SAMAM

Projektdokumentation

Von Alexander Böttger, Mathis Ludwig, Arthur Mehlmann, Sven Peters und Markus Rumpel

Inhalt

[Inhalt 1](#_Toc73452192)

[Versionshistorie 2](#_Toc73452193)

[Einleitung 2](#_Toc73452194)

[Motivation 2](#_Toc73452195)

[Product Vision 3](#_Toc73452196)

[Vision Board 3](#_Toc73452197)

[Zielgruppe 3](#_Toc73452198)

[Bedürfnisse 3](#_Toc73452199)

[Produkt 3](#_Toc73452200)

[Nutzen 3](#_Toc73452201)

[Vision Statement 3](#_Toc73452202)

[Personas 4](#_Toc73452203)

[Frank Meier (Admin) 4](#_Toc73452204)

[Max Freitag 4](#_Toc73452205)

[Lisa Will 4](#_Toc73452206)

[Userstories 5](#_Toc73452207)

[High Prio 5](#_Toc73452208)

[Low Prio 5](#_Toc73452209)

[Prototyp 7](#_Toc73452210)

[Mobiler Prototyp 8](#_Toc73452211)

[Desktop Prototyp 9](#_Toc73452212)

[Technologien 12](#_Toc73452213)

[Verwendete Technologien und deren Lizenzen 12](#_Toc73452214)

[Beschreibung wichtiger Technologien 12](#_Toc73452215)

[Technologie-Stack 12](#_Toc73452216)

[JavaScript-Erweiterung 13](#_Toc73452217)

[CSS-Framework 13](#_Toc73452218)

[CSS-Erweiterung 13](#_Toc73452219)

[Versionsverwaltung 13](#_Toc73452220)

[Softwarearchitektur 14](#_Toc73452221)

[Verteilungsdiagramm 14](#_Toc73452222)

[Komponentendiagramme 15](#_Toc73452223)

[Server 15](#_Toc73452224)

[Client 15](#_Toc73452225)

[Datenbankstruktur 16](#_Toc73452226)

# Versionshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Features | Datum | Umgesetzt durch |
| 0.0.0 | - Initiales Projekt | 23.03.2021 | Markus Rumpel, Alexander Böttger |
| 0.0.1 | - Projekt neu aufgesetzt mit Typescript als Programmiersprache | 10.04.2021 | Mathis Ludwig |
| 0.0.2 | - Ordentlicher Typescript Code | 11.04.2021 | Mathis Ludwig |
| 0.0.3 | - Initialisierung der DB bei Start des Servers­ - Automatische Erzeugung von Testdaten in der DB. | 13.04.2021 | Markus Rumpel |
| 0.0.4 | - Einbindung von Bulma.io als UI-Framework | 16.04.2021 | Mathis Ludwig |
| 0.0.5 | - Entfernen der NodeModules aus dem Git-Projekt | 25.04.2021 | Mathis Ludwig |
| 0.0.6 | - Erstellung von Kleinanzeigen ermöglichen | 04.05.2021 | Mathis Ludwig |
| 0.0.7 | - Landingpage - Umleitung zu Keycloak | 06.05.2021 | Markus Rumpel, Alexander Böttger |
| 0.0.8 | - Header - Listenanzeige der Kleinanzeigen | 07.05.2021 | Mathis Ludwig |
| 0.0.9 | - Detailansicht der Kleinanzeigen - Kleinanzeigen löschen und bearbeiten | 25.05.2021 | Mathis Ludwig |
| 0.0.10 | - Grundfunktionalität des Chats | 08.06.2021 | Sven Peters |
| 0.0.11 | - Laden einzelner Seiten von Kleinanzeigen statt aller auf einmal | 08.06.2021 | Mathis Ludwig |
| 0.0.12 | - Login mit Keycloak | 08.06.2021 | Markus Rumpel |
| 0.0.13 | - Abfrage auf User | 08.06.2021 | Markus Rumpel |
| 0.0.14 | - Fixes am Chat | 11.06.2021 | Sven Peters |
| 0.0.15 | - Erweiterungen der Kleinanzeigen um Preis und Ort - Sortierung nach Preis bei der Kleinanzeigen-Übersicht hinzugefügt - Anpassungen am Styling | 14.06.2021 | Mathis Ludwig |
| 0.0.16 | - Hochladen und anzeigen von einem Bild bei Kleinanzeigen | 16.06.2021 | Mathis Ludwig |
| 0.0.17 | - Suchfunktion bei Liste der Kleinanzeigen hinzugefügt | 16.06.2021 | Alexander Böttger, Mathis Ludwig |

# Einleitung

Dieses Dokument begleitet den Entwicklungsprozess der Anwendung „SAMAM“. Diese dient als Portal für Kleinanzeigen. Das Ziel ist es, die Zeitungsanzeigen in das digitale Zeitalter zu befördern. Die Anwendung soll schnell und einfach nutzbar sein und den Nutzern den nötigen Freiraum bieten, das zu suchen und zu finden, das für sie relevant ist.

# Motivation

Durch die Anwendung sollen die Nutzer sich mit anderen, der Hochschule verbundenen Personen verbinden können. Es soll eine regionale Plattform zum Austausch werden.

# Product Vision

Die Product Vison legt die Zielrichtung der Anwendung dar. Anhand dieser Vision wird die Anwendung entwickelt. Somit ist von Anfang an gegeben, wohin wir im Laufe des Projektes kommen wollen. Mit einer gemeinsamen Vision können alle Teammitglieder an einem Strang ziehen.

## Vision Board

|  |  |  |  |
| --- | --- | --- | --- |
| Zielgruppe Alle Angehörigen einer Hochschule:   * Dozenten * Lehrbeauftragte * Studenten * Angestellte der Hochschule * Gäste der Hochschule | Bedürfnisse Hochschulinterne Plattform zum Austausch | Produkt Eine Plattform/Web-App zum Erstellen, Anzeigen und Verwalten von Kleinanzeigen, Freizeitaktivitäten, Mitfahrgelegenheiten für eine geschlossene Benutzergruppe | Nutzen Ökologisch, da Produkte nicht neu gekauft werden müssen |
|  | Organisation von Mitfahrgelegenheiten | Ermöglicht die Kommunikation zwischen allen Angehörigen einer Hochschule | Geld sparend |
|  | Gemeinschaftliches planen von Freizeitaktivitäten | Die Verwendung findet plattformunabhängig im Browser statt | Umwelt schonend da, weniger Autos durch Fahrgemeinschaften |
|  | Knüpfen von sozialen Kontakten |  | Bessere Vernetzung der Hochschul-angehörigen untereinander |

## Vision Statement

Wir möchten eine plattformunabhängige Web-App zum Erstellen, Anzeigen und Verwalten von Kleinanzeigen, Freizeitaktivitäten und Mitfahrgelegenheiten entwickeln. Die App soll für alle Angehörigen der Hochschule zugänglich sein und erleichtert den Informationsaustausch aller Beteiligten. Außerdem kann ein internes Netzwerk geschaffen werden, dass sowohl umweltschonende als auch ökologische Aspekte berücksichtigt. Des Weiteren können Kosten gespart werden da, die Benzinkosten in den Fahrgemeinschaften geteilt werden können.

## Personas

Folgenden Personas beschreiben fiktive Personen. Diese sind Nutzer sollen Potentielle Nutzer der Anwendung beschreiben.

### Frank Meier (Admin)

Frank Meier 43, Professor an der Hochschule und Administrator der Anwendung. Er kümmert sich darum, dass es keine Anwendungen gibt, die nicht auf die Plattform gehören. Hierfür benötigt er besondere Rechte. Außerdem verkauft er gerne alte Bücher an andere Professoren und Studierende.

Abbildung 1 Von Olaf Kosinsky - Eigenes Werk, CC BY-SA 3.0 de, https://commons.wikimedia.org/w/index.php?curid=63876156

### Max Freitag

Max Freitag ist 23 und studiert im 6. Semester Technische Informatik. Er lernt gerne neue Leute kennen und möchte die Anwendung daher nutzen, um mit anderen Studierenden Aktionen zu starten. Da er immer etwas knapp bei Kasse ist und ihm viel an der Umwelt liegt, kauf die oft gebrauchten Klamotten oder Bücher. Auch hierfür will er die Anwendung in Zukunft nutzen.

Abbildung 2 Von Olaf Kosinsky - Eigenes Werk, CC BY-SA 3.0 de, https://commons.wikimedia.org/w/index.php?curid=63876572

### Lisa Will

Lisa Will ist 18 Jahre alt und kommt gerade von der Schule. Sie beginnt demnächst ihr Studium der Sozialen Arbeit an der Hochschule Esslingen. Die gebürtige Münchenerin möchte regelmäßig in die Heimat fahren, um ihre Familie zu sehen. Da ihr die Bahntickets zu teuer sind, sucht sie über die Anwendung eine Mitfahrgelegenheit. Somit will sie Geld sparen und direkt neue Leute kennenlernen.

Abbildung 3 Von Gage Skidmore, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=61334721

# Userstories

## High Prio

Als Ersteller einer Anzeige möchte ich ein „suche“ Angebot einstellen können, um Angebote zu erhalten.

Als Ersteller einer Anzeige möchte ich ein „biete“ Angebot einstellen können, um Gegenstände oder Tätigkeiten anzubieten.

Als Ersteller einer Anzeige möchte ich ein Angebot ändern können, um Fehler zu beheben.

Als Ersteller einer Anzeige möchte ich ein Angebot löschen können, wenn es nicht mehr gültig ist.

Als Nutzer der Plattform möchte ich nach Angeboten suchen können, die mich interessieren, um mir relevante Kleinanzeigen anzeigen zu lassen.

Als Nutzer der Plattform möchte ich mit Angebotserstellern kommunizieren können, um Unklarheiten zu klären oder zu verhandeln.

Als Administrator möchte ich ein Angebot löschen, das nicht auf die Seite gehört.

Als Nutzer der Plattform möchte ich Angebote markieren können, um sie später wiederzufinden.

Als Nutzer der Plattform möchte ich nach Namen der Anzeigen, Tags, Einstelldatum und Preis suchen können, um eine bessere Angebotsauswahl zu finden.

Als Ersteller einer Anzeige möchte ich Bilder zu meinen Angeboten einstellen können, um die Angebote besser präsentieren zu können.

Als Hochschulangehöriger möchte ich mich nicht registrieren müssen, um die Anwendung zu nutzen.

Als Hochschulangehöriger möchte ich mich über die Hochschulseite einloggen können, um den Vorteil von SSO nutzen zu können.

Als Ersteller einer Anzeige möchte ich Tags zu meinen Anzeigen hinzufügen, um diese leichter auffindbar zu machen.

## Low Prio

Als Nutzer der Plattform möchte ich Leuten folgen können, um Personen mit interessanten Angeboten wiederfinden zu können.

Als Nutzer der Plattform möchte ich während des gesamten Prozesses anonym bleiben, um meine Privatsphäre zu schützen.

Als Nutzer der Plattform möchte ich nach Namen der Anzeige, Einstelldatum oder Preis sortieren können, um für mich relevante Angebote zu finden.

Als Nutzer der Plattform möchte ich ein Profilbild einstellen können, um in Chats eine persönlichere Ebene zu erreichen.

Als Nutzer der Plattform möchte ich Gruppenchats machen können, um mich mit mehreren Nutzern gleichzeitig kurzschließen zu können.

Als Nutzer der Plattform möchte ich angezeigt bekommen, wenn ich eine Nachricht erhalte, um keine Nachrichten zu verpassen.

Als Nutzer der Plattform möchte ich, dass alte Angebote automatisch gelöscht werden, um nicht von zu vielen, teils nicht mehr relevanten Angeboten, überfordert zu sein.

Als Nutzer der Plattform möchte ich automatisch relevante Vorschläge erhalten, um die perfekten Angebote für mich zu finden.

Als Nutzer der Plattform möchte ich Ortsangaben auf einer Karte sehen, um meinen abzuschätzen, wie weit die Entfernung von meinem Standort ist.

Als Ersteller einer Anzeige möchte ich Bewertungen erhalten, um meine Seriosität zu steigern.

Als Ersteller einer Anzeige möchte ich automatisch Tags zu meinen Angeboten generiert haben, um nicht selbst Tags eintragen zu müssen.

# Prototyp

Zum schnellen Testen haben wir einen UI-Prototyp entwickelt. Diesen haben wir mir der Anwendung Adobe XD gestaltet.

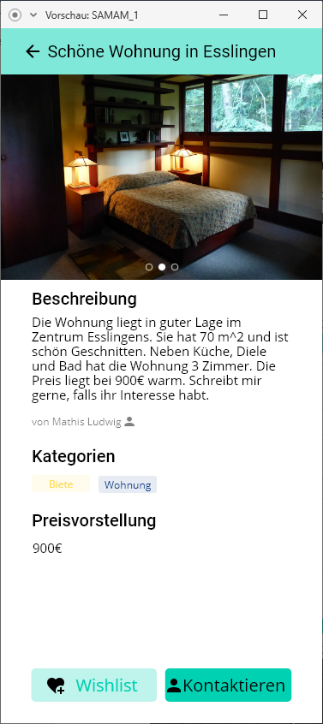
Bei der Gestaltung haben wir uns bereits an dem Framework „Bulma.io“ orientiert, da wir dieses zur Implementierung nutzen wollen.

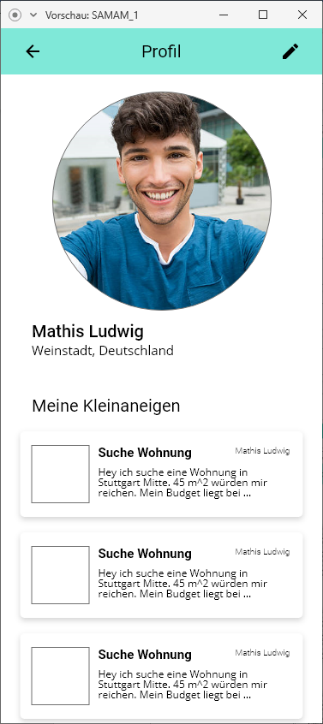
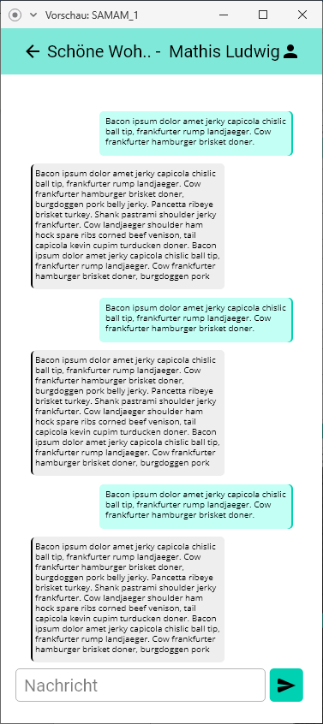
Begonnen haben wir mit den Prototypen für einen mobile Ansicht. Danach haben wir die Elemente auf eine Desktop-Ansicht angepasst.

Die Navigation soll für jeden Nutzer einfach zu verstehen sein. Deshalb möchten wir bekannte Nutzungskonzepte, wie ein Hamburgermenü, verwenden. Was Usability an geht, gilt es, nicht zu versuchen das Rad neu zu erfinden. Für die meisten Nutzer sind bekannte Strukturen am leichtesten zu Nutzen und sorgen für die beste User Experience.

Außerdem nutzen wir bekannte Icons im Material Design. Diese unterstützen den Nutzer bei der Navigation.

## Mobiler Prototyp

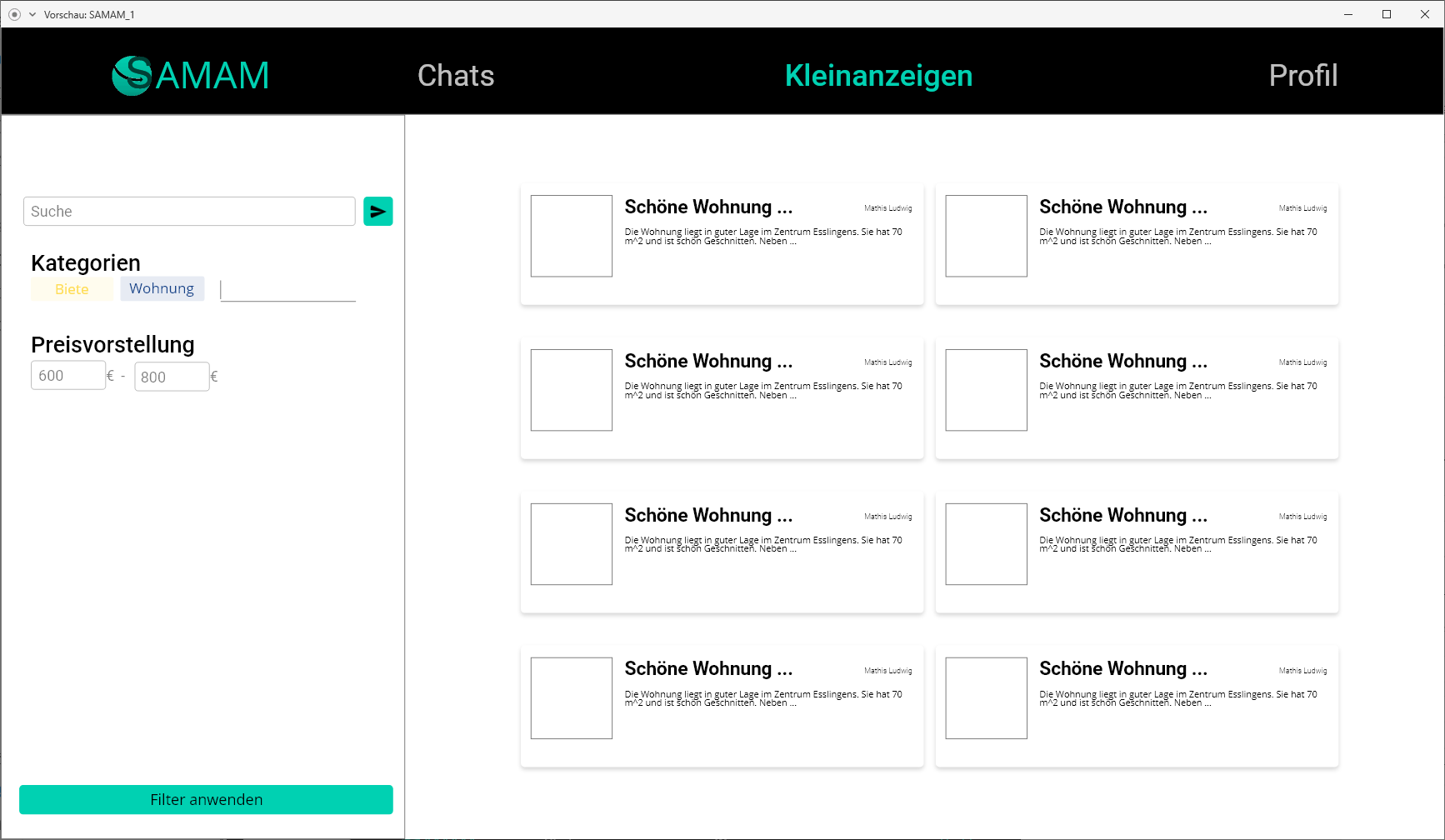


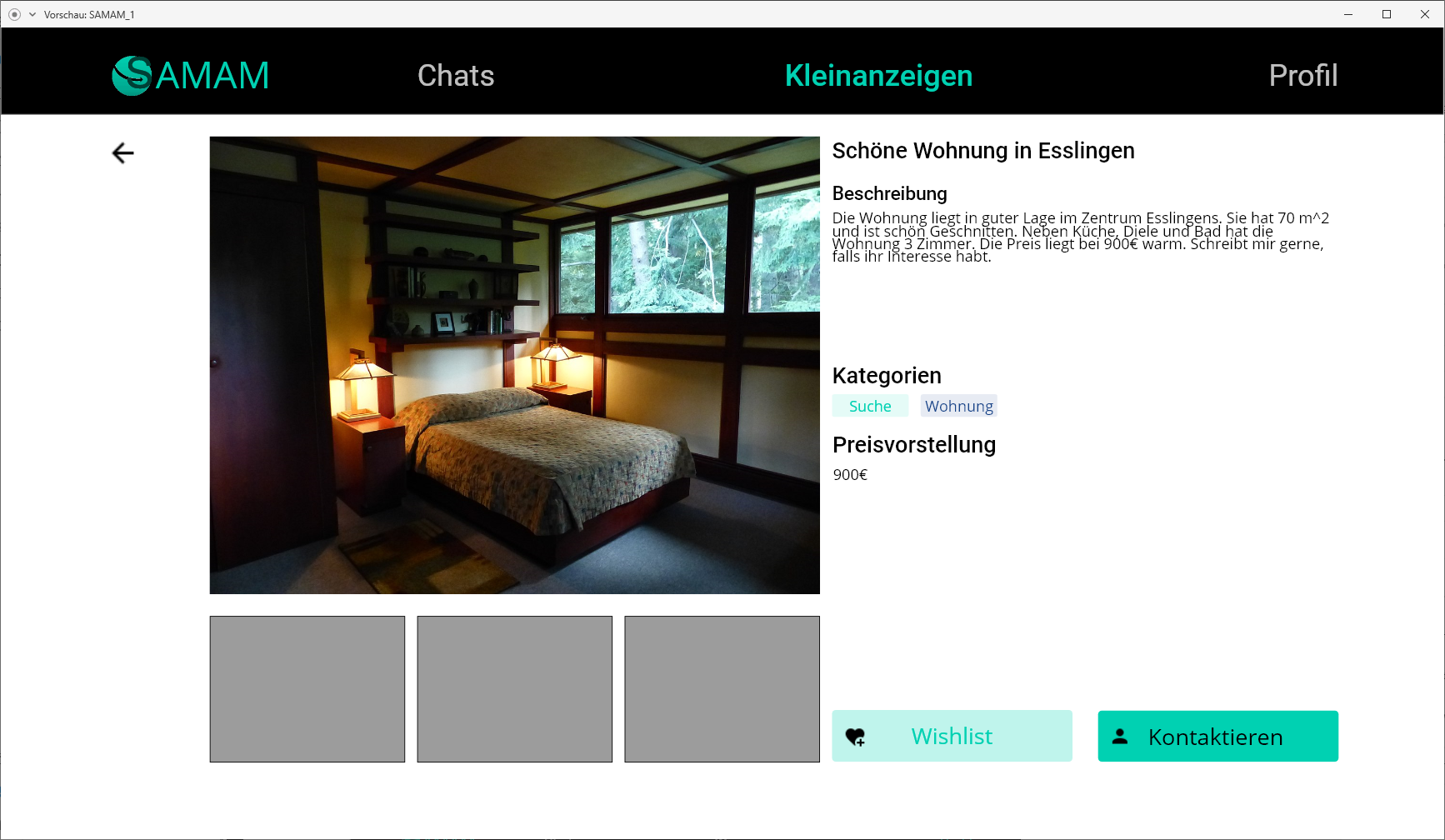


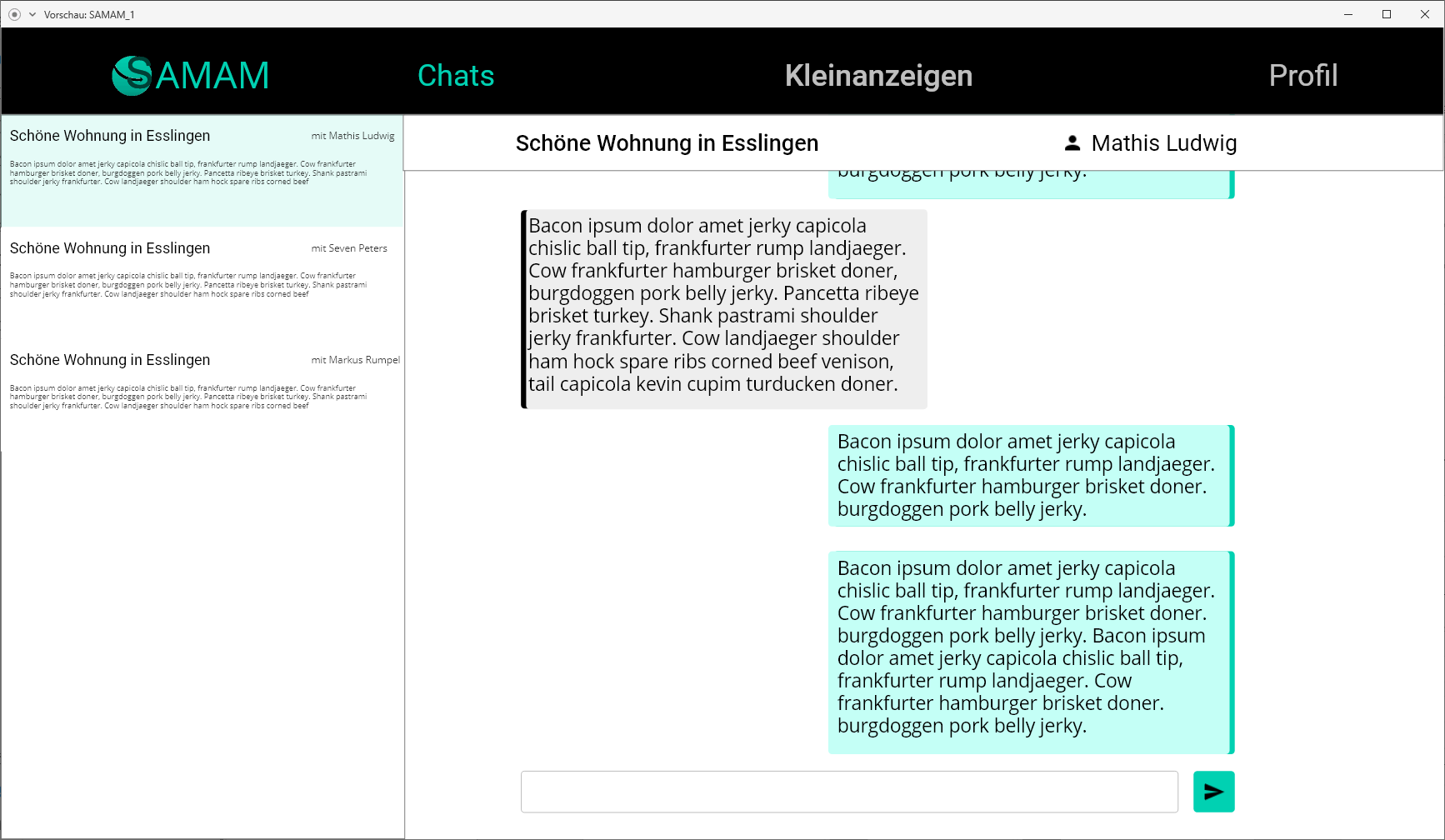
## Desktop Prototyp

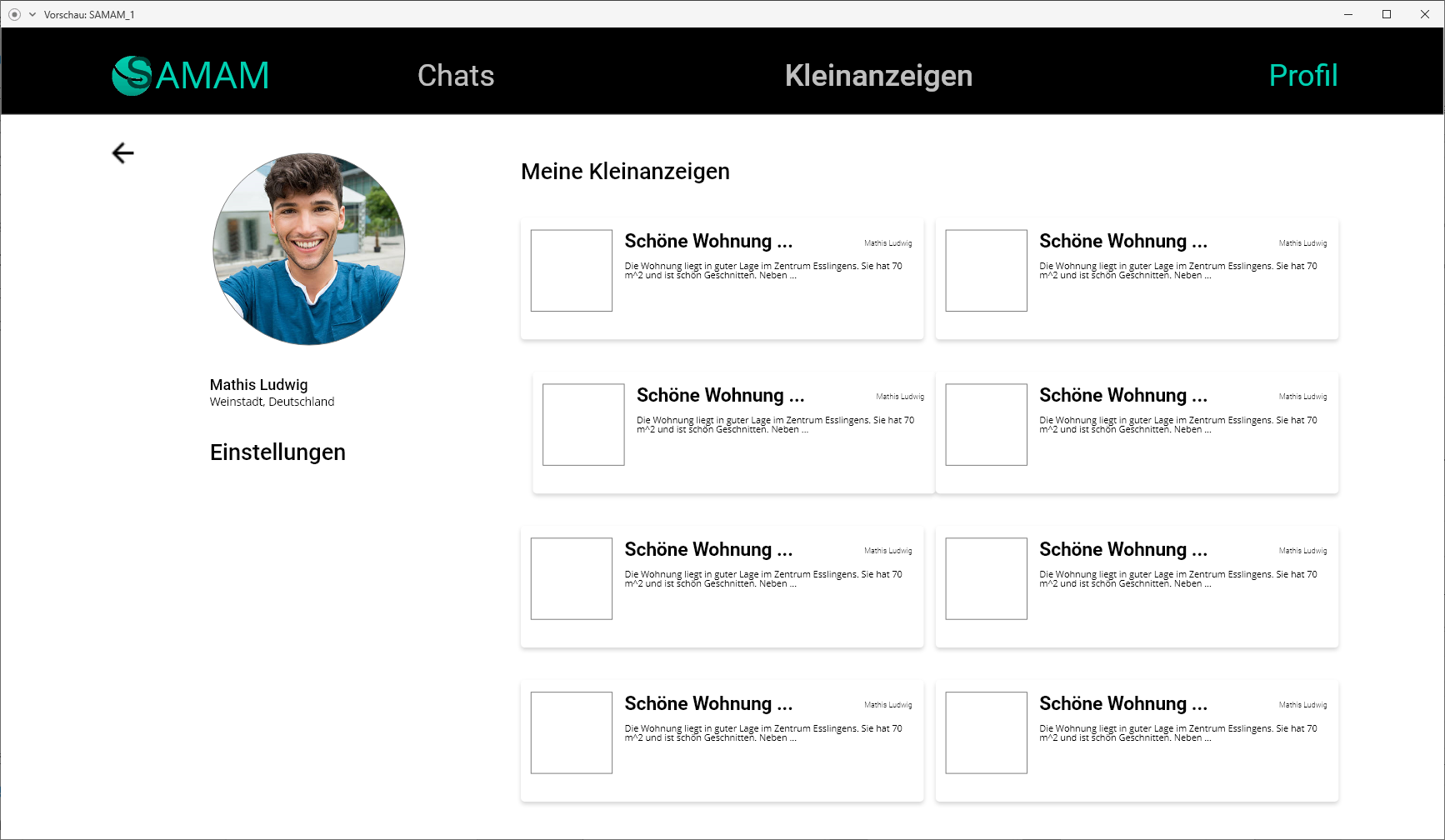
Der Desktop Prototyp ist sehr ähnlich aufgebaut wie der, für die mobile Nutzung. Somit können wir Komponenten einfach übertragen, anstatt neue Komponenten für verschiedene Displaygrößen zu schreiben. Auch die Nutzer erfreuen sich so an einer einheitlichen Usability, wenn sie das Gerät wechseln.  
Es wurden lediglich mehr Komponenten auf einzelne Screens gelegt, da der Desktop eben wesentlich mehr Platz bietet.











# Technologien

## Verwendete Technologien und deren Lizenzen

|  |  |  |
| --- | --- | --- |
| Technologie | Lizenz | Quelle |
| Mevn-cli | MIT | https://github.com/MadlabsInc/mevn-cli |
| Mogoose | MIT | https://github.com/Automattic/mongoose |
| Express | MIT | https://github.com/expressjs/express |
| Typescript | Apache-2.0 | https://github.com/Microsoft/TypeScript |
| Socket.io | MIT | https://github.com/socketio/socket.io |
| Bulma | MIT | https://github.com/jgthms/bulma |
| Vue | MIT | https://github.com/vuejs/vue |
| Node.js | Custom License https://github.com/nodejs/node/blob/master/LICENSE | https://github.com/nodejs/node |
| multer | MIT | https://github.com/expressjs/multer |
| GridFS | No License File | https://github.com/siddMahen/GridFS |
| Cors | MIT | https://github.com/expressjs/cors |
| Axios | MIT | https://github.com/axios/axios |
| Eppsilon/keykloak | Apache-2.0 | https://github.com/eppsilon/keycloak |

## Beschreibung wichtiger Technologien

### Technologie-Stack

Wir haben uns für dieses Projekt für den sogenannten „**MEVN**“-Stack entschieden. Dieser besteht aus MongoDB, Express, Vue und NodeJs.

**MongoDB** ist eine gut skalierbare, beliebte NoSQL Datenbank. Da wir alle bisher nur mit Datenbanken gearbeitet haben, die auf SQL basieren, können wir somit im Rahmen des Projektes neue Dinge lernen und neue Erfahrungen sammeln.

Auch **Express** ist aktuell ein sehr beliebtes Framework. Wir nutzen dieses, um unsere API zu erstellen.

Als Client-seitiges JavaScript-Framework nutzen wir „**Vue**“. Vue ist ein einsteigerfreundliches Framework, das jedoch Kenntnisse in HTML, CSS und JavaScript voraussetzt. Da wir diese alle innehaben, ist das jedoch kein Hindernis. Die Dokumentation von Vue ist gut aufgebaut, was für ein Projekt, bei dem man viele neue Technologien kennenlernt, sehr hilfreich ist.

**NodeJs** ist als Laufzeitumgebung durch den Kunden vorgegeben.

### JavaScript-Erweiterung

Bei einem Projekt dieser Größe, an dem mehrere Entwickler arbeiten, ist JavaScript durch das Fehlen von Datentypen nicht optimal geeignet. Deshalb entschieden wir uns, unseren Code in **Typescript** zu schreiben.

### CSS-Framework

Bei der Wahl des CSS-Frameworks haben wir uns bewusst gegen das mit Abstand am weitesten verbreiteten Framework „Bootstrap“ entschieden. Das liegt daran, dass durch die weite Verbreitung von Bootstrap viele Webseiten sehr ähnlich aussehen. Sie haben den typischen Bootstrap-Look. Wir haben uns stattdessen für „**Bulma**“ entschieden. Bulma ist ein responsive, mobile first CSS-Framework, das sehr einfach zu nutzen ist. Zudem ist es modular, was dafür sorgt, dass man nur das importieren kann, das man auch nutzt. Wir erhoffen uns hierdurch selbst nur wenig CSS schreiben zu müssen.

### CSS-Erweiterung

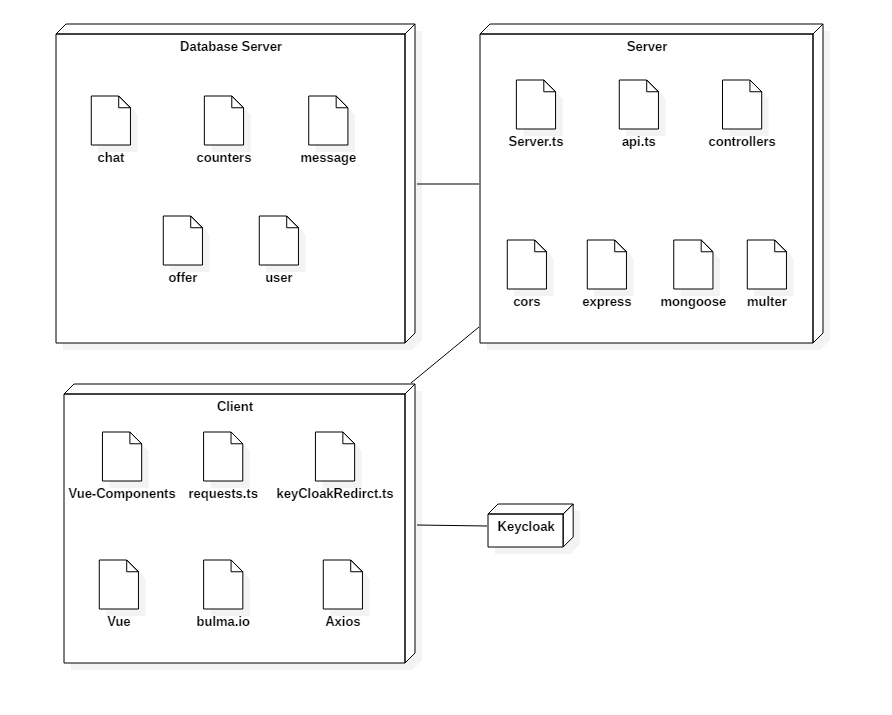
Wenn wir jedoch die Styles anpassen wollen, schreiben wir kein CSS, sondern nutzen „**Sass**“. Sass ist ein CSS-Präprozessor, der im Vergleich zu einem CSS eine bessere Nutzung verspricht. Zudem ist Bulma in Sass geschrieben.

### Versionsverwaltung

Als Software zur Versionskontrolle wurde „**git**“ vorgegeben. Wir einigten uns darauf, „Git Extentions“ als UI-Anwendung zu verwenden. Somit können wir uns bestmöglich gegenseitig unterstützen.

# Softwarearchitektur

## Verteilungsdiagramm



Das Verteilungsdiagramm beschreibt, welche Komponenten und Frameworks in welchem Teil des Systems zum Einsatz kommen. Hierfür haben wir die Datenbank, den Server und den Client. Auf dem Server sind die verschiedenen Collections, die in SQL-Datenbanken Tabellen wären, dargestellt. Der Server beinhaltet die Controller, über die der Zugriff auf die Datenbank geregelt wird. Clientseitig laufen die Vue Komponenten. Außerdem wird Keycloak zur Identifizierung der Nutzer verwendet.

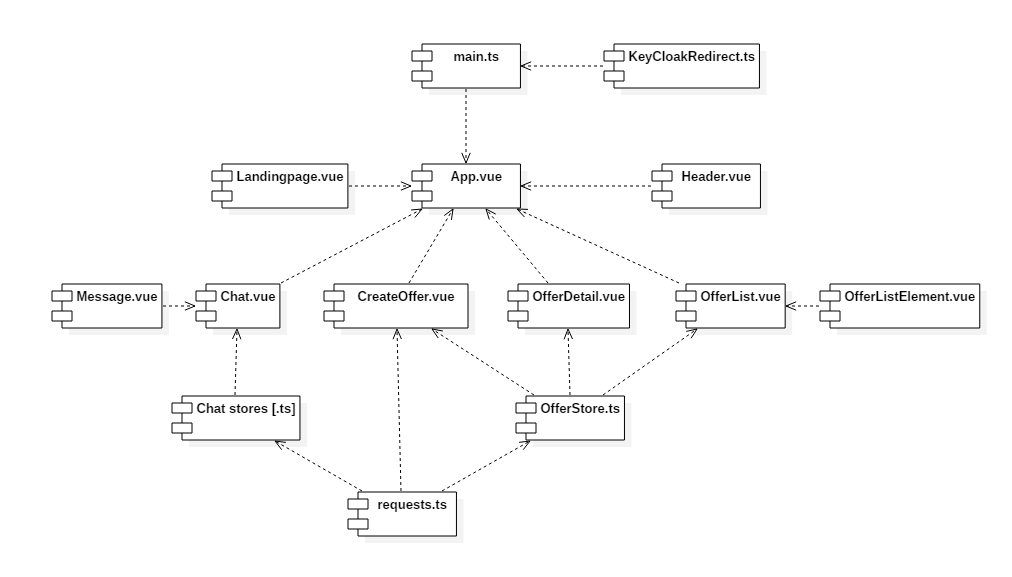
## Komponentendiagramme

### Server



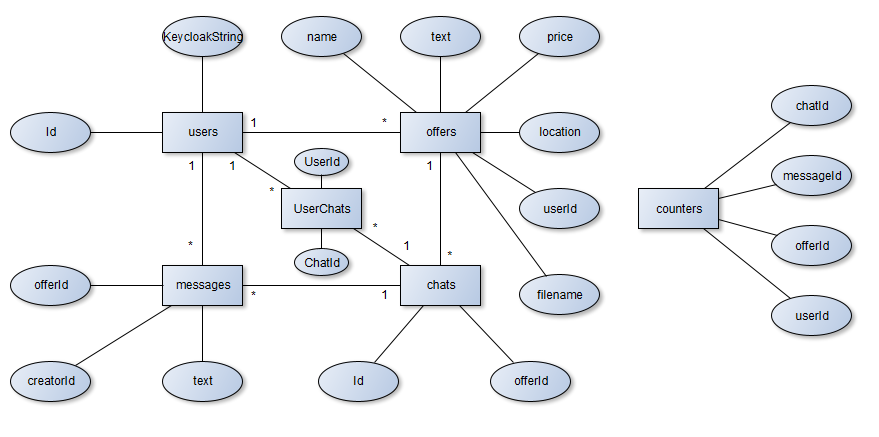
Die Basis der Serverstruktur ist die MongoDB. Auf der Datenbank werden alle persistenten Daten gesichert. Auf diese wird über die verschiedenen Controller zugegriffen.   
Diese Controller Existieren für jedes Modell der Datenbank. Für die grundlegenden Funktionen greifen alle Controller auf einen BaseContoller zu, soweit es möglich ist. Somit werden Dopplungen im Code verhindert. Die Requests werden von einem Express-Router empfangen.

### Client



Das Diagramm zeigt die App.vue, um die die Anwendung gebaut wurde. Gestartet wird sie über die main.ts, welche wiederum Keycloak initialisiert. In der App.Vue werden wiederum verschiedene Vue-Componenten angezeigt. Die Kommunikation zum Server wird in die requests.ts ausgelagert.

## Datenbankstruktur

  
Das Datenbankdiagramm zeigt die einzelnen Beziehungen zwischen den verschiedenen Entitäten. Die Counters Klasse dient nur zur Erzeugung der Ids uns hat daher keinerlei Verbindung zu den anderen Entitäten.