Mobile Project TTOW0630

Note Keeper

Artur Veselovski

Panu Partanen

Mikko Tiitinen

Projekti

Joulukuu 2017

Tieto- ja viestintätekniikan koulutusohjelma

Sisällysluettelo

[1 Johdanto 2](#_Toc500262874)

[2 Tavoitteet 2](#_Toc500262875)

[2.1 Projektin tavoitteet 2](#_Toc500262876)

[2.2 Tulevat tavoitteet 2](#_Toc500262877)

[3 Käytetyt teknologiat 4](#_Toc500262878)

[3.1 Pääasialliset - MERN 4](#_Toc500262879)

[3.1.1 MongoDB 4](#_Toc500262880)

[3.1.2 Express.js 4](#_Toc500262881)

[3.1.3 React Native 4](#_Toc500262882)

[3.1.4 Node.js 4](#_Toc500262883)

[3.2 Toissijaiset 4](#_Toc500262884)

[3.2.1 Mongoose 4](#_Toc500262885)

[3.2.2 Redux 4](#_Toc500262886)

[3.2.3 JSON Web Tokens (JWT) 4](#_Toc500262887)

[4 Käyttäjätarina 5](#_Toc500262888)

[4.1 Autentikointi 5](#_Toc500262889)

[4.2 Muistutuksen tai tehtävän lisääminen 5](#_Toc500262890)

[4.3 Katselmointi 5](#_Toc500262891)

[4.4 Muokkaaminen 6](#_Toc500262892)

[5 Pääasialliset ominaisuudet 6](#_Toc500262893)

[5.1 Käyttäjä 6](#_Toc500262894)

[5.2 Sovellus 6](#_Toc500262895)

[6 Projektin rakenne 7](#_Toc500262896)

[6.1 Tietokanta 7](#_Toc500262897)

[6.2 UML 8](#_Toc500262898)

[7 Resurssit 8](#_Toc500262899)

[8 Itsearviointi 8](#_Toc500262900)

# Johdanto

Kehitetään cross-platform full-stack mobiilisovellusta React Nativen avulla. Node ympäristöä käyttäen kehitetään REST API, johon sovellus on yhteydessä. REST:iä kehitetään niin, että tulevaisuudessa siihen voi liittää esimerkiksi desktop-sovellus.

Projektin idea tiivistetysti on pitää yllä käyttäjän muistutuksia ja tehtäviä yksinkertaisella ja helpolla sovelluksella.

# Tavoitteet

## Projektin tavoitteet

Projektin ensimmäinen lähtökohta oli tehdä back-end, jota pystyisi käyttämään useissa eri alustoissa, tässä tapauksessa laajuus keskittyi IOS:in ja Androidin välille.

Toinen merkittävä lähtökohta oli oppia tekemään laaja cross-platform mobiilisovellus, johon pystyisi liittämään full-stack developerin vaatimuksia eli tekemään projektin kaikki erilaiset tasot sisältäen tietokannan, back-endin ja front-endin.

Kolmantena tavoitteena oli syventyä javascriptiin ja selvittää mihin kaikkeen se pystyy ja missä sen rajat tulevat vastaan.

## Tulevat tavoitteet

Lisäominaisuutena tulisi yhteiset käyttötilit, missä käyttäjätilillä voisi olla IOS- ja Android-puhelin, mitkä jakaisivat saman tietokannan ja ne voisivat kommunikoida keskenään esimerkiksi notifikaatioiden avulla.

Projektin jälkeen tavoitteena oli myös kehittää desktop-sovellus, jossa käytetään samaa back-endiä kuin mobiiliprojektissa.

# Käytetyt teknologiat

## Pääasialliset - MERN

### MongoDB

MongoDB toimi projektin tietokantana

### Express.js

Express.js on nopea ja helppo tapa tehdä serveri.

### React Native

React Native on front-end framework, joka mahdollistaa tekemään natiivit mobiilisovellukset JavaScriptia käyttäen.

### Node.js

Node.js on selkäranka, jonka ympärillä kaikki pyörivät

## Toissijaiset

### Mongoose

Mongoose kommunikoi databasen kanssa.

### Redux

Redux hallitsee sovelluksen statea.

### JSON Web Tokens (JWT)

Autentikointi

# Käyttäjätarina

## Autentikointi

Käyttäjä lataa sovelluksen ja rekisteröi sähköpostilla ja salasanalla. Nämä tiedot tallennetaan käyttäjätietokantaan encryptattuna ja lähetetään käyttäjälle JSON Web Token (JWT), joka tallentuu käyttäjän laitteeseen eli joka kerta kun käyttäjä kirjautuu sisään, niin kirjautuminen tapahtuu automaattisesti ja käyttäjä viedään päänäkymään. Jos käyttäjä kirjautuu ulos, sen tallennettu JWT poistetaan laitteesta ja käyttäjä viedään kirjautumissivulle. Yllä mainittua JWT käytetään API-requesteissa.

## Muistutuksen tai tehtävän lisääminen

Käyttäjä painaa lisää nappia, joka vie luontinäkymään, jossa käyttäjä täyttää tarvittavat tiedot ja valitsee kategorian, muistutus tai tehtävä ja prioriteetti. Tallennuksen jälkeen käyttäjä viedään päänäkymäään.

## Katselmointi

Päänäkymässä on kolme eri välilehteä; ensimmäisenä tehtävät, jotka on värikoodattu ja prioriteettijärjestyksessä, toisena on tehdyt tehtävät ja viimeisenä välilehtenä muistutukset.

Päänäkymässä käyttäjä voi selata tehtäviä ja merkitä ne tehdyiksi, minkä jälkeen ne siirtyvät tehdyt-välilehteen. Tehdyt-näkymässä käyttäjä voi selata tehdyt tehtävät ja painamalla miinus-näppäintä käyttäjä voi palauttaa niitä tai painamalla pitkään pohjassa poistamalla ne. Poistaessaan tulee varmistusikkuna, jossa kysytään haluaako käyttäjä varmasti poistaa valitun tehtävän.

Muistus-näkymässä käyttäjä voi selata muistutuksia ja poistamalla ne painamalla roskakori-näppäintä.

## Muokkaaminen

Kortin muokkaa-nappia painamalla käyttäjä viedään lisäämisnäkymään, jossa kortit tiedot ovat täytetty valmiiksi, missä käyttäjä pääsee muokkaamaan korttia.

# Pääasialliset ominaisuudet

## Käyttäjä

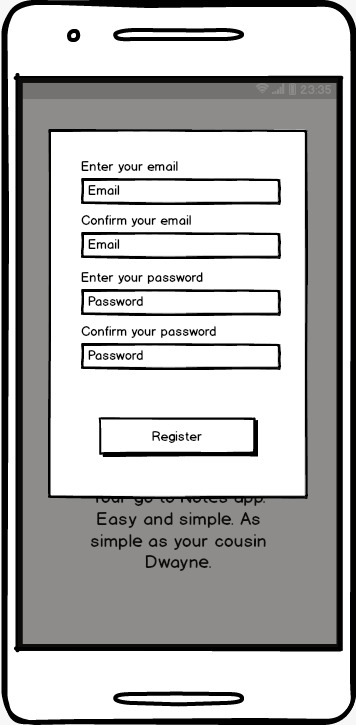
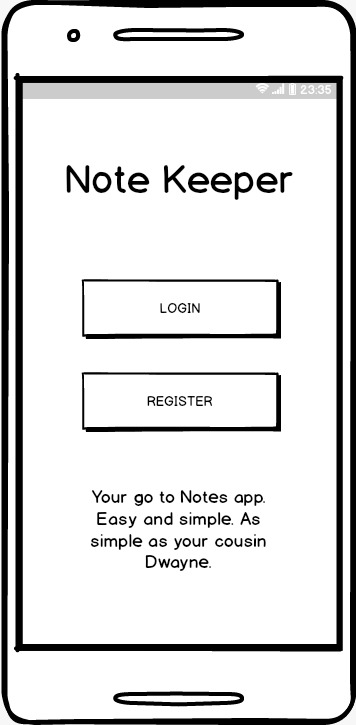
* Rekisteröityminen
* Kirjautuminen

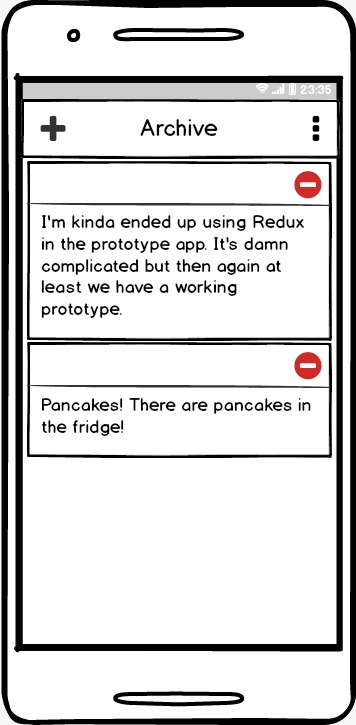
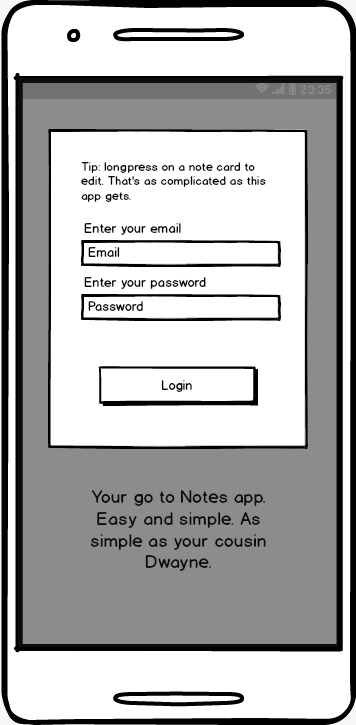
## Sovellus

* Muistutus- tai tehtäväkortin lisääminen
* Muistutus- tai tehtäväkortin poistaminen
* Muistutus- tai tehtäväkortin muokkaaminen
* Tehtäväkorttien hallinta

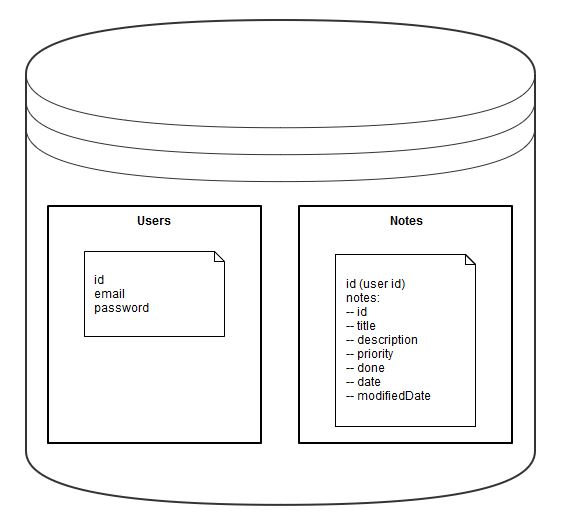
# Projektin rakenne

## Mockupit



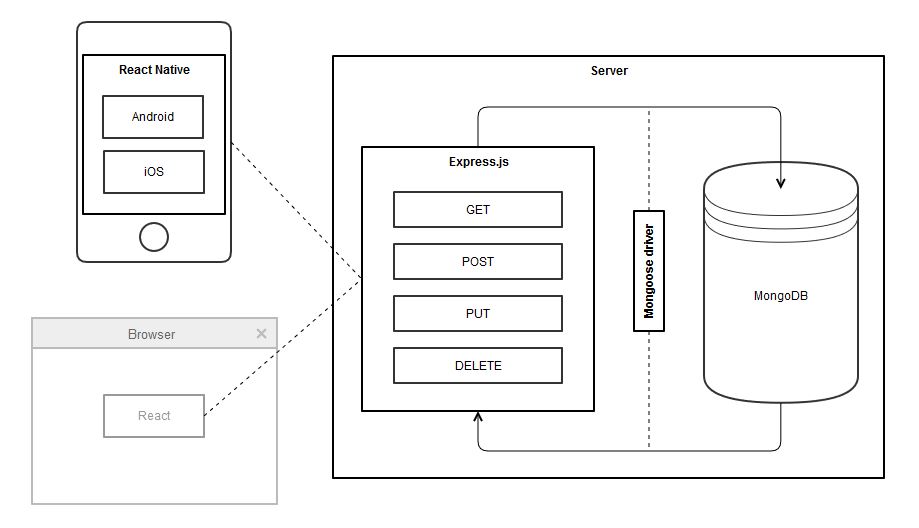


## Tietokanta



Kuva 1 Tietokanta rakenne

## UML



Kuva 2 Projektin rakenne

# Resurssit