Mettmann

Studienarbeit

Entwicklung eines Softwareprojektes

Spiele Lizenzschlüssel Preisvergleich

Prüfer:

Dr. Thomas Ströder

Verfasser:

Niklas Hardes
A
A
Matrikelnummer: 000000

Nicolas Groß
Winkelstraße 66
45966 Gladbeck
Matrikelnummer: 101669

Robert Hesselmann
Kronprinzenstraße 83
40217 Düsseldorf
Matrikelnummer: 101672

Studiengang : Angewandte Informatik

Abgabetermin:

15. Januar 2023

Vorbemerkung

In dieser Arbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltextverzeichnis

1 Einleitung

2 UserStories & Use Cases

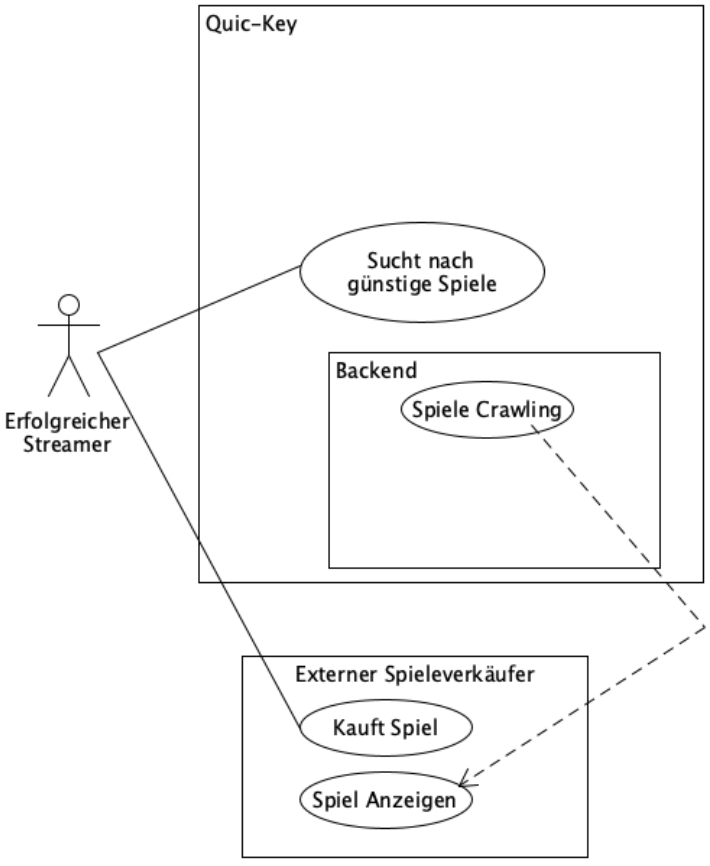
Niklas Hardes

Ich als Gamer möchte wissen, welche Spiele am beliebtesten sind, am meisten gespielt werden und infolgedessen für mich am interessantesten sein können.

Robert Hesselmann

Ein erfolgreicher Streamer auf der Livestreaming Plattform Twitch sucht nach neuen Spielen, die er in seinen Livestreams spielen kann. Um seinen Gewinn zu maximieren sucht er nach einem Spiel zu einem günstigen Preis.

Abbildung 1: Use Case Streamer



Quelle: Eigene Darstellung

Nicolas Groß

Hallo Welt

3 Technologieentscheidungen

Nicolas Groß

Bei der Entwicklung des Frontend haben wir uns für das Ionic-Framework entschieden. Das Ionic-Framework ist ein Open-Source UI Toolkit für die Entwicklung von Hybrid-Apps. Hybrid-Apps sind Anwendungen, welche auf Web-Technologien wie HTML, CSS und JavaScript basieren und die nativen APIs und Funktionen des Betriebssystems nutzen können.[1] Ionic bietet Vorteile wie Performance-Optimierung, Cross-Plattform und die automatische Bereitstellung einer Hellen und Dunklen Ansicht.Ionic bietet zahlreiche anpassbare UI-Komponenten, welche helfen die Strukturierung und Gestaltung der Webanwendung vorzunehmen. Zudem bietet Ionic eine ausführliche Dokumentation und eine große Community, wodurch das Einbauen unbekannter Komponenten schnell und einfach ist. Ionic unterstützt neben dem klassischen JavaScript die Frameworks Angular, React und Vue.[2]

Hierbei fiel unsere Entscheidung aus erfahrungsgründen auf Angular. Angular ist ein von Google entwickeltes auf TypeScript basierendes Open-Source Webframework zur Entwicklung von Mobile und Desktop Webanwendungen. Angular stellt ein Command-Line-Interface bereit, welches Entwicklern bei der Erstellung und Entwicklung der Projekte unterstützt. Die Entwicklung von Single-Page-Apps wird durch Angular Routing ermöglicht, wodurch Webanwendungen schneller auf Benutzereingaben reagieren und die Ladezeiten zwischen einzelnen Seiten stark vermindern. Zudem können Inhalte der Webseite dynamisch geladen und angepasst werden. Angular bietet genau wie Ionic eine ausführliche Dokumentation, Lehrunterlagen und eine aktive Community. Durch Angular ergibt sich die Möglichkeit umfangreiche Bibliotheken einzubinden und deren Ressourcen für die Entwicklung des Projektes zu nutzen.[3]

3.1 Docker (Robert Hesselmann)

Bei der Planung unseres Softwaresystems haben wir uns für Docker als Laufzeitumgebung entschieden. Dadurch kann unsere Software auf allen System laufen die Docker unterstützen. Der Ablauf vom Push eines Commits bis zum Deploy der Software, geht die Software du die Schritte des Bau

[1] Anonymus, n.d. .vgl.
[2] Drifty Co., 2013a, .vgl.
[3] Google LLC., 2016, .vgl.

4 Entwicklungsprinzipien

Nicolas Groß

Modularisierung ist die Unterteilung eines Systems in kleinere Module. Die Modularisierung bietet Eigenschaften, welche bei der Entwicklung und Wartung von Software, Vorteile einbringen. Zu diesen Vorteilen zählen Verständlichkeit, Kombinierbarkeit, Lokalität und die parallele Entwicklung. Die Verständlichkeit einer Software wird durch die Modularisierung verbessert, da einzelne Bestandteile weitgehend unabhängig von anderen Bestandteilen gekapselt und verständlich sind. Die einzelnen Module einer Software können auf unterschiedliche Arten miteinander kombiniert werden, um neue Systeme zusammenzufügen. Dies bietet den Vorteil das Module idealerweise unabhängig vom restlichen System funktionieren und wiederverwendet werden können. Durch die Lokalität zeichnet sich aus, dass Änderungen an einzelnen Modulen keine größeren Änderungen im Gesamtsystem zufolge haben. Ein entscheidender Vorteil für die Entwicklung im Team ist die Möglichkeit zur parallelen Entwicklung. Hierbei können einzelne Teammitglieder an unterschiedlichen Modulen arbeiten, ohne mit den Entwicklungen der anderen Mitglieder zu kollidieren. Das System wird dann zu einem späteren Zeitpunkt zusammengesetzt.[4]

Durch Angular ist eine hohe Modularisierung bereits von Beginn des Projekts gegeben, da die Angular-CLI Komponenten und Services in einzelnen Dateien erzeugen. Auch die Projektstruktur wird von der Angular-CLI angepasst, um Komponenten und Services anzulegen. Die einzelnen Komponenten können durch die Verwendung von Variablen wiederverwendbar gestaltet werden, um diese beliebig zu kombinieren. Die Änderungen an einer Komponente haben hierbei in der Regel keinen Einfluss auf andere Komponenten oder Services, wodurch Änderungen meist leicht umgesetzt werden können und dennoch Projektweit geltend sind.

[4]Schmidauer, 2002, .vgl.

5 Visualisierte Sichten auf das Gesamtsystem

6 Entwurfsmuster

Nicolas Groß

Bei Anwendungen wie in unserem Fall, einer Single-Page-Application (SPA), werden alle Teile auf einmal geladen, um dem Nutzer ein möglichst flüssiges Erlebnis zu bieten. Dadurch werden häufig auch unbenutzte Module der Anwendung geladen. Bei kleineren Projekten stellt dies keine Probleme da, allerdings kann es bei größeren Projekten zu langen Ladezeiten führen. Um dem entgegenzuwirken, bietet sich Lazy Loading. Beim Lazy Loading werden Inhalte einer Anwendung erst geladen, sobald sie benötigt werden. Dadurch werden Wartezeiten beim Start der Anwendung verkürzt. In Angular haben Module die Möglichkeit via Lazy Loading geladen zu werden. Durch den entsprechenden Code im Routing-Module wird das Lazy Loading auf die Module angewendet.

Quelltext 1: Codeausschnitt des Tabs Routingmodule

```
1 {
2   path: 'home',
3   loadChildren: () => import('../tab1/tab1.module').then(m => Tab1
        PageModule),
4   data: {title: 'Home', icon: 'home'}
5 }
```

Wie in diesem Codeausschnitt zu sehen, wird das Modul erst beim Aufruf der Route importiert und geladen. Somit wurde Lazy Loading in Angular-Routing implementiert.[5]

Neben dem Routing bedienen wir uns noch zwei weiter Male den Lazy Loading. Der von Ionic bereitgestellte HTML-Tag „<ion-img>“, welche zum Anzeigen Bildern verwendet wird, lädt Bilder standardmäßig über Lazy Loading. Den deutlichsten unterschied der Ladezeiten weist allerdings die InfinityScroll der „Search“-Seite auf. Bei einer Suche werden die Einträge paketweise der Liste hinzugefügt. Weiter Einträge werden erst der Liste erst hinzugeladen, sobald der Nutzer bis kurz vorm unteren Ende der Liste scrollte.[6]

[5] Arjav Dave, 2021, .vgl.
[6] Drifty Co., 2013b, .vgl.

7 Architekturstile

8 Zusammenfassung

Anhang

Anhangsverzeichnis

Quellenverzeichnis

Internetquellen

Anonymus (n.d.). *Mobile App: Native App vs Hybride App, Web App & PWA*. URL: https://www.brightsolutions.de/blog/native-vs-hybride-vs-web-app/ (besucht am 13. Jan. 2023).

Arjav Dave (2021). *Lazy Loading in Angular – A Beginner's Guide to NgModules*. URL: https://www.freecodecamp.org/news/lazy-loading-in-angular-intro-to-ngmodules (besucht am 13. Jan. 2023).

Drifty Co. (2013a). *Ionic Framework*. URL: https://ionicframework.com/ (besucht am 13. Jan. 2023).

Drifty Co. (2013b). *Ionic Framework*. URL: https://ionicframework.com/docs/api/img (besucht am 13. Jan. 2023).

Google LLC. (2016). *Angular*. URL: https://angular.io/ (besucht am 13. Jan. 2023).

Schmidauer, Helmut (2002). *Modularisierung*. URL: https://ssw.jku.at/Teaching/Lectures/Sem/2002/reports/Schmidauer/ (besucht am 13. Jan. 2023).

Ehrenwörtliche Erklärung

Hiermit erklären wir, dass wir die vorliegende Studienarbeit selbständig angefertigt haben. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut haben wir als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mettmann, 15. Januar 2023

Niklas Hardes

Nicolas Groß

Robert Hesselmann