

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



SOFTWARE ENGINEERING (CO3001)

Assignment

UWC 2.0

Advisor: Mr. Tho Quan Thanh ,and Mr. Anh Nguyen Duc
Students: Cao Chánh Trí - 2153917
Lê Trọng Đức - 2152523
Trần Hoàng Khôi Tuấn - 2012359
Mai Tôn Đăng Khánh - 2152122
Đinh Lê Dũng - 2152483

HO CHI MINH CITY, FEBRUARY 2023



Contents

1 Business Context	3
1.1 Introduction	3
1.2 Relevant Stakeholders: What problems and needs do they currently face?	3
1.3 UWC 2.0 benefits	5
1.4 Requirements	6
1.4.1 Functional Requirements	6
1.4.2 Non-functional Requirements	6
1.4.3 Use-case Diagram for the whole system	7
1.5 Task management module use case diagram	9
1.5.1 Use case diagram	9
1.5.2 Table description	9
2 System modelling	13
2.1 Activity Diagram	13
2.1.1 Task Management	13
2.1.2 Check Calendar	14
2.1.3 View Task	15
2.2 Sequence Diagram	16
2.3 Class diagram	17
2.4 User interface (Desktop-view for back-officers)	18
3 Architecture design	23
3.1 Task 3.1	23
3.1.1 Layered Architecture	23
3.1.2 Presentation strategy	23
3.1.3 Data storage approach	24
3.1.4 API management	24
3.2 Component diagram	26
3.2.1 Component diagram	26
3.2.2 Description	26



Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Cao Chánh Trí	2153917	- Task 1.1 - Task 2.1 - Task 3.2 - ... - ...	20%
2	Lê Trọng Đức	2152523	- Task 1.3 - Task 2.2 - Task 3.1 - ... - ...	20%
3	Trần Hoàng Khôi Tuấn	2012359	- Task 1.2 - Task 2.3 - Task 3.1 - ... - ...	20%
4	Mai Tôn Đăng Khánh	2152122	- Task 1.3 - Task 2.3 - Task 3.2 - ... - ...	20%
5	Dinh Lê Dũng	2152483	- Task 1.2 - Task 2.4 - Task 3.1 - ... - ...	20%



1 Business Context

1.1 Introduction

Sustainable Development Goal (SDG) 11: Sustainable Cities and Communities and SDG 6: Clean Water and Sanitation both list urban waste management as one of the major issues that many nations around the world confront and are thus working to address. Developing countries that continue to place a high priority on development and economic expansion are accorded special consideration. Solid waste management is expensive and inefficient in urban settings. Governments and organizations place a strong emphasis on improving garbage collection and management due to the positive effects it has on cities, societies, and the environment.

Although UWC 1.0 is a respectable system, there are still several flaws, leading to the development of UWC 2.0. Compared to UWC 1.0, UWC 2.0 is a more fully developed version.

Back officers, Collectors, and Janitors can all handle information and work schedules more effectively and conveniently with UWC 2.0. It also allows the administration of vehicle information and trash collecting stations, making it a practical and affordable operation solution. Additionally, UWC 2.0 incorporates a messaging feature that enables contact information to be updated quickly and effectively to aid in the problem-solving process.

1.2 Relevant Stakeholders: What problems and needs do they currently face?

Waste collection is frequently delegated to a professional waste management organization. In a typical waste collection process, there are three major parties involved:

- **Back officers** who operate a central system to create a calendar and coordinate front collectors and janitors
- **Collectors** who operate various types of vehicles
- **Janitors** who manually collect trash from Major Collecting Points (MCPs).



Back Officers

Needs	Problems
<ul style="list-style-type: none">- Real-time messages for tasks assignments- Have an overview of Janitors' and Collectors' work calendars- Have an overview of vehicles and their technical details- Have an overview of all MCPs and information about their capacity	<ul style="list-style-type: none">- Significant communication delay- Not having an accessible working calendar of Collectors/Janitors,- Management module, route arrangement are labor intensive- Inefficient task distribution- Insufficient amount of features and statistics- Technical equipment and MCPs status is unavailable

Collectors and Janitors

Needs	Problems
<ul style="list-style-type: none">- Have an overview of their work calendar- Have a detail view of their task on a daily and weekly basic- Be able to communicate with collectors, other janitors and back officers- Check in / check out task every day- Be notified about the MCPs if they are fully loaded- Feedback about their task to balance the work arrangement- Be informed of the state of their equipment/vehicle if there are any problems- Optimized predetermined route	<ul style="list-style-type: none">- Not having an accessible working calendar nor an friendly calendar display- Assignment schedules between janitors/collectors may overlap- Slow/suboptimal task distribution- Difficulties in checking work shifts- Unstable notification and communication system- Vehicles/trollers and MCPs status is unavailable- Wasting the resources (gasoline, time, ...)- Duplicate the MCPs or collect path



1.3 UWC 2.0 benefits

UWC 2.0 is anticipated to address the aforementioned issues of relevant stakeholders, namely back officers, collectors and janitors, with:

- High response means of communication will help to solve a sudden problem quickly and in a timely manner.
- Make an efficient operation and cost reduction
- Friendly working calendar display
- Automatic and adaptable task distribution
- Speed up employee information retrievals
- Deliver insightful statistics for working progress
- Frequently update MCPs/vehicles status
- Can check and confirm their next tasks
- Check in/out daily tasks with ease
- Work route will be completely optimized avoiding traffic congestion



1.4 Requirements

1.4.1 Functional Requirements

Back officers	<ul style="list-style-type: none">- Be able to access the software using an admin account.- Can manage and retrieve employee information and work schedule (resume, days off,etc).- Be able to manage all vehicle details (weight, capacity, fuel consumption, etc).- Have information of all MCPs capacity (should be updated every 15 minutes).- Be able to assign vehicles to collectors and janitors.- Be able to assign janitors and collectors to MCPs.- Can assign routes which are optimized in terms of fuel consumption and distance for each collector.- A communication channel with collectors and janitors.
Collectors and Janitors	<ul style="list-style-type: none">- Keeping track of the work schedule, daily tasks, and weekly tasks all on one page without having to scroll down.- Have access to and control over personal information stored on the system.- A communication channel.- A system for task reports and daily attendance checks.- Have the ability to send feedback about the status of MCPs.

1.4.2 Non-functional Requirements

Performance	<ul style="list-style-type: none">- Janitors and collectors can receive messages from back officers in real time with delay less than 1 second.- Having the capacity to manage 1000 users' feedback at once.- System's response time must be less than 1 second.- The time it takes to fully load all images, text, and other content, should be as short as possible, ideally within a few seconds.- Data latency less than 100 milliseconds (Message system).
Reliabilities	<ul style="list-style-type: none">- The failure rate of real-time access is 0.003 (3 fail access out of 1000 access).- The system must work at all time.- The inaccessible time must less than 15 minutes in operating time (from 7 a.m to 5 p.m), and must be less than 30 minutes outside working time.
Security	<ul style="list-style-type: none">- Alert for server infiltration.- The integrity of data should be ensured, such as user information and other critical data, and minimize the risk of data loss or corruption.- Grant access and privileges depends on accounts' type (Back officer and Employee).- Prevent XSS (Cross-site scripting) attacks.
Ease of use	<ul style="list-style-type: none">- Users' guide is available within the application.- Be able to use all features of the program with proficiency after 30 minutes of training.- User-friendly interface, with an intuitive layout and easily understandable language.- Be able to switch languages between Vietnamese and English.

1.4.3 Use-case Diagram for the whole system

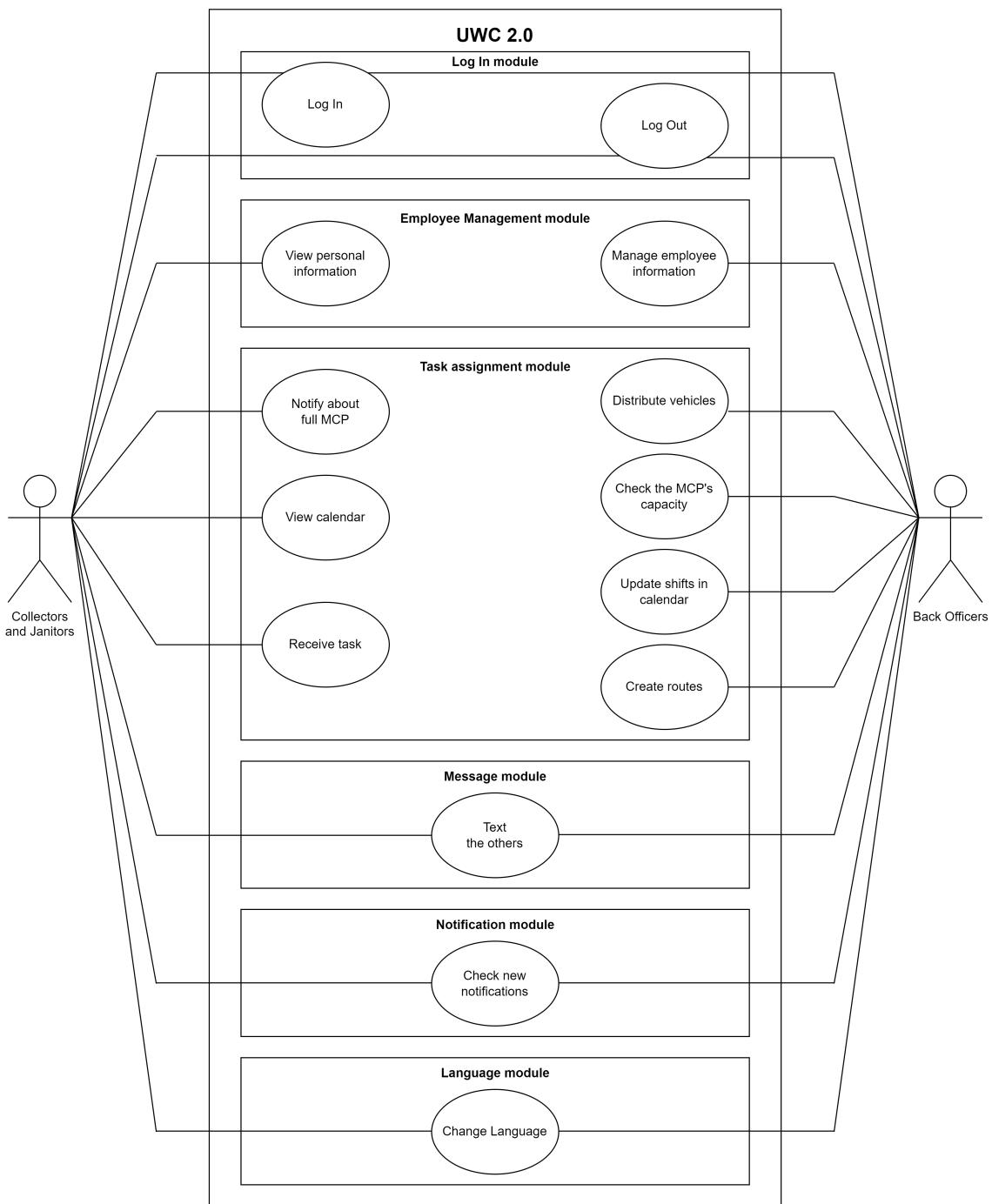


Figure 1. Use-case Diagram for the whole system



Table of actors

No.	Actor
1	Back officers
2	Collectors and Janitors

Table of Use-Cases

Use-case ID	Use-case name	Description
1	Log in	Collector, Janitor and Back Officer use to log into UWC 2.0.
2	Log out	Collector, Janitor and Back Officer use to log out UWC 2.0.
3	View personal information	Collector and Janitor can have a summary of their personal information.
4	Manage employee information	Back Officer can add or eliminate a staff, as well as manage the working schedule and personal information.
5	Notify about full MCP	Collector and Janitor are notified to get to the assigned location where MCPs are fully loaded to collect the garbage.
6	View calendar	Collector and Janitor can turn on the calendar to view their working day.
7	Receive task	Collector and Janitor receive list of tasks to do in the day.
8	Distribute vehicles	Back Officer distributes vehicles to the fully loaded MCPs' location.
9	Check the MCP's capacity	Back Officer manages the status of MCPs to see whether it is fully loaded.
10	Update shifts in calendar	Back Officer assigns tasks for the staff in their calendar.
11	Create routes	Back Officer navigates the routes for the vehicles.
12	Text the others	Collector, Janitor and Back Officer can communicate with the others via messages, including the following operations: Send, receive and view messages.
13	Check new notifications	List of notifications is visible to Collector, Janitor and Back Officer from "Update notification" use-case.
14	Change language	Collector, Janitor and Back Officer can alter between English and Vietnamese.

1.5 Task management module use case diagram

1.5.1 Use case diagram

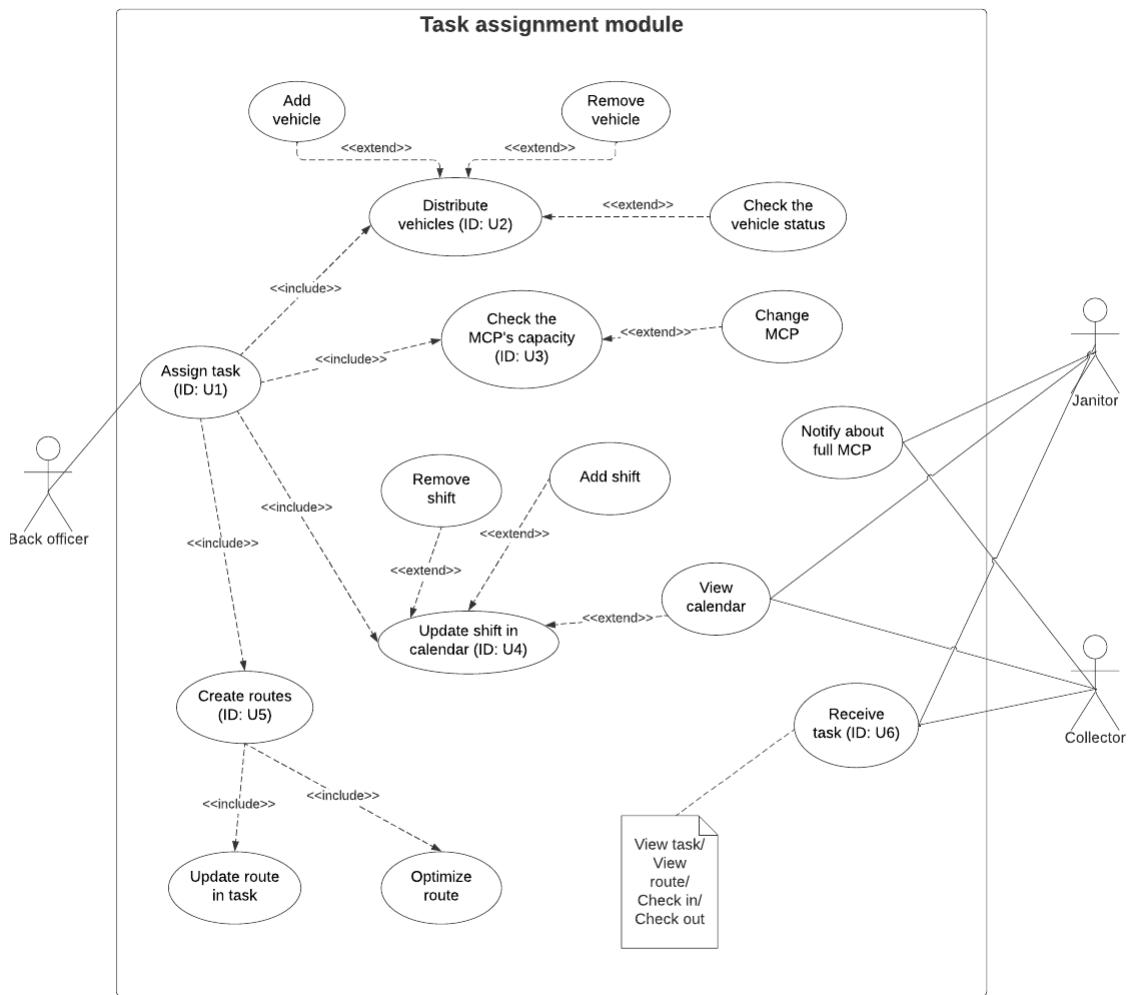


Figure 2. Use-case diagram

1.5.2 Table description



Use-case ID	U1
Use-case Name	Assign task
Use-case overview	Back officer check some information related to Task and distribute for janitors and collectors
Actor	Back officer
Precondition	- System is running - Vehicle's status and MCPs' capacity have already been updated - Internet Connection is stable
Trigger	Back officer click on "Arrange Task" button
Steps	1. Check vehicles' status and MCPs' capacity 2. Arrange tasks for employers 3. Update information about tasks on the users' calendars
Post conditions	Required information about tasks is updated adequately on employers' screens
Exception flow	None

Use-case ID	U2
Use-case Name	Distribute vehicles
Use-case overview	Back officer check all of vehicles and distribute for collectors
Actor	Back officer
Precondition	- System is running - Vehicle's status have already been updated - Internet Connection is stable
Trigger	Manage Back officer click on "Manage vehicles" button
Steps	1. Have information about vehicles(well-worked or need to fix) 2. Select vehicle by ID and put to the calendar 3. Display vehicle's ID on users' devices
Post condition	Required information about vehicles' IDs are displayed on collectors' calendars
Exception flow	- Several vehicles are just bought, so they don't have their IDs - Back officer enters new vehicles on the management list and create new IDs - Distribute new and old vehicles for staff



Use-case ID	U3
Use-case Name	Check the MCP's capacity
Use-case overview	Back officer check MCPs' capacity in order to notify to collectors or to change the MCP
Actor	Back officer
Precondition	- System is running - MCPs' capacity is updated every 15 minutes - Internet Connection is stable
Trigger	Manage Back officer click on "Manage MCP" button
Steps	1. Display MCPs' capacity on back officer's screen 2. If it is full, notify to the collectors and rearrange the location (if necessary) 3. If it is not full, update to the map to let janitors know
Post condition	Required information about MCP is accurate and real-time updated, notified continuously on employers' screens
Exception flow	None

Use-case ID	U4
Use-case Name	Update shift in calendar
Use-case overview	The back officer arrange shifts of the staff and will be updated to each calendar
Actor	Back officer
Precondition	- System is running - Already discuss with the staff about their shift - Internet Connection is stable
Trigger	Manage Back officer click on "Update Shift" button
Steps	1. Select staff by ID of the staff 2. Choose the shift that they have to work for each day 3. Save and update 4. Return to home screen
Post condition	The calendar must be able to check for overlapping events, handle important notes and real-time sync with popular calendars like Google, Apple and Outlook
Exception flow	None



Use-case ID	U5
Use-case Name	Create routes
Use-case overview	Back officer create route and send that position on map to janitors and collectors
Actor	Back officer
Precondition	- System is running - All routes' status is updated - Internet Connection is stable
Trigger	Back officer click on "Create route" button
Steps	1. Choose the ID of the vehicle 2. Select the destination 3. The algorithms will autonomously calculate the shortest path 4. After 1s, the path will be displayed
Post condition	Required well sync with Google map, the position of the route is displayed on employers' screens
Exception flow	None

Use-case ID	U6
Use-case Name	Receive task
Use-case overview	Collectors and janitors will receive the tasks and view the details of the tasks
Actor	Collectors and Janitors
Precondition	- System is running - The back officer assign some tasks - Internet Connection is stable
Trigger	After the back officer assigns the task
Steps	1. The staff choose the icon "My tasks" 2. The system will display all the tasks of the staff 3. There are some options beside the task such as : a tick box to check out, a hamburger icon to view detail, a location icon to view a map 4. After viewing the map, the staff chooses the "X" button to end viewing, the system will return to the home screen
Post condition	All the information related to the task must be updated frequently
Exception flow	None

This marked the end of task 1.

2 System modelling

2.1 Activity Diagram

2.1.1 Task Management

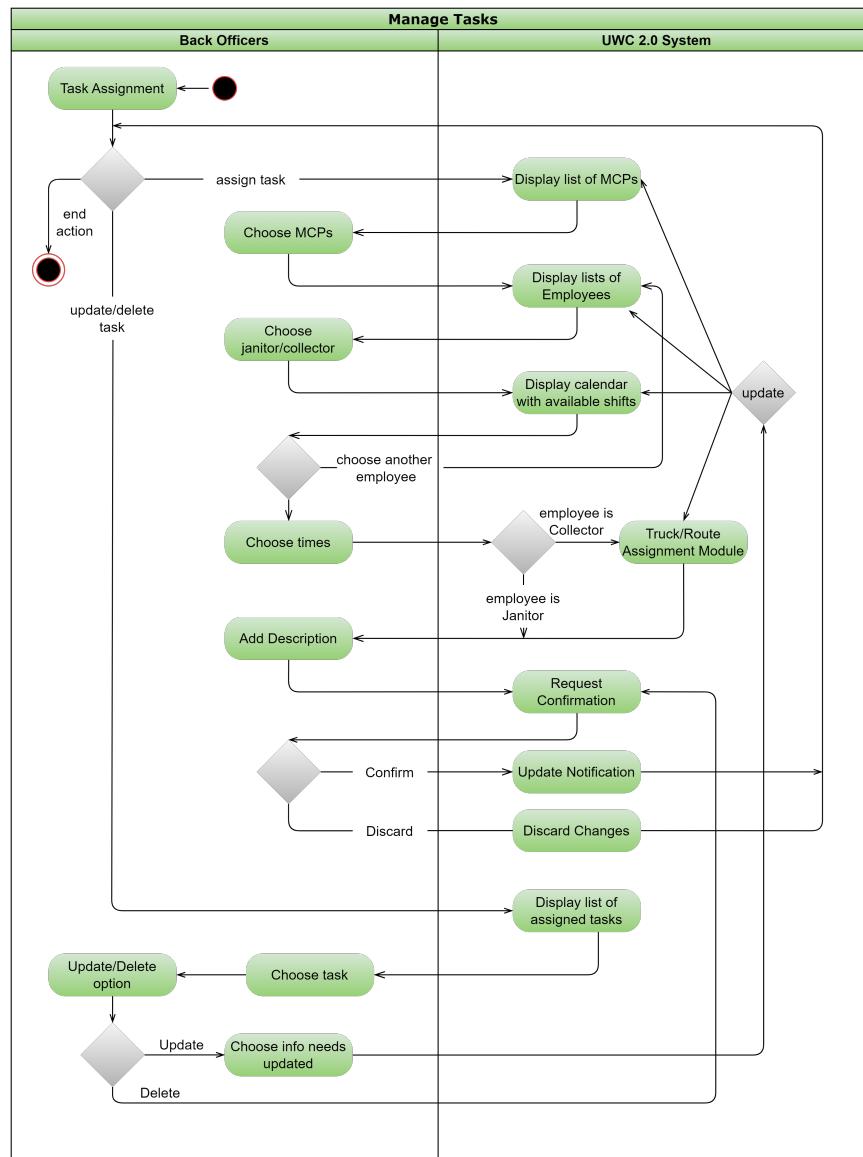


Figure 3. Overview of Task Management

Back officers are supposed to perform three main tasks in the Manage Tasks module: assign new tasks, update existing tasks, and delete existing tasks. Choosing an MCP or list of MCPs (with their information displayed) to give to a group of janitors or collectors is the first step

in creating a new task for back officers. Additionally, the system shows the chosen employees' calendar availability. Officers can go back and choose different workers. Back officers are given a summary of suggested routes (customizable) after selecting the available shifts, as well as a list of trucks and their condition (if Collectors were selected). Back officers may include a thorough description of a new task. After receiving confirmation from back officers, the system either updates the modifications and notifies the responsible employees, or it discards this task. The system offers a list of assigned tasks with a comprehensive description that can be chosen to make updates or deletion. The system will navigate to the precise stage that needs to be changed after the back officers choose which action to change. The system keeps track of all the modifications and requests authorization from back officers before updating employee alerts.

2.1.2 Check Calendar

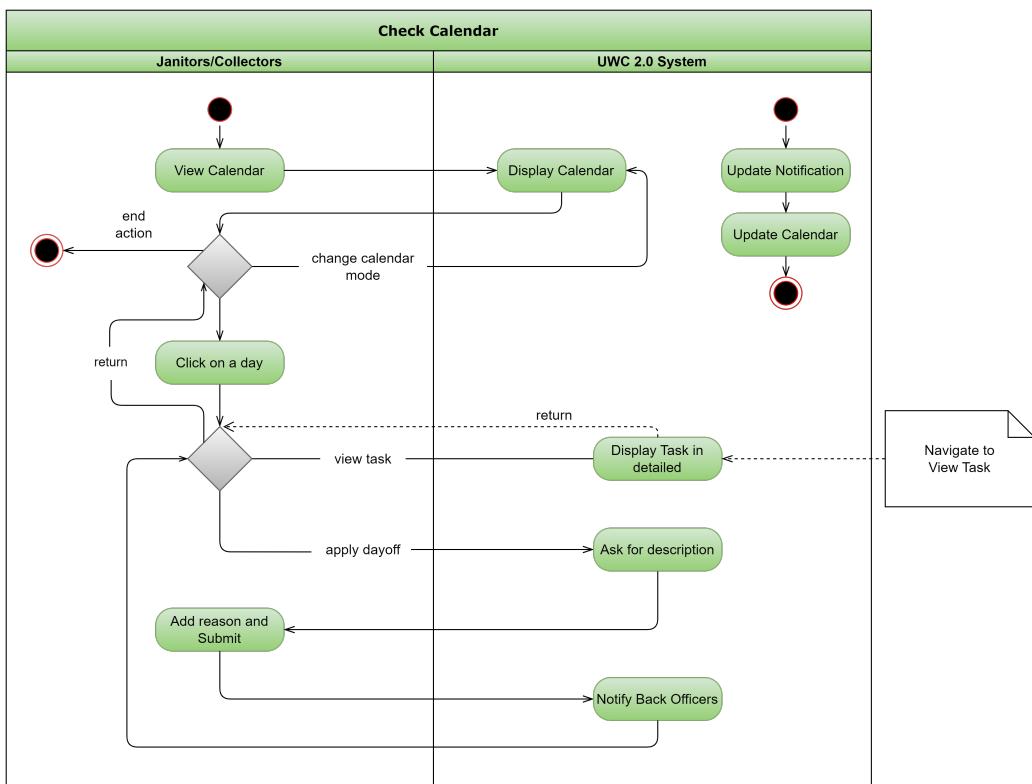


Figure 4. Check calendar

Employees have access to a calendar that shows highlighted work hours and can switch between weekly and monthly view (default weekly). When employees select a day, a task summary for that day is presented (if it is available), and they can also access complete detailed information about the task. They can also submit applications for days off with reasoning.

2.1.3 View Task

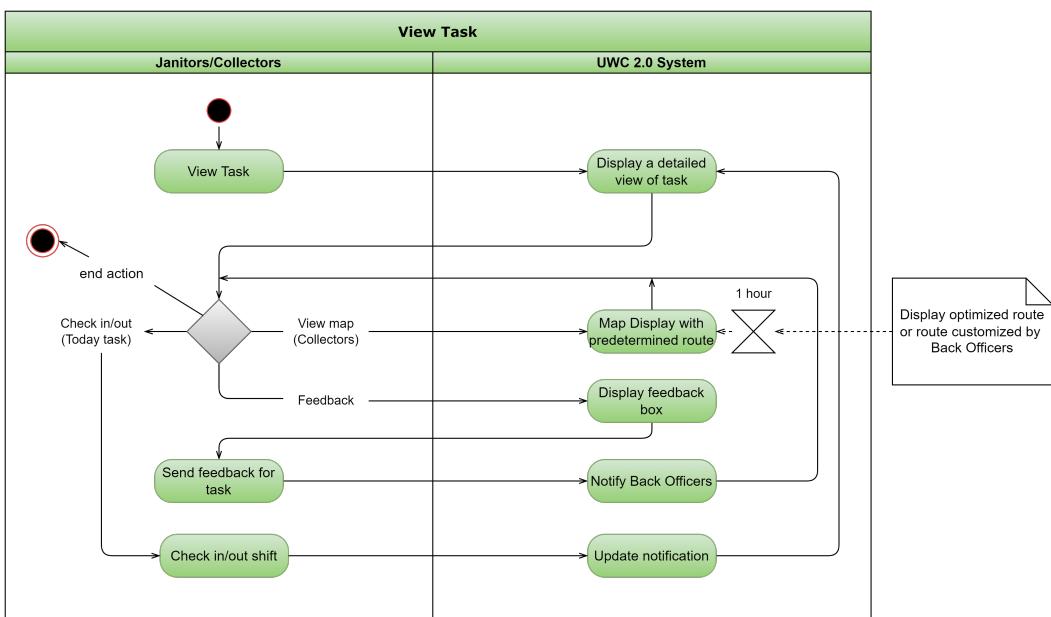


Figure 5. View task

Employees can examine detailed information about a task by navigating to it (from the calendar or notifications). Collectors have access to a map of their journey (re-optimize path every hour). Every employee can give feedback to the task they received, so back officers can later make changes in the Manage Tasks module. They will be able to check in and out during their shifts, and notifications will prevent them from missing duties.

2.2 Sequence Diagram

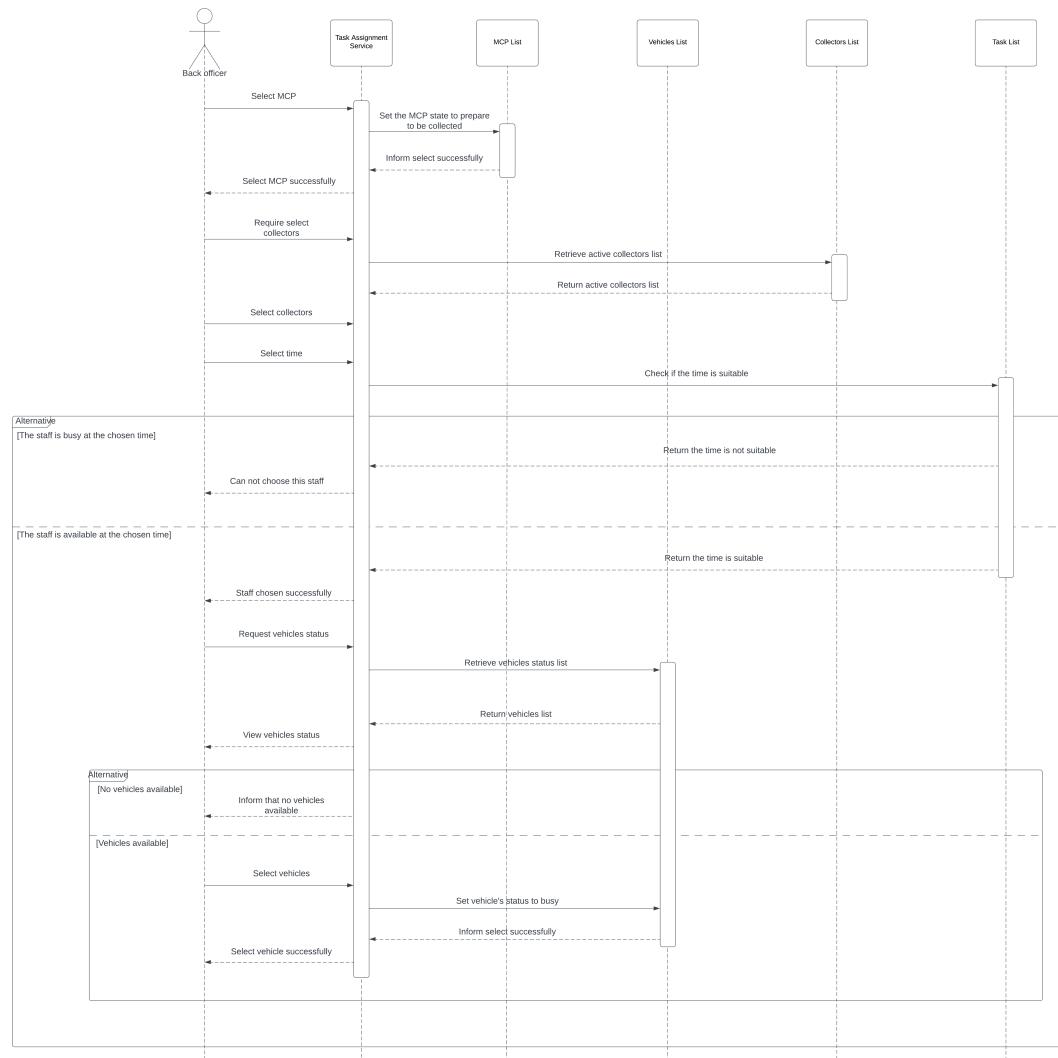


Figure 6. Sequence diagram of Vehicle Assignment module

For further inspection [Click here](#).

Brief description: At first, the back officer will select the MCP that needs to be collected. Then, he needs to select the collectors through the collectors' list. Next, he will set the time and shift for that chosen staff. Finally, the back officer needs to select the vehicle.

2.3 Class diagram

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

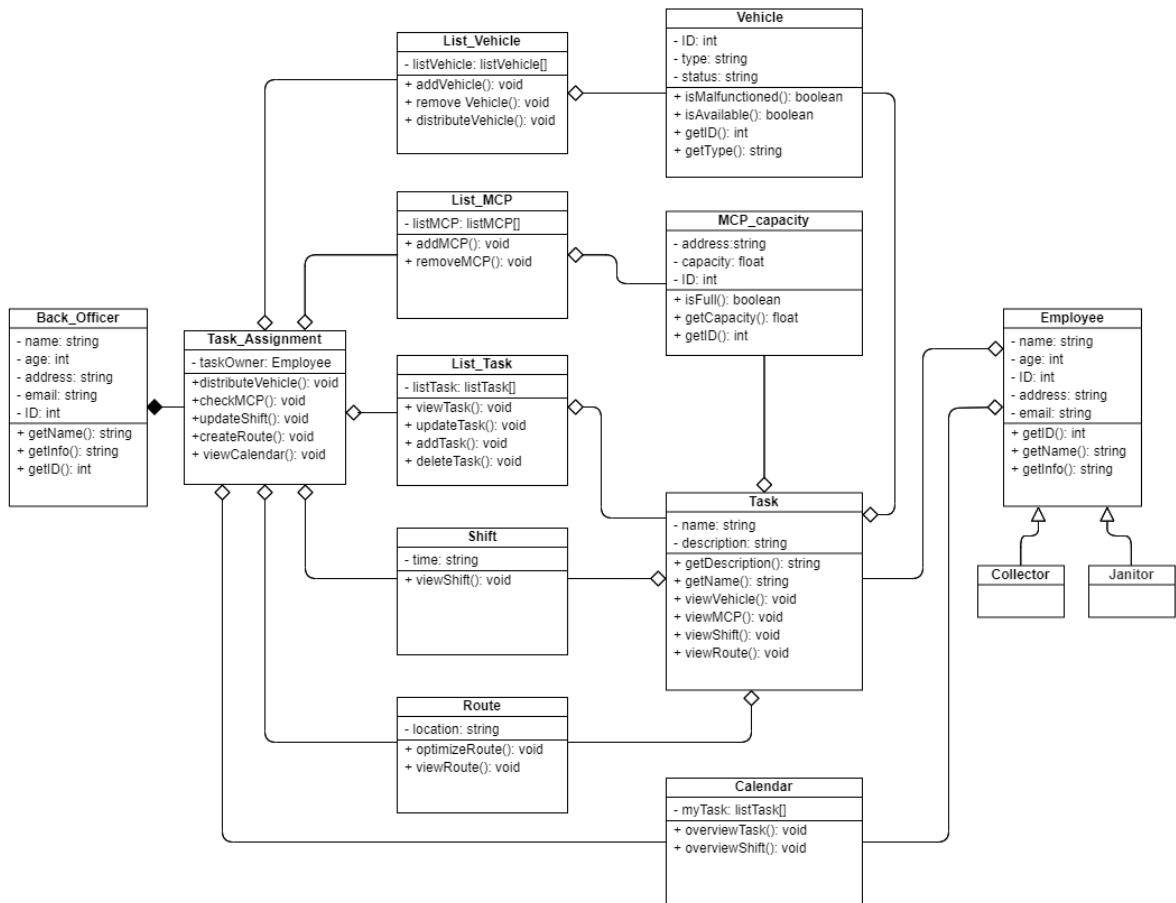


Figure 7. Class diagram of Task Assignment module

For further inspection, [Click here](#).



2.4 User interface (Desktop-view for back-officers)

In this section, our team use Figma to design. For inspection and presentation of the wireframe, [Click here](#)

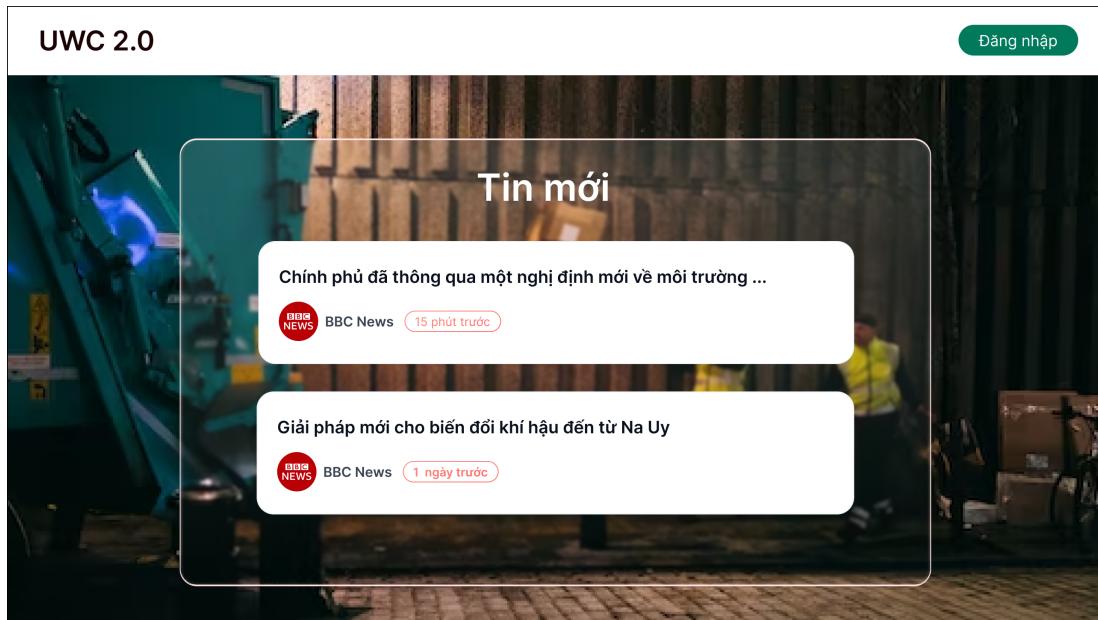


Figure 8. Page view when not signed in

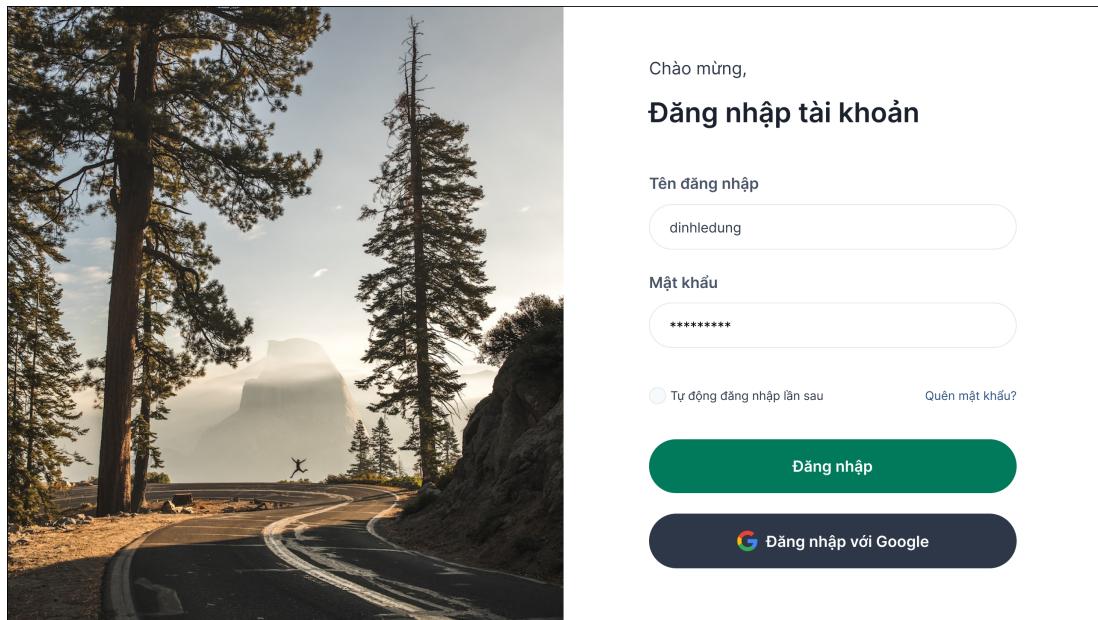


Figure 9. Login page

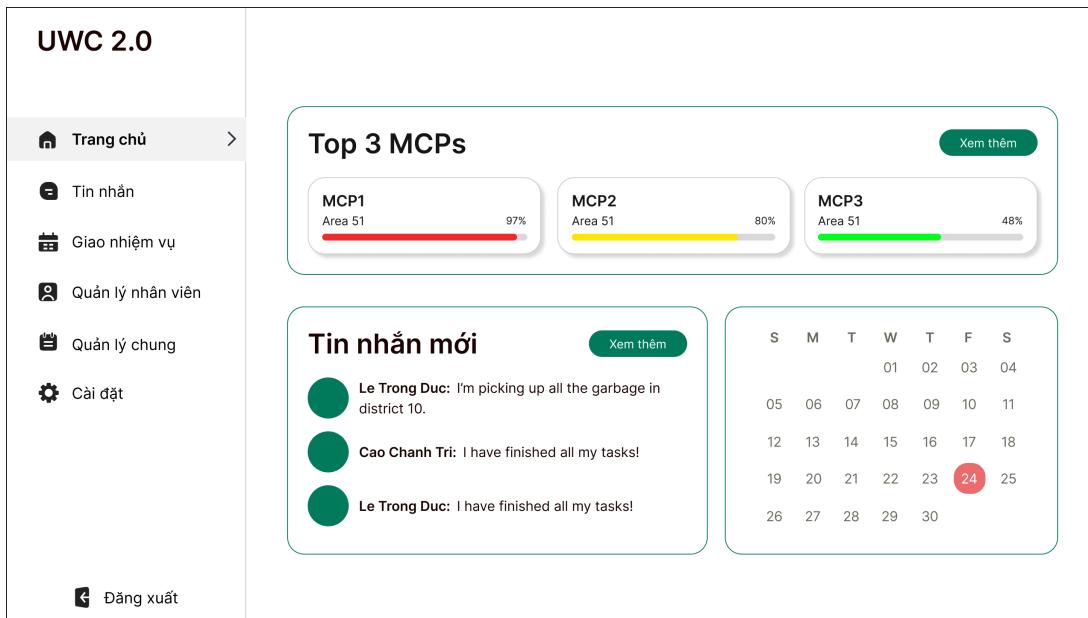


Figure 10. Homepage

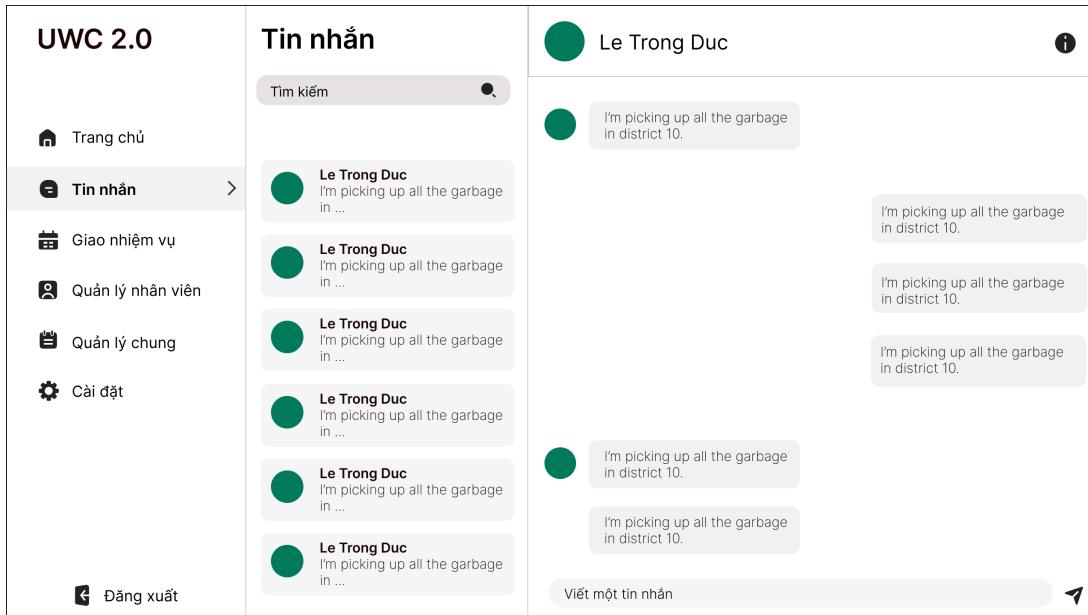


Figure 11. Chatting page



UWC 2.0

- [Trang chủ](#)
- [Tin nhắn](#)
- [Giao nhiệm vụ >](#)
- [Quản lý nhân viên](#)
- [Quản lý chung](#)
- [Cài đặt](#)

MCP Thêm địa điểm

Nhân viên Nhân viên thu gom Thêm nhân viên

Lịch trình Thêm lịch trình

Lời nhắn Thêm lời nhắn

Gửi nhiệm vụ

[Đăng xuất](#)

Figure 12. Task assignment page

UWC 2.0

- [Trang chủ](#)
- [Tin nhắn](#)
- [Giao nhiệm vụ](#)
- [Quản lý nhân viên >](#)
- [Quản lý chung](#)
- [Cài đặt](#)

Nhân viên Tim kiếm Nhân viên thu gom

Le Trong Duc	Le Trong Duc	Le Trong Duc
Mã số: 216763	Mã số: 216763	Mã số: 216763
Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1
Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1
Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1 	Đang làm việc Khu vực 1

[Đăng xuất](#)

Figure 13. Staff list page



UWC 2.0

- Trang chủ
- Tin nhắn
- Giao nhiệm vụ
- Quản lý nhân viên
- Quản lý chung >
- Cài đặt
- Đăng xuất

Quản lý MCP ▾

Tim kiếm

Tổng cộng: 50

MCP1
Area 51 97%

MCP1
Area 51 80%

MCP1
Area 51 48%

MCP1

Khu vực 1

Địa điểm: 268 Ly Thuong Kiet Street, Ward 14, District 10, Ho Chi Minh City, Vietnam.

Trạng thái: 97%

Figure 14. MCP management page

UWC 2.0

- Trang chủ
- Tin nhắn
- Giao nhiệm vụ
- Quản lý nhân viên
- Quản lý chung >
- Cài đặt
- Đăng xuất

Quản lý phương tiện ▾

Tim kiếm

Tổng cộng: 50

Đang hoạt động: 10

Xe 1
Mã số xe: #PT1234
Tuyến đường: MCP1 → MCP2 → MCP3

Xe 2
Mã số xe: #PT5678
Tuyến đường: MCP1 → MCP2 → MCP3

Xe 1 #PT1234 Đang hoạt động

Sức chứa: 10000 kg

Nhiên liệu: 18 / 22 L

Tuyến đường

Figure 15. Vehicles management page



The screenshot shows the 'UWC 2.0' application interface. On the left, there is a sidebar with the following menu items:

- Trang chủ
- Tin nhắn
- Giao nhiệm vụ
- Quản lý nhân viên
- Quản lý chung
- Cài đặt >
- Đăng xuất

The main content area is divided into two sections:

Tài khoản

Họ và tên	Đinh Lê Dũng	
Tên đăng nhập	dinhledung	
Mật khẩu	*****	

A large green circular placeholder for a profile picture is visible.

Thông tin cá nhân

Email	dinhledung@gmail.com		Số điện thoại	0326968305	
Ngôn ngữ					

Figure 16. Setting page

This marked the end of task 2.

3 Architecture design

3.1 Task 3.1

3.1.1 Layered Architecture

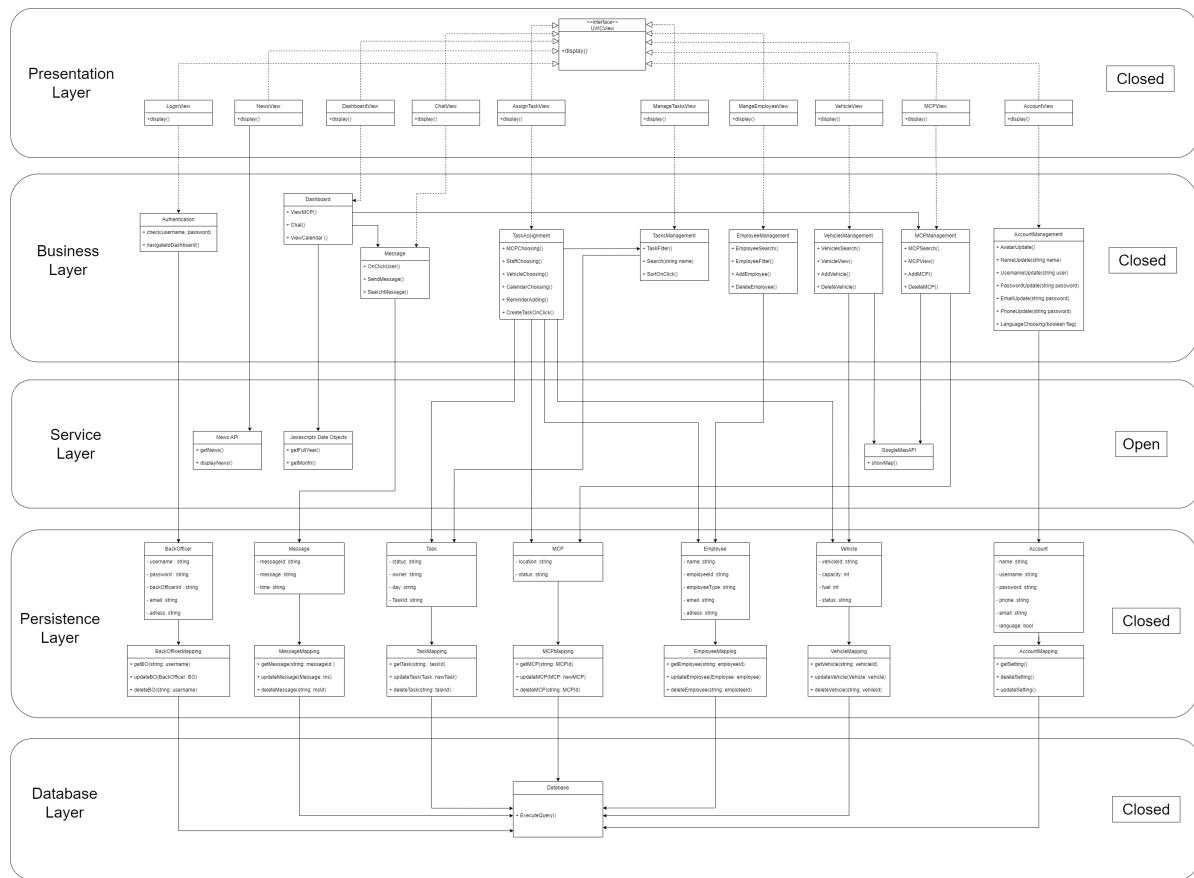


Figure 17. Layered Architecture

For further inspection, [Click here](#)

3.1.2 Presentation strategy

For the Presentation strategy, we develop about 10 pages, which include Login view, News view, Dashboard view, Chat view, Assigning tasks view, Managing tasks view, Manage employees view, Vehicle view, MCP view, Account view.

Our goals are:

- Develop a responsive UI design
- Use interactive data visualizations to present complex information in an easy-to-understand format.



- Incorporate multimedia elements such as images, videos, and animations to enhance the user experience.
- Use color psychology and typography to create a compelling visual hierarchy that guides users through the content.
- Offer users the ability to customize the interface to their personal preferences, such as color schemes (switching between light and dark).

3.1.3 Data storage approach

We choose to store data hardcoded on a website using JavaScript involves declaring variables in the code and assigning values to them. This can be done for a variety of data types, including strings, numbers, arrays, objects, and more. We will save the data hardcoded as tasks, users data, staff list, maps location, e.t.c

To store data hardcoded on a website using JavaScript, you would typically follow these steps:

1. Identify the data that you want to store and decide on the appropriate data type(s) to use.
In this case, it is new-create tasks, users data, staff list, maps location, e.t.c
2. In JavaScript section of your website and declare a variable using the appropriate syntax for the chosen data type.
3. Assign the desired value(s) to the variable.
4. Repeat steps 2 and 3 as needed for additional variables.
5. Save your changes and reload your website to access the stored data.

While hardcoding data in this way can be useful for small datasets or simple applications, it is generally not recommended for larger datasets or more complex applications, as it can be difficult to manage and update the data. In those cases, it is usually better to use a database or other data storage solution.

3.1.4 API management

Accessing external APIs from a website involves several steps, and here is a general overview of the steps involved:

1. Identify the API: Firstly, we have to identify the external API we want to use. This can include researching several APIs and selecting one that meets our needs. In this case, we use external APIs for two modules: map and update MCPs' status.
2. Read the API documentation: Reading the API's documentation is very helpful. Information on the request and response formats, authentication and authorization procedures, error handling, and API endpoints should all be included in the documentation.
3. Obtain an API key: For request authentication, many APIs demand an API key or token. To get an API key, if required, we should adhere to the directions in its documentation.
4. Choose a programming language or framework: We can select a programming language or framework to access the API depending on the website and the API. Python, JavaScript, and PHP are common choices. In this instance, we proceed using JavaScript.



5. Make API requests: We have to create a request and submit it to the API endpoint with the necessary arguments in order to send a request to the API. We use `fetch()` function in JavaScript to make API requests. It is frequently used to get data in JSON format and make API requests.
6. Handle API responses: Once we have made a request to the API, we will receive a response. We have to handle the response according to the API documentation, and handle any errors or exceptions that may occur. The data that is being fetched is in JSON format, a lightweight data exchange format that is simple for both humans and machines to read, write, and parse. The syntax is based on a subset of JavaScript, and it uses key-value pairs with keys that are always strings and values that can be strings, numbers, objects, arrays, booleans, or null. Complex data structures may be created by nesting the values.
7. Integrate the API response into our website: Once we have successfully accessed the API and received a response, we can integrate the response data into your website as needed.

Overall, using external APIs for a website requires careful preparation, comprehension of the API documentation, and appropriate usage of programming languages and tools. It's crucial to guarantee the security, dependability, and handling of API requests and answers as well as any potential mistakes or exceptions.

3.2 Component diagram

3.2.1 Component diagram

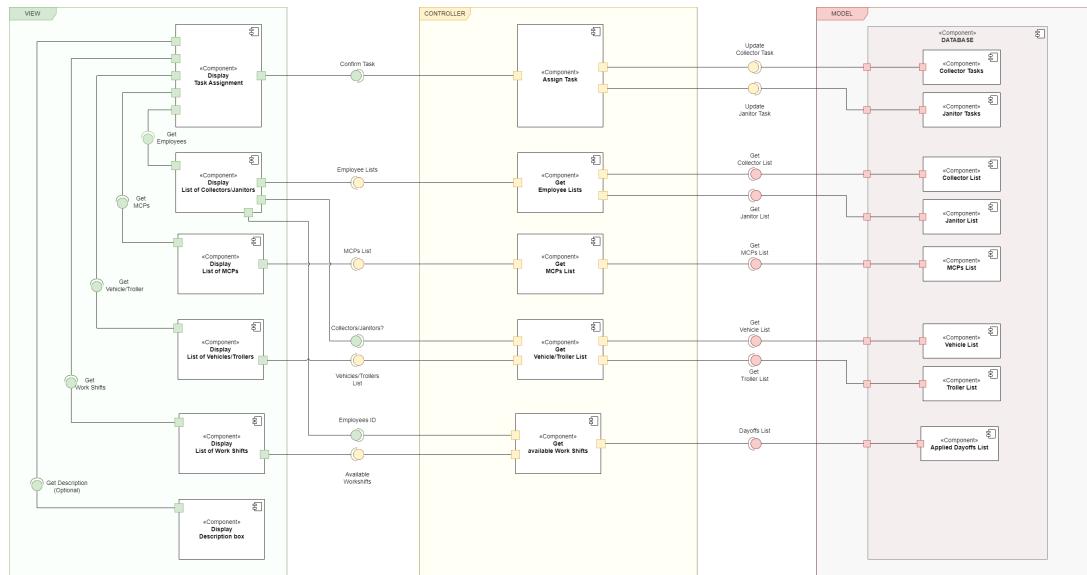


Figure 18. Component Diagram

For further inspection, [Click here](#)

3.2.2 Description

Task Assignment and Task Management are the two main components of the Task Distribution module. Lists of Employees, MCPs, Vehicles/Trollers, and available Work Shifts are all displayed within the Task Assignment page with the aid of the corresponding controller components. Information about all previously created jobs and job deletion functions are displayed in Task Management.

After Back Officers select a Collector or a group of Janitors for Task Assignment, the name of the employee(s) will be forwarded to controller components, which will decide what information to request from the database, trucks or trollers, and the available work shifts. The Assign Task component will be in charge of updating the relevant table in the database after Back Officers have finished making a task.