



## REPORT TASK 1-2-3 SOFTWARE ENGINEERING

Group name: **Nontech**

# UWC 2.0

Teacher: Lê Đình Thuận

Họ và tên	Mã số sinh viên
Lê Duy Hào	2011142
Đỗ Thành Minh	2011610
Trần Minh Tuấn	2012360
Phạm Quang Sơn	1914973
Đặng Văn Vĩnh	1915962
Nguyễn Duy Hà	2011130

**Mục lục**

Mục lục	1
I. Introduction	2
II. Task 1	3
1. Functions	3
2. Non-Functions	4
3. Use case diagram	4
4. Use case description Task assignment module	7
III. Task 2	23
2.1. Activity diagram:	23
2.2: Route planning task.	24
2.3. Class diagram	27
IV. Task 3	31
3.1. Activity diagram:	31
3.2. Draw implementation diagram	34
V. Task 4	38
4.1. Setting up	38
4.2. Adding documents	38
4.3. Implement MVP1	38

## I. Introduction

In today's metropolitan cities, millions of people live packed together in densely populated areas and they all contribute to the tons of waste that end up in dumpsters daily. If someone tried to manually manage all that garbage, they would be absolutely overwhelmed; but with the help of municipal waste management software, they could collect the garbage with a fraction of the time and effort!

In order to avoid the previously mentioned scenarios, the responsible party for waste collection must adopt an effective method. The most efficient way to do this would be to turn to smart waste management and using municipal waste management software. Thanks to modern technology, the people responsible for waste collection are able to monitor and manage their devices, billing, inventory, and regulate safety standards. Additionally, by using municipal solid waste software, they could save lots of time, garbage truck fuel consumption, and other resources by measuring containers' fullness levels and not going to empty containers; pick the most optimal route for collecting waste around the city; determine the demand (for example, a country with warmer weather would require more frequent waste collection to avoid odor); as well as preventing visual, odor, noise, and water pollution.

Municipal waste management software lends a helping hand to the company that uses it in terms of reducing losses and increasing profits, and to the residents of a metropolitan area by making their lives easier. So, in its essence, waste collection software is more than just business: It is a service for a company, the people, and the environment!

### ***Who are relevant stakeholders?***

External stakeholders: Citizens, governments and organizations.

Internal stakeholders: Back officers, collectors and janitors.

### ***What are their current needs?***

- Governments and organizations: Find the best solutions to solve the problem of urban waste management.
- Citizens: Waste should be treated as quickly as possible.
- Back officers: There is a way to manage the garbage collection system effectively and quickly than the previous manual management.
- Collectors and janitors: It is possible to save time moving between garbage collection points, as well as to know which area of garbage is full to avoid the effort without gathering garbage..

### ***What could be their current problem?***

- Collectors and janitors:
  - The optimal plan has not been found to be able to solve the problem of waste collection and management effectively.
  - Spending a lot on garbage collection but not effectively.
- Citizens: Loping waste affects health, causes discomfort.
- Back officers:
  - There is no optimal solution to the problem of assigning work schedules to collectors and janitors.
  - It is not possible to optimize the movement of garbage trucks and it is impossible to know when the MCPs will be full.
- Collectors and janitors: Always working but efficiency in work is very low.

### ***In your opinion, what benefits UWC 2.0 will be for each stakeholder?***

- Government and organizations:
- Can manage and operate the collection in the most detailed and reasonable way.
- Save more resources than before (from employees to money).
- Citizen: Improve people's lives through improving the quality of the surrounding living environment.
- Officer in charge:
- Can be more proactive in division of work for collectors and sanitation workers in a detailed and clear manner.
- Get an overview of all MCPs and information about their capacity.
- It is possible to create separate routes for each collector, in order to optimize fuel and travel distance.
- Collectors and janitors:
- Only collect at that MCP when it knows it's full, saving time traveling to check.
- It is possible to know your detailed tasks in a day, and even beyond a week.

## II. Task 1

### 1. Functions

This section defines the functions that each position (back officer, collector, janitor) and the general system must be qualified to perform.

Role	Functional requirement
<b>General System</b>	<ul style="list-style-type: none"> <li>- Import and use the database from the current system UWC 1.0, but make sure to contain these information below for task specification</li> <li>- Store database to manage information of employees:               <ul style="list-style-type: none"> <li>+ General information (name, age, gender, address,..)</li> <li>+ Work information: work calendar, assigned vehicle, assigned MCPs and area, daily check in / check out information</li> </ul> </li> <li>- Store database to manage vehicles :               <ul style="list-style-type: none"> <li>+ Technical details: weight, capacity, fuel consumptions,..</li> <li>+ Assigned route</li> </ul> </li> </ul>
<b>Back officer</b>	<ul style="list-style-type: none"> <li>- Have an overview of janitors and collectors, their work calendar</li> <li>- Have an overview of vehicles and their technical details</li> <li>- Have an overview of all MCPs and information about their capacity, with the information is being updated continuously</li> <li>- Assign vehicles to janitors and collectors</li> <li>- Assign janitors and collectors to MCPs</li> <li>- Create a route for each collector. Assigned route is optimized in term of fuel consumption and travel distance</li> <li>- Be able to send message to collectors and janitors</li> </ul>
<b>Collector and Janitor</b>	<ul style="list-style-type: none"> <li>- Have an overview of their work calendar</li> <li>- Have a detail view of their task on a daily and weekly basic</li> <li>- Be able to communicate with collectors, other janitors and back officers</li> <li>- Check in / check out task every day</li> <li>- Be notified about the MCPs if they are fully loaded</li> </ul>

## 2. *Non-Functions*

This section defines the attributes or constraints that indicates how the system operates. They make our application or software run more efficiently and illustrates the system's quality.

These are several aspects of requirement:

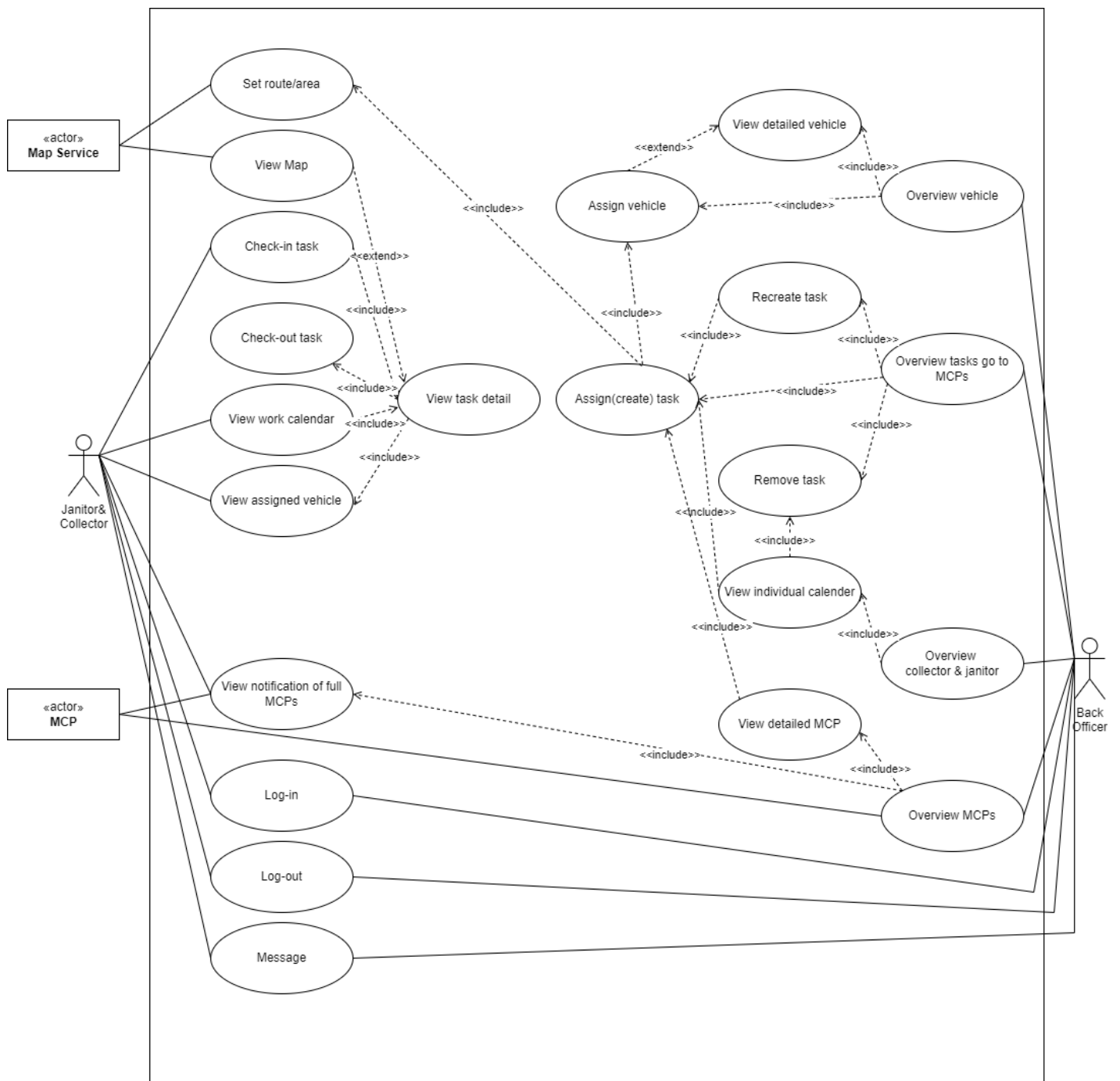
Criteria	Non-functional requirement
Reliability	<ul style="list-style-type: none"> <li>- When problems occur (such as system crash or power failure), the system must restore the previous state.</li> <li>- Record the description of problem:               <ul style="list-style-type: none"> <li>+ Type of problem</li> <li>+ Number of times occur</li> <li>+ The time when problem occurs</li> </ul> </li> </ul>
Security	<ul style="list-style-type: none"> <li>- Any users must have an account to log in</li> <li>- With the back officers, their power is bigger, so the security is more strict               <ul style="list-style-type: none"> <li>+ Require stronger password when create account</li> <li>+ After a certain number of fail login attempts, the system will lock this account</li> <li>+ Only allow user to log in by one device of company office</li> </ul> </li> <li>- With the janitors and collectors, whose knowledge about tech may be not good and the power is not important, the security requirement is easier</li> </ul>
Speed	<ul style="list-style-type: none"> <li>- Information of MCPs should be updated every 15 minutes with the availability of at least 95% of their operating time</li> <li>- Messenger engine among users makes sure delay less than 1 second</li> <li>- The notification to janitors when MCPs is full makes sure delay less than 1 second</li> </ul>
Portability	<ul style="list-style-type: none"> <li>- UWC 2.0 is expected to import and to use the existing data from UWC 1.0</li> <li>- The Task Management is expected to be inter-operable with the UWC 1.0 as much as possible.</li> </ul>
Usability	<ul style="list-style-type: none"> <li>- When collectors and janitors view their task on a daily and weekly basic, all important information should be displayed in one view (without scrolling down)</li> <li>- The user interface (UI) should be simple and friendly as much as possible for user who is not familiar to tech can easily get used to.</li> </ul>
Capacity	The system should be able to handle real-time data from at least 1000 MCPs at the moment and 10.000 MCPs in five years
Localization	The system interfaces should be in Vietnamese, with an opportunity to switch to English in the future

## 3. *Use case diagram*

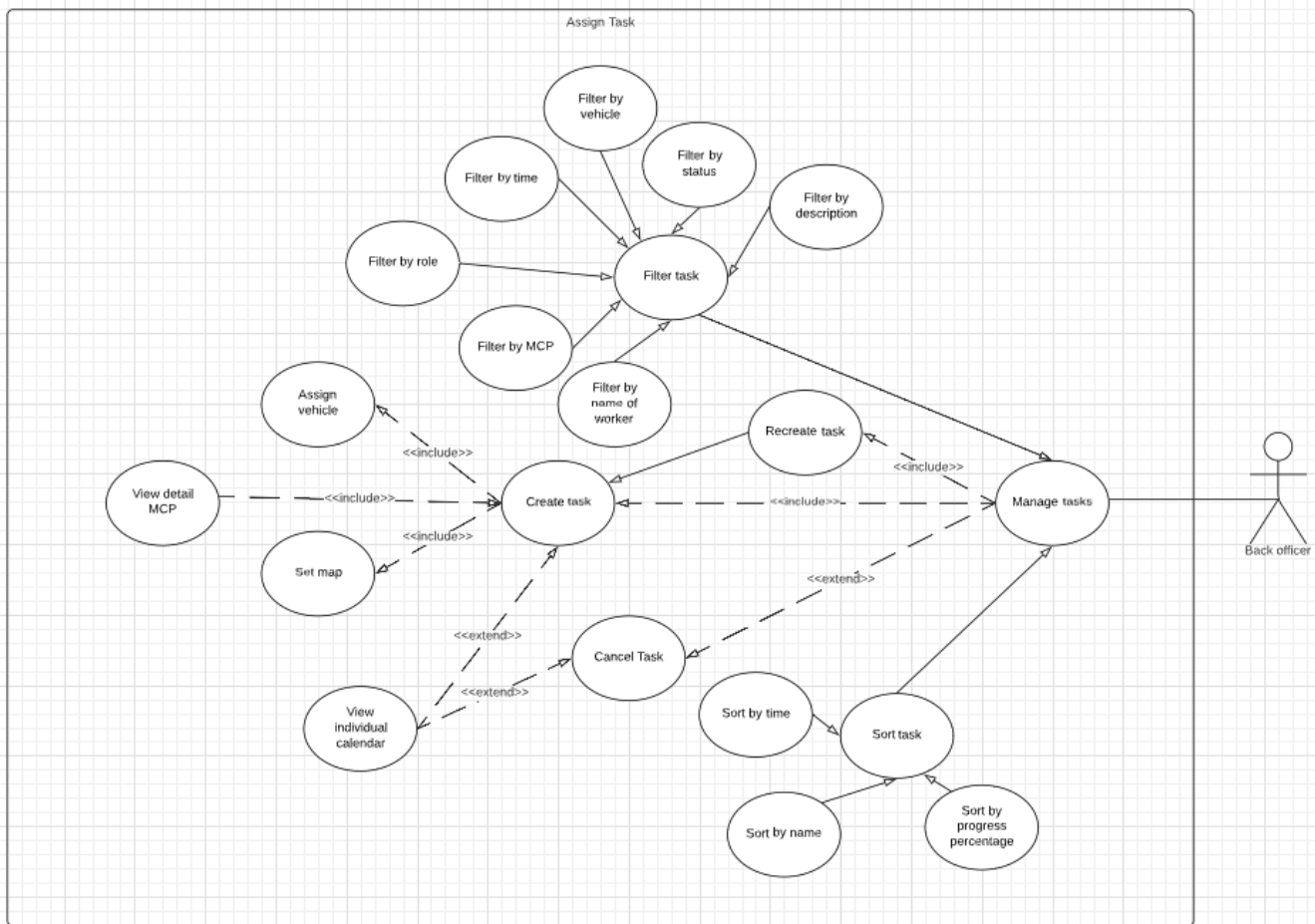
Table describes main use cases of the system

Use case name	General description
Log-in	Log in account registered on the system

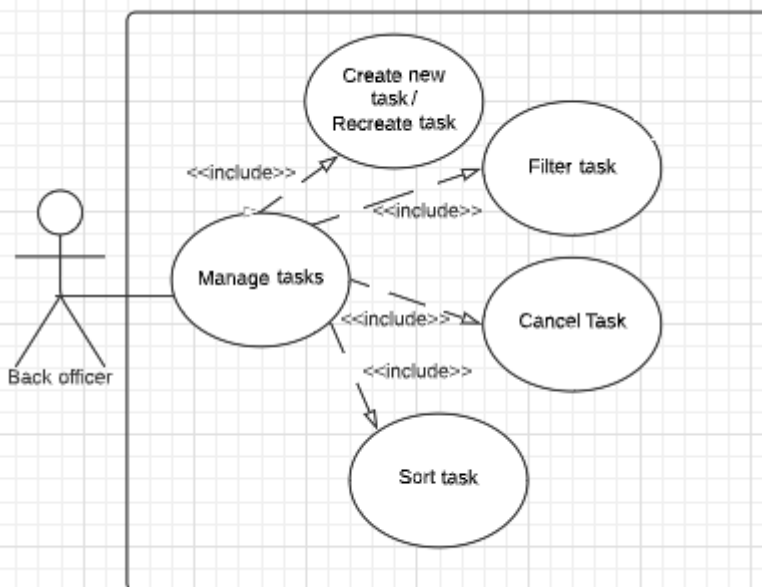
Log-out	Log out account registered on the system
Message	Communicate among the staff
Overview vehicle	View vehicles' detail and assign them for the collector/janitor
Overview task go to MCPs	<ul style="list-style-type: none"> <li>- Overview calendar of all tasks</li> <li>- Create a new task, includes assign staff, vehicle and route</li> <li>- Cancel a task</li> </ul>
Overview janitor/collector	Overview list of all janitors/collectors, their work calendars and assign task for them
Overview MCPs	Overview list of all MCPs and their details include storage status, position on map
View work calendar	View the schedule and detail of task has been assigned
View assigned vehicle	View which vehicle has been assigned to use for the tasks
View notification of full MCPs	Be notified when their assigned MCPs was full
View map	View all MCPs on the maps
Set route/area	Set a route for each collector
Check-in task	Check-in function for employee before work
Check-out task	Check-out function for employee after work
View task detail	Show detail information of employee's task
View detail vehicle	Show detail information of vehicle
Assign (create) task	Assign daily and weekly task to do for employee
Assign vehicle	Assign vehicle to use on duty to collector
Recreate task	Replace previously assigned task from employee with different task
Remove task	Remove task from employee
View individual	Show information of each employee



#### 4. *Use case description Task assignment module*



#### Manage task



Use-case name	<b>Manage tasks</b>
Actor	Back officers



Description	Back officers overview all created tasks, be able to add, cancel, sort in the list of tasks
Trigger	Back officers click on the button “Task” to access the UI
Preconditions	Back officers login validly into the system with manager permission account to access back officers UI.
Normal Flow	<ol style="list-style-type: none"> <li>1. UI displays tasks in spreadsheet, each task displays its detail</li> <li>2. Back officers from the UI can perform functions such as add, cancel, filter, sort task to edit: <ol style="list-style-type: none"> <li>a. Back officers click button “Create task” or button “Recreate” from existing task to perform action create a new task, the UI transform to ‘Create new task’ UI</li> <li>b. Back officers have option to “Cancel” with tasks have status ‘Scheduled’ or ‘Pending’, UI pop-ups to confirm cancelation, after confirm turn back to manage task UI</li> <li>c. Back officers click button “Apply filters” to filter task</li> <li>d. In the list of tasks Back officers can sort task according to each column</li> </ol> </li> </ol>
Alternative Flows	<p>Alternative flow 1: at step 2c</p> <p>- Back officers click “Clear filters”, task list turn back to sheet displays all tasks</p>

## Wireframe:

**Manage tasks**

Create task

MATCH ALL OF THE FOLLOWING:

Name Type... textbox

AND

Role Select dropdown list

AND

MCP Type or select datalist

AND

Vehicle Type or select datalist

AND

Date time datetime select

AND

Status Type or select datalist

AND

Select attribute dropdown list

Add

Clear filters

Apply filters

**List of tasks:**

grid view

sort

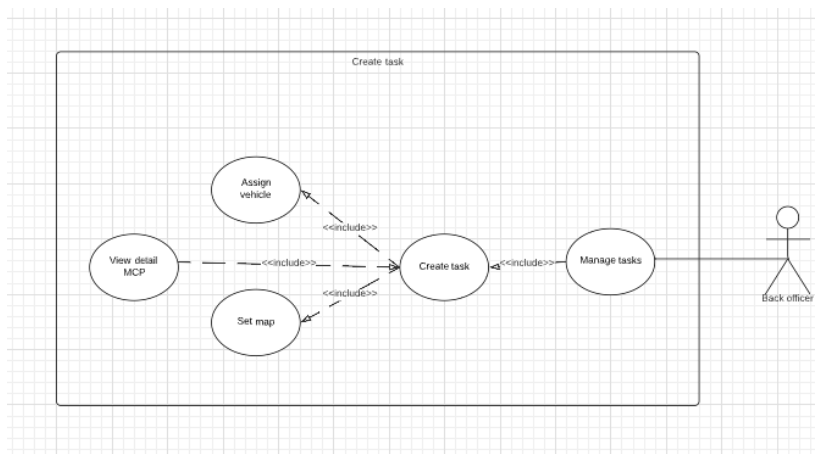
Name	ID	Role	Group	MCP	Vehicle	Date time	Status	Progress %	Description		
Richard Nguyen	J81	Janitor	J12	MCP 120	Trolley 150	12/27/2022 15:23	Pending	50%	check basement	Recreate	Cancel
Nguyen Thi A	C82	Collector	No group	MCP 121	Hino XZU342L 12	9/26/2022 16:03	Pending	30%	go quickly	Recreate	Cancel
Nguyen Van B	C92	Collector	No group	MCP 222	Hino XZU342L 29	9/9/2022 13:00	Done	100%	MCP nearly full	Recreate	Cancel
Tran Minh C	J21	Janitor	J111	MCP 321	Trolley 888	9/2/2022 4:20	Cancelled	0%	check on street	Recreate	Cancel
Pham Hoang D	J97	Janitor	No group	MCP 624	Trolley 1	1/21/2022 3:23	Scheduled	0%	ASAP	Recreate	Cancel

<< 1 2 3 4 5 >> paging

View all 10000 records

50 of 10,000 records

### 3.3.2 Create Task



Use case name	<b>Create tasks</b>
Actor	Back officers
Description	Back officers can create a new task on the system.
Preconditions	Back officers must log in to the system with an account with management rights.
Trigger	Back officers click on “Create task” button on the Manage interface
Postconditions	A new task is created according to the information entered in the fields, the new task is displayed on the Manage interface.
Normal flow	<ol style="list-style-type: none"> <li>1. The system opens the "Create task" interface.</li> <li>2. Back officers fill in the information of the task that you want to create in the fields: MCP, Date-time, Description</li> <li>3. The system displays a table “Person selected(*)” containing information about selected employees</li> <li>4. The system has a filter to select employees with the following attributes text “Choose person match”.</li> <li>5. Back officers fill in the information that you want to filter for employees in the fields: Name, ID, Role, Vehicle.</li> <li>6. Back officers select properties to filter to find employees.</li> <li>7. Back officers click on the "Apply filters" button to filter employees, the interface displays the employee table with the right attributes to filter.</li> <li>8. Back officers click on “Select” to select the appropriate employee or “Select all” to select all, then the employee will be displayed on the “Person</li> </ol>

	<p>selected” panel, click on “Deselect” to remove the employee from the “List” panel. of workers matched”.</p> <p>9. Back officers click on "Clear filters", the interface clears all information being filtered.</p> <p>10. Back officers click on “Next” to complete task creation and switch to “Assign Vehicle” interface.</p>
Exceptions	<ul style="list-style-type: none"> <li>- Exceptions 1: step 2 If Back officers enter MCP that does not exist, the system displays “not found” under the field “MCP(*)”</li> <li>- Exceptions 2: step 8 If Back officers enter a role that does not exist, the system displays “not found” under the “Role” field.</li> <li>- Exceptions 3: step 9 If Back officers enter the MCP that does not exist, the system displays “not found” under the “Vehicle” field.</li> <li>- Exceptions 4: step 14 If Back officers enter missing information, when clicking "Next", the system displays "Insufficient information" at the bottom of the interface.</li> </ul>
Alternative flows	No

**User story:** As a Back officer, after receiving a task from my superior, I have a need to create a task to assign to employees. Therefore, I click on "Create task", fill in all the information in accordance with the actual task requirements, complete the system to record the new task and display it on the Manager interface.

**Wireframe:**

[Create tasks](#)

[Next](#)

MCP(\*)

Date-time(\*)

Description

Person selected(\*)

Choose person match:

Name

AND

ID

AND

Role

AND

Vehicle

AND

Select attribute

Type then select ▼

Type...

Type...

Type or select ▼

Type or select ▼

Select attribute ▼

data list

datetime select

textbox

textbox

data list

data list

button

+ Add condition

Clear filters

Apply filters

[List of workers matched:](#)

grid view

Name	ID	Role	Group	Vehicle	Availability
Nguyen Van F	J81	Janitor	J12	Trolley 150	Available
Bui Van H	J82	Janitor	J12	Trolley 151	Unavailable
Nguyen Van B	J83	Janitor	J12	Trolley 22	System is checking ...

Deselect all

Deselect

Deselect

Deselect

Name	ID	Role	Group	Vehicle
Richard Nguyen	J81	Janitor	J12	Trolley 150
Nguyen Thi A	C82	Collector	No group	Hino XZU342L 12
Nguyen Van B	C92	Collector	No group	Unassigned
Tran Minh C	J21	Janitor	J111	Trolley 888

Select all

Select

Select

Select

Select

**Create tasks**

Next

MCP(\*) MCP 113 data list

Date-time(\*) 11/10/2022 datetime select

Description This task is .....

Person selected(\*)

Name	ID	Role	Group	Vehicle	Availability	
Nguyen Van F	J81	Janitor	J12	Troller 150	Available	<span>Deselect all</span>
Bui Van H	J82	Janitor	J12	Troller 151	Unavailable	<span>Deselect</span>

Choose person match:

Name Type... textbox

AND Type... textbox

AND Type or select data list

AND Type or select data list

AND Select attribute

+ Add condition

button

Clear filters

Apply filters

**List of workers matched:**

grid view

Name	ID	Role	Group	Vehicle	
Richard Nguyen	J81	Janitor	J12	Troller 150	<span>Select all</span>
Nguyen Thi A	C82	Collector	No group	Hino XZU342L 12	<span>Select</span>
Nguyen Van B	C92	Collector	No group	Unassigned	<span>Select</span>
Tran Minh C	J21	Janitor	J111	Troller 888	<span>Select</span>
Pham Hoang D	J97	Janitor	No group	Unassigned	<span>Deselect</span>

**Deselect employee**

[Create tasks](#) Next

MCP(\*) MCP 113 datalist

Date-time(\*) 11/10/2022 datetime select

Description This task is .....

Person selected(\*)

Name	ID	Role	Group	Vehicle	Availability	
Nguyen Van F	J81	Janitor	J12	Troller 150	Available	<span>Deselect</span>
Bui Van H	J82	Janitor	J12	Troller 151	Unavailable	<span>Deselect</span>

Choose person match:

Name Nguyễn Văn A textbox

AND

ID CO1110 textbox

AND

Role Collectors datalist

AND

Vehicle Type or select datalist

AND

Select attribute Names, ID, Role

+ Add condition Clear filters Apply filters

List of workers matched:

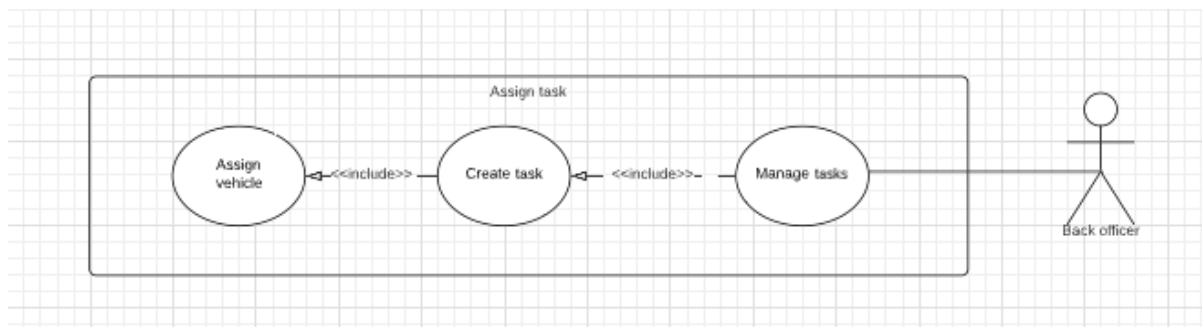
Name	ID	Role	Group	Vehicle	
Nguyen Van A	CO1110	Collectors	J12	Troller 150	<span>Select all</span>
Nguyen Van A	CO1110	Collectors	G40	Troller 209	<span>Select</span>

<< < 1 2 3 4 5 > >> 4 of 1,000 records

[View all 1,000 records](#)

### Filter employee

## 3.3 Assign vehicle



Use-case name	<b>Assign vehicle</b>
Actor	Back officers
Description	Back officers check assignment of vehicles to employees.
Trigger	Back officers click “Next” on the first page to move to the 2nd page: “Assign vehicle”.

Preconditions	Complete the content on the first page
Normal Flow	<ol style="list-style-type: none"> <li>1. The interface switches to the 2nd page: “Assign vehicle”. Display the list of selected employee and their vehicles</li> <li>2. Back officers click “Assign vehicle” next to the employee they want to assign.</li> <li>3. Back officers select the vehicle they want to assign in the list of vehicles’ information appeared and press the “Assign” button to assign that vehicle to the employee they had chosen.</li> <li>4. Press the “Done” button to complete the vehicle assignment.</li> <li>5. The system updates changes and returns the employee list interface.</li> <li>6. Back officers click “Next” to switch to the 3rd page: “Set map”.</li> </ol>
Alternative Flows	<ul style="list-style-type: none"> <li>- Alternative flow 1: at step 1 If Back officers do not want to continue assigning vehicles, click “Cancel” to return to the task list interface.</li> <li>- Alternative flow 2: at step 2 Back officers have not completed the assignment of vehicles (vehicle is “Unassigned”), then when clicking “Next”, the interface will display “No vehicle specified” next to the names of employees.</li> </ul>

**User story:**

As a Back office, I want to assign vehicles to them after getting a list of my employees to manage. Therefore, I switched from the “Create task” page to the “Assign vehicle” page, selected the name of the employee I wanted to assign, and a list of existing vehicles appeared on the screen for me to choose from. After I made the selection, the system updated the employee's vehicle to the interface.

**Wireframe:**

[Check out vehicles](#)

Back Next

label Person selected(\*)

Name	ID	Role	Group	Vehicle(*)
Nguyen Van F	J81	Janitor	J12	Troller 150
Bui Van H	J82	Janitor	J12	Unassigned
Nguyen Van B	J83	Janitor	J12	Troller 22

Assign vehicle Deselect Assign vehicle Assign vehicle

**Assign vehicle:**

Person selected:

Name	ID	Role	Group	Vehicle(*)
Bui Van H	J82	Janitor	J12	Unassigned

Choose vehicle

Vehicle	ID	Current user	Previous user	Form-To
Troller 150	381	None	Nguyen Van F	
Troller 888	999	None	Bui Van H	
Troller 22	23	Nguyen Van B	Nguyen Van B	20/9/2022-29/9/2024

Assign Detail Assign Detail Retake Detail Done

[Check out vehicles](#)

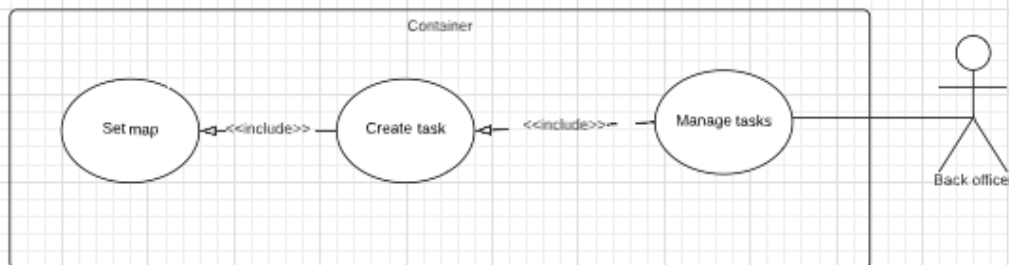
Back Next

Person selected(\*)

Name	ID	Role	Group	Vehicle(*)
Nguyen Van F	J81	Janitor	J12	Troller 150
Bui Van H	J82	Janitor	J12	Troller 888
Nguyen Van B	J83	Janitor	J12	Troller 22

Assign vehicle Assign vehicle Assign vehicle

## Set map



Use-case name	<b>Set map</b>
Actor	Back officers
Description	Back officers set route for Collectors or choose area work for Janitors
Trigger	Back officers press the button “Next” at step 2: “Assign vehicle”.
Preconditions	The form in step 2 has been filled out all required information
Normal Flow	<ol style="list-style-type: none"> <li>System goes to page step 3: “Set map”, there are 2 cases: <ol style="list-style-type: none"> <li>If this task is set for Janitors, the page displays nearby areas to choose.</li> </ol> </li> </ol>



	<p>b. If this task is set for Collectors, the page displays the list of MCPs selected and some predetermined routes with estimated time and distance.</p> <p>2. Finish setting map, proceed to create task</p>
Alternative Flows	<p>- Alternative flow 1: at step 1</p> <p>Back officer press Back, the app go back to step 2 “Set vehicle”</p>

### User story:

As a Back office, after getting a list of my employees to manage, I want to assign vehicles to them. Therefore, I switched from the “Create task” page to the “Assign vehicle” page, selected the name of the employee I wanted to assign, and a list of existing vehicles appeared on the screen for me to choose from. After I made the selection, the system updated the employee's vehicle to the interface.

### Wireframe:

#### Routing for collector:

[Routing](#)

Vehicle

Hino XZU342L 12

Back

Finish

MCP	Attitude	Current Load	Last checked	Capacity
MCP 123	9.6856637,105.5630432	90%	12 mins ago	130 kg
MCP 999	10.6856637,95.5630432	70%	11 mins ago	1000 kg
MCP 008	9.6949704,105.5650387	55%	5 mins ago	1129 kg

Auto-estimated time

40 mins

Choose

Total distance

16,1 km

Auto-estimated time

42 mins

Choose

Total distance

16,7 km

Auto-estimated time

44 mins

Choose

Total distance

17 km

**Routing**

Vehicle

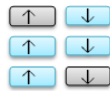
Hino XZU342L 12

-MCP selected(\*)

MCP	Attitude	Current Load	Last checked	Capacity
MCP 123	9.6856637,105.5630432	90%	12 mins ago	130 kg
MCP 999	10.6856637,95.5630432	70%	11 mins ago	1000 kg
MCP 008	9.6949704,105.5650387	55%	5 mins ago	1129 kg

Back

Finish



Auto-estimated time 40 mins  
Total distance 16,1 km

Auto-estimated time 42 mins  
Total distance 16,7 km

Choose

Auto-estimated time 44 mins  
Total distance 17 km

Choose

**Choose work area for janitor:****Work area**

Vehicle

Troller 887

Area selected(\*)

Not selected

MCP destination

MCP 008

Near-by area suggestion:

Area name Area 51  
Distance from MCP 0.5 km

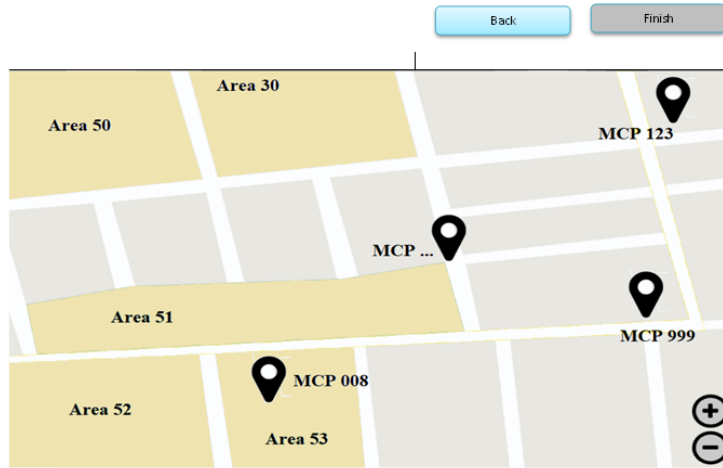
Add

Area name Area 52  
Distance from MCP 0.7 km

Add

Area name Area 53  
Distance from MCP 1 km

Add

**Routing**

Vehicle

Troller 887

Area selected(\*)

Area	Distance from MCP
Area 51	0.5 km
Area 52	0.7 km
Area 53	1 km

X

X

X

MCP destination

MCP 008

Near-by area suggestion:

Area name Area 50  
Distance from MCP 1 km

Add

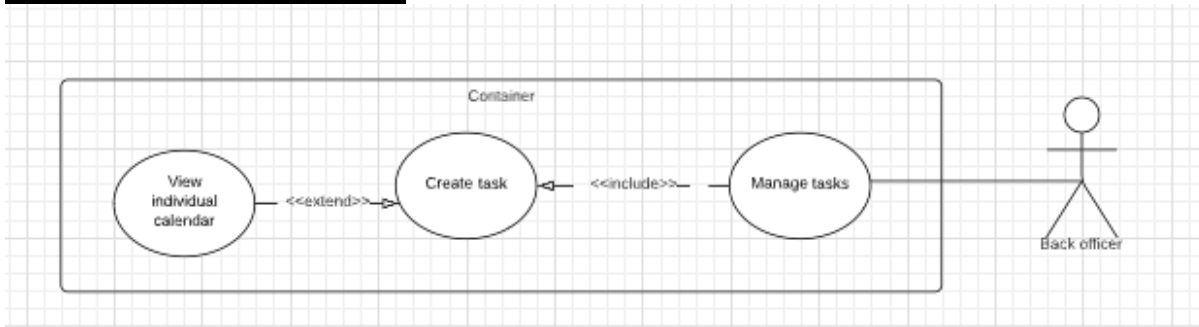
Area name Area 30  
Distance from MCP 2 km

Add

Area name Area 32  
Distance from MCP 3 km

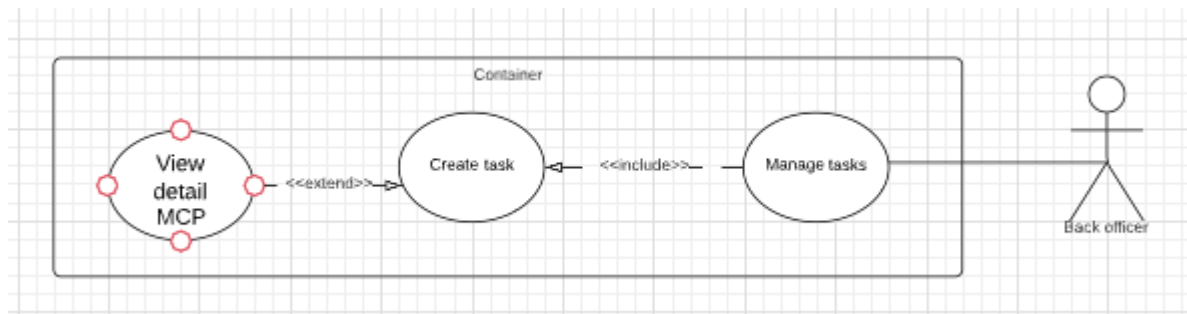
Add



**View individual calendar**

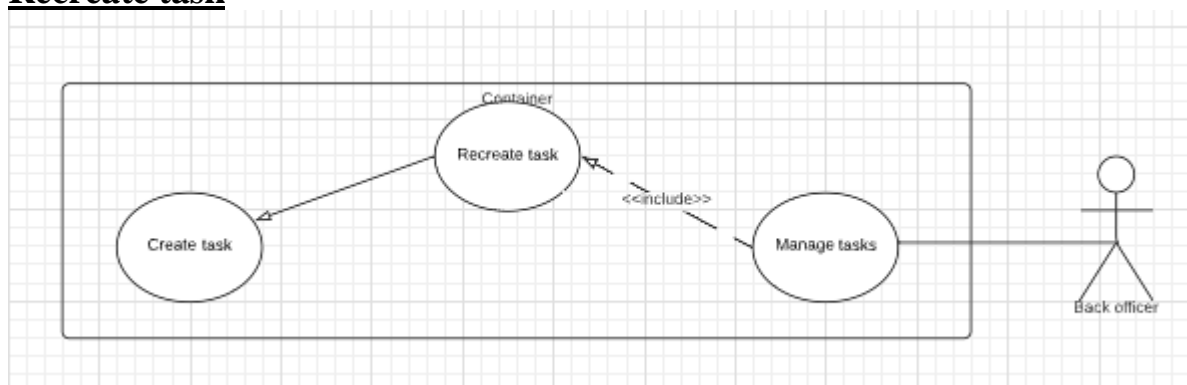
Use-case name	<b>View individual calendar</b>
Actor	Back officers
Description	Back officers view the work calendar of a specific collector/janitor
Trigger	Back officers press button “View individual calendar” in the “Manage Collector/Janitor” page
Preconditions	No
Normal Flow	<ol style="list-style-type: none"> <li>1. The app display the individual calendar in one view</li> <li>2. Back officers press button “Create task” to create task for this individual</li> <li>3. System proceed to go to “Create task”, the interface of create task is already filled the name of the individual. Back officers finish creating task and the system go back to the individual calendar.</li> <li>4. At the individual calendar, back officers choose a specific task and press “Cancel task” to cancel the task assigned to that individual, after canceling, the system go back to the individual calendar</li> </ol>
Alternative Flows	- Alternative flow 1: at step 3 Back officers press “Cancel”, the app goes back to the individual calendar

**Wireframe:**  
**View detail MCP**



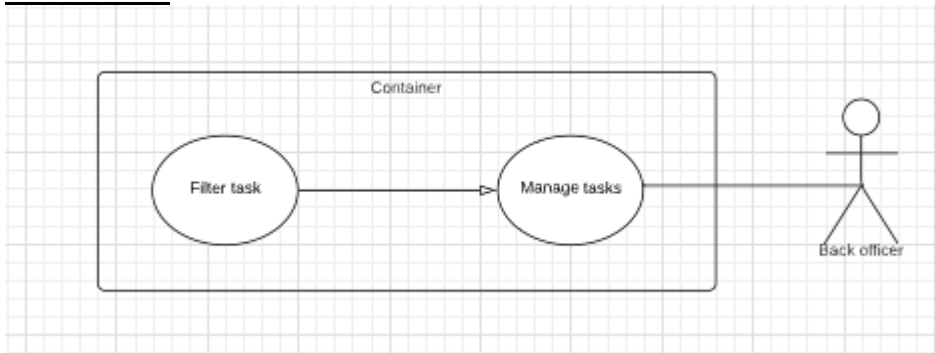
Use-case name	<b>View detail MCP</b>
Actor	Back officers
Description	Back officers assign work to a specific MCP
Trigger	Back officers press “MCPs” button
Preconditions	None
Normal Flow	<ol style="list-style-type: none"> <li>1. The system lists all MCPs on user display.</li> <li>2. Back officers select the search bar, type the needed MCP, then the system will come up with the result.</li> <li>3. Click on MCP, the user interface will show the location of that MCP on the maps.</li> <li>4. Back officers select “Create task”, the user interface display a new task whose MCP field have been filled in automatically.</li> <li>5. To finish creating task, select “Finish”, then the user interface will go back to the maps with the location of the current MCP.</li> </ol>
Alternative Flows	<ul style="list-style-type: none"> <li>- Alternative flow 1: at step 2 Back officers clear the search bar, the user interface go back to the list of all MCP.</li> <li>- Alternative flow 2: at step 3 Back officers select “Cancel”, stop creating new task, the user interface go back to the maps with the location of current MCP</li> </ul>

### Recreate task



Use-case name	<b>Recreate task</b>
Actor	Back officers
Description	Back officers choose to recreate a task using the same information extracted from the task selected
Trigger	Back officers press “Recreate” button
Preconditions	Task is exist in the system database
Normal Flow	<ol style="list-style-type: none"> <li>1. The interface displays the “Create task” interface with all completed information fields similar to the selected task recreate.</li> <li>2. Back officers make information corrections as desired.</li> <li>3. Click “Finish” to finish creating a new task, the interface returns to the task list.</li> </ol>
Alternative Flows	- Alternative flow 1: at step 1 Back officers press “Cancel” button, stop creating new tasks, and return to the task list interface task.

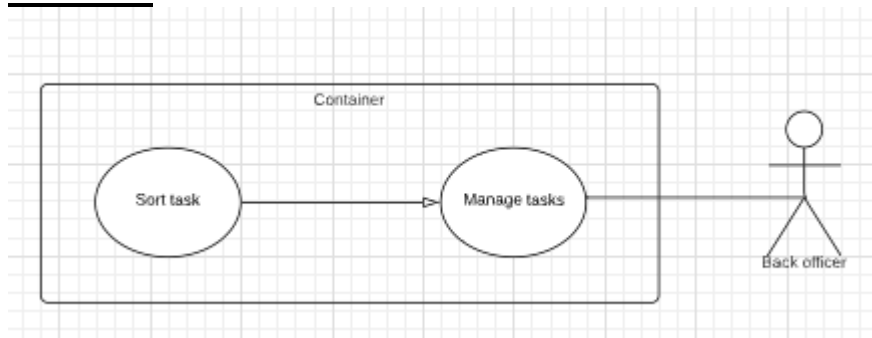
### **Filter task**



Use-case name	<b>Filter task</b>
Actor	Back officers
Description	Back officers filter tasks with information to find
Trigger	Back officers press “Filter” button
Preconditions	No
Normal Flow	<ol style="list-style-type: none"> <li>1. The interface displays a list of filterable fields, including:             <ol style="list-style-type: none"> <li>a. Description</li> <li>b. Status</li> <li>c. Vehicle</li> <li>d. Time</li> </ol> </li> </ol>

	<p>e. Role f. MCP g. Name of worker</p> <p>2. Back officers select one of the fields, enter the information to be filtered, click “Filter”.</p> <p>3. The system filters on demand and the interface displays a list of tasks with information Back officers need to find.</p>
Alternative Flows	<p>- Alternative flow 1: at step 3 Back officers delete filtered content, press “filter” to return to the filterable fields list interface.</p> <p>- Alternative flow 2: at step 3 Back officers enter the filter content that does not exist, the interface does not display any task, shows the text “No task”.</p>

### Sort task



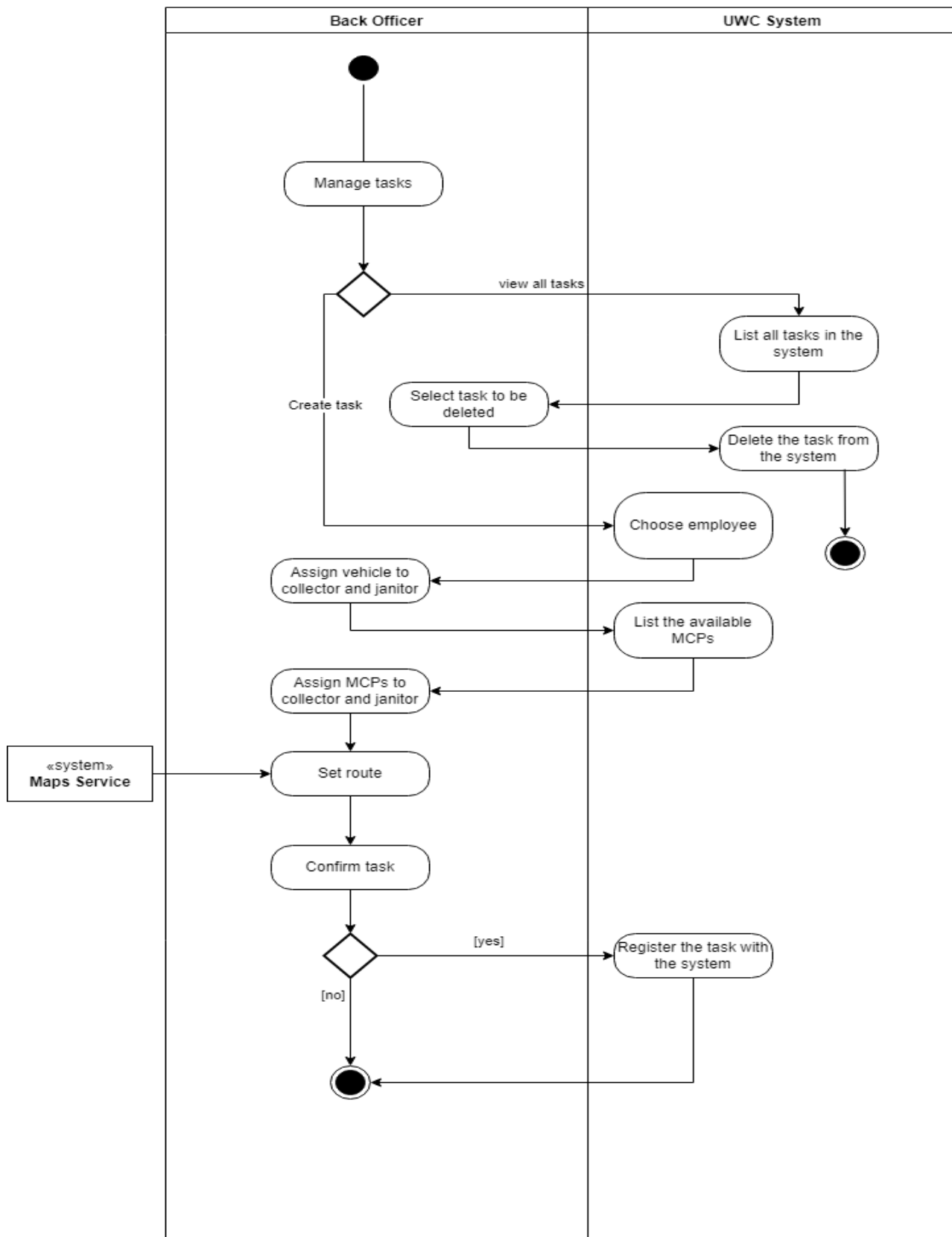
Use-case name	<b>Sort task</b>
Actor	Back officers
Description	Back officers want to organize tasks that have the information they need to find
Trigger	Back officers click “Sort”
Preconditions	No
Normal Flow	<ol style="list-style-type: none"> <li>The interface shows a list of fields that can be sorted, including: <ol style="list-style-type: none"> <li>Time</li> <li>Name</li> <li>Progress percentage</li> </ol> </li> <li>Back officers select one of the fields, enter the information to be filtered, and then click the “Sort” button.</li> <li>The interface displays all tasks sorted by the field selected by Back officers.</li> <li>Back officers arrange the jobs in either ascending or descending</li> </ol>

	order by pressing "A-Z" or "Z-A."
Alternative Flows	- Alternative flow 1: at step 3 Back officers delete the filtered content and then click “Filter” to return to the interface of the list of sortable fields.

### III. Task 2

#### 2.1. Activity diagram:

Task Assignment module



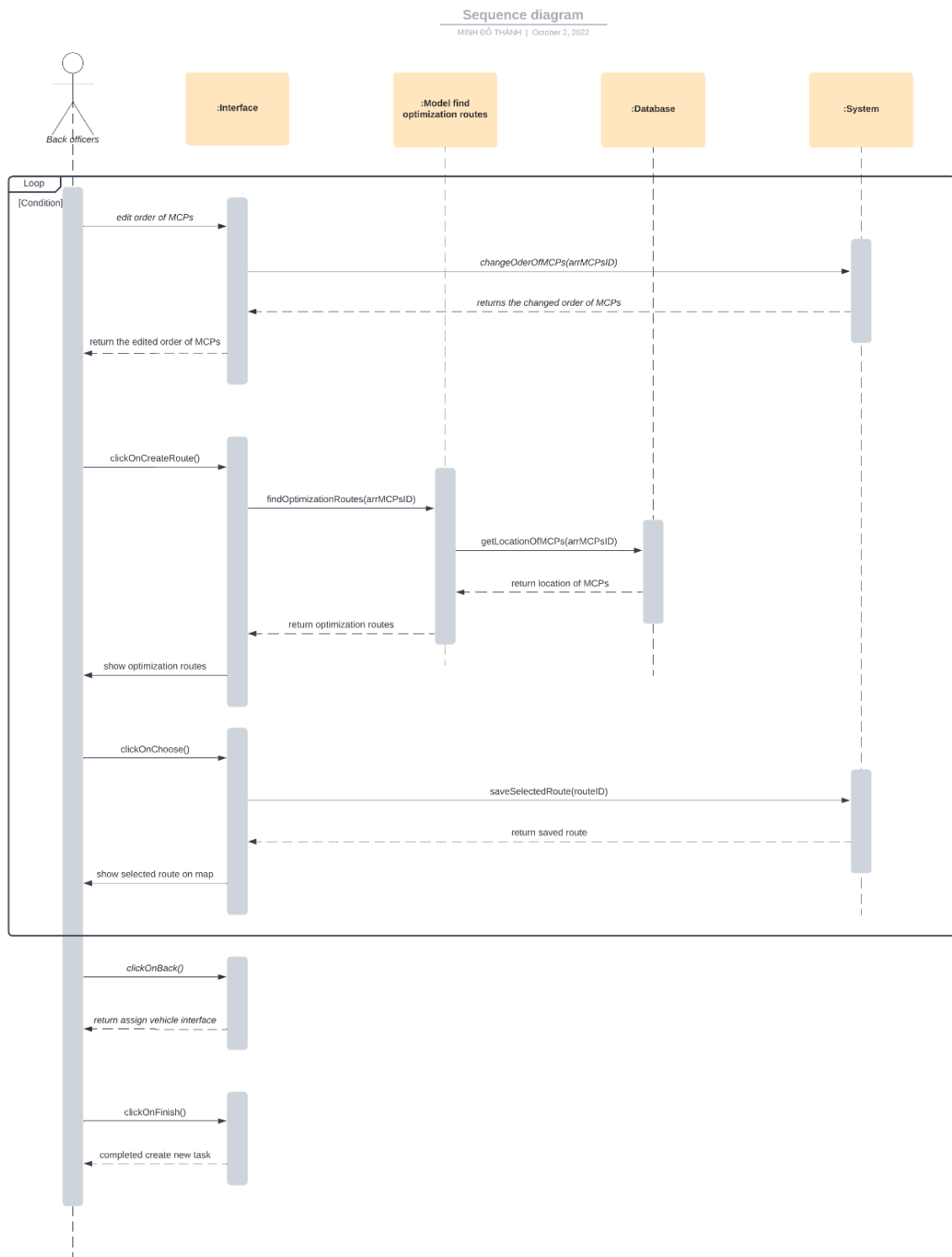


## **2.2: Route planning task.**

Proposal: With the desire to be able to offer a fast route, saving time, effort and money, and at the same time minimizing the impact on the environment when using petrol-powered transportation vehicles to collect waste at MCPs, the team proposed export solution for “Create route” feature.

To create convenience and reduce effort, the team will take advantage of services provided by 3rd parties (maybe Google Maps, HERE WeGo, MAPS.ME, ...) to select destinations that are MCPs for the type of vehicle desired by the user. From there, the system runs algorithms to calculate the optimal route and display it for the user to choose.

**Sequence diagram:**



**Routing**

Vehicle: Hino XZU342L 12

MCP selected(\*)



MCP	Attitude	Current Load	Last checked	Capacity
MCP 123	9.6856637,105.5630432	90%	12 mins ago	130 kg
MCP 999	10.6856637,95.5630432	70%	11 mins ago	1000 kg
MCP 008	9.6949704,105.5650387	55%	5 mins ago	1129 kg

Auto-estimated time: 40 mins  
Total distance: 16,1 km

Auto-estimated time: 42 mins  
Total distance: 16,7 km

Auto-estimated time: 44 mins  
Total distance: 17 km

### Description:

- Back officers through the "Routing" interface to select the optimal path for Collectors through pre-selected MCPs.
- The "Routing" interface displays the MPCs selected in the "Create Task" interface with the following attributes: MCP ID, Attitude, Current Load, Last checked, Capacity
- The back officer can change the order of MCPs that Collectors must follow by:
  - + Click “” to swap the order of the same row and bottom MCP.
  - + Click “” to swap the order of the MCP of same row and row above.
- The changed order of MCPs is sent to the system, which processes and returns the order interface of the changed MCPs.
- Back officers click “Create route” to create routes through all selected MCPs. At this point, the software switches to a third-party service, retrieves the location of the MCPs from the database to calculate, returning the routes through all the MCPs.
- Back officers click “Choose” to choose the most optimal route among the proposed routes, the system proceeds to save the selected route, the interface displays the route on the map.
- Changing the order of MCPs, calculating and choosing the optimal path for the Collectors can be repeated many times.
- Back officers can click “Back” to switch to the “Assign Vehicle” interface.
- Back officers click “Finish” to switch back to complete the task of creating a new task.

## 2.3. Class diagram

### 2.3.1. Log in

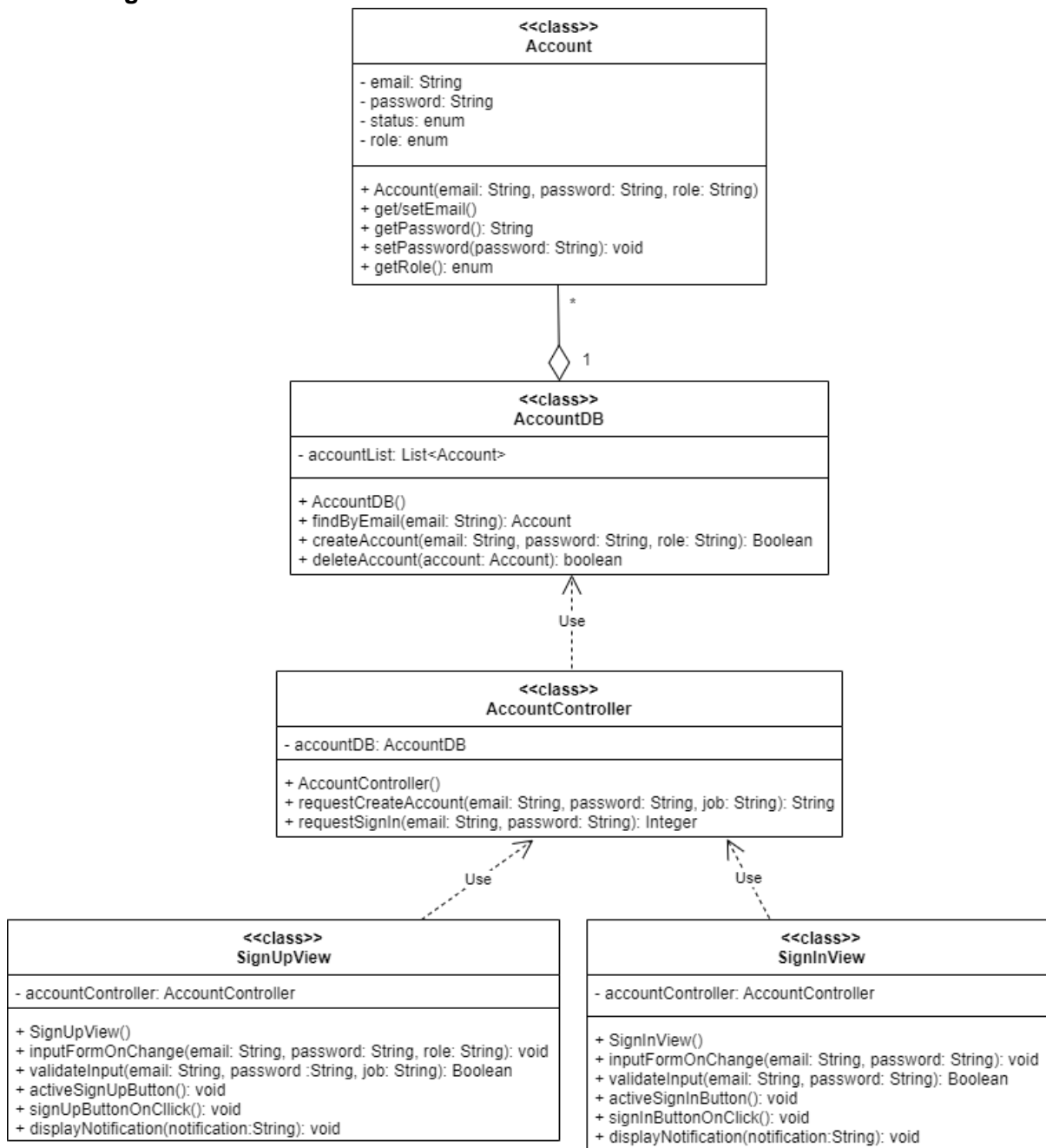


Diagram 1. User Sign in and Sign up Activity

### 2.3.2. Employee management

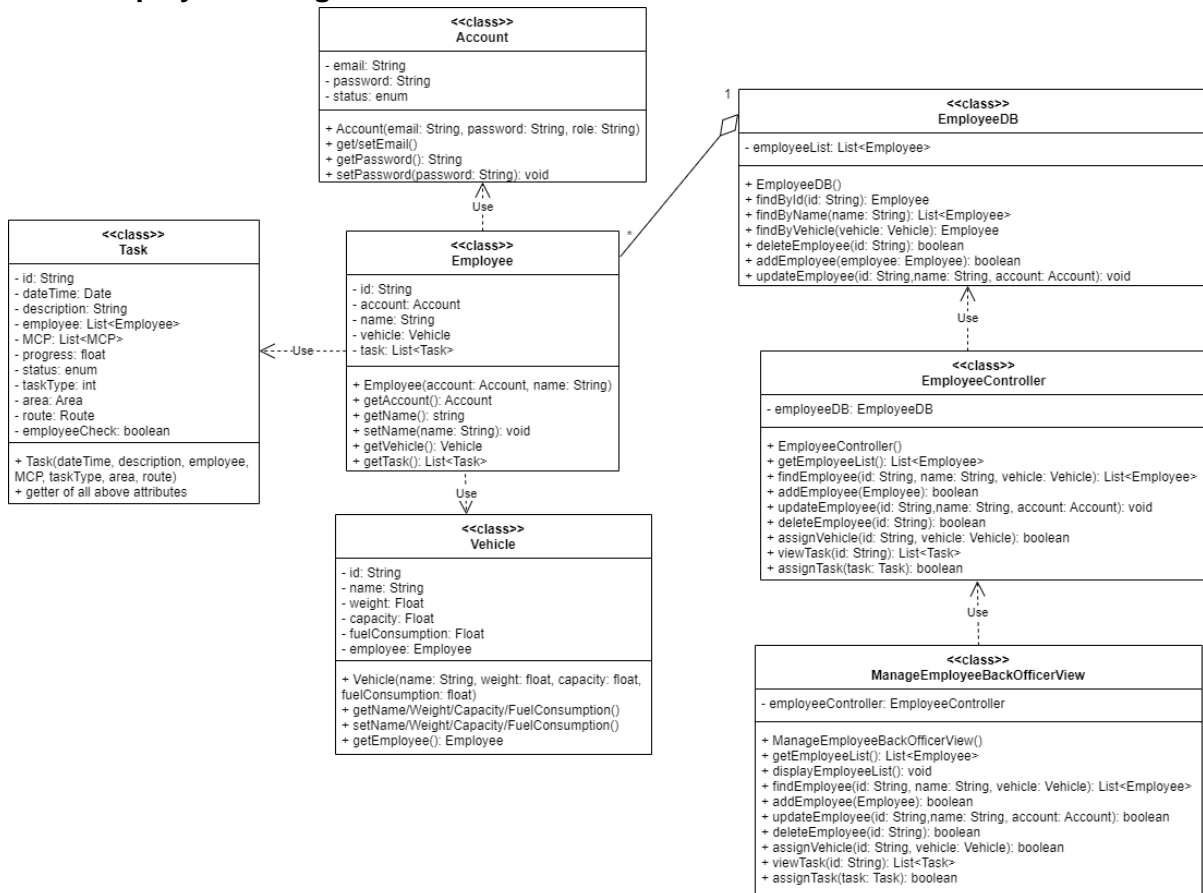


Diagram 2. Back officer Manage employee Activity

### 2.3.3. Vehicle management

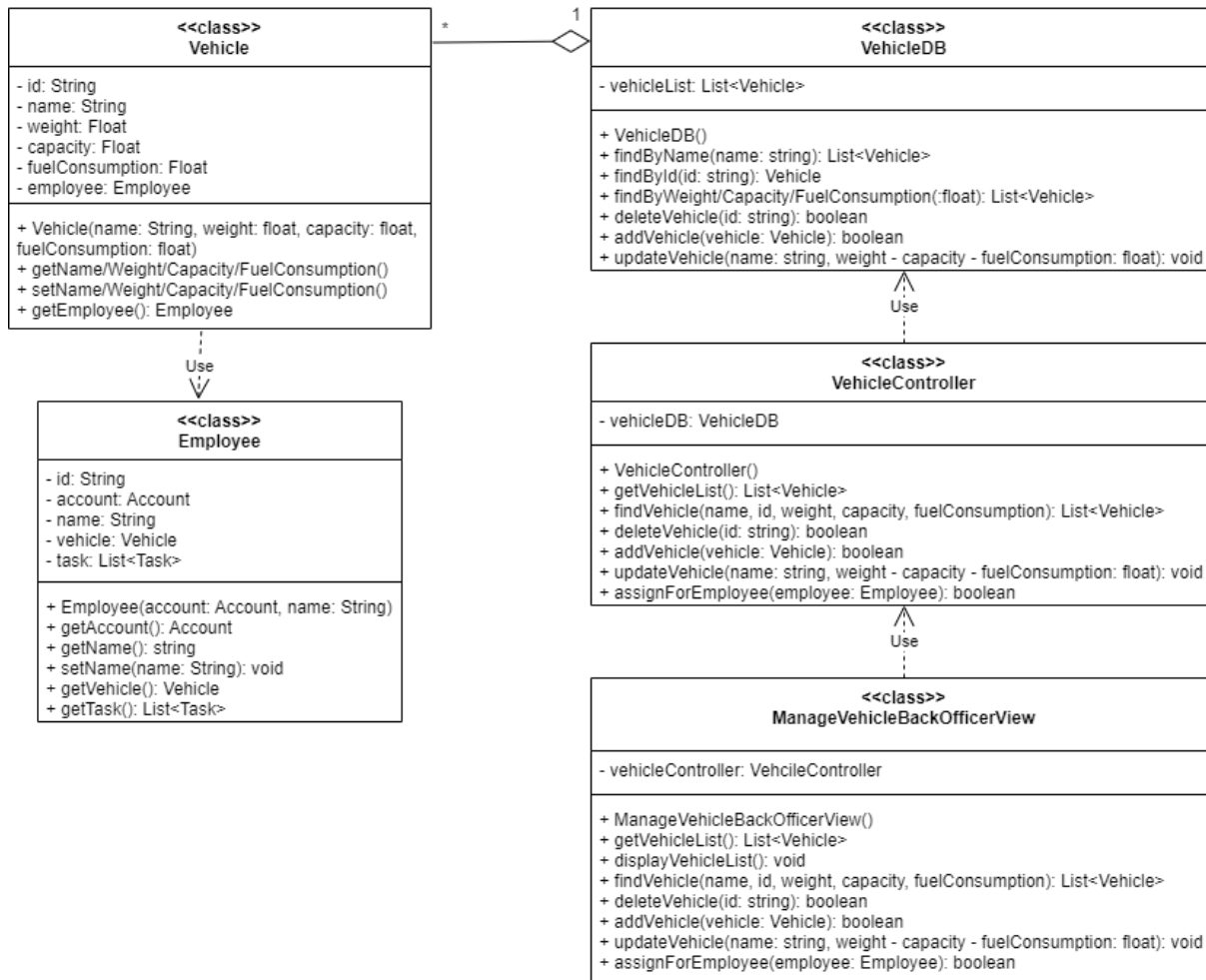


Diagram 3. Back officer Manage vehicle Activity

### 2.3.4. Task Management

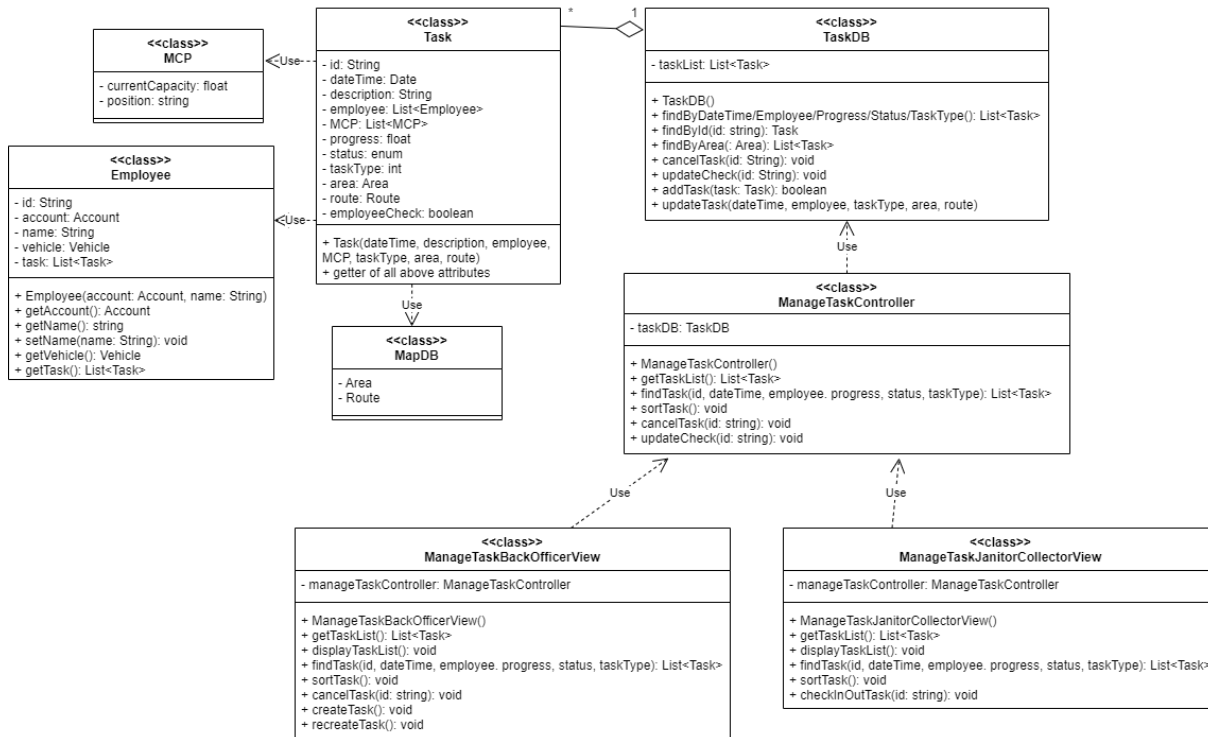


Diagram 4. Back officer / Janitor – Collector Manage task Activity

### 2.3.5. Create task

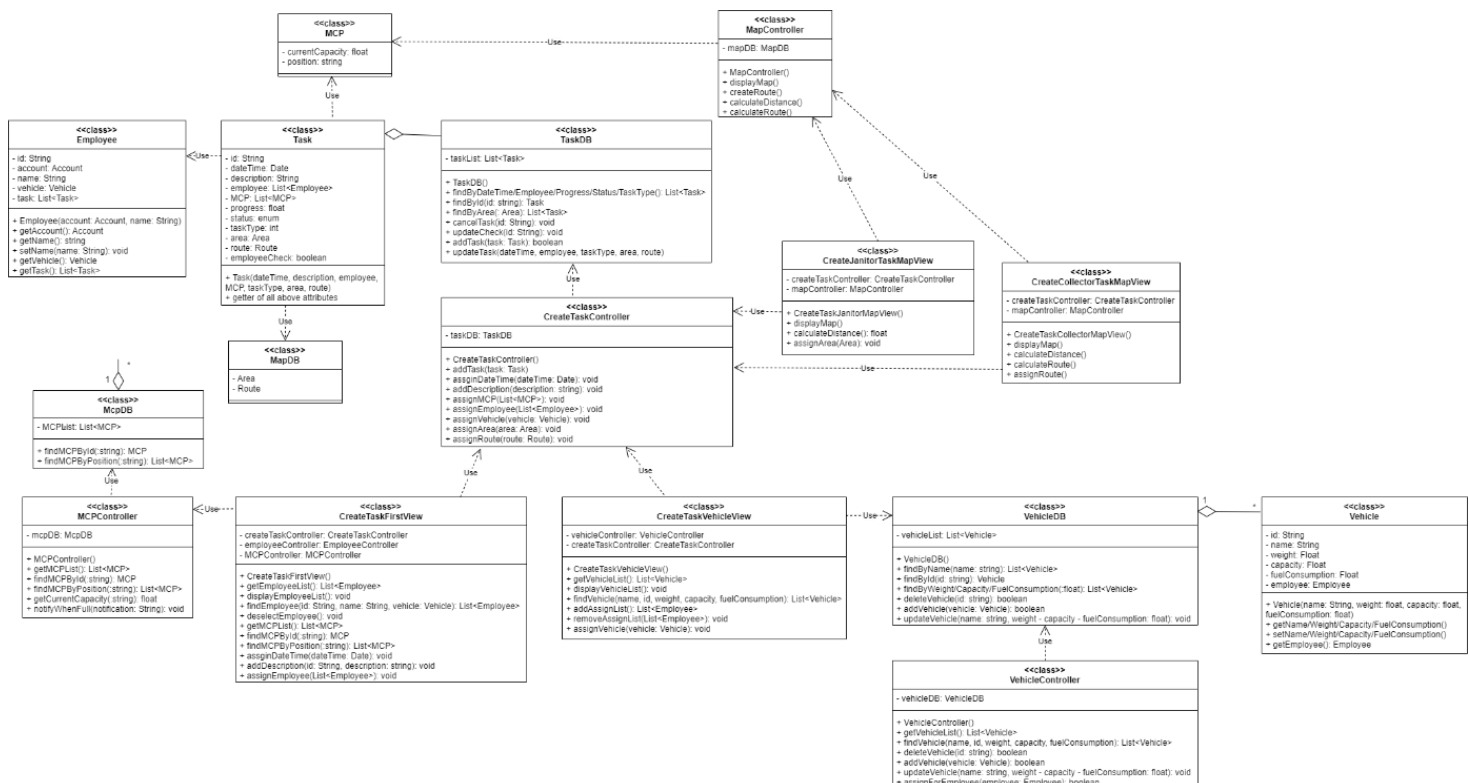


Diagram 5. Back officer Create task Activity

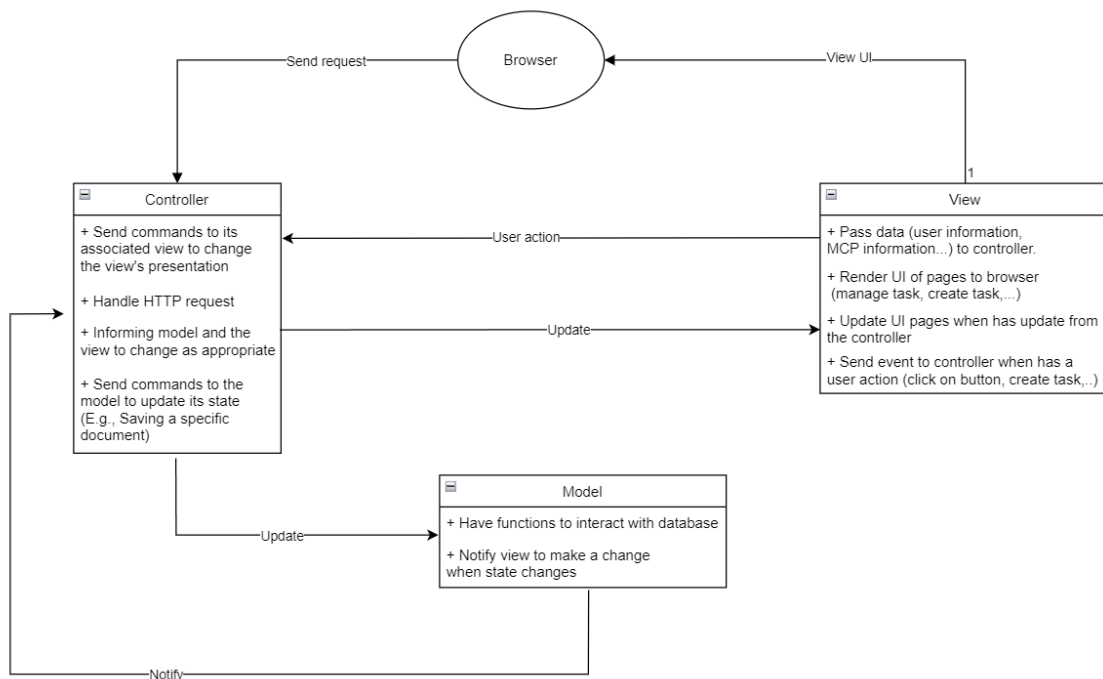
Google Drive:

[https://drive.google.com/drive/folders/1nUJXA\\_6PSIWp5ORsfGcAvIXZAEo-IWeD](https://drive.google.com/drive/folders/1nUJXA_6PSIWp5ORsfGcAvIXZAEo-IWeD)

## IV. Task 3

### 3.1. Activity diagram:

#### 3.1.1. Describe an architectural approach you will use to implement the desired system.



### Mô tả:

#### - View:



+ Send the entered data into the interface to the Controller for processing:

- User login information such as account, password, role; account registration information; change the account information
- Search for employees by attributes (id, name,...); add, delete or update an employee's information; assign tasks and vehicles to employees
- Search vehicle by attributes (id, name, weight, capacity,...); add, delete, update information of a vehicle.
- Find task information through properties (DateTime, Employee, Process, ID, Area,...); create task; cancel task; update check; add tasks; update task; sort task; recreate task.
- Search MCP through attributes (ID, Position); create route, calculate distance, calculate route.

+ Render UI for page:

- Log in page: render login forms, account registration, buttons such as: confirmation, login, forget password,...
- Employee management page: render an employee management interface including a list of employees, a search bar, a sort tool by attributes, a button to switch to an employee's information interface, task, vehicle.



- Vehicle management page: render the vehicle management interface including a list of vehicles, a search bar, a button to switch to a vehicle's information interface, and an employee interface.
- Task management page: render task management interface including task list, search bar, sort tool; The “create task” button switches to the create task interface, the “recreate” button switches to the recreate task interface, the “cancel” button to cancel the task, the “clear filters” button, the “apply filters” button is used for task filtering.
- Create task page: render create task interface, including a task information input form; button “clear filters”, “apply filters” for filtering workers, “select” button, “deselect” for canceling and selecting workers, “next” button to switch to “Assign vehicle” interface.
- Assign vehicle page: render interface including information table “person selected”, “assign vehicle”, “checkout vehicle”, buttons “assign”, “detail”, “assign vehicle” interact with vehicle, button “done” to complete, button “back” to return to “create task” interface, “next” button to switch to “set map” interface.
- Set map page: render the interface includes the "MCP selected" table containing information about the selected MCPs; map shows the MCPs; buttons   to change the order of MCP, button “choose” to select route, “back” button to return to “assign vehicle” interface, “finish” button to complete task creation.

+ Update UI pages when has update from the controller: perform update is the interface when receiving changes from the controller, retrieve data from the model to do it accordingly

+ Send event to controller when has a user action:

- Send events performed by the user on the interface such as click on button, dropdown list, ... to the controller for retrieval and processing.

## - Controller:

+ Returns a view depending on the requested page.

+ Calling functions from the model to perform operations on data:

- Use the model's methods to do things like:
  - Update task, vehicle, employee information, route, area of the MCP.
  - Delete task, vehicle, employee information, route, area of the MCP.
  - Edit details of task, vehicle, employee information, route, area of the MCP.
- Use methods from the model to perform registration, login, and user account management operations.

+ Calling functions from models to use APIs from external systems:

- View details of each employee's individual calendar.
- View detailed information about the system's MCPs.
- View detailed information of a vehicle in the list.

+ Returns the response to the user with the corresponding request.

### - Model:

+ Provides methods to handle the assignment of collectors and janitors to optimize collection quickly and efficiently.

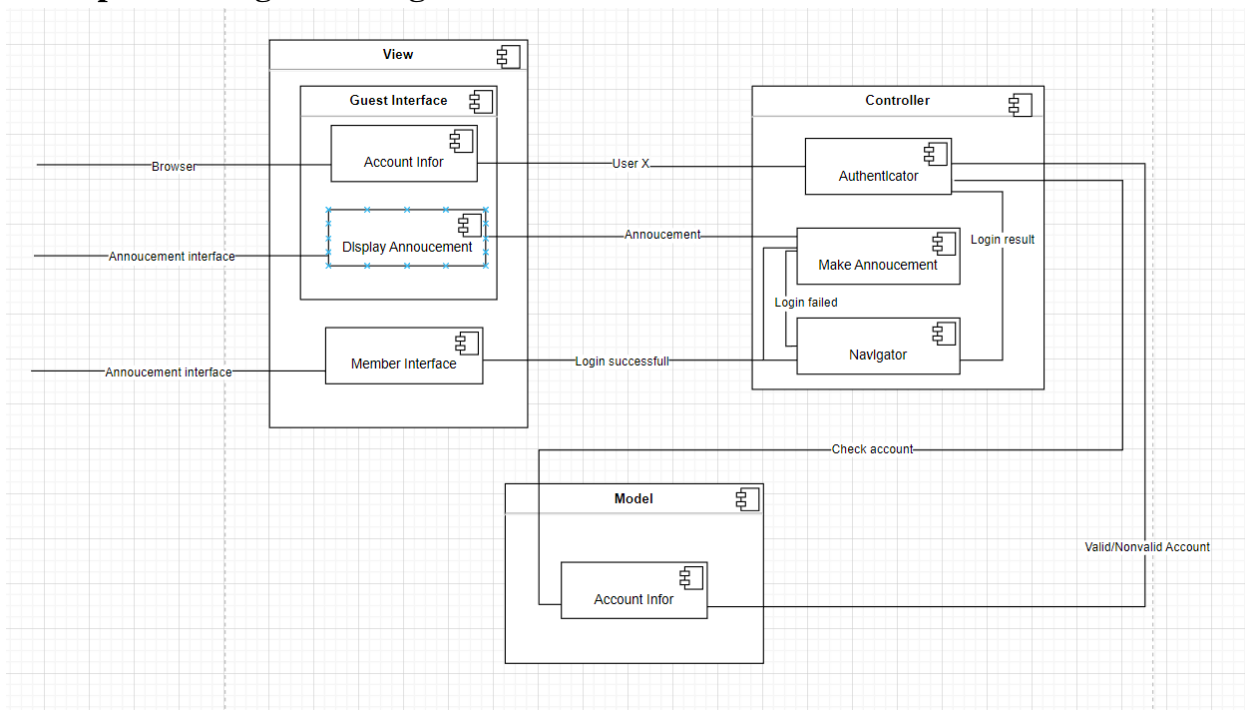
+ Provides methods to interact with the database to perform login, user account management Back officers, collectors and janitors.

+ Provide methods to perform database operations (add, delete, edit, assign...) with the features of route management, vehicle management, and MCP management.

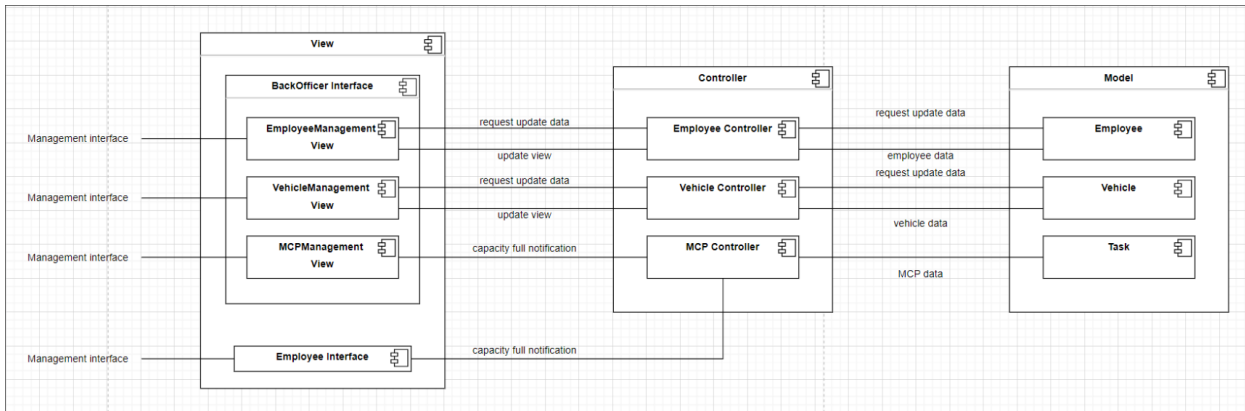
+ When the model's state (about the amount of garbage contained at the MCP, the nearest route, the means of collection...) changes => notify the view to refresh and update the view again.

### 3.1.2. How many modules you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module.

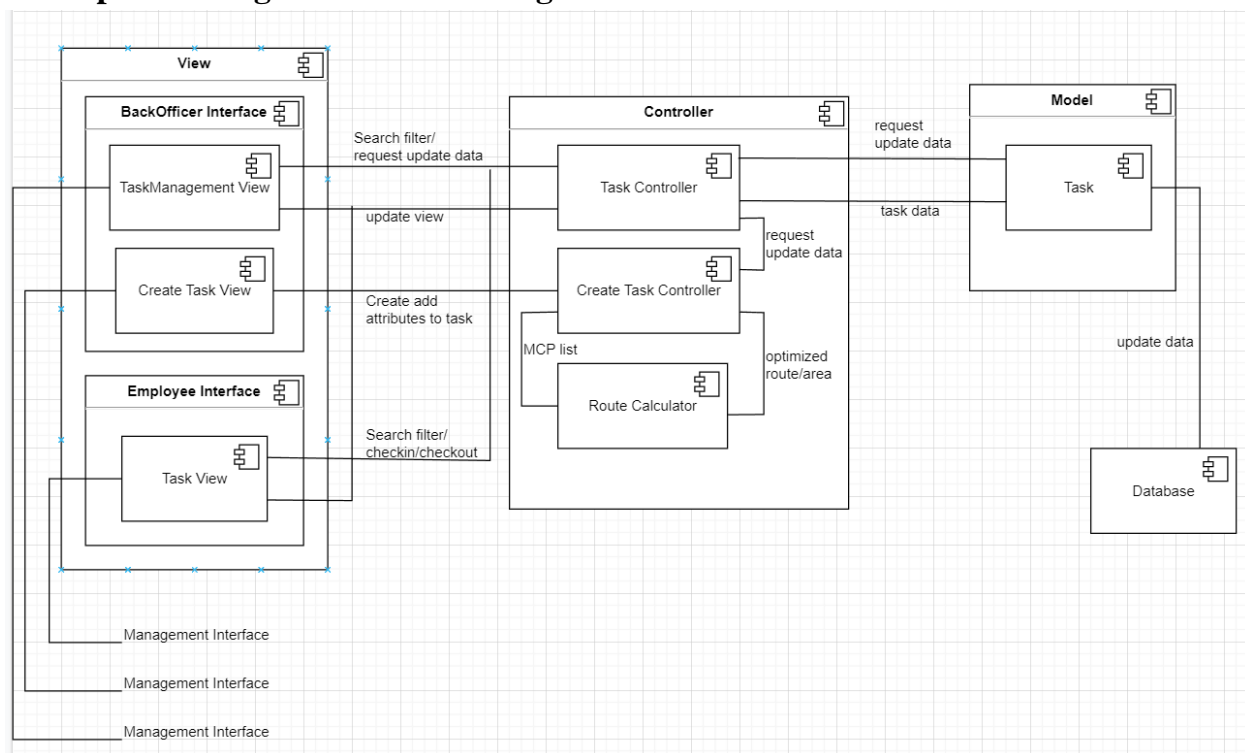
#### - Component diagram – Log in.



#### - Component diagram – Vehicle, Employee and MCP Management.



### - Component diagram – Task Management.

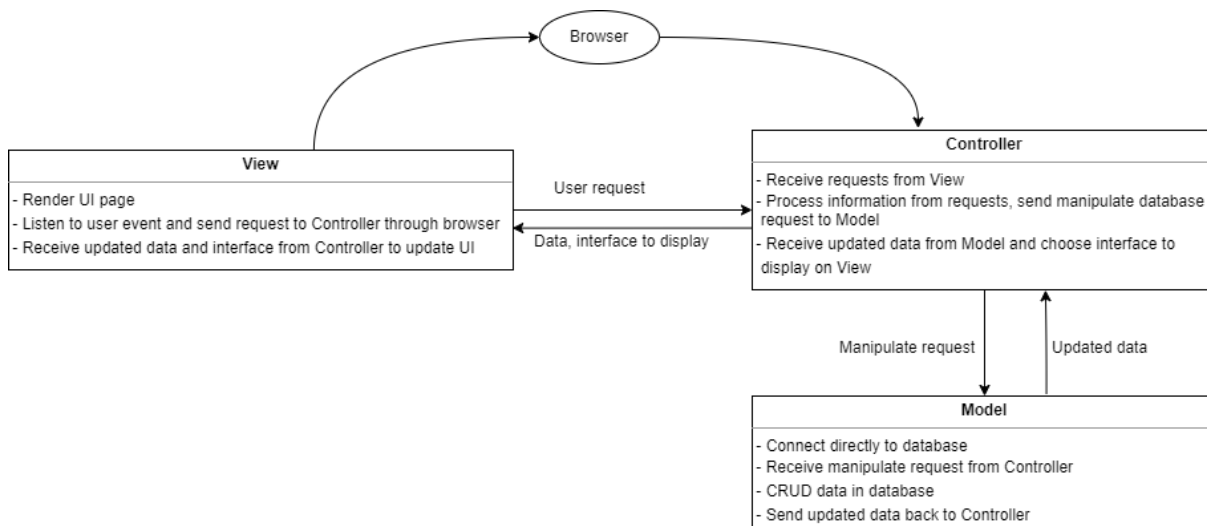


### 3.2. Draw implementation diagram

For higher quality of images and diagram file, please take a look in our GG Drive:

[https://drive.google.com/drive/folders/1ZEsLQrs6oZ98L3Wqgmdqe9oZ0S\\_uGryQ?usp=sharing](https://drive.google.com/drive/folders/1ZEsLQrs6oZ98L3Wqgmdqe9oZ0S_uGryQ?usp=sharing)

**Approach:** Use MVC (Model – View – Controller) model, description

**Description:**

- *Controller* and *Controller* connect to each other through browser.

- *Controller*:

+ Render UI element, data receive from Controller display to the user

+ Listen to event from user like sort, search, create, update, delete and send requests to Controller to handle.

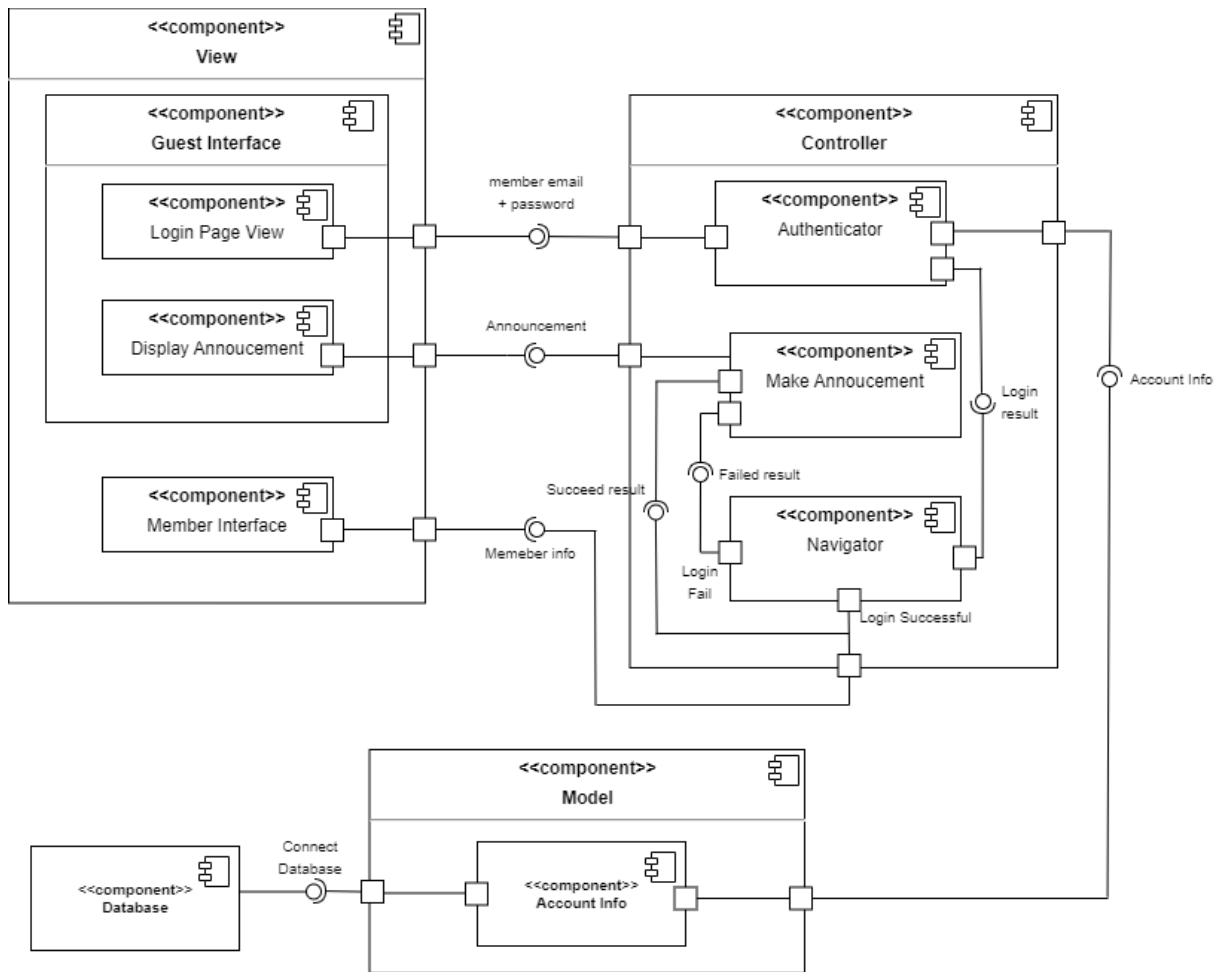
- *Controller*: Handle request and input data from View and request manipulate request to Model, receive data back from user and send to update *View*

- *Model*: connect and can change data in database, send data to *Controller*

**Advantages:** Separate into three parts, easier to maintain and can be used for more than one component, like *Controller* can be use for both Employee, Vehicle and MCP management.

**Difficulty:** Make sure the logic between the components work well

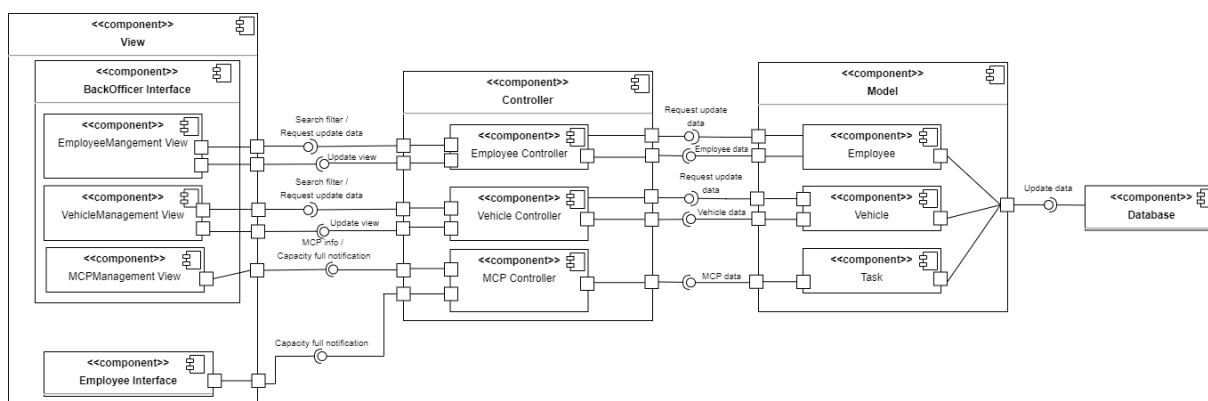
**3.2.1. Component diagram – Log in**



### Description:

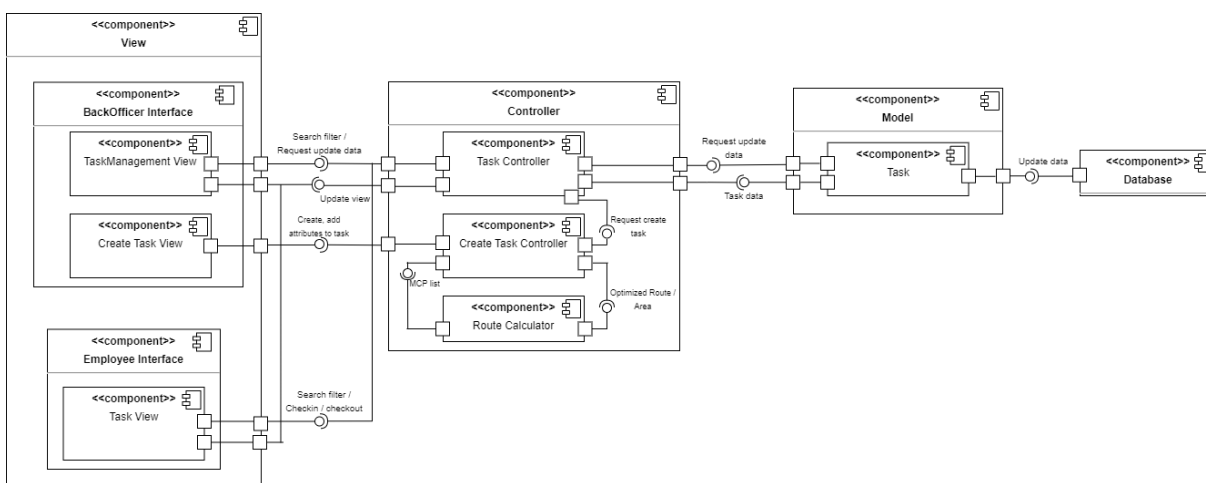
- At the login page in *<Guest Interface>*, the employees (officer and janitor / collector) input *member email* and *password*. This information are received by *<Authenticator>* component, and compared with data from *<Account Info>* in *<Model>*. After that, the result is sent to *<Navigator>* component to navigate, and notifies to the employee through *<Display Annoucement>* component.
- In case find corresponding account, employee will be navigated to the appropriate *<Member Interface>* with their role and information.
- In case cannot find corresponding account, employee will be notified and requested to login again.

### 3.2.2. Component diagram – Vehicle, Employee and MCP Management



**Description:**

- According to *MVC* model, from <View>, at <BackOfficer Interface>, the back officer has three functions to manages *Employee*, *Vehicle* and *MCP*, which can request to *sort*, *filter to search* an update data in database.
- Requests from <View> are sent to and handled by <Controller>, finally sent to <Model> to manipulate data in <Database>. And again the change of database (include notification when a MCP is full) is sent back to <Controller> to update <View>.
- In <Employee Interface> include janitor and collector, they can't send request to manipulate data, but also receive notification when MCP is full.

**3.2.3. Component diagram – Task Management****Description:**

With *Task Management* module, the back officer has full fuctions CRUD to the task database (create, read, update, delete (cancel)).

About read, update and cancel tasks, the work flow is same as Employee/ Vehicle/ MCP Management module was presented in previous section, following the MVC model.

The <Task Assignment> / <Create new task> component has own interface, where the back officer can select information for a new task manually, includes time, employee, vehicle, list of MCPs,... This information is sent to <Create Task Controller> component.

- The list of assigned MCP is sent to <Route Calculator> component, which is implemented to calculate the optimized area (for janitor) or route (for collector). The output is sent back to <Create Task Controller>, now has full information of a task, will send request to create a new task to <Task Controller>, the Controller finally work with Model to manipulate database.

About <Employee Interface> of janitor/ collector, they can search, sort and change the state of check in / check out their own tasks.

## V. Task 4

### 4.1. Setting up

The team creates an online repository using Github as the 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.

Github link:

<https://github.com/sonqp/SE221>

### 4.2. Adding documents

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: attached with log.txt in submission.

### 4.3. Implement MVP1

**Desktop-view central dashboard for Task Management for back-officers**

#### Login

UWC 2.0

https://www.uwc.com

# CÔNG TY MÔI TRƯỜNG

## UWC 2.0

Đăng nhập

email

Mật khẩu

☐ Ghi nhớ đăng nhập [Quên mật khẩu?](#)

Đăng nhập

## Task manage

Quản lý công việc UWC 2.0

127.0.0.1:5500/task.html

QUẢN LÝ CÔNG VIỆC

Create task (+)

Đăng xuất

Show 10 entries Search:

ID	Tên	ID nhân viên	Vai trò	Nhóm	MCP	Phương tiện	Thời gian tạo	Trạng thái	Tiến độ (%)	Mô tả	Bắt đầu	Kết thúc		
1665849098	Le Duy Hao	000002	collector	C21	321.456.789	019	15/10/2022 22:51:38	done	100	You must be quickly	16/10/2022 23:19:58	16/10/2022 23:19:59	Recreate	
1665849120	Le Duy Hao	000002	collector	C21	012.210	019	15/10/2022 22:52:00	doing	45	No need to rush	16/10/2022 23:53:23		Recreate	Cancel
1665849127	Le Duy Hao	000002	collector	C21	321.123.444	019	15/10/2022 22:52:07	scheduled	0	ASAP	22/10/2022 23:36:11		Recreate	Cancel
1665849134	Le Duy Hao	000002	collector	C21	987.654.789	019	15/10/2022 22:52:14	scheduled	0	You must hurry up	22/10/2022 23:36:11		Recreate	Cancel
1665849181	Pham Quang Son	000004	janitor	J12	321	929	15/10/2022 22:53:01	done	100	Nothing	15/10/2022 23:34:14	16/10/2022 11:22:36	Recreate	
1665849186	Pham Quang Son	000004	janitor	J12	456	929	15/10/2022 22:53:06	doing	50	Clear it carefully	17/10/2022 10:36:11		Recreate	Cancel
1665849196	Pham Quang Son	000004	janitor	J12	789	929	15/10/2022 22:53:16	scheduled	0		27/10/2022 10:36:11		Recreate	Cancel
1665849200	Pham Quang Son	000004	janitor	J12	987	929	15/10/2022 22:53:20	scheduled	0	Clear it carefully	27/10/2022 23:36:11		Recreate	Cancel
1665849207	Dang Van Vinh	000005	janitor	J12	123	333	15/10/2022 22:53:27	done	100	As soon as possible	15/10/2022 23:34:14	16/10/2022 11:22:36	Recreate	
1665849207	Dang Van Vinh	000005	janitor	J12	123	929	15/10/2022 22:53:30	doing	20	Clean it carefully	22/10/2022 23:36:11		Recreate	Cancel

Previous 1 2 Next

## Create task

Tạo nhiệm vụ mới UWC 2.0

https://www.uwc20.com.vn/create\_task

TẠO NHIỆM VỤ MỚI

Back Next

Đăng xuất

Vai trò

☒ Collector ☐ Janitor

Các MCP đã chọn

MCP 123

Xóa MCP

Mô tả

Nhập mô tả

Thời gian bắt đầu

08/01/2022 01:55 CH

Nhân viên đã chọn

Tải danh sách

Xóa

200. MCP: 123

Sức chứa: 150kg

Tải trọng hiện tại: 115kg

Tỉ lệ: 76%

Cập nhật lần cuối: 16/10/2022, 10:26:09

Select

## Assign vehicle



UWC20

https://www.uwc20.com.vn/assign\_vehicle

PHÂN CÔNG PHƯƠNG TIỆN

Back

Next

Vai trò	Janitor
Nhân viên đã chọn	000002

Show 10 entries

Search: 000005

Tên phương tiện	ID	Employee đang được phân công	Employee đã sử dụng lần cuối	Thời điểm chỉ định	Ghi chú	Thể tích (m3)	Đang được sử dụng?	Task ID đang sử dụng		
Troller 150	333	Dang Van Vinh (000005)	Pham Quang Son (000004)	15/10/2022, 22:53:30	bad wheels	350	True	1665849210	Assign	Detail

Showing 1 to 1 of 1 entries

Previous

1

Next

## Routing

UWC 2.0

https://www.uwc.com

Định Tuyến Đường

Back

Finish

Đăng xuất

Vai trò	Janitor
MCP đã chọn	321
Mô tả	
Thời gian bắt đầu	13:55:00, 08/01/2022
Nhân viên đã chọn	000004

Điểm bắt đầu

Điểm đến

qua Alexandre de Rhodes, đường trục chính 10  
7 min  
2.2 km

qua Alexandre de Rhodes, đường trục chính 12  
20 min  
1.6 km

7 min (2.2 km)  
qua Alexandre de Rhodes/  
đường trục chính 10

Điểm bắt đầu  
Đi về hướng Tây Nam  
Qua Trạm Xe buýt Bách Khoa  
Xe 50 (ở phía bên phải)  
200 m

Rẽ trái vào Đường Tạ Quang Bửu  
Qua A14 - Đồng Nai (ở phía bên phải)  
400 m

Rẽ phải vào Đ. Thánh Gióng  
Điểm kết thúc