

Smart DJ CoffeeMaker

Stage 2140. Design

T1

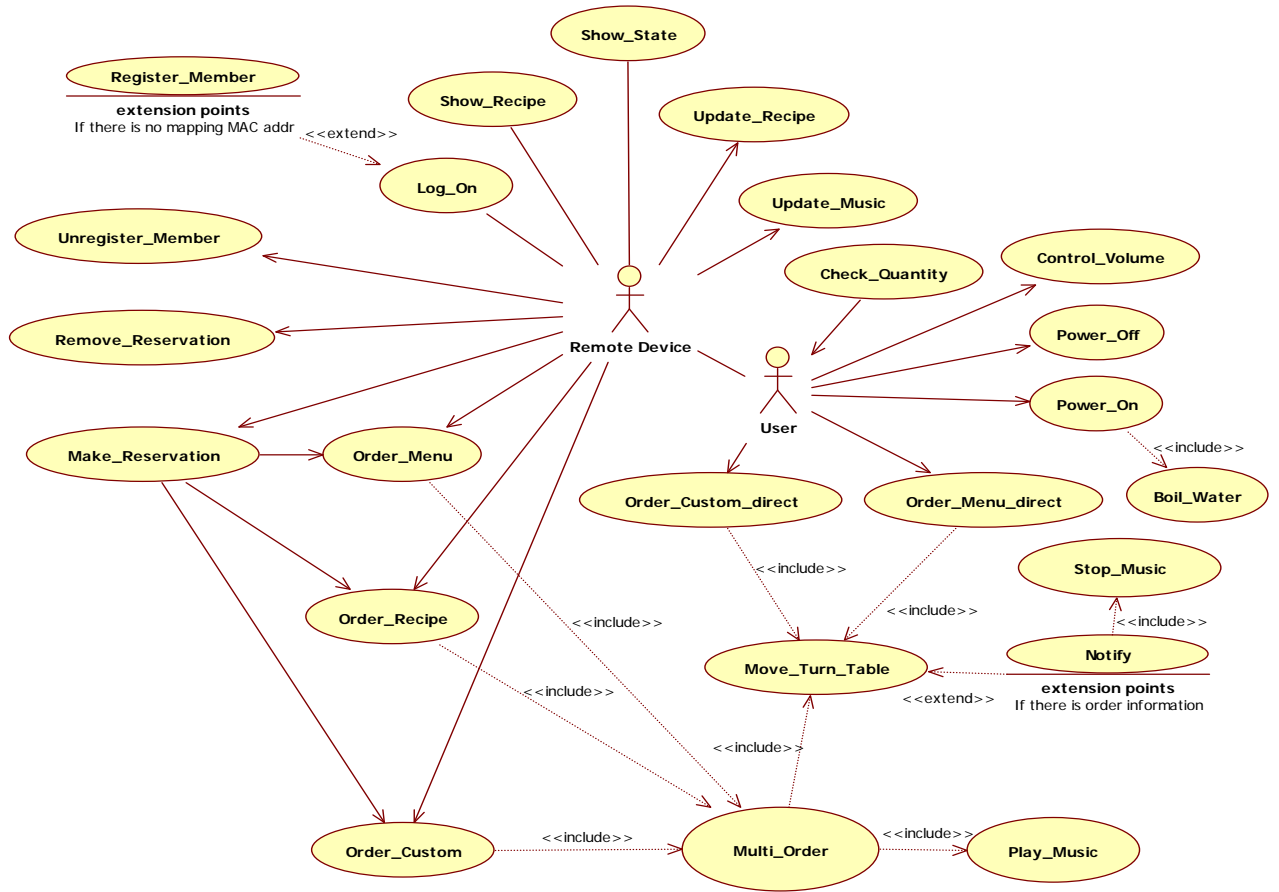
200611499 이낙원 , 200611521 최정명

200611460 김정태 , 200611481 송준현

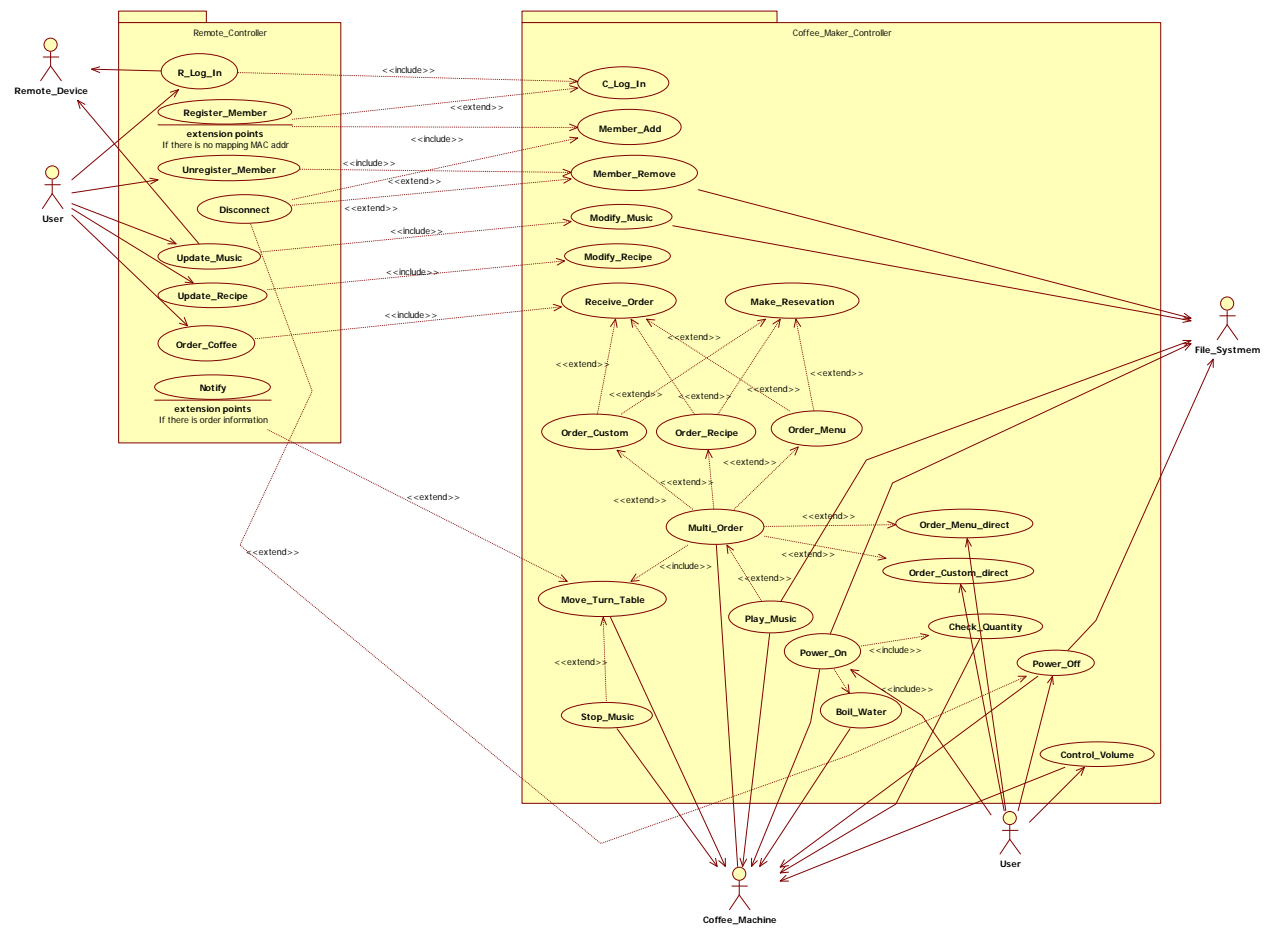
<changed use case>

Reference	Function	Usecase Number and Name
R1.1	power on	1. Power_On
R1.2	power off	2. Power_Off
R2	check quantity of materials	3. Check_Quantity
R3.1.1	order menu	4. Order_Menu
R3.1.2	order menu direct	5. Order_Menu_Direct
R3.2	order recipe	6. Order_Recipe
R3.3.1	order custom	7. Order_Custom
R3.3.2	order custom direct	8. Order_Custom_Direct
R4	multi order	9. Multi_Order
R5.1	order coffee	10. Order_Coffee
R5.2	order coffee	11. Receive_Order
R6	book an order	12. Make_Reservation
R7	boil water	13. Boil_Water
R8.1	log in	14. R_Log_In
R8.2	log in	15. C_Log_In
R9.1	register member	16. Register_Member
R9.2	register member	17. Member_Add
R10.1	unregister member	18. Unregister_Member
R10.2	unregister member	19. Member_Remove
R11.1	update recipe	20. Update_Recipe
R11.2	update recipe	21. Modify_Recipe
R12.1	update music	22. Update_Music
R12.2	update music	23. Modify_Music
R13	control volume	24. Control_Volume
R14	play music	25.. Play_Music
R15	play music	26. Stop_Music
R16	notify	27. Notify
R17	move turn table	28. Move_Turn_Table
R18		29. Disconnect

Previous Use case Diagram



Modified Use case Diagram



Phase 2140. Design

Activity 2141. Define Real Use Cases

Use Case	1. Power_On
Actors	User
Purpose	- Run the <i>Coffee_Machine</i> .
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R1.1 Use-Case :
Pre-Requisites	<i>Coffee_Machine</i> must be off state.
Typical Courses of Events	<ol style="list-style-type: none"> 1 (A) User push the <i>Power</i> Button in front of <i>Coffee_Machine</i>. 2 (S) Invoke "Check_Quantity" use-case 3 (S) Invoke "Boil_Water" use-case 4 (S) Load Member information from <i>File_System</i>.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	2. Power_Off
Actors	User
Purpose	Power off the <i>Coffee_Machine</i> .
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R1.2 Use-Case :
Pre-Requisites	<i>Coffee_Machine</i> must be running state.
Typical Courses of Events	<ol style="list-style-type: none"> 1 (A) User push the <i>Power</i> Button in front of <i>Coffee_Machine</i>. 2 (S) Save Member information to <i>File_System</i>. 3 (S) Invoke "Disconnect" use case. 4 (S) Turn off the <i>Coffee_Maker</i> Machine.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line1: When <i>Coffee_Maker.order_cout</i> is not 0, the machine can't be turned off.

Use Case	3. Check_Quantity
Actors	None
Purpose	- <i>Coffee_Machine</i> check the materials quantity and represent short of materials. - <i>Coffee_Machine</i> check the coffee grounds and represent quantity of that
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R2 Use-Case : "Power_On"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Check <i>Material.Water.quantity</i> and <i>Material.Water.temperature</i>. 2. (S) Check <i>Material.Coffee.quantity</i>. 3. (S) Check <i>Material.Grounds.quantity</i>. 4. (S) Check <i>Material.Syrup.quantity</i>. 5. (S) Check <i>Material.Milk.quantity</i>.
Alternative Courses of Events	If Material's <i>if_FLAGS</i> meets specified limit, set <i>if_FLAGS</i> to true, and then <i>turn_off_MATERIAL's_LED()</i> .
Exceptional Courses of Events	<p>Line 1. if <i>Material.Water.quantity</i> and <i>temperature</i> are lower than specified limit, set <i>Material.if_Water</i> to false and then call <i>turn_on_water_LED()</i>.</p> <p>Line 2. if <i>Material.Coffee.quantity</i> is lower than specified limit, set <i>Material.if_Coffee</i> to false and then call <i>turn_on_Coffee_LED()</i>.</p> <p>Line 3. if <i>Material.Grounds.quantity</i> is lower than specified limit, set <i>Material.if_Grounds</i> to false and then call <i>turn_on_Grounds_LED()</i>.</p> <p>Line 4. if <i>Material.Syrup.quantity</i> is lower than specified limit, set <i>Material.if_Syrup</i> to false, and then call <i>turn_on_Syrup_LED()</i>.</p> <p>Line 5. if <i>Material.Milk.quantity</i> is lower than specified limit, set <i>Material.if_Milk</i> to false and then call <i>turn_on_Milk_LED()</i>.</p>

Use Case	4. Order_Menu
Actors	None
Purpose	Order Coffee Menu.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R3.1.1 R4 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Compare <i>Order.menu_type</i> with <i>Coffee_Maker.Default_Recipe.menu_type</i>. 2. (S) If two variables are same, set <i>Order.recipe</i> to <i>Coffee_Maker.Default_Recipe</i>. 3. (S) Input <i>Order</i> to <i>Coffee_Maker.order_queue</i>. 4. (S) Increase <i>Coffee_Maker.order_count</i> by 1. 5. (S) Invoke "Multi_Order" use-case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4. If <i>Coffee_Maker.is_Working</i> true, do not invoke "Multi_Order" use-case.

Use Case	5. Order_Menu_Direct
Actors	User
Purpose	User orders coffee menu directly.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R3.1.2 R15 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User select Coffee menu at <i>Menu Panel</i>. 2. (S) Create <i>Order</i> object 3. (S) Set <i>Order.time</i>. 4. (S) Set <i>Order.id</i>. 5. (S) Set <i>Order.menu_type</i> and <i>Order.order_type</i>. 6. (S) Compare <i>Order.menu_type</i> with <i>Coffee_Maker.Default_Recipe.menu_type</i>. 7. (S) If two variables are same, set <i>Order.recipe</i> to <i>Coffee_Maker.Default_Recipe</i> 8. (S) Input <i>Order</i> to <i>Coffee_Maker.order_queue</i>. 9. (S) Increase <i>Coffee_Maker.order_count</i> by 1. 10. (S) Invoke "Multi_Order" use-case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 10. If <i>Coffee_Maker.is_Working</i> true, do not invoke "Multi_Order" use-case.

Use Case	6. Order_Recipe
Actors	None
Purpose	User order coffee by his own stored recipe.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R3.2 R4 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Search corresponding <i>Order.id</i> in <i>member</i>. 2. (S) Set <i>Order.recipe</i> to <i>Member.own_recipe</i>. 3. (S) Input <i>Order</i> to <i>Coffee_Maker.order_queue</i>. 4. (S) Increase <i>Coffee_Maker.order_count</i> by 1. 5. (S) Invoke "Multi_Order" use-case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 5. If <i>Coffee_Maker.is_Working</i> true, do not invoke "Multi_Order" use-case.

Use Case	7. Order_Custom
Actors	None
Purpose	User order coffee by customized recipe.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R3.3.1 R4 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Input <i>Order</i> to <i>Coffee_Maker.order_queue</i>. 2. (S) Increase <i>Coffee_Maker.order_count</i> by 1. 3. (S) Invoke "Multi_Order" use-case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 3. If <i>Coffee_Maker.is_Working</i> true, do not invoke "Multi_Order" use-case.

Use Case	8. Order_Custom_Direct
Actors	User
Purpose	User order coffee directly at the <i>Coffee_Machine</i> by customized recipe.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R3.3.2 R15 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User sets value of <i>Coffee, Water, Milk, Syrup</i> of <i>Custom Panel</i>. 2. (A) then, user push <i>Order</i> button. 3. (S) Create <i>Order</i> object. 4. (S) Set <i>Order.time</i>. 5. (S) Set <i>Order.id</i>. 6. (S) Set <i>Order.menu_type</i> and <i>Order.order_type</i>. 7. (S) Create <i>Recipe</i> object. 8. (S) Set <i>Recipe.coffee, Recipe.water, Recipe.milk, Recipe.syrup</i>. 9. (S) Input <i>Order</i> to <i>Coffee_Maker.order_queue</i>. 10. (S) Increase <i>Coffee_Maker.order_count</i> by 1. 11. (S) Invoke "Multi_Order" use-case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 11. If <i>Coffee_Maker.is_Working</i> true, do not invoke "Multi_Order" use-case.

Use Case	9. Multi_Order
Actors	None
Purpose	Make coffee according to order information.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R4 R12 R15 Use-Case : "Play_Music", "Move_Turn_Table"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Check <i>Coffee_Maker.order_count</i> 2. (S) Set <i>Coffee_Maker.is_working</i> to true. 3. (S) Dequeue <i>Order</i> object in the <i>Coffee_Maker.order_queue</i>. 4. (S) Decrease <i>Coffee_Maker.order_count</i> by 1. 5. (S) Check <i>Material.is_Water</i>, <i>Material.if_Coffe</i>, <i>Material.if_Grounds</i>, <i>Material.if_Milk</i>, <i>Material.if_Syrup</i> with <i>Order.recipe</i>. 6. (S) Find <i>Coffee_Maker.member[i]</i> corresponding <i>Order.id</i>. 7. (S) Invoke "Play_Music" with <i>Coffee_Maker.member[i].music_path</i>. 8. (S) Request for making coffee to <i>Coffee_Machine</i>. 9. (S) Wait for coffee making. 10. (S) Invoke "Move_Turn_Table" use-case. 11. (S) Go to Line 1.
Alternative Courses of Events	Line 6. If there is no corresponding <i>Order.id</i> , go to Line 8.
Exceptional Courses of Events	Line 1. If <i>Coffee_Maker.order_count</i> is 0, terminate this use case. Line 5. If There is short of Materials, go to Line 1.

Use Case	10. Order_Coffee
Actors	User
Purpose	Order coffee.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R12 R4 Use-Case : "Receive_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User check the order type in <i>Oder Type</i>. 2. (A) User push <i>Order</i> button or <i>Reservation</i> button. 3. (S) Check if <i>Remote_Control.is_connect</i> is true. 4. (S) Invoke "Receive_Order" with order information.
Alternative Courses of Events	Line 1. If <i>order type</i> is <i>Menu</i> , user should select coffee menu in <i>Menu Panel</i> . Line 1. If <i>order type</i> is <i>Custom</i> , user set quantity of coffee in <i>Quantity Panel</i> .
Exceptional Courses of Events	Line 3. If <i>the value</i> is <i>false</i> , terminate this use case.

Use Case	11. Receive_Order
Actors	None
Purpose	Classify order's kind.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R12 R14 Use-Case : "Make_Reservation", "Order_Custom", "Order_Recipe", "Order_Menu"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Find <i>Coffee_Maker.Member[i]</i> corresponding <i>Remote_Control.physical_addr</i>. 2. (S) Create "Order", "Recipe" object. 3. (S) Set <i>Order.id</i> to <i>Coffee_Maker.Member[i].id</i>. 4. (S) Set <i>Order.time</i>, <i>Order.order_type</i>, <i>Order.menu_type</i>, <i>recipe</i> with order information. 5. (S) Check <i>Order.time</i>. 6. (S) Check <i>Order.order_type</i>.
Alternative Courses of Events	<p>Line 5. If <i>Order.time</i> is Reservation, put <i>Order</i> in <i>Coffee_Maker.reservation_array</i> and then invoke "Make_Reservation" use case with <i>Order</i>.</p> <p>Line 6. If <i>Order.order_type</i> is Custom, invoke "Order_Custom" use case with <i>Order</i>. If <i>Order.order_type</i> is Menu, invoke "Order_Menu" use case with <i>Order</i>. If <i>Order.order_type</i> is Recipe, invoke "Order.Menu" use case with <i>Order</i>.</p>
Exceptional Courses of Events	N/A

Use Case	12. Make_Reservation
Actors	None
Purpose	Reserve order.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R13.1 R3.11 R3.2 R3.3.1 Use-Case : "Order_Menu", "Order_Recipe", "Order_Custom", "Receive_Order"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Check <i>Coffee_Maker.reservation_array[i].Order.time</i> with current time. 2. (S) If the <i>time</i> corresponds with current time, check <i>Coffee_Maker.reservation_array[i].Order.menu_type</i>. 3. (S) Go to Line 1.
Alternative Courses of Events	<p>Line 2. If <i>Order.order_type</i> is Custom, invoke "Order_Custom" use case with <i>Order</i>. If <i>Order.order_type</i> is Menu, invoke "Order_Menu" use case with <i>Order</i>. If <i>Order.order_type</i> is Recipe, invoke "Order.Menu" use case with <i>Order</i>.</p>
Exceptional Courses of Events	N/A

Use Case	13. Boil_Water
Actors	None
Purpose	Keep temperature and quantity of water.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R5 Use-Case :
Pre-Requisites	<i>Coffee_Machine</i> must be working.
Typical Courses of Events	1. (S) Check <i>Coffee_Maker.Material.Water.quantity</i> . 2. (S) Check <i>Coffee_Maker.Material.Water.temperature</i> .
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 1. If <i>Coffee_Maker.Material.Water.quantity</i> is lower than the limit, call <i>supply_Water()</i> to <i>Coffee_Machine</i> . Line 2. If <i>Coffee_Maker.Material.Water.temperature</i> is lower than the limit, call <i>boil_Water()</i> to <i>Coffee_Machine</i> .

Use Case	14. R_Log_In
Actors	User
Purpose	Connect device to Coffee_Maker.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R6 R7.1 Use-Case : "C_Log_In"
Pre-Requisites	N/A
Typical Courses of Events	1. (A) User executes remote control program. 2. (S) Set <i>Remote_Control.is_connect</i> to true. 3. (S) Invoke "C_Log_In" with <i>Remote_Control.physical_addr</i> .
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	15. C_Log_In
Actors	None
Purpose	Connect device to Coffee_Maker.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R6 R7.1 Use-Case : "Register_Member", "R_Log_In"
Pre-Requisites	N/A
Typical Courses of Events	1. (S) Compare <i>Coffee_Maker.Member[i].address</i> with <i>Remote_Control.physical_addr</i> . 2. (S) connect device and <i>Coffee_Machine</i> .
Alternative Courses of Events	Line 1. If there is no mapping <i>Coffee_Maker.Member[i].address</i> with <i>Remote_Control.physical_addr</i> ; invoke "Register_Member" use case.
Exceptional Courses of Events	N/A

Use Case	16. Register_Member
Actors	None
Purpose	Register user device on to Coffee_Maker.
Overview	- (As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R7.1 R6 Use-Case : "C_Log_In", "Member_Add"
Pre-Requisites	There no mapping <i>Remote_Control.physical_addr</i> .
Typical Courses of Events	<ol style="list-style-type: none"> (A) User enter the serial number in <i>Insert serial number</i> Textbox. (A) User push <i>Send</i> button. (S) Invoke "Member_Add" use case with serial number.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	17. Member_Add
Actors	None
Purpose	Register user device on to Coffee_Maker.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R7.1 R6 Use-Case : "Register_Member"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> (S) Check if serial number is correct. (S) Find <i>Coffee_Maker.Member[i]</i> that do not have <i>Coffee_Maker.Member[i].address</i>. (S) Set <i>Coffee_Maker.Member[i].address</i> to <i>Remote_Control.physical_addr</i>.
Alternative Courses of Events	N/A
Exceptional Courses of Events	<p>Line 1. If serial number is incorrect, invoke "Disconnect" use case.</p> <p>Line 2. If there is no <i>Coffee_Maker.Member[i]</i> that do not have <i>Coffee_Maker.Member[i].address</i>, invoke "Disconnect" use case.</p>

Use Case	18. Unregister_Member
Actors	User
Purpose	Remove member information.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R7.2 R7.1 Use-Case : "Member_Remove"
Pre-Requisites	User device's <i>Remote_Control.physical_addr</i> is mapped on <i>Coffee_Maker.member</i>
Typical Courses of Events	<ol style="list-style-type: none"> (A) User push <i>Unregister</i> button. (S) Check if <i>Remote_Control.is_connect</i> is <i>true</i>. (S) Invoke "Member_Remove" use case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2. If <i>the value</i> is <i>false</i> , terminate this use case.

Use Case	19. Member_Remove
Actors	None
Purpose	Remove member information.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R7.2 R7.1 Use-Case : "Unregister_Member"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Find <i>Coffee_Maker.Member[i]</i> corresponding <i>Remote_Control.physical_addr</i>. 2. (S) Remove <i>Coffee_Maker.Member[i].address</i>. 3. (S) Initialize <i>Coffee_Maker.Member[i].own_recipe</i>. 4. (S) Remove Music File on <i>Coffee_Maker.Member[i].music_path</i> from <i>File_System</i>. 5. (S) Initialize <i>Coffee_Maker.Member[i].music_path</i>. 6. (S) Invoke "Disconnect" use case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	20. Update_Recipe
Actors	User
Purpose	Update user recipe.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R8.1 Use-Case : "Modify_Recipe"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User sets quantity of materials on <i>Quantity</i> label. 2. (A) User push <i>Update Recipe</i> button. 3. (S) Check if <i>Remote_Control.is_connect</i> is <i>true</i>. 4. (S) Invoke "Modify_Recipe" use case.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 3. if <i>the value</i> is <i>false</i> , terminate this use case.

Use Case	21. Modify_Recipe
Actors	None
Purpose	Update user recipe.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R8.1 Use-Case : "Update_Recipe"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Find <i>Coffee_Maker.Member[i]</i> corresponding <i>Remote_Control.physical_addr</i>. 2. (S) Create <i>Recipe</i> object according to received information. 3. (S) Set <i>Coffee_Maker.Member[i].Recipe</i> to the <i>Recipe</i>.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	22. Update_Music
Actors	User
Purpose	Update user music.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R8.2 Use-Case : "Modify_Music"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User pushes <i>Update Music</i> button. 2. (A) User selects music file to upload. 3. (A) User pushes <i>Update</i> button. 4. (S) Check if <i>Remote_Control.is_connect</i> is <i>true</i>. 5. (S) Check if the music is valid. 6. (S) Invoke "Modify_Music" use case with the music file.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 4. If the <i>value</i> is <i>false</i> , terminate this use case. Line 5. If the music file is not valid, display "error message".

Use Case	23. Modify_Music
Actors	None
Purpose	Update user music.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R8.2 Use-Case : "Update_Music"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Find <i>Coffee_Maker.Member[i]</i> corresponding <i>Remote_Control.physical_addr</i>. 2. (S) Set <i>Coffee_Maker.Member[i].music_path</i> according to <i>File_System</i>.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	24. Control_Volume
Actors	User
Purpose	Control <i>Coffee_Maker</i> music volume.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R11 Use-Case :
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (A) User push volume button on <i>Volume Panel</i>. 2. (S) Set <i>Coffee_Maker.volume</i> according to user input 3. (S) Call <i>Coffee_Machine.control_volume()</i>.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	25. Play_Music
Actors	None
Purpose	Play music while Coffee_Maker is working.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R12 R4 Use-Case : "Multi_Order"
Pre-Requisites	N/A
Typical Courses of Events	1. (S) Call <i>Coffee_Machine.Play_Music()</i> with <i>Coffee_Maker.Member[i].music_path</i> .
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	26. Stop_Music
Actors	None
Purpose	Stop music
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R12 R14 Use-Case : "Move_Turn_Table"
Pre-Requisites	N/A
Typical Courses of Events	1. (S) Call <i>Coffee_Machine.Stop_Music()</i> .
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	27. Notify
Actors	None
Purpose	Notify of completion of coffee and turn table number.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R14 R15 Use-Case : "Move_Turn_Table"
Pre-Requisites	N/A
Typical Courses of Events	1. (S) Show <i>Notify Window</i> with turn table number to user.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	28. Move_Turn_Table
Actors	None
Purpose	Turn 1 cycle the turn table and stop playing music.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R15 R14 Use-Case : "Notify", "Multi_Order", "Stop_Music"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Check <i>Order.id</i> if Coffee order is direct or not. 2. (S) Invoke "Stop_Music" use case. 3. (S) Invoke "Notify" use case with <i>Coffe_Maker.table_num</i> and <i>Coffee_Maker.Member[{}].address..</i> 4. (S) Call <i>Coffee_Machine.move_turn_table()</i>. 5. (S) Increase <i>Coffee_Maker.table_num</i> by 1.
Alternative Courses of Events	Line 1. If <i>Order.id</i> is not exist, go to Line 4. Line 4. If <i>Coffee_Maker.table_num</i> bigger than 6, turn <i>the value</i> to 1.
Exceptional Courses of Events	N/A

Use Case	29. Disconnect
Actors	None
Purpose	Control Coffee_Maker music volume.
Overview	(As in the Business use case)
Type	Primary & Essential
Cross Reference	Functional Requirement : R11 Use-Case : "Member_Add", "Member_Remove", "Power_off"
Pre-Requisites	N/A
Typical Courses of Events	<ol style="list-style-type: none"> 1. (S) Set <i>Remote_Control.is_connect</i> to false.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Activity 2142. Define Reports, UI, and Storyboards

- **Basic UI**

-

- **Update Music**

-

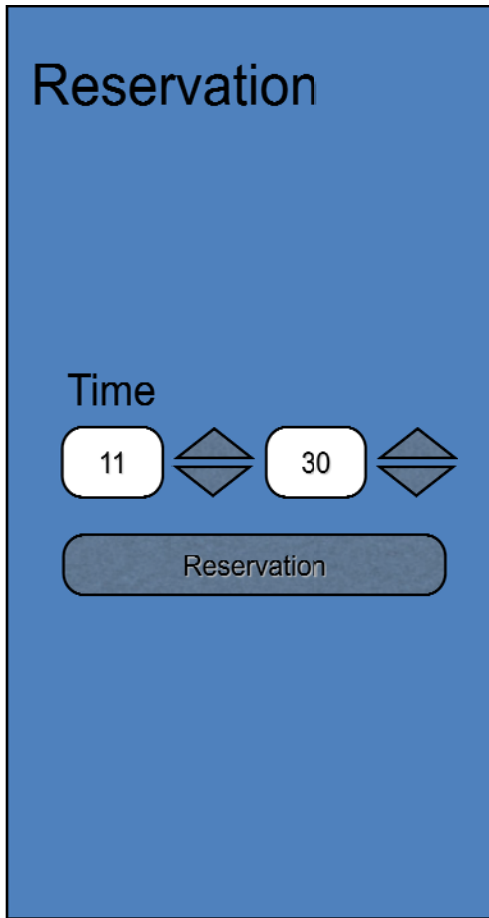
- Reservation

Reservation

Time

11 30

Reservation

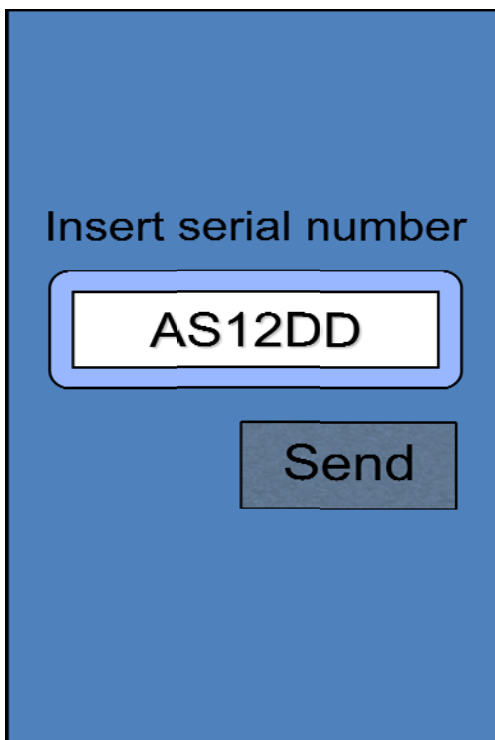
A blue rectangular screen with the title "Reservation" at the top. Below the title, the word "Time" is displayed. Underneath "Time", there are two white rounded rectangular input fields. The first field contains the number "11" and the second field contains the number "30". Between the two input fields and to the right of the second field are diamond-shaped navigation arrows. Below the input fields is a grey rounded rectangular button with the text "Reservation".

- Authorization

Insert serial number

AS12DD

Send

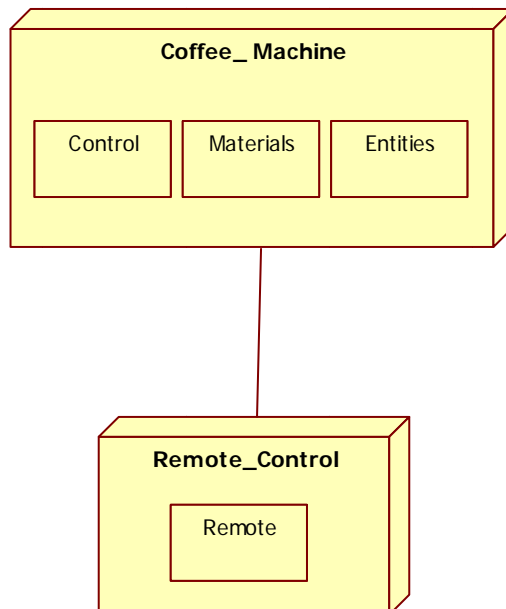
A blue rectangular screen with the text "Insert serial number" at the top. Below this text is a white rounded rectangular input field with a blue border, containing the text "AS12DD". Below the input field is a grey rounded rectangular button with the text "Send".

- Coffee Maker

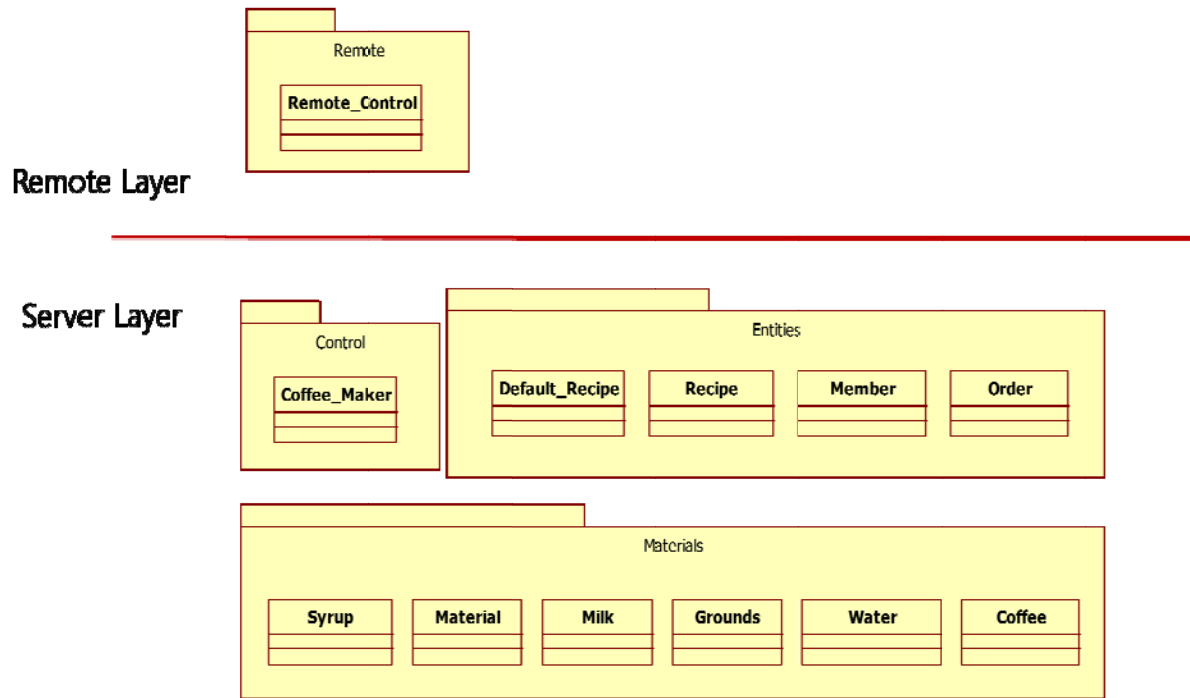
-

Activity 2143. Refine System Architecture

-Deployment Diagram

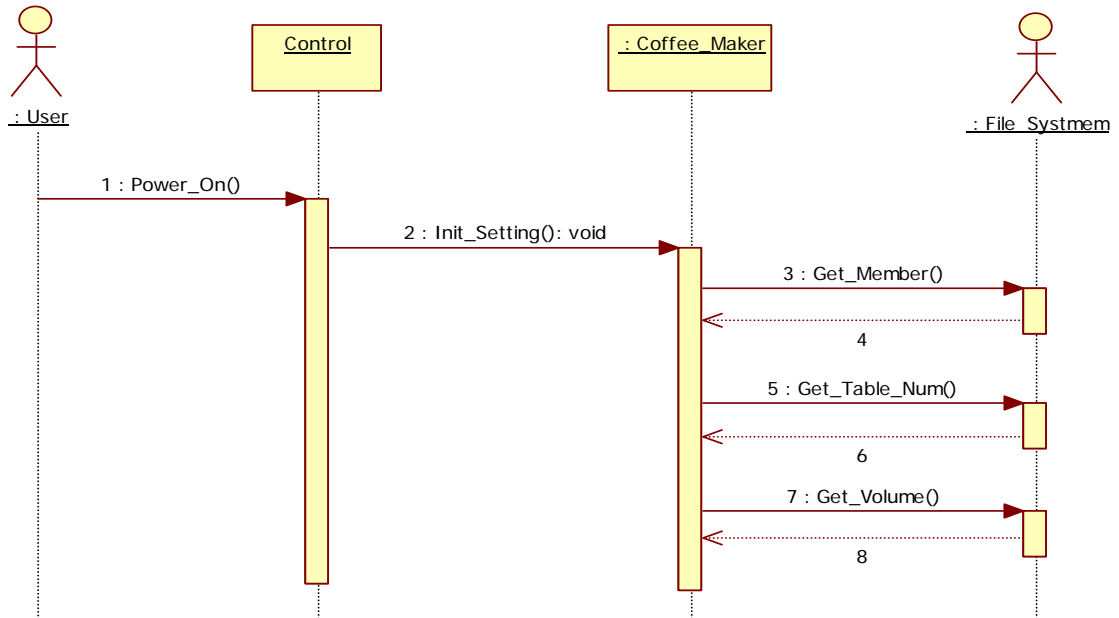


-Package Diagram

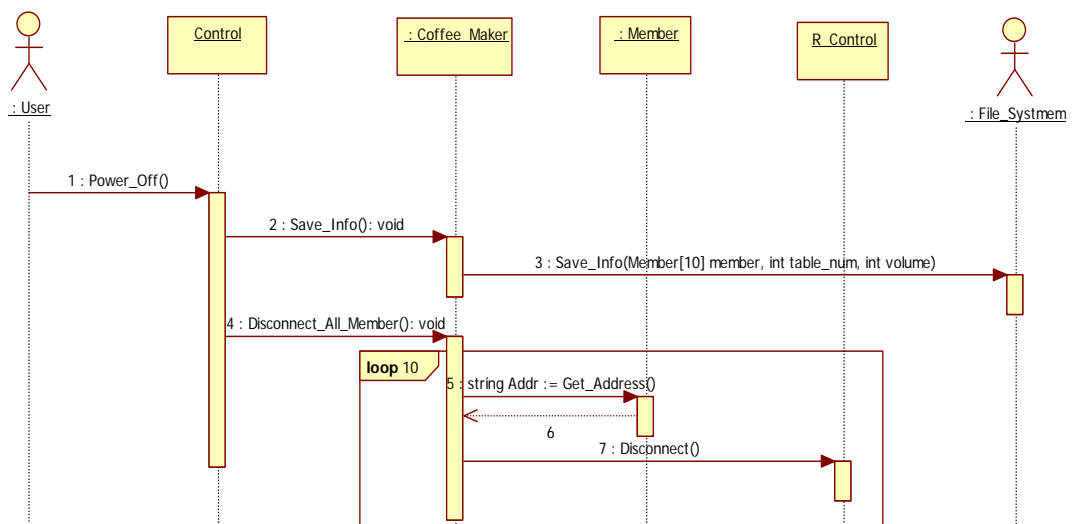


Activity 2144. Define Interaction Diagrams

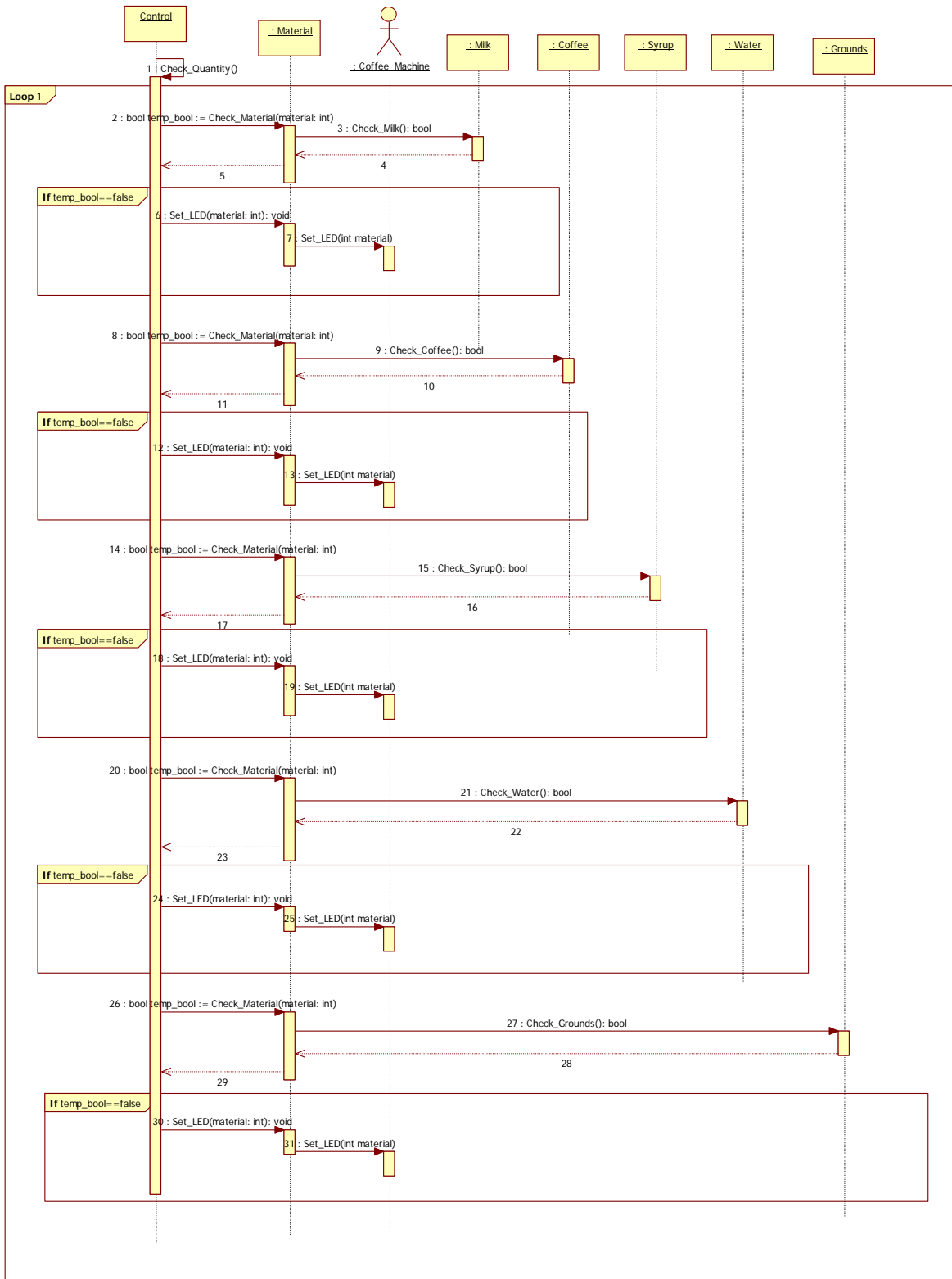
1. Power_On



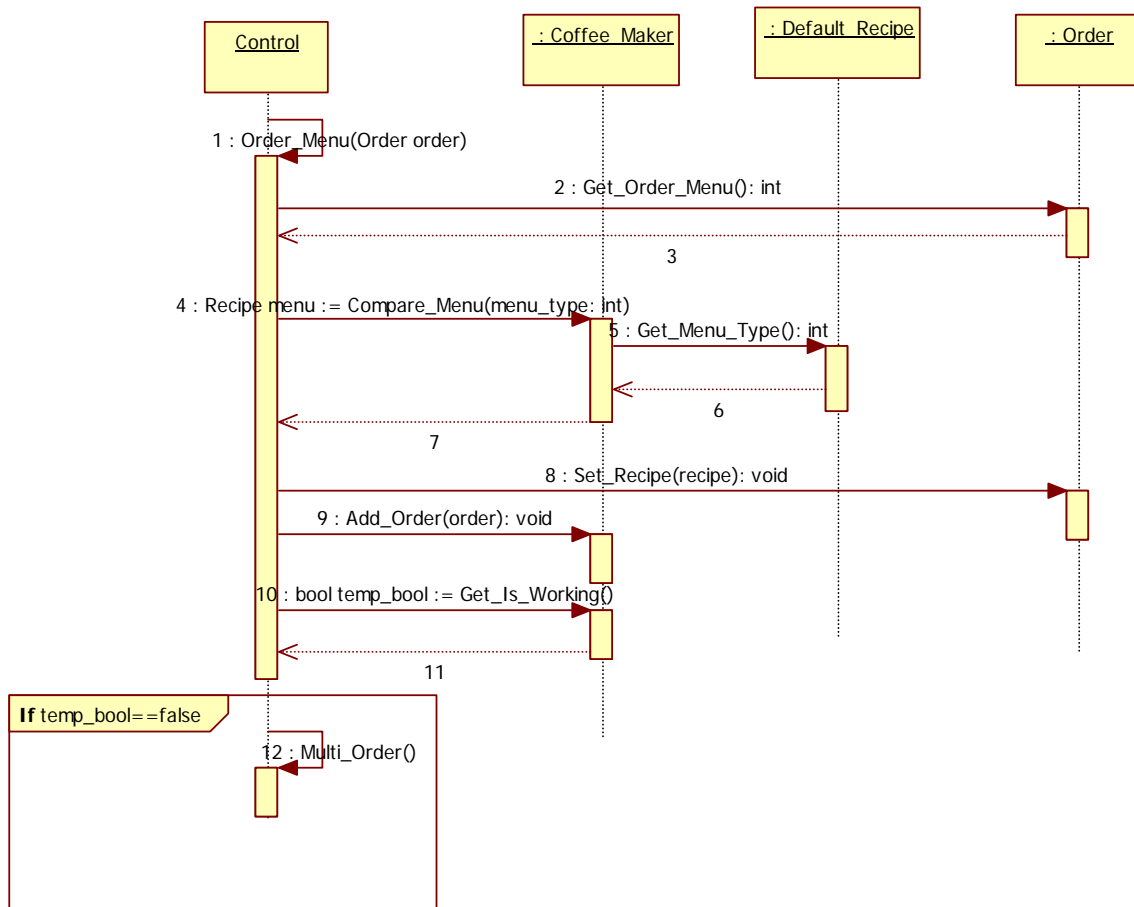
2. Power_Off



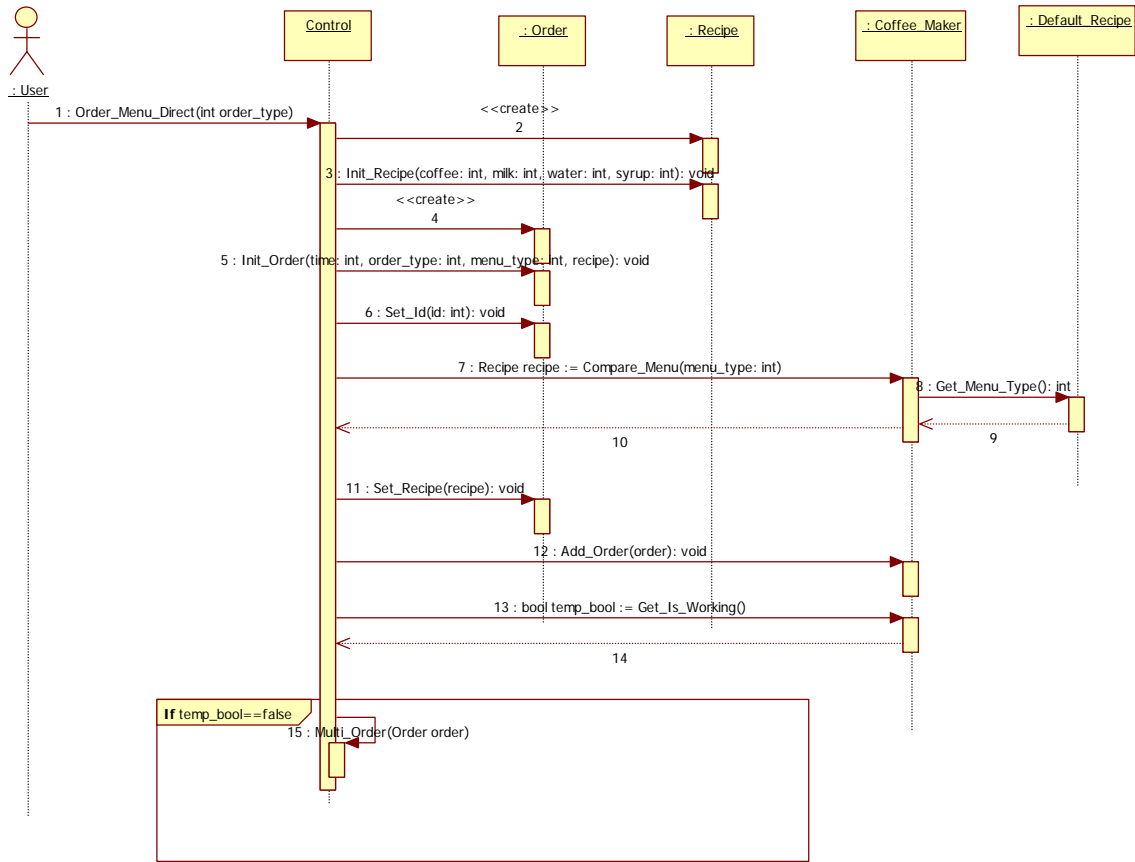
3. Check_Quantity



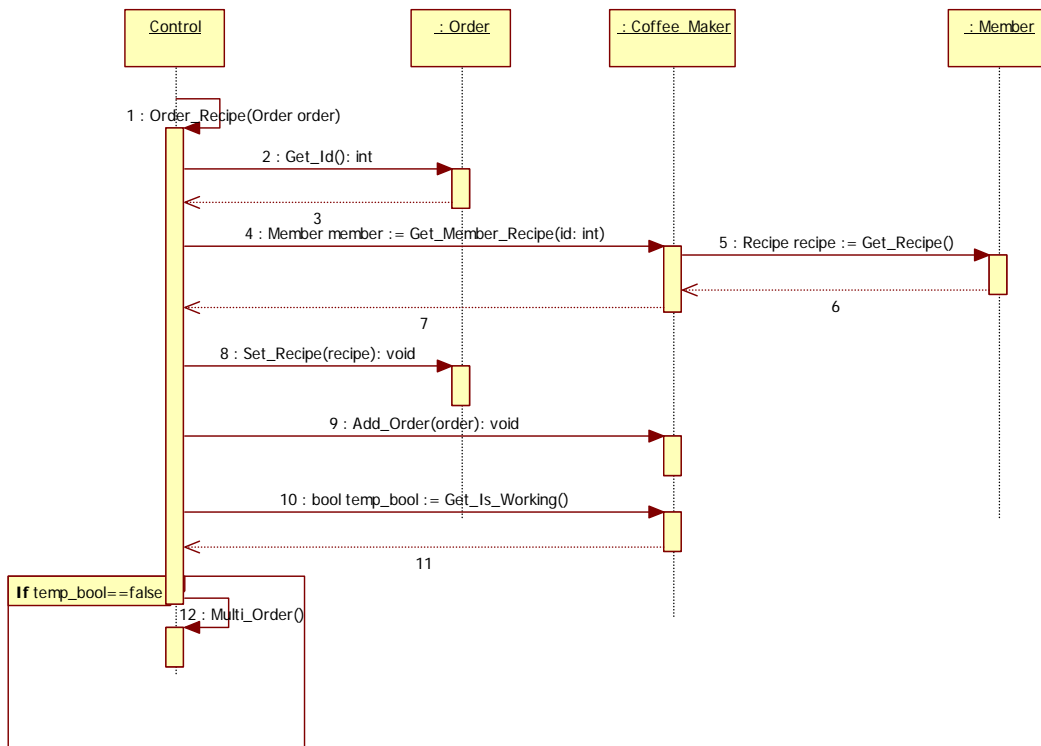
4. Order_Menu



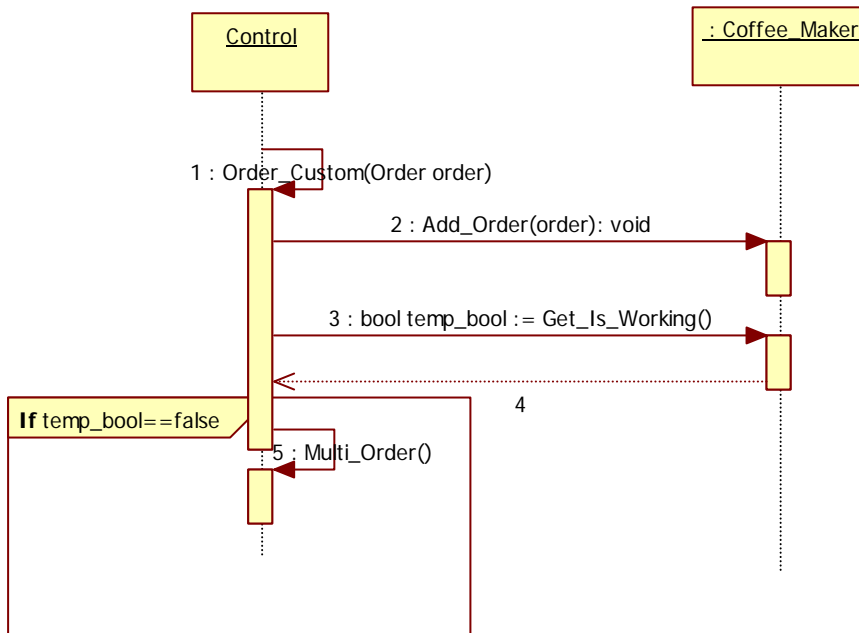
5. Order_Menu_Direct



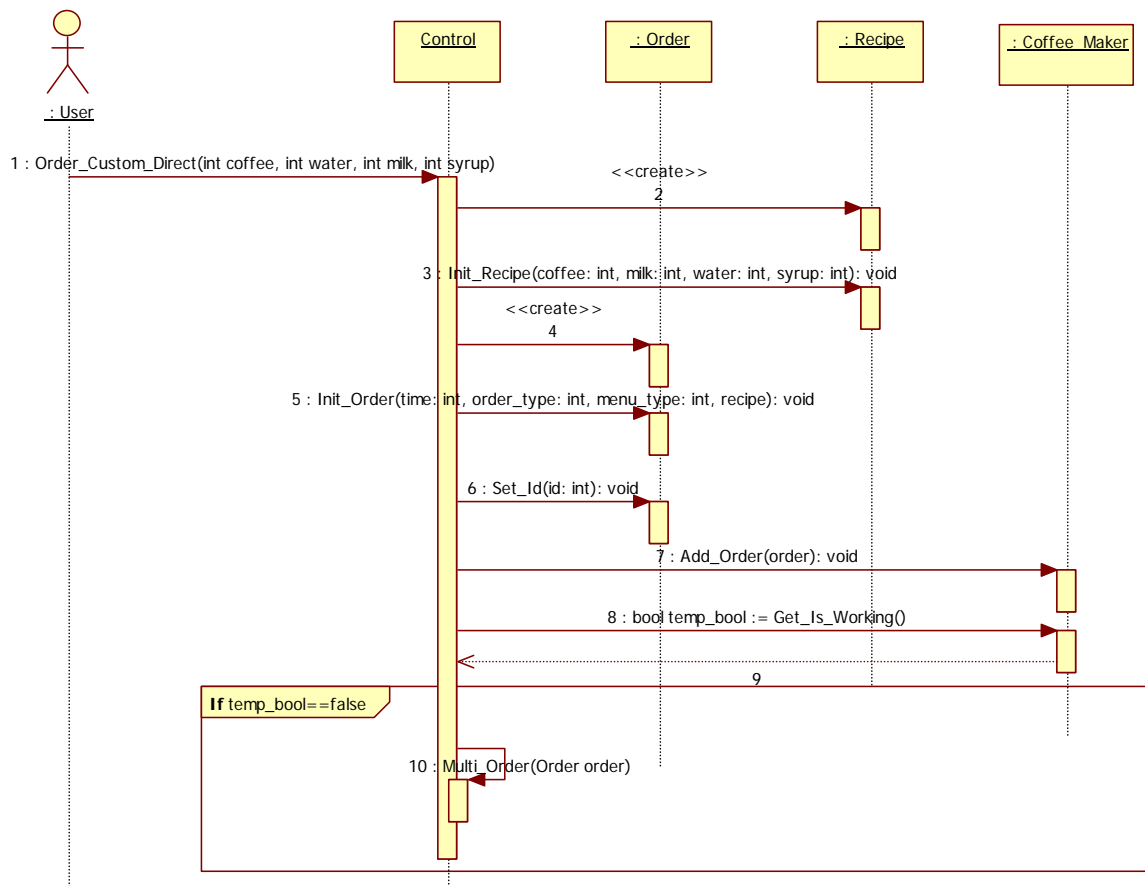
6. Order_Recipe



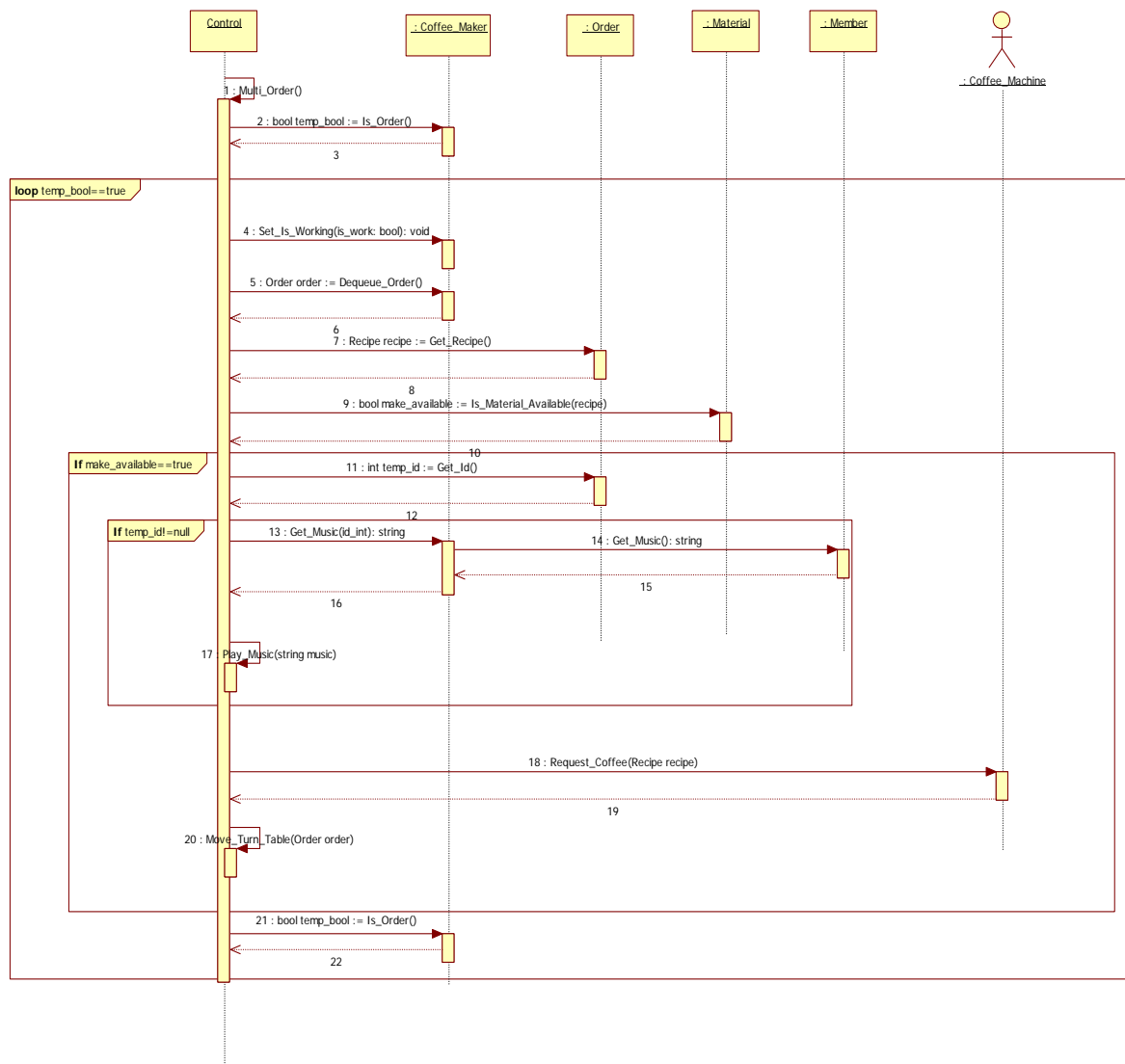
7. Order_Custom



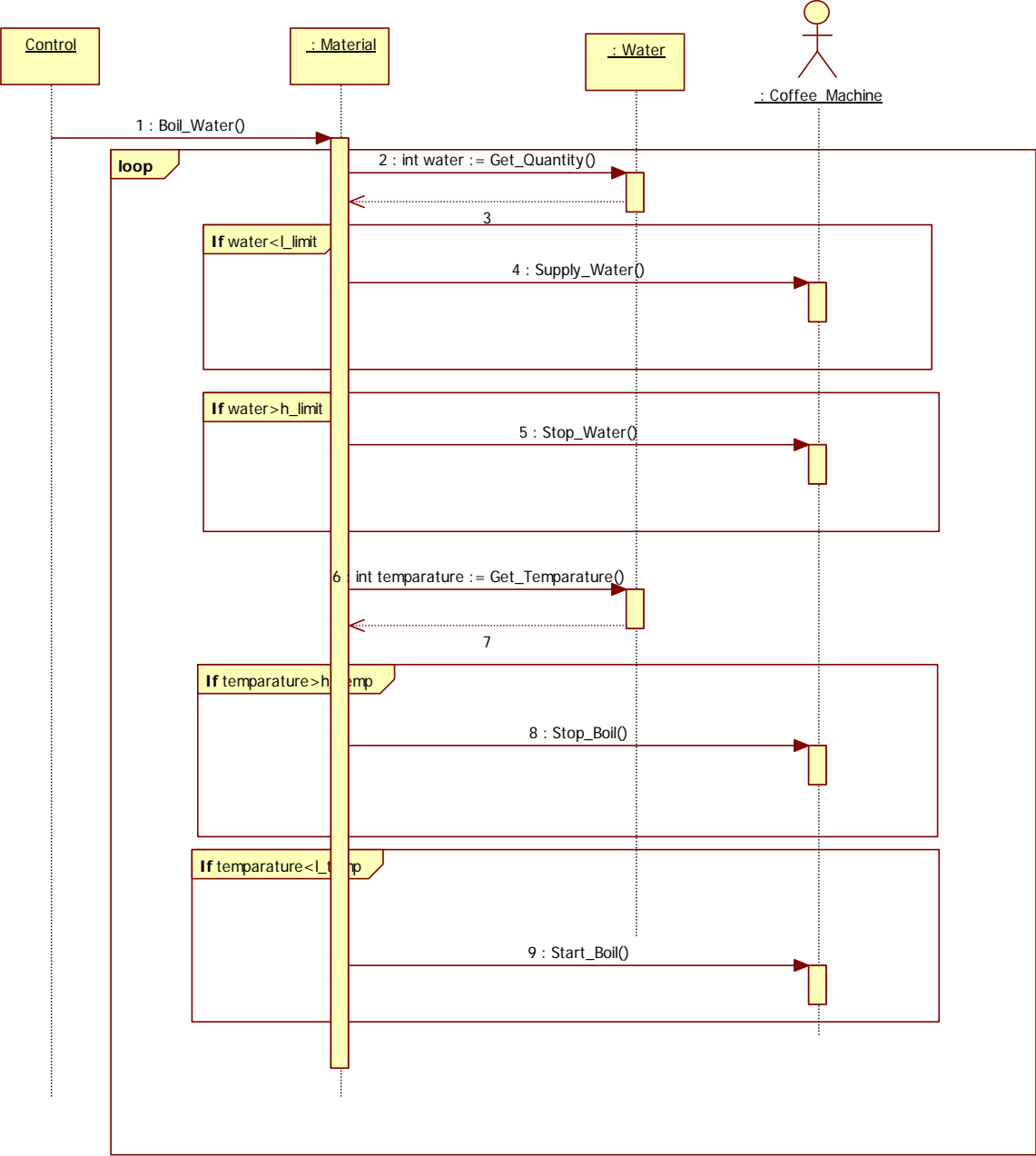
8. Order_Custom_Direct



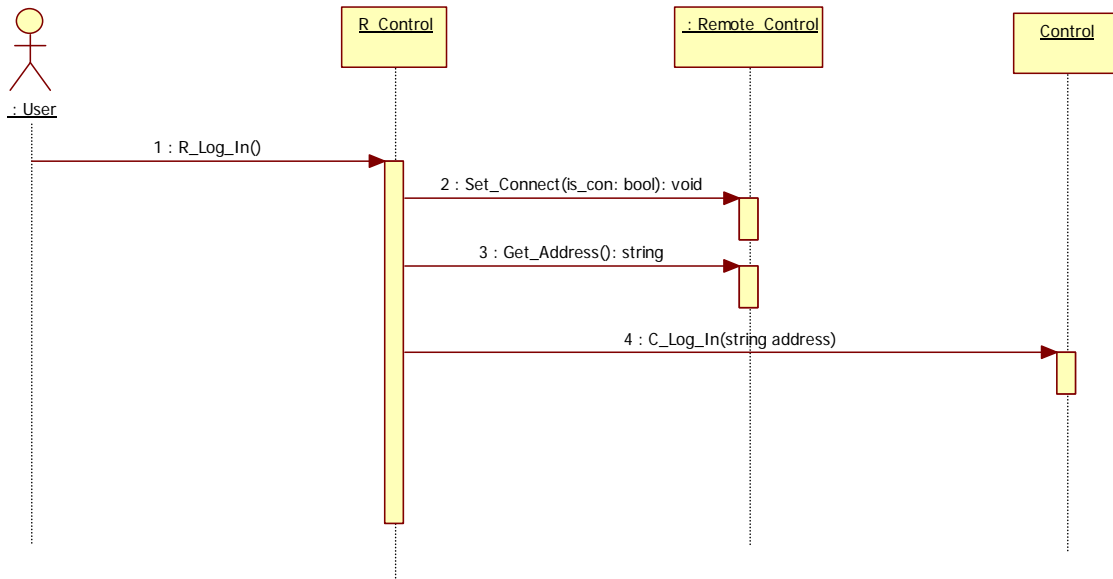
9. Multi_Order



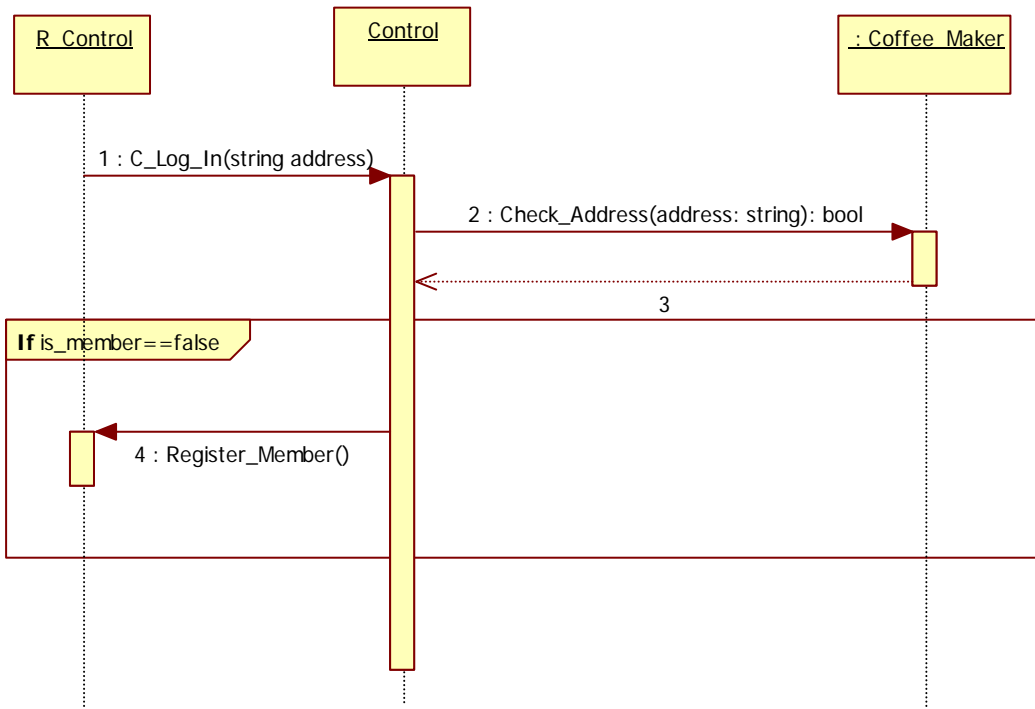
10. Boil_Water



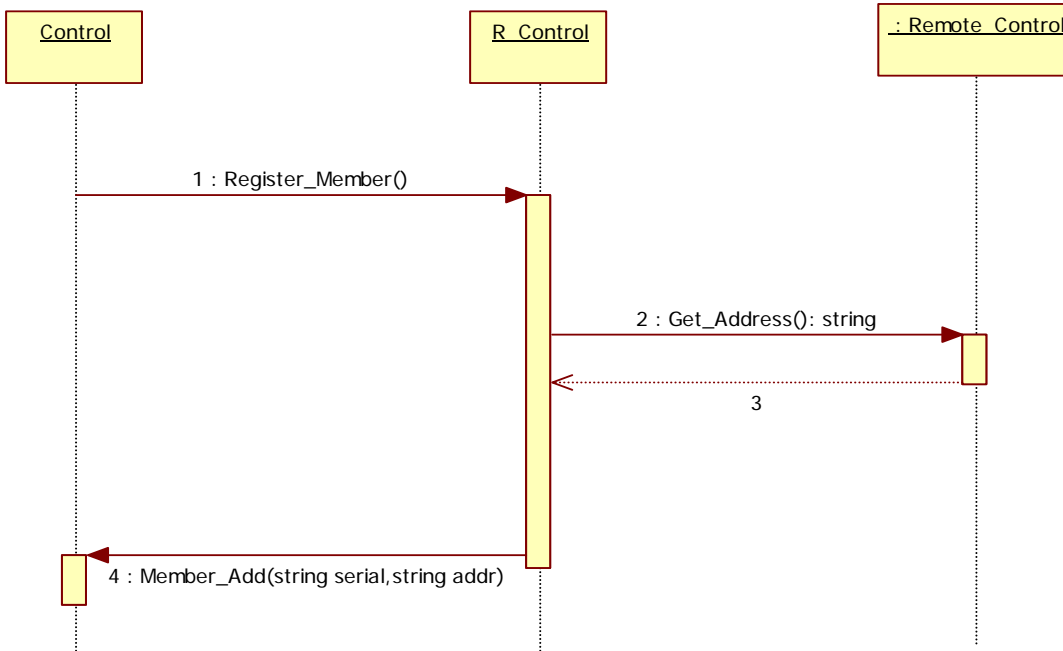
11. R_Log_In



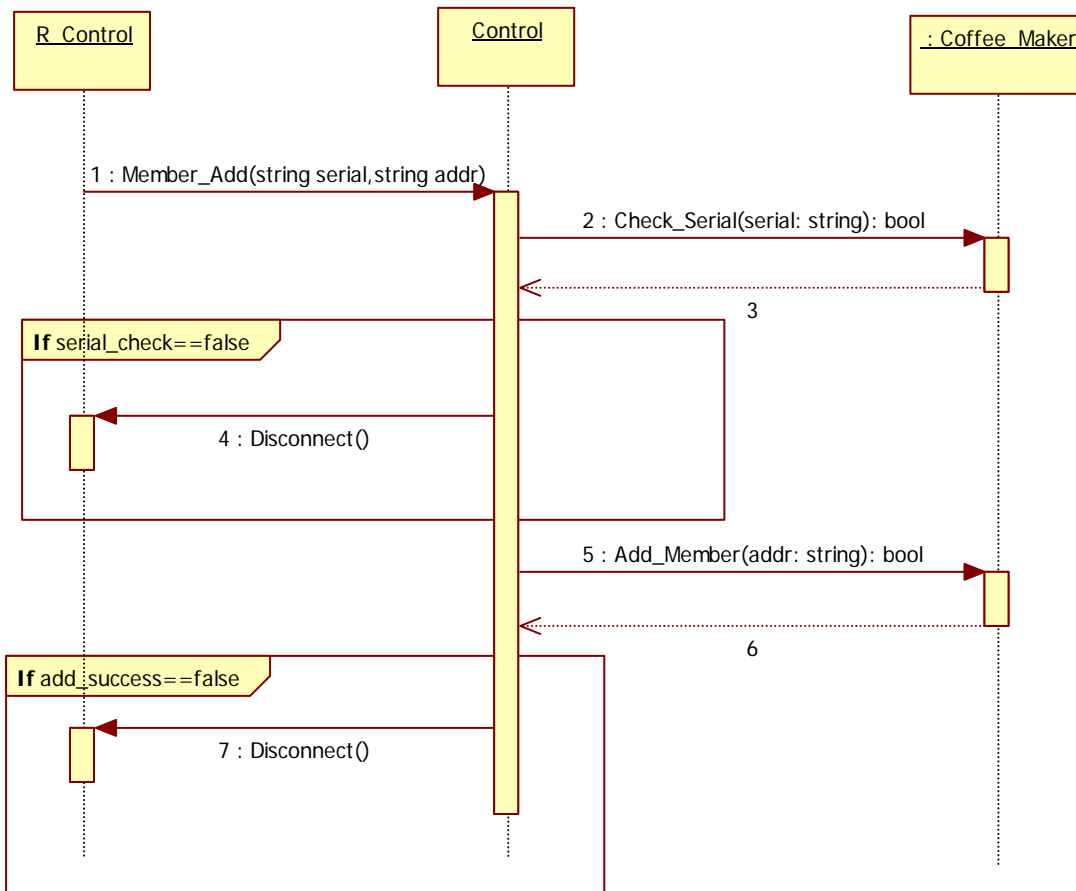
12. C_Log_In



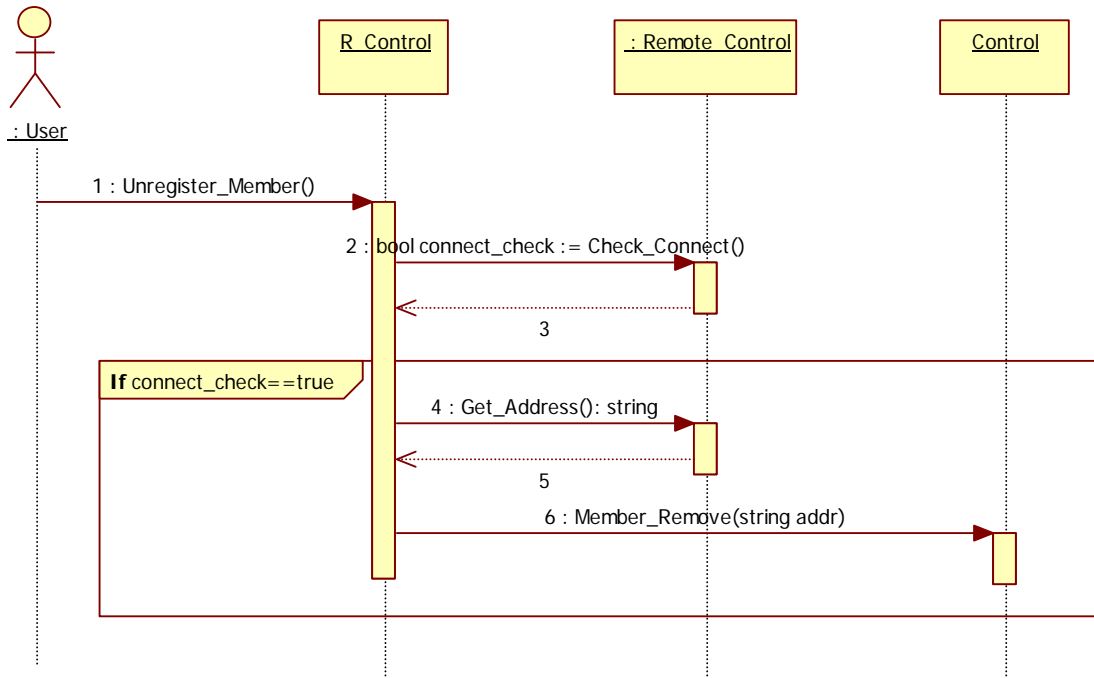
13. Register_Member



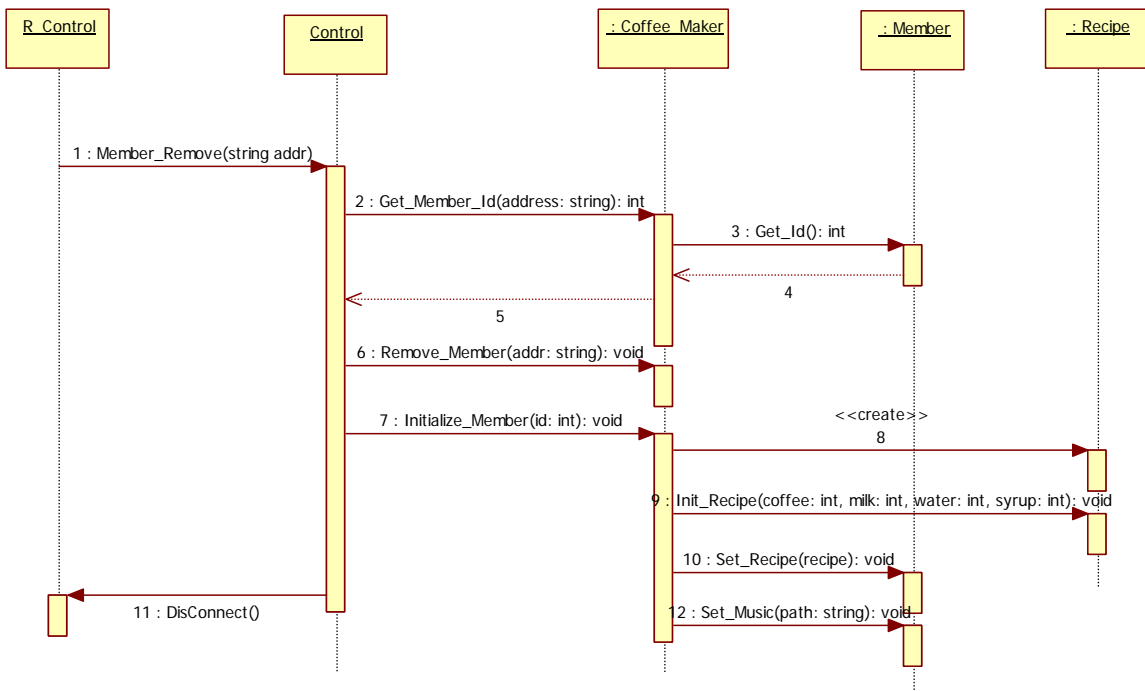
14. Member_Add



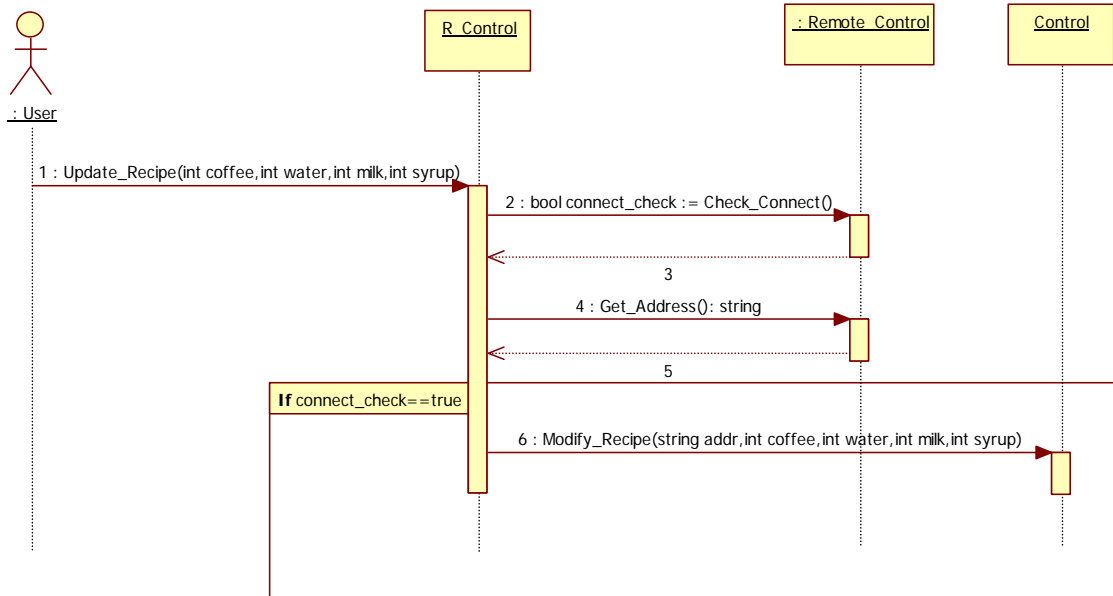
15. Unregister_Member



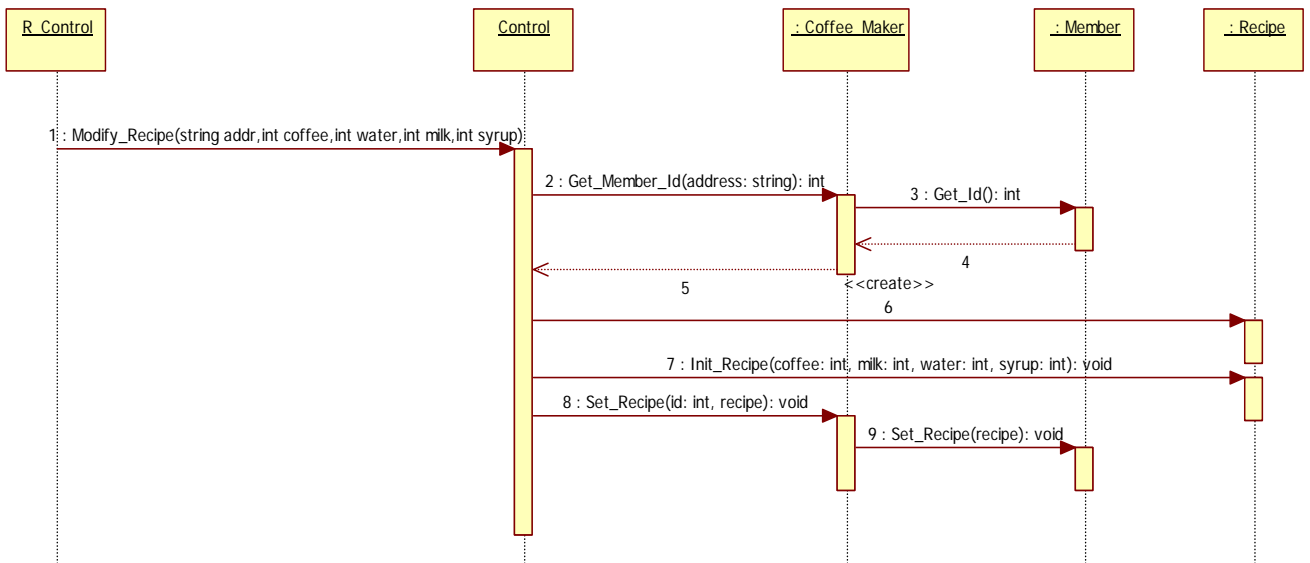
16. Member_Remove



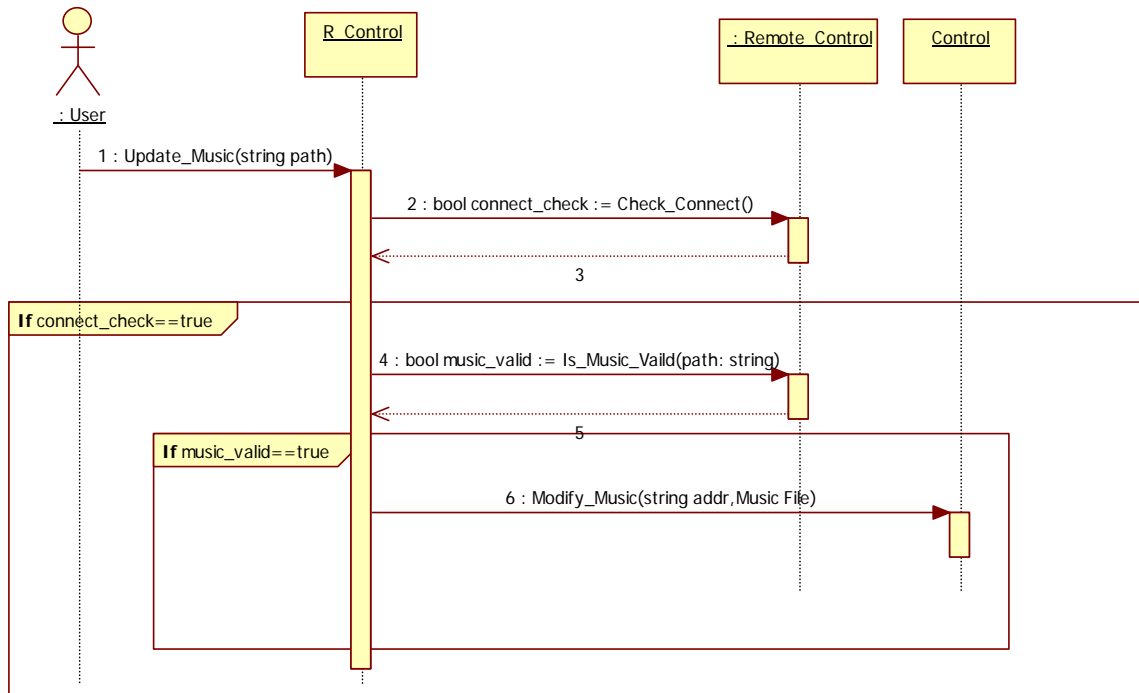
17. Update_Recipe



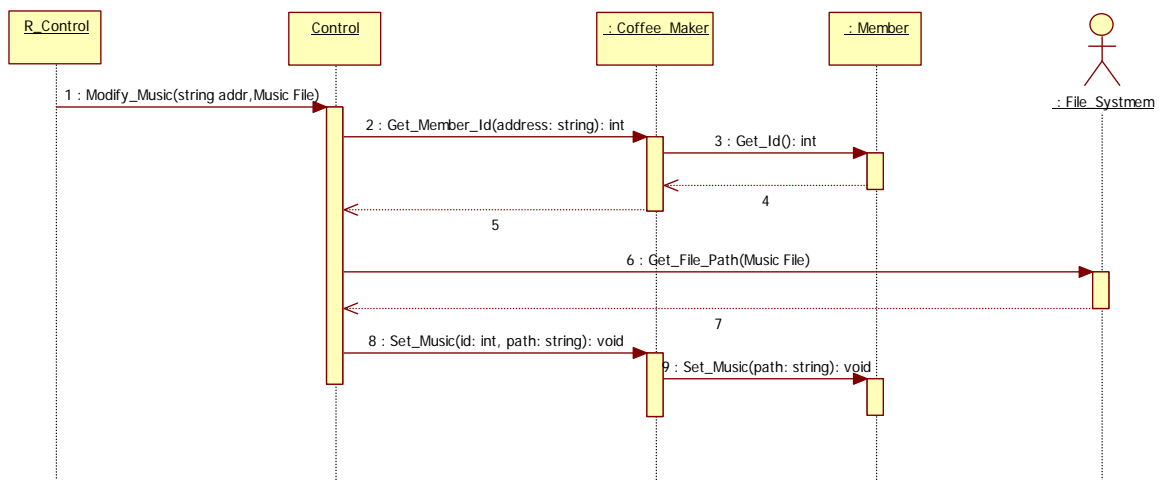
18. Modify_Recipe



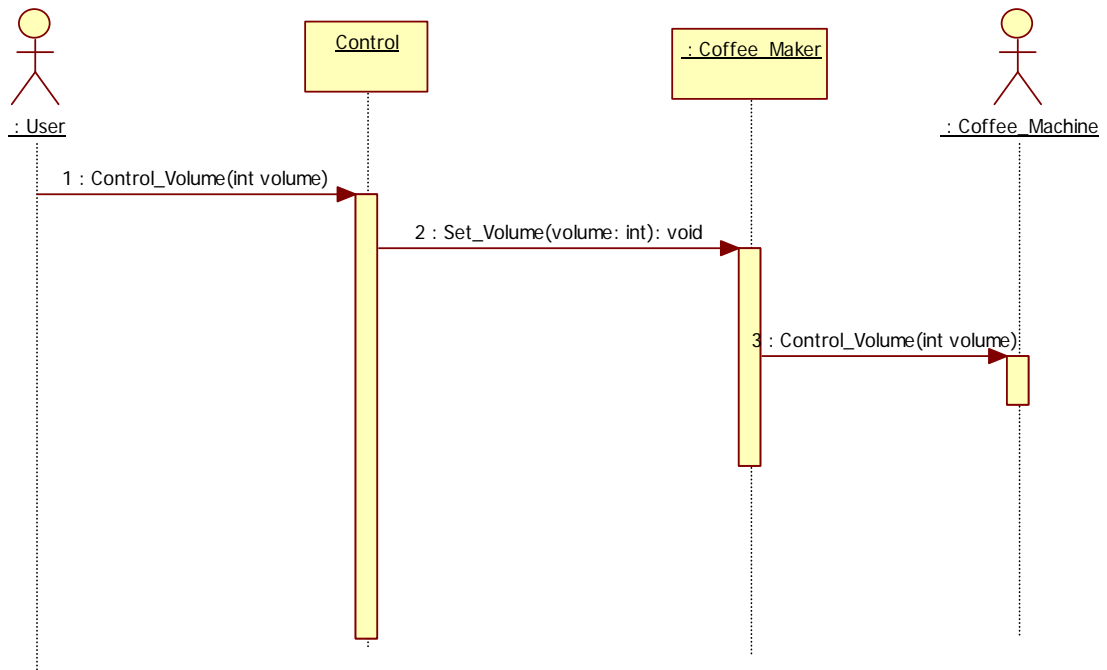
19. Update_Music



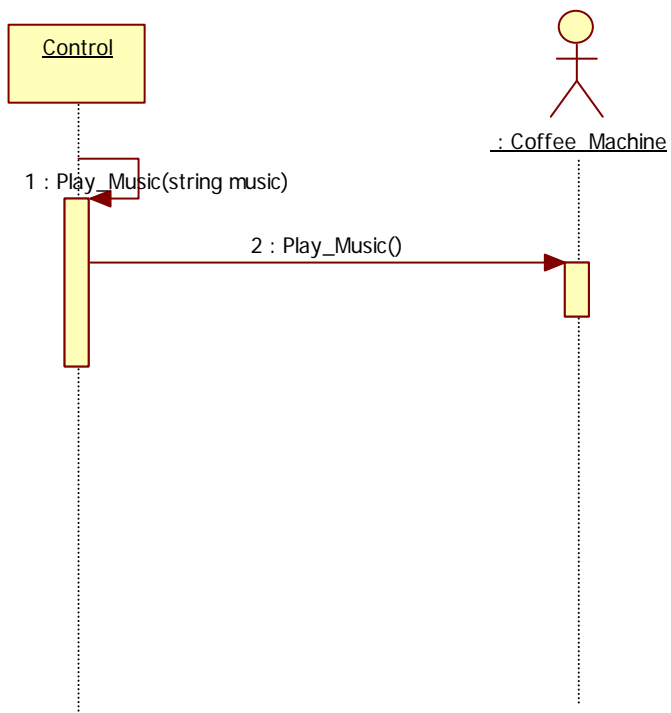
20. Modify_Music



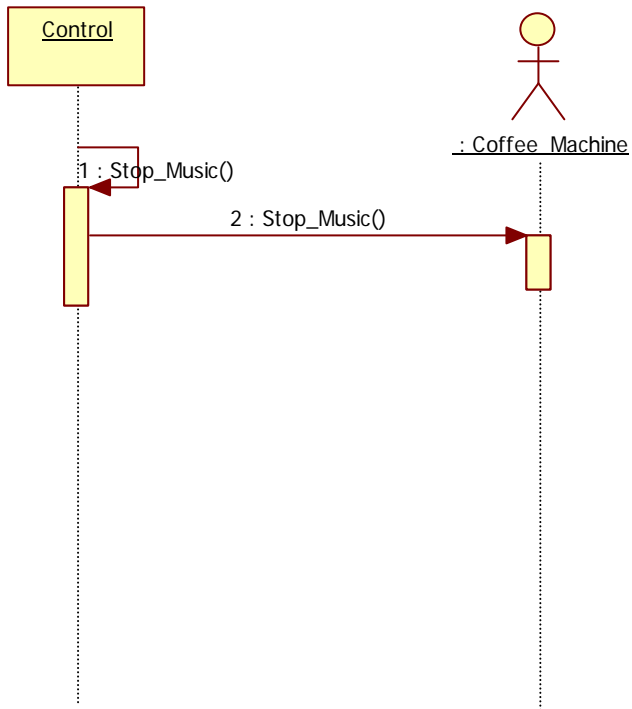
21. Control_Volume



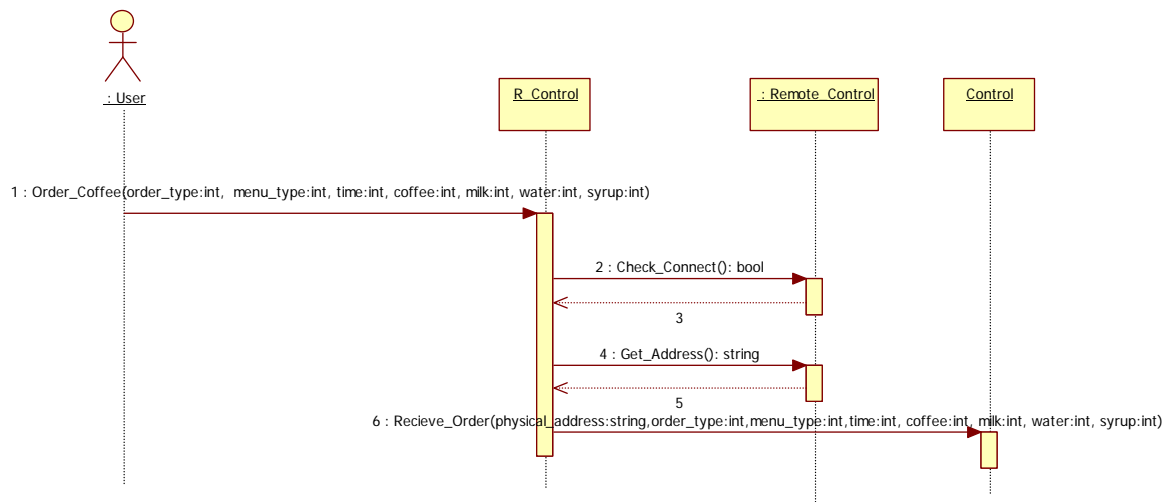
22. Play_Music



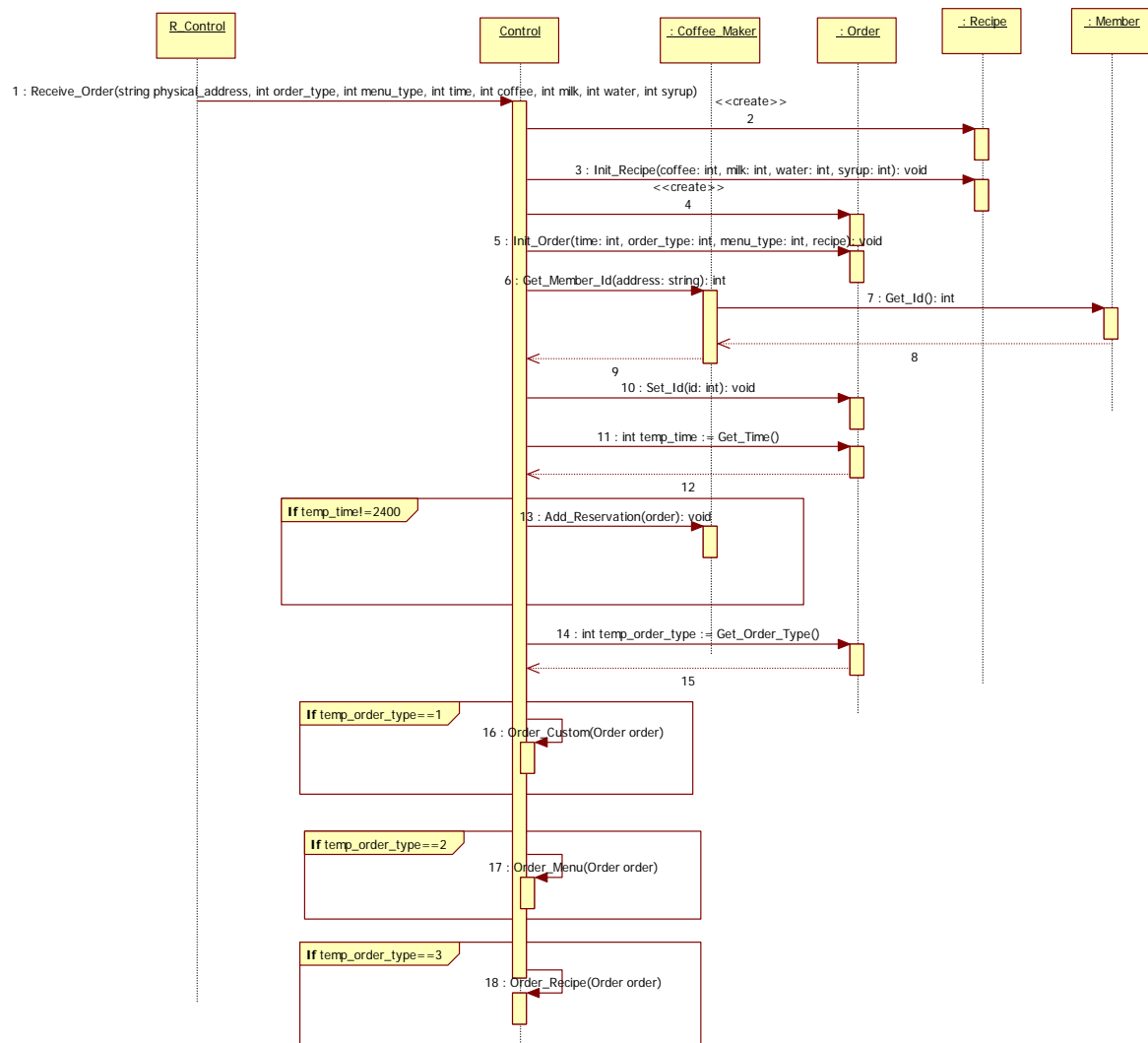
23. Stop_Music



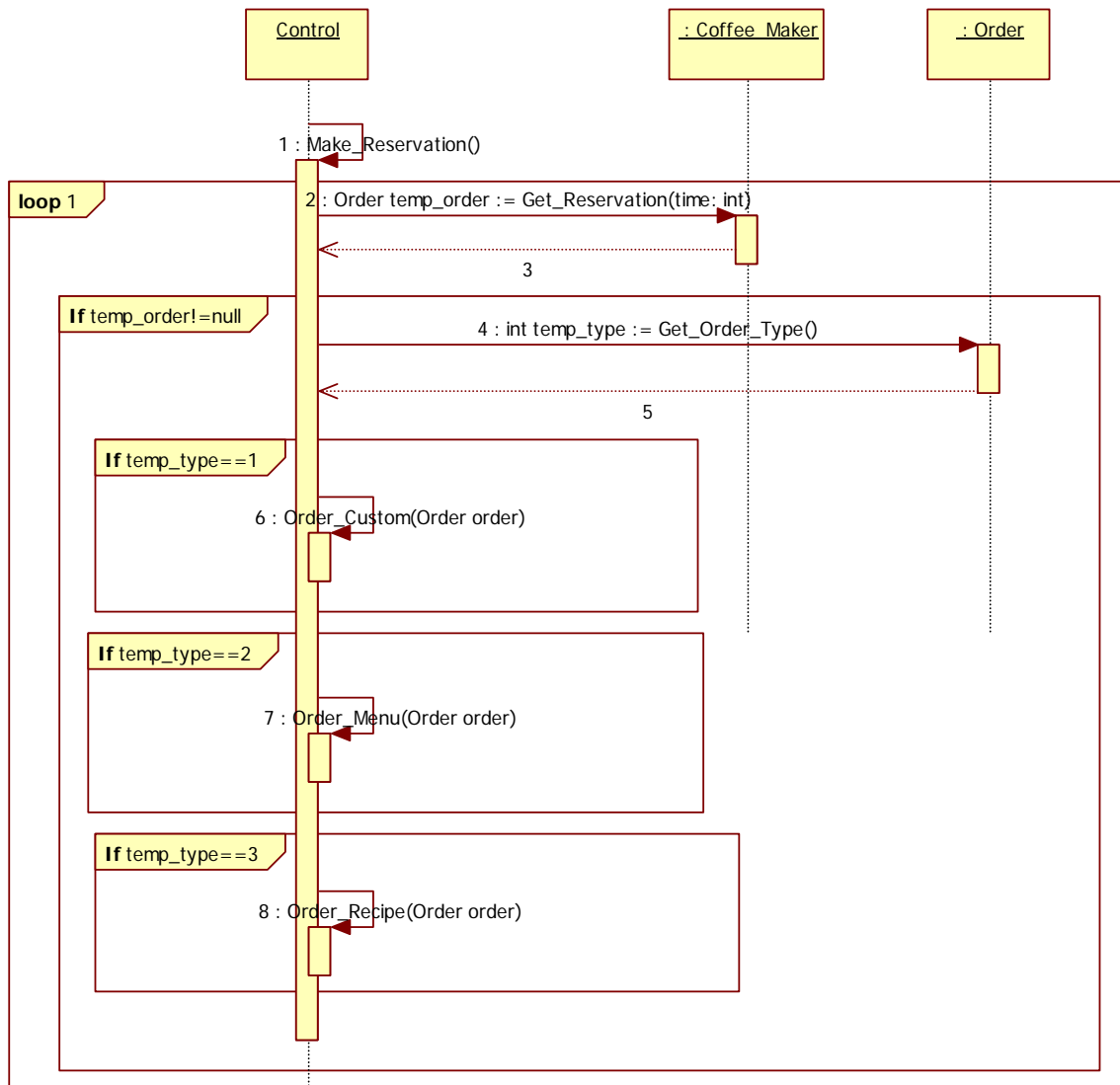
24. Order_Coffee



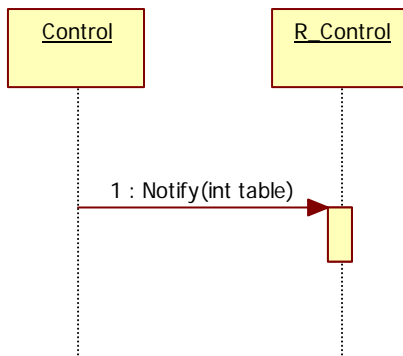
25. Receive_Order



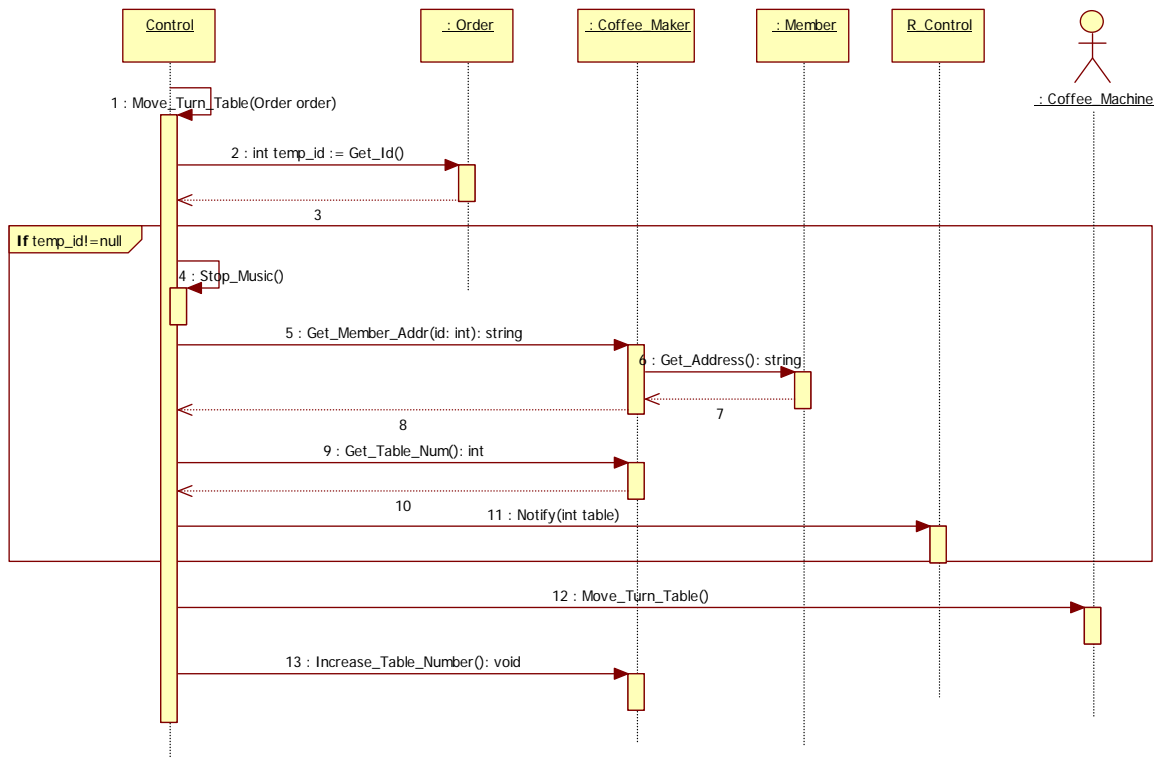
26. Make_Reservation



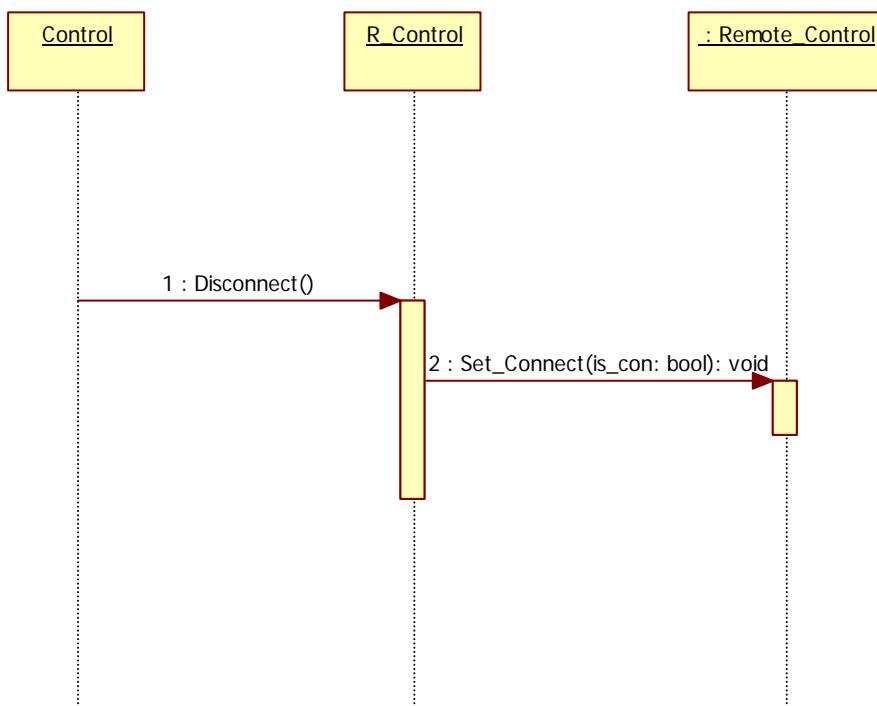
27. Notify



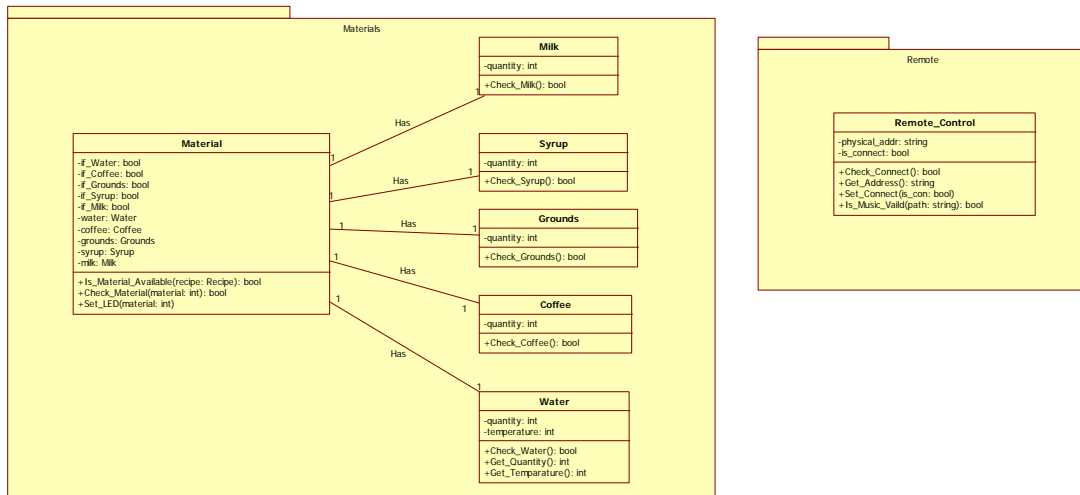
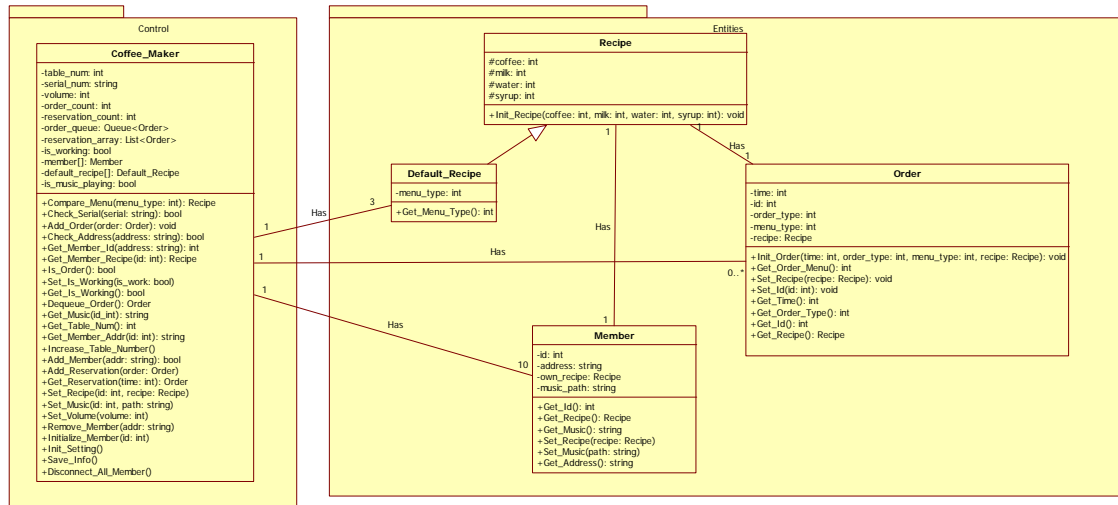
28. Move_Turn_Table



29. Disconnect



Activity 2145. Define Design Class Diagrams



Activity 2146. Define Database Schema