# Suunnittelu

## Visio

Visiona oli luoda kalenterisovellus, jossa olisi helppokäyttöisiä ja selkeitä ominaisuuksia ryhmätapaamisten ja tapahtumien luomiseksi. Ajatuksena oli, että pohjalla olisi jokaisen käyttäjän henkilökohtainen kalenteri, johon käyttäjä voisi luoda omia merkintöjä ja linkittää niihin muistiinpanoja. Sovelluksessa olisi mahdollista myös luoda tapahtumia, joihin pystyisi kutsumaan muita sovelluksen käyttäjiä. Käyttäjien olisi mahdollista muodostaa ryhmiä, jonka jäsenien kalenterit saisi sulautettua yhteen ryhmäkalenteriin yhdellä klikkauksella. Näin käyttäjän olisi helppo katsoa kaikille sopiva ajankohta tapaamiselle. Meillä oli alusta asti tiedossa, että sovelluksessamme tulee olemaan paljon samaa kuin Googlen omassa kalenterissa. Halusimme kuitenkin luoda oman itsenäisen ja helppokäyttöisen sovelluksen, joka palvelisi optimaalisesti juuri meidän tarpeitamme.

## Ulkoasu

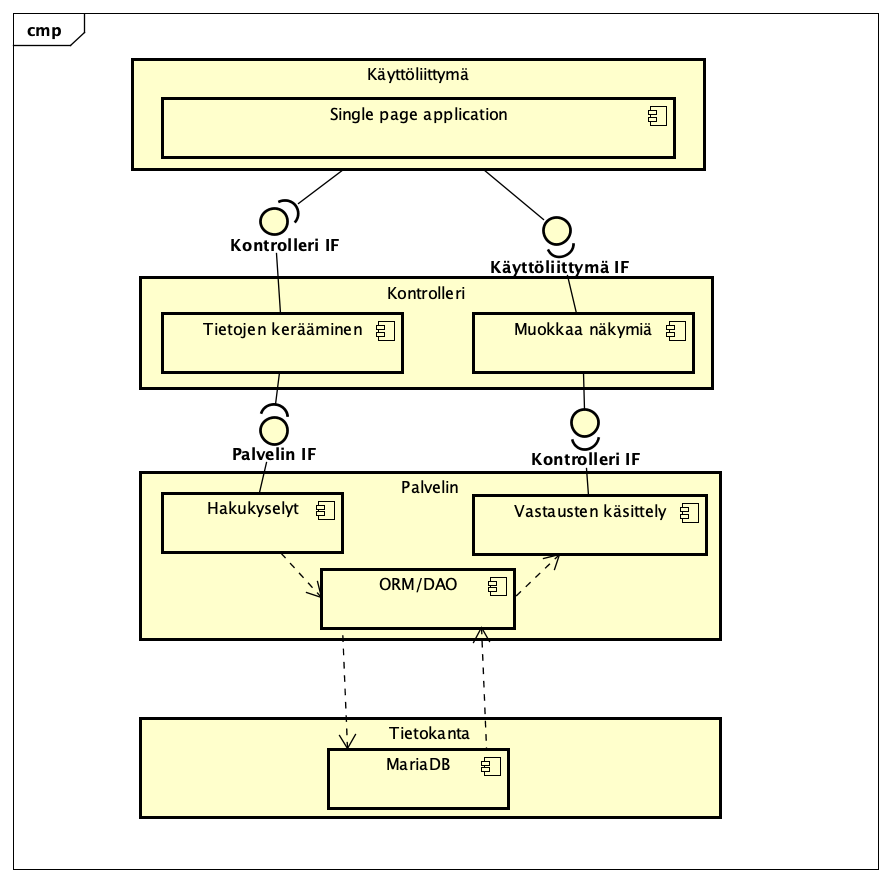
Tarkoitus oli tehdä niin sanottu ”single page application”, eli sovellusta käyttäessä ladataan vain yksi html-sivu. Sovellusta käyttäessä sisältö luotaisiin toimintojen mukaan dynaamisesti JavaScriptin avulla. Sovelluksella olisi kiinteät sivu- ja yläpalkit, joiden toimintoja klikkaamalla osa sivun näkymästä muuttuu tai popup ilmestyy. Sivupalkissa olisi valittavissa kuvakkeet profiilin tarkasteluun, muistiinpanon tekemiseksi, ryhmän tarkasteluun ja hallintaan sekä tapahtuman ja merkinnän luomiseksi. Painikkeita klikkaamalla sivupalkki laajenisi ja siinä näytettäisiin käyttäjän klikkaaman kuvakkeen toiminnot. Käyttäjän kalenteri olisi koko ajan näkyvissä. Yläpalkkiin siirrettäisiin toiminnot, jotka eivät ole samalla tavalla käyttäjän aktiivisuutta vaativia toimintoja kuin sivupalkin toiminnot. Yläpalkkiin sijoitettaisiin kuvake asetuksille, hakukenttä sekä käyttäjän ilmoitusten hallinta. Kuva 1.



Kuva 1 Suunnitelma sovelluksen ulkoasusta

# Arkkitehtuuri

Tuotteen arkkitehtuuri koostuu neljästä kerroksesta ja näiden välisistä rajapinnoista (kuva 2).



Kuva 2 Sovelluksen arkkitehtuurikaavio

## Käyttöliittymä

Käyttöliittymä on toteutettu Reactilla, HTLM:llä ja CSS:llä. Käyttöliittymä vastaa tällä hetkellä toiminnallisuuksiltaan sitä mitä suunniteltiin, ulkoasu on hieman eri. Käyttöliittymän kalenteri on toteutettu Reactin oman kalenteritoiminnon avulla. Lomakkeet, joita käyttöliittymässä on (esimerkiksi asetukset tai merkintöjen luonti), on toteutettu HTML:llä.

## Kontrolleri

Lomakkeissa annetut tiedot validoidaan ja tallennetaan muuttujiin myöhempää käsittelyä varten JavaScript-funktioissa. Muuttujiin tallennetut tiedot lähetetään eteenpäin palvelimelle http-kirjaston http-request -luokan avulla.

## Palvelin

Palvelinkerroksessa NodeJS-kielinen scripti ottaa pyynnöt vastaan. Tähän käytetään Express-moduulia, joka tarjoaa mahdollisuuden vastata eri osoitepyyntöihin niille räätälöidyillä skripteillä. Eri osoitteille on määritelty täten eri skriptit vastaanottamaan httpprequestin lähettämiä post, put, get, delete -pyyntöjä. Skriptit on rakennetu toimimaan REST-tyylisesti, mutta kaikkia RESTin vaatimuksia ei ole täytetty (mm. headerit) sillä datan ei ole tarkoitus olla avointa.

## Tietokanta

Aluksi tietokannasta tehtiin ER-kaavio, jonka avulla hahamoteltiin tauluja ja tietueita, jotka ovat tietokannassa tarpeellisia. Kaaviosta muodostettiin SQL-scripti. Tietokantascriptiä olemme joutuneet muokkaamaan projektin edetessä huomatessamme, että joitain tarpeellisia tauluja tai tietueita puuttuu, tai jokin ratkaisu ei toimi kuten toivoimme. Tietokanta on rakennettu NodeJS:n Sequelize-ORM-moduulilla, joka mahdollistaa tietokannan käsittelyn NodeJS:llä.

# Edistyminen

Kokonaisia käyttäjätarinoita olemme tällä hetkellä saaneet toteutettua kaksi. Syynä tähän on backend-puolen ongelma (express-reittien konfigurointi), jonka johdosta meni pari viikkoa, ettei yhtään käyttäjätarinaa saatu vietyä loppuun asti. Tänä aikana osa tiimistä teki muihin käyttäjätarinoihin käyttöliittymä-, kontrolleri- ja tietokantaosuuksien toteutuksia. Näin ollen seitsemän käyttäjätarinaa on inprogress-tilassa, ja backend-puolen toimintoja vailla valmiita.

Tällä hetkellä meillä on kirjautumissivu, johon syöttämällä tietokantaan lisätyt tunnuksen ja salasanan pääsee kirjautumaan sisään ja siirtymään sovellukseen. Kirjautumissivulta on mahdollista klikata rekisteröitymispainiketta. Tällöin aukeaa rekisteröitymislomake, johon käyttäjä syöttää tietonsa. Kaikkien kenttien sisältö varmistetaan, salasana kryptataan ja tiedot lähetetään tietokantaan. Tämän jälkeen käyttäjä saa ilmoittamaansa sähköpostiin vahvistuslinkin, joka ohjaa sivulle, missä käyttäjän on vielä vahvistettava sähköposti ja salasanansa. Toiminto ei tällä hetkellä ole käyttäjäystävällinen, vaan aika työläs käyttää. Tarkoitus on estää spämmäystä ja turhien tilien luomista. Jos käyttäjä ei vahvista sähköpostiaan tietyn ajan kuluessa, poistetaan profiili tietokannasta (toiminnon toteutus vielä kesken). Toimintoa kehitetään koko ajan helpommaksi ja vaivattomammaksi käyttää.

Käyttäjän kirjauduttua sisään, näkee hän halutessaan profiilinsa tiedot. Käyttäjä voi käyttöliittymässä muokata tietoja, mutta muokattuja tietoja ei vielä tällä hetkellä tallenneta tietokantaan. Sama koskee asetuksia. Käyttäjä näkee sovelluksessa kalenterinsa, ja pystyy selaamaan sitä eteenpäin. Käyttöliittymässä on mahdollisuus luoda tapahtumia ja merkintöjä, ja näiden tiedot tallennetaan kontrollerissa javascript-muuttujiin. Muuttujien tietoja ei vielä viedä tietokantaan.