

Suunnitteludokumentti

Karstula

Helsinki 3.5.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 + 1 op)

Projektiryhmä

Juha-Pekka Juutilainen

Hannu Kukko

Antto Mäkinen

Antti Rajasärkkä

Ari Raunio

Mika Tantarimäki

Asiakas

Eija Mäntyharju

Johtoryhmä

Sanna Keskiöja

Kimmo Simola

Kotisivu

<http://www.cs.helsinki.fi/group/karstula/>

Versiohistoria

| Versio | Päiväys | Tehdyt muutokset |
|--------|-----------|---|
| 0.1 | 27.2.2007 | Ensimmäinen versio |
| 0.2 | 6.3.2007 | Tarkastustilaisuuden jälkeinen korjattu versio |
| 0.3 | 9.3.2007 | Kolmas versio |
| 0.4 | 14.3.2007 | Toteutuksen aloittamista ennen täydennetty versio |
| 1.0 | 15.3.2007 | Jäädetytty versio |

Sisältö

| | | |
|----------|---------------------------------------|-----------|
| 1 | Johdanto | 1 |
| 2 | Toteutusympäristö | 1 |
| 3 | Arkkitehtuuri | 1 |
| 4 | Poikkeukset | 2 |
| 4.1 | Poikkeushierarkia | 3 |
| 5 | Käyttöliittymä | 3 |
| 5.1 | Toimintojen kuvaus | 3 |
| 5.1.1 | Päänäkymä | 3 |
| 5.1.2 | Teorianäkymä | 4 |
| 5.1.3 | Harjoittelunäkymä | 5 |
| 5.1.4 | Testinäkymä | 8 |
| 5.1.5 | Ylläpitonäkymä | 12 |
| 5.2 | Tekninen kuvaus | 16 |
| 5.2.1 | Yleisrakenne | 16 |
| 5.2.2 | Tarkkailija-malli | 16 |
| 5.2.3 | ParentView | 18 |
| 5.2.4 | MainView | 18 |
| 5.2.5 | TheoryView | 18 |
| 5.2.6 | PractiseView | 18 |
| 5.2.7 | TestView | 20 |
| 5.2.8 | AdminView | 20 |
| 6 | Logiikkajärjestelmä | 20 |
| 6.1 | Luokkakaavio | 21 |
| 6.2 | Luokkien rajapintakuvaukset | 22 |
| 6.2.1 | Practice | 22 |
| 6.2.2 | Statistics | 24 |
| 6.2.3 | Maintenance | 25 |
| 6.2.4 | Test | 28 |

| | | |
|----------|---|-----------|
| 6.2.5 | Print | 29 |
| 6.2.6 | Guard | 30 |
| 6.2.7 | Theory | 31 |
| 7 | Tietokanta | 32 |
| 7.1 | Tietokantajärjestelmän luokkakaavio | 33 |
| 7.2 | Tietokannan käyttämät tiedostot | 33 |
| 7.3 | Salauskomponentti | 34 |
| 7.4 | XML-komponentti | 35 |
| 7.5 | Sekvenssikaavio tietokantajärjestelmän toiminnasta. | 36 |
| 7.6 | Luokkien rajapintakuvaukset | 37 |
| 7.6.1 | DBHandle | 37 |
| 7.6.2 | XMLParser | 39 |
| 7.6.3 | Crypt | 42 |
| 7.6.4 | QuestionDB | 43 |
| 7.6.5 | SettingsDB | 45 |
| 7.6.6 | SecretSettingsDB | 45 |
| 7.6.7 | TheoryDB | 46 |
| 7.6.8 | Question | 46 |
| 7.6.9 | Answer | 47 |

1 Johdanto

Karstula-projekti on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojektiryhmän toteuttama lääkelaskuopetusohjelmisto.

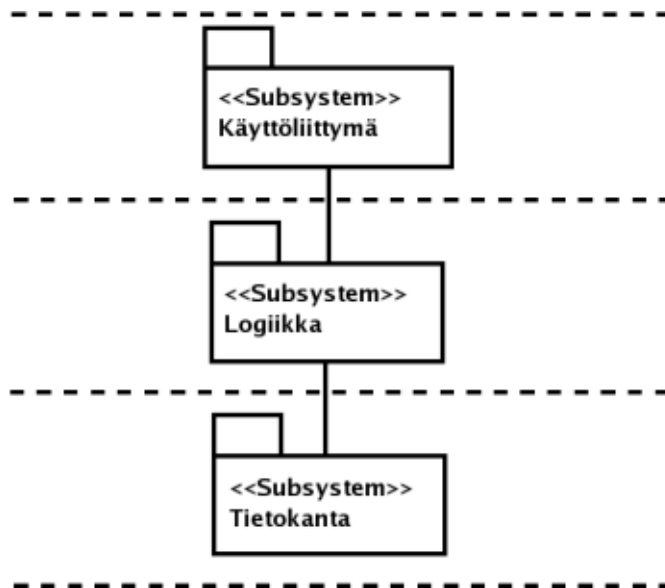
Tämä dokumentti kuvaa ohjelmiston yleisarkkitehtuurin, komponenttijaon ja luokkien julkiset rajapinnat.

2 Toteutusympäristö

Asiakkaan laitekanta koostuu Windows XP -koneista. Lääkelaskuohjelma kuitenkin toteutetaan Javalla, joten ohjelmiston pitäisi toimia asiakkaiden koneiden nykyisten käyttöjärjestelmien lisäksi jokaisella käyttöjärjestelmällä, johon saa asennettua Sunin Java 6.0 -ympäristön.

3 Arkkitehtuuri

Ohjelmisto jakautuu kolmeen kerroksittaiseen osajärjestelmään: käyttöliittymään, ohjelmalogiikkaan ja tietokantaan. Osajärjestelmät ovat toistaan riippumattomia, ja ne käyttävät toistensa palveluita seuraavasti:



Kuva 1: Ohjelmiston arkkitehtuuri

4 Poikkeukset

Ohjelman poikkeukset abstraktoidaan jokaiselle kerrokselle sopivaksi. Ylemmille kerroksille heitetään vain ketjutettuja poikkeuksia, jotka sisältävät tarkempaa tietoa tapahtuneesta virheestä.

Jos käyttäjälle näytetään poikkeustilanteessa virhe, niin virhedialogi sisältää aluksi poikkeuksen tyyppin perusteella päätellyn virheilmoituksen, sitten tämän poikkeuksen sisältämän viestin, ja sen jälkeen alempien poikkeusten viestit.

Esimerkki poikkeusten toiminnasta:

Käyttäjä siirtyy ylläpito-osioon, jolloin AdminView-luokka yrittää hakea tietokannasta kaikki koekysymykset. Aliohjelmien kutsuketju päättyy tällöin lopulta luokalle XMLParser, joka yrittää parsia XML-tiedoston, ja tältä luokalta vielä itse parsimisen hoitavalle luokalle (ei itse tehdyille).

Jos koekysymyksiä sisältävässä XML-tiedostossa on kuitenkin virhe, niin XMLParserin parsimiseen käyttämä luokka heittää parsimisvirhettä tarkoittavan SAXException-poikkeuksen. Parsimiseen käytetty luokka liittää myös poikkeukseen virheilmoituksen. Esimerkiksi "Invalid character at line 10".

XMLParser sieppaa SAXException-poikkeuksen ja muuntaa sen XMLParserExceptioniksi, johon XMLParser lisää alkuperäisen SAXException-poikkeuksen sisältämän virheilmoituksen.

DBHandle saa tämän poikkeuksen XMLParserilta, sieppaa sen ja muuntaa (ketjuttaen) sen omaksi poikkeukseksi "DBException". Koska poikkeus tapahtui XMLParseria käytettäessä, DBHandle liittää poikkeukseen ilmoituksen "Virhe parsiessa XML-tiedostoa TestQuestions.xml".

Maintenance-luokka sieppaa DBHandlein heittämän poikkeuksen, ja jälleen muuntaa poikkeuksen tyyppiä DBLogicException. Muuntaessaan se lisää poikkeukseen ilmoituksen "Tietokantavirhe".

Nyt AdminView saa poikkeuksen DBLogicException, ja näyttää käyttäjälle tilanteeseen ja poikkeuksen tyyppiin sopivan ilmoituksen, jonka lopussa on suluisa ketjutettujen poikkeusten ilmoitukset järjestyksessä. Ilmoitus voi olla vaikka tällainen:

```
*****
*
*   Virhe koekysymyksiä haettaessa
*
*   (Tietokantavirhe:
*   Virhe parsiessa XML-tiedostoa TestQuestions.xml:
*   Invalid character at line 10)
*
*****
```

4.1 Poikkeushierarkia

- LogicException Logiikkakerroksen heittämiä poikkeusten ylikuokka.
 - DBLogicException
 - InvalidQuestionIDException
 - NotEnoughQuestionsException
 - SSNInvalidException
- DBException Tietokannan heittämiä abstraktoitu poikkeus.
 - XMLParserException XML-parserin heittämiä poikkeus.
 - CryptException Salausmoduulin heittämiä poikkeus.
- FormatException Answer-luokan heittämiä poikkeus.

5 Käyttöliittymä

Tässä luvussa kuvataan käyttöliittymäkomponentin toiminta siten, että ensin käsitellään käyttöliittymän toiminta käyttäjän näkökulmasta ja sen jälkeen toteutuksen kannalta. Käyttöliittymä toteutetaan Javan Swing-kirjastolla.

5.1 Toimintojen kuvaus

Tässä luvussa esitellään toiminnot kuvattuna käyttäjän näkökulmasta.

5.1.1 Päänäkymä

Päänäkymä koostuu suuresta tilastonäytöstä, jossa käyttäjä näkee eri osioiden harjoittelun tuloksensa. Jokaisesta osiosta voi siirtyä suoraan kyseisen osion teoriaan tai harjoituksiin. Lisäksi näkyvillä on mahdollisuus käynnistää koe tai siirtyä ylläpitoon antamalla salasanana.

Lääkelaskuohjelma – Päävalikko

Harjoittelu

| Laskutyyppi | Teoria | Harjoitukset | Viimeisin harjoitus | Yhteensä |
|-----------------|-------------------------------------|--|---------------------|----------|
| Murtoluvut | <input type="text" value="Teoria"/> | <input type="text" value="Harjoittele"/> | | |
| Desimaaliluvut | <input type="text" value="Teoria"/> | <input type="text" value="Harjoittele"/> | | |
| Prosenttilaskut | <input type="text" value="Teoria"/> | <input type="text" value="Harjoittele"/> | | |
| Verrantolaskut | <input type="text" value="Teoria"/> | <input type="text"/> | | |

Testi

Salasana Testin voi avata vain koetilaisuuden valvoja

Ylläpito

Salasana Muuta ohjelman asetuksia

Kuva 2: Päävalikko

5.1.2 Teorianäkymä

Teorianäkymässä käyttäjä voi lukea tietokantaan tallennettuja teoriadokumentteja eri osa-alueilta sekä siirtyä suoraan avoimna olevan osa-alueen harjoituksiin. Jos käyttäjä on avan- nut teorianäkymän harjoitusnäköymästä, hänellä on ainoastaan mahdollisuus palata käyn- nissä olleeseen harjoitukseen tai lopettaa harjoitusjakso palaamalla päävalikkoon.

| Murtoluvut | Desimaaliluvut | Prosenttilaskut | Verrannot |
|------------|----------------|-----------------|-----------|
|------------|----------------|-----------------|-----------|

Sisältö

1. Yhteenlasku
2. Kertolasku
3. Jakolasku

1. Yhteenlasku

$$\frac{2 \leftarrow \text{Osoittaja}}{5 \leftarrow \text{Nimittäjä}}$$

Jotta kaksi murtolukua voidaan laskea yhteen, ne on ensin lavennettava samannimisiksi eli molempien nimittäjäksi on saatava sama luku.

Palaa päävalikkoon Siirry harjoituksiin

Kuva 3: Teorianäkymä

5.1.3 Harjoittelunäkymä

Harjoittelu tapahtuu harjoitusjaksoina, joiden pituus määräytyy käyttäjän ilman vihjeitä oikein vastattujen kysymysten määrän mukaan. Oletuksena ohjelma vaatii viisi oikeaa vastausta, mutta tätä määrää ylläpitäjä voi vaihtaa ohjelman asetuksista.

Läkelaskuohjelma-harjoitteluosio

| Murtoluvut 0/0 | Desimaaliluvut 0/0 | Prosenttilaskut 0/0 | Verrannot 0/0 |
|----------------|--------------------|---------------------|---------------|
|----------------|--------------------|---------------------|---------------|

Kysymys 1: Paljonko on $1/4 + 2/5$?

Vastaus: **Vastaa**

Olet vastannut ilman vihjeitä oikein 0 kysymykseen

Vihje 1: Lavenna luvut ensiksi samannimisiksi.

Siirry teoriaan

Anna vihje

Ohita kysymys

Näytä vastaus

Palaa päävalikkoon

Kuva 4: Harjoittelunäkymä

Harjoittelunäkymässä käyttäjälle näytetään ensin kysymys ja vastauskentän jälkeen vastauksen yksikkö. Käyttäjä voi joko vastata, pyytää näyttämään vihjeen tai siirtyä lukemaan kyseessä olevan osa-alueen teoriaa. Jos käyttäjä vastaa oikein, lisätään hänen tililleen yksi oikea vastaus. Jos käyttäjä vastaa väärin tai hän pyytää näyttämään vihjeen, ohjelma näyttää ensimmäisen vihjeen. Tämän jälkeen käyttäjän oikeasta vastauksesta ei enää anneta pistettä. Samoin käyttäjän siirtyminen teorianäkymään tulkitaan vihjeeksi, eikä teoriasta palaamisen jälkeen oikeasta vastauksesta anneta pistettä.

Ensimmäisen vihjeen jälkeen käyttäjä voi edelleen joko vastata, siirtyä teoriaosioon tai pyytää toisen vihjeen. Väärä vastaus tai käyttäjän pyyntö näyttävät toisen vihjeen, jonka jälkeen väärä vastaus tuo käyttäjälle kaksi uutta toimintoa: näytä vastaus ja seuraava kysymys. Näytä vastaus tuo näkyville oikean vastauksen ja sen laskukaavan. Tämän jälkeen käyttäjä voi enää valita siirtymisen teoriaan tai seuraavan kysymyksen.

Läkelaskuohjelma-harjoitteluosio

| | | | |
|----------------|--------------------|---------------------|---------------|
| Murtoluvut 0/0 | Desimaaliluvut 0/0 | Prosenttilaskut 0/0 | Verrannot 0/0 |
|----------------|--------------------|---------------------|---------------|

Kysymys 1: Paljonko on $1/4 + 2/5$?

Vastaus: **Vastaa**

Olet vastannut ilman vihjeitä oikein 0 kysymykseen

Siirry teoriaan

Anna vihje

Ohita kysymys

Näytä vastaus

Vihje 1: Lavenna luvut ensiksi samannimisiksi.

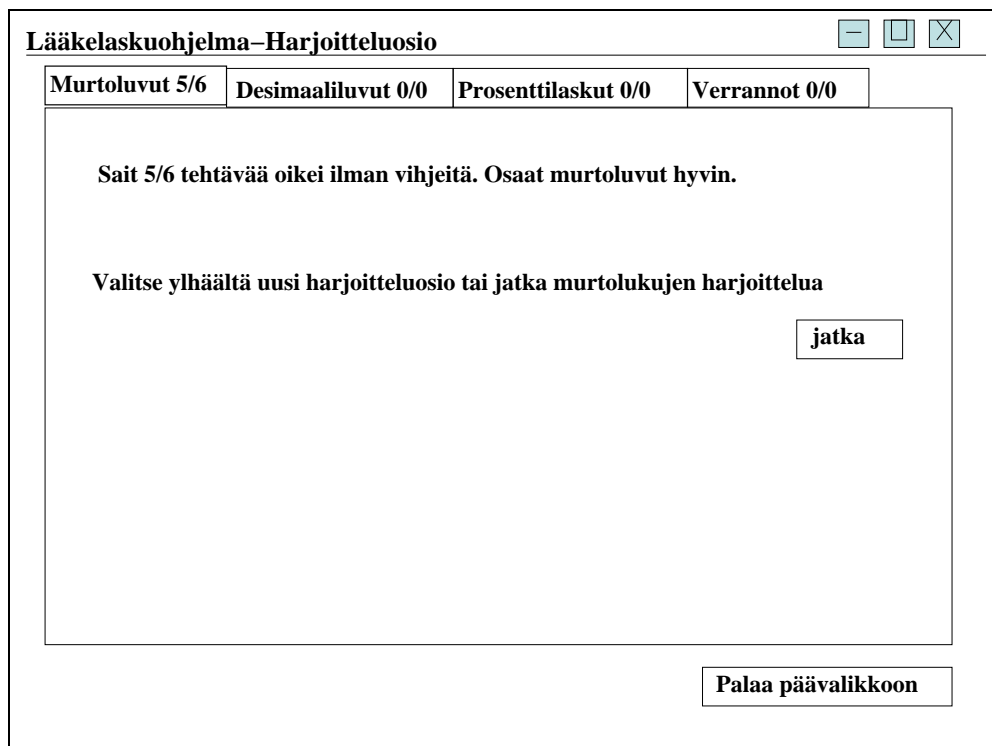
Vihje 2: Lavenna $1/4$ 5:llä ja $2/5$ 4:llä ja laske osoittajat yhteen.

Palaa päävalikkoon

Kuva 5: Harjoittelunäkymä kahden vihjeen jälkeen

Kun käyttäjä lopulta vastaa oikein tai siirtyy seuraavaan kysymykseen, ohjelma päivittää tilastotietojaan kahteen eri laskuriin: laskuri ohjelman käynnistämisestä ja tämän harjoittelujakson laskuri. Molemmat laskurit ovat näkyvissä sekä harjoittelu- että päänäkymässä. Käyttäjälle näytetään seuraava kysymys. Kuitenkin jos käyttäjä sai viidennen pisteensä (eli vastasi viidennen kerran oikein ilman vihjeitä), harjoittelujakso päättyy ja käyttäjälle näytetään raportti, jossa kerrotaan hänen harjoittelujaksonsa tilastot ja sanallinen raportti ("Vastasit ilman vihjeitä oikein 5 / 8 kysymyksestä -> harjoittele lisää!"). Käyttäjällä on mahdollisuus joko jatkaa saman osa-alueen harjoittelua tai valita jokin muista osa-alueista.

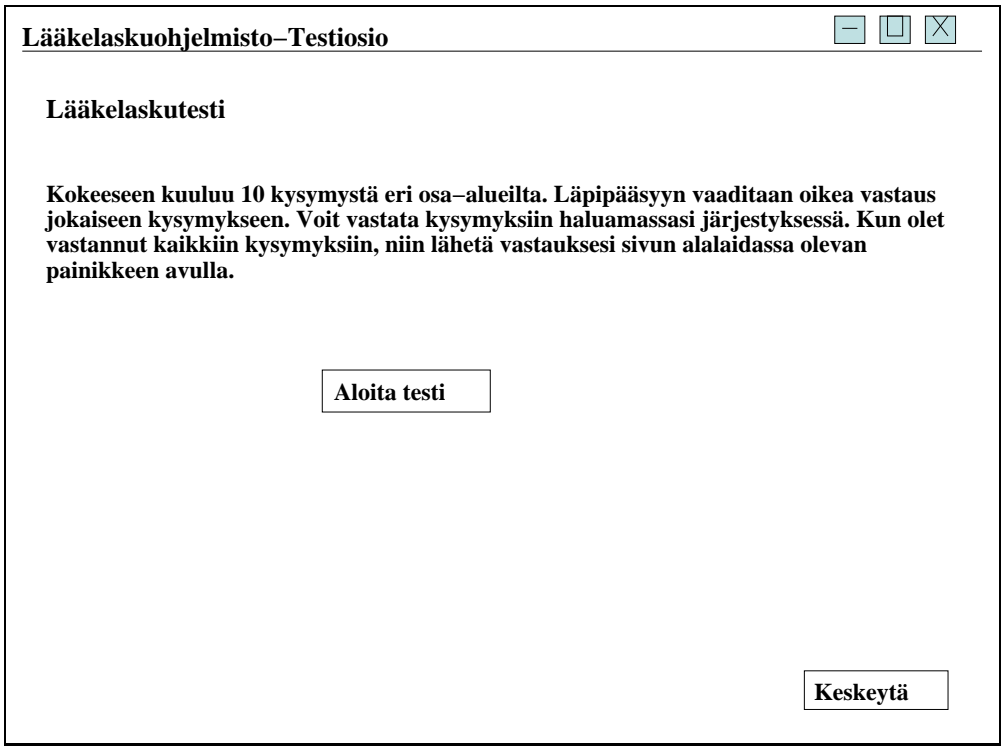
Koko harjoittelujakson ajan käyttäjällä on mahdollisuus lopettaa harjoittelujakso palaamalla päävalikkoon. Tällöin ohjelma nolaa harjoittelujakson laskurin, mutta kesken jääneen jakson vastaukset jäävät kokonaistilastoon.



Kuva 6: Harjoittelunäkymä kun käyttäjä on vastannut oikein viiteen harjoittelukysymykseen

5.1.4 Testinäkömä

Testinäkömän voi avata vain antamalla ensin salasanan. Tällöin ohjelma näyttää kokeen aloitussivun, jossa on ohjeita kokeen suorittajalle ja mahdollisuus joko aloittaa koe tai keskeyttää (ja palata päävalikkoon). Keskeytä-napin painaminen pyytää käyttäjältä vielä varmistuksen, haluaako hän varmasti lopettaa kokeen kesken.



Kuva 7: Testinäkö

Kun kokeen suorittaja aloittaa kokeen, ohjelma näyttää hänelle kerralla kymmenen kysymystä, jotka arvotaan koekysymysten joukosta eri osa-alueilta erikseen ylläpito näytön kautta määritellyssä suhteessa (esim. 2 murtolukulaskua, 2 desimaalilukulaskua, 3 prosenttilaskua ja 3 verrantoa). Käyttäjä voi syöttää vastauksensa kysymyksiin missä järjestyksessä tahansa. Ohjelma ei salli vastausten lähettämistä, ennen kuin kaikkiin kysymyksiin on vastattu. Käyttäjä voi myös keskeyttää kokeen ja palata päävalikkoon.

The screenshot shows a window titled "Lääkelaskuohjelma-Testiosio" with standard window controls (minimize, maximize, close) in the top right corner. The main content area contains five rows, each representing a question. Each row has a question number and text on the left, a light blue rectangular input field in the middle, and a unit label on the right. The questions are:

- 1. Ensimmäinen kysymys... millilitraa
- 2. Toinen kysymys... grammaa
- 3. Kolmas kysymys... desilitraa
- 4. Neljäs kysymys... milligrammaa
- 5. Viides kysymys... tablettia

At the bottom of the window, there are two buttons: "Keskeytä" (Stop) on the left and "Koe valmis!" (Exam finished!) on the right. A vertical scrollbar is visible on the right side of the question list.

Kuva 8: Testinäkömön kysymykset

Vastausten lähettämisen jälkeen ohjelma tarkistaa vastaukset ja kertoo tuloksen välittömästi käyttäjälle. Jos käyttäjä vastasi kaikkiin kysymyksiin oikein, ohjelma pyytää käyttäjää antamaan nimensä ja henkilötunnuksensa ja antaa käyttäjän tulostaa kolme läpipääsytodistusta. Jos käyttäjällä oli yksikin väärä vastaus, ohjelma näyttää väärin menneet kysymykset ja niiden oikeat vastaukset. Todistusten tulostuksen tai väärin vastauksien lukemisen jälkeen käyttäjä voi sulkea testinäkömään.

Lääkelaskuohjelma-Testiosio ☐ ☐ ☒

Onneksi olkoon! Läpäisit kokeen onnistuneesti tuloksella 10/10!

Todistuksen tulostus

| | | |
|-------------------|----------------------|---|
| Sukunimi, Etunimi | <input type="text"/> | Jos tulostuksessa on ongelmia, niin kutsu kokeen valvoja paikalle, äläkä sulje tätä ikkunaa! |
| Henkilötunnus | <input type="text"/> | |

Kuva 9: Testi suoritettu onnistuneesti

5.1.5 Ylläpitonäkymä

Läakelaskuohjelma-ylläpito-osio

Asetukset Harjoittelukysymykset Koekysymykset Todistus Salasana

Tulostusasetukset

Oletustulostin Karstula/atk/printer5

Tulosteiden lukumäärä 3

Tiedostojen sijainti

Teoria F:\group\laakelasku\teoria\ Selaa

Kysymykset F:\group\laakelasku\kysymykset\ Selaa

Muut asetukset

Harjoittelussa vaadittavien oikeiden vastausten lkm 5

Testin kysymysten jakauma

| | |
|-----------------|-----------|
| Murtoluvut | 2 |
| Desimaaliluvut | 2 |
| Prosenttiluvut | 3 |
| Verrannot | 3 |
| Yhteensä | 10 |

Tallenna Palaa päävalikkoon

Kuva 10: Ylläpitonäkymä

Ylläpitonäkymässä käyttäjä voi

- vaihtaa salasanan
- muuttaa harjoitusjaksossa vaadittavaa oikeiden vastausten määrää
- muuttaa koekysymysten jakaumaa
- muuttaa asetustiedostojen sijaintia
- vaihtaa oletustulostinta ja tulostettavien todistusten lukumäärää
- lisätä, muuttaa ja poistaa kysymyksiä
- tulostaa läpipääsytodistuksen.

Käyttäjä voi muokata asetuksia haluamassaan järjestyksessä. Muutokset tallennetaan vasta käyttäjän komennosta.

Lääkelaskuohjelma-ylläpito-osio [-] [] [X]

| Asetukset | Harjoittelukysymykset | Koekysymykset | Tulostus | Salasana | | | | | | | | | | | | | | | | | | | | |
|--|---|--------------------|---|---|------------|--------------------------------|-----------------|---|---------------|--------------------|-----------------------------------|--------------------|---------------------|---|--------------------|--|-----------------|--|--------|-------------------------------|---|-----------------|--|--|
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">ID</th> <th style="width: 15%;">Laskutyyppi</th> <th style="width: 15%;">Vaikeusaste</th> <th style="width: 65%;">Kysymys</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="text-align: center;"><Lisää uusia></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Verranto</td> <td style="text-align: center;">Normaali</td> <td>1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">Murtoluku</td> <td style="text-align: center;">Helppo</td> <td>Paljonko on 50 % luvusta 100.</td> </tr> </tbody> </table> | | | | | ID | Laskutyyppi | Vaikeusaste | Kysymys | <Lisää uusia> | | | | 1 | Verranto | Normaali | 1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään | 2 | Murtoluku | Helppo | Paljonko on 50 % luvusta 100. | | | | |
| ID | Laskutyyppi | Vaikeusaste | Kysymys | | | | | | | | | | | | | | | | | | | | | |
| <Lisää uusia> | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Verranto | Normaali | 1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään | | | | | | | | | | | | | | | | | | | | | |
| 2 | Murtoluku | Helppo | Paljonko on 50 % luvusta 100. | | | | | | | | | | | | | | | | | | | | | |
| <p>Valittu rivi</p> <table style="width: 100%;"> <tr> <td style="width: 15%;">ID:</td> <td style="width: 10%;"><input type="text" value="1"/></td> <td style="width: 15%;">Kysymys:</td> <td colspan="2" style="border: 1px solid black; padding: 2px;">1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään 800 000 yksikköön?</td> </tr> <tr> <td>Koekysymys:</td> <td><input type="text" value="Ei"/> ▼</td> <td>Laskukaava:</td> <td style="border: 1px solid black; padding: 2px;">$800000/1000000*50$</td> <td>Vastaus: <input type="text" value="40"/> Yksikkö: <input type="text" value="ml"/></td> </tr> <tr> <td>Vaikeusaste</td> <td><input type="text" value="Normaali"/> ▼</td> <td>Vihje 1:</td> <td colspan="2" style="border: 1px solid black; padding: 2px;">Laske ensin sekoitettavien yksiköiden suhde.</td> </tr> <tr> <td>Laskutyyppi</td> <td><input type="text" value="Verranto"/> ▼</td> <td>Vihje 2:</td> <td colspan="2" style="border: 1px solid black; padding: 2px;">Suhteen saat laskettua jakamalla 800 000 / 1 000 000</td> </tr> </table> <p style="text-align: right; margin-top: 10px;">Poista kysymys</p> <p style="text-align: right; margin-top: 10px;">Palaa päävalikkoon</p> | | | | | ID: | <input type="text" value="1"/> | Kysymys: | 1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään 800 000 yksikköön? | | Koekysymys: | <input type="text" value="Ei"/> ▼ | Laskukaava: | $800000/1000000*50$ | Vastaus: <input type="text" value="40"/> Yksikkö: <input type="text" value="ml"/> | Vaikeusaste | <input type="text" value="Normaali"/> ▼ | Vihje 1: | Laske ensin sekoitettavien yksiköiden suhde. | | Laskutyyppi | <input type="text" value="Verranto"/> ▼ | Vihje 2: | Suhteen saat laskettua jakamalla 800 000 / 1 000 000 | |
| ID: | <input type="text" value="1"/> | Kysymys: | 1 000 000 yksikköön penisiliiniä lisätään 50 ml nestettä Kuinka monta ml nestettä lisätään 800 000 yksikköön? | | | | | | | | | | | | | | | | | | | | | |
| Koekysymys: | <input type="text" value="Ei"/> ▼ | Laskukaava: | $800000/1000000*50$ | Vastaus: <input type="text" value="40"/> Yksikkö: <input type="text" value="ml"/> | | | | | | | | | | | | | | | | | | | | |
| Vaikeusaste | <input type="text" value="Normaali"/> ▼ | Vihje 1: | Laske ensin sekoitettavien yksiköiden suhde. | | | | | | | | | | | | | | | | | | | | | |
| Laskutyyppi | <input type="text" value="Verranto"/> ▼ | Vihje 2: | Suhteen saat laskettua jakamalla 800 000 / 1 000 000 | | | | | | | | | | | | | | | | | | | | | |

Kuva 11: Kysymysten päivitystä

Kysymysten päivitys tapahtuu siten, että yhdellä välilehdellä näytetään kaikki harjoitus-kysymykset ja yhdellä välilehdellä kaikki koekysymykset. Käyttäjä voi muokata kysymystä valitsemalla rivin taulukosta ja kun käyttäjä valitsee toisen rivin tai sulkee ikkunan, hänen tekemät muutoksensa tallennetaan. Uuden kysymyksen käyttäjä voi lisätä valitsemalla taulukosta tyhjän rivin.

Kysymysten poisto tapahtuu painamalla kyseisen rivin alussa olevaa poista-nappulaa. Nappulan painamisen jälkeen ohjelma varmistaa, että käyttäjä haluaa poistaa kysymyksen ja tarkistaa, että ko. kysymyskategoriaan jää riittävä määrä kysymyksiä koetta varten.

Läkelaskuohjelma-ylläpito-osio

| | | | | |
|-----------|-----------------------|---------------|----------|----------|
| Asetukset | harjoittelukysymykset | Koekysymykset | Tulostus | Salasana |
|-----------|-----------------------|---------------|----------|----------|

Salasanan vaihto

Vanha salasana:

Uusi salasana:

Uusi salasana uudelleen:

Vaihda salasana

Palaa päävalikkoon

Kuva 12: Salasanan vaihto

Läkelaskuohjelma-ylläpito-osio

Asetukset Harjoittelukysymykset Koekysymykset **Tulostus** Salasana

Tulostus

Sukunimi, etunimi:

Henkilötunnus:

Tulosta

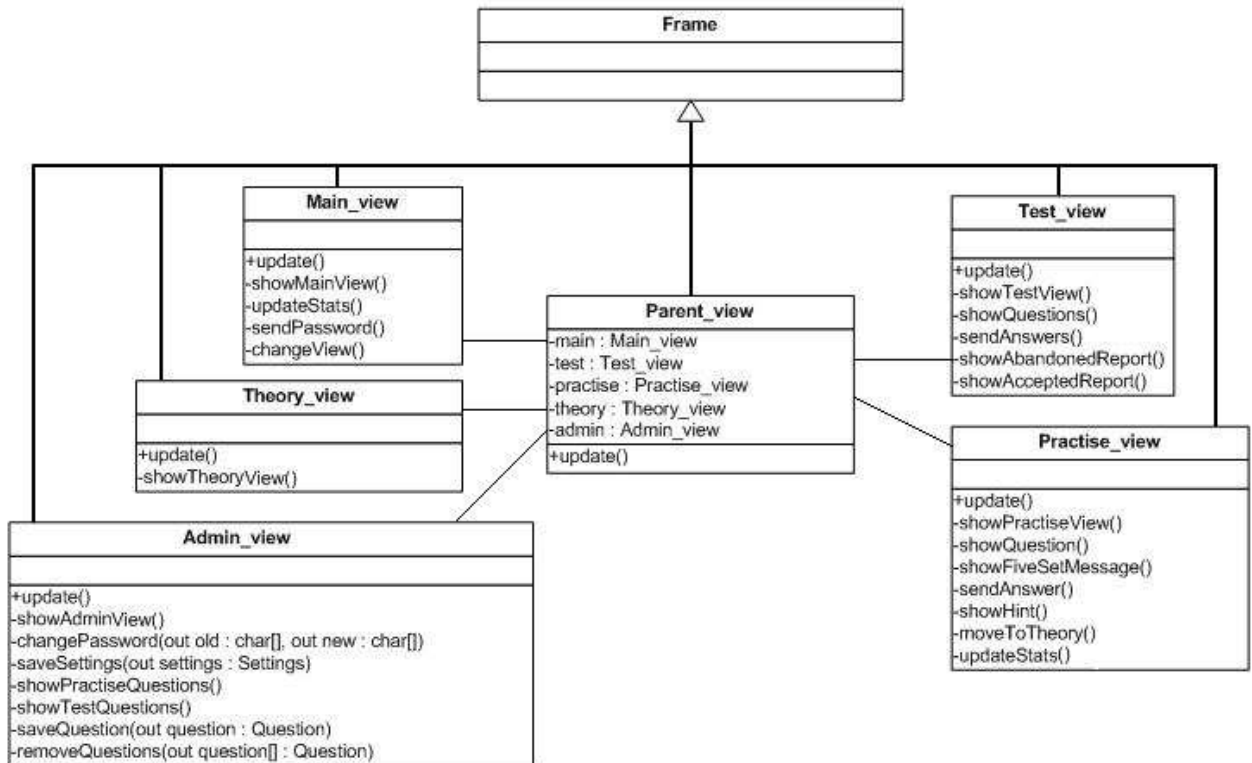
Palaa päävalikkoon

Kuva 13: Läpikäysytodistusten tulostaminen

5.2 Tekninen kuvaus

Seuraavassa kuvataan järjestelmän tekninen toteutus kunkin näkymän osalta.

5.2.1 Yleisrakenne



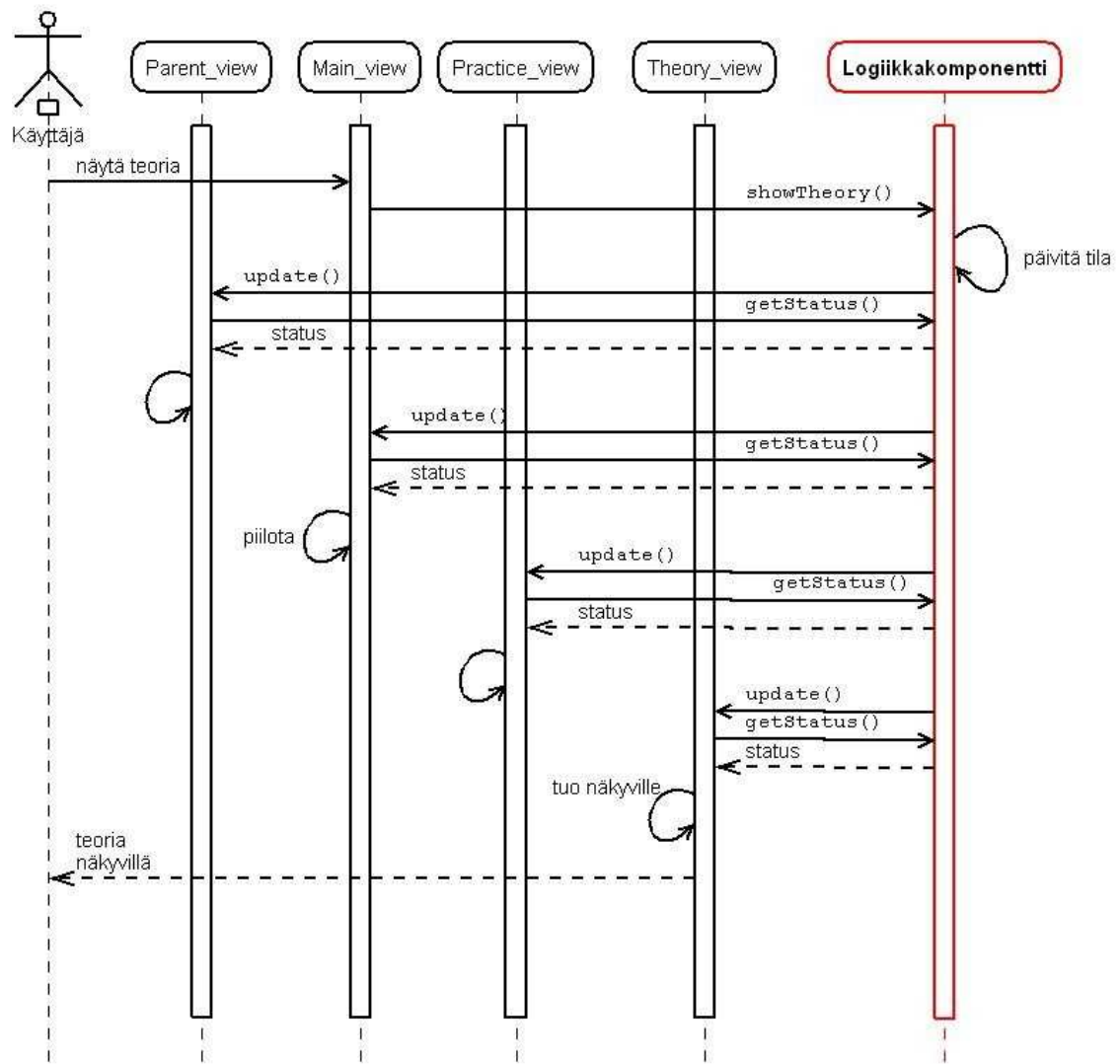
Kuva 14: Käyttöliittymän luokkakaavio

Käyttöliittymä käyttää logiikkakomponenttia ohjelman tilatietojen säilyttämiseen sekä käyttäjälle näytettävien tietojen hakemiseen tietovarastoista.

Käyttöliittymä koostuu viidestä näkymästä, joista jokaista varten on oma luokkansa. Lisäksi tarvitaan vielä yksi ”äitinäkymä”, jonka tehtävänä on vastata ikkunoiden yhteisistä osista (mahdolliset valikkopalkit ym.) sekä huolehtia käyttäjälle näkyvien ikkunoiden vaihtamisesta.

5.2.2 Tarkkailija-malli

Kun käyttöliittymä haluaa vaikkapa käyttäjän syötteiden perusteella vaihtaa näkyvissä olevan näkymän toiseen, se kutsuu logiikkakomponentin näkyviin tuotavaa näkymää vas-



Kuva 15: Tarkkailija-mallin sekvenssikaavio

taavaa `moveTo[Name]()`-metodia. Logiikkakomponentti päivittää sisäisen tilansa vastaamaan käyttöliittymältä tullutta pyyntöä ja sen jälkeen kutsuu sille rekisteröityneiden tarkkailijoiden eli kaikkien näkymien `update()`-metodeja. Näin jokainen näkymä käy tarkistamassa logiikkakomponentilta, pitääkö sen olla näkyvillä vai ei.

Samalla tavalla jokainen näkymä tarkistaa `update()`-metodissa tarvitsemansa muut tilatiedot (kuten tilastot) logiikkakomponentilta. Kun logiikkakomponentti päivittää sisäisiä laskureitaan, se kutsuu myös kaikkien tarkkailijoiden `update()`-metodia. `update`-metodi puolestaan kyselee logiikalta päivittyneet tilastot ja hoitaa oman olionsa näkyvien osien päivityksen.

5.2.3 ParentView

Kun ohjelma käynnistetään, kaikista näkymäluokista luodaan yksi ilmentymä. Parent_view-luokka säilyttää viitteet kaikkiin näkymäolioihin. Lisäksi Parent_view-olio sisältää kaikille näkymille yhteiset osat kuten valikkopalkin.

5.2.4 MainView

Päänäkymä tarvitsee näytölleen vain tilastotietoja, joita se hakee logiikkakomponentilta. Tämä tilastotietojen päivityslogiikka on kuvattu kohdassa ”tarkkailijamalli”.

5.2.5 TheoryView

Teorianäkymä pyytää käyttäjän valitseman osa-alueen mukaisen sisällön Logiikkakomponentilta. Sisältö palautetaan sille valmiiksi muotoiltuna HTML-tekstinä.

Teorianäkymä tarkistaa aina tullessaan näkyviin logiikkakomponentilta, ollaanko harjoittelutilassa vai ei.

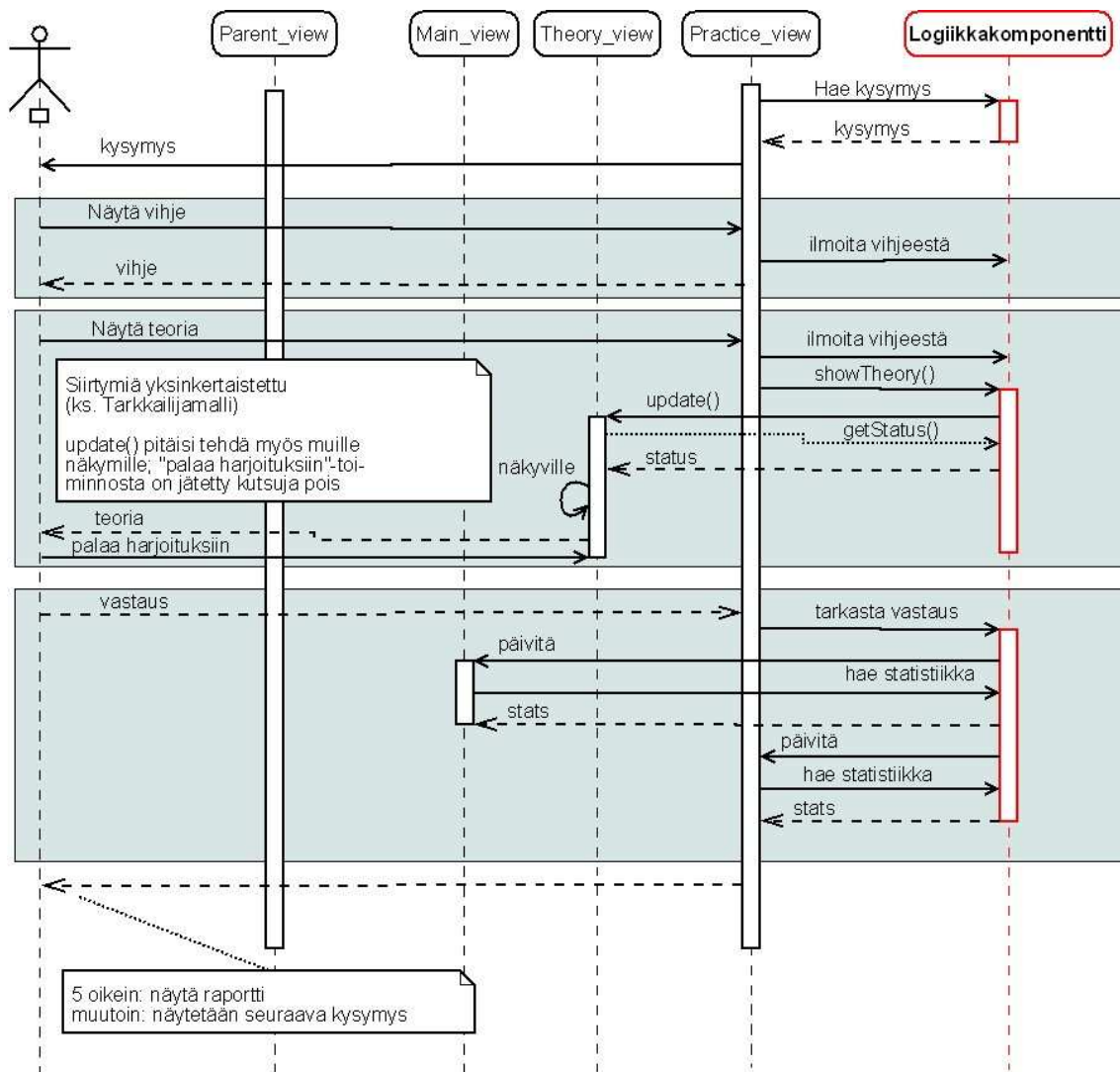
5.2.6 PractiseView

Harjoittelunäkymä hakee näytettävän kysymyksen logiikkakomponentilta. Kysymys on Question-olio, joka sisältää kaikki kysymykseen kuuluvat tiedot (itse kysymys, vihje1, vihje2, laskukaava, vastaus, yksikkö jne.).

Kun käyttäjä vastaa, vastauksen oikeamuotoisuus tarkistetaan jo käyttöliittymässä: (Answer-olio huolehtii siitä). Jos vastaus ei ole oikeanmuotoinen, käyttäjän annetaan korjata vastausta muuttamatta ohjelman tilaa mihinkään.

Kun vastaus on tarkistettu oikeanmuotoiseksi, harjoittelunäkymä lähettää vastauksen tarkastettavaksi logiikkakomponentille (vaikka vastaus jo olisikin käyttöliittymän tiedossa). Jos vastaus oli oikein, logiikkakomponentti päivittää sisäisiä laskureitaan ja kutsuu sille rekisteröityneitä tarkkailijoiden updateStats-metodia aiemmin kuvatun tarkkailijamallin mukaisesti.

Jos käyttäjä itse haluaa vihjeen tai hän siirtyy teoriaosioon, harjoittelunäkymä ilmoittaa siitä logiikkakomponentille notifyHintUsed-kutsulla. Logiikkakomponentti tarvitsee tämän tiedon, jotta seuraavasta oikeasta vastauksesta ei anneta pistettä, koska käyttäjä tarvitsi vihjeitä.



Kuva 16: Käyttöliittymän ja harjoittelukomponentin sekvenssikaavio

5.2.7 TestView

Logiikkakomponentti säilyttää tilatietoa siitä, onko käyttäjä kirjautuneena sisään. Testinäkömä on tarkistettava tämä joka kerta tullessaan näkyville.

Kun käyttäjä aloittaa kokeen, testinäkömä hakee kaikki testikysymykset logiikkakomponentilta ja näyttää ne kerralla käyttäjälle. Kun käyttäjä on vastannut kaikkiin kysymyksiin, käyttöliittymä tarkistaa niiden oikeanmuotoisuuden samalla tavalla kuin harjoitteluosiossa (Answer-olio hoitaa). Oikeinmuodostetut vastaukset lähetetään logiikkakomponentille tarkastettavaksi.

Logiikkakomponentti tallentaa tiedot vastauksista omiin laskureihinsa ja kutsuu tarkkailijoiden update()-metodeja. TestView kysyilee omassa update-metodissaan tiedot oikeista vastauksista. Tämän perusteella testinäkömä näyttää väärin menneet kysymykset ja niiden oikeat vastaukset (ne ovat jo tiedossa Question-olioissa). Jos kaikki vastaukset menivät oikein, testinäkömä lähettää käyttäjän syöttämät nimen ja henkilötunnuksen logiikkakomponentille tulostusta varten.

5.2.8 AdminView

Ylläpitoinäkömä vaatii salasanan samalla tavalla kuin testinäkömäkin.

Salasanan vaihto tapahtuu nappia painamalla, ensin ylläpitoinäkömä pyytää logiikkakomponenttia tarkastamaan vanhan salasanan ja sen jälkeen lähettää uuden salasanan logiikkakomponentille tallennettavaksi.

Kysymysten ylläpito hoidetaan niin, että ylläpitoinäkömä saa logiikkakomponentilta kaikki harjoitus- tai testikysymykset (riippuen välilehdestä) joko Question[]-taulukkona tai jonain muuna sopivana tietorakenteena. Käyttäjälle näytetään aina valitun rivin mukaisen Question-olion tiedot, joita hän voi muokata. Muokatut tiedot tallennetaan siten, että ylläpitoinäkömä välittää koko Question-olion logiikkakomponentille tallennettavaksi.

Kysymystä lisättäessä ylläpitoinäkömä pyytää logiikkakomponentilta uuden Question-olion, jonka se käyttäjän syötteiden jälkeen välittää logiikkakomponentille tallennettavaksi samalla tavalla kuin kysymystä päivitettäessä. Kysymysten poistossa kerätään käyttäjän valitsemat Question-oliot taulukoksi, joka taas välitetään logiikkakomponentille.

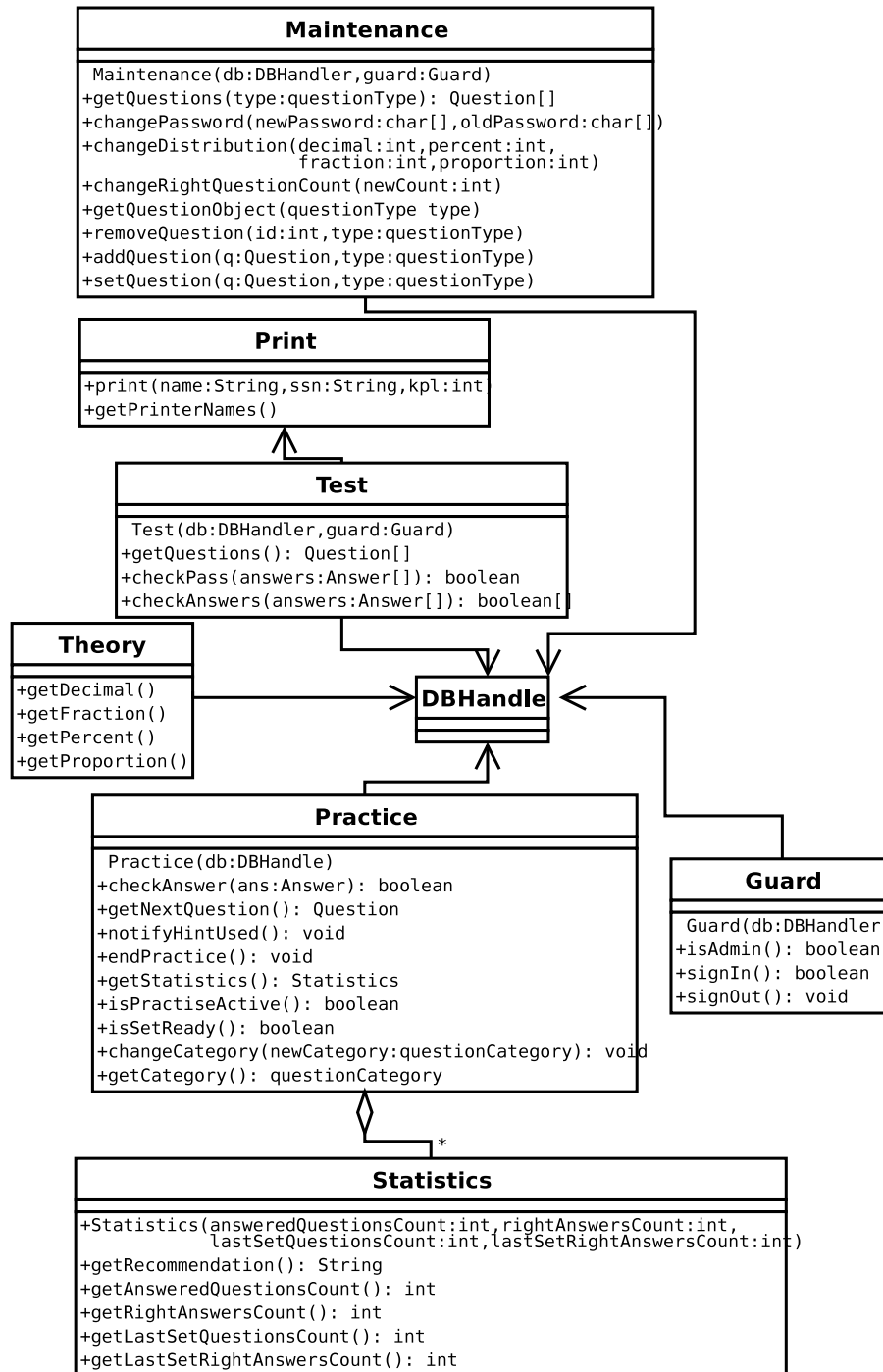
Jos käyttäjä haluaa muuttaa kysymyksen harjoituskysymyksestä koekysymykseksi tai päinvastoin, ylläpitoinäkömä välittää vaihdettavan kysymyksen logiikkakomponentille, joka tekee muutoksen. Tämän jälkeen logiikkakomponentti kutsuu tarkkailijamallin mukaisesti tarkkailijoiden update-metodeja, ja ylläpitoinäkömä päivittää listansa.

6 Logiikkajärjestelmä

Ohjelmalogiikkajärjestelmä koostuu useista logiikkaluokista, joiden avulla prosessoidaan käyttäjän syötteitä ja pidetään tietoa yllä käyttäjän toimenpiteistä. Logiikkajärjestelmä

koostuu kuudesta pääluokasta ja niiden käyttämistä apuluokista. Pääluokat ovat Test, Practice, Print, Maintenance, Guard ja Theory.

6.1 Luokkakaavio



Kuva 17: Logiikkajärjestelmän luokkakaavio

6.2 Luokkien rajapintakuvaukset

6.2.1 Practice

Luokka vastaa käyttäjän harjoittamisen logiikasta ja harjoitustilastotiedon ylläpidosta. Tämän jälkeen harjoittelulogiikka arpoa seuraavan kysymyksen helppojen joukosta, kunnes käyttäjä on vastannut yhteen oikein, minkä jälkeen annetaan kysymys normaalien kysymysten joukosta.

Mikäli käyttäjä pyytää vihjeitä, niin luokasta luotu olio pitää siitä kirjata ja merkkaa vastauksen niin, ettei sitä lasketa kokonaistilastoissa oikein menneeksi vastaukseksi. Samoin tehdään myös käyttäjän siirtyessä lukemaan laskujen teoriaa.

Konstruktorit

Public Practice(DBHandle db)

Metodit

CheckAnswer

public boolean checkAnswer(Answer ans)

Tarkistaa, onko annettu vastaus oikein.

Palauttaa

True, jos vastaus oli oikein, muuten false

Argumentit

ans Vastaus, jonka oikeellisuus tarkistetaan

Poikkeukset

getNextQuestion

public Question getNextQuestion() throws DBLogicException, NotEnoughQuestionsException

Palauttaa seuraavan harjoittelukysymyksen. Jos käyttäjä ei ole vielä vastannut yhtenkään kysymykseen oikein, palauttaa kysymyksen, joka on tyypiltään helppo.

Palauttaa

Seuraavan harjoittelukysymyksen

Argumentit

Poikkeukset

DBLogicException

Tietokantavirhe.

NotEnoughQuestionsException

Tietokannassa ei ole enää uusia osa-alueen tai vaikeustason mukaisia kysymyksiä.

notifyHintUsed*public void notifyHintUsed()*

Kertoo luokalle Practice, että käyttäjä otti vihjeen tai siirtyi lukemaan teoriaa. Kun tätä metodia on kutsuttu, luokka Practice ei enää anna käyttäjälle pistettä nykyisestä kysymyksestä, vaikka käyttäjä vastaisikin siihen oikein.

Palauttaa**Argumentit****Poikkeukset****endPractice***public void endPractice()*

Lopettaa harjoittelun. Nykyisen kysymyssarjan tiedot nollataan eli harjoittelu alkaa alusta.

Palauttaa**Argumentit****Poikkeukset****getStatistics***public Statistics getStatistics()*

Palauttaa Statistics-tietueen, jossa on tilastotietoa harjoittelusta.

Palauttaa

Tilastotietoa harjoittelusta Statistics-tietueessa

Argumentit**Poikkeukset****isSetReady***public boolean isSetReady()*

Palauttaa tiedon, onko käyttäjä vastannut oikein niin moneen kysymykseen, että nykyinen harjoittelusarja on valmis.

Palauttaa

True, jos nykyinen harjoittelusarja on valmis, muuten false

Argumentit**Poikkeukset****changeCategory**

public void changeCategory(questionCategory newCategory)

Vaihtaa harjoittelussa kysyttävien kysymysten kategoriaa. Samalla nollaa nykyisen kysymyssarjan tiedot kuten endPractice().

Palauttaa

Argumentit

newCategory Uusi kysymysten kategoria harjoittelussa.

Poikkeukset

getCategory

public questionCategory getCategory()

Palauttaa tiedon, mitä kysymyskategoriaa harjoittelussa tällä hetkellä ollaan harjoittelemassa.

Palauttaa

Harjoittelussa tällä hetkellä kysyttävien kysymysten kategorian

Argumentit

Poikkeukset

6.2.2 Statistics

Tietue sisältää tilastot käyttäjän suorittamasta harjoittelusta. Tietueen tiedot eivät päivity automaattisesti, vaan uusin tilanne haetaan luokalta Practice metodilla getStatistics().

Konstruktorit

public Statistics(int answeredQuestionsCount, int rightAnswersCount, int lastSetQuestionsCount, int lastSetRightAnswersCount)

Luo Statistics-tietueen. Asettaa tietueen kentät konstruktorin vastaavan nimisten argumenttien mukaisiksi.

Metodit

getAnsweredQuestionsCount

public int getAnsweredQuestionsCount()

Palauttaa tiedon siitä, kuinka moneen harjoittelukysymykseen käyttäjä on vastannut ohjelman suoritusaikana.

getRightAnswersCount

public int getRightAnswersCount()

Palauttaa tiedon, kuinka moneen harjoittelukysymykseen käyttäjä on vastannut oikein ohjelman suoritusaikana.

getLastSetQuestionsCount*public int getLastSetQuestionsCount*

Palauttaa tiedon, kuinka moneen harjoittelukysymykseen käyttäjä on vastannut edellisen valmistuneen harjoittelusarjan aikana.

getLastSetRightAnswersCount*public int getLastSetRightAnswersCount*

Palauttaa tiedon, kuinka moneen harjoittelukysymykseen käyttäjä on vastannut oikein edellisen valmistuneen harjoittelusarjan aikana.

getRecommendation*public String getRecommendation()*

Palauttaa käyttäjälle annettavan tekstimuotoisen suosituksen, kannattaako hänen jatkaa harjoittelua. Suositus perustuu edelliseen valmistuneeseen harjoittelusarjaan.

6.2.3 Maintenance

Luokka vastaa ylläpito näkymän toimintalogiikasta.

Konstruktorit*Maintenance (DBHandler db)***Metodit****getQuestions***Question[] getQuestions(questionType type)*

Palauttaa taulukossa kaikki tietokannassa olevat kysymykset.

Palauttaa

Question[]-taulukon kaikista tietokannassa olevista koekysymyksistä.

Argumentit

type Palautetaanko koe- vai testikysymykset.

Poikkeukset**changePassword***public boolean changePassword(char[] newPassword, char[] oldPassword) throws DBLogicException*

Vaihtaa ylläpitäjän salasanan. Jos vanha salasana on virheellinen, salasanaa ei vaihdeta.

Palauttaa

True, jos vanha salasana oli väärin, muuten false.

Argumentit

newPassword Uusi salasana.

oldPassword Vanha salasana.

Poikkeukset

DBLogicException

Tietokantavirhe.

changeDistribution

public void changeDistribution(int decimal, int percent, int fraction int proportion)

Muuttaa koekysymysten jakauman tietueen dist mukaiseksi.

Palauttaa**Argumentit**

decimal Uusi desimaalikysymysten lukumäärä.

percent Uusi prosenttikysymysten lukumäärä.

fraction Uusi murtolukukysymysten lukumäärä.

proportion Uusi verrantokysymysten lukumäärä.

Poikkeukset**changeRightQuestionCount**

public void changeRightQuestionCount(int newCount)

Muuttaa yhden harjoittelusarjan päättymiseen vaadittavien oikeiden vastausten määrää.

Palauttaa**Argumentit**

newCount Uusi oikeiden vastausten määrä.

Poikkeukset**getQuestionObject**

Lisää *getQuestionObject(questionType type)*

Palauttaa Question-olion, joka sisältää seuraavan vapaan id-numeron.

Palauttaa

Question-olion vapaalla id-numerolla.

Argumentit

type Harjoitus- vai testikysymys.

Poikkeukset

RemoveQuestion

public void removeQuestion(int id, questionType type) throws InvalidQuestionIDException

Poistaa tietokannasta kysymyksen, jonka tunnus on id ja tyyppi type.

Palauttaa

Argumentit

id Poistettavan kysymyksen tunnus.

type Harjoitus- vai testikysymys.

Poikkeukset

InvalidQuestionIDException

Tietokannassa ei ollut kysymystä argumenttina annetulla id:llä

addQuestion

public void addQuestion(Question q, questionType type) throws InvalidQuestionIDException

Lisää tietokantaan kysymyksen q.

Palauttaa

Argumentit

q Tietokantaan lisättävä kysymys.

type Harjoitus- vai testikysymys.

Poikkeukset

InvalidQuestionIDException

Tietokannassa oli jo kysymys samalla id:llä kuin argumenttina annetussa Question-oliossa.

setQuestion

public void setQuestion(Question q, questionType type) throws InvalidQuestionIDException

Muuttaa tietokannassa olevan kysymyksen, jolla on sama tunnus ja tyyppi kuin kysymyksellä q, kysymyksen q mukaiseksi.

Palauttaa

Argumentit

q Muutettavan kysymyksen kaikki kentät korvataan argumenttina annetun olion kentillä.

type Harjoitus- vai testikysymys.

Poikkeukset

InvalidQuestionIDException

Tietokannassa ei ollut kysymystä samalla id:llä kuin argumenttina annetussa Question-oliossa.

6.2.4 Test

Luokka toteuttaa testitilan logiikan. Luokasta luodaan testiä aloitettaessa uusi olio.

Ensimmäiseksi luokasta luotu olio pyytää tietokannalta eri osa-alueilta näytettävien kysymysten jakaumat. Tämän jälkeen komponentti arpoo, mitkä kysymykset käyttäjälle välitetään.

Kun käyttöliittymäkomponentti on välittänyt käyttäjän syöttämät vastaukset Test-luokasta rakennetulle oliolle, niin olio tarkistaa vastausten oikeellisuuden. Pyydettyessä olio lähettää boolean-taulukon vastauksista käyttöliittymäkomponentille.

Jos kaikki vastaukset olivat oikein, niin komponentti myös pyydettyessä tulostaa läpipääsytodistuksen.

Konstruktorit

Public Test(DBHandler db) throws DBLogicException, NotEnoughQuestionsException

Poikkeukset

DBLogicException

Tietokantavirhe

NotEnoughQuestionsException

Tietokannassa ei ole testikysymyksiä vähintään yhtä paljon kuin on asetettu yhteensä eri osa-alueiden kysymyslukumääräksi.

Metodit

getQuestions

public Question[] getQuestions()

Palauttaa kokeessa käyttäjältä kysyttävät kysymykset. Kysymykset arvotaan satunnaisesti koekysymysten joukosta siten, että jokaisesta kysymyskategoriasta tulee se määrä kysymyksiä, mitä ohjelman asetusvalikossa on määrätty.

Palauttaa

Kokeessa kysyttävät kysymykset

Argumentit

Poikkeukset

checkPass

public boolean checkPass(Answer[] answers)

Tarkistaa, pääsiko käyttäjä kokeesta läpi.

Tarkistaa sen vertaamalla answers-taulukossa olevia käyttäjän antamia vastauksia viime getQuestions()-metodin kutsulla annettuihin kysymyksiin.

Palauttaa

True, jos käyttäjä pääsi läpi, muuten false.

Argumentit

answers Vastaukset, joiden pohjalta läpipääsy päätetään.

Poikkeukset

checkAnswers *public boolean[] checkAnswers(Answer[] answers)*

Tarkistaa, mitkä answers-taulukon vastaukset ovat oikeat. Palauttaa totuusarvotaulukon, jossa jokainen alkio vastaa samanindeksistä vastausta answers-taulukossa, ja sen arvo on true, jos vastaus oli oikea, muuten false.

Palauttaa

Totuusarvotaulukon vastausten oikeellisuudesta

Argumentit

answers vastaukset, joiden oikeellisuus tarkistetaan

Poikkeukset**6.2.5 Print**

Luokka vastaa testin onnistuessa todistusten tulostuksesta.

Konstruktorit

public Print()

Metodit

setPersonalInformation(String name, String ssn) throws SSNInvalidException

Metodi välittää kokeen suorittaneen henkilön nimen ja henkilötunnuksen tulostettavaksi.

Argumentit

name Henkilön nimi, jolle todistus tulostetaan.

ssn Henkilötunnus, jolle todistus tulostetaan

Poikkeukset

SSNInvalidException

Henkilötunnuksen tarkistusmerkki oli väärä. Vaikka metodi heittäisikin poikkeuksen, niin annettu nimi ja henkilötunnus tallennetaan joka tapauksessa todistukseen tulostettavaksi.

print

public boolean print(int kpl) throws PrinterException

Tulostaa käyttäjälle todistuksen läpipääsystä.

Argumentit

kpl tulostettavien todistusten lukumäärä

Palauttaa**Poikkeukset**

PrinterException

getPrinterNames

public String[] getPrinterNames()

Palauttaa String-taulukon järjestelmän tulostimista.

Palauttaa**Argumentit****Poikkeukset****6.2.6 Guard**

Luokka pitää kirjaa ohjelmiston käyttötilasta. Ohjelma on joko ylläpito- sekä testitilassa tai tavallisessa tilassa.

Konstruktorit

Guard(DBHandler db)

Metodit**isAdmin**

public boolean isAdmin()

Palauttaa tiedon siitä, onko ylläpitäjä kirjautunut ohjelmaan

Palauttaa

True, jos ylläpitäjä on kirjautunut ohjelmaan, muuten falsen.

Argumentit

Poikkeukset**signIn**

public boolean signIn(char[] password) throws DBLogicException

Kirjaa ylläpitäjän sisään ohjelmaan. Palauttaa tiedon siitä, oliko salasana oikea.

Palauttaa

True, jos salasana oli oikea, muuten false

Argumentit

password Salasana, jolla ohjelmaan yritetään kirjautua.

Poikkeukset

DBLogicException

Tietokantavirhe

signOut

public void signOut()

Kirjaa ylläpitäjän ulos ohjelmasta.

Argumentit**Palauttaa****Poikkeukset****6.2.7 Theory**

Theory-luokalta käyttöliittymä voi pyytää eri osa-alueilta näytettävien teoriadokumenttien polut.

Konstruktorit

Theory(DBHandler db)

Metodit**getDecimal()**

public String getDecimal()

Palauttaa polun desimaaliteoriaan.

getPercent()

public String getPercent()

Palauttaa polun prosenttiteoriaan.

getFraction()

public String getFraction()

Palauttaa polun murtolukuteoriaan.

getProportion()

public String getProportion()

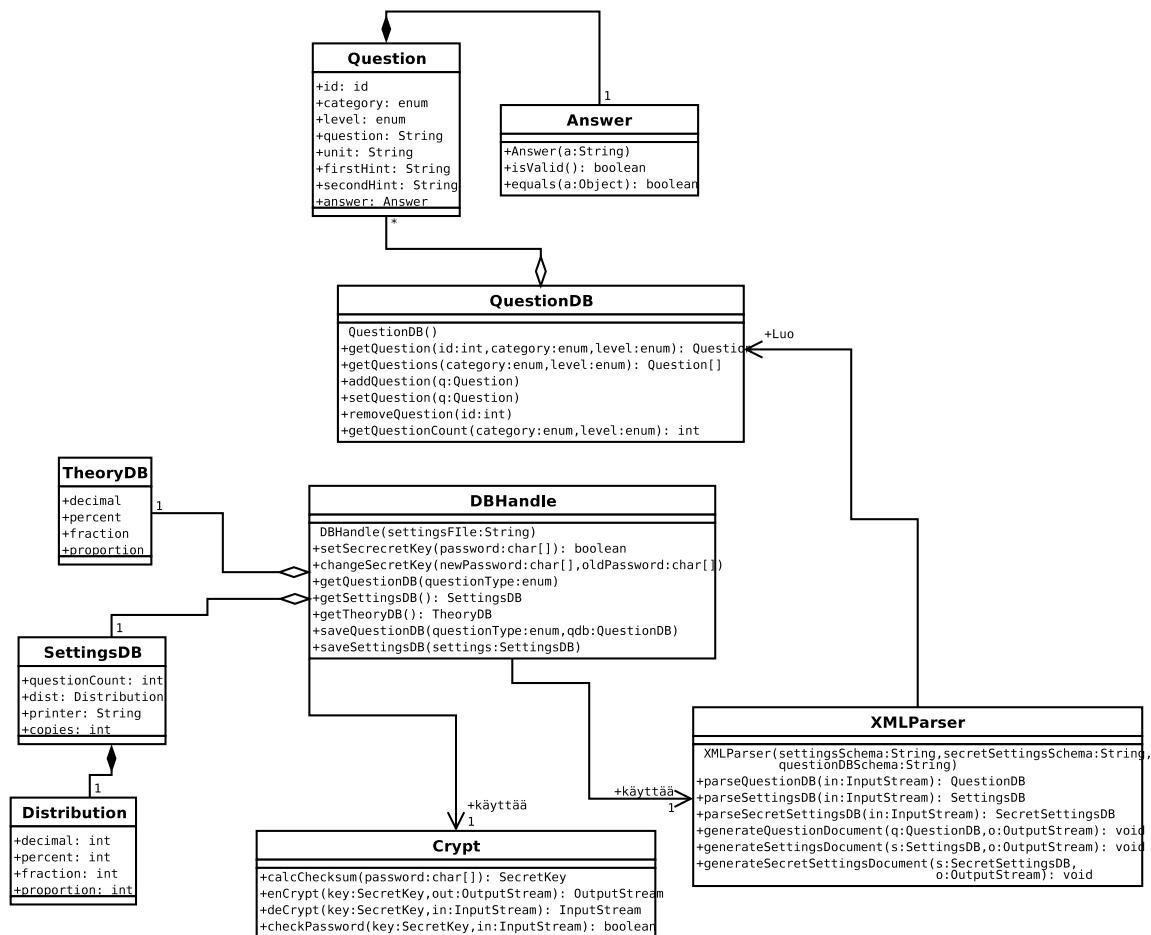
Palauttaa polun verrantoteoriaan.

7 Tietokanta

Tietokantajärjestelmä koostuu XML-käsittelykomponentista, salauskomponentista ja tietokantahallintakomponentista.

Hallintakomponentti vastaa abstraktiosta logiikkajärjestelmän ja tietokannan välillä. Sen tarjoamat operaatiot räätälöidään logiikkajärjestelmän käyttöön. XML- ja salauskomponentit ovat hallintakomponentin käyttämiä apukomponentteja, ja niiden palvelut ovat muille osajärjestelmille näkymättömiä.

7.1 Tietokantajärjestelmän luokkakaavio



Kuva 18: Tietokantajärjestelmän luokkakaavio

7.2 Tietokannan käyttämät tiedostot

- Asetustiedosto
 - Harjoitusosan kysymyssarjan kysymyslukumäärä
 - tulostinasetukset: tulosteiden lukumäärä, oletustulostin
 - polut teoriaosan HTML-tiedostoihin
 - polut kysymystiedostoihin
- Harjoituskysymystiedosto
 - kysymyksen id-numero

- osa-alue (desimaali, verranto, prosentti, jakolasku)
 - vaikeusaste (helppo vai normaali)
 - itse kysymys
 - vastaus
 - vastauksen yksikkö (ei pakollinen)
 - laskukaava
 - ensimmäinen vihje (ei pakollinen)
 - toinen vihje (ei pakollinen)
- Koekysymystiedosto
 - kysymyksen id
 - osa-alue
 - itse kysymys
 - vastaus
 - vastauksen yksikkö
 - laskukaava
- Koekymysten osa-aluejakauman sisältämä tiedosto
 - verrantokysymysten lukumäärä
 - prosenttikysymysten lukumäärä
 - murtolukukysymysten lukumäärä
 - desimaalikysymysten lukumäärä
- Salasanatarkistustiedosto
 - Salausavaimella salattu ”magicword”-merkkijono UTF-16-muodossa.

Lisäksi kantaan kuuluu neljä teorian dokumentoivaa HTML-sivua skeematiedostot XML-parserille.

7.3 Salauskomponentti

Salausmoduuli on apukomponentti, jota tietokantajärjestelmä käyttää tenttikysymystiedoston salaukseen ja salauksen purkuun. Salausmoduuli laskee käyttäjän antamasta salasanasta SHA-1-tiivistysalgoritmeilla salaisen avaimen.

7.4 XML-komponentti

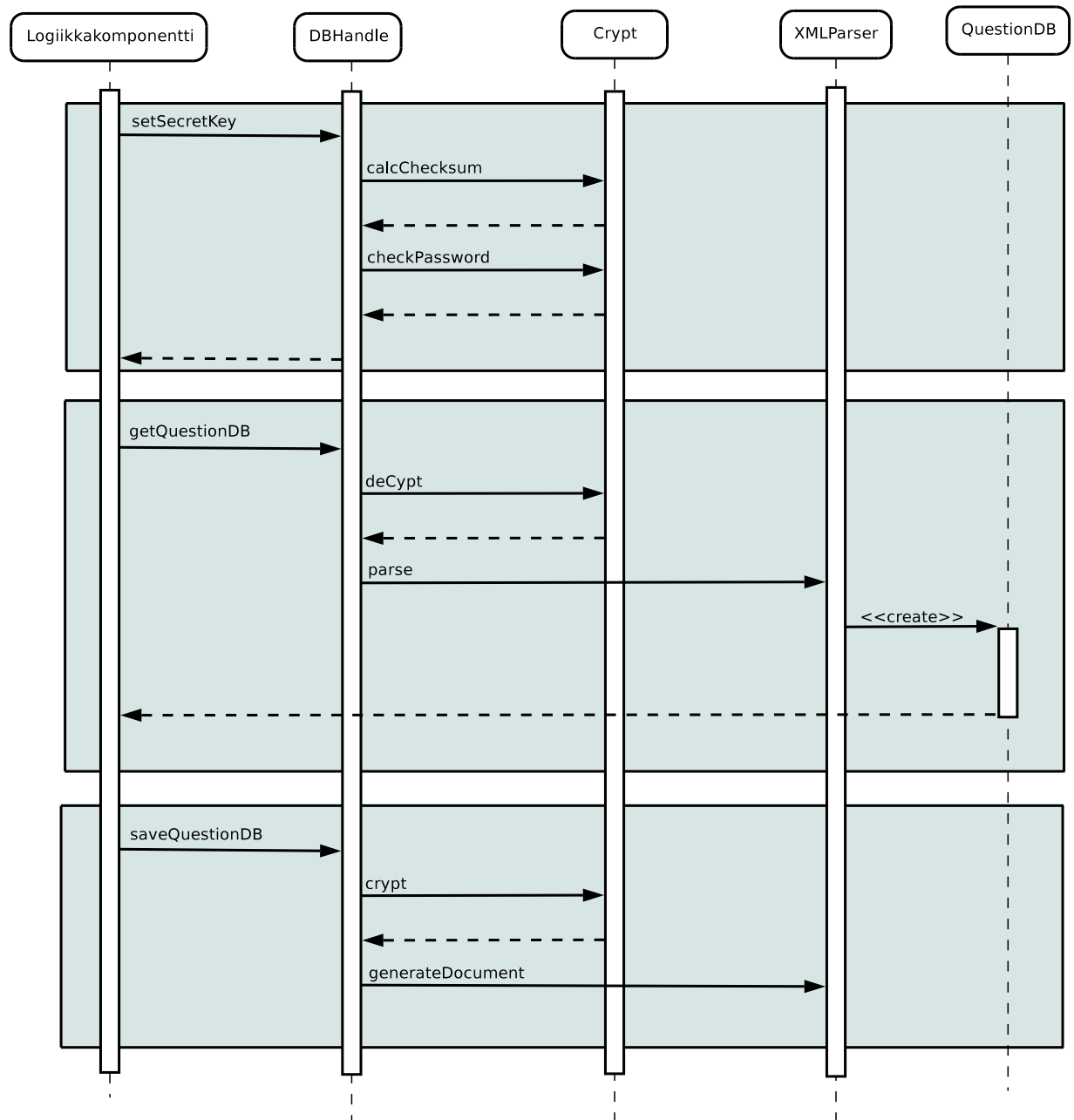
XML-komponentti jäsentee kysymystiedostot sekä palauttaa jäsenneetyt elementit sopivassa tietorakenteessa jatkokäsittelyä varten. Lisäksi komponentti generoi kysymystietorakenteesta XML-tiedoston.

Esimerkki XML-tiedostosta:

```
<questiondb>
  <question id="123" category="desimaali" level="helppo">
    <questionString>Montako litraa...</questionString>
    <answer>3</answer>
    <unit>litraa</unit>
    <formula>1+2=3</formula>
    <hint1>Laske ensin 1+2</hint1>
    <hint2>4-1</hint2>
  </question>
</questiondb>
```

7.5 Sekvenssikaavio tietokantajärjestelmän toiminnasta.

Ensimmäiseksi käyttäjä kirjautuu ylläpitotilaan. Seuraavaksi käyttäjä avaa testitilan. Lopuksi ylläpitäjä muuttaa tietokannassa olevia testikysymyksiä ja tallentaa uudet kysymykset.



Kuva 19: Sekvenssikaavio tietokantajärjestelmästä

7.6 Luokkien rajapintakuvaukset

7.6.1 DBHandle

Abstraktoi tietokannanhallinnan logiikkajärjestelmälle.

Konstruktorit

DBHandle(String settingsFile, String secretSettingsFile)

Metodit

public boolean setSecretKey(password char[]) throws DBException

Antaa tietokannalle käyttäjän salasanan, josta lasketaan SHA-1-tiivistysalgoritmilla salausavain.

Palauttaa

True, jos käyttäjän antamasta salasanasta onnistuttiin laskemaan sama salausavain kuin tietokannassa ennestään käytettiin. Epäonnistuessa metodi palauttaa falsen.

Argumentit

Password Käyttäjän salasana.

Poikkeukset

DBException

changeSecretKey

public boolean changeSecretKey(char[] newPassword, char[] oldPassword) throws DBException

Vaihtaa tietokannan käyttämää salausavainta. Jos metodin sama vanha salasana on oikea ja uusi tarpeeksi pitkä, niin metodi purkaa vanhalla salasanalla salatut tiedostot ja salaa kyseiset tiedostot uudella avaimella.

Salasanan pitää olla joko 0 tai vähintään 8 merkkiä pitkä. 0-merkkinen salasana tarkoittaa sitä, ettei salanaa ole asetettu lainkaan.

Palauttaa

Truen, jos vanha salasana oli oikea ja uusi salasana oli 0 tai yli 8 merkkiä.

Falsen, jos vanha salasana väärä tai uusi liian lyhyt.

Argumentit

newPassword Uusi salasana, josta generoidaan uusi avain.

oldPassword Vanha tietokannan käyttämä salasana.

Poikkeukset

DBException

getQuestionDB

public QuestionDB getQuestionDB(DBType type) throws DBException

Palauttaa viitteen kysymystietokantaan.

Argumentit

type Testi vai harjoitus.

Palauttaa

Viitteen QuestionDB-olioon.

Poikkeukset

DBException

getSettingsDB

public SettingsDB getSettingsDB() throws DBException

Palauttaa viitteen asetustietokantaan.

Argumentit**Palauttaa**

Viitteen SettingsDB-olioon.

Poikkeukset

DBException

getSecretSettingsDB

public SecretSettingsDB getSecretSettingsDB() throws DBException

Palauttaa viitteen koekysymysjakaumatietokantaan.

Argumentit**Palauttaa**

Viitteen SecretSettingsDB-olioon.

Poikkeukset

DBException

getTheoryDB

public TheoryDB getTheoryDB()

Palauttaa viitteen teoriatietokantaan.

Argumentit**Palauttaa**

Viitteen TheoryDB-olioon.

Poikkeukset

DBException

saveQuestionDB*Public void saveQuestionDB(enum DBType) throws DBException*

Tallentaa kysymystietokannan.

Argumentit*DBType* Koe- vai harjoituskysymystietokanta.**Palauttaa** Viitteen kysymystietokantaan.**Poikkeukset**

DBException

saveSettingsDB*Public void saveSettingsDB() throws DBException*

Tallentaa asetustietokannan.

Argumentit**Palauttaa****Poikkeukset**

DBException

saveSecretSettingsDB*Public void saveSettingsDB() throws DBException*

Tallentaa koekysymysjakaumatietokannan.

Palauttaa**Poikkeukset**

DBException

7.6.2 XMLParser

Luokka jäsentää XML-dokumentit.

Konstruktorit*public XMLParser(String settingsSchema, String secretSettingsSchema, String questionDBSchema)***Argumentit**

settingsSchema Polku asetustiedoston skeemaan.

secretSettingsSchema Polku salaisen asetustiedoston skeemaan.

questionDBSchema Polku kysymystiedoston skeemaan.

Metodit

parseQuestionDB

Public QuestionDB parseQuestionDB(in InputStream) throws XMLParserException

Jäsentää XML-tiedoston kysymykset.

Argumentit

in Avattu XML-tiedostovirta.

Palauttaa

QuestionDB-olion, jossa on kaikki XML-puun sisältämät kysymykset.

Poikkeukset

XMLParserException Virhe parsiessa XML-tiedostoa.

parseSettingsDB

Public SettingsDB parseSettingsDB(in InputStream) throws XMLParserException

Jäsentää asetustiedoston.

Argumentit

in Avattu XML-tiedostovirta.

Palauttaa

SettingsDB-olion, jossa on kaikki XML-puun sisältämät asetukset.

Poikkeukset

XMLParserException

parseSecretSettingsDB

Public SecretSettingsDB parseSecretSettingsDB(in InputStream) throws XMLParserException

Jäsentää koekysymysjakaumatiedoston.

Argumentit

in Avattu XML-tiedostovirta.

Palauttaa

SecretSettingsDB-olion, joka sisältää koekysymysjakauman.

Poikkeukset

XMLParserException

generateQuestionDocument

public generateQuestionDocument(QuestionDB q, OutputStream o) throws XMLParserException

Jäsentää QuestionDB-tyyppisen olion sisältämät kysymykset XML-puuksi.

Argumentit

q Pitää sisällään joko kaikki ohjelmiston harjoituskysymykset tai testikysymykset.

o Tiedostovirta, jonne XML-dokumentti tallennetaan.

Palauttaa

Poikkeukset

XMLParserException

generateSettingsDocument

Public void generateSettingsDocument(SettingsDB s, OutputStream o) throws XMLParserException

Generoi asetustietokannasta XML-tiedoston.

Argumentit

s Tietokanta ohjelmiston asetuksista.

o Tiedostovirta, jonne generoitu XML-tiedosto tallennetaan.

Palauttaa

Poikkeukset

XMLParserException

generateSecretSettingsDocument

Public void generateSecretSettingsDocument(SecretSettingsDB s, OutputStream o) throws XMLParserException

Generoi koekysymysjakaumatietokannasta XML-tiedoston.

Argumentit

s Tietokanta ohjelmiston salaisista asetuksista.

o Tiedostovirta, jonne generoitu XML-tiedosto tallennetaan.

Palauttaa

Poikkeukset

XMLParserException

7.6.3 Crypt

Luokka laskee käyttäjän antamasta salasanasta SHA-1-tiivistealgoritmilla salausavaimen ja salaa sekä purkaa testikysymystiedoston.

Metodit

calcChecksum

Public SecretKey calcChecksum(password char[]) throws CryptException

Metodi laskee SHA-1-tiivistealgoritmilla salausavaimen käyttäjän salasanasta.

Argumentit

password Käyttäjän salasana.

Palauttaa

Salausavaimen sisältämän SecretKey-olion.

Poikkeukset

CryptException

encrypt

public OutputStream encrypt(key SecretKey, OutputStream out) throws CryptException

Palauttaa tiedostovirran, jonne kirjoitettu data salataan annetulla avaimella ja AES-algoritmilla.

Argumentit

key Salasanasta SHA-1-tiivistealgoritmilla muodostettu salausavain.

out Tiedostovirta, jonne kirjoitettava data halutaan salata.

Palauttaa

OutputStream-tiedostovirta, jonne kirjoitettava data salataan.

Poikkeukset

CryptException

decrypt

public InputStream decrypt(key SecretKey, InputStream in) throws CryptException

Palauttaa tiedostovirran, josta voi lukea salatun tiedoston salaamattomana.

Argumentit

key Salasanasta SHA-1-tiivistealgoritmilla muodostettu salausavain.

in Tiedostovirta, joka sisältää salatun tiedoston.

Palauttaa

InputStream-tiedostovirran, josta voi lukea salatun tiedoston salaamattomana.

Poikkeukset

CryptException

checkPassword

Public boolean checkPassword(key SecretKey, InputStream in) throws CryptException

Metodi tarkistaa onko annetulla avaimella salattu argumenttina annettu tiedosto.

Tiedosto sisältää joko salatun tai salaamattoman merkkijonon "magicword". Salaamaton merkkijono merkitsee sitä, ettei salasanaa ole asetettu lainkaan.

Argumentit

key Salasanasta SHA-1-tiivistealgoritmilla muodostettu salausavain.

in Salattu tiedostovirta.

Palauttaa

True, jos annettu tiedosto on salattu annetulla avaimella muuten falsen.

Poikkeukset

CryptException

7.6.4 QuestionDB

Luokka vastaa kysymysten hallinnasta

Konstruktorit

QuestionDB()

getQuestions

public Question[] getQuestions(enum category, enum level)

Argumentit

category Osa-alue.

level Vaikeustaso.

Palauttaa

Kaikki annettuun kategorian ja vaikeustason mukaiset kysymykset.

Poikkeukset

getQuestion

Question getQuestion(int index, enum category, enum level)

Palauttaa kysymyksen indeksin, kategorian ja vaikeusasteen mukaan.

Argumentit

index Kysymyksen indeksi tietokannassa.

category Osa-alue.

level Vaikeustaso.

Poikkeukset

getQuestionCount

int getQuestionCount(enum category, enum level)

Palauttaa tietokannan sisältämän kysymyslukumäärän kategorian ja vaikeustason mukaisesti.

Argumentit

category Osa-alue.

level Vaikeustaso.

Poikkeukset

addQuestion

public void addQuestion(Question q) throws InvalidQuestionIDException

Lisää tietokantaan uuden kysymyksen.

Argumentit

q Question-olio, joka sisältää kaikki uuden kysymyksen tiedot.

Palauttaa

Poikkeukset

InvalidQuestionIDException

Kannassa on jo Question-oliossa annetulla id:llä kysymys.

setQuestion

public void setQuestion(Question q) throws InvalidQuestionIDException

Muokkaa tietokannassa olevaa kysymystä

Argumentit

q Question-olio, joka sisältää kaikki muutettavan kysymyksen tiedot.

Palauttaa

Poikkeukset

InvalidQuestionIDException

Kannassa ei ole Question-oliossa annetulla id:llä kysymystä.

removeQuestion

public void removeQuestion(id int) throws InvalidQuestionIDException

Poistaa tietokannasta kysymys.

Argumentit

id Poista tietokannasta tämän perusteella kysymys.

Palauttaa

Poikkeukset

InvalidQuestionIDException

Kannassa ei ole Annetulla id:llä kysymystä.

getUnusedID

public int getUnusedID()

Palauttaa seuraavan vapaan id-tunnuksen.

Argumentit

palauttaa

Poikkeukset

7.6.5 SettingsDB

Luokka säilyttää harjoittelusarjan kysymyslukumäärän ja tulostusasetukset.

Konstruktorit

SettingsDB(int setCount, String theoryPath, String questionPath, int printCopies, String defaultPrinter)

Argumentit

Konstruktori alustaa olion luokan kenttien mukaisesti.

Kentät

setCount Yhden kysymyssarjan kysymysten lukumäärä.

theoryPath Teoriatiedostojen polku.

questionPath Kysymystiedostojen polku.

printCopies Tulostettavien läpipääsytodistusten lukumäärä.

defaultPrinter Oletustulostin.

7.6.6 SecretSettingsDB

Luokka säilyttää testikysymysten jakaumaa.

Konstruktorit

SecretSettingsDB(int decimal, int percent, int fraction, int proportion)

Argumentit

Konstruktori alustaa olion luokan kenttien mukaisesti.

Kentät

decimal Desimaalikysymysten määrä.

percent Prosenttikysymysten määrä.

fraction Murtolukukysymysten määrä.

proportion Verrantokysymysten määrä.

7.6.7 TheoryDB**Konstruktorit**

TheoryDB(String decimal, String percent, String fraction, String proportion)

Luokan kentissä ovat teoriadokumenttien tiedostonimet polkuineen.

Argumentit

Konstruktori alustaa olion luokan kenttien mukaisesti.

Kentät

decimal Desimaaliteoria.

percent Prosenttiteoria.

fraction Murtolukuteoria.

proportion Verrantoteoria.

7.6.8 Question

Luokka pitää sisällään kaikki tiedot yhdestä kysymyksestä.

Kentät

int id Kysymyksen id-numero

enum category Kysymyskategoria

enum level Kysymyksen vaikeusaste

String question Itse kysymys

String unit Yksikkö

String firstHint Ensimmäinen vihje

String secondHint Toinen vihje

Answer answer Kysymyksen vastaus

String formula Laskukaava

7.6.9 Answer

Konstruktorit

Answer(String a)

Tarkistaa vastauksen muodon oikeellisuuden.

Poikkeukset

FormatException Annetun vastauksen muoto on väärin.

Metodit

equals

public boolean equals(Object o)

Tarkistaa onko olion sisältämä vastaus sama kuin argumenttina annettu.