

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)

Bài tập lớn :

Urban waste collection aid - UWC 2.0

Giáo viên hướng dẫn : Lê Đình Thuận

Lớp : L01

Sinh viên thực hiện: Hồ Ngọc An - 2012548

Lê Văn Lâm - 2013596

Phạm Ngọc Khánh Linh - 2011533

Nguyễn Phi Nam - 2013827

Đỗ Đình Phú Quý - 2014289

Trần Văn Tài - 2014420

Nguyễn Đăng Trường - 2014914

Tp. Hồ Chí Minh , Tháng 09/2022



MỤC LỤC

Nhiệm vụ 1: Tìm hiểu yêu cầu.....	3
1.1 Phân tích tình huống	3
1.1.1 Bối cảnh của dự án.....	3
1.1.2 Các bên liên quan (Stakeholders)	3
1.1.3 Nhu cầu hiện tại của các stakeholders	4
1.1.4 Vấn đề hiện tại của các stakeholders	5
1.1.5 Lợi ích mà UWC 2.0 mang lại.....	5
1.2 Yêu cầu chức năng và phi chức năng.....	6
1.2.1 Yêu cầu chức năng – <i>Functional Requirements</i>	6
1.2.2 Yêu cầu phi chức năng – <i>Non-functional Requirements</i>	7
1.2.3 <i>Use-case Diagram</i>	8
1.3 Mô hình hóa các yêu cầu bằng <i>Usecase Diagram</i>	9
Nhiệm vụ 2: Mô hình hóa hệ thống.....	19
2.1 <i>Activity Diagram</i>	19
2.1.1 <i>Activity Diagram: Assign Task</i>	19
2.1.2 <i>Activity Diagram: View Assign task</i>	21
2.2 <i>Sequence Diagram</i>	23
2.2.1 <i>Khái niệm về giải pháp:</i>	23
2.2.2 <i>Sequence diagram</i>	24
2.3 <i>Class Diagram</i>	26
Nhiệm vụ 3: Thiết kế kiến trúc	28
3.1 Mô tả kiến trúc hệ thống	28
3.1.1 Kiến trúc hệ thống: Mô hình MVC	28
3.1.2 Các Module dự kiến	32
3.2 <i>Component Diagram:</i>	33

Nhiệm vụ 1: Tìm hiểu yêu cầu

1.1 Phân tích tình huống

1.1.1 Bối cảnh của dự án

Ngày nay, với sự phát triển mạnh mẽ của các nền sản xuất công nghiệp lớn trên thế giới đã tạo ra những bước đột phá trong khoa học kỹ thuật, nó khiến cho cuộc sống của con người trở nên thuận tiện và dễ dàng hơn. Thế nhưng đi cùng với điều đó là lượng rác thải được tạo ra cũng rất lớn. Vậy nên việc quản lý rác thải đô thị bây giờ được xem là một trong những vấn đề cấp bách mà nhiều quốc gia cần phải đối mặt.

Điều này cũng được quan tâm như là điểm trọng tâm cần phải cải thiện trong Mục tiêu phát triển bền vững (Sustainable Development Goal) 11: Thành phố và hội đồng bền vững và SDG 6: Nước sạch và vệ sinh môi trường. Đặc biệt chú ý đến các nước đang phát triển trong việc tiếp tục ưu tiên phát triển và tăng trưởng kinh tế. Cải thiện việc quản lý và thu gom rác thải được nhấn mạnh bởi chính phủ và các tổ chức cho các tác động tích cực lên thành phố, xã hội và môi trường.

1.1.2 Các bên liên quan (Stakeholders)

- Định nghĩa : Stakeholders hay các bên liên quan là thuật ngữ dùng để chỉ cá nhân, nhóm người hoặc tổ chức có quan hệ mật thiết với doanh nghiệp, quan tâm đến hoạt động và sự thành công của dự án.

- Các bên liên quan được xác định trong dự án này là :

- **Tổ chức quản lý rác thải:** Đầu tư dự án, chịu trách nhiệm cho hoạt động thu gom rác thải đô thị.

- **Nhân viên lái xe (Collectors):** Điều khiển các phương tiện giao thông khác nhau đến những nơi thu gom rác thải (**MCPs**) để thu gom rác thải.
- **Nhân viên thu gom rác (Janitors):** Sử dụng các trollers để thu gom rác từ các khu vực được giao đến MCPs.
- **Cán bộ phụ trách (Back officers):** hoạt động ở hệ thống trung tâm để tạo lên lịch trình, điều phối collectors và janitors.

1.1.3 Nhu cầu hiện tại của các stakeholders

- **Tổ chức quản lý rác thải:** Cần ứng dụng UWC 2.0 xử lý dữ liệu real-time ít nhất 1000 MCPs, dễ dàng tiếp cận và sử dụng cho nhân viên và các chức năng phục vụ đúng nhu cầu công việc quản lý rác thải. Từ đó tối ưu hoá công việc thu gom rác thải và đem về lợi nhuận cao.

- **Back officers (cán bộ quản lý):** Cần một hệ thống để có thể quản lý thời gian biểu và lịch trình làm việc của người thu gom rác và người lái xe. Bên cạnh đó họ còn có nhu cầu nhắn tin với nhân viên, phân công công cụ hợp lý và chọn tuyến đường phù hợp để tối ưu nhiên liệu.

- **Collectors (lái xe):** Phải nắm rõ được thời gian biểu và lịch trình làm việc của mình, biết được địa điểm tập kết rác và cần phải điểm danh mỗi ngày làm việc.

- **Janitors (nhân viên thu gom rác):** Cần phải biết được thời gian biểu, lịch trình làm việc mỗi ngày. Nắm được địa điểm thu gom rác (MCPs) để tập kết rác. Bên cạnh đó hệ thống cần phải đáp ứng nhu cầu giao tiếp của các nhân viên thu gom để thông báo về độ đầy của MCPs và chức năng điểm danh mỗi ngày.

1.1.4 Vấn đề hiện tại của các stakeholders

- **Tổ chức quản lý rác thải:** ứng dụng UWC 1.0 chưa đáp ứng một số chức năng cần thiết trong công việc, lợi nhuận chưa cao.

- **Back officers (cán bộ quản lý):** Hệ thống hiện tại chưa hỗ trợ đầy đủ các chức năng, tạo nên sự khó khăn trong khâu quản lý thời gian và lịch làm việc của các nhân viên. Chưa hỗ trợ được việc giao tiếp trong hệ thống và nắm được về mức độ đầy của các MCPs. Bên cạnh đó việc sắp xếp chọn lựa quãng đường đi cũng chưa hợp lý, gây lãng phí nhiên liệu.

- **Collectors (lái xe):** Gặp bất tiện trong việc theo dõi lịch làm việc hàng ngày. Tuyến đường đi chưa được sắp xếp hợp lý gây nên nhiều bất cập như tốn nhiên liệu, tắc nghẽn giao thông,... Giao diện của hệ thống chưa thân thiện, chưa đáp ứng được việc giao tiếp, điểm danh.

- **Janitors (nhân viên thu gom rác):** Chưa có phương thức giao tiếp rõ ràng, lịch trình của các nhân viên chưa được phân công hợp lý. Chưa nắm được cụ thể và kịp thời độ đầy của các MCPs.

1.1.5 Lợi ích mà UWC 2.0 mang lại

- **Tổ chức quản lý rác thải:** Giúp cho quy trình quản lý rác thải của nhà cung cấp hoạt động trơn tru và có hiệu suất tốt hơn đem về lợi nhuận cao cho tổ chức.

- **Back officers (cán bộ quản lý):** Hệ thống sẽ giúp cho việc quản lý và sắp xếp công việc cho nhân viên của các Back officers trở nên dễ dàng và thuận tiện hơn. Sự giao tiếp giữa các nhân viên trong hệ thống cũng được cải thiện. Các Back officers có thể nắm kịp thời về độ đầy của các MCPs để có thể sắp xếp hợp lý.

- **Collectors (lái xe):** Giúp cho các Collectors có một giao diện thân thiện để điểm danh, kiểm tra thời khóa biểu của mình. Các MCPs đây sẽ được thông báo một cách nhanh chóng để các Collectors kịp thời thu gom. Bên cạnh đó các tuyến đường cũng được chọn lựa một cách tối ưu nhất giúp cho tình trạng tắc đường hạn chế xảy ra và tiết kiệm được nhiên liệu.

- **Janitors (nhân viên thu gom rác):** UWC 2.0 sẽ hỗ trợ các nhân viên thu gom rác một hệ thống thân thiện dễ dùng để cập nhật kịp thời lịch trình làm việc. Nắm được nhanh chóng độ đầy của các MCPs, giao tiếp dễ dàng giữa các nhân viên trong tổ chức và còn hỗ trợ điểm danh mỗi ngày.

1.2 Yêu cầu chức năng và phi chức năng

1.2.1 Yêu cầu chức năng – *Functional Requirements*

* *Back officers:*

- Đăng nhập vào hệ thống để làm việc, đăng xuất khỏi hệ thống.
- Quản lý thông tin của janitors và collectors.
- Quản lý thông tin của các phương tiện (vehicles).
- Quản lý thông tin của các MCPs.
- Giao phương tiện cho janitors và collectors.
- Giao nhiệm vụ đến các MCPs cho janitors và collectors.
- Tạo tuyến đường cho mỗi collectors, tuyến đường được tối ưu hóa về mặt nhiên liệu và quãng đường đi.
- Nhắn tin với các back officers, janitors và collectors khác.

* *Janitors, Collectors:*

- Quản lý tài khoản của họ trong hệ thống (Đăng nhập, đăng xuất, cập nhật thông tin).

- Kiểm tra thông tin về lịch làm việc.
- Kiểm tra thông tin tác vụ công việc hàng ngày và hàng tuần.
- Nhắn tin với các collectors, janitors và back officers khác.
- Kiểm tra/ xác nhận hoàn thành công việc hàng ngày.
- Nhận thông báo về các MCPs đã chứa đầy.

*** UWC 2.0 system:**

- Thông báo về các MCPs đã chứa đầy cho Collectors.

1.2.2 Yêu cầu phi chức năng – Non-functional Requirements

***Yêu cầu về hiệu suất, tốc độ:**

- Nhắn tin giữa các thành viên với công nghệ thời gian thực cùng độ trễ dưới 1 giây.
- Thông tin về MCPs được cập nhật sau mỗi 15 phút và khả dụng trong ít nhất 95% thời gian hoạt động.
- Hệ thống sẽ có thể xử lý dữ liệu theo thời gian thực ít nhất 1000 MCP tại một thời điểm.
- Phản hồi của hệ thống đối với các thao tác của người dùng không được quá 1 giây.

*** Yêu cầu về tính dễ sử dụng**

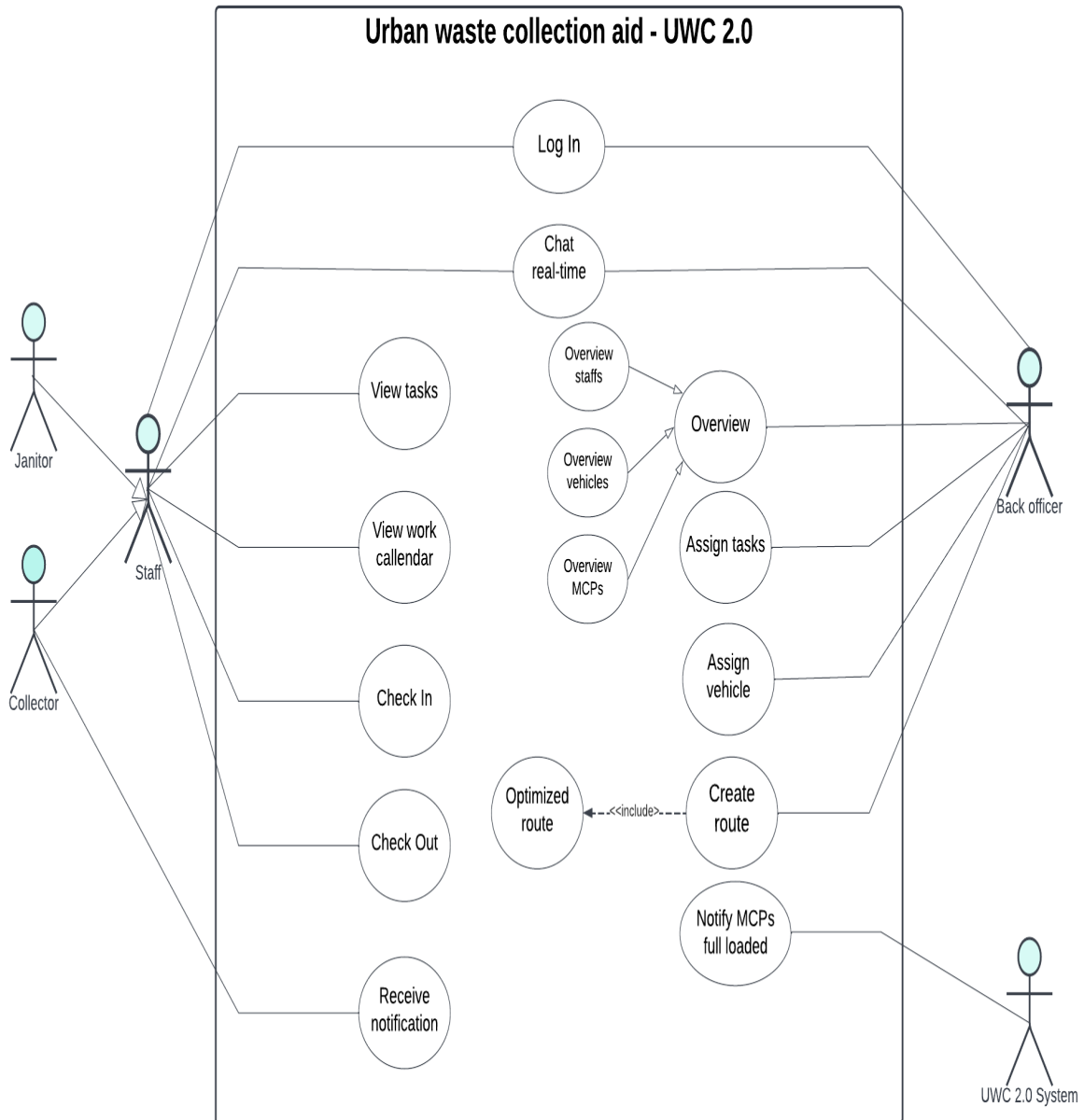
- Tất cả thông tin quan trọng về lịch trình công việc được hiển thị trong một trang (không cần cuộn trang).
- Các nhân viên thành thạo sử dụng hệ thống sau 1 ngày training.

***Yêu cầu về tính mới rộng**

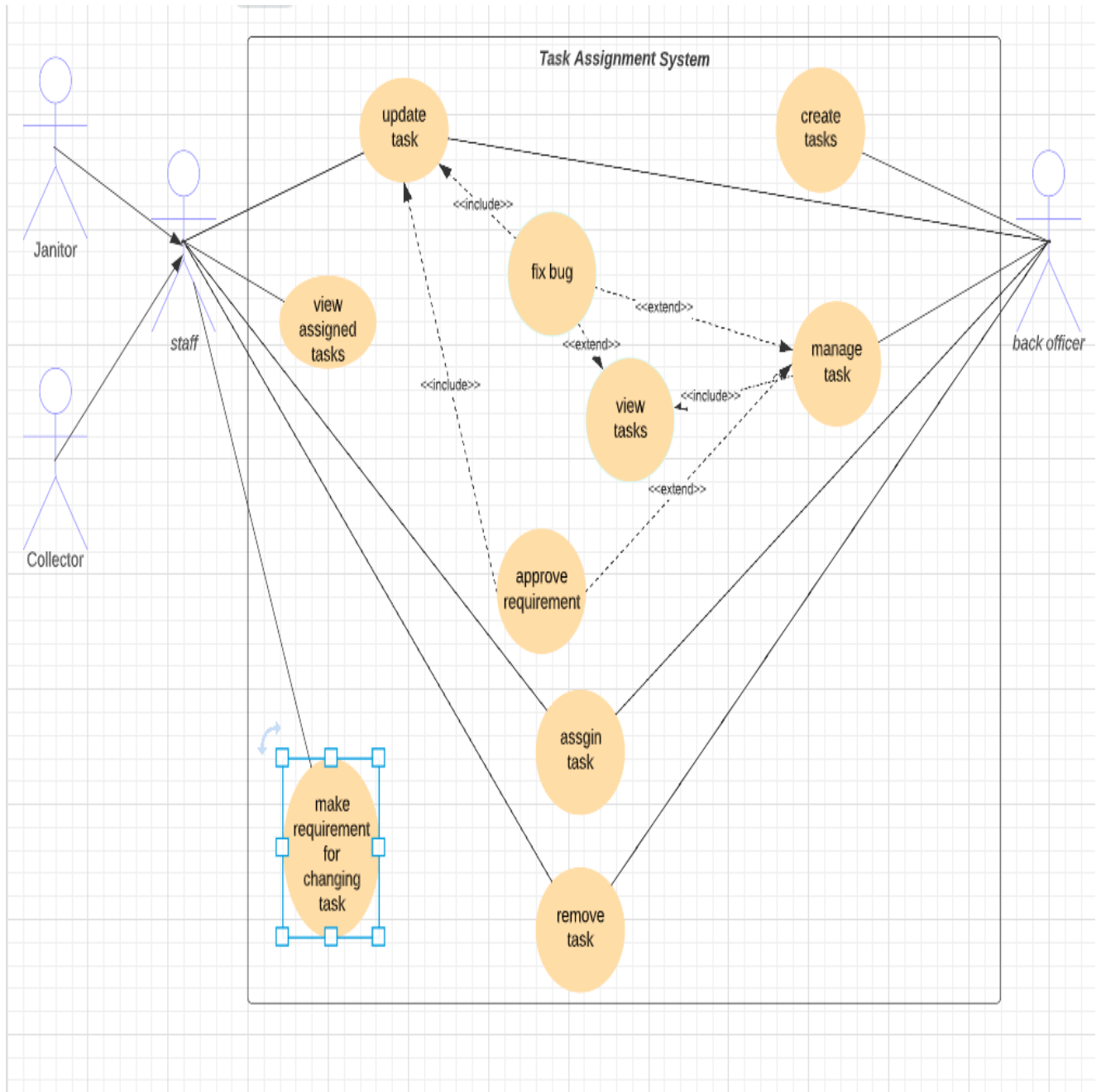
- Hệ thống có thể thêm giao diện ngôn ngữ Tiếng Anh trong tương lai.

- Hệ thống nâng cấp sẽ có thể xử lý dữ liệu theo thời gian thực ít nhất 10000 MCPs.

1.2.3 Use-case Diagram



1.3 Mô hình hóa các yêu cầu bằng Usecase Diagram



❖ Describe the use-case using a table format:

➤ Manage task:

Use-case name	Manage task
Actor	Back officer
Description	Quản lý các tasks của hệ thống
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Manage tasks trên trang chủ
Postconditions	Không
Normal flow	<ol style="list-style-type: none">1. Từ giao diện chính chọn manage task2. Officer chọn các option sau:<ul style="list-style-type: none">● Fix bug● Approve requirement● View tasks3. Hệ thống chuyển sang giao diện của Option đã chọn
Alternative Flow	Tại bước 2, Officer có thể chọn “Back” để quay về trang chủ
Exceptions	Không

➤ Create task

Use-case name	Create task
----------------------	-------------

Actor	Back officer
Description	Tạo task mới vào trong hệ thống
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Create tasks trên trang chủ
Postconditions	1.Task được tạo thành công 2.Thông tin của task được đưa lên bảng phân công nhiệm vụ
Normal flow	1. Hệ thống hiển thị bảng phân công nhiệm vụ 2. Officer tạo task 3. Điền thông tin về task (vd: tuyến đường hoạt động trong ngày, vệ sinh công cụ lao động sau giờ làm,...) 4. Sau khi điền đầy đủ thông tin, Officer nhấn nút “Thêm” 5. Hệ thống hiển thị thông báo “Bạn có chắc muốn thêm task này vào hệ thống?” 6. Officer chọn nút “OK” để xác nhận 7. Hệ thống trở về giao diện chính
Alternative Flow	Tại bước 6: Nếu Back Officers không muốn Create task này nữa thì nhấn nút “Hủy” để trở về giao diện ở bước 1
Exceptions	Không

➤ **Remove task**

Use-case name	Remove task
Actor	Back officer
Description	Xóa task ra khỏi hệ thống
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Remove task trên trang chủ
Postconditions	Task được xóa thành công
Normal flow	<ol style="list-style-type: none">1. Hệ thống hiển thị bản phân công nhiệm vụ2. Officer chọn 1 task muốn xóa3. Nhấn nút “Xóa”4. Hệ thống hiển thị thông báo “Bạn có chắc muốn xóa task này ra khỏi hệ thống?”5. Officer chọn nút “OK” để xác nhận6. Hệ thống trở về giao diện chính
Alternative Flow	Tại bước 3: Nếu Back Officers không muốn xóa task này nữa thì nhấn nút “Hủy” để trở về giao diện ở bước 1
Exceptions	không

➤ **Update task**

Use-case name	Update task
Actor	Back officer
Description	Back officer muốn cập nhật thông tin của task



Preconditions	đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Update task trên trang chủ
Postconditions	Task được update thành công
Normal flow	<ol style="list-style-type: none">1. Hệ thống chuyển sang giao diện hiển thị đầy đủ các task đã được phân công.2. Officer chỉnh sửa những thông tin cần thay đổi.3. Sau khi chỉnh sửa, nhấn nút “Lưu” (hoặc “Cập Nhật”)4. Hệ thống trở về giao diện chính
Alternative Flow	Tại bước 3: Nếu Back Officers không muốn update task này nữa thì nhấn nút “Hủy” để trở về giao diện ở bước 1
Exceptions	không

➤ **Assign task:**

Use-case name	Assign task
Actor	Back officer
Description	Phân công các task cho Collectors và Janitor
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Assign task trên trang chủ
Postconditions	<ol style="list-style-type: none">1. Phân công nhiệm vụ thành công2. Staffs có thể xem được các task mà mình được giao

Normal flow	<ol style="list-style-type: none">1. Hệ thống chuyển sang giao diện hiển thị đầy đủ các task2. Điền thông tin người được phân công vào trong task3. Sau khi điền thông tin, nhấn nút “Lưu” (hoặc “Cập nhật”)4. Hệ thống trở về giao diện chính
Alternative Flow	<ol style="list-style-type: none">1. Tại bước 3: nếu Back Officers không muốn assign task này nữa thì nhấn nút “hủy” để trở về giao diện ở bước 12. Tại bước 3: sau khi nhấn nút “Lưu” mà bị Exceptions thì có thể nhấn nút “Quay lại” để chỉnh sửa task
Exceptions	Hệ thống hiển thị task không thể được phân công (do trùng thời gian, ngày không làm việc, ...)

➤ **Fix bug**

Use-case name	Fix bug
Actor	Back officer
Description	Back officer muốn sửa các lỗi liên quan đến task
Preconditions	Đăng nhập vào hệ thống, đang ở trang chủ hoặc ở trang View tasks
Trigger	Nhấn vào thanh Fix bug
Postconditions	Các bugs của hệ thống đã được fix
Normal flow	<ol style="list-style-type: none">1. Từ giao diện chính, chọn manage task



	<p>2. Sau đó Back officer chọn option fix bug (hoặc chọn view tasks để xem cái task, khi thấy lỗi thì chọn fix bug)</p> <p>3. Hệ thống sẽ chuyển sang giao diện fix bug đã được chọn</p> <p>4. Officer hoàn tất hành động fix bug các task bằng cách nhấn nút “Fix bug”</p> <p>5. Hệ thống trở về giao diện chính</p>
Alternative flow	Tại bước 4, nếu các lỗi không được fix, nhấn nút chọn “Liên hệ đội kỹ thuật để xử lý”
Exceptions	Trường hợp không fix được bug thủ công, liên hệ nhân viên sửa chuyên nghiệp

➤ **Approve requirement:**

Use-case name	Approve requirement
Actor	Back officer
Description	Chấp nhận các yêu cầu thay đổi task cho Collectors và Janitor
Preconditions	Đăng nhập vào hệ thống, đang ở trang chủ hoặc ở Manage task ,có yêu cầu thay đổi
Trigger	Nhấn vào thanh Approve requirement
Postconditions	Các requirement của staff sẽ được kiểm duyệt, được chấp nhận hoặc từ chối
Normal flow	<p>1. Từ giao diện chính, chọn manage task</p> <p>2. Sau đó Back officer chọn option approve</p>

	<p>requirement</p> <p>3. Hệ thống sẽ chuyển sang giao diện approve requirement đã được chọn</p> <p>4. Officer thực hiện hành động approve các task</p> <p>5. Sau khi chọn những task hợp lệ, Officer chọn nút “OK” để xác nhận</p> <p>6. Hệ thống trở về giao diện chính</p>
Alternative flow	<p>Tại bước 4, chọn “Back” để quay lại Manage tasks nếu không muốn thực hiện thao tác Approve nữa</p> <p>Tại bước 5, nếu Officer không muốn lưu các thao tác của mình chọn “Cancel” để hủy vào chuyển về bước 3</p>
Exceptions	<p>Hệ thống hiển thị task không thể được đổi được task cho staff (do trùng thời gian, ngày đầy lịch, quá số lần đổi trong 1 tháng, ...)</p>

➤ **View tasks**

Use-case name	View tasks
Actor	Back officer
Description	Nhìn thấy toàn bộ các task giao cho staff
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh View tasks trên trang
Postconditions	Có thể xem được bản phân công nhiệm vụ
Normal flow	1. Từ giao diện chính, chọn manage task



	2. Sau đó Back officer chọn option view task 3. Hệ thống sẽ chuyển sang giao diện view tasks đã được chọn 4. Officer hoàn tất hành động view tasks 5. Hệ thống trở về giao diện chính
Alternative flow	Không
Exceptions	không

➤ **View assigned tasks:**

Use-case name	View assigned tasks
Actor	Staff
Description	Staff muốn xem các task mà mình đã được giao
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh View tasks trên trang chủ
Postconditions	Có thể xem được bản phân công nhiệm vụ
Normal flow	1. Từ giao diện chính, chọn manage task 2. Sau đó Staff chọn option view tasks 3. Hệ thống sẽ chuyển sang giao diện view tasks đã được chọn 4. Staff hoàn tất hành động view các task 5. Hệ thống trở về giao diện chính



Alternative flow	Không
Exceptions	không

➤ **Make requirement for changing task:**

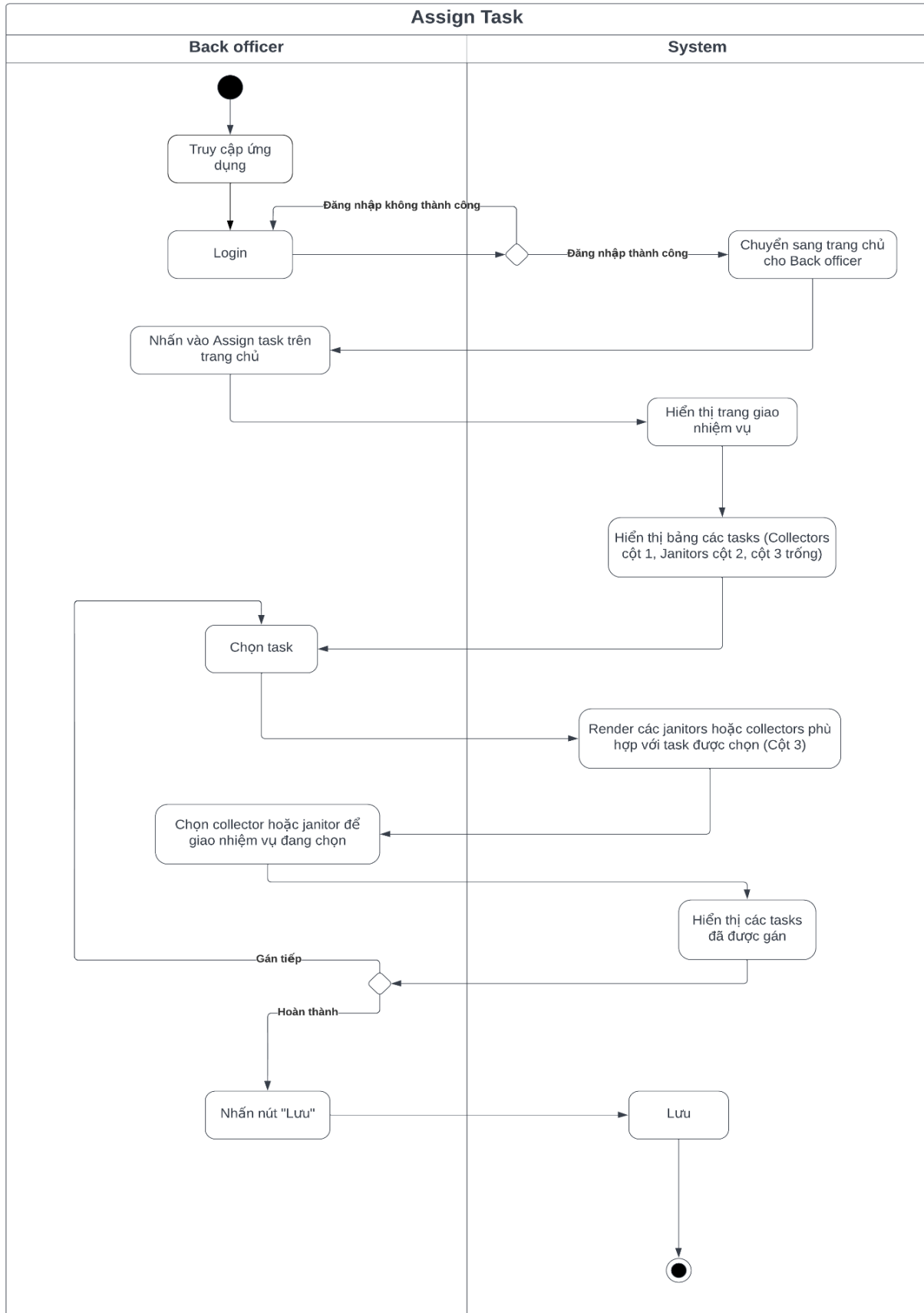
Use-case name	Make requirement for changing task
Actor	Staff
Description	Staff muốn đổi task của mình
Preconditions	Đăng nhập vào hệ thống
Trigger	Nhấn vào thanh Make requirement task trên trang
Postcondition	Các requirement của staff được gửi đến hệ thống
Normal flow	<ol style="list-style-type: none">1. Từ giao diện chính, chọn manage task2. Sau đó Staff chọn option changing task3. Hệ thống sẽ chuyển sang giao diện changing task đã được chọn4. Staff hoàn tất hành động changing các task5. Hệ thống trở về giao diện chính
Alternative flow	<p>Tại bước 4, chọn “Back” để quay lại trang trước vào hủy việc tạo requirement</p> <p>Tại bước 6, chọn “Cancel” để hủy các thao tác quay lại bước 3</p>
Exceptions	Hệ thống không còn lịch để thay đổi.



Nhiệm vụ 2: Mô hình hóa hệ thống

2.1 Activity Diagram

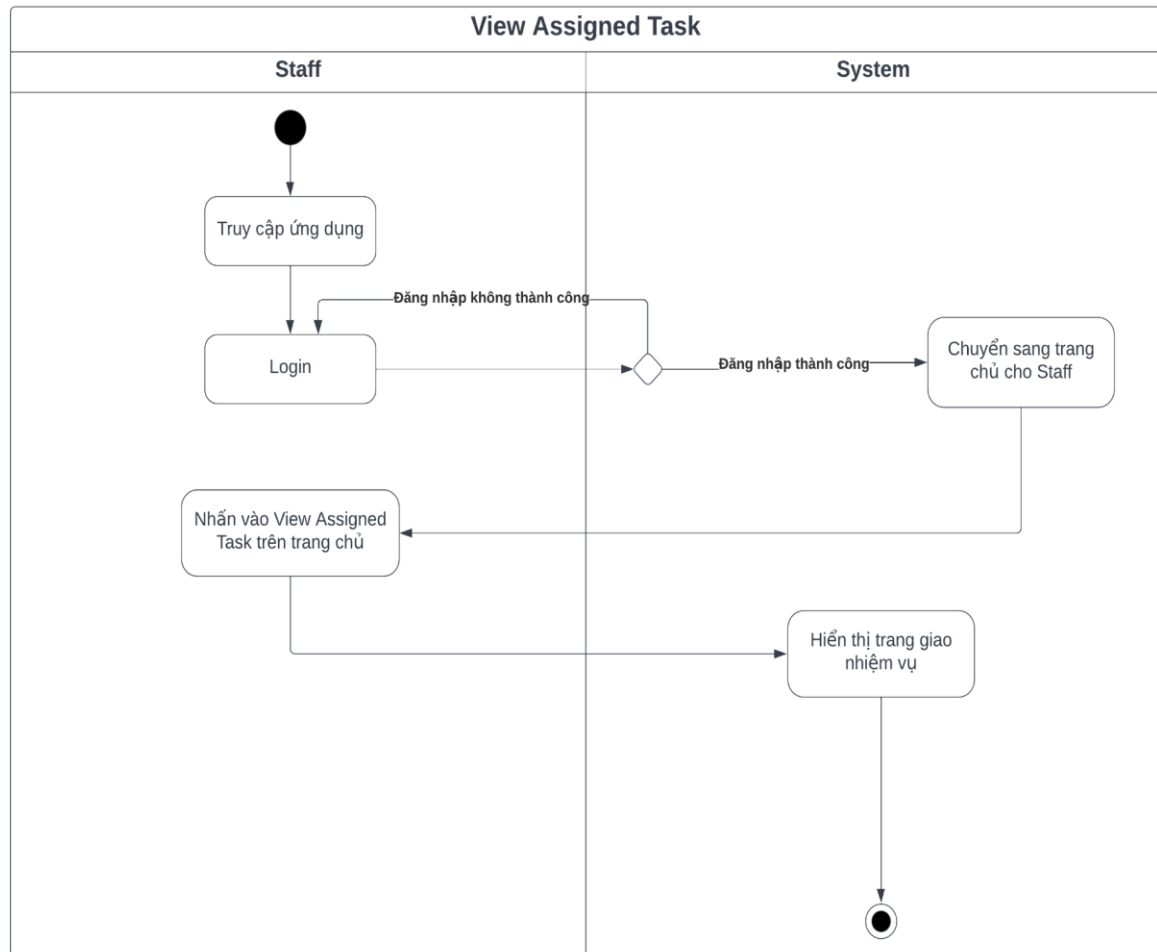
2.1.1 Activity Diagram: Assign Task



Mô tả Diagram:

- Back officer truy cập ứng dụng sau đó đăng nhập vào hệ thống bằng tài khoản đã đăng ký, hệ thống sẽ mở ra giao diện của trang chủ sau khi đăng nhập thành công.
- Sau đó, User bấm vào nút “Assign Task” trên thanh công cụ để mở trang giao nhiệm vụ. Tại đây, hệ thống sẽ hiện ra bảng gồm ba cột, cột 1 là task dành cho collector, cột 2 là task dành cho janitor và cột còn lại còn trống dùng để hiển thị danh sách các collector hoặc janitor phù hợp với task được chọn.
- User chọn Task muốn phân công, hệ thống sẽ hiển thị ra danh sách các collector hoặc janitor phù hợp với task được chọn.
- User chọn collector hoặc janitor muốn gán cho task này hệ thống sẽ hiển thị ra task đã được gán ở bên dưới bảng nhiệm vụ.
- User bấm nút lưu để lưu các task đã gán xong và kết thúc.

2.1.2 Activity Diagram: View Assign task

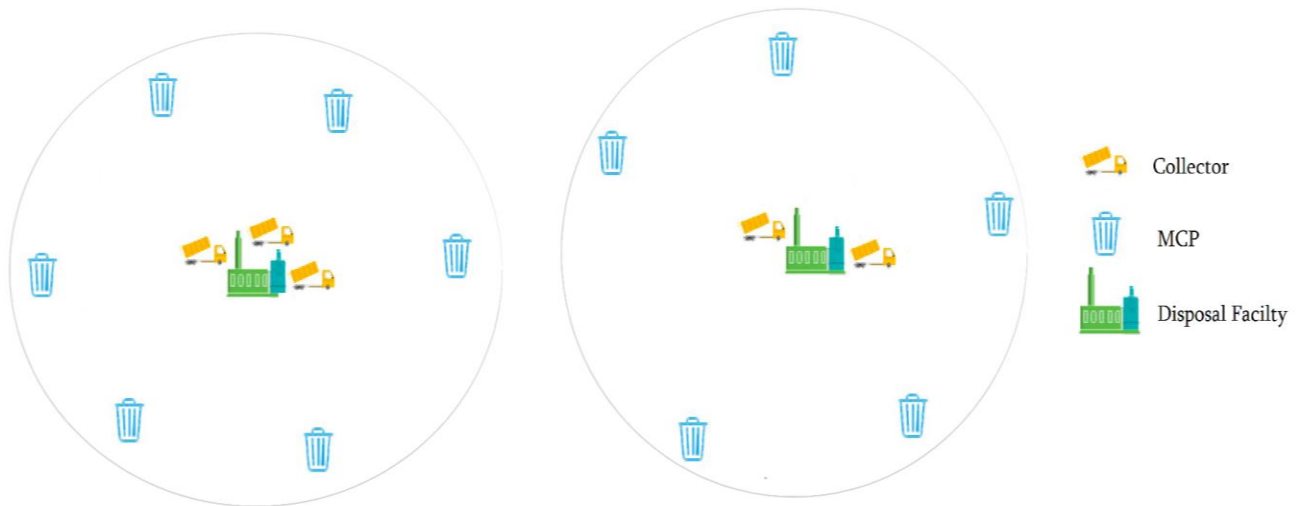


Mô tả Diagram:

- Collector hoặc Janitor truy cập ứng dụng sau đó đăng nhập vào hệ thống bằng tài khoản đã đăng ký, hệ thống sẽ mở ra giao diện của trang chủ sau khi đăng nhập thành công.
- Sau đó, user bấm vào nút “View Assign Task” trên thanh công cụ, hệ thống sau đó sẽ hiện ra các task cần phải hoàn thành. User có thể xem thông tin của các task này như deadline, MCPs, equipment,...

2.2 Sequence Diagram

2.2.1 Khái niệm về giải pháp:



*Giả thiết về hệ thống:

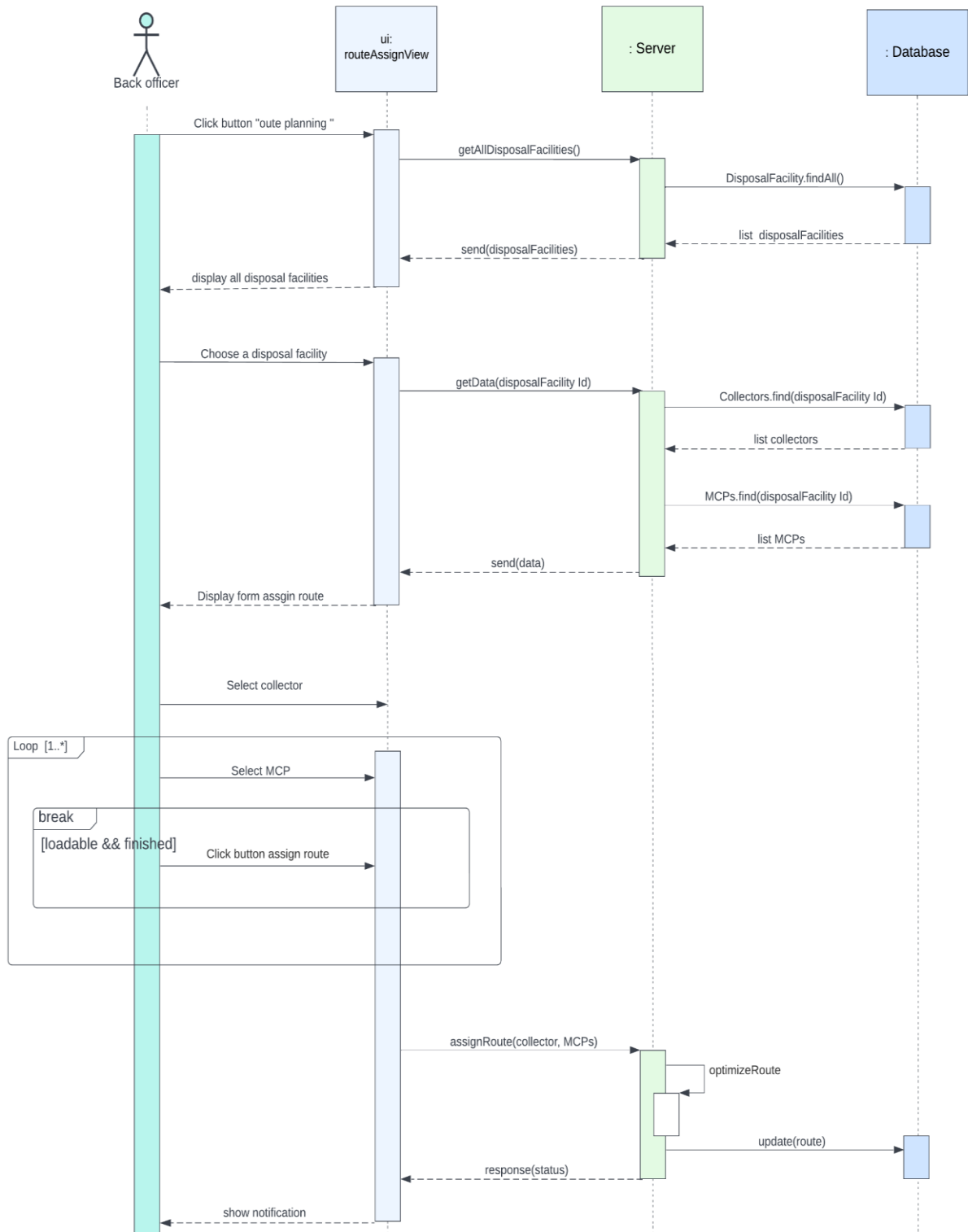
- Mỗi **Disposal facility** (cơ sở xử lý rác thải) sẽ xử lý rác từ các **MCPs** trong một khu vực nhất định, mỗi **MCPs** sẽ thuộc quản lý của duy nhất 1 **Disposal facility**.
- Mỗi **Collector** sẽ được phân công điều khiển 1 phương tiện vận chuyển và làm tại một **Disposal facility** (theo Task Assignment). Các phương tiện lưu trú ở các bãi xe ngay tại các **Disposal facilities**.
- Khi lập kế hoạch tuyến đường, **Back officer** sẽ tiến hành gán tuyến đường cho một **collector** bao gồm các **MCPs** cần được thu gom thuộc quyền quản lý của **Disposal facility** đó.
- Việc lựa chọn các **MCPs** cần thỏa mãn điều kiện không tổng lượng rác xử lý không vượt quá trọng tải xe thu gom.
- Tuyến đường được lên kế hoạch sẽ bắt đầu từ **Disposal facility** mà **Collector** làm việc và phải đi qua tất cả các điểm **MCPs** đã phân công, cuối cùng quay

về lại *Disposal facility*. Tuyến đường qua các điểm *MCPs* cần được tối ưu về khoảng cách và nhiên liệu.

***Giả thiết về tối ưu đường đi:**

- Ta sẽ tìm ra tất cả các chu trình có thể có bằng việc sử dụng thuật toán quay lui, sau đó chọn ra các chu trình chứa các điểm yêu cầu và từ đó lựa ra chu trình có đường đi ngắn nhất.

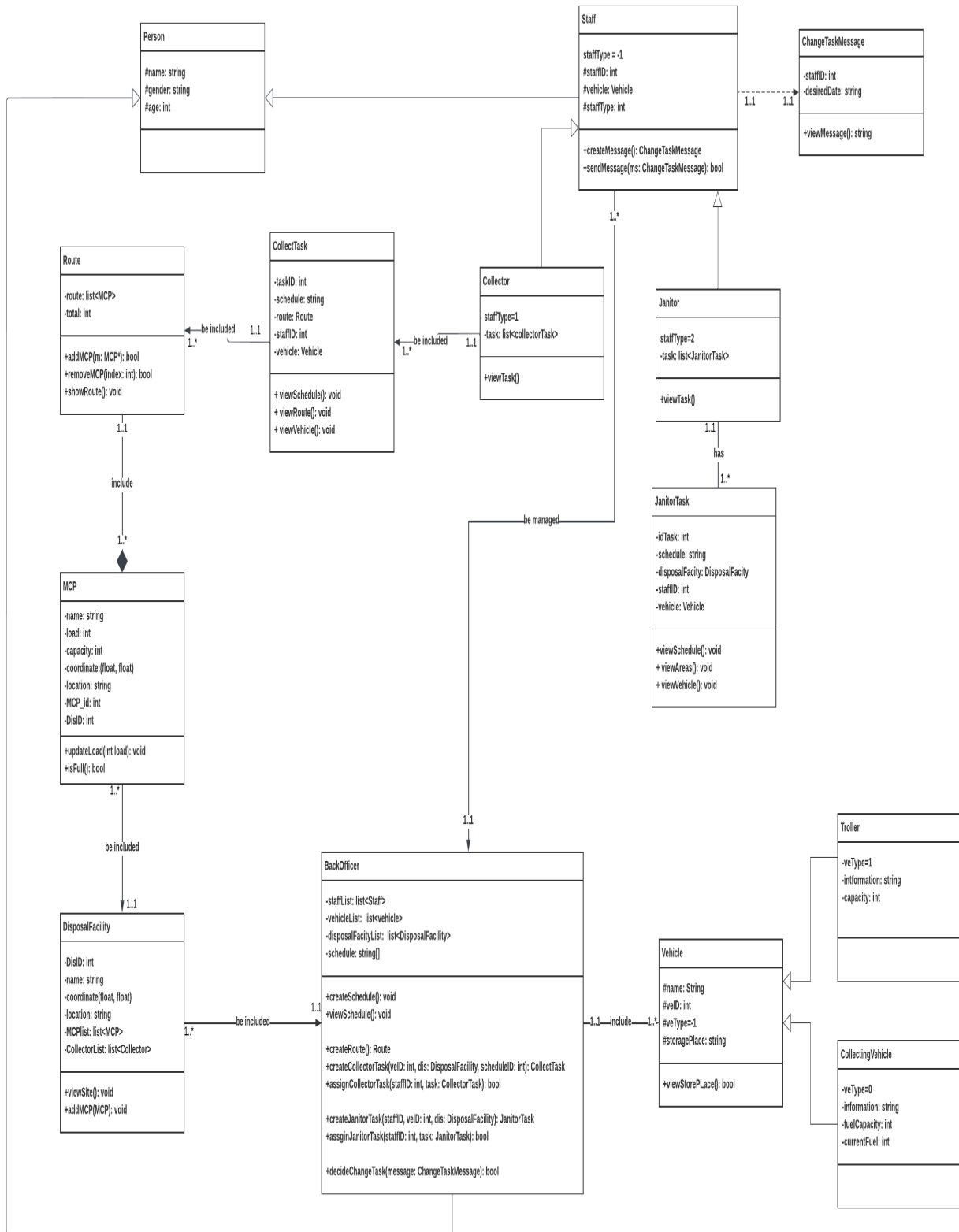
2.2.2 Sequence diagram



**Mô tả về Diagram:*

- Tại giao diện chính của **Back officer**, khi chọn tác vụ lập kế hoạch tuyến đường, hệ thống tiến hành truy xuất dữ liệu trả về giao diện một danh sách **các Disposal facilities** đang được quản lý.
- Hệ thống yêu cầu **Back officer** chọn một trong số các **Disposal facilities** và tiến hành truy xuất dữ liệu lấy các thông tin về **Collectors** và **Mcps** của **Disposal facility** đó và gửi dữ liệu trả lại giao diện một form cho tác vụ lập kế hoạch tuyến đường.
- **Back officer** thực hiện chọn một collector và nhiều **MCPs** và gửi dữ liệu lên hệ thống. Việc chọn các MCPs phải thoả mãn điều kiện tổng trọng rác cần xử lý ở các MCPs đó phải phù hợp với trọng tải phương tiện vận chuyển của **Collector** đó, nếu không **Back officer** cần thực hiện chọn lại.
- Hệ thống xử lý các thông tin để tìm tuyến đường tốt nhất và cập nhật dữ liệu vào hệ thống.
- Cuối cùng, hệ thống hiển thị thông báo đã tạo tuyến đường thành công.

2.3 Class Diagram



Link Class Diagram:

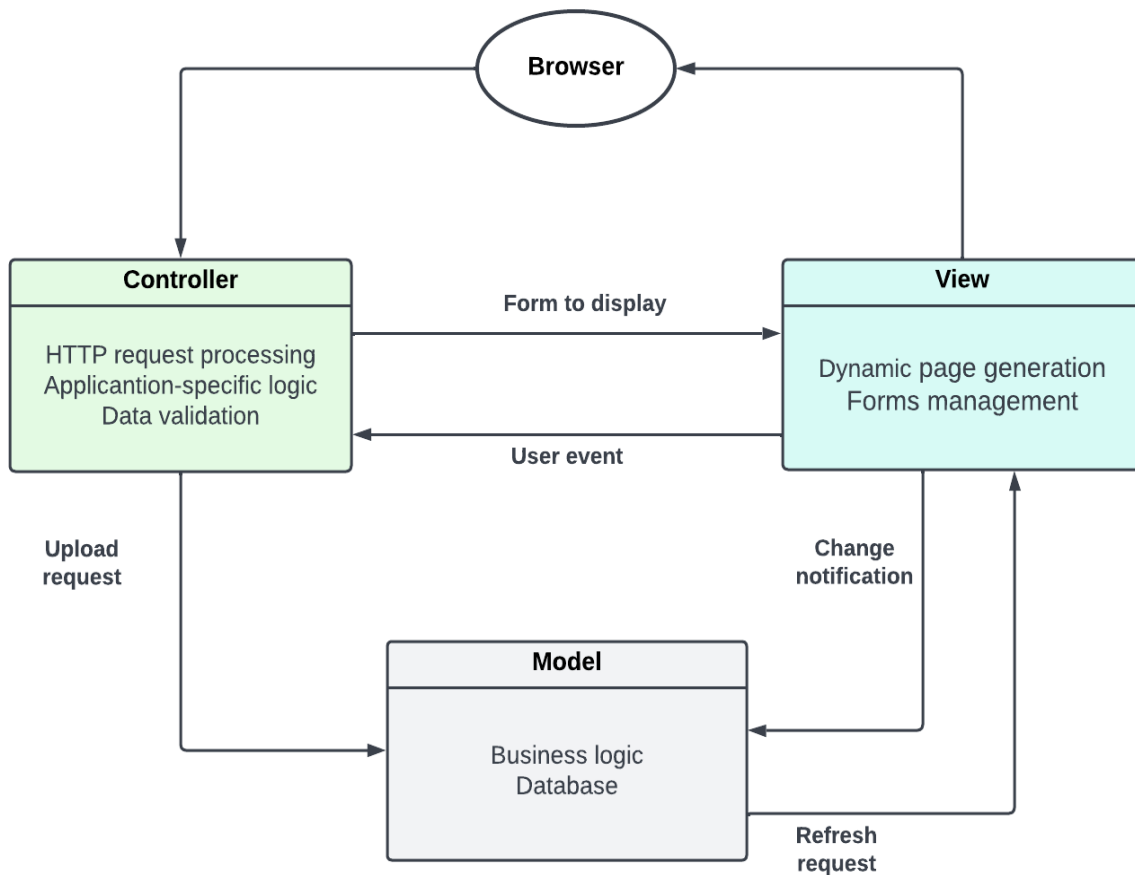
- <https://lucid.app/publicSegments/view/afa889e7-0aed-4a5e-8891-666ce7c256ba>

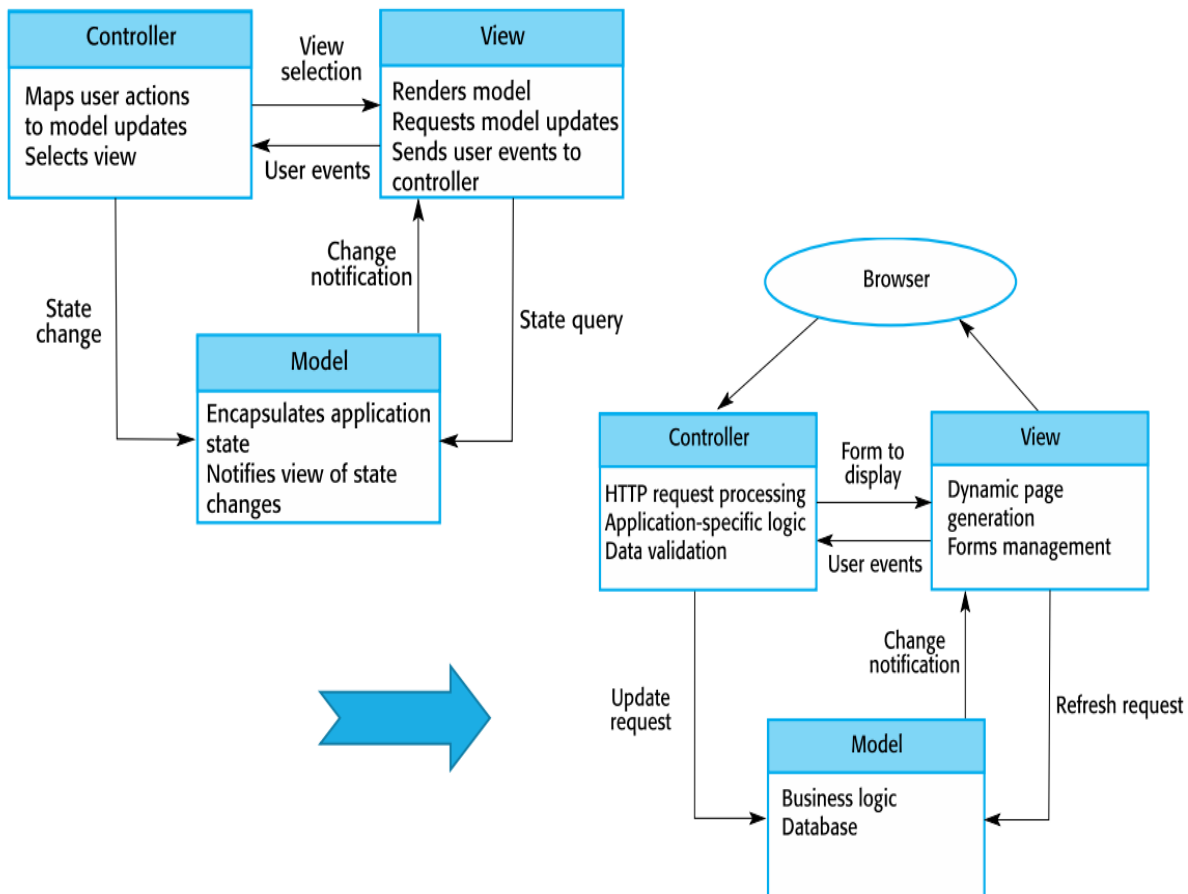
Nhiệm vụ 3: Thiết kế kiến trúc

3.1 Mô tả kiến trúc hệ thống

3.1.1 Kiến trúc hệ thống: Mô hình MVC

Mô hình MVC





Mô hình MVC là một mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Mô hình source code được chia thành 3 phần tương ứng **Model – View – Controller**.

***View:** giữ các vai trò

- Render các UI components cho người dùng:

- + Render ra form đăng nhập, đăng ký, thanh điều hướng, button,... của trang web.
- + Render ra các bảng phân công nhiệm vụ (Task), lịch trình làm việc (Calendar),... cho Janitors và Collectors.

+ Render ra bảng quản lý nhiệm vụ (Task Assignment) cho các Back Officers.

+ Tạo các hiệu ứng có nhiệm vụ tương tác với người sử dụng.

- Tương tác và gửi yêu cầu cập nhật dữ liệu tới Controller.

+ Gửi thông tin đăng nhập, đăng kí.

+ Gửi thông tin các nhiệm vụ được cập nhật.

+ Gửi các sự kiện Click, Submit form,... của người dùng.

***Controller:** giữ các vai trò

- Bao gồm những Class/ Function có khả năng xử lý nhiều nghiệp vụ Logic

+ Xác thực người dùng khi gửi yêu cầu lên hệ thống.

+ Tìm đường đi tối ưu giữa các MPCs.

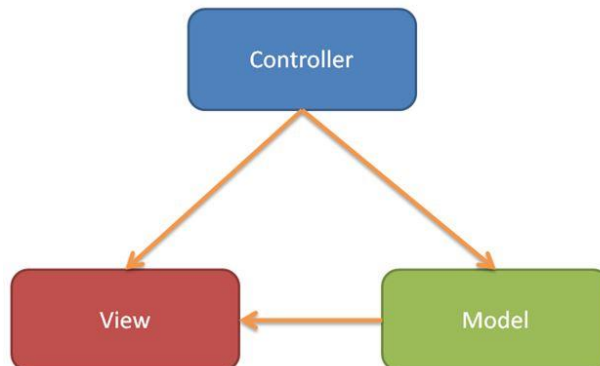
- Xử lý các yêu cầu người dùng đưa đến thông qua View

+ Lấy thông tin về lịch phân công nhiệm vụ.

+ Xem chi tiết từng task.

+ Tạo, sửa, xóa các task.

- Đóng vai trò trung gian giữa Model và View, Controller gửi lệnh đến Model cập nhật dữ liệu, lệnh sẽ tiếp tục được truyền đến View để cập nhật giao diện hiển thị cho đúng với dữ liệu đã cập nhật trong Model cho người dùng thấy.



+ **Tương tác Controller - View:** lấy những hình ảnh, nút bấm, hay hiển thị những dữ liệu được trả ra từ Controller để giúp cho người dùng có thể quan sát và thao tác dễ dàng. Sự tương tác này cũng có thể có những dữ liệu không được lấy trực tiếp từ Model, lúc này chỉ có nhiệm vụ hiển thị đơn thuần như những hình ảnh và nút bấm mà thôi.

+ **Tương tác Controller - Model:** là luồng xử lý khi Controller tiếp nhận những yêu cầu và tham số đầu vào ở người dùng. Controller lúc này sẽ sử dụng những class / function có trong model để lấy ra những dữ liệu chính xác nhất mà người dùng cần.

- Hai thành phần **View - Model** có thể tương tác với nhau mà không cần thông qua **Controller**. Lúc này nó chỉ đảm nhận nhiệm vụ hiển thị dữ liệu mà không cần thông qua quá trình xử lý logic nào. Tương tự với các vùng hiển thị tĩnh trên các Website giống như Block Sidebar.

***Model:** giữ các vai trò

- Cung cấp các phương thức tương tác trực tiếp với database để thực hiện các chức năng của hệ thống.

- Cung cấp các phương thức để thực hiện các thao tác với database (thêm, xóa, sửa) với các feature tạo và gán nhiệm vụ, cập nhật thông tin user,

- Khi trạng thái của model (về dữ liệu,...) thay đổi mà hệ thống cần cập nhật nhanh lên view => thông báo cho view để thực hiện refresh và cập nhật lại view.

3.1.2 Các Module dự kiến

1. Task Assignment

Input: Back officer

Output: Xác nhận việc gán task thành công.

Function: Tạo và gán task cho Collector và Janitor

2. Attendance

Input: Collector và Janitor

Output: User xác nhận công việc, xác nhận hoàn thành công việc

Function: Check-in task, Check- out task

3. View Task

Input: Collector and Janitor

Output: User xem được lịch trình làm việc và những công việc được giao.

Function: View Task Assignment, View work calendar

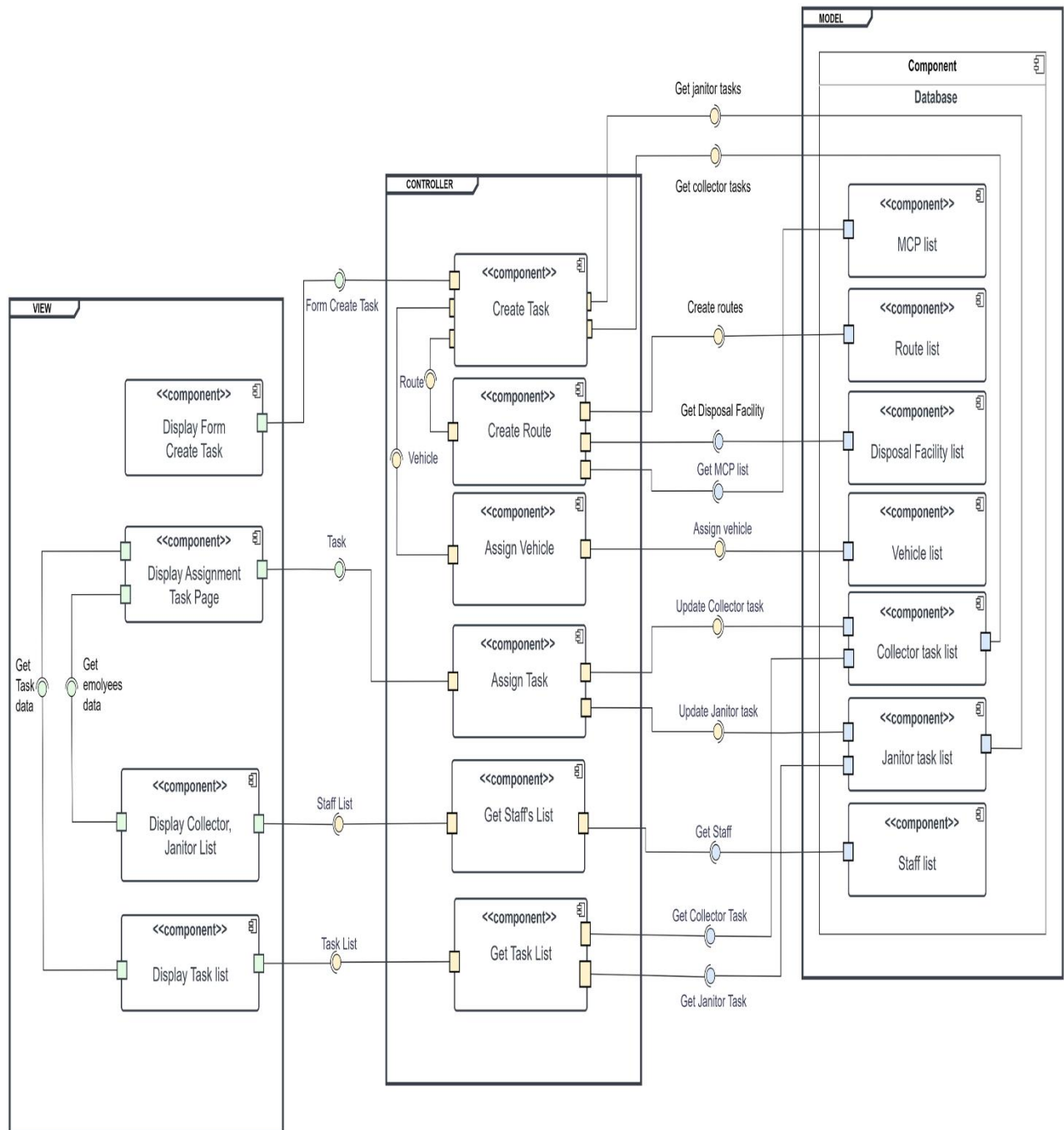
4. Communicate

Input: Collector, Janitor, Back officer

Output: User nhận thông báo từ hệ thống, chat thời gian thực với nhau.

Function: Notification, Chat

3.2 Component Diagram:



***Link Diagram:**

https://drive.google.com/file/u/2/d/1ge4Z9UzqwWiWu2ld5LvzGvYApVQtzoSG/view?usp=sharing&fbclid=IwAR1FBjrtsINpsc4Lp6AQJx60IcorC0tkuTLOk_xETtKOXZSrOx0athBCg6E

***Mô tả Diagram:**

- Khi Back officer truy cập vào trang phân công nhiệm vụ ở component Display Assignment Task Page sẽ được hiển thị các thông tin về danh sách các nhân viên (Janitor, Collector) thông qua component Display Collector, Janitor list ở View. Ngoài ra, thông tin các Task đã được tạo trước đó sẽ được hiển thị thông qua component Display task list.
- Back officer có thể chọn nút Create task để hiển thị form tạo task thông qua component Display Form Create Task. Tại component Create Task ở controller, nó sẽ sử dụng các component Create Route, Assign Vehicle để tạo ra các route giữa các MCPs, chọn phương tiện dùng để vận chuyển rác. Sau đó thông tin của task sẽ được chuyển xuống hệ thống và lưu vào database Janitor Task list và Collector Task list tại Model.
- Tại trang phân công nhiệm vụ, Back officer sẽ chọn 1 staff từ danh sách các staff, và có thể nhiều task từ danh sách các task. Sau đó gán các task đã chọn cho staff đó. Tại component Assign Task ở controller nó sẽ thực hiện các logic xử lý như việc task đang gán trùng lịch với lịch làm việc của staff đó, ... Sau khi xử lý xong, nó sẽ update thông tin nhân viên đã chọn vào task đó và lưu vào database.