

SAVONIA

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIKAN JA LIIKENTEEN ALA

KALAPÄIVÄKIRJA

Verkkosovellus

TEKIJÄ/T Juuso Kurkinen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Juuso Kurkinen	
Työn nimi Kalapäiväkirja	
Päiväys 10.10.2023	Sivumäärä/Liitteet 35
Toimeksiantaja/Yhteistyökumppani(t) Growly Oy	
<p>Tiivistelmä</p> <p>Tässä opinnäytetyössä käsiteltiin Kalapäiväkirja-verkkosovellusprojektin toteutusta ja siihen liittyviä teknisiä ratkaisuja. Sovellus tarjoaa käyttäjilleen alustan, johon he voivat tallentaa kalasaaliitaan joko muiden kalastajien nähtäväksi tai yksityisiin muistiinpanoihin. Lisäksi sovellus luo yhteisöllisen ympäristön, jossa kalastajat voivat kommentoida toistensa julkaisemia kalasaaliita. Projektissa hyödynnettiin moderneja verkkokehityksen tekniikoita, ja sen tuloksena syntyi responsiivinen ja käyttäjäystävällinen sovellus.</p> <p>Sovellus vaatii käyttäjätilin luomisen ja tilille kirjautumisen. Jotta käyttäjä voi hyödyntää sovelluksen ominaisuuksia, hänellä tulee siis olla käyttäjätunnus. Kirjautuneet käyttäjät voivat tarkastella tilastotietoja, jotka perustuvat sovellukseen tallennettuihin kalasaaliisiin. He voivat myös tutustua muihin sovelluksen käyttäjiin ja vieraila heidän profiilisivuillaan.</p> <p>Projektin tavoitteena oli luoda responsiivinen ja käyttäjäystävällinen sovellus, joka täyttää sille asetetut vaatimukset. Projektin tavoitteena ei ollut saada valmiiksi tuotantokelpoista sovellusta, vaan toteuttaa toimiva perusta, jota voidaan jatkokehittää tulevaisuudessa.</p> <p>Tässä opinnäytetyöraportissa esitellään sovelluksen käyttöliittymänäkymät ja avataan lukijalle niiden taustalla olevaa logiikkaa. Yhteenvedossa kerrotaan, miten projektissa saavutettiin tavoitteet, mitä vaatimuksia saavutettiin ja mitä haasteita projektiin liittyi.</p>	
Avainsanat Kalapäiväkirja, verkkosovellus, web-sovellus, React, Node.js, MySQL	

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology	
Author(s) Juuso Kurkinen	
Title of Thesis Kalapäiväkirja	
Date 10.10.2023	Pages/Appendices 35
Client Organisation /Partners Growly Oy	
<p>Abstract</p> <p>This thesis covers the implementation of the Kalapäiväkirja web application project and the associated technical solutions. The application provides users with a platform to record their fishing catches either privately or for public viewing by other fishers. Additionally, the application creates a community environment where fishermen can comment on each other's published catches. The project utilized modern web development techniques, resulting in a responsive and user-friendly application.</p> <p>The application users are required to create an account and to log in. Thus, to take advantage of the application's features, users need to have a user account. Logged-in users can view statistical data based on the fishing catches they have recorded in the application. They can also explore other users of the application and visit their profile pages.</p> <p>The goal of the project was to create a responsive and user-friendly application that meets the specified requirements. The objective was not to create a production-ready application but to establish a functional foundation that can be further developed in the future.</p> <p>The thesis presents the user interface views of the application and explains the underlying logic to the reader. In the summary, we describe how the project achieved its goals, which requirements were met, and what challenges were encountered in the project.</p>	
<p>Keywords</p> <p>Kalapäiväkirja, verkkosovellus, web-sovellus, React, Node.js, MySQL</p>	

SISÄLTÖ

TERMISTÖ JA LYHENTEET	6
1 JOHDANTO	8
2 KALAPÄIVÄKIRJA.....	9
2.1 Projektin toimeksiantaja	9
2.2 Sovelluksen tarve.....	9
2.3 Sovellukselle asetetut vaatimukset ja tavoitteet	9
3 SOVELLUKSEN ARKKITEHTUURI JA TEKNIIKAT.....	10
3.1 Käyttöliittymä	10
3.1.1 React.....	11
3.2 Palvelinpuoli	13
3.2.1 Autentikointi.....	13
3.2.2 Node.js	14
3.2.3 Express.js	15
3.3 Sovelluksen riippuvuudet.....	16
3.3.1 Palvelinpuolen riippuvuudet.....	17
3.3.2 Käyttöliittymän riippuvuudet.....	18
3.4 Tietokanta.....	18
3.4.1 MySQL	19
3.5 Pilvipalvelut	20
4 KEHITYSTYÖKALUT JA VERSIONHALLINTA.....	22
5 ULKOASU JA NÄKYMÄT	23
5.1 Käyttäjätunnuksen luonti	23
5.2 Sovellukseen kirjautuminen	24
5.3 Sovelluksen etusivu.....	25
5.3.1 Näkymä kirjautumattomalle käyttäjälle.....	25
5.3.2 Näkymä kirjautuneelle käyttäjälle	26
5.4 Kalasaaliit.....	27
5.5 Yksittäinen kalasaalis	28
5.6 Käyttäjät	29
5.7 Käyttäjän profiili.....	30
5.8 Saaliin lisäys	31

5.9	Statistiikka.....	33
6	YHTEENVETO.....	35
	LÄHTEET	36

KUVALUETTELO

Kuva 1.	Arkkitehtuurikaavio (Kurkinen 2023)	10
Kuva 2.	Komponenttipuu (Kurkinen 2023)	11
Kuva 3.	Esimerkki React-sovelluksen sivunäkymästä (Kurkinen 2023).....	12
Kuva 4.	DOM:in vertailu ja päivitys prosessi (Kurkinen 2023)	13
Kuva 5.	Havainnollistava kuva käyttäjän tunnistautumisesta (Kurkinen 2023).....	14
Kuva 6.	Node.js arkkitehtuuri (Kurkinen 2023)	15
Kuva 7.	Esimerkki kuva package.json tiedostosta (Kurkinen 2023)	17
Kuva 8.	Tietokannan ER-kaavio (Kurkinen 2023)	19
Kuva 9.	Yksinkertainen esimerkki SQL-syntaksista (Kurkinen 2023)	20
Kuva 10.	Kuvankaappaus clouinary.js tiedostosta (Kurkinen 2023)	20
Kuva 11.	Kuvankaappaus imgToCloudinary.js-tiedostosta (Kurkinen 2023)	21
Kuva 12.	Tunnuksen luonti (Kurkinen 2023)	24
Kuva 13.	Kirjautumislomake (Kurkinen 2023).....	25
Kuva 14.	Etusivu kirjautumattomalle käyttäjälle (Kurkinen 2023).....	26
Kuva 15.	Etusivu kirjautuneelle käyttäjälle (Kurkinen 2023)	27
Kuva 16.	Kalasaaliit-näkymä (Kurkinen 2023).....	28
Kuva 17.	Yksittäisen kalasaaliin näkymä (Kurkinen 2023).....	29
Kuva 18.	Käyttäjät-näkymä (Kurkinen 2023)	30
Kuva 19.	Yksittäisen käyttäjän tiedot (Kurkinen 2023)	30
Kuva 20.	Yksittäisen käyttäjän lisäämät kalasaaliit (Kurkinen 2023)	31
Kuva 21.	Kalasaaliin lisäyslomake, sivu 1 (Kurkinen 2023)	32
Kuva 22.	Kalasaaliin lisäyslomake, sivu 2 (Kurkinen 2023)	32
Kuva 23.	Kalasaaliin lisäyslomake, tarkan sijainnin valinta (Kurkinen 2023)	33
Kuva 24.	Kalasaaliin lisäyslomake, sivu 3 (Kurkinen 2023)	33
Kuva 25.	Sovelluksen statistiikkanäkymä (Kurkinen 2023).....	34

TERMISTÖ JA LYHENTEET

Termi	Suomennos	Selite
API	Ohjelmointirajapinta	Application Programming Interface. Mahdollistaa tietojen liikkumisen sovellusten välillä.
CSS	Tyylitiedosto	Cascading Styling Sheets. CSS on tyylien kuvauskieli, jota käytetään verkkosivujen ja sovelluksien tyylittelyyn
DOM	Dokumenttioniomalli	Document Object Model
Framework	Sovelluskehys	Ohjelmistokehityksessä käytetty kehys, joka usein helpottaa ja yksinkertaistaa ohjelmointityötä.
HTML	HTML	HyperText Markup Language
HTTP/S	HTTP/S	sovelluserroksen protokolla, jota selaimet ja palvelimet käyttävät tiedonsiirtoon. HTTPS on HTTP- ja TLS/SSL-protokollien yhdistelmä, joka mahdollistaa suojatun tiedonsiirron.
JavaScript	JavaScript	Pääosin webympäristössä käytettävä ohjelmointikieli
REST	REST	Representational State Transfer. Rajapintojen arkkitehtuurimalli
JSON	JSON	JavaScript Object Notation. Ohjelmoinnissa yleisesti käytetty tiedostomuoto tiedonsiirtoon.
VDOM	Virtuaalinen dokumenttioniomalli	VirtualDOM
Tailwind CSS	Tailwind CSS	Avoimen lähdekoodin CSS-kehys
Library	Kirjasto	Kokoelma valmiiksi kirjoitettuja toiminnallisuuksia
VS Code	VS Code	Visual Studio Code on suosittu avoimenlähdekoodin koodieditori.
Git	Git	Hajautettu versionhallintajärjestelmä
Npm	Npm	Node Package Manager. JavaScript-ohjelmointikielen pakettinhallintajärjestelmä.
Full-stack	kokonaisvaltainen kehitys	Viittaa sovelluskehityksessä työhön, joka kattaa sekä käyttöliittymän että palvelinpuolen kehityksen.

MySQL	MySQL	MySQL on avoimen lähdekoodin tietokantajärjestelmä, joka mahdollistaa tietojen tallentamisen, hallinnan ja kyselyt.
SQL	SQL	Structured Query Language. Standardisoitu kyselykieli, jota käytetään relaatiotietokantojen hallintaan ja tiedonhallintaan.
React	React	React on JavaScript-kirjasto, jota käytetään käyttöliittymien rakentamiseen verkkosovelluksissa.
Vite	Vite	Vite on nopea ja moderneja verkkosovelluksia varten suunniteltu kehitysympäristö ja rakennustyökalu.
Context API	Context API	Context API on React-kirjastossa sisäänrakennettu työkalu, joka mahdollistaa tilan ja muiden tietojen jakamisen komponenttien välillä.
SQL-injektio	SQL-injektio	SQL-injektio on tietoturvaongelma, jossa pyritään manipuloimaan tietokantakyselyitä sovelluksessa hyödyntämällä sovelluksen mahdollisia haavoittuvuuksia.
NodeJS	NodeJS	Avoimen lähdekoodin JavaScript-ympäristö, joka mahdollistaa JavaScriptin suorittamisen palvelinpuolella.
Express.js	Express.js	Avoimen lähdekoodin web-sovelluskehys Node.js-ympäristölle.
JWT	JWT	JSON Web Token. Standardi, joka määrittelee tokenin muodon ja sisällön sekä takaa tietojen turvalliseen välittämisen kahden osapuolen välillä.

1 JOHDANTO

Tässä raportissa kerrotaan Kalapäiväkirja-verkkosovelluksen toteutusprojektista. Kalapäivä-sovelluksen tarkoitus on tarjota kalastajille alusta, johon he voivat tallentaa saadut kalasaaliinsa muistiin. Sovellus tarjoaa myös yhteisöllisen alustan, jossa käyttäjät voivat julkaista saaliinsa muiden nähtäväksi ja kommentoida niitä.

Sovellusprojekti sai alkunsa käytännön ongelmasta. Huomattiin, että kalastusharrastuksen myötä saaliiden kirjaaminen muistiin olisi hyödyllistä, mutta tähän sopivaa sovellusta ei ollut saatavilla. Vähitellen sovelluksen ominaisuudet ja vaatimukset laajenivat nykyiseen muotoonsa.

Projektin tavoitteena on luoda toimiva ja käyttäjäystävällinen sovellus, joka vastaa kalastajien tarpeisiin. Projektilla ei ole taloudellisia tavoitteita. Raportin alkupuolella käsitellään sovelluksen tarkempia vaatimuksia ja tavoitteita, joista keskeisin on helpottaa kalastajien saaliiden kirjaamista.

Sovelluksen kohdeyleisöä ovat kaikki kalaharrastajat riippumatta käytetystä kalastusmuodosta, tai siitä, ovatko he satunnaisia vai aktiivisia kalastajia. Sovellus on suunnattu suomalaisille käyttäjille.

Kalapäiväkirja on verkkopohjainen sovellus, joka on toteutettu moderneilla tekniikoilla. Sovellus on niin sanottu "full-stack"-sovellus, jossa käyttöliittymä on rakennettu React-JavaScript-kirjastolla ja palvelinpuoli Node.js-ympäristössä hyödyntäen Express.js-ohjelmistokehystä. Sovelluksen tiedot tallennetaan MySQL-tietokantaan.

Projekti on suunniteltu toteutettavaksi suhteellisen nopealla aikataululla ja sovelluksen vaatimukset on asetettu sen mukaisiksi. Tämän opinnäytetyön tavoitteena ei ole tuottaa valmiiksi julkaistavaa sovellusta, vaan saada sovellus vaiheeseen, jossa sitä voidaan testata ja jatkokehittää.

2 KALAPÄIVÄKIRJA

2.1 Projektin toimeksiantaja

Projektin toimeksiantajana on kuopiolainen digitoimisto nimeltään Growly Oy. Yritys perustettiin vuonna 2021 ja sillä on tällä hetkellä noin 10 työntekijää. Growly palvelee asiakkaita erikokoisista yrityksistä, keskittyen pääasiassa verkkokauppojen ja verkkosivujen suunnitteluun ja toteutukseen.

2.2 Sovelluksen tarve

Idea kalapäiväkirjasta sai alkunsa, kun olimme kalastamassa projektin toimeksiantajan kanssa. Huomasimme tarpeen kirjata ylös kalasaaliimme kaikkine tietoineen, jotka johtivat tärppiin. Monet kalastusharrastajat pitävät fyysistä päiväkirjaa saaliistaan, ja halusimme tarjota tähän modernin ratkaisun. Tämä ajatus johti web-pohjaisen sovelluksen konseptointiin, joka nopeasti kehittyi siihen pisteeseen, että päätimme tehdä siitä julkisen. Alun perin ajattelimme sovelluksen olevan tarkoitettu vain omien kalasaaliiden tallentamiseen, mutta pian huomasimme, että se voisi soveltua myös laajemmalle yleisölle.

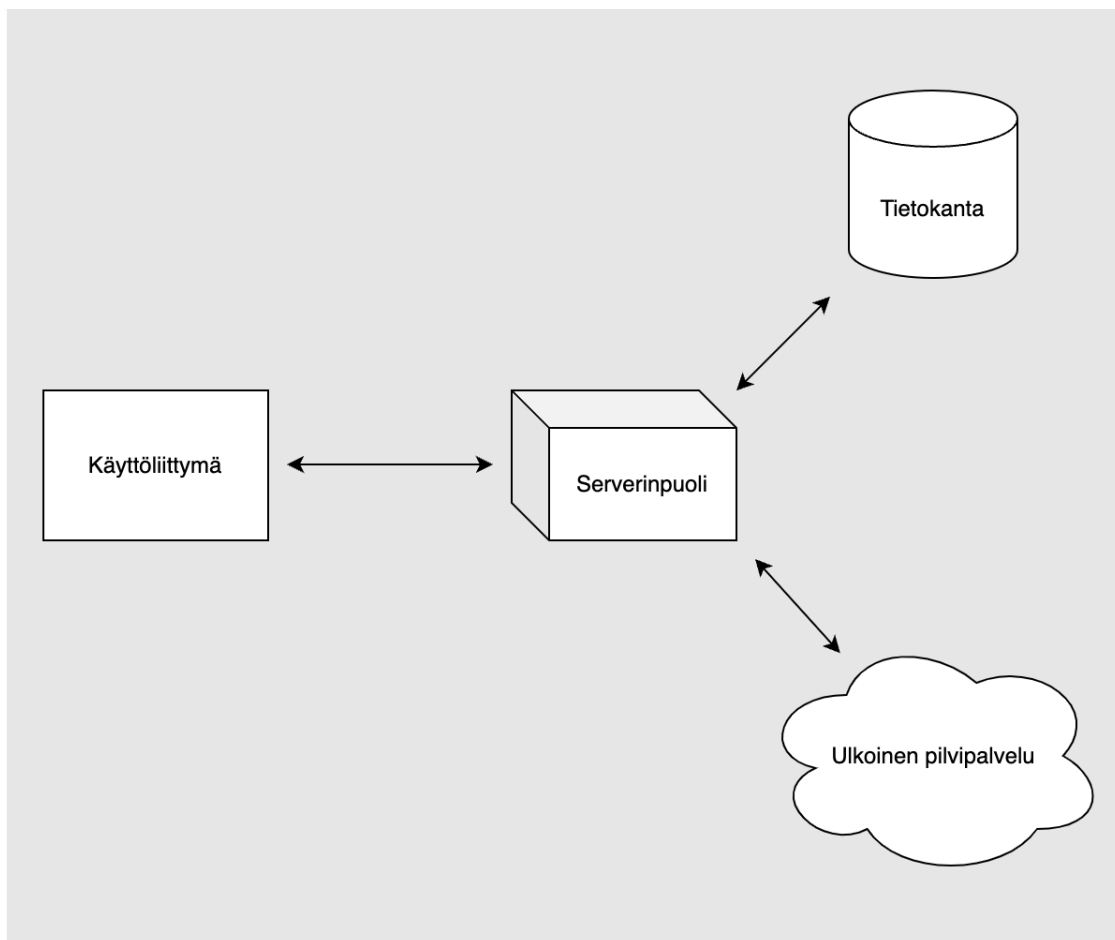
2.3 Sovellukselle asetetut vaatimukset ja tavoitteet

Ennen projektin aloittamista sovellukselle asetettiin seuraavat tavoitteet:

1. Käyttäjähallinta
 - Mahdollisuus luoda käyttäjätili
 - Mahdollisuus kirjautua sovellukseen
 - Mahdollisuus kirjautua ulos
2. Kalasaaliiden hallinta
 - Mahdollisuus lisätä kalasaaliita
3. Kalasaaliiden listaus
 - Julkiset kalasaaliit tulee näkyä listattuna
 - Julkisia kalasaaliita tulee voida suodattaa
 - Näkymä, joka näyttää käyttäjän omat kalasaaliit
4. Kalasaaliin näyttäminen
 - Näkymä yksittäiselle kalasaalille sisältäen saaliin tarkat tiedot
5. Käyttäjän toiminnot
 - Mahdollisuus muokata käyttäjätiliä
 - Mahdollisuus poistaa käyttäjätili
 - Mahdollisuus kommentoida kalasaalista
6. Käyttöliittymä
 - Tyylikäs, responsiivinen ja helppokäyttöinen käyttöliittymä
7. Tietoturva
 - Turvallinen tietojenhallinta ja varastointi

3 SOVELLUKSEN ARKKITEHTUURI JA TEKNIIKAT

Kalapäiväkirja on verkkosovellus, joka on rakennettu hyödyntäen JavaScript-ohjelmointikielen moderneja kirjastoja ja ohjelmistokehyksiä. Sovelluksen käyttöliittymä luodaan asiakaspäässä, eli verkkoselaimessa. Sovellus voidaan jakaa neljään pääkomponenttiin, jotka välittävät tietoa keskenään. Sovelluksen käyttöliittymä kommunikoi palvelinpään kanssa REST-rajapinnan välityksellä. Palvelinpuoli puolestaan viestii käyttöliittymän, tietokannan ja ulkoisten pilvipalveluiden kanssa.



Kuva 1. Arkkitehtuurikaavio (Kurkinen 2023)

Kuten arkkitehtuurikaaviosta huomaamme (Kuva 1), serverinpuoli on periaatteessa sovelluksen keskiö, jonka kautta tieto liikkuu sovelluksen pääkomponenttien välillä.

3.1 Käyttöliittymä

Sovelluksen käyttöliittymä on toteutettu React-JavaScript-kirjastolla. Käytännössä tämä tarkoittaa sitä, että JavaScriptistä, HTML:stä ja CSS:stä koostuva sovelluksen käyttöliittymä renderöidään verkkoselaimessa. React puolestaan luodaan sovelluksessa käyttäen Vite-kehitystyökalua. Sovelluksen

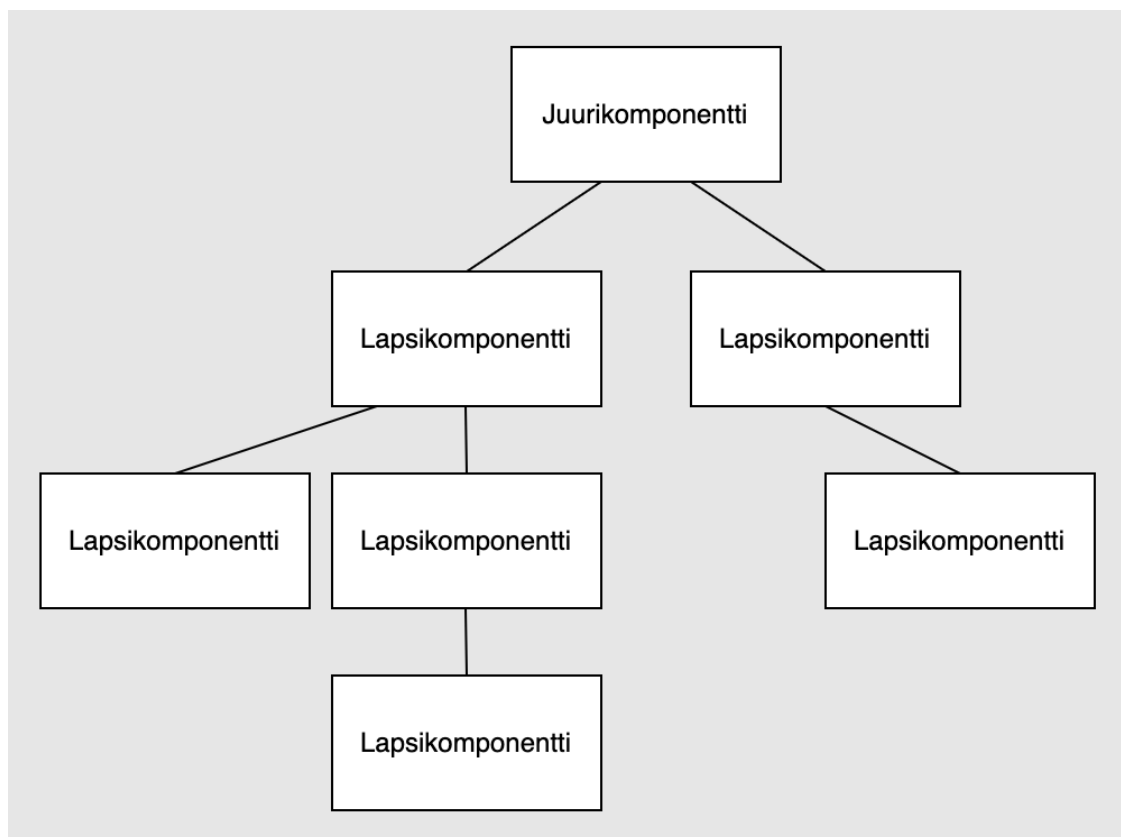
ulkoasun tyylittelyyn käytetään Tailwind CSS -ohjelmistokehystä, joka nopeuttaa ja helpottaa käyttöliittymän muotoilua ja tyylittelyä. Käyttöliittymä kommunikoi palvelimen kanssa HTTP-pyyntöjen välityksellä.

Käyttöliittymän tilanhallintaan valittiin React:iin sisään rakennetun Context API:n, koska se sopii hyvin tämän kokoluokan sovellukseen ja koska siihen oltiin perehdytty aiemmin.

3.1.1 React

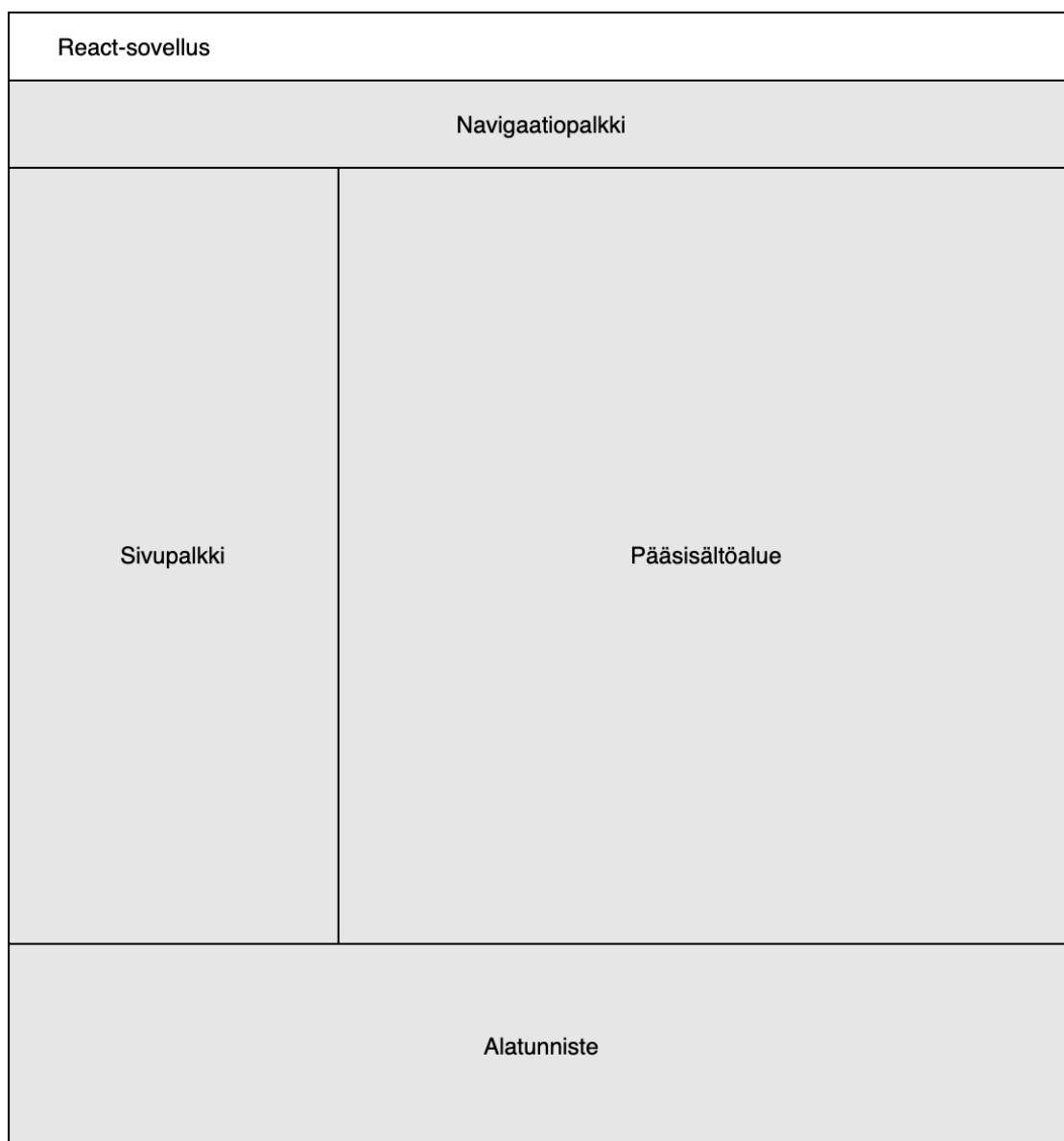
React.js on JavaScript-ohjelmointikielen kirjasto, joka mahdollistaa nopeiden ja interaktiivisten käyttöliittymien rakentamisen. Facebook kehitti React:in vuonna 2011. Vuonna 2023 Stack Overflow -sivuston toteuttaman kyselyn mukaan React osoittautui suosituimmaksi käyttöliittymäohjelmistokehykseksi.

React-sovellus koostuu komponenteista, jotka mahdollistavat käyttöliittymän hajauttamisen pieniin osiin. Itsenäiset ja uudelleen käytettävät komponentit vähentävät myös kirjoitettavan koodin määrää. Esimerkiksi komponenttia, joka sisältää painikkeen, voidaan uudelleen käyttää eri osissa sovellusta. Jokainen React-sovellus sisältää vähintään yhden komponentin, jota kutsutaan juurikomponentiksi. Juurikomponentti sisältää yleensä lapsikomponentteja, jotka voivat puolestaan sisältää alikomponentteja ja niin edelleen. React-sovelluksen rakenne voidaan kuvata komponenttipuuna. (Mosh Hamedani 2018.)



Kuva 2. Komponenttipuu (Kurkinen 2023)

Sovelluksessa on pyritty käyttämään mahdollisimman paljon uudelleenkäytettäviä komponentteja. Seuraavassa kuvassa (Kuva 3) on esimerkki, miten React-sovelluksen yksittäinen näkymä voisi koostua komponenteista.



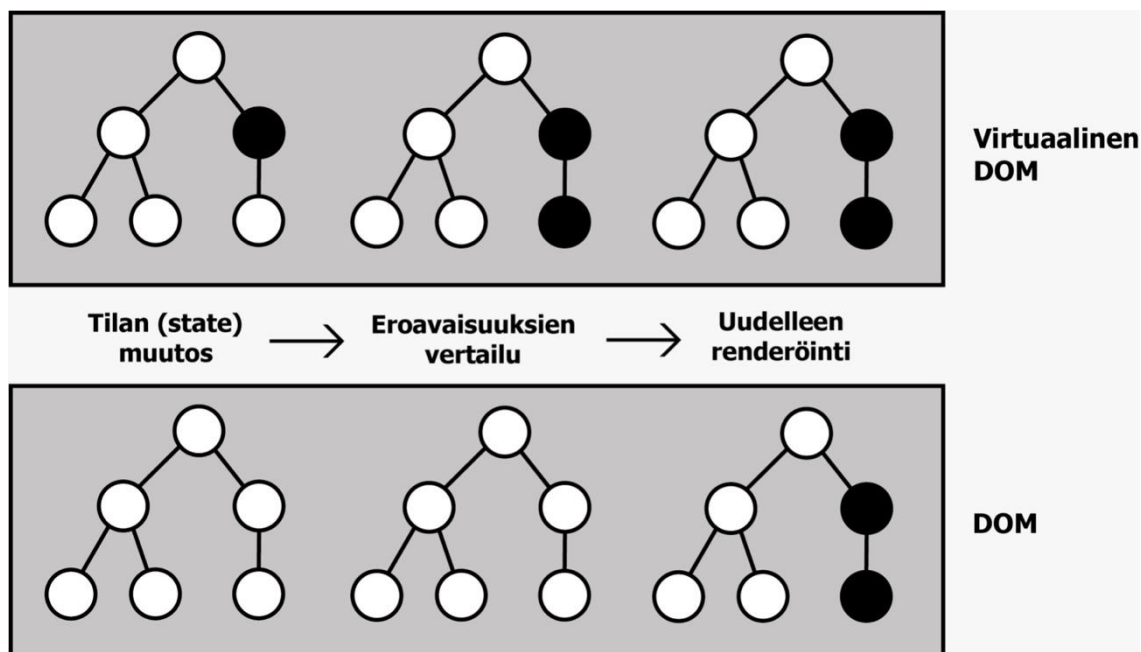
 = Komponentti

Kuva 3. Esimerkki React-sovelluksen sivunäkymästä (Kurkinen 2023)

Yleensä komponentti sisältää tilan ja palauttaa React-elementin. React-elementti on JavaScript-objekti, joka kuvastaa DOM-elementtiä, mutta ei ole suoraan DOM-elementti. Sen sijaan React-elementti on ikään kuin virtuaalinen versio DOM-elementistä, jota React käyttää muutosten seuraamiseen. Tätä virtuaalista rakennetta kutsutaan virtuaaliseksi DOM-malliksi. (Mosh Hamedani 2018.)

Toisin kuin oikea DOM, virtuaalinen DOM on kevyt ja edullinen luoda. Kun komponentin tila muuttuu, React luo uuden React-elementin ja vertaa sitä edelliseen. Se päivittää sitten vain ne osat oikeasta DOM:ista, jotka ovat muuttuneet. (Mosh Hamedani 2018.)

Yksi Reactin eduista on, ettei sitä käytettäessä tarvita vuorovaikutusta DOM API:n kanssa, kuten perinteisessä JavaScriptissä tai jQueryssä. (Mosh Hamedani 2018.)



Kuva 4. DOM:in vertailu ja päivitys prosessi (Kurkinen 2023)

3.2 Palvelinpuoli

Sovelluksen palvelinpuoli on toteutettu Node.js-suoritusympäristöllä. Palvelinpuoli käyttää Express.js ohjelmistokehystä, joka nopeuttaa ja helpottaa palvelinpuolen ohjelmointityötä. Palvelin tarjoilee REST-rajapintaa, jonka kautta käyttöliittymä hakee ja lähettää tietoa HTTP-metodien avulla. Sovelluksen palvelinpää on yhteydessä MySQL-tietokantaan, josta se hakee, poistaa, muokkaa tai lisää tietoja käyttöliittymän pyyntöjen mukaisesti. Palvelinpään ohjelmointia tehtäessä on otettu huomioon yleisiä tietoturvakäytäntöjä, jotta sovellus olisi kokonaisuudessaan mahdollisimman tietoturvallinen.

Palvelinpää on tietokantaan yhteydessä MySQL 2-kirjaston avulla. SQL-kyselyt on toteutettu modernilla ja tietoturvaisilla tavoilla. Esimerkiksi SQL-injektiot on otettu huomioon ja sovellus käyttää ainoastaan parametrisoituja SQL-kyselyitä.

3.2.1 Autentikointi

Sovellus käyttää JSON-merkki (JWT) pohjaista autentikointia. JWT (JSON Web Token) -pohjainen autentikointi käyttää usein Access Tokenia ja Refresh Tokenia. (RFC 6749.)

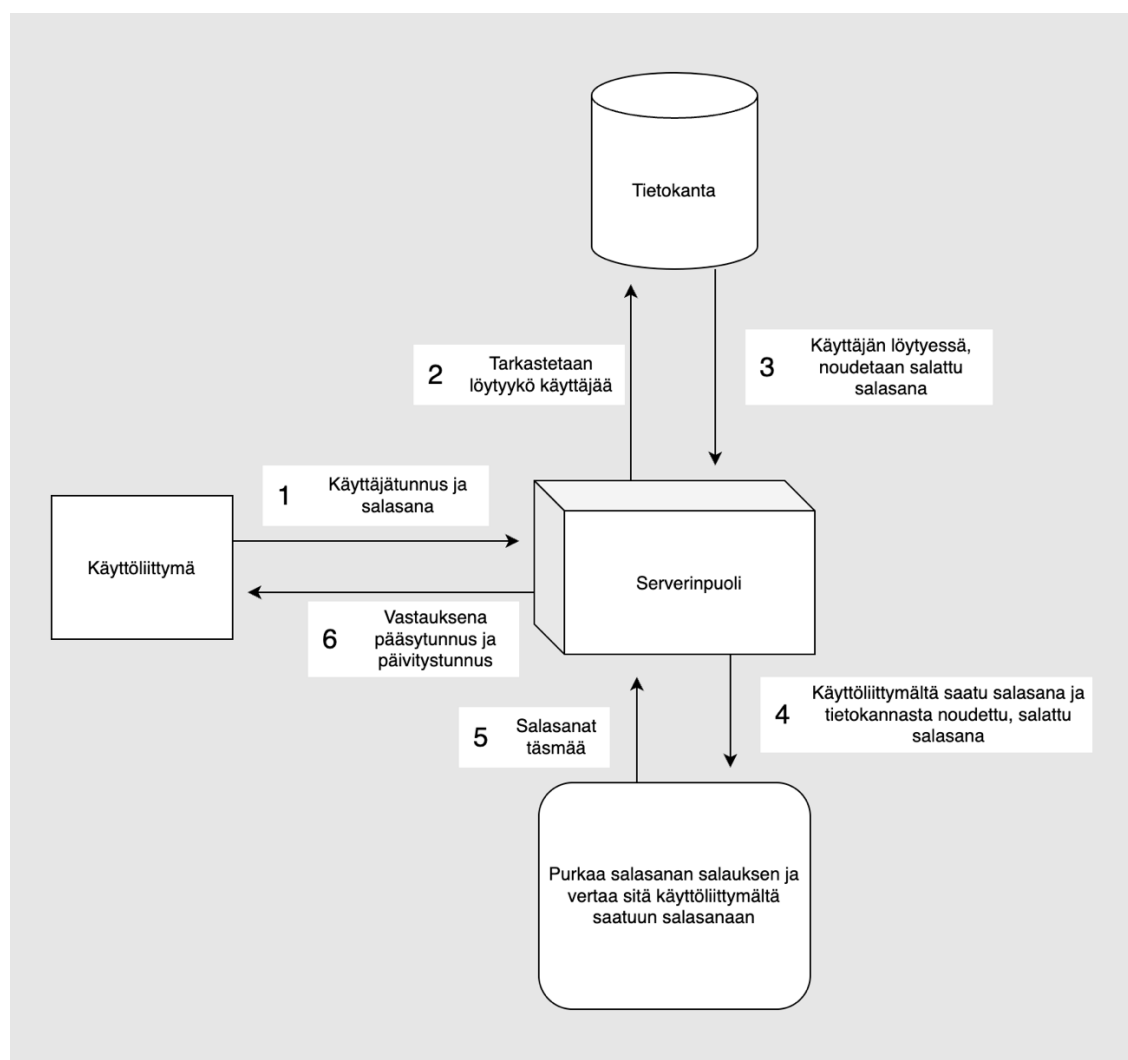
Access Token on lyhytikäinen token, joka myönnetään käyttäjälle autentikoinnin jälkeen. Se mahdollistaa pääsyn sovelluksen suojattuihin resursseihin, kuten API-palveluihin. Access Tokenin tarkoitus on varmistaa, että käyttäjä, joka pyytää suojattua resurssia, on asianmukaisesti autentikoitu. Access Tokenin voimassaoloaika on tyypillisesti lyhyt, mikä minimoi turvallisuusriskejä, jos se joutuu väärin käsiin. (RFC 6749.)

Refresh Token on pidempään voimassa oleva token, joka myönnetään yhdessä Access Tokenin kanssa. Kun Access Token vanhenee, käyttöliittymä voi käyttää Refresh Tokenia hankkiakseen uuden Access Tokenin ilman, että käyttäjän tarvitsee syöttää uudelleen kirjautumistietojaan. Refresh Tokenia tulee käsitellä erityisen turvallisesti, sillä se antaa mahdollisuuden uusien Access Tokenien

ilman suoraa käyttäjän interaktiota. Tämä mahdollistaa pitkäaikaisemman istunnon ylläpidon ilman jatkuvaa uudelleenkirjautumista. (RFC 6749.)

Käydään seuraavaksi läpi sovelluksen logiikkaa skenaariossa, jossa käyttäjä kirjautuu sovellukseen. Käyttäjän kirjautuessa sovellukseen käyttöliittymän kautta, käyttöliittymä lähettää HTTP-pyyntöön palvelinpäähän. Tämä HTTP-pyyntö sisältää käyttäjän käyttäjätunnuksen ja salasanan JSON-muodossa. Palvelinpää vastaanottaa pyynnön ja etsii tietokannasta käyttäjänimeä, jonka se sai asiakaspään pyynnöstä. Käyttäjänimen löytyessä tietokannasta pyydetään käyttäjän tietokantaan tallennettu salattu salasana. Seuraavaksi palvelinpää vertaa salattua salasanaa bcrypt.js-kirjaston avulla asiakaspäästä saamaansa salasanaan. Salasanojen täsmätessä sovellus lähettää käyttöliittymälle HTTP-vastauksen, joka sisältää pääsytunnuksen JSON-muodossa ja päivitystunnuksen HTTP-turvallisen evästeen avulla.

Seuraavassa kuvassa (Kuva 5) on visualisoitu sovelluksen autentikointi prosessi.



Kuva 5. Havainnollistava kuva käyttäjän tunnistautumisesta (Kurkinen 2023)

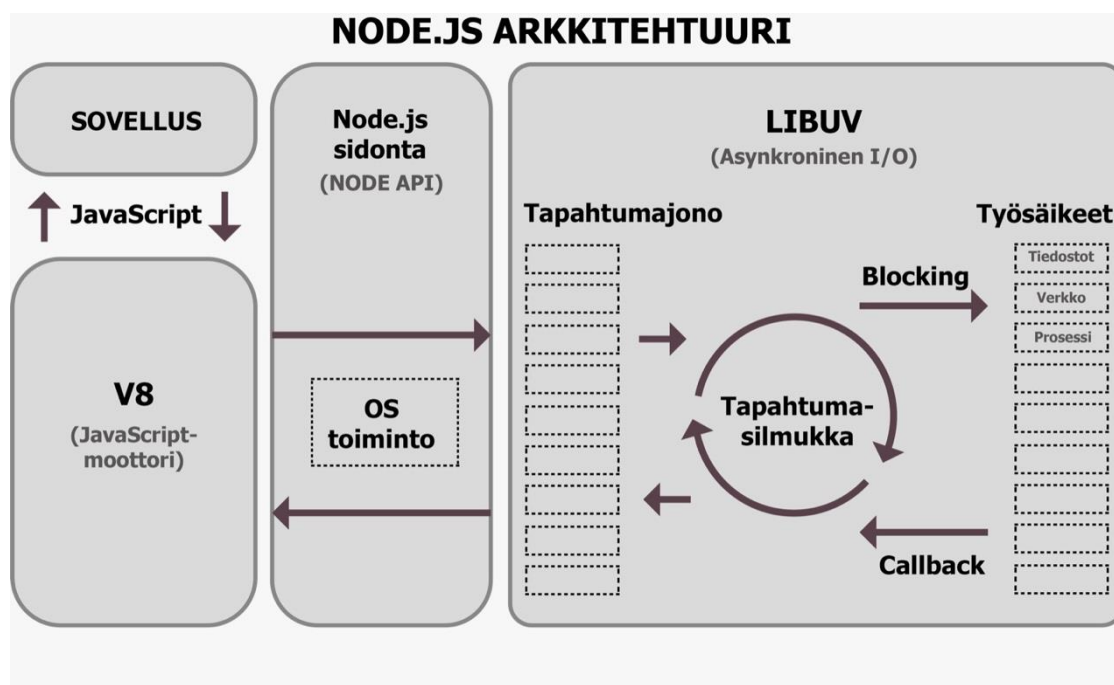
3.2.2 Node.js

Node.js on kevyt ja nopea JavaScript-ympäristö, joka soveltuu palvelin- ja työpöytäsovelluksiin. Käsiteltäessä verkkopalvelimia, keskeisiä huomioon otettavia tekijöitä ovat skaalautuvuus, viive ja suorituskyky. Node.js erottuu perinteisestä tavasta käsitellä palvelupyyntöjä pyrkimällä pitämään viiven

pienenä ja suorituskyvyn korkeana. Tämä saavutetaan käyttämällä lähestymistapaa, jossa pyyntöjen käsittelijä ei odota passiivisesti I/O-pyyntöjen valmistumista, vaan sallii muiden tehtävien käsittelyn samanaikaisesti. (Martin Heller 2020, 1.)

Perinteisesti jokainen saapuva palvelupyyntö tai yhteys aiheuttaa uuden suoritusketjun tai prosessin luomisen pyynnön käsittelyä varten. Vaikka tämä teoriassa on järkeenkäypää, käytännössä se aiheuttaa ylimääräistä kuormitusta. Säikeiden luominen on tehokkaampaa kuin prosessien haarautuminen, mutta se voi silti olla resurssitehotonta ja vaikuttaa negatiivisesti järjestelmän suorituskykyyn. (Martin Heller 2020, 1–2.)

Node.js käyttää yksisäikeistä tapahtumasykliä, joka rekisteröidään järjestelmään jokaisen uuden yhteyden osalta. Jokainen uusi yhteys käynnistää JavaScript-palautustoiminnon, joka käsittelee pyyntöjä samalla mahdollistaen estävien tai suorituskykyä vaativien tehtävien jakamisen suorittimen ytimien välillä. (Martin Heller 2020, 2.)



Kuva 6. Node.js arkkitehtuuri (Kurkinen 2023)

Node.js tarjoaa merkittävää hyötyä työpöytäsovelluksille ja palvelimille. On huomionarvoista, että Node-sovellukset eivät ole sidottuja yksinomaan JavaScriptiin. Node-sovelluksia voidaan kirjoittaa useilla eri kielillä, jotka muunnetaan JavaScriptiksi, kuten TypeScript ja CoffeeScript. Node.js sisältää Google Chromen V8 JavaScript-moottorin, joka tukee ECMAScript 2015 (ES6) syntaksia ilman tarvetta käyttää ES6-ES5-muuntajaa, kuten Babel. (Martin Heller 2020, 2.)

Suurin osa Noden hyödyllisyydestä johtuu sen laajasta pakettiluettelosta, joka on käytettävissä npm-komennon avulla. NPM, Node-paketinhallinta, sisältyy standardina Node.js-asennukseen, vaikka sillä on myös oma verkkosivusto. (Martin Heller 2020, 2.)

3.2.3 Express.js

Express.js on verkkosovelluskehys, joka perustuu Node.js:ään. Se tarjoaa käyttäjilleen yksinkertaisen käyttöliittymän sekä kaikki tarvittavat työkalut web-sovellusten kehittämiseen. Express.js tuo

joustavuutta sovelluskehitykseen, koska npm (Node Package Manager) tarjoaa laajan valikoiman moduuleja, jotka voidaan helposti integroida Expressiin. Tämä helpottaa palvelimen ja reittien välisen tiedonsiirron hallintaa palvelinpuolen sovelluksissa. Sovelluskehityksen kehittäjänä toimi alun perin TJ Holowaychuk, ja Express.js julkaistiin ensimmäisen kerran vuonna 2010. Aiemmin IBM hallinnoi sitä, mutta tällä hetkellä se kuuluu Node.js-säätiön alaisuuteen. Express.js on avoimen lähdekoodin ohjelmisto ja se on lisensoitu MIT-lisenssillä. (Zulaikha Geer 2019.)

3.3 Sovelluksen riippuvuudet

Kalapäiväkirja-sovelluksessa käytetään useita ohjelmistokirjastoja, niin käyttöliittymässä kuin serverinpäässä, joiden tarkoitus on helpottaa ohjelmointitaakkaa. Projektin ohjelmistokirjastot on asennettu npm-pakettien hallintatyökalulla. Näitä kirjastoja kutsutaan myös npm-paketeiksi.

Noden ohella npm on yksi tärkeimmistä työkaluista Node-sovellusten kehityksessä. npm:ää käytetään hakemaan kaikki tarvittavat paketit, eli JavaScript-kirjastot, sovelluksen kehitystä, testausta ja tuotantoa varten. Lisäksi npm:ää voidaan käyttää testien suorittamiseen ja tarvittavien kehitystyökalujen käynnistämiseen kehitysprosessissa. (MDN Web Docs 2023.)

Npm:ää voidaan käyttää manuaalisesti, ja jokainen tarvittava paketti voidaan hakea erikseen. Yleisesti kuitenkin riippuvuudet hallitaan käyttämällä tekstitiedostoa nimeltään package.json. Tämä tiedosto luettelee kaikki tietyn JavaScript-paketin riippuvuudet, mukaan lukien paketin nimi, versio, kuvaus, alkuperäinen suoritettava tiedosto, tuotantoriippuvuudet, kehitysriippuvuudet, Node-versiot, joiden kanssa se voi toimia ja niin edelleen. (MDN Web Docs 2023.)

Npm-paketti voi olla joko koko JavaScript-kirjasto tai kehys, kuten React tai Vue, tai sitten hyvin pieni apuohjelma, kuten kirjasto, joka muuttaa päivämäärän esitystapaa. Lisäksi se voi olla myös koodin muotoilu- ja tarkastustyökalu kuten Prettier tai ESLint. (MDN Web Docs 2023.)

Seuraava kuva (Kuva 7) on kuvankaappaus sovelluksen palvelinpään package.json-tiedostosta.


```

package.json > ...
1  {
2    "name": "server",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start": "nodemon app.js"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "bcrypt": "^5.1.0",
15     "cloudinary": "^1.40.0",
16     "cookie-parser": "^1.4.6",
17     "cors": "^2.8.5",
18     "date-fns": "^2.30.0",
19     "express": "^4.18.2",
20     "jsonwebtoken": "^9.0.1",
21     "mysql2": "^3.6.0"
22   },
23   "devDependencies": {
24     "dotenv": "^16.3.1",
25     "nodemon": "^3.0.1"
26   }
27 }
28

```

Kuva 7. Esimerkki kuva package.json tiedostosta (Kurkinen 2023)

3.3.1 Palvelinpuolen riippuvuudet

Sovelluksen palvelinpuoli käyttää seuraavia riippuvuuksia.

- bcrypt, jota käytetään sovelluksessa tiedon salaamiseen.
- cloudinary. Käytetään sovellukseen ladattavien kuvien lisäämisessä cloudinary-pilvipalveluun.
- cookie-parser. Auttaa käsittelemään HTTP-evästeitä.
- cors. Käytetään sovelluksessa helpottamaan Cross-Origin Resource Sharing (CORS) -käytännöjen hallintaa.
- date-fns. Käytetään muokkaamaan päivämäärän esitystapaa.
- express. Käytetään yksinkertaistamaan ja nopeuttamaan palvelinpään ohjelmointia.
- jsonwebtoken. Käytetään sovelluksen autentikoinnissa.
- mysql2. Käytetään MySQL tietokannan käsittelyyn palvelinpäässä.

Sovelluksen palvelinpuoli käyttää seuraavia kehitysajan riippuvuuksia

- dotenv. Käytetään sovelluksen ympäristömuuttujien hallintaan.
- nodemon. Käytetään helpottamaan serverinpään kehitystä.

3.3.2 Käyttöliittymän riippuvuudet

Sovelluksen käyttöliittymä käyttää seuraavia riippuvuuksia

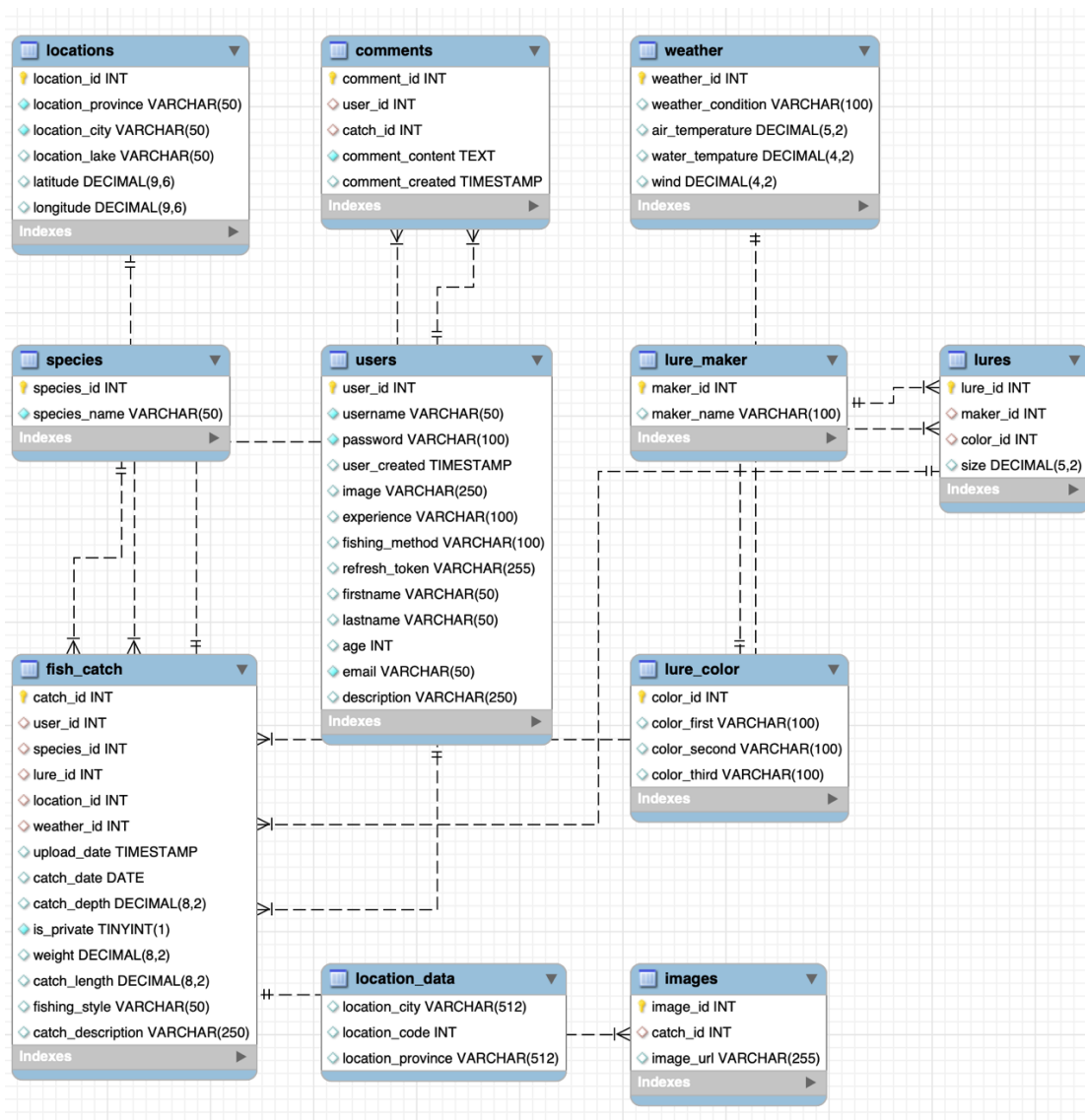
- @react-google-maps/api. Tarjoaa sovellukselle Google-Maps palvelut.
- @tabler/icons-react. Ikoni-kirjasto, jonka tarjoamia ikoneita käytetään eri osissa käyttöliittymää.
- axios. Käytetään serverinpäähän lähetettävissä HTTP-pyyntöissä.
- prop-types. Käytetään varmistamaan, että komponenteille annetut "props"-arvot ovat odotettua tyyppiä ja että ne täyttävät mahdolliset validointiehdot.
- react. Käytetään käyttöliittymän rakentamiseen.
- react-chartjs-2. Käytetään sovelluksen kaavioiden ja diagrammien luomiseen.
- react-dom. Tulee React: in asennuksen mukana. Käytetään DOM: n manipulointiin.
- react-router-dom. Kirjasto, jota käytetään sovelluksen reitityksessä.
- react-spinners. Käytetään käyttöliittymän osissa, joissa tarvitaan latausindikaattoria.
- swiper. Tarjoaa käyttöliittymän kuvakarusellit.

Sovelluksen käyttöliittymä käyttää seuraavia sovellusajan riippuvuuksia

- eslint. Koodianalysaattori. Auttaa noudattamaan projektille asetettuja koodaustyyliä.
- tailwindcss. Käytetään käyttöliittymän tyylittelyyn.
- vite. Käytetään projektin JavaScript rakennustyökaluna.

3.4 Tietokanta

Kalapäiväkirja sovellus käyttää MySQL-tietokantaa. Kaikki tieto on tallennettu yhteen tietokantaan ja osa tietokannan tauluista pelkästään tarjoaa tietoa käyttöliittymälle. Nämä taulut toimivat ikään kuin tietopankkina, joita ei sovelluksesta käsin voi muokata, poistaa tai lisätä, ainoastaan noutaa.



Kuva 8. Tietokannan ER-kaavio (Kurkinen 2023)

Edellä mainittuja tauluja ovat esimerkiksi tietokannan ER-kaaviokuvan (Kuva 8) "location_data"-taulu, joka tarjoilee käyttäliittymään Suomen maakunnat ja kunnat, sekä "lure_maker"-taulu, joka tarjoilee käyttäliittymään kattauksen tunnettuja viehevalmistajia.

3.4.1 MySQL

Oraclen julkaiseman artikkelin mukaan MySQL on maailman suosituin avoimen lähdekoodin tietokanta. DB-Enginesin mukaan MySQL sijoittuu toiseksi suosituimmaksi tietokannaksi Oracle-tietokannan jälkeen. Tietokannat ovat olennainen tietovarasto kaikille ohjelmistoille. Esimerkiksi aina kun joku suorittaa verkkohaun, kirjautuu tililleen tai tekee tapahtuman, tietokantajärjestelmä tallentaa tiedot, jotta niitä voidaan käyttää myöhemmin. (Oracle julkaisuaika tuntematon.)

Relaatiotietokanta tallentaa tiedot erillisiin taulukoihin sen sijaan, että kaikki tiedot olisivat yhdessä suuressa varastossa. Tietokannan rakenne on järjestetty fyysisiin tiedostoihin nopeuden optimoimiseksi. Looginen tietomalli, jossa on objekteja kuten tietotaulukoita, näkymiä, rivejä ja sarakkeita, tarjoaa joustavan ohjelmointiympäristön. Voit määritellä säännöt eri tietokenttien välisille suhteille, kuten yksi-yhteen, yksi-moneen, yksilöllinen, pakollinen tai valinnainen, sekä "osoittimet" eri taulujen välillä. Tietokanta noudattaa näitä sääntöjä, joten hyvin suunnitellulla tietokannalla sovellus ei

koskaan näe ristiriitaista, päällekkäistä, irrallaan olevaa, vanhentunutta tai puuttuvaa tietoa. (Oracle julkaisuaika tuntematon.)

SQL on rakenteellinen kyselykieli, jonka nimi tulee sanoista "Structured Query Language". SQL on yleisin standardoitu kieli tietokantojen käyttämiseen. Riippuen ohjelmointiympäristöstä SQL:ta voi syöttää suoraan (esimerkiksi raporttien luomiseksi), upottaa SQL-lauseet toiseen kieleen kirjoitetun koodin joukkoon tai käyttää kielikohtaista ohjelmointirajapintaa (API), joka piilottaa SQL:n syntaksin. (Oracle julkaisuaika tuntematon.)

```
SELECT * FROM user WHERE username = "Example name";
```

Kuva 9. Yksinkertainen esimerkki SQL-syntaksista (Kurkinen 2023)

3.5 Pilvipalvelut


Kalapäiväkirja käyttää cloudinary pilvipalvelua tallennuspaikkana sovellukseen ladattaville kuville. Sovelluksen palvelinpää hyödyntää cloudinaryn tarjoamaa npm-pakettia, joka mahdollistaa palvelimen yhdistämisen pilvipalveluun. Seuraava kuvankaappaus sisältää palvelinpään JavaScript-tiedoston, jota palvelin käyttää yhdistyäkseen cloudinaryn kanssa.

```
JS cloudinary.js > ...
1  const cloudinary = require('cloudinary').v2;
2  require("dotenv").config();
3
4  cloudinary.config({
5    cloud_name: process.env.CLOUD_NAME,
6    api_key: process.env.API_KEY,
7    api_secret: process.env.API_SECRET
8  });
9
10 module.exports = cloudinary;
```

Kuva 10. Kuvankaappaus cloudinary.js tiedostosta (Kurkinen 2023)

Cloudinary mahdollistaa ladattavan kuvan muokkauksen, kuvaa lähetettäessä palvelimelta pilvipalveluun. Cloudinaryn tarjoama npm-paketti sisältää uploader-metodin, johon parametreina syötetään ladattava tiedosto sekä JavaScript-objekti, joka sisältää mahdolliset kuvanmuokkaus valinnat. Seuraavasta kuvasta on nähtävissä sovelluksen serverinpään funktio, joka suorittaa kuvien lisäämisen cloudinary pilvipalveluun.

```

helpers >  imgToCloudinary.js > ...
1  const cloudinary = require('../cloudinary');
2
3  const handleCloudinaryUp = async (imgs) => {
4      const cloudinaryUploadPromises = imgs.map(async (file) => {
5          const result = await cloudinary.uploader.upload(file, {
6              folder: "Kalapaivakirja/uploads",
7              format: "jpg",
8              resource_type: "auto",
9              height: 920,
10             crop: "scale"
11          });
12          return result.secure_url;
13      });
14      const uploadedUrls = await Promise.all(cloudinaryUploadPromises);
15      return uploadedUrls;
16  };
17  module.exports = { handleCloudinaryUp };
18

```

Kuva 11. Kuvankaappaus imgToCloudinary.js-tiedostosta (Kurkinen 2023)

Funktio palauttaa ladattujen kuvien URL-osoitteet taulukossa, jotka sovellus myöhemmin tapahtumaketjussa tallentaa tietokantaan.

4 KEHITYSTYÖKALUT JA VERSIONHALLINTA

Kalapäiväkirja-sovelluksen kehitystyössä on käytetty useita eri kehitystyökaluja ja apuvälineitä kehitystyön helpottamiseksi. Projektin koodieditorina toimi Visual Studio Code. Visual Studio Code on kevyt mutta suorituskykyinen työpöytäpohjainen koodieditori, joka on saatavilla Windowsille, macOS:lle sekä Linuxille. (Visual Studio Code julkaisuaika tuntematon.)

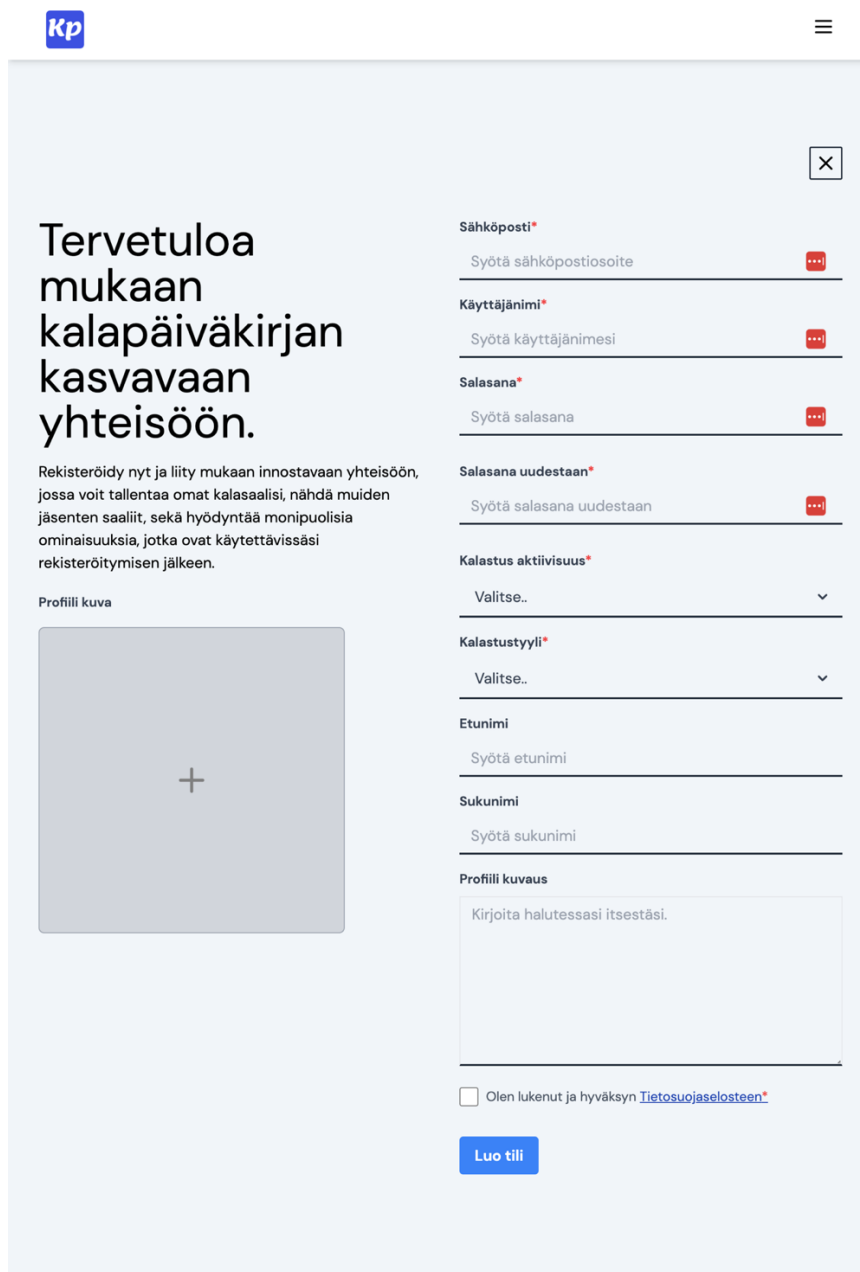
Kehitystyön aikaisena käyttöjärjestelmänä käytössä oli macOS. Serverinpään rajapintojen testaamiseen käytin Insomnia nimistä avoimen lähdekoodin HTTP- ja REST-asiakasohjelmaa. Insomnia on kevyt ja kätevä työkalu, jolla voidaan lähettää HTTP-pyyntöjä rajapintoihin.

Projektin versionhallintaan käytin GitLab-palvelua. GitLab tarjoaa esimerkiksi Git-versionhallinnan sekä useita tehtävänhallintatoimintoja. Projektin palvelinpuolella ja käyttöliittymällä oli käytössä omat säilytyspaikat (repositoryt) GitLab:issa.

5 ULKOASU JA NÄKYMÄT

5.1 Käyttäjätunnuksen luonti

Vierailevan käyttäjän on mahdollista luoda sovellukseen käyttäjätili sovelluksen etusivulla sijaitsevasta "Luo tunnus"-painikkeesta tai kirjautumislomakkeen "Puuttuuko tili?"-linkistä. Molemmat vaihtoehdot navigoivat sovelluksessa sivulle, joka sisältää tilin luontilomakkeen. Lomakkeen pakolliset kentät ovat: sähköpostiosoite, käyttäjänimi, salasana, kalastus aktiivisuus ja kalastustyyli. Sähköpostin ja käyttäjänimen tulee olla uniikkeja. Palvelin tarkistaa, ettei em. tietoja löydy tietokannasta ennen uuden käyttäjätilin luomista. Lomakkeen vapaavalintaisia kenttiä ovat: Profiili kuva, etunimi, sukunimi ja profiili kuvaus. Lomake vaatii myös tietosuojaselosteen hyväksynnän. Kun kaikki lomakkeen vaatimat tiedot on syötetty, "Luo tili" -painike aktivoituu ja tiedot voidaan lähettää palvelinpuolelle käyttäjätilin luomista varten. Sovellus tarkastaa vaadittujen tietojen oikeellisuuden käyttöliittymän sekä serverinpuolella. Esimerkiksi sähköpostiosoitteen tulee olla oikeaa muotoa ja salasanan tulee olla vähintään seitsemän merkkiä pitkä.



Kp

Tervetuloa mukaan kalapäiväkirjan kasvavaan yhteisöön.

Rekisteröidy nyt ja liity mukaan innostavaan yhteisöön, jossa voit tallentaa omat kalasaalis, nähdä muiden jäsenten saaliit, sekä hyödyntää monipuolisia ominaisuuksia, jotka ovat käytettävissäsi rekisteröitymisen jälkeen.

Profiili kuva

+

Sähköposti*

Syötä sähköpostiosoite

Käyttäjänimi*

Syötä käyttäjänimesi

Salasana*

Syötä salasana

Salasana uudestaan*

Syötä salasana uudestaan

Kalastus aktiivisuus*

Valitse..

Kalastustyyli*

Valitse..

Etunimi

Syötä etunimi

Sukunimi

Syötä sukunimi

Profiili kuvaus

Kirjoita halutessasi itsestäsi.

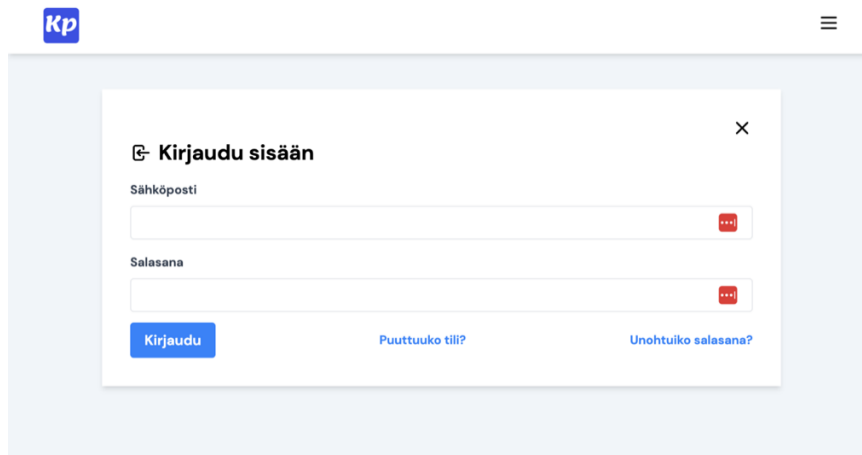
☐ Olen lukenut ja hyväksyn [Tietosuojaselosteen](#)*

Luo tili

Kuva 12. Tunnuksen luonti (Kurkinen 2023)

5.2 Sovellukseen kirjautuminen

Sovelluksen kirjautumisnäkymään päästään yläreunan navigointipalkin "Kirjaudu sisään"-painikkeesta. Kirjautumislomake vaatii yksinkertaisuudessaan syötettäväksi käyttäjän sähköpostiosoitteen ja salasanan. Kun vaaditut tiedot on syötetty, "Kirjaudu"-painike aktivoituu ja kirjautumisyritys voidaan lähettää serverinpäähän. Lomake sisältää myös linkin, joka ohjaa vierailijan tunnuksen luontinäkymään, käyttäjätilin puuttuessa.



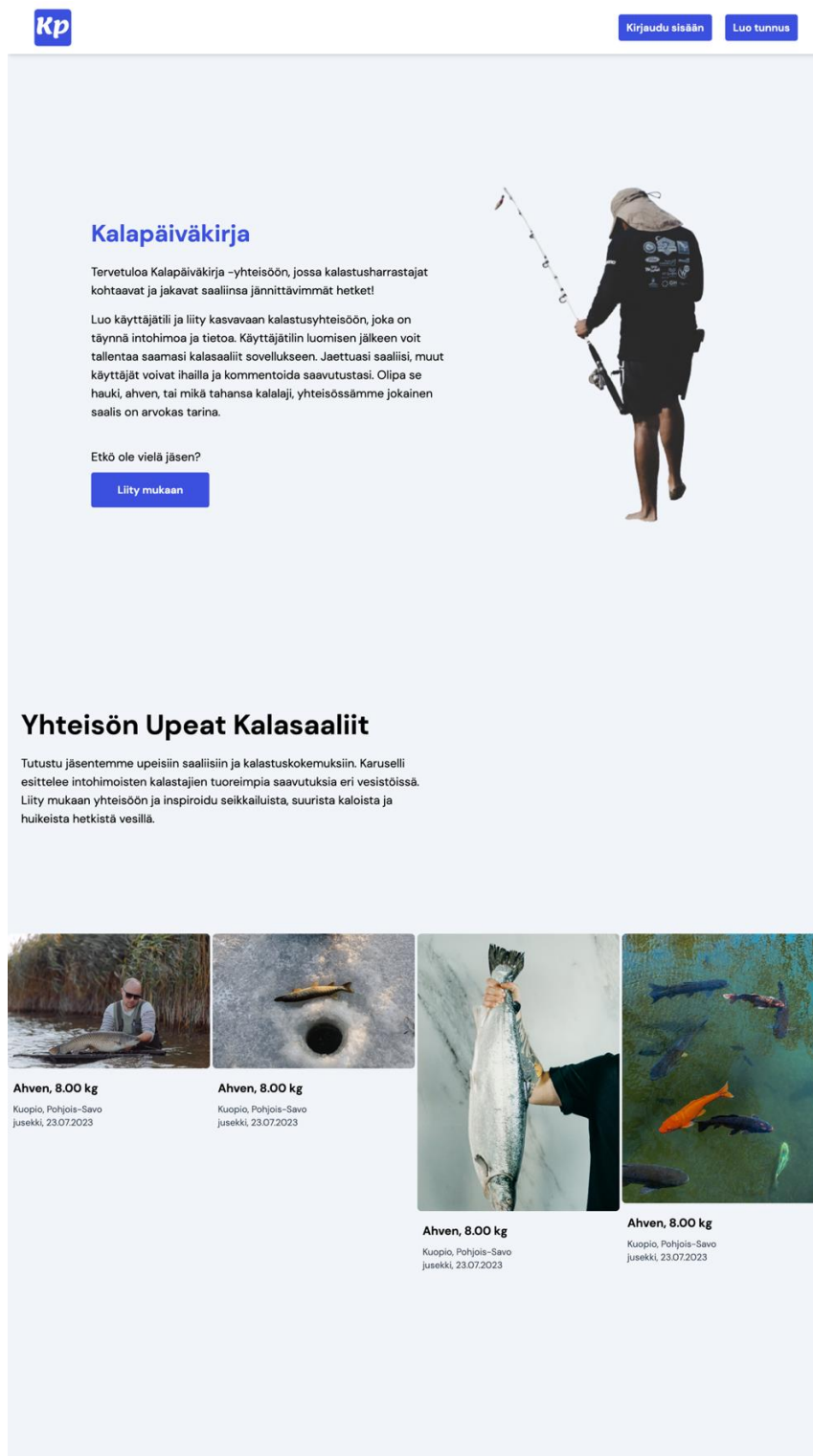
Kuva 13. Kirjautumislomake (Kurkinen 2023)

5.3 Sovelluksen etusivu

Vierailijan saapuessa Kalapäiväkirja-sovellukseen hän näkee ensimmäisenä sovelluksen etusivun. Sovellukseen kirjautumattomalle käyttäjälle etusivu ei tarjoa muuta toiminnallisuutta kuin mahdollisuuden kirjautua ja luoda käyttäjätilin. Etusivu on ikään kuin mainos sovelluksesta, jonka tarkoituksena on houkutella vierailija sovelluksen käyttäjäksi. Näkymässä kerrotaan toiminnallisuuksista, jota sovellus tarjoaa käyttäjilleen. Etusivu esittelee myös kuusitoista viimeisintä sovellukseen lisättyä kalasaalista listattuna automaattisesti liikkuvaan karuselliin.

5.3.1 Näkymä kirjautumattomalle käyttäjälle

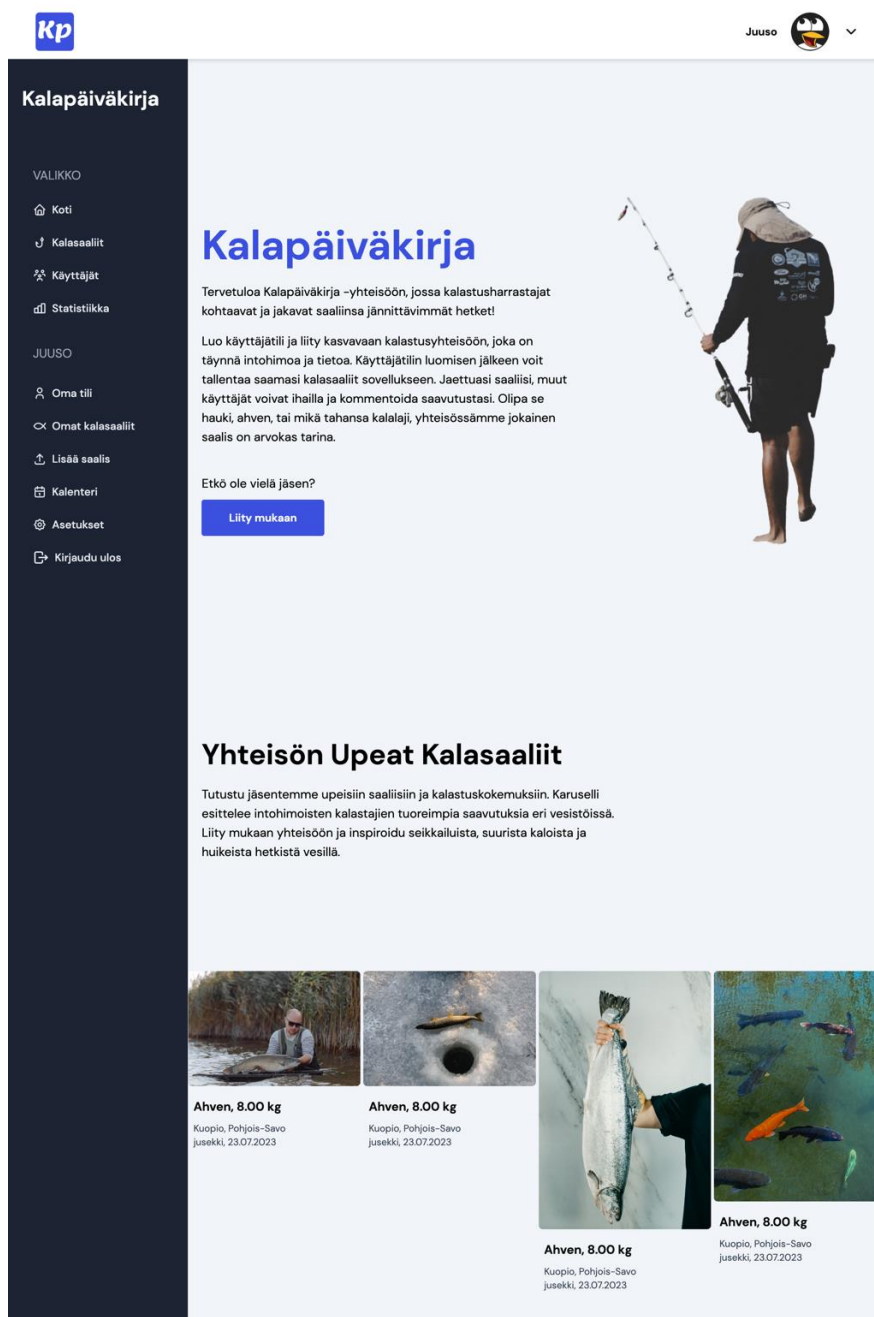
Näkymä kirjautumattomalle käyttäjälle sisältää mahdollisuuden luoda käyttäjätunnuksen ja kirjautumaan sovellukseen. Näkymällä pyritään houkuttelemaan uusia käyttäjiä ja siksi tarjoaa karusellin uusimmista sovelluksen kalasaaliista.



Kuva 14. Etusivu kirjautumattomalle käyttäjälle (Kurkinen 2023)

5.3.2 Näkymä kirjautuneelle käyttäjälle

Kirjautunut käyttäjä näkee näkymän vasemmassa reunassa navigointi valikon, joka mahdollistaa liik-
kumisen eri näkymiin ja sovelluksen toiminnallisuuksiin.

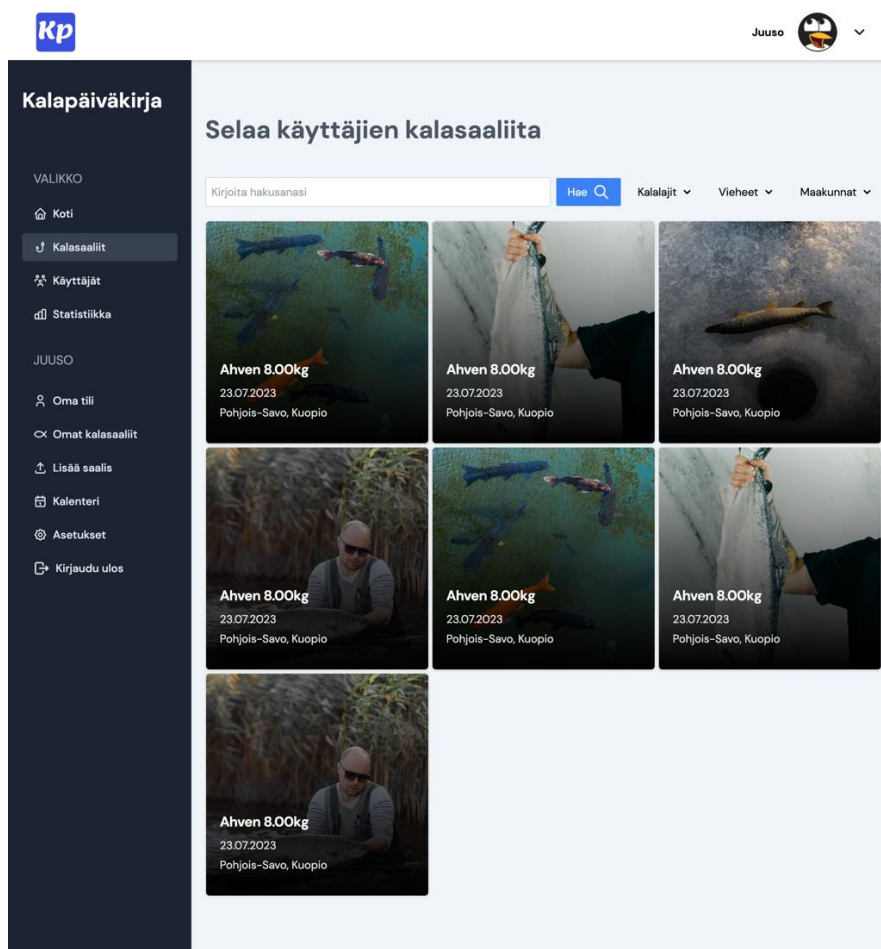


Kuva 15. Etusivu kirjautuneelle käyttäjälle (Kurkinen 2023)

5.4 Kalasaaliit

Sovelluksen kalasaaliit-näkymässä kirjautunut käyttäjä voi selata sovellukseen lisättyjä, käyttäjien julkiseksi asettamia kalasaaliita. Näkymässä yksittäinen kalasaalis esitetään korttityylisesti, jonka taustakuvana on kalasaaliin kuva. Kortti sisältää saaliin kalalajin, painon, päivämäärän ja sijaintitiedot.

Kalasaaliiden listaus on toteutettu modernilla "Infinite Scroll" -tekniikalla. Käytännössä tämä tarkoittaa, että sovellus noutaa serverinpäästä kerrallaan kaksikymmentä saalista. Kun näkymää rullataan alaspäin viimeiseen saaliiseen asti, sovellus pyytää serverinpäästä seuraavat kaksikymmentä saalista jne. Infinite Scroll -tekniikka parantaa käyttöliittymän suorituskykyä ja käyttömukavuutta. Näkymässä käyttäjä voi suodattaa kalasaaliita kalalajin, viehevalmistajan ja maakunnan perusteella.



Kuva 16. Kalasaaliit-näkymä (Kurkinen 2023)

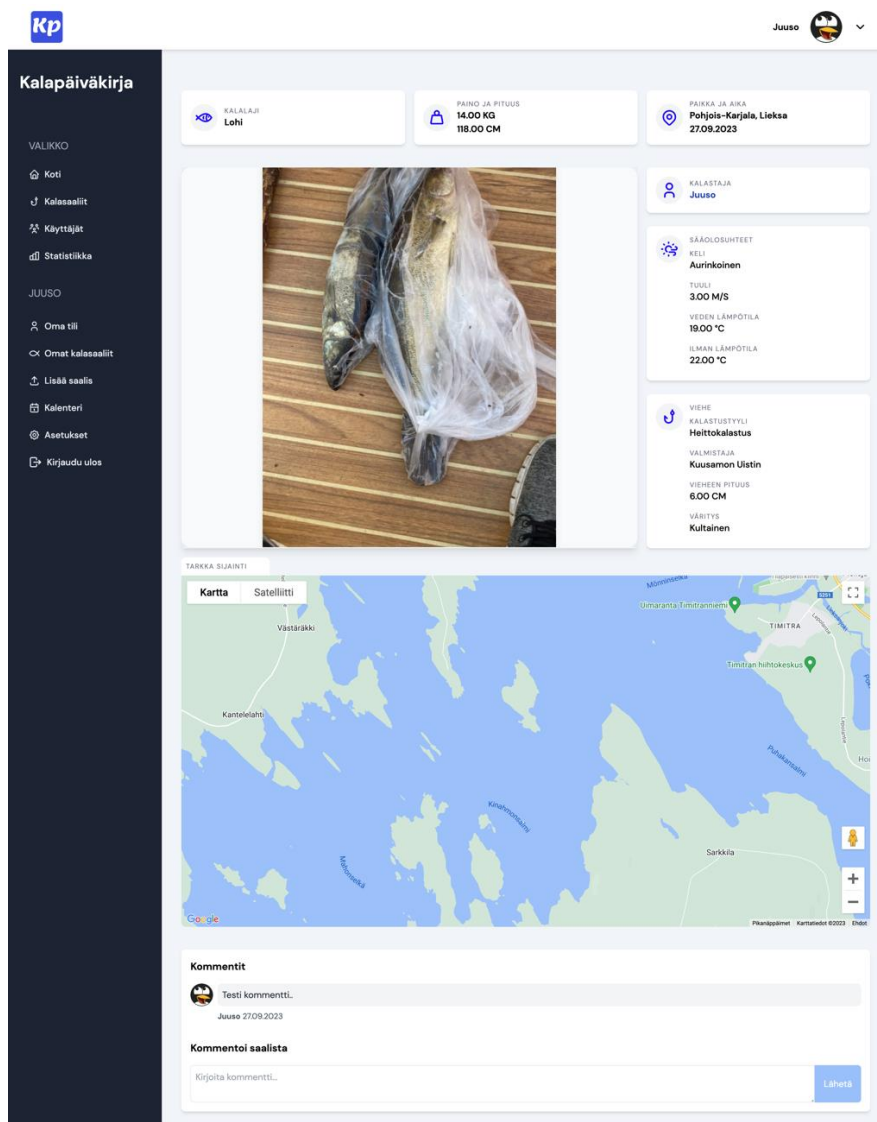
Kalasaalis-korttia klikkaamalla käyttäjä navigoidaan näkymään, joka sisältää tarkemmat tiedot klika-
tusta saaliista.

5.5 Yksittäinen kalasaalis

Yksittäisen kalasaaliin näkymä näyttää seuraavat tiedot saaliista:

1. Saaliin kalalajin
2. Saaliin painon ja pituuden
3. Kalastuspaikan ja ajan
4. Kalastajan (saaliin sovellukseen lisänneen käyttäjän)
5. Sääolosuhteet (kelin, tuulennopeuden sekä veden- ja ilmanlämpötilat)
6. Kalastustyylin ja vieheen tiedot (valmistaja, pituus sekä väritys)
7. Tarkan sijainnin saaliista Google maps-kirjaston avulla

Näkymä tarjoaa myös käyttäjille mahdollisuuden kommentoida saalista. Kommenttilistaus sisältää kommentin, profiilikuvan, käyttäjänimen sekä kommentointi ajan. Näkymässä kuvat esitellään karu-
selli tyylisesti swiper-kirjaston avulla.



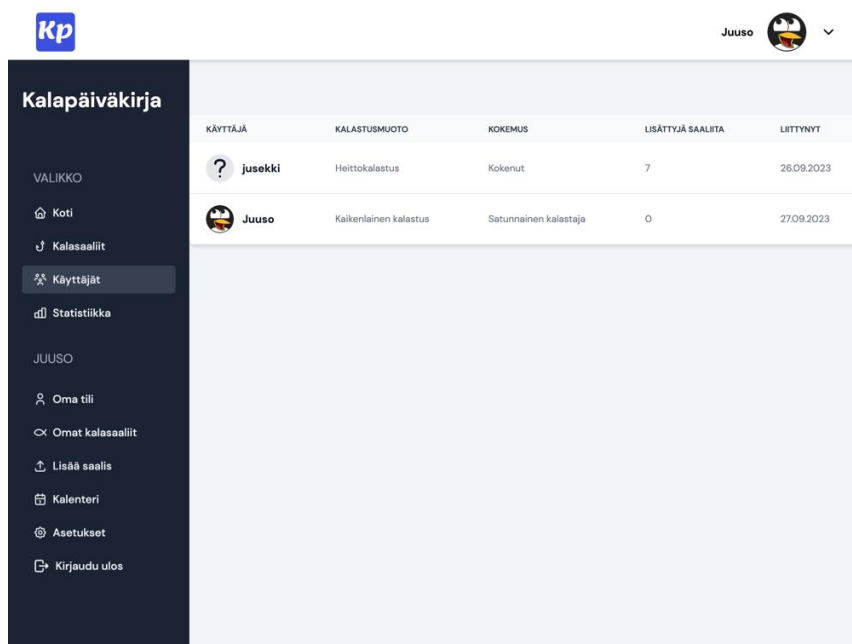
Kuva 17. Yksittäisen kalasaaliin näkymä (Kurkinen 2023)

5.6 Käyttäjät

Käyttäjät-näkymässä näkyy käyttäjät taulukkomuodossa. Jokainen taulukon rivi sisältää seuraavat tiedot käyttäjästä:

1. Käyttäjän käyttäjänimi ja kuva
2. Käyttäjän harrastama kalastusmuoto
3. Käyttäjän aktiivisuustaso
4. Käyttäjän lisäämät saaliit
5. Liittymispäivä

Taulukon riviä klikkaamalla päästään näkymään, joka sisältää perusteellisemmat tiedot valitusta käyttäjästä.

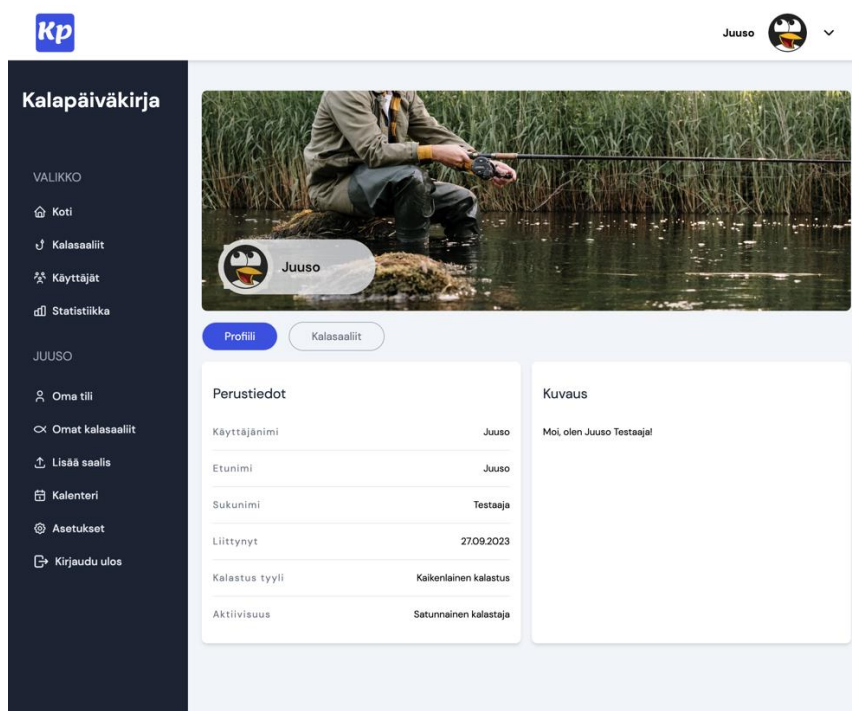


KÄYTTÄJÄ	KALASTUSMUOTO	KOKEMUS	LISÄTTYÄ SAALITA	LIITTYNYT
? jusekki	Heittokalastus	Kokenut	7	26.09.2023
Juuso	Kaikenlainen kalastus	Satunnainen kalastaja	0	27.09.2023

Kuva 18. Käyttäjät-näkymä (Kurkinen 2023)

5.7 Käyttäjän profiili

Käyttäjän profiili näkymä on yksinkertainen, sisältäen käyttäjän profiilikuvan, perustiedot ja kuvauksen. Näkymässä on "kytkin" -tyyliset painikkeet, kuten "Profiili" ja "Kalasaaliit", joilla voi vaihtaa pääalueen näkymän käyttäjätietojen ja kalasaaliiden välillä. Kalasaalisnäköymä listaa käyttäjän julkaisemat julkiset kalasaaliit.



Kp

Juuso

Kalapäiväkirja

VALIKKO

Koti

Kalasaaliit

Käyttäjät

Statistikka

JUUSO

Oma tili

Omat kalasaaliit

Lisää saalis

Kalenteri

Asetukset

Kirjaudu ulos

Juuso

Profiili

Kalasaaliit

Perustiedot

Käyttäjänimi

Juuso

Etunimi

Juuso

Sukunimi

Testaaja

Liittynyt

27.09.2023

Kalastus tyyli

Kaikenlainen kalastus

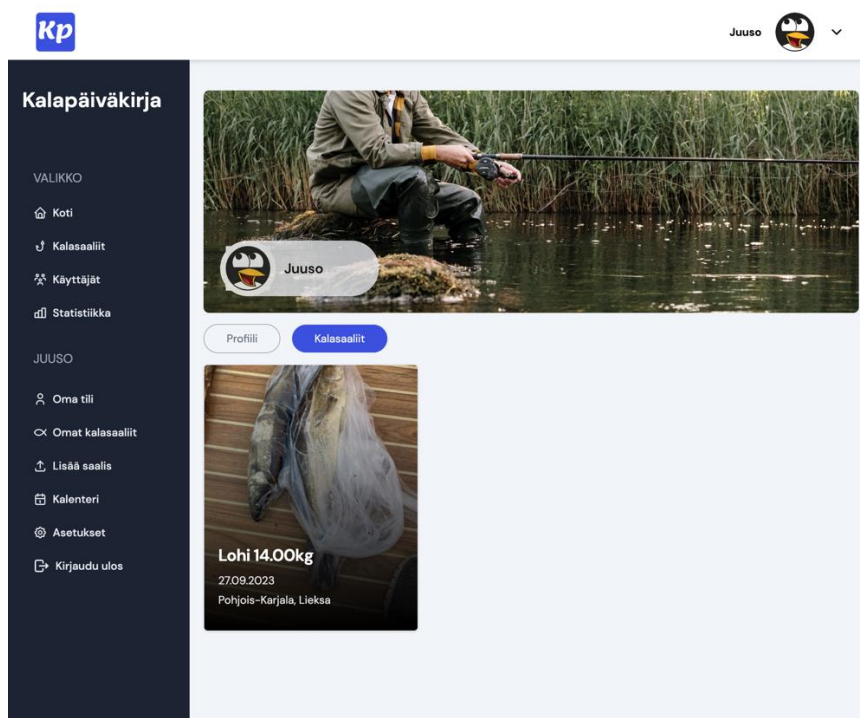
Aktiivisuus

Satunnainen kalastaja

Kuvaus

Moi, olen Juuso Testaaja!

Kuva 19. Yksittäisen käyttäjän tiedot (Kurkinen 2023)



Kuva 20. Yksittäisen käyttäjän lisäämät kalasaaliit (Kurkinen 2023)

5.8 Saaliin lisäys

Saaliin lisäys on toteutettu kolmevaiheisella lomakkeella. Lomake sisältää staattisen ohjeen lomakkeen täyttämiseen, dynaamisen vaiheindikaattorin otsikoineen sekä dynaamisen osion tiedon syöttämiselle. Lomake käyttää paljon alavetovalikoita, jotka tarjoavat valintavaihtoehtoja. Tämä helpottaa tietokantaan kerätyn tiedon analysoinnissa.

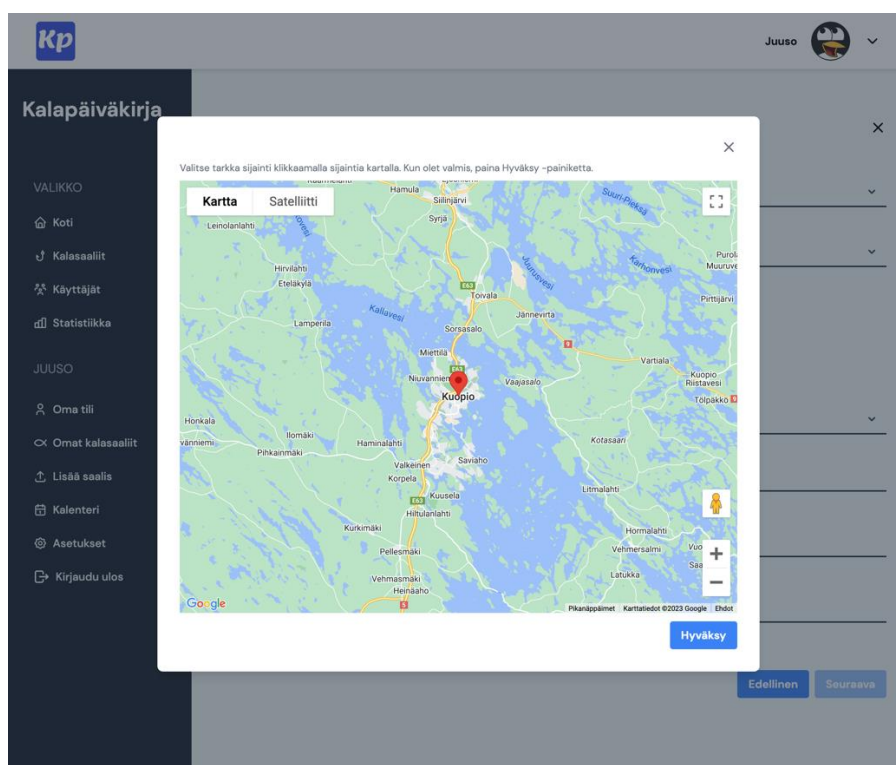
Lomakkeen ensimmäisen vaihe vaatii seuraavat tiedot lisättävästä kalasaaliista: Kalalaji, paino, kalastusmuoto, viehevalmistaja sekä vieheen väritys. Vapaavalintaisina tietoina ovat: saaliin pituus, vieheen pituus sekä uintisyvyys. Käyttäjä ei pääse lomakkeessa seuraavaan vaiheeseen ennen kuin pakolliset tiedot on syötetty.

Kuva 21. Kalasaaliin lisäyslomake, sivu 1 (Kurkinen 2023)

Lomakkeen toisessa vaiheessa ilmoitetaan saaliin sijainti ja säätiedot. Vaiheen pakollisina tietoina ovat: Maakunta, kunta, päivämäärä ja kellonaika, keli sekä ilman lämpötila. Vapaaehtoisina tietoina ovat: Saaliin tarkka sijainti, veden lämpötila sekä tuulen nopeus.

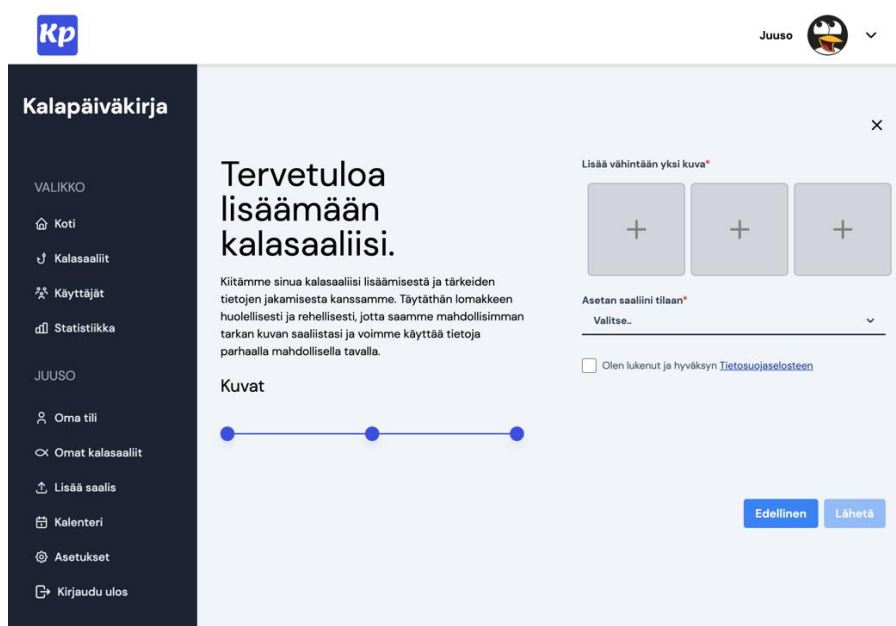
Kuva 22. Kalasaaliin lisäyslomake, sivu 2 (Kurkinen 2023)

Lomakkeen toisessa vaiheessa käyttäjän on mahdollista lisätä saaliilleen tarkka sijainti. "Avaa kartta"-painiketta painalla aukeaa modaali, joka sisältää Google maps-kartan, jota klikkaamalla voidaan merkitä kalastuspaikka. "Hyväksy"-painikkeesta sovellus tallentaa valitun sijainnin koordinaatit soveluksen tilan hallintaan, jotka myöhemmässä vaiheessa lähetetään serverinpäähän.



Kuva 23. Kalasaaliin lisäyslomake, tarkan sijainnin valinta (Kurkinen 2023)

Kalasaaliin lisäyslomakkeen viimeisessä vaiheessa saaliille voidaan lisätä kuvia sekä valita kalasaaliin tila (julkinen tai yksityinen). Tämä vaihe vaatii saaliista vähintään yhden kuvan lisättäväksi, mutta mahdollistaa myös usean kuvan lisäämisen (enintään kolme kuvaa). Tilan valinta sekä tietosuojaselosteen hyväksyntä ovat pakollisia. Kun kaikki lomakkeen pakolliset tiedot on syötetty, sallitaan lomakkeen tietojen lähettäminen serverinpäähän.

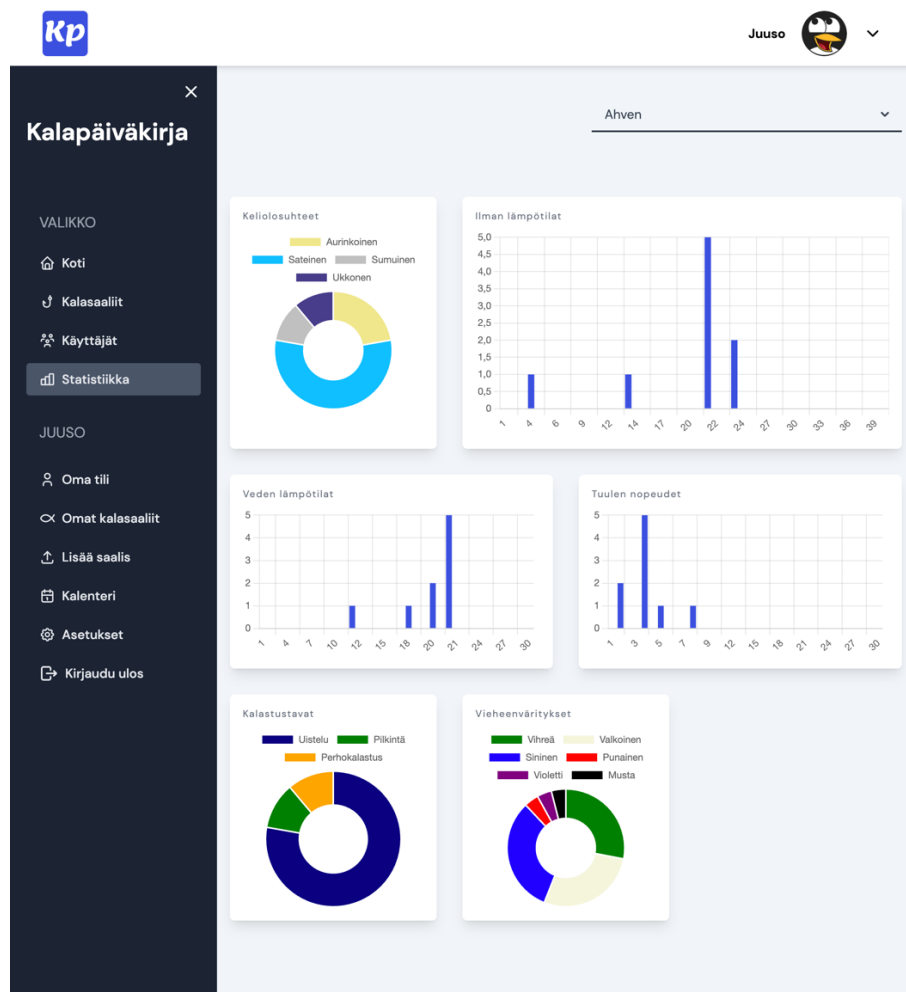


Kuva 24. Kalasaaliin lisäyslomake, sivu 3 (Kurkinen 2023)

5.9 Statistiikka

Statistiikkanäkymä hyödyntää sovellukseen lisättyä tietoa kalasaaliiden lisäämisen yhteydessä. Tässä näkymässä tiedot esitetään kalalajin perusteella, ja siihen on integroitu "react-chartjs-2" -kirjasto

diagrammien luomiseen. Käyttäjä voi valita alasvetovalikosta haluamansa kalalajin ja tarkastella sitä koskevia ympyrä- ja pylväskaavioita. Nämä kaaviot antavat tietoa kalasaaliista, kuten pyydystysmäärät, olosuhteet, viehevärit ja pyyntitavat.



Kuva 25. Sovelluksen statistiikkanäkymä (Kurkinen 2023)

Käyttäjä voi hyödyntää kaavioiden tarjoamaa tietoa omassa kalastusharrastuksessaan.

6 YHTEENVETO

Kalapäiväkirja-sovellusprojekti pysyi aikataulussa. Tekijä sekä projektin toimeksiantaja ovat tyytyväisiä sovelluksen nykytilanteeseen. Sovelluksessa käyttäjä voi nyt luoda käyttäjätilin, kirjautua sisään, lisätä, selata ja kommentoida kalasaaliita. Lisäksi käyttäjä voi julkaista saaliinsa joko julkisesti, jolloin se on kaikkien sovelluksen käyttäjien nähtävillä, tai yksityisesti, jolloin se näkyy vain lisääjälle itselleen. Käyttäjä voi tutustua muiden käyttäjien profiilisivuihin, joissa näkyvät yleiset tiedot ja käyttäjän lisäämät kalasaaliit. Lisäksi sovellus sisältää tilastosivun, jossa käyttäjä voi tarkastella sovellukseen syötettyjä tietoja kalalajin perusteella. Sovellus toimii myös responsiivisesti, käyttäjän näyttökoosta riippumatta.

Projekti täytti kaikki sille asetetut tavoitteet paitsi sen, että sovellusta ei saatu vietyä yksityiseen testikäyttöön pilvipalveluun. Projektin aikana ei ilmennyt merkittäviä viivästyksiä tai yllätyksiä. Eniten aikaa ja vaivaa kului käyttäjän tunnistautumiseen liittyvään ohjelmointiin. Tämä johtui siitä, että aihealue oli minulle suhteellisen tuntematon ennen projektin aloittamista, joten jouduin opiskelemaan sitä hieman ja halusin tehdä sen huolellisesti varmistaakseni, että tunnistautuminen olisi tuotantokelpoisella tasolla. Toinen suuri haaste oli äärettömän vierityksen toteutus, joka oli myös minulle aiemmin tuntematon, joten jouduin opiskelemaan sen toteutuslogiikkaa jonkin verran. Loppujen lopuksi olin kuitenkin tyytyväinen siihen, miten molemmat edellä mainitut osa-alueet saatiin mielestäni toteutettua virheettömiksi.

Sovellus saataessa tulevaisuudessa mahdollisesti tuotantoon asti, tulee helpottamaan kalastajien saaliiden merkkauksessa muistiin. Sovellus tulee myös tarjoamaan mahdollisen yhteisön kalastajille, jossa kalastajat voivat halutessaan jakaa omia kalastus paikkojaan ja esitellä saaliitaan. Sovelluksella on vielä kuitenkin pitkä matka edessä, jotta se olisi tuotantokelpoisella tasolla.

Loppujen lopuksi projektin toteuttaminen oli minulle mieluista ja todennäköisesti tulevaisuudessa tulen työskentelemään projektin jatkokehityksen parissa, jotta projekti saataisiin tuotantoon asti. Projektin aikana opin paljon uutta ja sain uusia näkökulmia sovelluskehitykseen, jotka varmasti hyödyttävät minua tulevaisuudessa.

LÄHTEET

Mosh Hamedani 2018. What Is React (React.js) & Why Is It So Popular? Video. YouTube-videopalvelu, julkaistu 27.6.2018. <https://youtu.be/N3AkSS5hXMA?si=A1hEngNVflmomFk3>. Viitattu 25.9.2023.

Stack Overflow Survey 2023. Most Popular Technologies. Verkkojulkaisu. <https://survey.stackoverflow.co/2023/#most-popular-technologies-webframe>. Viitattu 25.9.2023.

Zulaikha Geer 2019. Express.js Fundamentals. Verkkojulkaisu. <https://medium.com/edureka/expressjs-tutorial-795ad6e65ab3>. Viitattu 25.9.2023.

RFC 6749 2012. OAuth 2.0 Authorization Framework. Verkkojulkaisu. <https://datatracker.ietf.org/doc/html/rfc6749>. Viitattu 23.11.2023.

Heller, Martin 2020. What is Node.js? The JavaScript runtime explained. Verkkojulkaisu. https://courses.cs.duke.edu/compsci290/cps196/compsci290/spring21/readings/What_is_Node.js_InfoWorld.pdf. Viitattu 10.10.2023.

MDN Web Docs 2023. Setting up a Node development environment. Verkkojulkaisu. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/development_environment. Viitattu 29.9.2023.

MDN Web Docs 2023. Package management basics. Verkkojulkaisu. https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Understanding_client-side_tools/Package_management. Viitattu 29.9.2023.

Oracle julkaisuaika tuntematon. What is MySQL?. Verkkojulkaisu. <https://www.oracle.com/mysql/what-is-mysql/>. Viitattu 29.9.2023.

Visual Studio Code julkaisuaika tuntematon. Getting Started. Verkkojulkaisu. <https://code.visualstudio.com/docs>. Viitattu 2.10.2023.