

# DISTRIBUTED VENDING MACHINE (DVM)

## OOPT STAGE 2040 OOD



Project Team  
Team 2

Date  
2021.04.29

Team Members

201310513	황인우
201311255	최우석
201512265	박인우
201711306	박정현

# Index

**Activity 2041. Design Real Use Cases**

**Activity 2043. Define Interaction Diagrams**

**Activity 2044. Define Design Class Diagrams**

**Activity 2045. Refine System Architecture**

**Activity 2047. Traceability Analysis**

## Activity 2041. Design Real Use case

<b>Use Case</b>	1. Show Menu
<b>Actor</b>	System
<b>Purpose</b>	사용자에게 초기 선택 화면을 보여줍니다.
<b>Overview</b>	사용자에게 세가지 주요 기능 중 하나를 선택할 수 있는 초기 선택 메뉴를 보여줍니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose Item 3. Pay by Cash 4. Pay by Card
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (S) System이 상품 선택 활성화하기, 쿠폰 사용, Login 버튼을 Software-GUI 화면에 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	2.Choose Item
<b>Actor</b>	User
<b>Purpose</b>	사용자가 시스템에 등록된 상품 중 하나를 선택합니다.
<b>Overview</b>	사용자가 구입하고자 하는 시스템에 등록된 상품 중 하나를 선택합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	1. Show Menu 3. Pay by Cash 4. Pay by Card
<b>Pre-Requisites</b>	시스템에 상품이 하나 이상 등록되어 있어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 Software-GUI 화면의 상품 선택 활성화하기 버튼을 누릅니다. 2. (S) System이 아이템 리스트를 화면에 출력합니다. 3. (U) User가 Software-GUI에 출력되어 있는 상품 중 원하는 상품 하나를 선택하여 누릅니다. 4. (S) System이 User가 선택한 상품의 재고를 확인합니다. 5. (S) User가 선택한 상품의 재고가 있거나 본 Use Case의 A.7.3으로부터 넘어온 경우, System이 카드 결제, 현금 결제, 취소 버튼이 있는 결제 방식 선택 화면을 Software-GUI 화면에 출력합니다.

	<p>6. (U) User가 출력된 버튼 중 하나를 선택합니다.</p> <p>7. (S) User가 현금 결제 버튼을 선택하였을 경우, Pay by Cash로 넘어갑니다.</p>
<b>Alternative Courses of Events</b>	<p>A. 5.1: User가 선택한 상품의 재고가 없을 경우, 사용자에게 해당 상품은 재고가 없어 선결제를 통한 코드 발급만 가능하다는 메시지와 함께 다른 자판기의 위치 보기, 쿠폰 결제하기, 취소 버튼이 있는 결제 방식 선택 화면을 Software-GUI 화면에 출력합니다.</p> <p>A. 7.1: User가 카드 결제 버튼을 선택하였을 경우, Pay by Card로 넘어갑니다.</p> <p>A. 7.2: User가 다른 자판기의 위치 보기 버튼을 선택하였을 경우, Get None Item Location으로 넘어갑니다.</p> <p>A. 7.3: User가 쿠폰 결제하기 버튼을 선택하였을 경우, 본 Use Case의 5를 수행합니다.</p>
<b>Exceptional Courses of Events</b>	<p>E. 7.4: User가 취소 버튼을 선택하였을 경우 Show Menu로 돌아갑니다.</p>

<b>Use Case</b>	3. Pay by Cash
<b>Actor</b>	User
<b>Purpose</b>	User가 현금으로 결제하는 경우입니다
<b>Overview</b>	결제를 현금으로 진행하게 해줍니다
<b>Type</b>	Evident
<b>Cross Reference</b>	2. Choose Item 6. Calculate Price(Cash)
<b>Pre-Requisites</b>	System에 구매하려는 목록이 등록되어 있고 상품이 추가되어 있어야 합니다.
<b>Typical Courses of Events</b>	<p>(U): User, (S): System</p> <p>1. (U) User가 현금 결제 버튼을 선택합니다.</p> <p>2. (S) System이 현금 입력 창을 Software-GUI 화면에 출력합니다.</p> <p>3. (U) User가 지급할 현금을 Software-GUI 화면에 입력합니다.</p> <p>4. (S) System이 현금을 지급받으면 6. Calculate Price(Cash)로 넘어갑니다.</p>
<b>Alternative Courses of Events</b>	A 1.1. Calculate Price(Cash)의 Alternative Course 1에 의해 반환된 경우, 알림 메시지와 함께 2 를 실행합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	4. Pay by Card
<b>Actor</b>	User
<b>Purpose</b>	User 가 카드로 결제하려는 경우입니다.
<b>Overview</b>	결제를 카드로 진행해주게 합니다.

<b>Type</b>	Evident
<b>Cross Reference</b>	2.Choose Item 7.Calculate Price(Card)
<b>Pre-Requisites</b>	System에 구매하려는 목록이 등록되어 있고 상품이 추가되어 있어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 카드 결제 버튼을 누릅니다 2. (S) System이 카드 결제 창을 Software-GUI 화면에 출력합니다. 3. (U) User가 카드번호를 Software-GUI 화면에 입력합니다 4. (S) System에 카드번호가 입력되면 7. Calculate Price(Card)로 넘어갑니다.
<b>Alternative Courses of Events</b>	A 1.1. Calculate Price(Card)의 Alternative Course 1에 의해 반환된 경우, 알림 메시지와 함께 2를 실행합니다. A 3.1 User가 입력창을 비워두고 진행 하려 할 경우, 알림 메시지와 함께 2를 실행하여 카드번호를 받습니다
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	5. Use Code
<b>Actor</b>	User
<b>Purpose</b>	User가 코드를 사용하려는 경우입니다.
<b>Overview</b>	사용자가 코드를 입력하여 사용할 수 있도록 합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	1. Show Menu 8. Code Validation 10. Give code
<b>Pre-Requisites</b>	사용자는 이전에 Give Code로부터 제공받은 코드가 존재해야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) User가 쿠폰 사용 버튼을 누릅니다. 2. (S) System이 코드를 입력받을 창을 Software-GUI 화면에 출력합니다. 3. (U) User가 코드를 입력합니다 4. (S) System이 코드를 받으면 8. Code Validation으로 넘어갑니다.
<b>Alternative Courses of Events</b>	A 1.1. Code Validation의 Alternative Course 1.1, 2.1에 의해 반환된 경우, 알림 메시지와 함께 2를 실행합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	6. Calculate Price (Cash)
-----------------	---------------------------

<b>Actor</b>	System
<b>Purpose</b>	User가 현금으로 결제할시 결제를 진행하는 과정입니다
<b>Overview</b>	투입된 현금과 합산된 상품 값을 비교하여 결제를 진행합니다
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose Item 4. Pay by Cash 9. Choose Item or Code
<b>Pre-Requisites</b>	Pay by Cash에서 현금을 지급받은 상태여야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 투입된 현금의 합과 상품의 가격을 비교합니다. 2. (S) 투입된 현금의 합이 상품의 가격과 같거나 높을 경우 결제를 진행합니다. 3. (S) 현금의 합이 상품의 가격보다 높거나 같고 결제가 성공적이면 자판기가 지급해야 할 거스름돈을 산출한 뒤 자판기가 보유한 현금과 비교합니다. 4. (S) 자판기가 지급해야 할 거스름돈이 자판기가 보유한 현금보다 적거나 같을 경우 <b>Choose Item or Code</b> 으로 넘어갑니다.
<b>Alternative Courses of Events</b>	A 2.1: 현금의 합이 상품의 가격보다 적을 경우 Pay by Cash로 돌아가 추가 현금을 받습니다.
<b>Exceptional Courses of Events</b>	E 4.1: 자판기가 지급해야 할 거스름돈이 자판기가 보유한 현금보다 많을 경우 지급할 거스름돈이 부족하다는 메시지를 Software-GUI 화면에 출력한 뒤 과정을 종료합니다.

<b>Use Case</b>	7. Calculate Price (Card)
<b>Actor</b>	System
<b>Purpose</b>	User가 카드로 결제할시 결제를 진행하는 과정입니다
<b>Overview</b>	카드의 지불능력과 합산된 상품 값을 비교하여 결제를 진행합니다
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose Item 4. Pay by Card 9. Choose Item or Code
<b>Pre-Requisites</b>	Pay by Card에서 카드번호가 입력되어 있어야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 카드의 유효성을 체크합니다. 2. (S) 카드가 유효할 경우 카드의 잔액과 상품의 가격을 비교합니다 3. (S) 카드의 잔액이 상품의 가격보다 높거나 같을 경우 결제를 진행합니다 4. (S) 결제가 성공적이면, <b>Choose Item or Code</b> 로 넘어갑니다
<b>Alternative Courses of Events</b>	A1.1: 카드가 유효하지 않을 경우 카드가 유효하지 않다는 메시지를 Software-GUI 화면에 출력하고 Pay by Card로 돌아가 다른 카드를 투입 받습니다.

	A2.1: 카드의 잔액이 상품의 가격보다 낮을 경우 잔액이 부족하다는 메시지를 Software-GUI 화면에 출력하고 Pay by Card로 돌아가 다른 카드를 투입 받습니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	8. Code Validation
<b>Actor</b>	System
<b>Purpose</b>	코드의 유효여부를 확인합니다.
<b>Overview</b>	사용자가 입력한 코드의 유효성 및 해당 자판기의 재고 여부를 확인합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	2. Choose Item 5. Use Code 11. Give Item
<b>Pre-Requisites</b>	Use Code에서 사용자가 코드를 입력한 상태여야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 코드의 유효성을 모든 DVM에서 체크합니다. 2. (S) 유효한 코드일 경우 코드로부터 확인된 상품의 재고가 해당 시스템에 존재하는지 확인합니다. 3. (S) 상품의 재고가 있을 경우, 해당 코드를 사용한 것으로 처리하고 해당 정보를 다른 시스템에게도 알립니다. 4. (S) Give Item으로 넘어갑니다
<b>Alternative Courses of Events</b>	A2.1: 유효하지 않은 코드일 경우, 유효하지 않은 코드라는 메시지를 Software-GUI 화면에 출력하고 Use Code로 돌아가 다시 코드를 입력하도록 합니다.
<b>Exceptional Courses of Events</b>	E3.1: 상품의 재고가 없을 경우 해당 코드의 상품의 재고가 없다는 메시지를 Software-GUI 화면에 출력하고 과정을 종료합니다.

<b>Use Case</b>	9. Choose Item or Code
<b>Actor</b>	System
<b>Purpose</b>	시스템이 사용자가 재고가 있는 상품을 선택했는지 재고가 없는 상품을 선택했는지 확인합니다.
<b>Overview</b>	시스템이 사용자가 재고가 있는 상품을 선택했는지 재고가 없는 상품을 선택했는지 확인하고 이에 따라 Give Item 혹은 Give Code를 진행합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	3.Pay By Cash 4.Pay By Card

<b>Pre-Requisites</b>	Pay By Cash, Pay By Card 중 하나의 과정에서 결제가 마무리된 상황이어야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (S) 시스템이 사용자가 재고가 있는 상품을 선택했는지, 재고가 없는 상품을 선택했는지 확인합니다. 2. (S) 재고가 있는 상품을 선택했을 경우 Give Item 과정을 진행하고, 재고가 없는 상품을 선택했을 경우 Give Code 과정을 진행합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	10.Give Code
<b>Actor</b>	System
<b>Purpose</b>	사용자가 선택한 상품에 대해 코드를 제공합니다.
<b>Overview</b>	사용자가 선택한 상품을 구입할 수 있는 코드를 제공합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	6. Calculate Price(Cash) 7. Calculate Price(Card) 9. Choose Item or Code
<b>Pre-Requisites</b>	Choose item or code에서 code을 선택해야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 시스템이 사용자가 선택한 상품에 대한 코드를 생성합니다. 2. (S) 시스템이 다른 시스템에 해당 코드 정보를 공유합니다. 3. (S) 코드를 화면에 출력합니다.
<b>Alternative Courses of Events</b>	A3.1 (S) 만약 6. Calculate Price(Cash) 의 A3.1 에서 거스름돈이 있으면, 거스름돈도 같이 사용자에게 반환(Software-GUI 화면에 출력)합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	11.Give Item
<b>Actor</b>	System
<b>Purpose</b>	사용자가 선택한 상품을 제공합니다.
<b>Overview</b>	사용자가 선택한 상품을 제공하고, 제공한 수량 만큼을 시스템 내부 재고량 에서 뺍니다.
<b>Type</b>	Hidden



<b>Cross Reference</b>	2. Choose Item 6. Calculate Price(Cash) 8. Code Validation 9. Choose Item or Code
<b>Pre-Requisites</b>	Choose Item or Code에서 Item을 선택하거나 사용자가 코드를 사용하고 해당 코드가 Code Validation 과정을 통과해야 합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 시스템이 사용자가 선택한 상품을 반환(Software-GUI 화면에 출력)합니다. 2. (S) 시스템이 사용자에게 제공한 상품의 재고를 1 감소시킵니다.
<b>Alternative Courses of Events</b>	A1.1 (S) 만약 6. Calculate Price(Cash) 의 A3.1 에서 거스름돈이 있으면, 거스름돈도 같이 사용자에게 반환(Software-GUI 화면에 출력)합니다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	12. Get None Item Location
<b>Actor</b>	User
<b>Purpose</b>	사용자가 원하는 재고가 없는 상품을 선택합니다.
<b>Overview</b>	사용자가 위치정보를 원하는, 재고가 없는 상품을 선택하도록 합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	2. Choose Item 13. Get Another DVM Info
<b>Pre-Requisites</b>	재고가 없는 상품이 존재해야 합니다.
<b>Typical Courses of Events</b>	(U): User, (S): System 1. (U) 사용자가 다른 자판기의 위치 보기 버튼을 누릅니다. 2. (S) Get Another DVM Info를 실행합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	13. Get Another DVM Info
<b>Actor</b>	System
<b>Purpose</b>	시스템이 다른 시스템에 정보(재고현황, 위치)를 요청합니다.

<b>Overview</b>	사용자가 요청한 재고가 없는 상품에 대한 정보를 얻기 위해 다른 시스템에 정보(재고현황, 위치)를 요청합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	12. Get None Item Location 14. Give DVM Info 15. Give Item Location
<b>Pre-Requisites</b>	User에게 Get None Item Location으로 요청을 받아야합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) 사용자가 선택한 물품의 정보를 다른 시스템 전체에 전달하고 해당 시스템에서의 재고 현황과 해당 시스템의 위치를 요청합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	14. Give DVM Info
<b>Actor</b>	Another DVM
<b>Purpose</b>	타 DVM이 현 DVM에게 요청한 재고의 유무를 알려줍니다.
<b>Overview</b>	현 DVM에게 정보(해당 상품 재고, 타 DVM 위치)를 반환합니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	13. Get Another DVM Info 15. Give Item Location
<b>Pre-Requisites</b>	타 DVM에게 Get Another DVM Info로 요청을 받아야 합니다
<b>Typical Courses of Events</b>	(S): System, (D): Another DVM 1. (D) System에게 재고와 위치를 요청합니다 2. (S) 본 기기의 해당 재고 유무를 파악합니다. 3. (S) 기기의 재고와 위치를 Another DVM에게 반환합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	15. Give Item Location
<b>Actor</b>	System
<b>Purpose</b>	사용자에게 상품이 존재하는 다른 DVM의 정보(위치, 재고현황)을 알려줍니다.

<b>Overview</b>	사용자가 요청한 상품이 존재하는 다른 DVM의 정보(위치, 재고현황)을 알려줍니다.
<b>Type</b>	Hidden
<b>Cross Reference</b>	12. Get None Item Location 13. Get Another DVM Info 14. Give DVM Info
<b>Pre-Requisites</b>	Get None Item Location으로 요청된 상품이 존재해야 합니다. Another DVM들의 Give DVM Info로 재고정보 및 위치정보를 받아야합니다.
<b>Typical Courses of Events</b>	(S): System 1. (S) Another DVM의 Give DVM Info로 전달받은 정보를 기반으로 재고가 존재하는 DVM을 구분합니다. 2. (S) 구분된 DVM들을 현재 시스템에서 가까운 순서대로 정렬합니다. 3. (S) 정렬된 DVM 위치의 목록을 Software-GUI 화면에 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E1.1: 재고가 있는 DVM이 없을 경우 재고가 존재하는 자판기가 없다는 내용을 Software-GUI 화면에 출력합니다.

<b>Use Case</b>	16. Login
<b>Actor</b>	Administrator
<b>Purpose</b>	시스템 관리자가 관리자임을 확인해 관련 작업을 수행할 수 있도록 합니다.
<b>Overview</b>	시스템 관리자가 아이디와 비밀번호를 입력하고 이를 확인하여 관리자 권한을 부여합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	17. Manage Amount 18. Manage Amount Cash
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A): Administrator, (S) : System 1. (A) 관리자가 Software-GUI 화면의 Login 버튼을 누릅니다. 2. (S) 시스템이 로그인 Software-GUI를 띄웁니다 3. (A) 관리자가 Software-GUI 화면의 아이디와 비밀번호를 입력합니다. 4. (S) 시스템이 등록된 아이디와 비밀번호와 비교하고 일치할 경우 관리자 메뉴(상품재고 관리, 현금 관리)를 Software-GUI 화면에 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	E 3.1: 아이디와 비밀번호가 일치하지 않을 경우 아이디 혹은 비밀번호가 일치하지 않다는 메시지를 Software-GUI 화면에 출력하고

	과정을 종료합니다.
--	------------

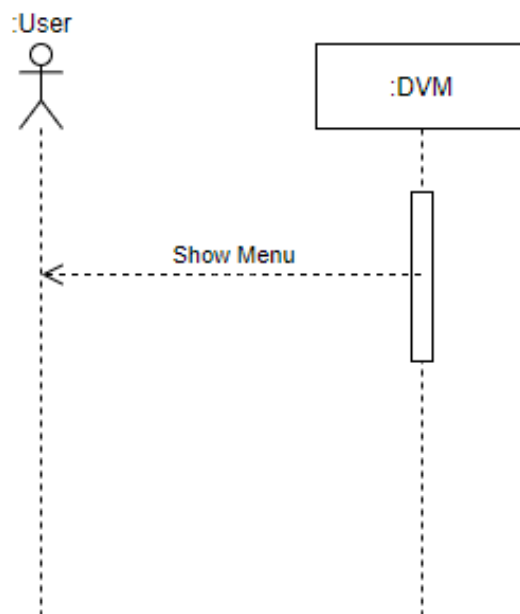
<b>Use Case</b>	17. Manage Amount
<b>Actor</b>	Administrator
<b>Purpose</b>	관리자가 상품의 재고를 관리합니다.
<b>Overview</b>	관리자가 상품의 재고를 수정합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	16. Login
<b>Pre-Requisites</b>	Login을 통해 관리자 권한을 얻은 상태여야 합니다. 재고를 추가하려는 상품이 시스템에 등록되어 있어야 합니다.
<b>Typical Courses of Events</b>	(A): Administrator, (S) : System 1. (A) 관리자가 상품재고 관리 버튼을 누릅니다. 2. (S) 시스템이 자판기가 보유한 모든 상품과 재고를 Software-GUI 화면에 출력합니다. 3. (A) 관리자가 상품을 선택하고 수정할 재고량을 입력합니다. 4. (S) 시스템이 입력된 정보를 기반으로 상품 재고량을 수정하고 수정이 완료되었다는 내용을 Software-GUI 화면에 출력합니다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	18. Manage Amount Cash
<b>Actor</b>	Administrator
<b>Purpose</b>	관리자가 현금의 재고를 관리합니다.
<b>Overview</b>	관리자가 현금의 재고를 수정합니다.
<b>Type</b>	Evident
<b>Cross Reference</b>	16. Login
<b>Pre-Requisites</b>	Login을 통해 관리자 권한을 얻은 상태여야 합니다.
<b>Typical Courses of Events</b>	(A): Administrator, (S): System 1. (A) 관리자가 현금 관리 버튼을 누릅니다. 2. (S) 시스템이 자판기가 보유한 현금 양을 Software-GUI 화면에 출력합니다. 3. (A) 관리자가 현금 양을 입력합니다. 4. (S) 시스템이 입력된 정보를 기반으로 현 DVM의 현금 재고량을 수정하고 수정이 완료되었다는 내용을 Software-GUI 화면에 출력합니다.

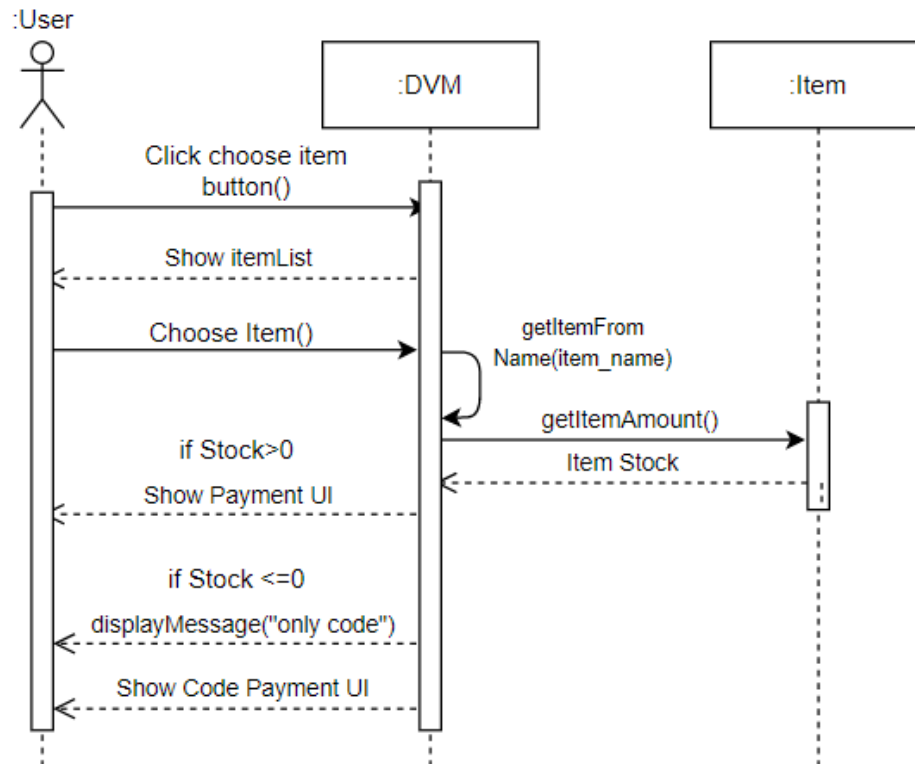
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## Activity 2043. Define Interaction Diagram

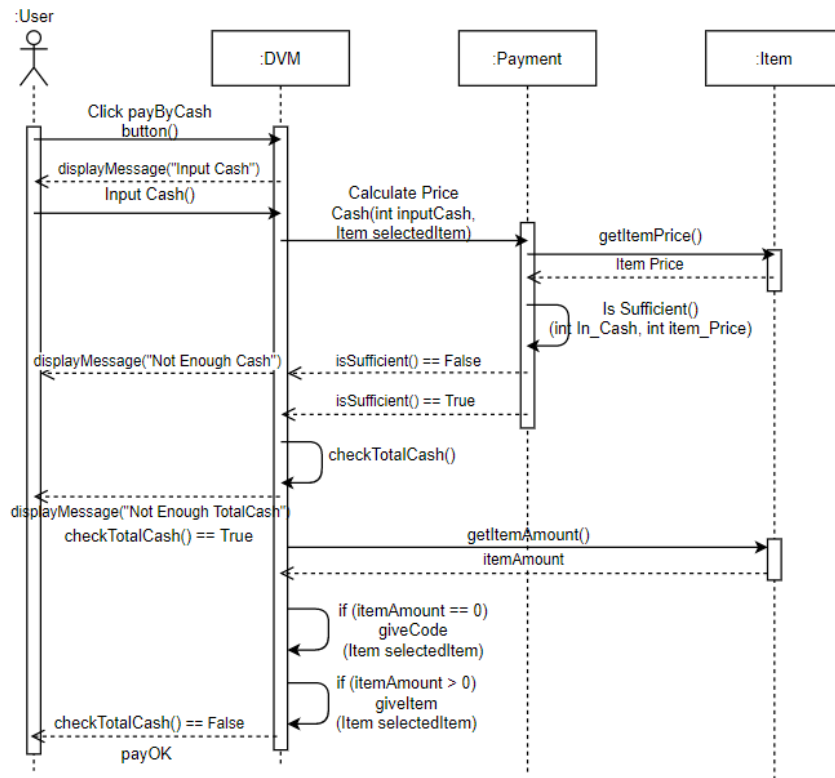
### 1. Show menu



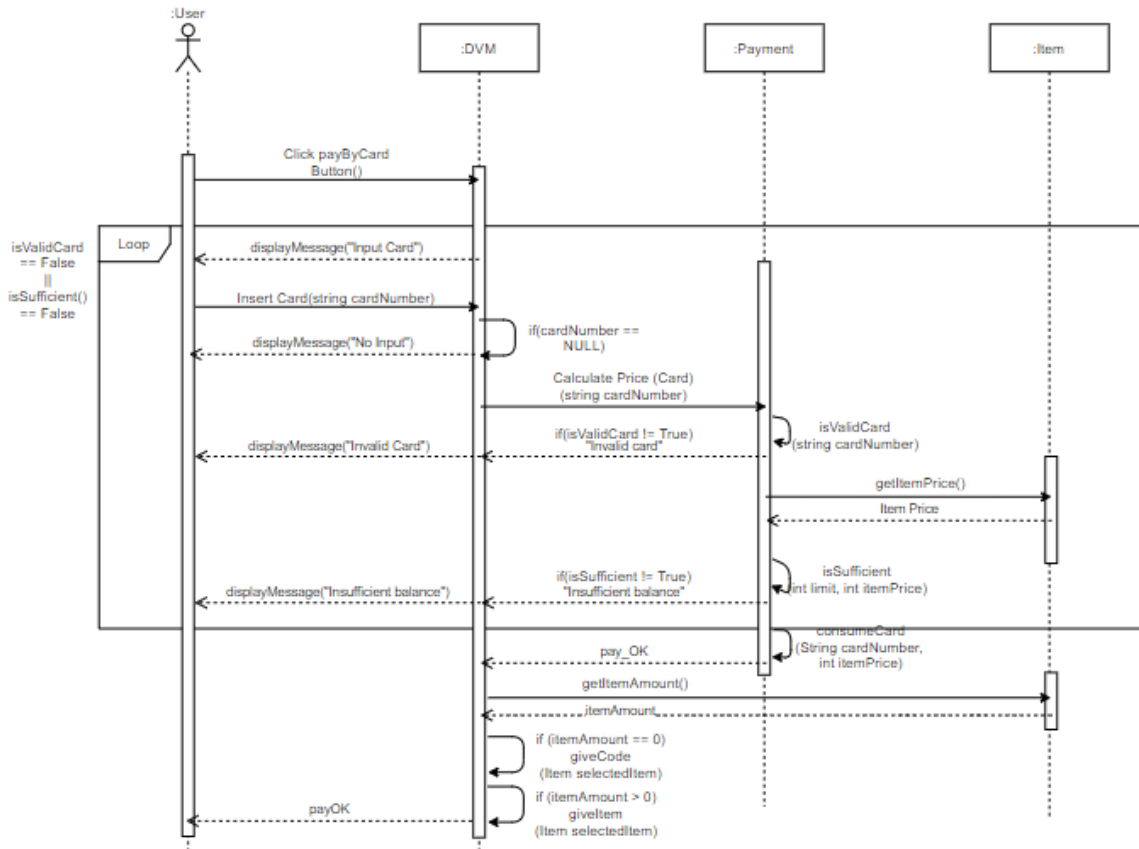
### 2. Choose Item



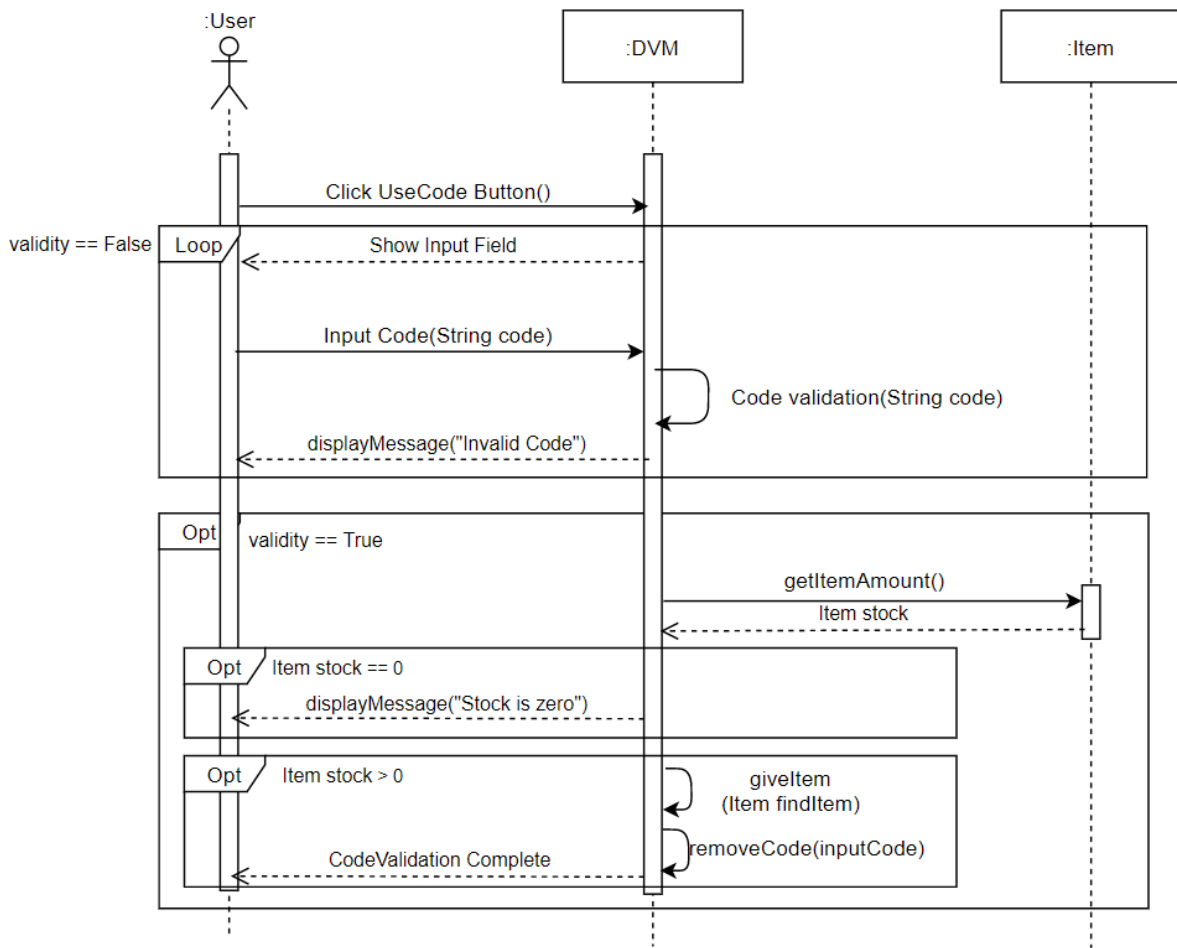
### 3. Pay by Cash & 6. Calculate Price Cash



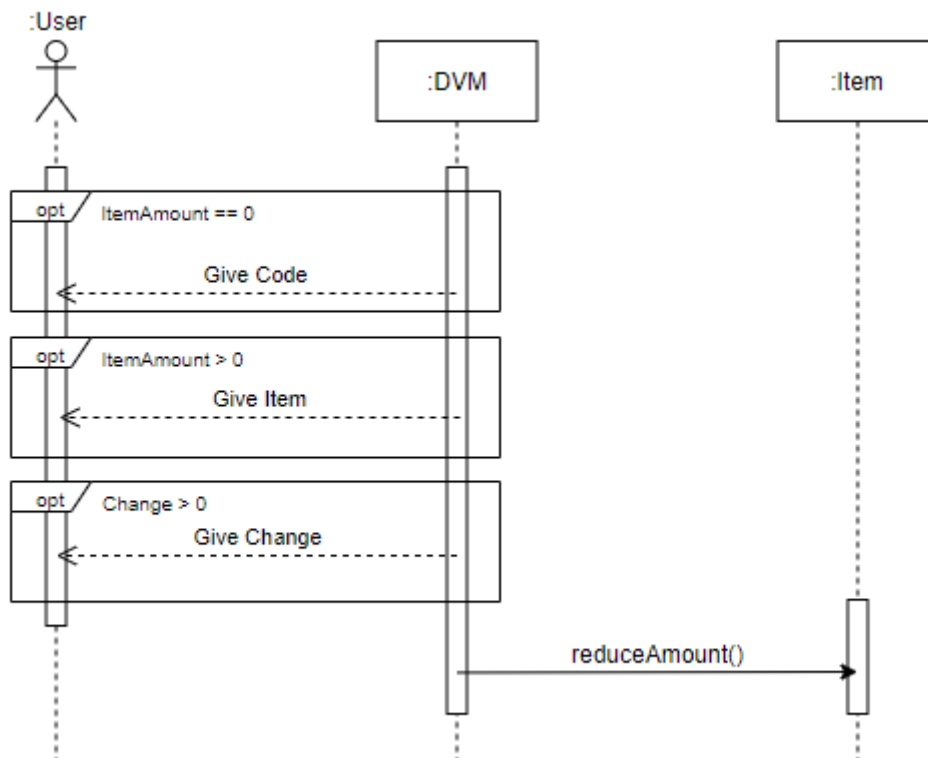
### 4. Pay by Card & 7. Calculate Price Card



5. Use Code & 8. Code validation

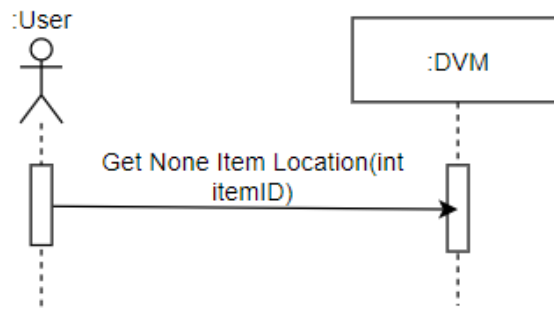


10. Give Code & 11. Give Item

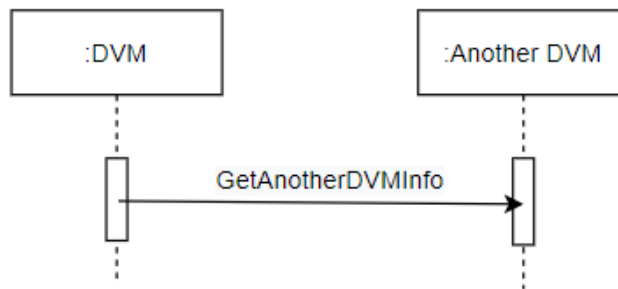




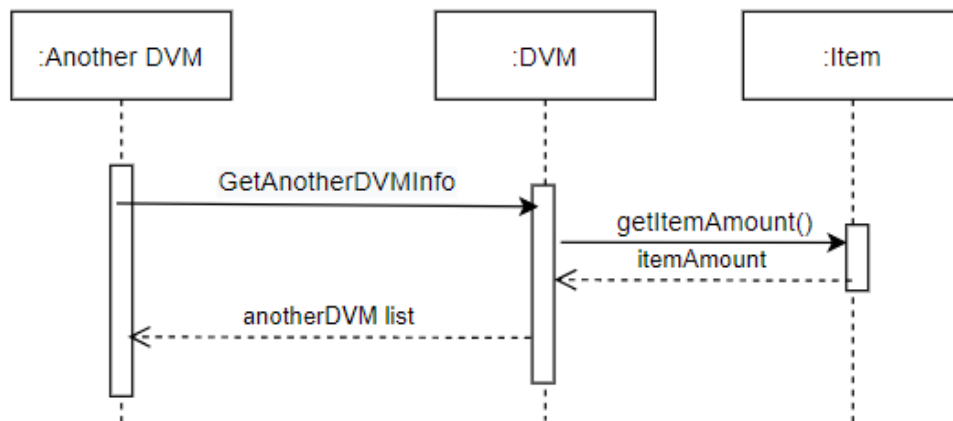
12. Get None Item Location



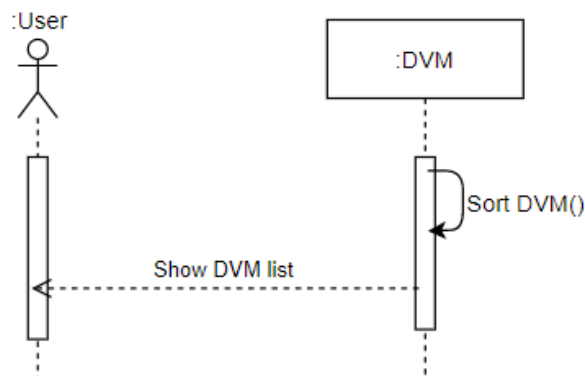
13. Get Another DVM Info



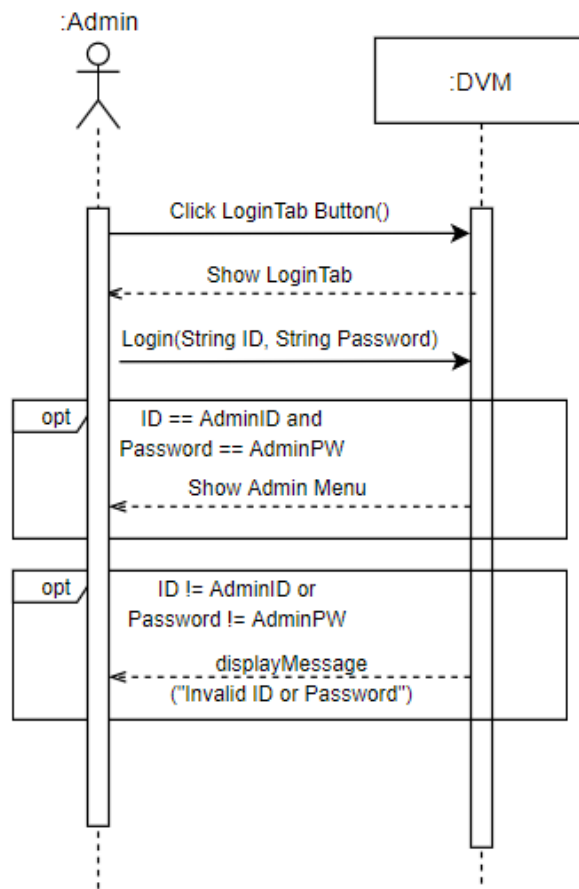
14. Give DVM Info



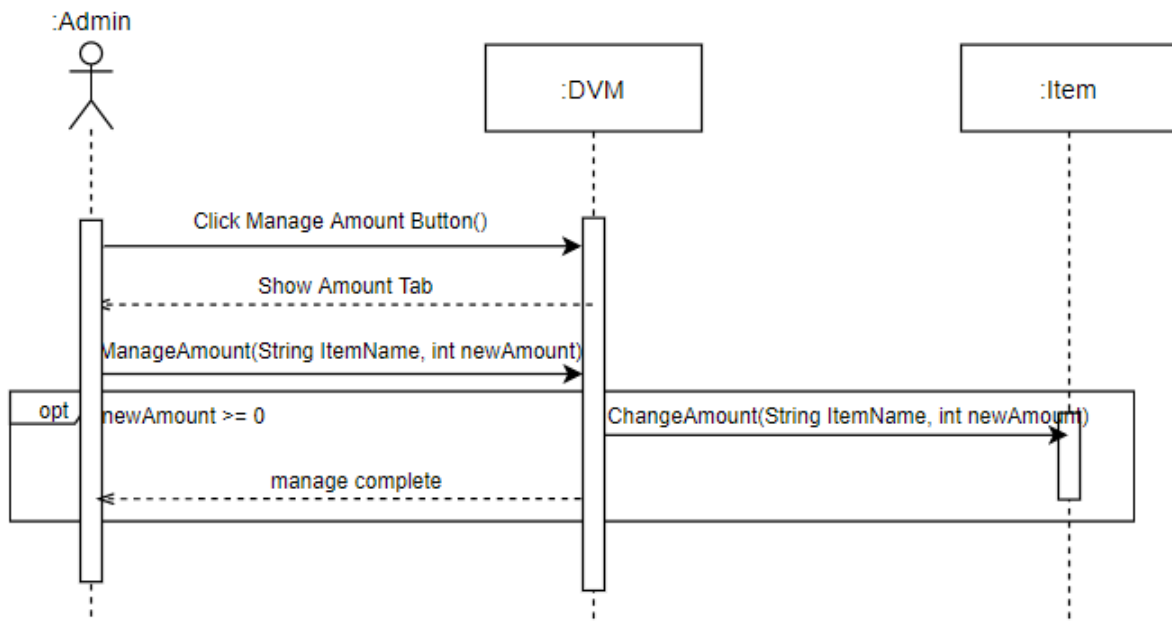
15. Give Item Location



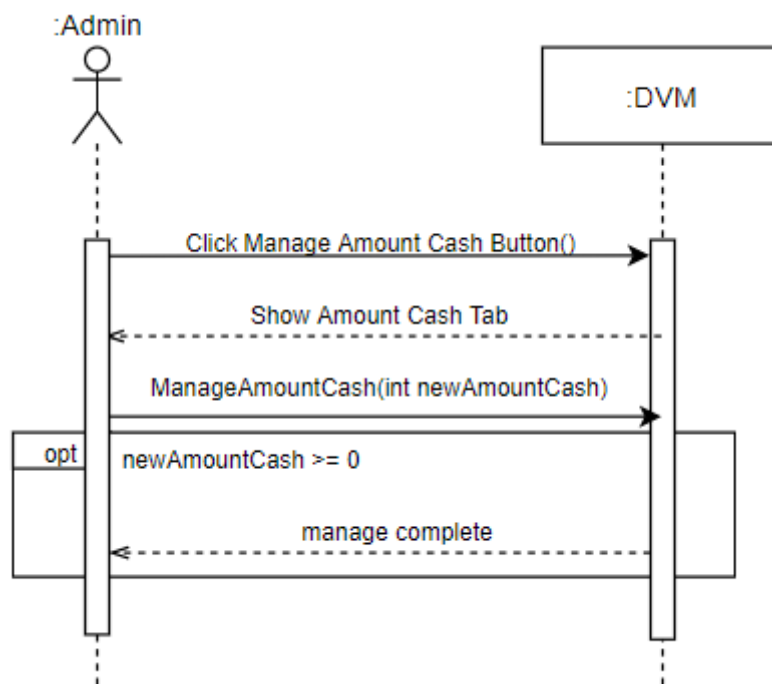
## 16. Login



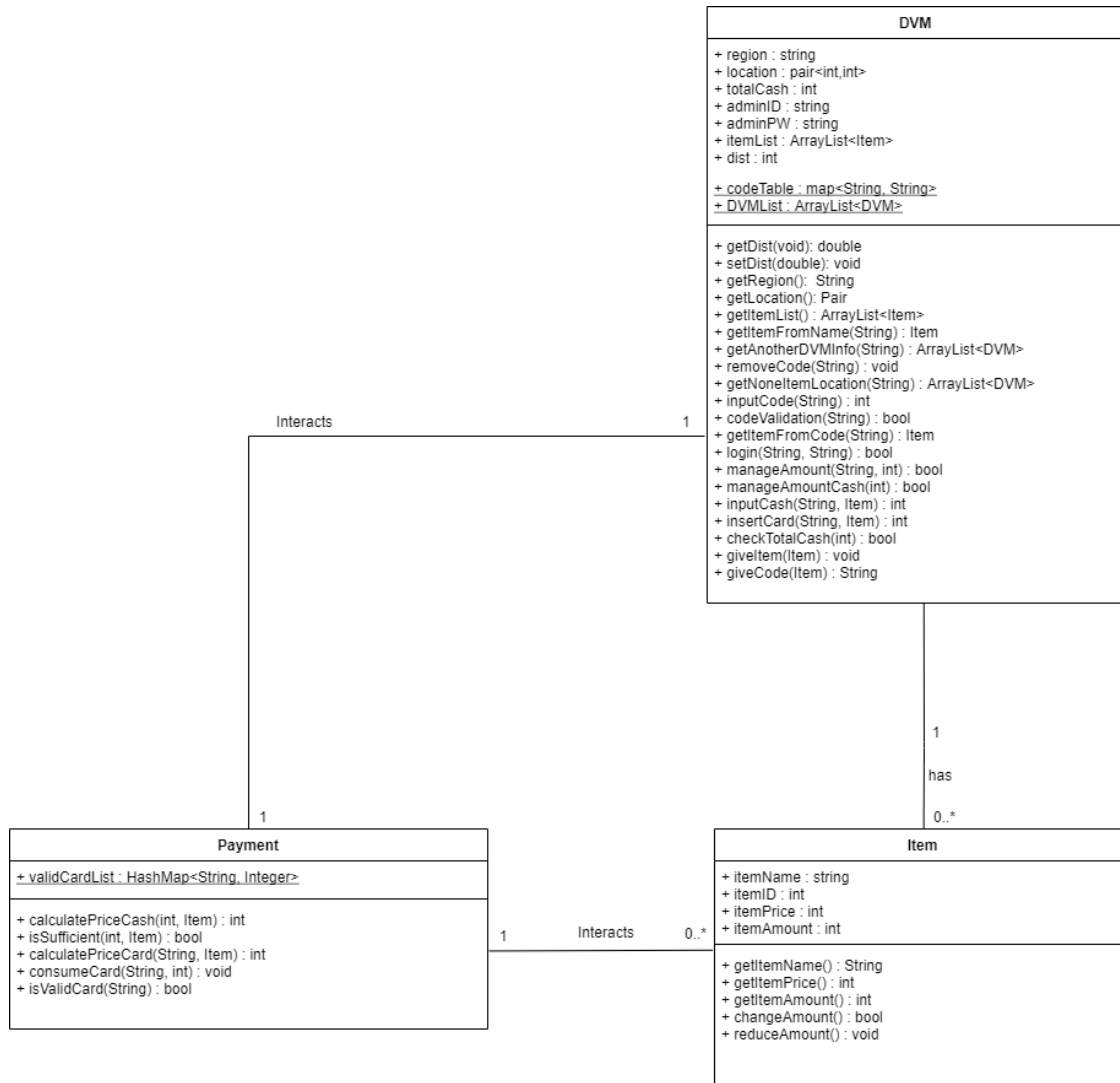
## 17. Manage Amount



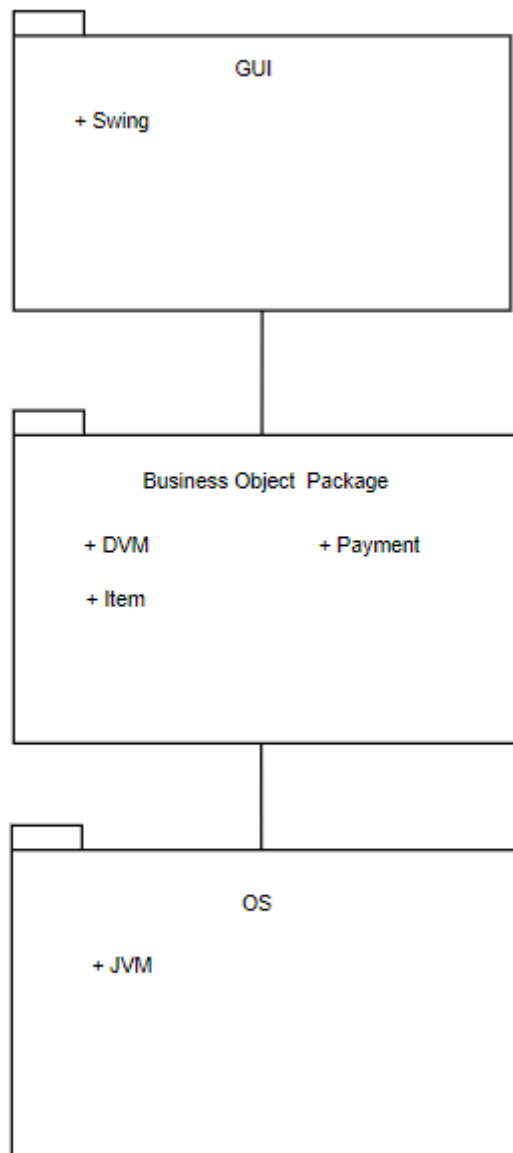
## 18. Manage Amount Cash



## Activity 2044. Define Design Class Diagrams



## Activity 2045. Refine System Architecture



## Activity 2047. Traceability Analysis

Operations in System Sequence Diagrams	Matching Operations in Interaction Diagrams
1.Activate Choices	Click choose item button()
2.Choose Item	Choose Item()
	getItemAmount()
3.Pay By Cash	Click payByCash button()
4.Input Cash	InputCash()
	Calculate Price Cash()
	getItemPrice()
	reduceAmount()
5.Pay By Credit	Click PayByCard button()
6.Insert Card	getItemPrice()
	Insert Card()
	Calculate Price Card()
	reduce Amount()
7.Use code	Click UseCode button()
8.Input Code	Input code()
	Code validation()
	reduce Amount()
9.GetNoneltemLocation	getNoneltemLocation()
10.GiveDVMInfo	get AnotherDVMInfo()
11.Activate Login Tab	Click LoginTab button()
12.Login	Login()
13.Activate Manage Amount	Click ManageAmount Button()
14.Manage Amount	Manage Amount()
	Change Amount()

15. Activate Manage Cash	Click Manage Amount Cash button()
16. Manage Cash	Manage Amount Cash()

Operations in Interaction Diagrams	Matching Methods	Matching Classes
Click Choose Item Button()	getItemAmount()	Item
Choose Item()	getItemList()	DVM
	getItemFromName(String)	
	giveItem(Item)	
	giveCode(Item)	
	getItemName()	Item
getItemAmount()	getItemAmount()	
Click payByCash Button()	inputCash(String, Item)	DVM
Input Cash()		
Calculate Price Cash()		
Calculate Price Cash()	checkTotalCash(int)	Payment
	calculatePriceCash(int, Item)	
	isSufficient(int, Item)	
getItemPrice()	getItemPrice()	Item
Click payByCard Button()	insertCard(String, Item)	DVM
Insert Card()		
Calculate Price Card()	isSufficient(int, Item)	Payment
	calculatePriceCard(String, Item)	
	consumeCard(String, int)	
	isValidCard(String)	
Click UseCode Button()	inputCode(String)	DVM
Input Code()		
Code Validation()		
Code Validation()	removeCode(String)	DVM
	codeValidation(String)	
	getItemFromCode(String)	
reduceAmount()	reduceAmount()	Item

getNonItemLocation()	getDist(void)	DVM
	setDist(double)	
	getRegion()	
	getLocation()	
	getNonItemLocation(String)	
getAnotherDVMInfo()	getAnotherDVMInfo(String)	DVM
Click Login Tab Button()	login(String, String)	
Login()		
Click Manage Amount Button()	manageAmount(String, int)	
ManageAmount()		
ChangeAmount()	changeAmount	Item
Click Manage Amount Cash Button()	manageAmountCash(int)	DVM
Manage Amount Cash()		