Use Case Modelling II

CSIT883 System Analysis and Project Management

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Outline

- Fully Developed Use Case Description
- Activity Diagram
- System Sequence Diagram
- CRUD Analysis

Use Cases Modelling

- Requirements reflect users' need and specify business processes.
- Fact-finding skills are used in requirement discovery
- Two primary aspects of functional requirements: Use Cases and Problem Domain Classes
- Additional details about use cases needs to be documented.
  - With fully-developed use case description, activity diagram and system sequence diagram

Fully Developed Use Case Description

- *A list of use cases* and *use case diagrams* provide an overview of all the use cases for a system.
- **Use case descriptions** – a textual model that lists and describes the processing details for a use case.
- The brief use case description is a one-sentence description.
  - Gives enough details for very simple use cases
  - An intermediate step to build the fully developed use case description
  - E.g., a brief use case for "create customer account": *user/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record*

Fully Developed Use Case Description

The most formal method for documenting a use case

A typical **template** including:

Fully Developed Use Case Description

- The most formal method for documenting a use case
- A typical **template** including:
  - **Use case name** (verb-noun)
  - **Scenario** (if any) – a special case or more specific path in a use case

Fully Developed Use Case Description

- The most formal method for documenting a use case
- A typical **template** including:
  - **Use case name** (verb-noun)
  - **Scenario** (if any) – a special case or more specific path in a use case
  - **Triggering event** (based on event decomposition technique)
  - **Brief description** (written previously when use case was identified)
  - **Actors** – one or more users from use case diagrams

Fully Developed Use Case Description

- The most formal method for documenting a use case
- A typical **template** including:
  - **Use case name** (verb-noun)
  - **Scenario** (if any) – a special case or more specific path in a use case
  - **Triggering event** (based on event decomposition technique)
  - **Brief description** (written previously when use case was identified)
  - **Actors** – one or more users from use case diagrams
  - **Related use cases** (<<includes>> relationships in a use case diagram)
  - **Stakeholders** – who are interested parties other than specific actors

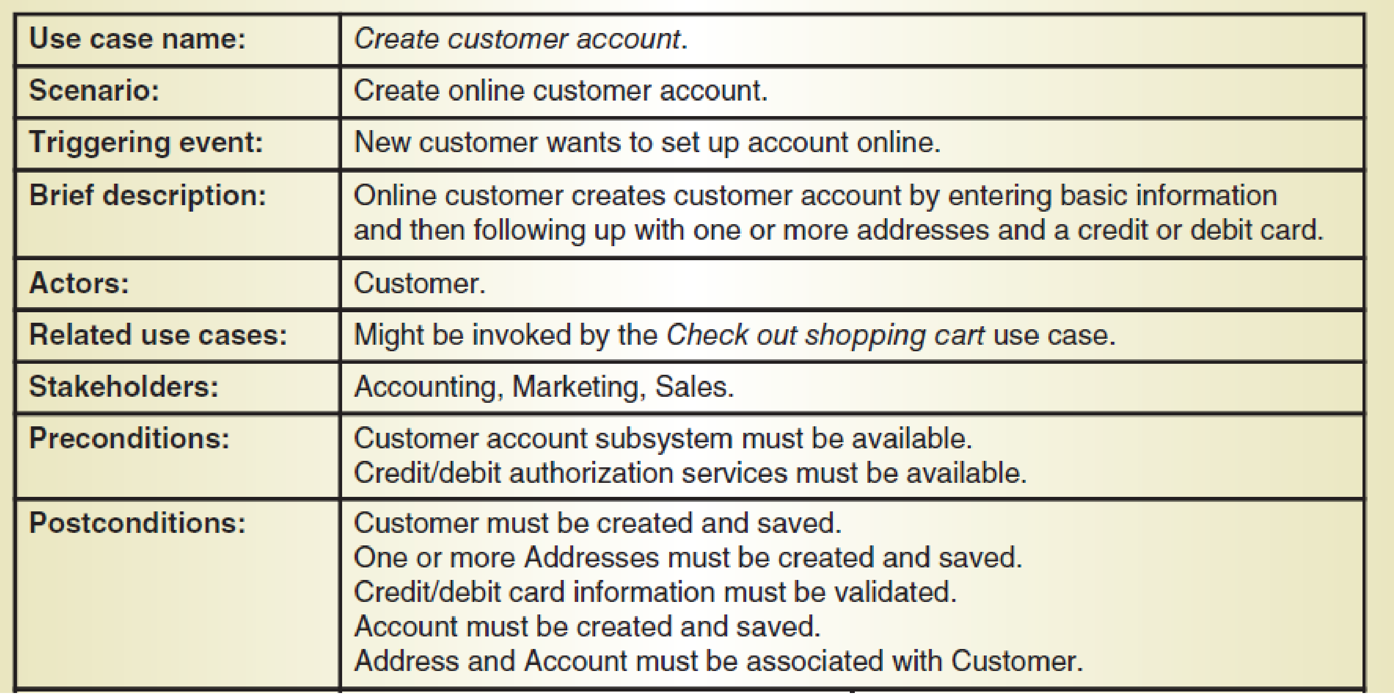
Fully Developed Use Case Description

- The most formal method for documenting a use case
- A typical **template** including:
  - **Use case name** (verb-noun)
  - **Scenario** (if any) – a special case or more specific path in a use case
  - **Triggering event** (based on event decomposition technique)
  - **Brief description** (written previously when use case was identified)
  - **Actors** – one or more users from use case diagrams
  - **Related use cases** (<<includes>> relationships in a use case diagram)
  - **Stakeholders** – who are interested parties other than specific actors
  - **Preconditions** – conditions that must be true before a use case begins
  - **Postconditions** – what must be true upon the successful completion of a use case
  - **Flow of activities** – in two columns: actor and system
  - **Exception conditions** – alternative activities that don’t belong to the main flow

Fully Developed Use Case Description

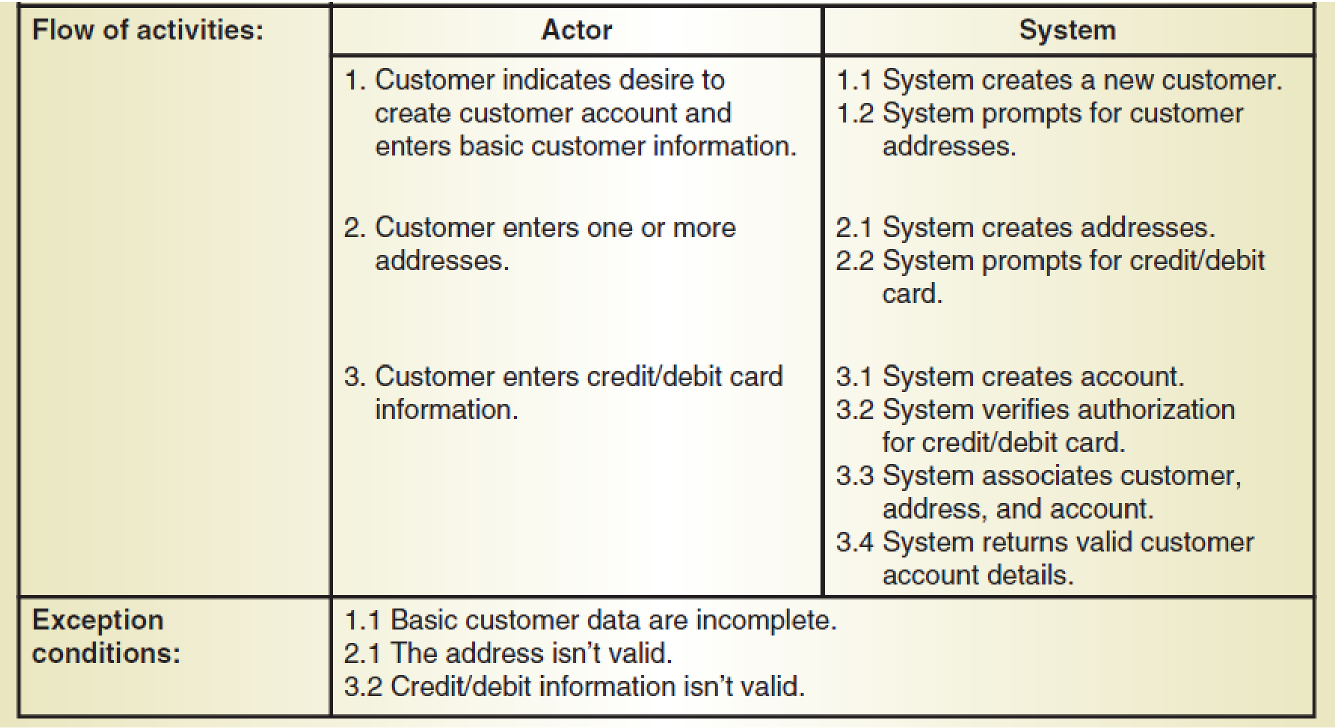
- The most formal method for documenting a use case
- A typical **template** including:
  - **Use case name** (verb-noun)
  - **Scenario** (if any) – a special case or more specific path in a use case
  - **Triggering event** (based on event decomposition technique)
  - **Brief description** (written previously when use case was identified)
  - **Actors** – one or more users from use case diagrams
  - **Related use cases** (<<includes>> relationships in a use case diagram)
  - **Stakeholders** – who are interested parties other than specific actors
  - **Preconditions** – conditions that must be true before a use case begins
  - **Postconditions** – what must be true upon the successful completion of a use case
  - **Flow of activities** – in two columns: actor and system
  - **Exception conditions** – alternative activities that don’t belong to the main flow

Fully Developed Use Case Description *Create customer account* (part 1)



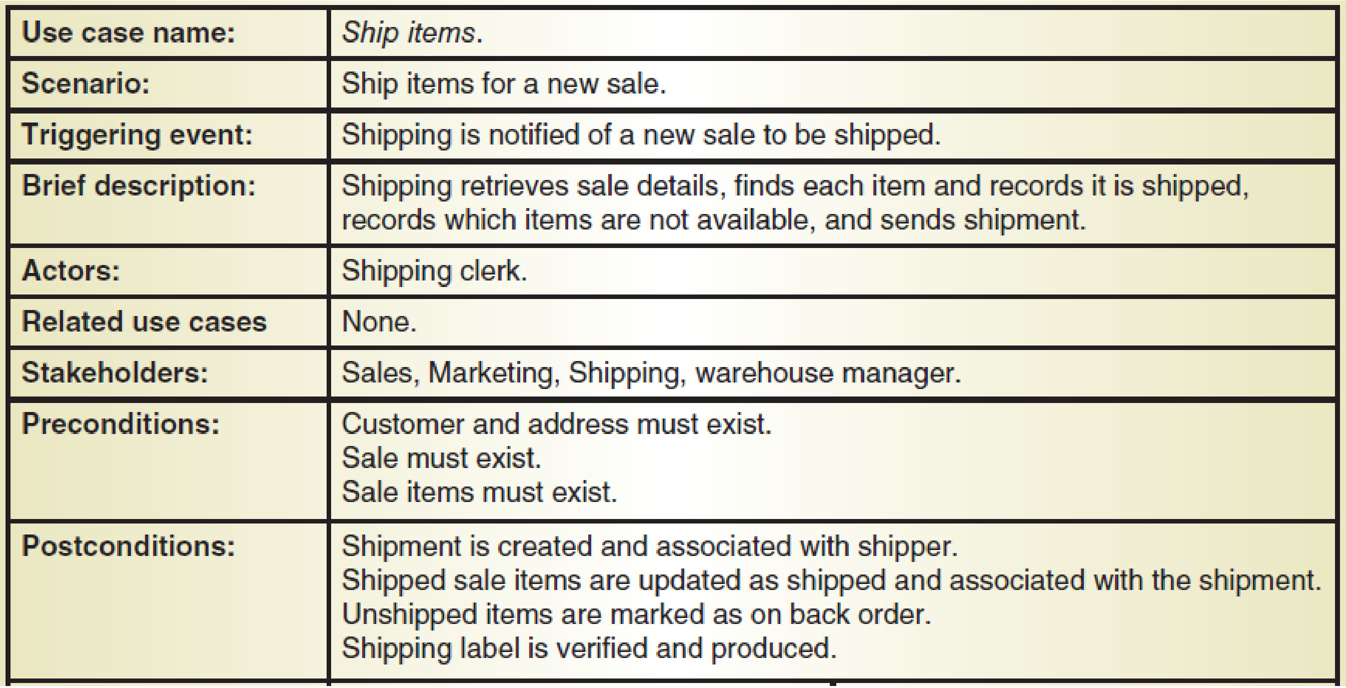
Use case name:	*Create customer account.*
Scenario:	Create online customer account.
Triggering event:	New customer wants to set up account online.
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.
Actors:	Customer.
Related use cases:	Might be invoked by the *Check out shopping cart* use case.
Stakeholders:	Accounting, Marketing, Sales.
Preconditions:	Customer account subsystem must be available. Credit/debit authorization services must be available.
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.

Fully Developed Use Case Description *Create customer account* (part 2)



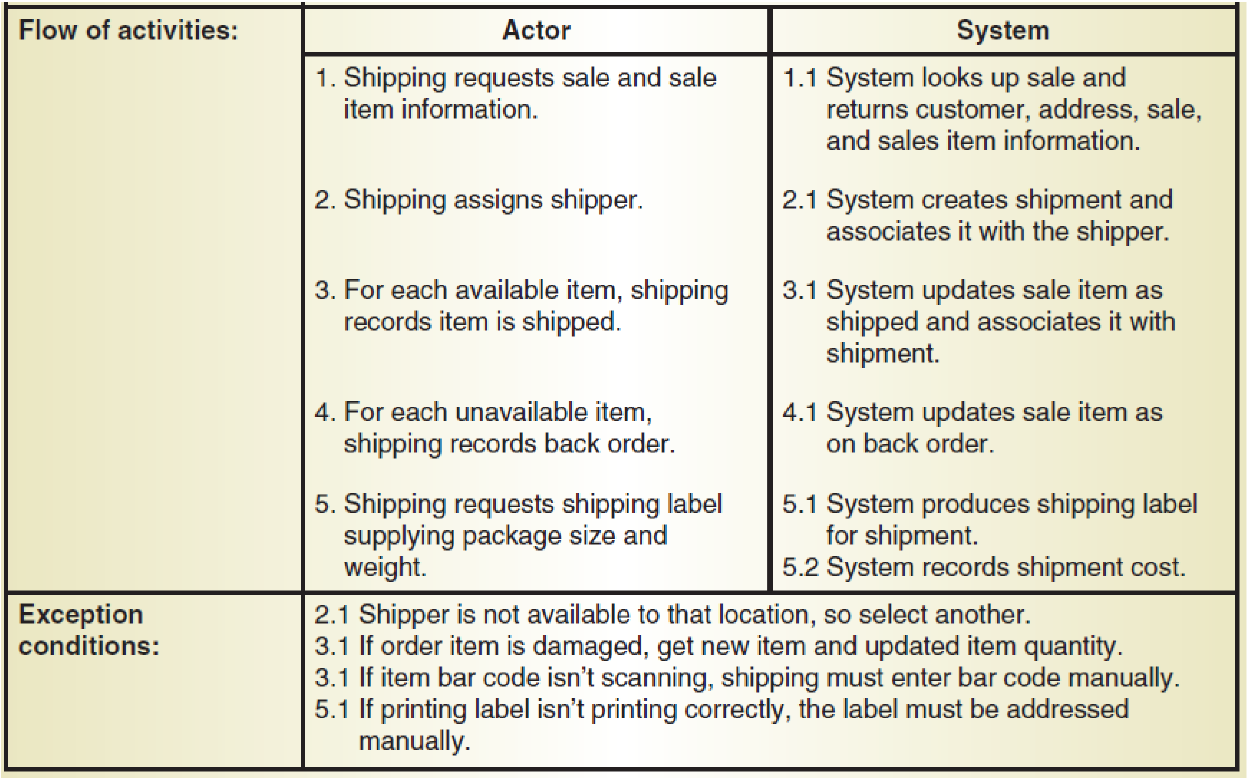
Flow of activities:	Actor	System
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

Fully Developed Use Case Description *Ship items* (part 1)



Use case name:	*Ship items*.
Scenario:	Ship items for a new sale.
Triggering event:	Shipping is notified of a new sale to be shipped.
Brief description:	Shipping retrieves sale details, finds each item and records it is shipped, records which items are not available, and sends shipment.
Actors:	Shipping clerk.
Related use cases	None.
Stakeholders:	Sales, Marketing, Shipping, warehouse manager.
Preconditions:	Customer and address must exist. Sale must exist. Sale items must exist.
Postconditions:	Shipment is created and associated with shipper. Shipped sale items are updated as shipped and associated with the shipment. Unshipped items are marked as on back order. Shipping label is verified and produced.

Fully Developed Use Case Description *Ship items* (part 2)



Flow of activities:	Actor	System
	1. Shipping requests sale and sale item information.	1.1 System looks up sale and returns customer, address, sale, and sales item information.
	2. Shipping assigns shipper.	2.1 System creates shipment and associates it with the shipper.
	3. For each available item, shipping records item is shipped.	3.1 System updates sale item as shipped and associates it with shipment.
	4. For each unavailable item, shipping records back order.	4.1 System updates sale item as on back order.
	5. Shipping requests shipping label supplying package size and weight.	5.1 System produces shipping label for shipment. 5.2 System records shipment cost.
Exception conditions:	2.1 Shipper is not available to that location, so select another. 3.1 If order item is damaged, get new item and updated item quantity. 3.1 If item bar code isn't scanning, shipping must enter bar code manually. 5.1 If printing label isn't printing correctly, the label must be addressed manually.	