Toiminnallinen määrittely

Biljardipöytien varausjärjestelmä

Tekijä: Eetu Isokirmo

Hyväksyjä:

Hyväksytty: pvm

Versio:

Dokumentin versiohistoria

|  |  |  |  |
| --- | --- | --- | --- |
| **Pvm** | **Versio** | **Tekijä** | **Tila / Muutos** |
| 5.9.2022 | 0.1 | Eetu Isokirmo | Dokumenttipohja luotu ja muotoilut |
| 5.9.2022 |  | Eetu Isokirmo | Katselmointipalaverin muutokset |
|  |  |  |  |
| x.x.2020 | 1.0 |  | Viimeistely ja julkaisuversio |

Sisällysluettelo

[1 Johdanto 3](#_Toc63673095)

[1.1 Tarkoitus ja kattavuus 3](#_Toc63673096)

[1.2 Tuote ja sen ympäristö 3](#_Toc63673097)

[1.3 Määritelmät, termit ja käytetyt lyhenteet 3](#_Toc63673098)

[2 Vaatimukset 3](#_Toc63673099)

[2.1 Toiminnalliset vaatimukset 3](#_Toc63673100)

[2.2 Ei-toiminnalliset vaatimukset 3](#_Toc63673101)

[3 Yleiskuvaus 3](#_Toc63673102)

[3.1 Ympäristö 3](#_Toc63673103)

[3.2 Toiminta 3](#_Toc63673104)

[3.3 Käyttäjät 4](#_Toc63673105)

[3.4 Yleiset rajoitteet 4](#_Toc63673106)

[3.5 Oletukset ja riippuvuudet 4](#_Toc63673107)

[4 Tiedot ja tietokanta 4](#_Toc63673108)

[4.1 Tietosisältö 4](#_Toc63673109)

[4.2 Käyttöintensiteetti 4](#_Toc63673110)

[4.3 Kapasiteettivaatimukset 4](#_Toc63673111)

[4.4 Tiedosto ja asetustiedostot 4](#_Toc63673112)

[5 Arkkitehtuuri 4](#_Toc63673113)

[5.1 Looginen arkkitehtuuri 4](#_Toc63673114)

[5.2 Tekninen arkkitehtuuri 4](#_Toc63673115)

[6 Käyttäjät 4](#_Toc63673116)

[6.1 Käyttäjäluettelo 4](#_Toc63673117)

[6.2 Internetin käyttäjät 4](#_Toc63673118)

[7 Käyttötapaukset ja käyttöliittymä 4](#_Toc63673119)

[7.1 Käyttötapaukset 5](#_Toc63673120)

[7.2 Käyttöliittymien yleiskuvaus 5](#_Toc63673121)

[7.3 Toiminnot 5](#_Toc63673122)

[7.3.1 Toiminto 1 5](#_Toc63673123)

[7.3.2 Toiminto 2 5](#_Toc63673124)

[7.3.3 Toiminto 3 5](#_Toc63673125)

[7.3.4 Toiminto 4 5](#_Toc63673126)

[8 Ulkoiset liittymät 6](#_Toc63673127)

[8.1 Laitteistoliittymä 6](#_Toc63673128)

[8.2 Ohjelmistoliittymät 6](#_Toc63673129)

[8.3 Tietoliikenneliittymät 6](#_Toc63673130)

[9 Muut ominaisuudet 6](#_Toc63673131)

[9.1 Suorituskyky ja vasteajat 6](#_Toc63673132)

[9.2 Käytettävyys, toipuminen, turvallisuus, suojaukset 6](#_Toc63673133)

[9.3 Siirrettävyys ja yhteensopivuus 6](#_Toc63673134)

[9.4 Operoitavuus 6](#_Toc63673135)

[10 Suunnittelurajoitteet 6](#_Toc63673136)

[10.1 Standardit 6](#_Toc63673137)

[10.2 Laitteistorajoitteet 6](#_Toc63673138)

[10.3 Ohjelmistorajoitteet 6](#_Toc63673139)

[10.4 Muut rajoitteet 6](#_Toc63673140)

[11 Hylätyt ratkaisuvaihtoehdot 7](#_Toc63673141)

[12 Jatkokehitysajatuksia 7](#_Toc63673142)

[LÄHTEET 7](#_Toc63673143)

[LIITTEET 7](#_Toc63673144)

## 1 Johdanto

### 1.1 Tarkoitus ja kattavuus

Tämän dokumentin on tarkoitus määritellä biljardipöytien varausjärjestelmän toiminnallisuudet. Dokumentti kattaa palvelin ei-toiminnalliset ja toiminnalliset vaatimukset. Siinä myös määritellään käyttäjätarinat ja käyttötapaukset omina tapahtuminaan. dokumentissa otetaan kantaa teknologiaratkaisuihin. tämä dokumentti pitää sisällään myös käyttöliittymämalleja.

Tämä dokumentti tulee olemaan suunnitelma sovelluksen kehitystiimin työtä varten.

### 1.2 Tuote ja sen ympäristö

Kehitettävän palvelun nimi on Jykevät mailat, Jolla tarjotaan palvelu asiakkaille, jotka haluavat varata biljardipöydän liikkeen biljardisalista. Asiakas voi varata peliaikaa haluamansa määrän tiettynä ajankohtana.

järjestelmän on tarkoitus palvella liikkeen asiakkaita ja tuottama arvo on selkeyttää biljardipöytien varausten hallinnointia.

palvelu tullaan julkistamaan liikkeen verkkosivuilla osoitteessa [www.jykevatmailat.fi](http://www.jykevatmailat.fi) Palvelun käyttäjiä ovat liikkeen asiakkaat ja yrityksen omistajat tai sen edustajat/työntekijät

### 1.3 Määritelmät, termit ja käytetyt lyhenteet

CSS Web-dokumentin tyyliohje.

Drupal/WordPress Sisällönhallinta järjestelmä.

HTML Hypertekstin kuvauskieli.

JavaScript Komentosarjakieli Web-ohjelmointiin.

MySQL SQL-tietokannan hallintajärjestelmä

PHP Ohjelmointikieli, jota hyödynnetään dynaamisten Web-sivujen tekemiseen.

Tuotteen/palvelun nimi Jykevät Mailat

Toteutusprojektin Kehitysprojektin nimi

Mean stack Mongo DB, Express, Angular, Node Js

Mern Stack Mongo DB, Express, React, Node Js

Bootstrap Responsiivinen verkkosivunkehitystyökalu

jQuery Javascript kirjasto/työkalu

Tech XYZ Lyhenne teknologiasta xyz.

## 2 Vaatimukset

Käyttäjäroolit: Asiakkaat, työntekijät tai omistajat.

Mahdollisia muita käyttäjiä tulevaisuudessa? Yritykset, kouluttajat tms.

### 

### 2.1 Toiminnalliset vaatimukset

* Asiakas
  + - Rekisteröityminen
    - Kirjautuimen
    - uloskirjautuminen
    - selata vapaita aikoja, eli näkee muiden varauksia
    - perua varauksen
    - muokata varauksia
    - selata varauksia
    - nähdä varaushistoria
* Omistaja/Henkilökunta
  + - Kirjautua
    - Kirjautua ulos
    - Selata kaikkia varausaikoja
    - selata varauksia pöytäkohtaisesti
    - selata kaikkia asiakkaita haun perusteella
    - muokata asiakkaita
    - poistaa asiakas
    - selata varauksia asiakaskohtaisesti
    - Poistamaan varauksia
    - biljardipöydän lisäys
    - biljardipöydän poistaminen
    - biljardipöydän muokkaaminen
    - aukioloaikojen muokkaus
    - Tiedotteen lisäys/poisto/muokkaus

### 2.2 Ei-toiminnalliset vaatimukset

Laadullisesti ohjelman tulee noudattaa tämän dokumentin mukaista suunnitelmaa.

Kehitystä varten luodaan automatisoitu testausympäristö, joka mahdollistaa palvelun testaamisen tehokkaasti.

Järjestelmään pääsee vain asiakkaat ja omistaja. Omistajalla on omat Admin tunnukset ja niillä voidaan hallita

varausjärjestelmän eri osa-alueita.

Palvelu on websivusto, joka tulee toimia tietokoneella, Android ja IOS Laitteilla selainta käyttäen.

Palvelua tulee pystyä käyttämään minimissään kaksikymmentä (20) henkilöä samaan aikaan.

Palvelun kehitystiimiin kuuluu vaihtelevasti kolme-kuusi henkilöä. Henkilöt rooleineen ovat: (kirjoita tähän lista)

Ulkopuolisia konsultteja ovat (tähän lista)

testauksesta vastaa (tähän lista)

Domaini ja hostingpalvelu otetaan hostingpalvelu.fi:n kautta

Sovellukseen ei tule maksuliikennettä. Asiakkaat maksavat varausmaksut liikkeen kassalla.

Palvelu ei tarvitse aktiivista moderointia. Omistaja näkee kaikki varaukset. Sovelluksen tulee olla nopea ja helppokäyttöinen ja toimia ilman viiveitä.

Sovellus tallentaa asiakkaista tietoa tietokantaan, joten sovellukselle tulee tehdä GDPR:n mukaiset dokumentit.

## 3 Yleiskuvaus

Kuvataan järjestelmää ja sen ympäristöä, toimintaa, käyttäjiä, rajoitteita, riippuvuuksia ja oletuksia. Tämä luku on eräänlainen yhteenveto siitä, mitä tulevissa kappaleissa on käyty läpi tarkemmin.

### 3.1 Ympäristö

### 3.2 Toiminta

### 3.3 Käyttäjät

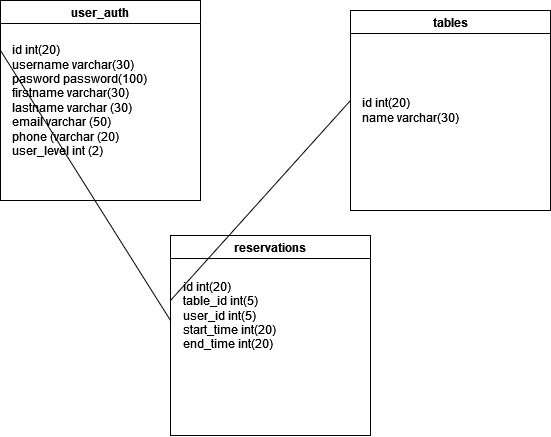
### 3.4 Yleiset rajoitteet

### 3.5 Oletukset ja riippuvuudet

## 4 Tiedot ja tietokanta

Määritellään järjestelmän tietosisältöä, käsitteitä, tietokantaa ja sen käyttöön liittyviä asioita laajemmassa mittakaavassa. Kokonaisuudessaan luku kuvaa tietokannan rakennetta ja sinne kuuluvia tietoja tarkasti.

### 4.1 Tietosisältö



### 4.2 Käyttöintensiteetti

Oletettavasti palvelua käytetään päivittäin. yksittäinen asiakas ei kuitenkaan käytä palvelua joka päivä.

Omistajalla tulee olla pääsy Admin osioon 24/7, jotta hän voi tarkastaa ja muuttaa varauksia.

Palvelu tulee olla päällä jatkuvasti.

### 4.3 Kapasiteettivaatimukset

palvelu tulee testata minimissään kahdellakymmenellä (20) samanaikaisella käyttäjällä.

Järjestelmässä ei tule olla rajoituksia käyttäjien määrässä, pöytien määrässä tai varauksien määrässä.

### 4.4 Tiedosto ja asetustiedostot

varausjärjestelmä/index.php (sovelluksen etusivu)

varausjärjestelmä/mysql.php (asetukset tietokannan yhteyttä varten)

varausjärjestelmä/css (css tiedostot)

varausjärjestelmä/images (kuvat)

varausjärjestelmä/js (javascript tiedostot)

varausjärjestelmä/admin/index.php (Admin puolen sivuston etusivu)

varausjärjestelmä/admin/mysql.php (yhteys tietokantaan)

varausjärjestelmä/admin/css (css tiedostot)

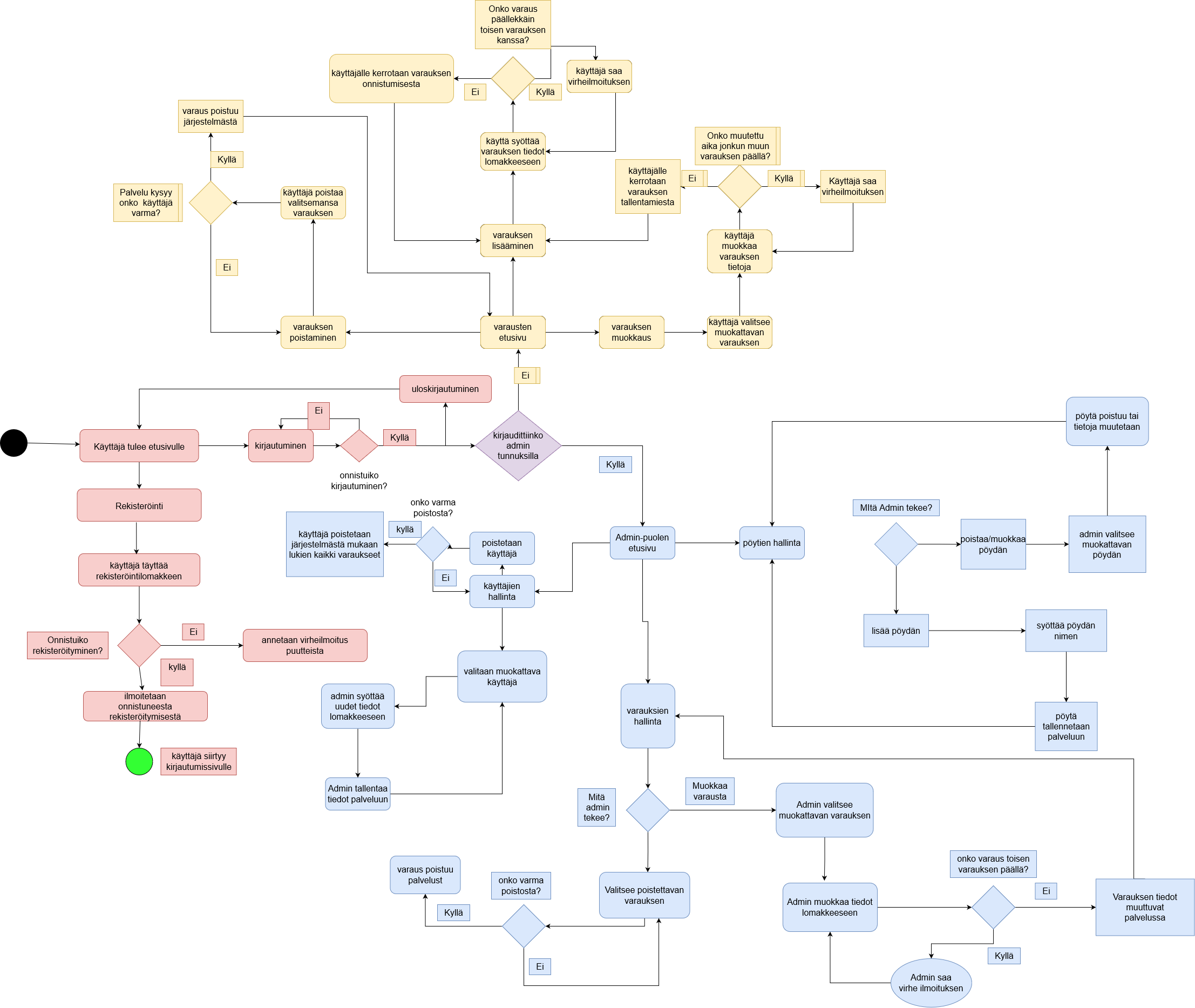
varausjärjestelmä/admin/images (kuvat)

varausjärjestelmä/admin/js (javascript tiedostot)

## 5 Arkkitehtuuri

Kuvataan järjestelmän arkkitehtuuria. Luku pitää sisällään kaavioita järjestelmän fyysisestä, loogisesta ja suunnitteluarkkitehtuurista.

### 5.1 Looginen arkkitehtuuri



### 5.2 Tekninen arkkitehtuuri

Palvelu tullaan toteuttamaan PHP ohjelmointi kielellä. Palvelu yhdistetään MySQL tietokantaan ja palvelussa olevat lomakkeet toimivat

syötteinä tietokantaan tuleville tiedoille.

Dynaamiset tietokantahaut toteutetaan AJAX-kutsuilla. Näin ei tarvitse päivittää selainta kun otetaan yhteyttä tietokantaan.

Kaikki tallennettava päivämäärädata on oltava Unix timestamp muodossa.

Muuten sivusto on yhdistelmä HMTL merkintäkieltä ja CSS tyylejä.

Tähän voi kirjoittaa muutakin informaatiota jos esim. käytetään bootstrappiä.

Tämä osio olisi hyvin erilainen jos käytössä olisi jokin Javascript-pohjainen teknologia (MEAN/MERN)

## 6 Käyttäjät

Kuvataan järjestelmän käyttäjäryhmät ja niihin kuuluvat käyttäjät, heidän roolinsa ja tehtävänsä järjestelmässä.

### 6.1 Käyttäjäluettelo

Asiakas: Asiakas on sovelluksen pääasiallinen käyttäjä. Asiakas on samalla yrityksen asiakas, joka pelaa biljardia ja tekee varauksia järjestelmään.

Omistaja: Omistaja hallinnoi varausjärjestelmää ja ylläpitää sitä. Omistajan oletustunnukset ovat admin/1234.

## 7 Käyttötapaukset ja käyttöliittymä

Kerrotaan ja kuvataan käyttöliittymiä ja niihin liittyviä käyttötapauksia ja toimintoja yksityiskohtaisesti kuvilla ja tekstillä.

#### 7.1 KÄYTTÄJÄTARINAT

Minä asiakas haluan pystyä kirjautumaan, jotta pystyn varaamaan aikoja omilla tiedoillani.

Minä asiakas haluan pystyä rekisteröitymään jotta pääsen kirjautumaan omilla tiedoillani.

Minä asiakas haluan pystyä uloskirjautumaan, jotta kukaan muu ei pääse kirjautumaan tiedoillani.

Minä asiakas haluan pystyä selaamaan vapaita aikoja, jotta tiedän milloin on vapaata.

Minä asiakas haluan pystyä varaamaan aikoja, jotta pääsen pelaamaan varmasti haluamaani aikana.

Minä asiakas haluan pystyä perumaan varaukseni, jotta pystyn muuttamaan suunnitelmiani.

Minä asiakas haluan pystyä muokkaamaan varaustani, jotta pystyn vaihtamaan ajan kohtaa.

Minä asiakas haluan pystyä selaamaan varaus historiaani, jotta näen aikaisempia varauksiani.

Minä Omistaja haluan pystyä kirjautumaan, jotta pääsen käsiksi oikeisiin tietoihin.

Minä omistaja haluan pystyä kirjautumaan ulos jotta kukaan muu ei pääse tietoihin käsiksi.

Minä omistaja haluan pystyä selaamaan kaikkia aikoja, jotta tiedän onko aikoja vapaana.

Minä omistaja haluan pystyä selaamaan asiakkaita haun perusteella, jotta löydän asiakkaat nimellä helposti.

Minä omistaja haluan pystyä muokkaamaan asiakkaan tietoja, jotta pääsen muokkaamaan asiakkaan yhteystietoja.

Minä omistaja haluan pystyä poistamaan asiakkaan, jotta asiakkaan saa pois rekisteristä.

Minä omistaja haluan pystyä selaamaan varauksia pöytä kohtaisesti, jotta näen kuka on varannut minkäkin pöydän.

Minä omistaja haluan pystyä poistamaan varauksia, jos vaikka tapahtunut virheitä.

### 7.2 Käyttötapaukset

Katso käyttötapaukset kansiosta ”käyttötapaukset”

0 alkuiset ovat asiakkaiden käyttötapauksia

1 alkuiset ovat omistajan käyttötapauksia

### 7.3 Käyttöliittymien yleiskuvaus

Käyttöliittymä halutaan pitää hillittynä, yksinkertaisena, helppokäyttöisenä ja nopeana käyttää.

Päävalikko sijaitsee yläreunassa, jonka alle palvelu listaa kaiken tarpeellisen toiminnallisuuden.

Oikea ylänurkka varataan kirjautumis- ja rekisteröitymistoiminnoille.

Valikossa pyritään välttämään dropdown toimintoja, jotta valikko on yksinkertainen käyttää.

### 7.4 Toiminnot

### 7.4.1 Toiminto 1

*Kuva: Toiminto 1*

Toiminnon nimi  
Toiminnon kuvaus

Toiminnon tarkoitus  
Toimijat  
Alkuehdot

Tyypillinen käyttötapauksen kulku (toiminnon määritys) / Vaihtoehtoinen käyttötapauksen kulku

Syöte  
Virhetilanteet

### 7.3.2 Toiminto 2

*Kuva: Toiminto 2*

Toiminnon nimi  
Toiminnon kuvaus

Toiminnon tarkoitus  
Toimijat  
Alkuehdot

Tyypillinen käyttötapauksen kulku (toiminnon määritys) / Vaihtoehtoinen käyttötapauksen kulku

Syöte  
Virhetilanteet

### 7.3.3 Toiminto 3

*Kuva: Toiminto 3*

Toiminnon nimi  
Toiminnon kuvaus

Toiminnon tarkoitus  
Toimijat  
Alkuehdot

Tyypillinen käyttötapauksen kulku (toiminnon määritys) / Vaihtoehtoinen käyttötapauksen kulku

Syöte  
Virhetilanteet

### 7.3.4 Toiminto 4

*Kuva: Toiminto 4*

Toiminnon nimi  
Toiminnon kuvaus

Toiminnon tarkoitus  
Toimijat  
Alkuehdot

Tyypillinen käyttötapauksen kulku (toiminnon määritys)

Syöte  
Virhetilanteet

## 8 Ulkoiset liittymät

Kuvataan sanallisesti sitä, mihin ohjelmistoihin, laitteistoihin ja tietoliikenteeseen järjestelmä liitetään.

### 8.1 Laitteistoliittymä

### 8.2 Ohjelmistoliittymät

### 8.3 Tietoliikenneliittymät

## 9 Muut ominaisuudet

Kerrotaan järjestelmän muista ei-toiminnallisista ominaisuuksista, kuten ylläpidettävyys, vasteajat ja käytettävyys. Luku pitää sisällään näiden määritelmät ja suunnitelmat.

### 9.1 Suorituskyky ja vasteajat

### 9.2 Käytettävyys, toipuminen, turvallisuus, suojaukset

### 9.3 Siirrettävyys ja yhteensopivuus

### 9.4 Operoitavuus

## 10 Suunnittelurajoitteet

Kohdassa käsitellään järjestelmää laadittaessa huomioon otettavat standardit ja muut rajoitteet.

### 10.1 Standardit

### 10.2 Laitteistorajoitteet

### 10.3 Ohjelmistorajoitteet

### 10.4 Muut rajoitteet

## 11 Hylätyt ratkaisuvaihtoehdot

Listaus projektin aikana hylätyistä ratkaisuvaihtoehdoista. Hylätyt ratkaisuvaihtoehdot kuvataan aika- ja asiajärjestyksessä, jossa ne ovat ilmenneet.

## 12 Jatkokehitysajatuksia

Jatkokehitysajatuksiin voi kerätä projektin aikana syntyneitä jatkokehitysajatuksia järjestelmälle tulevaisuudessa.

## LÄHTEET

## LIITTEET

Esimerkkejä liitteistä:

- järjestelmän tietokantataulut/tietokantakaavio

- luokkakaavio

- käyttötapauskaavio

- UML-kaavio

- sekvenssikaavio  
- arkkitehtuurikuvaukset

- asiakasvaatimukset/esikartoitus

- rautalanka/wire frame

- sivukartta

- käyttöliittymän prototyyppikuvat

- ylläpito-ohje