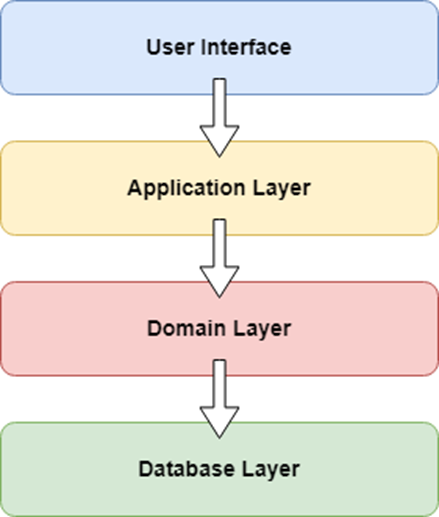
**Architecture design**

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

Nhóm em xây dựng kiến trúc của hệ thống WMC 2.0 system theo layered pattern (mẫu phân lớp):



Trong đó:

* Lớp thứ nhất: là những chương trình chịu trách nhiệm cho việc người dùng tương tác với hệ thống phần mềm.
* Lớp thứ hai: chứa các logic đáp ứng các yêu cầu chức năng.
* Lớp thứ ba: chứa các thành phần thuật toán và lập trình.
* Lớp thứ tư: là cơ sở dữ liệu dùng chung cho cả hệ thống.

Lý do nhóm em chọn kiến trúc này vì:

* Tính đơn giản: Kiến trúc này khá đơn giản và dễ học, dễ hiện thực
* Giảm sự phụ thuộc vì chức năng của mỗi lớp là tách biệt
* Dễ kiểm tra vì các thành phần được tách biệt, dễ dàng kiểm tra từng cái.

Nhóm em sẽ sử dụng 6 module để hiện thực WMC 2.0 system:

|  |  |
| --- | --- |
| Module | Account |
| Description | Người dùng sử dụng chức năng để truy cập vào hệ thống thông qua tài khoản. |
| Input | Đưa ra các thông tin để thực hiện tạo hoặc đăng nhập tài khoản, hoặc thao tác đăng xuất tài khoản. |
| Output | Hệ thống sẽ phản hồi đối với thông tin hoặc thao tác người dùng đưa ra để cho phép người dùng truy cập vào hệ thống hoặc thoát hệ thống. |
| Function | login: Người dùng đăng nhập vào hệ thống bằng thông tin đăng nhập  logout: Người dùng đăng xuất khỏi hệ thống  register: Người dùng đăng ký tài khoản mới.  updateInformation: Người dùng cập nhật thông tin của tài khoản |

|  |  |
| --- | --- |
| Module | Task management |
| Description | Nhân viên hậu cần sử dụng chức năng để quản lý thông tin các nhiệm vụ. |
| Input | Nhân viên hậu cần gửi các yêu cầu thao tác các nhiệm vụ |
| Output | Hệ thống sẽ phản hồi theo yêu cầu của người dùng |
| Function | CreatTask: tạo nhiệm vụ mới để có thể giao cho các nhân viên.  EditTask: chỉnh sửa thông tin nhiệm vụ như thay đổi khu vực, thay đổi nhân viên,...  removeTask: Xóa nhiệm vụ  UpdatedSchedule, - Cập nhật lịch trình  getÌnoSchedule: lấy thông tin của lịch trình  getInfoTask: lấy thông tin của nhiệm vụ  sentNotification gửi thông báo đến các nhân viên  filterTask: dùng để lọc các nhiệm vụ theo tiêu chí người dùng (từ khóa, ngày, thời gian, …) |

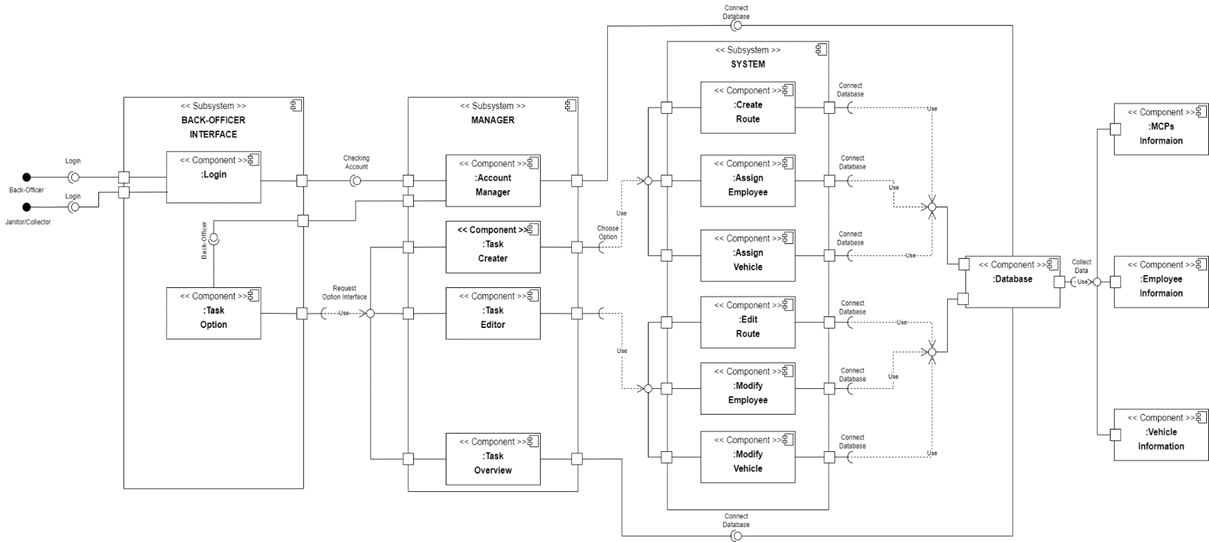
|  |  |
| --- | --- |
| Module | Task assignment |
| Description | Nhân viên hậu cần sử dụng chức năng có thể phân công nhiệm vụ cho các nhân viên khác với thời gian 2 tuần/lần. |
| Input | Người hậu cầu gửi yêu cầu phân công nhiệm vụ cho các nhân viên theo thời gian 2 tuần/lần. |
| Output | Hệ thống sẽ cập nhật lịch phân công nhiệm vụ theo thời gian 2 tuần/lần trên cơ sở dữ liệu và sẽ được gửi thông tin tới tài khoản của nhân viên được phân công. |
| Function | createTask: tạo lịch phân công nhiệm vụ cho 2 tuần tới, thường sẽ được tạo vào cuối tuần trước đó.  editTask: thay đổi các nhân viên làm việc trong một ngày cụ thể hay thay đổi các phương tiện trong ngày đó.  createRoute:  assignEmployee:  assignVehicle:  updateAssignment: dùng để cập nhật các nhiệm vụ lên trên cơ sở dữ liệu khi nhân viên hậu cần thực hiện xong createTask, editTask.  AssignNotification: để thông báo đến các nhân viên khi có lịch mới. |

|  |  |
| --- | --- |
| Module | Task performance |
| Description | Người thu gom và người lao công sử dụng chức năng để check-in, check-out nhiệm vụ được giao, |
| Input | Người dùng gửi các yêu cầu check-in, check-out các nhiệm vụ được chọn |
| Output | Các yêu cầu sẽ được hệ thống ghi lại trong cơ sở dữ liệu và nhân viên hậu cần có thể kiểm tra các thông tin đó. |
| Function | checkIn: nhân viên gửi yêu cầu đăng ký bắt đầu thực hiện nhiệm vụ.  checkOut: nhân viên gửi yêu cầu xác nhận đã hoàn thành nhiệm vụ. |

|  |  |
| --- | --- |
| Module | Communication |
| Description | Người dùng sử dụng chức năng có thể gửi và xem tin nhắn, nhận thông báo từ BO về MCPs. |
| Input | Gửi yêu cầu thực hiện các tin nhắn, gọi điện tới các nhân viên khác |
| Output | Các tin nhắn sẽ được gửi đến hòm thư người nhận, các cuộc gọi được cho phép kết nối |
| Function | updateMessages: cập nhật các tin nhắn và nhận các cuộc gọi đến hòm thư người dùng hoặc load các tin nhắn được gửi đến hòm thư trên cơ sở dữ liệu.  pushNotification: để hiển thị thông báo có tin nhắn hoặc cuộc gọi mà hòm thư vừa nhận được hoặc thông báo từ MCP.  init: dùng để khởi động updateMessages khi người dùng mở ứng dụng lần đầu sau khi ứng dụng được tắt hoàn toàn.  calling: dùng để duy trì kết nối với người ở đầu bên kia cuộc gọi mà người dùng gọi.  receiveMCPStatus: nhận các thông tin về trạng thái của các MCP từ cơ sở dữ liệu, đặc biệt là khi MCPs đầy  retrieveMessages: dùng để thực hiện truy vấn đến cơ sở dữ liệu và nhận các tin nhắn và cuộc gọi mà hòm thư của người dùng đã nhận được để lưu trong thiết bị dùng để gọi hàm  sendMessage: dùng để gửi tin nhắn và người dùng đã nhập đến hòm thư chỉ định.  makeCall: dùng để tạo cuộc gọi đến 1 người dùng được chỉ định.  filterMessages: dùng để lọc các tin nhắn hoặc cuộc gọi theo tiêu chí người dùng (từ khóa, ngày, thời gian, …)  pendingMessages: dùng để tạm thời lưu trữ các tin nhắn khi chưa thể gửi đi và sẽ đồng bộ chúng lên cơ sở dữ liệu và gửi đi khi có thể. |

|  |  |
| --- | --- |
| Module | View information |
| Description | Người dùng sử dụng chức năng có thể xem các thông tin được lưu trong cơ sở dữ liệu của chương trình. |
| Input | yêu cầu về các thông tin mà người dùng muốn được xem, yêu cầu được xác định khi người dùng dùng lệnh hoặc nhấn các nút trên GUI. |
| Output | các thông tin mà người dùng yêu cầu sẽ được trả về qua các dạng đã được xử lý như chuỗi ký tự được định dạng hoặc các cửa sổ GUI. |
| Function | initUI: hàm được chạy tự động mỗi khi chức năng được kích hoạt, dùng để khởi tạo UI mặc định và các chức năng của nó để chào đón người dùng và sẵn sàng nhận yêu cầu từ người dùng.  displayQueryResult: hàm được gọi sau khi các truy vấn đến cơ sở dữ liệu hoàn tất, dùng để xử lý các kết quả truy vấn và tổ chức cách hiện thị chúng cho người dùng.  getEmployee: hàm nhận lệnh từ UI và sau đó tạo nên các truy vấn đến cơ sở dữ liệu và thu các dữ liệu liên quan đến nhân viên (tên, ngày sinh, số điện thoại, địa chỉ, số căn cước, công việc, ngày vào làm, …)  getVehicle: hàm nhận lệnh từ UI và sau đó tạo nên truy vấn đến cơ sở dữ liệu và thu các dữ liệu liên quan đến các phương tiện hiện có của công ty (tải trọng, tiêu thụ nhiên liệu, số xe, sức chứa, . . .)  getRoute: hàm thực hiện truy vấn về các lộ trình đã được tạo với cơ sở dữ liệu và nhận kết quả trả về từ cơ sở dữ liệu.  getWorkingCalendar: hàm thực hiện truy vấn lịch làm việc trong tuần hoặc ngày của tất cả các nhân viên đến cơ sở dữ liệu.  getTask: truy vấn cơ sở dữ liệu các (task) của các nhân viên.  queryResultFilter: hàm dùng để lọc các kết quả truy vấn theo ý muốn của người dùng.  importExport: dùng để nhập lịch trình đã có và hiển thị hoặc xuất lịch trình ra các dạng khác (pdf, excel, …) |

**2) Draw an implementation diagram for Task Assignment module**

****

**Description:**

Hệ thống của chức năng Task Assignment gồm 3 Subsystems: Back-Officer Interface ( Giao diện người dùng ), Manager ( Trình quản lý chức năng ) và System ( Chức năng hệ thống ).

1. Back-Officer Interface: Gồm 2 component:

· Login: Đăng nhập tài khoản.

· Task Option: Chọn công việc cần thực hiện.

2. Manager: Hệ thống quản lý: Gồm 4 component:

· Account manager: Quản lý/Kiểm tra tài khoản có hợp lệ không thông qua dữ liệu của database.

· Task Creater: Tạo công việc cho các Janitors và Collectors: bao gồm tạo khu vực/đường làm việc; chỉ định nhân viên và chỉ định phương tiện di chuyển/làm việc phù hợp.

· Task Editor: Thay đổi/Chỉnh sửa công việc/lịch trình cũng như khu vực làm việc của các Janitors và Collectors.

· Task Overview: Tổng quan các công việc/lịch trình làm việc của Janitors và Collectors.

Tất cả dữ liệu đều lấy từ hệ cơ sở dữ liệu Database, ngoài ra Database còn tiếp nhận thêm các dữ liệu về các MCPs, dữ liệu của nhân viên và của các phương tiện được cập nhật thường xuyên.