

KittenGrowthCurve -ohjelmisto

Helsinki 3.1.2021
Ohjelmistotekniikka, syksy 2020
Maria Myrskyvuori, 015315065
HELSINGIN YLIOPISTO

Sisältö

Vaatimusmäärittely	1
Päivitetty vaatimusmäärittely	1
Yleiskuvaus	1
Käyttäjät	1
Toiminnalliset vaatimukset	1
Projektisuunnitelma	2
Tarvekartoitus	2
Käyttöohje	3
Asennusohje	3
Käyttöohje	4
Yleistä	4
Perusnäky	5
Virhetilanteet	7
Ohjelman rajoitteet	7
Rakenteen ja toiminnan kuvaus	7
Käyttöliittymä	8
Sovelluslogiikka	8
Tietokanta	8
UML-pakkausrakennekaavio	9
Sovelluslogiikka	10
Sovelluslogiikan tarkempi kuvaus	10
Luokkakaavio	12
Sekvenssikaavio: pennun lisääminen ja tietojen muokkaaminen	13
Sekvenssikaavio: pennun punnitseminen	14
Testaus	15
Testausselostus	15
Itsereflektio	17
Omaa pohdintaa	17
Työn eteneminen iteraatioittain	19
Työpäiväkirja	22

Liitteet

Liite 1: Luokkakaavio

README, harjoitustehtävä Laskuri ja vertaisarviot löytyvät ohjelmiston dokumenttikansioista.

Vaatimusmäärittely

Päivitetty vaatimusmäärittely

Yleiskuvaus

Ohjelmisto on työkalu kissankasvattajille. Ohjelmisto seuraa pentujen painon kehitystä luovutusikään (14vk) saakka ja luo pennuille kasvukäyrät, joiden avulla painon kehitystä voi tarkastella yksittäin sekä pentueittain.

Ohjelmistoon luodaan pentueita, joita käsitellään jatkossa yksittäin. Luontivaiheessa pentueelle syötetään astutuspäivä ja tämän avulla ohjelmisto arvioi päivän synnytykselle ja luovutusajankohdalle. Synnytyksen jälkeen synnytyspäivä tarvittaessa korjataan oikeaksi ja luovutusajankohta tarkentuu tämän mukaan. Tässä vaiheessa ohjelmistoon syötetään tiedot syntyneistä pennuista, jonka jälkeen pennuille on mahdollista lisätä painotietoja. Jokaisen punnituksen yhteydessä, uusi painotieto syötetään ohjelmistoon ja tieto päivittyy painokäyrään. Lisäksi pentueella on päiväkirja, jonne voi lisätä muistiinpanoja pentujen kehityksestä. Päiväkirja laskee pentujen iän automaattisesti jokaiseen merkintään.

Käyttäjät

Ohjelmalla on yksi käyttäjä, pentueen kasvattaja.

Toiminnalliset vaatimukset

Ohjelman täytyy toteuttaa seuraavat vaatimukset:

- Pentueen tietojen tallentaminen
- Pentueen tietojen tarkasteleminen
- Syntymäajan sekä luovutuspäivän laskeminen astutuspäivän perusteella
- Pennun iän laskeminen
- Pennun tietojen tallentaminen
- Pennun tietojen tarkasteleminen
- Kasvukäyrän tuottaminen jokaiselle pennulle
- Kasvukäyrien vertailu yksittäin sekä pentueittain
- Päiväkirjamerkintöjen tallentaminen
- Päiväkirjamerkintöjen tarkasteleminen
- Virheellisten tallennusten muokkaaminen
- Tallennusten poistaminen

Projektisuunnitelma

Tarvekartoitus

Ajatus ohjelmiston kehittämisestä syntyi keskusteluista viiden kissankasvattajan kanssa. Kävi ilmi, että painon seuranta on oleellinen osa kasvatustyötä, mutta kasvattajilla ei ole tällä hetkellä siihen sopivaa työvälinettä. Moni kasvattaja on kynän ja ruutupaperin varassa ja yksinkertainen painon seurantaan tarkoitettu ohjelmisto sekä tietojen tallennustapa on tervetullut.

Suunnittelen ohjelmiston rakenteen.

- luokkakaavio
- tietokantakaavio
- käyttöliittymä

Ensimmäisessä iteraatiossa luodaan toiminto pentueen lisäämiseksi ja pentueiden listaus. Vähintään yksi testi.

Toisessa iteraatiossa luodaan toiminto pentujen lisäämiseen. Lisää testejä.

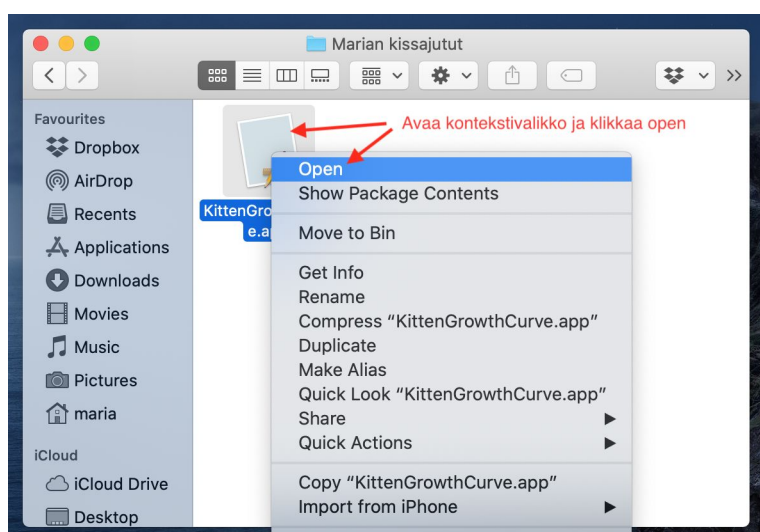
Kolmannessa iteraatiossa kasvukäyrät ja niiden vertailu.

Neljännessä iteraatiossa päiväkirja ja tallennusten muokkaaminen.

Käyttöohje

Asennusohje

Loppukäyttäjälle annetaan ohjelma asennustiedostona KittenGrowthCurve-1.0-SNAPSHOT.dmg. Kaksoisklikkaamalla kuvaketta hiirellä, avautuu kansio, jossa ohjelmatiedosto KittenGrowthCurve.app sijaitsee. Tämä ohjelmatiedosto kopioidaan omalle koneelle, josta se on käynnistettävissä kontekstivalikon kautta. Toistaiseksi helposti asennettava ohjelma on saatavilla vain OSX:lle. Kyseisen tiedoston luontiohje on README:ssa ja tiedosto generoituu target hakemistoon.

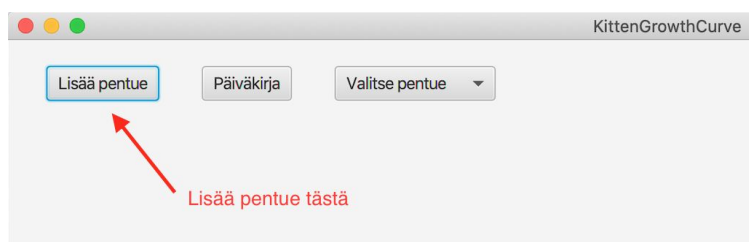


Käyttöohje

Yleistä

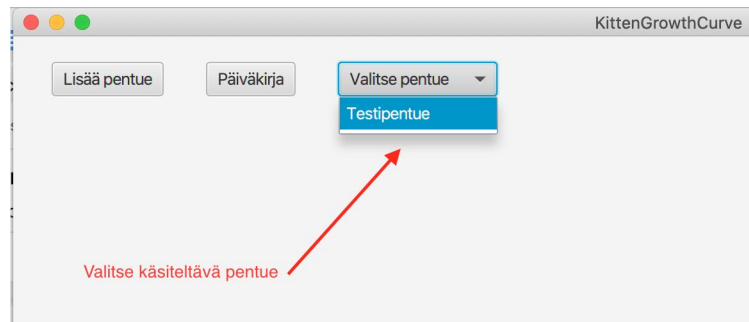
1. Päivämäärät syötetään päivämääräkentän oikeassa reunassa olevan symbolin kautta kalenterinäkymään.
2. Painotiedot syötetään grammoina.
3. Lomakenäkymistä palataan perusnäkymään vasemmassa alakulmassa sijaitsevan Takaisin -painikkeen avulla.

Ensimmäisessä näkymässä on kolme painiketta: Lisää pentue , Päiväkirja ja Valitse pentue.



Lisää pentue -painike avaa lomakkeen pentueen tietojen syöttämistä varten. Syötä pyydetyt tiedot (emo, uros, astutuspäivä ja pentueen nimi) ja paina Tallenna -painiketta. Vasemmassa alakulmassa on painike Takaisin, josta pääsee palaamaan alunäkymään.

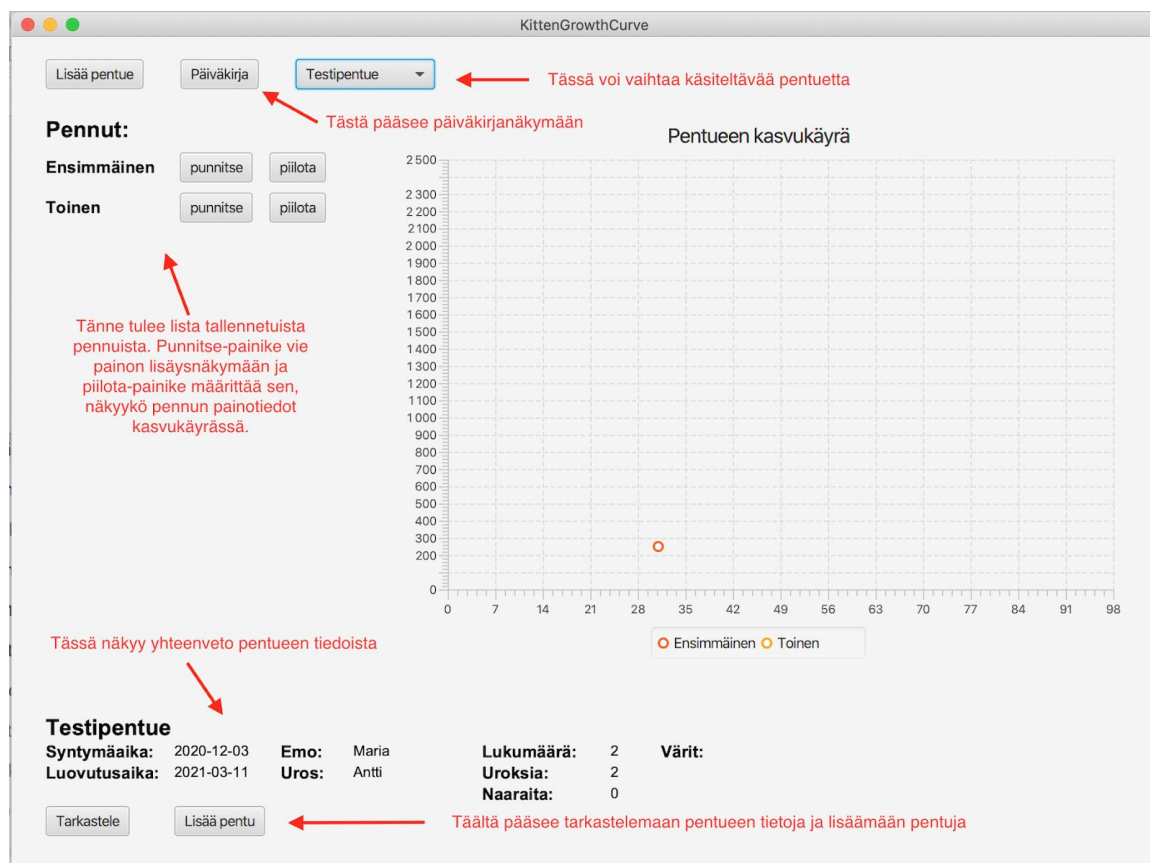
Valitse pentue -painike näyttää tallennetut pentueet ja pentueen nimeä klikkaamalla avautuu valitun pentueen perusnäkö.



Perusnäkö

Perusnäkössä on yläreunassa painikkeet pentueen lisäämisestä ja pentueen valitsemista varten sekä Päiväkirja -painike (3).

Alareunassa näkyy valitun pentueen tiedot. Ennen synnytystä syntymäpäivänä näkyy astutuspäivän perusteella laskettu päivä sekä sen mukaan laskettu luovutuspäivä. Näköm pohjalta löytyvät Tarkastele -painike (1) ja Lisää pentu -painike (2).



1. Tarkastele -painike avaa näkymän pentueen tietojen tarkastelemista ja muokkaamista varten. Täällä voi korjata mahdolliset virheet sekä syöttää toteutuneen syntymäpäivän. Pentujen tallentamisen jälkeen täällä voi myös tarkastella, muokata ja poistaa yksittäisen pennun tietoja. Tallenna -painike tallentaa korjatut tiedot ja takaisin perusnäkömään pääset Takaisin -painikkeen avulla.

2. Lisää pentu -painike avaa näkymän pennun lisäämistä varten. Lomakkeen pakollisia tietoja ovat nimi ja sukupuoli. Tallenna -painike tallentaa pennun ja kertoo onnistuiko tallennus. Nyt voit syöttää tiedot toiselle pennulle tai palata perusnäkömään Takaisin -painikkeen avulla.
3. Päiväkirja -painike avaa näkymän päiväkirjamerkintöjä varten. Syötä teksti sekä merkinnän päivämäärä. Tallenna -painike tallentaa merkinnän. Tallennetut merkinnät näkyvät oikealla ikkunassa. Muokkaa -painike palauttaa merkinnän lomakkeeseen muokattavaksi ja Poista -painike poistaa merkinnän. Päiväkirjamerkintöihin lisätään automaattisesti pentujen ikä päivinä.

Vasemmassa reunassa näkyy tallennetut pennut. Pennulla on punnitse -painike (1) ja piilota/näytä -painike (2).

1. Punnitse -painike avaa näkymän painotietojen lisäämistä varten. Syötä lomakkeeseen punnituspäivä sekä paino grammoina. Lomakkeen alla näkyy syöttämäsi painot ja niiden perässä on poista -painike, jolla voit poistaa virheellisesti tallennetut tiedot. Voit palata perusnäkömään Takaisin -painikkeen avulla.
2. Piilota/näytä -painikkeen avulla määrität sen, näkyykö kyseisen pennun painotiedot painograafissa.

Painograafi näyttää pentueen kasvukäyrät. Pystyakselilla on paino grammoina ja vaaka-akselilla ikä päivinä. Oletuksena graafissa näkyy kaikkien pentueen pentujen kaikki tallennetut painot. Vasemmasta reunasta pentujen nimien yhteydestä piilota/näytä -painikkeella voit piilottaa tai palauttaa näkyviin yksittäisten pentujen kasvukäyrät.

Virhetilanteet

“Tallennus ei onnistunut”

Syötitkö lomakkeeseen kaikki tarvittavat tiedot? Pentue- ja punnituslomake tarvitsevat kaikki pyydettyt tiedot, pennun tallentamiseen riittää nimi ja sukupuoli. Tarkista millaisia ohjeita saat lomakkeen kenttien alla.

Ohjelman rajoitteet

Painograafi näyttää vain yhden datapisteen päivässä, joten syötettäessä päivälle useampia painoja, painograafi näyttää niistä suurimman. Kissanpennun maksimipainoksi on asetettu 2500g ja pentueen info-osioiden koot on optimoitu käsittelemään maksimissaan kymmenen pennun tietoja.

Rakenteen ja toiminnan kuvaus

Ohjelmisto on rakennettu kolmeen kerrokseen:

1. Käyttöliittymä, joka ottaa käyttäjältä syötteitä ja näyttää sovelluksen tilan
2. Seuraavana sovelluslogiikka, joka keskustelee tietokannan kanssa.
3. Alimpana on SQL tietokanta, jossa on taulut Litter, Kitten, Weight ja Diary

Käyttöliittymä

App -luokan ainoa tehtävä on käynnistää ohjelmisto ja luoda luokka UI. Launcher -luokka on olemassa ajettavan jar-tiedoston käynnistämiseksi. Varsinainen käyttöliittymä toimii luokassa UI. UI sisältää ohjelmiston toiminnallisuuden ja sen metodit edustavat omia näkymiään tai päänäkökuvan osia. UI luo luokan KittenService ja kutsuu siltä aina perusnäkökuvaa tultaessa kaiken datan ja lisäksi tarvittaessa toiminnallisiin näkymiin siirryttäessä ja niissä toimiessa näkökuvan toiminnoissa tarvittavan datan, jotta näkymät pysyvät ajantasaisena.

Käyttöliittymälogiikka on pyritty pitämään yksin UI -luokassa, joten näkymien ja lomakkeiden muotoilut on siirretty omiin luokkiinsa. Näkymien muotoilut löytyvät paketista views ja lomakkeiden kenttäluokat paketista formfields. Util -paketissa on näkymien muotoiluun liittyviä apumetodeita, joiden vuoksi ei tuntunut järkevältä luoda erillistä view -luokkaa.

Sovelluslogiikka

KittenService kommunikoi tietokannan kanssa ja huolehtii tallentamisesta, tiedon hakemisesta ja sen päivittämisestä omissa metodeissaan sekä muodostaa oikeanlaista dataa käyttöliittymälle ja kertoo, onnistuiko se tehtävässään. Toimintoja sisältävät erilliset näkymät on myös pidettävä ajantasaisina jokaisen tallennuksen, muokkauksen tai poiston jälkeen.

Tietokanta

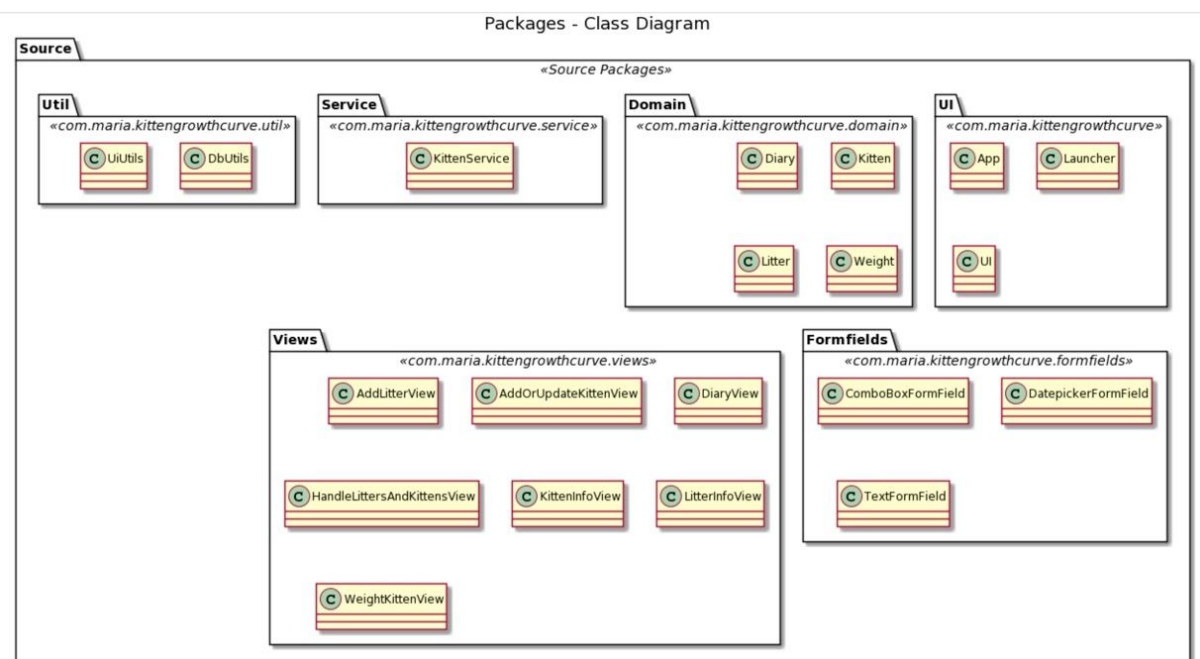
Tietokannan Litter -taulu on keskeisin ja siihen tallennetaan tieto ensimmäisenä. Taulussa on sarakkeet id, litterName, dam, sire, establishment, birth ja delivery. Id syntyy automaattisesti kantaan uuden pentueen tallennuksen yhteydessä. Pentueen nimi, emo, isä ja astutuspäivä on käyttäjän syötettävä pentuetta luotaessa. Astutuspäivän mukaan lasketaan aluksi arvioitu synnytyspäivä ja luovutusaika. Synnytyksen jälkeen käyttäjä voi muokata syntymäpäivän oikeaa vastaavaksi ja luovutuspäivä muuttuu tämän myötä automaattisesti.

Kitten ja Weight -taulut tulevat käyttöön synnytyksen jälkeen pentujen luomisen yhteydessä. Kitten -taulussa on sarakkeet id, litter_id, name, sex, regNo, emsCode ja officialName. Pentu luodaan aina jo olemassa olevalle pentueelle, joten sekä id että litter_id muodostuvat automaattisesti. Käyttäjän on syötettävä pennulle nimi ja sukupuoli, jotka tallentuvat tähän tauluun. Myöhemmin pennulle voi lisätä halutessaan loput tiedot, mutta nämä eivät ole välttämättömiä.

Weight -taulussa on sarakkeet id, kitten_id, weight ja date. Id ja kitten_id syntyvät automaattisesti painotietoja tallennettaessa.

Diary -taulussa on sarakkeet id, litter_id, date ja text. Päiväkirja on liitetty omaan näkymäänsä siirryttäessä valmiiksi pentueeseen, joten käyttäjän on syötettävä päiväkirjamerkintöjä tehdessään vain päivämäärä ja tallennettava teksti.

UML-pakkausrakennekaavio



Sovelluslogiikka

Sovelluslogiikan tarkempi kuvaus

Launcher -luokka käynnistää App -luokan, joka luo ja käynnistää käyttöliittymäluokan UI. UI muodostaa getInitialLayout()-metodin avulla päänäköymän ja pyytää KittenService-luokkaa hakemaan tietokannasta kaiken datan päiväkirjamerkintöjä lukuun ottamatta. KittenService hakee kannasta kaikki pentueet, liittää niihin kaikki pennut ja liittää niihin kaikki tallennetut painotiedot. Tämä tehdään aina päänäköymään tultaessa, jotta tiedot pysyvät ajantasaisina. Aluksi näkyviin tulee kaksi toimivaa painiketta, joista toinen avaa näköymän pentueen lisäämistä varten ja toinen käyttää pentuelistaa tarjotakseen käyttäjälle mahdollisuuden valita yhden tallennetuista pentueista tarkasteltavaksi.

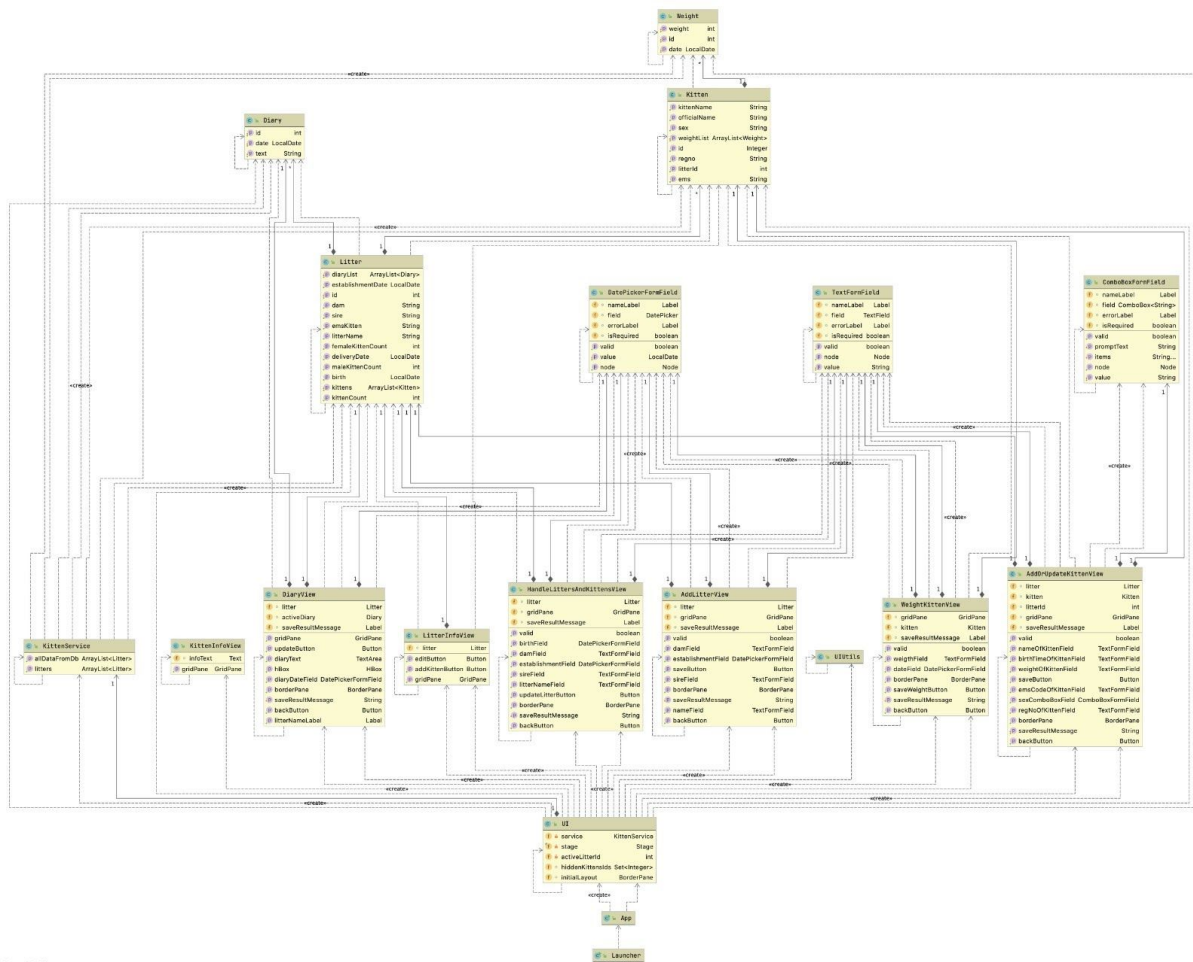
Päänäköymä muodostuu valuttamalla haetut tiedot omiin osioihinsa näköymästä. Yläosassa on edelleen painikkeet pentueen lisäämiseksi sekä pentueen valitsemiseksi ja niiden lisäksi painike, joka vie päiväkirjanäköymään. Päiväkirjanäköymässä voi lisätä pentueelle päiväkirjamerkintöjä sekä muokata vanhoja merkintöjä. Tallennetut merkinnät listataan näkyville, josta merkinnän voi valita muokattavaksi. Päiväkirjanäköymään haetaan kannasta päivitetty tieto saavuttaessa sekä jokaisen tallennuksen jälkeen. Vasemmalla näytetään lista pentueen pennuista ja tarjotaan mahdollisuus painotietojen lisäämiseen ja painotietojen piilottamiseen/näyttämiseen painograafissa. Painotietojen lisääminen tapahtuu lomakkeella omassa näköymässä. Syötetyt painot listataan näkyville, josta virheellisesti tallennetun painon voi poistaa tarvittaessa. Painonäköymään haetaan kannasta päivitetty tieto jokaisen tallennuksen jälkeen. Keskellä näytetään pentujen painotiedot graafin muodossa. Graafin luomiseksi lasketaan vaaka-akselille pentujen ikä päivinä ja oletuksena näytetään kaikkien pentujen painotiedot. Pohjalla näytetään tietoja pentueesta ja tarjotaan mahdollisuus joko muokata pentueen olemassaolevia tietoja tai lisätä pentueeseen pentuja. Pentueen muokkausnäköymässä listataan tallennetut pennut ja tarjotaan mahdollisuus siirtyä pennun muokkaus -näköymään.

Kaikki nämä näköymät painograafia lukuunottamatta muodostetaan omissa näköymäluokissaan, jotka sijaitsevat paketissa views. Toiminnallisissa näköymissä käytettävät lomakkeet muodostetaan omien kenttäluokkien avulla, jotka löytyvät paketista formfields ja jotka validoivat itsensä.

Tiedon lisääminen tai muokkaaminen

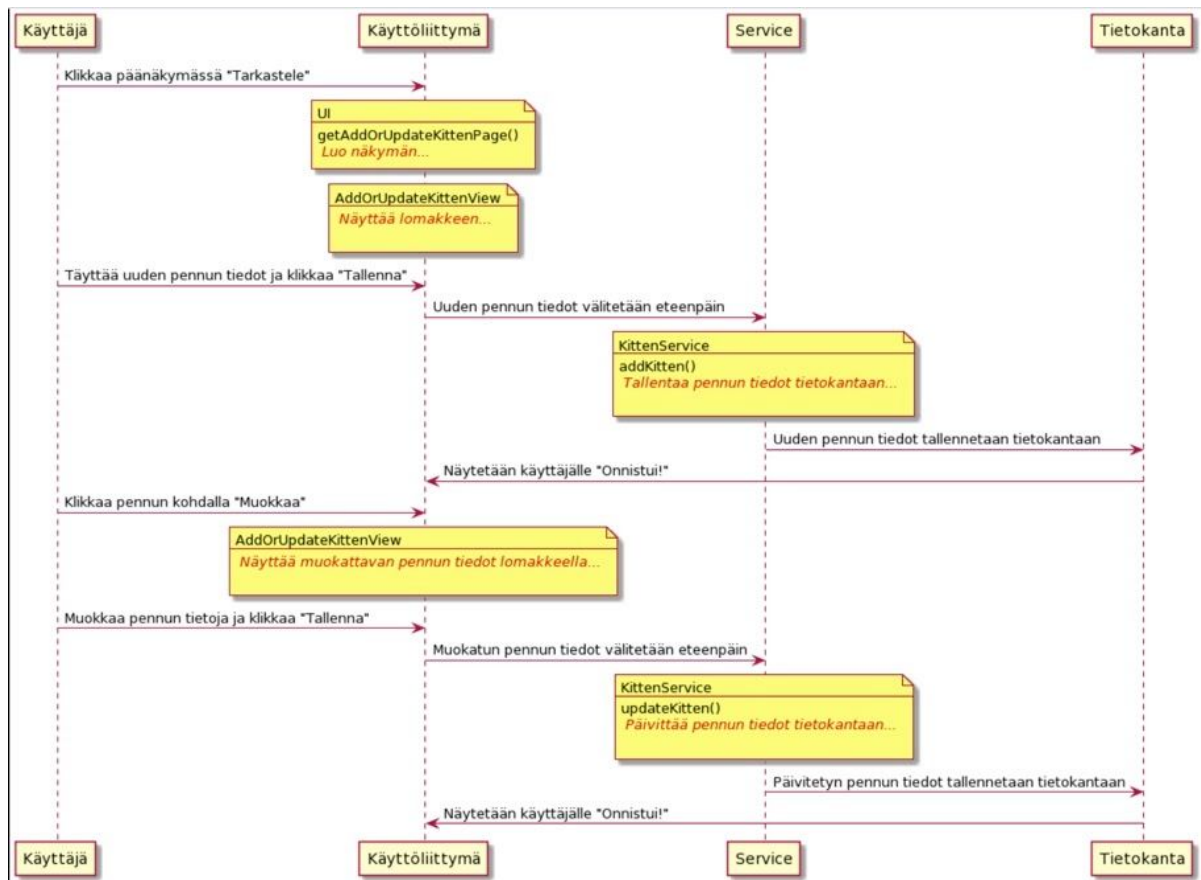
- UI luo views-paketista löytyvän näkymäluokan tarvittavalle toiminnolle
- Tämä luokka luo näkymän jossa on kentät tietojen vastaanottamiseen
- Mikäli kyseessä on tietojen muokkaaminen on konstruktorin parametrina olio. Tällöin lomake esitätetään oliossa olevilla tiedoilla.
- Näkymäluokka ottaa vastaan käyttäjän syötteet
- UI:sta kutsutaan näkymäluokan validointi-metodia, joka taas kutsuu lomakkeen kenttäluokkien validointi-metodeita. UI:lle palautuu tieto syötteiden oikeellisuudesta
 - Muussa tapauksessa käyttäjälle tulostuu ohjeviestit virheellisten syötteiden alle
- UI kutsuu KittenService-luokkaa näkymäluokan metodeilta saatavilla oikeellisilla tiedoilla
- KittenService tallentaa tiedot kantaan ja palauttaa UI:lle tiedon tallennuksen onnistumisesta
- UI kutsuu näkymäluokkaa, joka tulostaa käyttäjälle tiedon tallennuksen onnistumisesta

Luokkakaavio

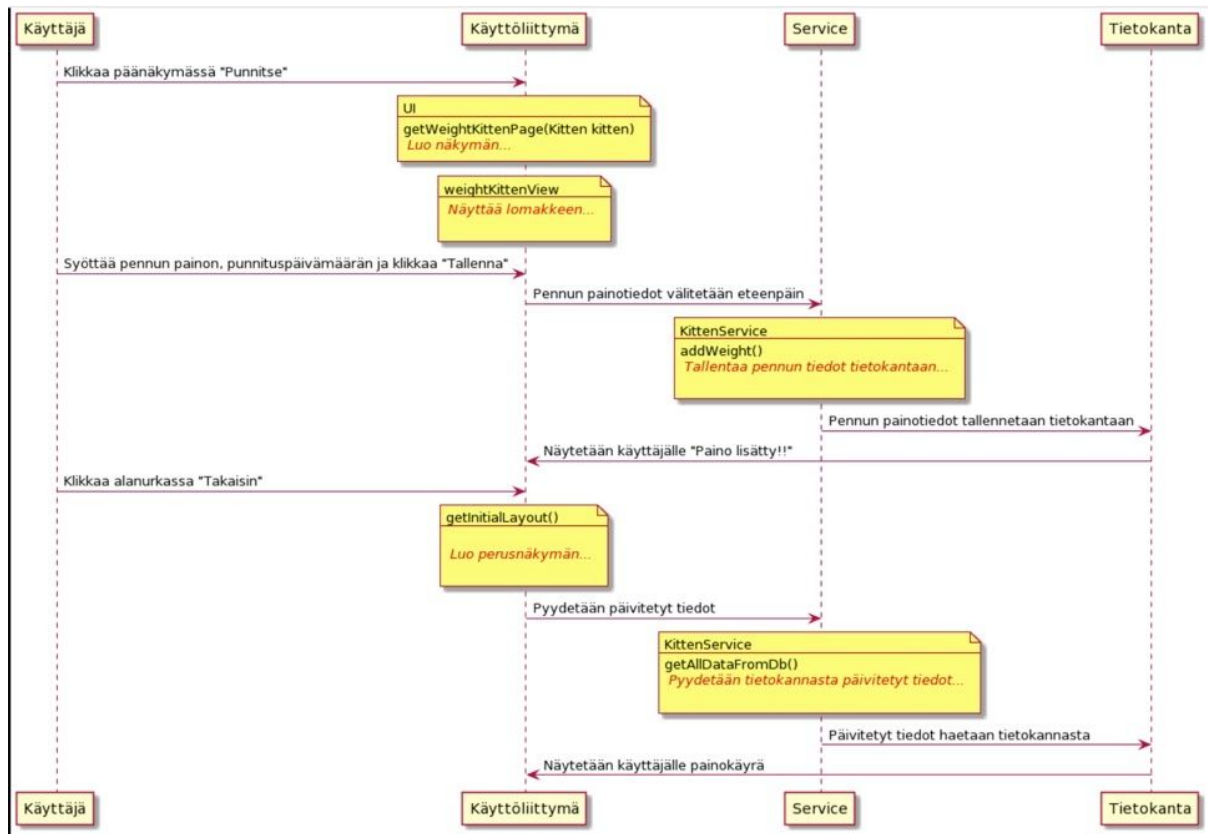


Suurempana ks liite 1

Sekvenssikaavio: pennun lisääminen ja tietojen muokkaaminen



Sekvenssikaavio: pennun punnitseminen



Testaus

Testausselostus

Ohjeet erilaisten generoitujen raporttien luomiselle löytyy README.md -tiedostosta.

Yksikkötestaus

Yksikkötestauksessa on pyritty testaamaan KittenService-sovelluslogiikkaluokan sekä domain-paketin luokkien metodit JUnitin automatisoituna yksikkötestauksena. Tässä on pyritty 70% testikattavuuteen. Käyttöliittymäluokat on jätetty automatisoidun testauksen ulkopuolelle ja rajattu pois testausraportista, mutta järjestelmätestaus osoittaa niiden toimivuuden.

Integroititestaus

Integroititestauksessa on pyritty testaamaan eri luokkien välistä vuorovaikutusta ja toiminnallisuutta. Tämä toteutuu testaamalla jo yksikkötestauksessa käytettyjä sovelluslogiikkaluokkia KittenServicen metodeita, jotka käyttävät sekä kantayhteyttä että domain-paketin luokkien metodeita.

Järjestelmätestaus

Järjestelmätestauksessa on pyritty testaamaan ohjelmistoa sille määriteltyjen vaatimusten suhteen. Järjestelmätestausta on toteutettu ohjelmoinnin edetessä aina uuden toiminnon, muun lisäyksen tai muutoksen yhteydessä luomalla testipentueita ja syöttämällä niille tarvittavia tietoja. Testauksissa on pyritty ottamaan huomioon virheelliset syötteet ja tarkasteltu niiden vaikutusta sekä koodin toiminnassa että tietokannassa.

Testitapauksia:

1. Luodaan uusi pentue vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla pentue tallentuu kantaan, syötekentät tyhjentyvät ja käyttäjälle tulostuu viesti "Pentue lisätty".
 - Vajavaisilla tiedoilla käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvien syötekenttien alle tulostuu kehoitus täyttää kenttä.
2. Vaihdetaan pentuetta päänäkyymässä
 - Valitun pentueet tiedot päivittyvät odotetusti päänäkyymään.

3. Tallennetaan päiväkirjamerkintä vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla merkintä tallentuu kantaan, näkyy oikealla merkintälistassa odotetusti ja syötekentät tyhjäntyvät.
 - Ilman päivämäärän valintaa käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvan syötteen kentän alle tulostuu kehoitus täyttää kenttä.
 - Puuttuvan tekstisyötteen teksti "Kirjoita tähän" tallentuu päiväkirjamerkinnäksi ja syötekentät tyhjäntyvät.
4. Muokataan päiväkirjamerkintää vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla merkintä tallentuu kantaan, merkintä näkyy oikealla merkintälistassa odotetusti ja syötekentät tyhjäntyvät.
 - Tyhjennetty tekstikenttä tallentuu tyhjäksi tekstiksi päiväkirjamerkintöihin ja syötekentät tyhjäntyvät.
5. Tallennetaan painotieto vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla painotieto tallentuu kantaan, näkyy painotietolistassa lomakkeen alapuolella ja käyttäjälle tulostuu viesti "Paino lisätty!".
 - Vajavaisilla tiedoilla käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvien syötekenttien alle tulostuu kehoitus täyttää kenttä.
6. Tallennetaan painotieto graafin ulkopuolisilla arvoilla
 - Painotieto tallentuu kantaan, näkyy painotietolistassa lomakkeen alapuolella ja käyttäjälle tulostuu viesti "Paino lisätty!".
 - Syntymää aikaisemmaksi merkitty punnitustieto ei näy graafissa.
 - Luovutusajan jälkeen asetettu tai yli 2500g:n ja alle 0g:n painotieto näkyy graafissa ilman päätepistettä.
7. Poistetaan painotieto
 - Painotieto poistuu kannasta ja painotietolistasta odotetusti.
8. Piilotetaan pennut graafinäkömystä yksittäin ja yhdessä
 - Pentujen painokäyrä näkyy/poistuu näkyvistä odotetusti.
9. Muokataan pentueen tietoja vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla tieto päivittyy kantaan odotetusti ja käyttäjälle tulostuu viesti "Onnistui!".
 - Vajavaisilla tiedoilla käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvien syötekenttien alle tulostuu kehoitus täyttää kenttä.
10. Muokataan pennun tietoja vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla tieto päivittyy kantaan odotetusti ja käyttäjälle tulostuu viesti "Onnistui!".
 - Vajavaisilla tiedoilla käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvien syötekenttien alle tulostuu kehoitus täyttää kenttä.
11. Lisätään uusi pentu vaadituilla/vajavaisilla tiedoilla
 - Vaadituilla tiedoilla tieto päivittyy kantaan odotetusti ja käyttäjälle tulostuu viesti "Onnistui!".
 - Vajavaisilla tiedoilla käyttäjälle tulostuu viesti "Ei onnistunut!" ja puuttuvien syötekenttien alle tulostuu kehoitus täyttää kenttä.

12. Poistetaan pentu

- Alert-ikkuna ilmestyy odotetusti ja varmistaa poistotapahtuman, tämän jälkeen poistaminen tapahtuu odotetusti.

13. Poistetaan pentue

- Alert-ikkuna ilmestyy odotetusti ja varmistaa poistotapahtuman, tämän jälkeen poistaminen tapahtuu odotetusti kaikista tietokannan tauluista.

Itsereflektio

Omaa pohdintaa

Kissanpentujen kasvu -ohjelmiston syntyminen

Alkuperäisestä vaatimusmäärittelystä poimittua:

Yleiskuvaus

Ohjelma on työkalu kissankasvattajalle. Ohjelma seuraa tiineysajan kestoa ja synnytyksen jälkeen pentujen painonkehitystä luovutusikään (14vk) saakka. Ohjelma tuottaa pennuille kasvukäyrät, joita voi tarkastella yksittäin, pentueittain tai useampaa pentuetta keskenään vertaamalla. Lisäksi pentueella on päiväkirja, jonne voi lisätä muistiinpanoja. Päiväkirja laskee pentujen iän automaattisesti jokaiseen merkintään.

Toiminnalliset vaatimukset

Ohjelman täytyy toteuttaa seuraavat vaatimukset:

- *Pentueen luominen astutuspäivästä*
- *Pentueen nimeäminen*
- *Tiineysajan laskeminen astutuspäivän perusteella*
- *Profiilin luominen jokaiselle pennulle*
- *Kasvukäyrän tuottaminen jokaiselle pennulle*
- *Kasvukäyrien vertailu pentueittain*
- *Päiväkirjaan iän laskeminen automaattisesti*
- *Pentujen nimeäminen*
- *Virheellisten tallennusten korjaaminen*

Vaatusmääritylun toteutuminen

Edellä määriteltyt osa-alueet toteutuvat lukuun ottamatta pentueiden kasvukäyrien keskenään vertailemista, joka jäi aika aikaisessa vaiheessa pois työlistalta. Tiineysajan laskeminen oli huonosti muotoiltu vaatimusmäärittelyyn. Tarkoitin tällä arvioitua synnytyspäivää ja lisäsin tähän myös arvion luovutuspäivästä. Luovutuspäivä tarkentuu, kun synnytyksen todellinen päivä selviää. Ohjelmisto osoittautui monimutkaisemmaksi kuin osasin odottaa ja jouduin muuttamaan työsuunnitelmaani parikin kertaa. Toteutus myöhästyi hieman etukäteen arvioidusta monimutkaisuuden yllättäessä ja vielä viimeisen iteraation jälkeen muutin näkymien rakennetta. Ilmoittauduin kurssille myöhässä ja tämä vaikutti kurssin alkuosan suoritukseen siinä määrin, että palautin ensimmäisen iteraation kolme päivää myöhässä. Pohjatietoni olivat hieman vajavaiset ja kurssi oli sen vuoksi todella haastava. Opin kuitenkin tällä kurssilla paljon uutta.

Ohjelmiston luomiseksi tein taustatyötä ja haastattelin kissankasvattajia. Olisin kuitenkin voinut paneutua tähän työhön vielä huolellisemmin. Jos lähtisin toteuttamaan tätä tehtävää uudelleen, tekisin ensin haastattelukierroksen muodostaakseni tarvekartoituksen, tämän jälkeen suunnittelisin ohjelmiston rakenteen ja miettin toiminnallisuuden.

Ohjelmistoa suunnitellessani en osannut arvata etukäteen, millaisiin haasteisiin tulen törmäämään. Projektin loppuvaiheeseen jäi paljon sellaista työtä, jota en osannut ennakoida. Olin innokas etenemään ja tein asioita osaamatta tehdä tarpeeksi yksityiskohtaisia suunnitelmia. Tein stylecheckin ja suurimman osan yksikkötestauksesta vasta siinä vaiheessa, kun ohjelmisto oli jo toiminnallisesti valmis. Tästä johtuen jouduin tekemään ylimääräistä työtä metodien lyhentämiseksi. Käyttäessäni suurimman osan ajastani ohjelmiston toiminnallisuuden suunnitteluun ja aikataulussa pysymiseen, jäi muille projektin osille liian vähän aikaa.

Olen tyytyväinen lopputulokseen, vaikka ohjelmistossa olisikin vielä kehittämistä. Jos aikaa olisi enemmän, muokkaisin ulkoasua miellyttävämmäksi käyttäjälle. Jotta ohjelmisto toimisi täydellisenä työvälineenä kasvattajille, pitäisi siihen lisätä muutama toiminto. Muun muassa pentukuolemien käsitteleminen tulisi lisätä toimintoihin. Kasvattajat pitävät kirjaa kuolleiden pentujen määristä ja kuolemien syistä ja ohjelmistoon saisi vaivattomasti lisättyä kyseisen toiminnon. Kurssin aikatauluun nähden työtä oli jo nyt tarpeeksi, joten päätin jättää tämän lisäämisen mahdolliseen seuraavaan ohjelmistoversioon. Olisi myös hyödyllistä, että ohjelmisto tuottaisi raportteja pentueista. Raporttiin tulostuisi tilastoja tiineysajan kestoista ja muista hyödyllisistä asioista. Kissaliitto määrittää sen, kuinka usein pentueita saa tehdä ja ohjelmisto voisikin laskea näitä synnytysten välejä, jolloin ohjelmistoa voisi käyttää työvälineenä myös pentueiden suunnittelussa. Olisi käytännöllistä, jos tämä ohjelmisto tuottaisi valmiit materiaalit kaikkeen kasvatukseen liittyvään byrokratiaan. Jos tekisin ohjelmiston web-pohjaiseksi, voisin olla yhteydessä Kissaliittoon ja keskustella yhteistyöstä heidän kanssaan, opittuani tarpeeksi tietoturvasta ja erilaisista tietokannoista. Ohjelmisto voisi toimia myös mobiilisovelluksena ja sitä voisi muokata vastaamaan myös koirankasvattajien tarpeita.

Ohjelman sisään on liitetty Java, jotta vaikea asennusprosessi voidaan ohittaa. Tästä johtuen tiedosto on suuri. Tämä on mielestäni kuitenkin parempi vaihtoehto, kuin vaikeasti asennettava java-ympäristö. Minulla ei ole käytössäni Windows-konetta, joten suorittavan asennuspaketin luominen niihin ei tässä vaiheessa onnistunut.

Työn eteneminen iteraatioittain

***Ensimmäisessä iteraatiossa** luodaan toiminto pentueen lisäämiseksi ja pentueiden listaus. Vähintään yksi testi.*

Varsinainen ohjelmointityö käynnistyi kiireellä, enkä ehtinyt tehdä testejä tässä vaiheessa. Ilmoittauduin kurssille myöhässä ja olin katsonut ensimmäisen iteraation palautuspäivän väärin, joten palautin tämän kolme päivää myöhässä ja nopeasti laadittuna. Pentueen lisääminen ja niiden listaaminen oli yksinkertainen toteuttaa. Maven-projekti oli vaikea luoda ja suoritettavan jar-tiedoston generoinnin mahdollistava toiminto todella hankala saada tehdyksi. Ylipäättään se, että ohjelmiston sai pyörimään niin, että java fx tulee mavenista, oli ison työn takana. Se, että kirjastot voivat olla joko editorissa tai mavenissa, tuli uutena tietona minulle. Huomasin mavenin käytön hyödyn käytännössä, kun kävin myöhemmin katsomassa useamman kurssikaverin ohjelmistoja ja yhdessä näistä java fx ei ollut mavenissa siinä vaiheessa.

Tässä vaiheessa olisi ollut hyvä jo olla valmis suunnitelma rakenteesta. Minulla oli suunnitelma luokista ja tietokannan rakenteesta, mutta suunnitelmani oli niin kevyt, että siitä ei juurikaan ollut käytännössä hyötyä. Tässä vaiheessa kuvittelin ohjelmiston vielä yksinkertaiseksi ja mietin, täyttääkö tämä edes kunnolla kurssin vaatimustasoa. Nyt tiedän enemmän siitä, millaisia asioita on otettava huomioon jo tässä vaiheessa ja miten paljon tarkemmin rakenne on suunniteltava. Kokonaisuudessaan lisäisin suunnittelua alkuvaiheeseen. Minulla ei ollut tarpeeksi yksityiskohtaisesti mietittynä, mitä näkymiä ohjelmistossa tulisi olla ja miten ohjelmiston käyttäminen olisi sujuvaa. Tiedostan sen, että suunnitelmaa voi olla joskus tarpeen muokata työn edetessä, mutta nyt jouduin muuttamaan sitä moneen kertaan vain siksi, etten osannut miettiä tätä riittävän tarkasti etukäteen. Jokainen muutos oli kuitenkin opettavainen kokemus, enkä tiedä tätä paremmin itselleni sopivaa tapaa oppia.

Toisessa iteraatiossa luodaan toiminto pentuprofiilien luomiseen ja painotietojen lisäämiseen. Lisää testejä.

Tässä vaiheessa aloin todella ymmärtää kuinka paljon merkitystä on rakenteen suunnittelemisella. Eniten aikaa kului sen pohtimiseen. Jos tekisin tämän uudelleen, piirtäisin näkymät jo suunnitteluvaiheessa ja esittelisin ne pääpiirteittäin testiryhmälle.

Aloitin testien tekemisen, mutta en saanut niitä kaikkia vielä toimimaan, joten iteraation palautukseen kommentoin toimimattomat pois ja korjasin ne myöhemmin. Tässä vaiheessa koodia syntyi ja aloitin ulkoasun rakentamisen ja tietokanta muodostui suunnitelman mukaisesti. Toiminnallisesti nyt oli mahdollista lisätä pentue sekä pentuja pentueille syntymäpainoineen. Päivämäärät oli syötettävä tietyssä muodossa ja arvelin tästä aiheutuvan jatkossa ongelmia. Lisäsin tämän korjauksen työlistalleni. Kaavioiden tekeminen tuntui hieman epävarmalta, koska ymmärsin suunnitelmani olevan vajavainen ja näihin jouduinkin palaamaan myöhemmin rakennemuutosten vuoksi.

Sain palautetta, että luokkia olisi hyvä olla enemmän ja huomasin tämän itsekin. Lisäsin tämänkin korjauksen työlistaani.

Kolmannessa iteraatiossa kasvukäyrät ja niiden vertailu.

Kasvukäyrien toteuttaminen osoittautui haastavaksi. Suurin vaikeus oli saada Javan LocalDate ja SQL:n Date kommunikoimaan keskenään. Käytin paljon aikaa tämän asian selvittämiseen, enkä ole vielä ikinä ihan varma, onko ratkaisuni paras mahdollinen. Vielä palautushetkellä tämän kanssa oli haasteita ja sain ratkaistua ne vasta palautuksen jälkeen. Koska minua vertaisarvioiva kurssikaverini oli vahingossa palauttanut väärän tiedoston, otin häneen yhteyttä ja samalla kun sain hänen työnsä arvioitavaksi, toimitin hänelle korjatun tiedoston omastani, jotta hänen tarvitsisi käyttää aikaansa käännösongelmaan. En ole varma, korjaantuiko ongelma lopulta vain sillä, että tyhjensin kannan, jossa minulla oli päivämääriä kahdessa eri muodossa kokeiltuani vähän kaikkea vai sillä, että tajusin Daten olevan tekstiä ja aloin kohdella sitä selkeämmin Stringinä. Kasvukäyrä näytti tässä vaiheessa kaikki pentueen käyrät, eikä pentujen painoja ollut mahdollista tarkastella visuaalisesti yksittäin.

Tässä vaiheessa näytin ohjelmiston perusnäköymän kissankasvattajien testiryhmälle ja ymmärsin, että rakennetta täytyy edelleen selkeyttää. Olin pitänyt koko ajan mielessä, että ohjelmiston käyttämisen täytyy olla helppoa, mutta sen osalta oli vielä parantamisen varaa. Tämä vaihe oli lähinnä erilaisten Pane-elementtien yhteensovittamista. Jatkossa tiedän, että nämäkin on mahdollista suunnitella etukäteen.

Neljännessä iteraatiossa päiväkirja ja tallennusten muokkaaminen.

Pentueen, pennun ja punnitusten tietojen muokkaaminen Päiväkirja toteutui, mutta tallennusten listaaminen ja muokkaaminen jäi hieman kesken. Päiväkirjanäkymän luominen oli monimutkaista ja huomasin, että sen osalta koodista alkoi tulla vaikeasti luettavaa. Keskityin vertaisarviossa saamiini huomioihin. Loin formFields -kansion ja tein sinne luokat lomakkeiden kenttätyyppien hallintaan. Lisäsin selitteet virheviesteihin ja toteutin validoinnit lomakkeisiin. Tajusin, että pentua luodessa kysytty syntymän kellonaika on tarpeeton ja hyödyllisempää on tallentaa pennun virallinen nimi. Tämä ei vaatinut suuria muutoksia, koska syntymäkellonaika oli String jo valmiiksi ja tämä vaikutti ainoastaan sarakkeiden leveyksien määrittelyyn pentueen muokkaus -näkyvässä. Tämän iteraation palautuksen jälkeen tehtäväksi jäi vielä korjata syntymäkellonaikojen uudet nimeämiset koodiin ja kantaan.

Iteraatioiden jälkeen.

Sain päiväkirjamerkintöjen muokkaamisen toimimaan ja lisäsin muokkausmahdollisuuden myös pentueen nimelle. Korjasin uudet nimet koodiin ja kantaan pennun syntymäkellonajan viralliseksi nimeksi muuttamisen vuoksi. Näiden korjausten jälkeen totesin, että toiminnallisuuden osalta ohjelmisto on nyt valmis. Jäljellä oli bugi päivämäärien kohdalla. Tyhjän päivämääräkentän virheilmoitus tuli toivotusti, mutta ei toiminut väärin syötettyjen päivämäärien kohdalla. Ohjelmisto ei tallentanut kantaan väärin syötettyjä päivämääriä, mutta ei myöskään ilmoittanut käyttäjälle, miksi tallentaminen ei onnistunut.. Tämän ratkaiseminen osoittautui hankalaksi ja päädyin estämään päivämäärien syöttämisen käsin kokonaan. Ohjelmistossa ei ole mahdollista poistaa pentuetta eikä päiväkirjamerkintää. Ainoastaan muokkaaminen on mahdollista. Tämä on ratkaistavissa yksinkertaisesti.

Tässä vaiheessa tein stylecheckin ja testikattavuusraportit. Metodini olivat todella pitkiä ja päädyin luomaan omat luokat kaikille näkymille. Jaoin metodeita palasiksi ja pääsin lähemmäksi sallittuja pituuksia. Joitakin metodeita jätin pitkiksi, koska mielestäni niiden lyhentäminen ei toisi lisäarvoa, vaan hankaloittaisi koodin lukemista.

Nyt koodissa ilmeni bugi, jossa komponentti oli liitetty kahteen eri sceneen ja tämä synnytti virheen. Muutin koodin rakennetta niin, että alussa otetaan vain yksi scene käyttöön, eivätkä metodit palauta enää scenejä, vaan jonkin käyttöliittymäkomponentin.

Kirjoitin käyttöohjeen ja arvioitin sen yhdellä testiryhmän jäsenellä. Sain palautetta, että ohjeiden perusteella ohjelmiston käyttäminen todennäköisesti onnistuu. Edelleen kuitenkin epäilen, ettei ohjelmiston asentaminen tule onnistumaan kaikilta käyttäjiltä ja olen varautunut siihen, että joudun auttamaan siinä. Tarkoituksena kuitenkin on saada ohjelmisto käyttöön tälle testiryhmälle, jotta saan palautetta käyttökokemuksesta ja voin myöhemmin kehittää ohjelmistoa.

Työpäiväkirja

Työpäiväkirjan ajan tasalla pitäminen osoittautui mahdottomaksi. Vietin koneen ääressä kaiken aikani ja voin vain karkeasti arvioida käyttämäni tehokasta työaikaa.

Lokakuu:

- vaatimusmäärittely 3t
- suunnittelu 3t

Marraskuu:

- mavenprojektin luominen 12t
- 1. iteraatio 5t
- suunnittelu 18t
- 2. iteraatio 30t
- 3. iteraatio 30t
- testit 12t

Joulukuu:

- 4. iteraatio 30t
- korjaukset 40t
- testit 8t
- loppudokumentti 15t

Liite 1: Luokka....

