

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



**CÔNG NGHỆ PHẦN MỀM (CO3001)**

---

**Assignment (Semester: 222, Duration: 12 weeks)**

# **Urban waste collection aid UWC 2.0**

---

Advisor: Phan Trung Hiếu  
Students: Ngô Huy Hoàng - 2013223 - L02  
Mai Lê Cường - 2012764 - L02  
Nguyễn Văn Công - 2012754 - L02  
Lê Trần Hoàng Vương - 2015107 - L02  
Trần Thế Phong - 2014130 - L02  
Lê Quang Tuấn Hào - 2013048 - L02  
Nguyễn Khôi Nguyên - 2013923 - L02

HO CHI MINH CITY, JANUARY 2023



## Contents

<b>1</b>	<b>Task 1: Requirement elicitation</b>	<b>4</b>
1.1	Task 1.1: Các bên liên quan của hệ thống và các vấn đề đặt ra: . . . . .	4
1.2	Task 1.2: Yêu cầu chức năng và phi chức năng . . . . .	4
1.2.1	Yêu cầu chức năng: . . . . .	4
1.2.2	Yêu cầu đối với hệ thống: . . . . .	5
1.2.3	Yêu cầu phi chức năng: . . . . .	5
1.2.4	Use case tổng quát toàn hệ thống: . . . . .	6
1.3	Task 1.3: Task assignment module . . . . .	7
1.3.1	Use case diagram: . . . . .	7
1.3.2	Mô tả chi tiết use-case diagram: . . . . .	8
<b>2</b>	<b>Task 2: System modeling</b>	<b>12</b>
2.1	Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module . . . . .	12
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 . . . . .	13
2.3	Draw a class diagram of Task Assignment module as comprehensive as possible . . . . .	15
2.3.1	Class Account: . . . . .	15
2.3.2	Class Employee: . . . . .	16
2.3.3	Class Back Officer: . . . . .	16
2.3.4	Class Janitor/Collector: . . . . .	16
2.3.5	Class Vehicle: . . . . .	17
2.3.6	Class Troller: . . . . .	17
2.3.7	Class MCP: . . . . .	17
2.3.8	Class Task . . . . .	17
2.3.9	Class Task Assignment View: . . . . .	18
2.3.10	Class Task Controller . . . . .	18
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 . . . . .	19
<b>3</b>	<b>Task 3: Architecture Design</b>	<b>22</b>
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. . . . .	22
3.1.1	Use a layered architecture to design the UWC 2.0 system. . . . .	22
3.1.2	Describe how will you present your User Interface. . . . .	24
3.1.3	Describe how will you store your data. . . . .	24
3.1.4	Describe how you will access to external services/ APIs. . . . .	25
3.2	Draw a component diagram for the Task Assignment module . . . . .	25
3.2.1	Component Diagram . . . . .	25
3.2.2	Mô tả . . . . .	26



<b>4</b>	<b>Task 4: Implementation - Sprint 1</b>	<b>27</b>
4.1	Setting up an online repository (github, bitbucket, etc) for version control. . . . .	27
4.2	Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files. . . . .	27
4.3	Conducted a usability test with the user interface you developed in MVP1. . . .	27



## Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Ngô Huy Hoàng	2013223	- Làm Task 1	14.3%
2	Trần Thế Phong	2014130	- Làm task 4	14.3%
3	Nguyễn Khôi Nguyên	2013923	- Làm task 2.2 và 2.3	14.3%
4	Lê Trần Hoàng Vương	2015107	-Làm task 5	14.3%
5	Lê Quang Tuấn Hào	2013048	- Làm task 2.1 và 2.4	14.3%
6	Nguyễn Văn Công	2012754	- Làm task 3	14.3%
7	Mai Lê Cường	2012764	- Làm task 3	14.3%

## 1 Task 1: Requirement elicitation

### 1.1 Task 1.1: Các bên liên quan của hệ thống và các vấn đề đặt ra:

Các bên liên quan mà cần chú ý đến trong hệ thống này gồm có:

- Back officers: những người điều khiển hệ thống trung tâm và đồng thời giao việc và điều khiển các Janitors và Collectors
- Collectors: những người làm nhiệm vụ lái phương tiện thu gom chất thải
- Janitors: những người làm nhiệm vụ thu gom rác thải tại các điểm tập kết rác (MCPs)

Các vấn đề mà các Back officers có thể đang gặp phải:

- Vấn đề trong việc theo dõi lịch trình làm việc của các Janitors và Collectors một cách trực quan và chi tiết đồng thời nắm được thông tin của các Janitors và Collectors
- Vấn đề trong việc liên lạc với các Janitors và Collectors
- Vấn đề trong việc nắm thông tin các MCPs, đặc biệt là các MCPs sắp đầy hoặc đã đầy

Các vấn đề mà các Janitor và Collectors có thể đang gặp phải:

- Vấn đề trong việc theo dõi lịch trình cá nhân trong hiện tại và các nhiệm vụ hằng ngày.
- Vấn đề trong việc liên lạc với các Janitors và Collectors khác
- Vấn đề trong việc nắm thông tin các MCPs, đặc biệt là các MCPs sắp đầy hoặc đã đầy

Các nhu cầu có thể có của các Back officers:

- Cần một hệ thống giúp nắm bắt và điều chỉnh lịch trình cho các Janitors và các Collectors
- Cần phương pháp liên lạc với các Janitors và Collectors một cách hiệu quả
- Cần nắm rõ thông tin các phương tiện phục vụ cho việc thu gom rác.
- Cần nắm rõ các thông tin của các MCPs

Các nhu cầu có thể có của các Janitors và Collectors:

- Cần hệ thống giúp theo dõi lịch trình làm việc hằng ngày, hằng tuần và các nhiệm vụ được giao
- Cần phương pháp liên lạc với các Janitors và Collectors khác một cách hiệu quả
- Cần nắm rõ các thông tin của các MCPs, đặc biệt và các MCPs sắp đầy hoặc đã đầy

### 1.2 Task 1.2: Yêu cầu chức năng và phi chức năng

#### 1.2.1 Yêu cầu chức năng:

Yêu cầu chức năng hệ thống đối với Back officers:

- Giúp họ có cái nhìn tổng quan về các Janitors và các collectors cũng như nắm được lịch trình làm việc của họ

- Giúp họ có cái nhìn tổng quan về các phương tiện và các thông số kỹ thuật của chúng (khối lượng, sức chứa, lượng nhiên liệu tiêu thụ,...)
- Cho Phép họ nắm được thông tin của các MCPs đặc biệt là về sức chứa của chúng.
- Cho phép họ giao nhiệm vụ cho các Janitors và collectors
- Cho Phép họ gán các Janitors và Collectors với các MCPs tùy ý
- Cho Phép họ tạo tuyến đường cho mỗi Collector và có thể có tối ưu tuyến đường để tiết kiệm nhiên liệu đồng thời rút ngắn đường đi
- Cho Phép họ liên lạc với Janitors và Collectors

Yêu cầu chức năng hệ thống đối với Janitor, Collectors:

- Giúp họ theo dõi, nắm rõ được lịch trình làm việc của mình
- Giúp các họ theo dõi, nắm được nhiệm vụ hằng tuần và hằng ngày của mình.
- Giúp các Janitors và Collectors liên lạc được với nhau và liên lạc với Back officers
- Cho phép Janitors và Collectors nhận và trả nhiệm vụ mỗi ngày
- Được thông báo nếu các MCPs đã đầy.

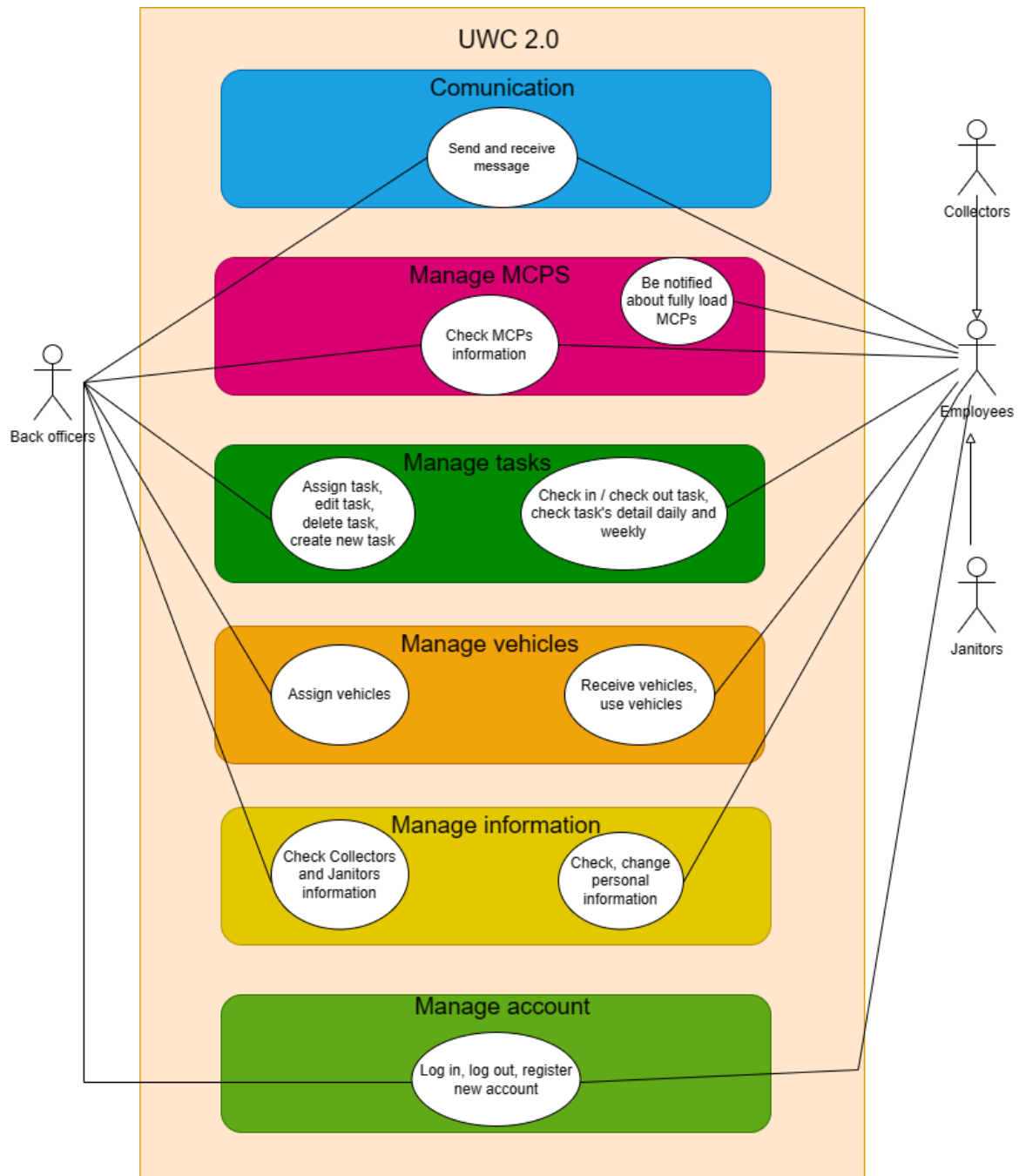
#### 1.2.2 Yêu cầu đối với hệ thống:

- Thông tin về sức chứa của các MCPs sẽ được cập nhật mỗi 15 phút 1 lần trong tối thiểu 95% thời gian hoạt động của chúng
- Các thông tin về nhiệm vụ của các Janitors và Collectors phải được hiển thị trên 1 trang mà không cần cuộn chuột
- Tin nhắn được gửi giữa các bên trong hệ thống phải được truyền đi trong thời gian thực với độ trễ thấp hơn 1 giây
- Sản phẩm có chức năng tra cứu, tìm kiếm các thông tin liên quan đến các Back Officer, Janitors, Collectors đồng thời cho phép các task
- Cho phép xem thông tin chi tiết của các MCPs trong thời gian thật

#### 1.2.3 Yêu cầu phi chức năng:

- Hệ thống có thể hoạt động liên tục 24/7
- Hệ thống có thể chạy được trên nhiều hệ điều hành khác nhau
- Hệ thống phải có giao diện thân thiện, dễ sử dụng kể cả với người dùng mới.
- Hệ thống dễ quản lý và bảo trì, nâng cấp, mở rộng nếu cần thiết
- Hệ thống có khả năng bảo mật tốt
- Truy suất dữ liệu nhanh và lưu trữ dữ liệu hiệu quả.

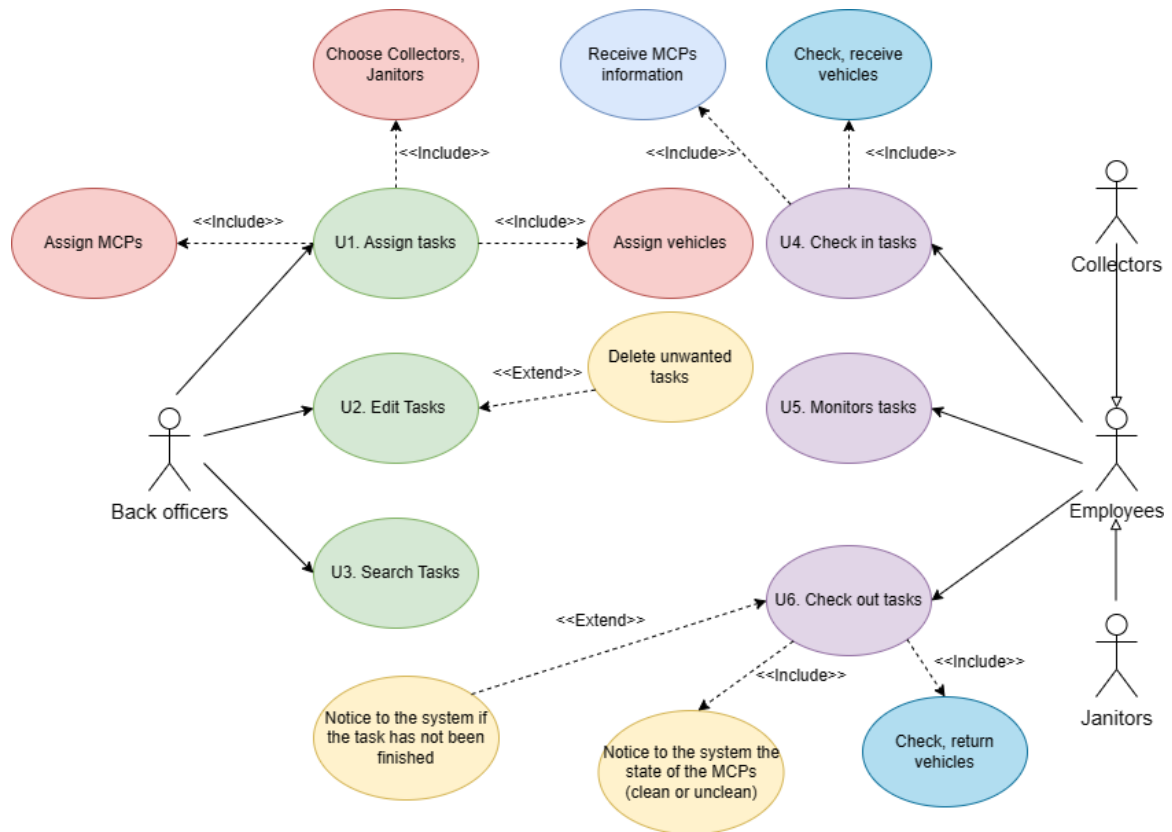
#### 1.2.4 Use case tổng quát toàn hệ thống:



Hình 1: General use-case diagram

### 1.3 Task 1.3: Task assignment module

#### 1.3.1 Use case diagram:



Hình 2: Task assignment module use-case diagram



### 1.3.2 Mô tả chi tiết use-case diagram:

<b>Use-case ID</b>	U1
<b>Use-case Name</b>	Assign task
<b>Use-case overview</b>	Dùng để giao task cho các Janitors và Collectors
<b>Actors</b>	Back officers
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. Internet đang khả dụng</li><li>2. Database có kết nối với MCPs, phương tiện, công cụ, danh sách Janitors và Collectors</li><li>3. Hệ thống đang hoạt động bình thường</li></ol>
<b>Trigger</b>	Người dùng click vào nút "Assign task"
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Đăng nhập vào hệ thống với tài khoản có sẵn</li><li>2. Nhấn vào nút "Tasks" để hiện lên bảng quản lý tất cả các tasks</li><li>3. Nhấn vào nút "Create new task" để tạo một task mới</li><li>4. Chọn Janitors và Collectors phù hợp với task cần giao</li><li>5. Chọn khu vực có MCPs phù hợp</li><li>6. Chọn phương tiện phù hợp cho Collector và Janitor bao gồm: xe chuyên chở và xe đẩy chuyên dụng phù hợp</li><li>7. Xác nhận với hệ thống về việc giao task bằng việc nhấn nút "Create"</li></ol>
<b>Post conditions</b>	<ol style="list-style-type: none"><li>1. Các Janitors và Collectors sẽ nhận được thông báo về task vừa được giao</li><li>2. Task mới được giao sẽ hiện trên lịch làm việc của các Janitors và Collectors có liên quan</li><li>3. Task mới được giao sẽ hiện trên bảng theo dõi task của Back officers</li></ol>
<b>Exception flow</b>	không

Hình 3: Mô tả chi tiết sơ đồ use-case của task assignment module

<b>Use-case ID</b>	U2
<b>Use-case Name</b>	Edit task
<b>Use-case overview</b>	Dùng để thay đổi thông tin của task
<b>Actors</b>	Back officers
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Internet đang khả dụng</li> <li>2. Database có kết nối với MCPs, phương tiện, công cụ, danh sách Janitors và Collectors</li> <li>3. Hệ thống đang hoạt động bình thường</li> </ol>
<b>Trigger</b>	Người dùng click vào nút “Edit task” bên cạnh mỗi task trong bảng theo dõi task của Back officers
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống với tài khoản có sẵn</li> <li>2. Nhấn vào nút “Tasks” để hiện lên bảng quản lý tất cả các tasks</li> <li>3. Chọn task muốn thực hiện việc thay đổi thông tin</li> <li>4. Kiểm tra thông tin task hiện tại</li> <li>5. Nhấn nút “Edit” để bắt đầu thực hiện thay đổi</li> <li>6. Thay đổi thông tin của 1 trong các trường tùy ý: Nhân viên nhận task, MCPs, phương tiện (xe đẩy, xe chuyên chở) bằng việc xóa thông tin cũ và thay bằng thông tin mới</li> <li>7. Nhấn nút “Confirm” để xác nhận việc thay đổi với hệ thống</li> </ol>
<b>Post conditions</b>	<ol style="list-style-type: none"> <li>1. Các thông tin của task sẽ được thay đổi sao cho phù hợp với yêu cầu của Back officer</li> <li>2. Các thông tin mới của task sẽ được lưu lại</li> </ol>
<b>Exception flow</b>	Các Back officers xóa task không mong muốn: <ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống với tài khoản có sẵn</li> <li>2. Nhấn vào nút “Tasks” để hiện lên bảng quản lý tất cả các tasks</li> <li>3. Chọn task không mong muốn</li> <li>4. Nhấn nút “Delete” để thực hiện xóa task đã chọn</li> </ol>

Hình 4: Mô tả chi tiết sơ đồ use-case của task assignment module

<b>Use-case ID</b>	U3
<b>Use-case Name</b>	Search task
<b>Use-case overview</b>	Dùng để tìm kiếm task trong bảng quản lý task của Back officers
<b>Actors</b>	Back officers
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Internet đang khả dụng</li> <li>2. Database có kết nối với MCPs, phương tiện, công cụ, danh sách Janitors và Collectors</li> <li>3. Hệ thống đang hoạt động bình thường</li> </ol>
<b>Trigger</b>	Người dùng điền tên task cần tìm vào thanh tìm kiếm trong bảng quản lý task của Back officer sau đó nhấn nút “Search”
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống với tài khoản có sẵn</li> <li>2. Nhấn vào nút “Tasks” để hiện lên bảng quản lý tất cả các tasks</li> <li>3. Gõ tên task muốn tìm vào thanh tìm kiếm</li> <li>4. Click nút “Search”</li> <li>5. Nhận các kết quả trả về</li> <li>6. Chọn task muốn tìm từ trong các kết quả trả về</li> </ol>
<b>Post conditions</b>	Các kết quả trả về hiện trên màn hình, các task không liên quan sẽ được bỏ bớt
<b>Exception flow</b>	Không

Hình 5: Mô tả chi tiết sơ đồ use-case của task assignment module

<b>Use-case ID</b>	U4
<b>Use-case Name</b>	Check in task
<b>Use-case overview</b>	Dùng để xác nhận việc nhận task của các Janitors và Collectors
<b>Actors</b>	Janitors và Collectors
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Internet đang khả dụng</li> <li>2. Database có kết nối với MCPs, phương tiện, công cụ</li> <li>3. Hệ thống đang hoạt động bình thường</li> </ol>
<b>Trigger</b>	Collectors nhấn vào nút “Receive task” bên cạnh bảng theo dõi task của cá nhân
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống bằng tài khoản có sẵn</li> <li>2. Nhấn nút “Schedule” để hiển thị task được giao trong ngày</li> <li>3. Nhận và kiểm tra thông tin của các MCPs được giao</li> <li>4. Nhận các phương tiện cần thiết: xe chuyên chở, xe đẩy.</li> <li>5. Kiểm tra thông tin về trạng thái hiện tại của các phương tiện, công cụ liệu có hư hỏng gì không.</li> <li>6. Nhấn nút “Accept” để xác nhận nhận task nếu tất cả các thông tin đều ổn</li> </ol>
<b>Post conditions</b>	Hệ thống xác nhận task đã được nhận và đang trong quá trình được thực hiện
<b>Exception flow</b>	<p>Có phương tiện, công cụ bị hư hỏng hoặc sai thông tin:</p> <ol style="list-style-type: none"> <li>1. Nhấn nút “Report” bên cạnh task được chọn để hiện form báo cáo vấn đề với Back officer</li> <li>2. Trong mục “Your name” điền họ và tên đầy đủ</li> <li>3. Trong mục “What is your problem” thì chọn “Vehicles and tools”</li> <li>4. Trong mục “Describe your problem” thì điền mô tả vấn đề mình gặp phải như: phương tiện đã có người sử dụng hoặc sai thông tin về phương tiện hoặc có dấu hiệu hư hỏng, không hoạt động được</li> <li>5. Xác nhận các thông tin vừa điền vào form</li> <li>6. Nhấn nút “Send” để gửi form tới hệ thống để các bên có thẩm quyền xử lý.</li> </ol>

Hình 6: Mô tả chi tiết sơ đồ use-case của task assignment module

<b>Use-case ID</b>	U5
<b>Use-case Name</b>	Monitor task
<b>Use-case overview</b>	Dùng để theo dõi các task được giao hằng ngày, hằng tuần
<b>Actors</b>	Janitors và Collectors
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Internet đang khả dụng</li> <li>2. Database có kết nối với MCPs, phương tiện, công cụ</li> <li>3. Hệ thống đang hoạt động bình thường</li> <li>4. Chức năng gửi và nhận tin nhắn đang hoạt động bình thường</li> </ol>
<b>Trigger</b>	Người dùng click vào nút “Schedule” để vào bảng danh sách task được giao cho bản thân hằng ngày, hằng tuần
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống với tài khoản có trước</li> <li>2. Click vào nút “Schedule” trong trang chính của hệ thống để vào mục lịch công việc của mình</li> <li>3. Kiểm tra task được giao hằng ngày, hằng tuần (nếu có)</li> <li>4. Kiểm tra thông tin của các task chưa hoàn thành và đã hoàn thành</li> <li>5. Thực hiện thông báo với Back officer nếu danh sách task đã đầy bằng việc sử dụng chức năng nhận và gửi tin nhắn</li> </ol>
<b>Post conditions</b>	Người dùng nắm được lịch làm việc của cá nhân hằng tuần và hằng ngày
<b>Exception flow</b>	không

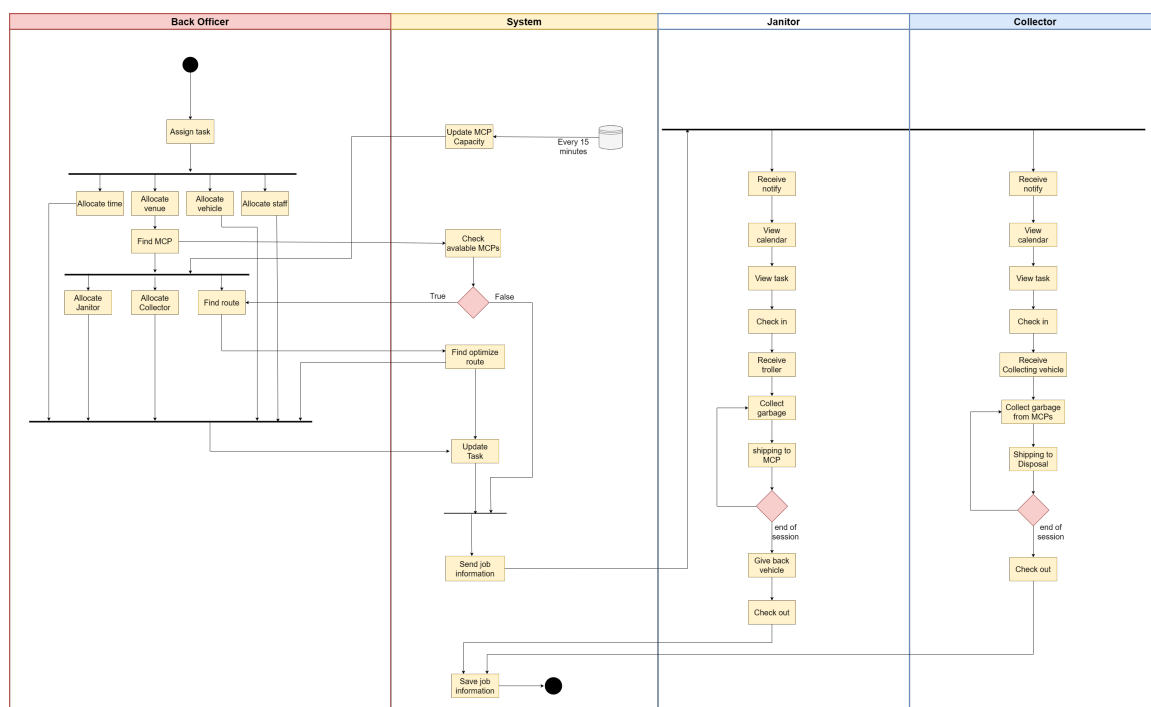
Hình 7: Mô tả chi tiết sơ đồ use-case của task assignment module

<b>Use-case ID</b>	U6
<b>Use-case Name</b>	Check out task
<b>Use-case overview</b>	Dùng để thực hiện tác vụ trả task khi đã kết thúc ca làm việc trong ngày
<b>Actors</b>	Janitors và Collectors
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Internet đang khả dụng</li> <li>2. Database có kết nối với MCPs, phương tiện, công cụ</li> <li>3. Hệ thống đang hoạt động bình thường</li> <li>4. Chức năng gửi và nhận tin nhắn đang hoạt động bình thường</li> </ol>
<b>Trigger</b>	Collectors nhấn vào nút “Check out task” bên cạnh bảng theo dõi task của cá nhân
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống bằng tài khoản có sẵn</li> <li>2. Nhấn nút “Schedule” để hiển thị task được giao trong ngày</li> <li>3. Tìm task vừa hoàn thành trong ngày</li> <li>4. Xác nhận trạng thái hiện tại của các MCPs</li> <li>5. Xác nhận đã trả phương tiện và công cụ lao động</li> <li>6. Kiểm tra nếu phương tiện lao động có hư hỏng hay không, nếu có phải thông báo với Back officer thông qua hệ thống tin nhắn</li> <li>7. Nhấn nút “Finished” để xác nhận với hệ thống là task đã hoàn thành.</li> </ol>
<b>Post conditions</b>	Hệ thống nhận được thông tin là task đã hoàn thành và sẽ gửi cho các bên có thẩm quyền để xử lý thông tin đó
<b>Exception flow</b>	<p>Task được giao khi đến lúc check out vẫn chưa được hoàn thành:</p> <ol style="list-style-type: none"> <li>1. Đăng nhập vào hệ thống bằng tài khoản có sẵn</li> <li>2. Nhấn nút “Schedule” để hiển thị task được giao trong ngày</li> <li>3. Tìm task được giao trong ngày nhưng vẫn chưa hoàn thành</li> <li>4. Ghi lại trạng thái hiện tại của các MCPs được giao</li> <li>5. Trả phương tiện và công cụ lao động</li> <li>6. Kiểm tra nếu phương tiện lao động có hư hỏng hay không, nếu có phải thông báo với Back officer bằng việc nhấn nút “Report” và thực hiện các bước như trong Exception flow của use-case có ID U4</li> <li>7. Nhấn nút “Unfinished” để xác nhận với hệ thống là task chưa hoàn thành.</li> </ol>

Hình 8: Mô tả chi tiết sơ đồ use-case của task assignment module

## 2 Task 2: System modeling

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



Hình 9: Task assignment module use-case diagram

#### Mô tả

Ta chọn start point là ở góc nhìn của Back Officer

- Ở bước Truy cập, người dùng sẽ tùy chọn các tính năng: Assign task và View Calendar.
- Tiếp theo, nếu người dùng chọn chức năng Assign task - giao nhiệm vụ, sẽ có 3 lựa chọn:
  - Create task - tạo task mới, chức năng này bao gồm gán yêu cầu thời gian (allocate time), gán địa điểm (venue), phương tiện (vehicle), nhân viên (staff).

Ở tính năng gán địa điểm (allocate venue), yêu cầu hệ thống tính toán, tìm được các MCPs cần thu gom, và từ đó tìm ra đường đi tối ưu để giảm thiểu chi phí. Nếu hệ thống tìm được available MCPs → tìm ra đường đi tối ưu. Nếu không có MCPs nào, thì hệ thống gửi thông báo về cho Back Officer và trở về trang chủ.

- Create task - sửa task (thay đổi người thực hiện, thay đổi vị trí của MCPs,...)
- Delete task - xóa task

Ở cả 3 tùy chọn trên, nếu thực hiện thành công, hệ thống sẽ tự động update task, calendar. Và gửi thông báo về cho những người liên quan (Back officer, collector và janitor). Sau đó, hệ thống sẽ đưa người dùng trở về trang chủ.

Đối với Janitor và Collector, sau khi được Assign task, họ sẽ nhận được thông báo và họ có

thể xem sau khi đã đăng nhập vào hệ thống. Từ đó, người dùng có các lựa chọn:

- Họ có thể xem task mà mình đã được phân công từ bảng thông báo.
- Thực hiện lựa chọn các tính năng khác như Check in, Check out
- Hoặc thoát khỏi chương trình.

→ Từ bảng thông báo, người dùng cũng có thể vào tính năng Check in, Check out để thực hiện Checkin hoặc Check out công việc đã được phân công.

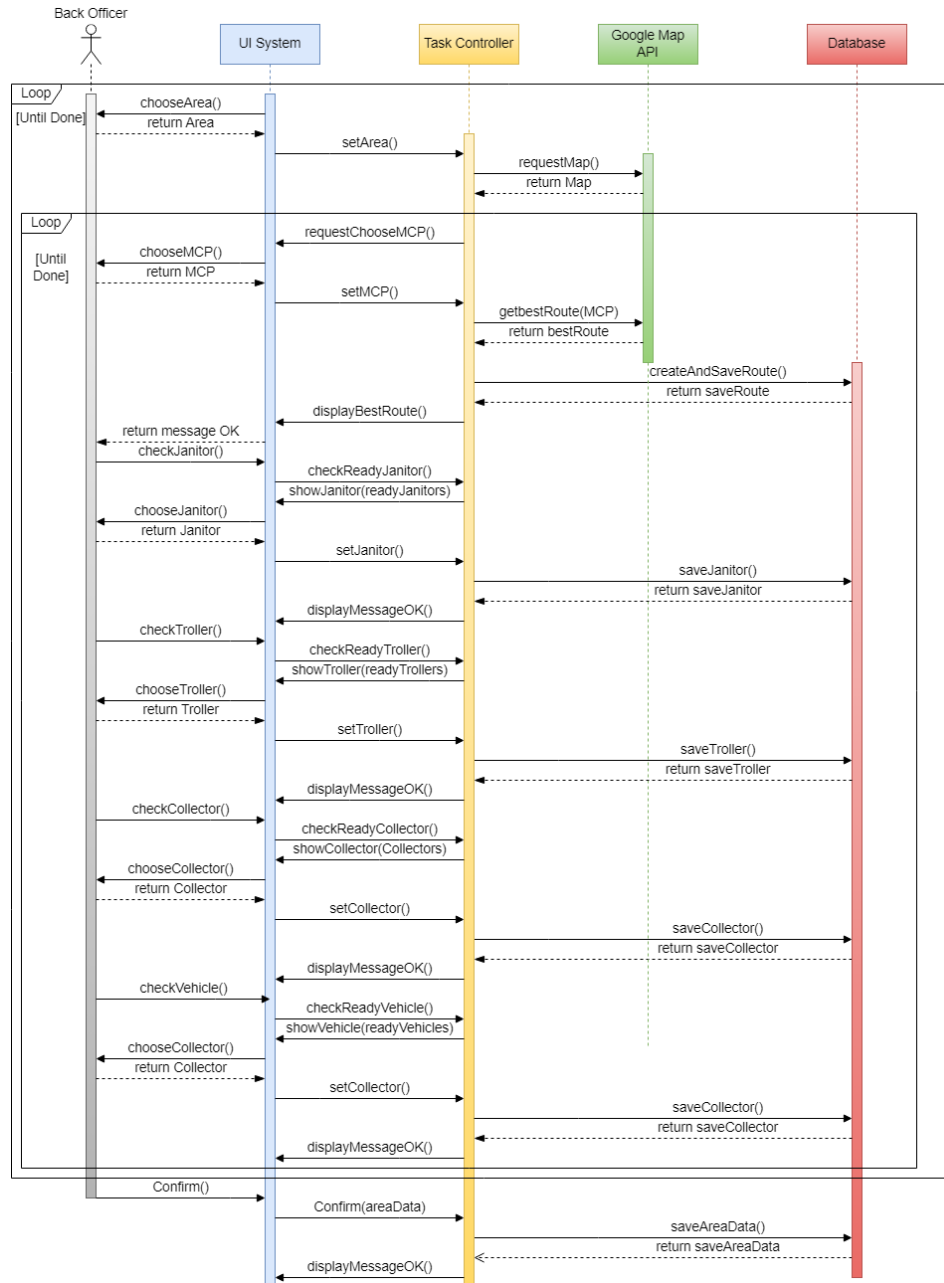
→ Nếu Check in, Check out thành công, hệ thống sẽ update tình trạng của task tự động → từ đó update calendar.

## 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

**Mô tả:**

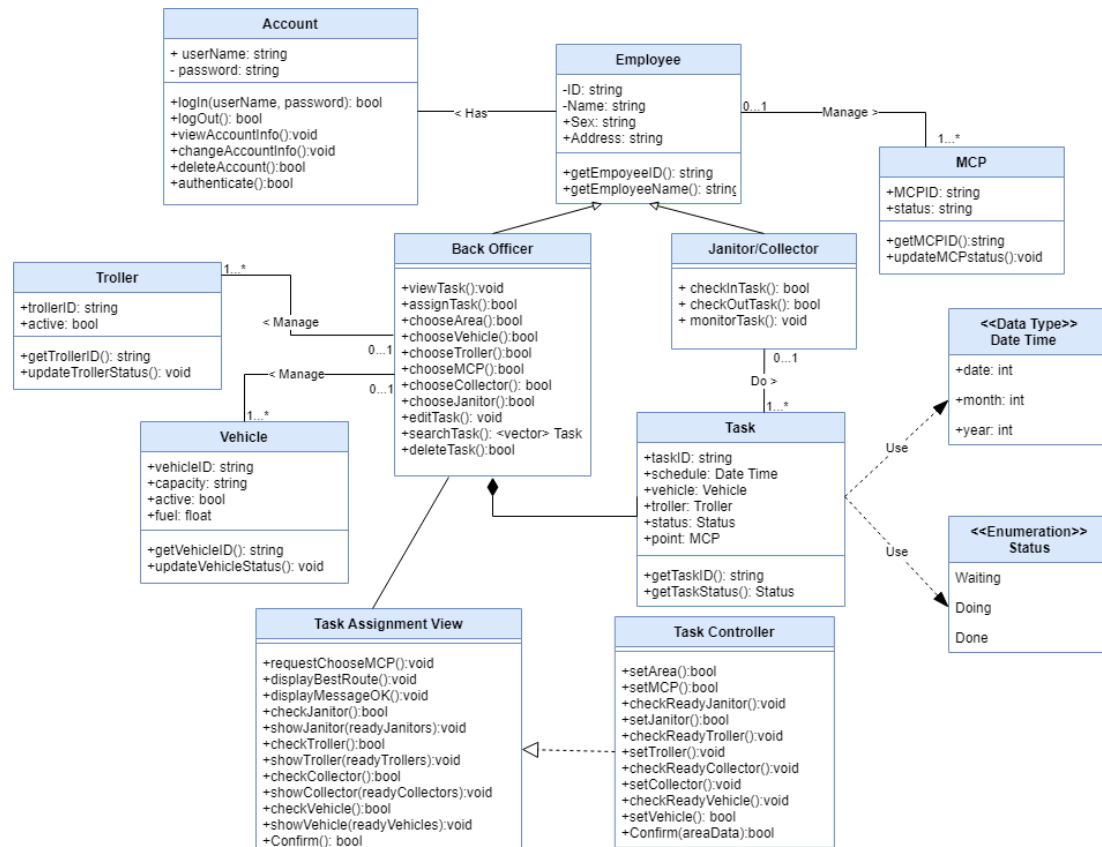
- Một khu vực sẽ bao gồm các thông tin: địa điểm khu vực công việc, các vị trí MCP trong khu vực, các tuyến đường (được tối ưu thông qua hệ thống thứ ba - Google Map API), danh sách nhân viên tham gia công việc (Janitor và Collector).
- Người dùng chọn khu vực xác định công việc thông qua bản đồ điện tử (Map) tạo bởi Google Map. Khi đó hệ thống khởi tạo một khu vực bao gồm các MCP và yêu cầu người dùng chọn **một** MCP (Major Collecting Point).
- Người dùng chọn điểm MCP thông qua Map. Hệ thống sẽ cập nhật thông tin MCP, sau đó đề xuất các đường đi tối ưu để đến MCP đã chọn.
- Tiếp theo người dùng kiểm tra các Janitor sẵn sàng cho công việc, chọn Janitor và chọn troller (trong trạng thái sẵn sàng) cho các Janitor đó.
- Sau đó người dùng kiểm tra các Collector sẵn sàng cho công việc, chọn Collector và chọn collecting vehicle (trong trạng thái sẵn sàng) cho Collector đó.
- Việc chọn tuyến đường, nhân viên và gán phương tiện cho (nhóm) nhân viên được lặp lại đến khi hoàn tất.
- Sau cùng, người dùng thông qua giao diện confirm, bấm nút Confirm. Thông tin chi tiết của area sẽ lưu vào database ở phía server và gửi các message tương ứng.

**Lưu ý:** Nhằm tối ưu việc đặt tên hàm cũng như tên biến (tránh việc đặt tên quá dài), nhóm em xin được viết tắt collecting vehicle thành vehicle.



Hình 10: Sequence diagram for back officer to assign vehicle(trolley + collecting vehicle) to janitor & collector  
Ref: [https://drive.google.com/file/d/1UKONlUyhYn81nzqkqWzgNcPCvdyb1cXMoI/view?usp=share\\_link](https://drive.google.com/file/d/1UKONlUyhYn81nzqkqWzgNcPCvdyb1cXMoI/view?usp=share_link)

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



Hình 11: Class diagram of Task Assignment module

Ref: [https://drive.google.com/file/d/1i7d9emLC44anud6jZdELoYPslQ6dPwWg/view?usp=share\\_link](https://drive.google.com/file/d/1i7d9emLC44anud6jZdELoYPslQ6dPwWg/view?usp=share_link)

### 2.3.1 Class Account:

#### 1. Các thuộc tính của lớp Account:

- **userName**: tên đăng nhập tài khoản - kiểu dữ liệu: string.
- **password**: mật khẩu đăng nhập - kiểu dữ liệu: string.

#### 2. Các phương thức của lớp Account:

- **login**: đăng nhập vào tài khoản, gồm 2 đối số là userName và password.
- **logout**: đăng xuất khỏi tài khoản hiện tại.
- **viewAccountInfo**: xem thông tin của tài khoản.
- **changeAccountInfo**: thay đổi thông tin tài khoản.



- **deleteAccount**: xóa tài khoản.
- **authenticate**: xác thực tài khoản.

### 2.3.2 Class Employee:

#### 1. Các thuộc tính của lớp Employee:

- **ID**: mã số nhân viên - kiểu dữ liệu: string.
- **Name**: tên nhân viên - kiểu dữ liệu: string.
- **sex**: giới tính của nhân viên - kiểu dữ liệu: string.
- **address**: địa chỉ của nhân viên - kiểu dữ liệu: string.

#### 2. Các phương thức của lớp Employee:

- **getEmployeeName**: dùng để lấy tên của nhân viên.
- **getEmployeeID**: dùng để lấy mã số của nhân viên.

### 2.3.3 Class Back Officer:

*Các phương thức của lớp Back Officer:*

- **viewTask**: xem tình trạng các Task hiện tại.
- **assignTask**: dùng để phân công nhiệm vụ.
- **chooseArea**: chọn khu vực chứa các MCP cần thu gom.
- **chooseVehicle**: chọn phương tiện cho Task.
- **chooseMCP**: chọn MCP cho Task.
- **chooseJanitor**: dùng để chỉ định nhiệm vụ cho các Janitors.
- **chooseCollector**: dùng để chỉ định nhiệm vụ cho các Collector.
- **chooseTroller**: chọn Troller cho Task.
- **editTask**: dùng để chỉnh sửa task đã phân công.
- **searchTask**: dùng để tìm kiếm (các) task.
- **deleteTask**: dùng để xóa 1 hoặc nhiều task đã phân công.

### 2.3.4 Class Janitor/Collector:

*Các phương thức của lớp Janitor/Collector:*

- **checkInTask**: để Janitor và Collector nhận việc của mình.
- **checkOutTask**: để Janitor và Collect xác nhận hoàn tất task.
- **monitorTask**: để Janitor và Collector theo dõi các task được phân công.

### 2.3.5 Class Vehicle:

**Lưu ý:** Nhằm tối ưu việc đặt tên hàm cũng như tên biến (tránh việc đặt tên quá dài), nhóm em xin được viết tắt collecting vehicle thành vehicle.

#### 1. Các thuộc tính của lớp Vehicle:

- **vehicleID:** mã số phương tiện - kiểu dữ liệu: string.
- **capacity:** sức chứa còn lại của phương tiện - kiểu dữ liệu: string.
- **active:** trạng thái hoạt động hiện tại - kiểu dữ liệu: string.
- **fuel:** nhiên liệu còn lại của phương tiện - kiểu dữ liệu: float.

#### 2. Phương thức của lớp Vehicle:

- **updateVehicleStatus:** dùng để cập nhật trạng thái hiện tại của phương tiện.
- **getVehicleID:** dùng để lấy mã số của phương tiện.

### 2.3.6 Class Troller:

#### 1. Các thuộc tính của lớp Troller:

- **trollerID:** mã số troller - kiểu dữ liệu: string.
- **active:** trạng thái hoạt động hiện tại - kiểu dữ liệu: string.

#### 2. Phương thức của lớp Troller:

- **updateTrollerStatus:** dùng để cập nhật trạng thái hiện tại của troller.
- **getTrollerID:** dùng để lấy mã số của troller.

### 2.3.7 Class MCP:

#### 1. Các thuộc tính của lớp MCP:

- **MCPID:** mã số của MCP - kiểu dữ liệu: string.
- **status:** sức chứa còn lại của MCP - kiểu dữ liệu: string.

#### 2. Phương thức của lớp MCP:

- **updateMCPStatus:** dùng để cập nhật trạng thái hiện tại của MCP.
- **getMCPID:** dùng để lấy mã số của MCP.

### 2.3.8 Class Task

#### 1. Các thuộc tính của lớp Task:

- **taskID:** mã số của task được phân công - kiểu dữ liệu: string.
- **schedule:** lịch làm việc của task - kiểu dữ liệu: Date Time.
- **vehicle:** phương tiện thực hiện task - kiểu dữ liệu: Vehicle.
- **troller:** troller thực hiện task - kiểu dữ liệu: Troller.
- **status:** trạng thái hiện tại của task - kiểu dữ liệu: Status.

- **point**: MCP cần thu gom của task - kiểu dữ liệu: MCP.

## 2. Các phương thức của lớp Task:

- **getTaskID**: dùng để lấy mã số task.
- **getTaskStatus**: dùng để lấy trạng thái của task.

### 2.3.9 Class Task Assignment View:

#### Các phương thức của lớp Task Assignment View

- **requestChooseMCP**: yêu cầu người dùng chọn 1 MCP để thu gom rác.
- **displayBestRoute**: dùng để hiển thị đường đi tốt nhất.
- **displayMessageOK**: dùng để hiển thị thông báo OK.
- **checkJanitor**: yêu cầu hệ thống (Task Controller) kiểm tra các Janitor.
- **showJanitor**: dùng để hiển thị các Janitor đã sẵn sàng cho việc thu gom rác, gồm 1 đối số là readyJanitors.
- **checkTroller**: yêu cầu hệ thống (Task Controller) kiểm tra các Troller.
- **showTroller**: dùng để hiển thị các troller đã sẵn sàng cho việc thu gom rác, gồm 1 đối số là readyTrollers.
- **checkCollector**: yêu cầu hệ thống (Task Controller) kiểm tra các Collector.
- **showCollectors**: dùng để hiển thị các Collector đã sẵn sàng cho việc thu gom rác, gồm 1 đối số là readyCollectors.
- **checkVehicle**: yêu cầu hệ thống (Task Controller) kiểm tra phương tiện.
- **showVehicle**: dùng để hiển thị các phương tiện đã sẵn sàng cho việc thu gom rác, gồm 1 đối số là readyVehicles.
- **Confirm**: người dùng xác nhận việc thiết lập.

### 2.3.10 Class Task Controller

#### Các phương thức lớp Task Controller:

- **setArea**: dùng để thiết lập thông tin khu vực.
- **setMCP**: dùng để thiết lập thông tin MCP.
- **checkReadyJanitor**: dùng để kiểm tra các Janitor nào sẵn sàng cho nhiệm vụ.
- **setJanitor**: dùng để thiết lập thông tin Janitor cho task.
- **checkReadyTroller**: dùng để kiểm tra các troller nào sẵn sàng cho nhiệm vụ.
- **setTroller**: dùng để thiết lập thông tin troller cho task.
- **checkReadyCollector**: dùng để kiểm tra các Collector nào sẵn sàng cho nhiệm vụ.

- **setCollector**: dùng để thiết lập thông tin Collector cho task.
- **checkReadyVehicle**: dùng để kiểm tra các phương tiện nào sẵn sàng cho nhiệm vụ.
- **setVehicle**: dùng để thiết lập thông tin phương tiện cho task.
- **Confirm**: dùng để xác nhận thông tin khu vực, gồm một đối số là `areaData`.

**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**



Tài khoảnDịch vụHệ thống

Bạn cần tìm gì ?

Danh mục

Home

Work calendar

Chat

CSKH

Trang chủ >> Đăng nhập

### ĐĂNG NHẬP TÀI KHOẢN

Tên tài khoản

Mật khẩu

Đăng nhập

Bạn quên mật khẩu? Nhập địa chỉ email để lấy lại mật khẩu qua email.

Email

Lấy lại mật khẩu

**CSKH**  
Câu hỏi thường gặp  
Hướng dẫn sử dụng

**Về chúng tôi**  
Đội phát triển  
Liên hệ hỗ trợ

**Follow us**  
Facebook  
Instagram

Địa chỉ: Tầng 6, Trường Đại học Bách Khoa (Cơ sở 2), Phường Đồng Hoà, Thành phố Dĩ An, Tỉnh Bình Dương  
Tổng đài hỗ trợ: 1900xxxx - Email: sang.nguyenvinh@hcmut.edu.vn

©2022 - Bản quyền thuộc về Công ty Bình Minh

Hình 12: Sign in

Tài khoảnDịch vụHệ thống

Sản phẩm cần tìm

Danh mục

Home

Work calendar

Chat

CSKH

Trang chủ >> Đăng kí

### ĐĂNG KÍ TÀI KHOẢN

Họ

Tên

Tên tài khoản

Email đăng kí

Mật khẩu

Xác nhận mật khẩu

Đăng kí

**CSKH**  
Câu hỏi thường gặp  
Hướng dẫn sử dụng

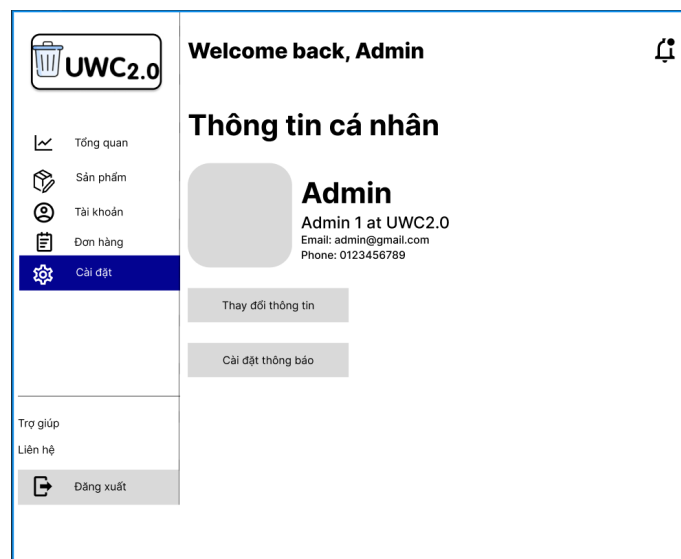
**Về chúng tôi**  
Đội phát triển  
Liên hệ hỗ trợ

**Follow us**  
Facebook  
Instagram

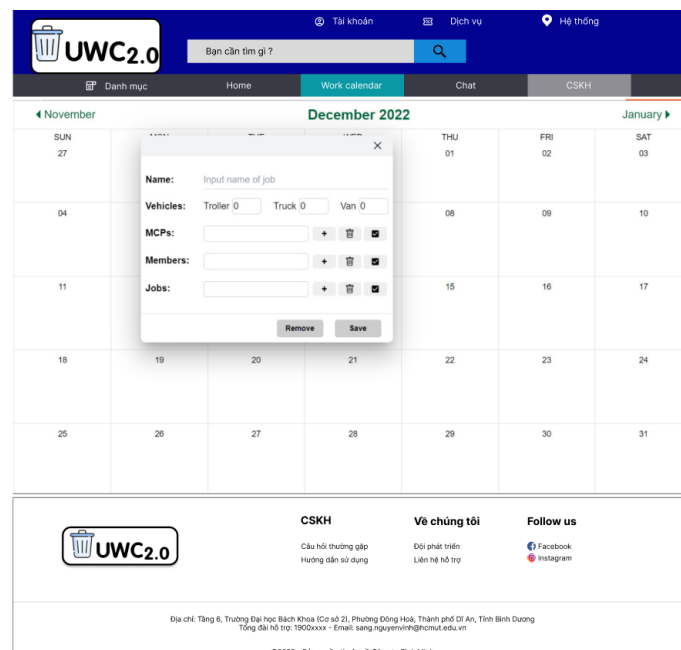
Địa chỉ: Tầng 6, Trường Đại học Bách Khoa (Cơ sở 2), Phường Đồng Hoà, Thành phố Dĩ An, Tỉnh Bình Dương  
Tổng đài hỗ trợ: 1900xxxx - Email: sang.nguyenvinh@hcmut.edu.vn

©2023 - Bản quyền thuộc về Công ty Bình Minh

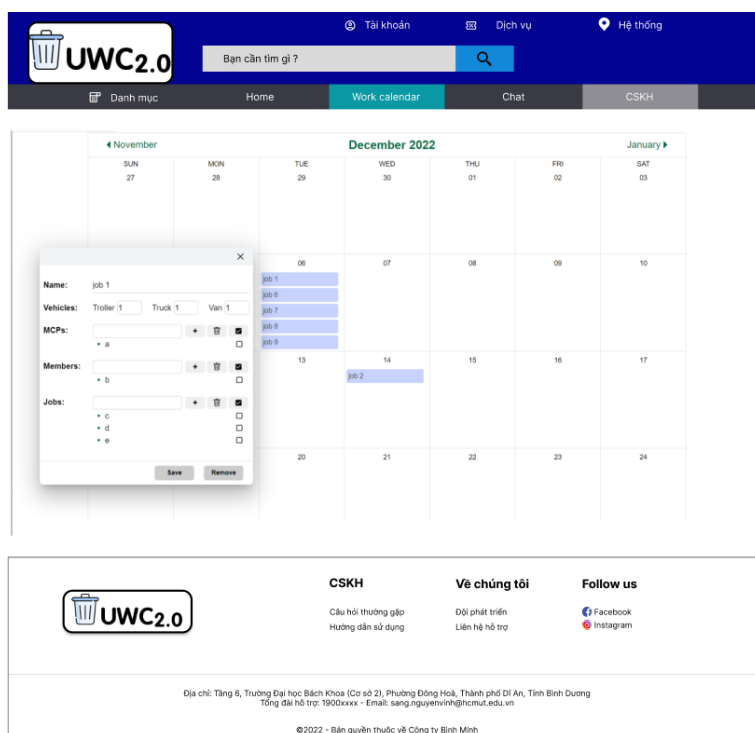
Hình 13: Sign up



Hình 14: Admin view Profile



Hình 15: Choose day and assign task



Hình 16: Choose day and assign task

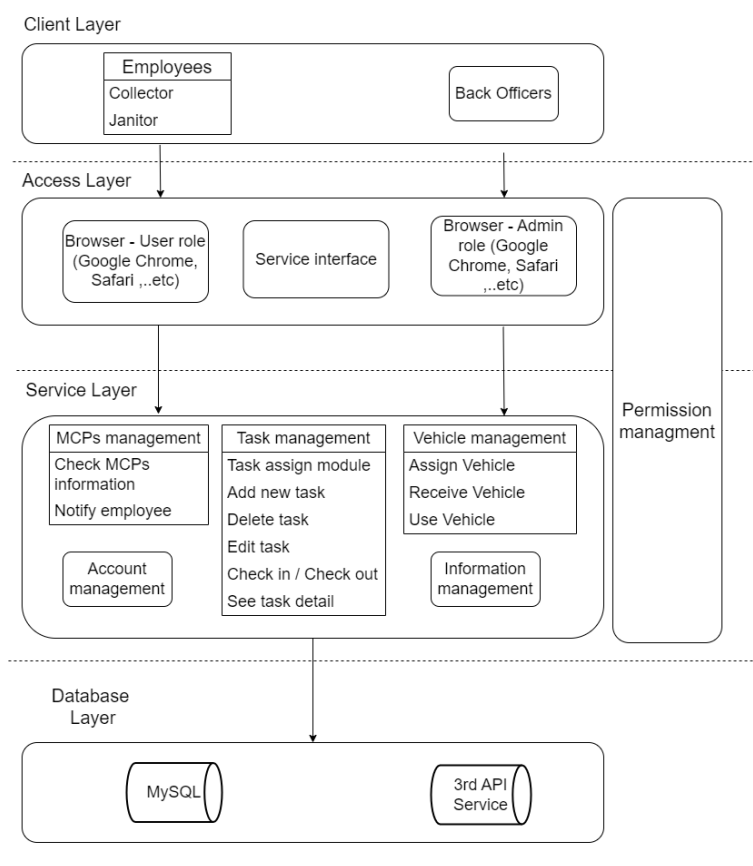
### 3 Task 3: Architecture Design

**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.**

**3.1.1 Use a layered architecture to design the UWC 2.0 system.**

Layered Architecture là một cách tổ chức hệ thống thành các lớp với chức năng liên quan được liên kết với mỗi lớp. Thông thường gồm 3 lớp chính bao gồm: Lớp giao diện người dùng (Presentation Layer), Lớp logic nghiệp vụ (Business-Logic Layer) và Lớp dữ liệu (Database Layer). Một lớp cung cấp các dịch vụ cho lớp trên nó, vì vậy các lớp ở mức thấp nhất đại diện cho các dịch vụ cốt lõi có khả năng được sử dụng trên toàn hệ thống. Dưới đây là phần hiện thực của kiến trúc lớp của hệ thống.

Hệ thống được hiện thực sử dụng 4 lớp bao gồm Client Layer, Access Layer, Services Layer và Database Layer. Mỗi lớp chỉ giao tiếp với các lớp liền kề của nó và không biết đến các lớp còn lại. Sự tách biệt các mối quan tâm này cho phép tính linh hoạt và khả năng mở rộng cao hơn trong thiết kế hệ thống cũng như đảm bảo thiết kế mô-đun, khả năng bảo trì và tách biệt các mối quan tâm trong quá trình phát triển phần mềm. Nó cũng giúp dễ dàng xác định và cách ly các sự cố, kiểm tra các thành phần một cách độc lập và thay thế hoặc nâng cấp các bộ phận của



Hình 17: Layered Architecture Design



hệ thống mà không ảnh hưởng đến toàn bộ ứng dụng.

### 3.1.2 Describe how will you present your User Interface.

Khi sử dụng kiến trúc phân lớp, lớp giao diện người dùng chịu trách nhiệm trình bày dữ liệu và chức năng của ứng dụng cho người dùng cuối. Lớp giao diện người dùng thường bao gồm các thành phần giao diện người dùng như biểu mẫu, nút, menu và hộp thoại, được sử dụng để tương tác với người dùng. Các thành phần này thường được tạo bằng HTML, CSS, JavaScript.

Trong kiến trúc phân lớp, lớp giao diện người dùng giao tiếp với lớp ứng dụng để truy xuất và cập nhật dữ liệu. Lớp ứng dụng chứa chức năng cốt lõi của ứng dụng và chịu trách nhiệm xử lý các yêu cầu từ lớp giao diện người dùng.

Để đảm bảo tách biệt rõ ràng, quan trọng là giữ cho lớp giao diện người dùng càng tối ưu càng tốt. Điều này có nghĩa là hầu hết logic nghiệp vụ nên được triển khai trong lớp ứng dụng, với lớp giao diện người dùng chỉ cung cấp giao diện thân thiện với người dùng để người dùng tương tác.

Tóm lại, lớp giao diện người dùng được thiết kế có tính đến người dùng, cung cấp giao diện rõ ràng và trực quan giúp người dùng dễ dàng thực hiện các tác vụ họ cần hoàn thành.

Có hai thành phần chính sau đây với những tác vụ cụ thể :

- UI Components : gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng. . .
- UI Process Components : là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI
  - Hiển thị màn hình Sign in
  - Hiển thị màn hình Sign up
  - Hiển thị màn hình User information
  - Hiển thị màn hình Assign Vehicle
  - Hiển thị màn hình Assign Task

### 3.1.3 Describe how will you store your data.

Trong kiến trúc phân lớp, việc lưu trữ và truy xuất dữ liệu thường được xử lý bởi một lớp riêng biệt. Lớp này chịu trách nhiệm trừu tượng hóa các chi tiết của cơ chế lưu trữ dữ liệu cơ bản (chẳng hạn như cơ sở dữ liệu) và cung cấp giao diện chung cho lớp logic nghiệp vụ để truy cập và thao tác dữ liệu.

Lớp logic nghiệp vụ giao tiếp với dữ liệu để truy xuất và thao tác dữ liệu. Lớp logic nghiệp vụ cũng có thể thực hiện xác thực và quy tắc nghiệp vụ trên dữ liệu trước khi lưu trữ nó trong cơ chế lưu trữ dữ liệu.

Trong kiến trúc phân lớp, quan trọng là phải tách biệt và tách rời các lớp khác nhau. Điều này có nghĩa là lớp logic nghiệp vụ không được có bất kỳ kiến thức nào về cơ chế lưu trữ dữ liệu cơ bản và chỉ nên tương tác với lớp dữ liệu thông qua giao diện được xác định rõ.

Cơ chế lưu trữ dữ liệu nên được chọn dựa trên các yêu cầu cụ thể của ứng dụng, có tính đến các yếu tố như khả năng mở rộng, hiệu suất và tính toàn vẹn của dữ liệu. Lớp dữ liệu phải được thiết kế để trừu tượng hóa các chi tiết của cơ chế lưu trữ dữ liệu và cung cấp giao diện đơn giản và nhất quán để lớp logic nghiệp vụ tương tác một cách dễ dàng.

Khi nói đến việc lưu trữ dữ liệu trong một ứng dụng, trách nhiệm thuộc về lớp truy cập dữ liệu. Dưới đây là các bước lưu trữ dữ liệu trong ứng dụng web sử dụng kiến trúc phân lớp:

- Xác định lớp truy cập dữ liệu: Lớp truy cập dữ liệu chịu trách nhiệm tương tác với cơ sở dữ liệu và thực hiện các thao tác CRUD (Create, Read, Update, Delete). Định nghĩa các lớp, giao diện và phương thức cần thiết để kết nối với cơ sở dữ liệu và thao tác dữ liệu.
- Thiết lập kết nối đến cơ sở dữ liệu: Sử dụng trình kết nối cơ sở dữ liệu, ở đây ứng dụng sẽ sử dụng hệ cơ sở dữ liệu MySQL.
- Tạo mô hình dữ liệu: Xác định mô hình dữ liệu biểu thị cấu trúc của dữ liệu sẽ được lưu trữ trong cơ sở dữ liệu.
- Triển khai các phương thức truy cập dữ liệu: Triển khai các phương thức trong lớp truy cập dữ liệu cho phép thực hiện các thao tác CRUD trên các mô hình dữ liệu. Các phương pháp này sử dụng trình kết nối cơ sở dữ liệu để tương tác với cơ sở dữ liệu.
- Sử dụng lớp truy cập dữ liệu từ lớp logic nghiệp vụ: Lớp logic nghiệp vụ nên sử dụng lớp truy cập dữ liệu để tương tác với cơ sở dữ liệu. Khi lớp logic nghiệp vụ cần lưu trữ hoặc truy xuất dữ liệu, nó sẽ gọi phương thức thích hợp trong lớp truy cập dữ liệu.
- Sử dụng lớp logic nghiệp vụ từ lớp giao diện người dùng: Lớp giao diện người dùng nên sử dụng lớp logic nghiệp vụ để truy xuất hoặc thao tác dữ liệu. Khi người dùng tương tác với trang web, lớp trình bày sẽ gọi phương thức thích hợp trong lớp logic nghiệp vụ để truy xuất hoặc thao tác dữ liệu.

#### 3.1.4 Describe how you will access to external services/ APIs.

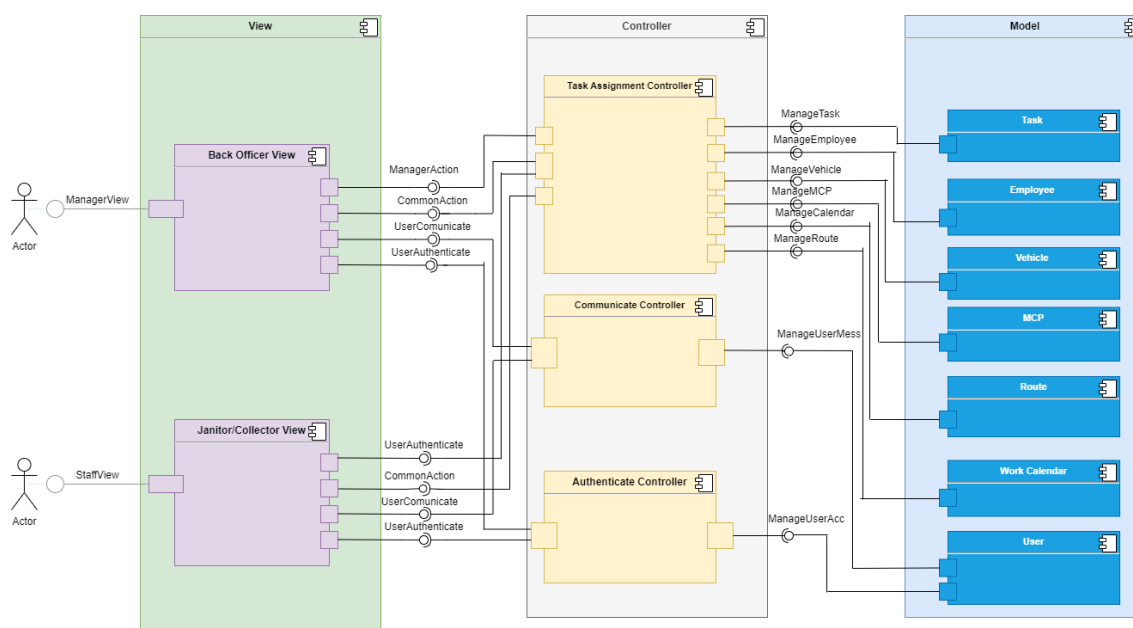
Trong kiến trúc phân lớp, việc truy cập các dịch vụ và API bên ngoài thường được thực hiện thông qua một lớp dịch vụ. Lớp dịch vụ chịu trách nhiệm trừu tượng hóa các chi tiết của dịch vụ bên ngoài hoặc API và cung cấp giao diện chung để lớp logic nghiệp vụ tương tác. Dịch vụ API có thể được triển khai bằng nhiều kỹ thuật và công nghệ khác nhau, tùy thuộc vào các yêu cầu cụ thể của ứng dụng. Một số cách tiếp cận phổ biến bao gồm:

- API RESTful: Nhiều dịch vụ và API bên ngoài hiển thị API RESTful có thể được truy cập bằng các yêu cầu HTTP. Lớp tích hợp có thể được triển khai bằng thư viện máy khách REST như RestSharp hoặc Refit.
- API SOAP: Một số dịch vụ và API bên ngoài hiển thị API SOAP có thể được truy cập bằng các thông báo SOAP dựa trên XML. Lớp tích hợp có thể được triển khai bằng thư viện máy khách SOAP như WCF hoặc Apache CXF.
- Messaging: Một số dịch vụ và API bên ngoài sử dụng các giao thức nhắn tin như AMQP hoặc MQTT để trao đổi dữ liệu.

### 3.2 Draw a component diagram for the Task Assignment module

#### 3.2.1 Component Diagram

Link for the diagram : <https://drive.google.com/file/d/1705VFWRB5KXmT058kUFdu4sSxSEChzHa/view>



Hình 18: Sơ đồ Component Diagram

### 3.2.2 Mô tả

Giao diện của Back Officer được khối Back Officer View cung cấp. Ở giao diện này, Back Officer có thể thực các chức năng liên quan đến quản lý và gán công việc, nhắn tin với các nhân viên, các chức năng liên quan đến tài khoản người dùng. Các chức năng này nằm trong provided interface ManagerView. Cụ thể:

- Khi Back Officer yêu cầu các chức năng liên quan đến quản lý và gán công việc thì khối Back Officer view sẽ gửi yêu cầu đến khối Task Assignment Controller thông qua 2 required interface ManagerAction và CommonAction. Task Assignment Controller sẽ xử lý các yêu cầu liên quan đến task bao gồm: tạo task, tạo tuyến đường, tối ưu hóa tuyến đường, tạo MCP, điều chỉnh MCP, xem thông tin về work calendar, MCP, tuyến đường.
- Sau khi nhận được yêu cầu Task Assignment Controller sẽ gọi các khối Task, Employee, Vehicle, MCP, Work calendar để xử lý các vấn đề liên quan đến data của task, nhân viên, phương tiện, MCP, work calendar thông qua các required interface ManageTask, ManageEmployee, ManageVehicle, ManageMCP, ManageRoute, ManageCalendar tương ứng.
- Khi Back Officer yêu cầu các chức năng liên quan đến nhắn tin thì khối Back Officer View sẽ gửi yêu cầu đến khối Communicate Controller thông qua required interface UserCommunicate. Communicate controller sẽ xử lý các yêu cầu liên quan đến tin nhắn bao gồm: gửi tin nhắn, xem tin nhắn và tìm kiếm.
- Sau khi nhận được yêu cầu Communicate controller sẽ gọi khối User thông qua required interface ManageUserMess để xử lý các vấn đề liên quan đến data tin nhắn người dùng như: lấy dữ liệu tin nhắn và cập nhật dữ liệu tin nhắn.

- Khi back officer yêu cầu các chức năng đăng nhập, đăng xuất, quản lý tài khoản thì khối Back officer view sẽ gửi yêu cầu đến khối Authenticate controller thông qua required interface UserAuthenticate. Authenticate controller sẽ xử lý các yêu cầu liên quan đến tài khoản như chỉnh sửa tài khoản, tạo tài khoản, login, logout.
- Sau khi nhận được yêu cầu Authenticate controller sẽ gọi khối User thông qua required interface ManageUserAcc để xử lý các vấn đề liên quan đến data tài khoản người dùng như: lấy tên đăng nhập, lấy mật khẩu, lấy lịch sử hoạt động, cập nhật tên đăng nhập, cập nhật mật khẩu.

Giao diện của janitor/collector được khối Janitor/Collector view cung cấp. Ở giao diện này janitor/collector có thể thực các chức năng liên quan đến quản lý công việc, nhắn tin với các nhân viên, các chức năng liên quan đến tài khoản người dùng. Các chức năng này nằm trong provided interface StaffView. Cụ thể:

- Khi Back Officer yêu cầu các chức năng liên quan đến quản lý và gán công việc thì khối Back officer view sẽ gửi yêu cầu đến khối Task assignment controller thông qua 2 required interface ManangerAction và CommonAction. Task assignment controller sẽ xử lý các yêu cầu liên quan đến task bao gồm: tạo task, tạo tuyến đường, tối ưu hóa tuyến đường, tạo MCP, điều chỉnh MCP, xem thông tin về work calendar, MCP, tuyến đường.
- Sau khi nhận được yêu cầu Task assignment controller sẽ gọi các khối Task, Employee, Vehicle, MCP, Work calendar để xử lý các vấn đề liên quan đến data của task, nhân viên, phương tiện, MCP, work calendar thông qua các required interface ManageTask, ManageEmployee, ManageVehicle, ManageMCP, ManageRoute, ManageCalendar tương ứng.
- Khi janitor/collector yêu cầu các chức năng liên quan đến nhắn tin thì hệ thống sẽ thực hiện các bước giống với Back Officer đã trình bày ở trên.
- Khi janitor/collector yêu cầu các chức năng đăng nhập, đăng xuất, quản lý tài khoản thì hệ thống sẽ thực hiện các bước giống với Back Officer đã trình bày ở trên.

## 4 Task 4: Implementation - Sprint 1

### 4.1 Setting up an online repository (github, bitbucket, etc) for version control.

Link github: <https://github.com/ThePhong1803/UWC2.0>

### 4.2 Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files.

### 4.3 Conducted a usability test with the user interface you developed in MVP1.