**TESTONTWERP & -REALISATIE**

|  |  |
| --- | --- |
| **Naam** | Siddhart Ghogli |
| **Studentnummer** | 21067392 |
| **Klas** | 1 |

**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.

De applicatie is een managementtool waarmee medewerkers projecten kunnen aanmaken, beheren en toewijzen aan medewerkers. Het biedt ook functionaliteit voor het bijhouden van budgetten, klanten en medewerkers. Het doel is om het projectbeheerproces te stroomlijnen en te vereenvoudigen voor bedrijven.

**GITHUB**

Maak je GitHub-repo Private (zie de [‘Uitleg’ op Brightspace](https://brightspace.hhs.nl/d2l/le/lessons/18624/topics/549422) voor het GitHub-account van je docent en vergeet niet om je docent toegang te geven tot je repo):

https://github.com/Siddhart/OPT3-v2

**GESLAAGDE TESTS**

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

Afbeelding met tekst, Multimediasoftware, software, schermopname

Automatisch gegenereerde beschrijving

**SAMENGESTELDE DECISION**

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

Voorwaarde A: De einddatum van het project is later dan de startdatum.

Voorwaarde B: De medewerker die aan het project is toegewezen, bestaat.

Voorwaarde C: De klant die aan het project is gekoppeld, bestaat.

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

D = A && B && C

**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 = A && B && C** |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| <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 = A && B && C** |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| <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 = A && B && C** |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| <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:

D = (Start < End && klant bestaat && medewerker bestaat)

|  |  |  |  |
| --- | --- | --- | --- |
| **Voorwaarde A** | **Voorwaarde B** | **Voorwaarde C** | **D = A && B && C** |
| 2023-01-01, 2023-02-02 | 1 | 1 | Project aanmaken |
| 2023-02-02, 2023-01-01 | 1 | 1 | Geen project |
| 2023-01-01, 2023-02-02 | 0 | 1 | Geen project |
| 2023-01-01, 2023-02-02 | 1 | 0 | Geen project |
| <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 = A && B && C** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| <waarde> | <waarde> | <waarde> | <waarde> |

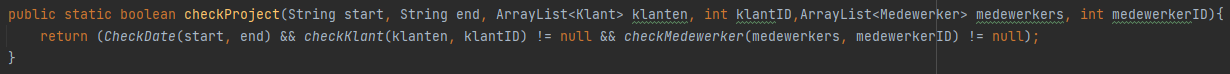
**MODIFIED CONDITION/DECISION COVERAGE (JUNIT)**

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

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

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



**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 | Budget tot 1001 | Project krijgt de tag “klein” |
| Equivalentieklasse 2 | Budget 1001 tot 5000 | Project krijgt de tag “normaal” |
| Equivalentieklasse 3 | vanaf 5000 | Project krijgt de tag “groot” |

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):

|  |  |  |
| --- | --- | --- |
| Equivalentieklasse | Fysiek testgeval | Besluit |
| Budget tot 1001 | 0 | klein |
| Budget tot 1001 | 1 | klein |
| Budget tot 1001 | 2 | klein |
| Budget tot 1001 | 999 | klein |
| Budget tot 1001 | 1000 | klein |
| Budget 1001 tot 5000 | 1001 | normaal |
| Budget 1001 tot 5000 | 1002 | normaal |
| Budget 1001 tot 5000 | 4999 | normaal |
| vanaf 5000 | 5000 | groot |
| vanaf 5000 | 5001 | groot |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |
| <Equivalentieklasse> | <Testgeval> | <Besluit> |

**EQUIVALENTIEKLASEN EN RANDWAARDEN (JUNIT)**

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

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

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

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

**PAIRWISE TESTING**

Voor Pairwise Testing maak ik gebruik van de volgende parameters (op basis waarvan een besluit wordt genomen; vereenvoudig 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** |
| budget | Kleiner dan 0 | Gelijk aan 0 | Groter dan 0 |
| projectPeriode | Geldige datum | Ongeldige datum | <Optie 3> |
| medewerker | Bestaat | Bestaat niet | <Optie 3> |
| klant | Bestaat | Bestaat niet | <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):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **budget** | **datum** | **medewerker** | **klant** | **Decision** |
| **Kleiner dan 0** | **Geldige datum** | **Betaat** | **Bestaat** | **Geen project** |
| **Kleiner dan 0** | **Ongeldige datum** | **Bestaat niet** | **Bestaat niet** | **Geen project** |
|  |  |  |  |  |
| **Gelijk aan 0** | **Geldige datum** | **Bestaat niet** | **Bestaat niet** | **Geen project** |
| **Gelijk aan 0** | **Ongeldige datum** | **Bestaat** | **Bestaat** | **Geen project** |
|  |  |  |  |  |
| **Groter dan 0** | **Geldige datum** | **Bestaat** | **Bestaat** | **Project aanmaken** |
| **Groter dan 0** | **Ongeldige datum** | **Bestaat niet** | **Bestaat niet** | **Geen project** |

**PAIRWISE TESTING (JUNIT)**

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

Afbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

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

Afbeelding met schermopname, tekst, Lettertype

Automatisch gegenereerde beschrijving