

Use Case Description	
<b>System:</b>	<b>Student Group ID:</b>
<b>Use Case name:</b>	<b>UC ID:</b>
<b>Primary actor:</b> ( <i>User/System initiating UC</i> )	<b>Priority (H,M,L):</b>
<b>Stakeholders:</b> <i>have interest in the UC</i>	
<b>Goal:</b>	
<b>Trigger:</b> ( <i>e.g. user calls, Inventory low</i> ) – ( <i>if time driven indicate temporal condition e.g. end-of-month</i> )	
<b>Relationships</b> <ul style="list-style-type: none"> <li>▪ <b>Includes:</b></li> <li>▪ <b>Extends:</b></li> <li>▪ <b>Generalization:</b></li> <li>▪ <b>Extension points:</b> <i>at which extension UCs may extend this UC</i></li> </ul>	
<b>Input:</b>	
<b>Pre-conditions:</b> <i>What validity checks or constraints apply on the inputs (or the internal system as a whole) before the UC begins).</i> <ul style="list-style-type: none"> <li>▪</li> </ul>	
<b>Normal (Basic) flow of events – Happy path –Successful path</b>	
<b>Steps:</b>	
<b>Actor</b>	<b>System</b>
1. Actor does ....  3. 4.	2. Sys does .... (related artifacts #, if any, e.g. decision table, decision tree, condition/response table, algorithm, ..)
<b>Alternate flows:</b> ( <i>Variations with successful UC</i> )	
<b>Exceptional flows:</b> ( <i>UC failure leading to “post condition on failure”</i> )	
4.a <i>Exceptional flow name : description ....</i>	
<b>Post-conditions on success:</b> <i>changes the UC makes to the internal system state.</i> <ul style="list-style-type: none"> <li>▪</li> </ul>	
<b>Post-conditions on failure:</b>	
<b>Output:</b>	
<b>Test Cases:</b>	
<ul style="list-style-type: none"> <li>▪ <b>Unit testing:</b></li> <li>▪ <b>Functional testing:</b></li> </ul>	
UCP (Use Case Points) “ <i>effort e.g. in man hour</i> ”	

<b>Example: Use Case Description</b>	
<b>System:</b> Hotel Reservation	<b>Group ID:</b> group A
<b>Use Case name:</b> Make a reservation	<b>UC ID:</b> 4
<b>Primary actor:</b> Client	<b>Priority (H,M,L):</b> H
<b>Stakeholders:</b>	<b>UCP (Use Case Points):</b> 15 man-hour
<b>Goal:</b> Reserve a room at a hotel	
<b>Trigger:</b> Client accesses online reservation	
<b>Relationships</b> <ul style="list-style-type: none"> <li>▪ Association:</li> <li>▪ Includes:</li> <li>▪ Extends:</li> <li>▪ Generalization:</li> <li>▪ Extension points:</li> </ul>	
<b>Input:</b> hotel, arrival and departure dates, room type, name and post code, email address	
<ul style="list-style-type: none"> <li>▪ <b>Pre-conditions:</b> Client is logged and has access to hotels site</li> </ul>	
<b>Normal (Basic) flow</b> of events – <b>Happy</b> path – <b>Successful</b> path – <b>Main Success</b> Scenario Steps:	
<b>Actor</b>	<b>System</b>
1. Client enters a gateway for hotels. 2. Client <b>selects</b> hotel, arrival and departure dates, and room type. <b>Artifact 4.1</b> 3. Client accepts and asks for a room. 4. Client provides name and post code. 5. Client provides his email address.	3. System provides availability and price. <b>Artifact 4.2, FT4.1</b>  7. System makes a reservation and allocates a reservation number. 8. System shows reservation number to client. <b>FT4.3</b> 9. System creates and sends a confirmation to client by email. <b>FT4.4</b>
<b>Alternate and Exceptional flows:</b> 3.1 Required room type not available <ul style="list-style-type: none"> <li>a. System offers alternative rooms <ul style="list-style-type: none"> <li>a1. Client accepts and selects from alternative rooms</li> <li>a2. Resume 4</li> </ul> </li> <li>b1. Client refuses alternatives</li> <li>b2. <b>Fail (FT4.6)</b></li> </ul> 4.1 Client declines offer <ul style="list-style-type: none"> <li>a. <b>Fail</b></li> </ul> 6.1 Client is registered and his info already on file <ul style="list-style-type: none"> <li>a. system offers a discount <b>Artifact 4.3, FT4.2</b></li> </ul> Resume 7	
<b>Post-conditions on success:</b> Database updated with client reservation and client info <b>FT4.5</b>	
<b>Post-conditions on failure:</b> Record failure reason and date in Database <b>FT4.6</b>	
<b>Output:</b> reservation number, confirmation to client by email. ( <b>FT4.4</b> )	
<b>Test Cases:</b> <ul style="list-style-type: none"> <li>▪ Unit testing: UT4.1, UT4.2</li> <li>▪ Functional testing: FT4.1, FT4.2, FT4.3, FT4.4, FT4.5, FT4.6</li> </ul>	

### Artifact 1 (list)

Room type: single, double, suite, sea-side single , sea-side double

### Artifact 2 (algorithm)

Normal rates are: single 300, double 400, suite 600

If high session (month 2, 8) rates are: single 400, double 600, suite 800

### Artifact 3 (algorithm)

If registered client, if up to than 5 times discount 10% - if more than 5 discount 20%

### Functional testing

FT4.1: Check available rooms and prices according to artifact 2

FT4.2: Check discount according to artifact 3

FT4.3: Check reservation # sequence

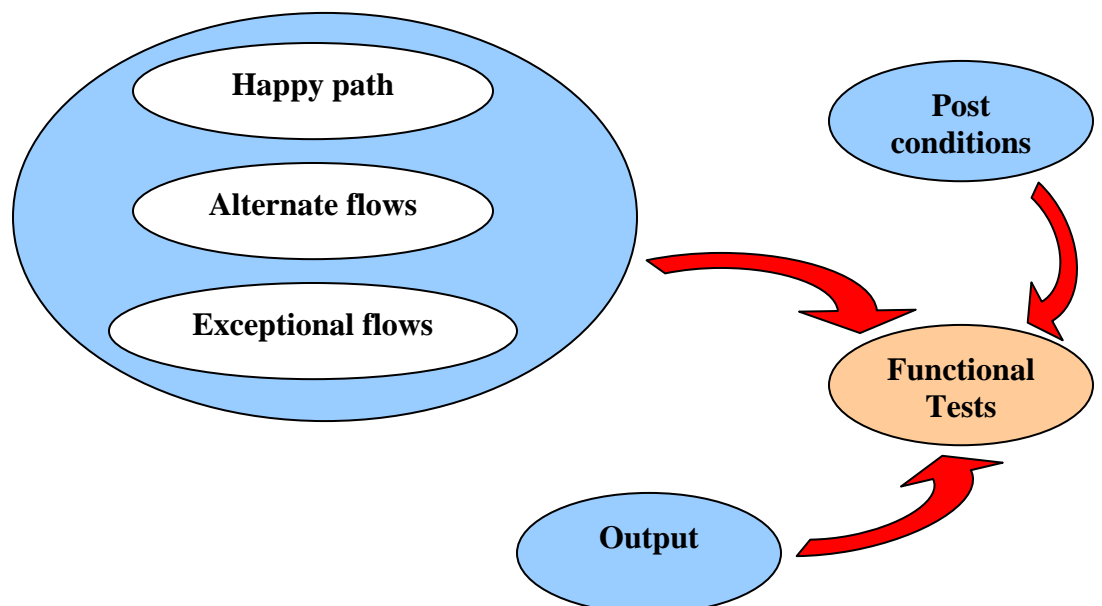
FT4.4: Check client receives email with reservation #

FT4.5: Check post-condition on success

FT4.6: Check post-condition on failure

### How to write UC

1. Requirements elicitation (Facts finding)
2. Fill template:
  - a. ...
  - b. Preconditions
  - c. Paths (Happy path, alternate and Exceptional flows)
    - i. Identify functional tests
    - ii. Identify artifacts
  - d. Post-conditions: on success & on failure
    - i. Identify functional tests
  - e. Output: Identify functional tests



## Decision tables

### Example of use case with **decision table**:

#### System Use Case: Process Life Insurance Application

##### Basic flow:

1. User enters application information.
2. System validates eligibility- (**Artifact 1**)
3. System adds application to “Adjuster” queue.

##### Alternate flows

- 3a Referred application:
- .1 System adds application to referral queue.
  - .2 The use case ends.

##### Exception flows

- 3b Rejected application:
- .1 System adds application to rejection queue.
  - .2 The use case ends in failure.