

# Software Design Specification for Distributed Vending Machine

---

Team 2

201810286 박혜린

201811264 변장훈

201311312 정인욱

<b>2141. Design Real Use Cases</b>	<b>3</b>
<b>2143. Define Interaction Diagrams</b>	<b>13</b>
1. Use Case 1 - Card Payment	14
2. Use Case 2 - Card Payment(Prepay)	14
3. Use Case 3 - Manage Stock	15
4. Use Case 4 - Input Admin Code	15
5. Use Case 5 - Set DVM Setting	16
6. Use Case 6 - Select Menu	16
7. Use Case 7 - Input Verification Code	17
8. Use Case 8 - Select Other DVM	17
9. Use Case 9 - Respond VM On	18
10. Use Case 10 - Request VM Off	18
11. Use Case 11 - Respond Vm Off	19
12. Use Case 12 - Receive Operating	19
13. Use Case 13 - Receive Stock	19
14. Use Case 14 - Respond Refund	20
15. Use Case 15 - Receive Verification Code	20
16. Use Case 16 - Respond Verification Code	21
17. Use Case 17 - Respond Stock	22
<b>2144. Define Design Class Diagrams</b>	<b>23</b>
<b>2145. Refine System Architecture</b>	<b>24</b>
1. Deployment Diagram	24
2. Package Diagram	25
<b>2147. Perform 2040 Traceability Analysis</b>	<b>26</b>

## 2141. Design Real Use Cases

<b>Use Case</b>	1. Card Payment
<b>Actor</b>	User
<b>Purpose</b>	카드 결제를 시행한다.
<b>Overview</b>	사용자가 음료 구매를 결정하고 해당 음료의 재고가 1 이상인 경우 사용자로부터 카드 번호를 입력받아 결제를 진행한후 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function : R 1.1
<b>Pre-Requisites</b>	사용자가 선택한 음료의 재고가 0초과이며 자판기가 카드 결제대기 상태여야 한다.
<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>1. (<math>A_U</math>) 사용자가 카드 번호를 입력한다.</li> <li>2. (S) 카드가 유효한 상태(bool)를 체크한다.</li> <li>3. (S) 카드의 잔액을 확인한다.</li> <li>4. (S) 사용자가 선택한 음료의 가격을 카드 잔액에서 차감한다.</li> <li>5. (S) 해당 음료의 재고를 1 감소한다.</li> <li>6. (S) 사용자가 선택한 음료를 제공한다.</li> <li>7. (S) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
<b>Alternative Courses of Events</b>	<ol style="list-style-type: none"> <li>1a. (<math>A_U</math>) 사용자가 취소를 선택한다.</li> <li>2a. (S) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
	<ol style="list-style-type: none"> <li>2b. (S) 사용자가 입력한 카드가 유효한 카드가 아니라면 에러 메시지를 출력한다.</li> <li>3b. (S) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
	<ol style="list-style-type: none"> <li>3c. (S) 사용자가 입력한 카드의 잔액이 부족하다면 에러 메시지를 출력한다.</li> <li>4c. (S) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	2. Card Payment (Prepay)
<b>Actor</b>	User
<b>Purpose</b>	카드 결제를 시행한다.
<b>Overview</b>	사용자가 음료 선결제를 결정하면 사용자로부터 카드 번호를 입력받아 결제를 진행한 후 인증코드 생성을 요청한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function : R 1.2
<b>Pre-Requisites</b>	사용자가 선택한 음료의 재고가 0이며 자판기가 카드 결제대기 상태여야 한다.
<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>(<math>A_U</math>) 사용자가 카드 번호를 입력한다.</li> <li>(S) 카드의 유효성을 검사한다.</li> <li>(S) 카드의 잔액을 확인한다.</li> <li>(S) 사용자가 선택한 음료의 가격을 카드 잔액에서 차감한다.</li> <li>(S) 다른 자판기에 인증코드 생성을 요청한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	1a. ( $A_U$ ) 사용자가 취소를 선택한다.
	2a. (S) 메뉴 선택 대기 상태로 전환한다.
	2b. (S) 사용자가 입력한 카드가 유효한 카드가 아니라면 에러 메시지를 출력한다.
	3b. (S) 메뉴 선택 대기 상태로 전환한다.
	3c. (S) 사용자가 입력한 카드의 잔액이 부족하다면 에러 메시지를 출력한다.
	4c. (S) 메뉴 선택 대기 상태로 전환한다.

<b>Use Case</b>	3. Manage Stock
<b>Actor</b>	Admin
<b>Purpose</b>	재고를 업데이트한다.
<b>Overview</b>	관리자가 해당 기능을 실행하면 Stock Counter가 재고를 측정하여 그 값을 자판기가 입력받아 수량에 반영한 후 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function : R 2.1

<b>Pre-Requisites</b>	자판기가 관리자 모드 상태여야한다.
<b>Typical Courses of Events</b>	( $A_U$ ): User, ( $A_M$ ): Admin, ( $A_D$ ): Other DVM, (S): System 1. ( $A_M$ ) 관리자가 재고 업데이트 기능을 실행한다. 2. (S) Stock Counter가 재고를 측정하여 그 값을 자판기의 재고 수량에 반영한다. 3. (S) 메뉴 선택 대기 상태로 전환한다.
<b>Alternative Courses of Events</b>	1a. ( $A_M$ ) 관리자가 취소를 선택한다. 2a. (S) 메뉴 선택 대기 상태로 전환한다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	4. Input Admin Code
<b>Actor</b>	Admin
<b>Purpose</b>	관리자 모드로 전환한다.
<b>Overview</b>	관리자가 인증코드 입력 화면에서 관리자 코드를 입력하면 자판기가 관리자 모드로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function : R 2.2
<b>Pre-Requisites</b>	인증코드 입력 대기상태
<b>Typical Courses of Events</b>	( $A_U$ ): User, ( $A_M$ ): Admin, ( $A_D$ ): Other DVM, (S): System 1. ( $A_M$ ) 관리자가 관리자 코드를 입력한다. 2. (S) 관리자 코드와 일치하는지 확인한다. 3. (S) 관리자 모드로 전환한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	2a. (S) 관리자 코드와 일치하지 않는다면 유효하지 않은 인증코드라는 오류 문구를 출력한다. 3a. (S) 메뉴 선택 대기 상태로 전환한다.

<b>Use Case</b>	5.Set DVM Setting
<b>Actor</b>	Admin
<b>Purpose</b>	관리자가 자판기 정보를 설정한 후 자판기를 시작한다.

<b>Overview</b>	관리자가 자판기 ID, 자판기의 주소, 관리자 코드를 파일에 작성하고 자판기를 시작하면 해당 파일을 DVM이 읽고 저장한 후 1~10까지의 자판기들 중 현재 자판기를 제외한 모든 자판기에 시작 메시지를 전송한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 2.3
<b>Pre-Requisites</b>	자판기가 Power off인 상태
<b>Typical Courses of Events</b>	(A <sub>U</sub> ): User, (A <sub>M</sub> ): Admin, (A <sub>D</sub> ): Other DVM, (S): System 1. (A <sub>M</sub> ) 관리자가 자판기 ID, 자판기의 주소, 관리자 코드를 파일에 작성하고 자판기를 시작한다. 2. (S) 입력한 정보의 유효성(ID는 1 이상 10 이하의 자연수, 자판기 주소는 길이가 1 이상 10이하인 String, 관리자 코드는 6자리의 숫자로만 구성된 String)을 검사한다. 3. (S) Broadcast로 모든 자판기에게 ‘자판기 시작 메시지’를 전송한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	2a. 입력한 정보가 유효하지 않다면 자판기를 종료한다.

<b>Use Case</b>	6. Select Menu
<b>Actor</b>	User
<b>Purpose</b>	사용자가 메뉴를 선택한다.
<b>Overview</b>	사용자가 선택한 메뉴의 재고가 있을 경우 카드 결제를 요구하고, 재고가 없을 경우 다른 자판기에게 재고를 확인한다. 사용자가 인증코드 입력을 선택하면 인증코드 입력 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 1.1 , R 3.1 , R 3.2
<b>Pre-Requisites</b>	자판기가 메뉴 선택 대기 상태여야 한다.
<b>Typical Courses of Events</b>	(A <sub>U</sub> ): User, (A <sub>M</sub> ): Admin, (A <sub>D</sub> ): Other DVM, (S): System 1. (A <sub>U</sub> ) 사용자가 메뉴를 선택한다. 2. (S) 음료의 재고를 확인한다. 3. (S) 카드 결제 대기 상태로 전환한다.

	4. (S) Card Payment을 실행한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	3a. (S) 재고가 0인 경우 VMList의 모든 다른 자판기에게 ‘재고 여부 확인 메시지’를 전송한다. 1b. (A <sub>U</sub> ) 사용자가 “인증코드 입력으로 전환”을 선택한다. 2b. (S) 인증코드 입력 대기 상태로 전환한다.

<b>Use Case</b>	7. Input Verification Code
<b>Actor</b>	User
<b>Purpose</b>	사용자가 인증코드를 입력한다.
<b>Overview</b>	사용자가 다른 자판기에서 선결제를 통해 받은 인증코드를 입력하면, 해당 인증코드의 유효성을 검사한 후 해당 메뉴를 사용자에게 제공한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 3.2
<b>Pre-Requisites</b>	자판기가 인증코드 입력 대기 상태여야 한다.
<b>Typical Courses of Events</b>	(A <sub>U</sub> ): User, (A <sub>M</sub> ): Admin, (A <sub>D</sub> ): Other DVM, (S): System 1. (A <sub>U</sub> ) 사용자가 인증코드를 입력한다. 2. (S) 유효한 인증코드인지 검사한다. 3. (S) 사용자에게 해당 음료를 제공한다. 4. (S) 메뉴 선택 대기 상태로 전환한다.
<b>Alternative Courses of Events</b>	3a. (S) 유효한 인증 코드가 아니라면 에러 메시지를 출력한다. 4a. (S) 메뉴 선택 대기 상태로 전환한다.
	1b. (A <sub>U</sub> ) 사용자가 취소를 선택한다. 2b. (S) 메뉴 선택 대기 상태로 전환한다.
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	8. Select Other DVM
<b>Actor</b>	User
<b>Purpose</b>	사용자가 선결제를 할 다른 자판기를 선택한다.

<b>Overview</b>	사용자가 재고가 있는 다른 자판기의 목록에서 선결제를 진행할 자판기를 선택하여 카드 결제 대기 상태로 가거나 취소를 선택하여 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 1.2 , R 3.3
<b>Pre-Requisites</b>	자판기가 다른 자판기 선택 대기 상태여야 한다.
<b>Typical Courses of Events</b>	(A <sub>U</sub> ): User, (A <sub>M</sub> ): Admin, (A <sub>D</sub> ): Other DVM, (S): System 1. (A <sub>U</sub> ) 사용자가 선결제를 할 다른 자판기를 선택한다. 2. (S) 카드 결제 대기 상태로 전환한다. 3. (S) Card Payment (Prepay)를 실행한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	1a. (A <sub>U</sub> ) 사용자가 취소를 선택한다. 2a. (S) 메뉴 선택 대기 상태로 전환한다.

<b>Use Case</b>	9. Respond Vm On
<b>Actor</b>	Other DVM
<b>Purpose</b>	자판기 시작 메시지에 응답한다.
<b>Overview</b>	다른 자판기로부터 '자판기 시작 메시지'를 수신하면 해당 자판기를 VMList에 추가하고, 현재 자판기가 '작동중임을 알리는 메시지'를 해당 자판기에 전송한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.1
<b>Pre-Requisites</b>	모든 상태
<b>Typical Courses of Events</b>	(A <sub>U</sub> ): User, (A <sub>M</sub> ): Admin, (A <sub>D</sub> ): Other DVM, (S): System 1. (A <sub>D</sub> ) 다른 자판기가 Broadcast로 '자판기 시작 메시지'를 전송한다. 2. (S) 해당 자판기를 VMList에 추가한다. 3. (S) 자판기가 동작중임을 알리는 메시지 전송을 실행한다.

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

<b>Use Case</b>	10. Request Vm Off (VMList)
<b>Actor</b>	Admin
<b>Purpose</b>	관리자가 자판기를 종료시킨다.
<b>Overview</b>	관리자가 자판기 종료를 선택하면 VMList의 모든 자판기에 종료 메시지를 전송한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.2
<b>Pre-Requisites</b>	자판기가 관리자 모드여야 한다.
<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (<math>S</math>): System</p> <ol style="list-style-type: none"> <li>(<math>A_M</math>) 관리자가 자판기 종료를 선택한다.</li> <li>(<math>S</math>) 현재 자판기가 종료되었다는 메시지를 VMList의 모든 자판기에게 전송한다.</li> <li>(<math>S</math>) 자판기를 종료한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	11. Respond Vm Off
<b>Actor</b>	Other DVM
<b>Purpose</b>	자판기 종료 메시지에 응답한다.
<b>Overview</b>	다른 자판기로부터 '자판기 종료 메시지'를 수신하면 VMList에서 해당 자판기를 삭제한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.2 , R 4.3
<b>Pre-Requisites</b>	모든 상태
<b>Typical Courses of Events</b>	( $A_U$ ): User, ( $A_M$ ): Admin, ( $A_D$ ): Other DVM, ( $S$ ): System

	<ol style="list-style-type: none"> <li>1. (<math>A_D</math>) 다른 자판기가 Request Vm Off를 실행한다.</li> <li>2. (<math>S</math>) 해당 자판기를 VMList에서 제거한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	12. Receive Operating
<b>Actor</b>	Other DVM
<b>Purpose</b>	자판기가 작동 중인 다른 자판기의 메시지를 수신한다.
<b>Overview</b>	현재 자판기가 '작동 중인 다른 자판기'의 ID와 위치를 VMList에 추가한 후 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.4
<b>Pre-Requisites</b>	자판기 시작 메시지를 보낸 후
<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (<math>S</math>): System</p> <ol style="list-style-type: none"> <li>1. (<math>A_D</math>) 다른 자판기가 작동중임을 알리는 메시지를 보낸다.</li> <li>2. (<math>S</math>) 해당 자판기를 VMList에 추가한다.</li> <li>3. (<math>S</math>) Stock Counter가 재고를 측정하여 그 값을 자판기의 재고 수량에 반영한다.</li> <li>4. (<math>S</math>) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	13. Receive Stock
<b>Actor</b>	Other DVM
<b>Purpose</b>	현재 자판기가 다른 자판기로부터 재고 여부 메시지를 받는다.
<b>Overview</b>	현재 자판기가 다른 자판기로부터 재고 여부 메시지를 받아 재고가 있는 자판기들의 리스트를 출력한 후 자판기 목록 선택 대기 상태로 전환한다

<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.5
<b>Pre-Requisites</b>	VMList의 자판기에게 ‘재고 여부 확인 메시지’를 전송한 상태
<b>Typical Courses of Events</b>	<p>(A<sub>U</sub>): User, (A<sub>M</sub>): Admin, (A<sub>D</sub>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>(A<sub>D</sub>) 다른 자판기가 ‘재고 여부 메시지’를 보낸다.</li> <li>(S) VMList의 자판기로부터 메시지를 받는다.</li> <li>(S) 재고가 있다고 응답한 자판기들의 목록에 추가된다.</li> <li>(S) 자판기 목록 선택 대기 상태로 전환한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	14. Respond Refund
<b>Actor</b>	Other DVM
<b>Purpose</b>	자판기가 환불 요청을 받아 환불 기능을 실행한다.
<b>Overview</b>	선결제 과정 중 다른 자판기의 해당 제품 재고가 0이 될 경우 다른 자판기로부터 ‘환불 요청 메시지’를 받는다. 현재 자판기가 ‘환불 요청 메시지’를 받을 경우 환불을 실행하고 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.6
<b>Pre-Requisites</b>	다른 자판기에게 ‘인증코드 생성 요청 메시지’를 보낸 후 메시지 수신 대기 상태
<b>Typical Courses of Events</b>	<p>(A<sub>U</sub>): User, (A<sub>M</sub>): Admin, (A<sub>D</sub>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>(A<sub>D</sub>) 다른 자판기가 ‘환불 요청 메시지’를 전송한다.</li> <li>(S) 환불을 실행한다.</li> <li>(S) 메뉴 선택 대기 상태로 전환한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	15. Receive Verification Code
-----------------	-------------------------------

<b>Actor</b>	Other DVM
<b>Purpose</b>	생성된 <b>Verification Code</b> 를 받아서 화면에 출력한다.
<b>Overview</b>	‘인증코드 생성 요청 메시지’를 전송한 뒤, 다른 자판기에서 생성된 인증코드를 받아 화면에 출력하고 취소 선택하거나 일정 시간이 지난 경우 메뉴 선택 대기 상태로 전환한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.7
<b>Pre-Requisites</b>	다른 자판기에게 ‘인증코드 생성 요청 메시지’를 보낸 후 메시지 수신 대기 상태
<b>Typical Courses of Events</b>	( $A_U$ ): User, ( $A_M$ ): Admin, ( $A_D$ ): Other DVM, ( $S$ ): System 1. ( $A_D$ ) 다른 자판기가 생성된 인증코드를 보낸다. 2. ( $S$ ) 30초간 해당 인증코드를 화면에 출력한다. 3. ( $S$ ) 메뉴 선택 대기 상태로 전환한다.
<b>Alternative Courses of Events</b>	2a. ( $S$ ) 사용자로부터 취소를 입력 받는다. 3a. ( $S$ ) 메뉴 선택 대기 상태로 전환한다.
<b>Exceptional Courses of Events</b>	N/A

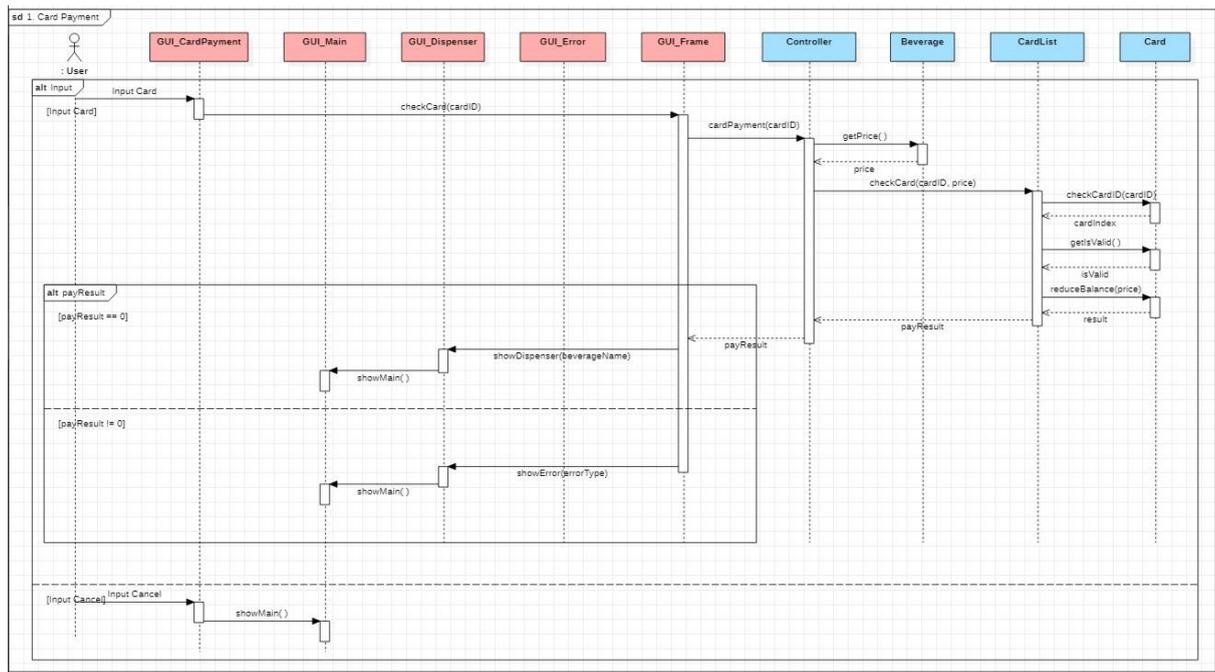
<b>Use Case</b>	16. Respond Verification Code
<b>Actor</b>	Other DVM
<b>Purpose</b>	현재 자판기가 다른 자판기의 인증코드 생성 요청에 응답한다.
<b>Overview</b>	다른 자판기의 ‘인증코드 생성 요청 메시지’를 받을 경우 자신의 재고를 확인해 재고가 있으면 인증코드를 생성하여 메시지로 전송하고 재고가 없으면 ‘환불을 요청 메시지’를 전송한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.8
<b>Pre-Requisites</b>	모든 상태

<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>(<math>A_D</math>) 다른 자판기가 인증코드 생성을 요청한다.</li> <li>(S) 해당하는 음료의 재고를 확인한다.</li> <li>(S) 해당하는 음료의 재고를 1 감소 시킨다.</li> <li>(S) 중복 없는 인증코드를 생성한다.</li> <li>(S) 생성된 인증코드를 인증코드 리스트에 추가한다.</li> <li>(S) 생성된 인증코드를 전송한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	2a. (S) 재고가 없을 경우 ‘환불 요청 메시지’를 전송한다.

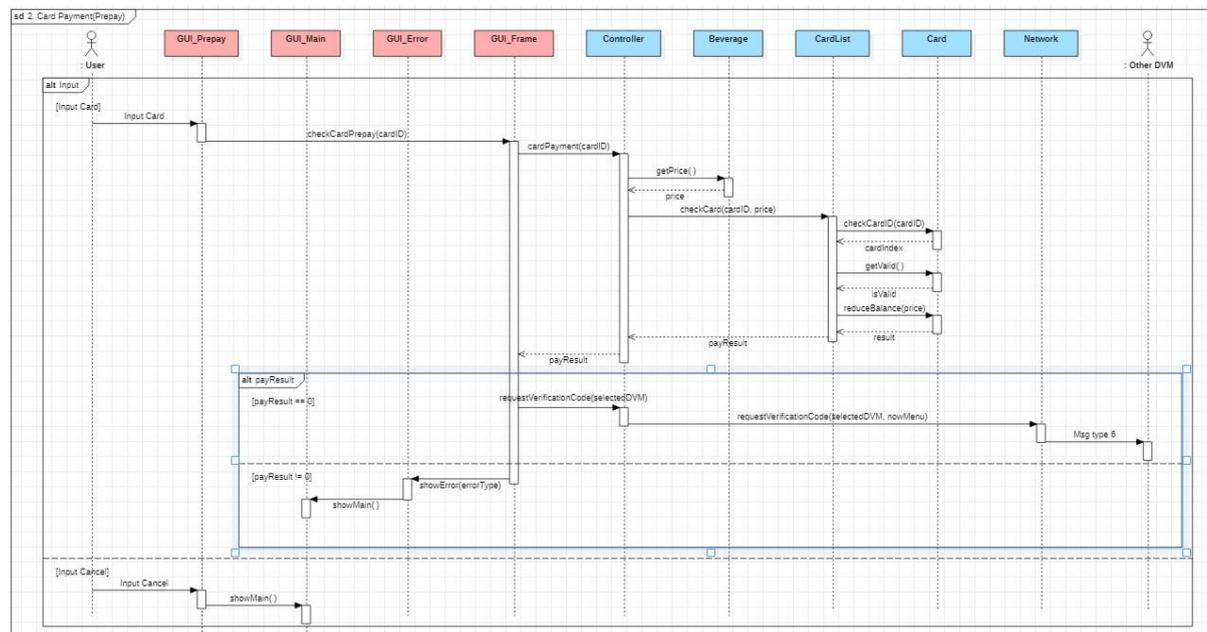
<b>Use Case</b>	17. Respond Stock
<b>Actor</b>	Other DVM
<b>Purpose</b>	재고 있음을 응답한다.
<b>Overview</b>	다른 자판기의 ‘재고 여부 확인 메시지’를 받아 현재 자판기에 해당 음료의 재고를 확인 후 재고 여부(boolean)를 메시지로 응답한다.
<b>Type</b>	Evident
<b>Cross Reference</b>	System Function: R 4.9
<b>Pre-Requisites</b>	모든 상태
<b>Typical Courses of Events</b>	<p>(<math>A_U</math>): User, (<math>A_M</math>): Admin, (<math>A_D</math>): Other DVM, (S): System</p> <ol style="list-style-type: none"> <li>(<math>A_D</math>) 다른 자판기가 ‘재고 여부 확인 메시지’를 보낸다.</li> <li>(S) 자판기가 자신의 재고를 확인한다.</li> <li>(S) 자판기가 재고 여부를 다른 자판기에게 전송한다.</li> </ol>
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

## 2143. Define Interaction Diagrams

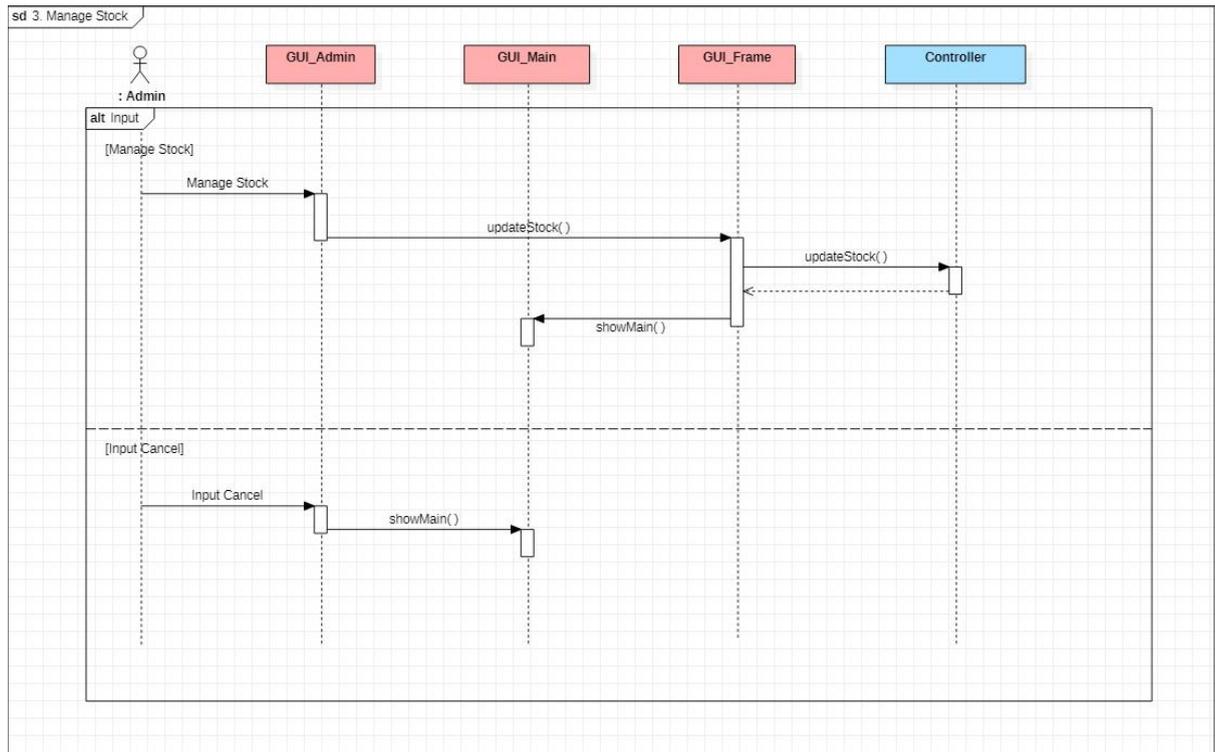
# 1. Use Case 1 - Card Payment



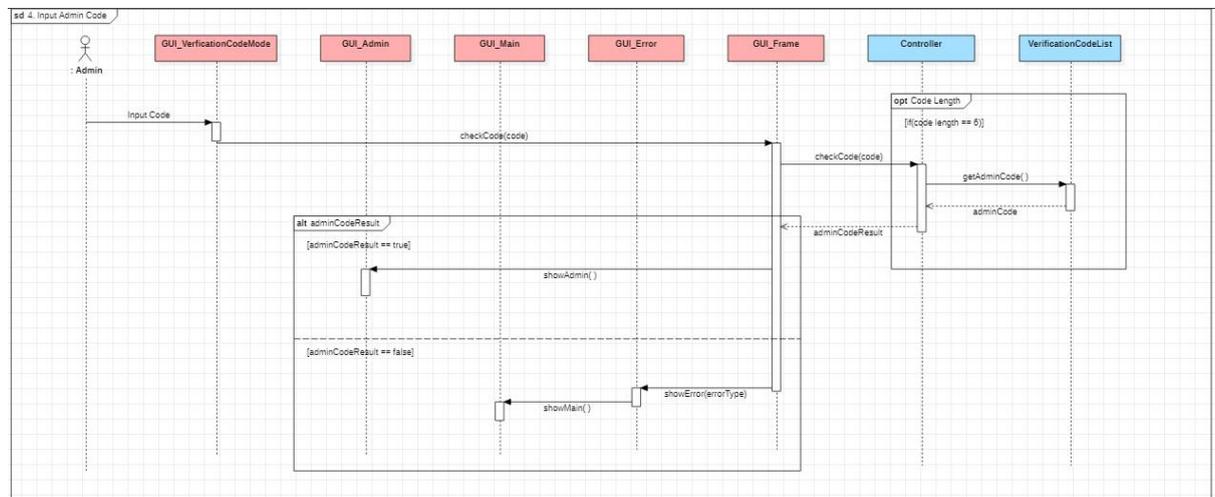
# 2. Use Case 2 - Card Payment(Prepay)



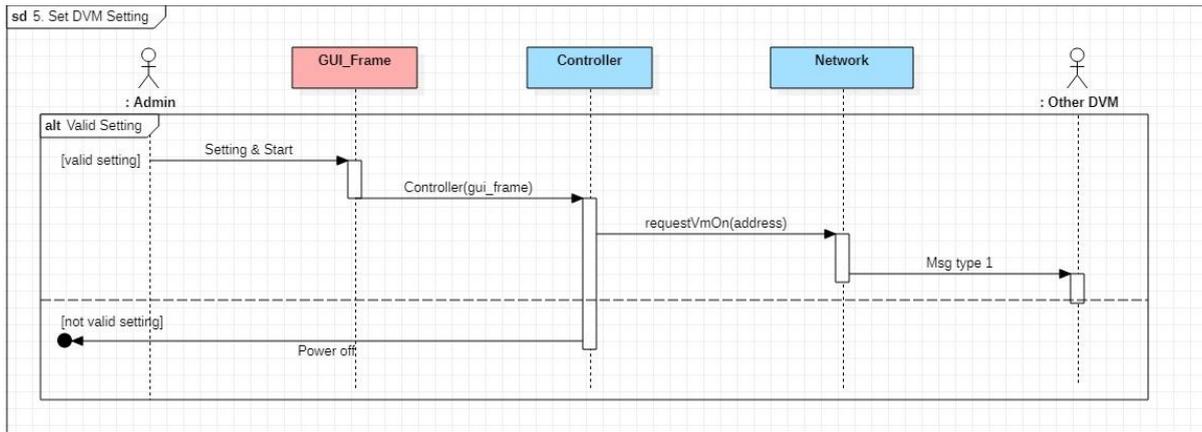
### 3. Use Case 3 - Manage Stock



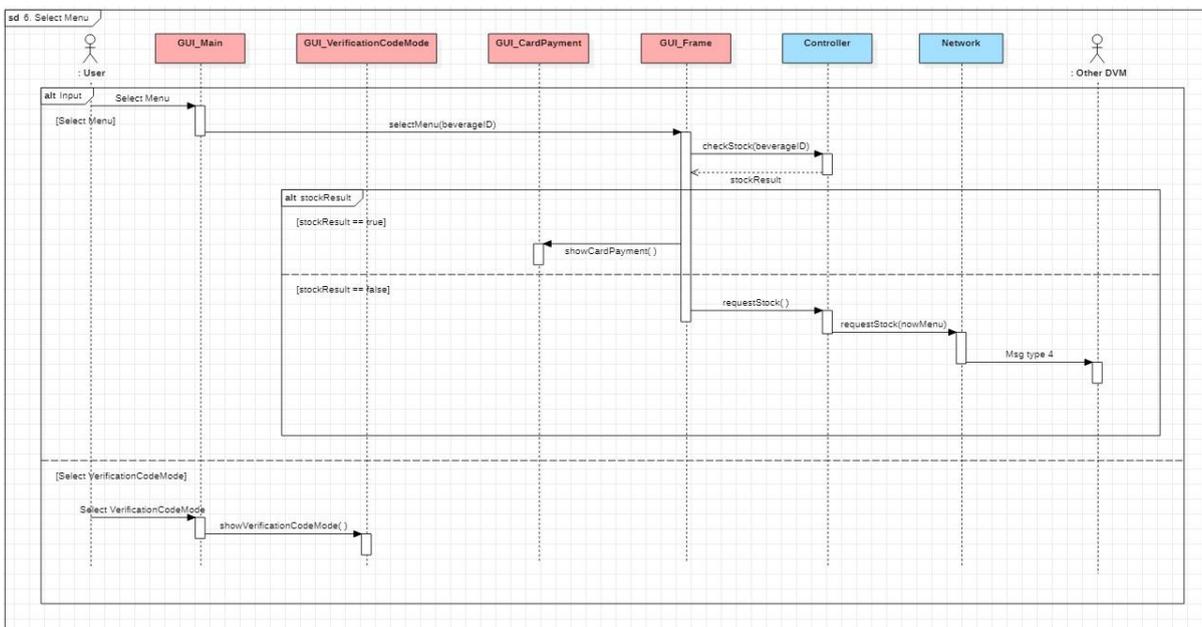
### 4. Use Case 4 - Input Admin Code



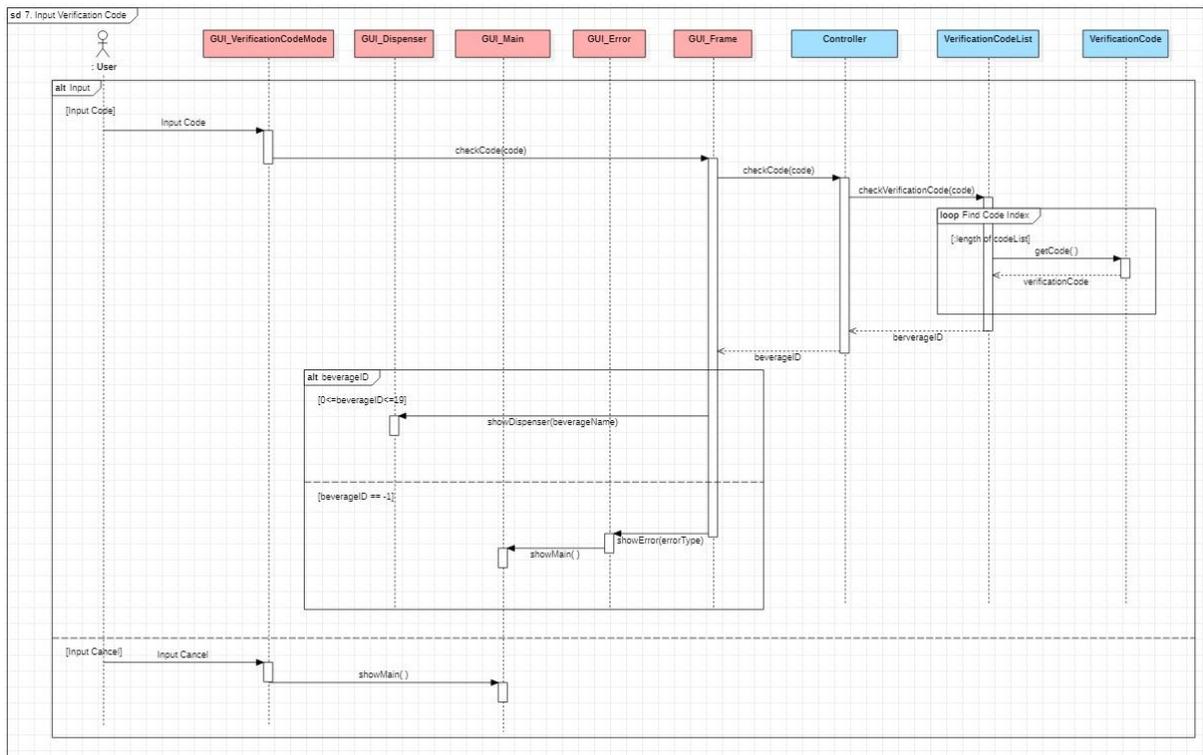
### 5. Use Case 5 - Set DVM Setting



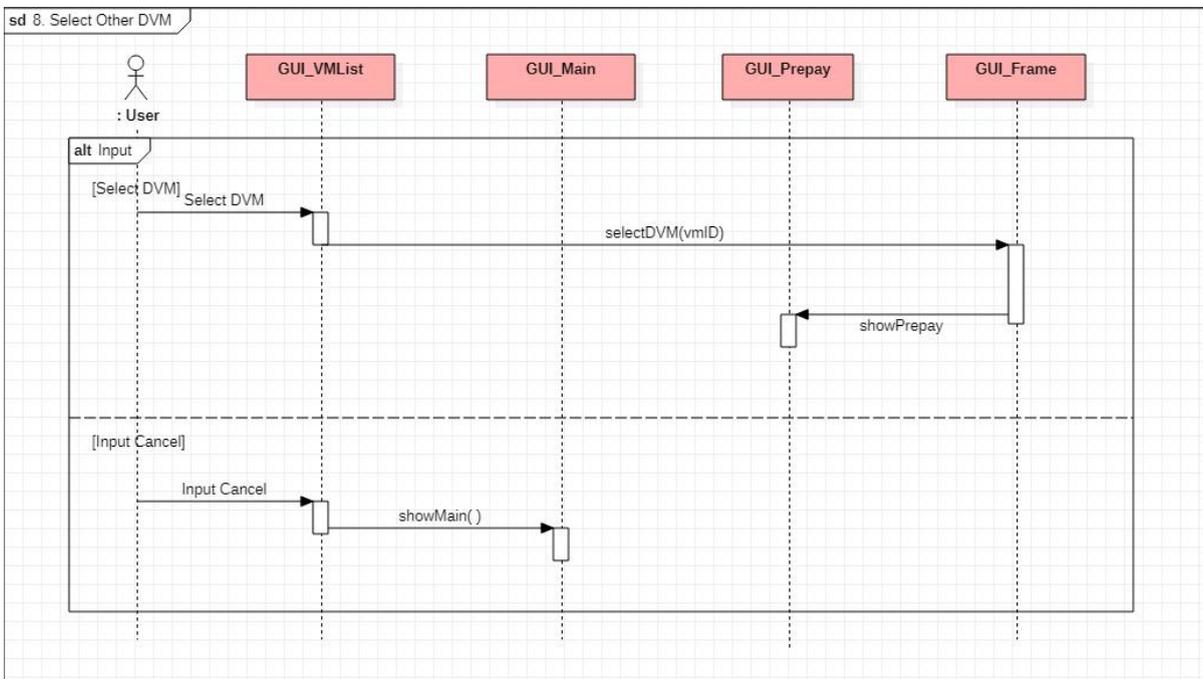
### 6. Use Case 6 - Select Menu



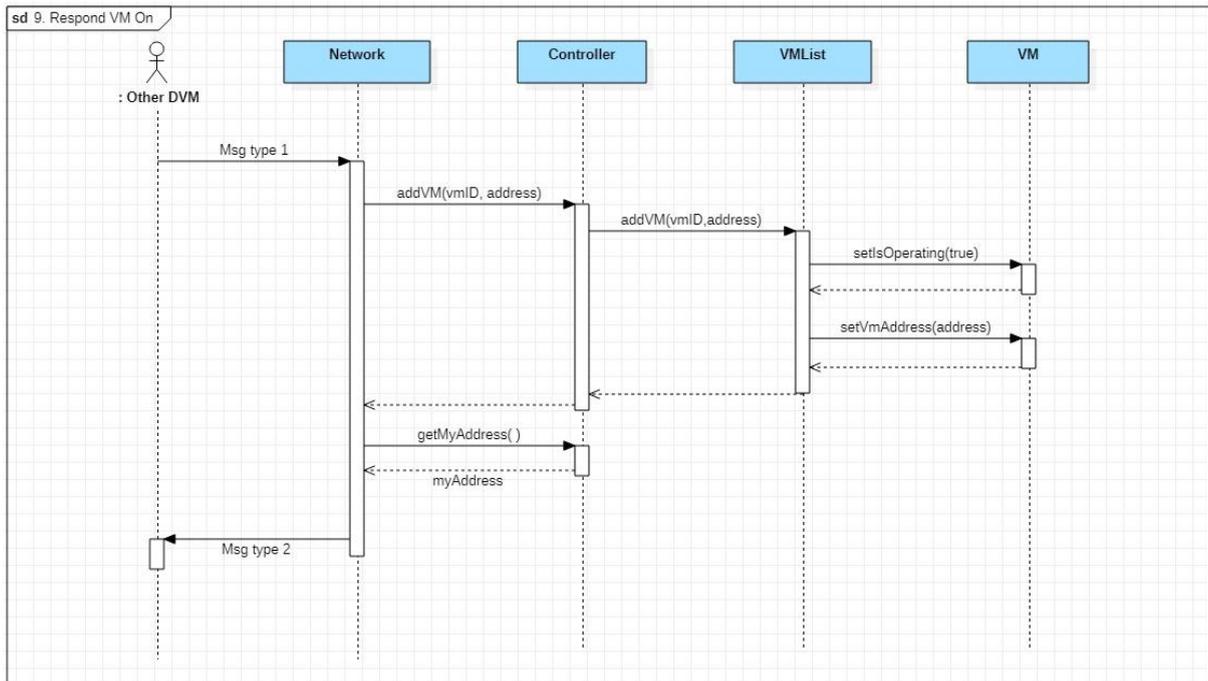
## 7. Use Case 7 - Input Verification Code



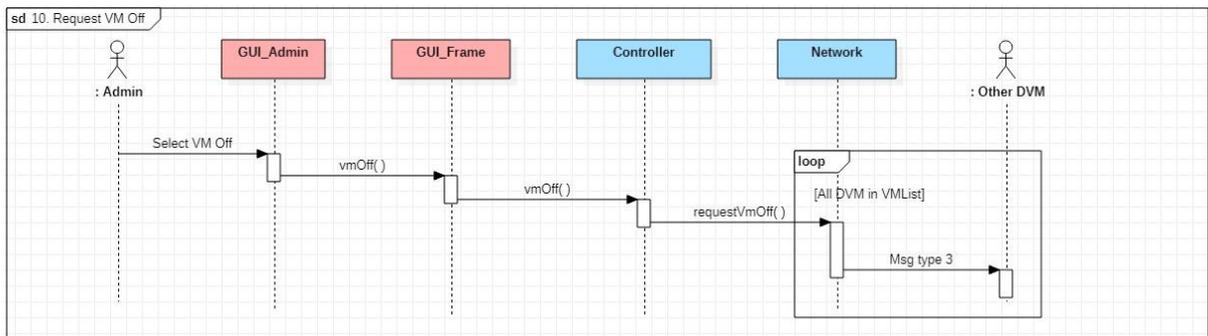
## 8. Use Case 8 - Select Other DVM



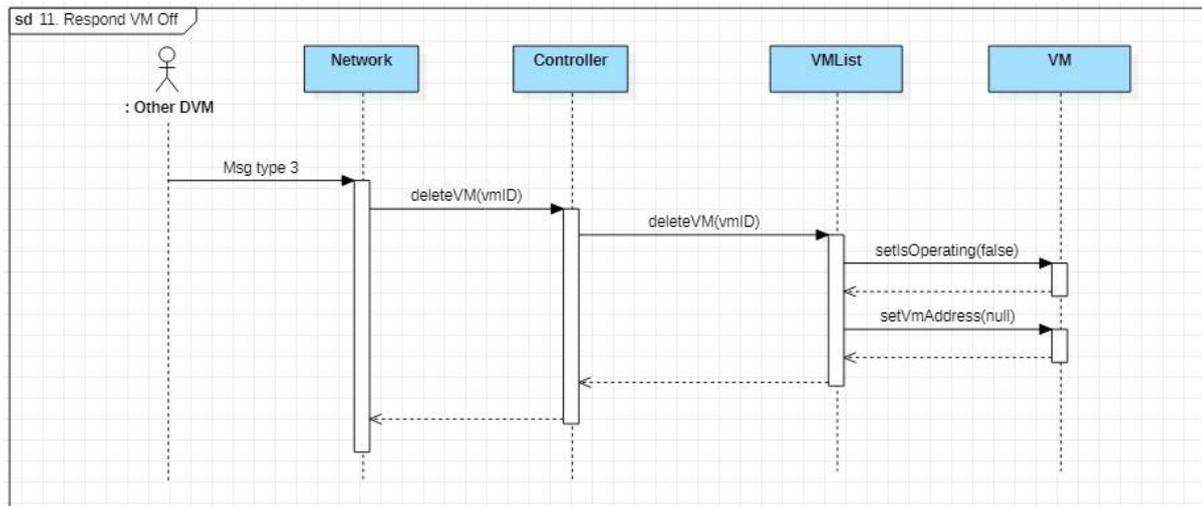
### 9. Use Case 9 - Respond VM On



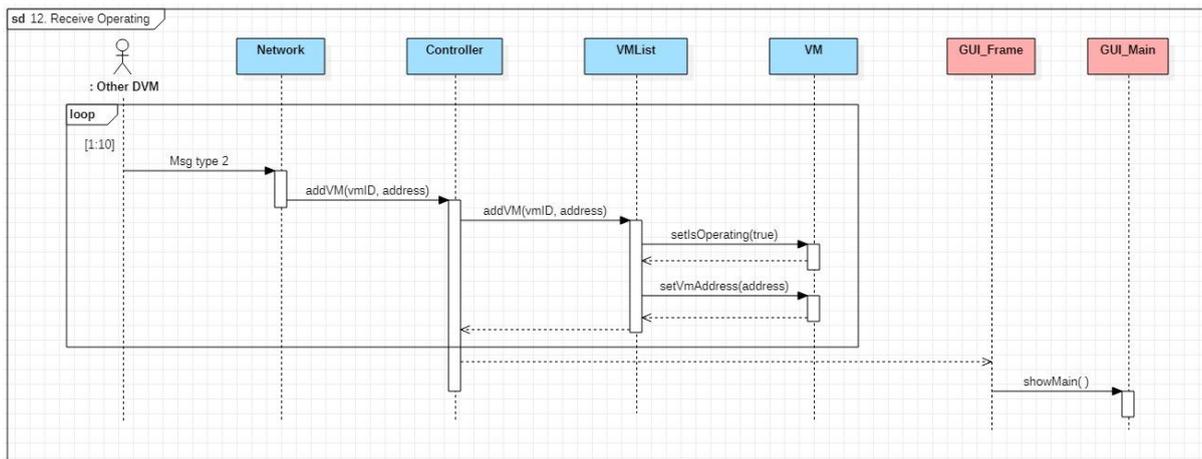
### 10. Use Case 10 - Request VM Off



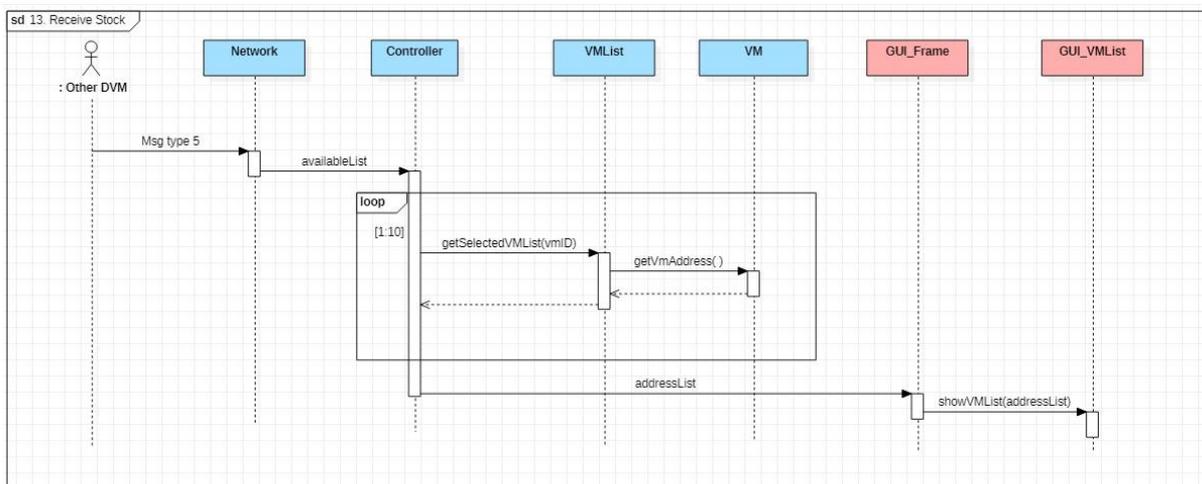
### 11. Use Case 11 - Respond Vm Off



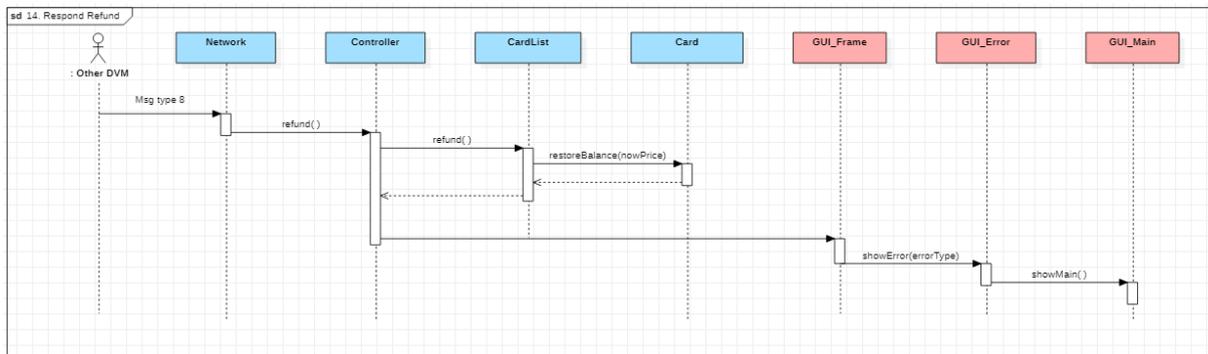
### 12. Use Case 12 - Receive Operating



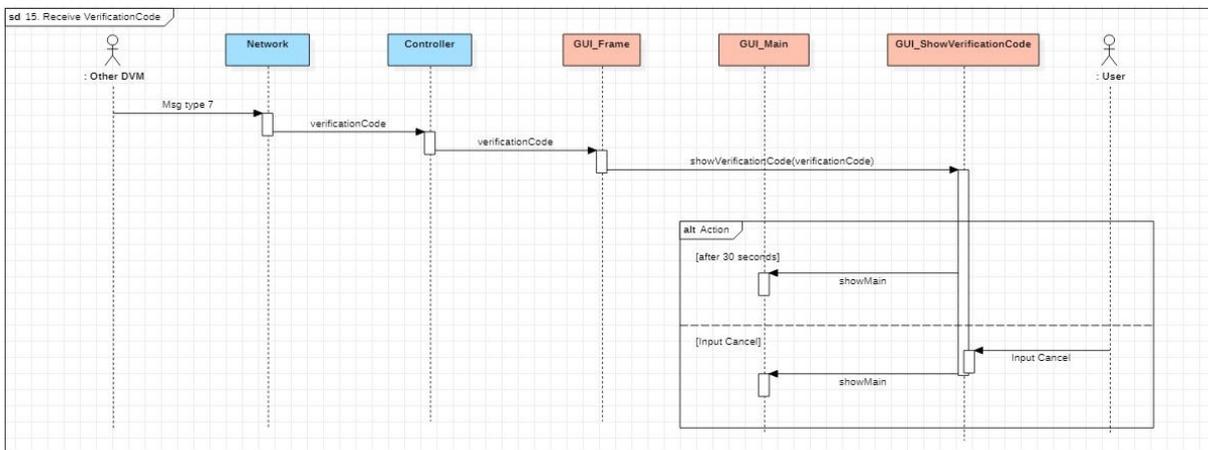
### 13. Use Case 13 - Receive Stock



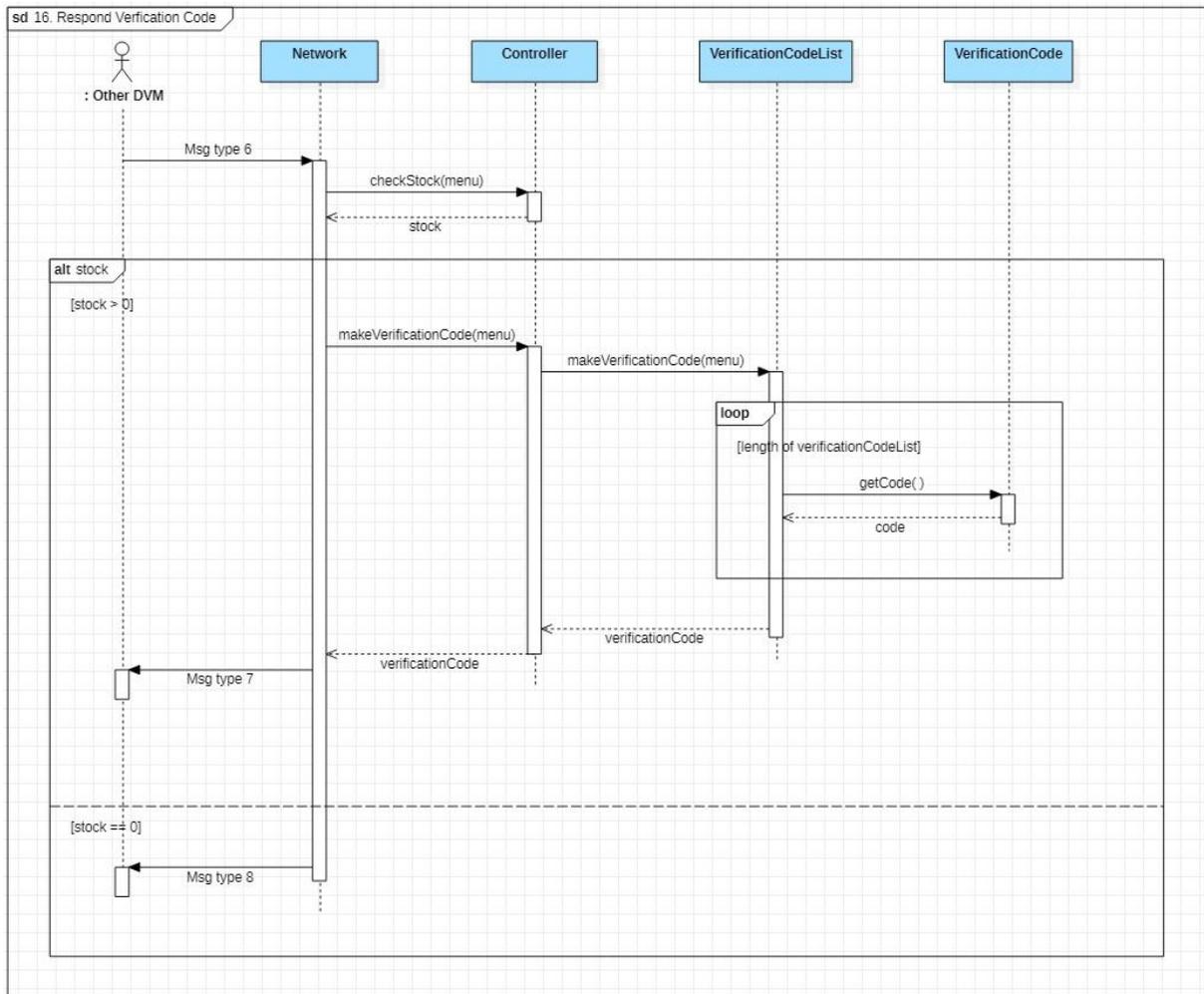
### 14. Use Case 14 - Respond Refund



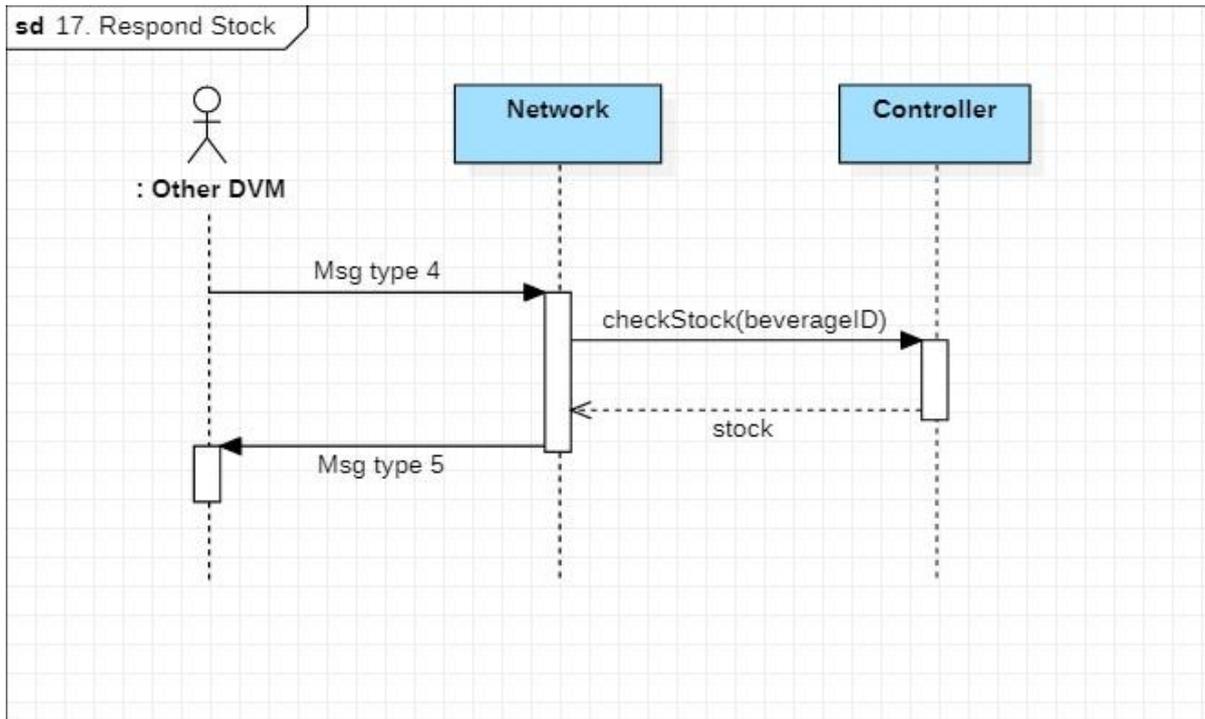
### 15. Use Case 15 - Receive Verification Code



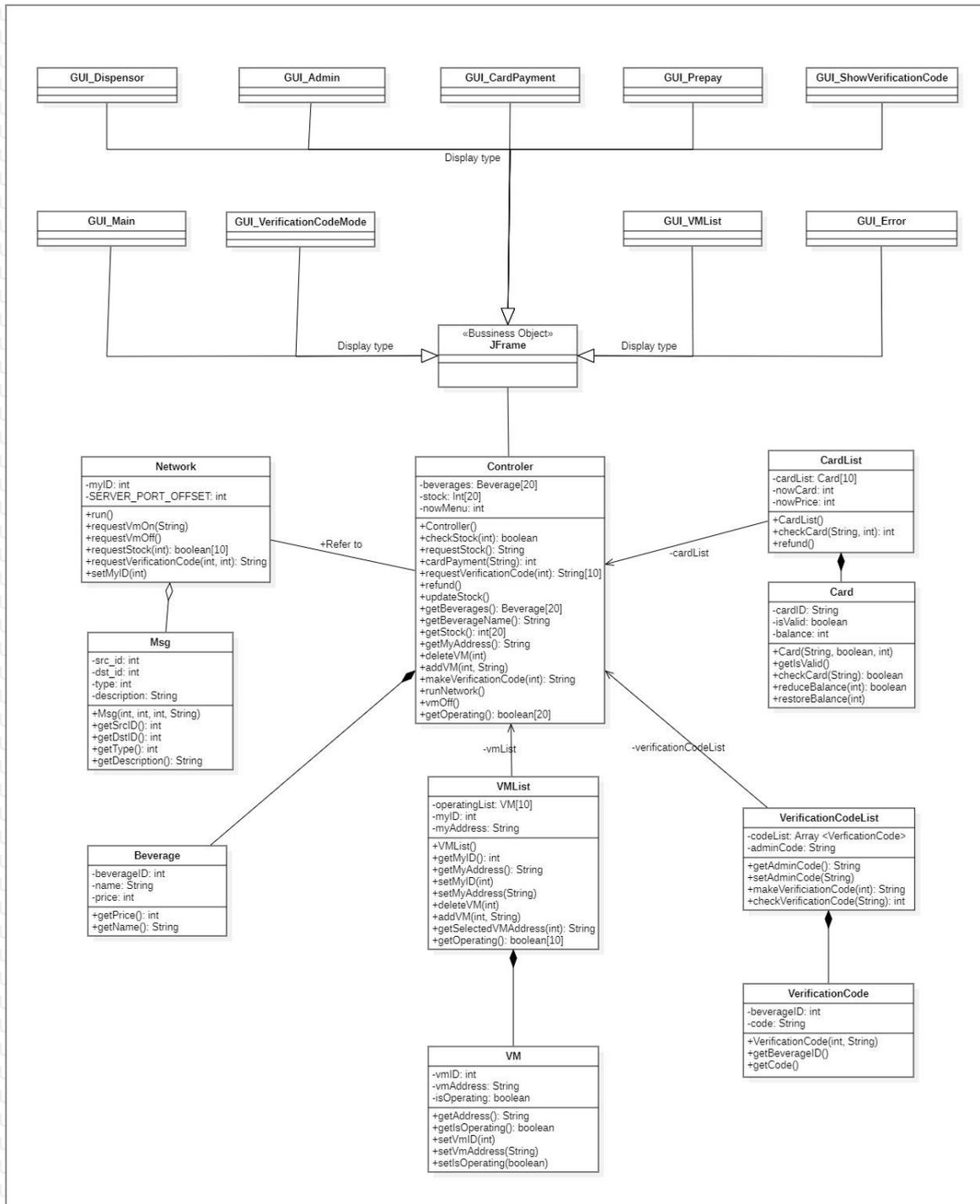
## 16. Use Case 16 - Respond Verification Code



### 17. Use Case 17 - Respond Stock

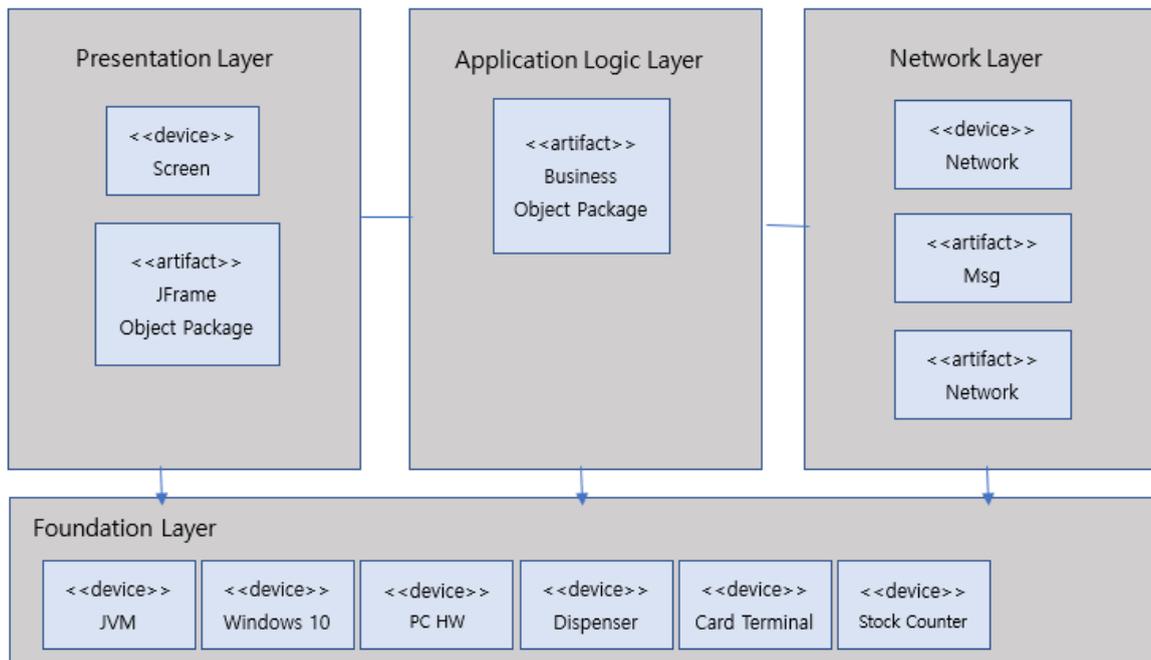


# 2144. Define Design Class Diagrams



# 2145. Refine System Architecture

## 1. Deployment Diagram





## 2147. Perform 2040 Traceability Analysis

Essential Use Case	S-Link
Card Payment	S1,S2,S3
Card Payment (Prepay)	S1,S2,S3
Manage Stock	S2,S3,S4
Input Admin Code	S2,S5,S6
Set DVM Setting	S7,S8
Select Menu	S9,S10,S11
Input Verification Code	S2,S3,S12
Select Other DVM	S2,S3,S13

Respond Vm On	S8,S14
Request Vm Off (VMList)	S15,S16
Respond Vm Off	S16
Receive Operating	S2,S14
Receive Stock	S10,S17
Respond Refund	S2,S18
Receive Verification Code	S2,S3,S19
Respond Verification Code	S18,S20
Respond Stock	S10,S21

SID	Operation in Sequence Diagram	M_Link
S1	inputCard()	M12, M13, M17, M18, M19, M22, M23, M46, M49, M50, M51
S2	showMain()	
S3	selectCancel()	
S4	updateStock()	M20

S5	inputAdminCode()	M53
S6	showAdmin()	
S7	inputVmInfo()	M2,M6,M14,M21,M28,M33,M34,M35,M37,M42,M43,M44,M45,M48, M54
S8	requestVmOn()	M1,M7, M24,M26,M31,M32,M33,M37,M43,M44
S9	selectMenu()	M4, M12, M15, M16,M38,M39,M40,M41
S10	requestStock()	M1,M7,M8,M9,M10,M11,M32
S11	selectVerificationCode	
S12	inputCode()	M56, M58, M59
S13	selectPrePayDVM()	M5, M17
S14	sendOperating()	M8,M9,M10,M11,M26,M30,M32
S15	selectVmOff()	M3,M39,M41
S16	requestVmOff()	M1,M7,M8,M9,M10,M11,M25,M29,M32,M36,M44
S17	showDVMSelect()	
S18	requestRefund()	M7,M8,M9,M10,M11,M32,M47, M52
S19	sendVerificationCode()	M8,M9,M10,M11,M27

S20	requestVerificationCode()	M1,M7,M8,M9,M10,M11,M32, M55, M57
S21	sendStock()	M8,M9,M10,M11

MID	Method	Class
M1	run()	Network
M2	requestVmOn(String)	Network
M3	requestVmOff()	Network
M4	requestStock(int) : boolean[10]	Network
M5	requestVerificationCode(int,int) : String	Network
M6	setMyID(int)	Network
M7	Msg(int,int,int,String)	Msg
M8	getSrcID() : int	Msg
M9	getDstID() : int	Msg
M10	getType() : int	Msg
M11	getDescription() : String	Msg

M12	getPrice() : int	Beverage
M13	getName() : String	Beverage
M14	Controller()	Controler
M15	checkStock(int) : boolean	Controler
M16	requestStock() : String	Controler
M17	cardPayment(String) : int	Controler
M18	requestVerificationCode(int) : String	Controler
M19	refund()	Controler
M20	updateStock()	Controler
M21	getBeverage() : Beverage[20]	Controler
M22	getBeverageName() : String	Controler
M23	getStock() : int[20]	Controler
M24	getMyAddress() : String	Controler
M25	deleteVM(int,String)	Controler
M26	addVM(int,String)	Controler

M27	makeVerificationCode(int) : String	Controler
M28	runNetwork()	Controler
M29	vmOff()	Controler
M30	getOperating() : boolean[20]	Controler
M31	VMList()	VMList
M32	getMyID() : int	VMList
M33	getMyAddress() : String	VMList
M34	setMyID(int)	VMList
M35	setMyAddress(String)	VMList
M36	deleteVM(int)	VMList
M37	addVM(int,String)	VMList
M38	getSelectedVMAddress(int) : String	VMList
M39	getOperating() : boolean[10]	VMList
M40	getAddress() : String	VM
M41	getIsOperating() : boolean	VM

M42	setVmID(int)	VM
M43	setVmAddress(String)	VM
M44	setIsOperating(boolean)	VM
M45	CardList()	CardList
M46	checkCard(String,int) : int	CardList
M47	refund()	CardList
M48	Card(String,boolean,int)	Card
M49	getIsValid()	Card
M50	checkCard(String) : boolean	Card
M51	reduceBalance(int) : boolean	Card
M52	restoreBalance(int)	Card
M53	getAdminCode() : String	VerificationCodeList
M54	setAdminCode(String)	VerificationCodeList
M55	makeVerificationCode(int) : String	VerificationCodeList
M56	checkVerificationCode(String) : int	VerificationCodeList

M57	VerificationCode(int,String)	VerificationCode
M58	getBeverageID()	VerificationCode
M59	getCode()	VerificationCode