# **Context**

De ideeenbord applicatie zorgt voor een manier, waarbij mensen van Schiphol makkelijk ideeen kunnen delen. Het is voor de medewerkers van Schiphol die graag een idee willen delen, maar ook de mogelijkheid hebben om een idee te supporten. Hierbij een visueel beeld van de context:

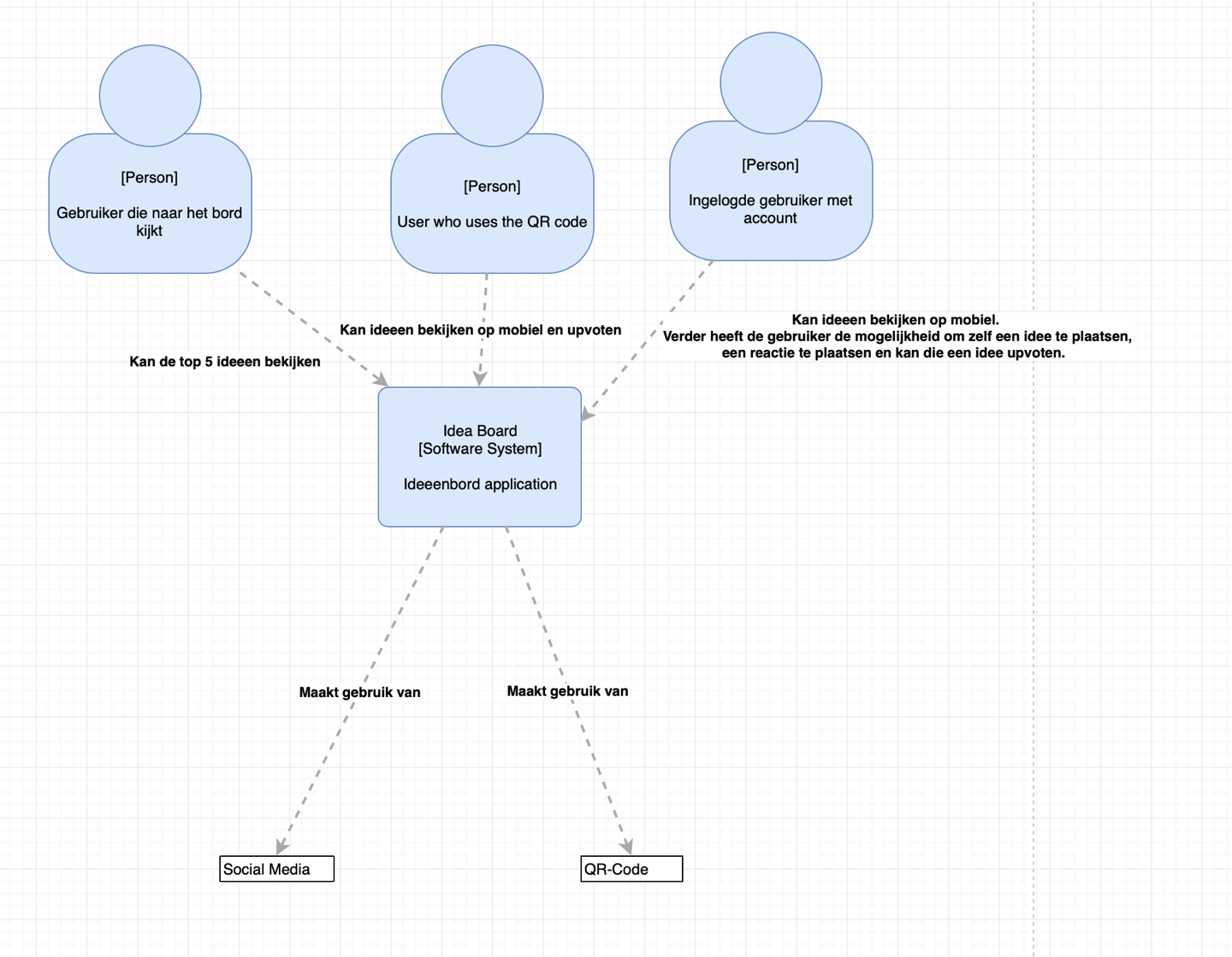


Figure 1 Context diagram

## **De bedoeling van de website is:**

1. Ideeen delen om verandering aan te brengen op de werkvloer
2. Kunnen stemmen zodat je kan bevestigen welke idee je goed vind

## **Gebruikers:**

1. Anoniem: Gebruikers die de bord bekijken
2. Qrcode Authenticated: Gebruikers die via een QR code op de website zijn geauthenticeerd
3. Geregistreerde gebruikers: Gebruikers die geauthenticeerd zijn via registratie

## **Externe Systemen**

Er wordt gebruik gemaakt van 1 externe systeem.

1. Yammer: Mogelijkheid om via Yammer een idee te delen.

# **Functional Overview**

Dit sectie bevat een samenvatting van de functionaliteit van de ideeenbord applicatie.

## **Gebruikers**

Er zijn drie verschillende types van gebruikers

### Anonieme gebruikers

Anonieme gebruikers zijn de gebruikers die naar de top 5 ideeenbord kijken

### QR-code geauthenticeerde gebruikers

Qr-code geauthenticeerde gebruikers zijn de gebruikers die door middel van een QR-code inloggen. Ze hebben de mogelijkheid om ideeen te upvoten maar niet om zelf een idee te plaatsen

### Geregistreerde gebruikers

Gebruikers die door middel van registreren geauthenticeerd zijn. Zij kunnen op ideeen stemmen, maar ook zelf een idee plaatsen. Verder hebben ze de mogelijkheid om op een idee te reageren

## **Content**

De ideeenbord apllicatie heeft als content alleen ideeen, waarbij er reacties op gegeven kunnen worden. Ook kan elke idee geupvote worden. Elke content is geassocieerd aan een gebruiker

## **Ideeenbord beamer**

Er bestaat een ideeenbord waarbij de top 5 ideeen worden getoond van de desbetreffende ideeenbord. Onderaan staat een Qr-code die je kunt scannen via je je mobiel. Verder wordt er bovenaan het nieuwste idee getoond

## **Ideeenbord client**

Er bestaat een ideeenbord waarbij alle ideeen zichtbaar zijn. Je kunt op de ideeenbord client komen door de QR-code te scannen van de ideenbord beamer. Dit is op de mobiel

## **Qr-code**

Onderaan de ideeenbord beamer staat er een qr-code die gescand kan worden, die leidt naar de desbetreffende bord

## **Idee**

De geregistreerde gebruikers kunnen een idee plaatsen met een titel en details

## **Upvotes**

Elke idee kan upvotes hebben. Dit houdt in hoeveel gebruikers het idee goed vinden

## **Reactie**

Op elke idee kan er een reactie geplaatst worden door een geregistreerde gebruiker

# **Quality Attributes**

Dit sectie bevat informatie over de kwaliteit attributes van de applicatie

## **Performance**

De applicatie moet via Websockets functioneren. Het moet een werkvloer aan kunnen, dus een schatting van 50 actuele gebruikers. Verder zou het moeten laden binnen 10 seconde.

## **Websockets**

Het moet een near-realtime applicatie zijn. Dit houdt in wanneer er upvote wordt op het mobiel dat het op de bord ook gewijzigd wordt, indien de idee op het bord staat.

## **Scalability**

De applicatie moet een gemiddelde werkvloer aankunnen dus de volgende data:

* 10000 ideeen??
* 50000 reacties??

## **Security**

De applicatie kan door verschillende gebruikers gebruikt worden. De qr-code geauthenticeerde gebruikers zijn beveiligd door een sessie. De geregistreerde gebruikers hebben een wachtwoord en gebruikersnaam. De wachtwoord is gehasht opgeslagen in de database.

## **Availability**

Hoewel de applicatie ook door anonieme gebruikers bekeken kan worden is dit door gebruikers van de werkvloer. Dus de bereikbaarheid van de app is op de werkvloer

## **Compability**

Het kan gerund worden op de volgende besturingssystemen van de mobiel:

* IOS
* Android

Verder support het de volgende browsers:

* Safari
* Chrome
* Firefox

# **Constraint**

Dit sectie bevat informatie over de restricties van de ontwikkeling van de applicatie

??

# **Principles**

Dit sectie bevat informatie over de principes die geadopteerd zijn voor de ontwikkeling van de applicatie

Geautomatiseerde testing

Unit tests: Via kleine snelle tests kleine methodes en classes testen.

# **Software Architecture**

Dit sectie bevat een overzicht van de software achitectuur van de applicatie

Containers

De volgende container beschrijft de logische containers van de systeem.

A close up of a map

Description automatically generated

Figure 2 Container model

NoSql Database:

MongoDB wordt er gebruikt als NoSQL database waar alle data wordt opgeslagen

Server:

Bij de server wordt er gebruik gemaakt van NODE js, express, nodemon en Websockets.

Web App:

De web applicatie is de beamer client van de applicatie, waarop de top 5 ideeen worden getoond, en de meest recente idee. Voor de rest worden er de upvotes weergegeven na de idee en de Qr-code

Mobiele app:

De mobiele app bestaat uit de volledige lijst van de ideeen. Waarbij je dus ook de mogelijkheid hebt om te reageren op ideeen, en zelf ideeen te plaatsen.

Components- Web App

A screenshot of a cell phone

Description automatically generated

Figure 3 Components - web app

De Websockets communication is er voor near real-time connection tussen de web app en de mobiele app.

Redux … ?

Controller..?

Components – Client app

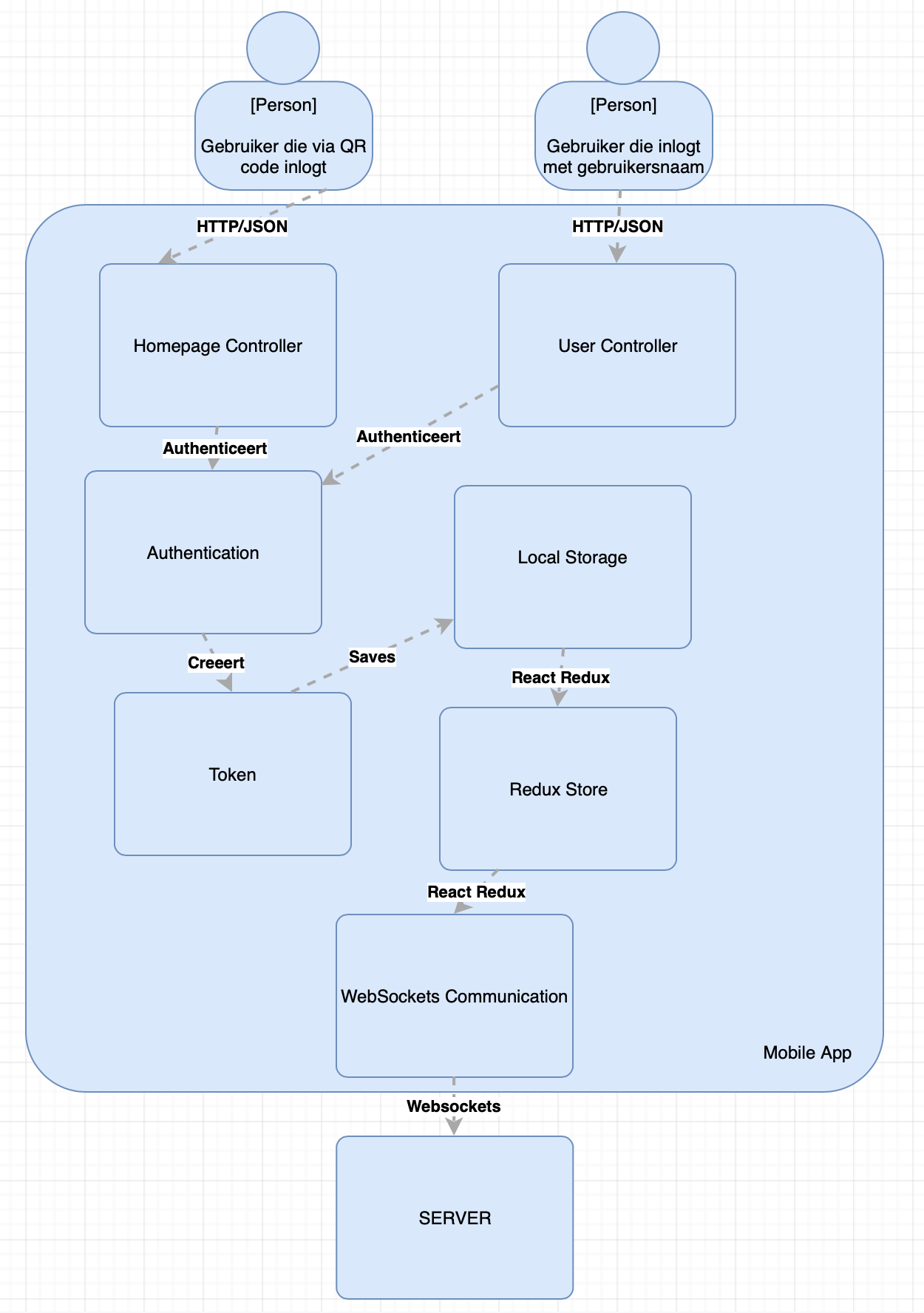


Figure 4 Components - mobiele app

In De homepage en user controller wordt de acties van de homepage en user beheert

Authentication:

Hierin wordt de teken gecreeerd waarmee de authenticatie mee wordt bevestigd. De Token wordt opgeslagen in de local storage.

Components – Server

A picture containing screenshot

Description automatically generated

Figure 5 Components - Server

Content updater:

Dit component pakt de gegevens van Yammer en mongoose. Dit wordt door middel van Websockets doorgegeven aan de mobiele app en web app.

Yammer Connector: Dit component is verantwoordelijk met het connecten van Yammer om ideeen te delen via Yammer

Mongoose wordt er gebruikt om gebruik te maken van de MongoDB database via Wire control op port 27017

# **Infrastructure Architecture**

Dit sectie bevat informatie over infrastructuur architectuur van de applicatie

Live environment

De live environment die er gebruikt wordt is een cloud server van Heroku. Hiermee kan je als developer builden,runnen, en operaten volledig in de cloud

Operating system:?

Server type ?

De gebruikergegevens voor de Heroku dashboard is:

Username:

Gebruikersnaam:

# **Deployment**

Dit sectie bevat de informatie over de overbrugging tussen de software architecture en de infrastructure architecture.

Software

De live environment die wij gebruiken is de Heroku server. Om de applicatie goed te runnen heb je MongoDB 2.2.x nodig en de Heroku CLI.

Bouwen van de applicatie

Het bouwen van de applicatie wordt geschreven en gepushd via Github. Voor de rest wordt het getest via geautomatiseerde testen door middel van Jest. Via Heroku CLI wordt het gepushd naar de cloud server.

Deployment:

* Uitleggen hoe deployment van Heroku werkt…

# **Operation and Support**

Dit sectie bevat informatie over de operationele en support aspecten van de applicatie

Starting Mongodb:

MongoDB is geinstalleerd als een service en wordt gerund als de server restart. Dit kan je controleren door de volgende commando:

1 sudo netstat -tap | grep mongo

2 tail /var/log/mongodb/mongodb.log

Als je mongoDb wilt starten gebruik je de volgende code:

Sudo mongod