##### **1. Johdanto**

##### Tämä on vaatimusmäärittelydokumentti, jossa määritellään LOKI ohjelmiston / verkkosivuston kehitykseen liittyvät vaatimukset, ominaisuudet sekä projektissa käytetyt ratkaisut.

##### Loki sivusto toteutetaan Hämeen Ammattikorkeakoulun kurssityönä, joten sivustolla ei ole varsinaista tilaajaa tai tässä tapauksessa sellaisena toimii HAMK sekä Ohjelmistokehityksen kurssin pitäjä Joni Järvenpää.

##### **1.1 Tarkoitus ja kattavuus**

##### Ohjelman tarkoitus on toimia mittaustyökaluna unen määrän sekä laadun seurantaan. Ohjelman ensimmäinen valmis versio kattaa unen määrän numeerisen seurannan sekä mahdollistaa tätä kautta joidenkin yksinkertaisten laskutoimenpiteiden suorittamisen kunkin käyttäjän unidatalla.

##### Ensimmäisessä versiossa voidaan myös tallentaa sekä tulostaa teksti muotoisia selitteitä päivämääräkohtaisesti unen laadusta ikään kuin muistio muodossa.

##### **1.2 Tuote ja ympäristö**

##### Tuote on selainpohjaisella käyttöliittymällä toimiva ohjelmisto, joka toimii toistaiseksi vain paikallisesti kunkin käyttäjän tietokoneella eikä sitä ilman erillisiä ohjelmistomuutoksia voi saada toimimaan tietokoneen ulkoisessa verkkoympäristössä.

##### Ohjelma on kehitetty Microsoft Windows ympäristöön, joten sitä ei muilla järjestelmillä toistaiseksi voi käyttää.

##### Tuote mahdollistaa datan tallentamisen paikallisesti käyttäjän tietokoneelle . JSON tiedostomuodossa. Tuotteelle on eriytetty käyttäjän yksilöllinen data omaan salattuun tiedostoon muusta datasta.

##### **1.3 Määritelmät, termit ja lyhenteet**

* **Data**; Datalla tarkoitetaan yleisesti mitä tahansa kirjattavaa tietoa ja tässä tapauksessa tietoa, jota ohjelmisto tallentaa eri muodoissa.
* **Järjestelmä;** Käsittää samassa kokonaisuudessa toimivat osat, joihin kuuluu fyysinen laitteisto (mm. Kannettavatietokone), laitteistolle asennetut kokonaisuuden toimintaan vaaditut ohjelmistot (mm. Käyttöjärjestelmä sekä käytettävät sovellukset).
* **Paikallinen tallennus;** Tällä tarkoitetaan sitä, että tiedon tallennus tapahtuu nimenomaan käytössä olevan järjestelmän fyysiselle tallennuslevylle ja sitä ettei tietoa voida saavuttaa laitteen ulkopuolelta.
* **Selain;** Tietokoneohjelma tai sovellus, jolla voi selata verkkosivuja ja käyttää internetiä (mm. Chrome, Firefox, Microsoft Edge).
* **Ohjelmisto;** Tekninen kokonaisuus, joka koostuu useista eri tiedostoista sekä niissä olevan koodin lukijoista / ohjelmista tietokoneella. Ohjelmisto voi pitää sisällään useaa eri ohjelmointikieltä sekä aliohjelmistoa toimiakseen toivotulla tavalla. Ohjelmisto on tarkoituksen mukaisesti toteutettu löytämään itsenäisesti kaikki tarvittavat osat sen toimintaan liittyen, ilman tarvetta käyttäjän toimenpiteille.
* **Microsoft Windows;** Microsoft yhtiön kehittämä käyttöjärjestelmä, jonka uusin versio / tuoteperhe on Windows 11.

##### **Salaus**; Salauksella, eli kryptauksella tarkoitetaan tietoteknistä prosessia, jossa data koodataan matemaattisella toiminnolla ihmiselle lukukelvottomaan muotoon.

* **Purkuprosessi;** Purkuprosessilla, eli dekryptauksella tarkoitetaan lukukelvottomassa muodossa olevan datan muuntamista matemaattisella toiminnolla takaisin luettavaan muotoon.
* **Salausavain;** Erityinen avain, jota tarvitaan salauksen purkamiseen.

##### **1.4 Viitteet**

##### **1.5 Yleiskatsaus dokumenttiin**

##### Dokumentti on laadittu selventämään ohjelmiston käyttötarkoituksia, sille sovittuja vaatimuksia, sen teknistä toteutusta sekä kuvaamaan sen ensimmäisen valmiin version toimintoja.

Dokumentti on tarkoitettu luettavaksi kenelle tahansa ohjelman parissa työskentelevälle taholle. Se on pyritty kirjoittamaan tavalla, joka on mahdollista ymmärtää mahdollisimman helposti myös sellaisten lukijoiden, jotka eivät tunne entuudestaan teknistä termistöä tai omaa aiempaa tietämystä aihepiiristä.

Dokumentin Yleiskuvauksessa käydään läpi ohjelman vaatimaa toimintaympäristöä sekä sen käyttötarkoituksia. Osiossa Tiedot ja Tietokanta käydään läpi ohjelman tallentamaa tietoa sekä tapaa, jolla tallennusominaisuus on teknisesti toteutettu. Toiminnot kohdassa on lueteltuna kaikki ohjelman käytettävät toiminnot sekä niiden tarkoitus. Osiossa Muut Ominaisuudet kerrotaan ohjelmiston suorituskyvystä, tietoturvasta sekä rajoitteista.

##### **2. Yleiskuvaus**

##### **2.1 Ympäristö**

##### Loki ohjelmisto on tehty toimimaan Windows käyttöjärjestelmällä ja vaatii toimiakseen ASP.NET Core Runtime 8.0 ohjelmiston asennuksen sekä selaimen käyttöliittymälle. Ohjelmistoa voi käyttää usealla eri tietokoneelle asennetulla selaimella (Google Chrome, Brave, Firefox, Microsoft Internet Edge).

##### **2.2 Toiminta**

##### Loki on pohjimmiltaan tiedon tallennus- sekä hallintajärjestelmä. Loki tallentaa nukkumiseen liittyvää dataa ja auttaa tämän avulla seuraamaan unen laatua. Ohjelma mahdollistaa syötteiden tallennuksen kalenteriin kullekin päivälle nukutun unen pituudesta sekä vapaan tekstin muodossa sen laadusta. Näin ohjelman kautta voi helposti seurata järjestelmällisesti omaa unta pidemmältä ajalta.

##### **2.3 Käyttäjät**

Ohjelman käyttäjät ovat henkilöitä, jotka tekevät merkintöjä omasta unen määrästä ja sen laadusta seuratakseen omaa unen laatua.

##### Loki ohjelmistoon voi tehdä samalle järjestelmälle useita käyttäjiä paikallisesti. Käyttäjille ei ole asetettu erillisiä rooleja, vaan kaikilla on hallussa samat oikeudet, jotka rajoittuvat ohjelman perusominaisuuksien käyttöön.

Ohjelmaan kirjautunut käyttäjä voi nähdä ja muokata vain omia tietojaan eikä hänellä ole muiden käyttäjien tietoihin pääsyä.

Käyttäjälle saatavat ohjelman käyttötavat ovat toistaiseksi rajoittuneet selaimella toimivaan käyttöliittymään ja sen asettelu toimii parhaiten tietokoneen ruudulla.  
**2.4 Yleiset rajoitteet**

Ohjelmiston tallennus hoidetaan JSON tiedostoon, sen struktuurin mukaisesti. Käyttäjän syöttämän uneen liittyvän datan tallennus on hoidettu tallentamalla kaikki yhteen aiheeseen (esimerkiksi tekstimuotoinen unenlaatu data) liittyvä data yhden muuttujan taakse yhdeksi *string* tyypin tallenteeksi, joka käsitellään C# listan kautta. Jatkoa ja laajempia tallennuksia ajatellen tulee rajoitteita koska lista ei voi pitää käsittelyn aikana sisällään loputtomasti tietoa. Tosin järjestelmän arkkitehtuurista riippuen, listan kapasiteetti on tyypillisesti sen verran suuri, että kohtuu isoillakin datamäärillä käsittely onnistuu kyseisellä tavalla.  
**2.5 Oletukset ja riippuvuudet**

Ohjelman toimivuuden kannalta on oletettu, että käyttäjä käyttää ohjelmistoa Windows käyttöjärjestelmällä. Ohjelmisto on riippuvainen Windows ympäristöstä, johtuen tallennus toiminnosta. Toimintoon on osittain kovakoodattu tallennettavien .json tiedostojen tiedostosijainti ja tämä voi aiheuttaa virheitä muun tyyppisillä järjestelmillä.

Ohjelmisto käyttää C# ohjelmointikieltä ja sen erillisiä toimintoja sekä kirjastoja. Loki ohjelmisto on tästä syystä riippuvainen ASP.NET Core Runtime 8.0 ohjelmistopaketista.

Lokin käyttöliittymä on tehty HTML (hypertekstin merkintäkielellä), joten se on riippuvainen järjestelmään asennetusta selaimesta. Jokin yllä mainituista selaimista on ainut tapa, jolla käyttöliittymää voi käyttää.

**3. Tiedot ja tietokanta**

**3.1 Tietosisältö**

Ohjelmisto käsittelee käyttäjän tietoja kohtuu rajallisesti. Käyttäjän tietoihin tallennetaan ainoastaan valinnainen käyttäjänimi sekä sille salasana, jota käytetään ohjelmaan kirjautuessa. Käyttäjätietoihin tallennetaan myös yleisluontoinen esittelyteksti käyttäjästä.

Itse ohjelmiston toimintoihin liittyvä unidata käsitellään erillisenä käyttäjädatasta ja molemmat tallennukset tehdään samaan tietojärjestelmän sijaintiin erillisinä tiedostoina, joista käyttäjätietojen tiedosto on salattu.

##### **3.2 Käyttöintensiteetti**

##### Ohjelmistolla ei ole suurta käyttöintensiteettiä, koska se on tehty kunkin käyttäjän tietokoneella paikallisesti ja vaikka käyttäjiä voi olla useampi luotuna, voi ohjelmaa käyttää vain yksi käyttäjä kerrallaan. **3.3 Kapasiteettivaatimukset**

Ohjelmisto on suunniteltu toimimaan ilman suuria vaatimuksia nykyaikaisilla tietokoneilla ja niiden käyttöjärjestelmillä. Ohjelmisto vaatii käyttäjältä asennetun selainohjelman nykyaikaisen käyttöjärjestelmän päälle (tarkemmat tiedot kohdasta 2.1).

**3.4 Tiedostot ja asetustiedostot**

Kaikki ohjelman toimintaan vaaditut tiedostot sekä asetustiedostot löytyvät projektin GitHub profiilin kautta, kopioimalla sen hakemiston verkosta linkillä; <https://github.com/sasinkkon/LOKI.git> .

Kun tiedostot kopioitu tietokoneelle verkosta, saadaan ohjelma käynnistettyä sen .exe tiedostolla. Kyseinen tiedosto löytyy ladatun tiedostokansion alikansioihin seuraavassa järjestyksessä; LOKI- , Blazori- , bin- , Release- , net8.0- ja kaksoisklikkaamalla ObjectOriented\_Template.exe tiedostoa.

**4. Toiminnot**

Ohjelman päätoiminto on siihen tehty kalenteri, johon käyttäjä voi tehdä uni merkintöjä. Kalenteri merkinnän saa aikaiseksi navigoitumalla välisivulle “create”, jossa avautuu kalenteri. Kalenteriin saa merkinnän klikkaamalla jotain päivämäärää kahdesti ja kirjoittamalla kenttiin tiedot unen pituudesta sekä laadusta.

**5. Ulkoiset liittymät**

Ohjelmisto ei käytä ulkoisia liitäntöjä.  
  
**6. Muut ominaisuudet**

Tässä osiossa käsittelemme ohjelmistoon liittyviä muita ominaisuuksia liittyen käytettävyyteen, ohjelmiston suorituskykyyn ja ohjelmiston ylläpitoon.

##### **6.1 Suorituskyky ja vasteajat**

Suorituskyky ohjelmistossa on käyttäjän omasta käyttöjärjestelmästä riippuvainen, sillä ohjelmisto ei varsinaisesti nojaa suureen tiedon prosessointiin, jolloin kaikki ohjelmiston toiminnot tulisi tapahtua 3 sekunnin sisällä ohjelmistoon annetusta “komennosta”.

##### **6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset**

Ohjelmiston pitäisi olla saavutettavuuden kannalta hyvä, sillä sen toiminta varsinaisesti perustuu jo valmiiksi toimivaan pohjakoodiin, jonka projektiryhmä on luonut, ja käyttäjän toiminta ei varsinaisesti pitäisi aiheuttaa ongelmaa. Emme ainakaan ole itse ryhmänä kohdanneet sellaista ongelmaa joka tämän tilanteen aiheuttaisi.

Turvallisuuden ja suojauksen kannalta päädyimme opettajankin ohjeistuksen kanssa siihen että tallentamamme ja salattava tieto tallentuu vain paikallisesti ja salauksen purkuavaimet tallentuvat myös vain paikallisesti käyttäjän omalle laitteelle, jolloin vältämme mahdollisesti kriittisen datan (salasanat) päättymisen serverin puolelle jossa se olisi alttiimpi tietoturvamurtoon.

##### **6.3 Ylläpidettävyys**

Luotu ohjelmisto ja sen ylläpidettävyys nojautuvat vahvasti Windows käyttöjärjestelmän toimivuuteen, sekä käyttämämme Blazor Server .NET 8 frameworkin toimivuuteen. Toisin sanoen ohjelman ylläpidollinen toimivuus koodia pois lukien ei ole ryhmän jäsenten vaikutuksen ulottuvissa. (Tietääksemme.)

##### **6.4 Siirrettävyys ja yhteensopivuus**

Ohjelmisto on luotu toimimaan Windowsin käyttöjärjestelmälle ja niitä käyttävien alustojen yleisimmille nettiselaimille i.e. Chrome, Edge tai Firefox. On mahdollista, että ohjelmiston saisi muunnettua toimimaan mobiilisovellus ympäristössä, mutta emme ole perehtyneet siihen kuinka sen saisi näin toimimaan.

##### **6.5 Operointi**

Ohjelmisto ja sen operointi on luotu toimimaan simppelinä verkkoselain sovelluksena ja sen käyttöliittymä ja ohjelmiston asettelu on mukautettu toimimaan yleisillä tietokonenäytöillä ja niiden yleisillä resoluutio asetuksilla.

##### **6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys**

##### **7. Suunnittelurajoitteet**

Tässä osiossa kuvataan ne tekijät, jotka rajoittavat tai ohjaavat projektin suunnittelua ja toteutusta. Rajoitteet voivat liittyä käytettäviin standardeihin, laitteistoihin, ohjelmistoihin sekä muihin projektin toteutukseen vaikuttaviin seikkoihin.

##### **7.1 Standardit**

Projektissa käsitellään käyttäjätietoja, jotka salataan ja puretaan ennen tallennusta ja käytön aikana. Tietojen käsittelyssä noudatetaan GDPR:n perusperiaatteita kuten tietojen minimointia ja suojattua säilytystä, vaikka kyseessä on opiskeluprojekti.

Ryhmän jäsenet ovat hyödyntäneet C#:n Aes-luokkaa (System.Security.Cryptography.Aes), joka noudattaa AES (Advanced Encryption Standard) -salausstandardia.

Käyttöliittymän suunnittelussa on pyritty selkeyteen ja saavutettavuuteen, WCAG 2.1 - periaatteiden mukaisesti.

Koodissa noudatetaan .NET-kehityksen yleisiä koodauskäytäntöjä ja yhtenäistä nimeämistapaa.

Dokumentointi on laadittu IEEE 830-1998 Standardinmukaisen vaatimusmäärittelyn - rakenteen mukaisesti

##### **7.2 Laitteistorajoitteet**

Ohjelmisto on suunniteltu käytettäväksi Windowsin työpöytäkoneilla ja kannettavilla tietokoneilla selaimen kautta. Kehitys- ja testausympäristöinä toimivat projektin jäsenten omat tietokoneet. Ohjelmiston käyttö vaatii modernin selaimen (Chrome, Edge, Brave tai Firefox). Ohjelmisto ei vaadi erityisiä oheislaitteita tai ulkoisia laiteyhteyksiä.

##### **7.3 Ohjelmistorajoitteet**

Projekti on toteutettu C#-kielellä Blazor Server (.Net 8) -ympäristössä (Opettajan tarjoama template).

Käyttäjätiedot tallennetaan paikallisesti JSON-muodossa, ja tiedostot salataan ennen tallennusta. Tietojen purku tapahtuu ohjelman sisällä vain käyttöhetkellä. Käytetty salausalgoritmi on C#:n Aes-luokka (System.Security.Cryptography.Aes), joka noudattaa AES (Advanced Encryption Standard) -salausstandardia (Mainittu aikaisemmin).

Käyttöliittymän kieli on englanti, eikä lokalisaatiota ole toteutettu.

Kehitystyökalut: Visual Studio 2022, GitHub-versionhallinta

**7.4 Muut rajoitteet**

Projekti on opiskelijaryhmän harjoitustyö, ei tuotantokäyttöön tarkoitettu ohjelmisto.

Aikataulu on rajattu kurssin kestoon (8 viikkoa) ja projektin varsinainen suoritusaika on ollut 5 viikkoa.

Ryhmässä työskentelee 5 opiskelijaa, joilla on yhdessä jaetut ja vaihtelevat vastuut (frontend, backend, tietoturva, dokumentointi)

**8. Hylätyt ratkaisuvaihtoehdot**

Ohjelmiston alkuperäinen mittakaava oli laajempi ja sen toiminnallisuus olisi pitänyt kattaa vielä erillinen palkanlaskenta osio, josta myöhemmin luovuttiin. Tiedontallennuksen ja kalenterin kanssa hetken aikaa pohdittiin mahdollista 3.osapuolen API ratkaisua, mutta nekin jätettiin pois ja päädyttiin nykyisiin vaihtoehtoihin.

**9. Jatkokehitysajatuksia**

Ohjelmistoa pystyisi kehittämään jatkossa olemaan vielä selkeämpi ja helppokäyttöisempi käyttäjälle. Ohjelmistoa voisi muokata siten, että sen saisi vietyä uuteen käyttöympäristöön (Mobiilisovellus).