

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



CÔNG NGHỆ PHẦN MỀM (CO3001)

---

Báo cáo Bài tập lớn

# HỆ THỐNG HỖ TRỢ THU GOM RÁC THẢI ĐÔ THỊ - UWC 2.0

---

Lớp: **L01**

Giảng viên hướng dẫn: **Lê Đình Thuận**

Thành viên nhóm	Mã số sinh viên
<b>Nguyễn Hữu Danh</b>	<b>2010174</b>
<b>Phạm Khai Anh Duy</b>	<b>2011015</b>
<b>Nguyễn Huỳnh Tuấn Hưng</b>	<b>2011329</b>
<b>Lê Minh Nghĩa</b>	<b>2010445</b>
<b>Trần Thiện Nhân</b>	<b>2010481</b>
<b>Bùi Khánh Vĩnh</b>	<b>2010091</b>
<b>Trần Nguyên Vũ</b>	<b>2012445</b>

Thành phố Hồ Chí Minh, Tháng 11 năm 2022

## Mục lục

<b>1 Task 1: Requirement elicitation</b>	<b>3</b>
1.1 Identify the context of this project. Who are relevant stakeholders? What are their current needs? What could be their current problem? In your opinion, what benefits UWC 2.0 will be for each stakeholder? . . . . .	3
1.2 Describe all functional and non-functional requirements that can be inferred from the project description. Draw a use-case diagram for the whole system . . . . .	6
1.2.1 Functional requirements . . . . .	6
1.2.2 Non-functional requirements . . . . .	7
1.2.3 Use-case diagram cho hệ thống . . . . .	8
1.3 For the Task assignment module, draw its use-case diagram and describe the use-case using a table format . . . . .	9
1.3.1 Usecase diagram for Task assignment: . . . . .	9
1.3.2 Usecase scenario for Task assignment . . . . .	10
<b>2 Task 2: System modelling</b>	<b>17</b>
2.1 Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module . . . . .	17
2.1.1 Log in . . . . .	17
2.1.2 Main Screen . . . . .	17
2.2 Proposal a conceptual solution for the route planning task and draw a sequence diagram to illustrate it. . . . .	22
2.2.1 Create new route: . . . . .	22
2.3 Draw a class diagram of Task Assignment module as comprehensive as possible . . . . .	24
2.3.1 Assign task to employees . . . . .	24
2.3.2 Assign task to janitors . . . . .	25
2.3.3 Assign task to collectors . . . . .	26
<b>3 Task 3: Architecture design</b>	<b>27</b>

3.1	Describe an architectural approach you will use to implement the desired system. How many modules you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module . . . . .	27
3.2	Draw an implementation diagram for Task Assignment module . . . . .	32
<b>4</b>	<b>Task 4: Implementation – Sprint 1</b>	<b>35</b>
4.1	Setting up. The team creates an online repository (github, bitbucket, etc) for version control. folders this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files . . . . .	35
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 . . . . .	35
4.3	Implement MVP1 – design an interface 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. Design use a wireframe tool. . . . .	49
<b>5</b>	<b>Task 5: Implementation – Sprint 2</b>	<b>50</b>
5.1	Implement MVP2 – realize the design in MVP1 with a programming language (HTML, Javascript, Python, C#, etc) . . . . .	50
5.2	Demonstrate the whole project from Task 1 to Task 5 . . . . .	50
<b>6</b>	<b>Kết luận</b>	<b>51</b>

## 1 Task 1: Requirement elicitation

### 1.1 Identify the context of this project. Who are relevant stakeholders? What are their current needs? What could be their current problem? In your opinion, what benefits UWC 2.0 will be for each stakeholder?

Xác định ngữ cảnh của project (Identify the context of this project) :

- Với cuộc sống hiện đại ngày nay lượng rác thải mà các hộ gia đình, công ty và xí nghiệp thải ra ngày một nhiều hơn dẫn đến sự quá tải khiến cho việc quản lý chất thải đô thị là một trong những vấn đề quan trọng mà nhiều quốc gia trên thế giới phải đối mặt. Đây là một vấn đề cần phải xem xét kỹ lưỡng và tìm các biện pháp nhằm mục đích cải thiện trong Mục tiêu Phát triển Bền vững (SDG) 11: các thành phố và cộng đồng bền vững và SDG 6: nước sạch và vệ sinh. Trong bối cảnh đô thị, việc quản lý chất thải rắn rất tốn kém và không hiệu quả. Các chính phủ và tổ chức coi việc thu gom và quản lý chất thải được cải thiện là động lực tích cực để cải thiện thành phố, xã hội và môi trường.



Hình 1: Sustainable Development Goals

- Việc thu gom chất thải thường phải do các nhà cung cấp dịch vụ xử lý chuyên nghiệp chịu trách nhiệm. Quy trình thu gom chất thải điển hình bao gồm (1) người phụ trách (back officer), người vận hành hệ thống trung tâm để tạo lịch, điều phối người thu gom và người vệ sinh, (2) người thu gom (collector), người lái các loại phương tiện khác nhau để vận chuyển rác từ các "điểm thu gom lớn" (Major Collecting Point hay MCP) và mang đến các khu xử lý rác thải, (3) nhân viên vệ sinh (janitor) sẽ thu gom rác từ các smart bin trong khu vực và mang đến các MCP. Việc lên lịch và phân phối công việc được thực hiện giữa các nhóm vệ sinh và do các người phụ trách phối hợp thực hiện. Những nhiệm vụ này thường được giao hàng tuần. Như chúng ta biết thì có rất nhiều loại rác thải khác như rác dễ phân hủy, rác khó phân hủy, rác tái chế nên với mọi loại rác sẽ có những cách xử lý rác khác nhau dẫn đến sẽ có những khu xử lý rác khác. Ở đây, ta giả định rằng người dân cần phân loại rác thải tại nhà trước khi Janitor đến thu gom, nếu không có thể bị đưa vào blacklist. Nếu các khu xử lý rác thải đầy hoặc có vấn đề, rác thải sẽ được vận chuyển đến khu xử lý dự bị. Ngoài ra, việc thu gom rác thải sẽ được tổ chức và thực hiện theo một lịch trình cố định.
- Trước nhu cầu đơn giản hóa việc quản lý rác thải đô thị, tổ chức X đã ký hợp đồng phát triển hệ thống quản lý thông tin có tên là UWC 2.0 để nâng cao hiệu quả thu gom rác thải của các nhà cung cấp dịch vụ thu gom rác thải.

### **Các bên có liên quan (Relevant Stakeholders) :**

#### **External Stakeholders:**

- Chính phủ (Government)
- Người dân
- Tổ chức X (Organisation X)
- Nhà cung cấp dịch vụ Y (Service Provider Y)

#### **Internal Stakeholders:**

- Người phụ trách (Back officer)
- Nhân viên thu gom (Collector)
- Nhân viên vệ sinh (Janitor)
- Khu xử lý rác thải (Waste Processing Plant)

### **Nhu cầu của các bên liên quan (The stakeholders' current needs) :**

- **Chính phủ:** tìm các biện pháp nhằm mục đích cải thiện trong Mục tiêu Phát triển Bền vững (SDG) 11: các thành phố và cộng đồng bền vững và SDG 6: nước sạch và vệ sinh.
- **Người dân:** cần môi trường sống không bị ô nhiễm, cần có những biện pháp quản lý rác thải tốt hơn để môi trường xung quanh luôn trong lành và sạch đẹp. Ngoài ra, người dân cần thông tin về kế hoạch thu gom rác thải vì họ có thể phân loại rác thành các loại khác nhau và giao cho janitor theo đúng lịch trình.
- **Nhà cung cấp dịch vụ Y:** nâng cao hiệu quả thu gom rác thải của các nhà cung cấp dịch vụ thu gom rác thải.
- **Back officer:** Để quản lý các collector và janitor, back officer cần có quyền truy cập vào lịch làm việc của họ, thông tin chi tiết về phương tiện ... để phân công họ đến các MCP. Back officer cũng cần tạo lộ trình cho mỗi collector sao cho hiệu quả về mức tiêu thụ nhiên liệu và quãng đường di chuyển, vì thế họ phải có thông tin chi tiết về tất cả các MCP và sức chứa của chúng. Ngoài ra, nếu các khu xử lý có vấn đề thì back officer cần được thông báo ngay lập tức để có thể nhanh chóng điều phối collectors đến các khu xử lý rác thải khác.
- **Collectors:** Collector sẽ lái các loại xe thu gom rác để gom rác từ các MCP và vận chuyển đến các khu xử lý rác thải. Họ cần thông tin chi tiết về lịch làm việc hàng ngày, phương tiện đi lại, lộ trình ... Ngoài ra, Collector cần liên lạc, giao tiếp với các back officer và các janitor.
- **Janitors:** Janitor sử dụng xe đẩy để thu gom rác trong khu vực được phân công của họ và mang đến các MCP mỗi ngày. Họ cần thông tin chi tiết về lịch làm việc của mình cũng như khu vực được phân công của mình và có thể liên lạc với collector, các janitor khác và các back officer.
- **Khu xử lý rác thải:** Các khu xử lý rác thải cần có cách thông báo cho back officer trong trường hợp có vấn đề xảy ra khiến khu xử lý từ chối nhận thêm rác.

### **UWC 2.0 có thể mang lại lợi ích gì cho các bên liên quan (What benefits UWC 2.0 will be for each stakeholder) :**

**Back officers:**

- Có thể truy cập vào lịch làm việc của janitor và collector, có thể nhắn tin cho họ nếu cần.
- Có thể dễ dàng quản lý các phương tiện cũng như các MCP.
- Có thể dễ dàng phân công janitor và collector đến các MCP.
- Có thể dễ dàng tạo lộ trình cho collector.
- Có thể nhận được thông báo nhanh chóng khi có vấn đề xảy ra với các khu xử lý rác thải hoặc các MCP,
- Có danh sách những hộ gia đình bất hợp tác trong quá trình xử lý rác và có phương án xử lý phù hợp.

### **Collectors and Janitors:**

- Có thể xem lịch biểu, nhiệm vụ hằng ngày dễ dàng hơn.
- Có thể giao tiếp với các collector, janitor khác và back officer thuận tiện hơn.
- Có thể nhận thông báo nhanh chóng về những thay đổi trong lịch trình, nhiệm vụ, biến cố...
- Có thể ghi nhận các hộ gia đình bất hợp tác trong quá trình xử lý, phân loại rác cũng như có thể xem danh sách các hộ gia đình đó.

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

### **1.2.1 Functional requirements**

- Back officer:
  - Có thể xem task hiện tại của Collector và Janitor.
  - Chọn các khu vực (đơn vị nhỏ nhất là phường), trong mỗi khu vực đó sẽ phân công các janitors và trollers vào để hoạt động
  - Tạo hoặc chọn route dẫn tới MCP và khu xử lý rác, đồng thời phân công collector và vehicle vào.
  - Xem tình trạng của các khu chế xuất.
  - Được thông báo khi khu chế xuất có vấn đề.

- Được cảnh báo khi các MCPs bị đầy (quá 90% sức chứa).
- Gửi tin nhắn cho các janitors và collectors.
- Có thể truy cập vào blacklist những hộ gia đình bất hợp tác trong quá trình xử lý rác.

- Janitor

- Xem được lịch làm việc và cụ thể các công việc của họ theo đơn vị ngày và tuần.
- Liên lạc được với các collectors, janitors và back officers khác
- Có thể báo cáo nhà nào không phân loại rác.
- Xem được khu vực mình hoạt động
- Có thể nhận được thông tin về những thay đổi đột xuất.

- Collector

- Xem được lịch làm việc và cụ thể các công việc của họ theo đơn vị ngày và tuần.
- Liên lạc được với các collectors, janitors và back officers khác
- Có thể xem được route của mình
- Có thể nhận được thông tin về những thay đổi đột xuất.

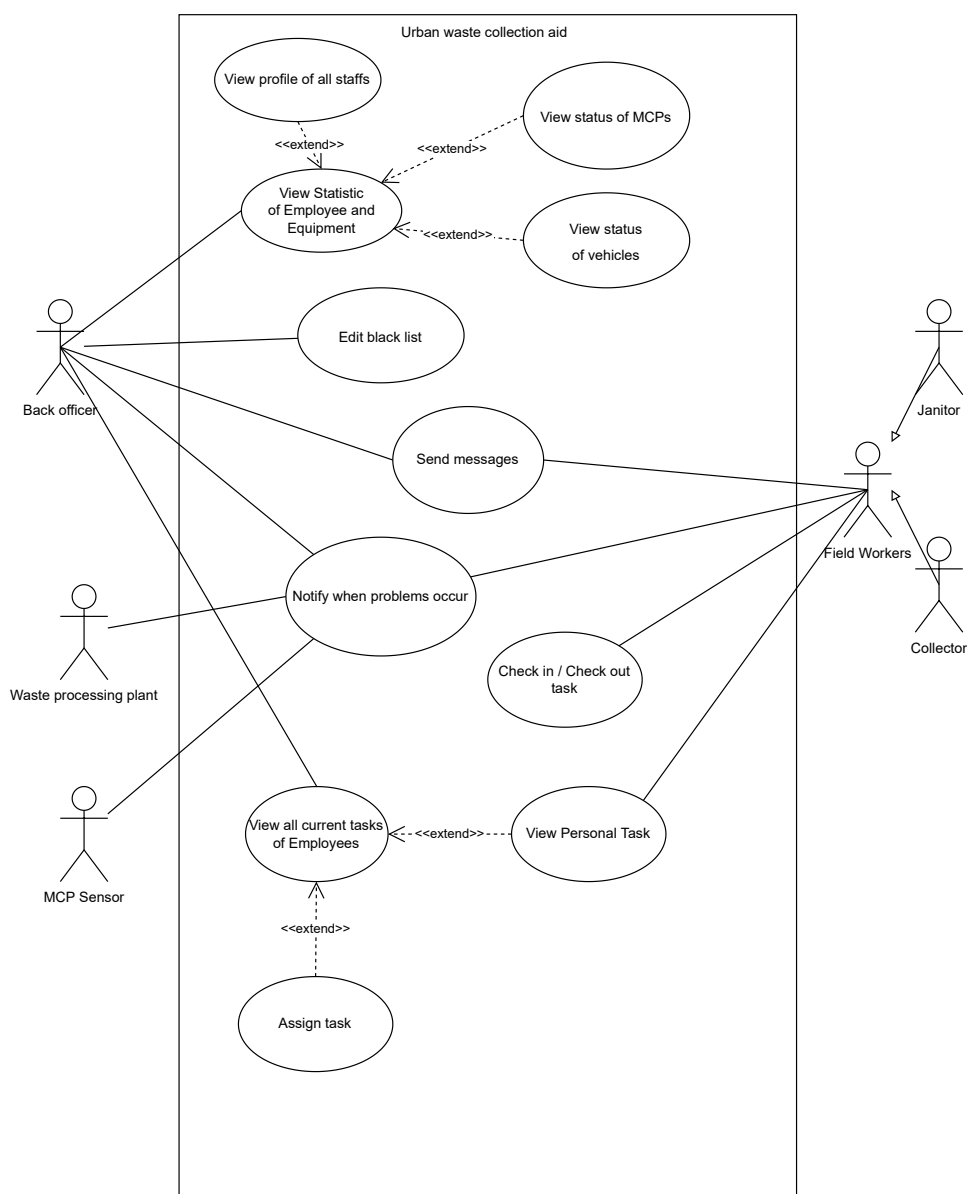
### 1.2.2 Non-functional requirements

- Giao diện thân thiện, có thể hiểu ngay cách sử dụng khi xem qua hướng dẫn 1 lần.
- Các thông tin về lịch làm việc phải được trình chiếu đầy đủ trên màn hình mà không cần phải lăn chuột xuống để xem thêm
- Dữ liệu về sức chứa của các MCPs nên được cập nhật liên tục mỗi 15 phút, và được cập nhật đầy đủ trong ít nhất 95% thời gian làm việc của họ
- Hệ thống phản hồi thông báo mới trong vòng 0,8s, đồng thời có thể gửi tin nhắn theo thời gian thực với độ trễ dưới 1s
- Có thể chứa dữ liệu thực của 1000 MCPs ở thời điểm hiện tại và 10000 MCPs trong vòng 5 năm tới
- Hệ thống có thể đáp ứng tối đa 200 users cùng lúc
- Chạy được trên hai nền tảng Google và Firefox



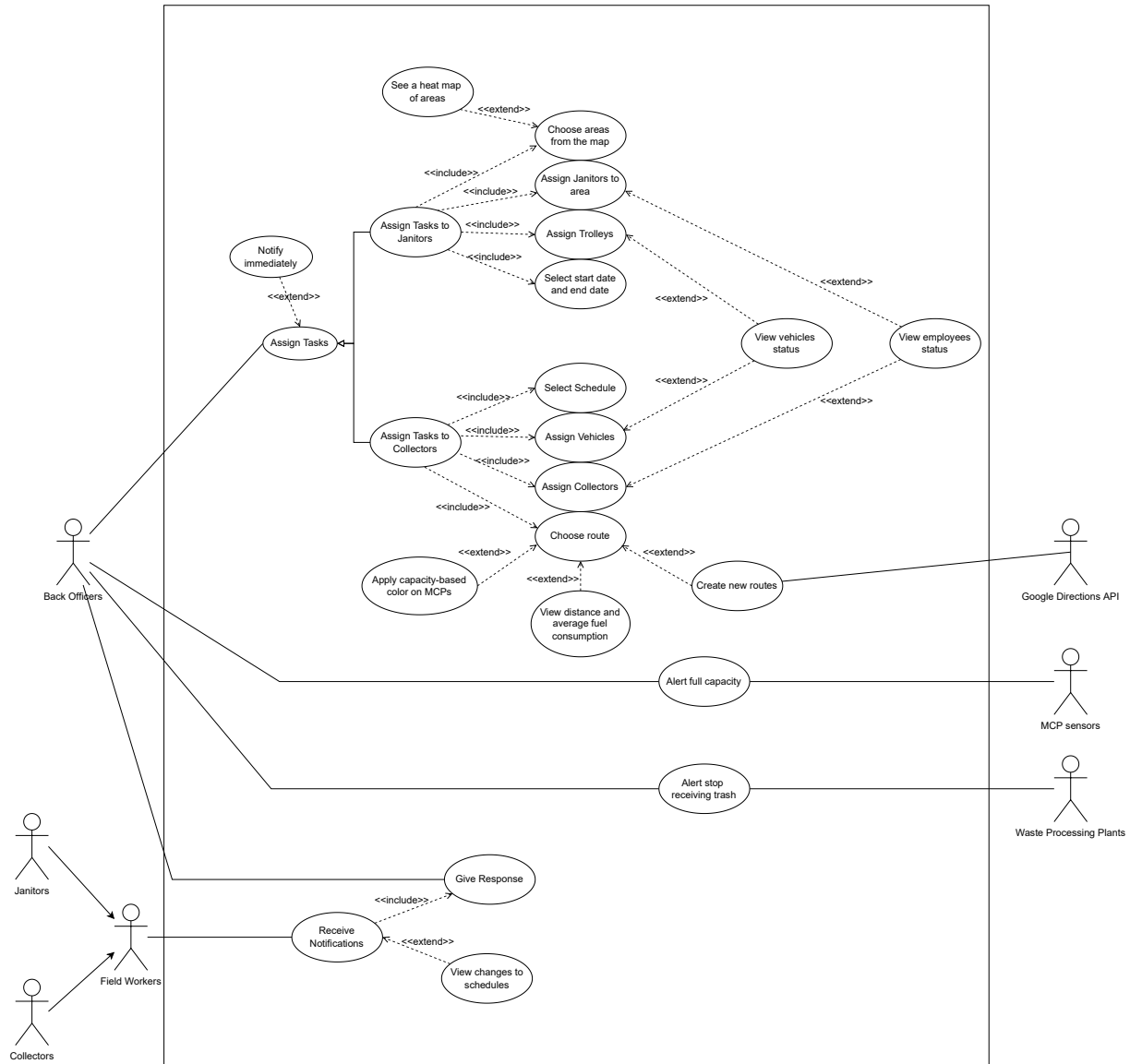
- Ứng dụng responsive, có hỗ trợ chạy trên desktop và mobile
- Hỗ trợ ngôn ngữ tiếng Việt, có thể hỗ trợ chuyển ngữ sang tiếng Anh trong tương lai

### 1.2.3 Use-case diagram cho hệ thống



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

#### 1.3.1 Usecase diagram for Task assignment:



### 1.3.2 Usecase scenario for Task assignment

#### 1. Usecase scenario for Assign task to Janitor

Use-case name	Assign task to Janitor
Description	Back officers phân công công việc cho Janitors hàng tháng.
Actor	Back officers và Janitors.
Trigger	Back officers nhấn vào nút assign task và chọn đối tượng là Janitor
Precondition	Back officer có task cần giao đến janitor
Postcondition	Mọi sự sửa đổi đối với Janitors đều được lưu vào hệ thống, cập nhật đến Lịch của Janitors và thông báo đến Janitors.
Normal flow	<p>Hiển thị form:</p> <ol style="list-style-type: none"> <li>Software hiển thị một form cho back officers điền và lưu lại thông tin Janitors, khu vực mà Back officers muốn phân và xe đẩy.</li> <li>Back officer điền các thông tin cần thiết vào form gồm: <ul style="list-style-type: none"> <li>• Chọn khu vực</li> <li>• Chọn Janitors</li> <li>• Chọn Trollers</li> <li>• Chọn Lịch</li> </ul> </li> <li>Back officers nhấn vào nút “Lưu thay đổi”, dữ liệu sẽ được cập nhật lên lịch của nhân viên.</li> </ol>
Alternative flow	<p><b>Để biết thêm thông tin chi tiết về số lượng dân cư của một khu vực, Back officers có thể xem “heat map” về phân bố dân cư:</b></p> <ol style="list-style-type: none"> <li>2.1.1 Back officers ấn vào nút “heat map” .</li> <li>2.1.2 Hệ thống lấy dữ liệu từ Database để cập nhật lên Heatmap.</li> <li>2.1.3 Một lớp “Heat map” dựa trên data đã lấy hiện lên trên bản đồ đã có sẵn.</li> </ol> <p>Đi đến bước 2</p>

Exception	<p><b>Khi không có mạng, không thể cập nhật thông tin đã điền lên Database</b></p> <p>3.1 Màn hình sẽ hiện ra “Không có kết nối internet”</p> <p>3.2 Bấm vào nút thử lại trong thông báo.</p> <p>Quay lại bước 3.</p> <p><b>Khi không có mạng, không thể lấy thông tin từ database trở về</b></p> <p>2.2 Màn hình sẽ hiện ra “không có kết nối internet “</p> <p>Quay lại bước 2.</p> <p><b>Khi đã phân bố hết Trollers</b></p> <p>2.3 Khi đã phân bố hết trollers, lúc này khung “chọn trollers” sẽ hiện lên một ô màu xám (nghĩa là không còn trollers nào để chọn, back officers không thể bấm vào được)</p> <p>Quay về bước 1 (tức là quay về form ban đầu)</p> <p><b>Khi đã phân bố hết Janitors</b></p> <p>2.4 Khi đã phân bố hết janitors, lúc này khung “chọn janitors” sẽ hiện lên một ô màu xám (nghĩa là không còn janitors nào để chọn, back officers không thể bấm vào được)</p> <p>Quay về bước 1 (tức là quay về form ban đầu)</p>
-----------	---

## 2. Usecase scenario for Assign task to Collectors

Use-case name	Assign task to Collectors
Description	Back officers phân công công việc cho Collectors hàng tháng.
Actor	Back officers và Collectors.
Trigger	Back officers nhấn vào nút assign task và chọn đối tượng là Collectors
Precondition	Back officers có task cần giao cho Collector
Postcondition	Mọi sự sửa đổi đối với Collectors đều được lưu vào hệ thống, cập nhật đến Lịch của Collectors và thông báo đến Collectors.

Normal flow	<p>Hiển thị form:</p> <ol style="list-style-type: none"><li>1. Khi Back officers bấm vào chức năng “Assign task to collectors”, trên màn hình sẽ hiện thị ra một form bao gồm các cột : Select Schedule, Assign Vehicles, Assign Collectors và Create Route thể hiện việc Back officers có thể chọn lịch trình cho collectors, phân xe, phân collectors và tạo lộ trình cho collectors.</li><li>2. Back officers điền các chọn các thông tin cần thiết trong form<ul style="list-style-type: none"><li>• Select Schedule</li><li>• Assign Vehicles</li><li>• Assign Collectors</li><li>• Choose route</li></ul></li><li>3. Khi bấm "Lưu thay đổi", dữ liệu sẽ được cập nhật vào hệ thống và cập nhật trên lịch của nhân viên.</li></ol>
Alternative flow	<p><b>Đối với những MCPs mới được thành lập, những khu vực mới được thêm vào cơ sở dữ liệu và chưa có tuyến đường đi qua, ta có thể tạo thêm những tuyến đường mới:</b></p> <p>2.1.1 Back officers nhấn vào nút “Create new route”. Back officers có thể chọn các MCP theo một thứ tự nhất định vào để tạo thành một lộ trình mới, đường đi giữa các MCP và khu chế xuất rác sẽ được tính toán bởi Google Directions API.</p> <p>2.1.2 Sau đó, các lộ trình mới được tạo sẽ được cập nhật lên bản đồ và Database.</p> <p><b>Ngoài ra, ta còn có thể coi được khoảng cách của lộ trình mà collectors đi, tính toán được lượng nhiên liệu tiêu thụ từ đó giúp tiết kiệm hơn</b></p> <p>2.2.1 Back officers có thể nhấn vào chức năng “View distance and average fuel consumption” để xem chiều dài lộ trình và nhiên liệu tiêu thụ</p> <p>2.2.2 Back officers bấm phím back ở góc trái màn hình để thoát chức năng.</p>

	<p><b>Tùy vào lượng rác mà Back officers có thể cân đối làm sao cho đủ dung tích của xe chở rác, cũng như tạo được một lộ trình hợp lý:</b></p> <p>2.3.1 Back officers nhấn vào nút chức năng “Apply capacity-based color on MCPs” để thêm màu cho những MCPs ứng với dung tích của từng MCPs</p> <p>2.3.2 Back officers nhấn “Lưu thay đổi” và nhấn phím back ở góc trái màn hình để thoát.</p>
Exception flow	<p><b>Khi không có mạng, không thể cập nhật thông tin đã điền lên Database</b></p> <p>3.1 Màn hình sẽ hiện ra “Không có kết nối internet”</p> <p>3.2 Bấm vào nút thử lại trong thông báo.</p> <p>Quay lại bước 3.</p> <p><b>Khi không có mạng, không thể lấy thông tin từ database trở về</b></p> <p>2.4 Màn hình sẽ hiện ra “không có kết nối internet “(và không thể load được lịch)</p> <p>Quay lại bước 2</p> <p><b>Khi đã phân bố hết Collectors</b></p> <p>2.5 Khi đã phân bố hết collectors, lúc này khung “chọn collectors” sẽ hiện lên một ô màu xám (nghĩa là không còn collectors nào để chọn, back officers không thể bấm vào được)</p> <p>Quay về bước 1 (tức là quay về form ban đầu)</p> <p><b>Khi đã phân bố hết Vehicles</b></p> <p>2.6 Khi đã phân bố hết vehicles, lúc này khung “chọn vehicles” sẽ hiện lên một ô màu xám (nghĩa là không còn vehicles nào để chọn, back officers không thể bấm vào được)</p> <p>Quay về bước 1 (tức là quay về form ban đầu)</p>

### 3. Usecase scenario for Alert full capacity

Use-case name	Alert full capacity
---------------	---------------------

Description	Ở các MCPs được gắn những cảm biến để đo lượng rác, nếu lượng rác trên 90% sức chứa của MCPs, các MCPs sensors sẽ gửi tín hiệu đến back officers, từ đó back officers có thể đưa ra quyết định thông báo thu gom khẩn cấp các MCPs trên.
Actor	MCPs sensors, Back officers.
Trigger	Khi các MCPs sensors đo được lượng rác vượt quá 90% sức chứa của các MCPs
Precondition	x.
Postcondition	x.
Normal flow	<ol style="list-style-type: none"><li>1. Khi cảm biến của các MCPs đo được lượng rác vượt quá 90% sức chứa của MCPs, cảm biến sẽ gửi tín hiệu đến “thanh thông báo trạng thái” của back officers.</li><li>2. Lúc này, “thanh thông báo trạng thái” của back officers sẽ hiển thị màu đỏ và dòng chữ “MCPs A đã vượt quá công suất”. Back officers có thể bấm vào thông báo đó,.</li><li>3. Back officers bấm vào thông báo trên sẽ hiện ra bản đồ với MCP đã đầy được highlight lên, trong đó khu vực được thông báo sẽ hiện màu đỏ đậm.</li><li>4. Khi bấm vào khu vực màu đỏ đậm, phần mềm sẽ đi đến Assign task to Collectors.</li></ol>
Alternative flow	<p><b>Khi Back Officer bỏ qua thông báo</b></p> <p>3.1 Khi Back officer không bấm vào thông báo trên thanh thông báo trạng thái, thanh thông báo trạng thái sẽ cứ hiện màu đỏ cho tới khi cảm biến ở MCPs đó đo được lượng rác dưới 90%, thông báo sẽ biến mất.</p> <p>Quay lại bước 3</p> <p><b>Khi Back Officer không muốn assign thêm collector</b></p> <p>4.1 Nếu Back officers sau khi xem heat map mà không muốn Assign thêm Collectors để thu gom, Back officer có thể thoát thông báo thông qua nút back ở góc trái màn hình.</p> <p>Quay lại màn hình chính</p>

Exception flow	<b>Khi các cảm biến bị hư hỏng, những cảm biến sẽ không cập nhật tình trạng trong 90% thời gian hoạt động</b> 1.1 Lúc này hệ thống sẽ đưa ra cảnh báo cho back officers về những cảm biến đã không gửi phản hồi trong 90% thời gian hoạt động
----------------	---

#### 4. Usecase scenario for Stop receiving trash

Use-case name	Alert stop receiving trash
Description	Khi những khu chế xuất hoạt động quá công suất (rác quá nhiều và khu xử lý rác chưa xử lý xong), lúc này khu xử lý rác có thể thông báo cho hệ thống để back officers có thể tính toán các đường đi để đưa lượng rác đó đến các khu xử lý khác.
Actor	Waste processing plants, Back officers.
Trigger	Khi những khu chế xuất hoạt động quá công suất, các khu chế xuất gửi thông báo đến Back officers
Precondition	x
Postcondition	x.
Normal flow	<ol style="list-style-type: none"><li>1. Khi các khu chế xuất rác vượt quá công suất, khu chế xuất gửi thông báo lên hệ thống, hệ thống gửi thông báo đến Back officers.</li><li>2. Trên “thanh thông báo trạng thái của back officers” sẽ hiện lên thông báo “Khu xử lý A đã vượt quá công suất”. Back officers có thể bấm vào thông báo đó</li><li>3. Khi Back officers bấm vào thông báo, lúc này hệ thống sẽ hiện ra bản đồ lộ trình, trong đó Khu chế xuất vừa thông báo đã vượt quá công suất sẽ được chuyển màu đỏ đậm.</li><li>4. Khi bấm vào icon “Khu chế xuất rác” màu đỏ đậm, sẽ có một phím chức năng “Bạn có muốn thay đổi lộ trình các collector đi đến khu chế xuất này”.</li><li>5. Khi bấm vào phím chức năng trên, back officers có thể điều chỉnh lại lộ trình đến các khu chế xuất khác có công suất chưa vượt quá quy định, sự thay đổi này sẽ được gửi đến collector thông qua thông báo.</li></ol>

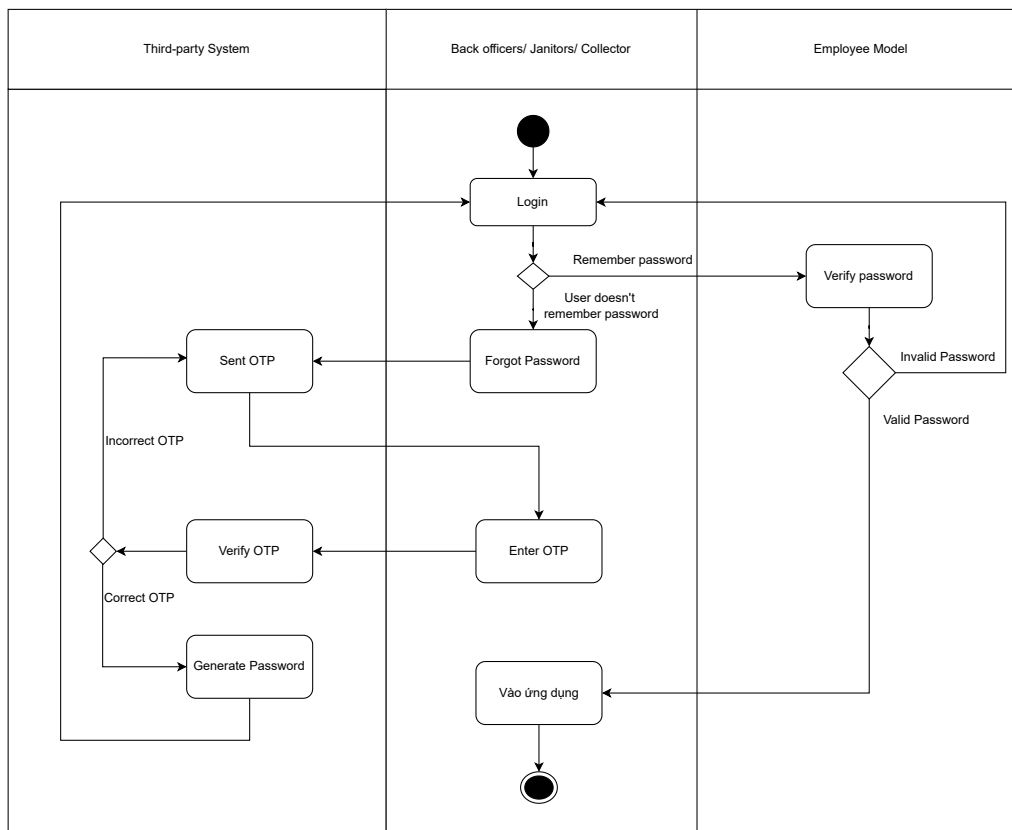


Alternative flow	3.1 Khi Back officer không bấm vào thông báo trên thanh thông báo trạng thái, thanh thông báo trạng thái sẽ cứ hiện màu đỏ cho tới khi các khu chế xuất confirm lại rằng công suất đã giảm dưới mức quy định thông báo sẽ biến mất. 4.1 Nếu Back officers sau khi xem bản đồ các khu chế xuất xung quanh mà không có khu chế xuất nào rảnh (có công suất thấp hơn mức quy định), back officers có thể assign vào khu dự bị của công ty.
Exception flow	x

## 2 Task 2: System modelling

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

#### 2.1.1 Log in

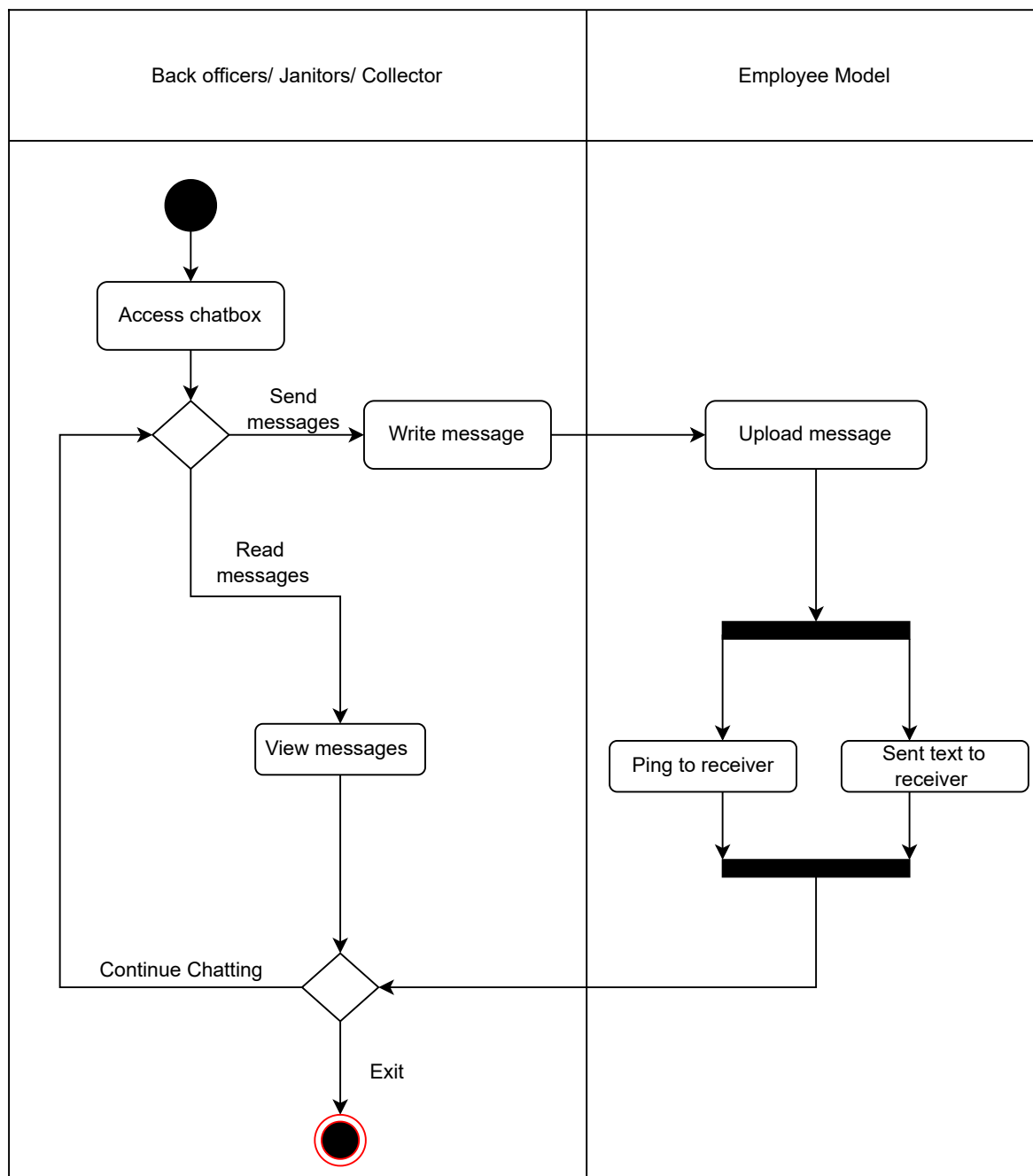


#### 2.1.2 Main Screen

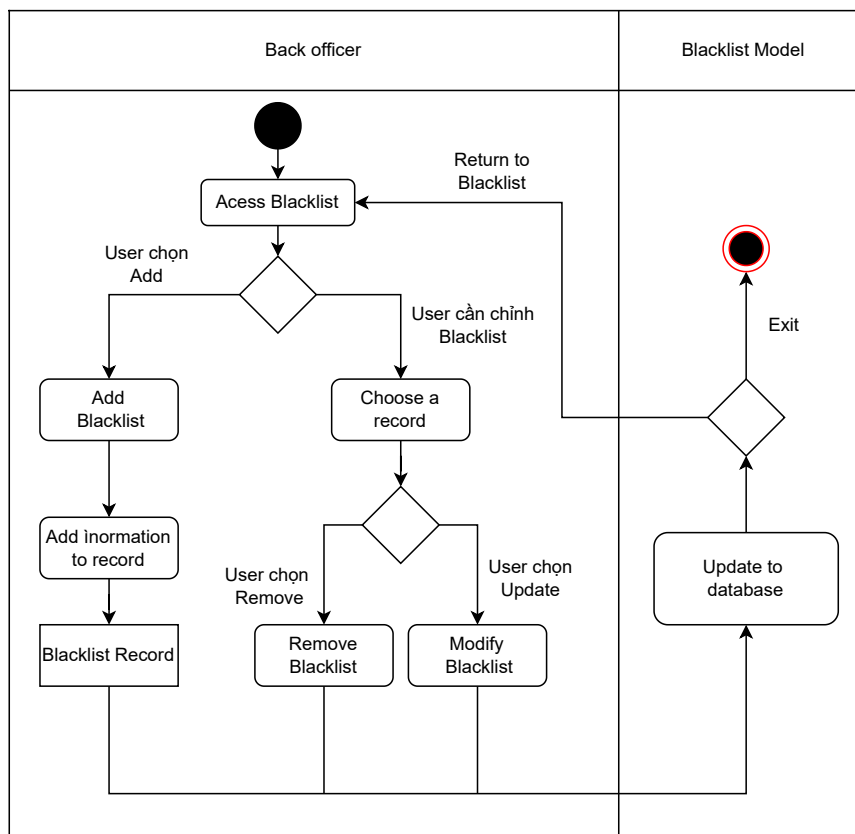
Ta sẽ chia mô hình của bài toán thành các module nhỏ hơn, cụ thể như sau

- Sent and read messages
- Recieve notification
- Record blacklist
- Manage Task

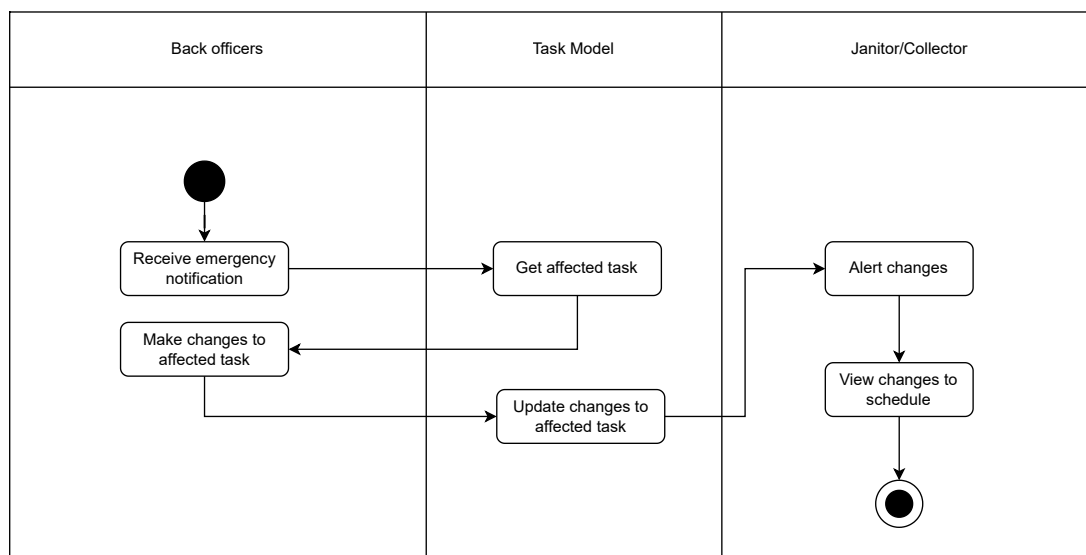
Với module sent messages thì activity diagram cụ thể như sau



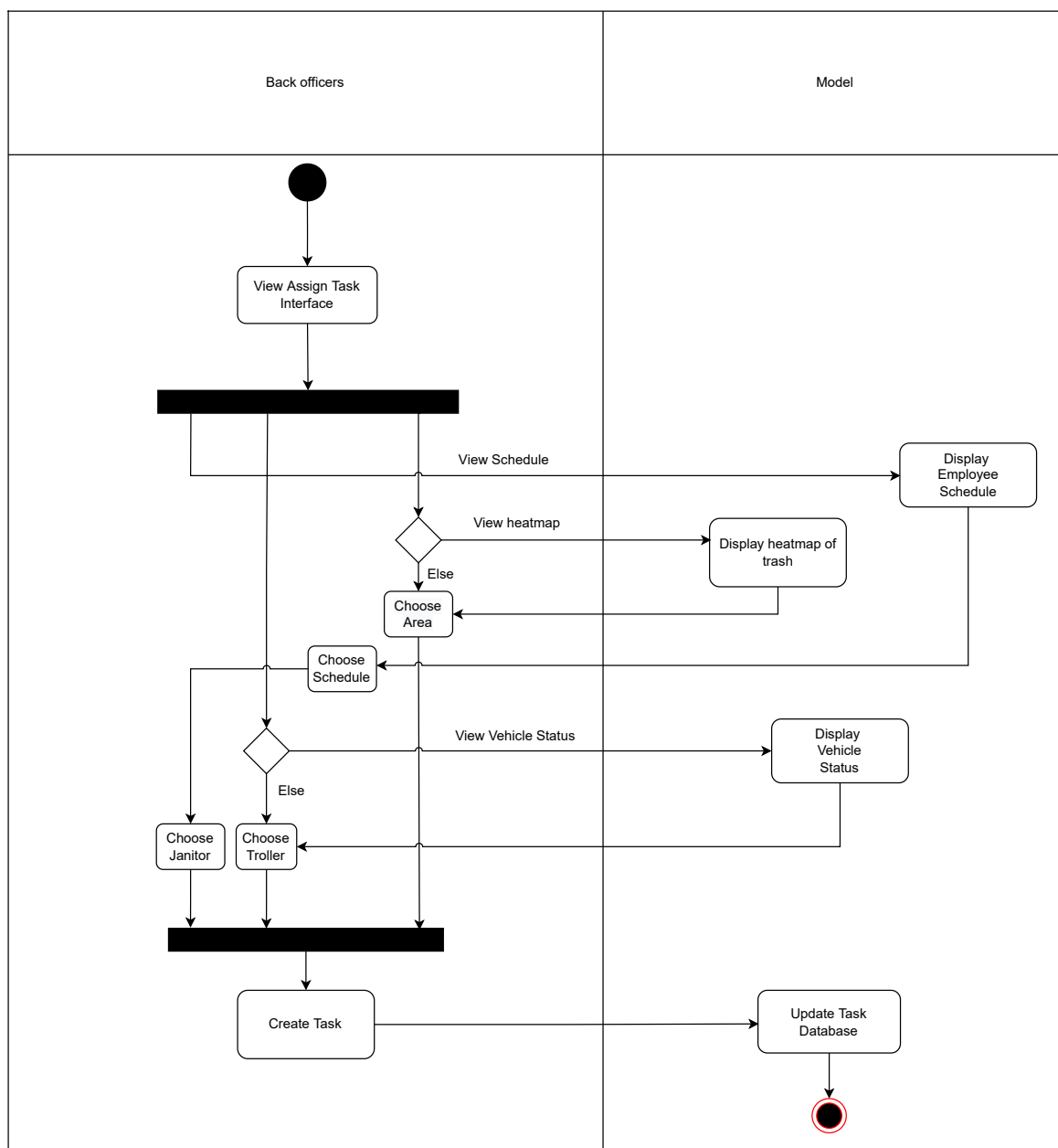
Với module record blacklist thì activity diagram cụ thể như sau



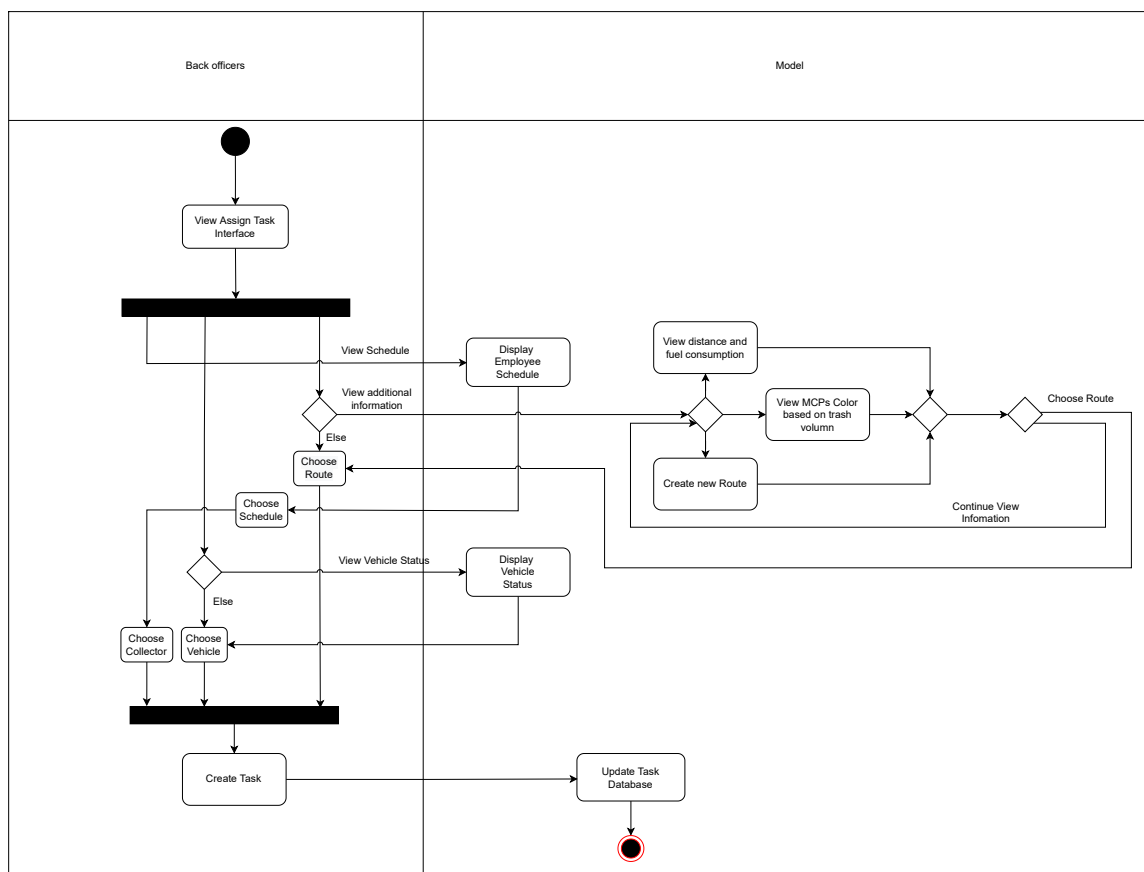
Với module recieve notification thì activity diagram cụ thể như sau



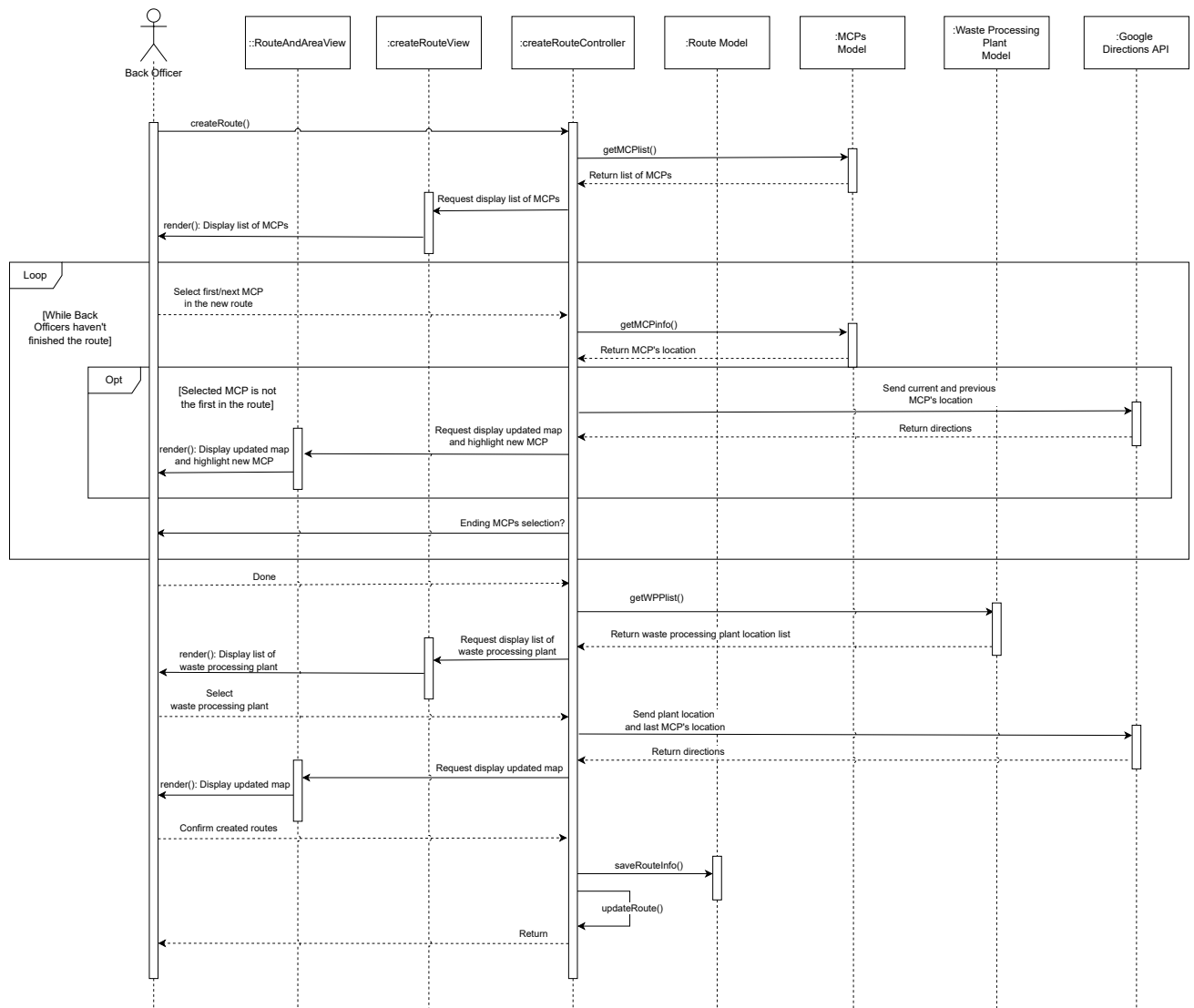
Với module manage task for janitor thì activity diagram cụ thể như sau



Với module manage task for collector thì activity diagram cụ thể như sau



## 2.2 Proposal a conceptual solution for the route planning task and draw a sequence diagram to illustrate it.



Hình 2: Sequence Diagram for Route Planning

### 2.2.1 Create new route:

Người dùng truy cập vào mục “Khởi tạo route mới”, Software UI sẽ gửi yêu cầu về danh sách các MCP cho Database của MCP và nhà máy xử lý rác thải (tạm thời gọi tắt là Database MCP). Database MCP sẽ trả về và hiển thị danh sách các MCP trên bản đồ. Software UI sẽ hiển thị thông báo yêu cầu người dùng hãy chọn các MCP.

Back officer sẽ chọn MCP đầu tiên (hoặc MCP tiếp theo). Software UI sẽ gửi yêu cầu về vị trí của MCP cho Database MCP. Nếu MCP được chọn không phải là MCP đầu

tiên, Database MCP sẽ tiếp tục gửi dữ liệu về vị trí của các MCP đã được chọn và MCP đang chọn cho Google Direction API. Google Direction API sẽ trả về lộ trình ngắn nhất nối giữa các MCP trên với nhau. Hệ thống sẽ cập nhật lộ trình hiện tại lên bản đồ đồng thời highlight MCP vừa được thêm vào để người dùng dễ dàng nhận biết. Cuối cùng Software UI sẽ gửi thông báo xác nhận rằng người dùng có muốn tiếp tục chọn các MCP hay không. Nếu người dùng xác nhận đã hoàn thành việc chọn MCP, hệ thống sẽ chuyển sang bước kế tiếp.

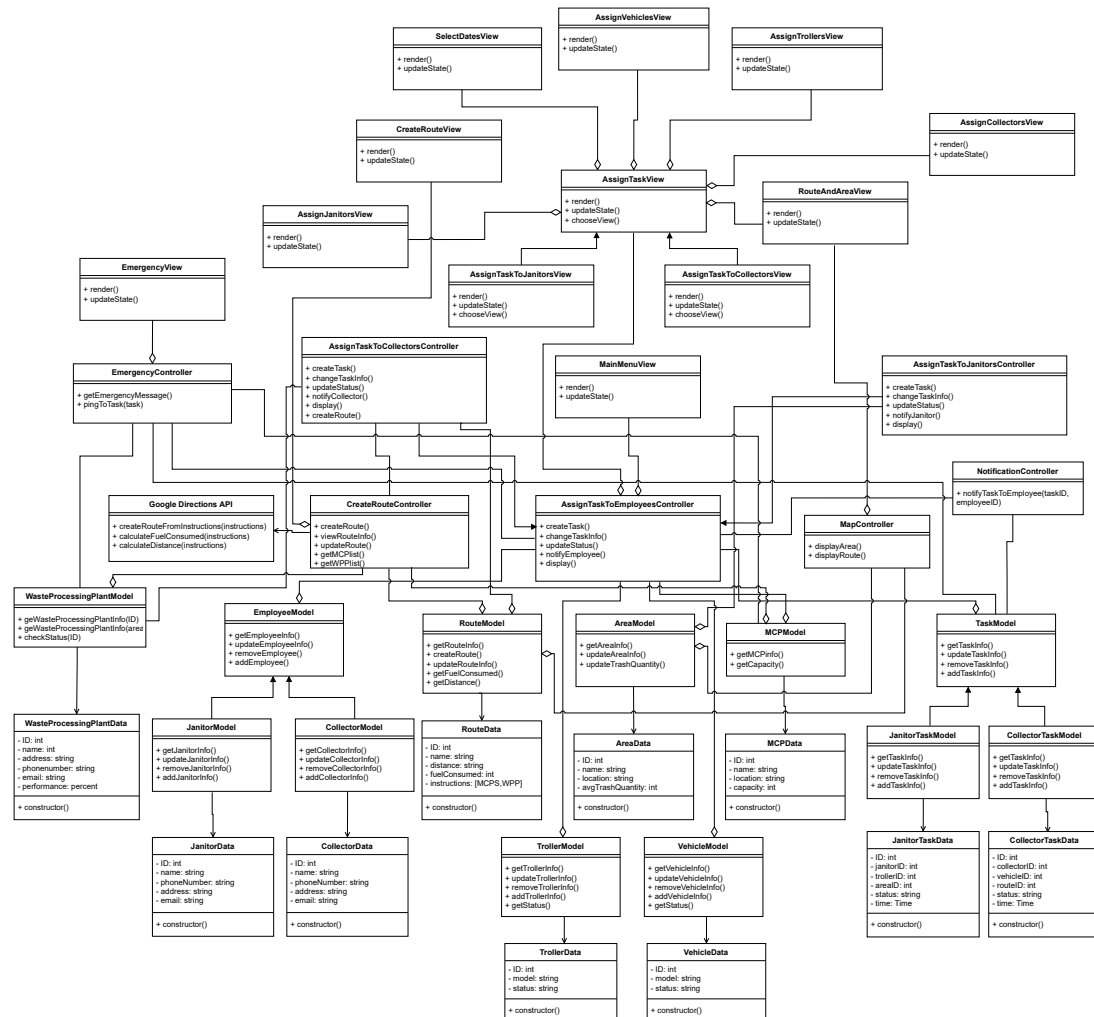
Sau khi hoàn thành chọn các MCP, người dùng sẽ chọn nhà máy xử lý rác thải. Software UI sẽ gửi yêu cầu về danh sách các nhà máy cho Database MCP. Database MCP sẽ trả về danh sách các nhà máy trên bản đồ. Software UI sẽ gửi yêu cầu người dùng chọn nhà máy xử lý rác thải trong danh sách.

Khi người dùng chọn một nhà máy nào đó, Software UI sẽ gửi yêu cầu về vị trí của nhà máy đó cho Database MCP. Database MCP sẽ gửi dữ liệu về vị trí của nhà máy được chọn và các MCP đã được chọn trước đó cho Google Direction API. Google Direction API sẽ thiết lập lộ trình và trả dữ liệu về lộ trình cho Database MCP. Hệ thống sẽ cập nhật lộ trình hiện tại lên bản đồ. Software UI sẽ yêu cầu người dùng xác nhận rằng mình muốn xác lập lộ trình này. Nếu người dùng xác nhận đồng ý, lộ trình sẽ được lưu vào Route Database đồng thời được cập nhật lên form.



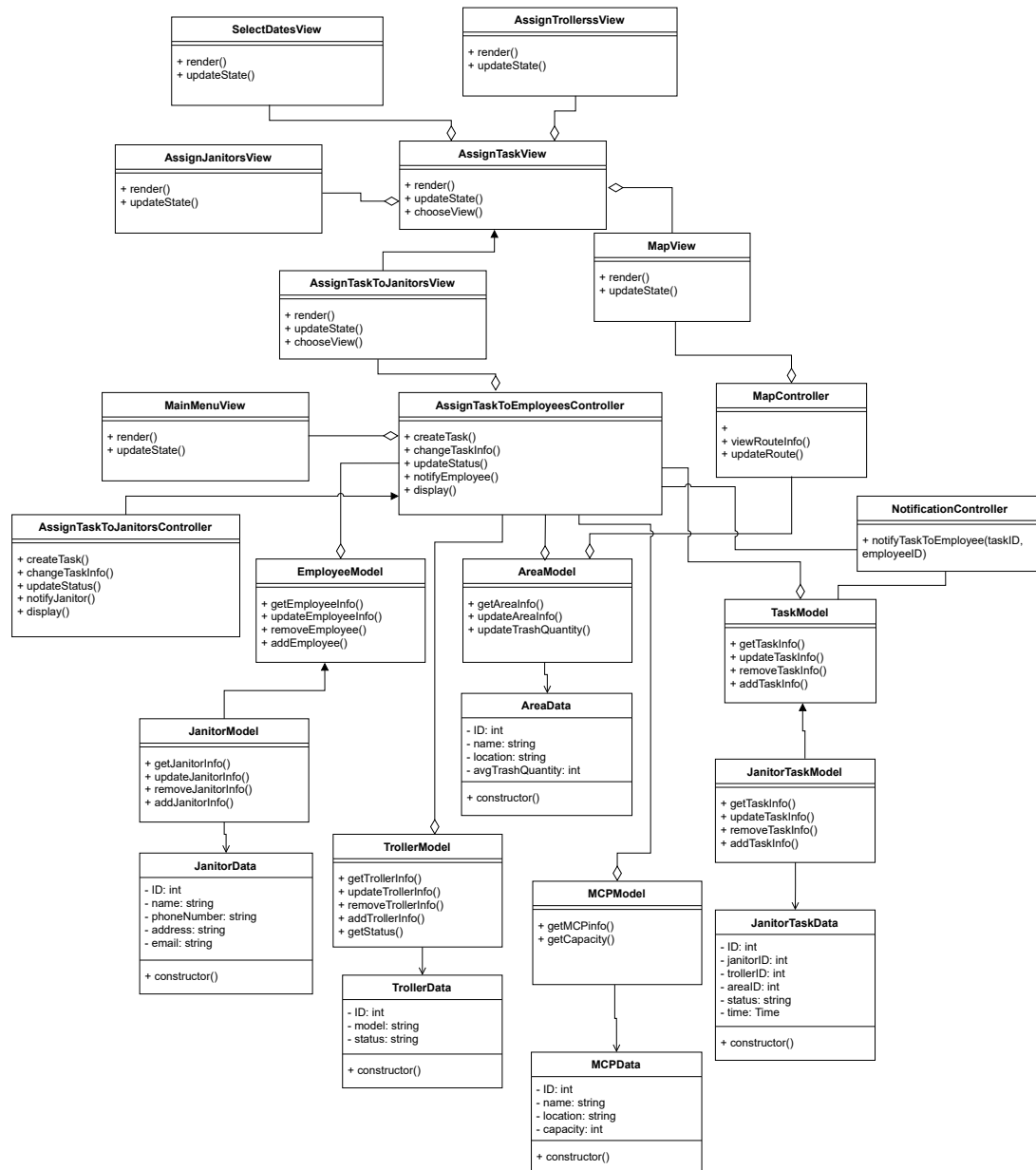
## 2.3 Draw a class diagram of Task Assignment module as comprehensive as possible

### 2.3.1 Assign task to employees



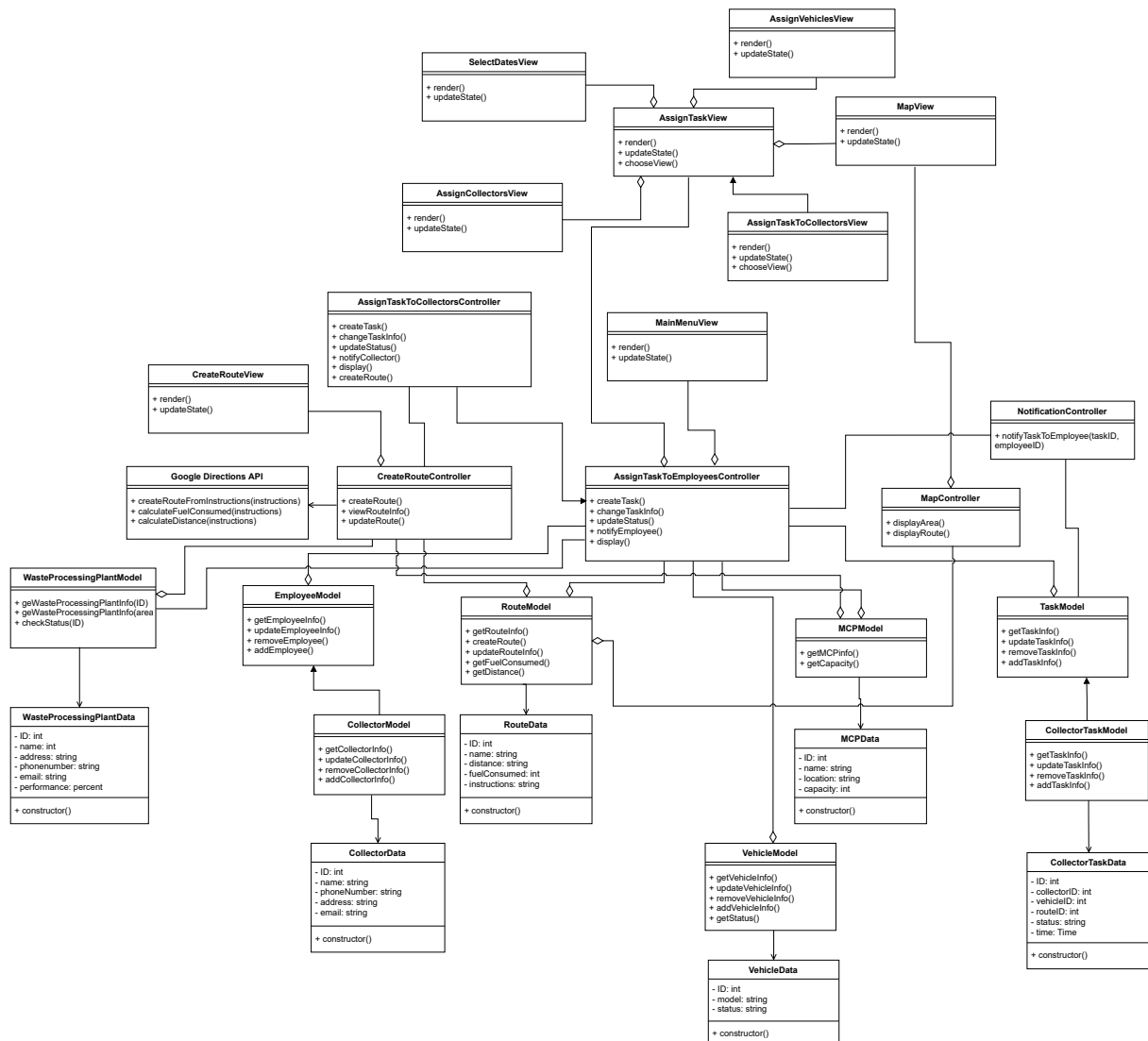
Hình 3: Class diagram for assign task to employees

### 2.3.2 Assign task to janitors



Hình 4: Class diagram for assign task to janitors

### 2.3.3 Assign task to collectors

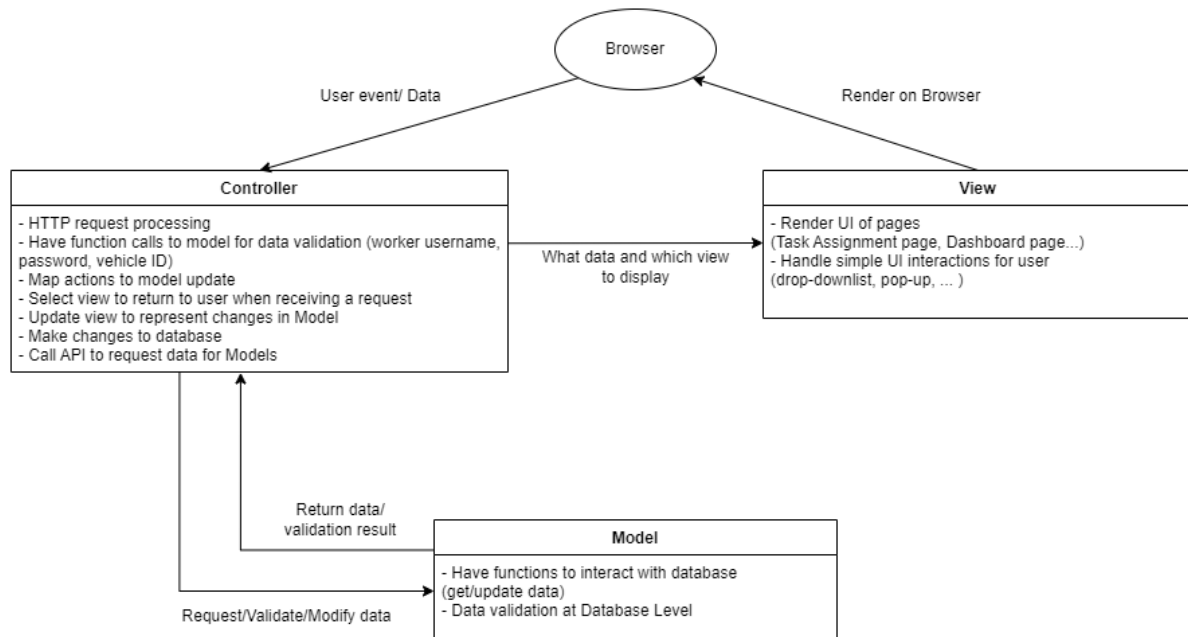


Hình 5: Class diagram for assign task to janitors

### 3 Task 3: Architecture design

#### 3.1 Describe an architectural approach you will use to implement the desired system. How many modules you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module

Nhóm sẽ sử dụng mô hình MVC để hiện thực hệ thống này



Hình 6: Sơ đồ mô hình MVC

Tên	Chi tiết kiến trúc MVC của hệ thống UWC 2.0
Mô tả	<p>- View:</p> <p>+ Render các UI của từng trang khác nhau từ sự điều khiển của Controller:</p> <ul style="list-style-type: none"><li>• Render các trang như xem lịch, assignment cho collector và janitor, trao đổi tin nhắn giữa back officer và employee</li><li>• Render các form, button... của các trang thực hiện assign task, gửi tin nhắn</li><li>• Render các form đăng nhập và đăng ký dành cho người dùng</li></ul> <p>+ Tương tác với người dùng bằng việc xử lý các sự kiện ngay trong view:</p> <ul style="list-style-type: none"><li>• Dropdown danh sách các vehicle, collector và janitor</li><li>• Hiện thông báo nổi khi có cảnh báo các MCPs bị đầy</li></ul> <p>- Model:</p> <p>+ Có thể thực hiện các thao tác lấy dữ liệu, cập nhật dữ liệu theo yêu cầu của Controller trong các chức năng như:</p> <ul style="list-style-type: none"><li>• Lấy danh sách nhân viên.</li><li>• Lấy danh sách các phương tiện.</li><li>• Lấy danh sách các tuyến đường, khu vực.</li></ul> <p>+ Kiểm tra dữ liệu ở mức Database và trả kết quả về cho Controller(kiểm tra xem có tồn tại nhân viên, phương tiện đã được gán cho nhân viên nào,...)</p>

	<p>- Controller:</p> <p>+ Xử lý các những HTTP Request đến từ user để chọn ra những trang thích hợp để render lên cho user.</p> <p>+ Cung cấp các phương thức để:</p> <ul style="list-style-type: none"> <li>• Nhận thông tin từ các hoạt động đăng nhập, đăng ký, truy xuất dữ liệu(MCP, nhân viên, lịch làm việc,...)</li> <li>• Kiểm tra kỹ càng các thông tin cũng như điều kiện hợp lệ trước khi truyền cho Model để truy xuất hoặc cập nhật dữ liệu(đúng cú pháp, quyền truy cập,...)</li> <li>• Sử dụng các phương thức từ model để thực hiện: <ul style="list-style-type: none"> <li>– Cập nhật task, thông tin vehicle, employee, lịch làm việc</li> <li>– Chỉnh sửa thông tin của task, route</li> <li>– Xóa task từ phía back officer, xóa các tin nhắn được trao đổi giữa các bên</li> </ul> </li> </ul> <p>+ Cung cấp các phương thức để xử lý việc gọi các API về các MCP trên bản đồ để thực hiện việc phân đường dành cho collector.</p> <p>+ Khi dữ liệu bên phần model có sự thay đổi (được thông báo bởi việc thay đổi trạng thái) thì phần view sẽ nhận được thông báo để cập nhật lại trang.</p>
Lợi ích	<p>- Việc tách ra thành ba thành phần giúp dễ quản lý, kiểm soát và khắc phục các vấn đề phát sinh khi sử dụng sản phẩm</p> <p>- Dễ dàng thêm các thành phần, requirement mới độc lập với những gì đang được hiện bên view của người dùng</p> <p>- Giúp tối ưu băng thông khi tránh sử dụng quá nhiều yêu cầu cùng một lúc gây tắc nghẽn đường truyền mạng</p>
Khó khăn	<p>- Đối với mô hình không quá phức tạp thì việc hiện thực code sẽ phức tạp hơn</p> <p>- Cần đảm bảo hiệu quả về tốc độ truyền tải khi tương tác giữa model - view - controller</p>

## **Những module sẽ được sử dụng trong phần mềm**

### **• Trao đổi tin nhắn giữa Back Officer và Collector/Janitor**

- Input: Người dùng sẽ nhập tin nhắn mà mình muốn gửi đi đến cho người nhận vào cho hệ thống, đồng thời người dùng cũng có thể chọn loại tin nhắn là tin nhắn thường hoặc tin nhắn với chức năng đặc biệt (thông báo về những thay đổi trong kế hoạch làm việc hoặc sự kiện khẩn cấp)
- Output: Hệ thống sẽ thông báo cho người nhận khi có tin nhắn mới, và đồng thời dẫn sang tính năng xem lịch khi là tin nhắn khẩn cấp
- Function: Giúp duy trì liên lạc giữa Back Officer và Collector/Janitor, có thể thông báo cho nhau khi có tình huống khẩn cấp xảy ra.

### **• Nhận thông tin về lượng rác hiện tại bên trong MCP**

- Input: Thông tin từ các sensor của MCP sẽ được truyền về cho ứng dụng trước khi được lưu vào Database.
- Output: Nếu như lượng rác vượt quá một ngưỡng nhất định thì sẽ hiện thông báo lên cho Back Officer để có thể đưa ra phương án xử lý. Nếu vẫn nằm ở mức ổn định thì sẽ lưu vào Database để lưu trữ.
- Function: Giúp Back Officer có thể thấy được tình trạng về lượng rác của MCP qua các thời điểm, cũng như xử lý kịp thời khi có tình trạng quá tải.

### **• Nhận báo động khi trung tâm xử lý rác bị quá tải**

- Input: Khi trung tâm xử lý rác bị quá tải thì ở phía trung tâm sẽ báo tín hiệu về cho ứng dụng. Ứng dụng sẽ hiện thông báo lên cho Back Officer.
- Output: Hệ thống sẽ báo động cho các Back Officer về việc đó cũng như những tuyến đường bị ảnh hưởng. Back Officer sẽ phải xử lý những tuyến đường đó.
- Function: Khi mà một trung tâm xử lý rác không còn khả năng xử lý rác (có thể do quá tải hoặc xảy ra vấn đề), những người quản lý ở trung tâm đó có thể thông báo cho các Back Officer để có thể xử lý.

### **• Tạo Task mới cho Janitor và Collector**

- Input: Back Officer sẽ nhập vào những thông tin cần thiết như nhân viên nhận Task, phương tiện, khu vực thực hiện, lịch trình vào và cập nhật lên hệ thống nếu không xảy ra sai sót.

- Output: Ứng dụng sẽ cập nhật vào Database về một Task mới được tạo, kèm theo các thông tin cần thiết. Sau đó, khi đến lịch trình của các Task thì sẽ thông báo cho nhân viên tương ứng.
- Function: Giúp Back Officer có thể phân công cho các nhân viên khác thực hiện được nhiệm vụ của mình một cách dễ dàng.

- **View tổng quát về các Task đang được thực hiện dành cho Back Officer**

- Input: Back Officer yêu cầu những thông tin như lịch trình, nhân viên, phương tiện, địa điểm của các Task.
- Output: Ứng dụng sẽ hiện lên một bảng tóm tắt của thông tin được yêu cầu, đồng thời cung cấp cho Back Officer những bảng thông tin chi tiết nếu cần.
- Function: Phục vụ mục đích thống kê cũng như giúp Back Officer nắm được các thông tin về những Task đã được phân công nhằm đưa ra quyết định khi cần tạo một Task mới.

- **View cụ thể về Task của các nhân viên**

- Input: Các nhân viên yêu cầu ứng dụng hiển thị những Task trong một khoảng thời gian nhất định.
- Output: Ứng dụng sẽ lấy dữ liệu về các Task và hiển thị những Task của nhân viên đó hoặc đối với Back Officer thì Back Officer sẽ có thể xem những Task của từng nhân viên cụ thể.
- Function: Giúp cho nhân viên Janitor/Collector có thể biết được công việc cụ thể của mình. Giúp cho Back Officer có thể giám sát được quá trình thực hiện Task của các nhân viên để có thể có những điều chỉnh nếu cần thiết.

- **Ghi nhận vào Blacklist các hộ gia đình không tuân thủ quy định, bất hợp tác**

- Input: Các nhân viên Janitor khi thu gom rác có thể ghi nhận lại thông tin của các hộ gia đình không phối hợp trong quá trình thu gom rác và gửi về hệ thống.
- Output: Ứng dụng khi nhận được thông tin sẽ ghi chú lại vào danh sách Blacklist và thông báo lại cho Back Officer.
- Function: Back Officer dựa vào những thông tin về những hộ gia đình trong Blacklist có thể thống kê và đưa ra những quyết định phù hợp trong việc xử lý những trường hợp đó.



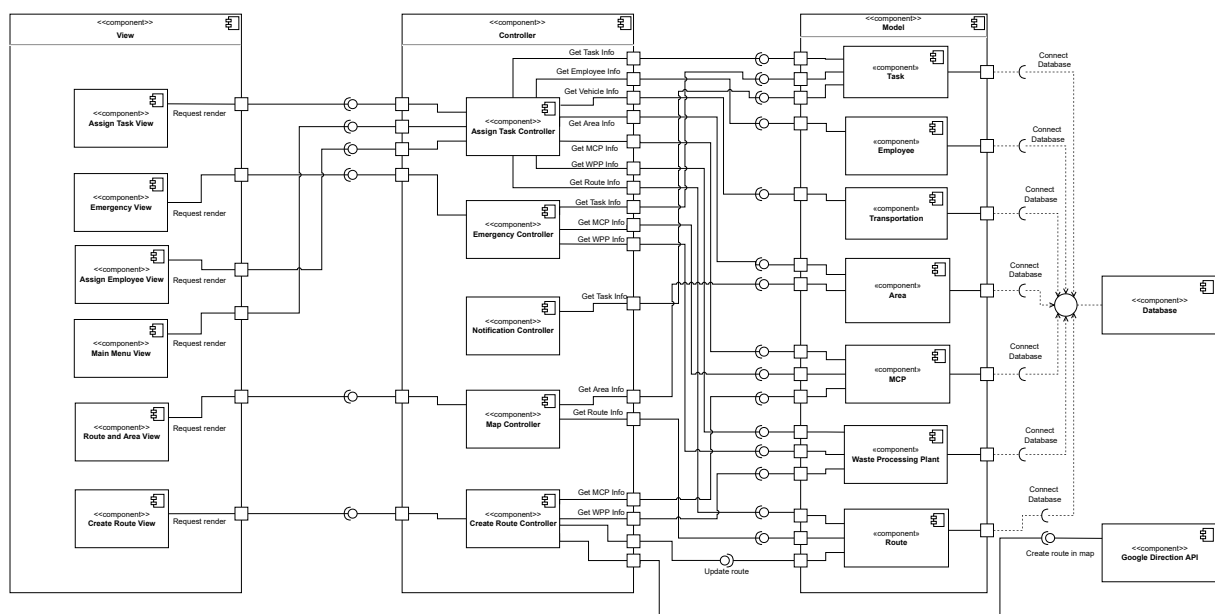
### • Tạo một Route mới

- Input: Nhận về một danh sách các MCP, thể hiện thứ tự mà Collector cần phải đi qua.
- Output: Ứng dụng sẽ cần phải đưa từng MCP theo thứ tự cho Google Directions API để biết được đường đi cụ thể giữa các MCP.
- Function: Giúp cho Back Officer có thể tạo route giữa các MCP một cách hiệu quả nhất.

### • Thể hiện các khu vực và đường đi lên bản đồ

- Input: Ứng dụng sẽ nhận được yêu cầu là phải thể hiện lên trên bản đồ một khu vực/đường đi nhất định, đưa yêu cầu này xuống Database để lấy được các dữ liệu cần thiết.
- Output: Ứng dụng, dựa vào những thông tin đã nhận được từ Database, sẽ thể hiện lên trên bản đồ những thông tin cần thiết như các MCP bên trong và dung lượng của chúng đối với một khu vực, độ dài quãng đường cũng như hao phí xăng đối với một đường đi.
- Function: Giúp cho Back Officer có thể đưa ra quyết định khi cần tạo một Task mới cho nhân viên.

## 3.2 Draw an implementation diagram for Task Assignment module



Mô tả component diagram:

- **Hệ thống có 3 component lớn:**

- *View component*

- \* Chứa các view component con
    - \* Các view component đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1
    - \* Các view component nhận request từ Controller và thực hiện render để hiển thị lên màn hình.

- *Controller component*

- \* Chứa nhiều controller component con.
    - \* Các controller component đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1
    - \* Các controller component cung cấp các interface cho view để thực hiện các request, sử dụng các interface từ model để thực hiện thao tác với cơ sở dữ liệu.

- *Model component*

- \* Chứa nhiều model component con
    - \* Các model component đảm nhận các nhiệm vụ như đã được mô tả trong task 3.1
    - \* Các model component cung cấp các interface để controller thực hiện thao tác với cơ sở dữ liệu.
    - \* Ngoài ra, model component còn sử dụng các interface từ các database component để thực hiện các tác vụ thao tác trực tiếp trên cơ sở dữ liệu tương ứng (đã mô tả trên lược đồ).
    - \* Các model component cung cấp và sử dụng các interface của nhau (đã mô tả trên lược đồ).

- **Ngoài ra còn có các component**

- *Database Component*

- \* Thao tác trực tiếp với cơ sở dữ liệu.
    - \* Cung cấp các interface để model thực hiện truy xuất trực tiếp trên cơ sở dữ liệu

- *Google Direction API*



- \* Cung cấp các interface cho model để tính toán đường đi, nguyên liệu... giữa các MCPs cũng như Waste Processing Plant.

## 4 Task 4: Implementation – Sprint 1

**4.1 Setting up.** The team creates an online repository (github, bitbucket, etc) for version control. folders this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files

Thầy có thể xem các thư mục yêu cầu trên link github:

<https://github.com/LMN1590/LauDaiTinhAi-UWC>

**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

Git log commit

```
commit 9c9451a97abf8b5fb261fcb68cf3aa03a97d3abb (HEAD -> master ,
origin/master)
```

```
Author: Nghia Le <minhngghiale1590@gmail.com>
```

```
Date: Wed Dec 14 23:06:22 2022 +0700
```

Updated all diagram

```
commit 42ea12551045188c36c2b2773036c963b2316dfa
```

```
Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>
```

```
Date: Sun Dec 11 16:58:10 2022 +0700
```

Seem lookable now

```
commit cce703e1aca13aa55ada04fec54c073cf30d88bc
```

```
Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>
```

```
Date: Sun Dec 11 14:07:33 2022 +0700
```

ticket

```
commit 0e0139b0556ee091fbbaa7edfa8911b181532d37c
```

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sun Dec 11 12:48:55 2022 +0700

update tasklist

commit a7800cafa9585d172930cbc51e61f6c5c37a8e28

Merge: afe6403 fc69e28

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sat Dec 10 21:59:54 2022 +0700

update thu

commit afe6403a5b75ca8ad1c64515de1cd21afa9fd7c3

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sat Dec 10 21:58:43 2022 +0700

update thu

commit fc69e28670154aec7a0051a1440d378e75e20410

Merge: b994ee0 25b2803

Author: VuTran2601 <75901482+vutran2601@users.noreply.github.com>

Date: Sat Dec 10 13:29:58 2022 +0700

Merge pull request #6 from LMN1590/Vu

Updates

commit 25b2803be2038cbc2a9154adafd2651c1ea9fab8

Author: VuTran2601 <nguyenvutran2601@gmail.com>

Date: Sat Dec 10 10:10:43 2022 +0700

Updates

commit b994ee0d3d207a992b9628bcfc08b491c1a827b9

Merge: 11c1239 07d2d1e

Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>

Date: Thu Dec 8 10:56:04 2022 +0700

Merge pull request #5 from LMN1590/hung

Finish linechart

commit 07d2d1e1bb4ec43195104bf5a06f23924ab8d190

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Wed Dec 7 18:58:36 2022 +0700

Finish linechart

commit 11c1239a23c7ad159dae9c6853238e17cca88b5c

Merge: a7f0e8c 69e3777

Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>

Date: Tue Dec 6 22:00:18 2022 +0700

Merge pull request #4 from LMN1590/hung

Edit components

commit 69e3777ffd4df6fecc5ffcfd6a879ee0f7a332d2

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Tue Dec 6 21:58:02 2022 +0700

Edit components

commit a7f0e8c708d31c19cf1c1c289041493a44145e85

Merge: 0b6b62e 652c34f

Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>

Date: Sun Dec 4 10:29:38 2022 +0700

Merge pull request #3 from LMN1590/hung

create react app and finish sidebar with header

commit 652c34f3e1741069ade3dff0418c4d6d7ae19318

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Sun Dec 4 10:28:10 2022 +0700

create react app and finish sidebar with header

commit 0b6b62e930c57a0ec2d866a720ad6ef0f18e80bc (HEAD -> master, origin/master, origin/HEAD)

Author: Vinh <khanhvinhbui0512@gmail.com>

Date: Mon Nov 14 19:06:28 2022 +0700

Mot so Activity Diagram

commit 438877f033e939b317946a797446f21030874261

Author: nhantran18 <80414510+nhantran18@users.noreply.github.com>

Date: Mon Nov 14 17:43:43 2022 +0700

Delete Report.md

commit 3f9bbca37648a36a6480e7e6ed9b722c50ccb6e4

Merge: e821362 76af436

Author: VuTran2601 <75901482+vutran2601@users.noreply.github.com>

Date: Mon Nov 14 17:21:41 2022 +0700

Merge pull request #2 from LMN1590/Vu

Vu

commit e821362df1f44f46e69177971a5c3af927308654

Author: nhantran18 <80414510+nhantran18@users.noreply.github.com>

Date: Mon Nov 14 16:06:31 2022 +0700

Report

commit 76af4369dd7c5b62d8eacc8ef50a492c2f04a2c0 (origin/Vu)

Author: VuTran2601 <nguyenvutran2601@gmail.com>

Date: Mon Nov 14 14:36:39 2022 +0700

Add **in** Task 2.3 – Alert Stop Receiving Trash

commit e502e557d05b188fb22931703461a230a7b4ef9f

Author: VuTran2601 <nguyenvutran2601@gmail.com>

Date: Mon Nov 14 14:36:21 2022 +0700

Add **in** Task 2.1 – Record Blacklist

commit cb9c40d1c87ad399845bc0fa8c5b223895e40a08

Merge: 874ea88 8fafb73

Author: Nguyen Huu Danh <113421911+DanhNguyen02@users.noreply.github.com>

Date: Mon Nov 14 10:29:37 2022 +0700



Add some documents by Danh – Merge pull request from branch  
Danh

Add some documents by Danh

```
commit 8fafb731c85b63508747f9032c2002a8c6a9f30a (origin/danh)
Author: NguyenHuuDanh <113421911+DanhNguyen02@users.noreply.
github.com>
Date: Mon Nov 14 10:18:31 2022 +0700
```

Add Use **case** diagram of UWC 2.0 – Task 1.2

```
commit fb714849e933b498b422bc80145fe556432177d9
Author: NguyenHuuDanh <113421911+DanhNguyen02@users.noreply.
github.com>
Date: Mon Nov 14 10:18:09 2022 +0700
```

Add class diagram of Assign Collectors module – Task 2.3

```
commit 874ea88de5f3b28aa71e09e70d2cc976a81a7791
Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>
Date: Mon Nov 14 09:41:18 2022 +0700
```

Added Use–**case** Task Assignment and Class Diagram – Receive  
Noti

```
commit dc82c50ade6293085cb017e4f5e41f6406a37749
Author: AD72001 <duy.pham20092001@hcmut.edu.vn>
Date: Sun Nov 13 18:47:15 2022 +0700
```

Added Activity Diagram – Login – Task 2.1

```
commit c7413afd05dded81968af2c3458e7caa132b9bda
```

Author: AD72001 <duy.pham20092001@hcmut.edu.vn>

Date: Sun Nov 13 18:46:51 2022 +0700

Added ClassDiagram–Alert Full Capacity – Task 2.3

commit 3dce7488a8c53c76ccee353a4757cc4a5f1249e7

Author: nhantran18 <80414510+nhantran18@users.noreply.github.com>

Date: Sun Nov 13 17:11:40 2022 +0700

Add files via upload

Class Diagram Assign Janitor

commit 0dd5ee39612d8c2832973e9943a171d06b996683

Author: LMN1590 <74974626+LMN1590@users.noreply.github.com>

Date: Sun Nov 13 00:19:51 2022 +0700

Create README.md

commit 3aa6c7b34770532e02ede77bf237b1fe725376d0

Author: unknown <minhngghiale1590@gmail.com>

Date: Sun Nov 13 00:18:34 2022 +0700

Added MVC Model – Task 3.1

commit 5b0e24de352c305536513db1d8420f981e91e731

Author: unknown <minhngghiale1590@gmail.com>

Date: Sun Nov 13 00:15:14 2022 +0700

Added Sequence Diagram **for** Task Assignment module – Task 2.2

commit 795f246e17bf55a0d1ee9d3b3f6c6a1015ed73f8

Author: unknown <minhngghiale1590@gmail.com>

Date: Sat Nov 12 22:01:56 2022 +0700

Initialize version control system  
(END)

```
commit 795f246e17bf55a0d1ee9d3b3f6c6a1015ed73f8
Author: unknown <minhngghiale1590@gmail.com>
Date: Sat Nov 12 22:01:56 2022 +0700

    Initialize version control system
(END)
```

Hình 7: Thêm Requirement

```
commit 5b0e24de352c305536513db1d8420f981e91e731
Author: unknown <minhngghiale1590@gmail.com>
Date: Sun Nov 13 00:15:14 2022 +0700

    Added Sequence Diagram for Task Assignment module - Task 2.2
```

Hình 8: Thêm Sequence Diagram cho Module Task Assignment

```
commit 3aa6c7b34770532e02ede77bf237b1fe725376d0
Author: unknown <minhngghiale1590@gmail.com>
Date: Sun Nov 13 00:18:34 2022 +0700

    Added MVC Model - Task 3.1
```

Hình 9: Thêm MVC Model

```
commit 0dd5ee39612d8c2832973e9943a171d06b996683
Author: LMN1590 <74974626+LMN1590@users.noreply.github.com>
Date: Sun Nov 13 00:19:51 2022 +0700

    Create README.md
```

Hình 10: Tạo file README.md

```
commit 3dce7488a8c53c76ccee353a4757cc4a5f1249e7
Author: nhantran18 <80414510+nhantran18@users.noreply.github.com>
Date: Sun Nov 13 17:11:40 2022 +0700
```

Add files via upload

Class Diagram Assign Janitor

Hình 11: Thêm Class Diagram - Assign Janitor

```
commit c7413afd05dded81968af2c3458e7caa132b9bda
Author: AD72001 <duy.pham20092001@hcmut.edu.vn>
Date: Sun Nov 13 18:46:51 2022 +0700
```

Added ClassDiagram-Alert Full Capacity - Task 2.3

Hình 12: Thêm Class Diagram - Full Capacity

```
commit dc82c50ade6293085cb017e4f5e41f6406a37749
Author: AD72001 <duy.pham20092001@hcmut.edu.vn>
Date: Sun Nov 13 18:47:15 2022 +0700
```

Added Activity Diagram - Login - Task 2.1

Added Activity Diagram - Login - Task 2.1

Hình 13: Thêm Activity Diagram - Log In

```
commit 874ea88de5f3b28aa71e09e70d2cc976a81a7791
Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>
Date: Mon Nov 14 09:41:18 2022 +0700
```

Added Use-case Task Assignment and Class Diagram - Receive Noti

Hình 14: Thêm Use-case Diagram Task Assignment và Class Diagram - Receive Notification

```
commit fb714849e933b498b422bc80145fe556432177d9
Author: NguyenHuuDanh <113421911+DanhNguyen02@users.noreply.github.com>
Date: Mon Nov 14 10:18:09 2022 +0700
```

Add class diagram of Assign Collectors module - Task 2.3

```
commit 8fafb731c85b63508747f9032c2002a8c6a9f30a (origin/danh)
Author: NguyenHuuDanh <113421911+DanhNguyen02@users.noreply.github.com>
Date: Mon Nov 14 10:18:31 2022 +0700
```

Add Use case diagram of UWC 2.0 – Task 1.2

```
commit cb9c40d1c87ad399845bc0fa8c5b223895e40a08
Merge: 874ea88 8fafb73
Author: Nguyen Huu Danh <113421911+DanhNguyen02@users.noreply.github.com>
Date: Mon Nov 14 10:29:37 2022 +0700
```

Add some documents by Danh – Merge pull request from branch Danh

Add some documents by Danh

Hình 15: Thêm Use-case Diagram UWC 2.0 và Class Diagram - Assign Collector

```
commit e502e557d05b188fb22931703461a230a7b4ef9f
Author: VuTran2601 <nguyenvutran2601@gmail.com>
Date: Mon Nov 14 14:36:21 2022 +0700
```

Add in Task 2.1 – Record Blacklist

Hình 16: Thêm Activity Diagram - Record Blacklist

```
commit 76af4369dd7c5b62d8eacc8ef50a492c2f04a2c0 (origin/Vu)
Author: VuTran2601 <nguyenvutran2601@gmail.com>
Date: Mon Nov 14 14:36:39 2022 +0700
```

Add in Task 2.3 – Alert Stop Receiving Trash

Hình 17: Thêm Class Diagram - Alert Stop Receiving Trash

```
commit e821362df1f44f46e69177971a5c3af927308654
Author: nhantran18 <80414510+nhantran18@users.noreply.github.com>
Date: Mon Nov 14 16:06:31 2022 +0700
```

Report

Hình 18: Thêm link tới Report

```
commit 3f9bbca37648a36a6480e7e6ed9b722c50ccb6e4
Merge: e821362 76af436
Author: VuTran2601 <75901482+vuTRAN2601@users.noreply.github.com>
Date: Mon Nov 14 17:21:41 2022 +0700

Merge pull request #2 from LMN1590/Vu

Vu
```

Hình 19: Merge Branch phụ vào Branch 'main'

```
commit 438877f033e939b317946a797446f21030874261
Author: nhantran18 <804114510+nhantran18@users.noreply.github.com>
Date: Mon Nov 14 17:43:43 2022 +0700

Delete Report.md
```

Hình 20: Delete file Report.md

```
commit 0b6b62e930c57a0ec2d866a720ad6ef0f18e80bc (HEAD -> master, origin/master, origin/HEAD)
Author: Vinh <khanhvinhbui0512@gmail.com>
Date: Mon Nov 14 19:06:28 2022 +0700

Mot so Activity Diagram
```

Hình 21: Thêm Activity Diagram - Receive Notification, Record Blacklist, Sent and Read Message

**commit 42ea12551045188c36c2b2773036c963b2316dfa**

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Sun Dec 11 16:58:10 2022 +0700

Seem lookable now

**commit cce703e1aca13aa55ada04fec54c073cf30d88bc**

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sun Dec 11 14:07:33 2022 +0700

ticket

**commit 0e0139b0556ee091fbaa7edfa8911b181532d37c**

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sun Dec 11 12:48:55 2022 +0700

update tasklist

**commit a7800cafa9585d172930cbc51e61f6c5c37a8e28**

Merge: afe6403 fc69e28

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sat Dec 10 21:59:54 2022 +0700

update thu

**commit afe6403a5b75ca8ad1c64515de1cd21afa9fd7c3**

Author: khanhvinhbui0512 <khanhvinhbui0512@gmail.com>

Date: Sat Dec 10 21:58:43 2022 +0700

update thu

**commit fc69e28670154aec7a0051a1440d378e75e20410**

Merge: b994ee0 25b2803

Author: VuTran2601 <75901482+vutran2601@users.noreply.github.com>

Date: Sat Dec 10 13:29:58 2022 +0700

Merge pull request #6 from LMN1590/Vu

Updates

**commit 25b2803be2038cbc2a9154adafd2651c1ea9fab8**

Author: VuTran2601 <nguyenvutran2601@gmail.com>

Date: Sat Dec 10 10:10:43 2022 +0700

Updates

**commit b994ee0d3d207a992b9628bcfc08b491c1a827b9**

Merge: 11c1239 07d2d1e

Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>

Date: Thu Dec 8 10:56:04 2022 +0700

Merge pull request #5 from LMN1590/hung

Finish linechart

**commit 07d2d1e1bb4ec43195104bf5a06f23924ab8d190**

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Wed Dec 7 18:58:36 2022 +0700

Finish linechart

**commit 11c1239a23c7ad159dae9c6853238e17cca88b5c**

Merge: a7f0e8c 69e3777

Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>

Date: Tue Dec 6 22:00:18 2022 +0700

Merge pull request #4 from LMN1590/hung

Edit components

**commit 69e3777ffd4df6fecc5ffcf6a879ee0f7a332d2**

Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>

Date: Tue Dec 6 21:58:02 2022 +0700

Edit components



```
commit a7f0e8c708d31c19cf1c1c289041493a44145e85
Merge: 0b6b62e 652c34f
Author: HungNg1012 <104000102+HungNg1012@users.noreply.github.com>
Date: Sun Dec 4 10:29:38 2022 +0700

Merge pull request #3 from LMN1590/hung

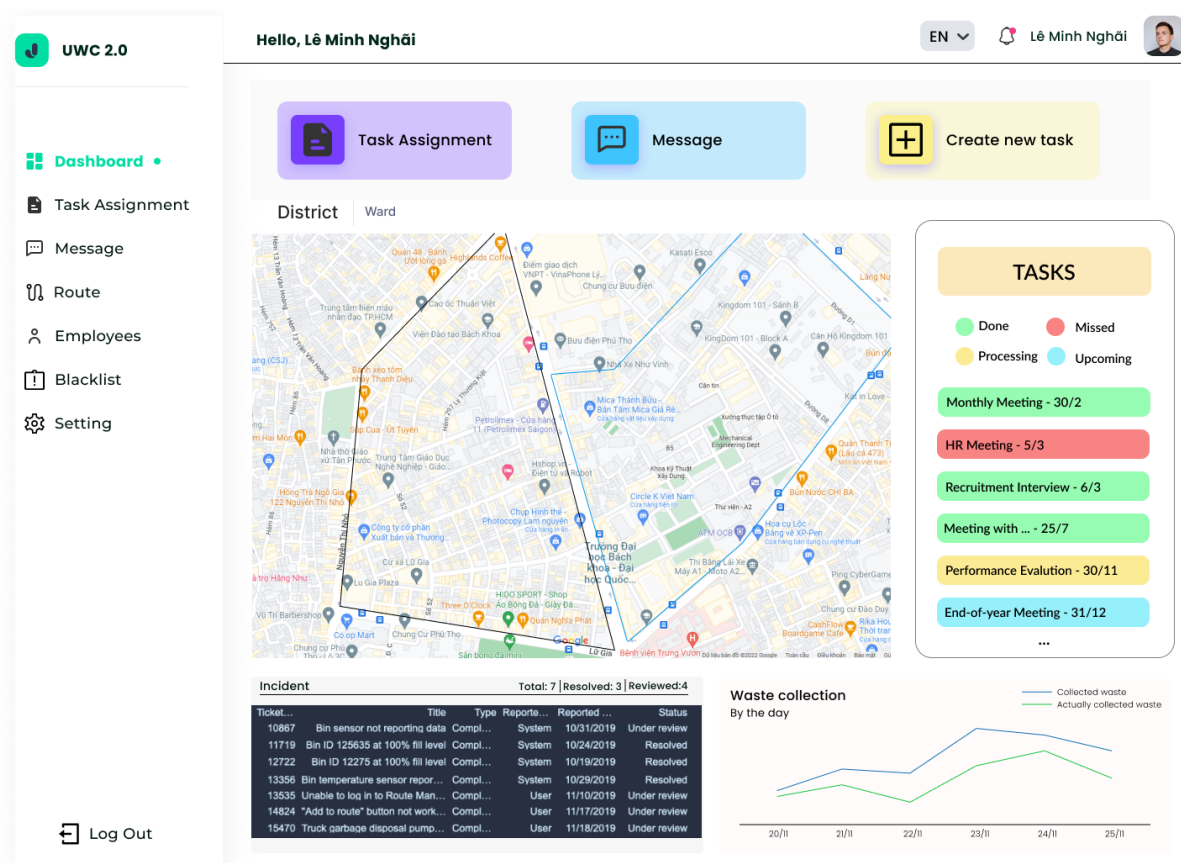
create react app and finish sidebar with header

commit 652c34f3e1741069ade3dfff0418c4d6d7ae19318
Author: HungNg1012 <hung.nguyen1012@hcmut.edu.vn>
Date: Sun Dec 4 10:28:10 2022 +0700

create react app and finish sidebar with header
```

Hình 22: Hiện thực bảng Dashboard

#### 4.3 Implement MVP1 – design an interface 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. Design use a wireframe tool.

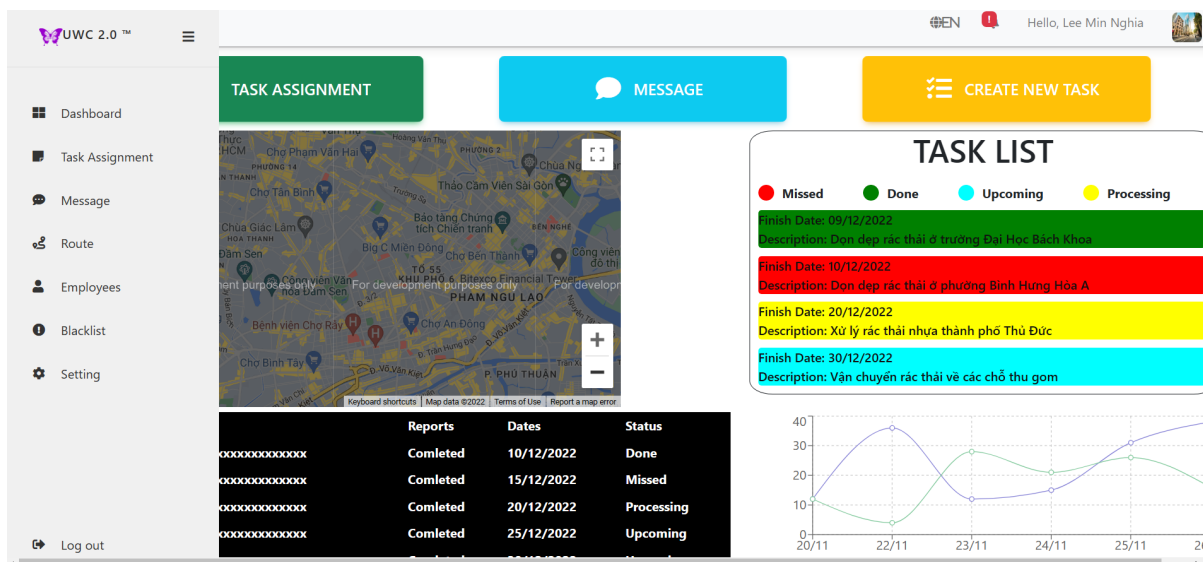


Hình 23: MVP1-design for Back officers in desktop

## 5 Task 5: Implementation – Sprint 2

### 5.1 Implement MVP2 – realize the design in MVP1 with a programming language (HTML, Javascript, Python, C#, etc)

Nhóm hiện thực lại phần design Figma từ task 4.3 thông qua thư viện ReactJS



Hình 24: Dashboard của UWC 2.0 dành cho back officers

### 5.2 Demonstrate the whole project from Task 1 to Task 5

Đây là link drive dẫn đến video thuyết trình của nhóm:

[https://drive.google.com/file/d/16bA68Bmy945\\_zDxyl6pSxt85iUxV1iRz/view?usp=sharing](https://drive.google.com/file/d/16bA68Bmy945_zDxyl6pSxt85iUxV1iRz/view?usp=sharing)

Timestamp:

- 0:00 - Task 1: Bùi Khánh Vĩnh
- 8:21 - Task 2.1: Trần Thiện Nhân
- 24:34 - Task 2.2: Phạm Khai Anh Duy
- 29:40 - Task 2.3: Nguyễn Hữu Danh, Lê Minh Nghĩa
- 36:41 - Task 3.1: Nguyễn Huỳnh Tuấn Hưng
- 43:30 - Task 3.2: Phạm Khai Anh Duy
- 47:50 - Task 4,5: Trần Nguyên Vũ

## 6 Kết luận

Thông qua Bài tập lớn về việc thiết kế một hệ thống quản lý thông tin UWC, hỗ trợ trong quá trình thu gom rác tự động, nhóm chúng em đã học hỏi thêm được rất nhiều về quá trình xây dựng, phát triển một ứng dụng nói chung và ứng dụng UWC nói riêng.

Qua đề tài này, các thành viên trong nhóm đã ứng dụng các kiến thức học hỏi được vào một vấn đề thực tế, qua đầy đủ các công đoạn, từ việc phân tích yêu cầu, lập nên một Use Case Diagram thể hiện các chức năng cần có, sau đó là xây dựng nên các Diagram nền tảng cần thiết cho quá trình phát triển ứng dụng như Sequence Diagram, Activity Diagram, Class Diagram. Ngoài ra, nhóm cũng đã có thêm kinh nghiệm trong việc quản lý và làm việc nhóm.