

VIETNAM NATIONAL UNIVERSITY,
HCMC UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



SOFTWARE ENGINEERING (CO3001)
URBAN WASTE COLLECTION AID
UWC 2.0

Task 3: Architecture Design

Professors

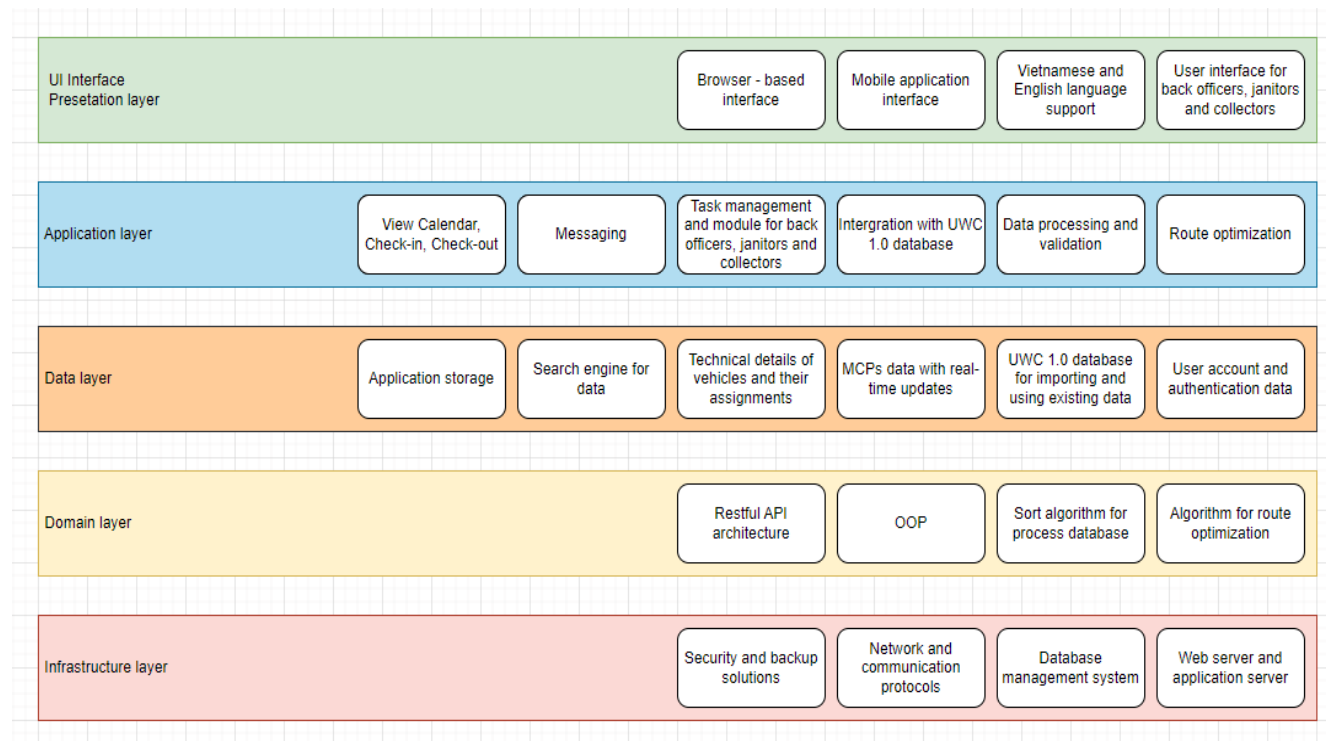
Bùi Hoài Thắng & Nguyễn Đức Anh

Members

Trần Phạm Minh Đăng	2052070
Ngô Trương Trọng Nghĩa	2053264
Đinh Xuân Quang	2053359
Nguyễn Hoàng Thuận	2052729
Nguyễn Trần Quốc	2053384

Ho Chi Minh city, March 2023

3.1 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.



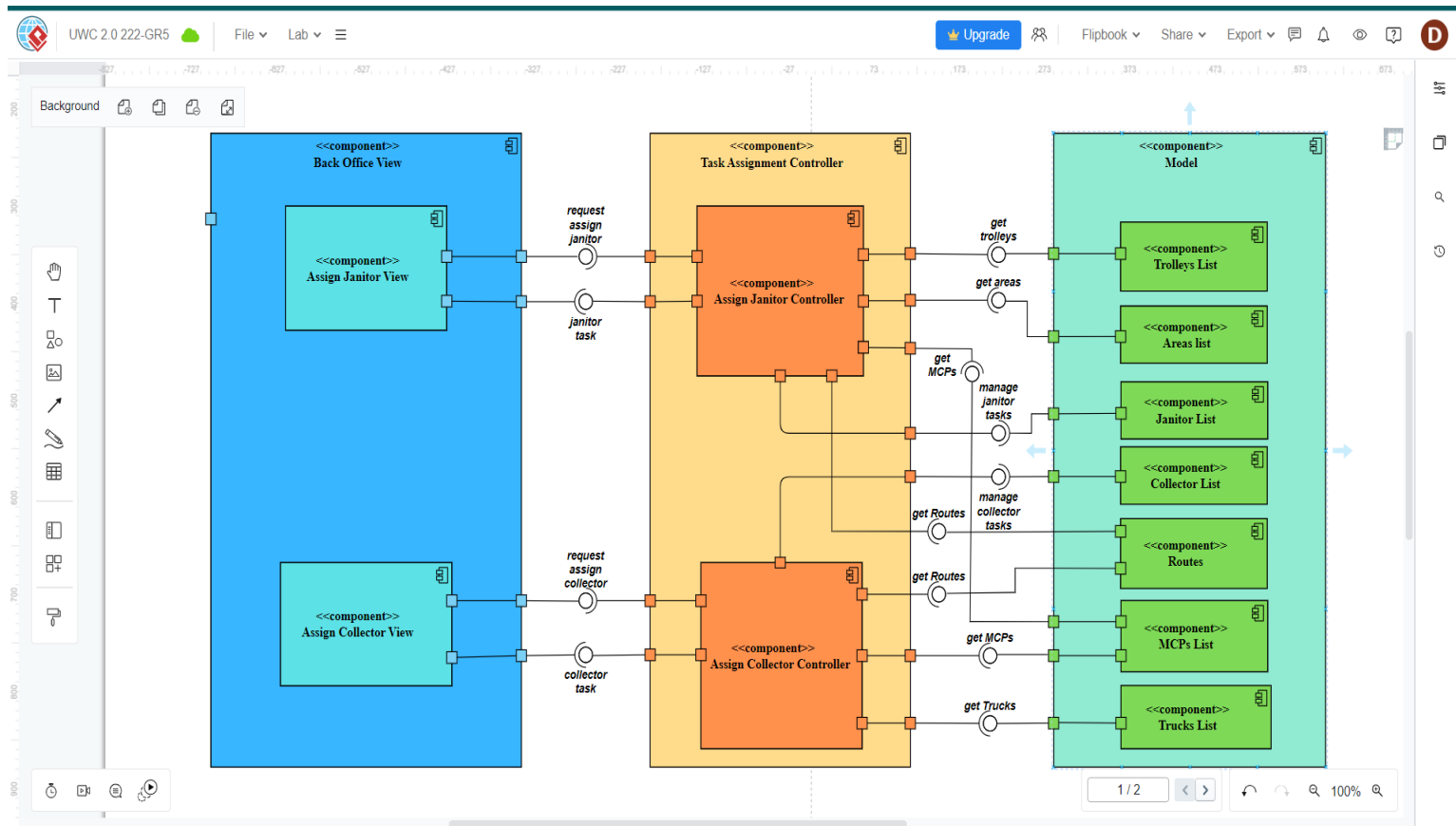
- **The presentation layer** of the UWC 2.0 system will be implemented using a web-based interface with HTML, CSS, and JavaScript. The interface will prioritize ease of use and quick access to important information. To minimize horizontal scrolling, all critical information will be displayed in a single view, and lists will be sorted alphabetically by default. The system will be designed to be compatible with most devices and operating systems. Further enhancements to the user interface will be based on user feedback and usability testing.

- **The data generated** by the UWC 2.0 system uses a database management system. MySQL, a popular open-source relational database management system, can provide a reliable and efficient way to store and manage the data. MySQL can handle large amounts of data and is easy to scale up or down based on the system's needs. Using MySQL will allow the UWC 2.0 system to store, manage, and retrieve data efficiently, ensuring that the system can handle the expected growth in data volume over the next

five years. More convenient for us to customize worker, vehicle, trolley, area, MCP information.

- **The API management** strategy for UWC 2.0 will be based on a RESTFUL API design pattern. The APIs will be designed to be easy to use and will follow industry-standard conventions for resource naming, HTTP methods, and response formats. The APIs will be secured using OAuth2.0 authentication and will support rate limiting to prevent abuse. API management refers to the process of designing, publishing, securing, monitoring, and analyzing the use of application programming interfaces (APIs) in a scalable and secure manner. APIs are sets of protocols, routines, and tools for building software applications that allow different software components to communicate with each other. Google Maps Platform API to generate the route based on the location of MCP as well as Area. Additionally, the route contains information about the traffic intensity and the estimated time.

3.2 Draw a component diagram for the Task Assignment module



Descriptions: The Component diagram for Task Assignment module includes 3 big component

- **Back Officer Interface** component which is composed of **two** sub-components: the **Assign Janitor View** and the **Assign Collector View**. These sub-components utilize interfaces by use provided interfaces from Task Assignment Controller component to implement requests.

- **Task Assignment Controller** component which is composed of **two** sub-components: the **Assign Janitor Controller** and the **Assign Collector Controller**. Component **Assign Janitor Controller** will receive data from component **Trolleys List**, **Areas list**, **Routes** and **MCPs List** through interface **get trolleys**, **get areas**, **get Routes** and **get MCPs**. Besides, component **Assign Collector Controller** will receive data from component **Routes**, **MCPs** and **Trucks List** through interface **get Routes**, **get MCPs** and **get Trucks**. Moreover, Task Assignment Controller component then sends request to

component Janitor List and Collector List to store data about the new task in the database, then it will send response about the new task to component **Assign Janitor View** and **Assign Collector View**.

- **Model** component which is composed of **seven** sub-components: **Trolleys List, Areas List, Janitor List, Collector List, Routes, MCPs List** and **Trucks List**. Provide Interfaces for **Assign Janitor Controller** and the **Assign Collector Controller**.