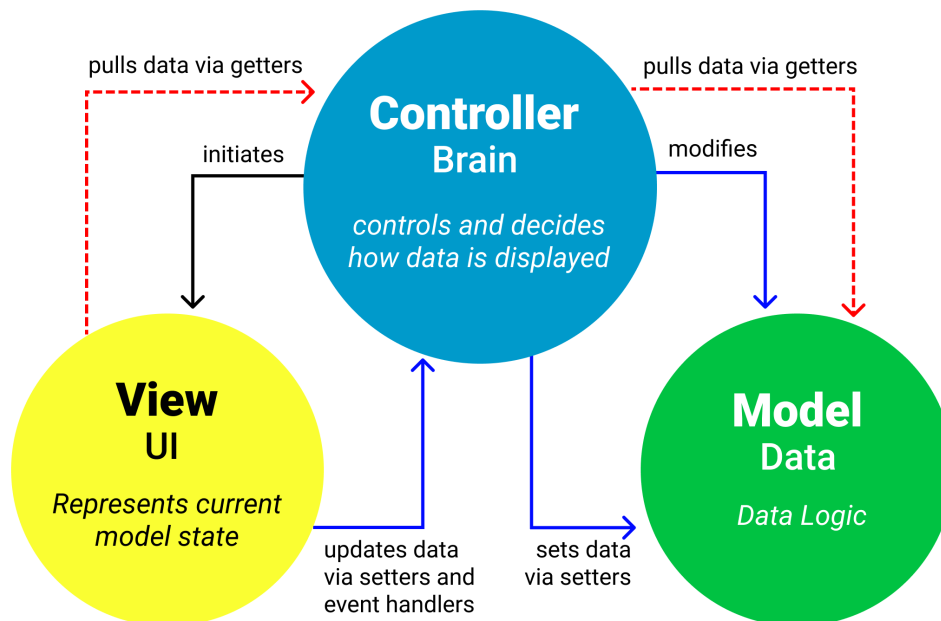


## I. Architecture Design - Thiết kế kiến trúc

1. Use a layered architecture to design the UWC 2.0 system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/APIs.

### 1.1 Architectural approach

## MVC Architecture Pattern



Hình 1: Mô hình kiến trúc MVC

### Hướng tiếp cận kiến trúc hệ thống: Mô hình MVC (Model - View - Controller)

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

**Model:** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. 1 controller thường có 2 chức năng:

- Application-specific logic là các chức năng cụ thể để phục vụ cho hệ thống như là lựa chọn Map, MCPs để hiển thị năng chính

- Data validation là các chức năng để kiểm tra tính hợp lệ của dữ liệu nhập vào như: tài khoản, mật khẩu của nhân viên,...

**View:** Đây là phần giao diện (theme) dành cho nhân viên. View là phương tiện hiển thị các đối tượng trong một hệ thống. Chẳng hạn như hiển thị giao diện trang chủ, giao diện tin nhắn, .... Nó bao gồm bất cứ thứ gì mà nhân viên có thể nhìn thấy được.

**Controller:** Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng. Như thực hiện cập nhật các phương tiện di chuyển hay cập nhật thông tin của các nhân viên ...

## 1.2 Modules

- **User management** : Quản lý các hoạt động liên quan đến danh tính người dùng, bao gồm xác thực (authentication) và phân quyền (authorization).  
-Authentication kiểm tra thông tin tài khoản (credentials) của người dùng nhập có đúng như đã lưu trong database hay không, nhằm xác định có cho phép người dùng đó vào hệ thống hay không.  
- Authorization kiểm tra người dùng có được cấp quyền để xem hoặc thực hiện một tác vụ nào đó hay không.

	Xác thực	Phân quyền
Input	Username hoặc email và password.	Tác vụ hoặc nội dung người dùng muốn dùng / xem, role của người dùng
Output	Cho phép người dùng vào hệ thống nếu thông tin đúng, từ chối nếu ngược lại.	Cho phép người dùng sử dụng / xem tác vụ hoặc nội dung nếu người dùng được cấp quyền, từ chối nếu ngược lại.

- **Database management**: Quản lý các hoạt động liên quan đến cơ sở dữ liệu, bao gồm tạo cơ sở dữ liệu và xử lý yêu cầu (xem thông tin và cập nhật dữ liệu cho MCP và nhân viên).

	Tạo CSDL	Xử lý yêu cầu
Input	Yêu cầu nghiệp vụ của ứng dụng.	Yêu cầu (xem, thêm, xóa hay sửa), đối tượng (nhân viên hay MCP) và dữ liệu tương ứng
Output	CSDL thỏa mãn yêu cầu được tạo, bao gồm các schema, model, relation, ... để lưu được thông tin, lịch làm việc của nhân viên và thông tin MCP và phương tiện.	CSDL được cập nhật hoặc trả về dữ liệu thích hợp cho controller tùy vào yêu cầu và dữ liệu tương ứng.

- **View management**: Quản lý giao diện (động và tĩnh) được trình bày ra đối với người dùng, bao gồm quản lý các thành phần UI liên quan (thành phần hiển thị thông tin nhân viên và phương tiện, lịch làm việc, ...) và xử lý logic cùng với user event.

	Quản lý thành phần UI	Xử lý logic và user event
Input	Thiết kế các thành phần UI (mockup, wireframe, ...).	Các điều kiện logic (if else) và user event (click, rê chuột, ấn phím, ...).
Output	Thành phần UI tương ứng được hiển thị trên màn hình cho người dùng (trang web thông tin, trang web lịch làm việc, ...).	Hiển thị các yếu tố nhất định trên UI tùy thuộc vào điều kiện logic; thực hiện các tác vụ tùy thuộc vào user event.

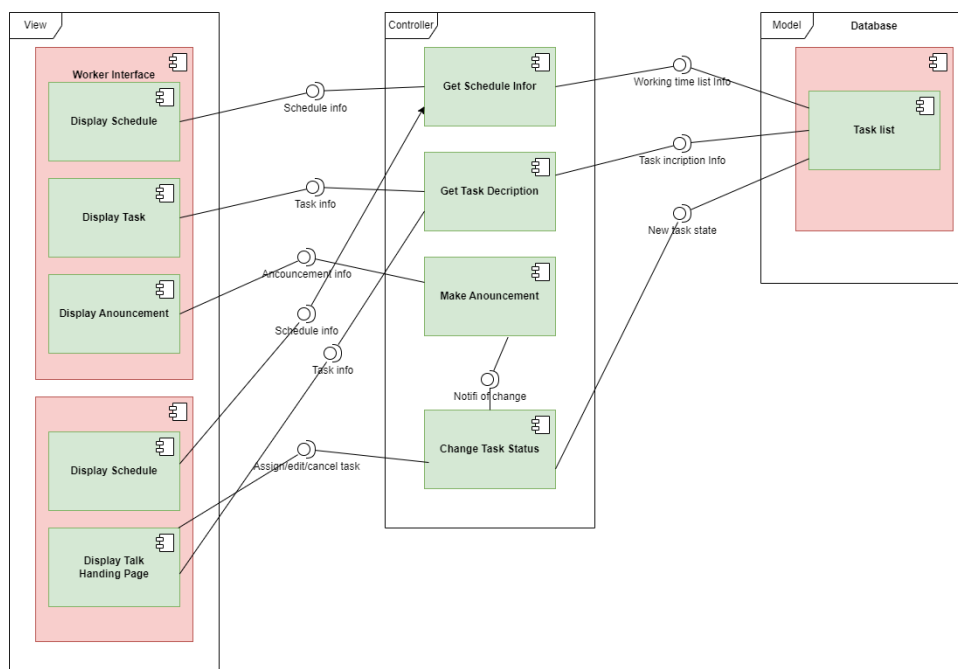
- **Application logic processing**: Xử lý toàn bộ các logic của ứng dụng, bao gồm giao việc (back officer cho collector và janitor), chọn người và phương tiện thích hợp, tạo đường đi và gửi thông báo / giao tiếp giữa các nhân viên.

	Chọn người, phương tiện	Tạo đường đi	Thông báo / giao tiếp	Giao việc
Input	Tình trạng các collector, janitor và phương tiện.	Tuyến đường thực tế (map) và thông tin các MCP.	Người gửi, người nhận và thông báo / tin nhắn cần chuyển	Thông tin hoàn chỉnh về đường đi, phương tiện và người được chọn.
Output	Người và phương tiện thích hợp được chọn.	Đường đi tối ưu được tạo.	Thông báo / tin nhắn được gửi như mong muốn.	Người được chọn nhận được thông tin về tuyến đường và phương tiện.

UWC 2.0 sẽ có 4 module: User management (quản lý người dùng), database management (quản lý cơ sở dữ liệu), view management (quản lý giao diện) và application logic processing (xử lý logic ứng dụng).

## 2. Draw a component diagram for the Task Assignment module.

### 2.1 Assign Task

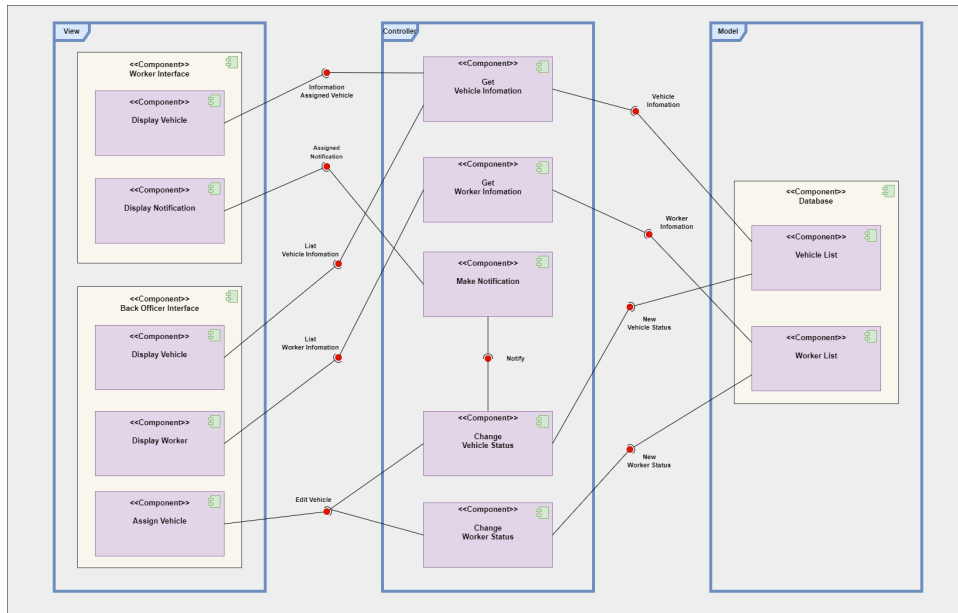


Hình 2: Component Diagram - Assign Task

#### Mô tả

- Trong giao diện của Collector và Janitor đều có khối hiển thị thời khóa biểu (Display Schedule), khối hiển thị nhiệm vụ (Display Tasks) và khối hiển thị thông báo (Display Announcements).
- Khi Collector và Janitor xem thời khóa biểu thì khối Schedule Info sẽ lấy thời gian làm việc từ Database Task List và hiển thị trên khối Display Schedule, tương tự với khối Display Tasks
- Trong giao diện của Back Officer có khối Display Schedule tương tự với Janitor và Collector, khối xem và chỉnh sửa nhiệm vụ (Display Tasks Landing Page)
- Khi Back Officer thêm/thay đổi/hủy một nhiệm vụ thì thông qua khối Change Task Status sẽ gửi trạng thái thay đổi đến Database Task List và tạo ra thông báo gửi đến Janitor và Collector

## 2.2 Assign Vehicle

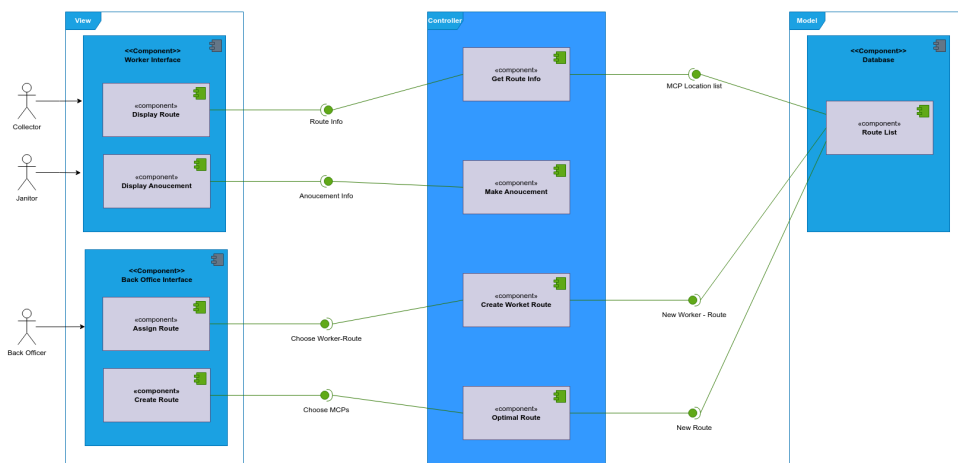


Hình 3: Component Diagram - Assign Vehicle

### Mô tả

- Trong giao diện của Collector và Janitor đều có khối hiển thị phương tiện được giao (Display Vehicle) và hiển thị thông báo (Display Notification).
- Trong giao diện của Back Officer có khối giao phương tiện (Assign Vehicle), khối hiển thị công nhân (Display Worker) và khối hiển thị phương tiện (Display Vehicle).

## 2.3 Create Route



Hình 4: Component Diagram - Create Route

### Mô tả

- Trong giao diện của Collector và Janitor đều có khối hiển thị đường đi (Display Route) và hiển thị thông báo (Display Announcements).
- Trong giao diện của Back Officer có khối tạo đường đi (Create Route) và gán đường đi (Assign Route).