# Evaluating a Use Case Script

## Background

Use Case diagrams express the functions of a system in an abstract, readable way. Use Cases provide the process step detail to explain exactly how a use case succeeds. One way to learn about writing good use cases is to analyse poor ones.

## Tasks:

1. Examine the excerpts from the following use cases
2. Find the errors in the interactions
   * Re-write as necessary
3. Explain reasoning

# Use Case 1:

## Use case: Log In

**Actor/s:** Client, Administrator

## Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Client clicks on log in button | 2. System requests identification for user |
| 3. Client enters account name and password then clicks on log in button | 4. System authenticates details by matching it with the details on the record |
|  | 5. If correct then system signs in client |

## Errors:

* Step 1 and Step 3 refers to GUI
  + Essential use cases should not make any references to GUI or technology
* Step 3 abstraction too low
* Step 4 abstraction too low
  + Too specific, too much detail of processes
* Repeated actions not modelled correctly
  + Should use one of the two repeat styles shown in lecture
* Alternative flow is incorrectly modelled
  + Should either add description of variation in alternative flow section
  + Or modify body of use case description to show alternative flow
* Is missing the following sections
  + Purpose
  + Overview
  + Preconditions
  + Postconditions
* Actor Administrator is not active in use case and should be removed

# Use Case 1 Re-write:

## Use case: Log In

**Actor/s:** Client  
**Purpose:** This use case allows client to log into the system.  
**Overview:** Client logs into system and is authenticated. If authentication is failed after 3 attempts, use case ends.  
**Preconditions:** Client is logged out of system and is attempting to log in.  
**Postconditions:** Client is successfully logs into system.

## Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| Repeat | |
| 1. Client enters user account details | 2. System authenticates details |
| Until successfully logged in, or after 3 failed attempts | |

## Alternative Courses:

None

## Related Use Cases

**Includes:** None  
**Extends:** None

# Use Case 2:

## Use case: Payment

**Actor/s:** Client  
**Overview:** This Use Case begins when the client buys the insure product  
This Use Case allows client to pay or part-pay invoice online.  
And also facilitate client to pay through different methods via cheque, cash or through credit card.

## Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Client pay invoice online. | 2. System shows appointment details and balance |
| 3. Client checks the payment amount and chooses to pay or part-pay the invoice and enters the account number, name, security number and other information | 4. System authenticates payment information |
|  | 5. If correct then system records paid or part-paid |

## Alternative Courses

**Step 2:** If payment information is incorrect then back to step 3 and display “payment unsuccessful” and use case ends.

## Errors:

* Is missing the following sections
  + Purpose
  + Preconditions
  + Postconditions
* Overview has spelling error “insure”
* Overview includes details that should have been included in Purpose and Precondition
* Overview mentions that the use case allows client to pay via different payment methods which are not present in the action steps
* Step 1 abstraction too high
* Step 2 mentions appointment details, not sure how this is related to payment
* Step 3 abstraction too low
* Requires «extend» at Step 3
* Alternative course refers to incorrect step, should refer to Step 4

# Use Case 2 Re-write:

## Use case: Payment

**Actor/s:** Client  
**Purpose:** This Use Case allows client to pay or part-pay an invoice online via selected payment method.  
**Overview:** Client selects to either pay or part-pay the invoice, and decides on payment method. If payment is made via credit card, system authenticates payment information. If payment is authenticated, the payment is recorded.  
**Preconditions:** The Use Case begins when the client purchases an insurance product.  
**Postconditions:** System provides client with payment methods. If payment is made via Credit Card, a successful payment is recorded by the system.

## Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Client purchases insurance product | 2. System displays balance and option for full or part payment |
| 3. Client selects part payment  *Extension Point:* Make full payment | 4. System displays payment method options |
| 5. Client selects payment via Credit Card and enters details  *Extension Point:* Pay via cheque  *Extension Point* Pay via cash | 6. System authenticates payment information |
|  | 7. System records payment and displays new balance |

## Alternative Courses:

**Step 3:** If client selects full payment, continue with Step 4 **Step 5:** If client selects Cheque payment method, system displays balance owing, business mailing address and business physical address, use case ends  
**Step 5:** If client selects Cash payment method, system displays balance owing and business physical address, use case ends  
**Step 7:** If payment details fail authentication, system displays “payment unsuccessful” and go to Step 5, use case ends

## Related Use Cases

**Includes:** None  
**Extends:** Make full payment, Pay via cheque, Pay via cash

# Use Case 3:

## Use case: Creating Invoice

**Actor/s:** Administrator  
**Purpose:** Creating invoice  
**Preconditions:** N/A  
**Postconditions:** Payment is to be made by client from one of the given method of online payment

## Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
|  | 1. Create the payment invoice |
|  | 2. Send to the client |

## Alternative courses

N/A

## Errors:

* Missing following sections
  + Overview
* Postcondition does not match with typical course of events
* Actor administrator does not have any actions
* No actor actions to begin use case
* No actor actions exist at all
* Step 1 abstraction too high, what is the system creating the invoice for
* Step 2 abstraction too high, how is invoice sent to customer

# Use Case 3 Re-write:

## Use case: Creating Invoice

**Actor/s:** Client  
**Purpose:** This Use Case generates an invoice when a customer orders *product.* **Overview:** Client orders *product* and system generates invoice which is then emailed to the customer for payment.   
**Preconditions:** Client has ordered *product.*   
**Postconditions:** Invoice is sent to the customer.

## Typical Course of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Client orders product | 2. System displays balance owing and asks client to confirm purchase |
| 3. Client confirms order | 4. System generates and emails invoice to client |

## Alternative courses

**Step 4:** Client declines confirmation, system displays message “Order cancelled” and go to Step 1, use case ends   
**Step 4:** Client does not respond to confirmation, after 5 minutes system displays message “Order timeout” and go to Step 1, use case ends

## Related Use Cases

**Includes:** None  
**Extends:** None