Architekturentwurf Wohngemeinschafts-Verwaltung



Abbildung 1: Peggy und Marco Lachmann-Anke / pixabay.com

von Florian Kunzweiler

Matrikel-Nr. 5028828

für den Auftraggeber DHBW-Karlsruhe Sascha Alpers

Version: 1.0 – 14.06.2020

**Versionstabelle:**

|  |  |  |
| --- | --- | --- |
| Versionsnummer | Autor | Änderungsvermerk |
| 1.0 | Florian Kunzweiler | Initiale Fassung |

Inhaltsverzeichnis

[Abbildungsverzeichnis 4](#_Toc43045091)

[Tabellenverzeichnis 4](#_Toc43045092)

[Einführung 5](#_Toc43045093)

[Architekturstrategie 6](#_Toc43045094)

# Abbildungsverzeichnis

[Abbildung 1: Peggy und Marco Lachmann-Anke / pixabay.com 1](#_Toc43043689)

# Tabellenverzeichnis

# Einführung

In größeren Wohngemeinschaften kommt es oft zu Konflikten, weil einer der Mitbewohner Dinge, wie etwa einen Putzdienst, vergisst oder konsequent nicht erledigt. Durch fehlende Sauberkeit sinkt gleichermaßen das Wohlbefinden der restlichen WG-Mitglieder. Zusätzlich möchte man vermeiden die Kosten für den Kauf von Verbrauchsgegenstände wie beispielsweise Spülmittel, Öl, Waschmittel, etc. ungerecht auf einzelne WG-Mitglieder zu verteilen.

Mitglieder einer Wohngemeinschaft sollen die Möglichkeit bekommen Putzdienste und den Kauf von Verbrauchsgegenständen selbst und zuverlässig organisieren zu können, ohne großen Aufwand dafür betreiben zu müssen. So können unnötige Konflikte innerhalb einer Wohngemeinschaft unter Umständen vermieden werden.

Hierzu soll im Rahmen des Projektes eine Webapplikation zur digitalen Verwaltung und Koordination einer Wohngemeinschaft entwickelt werden.

Dieses Architekturdokument wurde für eine erste prototypische Implementierung geschrieben. Es wird daher kein Anspruch auf Vollständigkeit und Rechtskonformität geboten.

# Architekturstrategie

Um möglicherweise in Zukunft (außerhalb dieses Kurses) die Anwendung als mobile Applikation zur Verfügung stellen und erweitern zu können wird der Fokus auf die Erweiterbarkeit gelegt. Somit könnte es sein, dass die Auswahl mancher Technologien im Rahmen dieses Projektes nicht ideal ist.

Des Weiteren werden Back-End und Front-End nicht in einer einzelnen Anwendung umgesetzt, sondern aufgeteilt, um später die Verwaltung und das Beheben von Fehlern zu erleichtern.

Auf Basis dieser Aufteilung erfolgt die Auswahl der Technologien.

# Auswahl des Technologie Stacks

Zu betrachtende Merkmale:

* Erfahrung des Entwicklers
* Verwaltbarkeit
* Deployment (Im Sinne von Komplexität & Aufwand)
* Skalierbarkeit
* Geschwindigkeit
* Implementierung
* Verfügbarkeit (Für mehr als einen User, wenn Webapplikation auf Server deployet ist)

K.O. Kriterien:

* Erfahrung des Entwicklers
* Verfügbarkeit

Bewertungsskala:

|  |  |
| --- | --- |
| 1 Punkt | Sehr schlecht |
| 2 Punkte | Mittel |
| 3 Punkte | Sehr gut |

## Technologieauswahl Datenbank

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | NoSQL | MySQL | GraphQL | DGraph | Dateisystem |
| Erfahrung des Entwicklers | 1 | 3 | 2 | 3 | 3 |
| Verwaltbarkeit | 3 | 3 | 2 | 3 | 3 |
| Deployment | 3 | 3 | 2 | 2 | 1 |
| Skalierbarkeit | 1 | 2 | 3 | 3 | 3 |
| Geschwindigkeit | 1 | 2 | 3 | 3 | 2 |
| Implementierung | 2 | 2 | 2 | 3 | 2 |
| Verfügbarkeit | 3 | 3 | 3 | 3 | 1 |
| Bewertung | 14 | 18 | 17 | 20 | 15 |

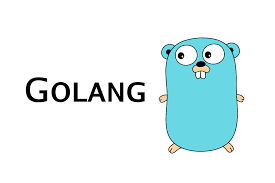
Anhand der Bewertungsmatrix trifft die Entscheidung auf DGraph. Zwar ist dies eine Graph-Datenbank und für diesen Zwecks nicht 100% geeignet, wird aber später wichtig bei der weiteren Entwicklung der Software (außerhalb dieses Projektes).

## Technologieauswahl Back-End

Für das Backend wurde die Programmiersprache Golang gewählt, da die zuvor gewählte Datenbank Technologie mit dieser Sprache entwickelt wurde und somit keine Kompatibilitätsprobleme zu erwarten sind. Zusätzlich dazu, hat der Entwickler sehr gute Erfahrung in dieser Sprache, somit fällt die Phase des Einlernens weg.

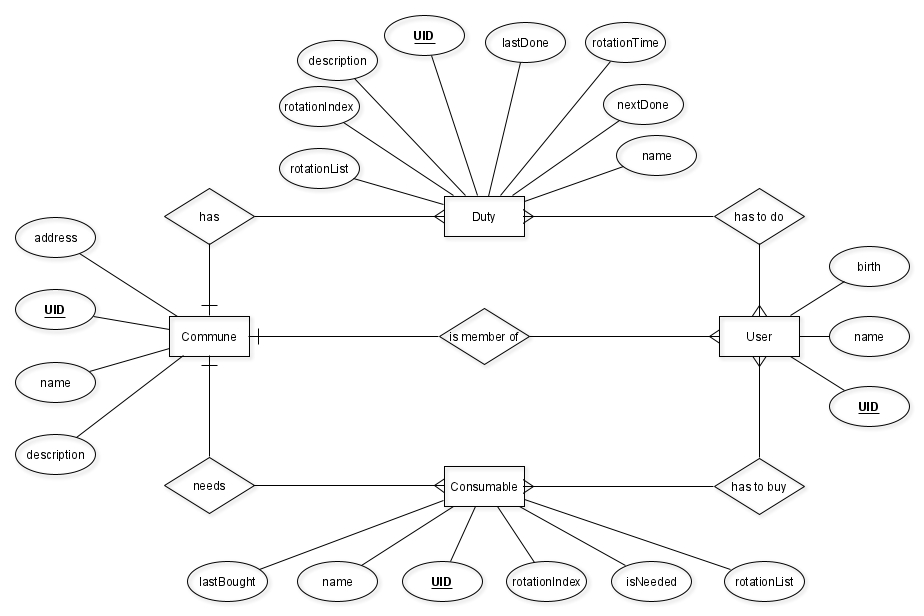
### Technologieauswahl Front-End

Da die Applikation, außerhalb dieses Projektes später für Smartphones zur Verfügung gestellt werden soll, fiel die Wahl auf Flutter. Flutter ist ein UI Toolkit von Google, das für die Entwicklung von Cross-Plattform Anwendungen konzipiert ist und Material-UI für das Design verwendet. Als Sprache benutzt es Dart, eine Objektorientierte Programmiersprache. Da der Entwickler in diesem Bereich auch gute Erfahrungen hat, ist der Aufwand relativ überschaubar.





# Statische Sicht – zu persistierende Daten



Für das WG-Verwaltungssystem sind die Entitäten Commune (Wohngemeinschaft), Consumable (Verbrauchsgegenstand), User (Benutzer / Mitglied) und Duty (Dienste) ausschlaggebend.

Als Primärschlüssel wurde UID (Unique Identifier) gewählt, sodass jede Entität zweifellos identifiziert werden kann.

## Datenbankschema

Hinweis: In DGraph bekommt jeder Datensatz automatisch eine fortlaufende UID (z.B. 0x34) zugewiesen. Listen von Abhängigkeiten zu anderen Entitäten werden über [uid] definiert. Zur besseren Lesbarkeit wurden in der Tabelle die Entitäten dahinter ergänzt. Zeitstempel werden in Form eines Integers gespeichert (Unix-Time), damit das Umwandeln in Date-Objecte zwischen Golang, DGraph und Flutter vereinfacht wird.

|  |  |  |
| --- | --- | --- |
| Entität | Attribut | Datentyp |
| Commune | name | string |
| description | string |
| created | int |
| members | [uid] User |
| address | uid |
| duties | [uid] Duty |
| consumables | [uid] Consumable |
| Address | created | int |
| street | string |
| city | string |
| zip | int |
| Consumable | created | int |
| changed | int |
| name | string |
| isNeeded | bool |
| lastBought | int |
| rotationIndex | int |
| rotationList | [uid] User |
| Duty | created | int |
| changed | int |
| name | string |
| description | string |
| lastDone | int |
| nextDone | int |
| rotationTime | int |
| rotationList | [uid] User |
| rotationIndex | int |
| User | name | string |
| created | int |
| birth | int |
| active | bool |
| token | string |

## Beispieldaten

Hinweis: Aufgrund des Aufwandes wurde auf sinnvolle Timestamps verzichtet.

Commune

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UID | name | description | created | members | address | duties | consumables |
| 0x32 | Die vier coolen Fünf | Hier könnte Ihre Beschreibung stehen. | 1592172000 | 0x67, 0x97, 0x5 | 0x34 | 0x45, 0x49 | 0x15, 0x89 |

User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UID | name | birth | created | active | token |
| 0x5 | Florian Müller | 799459200 | 1591920000 | true | abcd |
| 0x67 | Peter Schneider | 786240000 | 1531340000 | true | zefh |
| 0x97 | Andreas Bäcker | 644414400 | 1531340345 | false | hdsdl |

Hinweis: token kommt von einem zuvor selbst entwickelten Authentifizierungs-Dienst und wird hier nicht weiter erläutert. Die Struktur des Tokens bleibt ein Geheimnis.

Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UID | created | street | city | zip |
| 0x34 | 1592172000 | Kaiserallee 88 | Karlsruhe | 76133 |

Hinweis: Zur Vereinfachung ist Address ein eigener Datentyp in DGraph

Duty

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UID | created | changed | name | description | lastDone | nextDone | rotationTime | rotationList | rotationIndex |
| 0x45 | 1591920000 | 1592172000 | Bad putzen | Putzmittel im Oberschrank | 5894237823 | 249823498 | 8015 | 0x67, 0x97 | 1 |
| 0x49 | 1592172000 | 1592145000 | Küche saugen | Staubsauger im Hausgang | 478932798 | 247843287 | 76847 | 0x5, 0x67, 0x97 | 2 |

* lastDone: Timestamp, wenn der Dienst das letzte mal erledigt wurde
* nextDone: Deadline für die nächste Erledigung des Dienstes
* rotationTime: Dauer in Sekunden, in welchem Abstand der Dienst erledigt werden muss (wöchentlich, täglich, etc…)
* rotationList: Liste der Benutzer, die diesen Dienst erledigen müssen
* rotationIndex: Index für die Liste der Benutzer, wer als nächstes mit dem Dienst an der Reihe ist

Consumables

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UID | created | changed | name | isNeeded | lastBought | rotationIndex | rotationList |
| 0x15 | 2423453 | 136745 | Kloreiniger | false | 4372374973 | 2 | 0x5, 0x67, 0x97 |
| 0x89 | 234324 | 986786 | Staubsaugerbeutel | true | 3249873294 | 1 | 0x67, 0x97 |

* isNeeded: true oder false, je nachdem ob der Verbrauchsgegenstand als benötigt markiert wurde oder nicht
* lastBought: Timestamp, wenn der Gegenstand zuletzt gekauft worden ist
* rotationIndex & rotationList wie bei Duty

# Statische Sicht – Objekttypen zur Laufzeit