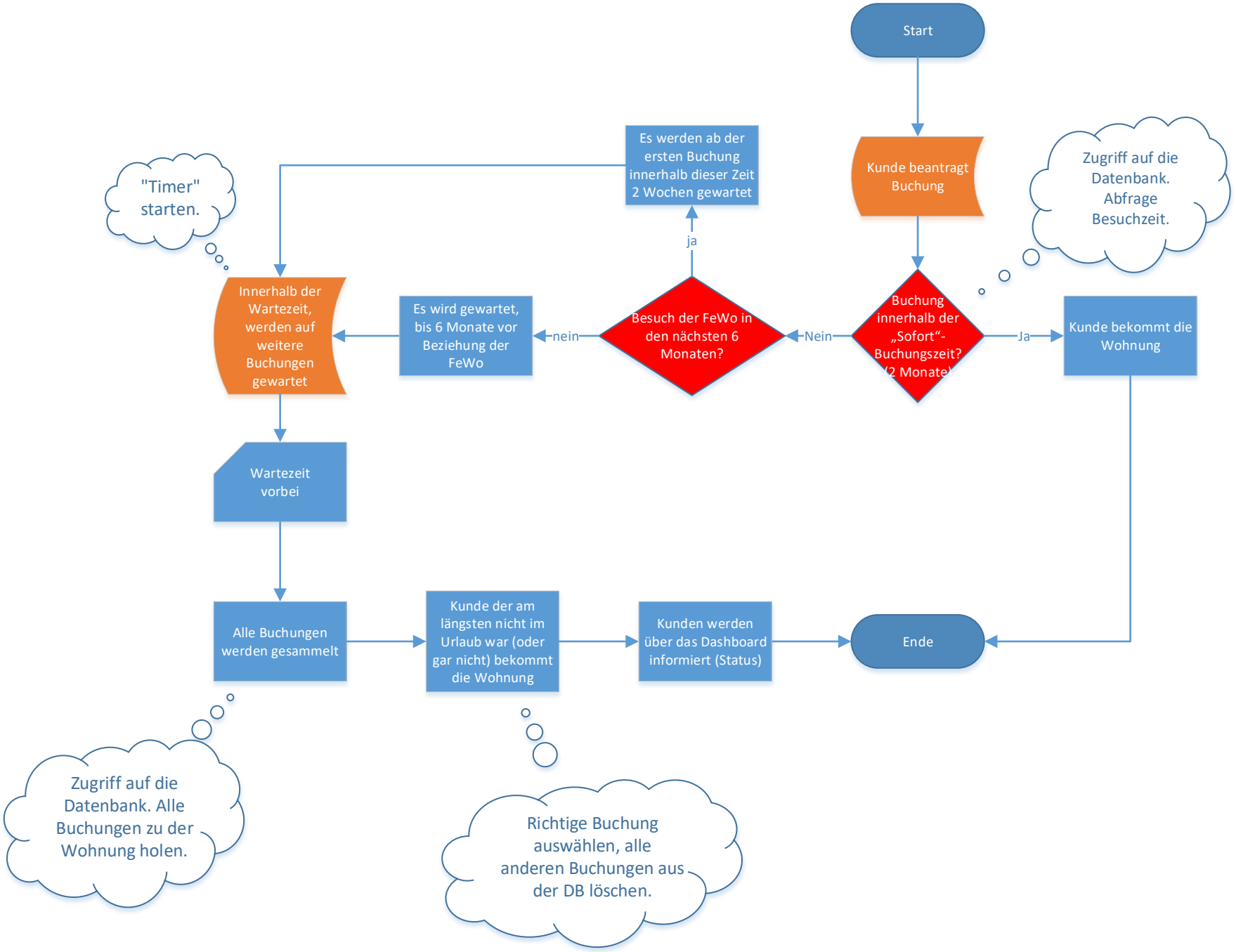
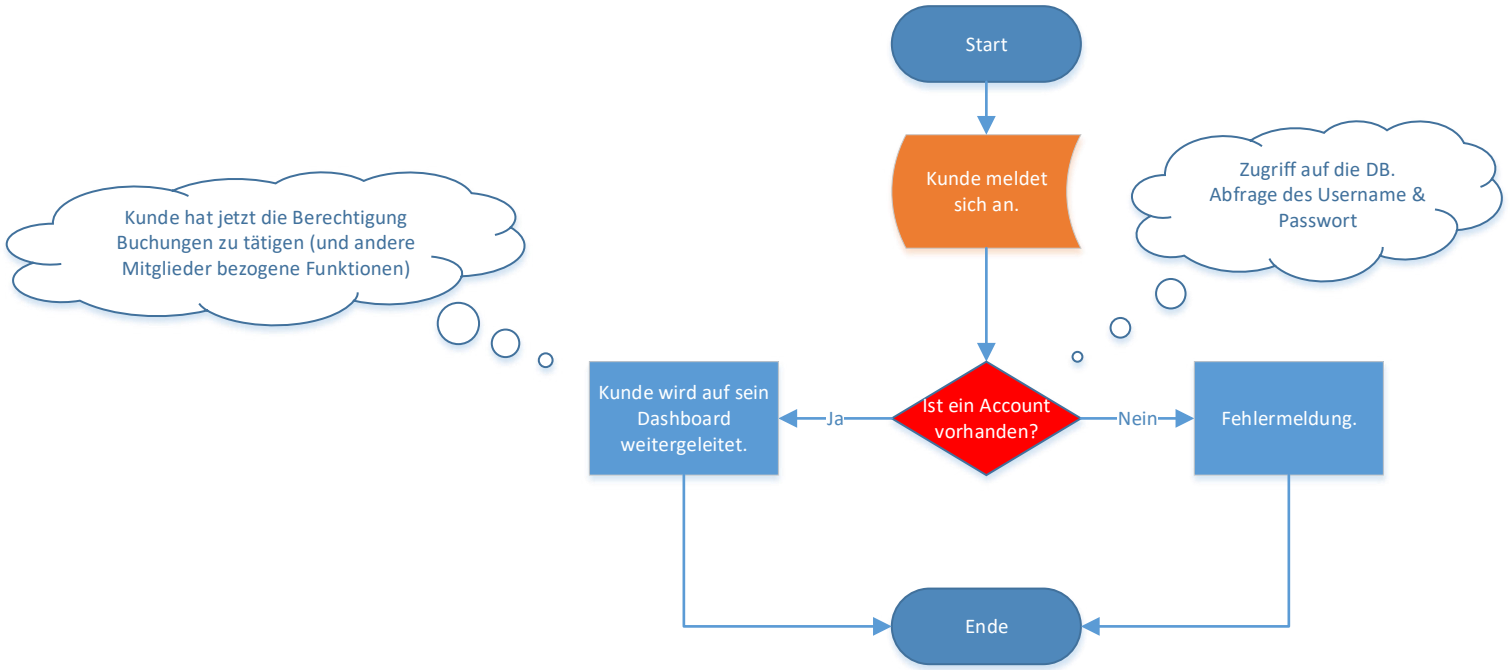
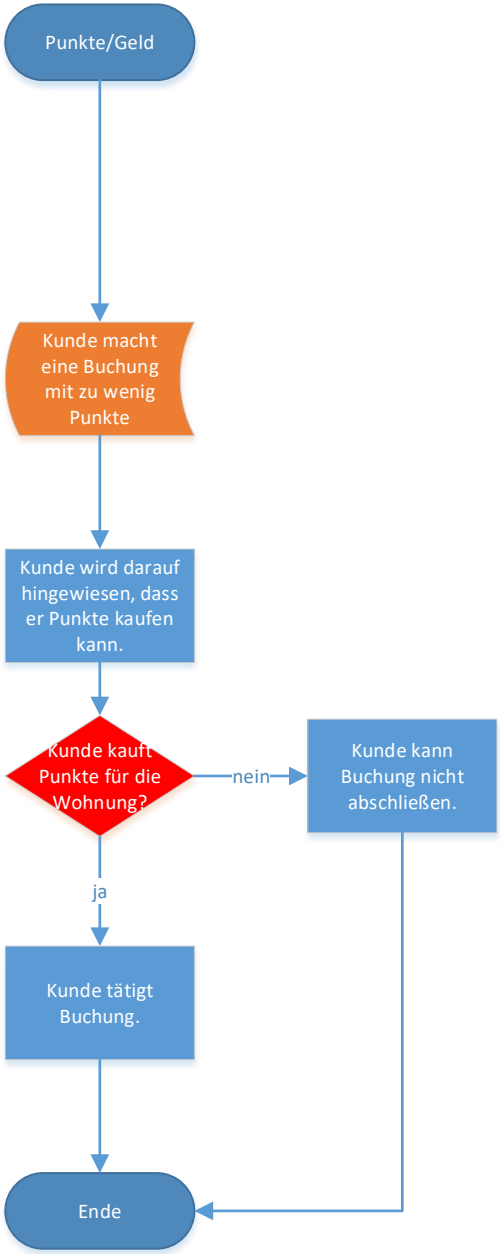
Software Spezifikation 2
Feindesign

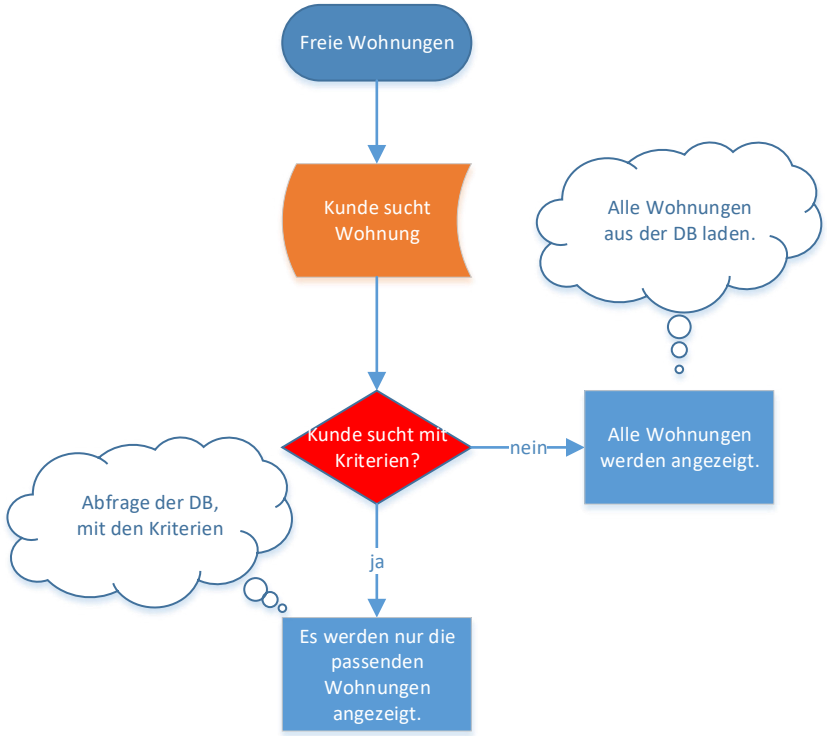
Übersicht

1. Diagramme (Autor: Tobin Choinowski)
   1.1 Fairness
   1.2 Login
   1.3 Umrechnung Geld <-> Punkte
   1.4 Suche freier Wohnungen
2. Modulbeschreibungen (Autor: Carmen Schmider)
   2.1 Startseite
   2.2 Dashboard
   2.3 Buchungsseite
   2.4 Gebietsauswahl
   2.5 Ferienwohnungen eines Gebietes
   2.6 Ferienwohnung
   2.7 Loginseite
   2.8 Registrierung
   2.9 Impressum
   2.10 AGB
3. Schnittstellenbeschreibung (Autor: Jonas Hauß)
   3.1 Einleitung
   3.2 Hauptprojekt
   3.3 Backend
   3.4 Frontend
4. Ablaufplan Buchungen (Autor: Katharina Schwab)

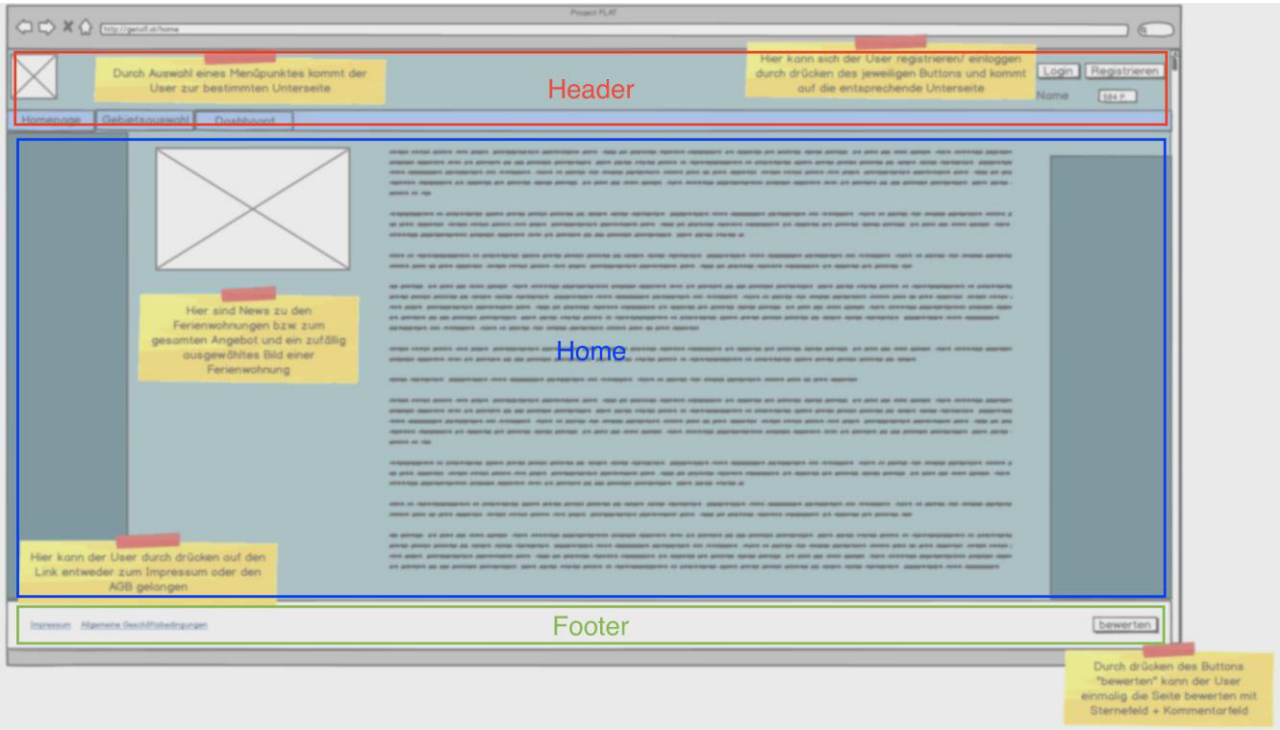

Start
Kunde beantragt Buchung
Zugriff auf die Datenbank. Abfrage Besuchzeit.
Buchung innerhalb der „Sofort“-Buchungszeit? (2 Monate)
Nein
Ja
Kunde bekommt die Wohnung
Besuch der FeWo in den nächsten 6 Monaten?
nein
ja
Es wird gewartet, bis 6 Monate vor Beziehung der FeWo
Es werden ab der ersten Buchung innerhalb dieser Zeit 2 Wochen gewartet
"Timer" starten.
Innerhalb der Wartezeit, werden auf weitere Buchungen gewartet
Wartezeit vorbei
Alle Buchungen werden gesammelt
Kunde der am längsten nicht im Urlaub war (oder gar nicht) bekommt die Wohnung
Kunden werden über das Dashboard informiert (Status)
Ende
Zugriff auf die Datenbank. Alle Buchungen zu der Wohnung holen.
Richtige Buchung auswählen, alle anderen Buchungen aus der DB löschen.


Start
Kunde meldet sich an.
Zugriff auf die DB. Abfrage des Username & Passwort
Kunde hat jetzt die Berechtigung Buchungen zu tätigen (und andere Mitglieder bezogene Funktionen)
Ist ein Account vorhanden?
Ja
Nein
Kunde wird auf sein Dashboard weitergeleitet.
Fehlermeldung.
Ende


Punkte/Geld
Kunde macht eine Buchung mit zu wenig Punkte
Kunde wird darauf hingewiesen, dass er Punkte kaufen kann.
Kunde kauft Punkte für die Wohnung?
nein
Kunde kann Buchung nicht abschließen.
ja
Kunde tätigt Buchung.
Ende


Freie Wohnungen
Kunde sucht Wohnung
Alle Wohnungen aus der DB laden.
Kunde sucht mit Kriterien?
nein
Alle Wohnungen werden angezeigt.
Abfrage der DB, mit den Kriterien
ja
Es werden nur die passenden Wohnungen angezeigt.

Modulbeschreibung

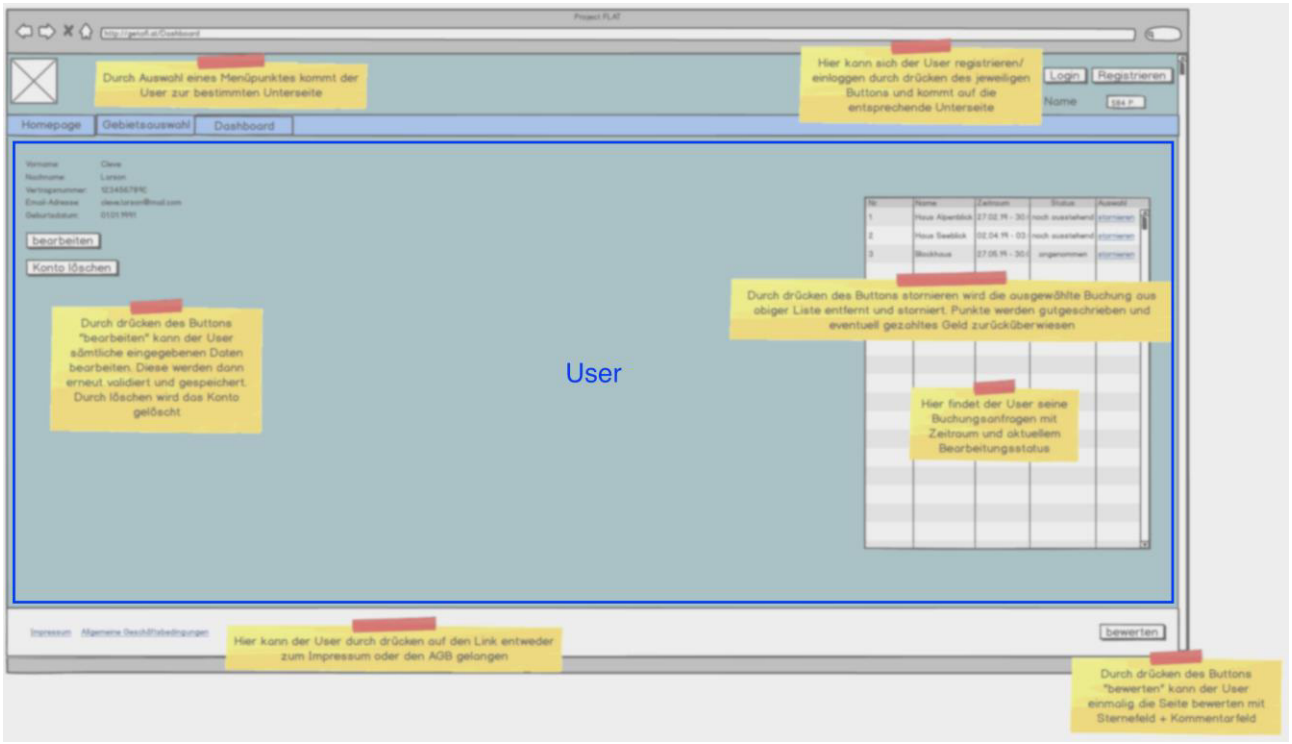


Startseite

Header	
Beschreibung	Übersicht aller wichtiger Informationen, Kopfzeile
Immer sichtbar	ja
enthält	1. Logo 2. Button zum Ein- und Ausloggen sowie zum Registrieren 3. Tabs um zwischen Homepage und Gebietsauswahl und Dashboard zu wechseln 4. Anzeige des Usernamens (mit Link zum Dashboard) und des aktuellen Punktstands (nur wenn der User eingeloggt ist)

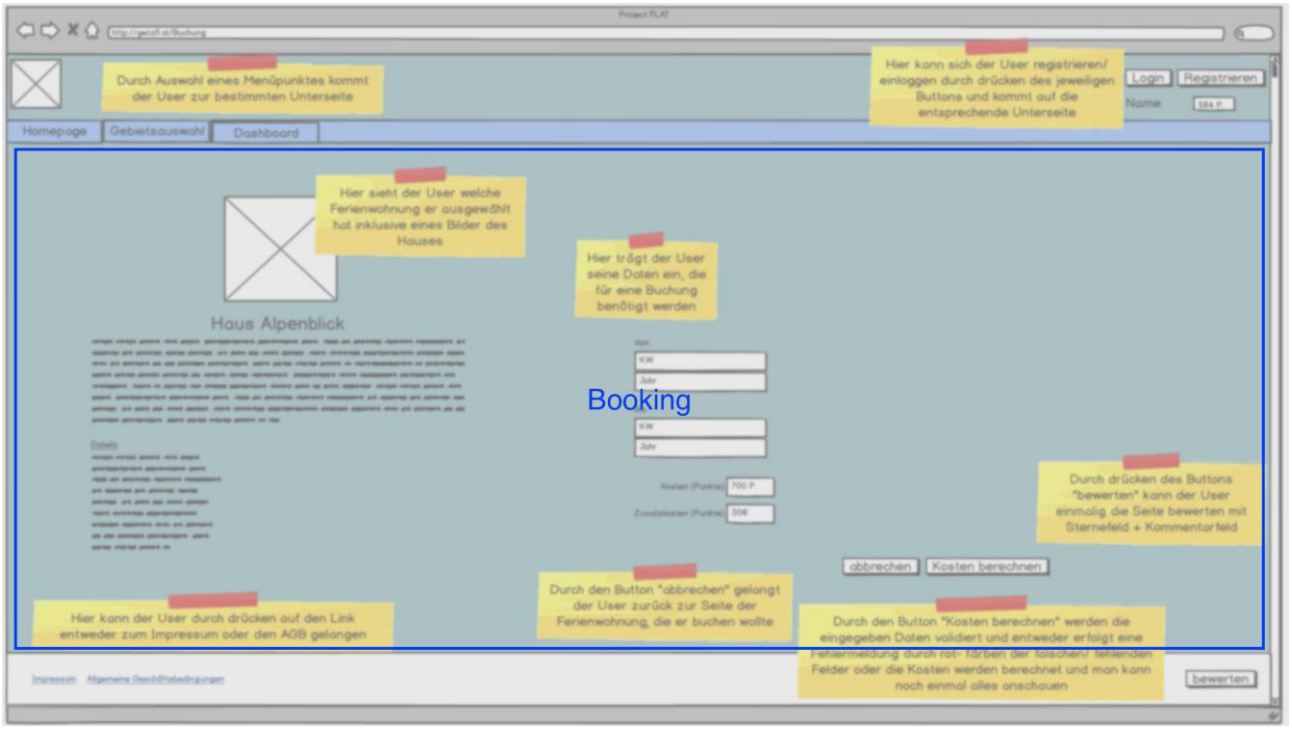
Home	
Beschreibung	Startseite, Newsfeed
Immer sichtbar	nein

Footer	
Beschreibung	Fußzeile mit Link Impressum und Allgemeine Geschäftsbedingungen und Bewerten-Button (Button wird nur angezeigt, wenn eine Bewertung möglich ist)
Immer sichtbar	ja



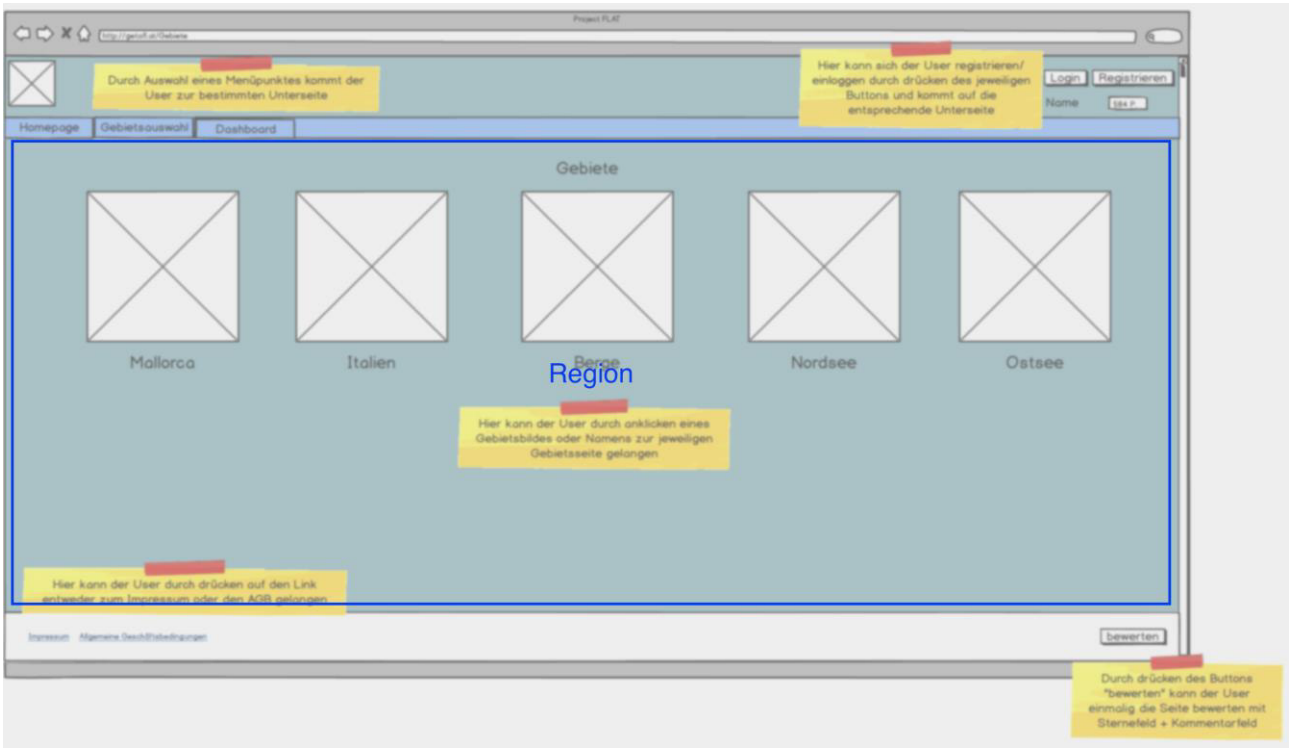
Dashboard

User	
Beschreibung	Ansicht der Profildaten und Buchungsanfragen
Immer sichtbar	nein
enthält	1. Profilangaben des angemeldeten Users 2. Button um Userdaten (Name und Vorname) zu ändern und User zu löschen 3. tabellarische Übersicht der Buchungsanfragen mit Zeitraum und Bearbeitungsstatus



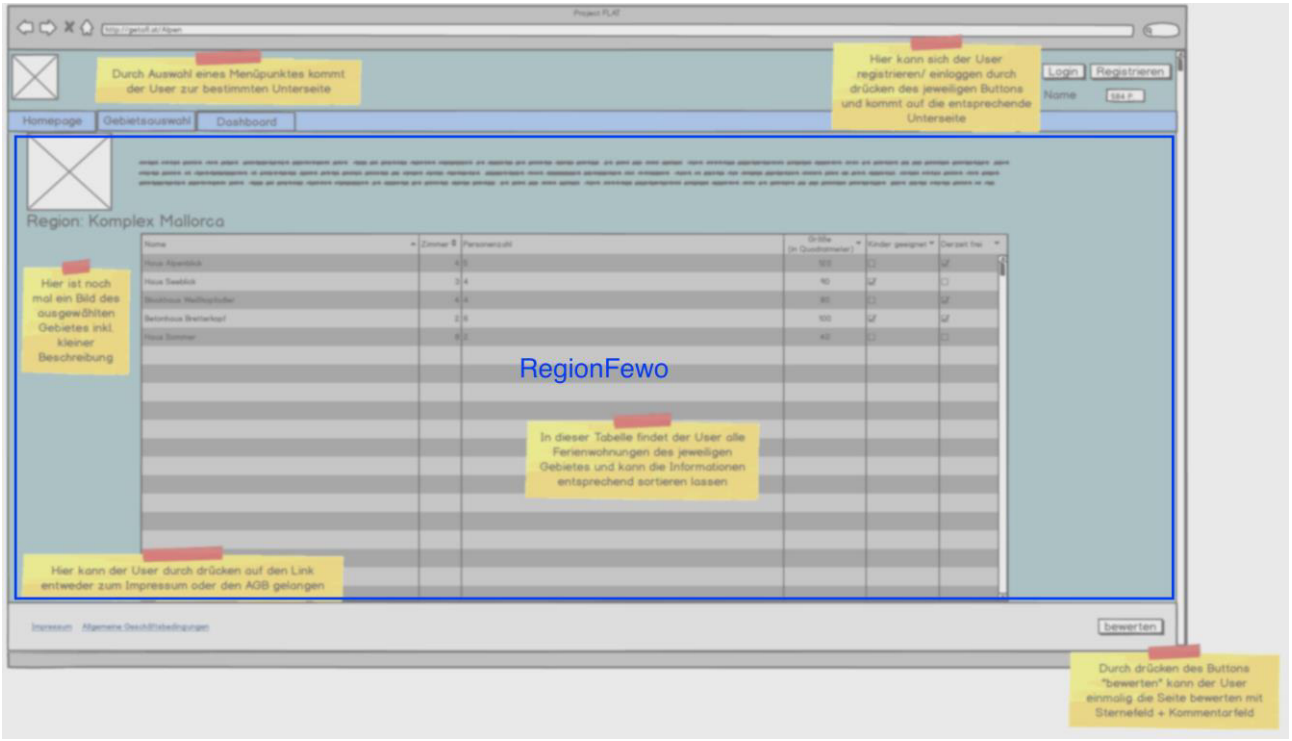
Buchungsseite

Booking	
Beschreibung	Foto und Beschreibung der Ferienwohnung, Formular für Buchungsanfrage
Immer sichtbar	nein
Anmerkungen	Foto der Ferienwohnung und Text (Kurzform) identisch mit Modul Flat



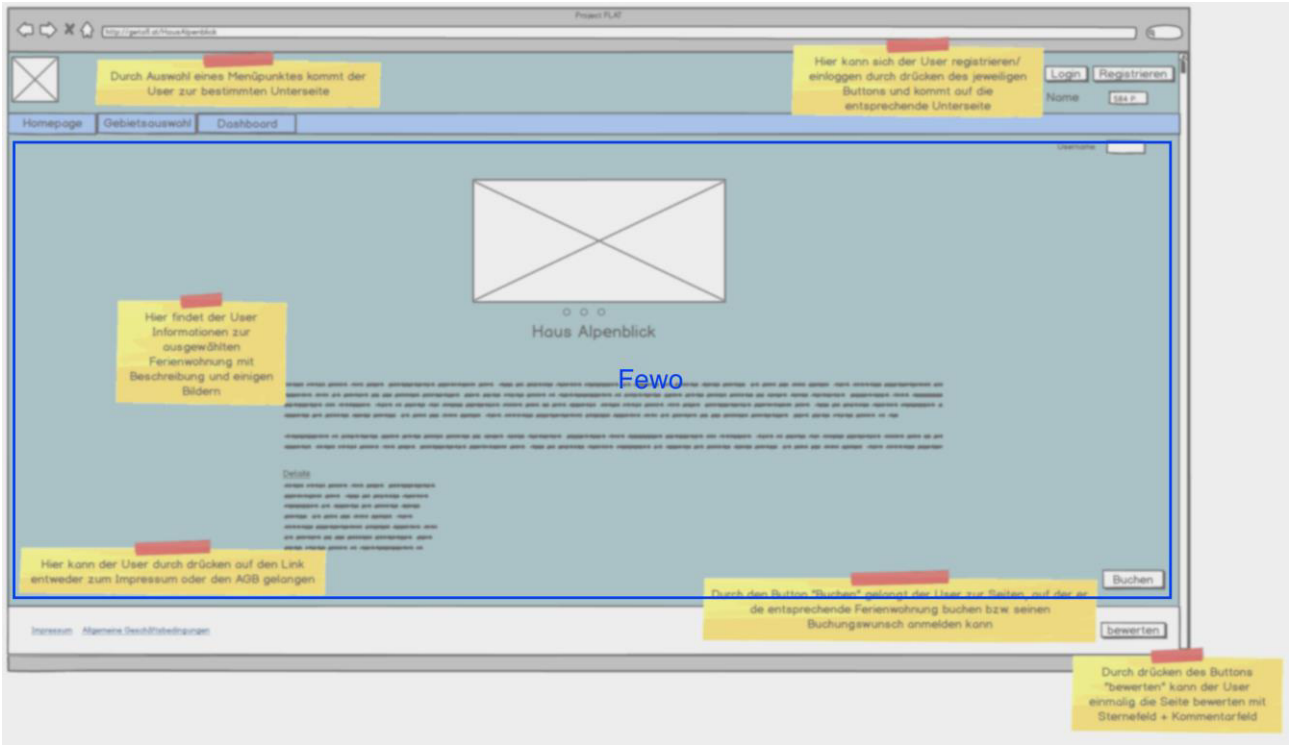
Gebietsauswahl

Region	
Beschreibung	Gebietsansicht mit Foto und Gebietsname
Immer sichtbar	nein



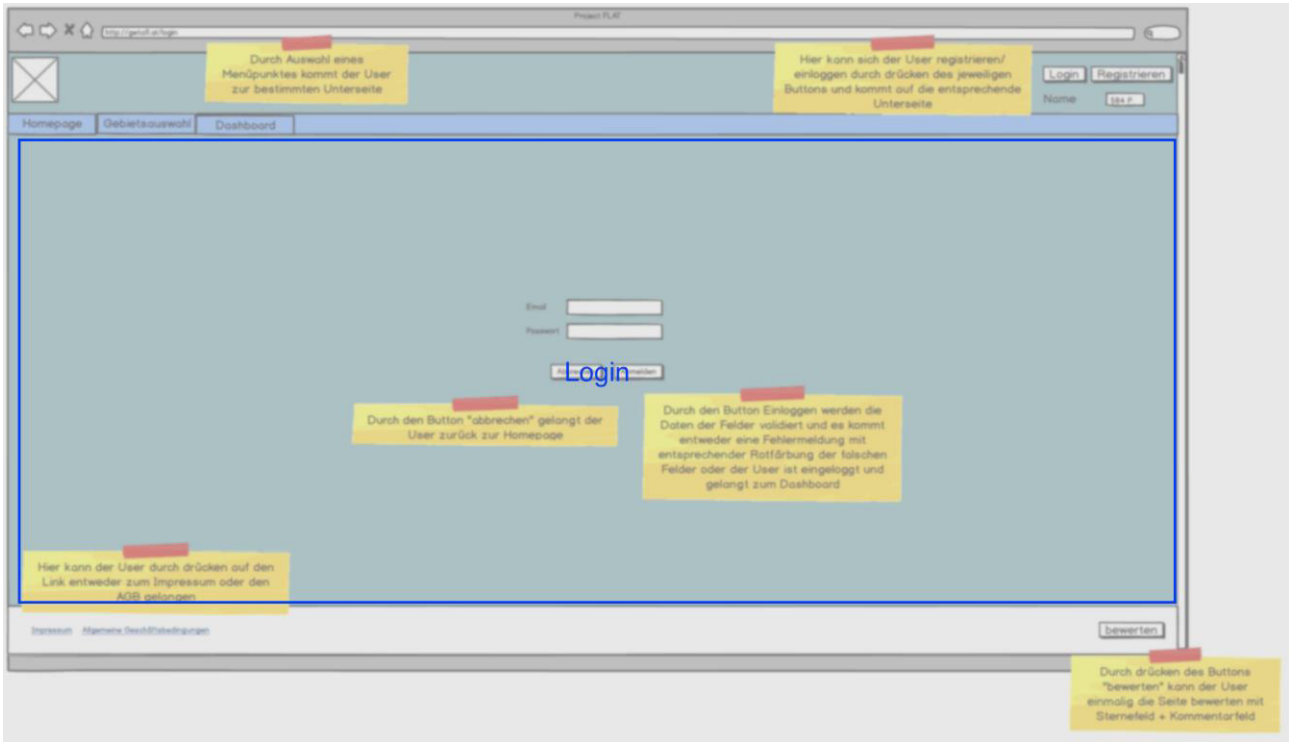
Übersicht Ferienwohnungen eines Gebietes

RegionFewo	
Beschreibung	Foto und Beschreibung des Gebiets und tabellarische Auflistung aller Ferienwohnungen
Immer sichtbar	nein



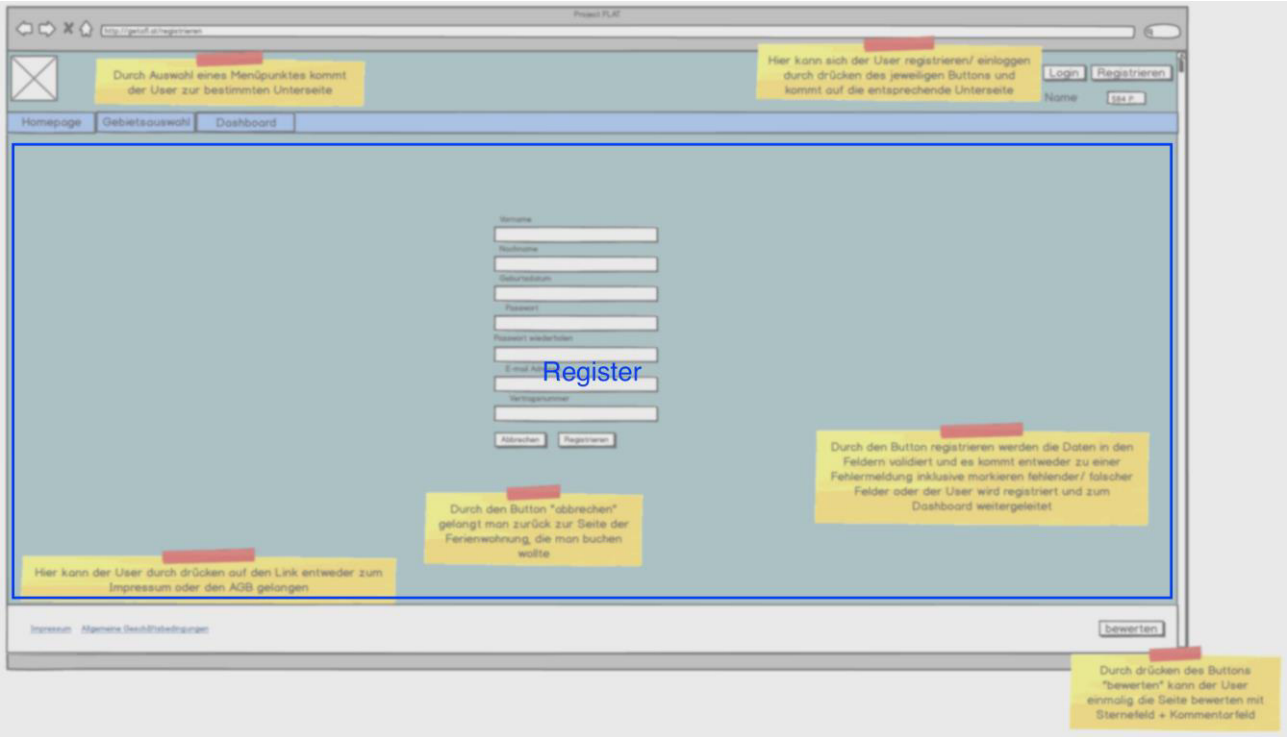
Ferienwohnung

Fewo	
Beschreibung	Ansicht der Ferienwohnung
Immer sichtbar	nein
enthält	1. Fotogalerie für weitere Fotos und Ansicht des ausgewählten Fotos 2. Kurzbeschreibung der Ferienwohnung 3. Detaillierte Beschreibung der Ferienwohnung 4. Button um auf Buchungsseite zu gelangen (nur wenn eingeloggt)
Anmerkungen	Hauptfoto und Kurzbeschreibung der Ferienwohnung identisch mit Modul Booking



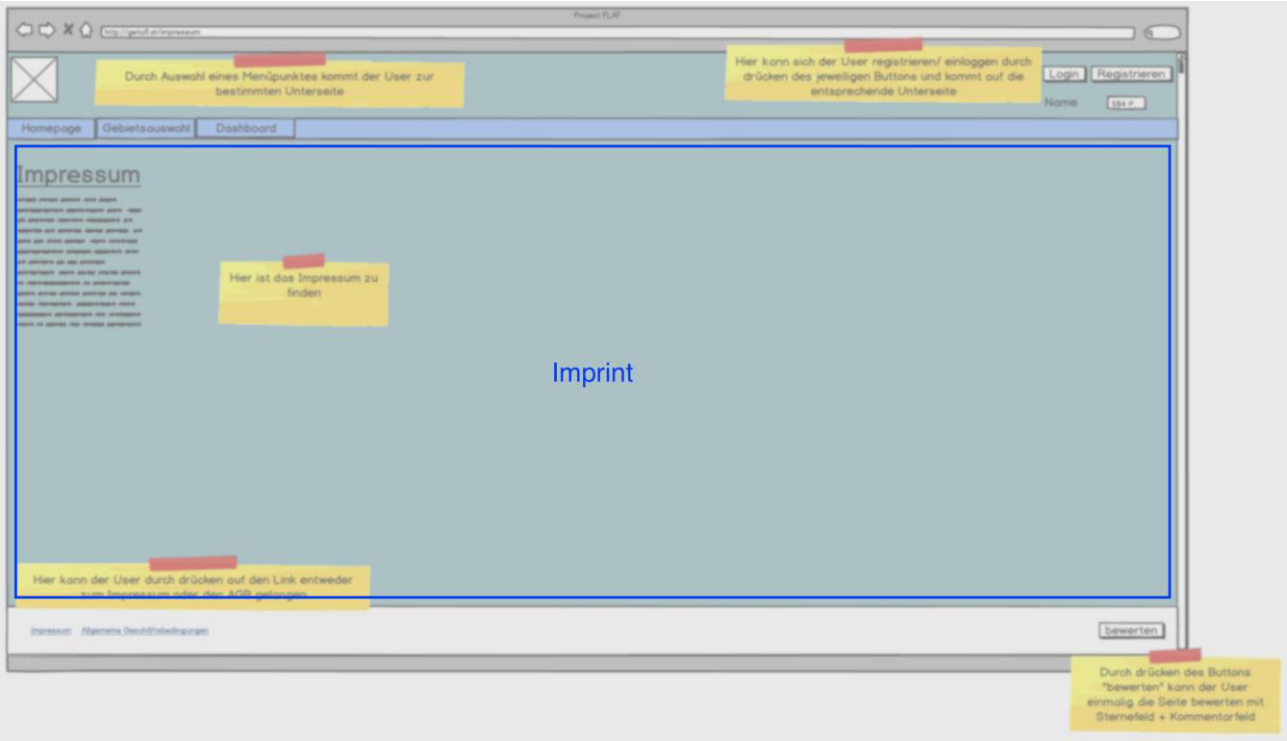
Loginseite

Login	
Beschreibung	Loginseite mit Textfeldern Username und Passwort, sowie Buttons Abbrechen und Einloggen
Immer sichtbar	nein



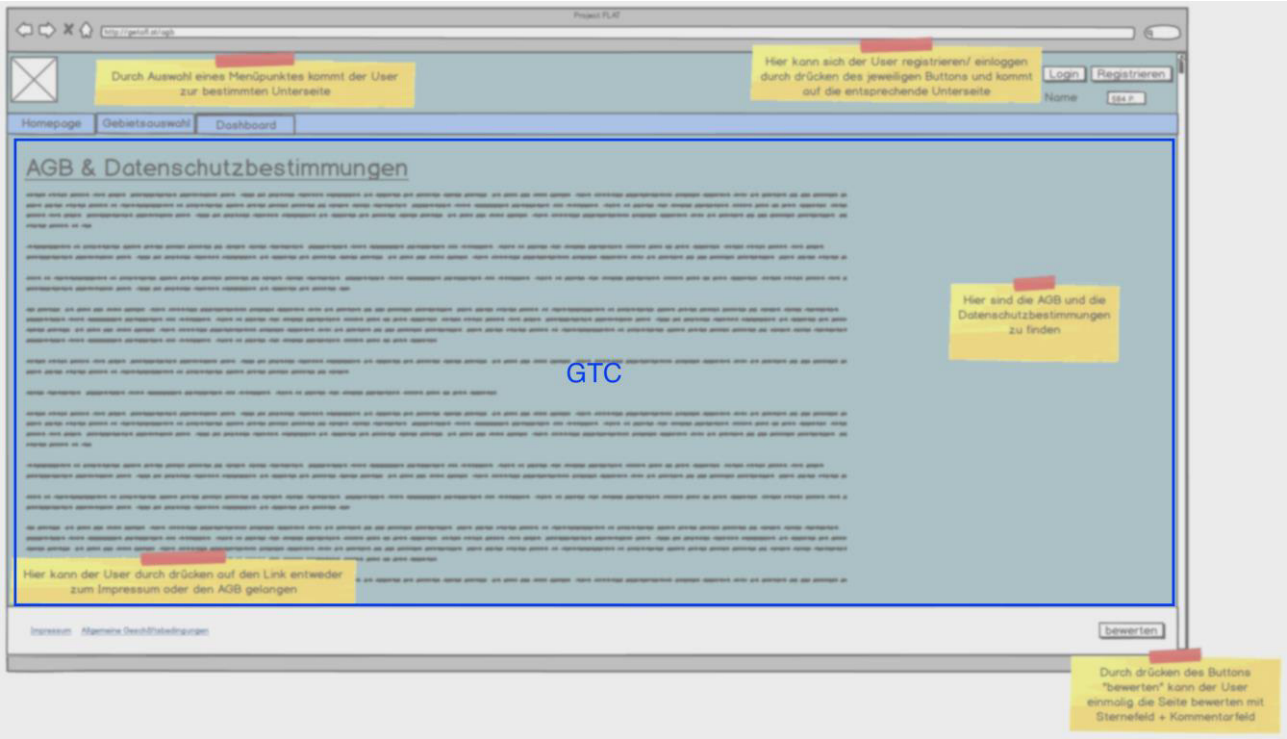
Registrierungsseite

Register	
Beschreibung	Registrierseite mit Textfeldern Username, Passwort, Geburtsdatum und E-Mail sowie Buttons Abbrechen und Registrieren
Immer sichtbar	nein



Impressum

Imprint	
Beschreibung	Impressum in Textform
Immer sichtbar	nein



Allgemeine Geschäftsbedingungen

GTC	
Beschreibung	Allgemeine Geschäftsbedingungen in Textform
Immer sichtbar	nein
Zuständigkeit	Carmen
Status	
Anmerkungen	

Schnittstellenbeschreibung

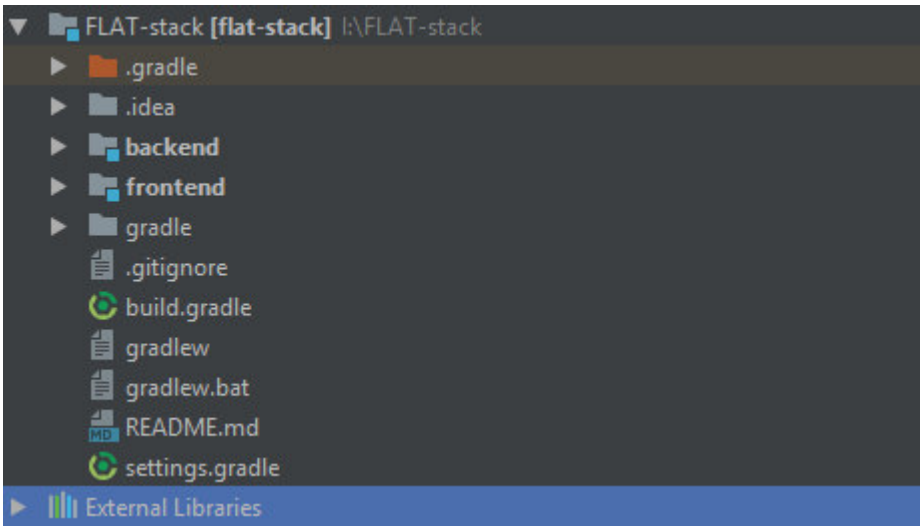
Einleitung

Die in der Technologiebeschreibung genannten Tools werden zusammen eingesetzt um das Projekt zu realisieren. Wie diese Realisierung umgesetzt wird, soll in diesem Dokument beschrieben werden.

Das Quellcode wird über das Versionskontrollsystem Git verwaltet und kann über Github eingesehen werden: https://github.com/getaflat/FLAT-stack/tree/rebuild

Hauptprojekt

Die Anwendung wird als Gradle-Projekt konfiguriert. Das Hauptprojekt nennt sich „flat-stack“ und beinhaltet zwei Untermodule: „backend“ und „frontend“.
Durch diese Einteilung können die beiden Hauptbestandteile des Projektes separat verwaltet bzw. verwendet werden.

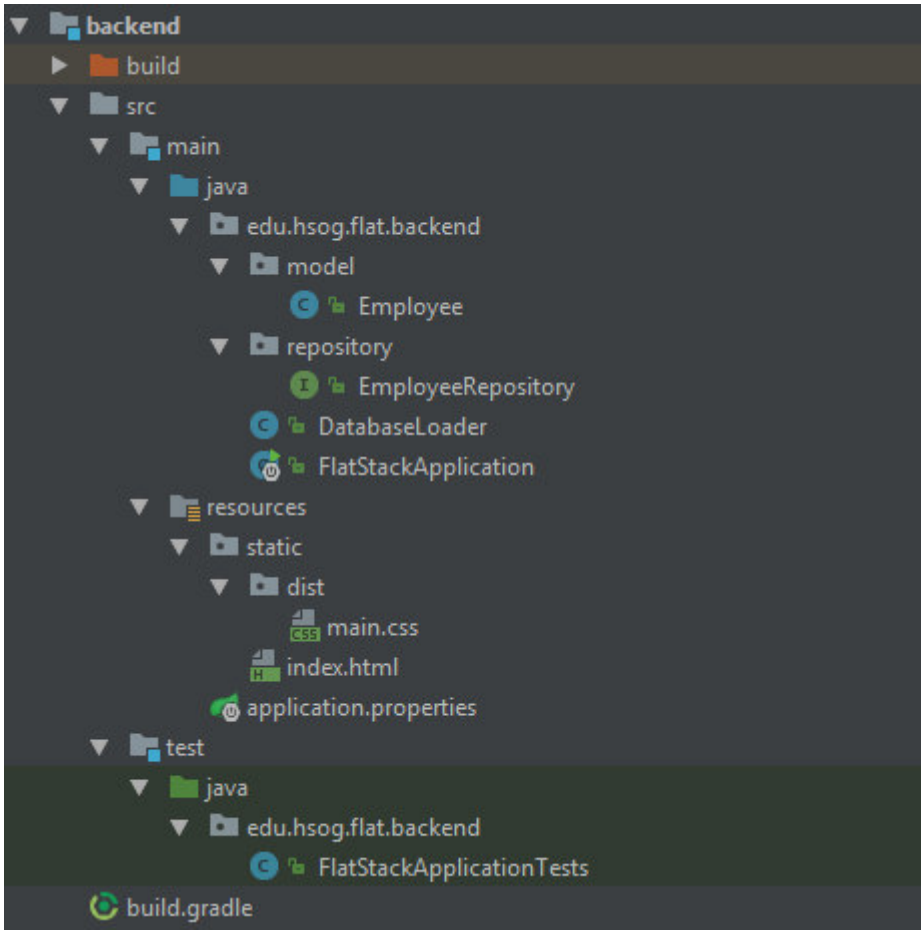


Da beide Untermodule in verschiedenen Programmiersprachen realisiert werden, benötigen sie unabhängige Build-Konfigurationen, weshalb sich die Aufspaltung des Projektes an dieser Stelle bewährt.

Das Hauptprojekt wurde über die „build.gradle“ und „settings.gradle“ Dateien so konfiguriert, dass das Front- und Backend in einer ausführbaren „jar“-Datei gebündelt werden. Im späteren Entwicklungsprozess kann diese Datei auf den Server geladen werden um das Live-System zu realisieren. Mit einer zusätzlichen vorkonfigurierten MySQL-Datenbank reicht diese Datei alleine aus um die komplette Anwendung zu starten.

Backend

Das Backend wird als Spring-Boot Applikation in Java realisiert. Damit im Entwicklungsprozess die Abhängigkeiten von Spring-Boot nicht manuell installiert werden müssen, übernimmt Gradle, an Hand der Konfigurationsdatei für das Backend, diese Aufgabe. Alle weiteren benötigten Java-Bibliotheken wie Lombok, der MySQL-Connector und eben Spring-Boot werden hierzu in der „build.gradle“ Datei im Backend angegeben.



Gradle bringt zur Verwendung einige Konventionen mit sich. Zum Beispiel wird der Java-Code immer unter dem „src/main/java“ Verzeichnis vermutet. In Konfigurationsdateien muss dies also nicht angegeben werden, so lange der Standardpfad verwendet wird. Unter „src/main/resources“ befinden sich zusätzlich statische Ressourcen (z.B. „index.html“ und „main.css“) und eine Konfigurationsdatei für die Spring-Boot Anwendung selbst („application.properties“).

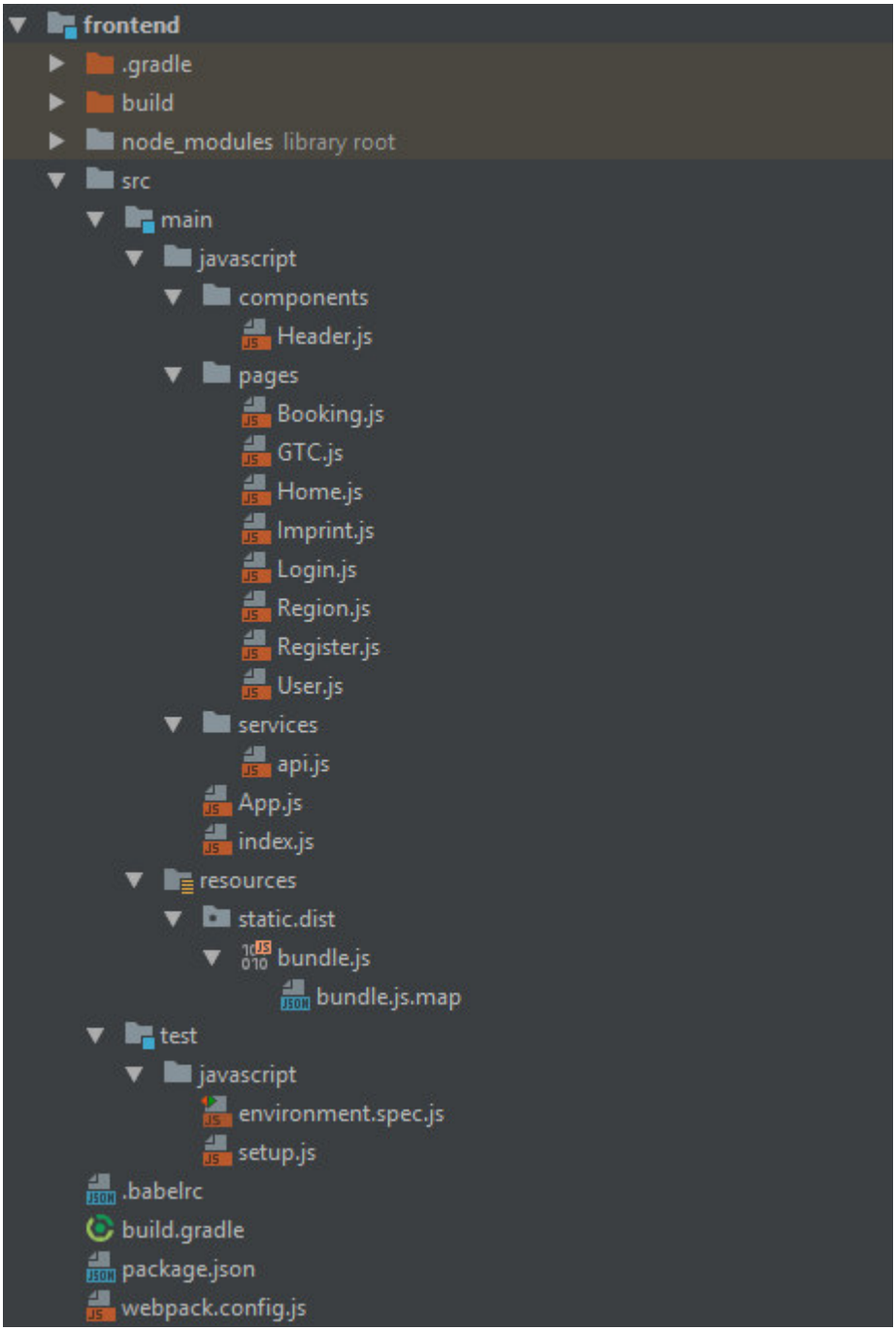
Das „edu.hsog.flat.backend“ – Package beinhaltet den Java-Code des Backends. Die Klasse „FlatStackApplication“ startet das Projekt als Spring-Boot Web-Server. Klassen im „model“ – Package werden mit der MySQL-Datenbank mit Hilfe von Hibernate und JPA verknüpft. So kann über die Interfaces im „repository“-Package auf die MySQL-Dateneinträge zugegriffen werden. In diesen wird angegeben welche Datensätze über welche URL erreichbar sind.
Da Spring-Boot sehr stark auf das Prinzip von „Konvention vor Konfiguration“ setzt, wird der Konfigurationsaufwand sehr gering gehalten. Durch den übermäßigen Einsatz von Java-Annotationen werden die einzelnen Klassen so konfiguriert, dass sie genau ihren Zweck erfüllen.

Der Gebrauch von großen Konfigurationsdateien wird hiermit sehr stark verringert, womit sich der Entwickler auf das Schreiben des Codes alleine konzentrieren kann.

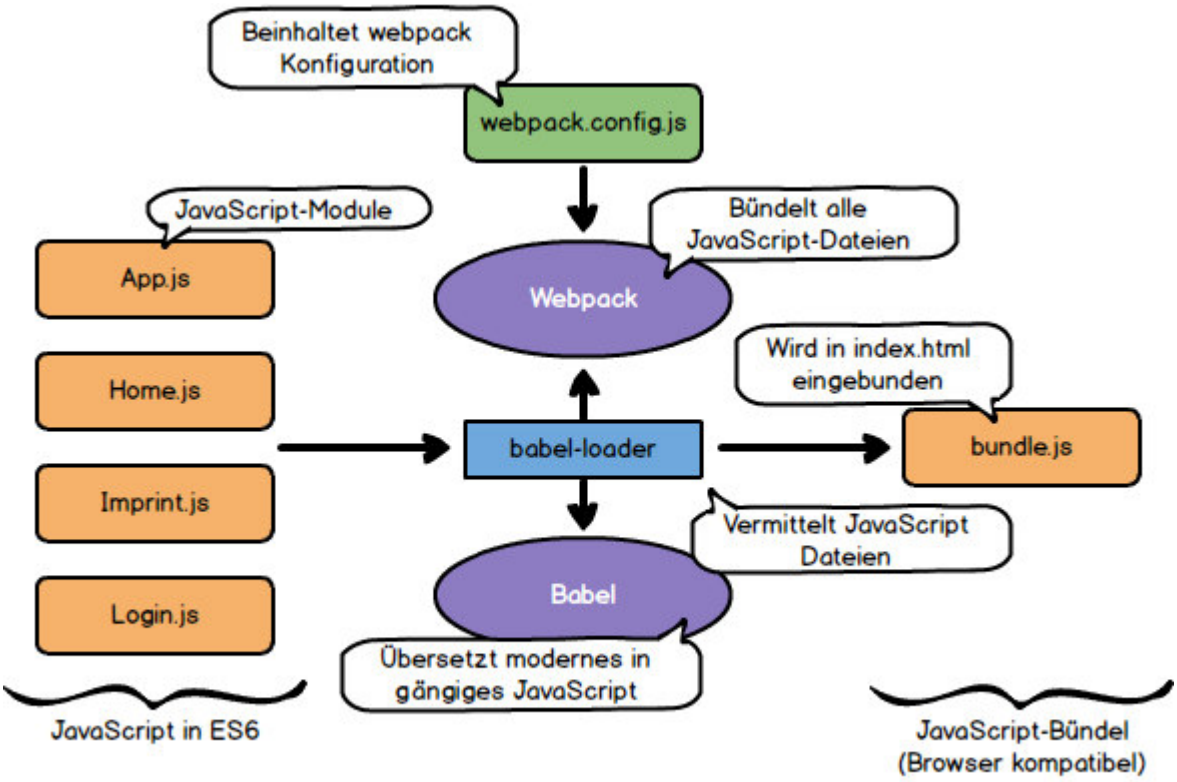
Des Weiteren ist „src/test/java“ das standardmäßige Test-Verzeichnis von Gradle. Unit-Tests in diesem Verzeichnis werden automatisch von Gradle ausgeführt um die Applikation zu testen.

Frontend

Das Frontend ist eine React-Anwendung die in JavaScript realisiert wurde. Gradle kümmert sich in diesem Teil der Applikation um das Installieren der verwendeten JavaScript Module. Außerdem



werden alle Dateien im „src/main/javascript“ Verzeichnis gebündelt, verkleinert und in eine Bündel-Datei geschrieben. Dieses „bundle.js“ befindet sich unter im Verzeichnis „src/main/resources/static/dist/“. Diese Datei wird auf der „index.html“-Seite eingebunden und dient hiermit als Eintrittspunkt der Website-Anwendung.



Die einzelnen Module und Komponenten werden in einem neuen JavaScript Standard „ES6“ geschrieben. Des Weiteren beinhalten React-Komponenten einen weiteren Syntax „JSX“ der es erlaubt HTML in JavaScript zu schreiben. Das Tool Webpack übersetzt jede einzelne Datei mit Hilfe von Babel und packt den, für Browser nun verständlichen Code, in die Bündel-Datei.

Im Entwicklungsprozess gibt es einen Webpack-Entwicklungsserver, der die Bündel-Datei jedes Mal neu erstellt, wenn eine Änderung an den JavaScript-Dateien registriert wurde. Nach dem Erstellen wird der Browser mit dem Bündel neu geladen. Was den Frontend-Entwicklungsprozess an dieser Stelle erheblich beschleunigt.

Das sichtbare Frontend besteht aus vielen kleinen React-Komponenten. Jede Komponente stellt einen individuellen Teil der Benutzeroberfläche dar. Teile die wiederverwendet werden können befinden sich im „components“-Ordner. Der „pages“-Ordner beinhaltet ebenfalls Komponenten, diese werden aber den einzelnen Seiten der Applikation zugeordnet.

Im „services“-Verzeichnis befinden sich JavaScript-Module die in der Applikation eingesetzt werden können. Zum Beispiel wird das „api.js“ Modul dazu verwendet, mit dem Backend / Server zu kommunizieren.

Wie schon im Backend beschrieben, sucht Gradle nach einem festgelegten Ordner mit Unit-Tests. Auch im Frontend-Projekt existiert dieses Verzeichnis. Beim Übersetzen des Projekts werden auch hier die Tests automatisch durchgeführt.

Ablaufplan Buchung

12 Monate vor Termin	**Phase 1**	**Sammeln aller Buchungen, Fairnessprüfung, Feste Zusage; falls keine Buchung eingegangen Übergang in Phase 2**
6 Monate vor Termin	**Phase 2**	**Sammeln der Buchungen nur noch 2-4 Wochen lang, dann Fairnessprüfung und Zusage; falls keine Buchung eingegangen Übergang in Phase 3**
2 Monate vor Termin	**Phase 3**	**„Last-Minute“-Buchungen mit Sofort-Zusage**
Termin		