

Test af Klasser

Test af klasse: DataService

Metode 1): AnvendOrdination(int id, Dato dato)

Postcondition: Returnerer "Ordination anvendt!", hvis dato er inden for ordinationsperioden, returnerer "Dato ikke accepteret!!" hvis dato ikke er godkendt og returnere "Ordination ikke fundet" hvis ordinationen ikke kan findes.

Gyldige data: id > 0, id skal findes i databasen, dato skal være inden for ordinationsperioden

Ækvivalensklasser gyldige data:

- id > 0 && ordination startdato <= dato <= ordination slutdato

Ugyldige data: id < 0, id findes ikke i databasen, dato er uden for ordinationsperioden.

Ækvivalensklasser ugyldige data:

- id < 0 || ordination startdato >= dato >= ordination slutdato

Basisdata:

- Patient("121256-0512", "Jane Jensen", 63,4)
- Laegemiddel("Paracetamol", 0,1, 0,15, 0,16, "Styk")
- PN.OrdinalionId(2)
- Startdato: 01.01.2021
- Slutdato: 12.01.2021

Testcase med gyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC1:	id = 2 dato = 01.01.2021	string = "Ordination anvendt!" Antal ordinationer = 1	string = "Ordination anvendt!" Antal ordinationer = 1	✓
TC2:	id = 2 dato = 02.01.2021	string = "Ordination anvendt!" Antal ordinationer = 2	string = "Ordination anvendt!" Antal ordinationer = 2	✓

TC3:	id = 2 dato = 12.01.2021	string = "Ordination anvendt!" Antal Ordinationer = 3	string = "Ordination anvendt!" Antal ordinationer = 3	✓
TC4:	id = 2 dato = 11.01.2021	string = "Ordination anvendt!" Antal ordinationer = 4	string = "Ordination anvendt!" Antal ordinationer = 4	✓
TC5:	id = 2 dato = 06.01.2021	string = "Ordination anvendt!" Antal ordinationer = 5	string = "Ordination anvendt!" Antal ordinationer = 5	✓

Testcase med ugyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC6:	id = -1 dato = 06.01.2021	string = "Ordination ikke fundet" Antal ordinationer = 5	string = "Ordination ikke fundet" Antal ordinationer = 5	✓
TC7:	id = 2 dato = 22.01.2021	string = "Dato ikke accepteret!!" Antal ordinationer = 5	string = "Dato ikke accepteret!!" Antal ordinationer = 5	✓
TC8:	id = 2 dato = 13.1.2021	string = "Dato ikke accepteret!!" Antal ordinationer = 5	string = "Dato ikke accepteret!!" Antal ordinationer = 5	✓
TC9:	id = -1 dato =	string = "Ordination ikke fundet"	string = "Ordination ikke	✓

	23.11.2022	Antal ordinationer = 5	fundet" Antal ordinationer = 5	
--	------------	------------------------	-----------------------------------	--

Test af klasse: Ordination

Metode 1): antalDage()

Postcondition: Returnerer antal dage mellem to datoer, hvis datoerne er gyldige og returnerer -1 hvis ikke.

Gyldige data: slutDen skal være større eller lig med startDen

Ækvivalensklasser gyldige data:

- startDen <= slutDen

Ugyldige data: slutDen er mindre end startDen

Ækvivalensklasser ugyldige data:

- startDen > slutDen

Testcase med gyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC1:	startDen = 20.11.2022 slutDen = 23.11.2022	3	3	✓
TC2:	startDen = 20.11.2022 slutDen = 20.11.2022	0	0	✓

Testcase med ugyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC3:	startDen = 23.11.2022 slutDen = 20.11.2022	-1	-1	✓

Test af klasse: DagligFast

Metode 1): doegnDosis()

Postcondition: Returnerer et antal der er ≥ 0 . Giver et samlet antal enheder pr. dag over en gyldig ordinationsperiode. Returnerer -1 hvis én eller flere af dagsdoserne er mindre end 0.

Note: Tester kun en af de fire variabler, da vi kender til funktionaliteterne af dem alle. Logikken er den samme bag de fire variabler, så vi indtager at det kun er nødvendigt at teste én.

Gyldige data: Morgen-, Middag-, Aften- og NatDosis antal skal være ≥ 0 .

Ækvivalensklasser gyldige data:

- MorgenDosis.antal ≥ 0
- MiddagDosis.antal ≥ 0
- AftenDosis.antal ≥ 0
- NatDosis.antal ≥ 0

Ugyldige data: Morgen-, Middag-, Aften- og NatDosis antal er < 0

Ækvivalensklasser ugyldige data:

- MorgenDosis.antal < 0
- MiddagDosis.antal < 0
- AftenDosis.antal < 0
- NatDosis.antal < 0

Testcase med gyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC1:	MorgenDosis = 1 MiddagDosis = 0 AftenDosis = 0 NatDosis = 0	1	1	✓
TC2:	MorgenDosis = 0 MiddagDosis = 0 AftenDosis = 0 NatDosis = 0	0	0	✓

Testcase med ugyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC3:	MorgenDosis = -1 MiddagDosis = 1 AftenDosis = 1 NatDosis = 1	-1	-1	✓

**Test af klasse: DagligSkæv
Metode 1): doegnDosis()****Postcondition:**

Finder det totale antal doser for hver dag i en gyldig ordineringsperiode

Gyldige data:

Positive tal

Ækvivalensklasser gyldige data:

- doser.antal >= 0

Ugyldige data:

Antal enheder under 0 eller null

Ækvivalensklasser ugyldige data:

- doser.antal < 0
- null

Testcase med gyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC1:	dosis1 = 5 dosis2 = 10	15	15	✓
TC2:	dosis3 = 0	0	0	✓

Testcase med ugyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC3:	dosis4 = -1	0	0	✓
TC4:	dosis5 = -10	0	0	✓

Test af klasse: PN

Metode 1): givDosis(Dato givesDen)

Postcondition:

Registrerer at der er givet en dose på en givet dato. Returnerer true hvis givesDen er inden for ordinationens gyldighedsperiode, og returnerer false ellers.

Gyldige data:

Dato som ligger indenfor ordinations gyldighedsperiode samt datoer der findes.

Ækvivalensklasser gyldige data:

- StartDato <= givesDen <= Slutdato

Ugyldige data:

Dato som ligger udenfor ordinations gyldighedsperiode samt datoer som ikke findes.

Ækvivalensklasser ugyldige data:

- Startdato > givesDen
- Slutdato < givesDen

Basisdata

- Startdato: 01.01.2021
- Slutdato: 12.01.2021

Testcase med gyldig data:

TestCase	Input	Forventet resultat	Aktuelt output	State
TC1:	givesDen = 05.01.2021	True	True	✓
TC2:	givesDen = 01.01.2021	True	True	✓

TC3	givesDen = 12.01.2021	True	True	✓
-----	--------------------------	------	------	---

Testcase med ugyldig data:



TestCase	Input	Forventet resultat	Aktuelt output	State
TC4:	givesDen = 31.12.2020	False	False	✓
TC5:	givesDen = 13.01.2021	False	False	✓

SYSTEMTEST

Test af UC Opret Ordination :

Use Case Name:	Opret Ordination	
Scenario:		
Trigger Event:	Patient er diagnosticeret og skal have en medicinsk behandling	
Brief Description:		
Actors:	Sygeplejerske	
Related use cases:		
Stakeholders:	Patient, læge, sygeplejerske	
Precondition:	Patient og lægemidler er registreret i systemet	
Postcondition:		
Flow of events:	<p>Actor</p> <ol style="list-style-type: none"> 1. Find patient frem ud fra cpr-nr 2. Vælg lægemiddel 3. Vælg typen af ordination og gå til opret 4. Hvis ej PN beslut ud fra beregnet dosisPrDøgn hvor mange dosis og med hvilke antal enheder på doserne, der skal oprettes. 5. Hvis dagligFast angiv start og slutdato og de 4 gange antal enheder og vælg opret doser. Hvis PN angiv start og slutdato, angiv antal enheder. Hvis dagligSkæv angiv start og slutdato og så længe der ønskes oprettet flere og den anbefalede dosisPrDøgn ikke overstiges, angiv antal enheder og en tid og vælg opret dosis. 6. Afslut oprettelsen 	<p>System</p> <p>3.1 Beregner anbefalet dosisPrDøgn og viser det på skærmen</p> <p>5.1 Hvis dagligSkæv holder systemet øje med den samlede mængde af de indtil nu oprettede doser og viser dem på skærmen. Der gives besked hvis anbefalet dosis overskrides.</p>
Exceptional Flows:	<p>5 Hvis startdato større end slutdato så gives fejlmeddelelse</p>	

Testcase ID	What do we test?	Initial system state	Input	Expected output	State
-------------	------------------	----------------------	-------	-----------------	-------

TC 1	Normal flow	'Opret ordination' screen	Find CPR-nummer på listen over patienter Vælger lægemiddel på listen Vælg ordinationstype	Den valgte patient bliver markeret Det valgte lægemiddel bliver markeret Valgte ordination bliver noteret	
TC2	Exceptional flow	Opret PN, DagligSkæv eller DagligFast	Større startDato end slutDato	Systemet giver en fejlmeddelelse og ordination bliver ikke oprettet	

Overvejelser:

- Vi kunne godt have lavet flere TestCases, men da vi vurderede, at UseCase flow-of-events nr. 4-6 ikke kan testes og / eller tager udgangspunkt i nogle funktionaliteter, som ikke er implementeret i vores program, har vi undladt dette.