

Konzeption und Prototyp eines Verwaltungstool für Buchungen von Wohnräumen

Analyse der verteilten Anwendung

Eine Arbeit von
Stefan Baumann, Florian Gumbel
entstanden im Sommersemester 2016 im Fach
„Entwicklung verteilter Anwendung“ bei
Prof. Dr. Karim Kremer
(16.09.2016)



Technische Hochschule Mittelhessen
Wilhelm-Leuschner-Straße 13
61169 Friedberg

Zusammenfassung

Eine Agentur, die Wohnräume vermietet, möchte ein Programm haben um die Buchungen zu Verwalten. Zu den Prämissen gehört, dass es Betriebssystem unabhängig nutzbar ist und Änderungen auf schnellstmöglichen weg an die Mitarbeiter kommuniziert wird um Doppebuchungen zu vermeiden. Es soll ebenfalls möglich sein zur gleichen Zeit Buchungen, Kunden und Wohnungen hinzufügen zu können.

Inhaltsverzeichnis

1	Einleitung	1
2	Technologie	3
2.1	Angular	3
2.2	Firebase	3
2.3	Material Design	3
2.4	Grunt	3
2.5	NPM	3
2.6	Bower	3
3	Projekt	5
3.1	Zielsetzung	5
3.2	Umsetzung	5
3.2.1	Backend	6
3.2.2	Frontend	6
3.2.3	Funktionen	6
4	Aufbau Projekt	7
4.1	Frontend	7
4.2	Backend	7
5	Fazit	9

1 Einleitung

Für die Vermietung seiner Ferienwohnungen und Häuser benötigt ein Unternehmen eine Software zur Verwaltung der jeweiligen Buchungen. Da das Unternehmen über mehrere Buchungsagenten verfügt, ist eine Anwendung nötig, die das parallele Arbeiten an Verschiedenen Geräten ermöglicht. Dazu gehören nicht nur Computer und Laptops sondern auch Tablets um schnell von unterwegs eine Buchung einzutragen. Besonders wichtig ist dem Unternehmen die schnelle Aktualisierung der Daten in der Software um eventuelle Doppelbuchungen zu vermeiden und das gleichzeitige Arbeiten zu ermöglichen.

2 Technologie

2.1 Angular

2.2 Firebase

2.3 Material Design

2.4 Grunt

2.5 NPM

2.6 Bower

3 Projekt

Durch die in der Einleitung erwähnte Problemstellung ergeben sich folgende Zielsetzung und Umsetzung.

3.1 Zielsetzung

Prototypische Entwicklung einer Software zur Verwaltung von Buchungen für Ferienhäuser und Wohnungen. Eine der Anforderungen an die Software war die Nutzung auf verschiedenen Endgeräten. Das hat zur Folge, dass auch die Betriebssysteme variieren können. Daraus ergibt sich, dass die Anwendung plattformunabhängig nutzbar sein muss.

Für diesen Fall eignet sich eine Webanwendung besonders gut. Sie ist von jedem Gerät aus nutzbar und benötigt in der Regel nicht mehr als einen Internetbrowser. Auch die Installation eines extra Programmes erübrigt sich. Allerdings ist eine stetige Internetverbindung Voraussetzung. In Anbetracht der weiteren Bedingungen an die Software ist diese aber auch bei einer nativ installierten Anwendung Voraussetzung. Da sie für den stetigen Abgleich der Daten zwingend vorhanden sein muss. Das Ziel ist es also eine Webanwendung mit einer Benutzeroberfläche zu erstellen, die auf jedem Gerät angezeigt werden kann, sofern es über Internet und einen entsprechenden Browser verfügt. Des Weiteren wird ein entsprechendes Backend benötigt, das die Daten speichert.

Weitere Anforderungen an die Anwendungen sind die Verwaltung von Kunden, der zu vermietbaren Ferienhäuser und Wohnungen sowie die eigentliche Buchung. Das Ganze soll übersichtlich dargestellt werden und einfach und schnell benutzbar sein.

3.2 Umsetzung

Für die Umsetzung der eigentlichen Webanwendung kommen JavaScript, HTML und CSS zum Einsatz. Zu dem werden die in Kapitel 2 genannten Frameworks und Tools zur Unterstützung eingesetzt.

3.2.1 Backend

Für das Speichern der Daten soll die in Firebase integrierte objektbasierte Datenbank zum Einsatz kommen. Ebenfalls übernimmt ein weiterer Dienst von Firebase die Authentifizierung bei der Anmeldung.

3.2.2 Frontend

Neben den oben genannten Script- und Auszeichnungssprachen soll für die Logik und Strukturierung der Anwendung das Framework AngularJS genutzt werden. Dazu passend soll das Layout mittels der Angular Version von Materialize gestaltet werden.

3.2.3 Funktionen

Die Funktionen sollen sinnvoll in die Webanwendung integriert werden. Dabei sollen sie schnell und ohne Umwege erreichbar sein. Zu den Hauptfunktionen zählt das Hinzufügen und Bearbeiten von Kundendaten, Ferienwohnungen und der Buchung selbst. Folgende Informationen sollten für jeden Kunden zur eindeutigen Identifizierung gespeichert werden:

- Firma (Falls vorhanden)
- Vor- und Nachname
- Strasse und Hausnummer
- Postleitzahl und Stadt
- Telefonnummer
- Geburtstag
- Feld für Zusatzinformationen

Die zu vermietenden Ferienhäuser und Wohnungen sollten folgenden Attribute haben:

- Name des Objektes
- Anzahl der Personen die es maximal beherbergen kann

Um eine Buchung zu erstellen sollte die folgenden Informationen festgehalten werden:

- Kunde, der das Objekt bucht
- Objekt, das gebucht wird
- Anzahl der Personen
- Startdatum
- Enddatum

4 Aufbau Projekt

4.1 Frontend

4.2 Backend

5 Fazit

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Glossary

Cascading Style Sheet Gestaltungssprache für elektronische Dokumente und Programme wie HTML-Webseiten oder JavaFX-Anwendungen..

CSS *siehe* Cascading Style Sheet.

Database Management System System, welches die Datenbank aufbaut und verwaltet. Ziel ist ein möglichst effizientes und einfach zu bedienendes System..

DBMS *siehe* Database Management System.

Extensible Markup Language Kompaktes Datenformat, in etwa vergleichbar mit JSON..

Graphical User Interface Frontend-Ansicht, welche dem Benutzer eines Programms ausgeliefert wird..

GUI *siehe* Graphical User Interface.

HTML *siehe* Hypertext Markup Language.

HTTP *siehe* Hypertext Transfer Protocol.

Hypertext Markup Language Eine textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten..

Hypertext Transfer Protocol Protokoll zur Übertragung von Daten über ein Rechnernetz..

JAVAFX Framework zum Aufbau von einfachen bis komplexen Java-Frontendsystemen..

JavaScript Object Notation Kompaktes Datenformat in leicht verständlicher Textform, in etwa vergleichbar mit XML..

JSON *siehe* JavaScript Object Notation.

Plain Old Java Object Ein Java-Objekt im herkömmlichen Sinne..

POJO *siehe* Plain Old Java Object.

QR *siehe* Quick Response Code.

Quick Response Code Zweidimensionaler Code, der schnell durch Maschinen gelesen werden kann..

SQL *siehe* Structured Query Language.

Structured Query Language Datenbanksprache, um durch Datenbanken zu navigieren, Bearbeitungen, Löschvorgänge und Neueintragungen vorzunehmen..

TOKEN Methode zur Autorisierung von Software-Diensten..

UI *siehe* User Interface.

Uniform Resource Locator Identifizierung einer Netzwerkressource, beispielsweise ein Server..

URL *siehe* Uniform Resource Locator.

User Interface Siehe Graphical User Interface.

XML *siehe* Extensible Markup Language.