Team-C

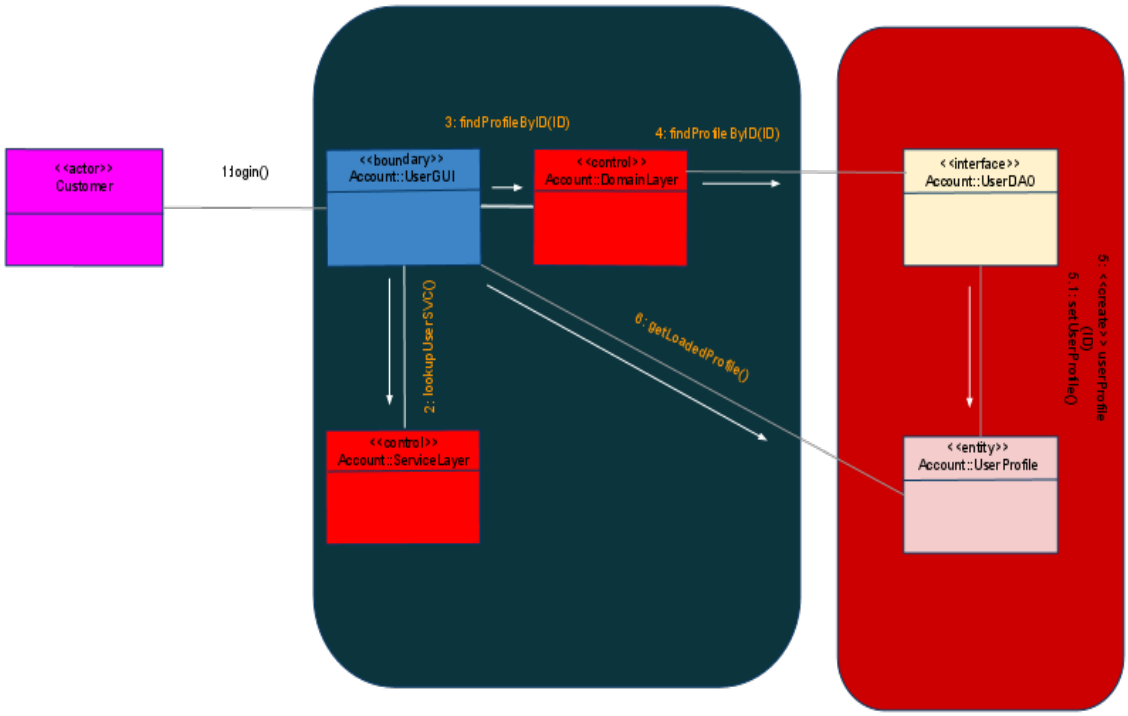
Movies R US

Use Case Realizations

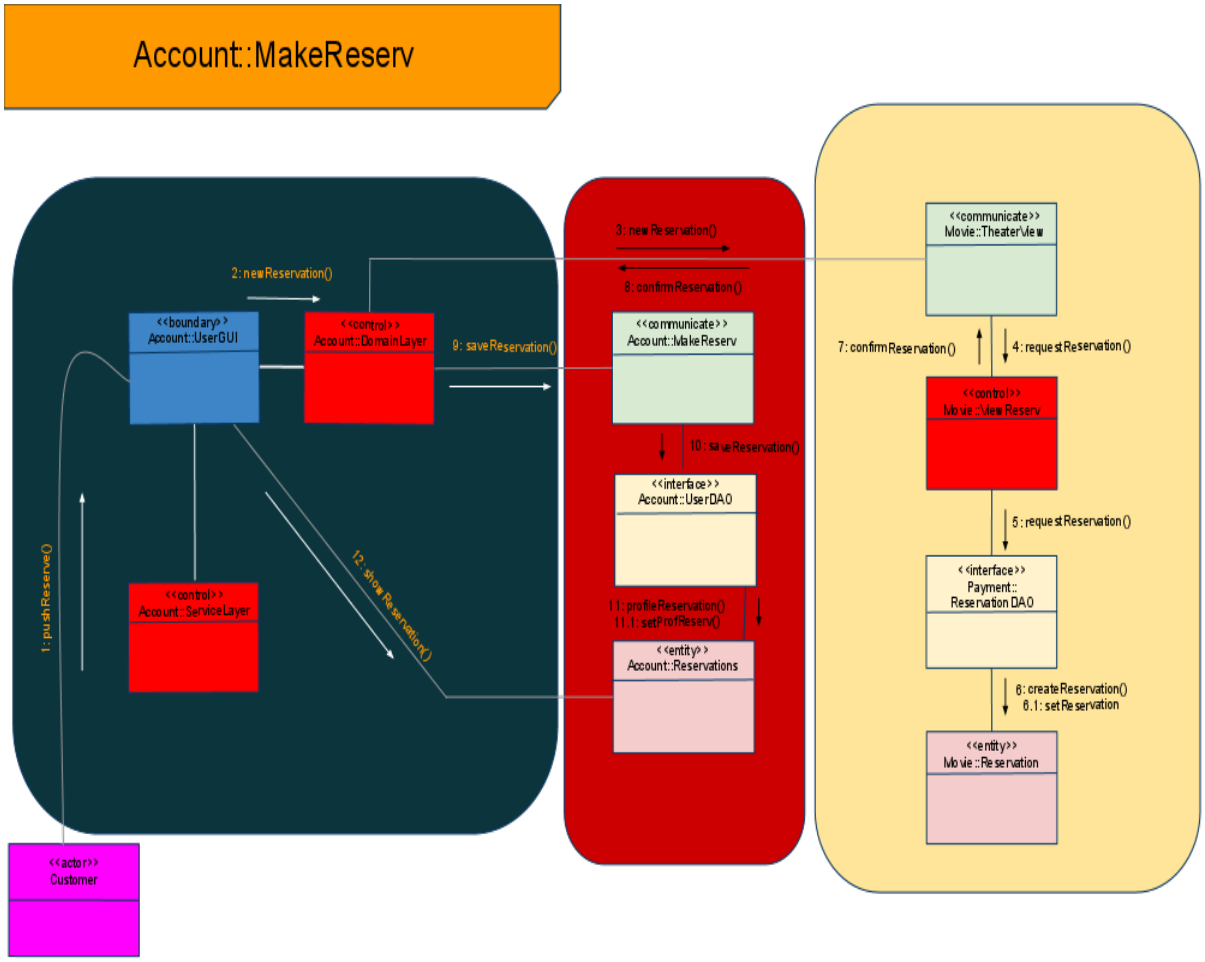
Introduction: Movies R US is broken up into four Objects: Account, Movie, Payment, and Report. These Objects all contain a variety of packages containing Realized Use Case that are described below. Full sizes images may be downloaded from the Team-C forum or viewed here in Google Docs. The SRS which accompanies these USE Cases may be found here: SRS. This document may also be retrieved for download in a variety of formats from here: Realizations.

Object Account handles all actions concerning the user profile. Therefore anytime the Employee or the Customer uses the web application, kiosk or web based application at the register they are loading and utilizing their profile through Account.

Account::UserGUI

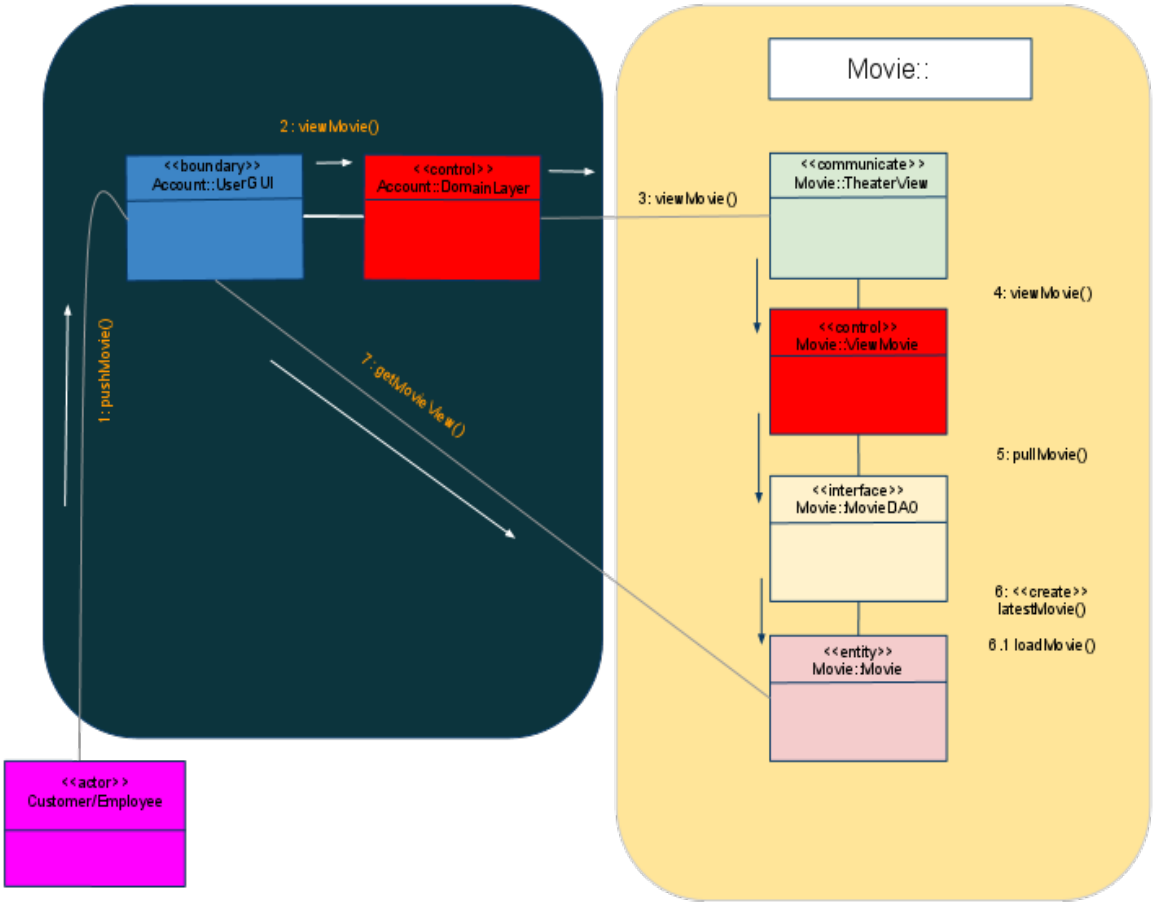


ID:1


Account::MakeReserv
<<boundary>>
Account::UserGUI
<<control>>
Account::DomainLayer
<<control>>
Account::ServiceLayer
<<actor>>
Customer
<<communicate>>
Account::MakeReserv
<<interface>>
Account::UserDAO
<<entity>>
Account::Reservations
<<communicate>>
Movie::Theater\View
<<control>>
Movie::View Reserv
<<interface>>
Payment::
Reservation DAO
<<entity>>
Movie::Reservation
1: pushReserve()
2: newReservation()
3: new Reservation()
4: request Reservation()
5: request Reservation()
6: createReservation()
6.1: setReservation
7: confirmReservation()
8: confirmReservation()
9: saveReservation()
10: saveReservation()
11: profileReservation()
11.1: setProfReserv()
12: showReservation()

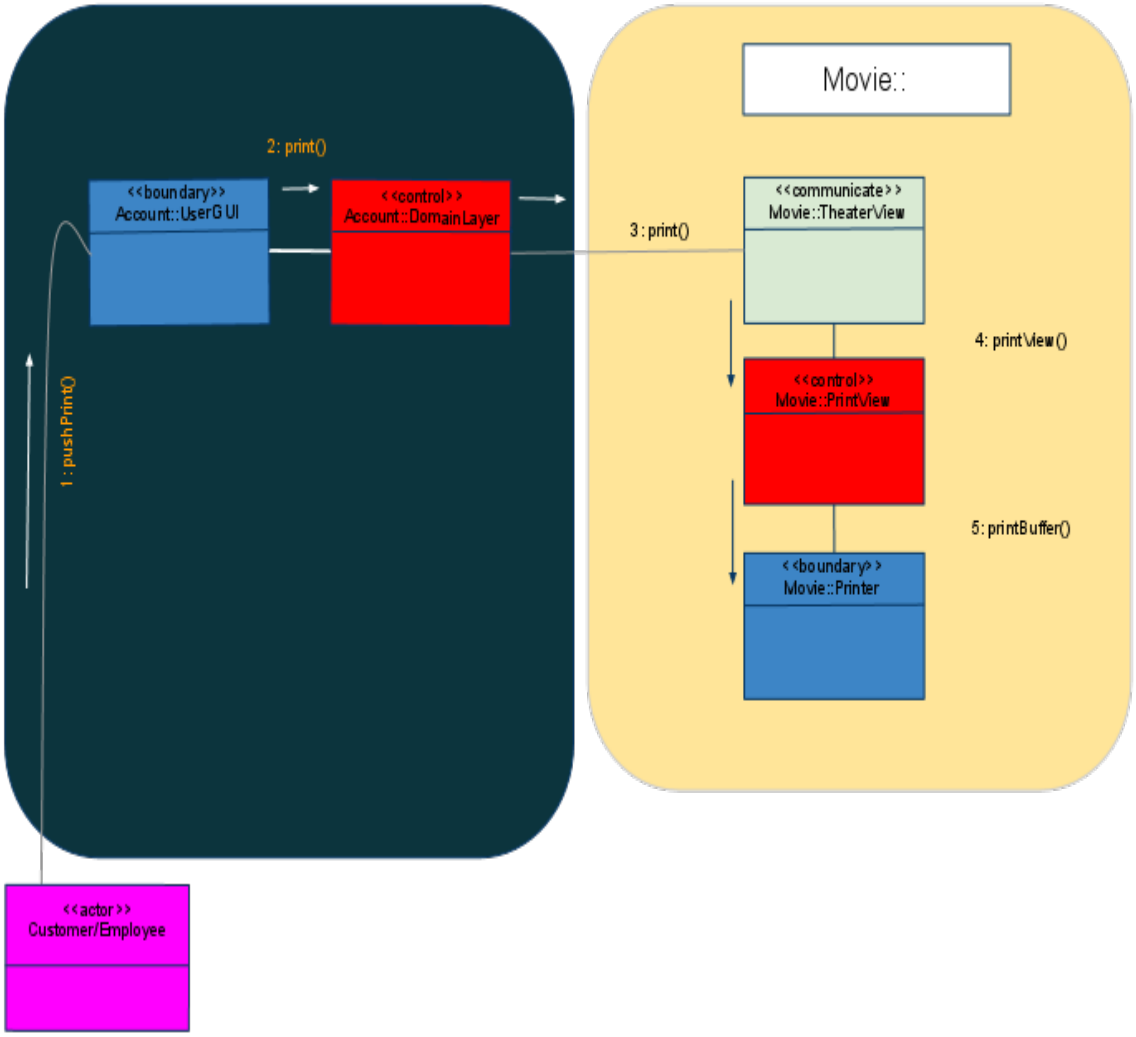
ID:2

Object Movie contains packages for both Customer and Employee to retrieve items such as tickets and gift cards. This is only for retrieval of said items not creation of tickets and gift cards but these packages handle requesting views of these items. A package is also included for printing views after loading. The Movies and Schedule of movies may be viewed here by the Customer while the Employee may view and edit both. To generalize Object Movie handles the most common viewing and editing of views for both the Customer and Employee. Each Use Case has a very short and specific Use to keep package simple.



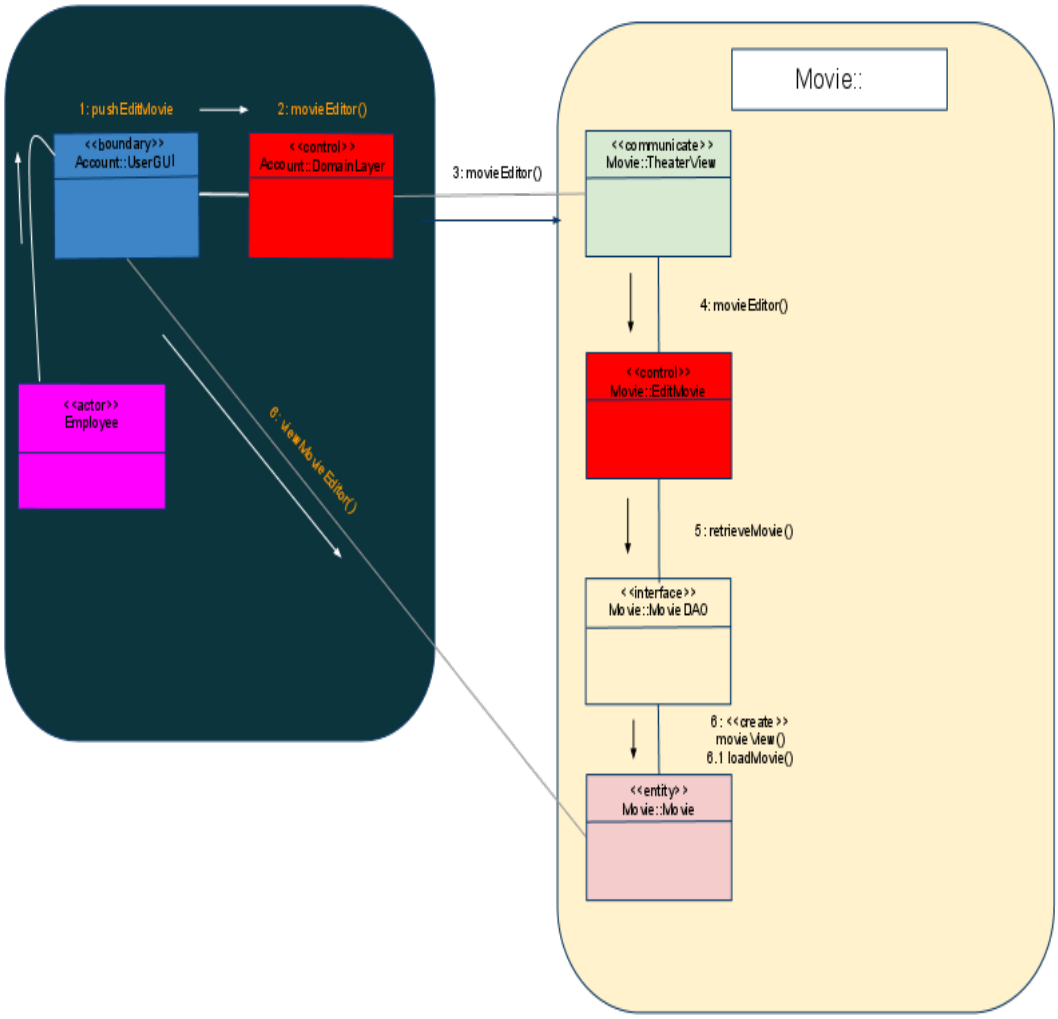
ID:3

Movie::PrintView



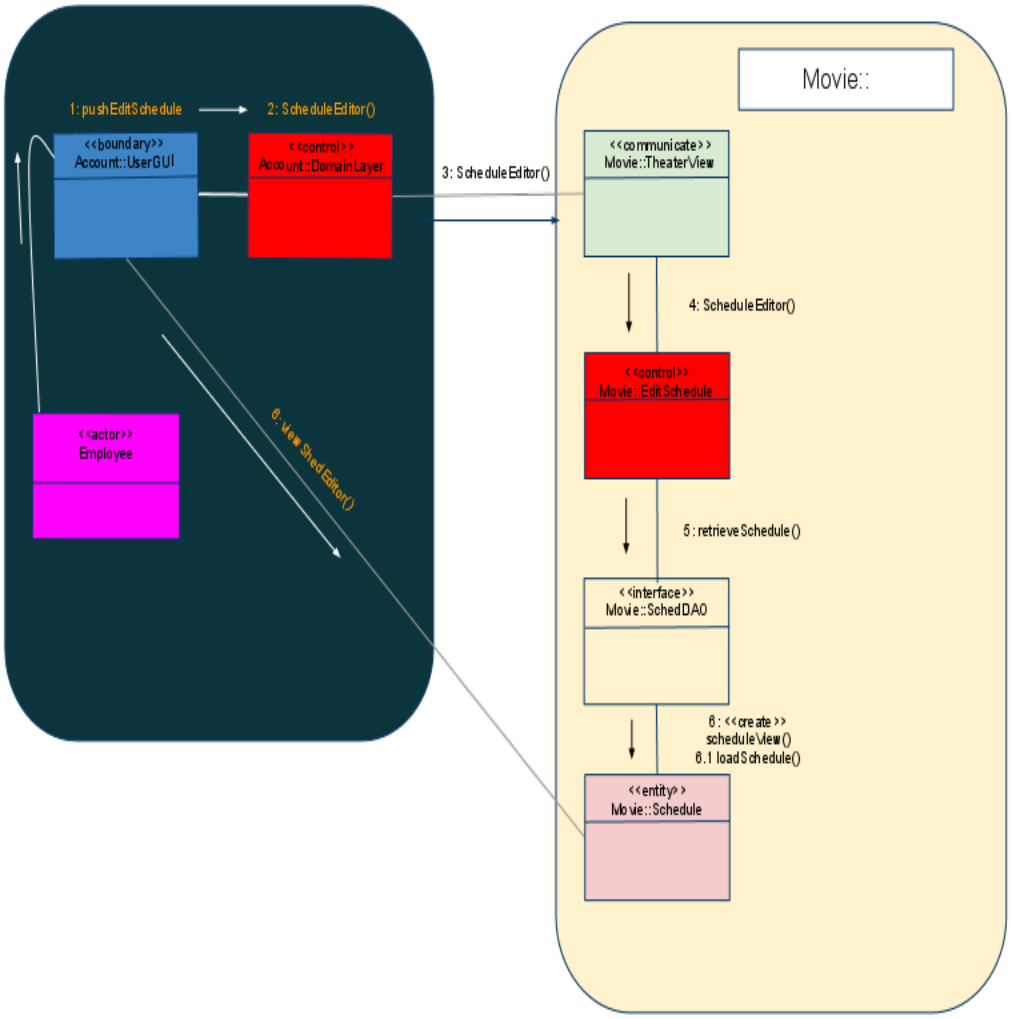
ID:4

Movie::EditMovie



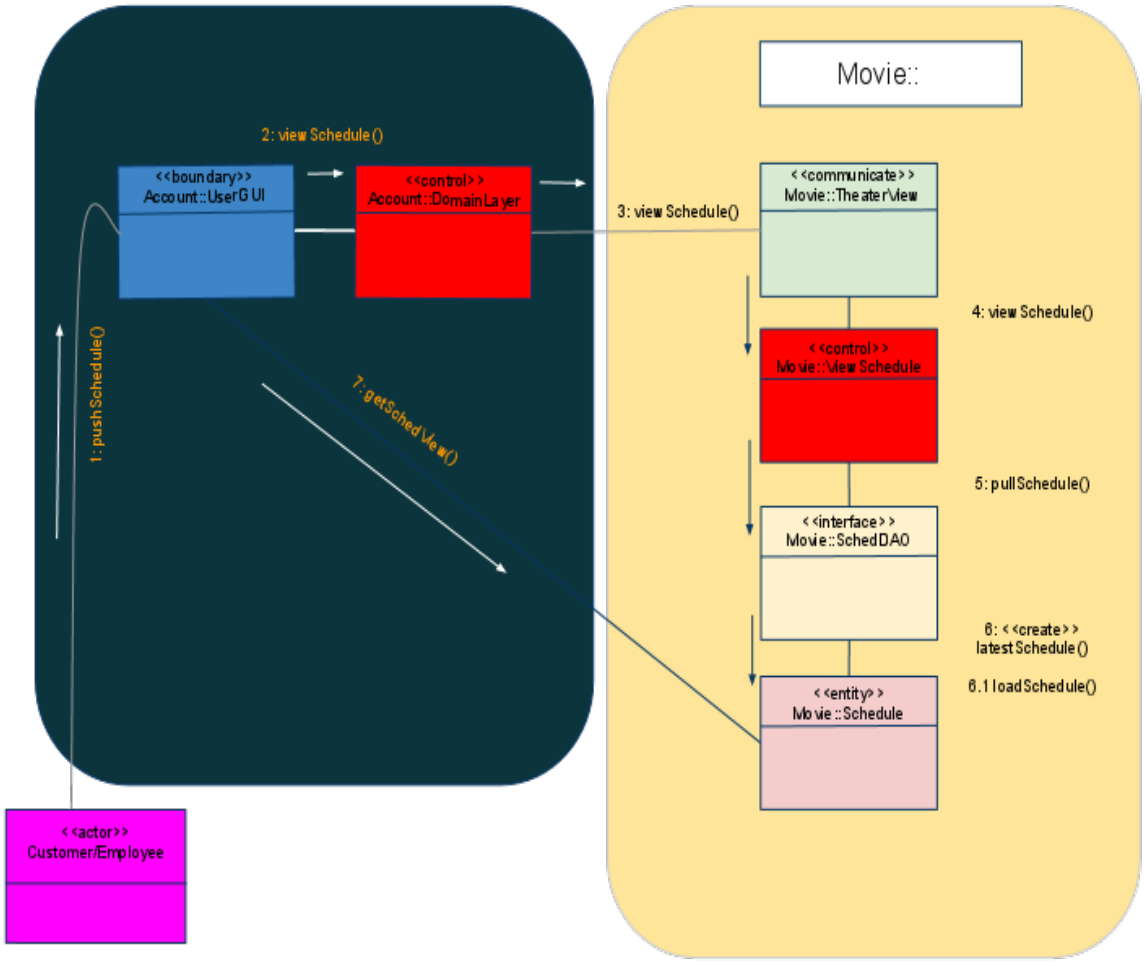
ID:5

Movie::EditSchedule



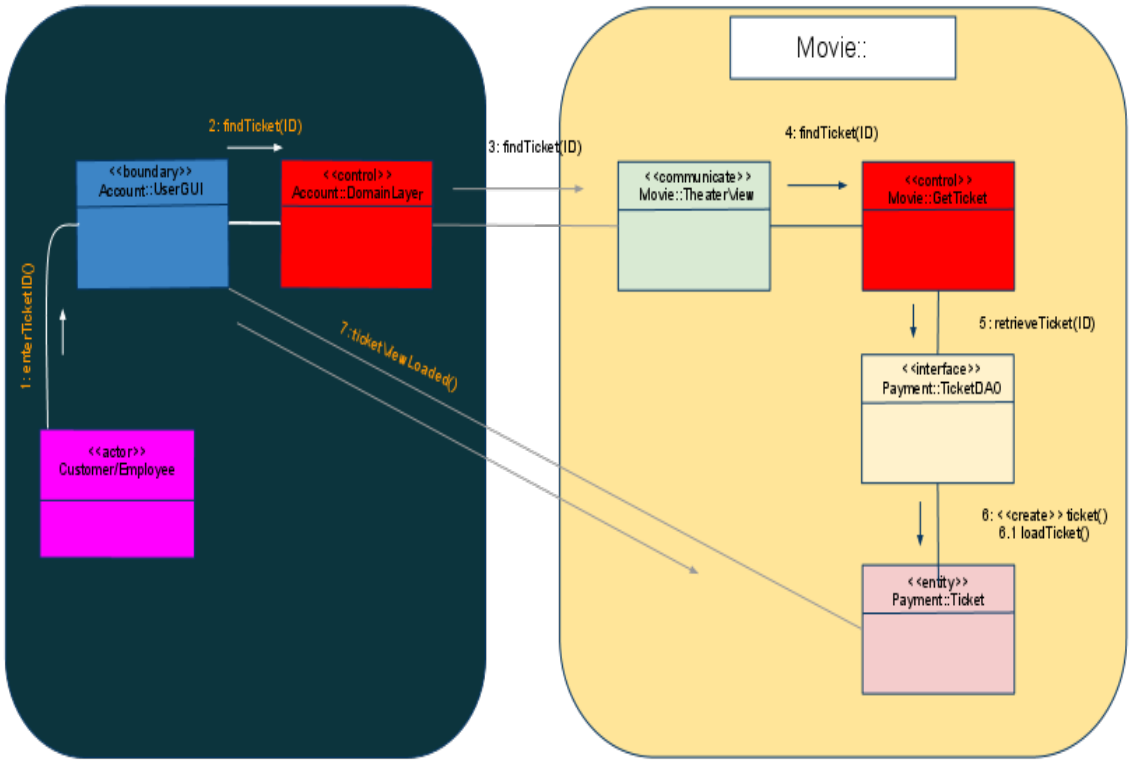
ID:6

Movie::ViewSchedule



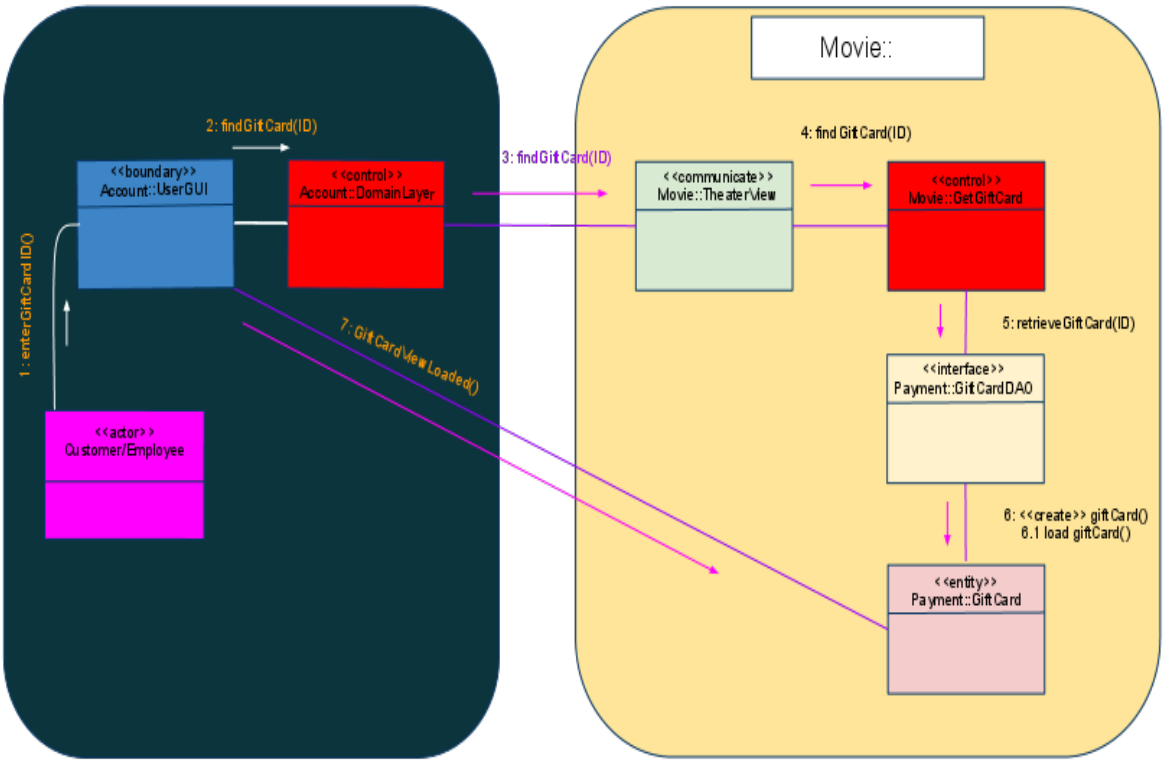
ID:7

Movie::GetTicket



ID:8

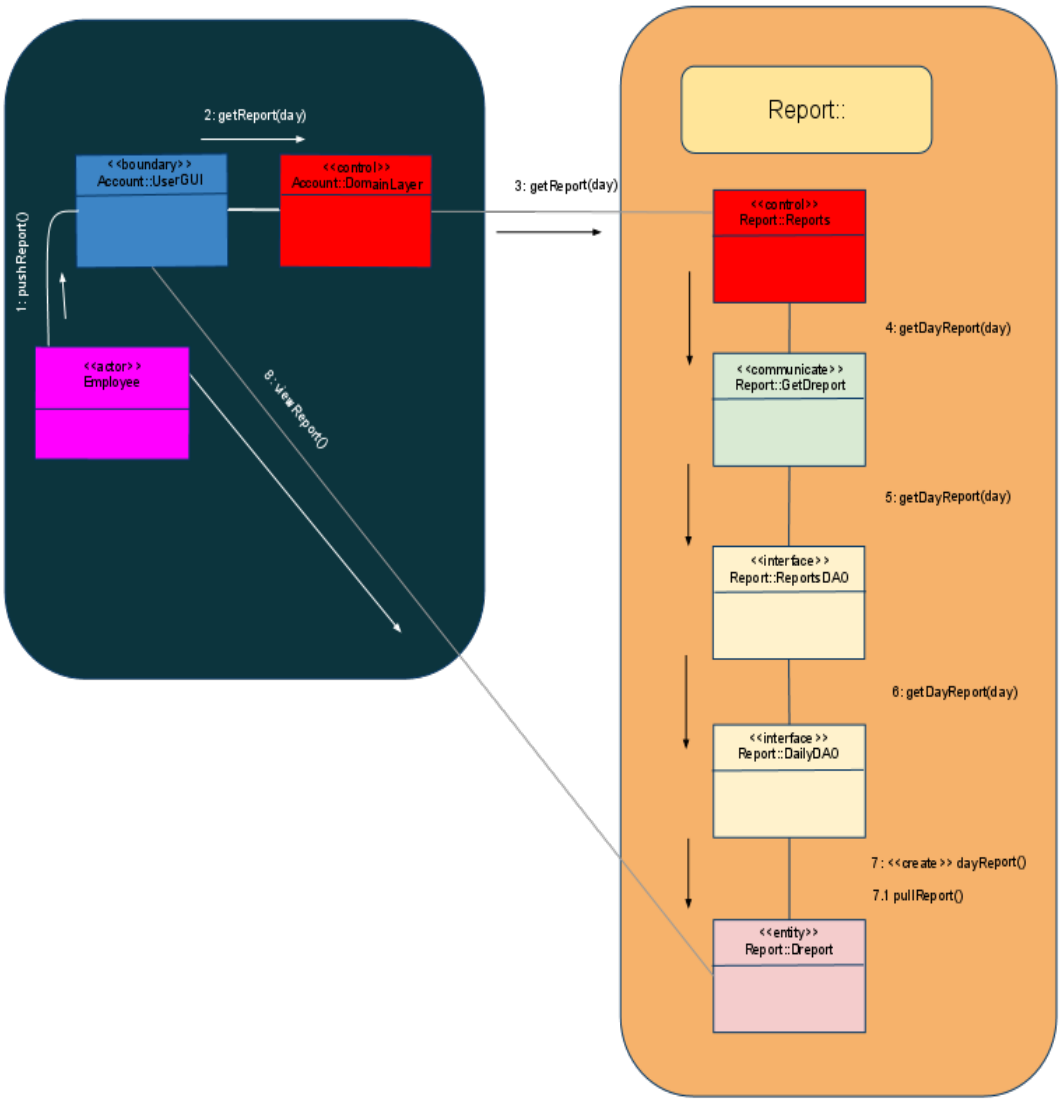
Movie::GetGiftCard



ID:9

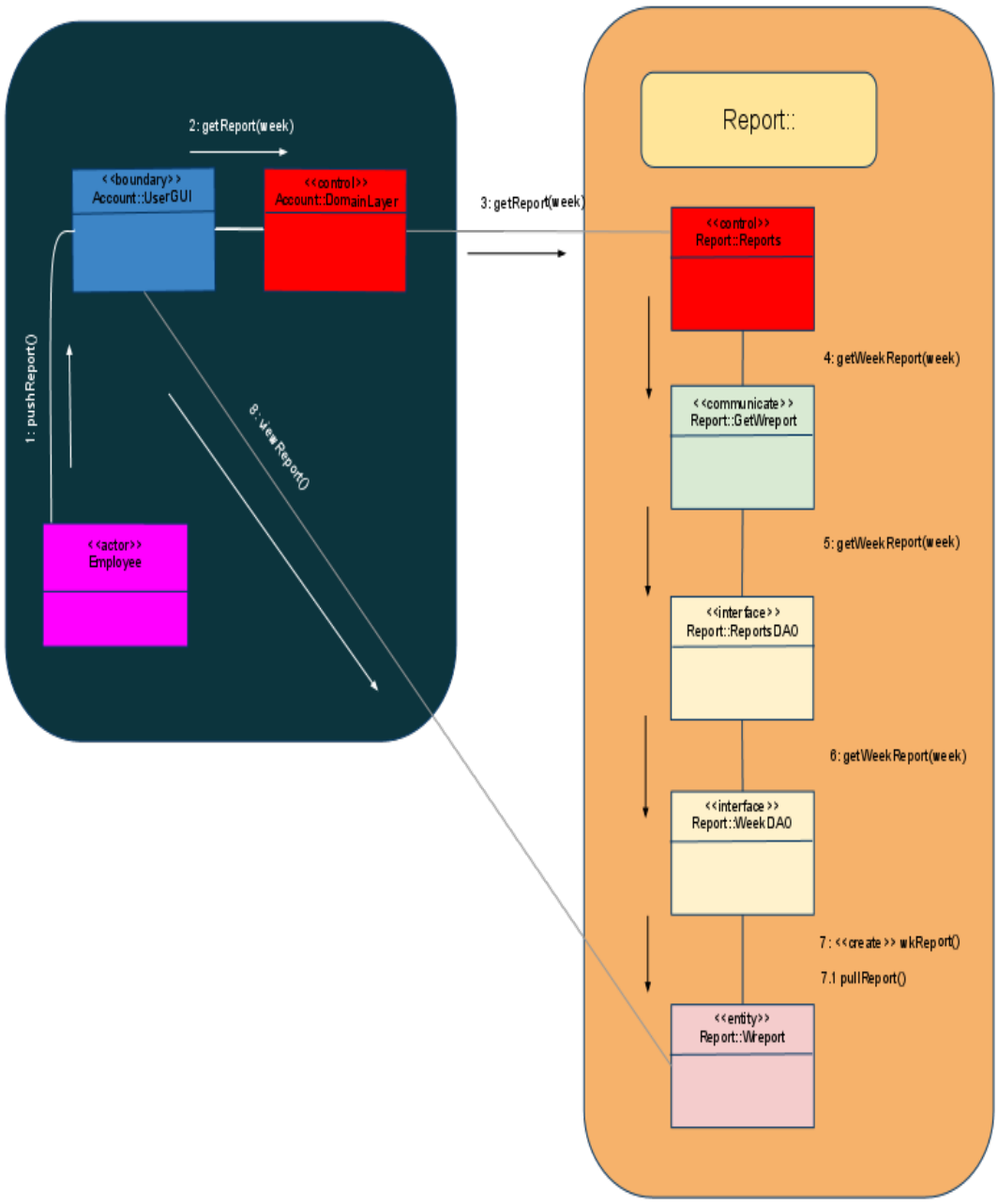
Object Report contains three simple packages that handle reading of reports for the Employee. It is assumed that the business server handles the creation of these reports so it only handles reading the reports as properly labeled. Each use case handles reports of a particular block of time such as daily, weekly and monthly.

Report::GetDreport



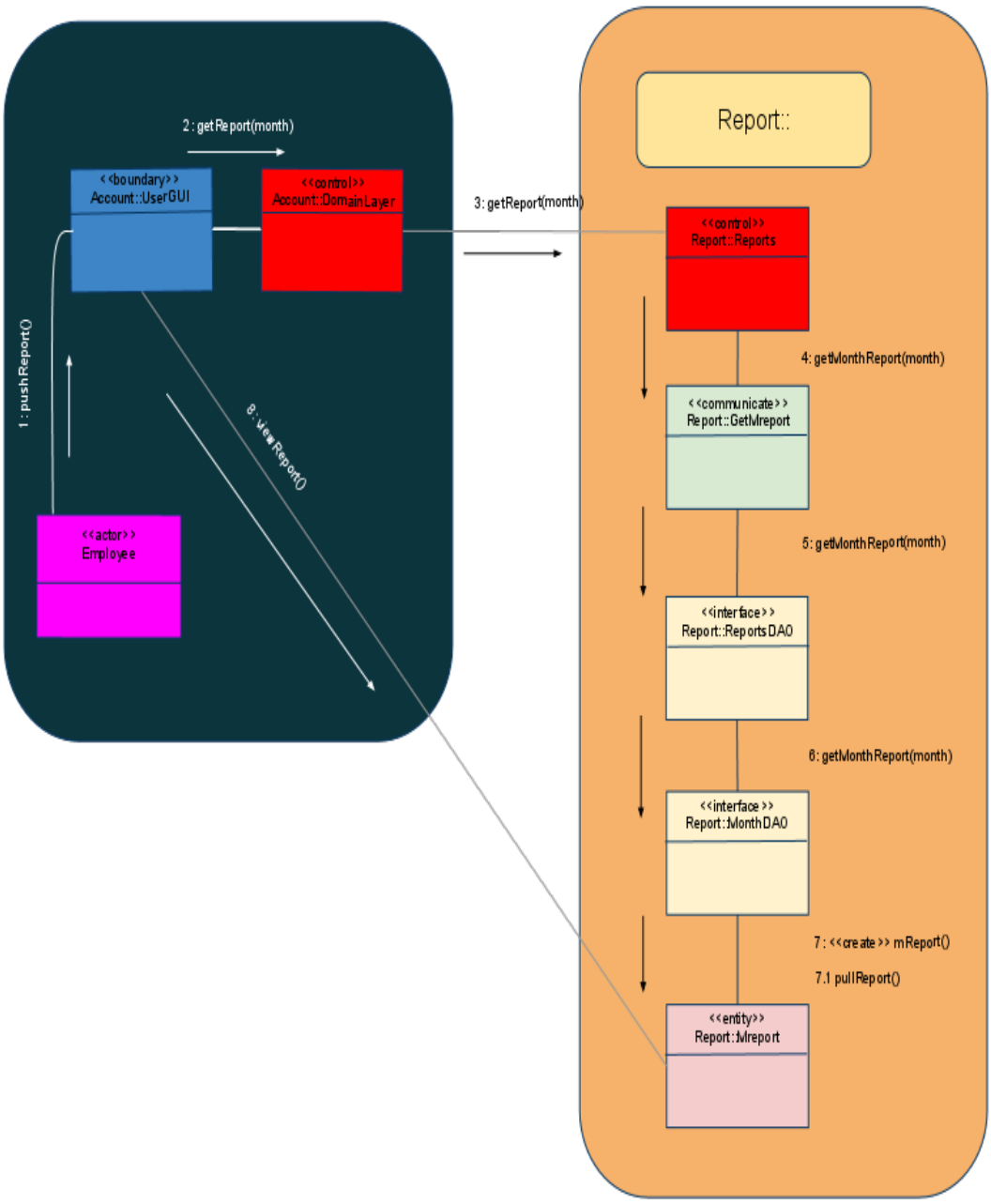
ID:10

Report::GetWreport



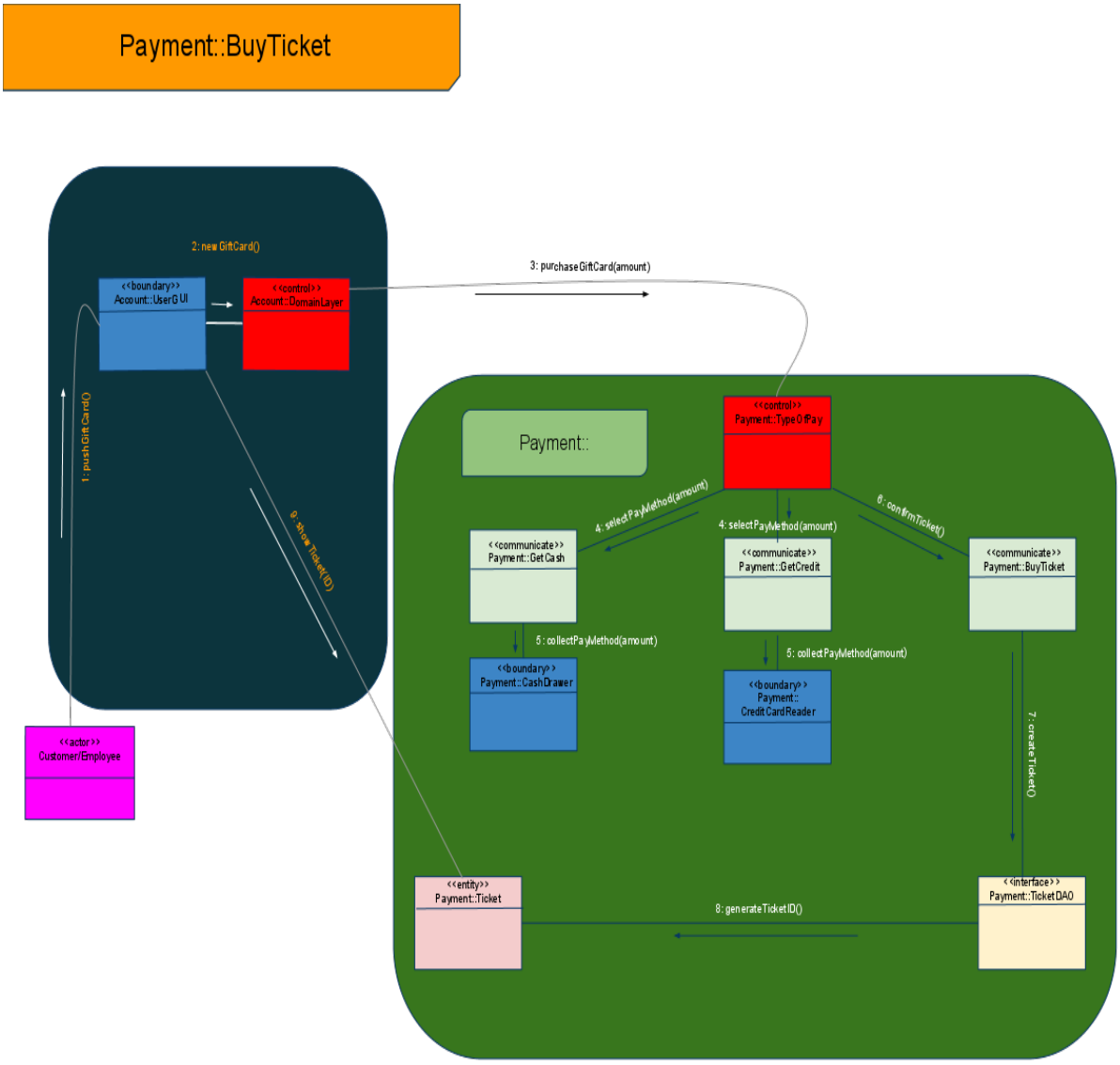
ID:11

Report::GetMreport



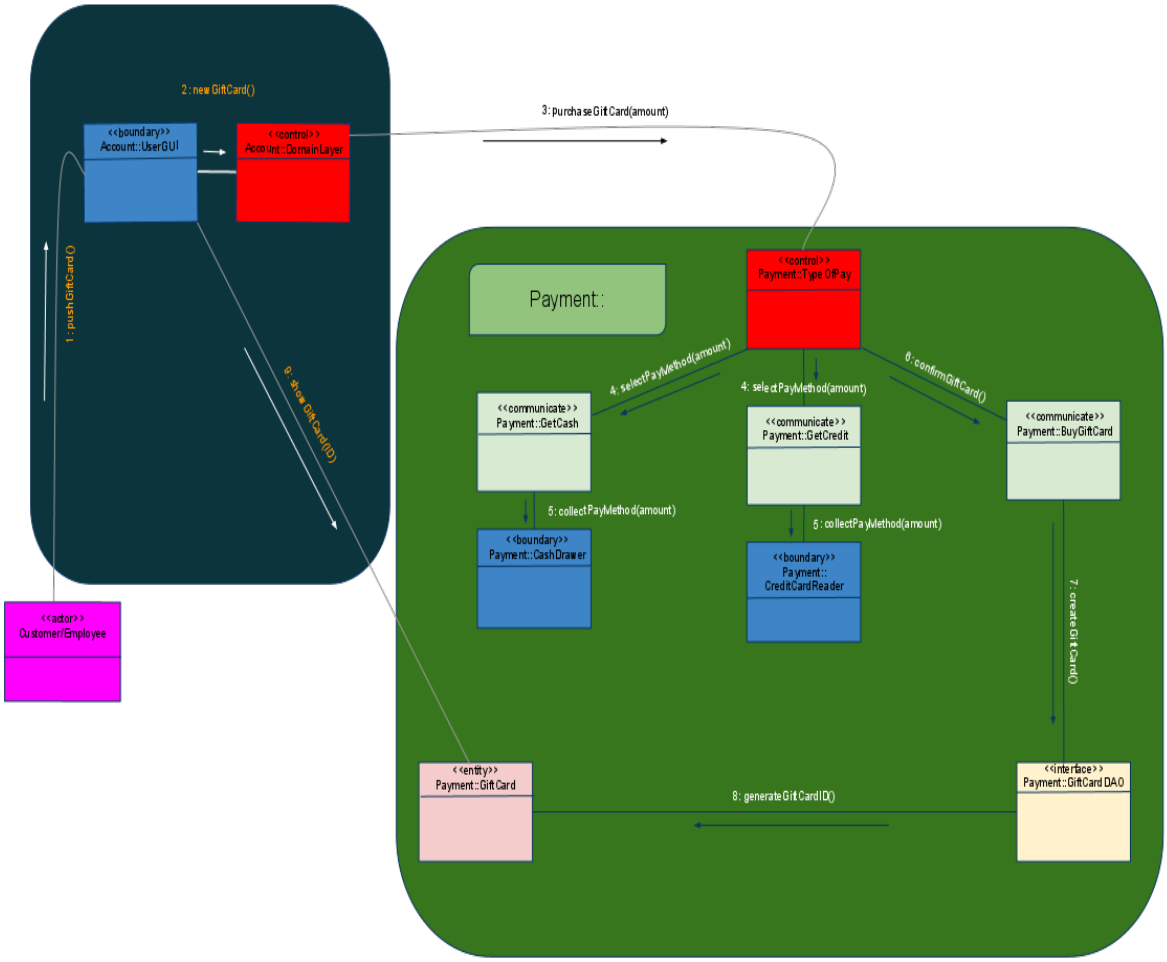
ID:12

Object Payment is the Object which handles financial transactions and stores these purchases in a data object. Payment does not handle the retrieval of tickets or Gift Cards but only the purchasing aspect and creation or removal of these data objects. However, notice Object Movie does in fact retrieve purchase information from the DAO and data objects created by Payment packages.



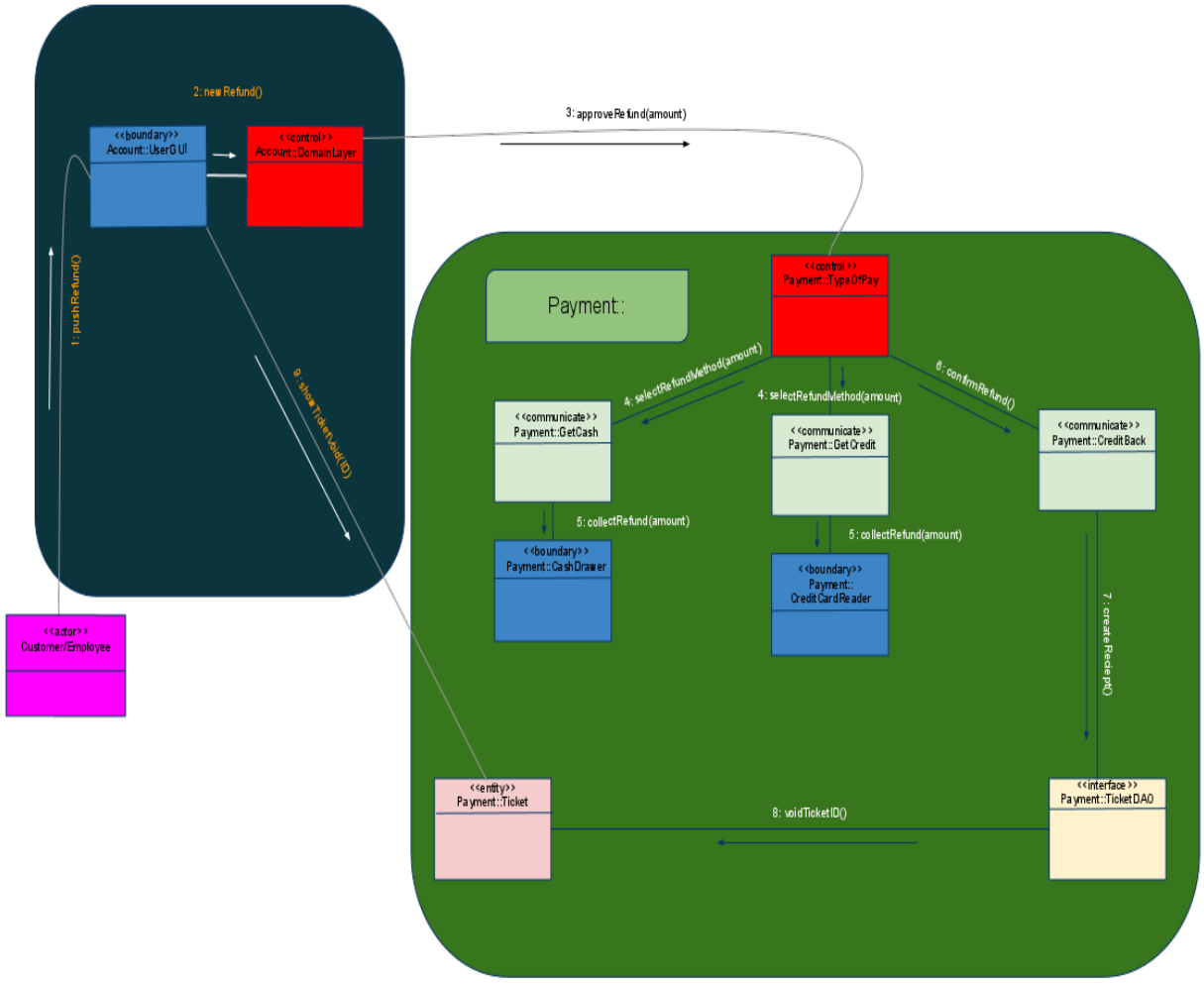
ID:13

Payment::BuyGiftCard



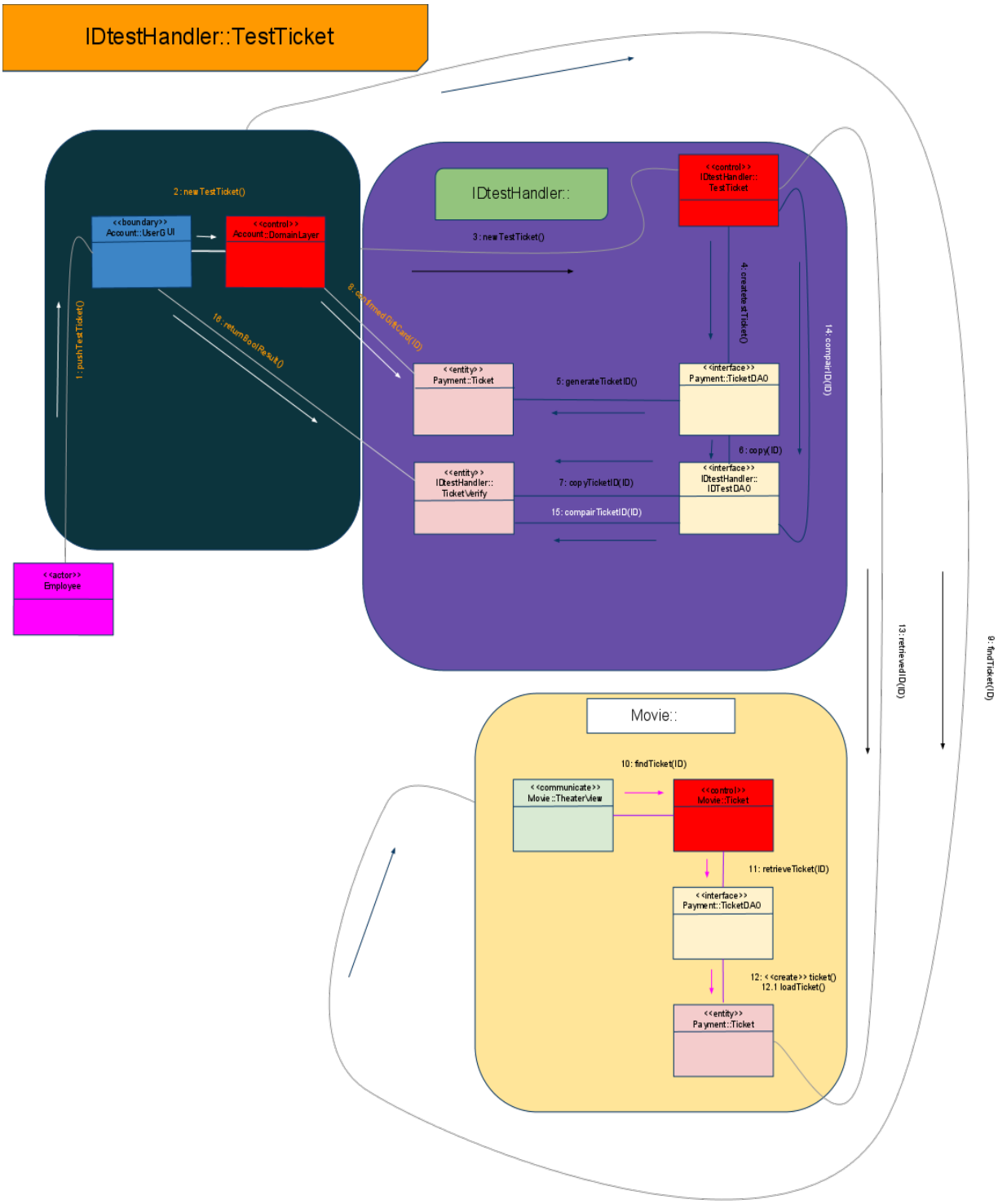
ID:14

Payment::CreditBack

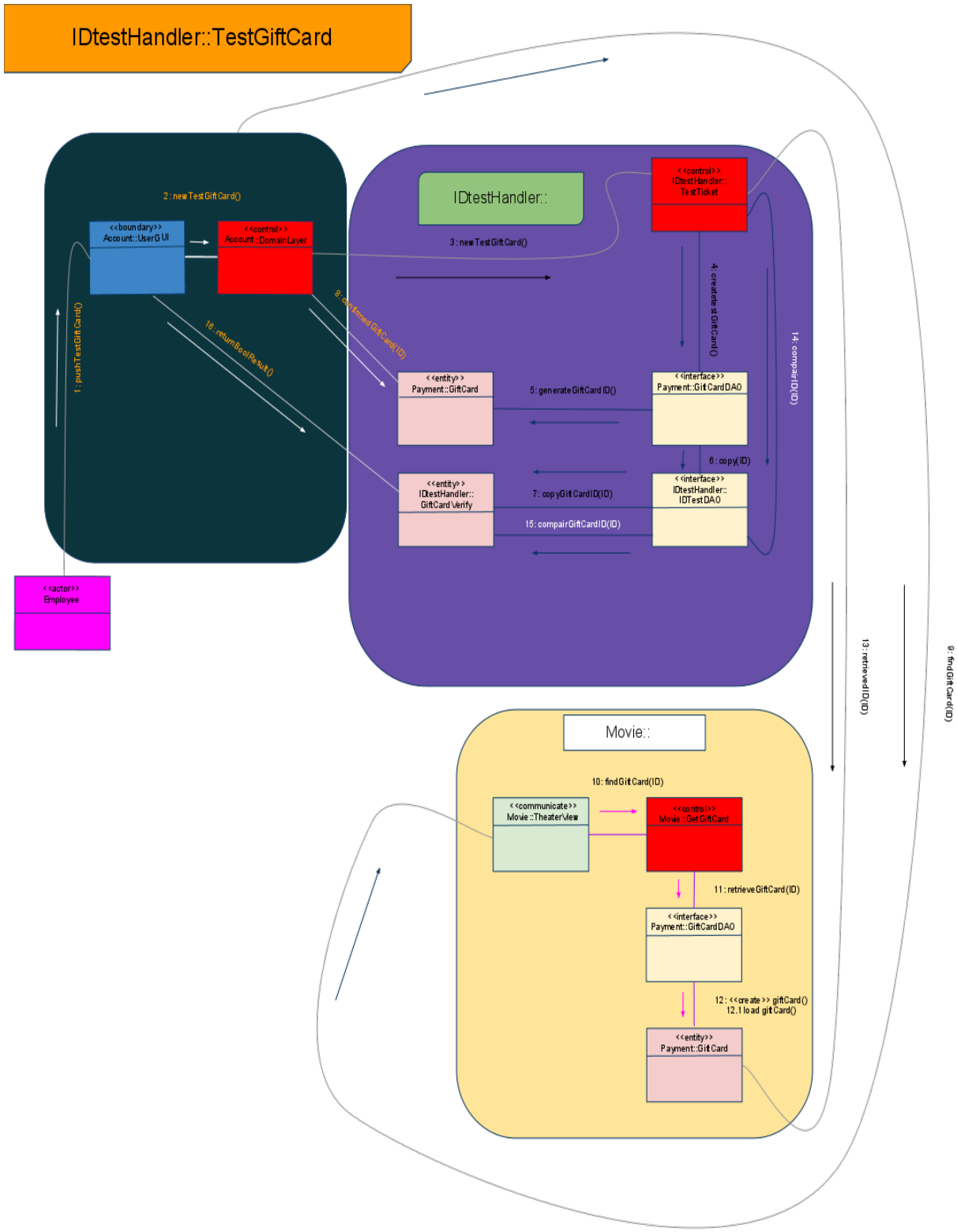


ID:15

Object IDtestHandler is the final Object which handles testing of the data communication such as creation and retrieval of data objects.



ID:16


IDtestHandler::TestGiftCard
2 : newTestGiftCard()
<<boundary>>
Account::UserGUI
<<control>>
Account::DomainLayer
1 : pushTestGiftCard()
16: returnBoolResult()
8: confirmedGiftCard(ID)
IDtestHandler::
3 : newTestGiftCard()
<<control>>
IDtestHandler::
TestTicket
4: createtestGiftCard()
14: compairID(ID)
<<entity>>
Payment::GiftCard
5: generateGiftCardID()
<<interface>>
Payment::GiftCardDAO
6 : copy(ID)
<<entity>>
IDtestHandler::
GiftCardVerify
7: copyGiftCardID(ID)
15: compairGiftCardID(ID)
<<interface>>
IDtestHandler::
IDTestDAO
<<actor>>
Employee
13 : retrievedID(ID)
9 : findGiftCard(ID)
Movie::
10: findGiftCard(ID)
<<communicate>>
Movie::TheaterView
<<control>>
Movie::GetGiftCard
11 : retrieveGiftCard(ID)
<<interface>>
Payment::GiftCardDAO
12 : <<create>> giftCard()
12.1 load giftCard()
<<entity>>
Payment::GiftCard

ID:17