa vào giỏ hàng, ta thực hiện ấn chọn một Item hàng hóa bất kỳ (selectedProduct) ở màn hình “List Product (customer)”, lúc này màn hình “Product detail” sẽ hiển thị cùng các thông tin của selectedProduct, thực hiện chọn các thuộc tính hoặc ấn chọn “Thêm vào giỏ hàng” để mở Bottom sheet “Adding in Cart”.   * If (hasChoseProperty()) => open(“Adding in cart”) * “Adding in cart”.listPropSelectedOpt = “Detail Product”.listPropSelectedOtp * Adding in cart”.listPropSelection = selectedProduct.listPropSelection * “Adding in cart”.amount = 1. * “Adding in cart”.idProduct = selectedProduct.idProduct * “Adding in cart”.nameProduct = selectedProduct.nameProduct * “Adding in cart”.image = selectedProduct.image * “Adding” in cart”.price = selectedProduct.price * Nếu đã chọn các thuộc tính ở màn hình “Detail Product” rồi thì khi mở “Adding in cart” sẽ được giữ nguyên lựa chọn. (listPropselectedOtp) * Khi mở màn hình “Adding in cart” một số thông tin sản phẩm sẽ được truyền qua. (listPropSelection, id, name, price, image) * **listPropSelection là danh sách cần lựa chọn, listPropSelectedOtp là danh sách các lựa chọn đã chọn.** * Else => open("Adding in cart”) * “Adding in cart”.listPropSelectedOtp = NULL * “Adding in cart”.listPropSelection = selectedProduct.listPropSelection * “Adding in cart”.amount = 1 * “Adding in cart”.idProduct = selectedProduct.idProduct * “Adding in cart”.nameProduct = selectedProduct.nameProduct * “Adding in cart”.image = selectedProduct.image * “Adding” in cart”.price = selectedProduct.price * Nếu chưa chọn thì thuộc tính nội sản phẩm sẽ được giữ nguyên là chưa chọn * Khi mở màn hình “Adding in cart” một số thông tin sản phẩm sẽ được truyền qua. (listPropSelection, id, name, price, image)   Để tăng/giảm số lượng lúc thêm vào giỏ hàng thì tại lúc đang mở màn hình “Adding in cart” ta ấn chọn button “tăng số lượng”/”giảm số lượng”.   * If (Click “tăng số lượng”) => “Adding in cart”.amount ++ * If (Click “giảm số lượng”) => “Adding in cart”.amount - - * If (amount == 1) => [Do nothing] * Khi ấn chọn “tăng số lượng” thì sẽ tăng amount, khi click “giảm số lượng” sẽ giảm amount, khi giảm còn 1 thì sẽ không còn giảm được nữa.   Để tiến hành gửi đơn ta ấn chọn “Thêm vào giỏ hàng” trên màn hình “Adding in cart” để hệ thống kiểm tra thông tin để gửi vào giỏ hàng.   * If (Click “Thêm vào giỏ hàng”) * If (listPropSelectedOtp.numberOfProp != listPropSelection.numberOfProp) => nhận MSG21 * If (amount == 0) => nhận MSG22 * If (getAmountRemain < amount) => nhận MSG23 * If (getAmoutRemain <= 0) => nhận MSG24 * Else => newCartItem = new CartItem(listPropSelectedOtp, amount, idProduct, nameProduct, image, price) => pushToDB(newCartItem) * Hệ thống kiểm tra số lượng sản phẩm đặt vào giỏ hàng và sản phẩm còn lại của cửa hàng, kiểm tra đã chọn tất cả thuộc tính hay chưa.  1. **Chỉnh sửa trong giỏ hàng**   Để tiến hành chỉnh sửa giỏ hàng, ta thực hiện ấn chọn button “Đi đến giỏ hàng” trong màn hình “List Product (customer)” hoặc trong màn hình “Product detail”, lúc này màn hình “Cart” hiển thị các item giỏ hàng đã thêm vào:   * If (Click “đi đến giỏ hàng”) => listCart = getListFromDB(“Select \* from Cart with Cart.customerId = thisAccount.id”) * listDisplay = listCart   Nếu muốn thay đổi số lượng ta ấn chọn vào button “Thêm số lượng”/”Giảm số lượng” để thực hiện tăng giảm số lượng của 1 item giỏ hàng.   * If (Click “Tăng số lượng”) * If (amount == getNumberFromDB(select amount from Product where selectedProduct.id = product.id)) => [Do nothing] => nhận MSG 25 * Else => amount++ => updateToDB(amount) * Không thể tăng thêm số lượng khi đã đạt số lượng còn lại của sản phẩm đó * If (Click “Giảm số lượng”) * If (amount == 0) => [Do nothing] => Nhận MSG23 * If (oldAmount.Exceed(remainAmount)) => amount = remainAmount. => updateToDB(amount) * Else amount - - => updateToDB (amount) * Nếu số lượng bằng 0 thì không thể tiếp tục giảm, nếu số lượng cũ đã vượt số lượng còn lại tồn kho thì khi ấn “giảm số lượng” số lượng mới sẽ bằng số lượng còn lại tồn kho, nếu không thì trừ 1 như bình thường. * Mỗi lần tăng giảm đề sẽ thay đổi ở DataBase   **Nếu muốn xóa 1 Item giỏ hàng thì chỉ cần kéo item về bên trái => deleteItem([movingLeftItem]) => deleteOnDB([MovingLeftItem]).** |

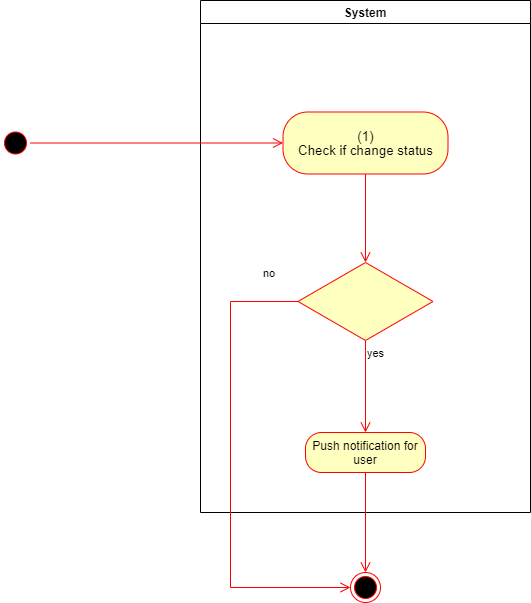
### UC26: Thêm/xóa wish list

|  |  |
| --- | --- |
| **Name** | **Thêm/xóa wish list** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc wish list |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “List Product (Customer)” * Hoặc khi người dùng vào mục “Wish list” |
| **Pre-condition** | Đăng nhập với tư cách khách hàng. |
| **Post-condition** | * Thêm xóa wishlist theo thao tác người dùng |

#### Activities Flow



*Figure 58: Add/delete from wishlist Activities Flow*

**

*Figure 59: Auto Notify from wishlist Activities Flow*

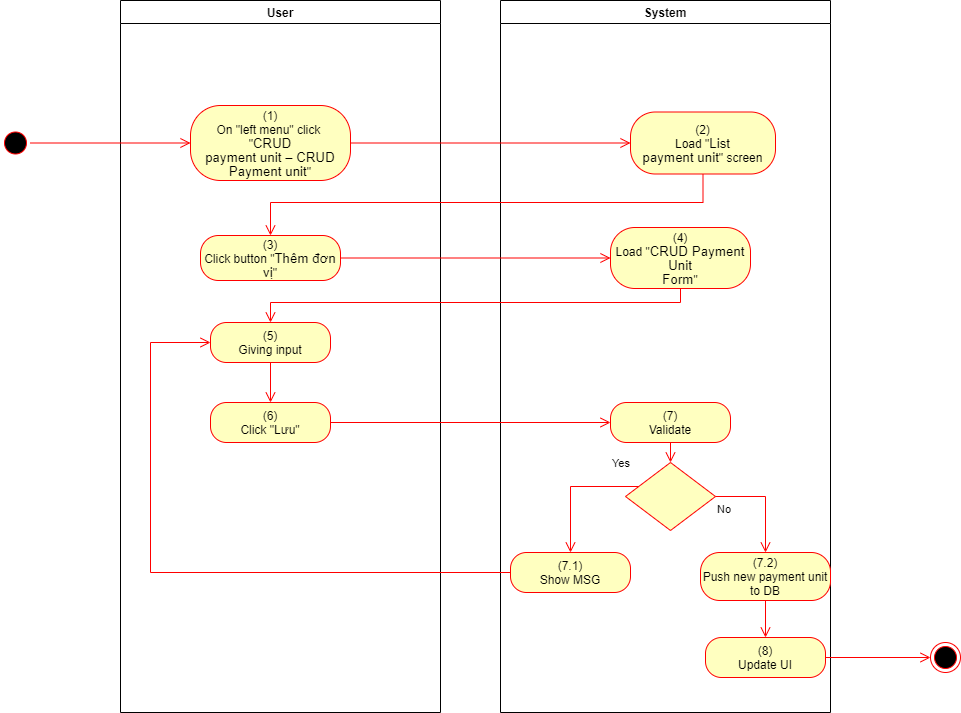
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Product (Customer)” khi người dùng đăng nhập thành công. * Hệ thống tải màn hình “Wishlist” khi người dùng ấn chọn mục “Wish list” |
| *(2)* | *BR2* | **Business Rules:**   1. **Xem wish list**   Khi người dùng muốn xem toàn bộ wishlist của bản thân, thực hiện ấn chọn vào mục “Wishlist” để mở màn hình “Wish list”, lúc này hệ thống sẽ tải toàn bộ sản phẩm mà bản thân đã thêm vào wish list.   * wishList = getListFromDB(“Select product from product inner join wishlist where product.id = wishlist.product and wishlist.idCustomer = thisAccount.id ”)   Để tiến hành lọc wish list, người dùng chọn các lựa chọn trong Drop Box để tiến hành lọc các wishlist theo:   * “Available” : các sản phẩm đang có thể thực hiện mua. * Select product from product inner join wishlist where product.id = wishlist.product and wishlist.idCustomer = thisAccount.id and product.remain > 0 * “unAvailable”: các sản phẩm đang hết hàng. * Select product from product inner join wishlist where product.id = wishlist.product and wishlist.idCustomer = thisAccount.id and product.remain <= 0  1. **Thêm và xóa khỏi wish list:**   Để tiến hành thêm hàng hóa và trong wish list, tại các màn hình có Item sản phẩm (SelectedProduct) như “List Product” hoặc trong “Product detail” ta ấn chọn nút “Thêm vào wishlist” để thêm/xóa sản phẩm khỏi wish list:   * If (Click “Thêm vào wishlist”) * If (Product.isInWishList) => getOutWishList(selectedProduct) * Else => getInWishList(SelectedProduct) * Nếu sản phẩm đang trong wishlist => ấn để đưa ra khỏi wish list * Nếu sản phẩm không trong wishlist => ấn để đưa vào trong wish list.  1. **Thông báo khi chuyển trạng thái**   Nếu sản phẩm đặt trong wishlist chuyển từ trạng thái “Unavailable” sang “Available”, hệ thống sẽ gửi thông báo đến cho người dùng.   * If (ChangeStatus([From “Unavailable” To “Available”])) => makeNotification()   Định dạng thông báo   |  |  | | --- | --- | | [nameCompany] | [Time] | | Sản phẩm [nameProduct] trong wishlist của bạn hiện tại đã sẵn sàng để mua ! | | |

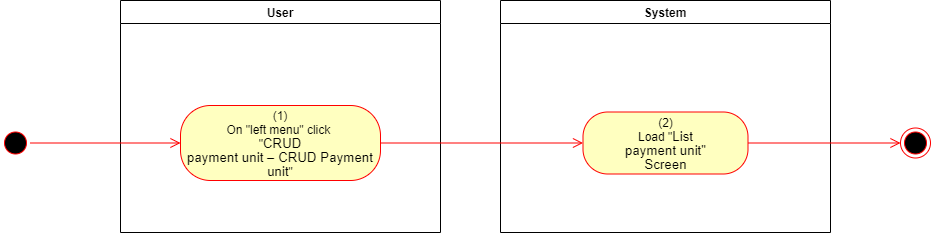
### UC27: CRUD đơn vị thanh toán

|  |  |
| --- | --- |
| **Name** | **CRUD đơn vị thanh toán** |
| **Description** | Chức năng cho phép người dùng CRUD đơn vị thanh toán |
| **Actor** | 1. User |
| **Trigger** | * Khi người dùng ấn chọn mục “CRUD Payment unit – CRUD payment unit” trong “left menu” |
| **Pre-condition** | Đăng nhập với tài khoản của BGD |
| **Post-condition** | * Thực thi CRUD được đơn vị thanh toán theo thao tác người dùng. |

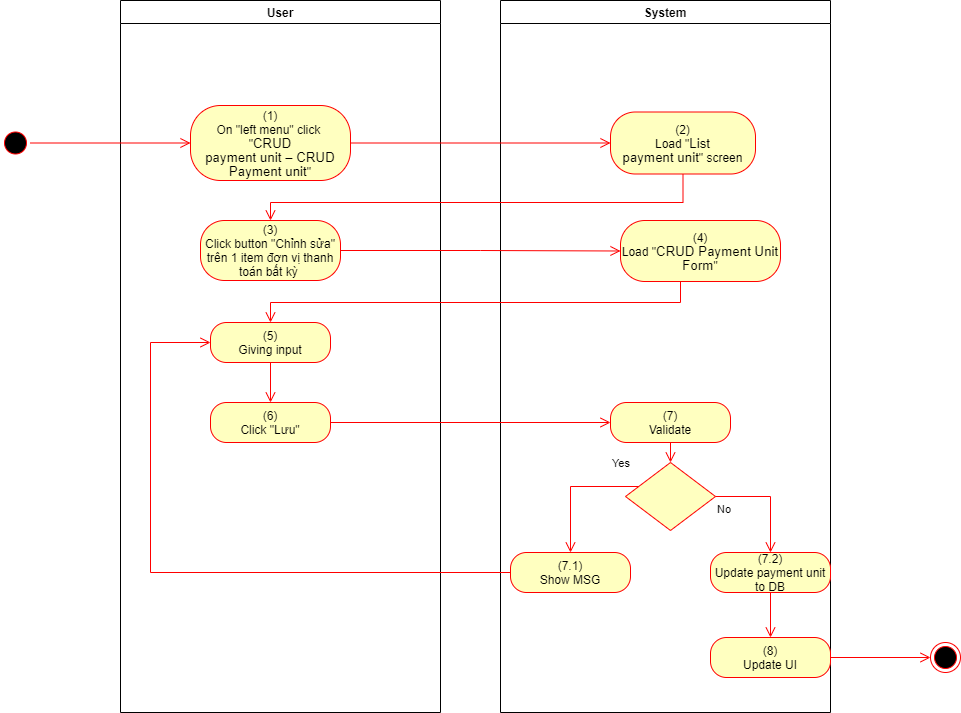
#### Activities Flow



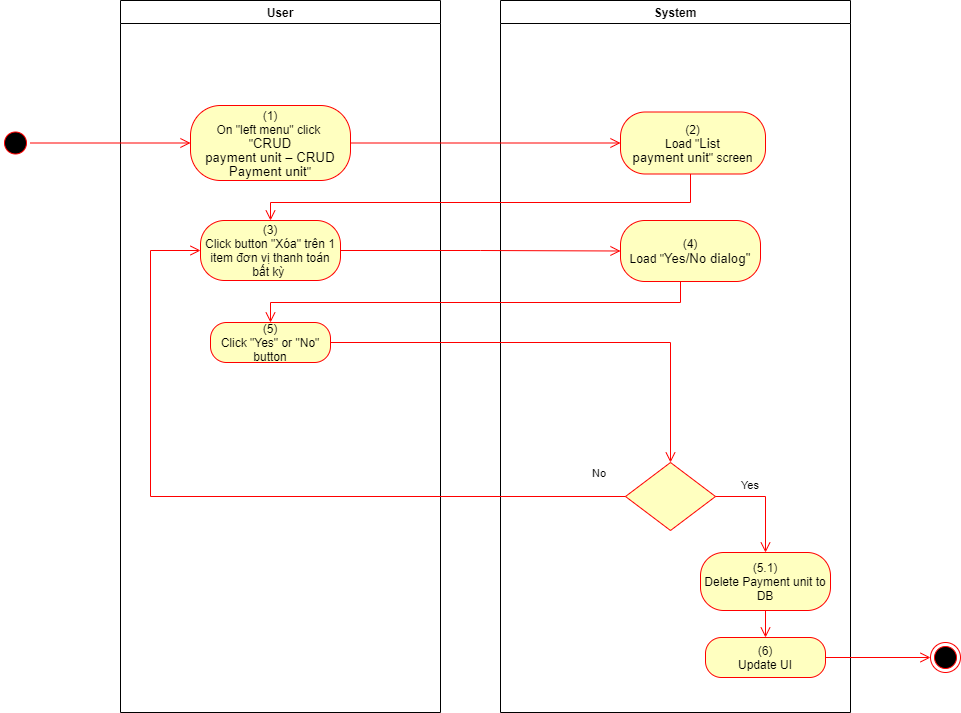
*Figure 60: Create Payment Unit Activities Flow*

**

*Figure 61: Read Payment Unit Activities Flow*

**

*Figure 62: Update Payment Unit Activities Flow*

**

*Figure 63: Delete Payment Unit Activities Flow*

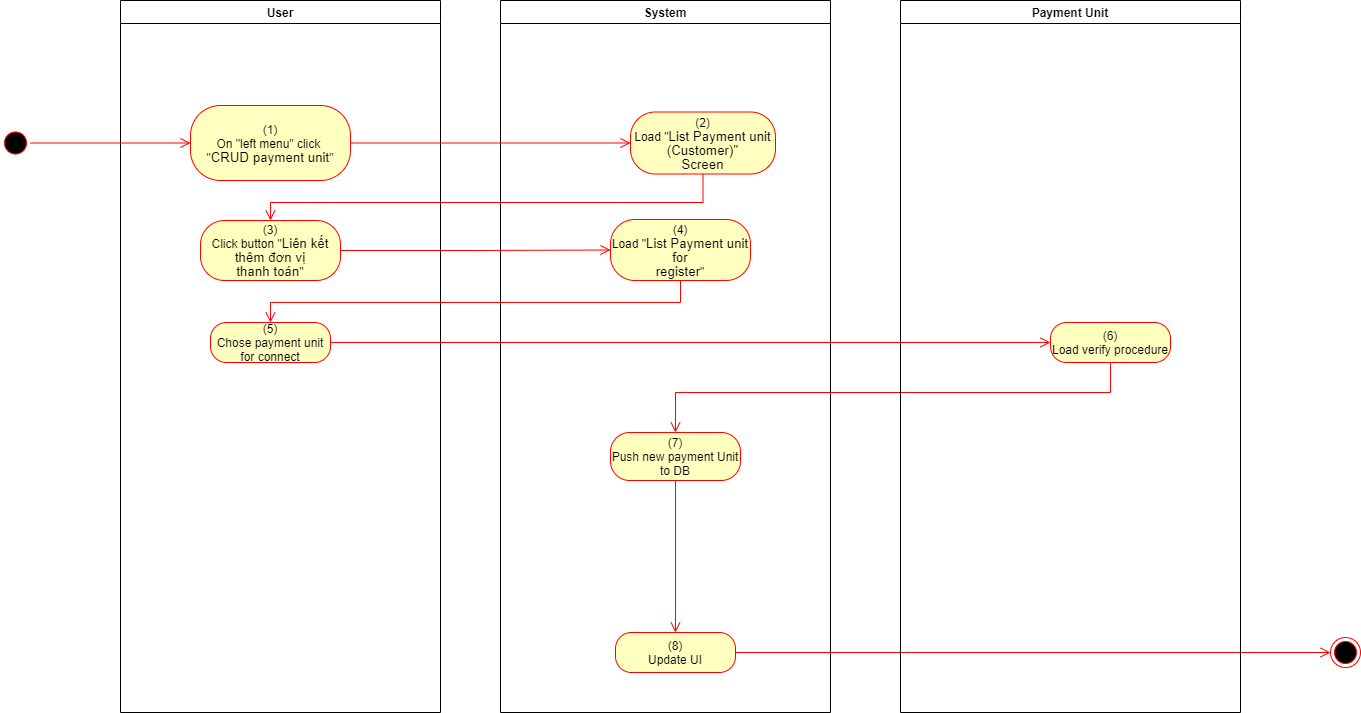
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Khi người dùng ấn chọn vào mục “CRUD payment unit – CRUD Payment unit” màn hình “List payment unit” sẽ được tải lên. |
| *(2)* | *BR2* | **Business Rules:**   1. **Read:**   Khi người dùng ấn chọn vào mục “CRUD payment unit – CRUD Payment unit” màn hình sẽ hiển thị màn hình “List payment unit” bao gồm nhiều paymentItem.  paymentItem sẽ chứa toàn bộ thông tin về một đơn vị vận chuyển, nhưng màn hình “List payment unit” sẽ hiển thị nhiều deliveryItem với các thông tin cơ bản như trong mô tả ở **View Description** :   * Khi thực hiện tải màn hình “List payment unit” sẽ thực hiện tải list từ database về: * listPaymentUnit = loadListFromDb("Select \* from PaymentUnit”).  1. **Create:**   Để thực hiện việt tạo thêm đơn vị vận chuyển, thực hiện ấn vào button “Thêm đơn vị” để hệ thống thực hiện mở “CRUD Payment Unit Form”. Tiến hành nhập các thông tin mà form yêu cầu, trong đó, các API sẽ được nhập theo một trong 2 định dạng sau:   |  |  |  |  | | --- | --- | --- | --- | | 1 | https:// | [Domain name].[top level domain] | /[endParh] | | 2 | Localhost:: | [portNumber] | /[endPath] | | Note: [endPath] nếu cần truyền pathvariable thì thực hiện như lúc khai báo code API => /{variableName}   * statusAPI: cần truyền {OrderID} để kiểm tra * checkAPi: cần truyền {OrderID} để kiểm tra | | | |  * Việc kiểm tra format sẽ được hàm checkFormatAPI(String api) thực hiện. (tự xây dựng)   Khi nhấn chọn button “Thêm” thì hệ thống sẽ thực hiện kiểm tra định dạng các thông tin và gửi các MSG tương ứng hoặc đưa đơn vị vận chuyển mới lên Database:   * If (Click “Thêm”) => ValidateInput() * ValidateInput: * If (namePayment == “” || account == “” || password = “” || statusAPI == “” || priceAPI == “”|| checkAPI == “” || connectAPI) => nhận MSG01. * If (account.contains(“ ”)|| password.contains(“ ”)) => nhận MSG04. * If(checkFormatAPI(statusAPI) || checkFormatAPI(priceAPI) || checkFormatAPI(connectAPI) || checkFormatAPI(checkAPI)) =>Nhận MSG04 * If (ValidateInput()) => SendToDB(new PaymentItem(namePayment, account, password, statusAPI, checkAPI, connectAPi, updateDate=DateTime.Now())). * Đóng gói dữ liệu và gửi dữ liệu lên database.   Nếu người dúng ấn chọn button “Thoát” thì dù thông tin có sai hay không thì hệ thộng sẽ không lưu lại mà quay lại màn hình “List Payment Unit”:   * If(Click “Thoát”) => Back().  1. **Update:**   Để thực hiện việc thay đổi, cập nhật một đơn vị thanh toán, thực hiện ấn vào button “Chỉnh sửa” trên mỗi item của listPaymentUnit trong màn hình “List Payment Unit” để thực hiện chỉnh sửa cho item đó. Khi ấn chọn button “Chỉnh sửa” hệ thống sẽ hiển thị “Update Payment Unit Form”. Người dùng tiến hành thực hiện các chỉnh sửa (Form có chưa sẵn các thông tin của item đã chọn).   * namePayment = paymentItem.namePayment * account = paymentItem.account * pasword = paymentItem.pasword * statusAPI = paymentItem.statusAPI * checkAPI = paymentItem.checkAPI * priceAPI = paymentItem.priceAPI * deleteAPI = paymentItem.deleteAPI * connectAPI = paymentItem.connectAPI * Thực hiện gán để các thông tin hiển thị trên Form.   Click chọn button “Lưu” để kiểm tra định dạng và tải dữ liệu mới lên database.   * If (Click “Lưu”) => ValidateInput() * ValidateInput: * If (namePayment == “” || account == “” || password = “” || statusAPI == “” || checkAPI == “”|| connectAPI == “” || priceAPI == “”) => nhận MSG01. * If (account.contains(“ ”)|| password.contains(“ ”)) => nhận MSG04. * If(checkFormatAPI(statusAPI) || checkFormatAPI(priceAPI)|| checkFormatAPI(connectAPI) || checkFormatAPI(checkAPI)) =>Nhận MSG04 * If (ValidateInput()) => updateToDB(paymentItem.ID ,namePayment, account, password, statusAPI, checkAPI, connectAPi, updateDate=DateTime.Now)). * Đóng gói dữ liệu và gửi dữ liệu lên database.   Nếu người dúng ấn chọn button “Thoát” thì dù thông tin có sai hay không thì hệ thộng sẽ không lưu lại mà quay lại màn hình “List Payment Unit”:   * If(Click “Thoát”) => Back().  1. **Delete:**   Để thực hiện việc xóa một đơn vị vận chuyển, thực hiện ấn vào button “Xóa” trên mỗi item của listDeliveryUnit trong màn hình “List Payment Unit” để thực hiện xóa item đó. Khi ấn chọn item “Xóa” hệ thống sẽ mở hộp thoại Yes/No để thự hiện chọn lựa có chắc chắn xóa hay không.   * If (Click “Yes”) => listPaymentUnit.deleteInListAndDB(paymentItem.ID) => updateUI(). * Khi click button “Yes” sẽ thực hiện xóa item ra khỏi list và database sau đó cập nhật lại giao diện. * If (Click “No”) => Back() * Nếu người dùng ấn chọn button “No” thì sẽ thực hiện tắt Yes/No dialog và quay lại màn hình “List Payment Unit”. |

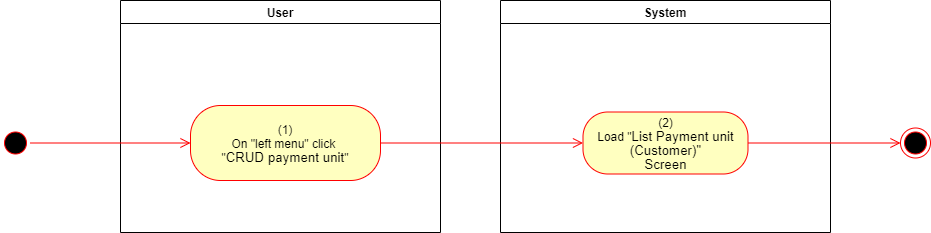
### UC28: CRUD Đơn vị thanh toán (khách hàng)

|  |  |
| --- | --- |
| **Name** | **CRUD đơn vị thanh toán (khách hàng)** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc đơn vị thanh toán |
| **Actor** | 1. User 2. System 3. Payment unit |
| **Trigger** | * Khi người dùng mở mục “CRUD payment unit“ |
| **Pre-condition** | Đăng nhập với tư cách khách hàng |
| **Post-condition** | * CRUD đơn vị thanh toán theo yêu cầu |

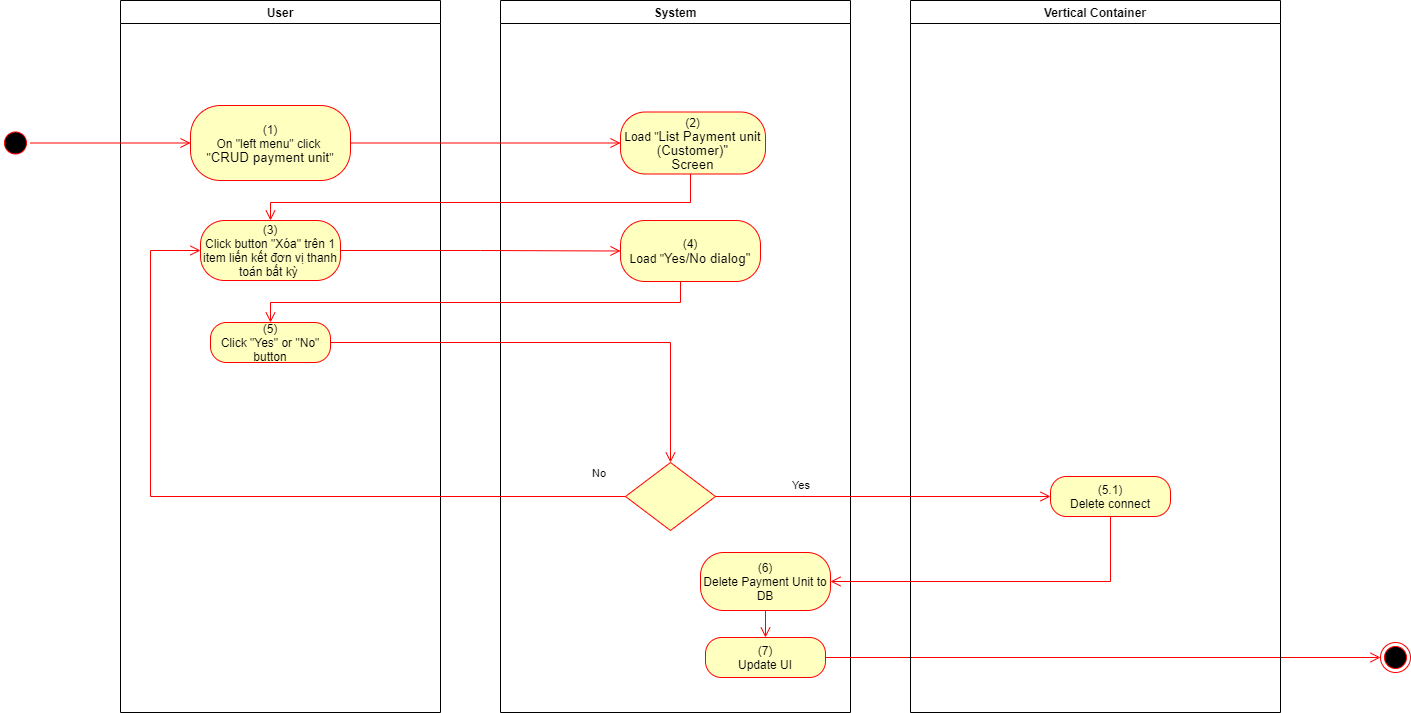
#### Activities Flow



*Figure 64: Create connect to payment unit Activities Flow*

**

*Figure 65: Read connect to payment unit Activities Flow*

**

*Figure 66: Delete connect to payment unit Activities Flow*

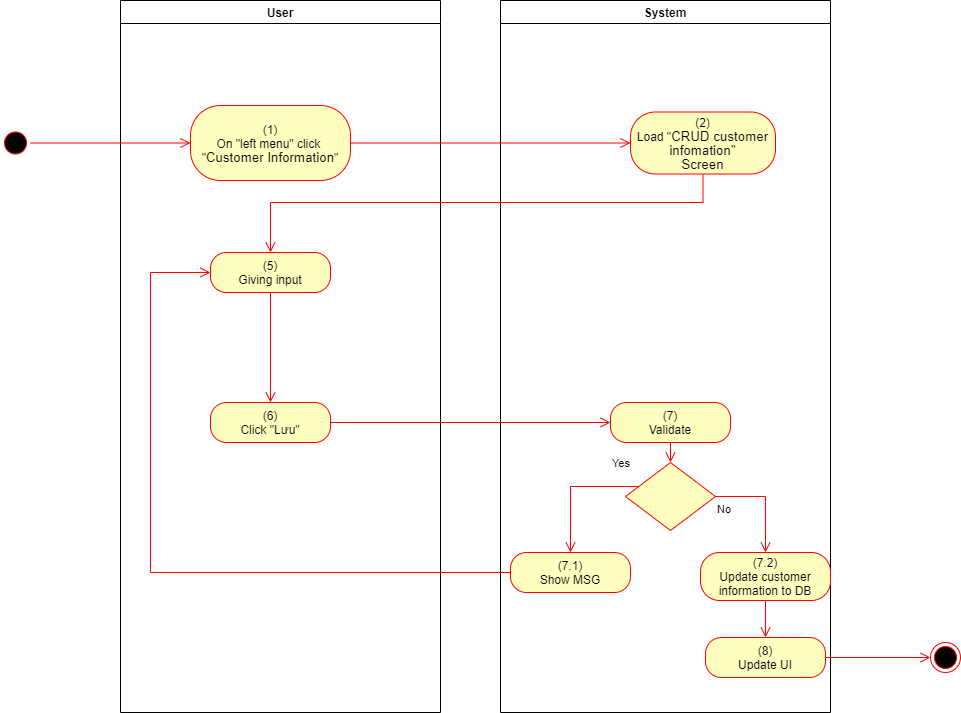
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Payment Unit (Customer)” khi người dùng chọn mục “CRUD payment unit” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “CRUD payment unit”, hệ thống thực hiện tải màn hình “List Payment unit (Customer)” bao gồm các ngành hàng đã được tạo (listProduct):   * If (Click “CRUD payment unit”) => open(“List Payment unit (Customer)” * listPayment = getListtFromDB(“Select \* from Customer\_Payment inner join customer where customer\_payment.custome.ID = customerID and customerIID = thisAccount.id ”) * listDisplay = listPayment * Hệ thống tải các liên kết thanh toán của người dùng đăng nhập hiện tại và hiển thị lên màn hình.  1. **Create**   Để tiến hành liên kết thêm đơn vị thanh toán, ta thực hiện ấn chọn button “Liên kết thêm đơn vị thanh toán” trên màn hình “List Payment Unit (Customer)”, lúc này form “List Payment unit for register” sẽ hiển thị và người dùng tiến hành chọn đơn vị cần thiết liên kết.   * If (Click “Liên kết thêm đơn vị thanh toán”) => open(“List Payment unit for register”) * listForRegis = getListFromDB(Select \* from Payment\_unit) * Hệ thống tải các liên kết thanh toán mà BGD app đã tạo và hiển thị lên màn hình để user chọn lựa.   Khi đã chọn đơn vị thanh toán cần liên kết người dùng tiến hành thực hiện theo những gì đơn vị đó cần:   * If (Click “[đơn vị thanh toán]”) => open(“connectAPI”) * getFromAPI (STK) * listPayment.add (new Payment(tên đơn vị , id đơn vị, STK)) * listDisplay.add (newPayment) * Khi click chọn một đơn vị thanh toán bất kỳ, sẽ tiến hành thực hiện theo những gì connectAPI bên đối tác đưa yêu cầu, sau đó hệ thống lấy về STK để tạo class Payment lưu lại cho user và cập nhật lại UI. * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Payment Unit (customer)”  1. **Delete**   Để tiến hành xóa một liên kết thanh toán, ta thực hiện ấn chọn button “Xóa” của một Item đơn vị thanh toán bất kỳ (SelectedPayment) trên màn hình “List Payment Unit (customer)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * DeleteOnPaymentUnitSide(deleteAPI + /{thisAccount.id}) * listDisplay.delete(selectedPaymentt) => deleteOnDB(selectedPayment) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Payment Unit (customer) |

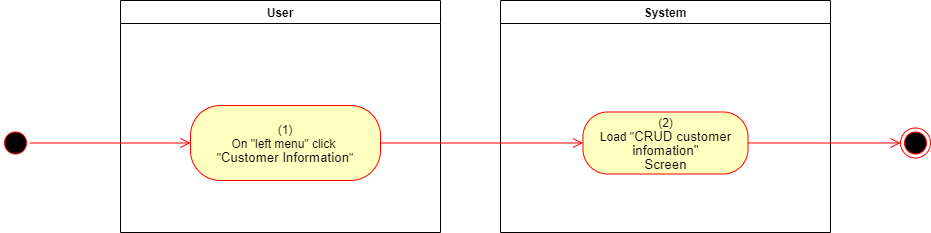
### UC29: CRUD thông tin cá nhân (Khách hàng)

|  |  |
| --- | --- |
| **Name** | **CRUD thông tin cá nhân (khách hàng)** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc thông tin cá nhân |
| **Actor** | 1. User |
| **Trigger** | * Khi người dùng mở mục “CRUD Customer information“ |
| **Pre-condition** | Đăng nhập với tư cách khách hàng |
| **Post-condition** | * CRUD thông tin cá nhân của khách hàng theo thao tác khách hàng |

#### Activities Flow



*Figure 67: Create-update customer information Activities Flow*

**

*Figure 68: Read customer information Activities Flow*

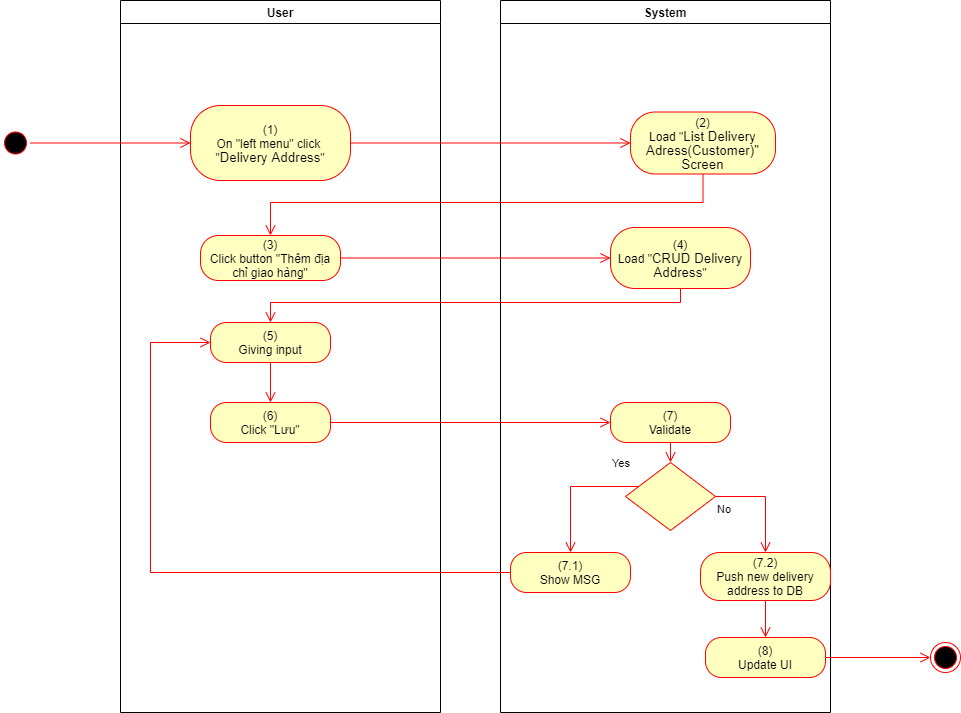
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “CRUD customer information khi người dùng chọn mục “Customer Information” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Customer Information”, hệ thống thực hiện tải màn hình “CRUD customer information” bao gồm các thông tin khách hàng nhập vào lúc đăng ký (thông tin về giới tính và địa chỉ sẽ được khách hàng cập nhật sau, nếu chưa thì sẽ là NULL):   * If (Click “Customer Information”) => open(“CRUD Customer Information”) * customer = thisAccount (biến customer sẽ chứ các thông tin và hiển thị lên màn hình) * customerID = customer.id * customerName = customer.name * gender = customer.gender * phoneNumber = customer.phoneNumber * residentID = customer.residentID * email = customer.email * province = customer.province * district = customer.district * ward = customer.ward * lane = customer.addressLane * Hệ thống tải các thông của người dùng đăng nhập hiện tại và hiển thị lên màn hình.  1. **Create**   Thông tin của người dùng đã được khởi tạo từ lúc đăng ký, tuy nhiên trường giới tính và địa chỉ sẽ trống, để thực hiện bổ sung, thực hiện vào màn hình “CRUD Customer Information” bằng cách nhấn chọn mục “Customer Infomaton” sau đó thực hiện các yêu cầu nhập:   * If (Click “Customer Infomaton”) => open(“CRUD Customer Infomaton”) * customer = thisAccount (biến customer sẽ chứ các thông tin và hiển thị lên màn hình) * customerID = customer.id * customerName = customer.name * gender = NULL * phoneNumber = customer.phoneNumber * residentID = customer.residentID * email = customer.email * province = NULL * district = NULL * ward = NULL * lane = “” * Hệ thống tải các liên kết thanh toán mà BGD app đã tạo và hiển thị lên màn hình để user chọn lựa.   Khi đã chọn đơn vị thanh toán cần liên kết người dùng tiến hành thực hiện theo những gì đơn vị đó cần:   * If (Click “lưu”) * If (customerName = “” || gender = NULL || phoneNumber = “” || province == NULL || district == NULL || ward == NULL) => nhận MSG01 * If (customerName == “” ) => MSG01 * If (customerName.contains([number] || customerName.contains([SpecialChracter])) => MSG02 * If (phoneNumber == “”) => MSG01 * If (phoneNumber.contains([nonNumber]) => MSG03 * If (PhoneNumberOf.length > 10 || phoneNumber < 10 || !PhoneNumber.startWith([validNumber])) => MSG4 * If (residentID == “”) => MSG01 * If (residentID.contains([nonNumber]) => MSG03 * If (residentID.length > 10 || PhoneNumber < 10) => MSG4 * If (email == “”) => MSG01 * If (email.checkRightEmailFormat()) => MSG04 * If (email.isExist) => MSG05 * customer.update(customerName, gender, phoneNumber, gender, province, district, ward, lane, email, residentID) * Khi ấn chọn “Lưu” hệ thống sẽ kiểm tra định dạng các thông tin nếu không nhận được bất kỳ MSG nào thì sẽ đưa lên database và cập nhật lại UI. * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Payment Unit (customer)”   Khi cần tạo hoặc thêm hoặc xóa địa chỉ, thực hiện nhập/chọn cho các thông tin ở các ô “Tỉnh”, “Huyện”, “Xã”, “Đường”.   * listDistrict = GetAllDistrictFromDB("Select \* from District”) * dbDistrict.isEnable = false (Drop box district) * dbWard.isEnable = false (Drop box ward) * tbLane.isEnable = false (text box Lane) * Ban đầu sẽ tự động tải các tỉnh, các DropBox của “Huyện” và “Xã” và textbox “Đường” sẽ bị “vô hiệu hóa” * If(Chose “Tỉnh”) => dbDistrict.isEnable = true => getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “huyện” và tải tất cả các huyện theo ID của tỉnh đã chọn. * If(Chose “Huyện”) => db.Ward.isEnable = true => getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “Xã” và tải tất cả các huyện theo ID của tỉnh đã chọn.   Khi thay đổi chọn một “tỉnh”, “huyện”, “xã” bất kỳ, thì các cấp bậc dưới sẽ được “vô hiệu hóa” và xóa thông tin tương ứng, thay đổi lại danh sách lựa chọn tương ứng.   * If (change “Tỉnh”) * dbDistrict.isEnable = true, dbDistrict.SelectedItem = NULL,dbDistrict.ListDistrict=getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * dbWard.isEnable = false, dbWard.selectedItem = NULL, dbWard.listWard = [Empty list]. * tbLane.text = “”, tbLane.isEnable = false. * If (change “Huyện”) * dbWard.isEnable = true, dbWard.SelectedItem = NULL,dbWard.ListDistrict=getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * tbLane.text = “”, tbLane.isEnable = false. * If (change “Xã”) * tbLane.text = “”, tbLane.isEnable = false |

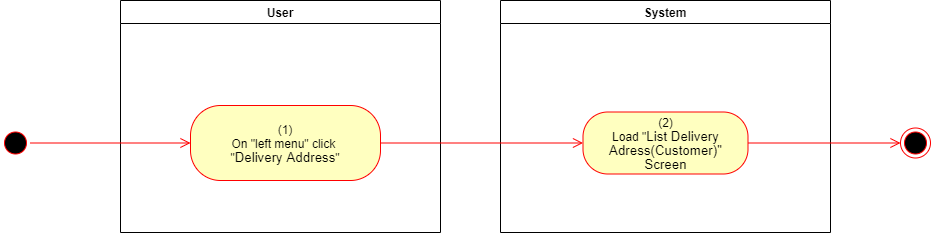
### UC30: CRUD địa chỉ giao hàng (khách hàng)

|  |  |
| --- | --- |
| **Name** | **CRUD địa chỉ giao hàng (khách hàng)** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc địa chỉ giao hà |
| **Actor** | 1. Khách hàng |
| **Trigger** | * Khi người dùng mở mục “Delivery address“ |
| **Pre-condition** | Đăng nhập với tư cách khách hàng |
| **Post-condition** | * CRUD địa chỉ giao hàng theo yêu cầu |

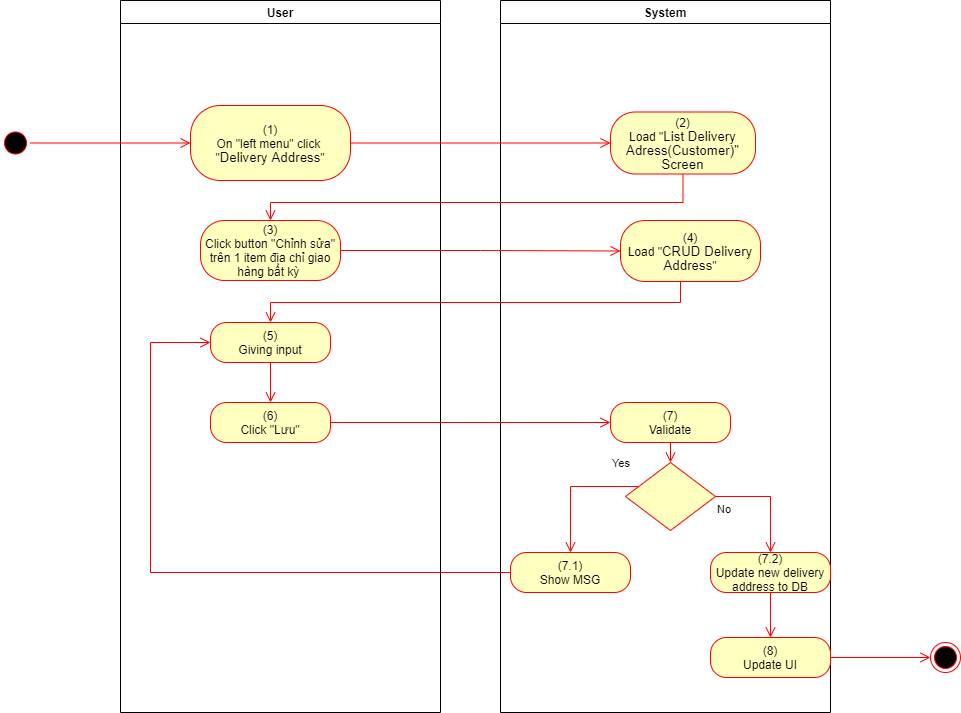
#### Activities Flow



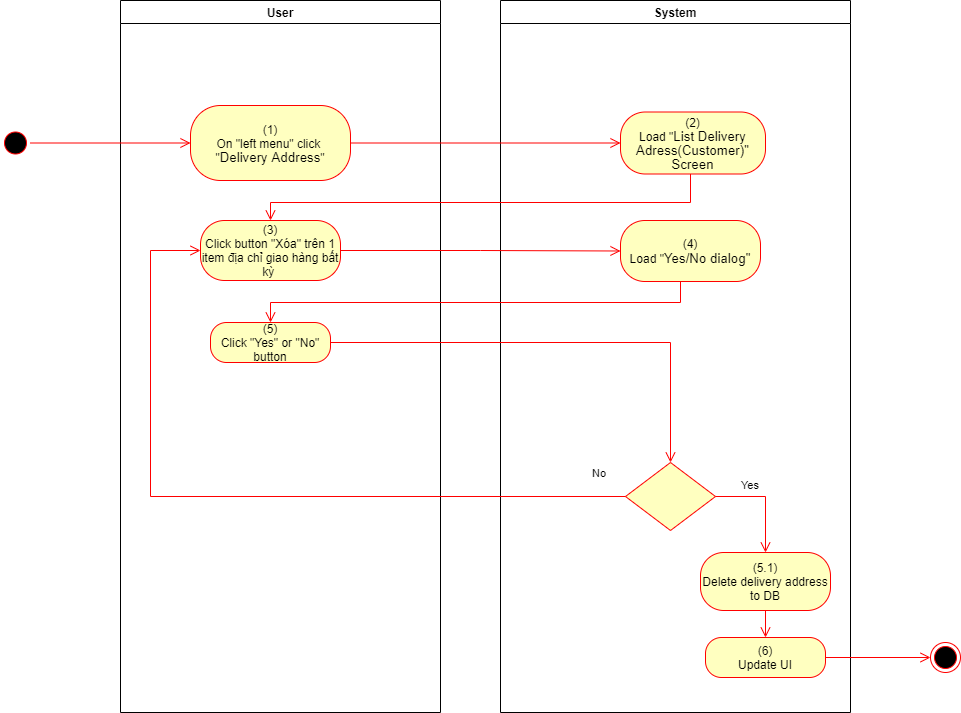
*Figure 69:Create delivery address Activities Flow*

**

*Figure 70:Read delivery address Activities Flow*

**

*Figure 71:Update delivery address Activities Flow*

**

*Figure 72:Delete delivery address Activities Flow*

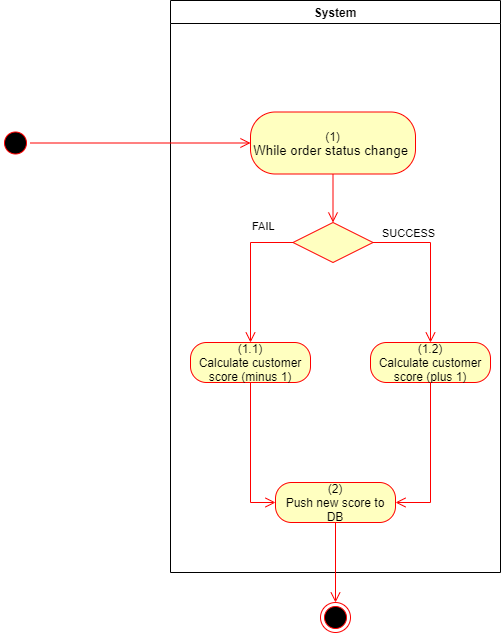
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Delivery Management (Customer)” khi người dùng chọn mục “Delivery Address” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Delivery Address”, hệ thống thực hiện tải màn hình “List Delivery Address(Customer)” bao gồm các địa chỉ giao hàng đã được tạo (listAddress):   * If (Click “Delivery Address”) => open(“List Delivery Address (Customer)”) * listAddress = getListtFromDB(“Select \* from Delivery\_Address inner join Customer where Delivery\_Address.custome.ID = customer.ID and customer.ID = thisAccount.id”) * listDisplay = listAddress * Hệ thống tải các địa chỉ giao hàng của người dùng đăng nhập hiện tại và hiển thị lên màn hình.  1. **Create**   Để tiến hành liên kết thêm địa chỉ giao hàng, ta thực hiện ấn chọn button “Thêm địa chỉ giao hàng” trên màn hình “List Delivery Address (Customer)”, lúc này form “CRUD Delivery Address” sẽ hiển thị và người dùng tiến hành nhập các thông tin cần thiết.   * If (Click “Thêm địa chỉ giao hàng”) => open(“CRUD Delivery Address”) * namePerson=”” * gender = NULL * phoneNumber = “” * province = NULL * district = NULL * ward = NULL * lane = “” * Hệ thống màn hình “CRUD Delivery Address” với các thông tin rỗng khi người dùng ấn chọn button “Thêm địa chỉ giao hàng” trong màn hình “List Delivery Address (Customer)”   Khi đã chọn nhập các thông tin cần thiết, để có thể lưu được lên database người dùng tiến hành nhấn button “Lưu” trong màn hình “CRUD delivery Address”:   * If (Click “lưu”) * If (“namePerson == “”, gender = NULL, phoneNumber= “”, province == NULL, district == NULL, ward ==NULL, lane == NULL”) => MSG01 * If (namePerson.contains([number] || namePerson.contains([SpecialChracter])) => MSG02 * If (phoneNumber == “”) => MSG01 * If (phoneNumber.contains([nonNumber]) => MSG03 * If (phoneNumber.length > 10 || phoneNumber < 10 || ! phoneNumber.startWith([validNumber])) => MSG4 * Else => newAddress = new Address(namePerson, gender, phoneNumber, province, district, ward, lane)=> listPayment.add (newAddress) * listDisplay.add (newAddress) * Khi click chọn button “Lưu”, hệ thống thực hiện kiểm tra các định dạng của dữ liệu vừa nhập nếu không nhận được bất kỳ MSG nào thì hệ thống sẽ cập nhật lại UI và cả database * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Delivery Address (customer)”   Khi cần tạo hoặc thêm hoặc xóa địa chỉ, thực hiện nhập/chọn cho các thông tin ở các ô “Tỉnh”, “Huyện”, “Xã”, “Đường”.   * listDistrict = GetAllDistrictFromDB("Select \* from District”) * dbDistrict.isEnable = false (Drop box district) * dbWard.isEnable = false (Drop box ward) * tbLane.isEnable = false (text box Lane) * Ban đầu sẽ tự động tải các tỉnh, các DropBox của “Huyện” và “Xã” và textbox “Đường” sẽ bị “vô hiệu hóa” * If(Chose “Tỉnh”) => dbDistrict.isEnable = true => getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “huyện” và tải tất cả các huyện theo ID của tỉnh đã chọn. * If(Chose “Huyện”) => db.Ward.isEnable = true => getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “Xã” và tải tất cả các huyện theo ID của tỉnh đã chọn.   Khi thay đổi chọn một “tỉnh”, “huyện”, “xã” bất kỳ, thì các cấp bậc dưới sẽ được “vô hiệu hóa” và xóa thông tin tương ứng, thay đổi lại danh sách lựa chọn tương ứng.   * If (change “Tỉnh”) * dbDistrict.isEnable = true, dbDistrict.SelectedItem = NULL,dbDistrict.ListDistrict=getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * dbWard.isEnable = false, dbWard.selectedItem = NULL, dbWard.listWard = [Empty list]. * tbLane.text = “”, tbLane.isEnable = false. * If (change “Huyện”) * dbWard.isEnable = true, dbWard.SelectedItem = NULL,dbWard.ListDistrict=getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * tbLane.text = “”, tbLane.isEnable = false. * If (change “Xã”) * tbLane.text = “”, tbLane.isEnable = false  1. **Update**   Để tiến hành chỉnh sửa một địa chỉ giao hàng, ta thực hiện ấn chọn button “Chỉnh sửa” trên một item địa chỉ bắt kỳ (SelectedAddress) màn hình “List Delivery Address (Customer)”, lúc này form “CRUD Delivery Address” sẽ hiển thị và người dùng tiến hành nhập các thông tin cần thiết.   * If (Click “Chỉnh sửa”) => open(“CRUD Delivery Address”) * namePerson=selectedAddress.name * gender = selectedAddress.gender * phoneNumber = selectedAddress.phoneNumber * province = selectedAddress.province * district = selectedAddress.district * ward = selectedAddress.ward * lane = selectedAddress.lane * Hệ thống màn hình “CRUD Delivery Address” với các thông tin rỗng khi người dùng ấn chọn button “Chỉnh sửa” trên một item địa chỉ bất kỳ trong màn hình “List Delivery Address (Customer)”   Khi đã chọn nhập các thông tin cần thiết, để có thể lưu được lên database người dùng tiến hành nhấn button “Lưu” trong màn hình “CRUD delivery Address”:   * If (Click “lưu”) * If (“namePerson == “”, gender = NULL, phoneNumber= “”, province == NULL, district == NULL, ward ==NULL, lane == NULL”) => MSG01 * If (namePerson.contains([number] || namePerson.contains([SpecialChracter])) => MSG02 * If (phoneNumber == “”) => MSG01 * If (phoneNumber.contains([nonNumber]) => MSG03 * If (phoneNumber.length > 10 || phoneNumber < 10 || ! phoneNumber.startWith([validNumber])) => MSG4 * Else => updateAddress = new Address(namePerson, gender, phoneNumber, province, district, ward, lane)=> listPayment.update (updateAddress) * listDisplay.update(updateAddress) * Khi click chọn button “Lưu”, hệ thống thực hiện kiểm tra các định dạng của dữ liệu vừa nhập nếu không nhận được bất kỳ MSG nào thì hệ thống sẽ cập nhật lại UI và cả database * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Delivery Address (customer)”   Khi cần tạo hoặc thêm hoặc xóa địa chỉ, thực hiện nhập/chọn cho các thông tin ở các ô “Tỉnh”, “Huyện”, “Xã”, “Đường”.   * listDistrict = GetAllDistrictFromDB("Select \* from District”) * dbDistrict.isEnable = false (Drop box district) * dbWard.isEnable = false (Drop box ward) * tbLane.isEnable = false (text box Lane) * Ban đầu sẽ tự động tải các tỉnh, các DropBox của “Huyện” và “Xã” và textbox “Đường” sẽ bị “vô hiệu hóa” * If(Chose “Tỉnh”) => dbDistrict.isEnable = true => getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “huyện” và tải tất cả các huyện theo ID của tỉnh đã chọn. * If(Chose “Huyện”) => db.Ward.isEnable = true => getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * Khi chọn đã chọn một tỉnh bất kỳ, hệ thống sẽ bỏ “vô hiệu hóa” ở DropBox “Xã” và tải tất cả các huyện theo ID của tỉnh đã chọn.   Khi thay đổi chọn một “tỉnh”, “huyện”, “xã” bất kỳ, thì các cấp bậc dưới sẽ được “vô hiệu hóa” và xóa thông tin tương ứng, thay đổi lại danh sách lựa chọn tương ứng.   * If (change “Tỉnh”) * dbDistrict.isEnable = true, dbDistrict.SelectedItem = NULL,dbDistrict.ListDistrict=getAllDistrictByProvinceID(“Select \* from District where province.ID = [selectedProvince.ID]) * dbWard.isEnable = false, dbWard.selectedItem = NULL, dbWard.listWard = [Empty list]. * tbLane.text = “”, tbLane.isEnable = fasle. * If (change “Huyện”) * dbWard.isEnable = true, dbWard.SelectedItem = NULL,dbWard.ListDistrict=getAllWardByDistrictID(“Select \* from Ward where ward.ID = [selectedWard.ID]) * tbLane.text = “”, tbLane.isEnable = false. * If (change “Xã”) * tbLane.text = “”, tbLane.isEnable = false  1. **Delete**   Để tiến hành xóa một địa chỉ giao hàng, ta thực hiện ấn chọn button “Xóa” của một Item đơn vị thanh toán bất kỳ (SelectedAddress) trên màn hình “List Delivery Address (customer)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * listDisplay.delete(selectedAddress) => deleteOnDB(selectedAddresst) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Delivery Address (customer)” |

### UC31: Tính điểm khách hàng

|  |  |
| --- | --- |
| **Name** | **Tính điểm khách hàng** |
| **Description** | Chức năng tự động hệ thống sẽ tính điểm cho khách hàng |
| **Actor** | 1. System |
| **Trigger** | * Khi một đơn hàng chuyển trạng thái “giao hàng” SUCCESS hoặc FAIL. |
| **Pre-condition** | None |
| **Post-condition** | * Tính được điểm khách hàng |

#### Activities Flow



*Figure 73: Calculate customer score Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Business rules:**  Hệ thống sẽ sẵn cho mỗi khách hàng tạo tài khoản 100 điểm, và sẽ cập nhật trên màn hình “List Customer (Seller)” (listCustomer).   * Each (“5000000”) => listChange = listStatusChange(“orderInfoAPI”) * Foreach (Item item in listChange) * if (item.status == “SUCCESS”) => customer = getCustomerFromBD(“Seletc \* from customer where customer.id = item.customerID”) => Customer.point += 1 * if (item.status == “FAIL”) => customer = getCustomerFromBD(“Seletc \* from customer where customer.id = item.customerID”) => Customer.point -= 1   Mỗi 5000000ms (5 phút) thì hệ thống sẽ gọi đến orderInfoAPI (lúc tạo các đơn vị vận chuyển có được) của tất cả đơn hàng không trong trạng thái “SUCCESS”/”FAIL” hiện tại, nếu có sự thay đổi sang “SUCCESS”/”FAIL” thì sẽ thực hiện tính điểm. |

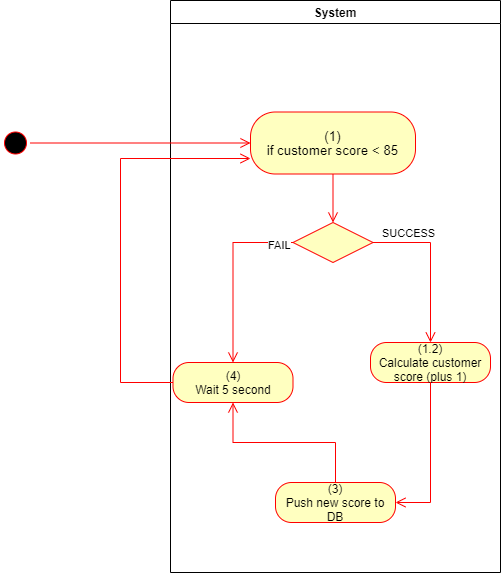
### UC32: Thêm/Xóa khỏi blacklist

|  |  |
| --- | --- |
| **Name** | **Thêm/xóa khỏi blacklist** |
| **Description** | Chức năng cho phép người dùng thêm xóa khách hàng khỏi blacklist |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng ấn chọn mục “Customer Management” |
| **Pre-condition** | Đăng nhập với tư cách BGD hoặc phòng kinh doanh |
| **Post-condition** | * Thêm xóa được khách hàng khỏi blacklist. |

#### Activities Flow



*Figure 74: byhand add/delete blacklist Activities Flow*

**

*Figure 75: auto add blacklist Activities Flow*

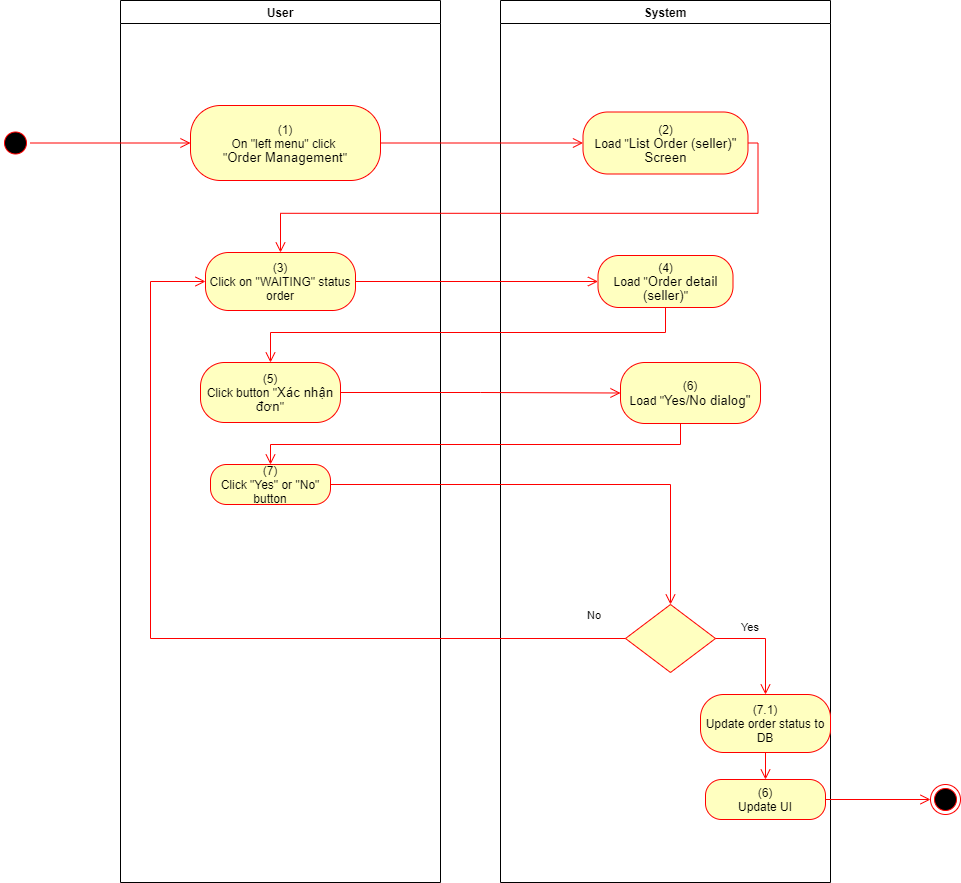
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Customer (seller)” khi người dùng ấn chọn mục “Customer Management” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**   1. **Thêm/xóa vào blacklist bằng thao tác**   Để có thể thêm một khách hàng vào blacklist, thực hiện ấn chọn mục “Customer Management” ở “left menu”, lúc này danh sách các khách hàng sẽ hiển thị.   * If (Click “Customer Management”) * listCustomer = getListFromDB(“Select \* from Customer innerjoin Points where points.customerID = Customer.id”) * listDisplay = listCustomer * Khi ấn chọn mục “Customer management” bên “left menu” thì sẽ tải thông tin từ database và hiển thị trên màn hình.   Để tiến hành thêm/xóa một khách hàng vào blacklist ta thực hiện ấn chọn vào CheckBox “Có phải blacklist” trên khách hàng đó (selectedCustomer) để tiến hành đưa vào blakclist, lúc này một Yes/No dialog sẽ hiên thị và hỏi liệu có chắc chắn muốn đưa vào/ra blacklist:   * If (Click Check box “Đưa vào blacklist”) * If (Click “Yes”) => selectedCustomer.isBlacklist = !selectedCustomer.isBlacklist * Nếu click “Yes” thì nếu đang trong blacklist sẽ được đưa ra ngoài, còn đang ở ngoài thì sẽ vào trong blacklist. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Customer (seller)”:  1. **Tự động thêm vào blacklist:**   Khi điểm của một customer dưới 85, sẽ được đưa tự động vào hệ thống và cập nhật ngay trên UI, database:   * Each("5000000”) * Foreach (Customer customer in listCusstomer) * If (customer.points <= 85) => customer.isBlacklist = true. |

### UC33: Xác nhận đơn hàng

|  |  |
| --- | --- |
| **Name** | **Xác nhận đơn hàng** |
| **Description** | Xác nhận đơn hàng (từ trạng thái đang chờ -> xác nhận), chức năng này dành cho người bán |
| **Actor** | 1. BGD 2. Phòng kinh doanh |
| **Trigger** | * Khi người dùng ấn chọn mục “Order Management” |
| **Pre-condition** | Đăng nhập với tư cách phòng kinh doanh hoặc BGD |
| **Post-condition** | * Chuyển đơn hàng từ đang chờ sang xác nhận |

#### Activities Flow



*Figure 76: Confirm order Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Order (seller)” khi người dùng ấn chọn mục “Order Management” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**  Để có thể chuyển trạng thái đơn hàng từ “đang chờ” sang “đã xác nhận”, người dùng ấn chọn vào mục “Order Management” ở “left menu”, hệ thống sẽ hiển thị màn hình “List Order (seller)”, nơi hiển thị tất cả các đơn hàng gửi đến từ khách hàng.   * If (Click “Order Management”) => open(“List Order (seller)”) * listOrder = getListFromDB(“Select \* from Order”) * listDisplay = listOrder (list mà sẽ hiển thị ra màn hình) * dbFilter.selection = “ALL” (Drop box “thanh lọc”) * Khi ấn chọn vào mục “Order Managemnet” ở “left menu” hệ thống sẽ mở “List Order (seller)” và tải danh sách tất cả đơn hàng từ database về và hiển thị, cài đặt lựa chọn hiện tại của thanh lọc là “ALL”. * Thanh lọc có thể lọc theo: "WAITING", "CONFIRMED", "DELIVERED", "SUCCESS", "CANCEL/FAIL"   Để có thể tiến hành chuyển trạng thai từ “Đang chờ” sang “đã xác nhận”, người dùng ấn chọn vào một đơn hàng đang hiển thị trạng thái “Đang chờ” bất kỳ (selectionOrder) trên màn hình “List Order (seller)”, hệ thống sẽ thực hiện tải màn hình “Order detail (seller)” theo thông tin của đơn hàng vừa chọn:   * If (Click [selectionOrder]) => open(“Order detail (seller)”) * Status = selectionOrder.status (ở đây là “Đang chờ”) * timeOrder = selectionOrder.timeOrder * idCustomer = selectionOrder.idCustomer * nameCustomer = selectionOrder.nameCustomer * gender = selectionOrder.gender * phoneNumber = selectionOrder.phoneNumber * province = selectionOrder.province * deliveryUnit = selectionOrder.deliveryUnit * paymentUnit = selectionOrder.paymentUnit * district = selectionOrder.district * ward = selectionOrder.ward * lane = selectionOrder.lane * listProduct = selectionOrder.listProduct * sum = selectionOrder.sum * listLocation = selectionOrder.listLocation * reasonCustomer = selectionOrder.reasonCustomer * reasonSeller = selectionOrder.reasonSeller * Khi ấn chọn một đơn hàng bất kỳ, màn hình “Order Detail (Seller)” sẽ hiển thị và thông tin là các thông tin của đơn hàng vừa chọn.   Để tiến hành chọn trạng thái, thực hiện ấn chọn button trạng thái đang không “Vô hiệu hóa”, ở đây (trạng thái “Đang chờ”), thì button “Xác nhận” và “Hủy” sẽ không bị vô hiệu hóa.   * Trạng thái vô hiệu hóa của các button chuyển trạng thái: * buttonCancelOrder.isEnable = True (button hủy đơn hàng) * buttonConfirm.isEnable = true (button xác nhận) * buttonDelivered.isEnable = false (button đã giao bên vận chuyển) * If (Click “Xác nhận đơn”) * If (Click “Yes”) => selectionOrder.status = “CONFIRM” * If (Click “No”) => back() * Khi ấn chọn button “Xác nhận” sẽ xuất hiện 1 Yes/No dialog và nếu click yes thì sẽ chuyển trạng thái, click NO thì sẽ back về màn hình “Order detail (seller)”. |

### UC34: Xác nhận đơn hàng (Giao bên vận chuyển)

|  |  |
| --- | --- |
| **Name** | **Xác nhận đơn hàng (giao bên vận chuyển)** |
| **Description** | Xác nhận đơn hàng (từ trạng thái đang chờ -> xác nhận), chức năng này dành cho người bán |
| **Actor** | 1. BGD 2. Phòng kinh doanh |
| **Trigger** | * Khi người dùng ấn chọn mục “Order Management” |
| **Pre-condition** | Đăng nhập với tư cách phòng kinh doanh hoặc BGD |
| **Post-condition** | * Chuyển đơn hàng từ xác nhận sang đã giao bên vận chuyển |

#### Activities Flow



*Figure 77: change status to delivered order Activities Flow*

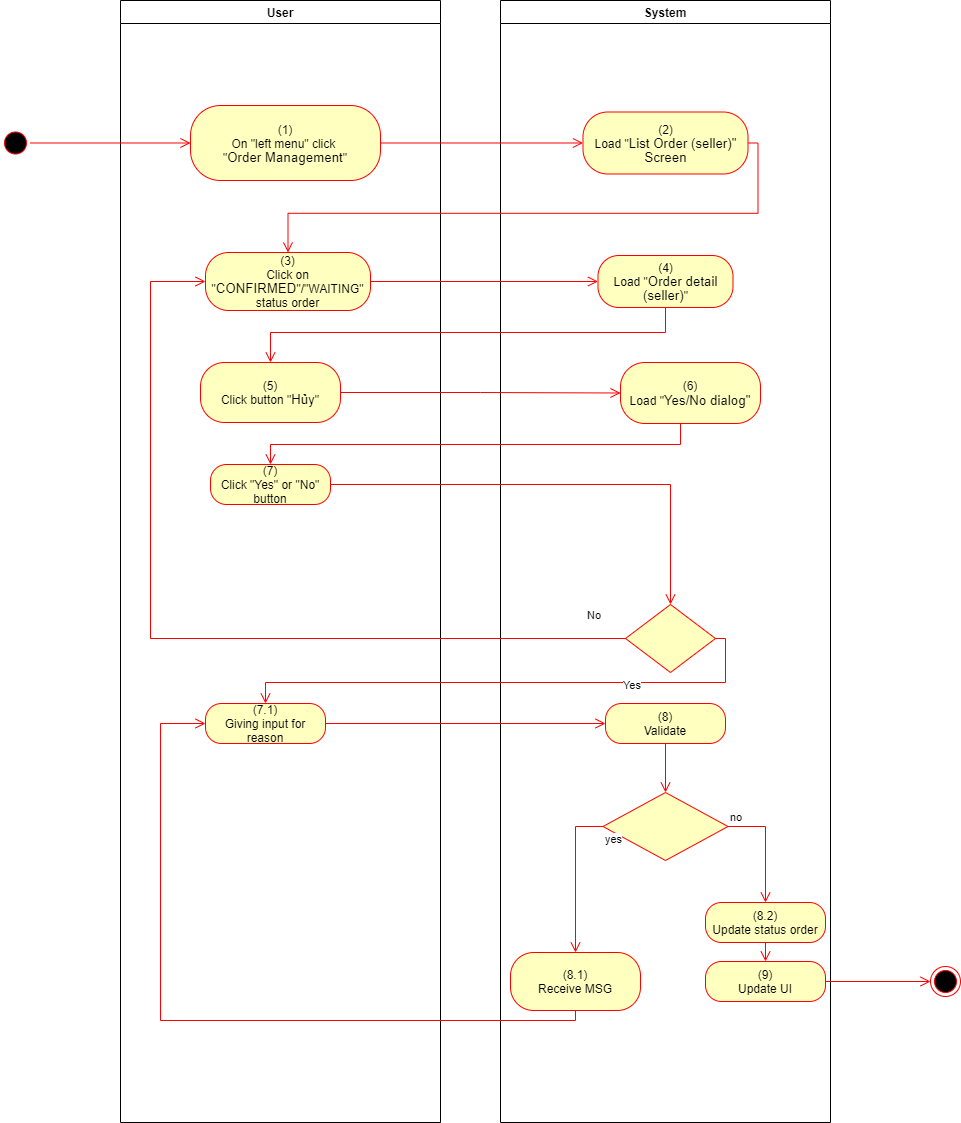
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Order (seller)” khi người dùng ấn chọn mục “Order Management” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**  Để có thể chuyển trạng thái đơn hàng từ “xác nhận” sang “đã giao bên vận chuyển”, người dùng ấn chọn vào mục “Order Management” ở “left menu”, hệ thống sẽ hiển thị màn hình “List Order (seller)”, nơi hiển thị tất cả các đơn hàng gửi đến từ khách hàng.   * If (Click “Order Management”) => open(“List Order (seller)”) * listOrder = getListFromDB(“Select \* from Order”) * listDisplay = listOrder (list mà sẽ hiển thị ra màn hình) * dbFilter.selection = “ALL” (Drop box “thanh lọc”) * Khi ấn chọn vào mục “Order Managemnet” ở “left menu” hệ thống sẽ mở “List Order (seller)” và tải danh sách tất cả đơn hàng từ database về và hiển thị, cài đặt lựa chọn hiện tại của thanh lọc là “ALL”. * Thanh lọc có thể lọc theo: "WAITING", "CONFIRMED", "DELIVERED", "SUCCESS", "CANCEL/FAIL"   Để có thể tiến hành chuyển trạng thai từ “Đang chờ” sang “đã xác nhận”, người dùng ấn chọn vào một đơn hàng đang hiển thị trạng thái “Đang chờ” bất kỳ (selectionOrder) trên màn hình “List Order (seller)”, hệ thống sẽ thực hiện tải màn hình “Order detail (seller)” theo thông tin của đơn hàng vừa chọn:   * If (Click [selectionOrder]) => open(“Order detail (seller)”) * Status = selectionOrder.status (ở đây là “Xác nhận”) * timeOrder = selectionOrder.timeOrder * idCustomer = selectionOrder.idCustomer * nameCustomer = selectionOrder.nameCustomer * gender = selectionOrder.gender * phoneNumber = selectionOrder.phoneNumber * province = selectionOrder.province * district = selectionOrder.district * deliveryUnit = selectionOrder.deliveryUnit * paymentUnit = selectionOrder.paymentUnit * ward = selectionOrder.ward * lane = selectionOrder.lane * listProduct = selectionOrder.listProduct * sum = selectionOrder.sum * listLocation = selectionOrder.listLocation * reasonCustomer = selectionOrder.reasonCustomer * reasonSeller = selectionOrder.reasonSeller * Khi ấn chọn một đơn hàng bất kỳ, màn hình “Order Detail (Seller)” sẽ hiển thị và thông tin là các thông tin của đơn hàng vừa chọn.   Để tiến hành chọn trạng thái, thực hiện ấn chọn button trạng thái đang không “Vô hiệu hóa”, ở đây (trạng thái “Xác nhận”), thì button “Giao bên vận chuyển” và “Hủy” sẽ không bị vô hiệu hóa.   * Trạng thái vô hiệu hóa của các button chuyển trạng thái: * buttonCancelOrder.isEnable = True (button hủy đơn hàng) * buttonConfirm.isEnable = false (button xác nhận) * buttonDelivered.isEnable = true (button đã giao bên vận chuyển) * If (Click “giao bên vận chuyển”) * If (Click “Yes”) => selectionOrder.status = “DELIVERED”, buttonCancelOrder.isEnable = false * If (Click “No”) => back() * Khi ấn chọn button “Xác nhận” sẽ xuất hiện 1 Yes/No dialog và nếu click yes thì sẽ chuyển trạng thái và đưa button “hủy đơn hàng” vào trạng thái vô hiệu hóa, click NO thì sẽ back về màn hình “Order detail (seller)”. |

### UC35: Hủy đơn hàng (bên bán)

|  |  |
| --- | --- |
| **Name** | **Hủy đơn hàng (bên bán)** |
| **Description** | Hỗ trợ hủy đơn hàng từ phía người bán. |
| **Actor** | 1. BGD 2. Phòng kinh doanh |
| **Trigger** | * Khi người dùng ấn chọn mục “Order Management” |
| **Pre-condition** | Đăng nhập với tư cách phòng kinh doanh hoặc BGD |
| **Post-condition** | * Hủy được đơn hàng |

#### Activities Flow



*Figure 78: Cancel order by seller Activities Flow*

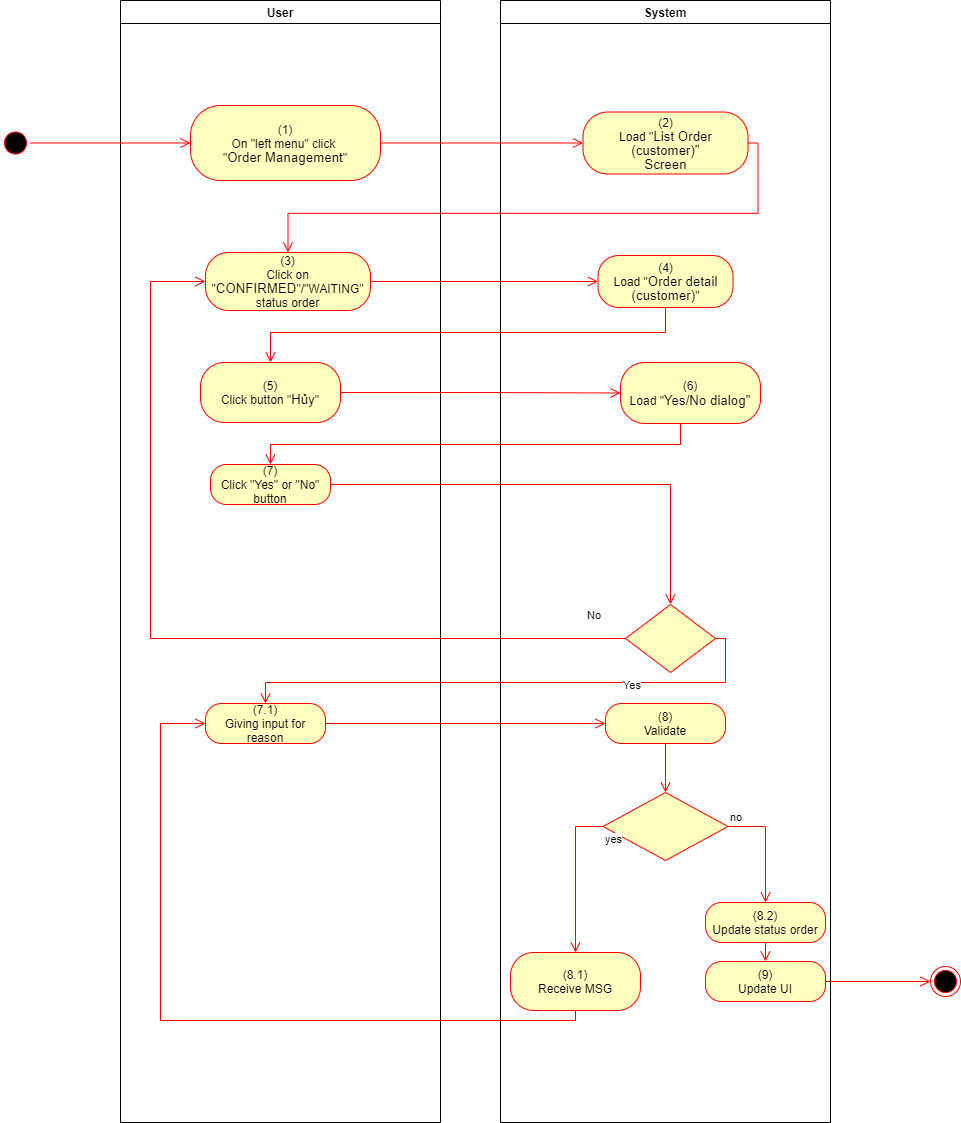
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Order (seller)” khi người dùng ấn chọn mục “Order Management” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**  Để có thể hủy đơn hàng, người dùng ấn chọn vào mục “Order Management” ở “left menu”, hệ thống sẽ hiển thị màn hình “List Order (seller)”, nơi hiển thị tất cả các đơn hàng gửi đến từ khách hàng.   * If (Click “Order Management”) => open(“List Order (seller)”) * listOrder = getListFromDB(“Select \* from Order”) * listDisplay = listOrder (list mà sẽ hiển thị ra màn hình) * dbFilter.selection = “ALL” (Drop box “thanh lọc”) * Khi ấn chọn vào mục “Order Managemnet” ở “left menu” hệ thống sẽ mở “List Order (seller)” và tải danh sách tất cả đơn hàng từ database về và hiển thị, cài đặt lựa chọn hiện tại của thanh lọc là “ALL”. * Thanh lọc có thể lọc theo: "WAITING", "CONFIRMED", "DELIVERED", "SUCCESS", "CANCEL/FAIL"   Để có thể tiến hành hủy đơn, người dùng ấn chọn vào một đơn hàng đang hiển thị trạng thái “Đang chờ” hoặc “Xác nhận” bất kỳ (selectionOrder) trên màn hình “List Order (seller)”, hệ thống sẽ thực hiện tải màn hình “Order detail (seller)” theo thông tin của đơn hàng vừa chọn:   * If (Click [selectionOrder]) => open(“Order detail (seller)”) * Status = selectionOrder.status (ở đây là “Xác nhận” hoặc “Xác nhận”) * timeOrder = selectionOrder.timeOrder * idCustomer = selectionOrder.idCustomer * nameCustomer = selectionOrder.nameCustomer * deliveryUnit = selectionOrder.deliveryUnit * paymentUnit = selectionOrder.paymentUnit * gender = selectionOrder.gender * phoneNumber = selectionOrder.phoneNumber * province = selectionOrder.province * district = selectionOrder.district * ward = selectionOrder.ward * lane = selectionOrder.lane * listProduct = selectionOrder.listProduct * sum = selectionOrder.sum * listLocation = selectionOrder.listLocation * reasonCustomer = selectionOrder.reasonCustomer * reasonSeller = selectionOrder.reasonSeller * Khi ấn chọn một đơn hàng bất kỳ, màn hình “Order Detail (Seller)” sẽ hiển thị và thông tin là các thông tin của đơn hàng vừa chọn.   Để tiến hành chọn trạng thái, thực hiện ấn chọn button trạng thái đang không “Vô hiệu hóa”, ở đây (trạng thái “Xác nhận”/”đang chờ”), thì button “Hủy” sẽ không bị vô hiệu hóa.   * Trạng thái vô hiệu hóa của các button chuyển trạng thái: * buttonCancelOrder.isEnable = True (button hủy đơn hàng) * If (Click “Hủy”) * If (Click “Yes”) => if (resonSeller == “”) => MSG01 * Else selectionOrder.status = “CANCEL/FAIL” => buttonCancelOrder.isEnable = False, buttonConfirm.isEnable = false, buttonDeliverd,isEnble = false. * If (Click “No”) => back() * Khi ấn chọn button “Xác nhận” sẽ xuất hiện 1 Yes/No dialog (có trường nhập lý do)và nếu click yes mà không nhận MSG nào thì sẽ chuyển trạng thái, click NO thì sẽ back về màn hình “Order detail (seller)”. |

### UC36: Hủy đơn hàng (bên mua)

|  |  |
| --- | --- |
| **Name** | **Hủy đơn hàng (bên mua)** |
| **Description** | Hỗ trợ hủy đơn hàng từ phía người mua. |
| **Actor** | 1. Khách hàng |
| **Trigger** | * Khi người dùng ấn chọn mục “Order Management” |
| **Pre-condition** | Đăng nhập với tư cách khách hàng |
| **Post-condition** | * Hủy được đơn hàng |

#### Activities Flow



*Figure 79:Cancel order by customer Activities Flow*

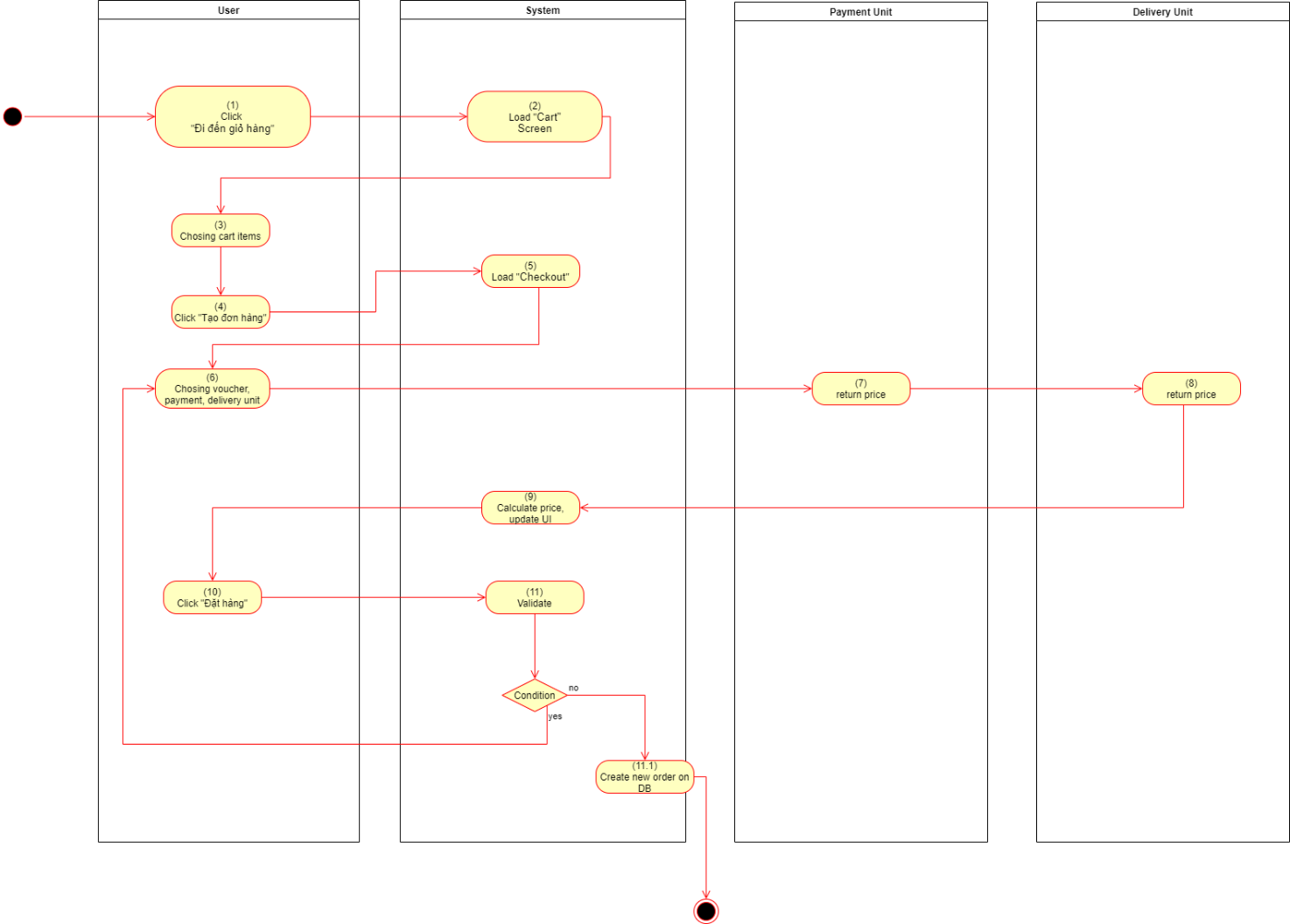
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Order (customer)” khi người dùng ấn chọn mục “Order Management” ở “left menu” |
| *(2)* | *BR2* | **Business Rules:**  Để có thể hủy đơn hàng, người dùng ấn chọn vào mục “Order Management” ở “left menu”, hệ thống sẽ hiển thị màn hình “List Order (customer)”, nơi hiển thị tất cả các đơn hàng gửi đến từ khách hàng.   * If (Click “Order Management”) => open(“List Order (customer)”) * listOrder = getListFromDB(“Select \* from Order”) * listDisplay = listOrder (list mà sẽ hiển thị ra màn hình) * dbFilter.selection = “ALL” (Drop box “thanh lọc”) * Khi ấn chọn vào mục “Order Managemnet” ở “left menu” hệ thống sẽ mở “List Order (customer)” và tải danh sách tất cả đơn hàng từ database về và hiển thị, cài đặt lựa chọn hiện tại của thanh lọc là “ALL”. * Thanh lọc có thể lọc theo: "WAITING", "CONFIRMED", "DELIVERED", "SUCCESS", "CANCEL/FAIL"   Để có thể tiến hành hủy đơn, người dùng ấn chọn vào một đơn hàng đang hiển thị trạng thái “Đang chờ” hoặc “Xác nhận” bất kỳ (selectionOrder) trên màn hình “List Order (customer)”, hệ thống sẽ thực hiện tải màn hình “Order detail (customer)” theo thông tin của đơn hàng vừa chọn:   * If (Click [selectionOrder]) => open(“Order detail (customer)”) * Status = selectionOrder.status (ở đây là “Xác nhận” hoặc “Xác nhận”) * timeOrder = selectionOrder.timeOrder * idCustomer = selectionOrder.idCustomer * nameCustomer = selectionOrder.nameCustomer * gender = selectionOrder.gender * phoneNumber = selectionOrder.phoneNumber * province = selectionOrder.province * district = selectionOrder.district * ward = selectionOrder.ward * deliveryUnit = selectionOrder.deliveryUnit * paymentUnit = selectionOrder.paymentUnit * lane = selectionOrder.lane * listProduct = selectionOrder.listProduct * sum = selectionOrder.sum * listLocation = selectionOrder.listLocation * reasonCustomer = selectionOrder.reasonCustomer * reasonSeller = selectionOrder.reasonSeller * Khi ấn chọn một đơn hàng bất kỳ, màn hình “Order Detail (customer)” sẽ hiển thị và thông tin là các thông tin của đơn hàng vừa chọn.   Để tiến hành chọn trạng thái, thực hiện ấn chọn button trạng thái đang không “Vô hiệu hóa”, ở đây (trạng thái “Xác nhận”/”đang chờ”), thì button “Hủy” sẽ không bị vô hiệu hóa.   * Trạng thái vô hiệu hóa của các button chuyển trạng thái: * buttonCancelOrder.isEnable = True (button hủy đơn hàng) * If (Click “Hủy”) * If (Click “Yes”) => if (reasonCustomer == “”) => MSG01 * If (thisAccount.isInBlacklist) => MSG26 * Else selectionOrder.status = “CANCEL/FAIL” => buttonCancelOrder.isEnable = False, * If (Click “No”) => back() * Khi ấn chọn button “Xác nhận” sẽ xuất hiện 1 Yes/No dialog (có trường nhập lý do)và nếu click yes mà không nhận MSG nào thì sẽ chuyển trạng thái, click NO thì sẽ back về màn hình “Order detail (customer)”. |

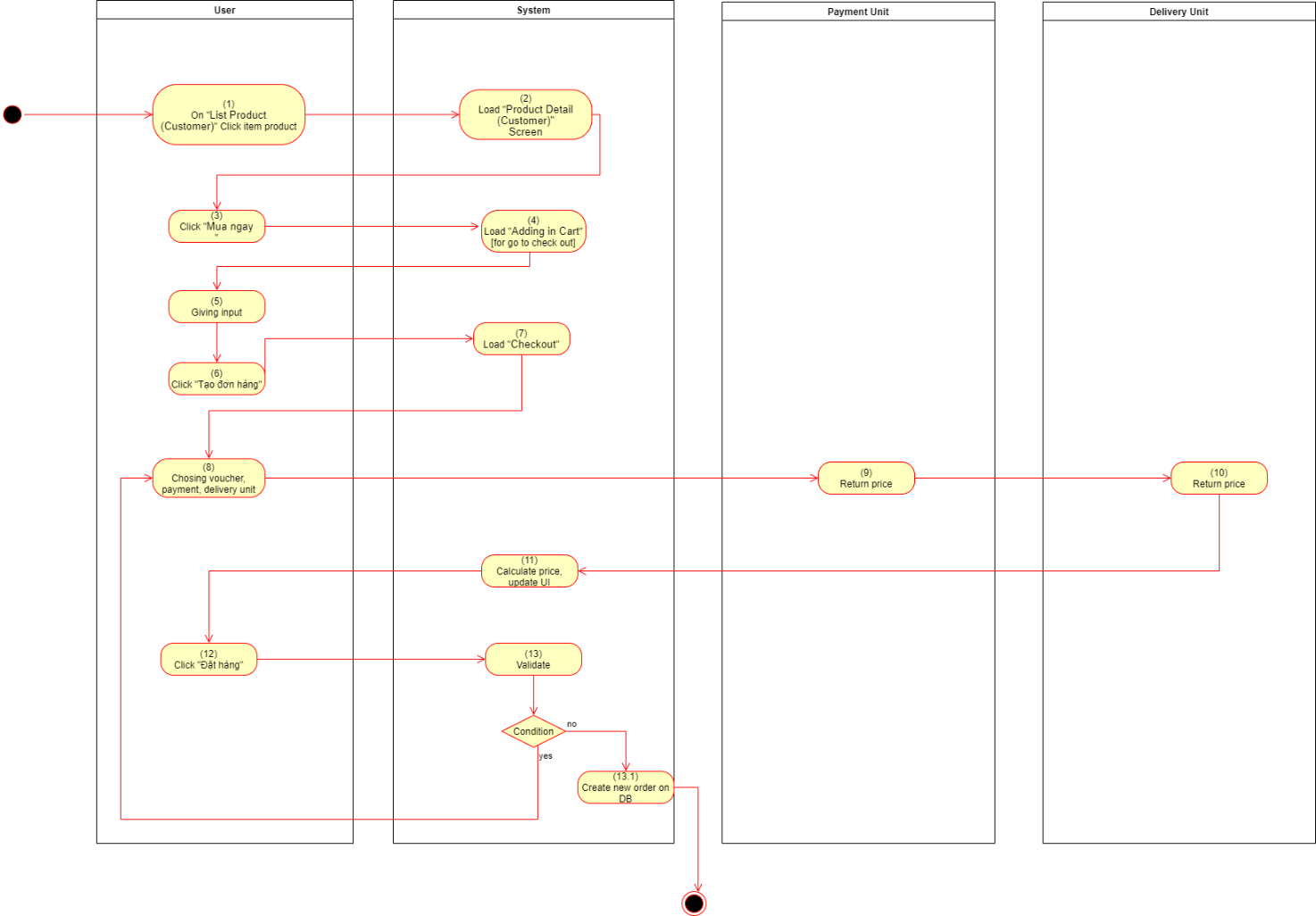
### UC37: Tạo đơn hàng (Khách hàng)

|  |  |
| --- | --- |
| **Name** | **Tạo đơn hàng (Khách hàng)** |
| **Description** | Hỗ trợ tạo đơn hàng từ phía người mua. |
| **Actor** | 1. User 2. System 3. Payment unit 4. Delivery unit |
| **Trigger** | * Khi người dùng ấn chọn mục “Tạo đơn hàng” ở màn hình “Cart” * Hoặc khi người dùng ấn chọn mục “Mua ngay” ở màn hình “Product Detail” |
| **Pre-condition** | None |
| **Post-condition** | * Tạo được đơn hàng |

#### Activities Flow



*Figure 80: Make order by cart Activities Flow*

**

*Figure 81: Make order directly Activities Flow*

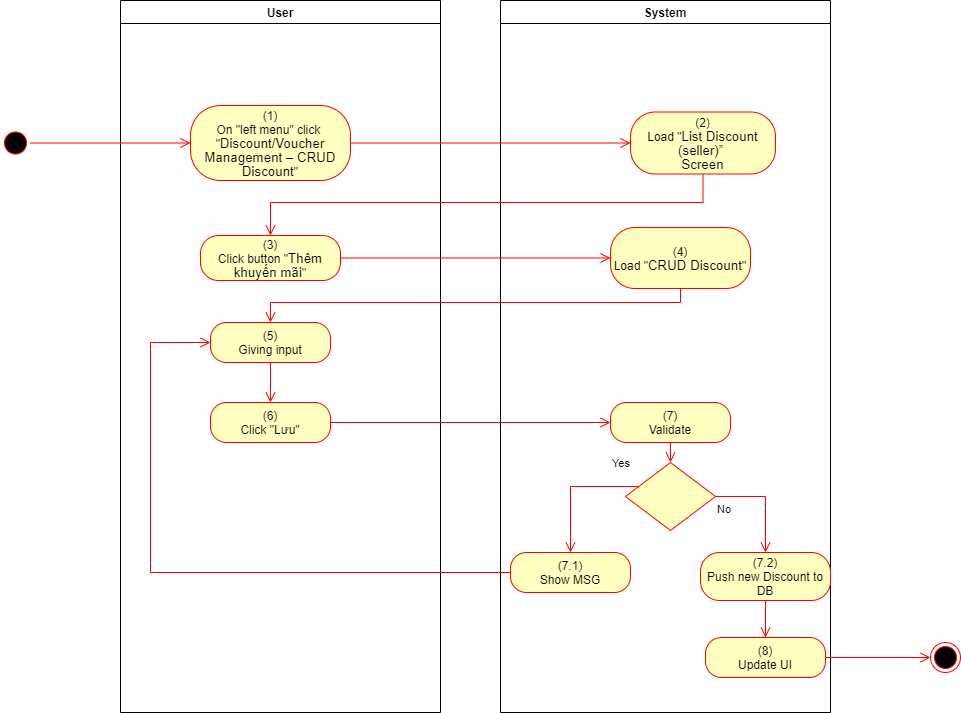
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “Cart” khi người dùng ấn chọn nút “Đi đến giỏ hàng”. * Hệ thống tải màn hình ”Checkout” khi người dùng ấn chọn nút “Tạo đơn hàng” ở màn hình “Cart” |
| *(2)* | *BR2* | **Business Rules:**   1. **Tạo đơn hàng bằng giỏ hàng:**   Để bắt đầu các bước tạo đơn hàng, đầu tiên bắt đầu vào màn hình “Cart” bằng cách ấn chọn nút “Đi đến giỏ hàng”, trong màn hình “Cart” người dùng tiến hành chọn các Item giỏ hàng sẽ đem đi thực hiện thanh toán (listChose).   * In (“Cart”) => [choose multi cartitem] = listChose   Để có thể tiến hành tạo đơn hàng, người dùng người dùng ấn chọn nút “Tạo đơn hàng” ở màn hình “Cart”, hệ thống sẽ thực hiện tải màn hình “Checkout” theo thông tin của các Item giỏ hàng vừa chọn:   * If (Click “Tạo đơn hàng”)=> open(“Checkout”) * nameCustomer = thisAccount.name * gender = thisAccount.gender * phoneNumber = thisAccount.phoneNumber * address = thisAccount. defaultAddress.lane + “, ” thisAccount. defaultAddress.ward + “, ” + thisAccount. defaultAddress.district + “, ” + thisAccount. defaultAddress.province * listDeliveryUnit = getListFromDB(“select \* from Delivery”) [Load màn hình “List Delivery Unit (Customer)” và tải list khi ấn chọn mục listDeliveryUnit (“Chọn đơn vị vận chuyển”), lấy đơn vị được chọn cuối cùng (ấn chọn) trước khi thoát “List Delivery Unit (Customer)”] * listPaymentUnit = getListFromDB(“select \* from payment where payment.customerid = thisAccount.id”) [Load màn hình “List Payment Unit (Customer)” và tải list khi ấn chọn mục listPaymentUnit (“Chọn đơn vị thanh toán”), lấy đơn vị được chọn cuối cùng (ấn chọn) trước khi thoát “List Payment Unit (Customer)”] * listVoucher = NULL * listProduct = listChose * foreach(Product product in listProduct) => sum += product.price \* product.amount [end loop] => sum+= DeliveryUnit.getPrice(selectedDelivery.priceAPI, product.address.province, product.address.district, product.address.ward) * sumForSubmit = sum * sum+= UnitPayment(selectedPayment.priceAPI, sum) * Khi ấn chọn button “Tạo đơn hàng”, màn hình “Checkout” sẽ hiển thị và thông tin là các thông tin của các item giỏ hàng vừa chọn, thông tin tài khoản, địa chỉ mặc định. * Để tính tổng tiền ta lấy giá tiền thông qua tổng tiền các sản phẩm, và tiền lấy qua API từ bên vận chuyển, API từ bên thanh toán * Tổng tiền để gửi đi lên DB sẽ không bao gồm tiền từ API thanh toán.   **\*Để biết được sẽ lấy API lại từ đơn vị vận chuyển (selectedDelivery) hay đơn vị thanh toán (selectedPayment) nào thì cần phải chọn đơn vị thanh toán và vận chuyển trước.**  **\*Để có thể chọn voucher, thực hiện Click vào gridview của voucher để mở màn hình “List Voucher (Customer)” để tiến hành chọn các voucher. Lưu ý: có thể chọn nhiều voucher, nhưng với mỗi sản phẩm đang mua hiện tại, không thể có cùng lúc 2 voucher tác động lên.**  **=> Hàm check: checkIsEffectAlready():**   * **Foreach (Product product in listProduct) if (product.IsInMoreThanOne(listVoucher)) => return true** * **Return false**   Để thực hiện thanh toán, thực hiện ấn chọn nút “Đặt hàng” trên màn hình “Checkout” để đặt hàng   * If (Click “Đặt hàng”) * If (checkIsEffectAlready()) => nhận MSG30 * PushtoDB(new Order(thisAccount, listProduct, deliveryUnit, paymentUnit, status = “WAITING”)) * Back() * Khi ấn chọn button “Đặt hàng” hệ thống sẽ đưa đơn hàng lên database và quay lại màn hình main screen.  1. **Tạo đơn hàng trực tiếp**   Để có thể tạo một đơn hàng một cách trực tiếp, người dùng thực hiện vào main screen nơi tải màn hình “List Product (Customer)”, để chọn một sản phẩm bất kỳ, sau đó màn hình “Product Detail (Customer)” sẽ hiển thị.   * If (Click [selectedProduct]) => open(“Product Detail (Customer)”) * displayProduct = selectedProduct. * Màn hình sẽ hiển thị thông tin về sản phẩm được chọn khi click chọn vào sản phẩm đó.   Để tiến hành đặt hàng, thực hiện chọn các thuộc tính hoặc ấn chọn “Mua ngay” để mở Bottom sheet “Adding in Cart” [lúc này button “Thêm vào giỏ hàng” thay bằng button “Tạo đơn hàng” (trong màn hình “Adding in Cart”)].   * If (hasChoseProperty()) => open(“Adding in cart”)   + “Adding in cart”.listPropSelectedOpt = “Detail Product”.listPropSelectedOtp   + Adding in cart”.listPropSelection = selectedProduct.listPropSelection   + “Adding in cart”.amount = 1.   + “Adding in cart”.idProduct = selectedProduct.idProduct   + “Adding in cart”.nameProduct = selectedProduct.nameProduct   + “Adding in cart”.image = selectedProduct.image   + “Adding” in cart”.price = selectedProduct.price * Nếu đã chọn các thuộc tính ở màn hình “Detail Product” rồi thì khi mở “Adding in cart” sẽ được giữ nguyên lựa chọn. (listPropselectedOtp) * Khi mở màn hình “Adding in cart” một số thông tin sản phẩm sẽ được truyền qua. (listPropSelection, id, name, price, image) * **listPropSelection là danh sách cần lựa chọn, listPropSelectedOtp là danh sách các lựa chọn đã chọn.** * Else => open("Adding in cart”) * “Adding in cart”.listPropSelectedOtp = NULL * “Adding in cart”.listPropSelection = selectedProduct.listPropSelection * “Adding in cart”.amount = 1 * “Adding in cart”.idProduct = selectedProduct.idProduct * “Adding in cart”.nameProduct = selectedProduct.nameProduct * “Adding in cart”.image = selectedProduct.image * “Adding” in cart”.price = selectedProduct.price * Nếu chưa chọn thì thuộc tính nội sản phẩm sẽ được giữ nguyên là chưa chọn * Khi mở màn hình “Adding in cart” một số thông tin sản phẩm sẽ được truyền qua. (listPropSelection, id, name, price, image)   Để tăng/giảm số lượng lúc thêm vào giỏ hàng thì tại lúc đang mở màn hình “Adding in cart” ta ấn chọn button “tăng số lượng”/”giảm số lượng”.   * If (Click “tăng số lượng”) => “Adding in cart”.amount ++ * If (Click “giảm số lượng”) => “Adding in cart”.amount - - * If (amount == 1) => [Do nothing] * Khi ấn chọn “tăng số lượng” thì sẽ tăng amount, khi click “giảm số lượng” sẽ giảm amount, khi giảm còn 1 thì sẽ không còn giảm được nữa.   Để tiến hành sang bước “Checkout” ta ấn chọn button “Tạo đơn hàng” trong “Adding in cart”   * If (Click “Tạo đơn hàng”)=> open(“Checkout”) * nameCustomer = thisAccount.name * gender = thisAccount.gender * phoneNumber = thisAccount.phoneNumber * address = thisAccount. defaultAddress.lane + “, ” thisAccount. defaultAddress.ward + “, ” + thisAccount. defaultAddress.district + “, ” + thisAccount. defaultAddress.province * listDeliveryUnit = getListFromDB(“select \* from Delivery”) [Load màn hình “List Delivery Unit (Customer)” và tải list khi ấn chọn mục listDeliveryUnit (“Chọn đơn vị vận chuyển”), lấy đơn vị được chọn cuối cùng (ấn chọn) trước khi thoát “List Delivery Unit (Customer)”] * listPaymentUnit = getListFromDB(“select \* from payment where payment.customerid = thisAccount.id”) [Load màn hình “List Payment Unit (Customer)” và tải list khi ấn chọn mục listPaymentUnit (“Chọn đơn vị thanh toán”), lấy đơn vị được chọn cuối cùng (ấn chọn) trước khi thoát “List Payment Unit (Customer)”] * listProduct =selectedProduct (listProduct.lenght == 1) * sum = selectedProduct.price \* selectedProduct.amount * sum += DeliveryUnit.getPrice(selectedDelivery.priceAPI, selectedProduct.address.province, selectedProduct .address.district, selectedProduct.address.ward) * sumForSubmit = sum * sum+= UnitPayment(selectedPayment.priceAPI, sum) * Khi ấn chọn button “Tạo đơn hàng”, màn hình “Checkout” sẽ hiển thị và thông tin là các thông tin của sản phẩm vừa chọn, thông tin tài khoản, địa chỉ mặc định. * Để tính tổng tiền ta lấy giá tiền thông qua tổng tiền các sản phẩm, và tiền lấy qua API từ bên vận chuyển, API từ bên thanh toán * Tổng tiền để gửi đi lên DB sẽ không bao gồm tiền từ API thanh toán.   **\*Để biết được sẽ lấy API lại từ đơn vị vận chuyển (selectedDelivery) hay đơn vị thanh toán (selectedPayment) nào thì cần phải chọn đơn vị thanh toán và vận chuyển trước.**  **\*Để có thể chọn voucher, thực hiện Click vào gridview của voucher để mở màn hình “List Voucher (Customer)” để tiến hành chọn các voucher. Lưu ý: có thể chọn nhiều voucher, nhưng với mỗi sản phẩm đang mua hiện tại, không thể có cùng lúc 2 voucher tác động lên.**  **=> Hàm check: checkIsEffectAlready():**   * **Foreach (Product product in listProduct) if (product.IsInMoreThanOne(listVoucher)) => return true**   **Return false**  Để thực hiện thanh toán, thực hiện ấn chọn nút “Đặt hàng” trên màn hình “Checkout” để đặt hàng   * If (Click “Đặt hàng”) * If (checkIsEffectAlready()) => nhận MSG30 * PushtoDB(new Order(thisAccount, listProduct, deliveryUnit, paymentUnit, status = “WAITING”)) * Back() * Khi ấn chọn button “Đặt hàng” hệ thống sẽ đưa đơn hàng lên database và quay lại màn hình main screen. |

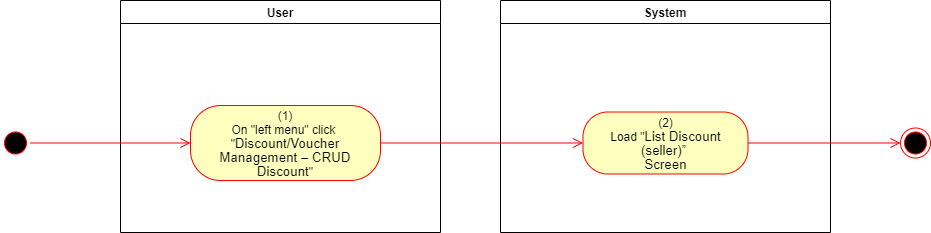
### UC38: CRUD khuyến mãi

|  |  |
| --- | --- |
| **Name** | **CRUD khuyến mãi** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc khuyến mãi |
| **Actor** | 1. User 2. System |
| **Trigger** | * Khi người dùng mở mục “Discount/Voucher Management – CRUD Discount” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD khuyến mãi theo yêu cầu |

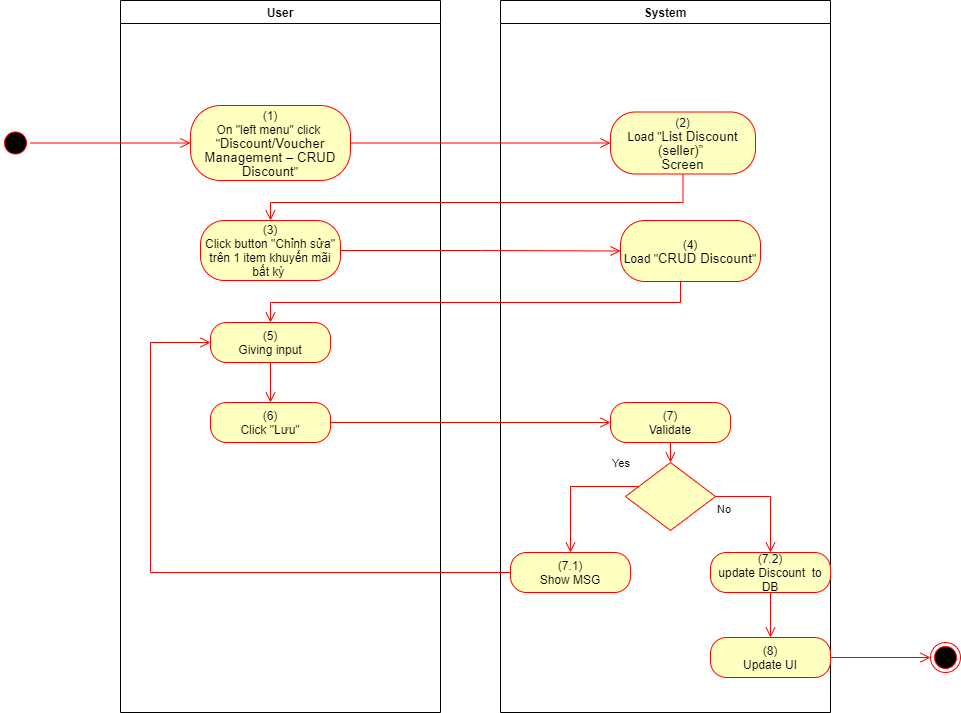
#### Activities Flow



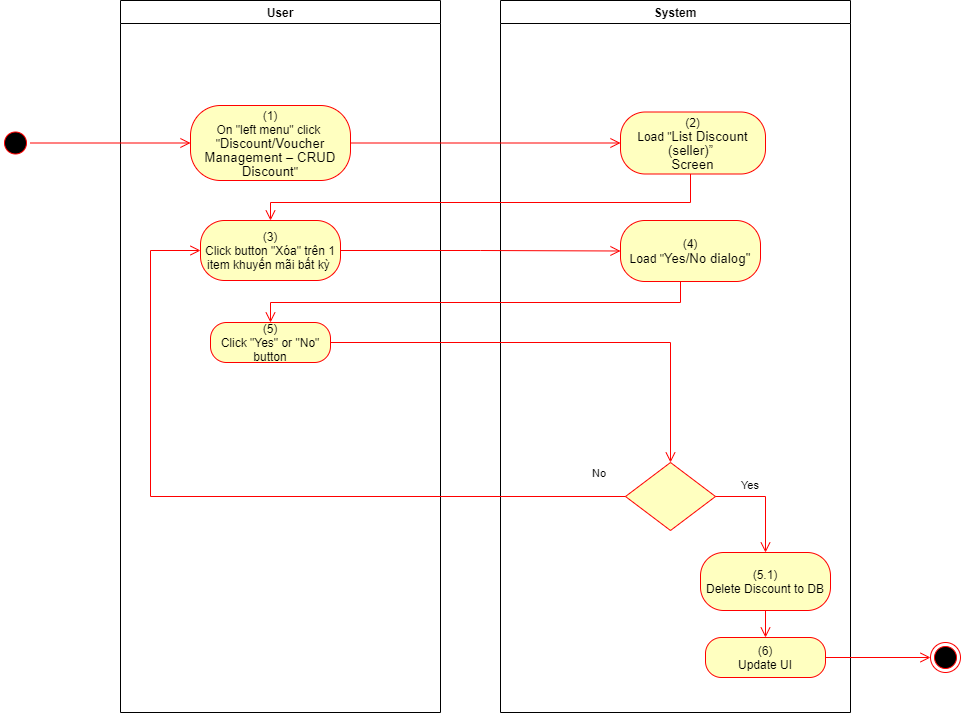
*Figure 82: Create discount Activities Flow*

**

*Figure 83: Read discount Activities Flow*

**

*Figure 84: Update discount Activities Flow*

**

*Figure 85: Delete discount Activities Flow*

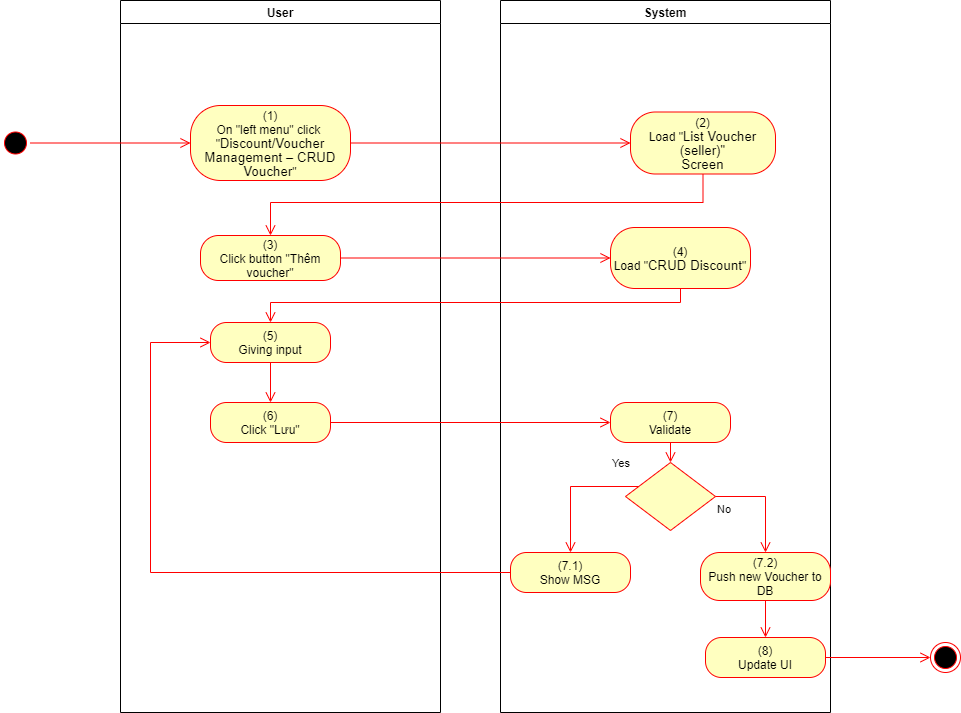
#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Discount (Seller)” khi người dùng chọn mục “Discount/Voucher Management – CRUD Discount” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Discount/Voucher Management – CRUD Discount”, hệ thống thực hiện tải màn hình “List Discount (seller)” bao gồm các khuyến mãi đã được tạo (listDiscount):   * If (Click “Discount/Voucher Management – CRUD Discount”) => open(“List Discount (seller)”) * listDiscount = getListFromDB(“Selec \* from Discount”)  1. **Create**   Để tiến hành thêm khuyến mãi, ta thực hiện ấn chọn button “Thêm khuyến mãi” trên màn hình “List Discount (Seller)”, lúc này form “CRUD Discount” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm khuyến mãi”) => open(“CRUD Discount”) * nameDiscount = “” * startDate = NULL * endDate = NULL * percent = “” * listIndustry = NULL * listKind = NULL * listProduct = NULL * isAll = false * Khi click button “Thêm khuyến mãi” tiến hành mở “CRUD Discount” và set các thông số được nhập là rỗng.   Để tiến hành lưu khuyến mãi ta thực hiện nhân chọn button “Lưu” trong màn hình “List Discount (Seller)” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameDiscount == “” || startDate == NULL || endDate == NULL || percent == “” || (listIndustry == NULL && listKind == NULL && listProduct == NULL && isAll == false)) => nhận MSG01. * Foreach(Product item in listProduct) => if (item.isIn(AnyIn(listKind)) || item.isIn(AnyIn(listIndustry))) => nhận MSG27 * Foreach(Kind item in listKind) => if (item.isIn(AnyIn(listIndustry))) => nhận MSG28 * Else => newDiscount = new Discount (nameDiscount, startDate, endDate, percent, listIndustry, listKind, listProduct, isAll) => pushToDB(newDiscount), listDisplay.add(newDiscount) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Đối với chọn ngành loại, sản phẩm sẽ áp dụng khuyến mãi, cả 3 không được phép rỗng cùng lúc nếu isAll (chọn khuyến mãi tất cả) đang bằng false. * Ngành hàng, loại hàng, hàng hóa đã nhập không được phép bao trùm nhau. * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Discount”  1. **Update**   Để tiến hành cập nhật khuyến mãi, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item khuyến mãi bất kỳ (SelectedDiscount) trên màn hình “List Discount (Seller)”, lúc này form “CRUD Discount” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Discount”) * nameDiscount = SelectedDiscount.nameDisscount * startDate = SelectedDiscount.startDate * endDate = SelectedDiscount.endDate * percent = SelectedDiscount.percent * listIndustry = SelectedDiscount.listIndustry * listKind = SelectedDiscount.listKind * listProduct = SelectedDiscount.listProduct * isAll = SelectedDiscount.listProduct * Khi click button “Chỉnh sửa” tiến hành mở “CRUD Discount” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành lưu khuyến mãi ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameDiscount == “” || startDate == NULL || endDate == NULL || percent == “” || (listIndustry == NULL && listKind == NULL && listProduct == NULL && isAll == false)) => nhận MSG01. * Foreach(Product item in listProduct) => if (item.isIn(AnyIn(listKind)) || item.isIn(AnyIn(listIndustry))) => nhận MSG27 * Foreach(Kind item in listKind) => if (item.isIn(AnyIn(listIndustry))) => nhận MSG28. * Else => updateDiscounty = new Discount (nameDiscount, startDate, endDate, percent, listIndustry, listKind, listProduct, isAll)=> pushToDB(updateDiscount), listDisplay.update(updateDiscount) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Đối với chọn ngành loại, sản phẩm sẽ áp dụng khuyến mãi, cả 3 không được phép rỗng cùng lúc nếu isAll (chọn khuyến mãi tất cả) đang bằng false. * Ngành hàng, loại hàng, hàng hóa đã nhập không được phép bao trùm nhau.  1. **Delete**   Để tiến hành xóa khuyến mãi, ta thực hiện ấn chọn button “Xóa” của một Item khuyến mãi bất kỳ (SelectedDiscount) trên màn hình “List Discount (Seller)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * Else => listDisplay.delete(selectedDiscount) => deleteOnDB(selectedDiscount) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database nếu hiện tại không có sản phẩm nào thuộc khuyến mãi đó. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Discount (seller)”   **\*Không áp dụng khuyến mãi chồng lên một sản phẩm, nếu có khuyến mãi chồng đang diễn ra sẽ lấy cái có % cao hơn. Việc biểu diễn có khuyến mãi sẽ được áp dụng trục tiếp vào giá sản phẩm, sản phẩm được áp dụng sẽ có một UI khác [~~giá cũ~~ giá mới]**   * When load listProduct() * listCurrentDiscount = “Select \* From Discount Where discount.startDate > DateTime.Now() and discount.endDate < DateTime.Now()” * listCurrentDiscount.ToAllProdcutID() [hàm đưa tất cả id loại hàng, ngành hàng thành id product] * forearch(Product product in listProduct) * if (product.isInMoreThanOne(listDiscount)) => product.applyHigher(listDiscount), product.loadDiscountUI() * else product.apply(listDiscount), product.loadDiscountUI() |

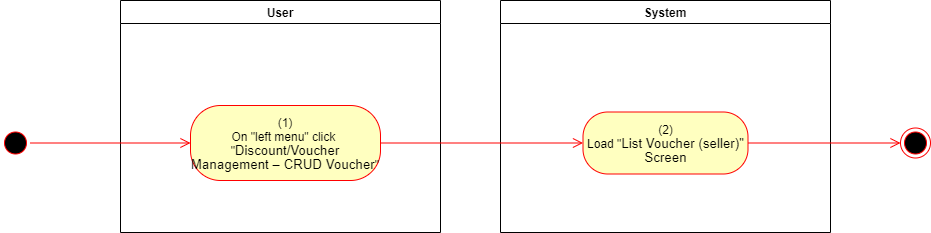
### UC39: CRUD voucher

|  |  |
| --- | --- |
| **Name** | **CRUD voucher** |
| **Description** | Hỗ trợ thực hiện các thao tác thêm, sửa, xóa, đọc voucher |
| **Actor** | 1. BGD 2. Phòng kinh doanh |
| **Trigger** | * Khi người dùng mở mục “Discount/Voucher Management – CRUD voucher” |
| **Pre-condition** | Đăng nhập với tư cách Phòng kinh doanh hoặc BGD. |
| **Post-condition** | * CRUD voucher theo yêu cầu |

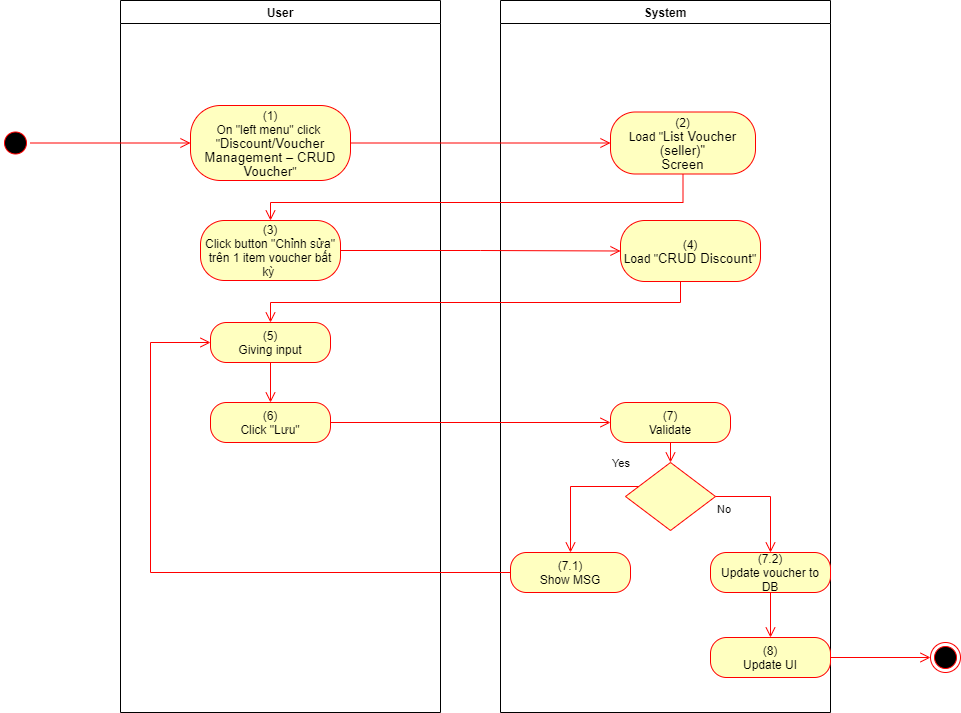
#### Activities Flow



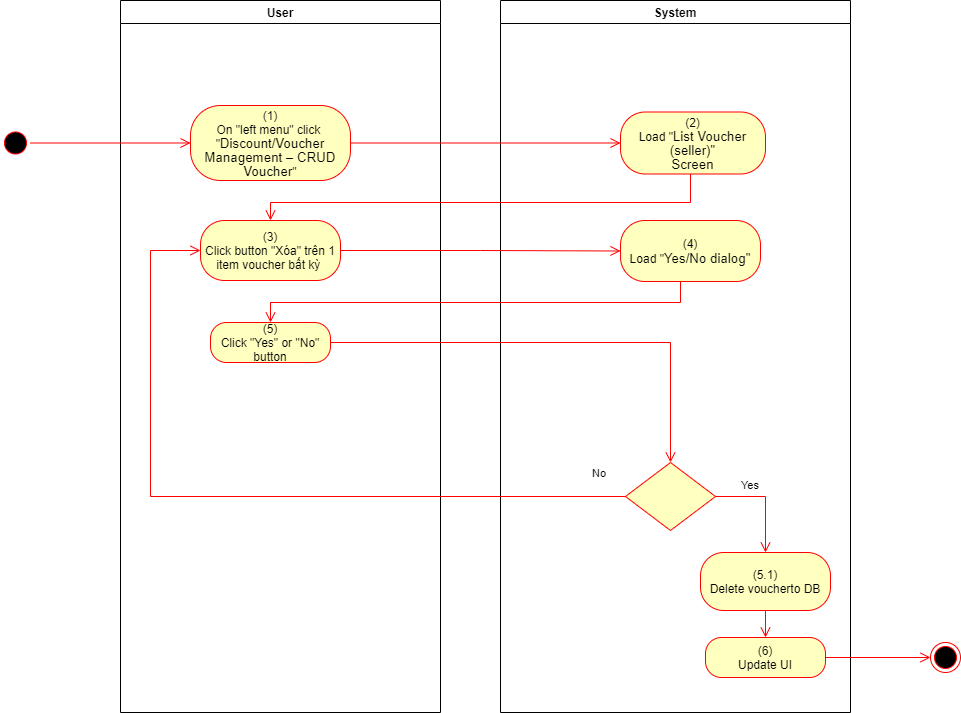
*Figure 86: Create voucher Activities Flow*

**

*Figure 87: Read voucher Activities Flow*

**

*Figure 88: Update voucher Activities Flow*

**

*Figure 89: Delete voucher Activities Flow*

#### Business Rules

|  |  |  |
| --- | --- | --- |
| **Activity** | **BR Code** | **Description** |
| *(1)* | *BR1* | **Tải màn hình:**   * Hệ thống tải màn hình “List Voucher (Seller)” khi người dùng chọn mục “Discount/Voucher Management – CRUD Discount” |
| *(2)* | *BR2* | **Business Rules:**   1. **Read**   Khi người dùng tiến hành nhân chọn mục “Discount/Voucher Management – CRUD Voucher”, hệ thống thực hiện tải màn hình “List Voucher (seller)” bao gồm các Voucher đã được tạo (list Voucher):   * If (Click “Discount/Voucher Management – CRUD Voucher”) => open(“List Voucher (seller)”) * list Voucher = getListtFromDB(“Selec \* from Voucher”)  1. **Create**   Để tiến hành thêm Voucher, ta thực hiện ấn chọn button “Thêm voucher” trên màn hình “List Voucher (Seller)”, lúc này form “CRUD Discount” sẽ hiển thị và người dùng tiến hành nhập thông tin cần thiết   * If (Click “Thêm voucher”) => open(“CRUD Voucher”) * name Voucher = “” * startDate = NULL * endDate = NULL * percent = “” * listIndustry = NULL * listKind = NULL * listProduct = NULL * isAll = false * amount = “” * Khi click button “Thêm Voucher” tiến hành mở “CRUD Voucher” và set các thông số được nhập là rỗng.   Để tiến hành lưu Voucher ta thực hiện nhân chọn button “Lưu” trong màn hình “List Voucher (Seller)” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (name Voucher == “” || startDate == NULL || endDate == NULL || percent == “” || (listIndustry == NULL && listKind == NULL && listProduct == NULL && isAll == false) || amount == “”) => nhận MSG01. * If (amount < 0) => MSG29 * Foreach(Product item in listProduct) => if (item.isIn(AnyIn(listKind)) || item.isIn(AnyIn(listIndustry))) => nhận MSG27 * Foreach(Kind item in listKind) => if (item.isIn(AnyIn(listIndustry))) => nhận MSG28 * Else => new Voucher = new Voucher (name Voucher, startDate, endDate, percent, listIndustry, listKind, listProduct, isAll, amount) => pushToDB(new Voucher), listDisplay.add(new Voucher) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Đối với chọn ngành loại, sản phẩm sẽ áp dụng Voucher, cả 3 không được phép rỗng cùng lúc nếu isAll (chọn Voucher áp dụng cho tất cả sản phẩm) đang bằng false. * Ngành hàng, loại hàng, hàng hóa đã nhập không được phép bao trùm nhau. * If (Click “Thoát”) => Back() * Nếu nhấn chọn “thoát” thì sẽ không có gì được lưu lại dù có sai hay không, và quay lại màn hình “List Voucher (seller)”  1. **Update**   Để tiến hành cập nhật Voucher, ta thực hiện ấn chọn button “Chỉnh sửa” của một Item Voucher bất kỳ (Selected Voucher) trên màn hình “List Voucher (Seller)”, lúc này form “CRUD Voucher” sẽ hiển thị và người dùng tiến hành chỉnh sửa thông tin   * If (Click “Chỉnh sửa”) => open(“CRUD Voucher”) * nameVoucher = SelectedVoucher.name Voucher * startDate = SelectedVoucher.startDate * endDate = SelectedVoucher.endDate * percent = SelectedVoucher.percent * listIndustry = SelectedVoucher.listIndustry * listKind = SelectedVoucher.listKind * listProduct = SelectedVoucher.listProduct * isAll = SelectedVoucher.listProduct * amount = SelectedVoucher.amount * Khi click button “Chỉnh sửa” tiến hành mở “CRUD Discount” và set các thông số được nhập là thông số của item đã chọn.   Để tiến hành lưu khuyến mãi ta thực hiện nhân chọn button “Lưu” để tiến hành kiểm tra định dạng thông tin và đưa lên trên database.   * If (Click “Lưu”) * If (nameVoucher == “” || startDate == NULL || endDate == NULL || percent == “” || (listIndustry == NULL && listKind == NULL && listProduct == NULL && isAll == false) || amount == “”) => nhận MSG01. * If (amount < 0) => MSG29 * Foreach(Product item in listProduct) => if (item.isIn(AnyIn(listKind)) || item.isIn(AnyIn(listIndustry))) => nhận MSG27 * Foreach(Kind item in listKind) => if (item.isIn(AnyIn(listIndustry))) => nhận MSG28. * Else => update Voucher = new Voucher (nameVoucher, startDate, endDate, percent, listIndustry, listKind, listProduct, isAll, amount)=> pushToDB(update Voucher), listDisplay.update(update Voucher) * Khi ấn chọn lưu hệ thống tiến hành kiểm tra định dạng các thông tin nếu không có bất kỳ thông điệp MSG báo lỗi nào thì hệ thống sẽ cập nhật lại trên database và UI (listDisplay). * Đối với chọn ngành loại, sản phẩm sẽ áp dụng Voucher, cả 3 không được phép rỗng cùng lúc nếu isAll (chọn voucher áp dụng tất cả sản phẩm) đang bằng false. * Ngành hàng, loại hàng, hàng hóa đã nhập không được phép bao trùm nhau.  1. **Delete**   Để tiến hành xóa Voucher, ta thực hiện ấn chọn button “Xóa” của một Item Voucher bất kỳ (SelectedDiscount) trên màn hình “List Voucher (Seller)”, lúc này một Yes/No dialog sẽ hiển thị và hỏi liệu có chắc chắn muốn xóa:   * If (Click “xóa”) * If (Click “Yes”) * Else => listDisplay.delete(selected Voucher) => deleteOnDB(selected Voucher) * Khi click button “xóa”sẽ thực hiện xóa ở cả UI và database nếu hiện tại không có sản phẩm nào thuộc khuyến mãi đó. * If (Click “No”) => Back() * Không làm gì mà quay lại form “List Voucher (seller)” |

# Danh sách mô tả



# Mô tả hiển thị (view):



# Yêu cầu phi chức năng

## User Access and Security

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Group**  **Function / Data** | **PSS User** | **PSS System** | **Delivery unit** | **Payment unit** | **Non-user** |
| **Account Management** |  |  |  |  |  |
| Đăng ký (cho nhân viên) | X | X |  |  |  |
| Đăng ký (cho khách hàng) | X | X |  |  | X |
| Đăng nhập | X | X |  |  | X |
| Quên mật khẩu | X | X |  |  | X |
| **HR Management** |  |  |  |  |  |
| Tuyển dụng | X | X |  |  |  |
| Tính lương | X | X |  |  |  |
| Sa thải | X | X |  |  |  |
| **Message Management** |  |  |  |  |  |
| Nhắn tin giữa các phòng ban | X | X |  |  |  |
| Nhắn tin giữa bên bán và bên mua | X(\*) | X |  |  | X(R) |
| **CRUD Delivery** |  |  |  |  |  |
| CRUD đơn vị vận chuyển | X | X |  |  |  |
| **CRUD Seller Information** |  |  |  |  |  |
| CRUD Thông tin người bán | X | X |  |  |  |
| Xem thông tin người bán | X | X |  |  | X |
| **CRUD Advertising** |  |  |  |  |  |
| CRUD quảng cáo | X | X |  |  |  |
| Xem quảng cáo | X | X |  |  | X |
| **Accounting - Finance** |  |  |  |  |  |
| CRUD nghiệp vụ kế toán | X | X |  |  |  |
| CRUD định khoản kế toán | X | X |  |  |  |
| Lập bảng cân đối tài khoản | X | X |  |  |  |
| Lập bảng cân đối kế toán | X | X |  |  |  |
| Lập báo cáo kết quả hoạt động kinh doanh | X | X |  |  |  |
| **Product Management** |  |  |  |  |  |
| CRUD ngành hàng | X | X |  |  | X(R) |
| CRUD Loại hàng | X | X |  |  | X(R) |
| CRUD thuộc tính khách biệt nội sản phẩm | X | X |  |  | X(R) |
| CRUD hàng hóa | X | X |  |  | X(R) |
| Đánh giá | X | X |  |  |  |
| Thêm xóa giỏ/hàng | X(\*) | X |  |  |  |
| Thêm xóa wishlist | X(\*) | X |  |  |  |
| **CRUD Payment Unit** |  |  |  |  |  |
| CRUD đơn vị thanh toán (người bán) | X | X |  |  |  |
| CRUD đơn vị thanh toán (khách hàng) | X(\*) | X |  | X |  |
| **Customer Management** |  |  |  |  |  |
| CRUD thông tin cá nhân | X(\*) | X |  |  |  |
| CRUD địa chỉ giao hàng | X(\*) | X |  |  |  |
| Tính điểm khách hàng |  | X |  |  |  |
| Thêm/xóa khỏi blacklist | X | X |  |  |  |
| **Order Management** |  | X |  |  |  |
| Xác nhận đơn hàng | X | X |  |  |  |
| Xác nhận đơn hàng (giao bên vận chuyển) | X | X |  |  |  |
| Hủy đơn hàng (bên bán) | X | X |  |  |  |
| Hủy đơn hàng (khách hàng) | X(\*) | X |  |  |  |
| Tạo đơn hàng (khách hàng) | X(\*) | X | X | X |  |
| **Discount Management** |  | X |  |  |  |
| CRUD khuyến mãi | X | X |  |  |  |
| CRUD voucher | X | X |  |  | X(R) |

X: Người dùng có toàn quyền.

X(\*): Người dùng chỉ được tác động lên những thứ của người đó.

X(R): chỉ được phép xem (read).

## Performance Requirements

**Số lượng người dùng**

* Số lượng người dùng đồng thời: 100 ~ 200 người dùng
* Số lượng người dùng tổng thể: 2000 ~ 3000 người dùng

**Dung lượng dữ liệu**

* Số lượng tài liệu: 5MB ~ 15 file
* Data growth rate: 5MB/ ngày

**Cấp độ sẵn sàng**

Cấp độ sẵn sàng cần thiết cho ứng dụng này là: 24\*7

**Tần suất sử dụng**

Tần suất sử dụng: hàng ngày

## Yêu cầu triển khai (implementation)

**Vị trí**

Vị trí để tiến hành deploy ứng dụng sẽ là ở Việt Nam.

**Read-only Duration**

Từ 4.5-6 giờ.

**Read-only Timeframe**

0 giờ sáng (AM).

**Maintenance Window**

Phần mềm sẽ được bảo trì mỗi 1 tháng.

**Overall conversion timeline**

* Mỗi cuối tháng tức ngày 28 29 30 31 tùy tháng sẽ thực hiện tính lương tự động.
* Mỗi 5 giây sẽ cập nhật điểm khách hàng và trạng thái đơn hàng.

**UI Wireframe:**

Link:<https://www.figma.com/file/xMjX9hO2chNBWavAUoKulw/PSS-System?node-id=0%3A1>

# Các yêu cầu khác

## Archive Function

|  |  |  |
| --- | --- | --- |
| **List** | **Actor** | **Condition** |
|  |  |  |

## Security Audit Function

Enable Security Audit Function for BGD to tracking any modification on user’s permission.

# Các yêu cầu hệ thống

## Sites

|  |  |  |
| --- | --- | --- |
| **#** | **Site Name** | **Description** |
| *1* | ***pss.[store].com*** | * Sites để kết nối API đến ứng dụng (Truyền tên cửa hàng vào store) |

## Lists

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **List Code** | **List Name** | **Description** |
| *1* | ***N/A*** | * N/A | * N/A |

## Permission Levels

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Permission Level** | **Permissions** | **Description** |
| *1* | *BGD* | * Toàn quyền |  |
| *2* | *Nhân viên phòng ban* | * Có những quyền của riêng phòng ban đó |  |
| *3* | *Khách hàng* | * Tất cả quyền của khách hàng như mua hàng,… |  |
| *4* | *Non-user* | * Đọc các thông tin sản phẩm, quảng cáo |  |

## Groups

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Group Name** | **Permission Level** | **Description** |
| *1* | *User* | * Các quyền được phân theo nhân viên các phòng, BGD, khách hàng | * Bao gồm BGD, nhân viên các phòng ban |
| *2* | *None-user* | * Xem các thông tin về sản phẩm, quảng cáo. | * Bao gồm tất cả người dùng không có tài khoản. |

## Web Parts

|  |  |  |
| --- | --- | --- |
| **#** | **Web Part** | **Description** |
|  | N/A | * N/A |

## Scheduled Agents

There is no scheduled agent implemented in this application.

## Technical Concern

* Quá nhiều Product được tải trong cùng một màn hình “List Product” (cả màn hình của customer lần nhân viên).
* Giải quyết được vấn đề sự đa dạng của hàng hóa (mỗi hàng hóa có các thuộc tính khác nhau).
* Mâu thuẫn trong khi mua hàng (hai người cùng lúc mua một món hàng khi món hàng còn 1 số lượng nhỏ)
* Xử lý link API kết nối với bên đơn vị vận chuyển và thanh toán.
* Lắng nghe được thay đổi từ DB.

# Phụ lục

## Glossary

The list below contains all the necessary terms to interpret the document, including acronyms and abbreviations.

|  |  |
| --- | --- |
| **Term** | **Description** |
| *BR* | **B**usiness **R**ule |
| *JD* | **J**ob **D**escription |
| *DB* | **D**ata**b**ase |
| *MSG* | **M**es**s**a**g**e |
| *UC* | **U**se **C**ase |
| *N/A* | **N**ot **A**vailable or **N**ot **A**pplicable, used to indicate when information in a certain section could not be provided because it does not apply to this application. |
| *UI* | **U**ser **I**nterface |
| *SRS* | **S**oftware **R**equirements **S**pecification |

## Messages

This section describes the details of messages used in business rules e.g. error messages, confirmation messages, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Message Code** | **Message Content** | **Button** | **Need insert name** |
| *MSG 01* | {String} + “không thể bỏ trống” |  | X |
| *MSG 02* | “Tên không thể chứa số hoặc ký tự đặc biệt” |  |  |
| *MSG 03* | {String} + “Không thể chứa gì ngoài số” |  | x |
| *MSG 04* | {String} + “không đúng định dạng” |  | x |
| *MSG 05* | “Email đã tồn tại trong công ty” |  |  |
| *MSG 06* | “Password xác nhận không giống password đã nhập” |  |  |
| *MSG 07* | “Tài khoản hoặc mật khẩu không tồn tại” |  |  |
| *MSG 08* | “Mã xác thực sai, vui lòng kiểm tra lại” |  |  |
| *MSG 09* | “Ngày bắt đầu không thể nhỏ hơn ngày hiện tại” |  |  |
| *MSG 10* | “Ngày bắt đầu không thể lớn hơn ngày kết thúc” |  |  |
| *MSG 11* | “File không đúng định dạng” |  |  |
| *MSG 12* | “Dữ liệu không khớp hệ thống” |  |  |
| *MSG 13* | “Vui lòng chọn hình ảnh để gửi” |  |  |
| *MSG 14* | “Vui lòng chọn video để gửi” |  |  |
| *MSG 15* | {String} + “không tồn tại” |  | x |
| *MSG 16* | “Trong một nghiệp vụ kinh tế phát sinh tổng Số phát sinh bên Có phải bằng số phát sinh ghi bên Nợ” |  | x |
| *MSG 17* | “Không thể xóa ngành hàng đang có chứa sản phẩm” |  |  |
| *MSG18* | “Không thể xóa loại hàng đang có chứa sản phẩm” |  |  |
| *MSG19* | “Định dạng nhập cho màu sắc không đúng vui lòng nhập #xxxxxx với x là chữ số (không ký tự đặc biệt)” |  |  |
| *MSG20* | “Không thể xóa thuộc tính nội sản phẩm đang áp dụng lên sản phẩm” |  |  |
| *MSG 21* | “Vui lòng chọn đầy đủ thuộc tính” |  |  |
| *MSG 22* | “Không thể thêm vào giỏ hàng số lượng bằng 0” |  |  |
| *MSG 23* | “Sản phẩm chỉ còn {remain} sản phẩm vui lòng điều chỉnh số lượng” |  | x |
| *MSG 24* | “Sản phẩm hiện tại không còn hàng” |  |  |
| *MSG 25* | “Số lượng đã chạm tối đa của sản phẩm” |  |  |
| *MSG26* | “Tài khoản nằm trong blacklist không thể hủy đơn hàng” |  |  |
| *MSG27* | “Có trùng lặp giữa sản phẩm đối với ngành hàng, loại hàng” |  |  |
| *MSG28* | “Có trùng lặp giữa loại hàng đối với ngành hàng” |  |  |
| *MSG29* | “Số lượng không thể âm” |  |  |
| *MSG 30* | “Giá trị không đúng định dạng [Value]///[giá tiền tăng thêm]” |  |  |

## Issues List

N/A