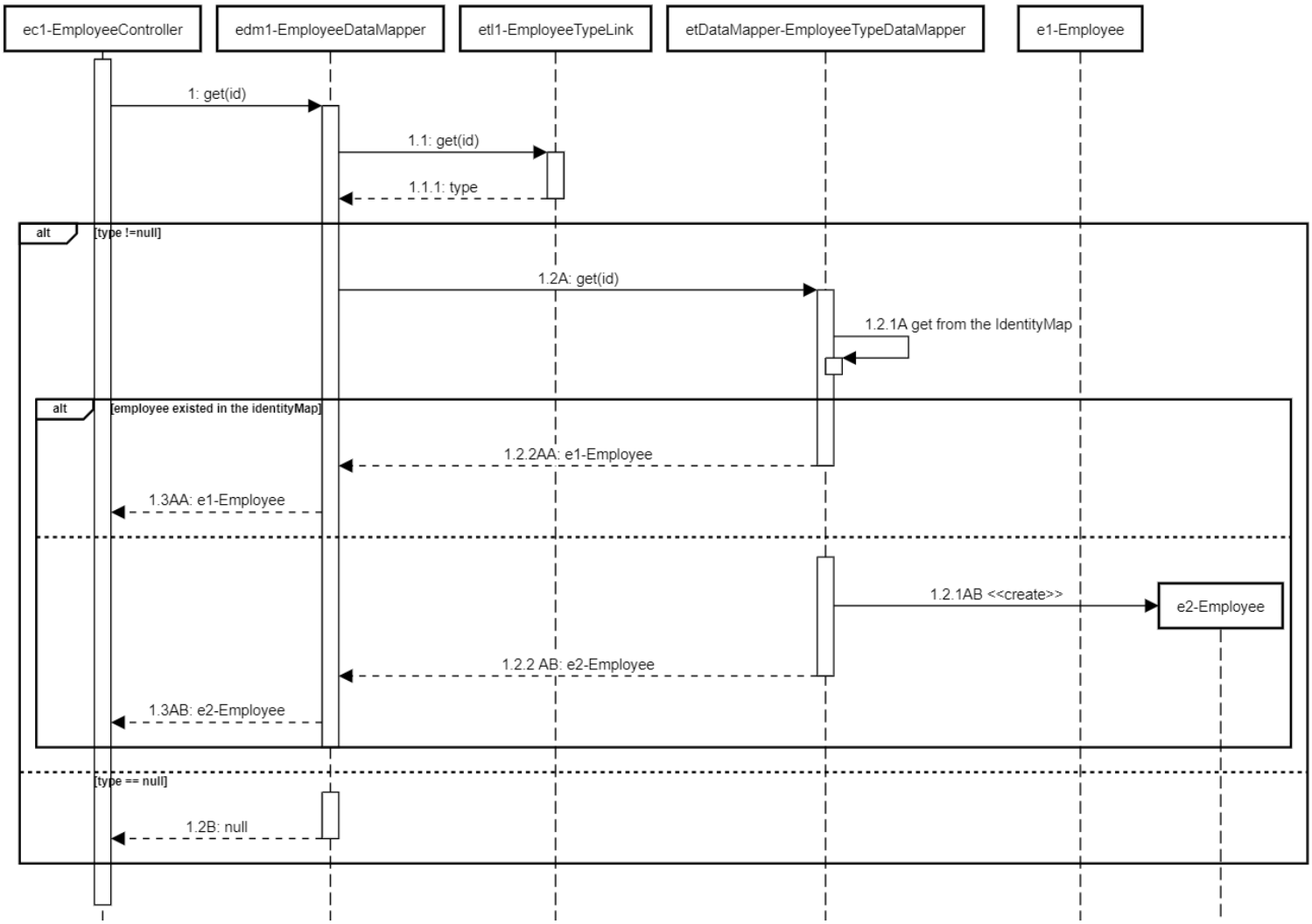
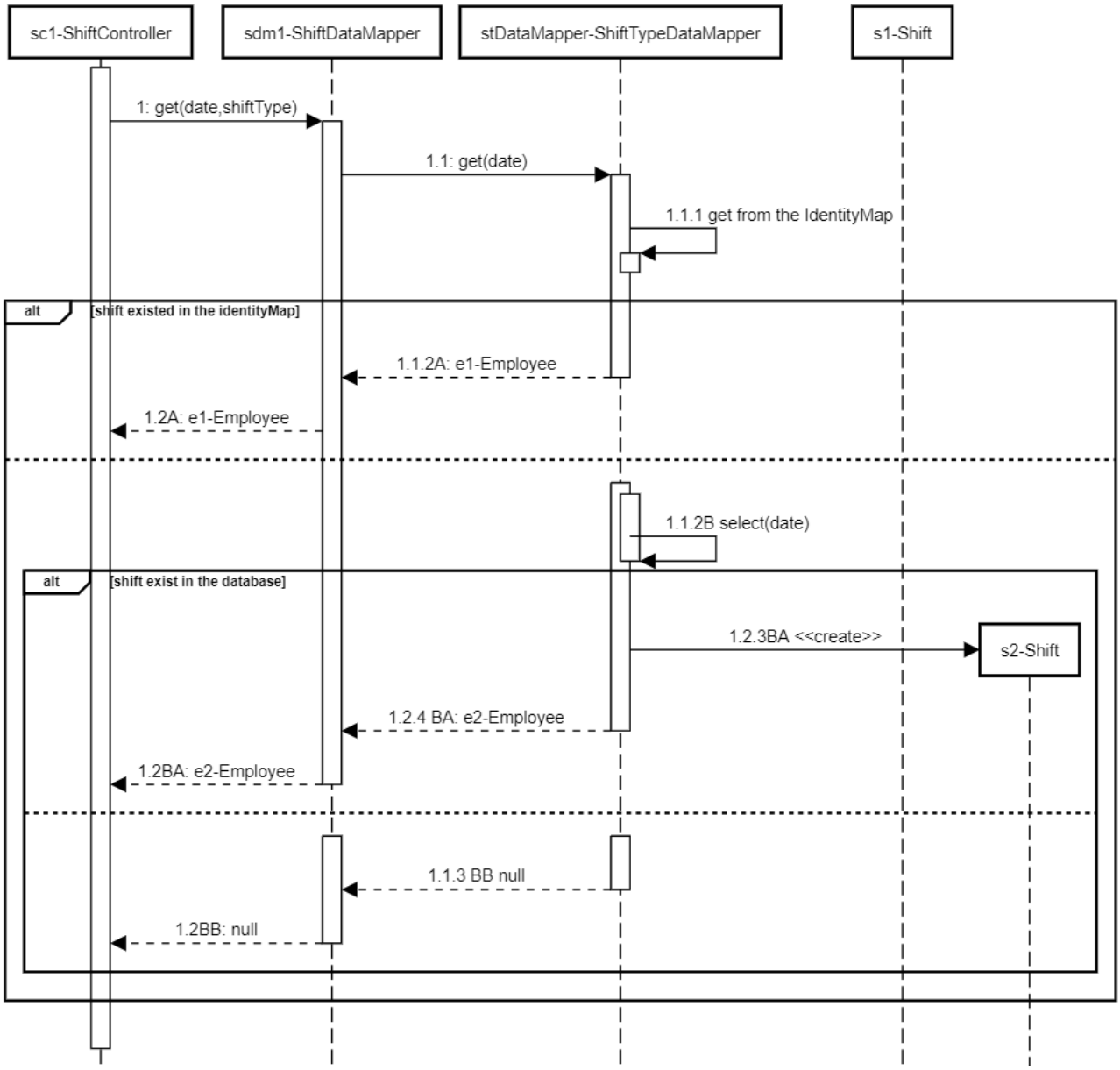
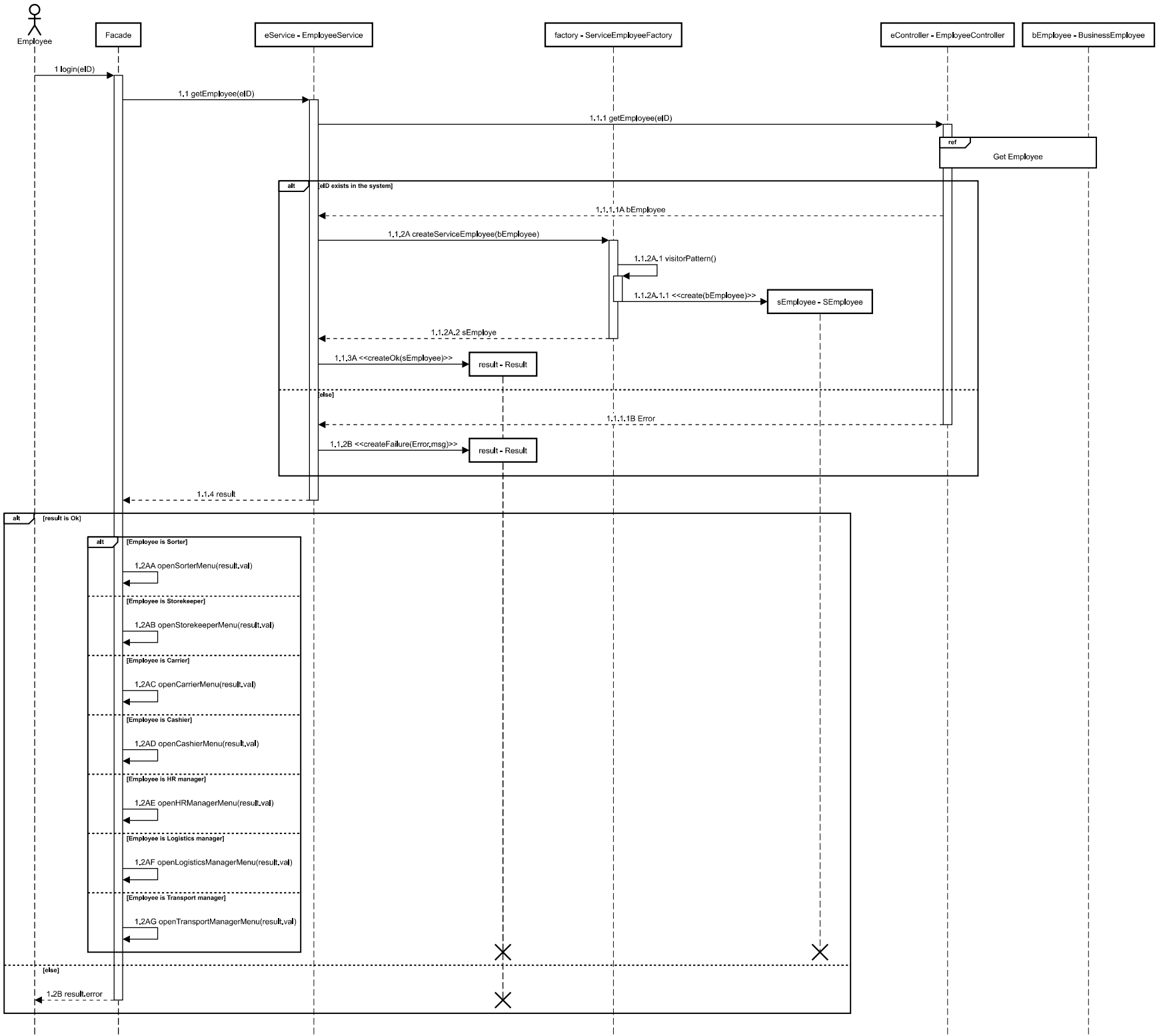
Get Employee



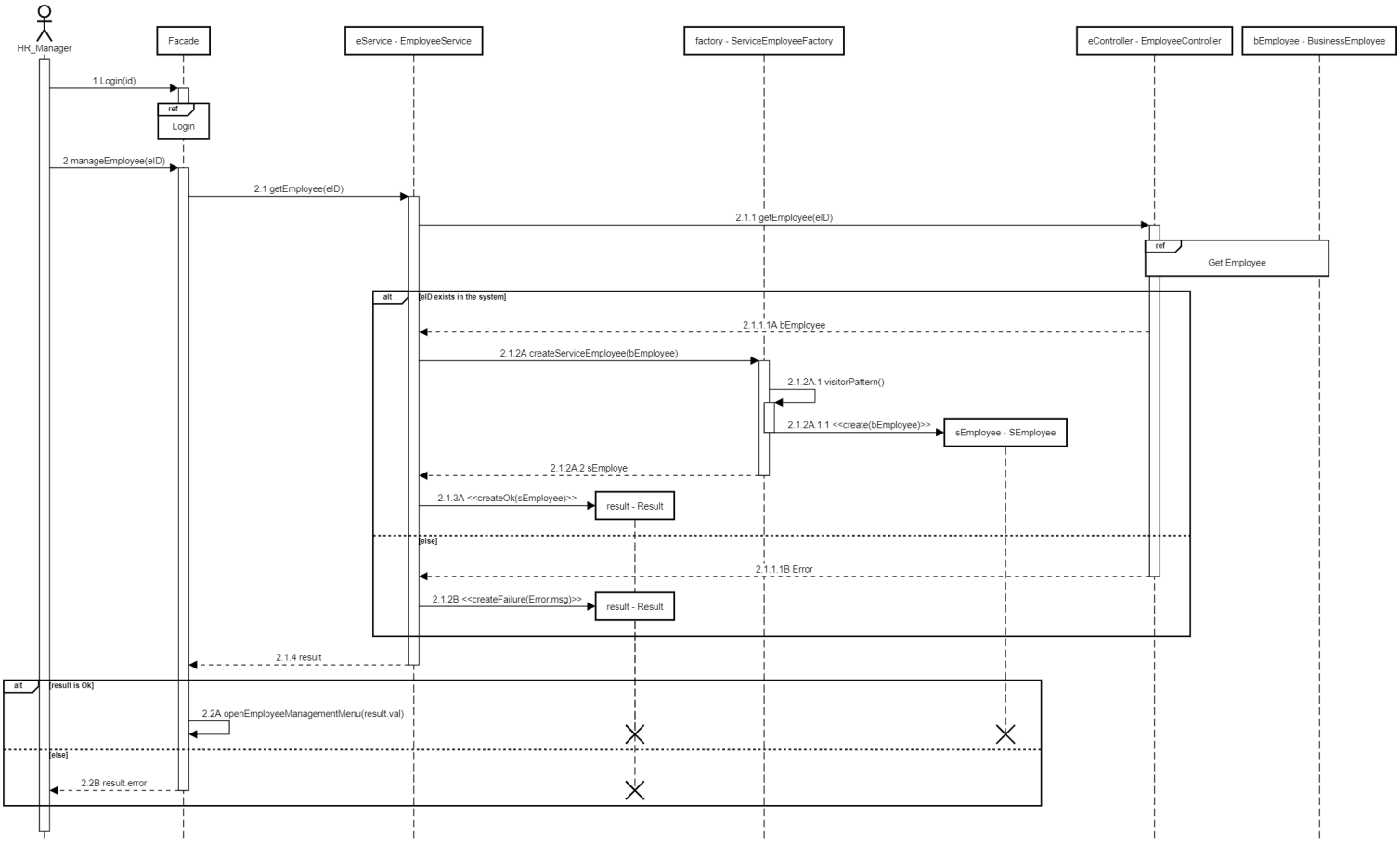
Get Shift


sc1-ShiftController
sdm1-ShiftDataMapper
stDataMapper-ShiftTypeDataMapper
s1-Shift
1: get(date,shiftType)
1.1: get(date)
1.1.1 get from the IdentityMap
alt
[shift existed in the identityMap]
1.1.2A: e1-Employee
1.2A: e1-Employee
1.1.2B select(date)
alt
[shift exist in the database]
1.2.3BA <<create>>
s2-Shift
1.2.4 BA: e2-Employee
1.2BA: e2-Employee
1.1.3 BB null
1.2BB: null

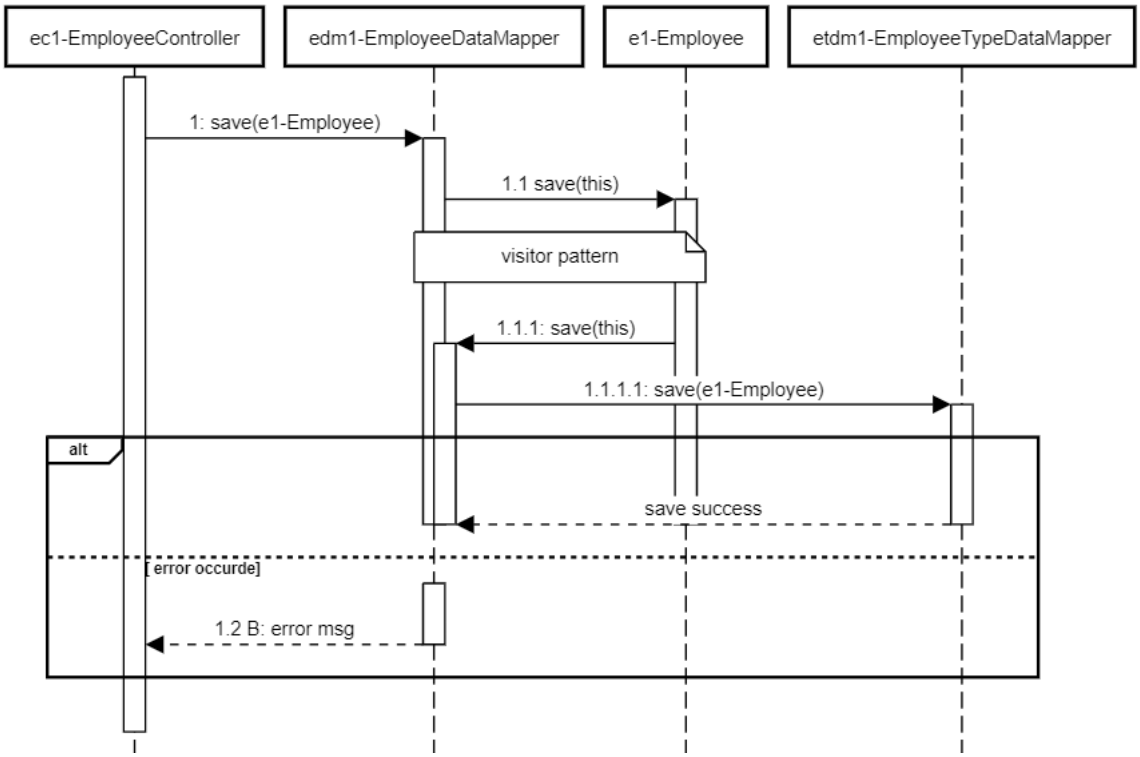
Login


Employee
Facade
eService - EmployeeService
factory - ServiceEmployeeFactory
eController - EmployeeController
bEmployee - BusinessEmployee
1 login(eID)
1.1 getEmployee(eID)
1.1.1 getEmployee(eID)
ref
Get Employee
alt
[eID exists in the system]
1.1.1.1A bEmployee
1.1.2A createServiceEmployee(bEmployee)
1.1.2A.1 visitorPattern()
1.1.2A.1.1 <<create(bEmployee)>>
sEmployee - SEmployee
1.1.2A.2 sEmploye
1.1.3A <<createOk(sEmployee)>>
result - Result
[else]
1.1.1.1B Error
1.1.2B <<createFailure(Error,msg)>>
result - Result
1.1.4 result
alt
[result is Ok]
alt
[Employee is Sorter]
1.2AA openSorterMenu(result.val)
[Employee is Storekeeper]
1.2AB openStorekeeperMenu(result.val)
[Employee is Carrier]
1.2AC openCarrierMenu(result.val)
[Employee is Cashier]
1.2AD openCashierMenu(result.val)
[Employee is HR manager]
1.2AE openHRManagerMenu(result.val)
[Employee is Logistics manager]
1.2AF openLogisticsManagerMenu(result.val)
[Employee is Transport manager]
1.2AG openTransportManagerMenu(result.val)
[else]
1.2B result.error

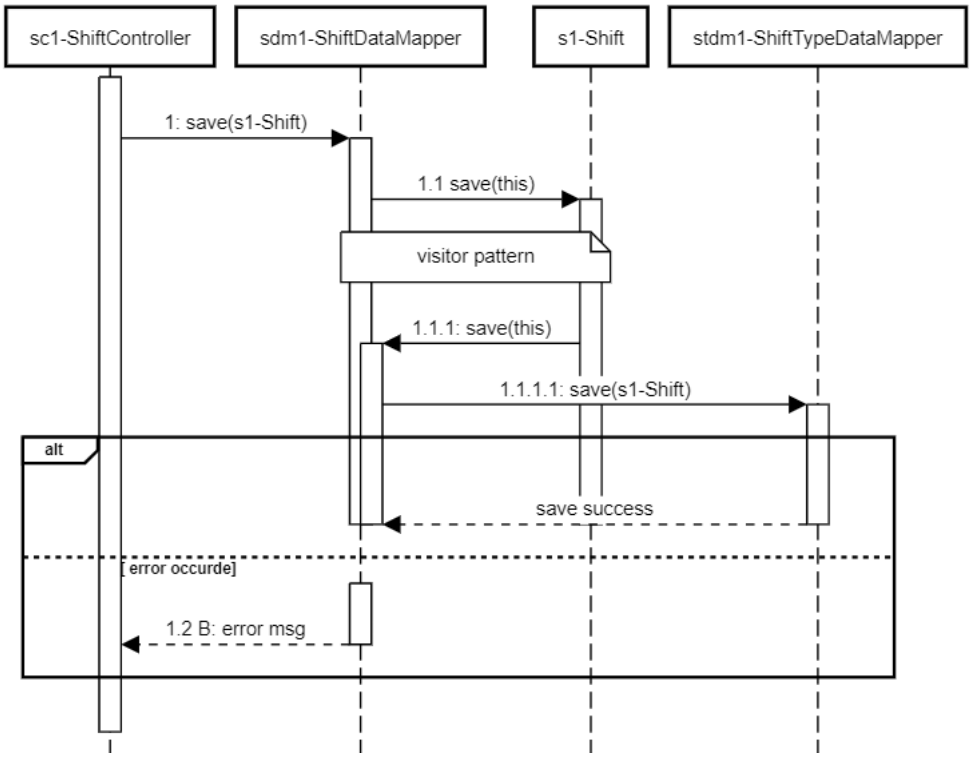
Opening Employee Management Menu



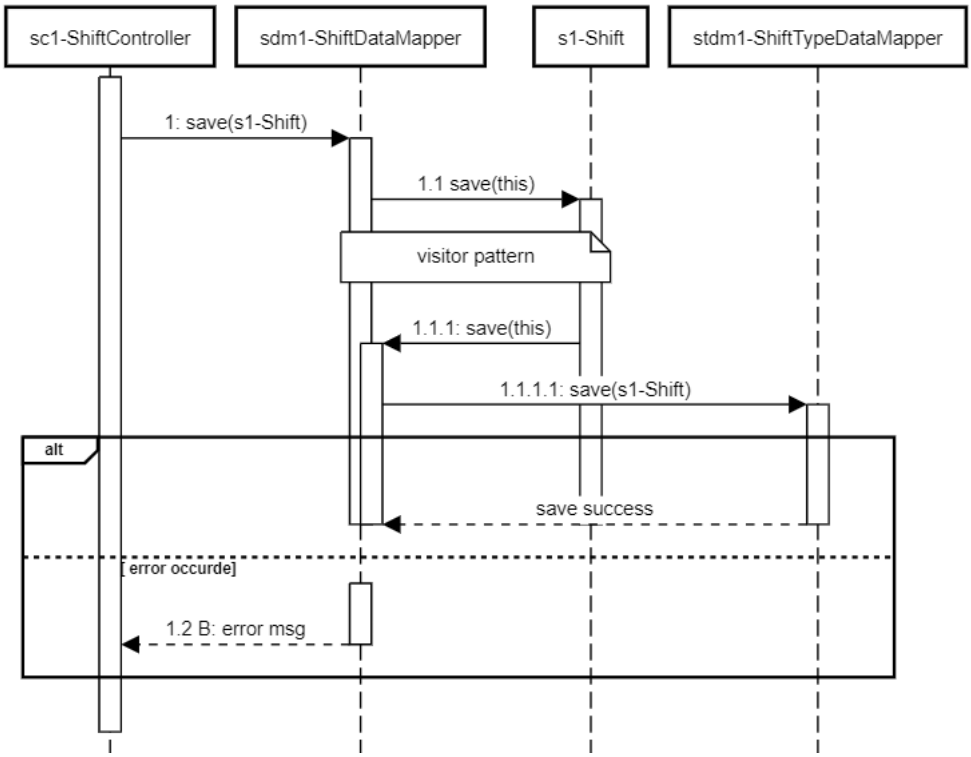
Save Employee In DB

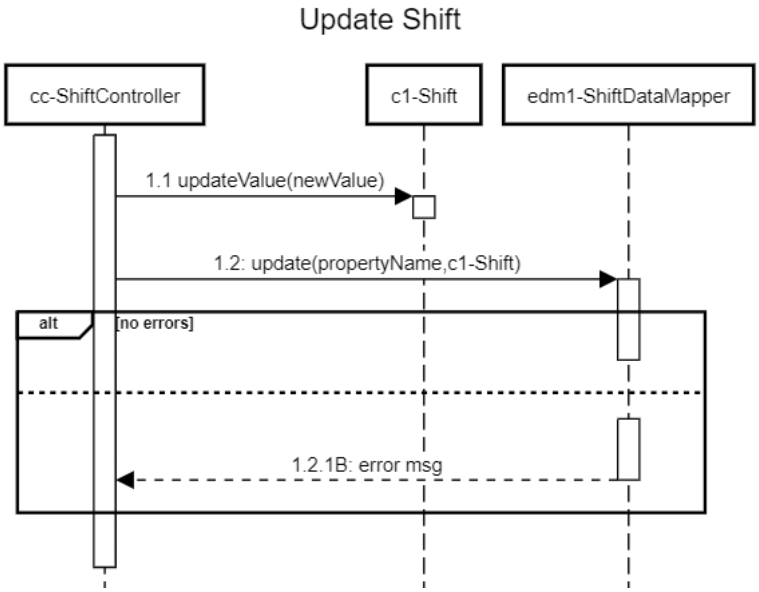


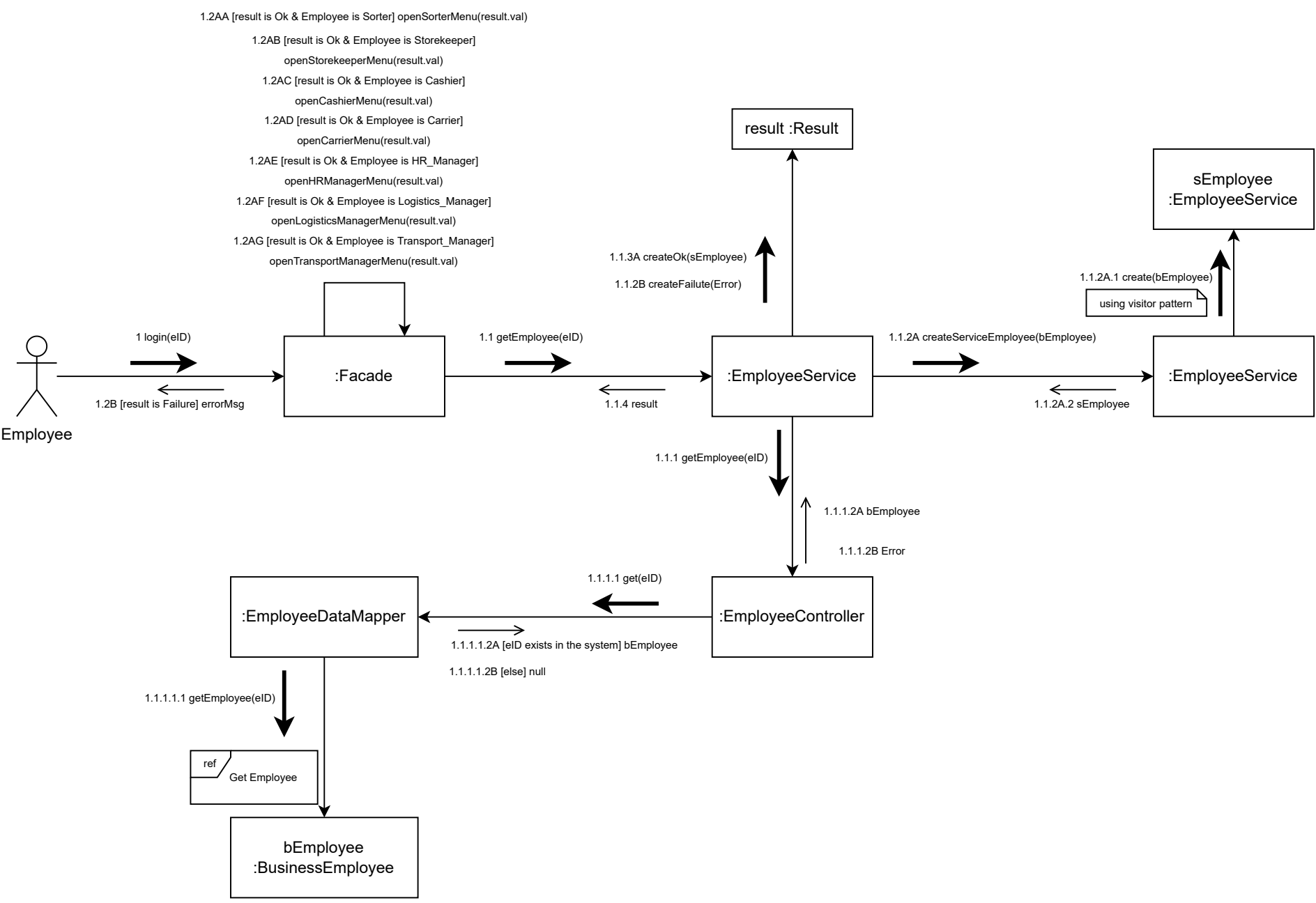
Save shift In DB

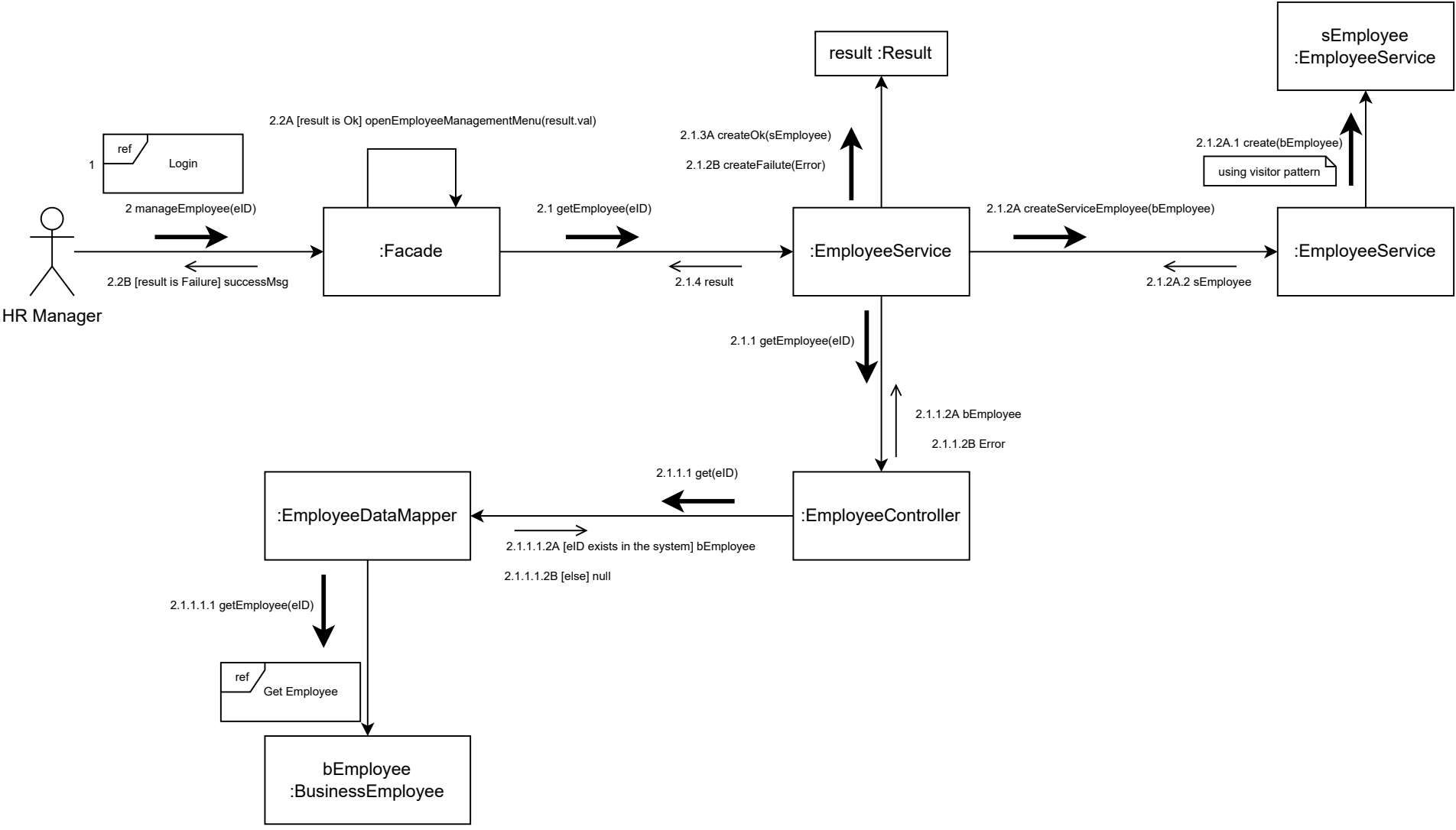


Save shift In DB




Update Shift
cc-ShiftController
c1-Shift
edm1-ShiftDataMapper
1.1 updateValue(newValue)
1.2: update(propertyName,c1-Shift)
alt
[no errors]
1.2.1B: error msg


1.2AA [result is Ok & Employee is Sorter] openSorterMenu(result.val)
1.2AB [result is Ok & Employee is Storekeeper]
openStorekeeperMenu(result.val)
1.2AC [result is Ok & Employee is Cashier]
openCashierMenu(result.val)
1.2AD [result is Ok & Employee is Carrier]
openCarrierMenu(result.val)
1.2AE [result is Ok & Employee is HR_Manager]
openHRManagerMenu(result.val)
1.2AF [result is Ok & Employee is Logistics_Manager]
openLogisticsManagerMenu(result.val)
1.2AG [result is Ok & Employee is Transport_Manager]
openTransportManagerMenu(result.val)
result :Result
sEmployee
:EmployeeService
1.1.3A createOk(sEmployee)
1.1.2B createFailute(Error)
1.1.2A.1 create(bEmployee)
using visitor pattern
1 login(eID)
1.2B [result is Failure] errorMsg
:Facade
1.1 getEmployee(eID)
1.1.4 result
:EmployeeService
1.1.2A createServiceEmployee(bEmployee)
1.1.2A.2 sEmployee
:EmployeeService
Employee
1.1.1 getEmployee(eID)
1.1.1.2A bEmployee
1.1.1.2B Error
1.1.1.1 get(eID)
:EmployeeDataMapper
:EmployeeController
1.1.1.1.2A [eID exists in the system] bEmployee
1.1.1.1.2B [else] null
1.1.1.1.1 getEmployee(eID)
ref
Get Employee
bEmployee
:BusinessEmployee


HR Manager
1
ref
Login
2 manageEmployee(eID)
2.2B [result is Failure] successMsg
2.2A [result is Ok] openEmployeeManagementMenu(result.val)
:Facade
2.1 getEmployee(eID)
2.1.4 result
result :Result
2.1.3A createOk(sEmployee)
2.1.2B createFailute(Error)
:EmployeeService
2.1.2A createServiceEmployee(bEmployee)
2.1.2A.2 sEmployee
:EmployeeService
sEmployee
:EmployeeService
2.1.2A.1 create(bEmployee)
using visitor pattern
2.1.1 getEmployee(eID)
2.1.1.2A bEmployee
2.1.1.2B Error
2.1.1.1 get(eID)
:EmployeeController
:EmployeeDataMapper
2.1.1.1.2A [eID exists in the system] bEmployee
2.1.1.1.2B [else] null
2.1.1.1.1 getEmployee(eID)
ref
Get Employee
bEmployee
:BusinessEmployee