

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



# **SOFTWARE ENGINEERING (CO3001)**

---

## **ASSIGNMENT: The UWC 2.0**

### **Final Report**

---

Instructor:	Quản Thành Thơ	
Class:	CC02	
Group:	QWERTY	
Members:	Nguyễn Duy Bảo	2052399
	Nguyễn Đức Danh	2052904
	Trần Công Huy Hoàng	2052482
	Hoàng Đình Huy	2053034
	Nguyễn Tân Lộc	2053197

HO CHI MINH CITY, November 2022



## Contribution

No.	Name	Student Id	Contribution	Workload
1	Nguyễn Duy Bảo	2052399	28%	Leader, do the main part of each Task.
2	Trần Công Huy Hoàng	2052482	24%	Big contribution in Design part (Task 1 to Task 4)
3	Nguyễn Tấn Lộc	2053197	21%	Big contribution in Implementation part (Task 5)
4	Hoàng Đình Huy	2053034	15%	Supporter in each Task
5	Nguyễn Đức Danh	2052904	12%	Supporter in each Task



# CONTENT

<b>1 TASK 1: Requirement elicitation</b>	<b>3</b>
1.1 Introduction	3
1.1.1 Identify the context of this project	3
1.1.2 Who are the relevant stakeholders?	3
1.1.3 What are their current needs?	3
1.1.4 What could be their current problem?	4
1.1.5 What benefits the UWC 2.0 will be for each stakeholder?	5
1.2 Requirements	6
1.2.1 Functional Requirements	6
1.2.2 Non-functional Requirements	7
1.2.3 Use-case Diagram for the whole system	8
1.3 Use-case Diagram and Description	11
1.3.1 Log in/Log out	11
1.3.2 Employee Management	14
1.3.3 Task Management	19
1.3.4 Message	28
1.3.5 Notification	31
1.3.7 Language	32
<b>2 TASK 2: System Modelling</b>	<b>34</b>
2.1 Activity Diagram	34
2.1.1 Check Calendar	34
2.1.2 Update MCP	35
2.1.3 View Task	36
2.1.4 Manage Task	37
2.1.5 Manage Facility	38
2.1.6 Message	39
2.1.7 Notification	40
2.2 Sequence Diagram	41
2.3 Class Diagram	42
<b>3 TASK 3: Architecture Design</b>	<b>43</b>
3.1 Describe the architectural approach	43
3.1.1 Architecture Approach	43
3.1.2 For the whole UWC 2.0 system, we plan to design 6 modules:	45
3.2 Draw an implementation diagram for Task Assignment module	46



<b>4 TASK 4: Implementation - Sprint 1</b>	<b>48</b>
4.1 Online repository (GitHub)	48
4.2 Adding documents, materials, and folders	48
4.3 Implement MVP1 – design an interface	49
4.3.1 Login Page	49
4.3.2 Task Management Dashboard	49
4.3.3 Manage Employee	50
4.3.4 Manage Task	51
4.3.5 Manage Facility	52
4.3.6 Message	53
4.3.7 Notification	54
4.3.8 Setting	54
<b>5 TASK 5: Implement - Sprint 2</b>	<b>55</b>



## 1 TASK 1: Requirement elicitation

### 1.1 Introduction

#### 1.1.1 Identify the context of this project

- Along with economic development, urban waste management is one of the biggest challenges we face today, especially in big cities. The authorities need to come up with solutions to solve this problem in an efficient, cost-effective way to achieve the development goals set out in SDG 11 and SDG 6. This is the reason for the development of the UWC 1.0 information management system.
- UWC 1.0 is a decent system but there are still many drawbacks, thus after 5 years, UWC 2.0 is developed.
- UWC 2.0 is a more well-rounded developed version than UWC 1.0. With UWC 2.0, Back officers, Collectors, and Janitors can manage information and work schedules more easily and efficiently; At the same time, it also supports the management of vehicle information and waste collection points, thereby providing an efficient and cost-effective operation solution. UWC 2.0 also integrates a messaging function that allows contact information to be updated quickly and promptly to facilitate the resolution of arising problems.

#### 1.1.2 Who are the relevant stakeholders?

- Back officers
- Janitors
- Collectors

#### 1.1.3 What are their current needs?

- As a Back officer, I need to have an overview of Janitors' and Collectors' work calendars.
- As a Back officer, I need to have an overview of vehicles and their technical details (including weight, capacity, fuel consumption, ...)
- As a Back officer, I need an overview of all MCPs (major collecting points) and information about their capacity. MCPs' data should be



updated every 15 minutes with the availability of at least 95% of their operating time.

- As a Back officer, I need to assign vehicles to Janitors and Collectors.
- As a Back officer, I need to be able to assign Janitors and Collectors to MCPs.
- As a Back officer, I need to create a route for each Collector. The route should be optimized in terms of fuel consumption and travel distance.
- As a Back officer, I need to be able to send messages to Collectors and Janitors.
- As a Collector/Janitor, I need to have an overview of my work calendar.
- As a Collector/Janitor, I need to have an overview of my tasks on a daily and weekly basis.
- As a Collector/Janitor, I need to be able to communicate with other Collectors, Janitors, and Back officers.
- As a Collector/Janitor, I need to be able to check in / check out every day.
- As a Collector/Janitor, I need to be notified about MCPs' capacity.

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

##### Employee

- Hard to manage employees (Collectors/Janitors):
  - Hard to check daily and weekly tasks
  - Hard to manage the human resource
- Not having an accessible working calendar for Collectors/Janitors, Back officer:
  - Can not effectively organize and distribute the tasks to staff
  - Collectors/Janitors' productivity may be decreased

##### Technical equipment

- Can not track technical equipment status (collecting vehicles, trollers)
- Not having a real-time means of communication:
  - Can not handle spontaneous problems



- Delay information and report about the progress from Collectors/Janitors to Back officer

#### Working progress

- Not having an optimized route or path for J/C:
  - Wasting the resources (gasoline, time, ...)
  - Duplicate the MCPs or collect path
- Not having a real-time management system:
  - Hard to track the working progress of J/C
  - Hard to keep records of working progress

#### 1.1.5 What benefits the UWC 2.0 will be for each stakeholder?

- Back officers can manage and retrieve information from Collectors and Janitors rapidly.
- Back officers can follow the status of all vehicles and Major collecting points (MCPs), then can make an efficient operation and cost-reduction.
- Back officers, Collectors, and Janitors can observe their working calendar easily, thus avoiding time-consuming and improving efficiency.
- Having a high response means of communication will help to solve a sudden problem quickly and in a timely manner.



## 1.2 Requirements

### 1.2.1 Functional Requirements

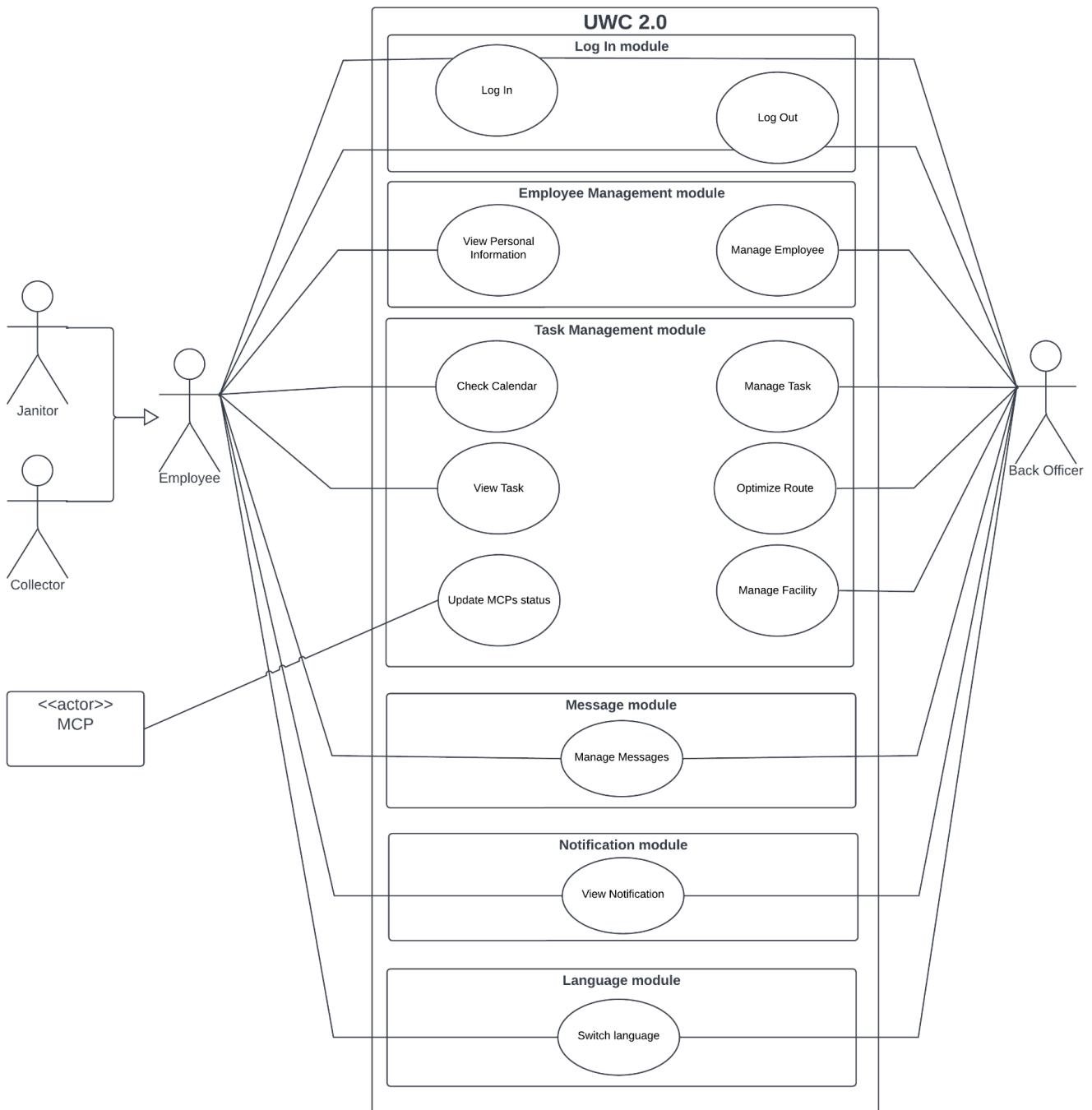
Back officers	<ul style="list-style-type: none"><li>- Be able to use an admin account to access the software.</li><li>- Be able to manage and retrieve employee information (resumes, number of days off, ...).</li><li>- Be able to manage and edit the work schedule of collectors and janitors.<ul style="list-style-type: none"><li>+ Assign janitors to each area..</li><li>+ Create logical routes and notify collectors.</li></ul></li><li>- Be able to manage the status of all vehicles (load, registration time, fuel, ...)<ul style="list-style-type: none"><li>+ Information about vehicles (fuel consumed and distance traveled) must be updated every day.</li></ul></li><li>- Be able to assign vehicles to collectors and janitors.</li><li>- AA tool to help calculate a reasonable route for collectors based on fuel quantity and distance.</li><li>- Be able to manage the status of MCPs (updated every 15 minutes).</li><li>- A communication channel with collectors and janitors.</li></ul>
Collectors and Janitors	<ul style="list-style-type: none"><li>- Be able to use an employee account to access the software.</li><li>- Be able to manage and edit personal information on the system.</li><li>- Be able to track work schedule, daily and weekly tasks in detail on one page (no need to scroll down)</li><li>- A communication channel.</li><li>- A system for daily attendance checks and task reports.</li><li>- Be able to send feedback about the status of MCPs.</li></ul>



### 1.2.2 Non-functional Requirements

Performance	<ul style="list-style-type: none"><li>- Website applications have a response speed of fewer than 2 seconds.</li><li>- Information traffic, data latency less than 1 second (Message system, MCPs' status feedback system).</li><li>- System response time must be less than 1 second.</li><li>- Be able to handle feedback from 100 users concurrently.</li></ul>
Ease of use	<ul style="list-style-type: none"><li>- Be able to use all functions of the application proficiently after 30 minutes of training.</li><li>- Language alteration (Vietnamese - English).</li><li>- User manual in every application.</li></ul>
Reliabilities	<ul style="list-style-type: none"><li>- The average number of failable real-time access to the system is 2 out of 1000 access times.</li><li>- The system must operate 24/7 and the inaccessible time must not exceed 15 minutes during the operation time (6:00 -&gt; 21:00) and no more than 30 minutes outside the operating time.</li></ul>
Security	<ul style="list-style-type: none"><li>- System grants access and privileges to accounts based on account types (Back officer and Employee)</li><li>- Warning for server intrusion.</li><li>- Preventing DDoS attacks.</li></ul>

### 1.2.3 Use-case Diagram for the whole system





- TABLE OF ACTORS:

No.	Actor
1	Back officer
2	Collector
3	Janitor

- TABLE OF USE-CASES:

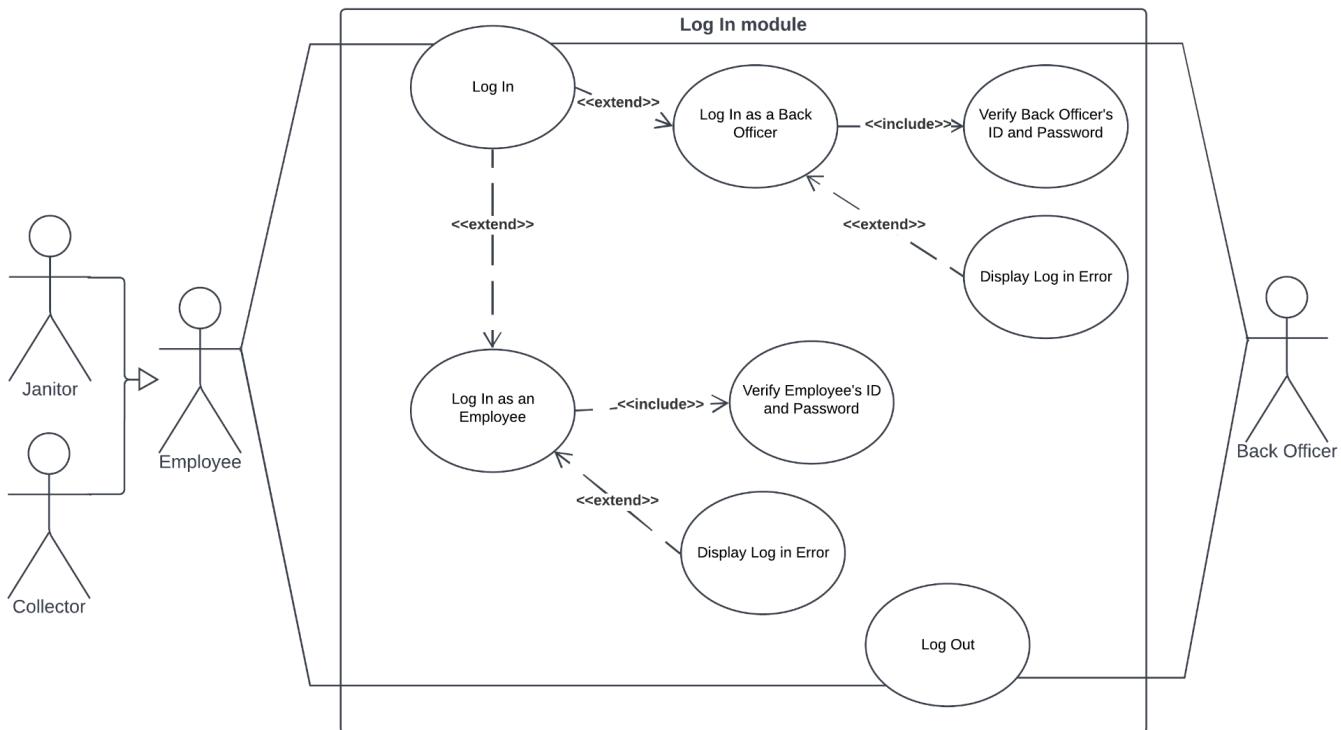
Use-case ID	Use-case name	Description
1	Log in	Collector, Janitor, Back Officer use to log into the UWC 2.0
2	Log out	Collector, Janitor, Back Officer use to log out the UWC 2.0
3	View Personal Information	Collector, Janitor can have an overview of their information.
4	Manage Employee	The Back Officer can add or remove an employee, as well as manage their information (such as their schedule or personal information).
5	Check Calendar	Collector and Janitor can have an overview of their work calendar.
6	Update MCPs status	MCP module can update the status of Major Collecting Points to the system.
7	View Task	Collector and Janitor can have a detail view of their daily tasks.



8	Manage Task	The Back Officer can manage an employee's tasks (such as add new tasks, edit tasks, remove tasks, track task progress).
9	Manage Facility	The Back Officer manages the status of MCPs and vehicles.
10	Manage Messages	Collector, Janitor and Back Officer can manage messages, including: Send Message, Receive Message and View Message.
11	View Notification	Collector, Janitor and Back Officer can see the list of Notification from “Update Notification” use-case.
12	Switch Language	Collector/Janitor can change language between English and Vietnamese.

## 1.3 Use-case Diagram and Description

### 1.3.1 Log in/Log out



<b>Use-case Name</b>	<b>Log In</b>
Actor	Collector, Janitor, Back Officer
Description	Collector, Janitor, Back Officer can log into the UWC 2.0.
Preconditions	1. Collector, Janitor, Back Officer have an UWC 2.0 account. 2. The system is connected to the internet and opened.
Normal Flow	1. Display the log-in interface. 2. User clicks the log-in button. The system display 2 options: + Log in as a Back Officer + Log in as an Employee (Collector, Janitor) 3. Collector, Janitor, Back Officer choose one of 2 above options.



	4. The system switches into one of 2 above options' display.
Exceptions	None
Alternative Flows	Alternative Flow 1: at Step 3 3a. Collector, Janitor, Back Officer choose “Log in as a Back Officer” and go to the use-case “Log-in as a Back Officer”. 3b. Collector, Janitor, Back Officer choose “Log in as an Employee” and go to the use-case “Log-in as an Employee”.

<b>Use-case Name</b>	<b>Log In as a Back Officer</b>
Actor	Collector, Janitor, Back Officer
Description	Collector, Janitor, Back Officer enter their ID and password to log into the UWC 2.0 Back Officer's system.
Preconditions	Collector, Janitor, Back Officer choose “Log in as a Back Officer” and go to the use-case “Log-in as a Back Officer”.
Normal Flow	1. The system requests that the Collector, Janitor, Back Officer enter their ID and password. 2. The Collector, Janitor, Back Officer enter their ID and password. 3. The system validates the entered ID and password. 4. The user gains access to the UWC 2.0 Back Officer's system.
Exceptions	Exception 1: at step 3 <ul style="list-style-type: none"><li>- The Collector, Janitor, Back Officer haven't entered anything in the ID and password form.</li><li>- The system requires the Collector, Janitor, Back Officer to enter their ID and password.</li></ul> Exception 2: at step 3 <ul style="list-style-type: none"><li>- The Collector, Janitor, Back Officer entered a wrong ID</li></ul>



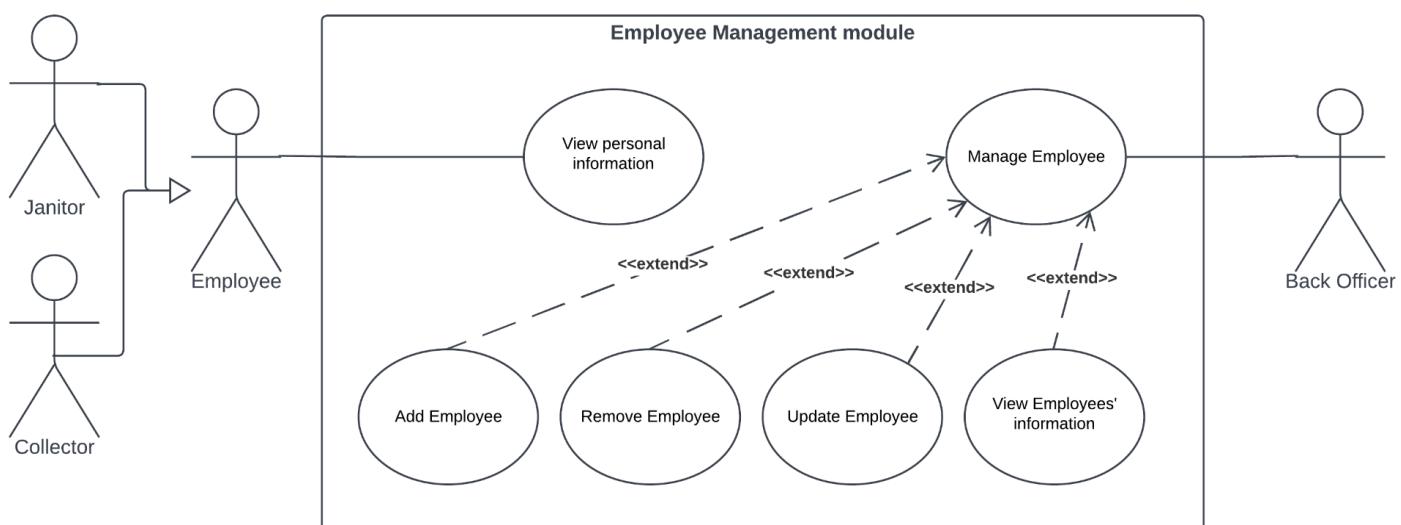
	<p>or a wrong password.</p> <ul style="list-style-type: none"><li>- The system displays a Log-in error and requires the Collector, Janitor, Back Officer to enter their ID and password again.</li></ul>
Alternative Flows	None

<b>Use-case Name</b>	<b>Log In as an Employee</b>
Actor	Collector, Janitor, Back Officer
Description	Collector, Janitor, Back Officer enter their ID and password to log into the UWC 2.0 Employee's system
Preconditions	Collector, Janitor, Back Officer choose "Log in as an Employee" and go to the use-case "Log-in as an Employee".
Normal Flow	<ol style="list-style-type: none"><li>1. The system requests that the Collector, Janitor, Back Officer enter their ID and password.</li><li>2. The Collector, Janitor, Back Officer enter their ID and password.</li><li>3. The system validates the entered ID and password.</li><li>4. The user gains access to the UWC 2.0 Employee's system.</li></ol>
Exceptions	<p>Exception 1: at step 3</p> <ul style="list-style-type: none"><li>- The Collector, Janitor, Back Officer haven't entered anything in the ID and password form.</li><li>- The system requires the Collector, Janitor, Back Officer to enter their ID and password.</li></ul> <p>Exception 2: at step 3</p> <ul style="list-style-type: none"><li>- The Collector, Janitor, Back Officer entered a wrong ID or a wrong password.</li><li>- The system displays a Log-in error and requires the Collector, Janitor, Back Officer to enter their ID and</li></ul>

	password again.
Alternative Flows	None

Use-case Name	Log Out
Actor	Collector, Janitor, Back Officer
Description	Collector, Janitor, Back Officer use to log out the UWC 2.0
Preconditions	Collector, and Janitorhave logged in into the UWC 2.0 Employee's system.
Normal Flow	1. Collector, Janitor, Back Officer click the “Log out”. Return to the “Log in” use-case.
Exceptions	None
Alternative Flows	None

### 1.3.2 Employee Management





Use-case Name	View personal information
Actor	Collector, Janitor
Description	Collector, Janitor can have an overview of their information
Preconditions	Collector, Janitor have logged in to the UWC 2.0 Employee's system.
Normal Flow	<ol style="list-style-type: none"><li>1. Collector or Janitor click the "Personal information" button.</li><li>2. The system displays the personal information of the Collector or Janitor.</li><li>3. Collector or Janitor finish viewing their information by closing the display.</li><li>4. The system returns to the main page.</li></ol>
Exceptions	None
Alternative Flows	None

Use-case Name	Manage Employee
Actor	Back Officer
Description	Back Officer can add or remove an employee, as well as manage their information (such as their schedule or personal information)
Preconditions	The Back Officer must log in with a back Officer account and press the "Manage Employee" button.
Normal Flow	<ol style="list-style-type: none"><li>1. Display the Employee Management interface with a menu of 4 options:<ul style="list-style-type: none"><li>- Add Employee</li><li>- Delete Employee</li><li>- Update Employee</li><li>- View Employees' information</li></ul></li><li>2. The Back Officer chooses one of 4 above options.</li><li>3. The system switches into the chosen option interface.</li><li>4. Back Officer finishes Employee Management..</li><li>5. Back to the home screen.</li></ol>
Exceptions	None



Alternative Flows	<p>Alternative Flow 1: at Step 2</p> <p>2a. The Back Officer chooses “Add Employee” and goes to the use-case “Add Employee”.</p> <p>2b. The Back Officer chooses “Remove Employee” and goes to the use-case “Remove Employee”.</p> <p>2c. The Back Officer chooses “Update Employee” and goes to the use-case “Update Employee”.</p> <p>2d. The Back Officer chooses “View Employees' information” and goes to the use-case “View Employees' information”.</p>
-------------------	--

<b>Use-case Name</b>	<b>Add Employee</b>
Actor	Back Officer
Description	The Back Officer can create a new account for the new employee.
Preconditions	The Back Officer chooses “Add Employee” and goes to the use-case “Add Employee”.
Normal Flow	<ol style="list-style-type: none"><li>1. The system shows a new employee sign up interface.</li><li>2. The Back Officer fills in information of the new employee.</li><li>3. The Back Officer confirms information and saves it to the system.</li><li>4. Back to Manage Employee screen.</li></ol>
Exceptions	<p>Exception 1: at step 2</p> <ul style="list-style-type: none"><li>- The system displays an error when the account ID is invalid or has been used.</li></ul> <p>Exception 2: at step 3</p> <ul style="list-style-type: none"><li>- The back Officer misses some mandatory information or some information is invalid.</li><li>- The system shows an error and requires the back Officer</li></ul>



	to fill in the missing information.
Alternative Flows	None

<b>Use-case Name</b>	<b>Remove Employee</b>
Actor	Back Officer
Description	The Back Officer can remove an old employee's account.
Preconditions	The Back Officer chooses “Remove Employee” and goes to the use-case “Remove Employee”.
Normal Flow	<ol style="list-style-type: none"><li>1. The system shows a list of existing accounts.</li><li>2. The Back Officer chooses an account and confirms.</li><li>3. The account is removed from the system.</li><li>4. Back to Manage Employee screen.</li></ol>
Exceptions	None
Alternative Flows	None

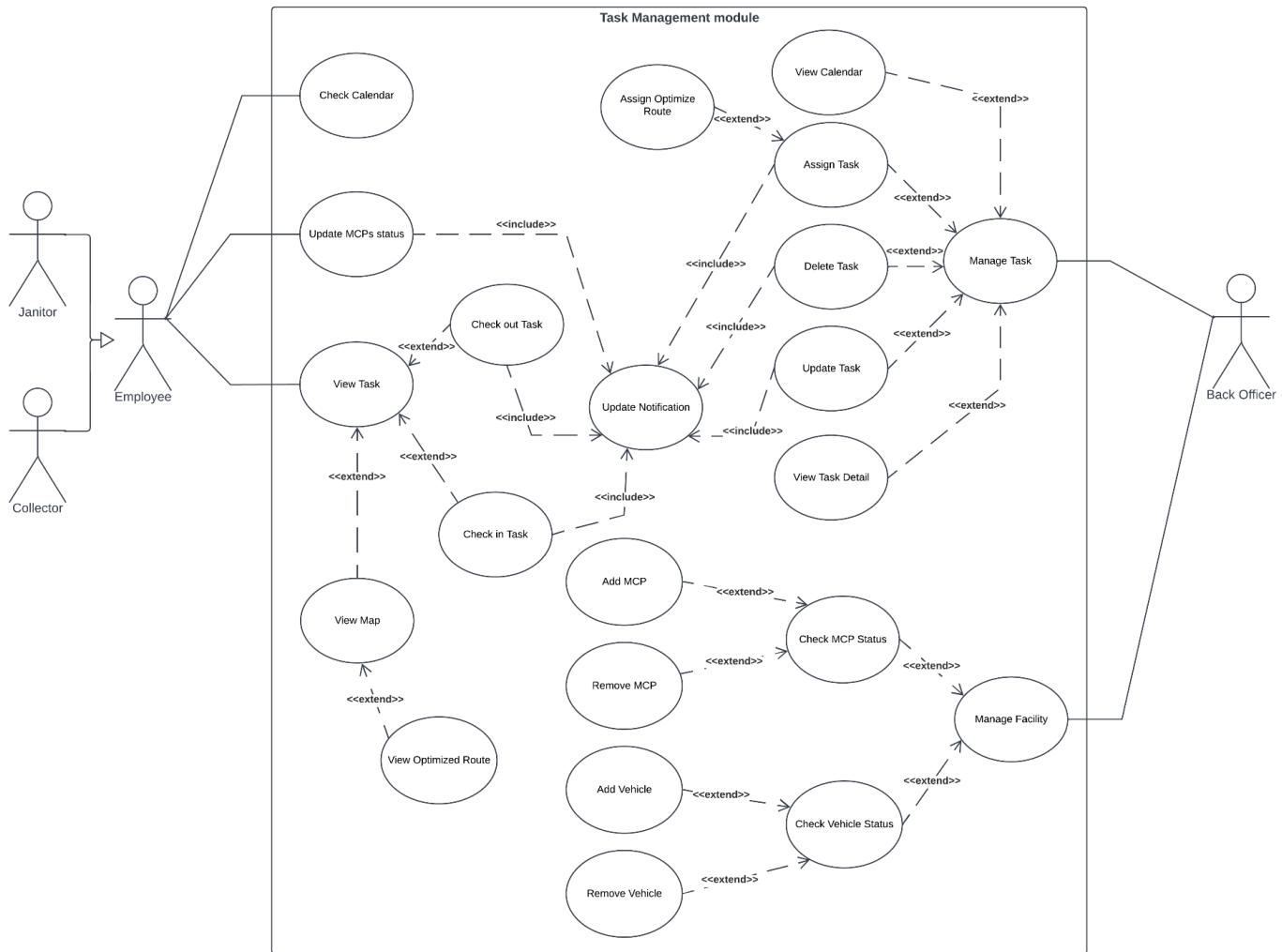
<b>Use-case Name</b>	<b>Update Employee</b>
Actor	Back Officer
Description	The Back Officer can update an employee's information.
Preconditions	The Back Officer chooses “Update Employee” and goes to the use-case “Update Employee”.
Normal Flow	<ol style="list-style-type: none"><li>1. The system shows a list of existing accounts.</li><li>2. The Back Officer chooses an account from the list.</li><li>3. The Back Officer updates information and confirms.</li></ol>



	<p>4. The system updates the account information.</p> <p>5. Back to Manage Employee screen.</p>
Exceptions	<p>Exception 1: at step 3</p> <ul style="list-style-type: none"><li>- The back Officer misses some mandatory information or some information is invalid.</li><li>- The system shows an error and requires the back Officer to fill in the missing information.</li></ul>
Alternative Flows	None

<b>Use-case Name</b>	<b>View Employees' information</b>
Actor	Back Officer
Description	The Back Officer can view an employee's information.
Preconditions	The Back Officer chooses “View Employees' information” and goes to the use-case “View Employees' information”.
Normal Flow	<p>1. The system shows a list of existing accounts.</p> <p>2. The Back Officer chooses an account from the list.</p> <p>3. The system displays the account information.</p> <p>5. Back to Manage Employee screen.</p>
Exceptions	None
Alternative Flows	None

### 1.3.3 Task Management



Use-case Name	Manage Task
Actor	Back Officer
Description	The Back Officer can manage an employee's tasks (such as add new tasks, edit tasks, remove tasks, track task progress).
Preconditions	The Back Officer must log in with a Back Officer account and press the “Manage Task” button.
Normal Flow	1. The system displays the Task Management interface with a list of short information of assigned tasks.



	<p>2. Back Officer choose one of these function:</p> <ul style="list-style-type: none"><li>- The Back Officer chooses “View Calendar” to view the employee's schedule.</li><li>- The Back Officer chooses “Assign Task” to assign a new task to an employee.</li><li>- The Back Officer chooses one or more tasks in the list and chooses “Delete Task” to delete the chosen task(s).</li><li>- The Back Officer chooses a task in the list and chooses “Update Task” to update the chosen task.</li><li>- The Back Officer chooses one task in the list and chooses “View Task Detail” to see the detail of the chosen task.</li></ul> <p>3. The Back Officer chooses one of 5 above options.</p> <p>4. The system switches into the chosen option interface.</p> <p>5. Back Officer finishes Task Management.</p> <p>6. Back to the home screen.</p>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: at Step 2</p> <p>2a. The Back Officer chooses “View Calendar” and goes to the use-case “View Calendar”.</p> <p>2b. The Back Officer chooses “Assign Task” and goes to the use-case “Assign Task”.</p> <p>2c. The Back Officer chooses “Delete Task” and goes to the use-case “Delete Task”.</p> <p>2d. The Back Officer chooses “Update Task” and goes to the use-case “Update Task”.</p> <p>2e. The Back Officer chooses “View Task Detail” and goes to the use-case “View Task Detail”.</p>



Use-case Name	<b>View Calendar</b>
Actor	Back Officer
Description	Back Officer view calendar with one or some employees' schedule.
Preconditions	The Back Officer chooses “View Calendar” and goes to the use-case “View Calendar”.
Normal Flow	<ol style="list-style-type: none"><li>1. The system shows a blank calendar.</li><li>2. The back Officer can choose one or some employees to show their schedule on the calendar.</li><li>3. Back to Manage Task screen.</li></ol>
Exceptions	None
Alternative Flows	None

Use-case Name	<b>Assign Task</b>
Actor	Back Officer
Description	The Back Officer assigns a new task to an employee.
Preconditions	The Back Officer chooses “Assign Task” and goes to the use-case “Assign Task”.
Normal Flow	<ol style="list-style-type: none"><li>1. The Back Officer chooses an employee from the employee list.</li><li>2. The Back Officer writes detailed information and description for the task. If the chosen employee is a collector, show an option to create an optimized route.</li><li>3. The Back Officer confirms to assign the task to an employee.</li><li>4. The system performs “Update Notification” (&lt;&lt;include&gt;&gt;).</li><li>5. Back to Manage Task screen.</li></ol>



Exceptions	<p>Exception 1: at step 2</p> <ul style="list-style-type: none"><li>- The employee assigned the task is offline or has already had a schedule on the task time.</li><li>- The system shows an error and requires the back Officer to choose another employee or change the task time.</li></ul>
Alternative Flows	<p>Alternative Flow 1: at step 2</p> <ul style="list-style-type: none"><li>- The Back Officer chooses the optimize route option and goes to the use-case “Optimize Route”.</li></ul> <p>Alternative Flow 2: at step 2</p> <ul style="list-style-type: none"><li>- The Back Officer does not confirm and the system goes back to the “Manage Task” screen.</li></ul>

Use-case Name	Delete Task
Actor	Back Officer
Description	Back Officer deletes assigned task(s) of an employee.
Preconditions	The Back Officer chooses “Delete Task” and goes to the use-case “Delete Task”.
Normal Flow	<ol style="list-style-type: none"><li>1. The Back Officer confirms to delete the chosen task(s).</li><li>2. The system performs “Update Notification” (&lt;&lt;include&gt;&gt;).</li><li>3. Back to Manage Task screen.</li></ol>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: at step 1</p> <ul style="list-style-type: none"><li>- The Back Officer does not confirm and the system goes back to the “Manage Task” screen.</li></ul>



Use-case Name	Update Task
Actor	Back Officer
Description	The Back Officer updates an assigned task of an employee.
Preconditions	The Back Officer chooses “Update Task” and goes to the use-case “Update Task”.
Normal Flow	<ol style="list-style-type: none"><li>1. The Back Officer replaces the existing detail with the new detail of the chosen task.</li><li>2. The Back Officer confirms to update the chosen task.</li><li>3. The system performs “Update Notification” (&lt;&lt;include&gt;&gt;).</li><li>3. Back to Manage Task screen.</li></ol>
Exceptions	None
Alternative Flows	Alternative Flow 1: at step 2 <ul style="list-style-type: none"><li>- The Back Officer does not confirm and the system goes back to the “Manage Task” screen.</li></ul>

Use-case Name	View Task Detail
Actor	Back Officer
Description	The Back Officer views the chosen task in detail.
Preconditions	The Back Officer chooses “View Task Detail” and goes to the use-case “View Task Detail”.
Normal Flow	<ol style="list-style-type: none"><li>1. The system shows the chosen task in detail.</li><li>2. The Back Officer finishes reading the detailed information.</li><li>3. Back to Manage Task screen.</li></ol>
Exceptions	None
Alternative Flows	None



Use-case Name	Optimize Route
Actor	Back Officer
Description	The Back Officer creates an optimized route and sends it to a collector.
Preconditions	The Back Officer must log in with a Back Officer account and press the “Optimize Route” button.
Normal Flow	<ol style="list-style-type: none"><li>1. The system will automatically create an optimal route in terms of distance.</li><li>2. The Back Officer can edit the optimal route based on real traffic conditions.</li><li>3. The Back Officer confirms the optimal route.</li><li>4. The Back Officer finishes the action.</li><li>5. Back to the “Assign Task” screen.</li></ol>
Exceptions	None
Alternative Flows	None

Use-case Name	Manage Facility
Actor	Back Officer
Description	The Back Officer manages the status of MCPs and vehicles.
Preconditions	The Back Officer must log in with a Back Officer account and press the “Facility Status” button.
Normal Flow	<ol style="list-style-type: none"><li>1. The system displays two options:<ul style="list-style-type: none"><li>- Check MCP Status.</li><li>- Check Vehicle Status.</li></ul></li></ol>



	<ol style="list-style-type: none"><li>2. Back officer chooses one of two options above.</li><li>3. Back officer finishes the action.</li><li>4. Back to the home screen.</li></ol>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: At step 1</p> <ol style="list-style-type: none"><li>1a. Back officer chooses “Check MCP Capacity” and goes to the use-case “Check MCP Status”.</li><li>1b. Back officer chooses “Check Vehicle Status” and goes to the use-case “Check Vehicle Status”.</li></ol>

Use-case Name	Check MCP Status
Actor	Back Officer
Description	The Back Officer can manage the status of MCP.
Preconditions	The Back Officer must log in with the Back Officer account and press the “Facility Status” button, then choose the “Check MCP Status” button.
Normal Flow	<ol style="list-style-type: none"><li>1. The system displays the list of MCPs describing the status of them.</li><li>2. The Back Officer can choose “Add MCP” or “Remove MCP” options.</li><li>3. The Back Officer finishes the action.</li><li>4. Back to the “Manage Facility” screen.</li></ol>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: At step 2:</p> <ol style="list-style-type: none"><li>2a. If the Back Officer chooses “Add MCP”, the system requires new MCP information and updates the new MCP.</li><li>2b. If the Back Officer chooses “Remove MCP”, the system removes the chosen MCP.</li></ol>



Use-case Name	Check Vehicle Status
Actor	Back Officer
Description	The Back Officer can manage the status of trucks, trollers.
Preconditions	The Back Officer must log in with the Back Officer account and press the “Facility Status” button, then choose the “Check Vehicle Status” button..
Normal Flow	<ol style="list-style-type: none"><li>1. The system displays the lists of trucks and trollers and the status of them.</li><li>2. Back officer can choose and have a detailed view of some trucks or trollers.</li><li>3. The Back Officer can choose “Add Vehicle” or “Remove Vehicle” options.</li><li>4. Back officer finishes the action.</li><li>5. Back to the “Manage Facility” screen.</li></ol>
Exceptions	None
Alternative Flows	Alternative Flow 1: At step 3 <ol style="list-style-type: none"><li>3a. If the Back Officer chooses “Add vehicle”, the system requires new vehicle information and updates the new vehicle.</li><li>3b. If the Back Officer chooses “Remove vehicle”, the system removes the chosen vehicle.</li></ol>

Use-case Name	Check Calendar
Actor	Collector, Janitor
Description	Collector and Janitor can have an overview of their work calendar.
Preconditions	Collector and Janitor have logged in to the UWC 2.0 Employee's system.
Normal Flow	<ol style="list-style-type: none"><li>1. Collector or Janitor click the “My Calendar” button.</li><li>2. The system displays their work calendar. There is a filter box for the monthly calendar.</li></ol>



	<ol style="list-style-type: none"><li>3. Collector or Janitor finish viewing their calendar by closing the display.</li><li>4. The system returns to the main page.</li></ol>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: At step 2</p> <p>3a. If the Collector or Janitor wants to have a brief description of their task on a particular day, they can press on that task.</p> <p>3b. If the Collector or Janitor want to view their calendar of a particular month, they can choose in the filter box.</p>

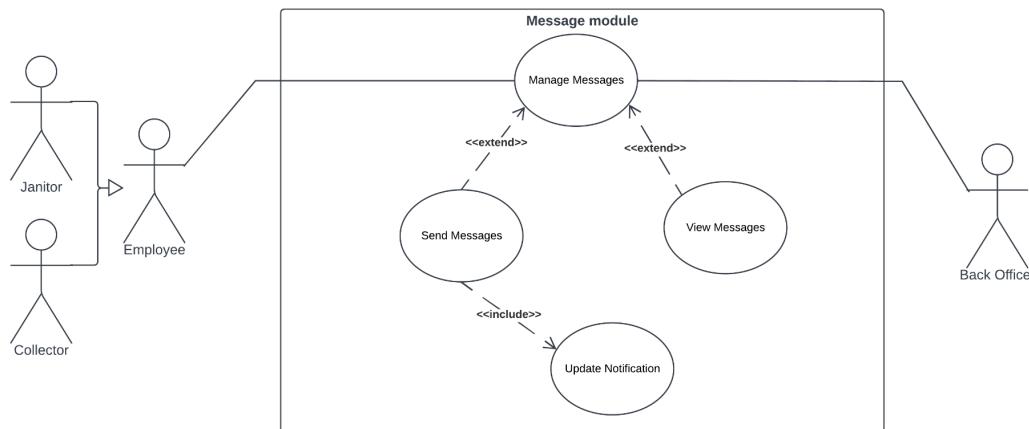
Use-case Name	<b>View Task</b>
Actor	Collector, Janitor
Description	Collector and Janitor can have an overview of today's tasks.
Preconditions	Collector and Janitor have logged in to the UWC 2.0 Employee's system.
Normal Flow	<ol style="list-style-type: none"><li>1. Collector or Janitor click the "My Tasks" button.</li><li>2. The system displays the Collector or Janitor's tasks in one detail view.</li><li>3. If the task is not checked in, the system shows "Check in Task", else shows "View Map" and "Check out Task" options.</li><li>3. Collector or Janitor finish viewing their tasks by closing the display.</li><li>4. The system returns to the main page.</li></ol>
Exceptions	None
Alternative Flows	<p>Alternative Flow 1: At step 3:</p> <p>3a. The task is not checked in and the employee chooses "Check in Task", the system changes the task status and performs "Update Notification" (&lt;&lt;include&gt;&gt;).</p> <p>3b. The task has been checked in, the employee choose "View Map" and goes to the use-case "View Map".</p> <p>3c. The task has been checked in and the employee chooses "Check out Task", the system changes the task status and performs "Update Notification" (&lt;&lt;include&gt;&gt;).</p>

Use-case Name	<b>View Map</b>
Actor	Collector, Janitor
Description	Collector and Janitor can have a detailed view about their route or their working area.
Preconditions	Collector and Janitor choose "View Map" and go to the use-case "View Map".
Normal Flow	<ol style="list-style-type: none"><li>1. The system displays their working area or route in their current task.</li><li>2. Collector can choose to see the optimized route on the map.</li></ol>

	3. The Collector or Janitor finishes viewing the map by closing the display. 4. The system returns to the “View Task” screen.
Exceptions	None
Alternative Flows	None

Use-case Name	Update MCPs status
Actor	MCP
Description	MCP updates its present status.
Preconditions	None
Normal Flow	1. MCP updates its status every time its status changes. 2. The system performs “Update Notification” (<<include>>).
Exceptions	None
Alternative Flows	None.

### 1.3.4 Message



Use-case name	Manage Messages
Actor	Collector, Janitor and Back Officer
Description	Collector, Janitor and Back Officer can communicate in real-time using Message module
Precondition	Collector, Janitor and Back Officer must log in with a valid account for managing messages.
Normal Flow	1. The system will display a list of conversations. 2. For each account, the user can send, and view messages from this



	<p>account.</p> <p>3. User clicks on the desired account.</p> <p>4. User finishes managing the message. There are 2 options:</p> <ul style="list-style-type: none"><li>+ Viewing message from the account.</li><li>+ Sending a message to the account.</li></ul> <p>5. User returns to the main interface by clicking the “Close” button.</p>
Exception	None
Alternative Flow	<p>Alternative 1: at step 4</p> <p>4a. User chooses to view the message.</p> <p>4b. User chooses to send messages.</p>

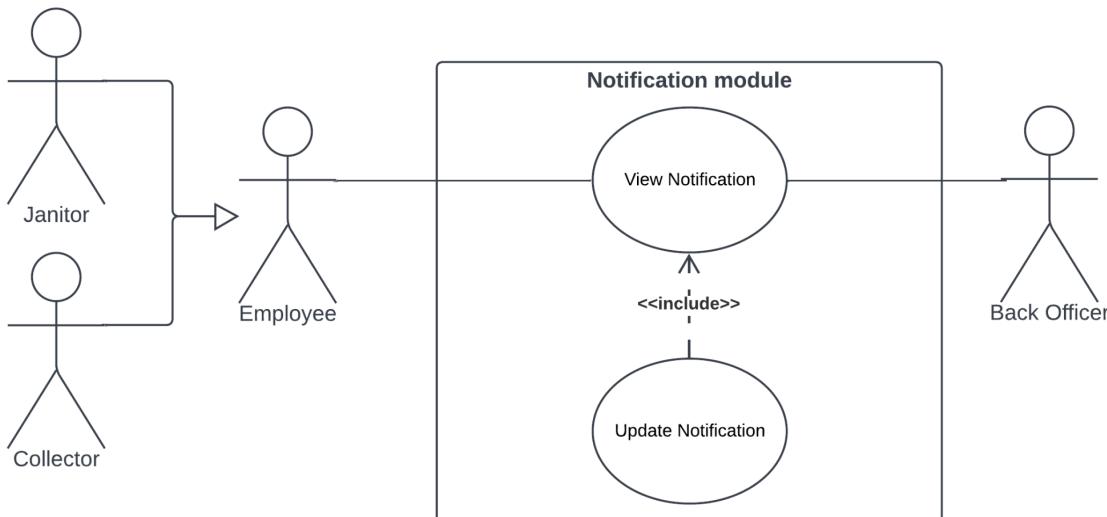
<b>Use-case name</b>	<b>Send Messages</b>
Actor	Collector, Janitor and Back Officer
Description	Collector, Janitor and Back Officer can send messages in the system.
Precondition	Collector, Janitor and Back Officer must log in with a valid account for sending message
Normal Flow	<p>1. User choose the account to send message.</p> <p>2. User clicks on the box displaying “Type a message ...”.</p> <p>3. User types what he/she wants to send to the other account.</p> <p>4. User clicks on the send icon next to the right of the message box.</p> <p>5. Messages which were sent successfully will display a tick next to the message in the chat box.</p> <p>6. User returns to the main interface by clicking the “Close” button.</p>
Exception	None
Alternative Flow	<p>Alternative 1: at step 4</p> <p>4a1. Messages which were failed to send will display a red exclamation mark next to the message in the chat box.</p> <p>4a2. User checks the internet connection.</p> <p>4a3. User refreshes application.</p>



	<p>4a4. User continues step 5 in the Normal Flow.</p> <p>Alternative 2: At Alternative Flow, step 4a3:</p> <p>3a1. User fails to send the message.</p> <p>3a2. User waits for the system to be back online (System might break down).</p> <p>3a3. Check the application every 15 minutes.</p> <p>3a4. If the application is back online, the user continues as the Normal Flow.</p>
--	---

Use-case name	<b>View Messages</b>
Actor	Collector, Janitor and Back Officer
Description	Collector, Janitor and Back Officer can view messages in the system.
Precondition	Collector, Janitor and Back Officer must log in with a valid account for viewing message
Normal Flow	<ol style="list-style-type: none"><li>1. User clicks on the account.</li><li>2. All the past messages will be displayed in the chat box, users can scroll up to view older messages.</li><li>3. User returns to the main interface by clicking the “Close” button.</li></ol>
Exception	None
Alternative Flow	None

### 1.3.5 Notification

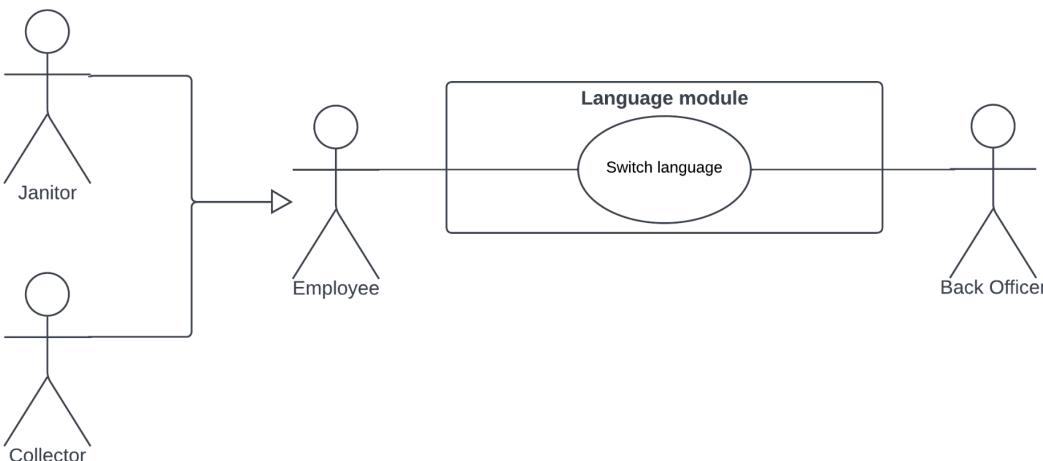


Use-case name	<b>View Notification</b>
Actor	Collector, Janitor and Back Officer
Description	Collector, Janitor and Back Officer can see the list of Notification from “Update Notification” use-case.
Precondition	Collector, Janitor, and Back Officer must log in with a valid account.
Normal Flow	<ol style="list-style-type: none"> <li>1. The system will display a list of notifications.</li> <li>2. User returns to the main interface by clicking the “Close” button.</li> </ol>
Exception	None
Alternative Flow	None

Use-case name	<b>Update Notification</b>
Actor	Collector, Janitor and Back Officer
Description	Collector/Janitor can be notified about Messages, Tasks assigned, deleted, or updated. The Back Officer can be notified about Messages,

	Task's status, MCPs' status.
Precondition	1. Collector, Janitor and Back Officer must have logged in with a valid account. 2. The sender performs a use-case that includes “Update Notification”.
Normal Flow	1. The system adds a notification at the top of the list of “View Notification” of the receiver.
Exception	None
Alternative Flow	None

### 1.3.7 Language



Use-case name	Switch Language
Actor	Collector, Janitor and Back Officer
Description	Collector/Janitor can change language between English and Vietnamese.
Precondition	Collector, Janitor and Back Officer must log in with a valid account.
Normal Flow	1. User clicks on the language button being displayed on the menu bar. 2. System displays two options: + Tiếng Việt. + English. 3. User clicks on the wanted language. The system switches the interface



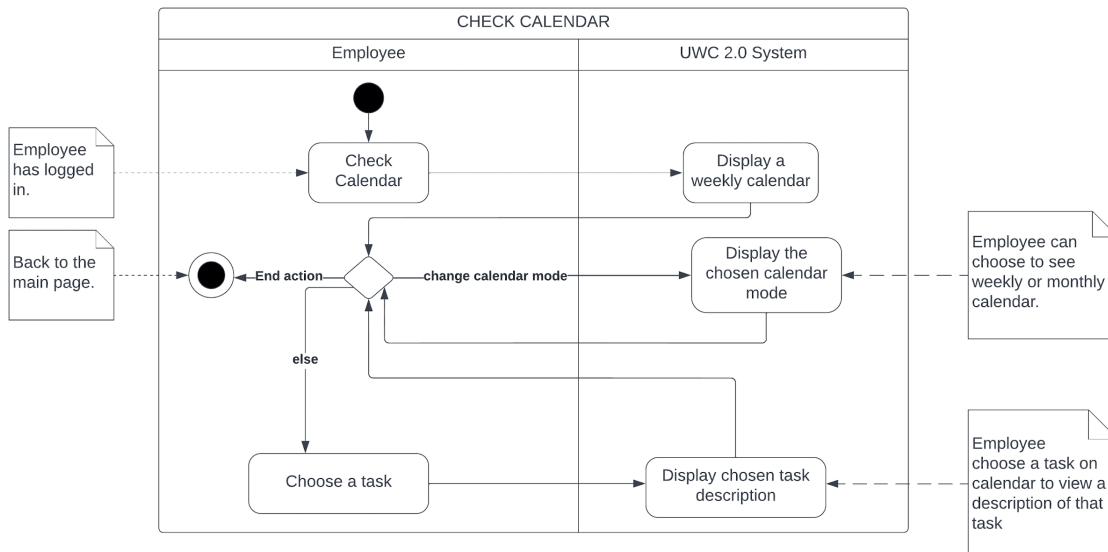
	in the chosen language. 4. User returns to the main interface by clicking the “Close” button.
Exception	None
Alternative Flow	None

## 2 TASK 2: System Modelling

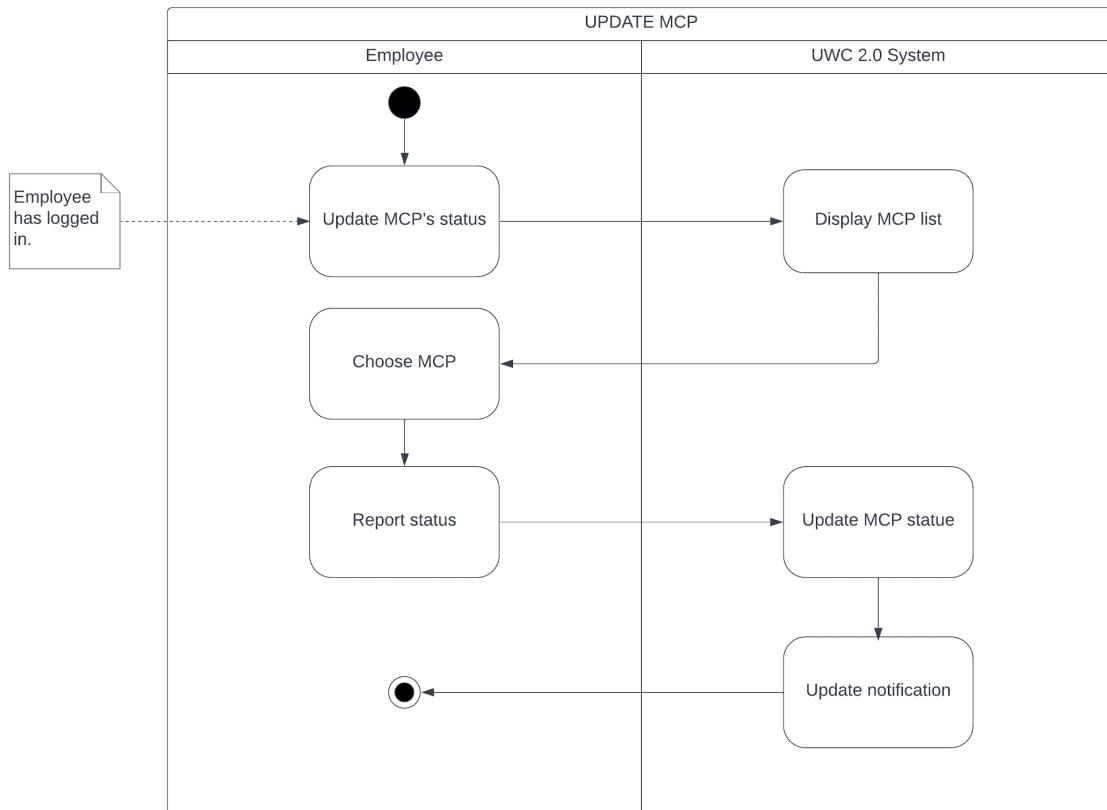
### 2.1 Activity Diagram

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

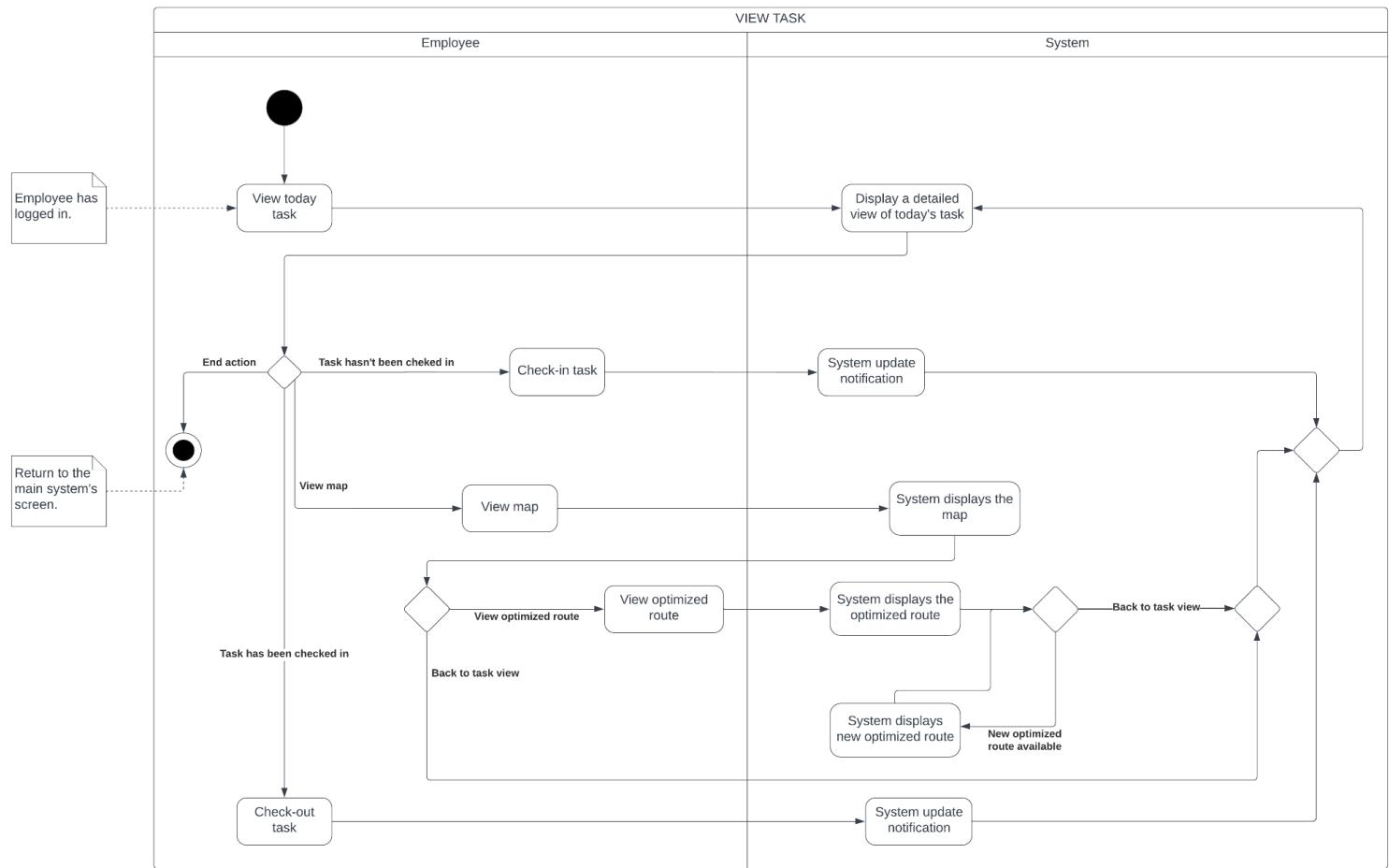
#### 2.1.1 Check Calendar



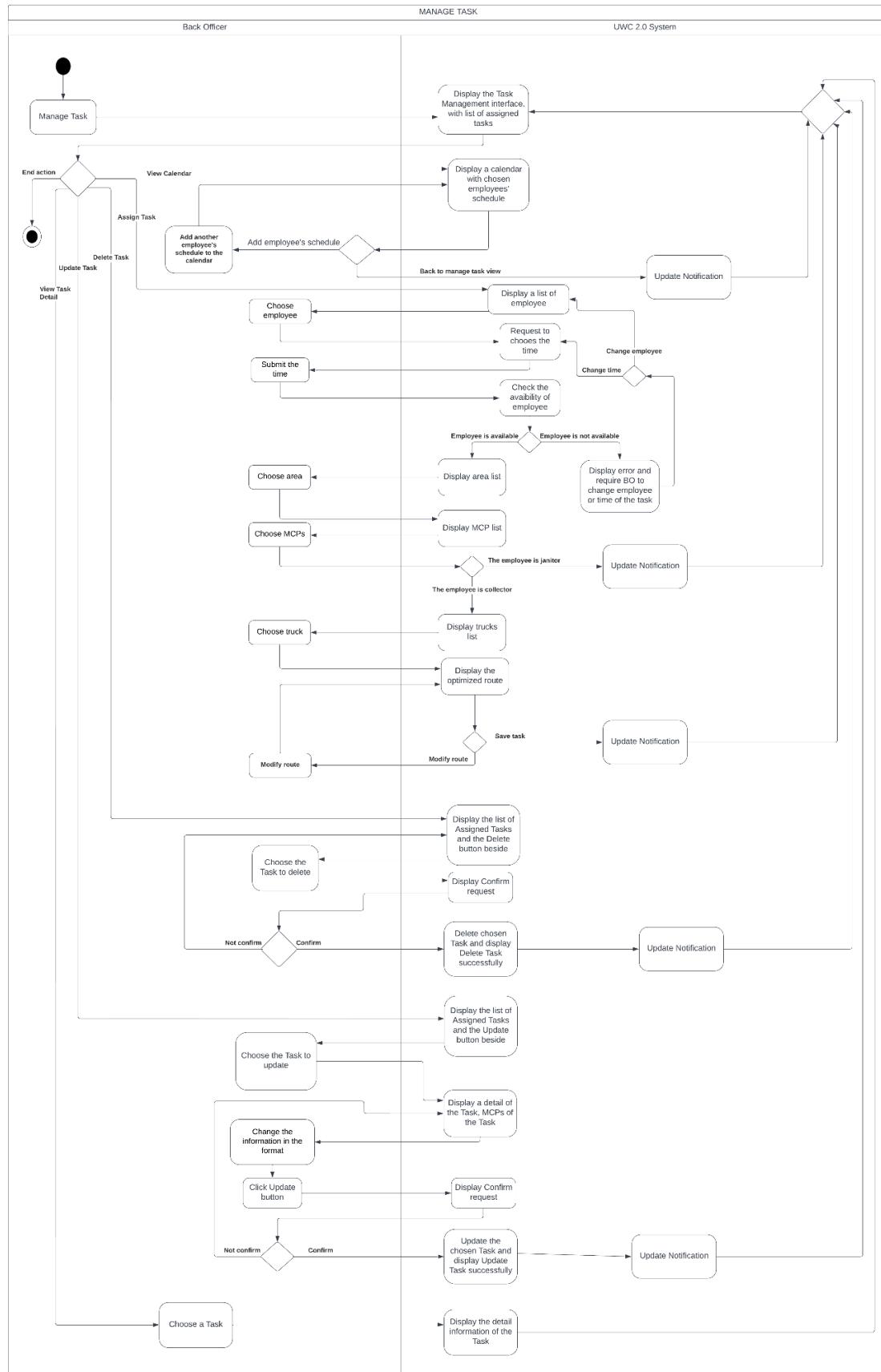
### 2.1.2 Update MCP



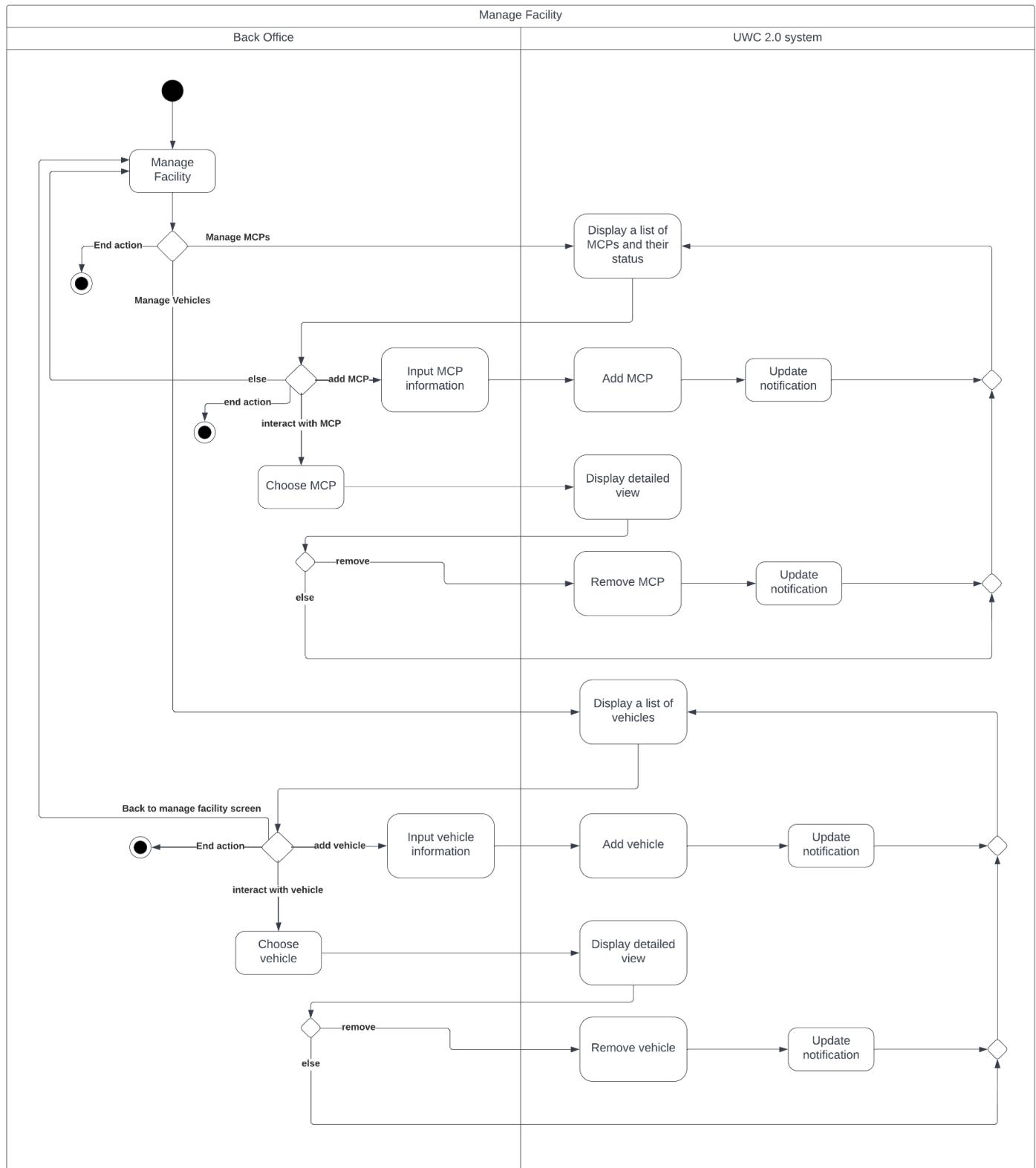
### 2.1.3 View Task



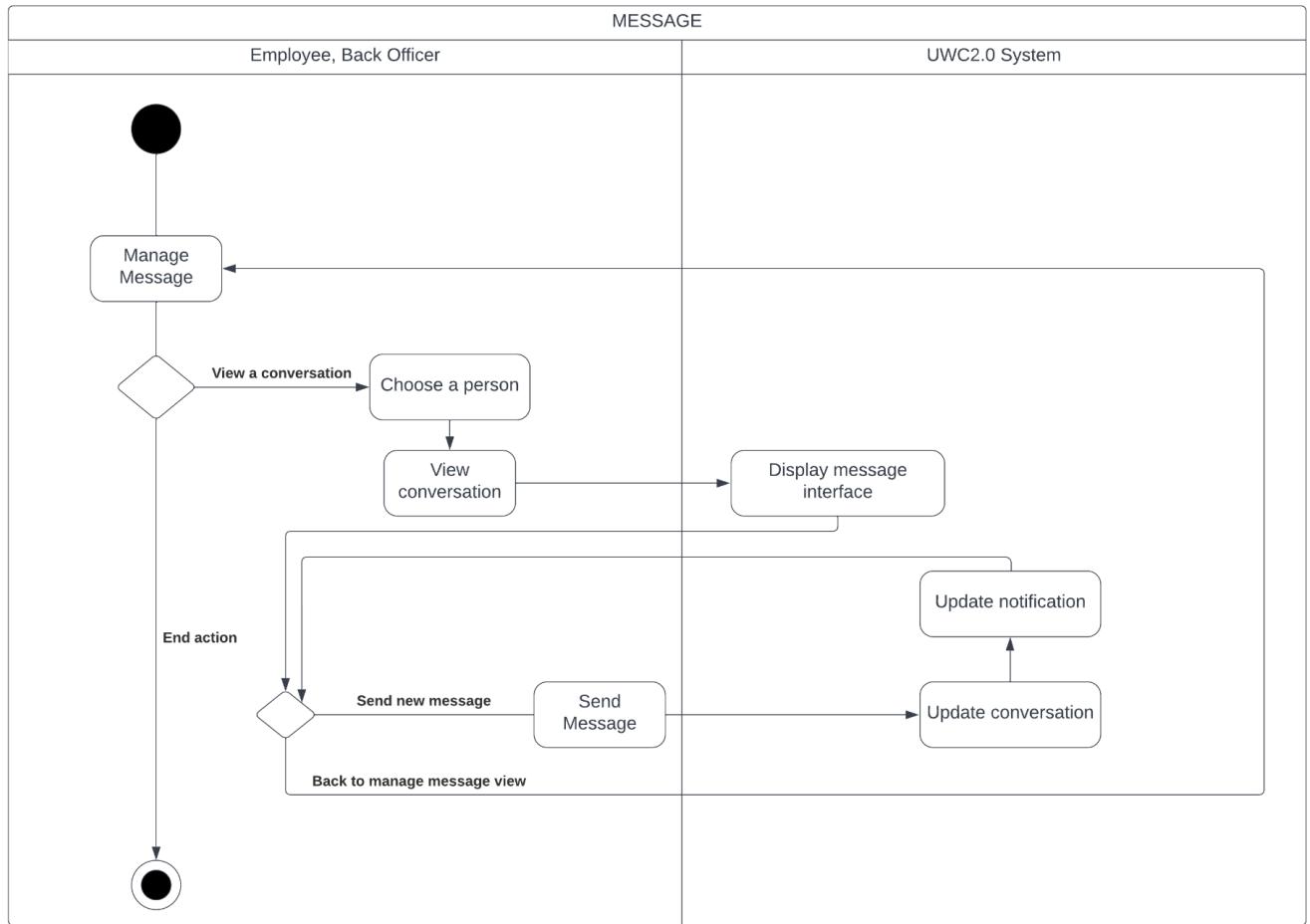
## 2.1.4 Manage Task



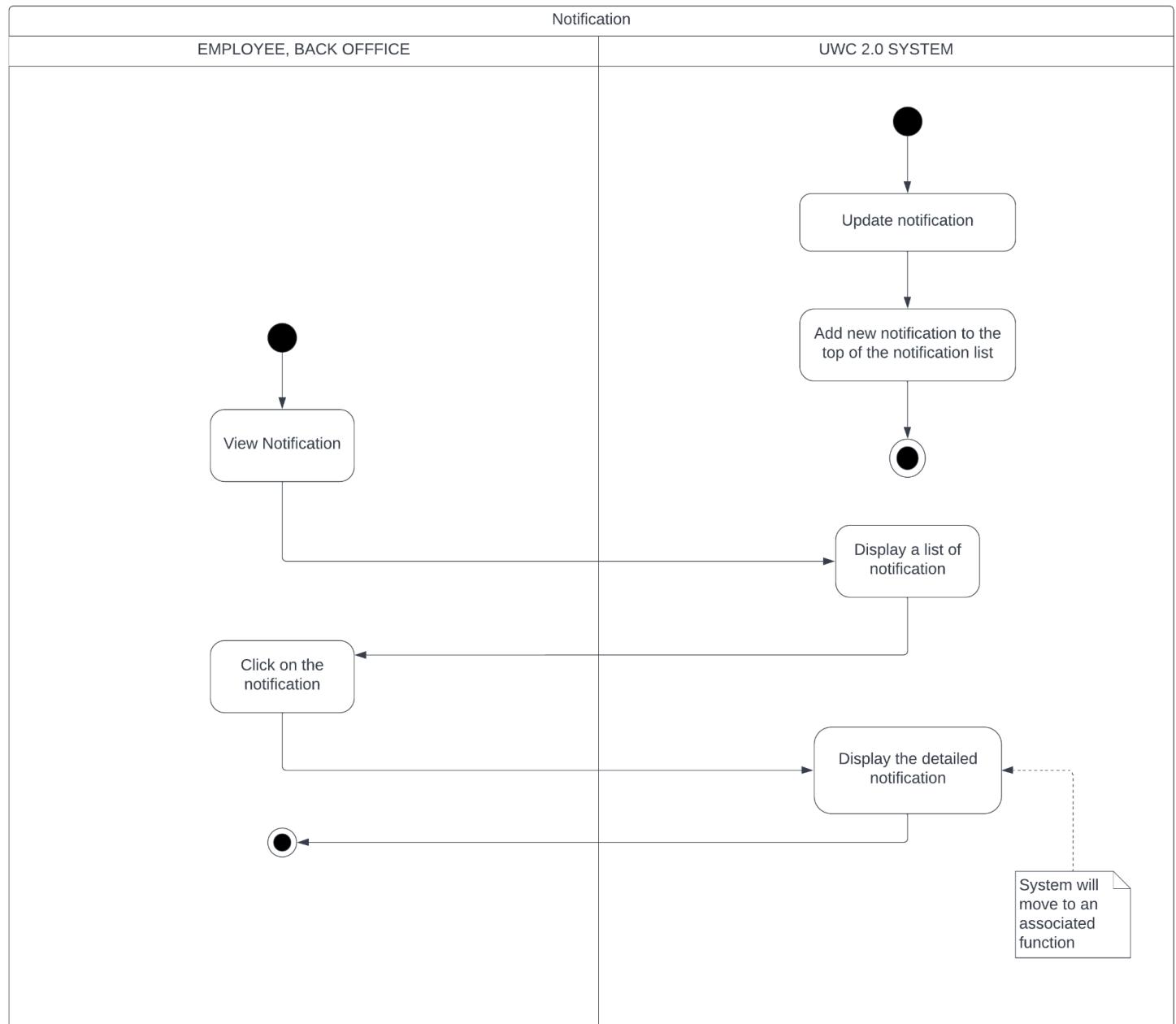
### 2.1.5 Manage Facility



## 2.1.6 Message

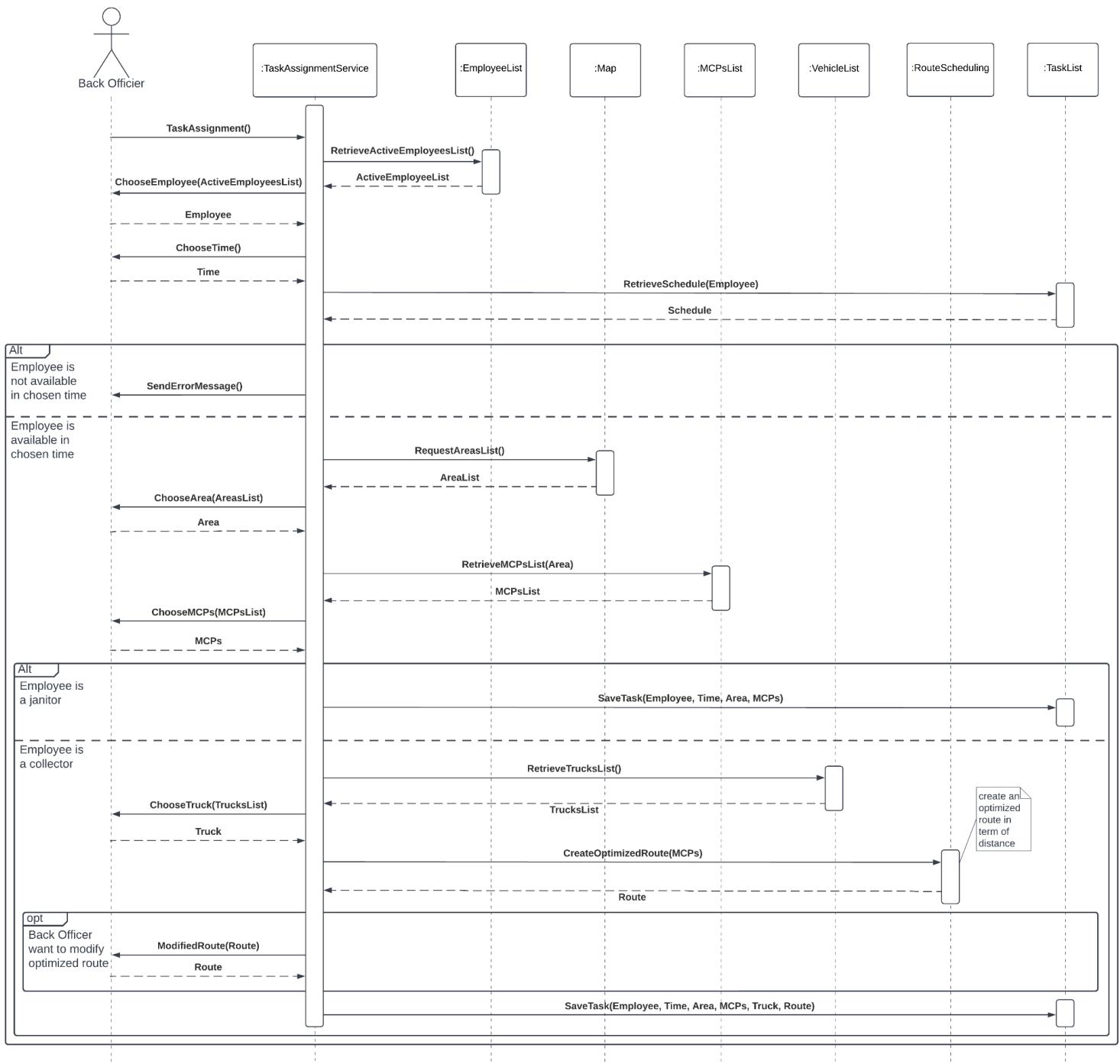


### 2.1.7 Notification



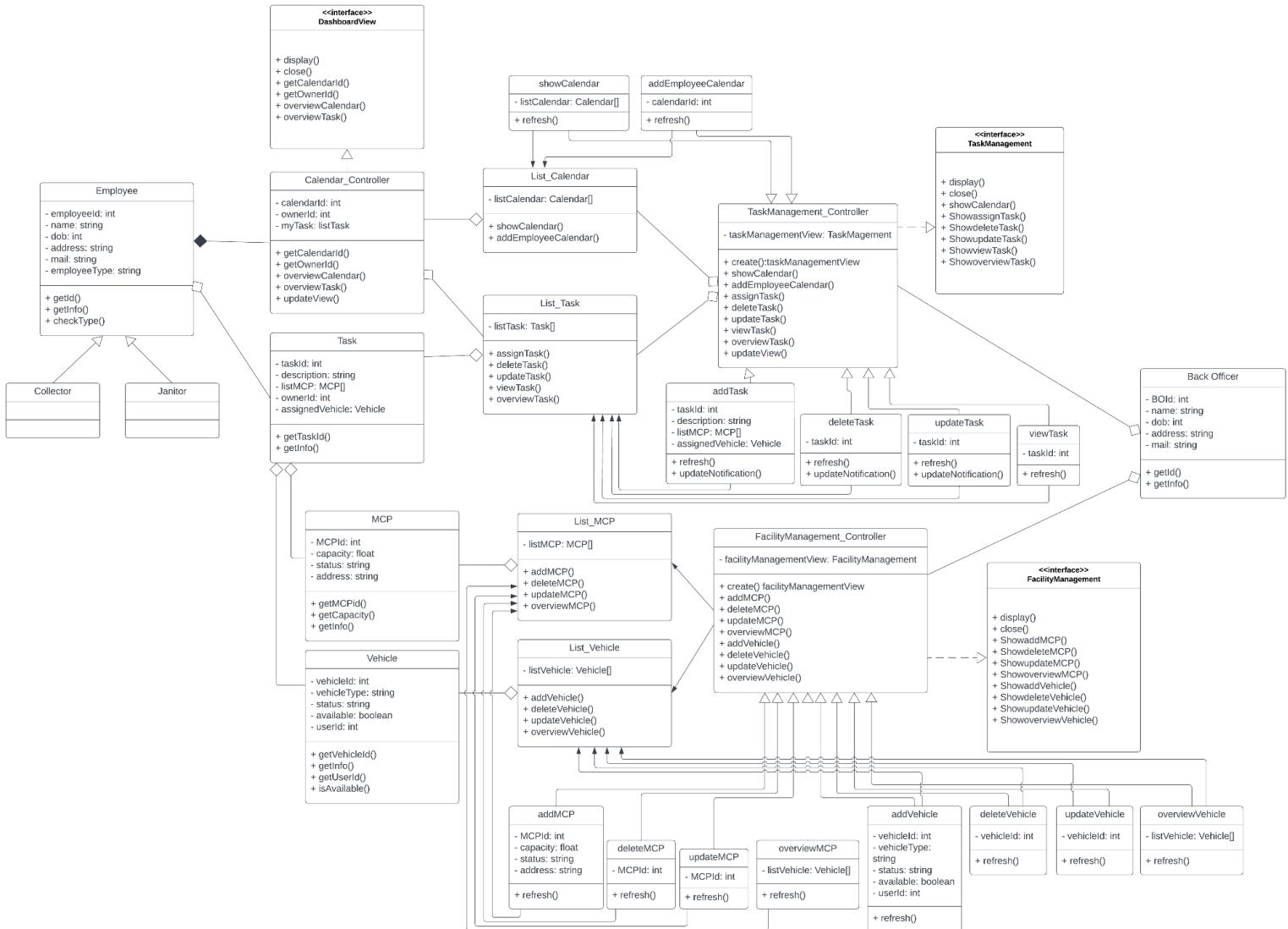
## 2.2 Sequence Diagram

*Proposal a conceptual solution for the route planning task. Draw a sequence diagram to illustrate it.*



## 2.3 Class Diagram

*Draw a class diagram of the Task Assignment module as comprehensive as possible.*

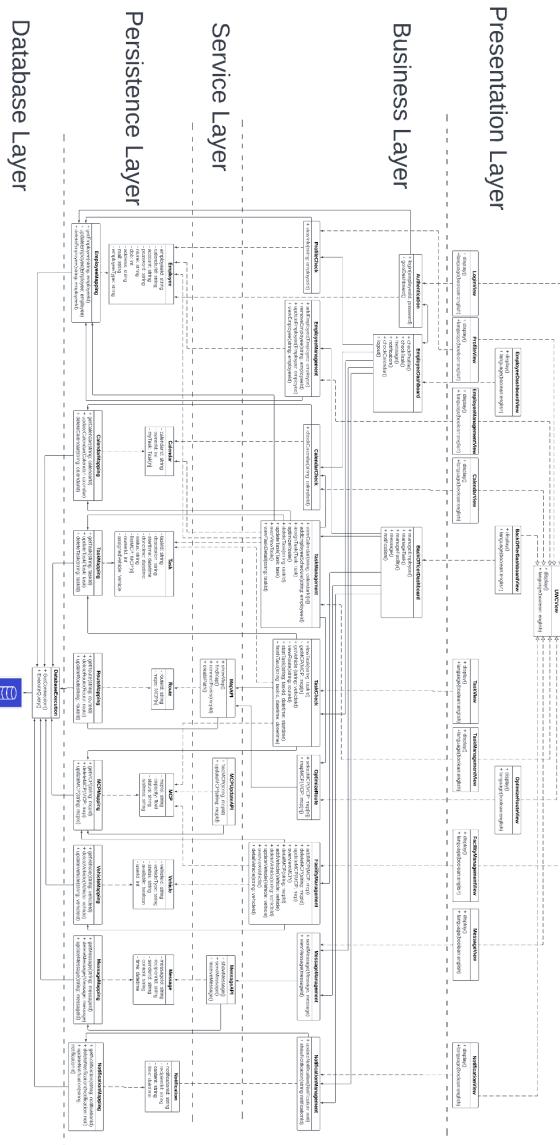


Note: For a better view of the diagrams above, you can use the PDF below:

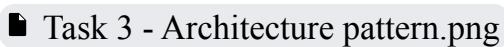
[https://drive.google.com/drive/folders/18D-DNscqYcIcofBanNmM9uvvyG3ezUsl?usp=share\\_link](https://drive.google.com/drive/folders/18D-DNscqYcIcofBanNmM9uvvyG3ezUsl?usp=share_link)

### **3 TASK 3: Architecture Design**

### **3.1 Describe the architectural approach**



*For a better view of the diagram, you can use the link below:*



### 3.1.1 Architecture Approach

After defining use-cases, activities, and sequences from the previous diagram, we decide to use *Layer Architecture*.

## Description



This layer organizes the system into layers, with related functions associated with each layer. A layer provides services to the layer above it, so the lowest level layers represent core services that are likely to be used throughout the system. In this system, we design 5 layers: Presentation, Business, Service, Persistence, and Database.

### Advantages

With this architecture, we can:

- Separate concerns among components. Components within a specific layer deal only with logic that pertains to that layer. For example, in the Business layer, components inside only have to deal with the logic in that layer.
- Moreover, each layer can be replaced by another better form of that layer without any change in other layers. This supports the opportunity to develop the system over time. For example, when we want another UI view, we just have to test, develop, govern or maintain only the Presentation layer.
- Easily apply APIs that support Message function, Notification function, and Map function, ... in the Service layer.

We also apply ORM (Object Relational Mapping) in the Persistence layer to easily gain data from the database. ORM is a programming technique that maps the recorded data into objects defined in the class. For example, in the Persistence layer, we create a class TaskMapping to implement a function that has SQL query to gain data from the database and transfer it to object Task. With this technique, we can:

- Gather all the SQL queries in the Persistence layer. Maintaining and re-using code is easier.
- Gain all benefits from OOP (Object Oriented Programming) without thinking too much.

Finally, we apply some of the Design Principles:

- SRP (Single Responsibility Principle): we separate ProfileCheck, CalendarCheck, TaskCheck, ... classes from EmployeeDashboard class to specify which class should be responsible for which function.



- OCP (Open-Closed Principle): we create many UI view classes inheriting from UWCView class, which help us to extend more UI when needed easily and closed for modification in UWCView class.
- ISP (Interface Segregation Principle): Employee and Back Officer use the data object created from Mapping class.

### 3.1.2 For the whole UWC 2.0 system, we plan to design 6 modules:

- Login-Logout:
  - + Input: Employee use accountId, and password to enter into UWC 2.0 system.
  - + Output: Log in successfully or unsuccessfully.
  - + Function: The system checks authentication between the input and data queried from the database to make the decision. If an employee wants to log out, the system will do that.
- Employee Management:
  - + Input: Employee information, Employee ID
  - + Output: Employee Information
  - + Function: The Back Officer can add, remove, update and check employee information.
- Task Management:
  - + Input: EmployeeID, TaskID, CalendarID, MCP[n], Vehicles
  - + Output: The assigned task for each employee, Task status, Optimized Route, and Calendar of each employee.
  - + Function: Back Officer can assign tasks for each employee, remove or update tasks' information. They can also create and modify optimized routes for collectors.
- Message:
  - + Input: Message, User ID.
  - + Output: Message.
  - + Function: Send a message and read the message.
- Notification:
  - + Input: User\_Id, Notification
  - + Output: Notification.

- + Function: Update notification.

## 3.2 Draw an implementation diagram for Task Assignment module

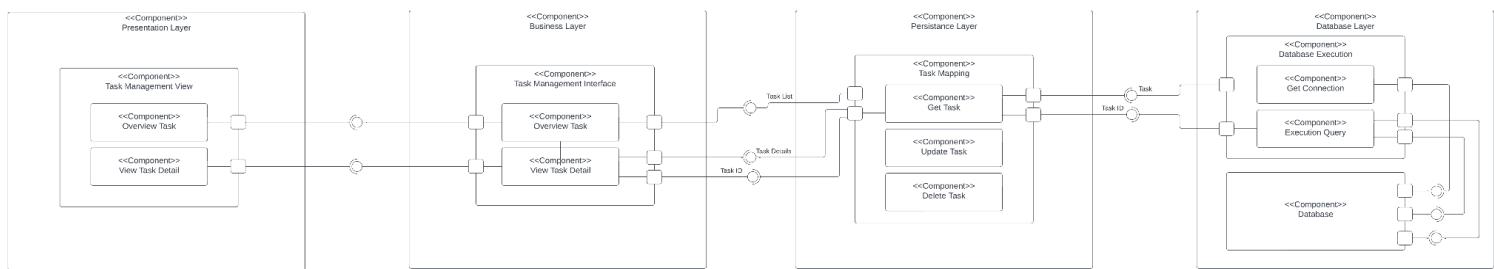
In each component, the UI is in the presentation layer. From the presentation layer, data is processed in the business layer and then sent back to the presentation layer.

In the business layer, data is sent by the persistent layer which contains a mapping object to the database layer.

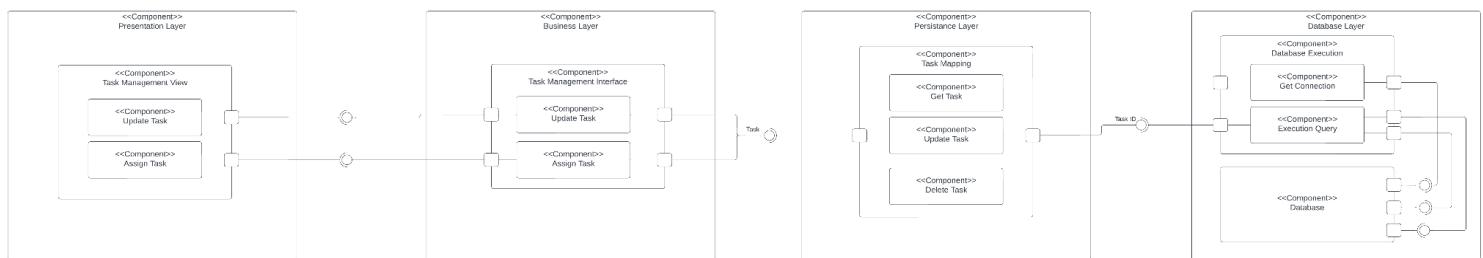
In the database layer, it contains a database and a database execution component which is responsible for processing queries and sending data to other layers.

There is also a service layer, which provides logic for the other layers to use.

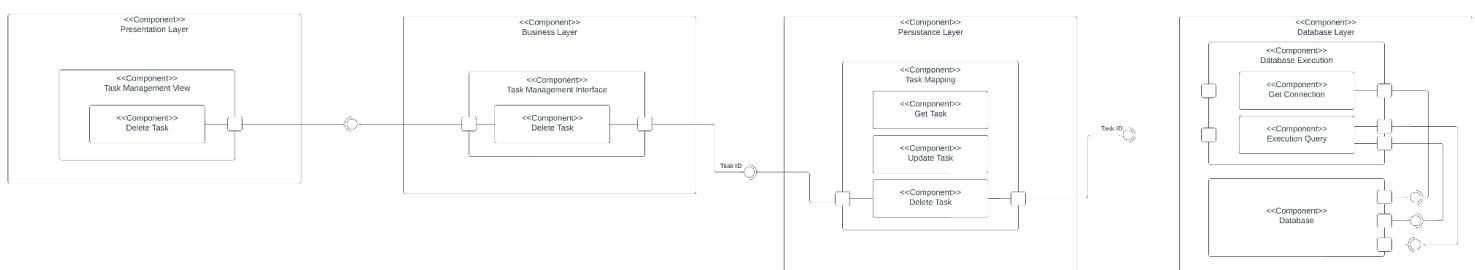
### Overview & View Task



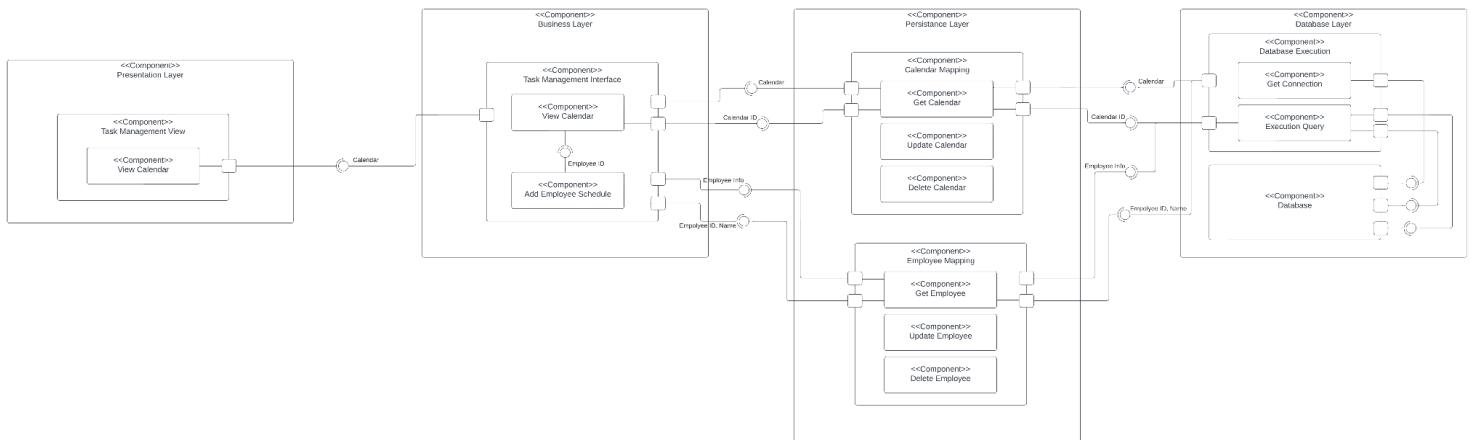
### Update & Assign Task



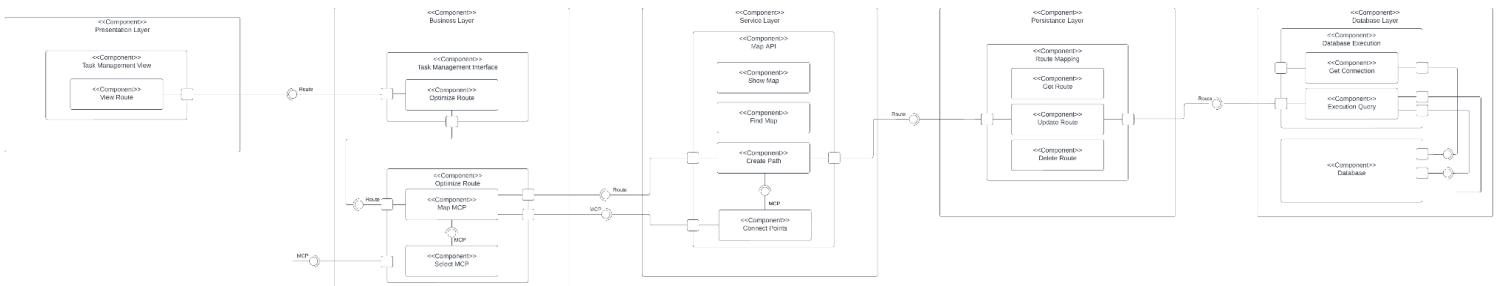
### Delete Task



[View Calendar](#)



**View Route**



For a better view of the diagram, you can use the link: [Task 3B](#)

## 4 TASK 4: Implementation - Sprint 1

### 4.1 Online repository (GitHub)

To be easier to manage and document, we decided to use GitHub as our online repository.

Link to GitHub:

<https://github.com/baonguyenduy2002/UWC2.0.git>

### 4.2 Adding documents, materials, and folders

You can access the GitHub link above to view the details of Requirements, System modeling, and Architectural design.

Below is a quick review of the documents, materials, and folders added to the online repository:

The screenshot shows the GitHub repository page for 'baonguyenduy2002/UWC2.0'. The repository is public and contains the following files:

- README.md
- Design
- Requirements
- README.md

The commit history shows:

- baonguyenduy2002 Add files via upload (7 hours ago)
- Design Add files via upload (7 hours ago)
- Requirements Create README.md (7 hours ago)
- README.md Update README.md (3 days ago)

The repository has 16 commits, 0 stars, 1 fork, and 1 watch.

**UWC2.0**

Project for Software Engineering course

I - Introduction

- Identify the context of this project
  - Along with economic development, urban waste management is one of the biggest challenges we face today, especially in big cities. The authorities need to come up with solutions to solve this problem in an efficient, cost-effective way to achieve the development goals set out in SDG 11 and SDG 6. This is the reason for the development of the UWC 1.0 information management system.
  - UWC 1.0 is a decision support system for waste collection. Thus after 5 years, UWC 2.0 is developed.
  - UWC 2.0 is a well-rounded developed version than UWC 1.0. With UWC 2.0, Back officers, Collectors, and Janitors can manage information and work schedules more easily and efficiently. At the same time, it also supports the management of vehicle information and waste collection points, thereby providing an efficient and cost-effective operation solution. UWC 2.0 also integrates a messaging function that allows contact information to be updated quickly and promptly to facilitate the resolution of arising problems.

2. Who are the relevant stakeholders?

- Back officers
- Janitors
- Collectors

3. What are their current needs?

- As a Back officer, I need to have an overview of Janitors' and Collectors' work calendars.
- As a Back officer, I need to have an overview of vehicles and their technical details (including weight, capacity, fuel consumption, ...)
- As a Back officer, I need to have an overview of all MCPs (major collecting points) and information about their capacity. MCPs should be updated every 15 minutes with the availability of at least 95% of their operating time.
- As a Back officer, I need to assign vehicles to Janitors and Collectors.
- As a Back officer, I need to be able to assign Janitors and Collectors to MCPs.
- As a Back officer, I need to create a route for each Collector. The route should be optimized in terms of fuel consumption and travel distance.
- As a Back officer, I need to be able to send messages to Collectors and Janitors.
- As a Collector/Janitor, I need to have an overview of my work calendar.
- As a Collector/Janitor, I need to have an overview of my tasks on a daily and weekly basis.
- As a Collector/Janitor, I need to be able to communicate with other Collectors, Janitors, and Back officers.
- As a Collector/Janitor, I need to be able to check in to check out every day.
- As a Collector/Janitor, I need to be notified about MCPs' capacity.

4. What could be their current problem? Employee

- Hard to manage employees (Collectors/Janitors):
  - Hard to check daily and weekly tasks
  - Hard to manage the human resources
  - Very difficult to generate working calendar for Collectors/Janitors, Back officer:
    - Can not effectively organize and distribute the tasks to staff
    - Collectors/Janitors' productivity may be decreased
    - Technical equipment
    - Not having a real-time means of communication
    - Can not handle spontaneous problems
    - Delay information and report about the progress from Collectors/Janitors to Back officer Working progress
    - Not having an optimized route or path for JC:
    - Wasting fuel (gasoline, time, ...)
    - During the MCP's route
    - Not having a real-time management system
    - Hard to track the working progress of JC
    - Hard to keep records of working progress

5. What benefits the UWC 2.0 will be for each stakeholder?

- Back officers can monitor and retrieve information from Collectors and Janitors rapidly.
- Back officers can follow the status of all vehicles and Major collecting points (MCPs), then can make an efficient operation and cost - reduction.
- Back officers, Collectors, and Janitors can observe their working calendar easily, thus avoiding time-consuming and improving efficiency.
- Having a high response means of communication will help to solve a sudden problem quickly and in a timely manner.

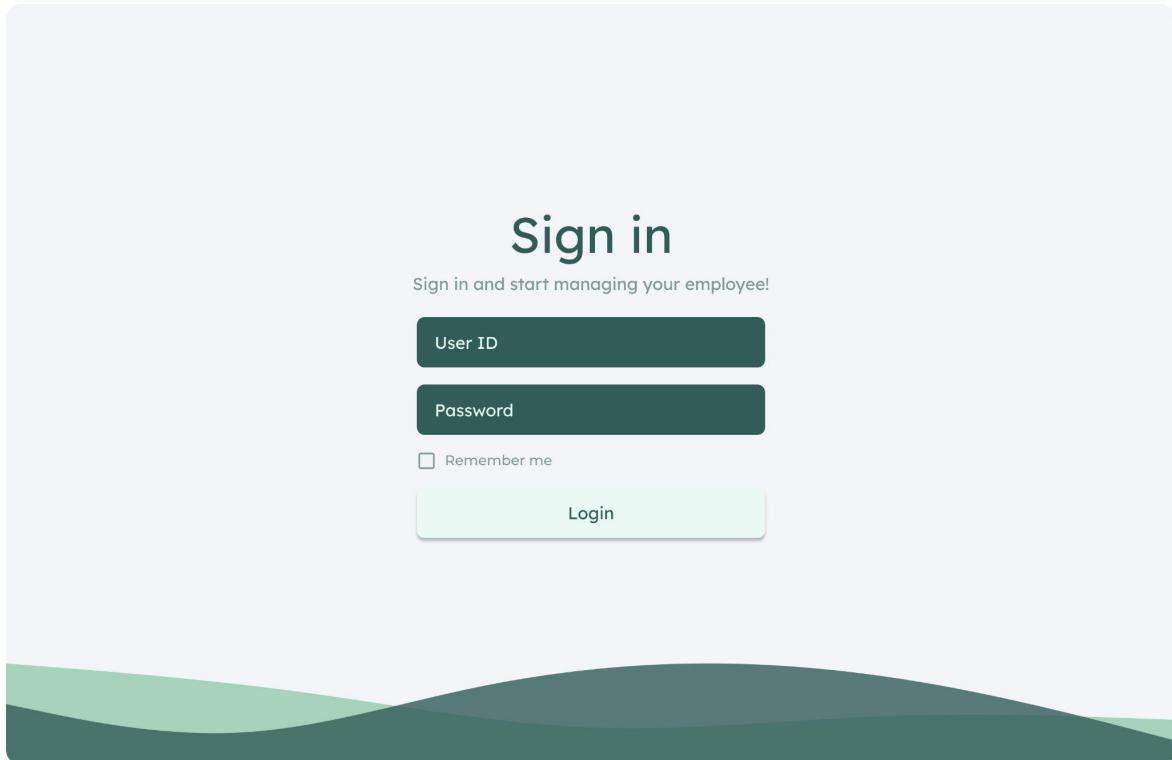


## 4.3 Implement MVP1 – design an interface

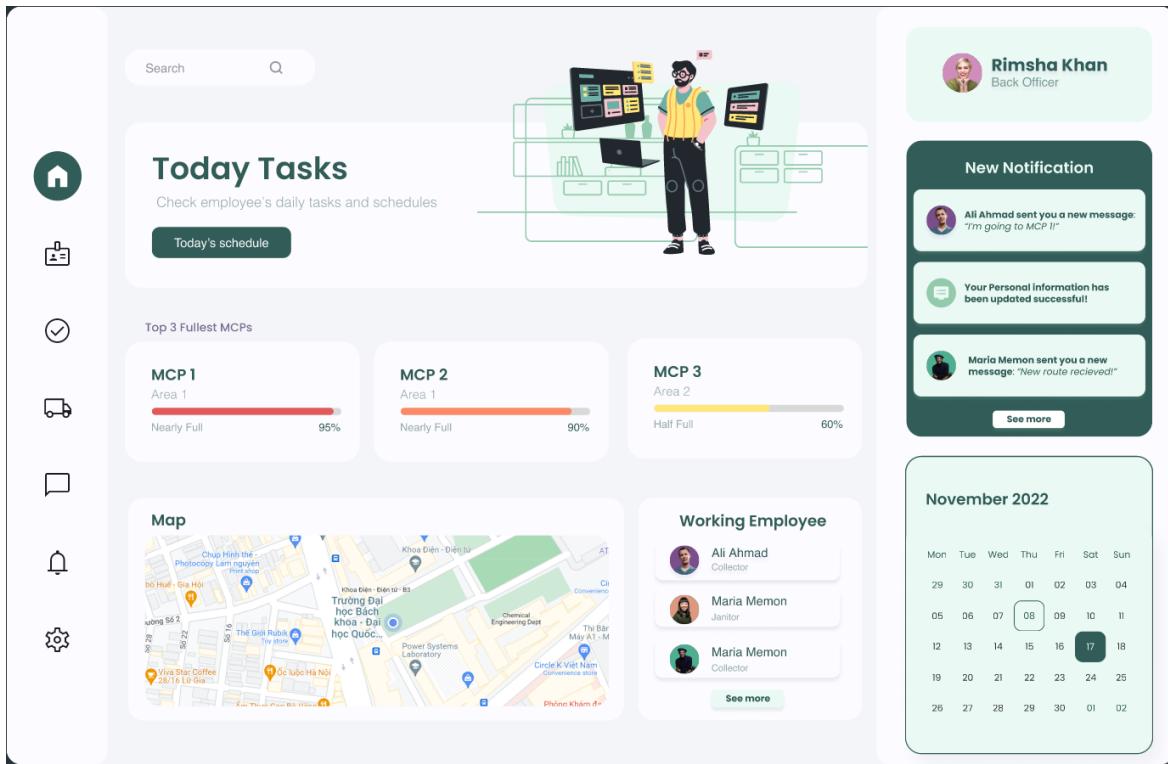
You can visit our Figma design:

<https://www.figma.com/proto/Z2wzvurY7D5TmXnIUApQBz/MVC-2.0-Task-management-dashboard?node-id=144%3A120&scaling=contain&page-id=0%3A1&starting-point-node-id=144%3A120>

### 4.3.1 Login Page

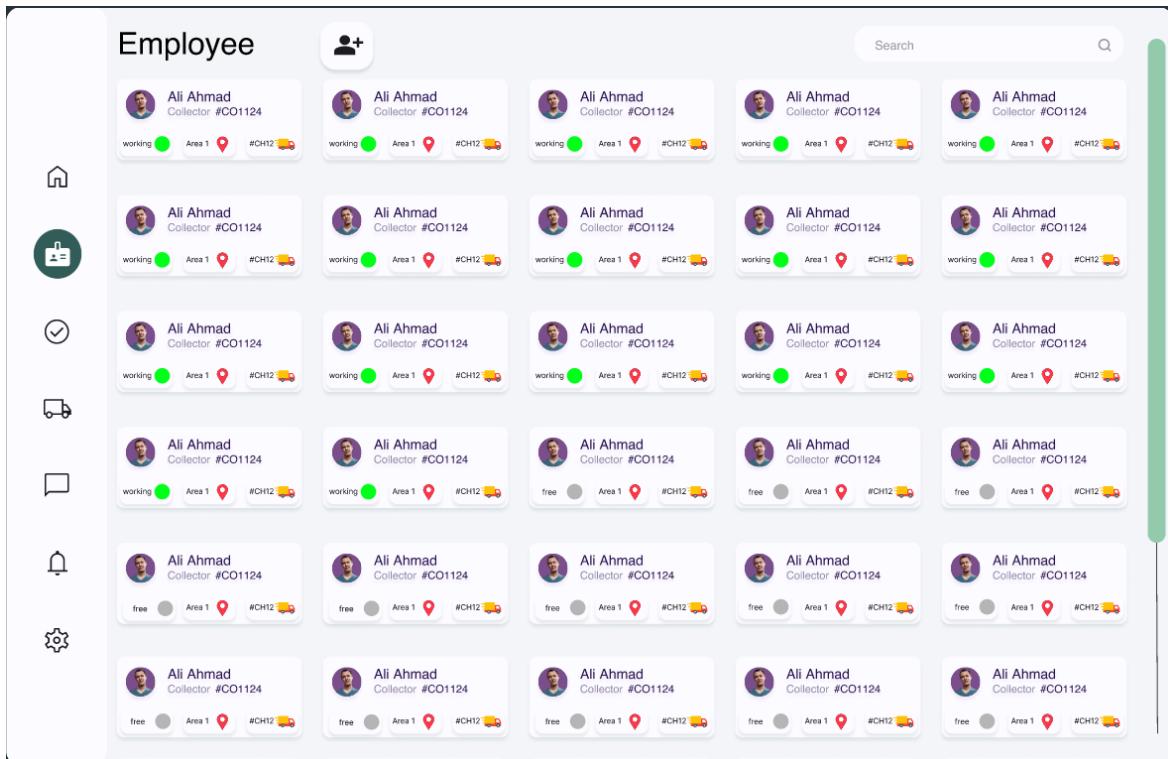


### 4.3.2 Task Management Dashboard

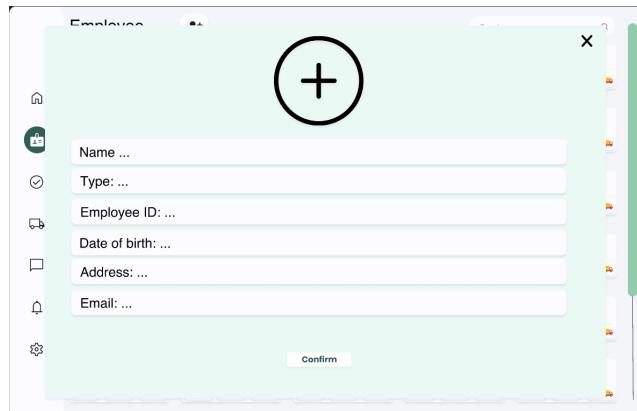


The dashboard features a sidebar with icons for Home, Schedule, Tasks, Notifications, Map, and Settings. The main area includes a search bar, a central illustration of a worker at a desk, and a "Today Tasks" section showing a schedule for the day. It also displays three MCPs (Multi-Point Collectors) with progress bars: MCP 1 (Area 1, Nearly Full, 95%), MCP 2 (Area 1, Nearly Full, 90%), and MCP 3 (Area 2, Half Full, 60%). A "Map" section shows the university campus layout with various buildings and landmarks. On the right, there's a "Working Employee" list (Ali Ahmad, Collector), a "New Notification" box (Ali Ahmad message, personal info update, Maria Memen message), and a "November 2022" calendar.

### 4.3.3 Manage Employee



The employee management interface shows a grid of employee cards for "Ali Ahmad, Collector #CO1124". Each card includes a profile picture, name, title, and status indicators (working or free). The sidebar on the left has icons for Home, Schedule, Tasks, Notifications, Map, and Settings. A detailed view of one employee card is shown at the bottom right, displaying personal information like date of birth (14/05/1996), address (142 To Hien Thanh, ward 14, district 10), and email (aliahmad@gmail.com).



#### 4.3.4 Manage Task

### Task Management

	Task List			
	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Gathering Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: None Route: None Status: Assigned	<b>Gathering Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: None Route: None Status: Checked in	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Checked in
	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned
	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned
	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned
	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned	<b>Collecting Garbage</b> Date: 16/11/2022 Time: 19h - 23h Collector: Ali Amad Working Area: Area 1 MCPs list: 1,2,6,8 Vehicle: #CH12 Route: Created Status: Assigned

### 4.3.5 Manage Facility

#### MCPs Management

	MCP 1	MCP 2	MCP 3	MCP 4
	MCP 1 Area 1 Nearly Full 95%	<b>MCP 2</b> Area: 1 #123456		
		Location: 268 Ly Thuong Kiet, 14 ward, 10 District, Ho Chi Minh City		
		Status: 90%		
		Last updated: 4 minutes ago		

**MCP 2**  
Area: 1  
#123456

Location: 268 Ly Thuong Kiet, 14 ward, 10 District, Ho Chi Minh City

Status: 90%

Last updated: 4 minutes ago

### 4.3.6 Message

The image shows a mobile application interface for a messaging service. On the left, there's a vertical navigation bar with icons for Home, Calendar, Checkmark, Share, Chat (highlighted in green), Bell, and Settings. The main area is titled "Chat". A search bar at the top says "Search for users". Below it is a list of messages from a user named "Rimsha Khan", each consisting of a profile picture and the text "Hello!!!". On the right, the conversation with Rimsha Khan is shown. Her messages are in light blue bubbles, and the responses are in grey bubbles. The messages are placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget ultricies diam pharetra vel. Vivamus lobortis eu mauris ac vehicula. Suspendisse eget bibendum mi. Orci varius natoque penatibus.", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget.", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget.", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget.", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget ultricies diam pharetra vel. Vivamus lobortis eu mauris ac vehicula. Suspendisse eget.", and "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris condimentum sodales nisl, eget ultricies diam pharetra vel. Vivamus lobortis eu mauris ac vehicula. Suspendisse eget.". At the bottom, there's a message input field with "Write a message..." and a send button icon.



The screenshot shows a mobile application interface. On the left, a vertical navigation bar contains icons for Home, Task List, Checkmark, Document, Chat (highlighted in green), Bell, and Settings. The main area has two tabs: 'Chat' (selected) and 'Profile'. The 'Chat' tab shows a list of messages from a user named 'Rimsha Khan', each containing the text 'Hello!!!'. The 'Profile' tab displays a circular profile picture of 'Rimsha Khan', followed by her details: Name (Rimsha Khan), Role (Back officer), Email (lorem@gmail.com), Date of birth (99/99/9999), and a timestamp (14:51). A message input field with a placeholder 'Write a message...' and a send button is at the bottom.

#### 4.3.7 Notification

The screenshot shows a 'Notifications' screen with a vertical navigation bar on the left containing icons for Home, Task List, Checkmark, Document, Chat, Bell (highlighted in green), and Settings. The main area lists seven notifications:

- Ali Ahmad sent you a new message: "I'm going to MCP 1!" (20 mintues ago)
- Your Personal information has been updated successful! (31 mintues ago)
- Maria Memon sent you a new message: "New route received!" (40 mintues ago)
- Your Personal information need to be updated! (45 mintues ago)
- Ali Ahmad sent you a new message: "I finished collecting at MCP 3!" (47 mintues ago)
- Maria Memon sent you a new message: "I'm going to MCP 2, I need a ro. . ." (50 mintues ago)
- Ali Ahmad sent you a new message: "I'm going to MCP 3!" (1 hour 10 mintues ago)



#### 4.3.8 Setting

**General**

Language English ▾

**Log out** Log out ➔

---

**Account**



Last Name Maria edit

Last Name Memon edit

Username maria.memon edit

Password \*\*\*\*\* edit

Date of Birth 14/09/1989 edit

Email maria.memon@gmail.com edit

Address 286 Ly Thuong Kiet, Ward 14, District 10, Ho Chi Minh City edit

---

## 5 TASK 5: Implement - Sprint 2

This is our prototype:

<https://github.com/baonguyenduy2002/UWC2.0>

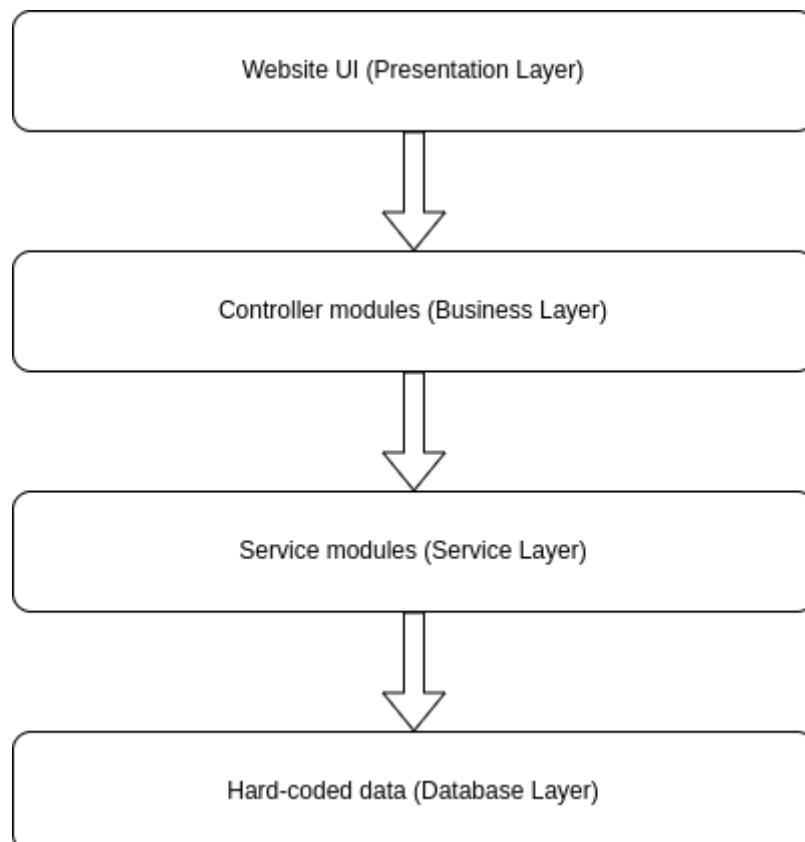
You can run this prototype by these steps:

1. Install Nodejs: <https://nodejs.org/en/>
2. Install the required package by run this command: npm install
3. Run the app in your localhost in port 3000: npm start

### 5.1 Programming environment

To implement MVC2, we use NodeJS and ReactJS as our environment.

For the data, we hard-code the data and implement a fake API (using Asynchronous Programming) on the front-end in order to simulate the Layer Architecture:





## 5.2 Git commit log

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

baonguyenduy2002 / UWC2.0 Private

Code Issues Pull requests Actions Projects Security Insights

1 develop

Commits on Dec 3, 2022

- Bao fixed the TaskSidebar  
mynameisba11 committed 4 days ago
- feat: config logout and remember me  
TanLoc-CS committed 4 days ago
- fix: fix task sidebar button size  
TanLoc-CS committed 4 days ago
- fix: fix sidebar button animation  
TanLoc-CS committed 4 days ago
- Merge pull request #1 from baonguyenduy2002/feature/tasksidebar  
TanLoc-CS committed 5 days ago
- feat: add icons and fix sidebar button  
TanLoc-CS committed 5 days ago

Commits on Dec 2, 2022

- Sidebar in ManageTask and need more code in Logout in page Setting  
mynameisba11 committed 5 days ago
- fix: fix loginpage.css  
TanLoc-CS committed 6 days ago

Commits on Dec 1, 2022

- feat: handle login logic and add mock data  
TanLoc-CS committed 6 days ago
- feat: finish login ui  
TanLoc-CS committed 6 days ago
- Update Login  
mynameisba11 committed 6 days ago

Commits on Nov 30, 2022

- feat: fix folder structure and add layout  
TanLoc-CS committed 7 days ago
- First implementation  
mynameisba11 committed 8 days ago

Commits on Nov 24, 2022

- Create README.md  
baonguyenduy2002 committed 14 days ago
- Delete MVC directory  
baonguyenduy2002 committed 14 days ago
- Create README.md  
baonguyenduy2002 committed 14 days ago

Commits on Nov 16, 2022

- Add files via upload  
baonguyenduy2002 committed 21 days ago
- Create README.md  
baonguyenduy2002 committed 21 days ago
- Add files via upload  
baonguyenduy2002 committed 21 days ago
- Delete 3.1 Class diagram with Architecture approach.png  
baonguyenduy2002 committed 21 days ago
- Add file via upload  
baonguyenduy2002 committed 21 days ago
- Add files via upload  
baonguyenduy2002 committed 21 days ago
- Add files via upload  
baonguyenduy2002 committed 21 days ago
- Create README.md  
baonguyenduy2002 committed 21 days ago

Commits on Nov 14, 2022

- Update README.md  
baonguyenduy2002 committed 23 days ago

Commits on Nov 7, 2022

- Initial commit  
baonguyenduy2002 committed on Nov 7

Newer Older



The screenshot shows the GitHub repository page for `baonguyenduy2002/UWC2.0`. The main navigation bar includes links for Search or jump to..., Pull requests, Issues, Codespaces, Marketplace, and Explore. The repository is private. The top right corner shows Watch (1), Fork (0), Star (0), and a gear icon.

The repository has 1 commit, 1 pull request, and 0 issues. The pull request is titled "Merge pull request #2 from baonguyenduy2002/feature/manageTaskfull" and was merged by `baonguyenduy2002` on Dec 6, 2022. It contains 13 commits from various contributors.

The commit history is organized into three sections based on commit date:

- Commits on Dec 6, 2022:**
  - Merge branch 'facility' of <https://github.com/baonguyenduy2002/UWC2.0> ... by `HuyHoang0712` committed yesterday
  - fix ui facility by `HuyHoang0712` committed yesterday
  - Half Finished by `HuyHoang0712` committed yesterday
  - feat: add message api by `TanLoc-CS` committed yesterday
  - Resolve conflict by `mynamesba11` committed 2 days ago
  - finish UI of Facility by `mynamesba11` committed 2 days ago
  - fix: fix syntax in employee service by `TanLoc-CS` committed 2 days ago
  - feat: add mcp api by `TanLoc-CS` committed 2 days ago
  - Merge pull request #2 from baonguyenduy2002/feature/manageTaskfull ... by `baonguyenduy2002` committed 2 days ago
- Commits on Dec 5, 2022:**
  - getTasks and removeTasks by `mynamesba11` committed 2 days ago
  - Employee Management Windows by `HuyHoang0712` committed 2 days ago
  - Chat by `Danh Nguyen` committed 2 days ago
  - bug 1 done :)
  - backup by `honitr0122` committed 2 days ago
  - feat: add new profile photo for each createEmployee by `TanLoc-CS` committed 2 days ago
  - fix: fix employee data by `TanLoc-CS` committed 2 days ago
  - feat: add data and handle employee api by `TanLoc-CS` committed 2 days ago
  - Merge branch 'Feature/backend' into feature/manageTaskfull by `mynamesba11` committed 2 days ago
  - new employee data by `mynamesba11` committed 2 days ago
  - fix bug dashboard by `honitr0122` committed 2 days ago
  - Fix Vehicle Tag by `HuyHoang0712` committed 2 days ago
  - Finish Setting by `Danh Nguyen` committed 2 days ago
  - add lib popup dialog by `honitr0122` committed 2 days ago
  - Dashboard and Task first design by `honitr0122` committed 2 days ago
  - Bao has done all Task Management function - Back End + folder structu... by `mynamesba11` committed 2 days ago
  - Update index.js by `HuyHoang0712` committed 2 days ago
  - Finished MCPs Management by `HuyHoang0712` committed 2 days ago
- Commits on Dec 4, 2022:**
  - Fix DropDownList by `HuyHoang0712` committed 3 days ago
  - Noti + Setting by `Danh Nguyen` committed 3 days ago
  - Basic facility window by `HuyHoang0712` committed 3 days ago
  - Setting 80% by `Danh Nguyen` committed 3 days ago
  - Back-End: Refactor folders and first step in back-end. 2g00 4/12/2022 by `mynamesba11` committed 4 days ago
  - remove component DropDownList by `HuyHoang0712` committed 4 days ago
  - Facility test by `HuyHoang0712` committed 4 days ago
- Commits on Dec 3, 2022:**
  - fix: fix RequireAuth logic by `TanLoc-CS` committed 4 days ago

At the bottom, there are "Newer" and "Older" buttons. The footer includes links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.



The screenshot shows the GitHub interface for a repository named 'baonguyenduy2002/UWC2.0'. The 'Code' tab is selected, displaying the commit history for the 'develop' branch. The commits are organized into two main sections: 'Commits on Dec 7, 2022' and 'Commits on Dec 6, 2022'. Each commit includes the author's name, date, message, and a copy icon.

**Commits on Dec 7, 2022:**

- Resolve merge conflict in develop (TarLoc-CS committed 1 hour ago)
- Merge: Manage Task (TarLoc-CS committed 1 hour ago)
- Finished Dashboard! (HuyHoang0712 committed 1 hour ago)
- Final draft (mynamesbao11 committed 2 hours ago)
- Merge branch 'feature/ManageTaskbackend' into develop (mynamesbao11 committed 2 hours ago)
- Added MCPsTag in Dashboard (HuyHoang0712 committed 2 hours ago)
- Resolve conflict in feature/employee (TarLoc-CS committed 2 hours ago)
- pending: employee ui (TarLoc-CS committed 3 hours ago)
- Done ManageTask (mynamesbao11 committed 3 hours ago)
- Add Facility Component (HuyHoang0712 committed 4 hours ago)
- Fix popup in manage employee mainscreen (HuyHoang0712 committed 4 hours ago)
- Merge: merge feature/backend > feature/backendfacility > Setting to d... (TarLoc-CS committed 10 hours ago)
- feat: connect api to facility (TarLoc-CS committed 10 hours ago)
- fix: change vehicle data format (TarLoc-CS committed 10 hours ago)
- fix bug (HuyHoang0712 committed 11 hours ago)
- a (HuyHoang0712 committed 12 hours ago)
- save work (mynamesbao11 committed 13 hours ago)
- fix: render DetailFacilityMCP (TarLoc-CS committed 14 hours ago)

**Commits on Dec 6, 2022:**

- save work (mynamesbao11 committed 17 hours ago)
- save work (mynamesbao11 committed 18 hours ago)
- save work (mynamesbao11 committed 20 hours ago)
- Merge branch 'feature/backend' into feature/ManageTaskbackend (mynamesbao11 committed 21 hours ago)
- On process, still bug in getMCP (mynamesbao11 committed 21 hours ago)
- add mui-material (HuyHoang0712 committed 22 hours ago)
- Update index.jsx (HuyHoang0712 committed 22 hours ago)
- Merge branch 'Facility' into feature/backendfacility (mynamesbao11 committed 23 hours ago)
- Test Personallinfo page (HuyHoang0712 committed yesterday)
- Facility UI done! (HuyHoang0712 committed yesterday)
- remove unnecessary import lines (hontri9122 committed yesterday)
- add add new task form to calendar, test mapbox api (hontri9122 committed yesterday)
- Merge branch 'Facility' of https://github.com/baonguyenduy2002/UWC2.0 ... (HuyHoang0712 committed yesterday)
- Fix Facility (HuyHoang0712 committed yesterday)
- nothing new (mynamesbao11 committed yesterday)
- Merge branch 'Facility' of https://github.com/baonguyenduy2002/UWC2.0 ... (mynamesbao11 committed yesterday)
- update UI Facility (mynamesbao11 committed yesterday)

Newer Older