ARCHITECTURE DESIGN

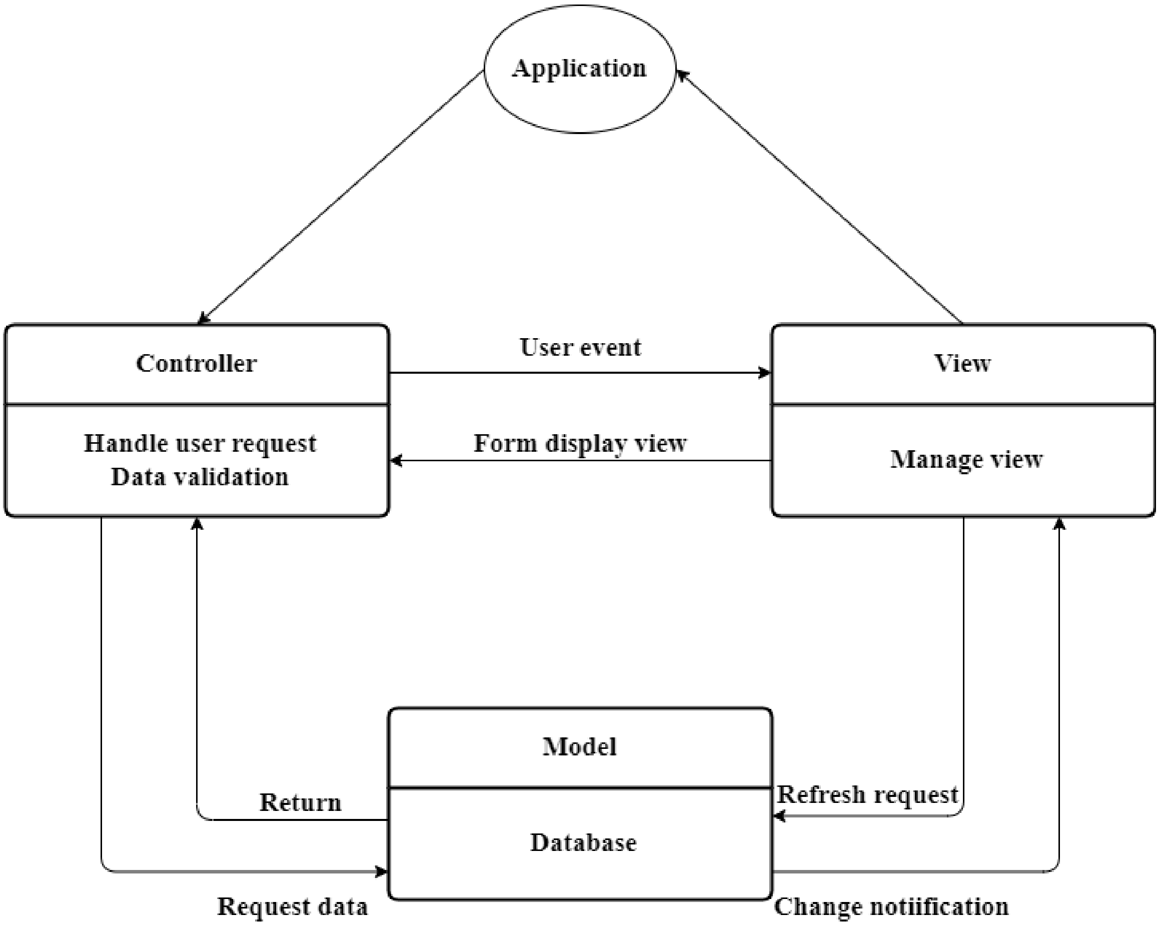
Update - 8 April, 2023

Contents

1 Architecture Design

1.1 Layered architecture

1.1.1 Architectural diagram



Hình 1: Overall design of UWC 2.0 system

We decided to go with the **MVC model** or it can be understood as **3-layer architecture model**, a software architectural pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements. Those three main components include Model, View, and Controller, each is built to handle specific aspects of an application.

- Handle user request: Handle all the actions performed on data through interaction with the user, for instance, assign janitor or collector to task, update task, and notify fully loaded MCPs,...
- Data validation: Check for valid data from user input like username, password,...
- Manage view: Displayed and update detail view for the user corresponding to their request.

- Database: Store information about the janitor, collector, vehicle, task, and MCPs,... It also includes methods to communicate with the data through basic query operations.

1.1.2 Presentation strategy

We will use the **module** named *Authorization* with **input** is *User_ role, signature*; **output** is *approriate UI/UX* and this **module's function** is to check the role of the user and see if the signature is valid. The module will give the appropriate UI/UX if the signature is valid.

1.1.3 Data storage

We will use the **module** named *Account Management (only the system admin can access)* with **input** is *account_ id, new_ role*; **output** is *state* and this **module's function** is after creating a new account for a new employee, or an employee gets promoted to a new position in the company, the admin will provide that account with a proper interface according to the account owner's position in the company.

1.1.4 API management

We will use the **module** named *Task Assignment* with **input** is *employee_ account_ id*; **output** is *update task* and this **module's function** is to connect to the database and give the back officers information about employees, shifts, routes, vehicles, and MCPs, allowing the back officers to assign the upcoming task based on that information. Save the task assigning information to the database.

1.2 Draw component diagram for Task assignment module

View Component

- Contain 2 sub-components which are the back-officer interface and the field-worker interface. The back-officer component also contains many sub-components.
- These components represent the interfaces for viewing workers, vehicles and MCPs information. Data is retrieved from the corresponding controller components.
- The Display task component will display the task information retrieved from the database.
- The Task assignment component provides an interface for the back officers to assign the workers, vehicles and MCPs information to a specific task. It will also send the assigned task back to the task processor.
- The Display announcement task is to announce any message the back officer give.
- The field-worker interface components display the given task of the employee.

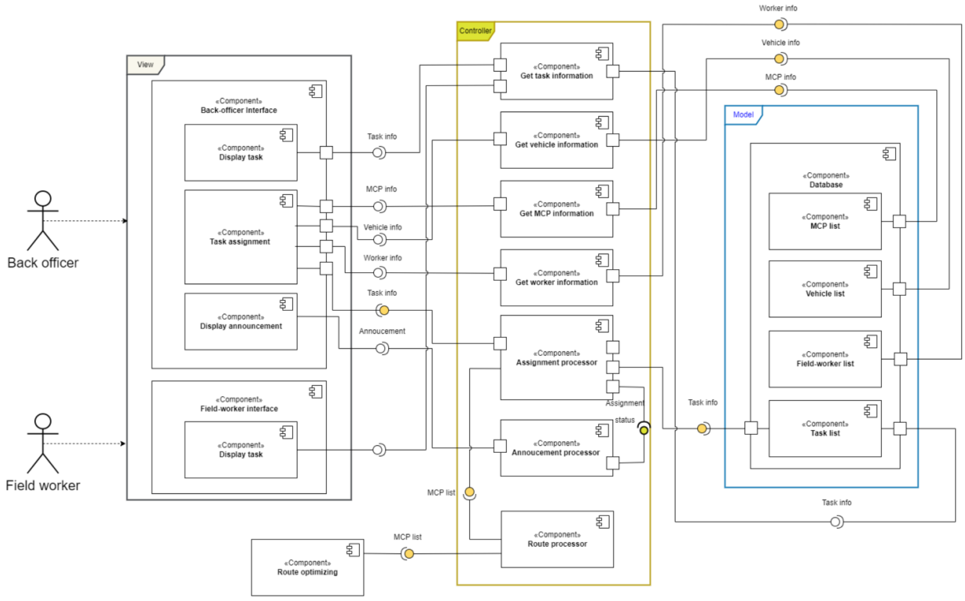
Controller Component

- Contains sub-components.
- Provide interfaces to update data in the database and also make annoucements.
- The get components for the task, vehicle, workers and MCPs retrieve data from the database and give it to the corresponding display components.

- The task processor retrieves the assigned data by the back-officer to edit/create or delete a task and update it to the database, it will also exchange MCPs information with the route processor to create a route for each assigned task.
- The announcement processor will announce each successful/failed task assignment.

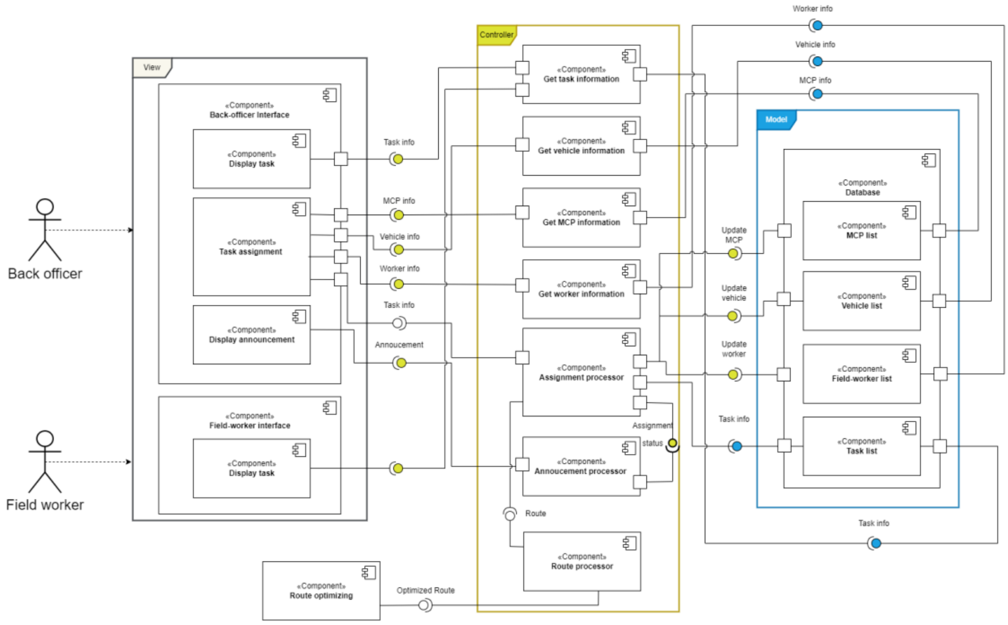
Model Component

- Contains sub-components
- The components use the interface from the controller components to send and retrieve data as well as make changes to the database.
- There is component which is only used to optimize the route based on the given MCP list, which information is given by the route processor components.



Hình 2: Component diagram

The Component diagram above describe the request side of the components. The view components request data from the controll component which request its data from model component. The controller also request data from view component and the route optimization for assigning task. The Component diagram above describe the response side of the components. The View component receive data from controller component. The controller component also receive data from model component. It also receive data from view component and route optimization processor for assigning task and make change to database in the model component.



Hình 3: Components diagram