

## TESTONTWERP & -REALISATIE

|                      |                |
|----------------------|----------------|
| <b>Naam</b>          | Oscar Rijpstra |
| <b>Studentnummer</b> | 20122144       |
| <b>Klas</b>          | 4              |

## OPDRACHT

Geef hieronder een korte beschrijving van je opdracht (een samenvatting van je productvisie) zodat je docent OPT3 ook begrijpt met welke opdracht jij bezig bent.

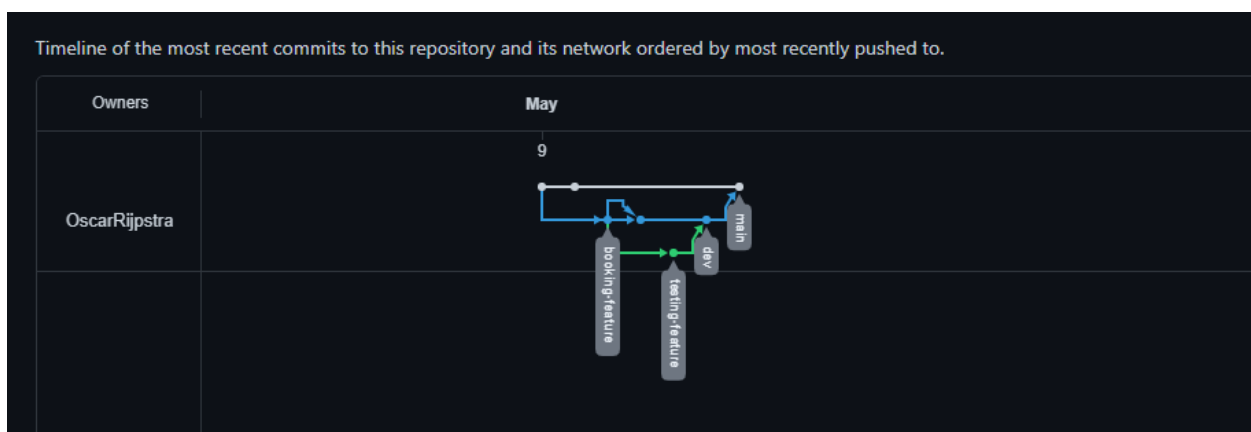
Momenteel is het in het huidige boekingssysteem van het AKZENT City Hotel Kleven niet overzichtelijk hoeveel boekingen er in verschillende periodes zijn geboekt. Door de boekingsdata te visualiseren wordt het gemakkelijk voor hoteleigenaren/managers om inzicht te krijgen in de boekingen. Hierdoor kunnen er bijvoorbeeld sneller beslissingen genomen worden om de bedrijfsvoering te verbeteren. Het is nog onduidelijk hoeveel kosten er bespaard worden aangezien deze oplossing niet direct leidt tot hogere boekingen. Daarom moet er een softwareoplossing gerealiseerd worden waarin bestaande data gevisualiseerd wordt in verschillende grafische overzichten.

## GITHUB EN TDD

Maak je GitHub-repo Private en geef je docent toegang tot je repo:

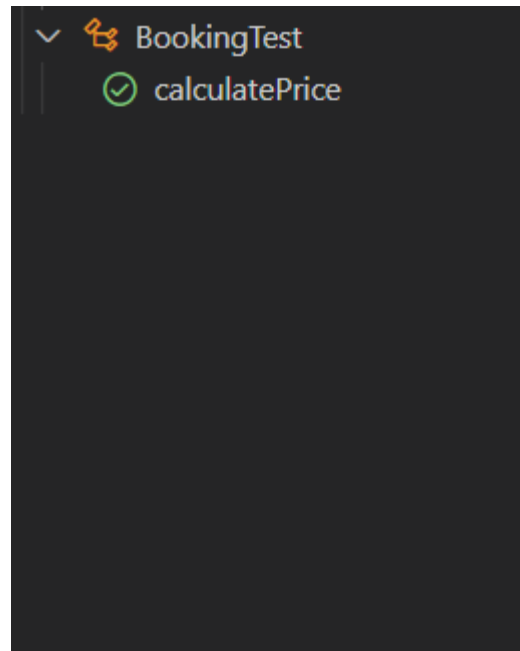
<https://github.com/OscarRijpstra/OPT-3.git>

Om aan te tonen dat je volgens TDD hebt gewerkt voor deze opdracht neem je hieronder een kopie van je netwerk over (zie de toelichting op het template voor instructie):



## GESLAAGDE TESTS

Toon met een relevante screenshot aan dat alle onderstaande tests inderdaad zijn geslaagd:



## SAMENGESTELDE DECISION

In mijn software neem ik een besluit op basis van de volgende voorwaarden (maximaal 3 te testen):

Voorwaarde A: <voorwaarde A>

Voorwaarde B: <voorwaarde B>

Voorwaarde C: <voorwaarde C>

Het besluit is gebaseerd op de volgende combinatie van deze voorwaarden (bijv.  $D = A \ \&\& \ (B \ || \ C)$ ):

D = <Decision>

## CONDITION COVERAGE

Als ik mijn software voor deze Decision zou testen, dan zou ik met Condition Coverage de volgende test cases testen (als er teveel regels in de tabel staan, kun je de waarden vervangen door een spatie):

| Voorwaarde A | Voorwaarde B | Voorwaarde C | D = <Decision> |
|--------------|--------------|--------------|----------------|
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |

## DECISION COVERAGE

Als ik mijn software voor deze Decision zou testen, dan zou ik met Decision Coverage de volgende test cases testen (als er teveel regels in de tabel staan, kun je de waarden vervangen door een spatie):

| Voorwaarde A | Voorwaarde B | Voorwaarde C | D = <Decision> |
|--------------|--------------|--------------|----------------|
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |

## CONDITION/DECISION COVERAGE

Als ik mijn software voor deze Decision zou testen, dan zou ik met Condition/Decision Coverage de volgende test cases testen (als er teveel regels in de tabel staan, kun je de waarden vervangen door een spatie):

| Voorwaarde A | Voorwaarde B | Voorwaarde C | D = <Decision> |
|--------------|--------------|--------------|----------------|
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |

## MODIFIED CONDITION/DECISION COVERAGE

Als ik mijn software voor deze Decision zou testen, dan zou ik met Modified Condition/Decision Coverage de volgende test cases testen (als er teveel regels in de tabel staan, kun je de waarden vervangen door een spatie). Geef met <l>/<f> aan met welke logische en fysieke test cases je werkt:

| Voorwaarde A | Voorwaarde B | Voorwaarde C | D = <Decision> |
|--------------|--------------|--------------|----------------|
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |

## MULTIPLE CONDITION COVERAGE

Als ik mijn software voor deze Decision zou testen, dan zou ik met Multiple Condition Coverage de volgende test cases testen (als er teveel regels in de tabel staan, kun je de waarden vervangen door een spatie):

| Voorwaarde A | Voorwaarde B | Voorwaarde C | D = <Decision> |
|--------------|--------------|--------------|----------------|
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |
| <waarde>     | <waarde>     | <waarde>     | <waarde>       |

## **MODIFIED CONDITION/DECISION COVERAGE (JUNIT)**

Ik heb de testcases voor Modified Condition/Decision Coverage omgezet naar de volgende JUnit-code om mijn software voor deze beslissing te testen.

<De JUnit-code (o.b.v. MC/DC)>

Hiermee test ik de volgende methode (van het type boolean, waarin de waarde van D = <Decision> wordt bepaald):

<De code van de geteste methode>

## EQUIVALENTIEKLASSEN EN RANDWAARDEN

Voor het nemen van een besluit in de software heb ik een domein dat bestaat uit de volgende equivalentieklassen (maximaal 3 te testen):

|                      | Logische weergave      | Besluit     |
|----------------------|------------------------|-------------|
| Equivalentieklasse 1 | <Equivalentieklasse 1> | <Besluit 1> |
| Equivalentieklasse 2 | <Equivalentieklasse 2> | <Besluit 2> |
| Equivalentieklasse 3 | <Equivalentieklasse 3> | <Besluit 3> |

Geef nu alle relevante fysieke testgevallen (incl. de meest intensieve test van randwaarden) en geef voor elk testgeval aan voor welke equivalentieklasse dit een test is (als je de laatste regels niet meer nodig hebt, kun je deze velden vullen met een spatie):

[illegible]



## **EQUIVALENTIEKLASEN EN RANDWAARDEN (JUNIT)**

Ik heb de testcases voor Equivalentieklases en randwaarden omgezet naar de volgende JUnit-code om mijn software voor deze beslissing te testen.

<De JUnit-code (o.b.v. equivalentieklases en randwaarden)>

Hiermee test ik de volgende methode (waarin de besluiten op basis van de equivalentieklases worden bepaald):

<De code van de geteste methode>

## PAIRWISE TESTING

Voor Pairwise Testing maak ik gebruik van de volgende parameters (op basis waarvan een besluit wordt genomen; vereenvoudigd tot een set van 4 parameters die ieder 3, 2, 2 en 2 mogelijkheden hebben (anders wordt handmatig testen met Pairwise Testing veel te ingewikkeld; vul een spatie in, als je een optie niet wilt gebruiken):

| Parameter     | Optie 1   | Optie 2   | Optie 3   |
|---------------|-----------|-----------|-----------|
| <Parameter 1> | <Optie 1> | <Optie 2> | <Optie 3> |
| <Parameter 2> | <Optie 1> | <Optie 2> | <Optie 3> |
| <Parameter 3> | <Optie 1> | <Optie 2> | <Optie 3> |
| <Parameter 4> | <Optie 1> | <Optie 2> | <Optie 3> |

Na de tussenstappen in het proces heb ik de volgende testtabel met fysieke testgevallen opgesteld (vul spaties in, als er teveel testgevallen in de tabel staan voor jouw casus):

| <Parameter 1> | <Parameter 2> | <Parameter 3> | <Parameter 4> | Decision |
|---------------|---------------|---------------|---------------|----------|
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |
| <waarde>      | <waarde>      | <waarde>      | <waarde>      | <waarde> |

## **PAIRWISE TESTING (JUNIT)**

Ik heb de testcases voor Pairwise Testing omgezet naar de volgende JUnit-code om mijn software voor deze beslissing te testen.

<De JUnit-code (o.b.v. Pairwise Testing)>

Hiermee test ik de volgende methode (waarin het besluit op basis van de meegegeven parameters wordt bepaald):

<De code van de geteste methode>