Task 3.1

Our system consists of a front-end and a back-end communicating with one another through API calls.

The architecture of our system consists of these modules (details about the architecture of the system is also described in terms of these modules.)

Database management module

The database management module manages three tables: Employee, Vehicle, and Task. Route data is stored in the Task table and schedule data is stored in the Employee table.

This module receives SQL commands as its input. As for retrieve commands, it returns approriate data as output.

Data access module

Data access module is an abstraction on top of the database management module. This module allows users (mostly programmers) to access their authorized database (employee, vehicle, and task.) Via this module, users can also access route and schedule data as if they were real tables (these virtual tables are automatically constructed with data retrieved from the employee, vehicle, and task tables.)

This module also handle data accesss authorization.

This module receives authorization token in every request header.

As for create and update API calls, this module receives additional data from the request's body as its input.

As for create, update, and delete API calls, this module returns HTTP status code OK if the command is successfully executed, otherwise error code.

As for view API calls, this module outputs data in the body of the response and a HTTP status code OK successfully executed, otherwise error code.

Task assignment module

This module retrieves information from the data access module and automatically schedule appropriate routes, calendar, vehicles for approriate employees, then update new information via data access module.

This module receives authentication token in API request's header. It returns HTTP status code OK and task data in the response's body if successfully executed, otherwise error code.

Message module

This module handles messaging between users of the system.

This module receives authentication token in API request's header and message content in API request's body. It returns OK if successfully executed, otherwise error code.

Authentication module.

This module handles authentication.

This module receives authenticate information in API request's body. It returns OK and authentication token if successfully executed, otherwise error code.

User interface module:

This module translate GUI interaction into equivalent API call in task assignment module, message module, and authentication module.

This module receives user interaction (clicking, typing...) as inputs.

This module display data and message into ouput formats approriate for non-coding users.

This module also display error code returned by API calls into user-friendly formats.

Task 3.2