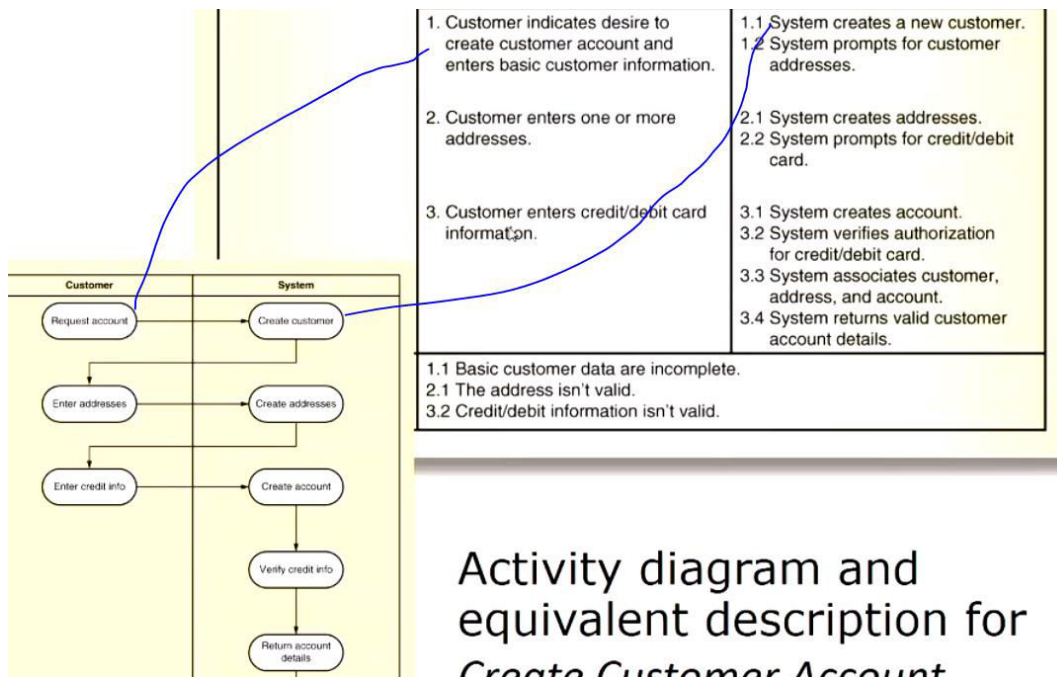


- [Analysis \*Phase\*]
  
- [CRUD technique]: Explain? Methods of use
  - [Verifying use cases]: Explain
  - [Cross checking use cases and domain classes]: Explain
  
- [Full developed use case descriptions]: Explain? Components
  - [Use-case name]: Explain
  - [Scenario]: Explain
  - [Triggering event]: Explain
  - [Brief description]: Explain
  - [Actors]: Explain
  - [Related use cases]: Explain
  - [Stakeholders]: Explain
  - [Preconditions]: Explain
  - [Postconditions]: Explain
  - [Flow of activities]: Explain
  - [Exception conditions]: Explain
  
- [Activity diagram]: Explain
  
- [System sequence diagram]: Explain? Notation
  - [Input messages]: Explain
  - [Return]: Explain
  - [Loop frame]: Explain
  - [Opt frame]: Explain
  - [Alt frame]: Explain
  - [Message]: Explain
- [Steps for creating SSD]: Explain
  
- [State machine diagram]: Explain? Components? Steps
  
- **[How to create activity diagram]: Explain**
- **[How to fully developed use case description]: Explain**
- **[How to system sequence diagram]: Explain**
- **[How to state machine diagram]: Explain**

## 1. Activity diagram

### Important points to create an activity diagram

- Think of activity diagram as → summarised flow of activities
- A *Single activity* in flow activities → *Single action* in activity diagram
- Rarely include a prompt



## 2. Fully developed use case description

### Important points to create flow of activities

- Again I only looked at sheet once and went my own separate way
- **Don't go into specific about what is sent. With the system sequence diagram:**
  - **It actually looks into and outlines in detail the input and output passed between actor and system**

### Important points to scenario

- Very similar to the use case name except with the inclusion of the words web/online and anything that pinpoints use case EG
  - Find caterers → Web Customer finds caterers
  - Create new order → Create new web order
  - Create new order → Create new telephone order

<b>Use Case Name:</b>	<i>Create new order</i>	
<b>Scenario:</b>	Create new telephone order	
<b>Triggering Event:</b>	Customer telephones RMO to purchase items from the catalog.	
<b>Brief Description:</b>	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
<b>Actors:</b>	Telephone sales clerk.	
<b>Related Use Cases:</b>	Includes: <i>Check item availability</i> .	
<b>Stakeholders:</b>	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
<b>Preconditions:</b>	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
<b>Postconditions:</b>	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
<b>Flow of Activities:</b>	<b>Actor</b>	<b>System</b>
	<ol style="list-style-type: none"> <li>1. Sales clerk answers telephone and connects to a customer.</li> <li>2. Clerk verifies customer information.</li> <li>3. Clerk initiates the creation of a new order.</li> <li>4. Customer requests an item be added to the order.</li> <li>5. Clerk verifies the item (<i>Check item availability</i> use case).</li> <li>6. Clerk adds item to the order.</li> <li>7. Repeat steps 4, 5, and 6 until all items are added to the order.</li> <li>8. Customer indicates end of order; clerk enters end of order.</li> <li>9. Customer submits payment; clerk enters amount.</li> </ol>	<ol style="list-style-type: none"> <li>2.1 Display customer information.</li> <li>3.1 Create a new order.</li> <li>5.1 Display item information.</li> <li>6.1 Add an order item.</li> <li>8.1 Complete order.</li> <li>8.2 Compute totals.</li> <li>9.1 Verify payment.</li> <li>9.2 Create order transaction.</li> <li>9.3 Finalize order.</li> </ol>
<b>Exception Conditions:</b>	<ol style="list-style-type: none"> <li>2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case.</li> <li>2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative.</li> <li>4.1 If an item is not in stock, then customer can <ol style="list-style-type: none"> <li>a. choose not to purchase item, or</li> <li>b. request item be added as a back-ordered item.</li> </ol> </li> <li>9.1 If customer payment is rejected due to bad-credit verification, then <ol style="list-style-type: none"> <li>a. order is canceled, or</li> <li>b. order is put on hold until check is received.</li> </ol> </li> </ol>	

<b>Use Case Name:</b>	Create new order	
<b>Scenario:</b>	Create new Web order	
<b>Triggering Event:</b>	Customer logs on to the RMO Web site and requests to purchase an item.	
<b>Brief Description:</b>	Customer logs on and requests the new order form. The customer searches the catalog online and purchases items from the catalog. The system adds the purchased items to the order. At the end the customer enters credit-card information.	
<b>Actors:</b>	Customer.	
<b>Related Use Cases:</b>	Includes: <i>Register new customer, Check item availability.</i>	
<b>Stakeholders:</b>	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
<b>Preconditions:</b>	Catalog, Products, and Inventory items must exist for requested items.	
<b>Postconditions:</b>	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
<b>Flow of Activities:</b>	<b>Actor</b>	<b>System</b>
	<ol style="list-style-type: none"> <li>1. Customer connects to the RMO home page and then links to the order page.</li> <li>2. If this is a new customer, then customer links to the customer account page and adds the appropriate information to establish a customer account.</li> <li>2a. If existing customer, customer logs on.</li> <li>3. Customer searches catalog.</li> <li>4. When customer finds the correct item, he/she requests it be added to the order.</li> <li>5. Repeat steps 3 and 4.</li> <li>6. Customer requests end of order.</li> <li>7. Customer makes any changes.</li> <li>8. Customer requests payment screen.</li> <li>9. Customer enters payment information.</li> </ol>	<ol style="list-style-type: none"> <li>2.1 Create new customer record.</li> <li>2a.1 Validate customer account.</li> <li>2.2 Create a new shopping cart order; display order form with catalog frame.</li> <li>3.1 Display products from catalog based on searches and selections.</li> <li>4.1 Add item to shopping cart order.</li> <li>6.1 Display shopping cart items, with totals and amounts due; edit and submit buttons.</li> <li>8.1 Display payment details screen.</li> <li>9.1 Accept payment, finalize order, send confirmation e-mail.</li> </ol>
<b>Exception Conditions:</b>	<ol style="list-style-type: none"> <li>4.1 If an item is not in stock, then customer can <ol style="list-style-type: none"> <li>a. choose not to purchase item, or</li> <li>b. request item be added as a back-ordered item.</li> </ol> </li> <li>8.1 If customer payment is rejected due to bad-credit verification, then <ol style="list-style-type: none"> <li>a. order is canceled, or</li> <li>b. order is put on hold until check is received.</li> </ol> </li> </ol>	

### 3. System sequence diagram

First step: Draw outline

Second step: Think about how use case would look like as activity diagram so break it down into steps (1→5)

Third step: Look at tutorial video with example

Fourth step: Look at lectures and notes on how to begin the diagram. *In this case always begin with indication you want to do something so you get a form.*

Fifth step: Break steps one piece after another piece. Don't assume all done in one step

Important points:

- **This actually goes into details about what information (Input/Output) is actually exchanged between the actor and system unlike the flow of activities section in the fully developed use case**
- Focus on the input/output that the actor sends and the system receives and responds back
- The **attributes** in domain class diagram → **parameters** in SSD
- Try to use same attributes as those in the SSD

#### Format for Input message

- Verb-Noun(Input1/attribute1, Input2/attribute2...) EG
  - addItem(ItemID, quantity)
    - Notice addItem is a use case and the parameters are attributes

#### Format for return message

- List the attributes system returns to user OR describe it in a sentence
- If any doubt display .... Confirm...
- No parameters

### 4. State machine diagram

Catering job states may include → **Completed**, Accepted, Rejected, Cancelled, **Closed?** **Open**, **Assigned**

Closed → If no quotes are received for catering job it closes

Completed → When job is completed by the caterer

Open → Job request is submitted and ready for people to fill

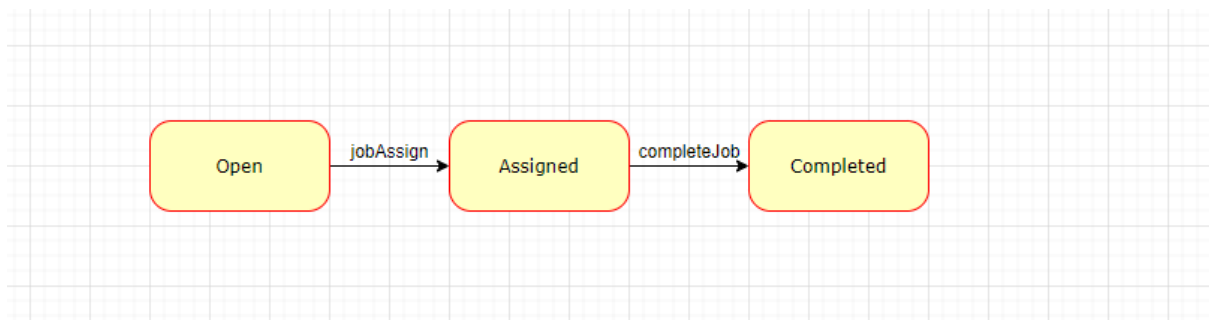
Assigned → Job request is assigned to a caterer

Core State- Transition causing exit:

Open – jobAssign

Assigned –jobComplete

Completed – no exit transition defined



- 1) Catering job states may include → [States]. During each state refer back to use case see if it makes sense and look at object in UML diagram
- 2) Open [state] → [describe what it does]
- 3) Refer to the lecture for steps + tutorial worksheet
- 4) So draw list of states and transition but the core
- 5) Continue with step 1