6.8. Example: Process Sale, Fully Dressed Style

Fully dressed use cases show more detail and are structured; they dig deeper.

In iterative and evolutionary UP requirements analysis, 10% of the critical use cases would be written this way during the first requirements workshop. Then design and programming starts on the most architecturally significant use cases or scenarios from that 10% set.

Various format templates are available for detailed use cases. Probably the most widely used and shared format, since the early 1990s, is the template available on the Web at *alistair.cockburn.us*, created by Alistair Cockburn, the author of the most popular book and approach to use-case modeling. The following example illustrates this style.

   Main Success Scenario and Extensions are the two major sections

First, here's the template:

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, *and* worth telling the reader?
Success Guarantee	What must be true on successful completion, *and* worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

Here's an example, based on the template.

Please note that this is the book's primary case study example of a detailed use case; it shows many common elements and issues.

It probably shows much more than you ever wanted to know about a POS system! But, it's for a real POS, and shows the ability of use cases to capture complex real-world requirements, and deeply branching scenarios.

Use Case UC1: Process Sale

Scope: NextGen POS application

Level: user goal

Primary Actor: Cashier

Stakeholders and Interests:

   - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.

   - Salesperson: Wants sales commissions updated.

   - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.

   - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.

   - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.

   - Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.

   - Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

Preconditions: Cashier is identified and authenticated.

Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Main Success Scenario (or Basic Flow):

   1. Customer arrives at POS checkout with goods and/or services to purchase.

   2. Cashier starts a new sale.

   3. Cashier enters item identifier.

   4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.

6. Cashier tells Customer the total, and asks for payment.

7. Customer pays and System handles payment.

8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).

9. System presents receipt.

10. Customer leaves with receipt and goods (if any).

Extensions (or Alternative Flows):

*a. At any time, Manager requests an override operation:

1. System enters Manager-authorized mode.

2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.

3. System reverts to Cashier-authorized mode.

*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.

2. System reconstructs prior state.

  2a. System detects anomalies preventing recovery:

  1. System signals error to the Cashier, records the error, and enters a clean state.

  2. Cashier starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.

2. System displays the state of the resumed sale, with subtotal.

  2a. Sale not found.

1. System signals error to the Cashier.

2. Cashier probably starts new sale and re-enters all items.

3. Cashier continues with sale (probably entering more items or handling payment).

2-4a. Customer tells Cashier they have a tax-exempt status (e.g., seniors, native peoples)

1. Cashier verifies, and then enters tax-exempt status code.

2. System records status (which it will use during tax calculations)

3a. Invalid item ID (not found in system):

  1. System signals error and rejects entry.

  2. Cashier responds to the error:

    2a. There is a human-readable item ID (e.g., a numeric UPC):

    1. Cashier manually enters the item ID.

    2. System displays description and price.

      2a. Invalid item ID: System signals error. Cashier tries alternate method.

    2b. There is no item ID, but there is a price on the tag:

    1. Cashier asks Manager to perform an override operation.

    2. Managers performs override.

    3. Cashier indicates manual price entry, enters price, and requests standard taxation for this amount (because there is no product information, the tax engine can't otherwise deduce how to tax it)

    2c. Cashier performs Find Product Help to obtain true item ID and price.

    2d. Otherwise, Cashier asks an employee for the true item ID or price, and does either manual ID or manual price entry (see above).

3b. There are multiple of same item category and tracking unique item identity not important (e.g., 5 packages of veggie-burgers):

    1. Cashier can enter item category identifier and the quantity.

3c. Item requires manual category and price entry (such as flowers or cards with a price on them):

    1. Cashier enters special manual category code, plus the price.

3-6a: Customer asks Cashier to remove (i.e., void) an item from the purchase:

This is only legal if the item value is less than the void limit for Cashiers, otherwise a Manager override is needed.

    1. Cashier enters item identifier for removal from sale.

    2. System removes item and displays updated running total.

        2a. Item price exceeds void limit for Cashiers:

            1. System signals error, and suggests Manager override.

            2. Cashier requests Manager override, gets it, and repeats operation.

3-6b. Customer tells Cashier to cancel sale:

    1. Cashier cancels sale on System.

3-6c. Cashier suspends the sale:

    1. System records sale so that it is available for retrieval on any POS register.

    2. System presents a "suspend receipt" that includes the line items, and a sale ID used to retrieve and resume the sale.

4a. The system supplied item price is not wanted (e.g., Customer complained about something and is offered a lower price):

    1. Cashier requests approval from Manager.

    2. Manager performs override operation.

    3. Cashier enters manual override price.

    4. System presents new price.

5a. System detects failure to communicate with external tax calculation system service:

    1. System restarts the service on the POS node, and continues.

        1a. System detects that the service does not restart.

    1. System signals error.

    2. Cashier may manually calculate and enter the tax, or cancel the sale.

5b. Customer says they are eligible for a discount (e.g., employee, preferred customer):

    1. Cashier signals discount request.

    2. Cashier enters Customer identification.

    3. System presents discount total, based on discount rules.

5c. Customer says they have credit in their account, to apply to the sale:

    1. Cashier signals credit request.

    2. Cashier enters Customer identification.

    3. Systems applies credit up to price=0, and reduces remaining credit.

6a. Customer says they intended to pay by cash but don't have enough cash:

    1. Cashier asks for alternate payment method.

        1a. Customer tells Cashier to cancel sale. Cashier cancels sale on System.

7a. Paying by cash:

    1. Cashier enters the cash amount tendered.

    2. System presents the balance due, and releases the cash drawer.

    3. Cashier deposits cash tendered and returns balance in cash to Customer.

    4. System records the cash payment.

7b. Paying by credit:

    1. Customer enters their credit account information.

    2. System displays their payment for verification.

    3. Cashier confirms.

        3a. Cashier cancels payment step:

            1. System reverts to "item entry" mode.

4. System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.

> 4a. System detects failure to collaborate with external system:
>
> > 1. System signals error to Cashier.
> >
> > 2. Cashier asks Customer for alternate payment.

5. System receives payment approval, signals approval to Cashier, and releases cash drawer (to insert signed credit payment receipt).

> 5a. System receives payment denial:
>
> > 1. System signals denial to Cashier.
> >
> > 2. Cashier asks Customer for alternate payment.
>
> 5b. Timeout waiting for response.
>
> > 1. System signals timeout to Cashier.
> >
> > 2. Cashier may try again, or ask Customer for alternate payment.

6. System records the credit payment, which includes the payment approval.

7. System presents credit payment signature input mechanism.

8. Cashier asks Customer for a credit payment signature. Customer enters signature.

9. If signature on paper receipt, Cashier places receipt in cash drawer and closes it.

7c. Paying by check...

7d. Paying by debit...

7e. Cashier cancels payment step:

> 1. System reverts to "item entry" mode.

7f. Customer presents coupons:

> 1. Before handling payment, Cashier records each coupon and System reduces price as appropriate. System records the used coupons for accounting reasons.
>
> > 1a. Coupon entered is not for any purchased item:

> > > 1. System signals error to Cashier.

9a. There are product rebates:

> 1. System presents the rebate forms and rebate receipts for each item with a rebate.

9b. Customer requests gift receipt (no prices visible):

> 1. Cashier requests gift receipt and System presents it.

9c. Printer out of paper.

> 1. If System can detect the fault, will signal the problem.
>
> 2. Cashier replaces paper.
>
> 3. Cashier requests another receipt.

Special Requirements:

> - Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
>
> - Credit authorization response within 30 seconds 90% of the time.
>
> - Somehow, we want robust recovery when access to remote services such the inventory system is failing.
>
> - Language internationalization on the text displayed.
>
> - Pluggable business rules to be insertable at steps 3 and 7.
>
> - ...

Technology and Data Variations List:

> *a. Manager override entered by swiping an override card through a card reader, or entering an authorization code via the keyboard.
>
> 3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
>
> 3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
>
> 7a. Credit account information entered by card reader or keyboard.
>
> 7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have
to do it?

This use case is illustrative rather than exhaustive (although it is based on a real POS system's
requirementsdeveloped with an OO design in Java). Nevertheless, there is enough detail and
complexity here to offer a realistic sense that a fully dressed use case can record many
requirement details. This example will serve well as a model for many use case problems.