## Use Case Description

<table>
<thead>
<tr>
<th>System:</th>
<th>Student Group ID:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case name:</td>
<td>UC ID:</td>
</tr>
<tr>
<td><strong>Primary actor:</strong> <em>(User/System initiating UC)</em></td>
<td><strong>Priority (H,M,L):</strong></td>
</tr>
<tr>
<td><strong>Stakeholders:</strong> <em>have interest in the UC</em></td>
<td></td>
</tr>
</tbody>
</table>

### Goal:

### Trigger:
*(e.g. user calls, Inventory low) – (if time driven indicate temporal condition e.g. end-of-month)*

### Relationships
- Includes:
- Extends:
- Generalization:
- **Extension points:** *at which extension UCs may extend this UC*

### Input:

**Pre-conditions:** *What validity checks or constraints apply on the inputs (or the internal system as a whole) before the UC begins.>*

### Normal (Basic) flow of events – Happy path – Successful path

#### Steps:

<table>
<thead>
<tr>
<th>Actor</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Actor does ….</td>
<td>2. Sys does …. <em>(related artifacts #, if any, e.g. decision table, decision tree, condition/response table, algorithm, ..)</em></td>
</tr>
<tr>
<td>3.</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
</tr>
</tbody>
</table>

### Alternate flows: *(Variations with successful UC)*

### Exceptional flows: *(UC failure leading to “post condition on failure”)*

#### 4.a Exceptional flow name: description ....

### Post-conditions on success:
*changes the UC makes to the internal system state.*

### Post-conditions on failure:

### Output:

### Test Cases:
- **Unit testing:**
- **Functional testing:**

### UCP (Use Case Points) "effort e.g. in man hour"
### Example: Use Case Description

<table>
<thead>
<tr>
<th><strong>System:</strong> Hotel Reservation</th>
<th><strong>Group ID:</strong> group A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case name:</strong> Make a reservation</td>
<td><strong>UC ID:</strong> 4</td>
</tr>
<tr>
<td><strong>Primary actor:</strong> Client</td>
<td><strong>Priority (H,M,L):</strong> H</td>
</tr>
<tr>
<td><strong>Stakeholders:</strong></td>
<td><strong>UCP (Use Case Points):</strong> 15 man-hour</td>
</tr>
<tr>
<td><strong>Goal:</strong> Reserve a room at a hotel</td>
<td><strong>Trigger:</strong> Client accesses online reservation</td>
</tr>
</tbody>
</table>

#### Relationships
- Association:
- Includes:
- Extends:
- Generalization:
- Extension points:

#### Input:
- hotel, arrival and departure dates, room type, name and post code, email address

- **Pre-conditions:** Client is logged and has access to hotels site

#### Normal (Basic) flow of events – **Happy** path – **Successful** path – **Main Success** Scenario

**Steps:**

<table>
<thead>
<tr>
<th>Actor</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Client enters a gateway for hotels.</td>
<td>3. System provides availability and price. Artifact 4.1, <strong>FT4.1</strong></td>
</tr>
<tr>
<td>2. Client <strong>selects</strong> hotel, arrival and departure dates, and room type. <strong>Artifact 4.1</strong></td>
<td></td>
</tr>
<tr>
<td>4. Client accepts and asks for a room. Art</td>
<td>7. System makes a reservation and allocates a reservation number.</td>
</tr>
<tr>
<td>5. Client provides name and post code.</td>
<td>8. System shows reservation number to client. <strong>FT4.3</strong></td>
</tr>
<tr>
<td>6. Client provides his email address.</td>
<td>9. System creates and sends a confirmation to client by email. <strong>FT4.4</strong></td>
</tr>
</tbody>
</table>

#### Alternate and Exceptional flows:

- 3.1 Required room type not available
  - a. System offers alternative rooms
    - a1. Client accepts and selects from alternative rooms
    - a2. Resume 4
  - b1. Client refuses alternatives
    - b2. **Fail (FT4.6)**

- 4.1 Client declines offer
  - a. **Fail**

- 6.1 Client is registered and his info already on file
  - a. system offers a discount **Artifact 4.3, FT4.2**
  - Resume 7

#### Post-conditions on success:
- Database updated with client reservation and client info **FT4.5**

#### Post-conditions on failure:
- Record failure reason and date in Database **FT4.6**

#### Output:
- reservation number, confirmation to client by email. (**FT4.4**)

#### Test Cases:
- Unit testing: UT4.1, UT4.2
- Functional testing: FT4.1, FT4.2, FT4.3, FT4.4, FT4.5, FT4.6
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.