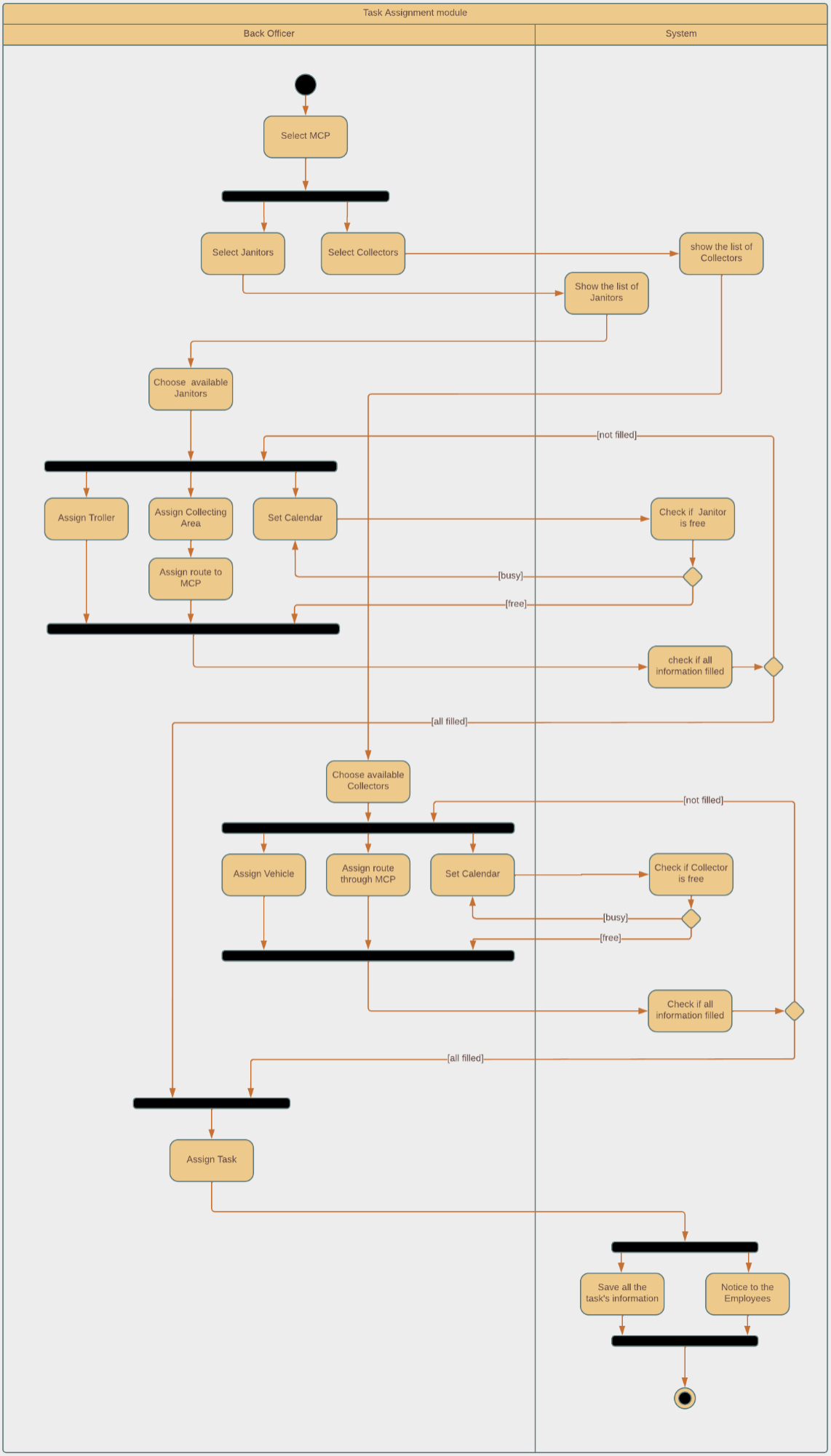
BTL Công Nghệ Phần Mềm - Quản lý rác thải đô thị UWC 2.0

Nhóm:

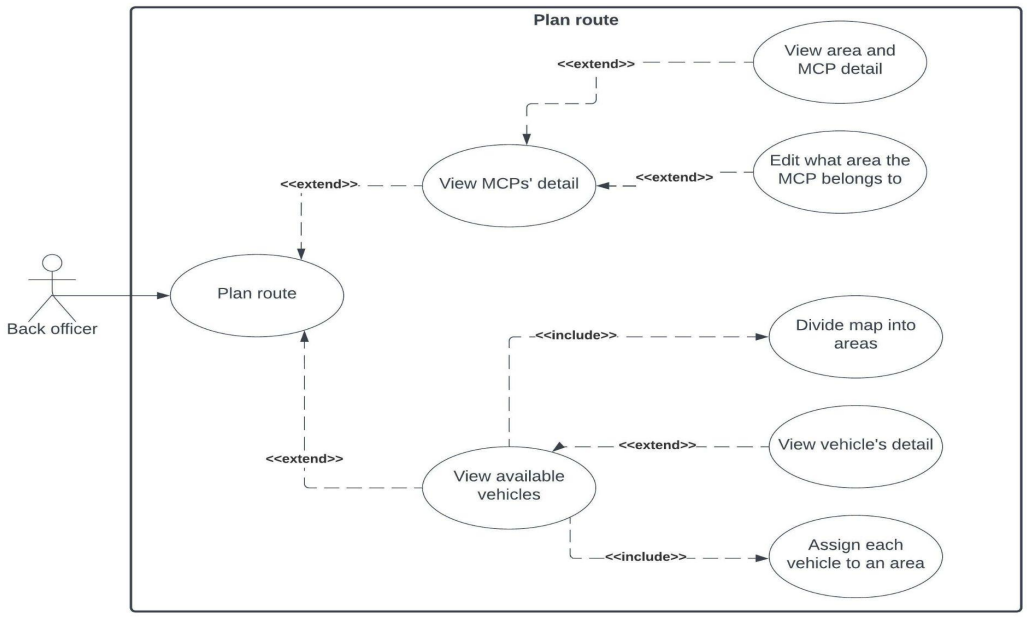
1. Trịnh Lâm Đăng Khoa - 2053143
2. Lại Đức Trung - 2053538
3. Nguyễn Võ Hàn Phong - 2052208
4. Đặng Nguyên Khánh - 2053109
5. Nguyễn Từ Hoàng - 2053016

2.1. Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module


Task Assignment module
Back Officer
System
Select MCP
Select Janitors
Select Collectors
show the list of Collectors
Show the list of Janitors
Choose available Janitors
[not filled]
Assign Troller
Assign Collecting Area
Set Calendar
Check if Janitor is free
Assign route to MCP
[busy]
[free]
check if all information filled
[all filled]
Choose available Collectors
[not filled]
Assign Vehicle
Assign route through MCP
Set Calendar
Check if Collector is free
[busy]
[free]
Check if all information filled
[all filled]
Assign Task
Save all the task's information
Notice to the Employees

2.2. Proposal a conceptual solution for the route planning task and draw a sequence diagram to illustrate it

2.2.1. Plan route



Use-case name	**Plan route**
Actor	Back officer
Description	Back officer wants to plan the route for vehicles
Trigger	Click “Plan Route” button
Pre-conditions	User is logged-in as Back Officer

Normal flow	1. System displays a map showing all the MCPs around the waste processing plant, a calendar and show 2 options: - View available vehicles - View MCPs' detail 2. Back officer choose the date to plan route in the calendar 3. Back officer click 1 of 2 options 4. System displays a smaller screen of the chosen list next to the map 5. Back officer clicks "Main Menu" button to leave the screen 6. System returns to main interface
Exceptions	None
Alternate Flows	**Alternate Flow 1: at step 2** 2a. Back officer choose "View available vehicles", System implements use-case "View available vehicles", ***continue to step 3 in Normal Flow***

	2b. Back officer choose “View MCPs’ detail”, System implements use-case “View MCPs’ detail”, ***continue to step 3 in Normal Flow***

2.2.1a. View available vehicles

Use-case name	**View available vehicles**
Actor	Back officer
Description	Back officer wants to view details of available vehicles and choose vehicle to put to work
Trigger	Click “View available vehicles” button
Pre-conditions	User is logged-in as Back Officer
Normal flow	1. System displays a list of available vehicles and their details (maximum load, fuel) 2. Back officer ticks the checkboxes next to the vehicles that need to put to work 3. Back officer clicks “Next” to confirm the choosing action

	4. System divides the map into areas with different colors (The number of areas is equal to the number of vehicles) (The map is divided using Google Distance Matrix API and algorithms) 5. Back officer drags the vehicle bar to each area to assign work for that vehicle 6. Back officer clicks “Save” to verify the assigning action 7. System displays the message “Are you sure to save these changes?” 8. Back officer clicks “Yes” button to verify modification 9. System updates the status of vehicles as “Assigned” in the Database of the chosen date and displays the message “Changes saved” 10. Back officer clicks “OK” button 11. System returns to “Plan Route” screen
Exceptions	**Exception Flow 1: at step 3** 3a. If no vehicle is chosen, System will display the message “Cannot divide areas. Please choose vehicles(s) and try again!”.

	Back officer ticks the checkboxes next to the vehicles, ***continue to step 3 in Normal Flow***
Alternate Flows	**Alternate Flow 1: at step 3** 3a. If Back officer wants to unchoose specific vehicles, then untick each vehicle or click "Untick all" button, ***continue to step 3 in Normal Flow*** **Alternate Flow 2: at step 6** 6a. If Back officer wants to unassign specific vehicles, then hover on each vehicle bar and click the Trash icon, or click "Unassign all" button on top of the list, ***continue to step 6 in Normal Flow*** **Alternate Flow 3: at step 8** 8a. If Back officer does not want to verify modification anymore, then click "Cancel" button, system returns to "View available vehicles" screen. ***Continue to step 3 in Normal Flow***

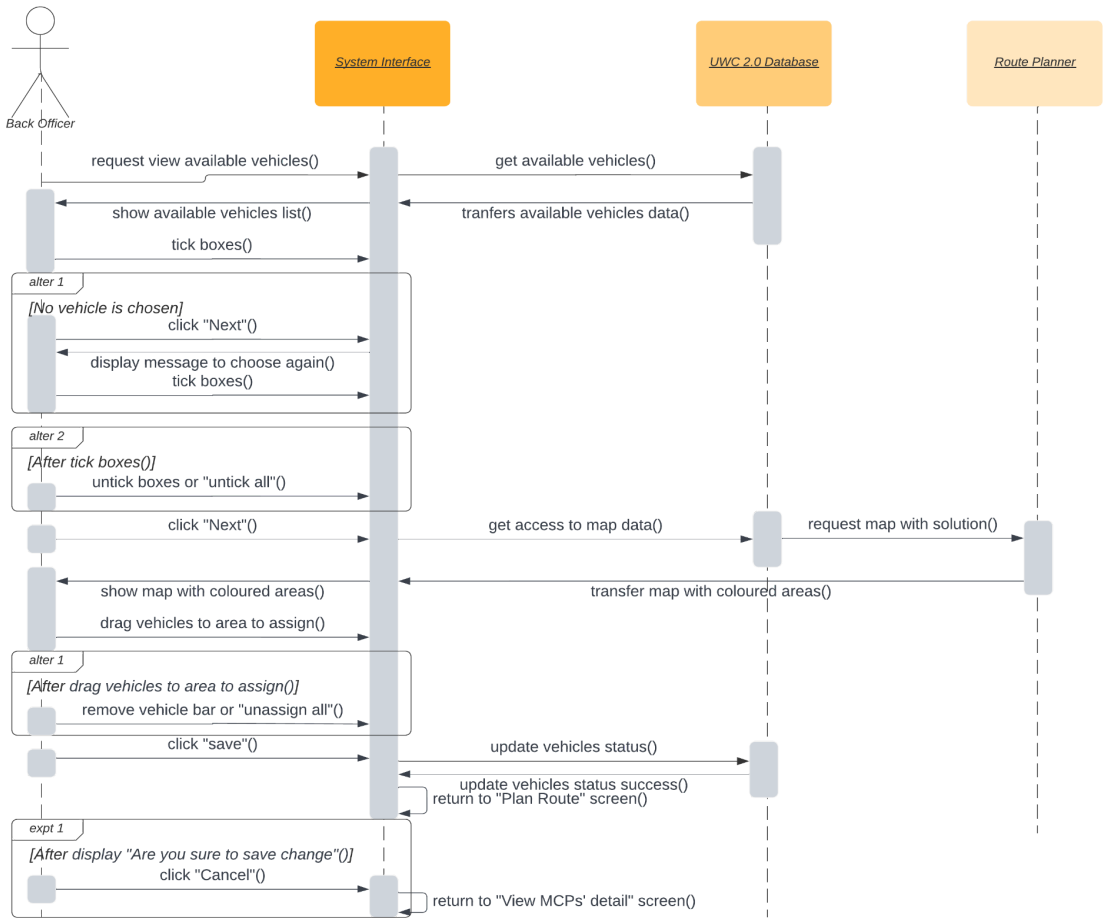
2.2.1b. View MCPs' detail

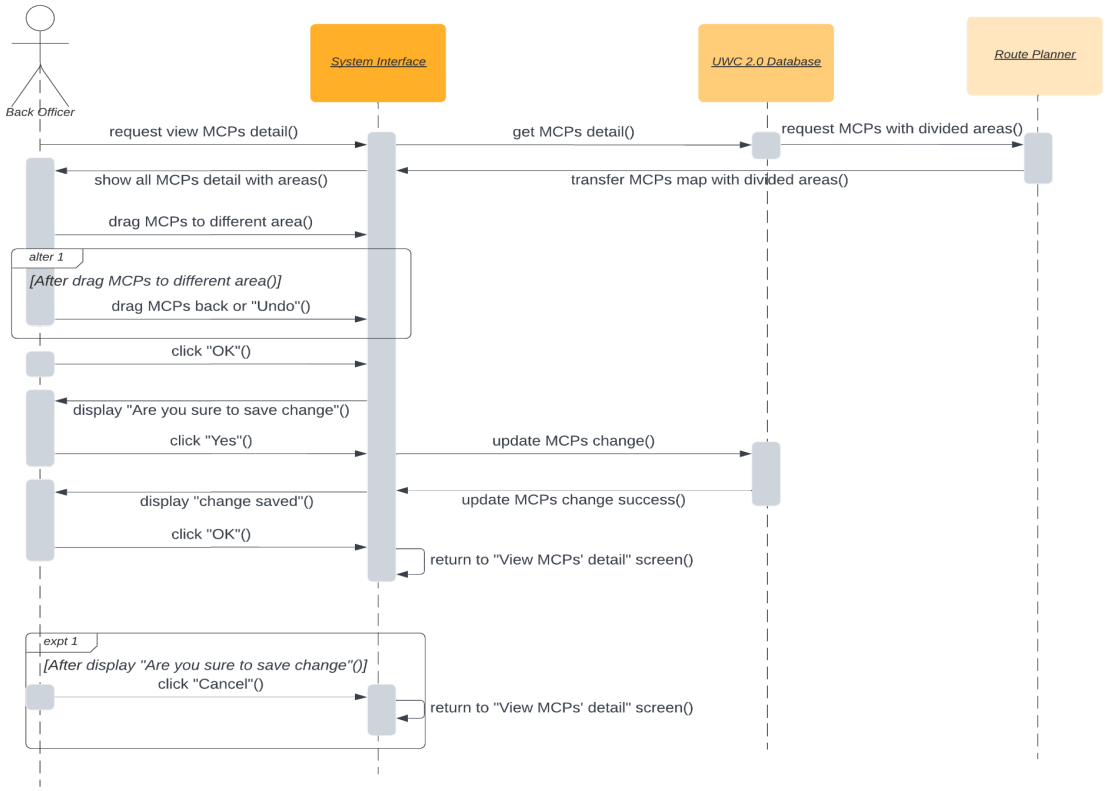
Use-case name	**View MCPs' detail**

Actor	Back officer
Description	Back officer wants to view details of MCPs in each divided areas and may be able to change MCP to different area
Trigger	Click “View MCPs’ detail” button
Pre-conditions	User is logged-in as Back Officer
Normal flow	1. System displays the list of all MCPs alphabetically or sorted by area (if Back officer had divided areas beforehand). A total distance indicator is located below each area 2. Back officer drags an MCP bar to change it to different area 3. Back officer clicks “OK” button 4. System displays the message “Are you sure to save these changes?” 5. Back officer clicks “Yes” button to verify modification 6. System displays the message “Changes saved” 7. Back officer clicks “OK” button 8. System returns to “Plan Route” screen

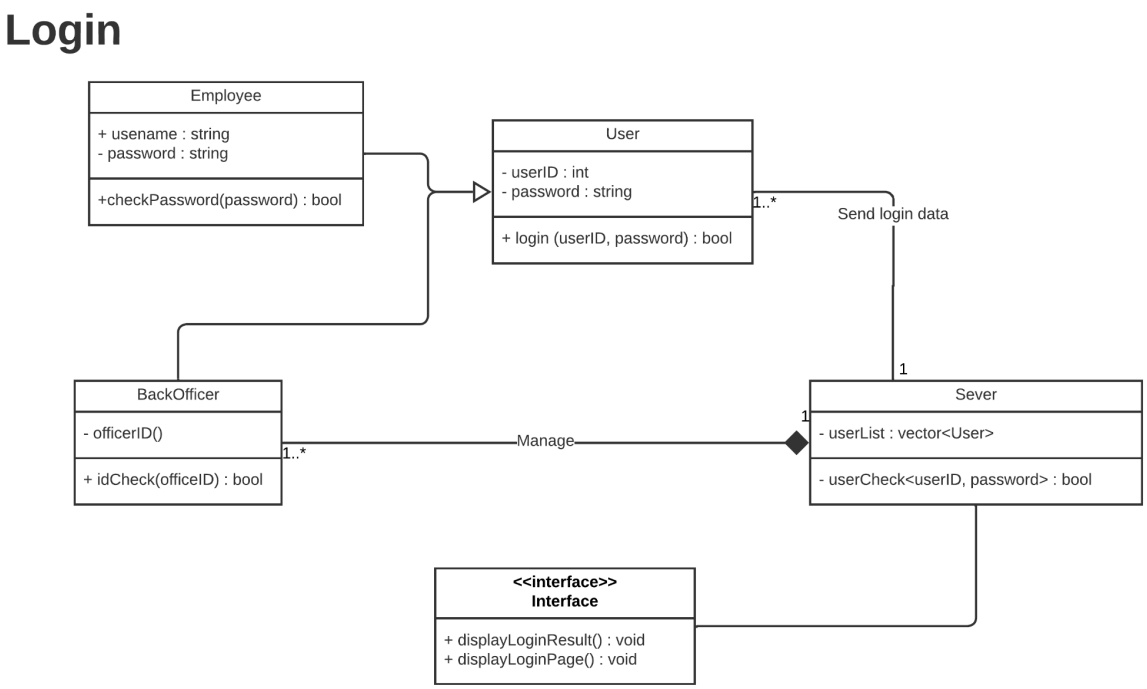
Exceptions	None
Alternate Flows	**Alternate Flow 1: at step 3** 2a. If Back officer does not want to change MCP to other area anymore, then drag the MCP bar back to the original area, or click “Undo” button. ***Continue to step 3 in Normal Flow*** **Alternate Flow 2: at step 5** 5a. If Back officer does not want to verify modification anymore, then click “Cancel” button, system returns to “View MCPs’ detail” screen. ***Continue to step 3 in Normal Flow***

2.2.2 Draw sequence diagram:

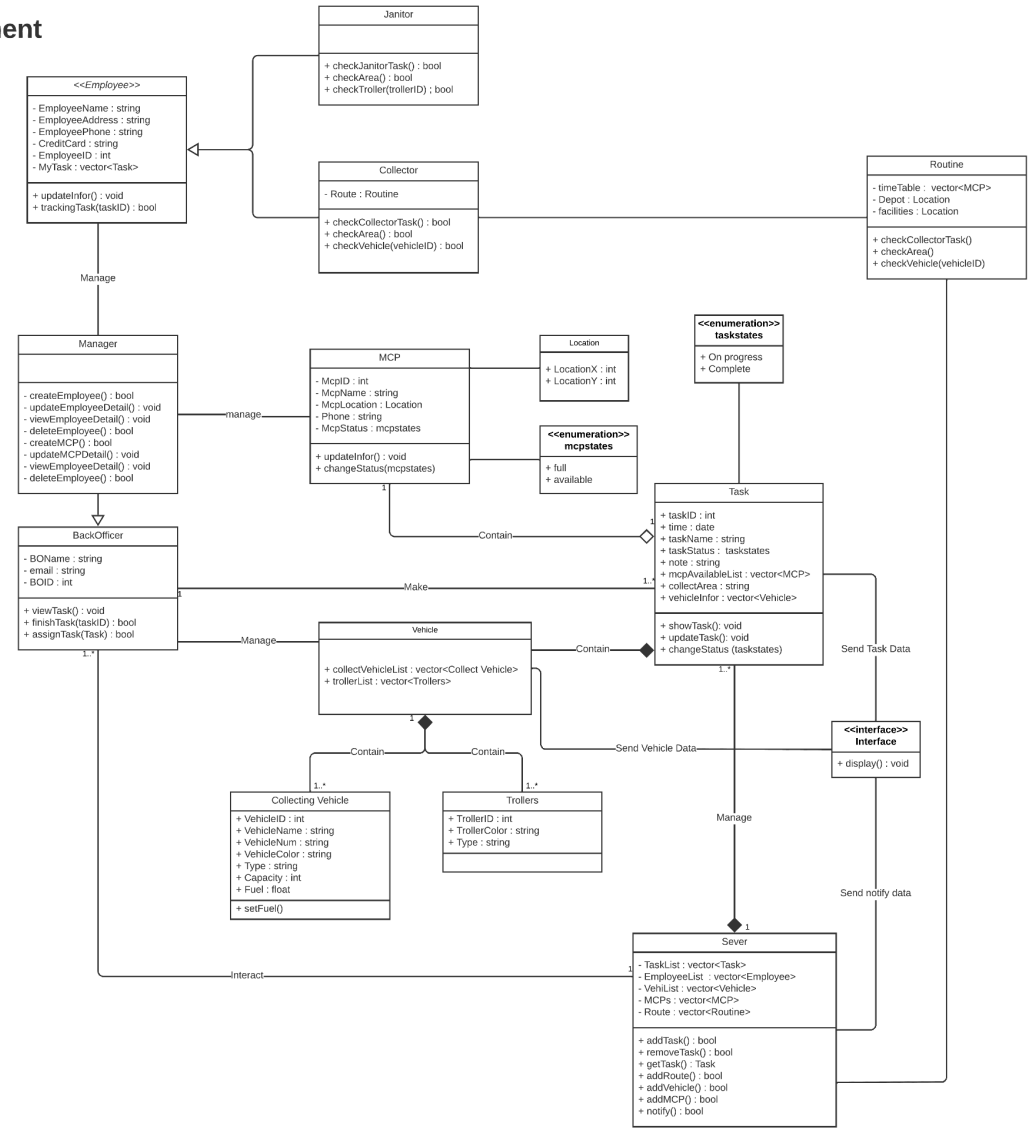

Back Officer
System Interface
UWC 2.0 Database
Route Planner
request view available vehicles()
get available vehicles()
show available vehicles list()
tranfers available vehicles data()
tick boxes()
alter 1
[No vehicle is chosen]
click "Next"()
display message to choose again()
tick boxes()
alter 2
[After tick boxes()]
untick boxes or "untick all"()
click "Next"()
get access to map data()
request map with solution()
show map with coloured areas()
transfer map with coloured areas()
drag vehicles to area to assign()
alter 1
[After drag vehicles to area to assign()]
remove vehicle bar or "unassign all"()
click "save"()
update vehicles status()
update vehicles status success()
return to "Plan Route" screen()
expt 1
[After display "Are you sure to save change"()]
click "Cancel"()
return to "View MCPs' detail" screen()


Back Officer
System Interface
UWC 2.0 Database
Route Planner
request view MCPs detail()
get MCPs detail()
request MCPs with divided areas()
show all MCPs detail with areas()
transfer MCPs map with divided areas()
drag MCPs to different area()
alter 1
[After drag MCPs to different area()]
drag MCPs back or "Undo"()
click "OK"()
display "Are you sure to save change"()
click "Yes"()
update MCPs change()
display "change saved"()
update MCPs change success()
click "OK"()
return to "View MCPs' detail" screen()
expt 1
[After display "Are you sure to save change"()]
click "Cancel"()
return to "View MCPs' detail" screen()

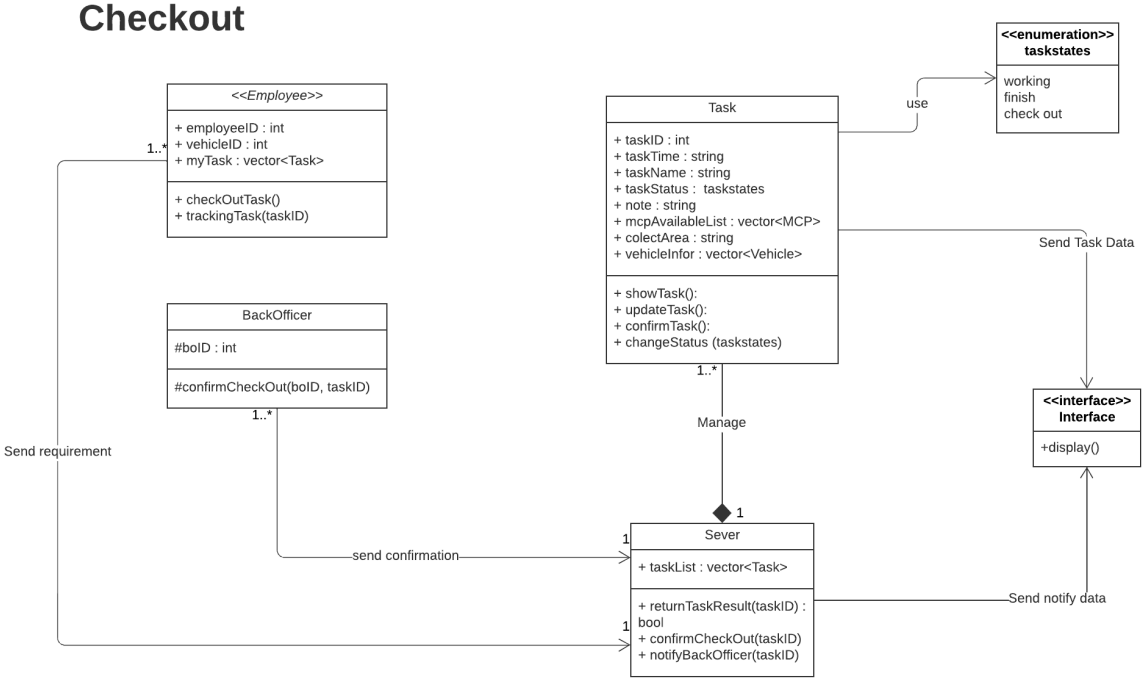
2.3. Draw a class diagram of Task Assignment module as comprehensive as possible



Management


Janitor
+ checkJanitorTask() : bool
+ checkArea() : bool
+ checkTroller(trollerID) ; bool
<<Employee>>
- EmployeeName : string
- EmployeeAddress : string
- EmployeePhone : string
- CreditCard : string
- EmployeeID : int
- MyTask : vector<Task>
+ updateInfor() : void
+ trackingTask(taskID) : bool
Collector
- Route : Routine
+ checkCollectorTask() : bool
+ checkArea() : bool
+ checkVehicle(vehicleID) : bool
Routine
- timeTable : vector<MCP>
- Depot : Location
- facilities : Location
+ checkCollectorTask()
+ checkArea()
+ checkVehicle(vehicleID)
Manage
Manager
- createEmployee() : bool
- updateEmployeeDetail() : void
- viewEmployeeDetail() : void
- deleteEmployee() : bool
- createMCP() : bool
- updateMCPDetail() : void
- viewEmployeeDetail() : void
- deleteEmployee() : bool
manage
MCP
- McpID : int
- McpName : string
- McpLocation : Location
- Phone : string
- McpStatus : mcpstates
+ updateInfor() : void
+ changeStatus(mcpstates)
Location
+ LocationX : int
+ LocationY : int
<<enumeration>>
mcpstates
+ full
+ available
<<enumeration>>
taskstates
+ On progress
+ Complete
Task
+ taskID : int
+ time : date
+ taskName : string
+ taskStatus : taskstates
+ note : string
+ mcpAvailableList : vector<MCP>
+ collectArea : string
+ vehicleInfor : vector<Vehicle>
+ showTask(): void
+ updateTask(): void
+ changeStatus (taskstates)
Contain
BackOfficer
- BOName : string
- email : string
- BOID : int
+ viewTask() : void
+ finishTask(taskID) : bool
+ assignTask(Task) : bool
Make
Manage
Vehicle
+ collectVehicleList : vector<Collect Vehicle>
+ trollerList : vector<Trollers>
Contain
Send Task Data
Contain
Contain
Send Vehicle Data
<<interface>>
Interface
+ display() : void
Collecting Vehicle
+ VehicleID : int
+ VehicleName : string
+ VehicleNum : string
+ VehicleColor : string
+ Type : string
+ Capacity : int
+ Fuel : float
+ setFuel()
Trollers
+ TrollerID : int
+ TrollerColor : string
+ Type : string
Manage
Send notify data
Interact
Sever
- TaskList : vector<Task>
- EmployeeList : vector<Employee>
- VehiList : vector<Vehicle>
- MCPs : vector<MCP>
- Route : vector<Routine>
+ addTask() : bool
+ removeTask() : bool
+ getTask() : Task
+ addRoute() : bool
+ addVehicle() : bool
+ addMCP() : bool
+ notify() : bool

Checkout


<<enumeration>>
taskstates
working
finish
check out
use
<<Employee>>
+ employeeID : int
+ vehicleID : int
+ myTask : vector<Task>
+ checkOutTask()
+ trackingTask(taskID)
Task
+ taskID : int
+ taskTime : string
+ taskName : string
+ taskStatus : taskstates
+ note : string
+ mcpAvailableList : vector<MCP>
+ colectArea : string
+ vehicleInfor : vector<Vehicle>
+ showTask():
+ updateTask():
+ confirmTask():
+ changeStatus (taskstates)
Send Task Data
1..*
BackOfficer
#boID : int
#confirmCheckOut(boID, taskID)
1..*
Manage
<<interface>>
Interface
+display()
Send requirement
send confirmation
1
Sever
+ taskList : vector<Task>
+ returnTaskResult(taskID) : bool
+ confirmCheckOut(taskID)
+ notifyBackOfficer(taskID)
Send notify data