# Case

Företag: Young travels

Ett företag som inriktar sig på att sälja komplette reselösningar till unga par eller grupper.

Affärsmodellen är att kunden säger vad för typ av semester den vill ha och Young Travels ska då ge förslag på Resemål där transport och boende är inbakat.

De har ett partnerskap med AirBnB vilket är deras vanligaste typ av boende.

Uppdrag till BI-konsulterna:

Young travels affärsmodell är att tjäna royalty på sålda flygsäten och AirBnB-boenden. Därför vill de få fram de städer där priserna på uthyrning trendar neråt. Detta eftersom billiga resor är attraktiva och Young travel kan hävda att med deras kundbas öka efterfrågan på de orter de väljer att ha som resmål och på så vis öka priset -> kan kräva bättre royalty.

Annan uppenbar data de vill få fram är hur nöjda tidigare kunder är på diverse boenden delat på priset (få fram bästa boende per krona)

Kunna kundanpassa letandet:

Vissa vill ha boende nära restauranger, andra vill ha ett säkert boende med många recensioner, någon vill bara ha det billigaste.

Vi ska kunna hitta boenden baserat på olika krav.

Kravet är alltså 2 olika databaser:

En som visar statisk information om vilka städer som är bra att fokusera uthyrning på. Den andra ska vara en dynamisk databas där man hittar bästa boendet baserat på kundens önskemål

# Programmering:

Dokumentation av kod:

Alla metoder man skapar bör dokumenteras.

Med att skriva ”///” ovanför den färdiga koden skapas ett kommentarsfält med alla nödvändiga parametrar som ska finnas med.

Exempel:

Koden “public int Addition(int number){int x=0; x= x+number; return x;}”

Kommer med /// att skapa

/// <summary>

///

/// </summary>

/// <param name="number"></param>

/// <returns></returns>

Kommentaren kan exempelvis då skrivas följande:

/// <summary>

/// <c>Addition</c> returns an int with the same value as “number”

/// </summary>

/// <param name="number">An integer, can be any valid integer</param>

/// <returns>an int with the value ”number”</returns>

XML standarden kan vara överkurs (den används för att automatiskt skapa dokumentation av koden) men det är bra att öva in vanan. Det viktiga är att kunna beskriva vad koden gör, så att den går att använda av andra utan att hen sätter sig in i hur koden fungerar.

Därför är det även viktig i större metoder att beskriva restriktioner på variablerna man skickar med metoden. Exempel: skriver man en divisions kod och tar med nämnaren som parameter är det viktigt att i kommentarsfältet skriva att nämnaren inte får vara noll, för då kraschar programmet.