

# Määrittelydokumentti

Aija

Helsinki 22.6.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Työn aihepiiri . . . . .	1
1.2	Määritelmät ja lyhenteet . . . . .	1
<b>2</b>	<b>Versio historia</b>	<b>2</b>
<b>3</b>	<b>Yleiskuvaus</b>	<b>3</b>
3.1	Arkkitehtuurin yleiskuvaus . . . . .	3
3.2	Yhteyskaavio . . . . .	4
<b>4</b>	<b>Käyttäjävaatimukset</b>	<b>4</b>
4.1	Sisäänkirjautuminen . . . . .	4
4.1.1	Sisäänkirjautuminen tunnistautumalla . . . . .	4
4.1.2	Sisäänkirjautuminen linkillä . . . . .	5
4.1.3	Epäonnistunut sisäänkirjautuminen linkillä . . . . .	5
4.1.4	Uusi käyttäjä . . . . .	5
4.1.5	Rekisteröityminen . . . . .	5
4.1.6	Salasanan lähetys . . . . .	5
4.1.7	Sähköpostiosoite käyttäjätunnuksena . . . . .	6
4.1.8	Järjestäjälle sallitut sähköpostiosoitteet . . . . .	6
4.1.9	Sähköpostiosoitteen muoto . . . . .	6
4.1.10	Sähköpostiosoitteen pakollisuus . . . . .	6
4.1.11	Salasana . . . . .	6
4.1.12	Salasanan vaihtaminen . . . . .	6
4.1.13	Sisäänkirjautumislinkki . . . . .	6
4.1.14	Epäonnistunut tunnistauminen . . . . .	7
4.2	Monikielisyyden tuki . . . . .	7
4.2.1	Tuetut kielet . . . . .	7
4.2.2	Kielen valinta . . . . .	7
4.2.3	Kielien ylläpito . . . . .	7
4.3	Ryhmän ja aikataulun luonti . . . . .	7
4.3.1	Oikeus ryhmän ja aikataulun luontiin . . . . .	7
4.3.2	Järjestäjän aloitusnäyttö . . . . .	7

4.3.3	Uuden ryhmän luonti . . . . .	8
4.3.4	Ryhmän perustiedot . . . . .	8
4.3.5	Aikataulun perustiedot . . . . .	8
4.3.6	Tietojen tallennus . . . . .	8
4.3.7	Tuntijako . . . . .	8
4.3.8	Tarjottavien aikojen syöttö . . . . .	8
4.3.9	Aikataulun pituuden ilmoittaminen . . . . .	9
4.3.10	Aikataulun tallentaminen . . . . .	9
4.3.11	CSV-tiedoston ohjeistus . . . . .	9
4.3.12	Osanottajien syöttö leikepöydältä . . . . .	9
4.3.13	Tunnistettavat sarakkeet . . . . .	9
4.3.14	CSV-tiedoston jäsenys . . . . .	9
4.3.15	Osallistujan poistaminen ryhmästä . . . . .	10
4.3.16	Yhden osallistujan lisääminen . . . . .	10
4.3.17	Osallistujan tietojen muuttaminen . . . . .	10
4.3.18	Osallistujien määrä . . . . .	10
4.4	Kutsut . . . . .	10
4.4.1	Kutsujen lähetys osanottajille . . . . .	10
4.4.2	Kutsun lähetys yhdelle osallistujalle . . . . .	10
4.4.3	Kutsutekstin muotoilu . . . . .	10
4.4.4	Muotoillun kutsutekstin tallennus . . . . .	11
4.4.5	Muuttuvat tiedot kutsussa . . . . .	11
4.4.6	Elementtien lisääminen . . . . .	11
4.4.7	Aikataulun poistaminen . . . . .	11
4.4.8	Ryhmän poistaminen . . . . .	11
4.5	Osallistujan vastaus kutsuun . . . . .	11
4.5.1	Sisäänkirjautuminen linkillä . . . . .	11
4.5.2	Sisäänkirjautuminen tunnistautumalla . . . . .	11
4.5.3	Aikataulun valinta linkillä . . . . .	12
4.5.4	Keskeneräinen aikataulu . . . . .	12
4.5.5	Aikataulun valinta edellisen istunnon perusteella . . . . .	12
4.5.6	Aikataulun valinta listalta . . . . .	12
4.5.7	Sopivien aikojen merkintä . . . . .	12

4.5.8	Sopivan ajan muuttaminen . . . . .	12
4.5.9	Järjestäjä merkitsee osallistujan sopivat ajat . . . . .	12
4.5.10	Sopivat ajat . . . . .	13
4.6	Tapaamisaikojen järjestely . . . . .	13
4.6.1	Ilmoitus sopivuuksien syöttämisestä . . . . .	13
4.6.2	Järjestelynäyttö . . . . .	13
4.6.3	Tapaamisajan sopivuuden esitys . . . . .	13
4.6.4	Ristiriitaisten varausten esitys . . . . .	13
4.6.5	Varatun tapaamisajan esitys . . . . .	14
4.6.6	Tapaamisaikojen varaaminen algoritmilla . . . . .	14
4.6.7	Varattujen tapaamisaikojen esitys . . . . .	14
4.6.8	Algoritmin toiminta . . . . .	14
4.6.9	Varausten uudelleenjärjestely . . . . .	14
4.6.10	Varausten uudelleenlajittelu . . . . .	14
4.6.11	Varauksen vaihtaminen . . . . .	14
4.6.12	Aikataulun lukitseminen . . . . .	15
4.6.13	Ilmoitus lukinnasta . . . . .	15
4.6.14	Aikataulun vapauttaminen . . . . .	15
4.6.15	Valmistuneesta aikataulusta tiedottaminen . . . . .	15
4.6.16	Ilmoituksen lähetys . . . . .	15
4.7	Aikataulunäyttö . . . . .	15
4.7.1	Tilanelaskurit . . . . .	15
4.7.2	Kalenteri . . . . .	16
4.7.3	Viikkonäyttö . . . . .	16
<b>5</b>	<b>Ympäristövaatimukset</b>	<b>16</b>
5.1	Yleiskäyttöinen tietokantarajapinta . . . . .	16
5.2	Laiteympäristö . . . . .	16
5.3	Sähköpostin lähetys . . . . .	16
5.4	CSV-tiedostojen jäsenitys . . . . .	16
5.5	Käyttöliittymän toteutus . . . . .	17
5.6	Erityiskäyttäjryhmät . . . . .	17
5.7	Samanaikaisuuden hallinta . . . . .	17
5.8	Monikielisyyden tuki . . . . .	17

5.9	PHP-skriptien suoritus . . . . .	17
<b>6</b>	<b>Toiminnalliset järjestelmävaatimukset</b>	<b>17</b>
6.1	Taustatoiminnot . . . . .	17
6.1.1	Vanhentuneiden tietojen poisto . . . . .	17
6.1.2	Vanhentunut aikataulu . . . . .	17
6.1.3	Vanhentunut käyttäjä . . . . .	18
6.1.4	Vanhentunut ryhmä . . . . .	18
6.2	Virheilmoitukset . . . . .	18
6.3	Algoritmin suoritus . . . . .	18
<b>7</b>	<b>Ei-toiminnalliset järjestelmävaatimukset</b>	<b>18</b>
7.1	Tuetut selaimet . . . . .	18
7.2	Tietoturva . . . . .	18
7.3	Aikataulun poistaminen . . . . .	19
7.4	Käsinsyötettyjen osallistujien poistaminen . . . . .	19
7.5	Ryhmän poistaminen . . . . .	19
7.6	Käyttäjät . . . . .	19
7.7	Henkilökohtaisten aikojenjakoalgoritmi . . . . .	19
7.7.1	Käsitteitä . . . . .	20
7.7.2	Aikojenjako-algoritmin toiminta . . . . .	20
<b>8</b>	<b>Käyttötapaukset</b>	<b>21</b>
8.1	Järjestäjä . . . . .	21
8.1.1	Onnistunut rekisteröityminen . . . . .	21
8.1.2	Epäonnistunut rekisteröityminen . . . . .	22
8.1.3	Järjestäjä ei nimeä ryhmää . . . . .	22
8.1.4	Järjestäjä antaa kahdelle ryhmälleen saman nimen . . . . .	22
8.1.5	Laskariryhmän ohjaisaikojen tarjoaminen . . . . .	23
8.1.6	Harjoitusryhmän aikataulun muodostaminen . . . . .	23
8.1.7	Harjoitusryhmän tapaamisen peruuttaminen . . . . .	24
8.2	Käyttäjä . . . . .	24
8.2.1	Järjestelmään tutustuminen . . . . .	24
8.2.2	Kielen valinta . . . . .	24

8.2.3	Salasanan antaminen . . . . .	25
8.2.4	Salasanan unohtaminen . . . . .	25
8.2.5	Unohtuneen salasanan lähettäminen . . . . .	25
8.2.6	Salasanan vaihtaminen . . . . .	26
8.2.7	Epäonnistunut sisäänkirjautuminen linkillä . . . . .	26
8.2.8	Osallistujien lisääminen Kurki-kurssinkirjanpitojärjestelmästä	26
8.2.9	Osallistujien lisääminen manuaalisesti . . . . .	27
8.3	Osallistuja . . . . .	27
8.3.1	Omien sopivuuksien ilmoittaminen . . . . .	27
8.3.2	Tapaamisaikojen vastaanottaminen . . . . .	28
<b>9</b>	<b>Toteutettava toiminnallisuus</b>	<b>28</b>
9.1	Ensimmäisen tason vaatimukset . . . . .	28
9.2	Toisen tason vaatimukset . . . . .	29
9.3	Lisäominaisuudet . . . . .	29
9.4	Erikois- ja virhetilanteet . . . . .	30
<b>10</b>	<b>Muut ominaisuudet</b>	<b>30</b>
10.1	Suorituskyky . . . . .	30
10.2	Käytettävyys . . . . .	31
10.3	Ylläpidettävyys . . . . .	31
10.4	Virheistä toipuminen . . . . .	31
<b>11</b>	<b>Testaus</b>	<b>31</b>
11.1	Yksikkötestaus . . . . .	32
11.2	Integroititestausta . . . . .	32
11.3	Validointitestausta . . . . .	32
11.4	Järjestelmätestaus . . . . .	32

# 1 Johdanto

Aija ajanjakojärjestelmä tarjoaa järjestäjälle palveluita sekä omien ryhmien hallintaan että aikataulujen järjestämiseksi ryhmän osallistujille. Järjestelmän palveluilla osallistujat ilmoittavat itselleen sopivat ajat järjestäjän tarjoamista ajoista. Aijan palveluihin kuuluvat myös viestityspalvelut, kuten kutsujen ja tiedotteiden lähetys osallistujille.

## 1.1 Työn aihepiiri

Projektin tavoitteena on valmistaa ohjelmisto, jolla esimerkiksi harjoitusryhmän ohjaaja laatii tapaamisaikataulun opiskelijoille. Tyypillisesti laitoksen harjoitustyöryhmiin voi ilmoittautua verkossa. Ohjaaja tapaa kaikki opiskelijat henkilökohtaisissa tapaamisissa kerran viikossa, useana peräkkäisenä viikkona, aina samaan aikaan viikosta. Opiskelijoita ryhmässä on tyypillisesti n. 10 ja tapaamiset kestävät n. 20 minuuttia. Kaikille sopivan tapaamisajan löytäminen vie kaikkien yhteistä aikaa, eikä ole tilanteena tasapuolinen, vaan suosii äänekkäitä ja rohkeita.

Henkilökohtaisten tapaamisten lisäksi ohjaaja tarjoaa ryhmätapaamiselle sopivia aikoja. Ryhmätapaamisen kesto on esimerkiksi 2 tuntia. Tapaamisia ei välttämättä ole joka viikko, mutta ne ovat kuitenkin aina samaan aikaan yhden kurssin aikana.

Muita henkilökohtaisesti sopivien aikojen löytymistä vaativia tapaamisia voisivat olla esimerkiksi esimiehen järjestämät kehityskeskustelut vuosittain.

## 1.2 Määritelmät ja lyhenteet

**Aija** Tapaamisaikataulujen laatimisjärjestelmä

**Aikataulu** Viikonäyttö, johon järjestäjä merkitsee tarjottavat ajat ja osallistuja itselleen sopivat ajat

**ANSI-SQL/92** Eräs SQL-standariversio

**CGI** Common Gateway Interface, eräs ohjelmatyyppi

**CSV** Comma separated values

**DTD** Document Type Definition, dokumentin tyyppimäärittely

**GUID** Globally unique identifier

**HTML** HyperText Markup Language, eräs sivunkuvauskieli

**ISO International Standard** kansainvälisen standardisoimisjärjestön ISO:n julkaisema standardi

**Järjestäjä** Ryhmän muodostaja, esimerkiksi ohjaaja, esimies, tuutori

**Käyttäjä** Henkilö, joka on kirjautumassa sisään

**Laskariryhmä** Laskuharjoitusryhmä

**Linux** Eräs tietokoneen käyttöjärjestelmä

**Ohtuprojekti** ohjelmistotuotantoprojekti

**Osallistuja** Henkilö, joka kutsutaan ryhmään ja osallistuu tapaamisiin, esimerkiksi opiskelija

**Osallistujaryhmä** Joukko osallistujia, jotka kutsutaan johonkin tapaamiseen

**PHP** PHP: Hypertext Preprocessor, eräs HTML-koodiin upotettu skriptikieli

**PostgreSQL** relaatiotietokantapalvelin

**RFC** Request for Comments

**RGB** red green blue, värikoodaus

**Ryhmä** Kokoaa yhteen osallistujia ja aikatauluja

**SMTP** Simple Mail Transfer Protocol, eräs sähköpostin välityspankollla

**Tapaaminen** Määritelty ajanjakso henkilökohtaiselle tai ryhmätapaamiselle

**TKTL** Helsingin yliopiston tietojenkäsittelytieteen laitos

**Windows** Eräs tietokoneen käyttöjärjestelmä

**XML** Extensible Markup Language, eräs kuvauskieli

## 2 Versio historia

versio	päiväys	tehdyt muutokset
1.0	13.6.2005	Ensimmäinen versio.
1.1	17.6.2005	Tarkastuksessa 13.6.2005 havaitut virheet korjattu.
1.2	21.6.2005	Kirjoitusvirheitä korjattu, sekä arkkitehtuuri- ja yleiskaavio korjattu.



## 3 Yleiskuvaus

Projekti tuottaa tapaamisaikataulujen laatimisjärjestelmän. Järjestäjä syöttää tiedot kursseista ja aikatauluista. Hän syöttää tarjottavat ajat manuaalisesti ja lisää opiskelijat leikepöydän kautta toisesta sovelluksesta tai syöttämällä tiedot manuaalisesti. Opiskelijat saavat tiedot tarjotuista ajoista ja valmistuneesta aikataulusta sähköpostilla. Opiskelijat syöttävät omat aikojen sopivuutensa tarjottuihin aikoihin. Järjestelmä tuottaa kaikille parhaan mahdollisen tapaamisaikataulun. Ohjelmisto toteutetaan selainpohjaisena järjestelmänä, jonka käyttäjinä ovat tietojenkäsittelytieteen laitoksen opiskelijat ja henkilökunta. Järjestäjältä vaaditaan sähköpostiosoite, joka kuuluu joko helsinki.fi- tai cs.helsinki.fi-toimialueeseen. Järjestelmä on asennettu laitoksen palvelimille ja, sitä voidaan käyttää Internetin välityksellä.

### 3.1 Arkkitehtuurin yleiskuvaus

Järjestäjän syöttää tarjottavat tapaamisajat aikatauluun, jotka tallentuvat järjestelmän tietokantaan. Järjestäjä syöttää osallistujat joko käsin tai tuomalla ne leikepöydän kautta toisesta sovelluksesta. Opiskelijat syöttävät aikatauluun omat aikojen sopivuutensa, jotka tallentuvat järjestelmän tietokantaan. Tapaamisaikojen merkitsemisestä ja aikataulun valmistumisesta ilmoitetaan sähköpostilla. Järjestelmä huolehtii vanhojen tietojen poistosta automaattisesti. Kuvassa 1 on arkkitehtuurikaavio.



Kuva 1: Arkkitehtuurikaavio.

Järjestelmä koostuu seuraavista komponenteista

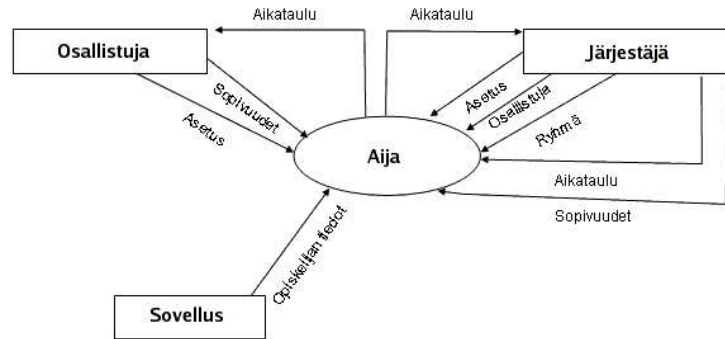
**Aija** Aija-järjestelmä on ohjelmiston ydin. Aija autentikoi käyttäjän. Järjestelmässä on tietokanta, johon tallennetaan järjestelmään syötetyt tiedot, aikataulutiedot ja ryhmätiedot. Aijam algoritmi laskee parhaan mahdollisen aikatauluuehdotuksen järjestäjälle. Aikatauluuehdotus lasketaan osallistujien antamien henkilökohtaisten sopivuuksien mukaan. Järjestäjä viestittää osallistujia sähköpostilla.

**Sovellus** Ulkoisesta sovelluksesta saadaan CSV-muodossa tiedot ryhmän osallistujista, jotka voidaan tuoda Aija-järjestelmään leikepöydän kautta.

**SMTP-palvelin** SMTP-palvelin välittää sähköpostiviestit.

## 3.2 Yhteyskaavio

Sidosryhmiin kuuluvat järjestelmän käyttäjät ja ulkoiset järjestelmät, jotka tuottavat järjestelmän toiminnan kannalta keskeistä tietoa. Kuvassa 2 on yhteyskaavio.



Kuva 2: Yhteyskaavio.

**Aija** Aija-järjestelmä antaa ehdotukset aikataulusta, välittää viestejä sähköpostilla järjestäjältä osallistujille ja Aijan viestejä järjestäjälle.

**Osallistuja** Osallistuja syöttää itselleen sopivia aikoja ja muuttaa henkilökohtaisia asetuksiaan.

**Järjestäjä** Järjestäjä luo uusia ryhmiä, syöttää osallistujia, luo aikatauluja, syöttää tapaamisaikoja tarjolle ja muuttaa henkilökohtaisia asetuksiaan. Järjestäjä lähettää viestejä osallistujille.

**Sovellus** Sovellus on ulkoinen järjestelmä, josta saadaan osallistujatietoja.

## 4 Käyttäjävaatimukset

### 4.1 Sisäänkirjautuminen

Käyttäjä voi kirjautua järjestelmään käyttämällä erityistä linkkiä, jonka järjestelmä on lähettänyt kutsun yhteydessä.

Lähde: Joni Salmi

#### 4.1.1 Sisäänkirjautuminen tunnistautumalla

Käyttäjä kirjautuu järjestelmään käyttämällä tunnistautumista antamalla käyttäjätunnuksiksi sähköpostiosoitteensa ja henkilökohtaisen salasanan. Järjestelmä tarkistaa annetut tiedot ja tunnistaa käyttäjän. Mikäli käyttäjän tietoja ei ole vielä

järjestelmässä, järjestelmä antaa tästä ilmoituksen ja käyttäjä rekisteröityy uudeksi käyttäjäksi. Jos käyttäjä ei muista salasanaanansa, järjestelmä lähettää käyttäjän pyynnöstä salasanan annettuun sähköpostiosoitteeseen.

Lähde: Matti Mattila

#### **4.1.2 Sisäänkirjautuminen linkillä**

Käyttäjä kirjautuu järjestelmään erityisellä linkillä ohittamalla erillisen tunnistautumisen. Linkissä on osoite aikatauluun sekä yksilöllinen tieto, jolla järjestelmä tunnistaa käyttäjän.

Lähde: Joni Salmi

#### **4.1.3 Epäonnistunut sisäänkirjautuminen linkillä**

Jos linkissä parametrina annettua yksilöllistä tietoa ei löydy järjestelmästä, järjestelmä ohjaa osallistujan sisäänkirjautumiseen ja antaa virheellisestä sisäänkirjautumisyrittäyksestä ilmoituksen osallistujalle.

Lähde: Matti Mattila

#### **4.1.4 Uusi käyttäjä**

Järjestäjän on voitava ryhtyä järjestelmän käyttäjäksi ilman ulkopuolista apua, mikäli hän omaa sallitun sähköpostiosoitteen.

Lähde: Joni Salmi

#### **4.1.5 Rekisteröityminen**

Jos käyttäjän tunnistetietoja, kuten sähköpostiosoitetta ja salasanaa ei ole vielä järjestelmässä, käyttäjä rekisteröityy uudeksi käyttäjäksi sisäänkirjautumisen sijaan. Kun käyttäjä on rekisteröity järjestelmään, se lähettää annettuun sähköpostiosoitteeseen generoidun salasanan.

Lähde: Matti Mattila

#### **4.1.6 Salasanan lähetys**

Mikäli käyttäjä on unohtanut salasanaanansa, järjestelmä lähettää sen käyttäjän pyynnöstä annettuun sähköpostiosoitteeseen. Jos annettua sähköpostiosoitetta ei löydy järjestelmästä, käyttäjä saa tästä ilmoituksen ja käyttäjä voi rekisteröityä uudeksi käyttäjäksi tai tarkistaa antamansa sähköpostiosoitteen.

Lähde: Matti Mattila

#### **4.1.7 Sähköpostiosoite käyttäjätunnuksena**

Sähköpostiosoitteeksi hyväksytään mikä tahansa osoite. Ainoastaan ennalta määrättyjen toimialueiden sähköpostiosoitteilla voidaan luoda uusia ryhmiä ja aikatauluja.

Lähde: Joni Salmi

#### **4.1.8 Järjestäjälle sallitut sähköpostiosoitteet**

Sallitut toimialueet on lueteltu järjestelmän tietovarastossa. Listaa täytyy voida ylläpitää helposti, esimerkiksi XML-tiedostolla.

Lähde: Matti Mattila

#### **4.1.9 Sähköpostiosoitteen muoto**

Sähköpostiosoitteen muodossa noudatetaan RFC2821 (Simple Mail Transfer Protocol) dokumentaatiota.

Lähde: Matti Mattila

#### **4.1.10 Sähköpostiosoitteen pakollisuus**

Järjestäjällä on oltava sähköpostiosoite.

Lähde: Matti Mattila

#### **4.1.11 Salasana**

Salasanassa saa olla erikoismerkkejä, numeroja sekä isoja ja pieniä kirjaimia. Salasanan on oltava vähintään 1 ja enintään 32 merkkiä pitkä, tyhjä ei kelpaa.

Lähde: Joni Salmi

#### **4.1.12 Salasanan vaihtaminen**

Käyttäjä voi milloin tahansa vaihtaa salasanansa.

Lähde: Matti Mattila

#### **4.1.13 Sisäänkirjautumislinkki**

Sisäänkirjautumislinkissä oleva yksilöllinen tieto ei saa olla helposti arvattavissa ja järjestelmän on tuotettava tämä tieto automaattisesti. Tietona voidaan käyttää GUID (globally unique identifier) arvoa.

Lähde: Matti Mattila

#### **4.1.14 Epäonnistunut tunnistautuminen**

Järjestelmä tunnistaa käyttäjän ja jos tunnistautuminen epäonnistuu, antaa järjestelmä tästä ilmoituksen ja tarvittaessa vie käyttäjän sisäänkirjautumistilaan. Käyttäjän sisäänkirjautumisyritysten määrää ei rajoiteta.

Lähde: Matti Mattila

## **4.2 Monikielisuuden tuki**

### **4.2.1 Tuetut kielet**

Järjestelmän on kyettävä tukemaan useita kieliä. Tuettujen kielten määrää ei ole rajoitettu.

Lähde: Joni Salmi

### **4.2.2 Kielen valinta**

Käyttäjä voi vaihtaa käyttöliittymän kielen milloin tahansa. Sisäänkirjautumisen yhteydessä valinta on selvästi esillä, jolloin käyttäjä kykenee vaihtamaan kielen helposti. Valittu kieli tallentuu muistiin seuraavia istuntokertoja varten, jolloin järjestelmä näyttää käyttöliittymän heti valitulla kielellä.

Lähde: Matti Mattila

### **4.2.3 Kielien ylläpito**

Tuettuja kieliä on kyettävä helposti ylläpitämään esimerkiksi XML-tiedostoina.

Lähde: Matti Mattila

## **4.3 Ryhmän ja aikataulun luonti**

### **4.3.1 Oikeus ryhmän ja aikataulun luontiin**

Käyttäjä voi luoda uuden ryhmän ja aikataulun vain jos hänen sähköpostiosoitteensa kuuluu sallittujen toimialueiden piiriin.

Lähde: Joni Salmi

### **4.3.2 Järjestäjän aloitusnäyttö**

Kun järjestäjä on kirjautunut sisään tunnistautumalla, on järjestelmä oletusarvoisesti uuden ryhmän luontitilassa mikäli sisäänkirjautuminen on järjestäjän ensimmäinen.

mäinen. Jos järjestäjä on aiemmin kirjautunut sisään joko tunnistautumalla tai sisäänkirjautumislinkillä, on aloitusnäytöllä edellisen istuntokerran näyttö.

Lähde: Joni Salmi

### **4.3.3 Uuden ryhmän luonti**

Järjestäjä voi luoda uuden ryhmän.

Lähde: Matti Mattila

### **4.3.4 Ryhmän perustiedot**

Ryhmällä on nimi, jonka enimmäispituus on 255 merkkiä.

Lähde: Matti Mattila

### **4.3.5 Aikataulun perustiedot**

Aikataululla on nimi, jonka enimmäispituus on 255 merkkiä, ja tyyppi (esimerkiksi henkilökohtainen- tai ryhmätapaaminen). Järjestäjälle on tarjolla valmiita tuntijakoja, joita voivat olla esimerkiksi 10, 15, 20, 30 ja 60 minuuttia. Tuntijakoa ei voi muuttaa, jos aikataulussa on merkintöjä. Järjestäjä valitsee listalta aikataulun aloitusajan ja lopetusajan kellonaikoina.

Lähde: Matti Mattila

### **4.3.6 Tietojen tallennus**

Järjestäjä kykenee tallentamaan annetut tiedot. Aikataulun tunnit jaetaan valitun tuntijaon mukaan, jos aikataulussa ei ole merkintöjä. Aikataulussa näkyy vain ajat annettujen aloitus- ja lopetusaikojen väliltä.

Lähde: Matti Mattila

### **4.3.7 Tuntijako**

Valmiita tuntijakointervalleja tulisi helposti kyetä ylläpitämään.

Lähde: Joni Salmi

### **4.3.8 Tarjottavien aikojen syöttö**

Järjestäjä merkitsee tarjottavat ajat aikataulun viikkonäytöllä.

Lähde: Matti Mattila

#### **4.3.9 Aikataulun pituuden ilmoittaminen**

Järjestäjä merkitsee kalenterista viikot, joiden aikana aikataulu on voimassa.

Lähde: Matti Mattila

#### **4.3.10 Aikataulun tallentaminen**

Järjestäjä voi tallentaa aikataulun.

Lähde: Matti Mattila

#### **4.3.11 CSV-tiedoston ohjeistus**

Näytöllä on tarkat ohjeet CSV-tiedoston vaaditusta muodosta sekä ohjeistus tiedoston käytöstä jäsennyksessä.

Lähde: Joni Salmi

#### **4.3.12 Osanottajien syöttö leikepöydältä**

Järjestäjä lisää osanottajia tapaamiselle liittämällä CSV-tiedoston sisällön leikepöydältä. Järjestelmä jäsentää tiedot ottamalla huomioon mahdollisen otsikkorivin sekä yrittämällä tunnistaa sarakkeet.

Lähde: Joni Salmi

#### **4.3.13 Tunnistettavat sarakkeet**

Järjestäjälle näytetään ohjeet CSV-tiedoston jäsentämisestä. Ohjeissa on kuvattu, missä muodossa CSV-tiedoston on oltava jäsentämistä varten. Kuvauksessa kerrotaan että tiedostossa on oltava otsikkorivi ensimmäisellä rivillä sekä sarakkeet ja niiden järjestys osallistujan nimeä ja sähköpostiosoitetta varten.

Lähde: Matti Mattila

#### **4.3.14 CSV-tiedoston jäsenys**

Järjestelmä jäsentää leikepöydältä liitetyn CSV-tiedoston, kun järjestäjä käynnistää toiminnon. Järjestelmä antaa järjestäjälle riittävät ilmoitukset jäsennyksen mahdollisesti epäonnistuessa.

Lähde: Matti Mattila

### **4.3.15 Osallistujan poistaminen ryhmästä**

Järjestäjä voi poistaa osallistujan ryhmästä. Osallistuja poistuu ryhmästä mutta ei järjestelmästä.

Lähde: Matti Mattila

### **4.3.16 Yhden osallistujan lisääminen**

Järjestäjä voi syöttää yhdenkin osallistujan tiedot erillisiin kenttiinsä käyttämättä CSV-tiedoston jäsenystä.

Lähde: Matti Mattila

### **4.3.17 Osallistujan tietojen muuttaminen**

Järjestäjä kykenee muuttamaan osallistujan tietoja, kuten sähköpostiosoitetta ja mahdollista nimeä.

Lähde: Matti Mattila

### **4.3.18 Osallistujien määrä**

Osallistujien määrää ryhmässä ei ole rajoitettu.

Lähde: Joni Salmi

## **4.4 Kutsut**

### **4.4.1 Kutsujen lähetys osanottajille**

Järjestäjä lähettää kutsut osanottajille sähköpostitse. Kutsussa osanottajia pyydetään merkitsemään itselleen sopivat ajat tarjotuista ajoista. Kutsussa lähetetään yksilöllinen sisäänkirjautumislinkki.

Lähde: Joni Salmi

### **4.4.2 Kutsun lähetys yhdelle osallistujalle**

Järjestäjä voi lähettää kutsun vain yhdellekin osallistujalle.

Lähde: Matti Mattila

### **4.4.3 Kutsutekstin muotoilu**

Järjestäjä kykenee muotoilemaan kutsussa olevaa tekstiä ennen kuin se lähetetään.

Lähde: Joni Salmi



#### **4.4.4 Muotoillun kutsutekstin tallennus**

Muotoiltu kutsuteksti jää muistiin seuraavia kertoja varten ja järjestelmä ehdottaa sitä seuraavilla istuntokerroilla.

Lähde: Matti Mattila

#### **4.4.5 Muuttuvat tiedot kutsussa**

Muuttuvat tiedot kutsussa merkitään XML-tyyppisin elementein.

Lähde: Matti Mattila

#### **4.4.6 Elementtien lisääminen**

Järjestäjä voi lisätä listalta elementtejä tekstipohjaan.

Lähde: Joni Salmi

#### **4.4.7 Aikataulun poistaminen**

Järjestäjä voi poistaa aikataulun.

Lähde: Joni Salmi

#### **4.4.8 Ryhmän poistaminen**

Järjestäjä voi poistaa ryhmän, jolloin ryhmään kuuluvat aikataulut ja osallistujatkin poistuvat. Osallistujat eivät poistu kuitenkaan järjestelmästä.

Lähde: Matti Mattila

### **4.5 Osallistujan vastaus kutsuun**

#### **4.5.1 Sisäänkirjautuminen linkillä**

Osallistuja kirjautuu järjestelmään valitsemalla linkin, jossa on yksilöllinen tieto, jolla osallistuja tunnistetaan. Linkki on lähetetty osallistujalle sähköpostikutsussa.

Lähde: Joni Salmi

#### **4.5.2 Sisäänkirjautuminen tunnistautumalla**

Osallistuja kirjautuu järjestelmään tunnistautumalla kirjoittaen käyttäjätunnukseensa sähköpostiosoitteensa sekä salasanan, jonka hän on saanut sähköpostikutsussa.

Lähde: Matti Mattila

#### **4.5.3 Aikataulun valinta linkillä**

Järjestelmä valitsee aikataulun valmiiksi sisäänkirjautumislinkissä olevan tiedon perusteella.

Lähde: Matti Mattila

#### **4.5.4 Keskenpäiväinen aikataulu**

Aikataulu on keskenpäiväinen, jos osallistujia ei ole merkinnyt aikojen sopivuuksiaan kaikkiin tarjottuihin aikoihin.

Lähde: Matti Mattila

#### **4.5.5 Aikataulun valinta edellisen istunnon perusteella**

Jos osallistuja kirjautuu sisään tunnistautumalla, järjestelmä valitsee vanhimman keskenpäiväisen aikataulun. Jos keskenpäiväistä aikataulua ei kyetä valitsemaan, järjestelmä valitsee sen aikataulun, joka on viimeksi ollut valittuna. Jos viimeksi valittua aikataulua ei kyetä valitsemaan, järjestelmä valitsee uusimman aikataulun. Aikataulua ei näytetä lainkaan, jos sitä ei kyetä valitsemaan.

Lähde: Matti Mattila

#### **4.5.6 Aikataulun valinta listalta**

Osallistuja voi valita aikataulun. Näytölle ilmestyy valitun aikataulun viikonäyttö.

Lähde: Matti Mattila

#### **4.5.7 Sopivien aikojen merkintä**

Osallistuja merkitsee itselleen sopivan tapaamisajan tarjotuista ajoista valitsemalla jonkin vaihtoehdon tapaamisajassa olevasta alavetovalikosta. Vaihtoehtoja tulee voida ylläpitää helposti. Paremmuusvaihtoehtona on oltava myös 'ei tiedossa'.

Lähde: Joni Salmi

#### **4.5.8 Sopivan ajan muuttaminen**

Osallistuja voi myöhemmin muuttaa aiemmin ilmoittamansa ajan sopivuuden.

Lähde: Joni Salmi

#### **4.5.9 Järjestäjä merkitsee osallistujan sopivat ajat**

Järjestäjä voi tarvittaessa merkitä aikojen sopivuudet osallistujan puolesta.

Lähde: Joni Salmi

#### **4.5.10 Sopivat ajat**

Oletusarvoisesti osallistujalle sopivat ajat ovat aluksi 'ei tiedossa'. Osallistujan on annettava kaikkiin tarjottuihin aikoihin paremmuusmerkintä.

Lähde: Joni Salmi

### **4.6 Tapaamisaikojen järjestely**

Tapaamisen järjestäjä voi halutessaan järjestellä osallistujille varattuja tapaamisaikoja. Varauksia voi muuttaa vaikka kaikki osallistujat eivät olisikaan vielä ilmoittaneet sopivia aikojaan.

Lähde: Joni Salmi

#### **4.6.1 Ilmoitus sopivuuksien syöttämisestä**

Järjestelmä ilmoittaa sähköpostilla järjestäjälle, kun kaikki osallistujat ovat merkinneet aikojen sopivuudet ja algoritmi on laatinut alustavan aikataulun

Lähde: Joni Salmi

#### **4.6.2 Järjestelynäyttö**

Tapaamisaikojen järjestelynäytöllä on pystysarakkeissa päivät ja tarjotut kellonajat. Vaakariveillä näkyvät osallistujat.

Lähde: Joni Salmi

#### **4.6.3 Tapaamisajan sopivuuden esitys**

Tarjotun ajan ja osallistujan risteyskohdassa eli tapaamisajassa näkyy osallistujan ilmaisema sopivuus tarjottuun aikaan.

Lähde: Matti Mattila

#### **4.6.4 Ristiriitaisten varausten esitys**

Ristiriitaiset varaukset, jotka menevät esimerkiksi päällekkäin ja vaativat järjestäjän huomiota, erotetaan selvästi muista merkinnöistä.

Lähde: Matti Mattila

#### **4.6.5 Varatun tapaamisajan esitys**

Osallistujalle varattu tapaamisaika esitetään selvästi muista merkinnöistä.

Lähde: Matti Mattila

#### **4.6.6 Tapaamisaikojen varaaminen algoritmilla**

Algoritmi järjestelee tapaamisajat ja ehdottaa alustavia varauksia. Algoritmi muodostaa aikataulua sitä mukaa kun osallistujat merkitsevät sopivia aikoja.

Lähde: Joni Salmi

#### **4.6.7 Varattujen tapaamisaikojen esitys**

Osallistujat esitetään järjestelynäytöllä varattujen aikojen mukaisessa järjestyksessä siten, että varaukset muodostavat diagonaalin vasemmasta yläkulmasta oikeaan alakulmaan.

Lähde: Joni Salmi

#### **4.6.8 Algoritmin toiminta**

Algoritmi muodostaa aikataulua, jos aikatauluun tulee muutoksia, kuten uusia tai muuttuneita osallistujien merkintöjä.

Lähde: Matti Mattila

#### **4.6.9 Varausten uudelleenjärjestely**

Järjestäjä voi käynnistää algoritmin järjestelemään ajanvaraukset uudelleen.

Lähde: Matti Mattila

#### **4.6.10 Varausten uudelleenlajittelu**

Järjestäjä voi lajitella ajanvaraukset uudelleen siten, että varaukset muodostavat diagonaalin vasemmasta yläkulmasta oikeaan alakulmaan.

Lähde: Joni Salmi

#### **4.6.11 Varauksen vaihtaminen**

Järjestäjä voi vaihtaa algoritmin ehdottaman tai järjestäjän itse aiemmin muuttaman tapaamisajan.

Lähde: Matti Mattila

#### **4.6.12 Aikataulun lukitseminen**

Järjestäjä voi milloin tahansa lukita aikataulun, jolloin osallistujat eivät voi muuttaa aikojen sopivuuksiaan eikä algoritmi voi tehdä muutoksia lukinnan aikana. Lukinnan aikana järjestäjä voi manuaalisesti järjestellä järjestelmän ehdottamia ajanvarauksia.

Lähde: Matti Mattila

#### **4.6.13 Ilmoitus lukinnasta**

Sekä järjestäjälle että osallistujalle ilmoitetaan selvästi, että aikataulu on lukittu järjestäjän järjestellessä aikataulua manuaalisesti. Osallistuja voi katsella aikataulua, mutta ei tehdä merkintöjä siihen.

Lähde: Matti Mattila

#### **4.6.14 Aikataulun vapauttaminen**

Järjestäjä voi milloin tahansa vapauttaa lukitun aikataulun.

Lähde: Matti Mattila

#### **4.6.15 Valmistuneesta aikataulusta tiedottaminen**

Järjestäjä lähettää osallistujille ilmoituksen valmistuneesta aikataulusta. Ilmoitus ryhmätapaamisiin laaditun aikataulun valmistumisesta lähetetään kaikille osallistujille, mutta henkilökohtaiset aikataulut kullekin osallistujalle erikseen.

Lähde: Joni Salmi

#### **4.6.16 Ilmoituksen lähetys**

Järjestäjä laatii osallistujille sähköpostiviestin, jossa on tiedot valmistuneesta aikataulusta. Järjestäjä voi ennen ilmoituksen lähettämistä muotoilla ilmoitusta samoin periaattein kuin kutsujakin.

Lähde: Matti Mattila

### **4.7 Aikataulunäyttö**

#### **4.7.1 Tilannelaskurit**

Näytöllä näkyy, montako tarjolla olevaa aikaa järjestäjä on merkinnyt per aikataulu. Jos on tiedossa, järjestelmä näyttää, montako osallistujaa tapaamiselle on syötetty

eli ryhmän koon sekä tarjolla olevien aikojen riittävyyden näille osallistujille, jos esimerkiksi ryhmän koko on suurempi kuin tarjottujen aikojen määrä.

Lähde: Joni Salmi

#### **4.7.2 Kalenteri**

Jos näytöllä on kalenteri ja siinä useita kuukausia, erotetaan kuukaudet toisistaan vaihtuvalla taustavärillä.

Lähde: Joni Salmi

#### **4.7.3 Viikkonäyttö**

Viikkonäytön kellonajat aloitetaan tasatunnista. Järjestäjä ilmaisee viikkonäytöllä vain tarjottavat ajat.

Lähde: Joni Salmi

## **5 Ympäristövaatimukset**

### **5.1 Yleiskäyttöinen tietokantarajapinta**

Ohjelmiston ja tietokannan välinen rajapinta on yleiskäyttöinen mahdollistaen vähintään ANSI-SQL/92-standardia noudattavien tietokantojen käytön.

Lähde: Mikko Alakunnas

### **5.2 Laiteympäristö**

Ohjelman tulee toimia TKTL:n laitteistossa.

Lähde: Joni Salmi

### **5.3 Sähköpostin lähetys**

Ohjelman on pystyttävä lähettämään sähköpostia järjestelmän käyttäjille Linux tai Windows ympäristössä.

Lähde: Mikko Alakunnas

### **5.4 CSV-tiedostojen jäsennys**

Ohjelma pystyy jäsentämään määrämuotoista CSV-tekstiaineistoa.

Lähde: Joni Salmi

## 5.5 Käyttöliittymän toteutus

Käyttöliittymän toteutuksessa käytetään HTML 4.01 Transitional DTD -dokumenttimäärittelyä.

Lähde: Mikko Alakunnas

## 5.6 Erityiskäyttäjryhmät

Käyttöliittymässä käytetään selainturvallisia värejä, jotka muodostetaan heksadesimaalisilla RGB-arvoilla 00,33,66,99,CC ja FF.

Lähde: Mikko Alakunnas

## 5.7 Samanaikaisuuden hallinta

390 käyttäjän on pystyttävä käyttämään järjestelmää yhtäaikaaisesti.

Lähde: Joni Salmi

## 5.8 Monikielisyyden tuki

Ohjelman on oltava laajennettavissa usealle luonnolliselle kielelle.

Lähde: Joni Salmi

## 5.9 PHP-skriptien suoritus

TKTL vaatimuksesta PHP-skriptit suoritetaan CGI-skripteinä.

Lähde: Mikko Alakunnas

# 6 Toiminnalliset järjestelmävaatimukset

## 6.1 Taustatoiminnot

### 6.1.1 Vanhentuneiden tietojen poisto

Järjestelmä poistaa vanhentuneet tiedot ilman ulkopuolista apua tietyin väliajoin, esimerkiksi kerran viikossa.

Lähde: Joni Salmi

### 6.1.2 Vanhentunut aikataulu

Aikataulu on vanhentunut, kun aikataulun päättymisestä on kulunut yli 30 päivää.

Lähde: Matti Mattila

### **6.1.3 Vanhentunut käyttäjä**

Käyttäjä on vanhentunut, kun viimeisimmästä sisäänkirjautumisesta on kulunut yli 14 kuukautta. Jos käyttäjä ei ole kirjautunut kertaakaan sisään, katsotaan käyttäjä vanhentuneeksi, kun käyttäjätietojen luonnista on kulunut yli 14 kuukautta. Käyttäjätietojen vanheneminen alkaa, kun käyttäjä ei ole enää yhdessäkään voimassaolevassa ryhmässä osallistujana tai järjestäjänä.

Lähde: Matti Mattila

### **6.1.4 Vanhentunut ryhmä**

Ryhmä on vanhentunut, jos sille ei ole luotu aikatauluja 14 kuukauteen.

Lähde: Joni Salmi

## **6.2 Virheilmoitukset**

Ohjelmisto antaa käyttäjälle ilmoituksen virhetilanteista.

Lähde: Pasi Röytiö

## **6.3 Algoritmin suoritus**

Järjestelmä ilmoittaa käyttäjälle, jos aikataulujen muodostusalgorithmi on käynnissä.

Lähde: Pasi Röytiö

# **7 Ei-toiminnalliset järjestelmävaatimukset**

## **7.1 Tuetut selaimet**

Järjestelmä tukee selaimia, jotka noudattavat HTML 4.01 suositusta.

Lähde: Pasi Röytiö

## **7.2 Tietoturva**

Ulkopuoliset, eli muut kuin käyttäjä ja mahdollinen järjestäjä, eivät saa päästä muuttamaan käyttäjän syöttämiä tietoja.

Lähde: Pasi Röytiö



### 7.3 Aikataulun poistaminen

Aikataulu poistetaan järjestelmästä 30 päivää aikataulun viimeisestä tapaamisesta.

Lähde: Joni Salmi

### 7.4 Käsinsyötettyjen osallistujien poistaminen

Järjestelmä säilyttää käsinsyötettyjen osallistujien tietoja yhden vuoden. Jos tietoja ei ole käytetty vuoteen, järjestelmä poistaa tiedot.

Lähde: Joni Salmi

### 7.5 Ryhmän poistaminen

Järjestelmä säilyttää ryhmää järjestelmässä 1 vuoden, jonka jälkeen ryhmä poistetaan järjestelmästä.

Lähde: Joni Salmi

### 7.6 Käyttäjät

Järjestelmää käyttävät laitoksen opiskelijat ja henkilökunta, joilla on järjestettävään kurssiin, joihin kuuluu henkilökohtaisia tapaamisia ja ryhmätapaamisia. Tapaamisia ja aikatauluja voivat luoda kaikki, joiden sähköpostiosoite kuuluu ennalta määriteltäviin toimialueisiin.

### 7.7 Henkilökohtaisten aikojenjakoalgoritmi

Sääntö aikojen jaon vertailemiseksi on seuraava: kahdesta aikojen jaosta parempi on se, jossa

1. on enemmän yli nollan olevia sopivuuksia
2. sopivuuksien summa on suurempi
3. sopivuuksien hajonta on pienempi

Ehdot arvioidaan järjestyksessä: myöhäisempi ehto arvioidaan vain, jos aikaisemmillä ehdoilla ei saatu paremmuutta ratkaistua.

### 7.7.1 Käsitteitä

Kaksijakoinen verkko määritellään seuraavasti:  $G(A,B,E)$ , jossa  $A$  ja  $B$  ovat solmujoukkoja ja  $E$  on verkon kaarijoukko. Joukon  $A$  solmujen välillä ei ole kaaria eikä myöskään joukon  $B$  solmujen välillä ole kaaria.

Parituksen koolla tarkoitetaan paritukseen kuuluvien kaarien määrää.

Verkon paritusongelma: annettu kaksijakoinen verkko  $G(A,B,E)$ . Etsittävä joukkojen  $A$  ja  $B$  välinen paritus, jossa on mahdollisimman paljon kaaria, mutta jokainen joukkojen  $A$  ja  $B$  solmu saavat kuulua vain yhteen parituksen kaareen.

Aikojenjako-ongelma on palautettavissa kaksijakoisen verkon paritusongelmaan seuraavalla kuvauksella  $f$ :

- $f: (\text{osallistujat, tarjotut ajat, sopivuudet}) \rightarrow G(A,B,E)$
- $A = \text{Osallistujat}$
- $B = \text{Järjestäjän tarjoamat ajat}$
- $E = \text{Osallistujien ilmoittamat sopivuudet järjestäjän tarjoamiin aikoihin}$

Täydellisellä parituksella tarkoitetaan paritusta, jossa jokaisesta joukon  $A$  alkioista on kaari johonkin joukon  $B$  alkioon ja jokainen joukkojen  $A$  ja  $B$  alkio kuuluu vain yhteen kaareen. Maksimaalisella parituksella tarkoitetaan paritusta, jossa parituksen kaarien painojen summa on mahdollisimman suuri.

### 7.7.2 Aikojenjako-algoritmin toiminta

Tässä luvussa kuvataan aikojenjako-algoritmin toimintaa yleisellä tasolla.

Palautetaan henkilökohtaisten aikojenjako-ongelma kaksijakoisen verkon paritusongelmaan. Mahdollisimman hyvän parituksen etsivä algoritmi voidaan jakaa kolmeen eri vaiheeseen:

1. Etsitään  $n$  kertaa täydellisiä parituksia. Tässä vaiheessa ei kiinnitetä huomiota kaarien painoihin. Jos jokin parituksen koko on suurempi kuin muiden löydettyjen paritusten koko, on löydetty paras mahdollinen paritus, jonka perusteella muodostetaan aikojen jakaminen osallistujille. Jos löydettiin  $m$  kappaletta parituksia, joiden koko on yhtä suuri ja suurempi kuin muiden, siirrytään kohtaan 2.
2. Lasketaan kunkin parituksen kaarien painojen summa. Jos jonkin parituksen summa on suurempi kuin kaikkien muiden, on löydetty voittaja. Jos on  $o$  kappaletta parituksia, joiden summa on yhtä suuri ja suurempi kuin muiden paritusten summa, siirrytään kohtaan 3.

3. Lasketaan paritukseen kuuluvien joukon B solmujen hajonta. Valitaan voittajaksi paritus, jonka hajonta on pienin. Jos on monta paritusta, joiden joukon B solmujen hajonta on pienin, valitaan niistä ensimmäinen.

Edellä kuvattu algoritmi tuottaa täydellisen parituksen, jos sellainen on olemassa, mutta se ei takaa maksimaalisen parituksen löytämistä, vaikka sellainen olisikin olemassa. Kohdat 1 ja 2 olisi mahdollista yhdistää esimerkiksi Unkarilaisen algoritmin avulla (Hungarian method/algorithm). Unkarilainen algoritmi löytää täydellisen parituksen (jos sellainen on olemassa), jonka kaarien painojen summa on mahdollisimman suuri. Algoritmin tarkka toteutustapa päätetään toteutusvaiheessa.

## 8 Käyttötapaukset

Tässä esitetyt käyttötapaukset eivät kuvaa valmiin järjestelmän toiminnallisuutta, vaan antavat kuvan käyttäjän tavoitteista, joiden avulla järjestelmän toiminnallisuus suunnitellaan ja testataan. Käyttötapauksista esitetään käyttäjän tavoite ja tilatietoja, jotka kuvaavat käyttäjän ja järjestelmän tietoja tapaushetkellä.

Käyttötapaukset on jaoteltu käyttäjäryhmien mukaan. Tapauksia kuvaamaan on esitelty fiktiivinen, mutta tyypillinen käyttäjäryhmän henkilö. Käyttäjäryhmät ovat järjestäjä, joka aloittaa aikataulun muodostamisen ja toimii kutsujana sekä osallistuja, joka vastaa kutsuu ja osallistuu myöhemmin tapaamisiin.

Lisäksi on olemassa kolmas ryhmä, käyttäjä, jolloin ei vielä kuulu kumpaankaan edelliseen ryhmään, vaan haluaa esimerkiksi vain tutustua järjestelmään ja sen palveluihin.

### 8.1 Järjestäjä

#### 8.1.1 Onnistunut rekisteröityminen

Minnan tavoite: TKTL:n sivutoimisella tuntiopettajalla, Minnalla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Minna on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän löytää linkin laitoksen sivulta ja päättää rekisteröityä ja käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on maanantai 19.9.2005. Minna on laitoksen tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Minna syöttää sähköpostiosoitteensa sekä salasanansa ja lähettää tiedot. Järjestelmä tarkistaa, onko Minnan sähköpostiosoite sallittujen toimialueiden joukossa. Minnan rekisteröityminen hyväksytään, ja linkki vie hänen aikataulujen luontisivulle.

### 8.1.2 Epäonnistunut rekisteröityminen

Marin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Marilla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Mari on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän löytää linkin laitoksen sivulta ja päättää rekisteröityä ja käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on maanantai 19.9.2005. Mari on laitoksen tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Mari syöttää sähköpostiosoitteensa sekä salasanan ja lähettää tiedot. Järjestelmä tarkistaa, onko Marin sähköpostiosoite sallittujen toimialueiden joukossa. Marin sähköpostiosoite ei kuitenkaan kuulu sallittujen joukkoon, joten järjestelmä rekisteröi hänet osallistujaksi. Mari ei voi luoda uusia ryhmiä eikä aikatauluja ennen kuin on vaihtanut sähköpostiosoitteensa järjestäjille sallitulle toimialueelle.

### 8.1.3 Järjestäjä ei nimeä ryhmää

Einin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Einillä, on ohjattavanaan alkukesällä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut viisi opiskelijaa. Eini on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän on saanut kurssin vastuuhenkilöltä linkin järjestelmään ja käskyn rekisteröityä sekä käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on sunnuntai 5.6.2005. Eini on kotonaan tietokoneella. Hän avaa sähköpostistaan saaman linkin, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Eini rekisteröityy järjestelmään ja aloittaa luomaan tapaamisaikatauluja Tietorakenteiden harjoitustyön tapaamisille. Hän löytää Uusi ryhmä-linkin ja miettiessään vahingossa painaa Tallenna-painiketta nimeämättä ryhmää. Hän saa ilmoituksen, joka kehottaa antamaan ryhmälle nimen.

### 8.1.4 Järjestäjä antaa kahdelle ryhmälleen saman nimen

Harrin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Harrilla, on ohjattavanaan alkukesällä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut viisi opiskelijaa. Harri on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän on saanut kurssin vastuuhenkilöltä linkin järjestelmään ja käskyn rekisteröityä sekä käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on sunnuntai 5.6.2005. Harri on kotonaan tietokoneella. Hän avaa sähköpostistaan saaman linkin, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Harri rekisteröityy järjestelmään ja aloittaa luomaan tapaamisaikatauluja Tietorakenteiden harjoitustyön tapaamisille. Hän luo Uusi ryhmä -linkistä uuden ryhmän ja nimeää sen. Vahingossa hän on antanut ryhmälle saman nimen, kuin toiselle hänellä yhtä aikaa olevalle ryhmälle. Hän ehtii kuitenkin painaa Tallenna-painiketta ja nimeää ryhmän. Hän saa ilmoituksen, joka kehottaa antamaan uuden nimen ryhmälle, koska hänellä on jo samanniminen ryhmä.

### **8.1.5 Laskariryhmän ohjausaikojen tarjoaminen**

Markun tavoite: TKTL:n sivutoimisella tuntiopettajalla, Markulla, on ohjattavanaan alkukesällä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut viisi opiskelijaa. Markku on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän on saanut kurssin vastuuhenkilöltä linkin järjestelmään ja käskyn rekisteröityä sekä käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on sunnuntai 5.6.2005. Markku on kotonaan tietokoneella. Hän avaa sähköpostistaan saaman linkin, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Markku rekisteröityy järjestelmään ja syöttää sinne sopivat ajat Tietorakenteiden harjoitustyön tapaamisille. Hän lähettää osallistujille järjestelmän avulla sähköpostitse tiedon asiasta.

### **8.1.6 Harjoitusryhmän aikataulun muodostaminen**

Minnan tavoite: TKTL:n sivutoimisella tuntiopettajalla, Minnalla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Minna on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen. Hän on löytänyt linkin laitoksen sivulta, rekisteröitynyt, syöttänyt tarjottavat ajat, ja järjestelmä on lähettänyt tiedot osallistujille sähköpostitse. Osallistujien syötettyä omat sopivuutensa, järjestäjä saa sähköpostiinsa tiedon, että aikataulu on valmis järjesteltäväksi.

tilatietoja: Nyt on keskiviikko 28.9.2005. Minna on laitoksella tietokoneella. Hän avaa laitoksen kotisivun, josta voi kirjautua Aija-järjestelmään.

Toteutunut skenaario: Minna kirjautuu järjestelmään ja aloittaa aikataulun järjestelyn. Hän tutkii järjestelmän luomaa henkilökohtaisten tapaamisten aikataulua. Kolmen opiskelijan parhaat sopivuudet menevät päällekkäin, mutta manuaalisesti vaihtamalla toiseksi ja kolmanneksi parhaita aikoja kaikki saavat ajan, joka sopii. Lisäksi Minna on tyytyväinen, kun hänen ei tarvitse tulla joka aamu klo 8.15 töihin, vaikka hän olikin kaikki ne ajat vapaaksi tarjonnut. Minna lopettaa aikataulun

järjestelyn ja lähettää tiedon valmistuneesta aikataulusta osallistujille sähköpostilla.

### **8.1.7 Harjoitusryhmän tapaamisen peruuttaminen**

Minnan tavoite: TKTL:n sivutoimisella tuntiopettajalla, Minnalla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Minna on luonut Aija-järjestelmän avulla henkilökohtaisille ja ryhmätapaamisille aikataulun, mutta nyt hän on sairastunut influenssaan ja joutuu perumaan koko viikon tapaamiset. Samalla pitäisi sopia uudet tapaamiset.

tilatietoja: Nyt on lauantai 1.10.2005. Minna on kotonaan tietokoneella. Hän avaa laitoksen kotisivun, josta voi kirjautua Aija-järjestelmään.

Toteutunut skenaario: Minna kirjautuu järjestelmään. Hän merkitsee aikatauluun itselleen sopivia aikoja peruuntuneiden aikojen korvaamiseksi, joista opiskelijat voivat valita puolestaan heille sopivimmat. Minna lähettää kaikille ryhmäläisille peruutukset koko viikon tapaamisista sähköpostilla ja pyytää merkitsemään uudet sopivat ajat peruuntuneen ajan tilalle.

## **8.2 Käyttäjä**

### **8.2.1 Järjestelmään tutustuminen**

Sallan tavoite: TKTL:n sivutoimisella tuntiopettajalla, Sallalla, on ohjattavanaan aina silloin tällöin harjoitustyön ryhmiä, mutta ei tällä hetkellä. Hän on kuitenkin kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen ja haluaisi tutustua siihen, jotta voisi sitä mahdollisesti myöhemmin käyttää.

tilatietoja: Nyt on maanantai 19.9.2005. Salla on kotonaan tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi tutustumislinkin avulla kirjautumatta ottaa selvää järjestelmän palveluista.

Toteutunut skenaario: Salla avaa tutustumislinkin ja ottaa selvää järjestelmän tarjoamista palveluista. Hän poistuu järjestelmästä rekisteröitymättä ja kirjautumatta sisään.

### **8.2.2 Kielen valinta**

Sallyn tavoite: TKTL:n sivutoimisella tuntiopettajalla, Sallylla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Sally on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen. Hän on löytänyt linkin laitoksen sivulta. Hänen äidinkieltensä on englantia.

tilatietoja: Nyt on keskiviikko 28.9.2005. Sally on laitoksella tietokoneella. Hän avaa laitoksen kotisivun, josta voi kirjautua Aija-järjestelmään.

Toteutunut skenaario: Minna avaa laitoksen sivun ja löytää Aija-järjestelmän linkin. Hän huomaa kielen vaihtomahdollisuuden ja vaihtaa käyttöliittymän kielen englanniksi.

### 8.2.3 Salasanan antaminen

Jarin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Jarilla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 8 opiskelijaa. Jari on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän löytää linkin laitoksen sivulta ja päättää rekisteröityä ja käyttää sitä ohjausaikojen sopimiseen.

tilatietoja: Nyt on maanantai 19.9.2005. Jari on laitoksen tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään

Toteutunut skenaario: Jari syöttää sähköpostiosoitteensa ja lähettää tiedot. Järjestelmä tarkistaa, onko Jarin sähköpostiosoite sallittujen toimialueiden joukossa ja luo salasanan. Jarin rekisteröityminen hyväksytään, ja linkki vie hänen aikataulujen luontisivulle.

### 8.2.4 Salasanan unohtaminen

Sarin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Sarilla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 8 opiskelijaa. Sari on käyttänyt Aija-järjestelmää opiskelijoiden tapaamisaikataulujen järjestämiseen ja nyt hän haluaisi katsoa kuinka moni opiskelijoista on jo sopivuutensa ilmoittanut järjestelmään.

tilatietoja: Nyt on keskiviikko 21.9.2005. Sari on laitoksen tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään

Toteutunut skenaario: Sari syöttää sähköpostiosoitteensa ja salasanaan ja lähettää tiedot. Järjestelmä kuitenkin ilmoittaa, että Sarin salasana on virheellinen. Sari antaa salasanan uudelleen. Salasana on edelleen virheellinen, eikä Sari muista salanaanansa.

### 8.2.5 Unohtuneen salasanan lähettäminen

Sarin tavoite: TKTL:n sivutoimisella tuntiopettajalla, Sarilla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 8 opiskelijaa. Sari on käyttänyt Aija-järjestelmää opiskelijoiden tapaamisaikataulujen järjestämiseen ja nyt hän haluaisi katsoa kuinka moni opiskelijoista on jo sopivuutensa ilmoittanut järjestelmään.

tilatietoja: Nyt on keskiviikko 21.9 2005 . Sari on laitoksen tietokoneella. Hän avaa laitoksen kotisivun, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Sari syöttää sähköpostiosoitteensa sekä salasanaan ja lähettää tiedot. Järjestelmä kuitenkin ilmoittaa, että Sarin salasana on virheellinen. Sari antaa salasanan uudelleen. Salasana on edelleen virheellinen, eikä Sari muista salasanaan. Hän lähettää "Lähetä salasana-painikkeesta pyynnön saada oma salasanaan sähköpostiosoitteeseensa. Järjestelmä lähettää salasanan hänelle.

### **8.2.6 Salasanan vaihtaminen**

Karin tavoite: TKTL:n opiskelijalla, Karilla, on suoritettavanaan syksyllä 2005 Tietorakenteiden harjoitustyö. Kari on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän saa linkin ohjaajalta ja käskyn merkitä tapaamisaikojen sopivuuksia järjestelmään.

tilatietoja:: Nyt on maanantai 19.9.2005. Kari on laitoksen tietokoneella. Hän avaa linkin, joka johtaa Aija-järjestelmän aikataulusivulle, josta voi merkitä sopivuuksia järjestelmään.

Toteutunut skenaario: Kari syöttää sopivuuksia ja huomaa Asetukset-linkin. Karin avaa linkin ja vaihtaa salasanaan helpommin muistettavaan.

### **8.2.7 Epäonnistunut sisäänkirjautuminen linkillä**

Larin tavoite: TKTL:n opiskelijalla, Larilla, on suoritettavanaan syksyllä 2005 Tietorakenteiden harjoitustyö. Lari on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä opiskelijoiden tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän saa linkin ohjaajalta ja käskyn merkitä tapaamisaikojen sopivuuksia järjestelmään.

tilatietoja:: Nyt on maanantai 19.9.2005. Lari on laitoksen tietokoneella. Hän avaa linkin, joka johtaa Aija-järjestelmän aikataulusivulle, josta voi merkitä sopivuuksia järjestelmään.

Toteutunut skenaario: Lari yrittää avata linkkiä, mutta linkki ei toimi. Lari päättää kirjautua laitoksen sivuilta Aija-järjestelmään, sähköpostissa saamallaan käyttäjätunnuksella ja salasalla.

### **8.2.8 Osallistujien lisääminen Kurki-kurssinkirjanpitojärjestelmästä**

Minnan tavoite: TKTL:n sivutoimisella tuntiopettajalla, Minnalla, on ohjattavanaan syksyllä 2005 yksi Tietorakenteiden harjoitustyön ryhmä. Ryhmään on ilmoittautunut 10 opiskelijaa. Minna on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä



opiskelijoiden tapaamisaikataulujen järjestämiseen. Hän on löytänyt linkin laitoksen sivulta, rekisteröitynyt ja päättää luoda osallistujaryhmän.

tilatietoja: Nyt on keskiviikko 28.9.2005. Minna on laitoksella tietokoneella. Hän avaa laitoksen kotisivun, josta voi kirjautua Aija-järjestelmään.

Toteutunut skenaario: Minna kirjautuu järjestelmään. Hän luo Tietorakenteiden ryhmän ja avaa Uusi osallistuja -linkin, jolloin osanottajien käsittelynäyttö ilmestyy. Osallistujat on saatu Kurki-kurssikirjanpitojärjestelmästä. Minna lisää opiskelijat osallistujaryhmäänsä.

### **8.2.9 Osallistujien lisääminen manuaalisesti**

Samin tavoite: TKTL:n professorilla, Samilla, on pidettävänä syksyllä 2005 alaisilleen kehityskeskustelut. Alaisia hänellä on kahdeksan. Sami on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän on löytänyt laitoksen sivulta linkin järjestelmään ja päättänyt käyttää sitä kehityskeskusteluaikojen sopimiseen.

tilatietoja: Nyt on sunnuntai 25.9.2005. Sami on kotonaan tietokoneella. Hän avaa laitoksen sivulta linkin, joka johtaa Aija-järjestelmän etusivulle, josta voi rekisteröityä järjestelmään.

Toteutunut skenaario: Sami kirjautuu järjestelmään. Hän luo Kehityskeskusteluryhmän ja avaa Uusi osallistuja -linkin, jolloin osanottajien käsittelynäyttö ilmestyy. Täällä hän kirjoittaa osallistujan nimen ja sähköpostiosoitteen ja tallentaa tiedot.

## **8.3 Osallistuja**

### **8.3.1 Omien sopivuuksien ilmoittaminen**

Jarkon tavoite: TKTL:n opiskelijalla, Jarkolla, on suoritettavanaan Tietorakenteiden harjoitustyö alkukesällä 2005. Jarkko on kuullut Ohtuprojektin tekemästä Aija-järjestelmästä tapaamisaikataulujen järjestämiseen, mutta hän ei ole käyttänyt kyseistä järjestelmää. Hän on saanut kurssin ohjaajalta sähköpostissa linkin järjestelmään ja käskyn käyttää sitä ohjausaikojen sopimiseen.

Tilatietoja: Nyt on perjantai 10.6.2005. Jarkko on kotonaan tietokoneella. Hän avaa sähköpostistaan saaman linkin, joka johtaa Aija-järjestelmän aikataulusivulle, jossa voi merkitä itselleen sopivat ajat järjestelmään.

Toteutunut skenaario: Jarkko kirjautuu järjestelmään automaattisesti saamallaan linkillä ja syöttää sinne itselleen sopivat ajat järjestäjän tarjoamista vaihtoehdoista Tietorakenteiden harjoitustyön henkilökohtaisille ja ryhmätapaamisille.

### 8.3.2 Tapaamisaikojen vastaanottaminen

Jarkon tavoite: TKTL:n opiskelijalla, Jarkolla, on suoritettavanaan Tietorakenteiden harjoitustyö alkukesällä 2005. Jarkko on ilmoittanut itselleen sopivat tapaamisaajat järjestäjän tarjoamista ajoista Aija-järjestelmän kautta. Hän on saanut kurssin ohjaajalta sähköpostissa linkin järjestelmään ja ilmoituksen tapaamisaikataulun valmistumisesta.

Tilatietoja: Nyt on perjantai 17.6.2005. Jarkko on kotonaan tietokoneella. Hän avaa sähköpostistaan saaman linkin, joka johtaa Aija-järjestelmän aikataulusivulle, jossa voi tutustua valmistuneeseen aikatauluun.

Toteutunut skenaario: Jarkko kirjautuu järjestelmään automaattisesti saamallaan linkillä ja tarkistaa oman henkilökohtaisen tapaamisensa ajankohdan. Sitten hän katsoo ryhmätapaamisen ajankohdan.

## 9 Toteutettava toiminnallisuus

Tässä luvussa esitellään toteutettavan ohjelmiston toiminnalliset ominaisuudet. Ominaisuudet on jaoteltu siten, että aluksi eritellään asiakkaan esittämät vaatimukset toiminnallisuudelle. Vaatimukset on priorisoitu niin, että ensin toteutetaan 1 tason vaatimukset ja sitten 2 tason vaatimukset. Lopuksi esitellään ryhmän toteutettavaksi ideoimat lisäominaisuudet. Näiden toteutuksen suhteen ei tehdä sitoumuksia.

### 9.1 Ensimmäisen tason vaatimukset

- Sisäänkirjautuminen linkillä ja tunnistautumalla
- Käyttäjän itserekisteröityminen
- Uloskirjautuminen
- Järjestäjälle sallitun sähköpostiosoitteen tunnistus
- Monikielisyyden tuki suomeksi ja englanniksi
- Kielien ylläpito XML-tiedostoilla
- Ryhmän luonti, muutos ja poisto
- Aikataulun luonti, muutos ja poisto
- Osallistujan lisäys määrämuotoisesta CSV-tiedostosta jäsentämällä
- Osallistujan manuaalinen lisäys, muutos ja poisto
- Kutsujen lähetys sähköpostilla valituille osallistujille

- Tarjottavien tapaamisaikojen merkintä aikatauluun
- Myös järjestäjä voi merkitä osallistujalle sopivia tapaamisaikoja aikatauluun
- Tapaamisaikojen järjestely algoritmilla taustalla
- Tapaamisaikojen manuaalinen järjestely
- Sähköposti-ilmoitus valmiudesta aikataulujen järjestelyä varten
- Sähköposti-ilmoitus valmiista aikataulusta osallistujille
- Sähköpostiviestipohja on sekä suomeksi että englanniksi samassa viestissä

## 9.2 Toisen tason vaatimukset

- Unohtuneen salasanan lähetys
- Salasanan vaihtaminen
- Oman nimen ja sähköpostiosoitteen vaihtaminen
- Kutsussa käytettävän sähköpostiviestin muotoilu
- Tiedotteessa käytettävän sähköpostiviestin muotoilu
- Vanhentuneiden aikataulujen ja osallistujien poisto automaattisesti
- Muotoillun sähköpostiviestipohjan tallennus

## 9.3 Lisäominaisuudet

- Sarakkeiden tunnistus mielivaltaisesta CSV-tiedostosta
- Elementtien lisäys sähköpostiviestipohjiin muuten kuin manuaalisesti
- Järjestelmän esittelymateriaali
- Yleiskäyttöinen tietokantaliittymä muille kuin PostgreSQL-tietokannalle
- Käyttöliittymä ylläpitotoiminnoille

## 9.4 Erikois- ja virhetilanteet

Erikoistilanteet ovat ohjelmoinnissa tiedostettuja tilanteita, jotka eivät liity ohjelmakoodissa tapahtuviin virheisiin, vaan ovat seurausta käyttötapauksesta, joka ei ole syystä tai toisesta loogisesti toteutettavissa. Virhetilanteet ovat tilanteita, joissa järjestelmä ei ole riittävän toimintakunnossa toteuttaakseen jonkin toiminnon.

Erikois- ja virhetilanteet eivät saa vaikuttaa pysyvästi talletettujen tietorakenteiden (tietokannan, konfiguraatiotiedostojen jne.) toimivuuteen. Molemmista ilmoitetaan aina käyttäjälle mahdollisimman informatiivisesti erillisellä virheilmoituksella. Erikoistilanteita ovat ainakin:

Session voimassaolon umpeutuminen: järjestelmä käyttää timeout-toteutusta, ts. session ollessa joutilas yli määrätyn ajan sen voimassaolo umpeutuu, jonka jälkeen käyttäjä joutuu kirjautumaan uudestaan sisälle päästäkseen toiminnallisuuteen käsiksi. Järjestäjällä ei ole hyväksyttävää sähköpostiosoitetta. Ainoastaan ennalta määrättyjen toimialueiden sähköpostiosoitteilla voidaan luoda uusia tapaamisia. Käyttäjä syöttää salasansa väärin. Hän saa ilmoituksen asiasta ja mahdollisuuden syöttää salasana uudestaan.

Virhetilanteita ovat esim. ainakin:

- Tietokantaan ei saada yhteyttä
- Sähköpostipalvelimeen ei saada yhteyttä

## 10 Muut ominaisuudet

Tässä luvussa käsitellään muita ohjelmistolta vaadittavia ominaisuuksia.

### 10.1 Suorituskyky

Järjestelmän käyttöliittymä, PHP-skriptit, algoritmit ja tietokantakyselyt suunnitellaan riittävän tehokkaiksi.

Järjestelmän tulee toimia Helsingin Yliopiston palvelimilla niin tehokkaasti, että järjestelmä pysyy käyttökelpoisena ottaen huomioon, että järjestelmässä saattaa olla pahimmillaan noin 400 yhtäaikaista käyttäjää. Selainpohjaisena järjestelmän tiedonsiirtotarve on suuri, mutta kerrallaan siirrettävät tietomäärät ovat melko pieniä. Aija-järjestelmästä aiheutumaton verkkoviive saattaa aiheuttaa järjestelmän käytön hidastumista. Jos verkkoviivettä ei oteta huomioon, järjestelmän vasteaikojen tulee pysyä pieninä.

## 10.2 Käytettävyys

Aija-järjestelmän käyttöliittymä on erityisesti tarkoitettu Helsingin Yliopiston tietojenkäsittelytieteen laitoksen harjoitustöiden tapaamisaikojen varaamiseen. Tästä huolimatta käyttöliittymän on oltava yleistettävä, jotta se sopisi myös muunlaisten tapaamisten sopimiseen. Käyttöliittymän tulee olla selkeä ja helppokäyttöinen. Erityistä huomiota kiinnitetään siihen, että tavallisimmat käyttötapaukset voidaan suorittaa helposti.

Järjestelmän tulee ilmoittaa käyttäjälle mahdollisista virhetilanteista. Virheitä voivat aiheuttaa mm. käyttäjän väärin syöttämät tiedot ja epäonnistunut sähköpostin lähettäminen.

## 10.3 Ylläpidettävyys

Järjestelmä pyritään kehittämään siten, että ohjelmisto on mahdollisimman helposti ylläpidettävä. Tämä asettaa vaatimuksia dokumentaatiolle, jonka on oltava kattava, selkeä ja kuvaava. Dokumentaatio kuvaa ohjelmiston asennus-, konfigurointi- ja poistotavat. Dokumentaatiosta pitää tulla hyvin ilmi ohjelmiston sisäinen rakenne ja ulkoiset riippuvuudet. Järjestelmän ylläpito sisältää seuraavia toimia:

- Kielen lisääminen. Kieli lisätään tekemällä uusi konfiguraatitiedosto, johon käyttäjälle näytettävät merkkijonot lisätään halutulla kielellä. Kieli lisätään järjestelmään muuttamatta järjestelmän ohjelmakoodia.
- Vanhentuneiden aikataulujen ja ryhmän tietojen poisto. Järjestelmä poistaa automaattisesti vanhentuneet käyttäjätiedot, aikataulut ja ryhmät.

## 10.4 Virheistä toipuminen

Virhetilanteissa ohjelmiston on annettava kuvaava virheilmoitus. Virhetilanteet eivät saa aiheuttaa pysyvää vahinkoa järjestelmälle. Virhetilanteet eivät saa vaikuttaa järjestelmän tietokannan toimivuuteen. Virhetilanteita käsitellään luvussa Erikois- ja virhetilanteet.

## 11 Testaus

Testauksen tavoitteena on parantaa ohjelman toimivuutta ja käytettävyttä. Ohjelmisto testataan projektin testausvaiheessa. Testausvaihe alkaa toteutusvaiheen loppupuolella ja loppuu toteutusvaiheen päättymisen jälkeen. Testaus suoritetaan vaiheittain yksikkötestauksena, integrointitestauksena, validointitestauksena ja järjestelmätestauksena. Lisäksi järjestelmän komponentteja ja niiden yhteistoimintaa testataan sitä mukaan kun ne valmistuvat.

Testauksesta tehdään erillinen testaussuunnitelma projektin suunnitteluvaiheessa. Testauksen lopussa valmistuu testausdokumentti.

### 11.1 Yksikkötestaus

Yksikkötestaus suoritetaan järjestelmän jokaiselle komponentille. Testauksen tavoitteena on varmistaa komponenttien toimivuus suunnitteludokumentin määräämällä tavalla. Löydetyt virheet dokumentoidaan ja korjataan. Valtaosa testeistä tehdään testaamalla komponenttien toimivuutta dokumentoitujen rajapintojen kautta black-box-testauksena. Aikataulun salliessa voidaan tärkeimmille komponenteille suorittaa white-box-testausta.

### 11.2 Integroititestausta

Integroititestauksessa testataan komponenttien muodostamia ryhmiä siten, että painopiste on komponenttien rajapintojen ja yhteistoiminnan testauksessa. Testaus suoritetaan black-box-testauksena komponenttien rajapintojen perusteella. Tavoitteena on varmistua komponenttien yhteistoiminnan oikeellisuudesta ja dokumentoida havaitut virheet. Löytyneet virheet korjataan.

### 11.3 Validointitestausta

Validointitestauksessa tutkitaan, kuinka hyvin toteutettu järjestelmä vastaa määrittely- ja asiakasdokumenttien vaatimuksia. Ryhmä suorittaa validointitestausta. Löytyneet virheet dokumentoidaan ja pyritään korjaamaan, mikäli aikaa korjausten tekemiseen on.

### 11.4 Järjestelmätestausta

Järjestelmätestausta suoritetaan toteutusvaiheen jälkeen. Järjestelmätestauksessa testataan koko järjestelmä laitteistoinen. Testaus suoritetaan yhteistyössä asiakkaan kanssa. Vaatimuksena on mm. todeta järjestelmän ulkoinen toiminta, suorituskyky ja toimintavarmuus. Järjestelmätestauksen tarkoituksena on todeta, että järjestelmä suoriutuu kaikista sille asetetuista toiminnoista määrittelyn mukaisesti. Mikäli järjestelmätestauksessa löytyy virheitä tai puutteita, ne dokumentoidaan.