Ampumapäiväkirja

Ampumistulosten kirjaussovellus

Hit & Huti Oy

INTIM23A6

Inka Kaalikoski  
Jiska Laaksovirta  
Eveliina Tuomioja  
Katja Venäläinen

# Vaatimusmäärittely

## 1.Johdanto

### 1.1 Tarkoitus ja kattavuus

Sovellukseen voidaan syöttää ampujien määrän, ammuttavien laukausten määrä per ammuntakierros, laskea ammuttujen laukausten tulos, ja tallentaa ne johonkin arvoon, mihin voi palata.

Vaatimusmäärittelydokumentin tarkoituksena on kuvata ohjelmiston pääpiirteet ja tärkeimmät toiminnot. Vaatimusmäärittelydokumentti toimii sopimuksena asiakkaan ja projektiryhmän välillä. Oleellista on, että molemmat osapuolet ymmärtävät mitkä ovat projektin tavoitteet.

Määrittele selkeästi, miksi uutta ohjelmistoa tarvitaan. Mitkä ovat projektin tavoitteet ja taustalla olevat ongelmat tai tarpeet, jotka ohjelmiston tulisi ratkaista? Listaa keskeiset liiketoiminnalliset tai toiminnalliset haasteet, joihin ohjelmisto pyrkii vastaamaan. Tarkenna, miten ohjelmisto parantaa nykytilaa tai tukee organisaation tavoitteita.

Määrittele, mitä ohjelmiston tulee tehdä ja mitä ei. Rajaa projektin kattavuus niin, että se on hallittavissa ja saavutettavissa. Kuvaile keskeiset ominaisuudet, toiminnot ja vaatimukset, jotka ohjelmistolta odotetaan. Käytä konkreettisia esimerkkejä, kun mahdollista. Tee ero välttämättömien ja toivottavien ominaisuuksien välillä. Tämä auttaa priorisoimaan kehitystyötä.

### 1.2 Tuote ja ympäristö

Ohjelmaan tallennetaan ampujien nimet, kierrokset, laukausten määrä, pisteet ja aseen numero. Ohjelmaan myös tallennetaan tiedot ampumaradasta ja etäisyydestä. Ohjelmaa on tarkoitus käyttää ampumaradalla, maasto-olosuhteissa.

Ohjelmistoa on tarkoitus käyttää tietokoneella, jolla yksi, tai useampi henkilö tallentaa tiedot järjestelmään.

Tee lyhyt, mutta informatiivinen kuvaus itse ohjelmistosta. Määritä, mikä sen tarkoitus on ja mitä se tekee. Kuvaa ohjelmisto, sen päätoiminnot ja lopullinen tavoite.

Kuvaillaan, missä ympäristössä ohjelmisto tulee toimimaan. Tämä sisältää käyttöjärjestelmät, laitteistovaatimukset ja mahdollisesti muut ohjelmistot tai järjestelmät, joiden kanssa ohjelmiston on oltava yhteensopiva.

### 1.3 Määritelmät, termit ja lyhenteet

Tämä osio selittää dokumentissa käytettyjä määritelmiä, avaa keskeisiä käsitteitä tai termejä, jotta kaikki osapuolet ymmärtävät ne samalla tavalla. Tämä myös varmistaa saman kielenkäytön kaikkien projektin osapuolten kesken. Tässä osiossa annetaan myös selitykset dokumentissa käytetyille lyhenteille, jotta lukija voi ymmärtää niiden merkityksen.

### 1.4 Viitteet

Laaditaan tarvittaessa myös tietosuojaseloste GDPR varten.

Tässä kohdassa luetellaan kaikki dokumentin ulkopuoliset resurssit, kuten standardit, tekniset ohjeet tai muut viitteet, joita vaatimusmäärittely käyttää tai joihin se viittaa.

### 1.5 Yleiskatsaus dokumenttiin

Lyhyt yhteenveto vaatimusmäärittelyn rakenteesta ja sisällöstä. Tämä auttaa lukijaa hahmottamaan nopeasti, mitä kukin osa-alue sisältää ja miten dokumentti on organisoitu. Esimerkki: "Dokumentti jakautuu käyttäjätarpeisiin, toiminnallisuuksiin, ei-toiminnallisiin vaatimuksiin ja teknisiin vaatimuksiin."

2. Yleiskuvaus

2.1 Ympäristö

Tässä kohdassa tarkennetaan ympäristöä, missä ohjelmisto toimii. Se sisältää tietoa käyttöjärjestelmistä, laitteistovaatimuksista ja muista tekijöistä, jotka vaikuttavat ohjelmiston suorituskykyyn ja toimintaan. Kerro millaisissa tilanteissa ja laitteilla ohjelmiston on toimittava. Huomioi mahdolliset rajoitteet tietyissä ympäristöissä.

### 2.2 Toiminta

Tämä osio tarjoaa yleisen kuvauksen ohjelmiston toiminnoista ja niiden tarkoituksesta. Se ei mene syvälle yksityiskohtiin, vaan antaa lukijalle ymmärryksen siitä, mitä ohjelmisto tekee. Kuvaa tekstissä, miten käyttäjät vuorovaikuttavat ohjelmiston kanssa sekä tunnista keskeiset prosessit ja niiden väliset riippuvuudet.

### 2.3 Käyttäjät

Ohjelmistoa tulee käyttämään ampumaradan toimihenkilö tai itse harrastaja.

Määrittelee ne henkilöt tai roolit, jotka käyttävät ohjelmistoa. Kuvataan käyttäjien tärkeimmät tarpeet ja vaatimukset. Kenelle ratkaisua/palvelua kehitetään? Kannattaa nostaa esiin lyhyesti mahdolliset loppukäyttäjä ja oleellisiin palvelusta hyötyviin sidosryhmät.

### 2.4 Yleiset rajoitteet

Ohjelmiston tuottamisen aikataulu on todella tiukka. Sovellus ohjelmoidaan annetussa aikarajassa käyttökuntoon, mutta sen jatkokehitys on tulevaisuudessa tarpeen.

Listaa tähän asioita, jotka todetaan heti alkuvaiheessa, asiat, toiminnot ja piirteet, joita ei tulla toteuttamaan. Tämä auttaa molempia osapuolia tarkistamaan, mitä projekti pitää sisällään ja mitä projektiin ei kuulu. Mainitaan myös yleiset rajoitteet, kuten aikataulu, resurssit, budjetti ja muut tekijät, jotka voivat vaikuttaa ohjelmiston kehitykseen ja käyttöön.

Eri ohjelmistojen/palvelujen toteutusta ja käyttöä ohjaavat usein lait ja säädökset. Näiden edellyttämät vaatimukset kirjataan vaatimusmäärittelyyn rajauksina.

### 2.5 Oletukset ja riippuvuudet

Tässä osiossa listataan ne oletukset, joita tehdään ohjelmiston suunnittelussa ja kehityksessä. Näitä voivat olla esimerkiksi tietyt tekniset oletukset tai odotukset projektin aikataulusta. Esimerkiksi: "Oletamme, että käyttäjät tulevat käyttämään moderneja selaimia ja päivittävät ne säännöllisesti."

Riippuvuuksina kuvaillaan ne ulkoiset tekijät, joihin ohjelmisto on riippuvainen. Tämä voi sisältää riippuvuudet muiden järjestelmien, laitteistojen tai kolmansien osapuolien kanssa.

## 3. Tiedot ja tietokanta

Kuvaa ohjelmiston käyttämät tiedot, niiden rakenteen, sisällön, käsitteet, käyttöintensiteetin, kapasiteettivaatimukset ja tiedostot. Tiedoilla voi olla eri rakenne ohjelman eri osissa, esimerkiksi tietokannassa ja varsinaisessa ohjelmassa.

### 3.1 Tietosisältö

Tässä kohdassa pitäisi esittää ohjelmiston käyttämien tietojen looginen malli, joka kuvaa tietojen väliset suhteet ja rajoitteet. Tietosisältö voidaan esittää esimerkiksi tietokantakaaviona, luokkakaaviona tai XML-kaaviona.

### 3.1 Käsitteet omiin alakohtiinsa

Käsitteet omiin alakohtiinsa: tässä kohdassa pitäisi määritellä ohjelmiston käyttämät keskeiset käsitteet ja termit, jotka liittyvät tietosisältöön. Käsitteet pitäisi selittää selkeästi ja yksiselitteisesti. Käsitteiden avulla voidaan viestiä ohjelman käyttäjien ja tekijöiden välillä samoilla termeillä.

### 3.2 Käyttöintensiteetti

Tässä kohdassa pitäisi arvioida kuinka usein ja paljon tietoja luetaan, kirjoitetaan, muokataan tai poistetaan. Käyttöintensiteetti vaikuttaa ohjelmiston suorituskykyyn ja sitä kautta vaadittavan laitteiston, tietoturvan ja tietokantojen hallinnan suorituskykyvaatimuksiin. Käyttöintensiteetti voidaan esittää esimerkiksi taulukkona, kaaviona tai kuvaajana.

#### Esimerkki

Kuinka usein laserkeilaustieto välitetään kulkijan ohjausjärjestelmään ja lokitustiedostoihin.

### 3.3 Kapasiteettivaatimukset

Tässä kohdassa pitäisi arvioida ohjelmiston tietojen kapasiteettivaatimuksia, eli sitä, kuinka paljon tietoa ohjelmisto tarvitsee tallentaakseen ja käsitelläkseen. Kapasiteettivaatimukset vaikuttavat ohjelmiston tilantarpeeseen, muistinkäyttöön ja tietokannan koonhallintaan. Kapasiteettivaatimukset voidaan esittää esimerkiksi taulukkona, kaaviona tai kuvaajana.

#### Esimerkki

Jokainen tuotantolinjan tuotteiden tarkastuskuva säilytetään reklamaatioiden varalta vuoden ajan.

### 3.4 Tiedostot ja asetustiedostot

Tässä kohdassa pitäisi kuvata ohjelmiston käyttämät tiedostot ja asetustiedostot, niiden sijainnit, nimet, formaatit, sisällöt ja käyttötarkoitukset. Tiedostot ja asetustiedostot ovat osa ohjelmiston välttämätöntä tietosisältöä. Tietoja voidaan esimerkiksi tarvita, jotta ohjelma voidaan asentaa ja käynnistää ensimmäistä kertaa. Asetustiedot eivät välttämättä ole osa tietokantaa. Tiedostot ja asetustiedostot voidaan esittää esimerkiksi listana, taulukkona tai kaaviona.

## 4. Toiminnot

Kuvaa ohjelmiston käyttötapausten toiminnot, niiden tarkoitukset, syötteet, tulokset, virhetilanteet ja käyttöliittymät.

### 4.1 Kukin toiminto omaan alakohtaansa

Tässä kohdassa esitellään ohjelmiston käyttötapaukset, ohjelmiston käyttäjän näkökulmasta. Jokainen käyttötapaus kuvataan omassa alakohdassaan, jossa pitäisi mainita toiminnon nimi, tarkoitus, syötteet, tulokset, virhetilanteet ja käyttöliittymät. Kuvaa myös toimintojen yhteydet ohjelmiston tietosisältöön. Toimintoja voidaan esittää esimerkiksi kuvauksina, kaavioina, kuvina tai pseudokoodina.

## 5 Ulkoiset liittymät

### 5.1 Laitteistoliittymät

Tämä osa kuvaa, mitä laitteistoja tai laitteita sovellus käyttää tai vuorovaikuttaa. Se voi sisältää esimerkiksi:

Minkälaisia fyysisiä laitteita sovellus käyttää, kuten kameraa, mikrofonia, GPS-paikannusta tai muita antureita.

Laitteiston vaatimukset, kuten käyttöjärjestelmäversion, laitteiston tuen ja resurssien (muisti, suoritin) vaatimukset.

### 5.2 Ohjelmistoliittymät

Tämä osa kuvaa, miten sovellus integroituu muihin ohjelmistoihin tai palveluihin. Se voi sisältää esimerkiksi:

API (Application Programming Interface) -rajapintojen kuvaamisen, joiden kautta sovellus voi kommunikoida muiden ohjelmistojen tai palveluiden kanssa.

Kolmannen osapuolen kirjastojen ja komponenttien käytön, jos sovellus käyttää niitä.

### 5.3 Tietoliikenneliittymät

Tämä osa kuvaa, miten sovellus kommunikoi tietoverkon kanssa. Se voi sisältää esimerkiksi:

Kuinka sovellus käyttää verkkoyhteyksiä, kuten WLANia, mobiilidataa tai Bluetoothia.

Tietoliikenneprotokollat ja -standardit, joita sovellus käyttää tiedonsiirtoon ja vuorovaikutukseen muiden verkkopalveluiden kanssa.

Tietoturvaan ja salaukseen liittyvät vaatimukset, jotka liittyvät tietoliikenteeseen.

## 6 Muut ominaisuudet

### 6.1 Suorituskyky ja vasteajat

Suorituskyky: Miten tehokkaasti ohjelmisto suorittaa tehtäviään, esimerkiksi resurssien käytön optimointi.

Vasteajat: Kuinka nopeasti ohjelmisto reagoi käyttäjän syötteisiin.

### 6.2 Saavutettavuus

Saavutettavuus: Ohjelmiston jatkuva saatavuus käyttäjille.

Toipuminen: Kyky palautua mahdollisista häiriöistä tai vioista.

Turvallisuus: Ohjelmiston suojaaminen haitallisilta toiminnoilta ja tietoturvariskeiltä.

Suojaukset: Käytetyt menetelmät ohjelmiston turvaamiseksi.

### 6.3 Ylläpidettävyys

Kuinka helppoa on ylläpitää ja päivittää ohjelmistoa, mukaan lukien koodin selkeys, dokumentaatio ja virheiden jäljitysmekanismit.

### 6.4 Siirrettävyys ja yhteensopivuus

Ohjelman kyky toimia eri ympäristöissä ja laitteilla, sekä sen mahdollisuus integroitua muihin järjestelmiin.

### 6.5 Operointi

Kuinka helposti ohjelmaa voidaan käyttää, käynnistää ja hallita.

### 6.6 Käytettävyys

Käytettävyys: Kuinka helppoa ohjelmaa on käyttää.

Käytön tehokkuus: Kuinka nopeasti käyttäjät voivat suorittaa tarvittavat tehtävät.

Käyttäjien tyytyväisyys: Käyttäjien kokemus ohjelmiston käytöstä.

## 7 Suunnittelurajoitteet

### 7.1 Standardit

Noudattaminen tietyille standardeille tai normeille.

### 7.2 Laitteistorajoitteet

Rajoitukset, jotka liittyvät käytettävissä olevaan laitteistoon, kuten tietokoneiden tehoon tai käytettävissä oleviin liitännöihin.

### 7.3 ohjelmistorajoitteet

Rajoitukset, jotka liittyvät ohjelmistoon, esimerkiksi tiettyjen ohjelmistojen käytettävyys.

### 7.4 Muut rajoitteet

Muut suunnittelua rajoittavat tekijät, joita ei ole mainittu edellä, kuten aikarajoitukset tai budjettirajoitteet.

8 Hylätyt ratkaisuvaihtoehdot

Tässä osiossa kuvaillaan niitä ratkaisuvaihtoehtoja, jotka harkittiin mutta päätettiin hylätä sovelluksen kehittämisessä. Näihin voi sisältyä teknisiä, toiminnallisia tai taloudellisia syitä. Esimerkkejä hylätyistä ratkaisuvaihtoehdoista:

### Monialustaisuus:

Harkittiin sovelluksen kehittämistä useille eri alustoille, kuten iOS, Android ja web, mutta resurssien ja aikataulun rajoitteiden vuoksi päätettiin keskittyä aluksi yhteen alustaan.

### Lokalisaatio:

Alun perin harkittiin sovelluksen lokalisoimista useille eri kielille, mutta päätettiin aloittaa yhdellä pääkielellä ja laajentaa lokalisaatiota myöhemmin.

### Monimutkainen käyttöliittymä:

Harkittiin monimutkaisen käyttöliittymän luomista runsaalla animaatiolla ja visuaalisilla tehokeinoilla, mutta päätettiin pitäytyä yksinkertaisemmassa käyttöliittymässä käyttäjäkokemuksen parantamiseksi.

## 9 Jatkokehitysajatuksia:

Tässä osiossa kuvaillaan mahdollisia jatkokehitysideoita tai ominaisuuksia, jotka voivat olla osa sovelluksen tulevaisuutta. Nämä voivat perustua käyttäjäpalautteeseen, markkinatrendien muutoksiin tai pitkän aikavälin visioihin. Esimerkkejä jatkokehitysajatuksista:

### Monialustaisuuden laajentaminen:

Jatkossa harkitaan sovelluksen saatavuutta useammille alustoille, jotta se tavoittaa laajemman käyttäjäkunnan.

### Lisää lokalisaatioita:

Suunnitellaan sovelluksen lokalisoimista useille eri kielille, jotta se voi palvella globaalia yleisöä paremmin.

### Laajempi käyttöliittymän muokattavuus:

Tutkitaan käyttäjien mahdollisuutta muokata sovelluksen käyttöliittymää enemmän omien mieltymystensä mukaiseksi.

Lisäominaisuudet:

Lisätään uusia toiminnallisuuksia, kuten käyttäjäystävällinen tila, teemavaihtoehdot tai sosiaalisen median integraatio.

### Tietoturva ja yksityisyysparannukset:

Päivitetään sovellusta vastaamaan uusimpia tietoturva- ja yksityisyysstandardeja.

### Analytiikka ja seuranta:

Parannetaan käyttäjädatan analysointia ja seurantaa, jotta sovelluksen suorituskykyä voidaan parantaa jatkuvasti.