Käyttöliittymäohjelmointi

**Harjoitustyöselostustemplate**

Oulun yliopisto

Tietojenkäsittelytieteiden laitos

Harjoitustyöselostus

Joonas Nygård

19.9.2107

Tiivistelmä

HUOM! Älä jätä palautukseesi template-tekstejä muihin kuin otsikoihin!

Kerro lyhyesti harjoitustyösi aihe ja tarkoitus, sekä mahdollisesti muuta prosessin aikana ilmennyttä.

Sisällys

Tiivistelmä 2

Sisällys 3

1. Vaatimusmäärittely 4

1.1 Toiminnallisuuden määrittely 4

1.2 Käyttäjäryhmien identifiointi 4

1.3 Käytön kontekstin määrittely 4

2. Käyttöliittymäkonseptit 6

3. Käyttöliittymän prototypointi 7

4. Käyttöliittymän evaluointi 8

4.1 Heuristinen evaluointi 8

4.2 Kognitiivinen läpikäynti 8

4.3 Skenaariopohjainen evaluointi 8

4.4 Käyttäjätestaus 9

4.5 Vertaisarvioinnit 9

4.6 Evaluointien löydökset ja vaikutukset 9

5. Viimeistelty suunnitelma käyttöliittymästä 10

6. WWW-puolen vaatimusmäärittely 11

6.1 Toiminnallisuuden määrittely 11

6.2 Käyttäjäryhmien identifiointi 11

6.3 Käytön kontekstin määrittely 11

7. WWW-Käyttöliittymäkonseptit 13

8. WWW-Käyttöliittymän prototypointi 14

9. WWW-Käyttöliittymän evaluointi 15

9.1 Heuristinen evaluointi 15

9.2 Kognitiivinen läpikäynti 15

9.3 Skenaariopohjainen evaluointi 16

9.4 Käyttäjätestaus 16

9.5 Vertaisarvioinnit 16

9.6 Evaluointien löydökset ja vaikutukset 16

10. Viimeistelty suunnitelma WWW-käyttöliittymästä 17

11. Käyttöohje 18

12. Poikkeamat suunnitelmasta 19

13. Harjoitustyön tekijöiden palautustiedot 20

# Vaatimusmäärittely

Tällä järjestelmällä on tarkoitus suunnitella erityisesti urheilijoiden viikkoruokavalio työpöytäympäristössä.

## Toiminnallisuuden määrittely

* Järjestelmän on tarkoitus olla apuna viikkoruokavalion suunnittelussa ja energiansaannin seurannassa.
* Käyttäjä pystyy lisäämään järjestelmään eri ruoka-aineita/100g sekä valmiita ateria templateja, sen mukaan, mistä hänen ateriansa yleensä koostuvat. Ruoka-aine/100g koostuu ainakin nimestä ja kalorimäärästä/100g. Ateria voi koostua usesata ruoka-aineesta ja niiden määrästä grammoina, jolloin voidaan laskea aterian kokonais kalorimäärä. Näin saadaan päivittäinen ja viikon ruokavalio ja energiamäärä selville ja ruokavaliota helppo hioa, jos energiamäärä ei vastaa tavoiteltua.

## Käyttäjäryhmien identifiointi

* Käyttäjäryhmiä on paljon. Lähes joka lajin urheilijat suunnittelevat ravinnonsaantinsa tarkasti. Ammatti-ja harrastelijaurheilijat, laihduttajat, ravintoterapeutit, personaltrainerit ja valmentajat.
* Ammattiurheilijoiden ruokavalio on yleensä hyvin säännönmukainen ja koostuu samoista ruoka-aineista. Käytännössä ammattiurheilijoilla ruokavalio suunnitellaan yhdessä valmentajan kanssa, mikäli urheilijalla on erilline valmentaja. Revintoterapeutti voi suunnitella järjestelmän avulla asiakkaille ruokavalion.
* Urheilijat/valmentajat ovat tärkein käyttäjäryhmä, jonka tarpeisiin järjestelmä suunnitellaan.

### Jaakko – Kehonrakentaja – Ammattiurheilija – Ensisijainen käyttäjä

25 vuotias kehonrakennusta. Takana 5 vuotta saliharrastusta ja 2 vuotta ammattimaista kehonrakennusta. Jaakolla ei ole erillistä valmentajaa, vaan hän suunnittelee itse treenit ja ruokavalionsa. Jaakolle on tärkeää seurata tarkasti energiansaantiaan ennen kisoja. Järjestelmän avulla hän pystyy suunnittelemaan diettinsä ruokavalion etukäteen, sekä lisäämään päiväkirjaan toteutuneet ateriat.

### Sari – Hiihtovalmentaja

Sari 44 on suomen naisten hiihtomaajoukkueen päävalmentaja ja suunnittelee tarkan treeniruokavalion yhdessä jokaisen hiihtäjän kanssa. Järjestelmän avulla on helppo suunnitella raamit koko joukkueelle ja tehdä pieniä muutoksia yksilötasolla eri hiihtäjien välillä.

### Teppo – Veturinkuljettaja – Laihduttaja

Teppo on 55 vuotias veturinkuljettaja, joka on ottanut projektiksi ylimääräisten kilojen karistamisen. Järjestelmän avulla hän pystyy suunnittelemaan diettinsä ja lisäämään toteutuneet ateriat päiväkirjaan. Tepon dieetin ruokavalio ei ole yhtä säännönmukainen kuin ammattiurjeilija.

### Sanna – Ravintoterapeutti

Sanna 30v on ravintoterapeutti, joka suunnittelee asiakkailleen tapauskohtaisesti suositellun ruokavalion. Järjestelmän avulla suunnittelu onnistuu helposti ja hän voi asiakkaan kanssa työstää ruokavaliota yhdessä helposti, niin että ravintoarvot pysyvät rajoissa.

## Käytön kontekstin määrittely

### Fyysinen konteksti

Ohjelman käyttö on tarkoitettu tapahtuvan pääsääntöisesti koti/toimistoympäristössä pöytäkoneella tai kannettavalla. Ohjelman käyttö ei vaadi internetyhteyttä. Tulostusmahdollisuus olisi hyvä viikkoruoka-aikataulun tulostusta varten.

### Sosiaalinen konteksti

Ohjelman käyttö tapahtuu pääsääntöisesti yksin ja jossain vaiheessa esimerkiksi valmentaja ja urheilija yhdessä. Tilanteet jossa yhtäaikaisia suunnittelijoita olisi enemmän kuin 2 ovat harvinaisia.

### Organisatorinen konteksti

### Toiminnallinen konteksti

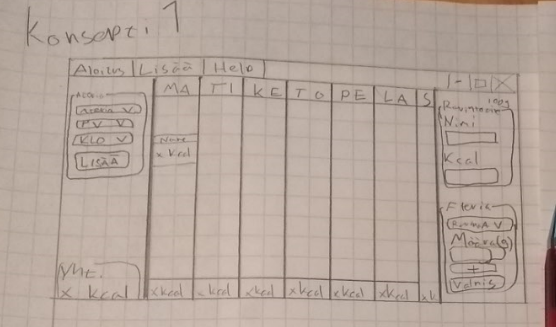
### Tekninen konteksti

Ohjelman käyttö tapahtuu windows ympäristössä pöytä/kannettavalla tietokoneella. Käytön kannalta on oleellista että käyttäjällä on tulostin käytettävissä, jotta ruoka-aikataulun voi tulostaa.

# Käyttöliittymäkonseptit

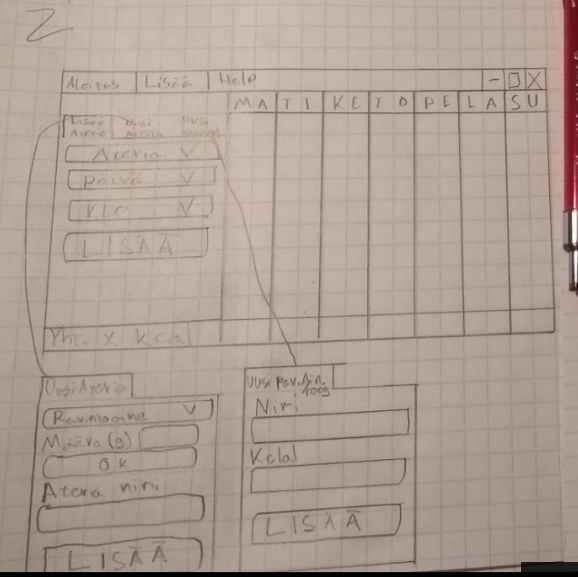
### Konsepti #1

Ensimmäisessä konseptissa kaikki elementit ovat yhtäaikaa näkyvillä, joka tekee konseptista ehkä hieman sekavan.



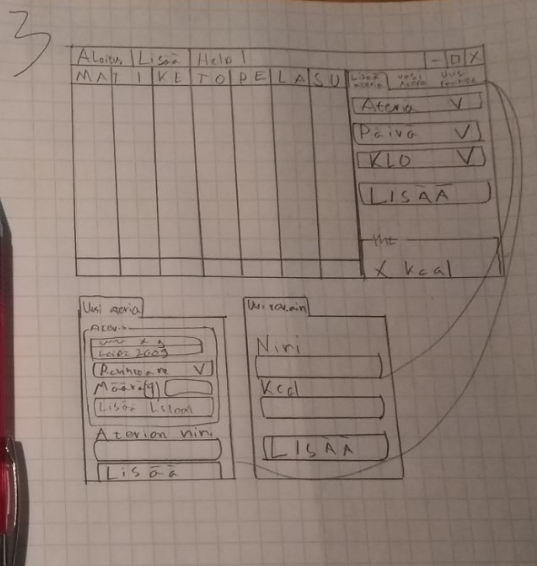
### Konsepti #2

Toisessa konseptissa on käytetty välilehtiä selkeyttämään käyttöliittymää ja vähentämään kerralla esitettävien elementtien määrää.

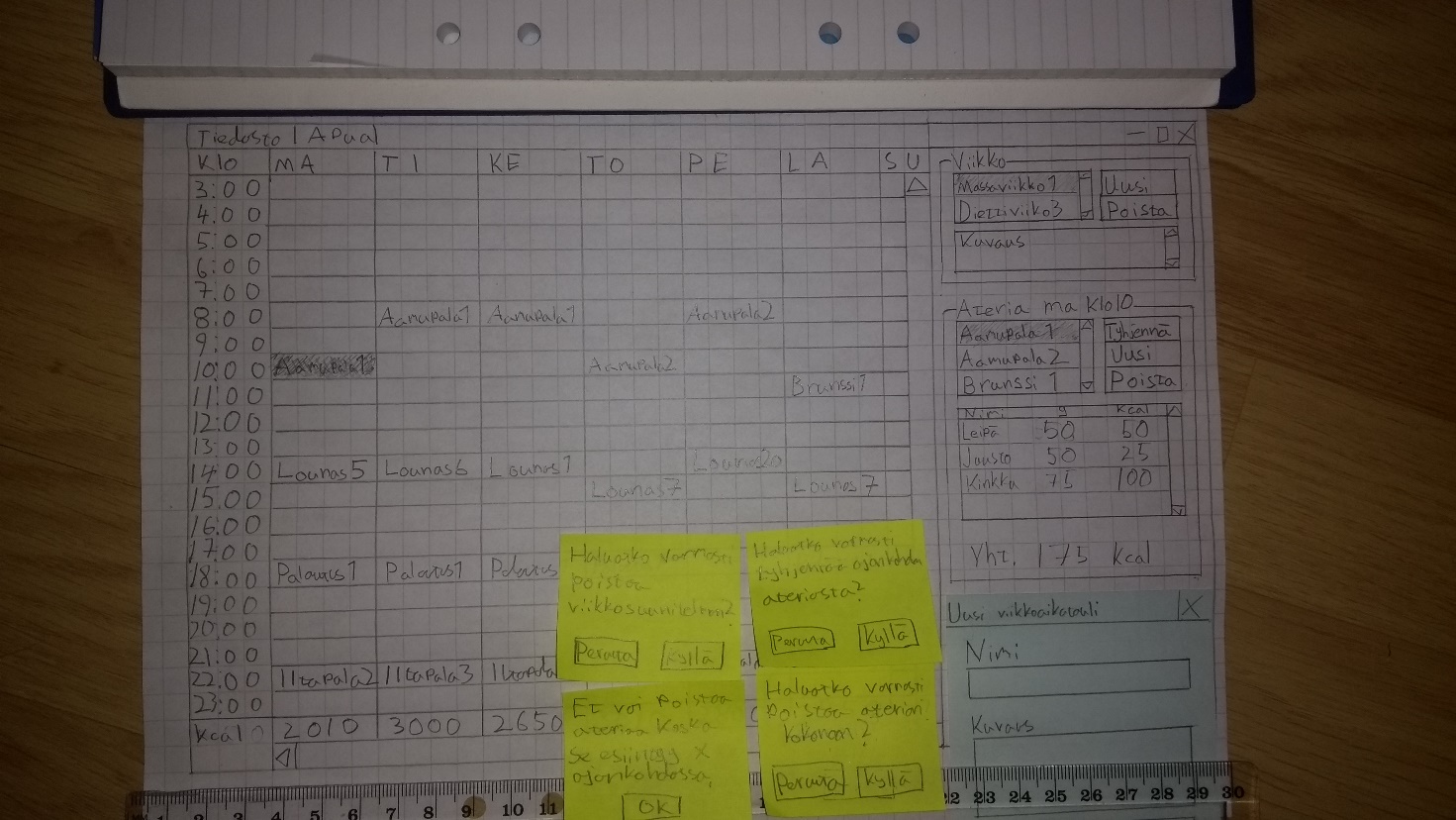


### Konsepti #3 VALITTU

Kolmas konsepti ei eroa paljoa toisesta, mutta viikkonäkymä on siirtynyt vasemmalle puolelle. Tämä konsepti miellyttää eniten omaa silmääni ja vaikuttaa luonnollisimmalta. Viikkonäkymä johon käyttäjän lisäämät ateriat esitetään on parhaalla paikalla vasemmalla ja käyttäjä voi lisätä uusia aterioita näkymään oikeanpuoleisesta näkymästä.



# Käyttöliittymän prototypointi



Päänäkymä. Koostuu taulukosta joka kuvastaa viikkonäkymää. Käyttäjä voi luoda uusia viikkonäkymiä “Uusi”-napista oikealta yläreunasta ->(oik. alareunassa dialogi-ikkuna) ja valita listasta tarkasteltavan viikon, jolloin taulukkonäkymä muuttuu kuvaamaan kyseistä viikkoa.

Käyttäjä voi muokata viikkonäkymää valitsemalla taulukosta tuntia kuvaavan solun, johon voi liitttää Aterian oikealla aterialistasta, jolloin Aterian nimi ilmestyy soluun ja myöhemin solua painamalla oiealla näytetään aina tähän liitetty ateria ja sen sisältö. Käyttäjä voi myös tyhjentää solun, luoda uuden ateruia, jos listasta ei löydy oikeaa ateriaa tai poistaa Aterian kokonaan. Aterian poistaminen onnistuu vain, jos sitä ei ole liitetty mihinkään muuhun soluun, missään viikonäkymässä. Poistamiset ja tyhjentämiset varmistetaan aina dialogi-ikkunalla.

HUOM! Kuvasta puuttuu ylävalikosta aukeavat alavalikot, mutta viikko ruokavalion tulostaminen ja ohjelman lopetus tapahtuu tiedosto valikosta ja apua valikosta vaihoehdot käyttöohjeet ja tietoja ohjelmasta, joista avautuisi uuteen ikkunaan yksinkertaiset käyttöohjeet ja tietoja ohjelmasta.

# C:\Users\QH900\AppData\Local\Microsoft\Windows\INetCache\Content.Word\IMG_20171012_211150323.jpg

Elintarvikkeiden lisäämiseen ja aterioiden luontiin tarkoitettu ikkuna. Elintarvike välilehdellä käytttäjä voi lisätä uusia elintarvikkeita ohjelmaan. Elintarvike sisältää sen nimen ja energiamäärän kilokaloreissa. Käyttäjä voi poistaa elintavikkeita ainoastaan, mikäli se ei jo sisälly johonkin ateriakokonaisuuteen. Jos käyttäjä muokkaa elintarvikettta, se vaikuttaa kaikkiin aterioihin, jotka sisältävät kys. elintarvikkeen, ja siitä ilmoitetaan käyttäjälle dialogi-ikkunalla.

Aterian luonti tapahtuu lisäämällä elintarvikkeita, valitsemalla haluttu elintarvike listasta ja niiden määrä grammoina tai millilitroina ja painamalla Lisää painiketta. Aterian kokonais kalorimäärä päivittyy sitämukaa (vasen alareuna), kun elintarvikkeita lisätään ateriaan. Jos syöte on virheellinen, ilmoitetaan käyttäjälle dialogi-ikkunassa. Mikäli aterialle annetaan nimi joka on jo käytössä, ilmoitetaan käyttäjälle. Mikäli käyttäjä syöttää cirheellisesti elintarvikkeen listaan, voi hän poistaa sen valitsemalla listasta elintarvikkeen ja poistamalla sen “Poista” painikkeella.

Kun ateria on valmis käyttäjä painaa “Tallenna”-painiketa, jolloin ateria tallennetaan ja ikkuna sulkeutuu.

# Käyttöliittymän evaluointi

* Tarkoituksena löytää ongelmia prototyypistä mahdollisimman paljon:
  + Hyvä evaluointi löytää ongelmia
  + Huono evaluointi puolustelee virheettömyyttä
* Käytettävä kahta evaluointimenetelmää, joista toisen tulee olla heuristinen evaluointi, ja toisen joko kognitiivinen läpikäynti, skenaariopohjainen evaluointi tai käyttäjätestaus.
* Lisäksi kuvattava vertaisarviointien tuloksia, jos niitä on saanut.
* löydettiinkö evaluoinneissa ongelmia?
* mitä johtopäätöksiä voidaan tehdä käyttöliittymän laadusta suhteessa vaatimusmäärittelyyn?
* miten evaluointien tulokset vaikuttavat prototyyppiin?
* Mieti mitä prototyypissä pitäisi tehdä toisin? Muista että evaluointi perustuu aina kappaleen 1 määrittelyihin ja tehdään niitä vastaan.

## Heuristinen evaluointi

* Kerro mitä heuristiikkakokoelmaa käytit, mutta keskity kertomaan heuristiikan avulla tehdyistä löydöksistä ennemmin kuin toistamaan heuristiikkakokoelman kuvaus.
* Löydä mahdollisimman paljon ongelmakohtia, älä pidä “puolustuspuheenvuoroa virheettömyydestä” (tähän mennessä ei vielä virheetöntä prototyyppiä ole tullut vastaan).

## Kognitiivinen läpikäynti

* Vaihtoehtoinen kohdan 4.3 ja 4.4 kanssa
* Käyttäjän tehtävä/tavoite jaetaan pienempiin, järjestyksessä suoritettaviin osiin
* Osatehtävät käydään läpi ja arvioidaan (Wharton 1994)

## Skenaariopohjainen evaluointi

* Vaihtoehtoinen kohdan 4.2 ja 4.4 kanssa
* Käydään läpi käyttäjälle tyypillisiä tehtäviä ja arvioidaan miten hyvin tietyt käyttäjät (vrt. luvut 1.2 ja 1.3) onnistuvat niissä
* Tehtävistä luodaan useita skenaarioita, joissa jokaisessa on kaksi osaa: tehtäväskenaario ja käyttöskenaario
* Ei vain kuvata skenaarioita, vaan evaluoidaan prototyyppiä niiden avulla, ts. skenaarioista pitäisi myös saada analysoitua löydöksiä ongelmista, tämäkään ei ole “puolustuksen puheenvuoro ongelmattoman käyttöliittymän puolesta”.

## Käyttäjätestaus

* Vaihtoehtoinen kohdan 4.2 ja 4.3 kanssa
* Formaali käytettävyystestaus esimerkkikäyttäjillä on raskas tätä tarkoitusta varten, mutta pienimuotoisempi “kysytään kaverilta mielipiteitä”-testi voi olla hyvänä lisänä antamassa näkökulmaa ja etäisyyttä omiin kehittäjälähtöisiin olettamuksiin.
* Parhaimmillaan jos onnistuu saamaan testaukseen oikean esimerkkikäyttäjän pääkäyttäjäryhmän edustajana. Huomaa, että itse työn tekijänä ei voi olla käyttäjätestaajana.

## Vertaisarvioinnit

* Valinnainen kohta.
* Mikäli olet saanut vertaisarvioita muilta kurssilaisilta (esim. Moodlen vertaisarviointialueella), niin niitä voi kuvata ja analysoida niiden vaikutuksia prototyyppiin.
* Toimii lisänä omiin evaluointeihin, ja tuo näkökulmaa muiltakin kuin ohjelman kehittäjältä. Näitä tuloksia voi myös tuoda esille osana muita evaluointeja.

## Evaluointien löydökset ja vaikutukset

* Yhteenveto kohdista 4.1 – 4.5
* LÖYDÖSTEN VAIKUTUKSET ANALYSOITAVA!
  + Mitä ongelmia oli?
  + Miksi ne vaikuttavat käyttöä hankaloittavasti?
  + Miten suunnitelmaa/prototyyppiä tulisi muuttaa, jotta käytettävyys paranisi?

# Viimeistelty suunnitelma käyttöliittymästä

* Kuten prototyyppi luvussa 3, joko piirrettynä kynä-viivoitin-ruutupaperi -menetelmällä tai kuvankäsittelyohjelmalla piirtäen.
* EI RUUDUNKAAPPAUKSIA!
  + Tässä vaiheessa prosessia ei pitäisi olla vielä toteutettua koodia, joten ruudunkaappausten käyttö kertoo prosessin menneen väärässä järjestyksessä. (Harjoitustyövaatimusten kohta: “GUI on suunniteltava ja suunnitelma dokumentoitava ennen käyttöliittymän toteutusta.”)
* Käyttöliittymä kuvataan siinä laajuudessaan kuin se toteutetaan
  + Edelleen mukana kaikki dialogit, infoikkunat, jne.
  + Tässä vaiheessa voi siis rajata toteutusta vielä pienemmäksi, jos evaluoinnissa on tullut ilmi että työstä tulisi muutoin liian iso.

# WWW-puolen vaatimusmäärittely

Tämän kohdan voi halutessaan esittää myös kohdan 1 yhteydessä, jos stand-alone- ja web-puolen ohjelmat liittyvät toisiinsa.

## Toiminnallisuuden määrittely

* mitä järjestelmällä on tarkoitus tehdä?
* mitä toimintoja järjestelmältä vaaditaan?

## Käyttäjäryhmien identifiointi

* millaisia eri käyttäjäryhmiä järjestelmälle voidaan löytää?
* miten eri käyttäjäryhmät eroavat toisistaan?
* mikä on tärkein käyttäjäryhmä (kenelle käyttöliittymä on tarkoitettu)?
* Esimerkkikäyttäjien kuvauksia (mitä enemmän asiaa/käyttäjä, sitä parempi)

### Esimerkkikäyttäjän #1 kuvaus

### Esimerkkikäyttäjän #2 kuvaus

### Esimerkkikäyttäjän #3 kuvaus

…

1.2.N Esimerkkikäyttäjän #N kuvaus

## Käytön kontekstin määrittely

* Määrittelykuvauksia siitä, missä ympäristössä järjestelmää käytetään.
* Huomioitava kaikki kontekstit sillä tavoin kuin ne kyseiseen järjestelmään vaikuttavat.

### Fyysinen konteksti

### Sosiaalinen konteksti

### Organisatorinen konteksti

### Toiminnallinen konteksti

### Tekninen konteksti

# WWW-Käyttöliittymäkonseptit

Konsepteja web-käyttöliittymän rakenteesta karkealla tasolla, mielellään kynä/paperipiirroksia joko skannattuina tai digikuvina. Ei ruudunkaappauksia tai mielellään edes koneella piirrettyjä. Aikaa konseptin tuottamiseen käytetty vain muutama minuutti (ei kiinnittymistä tässä vaiheessa vielä oman työn jälkeen, vaan heikommat konsepti-ideat on oltava mahdollista heittää pois ilman että niissä menee paljoa työtä ja siten sitoutumista mukana).

* Huomaa, että www-ympäristö asettaa omia rajoituksiaan ja mahdollisuuksia käyttöliittymälle.
* Keskitytään korkean tason perusideoihin
* Kuvataan vain perusnäkymä/-näkymät, ei vielä kaikkia näkymiä ohjelmistosta.
* arvioidaan konseptien hyviä ja huonoja puolia, valitaan perustellusti yksi prototypoitavaksi
* miten päädyttiin valittuun käyttöliittymäkonseptiin?
* tekstinä ja kuvina
* Useampia, lähestymistavaltaan oleellisesti toisistaan poikkeavia konsepteja vähintään 2-4 kpl

### Konsepti #1

### Konsepti #2

### Konsepti #3

…

2.1.N Konsepti #N

# WWW-Käyttöliittymän prototypointi

* prototypoitavan web-käyttöliittymän suunnitelman hiominen karkealta tasolta
* Elementtien sijoittelussa noudatetaan pohjaverkkoa/gridiä.
* Joko kynä-viivoitin-ruutupaperimenetelmällä skannattuna kuten edellä, tai piirto-ohjelmalla puhtaaksipiirrettyinä.
* EI RUUDUNKAAPPAUKSIA! (Ei ole tarkoitus tehdä koodiprototyyppiä, vaan paperiprototyyppi.)
* Kaikki näkymät ohjelmasta suunniteltava, ml. Dialogit, ilmoitukset, jne.
* Mahdollistettava evaluointi prototyypin pohjalta sillä tarkkuudella, että käytettävyysongelmat on mahdollista löytää.

# WWW-Käyttöliittymän evaluointi

* Tarkoituksena löytää ongelmia web-käyttöliittymän prototyypistä mahdollisimman paljon:
  + Hyvä evaluointi löytää ongelmia
  + Huono evaluointi puolustelee virheettömyyttä
* Käytettävä kahta evaluointimenetelmää, joista toisen tulee olla heuristinen evaluointi, ja toisen joko kognitiivinen läpikäynti, skenaariopohjainen evaluointi tai käyttäjätestaus.
* Lisäksi kuvattava vertaisarviointien tuloksia, jos niitä on saanut.
* löydettiinkö evaluoinneissa ongelmia?
* mitä johtopäätöksiä voidaan tehdä web-käyttöliittymän laadusta suhteessa vaatimusmäärittelyyn?
* miten evaluointien tulokset vaikuttavat prototyyppiin?
* Mieti mitä prototyypissä pitäisi tehdä toisin? Muista että evaluointi perustuu aina kappaleen 1 määrittelyihin ja tehdään niitä vastaan.

## Heuristinen evaluointi

* Kerro mitä heuristiikkakokoelmaa käytit, mutta keskity kertomaan heuristiikan avulla tehdyistä löydöksistä ennemmin kuin toistamaan heuristiikkakokoelman kuvaus.
* **Huomaa, että web-puolelle on omia heuristiikkojaan, joita voi myös käyttää.**
* Löydä mahdollisimman paljon ongelmakohtia, älä pidä “puolustuspuheenvuoroa virheettömyydestä” (tähän mennessä ei vielä virheetöntä prototyyppiä ole tullut vastaan).

## Kognitiivinen läpikäynti

* Vaihtoehtoinen kohdan 9.3 ja 9.4 kanssa
* Käyttäjän tehtävä/tavoite jaetaan pienempiin, järjestyksessä suoritettaviin osiin
* Osatehtävät käydään läpi ja arvioidaan (Wharton 1994)

## Skenaariopohjainen evaluointi

* Vaihtoehtoinen kohdan 9.2 ja 9.4 kanssa
* Käydään läpi käyttäjälle tyypillisiä tehtäviä ja arvioidaan miten hyvin tietyt käyttäjät (vrt. luvut 6.2 ja 6.3 tai 1.2 ja 1.3 ) onnistuvat niissä
* Tehtävistä luodaan useita skenaarioita, joissa jokaisessa on kaksi osaa: tehtäväskenaario ja käyttöskenaario
* Ei vain kuvata skenaarioita, vaan evaluoidaan prototyyppiä niiden avulla, ts. skenaarioista pitäisi myös saada analysoitua löydöksiä ongelmista, tämäkään ei ole “puolustuksen puheenvuoro ongelmattoman käyttöliittymän puolesta”.

## Käyttäjätestaus

* Vaihtoehtoinen kohdan 9.2 ja 9.3 kanssa
* Formaali käytettävyystestaus esimerkkikäyttäjillä on raskas tätä tarkoitusta varten, mutta pienimuotoisempi “kysytään kaverilta mielipiteitä”-testi voi olla hyvänä lisänä antamassa näkökulmaa ja etäisyyttä omiin kehittäjälähtöisiin olettamuksiin.
* Parhaimmillaan jos onnistuu saamaan testaukseen oikean esimerkkikäyttäjän pääkäyttäjäryhmän edustajana. Huomaa, että itse työn tekijänä ei voi olla käyttäjätestaajana.

## Vertaisarvioinnit

* Valinnainen kohta.
* Mikäli olet saanut vertaisarvioita muilta kurssilaisilta (esim. Moodlen vertaisarviointialueella), niin niitä voi kuvata ja analysoida niiden vaikutuksia prototyyppiin.
* Toimii lisänä omiin evaluointeihin, ja tuo näkökulmaa muiltakin kuin ohjelman kehittäjältä. Näitä tuloksia voi myös tuoda esille osana muita evaluointeja.

## Evaluointien löydökset ja vaikutukset

* Yhteenveto kohdista 9.1 – 9.5
* LÖYDÖSTEN VAIKUTUKSET ANALYSOITAVA!
  + Mitä ongelmia oli?
  + Miksi ne vaikuttavat käyttöä hankaloittavasti?
  + Miten suunnitelmaa/prototyyppiä tulisi muuttaa, jotta käytettävyys paranisi?

# Viimeistelty suunnitelma WWW-käyttöliittymästä

* Kuten web-käyttöliittymän prototyyppi luvussa 8, joko piirrettynä kynä-viivoitin-ruutupaperi -menetelmällä tai kuvankäsittelyohjelmalla piirtäen.
* EI RUUDUNKAAPPAUKSIA!
  + Tässä vaiheessa prosessia ei pitäisi olla vielä toteutettua koodia, joten ruudunkaappausten käyttö kertoo prosessin menneen väärässä järjestyksessä. (Harjoitustyövaatimusten kohta: “GUI on suunniteltava ja suunnitelma dokumentoitava ennen käyttöliittymän toteutusta.”)
* Käyttöliittymä kuvataan siinä laajuudessaan kuin se toteutetaan
  + Edelleen mukana kaikki dialogit, infoikkunat, jne.
  + Tässä vaiheessa voi siis rajata toteutusta vielä pienemmäksi, jos evaluoinnissa on tullut ilmi että työstä tulisi muutoin liian iso.

# Käyttöohje

Valmiin työn (lyhyt) käyttöohje esimerkkikäyttäjien näkökulman huomioiden.

Tarkastusta helpottamaan on lisäksi kerrottava ainakin seuraavaa:

* Stand-alone -puolelta
  + Mistä luokasta/ millä komennolla komentoriviltä ohjelma käynnistetään.
  + Tarvittaessa käyttäjätunnus/salasana tarkastusta varten.
  + Mikäli tarvitaan tietokantaa, niin mistä se löytyy ja mitä toimenpiteitä (esim. tunnelointi) pitää tehdä ennen ohjelman käynnistämistä.
* Web-puolelta:
  + Mistä osoitteesta toimiva sovellus löytyy? Testausta varten pelkkä paketin toimittaminen ei riitä, vaan ohjelma pitää olla asennettuna jollekin palvelimelle, josta sitä voi www-selaimella ajaa mistä päin internetiä tahansa.
  + Mikäli ohjelma käyttää tietokantaa, mistä se löytyy ja miten siihen pääsee tutustumaan.
  + Tarvittaessa testikäyttäjän käyttäjätunnus / salasana ohjelman testausta varten.

# Poikkeamat suunnitelmasta

Mikäli aiemmista suunnitelmista on poikettu toteutusvaiheessa, on jokainen poikkeama eriteltävä. Jokaisesta poikkeamasta on esitettävä seuraavat asiat:

* Poikkeaman kohta
* Poikkeamisen syyt ja perustelut (perusteluna “taidot ei riittäneet” tai “aika ei riittänyt” ei ole validi, vaan jokaisen poikkeaman tulee parantaa käyttöliittymän laatua ja käytettävyyttä)
* Poikkeaman vaikutukset käyttöliittymään ja käytettävyyteen

# Harjoitustyön tekijöiden palautustiedot

Tekijän/tekijöiden tiedot:

|  |  |
| --- | --- |
| Nimi |  |
| Hetu / op.numero |  |
| Tiedekunta ja laitos  (Aloitusvuosi) |  |
| Sähköpostiosoite |  |