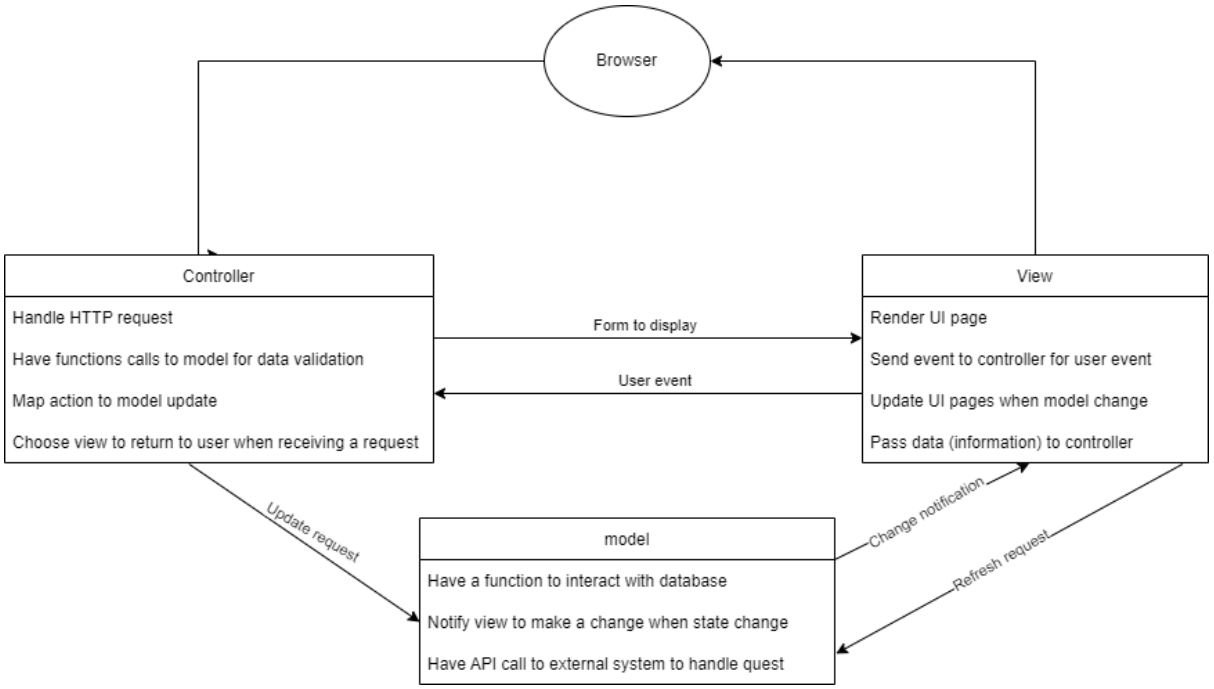
BTL Công Nghệ Phần Mềm - Quản lý rác thải đô thị UWC 2.0

Nhóm:

1. Trịnh Lâm Đăng Khoa - 2053143
2. Lại Đức Trung - 2053538
3. Nguyễn Võ Hàn Phong - 2052208
4. Đặng Nguyên Khánh - 2053109
5. Nguyễn Từ Hoàng - 2053016

3.1. Describe an architectural approach you will use to implement the desired system. How many modules do you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module



Name	Details of UWC architecture
Describe	- View + Render UI elements of different pages: ● BO pages ● Janitor and Collector pages + Interact with users by handling events automatically animated right in the view:

	● Auto sliders. ● Show Dropdown menu when handling click event user menu. + Pass data into the requests sent to the controller to handle the application logic: + When notified of state changes from the model, perform data retrieval from the model to refresh the view and update the new view. - Controller + Returns a view depending on the page requested. + Call functions from the model to perform operations on the data: + Call a function from a model to use APIs from external systems to data validation + Return response to user with corresponding request. - Model + Provides methods to process API calls to perform authentication for performing task validation with systems. + Provide methods to interact with the database to perform registration, login, and user account management. + Provide methods to perform database operations (add, delete, edit) with task management, MCP management features. + When the model's state (data,...) changes => notify the view to refresh and update the view.
Advantages of architecture	+ Each feature has many views and requires many operations with data: + For task management features and MCPs, users perform CRUD operations => need multiple views for the corresponding view, edit, and update operations. Each operation also needs to manipulate the corresponding data => separation between view, controller and model helps clear code implementation. + Using separation between model, view and controller makes it easy to extend the code. + Easily add requirements for interacting with data independently of display on the user's side.
troublesome	+ Code implementation is complex and needs to ensure efficient interaction between model - view - controller.

3.2. Draw an implementation diagram for Task Assignment module