

Entwurfsbeschreibung

Projekt Going-Home by Pws17

Ein Service für die Wohnungssuche

Gliederung

1. Allgemeines
2. Produktübersicht
3. Grundsätzliche Struktur- und Entwurfsprinzipien
4. Struktur- und Entwurfsprinzipien einzelner Pakete
 - 4.1 Client
 - 4.2 Server
5. Datenmodell
 - 5.1 Datenfluss des Clients
 - 5.2 Datenfluss des Servers
6. Glossar

1. Allgemeines

Dies ist die vierte Version der Entwurfsbeschreibung des Projektes „Going Home“ der Softwaretechnikpraktikumsgruppe Pws17.

2. Produktübersicht

Das fertige Produkt stellt dem Benutzer einen Service zur Verfügung über welche er Wohnobjekte (Häuser, Wohnungen, etc.) suchen kann. Zudem kann er die gefundenen Wohnobjekte nach ihren verschiedenen Attributen sowie deren Erreichbarkeit zu gewissen, für ihn interessanten, Punkten (POIs) bewerten lassen. Besonders sei hervorzuheben, dass die Wohnungen auf einer Karte als Markierungen angezeigt werden um die Suche für den Benutzer zu vereinfachen. Von den angezeigten Wohnobjekten kann sich der Benutzer weitere Details ansehen und diese Vergleichen oder zur Seite des Wohnungsanbieter wechseln um mit dem Anbieter des Wohnobjektes in Kontakt zu treten. Weiterhin ist es möglich sich, nach der Anmeldung, Objekte zu merken um sie sich später anzeigen lassen zu können.

3. Grundsätzliche Struktur- und Entwurfsprinzipien

Das Gesamtsystem ist aufgeteilt in 2 Teile. (siehe Abbildung 1)

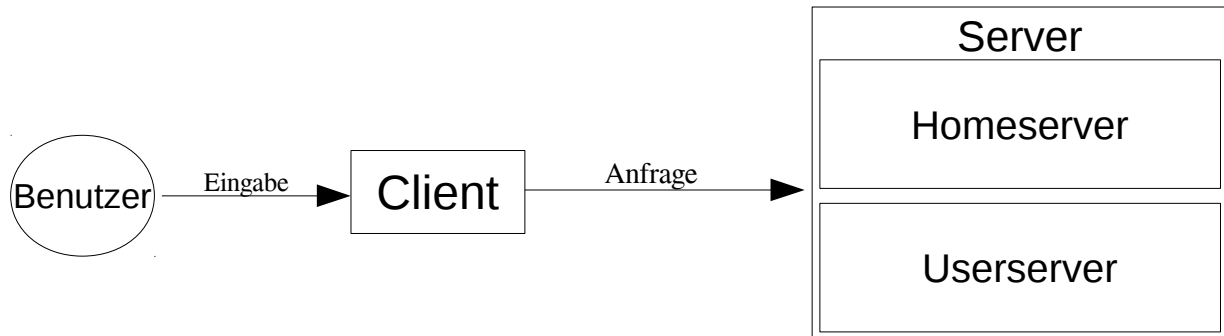


Abb.1

Der Client

Der Client stellt eine Website zur Verfügung über welche man Wohnobjekte suchen kann. Diese können weitergehend nach ihren Kriterien wie z.B. Preis, Wohnfläche, Lage (in Abhängigkeit der eingegebenen POI's) etc. bewertet werden.

Der Server

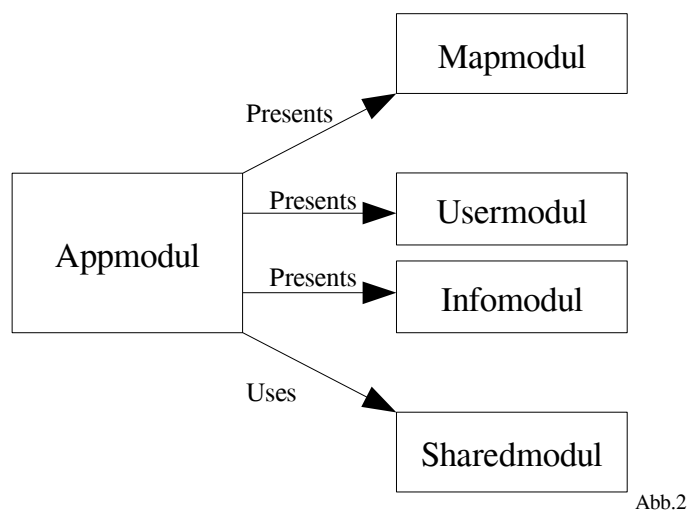
Der Server stellt eine API zur Verfügung. Dabei werden jedoch implikativ zwei Server implementiert. Den ersten nennen wir an dieser Stelle Homeserver. Welcher das Suchen der Wohnobjekte ermöglicht, indem er sie einmal täglich von Immobilienscout lädt und in einer eigenen Datenbank speichert.

Der zweite Server wird Userserver genannt. Dieser ermöglicht das Anlegen von Benutzern, sowie deren Anmeldung. Weiterhin ermöglicht es der Userserver, dass man favorisierte Wohnobjekte speichern und zu einem späteren Zeitpunkt wieder aufrufen kann.

4. Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Der Client

Der Client basiert auf dem Angular2 Framework und ist der größte Teil des Systems. Dieser besteht dabei insgesamt aus fünf Modulen (siehe Abbildung 2), welche mit App-, Map-, Shared-, Info- und User-Modul benannt sind. Der Aufbau dieser Module sind dabei in der Angular 2 Dokumentation eingehend erläutert.



Das App-Modul stellt in der Form der App-Komponente dabei den Einstiegspunkt der gesamten Anwendung dar und ruft weitere Komponenten auf.

Das Shared-Modul beinhaltet Fragmente des Systems, die an verschiedenen Stellen wiederverwendet werden und unabhängig von diesen agieren können. Das Shared-Modul besteht dabei aus folgenden Komponenten:

- *Edge-Control Component & Side-Control Component:* Diese beiden Komponenten stellen Kontrollfelder dar welche mit verschiedenen Funktionalitäten belegt werden können.
- *Query Service:* Dieser Service beinhaltet Funktionen zur Manipulation der Url-Query eines Browsers.

- *Transfer Service*: Über diesen Service können Informationen zwischen Komponenten ausgetauscht werden, ohne dass diese eine direkte Abhängigkeit erzeugen.

Das Info-Modul ermöglicht es den Benutzern Informationen anzeigen zu lassen, darunter z.B. wie man die Website bedient.

Das Map-Modul zeigt die Karte an, auf der man POI's eintragen kann und verfügbare Wohnungen angezeigt werden. Dieses Modul besteht dabei aus:

- dem *Home-Service*, welcher eine Schnittstelle zum Home-Server bildet
- dem *Geocode-Service*, wessen Aufgabe es ist eine Verbindung zu der Google-Maps-Geolocation-API herzustellen und somit Adressen zu Weltkoordinaten zu transformieren.
- der *Map-Komponente*, diese erlaubt es dem Benutzer auf einer Karte zu navigieren und alle benötigten Parameter zur Suche einer Wohnung anzugeben. Des Weiteren nutzt die Map Komponente den Home Service um dem Benutzer entsprechende Wohnungsobjekte anzuzeigen. Zuletzt kann der Benutzer einzelne Wohnobjekte auswählen, um sich diese von der Detail Komponente anzeigen zu lassen.
- der *Detail-Komponente*, welche die Anzeige von jeglichen Details zu einem Wohnobjekt ermöglicht.
- dem *Time-Service*, welcher den Route360-Service anspricht und somit die Route360-API, durch welche Reisezeiten zwischen den Punkten ermittelt werden.
- dem *Rating-Service*, welcher anhand der Reisezeiten, des Preises, der Wohnfläche usw. die Bewertungen der jeweiligen Wohnobjekte erstellt.

Das User-Modul sorgt sich um die Aktionen von Benutzern mit ihren Benutzerkonten. Dabei besteht es aus:

- dem *User-Service*, welcher das Speichern und Abrufen von Nutzerdaten und die Verwaltung von Benutzerkonten erlaubt.
- der *User-Komponente*, erlaubt die Registration und Anmeldung eines Benutzers.
- der *LoggedIn-Komponente*, stellt Funktionen zur Verfügung die ein Nutzer nach der Anmeldung bekommt.

4.2 Der Server

Wie oben beschrieben besteht der Server implikativ aus zwei Servern, aus einem der sich um Nutzerdaten etc. kümmert (Userserver) und einem der sich um mit den Wohnungsobjekten und deren Daten beschäftigt (Homeserver).

Der Server verwendet in der „index.ts“-Datei Expressjs um die Routen zu erstellen. Somit wird eine API nach außen zur Verfügung gestellt, welche dementsprechen die Anfragen zum Userserver oder zum Homeserver weiterleitet.

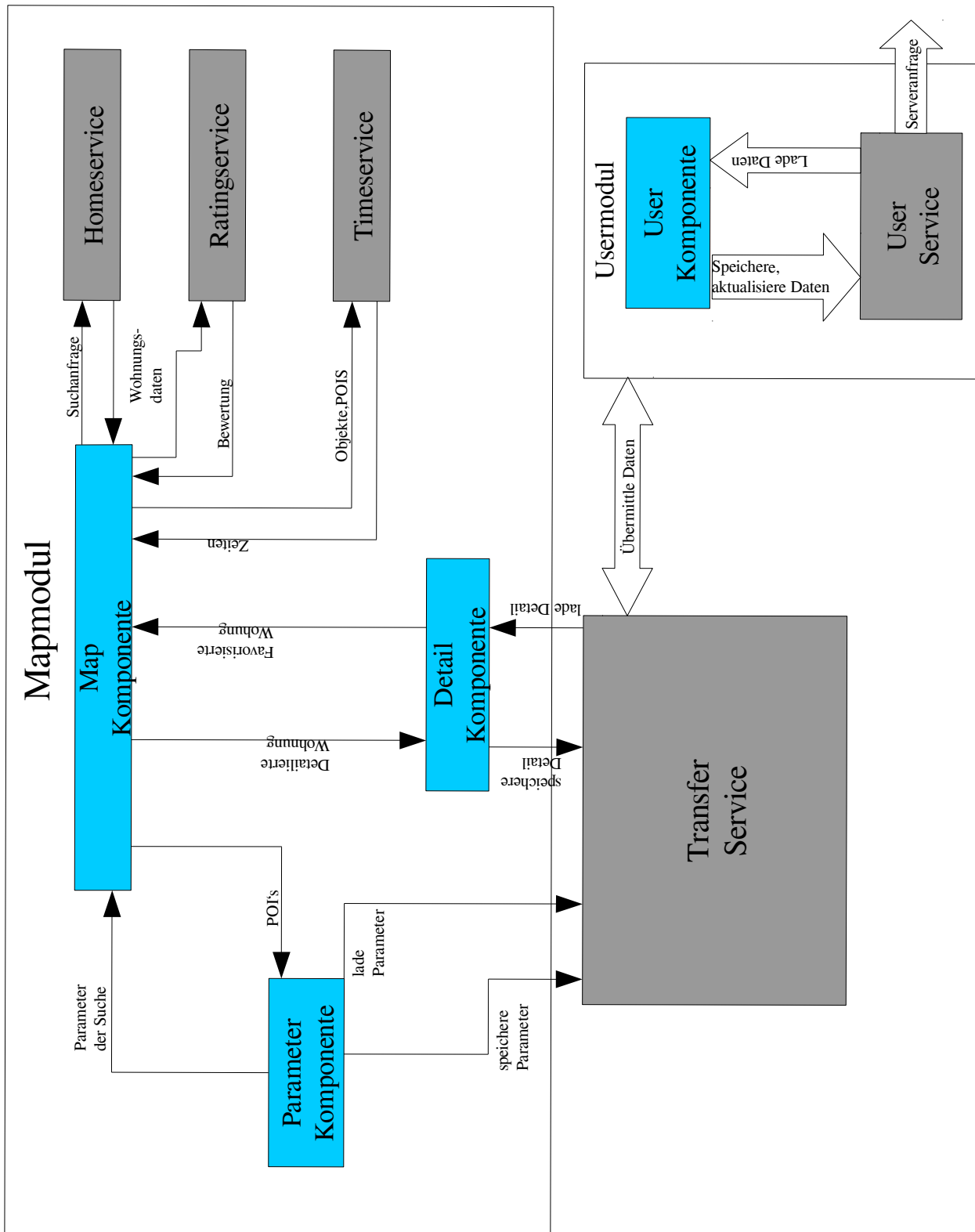
Der Teil des Servers der Homeserver genannt wird besteht aus

- einem Pipemanager, welcher die Funktion des Homservers bündelt. (Momentan nur noch die gethomesbyslice-Funktion)
- einem Datenbankservice, welcher je nach Anfrage mehrere oder nur spezielle Wohnobjekte zurückliefert
- und einem Crawler, welcher unabhängig von restlichen Server agiert und die Wohnungsdaten von externen Anbietern sammelt und in der Datenbank speichert.

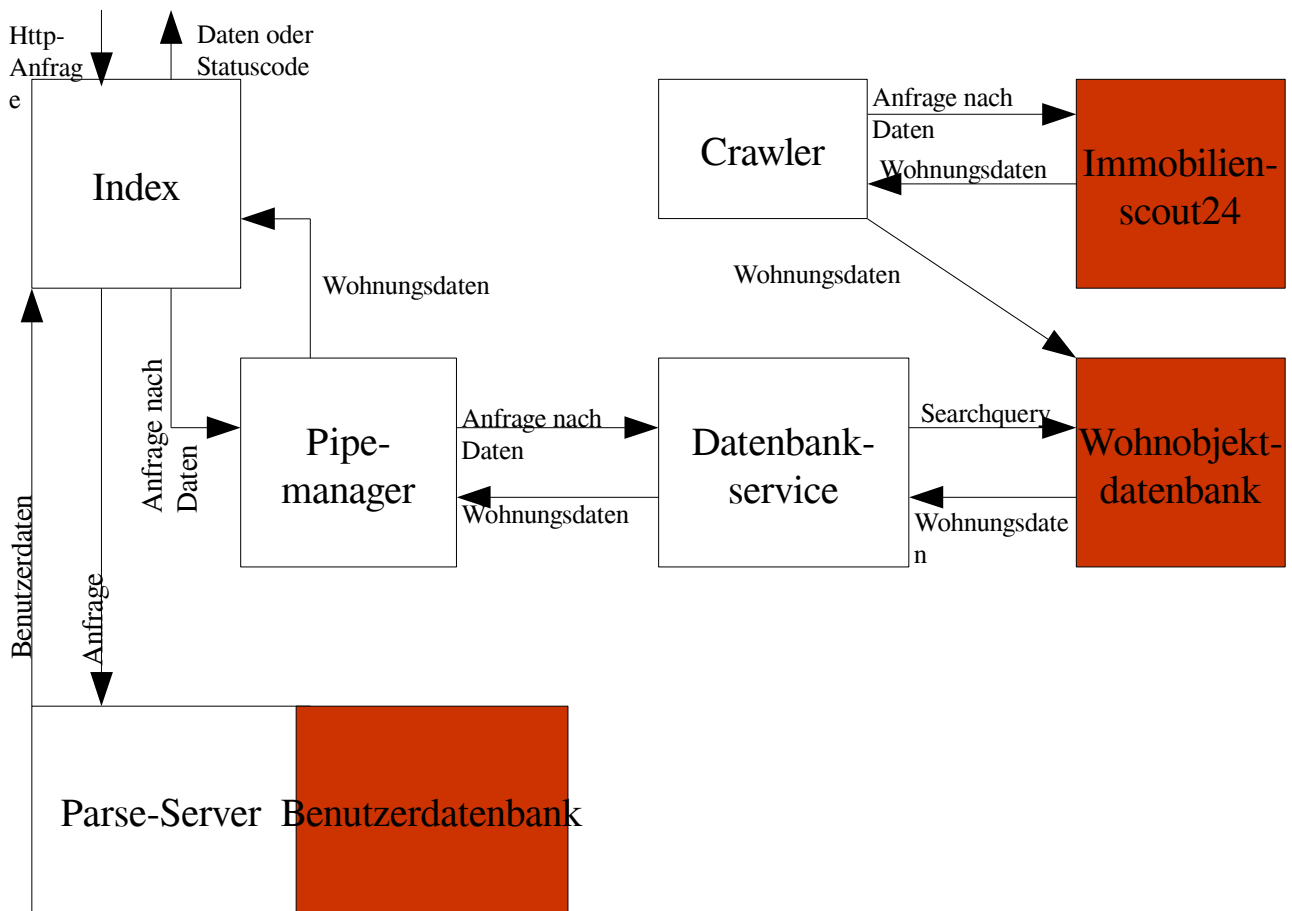
Der Userabschnitt des Servers implementiert mit Hilfe des Parse-Server Frameworks die Funktionen zur Benutzerverwaltung.

5. Datenmodelle

5.1 Datenfluss des Clients



5.2 Datenfluss des Servers



6. Glossar

- *Benutzer*: Der Benutzer ist in diesem Szenario eine Person welche den in diesem Dokument beschriebenen Service zu einem beliebigen Zeitpunkt nutzt.
- *POIs*: Points of Interest (engl.) oder auch Punkte von Interesse (dt.) bezeichnen in diesem Kontext geographische Punkte auf der Welt, welche für den Benutzer von Interesse sind.

- *Wohnobjekt*: Ein Wohnobjekt stellt ein beliebiges räumliches Objekt dar in dem ein Mensch leben kann, welches von einer Drittperson zur Verfügung gestellt wird. Wie z.B. ein Haus, eine Wohnung oder ein Zimmer.
- *Schnittstellen*: Eine Schnittstelle beschreibt in diesem Kontext die Schnittstelle zu einem anderen Programm, über welche Daten übermittelt werden können.
- *API*: Eine API (Application Programming Interface) oder auch Programmierschnittstelle (dt.) ist ein Softwarestück, welches als Softwareschnittstelle von beliebiger Software auf eine bestimmte Software dient.
- *REST / RESTful*: Ist eine Design Spezifikation für eine API mehr unter https://de.wikipedia.org/wiki/Representational_State_Transfer.
- *Benutzerdaten*: Sind beliebige Daten welche von einem Benutzer in unserer Anwendung verwaltet werden. Z.B. Benutzernamen, Passwörter, Emails, etc. .
- *Service*: Ein Service ist eine Dienstleistung die von einem bestimmten Stück Software angeboten wird.
- *Komponente*: Eine Komponente ist ein für sich frei agierendes Stück Software welches aber ohne einen größeren Kontext keine wirkliche Verwendung hat.
- *Modul*: Ein Modul besteht aus verschiedenen Komponenten oder Modulen und stellen ein vollständiges Softwaresystem dar welches allerdings weiterhin in einen noch größeren Kontext eingebunden werden kann.
- *Client*: Ein Client ist eine Softwareinstanz welche von einer anderen Softwareinstanz (dem Sever) meist über das Internet, einen Service bezieht.

- *Request & Respond*: Ein Request ist eine Anfrage an einen Server welcher mit einem Respond dem Client antwortet. Dabei geht es um den Datenaustausch zwischen diesen beiden.