VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY

Software Engineering
Assignment – Task 3

Author:

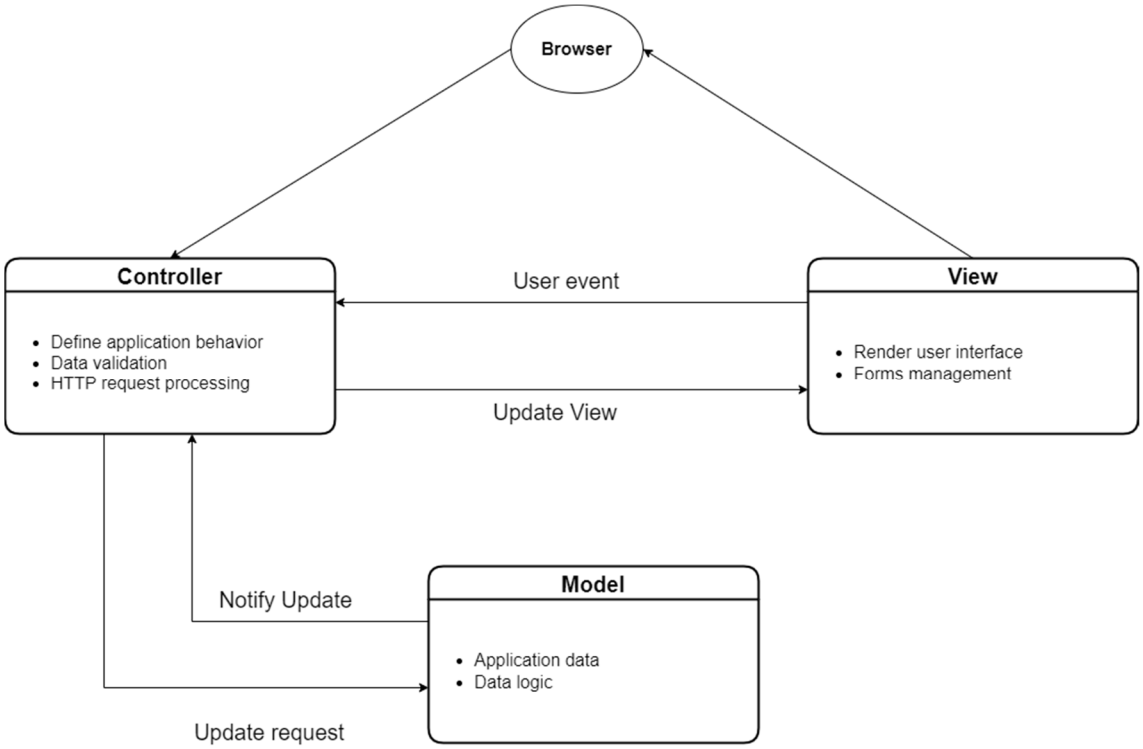
Name	Student ID	Class
Nguyễn Nam Kha	2052515	CC03
Nguyễn Ngọc Hòa	2052485	CC03
Hồ Nguyễn Ngọc Bảo	2052036	CC03
Phan Mai Tấn Lợi	2052158	CC03
Đào Quang Vinh	2053586	CC03

Ho Chi Minh city, 11/2022

Contents

3.1 Architectural approach

3.1.1 MVC Model



	Description
Model	Model is responsible for manage system data and interact with database as well as responds to request from Controller. • Application data structure: Provide data structure for data stored in database. • Data related logic: Provide data's related logic or operation to manipulate data such as create or update data (task, worker, vehicle, …). It has several functions such as storing user information, storing MCP's data, vehicles' data, checking MCP's capacity, change workers' status, assign task to workers…

View	View defines layout, displays application data through UI and processes user's interaction (click, scroll, …) and send to controller. • Render user interface: Provide UI to display data. In our application, it is responsible for displaying login screens, worker screen, task assignment screen, back officer calendar, dashboard, notification,… • Forms management: Handle user's input and pass to controller.
Controller	Controller manages user's interaction and contains logic that updates the Model or View in response to input from the users of the application. • Define application behavior: Mapping user's action to model updates or view updates. • Data validation: Validate input from user such as ID, password, …

3.1.2 Modules

1. Message

- Input: MCPs' data from database, message content, sender's ID and receiver's ID
- Output: None
- Function: This module allows Back Officers, Janitors and Collectors communicate to each other. Additionally, this module will notify Back officers, Janitors and Collectors whenever their assigned MCPs reaches max capacity.

2. User Authentication

- Input: User's ID and Password
- Output: None
- Function: Verify user ID and password with the information from the database and grant user access to different feature.

3. Worker(collectors, janitors) management

- Input: User's ID
- Output: Worker's info
- Function: This module will retrieve the collector or janitor's information such as work calendar, assigned vehicle. Also, this module will update the working status when worker check in or check out.

4. Vehicle/trolley management

- Input: User's ID or vehicle/trolley 's ID
- Output: Vehicle/trolley's info
- Function: This module provides the vehicle or trolley's information such as their technical details, their assigned worker.

5. MCP management

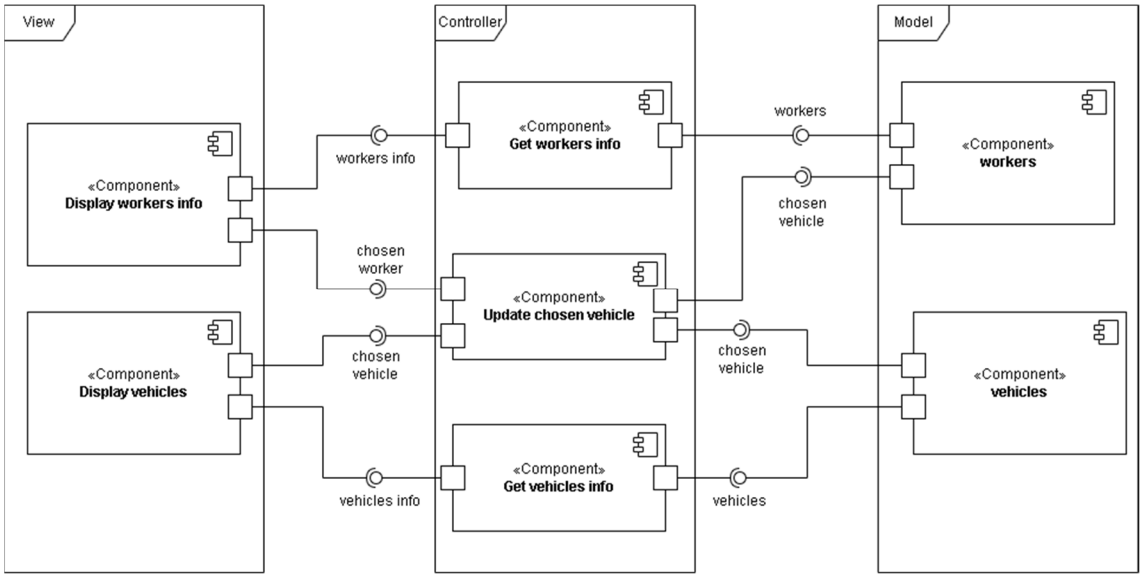
- Input : MCP's ID
- Output: MCP information
- Function: This module retrieves MCP information including: capacity, assign worker, location, name.

6. Task assignment

- Input: workers' data, MCPs' data, vehicle information
- Output: None
- Function:
  - Back officers can choose MCPs and create routes using a route-finding API that takes fuel consumption and distance into account
  - Back officers can then assign the chosen MCPs to workers and created routes to Collectors

3.2 Component diagram for Task Assignment module

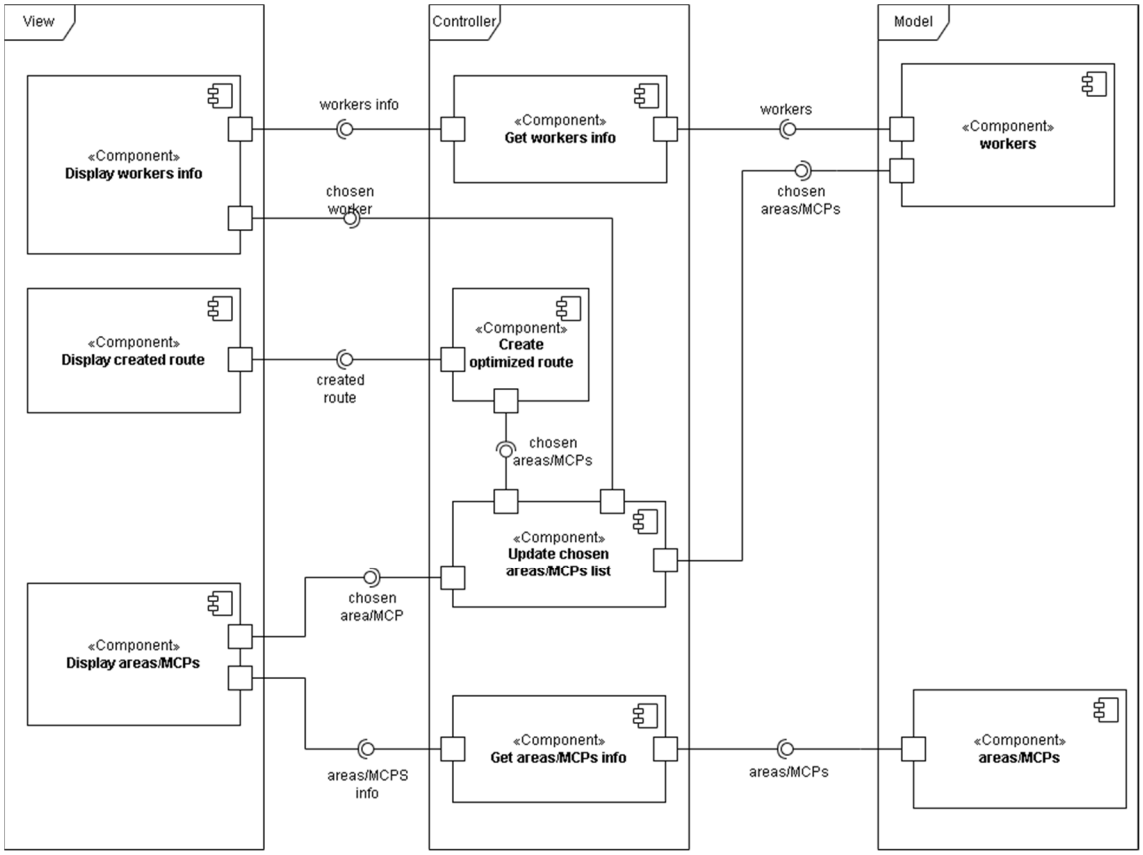
3.2.1 Assign vehicles module



Description:

- Back Officers' interfaces consist of *Display workers info* and *Display vehicles*.
- *Display workers info* sends request to retrieve workers' information from the database through the *Get workers info*, which will be displayed to the screen. The same behavior applies to *Display vehicles*.
- Back Officers choose one worker from displayed workers and one vehicle from displayed vehicle and send their information to *Update chosen vehicle*.
- Workers' information includes assigned vehicle and vice versa. *Update chosen vehicle* picks chosen worker and assign chosen vehicle to it, also, pick chosen vehicle and assign chosen worker to that vehicle.

3.2.2 Assign areas/MCPs/route module



Description:

- Back Officers' interfaces consist of *Display workers info*, *Display created route* and *Display areas/MCPs*.
- *Display workers info* sends request to retrieve workers' information from the database through the *Get workers info*, which will be displayed to the screen. The same behavior applies to *Display areas/MCPs*.
- Back Officers choose one worker from *Display workers info* and areas/MCPs from *Display areas/MCPs*, then create optimized route using API and display it on screen.
- *Update chosen vehicle* pick chosen worker and assign areas/MCPs to it, workers will not store optimized route because workers can always get optimized route via API.