

객체지향개발방법론

Distributed Vending Machine : MS129 Vending Machine

1팀 OOD 발표자료

Activity 2141. Design Real Use Cases

1. DVM들을 보여준다

| | |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------|
| Use Case | 1. DVM들을 보여준다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 1.1, R 1.2 Use cases: “음료 선택을 진행한다.” |
| Pre-Requisites | N/A |
| Typical Courses of Events | (A): Actor, (S): System 1. (S) 다른 자판기에 On/Off 및 작동 상태 확인을 요청한다 2. (S) Screen 에 사용 가능한 작동상태의DVM 목록을 보여준다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

2. 음료 선택을 진행한다.

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 2. 음료 선택을 진행한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 1.2, R 1,1 Use cases :“DVM들을 보여준다” |
| Pre-Requisites | N/A |
| Typical Courses of Events | (A): Actor, (S): System 1.(A) Customer 가 DVM 을 dialButton 을 통해,선택한다. 2.(S) 선택된 DVM 시스템을 사용한다. 3.(S) Screen을 통해, 선택된 자판기의 음료 메뉴를 Screen을 통해 보여준다. 이 때 0번부터 6번dialButton 이 현재 자판기 판매 음료이며, 6번부터 20번dialButton 이 현재 자판기에서 판매하지 않는 음료 메뉴이다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

3. 현재 자판기에 판매하는 음료수를 선택한다

| | |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 3. 현재 자판기에서 판매하는 음료수를 선택한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 2.1, R 1.2, R 2.3, R 3.1, R 3.2 Use cases :”음료 선택을 진행한다.”,”결제 방법 중 하나를 선택한다.”,”현재 자판기의 재고를 확인한다.”,”다른 자판기의 재고를 확인한다.” |
| Pre-Requisites | - 전체 20가지의 음료 종류를 알아야 한다. |
| Typical Courses of Events | (A1): Actor, (A2): Other DVMs, (S): System 1. (A1) Customer 가 현재 자판기에 판매하는 음료수를 dialButton 을 통해, 선택한다. 2. (S) “현재 자판기의 재고를 확인한다” 를 invoke 한다. 3. (S) 만약 현재 자판기에 재고가 있다면, “결제 방법 중 하나를 선택한다”를 invoke 한다. 4. (S) 만약 현재 자판기에 재고가 없다면, “다른 자판기의 재고 상태를 확인을 요청한다” 를 invoke 한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

4. 현재 자판기에서 판매하지 않는 음료수를 선택한다.

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 4. 현재 자판기에서 판매하지 않는 음료수를 선택한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 2.2, R 1.2, R 3.2, R5.1 Use cases :”음료 선택을 진행한다.”,”결제 방법 선택 메뉴를 보여준다.”, 다른 자판기에게 재고 상태를 확인을 요청한다”,”선결제를 진행한다.” |
| Pre-Requisites | - 전체 20가지의 음료 종류를 알아야 한다. |
| Typical Courses of Events | (A1): Actor, (A2): Other DVMs, (S): System 1. (A1) Customer 가 현재 자판기에서 판매하지 않는 음료수를 dialButton 을 통해, 선택한다. 2. (A2) “다른 자판기의 재고 상태를 확인을 요청한다”를 invoke한다. 3. (S) 만약 다른 자판기에 재고가 있다면 “선결제를 진행한다”를 invoke한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

5. 결제 방법 중 하나를 선택한다

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 5. 결제방법 중 하나를 선택한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 2.4, R 2.5 Use cases : “카드 결제를 진행한다”, “인증코드 결제를 진행한다” |
| Pre-Requisites | - 음료수 선택이 완료되어야한다 - 선택한 음료수의 재고가 존재해야한다 |
| Typical Courses of Events | (A): Actor, (S): System 1. (A) 카드결제는 0번 dialButton을, 코드결제는 1번 dialButton 임이 Screen에 표시되어, 사용자가 카드결제, 코드결제 중 하나를 선택한다. 2. (S) 카드결제를 선택할 경우 “카드결제를 진행한다” 를 invoke한다. 3. (S) 코드결제를 선택할 경우 “인증코드 결제를 진행한다”를 invoke 한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

6. 카드 결제를 진행한다.

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 6. 카드 결제를 진행한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 2.5, R 2.7, R 5.1 Use cases : “결제를 취소한다.”, “선결제를 진행한다.” |
| Pre-Requisites | - Customer 가 결제 방법 중 카드 결제를 선택해야 한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (A) Customer 는 카드 결제를 위해, 카드를 카드 단말기에 꽂는다. 2. (S) Customer 가 카드단말기에 사용한 카드가 결제를 할 수 있는 종류의 카드인지 확인한다. 3. (S) 결제할 음료의 가격을 Screen을 통해, 전달받는다. 4. (S) 결제 단말기에 사용한 카드의 잔액이 음료의 가격을 결제할 수 있는지 확인한다. 5. (S) 카드 결제를 완료한다. 6. (S) 재고를 업데이트한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | Line 2. 카드가 결제를 할 수 없는 종류의 카드인 경우, “결제를 취소한다.”를 invoke 한다. Line 3. 카드의 잔액이 음료의 가격을 결제할 수 없을 경우, “결제를 취소한다”를 invoke 한다. |

Activity 2141. Design Real Use Cases

7. 인증코드 결제를 진행한다

| | |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 7. 인증코드 결제를 진행한다. |
| Actor | Customer |
| Type | Evident |
| Cross Reference | Functions: R 2.5, R 2.6, R 6.1 Use cases : “결제를 취소한다.”, “입력한 인증코드가 유효한지 확인한다.” |
| Pre-Requisites | - Customer 가 결제 방법 중 인증코드 결제를 선택해야 한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (A) Customer 는 인증코드를 dialButton을 사용하여, 입력한 뒤, 확인 버튼을 누른다. 2. (S) 입력된 인증코드에 대해, “입력한 인증코드가 유효한지 확인한다” 를 invoke한다. 3. (S) 인증코드가 일치하면 인증코드 결제를 완료하고 음료를 제공한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | Line 3. 인증코드가 일치하지 않는다면 사용자에게 일치하지 않는다고 Screen에 알려준 후 초기 화면으로 돌아간다. |

Activity 2141. Design Real Use Cases

8. 결제를 취소한다

| | |
|--------------------------------------|--------------------------------------------------------------------------------|
| Use Case | 8. 결제를 취소한다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 2.6, R 2.4, R 2.5 Use cases : “카드 결제를 진행한다.”, “인증코드 결제를 진행한다.” |
| Pre-Requisites | - 카드 결제, 인증코드 결제 과정 중 결제 취소가 발생한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (S) 초기화면으로 돌아가, Screen에 나타낸다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

9. 현재 자판기의 재고를 확인한다

| | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Use Case | 9. 현재 자판기의 재고를 확인한다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 3.1, R 2.1 Use cases : “현재 자판기에서 판매하는 음료수를 선택한다.” |
| Pre-Requisites | - 현재 자판기에서 판매하는 음료를 선택하거나 다른 자판기에게 재고 상태 확인 요청을 받았을 때만 실행된다. |
| Typical Courses of Events | (A1): Actor, (A2): DVMs, (S): System 1. (A2) 현재 자판기의 재고를 시스템에게 알려준다. 2. (A2) 요청받은 음료의 재고가 없다면, 재고가 없다고 알려준다. |
| Alternative Courses of Events | 1(a). 만약 현재 자판기가 작동 불능상태라면 재고가 없다고 간주하고 재고 없음을 알려준다. |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

10. 다른 자판기의 재고를 확인한다.

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 10. 다른 자판기의 재고를 확인한다 |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 3,2, R 2.2 Use cases : “현재 자판기에서 판매하지 않는 음료수를 선택한다.” |
| Pre-Requisites | - 현재 자판기의 재고가 부족하거나, 현재 자판기에서 판매하지 않는 음료를 사용자가 선택할 경우에만 실행된다. -음료가 선택된 상태여야한다. |
| Typical Courses of Events | (A): Controller, (S): System 1. (A) 다른 자판기들에게 해당음료의 재고정보를 요청한다. 2. (S) 재고가 없다는 응답을 받을 경우, 재고가 없다는 것을 사용자에게 알린 뒤 초기화면으로 돌아간다. 3. (S) 재고가 존재할 경우, 재고가 존재하는 DVM 리스트를 전달한다.. |
| Alternative Courses of Events | 2 (a). 만약 요청한 DVM의 상태가 Off 혹은 작동 불능일 경우, 재고가 없다고 간주한다. 2 (b). 만약 일정 시간 이상 응답이 없다면, 재고가 없다고 간주한다. |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

11. 자판기의 위치를 보여준다.

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 11. 자판기의 위치를 보여준다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 4.1,R 5.1 Use cases : “선결제를 진행한다.” |
| Pre-Requisites | - Customer 가 선결제를 완료한 후 실행되어야 한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (S) 사용자가 선택한 음료를 판매하는 자판기들의 위치를 사용자에게 Screen을 통해,보여준다. 2. (S) 사용자가 선택한 음료를 판매하지 않는 자판기들의 위치는 표시하지 않는다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

Activity 2141. Design Real Use Cases

12. 선결제를 진행한다.

| | |
|--------------------------------------|----------------------------------------------------------------------------|
| Use Case | 12. 선결제를 진행한다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 5.1, R 2.4 Use cases : "카드 결제를 진행한다." |
| Pre-Requisites | - 현재 자판기에서 판매하지 않는 음료 또는 현재 자판기에 재고가 없는 음료가 선택되어야 한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (S) 카드 결제를 진행한다. 2. (S) 사용자에게 인증코드를 발급한다. |
| Alternative Courses of Events | N/A |
| Exceptional Courses of Events | N/A |

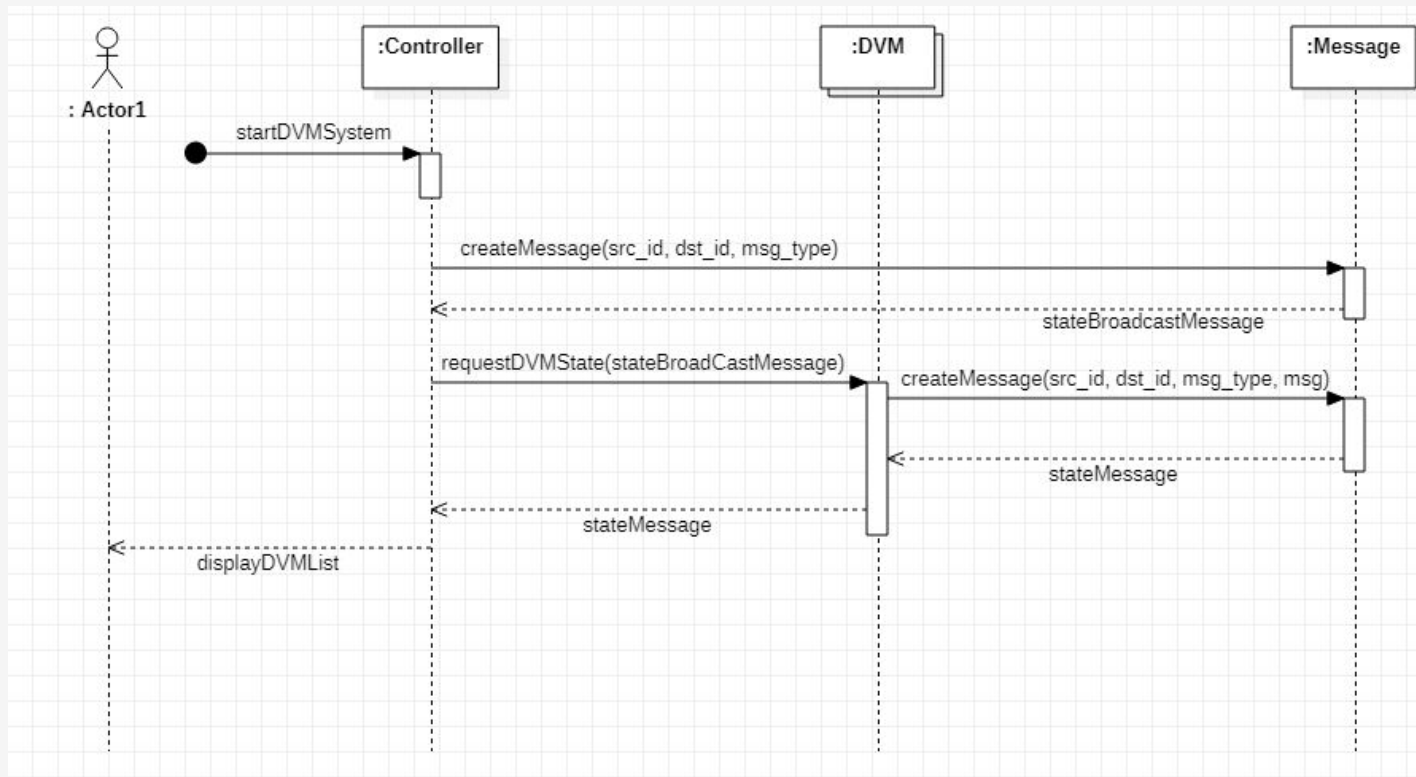
Activity 2141. Design Real Use Cases

13. 입력한 인증코드가 유효한지 확인한다.

| | |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case | 13. 입력한 인증코드가 유효한지 확인한다. |
| Actor | System |
| Type | Hidden |
| Cross Reference | Functions: R 6.1,R 2.5 Use cases : "인증코드 결제를 진행한다." |
| Pre-Requisites | - Customer가 선결제에서 발급받은 인증코드를 입력한다. |
| Typical Courses of Events | (A): Actor, (S): System 1. (S) 사용자가 dialButton을 사용하여 입력한 인증코드와 발급된 인증코드가 일치하지 않는다면, 일치하지 않는 인증코드라고 Screen을 통해 사용자에게 알려준다. 2. (S) 사용자가 입력한 인증코드와 발급한 인증코드가 일치한다면 일치하는 인증코드라고 알려준다. |
| Alternative Courses of Events | 1(a). 만약 인증코드를 발급한 내역이 없는 상태에서 사용자가 인증코드를 입력한다면 일치하지 않는 인증코드로 간주한다. |
| Exceptional Courses of Events | N/A |

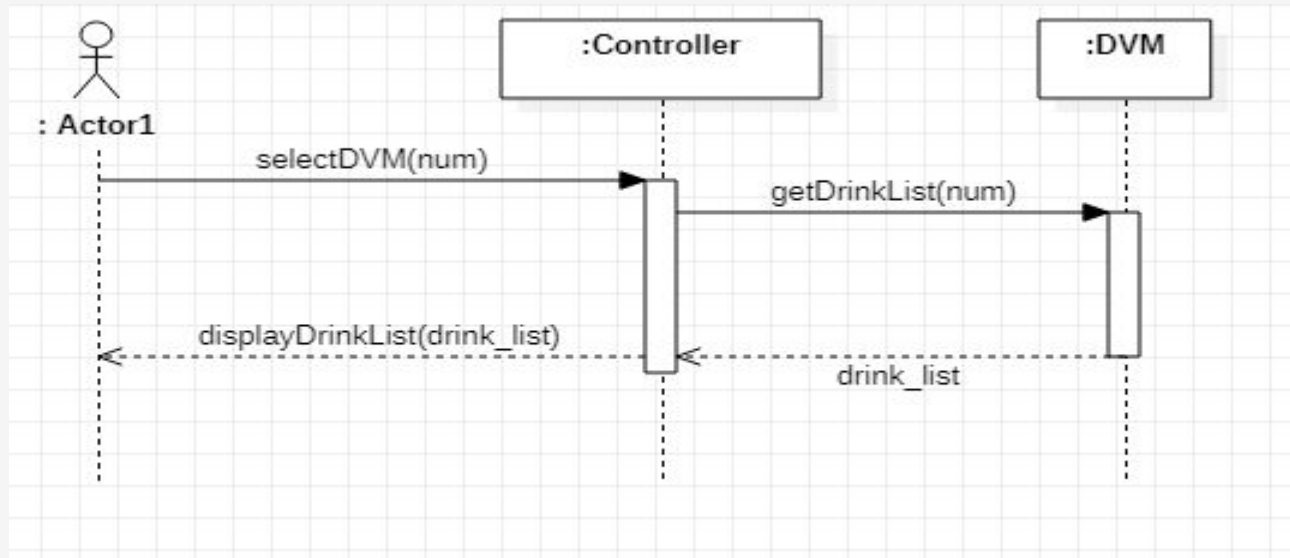
Activity 2143. Define Interaction Diagram

1. DVM들을 보여준다.



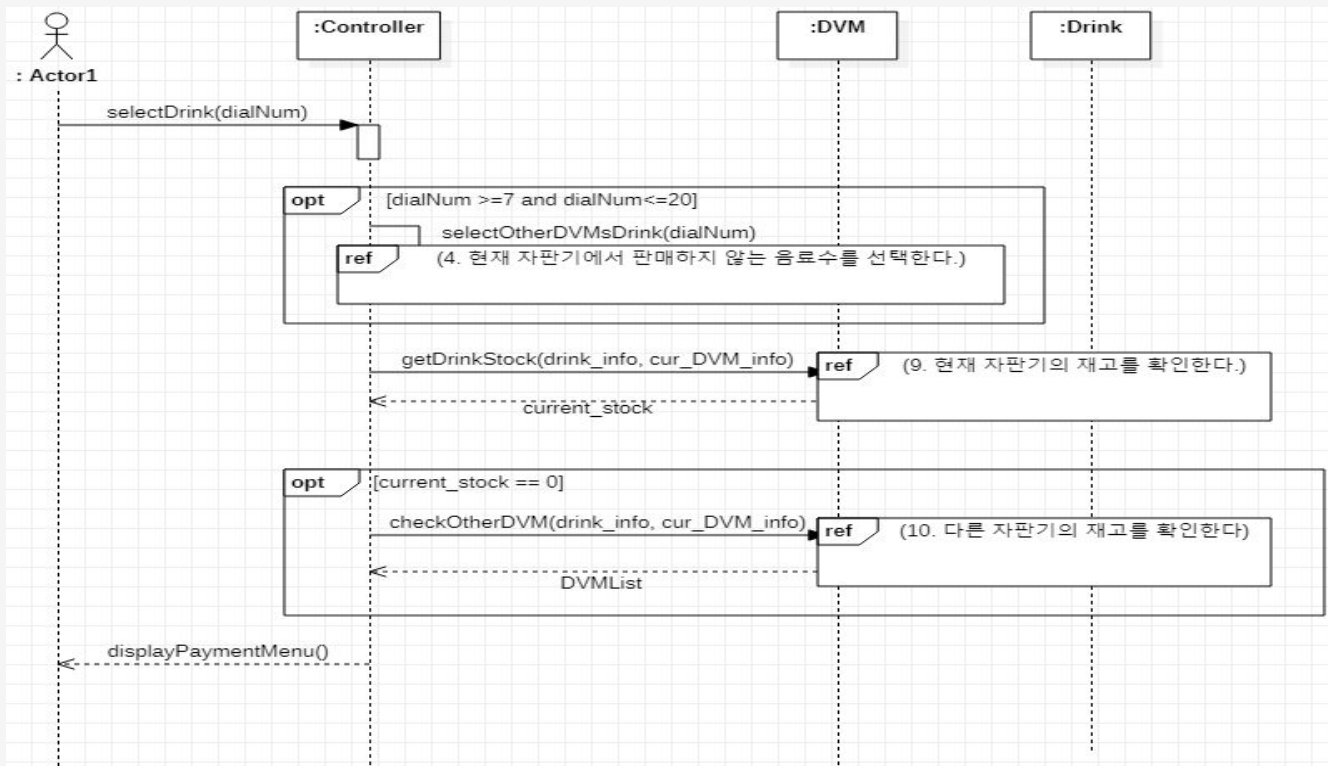
Activity 2143. Define Interaction Diagram

2. 음료 선택을 진행한다.



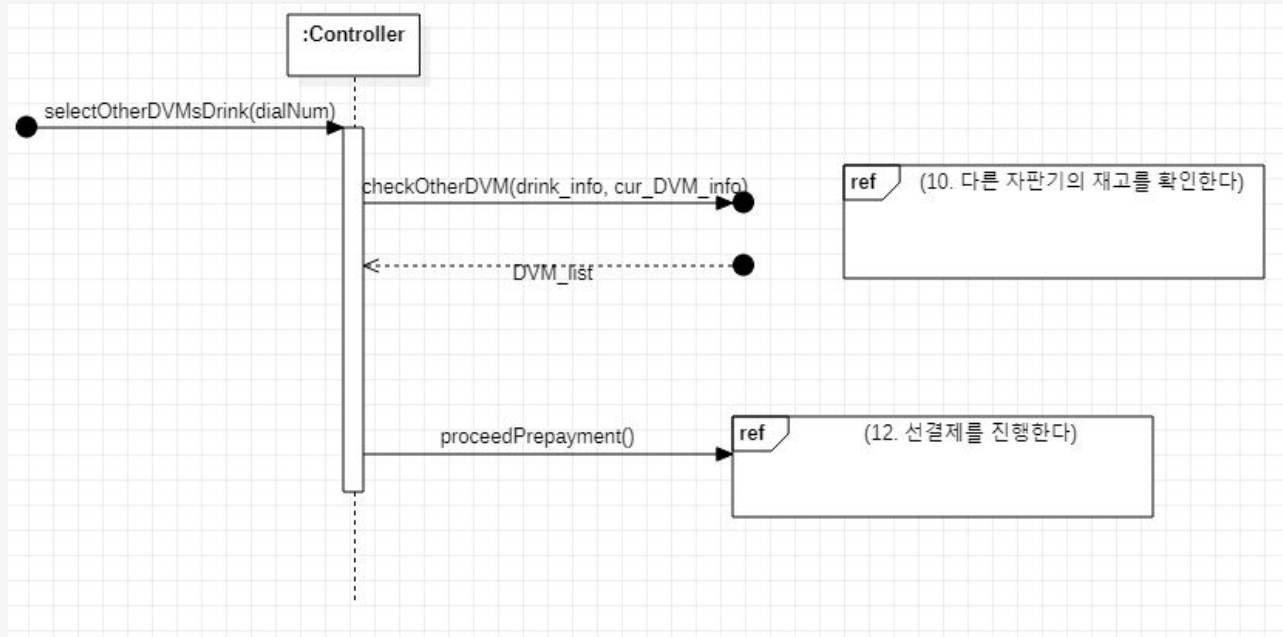
Activity 2143. Define Interaction Diagram

3. 현재 자판기에 판매하는 음료수를 선택한다.



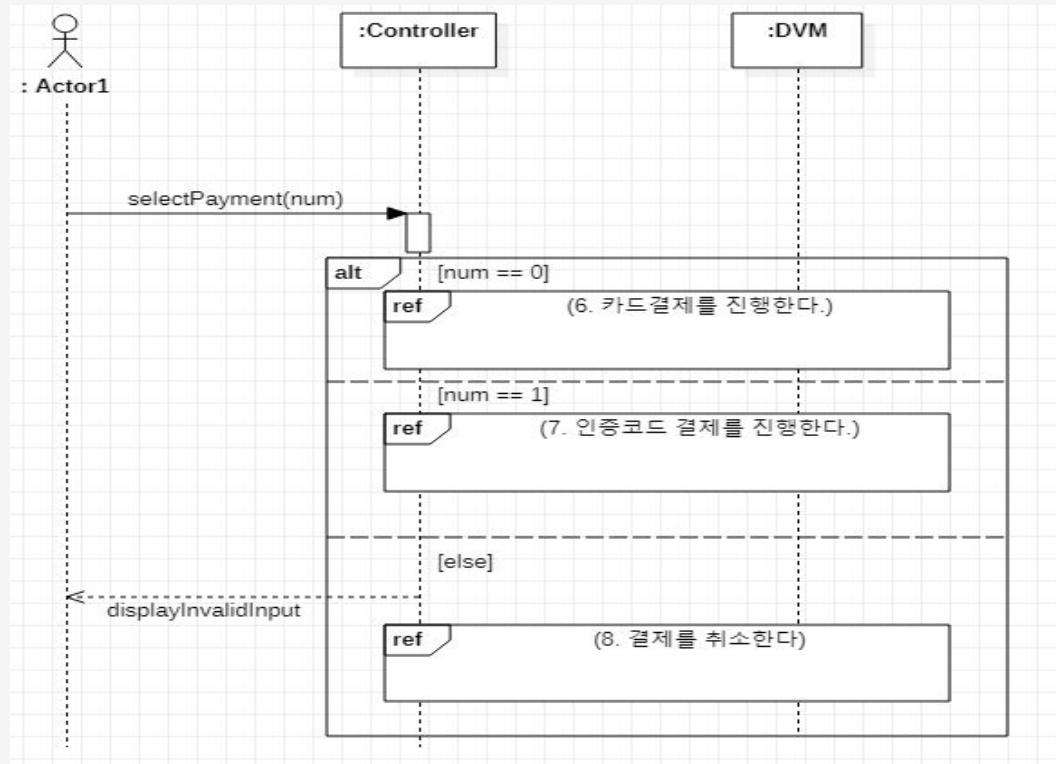
Activity 2143. Define Interaction Diagram

4. 현재 자판기에서 판매하지 않는 음료수를 선택한다.



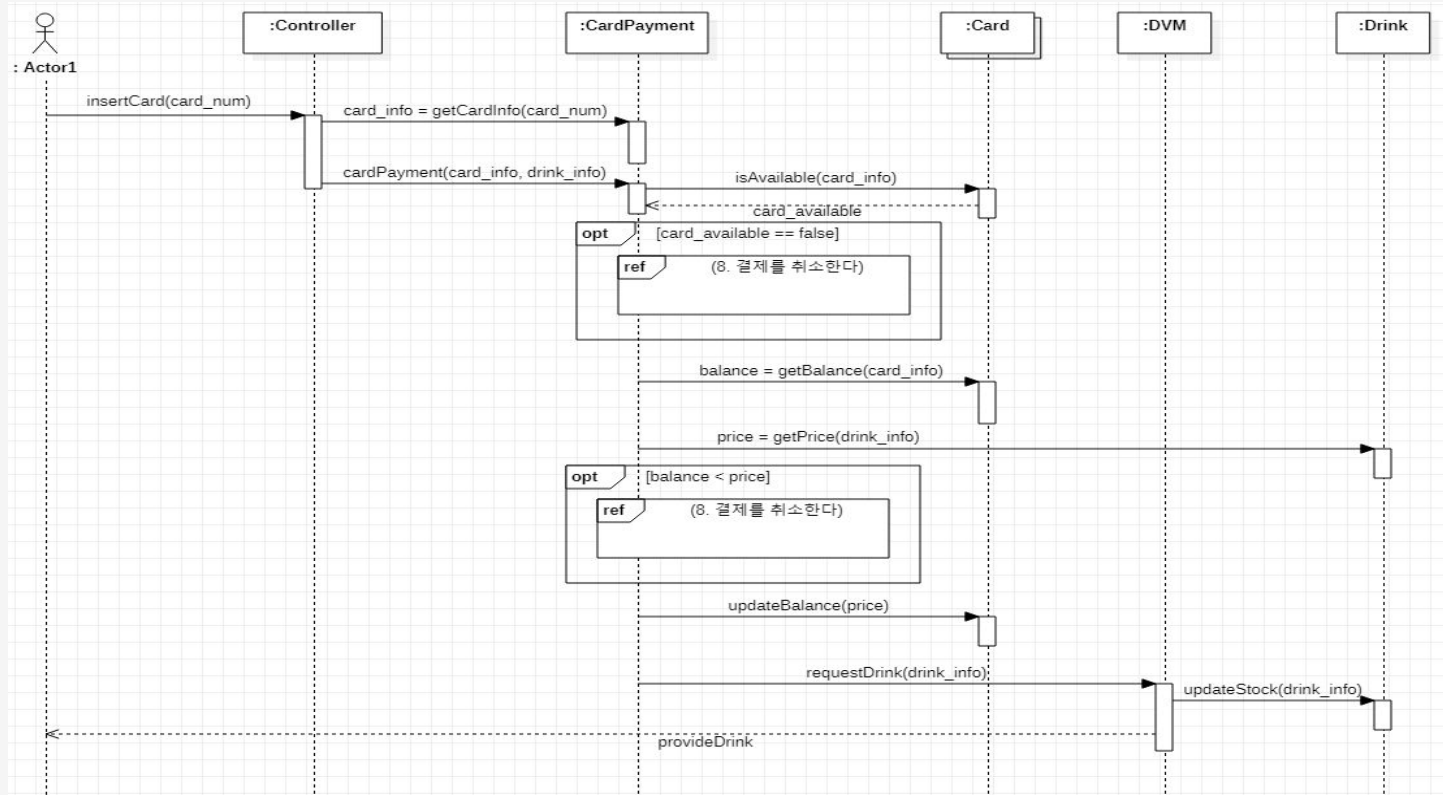
Activity 2143. Define Interaction Diagram

5.결제 방법 중 하나를 선택한다.



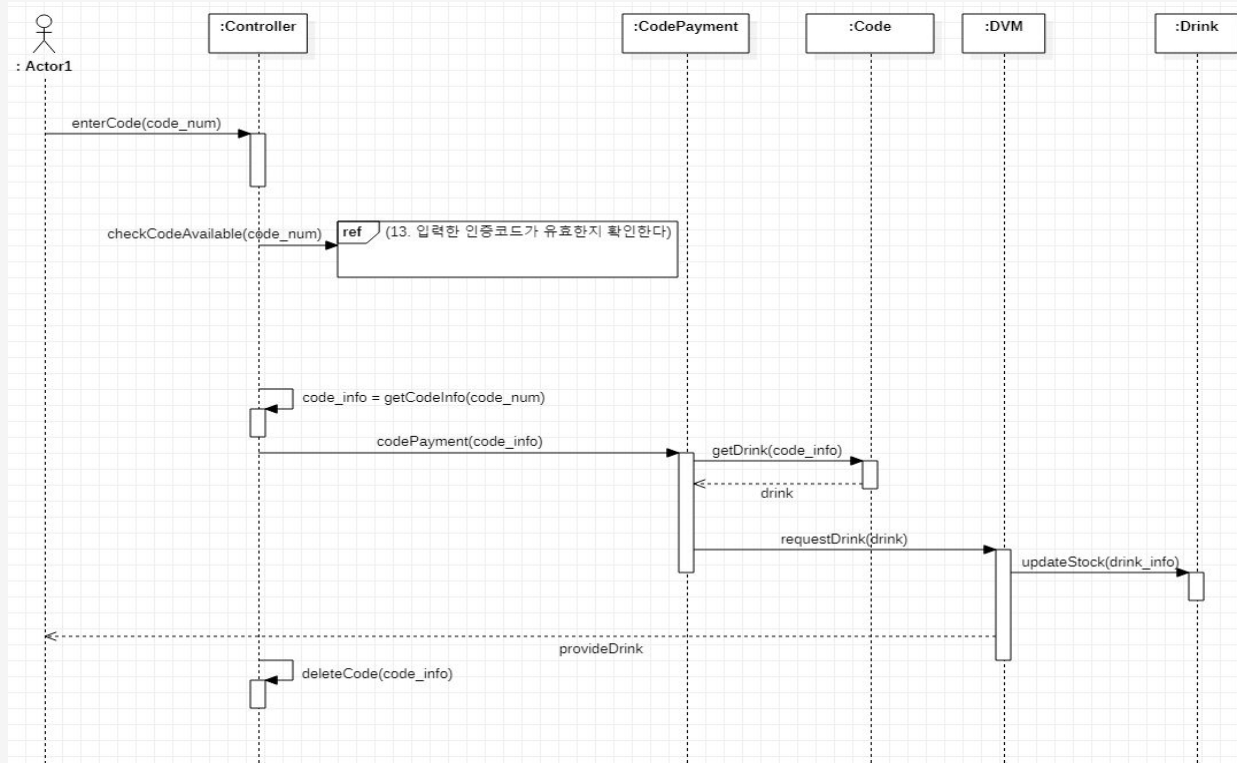
Activity 2143. Define Interaction Diagram

6. 카드 결제를 진행한다.



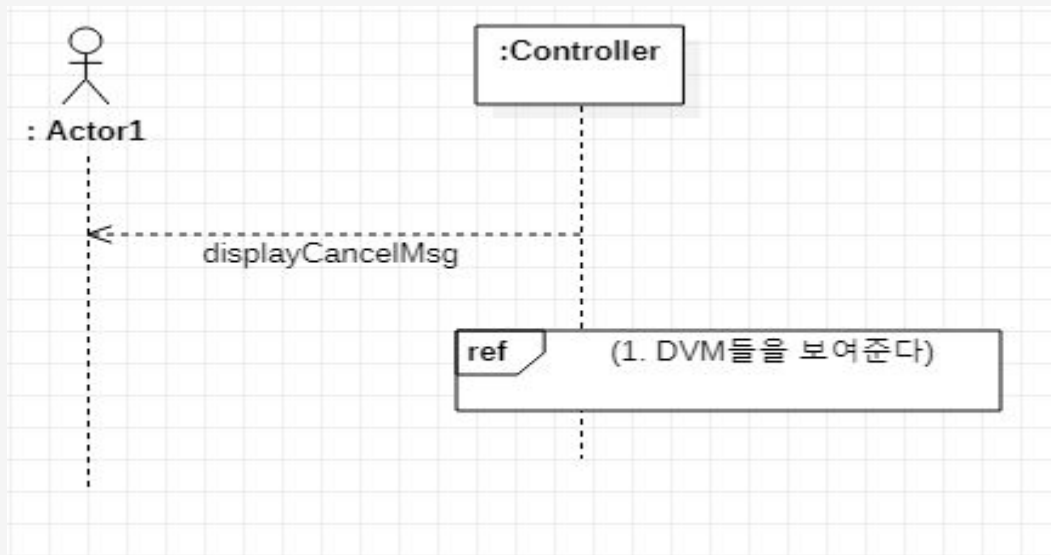
Activity 2143. Define Interaction Diagram

7.인증코드 결제를 진행한다.



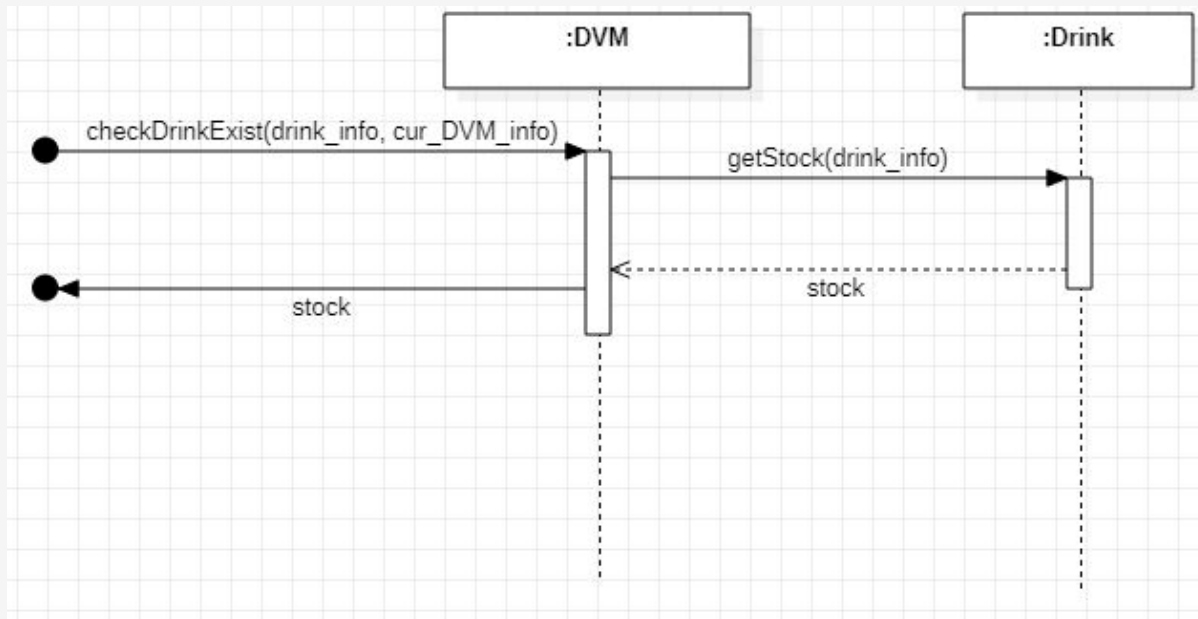
Activity 2143. Define Interaction Diagram

8. 결제를 취소한다.



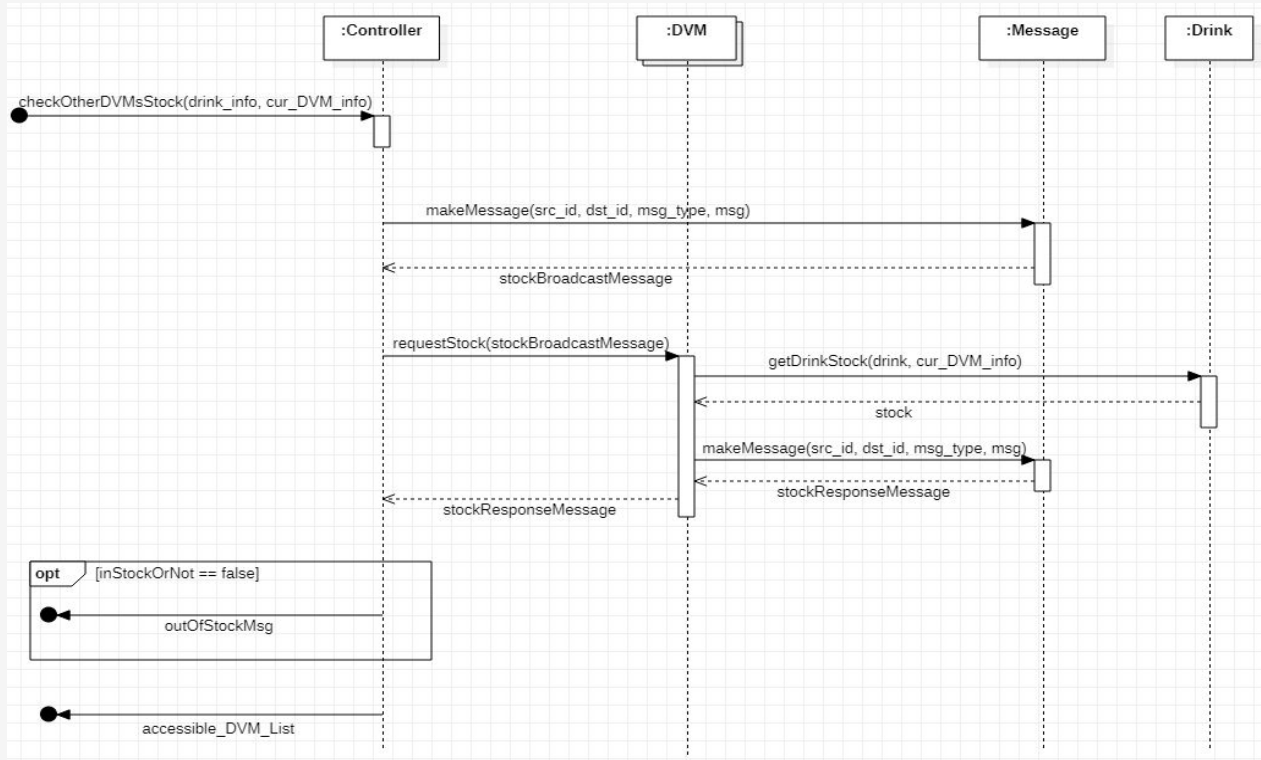
Activity 2143. Define Interaction Diagram

9. 현재 자판기의 재고를 확인한다.



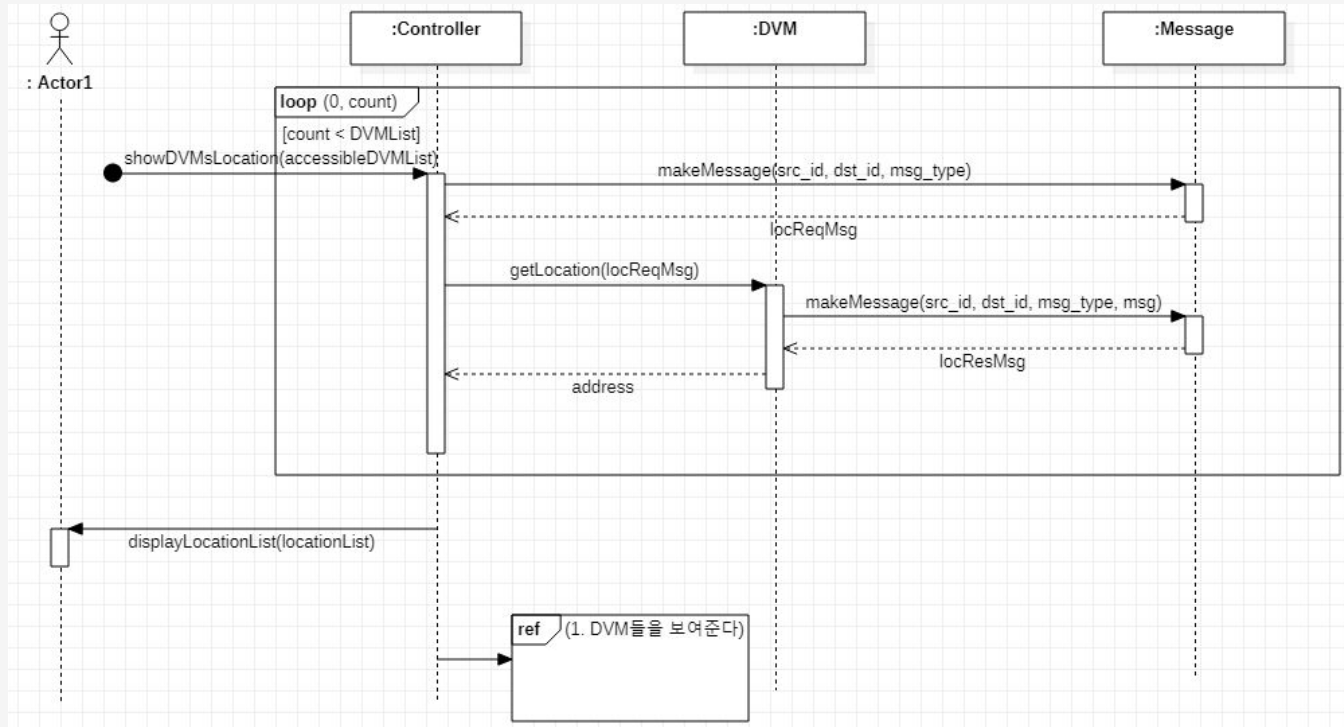
Activity 2143. Define Interaction Diagram

10. 다른 자판기의 재고를 확인한다.



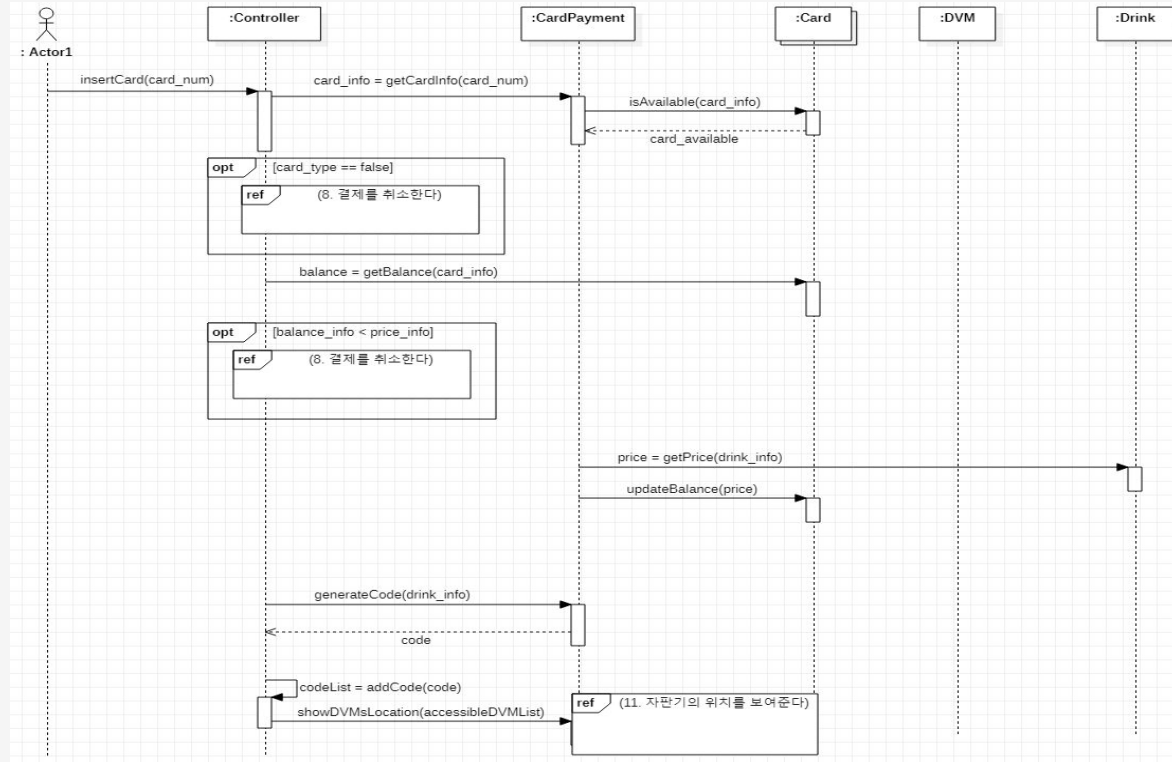
Activity 2143. Define Interaction Diagram

11. 재판기의 위치를 보여준다.



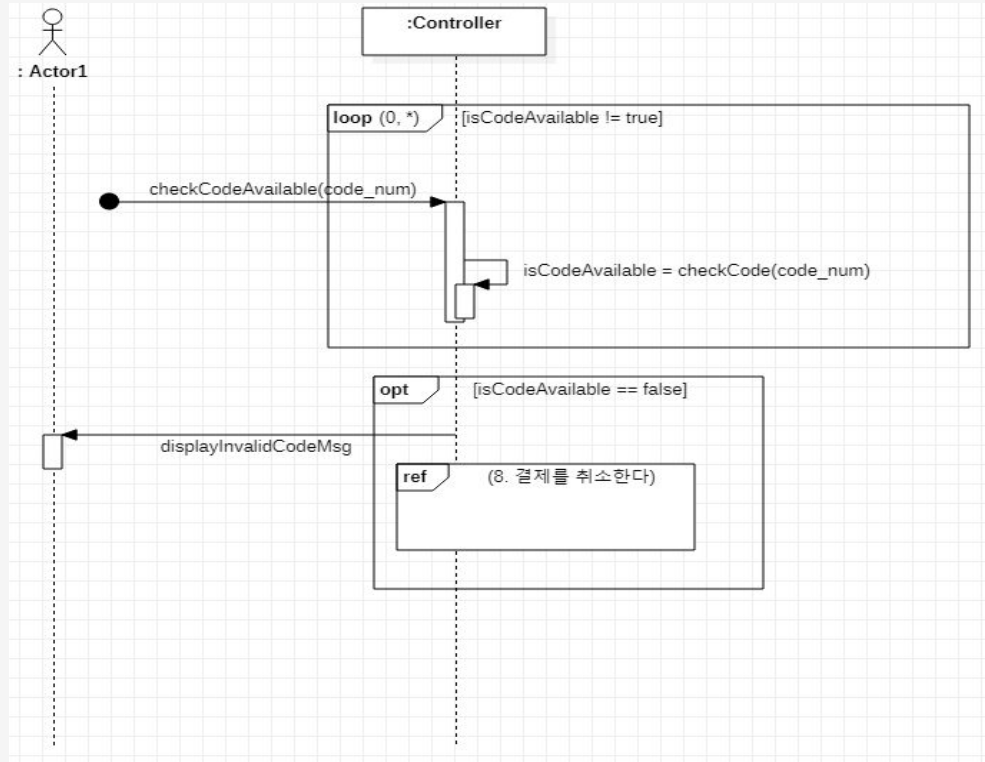
Activity 2143. Define Interaction Diagram

12. 선결제를 진행한다.

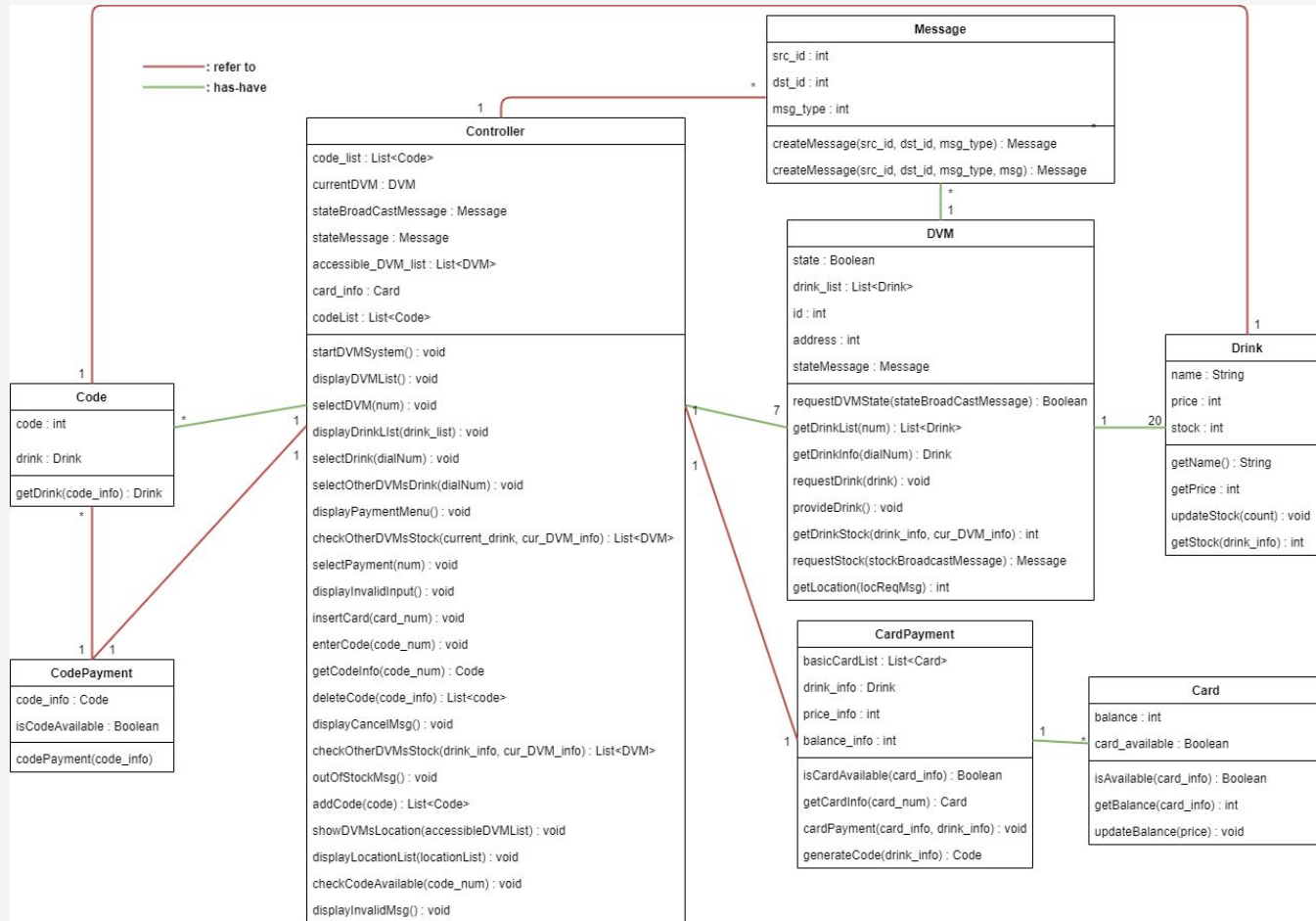


Activity 2143. Define Interaction Diagram

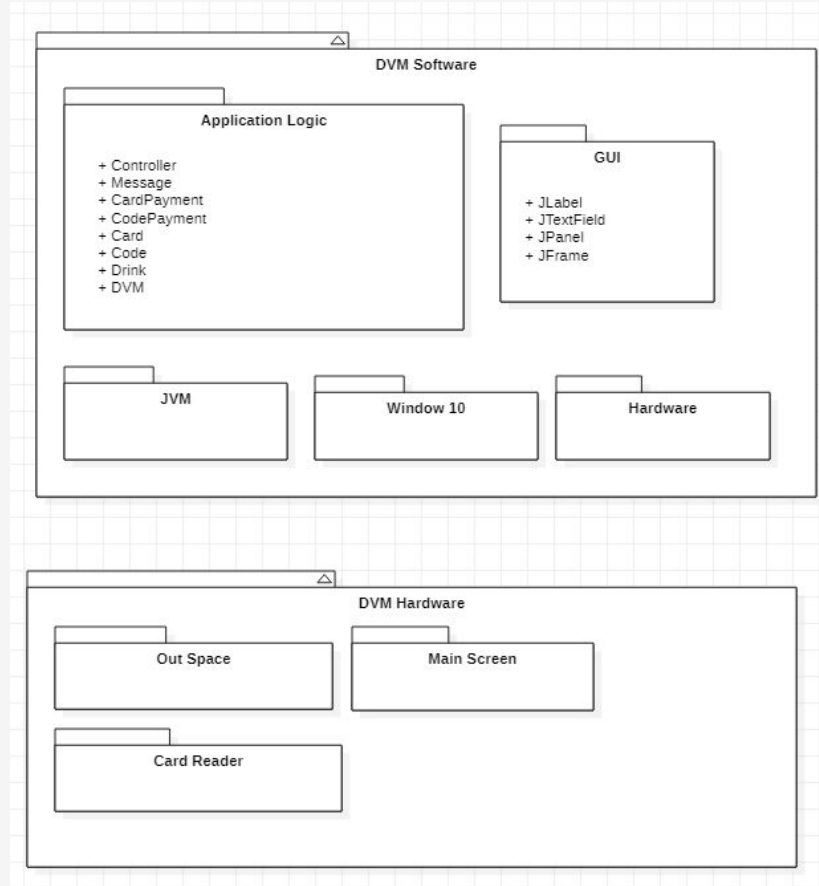
13. 입력한 인증코드가 유효한지 확인한다.



Activity 2144. Define Design Class Diagram



Activity 2145. Refine System Architecture



Activity 2147. Perform 2040 Traceability Analysis

| Essential Use-case | Operation | Method | Class |
|----------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| DVM들을 보여준다 | 1: Response My State | void startDVMSystem() void displayDVMList() | Controller |
| 음료 선택을 진행한다 | 2: Select DVM | void selectDVM(num) | |
| 현재 자판기에서 판매하는 음료수를 선택한다 | 3: Select Drink | void displayDrinkList(drink_list) void selectDrink(dialNum) | |
| 현재 자판기에서 판매하지 않는 음료수를 선택한다 | 4: Select Payment | void selectOtherDVMsDrink(dialNum) | |
| 결제 방법 중 하나를 선택한다 | 5: Insert Card | List<DVM> checkOtherDVMsStock(current_drink, cur_DVM_info) | |
| 카드 결제를 진행한다 | 6: Enter Code | void displayPaymentMenu() void proceedPrepayment() void selectPayment(num) | |
| 인증코드 결제를 진행한다 | 7: Get Location | void displayInvalidInput() | |
| 자판기의 위치를 보여준다 | | void insertCard(card_num) void enterCode(code_num) Code getCodeInfo(code_num) List<Code> deleteCode(code_info) void outOfStockMsg() | Controller |
| | | List<Code> addCode(code) | |
| | | void showDVMsLocation(accessibleDVMList) | |
| | | void displayLocationList(locationList) | |
| | | void checkCodeAvailable(code_num) void displayInvalidMsg() | |
| | | Boolean requestDVMState(int dialNum, codeInfo) | |
| | | List<Drink> getDrinkList(num) Drink getDrinkInfo(dialNum) void requestDrinks(drink) void provideDrink() | |
| | | int getDrinkStock(drink_info, cur_DVM_info) Message requestStock(stockBroadcastMessage) | DVM |
| | | int getLocation(locReqMsg) | |
| | | Message createMessage(src_id, dst_id, msg_type) Message createMessage(src_id, dst_id, msg_type, msg) | Message |
| | | Boolean isCardAvailable(card_info) Card getCardInfo(card_num) void cardPayment(card_info, drink_info) Code generateCode(drink_info) | CardPayment |
| | | Boolean isAvailable(card_info) int getBalance(card_info) void updateBalance(price) | Card |
| | | String getHame() | Drink |
| | | int getPrice() void updateStock(count) | |
| | | int getStock(drink_info) Drink getDrink(code_info) codePayment(code_info) | Code CodePayment |

| Essential Use-case | Operation | Method | Class |
|----------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| DVM들을 보여준다 | 1: Response My State | void startDVMSystem() void displayDVMList() | Controller |
| 음료 선택을 진행한다 | 2: Select DVM | void selectDVM(num) | |
| 현재 자판기에서 판매하는 음료수를 선택한다 | 3: Select Drink | void displayDrinkList(drink_list) void selectDrink(dialNum) | |
| 현재 자판기에서 판매하지 않는 음료수를 선택한다 | 4: Select Payment | void selectOtherDVMsDrink(dialNum) | |
| 결제 방법 중 하나를 선택한다 | 5: Insert Card | List<DVM> checkOtherDVMsStock(current_drink, cur_DVM_info) | |
| 카드 결제를 진행한다 | 6: Enter Code | void displayPaymentMenu() void proceedPrepayment() void selectPayment(num) | |
| 인증코드 결제를 진행한다 | 7: Get Location | void displayInvalidInput() | |
| 자판기의 위치를 보여준다 | | void insertCard(card_num) void enterCode(code_num) Code getCodeInfo(code_num) List<Code> deleteCode(code_info) void outOfStockMsg() | Controller |
| | | List<Code> addCode(code) | |
| | | void showDVMsLocation(accessibleDVMList) | |
| | | void displayLocationList(locationList) | |
| | | void checkCodeAvailable(code_num) void displayInvalidMsg() | |
| | | Boolean requestDVMState(int dialNum, codeInfo) | |
| | | List<Drink> getDrinkList(num) Drink getDrinkInfo(dialNum) void requestDrinks(drink) void provideDrink() | |
| | | int getDrinkStock(drink_info, cur_DVM_info) Message requestStock(stockBroadcastMessage) | DVM |
| | | int getLocation(locReqMsg) | |
| | | Message createMessage(src_id, dst_id, msg_type) Message createMessage(src_id, dst_id, msg_type, msg) | Message |
| | | Boolean isCardAvailable(card_info) Card getCardInfo(card_num) void cardPayment(card_info, drink_info) Code generateCode(drink_info) | CardPayment |
| | | Boolean isAvailable(card_info) int getBalance(card_info) void updateBalance(price) | Card |
| | | String getHame() | Drink |
| | | int getPrice() void updateStock(count) | |
| | | int getStock(drink_info) Drink getDrink(code_info) codePayment(code_info) | Code CodePayment |

Activity 2147. Perform 2040 Traceability Analysis

| Essential Use-case | Operation | Method | Class |
|----------------------------|----------------------|--------------------------------------------------------------|-------------|
| DVM를 보여준다 | 1: Response My State | void startDVMSystem() | Controller |
| 음료 선택을 진행한다 | 2: Select DVM | void displayDVMList() | |
| 음료 화면기에서 판매하는 음료수를 선택한다 | 3: Select Drink | void selectDVM(num) | |
| 원가 화면기에서 판매하지 않는 음료수를 선택한다 | 4: Select Payment | void displayDrinkList(drink_list) | |
| 결제 방법 중 하나를 선택한다 | 5: Insert Card | void selectDrink(dialNum) | |
| 카드 결제를 진행한다 | 6: Enter Code | void selectOtherDVMsDrink(dialNum) | |
| 인종코드 결제를 진행한다 | 7: Get Location | List<Drink> checkOtherDVMsStock(current_drink, cur_DVM_info) | |
| 자판기의 위치를 보여준다 | | void displayPaymentMenu() | DVM |
| | | void proceedRepayment() | |
| | | void selectPayment(num) | |
| | | void displayInvalidInput() | |
| | | void insertCard(card_num) | |
| | | void enterCode(code_num) | |
| | | Code getCodeInfo(code_num) | |
| | | List<Code> deleteCode(code_info) | |
| | | void outOfStockMsg() | |
| | | List<Code> addCode(code) | |
| | | void showDVMLocation(accessibleDVMList) | Message |
| | | void displayLocationList(locationList) | |
| | | void checkCodeAvailable(code_num) | |
| | | void displayInvalidMsg() | |
| | | Boolean requestDVMState(stateBroadcastMessage) | |
| | | List<Drink> getDrinkList(num) | |
| | | Drink getDrinkInfo(dialNum) | |
| | | void requestDrink(drink) | |
| | | void provideDrink() | |
| | | int getDrinkStock(drink_info, cur_DVM_info) | |
| | | Message requestStock(stockBroadcastMessage) | CardPayment |
| | | int getLocation(locReqMsg) | |
| | | Message createMessage(src_id, dst_id, msg_type) | |
| | | Message createMessage(src_id, dst_id, msg_type, msg) | |
| | | Boolean isCardAvailable(card_info) | |
| | | Card getCardInfo(card_num) | |
| | | void cardPayment(card_info, drink_info) | |
| | | Code generateCode(drink_info) | |
| | | Boolean isAvailable(card_info) | |
| | | int getBalance(card_info) | |
| | | void updateBalance(price) | Card |
| | | String getName() | |
| | | int getPrice() | |
| | | void updateStock(count) | |
| | | int getStock(drink_info) | |
| | | Drink getDrink(code_info) | |
| | | Code getCodeInfo() | |
| | | codePayment(code_info) | |
| | | | |
| | | | |

