**Juho Kivistö, Jussi Karhumaa, Marko Ruokoja & Rasmus Talvitie**

**Ohjelmistotuotannon ryhmätyö**

**IT00AK37-3002 Ohjelmistotuotanto**

****

**Syyslukukausi 2023**

**KÄSITTEIDEN MÄÄRITTELY**

**Tulospalvelu**

Tulospalvelu on palvelu tai järjestelmä, joka tarjoaa ajantasaista tietoa tai tuloksia tietyistä tapahtumista, kilpailuista, kokeista, peleistä tai muista suorituksista. Tulospalvelu voi kattaa monenlaisia aloja, kuten urheilu, koulutus, kilpailut ja monia muita toimialoja. Sen keskeisenä tehtävänä on tarjota käyttäjilleen tietoa suoritusten tuloksista, tilastoista, aikatauluista ja muista tapahtumiin liittyvistä tiedoista.

**KÄSITTEIDEN MÄÄRITTELY**

**SISÄLLYS**

[1 JOHDANTO 1](#_Toc152767596)

[2 Toimeksiantaja 2](#_Toc152767597)

[3 Sidosryhmät ja profiilit 4](#_Toc152767598)

[4 Tunnistetut vaatimukset 5](#_Toc152767599)

[5 Palveluun liittyvät asiakaspolut 6](#_Toc152767600)

[6 Palvelun järjestelmävaatimukset 7](#_Toc152767601)

[7 Palveluun liittyvät laitevaatimukset 8](#_Toc152767602)

[8 Palvelun toiminnalliset vaatimukset 9](#_Toc152767603)

[9 Palvelun ei-toiminnalliset vaatimukset 10](#_Toc152767604)

[Lähteet 11](#_Toc152767605)

[10 Projektisuunnitelma 12](#_Toc152767606)

[10.1 Projektin kuvaus 12](#_Toc152767607)

[10.2 Tuotteen kuvaus 12](#_Toc152767608)

[10.3 Projektin tavoitteet 12](#_Toc152767609)

[10.4 Projektin vaiheistus & resurssointi 12](#_Toc152767610)

[10.5 Organisaation rakenne 13](#_Toc152767611)

[10.6 Projektin riskienhallinta 14](#_Toc152767612)

[10.7 Riskien arviointi 15](#_Toc152767613)

[10.8 Riskien hallinta 15](#_Toc152767614)

[10.9 Projektin työvaiheet & aikataulu 15](#_Toc152767615)

[10.10 Projektin tekniikka 16](#_Toc152767616)

[10.11 Projektin dokumentointi 17](#_Toc152767617)

[10.12 Projektin seuranta & laadunvarmistus 18](#_Toc152767618)

[11 Arkkitehtuurisuunnittelu 19](#_Toc152767619)

[11.1 Moduulit 19](#_Toc152767620)

[11.2 Käyttöliittymä 21](#_Toc152767621)

[11.2.1 Ylätunniste 22](#_Toc152767622)

[11.2.2 Sivupalkki 22](#_Toc152767623)

[11.2.3 Pääsivu 23](#_Toc152767624)

[11.3 Saavutettavuus 24](#_Toc152767625)

[11.3.1 Saavutettavuuden toteutus 24](#_Toc152767626)

[11.3.2 Havaittavuus 24](#_Toc152767627)

[11.3.3 Hallittavuus 25](#_Toc152767628)

[11.3.4 Ymmärrettävyys 25](#_Toc152767629)

[11.3.5 Toimintavarmuus 25](#_Toc152767630)

[12 Testaussuunnitelma 26](#_Toc152767631)

[12.1 Johdanto 26](#_Toc152767632)

[12.2 Sanasto 26](#_Toc152767633)

[12.3 Testausympäristö 27](#_Toc152767634)

# JOHDANTO

Jalkapallo on maailmanlaajuisesti suosituin joukkueurheilulaji, jonka suosio kasvaa vuosi vuodelta. Jalkapallon seuraaminen ja tulosten saaminen on intohimoisten fanien ja harrastajien elämässä keskeinen osa. Tässä johdannossa esittelemme projektin jalkapallon tulospalvelusta, jonka tarkoituksena on tarjota jalkapallofaneille ja ammattilaisille helppokäyttöinen ja ajantasainen palvelu tulosten, tilastojen ja tiedon seuraamiseen.

Projektin tavoitteena on luoda jalkapallon tulospalvelu, joka tarjoaa monipuolisia ominaisuuksia, kuten reaaliaikaisten ottelutulosten seuranta, pelaajien tilastot, sarjataulukot, tulevien otteluiden aikataulut ja tärkeimmät uutiset. Tämän palvelun avulla jalkapallon ystävät voivat pysyä ajan tasalla suosikkijoukkueidensa suorituksista, seurata pelaajien kehitystä ja saada tietoa lajin tapahtumista.

Projektin kohdeyleisönä ovat sekä jalkapallon seuraajat että ammattilaiset, kuten valmentajat, pelaajat ja analyytikot, jotka tarvitsevat luotettavaa ja nopeaa tietoa otteluiden kulusta ja tuloksista. Palvelu on suunniteltu palvelemaan niin kotikatsomoita kuin myös median ja muiden sidosryhmien tarpeita.

Projektissa keskitymme paitsi käyttäjäystävälliseen käyttöliittymään myös siihen, että tieto on ajantasaista ja virheetöntä. Luotettavuus ja tietoturva ovat keskeisiä tekijöitä, kun tarjoamme tietoa suurelle yleisölle.

# Toimeksiantaja

Projekti liittyy Centria-ammattikorkeakoulun ohjelmistotuotannon kurssiin, jolla harjoitellaan ohjelmistotuotantoa projektiluontoisesti. Projekti tehdään neljän hengen ryhmässä, joka suorittaa annetut tehtävät kurssin aikataulun mukaisesti. Kurssin aikana on tarkoitus tutustua ohjelmistotuotannon eri osa-alueisiin kuten vaatimusmäärittely, projektisuunnitelman teko ja roolitus, moduulisuunnittelu, tuotteenhallinta ja testaus. Projektin kesto on kuusi viikkoa.

**4 Sidosryhmäkartta**

A diagram of a diagram

Description automatically generated

# Sidosryhmät ja profiilit

1. **Fanit**

Jalkapallo on maailman suosituin urheilulaji. Arvioiden mukaan noin puolet maailman väestöstä joko seuraa tai harrastaa jalkapalloa. Keskiverto jalkapallon seuraaja seuraa vähintään oman joukkueensa otteluita ja lisäksi kansainvälisiä otteluita.

1. **Seurat**

Suomessa jalkapallon kattojärjestönä toimii Suomen palloliitto, jonka alaisuudessa toimii noin 930 jalkapalloseuraa. Jalkapallojoukkue pelaa kaudessa oman sarjansa vähintään kerran läpi, jolloin on olennaista seurata miten oman sarjansa joukkueilla, menee muita joukkueita vastaan.

1. **Toimittajat**

Arviolta joka kolmas suomalainen seuraa jossain määrin jalkapalloa. Tämä tekee lajista mielenkiintoisen myös median näkökulmasta. Jalkapallosta kirjoitetaan useita artikkeleita päivittäin suomalaisessa mediassa. Tulosten seuraaminen reaaliajassa auttaa toimittajia kirjoittamaan uutisia välittömästi.

1. **Sponsorit**

Sponsorointi ja yhteistyökumppanit ovat olennainen osa kilpaurheilua. Yhteistyökumppanit mahdollistavat rahoituksellaan seurojen toiminnan, ja seurat antavat näkyvyyttä yhteistyökumppaneille. Sponsoreilla on usein halu pysyä ajan tasalla “oman” joukkueensa asioista, jolloin he tarvitsevat luotettavaa tulospalvelua.

1. **Kilpailujen järjestäjät**

Suomessa järjestetään vuosittain satoja jalkapalloon liittyviä tapahtumia. Näitä tapahtumia seuraan sadan tuhannet ihmiset ympäri maata. Kilpailun järjestäjillä on tarve saada tulokset luotettavasti talteen ja yleisön tietoisuuteen. Kilpailun järjestävät haluavat tulospalvelun olevan ehdottoman luotettava ja reaaliaikainen.

1. **Järjestelmän kehittäjät**

Järjestelmän kehittäjät varmistavat kehitystyössään järjestelmän luotettavuuden ja toimivuuden. Kehittäjille annetaan tietyt raja-arvot, joiden puitteissa palvelut tulee toimia missäkin olosuhteissa.

1. **Ylläpitäjät**

Ylläpitäjät pitävät palvelun pystyssä ja varmistavat toiminnallaan järjestelmien luotettavuuden. Ylläpitäjien vastuulla on reagoida mahdollisiin ongelmatilanteisiin ja raportoida eteenpäin, mikäli ongelma ei ole heidän ratkaistavissa.

# Tunnistetut vaatimukset

1. **Luotettavuus**

Tulosten tulee olla ehdottoman luotettavia.

1. **Riittävän nopea**

Tulosten tulee tulla niin reaaliaikaisia kuin mahdollista.

1. **Helppokäyttöinen ja selkeä**

Käyttöliittymän tulee olla selkeä ja kaikki tarpeellinen tieto tulee olla saatavilla välittömästi sovelluksen aukaisun jälkeen. Mahdollisuus valita oma joukkue, halutessaan lähettää ilmoituksen käyttäjälle.

1. **Turvallisuus**

Sovellukseen on mahdollistua kirjautua, että oman joukkueen tiedot liikkuvat käyttöliittymästä toiseen. Kirjautumistietojen hallinnan tulee olla täysin turvallista.

# Palveluun liittyvät asiakaspolut

1. **Urheilufanit**

23-vuotias Jani Oulusta on intohimoinen FC Oulun kannattaja. Jani kuuluu FC Oulun kannattajaklubiin ja elää ja hengittää jalkapalloa läpi vuoden. Hän seuraa myös ulkomaisia liigoja. Jani kuulee ystävältään uudesta palvelusta, josta hän voi seurata omien joukkueidensa otteluita ja jälkikäteen myös otteluiden tuloksia. Jani on käyttänyt lähes vastaavanlaisia palveluita, mutta käyttö on ollut vähän hankalaa ja monimutkaista. Jani päättää kokeilla, ja kokeilee sovellusta ollessaan tietokoneen äärellä.

1. **Toimittajat**

 Iltapäivälehden urheilutoimittaja seuraa erilaisia urheilutapahtumia päivittäin läpi vuoden. Hän harrastaa myös itse jääkiekkoa kaveriporukkansa kanssa. Hän seuraa sosiaalisessa mediassa lukuisia urheiluun liittyviä sivustoja. Hän näkee sosiaalisessa mediassa mainoksen uudesta palvelusta josta voi seurata kaikkia meneillään olevia jalkapallo-otteluita yhdellä silmäyksellä. Hän päättää käydä palvelun sivustolla ja kokeilla.

1. **Jalkapallojoukkueet**

 FC Oulun junioritoiminnasta vastaava Marko on 50-vuotias perheenisä. Hän on päätynyt juniorijalkapallon pariin omien lastensa kautta. Hänen tyttönsä, 11-vuotias Jasmin, pelaa FC Oulun junioreissa. Marko viettää mielellään 3–5 iltaa viikossa jalkapallokentän laidalla huolehtien, että joukkueella on kaikki tarvittava. Kesäisin hän matkustaa liki joka viikonloppu joukkueen mukana ympäri suomea turnauksissa. Marko ei ole mikään erityisen hyvä teknologiaan liittyvissä asioissa. Eräällä turnausmatkalla eräs joukkueen jäsenistä näyttää turnaukseen liittyviä taulukoita ja tuloksia puhelimestaan. Marko haluaa saman sovelluksen itselleen, ja kaveri näyttää kuinka hän sen saa.

1. **Yksilöurheilija**

Julia, 15 vuotta, harrastaa tavoitteellisesti jalkapalloa Revonlahden Huima joukkueessa. Hän käy harjoituksissa kesäisin 6 kertaa viikossa, ja talvisin hän harrastaa voimailua. Julia käy koulussa Revonlahden yläasteella yhdeksännellä luokalla, missä on myös muita samassa joukkueessa pelaavia. Vapaa-ajallaan hän seuraa lempipelaajansa Robin Lodin tekemisiä amerikkalaisessa liigassa. Hän näkee sosiaalisen median feedissä mainoksen palvelusta, josta voi seurata, sekä oman liigan, että myös ulkomaisten liigojen otteluista ja tapahtumista. Hän kirjautuu sivustolle ja kokeilee miten siitä saisi eniten irti.

1. **Vedonlyöjät**

 Pekka, 35 vuotta, on Pieksämäellä Kyrön sahalla työskentelevä mies. Hän asuu avopuolisonsa kanssa ja ovat puolisonsa kanssa molemmat innokkaita urheilun seuraajia. Heidän suosikkijoukkueensa on PSG, joka pelaa Pariisissa. He seuraavat myös muita liigoja ympäri maailmaa. Pekka hankkii sivutuloja lyömällä vetoa jalkapallo otteluiden tuloksista. Hän on suhteellisen hyvä siinä mitä tekee, sillä hän tekee taustatyönsä huolella. Hän tarvitsee sovellusta mikä pitää hänet reaaliajassa kärryillä mitä missäkin ottelussa on tilanne. Hänen työkaverinsa on törmännyt netissä sivustolle mikä pitää kirjaa kaikista meneillään olevista otteluista, ja antaa ilmoituksen, mikäli seurannassa oleva ottelussa tapahtuu jotakin merkittävää. Pekka kirjautuu sivustolle ja merkkaa suosikkinsa.

# Palvelun järjestelmävaatimukset

Tämä tulospalvelu on suunniteltu yksinomaan toimimaan verkossa ilman erillistä applikaatiota. Tässä esitellään perusideat ja tarpeet, joita järjestelmän on täytettävä.

Tulosten seurantapalvelun oleellisin toiminnallinen piirre on sen helppokäyttöisyys ja intuitiivisuus. Tulospalvelu sisältää reaaliaikaisen päivityksen ottelutuloksista, pelaajatilastoista ja taktisista tiedoista. Palvelun tietokantaratkaisun tulee olla tehokas ja laaja-alainen, jotta ottelutietojen hakeminen ja tallentaminen on mahdollista. Palvelun käyttäjämäärin mukaan on tärkeää varmistaa, että käytetty palvelinarkkitehtuuri on skaalautuva. Mikäli palveluun on mahdollista rekisteröityjä käyttäjäksi, on äärimmäisen tärkeää huomioida palvelun tietoturvallisuus. Tässä tilanteessa esimerkiksi HTTPS-suojaus on ehdoton.

Tulospalvelumme on verkkopohjainen palvelu, joten sen toiminnallisuuden varmistaminen erilaisilla päätelaitteilla on olennaista. Palvelun tulee toimia yhtä saumattomasti puhelimella, tietokoneella ja tietokoneella. Palvelun on myös toimittava eri selaimilla, kuten Google Cromella ja Firefoxilla. Palvelun monipuolisuus on tärkeää, jotta käyttäjäkunta on mahdollisimman laaja ja sitoutettava.

Yhteenvetona todettakoon, että seurantapalvelun menestys perustuu järjestelmäarkkitehtuuriin, joka on joustava ja vakaa. Käyttäjäkokemuksen kannalta on äärimmäisen tärkeää kehittää tulospalvelu, joka ottaa huomioon eri käyttäjäien mahdolliset käyttökanavat.

# Palveluun liittyvät laitevaatimukset

Kuten aikaisemmin järjestelmävaatimuksissa mainittiin, on tärkeää varmistaa palvelumme yhteensopivuus erilaisten päätelaitteiden kanssa. Tässä osiossa käsittelemme laitevaatimuksia ja sitä miten varmistamme palvelun käytettävyyden laitteiden osalta.

Palvelun tulee olla yhteensopiva erilaisten laitteiden ja verkkoalustojen kanssa. Palvelun tulisi toimia puhelimella, tabletilla sekä tietokoneella yhtä tehokkaasi. Käyttökokemuksen osalta on esimerkiksi tärkeää varmistaa, että palvelu on käytettävissä eri näyttö- ja resoluutiokoilla. Tämä varmistaa palvelun joustavan käytettävyyden.

Palvelu käsittelee suurta määrää referenssidataa joka hetki, joten meidän tulee varmistaa palvelun suorituskyky. Suorituskyvyn kannalta on tärkeää optimoida palvelun koodiratkaisut sekä varmistaa, että palvelu on suunniteltu resurssitehokkaasti. Palvelun suunnitteluvaiheessa on olennaista tunnistaa mahdolliset vaativat käyttötilanteet, jotta järjestelmällä on vaadittava kapasiteetti toiminnan suorittamiseen. Käyttökokemuksen osalta on myös olennaista varmistaa palvelun taukoamaton käytettävyys. Tämä vaatii verkkoyhteyden, joka on nopea ja luotettava. Tämä varmistaa myös ottelutuloksien saumattoman ja nopean saannin ilman sen kummempia viiveitä. Laitevaatimuksien osalta tulee myös huomioida itse laitteiden päivitys- ja huoltomekanismit. Käyttäjän kannalta saumattomin ratkaisu on luoda automatisoituja päivitysratkaisuja, jotka eivät vaadi käyttäjän omaa osallistumista.

# Palvelun toiminnalliset vaatimukset

Jalkapallon tulospalvelun oleellisin toiminto on tarjota käyttäjilleen luotettavaa ja ajantasaista informaatiota ottelutuloksista. Palvelun tulee ylläpitää tätä standardia joka hetki ja tarjota miellyttävä käyttäjäkokemus kaikilla alustoilla.

Palvelu hakee ottelutiedot tulosarkisosta, jonka tarjoaa ulkopuolinen kolmas osapuoli API-kutsuja hyödyntäen. Palveluiden välille on rakennettu rajapinta. Käyttäjä voi seurata palvelusta tuloksia manuaalisesti tai vaihtoehtoisesti kirjautumalla palveluun. Palvelu on toistaiseksi käytettävissä ainoastaan suomeksi. Käyttäjän on mahdollista kustomoida hakukokemustaan esimerkiksi rajaamalla eri seuroja pois tuloslistauksesta. Rekisteröityneellä käyttäjällä on mahdollisuus valita itselleen push-ilmoituksen, jolloin käyttäjä saa ilmoituksen reaaliajassa ottelutuloksien muutoksista.

Palvelu tarjoaa mahdollisuuden myös tarkastella eri pelaajien tilastoja, eri sarjojen tulotaulukoita, tulevien ottelujen aikatauluja ja jalkapallomaailman tärkeimpiä uutisia. Näistä käyttäjä voi halutessaan myös tilata push-notifikaatioita.

# Palvelun ei-toiminnalliset vaatimukset

Tulospalveluun liittyvistä vaatimuksista osa liittyy olennaisesti ei-toiminnallisiin ominaisuuksiin. Tulospalvelun käytön tulee olla helppoa, turvallista ja tehokasta kaikille osapuolille, sekä käyttäjille, että palvelun ylläpitäjille.

Palvelun tulee olla käyttöliittymältään selkeä ja helppokäyttöinen. Käyttäjien tulee pystyä navigoimaan palvelussa helposti. Palvelun tulee kyetä palvelemaan vähintään 10000 yhtäaikaista käyttäjää kerralla ilman että palvelun toiminnassa ilmenee havaittavia viiveitä. Palvelun tulee päivittää tarvittavat tulokset 1 sekunnin välein. Palveluun liittyvien käyttäjätunnusten salasanojen on oltava ehdottoman salattuja, sekä käyttäjistä kerättävää dataa tulee käsitellä nimettömästi. Palveluun tehtävien päivitysten tulee tapahtua niin, että palveluun ei aiheudu kohtuuttoman pitkiä katkoksia. Palvelun tulee toimia kaikilla nykyaikaisilla käyttöjärjestelmillä ja selaimilla.

# Lähteet

<https://www.theseus.fi/handle/10024/427088>

<https://www.theseus.fi/bitstream/handle/10024/147943/risikko_roosa-liina.pdf?sequence=1>

<https://www.smartium.com/projektisuunnitelma-nain-suunnittelet-projektin-ja-seuraat-sen-etenemista/>

<https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/>

<https://blog.lemonsoft.fi/millainen-on-hyva-projektisuunnitelma>

<https://unlimited.hamk.fi/yrittajyys-ja-liiketoiminta/miten-projektisuunnitelma-tehdaan/>

<https://www.theseus.fi/bitstream/handle/10024/125090/Opinnaytetyo_Kettunen.pdf;jsessionid=62AEB2CDA66C1C583A50BB1B4733FE7A?sequence=1>

<https://www.cs.helsinki.fi/group/playoff/docs/testaussuunnitelma/testaussuunnitelma_ja_raportti.pdf>

# Projektisuunnitelma

## Projektin kuvaus

Projektin toimeksiantaja on Centria ammattikorkeakoulu. Projekti toteutetaan ohjelmistotuotannon kurssilla kuuden viikon aikana, neljän hengen ryhmässä. Projektin tarkoituksena on tutustua ja oppia ohjelmistoprojektin suunnittelusta ja toteutuksesta. Valittuna aiheena on jalkapallo tulospalvelu ja tarkoituksena on rakentaa palvelu, joka tarjoaa informaatiota jalkapalloon liittyvistä aiheista ja tuloksista sen faneille, pelaajille, tiimeille yms.

## Tuotteen kuvaus

Tuote on jalkapallo tulospalvelu. Palvelu toimii verkossa ja tarjoaa käyttäjilleen pääasiassa ajantasaisia tuloksia ja tietoja kilpailuista, peleistä tms. Tulokset päivittyvät lähestulkoon reaaliaikaisesti ja hankitaan kolmannen osapuolen tulosarkistosta. Toistaiseksi palvelun kieli ainut tarjoama kieli on Suomi.  Palvelusta löytyy myös mahdollisuus tarkastella eri pelaajien tilastoja, menneiden ja meneillään olevien sarjojen tuloksia ja tulevien otteluiden aikatauluja. Käyttäjä löytää palvelusta myös tärkeimmät jalkapalloon liittyvät uutiset.  Käyttäjä voi luoda palveluun oman tilin joka helpottaa häntä kiinnostavien aiheiden seuraamisessa. Rekisteröitynyt käyttäjä voi myös tilata push-notifikaatioita haluamistaan tiedoista.

## Projektin tavoitteet

Tavoitteena on suunnitella aiemmin kuvattu tulospalvelu. Samalla on tavoite myös oppia projektisuunnitelman toteuttamisesta, projektin arkkitehtuuri- ja moduulisuunnitelusta, tuotteen hallinasta ja testauksesta. Palvelun tulee kyetä tukemaan vähintään kymmentä tuhatta käyttäjää kerallaan. Palvelun tulee olla helppo käyttöinen, luotettava ja turvallinen. Käyttäjien tietoja tulee käsitiellä mahdollisimman turvallisesti, joten HTTPS-salaus on ehdoton. Käyttäjistä kerättyä dataa tulee käsitellä anonyymisti, milloin se on mahdollista. Palvelun tulisi olla helposti skaalattavissa ja ylläpidettävissä. Käyttäjälle näkyvän datan tulee olla luotettavaa ja selkeää.    
   
Palvelun käyttöliittymän täytyy olla helppokäyttöinen ja selkeästi luettavissa. Sovelluksen auetessa käyttäjälle tulisi näkyä viimeisimmät tulokset ja tiedot. Halutessaan kirjautunut käyttäjä voi valita minkä joukkueen tai seuran tulokset hän haluaa nähdä sovelluksen avatessaan.   
   
Käyttäjien täytyy myös pystyä luottamaan palvelun tarjoamaan dataan. Täten palvelun täytyy huomata mahdolliset data virheet ja korjata ne ennen kuin ne julkaistaan käyttäjille. Palvelun täytyy myös toimia nopeasti ja miellyttävästi kaikilla tuetuilla alustoilla.

## Projektin vaiheistus & resurssointi

Projekti jakautuu karkeasti neljään vaiheeseen. Ensimmäisessä vaiheessa työlle päätetään tavoitteet ja määritellään vaatimukset. Työ tullaan tekemään viikoittaisia tapaamisia hyödyntäen ja niissä määritellään jokaiselle tiimin jäsenelle seuraavan viikon tehtävät. Toisella viikolla tehdään tarkempi vaatimusmäärittely, mitä muokataan projektin edetessä tarpeen tullen. Vaatimusmäärittelyssä listataan mahdollisimman tarkasti vaaditut vaatimukset, priorisoidaan ne ja jaetaan toiminnallisiin- ja ei toiminnallisiin.

Työvaiheiden suunnittelu perustuu kurssin viikko-ohjelmaan, jossa määräytyy kullekin viikolle oma osa-alue mihin tiimi seuraavaksi viikoksi pureutuu.

Tavoitteiden määrittelyyn ja työn suunnitteluun kuuluu olennaisesti myös projektisuunnitelman teko. Projektisuunnitelmasta tulee löytyä seuraavat asiat:

* Kuvaus projektista
* Tavoitteet ja lopputuloksen määritelmän
* Projektin vaiheet ja aikataulu
* Roolit ja vastuut
* Resurssit ja niiden käytön
* Riskit ja niiden hallintasuunnitelma
* Projektin dokumentointi

Toteutusvaiheessa projektia aletaan suunnittelemaan konkreettisesti. Suunnitellaan arkkitehtuuri ja moduulit. Lopputuotteelle määritellään ulkoasu sekä tekniset moduulit, jolla lopputuote saadaan rakennettua. Tässä vaiheessa toteutetaan myös ensimmäinen versio lopullisesta tuotteesta.

Katselmoinnissa tarkastetaan, että lopputuote täyttää vaatimuksissa määrätyt kohdat. Tässä vaiheessa tehdään viimeiset testit, joissa tarkastetaan toimivuus vaadituilla alustoilla vaadituissa ympäristöissä.

Retrospektiivissä eli viimeisessä vaiheessa käydään tiimin ja asiakkaan kanssa projekti läpi. Pohditaan esimerkiksi seuraavia asioita:

* Mikä meni hyvin?
* Mitä voimme seuraavan kerran tehdä paremmin?
* Mitä toimenpiteitä pitäisi tehdä?

Prosessissa käydään läpi kysymykset läpi yksi kerrallaan. Kaikki osallistujat saavat kertoa oman näkemyksen ja näkemykset kirjataan ylös. Lopuksi keskustellaan toimenpiteistä, jotka tulisi toteuttaa virheiden korjaamiseksi.

Tiimissä työt jaetaan neljälle tasavertaiselle tiimin jäsenelle, jotka työskentelevät itsenäisesti kunkin viikon ohjelman mukaisesti. Tiimi pitää viikoittain yhteisen palaverin, jossa käydään läpi edellisen viikon aikana tehty työ, ja seuraavan viikon ohjelma. Tässä tapaamisessa jaetaan seuraavan viikon työt ja käydään läpi seuraavan viikon mahdolliset haasteet. Palaverille varataan aikaa 2 tuntia. Lisäksi jokainen tiimin jäsen käyttää viikostaan 2 tuntia projektin edistämiseen. Tämän kahden tunnin aikana hoidetaan oma osuus projektin edistämisestä. Mikäli on tarpeen, tiimi voi järjestää useamman tapaamisen.

Projektin onnistunut lopputulos vaatii siis 4h\*4hlö\*6 viikkoa = 96 työtuntia.

## Organisaation rakenne

*Projektipäällikkö*

Juho

* Vastaa projektin aikataulusta
* Huolehtii “ison kuvan” toteutumisesta
* Vetää palaverit

*Viestintäpäällikkö*

Marko

* Huolehtii dokumentaation oikeellisuuden
* Tekee riskiarvion

*Tuotepäällikkö*

Rasmus

* Huolehtii, että kullekin vaiheelle määritellyt vaatimukset täyttyvät
* Vastaa teknisen toteutuksen suunnittelusta

*Käytettävyyspäällikkö*

Jussi

* Vastaa käyttöliittymäsuunnittelusta
* Vastaa saavutettavuudesta
* Vastaa testauksista

## Projektin riskienhallinta

Projektinhallinta on elämistä riskien kanssa ja siksi olennaisena osana projekteja kuuluu riskienhallinnan prosessit, jotka auttavat vähentämään epävarmuutta. Pieneltä tuntuvat ongelmat saattavat kertautua ajan kuluessa ja aiheuttaa suuria kustannusnousuja sekä ongelmia aikatauluissa pysymisiin. Yleensä mikään yksittäinen tekijä ei riskiä aiheuta, mutta riski tavallisesti toteutuu useista tekijöistä koostuvan tapahtumaketjun yhteisvaikutuksena. Mikäli tällainen tapahtumaketju saadaan katkaistua mistä kohtaa tahansa, pienenee riski olennaisesti tai peräti se saadaan poistumaan. Riskit täytyy tunnistaa ja todeta ja sen myötä pitää ryhtyä toimenpiteisiin riskien vähentämiseksi tai poistamiseksi ja siksi riskien hallintaan on oltava sovitut toimenpiteet, kuten riskien analysointi, riskilistan laatiminen, toimenpiteistä sopiminen ja

Riskienhallinta tai sen puute on merkittävä tekijä projektin hallinnan kannalta. Voihan olla joskus projekteja, joissa riskit eivät toteudu, mutta sen varaan ei kannata nojata. Onnistunut riskienhallinta käsittää mahdollisten riskien tunnistamisen ja niihin varautumisen ennakolta ja on tietenkin selvää, jos tämä osa-alue jää tekemättä tai on puutteellinen, niin se heijastuu heti projektin aikatauluun ja kustannuksiin.

A diagram of a risk management process

Description automatically generated

## Riskien arviointi

Projektin toteutumista voivat vaikeuttaa esimerkiksi tavoitteiden epäselvyys, liian pieni budjetti, tai aikataulussa pysyminen. Riskien tunnistaminen ja niiden ennalta suunnitellut varotoimet turvaavat projektin onnistumisen, vaikka jotain odottamatonta tapahtuisikin. Projektityöskentelyn ongelmat eivät yleensä liity tekniikkaan työvälineisiin tai lopputuotteen sisältökysymyksiin. Yleensä projektin epäonnistuminen johtuu hallinnan ja menetelmien riittämättömyydestä eikä niinkään teknisistä syistä. Yleisesti ottaen projektiryhmän asiantuntijat ovat tehtäviensä tasalla ja vaikeudet johtuvat silloin joko huonosta organisoinnista tai puutteellisesta suunnittelusta.

Yleinen ongelma projekteissa on, että tulee erimielisyyttä toimituksen sisällöstä ja rajoista. Projektin osapuolet pohtivat, mitä piti tehdä, kuka lupasi mitä, kenen olisi pitänyt tietää luvata ja mitä. Toinen yleinen ongelma on henkilöiden vaihtuvuus projekteissa ja kolmantena ongelmat ja ristiriidat projektin henkilöiden kesken. Suurimpaan osaan edellä mainituista asioista voidaan valmistautua ennakolta hyvällä suunnittelulla ja riskien hallinnalla, mutta henkilökemioihin ratkaisuksi tulee kylmästi henkilöiden vaihtaminen, mikäli neuvotteluteitse ei asia selviä.

## Riskien hallinta

Kaikki projektit ovat riskialttiita ja ne voivat epäonnistua, vaikka riskit ovat etukäteen tiedossa, jos riskienhallintaan ei ole mitään suunnitelmaa. Riskien tunnistaminen ja niihin varautuminen tulee suunnitella ennen projektin aloittamista ja niitä pitää myös seurata määräajoin jatkuvasti projektikokouksissa tai ohjausryhmän palavereissa. Riskianalyysi tulee olla osa projektisuunnitelmaa, josta se on jokaisen nähtävillä. Vaikka ongelmatilanteet ja ristiriidat ovat projektien hankalimmat osa-alueet, voidaan hyvällä ennakkovalmistautumisella jopa ennaltaehkäistä tai ainakin helpottaa näiden tilanteiden hoitamista. Riskianalyysitaulukon, jossa on merkitty ongelmien riskit, todennäköisyydet ja riskin vastuuhenkilö, läpikäynti kokouksissa palvelee niin tilaajaa kuin muita projektiryhmäläisiä tiedostamalla mahdolliset riskit.

## Projektin työvaiheet & aikataulu

Yleisesti ottaen projekti on jaettavissa eri työvaiheisiin, kuten suunnitteluun, kehitykseen, testaukseen ja käyttöönottoon. Suunnitteluvaiheen tärkein tavoite on asettaa projektin tavoitteet ja määritellä tarkasti toivotun järjestelmän keskeisimmät toiminnot. Suunnitteluvaiheessa yritetään myös tunnistaa riskejä ja haasteita, jotta niiden mahdollinen vaikutus minimoitaisiin jo projektin alussa. Suunnitteluvaiheenkeskeisimpiätyökaluja ovat esimeriksi Gantt-kaaviot, aikataulut ja resurssienhallinnalliset prosessit ja työkalut.   
   
Kehitysvaiheen aikana toteutetaan suunnitelma ja rakennetaan itse tulospalvelu. Projektimme kohdalla se tarkoittaisi ohjelmointia, tietokantarakenteiden suunnittelua ja itse käyttöliittymän kehittämistä. Tulospalvelun kaltaisessa projektissa olennaisia työkaluja kehitysvaiheen osalta voisivat olla esimerkiksi ohjelmistokielet ja kehitysympäristö.    
   
Seuraavaksi projekti siirtyy testausvaiheeseen, jonka tehtävä on varmistaa tuotetun ohjelmiston toimivuus. Testaamiseen voidaan käyttää joko manuaalisia tai automatisoituja testausmenetelmiä tai vaihtoehtoisesti niiden yhdistelmää. Palvelun laatu ja luotettavuus ovat äärimmäisen tärkeitä ja ne voidaan taata ainoastaan onnistuneella testausvaiheella.   
   
Projektin viimeinen vaihe on sen käyttöönotto. Käyttöönottoon sisältyy yleistä ylläpitoa, koulutusta ja julkaisua. Palvelu on saatavilla käyttäjille, joten sen jatkuva ylläpito jatkuu projektin päättymisestä huolimatta.   
   
Projektimme työvaiheiden määrittely tapahtuu opettajan määrittelemän viikkoaikataulun mukaisesti. Projektin ensimmäisen viikon aikana muodostimme ryhmän ja valitsimme aihealueen kurssityöllemme. Valitsimme aihealueeksemme jalkapallon tulospalvelun ja tapasimme ensimmäisen kerran. Sovimme jo ensimmäisellä viikolla tapaavamme kerran viikossa ja kokoamme yhteen viikon aikana itsenäisesti tehdyn työn ja editoimme lopputyötä yhdessä. Toisella viikolla aloitimme projektimme dokumentoinnin. Ensimmäisellä viikolla tuotimme projektistamme vaatimusmäärittelyn, joka on osana tätä kurssityötä. Tämän lisäksi kirjoitimme työstämme johdannon, joka pohjustaa palvelumme ominaisuudet.    
   
Kolmannella viikolla aloitimme projektisuunnitelman teon. Jaoimme projektisuunnitelman jälleen osioihin, joita jokainen työstää itsenäisesti. Kokoonnumme viikoittain jälleen kokoamaan tekemämme työn yhteen. Viikolla kolme otimme myös käyttöön Trello-työkalun ja loimme omalle projektillemme taulun, johon jokaisella meistä on pääsy. Projektin neljäs viikko on omistettu projektimme arkkitehtuurin sekä moduulien suunnitteluun. Samalla viikolla hoidamme myös projektin saavutettavuuden määrittelyn. Viikolla viisi keskitymme tuotteen hallintaan ja versiointiin. Viimeisellä kuudennella viikolla luomme testaussuunnitelman ja viimeistelemme raportin. Esittelemme työmme myös videomuodossa, jonka nauhoitamme.   
   
Alla myös kuvakaappaus Gantt-kaaviosta, johon olemme määritelleet opettajan määrittelemän viikkoaikataulun toteutuksen ja suunnittelun.

A screenshot of a computer

Description automatically generated

## Projektin tekniikka

Tulospalvelun tekninen ratkaisun keskiössä on varmastikin sen arkkitehtuuri. Sujuvan toiminnan takaamiseksi on tärkeää varmistaa arkkitehtuurin eri komponenttien yhteensopivuus. Tässä projektissa tietokanta, josta saamme pelaajatiedot, ottelutiedot sekä tulokset on ulkopuolinen tietokanta, joka on liitetty palveluumme rajapinnalla. Käytössämme tulisi olla myös jonkinlainen pilvipalvelu, kuten Azure, jotta varmistamme palvelun suorituskyvyn ja skaalautuvuuden. Front-endin näkökulmasta on olennaista hyödyntää toimivaa web-kehitystä, jotta itse käyttöliittymä on mahdollisimman käyttäjäystävällinen.

Itse ohjelman luonti vaatii ohjelmointikielen valinnan, joka voi olla esimerkiksi Python-ohjelmointikieli. Internet-sivuston kannalta on myös hyödyllistä hyödyntää Django Frameworkin kaltaista palvelua, sillä se mahdollistaa nopean kehityksen ja tarjoaa nopeita ratkaisuja yleisimpiin verkkosivujen ylläpitoon liittyviin ongelmiin.    
   
   
Projektissamme tulee huomioida myös palvelumme tietosuojaulottuvuus. Tietokantayhteyksien tulee olla salattuja ja mahdollinen rekisteröityminen verkkosivustolla tulee olla vahvan tunnistautumisen takana. Palvelun tulee myös huomioida tietosuojalainsäädäntö, kuten GDPR, jotta mahdollisten käyttäjien tietoturvallisuus on taattu.

## Projektin dokumentointi

Dokumentoinnin perusperiaate on luoda muistijälki sovituista tai tapahtuneista asioista sellaisessa muodossa, jossa se voidaan saattaa muiden tietouteen tai arkistoida tulevaisuuden tarpeita varten. Dokumentoinnin merkitys huomataan usein vasta, kun tarvittavaa dokumenttia ei löydy tai dokumentin sisältö on vajavainen.

Ihmisen muisti on lyhyt ja muistiin vaikuttavat monet asiat, kuten henkilön historia ja ajattelutapa. Tämä aiheuttaa tilanteita, joissa ihmiset riitautuvat muistaessaan sovitut asiat eri tavoin. Dokumentit esiintyvät näissä tilanteissa sovinnon avaintekijöinä, sillä niistä voidaan tarkistaa asiat, jotka on yhdessä sovittu. Jokainen dokumentti on edistysaskel toimijoiden välisen luottamuksen muodostumiselle, mikä edesauttaa sopimusten syntyä ja yhteistyösuhteiden muodostumista. Dokumentit toimivat todisteina siitä, että asiaan on käytetty aikaa, sitä on mietitty ja siihen suhtaudutaan vakavasti.

Dokumenttien eli tallenteiden avulla voidaan seurata projektien etenemistä ja edistystä. Projektissa tehdyt virheet sekä onnistumiset voidaan nähdä dokumenteista, jolloin niistä voidaan ottaa oppia tulevaisuudessa. Dokumenttien avulla voidaan myös palata työn eri vaiheisiin. Toimintatapojen kehittämisen kannalta on erittäin merkityksellistä dokumentoida omia toimintatapoja, jotta niitä voidaan kehittää tulevaisuutta varten. Tulevien projektien suunnittelussa, oman työn arvioinnissa ja projektin onnistumisen arvioinnissa voidaan hyödyntää projektissa kertyneitä dokumentteja, sillä ne toimivat automaattisena muistijälkenä projektin aikana kertyneistä kokemuksista.

Dokumentointi meidän ryhmämme osalta tapahtui suurimmalta osin Teams video- ja tekstipalavereissa.

A diagram of a company

Description automatically generated

## Projektin seuranta & laadunvarmistus

Projektin seuranta on olennainen osa projektinhallintaa. Projektin seuranta etenee yhdessä projektin toteutuksen kanssa. Seuranta auttaa arvioimaan, onko aikataulun tai resursoinnin muutoksille tarvetta. Muutokset suunnitelmiin ovat lähes väistämättömiä, mutta muutokset voivat johtaa myös hyviin asioihin.

Kaikki muutokset tulisi tehdä suunnitellusti, jotta vältetään projektin laajuuden hallitsematon kasvu. Näin voi käydä, jos projektin laajuutta ei ole määritelty, dokumentoitu tai valvottu riittävästi.

Projektivastaavien tulisi myös varmistaa, että projekti etenee kohti suunnitteluvaiheessa asetettuja tavoitteita.

Projektin seuranta voidaan toteuttaa muutamalla eri tavalla ja eri ryhmien kesken:

Johto- eli ohjausryhmän kokous

* Pari kolme kertaa / kausi (puoli vuotta)
* Päätetään merkittävistä asioista: projektisuunnitelman hyväksyminen, projektin päättäminen jne.
* Projektiryhmä valmistelee tilannekatsauksen (-raportin)

Katselmoinnit

* Yleensä jokaisen vaiheen lopussa
* Tehdään usein iteratiivisesti: ensin kollegan tekemänä, sitten laajempi työryhmä, lopuksi asiantuntija

Projektikokoukset

* N. viikon välein
* Projektin tilanne tehtävätasolla (who does what when)

Laadunvarmistus on yksi viimeisistä ja tärkeimmistä vaiheista kaikissa tuotesuunnitelmissa. Laadunvarmistus, jota käytetään useilla toimialoilla, mukaan lukien ohjelmistokehitys ja rakentaminen, varmistaa, että tuote tai palvelu on laadukkainta, mikä on olennainen prosessi jokaiselle yritykselle.

Laadunvarmistuksella tarkoitetaan digitalisaatiokontekstissa it-toiminnan, (esim. tietojärjestelmät, tietoliikenne) tai laajemmin koko tietohallinnon (prosessit, palvelut) laadunvarmistusta. Laadunvarmistuksella pyritään mittaamaan ja kehittämään laatua. Usein laadunvarmistuksesta käytetty termi on testaus. Tyypillisiä tarjonnan tyyppejä ovat testauspalvelut ja tietohallinnon toiminnan ja palveluiden laadunvarmistuksen konsultointi.

Digitaalinen konsepti menettää merkityksensä, mikäli sen käyttäjä ei pysty sitä käyttämään sellaisella päätelaitteella tai sellaisessa käyttöympäristössä, jossa konseptin on markkinoitu toimivan. Näin ollen ennen kuin markkinaosastoa kehotetaan laatimaan mainos uudesta päätelaiteriippumattomasta digitaalisesta ratkaisusta, on syytä varmistaa, ettei lupaus ole katteeton. Päätelaite määrittelee rajapinnan käytettävyydelle, joten viimeistään kehitysvaiheessa tulisi olla selvyys, millä päätelaitteilla konseptia tullaan käyttämään. Testaus tulee suunnitella sen mukaisesti.

# Arkkitehtuurisuunnittelu

## Moduulit

Palvelun tietokanta toteutetaan Microsoftin tarjoamalla Azure SQL palvelulla. Tietokanta sisältää palvelun tarvitsemat tiedot kuten, otteluiden tulokset, niiden pituuden, niissä osallistuneet pelaajat ja joukkueet. Tietokannassa säilötään myös salatut käyttäjien tiedot. Tietokantaa kutsutaan palvelun ”backend” palvelimen eri moduuleilta. Dataa ei säilötä vaan ”pötkössä” vaan rakennetaan erilaiselle datalle sille sopiva rakenne. Esimerkiksi pelaajien tietoja säilötään pelaajan nimen tai jonkin muun tunnisteen alla, jotta kaikki pelaajan liittyvä tieto on helposti ja nopeasti kerättävissä sitä tarvittaessa.

Esimerkki pelaajan tietorakenteesta:

A close up of a card

Description automatically generated

Palvelin ja sen moduulit toteutetaan C# koodikielellä .NET arkkitehtuurilla. Palvelimen pääasiainen työ on toimia käyttöliittymän ja moduulien tiedon välittäjänä. Palvelin hoitaa kaikki käyttöliittymältä tulevat pyynnöt ja tulkitsee ne. Tämän jälkeen palvelin ohjaa pyynnön oikealle moduulille ja palauttaa moduulilta saadun vastauksen.  Palvelin pyörittää myös kaikki palvelun tarvitsemat moduulit ja hoitaa niiden koordinaation.

Palvelimella on myös tiedon hankinnan hoitava moduuli. Moduuli ensin hakee tulosarkistosta uusimmat tiedot hyödyntäen API-kutsuja, jonka jälkeen se käsittelee saadun datan haluttuun muotoon. Tämän jälkeen moduuli tarkistaa datan mahdollisten virheiden varalta ja korjaa ne, jos mahdollista. Jos dataa ei ole mahdollista korjata, hakee moduuli sen uudelleen tulosarkistolta, jos data on edelleen virheellistä ilmoittaa se siitä palvelun ylläpitäjälle.  Lopuksi moduuli säilöö kaiken käsiteltyyn datan tietokantaan.   
 

Tiedonhakua hoitaa oma moduulinsa. Tämä moduuli hoitaa käyttöliittymältä tulevat dataa koskevat pyynnöt. Moduuli tulkitsee pyynnöstä mitä dataa käyttöliittymä tarvitsee ja hakee sen tietokannasta, muotoilee sen tarvitulla tavalla ja palauttaa sen palvelimen kautta käyttöliittymälle.  Tässä vaiheessa oletetaan, ettei tietokannassa ole virheellistä dataa.

Käyttäjä tietojen käsittely tapahtuu omassa moduulissaan. Tämä moduuli hoitaa käyttäjätietojen salauksen, käyttäjien sisään kirjautumisen ja rekisteröitymisen. Kun käyttäjä luo tilin palveluun, kaikki käyttäjään liittyvä data ohjataan tälle moduulille. Käyttäjän salasana salataan kryptografisesti ennen kuin se lähetetään palvelimelle ja se salataan + suolataan uudelleen ennen kuin se säilötään tietokantaan. Täten käyttäjän salasana ei ole palvelimella ikinä ”plain text” muodossa ja täten vähennetään mahdollisen tietovuodon aiheuttamia vahinkoja. Käyttäjän sähköposti myös salataan, mutta jotta sitä voidaan käyttää, se salataan tavalla, jolla se voidaan muuntaa takaisin teksti muotoon.    
   
Palvelun käyttöliittymä/frontend keskustelee palvelimen kanssa ja palvelin ohjaa käyttöliittymän pyynnöt eri moduuleille. Käyttöliittymän kautta ei pitäisi pystyä kommunikoimaan minkään moduulin kanssa suoraan, kaikkien pyyntöjen tulee kulkea palvelimen kautta.  Käyttöliittymä toteutetaan hyödyntäen .NET Azurea ja C#, HTML ja CSS ohjelmointi kieliä.

Esimerkki palvelimen, sen moduulien ja käyttöliittymän tiedonsiirrosta:

A diagram of a server

Description automatically generated

## Käyttöliittymä

A screenshot of a football match

Description automatically generated

Arkkitehtuurisuunnitelmassa esitellään verkkosivun rakenne, sekä toiminnalliset osat. Suunnitelmassa kerrotaan verkkosivun toiminnasta ja käydään läpi verkkosivun ulkoasu, sekä sen toiminnalliset moduulit. Suunnitelmassa käytetään esimerkkikuvia. Kuvat eivät ole kuvia lopullisesta tuotteesta, vaan ovat suunnitelmassa vain helpottamaan moduulien hahmottamista. Käyttöliittymä rakennetaan käyttämällä HTML ohjelmointikieltä, ja ulkoasu ohjelmoidaan CSS- ohjelmointikielellä. Käyttöliittymän tehtävänä on esittää tieto käyttäjälle, ja vastaanottaa käyttäjän syötteet. Käyttöliittymä koostuu elementeistä, joita voidaan muokata erillään toiminnallisista elementeistä. Tällä tavalla esimerkiksi käyttöliittymän ulkoasua voidaan päivittää ilman että toiminnallisella tasolla tehdään muutoksia. Sivusto toimii yhdellä lehdellä, jolloin sivusto on käytöltään mahdollisimman yksinkertainen.

Sivusto koostuu seuraavista elementeistä:

### Ylätunniste

A close-up of a sign

Description automatically generated

Ylätunniste on sivun ylälaidassa sijaitseva otsikkopalkki. Ylätunniste pitää sisällään sivuston osoitteen, sekä mahdollisuuden kirjautua palveluun. Ylätunnisteessa voi olla taustalla kuva, ja se voi sisältää palvelulle valitun logon. Lisäksi siinä voi olla palvelulle valittu slogan.

Ylätunnisteen tärkein yksittäinen osa on ylätunnisteen oikeassa reunassa oleva kirjautumismoduuli. Moduuli sisältää kentät, jossa käyttäjä voi syöttää kirjautumiseen vaadittavat käyttäjätunnuksen ja salasanan. Kirjautumalla sisään, käyttäjä saa etusivulle tiedot suosikiksi asettamistaan joukkueista.

### Sivupalkki

A screenshot of a phone

Description automatically generated

Verkkosivun vasemmassa laidassa on navigointipalkki, jossa näkyy meneillään olevat ottelut. Palkki kertoo ottelun tilanteen, ja pelatun peliajan. Ottelua klikkaamalla saat avattua ottelun tarkemmat tiedot.

Sivupalkki on moduuli, joka koostuu otsikosta, sekä pienemmistä moduuleista, joiden määrä riippuu milloinkin meneillään olevien otteluiden määrästä. Moduuli päivittyy automaattisesti 1 sekunnin välein, jolloin sivusto poistaa päättyneet ottelut, ja lisää alkaneet ottelut. Ottelumoduuli pitää sisällään tiedon joukkueista, ottelun tilanteesta, sekä pelattujen minuuttien määrän. Moduuli päivittyy automaattisesti 1 sekunnin välein, jolloin sivusto hakee päivitetyt tiedot otteluista.

### Pääsivu

A screenshot of a football match

Description automatically generated

Sivuston tärkein sisältö näkyy sivuston keskellä/oikealla. Tässä näkyy käyttäjän valitseman ottelun tarkemmat tiedot. Tarkempia tietoja on esimerkiksi maalintekijät ja annetut kortit. Pääsivu on moduuli, joka koostuu useammasta pienemmästä moduulista. Pääsivu päivittyy käyttäjän valitessa sivupalkista ottelun, jota haluaa seurata. Lisäksi pääsivu päivittyy automaattisesti 1 sekunnin välein. Mikäli käyttäjä on kirjautunut palveluun, ja on valinnut suosikkijoukkueen, pääsivu sisältää moduulin, joka kertoo suosikkijoukkueen edellisen ottelun lopputuloksen. Päättyneet ottelut- moduuli ei päivity automaattisesti.

## Saavutettavuus

Saavutettavuus verkkosivustoissa tai applikaatioissa tarkoittaa sitä, että ne on suunniteltu ja toteutettu, niin että kaikenlaiset käyttäjät voivat helposti käyttää ja ymmärtää niitä. Saavutettavuus varmistaa, että palvelut ovat käytettävissä myös näkö- ja kuulovammaisten sekä liikuntarajoitteisten henkilöiden toimesta.   
   
Saavutettavuuden varmistaminen on olennainen osa palvelun inklusiivista ja nykypäiväistä suunnittelua. Onnistunut saavutettavuus vahvistaa käyttäjäkokemusta, laajentaa käyttäjäkuntaa, parantaa hakukonenäkyvyyttä ja vahvistaa yrityksen brändiä. Saavutettavuus voi olla myös lainsäädännöllinen vaatimus. Suomessa jokainen julkishallinnon organisaatio on velvollinen tarjoamaan palvelunsa esteettömästi saavutettavuuden nimissä.

### Saavutettavuuden toteutus

Saavutettavuuden varmistamiseksi on tärkeää varmistaa lähdekoodin virheettömyys ja loogisuus. Maailmanlaajuisesti käytetään saavutettavuusstandardia Web Content Accessibility Guidelines (WCAG) 2.1. WCAG perustuu neljään periaatteeseen: havaittavuus, ymmärrettävyys, toimivuus ja saavutettavuus. Näiden tehtävä on ohjata suunnittelijoita ja kehittäjiä luomaan saavutettavia palveluita.

### Havaittavuus

Jalkapallon tulospalvelun saavutettavuus ei kuulu lakisääteisen velvoitteen piiriin, mutta sen varmistaminen perustasolla on silti palvelulle edullinen toimintaperiaate. Tulospalvelussa on esimerkiksi tärkeää tarjota tuloksien tiedot monimuotoisesti eli esimerkiksi kuva- sekä tekstimuodossa.

A screenshot of a football match

Description automatically generated

Kuva havainnollistaa kuinka seurojen logojen lisäksi tulos on saatavissa myös tekstimuodossa. Olennainen osa tulospalvelua on sen luettavuus. Sisällön tulee myös edetä loogisesti sekä sarkaimella että ruudunlukijalla. Käyttöliittymän visuaalisten elementtien tulee miellyttää silmää, joten sen värisuunnittelussa tulee huomioida värisokeuden mahdollisuus. Vaihtoehtoisesti voimme tarjota käyttäjille myös mahdollisuuden ALT-teksteihin, jolloin sivuston kuvien tulkinta onnistuu myös audion perusteella ja lisää sivuston saavutettavuutta.

### Hallittavuus

Hallittavuus jalkapallon tulospalvelussa tarkoittaa myös sen yksilöintimahdollisuuksia. Käyttäjät kykenevät filtteröimään ottelutuloksia esimerkiksi joukkueen, pelaajan, maan ja ottelun perusteella. Palvelu tulee luoda niin, että sen sivuotsikot, otsikot ja painikkeet ovat helppokäyttöisiä ja selkeitä. Tulospalvelulla on ehdottomasti oltava myös hakutoiminto, jolla käyttäjä voi hakea ottelutuloksia vapaalla sanahaulla. Kyseisen sanahaun voi toteuttaa myös äänikomennoin, joka lisää jälleen palvelun saavutettavuutta.

### Ymmärrettävyys

Ymmärrettävyyden avaintekijä on selkeä, yksinkertainen ja johdonmukainen esityskieli. Optimaalisessa tilanteessa verkkosivu sisältäisi tarvittavan tiedon jokaiselle, myös niille, jotka eivät ole syvällä jalkapallon maailmassa. Tätä voi vahvistaa esimerkiksi selittämällä alan termistöä auki verkkosivuilla. Graafisten tuloselementtien tulee olla helposti luettavia ja ymmärrettäviä ensisilmäyksellä. Tulospalveluissa aikaleimat ovat äärimmäisen hyödyllisiä, joten niiden esilletuonti tuloksien yhteydessä lisää kokonaisuuden ymmärrettävyyttä.

A screenshot of a video game

Description automatically generated

### Toimintavarmuus

Toimintavarmuuden varmistamiseksi tulee ensin varmistaa standardin mukainen www-tekniikka, josta aikaisemmin mainitsimme. Internetissä on tarjolla erilaisia validaattoreita, joiden tehtävä on varmistaa verkkosivujen koodin standardinmukaisuus, jota käytämme myös tulospalvelumme kehityksessä. Yksi mainittavista palveluista olkoon [www.w3.org](http://www.w3.org/). Toimintavarmuudelle on myös olennaista vakaa suorituskyky, joka edellyttää skaalautuvaa teknistä infrastruktuuria. Ongelmatilanteisiin tulee luoda toimivat prosessit, jotka ylläpitävät eri osien automaatioita ja varmistavat verkkosivujen nopean korjaamisen.

# Testaussuunnitelma

## Johdanto

Tämä dokumentti on testaussuunitelma ohjelmistosuunnittelun kurssityön tulospalvelujärjestelmästä. Projektin aikana on tarkoitus kuvata ja suunnitella tulospalvelujärjestelmä, joka esitellään tässä dokumentissa. Testauksen tavoite on löytää mahdollisia virheitä ohjelmistosta tai prosesseista sekä varmistaa, että ohjelma täyttää sille asetetut järjestelmälliset ja laadulliset vaatimukset. Ohjelmiston on toimittava vaaditulla tavalla ja tämän dokumentoinnin rooli on kuvata testauksen kulku selkeästi, systemaattisesti ja järjestelmällisesti.

## Sanasto

*SEO-testi*

SEO-testi (Search Engine Optimization) on testi tai arviointi, jolla arvioidaan verkkosivuston hakukoneoptimaatiota. Tämän tarkoituksena on lisätä verkkosivuston näkyvyyttä eri hakukoneissa, kuten Googlessa tai Yahoossa. Testin perusteella palvelun näkyvyyttä on helpompi kehittää. SEO-testin kautta havaittavia tekijöitä ovat esimerkiksi:

* Sisältö
* Linkkien laatu
* Käyttäjäkokemus
* Avainsanat

*SSL Labs*

SSL Labs on avoimen lähdekoodin työkalu, joka tarjoaa verkkopalvelujen SSL/TLS-tietoturvan arvioinnin. SSL Labs on palvelu, jolla tarkastetaan minkä tahansa verkkopalvelun SSL/TLS-tietoturvan tila. Palvelu suorittaa tarkastuksen, jonka päätteeksi saat sivustostasi arvosanan tietoturvan suhteen.

*Screaming Labs*

Screaming Frog on ohjelmisto, joka on suunniteltu verkkosivustojen hakukoneoptimoinnin tueksi. Ohjelmisto suorittaa analyysin hakukoneoptimoinnista tarkastelemalla sivuston sisältöä, linkkejä ja rakennetta. Palvelu tekee esimerkiksi URL-skannausta, sivukartan luomista ja linkkien tarkistusta.

*WAVE*

WAVE (Web Accessibility Evaluation Tool) on verkkopalvelu, jonka avulla voi arvioida verkkosivujen saavutettavuutta ja esteettömyyttä. WAVE sisältää sivuston skannaukseen, raportointiin ja sisällön tarkastukseen liittyviä työkaluja.

*Google PageSpeed Insights*

Google PageSpeed Insights on Googlen tarjoama työkalu, jonka avulla voidaan arvioida verkkosivuston suorituskapasiteettia ja löytää keinoja sivustonopeuden parantamiseksi. Työkalu osaa arvioida sivuston mobiili- ja työpöytäversioita. Oleellisia mittauskohteita ovat sivunopeus, optimoinnit, käyttäjäkokemus yleisesti ja renderöintiajat.

*Black Box & White Box*

Black Box & White Box -menetelmät ovat erilaisia lähestymistapoja järjestelmien tai prosessien testaamiseen tai suunnitteluun. Black Box -testaus keskittyy arvioimaan järjestelmää ulkopuolelta, tietämättä sen sisäisiä prosesseja, kun taas White Box –testauksessa järjestelmää tarkastellaan sisältä päin ja testaaja ymmärtää sovelluksen rakenteen.

## Testausympäristö

Palvelu testataan suurimmaksi osin ”Black Box” menetelmällä ja osa sen moduuleista testataan ”White Box” menetelmällä. Kaikki moduulien vaatimat yksikkö testit tehdään Microsoftin Visual Studiossa tarjoamilla Unit Testeillä. Toiminallinen testaus toteutetaan eri laiteilla ja siihen olemassa olevilla palveluilla.

### Moduulitestaus

Palvelun tärkeimmille moduuleille rakennetaan testit, kun ne valmistuvat tai kun kaikki moduulin toimintaan vaaditut komponentit on saatu valmiiksi Jos jokin moduuli valmistuu ennen muita sen vaatimia moduuleita, voidaan ne korvata tyngillä. Moduuleja rakentaessa mietitään myös mahdollisia testitapauksia ja kirjataan ne muistiin.

### Testattavat moduulit

Yksikkötestit rakennettaan datan haku ja käyttäjän dataa käsitelevälle moduulille. Datan hankinta moduulilla ei testata kokonaan koska se vaati kolmannen osapuolen data arkiston toiminnan. Sen virheiden käsittely kuitenkin testataan.

### Hyväksymiskriteerit

Testitapaukset suunnitellaan aikaisemmin tehtyjen muistiinpanojen avulla ja miettimällä jokaisen funktion syötteiden mahdollisia arvoalueita. Testit pitää suunnitella siten, että niiden antamalla syötteillä koko funktion toimivuus tulee testattua. Moduulitestaus todetaan hyväksytyksi, jos kaikki testitapaukset suoritetaan hyväksytysti. Testien tulisi kattaa vähintään 90 % testattavien moduulien koodiriveistä.

## Testauksen organisointi & raportointi

Testit kirjoittavat moduulin kirjoittaneet henkilöt. Löydetyt virheet myös korjaavat moduulin alkuperäiset tekijät. Testauksesta tehdään testiraportti, johon kirjataan kaikkien testien tulokset.

## Testattavat toiminnot

1. Palvelun kotisivun toiminta
2. Käyttäjän rekisteröityminen
3. Käyttäjän sisäänkirjautuminen

## Toimintojen testitapaukset

### Testitapaus: Käyttäjän rekisteröityminen ja kirjautuminen

**Kuvaus:**  
  
Tämä testitapaus tarkistaa, että käyttäjät voivat rekisteröityä palveluun.

**Esiehdot:**

Palvelu on avattu tuetulla laitteella

## Testiaskeleet:

* Siirry rekisteröitymissivulle.
* Syötä toivottu käyttäjänimi, salasana ja sähköpostiosoite rekisteröitymissivulla.
* Klikkaa "Rekisteröidy" -painiketta.
* Tarkista että voit kirjautua sisään juuri luoduilla tunnuksilla.

**Odotetut tulokset:**

Kirjautuminen onnistuu käyttäen juuri rekisteröityjä tunnuksia. Järjestelmä näyttää ilmoituksen onnistuneesta rekisteröitymisestä. Järjestelmä ilmoittaa, jos käyttäjä yrittää rekisteröidä jo varatulla sähköpostiosoitteella tai käyttäjänimellä.

**Hylkäämiskriteerit:**

Kirjautuminen epäonnistuu käyttäen juuri rekisteröityjä tunnuksia. Järjestelmä ei näytä ilmoitusta onnistuneesta rekisteröitymisestä. Järjestelmä antaa käyttäjän rekisteröityä jo olemassa rekisteröidyllä käyttäjänimellä tai sähköpostiosoitteella.

### Testitapaus: Palvelun etusivun toiminta

**Kuvaus:**

Tämä testitapaus tarkistaa, palvelun etusivu toimii tarkoitetusti.

**Esiehdot:**

Palvelu on avattu tuetulla laitteella.

**Testiaskeleet:**

1. Tarkista että kaikki etusivun elementit latautuivat oikein.
2. Tarkista että kaikki etusivulla olevat tiedot ovat oikein.
3. Kokeile että kaikki etusivulla olevat painikkeet toimivat oikein.

**Odotetut tulokset:**

Etusivu lataantuu oikein ja kaikki elementit latasivat tyylinsä oikein. Kaikki etusivulla olevat tiedot vastaavat oletettua. Kaikki etusivulla olevat painikkeet tekevät mitä niiden pitää.

**Hylkäämiskriteerit:**

Etusivu tai jokin sen elementeistä ei lataudu oikein. Jokin etusivulla esitetyistä tiedoista on väärin. Jokin etusivulla olveista painikkeista ei toimi suunnitellusti.

## Testausraportti

Järjestelmälle suoritettiin testaus, jossa erityistä huomiota kiinnitettiin vaatimusmäärittelyssä esille nousseisiin seikkoihin. Testissä testattiin palvelun helppokäyttöisyyttä, saavutettavuutta, nopeutta, ja yhteensopivuutta eri selainten kanssa. Lisäksi palvelua testattiin erilaisilla kuormilla, jolloin saatiin selville palvelun toimivuus, kun palvelussa on suuri määrä käyttäjiä yhtä aikaa. Palvelu testattiin myös tietoturva huomioiden.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Suoritetut testit | Havainnot | Ipad | OnePlus Nord | Desktop Firefox | Desktop Chrome |
| T1 | Ei havaittuja ongelmia | OK | OK | OK | OK |
| T2 | Ei havaittuja ongelmia | OK | OK | OK | OK |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| T3 | HYVÄKSYTTY |  |  |  |  |
| T4 | SSL HYVÄKSYTTY |  |  |  |  |
| T5 | SF HYVÄKSYTTY |  |  |  |  |
| T6 | HYVÄKSYTTY |  |  |  |  |
| T7 | WAVE HYVÄKSYTTY |  |  |  |  |
|  |  |  |  |  |  |

### Testien kuvaus

**T1: Yleinen toiminnallisuustesti**

Testataan, että perustoiminnot toimivat oikein, kuten navigointi, linkit, lomakkeiden lähettäminen ja hakutoiminto.

**T2: Responsiivisuustesti**

Testataan, että sivusto näyttää hyvältä eri laitteilla ja näytöillä. Testissä käytettiin Apple IPad tablettia (10.2 tuumaa), sekä Oneplus Nord puhelinta. Lisäksi testissä käytettiin desktop tietokonetta.

**T3: Suorituskykytesti**

Testataan palvelun nopeutta käyttämällä Google PageSpeed Insights palvelua. Palvelu arvio sivuston latausaikaa ja suorituskykyä.

**T4:  Turvallisuustesti**

Testataan sivuston tietoturvataso ja varmistetaan, että sivusto noudattaa haluttuja tietosuoja-asetuksia. Testissä tarkastettiin sivuston lähdekoodi manuaalisesti, varmistettiin SSL- sertifikaatin toiminta SSL Labs työkalulla.

**T5: SEO-testi**

Testataan sivuston hakukonenäkyvyyttä ja optimoimiseen liittyviä asioita. Testissä tehtiin hakusanatutkimus, sekä testattiin hakukoneystävällisyys Screaming Frog palvelulla.

**T6: Sosiaalisen median jakamistesti**

Testataan, että sivusto toimii oikein, kun käyttäjä jakaa sisältöä sosiaalisen median palveluissa. Testissä jaettiin linkkejä palveluun, ja varmistettiin linkkien toimivuus.

**T7: Saavutettavuustesti**

Testataan sivuston saavutettavuutta, ja varmistetaan, että sivusto on kokonaisuudessaan saavutettavissa kaikille mahdollisille käyttäjäryhmille. Testissä käytettiin WAVE (Web Accessibility Evaluation Tool) työkalua, joka testaa saavutettavuuden automaattisesti.

## Suunnitteluperiaatteet

Ohjelmiston suunnittelussa on tiettyjä keskeisiä periaatteita, joita noudattamalla saadaan suunnittelu ja toteutusprosessista sujuvampaa. Joitain keskeisimpiä periaatteita ovat muun muassa:

1. **Yksinkertaisuus ja suoraviivaisuus**

Ohjelmistoa suunnitellessa on tärkeää pitää ratkaisut mahdollisimman yksinkertaisina. Usein monimutkaisiinkin ongelmiin on löydettävissä yksinkertaisia ratkaisuja. Suunniteltu ohjelmisto voi asettaa joitain rajoitteita ja vaatimuksia, mutta näiden ympärillä on pyrittävä suoraviivaiseen ja yksinkertaiseen suunnitteluun. On myös tärkeää jättää tilaa tulevaisuudessa ilmeneville ongelmille (kapselointi).

1. **Osittaminen ja lokaalisuus**

Tavoitteena on jakaa iso kokonaisuus pieniin osiin toiminnallisuuden perusteella. Osat voidaan jakaa vielä pienempiin osiin tarvittaessa. Osittaminen on hyvä tapa “jakaa” suunnittelutaakkaa ja auttaa pitämään suunnittelun yksinkertaisena ja suoraviivaisena. Osittaminen voidaan toteuttaa esimerkiksi hierarkkisesti, jolloin alemman tason moduulit ikään kuin palvelevat ylemmän tason moduuleja. Lokaalisuus suunnittelussa tarkoittaa sitä, että järjestelmän komponentit kapseloidaan mahdollisimman hyvin. Kapselointi pitää huolen, että komponentit ovat mahdollisimman vähän riippuvaisia toisistaan joka. Komponenttien ollessa “itsenäisiä” helpottuu muutosten teko sillä muutokset yhdessä komponentissa eivät vaikuta toisen komponentin toimintaan. Hyvin toteutettu lokalisointi mahdollistaa ohjelmiston eri osien joustavan tuottamisen ja testaamisen.

1. **Abstrahointi**

Abstrahointi on järjestelmän tarkastelua monelta eri tasolta. Tarkastelu aloitetaan karkean yleiskuvan luonnilla ja aina seuraavalle tasolle siirryttäessä lisätään yksityiskohtaisuutta. Abstraktiossa pyritään tuomaan esille oleelliset asiat ja koteloimaan tarpeettomat yksityiskohdat abstraktion sisään. Abstraktiossa näkyy käyttäjälle vain rajapinta, sen sisään koteloitu toteutus on käyttäjälle tuntematon mikä auttaa ymmärrettävyyden ja muunneltavuuden kanssa

A diagram of a computer application

Description automatically generated

1. **Suunnittelufilosofia**

Suunnittelufilosofian yhdenmukaistaminen on tärkeä osa onnistunutta suunnittelu- ja toteutusprosessia. Yhtenäinen filosofia kiteyttää periaatteet ja rakenteet, joiden ajatellaan pysyvän muuttumattomina koko ohjelmiston elinkaaren ajan. Arkkitehtuuri ei siis ole vain pelkkiä moduuleita ja niiden välisiä yhteyksiä. Hyvä suunnittelufilosofia antaa kehittäjälle mallin siitä, miten uusi ominaisuus tulisi toteuttaa ja ylläpitäjälle tietoa, minkä tapaista ratkaisua on etsittävä ja mihin muutokset olisi hyvä kohdistaa.

Hyvälle arkkitehtuurille on ominaista, että jos jotain asiaa ei tiedä, se on pääteltävissä suunnittelu- ja toteutusfilosofian perusteella.

A diagram of a process

Description automatically generated

## Tuotteenhallinta

Tuotteenhallinta koostuu komponenteista niiden konfiguraatioista ja niiden toimintatavoista.

Komponentit ja konfiguraatiot koostuvat Versioinnista, Identifioinnista, Tuottamisesta ja muutosten hallinnasta.

**Versiointi:**

Versio on “valmis” hallinta-alkio, johon ei enää tehdä muutoksia. Hallinta-alkioon tehdyt muutokset julkaistaan kokonaan omana versionaan. Tuotteesta voidaan valmistaa monia eri versioita esimerkiksi eri asiakkaiden tarpeiden mukaan. Kaikilla tuote-alkioilla on oma versionsa. Standardin mukainen versionumerointi on 1–4 numeroinen (1–1.1.1.1). Ensimmäisen tason muuttaminen esimerkiksi muotoon V1.0 -> V2.0 tarkoittaa että tuote on kokenut suuria muutoksia. Toisen tason muuttaminen esimerkiksi muotoon V1.1 tarkoittaa toiminnallisia muutoksia. Kolmannen tason muuttaminen esimerkiksi muotoon V1.0.1 kertoo virheiden korjaamisesta (bug fixes). Neljäs taso on tuotekehityksen välitallennuksia varten. Tulospalvelumme toimii tällä hetkellä versiossa 1.0 joka on ensimmäinen tuottamamme versio tästä tuotteesta, joten vanhoihin versioihin ei tällä hetkellä mahdollista päästä käsiksi, mutta tulevaisuudessa version päivittyessä vanhat versiot varmuuskopioidaan Gitiin josta ne ovat helposti ladattavissa tarpeen tullen.   
   
**Identifiointi:**

Tuotteen identifiointi on tällä hetkellä yksinkertaista, sillä tuotteestamme on olemassa vain yksi versio, mutta tulevaisuudessa identifiointi tulee hoitamaan tehtävään koulutettu tuotteenhallintaryhmä (CBB).

**Tuottaminen:**

Tulospalvelun tuottaminen tapahtuu jo aikaisemmin vaatimusmäärittelyssä mainittuja ohjelmistoja käyttäen. Uusien konfiguraatioiden tuottamiseen ja hallintaan käytetään muun muassa skriptausmenetelmiä.

**Muutosten hallinta:**

Muutosten hallinta hoidetaan aina tilanteen vaativalla tavalla. Jos kyseessä on “bugi”, aloitetaan korjaaminen selvittämällä, millainen konfiguraatio on käytössä ja missä yhteydessä ongelma syntyi. Tätä ongelmaa varten rakennetaan vastaavanlainen konfiguraatio, jota käytetään ongelman uudelleen luontiin ja sitä kautta ongelman ratkaisuun. Ongelman löytyessä ryhdytään korjaustoimenpiteisiin, joka johtaa ohjelmiston uusiin versioihin ja modifikaatioihin. Komponentteja muokattaessa on tärkeää selvittää miten yhden komponentin muokkaus vaikuttaa muihin ohjelman komponentteihin. Täytyy myös selvittää, onko ongelmia aiheuttaneella konfiguraatiolla muita käyttäjiä, jotka eivät ole vielä törmänneet samaan ongelmaan ja opastaa heitä esimerkiksi päivittämään ohjelmisto uuteen versioon. Tulospalvelumme tapauksessa komponenttien kapselointi on hoidettu ammattimaisesti ja yhden komponentin muokkauksella on minimaaliset vaikutukset muihin komponentteihin. Tulospalvelun tapauksessa komponenttien muokkaus hoidetaan Git:tiä hyödyntämällä. Git mahdollistaa projektien kloonauksen ja monien osallistujien samanaikaisen työskentelyn.

## 

### 

## 

## 

### 

### 