

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



## SOFTWARE ENGINEERING

---

ASSIGNMENT - GROUP NAME: 3 diem qua mon

## URBAN WASTE COLLECTION AID - UWC 2.0

---

Under the guidance of: Prof. Quan Thanh Tho  
Accomplished by: Nguyen Van Quoc Chuong - 1950004  
Dau Gia Kien - 1952799  
Nguyen Huy Hoang - 1852382  
Le Tuan Minh - 2052821  
Dam Ngoc Hieu - 2052469

HO CHI MINH CITY, 9/2022



## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Requirement elicitation</b>	<b>2</b>
2.1	Task 1.1 . . . . .	2
2.2	Task 1.2 . . . . .	3
2.3	Task 1.3 . . . . .	4
<b>3</b>	<b>System Modeling</b>	<b>9</b>
3.1	Task 2.1 . . . . .	9
3.2	Task 2.2 . . . . .	9
3.3	Task 2.3 . . . . .	10
<b>4</b>	<b>Architecture design</b>	<b>12</b>
4.1	Task 3.1 . . . . .	12
4.2	Task 3.2 . . . . .	15
<b>5</b>	<b>Implementation sprint 1</b>	<b>17</b>



## 1 Introduction

Urban waste management is one of several significant problems faced by many countries in the world and thus considered one of the important points to be improved in Sustainable Development Goal (SDG) 11: sustainable cities and communities and SDG 6: clean water and sanitation. Particular attention is given to developing countries that continue to prioritize development and economic growth. In urban context, solid waste management is costly and ineffective. Improvement of waste collection and management is emphasized by governments and organizations for positive impacts on cities, societies and environments

## 2 Requirement elicitation

### 2.1 Task 1.1

*Solution:*

1. Context: Many countries are facing with urban waste management which is one of several significant problems. In urban area, solid waste management is costly and ineffective. That is the reason why UWC- 2.0 is created.
2. Stakeholders:
  - Organization X
  - Service provider Y
  - Back officers
  - Janitors
  - Collectors
3. Current needs:

**Organization X** need to develop an information management system called UWC 2.0.

**Service provider Y** need UWC 2.0 to improve efficiency of garbage collection.

**Back officers** need to have an overview of janitors and collectors, their work calendar. They also have an overview of vehicles and their technical details (weight, capacity, fuel consumptions, etc). Moreover, they shoule have an overview of all MCPs and information about their capacity, assign vehicles to janitors and collectors and also



assign them to MCPs. In addition, Back officers have to create a route for each collector.

**Janitors and collectors** have an overview of their work calendar, a detail view of their task on a daily and weekly basis. They also are able to communicate with collectors, other janitors and back officers, are checked in or checked out task every day, and are notified about the MCPs if they are fully loaded.

#### 4. Current problems:

**Organization X:** no problem.

**Service provider Y:** shortage in efficiency of garbage collection.

**Back officers:** shortage information in janitors and collectors of work calendar, vehicles and their technical details, capacity of all MCPs.

**Janitors and collectors:** shortage information in their task on a daily and weekly basis. They also have problems in communicate with other collectors, janitors and back officers.

#### 5. Benefits of UWC 2.0: This system can help all stakeholders with their problems which is mentioned above.

## 2.2 Task 1.2

#### 1. Functional Requirements:

- Back Officers can see a brief work calendar of janitors and collectors.
- Back Officers can see the details of vehicles.
- Back Officers can see the status of the MCPs about their capacity, The information should be updated every 15 minutes.
- Back Officers can assign vehicles to janitors and collectors.
- Back Officers can assign janitors and collectors to MCPs.
- Back Officers can make a route for each collector. The route is supported by the software to optimize the fuel and distance.
- Back Officers, Collectors and Janitors can message each other.
- Collectors and Janitors can view their work calendar.



- Collectors and Janitors can have a detail view of their task on a daily and weekly basic.
- Collectors and Janitors can check in and check out task everyday.
- Collectors and janitors should be notified when the MCPs in their route is full loaded.

## 2. Non-functional Requirements:

- The UWC 2.0 should use the old database from the primal version.
- The Task Management should be inter-operable with the UWC 1.0 as much as possible.
- The UWC 2.0 should be an optimized app that is usable on all types of mobile devices, tablets or computers, and performs well on multiple platforms.
- All important information that are viewed by Janitors and Collectors should be in one page for convenience.
- Messages sent through UWC 2.0 should have a delay time less than 1 second.
- The system should be able to handle real-time data from at least 1000 MCPs in one moment and 10.000 MCPs in five years
- The interface should be in Vietnamese.

The use case diagram for the system is [as follows](#)

### 2.3 Task 1.3

The use case diagram for the Task Assignment Module is [as below](#)

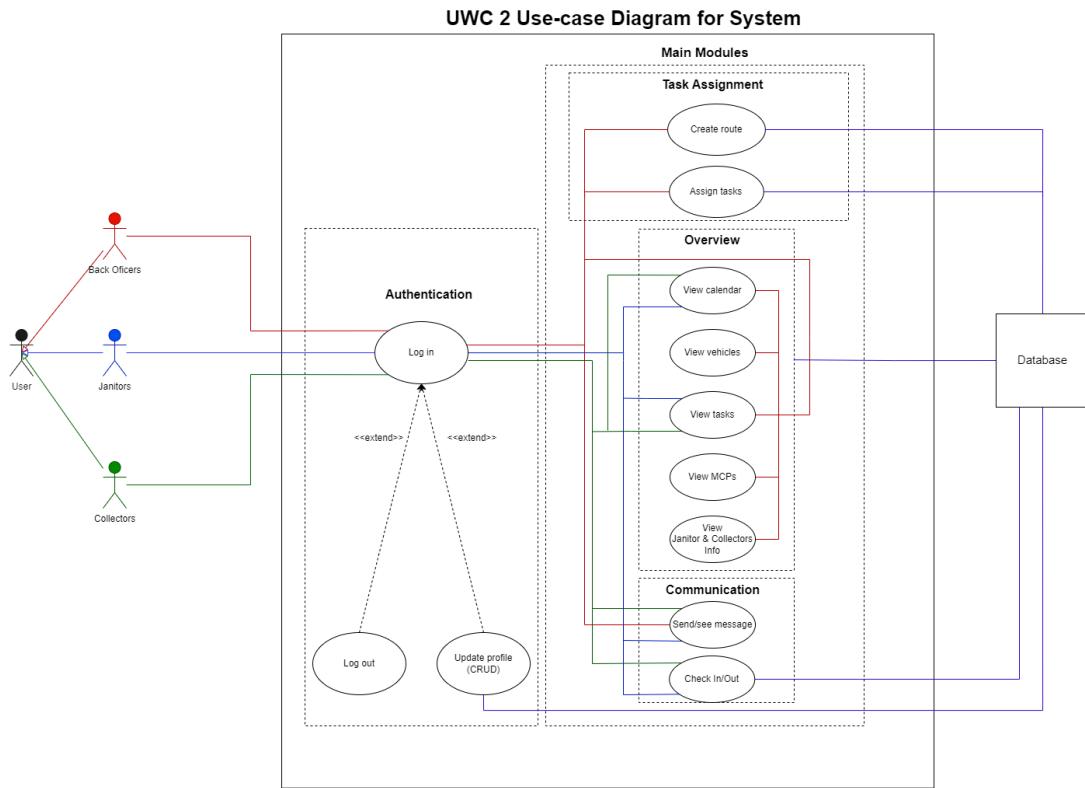


Figure 1: Use case diagram for the whole system

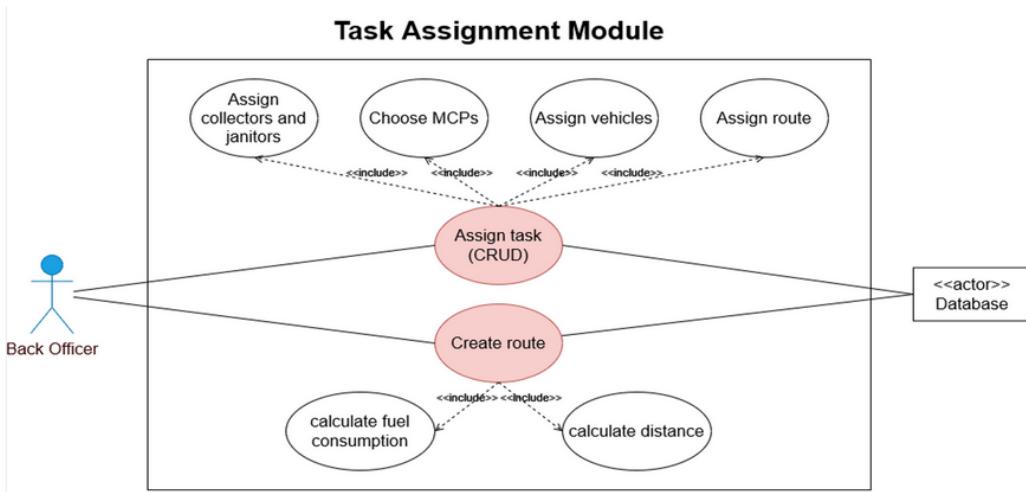


Figure 2: Use case diagram for Task Assignment Module



Author	Dau Gia Kien
Use case ID	1
Use case name	Create route
Description	Create a route
Actor(s)	Back officers
Pre-conditions	Have had calculated fuel consumption and distances before hand.
Post-conditions	A route will be created successfully
Normal flow	<ol style="list-style-type: none"><li>0. Use case begins when back officers click on “Create route”.</li><li>1. The system displays a form for the user to input new route information<ul style="list-style-type: none"><li>• Distance</li><li>• Fuel consumption</li></ul></li><li>2. Users click on the “create” button.</li></ol>
Exception flow	None

Author	Nguyen Van Quoc Chuong
Use case ID	2
Use case name	Assign task
Description	Assign a task for collectors and janitors
Actor(s)	Back officers
Pre-conditions	Back officers must be logged in to the application.
Post-conditions	Back officers must choose and select available collectors, janitors, MCPs, vehicles and a route.
Normal flow	<ol style="list-style-type: none"><li>0. Use case begins when back officers click on “Assign task”.</li><li>1. The system will ask the user to fill in the necessary information:<ul style="list-style-type: none"><li>• Collectors, Janitors</li><li>• MCPs</li><li>• Vehicles</li><li>• Route</li></ul></li><li>2. Users click on the “create” button.</li></ol>
Exception flow	None

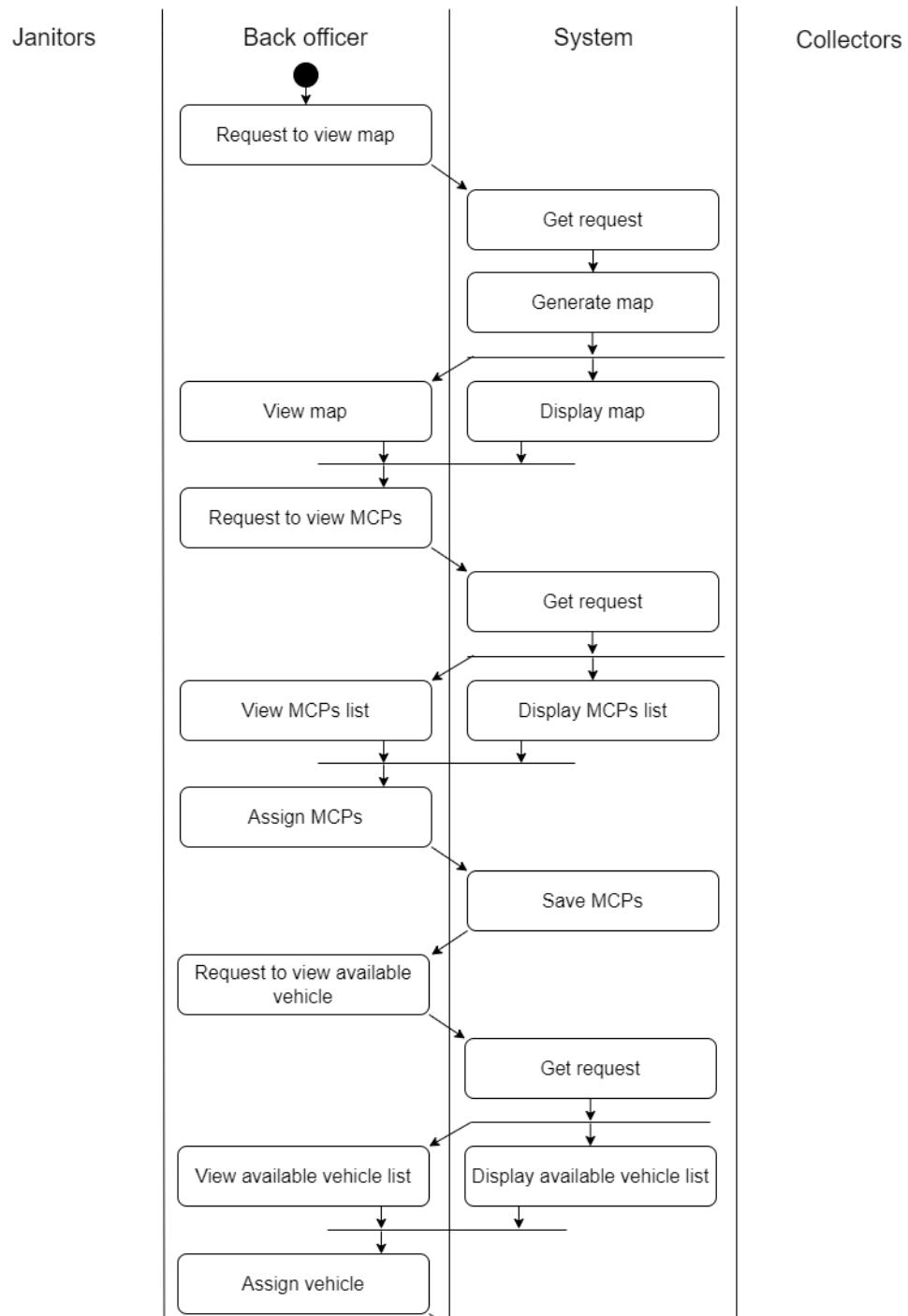


Figure 3: Activity diagram

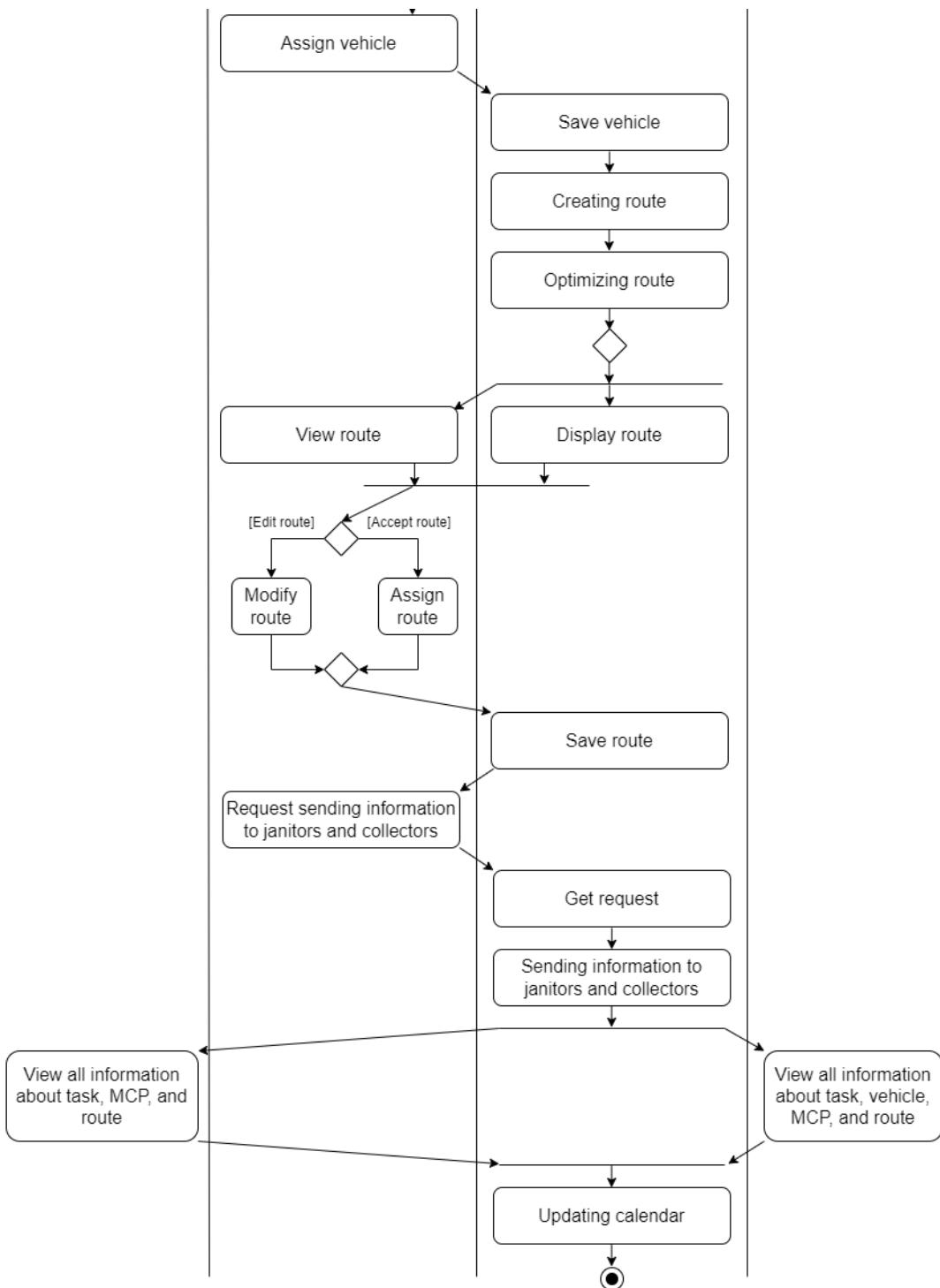


Figure 4: Activity diagram



### 3 System Modeling

#### 3.1 Task 2.1

**Description** of [Activity diagram](#) for the business process between systems and the stakeholders in Task Assignment module. The module has the following activities:

- View map
- Assign MCP
- Assign vehicle
- Create/modify route
- View tasks
- Communicate

#### 3.2 Task 2.2

This section describe **Conceptual solution** for route planning task and its [sequence diagram](#).

- One MCP should have a certain amount of trash to be ready for pick up. Therefore, the lists of MCPs that are sent to the Back Officers should only contain MCPs that are ready. Route created by the app required to be optimized in term of fuel consumption and travel distance
- Vehicle must have sufficient fuel, not broken, and can operate normally
- The MCP full pattern (how long it usually take for an MCP to be filled) should be understood to perform route planning accordingly

**Description:**

**In this feature:**

**First off** the back officer will receive a map of the working area generated by the app.

**Secondly** the back officer will send a request to a system in order for the system to choose which MCPs are ready to be picked up and make a list, then send it to the back officer.

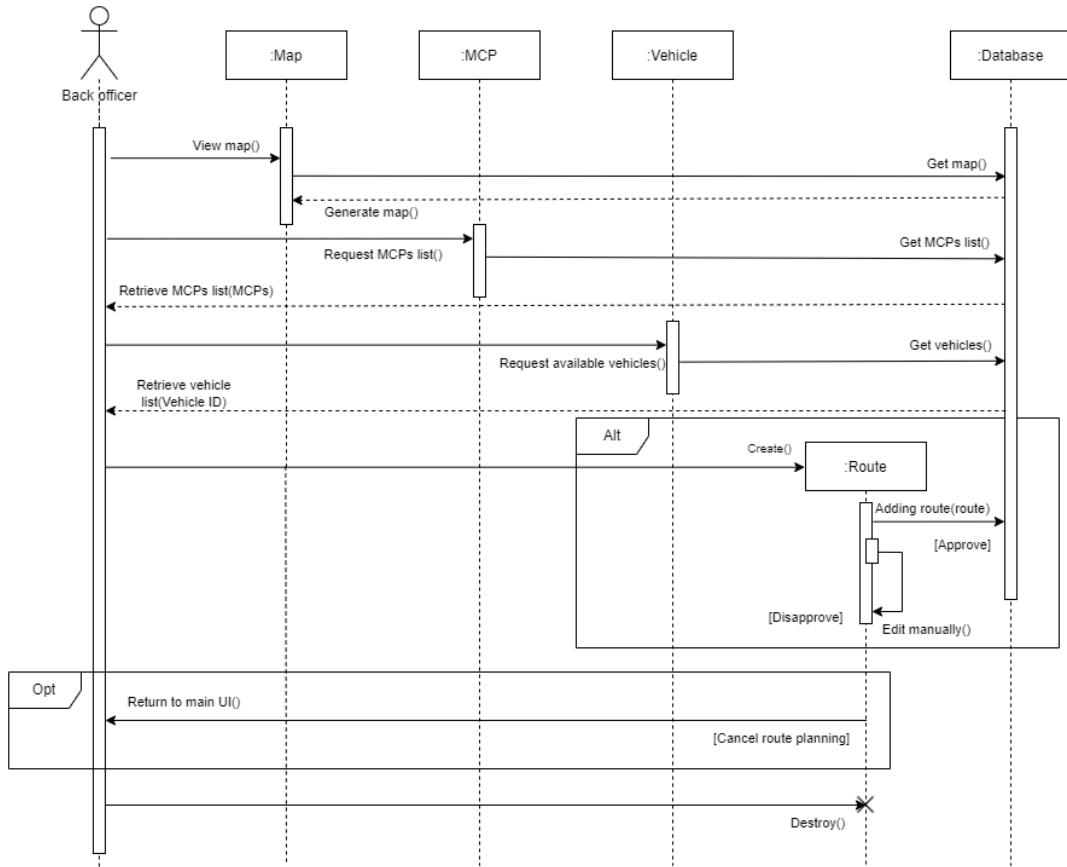


Figure 5: Sequence diagram for route planning task

Next off the Back officer send another request to view the available vehicles, the system will send the list including available vehicles to the Back officer.

Lastly The system will automatically make a route, and wait for the back officer to approve, If the back officer doesn't like the route he can destroy it and the system will make a new one instead.

### 3.3 Task 2.3

Description of [Class Diagram of Task Assignment Module](#):

- Back officer can use (*Assign route*) to a dedicated task and (*Assign task*) to the employees

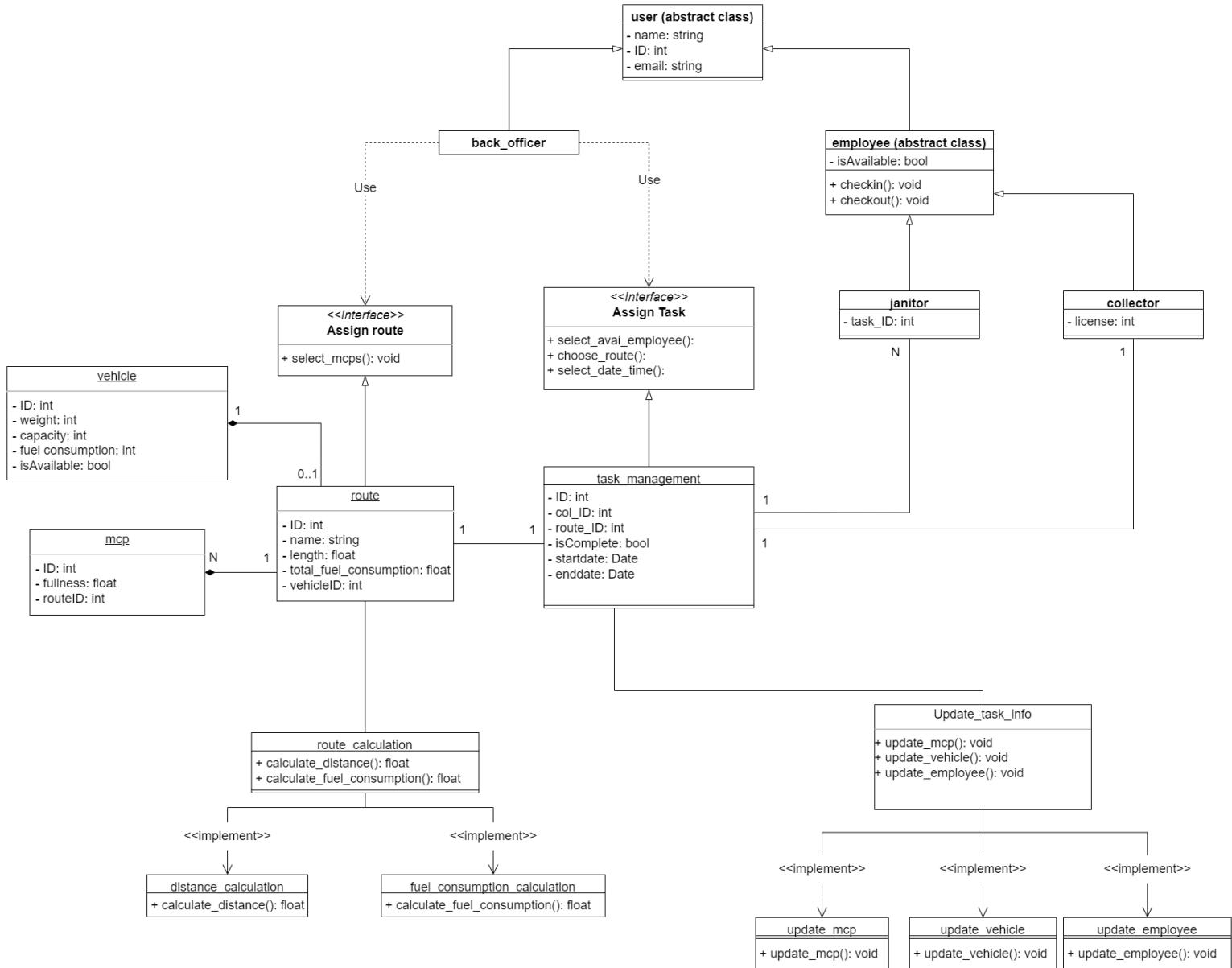


Figure 6: Detailed class diagram for Task Assignment Module



- (*Task management*) contain relevant information to keep track with data in system. Tasks are assigned correspondingly to janitors and collectors. startdate and enddate attribute are decided when janitors or collectors begin and finish their task.
- Each task is associated with a route, which is driven through by collectors to the MCPs.
- Route is created with (*route calculation*), and the dominant influence are MCPs and vehicles. A route is gone through by only 1 vehicle piloted by 1 collector.
- Vehicle, employees has a boolean attributes to keep track availability. They may not associated with any tasks or route which is indicated on the diagram by 0..1 relationship
- (*Update task info*) update (*Task management*) whenever back officer make changes of employee, mcp and vehicles to the task
- Many (*janitor*) can belong to 1 task, but only 1 (*collector*) belong to 1 task

## 4 Architecture design

### 4.1 Task 3.1

We choose [MVC design pattern](#) for our system, [here](#) is our explanation about our system modules.

- View
  - For Back officer:
    - \* Render icon, button, box,... for Log in or sign up Page
    - \* After logging in, Render components for a Dashboard with Selections boxes to go into different Pages for overviewing : Vehicles, Information of Janitors and Collectors, Work Calendar, MCPs information. And Highlighted box for the next step: Assignment.
    - \* Render components for vehicles viewing page.
    - \* Render components for Janitors and Collectors viewing page.
    - \* Render components for Work Calendar viewing page.
    - \* Render components for MCPs information viewing page.

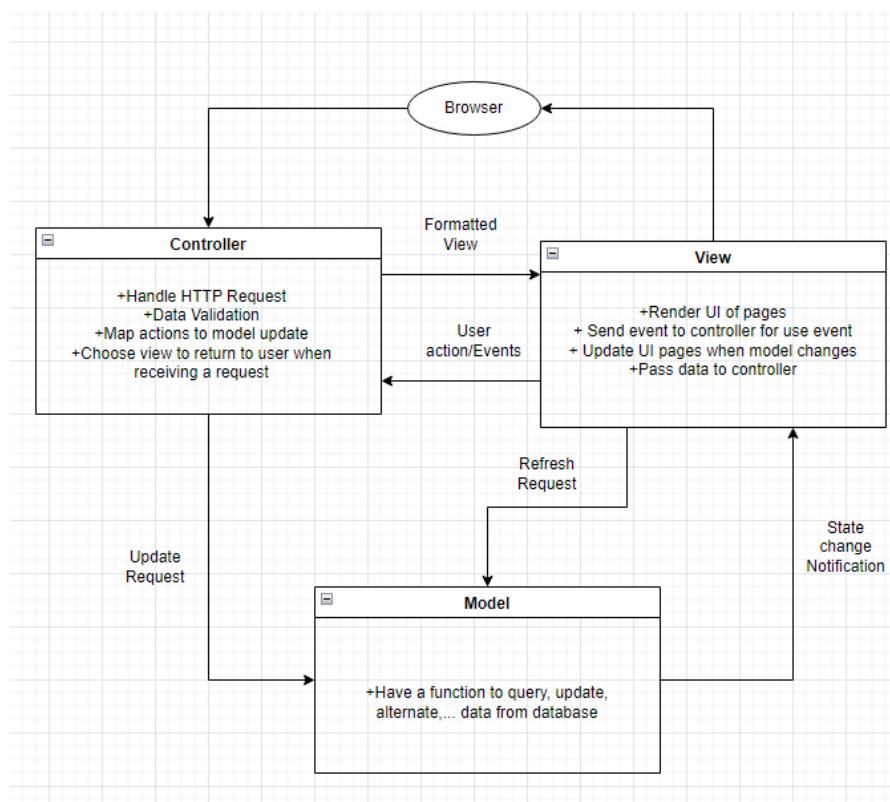


Figure 7: Architecture design pattern for the system



- \* Render form, button, text, ... for assignment page, with a special part for routing when assign tasks.
- \* Render button, icon, ... for a message box.
  - For Janitors, Collectors:
  - \* Render icon, button, box,... for Log in or sign up Page
  - \* Render button,icon, box, text,... a Dashboard page to select to overview one of these : Work Calendar, Tasks and a Check in check out choice
  - \* Render components for Work Calendar page.
  - \* Render components for Tasks assigned page.
  - \* Render button, box,.. for Check in/Check out page.
  - \* Render button, icon,... for a message box.
- Send events done by users back to the controller.
- Send user data inputs back to the controller in order to process logic of the application.
- When asked to refresh state by model, take the data from model in order to refresh view and update the new view.
- Controller
  - Return a View of a page requested by the user.
  - \* Update the work calendar, Tasks,Vehicles and routes of Janitors and Collectors, The information of MCPs.
  - \* Delete Janitors and Collectors information, Vehicles information, Tasks, Route and Work Calendar, The information of MCPs.
  - \* Alternate Janitors and Collectors information, Vehicles information, Tasks, Route and Work Calendar, the information of MCPs.
  - Using methods from model to log in, sign up, and manage user information.
- Model
  - Give us methods to interact with the database to log in, sign up, manage information of users.
  - Give us methods to interact with the database such as update, delete, alternate.
  - When the state of model changes it has to notify view so view can refresh and update.



In this project, we decided to implement 4 modules: Authentication module, Task assignment module, Overview module and Communication module. Here below is a table showing their input/output and functions respectively:

Module	Input	Output	Functions
Authentication	Login information input from user's keyboard (login ID and password)	User Interface for each type of users (backofficers, janitors, collectors)	To authenticate staffs of the UWC2 project To update user profile information
Task assignment	Commands from backofficers	Task notifications to janitors and collectors in Communication module Update task status to database	View current status on a map Create route Delete route Assign tasks Cancel tasks
Overview	Feedback information from database	Graphical display of desired information	For each type of users will be able to view a certain status: calendar, vehicles, tasks, MCPs, user's profile
Communication	Command from senders Notification feedback from database	Message notification to receiver	A portal to provide communication between staffs

Figure 8: Modules and input/output/function for each module

## 4.2 Task 3.2

### Description of Component diagram for Task Assignment Module:

- Model:
  - Model components notify View component of state changes
  - Model component encapsulates application state
- View:
  - View component Employee view, Vehicle view, Route view, Task assignment view belong to the Back officer. They help BA with task assigning.
  - View components requests model update from Model component and perform get methods to Controller component with users event.
  - View render Model to display data
- Controller:
  - Controller component maps users action and update the Model components
  - Controller tasks are as describe in section 3.1

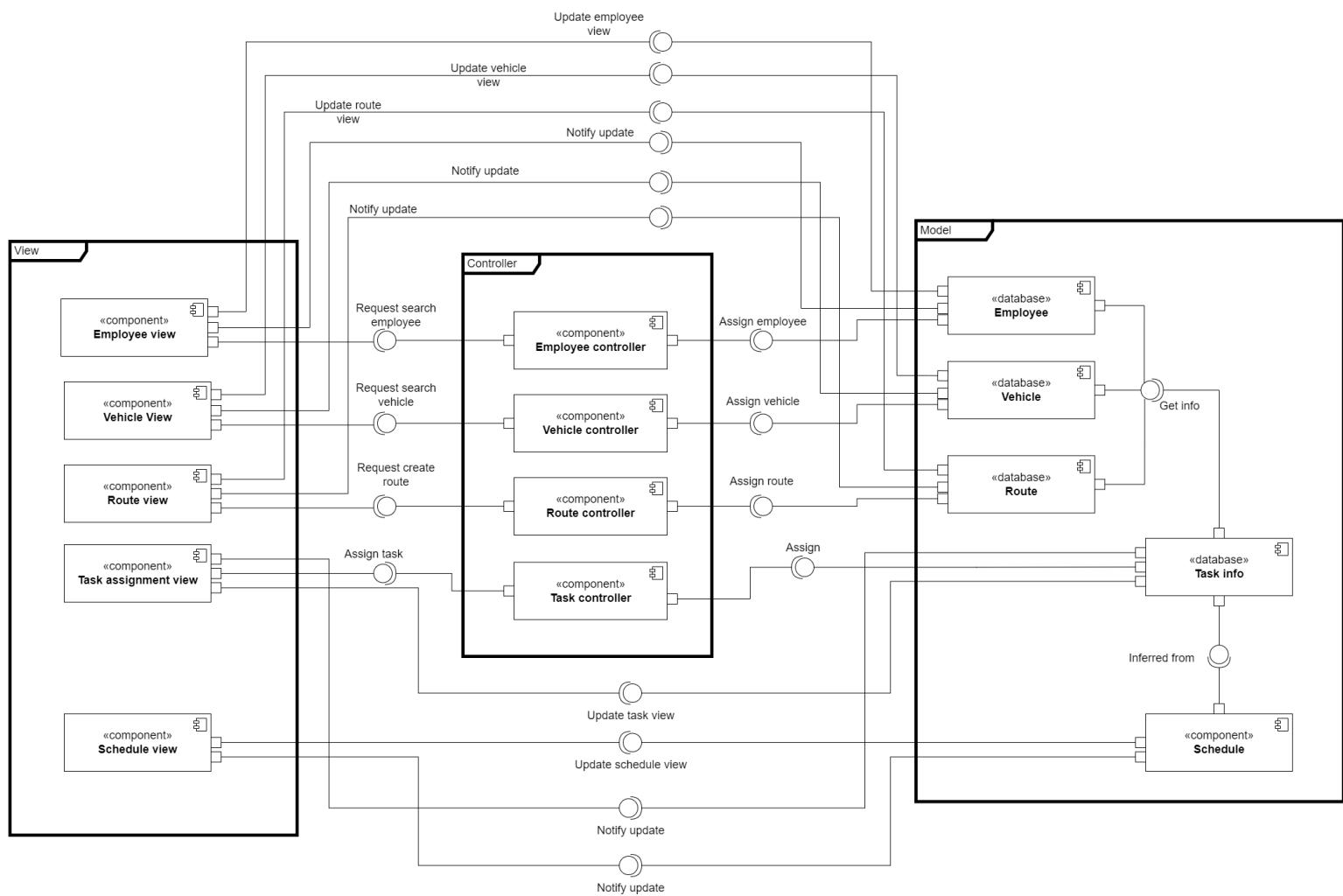


Figure 9: Component diagram for Task Assignment Module

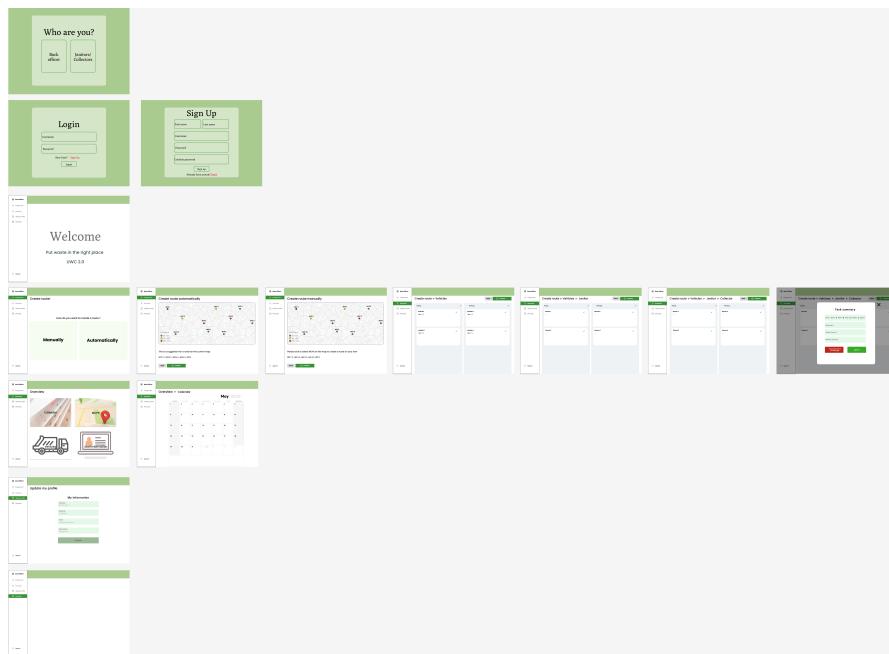


Figure 10: Task 4: Mock up for website application interface, please refer to repo for full picture

## 5 Implementation sprint 1

Here is our [Github repository link](#)

## References