Use Cases
Introduction

- Types:
  - business use cases
  - system use cases (our focus)

- Use cases tell a story of actors using a system.

- They illustrate functional requirements, by the stories they tell.

- Complementary with a function requirement list.
Identifying Use Cases

- Major distinct, complete, end-to-end processes of using a system.

- Not usually one step, but a complete story.
  - Common “mistake”!

- Examples
  - Rent Videos
  - Return Videos
  - Pay Fines
Use Case Diagram

- A way to conceive and illustrate the use cases.
- Usually created during the initial use case analysis.
# A Sample Detailed Use Case

## Use Case: Rent Items

### Typical Course of Events

<table>
<thead>
<tr>
<th>Actor Intentions</th>
<th>System Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Customer arrives at a checkout with videos (and/or less often, video games) to rent.</td>
<td>3. Presents membership information, and status of loans (usually nothing on loan, and no outstanding fines).</td>
</tr>
<tr>
<td>2. The Customer presents their membership identification to the Clerk, who enters it into the system.</td>
<td>4. For each video or game, the Clerk records the item identification into the system.</td>
</tr>
<tr>
<td>4. For each video or game, the Clerk records the item identification into the system.</td>
<td>5. Presents accumulating list of rental item titles, due dates, total rental fee, and any late charges.</td>
</tr>
<tr>
<td>6. Clerk informs Customer of total charge, and asks for payment.</td>
<td>9. If a credit payment, authorizes it.</td>
</tr>
<tr>
<td>7. Customer pays Clerk by cash or credit.</td>
<td>10. Generates receipt and loan report.</td>
</tr>
<tr>
<td>8. Clerk records payment into system.</td>
<td>11. Clerk gives receipt and loan report to Customer, who then leaves with the rental items.</td>
</tr>
</tbody>
</table>

### Alternative Courses

- Step 7. Customer has insufficient cash. Request a credit payment, cancel the transaction, or deduct rental items until transaction can be paid for.

- Step 7: Customer has unpaid late charges and will not pay them. Customer must pay them before renting more items, so either collect full payment, or cancel the transaction.

- Step 9. Failure to authorize credit payment, either because of insufficient credit or inactive authorization service. Request cash payment instead.
A Sample Summary Use Case

- Same principles a detailed use case, but simplifies steps and details, as a low-fidelity incomplete first draft.
  - Useful during early requirements and scope analysis

<table>
<thead>
<tr>
<th>Actor Intentions</th>
<th>System Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Customer presents items to rent.</td>
<td>3. Remember rented items.</td>
</tr>
<tr>
<td>2. Clerk records items.</td>
<td>4. Calculate and present price.</td>
</tr>
<tr>
<td>5. Customer pays.</td>
<td>6. Authorize and record payment.</td>
</tr>
</tbody>
</table>
Use Case Forms

- Short, “high-level”, summary form.
  - Write most of them in this form during Inception.
  - Write a few keys ones in the detailed form.

  - Write them this way during Elaboration.
Use Miscellany

- Simple “CRUD” use cases can be combined into one “Maintain <X>”.
  - Maintain Inventory

- Start name with a verb.

- Start with sentence 1 with “<Actor> does <event>”

- All systems have a Start Up and Shut Down use case (perhaps trivial).
Essential and Concrete Use Cases

- *Essential* use cases defer the details of the UI, and focus on the *intentions* of the actors, and responsibilities of the system.
  - Concrete (AKA Real) do not.

Essential: “The AccountHolder identifies themselves to the ATM”

Real: “The AccountHolder inserts their card in the reader. Window A is displayed. They enter their PIN on the numeric keypad, …”

As we move from analysis to design, we are more inclined to move from essential to concrete use case descriptions.
Relating Use Cases

When creating the use case diagram, it can be useful (in terms of comprehension and simplification) to:

- factor out shared sub-processes
  
  use the <<includes>> relationship

- show precedence order
  
  use the <<extends>> relationship
Video Store
Information System

Rent Items

Pay Fines

«includes» «includes»

Pay by Cash  Pay by Credit

«includes» «includes»

Rent Items

«extends»

Query For Items