

TESTAUSPÖYTÄKIRJA

Kipukalenteri

Jaana Pusa
pusa.jaana@gmail.com

Sisällysluettelo

1	Johdanto	2
2	Testauksen tulos	2
3	Testitapausten läpikäynti.....	2
3.1	Manuaalisten testien tulokset	2
3.1.1	Tietokannan testaus SQL Server Management Studioissa	2
3.1.2	Kontrollerien manuaalinen testaus Postman-sovelluksella.....	22
3.1.3	Käyttöliittymän manuaalinen testaus.....	44
3.2	Automaattinen testaus	75
3.2.1	Yksikkötestaus Jest-kirjastolla.....	75
3.2.2	Integraatiotestaus Alba-kirjastolla.....	87
3.2.3	End-to-end testaus Cypress-kirjastolla	99
3.3	Hyväksymistestaus.....	108
3.3.1	Use case 1: Kipumerkinnän tallennus, muokkaus ja poisto	109
3.3.2	Use case 2: Kivun sijainnin lisäys.....	110
3.3.3	Use case 3: Erillisen muistiinpanon lisäys	111
3.3.4	Use case 4: Kipumerkintöjen suodatus.....	112

1 Johdanto

Tässä projektissa tuotetaan Kipukalenteri -web-sovellus Careerian tieto- ja viestintätekniikan ohjelmistotuotannon opiskelijan toimesta. Projekti on osa opiskelijan näyttötutkinnon suoritusta. Tämä dokumentti on sovelluksen testauspöytäkirja.

Kipukalenteri-sovelluksen testauspöytäkirjassa käydään läpi testaussuunnitelman mukaan suoritettujen testien lopputulokset ja niistä mahdollisesti aiheutuneet virheiden korjaukset tai muut toimenpiteet.

Testauksen kohde ja tavoitteet sekä testausympäristö on kuvattu tarkemmin testaussuunnitelmassa. Yksittäiset testitapaikset tuloksineen on kuvattu tässä dokumentissa.

2 Testauksen tulos

Manuaalisen ja automaattisen testauksen perusteella sovellus sekä sen eri osat toimivat odotusten mukaan. Testauksessa ilmeni pääasiassa kehitystarpeita ja pieniä ulkoasun tai sovelluksen antamiin ilmoituksiin liittyviä korjauksia. Hyväksymistestauksen perusteella sovellus vastaa määrittelykuvaustossa laadittuja käyttötapauskuvauksia, eikä siinä ole olennaisia puutteita.

Testauksessa saatiin myös lisätietoa kehitystarpeista sovelluksen toimintaan ja lisäominaisuksiin.

3 Testitapausten läpikäynti

Testitapaikset ja testauksen rajaukset on käsitelty testaussuunnitelmassa. Tässä pöytäkirjassa käydään läpi testitapaus kerrallaan testauksen tulokset.

3.1 Manuaalisten testien tulokset

3.1.1 Tietokannan testaus SQL Server Management Studiossa

Tietokanta testattiin manuaalisesti lisäämällä, poistamalla ja muokkaamalla testidataa sekä suorittamalla tietokantahakuja. Tietokantaan syötettiin sekä oikeellista testidataa että virheellistä dataa yksi tai useita rivejä kerrallaan. Sen jälkeen tietokantataulujen tietoja poistettiin ja muokattiin.

Testaus suoritettiin tietokannan valmistuttua prototyppikäyttöön ja lopuksi uudestaan lopullisten tietokantamuutosten jälkeen. Tässä pöytäkirjassa raportoidut tulokset ovat lopullisten testien tuloksia.

Kaikki testit onnistuivat odotusten mukaan. Manuaalisen testauksen perusteella tietokannan toiminnot toimivat tietokantatauluissa SQL Serverin kautta normaalisti.

3.1.1.1 Notes-taulun manuaalinen testaus

3.1.1.1.1 Syötetään tauluun yksi rivi oikeellisia tietoja

Syötetään tauluun rivi, jossa on vähintään kaikki vaaditut tiedot oikeassa tietotyypissä.

```
Insert into Notes (NoteText, NoteDate, UserId) values ('SQL Server DB manual test 3.1.1.1A', '2022-06-03 10:57:00', 'testiad')
```

Odottettu tulos: SLQ Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty rivi löytyy tietokantataulusta.

Testin tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)
```

Completion time: 2022-06-03T17:55:02.7186095+03:00

Syötetty rivi löytyy Notes-taulusta suurimmalla NoteID:llä:

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1A	2022-06-03 10:57:00.000	testiad

Testi onnistui odotetun tuloksen mukaan.

3.1.1.1.2 Syötetään tauluun useita rivejä oikeellisia tietoja

Syötetään tauluun kolme riviä, joissa on vähintään vaaditut tiedot oikeassa tietotyypissä.

```
Insert into Notes (NoteText, NoteDate, UserId) values
('SQL Server DB manual test 3.1.1.1B', '2022-06-03 11:57:00', 'testiad'),
('SQL Server DB manual test 3.1.1.1B', '2022-06-03 12:57:00', 'testiad'),
('SQL Server DB manual test 3.1.1.1B', '2022-06-03 13:57:00', 'testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "3 rows affected" ja syötetyt rivit löytyvät tietokantataulusta.

Testin tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(3 rows affected)
```

Completion time: 2022-06-03T17:57:28.8970871+03:00

Syötetyt rivit löytyvät Notes-taulusta suurimmilla NoteID:illä:

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1A	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	164	SQL Server DB manual test 3.1.1.1B	2022-06-03 12:57:00.000	testiad
9	165	SQL Server DB manual test 3.1.1.1B	2022-06-03 13:57:00.000	testiad

Testi onnistui odotetun tuloksen mukaan.

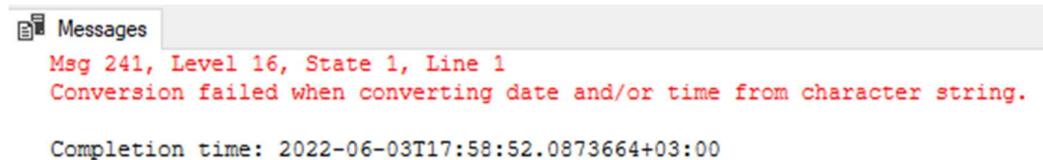
3.1.1.1.3 Syötetään tauluun yksi rivi virheellisiä tietoja

Syötetään tauluun yksi rivi, jossa NoteDate –kentässä on päivämääränsä sijaan tekstityyppistä tietoa.

```
Insert into Notes (NoteText, NoteDate, UserId) values
('SQL Server DB manual test 3.1.1.1C','Manual test','testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "Conversion failed when converting the date and/or time from character string."

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:



Messages

Msg 241, Level 16, State 1, Line 1
Conversion failed when converting date and/or time from character string.

Completion time: 2022-06-03T17:58:52.0873664+03:00

Syötettyä riviä ei löydy tietokantataulusta:

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1A	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	164	SQL Server DB manual test 3.1.1.1B	2022-06-03 12:57:00.000	testiad
9	165	SQL Server DB manual test 3.1.1.1B	2022-06-03 13:57:00.000	testiad

Testi onnistui odotetun tuloksen mukaan.

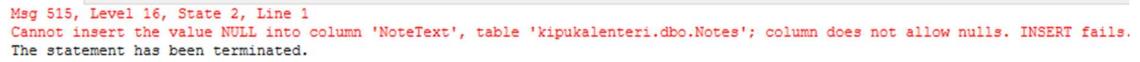
3.1.1.1.4 Syötetään tauluun rivi, josta puuttuu pakolliseksi määritellyjä tietoja

Syötetään tauluun rivi, jossa NoteText-kentästä puuttuu tieto.

```
Insert into Notes (NoteText, NoteDate, UserId) values
(NULL, '22-06-03 10:57:00', 'testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "Cannot insert the value NULL into Column 'NoteText', table 'kipukalenteri.dbo.Notes'; column does not allow nulls. INSERT fails. The statement has been terminated" eikä syötettyjä tietoja löydy tietokannasta.

Tulos: SLQ Server Management Studio antoi odotetun ilmoituksen:



Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'NoteText', table 'kipukalenteri.dbo.Notes'; column does not allow nulls. INSERT fails.
The statement has been terminated.

Completion time: 2022-06-03T18:01:49.1032034+03:00

Syötettyä riviä ei löydy tietokantataulusta:

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1A	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	164	SQL Server DB manual test 3.1.1.1B	2022-06-03 12:57:00.000	testiad
9	165	SQL Server DB manual test 3.1.1.1B	2022-06-03 13:57:00.000	testiad

Testi onnistui odotusten mukaan.

3.1.1.1.5 Muokataan tauluun syötettyjä tietoja

Muokataan kohdassa 3.1.1.1A syötettyä riviä vaihtamalla NoteText-kenttää teksti "SQL Server DB manual test 3.1.1.1E".

```
Update Notes Set NoteText = 'SQL Server DB manual test 3.1.1.1E' Where NoteID = 162
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty muutos löytyy oikealta tietokantariviltä.

Tulos: SLQ Server Management Studio antoi odotuksen mukaisen ilmoituksen

```
(1 row affected)
```

```
Completion time: 2022-06-03T18:18:31.9048606+03:00
```

Muokattu tieto löytyy tietokantataulusta oikealta NoteID:ltä.

	NoteID	NoteText	NoteDate	UserId
1	162	SQL Server DB manual test 3.1.1.1E	2022-06-03 10:57:00.000	testiad

Testi onnistui odotusten mukaan.

3.1.1.1.6 Haetaan taulun kaikki tiedot

Haetaan tietokantataulun kaikki rivit.

```
Select * From Notes
```

Odottettu tulos: Taulun rivit näkyvät Results –osiossa.

Tulos: Testi onnistui odotusten mukaan ja taulun rivit näkyvät Results-osiossa:

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1A	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	164	SQL Server DB manual test 3.1.1.1B	2022-06-03 12:57:00.000	testiad
9	165	SQL Server DB manual test 3.1.1.1B	2022-06-03 13:57:00.000	testiad

3.1.1.1.7 Haetaan tietoja määritellyllä hakuehdolla

Haetaan Notes-taulun rivi, jonka NoteID vastaa testitapaus 3.1.1.1A-kohdassa syötetyn rivin id:tä

```
Select * From Notes Where NoteID = 162
```

Odottettu tulos: Kohdassa 3.1.1.1A syötetty rivi näkyy Results-osiossa.

Tulos: Testi onnistui odotusten mukaan ja haettu rivi näkyy Results-osiossa:

	NoteID	NoteText	NoteDate	UserId
1	162	SQL Server DB manual test 3.1.1.1E	2022-06-03 10:57:00.000	testiad

3.1.1.1.8 Poistetaan taulusta rivi

Poistetaan yksi kohdassa 3.1.1.1B-syötetyistä riveistä tietokantataulusta päivämääräehdon perusteella.

```
Delete from Notes Where NoteDate = '2022-06-03 12:57:00'
```

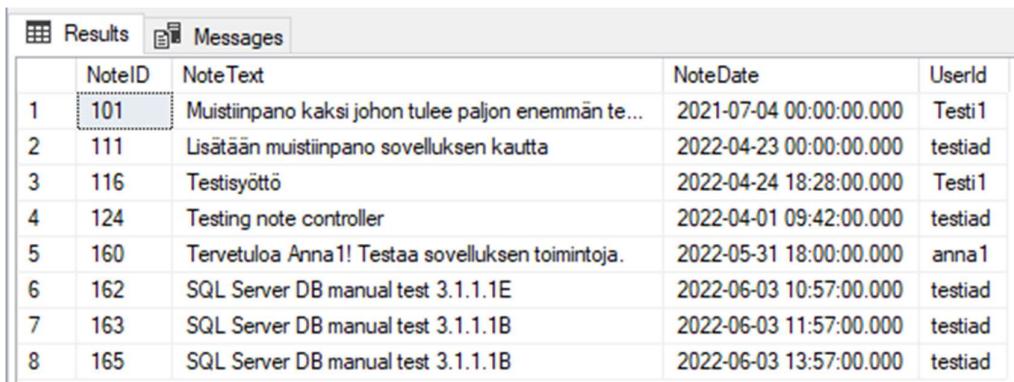
Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected", eikä poistettua riviä löydy tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotetun ilmoituksen:

```
(1 row affected)
```

```
Completion time: 2022-06-03T18:21:53.3987255+03:00
```

Poistettu rivi, jonka NoteID oli 164, ei löydy tietokantataulusta:



	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1E	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	165	SQL Server DB manual test 3.1.1.1B	2022-06-03 13:57:00.000	testiad

Testi onnistui odotusten mukaan.

3.1.1.2 PainLocation-taulun manuaalinen testaus

3.1.1.2.1 Syötetään tauluun yksi rivi oikeellisia tietoja

Syötetään tauluun rivi, jossa on vähintään kaikki vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLocation(LocationName) values ('SQL Server DB manual test 1')
```

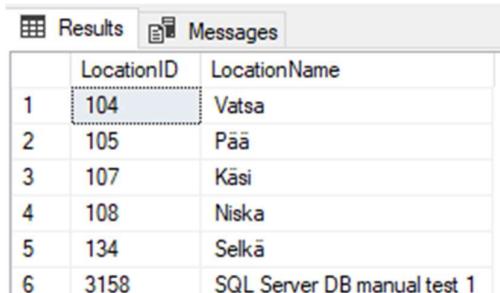
Odottettu tulos: SLQ Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty rivi löytyy tietokantataulusta.

Tulos: SLQ Server Management Studio antoi odotetun ilmoituksen:

```
(1 row affected)
```

```
Completion time: 2022-06-03T18:21:53.3987255+03:00
```

Rivi löytyy PainLocation-taulusta suurimmalla LocationID:llä.



	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 1

Testi onnistui odotusten mukaan.

3.1.1.2.2 Syötetään tauluun useita rivejä oikeellisia tietoja

Syötetään tauluun kolme riviä, joissa on vähintään vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLocation(LocationName) values
('SQL Server DB manual test 2'),
('SQL Server DB manual test 3'),
('SQL Server DB manual test 4')
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "3 rows affected" ja syötetyt rivit löytyvät tietokantataulusta.

Tulos: SLQ Server Management Studio antoi odotetun ilmoituksen:

(3 rows affected)

Completion time: 2022-06-03T17:57:28.8970871+03:00

Syötetty rivit löytyvät tietokantataulusta suurimmalla LocationID:llä:

	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 1
7	3159	SQL Server DB manual test 2
8	3160	SQL Server DB manual test 3
9	3161	SQL Server DB manual test 4

Testi onnistui odotusten mukaan.

3.1.1.2.3 Syötetään tauluun rivi, josta puuttuu pakolliseksi määritellyjä tietoja

Syötetään tauluun rivi, jossa LocationName-kentästä puuttuu tieto.

```
Insert into PainLocation(LocationName) values (NULL)
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "Cannot insert the value NULL into Column 'LocationName', table 'kipukalenteri.dbo.PainLocation'; column does not allow nulls. INSERT fails. The statement has been terminated" eikä syötettyjä tietoja löydy tietokannasta.

Tulos: SLQ Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'LocationName', table 'kipukalenteri.dbo.PainLocation'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```

Completion time: 2022-06-03T18:32:34.4316080+03:00

Syötettyä riviä ei löydy tietokantataulusta:

	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 1
7	3159	SQL Server DB manual test 2
8	3160	SQL Server DB manual test 3
9	3161	SQL Server DB manual test 4

Testi onnistui odotusten mukaan.

3.1.1.2.4 Muokataan tauluun syötettyjä tietoja

Muokataan kohdassa 3.1.1.2A syötettyä rivää vaihtamalla NoteText-kenttään teksti "SQL Server DB manual test 5".

```
Update PainLocation Set LocationName = 'SQL Server DB manual test 5' Where  
LocationID = 3158
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty muutos löytyy oikealta tietokantariviltä.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)
```

```
Completion time: 2022-06-03T18:34:39.4293475+03:00
```

Muokattu tieto löytyy oikealta tietokantataulun riviltä, jonka LocationID on annetun mukainen.

	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 5
7	3159	SQL Server DB manual test 2
8	3160	SQL Server DB manual test 3
9	3161	SQL Server DB manual test 4

Testi onnistui odotusten mukaan.

3.1.1.2.5 Haetaan taulun kaikki tiedot

Haetaan tietokantataulun kaikki rivit.

```
Select * From PainLocation
```

Odotettu tulos: Taulun rivit näkyvät Results -osiolla.

Tulos: Testi onnistui odotusten mukaan ja taulun rivit näkyvät Results-osiolla:

	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 5
7	3159	SQL Server DB manual test 2
8	3160	SQL Server DB manual test 3
9	3161	SQL Server DB manual test 4

3.1.1.2.6 Haetaan tietoja määritellyllä hakuehdolla

Haetaan taulun rivi, jonka LocationId vastaa testitapaus 3.1.1.2A-kohdassa syötetyn rivin päivämäärää.

```
Select * From PainLocation Where LocationID = 3158
```

Odottettu tulos: Kohdassa 3.1.1.2A haettu rivi näkyy Results-osiossa.

Tulos: Testi onnistui odotusten mukaan ja haluttu rivi näkyy Results-osiossa:

Results		Messages	
	LocationID	LocationName	
1	3158	SQL Server DB manual test 5	

3.1.1.2.7 Poistetaan taulusta rivi

Poistetaan yksi kohdassa 3.1.1.2B-syötetyistä riveistä tietokantataulusta.

```
Delete from PainLocation Where LocationID = 3160
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" ja poistettu rivi puuttuu tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)
```

```
Completion time: 2022-06-03T18:36:54.1688326+03:00
```

Poistettu rivi ei löydy tietokantataulusta:

Results		Messages	
	LocationID	LocationName	
1	104	Vatsa	
2	105	Pää	
3	107	Käsi	
4	108	Niska	
5	134	Selkä	
6	3158	SQL Server DB manual test 5	
7	3159	SQL Server DB manual test 2	
8	3161	SQL Server DB manual test 4	

Testi onnistui odotusten mukaan.

3.1.1.3 PainLog-taulun manuaalinen testaus

3.1.1.3.1 Syötetään tauluun yksi rivi oikeellisia tietoja

Syötetään tauluun rivi, jossa on vähintään kaikki vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserID)
values
```

```
(6, '2022-06-02 11:45:00', '2022-06-03 12:45:00', 107, 'SQL Server manual test
3.1.1.3A', 'testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "LogId-table after insert trigger Tr_CalculateDurationMinutes DONE" ja "1 row affected". Syötetty rivi löytyy tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaiset ilmoitukset:

```
LogId-table after insert trigger Tr_CalculateDurationMinutes DONE
(1 row affected)

Completion time: 2022-06-03T18:42:15.4777654+03:00
```

Syötetty rivi löytyy tietokantataulusta:

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3A	testiad

Testi onnistui odotusten mukaan.

3.1.1.3.2 Syötetään tauluun useita rivejä oikeellisia tietoja

Syötetään tauluun kolme riviä, joissa on vähintään vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserId)
values
(3, '2022-06-01 11:30:00', '2022-06-03 18:00:00', 107, 'SQL Server manual test
3.1.1.3B', 'testiad'),
(4, '2022-06-01 10:30:00', '2022-06-02 11:30:00', 108, 'SQL Server manual test
3.1.1.3B', 'testiad'),
(5, '2022-06-01 08:55:00', '2022-06-01 09:45:00', 108, 'SQL Server manual test
3.1.1.3B', 'testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoitukseen "LogId-table after insert trigger Tr_CalculateDurationMinutes DONE" ja "3 rows affected" ja syötetyt rivit löytyvät tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaiset ilmoitukset:

```
LogId-table after insert trigger Tr_CalculateDurationMinutes DONE
(3 rows affected)

Completion time: 2022-06-03T18:45:59.9845514+03:00
```

Syötetyt rivit löytyvät tietokantataulusta suurimmilla LogID:illä:

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3A	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4056	4	2022-06-01 10:30:00.000	2022-06-02 11:30:00.000	1500	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

Testi onnistui odotusten mukaan. Triggerin antama ilmoitus on kuitenkin hieman epäjohdonmukainen, mutta asia katsotaan tarkemmin triggerin testauksen yhteydessä.

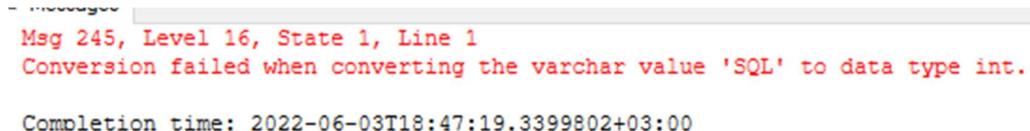
3.1.1.3.3 Syötetään tauluun yksi rivi virheellisiä tietoja

Syötetään tauluun yksi rivi, jossa LocationId –kentässä on numeron sijaan tekstityyppistä tietoa.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserId)
values
(3, '2022-06-01 11:30:00', '2022-06-03 18:00:00', 'SQL', 'SQL Server manual test
3.1.1.3B','testiad')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "Conversion failed when converting the varchar value 'SQL' to data type int. Riviä ei löydy tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:



Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'SQL' to data type int.
Completion time: 2022-06-03T18:47:19.3399802+03:00

Syötettyä riviä ei löydy tietokantataulusta:

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3A	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4056	4	2022-06-01 10:30:00.000	2022-06-02 11:30:00.000	1500	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

Testi onnistui odotusten mukaan.

3.1.1.3.4 Syötetään tauluun rivi, josta puuttuu pakolliseksi määritellyjä tietoja

Syötetään tauluun rivi, jossa UserId-kentästä puuttuu tieto.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserId)
values
(3, '2022-06-01 11:30:00', '2022-06-03 18:00:00', 107, 'SQL Server manual test
3.1.1.3B', NULL)
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "Cannot insert the value NULL into Column 'UserId', table 'kipukalenteri.dbo.PainLogs'; column does not allow nulls. INSERT fails. The statement has been terminated" eikä syötettyjä tietoja löydy tietokannasta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:



Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'UserId', table 'kipukalenteri.dbo.PainLog'; column does not allow nulls. INSERT fails.
The statement has been terminated.
Completion time: 2022-06-03T18:48:44.7659318+03:00

Syötettyä riviä ei löydy tietokantataulusta:

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3A	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4056	4	2022-06-01 10:30:00.000	2022-06-02 11:30:00.000	1500	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

Testi onnistui odotusten mukaan. Testitapausen mukainen tilanne ei käyttöliittymästä toteutettuna tule olemaan mahdollinen, koska UserId-tieto tulee lisättyihin tietoihin mukaan automaattisesti.

3.1.1.3.5 Muokataan tauluun syötettyjä tietoja

Muokataan kohdassa 3.1.1.3A syötettyä riviä vaihtamalla Notes-kenttään teksti "DB manual test 3.1.1.3E".

```
Update PainLog Set Notes = 'SQL Server DB manual test 3.1.1.3E' Where logID = 4054
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "LogId-table after update trigger Tr_CalculateDurationMinutes DONE" ja "LogId-table after insert trigger

Tr_CalculateDurationMinutes DONE" ja "1 row affected" ja syötetty muutos löytyy oikealta tietokantariviltä.

Tulos: SQL Server Management Studio antoi odotusten mukaisen ilmoitukseen:

```
LogId-table after update trigger Tr_CalculateDurationMinutes DONE
LogId-table after insert trigger Tr_CalculateDurationMinutes DONE

(1 row affected)

Completion time: 2022-06-03T18:50:11.3395027+03:00
```

Syötetty muutos löytyy oikealta riviltä:

	Results	Messages									
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4056	4	2022-06-01 10:30:00.000	2022-06-02 11:30:00.000	1500	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

Testi onnistui odotetun tuloksen mukaan.

3.1.1.3.6 Haetaan taulun kaikki tiedot

Haetaan tietokantataulun kaikki rivit.

```
Select * From PainLog
```

Odotettu tulos: Taulun rivit näkyvät Results –osiossa.

Tulos: Testi onnistui odotetun tuloksen mukaan, taulun rivit näkyvät Results-osiassa:

	Results	Messages									
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4056	4	2022-06-01 10:30:00.000	2022-06-02 11:30:00.000	1500	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

3.1.1.3.7 Haetaan tietoja määritellyllä hakuehdolla

Haetaan taulun rivit, joiden LocationID-kentässä on arvo 107

```
Select * From PainLog Where LocationID = 107
```

Odotettu tulos: Results-osiassa näkyvät rivit, joiden LocationID on 107.

Tulos: Testi onnistui odotetun tuloksen mukaan, halutut rivit näkyvät Results-osiassa:

	Results	Messages									
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad

3.1.1.3.8 Poistetaan taulusta rivi

Poistetaan yksi kohdassa 3.1.1.3B-syötetyistä riveistä tietokantataulusta hakuehtona LocationID:stä ja Durationia.

```
Delete from PainLog Where LocationID = 108 and Duration = 1500
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" eikä poistettua riviä löydy tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)
```

Completion time: 2022-06-03T18:50:11.3395027+03:00

Poistettua riviä ei löydy tietokantataulusta:

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad

Testi onnistui odotetun tuloksen mukaisesti.

3.1.1.4 Users-taulun manuaalinen testaus

3.1.1.4.1 Syötetään tauluun yksi rivi oikeellisia tietoja

Syötetään tauluun rivi, jossa on vähintään kaikki vaaditut tiedot oikeassa tietotyypissä.

```
Insert into Users (Username, Email, Password) values
('Testi2','testi@testi','ManualTest1')
```

Odotettu tulos: SLQ Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty rivi löytyy tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotetun mukaisen ilmoituksen:

```
(1 row affected)
```

Completion time: 2022-06-03T18:58:24.4029215+03:00

Syötetty rivi löytyy Users-tietokantataulusta:

	Username	Email	Password	Token
1	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
2	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	Testi1	pusajaana@gmail.com	KaivoKello01	NULL
4	Testi2	testi@testi	ManualTest1	NULL
5	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

Testi onnistui odotetun tuloksen mukaisesti.

3.1.1.4.2 Syötetään tauluun useita rivejä oikeellisia tietoja

Syötetään tauluun kolme riviä, joissa on vähintään vaaditut tiedot oikeassa tietotyypissä.

```
Insert into Users (Username, Email, Password) values
('Testi3','testi@testi','ManualTest1'),
('Testi4','testi@testi','ManualTest2'),
('Testi5','testi@testi','ManualTest3')
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "3 rows affected" ja syötetyt rivit löytyvät tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(3 rows affected)
```

```
Completion time: 2022-06-03T19:00:28.3303273+03:00
```

Syötetyt rivit löytyvät tietokantataulusta:

	Username	Email	Password	Token
1	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
2	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
4	Testi2	testi@testi	ManualTest1	NULL
5	Testi3	testi@testi	ManualTest1	NULL
6	Testi4	testi@testi	ManualTest2	NULL
7	Testi5	testi@testi	ManualTest3	NULL
8	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

Testi onnistui odotetun tuloksen mukaisesti.

3.1.1.4.3 Syötetään tauluun rivi, josta puuttuu pakolliseksi määritellyjä tietoja

Syötetään tauluun rivi, jossa Username-kentästä puuttuu tieto.

```
Insert into Users (Username, Email, Password) values  
(NULL, 'testi@testi', 'ManualTest1')
```

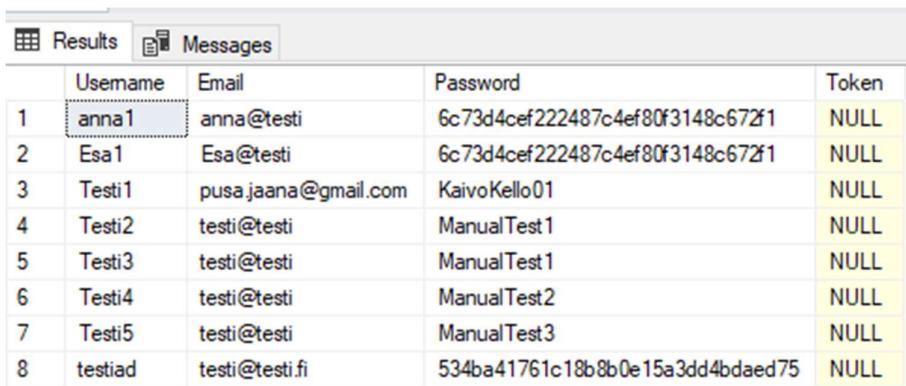
Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "Cannot insert the value NULL into Column 'Username', table 'kipukalenteri.dbo.Users'; column does not allow nulls. INSERT fails. The statement has been terminated" eikä syötettyjä tietoja löydy tietokannasta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
Msg 515, Level 16, State 2, Line 1  
Cannot insert the value NULL into column 'Username', table 'kipukalenteri.dbo.Users'; column does not allow nulls. INSERT fails.  
The statement has been terminated.
```

```
Completion time: 2022-06-03T19:01:23.1001901+03:00
```

Syötettyä riviä ei löydy tietokantataulusta:



	Username	Email	Password	Token
1	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
2	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
4	Testi2	testi@testi	ManualTest1	NULL
5	Testi3	testi@testi	ManualTest1	NULL
6	Testi4	testi@testi	ManualTest2	NULL
7	Testi5	testi@testi	ManualTest3	NULL
8	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

Testi onnistui odotetun tuloksen mukaisesti.

3.1.1.4.4 Muokataan tauluun syötettyjä tietoja

Muokataan kohdassa 3.1.1.4A syötettyä rivää vaihtamalla Username-kenttään teksti "ManualTest4D"

```
Update Users set Username = 'ManualTest4D' Where Username = 'Testi2'
```

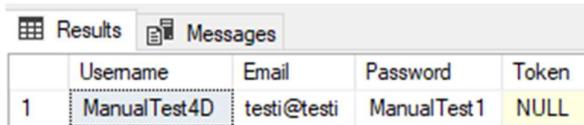
Odottettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" ja syötetty muutos löytyy oikealta tietokantariviltä.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)

Completion time: 2022-06-03T19:03:16.6400792+03:00
```

Muokattu rivi löytyi tietokantataulusta.



	Username	Email	Password	Token
1	ManualTest4D	testi@testi	ManualTest1	NULL

Testi onnistui odotetun tuloksen mukaisesti. Käyttöliittymässä käyttäjälle ei anneta mahdollisuutta muokata käyttäjänimeä.

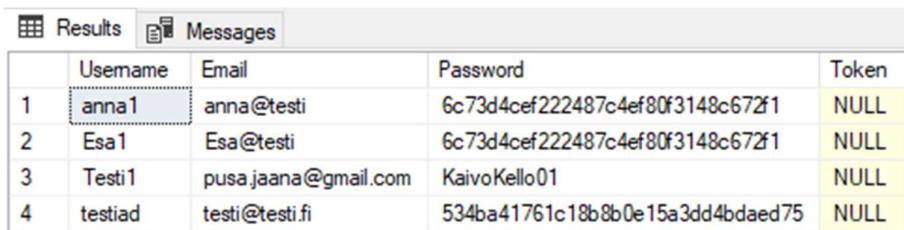
3.1.1.4.5 Haetaan taulun kaikki tiedot

Haetaan tietokantataulun kaikki rivit.

```
Select * From Users
```

Odottettu tulos: Taulun rivit näkyvät Results -osiossa.

Tulos: Testi onnistui odotetun tuloksen mukaisesti ja haetut rivit näkyvät Results-osiolla:



	Username	Email	Password	Token
1	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
2	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
4	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

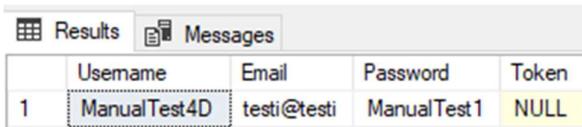
3.1.1.4.6 Haetaan tietoja määritellyllä hakuehdolla

Haetaan taulun rivi, jonka käyttäjänimi vastaa testitapaus 3.1.1.4D-kohdassa muokattua riviä.

```
Select * From Users Where Username = 'ManualTest4D'
```

Odotettu tulos: Kohdassa 3.1.1.4D syötetty rivi näkyy Results-osiossa.

Tulos: Testi onnistui odotetun tuloksen mukaisesti ja haettu rivi näkyy Results-osiossa:



	Username	Email	Password	Token
1	ManualTest4D	testi@testi	ManualTest1	NULL

3.1.1.4.7 Poistetaan taulusta rivi

Poistetaan yksi kohdassa 3.1.1.4B-syötetyistä riveistä tietokantataulusta.

```
Delete from Users Where Username = 'Testi4'
```

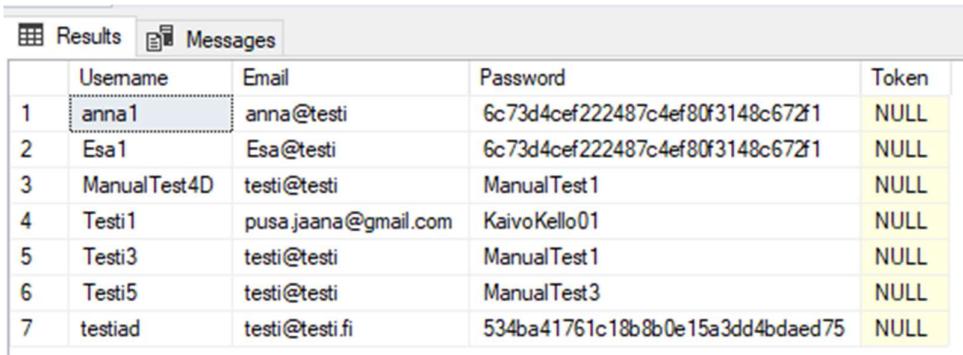
Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "1 row affected" ja poistettu rivi puuttuu tietokantataulusta.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
(1 row affected)
```

```
Completion time: 2022-06-03T19:05:40.0249447+03:00
```

Poistettua riviä ei löydy tietokantataulusta:



	Username	Email	Password	Token
1	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
2	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	ManualTest4D	testi@testi	ManualTest1	NULL
4	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
5	Testi3	testi@testi	ManualTest1	NULL
6	Testi5	testi@testi	ManualTest3	NULL
7	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

Testi onnistui odotetun tuloksen mukaisesti.

3.1.1.5 Tietokannan triggerin manuaalinen testaus

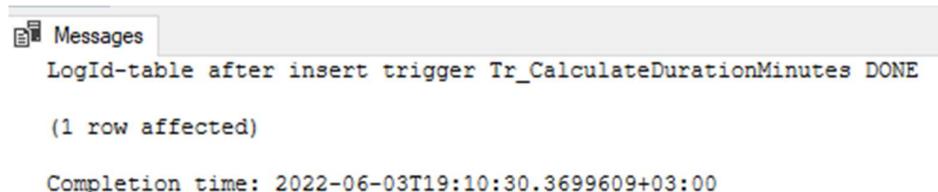
3.1.1.5.1 Syötetään PainLog-tauluun yksi rivi tietoja.

Syötetään tauluun rivi, jossa on vähintään kaikki vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserId)
values
(6, '2022-06-03 19:07:00', '2022-06-03 20:35:00', 107, 'SQL Server manual test
trigger 1', 'anna1')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen "PainLog-table after insert trigger Tr-CalculateDurationMinutes DONE" ja "1 row affected". Syötetty rivi löytyy tietokantataulusta ja kivun kesto on laskettu oikein.

Tulos: SQL Server Management Studio antoi odotetun ilmoituksen:



```
LogId-table after insert trigger Tr_CalculateDurationMinutes DONE
(1 row affected)

Completion time: 2022-06-03T19:10:30.3699609+03:00
```

Syötetty rivi löytyy tietokantataulusta (LogID 4061)

Kivun kesto on laskettu oikein: 3.6.2022 19:07 – 3.6.2022 20:35 = 60 minuuttia + 28 minuuttia = 88 minuuttia.

Testi onnistui odotusten mukaisesti. Triggerin ilmoitusta täytyisi kuitenkin korjata, koska "LogId-table" on väärä termi, kun kyseessä on PainLog-taulu. Triggerin antamaa ilmoitusta korjattu tämän johdosta.

3.1.1.5.2 Syötetään tauluun useita rivejä oikeellisia tietoja

Syötetään tauluun neljä riviä, joissa on vähintään vaaditut tiedot oikeassa tietotyypissä.

```
Insert into PainLog (PainIntensity, StartTime, EndTime, LocationID, Notes, UserId)
values
(6, '2022-06-02 10:05:00', '2022-06-03 10:35:00', 107, 'SQL Server manual test
trigger 2', 'anna1'),
(6, '2022-06-01 19:05:00', '2022-06-01 19:06:00', 108, 'SQL Server manual test
trigger 3', 'Esa1'),
(6, '2022-06-04 11:35:00', '2022-06-04 11:42:00', 105, 'SQL Server manual test
trigger 4', 'Testi1'),
(6, '2022-06-05 14:57:00', '2022-06-05 15:07:00', 108, 'SQL Server manual test
trigger 5', 'Esa1')
```

Odotettu tulos: SQL Server Management Studio antaa ilmoituksen triggerin käytöstä ja ilmoituksen "4 rows affected" ja ilmoituksen triggerin käytöstä. Syötetyt rivit löytyvät tietokantataulusta ja kivun kestot on laskettu oikein.

Tulos: SQL Server Management Studio antoi odotetun ilmoituksen:

```
Messages
PainLog-table after insert trigger Tr_CalculateDurationMinutes DONE
(4 rows affected)

Completion time: 2022-06-03T19:15:49.4688155+03:00
```

Syötetyt rivit löytyvät tietokantataulusta (LogID:t 4062-4065):

	Results	Messages									
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1
5	4062	6	2022-06-02 10:05:00.000	2022-06-03 10:35:00.000	1470	NULL	NULL	NULL	107	SQL Server manual test trigger 2	anna1
6	4063	6	2022-06-01 19:05:00.000	2022-06-01 19:06:00.000	1	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1
7	4064	6	2022-06-04 11:35:00.000	2022-06-04 11:42:00.000	7	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Testi1
8	4065	6	2022-06-05 14:57:00.000	2022-06-05 15:07:00.000	10	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1

Kestot on laskettu oikein:

4062: 2.6.2022 10:05 – 3.6.2022 10:35 = 24 h 30 minuuttia = 1470 minuuttia

4063: 1.6.2022 19:05 – 1.6.2022 19:06 = 1 minuutti

4064: 4.6.2022 11:35 – 4.6.2022 11:42 = 7 minuuttia

4065: 5.6.2022 14:57 – 5.6.2022 15:07 = 10 minuuttia

Testi onnistui odotusten mukaisesti.

3.1.1.5.3 Muokataan PainLog-taulun yhden riviin päivämääritietoa

Muokataan yhden kohdassa 3.1.1.5B syötetyn riviin alkupäivämäärä 10 minuuttia aiemmaksi.

```
Update PainLog set StartTime = '2022-06-05 14:47:00' Where logID = 4065
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen triggerin käytöstä ja ilmoituksen "1 row affected". Muokattu rivi löytyy tietokantataulusta ja kiven kesto on laskettu oikein.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```
Messages
PainLog-table after update trigger Tr_CalculateDurationMinutes DONE
PainLog-table after insert trigger Tr_CalculateDurationMinutes DONE
(1 row affected)

Completion time: 2022-06-03T19:19:37.0163415+03:00
```

Muokattu rivi löytyy tietokantataulusta (LogID 4065)

	Results	Messages									
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4057	5	2022-06-01 08:55:00.000	2022-06-01 09:45:00.000	50	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1
5	4062	6	2022-06-02 10:05:00.000	2022-06-03 10:35:00.000	1470	NULL	NULL	NULL	107	SQL Server manual test trigger 2	anna1
6	4063	6	2022-06-01 19:05:00.000	2022-06-01 19:06:00.000	1	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1
7	4064	6	2022-06-04 11:35:00.000	2022-06-04 11:42:00.000	7	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Testi1
8	4065	6	2022-06-05 14:47:00.000	2022-06-05 15:07:00.000	20	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1

Kivun kesto on laskettu oikein:

5.6.2022 14:47 – 5.6.2022 15:07 = 20 minuuttia

Testi onnistui odotusten mukaisesti.

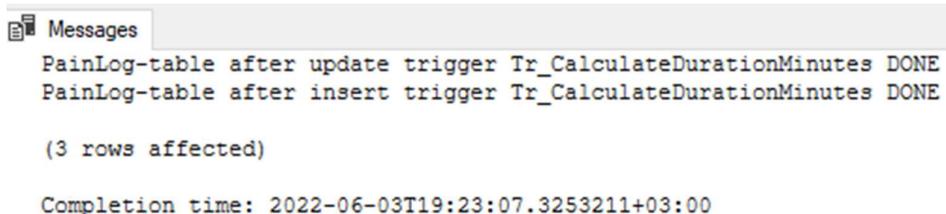
3.1.1.5.4 Muokataan PainLog-taulun useamman rivin tietoja

Muokataan niiden taulun rivien StartTime-tietoa, jossa LocationID on 108.

```
Update PainLog set StartTime = '2022-06-01 08:30:00' Where LocationID = 108
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen triggerin käytöstä ja "3 rows affected". Muokatut rivit löytyvät tietokantataulusta ja kivun kestot on laskettu oikein.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:



PainLog-table after update trigger Tr_CalculateDurationMinutes DONE
 PainLog-table after insert trigger Tr_CalculateDurationMinutes DONE
 (3 rows affected)

Completion time: 2022-06-03T19:23:07.3253211+03:00

Muokatut tiedot löytyvät taulusta (LogID:t 4057, 4063 ja 4065):

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4057	5	2022-06-01 08:30:00.000	2022-06-01 09:45:00.000	75	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1
5	4062	6	2022-06-02 10:05:00.000	2022-06-03 10:35:00.000	1470	NULL	NULL	NULL	107	SQL Server manual test trigger 2	anna1
6	4063	6	2022-06-01 08:30:00.000	2022-06-01 19:06:00.000	636	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1
7	4064	6	2022-06-04 11:35:00.000	2022-06-04 11:42:00.000	7	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Test1
8	4065	6	2022-06-01 08:30:00.000	2022-06-05 15:07:00.000	6157	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1

Kivun kestot on laskettu oikein:

4057: 1.6.2022 8:30 – 1.6.2022 9:45 = 1h 15 minuuttia = 75 minuuttia

4063: 1.6.2022 8:30 – 1.6.2022 19:06 = 10 h 36 minuuttia = 636 minuuttia

4065: 1.6.2022 8:30 – 5.6.2022 15:07 = 4 vrk 6h 37 minuuttia = 6157 minuuttia

3.1.1.5.5 Muokataan StartTime myöhäisemmäksi kuin EndTime

Muokataan yhden rivin StartTime eli alkuaika myöhäisemmäksi kuin EndTime

```
Update PainLog set StartTime = '2022-06-04 11:47:00' Where logID = 4064
```

Odottettu tulos: SQL Server Management Studio antaa ilmoituksen triggerin käytöstä ja "1 row affected". Muokatut rivit löytyvät tietokantataulusta ja kivun kesto on laskettu oikein.

Tulos: SQL Server Management Studio antoi odotuksen mukaisen ilmoituksen:

```

Messages
PainLog-table after update trigger Tr_CalculateDurationMinutes DONE
PainLog-table after insert trigger Tr_CalculateDurationMinutes DONE

(1 row affected)

Completion time: 2022-06-03T19:27:55.5387893+03:00

```

Muokattu rivi löytyy tietokantataulusta (LogID 4064):

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad
3	4057	5	2022-06-01 08:30:00.000	2022-06-01 09:45:00.000	75	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
4	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1
5	4062	6	2022-06-02 10:05:00.000	2022-06-03 10:35:00.000	1470	NULL	NULL	NULL	107	SQL Server manual test trigger 2	anna1
6	4063	6	2022-06-01 08:30:00.000	2022-06-01 19:06:00.000	636	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1
7	4064	6	2022-06-04 11:47:00.000	2022-06-04 11:42:00.000	-5	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Test1
8	4065	6	2022-06-01 08:30:00.000	2022-06-05 15:07:00.000	6157	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1

Kivun kesto on laskettu oikein:

4.6.2022 11:47 – 4.6.2022 11:42 = -5 minuuttia

Testi onnistui odotusten mukaisesti. Olisiko jatkokehitykseen mietittäväissä joko käyttöliittymään, tietokantaan tai back endiin jotain ehtoa, jonka mukaan StartTime ei saa olla myöhempää kuin EndTime.

3.1.1.6 Tietokantataulujen yhdistelyhakujen testaus

3.1.1.6.1 Haetaan PainLog-taulun rivejä, joihin yhdistetään Users-taulun tietoja

Haetaan PainLog-taulun kenttiä tietyltä käyttäjältä ja yhdistetään hakuun käyttäjää vastaavat Username ja Email -kentät Users-taulusta.

```
Select p.logID, p.PainIntensity, p.StartTime, p.Duration, p.LocationID,
u.Username, u.Email
from PainLog as p Join Users as u on p.UserId = u.Username Where p.UserId = 'Esa1'
```

Odotettu tulos: Haetut kentät löytyvät Results-osiosta

Tulos: Haku onnistui odotetun tuloksen mukaisesti ja haetut rivit löytyvät Results-osiosta

logID	PainIntensity	StartTime	Duration	LocationID	Username	Email
1	4063	6	2022-06-01 08:30:00.000	636	108	Esa1
2	4065	6	2022-06-01 08:30:00.000	6157	108	Esa1

3.1.1.6.2 Haetaan PainLog-taulun rivejä, joihin yhdistetään PainLocation-taulun tietoja

Haetaan PainLog-taulun kenttiä tietyltä käyttäjältä ja yhdistetään hakuun LocationID:tä vastaava sijainnin nimi PainLocation-taulusta:

```
Select p.logID, p.PainIntensity, p.StartTime, p.Duration, p.LocationID,
l.LocationName, p.UserId
from PainLog as p Join PainLocation as l on p.LocationID = l.LocationID Where
p.UserId = 'Esa1'
```

Odotettu tulos: Haetut rivit löytyvät Results.osiosta.

Tulos: Testi onnistui odotetun tuloksen mukaisesti ja haetut rivit löytyvät Results-osiosta.

	Results	Messages					
	logID	PainIntensity	StartTime	Duration	LocationID	LocationName	UserId
1	4063	6	2022-06-01 08:30:00.000	636	108	Niska	Esa1
2	4065	6	2022-06-01 08:30:00.000	6157	108	Niska	Esa1

3.1.1.6.3 Haetaan PainLog-taulun rivejä, joihin yhdistetään PainLocation-taulun tietoja

Haetaan PainLog-taulun kenttiä tietyltä käyttäjältä, yhdistetään hakuun sekä LocationID:tä vastaava LocationName PainLocation -taulusta sekä Username ja Email Users -taulusta.

```
Select p.logID, p.PainIntensity, p.StartTime, p.Duration, l.LocationName,
u.Username, u.Email
from PainLog as p Join PainLocation as l on p.LocationID = l.LocationID Join Users
as u on p.UserId = u.Username
Where p.UserId = 'Esa1'
```

Odottettu tulos: Haetut tiedot löytyvät Results osiosta.

Tulos: Testi onnistui odotetun tuloksen mukaisesti ja rivit löytyvät Results-osiosta:

	Results	Messages					
	logID	PainIntensity	StartTime	Duration	LocationName	Username	Email
1	4063	6	2022-06-01 08:30:00.000	636	Niska	Esa1	Esa@testi
2	4065	6	2022-06-01 08:30:00.000	6157	Niska	Esa1	Esa@testi

3.1.2 Kontrollerien manuaalinen testaus Postman-sovelluksella

Kontrollerien metodit testattiin Postman-sovelluksella suorittamalla sovelluksen kautta hakuja tietokantaan sekä lisäämällä, poistamalla ja muokkaamalla tietokannan testidataa sovelluksen vältäksellä. Yksittäiset testitapaukset on kuvattu tarkemmin testaussuunnitelmassa sekä tässä pöytäkirjassa.

Kaikki testit totetutuivat odotusten mukaan. Manuaalien testauksen perusteella ASP.NET Core API kontrollerien metodit `HttpGet` (kontrollerin metodit `GetAll()` ja `GetOne()`), `HttpPost` (kontrollerien metodi `CreateNew()`), `HttpPut` (kontrollerien metodi `Edit()`) ja `HttpDelete` (kontrollerien metodi `Delete()`) toimivat oikein. Osa kontrollereihin määritetyistä ilmoituksista oli kuitenkin testauksen perusteella epäjohdonmukaisia ja testauksen johdosta ne tarkistettiin ja korjattiin.

3.1.2.1 Note-kontrollerin manuaalinen testaus

3.1.2.1.1 Haetaan taulun kaikki tiedot

Haetaan taulun tiedot `HttpGet`-metodilla (kontrollerissa `GetAll()`-metodi).

The screenshot shows the Postman interface with a GET request to `https://localhost:5001/kipukalenteri/note`. The 'Params' tab is active, displaying a 'Query Params' table with one row: 'Key' and 'Value'. Other tabs visible include 'Headers', 'Body', 'Pre-request Script', 'Tests', and 'Settings'.

KEY	VALUE
Key	Value

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja taulun tiedot palautuvat Body-osiolla.

Tulos: Testi onnistui odotuksen mukaisesti, status on 200 Ok ja taulun tiedot palautuvat Body-osiolla:



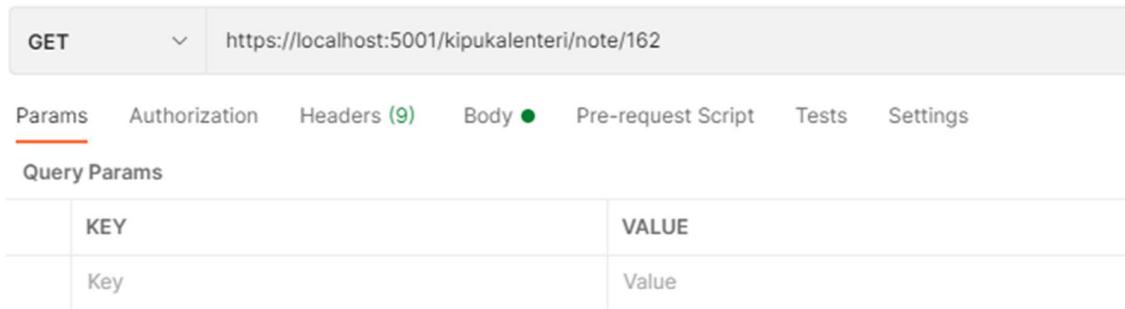
```

1 [
2   {
3     "noteId": 165,
4     "noteText": "SQL Server DB manual test 3.1.1.1B",
5     "noteDate": "2022-06-03T13:57:00",
6     "userId": "testiad",
7     "user": null
8   },
9   {
10    "noteId": 163,
11    "noteText": "SQL Server DB manual test 3.1.1.1B",
12    "noteDate": "2022-06-03T11:57:00",
13    "userId": "testiad",
14    "user": null
15  },
16  {
17    "noteId": 162,
18  }
]

```

3.1.2.1.2 Haetaan yksi rivi tietoa

Haetaan taulusta yksi rivi tietoa `HttpGet`-metodilla (kontrollerissa `GetOne()`-metodi).



KEY	VALUE
Key	Value

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja rivin tiedot näkyvät Body-osiolla.

Tulos: Testi onnistui odotetun tuloksen mukaisesti, rivin tiedot palautuvat Body-osiolla ja status on 200 OK



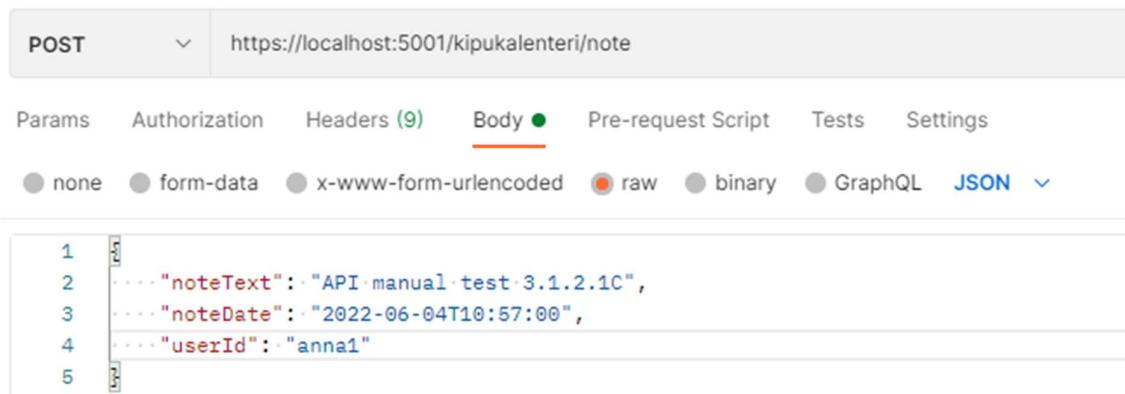
```

1 {
2   "noteId": 162,
3   "noteText": "SQL Server DB manual test 3.1.1.1E",
4   "noteDate": "2022-06-03T10:57:00",
5   "userId": "testiad",
6   "user": null
7 }

```

3.1.2.1.3 Lisätään yksi rivi tietoa

Lisätään tietoa `HttpPost`-metodilla (kontrollerissa `CreateNew()`-metodi)



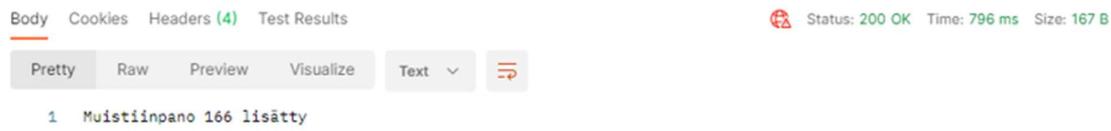
POST https://localhost:5001/kipukalenteri/note

Body (JSON)

```
1 {  
2   "noteText": "API manual test 3.1.2.1C",  
3   "noteDate": "2022-06-04T10:57:00",  
4   "userId": "anna1"  
5 }
```

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiassa on ilmoitus "Muistiinpano X lisätty".

Tulos: Postman antoi odotuksen mukaisen ilmoituksen ja Body-osiassa palautui ilmoitus "Muistiinpano 166 lisätty".



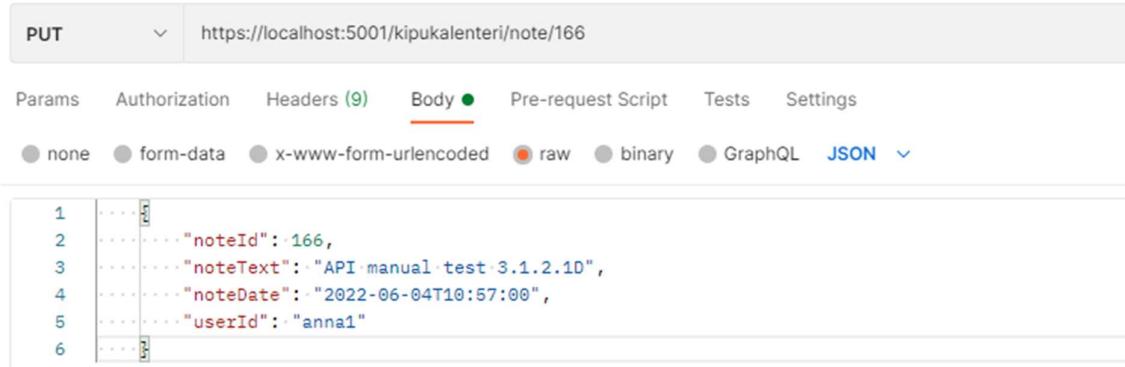
Status: 200 OK Time: 796 ms Size: 167 B

Pretty Raw Preview Visualize Text

```
1 Muistiinpano 166 lisätty
```

3.1.2.1.4 Muokataan tietokantaan syötettyä tietoa

Muokataan kohdassa 3.1.2.1A syötettyä rivää lisäämällä NoteText-kenttään teksti "Controller manual test via Postman 3.1.2.1D"



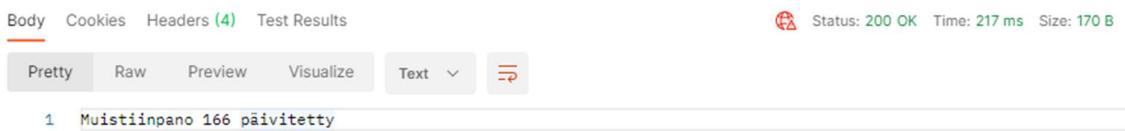
PUT https://localhost:5001/kipukalenteri/note/166

Body (JSON)

```
1 {  
2   "noteId": 166,  
3   "noteText": "API manual test 3.1.2.1D",  
4   "noteDate": "2022-06-04T10:57:00",  
5   "userId": "anna1"  
6 }
```

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiassa palautuu ilmoitus "Muistiinpano XXX päivitetty". Päivitetty tieto löytyy kannasta HttpGet-metodilla.

Tulos: Testi onnistui odotetun tuloksen mukaisesti. Status on 200 OK ja Body-osiassa palautui ilmoitus muistiinpanon päivittämisestä:

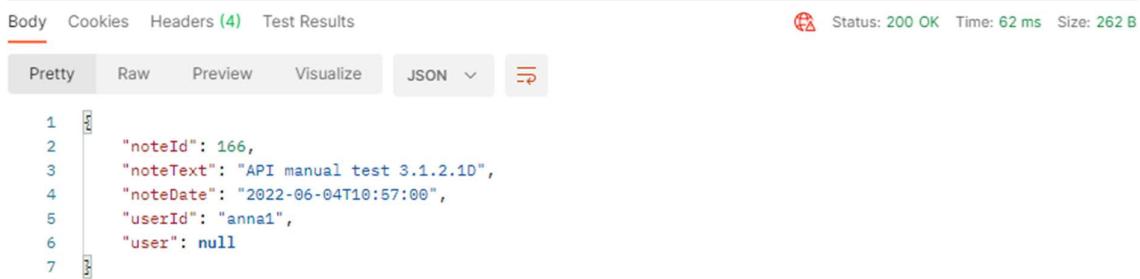


Status: 200 OK Time: 217 ms Size: 170 B

Pretty Raw Preview Visualize Text

```
1 Muistiinpano 166 päivitetty
```

Päivitetty tieto löytyi tietokannasta:

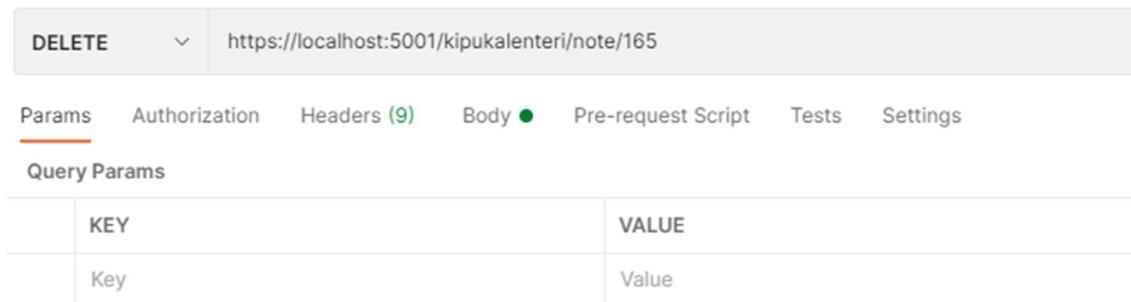


```

1   "noteId": 166,
2   "noteText": "API manual test 3.1.2.1D",
3   "noteDate": "2022-06-04T10:57:00",
4   "userId": "anna1",
5   "user": null
6
7
  
```

3.1.2.1.5 Poistetaan tietokantaan syötettyä tietoa

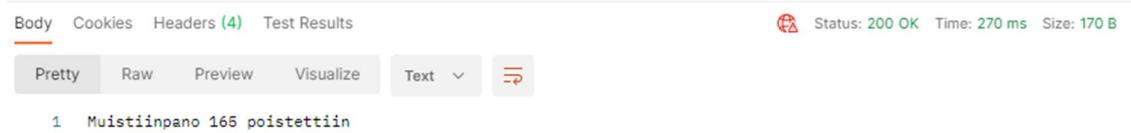
Poistetaan yksi kohdassa 3.1.1.1B syötetyistä riveistä tietokannasta.



KEY	VALUE
Key	Value

Odottu tulos: Postman antaa ilmoituksen ”Status: 200 Ok” ja Body-osiolla palautuu ilmoitus ”Muistiinpano 165 poistettiin”. Poistettua riviä ei löydy HttpGet-metodilla.

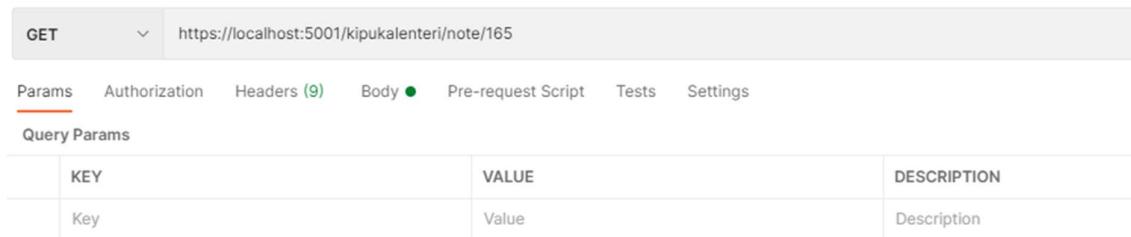
Tulos: Testi onnistui odotetun tuloksen mukaisesti. Status on 200 OK ja Body-osiolla palautui ilmoitus muistiinpanon poistamisesta:



```

1 Muistiinpano 165 poistettiin
  
```

Poistettua riviä ei löydy HttpGet-metodilla:



KEY	VALUE	DESCRIPTION
Key	Value	Description

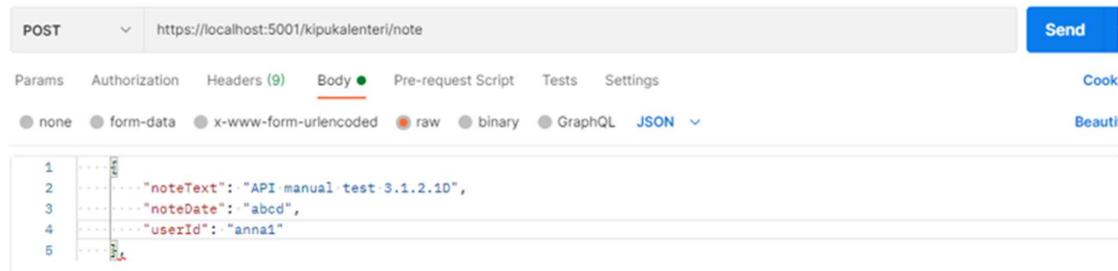


```

1
  
```

3.1.2.1.6 Syötetään rivi, joka sisältää väärän tyyppistä tietoa

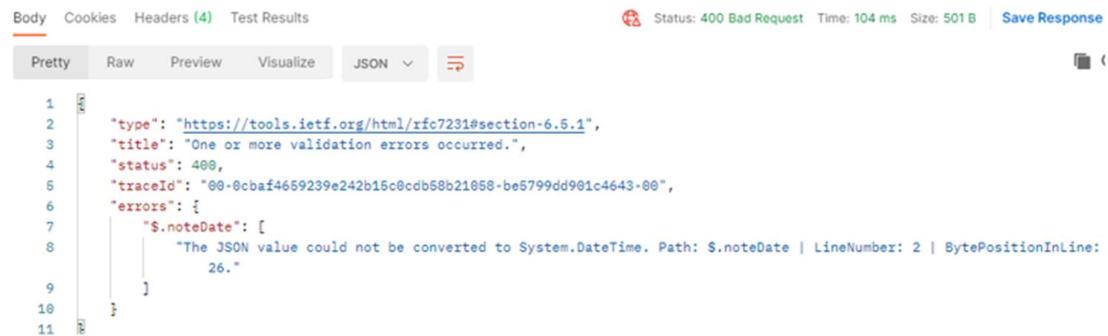
Syötetään HttpPost-metodilla rivi, jossa noteDate-kenttään yritetään syöttää testitietoa.



```
POST https://localhost:5001/kipukalenteri/note
Body (JSON)
{
  "noteText": "API manual test 3.1.2.10",
  "noteDate": "abcd",
  "userId": "anna1"
}
```

Odotettu tulos: Tiedon syöttäminen ei onnistu, Postman antaa statuksen 400 Bad Request.

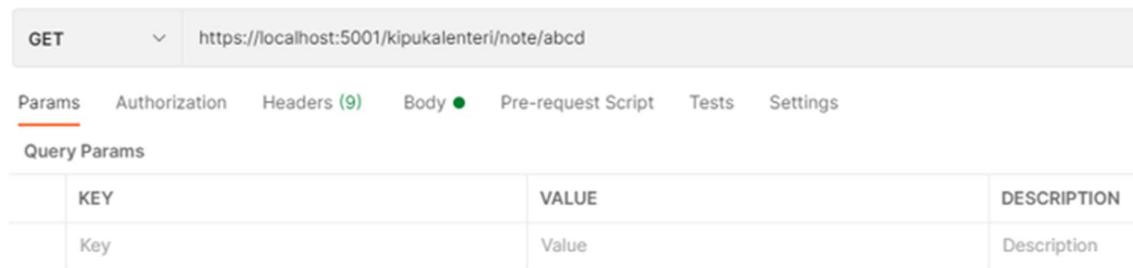
Tulos: Testi toteutui odotetun tuloksen mukaisesti, syöttö ei onnistunut ja status on 400 Bad Request.



```
Status: 400 Bad Request Time: 104 ms Size: 501 B Save Response
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-0cbaf4659239e242b15c0cdb58b21068-be5799dd901c4643-00",
  "errors": [
    {
      "$.noteDate": [
        "The JSON value could not be converted to System.DateTime. Path: $.noteDate | LineNumber: 2 | BytePositionInLine: 26."
      ]
    }
  ]
}
```

3.1.2.1.7 Haetaan rivi tietoa väärän tyyppisellä id:llä

Haetaan HttpGet-metodilla tietokantataulusta rivi id:llä, joka on tekstityyppinen.



```
GET https://localhost:5001/kipukalenteri/note/abcd
Query Params
  KEY          VALUE          DESCRIPTION
  Key          Value          Description
```

Odotettu tulos: Haun status on 400 Bad Request ja Postman antaa ilmoituksen, että hakuarvo ei ole kelpaava.

Tulos: Haku toteutui odotetun tuloksen mukaisesti. Haun status on 400 Bad Request ja Postman ilmoitti, että hakuarvo ei kelpaa.

The screenshot shows a Postman request for a DELETE operation on the URL `https://localhost:5001/kipukalenteri/7000`. The response status is 400 Bad Request, with a size of 411 B. The response body is a JSON object:

```
1
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3   "title": "One or more validation errors occurred.",
4   "status": 400,
5   "traceId": "00-4d1514d2265fffb4a931529ce9d38344f-a7569cf244a7ce46-00",
6   "errors": [
7     "noteid": [
8       "The value 'abcd' is not valid."
9     ]
10   }
```

3.1.2.1.8 Poistetaan rivi, jota ei ole olemassa

Poistetaan rivi id:llä jota ei löydy tietokantataulusta.

The screenshot shows a Postman request for a GET operation on the URL `https://localhost:5001/kipukalenteri/painlocation`. The response status is 404 Not Found, with a size of 99 B. The response body is a simple JSON object:

```
1
```

3.1.2.2 PainLocation-kontrollerin manuaalinen testaus

3.1.2.2.1 Haetaan taulun kaikki tiedot

Haetaan taulun tiedot `HttpGet`-metodilla (kontrollerissa `GetAll()`-metodi).

The screenshot shows a Postman request for a GET operation on the URL `https://localhost:5001/kipukalenteri/painlocation`. The response status is 200 OK, with a size of 100 B. The response body is a JSON object:

```
1
2   "id": 1,
3   "name": "PainLocation 1",
4   "description": "Tämä on ensimmäinen painlokalitettu taulu."
```



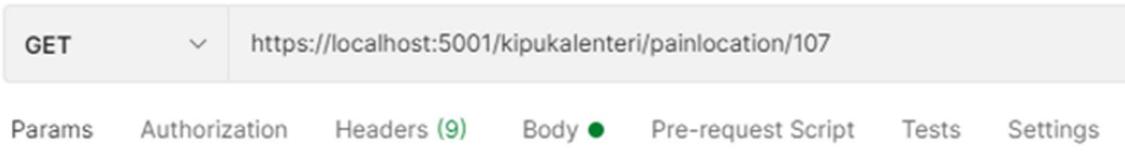
```

1 [
2   {
3     "locationId": 104,
4     "locationName": "Vatsa",
5     "painLogs": []
6   },
7   {
8     "locationId": 105,
9     "locationName": "Pää",
10    "painLogs": []
11  },
12  {
13    "locationId": 107,
14  }

```

3.1.2.2.2 Haetaan yksi rivi tietoa

Haetaan taulusta yksi rivi tietoa `HttpGet`-metodilla (kontrollerissa `GetOne()`-metodi).

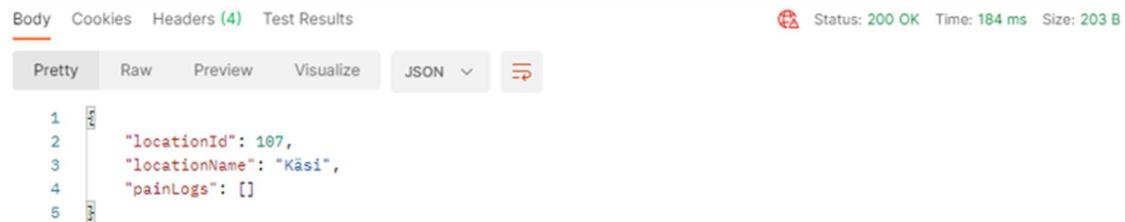


GET https://localhost:5001/kipukalenteri/painlocation/107

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja rivin tiedot näkyvät Body-osiolla.

Tulos: Testi toteutui odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja haetun rivin tiedot palautuvat Body-osiolla.



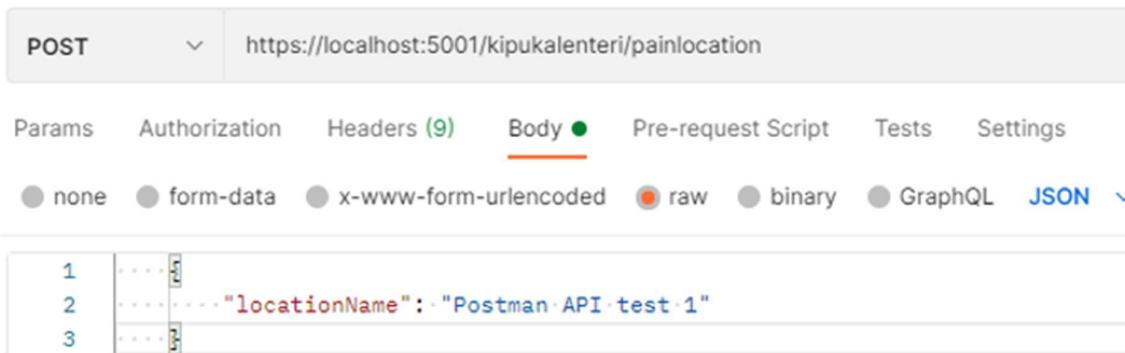
```

1 {
2   "locationId": 107,
3   "locationName": "Käsi",
4   "painLogs": []
5 }

```

3.1.2.2.3 Lisätään yksi rivi tietoa

Lisätään tietoa `HttpPost`-metodilla (kontrollerissa `CreateNew()`-metodi).



POST https://localhost:5001/kipukalenteri/painlocation

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```

1 {
2   "locationName": "Postman API test 1"
3 }

```

Odotettu tulos: Postman antoi ilmoituksen "Status: 200 Ok" ja Body-osiolla ilmoituksessa palautuu lisätyn sijainnin nimi. Lisättyt tiedot löytyvät `HttpGet`-metodilla tietokannasta.

Tulos: Testi onnistuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiossa palautuu ilmoitus sijainnin lisäyksestä.

The screenshot shows the Postman interface with the 'Test Results' tab selected. The status bar indicates 'Status: 200 OK Time: 363 ms Size: 178 B'. The body of the response is displayed in pretty JSON format:

```
1 Sijainti Postman API test 1 lisätty
```

```
1
2   {
3     "locationId": 3161,
4     "locationName": "SQL Server DB manual test 4",
5     "painLogs": []
6   },
7   {
8     "locationId": 3162,
9     "locationName": "Postman API test 1",
10    "painLogs": []
11  }
12 ]
```

Lisätty rivi löytyy tietokannasta:

```
1
2   {
3     "locationId": 3161,
4     "locationName": "SQL Server DB manual test 4",
5     "painLogs": []
6   },
7   {
8     "locationId": 3162,
9     "locationName": "Postman API test 1",
10    "painLogs": []
11  }
12 ]
```

3.1.2.2.4 Muokataan tietokantaan syötettyä tietoa

Muokataan aiemmin syötettyä rivää lisäämällä LocationName-kenttään teksti "Postman API test 2"

The screenshot shows a POST request to <https://localhost:5001/kipukalenteri/painlocation/3162>. The 'Body' tab is selected, showing the JSON payload:

```
1
2   {
3     "locationId": 3162,
4     "locationName": "Postman API test 2"
5   }
```

Odottettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiossa palautuu ilmoitus sijainnin päivittämisestä. Muokattu rivi löytyy HttpGet-metodilla.

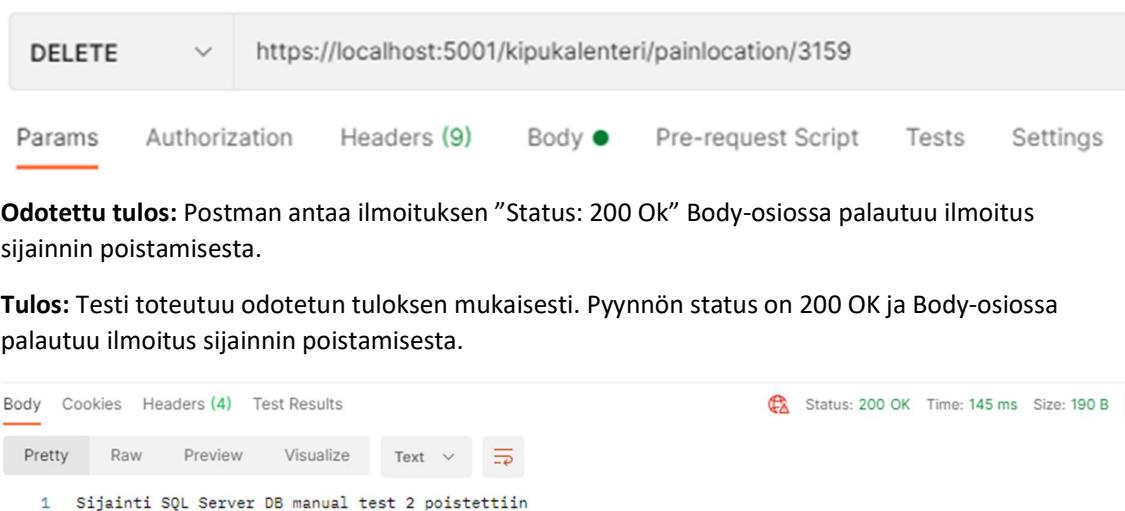
Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiossa palautuu ilmoitus sijainnin päivittämisestä.

The screenshot shows the Postman interface with the 'Test Results' tab selected. The status bar indicates 'Status: 200 OK Time: 388 ms Size: 181 B'. The body of the response is displayed in pretty JSON format:

```
1 Sijainti Postman API test 2 päivitetty
```

3.1.2.2.5 Poistetaan tietokantaan syötettyä tietoa

Poistetaan yksi aiemmin syötetyistä riveistä tietokannasta.



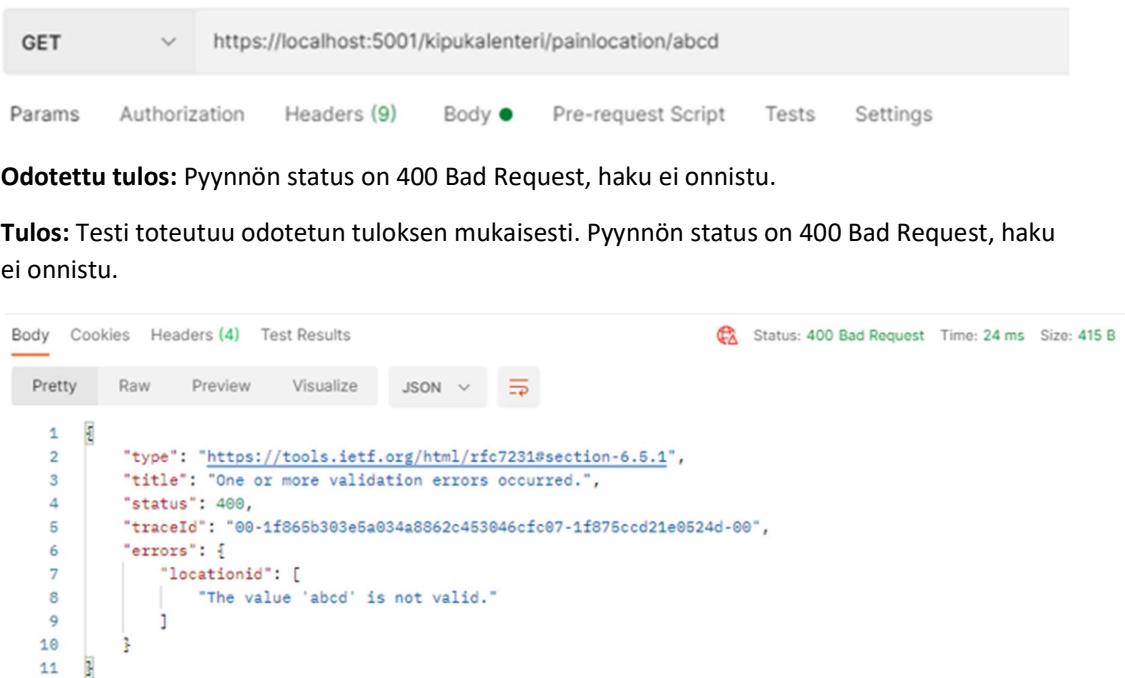
The screenshot shows a Postman DELETE request. The URL is <https://localhost:5001/kipukalenteri/painlocation/3159>. The Headers tab shows 9 items. The Body tab is selected and contains the JSON response:

```
1 Sijainti SQL Server DB manual test 2 poistettiin
```

The status bar indicates Status: 200 OK, Time: 145 ms, and Size: 190 B.

3.1.2.2.6 Haetaan tietoa vääräntyyppisellä id:llä

Haetaan tietoa tekstityyppisellä id:llä `HttpGet`-metodilla.



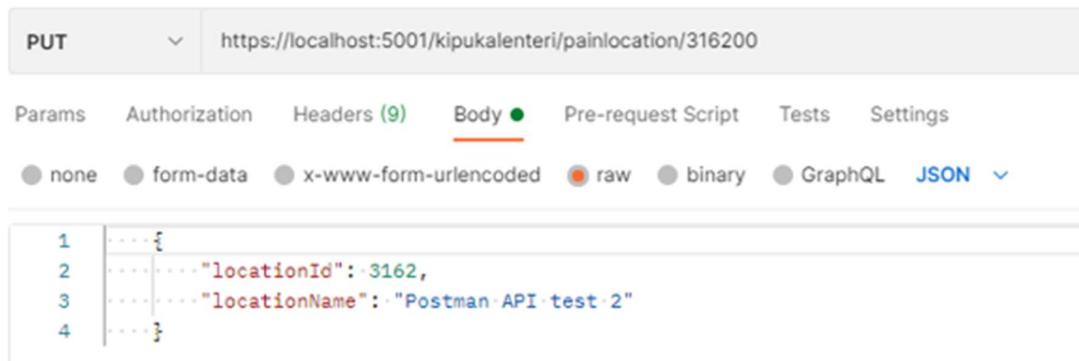
The screenshot shows a Postman GET request. The URL is <https://localhost:5001/kipukalenteri/painlocation/abcd>. The Headers tab shows 9 items. The Body tab is selected and contains the JSON response:

```
1
2 "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3 "title": "One or more validation errors occurred.",
4 "status": 400,
5 "traceId": "00-1f865b303e5a034a8862c453046cf07-1f875cccd21e0524d-00",
6 "errors": [
7     "locationid": [
8         "The value 'abcd' is not valid."
9     ]
10 ]
11 }
```

The status bar indicates Status: 400 Bad Request, Time: 24 ms, and Size: 415 B.

3.1.2.2.7 Muokataan riviä, jota ei ole olemassa

Laitetaan muokauspyyntö `HttpPut`-metodilla `locationId`:lle, jota ei ole tietokantataulussa.



PUT https://localhost:5001/kipukalenteri/painlocation/316200

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "locationId": 3162,  
3   "locationName": "Postman API test 2"  
4 }
```

Odotettu tulos: Pyynnön status on 404 Not Found ja Body-osiolla palautuu ilmoitus, että päivitettävää sijaintia ei löytynyt.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 404 Not Found ja Body-osiolla palautuu ilmoitus, että päivitettävää sijaintia ei löytynyt. Ilmoituksesta puuttuu välilyönti, joten sitä täytyy korjata REST-API -sovelluksessa.



Body Cookies Headers (4) Test Results

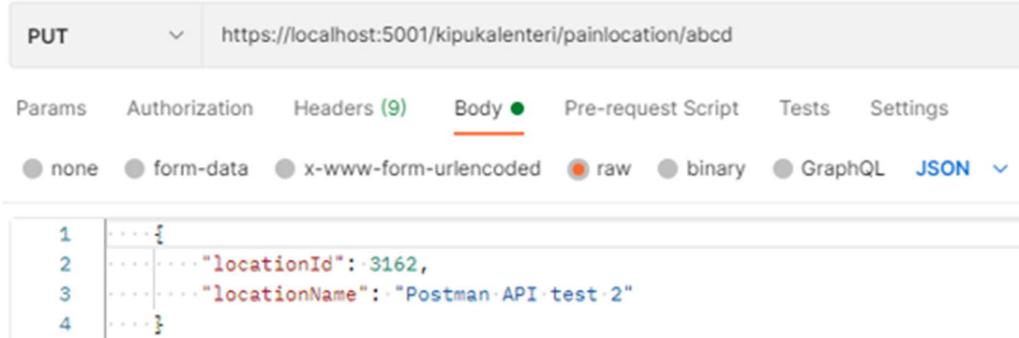
Pretty Raw Preview Visualize Text

Status: 404 Not Found Time: 34 ms Size: 195 B

```
1 Päivitettävää sijaintia316200 ei löytynyt
```

3.1.2.2.8 Muokataan riviä, jolle annetaan vääräntyyppinen id

Laitetaan muokkauspyyntö HttpPut-metodilla locationId:lle, joka on tekstityyppinen.



PUT https://localhost:5001/kipukalenteri/painlocation/abcd

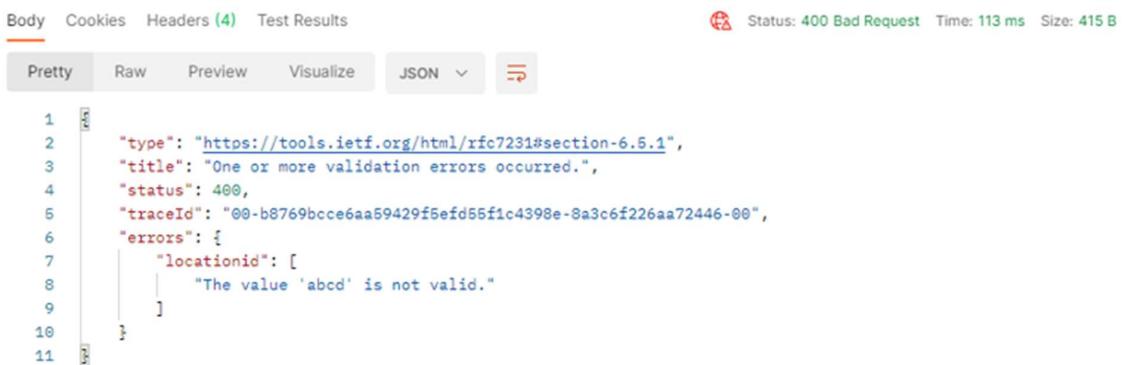
Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "locationId": 3162,  
3   "locationName": "Postman API test 2"  
4 }
```

Odotettu tulos Pyynnön status on 400 Bad Request, muokkaus ei onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 400 Bad Request, muokkaus ei onnistu.



The screenshot shows a Postman test result for a DELETE request. The status is 400 Bad Request. The response body is a JSON object with the following content:

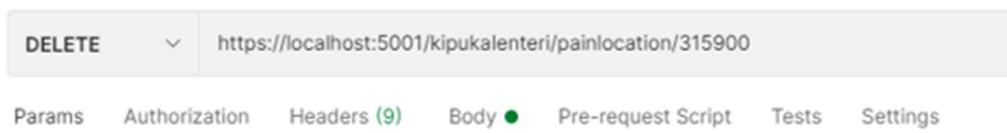
```

1   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
2   "title": "One or more validation errors occurred.",
3   "status": 400,
4   "traceId": "00-b8769bcce6aa59429f5efd55f1c4398e-8a3c6f226aa72446-00",
5   "errors": [
6     "locationid": [
7       "The value 'abcd' is not valid."
8     ]
9   ]
10
11

```

3.1.2.2.9 Poistetaan rivi, jota ei löydy tietokantataulusta

Laitetaan HttpDelete -poistopyyntö id:lle, jota ei ole tietokantataulussa.



The screenshot shows the Postman settings for a DELETE request. The URL is `https://localhost:5001/kipukalenteri/painlocation/315900`. The Headers tab shows 9 items. The Body tab is selected and contains the following JSON:

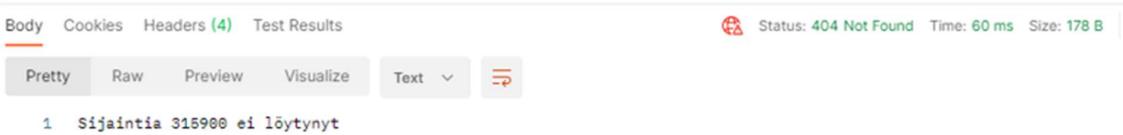
```

DELETE https://localhost:5001/kipukalenteri/painlocation/315900
{
}

```

Odottettu tulos: Pyynnön status on 404 Not Found ja Body-osiassa palautuu ilmoitus siitä, että sijaintia ei löydy. Poisto ei onnistu.

Tulos: Testi onnistui odotetun tuloksen mukaisesti. Pyynnön status on 404 Not Found ja Body-osiassa palautuu ilmoitus siitä, että sijaintia ei löydy, eikä poisto näin ollen onnistu.



The screenshot shows a Postman test result for a DELETE request. The status is 404 Not Found. The response body is a JSON object with the following content:

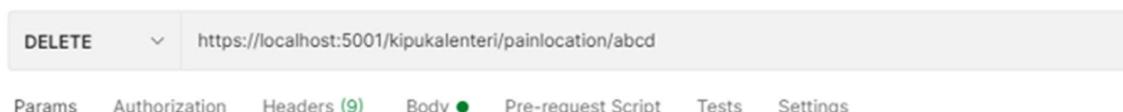
```

Body Cookies Headers (4) Test Results
Pretty Raw Preview Visualize Text
1 Sijaintia 315900 ei löytynyt

```

3.1.2.2.10 Poistetaan rivi, jolle annetaan väärän tyypin id

Lähetetään HttpDelete-pyyntö id:llä, joka on tekstityyppinen.



The screenshot shows the Postman settings for a DELETE request. The URL is `https://localhost:5001/kipukalenteri/painlocation/abcd`. The Headers tab shows 9 items. The Body tab is selected and contains the following JSON:

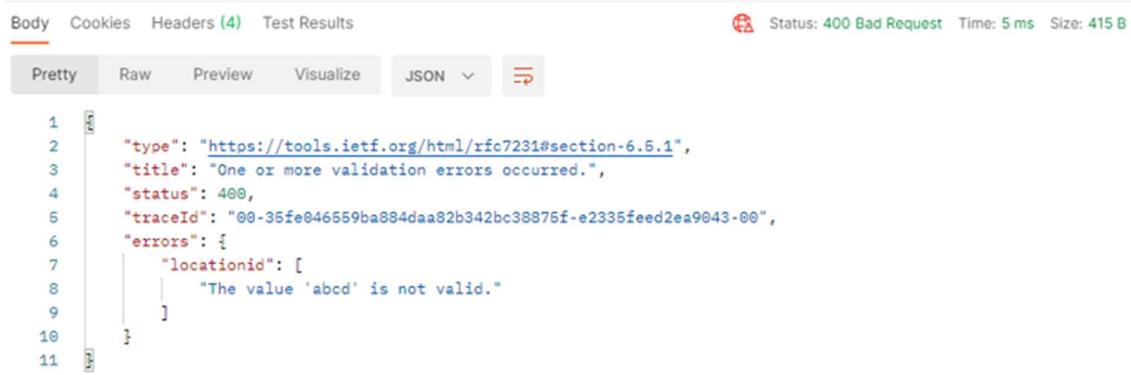
```

DELETE https://localhost:5001/kipukalenteri/painlocation/abcd
{
}

```

Odottettu tulos: Pyynnön status on 400 Bad Request, eikä poisto onnistu.

Tulos: Testi onnistuu odotetun tuloksen mukaisesti. Pyynnön status on 400 Bad Request, eikä poisto onnistu.



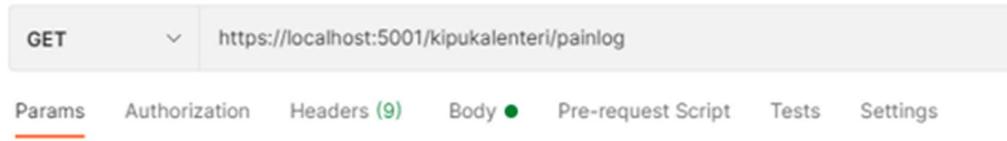
The screenshot shows a Postman test results page for a failed request. The status is 400 Bad Request, time is 5 ms, and size is 415 B. The response body is a JSON object:

```
1
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3   "title": "One or more validation errors occurred.",
4   "status": 400,
5   "traceId": "00-35fe046559ba884daa82b342bc38875f-e2335feed2ea9043-00",
6   "errors": [
7     "locationid": [
8       "The value 'abcd' is not valid."
9     ]
10  ]
11 }
```

3.1.2.3 PainLog-kontrollerin manuaalinen testaus

3.1.2.3.1 Haetaan taulun kaikki tiedot

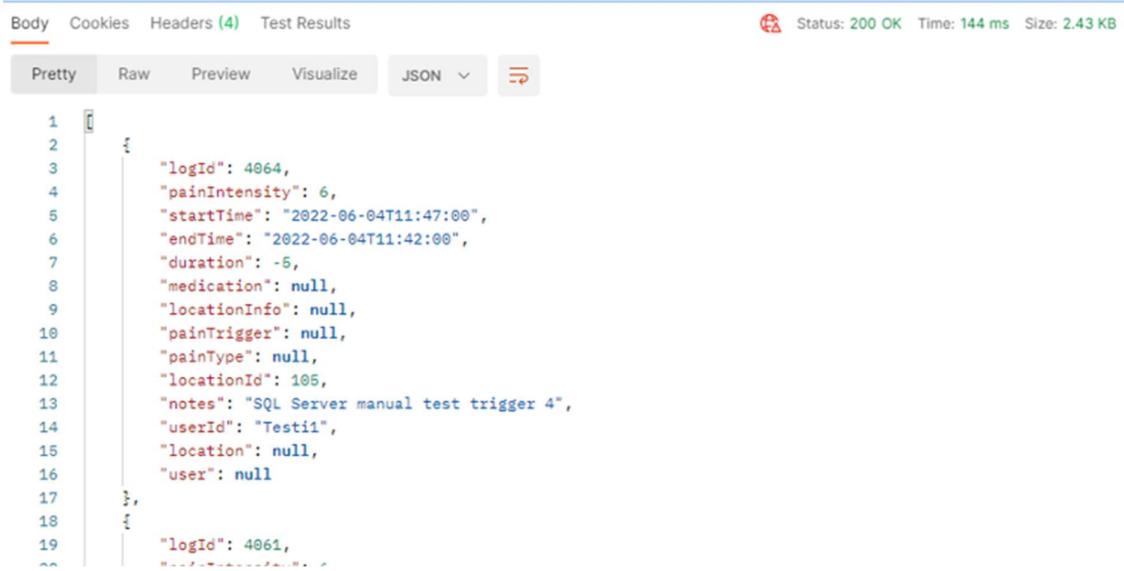
Haetaan taulun tiedot `HttpGet`-metodilla (kontrollerissa `GetAll()`-metodi).



The screenshot shows a Postman request configuration for a GET request to `https://localhost:5001/kipukalenteri/painlog`. The method is set to GET, and the URL is specified. The Headers tab shows 9 items, and the Body tab is selected.

Odottettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja taulun tiedot palautuvat Body-osiolla.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, pyynnön status on 200 OK ja taulun tiedot palautuvat Body-osiolla:



The screenshot shows a Postman test results page for a successful request. The status is 200 OK, time is 144 ms, and size is 2.43 KB. The response body is a JSON array of pain logs:

```
1
2 [
3   {
4     "logId": 4064,
5     "painIntensity": 6,
6     "startTime": "2022-06-04T11:47:00",
7     "endTime": "2022-06-04T11:42:00",
8     "duration": -5,
9     "medication": null,
10    "locationInfo": null,
11    "painTrigger": null,
12    "painType": null,
13    "locationId": 105,
14    "notes": "SQL Server manual test trigger 4",
15    "userId": "Testii",
16    "location": null,
17    "user": null
18  },
19  {
20    "logId": 4061,
```

3.1.2.3.2 Haetaan yksi rivi tietoa

Haetaan taulusta yksi rivi tietoa `HttpGet`-metodilla (kontrollerissa `GetOne()`-metodi).

GET <https://localhost:5001/kipukalenteri/painlog/4061>

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja rivin tiedot näkyvät Body-osiolla.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja haetun rivin tiedot palautuvat Body-osiolla.

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 133 ms Size: 438 B

Pretty Raw Preview Visualize JSON ↻

```

1
2   "logId": 4061,
3   "painIntensity": 6,
4   "startTime": "2022-06-03T19:07:00",
5   "endTime": "2022-06-03T20:35:00",
6   "duration": 88,
```

3.1.2.3.3 Lisätään yksi rivi tietoa

Lisätään tietoa HttpPost-metodilla (kontrollerissa CreateNew()-metodi).

POST <https://localhost:5001/kipukalenteri/painlog>

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON ↻

```

1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8   ...
9   ...
10  ...
11  ...
12  ...
```

Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiolla ilmoituksessa palautuu lisätyn sijainnin nimi. Lisätty tiedot löytyvät HttpGet-metodilla tietokannasta.

Tulos: Testi onnistuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiolla palautuu ilmoitus sijainnin lisäyksestä.

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 518 ms Size: 165 B

Pretty Raw Preview Visualize Text ↻

```

1 Merkintä 4066 lisätty
```

Lisätty rivi löytyy tietokannasta:



The screenshot shows a Postman API response. At the top, there are tabs for Body, Cookies, Headers (4), and Test Results. The Headers tab is selected, showing a count of 4. Below the tabs, there are buttons for Pretty, Raw, Preview, Visualize, and JSON (which is currently selected). The JSON response body is displayed as follows:

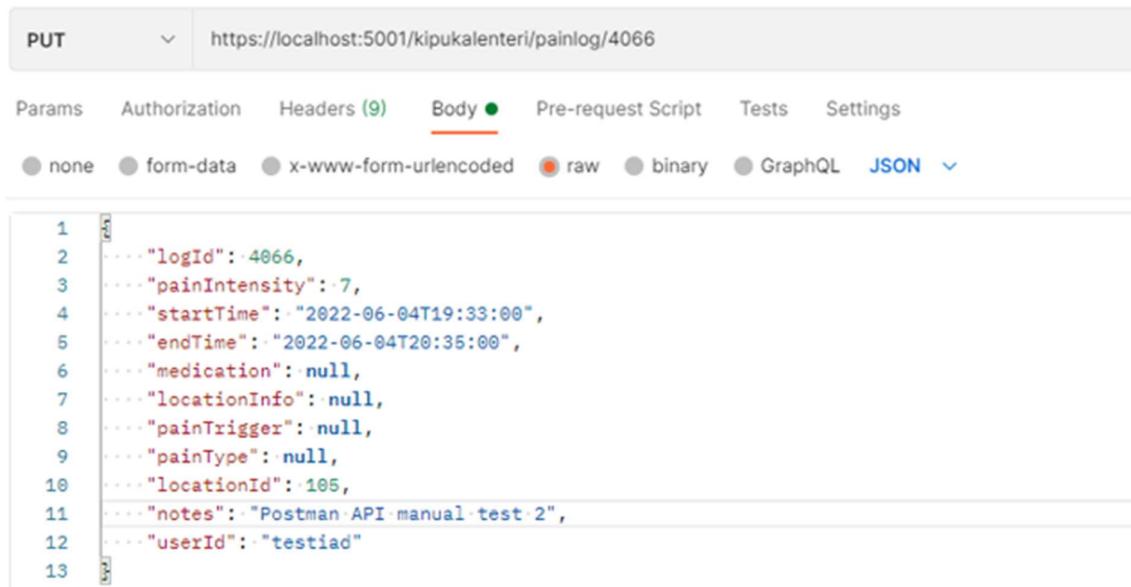
```

1
2   "logId": 4066,
3   "painIntensity": 5,
4   "startTime": "2022-06-04T19:33:00",
5   "endTime": "2022-06-04T20:35:00",
6   "duration": 62,
7   "medication": null,
8   "locationInfo": null,
9   "painTrigger": null,
10  "painType": null,
11  "locationId": 105,
12  "notes": "Postman API manual test 1",
13  "userId": "testiad",
14  "location": null,
15  "user": null
16

```

3.1.2.3.4 Muokataan tietokantaan syötettyä tietoa

Muokataan aiemmin syötettyä rivää lisäämällä notes-kenttään teksti "Postman API manual test 2"



The screenshot shows a Postman PUT request. The method is set to PUT, and the URL is https://localhost:5001/kipukalenteri/painlog/4066. The Headers tab is selected, showing a count of 9. The Body tab is selected, showing the raw JSON data for the update:

```

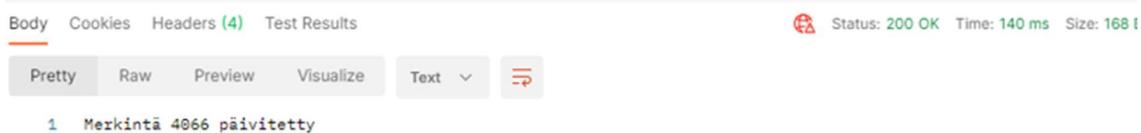
1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8   ...
9   ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
16  ...

```

The JSON data includes the original log entry fields and the new note field: "notes": "Postman API manual test 2".

Odottettu tulos: Postman antaa ilmoitukseen "Status: 200 Ok" ja Body-osiolla palautuu ilmoitus merkinnän päivittämisestä. Muokattu rivi löytyy HttpGet-metodilla.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiolla palautuu ilmoitus sijainnin päivittämisestä.



The screenshot shows a Postman API response. At the top, there are tabs for Body, Cookies, Headers (4), and Test Results. The Headers tab is selected, showing a count of 4. Below the tabs, there are buttons for Pretty, Raw, Preview, Visualize, and Text (which is currently selected). The Text response body is displayed as follows:

```

1 Merkintä 4066 päivitetty

```

Päivitetty tiedot löytyvät tietokannasta:

```

1
2   "logId": 4066,
3   "painIntensity": 7,
4   "startTime": "2022-06-04T19:33:00",
5   "endTime": "2022-06-04T20:35:00",
6   "duration": 62,
7   "medication": null,
8   "locationInfo": null,
9   "painTrigger": null,
10  "painType": null,
11  "locationId": 105,
12  "notes": "Postman API manual test 2",
13  "userId": "testiad",
14  "location": null,
15  "user": null
16

```

3.1.2.3.5 Poistetaan tietokantaan syötettyä tietoa

Poistetaan yksi aiemmin syötetyistä riveistä tietokannasta.

DELETE <https://localhost:5001/kipukalenteri/painlog/4062>

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

Odottettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" Body-osiossa palautuu ilmoitus sijainnin poistamisesta. Poistettua riviä ei löydy tietokannasta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiossa palautuu ilmoitus sijainnin poistamisesta.

```

1 Merkintä 4062 poistettiin

```

Poistettua riviä ei löydy tietokannasta:

GET <https://localhost:5001/kipukalenteri/painlog/4062>

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (3) Test Results

Status: 204 No Content Time: 47 ms Size: 100 B

Pretty Raw Preview Visualize Text

1

3.1.2.3.6 Haetaan tietoa vääräntyyppisellä id:llä

Haetaan tietoa tekstityyppisellä id:llä `HttpGet`-metodilla.

The screenshot shows a `GET` request to `https://localhost:5001/kipukalenteri/painlog/abcd`. The `Headers` tab shows 9 items. The `Body` tab is selected and contains a JSON response:

```
1
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3   "title": "One or more validation errors occurred.",
4   "status": 400,
5   "traceId": "00-580e7c1fe57bc945a3fc7b5dfa556c91-49b4355b4183c24d-00",
6   "errors": [
7     {
8       "painlogid": [
9         "The value 'abcd' is not valid."
10      ]
11    }
]
```

The status bar indicates `Status: 400 Bad Request Time: 48 ms Size: 414 B`.

3.1.2.3.7 Haetaan riviä, jota ei ole olemassa

Haetaan `HttpGet`-metodilla riviä id:llä, jota ei ole tietokantataulussa.

The screenshot shows a `GET` request to `https://localhost:5001/kipukalenteri/painlog/406133`. The `Headers` tab shows 3 items. The `Body` tab is selected and contains a JSON response:

```
1
```

The status bar indicates `Status: 204 No Content Time: 35 ms Size: 100 B`.

Odotettu tulos: Pyynnön status on 204 No Content eikä haku onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti.

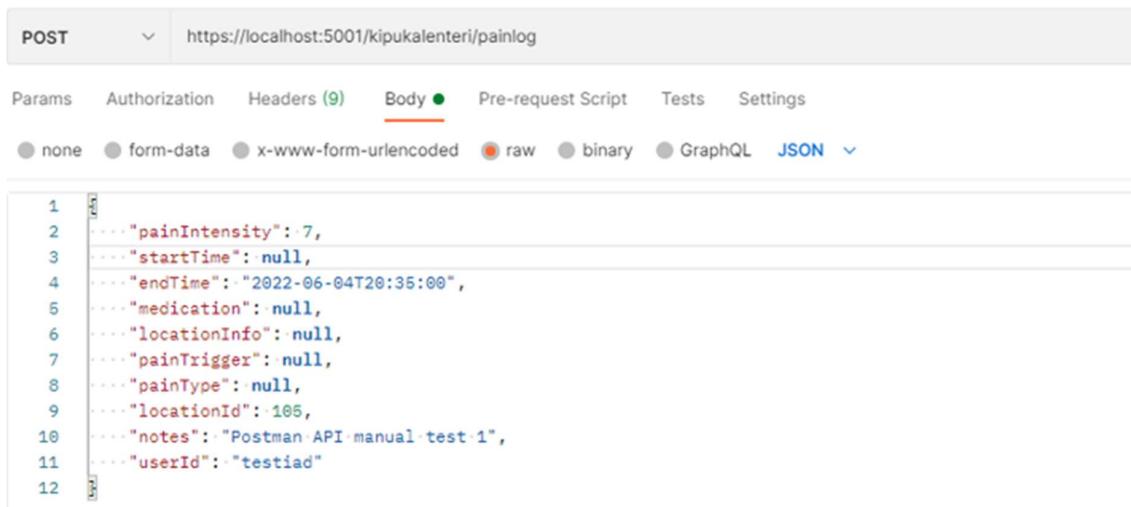
The screenshot shows a `GET` request to `https://localhost:5001/kipukalenteri/painlog/406133`. The `Headers` tab shows 3 items. The `Body` tab is selected and contains a JSON response:

```
1
```

The status bar indicates `Status: 204 No Content Time: 35 ms Size: 100 B`.

3.1.2.3.8 Syötetään rivi, josta puuttuu vaadittavia tietoja

Lisätään rivi, josta puuttuu tietokannassa pakolliseksi määritelty `StartTime`.



```
1
2     "logId": 4066,
3     "painIntensity": 7,
4     "startTime": null,
5     "endTime": "2022-06-04T20:35:00",
6     "medication": null,
7     "locationInfo": null,
8     "painTrigger": null,
9     "painType": null,
10    "locationId": 105,
11    "notes": "Postman API manual test 1",
12    "userId": "testiad"
```

Odotettu tulos: Pyynnön status on 400 Bad Request ja Body-osiolla palautuu ilmotelause "Jokin meni pieleen merkintää lisättäessä". Lisäys ei onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, lisäys ei onnistu.



Status: 400 Bad Request Time: 216 ms Size: 196 B

```
1 Jokin meni pieleen merkintää lisättäessä
```

3.1.2.3.9 Lähetetään muokkaus-pyynnössä väärän tyypistä tietoa

Laitetaan HttpPut-metodilla muokkauspyyntö, jossa on startTime-kentässä tekstityyppistä tietoa.



```
1
2     "logId": 4066,
3     "painIntensity": 7,
4     "startTime": "abcd",
5     "endTime": "2022-06-04T20:35:00",
6     "medication": null,
7     "locationInfo": null,
8     "painTrigger": null,
9     "painType": null,
10    "locationId": 105,
11    "notes": "Postman API manual test 2",
12    "userId": "testiad"
```

Odotettu tulos: Pyynnön status on 400 Bad Request, muokkaus ei onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti.

The screenshot shows a Postman interface with the following details:

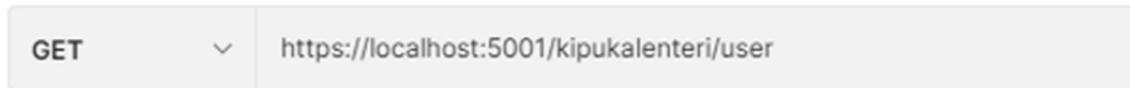
- Header tab: Headers (4)
- Status bar: Status: 400 Bad Request Time: 23 ms Size: 522 B Save Response
- Body tab: Selected
- Raw tab: Available
- Preview tab: Available
- Visualize tab: Available
- JSON tab: Available
- Content:

```
1  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
2  "title": "One or more validation errors occurred.",
3  "status": 400,
4  "traceId": "00-b0e690ae39b4534f86650c7172fe43dd-9d46a59aca39ed42-00",
5  "errors": [
6    {
7      "$.startTime": [
8        "The JSON value could not be converted to System.Nullable`1[System.DateTime]. Path: $.startTime | LineNumber: 3 |
9        BytePositionInLine: 24."
10     ]
11   }
]
```

3.1.2.4 UserControllerin manuaalinen testaus

3.1.2.4.1 Haetaan taulun kaikki tiedot

Haetaan taulun tiedot `HttpGet`-metodilla (kontrollerissa `GetAll()`-metodi).



Odottettu tulos: Postman antaa ilmoitukseen "Status: 200 Ok" ja taulun tiedot palautuvat Body-osiolla.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, pyynnön status on 200 OK ja taulun tiedot palautuvat Body-osiolla:

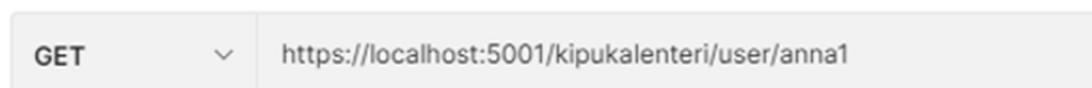
The screenshot shows a Postman interface with the following details:

- Header tab: Headers (4)
- Status bar: Status: 200 OK Time: 2.06 s Size: 975
- Body tab: Selected
- Raw tab: Available
- Preview tab: Available
- Visualize tab: Available
- JSON tab: Available
- Content:

```
1  [
2    {
3      "username": "annai",
4      "email": "anna@testi",
5      "password": "6c73d4cef222487c4ef80f3148c672f1",
6      "token": null,
7      "notes": [],
8      "painLogs": []
9    },
10   {
11     "username": "Esai",
```

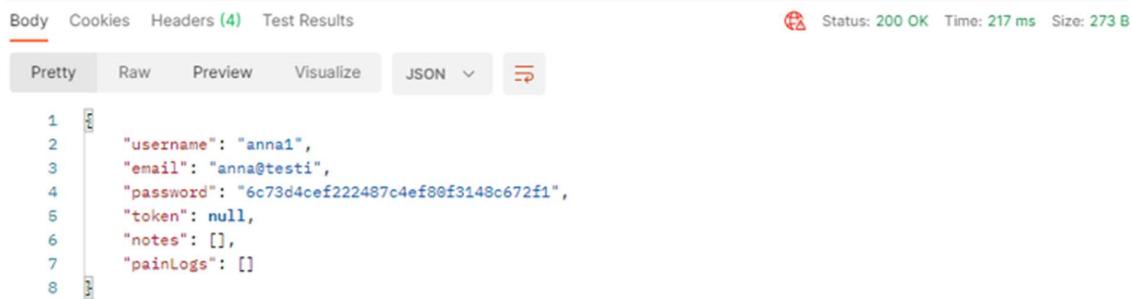
3.1.2.4.2 Haetaan yksi rivi tietoa

Haetaan taulusta yksi rivi tietoa `HttpGet`-metodilla (kontrollerissa `GetOne()`-metodi).



Odottettu tulos: Postman antaa ilmoitukseen "Status: 200 Ok" ja rivin tiedot näkyvät Body-osiolla.

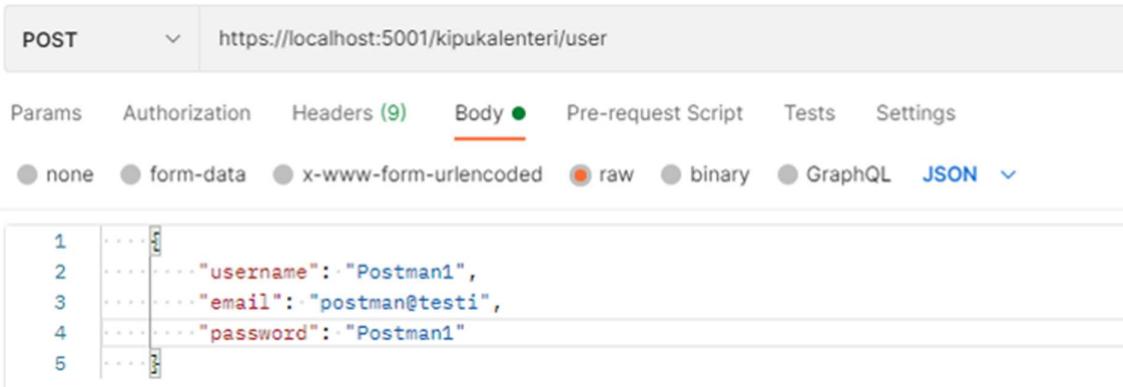
Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja haetun rivin tiedot palautuvat Body-osiosta.



```
1 "username": "anna1",
2 "email": "anna@testi",
3 "password": "6c73d4cef222487c4ef80f3148c672f1",
4 "token": null,
5 "notes": [],
6 "painLogs": []
```

3.1.2.4.3 Lisätään yksi rivi tietoa

Lisätään tietoa HttpPost-metodilla (kontrollerissa CreateNew()-metodi).



```
1 ...
2 ...
3 ...
4 ...
5 ...
```

```
1 ...
2 ...
3 ...
4 ...
5 ...
```

Odottettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiota ilmoituksessa palautuu lisätyn sijainnin nimi. Lisättyt tiedot löytyvät HttpGet-metodilla tietokannasta.

Tulos: Testi onnistuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiosta palautuu ilmoitus sijainnin lisäyksestä.



```
1 Muistiinpano Postman1 lisättyn
```

Lisätty rivi löytyy tietokannasta:

The screenshot shows the Postman interface with a JSON response. The response body contains two user objects:

```
26  {
27      "username": "ManualTest40",
28      "email": "testi@testi",
29      "password": "ManualTest1",
30      "token": null,
31      "notes": [],
32      "painLogs": []
33 },
34 {
35     "username": "Postman1",
36     "email": "postman@testi",
37     "password": "Postman1",
38     "token": null,
39     "notes": [],
40     "painLogs": []
41 },
42 }
```

3.1.2.4.4 Muokataan tietokantaan syötettyä tietoa

Muokataan aiemmin syötetyn rivin email-kenttää.

The screenshot shows a POST request in Postman. The URL is `https://localhost:5001/kipukalenteri/user/Postman1`. The Body tab is selected, showing the following JSON payload:

```
1   {
2     "username": "Postman1",
3     "email": "postman2@testi",
4     "password": "Postman1"
5 }
```

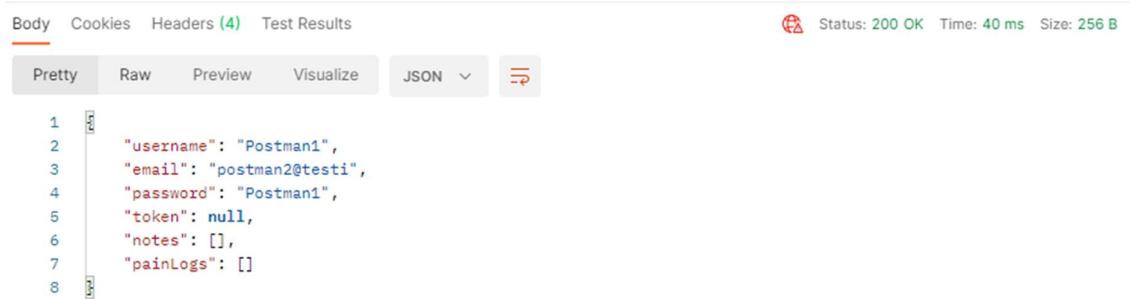
Odotettu tulos: Postman antaa ilmoituksen "Status: 200 Ok" ja Body-osiolla palautuu ilmoitus merkinnän päivittämisestä. Muokattu rivi löytyy HttpURLConnectionilla.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiolla palautuu ilmoitus sijainnin päivittämisestä.

The screenshot shows a successful 200 OK response in Postman. The response body is:

```
1   Käyttäjä Postman1 päivitetty
```

Päivitetty tiedot löytyvät tietokannasta:



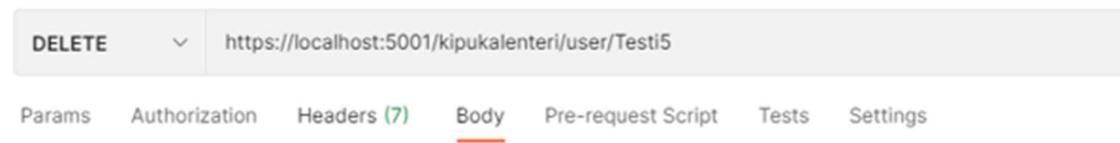
```

1
2   "username": "Postman1",
3   "email": "postman2@testi",
4   "password": "Postman1",
5   "token": null,
6   "notes": [],
7   "painLogs": []
8

```

3.1.2.4.5 Poistetaan tietokantaan syötettyä tietoa

Poistetaan yksi aiemmin syötetyistä riveistä tietokannasta.



Odotettu tulos: Postman antaa ilmoitukseen "Status: 200 Ok" Body-osiossa palautuu ilmoitus sijainnin poistamisesta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Pyynnön status on 200 OK ja Body-osiossa palautuu ilmoitus sijainnin poistamisesta.



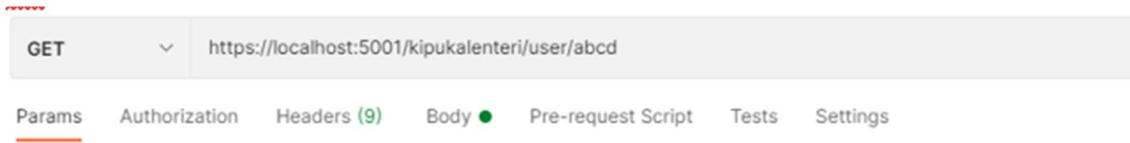
```

1   Käyttäjä Testi5 poistettiin

```

3.1.2.4.6 Haetaan riviä, jota ei ole olemassa

Haetaan tietoa HttpGet-metodilla päävaimella, jota ei ole Users-taulussa.



Odotettu tulos: Pyynnön status on 204 No Content, tuloksena ei tule mitään.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti.



```

1

```

3.1.2.4.7 Syötetään rivi, jonka username:lla on jo rivi tietokannassa

Lisätään tietokantatauluun rivi, jossa annetulla username-kentän päävaimella löytyy jo tietokannasta rivi.

```

1 ...
2 ...
3 ...
4 ...
5 ...
  
```

Odotettu tulos: Pyynnön status on 400 Bad Request ja Body-osiolla palautuu ilmoitus "Jokin meni pieleen käyttäjää lisättäessä". Lisäys ei onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, lisäys ei onnistu.

```

1 Jokin meni pieleen käyttäjää lisättäessä
  
```

3.1.2.4.8 Muokataan käyttäjänimeä

Laitetaan HttpPut-metodilla muokkauspyyntö, jossa muokataan username-kenttää (pääavain).

```

1 ...
2 ...
3 ...
4 ...
5 ...
  
```

Odotettu tulos: Pyynnön status on 400 Bad Request, muokkaus ei onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti.

```

1 Jokin meni pieleen käyttäjää Postman1 muokattaessa
  
```

3.1.2.4.9 Poistetaan rivi, jota ei löydy tietokantataulusta

Laitetaan HttpDelete-metodilla poistopyyntö pääavaimelle, jota ei löydy tietokantataulusta.

The screenshot shows the Postman interface with a DELETE request to the URL `https://localhost:5001/kipukalenteri/user/Testi555`. The Headers tab is selected, showing `Content-Type: application/x-www-form-urlencoded`. The Body tab is also visible. The response details show a status of 404 Not Found, time 36 ms, and size 184 B. The response body contains the message "Käyttäjää Testi555 ei löytynyt".

Odotettu tulos: Pyynnön status on 404 Not Found, poistoa ei toteuteta. Body-osiassa palautuu ilmoitus siitä, että käyttäjää ei löytynyt.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, eikä poistoa toteuteta koska käyttäjää ei löydy.

The screenshot shows the Postman interface with the response body displayed. It contains the message "Käyttäjää Testi555 ei löytynyt".

3.1.3 Käyttöliittymän manuaalinen testaus

Käyttöliittymän manuaalinen testaus suoritettiin valmiissa sovelluksessa lisäämällä, poistamalla ja muokkaamalla tietoja sovelluksen kautta. Yksittäiset testitapaukset on kuvattu tarkemmin testaussuunnitelmassa sekä tässä pöytäkirjassa.

Kaikki testit toteutuivat odotusten mukaan. Manuaalisen testauksen perusteella sovellus toimii oikein. Osa sovelluksen antamista ilmoituksista ja komponenttien ulkoasuista olivat kuitenkin testauksen perusteella puutteellisia ja niitä muokattiin. Testauksesta saatiin myös kehitystoimenpiteitä jatkoa varten.

3.1.3.1 Muistiinpanot-näkymän manuaalinen testaus

3.1.3.1.1 Lisätään muistiinpano oikeilla tiedoilla

Lisätään muistiinpano, jonka päivämäärä on aiemmin lisättyjä muistiinpanoja tuoreempi. Annetaan kaikki tietokannan vaativat tiedot (päivämäärä ja muistiinpanon teksti)

The screenshot shows the 'Muistiinpanot' (Notes) page. At the top, there is a header bar with icons for home, user, location, and file, followed by the text 'Kirjautuneena testiad' and a red square button with a white arrow. Below the header, the title 'Muistiinpanot' is displayed. A note card is shown with the following details:

- Päivämäärä: 04.06.2022 13.44
- WebApp Manual Test 1
- Käyttäjä-ID: testiad

At the bottom of the note card are two buttons: a red one with a white minus sign and a green one with a white checkmark.

Odottettu tulos: Sovellus antaa ilmoituksen ”Lisätty uusi muistiinpano” ja lisätty muistiinpano tulee näkymän ylimmäiseksi.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus antaa ilmoituksen muistiinpanon lisäyksestä ja muistiinpano löytyy ylimpänä listaukselta.

The screenshot shows a web application interface. At the top, there is a dark blue header bar with icons for home, refresh, location, file, user, and a red square button. The text "Kirjautuneena testiad" is displayed next to the user icon. Below the header, the title "Muistiinpanot" is shown in blue, followed by a green square button. A green-bordered box contains the text "Lisätty uusi muistiinpano". Underneath, there are two entries: "Päivämäärä: 4.6.2022 klo 13.44" and "WebApp Manual Test 1", each with a red delete icon and a green edit icon below it. Another entry follows: "Päivämäärä: 3.6.2022 klo 11.57" and "SQL Server DB manual test 3.1.1.1B", also with a red delete icon and a green edit icon.

3.1.3.1.2 Lisätään muistiinpano puutteellisilla tiedoilla

Lisätään muistiinpano ilman päivämäärää.

The screenshot shows the same application interface as the previous one. The header and "Muistiinpanot" title are identical. The note entry is now: "Päivämäärä" (date) is empty, "pp.kk.vvvv --.--", followed by a date picker icon. The note text is "App Manual Test 2". Below the note, the text "Käyttäjä-ID: testiad" is shown. At the bottom are two buttons: a red square with a white "X" and a green square with a white checkmark.

Odottettu tulos: Sovellus antaa virheilmoituksen eikä lisäys onnistu.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, lisäys ei onnistu. Virhekoodi ei ehkä kerro peruskäyttäjälle tarpeeksi, tämän tilanteen käsittelyä voisi parantaa.

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for home, back, location, and user, followed by the text "Kirjautuneena testiad" and a red refresh button. Below this is a section titled "Muistiinpanot" with a green plus icon. A red-bordered box highlights an error message: "Tapahtui virhe. Tässä lisätietoa: Error: Request failed with status code 400". Below the error message, there are two entries. The first entry has a red trash icon and a green edit icon. It shows the date "Päivämäärä: 3.6.2022 klo 11.57" and the text "SQL Server DB manual test 3.1.1.1B". The second entry also has a red trash icon and a green edit icon. It shows the date "Päivämäärä: 3.6.2022 klo 10.57" and the text "SQL Server DB manual test 3.1.1.1E". At the bottom, there is a note: "Päivämäärä: 23.4.2022 klo 00.00".

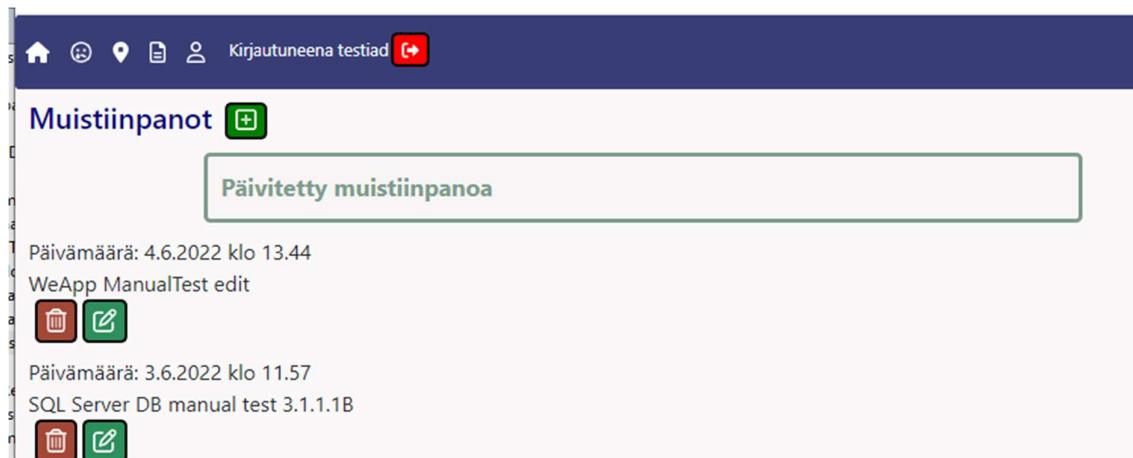
3.1.3.1.3 Muokataan olemassa olevaa muistiinpanoa

Muokataan aiemmin lisättynä muistiinpanoa muokkaamalla teksti-kenttää "WebApp ManualTest edit".

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for home, back, location, and user, followed by the text "Kirjautuneena testiad" and a red refresh button. Below this is a section titled "Muistiinpanot". A note is selected, shown in a light blue box. The note contains the text "Päivämäärä" above a date and time field "04.06.2022 13.44" with a calendar icon. Below this is a text area containing "WeApp ManualTest edit". Underneath the note, the text "Käyttäjä-ID: testiad" is visible. At the bottom of the note box are two buttons: a red one with a white "X" and a green one with a white checkmark.

Odottettu tulos: Sovellus antaa ilmoituksen "Päivitetty muistiinpanoa" ja muokattu teksti löytyy oikeasta muistiinpanosta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus ilmoittaa päivityksestä ja muokkaus onnistuu. Sovelluksen Lisää- ja Muokkaa- komponentteihin voisi lisätä otsikoinnit, jotta käyttäjä erottaa, kummassa toiminnossa hän on.

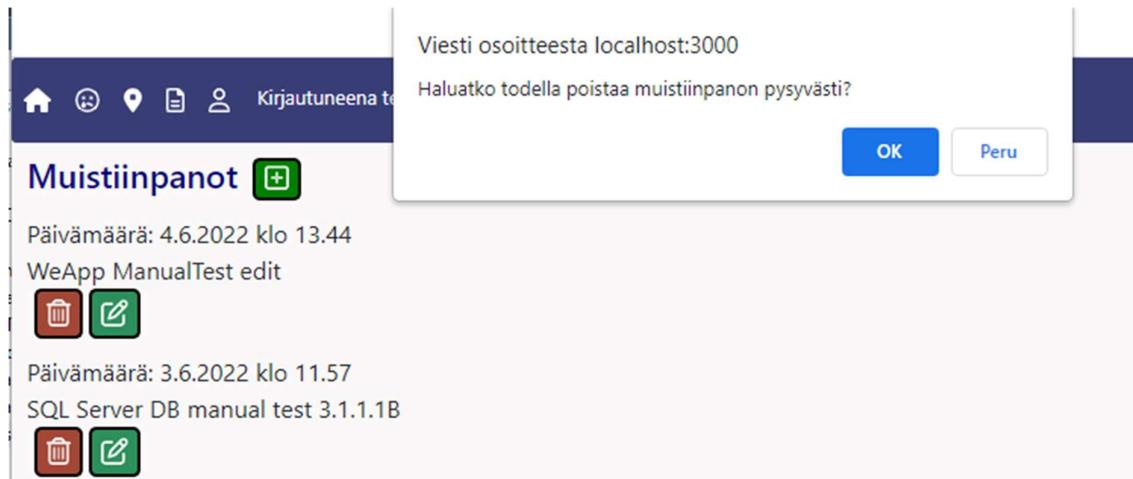


3.1.3.1.4 Poistetaan muistiinpano

Poistetaan aiemmin lisätty muistiinpano.

Odottettu tulos: Sovellus varmistaa, halutaanko poisto tehdä ja vahvistuksen jälkeen antaa ilmoituksen "Muistiinpanon poisto onnistui". Poistettua merkintää ei löydy muistiinpanoista.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Poiston vahvistuksen jälkeen sovellus antaa ilmoituksen poiston onnistumisesta, eikä poistettua muistiinpanoa löydy.



The screenshot shows a mobile application interface for 'Muistiinpanot' (Notes). At the top, there is a navigation bar with icons for home, refresh, location, and user, followed by the text 'Kirjautuneena testiaid' and a red logout button. Below the navigation bar, the title 'Muistiinpanot' is displayed with a green plus sign icon. A green-bordered box contains the message 'Muistiinpanon poisto onnistui' (Note deletion succeeded). Below this message, there are two entries. The first entry has a timestamp 'Päivämäärä: 3.6.2022 klo 11.57' and the text 'SQL Server DB manual test 3.1.1.1B'. It includes a red trash bin icon and a green edit icon. The second entry has a timestamp 'Päivämäärä: 3.6.2022 klo 10.57' and the text 'SQL Server DB manual test 3.1.1.1E'. It also includes a red trash bin icon and a green edit icon.

3.1.3.1.5 Tarkistetaan, että kaikki näkymän merkinnät ovat käyttäjän omia

Tarkistetaan, että kaikki näkymässä näkyvät merkinnät ovat myös tietokannassa merkitty kyseiselle käyttäjätunnuselle.

Odotettu tulos: Käyttäjän lisäämät muistiinpanot tietokannassa ja sovelluksessa vastaavat toisiaan.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, muistiinpanot tietokannassa ja sovelluksessa vastaavat toisiaan.

The screenshot shows a list of notes in the Kipukalenteri application. Each note entry includes a timestamp, text content, and edit/delete icons.

Päivämäärä	Sisältö	
3.6.2022 klo 11.57	SQL Server DB manual test 3.1.1.1B	[Delete] [Edit]
3.6.2022 klo 10.57	SQL Server DB manual test 3.1.1.1E	[Delete] [Edit]
23.4.2022 klo 00.00	Lisätään muistiinpano sovelluksen kautta	[Delete] [Edit]
1.4.2022 klo 09.42	Testing note controller	[Delete] [Edit]
1.4.2022 klo 09.42	Testing note controller	[Delete] [Edit]

The screenshot shows a database results window with a table of note data. The table has columns: NoteID, NoteText, NoteDate, and UserId.

	NoteID	NoteText	NoteDate	UserId
1	101	Muistiinpano kaksi johon tulee paljon enemmän te...	2021-07-04 00:00:00.000	Testi1
2	111	Lisätään muistiinpano sovelluksen kautta	2022-04-23 00:00:00.000	testiad
3	116	Testisyöttö	2022-04-24 18:28:00.000	Testi1
4	124	Testing note controller	2022-04-01 09:42:00.000	testiad
5	160	Tervetuloa Anna1! Testaa sovelluksen toimintoja.	2022-05-31 18:00:00.000	anna1
6	162	SQL Server DB manual test 3.1.1.1E	2022-06-03 10:57:00.000	testiad
7	163	SQL Server DB manual test 3.1.1.1B	2022-06-03 11:57:00.000	testiad
8	168	Testing note controller	2022-04-01 09:42:00.000	testiad

3.1.3.2 Sijainnit-näkymän manuaalinen testaus

3.1.3.2.1 Lisätään sijainti oikeilla tiedoilla

Lisätääni sijainti.



Odottettu tulos: Sovellus antaa ilmoituksen "Lisätty uusi sijainti" ja lisätty sijainti löytyy listauksesta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus antaa ilmoituksen sijainnin lisäyksestä ja sijainti löytyy listaukselta. Listaus tulee id:n mukaisessa järjestyksessä, sen voisi vaihtaa aakkosjärjestykseen.

ID	Sijainti	Vaihtoehdot
ID 104: Vatsa		
ID 105: Pää		
ID 107: Käsi		
ID 108: Niska		
ID 134: Selkä		
ID 3158: SQL Server DB manual test 5		
ID 3161: SQL Server DB manual test 4		
ID 3162: Postman API test 2		
ID 3165: App Manual Test 1		

3.1.3.2.2 Muokataan olemassa olevaa sijaintia

Muokataan aiemmin lisättyä sijaintia muokkaamalla kenttää "App Manual Test edit".



Odotettu tulos: Sovellus antaa ilmoituksen "Päivitetty sijaintia App Manual Test edit" ja muokattu sijainti löytyy listauksesta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus ilmoittaa päivityksestä ja muokkaus onnistuu.

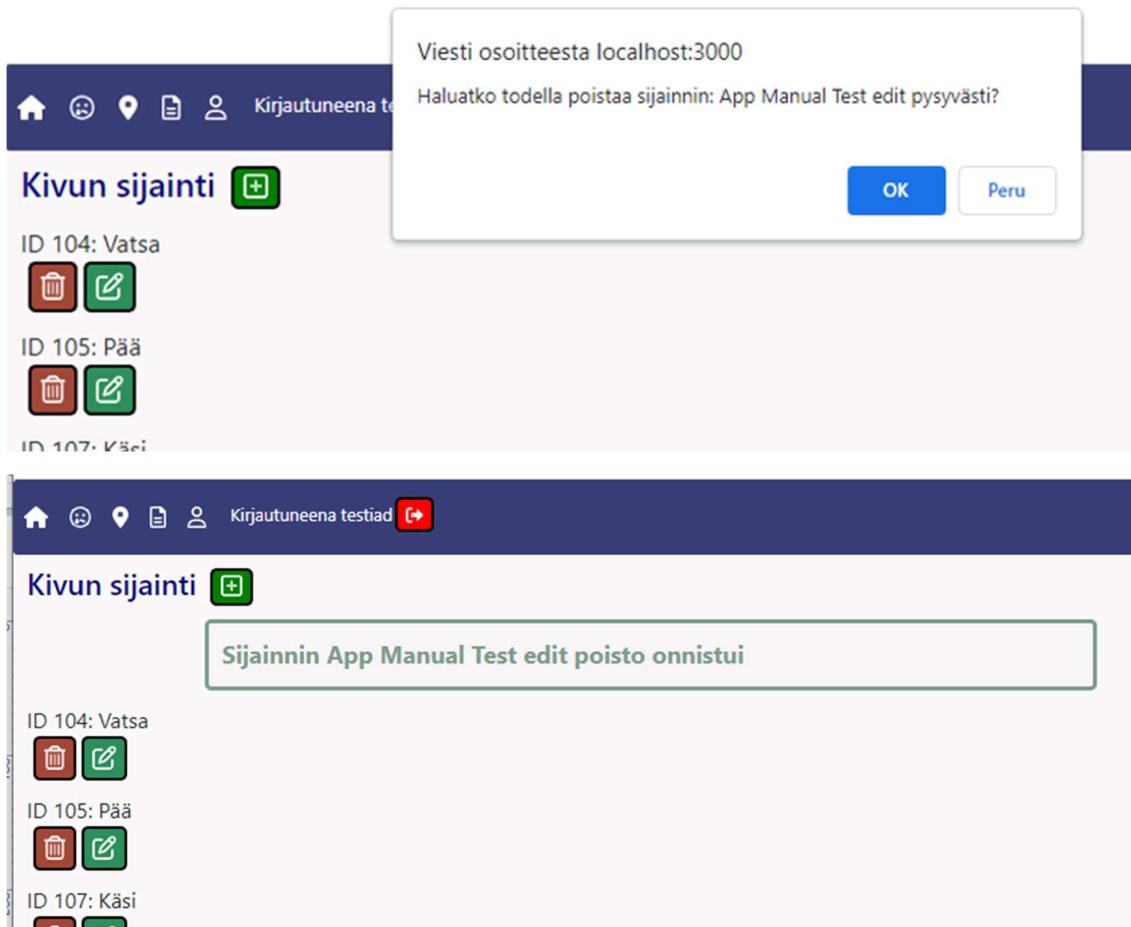
ID 104: Vatsa	[Edit Icon]
ID 105: Pää	[Edit Icon]
ID 107: Käsi	[Edit Icon]
ID 108: Niska	[Edit Icon]
ID 134: Selkä	[Edit Icon]
ID 3158: SQL Server DB manual test 5	[Edit Icon]
ID 3161: SQL Server DB manual test 4	[Edit Icon]
ID 3162: Postman API test 2	[Edit Icon]
ID 3165: App Manual Test edit	[Edit Icon]

3.1.3.2.3 Poistetaan sijainti

Poistetaan aiemmin lisätty muistiinpano.

Odottettu tulos: Sovellus varmistaa, halutaanko poisto tehdä ja vahvistuksen jälkeen antaa ilmoituksen "Sijainnin XXX poisto onnistui".

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Poiston vahvistuksen jälkeen sovellus antaa ilmoituksen poiston onnistumisesta.



3.1.3.2.4 Tarkistetaan, että käyttäjälle näkyvät kaikki sijainnit

Tarkistetaan, että kaikki tietokantaan tallennetut sijainnit näkyvät käyttäjälle Sijainnit-listauksessa.

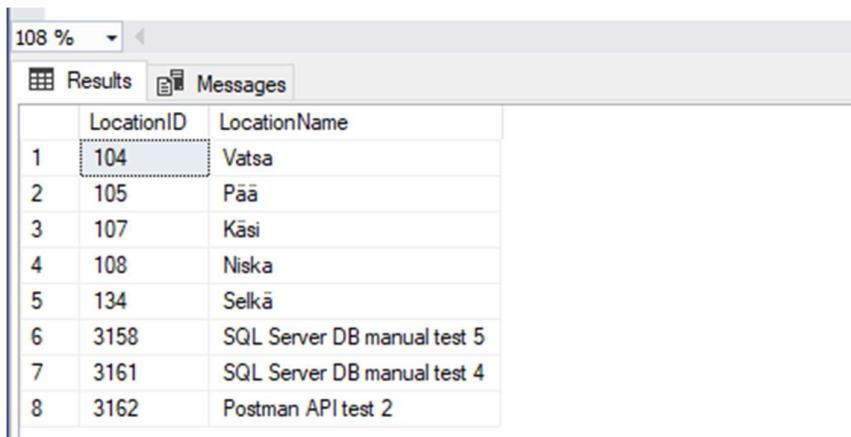
Odottettu tulos: Näkymän sijainnit ja tietokannan sijainnit vastaavat toisiaan.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sijainnit tietokannassa ja sovelluksessa vastaavat toisiaan.

The screenshot shows a mobile application interface with a dark blue header bar. In the top right corner of the header, there is a red button with a white arrow pointing right. The header also contains icons for home, refresh, location, file, user, and a red square.

The main content area has a light gray background. At the top left, the text "Kivun sijainti" is displayed in blue, followed by a green square icon containing a white plus sign. Below this, a list of items is shown, each consisting of a line of text and two small square icons below it (one red with a white trash bin, one green with a white edit/pencil icon).

- ID 104: Vatsa
- ID 105: Pää
- ID 107: Käsi
- ID 108: Niska
- ID 134: Selkä
- ID 3158: SQL Server DB manual test 5
- ID 3161: SQL Server DB manual test 4
- ID 3162: Postman API test 2

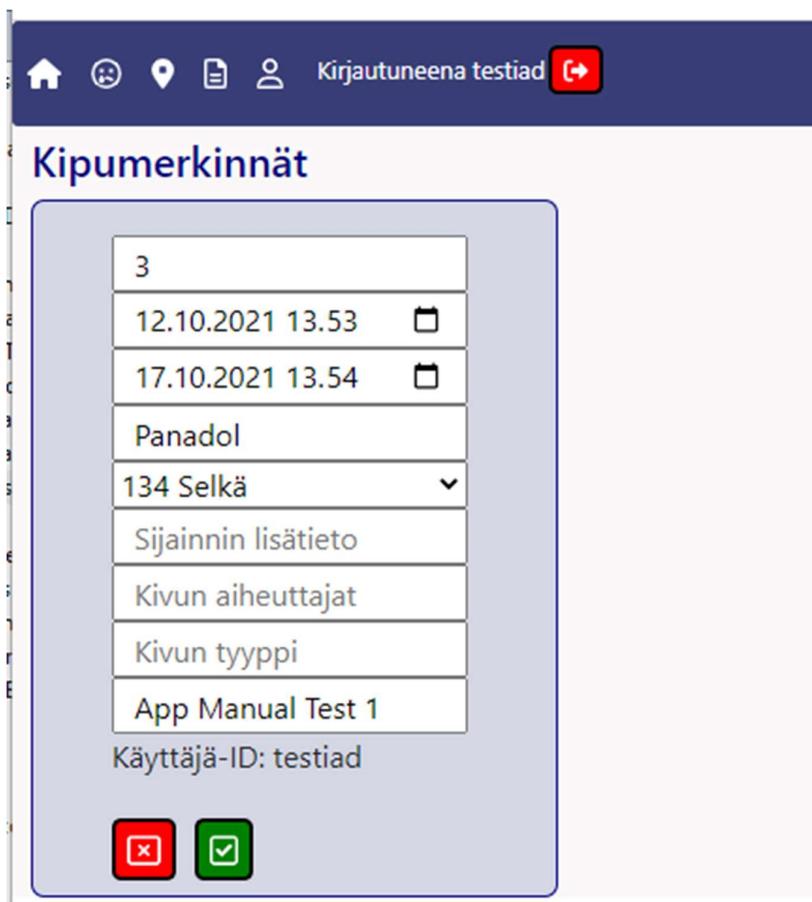


	LocationID	LocationName
1	104	Vatsa
2	105	Pää
3	107	Käsi
4	108	Niska
5	134	Selkä
6	3158	SQL Server DB manual test 5
7	3161	SQL Server DB manual test 4
8	3162	Postman API test 2

3.1.3.3 Kipumerkinnät-näkymän manuaalinen testaus

3.1.3.3.1 Lisätään merkintä oikeilla tiedoilla

Lisätään kipumerkintä ja täytetään kaikki tietokannan vaatimat kentät.



Kipumerkinnät

3
12.10.2021 13.53
17.10.2021 13.54
Panadol
134 Selkä
Sijainnin lisätieto
Kivun aiheuttajat
Kivun tyyppi
App Manual Test 1

Käyttäjä-ID: testiad

Odottettu tulos: Sovellus antaa ilmoituksen "Lisätty uusi merkintä" ja sijainti löytyy listauksesta. Kivun kesto tunneissa ja minuuteissa on laskettu oikein.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus antaa ilmoituksen merkinnän lisäyksestä ja merkintä löytyy listaukselta.

The screenshot shows a mobile application interface for managing health entries. At the top, there is a navigation bar with icons for home, refresh, location, and user, followed by the text "Kirjautuneena testiad" and a red logout button. Below the navigation bar, the title "Kipumerkinnät" is displayed with a green plus sign icon. Two dropdown menus are visible, both set to "Kaikki". A green rectangular box highlights the message "Lisätty uusi merkintä" (New entry added). Below this message, the text "Alkupäivämäärä: 4.6.2022 klo 19.33" and "Kivun sijainti: Pää" is shown, along with a red trash bin icon and a green edit icon. A horizontal dashed line follows. The next section contains the text "Kivun kesto on laskettu oikein". Below this, another set of details is shown: "Alkupäivämäärä: 12.10.2021 klo 13.53", "Kivun sijainti: Selkä", a red trash bin icon, and a green edit icon. A large blue rounded rectangle encloses a list of additional details: "Loppumisaika: 17.10.2021 klo 13.54", "Kivun kesto: 120 h 1 min", "Kivun intensiteetti (1-10): 3", "Lääkitys: Panadol", "Aiheuttaja:", "Kivun tyyppi:", "Lisätiedot: App Manual Test 1", and "Käyttäjä: testiad".

3.1.3.3.2 Muokataan olemassa olevaa merkintää

Muokataan aiemmin lisättyä merkintää muokkaamalla muistiinpanot-kenttää "App Manual Test edit".

The screenshot shows a mobile application interface for entering information about a kidney stone (Kipumerkinnät). The top navigation bar includes icons for home, settings, location, document, user, and sign-in status ('Kirjautuneena testiad'). A red 'Edit' button is also present.

Kivun alkamisaika: 12.10.2021 13.53 (with a calendar icon)

Kivun loppumisaika: 17.10.2021 13.54 (with a calendar icon)

Kivun intensiteetti 1 - 10: 3

Lääkitys: Panadol

Kivun sijainti: 134 Selkä (dropdown menu)

Sijainnin lisätieto: (empty input field)

Kivun aiheuttajat: (empty input field)

Kivun tyyppi: (empty input field)

Lisätiedot: App Manual Test edit

Käyttäjä-ID: testiad

At the bottom are two buttons: a red square with a white 'X' and a green square with a white checkmark.

Odottettu tulos: Sovellus antaa ilmoituksen "Päivitetty merkintää" ja muokattu merkintä löytyy listauksesta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus ilmoittaa päivityksestä ja muokkaus onnistuu.

The screenshot shows a mobile application interface for managing marked stones. At the top, there is a header bar with icons for home, location, and user, followed by the text "Kirjautuneena testiad" and a red sign-out button. Below the header, the title "Kipumerkinnät" is displayed with a green plus icon. Two dropdown menus are visible, both set to "Kaikki". A green rectangular box highlights the text "Päivitetty merkintää". Below this, three stone entries are listed:

- Alkupäivämäärä: 4.6.2022 klo 19.33
Kivun sijainti: Pää
[Delete] [Edit]
- Alkupäivämäärä: 4.6.2022 klo 11.02
Kivun sijainti: Pää
[Delete] [Edit]
- Alkupäivämäärä: 2.6.2022 klo 11.45
Kivun sijainti: Käsi
[Delete] [Edit]

Below these entries, a larger box contains the following information:

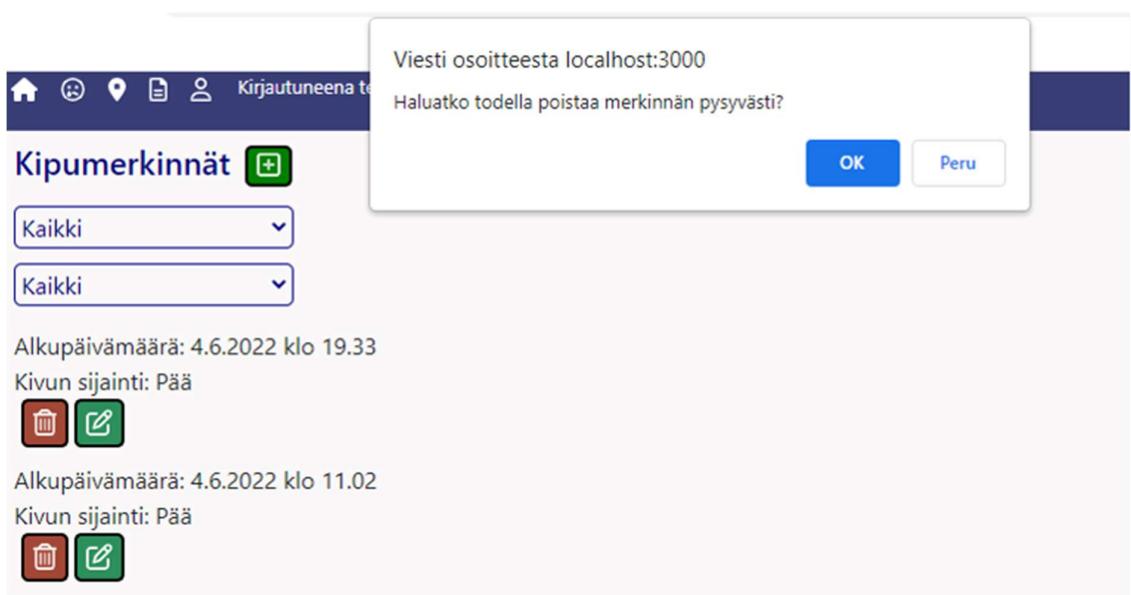
Loppumisaika: 17.10.2021 klo 13.54
Kivun kesto: 120 h 1 min
Kivun intensiteetti (1-10): 3
Lääkitys: Panadol
Aiheuttaja:
Kivun tyyppi:
Lisätiedot: App Manual Test edit
Käyttäjä: testiad

3.1.3.3.3 Poistetaan merkintä

Poistetaan aiemmin lisätty merkintä.

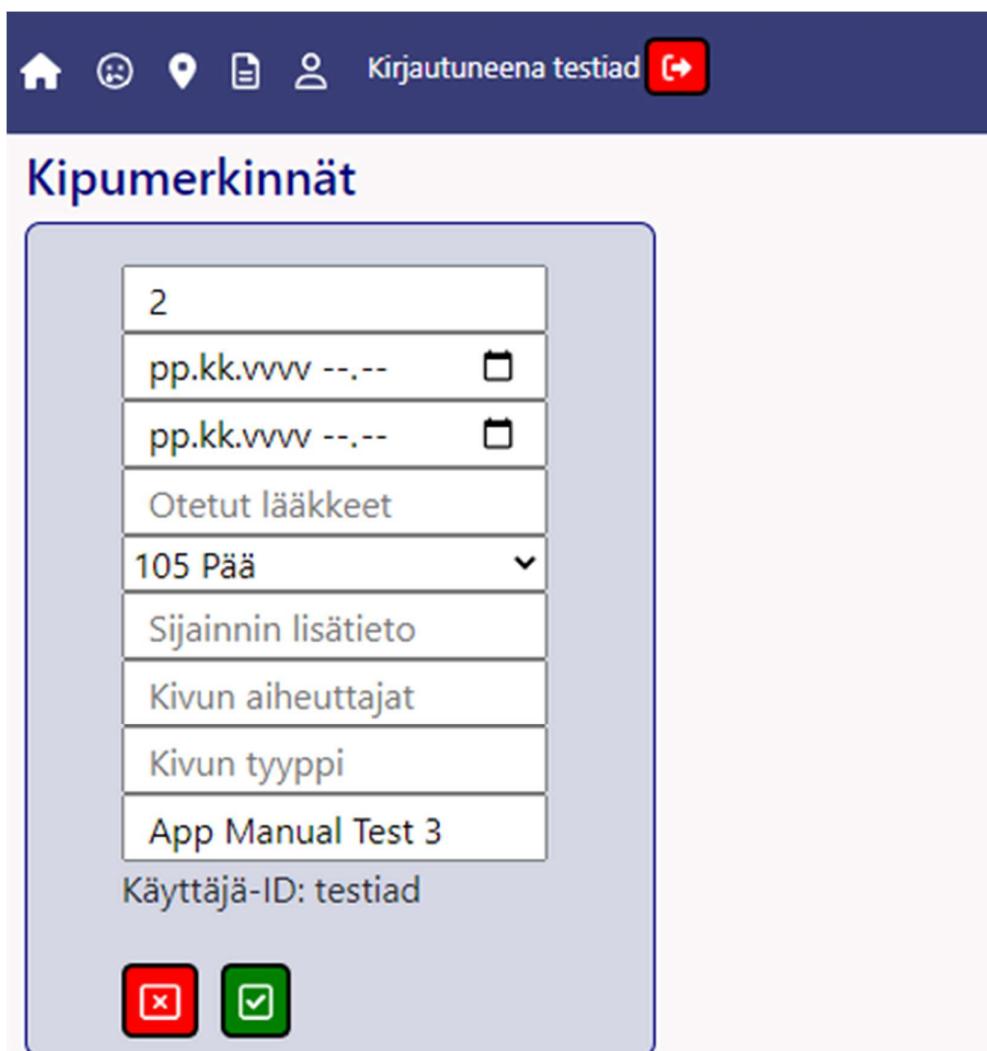
Odottettu tulos: Sovellus varmistaa, halutaanko poisto tehdä ja vahvistuksen jälkeen antaa ilmoituksen "Merkinnän poisto onnistui".

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Poiston vahvistuksen jälkeen sovellus antaa ilmoituksen poiston onnistumisesta.



ddd

- 3.1.3.3.4 Lisätään merkintä puutteellisilla tiedoilla
Lisätään kipumerkintä ilman alku- ja päätymispäivää.



Odottettu tulos: Sovellus antaa virheilmoituksen, eikä lisäys onnistu.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, eikä lisäys onnistu.



3.1.3.3.5 Lisätään merkintä, jossa alkamisaika on myöhemmin kuin päättymisaika
Lisätään merkintä, jossa alkamisaika on myöhemmin kuin päättymisaika.

The screenshot shows a mobile application interface for managing notes. At the top, there is a dark blue header bar with icons for home, refresh, location, and user, followed by the text "Kirjautuneena testiad" and a red "Logout" button. Below the header, the title "Kipumerkinnät" is displayed in blue. A vertical list of note details is shown in a light gray box:

2	
04.06.2022 14.00	<input type="button" value="Edit"/>
04.06.2022 13.01	<input type="button" value="Edit"/>
Otetut lääkkeet	
104 Vatsa	
Sijainnin lisätieto	
Kivun aiheuttajat	
Kivun tyyppi	
Lisätiedot	

Below the list, the text "Käyttäjä-ID: testiad" is displayed. At the bottom of the screen are two large buttons: a red one with a white "X" and a green one with a white checkmark.

Odottettu tulos: Lisäys onnistuu, mutta kivun kesto on -59 minuuttia

Tulos: Lisäys onnistuu, mutta kivun kesto on -1h 59 minuuttia. Kivun kesto lasketaan negatiivisena väärin. Sovellukseen täytyisi kehittää ratkaisu, jossa alkuaika ei voi olla myöhemmin kuin päättymisaika.

Alkupäivämäärä: 4.6.2022 klo 14.00

Kivun sijainti: Vatsa



Loppumisaika: 4.6.2022 klo 13.01

Kivun kesto: -1 h -59 min

Kivun intensiteetti (1-10): 2

Lääkitys:

Aiheuttaja:

Kivun tyyppi:

Lisätiedot:

Käyttäjä: testiad

3.1.3.3.6 Tarkistetaan, että käyttäjälle näkyvät vain hänen kipumerkintänsä

Tarkistetaan, että käyttäjälle näkyvät vain hänen omat kipumerkintänsä.

Odottettu tulos: Näkymän kipumerkinnät vastaavat tietokannassa käyttäjälle merkittyjä kipumerkintöjä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, kipumerkinnät tietokannassa ja sovelluksessa vastaavat toisiaan.

Kipumerkinnät 

Kaikki 

Kaikki 

Alkupäivämäärä: 4.6.2022 klo 19.33
Kivun sijainti: Pää
 

Alkupäivämäärä: 4.6.2022 klo 11.02
Kivun sijainti: Pää
 

Alkupäivämäärä: 2.6.2022 klo 11.45
Kivun sijainti: Käsi
 

Alkupäivämäärä: 1.6.2022 klo 11.30
Kivun sijainti: Käsi
 

Alkupäivämäärä: 1.6.2022 klo 08.30
Kivun sijainti: Niska
 

Results														Messages	
logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId				
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad				
2	4055	3	2022-06-01 11:30:00.000	2022-06-03 18:00:00.000	3270	NULL	NULL	NULL	107	SQL Server manual test 3.1.1.3B	testiad				
3	4057	5	2022-06-01 08:30:00.000	2022-06-01 09:45:00.000	75	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad				
4	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1				
5	4063	6	2022-06-01 08:30:00.000	2022-06-01 19:06:00.000	636	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1				
6	4064	6	2022-06-04 11:47:00.000	2022-06-04 11:42:00.000	-5	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Test1				
7	4065	6	2022-06-01 08:30:00.000	2022-06-05 15:07:00.000	6157	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1				
8	4066	7	2022-06-04 19:33:00.000	2022-06-04 20:35:00.000	62	NULL	NULL	NULL	105	Postman API manual test 2	testiad				
9	4070	2	2022-06-04 11:02:00.000	2022-06-04 11:02:00.000	1				108		testiad				

3.1.3.3.7 Kaikkien kipumerkintöjen kestot on oikein laskettu

Tarkistetaan, että kaikkien näkymän kipumerkintöjen kestot ovat oikein.

Odottettu tulos: Kaikkien kipumerkintöjen kestot on oikein laskettu.

Tulos: Kaikkien muiden kipumerkintöjen kestot on oikein laskettu, paitsi sen jossa kesto on negatiivinen eli kivun alkuaika myöhemmin kuin päättymisaika.

3.1.3.3.8 Sijaintisuodatin toimii

Testataan, että sijaintisuodatin suodattaa näkyviin oikeat merkinnät ja sijaintia voi suodattaa useita kertoja peräkkäin.

Odottettu tulos: Sijaintisuodattimen suodattamat merkinnät vastaavat tietokannan merkintöjä ja sijaintia voi suodattaa useita kertoja peräkkäin.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, suodatus sijainnilta toimii.

Ensimmäinen suodatus

Kipumerkinnät

105 Pää

Kaikki

Alkupäivämäärä: 4.6.2022 klo 19.33

Kivun sijainti: Pää

Alkupäivämäärä: 4.4.2022 klo 13.55

Kivun sijainti: Pää

Vastaan tietokannan tietoa

kipukalenteri.kipukalenteri - dbo.Users SQLQuery4.sql - kip...alenteriadmin (65)* X SQLQuery2.sql - not connected*

```
1 select * from painlog where UserId = 'testiad' and LocationId = 105
2
3
```

Results Messages

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4066	7	2022-06-04 19:33:00.000	2022-06-04 20:35:00.000	62	NULL	NULL	NULL	105	Postman API manual test 2	testiad
2	4070	2	2022-06-04 11:02:00.000	2022-06-04 11:03:00.000	1				105		testiad
3	4072	7	2022-06-04 13:55:00.000	2022-06-05 13:55:00.000	1440				105	App Manual Test 2	testiad

Toinen suodatus:

Kirjautuneena testiad

Kipumerkinnät

107 Käsi

Kaikki

Alkupäivämäärä: 2.6.2022 klo 11.45

Kivun sijainti: Käsi

Vastaan tietokannan tietoa

kipukalenteri.kipukalenteri - dbo.Users SQLQuery4.sql - kip...alenteriadmin (65)* X SQLQuery2.sql - not connected*

```
1 select * from painlog where UserId = 'testiad' and LocationId = 107
2
3
```

Results Messages

logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad

3.1.3.3.9 Aikasuodatin toimii

Testataan, että aikasuodatin suodattaa näkyviin oikeat merkinnät ja aikaa voi suodattaa useita kertoja peräkkäin.

Odottettu tulos: Aikasuodattimen suodattamat merkinnät vastaavat tietokannan merkintöjä ja merkintöjä voi suodattaa useita kertoja peräkkäin.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, suodatus sijainnilta toimii.

Ensimmäinen suodatus

Kipumerkinnät 

Kaikki 

Kuukausi 

Alkupäivämäärä: 4.6.2022 klo 19.33

Kivun sijainti: Pää

Alkupäivämäärä: 4.6.2022 klo 14.02

Kivun sijainti: Vatsa

Alkupäivämäärä: 4.6.2022 klo 14.00

Kivun sijainti: Vatsa

Alkupäivämäärä: 4.6.2022 klo 11.02

Kivun sijainti: Pää

Alkupäivämäärä: 2.6.2022 klo 11.45

Kivun sijainti: Käsi

Alkupäivämäärä: 1.6.2022 klo 08.30

Kivun sijainti: Niska

Toinen suodatus:

The screenshot shows a mobile application interface for 'Kipumerkinnät' (Markings). At the top, there is a header bar with icons for home, refresh, location, and user, followed by the text 'Kirjautuneena testiad' and a red sign-out button. Below the header, the title 'Kipumerkinnät' is displayed with a green '+' button. Two dropdown menus are visible: 'Kaikki' (All) and 'Kuusi kuukautta' (Six months), both with a blue downward arrow icon.

The main content area lists six marking entries, each with an 'Alkupäivämäärä' (Start date) and a 'Kivun sijainti' (Location):

- Alkupäivämäärä: 4.6.2022 klo 19.33
Kivun sijainti: Pää
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 4.6.2022 klo 14.02
Kivun sijainti: Vatsa
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 4.6.2022 klo 14.00
Kivun sijainti: Vatsa
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 4.6.2022 klo 11.02
Kivun sijainti: Pää
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 2.6.2022 klo 11.45
Kivun sijainti: Käsi
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 1.6.2022 klo 08.30
Kivun sijainti: Niska
Delete icon (red bin) | Edit icon (green pencil)
- Alkupäivämäärä: 4.4.2022 klo 13.55
Kivun sijainti: Pää
Delete icon (red bin) | Edit icon (green pencil)

Suodattimien tulokset vastaavat tietokannan tietoja:

108 %

	logID	PainIntensity	StartTime	EndTime	Duration	Medication	LocationInfo	PainTrigger	PainType	LocationID	Notes	UserId
1	4054	6	2022-06-02 11:45:00.000	2022-06-03 12:45:00.000	1500	NULL	NULL	NULL	NULL	107	SQL Server DB manual test 3.1.1.3E	testiad
2	4057	5	2022-06-01 08:30:00.000	2022-06-01 09:45:00.000	75	NULL	NULL	NULL	NULL	108	SQL Server manual test 3.1.1.3B	testiad
3	4061	6	2022-06-03 19:07:00.000	2022-06-03 20:35:00.000	88	NULL	NULL	NULL	NULL	107	SQL Server manual test trigger 1	anna1
4	4063	6	2022-06-01 08:30:00.000	2022-06-01 19:06:00.000	636	NULL	NULL	NULL	NULL	108	SQL Server manual test trigger 3	Esa1
5	4064	6	2022-06-04 11:47:00.000	2022-06-04 11:42:00.000	-5	NULL	NULL	NULL	NULL	105	SQL Server manual test trigger 4	Test1
6	4065	6	2022-06-01 08:30:00.000	2022-06-05 15:07:00.000	6157	NULL	NULL	NULL	NULL	108	SQL Server manual test trigger 5	Esa1
7	4066	7	2022-06-04 19:33:00.000	2022-06-04 20:35:00.000	62	NULL	NULL	NULL	NULL	105	Postman API manual test 2	testiad
8	4070	2	2022-06-04 11:02:00.000	2022-06-04 11:03:00.000	1					105		testiad
9	4071	3	2021-10-12 13:53:00.000	2021-10-17 13:54:00.000	7201	Panadol				134	App Manual Test edit	testiad
10	4072	7	2022-04-04 13:55:00.000	2022-04-05 13:55:00.000	1440					105	App Manual Test 2	testiad
11	4073	2	2022-06-04 14:00:00.000	2022-06-04 13:01:00.000	-59					104		testiad
12	4074	2	2022-06-04 14:02:00.000	2022-06-04 16:03:00.000	121					104	App HP Manual Test	testiad

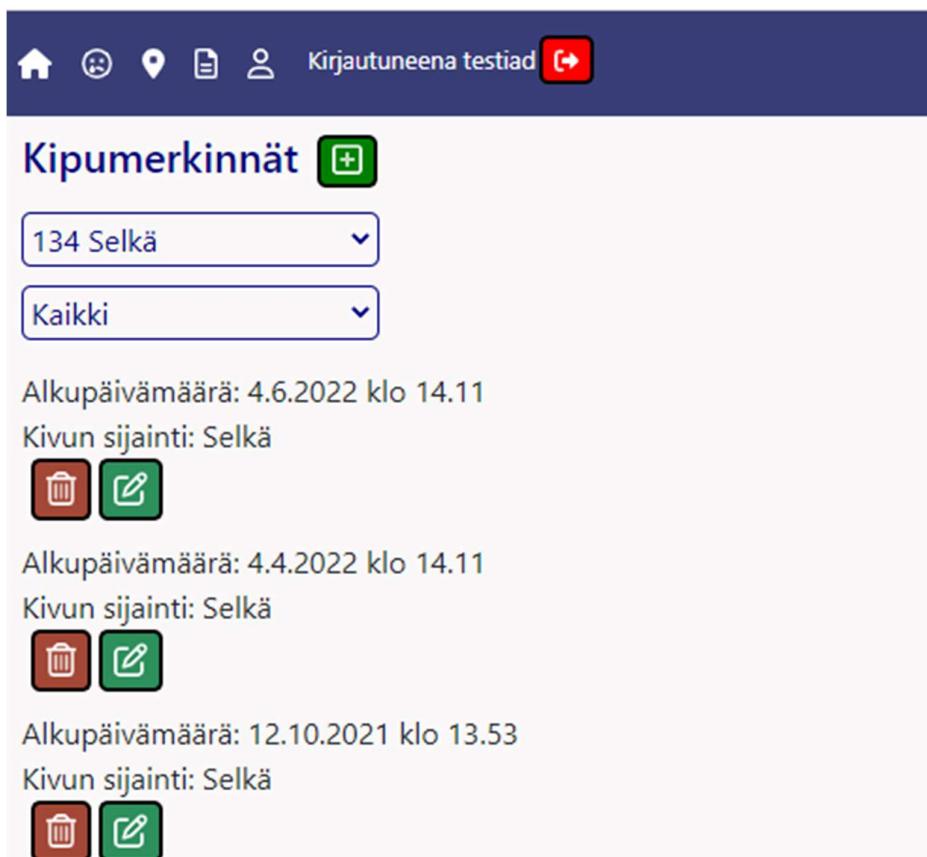
3.1.3.3.10 Molempia suodattimia voi käyttää yhtä aikaa useita kertoja peräkkäin

Testataan, että sekä aikasuodatinta että sijaintisuodatinta voi käyttää yhtä aikaa useita kertoja peräkkäin.

Odottettu tulos: Suodatuksit toimivat yhtä aikaa ja suodatusta voi muuttaa useita kertoja peräkkäin.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, suodattimet toimivat yhtä aikaa.

Ensimmäinen suodatus:



The screenshot shows the application's header with navigation icons and the text "Kirjautuneena testiad". Below the header, there are two dropdown menus for filtering:

- The first dropdown is set to "134 Selkä".
- The second dropdown is set to "Kaikki".

Below the filters, there are two entries listed:

- Alkupäivämäärä: 4.6.2022 klo 14.11**
Kivun sijainti: Selkä
With edit and delete icons.
- Alkupäivämäärä: 4.4.2022 klo 14.11**
Kivun sijainti: Selkä
With edit and delete icons.

At the bottom, there is another entry:

- Alkupäivämäärä: 12.10.2021 klo 13.53**
Kivun sijainti: Selkä
With edit and delete icons.

Toinen suodatus:

The screenshot shows the Kipumerkinnät (Markings) section of the application. At the top, there are navigation icons (Home, Refresh, Location, Document, User) and the text "Kirjautuneena testiad" followed by a red refresh button. Below this, the title "Kipumerkinnät" is displayed next to a green plus sign button. Two dropdown menus are visible: the first one is set to "134 Selkä" and the second to "Kuukausi" (Month). Below the dropdowns, the text "Alkupäivämäärä: 4.6.2022 klo 14.11" and "Kivun sijainti: Selkä" are shown. At the bottom are two buttons: a red trash can icon and a green edit/pencil icon.

Kolmas suodatus:

The screenshot shows the Kipumerkinnät section again. The navigation bar and user information are identical to the previous screenshot. The dropdown menus show "134 Selkä" and "Kuusi kuukautta" (Six months). The text "Alkupäivämäärä: 4.6.2022 klo 14.11" and "Kivun sijainti: Selkä" are displayed. Below these are the same red trash can and green edit buttons. Further down, the text "Alkupäivämäärä: 4.4.2022 klo 14.11" and "Kivun sijainti: Selkä" are shown, along with the red trash can and green edit buttons.

3.1.3.4 Käyttäjät-näkymän manuaalinen testaus

3.1.3.4.1 Lisätään käyttäjä sovelluksen kautta

Lisätään käyttäjä käyttäjänimellä "AppManualTest" sovellukseen.

Käyttäjät

manual@testi
AppManualTest
.....
.....

[Red X icon] [Green checkmark icon]

Odotettu tulos: Sovellus antaa ilmoituksen "Lisätty käyttäjä AppManualTest". Lisättyä käyttäjää ei näy näkymässä, mutta se löytyy tietokannasta.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus antaa ilmoituksen lisäyksestä.

Käyttäjät [+]

Lisätty käyttäjä AppManualTest

Käyttäjä: testiad

[Delete icon] [Edit icon]

Käyttäjä löytyy tietokannasta:

	Username	Email	Password	Token
1	AlbaUser	Alba@testi	Alba1	NULL
2	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	AppManualTest	manual@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
4	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
5	ManualTest4D	testi@testi	ManualTest1	NULL
6	Postman1	postman2@testi	Postman1	NULL
7	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
8	Testi3	testi@testi	ManualTest1	NULL
9	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

3.1.3.4.2 Muokataan käyttäjän sähköpostia

Muokataan käyttäjän sähköpostiksi testi11@testi.fi

Käyttäjät

testi11@testi.fi
.....
.....

Käyttäjä: testiad

X ✓

Odotettu tulos: Sovellus antaa ilmoituksen ”Päivitetty käyttäjää testiad” ja käyttäjän sähköposti vastaa päivitystä.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti, sovellus ilmoittaa päivityksestä ja muokkaus onnistuu.

Käyttäjät +

Päivitetty käyttäjää testiad

Käyttäjä: testiad

X ✓

Muokkaus onnistuu, tieto löytyy kannasta:

9	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
10	Testi3	testi@testi	ManualTest1	NULL
11	testiad	testi11@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL
12	testinen	testi	9627df7a4a5b849f67ce863e82adc71	NULL

3.1.3.4.3 Poistetaan käyttäjä

Yritetään poistaa käyttäjä.

Odotettu tulos: Sovellus varmistaa, halutaanko poisto tehdä. Vahvistuksen jälkeen sovellus antaa virheilmoituksen, eikä poistoa tehdä. Käyttäjällä on tietokannassa muistiinpanoja ja

kipumerkintöjä, eikä poistoa voi tehdä koska käyttäjänimi toimii viiteavaimena näissä merkinnöissä.

Tulos: Testi toteutuu odotetun tuloksen mukaisesti. Poiston vahvistuksen jälkeen sovellus antaa virheilmoituksen, eikä poistoa tehdä.

The screenshot shows a confirmation dialog box in the foreground with the following text:
Viesti osoitteesta localhost:3000
Haluatko varmasti poistaa käyttäjän testiad?

Below the dialog is a user list page with the following interface:
Header: Home, Refresh, Location, File, User, Kirjautuneena testiad (Logout)
Section: Käyttäjät [Add button]
List item: Käyttäjä: testiad [Delete icon, Edit icon]
Error message: Tapahtui virhe: Error: Request failed with status code 404

3.1.3.4.4 Tarkistetaan, että käyttäjälle näky vain hänen oma tunnuksensa
Tarkistetaan, että käyttäjälle näkyy vain hänen oma tunnuksensa.

Odottettu tulos: Käyttäjälle näkyy vain yksi käyttäjätunnus, hänen omansa.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, käyttäjälle näkyy vain hänen oma tunnuksensa. Tunnuksesta näkyy vain nimi, sovellusta voisi muuttaa että lisätietoja näkyisi klikkaamalla, kuten kipumerkinnöissä.

The screenshot shows a user list page with the following interface:
Header: Home, Refresh, Location, File, User, Kirjautuneena testiad (Logout)
Section: Käyttäjät [Add button]
List item: Käyttäjä: testiad [Delete icon, Edit icon]

	Username	Email	Password	Token
1	AlbaUser	Alba@testi	Alba1	NULL
2	anna1	anna@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
3	Esa1	Esa@testi	6c73d4cef222487c4ef80f3148c672f1	NULL
4	ManualTest4D	testi@testi	ManualTest1	NULL
5	Postman1	postman2@testi	Postman1	NULL
6	Testi1	pusa.jaana@gmail.com	KaivoKello01	NULL
7	Testi3	testi@testi	ManualTest1	NULL
8	testiad	testi@testi.fi	534ba41761c18b8b0e15a3dd4bdaed75	NULL

3.1.3.5 Kirjautumisen manuaalinen testaus

3.1.3.5.1 Kirjautuminen tunnuksilla, joita ei ole olemassa

Yritetään kirjautua käyttäjätunnuksella ja salasanalla, jota ei ole olemassa

The screenshot shows a login form with two input fields: 'admin' in the username field and '*****' in the password field. Below the fields are three buttons: a red square with a white 'X', a green square with a white right-pointing arrow, and a green square with a white plus sign. A large blue error message box is displayed above the form, containing the text 'VÄÄRÄ KÄyttäjätunnus tai salasana'.

Odottettu tulos: Kirjautuminen ei onnistu.

Tulos: Yritettäessä kirjautua kokonaan väärillä tunnuksilla, tyhjällä käyttäjätunnuksella ja/tai salasanalla tai väärällä salasanalla sovellus ei reagoi mitenkään, mitään ei tapahdu. Kirjautuminen ei onnistu, mutta käyttäjälle pitäisi antaa jonkinlainen ilmoitus.

3.1.3.5.2 Käyttäjän lisääminen kirjautumatta

Lisätään käyttäjä Manual1 kirjautumatta ja kirjaudutaan sen jälkeen luodulla tunnuksilla sovellukseen.

The screenshot shows a mobile application interface. At the top, there is a dark blue header bar. Below it, the word "Käyttäjät" (Users) is displayed in blue text. A list of users is shown in a table-like structure with four rows. The first row contains "app@manual". The second row contains "Manual1". The third row contains ".....". The fourth row contains ".....". At the bottom of the list are two buttons: a red button with a white "X" icon and a green button with a white checkmark icon.

Odotettu tulos: Lisäys onnistuu ja käyttäjä pääsee kirjautumaan sovellukseen.

Tulos: Käyttäjän lisäys onnistuu ja tunnukset toimivat. Käyttäjälle näkyvät kaikki sijainnit, mutta koska tunnuksella ei ole tehty lisäyksiä kipumerkintöihin tai muistiinpanoihin, sivulla näkyy vain teksti "Lataa..." tai "Ei tuloksia, päivitä sivu...". Näitä tekstejä voisi muokata.

3.1.3.5.3 Lisää kipumerkintä -nappi toimii

Testataan, että etusivun lisää kipumerkintä -nappi avaa Lisää kipumerkintä -komponentin ja kipumerkinnän lisäys toimii.

The screenshot shows a mobile application interface. At the top, there is a dark blue header bar with icons for home, refresh, location, document, and user, followed by the text "Kirjautuneena testiad" and a red arrow icon. Below the header, there are two blue buttons with white text: "Lisää kipumerkintä" and "Lisää sijainti".

Odotettu tulos: Nappi avaa Lisää kipumerkintä -komponentin. Kipumerkinnän lisäys onnistuu, lisäyksen jälkeen sovellus palaa etusivulle.

Tulos: Testi toteutui odotusten mukaisesti. Nappi toimii ja merkinnän lisäyksen jälkeen sovellus siirtyy etusivulle. Lisäyksestä ei kuitenkaan tule mitään ilmoitusta. Ilmoitus voisi selkiyttää toimintaa käyttäjälle.

Kipumerkinnät

Lisää kipumerkintä

Intensiteetti 1-10

pp.kk.vvvv --.-- pp.kk.vvvv --.--

Otetut lääkkeet



Sijainnin lisätieto

Kivun aiheuttajat

Kivun tyyppi

Lisätiedot

Käyttäjä-ID: testiad



Lisäys onnistuu:

Kipumerkinnät

Kaikki Kaikki 

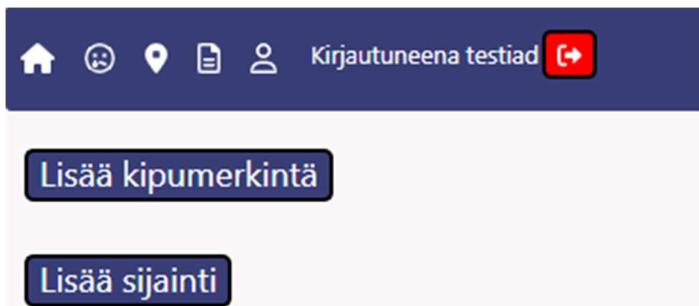
Alkupäivämäärä: 5.6.2022 klo 16.39

Kivun sijainti: Niska



3.1.3.5.4 Lisää sijainti -nappi toimii

Testataan, että etusivun lisää sijainti -nappi avaa Lisää sijainnin -komponentin ja sijainnin lisäys toimii.



Odottettu tulos: Nappi avaa Lisää sijainti -komponentin. Sijainnin lisäys onnistuu, lisäyksen jälkeen sovellus palaa etusivulle.

Tulos: Testi toteutui odotusten mukaisesti. Nappi toimii ja sijainnin lisäyksen jälkeen sovellus siirtyy etusivulle. Lisäyksestä ei kuitenkaan tule mitään ilmoitusta. Ilmoitus voisi selkiyttää toimintaa käyttäjälle.

Kivun sijainti



ID 3179: testing front page



3.2 Automaattinen testaus

3.2.1 Yksikkötestaus Jest-kirjastolla

Yksikkötestauksen kaikki testitapaukset toteutuvat testitapaikkuissa määriteltyjen odotettujen tulosten mukaisesti. Kaikki testitiedostot sekä testitiedostojen yksittäiset testit menivät testien ajossa läpi.

```
Test Suites: 12 passed, 12 total
Tests:      28 passed, 28 total
Snapshots:  0 total
Time:       6.67 s
Ran all test suites related to changed files.
```

3.2.1.1 Note-kansion komponenttien automaattinen testaus

3.2.1.1.1 Note-komponentin automaattinen testaus

Testi koostuu kolmesta osasta. Ensin testataan, että komponentti renderöi sille propsina annetun muuttujan tiedot oikein. Seuraavaksi testataan, että Poista- ja Muokkaa -napit kutsuvat tapahtumankäsittelijää vain kerran.

```
File UserEdit.test.js U JS Note.test.js U X UserEdit.jsx U
c > Note > JS Note.test.js > ⚡ describe('<Note/>') callback > [⌚] mockHandler
1 import React from 'react'
2 import '@testing-library/jest-dom/extend-expect'
3 import { render, fireEvent } from '@testing-library/react'
4 import Note from './Note'
5
6 describe('<Note/>', () => {
7   let component
8
9   const note = {
10     noteId: "1111",
11     noteDate: '2022-04-01T09:42:00',
12     noteText: "Testing note component"
13   }
14
15   const mockHandler = jest.fn()
16
17   beforeEach(() => {
18     component = render(
19       <Note note={note} handleDeleteClick={mockHandler} handleEditClick={mockHandler} />
20     )
21   })
22
23   test('renders the date and text of note', () => {
24     const div = component.container.querySelector('.notepage')
25     expect(div).toHaveTextContent(
26       'Testing note component'
27     )
28     expect(component.container).toHaveTextContent(
29       'Päivämäärä: 1.4.2022 klo 09.42'
30     )
31   })
32
33   test('Button click only calls the event handler once', async () => {
34     const button = component.container.querySelector('#notedelete')
35     fireEvent.click(button)
36     expect(mockHandler.mock.calls).toHaveLength(1)
37   })
38
39   test('Button click only calls the event handler once', async () => {
40     const button = component.container.querySelector('#noteedit')
41     fireEvent.click(button)
42     expect(mockHandler.mock.calls).toHaveLength(1)
43   })
44 })
```

Odottettu tulos: Komponentti renderöi annetun muuttujan noteDate- ja noteText-tiedot sivulle oikein. Napit kutsuvat tapahtumankäsittelijöitä vain kerran.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu renderöi päivämäärä ja muistiinpanon tekstin ja napit kutsuvat tapahtumankäsittelijöitä vain kerran.

3.2.1.1.2 NoteAdd-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS NoteAdd.test.js U X
src > Note > JS NoteAdd.test.js > ⚡ describe('<NoteAdd/>') callback
  1 import React from 'react'
  2 import { render, fireEvent } from '@testing-library/react'
  3 import '@testing-library/jest-dom/extend-expect'
  4 import NoteAdd from './NoteAdd'
  5
  6 describe('<NoteAdd/>', () => {
  7   let component
  8   const mockFunktio = jest.fn()
  9
 10  beforeEach(() => {
 11    component = render(
 12      <NoteAdd setAddNote={mockFunktio} setNotes={mockFunktio} notes={mockFunktio} setMessage={mockFunktio}
 13        setShowMessage={mockFunktio} setIsPositive={mockFunktio}/>
 14    )
 15  })
 16
 17  test('Eventhandler is called only once', async() => {
 18    const button = component.container.querySelector('#cancelnoteadd')
 19    fireEvent.click(button)
 20    expect(mockFunktio.mock.calls).toHaveLength(1)
 21  })
 22
 23  test('Form updates states and submit works', () => {
 24    const noteInput = component.container.querySelector('#notetext')
 25    const noteDate = component.container.querySelector('#notedate')
 26    const form = component.container.querySelector('form')
 27
 28    fireEvent.change(noteInput, {
 29      target: { value: 'Testnote' }
 30    });
 31
 32    fireEvent.change(noteDate, {
 33      target: {value: '2022-04-01T09:42:00'}
 34    })
 35
 36    fireEvent.submit(form)
 37
 38    setTimeout(() => {
 39      expect(mockFunktio.mock.calls).toHaveLength(1)
 40      expect(mockFunktio.mock.calls[0][0]).toBe('Testnote')
 41      expect(mockFunktio.mock.calls[0][1]).toHaveTextContent('Päivämäärä: 1.4.2022 klo 09.42')
 42    }, 0);
 43  })
 44})

```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran.

Lomake päivittää submitilla annetut tiedot hooks-muuttujien stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.1.3 NoteEdit-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS NoteEdit.test.js U X
src > Note > JS NoteEdit.test.js > ⚡ describe("<NoteEdit/>") callback
1  import React from 'react'
2  import { render, fireEvent } from '@testing-library/react'
3  import '@testing-library/jest-dom/extend-expect'
4  import NoteEdit from './NoteEdit'
5
6  describe('<NoteEdit/>', () => {
7    let component
8    const mockFunktio = jest.fn()
9
10   beforeEach(() => {
11     component = render(
12       <NoteEdit setEditNote={mockFunktio} setNotes={mockFunktio} notes={mockFunktio} setMessage={mockFunktio}
13         setShowMessage={mockFunktio} setIsPositive={mockFunktio} changedNote={mockFunktio}/>
14     )
15   })
16
17   test('Eventhandler is called only once', async() => {
18     const button = component.container.querySelector('#cancelnoteedit')
19     fireEvent.click(button)
20     expect(mockFunktio.mock.calls).toHaveLength(1)
21   })
22
23   test('Form updates states and submit works', () => {
24     const noteInput = component.container.querySelector('#textedit')
25     const noteDate = component.container.querySelector('#dateedit')
26     const form = component.container.querySelector('form')
27
28     fireEvent.change(noteInput, {
29       target: { value: 'Testnote' }
30     });
31
32     fireEvent.change(noteDate, {
33       target: {value: '2022-04-01T09:42:00'}
34     })
35
36     fireEvent.submit(form)
37
38     setTimeout(() => {
39       expect(mockFunktio.mock.calls).toHaveLength(1)
40       expect(mockFunktio.mock.calls[0][0]).toBe('Testnote')
41       expect(mockFunktio.mock.calls[0][1]).toHaveTextContent('Päivämäärä: 1.4.2022 klo 09.42')
42     }, 0);
43   })
44 })
```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujien stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.2 Location -kansion komponenttien automaattinen testaus

3.2.1.2.1 Location-komponentin automaattinen testaus

Testi koostuu kolmesta osasta. Ensin testataan, että komponentti renderöi sille propsina annetun muuttujan tiedot oikein. Seuraavaksi testataan, että Poista- ja Muokkaa -napit kutsuvat tapahtumankäsittelijää vain kerran.

```
S PainlogEdit.test.js U JS Location.test.js U X
src > Location > JS Location.test.js > ⚡ describe('<Location/>') callback > ⚡ test('Button click only calls the event handler once') callback > [x] button
  1 import React from 'react'
  2 import '@testing-library/jest-dom/extend-expect'
  3 import { render, fireEvent } from '@testing-library/react'
  4 import Location from './Location'
  5
  6 describe('<Location/>', () => {
  7   let component
  8
  9   const location = {
10     locationId: "1111",
11     locationName: "Testlocation"
12   }
13
14   const mockHandler = jest.fn()
15
16   beforeEach(() => {
17     component = render(
18       <Location location={location} handleDeleteClick={mockHandler} handleEditClick={mockHandler} />
19     )
20   })
21
22   test('renders the name and id of location', () => {
23
24     const div = component.container.querySelector('.notepage')
25     expect(div).toHaveTextContent(
26       'Testlocation'
27     )
28     expect(component.container).toHaveTextContent(
29       '1111'
30     )
31   })
32
33   test('Button click only calls the event handler once', async () => {
34     const button = component.container.querySelector('#locdelbutton')
35     fireEvent.click(button)
36     expect(mockHandler.mock.calls).toHaveLength(1)
37   })
38
39   test('Button click only calls the event handler once', async () => {
40     const button = component.container.querySelector('#loceditbutton')
41     fireEvent.click(button)
42     expect(mockHandler.mock.calls).toHaveLength(1)
43   })
44 })
```

Odotettu tulos: Komponentti renderöi annetun muuttujan locationId- ja locationName-tiedot sivulle oikein. Napit kutsuvat tapahtumankäsittelijötä vain kerran.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu renderöi annetut tiedot ja napit kutsuvat tapahtumankäsittelijötä vain kerran.

3.2.1.2.2 LocationAdd-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS LocationAdd.test.js U X
src > Location > JS LocationAdd.test.js > ⚡ describe('<LocationAdd/>') callback > [e] component
  1 import React from 'react'
  2 import { render, fireEvent } from '@testing-library/react'
  3 import '@testing-library/jest-dom/extend-expect'
  4 import LocationAdd from './LocationAdd'
  5
  6 describe('<LocationAdd/>', () => {
  7   let component
  8   const mockFunktio = jest.fn()
  9
 10  beforeEach(() => {
 11    component = render(
 12      <LocationAdd setAddLocation={mockFunktio} setLocations={mockFunktio} locations={mockFunktio} setMessage={mockFunktio}
 13      setShowMessage={mockFunktio} setIsPositive={mockFunktio}/>
 14    )
 15  })
 16
 17  test('Eventhandler is called only once', async() => {
 18    const button = component.container.querySelector('#cancellocadd')
 19    fireEvent.click(button)
 20    expect(mockFunktio.mock.calls).toHaveLength(1)
 21  })
 22
 23  test('Form updates states and submits the form', () => {
 24    const locationInput = component.container.querySelector('#locationInput')
 25    const form = component.container.querySelector('form')
 26
 27    fireEvent.change(locationInput, {
 28      target: { value: 'Testlocation' }
 29    });
 30    fireEvent.submit(form)
 31
 32    setTimeout(() => {
 33      expect(mockFunktio.mock.calls).toHaveLength(1) //ei toimi, tulee nollana...
 34      expect(mockFunktio.mock.calls[0][0]).toBe('Testlocation')
 35    }, 0);
 36  })
 37})
 38})
```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujan stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.2.3 LocationEdit-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS LocationEdit.test.js U X
src > Location > JS LocationEdit.test.js ...
1  import React from 'react'
2  import { render, fireEvent } from '@testing-library/react'
3  import '@testing-library/jest-dom/extend-expect'
4  import LocationEdit from './LocationEdit'
5
6  describe('<LocationEdit>', () => {
7    let component
8    const mockFunktio = jest.fn()
9
10   beforeEach(() => {
11     component = render(
12       <LocationEdit setEditLocation={mockFunktio} setLocations={mockFunktio} locations={mockFunktio} setMessage={mockFunktio}
13         setShowMessage={mockFunktio} setIsPositive={mockFunktio} changedLocation={mockFunktio} />
14     )
15   })
16
17   test('Eventhandler is called only once', async() => {
18     const button = component.container.querySelector('#cancellocedit')
19     fireEvent.click(button)
20     expect(mockFunktio.mock.calls).toHaveLength(1)
21   })
22
23   test('Form updates states and submits the form', () => {
24     const locationInput = component.container.querySelector('input')
25     const form = component.container.querySelector('form')
26
27     fireEvent.change(locationInput, {
28       target: { value: 'Testlocation' }
29     });
30
31     fireEvent.submit(form)
32
33     setTimeout(() => {
34       expect(mockFunktio.mock.calls).toHaveLength(1) //ei toimi, tulee nollana...
35       expect(mockFunktio.mock.calls[0][0]).toBe('Testlocation')
36     }, 0);
37   })
38 })
39 )
```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujien stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.3 PainLog -kansion komponenttien automaattinen testaus

3.2.1.3.1 PainLog-komponentin automaattinen testaus

Testi koostuu kolmesta osasta. Ensin testataan, että Poista- ja Muokkaa -napit kutsuvat tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentti renderöi sille propsina annetun muuttujan tiedot oikein.

```
JS Painlog.test.js U X
src > Painlog > JS Painlog.test.js > ⚙️ describe('<Painlog/>') callback
  1 import React from 'react'
  2 import '@testing-library/jest-dom/extend-expect'
  3 import { render, fireEvent } from '@testing-library/react'
  4 import Painlog from './Painlog'
  5
  6 describe('<Painlog/>', () => {
  7   let component
  8   const log = {
  9     logId: '1111',
 10     painIntensity: '1',
 11     startTime: '2022-04-01T09:42:00',
 12     endTime: '2022-04-01T10:42:00',
 13     medication: "testMedication",
 14     locationInfo: "testLocation",
 15     painTrigger: "testTrigger",
 16     painType: "testType",
 17     locationId: '107',
 18     notes: "Testnotes"
 19   }
 20
 21   const mockHandler = jest.fn()
 22
 23   beforeEach(() => {
 24     component = render(
 25       <Painlog log={log} handleDeleteClick={mockHandler} handleEditClick={mockHandler} />
 26     )
 27   })
 28
 29   test('Button click only calls the event handler once', async () => {
 30     const button = component.container.querySelector('#logdelete')
 31     fireEvent.click(button)
 32     expect(mockHandler.mock.calls).toHaveLength(1)
 33   })
 34
 35   test('Button click only calls the event handler once', async () => {
 36     const button = component.container.querySelector('#logedit')
 37     fireEvent.click(button)
 38     expect(mockHandler.mock.calls).toHaveLength(1)
 39   })
 40
 41   test('renders content properly', () => {
 42     const div = component.container.querySelector('.notepage')
 43     expect(div).toHaveTextContent(
 44       '1.4.2022 klo 09.42'
 45     )
 46   })
 47 })
```

Odottettu tulos: Komponentti renderöi annetun muuttujan startTime-tiedon sivulle oikein. Napit kutsuvat tapahtumankäsittelijötä vain kerran.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu renderöi päivämäärän oikein ja napit kutsuvat tapahtumankäsittelijötä vain kerran.

3.2.1.3.2 PainLogAdd-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS PainlogAdd.test.js U X
src > Painlog > JS PainlogAdd.test.js > ⚙️ describe('<PainlogAdd/>') callback > ⚙️ test('Painlog add form updates states and submits the form') callback
  1 import React from 'react'
  2 import { render, fireEvent } from '@testing-library/react'
  3 import '@testing-library/jest-dom/extend-expect'
  4 import PainlogAdd from './PainlogAdd'
  5
  6 describe('<PainlogAdd/>', () => {
  7   let component
  8   const mockFunktio = jest.fn()
  9
 10  beforeEach(() => {
 11    component = render(
 12      <PainlogAdd setAddPainlog={mockFunktio} setPainlogs={mockFunktio} painlogs={mockFunktio} setMessage={mockFunktio}
 13        setShowMessage={mockFunktio} setIsPositive={mockFunktio}>
 14    )
 15  })
 16
 17  test('Cancel-button eventhandler is called only once', async() => {
 18    const button = component.container.querySelector('#cancellogadd')
 19    fireEvent.click(button)
 20    expect(mockFunktio.mock.calls).toHaveLength(1)
 21  })
 22
 23  test('Painlog add form updates states and submits the form', () => {
 24    const intensity = component.container.querySelector('#intensity')
 25    const startDate = component.container.querySelector('#starttime')
 26    const endDate = component.container.querySelector('#endtime')
 27    const form = component.container.querySelector('form')
 28
 29    fireEvent.change(intensity, {
 30      target: { value: '5' }
 31    });
 32    fireEvent.change(startDate, {
 33      target: {value: '2022-04-01T09:42:00'}
 34    })
 35    fireEvent.change(endDate, {
 36      target: {value: '2022-04-01T09:42:00'}
 37    })
 38    fireEvent.submit(form)
 39
 40    setTimeout(() => {
 41      expect(mockFunktio.mock.calls).toHaveLength(1)
 42      expect(mockFunktio.mock.calls[0][0]).toBe('5')
 43      expect(mockFunktio.mock.calls[0][1]).toHaveTextContent('1.4.2022 klo 09.42')
 44      expect(mockFunktio.mock.calls[0][1]).toHaveTextContent('1.4.2022 klo 09.42')
 45    }, 0);
 46  })
 47})
```

Odottettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujan stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.3.3 PainLogEdit-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS PainlogEdit.test.js U X
src > Painlog > JS PainlogEdit.test.js > ⚡ describe('<PainlogEdit/>') callback > ⚡ test('Painlog edit form updates states and submits the form') callback
  1 import React from 'react'
  2 import { render, fireEvent } from '@testing-library/react'
  3 import '@testing-library/jest-dom/extend-expect'
  4 import PainlogEdit from './PainlogEdit'
  5
  6 describe('<PainlogEdit/>', () => {
  7   let component
  8   const mockFunktio = jest.fn()
  9
 10  beforeEach(() => {
 11    component = render(
 12      <PainlogEdit setEditPainlog={mockFunktio} setPainlogs={mockFunktio} painlogs={mockFunktio} setMessage={mockFunktio}>
 13        setShowMessage={mockFunktio} setIsPositive={mockFunktio} changedLog={mockFunktio} />
 14    )
 15  })
 16
 17  test('Cancel-button eventhandler is called only once', async() => {
 18    const button = component.container.querySelector('#canceleditlog')
 19    fireEvent.click(button)
 20    expect(mockFunktio.mock.calls).toHaveLength(1)
 21  })
 22
 23  test('Painlog edit form updates states and submits the form', () => {
 24    const intensity = component.container.querySelector('#painintensity')
 25    const startDate = component.container.querySelector('#startdate')
 26    const endDate = component.container.querySelector('#enddate')
 27    const form = component.container.querySelector('form')
 28
 29    fireEvent.change(intensity, {
 30      target: { value: '5' }
 31    });
 32    fireEvent.change(startDate, {
 33      target: { value: '2022-04-01T09:42:00' }
 34    })
 35    fireEvent.change(endDate, {
 36      target: { value: '2022-04-01T09:42:00' }
 37    })
 38    fireEvent.submit(form)
 39
 40    setTimeout(() => {
 41      expect(mockFunktio.mock.calls).toHaveLength(1)
 42      expect(mockFunktio.mock.calls[0][0]).toBe('5')
 43      expect(mockFunktio.mock.calls[0][1]).toHaveTextContent('1.4.2022 klo 09.42')
 44      expect(mockFunktio.mock.calls[0][2]).toHaveTextContent('1.4.2022 klo 09.42')
 45    }, 0);
 46  })
 47})
```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujien stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.4 *User -kansion komponenttien automaattiset testit*

3.2.1.4.1 User-komponentin automaattinen testaus

Testi koostuu kolmesta osasta. Ensin testataan, että Poista- ja Muokkaa -napit kutsuvat tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentti renderöi sille propsina annetun muuttujan tiedot oikein.

```
JS User.test.js U X
src > Users > JS User.test.js > ⚡ describe('<User/>') callback
  1 import React from 'react'
  2 import '@testing-library/jest-dom/extend-expect'
  3 import { render, fireEvent } from '@testing-library/react'
  4 import User from './User'
  5
  6 describe('<User/>', () => {
  7   let component
  8
  9   const user = {
10     username: 'JestTest'
11   }
12
13   const mockHandler = jest.fn()
14
15   beforeEach(() => {
16     component = render(
17       <User user={user} handleDeleteClick={mockHandler} handleEditClick={mockHandler} />
18     )
19   })
20
21   test('Button click only calls the event handler once', async () => {
22     const button = component.container.querySelector('#userdelete')
23     fireEvent.click(button)
24     expect(mockHandler.mock.calls).toHaveLength(1)
25   })
26
27   test('Button click only calls the event handler once', async () => {
28     const button = component.container.querySelector('#useredit')
29     fireEvent.click(button)
30     expect(mockHandler.mock.calls).toHaveLength(1)
31   })
32
33   test('renders the page content', () => {
34
35     const div = component.container.querySelector('.notepage')
36     expect(div).toHaveTextContent(
37       'JestTest'
38     )
39   })
40 })
```

Odottettu tulos: Komponentti renderöi annetun muuttujan username-tiedon sivulle oikein. Napit kutsuvat tapahtumankäsittelijöitä vain kerran.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu renderöi päivämäärän oikein ja napit kutsuvat tapahtumankäsittelijöitä vain kerran.

3.2.1.4.2 UserAdd-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```
JS UserAdd.test.js U
src > Users > JS UserAdd.test.js > ⚡ describe('<UserAdd/>') callback > ⚡ test('Form updates states and submits the form') callback > ✎ userinput
  6  describe('<UserAdd/>', () => {
  7    let component
  8    const mockFunktio = jest.fn()
  9
 10   beforeEach(() => {
 11     component = render(
 12       <UserAdd setAddUser={mockFunktio} setUsers={mockFunktio} users={mockFunktio} setMessage={mockFunktio}
 13         setShowMessage={mockFunktio} setIsPositive={mockFunktio}>
 14     )
 15   })
 16
 17   test('Eventhandler is called only once', async() => {
 18     const button = component.container.querySelector('#canceluseradd')
 19     fireEvent.click(button)
 20     expect(mockFunktio.mock.calls).toHaveLength(1)
 21   })
 22
 23   test('Form updates states and submits the form', () => {
 24     const userInput = component.container.querySelector('#userinput')
 25     const form = component.container.querySelector('form')
 26
 27     fireEvent.change(userInput, {
 28       target: { value: 'JestTest' }
 29     });
 30
 31     fireEvent.submit(form)
 32
 33     setTimeout(() => {
 34       expect(mockFunktio.mock.calls).toHaveLength(1)
 35       expect(mockFunktio.mock.calls[0][0]).toBe('JestTest')
 36     }, 0);
 37   })
 38 })
```

Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujan stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.1.4.3 UserEdit-komponentin automaattinen testaus

Testi koostuu kahdesta osasta. Ensin testataan, että Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Seuraavaksi testataan, että komponentin lomake päivittää hooks-muuttujien statet ja lomakkeen submit toimii.

```

JS UserEdit.test.js ×

src > Users > JS UserEdit.test.js > ⚡ describe('<UserEdit/>') callback > ⚡ test('Form updates states and submits the form') callback > [e] emailInput
17   test('Eventhandler is called only once', async() => {
18     const button = component.container.querySelector('#canceluseredit')
19     fireEvent.click(button)
20     expect(mockFunktio.mock.calls).toHaveLength(1)
21   })
22
23   test('Form updates states and submits the form', () => {
24     const emailInput = component.container.querySelector('#emailinput')
25     const passwordInput = component.container.querySelector('#passwordinput')
26     const passwordAgainInput = component.container.querySelector('#passwordagaininput')
27     const form = component.container.querySelector('form')
28
29     fireEvent.change(emailInput, {
30       target: { value: 'Jest@Test' }
31     });
32
33     fireEvent.change(passwordInput, {
34       target: { value: 'Jest' }
35     });
36
37     fireEvent.change(passwordAgainInput, {
38       target: { value: 'Jest' }
39     });
40
41     fireEvent.submit(form)
42
43     setTimeout(() => {
44       expect(mockFunktio.mock.calls).toHaveLength(1)
45       expect(mockFunktio.mock.calls[0][0]).toBe('Jest@Test')
46     }, 0);
47   })
48 })

```

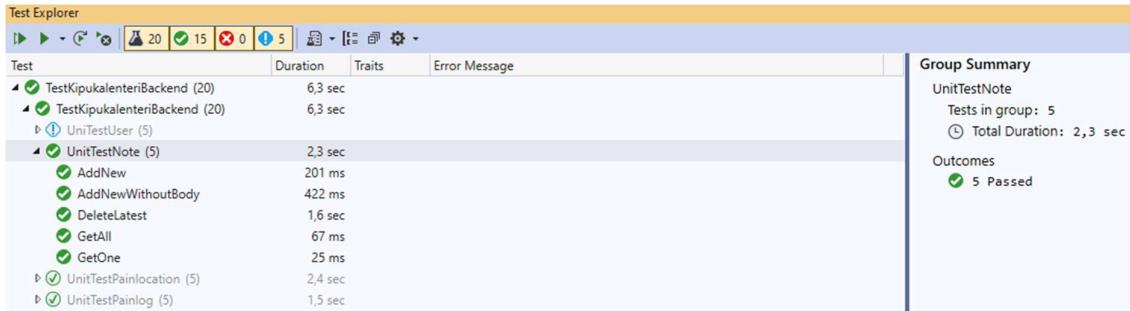
Odotettu tulos: Komponentin Peruuta-nappi kutsuu tapahtumankäsittelijää vain kerran. Lomake päivittää submitilla annetut tiedot hooks-muuttujien stateen ja submit kutsuu tapahtumankäsittelijää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, nappi kutsuu tapahtumankäsittelijää vain kerran ja lomake sekä submit toimivat odotetusti.

3.2.2 Integraatiotestaus Alba-kirjastolla

3.2.2.1 NoteControllerin integraatiotestaus

Testitiedosto koostuu viidestä yksittäisestä testistä. Kaikki testit on suoritettu samalla kertaa onnistuneesti, testien tulokset vastasivat alla testitapaikkuvaussissa asetettuja odotettuja tuloksia.



3.2.2.1.1 GetAll()-metodin testaus

Testataan, että kontrollerin GetAll()-metodi toimii ja tuloksena saatu tieto ei ole tyhjä

```
9  namespace TestKipukalenteriBackend
10 {
11     public class UnitTestNote
12     {
13         [Fact]
14         public async Task GetAll()
15         {
16             var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
17             using var system = new AlbaHost(hostBuilder);
18             {
19                 await system.Scenario(s =>
20                 {
21                     s.Get.Url("/kipukalenteri/note");
22                     s.StatusCodeShouldBeOk();
23                 });
24
25             var results = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/note");
26             Assert.NotEmpty(results);
27         }
28     }
}
```

Odotettu tulos: Tietojen haku GetAll()-metodilla toimii, tuloksena saatu tieto ei ole tyhjä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.1.2 GetOne() -metodin testaus

Testataan, että GetOne()-metodi toimii. Hakua testataan kolmella tavalla:

Haetaan Id:llä, jota tietokantataulussa ei ole

Haetaan Id:llä, joka taulusta löytyy

Haetaan Id:llä, joka on väärän tyypinen

```
[Fact]
public async Task GetOne()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/note/7554");
            s.StatusCodeShouldBe(204);

            s.Get.Url("/kipukalenteri/note/101");
            s.StatusCodeShouldBe(200);

            s.Get.Url("/kipukalenteri/note/note");
            s.StatusCodeShouldBe(400);
        });
    }
}
```

Odotettu tulos: Ensimmäisen kohdan pyynnöstä saadaan vastaukseksi status-koodi 204 No Content, toisen kohdan pyynnöstä saadaan status-koodi 200 OK ja kolmannen kohdan pyynnöstä status-koodi 400 Bad Request.

Tulos: Testi toteutui odotetun tuloksen mukaan, status-koodit vastasivat oletettua.

3.2.2.1.3 AddNewWithoutBody() - Create()-metodin testaus

Lähetetään HttpPost -metodille lisäyspyyntö laittamatta Body-osioon lähetettävän muistiinpanon tietoja.

```
52     | 0 references
53     public async Task AddNewWithoutBody()
54     {
55         var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
56         using var system = new AlbaHost(hostBuilder);
57         {
58             await system.Scenario(s =>
59             {
60                 s.Post.Url("/kipukalenteri/note");
61                 s.StatusCodeShouldBe(415);
62             });
63         }
64     }
```

Odottu tulos: Vastauksena saadaan status-koodi 415 Unsupported Media Type, lisäys ei onnistu.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.1.4 AddNew() - Create()-metodin testaus

Lähetetään HttpPost-metodilla lisäyspyyntö ja Body-osiossa lähetettävät tiedot.

```
65     [Fact]
66     | 0 references
67     public async Task AddNew()
68     {
69         var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
70         using var system = new AlbaHost(hostBuilder);
71         {
72             var results1 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/note");
73             var count1 = results1.Count();
74
75             var note = new Note
76             { NoteDate = DateTime.Parse("2022-04-01T09:42:00"), NoteText = "Testing note controller", UserId = "testiad" };
77
78             await system.Scenario(s =>
79             {
80                 s.Post.Json<Note>(note).ToUrl("/kipukalenteri/note");
81                 s.StatusCodeShouldBe(200);
82             });
83
84             var results2 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/note");
85             Assert.Equal(count1 + 1, results2.Count());
86         }
87     }
```

Odottu tulos: Lisäys onnistuu ja vastauksena saadaan status-koodi 200 OK. Notes-taulun tulosten määrä on lisääntynyt yhdellä ennen lisäystä lasketusta määrästä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.1.5 Delete() -metodin testaus

Poistetaan viimeisimmäksi lisätty rivi tietokantataulusta HttpDelete-metodilla.

```

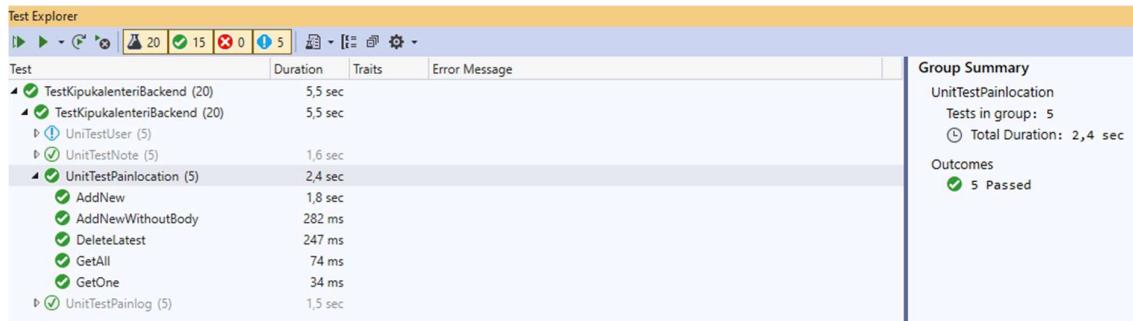
88 [Fact]
89     public async Task DeleteLatest()
90     {
91         var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
92         using var system = new AlbaHost(hostBuilder);
93         {
94
95             var results1 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/note");
96             var count1 = results1.Count();
97             int biggestId = (from r in results1 select r.NoteId).ToArray().Max();
98
99             await system.Scenario(s =>
100             {
101                 s.Delete.Url("/kipukalenteri/note/" + biggestId);
102                 s.StatusCodeShouldBe(200);
103             });
104
105             var results2 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/note");
106             Assert.Equal(count1 - 1, results2.Count());
107         }
108     }
109 }
```

Odotettu tulos: Poisto onnistuu, vastauksena saadaan status-koodi 200 OK ja taulun tulosten määrä on vähenytynyt yhdellä ennen poistoa lasketusta määristä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.2 PainLocationControllerin integraatiotestaus

Testitiedosto koostuu viidestä yksittäisestä testistä. Kaikki testit on suoritettu samalla kertaa onnistuneesti, testien tulokset vastasivat alla testitapauskuvauksissa asetettuja odotettuja tuloksia.



3.2.2.2.1 GetAll()-metodin testaus

Testataan, että kontrollerin GetAll()-metodi toimii ja tuloksena saatu tieto ei ole tyhjä

```
0 references
public class UnitTestPainlocation
{
    [Fact]
    0 | 0 references
    public async Task GetAll()
    {
        var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
        using var system = new AlbaHost(hostBuilder);
        {
            await system.Scenario(s =>
            {
                s.Get.Url("/kipukalenteri/painlocation");
                s.StatusCodeShouldBeOk();
            });

            var results = await system.GetAsJson<IEnumerable<PainLocation>>("/kipukalenteri/painlocation");
            Assert.NotEmpty(results);
        }
    }
}
```

Odotettu tulos: Tietojen haku GetAll()-metodilla toimii, tuloksena saatu tieto ei ole tyhjä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.2.2 GetOne() -metodin testaus

Testataan, että GetOne()-metodi toimii. Hakua testataan kolmella tavalla:

Haetaan Id:llä, jota tietokantataulussa ei ole

Haetaan Id:llä, joka taulusta löytyy

Haetaan Id:llä, joka on väärän tyypininen

```
[Fact]
0 | 0 references
public async Task GetOne()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/painlocation/7554");
            s.StatusCodeShouldBe(204);

            s.Get.Url("/kipukalenteri/painlocation/105");
            s.StatusCodeShouldBe(200);

            s.Get.Url("/kipukalenteri/painlocation/location");
            s.StatusCodeShouldBe(400);
        });
    }
}
```

Odotettu tulos: Ensimmäisen kohdan pyynnöstä saadaan vastaukseksi status-koodi 204 No Content, toisen kohdan pyynnöstä saadaan status-koodi 200 OK ja kolmannen kohdan pyynnöstä status-koodi 400 Bad Request.

Tulos: Testi toteutui odotetun tuloksen mukaan, status-koodit vastasivat oletettua.

3.2.2.2.3 AddNewWithoutBody() - Create()-metodin testaus

Lähetetään HttpPost -metodille lisäyspyyntö laittamatta Body-osioon lähetettävän sijainnin tietoja.

```
| :| [Fact]
| :| | 0 references
| :| public async Task AddNewWithoutBody()
| :| {
| :|     var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
| :|     using var system = new AlbaHost(hostBuilder);
| :|     {
| :|         await system.Scenario(s =>
| :|         {
| :|             s.Post.Url("/kipukalenteri/painlocation");
| :|             s.StatusCodeShouldBe(415);
| :|         });
| :|     }
| :| }
```

Odotettu tulos: Vastauksena saadaan status-koodi 415 Unsupported Media Type, lisäys ei onnistuu.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.2.4 AddNew() - Create()-metodin testaus

Lähetetään HttpPost-metodilla lisäyspyyntö ja Body-osiossa lähetettävät tiedot.

```
65 | :| [Fact]
66 | :| | 0 references
67 | :| public async Task AddNew()
68 | :| {
69 | :|     var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
70 | :|     using var system = new AlbaHost(hostBuilder);
71 | :|     {
72 | :|         var results1 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/painlocation");
73 | :|         var count1 = results1.Count();
74 | :| 
75 | :|         var location = new PainLocation
76 | :|         { LocationName = "TestingAlba" };
77 | :| 
78 | :|         await system.Scenario(s =>
79 | :|         {
80 | :|             s.Post.Json<PainLocation>(location).ToUrl("/kipukalenteri/painlocation");
81 | :|             s.StatusCodeShouldBe(200);
82 | :|         });
83 | :| 
84 | :|         var results2 = await system.GetAsJson<IEnumerable<Note>>("/kipukalenteri/painlocation");
85 | :|         Assert.Equal(count1 + 1, results2.Count());
86 | :|     }
87 | :| }
```

Odotettu tulos: Lisäys onnistuu ja vastauksena saadaan status-koodi 200 OK. PainLocation-taulun tulosten määrä on lisääntynyt yhdellä ennen lisäystä lasketusta määrästä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.2.5 Delete() -metodin testaus

Poistetaan viimeisimmäksi lisätty rivi tietokantataulusta HttpDelete-metodilla.

```

88     [Fact]
89     public async Task DeleteLatest()
90     {
91         var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
92         using var system = new AlbaHost(hostBuilder);
93
94
95         var results1 = await system.GetAsJson<IEnumerable<PainLocation>>("/kipukalenteri/painlocation");
96         var count1 = results1.Count();
97
98         int biggestId = (from r in results1 select r.LocationId).ToArray().Max();
99
100        await system.Scenario(s =>
101        {
102            s.Delete.Url("/kipukalenteri/painlocation/" + biggestId);
103            s.StatusCodeShouldBe(200);
104        });
105
106        var results2 = await system.GetAsJson<IEnumerable<PainLocation>>("/kipukalenteri/painlocation");
107        Assert.Equal(count1 - 1, results2.Count());
108
109
110    }
111
112 }
113

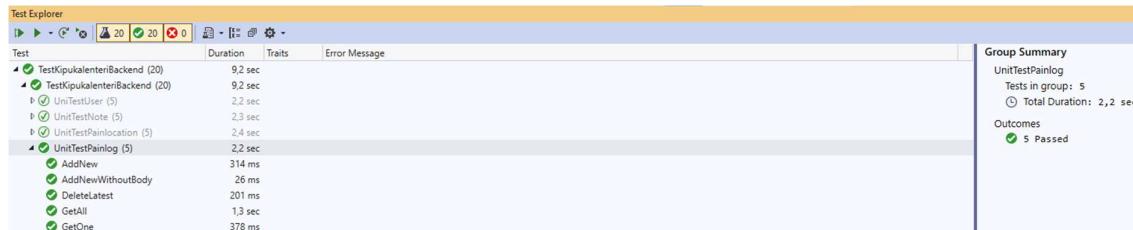
```

Odotettu tulos: Poisto onnistuu, vastauksena saadaan status-koodi 200 OK ja taulun tulosten määrä on vähenyt yhdellä ennen poistoa lasketusta määristä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.3 PainLogControllerin integraatiotestaus

Testitiedosto koostuu viidestä yksittäisestä testistä. Kaikki testit on suoritettu samalla kertaa onnistuneesti, testien tulokset vastasivat alla testitapauskuvauksissa asetettuja odotettuja tuloksia.



3.2.2.3.1 GetAll()-metodin testaus

Testataan, että kontrollerin GetAll()-metodi toimii ja tuloksena saatu tieto ei ole tyhjä

```

public async Task GetAll()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/painlog");
            s.StatusCodeShouldBeOk();
        });

        var results = await system.GetAsJson<IEnumerable<PainLog>>("/kipukalenteri/painlog");
        Assert.NotEmpty(results);
    }
}

```

Odotettu tulos: Tietojen haku GetAll()-metodilla toimii, tuloksena saatu tieto ei ole tyhjä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.3.2 GetOne() -metodin testaus

Testataan, että GetOne()-metodi toimii. Hakua testataan kolmella tavalla:

Haetaan Id:llä, jota tietokantataulussa ei ole

Haetaan Id:llä, joka taulusta löytyy

Haetaan Id:llä, joka on väärän tyypininen

```
[Fact]
| 0 references
public async Task GetOne()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/painlog/75544");
            s.StatusCodeShouldBe(204);

            s.Get.Url("/kipukalenteri/painlog/10114");
            s.StatusCodeShouldBe(200);

            s.Get.Url("/kipukalenteri/painlog/painlog");
            s.StatusCodeShouldBe(400);
        });
    }
}
```

Odottettu tulos: Ensimmäisen kohdan pyynnöstä saadaan vastaukseksi status-koodi 204 No Content, toisen kohdan pyynnöstä saadaan status-koodi 200 OK ja kolmannen kohdan pyynnöstä status-koodi 400 Bad Request.

Tulos: Testi toteutui odotetun tuloksen mukaan, status-koodit vastasivat oletettua.

3.2.2.3.3 AddNewWithoutBody() - Create()-metodin testaus

Lähetetään HttpPost -metodille lisäyspyyntö laittamatta Body-osioon lähetettävän sijainnin tietoja.

```
| 0 references
public async Task AddNewWithoutBody()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Post.Url("/kipukalenteri/painlog");
            s.StatusCodeShouldBe(415);
        });
    }
}
```

Odottettu tulos: Vastausena saadaan status-koodi 415 Unsupported Media Type, lisäys ei onnistu.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.3.4 AddNew() - Create()-metodin testaus

Lähetetään HttpPost-metodilla lisäyspyyntö ja Body-osiossa lähetettäväät tiedot.

```
[Fact]
➊ | 0 references
public async Task AddNew()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {

        var results1 = await system.GetAsJson<IEnumerable<PainLog>>("/kipukalenteri/painlog");
        var count1 = results1.Count();

        var painlog = new PainLog
        {
            PainIntensity = 5,
            StartTime = DateTime.Parse("2022-04-01T09:42:00"),
            EndTime = DateTime.Parse("2022-04-01T09:42:00"),
            Medication = "",
            LocationInfo = "",
            PainTrigger = "",
            PainType = "",
            LocationId = 105,
            Notes = "",
            UserId = "testiad"
        };

        await system.Scenario(s =>
        {
            s.Post.Json<PainLog>(painlog).ToUrl("/kipukalenteri/painlog");
            s.StatusCodeShouldBe(200);
        });

        var results2 = await system.GetAsJson<IEnumerable<PainLog>>("/kipukalenteri/painlog");
        Assert.Equal(count1 + 1, results2.Count());
    }
}
```

Odottettu tulos: Lisäys onnistuu ja vastauksena saadaan status-koodi 200 OK. PainLog-taulun tulosten määrä on lisääntynyt yhdellä ennen lisäystä lasketusta määrästä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.3.5 Delete() -metodin testaus

Poistetaan viimeisimmäksi lisätty rivi tietokantataulusta HttpDelete-metodilla.

```

100
101 [Fact]
102 public async Task DeleteLatest()
103 {
104     var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
105     using var system = new AlbaHost(hostBuilder);
106
107     var results1 = await system.GetAsJson<IEnumerable<PainLog>>("/kipukalenteri/painlog");
108     var count1 = results1.Count();
109
110     int biggestId = (from r in results1 select r.LogId).ToArray().Max();
111
112     await system.Scenario(s =>
113     {
114         s.Delete.Url("/kipukalenteri/painlog/" + biggestId);
115         s.StatusCodeShouldBe(200);
116     });
117
118     var results2 = await system.GetAsJson<IEnumerable<PainLog>>("/kipukalenteri/painlog");
119     Assert.Equal(count1 - 1, results2.Count());
120 }
121
122
123
124
125

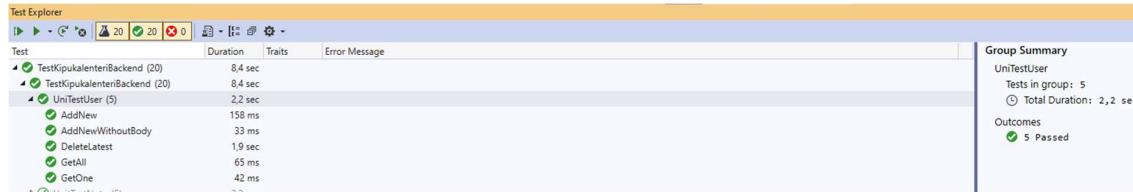
```

Odotettu tulos: Poisto onnistuu, vastauksena saadaan status-koodi 200 OK ja taulun tulosten määrä on vähentynyt yhdellä ennen poistoa lasketusta määristä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.4 UserControllerin integraatiotestaus

Testitiedosto koostuu viidestä yksittäisestä testistä. Kaikki testit on suoritettu samalla kertaa onnistuneesti, testien tulokset vastasivat alla testitapaikkuvaussissa asetettuja odotettuja tuloksia.



3.2.2.4.1 GetAll()-metodin testaus

Testataan, että kontrollerin GetAll()-metodi toimii ja tuloksena saatu tieto ei ole tyhjä

```
[Fact]
❶ | 0 references
public async Task GetAll()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/user");
            s.StatusCodeShouldBeOk();
        });

        var results = await system.GetAsJson<IEnumerable<User>>("/kipukalenteri/user");
        Assert.NotEmpty(results);
    }
}
```

Odotettu tulos: Tietojen haku GetAll()-metodilla toimii, tuloksena saatu tieto ei ole tyhjä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.4.2 GetOne() -metodin testaus

Testataan, että GetOne()-metodi toimii. Hakua testataan kahdella tavalla:

Haetaan Id:llä, jota tietokantataulussa ei ole

Haetaan Id:llä, joka taulusta löytyy

```
[Fact]
❶ | 0 references
public async Task GetOne()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Get.Url("/kipukalenteri/user/7554");
            s.StatusCodeShouldBe(204);

            s.Get.Url("/kipukalenteri/note/anna1");
            s.StatusCodeShouldBe(200);
        });
    }
}
```

Odotettu tulos: Ensimmäisen kohdan pyynnöstä saadaan vastaukseksi status-koodi 204 No Content ja toisen kohdan pyynnöstä saadaan status-koodi 200 OK.

Tulos: Testi toteutui odotetun tuloksen mukaan, status-koodit vastasivat oletettua.

3.2.2.4.3 AddNewWithoutBody() - Create()-metodin testaus

Lähetetään HttpPost -metodille lisäyspyyntö laittamatta Body-osioon lähetettävän sijainnin tietoja.

```
[Fact]
➊ | 0 references
public async Task AddNewWithoutBody()
{
    var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
    using var system = new AlbaHost(hostBuilder);
    {
        await system.Scenario(s =>
        {
            s.Post.Url("/kipukalenteri/user");
            s.StatusCodeShouldBe(415);
        });
    }
}
```

Odotettu tulos: Vastausena saadaan status-koodi 415 Unsupported Media Type, lisäys ei onnistu.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.4.4 AddNew() - Create()-metodin testaus

Lähetetään HttpPost-metodilla lisäyspyyntö ja Body-osiossa lähetettävät tiedot.

```
61   [Fact]
62   ➊ | 0 references
63   public async Task AddNew()
64   {
65       var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
66       using var system = new AlbaHost(hostBuilder);
67       {
68           var results1 = await system.GetAsJson<IEnumerable<User>>("/kipukalenteri/user");
69           var count1 = results1.Count();
70
71           var user = new User
72           { Username = "AlbaUser", Email = "Alba@testi", Password = "Alba1" };
73
74           await system.Scenario(s =>
75           {
76               s.Post.Json<User>(user).ToUrl("/kipukalenteri/user");
77               s.StatusCodeShouldBe(200);
78           });
79
80           var results2 = await system.GetAsJson<IEnumerable<User>>("/kipukalenteri/user");
81           Assert.Equal(count1 + 1, results2.Count());
82       }
83 }
```

Odotettu tulos: Lisäys onnistuu ja vastausena saadaan status-koodi 200 OK. Users-taulun tulosten määrä on lisääntynyt yhdellä ennen lisäystä lasketusta määrästä.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.2.4.5 Delete() -metodin testaus

Poistetaan viimeisimmäksi lisätty rivi tietokantataulusta HttpDelete-metodilla.

```

84 [Fact]
85     public async Task DeleteLatest()
86     {
87         var hostBuilder = Kipukalenteri.Program.CreateHostBuilder(new string[0]);
88
89         using var system = new AlbaHost(hostBuilder);
90         {
91
92             var results1 = await system.GetAsJson<IEnumerable<User>>("/kipukalenteri/user");
93             var count1 = results1.Count();
94
95             await system.Scenario(s =>
96             {
97                 s.Delete.Url("/kipukalenteri/user/" + "AlbaUser");
98                 s.StatusCodeShouldBe(200);
99             });
100
101             var results2 = await system.GetAsJson<IEnumerable<User>>("/kipukalenteri/user");
102             Assert.Equal(count1 - 1, results2.Count());
103         }
104     }
105 }
106 }
107 }
```

Odotettu tulos: Poisto onnistuu, vastauksena saadaan status-koodi 200 OK ja taulun tulosten määrä on vähentynyt yhdellä ennen poistoa lasketusta määristää.

Tulos: Testi toteutui odotetun tuloksen mukaisesti.

3.2.3 End-to-end testaus Cypress-kirjastolla

f

3.2.3.1 Muistiinpanot-näkymän testaus

Testitiedosto koostuu kolmesta osasta.

Ennen jokaista testiä suoritetaan beforeEach -osuuus eli siirrytään sivulle

<http://localhost:3000>Note>

```
cypress > integration > js note.spec.js > ...
1   describe('Note', function () {
2       beforeEach(function () {
3           cy.visit('http://localhost:3000>Note')
4       })
5   })
```

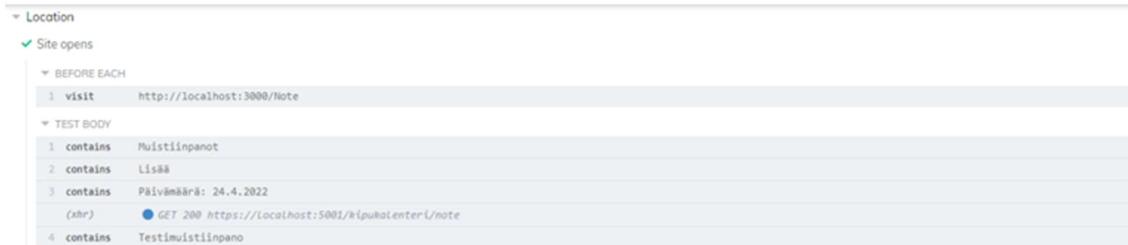
3.2.3.1.1 Sivu avautuu ja renderöi tietokantataulun sisällön

Testataan, että Muistiinpanot-sivu renderöi "Muistiinpanot" -otsikon lisäksi tietokannasta dataa sivulle.

```
6       it('Site opens', function() {
7           cy.contains('Muistiinpanot')
8           cy.contains('Lisää')
9           cy.contains('Päivämäärä: 24.4.2022')
10          cy.contains('Testimuistiinpano')
11      })
```

Odotettu tulos: Sivu avautuu ja renderöi määritellyt tiedot.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu avautuu oikein ja renderöi myös tietokannan dataa.



```

    1 visit http://localhost:3000/Note
    2 contains Muistiinpanot
    3 contains Lisää
    4 contains Päivämäärä: 24.4.2022
    5 contains GET 200 https://localhost:5001/kipukalenteri/note
    6 contains Testimuistiinpano
  
```

3.2.3.1.2 Muistiinpanon lisäys toimii

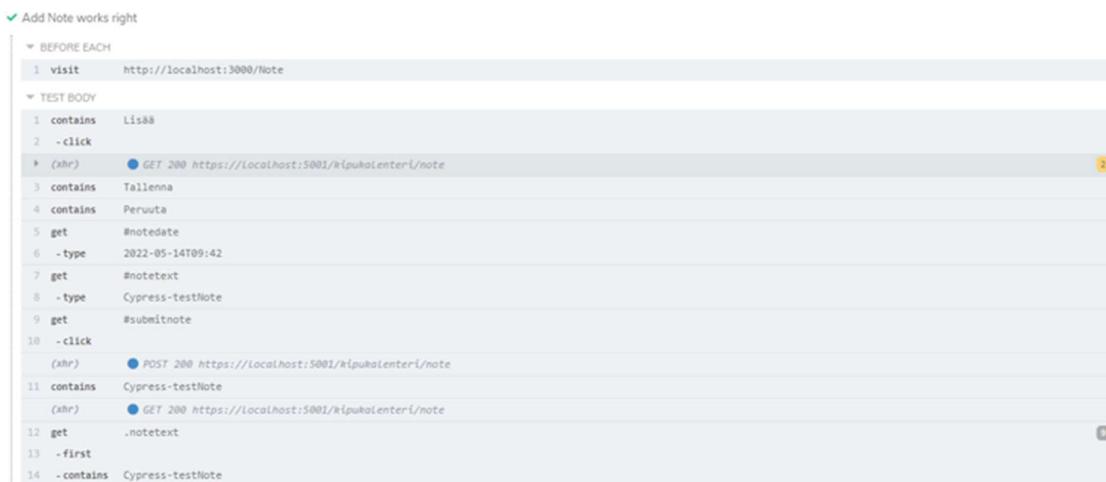
Testataan, että muistiinpanon lisäys toimii, lisätty muistiinpano siirryt tietokantaan ja lisätty tiedot renderöityvät sivulle.

```

13     it('Add Note works right', function () {
14       cy.contains('Lisää').click()
15       cy.contains('Tallenna')
16       cy.contains('Peruuta')
17       cy.get('#notedate').type('2022-05-14T09:42')
18       cy.get('#notetext').type('Cypress-testNote')
19       cy.get('#submitnote').click()
20       cy.contains('Cypress-testNote')
21       cy.get('.notetext').first().contains('Cypress-testNote')
22     })
23   }
  
```

Odotettu tulos: Muistiinpanon lisäys toimii, lisätty muistiinpano siirryt tietokantaan ja lisätty tiedot renderöityvät sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, muistiinpanon lisäys toimii.



```

    1 visit http://localhost:3000/Note
    2 contains Lisää
    3 -click
    4 (xhr) GET 200 https://localhost:5001/kipukalenteri/note
    5 contains Tallenna
    6 contains Peruuta
    7 get #notedate
    8 -type 2022-05-14T09:42
    9 get #notetext
    10 -type Cypress-testNote
    11 get #submitnote
    12 -click
    13 (xhr) POST 200 https://localhost:5001/kipukalenteri/note
    14 contains Cypress-testNote
    15 (xhr) GET 200 https://localhost:5001/kipukalenteri/note
    16 get .notetext
    17 -first
    18 -contains Cypress-testNote
  
```

3.2.3.1.3 Muistiinpanon poisto toimii

Testataan, että muistiinpanon poisto toimii, eikä poistettu muistiinpano enää renderöidy sivulle.

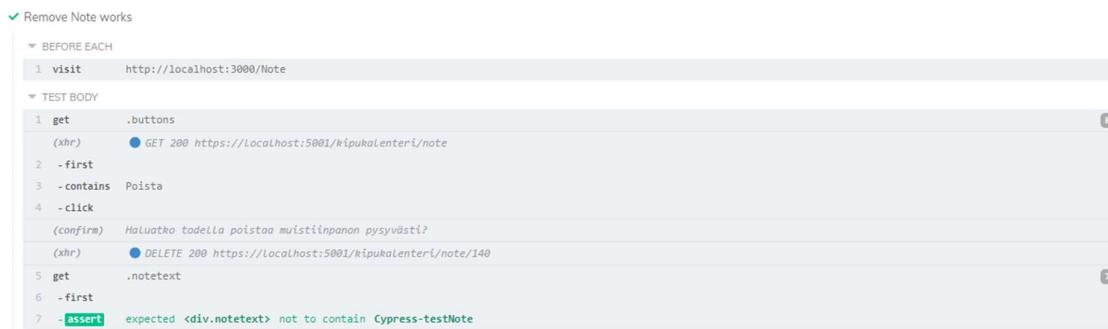
```

24     it('Remove Note works', function () {
25       cy.get('.buttons').first().contains('Poista').click()
26       cy.get('.notetext').first().should('not.contain', 'Cypress-testNote')
27     })
28   })
29 }

```

Odotettu tulos: Muistiinpanon poisto toimii, eikä poistettu muistiinpano enää renderöidy sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti ja poisto toimi.



```

✓ Remove Note works
  ✓ BEFORE EACH
    ✓ visit http://localhost:3000/Note
      ✓ TEST BODY
        ✓ get .buttons
          (xhr) ● GET 200 https://localhost:5001/kipukalenteri/note
        ✓ -first
        ✓ -contains Poista
        ✓ -click
          (confirm) Haluatko todella poistaa muistiinpanon pysyvästi?
          (xhr) ● DELETE 200 https://localhost:5001/kipukalenteri/note/140
        ✓ get .notetext
        ✓ -first
        ✓ -assert expected <div.notetext> not to contain Cypress-testNote

```

3.2.3.2 Sijainnit-näkymän testaus

Testitiedosto koostuu kolmesta osasta.

Ennen jokaista testiä suoritetaan beforeEach -osuuus eli kirjaudutaan sisään ja siirrytään sivulle <http://localhost:3000/Location>

```

6   beforeEach(function () {
7     cy.visit('http://localhost:3000/')
8     cy.get('#hpuserinput').type('testiad')
9     cy.get('#hpuserpass').type('Kaapeli01')
10    cy.get('#submitlogin').click()
11    cy.wait(500)
12    cy.visit('http://localhost:3000/Location')
13  })

```

3.2.3.2.1 Sivu avautuu ja renderöi tietokantataulun sisällön

Testataan, että Sijainnit-sivu renderöi "Kivun sijainti" -otsikon lisäksi tietokannasta dataa sivulle.

```

15  it('Site opens', function() {
16    cy.contains('Kivun sijainti')
17    cy.contains('ID 104')
18    cy.contains('Pää')
19  })

```

Odotettu tulos: Sivu avautuu ja renderöi määritellyt tiedot.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu avautuu oikein ja renderöi myös tietokannan dataa.

The screenshot shows the Cypress Test Runner interface with a test file named 'cypress/integration/location.spec.js'. The test suite is named 'Location' and contains a single test case 'Site opens'. The 'BEFORE EACH' block contains code to visit the local host, enter user credentials, and click a button, which results in an uncaught exception due to a missing function. The 'TEST BODY' block contains assertions to check if the location was added correctly.

```
1 visit      http://localhost:3000/
2 get        #hpuserinput
3 -type      testiad
4 get        #hpuserpass
5 -type      Kaapeli01
6 get        #submitlogin
7 -click
  (xhr)    ● POST https://kipukalenteriapi.azurewebsites.net/kipukalenteri/authentication
8 wait      500
  (uncaught exception) TypeError: setIsPositive is not a function
9 visit      http://localhost:3000/Location

1 contains   Kivun sijainti
  (xhr)    ● GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLocation
2 contains   ID 104
3 contains   Pää
```

3.2.3.2.2 Sijainnin lisäys toimii

Testataan, että sijainnin lisäys toimii, lisätty sijainti siirtyy tietokantaan ja lisättyt tiedot renderöityvät sivulle.

```
21  it('Add Location works right', function () {
22    cy.wait(500)
23    cy.get('.nappi').click()
24    cy.wait(500)
25    cy.contains('Lisää sijainti')
26    cy.get('#locationInput').type('Cypress-testLocation')
27    cy.get('#submitlocation').click()
28    cy.wait(500)
29    cy.get('.notepage').last().contains('Cypress-testLocation')
30  })
```

Odottu tulos: Sijainnin lisäys toimii, lisätty sijainti siirtyy tietokantaan ja lisättyt tiedot renderöityvät sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sijainnin lisäys toimii.

The screenshot shows the Cypress Test Runner interface with a single green checkmark indicating success. The test name is 'Add Location works right'. The test script is as follows:

```
1 visit      http://localhost:3000/
2 get        #hpuuserinput
3 -type     testiad
4 get        #hpuuserpass
5 -type     Kapeliai01
6 get        #submitlogin
7 -click
  (xhr)    POST https://kipukalenteriapi.azurewebsites.net/kipukalenteri/authentication
8 wait      500
  (uncaught exception) TypeError: setIsPositive is not a function
9 visit      http://localhost:3000/Location

▼ TEST BODY
10 wait     500
  (xhr)   GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
11 get      .nappi
12 -click
  (xhr)   GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
13 wait     500
14 contains Lisää sijainti
15 get      #locationInput
16 -type     Cypress-testlocation
17 get      #submitlocation
18 -click
  (xhr)   POST https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
19 wait     500
  (xhr)   GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
20 get      .notepage
21 -last
22 -contains Cypress-testlocation
```

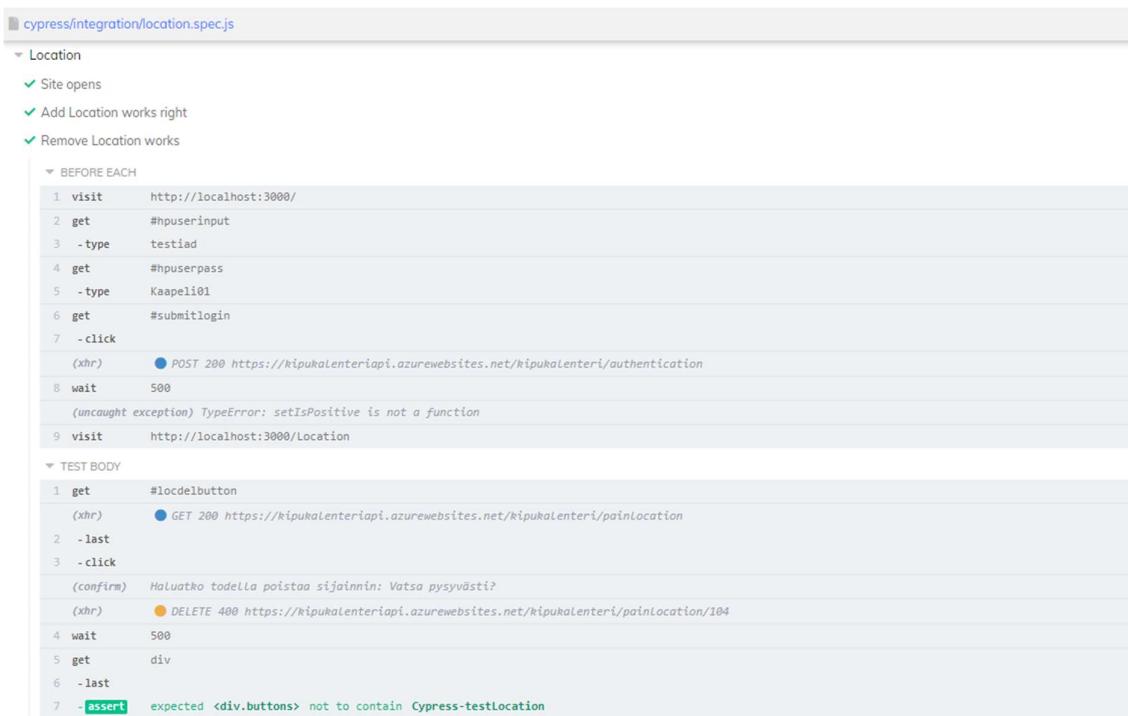
3.2.3.2.3 Sijainnin poisto toimii

Testataan, että sijainnin poisto toimii, eikä poistettu sijainti enää renderöidy sivulle.

```
32  it('Remove Location works', function () {
33    cy.get('#locdelbutton').last().click()
34    cy.wait(500)
35    cy.get('div').last().should('not.contain', 'Cypress-testLocation')
36  })
37
38 })
```

Odottettu tulos: Sijainnin poisto toimii, eikä poistettu sijainti enää renderöidy sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti ja poisto toimi.



The screenshot shows the Cypress Test Runner interface with a test file named 'cypress/integration/location.spec.js'. The test structure is as follows:

- Location** (suite)
 - Site opens** (test)
 - Add Location works right** (test)
 - Remove Location works** (test)
- BEFORE EACH** (block)
 - visit <http://localhost:3000/>
 - get #hpuserinput
 - type testiad
 - get #hpuserpass
 - type Kaapeli01
 - get #submitlogin
 - click
 - (xhr) POST 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/authentication
 - wait 500
 - (uncaught exception) TypeError: setIsPositive is not a function
 - visit <http://localhost:3000/Location>
- TEST BODY** (block)
 - get #locdelbutton
 - (xhr) GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLocation
 - last
 - click
 - (confirm) Haluatko todella poistaa sijainnin: Vatsa pysyvästi?
 - (xhr) DELETE 400 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLocation/104
 - wait 500
 - get div
 - last
 - assert expected <div.buttons> not to contain Cypress-testLocation

3.2.3.3 Kipumerkinnät-näkymän testaus

Testitiedosto koostuu kolmesta osasta.

Ennen jokaista testiä suoritetaan beforeEach -osuuksia eli siirrytään sivulle

<http://localhost:3000/Painlog>

```
cypress > integration > js painlog.spec.js > ...
  1  describe('Painlog', function () {
  2    beforeEach(function () {
  3      cy.visit('http://localhost:3000/Painlog')
  4    })
  5  })
```

3.2.3.3.1 Sivu avautuu ja renderöi tietokantataulun sisällön

Testataan, että Kipumerkinnät-sivu renderöi "Kipumerkinnät" -otsikon lisäksi tietokannasta dataa sivulle.

```
  6  it('Site opens', function() {
  7    cy.contains('Kipumerkinnät')
  8    cy.contains('Lisää')
  9    cy.contains('Alkupäivämäärä: 11.5.2022')
 10   cy.contains('Kivun sijainti: Pää')
 11  })
```

Odottettu tulos: Sivu avautuu ja renderöi määritellyt tiedot.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu avautuu oikein ja renderöi myös tietokannan dataa.



```

    ✓ Painlog
      ✓ Site opens
        ✓ BEFORE EACH
          1 visit http://localhost:3000/Painlog
        ✓ TEST BODY
          1 contains Kipumerkinnät
          2 contains Lisää
          3 contains Alkupäivämäärä: 11.5.2022
            (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painlog
          3 (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painLocation
          4 contains Kivun sijainti: Pää
    
```

3.2.3.3.2 Kipumerkinnän lisäys toimii

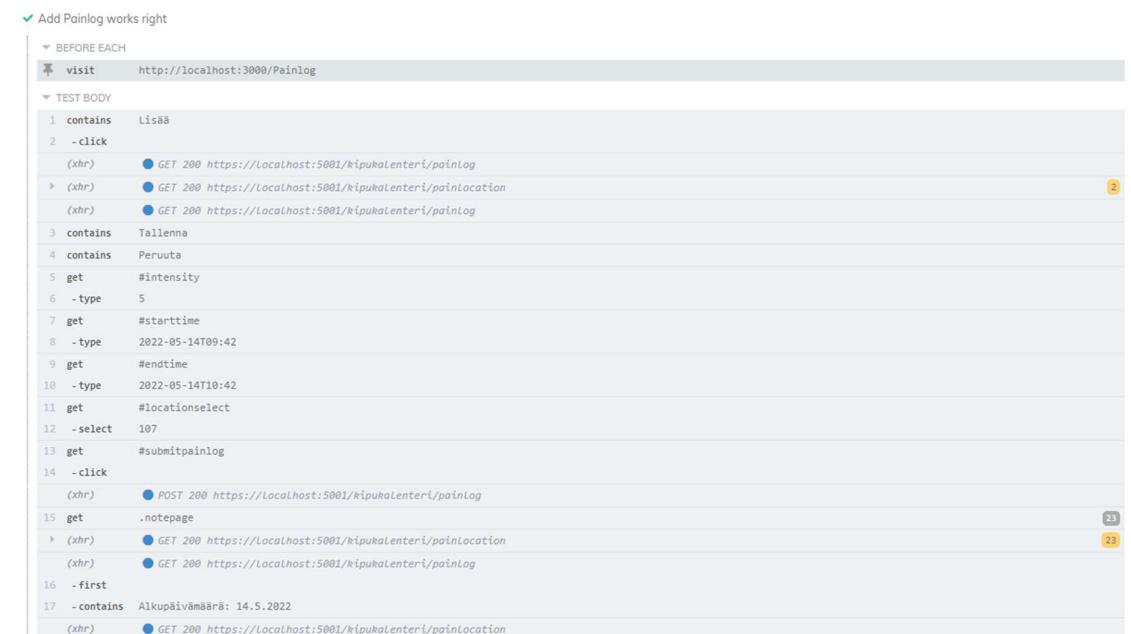
Testataan, että kipumerkinnän lisäys toimii, lisätty kipumerkintä siirrytään tietokantaan ja lisättyt tiedot renderöityvät sivulle.

```

13     it('Add Painlog works right', function () {
14       cy.contains('Lisää').click()
15       cy.contains('Tallenna')
16       cy.contains('Peruuta')
17       cy.get('#intensity').type(5)
18       cy.get('#starttime').type('2022-05-14T09:42')
19       cy.get('#endtime').type('2022-05-14T10:42')
20       cy.get('#locationselect').select('107')
21       cy.get('#submitpainlog').click()
22       cy.get('.notepage').first().contains('Alkupäivämäärä: 14.5.2022 ')
23     })
    
```

Odottettu tulos: Merkinnän lisäys toimii, lisätty merkintä siirrytään tietokantaan ja lisättyt tiedot renderöityvät sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, kipumerkinnän lisäys toimii.



```

    ✓ Add Painlog works right
      ✓ BEFORE EACH
        1 visit http://localhost:3000/Painlog
      ✓ TEST BODY
        1 contains Lisää
        2 -click
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painlog
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painlocation
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painLog
        3 contains Tallenna
        4 contains Peruuta
        5 get #intensity
        6 -type 5
        7 get #starttime
        8 -type 2022-05-14T09:42
        9 get #endtime
        10 -type 2022-05-14T10:42
        11 get #locationselect
        12 -select 107
        13 get #submitpainlog
        14 -click
          (xhr)   ● POST 200 https://localhost:5001/kipukalenteri/painlog
        15 get .notepage
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painlocation
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painLog
        16 -first
        17 -contains Alkupäivämäärä: 14.5.2022
          (xhr)   ● GET 200 https://localhost:5001/kipukalenteri/painLocation
    
```

3.2.3.3.3 Kipumerkinnän poisto toimii

Testataan, että kipumerkinnän poisto toimii, eikä poistettu merkintä enää renderöidy sivulle.

```
25    it('Remove Painlog works', function () {
26      cy.get('.buttons').first().contains('Poista').click()
27      cy.get('.notepage').first().should('not.contain', 'Käsi')
28    })
29
30  })
```

Odotettu tulos: Merkinnän poisto toimii, eikä poistettu merkintä enää renderöidy sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti ja poisto toimi.

3.2.3.4 Frontpage-näkymän testaus

Testitiedosto koostuu kolmesta osasta.

Ennen jokaista testiä suoritetaan beforeEach -osuuus eli kirjaudutaan sisään

```
6    beforeEach(function () {
7      cy.visit('http://localhost:3000/')
8      cy.get('#hpuserinput').type('testiad')
9      cy.get('#hpuserpass').type('Kaapeli01')
10     cy.get('#submitlogin').click()
11     cy.wait(500)
12   })
```

3.2.3.4.1 Kirjautuminen toimii ja sivu avautuu ja renderöi sisällön

Testataan, että kirjautuminen on onnistunut ja etusivu renderöityy oikein.

```
14    it('Login works and site opens', function() {
15      cy.contains('Kirjautuneena')
16      cy.contains('Lisää kipumerkintä')
17      cy.contains('Lisää sijainti')
18    })
```

Odotettu tulos: Sivulta löytyy teksti "Kirjautuneena" eli kirjautuminen on onnistunut ja sivulta löytyy napit "Lisää kipumerkintä" ja "Lisää sijainti".

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sivu avautuu oikein ja kirjautuminen toimii.

The screenshot shows a Cypress integration test named `frontpage.spec.js`. It contains a single test case under the `Frontpage` suite. The test is titled `Login works and site opens`. It includes a `BEFORE EACH` block where the user logs in with credentials and submits the login form. This results in a `POST` request to the API endpoint `/kipukalenteri/authentication`, which fails with a `500` error due to an uncaught exception: `TypeError: setIsPositive is not a function`. The test then checks the page body for three specific messages: `Kirjautuneena`, `Lisää kipumerkintä`, and `Lisää sijainti`.

```

cy.cypress/integration/frontpage.spec.js
  Frontpage
    Login works and site opens
      BEFORE EACH
        1 visit      http://localhost:3000/
        2 get        #hpuserinput
        3 -type     testiad
        4 get        #hpuserpass
        5 -type     Kaapelil01
        6 get        #submitlogin
        7 -click
        (xhr)      ● POST 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/authentication
        8 wait      500
        (uncaught exception) TypeError: setIsPositive is not a function
      TEST BODY
        1 contains   Kirjautuneena
        2 contains   Lisää kipumerkintä
        3 contains   Lisää sijainti
  
```

3.2.3.4.2 Sijainnin lisäys toimii

Testataan, että sijainnin lisäys toimii, lisätty sijainti siirtyy tietokantaan ja lisättyt tiedot renderöityvät Sijainnit-sivulle.

```

20  it('Add Location works right', function () {
21    cy.get('#hpaddloc').click()
22    cy.contains('Lisää sijainti')
23    cy.get('#locationInput').type('Cypress-testFrontpage')
24    cy.get('#submitlocation').click()
25    cy.wait(500)
26    cy.visit('http://localhost:3000/Location')
27    cy.contains('Postman API test 2')
28    cy.get('.notepage').last().contains('Cypress-testFrontpage')
29  })
  
```

Odottettu tulos: Sijainnin lisäys toimii, lisätty sijainti siirtyy tietokantaan ja lisättyt tiedot renderöityvät sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti, sijainnin lisäys toimii.

The screenshot shows the same Cypress integration test as before, but this time it is marked as passing. The test case `Add Location works right` runs through the same steps: logging in, adding a location, and then checking the results on the `Location` page. The test passes because the location is successfully added to the database and visible on the page.

```

✓ Add Location works right
  BEFORE EACH
    1 visit      http://localhost:3000/
    2 get        #hpuserinput
    3 -type     testiad
    4 get        #hpuserpass
    5 -type     Kaapelil01
    6 get        #submitlogin
    7 -click
    (xhr)      ● POST 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/authentication
    8 wait      500
    (uncaught exception) TypeError: setIsPositive is not a function
  TEST BODY
    1 get        #hpaddloc
    2 -click
    3 contains   Lisää sijainti
    4 get        #locationInput
    5 -type     Cypress-testFrontpage
    6 get        #submitlocation
    7 -click
    (xhr)      ● POST 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
    8 wait      500
    9 visit      http://localhost:3000/Location
    10 contains  Postman API test 2
    (xhr)      ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
    11 get        .notepage
    12 -last
    13 -contains Cypress-testFrontpage
  
```

3.2.3.4.3 Merkinnän lisäys toimii

Testataan, että kipumerkinnän lisäys toimii, lisätty merkintä siirtyy tietokantaan ja lisätty tiedot renderöityvät Kipumerkinnät-sivulle.

```

31      it('Add Painlog works right', function () {
32          cy.get('#hpaddpainlog').click()
33          cy.contains('Lisää kipumerkintä')
34          cy.get('#intensity').type(5)
35          cy.get('#starttime').type('2022-06-05T09:42')
36          cy.get('#endtime').type('2022-06-05T10:42')
37          cy.get('#locationselect').select('105')
38          cy.get('#submitpainlog').click()
39          cy.wait(500)
40          cy.visit('http://localhost:3000/Painlog')
41          cy.get('.notepage').first().contains('Alkupäivämäärä: 5.6.2022 ')
42          cy.get('.notepage').first().contains('Pää')
43      })

```

Odottettu tulos: Sijainnin poisto toimii, eikä poistettu sijainti enää renderöidy sivulle.

Tulos: Testi toteutui odotetun tuloksen mukaisesti ja poisto toimi.

```

✓ Add Painlog works right
  > BEFORE EACH
    > TEST BODY
      1 get      #hpaddpainlog
      2  -click
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLocation
      3 contains Lisää kipumerkintä
      4 get      #intensity
      5 -type   5
      6 get      #starttime
      7 -type   2022-06-05T09:42
      8 get      #endtime
      9 -type   2022-06-05T10:42
      10 get     #locationselect
      11 -select 105
      12 get     #submitpainlog
      13 -click
      (xhr)   ● POST 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      14 wait   500
      15 visit  http://localhost:3000/Painlog
      16 get   .notepage
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
      (xhr)   ● GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      > (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlog
      > (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlog
      > (xhr)   ● GET 200 https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painlocation
      (xhr)   ● GET https://kipukalenteriapi.azurewebsites.net/kipukalenteri/painLog
      17 -first
      18 -contains Alkupäivämäärä: 5.6.2022
      19 get   .notepage
      20 -first
      21 -contains Pää

```

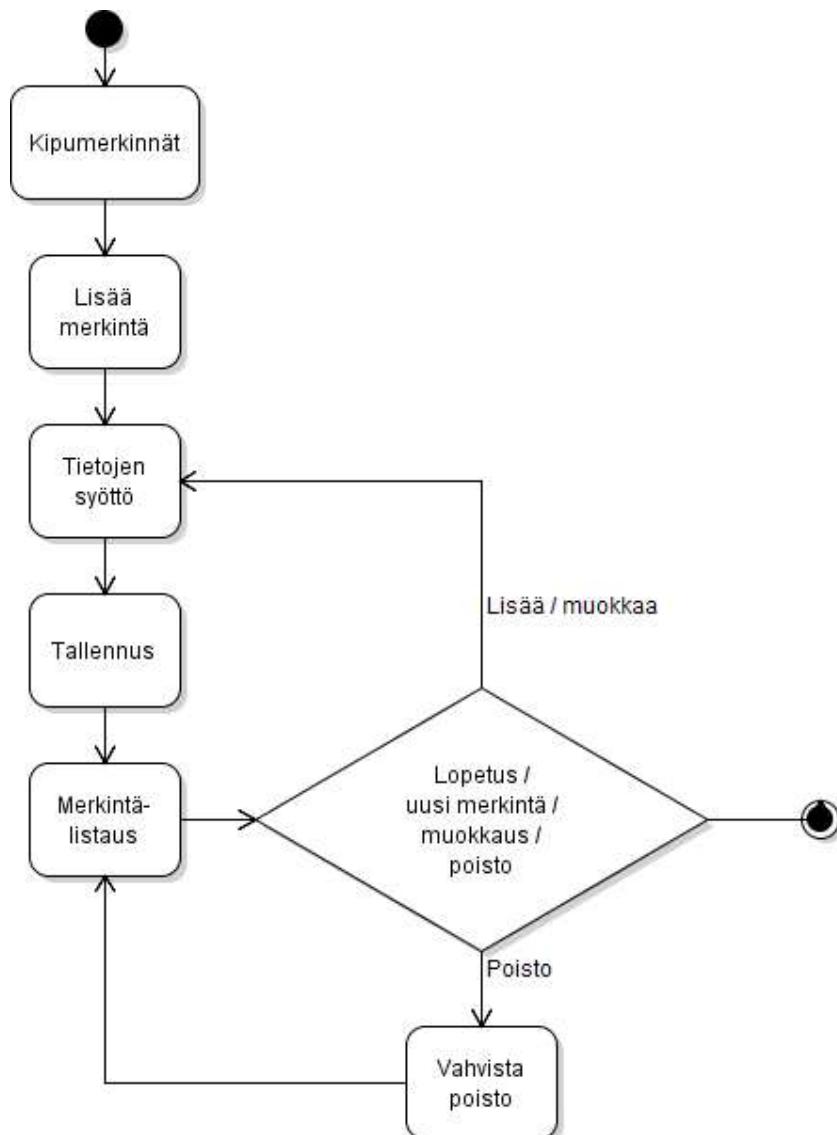
3.3 Hyväksymistestaus

Testauksen perusteella sovellus vastaa määrittelykuvastossa määritettyjä ja toteutettavaksi valittuja käyttötapauskuvauksia eikä siinä ole olennaisia toimintaan vaikuttavia puutteita.

3.3.1 Use case 1: Kipumerkinnän tallennus, muokkaus ja poisto

Testataan, onnistuuiko kipumerkinnän tallennus määrittelykuvaoston käyttötapaus 1:n kuvaoksen mukaisesti.

Kuvaus: Käyttäjä haluaa tallentaa merkinnän kivusta. Hänen valitsee Kipumerkinnät-sivulta "Lisää merkintä". Lisättynä tarpeelliset tiedot hänen painaa "Tallenna". Järjestelmä siirryy automaattisesti merkintälistaukseen, jossa mm. kiven päivämäärä, kesto ja kiven intensiteetti. Käyttäjä voi halutessaan lisätä näkymästä uuden kipumerkinnän tai muokata tai poistaa aiempia merkintöjä.



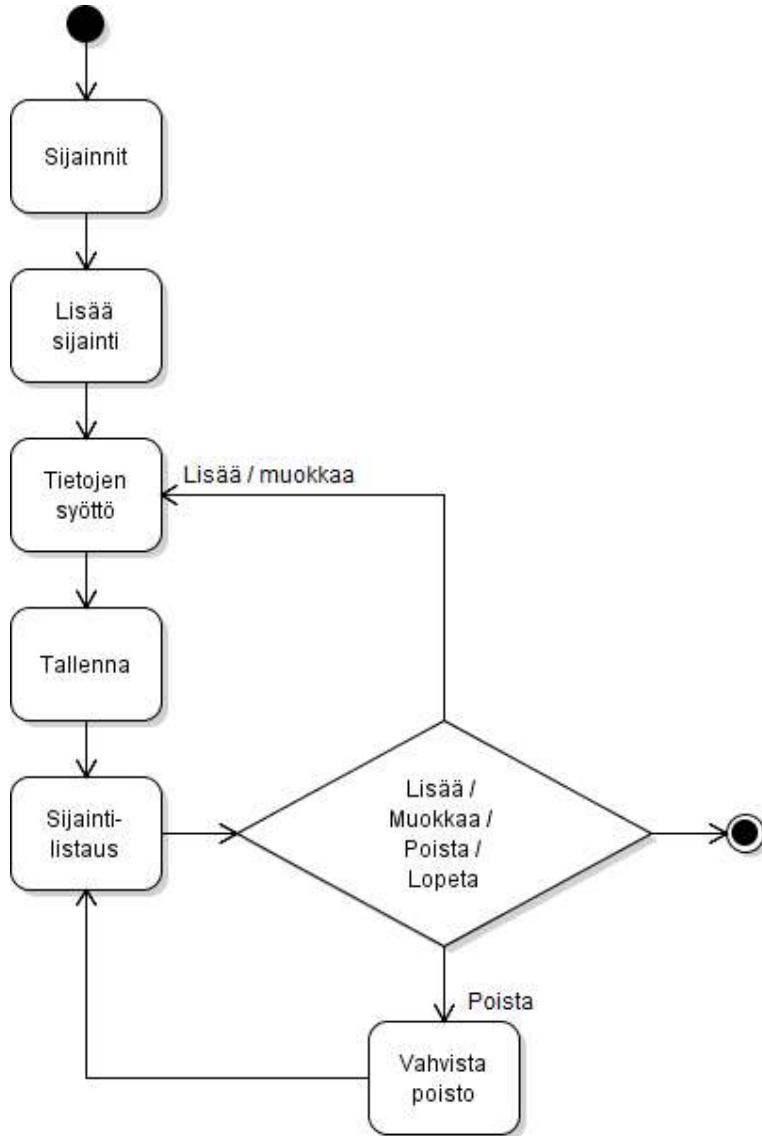
Odottettu tulos: Tiedon tallennus onnistuu olennaisilta osiltaan käyttötapauskuvaoksen mukaisesti.

Tulos: Testin tulos vastaa odottettua tulosta. Kipumerkinnän tallennus tapahtuu käyttötapauskuvaoksen ja -kaavion mukaisesti.

3.3.2 Use case 2: Kivun sijainnin lisäys

Testataan, onnistuuko sijainnin tallennus määrittelykuvaston käyttötapaus 1:n kuvaukseen mukaisesti.

Kuvaus: Käyttäjä haluaa lisätä kivulle uuden sijainnin. Hänen valitsee Sijainti-sivulta "Lisää sijainti" ja syöttää sijainnin nimen. Kun käyttäjä painaa tallenna, tieto siirtyy tietokantaan ja sovellus siirtyy sijaintien listaukseen. Käyttäjä voi lisätä uuden sijainnin, muokata tai poistaa aiemmin lisättyjä sijainteja tai lopettaa sovelluksen käytön.



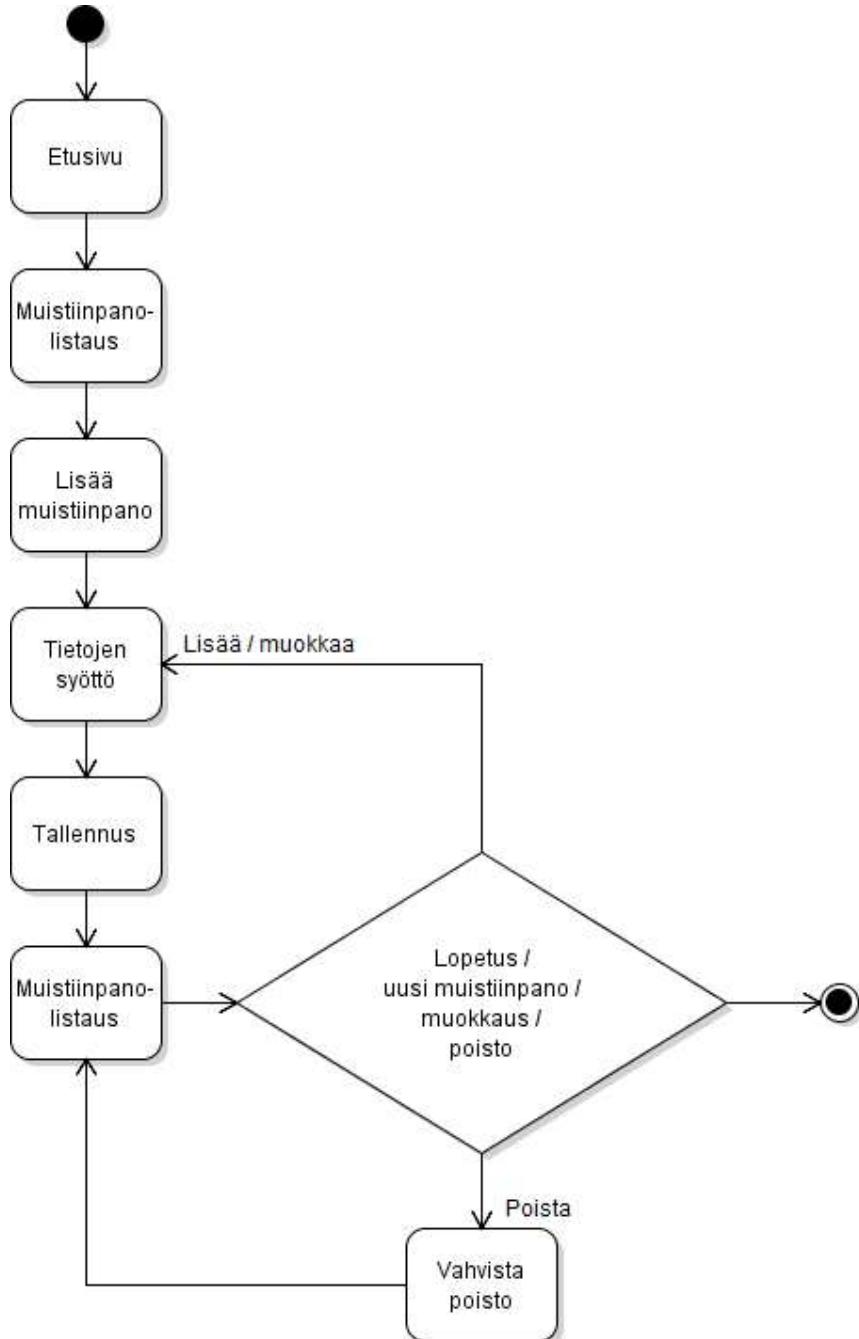
Odotettu tulos: Tiedon tallennus onnistuu olennaisilta osiltaan käyttötapauskuvaukseen mukaisesti.

Tulos: Testin tulos vastaa odotettua tulosta. Sijainnin tallennus tapahtuu käyttötapauskuvaukseen ja -kaavion mukaisesti.

3.3.3 Use case 3: Erillisen muistiinpanon lisäys

Testataan, onnistuuko sijainnin tallennus määrittelykuvaston käyttötapaus 1:n kuvaukseen mukaisesti.

Kuvaus: Käyttäjä haluaa lisätä uuden muistiinpanon. Hän siirtyy muistiinpanot -sivulle ja valitsee "Lisää", syöttää muistiinpanon tiedot ja tallentaa.



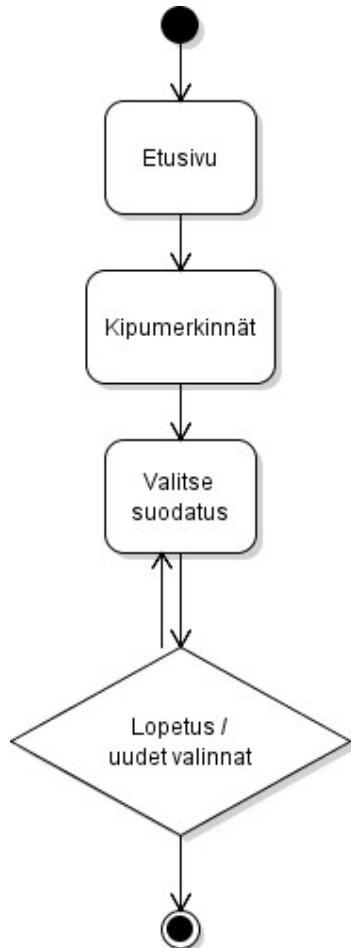
Odottettu tulos: Tiedon tallennus onnistuu olennaisilta osiltaan käyttötapauskuvaukseen mukaisesti.

Tulos: Testin tulos vastaa odotettua tulosta. Muistiinpanon tallennus tapahtuu käyttötapauskuvauksen ja -kaavion mukaisesti.

3.3.4 Use case 4: Kipumerkintöjen suodatus

Testataan, onnistuuko sijainnin tallennus määrittelykuvaston käyttötapaus 1:n kuvaukseen mukaisesti.

Kuvaus: Käyttäjä haluaa katsoa kipuraporttia. Hän valitsee valikosta haluamansa ajan (kuusi kuukautta, kuukausi) ja/tai kivun sijainnin. Sovellus näyttää tehdyt merkinnät halutulta ajalta ja/tai sijainnista. Käyttäjä voi valita uuden suodatuksen tai lopettaa käytön.



Odottettu tulos: Tietojen suodattaminen onnistuu joko aikaa tai sijaintia tai molempia suodattamalla.

Tulos: Testin tulos vastaa odotettua tulosta, suodattaminen toimii vaatimusten mukaan.