



AV-kioskin varausjärjestelmän vaatimusmäärittely

Samu Hynynen

Opinnäytetyö
Kesäkuu 2012
Tietojenkäsittely

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely

Samu Hynynen:
AV-kioskin varausjärjestelmän vaatimusmäärittely

Opinnäytetyö 36 sivua, josta liitteitä 23 sivua
Kesäkuu 2012

Tässä opinnäytetyössä käsitellään vaatimusten määrittelyn teoriaa yleisellä tasolla. Aihetta lähestytään myös käytännöllisemmästä näkökulmasta käyttäen tapausesimerkinä Tampereen ammattikorkeakoulun Taiteen, musiikin ja median yksikön audiovisuaalisen välinevaraston varausjärjestelmän vaatimusmäärittelyä.

Opinnäytetyön lopussa verrataan esitettyä teoriaa tapausesimerkissä sovellettuihin menetelmiin. Vertailun päätteeksi pohditaan, kuinka kattavasti vaatimusmäärittely dokumentoi juuri ne tarpeet ja odotukset, joita tilaajalla on valmiista tuotteesta. Pohdinnassa pyritään objektiiviseen tarkastelunäkökulmaan.

Opinnäytetyöhön liitetty vaatimusmäärittely johdattaa lukijansa sen tavoitteeseen ja tarkoitukseen sekä selittää käytettyä kohdealueen käsitteistöä ja teknistä termistöä lyhenteineen. Vaatimuksia havainnollistetaan käyttötapauksien ja -kaavioiden avulla. Vaatimusmäärittelyssä määritellään järjestelmälle asetetut toiminnalliset vaatimukset sekä reunaehdot, joissa tuote on voitava toteuttaa.

ABSTRACT

Tampere University of Applied Sciences
Bachelor's degree programme of
information and communications technology

Samu Hynynen:

Software Requirements Specification for Computerised Equipment Reservation System

Bachelor's thesis 36 pages, appendices 23 pages
June 2012

This thesis work deals with the theory of software requirements specification on a general level, in addition to which the issue is approached from a more practical point of view using an example of software requirements specification designed for the Art & Media programmes at Tampere University of Applied Sciences.

The thesis also compares the presented theory with the methods and practices used in the design of the case presented. At the end of the comparison, the thesis ponders how comprehensively the system requirements specification meet the needs and expectations the client has placed on the product.

The requirements specification attached to this thesis guides the reader to the terms used. The requirements are demonstrated by the means of use case reports and use case diagrams. The requirements specification defines the requirements set for the system and the boundaries it must follow.

Key words: software development, software design, programming, reservation system

SISÄLLYS

1	JOHDANTO.....	5
2	VAATIMUKSET JA NIIDEN MÄÄRITTELY	7
3	VARAUSJÄRJESTELMÄN VAATIMUSTEN MÄÄRITTELYSTÄ	9
4	POHDINTA.....	10
	LÄHTEET.....	12
	LIITTEET	13
	Liite 1. Vaatimusmäärittely varausjärjestelmälle	13

1 JOHDANTO

Suoritin työharjoitteluni Tampereen ammattikorkeakoulun Finlaysonin alueella sijaitsevan Taiteen, musiikin ja median yksikön audiovisuaalisen välinevaraston varausjärjestelmän jatkokehitystehtävissä. Kyseinen varausjärjestelmä on web-sovellus, jota käytetään varastoitavien laitteiden varaamiseen, lainaamiseen, palauttamiseen sekä inventointiin. Varausjärjestelmää käyttävät oppilaitoksen opiskelijat, opettajat sekä muu henkilökunta – pääasiassa kuitenkin välinevarastossa työskentelevät virkailijat.

Välinevarastossa työskentelevät virkailijat käyttävät varausjärjestelmää pääasiallisena työkalunaan kirjatessaan varaston asiakkaiden tekemiä lainoja, palautuksia sekä vikailmoituksia. Ennen jatkokehitysprojektia käytössä ollut varausjärjestelmä AVDB 1.0 oli otettu käyttöön vuonna 2005. Tuotantoympäristönä oli Tampereen Ammattikorkeakoulun tietokonekeskuksessa sijaitseva palvelin, jolle oli asennettuina PHP 4- ja MySQL 4-palvelinohjelmat. Jatkokehitysprojektin tavoitteena oli kehittää varausjärjestelmästä yhteensopiva uudempien PHP 5- sekä MySQL 5-palvelinohjelmaversioiden kanssa. Lähdekoodin analysointi sekä palaverit välinevaraston henkilökunnan kanssa toivat esille eräitä muitakin kehitystarpeita, jotka liittyivät pääasiassa sovelluksen käytettävyyteen ja tietoturvallisuuteen.

Jatkokehitysprojektin aikana ilmeni, että tuotannossa olleen järjestelmän kehittäminen ajantasaiseksi ja käyttäjäystävälliseksi työkaluksi varaston henkilökunnan tarpeisiin tulisi olemaan lähes yhtä työlästä ja aikaa vievää, kuin kokonaan uuden sovelluksen toteuttaminen. Jatkokehitysprojekti saatettiin loppuun tästä huolimatta. Tämän opinnäytetyön tavoitteena on laatia varausjärjestelmän vaatimusmäärittely Tampereen ammattikorkeakoulun Taiteen, musiikin ja median yksikön käyttöön. Työn tarkoituksena on laatia vaatimusmäärittely, joka on avuksi uutta järjestelmää hankittaessa taikka toteutettaessa. Tämä vaatimusmäärittely pyrkii siis muodostamaan uuden varausjärjestelmän hankintaan ja/tai kehitystyöhön osallistuvien sidosryhmien välille yhteisymmärryksen siitä, mitä uudelta järjestelmältä vaaditaan.

Vaatimusmäärittelyssä dokumentoidut toiminnalliset vaatimukset ovat muotoutuneet audiovisuaalisen välinevaraston henkilökunnan kanssa pidetyissä ryhmäpohjaisissa tapaamisissa. Koska minulla ei ollut aikaisempaa kokemusta vaatimusten määrittelystä, olen hyödyntänyt vapaamuotoisten palaverimuistiinpanojen lisäksi myös Julkisen

hallinnon tietohallinnon neuvottelukunnan ohjeistusta vaatimusten määrittelyyn (JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely). Poikkesin ohjeistuksesta jättämällä vaatimusten priorisoinnin pois, sillä kaikki laatimassani vaatimusmäärittelyssä esiintyvät vaatimukset ovat sovelluksen toiminnan kannalta tässä tapauksessa yhtä välttämättömiä. Sovellukselle asetetut laatuvaatimukset pohjautuvat suurimmilta osin omiin, jatkokehitysprojektin aikana kertyneisiin kokemuksiini, jotka syntyivät käytössä olevan varausjärjestelmän lähdekoodia ja järjestelmän toimintaa tarkasteltaessa. Tämän työn toisessa luvussa käsitellään vaatimusten määrittelyn teoriapohjaa sekä varausjärjestelmän vaatimusten määrittelyyn soveltamiani menetelmiä. Työn kolmannessa ja samalla viimeisessä luvussa peilataan työn tuloksia sen tavoitteiden ja esitetyn teorian rinnalla. Taiteen, musiikin ja median yksikölle toteutettu varausjärjestelmän vaatimusmäärittely on Liitteessä 1.

2 VAATIMUKSET JA NIIDEN MÄÄRITTELY

Tekniseltä toteutukseltaan moitteettominkaan tuote ei ole juuri tyhjää parempi, mikäli se ei täytä sille asetettuja vaatimuksia. Vaatimusten määrittelyn tavoitteena on selvittää ohjelmistolle asetettavat vaatimukset sellaisella tarkkuudella, että niiden perusteella voidaan kommunikoida eri osapuolille, millainen ohjelmiston halutaan olevan (JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely). Vaatimusmäärittelyn tarkoituksena on löytää ja dokumentoida ne tarpeet, jotka lopullisen tuotteen on täytettävä, jotta sen kehittäjät ja kehittämistyöhön osallistuvat sidosryhmät saavat mahdollisimman tarkan, kokonaisvaltaisen ja yhtenevän käsityksen halutusta lopputuloksesta. Vaatimusmäärittely on avuksi myös kehitystyöhön kuuluvien resurssien ja aikataulujen suunnittelussa (IEEE Std 830-1998).

Ohjelmiston suunnittelussa vaatimusmäärittelyä edeltävää vaihetta kutsutaan esiselvitykseksi. Esiselvityksessä kartoitetaan ohjelmiston toimintaympäristöä, tietoturvaseikkoja sekä kuvataan kehitystyön tavoitteita. Jotta kehitystyön tavoitteita voidaan kuvata, kehittämiskohteiden on luonnollisestikin oltava tiedossa (Hughes & Cotterell, 2006).

Kehittämiskohteet voidaan kartoittaa jo olemassa olevan järjestelmän puutteiden taikka laajentuneiden käyttötarpeiden pohjalta. Uusia käyttötarpeita voivat synnyttää muun muassa kasvava käyttäjämäärä, toteutustavan taikka sovellettujen standardien vanheneminen, muutokset kokonaisarkkitehtuurissa ja uusien tai laajennettujen ominaisuuksien tarve.

Kehittämiskohteiden lisäksi myös vaatimusmäärittelyn kohdesidosryhmien on oltava tiedossa, jotta vaatimusmäärittely voidaan kirjoittaa kaikkia osapuolia hyödyttävällä tavalla. Sidosryhmiä voivat olla esimerkiksi tuotteen toteutuksesta vastaava taho, tuotteen tilaaja ja tuotteen kohderyhmän edustajat. Hyvä vaatimusmäärittely ottaa huomioon kaikkien sidosryhmien tuotteelle asettamat vaatimukset.

Vaatimuksia voidaan havainnollistaa esimerkiksi käyttötapauskertomusten ja –kaavioiden avulla. Käyttötapauskertomusten ja –kaavioiden avulla myös järjestelmän ydintoiminta hahmottuu lukijalle (Leffingwell & Widrig, 2003).

Vaatimukset on tapana priorisoida niiden liiketoiminta-arvon perusteella, jotta ohjelmiston ydinosat saadaan toteutettua ennen niistä riippuvaisia lisäominaisuuksia, ja näin tuotteesta on esiteltävissä konseptitason prototyyppejä, joita voidaan verrata tuotteen sidosryhmien tarpeisiin jo ennen lopullista julkaisua.

Mikäli vaatimuksia ei määritellä välittömästi esiselvityksen jälkeen, kehittämiskohteisiin ja esiselvitykseen on mahdollisesti syytä luoda vielä silmäys ennen toimeen ryhtymistä, sillä niihin on voinut ilmetä päivitystarpeita.

Hyvä, yksiselitteinen vaatimus esitetään siten, että siihen ei sisälly muita vaatimuksia. Vaatimuksen on kuitenkin sisällettävä kaikki sen onnistuneeseen täyttämiseen tarvittava tieto. Toiminnalliset vaatimukset määrittelevät, millaisiin toimintoihin järjestelmän odotetaan kykenevän. Toiminnallinen vaatimus voi olla esimerkiksi seuraavanlainen: ”Käyttäjän on voitava vaihtaa salasanaansa”. Toiminnallisten vaatimusten lisäksi vaatimusmäärittelyssä määritellään myös ohjelmistolle asetetut reunaehdot, eli ei-toiminnalliset vaatimukset, joita ovat esimerkiksi ylläpidettävyys-, käytettävyys-, turvallisuus-, siirrettävyys- ja yhteensopivuusseikat. Vaatimusmäärittelyssä kuvataan tuotteelle asetettujen vaatimusten ohella myös sille asetettuja tavoitteita sekä keinoja, joilla asetetut tavoitteet voidaan täyttää onnistuneesti.

3 VARAUSJÄRJESTELMÄN VAATIMUSTEN MÄÄRITTELYSTÄ

Varausjärjestelmän vaatimusmäärittelyssä esiintyvät vaatimukset on laadittu yhteistyössä tilaajan ja tuotteen kohderyhmän edustajien kanssa. He perehdyttivät minut käytössä olevan varausjärjestelmän eri ominaisuuksiin ja toimintoihin työharjoitteluna suorittamani jatkokehitysprojektin aikana. Jatkokehitysprojektin aikana toteutimme opiskelijakollegani kanssa useita käytettävyy- ja tietoturvaparannuksia käytössä olevaan varausjärjestelmään.

Jatkokehitys osoittautui erittäin hankalaksi, koska sovelluksen ohjelmakoodia ei oltu toteutettu hyvien tapojen ja suositusten mukaisesti eikä sitä myöskään oltu dokumentoitu asianmukaisesti. Edellä mainittujen puutteiden johdosta jouduimme käymään lävitse koko sovelluksen lähdekoodin ja analysoimaan sen toimintaa. Jatkokehitysprojekti saatettiin loppuun, mutta ohjelmistoon jäi tämänkin jälkeen puutteita, joiden korjaaminen ei sovelluksen toteutustavasta johtuen ollut mahdollista. Tämän johdosta uuden varausjärjestelmän suunnitteleminen ja toteuttaminen osoittautui mielekkäämmäksi ja taloudellisemmaksi vaihtoehdoksi, kuin jo käytössä olevan järjestelmän jatkokehittäminen.

Järjestelmän kehityskohteet ilmenivät jatkokehitysprojektin aikana. Ilmoitin tilaajalle kehitystarpeista ja kerroin, että käytössä olevan järjestelmän jatkokehitystyö ei ole mahdollista annettujen resurssien puitteissa. Tätä seurasi tilaajalta saatu toimeksianto kokonaan uuden varausjärjestelmän vaatimusten määrittelemiseksi.

Muotoilin järjestelmälle asetetut vaatimukset nykyiseen muotoonsa ryhmäpohjaisten neuvonpitojen pohjalta. Neuvonpidoissa paneuduttiin järjestelmässä ilmenneisiin ongelmiin. Toin ongelmakohdat tilaajan tietoon ja esitin oman näkökulmani mahdollisista ratkaisumalleista. Saavutettuamme tilaajan kanssa yhteisymmärryksen uuden tuotteen kehitykseen sovellettavista menetelmistä, ryhdyin määrittelemään vaatimuksia.

4 POHDINTA

Tässä luvussa vertaan työn lopputuloksena syntynyttä tuotosta aiemmassa luvussa esittämäni vaatimusmäärittelyn teoriaan sekä johdannossa esitettyihin opinnäytetyön tavoitteisiin ja pyrin tarkastelemaan työtä niin kriittisessä ja objektiivisessä valossa, kuin asemassani on mahdollista.

Kenties olennaisin seikka vaatimusmäärittelyn oikeellisuutta arvioitaessa on vaatimusten paikkansapitävyys ja yksiselitteisyys: järjestelmälle asetetut toiminnalliset vaatimukset on dokumentoitu kieliasultaan riittävän selkeästi sekä yksiselitteisesti ja ennen kaikkea ne kuvaavat mitä suurimmalla todennäköisyydellä juuri niitä toiminnallisuuksia, joita toimeksiantaja lopputuloksena syntyvältä tuotteelta odottaa. Tässä suhteessa vaatimusmäärittelyn voidaan katsoa onnistuneen tavoitteessaan varsin mainiosti.

Vaatimusmäärittelyyn ei ole kerätty erikseen sovelluksen toiminnan kannalta välttämättömiä ydinvaatimuksia ja lähinnä käytettävyyseikkoihin kantaa ottavia käyttäjävaatimuksia. Työssä listattuja vaatimuksia ei ole myöskään priorisoitu, koska niissä määritellyt toiminnallisuudet ovat suurilta osin toisistaan riippuvaisia ja sovelluksen toiminnan kannalta yhtä välttämättömiä. Prioriteeteiltaan tasa-arvoisten vaatimusten pohjalta voi olla vaikeata kehittää konseptitason prototyyppiä esiteltäväksi tilaajalle ennen lopullista julkaisuversiota. Esittelykelpoisen prototyypin kehittäminen olisi erinomainen tapa varmistua siitä, että tuotteen kehitys edistyy oikeaan suuntaan.

Sisällöllisestä samankaltaisuudesta johtuen jokaisesta vaatimuksesta ei ole laadittu käyttötapauskertomusta tai -kaaviota. Tämän johdosta vaatimusmäärittely ei ole aivan niin kattava, kuin se mahdollisesti voisi olla. Kertomukset ja kaaviot välittävät mielestäni kuitenkin riittävän tarkan kuvan järjestelmästä toivotusta toiminnasta, jotta järjestelmän kehitystyöhön osallistuvat henkilöt voivat toteuttaa niiden pohjalta tilaajan tarpeet täyttävän ratkaisun.

Sovellukselle esitetyt suunnittelurajoitteet ja suositellut standardit pyrkivät varmistamaan, että järjestelmän mahdolliset jatkokehittäjät eivät joudu aloittamaan työtänsä samoista lähtökohdista, joista käytössä olevan varausjärjestelmän jatkokehitysprojektiin osallistunut organisaatio työnsä aloitti

Vaatusmääritys soveltuu parhaiten uuden järjestelmän toteuttamiseen alusta alkaen. Kustannusten pitämiseksi alhaisina, työ on suositeltavaa toteuttaa opiskelijaprojektina taikka esimerkiksi työharjoitteluna. Tampereen ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opiskelijoiden joukosta löytyy mitä suurimmalla todennäköisyydellä riittävää osaamista vaatimusmäärityksen mukaisen varausjärjestelmän toteuttamiseen.

LÄHTEET

Hughes, B. & Cotterell, M. 2006. Software Project Management (4th ed.). McGraw-Hill Higher Education.

ICT-Palvelujen kehittäminen: Vaatimusmäärittely. Osa ICT-palvelujen kehittäminen -suositussarjaa. 2009 . Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA). Luettu 19.2.2012. <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.html>

IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998, Revision of IEEE Std 830-1993). Luettu 19.2.2012. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574>

Leffingwell, D. & Widrig D. 2003. Managing Software Requirements: A Use Case Approach (2nd ed.). Addison-Wesley.

LIITTEET

Liite 1. Vaatimusmäärittely varausjärjestelmälle

VAATIMUSMÄÄRITTELY **VARAUSJÄRJESTELMÄLLE**

Laatinut: Samu Hynynen

Sisällysluettelo

1 Johdanto.....	3
1.1 Tavoite ja tarkoitus.....	3
1.2 Tuote ja ympäristö.....	3
1.3 Kehitettävään sovellukseen liittyvä termistö.....	4
2 Yleiskuvaus.....	6
2.1 Ympäristö.....	6
2.2 Toiminta.....	6
2.3 Käyttäjät.....	7
2.4 Yleiset rajoitteet.....	7
2.5 Oletukset ja riippuvuudet.....	7
3 Tiedot ja tietokanta.....	7
3.1 Tietosisältö.....	8
3.2 Kapasiteettivaatimukset.....	8
3.3 Tiedostot ja asetustiedostot.....	8
4 Toiminnalliset vaatimukset.....	9
4.1 Järjestelmän toiminta.....	9
4.2 Lainaaminen ja palauttaminen.....	11
4.3 Varaaminen.....	14
4.4 Laitetietojen käsittely.....	16
4.5 Lomakkeiden käsittely.....	18
5 Ulkoiset liittymät.....	20
5.1 Laitteistoliittymät.....	20
5.2 Ohjelmistoliittymät.....	21
6 Muut ominaisuudet.....	21
6.1 Käytettävyys, turvallisuus.....	21
6.2 Ylläpidettävyys.....	21
6.3 Yhteensopivuus ja siirrettävyys.....	21
7 Suunnittelurajoitteet.....	22
7.1 Standardit ja suositukset.....	22
7.2 Laitteistorajoitteet.....	22
7.3 Ohjelmistorajoitteet.....	22
8 Jatkokehitys.....	23

1 Johdanto

Tämä luku alalukuineen johdattaa lukijan vaatimusmäärittelyn tavoitteeseen, tarkoitukseen, suunnitellun tuotteen luonteeseen, sen tuotanto- ja käyttöympäristöön sekä selittää dokumentissa käytettyjä lyhenteitä ja termejä.

1.1 Tavoite ja tarkoitus

Tämän vaatimusmäärittelyn tavoite on määritellä ne vaatimukset, joita Tampereen ammattikorkeakoulun taiteen, median ja musiikin yksikön audiovisuaalisen varaston varausjärjestelmältä edellytetään. Tarkoituksena on saada aikaiseksi uusi, jo käytössä olevan, tietoturvaltaan ja käytettävyydeltään puutteellisen varausjärjestelmän korvaava sovellus.

1.2 Tuote ja ympäristö

Varausjärjestelmä on sovellus, jota käytetään tiedon tallettamiseen ja hakemiseen. Tuote on tarkoitettu Tampereen ammattikorkeakoulun taiteen, musiikin ja median yksikön audiovisuaalisen välinevaraston, AV-kioskin käyttöön. AV-kioski on varasto, josta oppilaitoksen opiskelijat sekä opettajat voivat lainata kalustoa opetuskäyttöä ja projekteja varten. Tässä tapauksessa varausjärjestelmän avulla tallennetaan ja haetaan siis tietoa varastossa ja lainassa olevista laitteista. Tätä suunnitteludokumenttia kirjoitettaessa AV-kioskin käytössä on vuonna 2005 käyttöön otettu varausjärjestelmä, AVDB. AVDB:n tuotantoympäristönä on Tampereen ammattikorkeakoulun tietokonekeskuksessa sijaitseva palvelintietokone, jonka ylläpidosta vastaa tietokonekeskuksen henkilökunta. AVDB:n käyttöympäristö on web-selain, jolla järjestelmään kirjaudutaan. Kustannusten minimoimiseksi tuote on tarkoitettu toteuttamaan siten, että se voidaan implementoida toimimaan samassa tuotanto- ja käyttöympäristössä kuin AVDB.

1.3 Kehitettävään sovellukseen liittyvä termistö

Tässä luvussa listataan kehitettävään sovellukseen liittyvä termistö selityksineen. Ensimmäinen taulukko sisältää kohdealueen käsitteet selityksineen, toisessa ovat listattuina tekniset termit ja muut kehitettävään sovellukseen liittyvät lyhenteet.

Käsite	Selitys
Käyttäjä	Henkilö, jolla on oikeus varata ja lainata laitteita AV-kioskista. Käyttäjät vievät lainaan tahtomansa laitteet AV-kioskin palvelutiskille, jossa ylläpitäjä kirjaa lainauksen varausjärjestelmään.
Lainaus	Laitteen ottaminen lainaan AV-kioskista. Lainaajat ovat laitekohtaisia.
Lainauskielto	Käyttäjä voidaan asettaa lainauskieltoon, mikäli tämä ei ole palauttanut lainaamiaan laitteita eräpäivään mennessä. Lainauskiellossa oleva käyttäjä ei voi lainata laitteita AV-kioskista.
Palautus	Lainatun laitteen palauttaminen AV-kioskiin. Käyttäjä tuo lainaamansa laitteen AV-Kioskiin, jonka palvelutiskillä päivystävä ylläpitäjä kirjaa laitteen palautetuksi.
Varaus	Laite voidaan varata tietylle ajanjaksolle. Varattu laite voidaan lainata ainoastaan käyttäjälle, joka varauksen on tehnyt.
Ylläpitäjä	Henkilö, joka työskentelee AV-kioskissa ja ylläpitää varausjärjestelmää. Ylläpitäjä kirjaa käyttäjien tekemät lainat ja palautukset järjestelmään.

Termi / Lyhenne	Selitys
AVDB	AV-Kioskin vuonna 2005 käyttöön ottama varausjärjestelmänä toimiva web-sovellus. AVDB on ohjelmoitu PHP:lla ja se käyttää MySQL-tietokantaa.
CSS	Kieli, jolla määritellään tyyliohjeet WWW-dokumenteille. Käytetään usein varsinkin XHTML-dokumenttien tyylien määrittelyyn.
CSV-tiedosto	Tekstitiedosto, joka sisältää pilkuilla toisistaan erotettuja arvoja.
HTML	Hypertekstuaalinen merkintäkieli, jolla kuvataan www-dokumentin rakennetta.
LAMP	Lyhennettä käytetään kuvaamaan erittäin yleistä web-palvelinteknologian kokoonpanoa, jossa palvelinkoneen käyttöjärjestelmänä on Linux, palvelinohjelmistona Apache, tietokantaohjelmistona on MySQL ja dynaamisuuden ja vuorovaikutteisuuden mahdollistavana palvelinpuolen skriptikielenä tyypillisimmin PHP (http://linux.fi/wiki/LAMP).
MySQL	MySQL on maailman käytetyin relaatiotietokantaohjelmisto. Useimmat tietokantapohjaiset web-palvelut käyttävät MySQL-tietokantaa.
PHP	Eräs suosituimmista dynaamisten web-palveluiden toteuttamiseen käytetyistä kielistä. PHP sisältää kattavan luokkakirjaston ja sen uusimmat versiot mahdollistavat oliopohjaiset ohjelmointiratkaisut.
TAMK	Tampereen ammattikorkeakoulu.

XHTML

HTML:n korvaajaksi suunniteltu merkintäkieli. XHTML on muotosäännöiltään HTML:ää tiukempaa ja sen oikeaoppinen käyttö takaa hyvän selainyhteensopivuuden.

2 Yleiskuvaus

Tässä luvussa kuvataan yleisellä tasolla sovelluksen toimintaa, sen ympäristöä, käyttäjiä, sille asetettuja rajoitteita ja oletuksia sekä riippuvuuksia.

2.1 Ympäristö

Varausjärjestelmän tuotantoympäristö on TAMKin tietokonekeskuksen tiloissa sijaitseva LAMP-tyyppinen palvelintietokone. Komentosarjakielenä se käyttää PHP:n viidettä versiota ja tietokantana MySQL:n viidettä versiota. Järjestelmän käyttöympäristö on web-selain, jolla käyttäjä lähettää tuotantopalvelimelle pyyntöjä.

2.2 Toiminta

Tuotantopalvelimella toimiva varausjärjestelmä käsittelee käyttäjän selaimelta vastaanottamansa pyynnön, jonka pohjalta se luo dynaamisesti www-dokumentin, joka lähetetään käyttäjän selaimelle vastauksena tältä saatuun pyyntöön. Ohjelmakoodia tulkaava palvelinohjelma tekee käyttäjän pyyntöjen perusteella hakuja tietokantaan, jonka vaste käsitellään ohjelmakoodin toimintalogiikan mukaisesti ja tulostetaan osaksi käyttäjälle lähetettävää XHTML-dokumenttia. Ohjelmakoodin toimintalogiikka määräytyy tässä suunnitteludokumentissa määriteltyjen vaatimusten mukaisesti.

2.3 Käyttäjät

Sovelluksen käyttäjäkunta koostuu oppilaitoksen opiskelijoista, opettajista sekä muusta henkilökunnasta. Käyttöoikeuksiltaan käyttäjät jakautuvat peruskäyttäjiin ja ylläpitäjiin. Ylläpitäjät huolehtivat, että varausjärjestelmän varastotietokanta vastaa tietosisällöltään AV-kioskin varaston tilaa. Peruskäyttäjät eivät voi tehdä lainauksia taikka palautuksia omin päin. Kaikki lainaukset ja palautukset tehdään AV-kioskin palvelutiskillä, jonka työasemalla päivystävä ylläpitäjä kirjaa ne varausjärjestelmään. Peruskäyttäjät voivat hakea laitteita tietokannasta, tarkastella niiden tietoja, tarkastella niistä tehtyjä varauksia sekä halutessaan varata laitteita käyttöönsä.

2.4 Yleiset rajoitteet

Järjestelmän tulee olla riittävän tietoturvallinen, jotta henkilöt, joilla ei ole järjestelmään käyttöoikeuksia (käyttäjätunnusta ja salasanaa), eivät pääse käsiksi mihinkään tietoihin tai toimintoihin. Peruskäyttäjät eivät saa päästä käsiksi ainoastaan ylläpitäjille tarkoitettuihin tietoihin ja toimintoihin.

2.5 Oletukset ja riippuvuudet

Koska suunniteltu sovellus on tarkoitettu AVDB:n korvaavaksi järjestelmäksi, sen on tarkoitus toimia samassa käyttö- ja tuotantoympäristössä kuin edeltäjänsäkin, jotta käyttäjät ja ylläpitäjät tottuvat siihen mahdollisimman ongelmattomasti ja nopeasti. Lisäksi sovellus tulee käyttämään samaa, jo olemassa olevaa tietokantaa.

3 Tiedot ja tietokanta

Tämä luku käsittelee sovelluksen hakemistorakennetta, tietokannan rakennetta ja sen sisältöä.

3.1 Tietosisältö

Varausjärjestelmä käyttää samaa laite- ja käyttäjätietokantaa kuin AVDB. Kaikki sovelluksen käyttämät tiedot sijaitsevat samassa tietokannassa.

3.2 Kapasiteettivaatimukset

Tuotantopalvelimen nykykapasiteetti täyttää varausjärjestelmälle asetetut kapasiteettivaatimukset oivallisesti. Käyttäjiä on satoja, mutta yhdenaikainen käyttö on vähäistä, sillä järjestelmää käytetään enimmäkseen AV-kioskin palvelutiskin työasemalta käsin. Vaikka kaikki käyttäjät kirjautuisivat järjestelmään samanaikaisesti tekemään tietokantahakuja, tuotantopalvelin suoriutuisi käyttökuormasta ongelmitta. Järjestelmän käyttöintensiteetti on joitakin kymmeniä käyttökertoja päivässä.

Name	Size	Type	Date Modified
class		File Folder	23.02.2012 17:02
conf		File Folder	14.02.2012 12:53
css		File Folder	23.02.2012 17:10
CVS		File Folder	14.02.2012 14:23
data		File Folder	27.01.2012 13:18
image		File Folder	09.02.2012 11:47
lib		File Folder	23.02.2012 17:02
index.php	2 KB	PHP Script	23.02.2012 13:55

Kuva 1: Sovelluksen mahdollinen hakemistorakenne. Näkymä hakemistosta, joka sisältää kaikki sovelluksen käyttämät tiedostot.

3.3 Tiedostot ja asetustiedostot

Kaikki sovelluksen käyttämät tiedostot ynnä niiden dokumentaatio tallennetaan tuotantopalvelimelle omiin hakemistoihinsa saman päähakemiston alle. Asetustiedostot sijoitetaan hyvän tavan mukaisesti omaan, havainnollisesti nimettyyn hakemistoonsa (esimerkiksi "conf/"). Asetustiedostot sisältävät tietokantapalvelimen käyttäjätunnuksen ja salasanan, joten

hakemiston tietoturvaan on syytä kiinnittää huomiota. Sovelluksen käyttämät tiedostot voidaan sijoittaa hakemistoon kuvassa 1 annetun esimerkin mukaisesti.

4 Toiminnalliset vaatimukset

Tässä luvussa kerrotaan sovellukselle asetetuista toiminnallisista vaatimuksista. Vaatimukset tunnusnumeroineen ovat listattuina taulukoihin. Tunnusnumeron tarkoitus ei ole kuvata minkään yksittäisen vaatimuksen prioriteettia suhteessa muihin vaatimuksiin, vaan ainoastaan helpottaa kehittäjien ja muiden kehitystyöhön osallistuvien sidosryhmien keskinäistä kommunikointia tiettyyn vaatimukseen viitattaessa. Jokaisessa alaluvussa vaatimuksia havainnollistetaan niihin liittyvien käyttötapaukset ja -kaavioiden avulla. Peruskäyttäjään viitataan käyttäjänä.

4.1 Järjestelmän toiminta

Tunnusnumero	Vaatus
1.1	Järjestelmään on voitava kirjautua oppilaitoksen opiskelija- tai henkilökuntatunnuksilla.
1.2	Järjestelmän on tunnistettava ylläpitäjät peruskäyttäjistä kirjautumistietojen perusteella.
1.3	Järjestelmän on tulostettava ylläpitäjälle ylläpito-ominaisuudet käsittävä näkymä (ylläpitosivu).
1.4	Järjestelmän on tulostettava käyttäjälle varauksien tekemiseen sekä omien lainojen ja varausten tarkasteluun tarkoitettu näkymä (käyttäjäsivu).

Käyttötapaus 1

Ylläpitäjä kirjautuu järjestelmään.

Vaatus

1.3 Järjestelmän on tulostettava ylläpitäjälle ylläpito-ominaisuudet käsittävä näkymä (ylläpitosivu).

Toimijat

Ylläpitäjä.

Alkutilanne

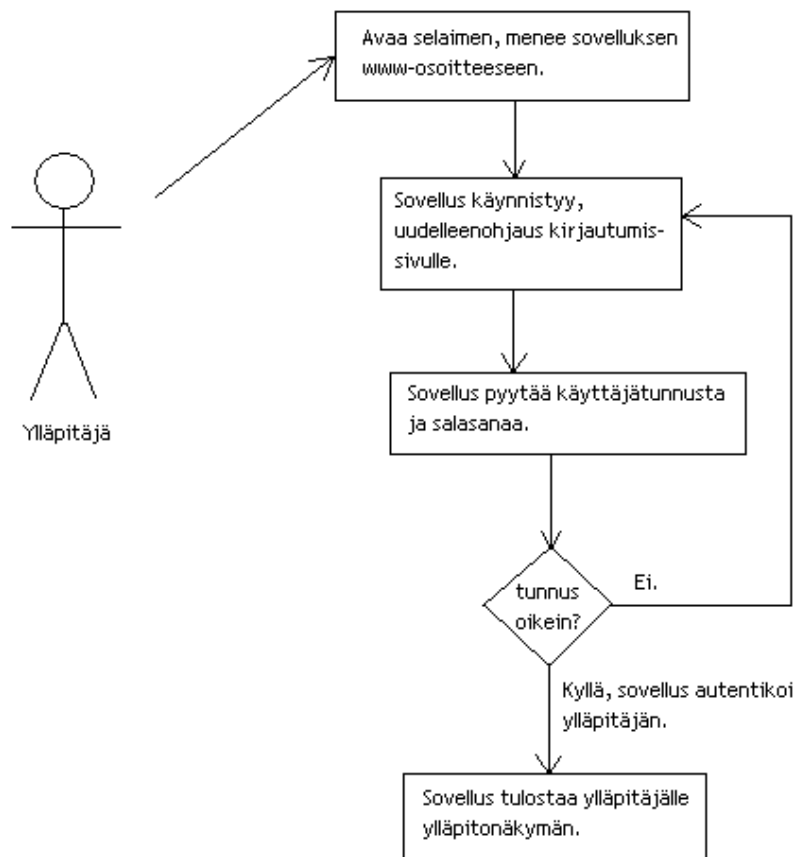
Ylläpitäjä haluaa kirjautua järjestelmään.

Lopputilanne

Ylläpitäjä on kirjautunut järjestelmään.

Kuvaus

Ylläpitäjä avaa työasemansa web-selaimen. Hän siirtyy varausjärjestelmän verkko-osoitteeseen. Sovellus uudelleenohjaa ylläpitäjän selaimen TAMKin intranetin kirjautumissivulle. Ylläpitäjä syöttää kirjautumissivun tekstikenttiin henkilökuntatunnuksensa ja salasanasensa. Sovellus käsittelee kirjautumistiedot, jotka tunnistettuaan tämä tulostaa ylläpitäjälle sivun, josta hän pääsee nopeasti käsiksi kaikkiin järjestelmän toimintoihin.



Kuva 2: Ylläpitäjä kirjautuu sovellukseen.

4.2 Lainaaminen ja palauttaminen

Tunnusnumero	Vaatus
2.1	Ylläpitäjän on voitava lisätä yksittäinen laite käyttäjän lainoihin.
2.2	Ylläpitäjän on voitava lisätä CSV-tiedoston sisältämät laitetunnukset käyttäjän lainoihin.
2.3	Ylläpitäjän on voitava poistaa laite käyttäjän lainoista.
2.4	Ylläpitäjän on voitava tarkastella kaikkia lainauksia.
2.5	Ylläpitäjän on voitava muuttaa lainauksen tietoja.

Käyttötapaus 2

Käyttäjä lainaa kaiutinprosessorin ja kaiuttimen AV-kioskista.

Vaatus

2.2 Ylläpitäjän on voitava lisätä CSV-tiedoston sisältämät laitetunnukset käyttäjän lainoihin.

Toimijat

Käyttäjä, ylläpitäjä.

Alkutilanne

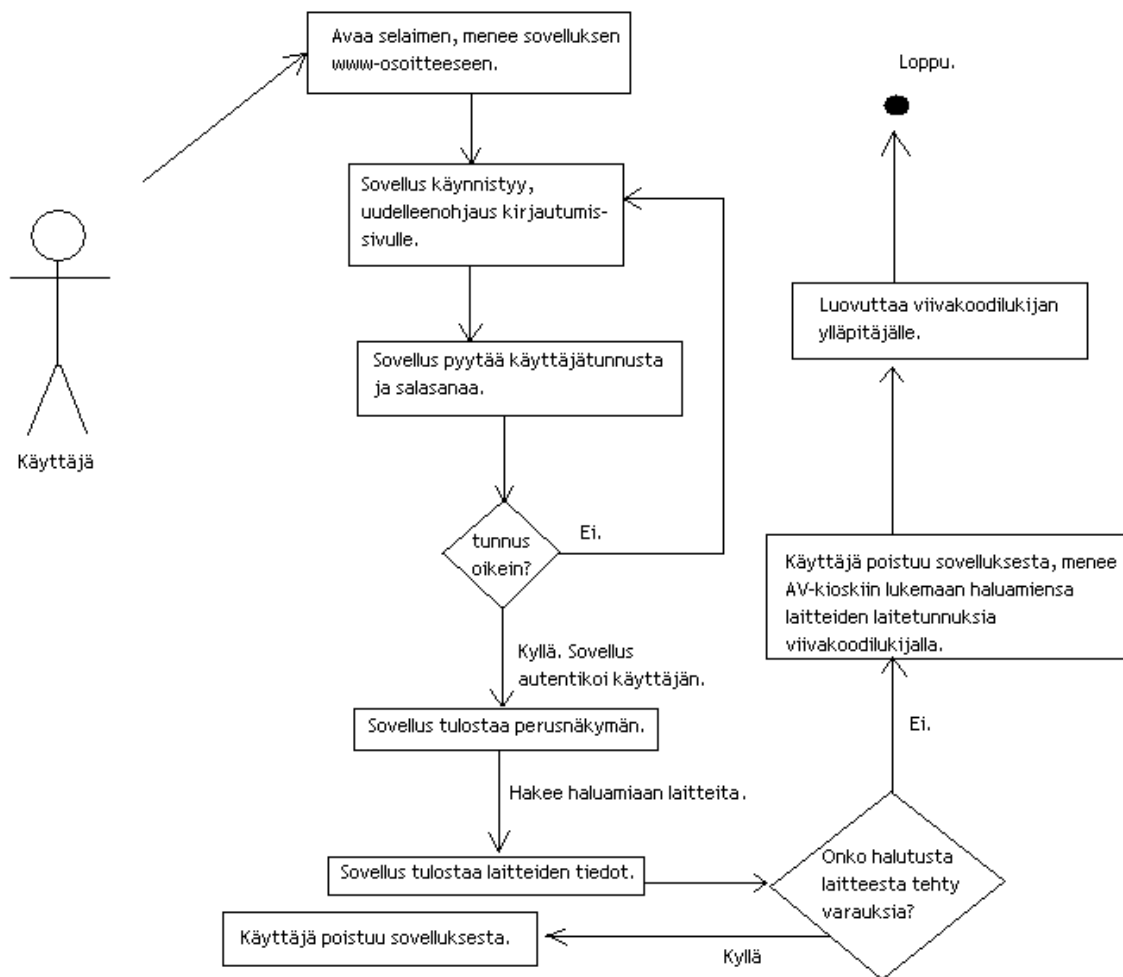
AV-kioskin varastossa on kaiutinprosessori ja kaiutin, joista ei ole tehty varauksia. Käyttäjä haluaa lainata kaiutinprosessorin ja kaiuttimen.

Lopputilanne

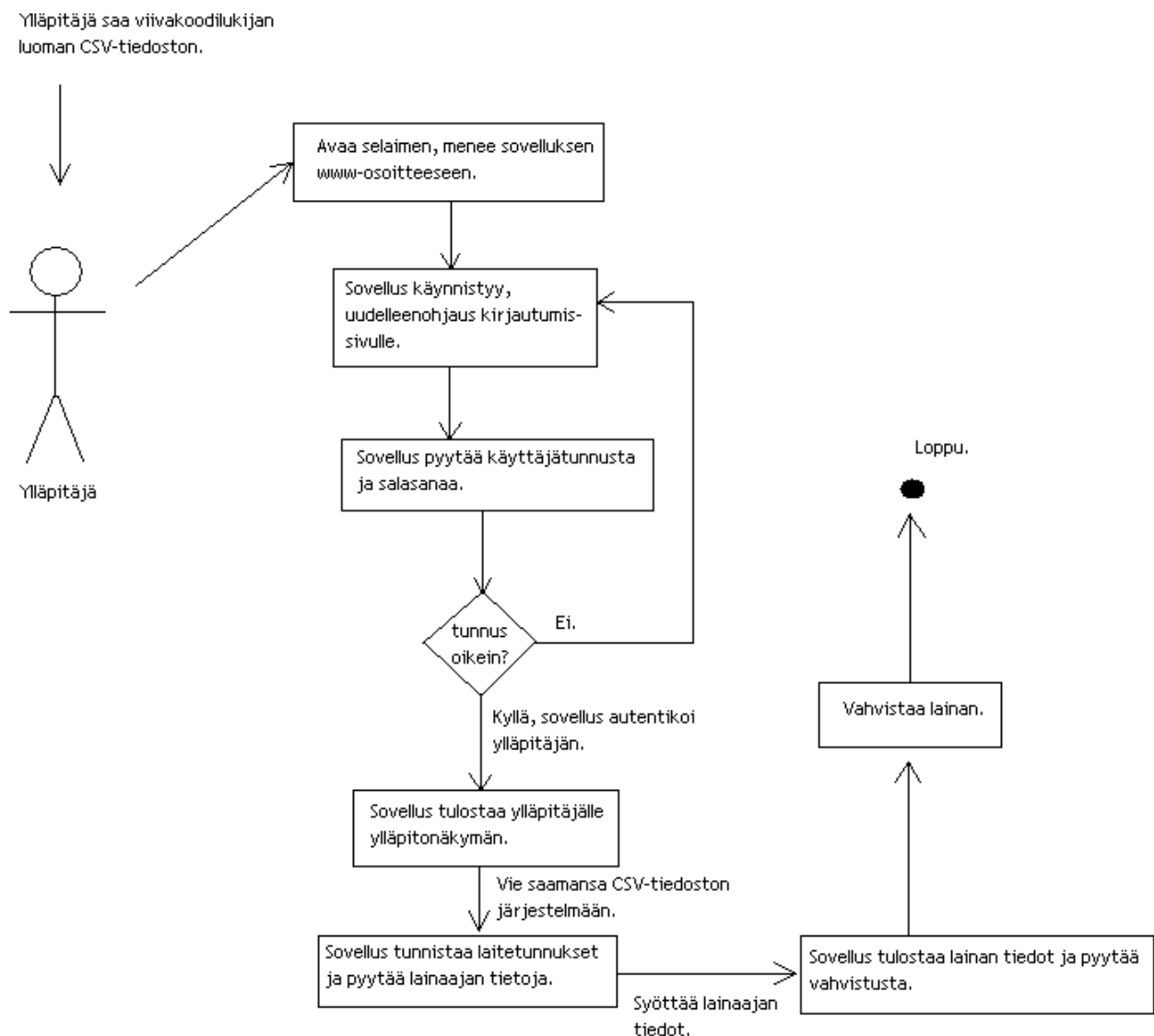
Kaiutinprosessori ja kaiutin on merkitty varausjärjestelmässä käyttäjän lainaamiksi ja laitteet ovat käyttäjän hallussa.

Kuvaus

Käyttäjä haluaa lainata kaiutinprosessorin ja kaiuttimen. Hän kirjautuu www-selaimellaan varausjärjestelmään tarkastaakseen, ovatko laitteet saatavilla, ovatko ne lainassa tai onko niistä tehty varauksia. Käyttäjä tekee haut laitteiden nimillä. Laitteet ovat lainattavissa. Käyttäjä asioi AV-kioskissa. Hän löytää varaston hyllystä etsimänsä kaiutinprosessorin ja kaiuttimen, lukee niiden laitetunnukset viivakoodilukijalla. Tämän jälkeen käyttäjä toimittaa viivakoodilukijan palvelutiskille, jossa päivystävä ylläpitäjä liittää sen työasemaansa, jonka selaimella hän vie viivakoodilukijan luoman CSV-tiedoston varausjärjestelmään. Varausjärjestelmä tunnistaa laitekoodit ja tulostaa lomakekentän, johon ylläpitäjä voi syöttää lainaajan käyttäjätunnuksen. Tämän jälkeen tulostuu ruutu, joka esittää tietoja lainatuista laitteista, lainaajasta sekä laina-ajoista. Ylläpitäjä vahvistaa lainan. Käyttäjä poistuu AV-kioskin tiloista lainaamiensa laitteiden kanssa.



Kuva 3: Käyttäjä lainaa AV-kioskista haluamansa laitteet. Osa 1/2.



Kuva 4: Käyttäjä lainaa AV-kioskista haluamansa laitteet. Osa 2/2.

4.3 Varaaminen

Tunnusnumero	Vaatus
3.1	Käyttäjän on voitava hakea laitetta nimen perusteella.
3.2	Käyttäjän on voitava varata laite haluamalleen ajanjaksolle.
3.3	Käyttäjän on voitava tarkastella laitteesta tehtyjä varauksia.
3.4	Käyttäjän on voitava poistaa tekemänsä varaus.
3.5	Ylläpitäjän on voitava poistaa laitteesta tehty varaus.

Käyttötapaus 3

Käyttäjä haluaa varata kaiutinprosessorin tulevan viikon ajaksi.

Vaatus

3.2 Käyttäjän on voitava varata laite haluamalleen ajanjaksolle.

Toimijat

Käyttäjä.

Alkutilanne

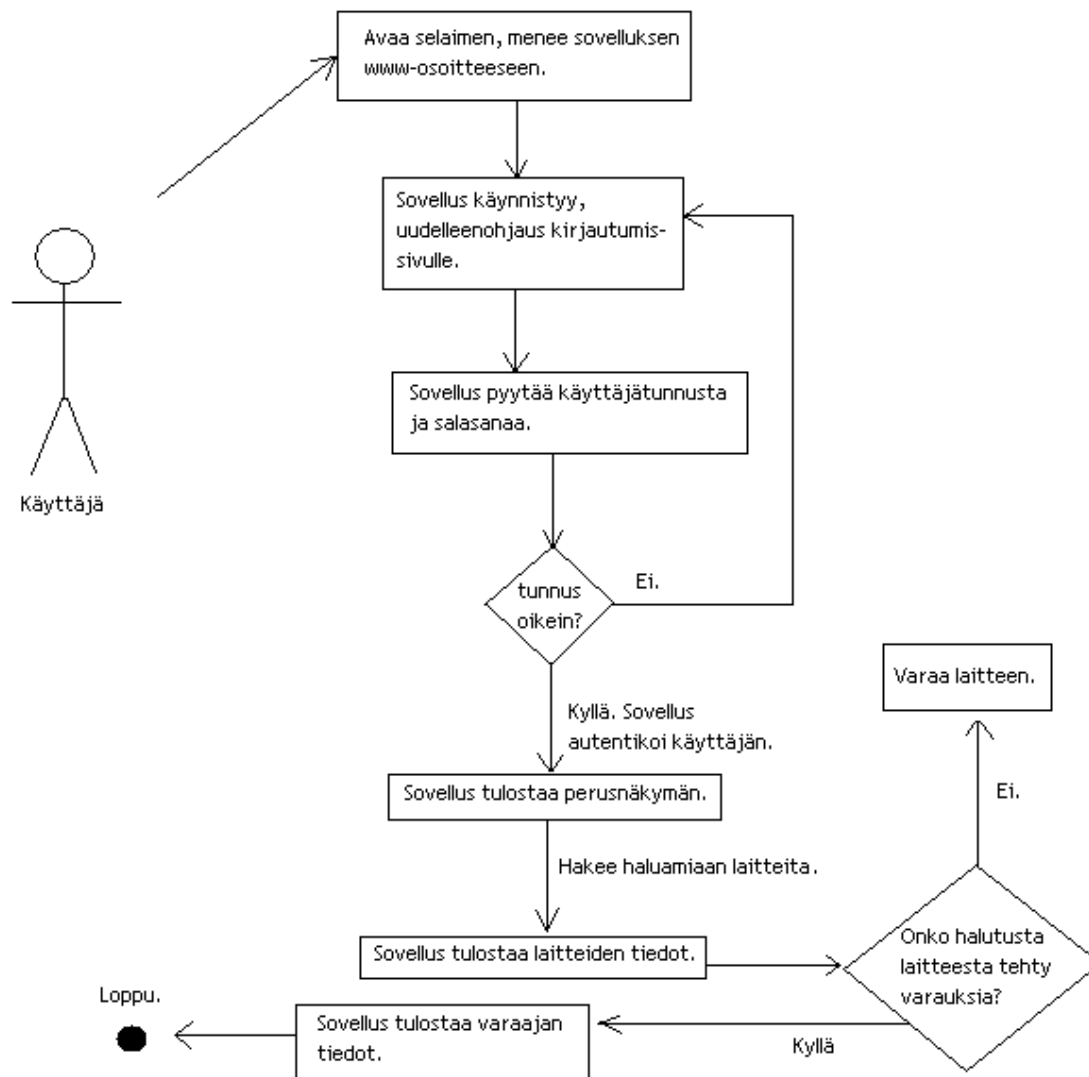
Käyttäjä haluaa varata kaiutinprosessorin tulevan viikon ajaksi. Kaiutinprosessorista on tehty varaus nimenomaiselle ajankohdalle.

Lopputilanne

Käyttäjän ei onnistunut varata laitetta, koska se oli jo varattu. Hänellä on kuitenkin tieto siitä, kenelle laite on varattu.

Kuvaus

Käyttäjä haluaa varata kaiutinprosessorin tulevan viikon ajaksi. Käyttäjä kirjautuu varausjärjestelmään opiskelijatunnuksillaan web-selainta käyttäen. Kirjautuminen onnistuu ja sovellus tulostaa käyttäjälle sivun, josta hän voi hakea AV-kioskin laitteita eri hakuperustein. Käyttäjä hakee laitetta nimellä "kaiutinprosessori". Sovellus tulostaa kalenterinäkymän, josta käy ilmi, että kaiutinprosessori on jo varattu tulevan viikon ajaksi. Sovellus kertoo myös, kenelle laite on varattu. Käyttäjä on yhteydessä laitteen varanneeseen henkilöön.



Kuva 5: Käyttäjä haluaa varata laitteen.

4.4 Laitetietojen käsittely

Tunnusnumero	Vaatus
4.1	Ylläpitäjän on voitava lisätä järjestelmään uusi laite.
4.2	Ylläpitäjän on voitava muuttaa kaikkia laitteen tietoja.
4.3	Ylläpitäjän on voitava poistaa laite järjestelmästä.

Käyttötapaus 4

Ylläpitäjä lisää järjestelmään uuden kaiutinprosessorin.

Vaatus

4.1 Ylläpitäjän on voitava lisätä järjestelmään uusi laite.

Toimijat

Ylläpitäjä.

Alkutilanne

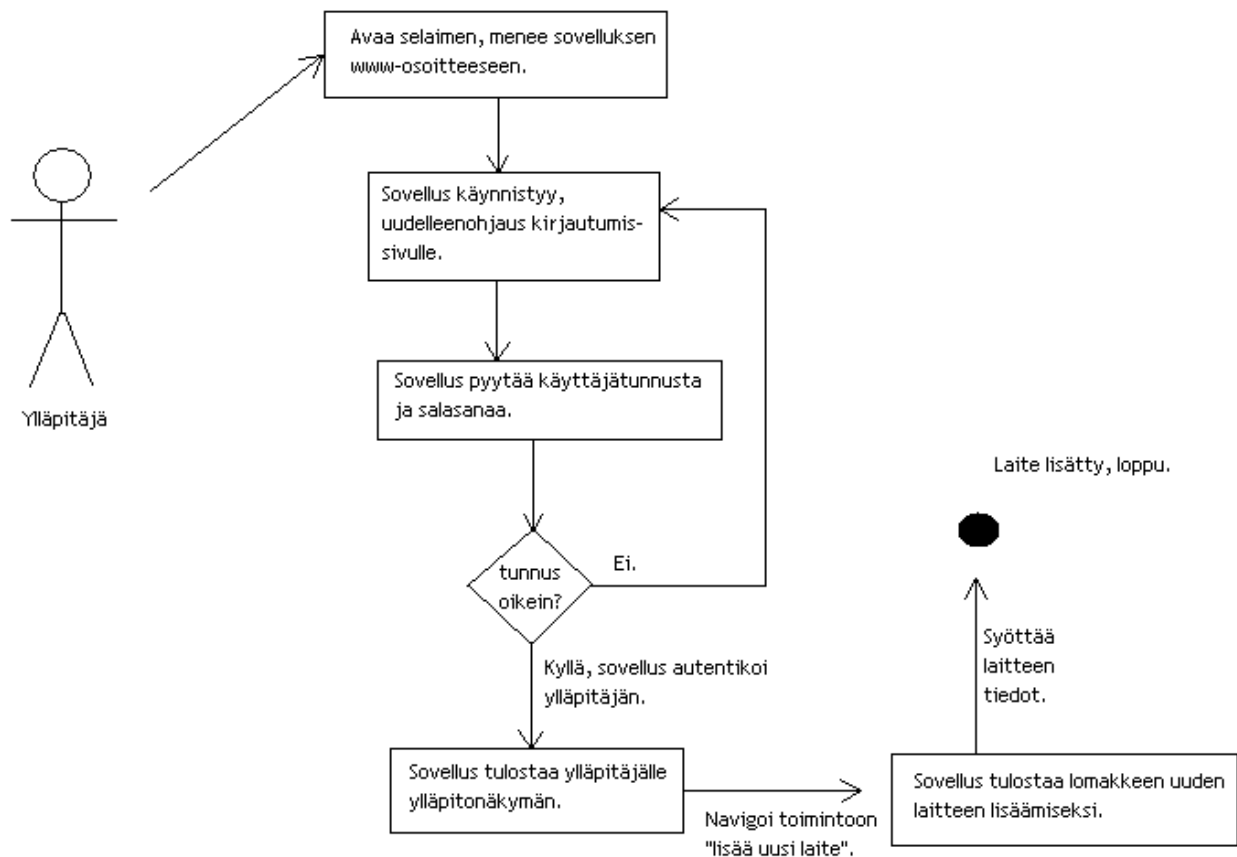
AV-kioskiin on saapunut uusi kaiutinprossori. Ylläpitäjän on lisättävä se järjestelmään.

Lopputilanne

Uuden kaiutinprossorin tiedot ovat järjestelmässä.

Kuvaus

AV-kioskiin on saapunut uusi kaiutinprossori. Ylläpitäjän on lisättävä se järjestelmään. Hän kirjautuu varausjärjestelmään henkilökuntatunnuksillaan käyttäen AV-kioskin työaseman web-selainta. Ylläpitäjä valitsee ylläpitosivulta toiminnon "lisää uusi laite", jonka jälkeen sovellus tulostaa hänelle XHTML-lomakkeen, johon ylläpitäjä täyttää laitteen tiedot. Täytettyään kaiutinprossorin tiedot lomakekenttiin hän tallentaa muutokset. Sovellus ilmoittaa ylläpitäjälle, että laite on lisätty järjestelmään onnistuneesti.



Kuva 6: Ylläpitäjä lisää uuden laitteen järjestelmään.

4.5 Lomakkeiden käsittely

Tunnusnumero	Vaatus
5.1	Ylläpitäjän on voitava tallentaa järjestelmään uuden asiakkaan tiedot.
5.2	Ylläpitäjän on voitava muuttaa asiakkaan tietoja.
5.3	Ylläpitäjän on voitava tulostaa vuokrasopimus haluamalleen laitteelle.
5.4	Ylläpitäjän on voitava tuoda asiakkaan tiedot lomakkeeseen, mikäli ne löytyvät jo järjestelmästä.
5.5	Ylläpitäjän on voitava laatia vikailmoituslomake.
5.6	Ylläpitäjän on voitava laatia vahinkoilmoituslomake.
5.7	Ylläpitäjän on voitava laatia poistomyyntisopimuslomake.

Käyttötapaus 5

Ylläpitäjä laatii kaiutinprosessorista vuokrasopimuksen.

Vaatus

5.3 Ylläpitäjän on voitava tuoda asiakkaan tiedot lomakkeeseen, mikäli ne löytyvät jo järjestelmästä.

Toimijat

Ylläpitäjä.

Alkutilanne

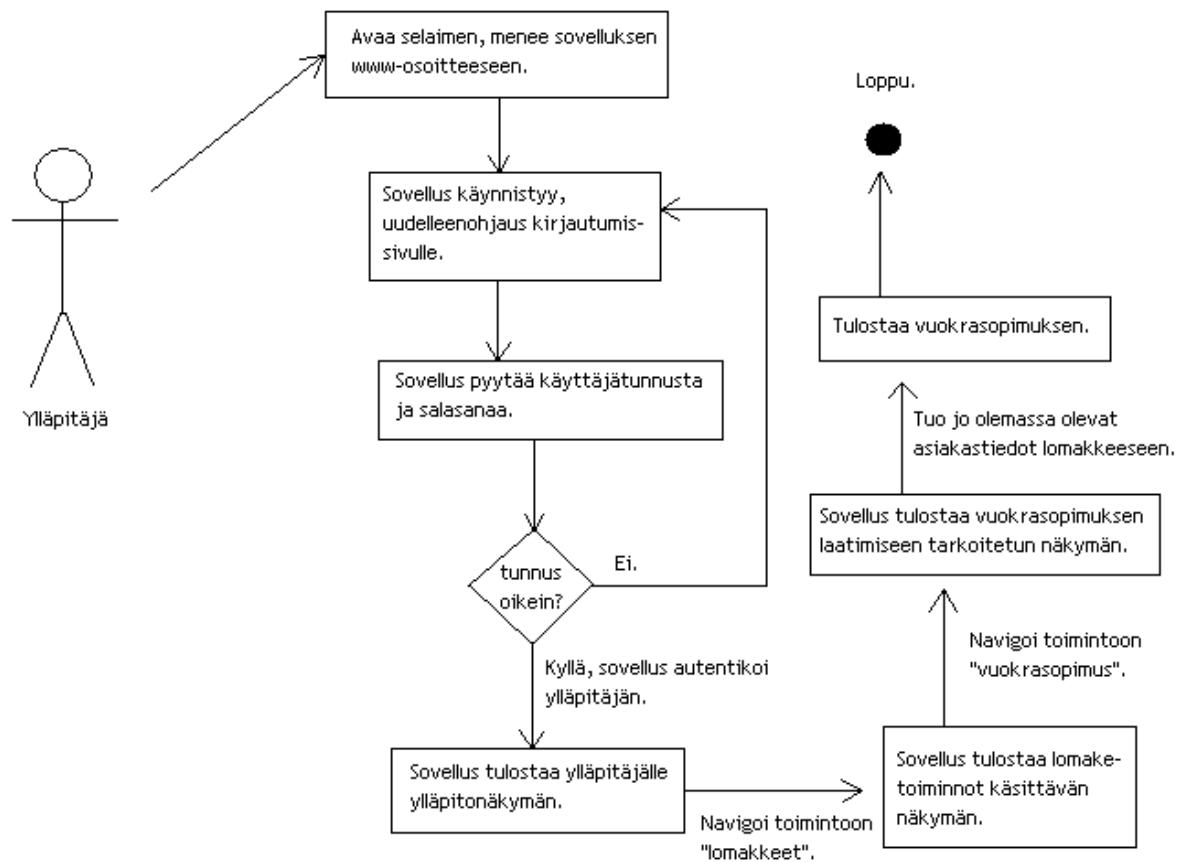
Paikallinen yrittäjä haluaa vuokrata kaiutinprosessorin AV-kioskista. Yrittäjä on vuokrannut AV-kioskin laitteita ennenkin. Ylläpitäjän täytyy laatia vuokrasopimus.

Lopputilanne

Vuokrasopimus on laadittu ja valmis allekirjoitettavaksi.

Kuvaus

Paikallinen yrittäjä haluaa vuokrata kaiutinprosessorin AV-kioskista. Ylläpitäjän täytyy laatia vuokrasopimus. Hän kirjautuu varausjärjestelmään henkilökuntatunnuksillaan käyttäen AV-kioskin työaseman web-selainta. Ylläpitäjä valitsee ylläpitosivulta toiminnon "lomakkeet". Järjestelmä palauttaa sivun, josta ylläpitäjä voi valita, minkä tyyppisen lomakkeen hän tahtoo laatia.



Kuva 7: Ylläpitäjä laatii ja tulostaa esitätetyn vuokrasopimuksen.

5 Ulkoiset liittymät

Tässä luvussa käsitellään järjestelmän toiminnan kannalta oleellista laitteistoa ja ohjelmistoja.

5.1 Laitteistoliittymät

Järjestelmä vaatii toimiakseen LAMP-kokoonpanon sujuvaan suorittamiseen kykenevän palvelintietokoneen. Muita sovelluksen käytön kannalta merkittäviä laitteita ovat viivakoodilukija sekä AV-kioskin palvelutiskin työasema, johon lukija liitetään CSV-tiedostoa luettaessa.

5.2 Ohjelmistoliittymät

Järjestelmä vaatii toimiakseen palvelinsovelluskehikon, joka käsittää Apachen taikka muun web-palvelinohjelman, PHP-komentosarjatulkin (versio 5) sekä MySQL-tietokantapalvelinohjelman (versio 5). Järjestelmän käyttäminen vaatii web-selaimen.

6 Muut ominaisuudet

Tämä luku käsittelee sovelluksen käytettävyyttä, turvallisuutta, ylläpidettävyyttä ja yhteensopivuutta sekä siirrettävyyttä.

6.1 Käytettävyys, turvallisuus

Sovelluksen käyttöliittymä on toteutettava siten, että sen eri toimintojen välillä navigointi on nopeaa ja vaivatonta. Koska sovellusta ajetaan internetissä, tietoturvaan on syytä kiinnittää erityishuomiota: etenkin käyttäjien pääsynhallinta ja autentikointi on toteutettava huolellisesti.

6.2 Ylläpidettävyys

Hyvän ylläpidettävyyden saavuttamiseksi, sovelluksen lähdekoodi toteutetaan siten, että sitä on helppo analysoida ja muokata. Käytännössä tämä onnistuu hyvän dokumentaation ja hyvien tapojen mukaisen ohjelmointityylin keinoin.

6.3 Yhteensopivuus ja siirrettävyys

Www-pohjaisuutensa vuoksi käyttöliittymä on laitteistoriippumaton. Sovellusta voi käyttää millä tahansa yleisimmistä selainohjelmista (Internet Explorer, Firefox, Chrome, Safari, Opera). Siirrettävyyseikat eivät aiheuta ongelmia, koska sovellus perustuu avoimen lähdekoodin ratkaisuihin, ja sitä voidaan ajaa millä tahansa LAMP-tyyppisellä palvelinkokoonpanolla.

7 Suunnittelurajoitteet

Tässä luvussa käsitellään sovellukselle asetettuja suunnittelurajoitteita, kuten esimerkiksi standardeja, suosituksia sekä laitteisto- ja ohjelmistorajoitteita.

7.1 Standardit ja suositukset

Sovelluksen PHP-lähdekoodin syntaksi on hyvä tarkastaa toimenpiteeseen tarkoitetun sovelluksen, kuten esimerkiksi PHPLint:in avulla (<http://www.icosaedro.it/phplint/>). PHP-tiedostot voidaan dokumentoida muodollisen PHPDoc-standardin mukaisesti (<http://www.phpdoc.de/>). Sovelluksen tulostama XHTML-merkintäkieli tarkastetaan W3C validaattorilla XHTML Strict 1.0 -standardin mukaisesti (<http://validator.w3.org/>). Myös sovelluksen käyttämä CSS-tyylitiedosto tarkastetaan W3C validaattorilla (<http://jigsaw.w3.org/css-validator/>).

7.2 Laitteistorajoitteet

Sovellusta voidaan ajaa vaatimattomassakin tuotantoympäristössä käyttäjämäärän pysyessä alhaisena.

7.3 Ohjelmistorajoitteet

Sovelluksen on pysyttävä niissä ohjelmistoalustan rajoitteissa, joita käytössä oleva tuotantoympäristö asettaa. Sovelluksen ei tule hyödyntää muita, kuin käytössä jo olevia ominaisuuksia ja palveluita (PHP 5, MySQL 5, Apache).

8 Jatkokehitys

Ohjelma on voitava toteuttaa siten, että jatkokehitys on mahdollisimman vaivatonta. Tässäkin dokumentissa suositeltujen ohjelmointi- ja dokumentaatiostandardien noudattaminen takaa lähdekoodin riittävän luettavuuden ja ymmärrettävyyden. Tätä vaatimusmäärittelyä voidaan myös jatkokehittää vaatimuksia lisäämällä, muuttamalla taikka poistamalla. Suositellut standardit ovat ainoastaan viitteellisiä ja kehitystyöhön osallistuvat sidosryhmät voivat niitä tarvittaessa muuttaa. Tehdyt muutokset ja syyt niiden taustalla on kuitenkin dokumentoitava.