

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



CÔNG NGHỆ PHẦN MỀM

Báo cáo Bài tập lớn Urban waste collection aid - UWC 2.0 Version 1.0

GVHD: Lê Đình Thuận
Mai Đức Trung

SV thực hiện: Nguyễn Ngọc Hùng - 2013368 - L02
Nguyễn Kiều Anh - 200859 - L02
Võ Công Danh - 2010986 - L03
Trịnh Tiến Đạt - 2012959 - L02
Lê Thanh Khải - 2011401 - L03
Nguyễn Hoàng Minh Trí - 2010734 - L03

Tp. Hồ Chí Minh, Tháng 9/2022



Mục lục

1 Task 1	2
1.1 Task 1.1:	2
1.2 Task 1.2:	2
1.3 Task 1.3:	5
2 Task 2	11
2.1 Task 2.1:	11
2.2 Task 2.2:	14
2.3 Task 2.3:	17
3 Task 3:	18
3.1 Task 3.1:	18
3.2 Task 3.2	19
4 Task 4	21
4.1 Task 4.1 & Task 4.2	21
4.2 Task 4.3	21
5 Task 5	32
6 Thay đổi so với phiên bản trước	32

1 Task 1

1.1 Task 1.1:

1.1.1 Bối cảnh của dự án:

Trong bối cảnh việc quản lý chất thải diễn ra tốn kém và không hiệu quả, chúng ta cần cải thiện lại các quá trình thu thập, quản lý chất thải và việc thu thập và quản lý này sẽ được giao cho các tổ chức thu gom chuyên nghiệp, có các dịch vụ quản lý hiệu quả đối với một địa bàn rộng lớn.

Và nhiệm vụ của chúng tôi khi thực hiện dự án này là để xây dựng nên một quy trình để có thể đáp ứng những yêu cầu đặt ra bên trên và phát triển thành một ứng dụng để các bên liên quan có thể dễ dàng sử dụng, quản lí.

1.1.2 Các bên liên quan:

- Tổ chức X: được ký hợp đồng để phát triển hệ thống quản lý được gọi là UWC 2.0 nhằm nâng cao hiệu quả thu gom rác
- Nhà cung cấp dịch vụ Y: là nhà cung cấp dịch vụ thu gom rác, thuê công ty X phát triển hệ thống quản lí UWC 2.0.
- Người dân: người sử dụng dịch vụ của công ty Y.

1.1.3 Nhu cầu hiện tại:

Có một hệ thống hiện tại UWC 1.0 với một cơ sở dữ liệu. UWC 2.0 dự kiến sẽ nhập và sử dụng dữ liệu hiện có từ UWC 1.0. Dự kiến rằng Quản lý tác vụ có thể hoạt động được với UWC 1.0 càng nhiều càng tốt. Hệ thống sẽ có thể xử lý dữ liệu thời gian thực từ ít nhất 1000 MCP tại thời điểm hiện tại và 10.000 MCP trong 5 năm. Giao diện hệ thống UWC 2.0 nên bằng tiếng Việt, có cơ hội chuyển sang tiếng Anh trong tương lai. Nâng cao hiệu quả quản lý thu gom rác, tiết kiệm chi phí nguyên liệu, điều hành hệ thống thu gom rác tối ưu và hiệu quả.

1.1.4 Vấn đề hiện tại:

- Tổ chức X: phát triển phần mềm UWC 2.0, đảm bảo đúng yêu cầu từ khách hàng Y, xây dựng hệ thống phần mềm quản lý với mục tiêu phát triển lâu dài, bền vững.
- Nhà cung cấp dịch vụ Y: việc quản lý thu gom rác chưa hiệu quả, UWC 1.0 chưa đáp ứng được yêu cầu.

1.1.5 Lợi ích của các bên liên quan:

- Tổ chức X: nhận được khoản tiền thỏa đáng cho việc phát triển phần mềm UWC 2.0
- Nhà cung cấp dịch vụ Y: quản lý thu gom rác hiệu quả, công việc đạt hiệu quả và năng suất cao, giảm bớt các chi phí vận hành, đem lại lợi nhuận.
- Người dân: đem lại thuận tiện cho người dân, đảm bảo môi trường khu vực dân cư. Dễ dàng phản ánh, góp ý cho đơn vị thu gom rác.

1.2 Task 1.2:

1.2.1 Những yêu cầu về chứng năng (functional requirements):

1.2.1.a Nhân viên quản lí (back officers):

- Có chức năng hỗ trợ bộ phận quản lý về trạng thái hoạt động của các nhân viên vệ sinh và nhân viên thu gom (lịch làm việc, trạng thái làm việc của họ, vị trí của họ ..).
- Định vị được các phương tiện thu gom đang ở địa điểm nào để có sự điều phối thích hợp. Và xem được các thông số kỹ thuật của chúng (trọng tải, sức chứa, xăng tiêu thụ, lượng rác đã chứa...)
- Ở các địa điểm tập trung rác (MCPs) thì chúng ta tích hợp các công nghệ thu thập dữ liệu (IOT) và gửi tín hiệu cho hệ thống. Từ đó hệ thống sẽ trả dữ liệu về cho người quản lý về trạng thái của các MCPs (khi đầy > 95 %) thì người quản lý sẽ có thể điều phối nhân viên thu gom đến để lấy rác.

- Phân chia xe cho nhân viên vệ sinh và nhân viên thu gom.
- Phân chia điểm tập trung rác cho nhân viên vệ sinh và nhân viên thu gom
- Có thể tạo ra cái tuyến đường hợp lý cho nhân viên thu gom
- Có thể giao tiếp với nhân viên vệ sinh và nhân viên thu gom

1.2.1.b Nhân viên vệ sinh và nhân viên thu gom

- Xem được lịch làm việc của họ
- Xem được công việc hàng ngày và hàng tuần
- Có thể giao tiếp với người quản lý, nhân viên thu gom và nhân viên vệ sinh khác
- Được thông báo khi MCPs đầy
- Báo cáo hoàn thành công việc

1.2.1.c Chủ nguồn rác thải (phát triển thêm)

- Có thể tạo tài khoản, thay đổi 1 số thông tin cá nhân, mật khẩu
- Xem được chi phí liên quan đến dịch vụ dọn rác
- Có chức năng thanh toán
- Xem lịch trình thu gom rác
- Có thể gửi khiếu nại nếu không hài lòng

1.2.2 Những yêu cầu phi chức năng (non-functional requirements):

1.2.2.a Yêu cầu về hiệu năng

- Điểm thu gom rác: cần phải được cập nhật mỗi 15p, khi nào lượng rác 95% thì sẽ báo về nhân viên để có thể kịp thời xử lý kịp lúc.
- Tin nhắn trong hệ thống được gửi đi với thời gian trễ nhỏ hơn 1 giây.
- Hệ thống phải xử lý dữ liệu real-time với ít nhất 1000 điểm tập trung rác ở hiện tại và 10000 trong vòng 5 năm tới.

1.2.2.b Yêu cầu về tính dễ sử dụng và giao diện

- Giao diện dễ dùng, người mới sử dụng có thể dễ dàng thao tác trên đó. Việc thao tác dễ dàng, dễ hiểu trên hệ thống cũng góp phần tiết kiệm thời gian cho người dùng, đạt được hiệu suất công việc hơn.
- Cập nhật dễ dàng

1.2.2.c Yêu cầu về độ tin tưởng

- Phần mềm không có lỗi
- Thông tin hiển thị phải chính xác, cập nhật thông tin nhanh chóng

1.2.2.d Yêu cầu về bảo mật

- Cần bảo mật thông tin người dùng
- Người dùng này không có khả năng vào được tài khoản của người khác
- Có những chức năng giúp người dùng lấy lại được tài khoản khi quên

1.2.2.e Yêu cầu về tính linh hoạt

- Hệ thống phù hợp với bất kì đối tượng nào (người dân, doanh nghiệp,...)
- Chạy được trên các hệ điều hành phổ biến hiện nay như: MacOS, Windows, Android, IOS,...

1.2.2.f Yêu cầu về đa ngôn ngữ

- Hiện tại hệ thống sử dụng Tiếng Việt nhưng phải có khả năng thêm tiếng Anh vào trong tương lai.

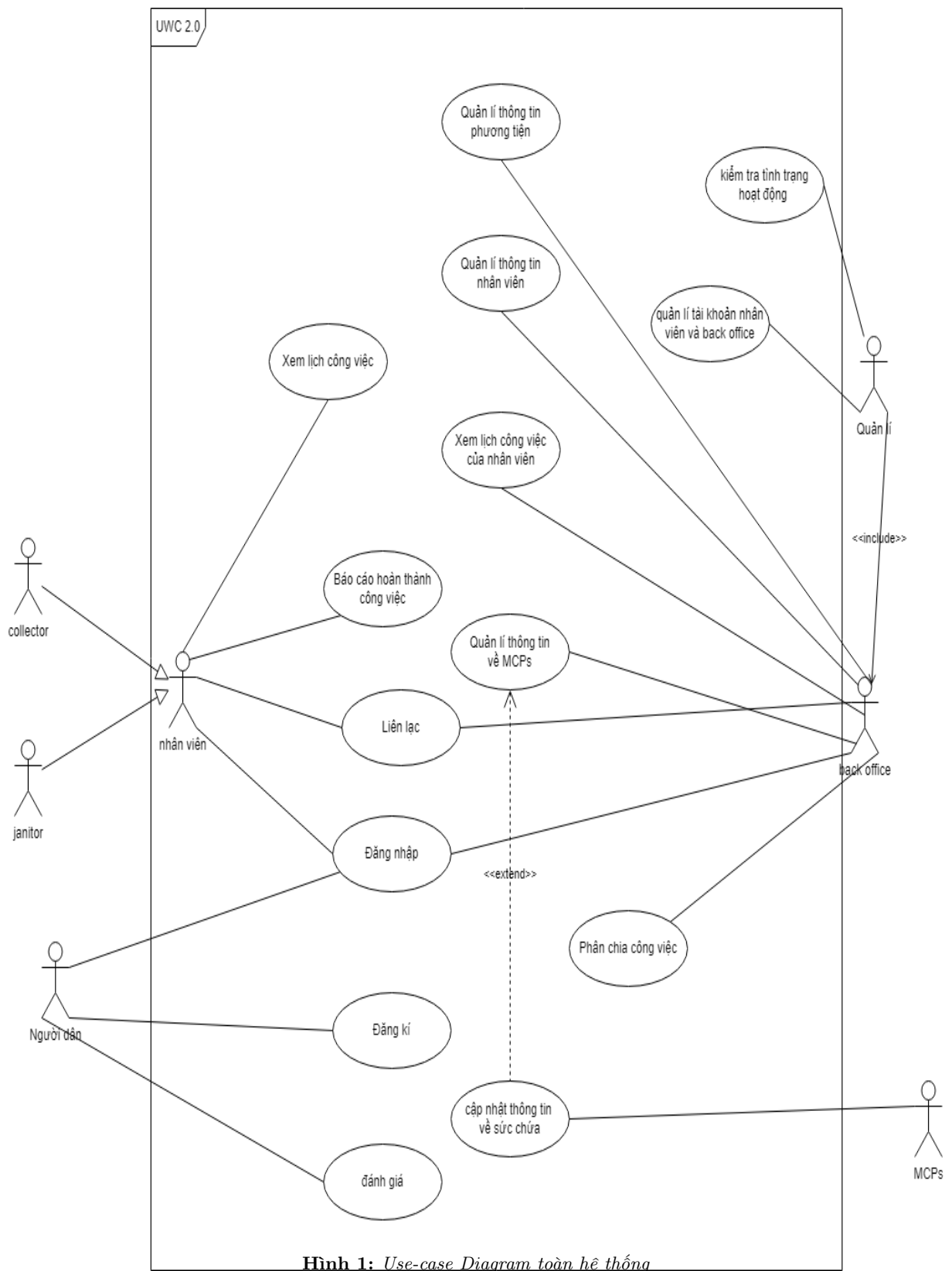
1.2.2.g Yêu cầu về system migration

- Hệ thống UWC 2.0 phải được thiết kế để sử dụng các dữ liệu có sẵn từ hệ thống UWC 1.0 đặc biệt là phân quản lí công việc (task management).

1.2.2.h Yêu cầu về an toàn

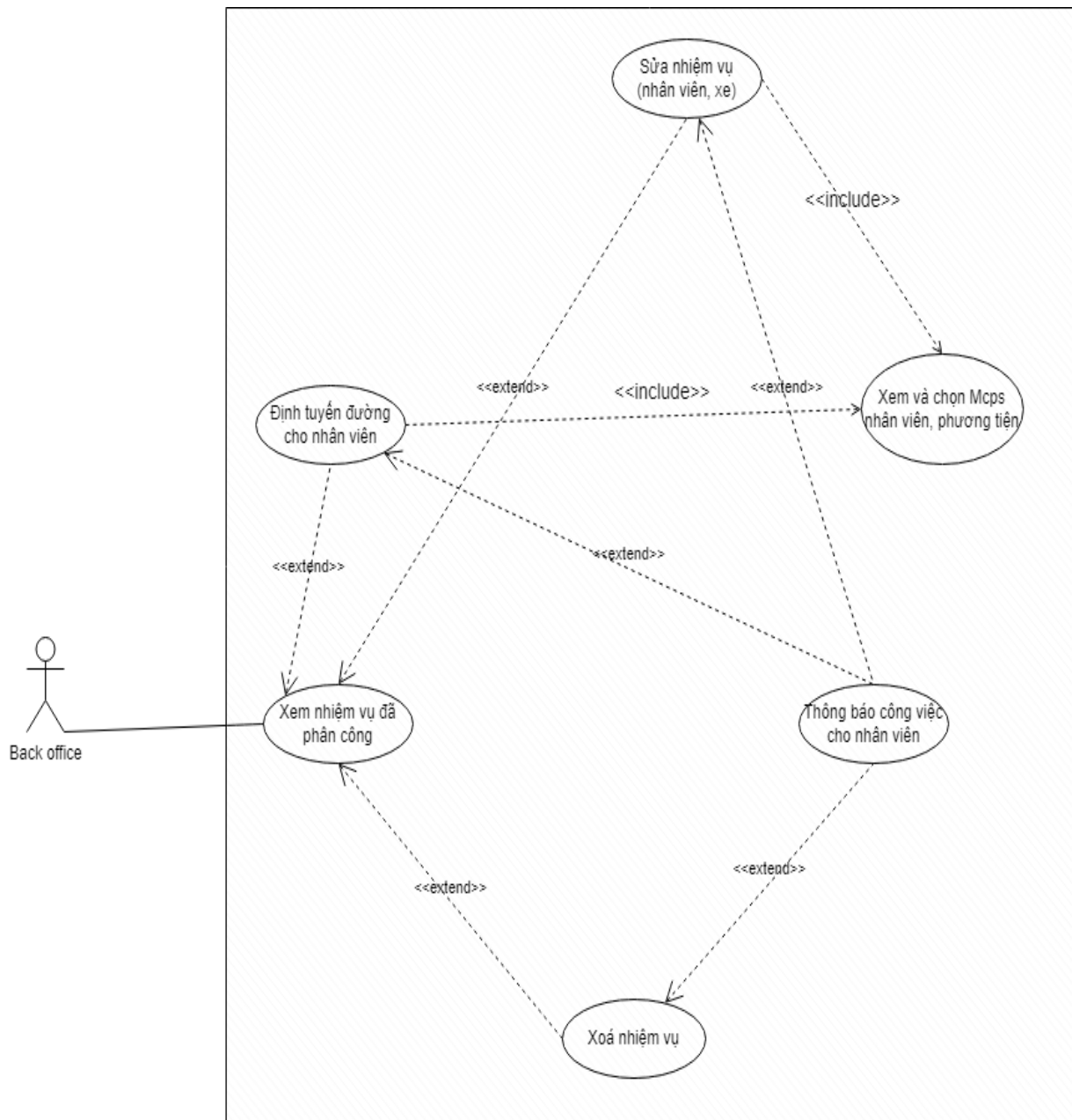
- Tính đảm bảo, giúp cho khi có sự thay đổi từ 1 phía nào đó sẽ không ảnh hưởng đến toàn bộ quy trình.

1.2.3 Use case diagram cho toàn bộ hệ thống:



1.3 Task 1.3:

1.3.1 Biểu đồ use-case diagram module phân công nhiệm vụ (task assignment module):



1.3.2 Mô tả biểu đồ use-case diagram module phân công nhiệm vụ (task assignment):

Xem nhiệm vụ đã phân công

Use-case name :	Xem nhiệm vụ đã phân công
Description :	Back offices xem tuyến đường đã phân công.
Actor(s) :	Back offices
Priority :	
Trigger :	Back offices nhấp vào “xem tuyến đường đã phân công”
Pre-Condition(s) :	Back offices đã đăng nhập vào hệ thống
Post-Condition(s) :	Hệ thống hiển thị các dữ liệu tuyến đường từ database
Normal Flow :	1. Back offices chọn “xem tuyến đường đã phân công ” 2. Hệ thống truy cập vào database và hiển thị tuyến đường và các thông tin liên quan.
Exception Flow:	
Alternative Flow :	
Extension Point :	+ Xoá nhiệm vụ + Định tuyến đường cho nhân viên + Sửa nhiệm vụ (nhân viên, xe)
Business Rules :	
Non-Functional Requirement :	

Định tuyến đường cho nhân viên

Use-case name :	Định tuyến đường cho nhân viên
Description :	Back offices phân công nhiệm vụ cho các nhân viên, định tuyến đường
Actor(s) :	Back offices
Priority :	
Trigger :	Back offices nhấp vào “Định tuyến đường”
Pre-Condition(s) :	<ul style="list-style-type: none"> + Back offices đã đăng nhập vào hệ thống + Back offices đang ở mục “xem tuyến đường đã phân công”
Post-Condition(s) :	<ul style="list-style-type: none"> + Back offices hoàn tất phân công nhiệm vụ cho nhân viên + Hệ thống ghi nhận dữ liệu đã được nhập vào database và thông báo cho nhân viên
Normal Flow :	<ol style="list-style-type: none"> 1. Back offices đang ở mục “xem tuyến đường đã phân công” 2. Back offices chọn “Định tuyến đường” 3. Hệ thống truy cập vào database kiểm tra và hiển thị các MCPs cần dọn ra màn hình 4. Hệ thống kiểm tra, chọn/ xuất các MCPs 5. Hệ thống truy cập vào database, kiểm tra và hiển thị các xe chưa được sử dụng ra màn hình. 6. Back offices chọn xe. 7. Hệ thống truy cập vào database, kiểm tra và hiển thị các nhân viên chưa có công việc ra màn hình. 8. Back offices chọn nhân viên. 9. Hệ thống định tuyến con đường đi tối ưu nhất. 9. Hệ thống ghi nhận dữ liệu mà Back offices đã nhập vào định tuyến đường vào database và thông báo cho nhân viên và quay lại trang chủ.
Exception Flow:	<ul style="list-style-type: none"> + Exception 1: Tại bước 3, nếu số MCPs cần dọn không đủ mức tối thiểu thông báo "số MCP chưa đủ" và quay lại trang chủ + Exception 2: Tại bước 6, nếu không còn nhân viên rảnh thông báo “đã hết nhân viên” và quay lại trang chủ + Exception 3: Tại bước 8, nếu không còn xe thì thông báo “đã hết xe” và quay lại trang chủ + Exception 3: Tại bước 3, nếu kiểm tra các MCPs
Alternative Flow :	
Business Rules :	Tối thiểu 95% các MCPs ở trạng thái đang hoạt động
Non-Functional Requirement :	

Sửa nhiệm vụ cho nhân viên

Use-case name :	Sửa nhiệm vụ (nhân viên, xe)
Description :	Back offices sửa nhiệm vụ đã được phân công cho các nhân viên, định tuyến đường
Actor(s) :	Back offices
Priority :	
Trigger :	Back offices nhấp vào “Sửa nhiệm vụ”
Pre-Condition(s) :	+ Back offices đã đăng nhập vào hệ thống + Back offices đang ở mục “xem tuyến đường đã phân công”
Post-Condition(s) :	+ Back offices hoàn tất chỉnh sửa nhiệm vụ cho nhân viên + Hệ thống ghi nhận dữ liệu đã chỉnh sửa vào database và thông báo cho nhân viên
Normal Flow :	<ol style="list-style-type: none">1. Back offices đang ở mục “xem tuyến đường đã phân công”2. Back offices chọn “Sửa nhiệm vụ”3. Hệ thống truy cập vào database và hiển thị các MCPs ra màn hình4. Back offices chọn tuyến đường muốn sửa5. Hệ thống truy cập vào database và hiển thị các thông tin nhân viên, xe ra màn hình6. Back offices chọn xe/nhân viên muốn sửa.7. Hệ thống truy cập vào database, kiểm tra và hiển thị các xe/nhân viên chưa có công việc ra màn hình.8. Back offices chọn xe/nhân viên thay thế.9. Hệ thống định tuyến con đường đi tối ưu nhất.9. Hệ thống ghi nhận dữ liệu mà Back offices đã sửa và định tuyến đường vào database và thông báo cho nhân viên và quay lại trang chủ.
Exception Flow:	Tại bước 7, nếu không còn nhân viên/xe rảnh thông báo “đã hết nhân viên” hoặc “đã hết xe” và quay lại trang chủ
Alternative Flow :	
Business Rules :	
Non-Functional Requirement :	

Xóa tuyến đường

Use-case name :	Xóa tuyến đường
Description :	Back offices xóa tuyến đường không cần sử dụng
Actor(s) :	Back offices
Priority :	
Trigger :	Back offices nhấp vào “xóa tuyến đường”
Pre-Condition(s) :	+ Back offices đã đăng nhập vào hệ thống + Back offices đang ở mục “xem tuyến đường đã phân công”
Post-Condition(s) :	+ Back offices xóa tuyến đường thành công + Hệ thống ghi nhận dữ liệu đã được nhập vào database và thông báo cho nhân viên
Normal Flow :	<ol style="list-style-type: none"> 1. Back offices đang ở mục “xem tuyến đường đã phân công” 2. Back offices chọn tuyến đường muốn xóa 3. Back offices nhấp vào “xóa tuyến đường” 4. Hệ thống yêu cầu nhập mật khẩu, xác nhận việc xóa tuyến đường 5. Back offices nhập mật khẩu 6. Hệ thống kiểm tra dữ liệu mật khẩu, xác nhận đúng sẽ xóa đi tuyến đường và thông báo thành công 7. Hệ thống thông báo thay đổi công việc cho nhân viên
Exception Flow:	6.1 Mật khẩu sai, hệ thống thông báo và yêu cầu Back offices nhập lại mật khẩu Usecase tiếp tục bước 6
Alternative Flow :	
Business Rules :	Mật khẩu sai lần thứ 6 liên tiếp, tài khoản của Back offices sẽ bị khóa hoạt động 30 phút
Non-Functional Requirement :	

2 Task 2

2.1 Task 2.1:

Mô tả activity diagram:

Hệ thống hiển thị công việc đã được phân công trên màn hình chính. Tại đây Back offices có thể thêm/xóa/sửa các nhiệm vụ:

- Back offices chọn thêm nhiệm vụ:

+ Hệ thống liệt kê và tính toán số lượng các MCPs trống, nếu số lượng MCPs cần dọn đã đủ mức yêu cầu, hệ thống sẽ chọn ra các MCPs cần dọn, liệt kê các xe đang trống ra màn hình cho Back offices chọn.

+ Nếu số lượng MCPs chưa đủ, hệ thống sẽ báo chưa cần đi dọn. Nếu không có xe trống hệ thống sẽ báo chưa có xe và đặt cảnh báo.

+ Sau khi chọn xe, hệ thống sẽ liệt kê các nhân viên đang rảnh, nếu không có nhân viên rảnh hệ thống sẽ báo chưa đủ nhân viên và đưa ra cảnh báo.

+ Back offices chọn nhân viên, sau đó hệ thống sẽ định tuyến đường tối ưu đi qua các MCPs, cập nhật thông tin vào database và thông báo cho các nhân viên.

- Back offices chọn chỉnh sửa nhiệm vụ:

+ Back offices chọn nhiệm vụ muốn chỉnh sửa.

+ Hệ thống sẽ liệt kê danh sách các xe trống cho Back offices chọn nếu không có xe trống hệ thống sẽ báo chưa có xe và đặt cảnh báo.

+ Sau khi chọn xe, hệ thống sẽ liệt kê các nhân viên đang rảnh, nếu không có nhân viên rảnh hệ thống sẽ báo chưa đủ nhân viên và đưa ra cảnh báo.

+ Back offices chọn nhân viên, sau đó hệ thống sẽ định tuyến đường tối ưu đi qua các MCPs, cập nhật thông tin vào database và thông báo cho các nhân viên.

- Back offices chọn xóa nhiệm vụ:

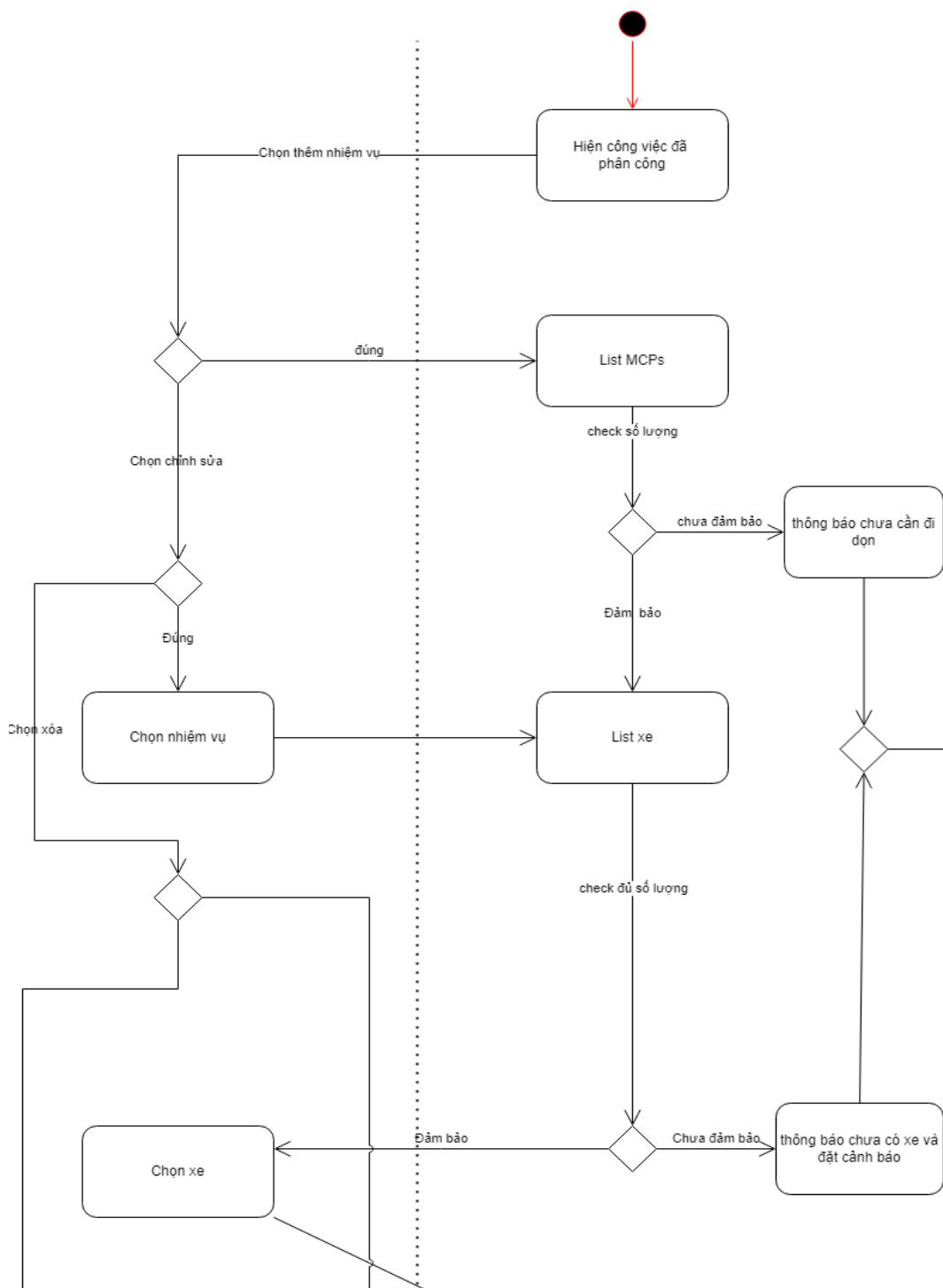
+ Back offices chọn nhiệm vụ muốn xóa, sau đó bấm xóa nhiệm vụ.

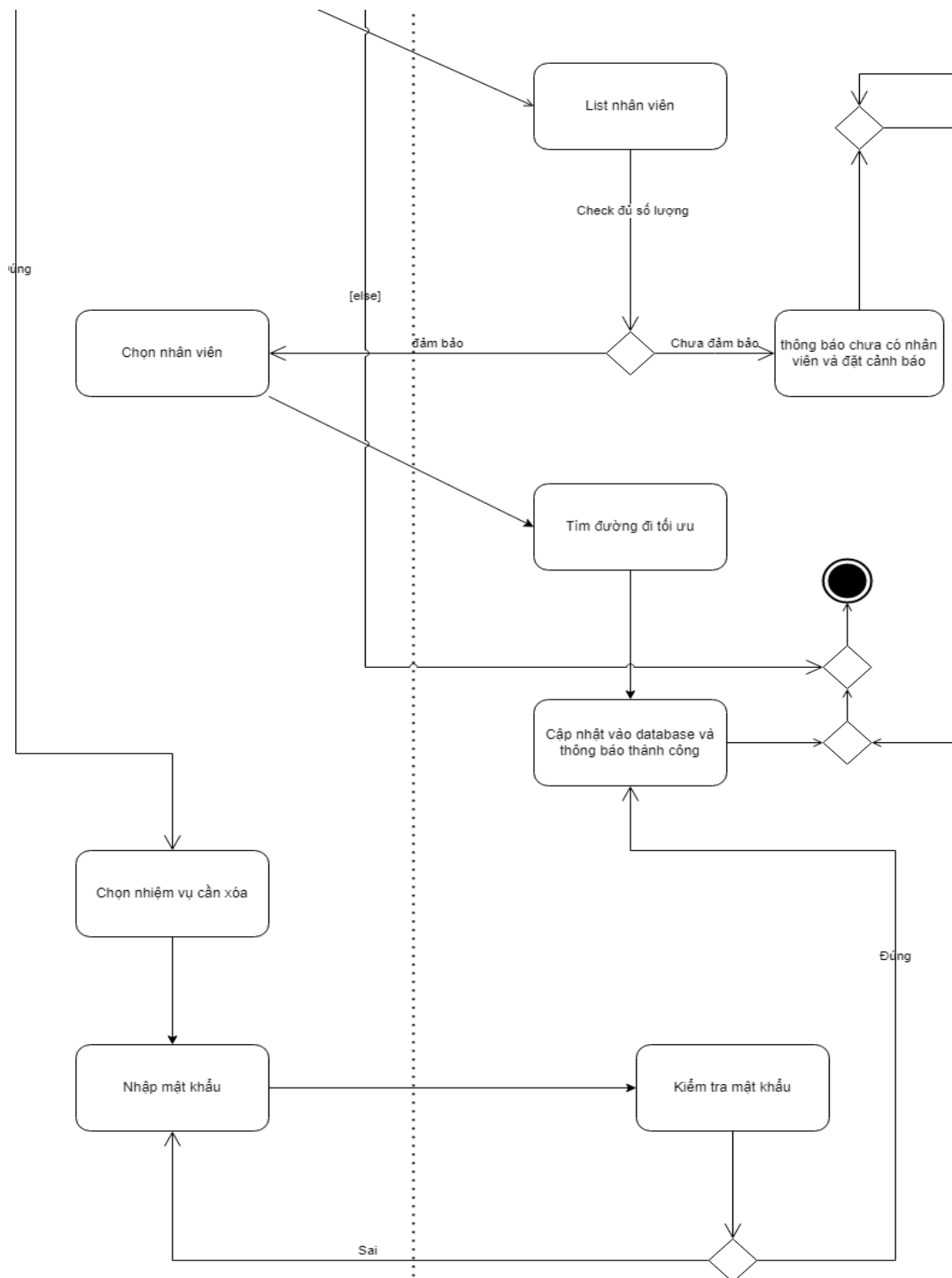
+ Hệ thống sẽ yêu cầu Back offices nhập mật khẩu tài khoản để đảm bảo tính bảo mật, nếu nhập sai mật khẩu, hệ thống sẽ yêu cầu Back offices nhập lại mật khẩu.

+ Xác nhận đúng mật khẩu, hệ thống sẽ cập nhật thông tin vào database và thông báo cho các nhân viên.

Link drive chứa ảnh activity diagram

https://drive.google.com/file/d/1Z6m_eEEvQBWui2B9wrSzucYRVAVY9XdM/view?usp=sharing



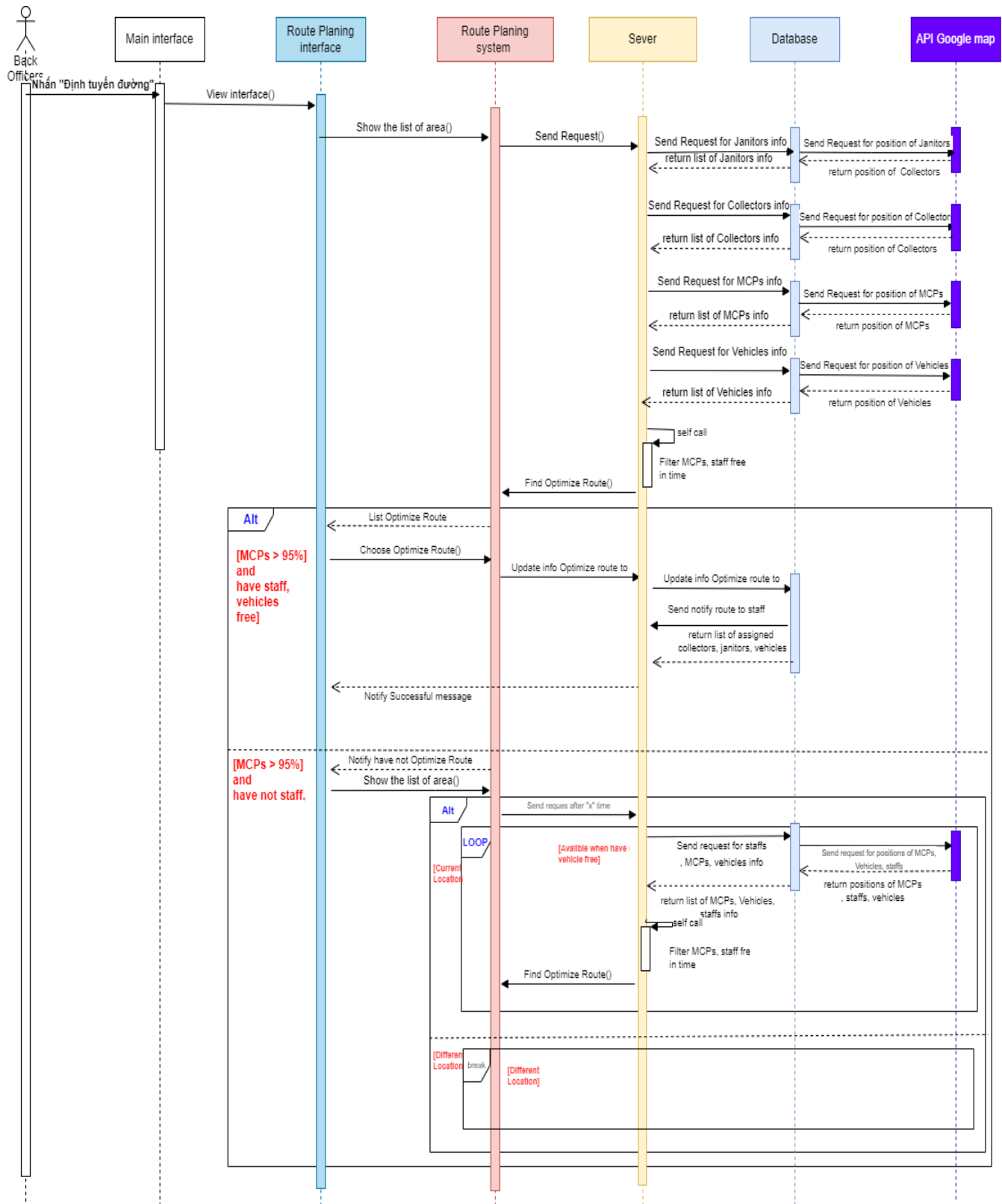


Hình 2: activity diagram

2.2 Task 2.2:

Draw a sequence diagram for use-case in Task 1.3

Feature: Route Planning



Hình 3: *Sequence diagram for feature route planning.*

Link Drive chứa ảnh sequence diagram:

<https://drive.google.com/file/d/1Cx8eh1xvWT-gk6RxFbtsVUyIKIAWeRTZ/view?usp=sharing>

Một số rules trong sequency diagram:

Những rules này dựa trên những rule đã được thiết lập ở task 1.3 và task 2.3 .

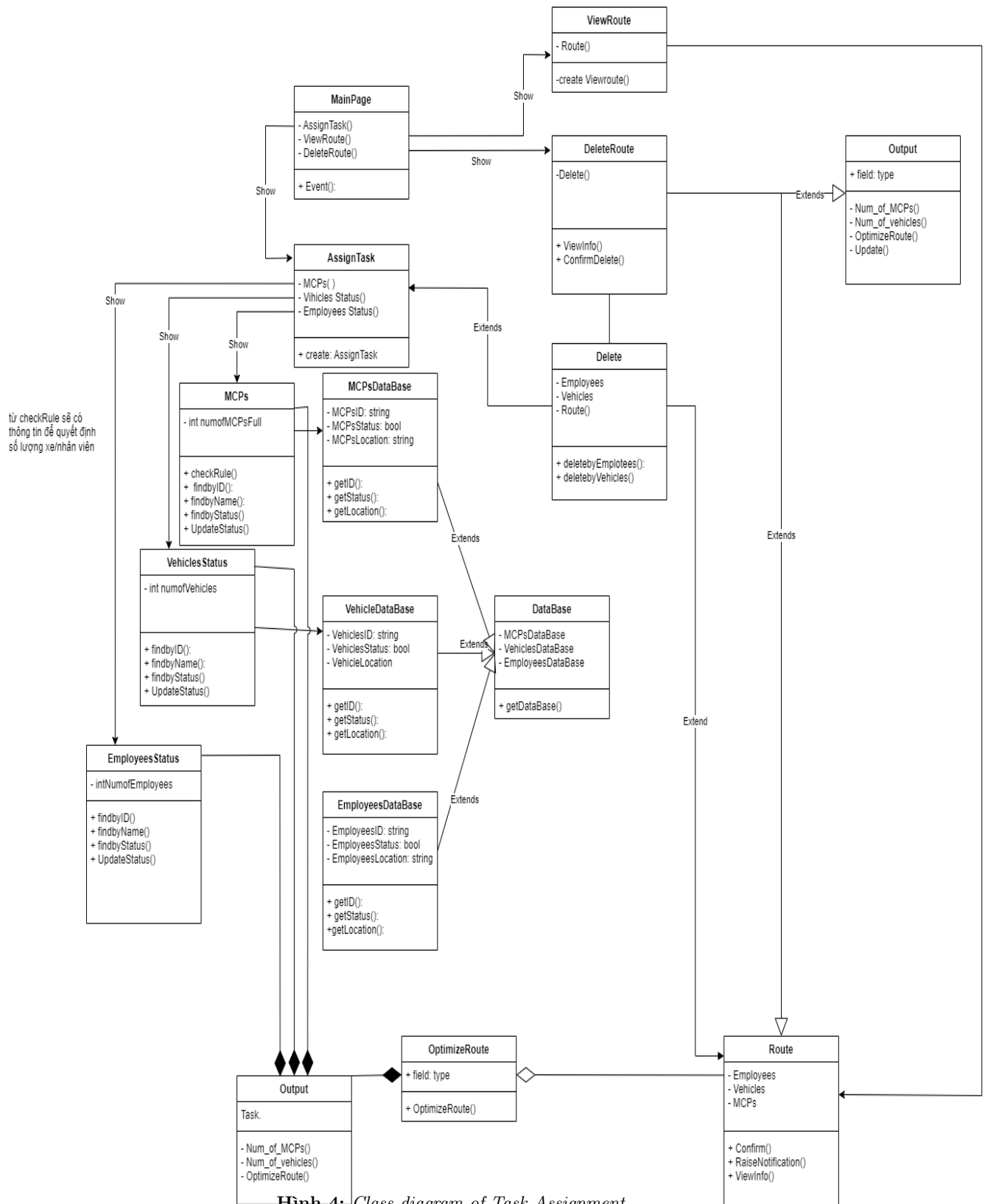
1. Trong Sequence Diagram có dùng trường “staffs = Janitors and collectors”.
2. Giả sử ta sẽ áp dụng hình thức thu gom rác theo chu kỳ, tức là cứ 2 ngày vào một khoảng thời gian nhất định nào đó chúng ta sẽ tiến hành lấy dữ liệu, tính toán để đưa xe đến các điểm MCPs thu gom. Tuy nhiên sẽ có trường hợp là chưa đến giờ thu gom mà số lượng MCPs > 95% trong khu vực đó nhiều (ví dụ 80% số MCPs trong khu vực đó >95% mà chưa đến giờ thu gom rác thì khi đạt tỉ lệ trên, chúng ta sẽ tiến hành điều phối xe rác đến thu).
3. Trong sequency có công đoạn lấy vị trí của các nhân viên (Janitors, Collectors) để định tuyến đường, thì việc này chỉ được thực thi khi có tình huống khẩn cấp như thiếu nhân lực mà chúng ta đang cần giải quyết gấp thì sẽ xác định vị trí nhân viên đó xem có đang ở gần khu vực đó không để có thể điều phối tới). Bình thường trong module "Định tuyến đường" thì chúng ta chỉ cần định các vị trí của các MCPs và các xe rác là đủ.

Mô tả sequency diagram:

- Người dùng thông qua giao diện “Phân công nhiệm vụ” và chọn chức năng “Định tuyến đường”.
- Giao diện gửi thông tin về máy chủ và chuyển sang giao diện phân công tuyến đường (Route planning interface).
- Do dự án của nhóm phát triển được áp dụng cho cả thành phố, giả sử thành phố này có 9 quận, thì giao diện sẽ hiện bảng chọn cho chúng ta chọn một trong các quận đó để tiến hành phân chia tuyến đường. Ví dụ: người dùng chọn quận 1.
- Giao diện sẽ tải lên bản đồ khu vực của quận 1 , và khi ta chọn quận 1 hệ thống sẽ gửi yêu cầu lên server, và từ server gửi yêu cầu lên Database để load các thông tin về trạng thái và vị trí của Janitors, Collectors, MCPs, và phương tiện ở khu vực đó.
- Sever sẽ trả về những dữ liệu mà chúng ta yêu cầu và tại sever, sẽ tiến hành thực thi các hàm lọc để chọn ra những nhân viên , phương tiện đang gánh và MCPs đã vượt mức (>95
- Sau khi lọc ra được những nhân viên trong khu vực đó đang rảnh, các xe sẵn sàng đi thu gom, có MCPs đã vượt mức thì sever sẽ gửi những data đã lọc này và yêu cầu hệ thống phân tuyến đường (Route Planning system: trong đây sẽ hiện thực các giải thuật để phân tuyến đường tối ưu nhất dựa theo API bản đồ từ google map và dữ liệu trạng thái của nhân viên, xe, MCPs vừa được load về từ server) và tiến hành tính toán để tìm ra đường đi tối ưu.
- **[IF MCPs > 95% and have staffs, vehicles free]**
 - Nếu tại thời điểm chúng ta tiến hành đưa xe đi thu tồn tại MCPs > 95% và có nhân viên gánh, có xe đang ở trạng thái có thể di chuyển thì sau khi tính toán , hệ thống phân tuyến đường sẽ trả về danh sách các tuyến đường tối ưu thông qua giao diện (Route Planning system).
 - Ta tiến hành chọn một trong các tuyến đường tối ưu mà hệ thống đề xuất.
 - Hệ thống “Route Planning system” sẽ cập nhật thông tin tuyến đường tối ưu đã chọn lên server, và từ sever cũng update thông tin này lên Database để lưu lại (có thể xem lại nếu cần thiết) .
 - Sau khi cập nhật dữ liệu về tuyến đường tối ưu xong, Database sẽ gửi yêu cầu sever hãy phát đi thông báo về tuyến đường cho nhân viên, để nhân viên biết tiến hành công việc. Và sever cũng gửi lại danh sách những nhân viên được chỉ định cụ thể (nhân viên này sẽ đi đến điểm MCPs A để thu gom cùng với phương tiện B nào đó ...).
 - Sever sẽ gửi thông báo “Định tuyến đường thành công, đã gửi thông báo đến những nhân viên được chỉ định” trên giao diện của back officers.
- **[IF MCPs > 95% and have not staffs, have not vehicles free]**

- Hệ thống sẽ đưa ra thông báo trên giao diện của back officers , không tìm được tuyến đường tối ưu (tức là hiện tại không thể điều phối nhân viên, hay tại thời điểm này chưa thể điều phối xe đi thu gom được).
- Giao diện sẽ hiển thị lại bảng cho ta chọn khu vực thu gom:
- **[Nếu ta chọn Current Location]:** Tức là vẫn phân tuyến đường ở khu vực này (khu vực được chọn lúc ban đầu).
 - * Giao diện sẽ hiện lên để chúng ta chọn khoảng thời gian tiến hành cập nhật lại trạng thái staffs, Vehicles (Send request after “x” time).
 - * Ta sẽ lập những quy trình sau cho đến khi thỏa đủ điều kiện đủ số lượng staffs tối thiểu free và đủ số lượng tối thiểu vehicles free để tiến hành đi thu gom.
 - * Sau khoảng thời gian chúng ta set up ở (Send request after “x” time) sever sẽ gửi những yêu cầu để lấy data từ Database.
 - * Khi Database trả dữ liệu về server thì ta sẽ tiến hành lọc những MCPs, Vehicles free ở thời điểm này.
 - * Tiến hành tìm lại đường đi tối ưu.
 - * Lập lại cho tới khi thỏa điều kiện “Available when have staffs, vehicles free).
- **[Nếu ta chọn Different Location]:** Ta sẽ chuyển sang phân tuyến đường khác trong thành phố.
 - * Hệ thống sẽ break , và quay lại giao diện “Route Planing interface”.

2.3 Task 2.3:

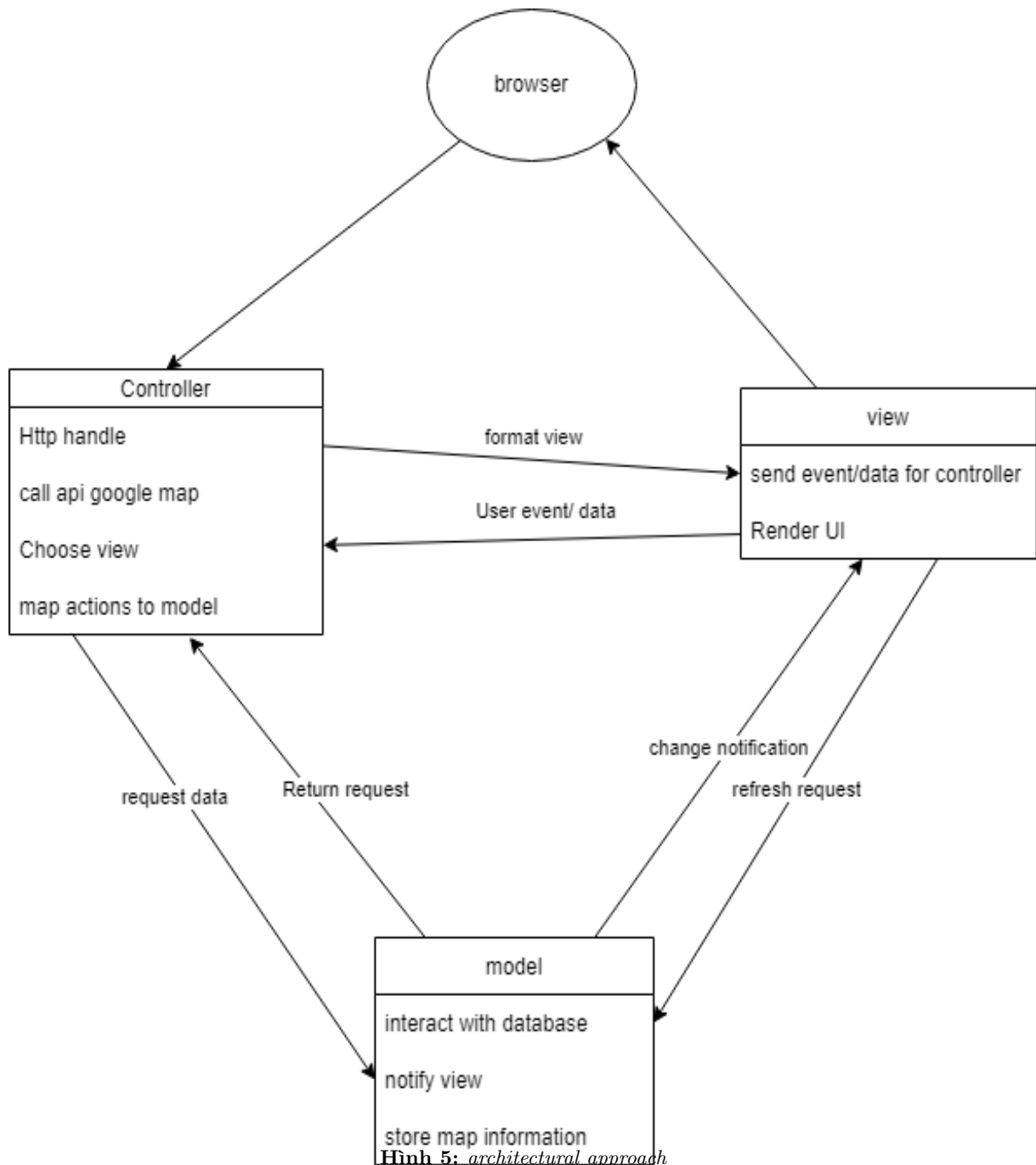


Link Drive chứa ảnh : Class diagram of Task Assignment.

<https://drive.google.com/file/d/170PEFiZg7Ec1bQhP05z6nYn9Xk4059FE/view?usp=sharing>

3 Task 3:

3.1 Task 3.1:

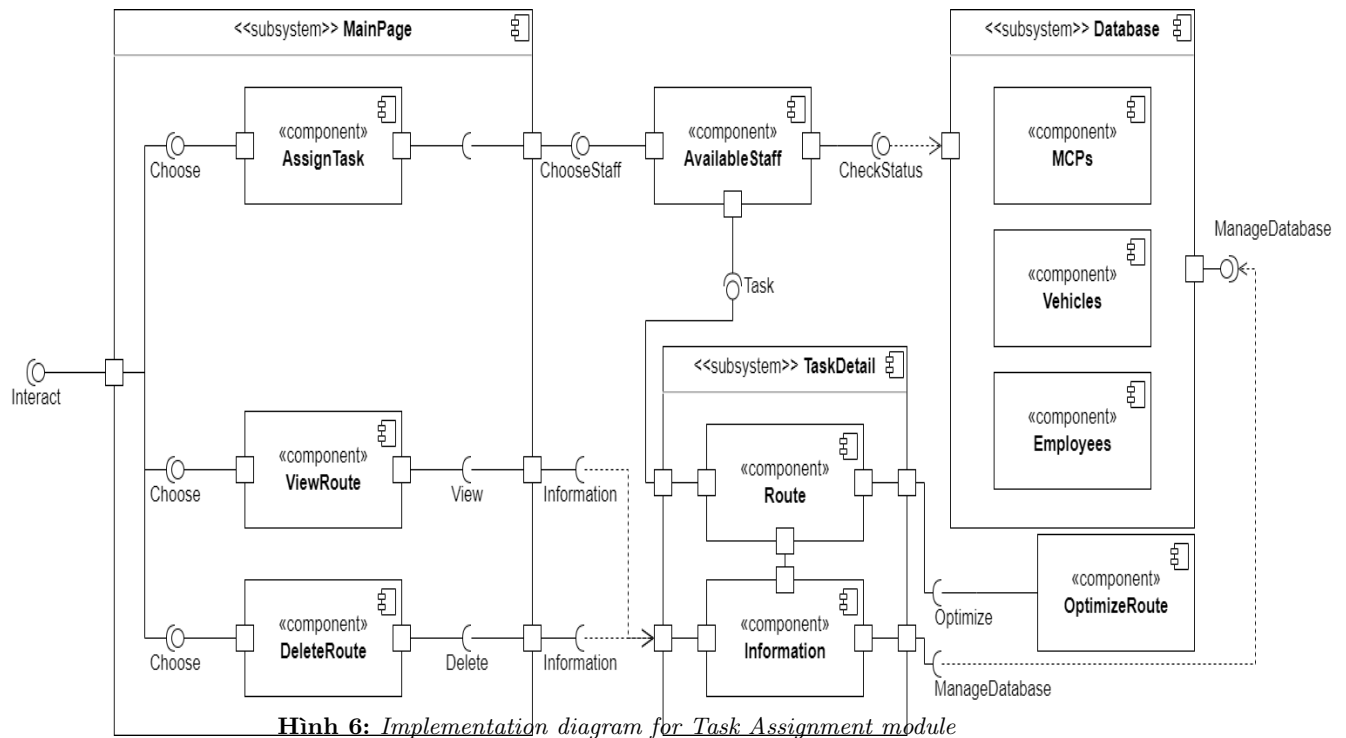


Hình 5: architectural approach

Nhóm sử dụng MVC design pattern để minh họa kiến trúc của hệ thống. Nó bao gồm ba thành phần chính được kết nối với nhau, đó là:

1. **View:** xác định và quản lý cách dữ liệu được trình bày cho người dùng. Tức là trong ứng dụng, view được sử dụng để hiển thị màn hình đăng nhập, tuyến đường, thông báo, ... cho người sử dụng.
 - Render UI
 - Render các thành phần thông tin của nhân viên, phương tiện, MCPs, chi nhánh.
 - Render ra các form đăng nhập hay đăng ký của người dùng
 - Render giao diện nhắn tin, tương tác giữa người dùng
 - Render tuyến đường đi cho nhân viên
 - Render ra các form để thực hiện các hành động CRUD đối với các trang phân công nhiệm vụ, tạo tuyến đường,...
 - Tương tác với người dùng bằng cách xử lý các sự kiện tự động ngay trong view:
 - Tạo thông báo cho nhân viên liên quan sau khi thực hiện phân công nhiệm vụ
 - Gửi các sự kiện thực hiện bởi người dùng (sự kiện chọn, click, submit form,...) về cho controller.
 - Hiện ra thông tin, nhiệm vụ, tuyến đường khi xử lý sự kiện click vào thông tin nhân viên
 - Khi được thông báo các thay đổi trạng thái từ model, thực hiện truy xuất dữ liệu từ model để thực hiện refresh lại view và update view mới.
 - Truyền dữ liệu vào các request gửi về cho controller để thực hiện xử lý các logic của hệ thống
2. **Controller:** quản lý tương tác của người dùng (ví dụ: các lần nhấn phím, nhấp chuột, ...) và chuyển các tương tác này đến view và model. Do đó, trong ứng dụng, các chức năng chính của nó là ánh xạ các hành động của người dùng để cập nhật mô hình và chọn chế độ xem để hiển thị.
 - Gọi API google map khi thực hiện tính toán tối ưu tuyến đường
 - Trả về một view tùy theo trang được request.
 - Trả về response cho khách hàng với request tương ứng.
 - Gọi các hàm từ model để thực hiện các thao tác trên dữ liệu:
 - Sử dụng các phương thức của model để thực hiện các hành động CRUD của phân công nhiệm vụ, tạo tuyến đường, thông tin nhân viên, phương tiện, MCPs,...
 - Sử dụng các phương thức từ model để thực hiện các thao tác đăng ký, đăng nhập, quản lý tài khoản người dùng.
3. **Model:** là thành phần trung tâm của pattern quản lý trực tiếp dữ liệu, logic và các quy tắc của ứng dụng. Trong ứng dụng, nó có một số chức năng bao gồm lưu trữ thông tin, kiểm tra thông tin mcps, xác nhận đăng nhập, cập nhật dữ liệu, tạo thông báo,..
 - Cung cấp các phương thức tương tác với database để thực hiện việc đăng ký, đăng nhập, quản lý tài khoản người dùng.
 - Cung cấp các phương thức để thực hiện các thao tác với database (thêm, xóa, sửa) với các feature quản lý nhân viên, phương tiện, phân công nhiệm vụ. . .
 - Khi trạng thái của model (về dữ liệu,...) thay đổi => thông báo cho view để thực hiện refresh và cập nhật lại view.
 - Lưu trữ thông tin tuyến đường, bản đồ

3.2 Task 3.2



Hình 6: Implementation diagram for Task Assignment module

Link Drive chứa ảnh Implementation diagram:

https://drive.google.com/file/d/1UxktRYYNDGOWFx5nK3DAdAc5Q_D4m89/view

4 Task 4

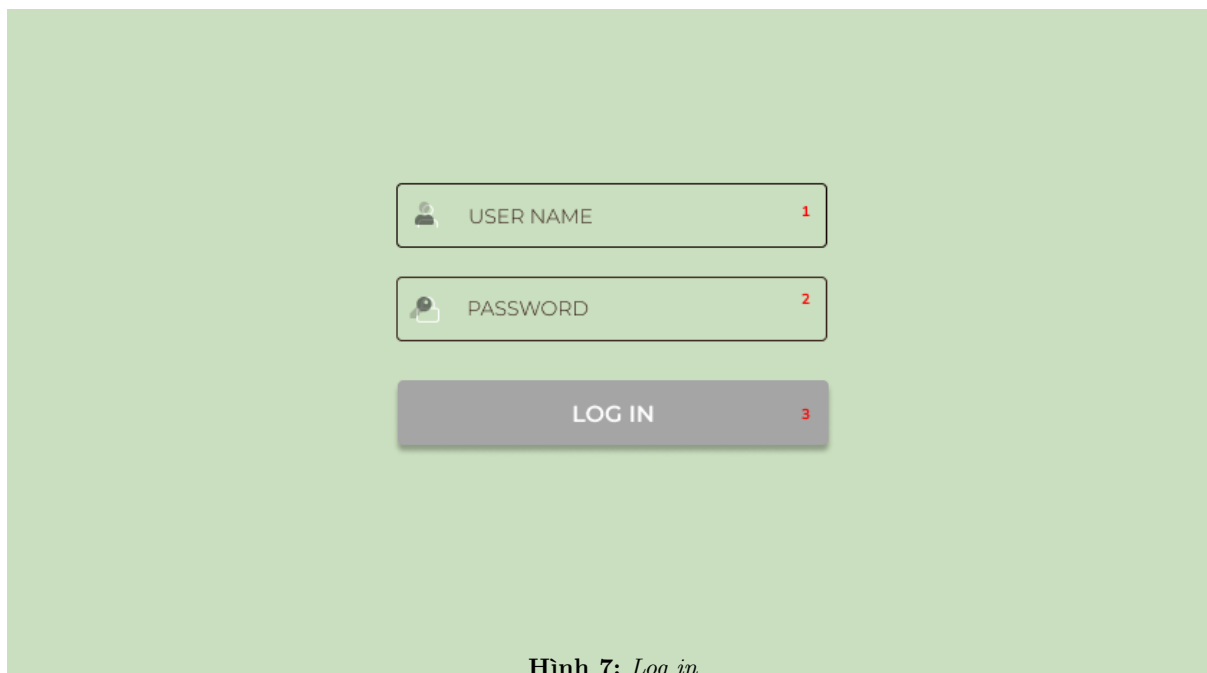
4.1 Task 4.1 & Task 4.2

Link github

<https://github.com/Henrywalker2002/taskAssign>

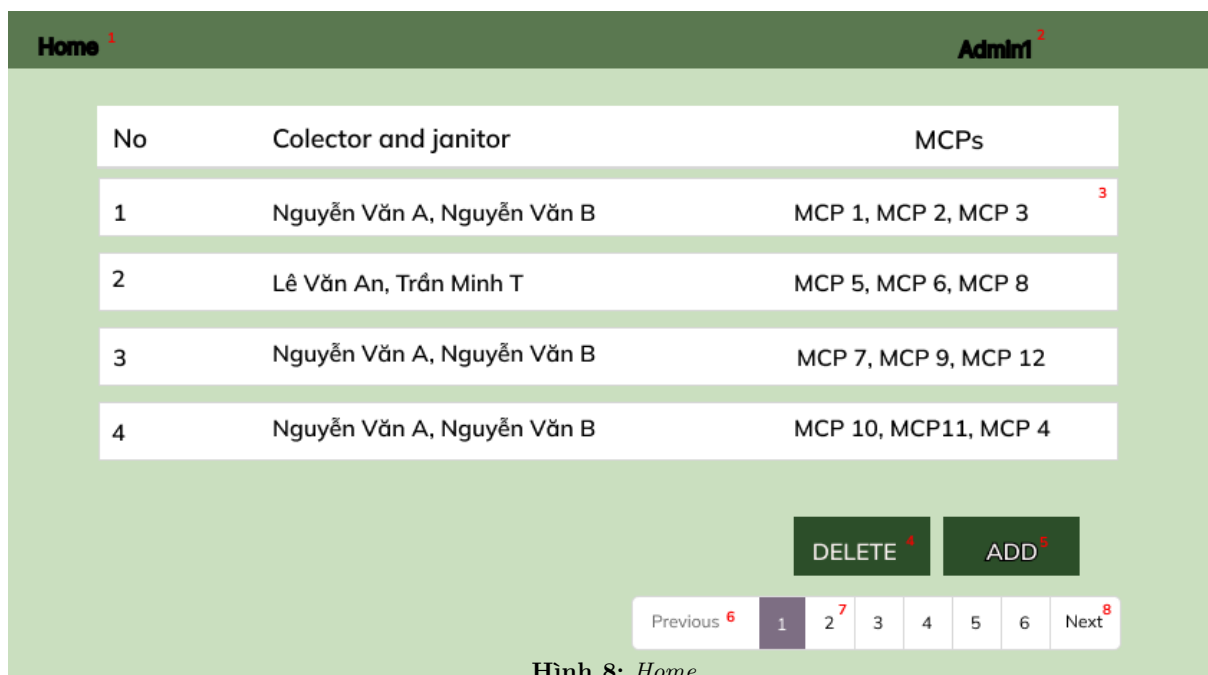
4.2 Task 4.3

4.2.1 Log in



	Type	Mô tả
1	Input	Nhập vào tên đăng nhập
2	Input	Nhập vào mật khẩu của tài khoản
3	Button	Đăng nhập, chuyển sang giao diện chính của hệ thống

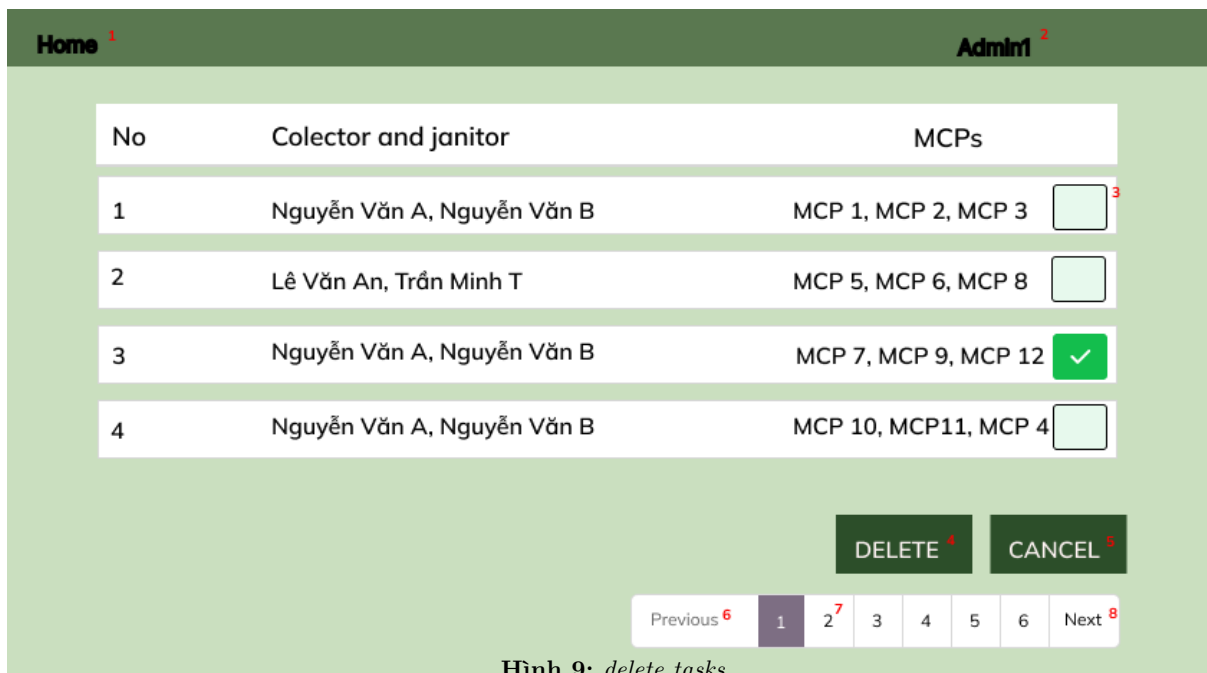
4.2.2 Home



Hình 8: Home

	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Button	Chọn nhiệm vụ để xem chi tiết hoặc sửa đổi (hình 18)
4	Button	Ấn vào để chuyển sang trang xóa (hình 9)
5	Button	Thêm nhóm đã chọn vào danh sách phân công nhiệm vụ và chuyển sang trang thêm (hình 11) hoặc hiện lên popup (hình 12) nếu không đủ MCPs trống
6	Button	Chuyển về danh sách trước đó
7	Button	Chuyển qua danh sách có số trang tương ứng với số được chọn
8	Button	Chuyển qua danh sách tiếp theo

4.2.3 Delete tasks



No	Collector and janitor	MCPs
1	Nguyễn Văn A, Nguyễn Văn B	MCP 1, MCP 2, MCP 3
2	Lê Văn An, Trần Minh T	MCP 5, MCP 6, MCP 8
3	Nguyễn Văn A, Nguyễn Văn B	MCP 7, MCP 9, MCP 12
4	Nguyễn Văn A, Nguyễn Văn B	MCP 10, MCP11, MCP 4

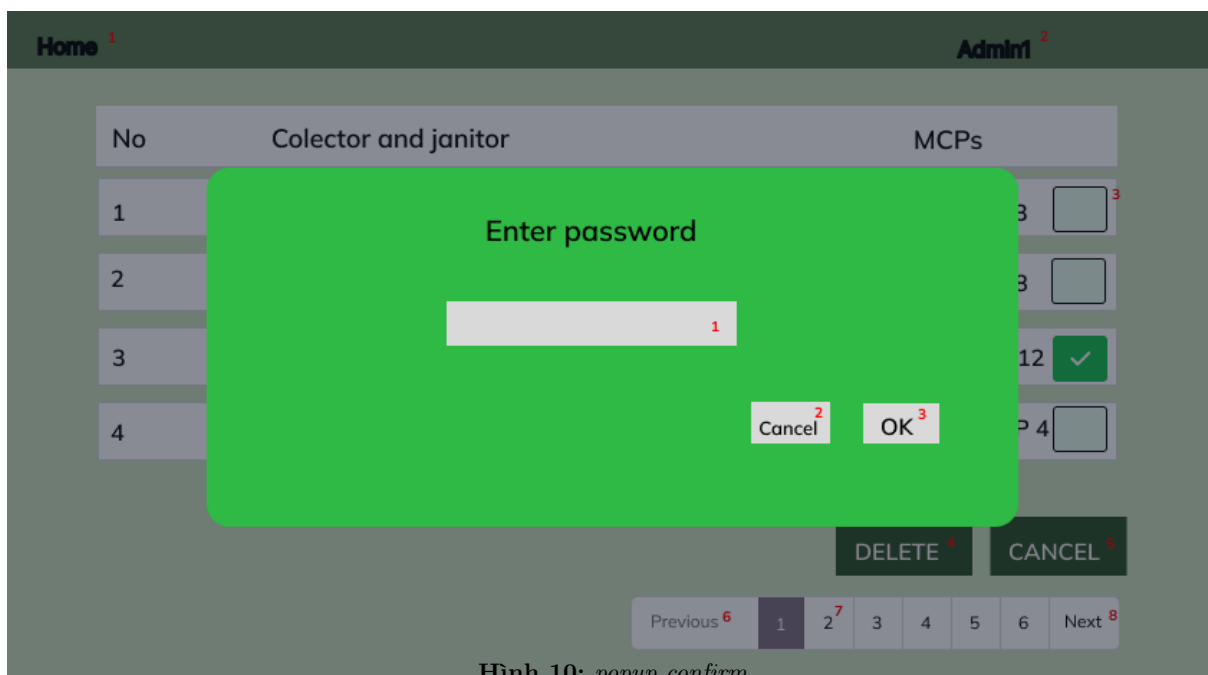
DELETE CANCEL

Previous 1 2 3 4 5 6 Next

Hình 9: delete tasks

	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Checkbox	Tích vào các ô checkbox để chọn task để thực hiện thao tác xóa bên dưới
4	Button	Hiện lên popup để xác nhận xóa các nhóm đã chọn
5	Button	Quay lại trang trước đó
6	Button	Chuyển về danh sách trước đó
7	Button	Chuyển qua danh sách có số trang tương ứng với số được chọn
8	Button	Chuyển qua danh sách tiếp theo

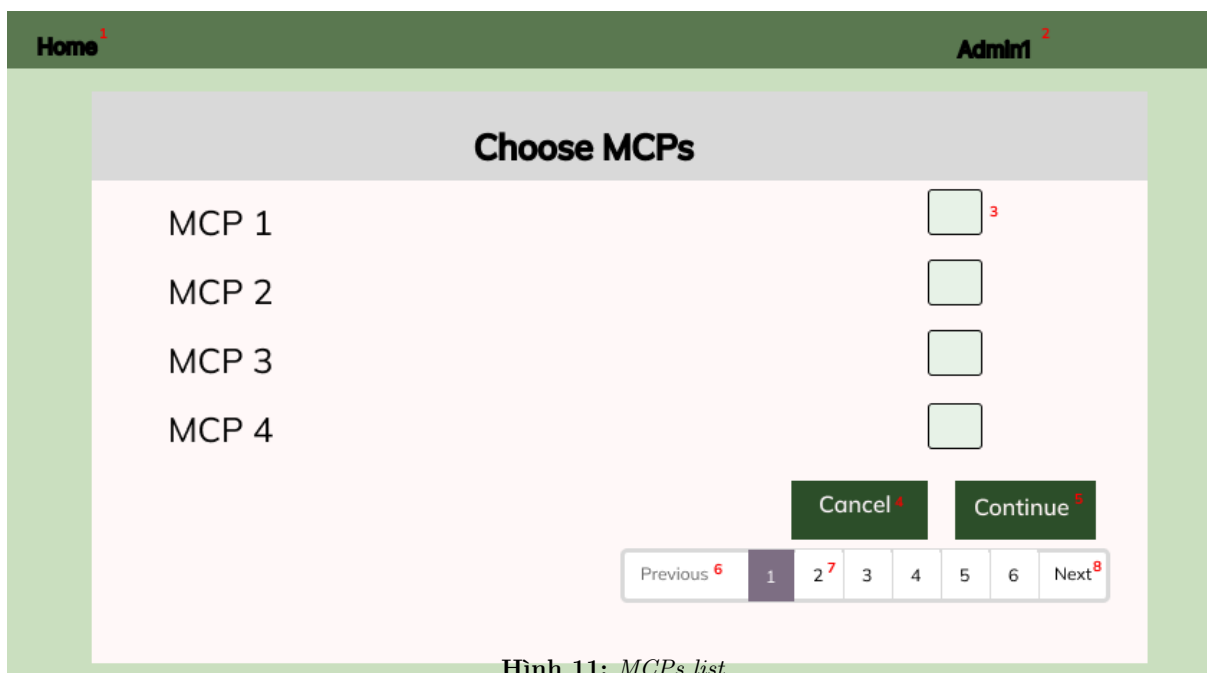
4.2.4 Pop up confirm



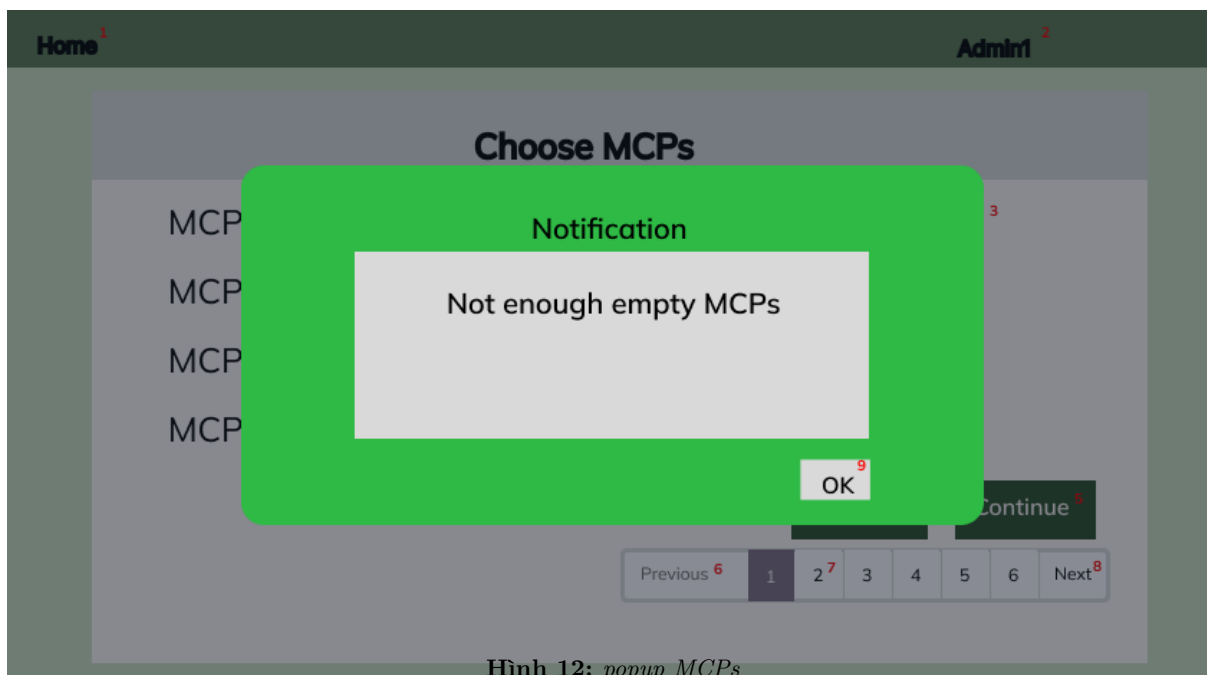
Hình 10: *popup confirm*

	Type	Mô tả
1	Input	Nhập mật khẩu để thực hiện thao tác xóa
2	Button	Hủy thao tác
3	Checkbox	Xác nhận xóa

4.2.5 Choose MCPs

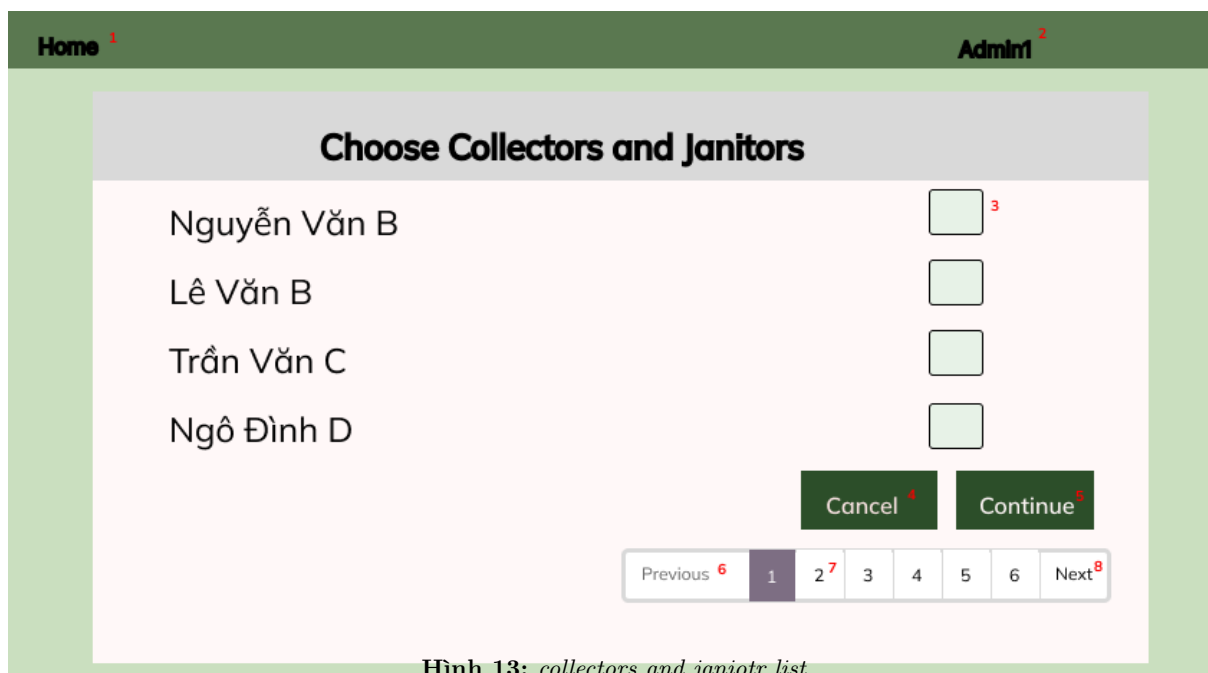


Nếu không có MCPs trống hệ thống sẽ xuất ra như hình dưới:



	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Checkbox	Chọn MCP để thực hiện phân công nhiệm vụ
4	Button	Quay về trang trước (trang chủ)
5	Button	Chuyển qua bước tiếp theo sau khi chọn các MCP để thực hiện phân công nhiệm vụ (hình 13) hoặc hiện lên popup (hình 14) nếu không đủ nhân viên
6	Button	Chuyển về danh sách trước đó
7	Button	Chuyển qua danh sách cố trang tương ứng với số được chọn
8	Button	Chuyển qua danh sách tiếp theo
9	Button	Quay về trang chủ

4.2.6 Choose Collectors and Janitors



Home Admin1

Choose Collectors and Janitors

Nguyễn Văn B ☐

Lê Văn B ☐

Trần Văn C ☐

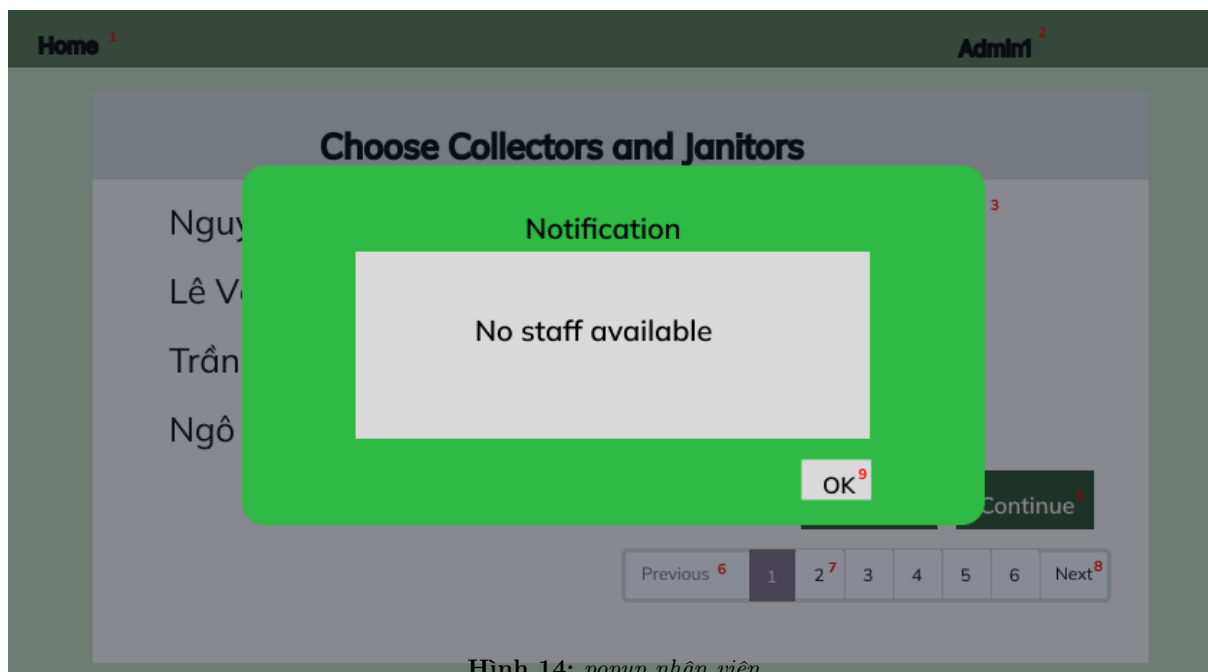
Ngô Đình D ☐

Cancel Continue

Previous 1 2 3 4 5 6 Next

Hình 13: collectors and janitor list

Nếu không có sẵn nhân viên đang rảnh hệ thống sẽ xuất ra hình dưới:



Home Admin1

Choose Collectors and Janitors

Notification

No staff available

OK

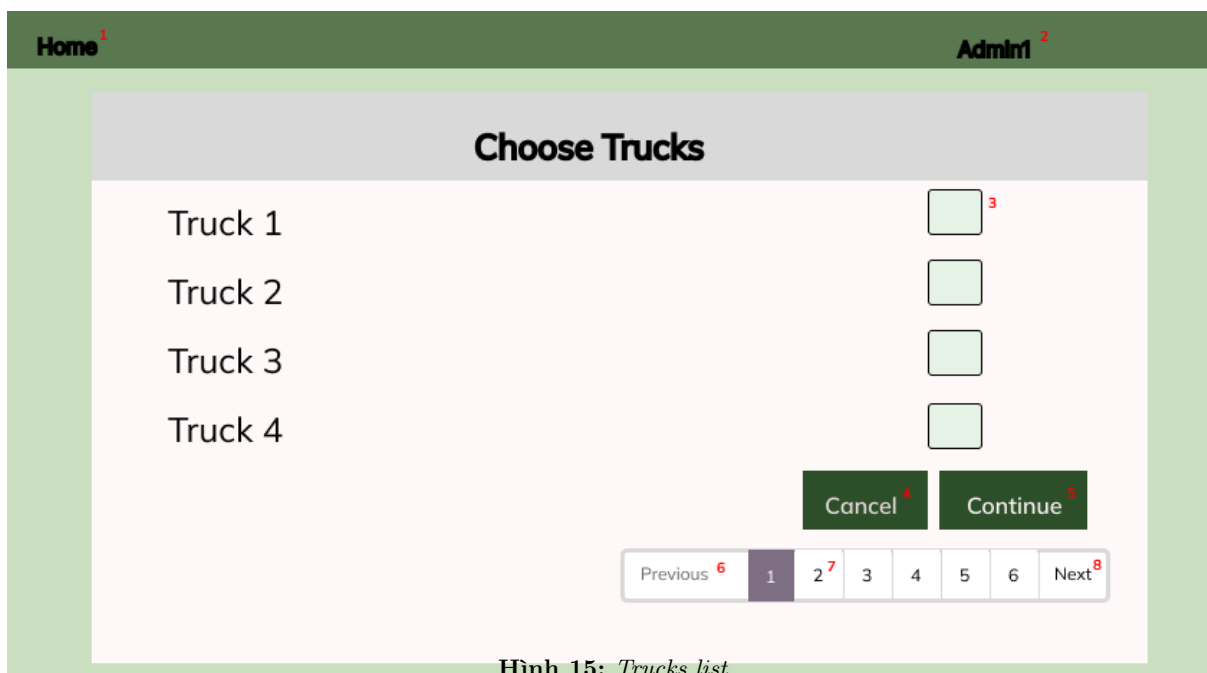
Continue

Previous 1 2 3 4 5 6 Next

Hình 14: popup nhân viên

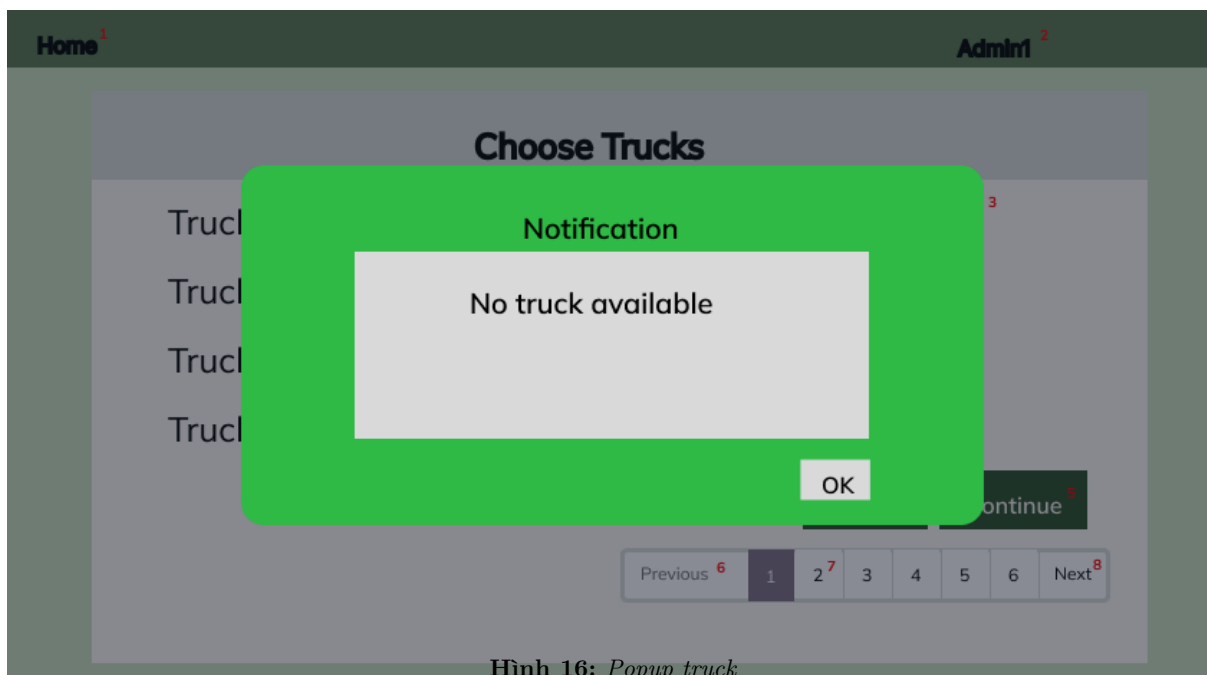
	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Checkbox	Chọn Collectors và janitors để thực hiện phân công nhiệm vụ
4	Button	Quay về trang phân công nhiệm vụ
5	Button	Chuyển qua bước tiếp theo sau khi chọn các Collectors và janitors để thực hiện phân công nhiệm vụ (hình 15) hoặc hiện lên popup nếu không đủ xe (hình 16)
6	Button	Chuyển về danh sách trước đó
7	Button	Chuyển qua danh sách có số trang tương ứng với số được chọn
8	Button	Chuyển qua danh sách tiếp theo
9	Button	Quay về trang chủ

4.2.7 Choose trucks



Hình 15: Trucks list

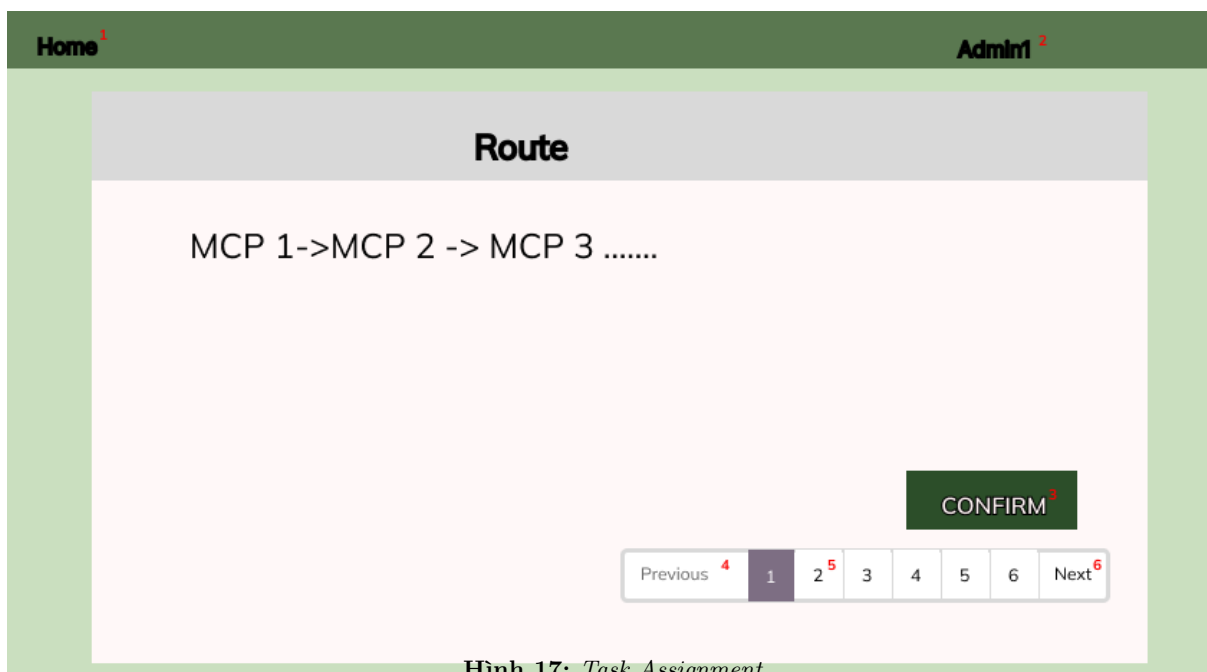
Nếu không có sẵn xe hệ thống sẽ xuất ra hình dưới:



Hình 16: Popup truck

	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Checkbox	Chọn phương tiện để thực hiện phân công nhiệm vụ
4	Button	Quay về trang phân công nhiệm vụ
5	Button	Chuyển qua bước tiếp theo sau khi chọn các phương tiện để thực hiện phân công nhiệm vụ (hình 17)
6	Button	Chuyển về danh sách trước đó
7	Button	Chuyển qua danh sách có số trang tương ứng với số được chọn
8	Button	Chuyển qua danh sách tiếp theo
9	Button	Quay về trang chủ

4.2.8 Task assignment



Hình 17: Task Assignment

	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Button	Xác nhận để chuyển qua bước tiếp theo sau khi chọn MCP, nhân viên, phương tiện để thực hiện phân công nhiệm vụ và quay về trang chủ
4	Button	Chuyển về danh sách trước đó
5	Button	Chuyển qua danh sách có số trang tương ứng với số được chọn
6	Button	Chuyển qua danh sách tiếp theo

4.2.9 Task assignment



Task No 1

Collectors and Janitor	Nguyễn Văn A, Nguyễn Văn B
MCPs	MCP 1, MCP 2, MCP 3
Trunk	Trunk 1
Route	MCP 2 -> MCP 1 -> MCP 3

Cancel ³ Save ⁴

Hình 18: Task Assignment

	Type	Mô tả
1	Button	Chuyển về trang chủ
2	Button	Chuyển đến trang thông tin cá nhân
3	Button	Hủy sửa nhiệm vụ đã phân công
4	Button	Lưu trạng thái nhiệm vụ đã sửa

5 Task 5

6 Thay đổi so với phiên bản trước

Tài liệu

[1] <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

[2] https://en.wikipedia.org/wiki/Class_diagram

[3] https://en.wikipedia.org/wiki/Activity_diagram

[4] https://en.wikipedia.org/wiki/Use_case_diagram