**USE CASE DIAGRAMS**

****

**USE CASE**

* A pattern of behaviour that the new system is required to exhibit
* A sequence of related actions performed by an actor and the system via a dialogue
* A use case is a typical interaction between a user and a computer system.
* captures some user-visible function
* may be small or large
* achieves a discrete goal for a user
* capture each discrete thing a user wants to do with the system - give it a name and write a short textual description - it will be further developed during the iteration in which it is built.
* show external interactions when other system initiates the use case, or show system if it is the one that needs the use case.
* Capture the relationship between actors and use cases
  + E.g. Campaign Manager [Add new client]
  + E.g. Accountant (Update Client Payment)
* Components
  + System boundary
  + Actor
  + Communication Association
  + Use Case

**ACTOR**

Actor - a role a user plays with regard to the system

* many users play same role
* some users play many roles
* use case may be used by many actors (ie required when using system in more than one role)

An actor can be -

* A person
* Other subsystems
* A role that different people play
* Another external system
* Hardware devices (not used in early use case diagram)

**IDENTIFYING ACTORS**

Look for

– the users who directly use the system

– also others who need services from the system

• To find actors that are people/roles ask:

1. Who will be a primary user of the system? (primary actor)
2. Who will need support from the system to do her daily tasks?
3. Who will maintain, administrate, keep the system working?

(secondary actor)

1. Who or what has an interest in the results that the system

* produces ?
* • To find actors that are external systems ask:

1. – Which hardware devices does the system need?
2. – With which other systems does the system need to interact with?

**FINDING USE CASES**

For each actor, ask the following questions:

1. Which functions does the actor require from the system?
2. What does the actor need to do ?
3. Does the actor need to read, create, destroy, modify, or store

some kind of information in the system ?

1. Does the actor have to be notified about events in the system?
2. Does the actor need to notify the system about something?
3. What do those events require in terms of system functionality?
4. Could the actor’s daily work be simplified or made more efficient

through new functions provided by the system?

**WHAT IS A GOOD USE CASE?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Good example** | **COMMENT** | **Bad Example** | **COMMENT** |
| Create Reservation | **We would not include updates and cancellations in this unless the use case name is changed to Manage Reservations** | Change Hotel Room | **Use cases that deal with a single operation are likely to be too simple** |
| Customer ChecksIn/ChecksOut | **Use cases should incorporate variations of a single theme**  **Common mistake is to break up these variations into multiple use cases** |  |  |

**WHAT IS A GOOD USE CASE DESCRIPTIONS**

* The text in a use case description is intended to show actor/system interaction, not all details of the policies and validations used to implement the use cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Good example** | **COMMENT** | **Bad Example** | **COMMENT** |
| Step 2. The customer provides a method of payment to hold the reservation |  | The system computes a checksum for the credit card according to the following formula | **THE USE CASE DESCRIPTION SHOULD BE INDEPENDENT OF ANY TECHNOLOGY** |
| The customer provides a reservation number |  | The customer selects a reservation number from a pull-down list |  |

**USE CASE DESCRIPTION**

* Captures what happens in the use case at a high level
* Is concerned with what, not how
* Is concerned with the ‘actors eye view’
* identifies the actor obtaining value from the use case
* describes the basic course
* describes any alternate courses