

## Sweet Dreams Software Analysis 2015 - 2016

<b><u>Docent(s):</u></b>	Luc Doumen, Nathalie Fuchs		
<b><u>Assignment:</u></b>	Create a domain model/ diagram		
<b><u>Due date</u></b>	Week 7 – 10/11/2015		
<b><u>Type of assignment:</u></b>	Group Assignment (4)		
<b><u>Student(s)</u></b>	Cedric Degeling Luc Vermeulen Raf Eerkens Filip Rutgeerts Samy Coenen	<b><u>Score:</u></b>	... / ...
<b><u>To save as</u></b>	<ul style="list-style-type: none"> <li>➤ PDF file</li> <li>➤ Printed version</li> </ul>		

## 1. Model solution: domain model

### 1.1. Create the business use case description

<b><u>ID version</u></b>	V1
<b><u>Summary</u></b>	Sweet Dreams: Boeken overnachting.
<b><u>Actor(s)</u></b>	Receptie, gasten, betaalsysteem, bank
<b><u>Precondition(s)</u></b>	a) Gast reserveert.
<b><u>Main scenario</u></b>	<ol style="list-style-type: none"> <li>1) Receptie ontvangt gast (inchecken).</li> <li>2) Receptie checkt reservering.</li> <li>3) Rekening wordt door bank nagekeken.</li> <li>4) Receptie keurt reservering goed.</li> <li>5) Receptie controleert vrije kamers.</li> <li>6) Receptie wijst gast een kamer toe.</li> <li>7) Extra kosten worden bijgehouden bij verblijf.</li> <li>8) Gast checkt uit bij receptie.</li> <li>9) Receptie print rekening.</li> <li>10) Bank controleert kaartgegevens klant.</li> <li>11) Klant betaalt rekening (via bank)</li> <li>12) Klant checkt uit.</li> </ol>
<b><u>Alternative flow(s)</u></b>	<ol style="list-style-type: none"> <li>1) Reservering is niet ok.</li> <li>2) Gast maakt nieuwe reservering bij receptie.</li> </ol>
<b><u>Post condition(s)</u></b>	Gast heeft probleemloos verblijf gehad.

## 1.2. Underline the nouns

<b><u>ID version</u></b>	V1
<b><u>Summary</u></b>	Sweet Dreams: Boeken overnachting
<b><u>Actor(s)</u></b>	Receptie, gasten, betaalsysteem, bank
<b><u>Precondition(s)</u></b>	a) Gast reserveert.
<b><u>Main scenario</u></b>	1) <u>Receptie</u> ontvangt <u>gast</u> (inchecken). 2) <u>Receptie</u> checkt <u>reservering</u> . 3) <u>Rekening</u> wordt door <u>bank</u> nagekeken. 4) <u>Receptie</u> keurt <u>reservering</u> goed. 5) <u>Receptie</u> controleert vrije <u>kamers</u> . 6) <u>Receptie</u> wijst <u>gast</u> een <u>kamer</u> toe. 7) <u>Extra kosten</u> worden bijgehouden bij <u>verblijf</u> . 8) <u>Gast</u> checkt uit bij <u>receptie</u> . 9) <u>Receptie</u> print <u>rekening</u> . 10) <u>Bank</u> controleert <u>kaartgegevens klant</u> . 11) <u>Klant</u> betaalt <u>rekening</u> (via <u>bank</u> ) 12) <u>Klant</u> checkt uit.
<b><u>Alternative Flow(s)</u></b>	1) Reservering is niet ok. 2) Gast maakt nieuwe reservering bij receptie
<b><u>Post Conditions</u></b>	Gast heeft probleemloos verblijf gehad.

## 1.3. Put in alphabetic order the nouns and count their frequency

<b><i>Concept</i></b>	<b><i>Frequency</i></b>
Bank	3
Gast/klant	5
Kaartgegevens	1
Kamers	2
Receptie	6
Rekening	2
Reservering	2
Verblijf	1

## 1.4. Determine the domain concepts and their description

<i>Concept</i>	<i>Freq.</i>	<i>Comments</i>
Bank	3	De bank waarvan het hotel gebruik maakt.
Gast/klant	5	De persoon die een overnachting wenst te boeken in het hotel.
Kaartgegevens	1	De gegevens van de debet/kredietkaart die de klant gebruikt
Kamers	2	De specifieke hotelkamer waar deze gast zijn overnachting zal doorbrengen in het hotel.
Receptie	6	De plaats waar de gasten worden onthaald als ze het hotel binnenstappen.
Rekening	2	De facturatie van de overnachting en eventuele extra kosten staan hier op vermeld.
Reservering	2	Het op voorhand vastleggen van een overnachting in het hotel om zeker te zijn van beschikbaarheid.
Verblijf	1	Een verblijf is een tijdelijke woonplaats.

### **1.5. Determine the domain classes**

De domeinklassen zijn:

- Bank
- Gast/klant
- Kaartgegevens
- Kamers
- Receptie
- Rekening
- Reservering
- Verblijf

### 1.6. Draw the domain class diagram

