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

Software Engineering

Capstone Project
Urban waste collection aid - UWC 2.0

Advisor:	Dr. Truong Tuan Anh
Student:	Huynh Thanh Tung - 2052781
	Trinh Hoai Thanh - 2053427
	Đang Cong Khanh - 2053105
	Ha Tan Khanh Nam - 2052603
	Tran Thanh Son - 2053405

HO CHI MINH CITY, September 2022

Contents

1 Task 1. Requirement elicitation

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?

- The context of this project:
  Require an application providing management and communication ability between employees inside Service provider Y in order to improve garbage collection ability of the organisation. The first application UWC 1.0 has been released, but it still emerges some problems with the management procedure and task distribution. Therefore, the upgraded version of this application is proposed called UWC 2.0 aiming to overcome the last version limitations and installed with extended features.

- Relevant stakeholders:
  Back officers, collectors, janitors, developer team.

- Current needs of stakeholders:

  1. Back officers
     - Be able to see information about employees work schedule, organisation's vehicles status, and current capacity of MCPs.
     - Assign tasks and communicate with employees.
     - Have rights to choose routes for collecting vehicles and assign any available vehicle to any route.
  2. Collectors and janitors
     - Load new tasks as soon as they are assigned to them.
     - Be able to checkin/checkout tasks.
     - Communicate with other employees or back officers.
     - Alerted if any MCP is full.
  3. Developer team
     - Develop UWC 2.0.
     - Fix any problem occurs.

- Current problems of stakeholders:

  1. Back officers:
     - Cannot appropriately assign the routes to collectors and janitors. (the route is not currently available,
     - Determine the workload between staff (unfair work distribution).
     - System can not synchronise data among devices.
  2. Collectors and janitors:
     - Can not communicate with their back officer.
     - Can not see the task and the calendar in an orderly way.

- Benefits of UWC 2.0 for each stakeholder:

  1. Back officers:
     - Have a clearer overview about employees and vehicles.
     - Chosen route is automatically optimised in case of fuel consumption.
  2. Collectors and janitors:
     - Have a clearer overview of tasks assigned to them.
     - Notice if any MCP is full.

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

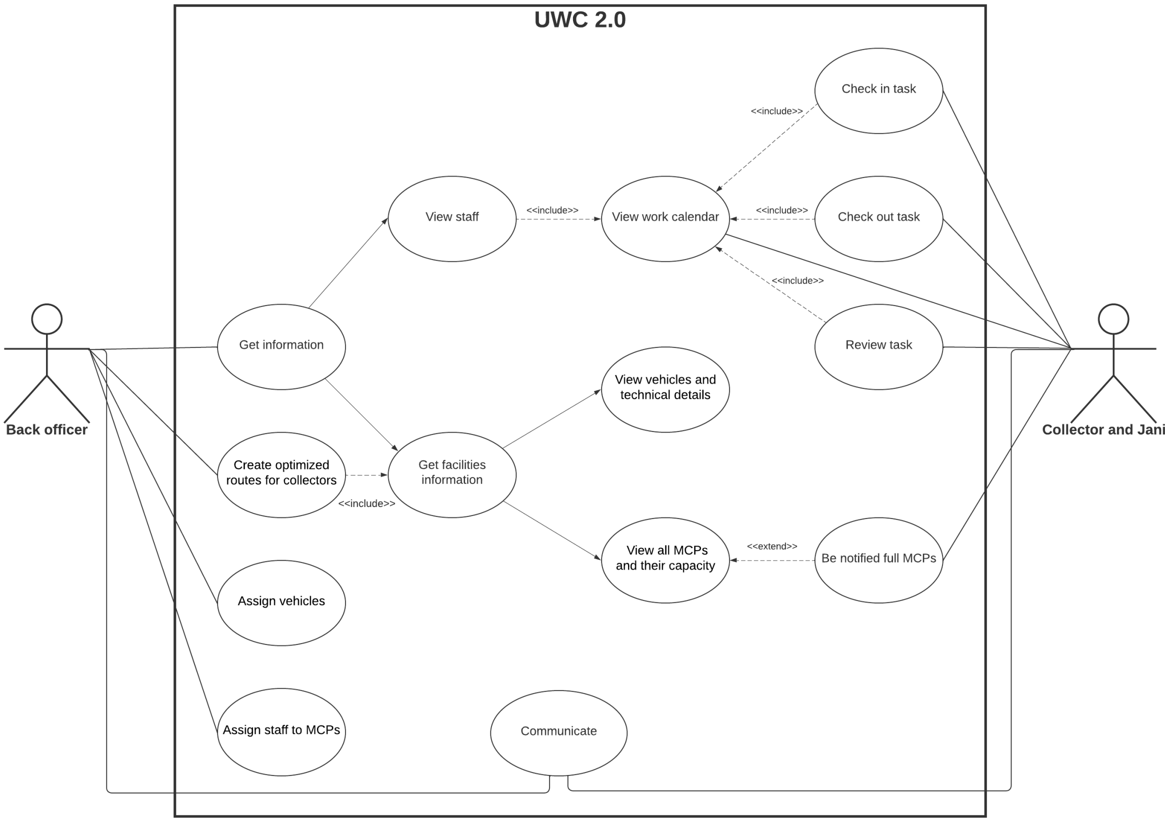
- Functional requirements:

  1. Back officers:
     - Have an overview of janitors and collectors, their work calendar.
     - Have an overview of vehicles and their technical details (weight, capacity, fuel consumption, etc).
     - Have an overview of all MCPs and information about their capacity.
     - Assign vehicles to janitors and collectors.
     - Assign janitors and collectors to MCPs (task).
     - Create a route for each collector. Assigned route is optimised in terms of fuel consumption and travel distance.
     - Be able to send messages to collectors and janitors.
  2. Collectors and janitors:
     - Have an overview of their work calendar.
     - Have a detailed view of their task on a daily and weekly basis.
     - Be able to communicate with collectors, other janitors and back officers.
     - Check in / check out tasks every day.
     - Be notified about the MCPs if they are fully loaded.

- Non functional requirements:

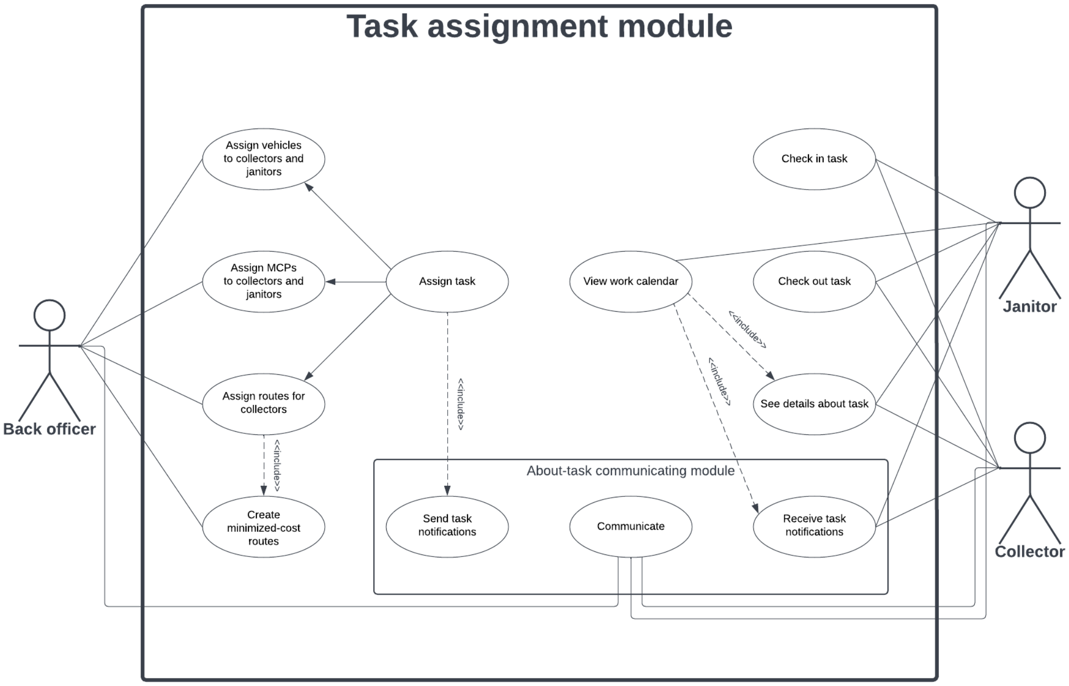
  1. Back officers:
     - Information should be updated from MCPs every 15 minutes with the availability of at least 95% of their operating time.
  2. Collectors and janitors:
     - All important information should be displayed in one view (without scrolling down).
     - The messages should be communicated in a real-time manner with delay less than 1 second.

- Use-case diagram for the system:



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

1. Use-case diagram for the task assignment module:



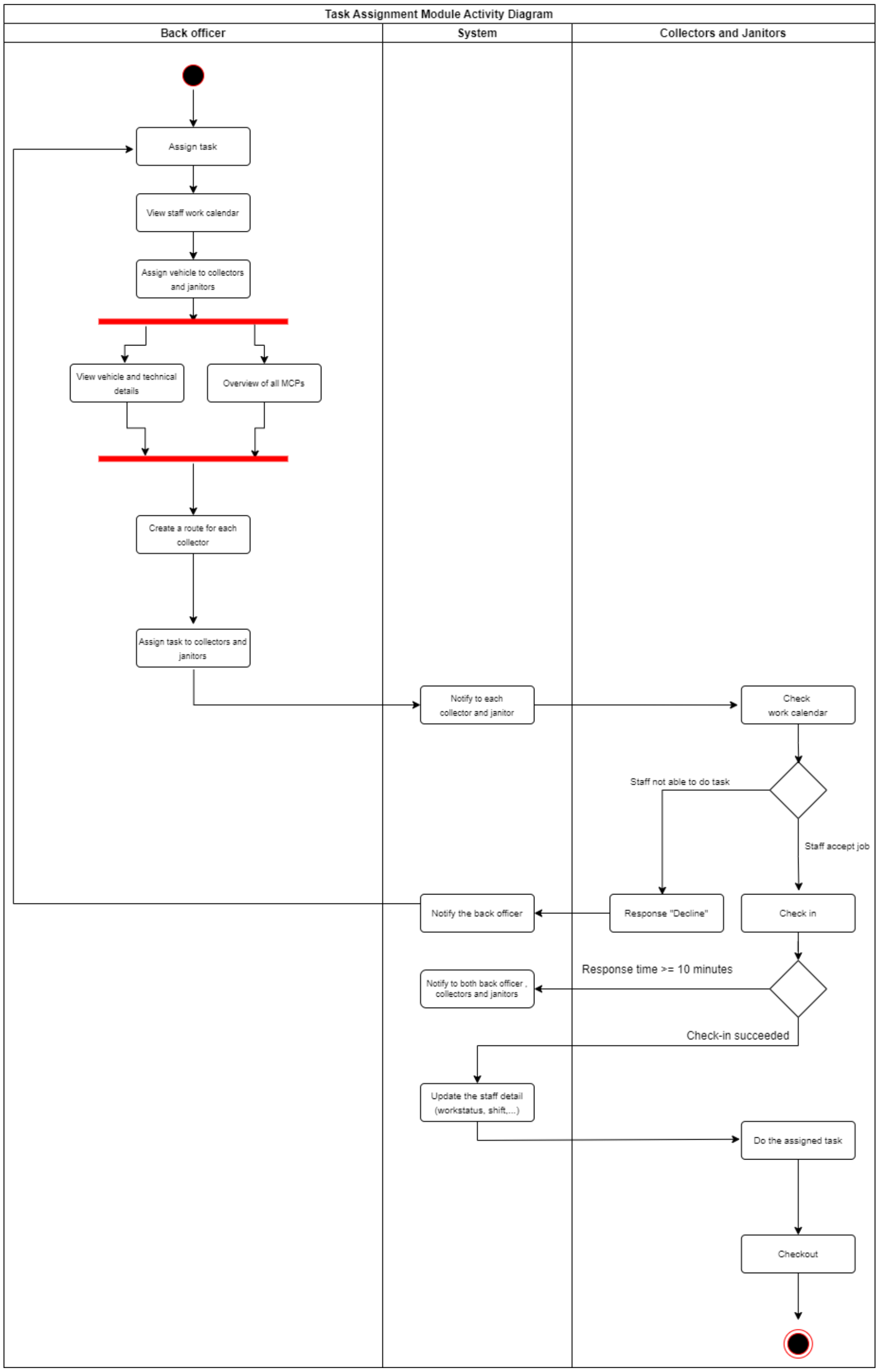
2. Table format

Name:	Task assignment
Primary actor:	Back officers
Secondary actor:	Collectors, janitors
Description:	Back officer manages the task and time check in and out of the staff. Communication among staffs.
Trigger:	When back officers assign tasks to collectors and janitors.
Preconditions:	There is at least one free collector or janitor
Postconditions:	The collectors and janitors are assigned with appropriate tasks
Normal flow:	1. The back officer assigns tasks for staff. 2. The back officer assigns vehicles to staff. 3. The system automatically creates the optimised route. 4. The back officer assigns the optimised route for the staff member. 5. The system sends a message (notification) to the janitors and collectors. 6. Staff receive the notification. 7. Staff checks the work calendar for the updated task. 8. Staff members view detailed tasks. 9. Staff can communicate with others via the message system. 10. Janitors and collectors check-in to do assigned jobs 11. After completing or finishing the work shift, the staff checks out. 12. End task assignment module.
Alternative flow:	Alternative 1, after step 9: Staff can give feedback about the assigned tasks; back officers can re-assign the task if staff is unable to handle the given task.
Exception:	1. Exception 1, at step 2: If there is no usable vehicle left: Notify the back officer. 2. Exception 2, after step 10: If a staff fail to check-in in 10 minutes: Notify the staff and the back officer. 3. Exception 3, at step 11: The staff closes the UWC 2.0, but there is still a task that is not checked out; the system will alert the user and prevent the staff from closing the application.

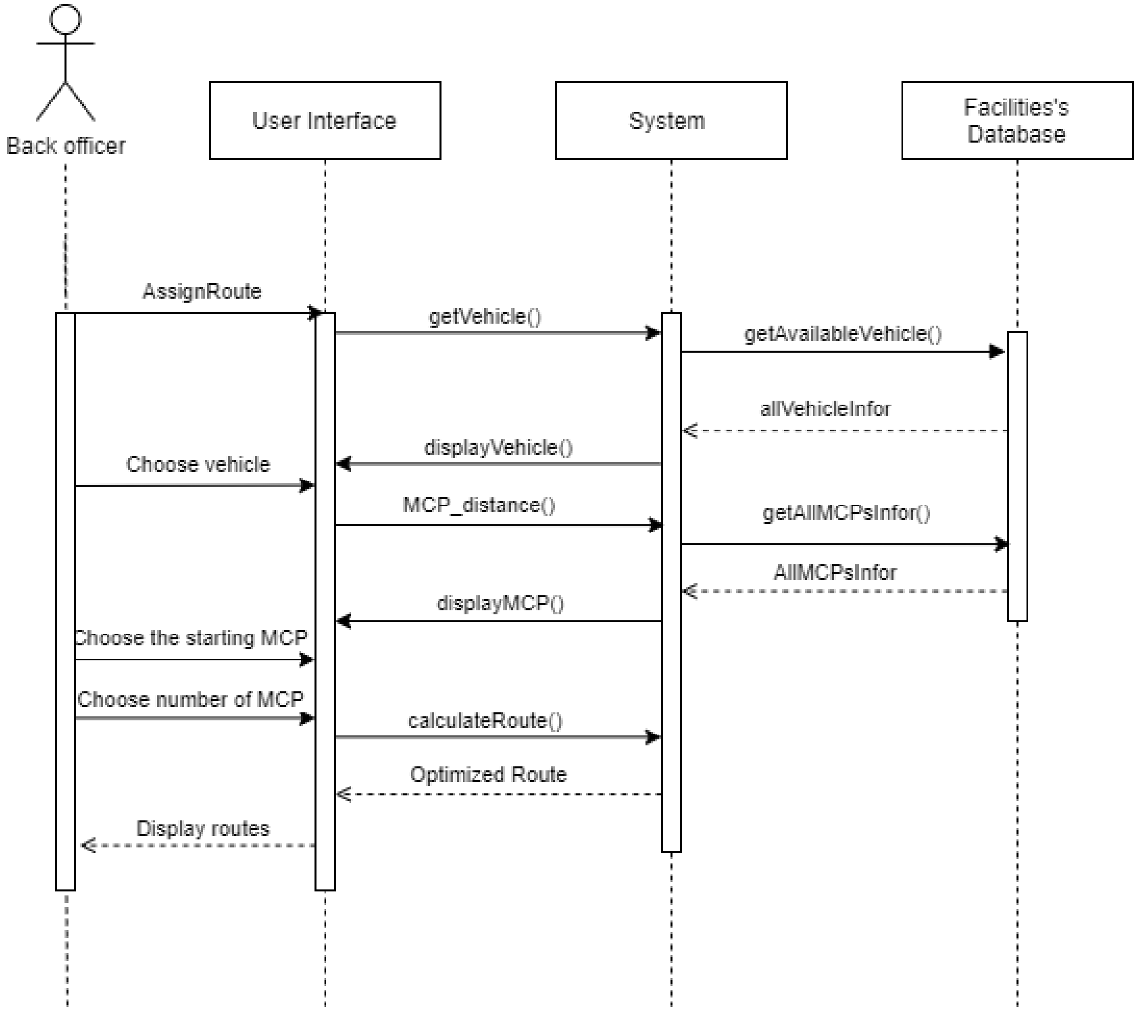
2 Task 2. System modelling

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

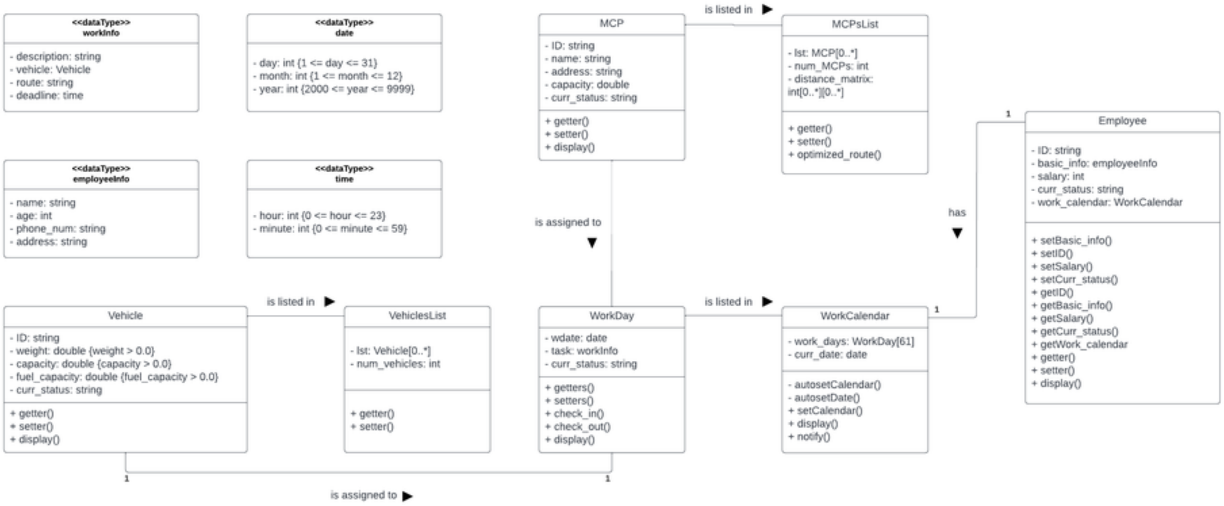
Here is the detailed diagram: IMAGE


Task Assignment Module Activity Diagram
Back officer
System
Collectors and Janitors
Assign task
View staff work calendar
Assign vehicle to collectors and janitors
View vehicle and technical details
Overview of all MCPs
Create a route for each collector
Assign task to collectors and janitors
Notify to each collector and janitor
Check work calendar
Staff not able to do task
Staff accept job
Notify the back officer
Response "Decline"
Check in
Response time >= 10 minutes
Notify to both back officer , collectors and janitors
Check-in succeeded
Update the staff detail (workstatus, shift,...)
Do the assigned task
Checkout

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



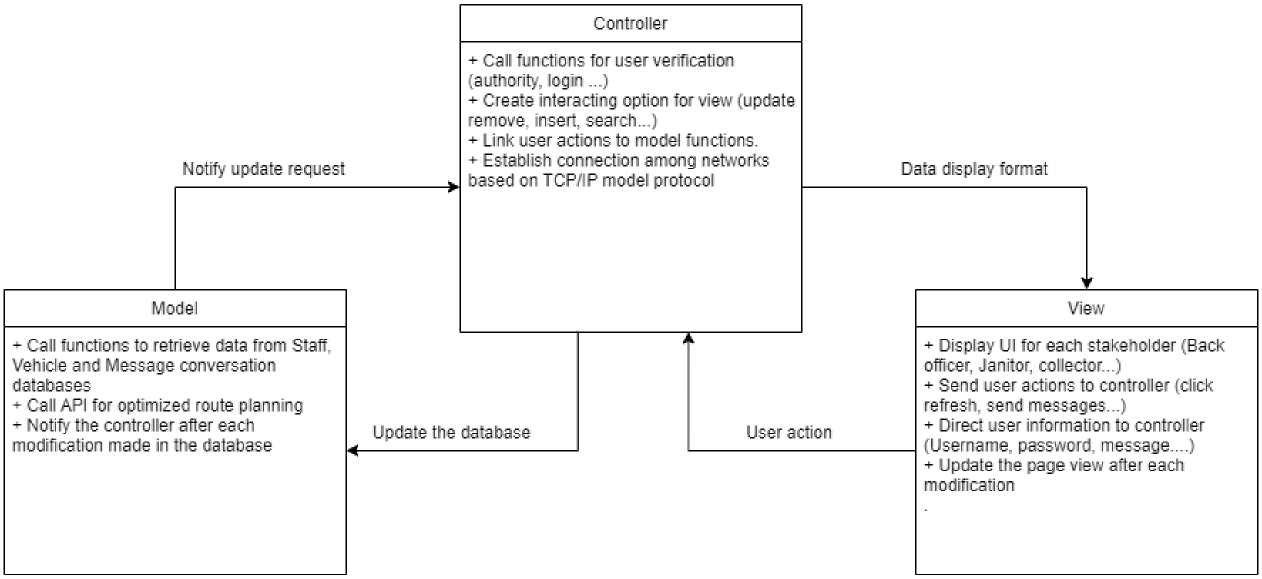
2.3 Draw a class diagram of Task Assignment module as comprehensive as possible.



3 Task 3: Architecture design

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

- Architectural approach:
  - Our system is a transaction processing system, which is mainly based on handling users' requests in getting an information or updating an information from a database (for example: janitors and collectors want to have their work calendar and information about their tasks, back officers want to update the calendar or MCPs states, as well as adding or removing information about staffs and facilities . . .)
  - The user will interact with an interface, with different action options depending on the role of that user (Manage, Update or view only. . .)
  - The application will be web-based which allows communication among users through Internet connection.
  - Because of the mentioned reasons, we decided to use the MVC (Model-view-controller) pattern for the architectural design.



Controller

+ Call functions for user verification (authority, login ...)
+ Create interacting option for view (update remove, insert, search...)
+ Link user actions to model functions.
+ Establish connection among networks based on TCP/IP model protocol

Model

+ Call functions to retrieve data from Staff, Vehicle and Message conversation databases
+ Call API for optimized route planning
+ Notify the controller after each modification made in the database

View

+ Display UI for each stakeholder (Back officer, Janitor, collector...)
+ Send user actions to controller (click refresh, send messages...)
+ Direct user information to controller (Username, password, message....)
+ Update the page view after each modification
.

- MVC description
    1. Model
        - Have methods/functions to access and retrieve information about staff and vehicle
        - Be able to call an API specialized in creating optimized routes from given vehicles and MCPs locations.
        - Whenever a change is made in the model database, it will send a request to the controller to refresh the page.
    2. Controller
        - Show responding UI based on user's action
        - Call functions in the model for CRUD task (update calendar, get staff information, delete a task,...)
        - Handle user verification and authority:
            - Login verification
            - Determine which category the user can modify/see (Janitor and collector cannot modify the work calendar)
        - Response user request/action
    3. View
        - Provide a responding interface for each stakeholder
        - Receive user input data (username, password, . . .) and actions (click button, refresh page,...)
        - Parse data to controller to call function
- Module details
    - We categorize our modules based on the use case, so there are 5 modules in total for our desired system.

Name	Communication module
Functionality	Provide an environment for communicating between staffs and back officers
Relation	Database: Access previous messages
Input	Communication request (send message)
Output	The request is successfully executed (message is delivered to the receiver)

Table 1: Communication module

Name	Information management module
Functionality	Get, assign, update or remove information about staffs and facilities
Relation	Database: Access the database for retrieving and updating information
Input	A user's command/action
Output	Requested data displayed and/or some updates on the database

Table 2: Information management module

Name	Work calendar management module
Functionality	Get, assign, update or remove information in staffs' work calendar Check in and check out task for staffs to keep track of progress
Relation	Database: Access the database for retrieving and updating information Task assignment module: Assigning and updating work calendar is a result of Task assignment module
Input	A user's command or a request from Task assignment module
Output	Requested data displayed and/or some updates on the database

Table 3: Work calendar management module

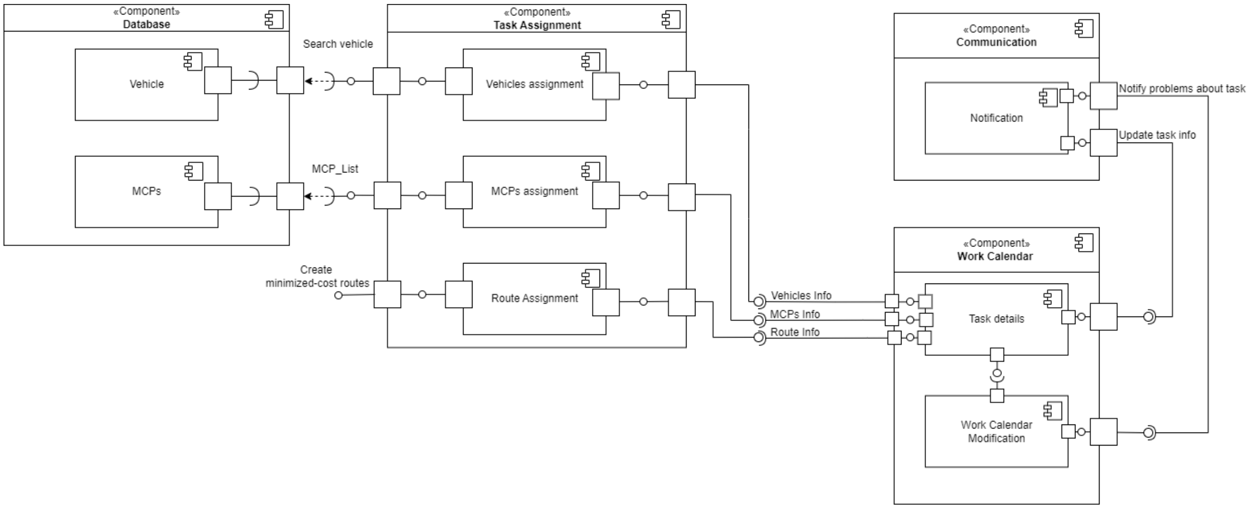
Name	Task assignment module
Functionality	Receive assignment request from back officers and generate appropriate task for staffs based on current state of facilities Notify the staffs when some updates happen
Relation	Information management module: task assignment needs information about the staffs and facilities
Input	Request of assigning a task to a staff
Output	A task is generated and scheduled in the work calendar

Table 4: Task assignment module

Name	Interface utility
Functionality	Language switching (Vietnamese - English)
Relation	Database: Access language mapping table
Input	Request of switching language
Output	Language changed

Table 5: Interface utility

3.2 Draw an implementation diagram for Task Assignment module


«Component»
Database
Vehicle
MCPs
Search vehicle
MCP_List
«Component»
Task Assignment
Vehicles assignment
MCPs assignment
Route Assignment
Create
minimized-cost routes
«Component»
Communication
Notification
Notify problems about task
Update task info
«Component»
Work Calendar
Task details
Work Calendar
Modification
Vehicles Info
MCPs Info
Route Info