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

SOFTWARE ENGINEERING (CO3001)

Assignment

Urban waste collection aid
UWC 2.0

Advisor: Quản Thành Thơ - Nguyen Duc Anh
Student: Bùi Thái Dương - 1852307
Nguyễn Ngọc Hưng - 2053075
Đặng Quốc Huy - 2053031
Nguyễn Lê Thanh Phúc - 2052656
Nguyễn Việt Thắng - 2052719

Memberlist and Workload

No.	Fullname	Student ID	Contribute	Percentage of Work
1	Đặng Quốc Huy	2053031	Task 1: 1.1 Task 2: 2.2 Task 3: 3.1 Task 4: 4.1, 4.3 Task 5: 5.2 + Vehicle, MCP view page	18%
2	Nguyễn Ngọc Hưng	2053075	Task 1: 1.1, 1.2, 1.3 Task 2: 2.2 Task 3: 3.1 Task 4: 4.1, 4.2, 4.3 Task 5: 5.2 + Task module + Login module + Vehicle, MCP view page	25%
3	Nguyễn Lê Thanh Phúc	2052656	Task 1: 1.3 Task 2: 2.1 Task 3: 3.2 Task 4: 4.3 Task 5: 5.2 + Workers view module	19%
4	Nguyễn Việt Thắng	2052719	Task 1: 1.3 Task 2: 2.1 Task 3: 3.2 Task 4: 4.3 Task 5: 5.2 + Workers view module	19%
5	Bùi Thái Dương	1852307	Task 1: 1.3 Task 2: 2.3 Task 3: 3.1 Task 4: 4.3 Task 5: 5.2 + Message module	19%

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

Some of the relevant stakeholders of the system:

- *Organization X's system manager:* IT staff who operate and maintain the system (website or server engineer).
- *Service provider Y's supervisor:* oversee all operations to ensure that the system runs smoothly.
- *Back officers:* operate a central system to create calendars, coordinate, and send messages to collectors and janitors.
- *Collectors and janitors:* receive tasks and check in/out tasks, send and receive messages.

Their current needs

Waste management and collection are more economical and practical.

Their current problems could be

- Solid waste management is costly and ineffective
- Operational inefficiencies of services
- Limited utilization of recycling activities
- Inadequate management of non-industrial hazardous waste
- Inadequate landfill disposal
- Inadequate service coverage

Benefits UWC 2.0 will be for each stakeholder:

UWC 2.0 helps:

- Service provider Y supervisor to easily keep track of employees.

- Back officers to be able to assign tasks, have an overview of the workers and send/receive messages from other employees.
- Collectors and janitors to be well-informed about their tasks and routes.

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

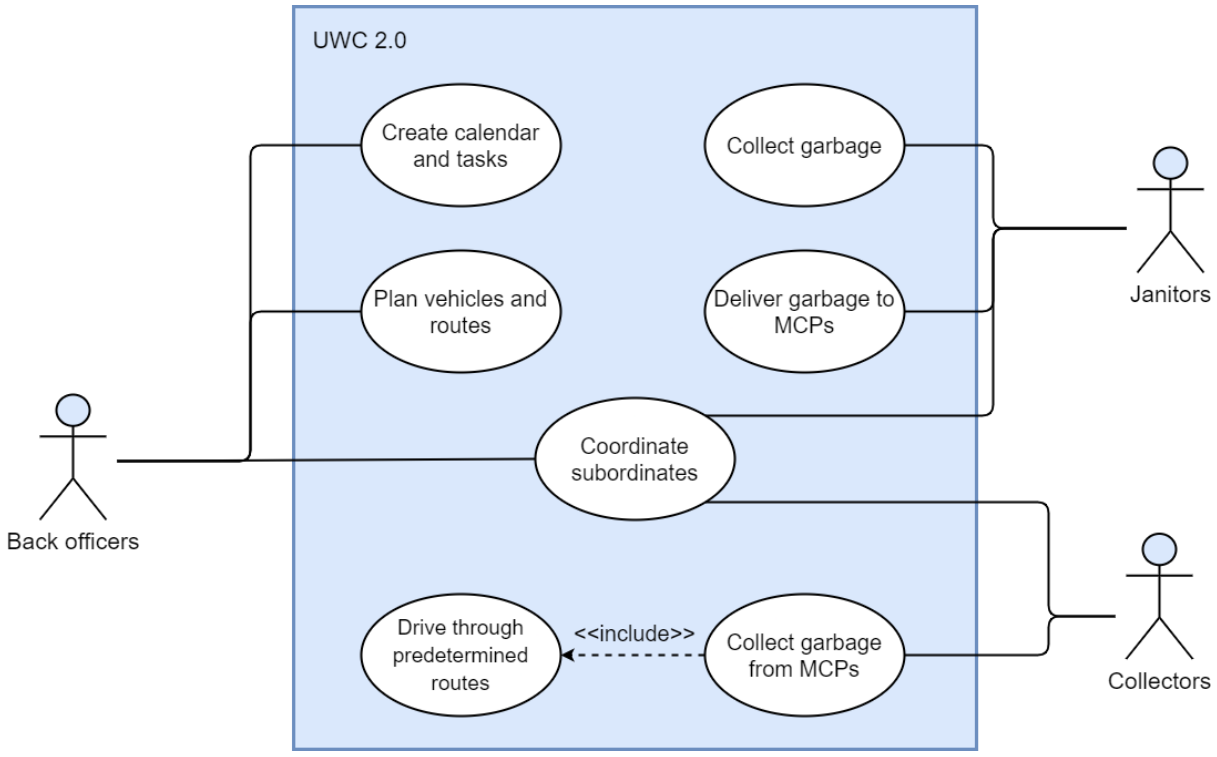
Functional requirements

- As an employee, I want the system to run every day I work in order to assign tasks and keep track of the assigned tasks.
- As a Service Provider Y manager, I want to be able to search for data about back officers, janitors, and collectors in order to keep track of their information.
- As an employee, I want the system to have a log-in function in order to identify the information of the other employees.
- As a back officer, I shall be able to search for information about collectors and janitors, information and the capacity of MCPs in order to ensure the suitability of tasks assigned.
- As a back officer, I shall be able to create and check collectors' and janitors' calendars in order to keep track of their work.
- As a back officer, I shall be able to look for vehicles and their technical details in order to plan the most suitable routes for the collectors.
- As a back officer, I shall be able to create and assign routes for the collectors based on vehicles' data in order to collect garbages from the MCPs.
- As an employee, I want the system to allow communication between back officers, collectors, and janitors (Realtime messages, Mails, etc) in order to send/receive the needed information.
- As a worker, I shall be able to check my working routine, tasks, and availability of MCPs in order to finish my job smoothly.
- As an employee, I want the system to have a task management system in order to help janitors and collectors view their tasks.
- As an employee, I want the system to have a report function for users in order to inform the problems to system managers of.

Non-functional requirements

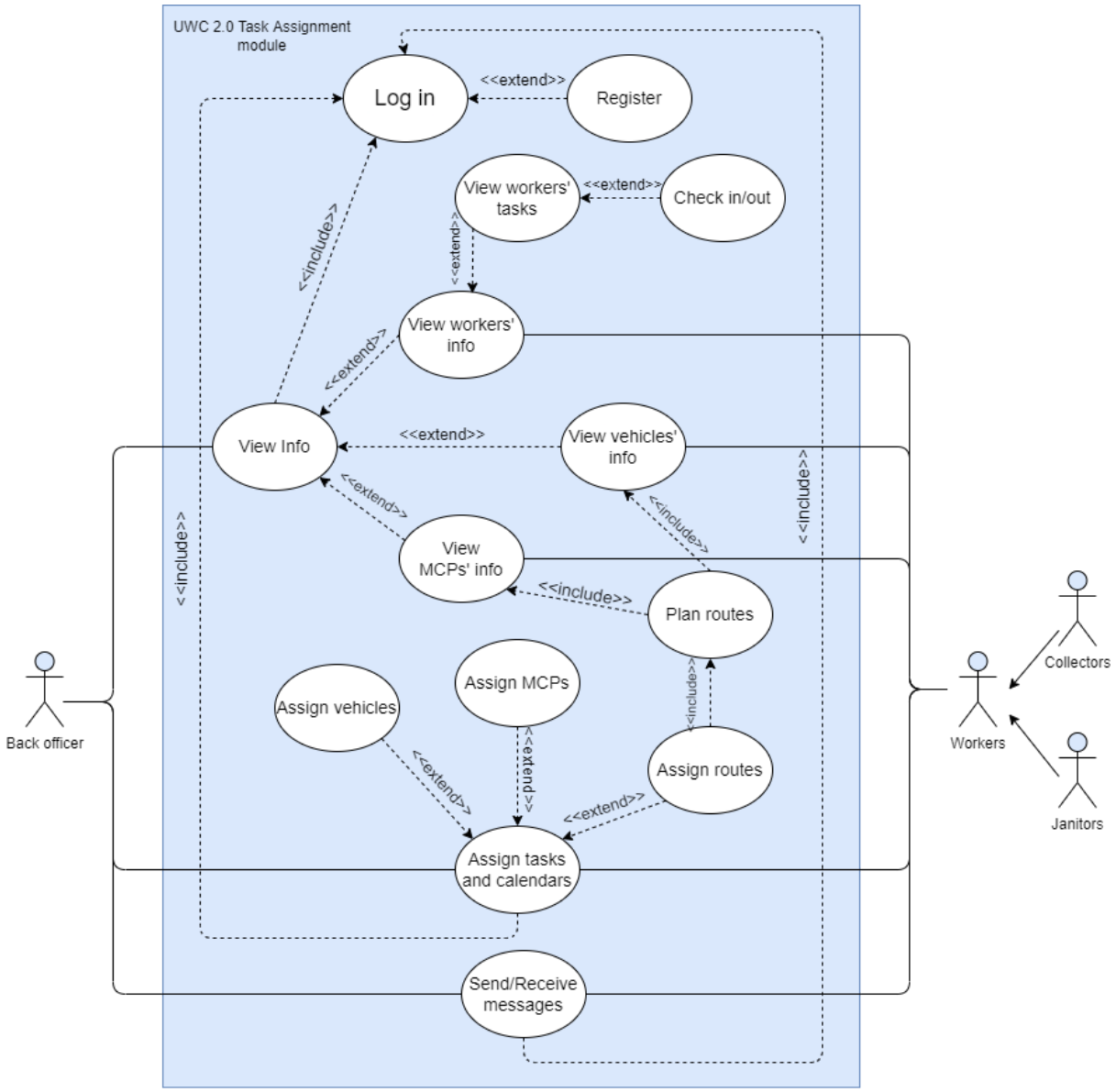
- As a back officer, I want the information should be updated from MCPs every 15 minutes with the availability of at least 95% of their operating time so that I can coordinate collectors to the right places.
- As an employee, I want real-time messages between back officers, collectors, and janitors that can't exceed 1 second of delay in order to have effective communications.
- As a worker, I want all critical information should be displayed in one view in order to have a detailed view of my tasks without scrolling down.
- As a manager, I want the system should be able to handle real-time data from at least 1000 MCPs at the moment and 10.000 MCPs in five years so that we ensure the steady development of the system.
- As a user, I want the system UI should be displayed in Vietnamese, with an opportunity to switch to English in the future.
- As a user, I must have the account provided by the organization to access the system in order to secure internal information.

Use-case diagram



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

Use-case diagram



Use-case table

Use-case 1: Log in

Use-case name	Log in
Actors	Workers, Back officers
Description	Employees should log in to the system to store their stats for the superiors to keep track of the subordinates. In order to view information, assign tasks and calendars or send/receive messages from others, employees shall be able to gain access by logging in to the system.
Trigger	Actors indicate that they want to log in to the system
Pre-condition	- Actor's account exists - Actor's account is authorized - Actor's device has an application and connects to the internet while logging in
Post-condition	- The actor successfully logs in
Normal Flow	1. User access to the system 2. User chooses a login and fills in username, password 3. The system verifies the credentials successfully and allows the user to access the application
Alternative Flow	2a. User chooses "Register" 2a1. Create a new account *Continue use-case 2-3*
Exception	3a. The system fails to verify the access and show the notification 3a1. Users decide to stop logging in. *End use-case* 3a2. Users choose "Register" *Come back to use-case 2a*

Use-case 2: View info

Use-case name	View info
Actors	Workers, Back officers
Description	As a back officer, I want to view all of the information about workers, vehicles, and MCPs in order to have an overview of my subordinates and assign tasks to them. As a worker, I can view and check in/out my tasks every day and have an overview of my assigned vehicles and collecting points.
Trigger	User indicates that he wants to view some information on the system
Pre-condition	- User logged in to the system. - The data that the user asked for is stored in the database of the system. - Users and the data they want to view have been authenticated.
Post-condition	- User successfully views the information
Normal Flow	1. Users access the system. 2. Users log in to the system. 3. Users choose what information to view from the system. 4. The system displays the information.
Alternative Flow	4a. The information that users looking for is not accurate or missing. 4a1. Users send a report of the problem to the system. 4a2. The system updates or sends an offer to the data owner to update the information. *End use-case*
Exception	2c. Users have not logged in to the system. 2c1. Users log in to the system. *Continue use-case 3-4*
Non-functional Requirements	NFR-1. The information of MCPs should be updated every 15 minutes with the availability of at least 95% of their operating time. NFR-2. The tasks of the workers should be displayed in

	one view.

Use-case 3: Assign tasks and calendars

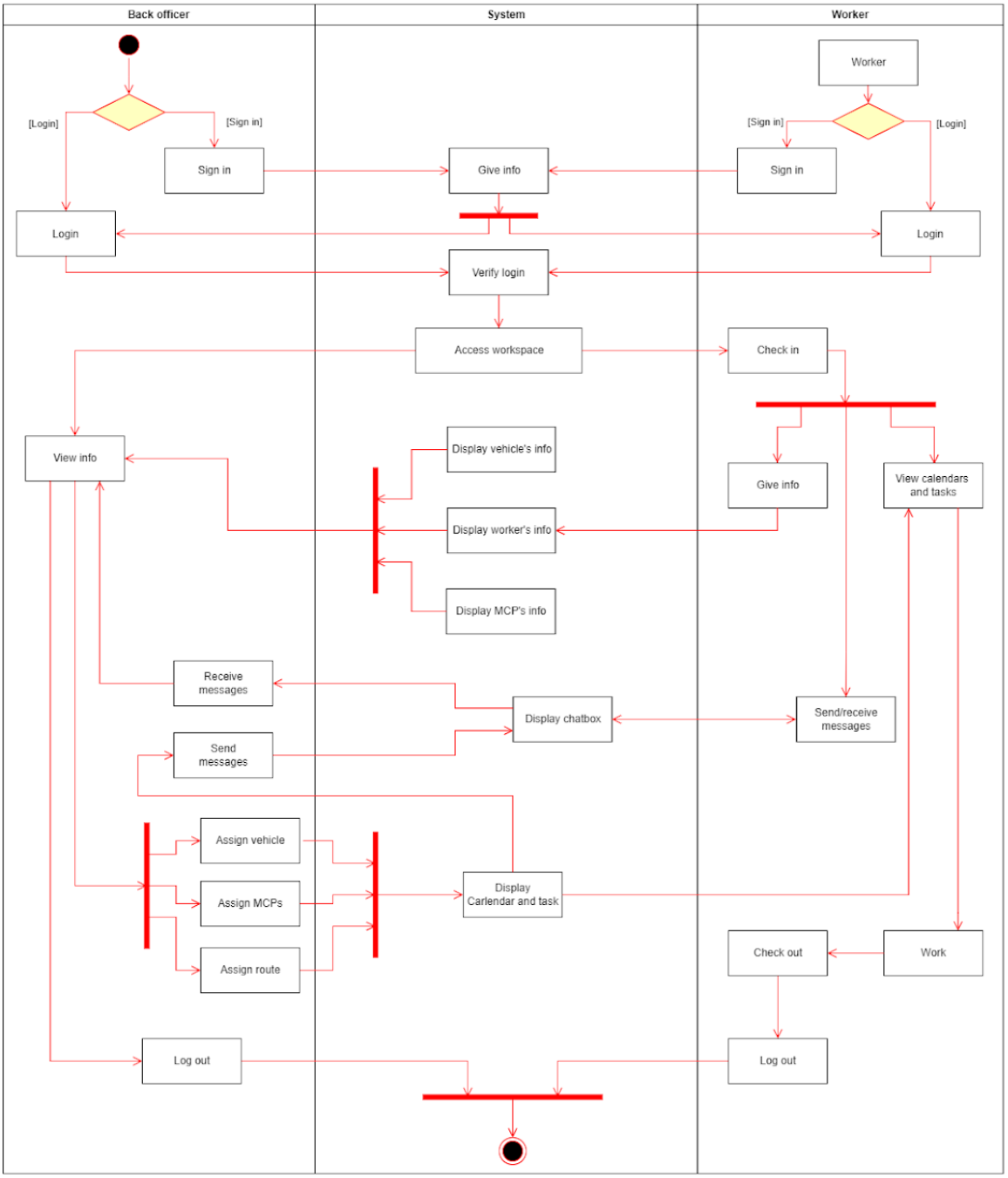
Use-case name	Assign tasks and calendars
Actors	- Primary actor: Back officers - Secondary actor: Workers
Description	Back officers give information about tasks and calendars to workers so they can view and receive routes, MCPs, and vehicles.
Trigger	Back officers indicate that they want to assign tasks and calendars to workers.
Pre-condition	- Users have an account logged in to the system. - Tasks and calendars are authorized.
Post-condition	- Back officers successfully assign tasks and calendars to workers. - Workers get and be able to view their assigned tasks.
Normal Flow	1. Users access the system. 2. Users log in to their accounts. 3. Back officers start to assign tasks to their subordinates. 4. Workers receive notifications that tasks have been successfully assigned.
Alternative Flow	4a. Workers have not received any tasks 4a1. Back officers check if the tasks assigned are valid or not. 4a2. Back officers resend tasks. Workers check again and confirm that the given tasks are successfully received. *End use-case*
Exception	2c. Users have not logged in to the system. 2c1. Users log in to their accounts. *Continue use-case 3-4*

Use-case 4: Send/Receive messages

Use-case name	Send/Receive messages
Actors	Workers, Back officers
Description	- Sending/Receiving messages is a function that allows employees to send announcements to the subordinates immediately and check whether they have read the messages or not. Besides, workers can create groups for private conversations via this function.
Trigger	Users indicate that they want to send or view received messages.
Pre-condition	- Users own at least one smart device. - Users logged in to their accounts. - Users turned on notifications about new messages.
Post-condition	- Users know that their messages have been sent successfully and whether the receivers have read the message or not.
Normal Flow	1. Users access the system. 2. Users log in to their accounts. 3. Users text new messages and choose which accounts they want to send. 4. Users receive notifications that the messages were sent.
Alternative Flow	4a. The receivers have not received any message. 4a1. The receivers check their Internet connection and reload the mailbox. 4a2. The receivers read the message and confirm to the system that the messages were sent successfully. *End use-case*

Exception	2a. The users' session has expired. 2a1. Users log in to their accounts again. *Continue use-case 3-4*

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



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

The **back officers** must have successfully logged into the app with their account and password.

The **back officer** will manage and assign tasks for **collectors** based on:

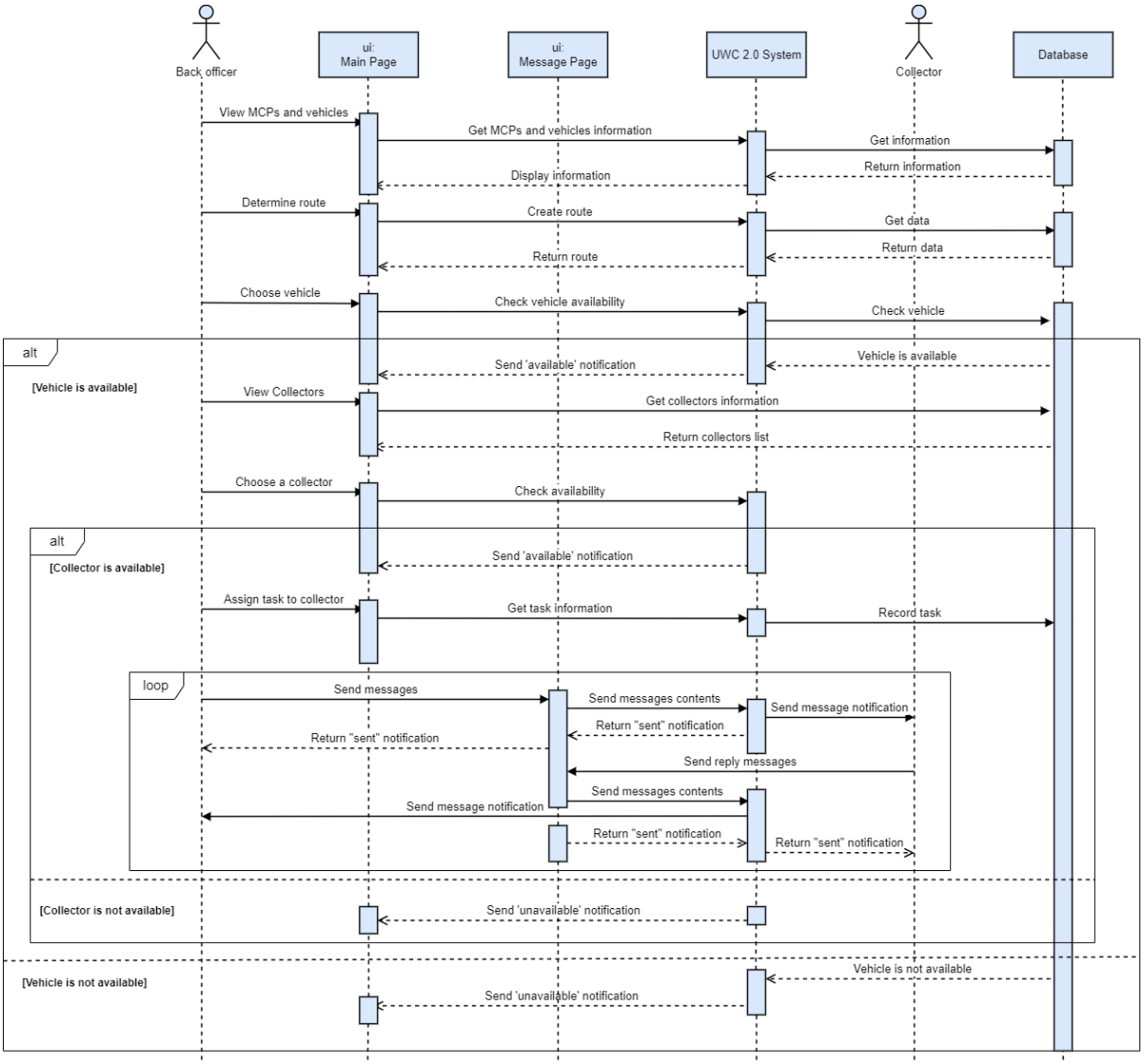
- Available vehicles and their technical details.
- The capacity of the MCPs at the moment and the distance from the **collectors** to the MCPs. (Updated every 15 minutes with the availability of at least 95% of their operating time)
- **Collectors'** information. The **back officer** will check the vehicle and assign it to each **collector**. Then, according to the MCPs' capacity at the moment, the **back officer** will have the system to create a route for each **collector** by optimizing fuel consumption and travel distance. After that, the task messages will be sent to **collectors** by the **back officer** through the message module and the **back officer** can watch their process in case of any problems.

When a **back officer** requests to see the **collectors'** information from the database, a list of **collectors** will be displayed including:

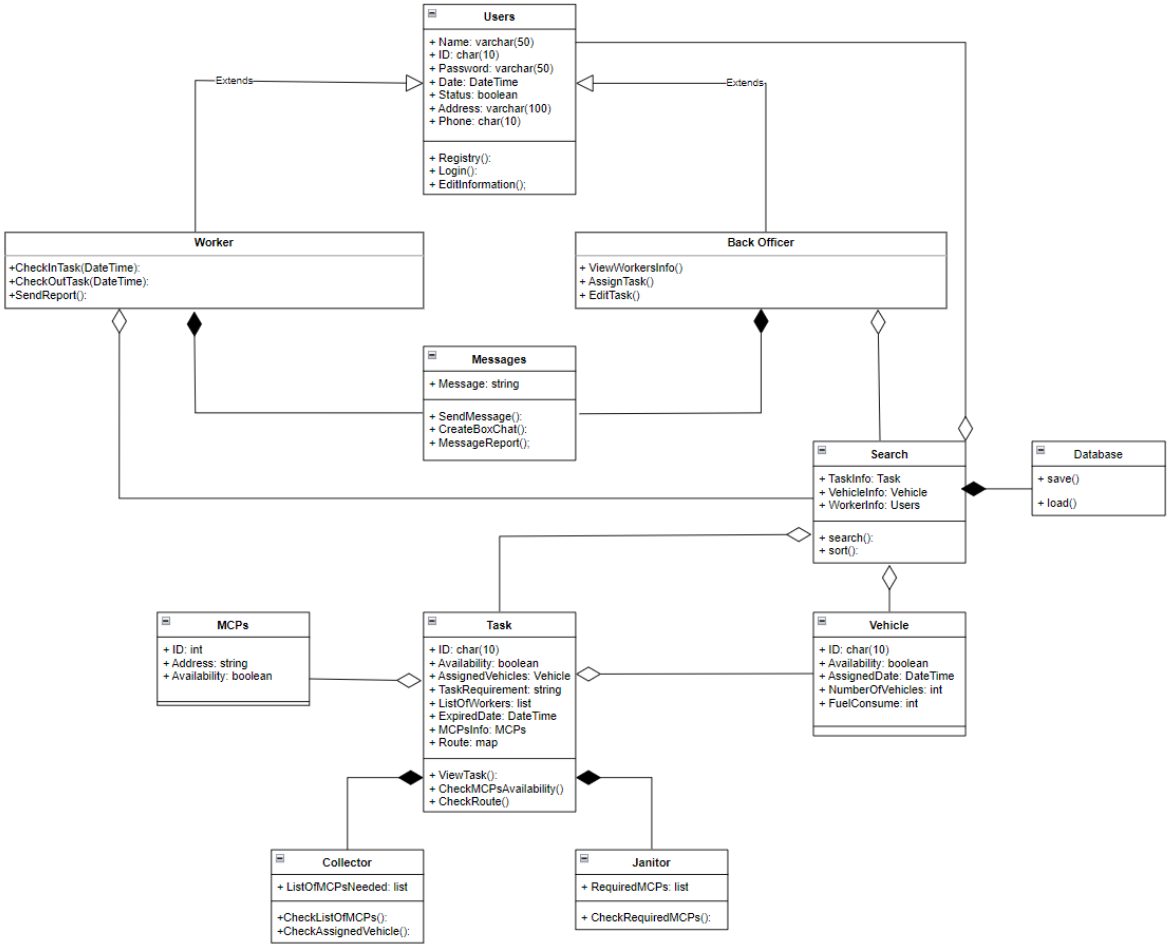
- **Collectors'** work calendar and their contact information for emergency cases.
- Availability of **collectors** at the moment.
- Whether the **collectors** check in/check out on time. They will be chosen if they have their shift or need to implement their KPI that day.
- Assign route form to each **collector**.

If there are any problems(broken vehicle, traffic jam, etc.) or **collectors** are not available at the moment. The system will send messages to the **back officer** so as to change the plan.

Sequence Diagram:



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



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

3.1.1 Introduction to MVC pattern

Definition: Standing for "Model-View-Controller.", MVC is an application design model consisting of three interconnected parts. They include the model (data), the view (user interface), and the controller (processes that handle input). MVC is commonly used for developing modern user interfaces, providing fundamental functions to design programs for web, mobile or even desktop applications.

Below is the description of each aspect of MVC:

1. Model

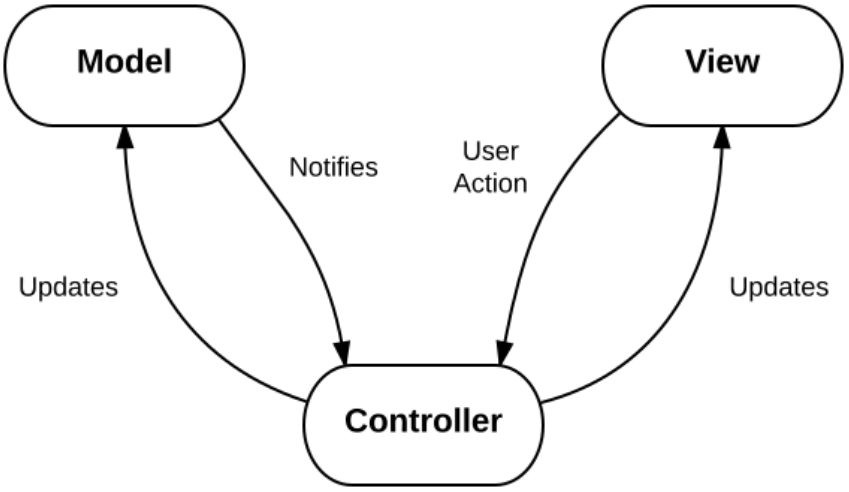
A model is data used by a program. This may be a database, file, or a simple object, such as an icon or a character in a video game.

2. View

A view is the means of displaying objects within an application. Examples include displaying a window or buttons or text within a window. It includes anything that the user can see.

3. Controller

A controller updates both models and views. It accepts input and performs the corresponding update. For example, a controller can update a model by changing the attributes of a character in a video game. It may modify the view by displaying the updated character in the game.



Pros and cons:

- **Pros:**

- Faster development process
- Ability to provide multiple views
- Support for asynchronous technique
- Modification does not affect the entire model
- MCV model returns the data without formatting
- SEO friendly Development platform

- **Cons:**

- The complexity is high to develop the applications using this pattern.
- Not suitable for small applications which have adverse effects on the application's performance and design.
- In terms of servlet and JSP, both often contain business logic and presentation tier.
- The isolated development process by UI authors, business logic authors and controller authors may lead to delays in their respective modules' development.

3.1.2 Applying to our model

To be more specific, we would like to explain some features of MVC architecture that we applied to our system:

Model: takes the data of the following classes from the database:

- *Account:* Each user will have their own account to log in to, the account will make it easier for them to assign tasks, receive notification about tasks, send/receive messages, view information,... Users must log in before interacting with the system.

- *Message:* The system tracks text messages by having message ID, contents, and time sent,... for each message sent by users.

- *MCPs:* Each MCPs has a unique ID with an address that helps the back officers and the workers to locate them to create route or to follow the route. Besides, there's also the availability of MCPs which is being updated every 15 minutes.

- *Vehicle:* Have vehicle ID, Name, availability, and fuel consumption for that back officers can choose the vehicle that suits the route.

- *Task:* Contain Task ID, requirements, responsible worker, route, date and time. Back officers can use this information to track the progress of tasks, workers can view this information, check in and out tasks while finishing their job.

View: the main graphical user interface to interact with the system. Here we break the view into many different view components, each for a different purpose:

- *Account view:* If users have not logged into the system, there will be a log-in box for the users to type their account name and password. If users are logged into the system, there will be a website interface that shows the accounts' information such as Name, Situation, DoB, Status, Address, Phone,....

- *Message view:* This website interface allows users to open a message box, choose another user to send message, send and view messages, report problems.

- *Vehicle view:* An interface that shows the vehicles' information: vehicle ID, Name, availability, and fuel consumption.

- *MCPs view:* Displays MCPs information: ID, Address, and availability.

- *Task view:* The system will have a form for the back officers to send tasks to specific workers based on the information about Vehicles and MCPs. There will be a Task review section for the back officers to review the tasks they sent.

Controller: Controller has the duty to identify user requests and render the corresponding view. The controller will contain all the logic flow of the UWC 2.0 system:

- *User controller:* The user controller will handle log-in and sign-up functions, manipulating account information. The user controller also has the duty to gather account data from Model and render account information view.

- *Task controller:* The task controller will handle task creation and sending requests from back officers and render task review through the task view component. This also helps record tasks into the database through the model component.

- *Message controller:* The message controller will handle message sending requests including chat box opening, message sending, and message reporting.

In our model, we include 3 modules:

Log-in:
The log-in module has the duty to allow users to provide their information, and helps the system manager and the back officers to track their working routine.

- Controller: User controller
- Input: E-mail, password
- Output: Account information view

Task Assignment:
The Task Assignment allows back officers to view the information of Vehicles, MCPs, workers and then create and send tasks to the workers through a submission form.

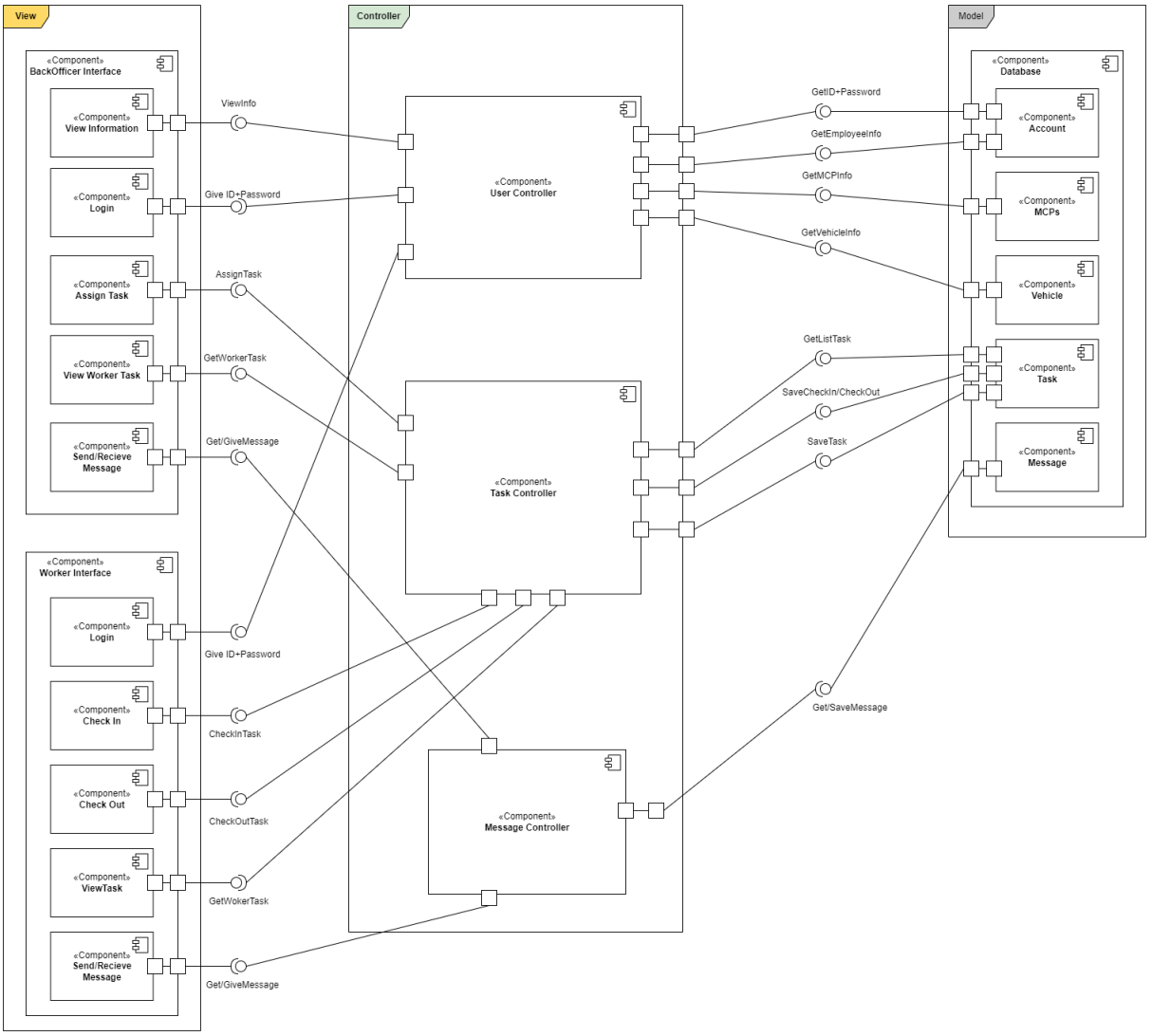
- Controller: Task controller
- Input:
    - Route, Vehicles, MCPs, Responsible worker, Task requirements
- Output: Tasks review

Send Message:
This module allows users to send messages to others, receive and view messages and report to the system manager when a system error occurs.

- Controller: Message controller
- Input:
    - Send: text messages
    - Report: report reason
- Output: Message view: Chatbox, sent and received messages

3.2 Draw an implementation diagram for the Task Assignment module

3.2.1 Component Diagram:



The system comprises three main sub-systems: View, Control, and Model.

The View component contains the Back_officer_Interface and Worker_Interface:

- The BackOfficer Interface component provides the interface for the back office to log in, view worker information, send/receive messages, assign and view worker tasks.
- The Worker Interface component provides the worker's interface to log in, check in, check out, send/receive messages and view worker tasks.

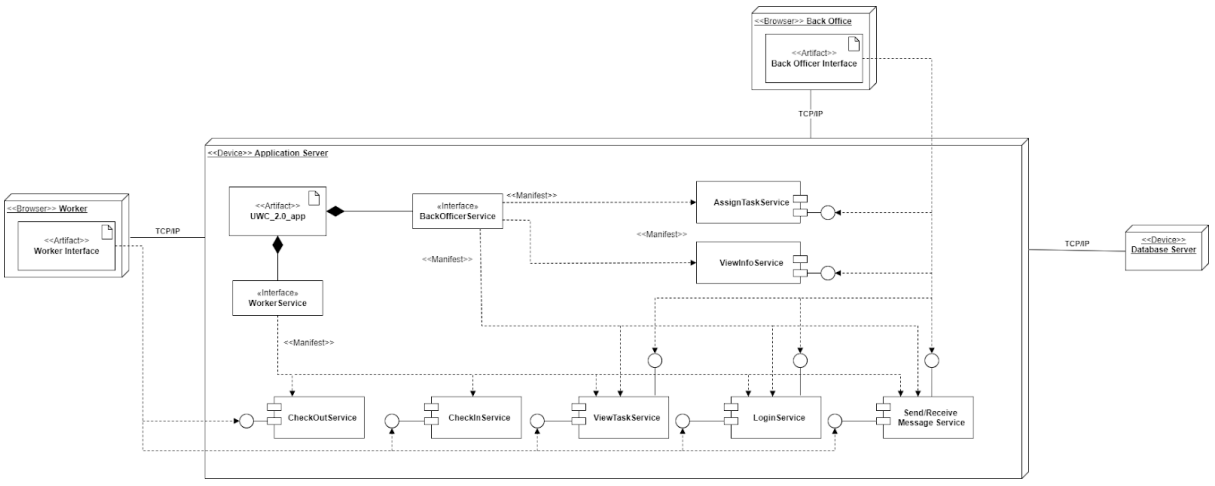
The Control component contains the user controller, task controller and message controller component.:

- User controller component provides the interface for log-in and sign-up functions, manipulating account information. Gather account data from Model and render account information view.
- Task controller component provides the interface for task creation, sending requests from back officers and rendering task review. This also helps assign tasks into the database.
- Message controller provides the interface for message sending requests including chat box opening, message sending, and message reporting.

The Model component contains a component Database with small components Employee, MCP, Vehicle, Task, Message:

- The Model component provides interfaces for the controller to perform operations with the database, providing interfaces for the view to update data when it receives a message from the model and tracks text messages.

3.2.2 Deployment diagram:



Back officer and the worker will connect to the app server by a web browser via protocol TCP/IP.

The app server will provide services and interface login, view worker's information, assign tasks, view tasks and send/receive messages for the back officer, login, view tasks, check in, check out and send/receive messages for the worker.

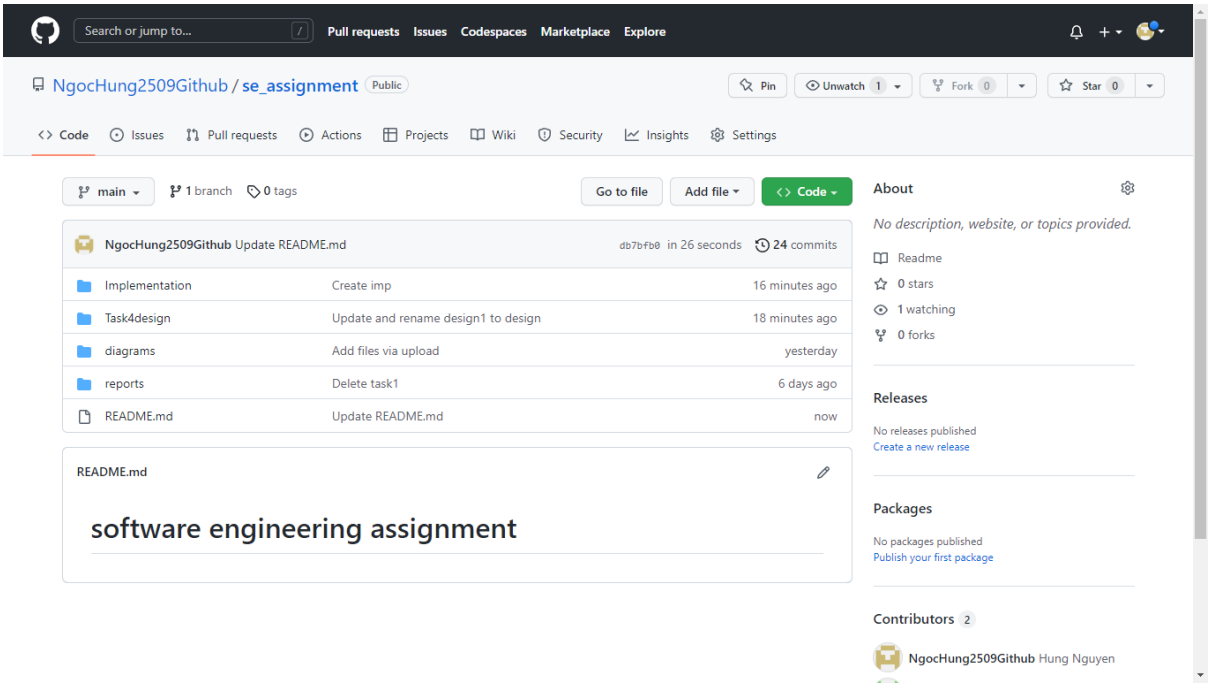
The app server will communicate with the database server via TCP/IP protocol.

Database server contains data about employees (Account), vehicle (Vehicle), MCP (MCPs) and tasks (Task).

4.1 Setting up.

We created an online repository on GitHub for version control.
The GitHub link: https://github.com/NgocHung2509Github/se_assignment.git

4.2 Adding documents, materials and folders for Requirements, System modeling, and Architectural design.



We have 4 folders:

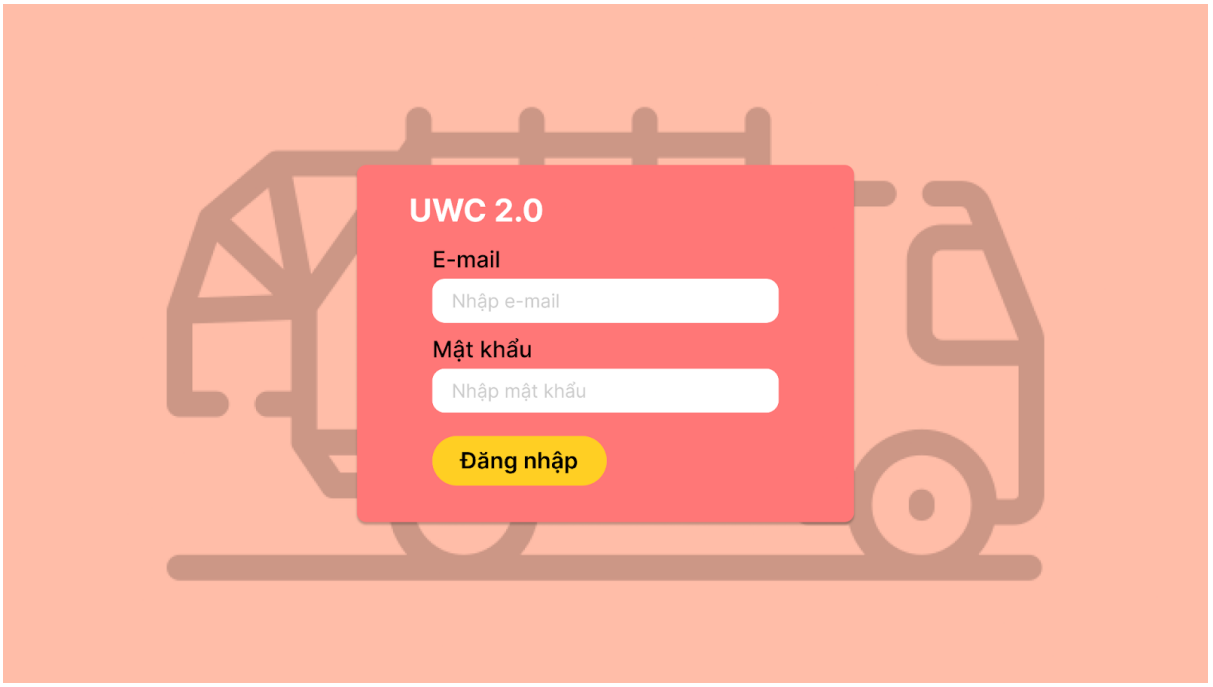
- **reports:** stores all reports from tasks 1 to 5
- **diagrams:** stores all diagrams that we drew for the requirements of tasks 1 to 5

- **Task4design:** stores a link to our Figma design for the UWC 2.0 user interface.
- **Implementation:** stores all of our implementation files.

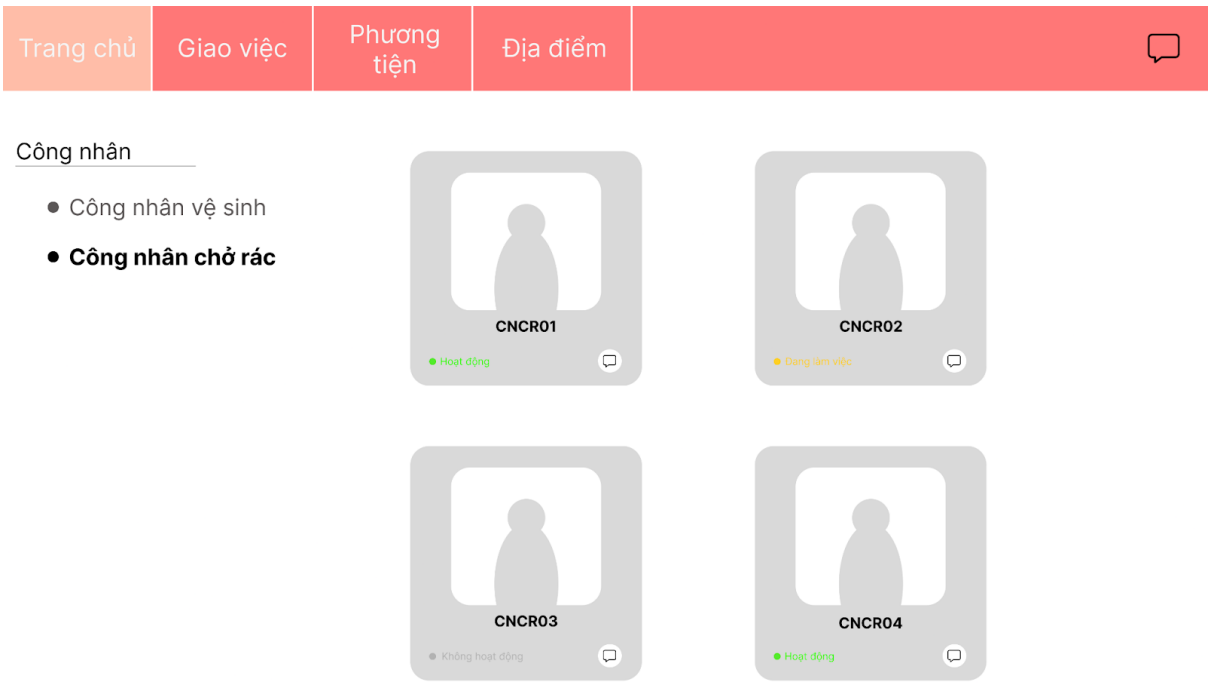
4.3 Implement MVP1.

Our UWC2.0 interface is implemented in Vietnamese due to the requirement of the project. The language also has an opportunity to be switched to English in the future.

This is a brief description of our design:



Firstly, the user needs an Email and Password given by the company to log in.

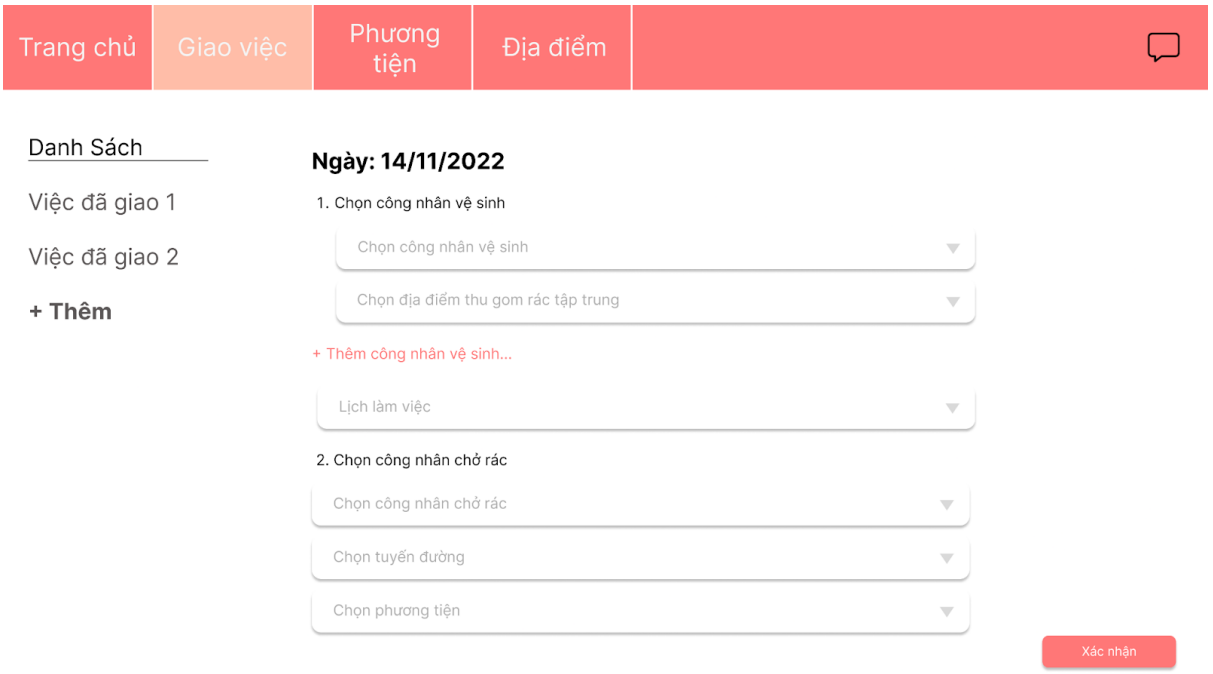


After logging in, the main site will display the name and the status of each collector and janitor.

When clicking on the profile, general information will be shown including the name, contact number, ID number, address and work schedule of the staff

Besides that, the site always has a little box chat icon that leads to the chat.

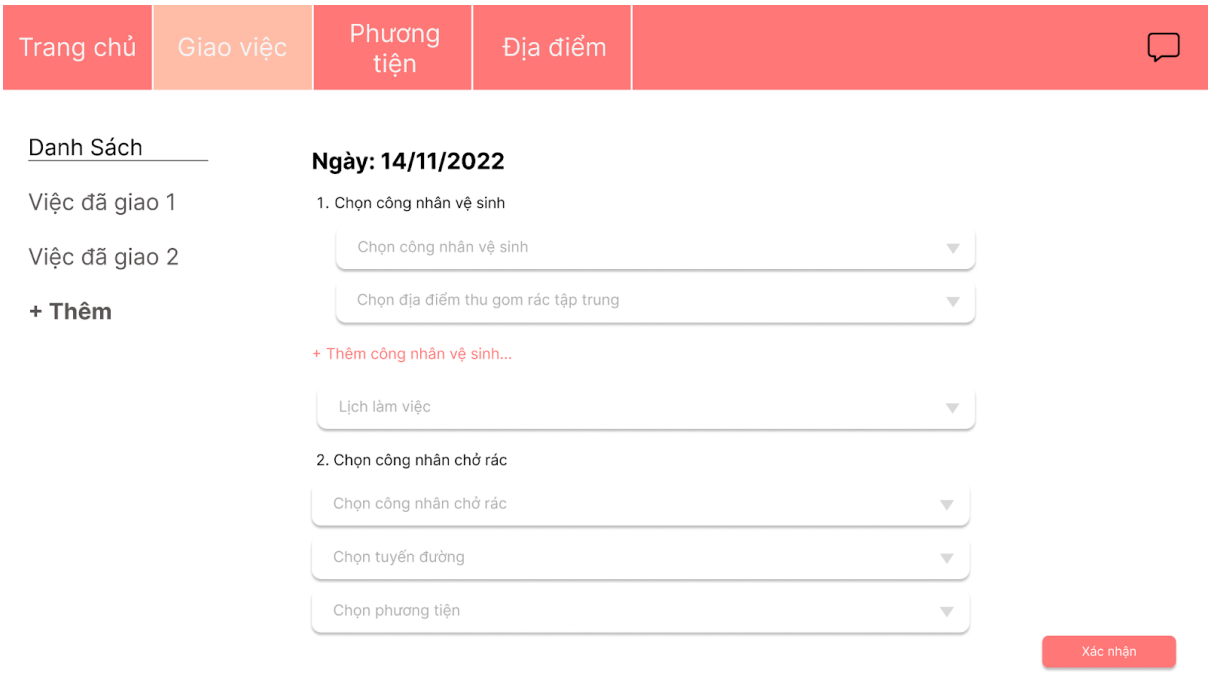
On the next page, there will be a list of tasks assigned to the collectors and janitors. Moreover, it also concludes the current date, checking attendance and content which includes: MCPs, vehicles and working progress.



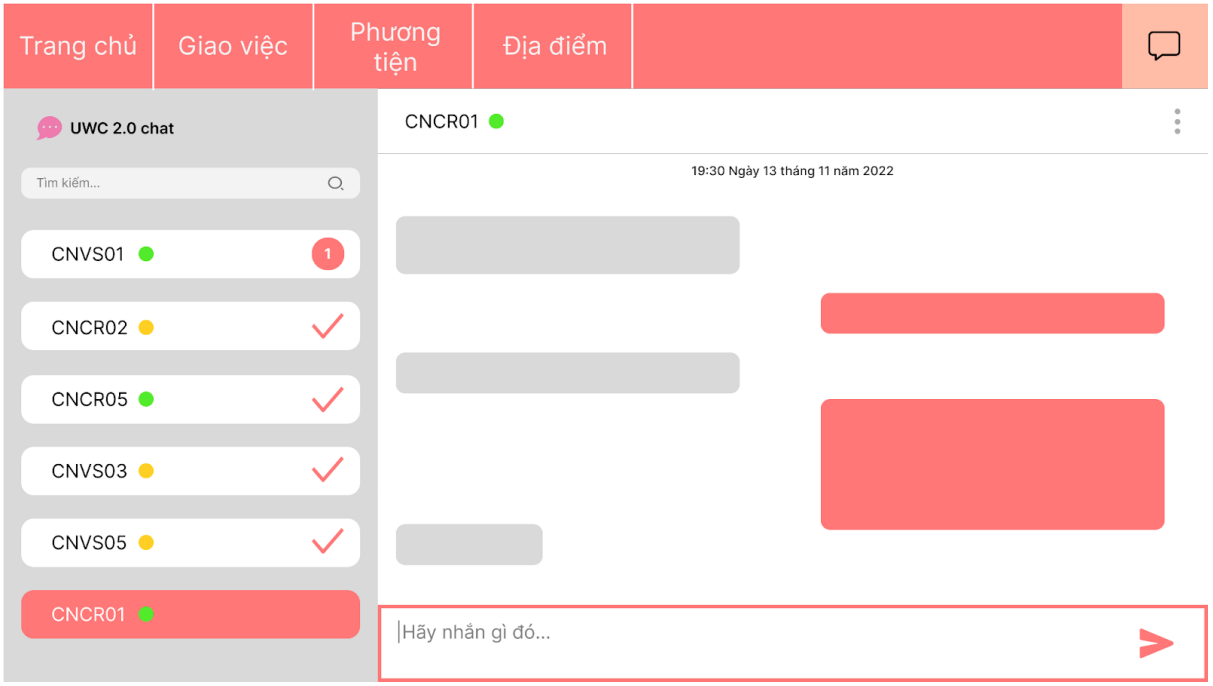
In order to assign a task, the back officer just needs to click on the button "+Thêm". Then, the app will lead the user to the site where they only need to fill in the information of the person they want to assign a task. The information

will be: the name of the collector/janitor, the vehicle they should use, and their working shift. After all, the back officer will click on the “Xác nhận” button to confirm and record the given task.

On the third page, the app will illustrate the technical details of vehicles.



The next page shows the capacity of each MCP. Based on the capacity status and the address, the back officers can effectively assign tasks for suitable collectors/janitors.



Finally, we have a chat interface that allows back officers to be able to see the conversations and the name + status of the worker according to them. Users have the capability to type messages and send them by clicking on the send button.

To have a better view of our UWC 2.0 UI, we also provide an illustration as the prototype of the system.

https://www.figma.com/proto/xXog4IdCLDEZfrdlFDyK6J/SEdes?node-id=125%3A256&scaling=contain&page-id=0%3A1&starting-point-node-id=125%3A256