# 1 Introduction

The goal of this excercise is to create a WebApplication, which allows a user to connect to a Webserver, that enabnles sychronized editing of a shoppinglist. To do achieve this, we created a team of 3:

## 1.1 Team

Project members:

- David Langheiter is responsible for creating the server with CouchDB
- Matthias Hofbauer will be responsible for setting up the synchronization tool via PouchDB
- Lukas Marek in turn will create the front end with React

# 2 Working enviorment

## 2.1 CouchDB

## 2.2 PouchDB

[3]PouchDB is an open-source JavaScript database that enables applications to store data locally while offline, then synchronize it with CouchDB and compatible servers when the application is back online, keeping the user's data in sync no matter where they next login. A crucial reason, why we didn't choose comparable services like Azure by Microsoft, is that they more often than not are only compatible with one specific database-service. As an example, Azure SQL-Data-Sync, which has a comparable performance to PouchDB, is only compatible with Azure SQL Databases.[1][2] Furthermore PouchDB and CouchDB work best together and have a large supporting library.

## 2.3 React

# 3 Implementation

## 3.1 David Langheiter

### 3.1.1 Design

## 3.2 Matthias Hofbauer

#### 3.2.1 Design

I needed to design an object for the database, which includes all the information that our application needs. At first we only took and ID, the name of an object and if it was purchased or not as a parameter. But as we progressed further with our project we realised, that we still need the amount of our product as a parameter, since just the name of a product is inefficient. Our team discussed which datatype we should use for the amount variable and finally decided on 'text', since different ingredients use different measurement units(liter, meter, kg, etc.). But since 'amount' was now just a text, calling it amount would probably be a little off, since it could just be used for text messages in general. So we changed the name of the variable to commentary, and will implement it as such.

#### 3.2.2 Code

The code itself wasn't to complicated, since I only needed to design basic methods for the database to get/remove/update/synchronize items from our database.

#### 3.2.3 Problems

I wantedanted to include PouchDB in the browser application and Node, with npm install pouchdb Had to restart the system though, since npm couldn't access the project.

### 3.3 Lukas Marek

#### 3.3.1 Design

## Literaturverzeichnis

[1] Microsoft authors. *Azure*. 21.11.2019. Microsoft. URL: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-sync-data.

[2] Symmetrics authors. *Symmetrics*. 21.11.2019. symmetrics. URL: https://www.symmetricds.org/.

[3] Wikipedia Autoren. *Systemtechnik*. 21.11.2019. Wikipedia. URL: https://pouchdb.com/.