

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



CÔNG NGHỆ PHẦN MỀM (CO3001)

URBAN WASTE COLLECTION AID - UWC 2.0

Task 3: Architecture design

Teacher: Phan Trung Hiếu

Bùi Công Tuấn

Class: L02

Students:	Lã Nguyễn Gia Hy	2110231
	Võ Văn Khả	2110264
	Đặng Quang Vinh	2110667
	Nguyễn Phan Hoàng Phúc	2110457
	Mai Đình Quốc Anh	2012595
	Trần Đức Tiến	2014732
	Hoàng Xuân Lộc	2013680

HO CHI MINH CITY, MARCH 2023

Mục lục

1	Phân chia công việc	2
2	Requirement elicitation	3
2.1	Bối cảnh và các bên liên quan trong dự án	3
2.2	Các yêu cầu chức năng, phi chức năng và biểu đồ use-case cho toàn hệ thống	4
2.3	Yêu cầu chức năng	4
2.4	Yêu cầu phi chức năng	4
2.5	Use-case Diagram cho toàn hệ thống	6
2.6	Mô tả các use-case trong module Task Assignment	7
2.6.1	Use-case diagram cho Task Assignment module	7
2.6.2	Mô tả Use-case	8
3	System Modeling	12
3.1	Activity Diagram	12
3.2	Sequence Diagram	14
3.3	Class Diagram	17
3.4	Thiết kế giao diện quản lý nhiệm vụ	24
3.4.1	Link Figma & Mockup	24
3.4.2	Mô tả chi tiết	24
4	Architecture design	32
4.1	Use a layered architecture to design the UWC 2.0 system	33
4.2	Describe how will you present your User Interface	34
4.3	Describe how will you store your data	34
4.4	Describe how you will access to external services/ APIs	35
4.5	Draw a component diagram for the Task Assignment module	37

1 Phân chia công việc

No.	Fullname	Student ID	Problems	Percentage of work
1	Hoàng Xuân Lộc	2013680	Task 1.3, 2.1, 3.1	100 %
2	Trần Đức Tiến	2014732	Task 1.1, 2.1, 3.1	100 %
3	Nguyễn Phan Hoàng Phúc	2110457	Task 1.2, 2.2, 3.2	100 %
4	Đặng Quang Vinh	2110667	Task 1.2, 2.2, 3.2	100 %
5	Mai Đình Quốc Anh	2012595	Task 1.3, 2.3, 3.1	100 %
6	Võ Văn Khả	2110264	Task 1.3, 2.4, 3.1	100 %
7	Lã Nguyễn Gia Hy	2110231	Task 1.3, 2.4, 4.3	100 %

2 Requirement elicitation

2.1 Bối cảnh và các bên liên quan trong dự án

Task 1.1

Describe the domain context of Urban waste management in Vietnam. Who are relevant stakeholders? What are their current needs? In your opinion, what benefits UWC 2.0 will be for each stakeholder?

Hiện nay, quản lý chất thải đô thị là một trong những vấn đề quan trọng mà chúng ta cần phải giải quyết ngay. Phần lớn chất thải đến từ các hộ gia đình, trường học, bệnh viện và địa điểm kinh doanh. Theo như số liệu thống kê tại Thành phố Hồ Chí Minh vào năm 2020 thì số lượng rác thải mỗi ngày lên đến 9.000 đến 9.500 tấn và tăng lên 6 – 10% mỗi năm, phần lớn chất thải đến từ các hộ gia đình, trường học, bệnh viện và địa điểm kinh doanh. Để có thể xử lý lượng rác thải khổng lồ này, một tổ chức X đã ký hợp đồng với nhà cung cấp dịch vụ Y để phát triển hệ thống hỗ trợ thu gom rác thải đô thị - UWC 2.0. Dựa theo quy trình thu gom rác điển hình, ta có thể thấy được 3 stakeholders trong projects này là: Back Officers – người vận hành hệ thống trung tâm để tạo lịch, điều phối người thu gom và lao công, Collectors – người lái các loại phương tiện khác nhau và Janitors – người thu gom rác thủ công từ các điểm tập kết rác lớn (MCPs). Từ hệ thống UWC 1.0, có thể xác định được nhu cầu của các stakeholders đối với UWC 2.0 như sau:

- Back Officers:
 - Quản lý được lịch làm việc và nhiệm vụ của Collectors và Janitors
 - Gửi thông tin tuyển thu gom và thời gian cho Collectors và Janitors
 - Lên kế hoạch về phương tiện sẽ sử dụng và tuyến đường của Collectors và Janitors
 - Có thể giao tiếp với Collectors và Janitors
- Collectors và Janitors:
 - Xem được lịch làm việc, nhiệm vụ hàng ngày và hàng tuần của họ
 - Giao tiếp được với Back Officers và các công nhân khác
 - Check in / check out công việc
 - Nhận thông báo nếu MCP đầy
 - Giao diện dễ dùng, dễ tiếp cận với những người ít sự hiểu biết về công nghệ

Theo em, UWC 2.0 sẽ giúp cải thiện việc quản lý rác thải đô thị và đem lại nhiều lợi ích cho những người công nhân như:

- Back Officers: Thuận tiện hơn trong việc quản lý các Collectors và Janitors, sắp xếp và phân chia công việc một cách hợp lý hơn. Từ đó đẩy mạnh tiến độ thu gom và xử lý rác thải
- Collectors và Janitors: Hiểu rõ được lịch làm việc và nhiệm vụ trong ngày và trong tuần, từ đó có thể sắp xếp thời gian hợp lý, giải quyết được các vấn đề đột ngột thông qua giao tiếp trực tiếp với Back Officers, thuận tiện trong việc check in/ check out => Nâng cao hiệu suất làm việc của công nhân

2.2 Các yêu cầu chức năng, phi chức năng và biểu đồ use-case cho toàn hệ thống

Task 1.2

Describe all functional and non-functional requirements that can be inferred from the project description. Draw a general use-case diagram for the whole system.

2.3 Yêu cầu chức năng

1. Đối với Back Officers

- Có tài khoản cá nhân trên hệ thống, có thể đăng nhập, đăng xuất.
- Có thể quản lý (xem, sửa, xóa) thông tin tổng quát của collectors và janitors, và lịch làm việc của họ.
- Có thể quản lý (xem, sửa, xóa) thông tin của MCPs, theo dõi trạng thái của chúng.
- Có thể quản lý (xem, sửa, xóa) thông tin các phương tiện thu gom rác, thông tin kỹ thuật chi tiết (cân nặng, sức chứa, độ tiêu thụ nhiên liệu,...).
- Có thể phân công, chỉ định công việc, phương tiện cho Collectors, Janitors.
- Có thể tạo tuyến đường cho mỗi collectors để tối ưu trong việc tiết kiệm nhiên liệu và hạn chế quãng đường di chuyển.
- Có thể gửi thông báo tới Collectors và Janitors về tình trạng MCPs, các thông báo về ngày nghỉ, các thông báo khác.
- Có thể gửi tin nhắn cho các Collectors và Janitors, cũng như các Back Officers khác.

2. Đối với Collectors

- Có tài khoản cá nhân trên hệ thống, có thể đăng nhập, đăng xuất.
- Có thể điểm danh, đánh dấu hoàn thành công việc hàng ngày (Check in / Check out).
- Có thể xem được lịch làm việc cá nhân (ngày, tuần, tháng), nhiệm vụ được giao, thời gian thực hiện, địa điểm thực hiện, phương tiện thực hiện, tuyến đường đi.
- Nhận được thông báo từ Back Officers khi MCPs đầy cũng như các thông báo khác.
- Có thể gửi tin nhắn cho các Back Officers và Janitors, cũng như các Collectors khác.

3. Đối với Janitors

- Có tài khoản cá nhân trên hệ thống, có thể đăng nhập, đăng xuất.
- Có thể điểm danh, đánh dấu hoàn thành công việc hàng ngày (Check in / Check out).
- Có thể xem được lịch làm việc cá nhân (ngày, tuần, tháng), nhiệm vụ được giao, thời gian thực hiện, địa điểm thực hiện, phương tiện thực hiện.
- Nhận được thông báo từ Back Officers khi MCPs đầy cũng như các thông báo khác.
- Có thể gửi tin nhắn cho các Back Officers và Collectors, cũng như các Janitors khác.

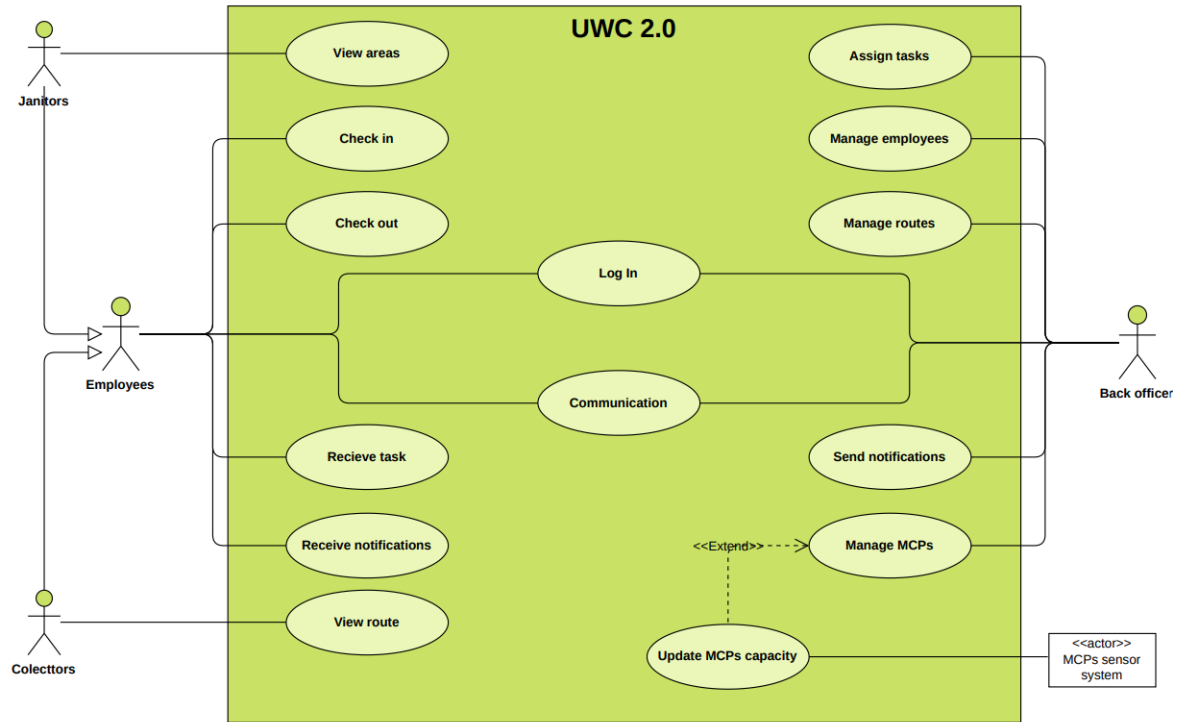
2.4 Yêu cầu phi chức năng

• Yêu cầu về giao diện

- Phải có chế độ sáng (Light) và chế độ tối (Dark) cho người dùng lựa chọn
- Những thông tin về lịch trình được hệ thống hiển thị chỉ trên một khung hình (không cần phải cuộn để xem).
- Các thao tác quan trọng như phân công công việc, check in, check out cần có thông báo hoặc hiệu ứng thao tác thành công (hoặc thất bại).
- Màu sắc chủ đạo là màu lạnh, chọn màu tương phản làm nổi bật thông báo và nhiệm vụ.

- Ngôn ngữ hiện tại là Tiếng Việt, trong tương lai có thể nâng cấp thành tiếng Anh.
- Có khả năng thích ứng với nhiều kích cỡ màn hình khác nhau (desktop, mobile, tablet). Có thể thích ứng với nhiều loại trình duyệt web (Google, Safari, FireFox, Internet Explorer).
- **Yêu cầu về hiệu năng, hiệu suất**
 - Hệ thống phải có khả năng kiểm soát dữ liệu thời gian thực từ ít nhất 1000 MCPs cùng lúc và 10 000 MCPs trong 5 năm.
 - Có khả năng cho phép hoạt động tốt khi có 200 người dùng truy cập đồng thời.
 - Thời gian tải các trang web dưới 2 giây.
 - Tốc độ xử lý một thao tác dưới 1 giây.
 - Nhấn tin trong thời gian thực, độ trễ dưới 1 giây.
- **Yêu cầu về độ tin cậy**
 - Hoạt động ổn định trong thời gian 5:00 - 23:00 tất cả các ngày (trừ các ngày lễ)
 - Xác suất hệ thống bị sập dưới 3%. Thời gian phục hồi nếu bị sập dưới 90 giây.
 - Thông tin nên được cập nhật từ các MCP mỗi 15 phút với độ khả dụng ít nhất 95% thời gian vận hành.
 - Thông báo mới được cập nhật không quá 10 giây kể từ khi có sự thay đổi.
- **Yêu cầu về tính bảo mật**
 - Chỉ có các tài khoản được cung cấp sẵn mới có thể đăng nhập (không có chức năng đăng kí).
 - Có khả năng phát hiện truy cập bất thường và thông báo tới người quản trị hệ thống.
- **Yêu cầu về tính hợp pháp**
 - Phù hợp với pháp luật hiện hành.
- **Yêu cầu về tổ chức hệ thống**
 - Sử dụng không quá 500MB RAM trong quá trình hệ thống chạy.
 - Có khả năng mở rộng và phát triển trong tương lai.
 - Dữ liệu được kế thừa từ UWC 1.0.

2.5 Use-case Diagram cho toàn hệ thống



Use-case Diagram cho cả hệ thống

Bảng mô tả use-case

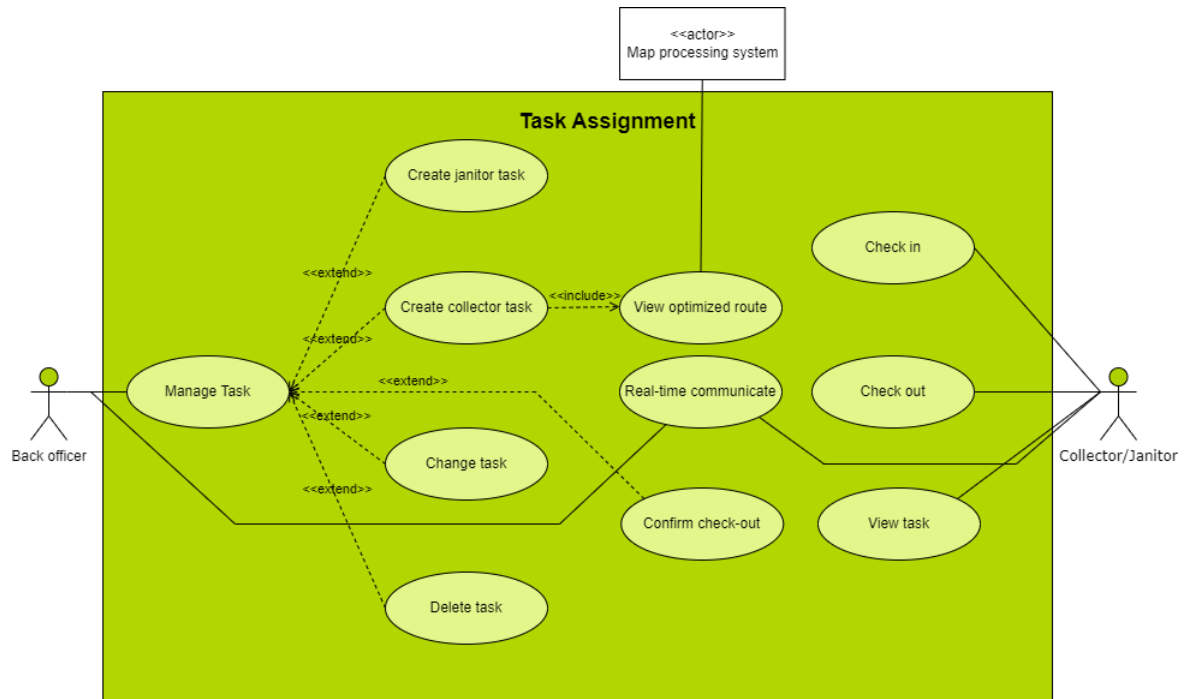
Use-case ID	Tên use-case	Mô tả
1	Login	Đăng nhập vào hệ thống
2	Communicate	Giao tiếp với nhau qua hệ thống bằng cách gửi/nhận tin nhắn
3	Check in	Điểm danh trước khi bắt đầu làm việc
4	Check out	Điểm danh kết thúc công việc
5	Receive notifications	Nhận thông báo
6	Receive task	Nhận phân công công việc
7	View route	Xem lộ trình tuyến đường
8	View areas	Xem khu vực chỉ định
9	Manage employees	Quản lý nhân viên (Xem và chỉnh sửa thông tin)
10	Send notifications	Gửi thông báo
11	Assign tasks	Phân công công việc
12	Manage MCPs	Quản lý MCPs
13	Manage routes	Quản lý lộ trình tuyến đường
14	Update MCPs capacity	Cập nhật sức chứa của MCPs

2.6 Mô tả các use-case trong module Task Assignment

Task 1.3

For the Task assignment module, draw its use-case diagram and describe the use-case using a table format

2.6.1 Use-case diagram cho Task Assignment module



2.6.2 Mô tả Use-case

Xem nhiệm vụ

Use case name	Xem nhiệm vụ
Actors	Collector/Janitor
Description	Collector/Janitor muốn xem chi tiết về thông tin nhiệm vụ được giao
Trigger	Collector/Janitor nhấn vào nút "Xem lịch làm việc"
Preconditions	- Thiết bị đã được kết nối với Internet và hệ thống - Collector/Janitor đã đăng nhập vào hệ thống và có quyền truy cập vào lịch làm việc
Postconditions	Không có
Normal flow	1. Hệ thống lấy dữ liệu danh sách các ca làm việc trong tuần của nhân viên tương ứng. 2. Hệ thống hiển thị danh sách các ca làm việc. 3. Collector/Janitor chọn một ngày làm việc nào đó 4. Hệ thống lấy dữ liệu chi tiết về ngày, giờ, địa điểm làm việc,... của nhiệm vụ 5. Hệ thống hiển thị thông tin chi tiết nhiệm vụ ở trang mới.
Alternate flow	Không có
Exception flow	Tại bước 2: 2.1 Hệ thống lấy dữ liệu thất bại và hiển thị thông báo Tại bước 5: 5.1 Hệ thống lấy dữ liệu thất bại và hiển thị thông báo

Xác nhận hoàn thành nhiệm vụ

Use case name	Xác nhận hoàn thành nhiệm vụ
Actors	Back officer
Description	Back officer muốn xác nhận hoàn thành nhiệm vụ cho Collector/Janitor
Trigger	Back officer muốn xác nhận cho một nhiệm vụ đang chờ xác nhận.
Preconditions	- Thiết bị đã được kết nối với Internet và hệ thống - Back officer đang ở cửa sổ quản lý nhiệm vụ - Thông báo xác nhận nhiệm vụ đã được Collector/Janitor gửi
Postconditions	- Trạng thái của nhiệm vụ được cập nhật mới - Hệ thống thông báo xác nhận thành công đến Collector/Janitor
Normal flow	1. Back officer nhấn vào nút "Xác nhận" của một thông báo xác nhận hoàn thành nhiệm vụ trong danh sách. 2. Hệ thống hiển thị thông báo xác nhận thành công. 3. Hệ thống cập nhật lại hiển thị trạng thái mới cho nhiệm vụ tương ứng và xóa thông báo xác nhận trong danh sách.
Alternate flow	Không có
Exception flow	Tại bước 2: 2.1 Hệ thống thông báo xác nhận thất bại

Quản lý nhiệm vụ

Use case name	Quản lý nhiệm vụ
Actors	Back officer
Description	Back officers muốn biết danh sách các ca làm việc một cách tổng quan và chi tiết để có phân công nhiệm vụ cho collectors và janitors
Trigger	Back officer sẽ chọn vào mục “Quản lý nhiệm vụ” trên màn hình trang chủ.
Preconditions	- Thiết bị của back officer phải kết nối với Internet và hệ thống. - Back officer phải có tài khoản đăng nhập và được phép vào mục “Quản lý nhiệm vụ”
Postconditions	Những dữ liệu được back officer cập nhật sẽ được lưu vào database và hiển thị lại trên màn hình
Normal flow	1. Back officer chọn mục “Quản lý nhiệm vụ” trên trang chủ. 2. Hệ thống sẽ lấy ra dữ liệu các ca làm việc và hiển thị trên trang “Quản lý nhiệm vụ” 3. Với mỗi ca làm việc sẽ có extension points như “Tạo mới”, “Xóa” và “Điều chỉnh”. 4. Back officer chọn một ca cụ thể nào đó thì hệ thống sẽ lấy dữ liệu và hiển thị thông tin chi tiết của ca được chọn
Alternate flow	Tại bước 3: 3.1 Khi Back officer chọn nút "Xóa". 3.2 Xuất hiện nút "Hoàn tác" để hoàn tác lại.
Exception flow	Tại bước 3 và 4: Hệ thống lấy dữ liệu từ database thất bại thì sẽ hiện thông báo lỗi.

Thay đổi nhiệm vụ

Use case name	Thay đổi nhiệm vụ
Actors	Back officer
Description	Back officers nhận thấy một ca làm việc nào đó chưa hợp lý và muốn điều chỉnh ca đó.
Trigger	Back officer chọn vào nút “Điều chỉnh” nằm bên phải hàng của ca tương ứng muốn chỉnh sửa.
Preconditions	- Thiết bị của back officer phải kết nối với Internet và hệ thống. - Back officer đang ở trang “Quản lý nhiệm vụ”. - Thông tin của ca có trong danh sách hiển thị trên màn hình giao diện.
Postconditions	Thông tin của ca vừa chỉnh sửa sẽ được cập nhật trong hệ thống và trên giao diện
Normal flow	1. Hệ thống lấy thông tin của ca làm việc được chọn từ database. 2. Hệ thống sẽ hiển thị những trường dữ liệu cần chỉnh sửa. 3. Back officer chỉnh sửa thông tin cho ca (ngày, giờ bắt đầu, kết thúc, trạng thái ...) và ấn nút “Lưu”, được hiển thị trên một bảng nhỏ. 4. Hệ thống thông báo chỉnh sửa thành công.
Alternate flow	Không có
Exception flow	Tại bước 3: 3.1 Back officer chọn nút “Hủy”. 3.2 Hệ thống sẽ thoát bảng điều chỉnh và trở về trang “Quản lý nhiệm vụ” . Use-case dừng lại. Tại bước 4: 5.1 Hệ thống thông báo “Trùng thời gian với nhiệm vụ khác!!!” 5.2 Hệ thống yêu cầu điều chỉnh lại.

Xóa nhiệm vụ

Use case name	Xóa nhiệm vụ
Actors	Back officer
Description	Back officers muốn xóa một ca nào đó ra khỏi danh sách.
Trigger	Back officer sẽ chọn vào nút “Xóa” nằm bên phải hàng của ca tương ứng muốn xóa.
Preconditions	- Thiết bị của back officer phải kết nối với Internet và hệ thống. - Back officer đang ở trang “Quản lý nhiệm vụ”. - Thông tin của ca có trong danh sách hiện thị trên màn hình giao diện.
Postconditions	Hàng được xóa không còn xuất hiện trong danh sách và dữ liệu của ca đó không còn trong database hệ thống.
Normal flow	1. Hệ thống lấy thông tin của ca làm việc được chọn từ database. 2. Hệ thống sẽ hiển thị thông báo xác nhận xóa. 3. Back officer ấn nút “OK” được hiển thị trên một bảng nhỏ. 4. Hệ thống thông báo xóa thành công.
Alternate flow	Tại bước 3: 4.1 Hệ thống thông báo hoàn tác trong 2 phút. 4.2 Back officer chọn “Hoàn tác”. 4.3 Hệ thống sẽ trở lại ban đầu và không xóa ca đó ra khỏi danh sách, đồng thời dữ liệu của ca đó trong database không bị mất. Use-case dừng lại
Exception flow	Tại bước 3: 3.1 Back officer chọn nút “Hủy”. 3.2 Hệ thống sẽ thoát bảng điều chỉnh và trở về trang “Quản lý nhiệm vụ” . Use-case dừng lại.

Tạo nhiệm vụ

Use case name	Tạo nhiệm vụ
Actors	Back officer
Description	Back officers muốn tạo thêm một nhiệm vụ nào đó vào danh sách.
Trigger	Back officer sẽ chọn nút “Tạo nhiệm vụ” có trong trang "Quản lý nhiệm vụ".
Preconditions	- Thiết bị của back officer phải kết nối với Internet và hệ thống. - Back officer đang ở trang “Quản lý nhiệm vụ”.
Postconditions	Nhiệm vụ mới tạo sẽ được thêm vào trong danh sách và dữ liệu của nhiệm vụ đó sẽ có trong database hệ thống.
Normal flow	1. Back officer chọn nút "Tạo nhiệm vụ". 2. Hệ thống sẽ hiển thị màn hình điền form tạo nhiệm vụ mới. 3. Back officer điền các thông tin liên quan đến nhiệm vụ mới (ngày, giờ bắt đầu, giờ kết thúc, phương tiện, nhân sự, trạng thái,...). 4. Hệ thống lưu thông tin nhiệm vụ mới và thông báo tạo thành công.
Alternate flow	Không có
Exception flow	Tại bước 4: Hệ thống báo lỗi thêm nhiệm vụ thất bại và điều hướng trở lại trang "Quản lý nhiệm vụ"

Check in nhiệm vụ

Use case name	Check in nhiệm vụ mỗi ngày
Actors	Collector / Janitor
Description	Kiểm tra các nhiệm vụ cần làm trước khi bắt đầu làm việc vào mỗi ngày
Trigger	Collector / Janitor nhấn vào button "Check in"
Preconditions	- Thiết bị đã được kết nối với Internet và hệ thống - Collector / Janitor đã đăng nhập vào hệ thống - Collector / Janitor đã được giao việc
Postconditions	Collector / Janitor check-in thành công và được cập nhật vào database
Normal flow	1. Chọn vào danh mục các nhiệm vụ check in. 2. Collector / Janitor kiểm tra danh sách các nhiệm vụ được hiện thị. 3. Hệ thống cập nhật trạng thái các nhiệm vụ. 4. Hệ thống thông báo hoàn thành quá trình cập nhật.
Alternate flow	Không có
Exception flow	Không có

Check out nhiệm vụ

Use case name	Check out nhiệm vụ mỗi ngày
Actors	Collector / Janitor
Description	Cập nhật lại trạng thái các nhiệm vụ đã được thực hiện vào cuối ngày
Trigger	Collector / Janitor nhấn vào button "Check out"
Preconditions	- Thiết bị đã được kết nối với Internet và hệ thống - Collector / Janitor đã đăng nhập vào hệ thống - Thực hiện vào cuối ngày
Postconditions	Hoàn thành check out nhiệm vụ
Normal flow	1. Collector/Janitor chọn vào danh mục các nhiệm vụ check out. 2. Tại màn hình hiển thị, check out các nhiệm vụ. 3. Hệ thống cập nhật trạng thái công việc cần check out. 4. Hệ thống thông báo hoàn thành quá trình cập nhật.
Alternate flow	Không có
Exception flow	Không có

Giao tiếp

Use case name	Giao tiếp
Actors	Collector / Janitor
Description	Giao tiếp nội bộ giữa collectors, janitors và back officer
Trigger	Khi người dùng nhấn vào button "Message" trên ứng dụng
Preconditions	- Thiết bị đã được kết nối với Internet và hệ thống - Người dùng đã đăng nhập vào hệ thống
Postconditions	Người dùng gửi, nhận, xem tin nhắn thành công
Normal flow	1. Chọn vào danh mục tin nhắn. 2. Tại màn hình hiển thị, thực hiện các thao tác cần thiết như xem, gửi tin nhắn. 3. Hoàn thành quá trình giao tiếp.
Alternate flow	Không có
Exception flow	Không có

3 System Modeling

3.1 Activity Diagram

Task 2.1

Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module

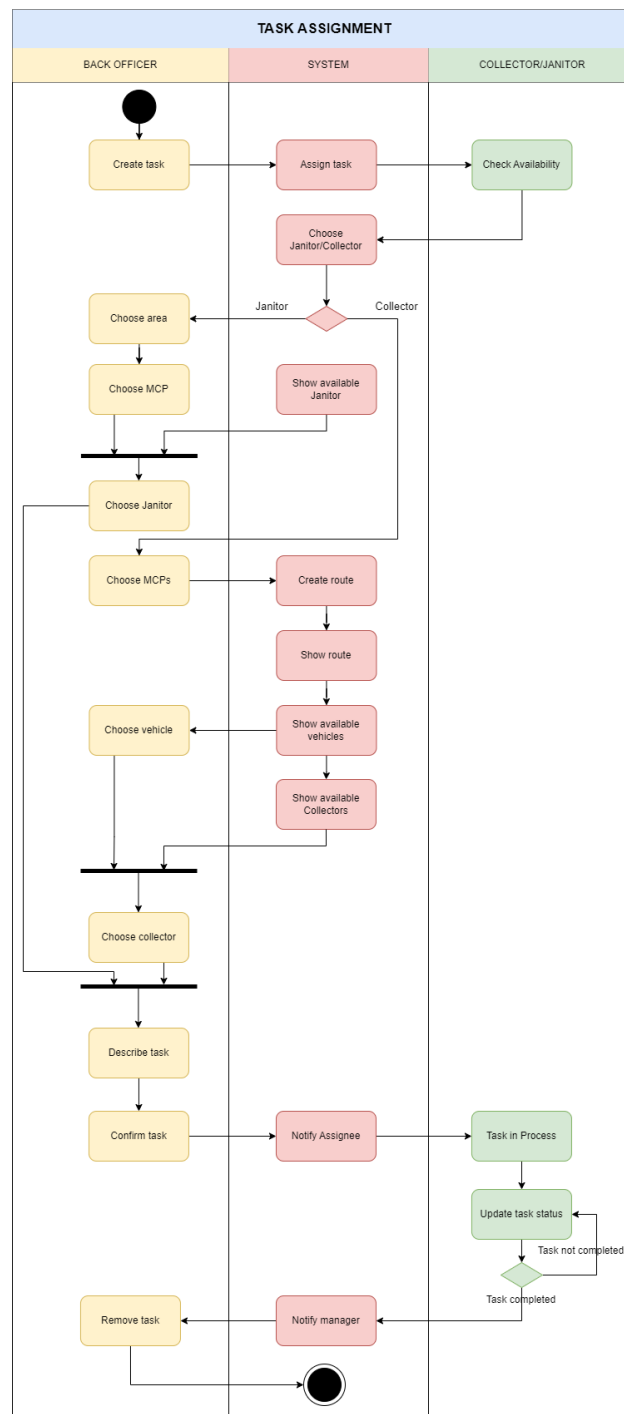


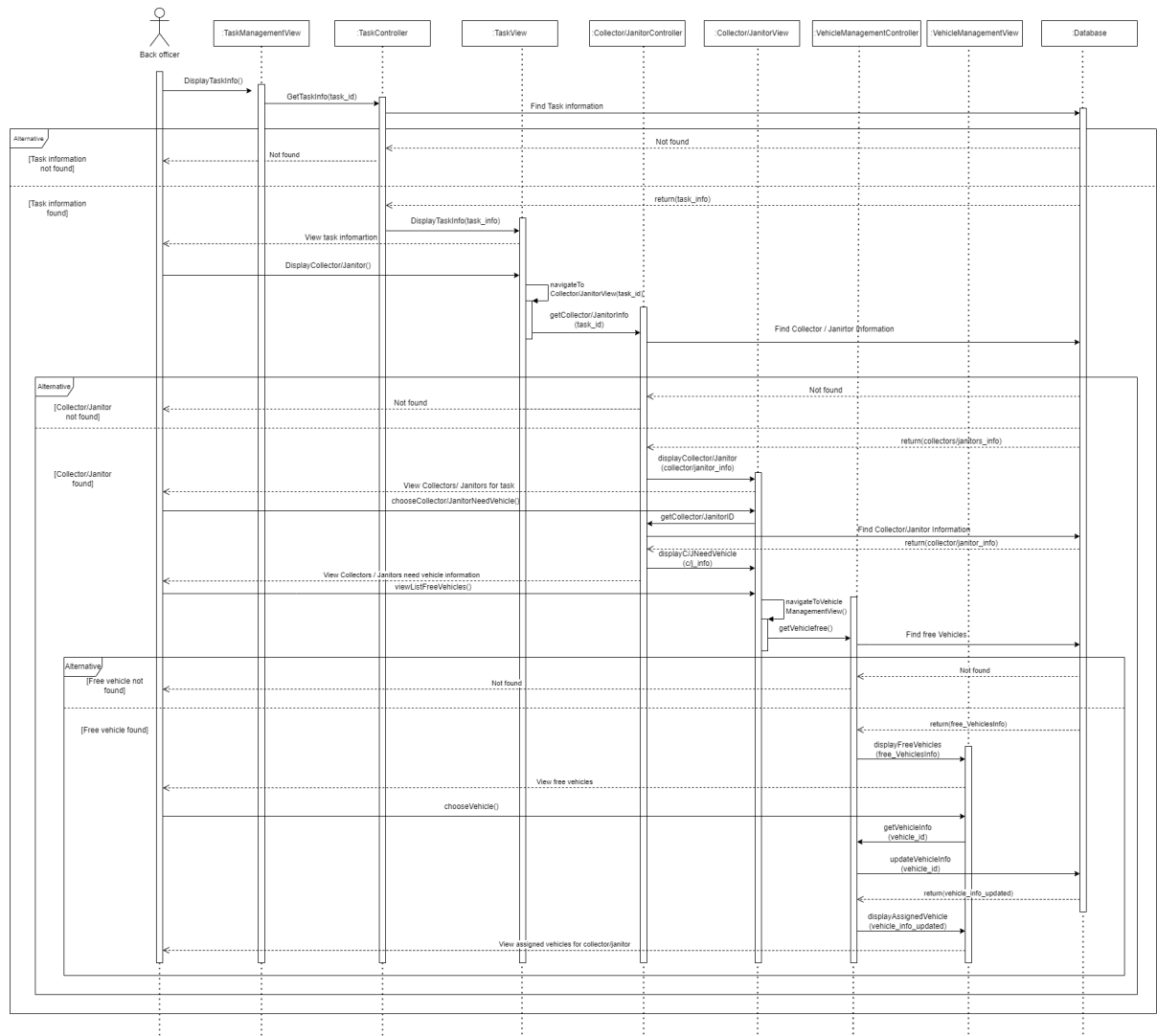
Diagram description:

- Back officer bắt đầu Task Assignment
- Back officer kiểm tra và chọn collectors/janitors đang có sẵn trên System
- Back officer chọn area và MCP cho collectors/janitors
- System tìm phương tiện đang có sẵn cho janitors ,tạo và cho thấy tuyến đường hoạt động. Sau đó, back officer sẽ chọn các janitors/collectors có thể làm việc
- Back officer mô tả và đăng tải nhiệm vụ lên System để System thông báo cho người nhận việc
- Janitors/ Collectors nhận những nhiệm vụ cần thực hiện. Nếu họ hoàn thành nhiệm vụ, họ có thể cập nhập tình trạng nhiệm vụ
- System thông báo cho người quản lý để họ có thể loại bỏ nhiệm vụ của Janitors/-Collectors và kết thúc công việc của họ
- End

3.2 Sequence Diagram

Task 2.2

Think about a possible way for a back officer to assign vehicles to janitors and collectors. Draw a sequence diagram to visualize this process



Sequence Diagram

Xem ảnh chi tiết: [Tại đây](#)

Để có thể phân phương tiện cho collector hoặc janitor, back officer phải dựa vào cụ thể từng task, trong các task sẽ có nhiều nhân viên cùng thực hiện, chọn một nhân viên cụ thể cần được phân xe, sau đó xem danh sách xe trống, cuối cùng là chọn xe phù hợp.

Mô tả các bước thực hiện cho back officer giao phương tiện cho Collectors và Janitors:

1. Back officer đang ở trang TaskManagementView, tại đó muốn xem thông tin cụ thể của một Task (về nhân viên, về công việc,...): gọi hàm **DisplayTaskInfo()**:

void của nó để mở trang **TaskView**. **TaskManagementView** sẽ không gọi trực tiếp **TaskView** mà sẽ truyền *task_id* mà người dùng chọn cho **TaskController**, sau đó **TaskController** mới vào **Database** tìm thông tin của task được chọn hiển thị.

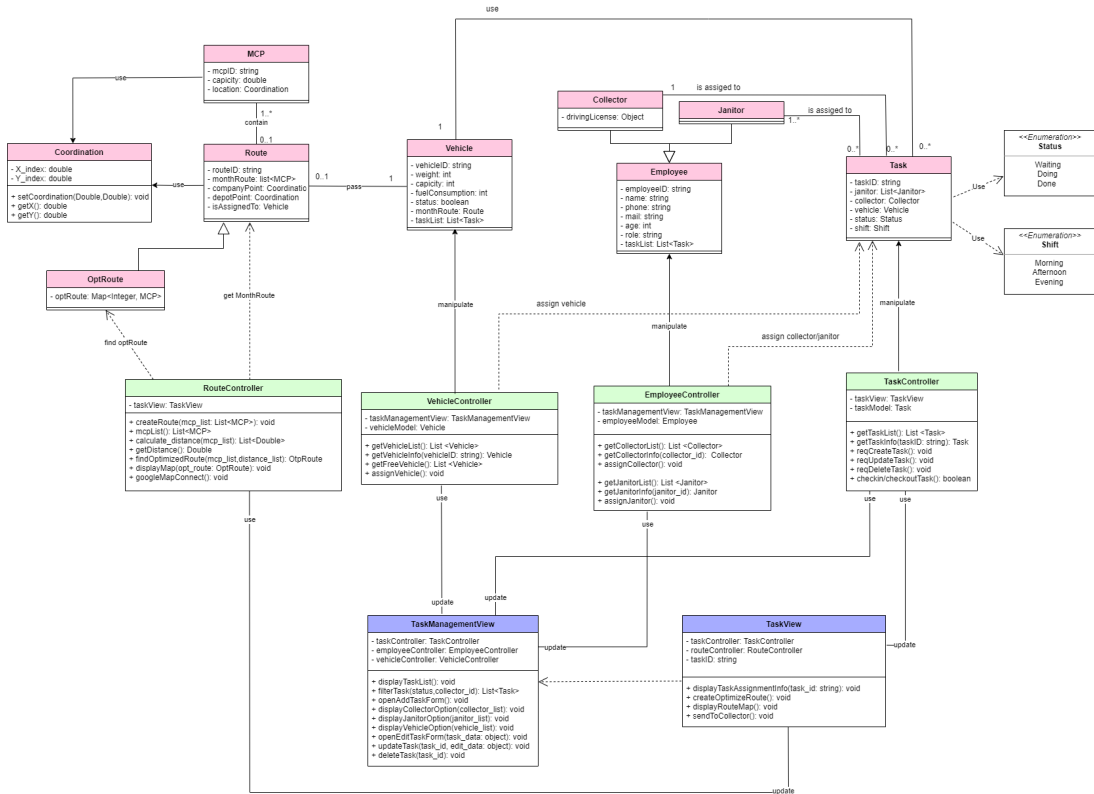
2. Khi **TaskController** vào **Database** tìm thông tin của task sẽ có 2 trường hợp xảy ra:
 - Nếu như không tìm thấy dữ liệu của task được chọn trong **Database**, **TaskController** sẽ báo lại người dùng là không tìm thấy (Not found)
 - Nếu như tìm thấy, **Database** sẽ trả về thông tin của task được chọn (*return(task_info)*), thông tin này sẽ được **TaskController** truyền cho **TaskView** và hiển thị cho **Back officer**.
3. Khi đã thấy được thông tin của Task (bao gồm công việc, tuyến đường,...), **Back officer** sẽ chọn hiển thị thông tin nhân viên cho task bằng hàm **DisplayCollector/Janitor()**, **TaskView** sẽ gọi hàm **navigateToCollector/JanitorView(task_id)** để chuyển hướng sang xem thông tin của các nhân viên cụ thể cho task đó.
4. Khi chuyển hướng, **TaskView** sẽ truyền id của task sang **Collector/JanitorController** bằng hàm **getCollector/Janitor(task_id)**, ở đây controller sẽ tìm trong **Database**:
 - Nếu không tìm thấy thông tin sẽ trả về Not found
 - Nếu tìm thấy thông tin các nhân viên phụ trách task có *task_id* tương ứng sẽ trả về **Collector/JanitorController** thông tin của nhân viên và hiển thị ra giao diện **Collector/JanitorView** bằng hàm **displayCollector/Janitor(collector/janitor_info)**.
5. Khi đã thấy được các nhân viên của task, lúc này **Back officer** sẽ chọn một nhân viên cụ thể để giao phương tiện bằng hàm **chooseCollector/JanitorNeedVehicle()**, hàm này nằm trong **Collector/JanitorView**, nó sẽ gọi hàm **getCollector/JanitorID()** trong **Collector/JanitorController** để lấy thông tin của nhân viên trong **Database**, **Database** sẽ lại trả về cho Controller thông tin cụ thể của nhân viên đó và Controller sẽ hiển thị ra View cho **Back officer** xem.
6. Khi **Back officer** thấy được thông tin của **Collector/Janitor** cụ thể cho task sẽ quyết định phân xe, đầu tiên sẽ cần ấn nút tìm xe trống, gọi hàm **viewListFreeVehicles()** xuống **Collector/JanitorController**, nó sẽ gọi hàm **navigateToVehiclesView()** để chuyển hướng sang **VehicleView**, sau đó hàm này sẽ gọi hàm **getVehiclefree()** ở **VehicleManagementController**.
7. **VehicleManagementController** sẽ lấy thông tin xe còn trống ở **Database**:
 - Nếu không còn xe trống, trả về Not found cho controller và controller sẽ thông báo cho **Back officer** không còn xe.
 - Nếu còn xe chưa dùng, trả về cho **VehicleManagementController** thông tin các xe trống (*return(free_vehicleInfo)*), **VehicleManagementController** sẽ đưa thông tin này sang **VehicleManagementView** bằng hàm **displayFreeVehicles(free_vehicleInfo)** hiển thị xe trống cho **Back officer** xem.
8. Sau khi thấy được thông tin các xe còn trống, còn khả dụng để phân cho **Collector/Janitor**, **Back officer** sẽ chọn một xe cụ thể bằng cách nhấn nút gọi

hàm **chooseVehicle()**, hàm này sẽ nằm ở **VehicleManagementView**, **VehicleManagementView** sẽ gọi hàm **getVehicleInfo(*vehicle_id*)** ở **VehicleManagementController**, truyền cho nó id để tìm trong Database thông tin của xe và cập nhật thông tin, sửa trạng thái của xe từ trống thành đang làm việc và tương ứng với *task_id* cũng như *collertor/janitor_id* bằng hàm **updateVehicleInfo(*vehicle_id*)**, sau khi Database cập nhật xong sẽ trả về thông tin đã được cập nhật(*return(vehicle_info_updated)*) cho controller, controller sẽ hiển thị ra view cho **Back officer** và kết thúc quá trình Assign Vehicle.

3.3 Class Diagram

Task 2.3

Draw a class diagram of Task Assignment module as comprehensive as possible



Class Diagram

Xem ảnh chi tiết: [Tại đây](#)

Mô tả các class và mối quan hệ giữa chúng:

- TaskManagementView:

1. **Chức năng chính:** hiển thị giao diện quản lý danh sách nhiệm vụ, có hỗ trợ lọc nhiệm vụ theo trạng thái và collector, nhận các thao tác thêm, cập nhật và xóa nhiệm vụ.

2. **Các thuộc tính:**

- taskController: TaskController
- employeeController: EmployeeController
- vehicleController: VehicleController

3. **Các phương thức:**

- [displayTaskList\(\)](#): hiển thị danh sách nhiệm vụ
- [filterTask\(status,collector_id\)](#): lọc nhiệm vụ theo trạng thái hoặc tài xế
- [openAddTaskForm\(\)](#): hiển thị form thêm nhiệm vụ
- [displayCollectorOption\(collector_list\)](#): hiển thị lựa chọn Collector cho nhiệm vụ

- `displayJanitorOption(anitor_list)`: hiển thị lựa chọn Janitor cho nhiệm vụ
- `displayVehicleOption(vehicle_list)`: hiển thị lựa chọn phương tiện di chuyển cho nhiệm vụ
- `openEditTaskForm(task_data: object)`: hiển thị Form chỉnh sửa nhiệm vụ
- `updateTask(task_id, edit_data: object)`: cập nhật nhiệm vụ
- `deleteTask(task_id)`: xoá nhiệm vụ

4. Mối quan hệ:

- **Với TaskController:** <association> chức năng điều khiển lấy, xoá nhiệm vụ từ TaskController
- **Với EmployeeController:** <association> chức năng điều khiển phân công nhiệm vụ từ EmployeeController
- **Với VehicleController:** <association> chức năng điều khiển phân công xe cho nhân viên từ VehicleController

• TaskView

1. **Chức năng chính:** hiển thị thông tin chi tiết của nhiệm vụ, nhận sự kiện yêu cầu tạo lộ trình tối ưu và hiển thị bản đồ lộ trình.

2. **Các thuộc tính:**

- `routeController`: RouteController
- `taskController`: TaskController
- `taskID`: string

3. **Các phương thức:**

- `displayTaskAssignmentInfo(task_id: string)`: hiển thị chi tiết thông tin nhiệm vụ (thời gian, phân công, trạng thái, lộ trình...)
- `createOptimizedRoute()`: nhận yêu cầu tạo lộ trình và gọi phương thức tạo lộ trình ở RouteController
- `displayRouteMap()`: hiển thị trực quan bản đồ lộ trình tích hợp trong GoogleMap
- `sendToCollector()`: nhận yêu cầu gửi lộ trình cho Collector của nhiệm vụ

4. **Mối quan hệ:**

- **Với TaskManagementView:** <dependency> được cung cấp ID nhiệm vụ dùng cho việc truy xuất thông tin nhiệm vụ
- **Với RouteController:** <association> gọi chức năng tạo lộ trình và lấy thông tin cho hiển thị lộ trình từ RouteController

• TaskController

1. **Chức năng chính:** điều khiển, xử lý luồng dữ liệu nhiệm vụ

2. **Các thuộc tính:**

- `taskView`: TaskView
- `taskModel`: Task

3. **Các phương thức:**

- `getTaskList()`: gọi truy vấn của taskModel để lấy danh sách nhiệm vụ
- `getTaskInfo(taskID: string)`: gọi truy vấn của taskModel để lấy thông tin nhiệm vụ

- `reqCreateTask()`: gọi truy vấn của taskModel để thêm nhiệm vụ
- `reqUpdateTask()`: gọi truy vấn của taskModel để cập nhật thông tin nhiệm vụ
- `reqDeleteTask()`: gọi truy vấn của taskModel để xóa nhiệm vụ
- `checkin/checkoutTask()`: kiểm tra trạng thái checkin/checkout của employee trong nhiệm vụ

4. Mỗi quan hệ:

- **Với TaskManagementView:** <association> cung cấp chức năng điều khiển luồng dữ liệu
- **Với TaskView:** <association> cung cấp chức năng điều khiển luồng dữ liệu
- **Với Task:** <association> gọi các truy vấn thao tác với dữ liệu nhiệm vụ từ Task

• EmployeeController

1. Chức năng chính: điều khiển, xử lý luồng dữ liệu nhân viên

2. Các thuộc tính:

- taskManagementView: TaskManagementView
- employeeModel: Employee

3. Các phương thức:

- `getCollectorList()`: gọi truy vấn của employeeModel để lấy danh sách Collector
- `getCollectorInfo(collector_id)`: gọi truy vấn của employeeModel để lấy thông tin Collector
- `assignCollector()`: gọi truy vấn của employeeModel để phân công Collector làm nhiệm vụ
- `getJanitorList()`: gọi truy vấn của employeeModel để lấy danh sách Janitor
- `getJanitorInfo(janitor_id)`: gọi truy vấn của employeeModel để lấy thông tin Janitor
- `assignJanitor()`: gọi truy vấn của employeeModel để phân công Janitor làm nhiệm vụ

4. Mỗi quan hệ:

- **Với TaskManagementView:** <association> cung cấp chức năng điều khiển luồng dữ liệu
- **Với Employee:** <association> gọi các truy vấn thao tác với dữ liệu từ nhân viên

• VehicleController

1. Chức năng chính: điều khiển, xử lý luồng dữ liệu phương tiện di chuyển; hỗ trợ lọc ra các xe đang trống

2. Các thuộc tính:

- taskManagementView: TaskManagementView
- vehicleModel: Vehicle

3. Các phương thức:

- `getVehicleList()`: gọi truy vấn của vehicleModel để lấy danh sách các xe

- `getVehicleInfo(vehicleID: string)`: gọi truy vấn của vehicleModel để lấy thông tin xe
- `getFreeVehicle()`: gọi truy vấn của vehicleModel để lọc và lấy ra danh sách các xe trống
- `assignVehicle()`: gọi truy vấn của vehicleModel để phân công xe làm nhiệm vụ

4. Mỗi quan hệ:

- **Với TaskManagementView**:<association> cung cấp chức năng điều khiển luồng dữ liệu
- **Với Vehicle**:<association> gọi các truy vấn thao tác với dữ liệu từ phương tiện di chuyển

● RouteController

1. **Chức năng chính:** điều khiển, xử lý luồng dữ liệu lộ trình; hỗ trợ tạo lộ trình tối ưu

2. **Các thuộc tính:**

- taskView: TaskView

3. **Các phương thức:**

- `createRoute(mcp_list: List<MCP>)`: tạo lộ trình tối ưu từ danh sách MCP
- `mcpList()`: List<MCP>: lấy danh sách MCP
- `calculate_distance(mcp_list)`: tính toán và trả về danh sách khoảng cách giữa các MCP
- `getDistance()`: lấy khoảng cách tương đối giữa 2 MCP
- `findOptimizedRoute(mcp_list, distance_list)`: cung cấp thuật toán để tìm ra lộ trình tối ưu dựa vào danh sách MCP và khoảng cách giữa chúng
- `displayMap(opt_route: OptRoute)`: hiển thị lộ trình tối ưu qua bản đồ
- `googleMapConnect()`: kết nối với GoogleMap để hiển thị trực quan

4. **Mỗi quan hệ:**

- **Với TaskView**:<association> cung cấp chức năng điều khiển luồng dữ liệu
- **Với Route**:<association> gọi các truy vấn thao tác với dữ liệu lộ trình

● Task

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu nhiệm vụ

2. **Các thuộc tính:**

- taskID: string
- janitor: List<Janitor>
- collector: Collector
- vehicle: Vehicle
- status: Status
- shift: Shift

3. **Mỗi quan hệ:**

- **Với TaskController**:<association> cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu nhiệm vụ cho TaskController

- **Với Vehicle:** một vehicle có thể được chỉ định cho nhiều Task. Một task chỉ sử dụng một vehicle

- Employee

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu nhân viên
2. **Các thuộc tính:**
 - employeeID: string
 - name: string
 - phone: string
 - mail: string
 - age: int
 - role: string
 - taskList: List<Task>
3. **Mối quan hệ:**
 - **Với EmployeeController:** <association> cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu nhân viên cho EmployeeController
 - **Với Collector:** là lớp cha
 - **Với Janitor:** là lớp cha

- Janitor

1. **Mối quan hệ:**
 - **Với Task:** một Janitor có thể được chỉ định cho nhiều Task. Một Task có thể có nhiều Janitor
 - **Với Employee:** là lớp con

- Collector

1. **Các thuộc tính:**
 - drivingLicense: object
2. **Mối quan hệ:**
 - **Với Task:** một Collector có thể được chỉ định cho nhiều Task. Một Task chỉ có một Collector
 - **Với Employee:** là lớp con

- Vehicle

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu phương tiện di chuyển
2. **Các thuộc tính:**
 - vehicleID: string
 - weight: int
 - capacity: int
 - fuelConsumption: int
 - status: boolean
 - monthRoute: Route
 - taskList: List<Task>

3. Mỗi quan hệ:

- **Với Task:** một Vehicle có thể được chỉ định cho nhiều Task. Một Task chỉ có một Vehicle
- **Với Route:**<association> một Vehicle đi qua một Route cố định trong tháng

• Route

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu lộ trình

2. **Các thuộc tính:**

- routeID: string
- monthRoute: list<MCP>
- companyPoint: Coordination
- depotPoint: Coordination
- isAssignedTo: Vehicle

3. Mỗi quan hệ:

- **Với RouteController:**<association> cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu lộ trình cho RouteController
- **Với Vehicle:**<association> một Vehicle đi qua một Route cố định trong tháng
- **Với OptRoute:** là lớp cha
- **Với Coordination:**<association> sử dụng như một kiểu dữ liệu
- **Với MCP:**<association> một Route bao gồm nhiều MCP. Một MCP thuộc một Route

• MCP

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu MCP

2. **Các thuộc tính:**

- mcpID: string
- capacity: double
- location: Coordination

3. Mỗi quan hệ:

- **Với Route:**<association> một Route bao gồm nhiều MCP. Một MCP thuộc một Route
- **Với Coordination:**<association> sử dụng như thuộc tính

• Coordination

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu tọa độ

2. **Các thuộc tính:**

- X_index: double
- Y_index: double

3. Mỗi quan hệ:

- **Với Route, MCP:**<association> được sử dụng

• OptRoute

1. **Chức năng chính:** cung cấp cách truy vấn đến dữ liệu lộ trình tối ưu
2. **Các thuộc tính:**
 - optRoute: Map<Integer, MCP>
3. **Mối quan hệ:**
 - **Với Route:** là lớp con

3.4 Thiết kế giao diện quản lý nhiệm vụ

Task 2.4

Develop MVP 1 as user interfaces of either a Desktop-view central dashboard for Task Management for back-officers OR a Mobile-view Task assignment for Janitors and Collectors. Decide yourself what to include in the view. Use a wireframe tool like Figma or Adobe XD, or Illustrator

3.4.1 Link Figma & Mockup

- Link Figma : <https://www.figma.com/Figma-Software-Engineering>
- Link Mockup: <https://www.figma.com/Mockup-Software-Engineering>

Lưu ý :

- Trong site nhân viên, khi thực hiện xong một thao tác (ví dụ như nhấn vào nút chỉnh sửa) thì sẽ được điều hướng đến các trang có liên quan. Nhưng khi đến trang cuối cùng của thao tác đó, cần phải click vào màn hình trang đó để về lại trang trước khi thực hiện thao tác. Ví dụ như thao tác xóa, khi nhấn xác nhận, để muốn quay về trang danh sách thì phải click vào màn hình.
- Đối với việc tìm kiếm nhân viên trong site nhân viên, để nhập tên hoặc ID mà không có trong danh sách, cần giữ chuột và kéo trên thanh nhập để có được **Hình 12** (Chỉ áp dụng cho Mockup)

3.4.2 Mô tả chi tiết

Chức năng lọc và xóa nhiệm vụ

Back Officer có thể vào site nhiệm vụ bằng có nhấn vào click "Nhiệm vụ" trên phần header. Ở đây, xuất hiện danh sách nhiệm vụ ứng với từng nhân viên và các trường cần lọc, cụ thể là trạng thái và ngày làm việc.

Tuyển đường

Nhiệm vụ

Nhân viên

Phương tiện

MCP

Danh sách nhiệm vụ

Trạng thái

Chọn ngày

Lọc

ID	Tên	Trạng thái	Ngày	Cà	Tùy chọn
1001	Nguyễn Văn A	Đang diễn ra	3/4/2022	Sáng	
1002	Nguyễn Văn B	Đang diễn ra	3/4/2022	Sáng	
1003	Nguyễn Văn C	Đã hoàn thành	3/4/2022	Sáng	
1004	Nguyễn Văn D	Đang diễn ra	3/4/2022	Chiều	
1005	Nguyễn Văn Q	Đang diễn ra	3/4/2022	Chiều	
1006	Trần Văn A	Đã hoàn thành	3/4/2022	Chiều	
1007	Trần Văn B	Đang diễn ra	3/4/2022	Tối	
1008	Trần Văn C	Đang diễn ra	3/4/2022	Tối	
1009	Trần Văn D	Chưa bắt đầu	3/4/2022	Tối	
1010	Trần Văn E	Đang diễn ra	3/4/2022	Tối	

Tạo mới

1-10 of 29

< >

Hình 1: *Danh sách nhiệm vụ*

Back Officer có thể lọc nhiệm vụ theo ngày hoặc theo trạng thái hoặc cả hai bằng cách chọn trường và nhấn nút "Lọc".

Tuyển đường

Nhiệm vụ

Nhân viên

Phương tiện

MCP

Danh sách nhiệm vụ

Đang diễn ra

Chọn ngày

Lọc

ID	Tên	Trạng thái	Ngày	Cà	Tùy chọn
1001	Nguyễn Văn A	Đang diễn ra	3/4/2022	Sáng	
1002	Nguyễn Văn B	Đang diễn ra	3/4/2022	Sáng	
1004	Nguyễn Văn D	Đang diễn ra	3/4/2022	Chiều	
1005	Nguyễn Văn Q	Đang diễn ra	3/4/2022	Chiều	
1007	Trần Văn B	Đang diễn ra	3/4/2022	Tối	
1008	Trần Văn C	Đang diễn ra	3/4/2022	Tối	
1010	Trần Văn E	Đang diễn ra	3/4/2022	Tối	

Tạo mới

1 - 6 of 6

< >

Hình 2: *Lọc theo trạng thái*

[Tuyến đường](#)
[Nhiệm vụ](#)
[Nhân viên](#)
[Phương tiện](#)
[MCP](#)

Danh sách nhiệm vụ

Lọc

ID	Tên	Trạng thái	Ngày	Ca	Tùy chọn
1021	Nguyễn Văn A	Chưa bắt đầu	5/4/2022	Sáng	
1022	Nguyễn Văn B	Chưa bắt đầu	5/4/2022	Sáng	
1023	Nguyễn Văn C	Chưa bắt đầu	5/4/2022	Sáng	
1024	Nguyễn Văn D	Chưa bắt đầu	5/4/2022	Chiều	
1025	Nguyễn Văn Q	Chưa bắt đầu	5/4/2022	Chiều	

Tạo mới

1-5 of 5



Copyright © 2023 UWC2.0

0123456789
uwc2_0@gmail.com

Hình 3: Lọc theo ngày

Trên mỗi hàng nhiệm vụ, sẽ có 2 trong nút ở cột tùy chọn với chức năng là chỉnh sửa và xóa. Đối với nút xóa, khi nhấn vào sẽ hiện ra một bảng nhỏ để xác nhận người dùng có chắc chắn với quyết định của mình chưa. Còn đối với nút chỉnh sửa, vì phần này có nhiều thông tin nên sẽ được trình bày ở phía dưới.

[Tuyến đường](#)
[Nhiệm vụ](#)
[Nhân viên](#)
[Phương tiện](#)
[MCP](#)

Danh sách nhiệm vụ

Lọc

ID	Tên	Trạng thái	Ngày	Ca	Tùy chọn
1021	Nguyễn Văn A	Chưa bắt đầu	5/4/2022	Sáng	
1022	Nguyễn Văn B	Chưa bắt đầu	5/4/2022	Sáng	
			5/4/2022	Sáng	
			5/4/2022	Chiều	
			5/4/2022	Chiều	
			6/4/2022	Chiều	
			6/4/2022	Tối	
1028	Trần Văn C	Chưa bắt đầu	6/4/2022	Tối	
1029	Trần Văn D	Chưa bắt đầu	6/4/2022	Tối	

Tạo mới

21-29 of 29

Copyright © 2023 UWC2.0

0123456789
uwc2_0@gmail.com

Hình 4: Xác nhận xóa nhiệm vụ

Giao diện tạo nhiệm vụ mới

Back Officer có thể nhấn vào nút “Tạo mới” để thêm một nhiệm vụ. Tại đây Back Officer có thể chọn ngày, ca làm việc, tài xế và phương tiện. Sau khi chọn xong tài xế và phương tiện, hệ thống sẽ hiển thị các thông tin chi tiết của tài xế (ID, bằng lái) và của phương

tiện (tải trọng tối đa, mức tiêu thụ). Đồng thời hệ thống sẽ hiện thông tin về các MCP đi qua trong lộ trình sau khi chọn phương tiện (giả sử rằng mỗi phương tiện đã được gán một lộ trình tương ứng trước khi giao nhiệm vụ cho nhân viên)

Hình 5: Giao diện tạo nhiệm vụ mới cho Collector

Sau đó, Back Officer nhấn vào nút “Tiếp theo” để giao nhiệm vụ cho Janitor bằng cách lựa chọn Janitor trong danh sách đã có sẵn. Hệ thống cho phép tối đa 3 Janitors có thể làm việc cho mỗi MCP. Sau khi hoàn thành giao nhiệm vụ cho Janitor thì Back Officer nhấn vào nút “Lưu nhiệm vụ” để lưu lại các thông tin đã chọn.

Hình 6: Giao diện tạo nhiệm vụ mới cho Collector

Giao diện xem chi tiết nhiệm vụ

Back officer có thể nhấn vào một nhiệm vụ bất kì trong danh sách nhiệm vụ để xem các

thông tin chi tiết của nhiệm vụ đó như: ngày và ca làm việc, tên và ID của nhân viên, mã lộ trình, mã phương tiện, địa chỉ của các MCP, trạng thái của hoàn thành nhiệm vụ của Collector và của các Janitor. Bên cạnh đó hệ thống cũng hiển thị tuyến đường tối ưu và độ dài của tuyến đường tối ưu đó. Nếu trạng thái của nhiệm vụ là “Chưa bắt đầu” (nghĩa là tất cả các trạng thái của Collector và Janitors đều là “Chưa bắt đầu”) thì Back Officer có thể nhấn vào nút “Gửi thông báo” để gửi thông tin nhiệm vụ đến các nhân viên, đồng thời nút “Gửi thông báo” này sẽ được chuyển thành nút “Xác nhận” xác nhận nhân viên đã check-out thành công.

Tuyến đường
Nhiệm vụ
Nhân viên
Phương tiện
MCP

Chi tiết nhiệm vụ #1001
 Ngày: 3/4/2022
 Ca: Sáng

Tài xế: Nguyễn Văn A
ID nhân viên: C001
 Mã phương tiện: 51A-12345
Mã lộ trình: R168
 Số lượng MCP: 3
Trạng thái: Chưa bắt đầu

Lộ trình tối ưu:
 Depot: 227 Bình Long, P. Phú Thạnh, Q. Tân Phú, TP. HCM
 MCP1: 332 Thoại Ngọc Hầu, P. Phú Trung, Q. Tân Phú, TP. HCM
 MCP2: 114 Tân Kỳ Tân Quý, P. Tân Sơn Nhì, Q. Tân Phú, TP. HCM
 MCP3: Số 6 Đường CN1, P. Bình Hưng Hòa, Q. Tân Phú, TP. HCM
 Disposal Facility: 12 Tây Thạnh, P. Tây Thạnh, Q. Tân Phú, TP. HCM

Phân công Janitor

	Nhân viên 1	Nhân viên 2	Nhân viên 3	Trạng thái
MCP1	Trần Thị A J140	Trần Thị C J142	Đỗ Văn C J146	Chưa bắt đầu
MCP2	Trần Thị B J141	Đỗ Văn A J144		Chưa bắt đầu
MCP3		Đỗ Văn B J145		Chưa bắt đầu

Tổng độ dài lộ trình: 11.4km

Quay lại Gửi thông báo

Hình 7: Giao diện chi tiết nhiệm vụ khi Back Officer chưa gửi thông báo

Sau khi tất cả các trạng thái của các nhiệm vụ của Collector và Janitors đều là “Đã hoàn thành” thì Back Officer có thể xác nhận nhiệm vụ hoàn thành bằng cách nhấn vào nút “Xác nhận”, sau đó hệ thống sẽ hiển thị Back Officer đã xác nhận thành công và nút “Xác nhận” sẽ được chuyển thành nút “Đã xác nhận”.

UWC

Tuyến đường

Nhiệm vụ

Nhân viên

Phương tiện

MCP

Chi tiết nhiệm vụ

#1003

Ngày: 3/4/2022

Ca: Sáng

Tài xế: Nguyễn Văn C

ID nhân viên: C009

Mã phương tiện: 51A-12346

Mã lộ trình: R168

Số lượng MCP: 3

Trạng thái: Đã hoàn thành

Lộ trình tối ưu:
 Depot: 227 Bình Long, P. Phú Thạnh, Q. Tân Phú, TP. HCM
 MCP1: 332 Thoại Ngọc Hầu, P. Phú Trung, Q. Tân Phú, TP. HCM
 MCP2: 114 Tân Kỳ Tân Quý, P. Tân Sơn Nhì, Q. Tân Phú, TP. HCM
 MCP3: Số 6 Đường CN1, P. Bình Hưng Hòa, Q. Tân Phú, TP. HCM
 Disposal Facility: 12 Tây Thạnh, P. Tây Thạnh, Q. Tân Phú, TP. HCM

Phân công Janitor

	Nhân viên 1	Nhân viên 2	Nhân viên 3	Trạng thái
MCP1	Trần Thị A J140	Trần Thị C J142	Đỗ Văn C J146	Đã hoàn thành
MCP2	Trần Thị B J141	Đỗ Văn A J144		Đã hoàn thành
MCP3		Đỗ Văn B J145		Đã hoàn thành

Quay lại

Tổng độ dài lộ trình: 11.4km

Đã xác nhận

Xác nhận thành công!

Copyright © 2023 UWC2.0

0123456789

uwc2_0@gmail.com

Hình 8: Giao diện chi tiết nhiệm vụ khi Back Officer xác nhận thành công

Giao diện chỉnh sửa nhiệm vụ

Back officer có thể nhấn vào nút ở một nhiệm vụ bất kì trong danh sách nhiệm vụ để chỉnh sửa nhiệm vụ đó (Hệ thống chỉ cho phép chỉnh sửa nếu trạng thái của nhiệm vụ đó là "Chưa bắt đầu" và Back officer chưa gửi thông báo). Hệ thống sau đó sẽ hiện ra một bảng nhỏ để chỉnh sửa nhiệm vụ, giao diện chỉnh sửa nhiệm vụ giống với giao diện tạo nhiệm vụ ở trên.

UWC

Tuyến đường

Nhiệm vụ

Nhân viên

Phương tiện

MCP

Danh sách nhiệm vụ

Tìm kiếm

ID nhiệm vụ

Trạng thái

Chọn ngày

Lọc

Chỉnh sửa nhiệm vụ

#1040

X

MCP1

Địa chỉ: 25 Trần Hưng Đạo, P. Nguyễn Cư Trinh, Quận 1

Nguyễn Xuân Anh

Nguyễn Xuân Bắc

Chọn nhân viên

MCP3

Địa chỉ: 53 Lý Chính Thắng, Phường 8, Quận 3

Chọn nhân viên

Chọn nhân viên

Nguyễn Xuân An

Nguyễn Xuân Cường

Nguyễn Xuân Dũng

Nguyễn Văn Phú

MCP5

Địa chỉ: 123 Hồng Lạc, Phường 10, Quận Tân Bình

Chọn nhân viên

Chọn nhân viên

Chọn nhân viên

Quay lại

Lưu nhiệm vụ

1029

Trần Văn D

Chưa bắt đầu

6/4/2022

Tối

Tạo mới

21-30 of 30

Copyright © 2023 UWC2.0

0123456789

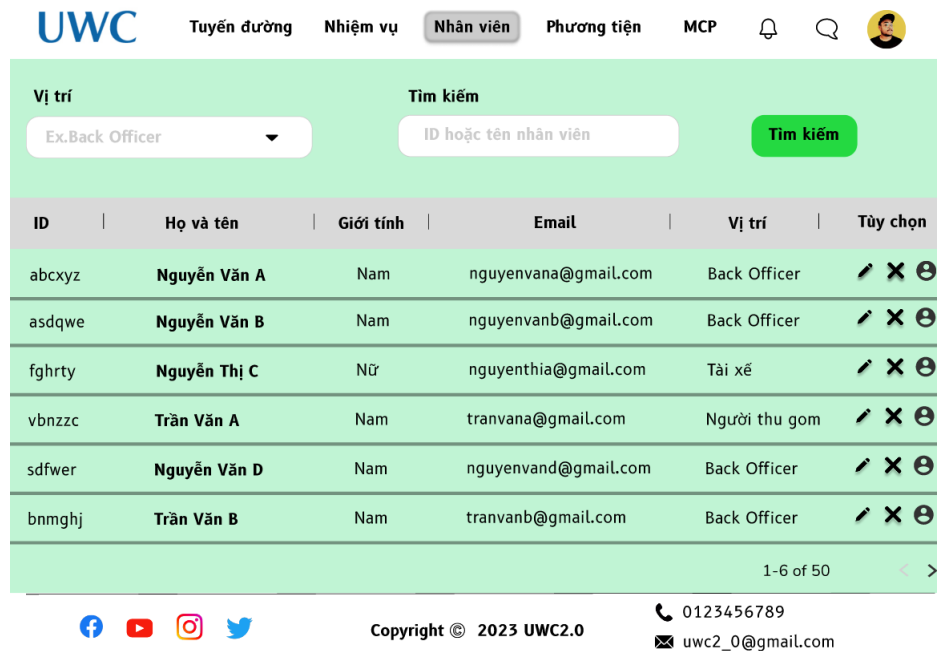
uwc2_0@gmail.com



















Hình 9: Giao diện chỉnh sửa nhiệm vụ cho Janitor

Quản lý nhân viên

Tại màn hình trang chủ, khi Back Officer nhấn vào "Nhân viên" thì sẽ được điều hướng

đến site danh sách nhân viên.



ID	Họ và tên	Giới tính	Email	Vị trí	Tùy chọn
abcxyz	Nguyễn Văn A	Nam	nguyenvana@gmail.com	Back Officer	  
asdqwe	Nguyễn Văn B	Nam	nguyenvanb@gmail.com	Back Officer	  
fghrty	Nguyễn Thị C	Nữ	nguyenthia@gmail.com	Tài xế	  
vbnzcc	Trần Văn A	Nam	tranvana@gmail.com	Người thu gom	  
sdfwer	Nguyễn Văn D	Nam	nguyenvand@gmail.com	Back Officer	  
bnmgjh	Trần Văn B	Nam	tranvanb@gmail.com	Back Officer	  

1-6 of 50

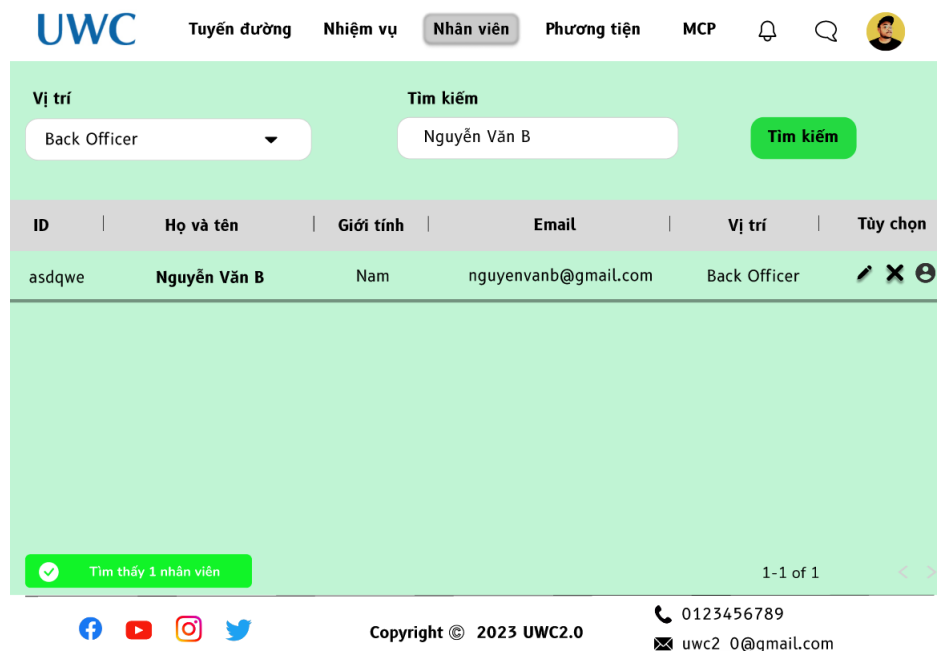
Facebook YouTube Instagram Twitter




Copyright © 2023 UWC2.0

0123456789
uwc2_0@gmail.com

Hình 10: Giao diện danh sách nhân viên

Tại đây, có hai trường vị trí và tìm kiếm theo ID hoặc nhân viên. Back Officer có thể tìm kiếm nhân viên theo hai trường đó bằng cách nhấn vào nút "Tìm kiếm"



ID	Họ và tên	Giới tính	Email	Vị trí	Tùy chọn
asdqwe	Nguyễn Văn B	Nam	nguyenvanb@gmail.com	Back Officer	  

Tìm thấy 1 nhân viên

1-1 of 1

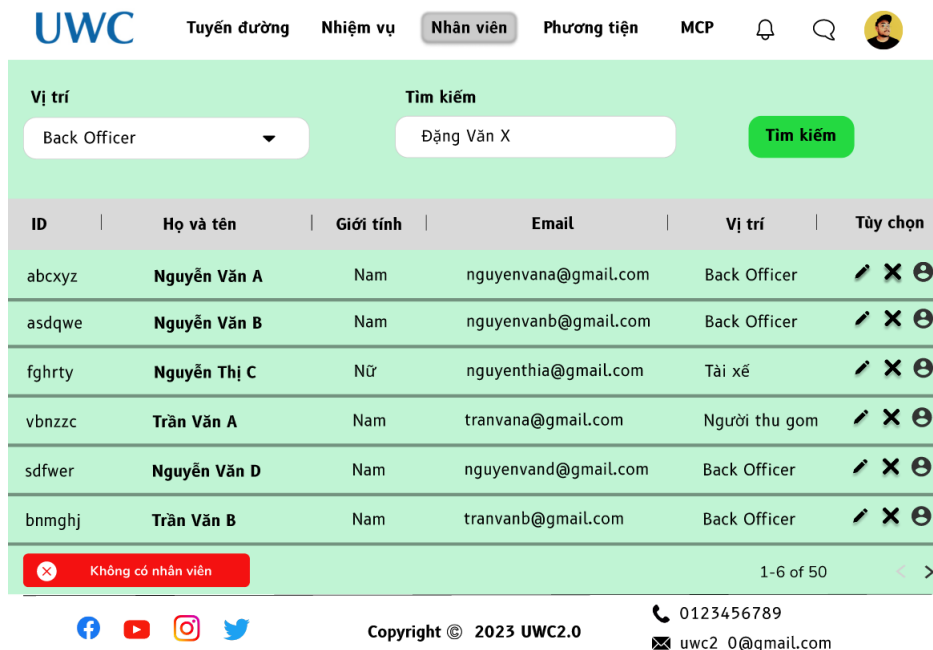
Facebook YouTube Instagram Twitter

Copyright © 2023 UWC2.0

0123456789
uwc2_0@gmail.com

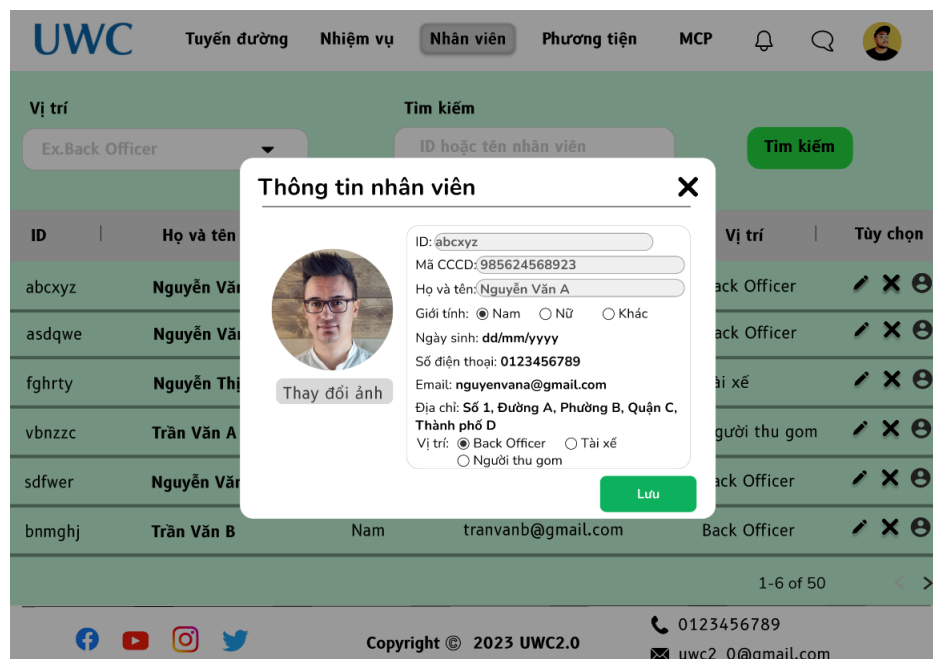
Hình 11: Tìm thấy nhân viên

Trong trường hợp không có nhân viên cần tìm kiếm, sẽ có thông báo không có nhân viên đó trong danh sách.

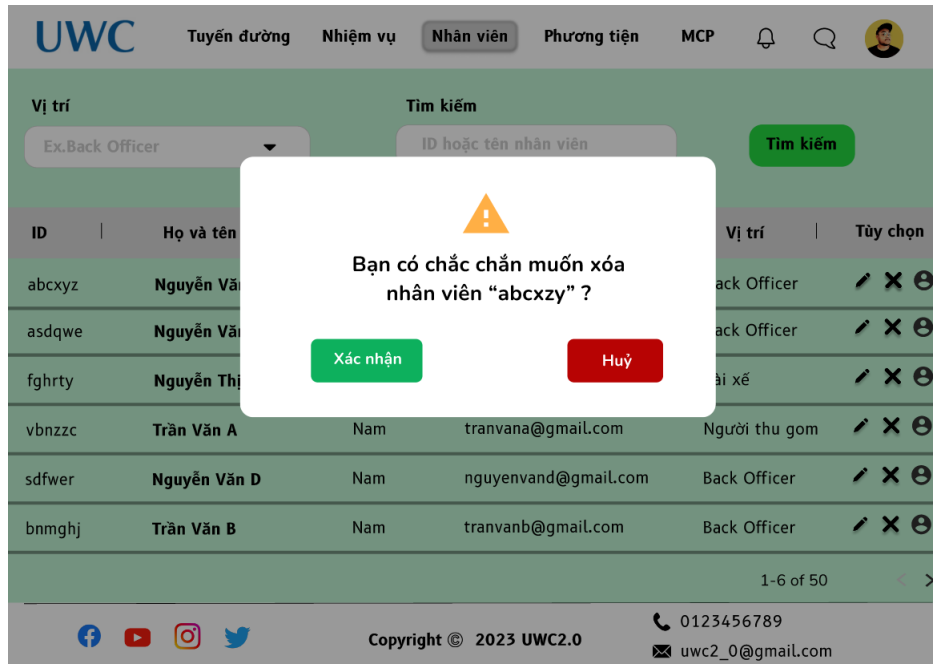


Hình 12: Không có nhân viên cần tìm

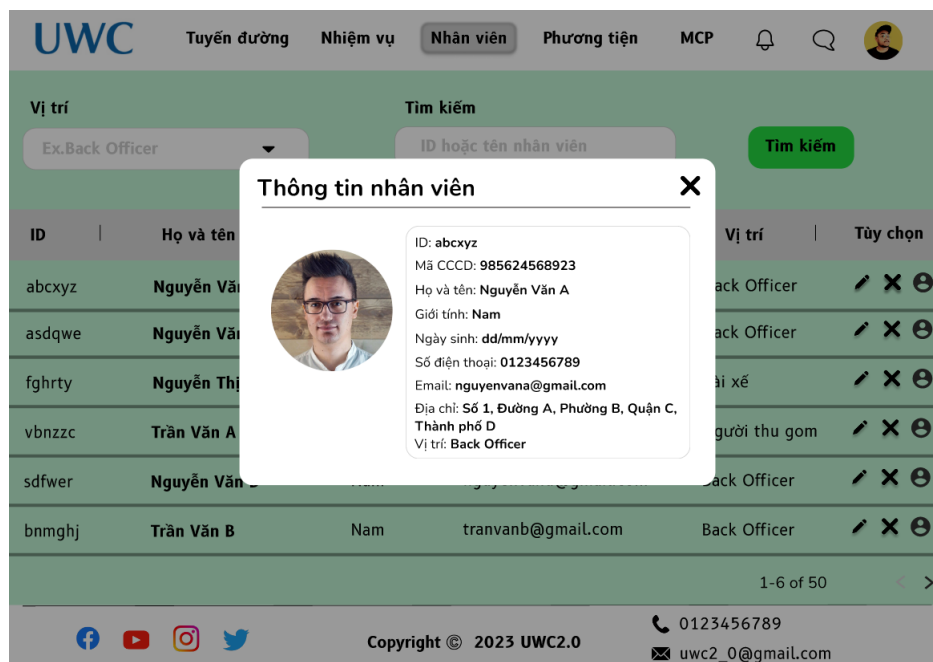
Ngoài ra trên mỗi hàng ứng với từng nhân viên, có ba nút trên cột "Tùy chọn", có chức năng lần lượt là chỉnh sửa thông tin nhân viên, xóa nhân viên và xem thông tin nhân viên. Đối với nút chỉnh sửa, khi Back Officer nhấn vào, sẽ xuất hiện một bảng nhỏ chứa thông tin, đồng thời có thể sửa lại những thông tin đó. Đối với nút xóa, tương tự như các nút xóa khác, cũng sẽ có một bảng xác nhận xóa. Đối với nút xem thông tin nhân có một bảng nhỏ hiện ra, chứa những thông tin của nhân viên như ID, tên, số điện thoại, vị trí, ảnh đại diện, ...



Hình 13: Giao diện chỉnh sửa thông tin nhân viên



Hình 14: Bảng xác nhận xóa thông tin nhân viên



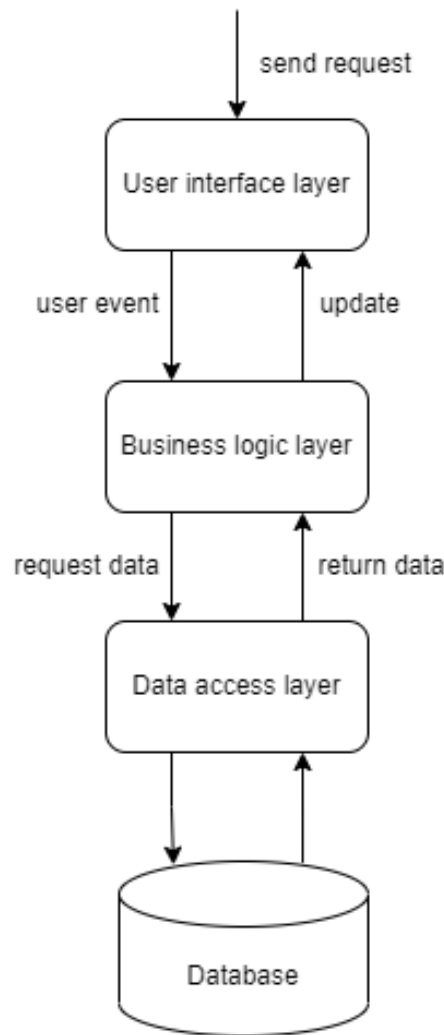
Hình 15: Giao diện thông tin nhân viên

4 Architecture design

Task 3.1

Use a layered architecture to design the UWC 2.0 system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/ APIs

4.1 Use a layered architecture to design the UWC 2.0 system



Kiến trúc phân lớp

Xem ảnh chi tiết: [Tại đây](#)

Kiến trúc theo mô hình phân lớp (layered architecture) trong đó:

- User interface layer: đây là phần giao diện người dùng, có nhiệm vụ tiếp nhận yêu cầu, lắng nghe các sự kiện từ tương tác của người dùng bên ngoài và truyền tải thông điệp từ client tới server. Đồng thời hiển thị ra những kết quả đã được các tầng tiếp theo xử lý và cập nhật.
- Business logic layer: đây là tầng nhận dữ liệu và thực hiện xử lý các logic nghiệp vụ
 - Thu nhận và xử lý các yêu cầu từ người dùng đến từ tầng trên rồi thực hiện các cập nhật tương ứng.
 - Chứa các phương thức chính của hệ thống, thực hiện các chức năng cụ thể để phục vụ hệ thống.
 - user interface authentication: xác thực người dùng dựa vào những thông tin đã nhập.
 - user interface authorization: cấp quyền cho người dùng tương ứng (back officer, janitor, collector).

- Data access layer: là nơi tương tác với cơ sở dữ liệu để lấy ra những dữ liệu cần thiết cho tầng trên.

4.2 Describe how will you present your User Interface

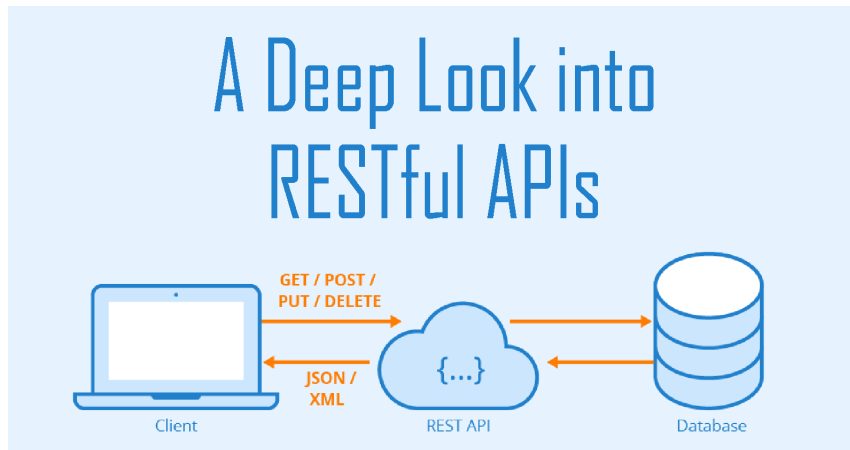
Trang web UWC 2.0 là một nơi cung cấp các công cụ quản lý công việc của nhân viên dọn rác – những người có ít kiến thức về các ứng dụng công nghệ. Vì thế, giao diện người dùng của trang web cần phải được thiết kế một cách đơn giản về dễ sử dụng để đảm bảo rằng nhân viên có thể sử dụng trang web một cách hiệu quả. Đầu tiên, khi truy cập trang web, ta sẽ thấy một cửa sổ dùng để đăng nhập vào web. Cửa sổ sẽ bao gồm 2 ô nhập thông tin tài khoản, mật khẩu và 1 nút đăng nhập. Người dùng sẽ nhập các thông tin đăng nhập (tên tài khoản, mật khẩu). Nếu như người nhập sai tên tài khoản hoặc mật khẩu thì sẽ xuất hiện thông báo sai tài khoản hoặc mật khẩu tương ứng. Ngược lại, nếu nhập đúng thông tin đăng nhập thì trang web sẽ chuyển hướng đến màn hình trang chủ. Sau khi đăng nhập, trang web sẽ chuyển đến tab quản lý nhiệm vụ của nhân viên. Phía bên trên góc phải là thông tin của nhân viên đang đăng nhập vào trang web. Phía bên trái trang web sẽ hiển thị các tab của trang web bao gồm danh sách nhiệm vụ, người dùng có thể nhấn vào những tab đó để chuyển tới những mục mình cần. Đầu tiên, ở tab quản lý nhiệm vụ của nhân viên, trang web sẽ hiển thị danh sách tasks của nhân viên. Mỗi task sẽ bao gồm ID nhiệm vụ, trạng thái, tên nhân viên đang thực hiện, ID MCPs mà nhân viên làm việc và ID của phương tiện nhân viên đang sử dụng. Bên trên danh sách sẽ có 1 bộ lọc theo thời gian và trạng thái của nhiệm vụ. Tiếp theo sẽ là tab nhân viên, tab này chứa thông tin của các nhân viên bao gồm: ID, tên, giới tính, ngày sinh, SDT, email để back officers có thể dễ dàng quản lý. Tương tự như tab trên, tab phương tiện cũng sẽ chứa thông tin các phương tiện của công ty quản lý như trọng lượng, mức tiêu thụ, sức chứa tối đa. Cuối cùng, ở tab MCPs, nó sẽ là một bản đồ được đánh dấu các địa điểm MCPs và đường đi mà nhân viên đi qua

4.3 Describe how will you store your data

Lưu trữ data bằng MySQL là một trong những phương pháp phổ biến trong lập trình Web, nó là một công cụ đáng tin cậy và hiệu quả để quản lý khối lượng lớn các data. MySQL là một công cụ mạnh mẽ trong hệ thống quản lý database cho phép người dùng xử lý nhiều loại dữ liệu và cung cấp nhiều tính năng để quản lý dữ liệu như hỗ trợ các giao dịch, chỉ số và lưu trữ các thông tin sản phẩm. Để sử dụng MySQL để lưu trữ dữ liệu, trước hết chúng ta cần set up MySQL server and client trên máy tính cá nhân hay server. Khi mọi thứ đã được chuẩn bị, chúng ta có thể tạo ra các database để lưu trữ dữ liệu của mình. Database là tập hợp các bảng liên quan tới nhau và được lưu trữ dưới nhiều dạng dữ liệu khác nhau. Mỗi bảng trong database gồm các cột thể hiện cấu trúc của data và mỗi hàng biểu diễn một tập dữ liệu đơn. Để tương tác với MySQL và quản lý dữ liệu, chúng ta có thể sử dụng các lệnh SQL để tạo, thay đổi và truy xuất dữ liệu. SQL là một chuẩn ngôn ngữ được sử dụng để quản lý các tác vụ liên quan tới database và chúng được hỗ trợ bởi MySQL. Các lệnh trong SQL có thể được sử dụng để tạo bảng, thêm data vào bảng, thay đổi dữ liệu, và truy xuất data từ bảng. Thêm vào đó, nhờ vào các câu lệnh SQL, MySQL cũng đồng thời hỗ trợ các tính năng nâng cao để quản lý data như liên kết, tạo liên kết, xem và các thủ tục lưu trữ. Các tính năng đó có thể giúp chúng ta quản lý data một cách hiệu quả và nâng cao hiệu suất.

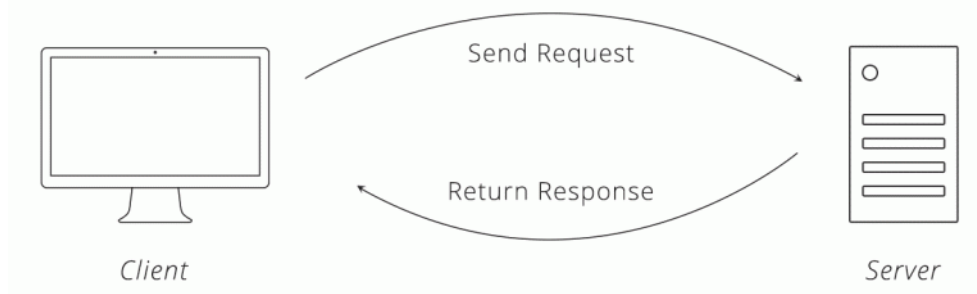
4.4 Describe how you will access to external services/ APIs

Về mặt kỹ thuật, API là viết tắt của Giao diện lập trình ứng dụng (Application Programming Interface), một trung gian phần mềm cho phép hai ứng dụng giao tiếp với nhau, có thể sử dụng cho web-based system, operating system, database system, computer hardware, or software library. Ở dạng đơn giản nhất, API là giao diện cho phép một ứng dụng giao tiếp với ứng dụng khác thông qua các lệnh đơn giản và cách các lệnh này được gửi và định dạng mà dữ liệu được truy xuất thông qua API có thể khác với **API SOAP** hoặc **REST**.



REST là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo ra giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE ... đến một URL để xử lý dữ liệu.

Cuộc sống của HTTP xoay quanh vòng luân chuyển: Request và Response. Client gửi request, server gửi lại response là liệu server có thể làm được những gì mà client muốn hay ko. Và API được xây dựng trên chính 2 thành phần: Request và Response. Cấu trúc của mỗi thành phần :

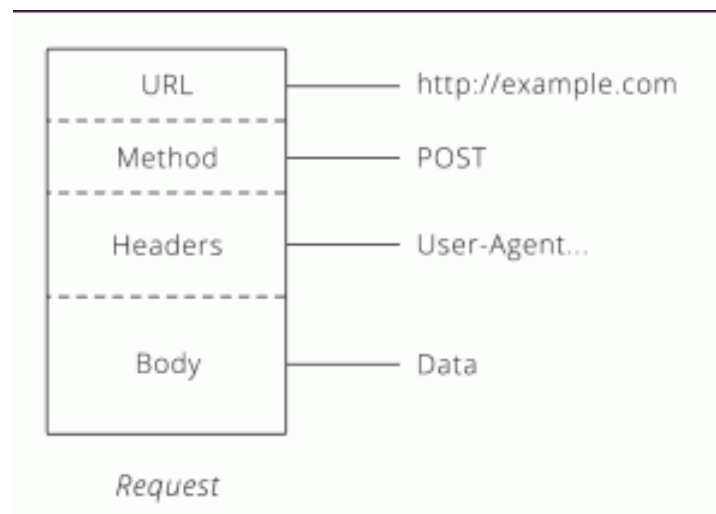


Request

Một request đúng chuẩn cần có 4 thứ:

- URL
- Method
- Headers

- Body



Hình 16: Caption

Chi tiết như sau :

- **URL** là 1 địa chỉ duy nhất cho 1 thứ (dùng danh từ), có thể là web page, image hoặc video. API mở rộng cái ý tưởng gốc của URL cho những thứ khác, ví dụ customers, products. Và như thế client dễ dàng cho server biết cái nó muốn là cái gì, những cái này còn được gọi chung là “resources” – nguồn lực.
- **Method** là hành động client muốn tác động lên “resources”, và nó thường là động từ. Có 4 loại method hay được dùng:
 - GET: Yêu cầu server đưa lại resource. Ví dụ như vào facebook và chúng ta kéo xuống thì sẽ có new feeds.
 - POST: Yêu cầu server cho tạo ra 1 resource mới. Ví dụ: đăng ký 1 chuyến đi ở GrabBike.
 - PUT: Yêu cầu server cho sửa / thêm vào resource đã có trên hệ thống. Ví dụ: Edit 1 post ở trên facebook.
 - DELETE: Yêu cầu server cho xóa 1 resource.
- **Headers**: nơi chứa các thông tin cần thiết của 1 request nhưng end-users không biết có sự tồn tại của nó. Ví dụ: độ dài của request body, thời gian gửi request, loại thiết bị đang sử dụng, loại định dạng response mà client có đọc được.
- **Body**: nơi chứa thông tin mà client sẽ điền. Giả sử bạn đặt 1 cái bánh pizza, thì thông tin ở phần body sẽ là: Loại bánh pizza, kích cỡ, số lượng đặt.

Respond

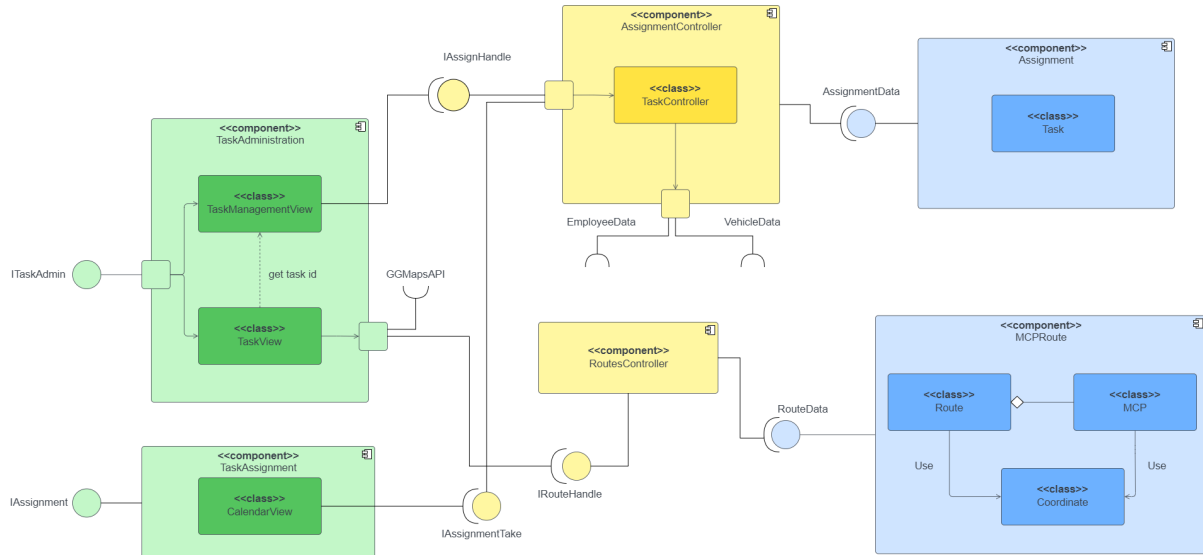
Sau khi nhận được request từ phía client, server sẽ xử lý cái request đó và gửi ngược lại cho client 1 response. Cấu trúc của 1 response tương đối giống phần request nhưng Status code sẽ thay thế cho URL và Method. Tóm lại, nó có cấu trúc 3 phần:

- Status code
- Headers
- Body

Task 3.2

Draw a component diagram for the Task Assignment module

4.5 Draw a component diagram for the Task Assignment module



Component diagram for Task Assignment module

Xem bản vẽ chi tiết: [Tại đây](#)

Phân tích Component Diagram:

1. Tầng View

(a) TaskAdministration

- **Chức năng chính:** hiện thực các class cung cấp giao diện quản lý ca làm việc và nhiệm vụ.
- **Gồm các class:** **TaskManagementView**, **TaskView**. Class **TaskManagementView** cung cấp giao diện xem thông tin của các task trong hệ thống (xem danh sách nhiệm vụ, phân nhiệm vụ mới, xóa, sửa nhiệm vụ); theo dõi danh sách các thông báo cập nhật tiến độ nhiệm vụ gửi từ Collector/Janitor để xác nhận cho chúng. Class **TaskView** cung cấp giao diện hiển thị thông tin cụ thể của một nhiệm vụ, bao gồm thêm các tính năng thêm nhân viên vào nhiệm vụ, giao xe và nhận yêu cầu tạo lộ trình tối ưu cho nhiệm vụ để hiển thị lên Bản đồ lộ trình và gửi Collector.
- **Mối quan hệ giữa các thành phần:** Class **TaskView** dựa vào ID nhiệm vụ được cung cấp bởi **TaskManagementView** để yêu cầu thông tin chi tiết của nhiệm vụ (bao gồm thông tin nhân viên thực hiện nhiệm vụ, thông tin của phương tiện cho nhiệm vụ, lộ trình cố định của phương tiện sử dụng bởi nhiệm vụ dùng cho việc yêu cầu tạo lộ trình tối ưu).
- **TaskManagementView**, **TaskView** hiện thực interface **ITaskAdmin**.

- Cả component yêu cầu interface **IAssignHandle**. **TaskView** yêu cầu thêm các interface liên quan đến tạo và hiển thị lộ trình gồm **GoogleMapAPI** và **IRouteHandle**.

(b) **TaskAssignment**

- **Chức năng chính:** hiện thực các class cung cấp giao diện xem lịch làm việc cá nhân và giao diện điểm danh cho ca làm việc.
- **Gồm class:** **CalendarView**, cung cấp giao diện xem lịch làm việc cá nhân: xem danh sách các nhiệm vụ và chi tiết từng nhiệm vụ được giao cho mình, nhận yêu cầu cập nhật tiến độ cho nhiệm vụ khi hoàn thành công việc.
- Các thành phần trong component hiện thực interface **IAssignment**.
- Các thành phần trong component yêu cầu interface **IAssignmentTake**

2. Tầng Controller

(a) **TaskController**

- **Chức năng chính:** hiện thực các class cung cấp phương thức xử lý sự kiện và điều khiển luồng dữ liệu liên quan đến phân công công việc (nhiệm vụ, thông báo cập nhật tiến độ).
- **Gồm class:** **TaskController** cung cấp các phương thức điều khiển luồng dữ liệu nhiệm vụ và thông báo cập nhật tiến độ nhiệm vụ.
- Component hiện thực interface **IAssignHandle**.
- **TaskController** yêu cầu các interface liên quan đến việc cung cấp danh sách các lựa chọn về nhân viên và phương tiện để phân công cho nhiệm vụ: **EmployeeData**, **VehicleData**.

(b) **RouteController**

- **Chức năng chính:** hiện thực các phương thức điều khiển dữ liệu về Lộ trình và tìm lộ trình tối ưu dựa trên lộ trình cố định hằng tháng của phương tiện và sức chứa hiện tại của các MCP trong lộ trình.
- Component hiện thực interface **IRouteHandle**.
- Component yêu cầu interface **RouteData**.

3. Tầng Model

(a) **Assignment**

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu liên quan đến phân công công việc (ca, nhiệm vụ, thông báo cập nhật tiến độ).
- **Gồm class:** **Task (Nhiệm vụ)**
- Component hiện thực interface **AssignmentData**.

(b) **MCPRoute**

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu liên quan đến MCP và lộ trình.
- Gồm các class: **Route (Lộ trình)**, **MCP (Điểm thu gom)**, **Coordinate(Tọa độ)**.
- Component hiện thực interface **RouteData**.

4. Các Interface

Tên interface	Chức năng	Các phương thức
ITaskAdmin	Cung cấp giao diện giúp Back Officer quản lý phân công công việc.	renderTaskManagementView(); renderTaskView(task_id); renderAddTaskView(); renderEditTaskView(task_id);
GoogleMapAPI	Cung cấp Google Map API giúp tích hợp bản đồ trực quan hiển thị lộ trình tối ưu.	displayGoogleMap(opt_route: Route)
IAssignment	Cung cấp giao diện giúp Collector và Janitor xem lịch làm việc và thực hiện điểm danh.	renderCalendarView();
IAssignHandle	Cung cấp các phương thức giúp điều khiển luồng dữ liệu phân công công việc liên quan đến các tác vụ quản lý của Back officer.	getTaskList(): List<Task> getTaskInfo(task_id: Integer): Task getAssignOption(): Object requestCreateTask(task_data: Object): void requestUpdateTask(task_id: Integer, edit_data: Object): void requestDeleteTask(task_id: Integer): void
IAssignmentTake	Cung cấp các phương thức giúp điều khiển luồng dữ liệu công việc liên quan đến các tác vụ xem và cập nhật tiến độ của Collector/Janitor.	getTaskList(): List<Task> getTaskInfo(task_id: Integer): Task updateCheckin(shift_id: Integer, employee_id: Integer): void updateCheckout(shift_id: Integer, employee_id: Integer): void
IRouteHandle	Cung cấp phương thức cho tác vụ yêu cầu tạo lộ trình tối ưu.	createRoute(mcp_list: List<MCP>): void
EmployeeData	Cung cấp API lấy dữ liệu nhân viên để phân công cho nhiệm vụ	getJanitorList(): List<Janitor> getCollectorList(): List<Collector>
VehicleData	Cung cấp API lấy dữ liệu phương tiện để chỉ định cho nhiệm vụ	getVehicleList(): List<Vehicle>
AssignmentData	Cung cấp API truy vấn dữ liệu liên quan đến phân công công việc	insertTask(data: object): void updateTask(data: object): void deleteTask(ID: interger): void selectTask(): void
RouteData	Cung cấp API lấy dữ liệu về lộ trình và các MCP	createRoute(mcp_list: List<MCP>): void
EmployeeData	Cung cấp API lấy dữ liệu nhân viên để phân công cho nhiệm vụ	selectRoute(): void selectMCP(): void