



Duale Hochschule Baden-Württemberg Mannheim

**Entwicklung eines Prototyps für die Online-
Reservierung von Kino-Tickets**

Studiengang Wirtschaftsinformatik
Software Engineering

Abgabetermin:	13.11.2020
Verfasser:	Patryk Jarzab, Niklas Borg, Noah Hölterhoff, Adrian Albrecht, Leon Flory, Marius Naujoks
Kurs:	MA-WWI19SEA
Dozent	Gregor Tielsch
Studiengangsleiter:	Prof. Dr. Thomas Holey

Inhaltsverzeichnis





1	Pflichtenheft	1
1.1	Zweck und Ziel dieses Dokuments	1
1.2	Projektbezug	1
1.3	Team	1
1.4	Review und Meeting-Protokolle	1
1.5	Konzept und Rahmenbedingungen	2
1.5.1	Ziel des Anbieters	2
1.5.2	Ziele und Nutzen des Anwenders	2
1.5.3	Benutzer / Zielgruppe.....	3
1.5.4	Systemvoraussetzungen.....	3
1.5.5	Anwendungsbereiche	3
1.6	Produktübersicht:.....	4
2	User-Story	5
3	Mockup	8
3.1	Home	8
3.2	Filme.....	9
3.3	Tickets reservieren	10
3.4	Als Gast kaufen	11
3.5	Kauf als Gast abschließen.....	12
3.6	Einloggen	13
3.7	Registrieren	14
3.8	Kauf mit Account abschließen	14
3.9	Einloggen ohne Kauf von Tickets	15
3.10	Mein Profil bearbeiten.....	16
3.11	Vorstellungen anzeigen	17
4	Verwendete Technologien	18
4.1	Backend	18
4.1.1	Java	18
4.1.2	MariaDB.....	19
4.2	Frontend.....	19
4.2.1	JavaScript	19
4.2.2	TypeScript.....	20
4.2.3	NodeJS	20

4.2.4	React	21
4.2.5	JSX/TSX	23
4.2.6	React Semantic UI	23
4.2.7	Axios	23
4.2.8	Webpack.....	24
4.2.9	FTP Server.....	24
5	Backend UML	25
6	GitHub - Online-Reservierung von Kinotickets	30
6.1	Frontend.....	30
6.2	Backend	30
7	Frontend Dokumentation.....	31

Ehrenwörtliche Erklärung

Hiermit erklären wir, dass wir die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen inländischen oder ausländischen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Die vorliegende Fassung entspricht der eingereichten elektronischen Version.

<u>Elztal, 13.11.2020</u>	<u>Marius, Naujoks</u>	
Ort, Datum	Vorname, Name	Unterschrift
<u>Dielheim, 13.11.2020</u>	<u>Patryk, Jarzab</u>	
Ort, Datum	Vorname, Name	Unterschrift
<u>Walldorf, 13.11.2020</u>	<u>Adrian, Albrecht</u>	
Ort, Datum	Vorname, Name	Unterschrift
<u>Mannheim, 13.11.2020</u>	<u>Niklas, Borg</u>	
Ort, Datum	Vorname, Name	Unterschrift
<u>Nußloch, 13.11.2020</u>	<u>Noah, Hölterhoff</u>	
Ort, Datum	Vorname, Name	Unterschrift
<u>Fischbach, 13.11.2020</u>	<u>Leon, Flory</u>	
Ort, Datum	Vorname, Name	Unterschrift

1 Pflichtenheft

1.1 Zweck und Ziel dieses Dokuments

Das vorliegende Pflichtenheft enthält die an das zu entwickelnde Produkt gestellten funktionalen Anforderungen. Es dient als Basis für die Entwicklung eines Prototyps für die Online-Reservierung von Kino-Tickets.

1.2 Projektbezug

Das Projekt wurde im Rahmen einer Fallstudie Systemanalyse an der Dualen Hochschule Baden-Württemberg Mannheim durchgeführt. Ziel des Projekts ist es einen Eindruck über die Arbeit an der Entwicklung von Informationssystemen zu bekommen. Im Laufe des Projekts soll ein Prototyp für die Online-Reservierung von Kino-Tickets entwickelt werden.

1.3 Team

Rolle / Rollen	Name	E-Mail
Projektleiter/ Entwickler	Noah Hölterhoff	noah.hoelterhoff@gmail.com
Entwickler	Patryk Jarzab	patryk.jarzab@web.de
Entwickler	Marius Naujoks	marius.naujoks@outlook.de
Entwickler	Adrian Albrecht	adrian.albrecht@yahoo.com
Entwickler	Leon Flory	leon.flory@t-online.de
Entwickler	Niklas Borg	niklas-borg@web.de

1.4 Review und Meeting-Protokolle

Erste Iteration 24. August bis 28. September

In der ersten Iteration wurde die erste Version des Kino-Ticket-Reservierungs-Systems vorgestellt. Dabei ist es uns gelungen einige Funktionen umzusetzen. Unter anderem ist es nach der ersten Iteration möglich, nach Filmen zu suchen und sich die neuesten

Filme anzeigen zu lassen. Zudem ist es möglich durch das anklicken eines Filmes weitere Details anzuzeigen und einen Trailer anzuschauen. Nachdem man sich für einen Film entschieden hat, kann der Benutzer die Uhrzeit auswählen und die Sitzplätze reservieren. Nach Eingabe seiner Kundendaten kann er derzeit nur die Vorortbezahlung als Bezahlungsmethode auswählen. Zudem kann sich ein Kunde derzeit mithilfe eines User Managements ein Konto anlegen, in welchem er all seine Bestellungen sehen kann.

Zweite Iteration 29. September bis 13. November

In der zweiten Iteration wurden einige Verbesserungen bzw. Erweiterungen an dem Kino-Ticket-System vorgenommen. Der Benutzer bekommt die neusten Filme von Beginn an angezeigt. Wenn er sich für eine Vorstellung entschieden hat, kann er die Uhrzeit für diese aussuchen und die Plätze auswählen. Falls er keinen Account hat, muss er sich einen anlegen und kann anschließend die Buchung mit einer PayPal-Bezahlung abschließen.

1.5 Konzept und Rahmenbedingungen

1.5.1 Ziel des Anbieters

Der Kunde möchte mithilfe des Kino-Ticket-Systems seinen Kunden die Reservierung der Tickets online ermöglichen. Derzeit ist es nur möglich die Tickets Vorort im Kino zu reservieren und zu bezahlen. Mithilfe des neuen Systems soll die Bezahlung der Tickets online über die Webseite möglich sein. Der Kinobesucher soll die aktuelle Filmauswahl angezeigt bekommen und sich selber die Sitzplätze aussuchen können. Im Anschluss soll er die Möglichkeit bekommen, die Tickets für seinen Film zu reservieren und zu kaufen.

1.5.2 Ziele und Nutzen des Anwenders

Der Kinobesucher möchte mithilfe des Kinosystems Filme angezeigt bekommen. Zudem soll es möglich sein, nachdem er sich für einen Film entschieden hat, diesen zu reservieren und die Bezahlungsmöglichkeit auszuwählen.

Der Mitarbeiter eines Kinos soll Vorort die Möglichkeit haben die Reservierung einzusehen, außerdem soll dieser die Möglichkeit haben die Vorstellungen zu bearbeiten.

1.5.3 Benutzer / Zielgruppe

- Mitarbeiter eines Kinos (Admin)
 - Damit ein Mitarbeiter das Programm problemlos bedienen kann, muss dieser bevor das Kino-Ticket-System eingeführt wird, eine Schulung diesbezüglich bekommen.
- Kunden eines Kinos
 - Kunden eines Kinos können jeden Alters sein, um die Nutzung des Kino-Ticket-Reservierungs-Systems für jede Altersgruppe attraktiv zu machen, sollte das System leicht zu bedienen sein.

1.5.4 Systemvoraussetzungen

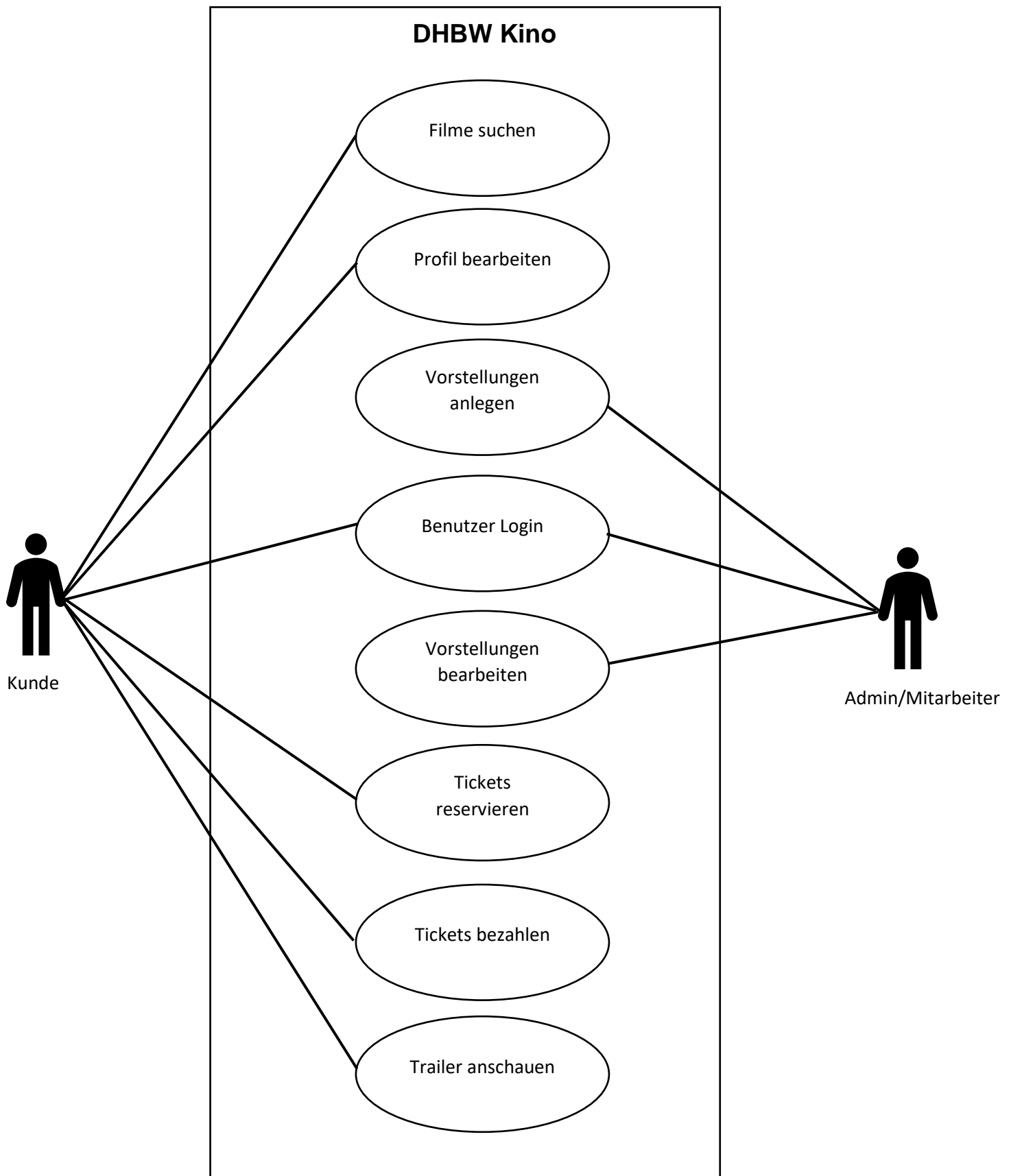
Für die Nutzung des Kino-Ticket-Reservierungs-Systems wird ein Zugang zum Internet vorausgesetzt, um die Webseite aufzurufen.

1.5.5 Anwendungsbereiche

Das System soll im Kino eingesetzt werden. Es soll dem Kunden rund um die Uhr die Reservierung der Kinotickets ermöglichen. Außerdem soll die Oberfläche benutzerfreundlich gestaltet werden, damit der Kinobesucher keine Probleme bei der Reservierung der Tickets hat.

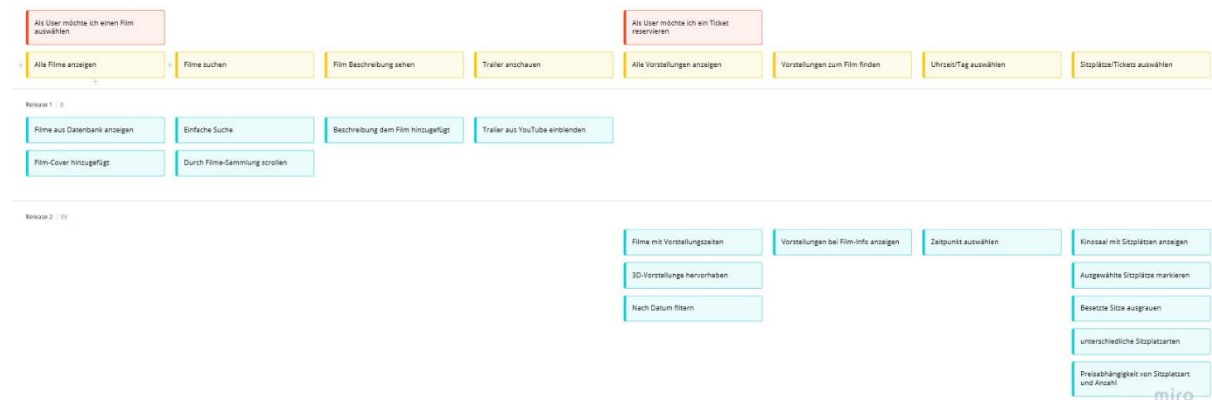
Die Betriebsbedingungen müssen für die Anwendung auf einem Webserver (Internet) spezifiziert werden. Die maximale Anzahl der Benutzer ist prinzipiell unbeschränkt. Der Zugang zur Datenbank, sowie zum Programm selber steht täglich, 24 Stunden zur Verfügung, sofern der Server dies zulässt.

1.6 Produktübersicht:

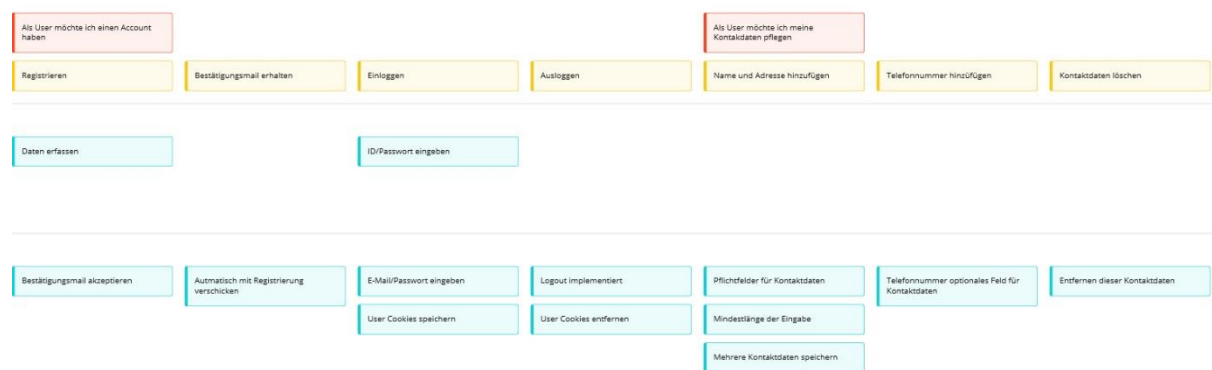


2 User-Story

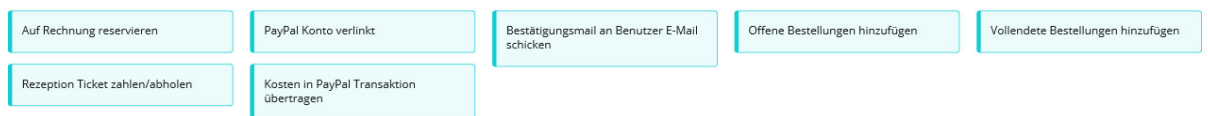
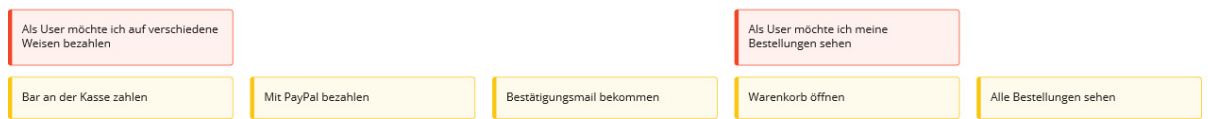
User Story Map



User_Story_01



User_Story_02



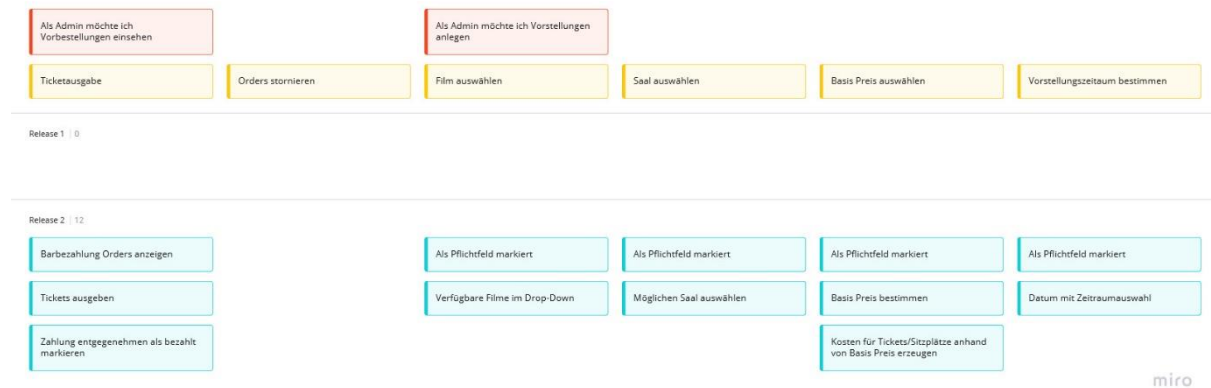
User_Story_03



miro

User_Story_04

User Story Map



miro

User_Story_Admin

3 Mockup

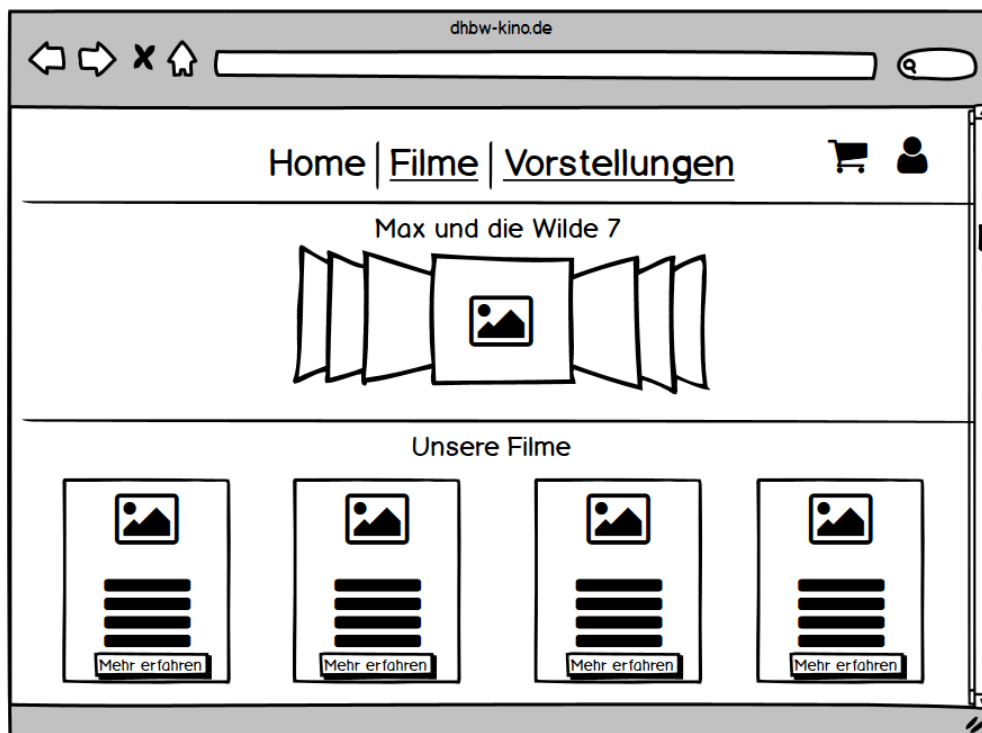


Abbildung 1

3.1 Home

Nachdem der Benutzer die Internetseite <https://dhw-kino.de/> aufgerufen hat. Gelangt dieser auf die Hauptseite (siehe Abbildung 1). Auf dieser kann er mithilfe von drei Buttons „Home“, „Filme“ und „Vorstellungen“ zwischen drei Seiten wechseln. Auf der Hauptseite (Home) bekommt der Kinobesucher auf den ersten Blick die angesagtesten Filme im Wechsel angezeigt. Zudem bekommt der Benutzer durch das runter scrollen weitere Filme angezeigt. Die Filme werden dem Kinobesucher mit einer kurzen Beschreibung angezeigt. Wenn er mehr zu dem jeweiligen Film erfahren möchte, kann er auf den Button „Mehr erfahren“ unter dem Film klicken. Danach gelangt der Kinobesucher auf die zweite Seite (siehe Abbildung 2).

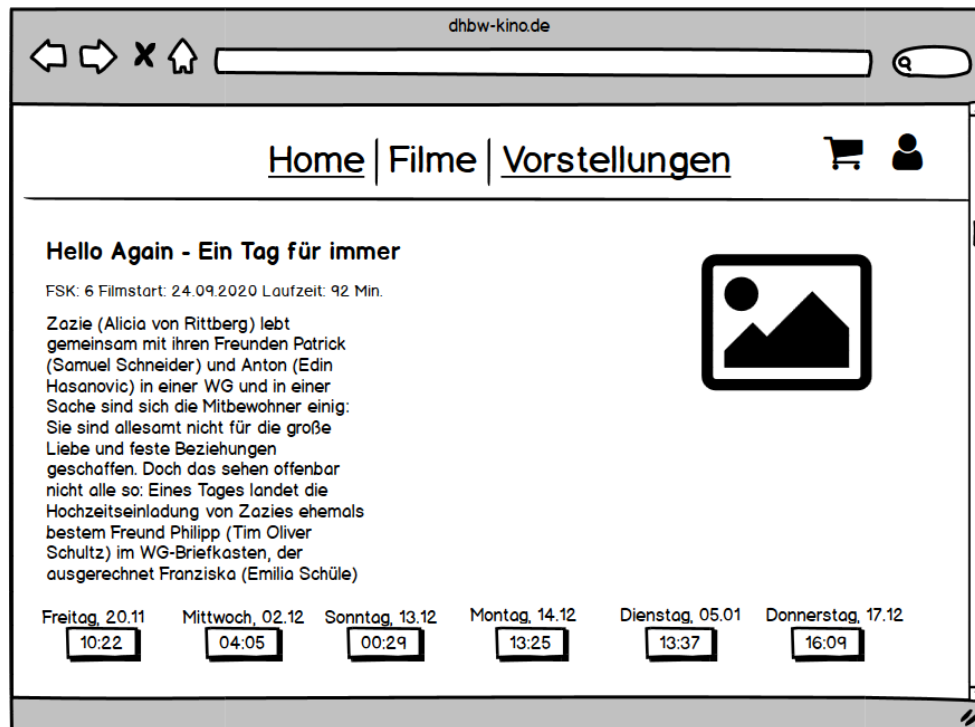


Abbildung 2

3.2 Filme

Durch das Betätigen des Buttons „Mehr erfahren“ gelangt der Kinobesucher zu dem jeweiligen Film (siehe Abbildung 2). Auf der Seite Filme werden dem Kinobesucher weitere Filmdetails angezeigt. Unter anderem kann er den Titel sowie das Cover des Filmes sehen. Unter dem Titel des Filmes wird ihm die komplette Filmbeschreibung angezeigt. Außerdem kann er sich direkt einen Trailer zu diesem Film anschauen. Zudem kann der Benutzer sehen, wann die Vorstellung des jeweiligen Filmes stattfindet. In dem unteren Bereich der Abbildung 2 bekommt der Benutzer das jeweilige Datum sowie die Uhrzeit angezeigt, somit kann dieser sehen, wann die Vorstellung stattfindet. Wenn der Benutzer sich für ein Datum und eine Uhrzeit entschieden hat, klickt er auf diese und gelangt zur nächsten Seite (siehe Abbildung 3).



Abbildung 3

3.3 Tickets reservieren

In Abbildung 3 sieht der Benutzer durch den Text „Tickets reservieren“ in der oberen Hälfte der Abbildung, dass er sich bei der Reservierung der Tickets befindet. In der oberen Hälfte der Abbildung befindet sich der Filmtitel sowie weitere Details zu diesem Film. Auf der rechten Seite der Abbildung wird das Cover des Films angezeigt. In der unteren Hälfte der Abbildung kann der Benutzer sich seine Sitzplätze aussuchen. Dafür muss er ein Personen-Icon auswählen. Dadurch markiert er ein oder mehrere Sitzplätze. Nachdem der Benutzer seine Sitzplätze ausgewählt hat, muss dieser auf den Button „Tickets kaufen“ klicken. Danach gelangt der Benutzer zur nächsten Seite (siehe Abbildung 4).



Abbildung 4

3.4 Als Gast kaufen

In Abbildung 4 muss sich der Benutzer entscheiden wie er fortfahren möchte. Dieser kann sich Einloggen und mit seinem Konto die Tickets kaufen oder als Gast die Tickets kaufen. Wenn der Benutzer sich entscheidet als Gast die Tickets zu kaufen, gelangt dieser zur nächsten Seite (siehe Abbildung 5).

dhw-kino.de

Home | Filme | Vorstellungen

Bitte füllen Sie das Formular aus um den Kauf abzuschließen

Name Anrede

Straße Hausnummer Postleitzahl Stadt

Vorwahl Telefonnummer

Ihre Bestellung:
Loge * 1 = 11,5€
Gesamtpreis: 11,5€ inkl. MwSt.

Bezahlung wählen:
☐ Barzahlung
☐ Kreditkarte
☐ PayPal
Bezahlen

Abbildung 5

3.5 Kauf als Gast abschließen

In Abbildung 5 muss der Benutzer seine persönlichen Daten eingeben. Nachdem der Benutzer seinen Namen, die Adresse sowie seine Telefonnummer eingegeben hat, sieht er seine Bestelldetails in der linken unteren Ecke. Zum Schluss muss der Benutzer seine Zahlungsart auswählen und auf den Button „Bezahlen“ klicken.

Wenn der Benutzer jedoch ein Konto besitzt und sich Einloggen möchte, kann er auf Login (siehe Abbildung 4) klicken. Danach gelangt dieser zur Login-Seite (siehe Abbildung 6).

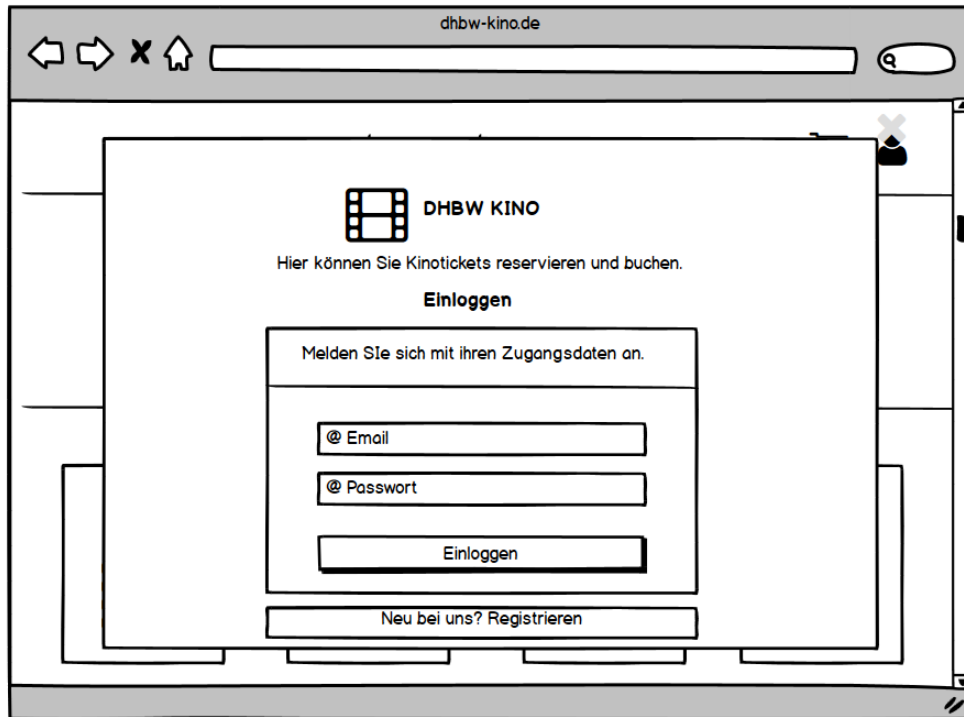


Abbildung 6

3.6 Einloggen

Jetzt muss der Kinobesucher seine Anmeldedaten eingeben. Diese setzen sich aus der E-Mail und dem Passwort zusammen. Wenn der Kinobesucher noch keinen Account hat, kann sich dieser registrieren, indem er auf den Button „Registrieren“ klickt. Danach gelangt dieser zur Registrierungsseite (siehe Abbildung 7).

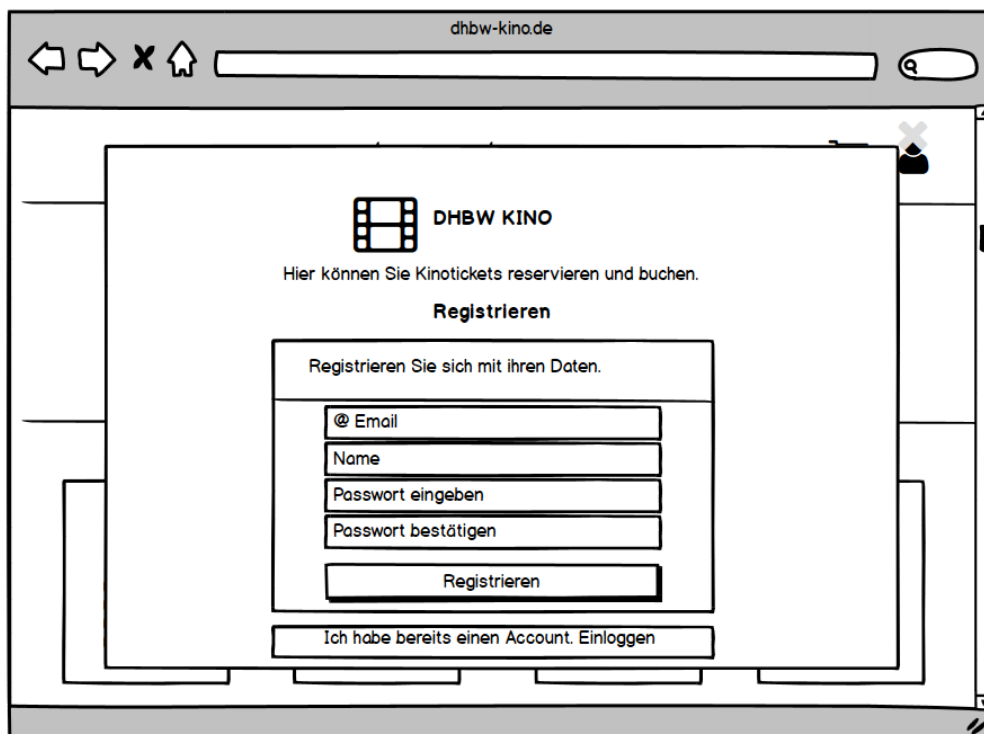


Abbildung 7

3.7 Registrieren

Durch die Eingabe seiner E-Mail, Namen und Passwort kann der Kinobesucher einen Account erstellen. Nachdem er seine gesamten Daten eingegeben hat und den Button „Registrieren“ betätigt hat, bekommt er eine E-Mail, in der er seinen Account bestätigen muss. Nachdem der Benutzer sich einen Account erstellt hat, kann er sich Einloggen (siehe Abbildung 6). Sobald er sich beim Bestellvorgang (Abbildung 4) einloggt, gelangt er zur nächsten Seite (siehe Abbildung 8).

dhbw-kino.de

Home | Filme | Vorstellungen

Bitte füllen Sie das Formular aus um den Kauf abzuschließen

Adresse auswählen:

Adresse hinzufügen Berlinerstraße, 1, 5484 Michendorf

Ihre Bestellung:

X-Men: New Mutants
Loge * 1 = 12.5€

Ihre Bestellung:

Loge * 1 = 11.5€
Gesamtpreis: 11.5€ inkl. MwSt.

Bezahlung wählen:

☐ Barzahlung
☐ Kreditkarte
☐ PayPal

Bezahlen

Abbildung 8

3.8 Kauf mit Account abschließen

In Abbildung 8 muss der Kunde eine seiner hinterlegten Adressen auswählen oder eine neue Adresse hinzufügen, um die Bezahlung fortzuführen. Eine neue Adresse wird hinzugefügt, indem der Benutzer auf den Button „Adresse hinzufügen“ klickt. Hiernach kann er seine Zahlart auswählen und auf den Button „Bezahlen“ klicken.

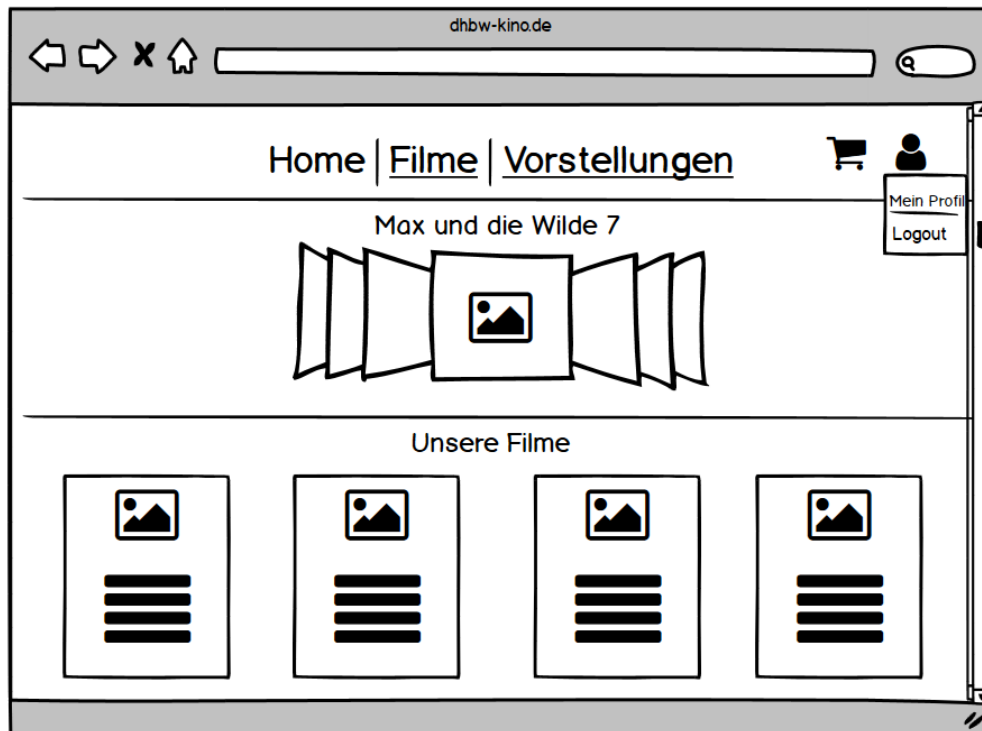


Abbildung 9

3.9 Einloggen ohne Kauf von Tickets

Nachdem der Benutzer sich eingeloggt hat, kann er auf der Home Seite (siehe Abbildung 1), indem er auf den Personen-Button in der rechten oberen Ecke klickt (siehe Abbildung 9), sich ausloggen oder sein Profil aufrufen.

Durch das Betätigen des Buttons „Mein Profil“ (siehe Abbildung 9), gelangt der Kinobesucher zur nächsten Seite (siehe Abbildung 10).

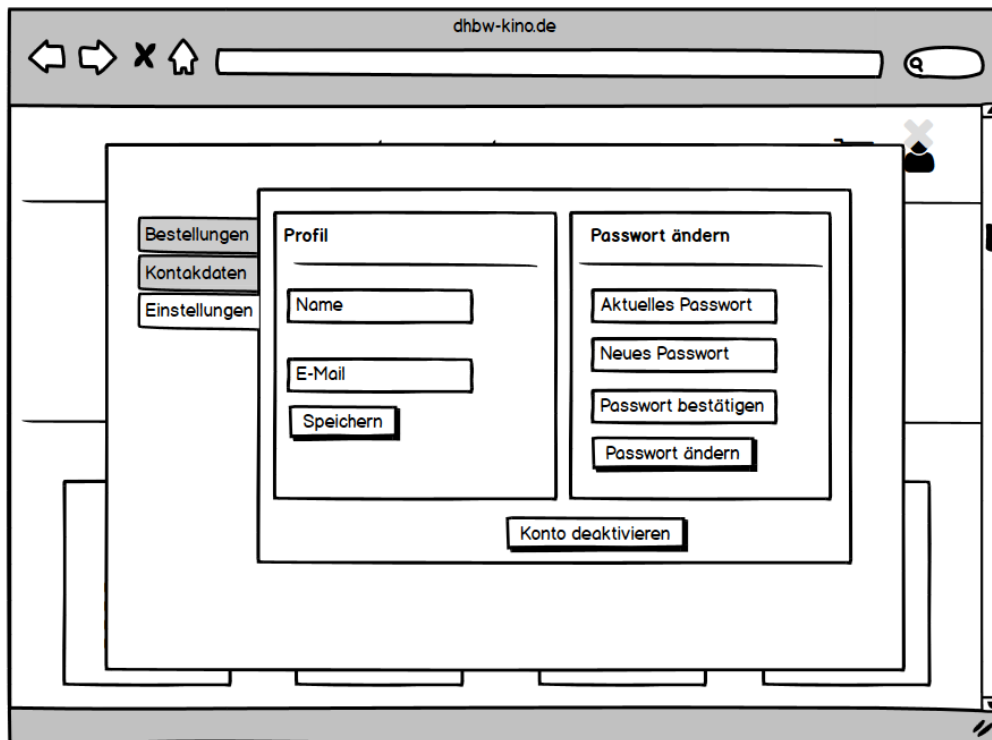


Abbildung 10

3.10 Mein Profil bearbeiten

Unter seinem Profil bekommt der Kinobesucher eine Übersicht seiner Bestellungen. Außerdem kann der Kinobesucher seinen Namen, seine E-Mail sowie das Passwort ändern. Durch das Betätigen des „Kreuz“ Buttons in der rechten oberen Ecke gelangt der Kinobesucher zur Hauptseite (siehe Abbildung 1).

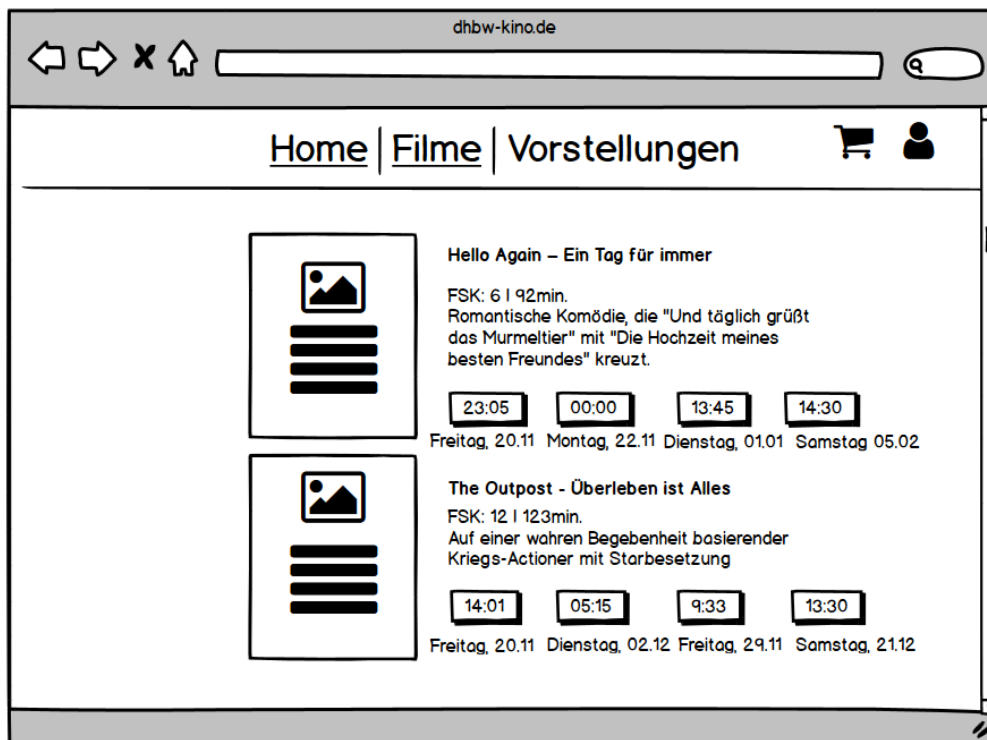


Abbildung 11

3.11 Vorstellungen anzeigen

In Abbildung 11 kann der Benutzer alle Vorstellungen anschauen. Zu jeder Vorstellung wird ein Cover und der Name des Films angezeigt. Wenn sich der Benutzer für einen Film entschieden hat, kann er durch das anklicken eines der Uhrzeiten Tickets für diesen Film buchen. Danach gelangt er zu Abbildung 3.

4 Verwendete Technologien

4.1 Backend

4.1.1 Java

Java ist eine plattformunabhängige, objektorientierte Programmiersprache die 1995 von Sun Microsystems veröffentlicht wurde. Java ist eine besondere Programmiersprache, die sich an der Syntax der Programmiersprachen C und C++ anlehnt. Bei der Entwicklung von Java wurde von Beginn an genauestens darauf geachtet, dass bestimmte schwere Programmierfehler erst gar nicht auftreten können. So gibt es in Java keine Pointer und somit kann der Programmierer nicht aus Versehen auf die falsche Speicheradresse zugreifen. Viele anspruchsvolle Funktionen, wie Ausnahmebehandlung, Nebenläufigkeit und grafische Benutzeroberflächen wurden in Java integriert. Am besten eignet sich diese Sprache für den Backendbereich in großen Webanwendungen. In Java sind alle Eigenschaften moderner objektorientierter Programmiersprachen enthalten. Folgende Merkmale zeichnen die Programmiersprache Java besonders aus:

- **Ähnliche Ausdrücke:** Der Umstieg von C und C++ ist leicht möglich, da die gleichen Ausdrücke verwendet werden. Zum Beispiel if-, break- und continue-Anweisung oder for-, while und do-Schleife.
- **Datentypen:** Die meisten populären Datentypen sind in Java integriert. Ein logischer Datentyp ist mit boolean bereits vorhanden.
- **Arrays:** In Java werden Arrays als Objekte interpretiert und Zugriffe auf Arrays werden zur Laufzeit auf Korrektheit überprüft. Durch Verschachtelung einfacher Arrays können mehrdimensionale Strukturen nachgebildet werden.
- **Keine GoTo-Anweisung:** Damit wurde eine häufige Fehlerquelle beseitigt.
- **Definition von Klassen:** In Java werden aus Klassen Objekte erzeugt. Spezielle Konstruktoren initialisieren die Objekte und sind für die Zerstörung der Objekte verantwortlich.
- **Sichtbarkeit:** Mit den Schlüsselwörtern private, protected und public wird die Sichtbarkeit von Elementen eingeschränkt.

- **Speicherverwaltung:** Java besitzt ein automatisches Speichermanagement, den sogenannten Garbage-Collector. Ungenutzter Speicher wird bereits zur Laufzeit wieder freigegeben.
- **Vererbung:** Unter Java ist es möglich Eigenschaften von Klassen zu vererben.
- **Ausnahmebehandlung:** Um Laufzeitfehler kontrolliert behandeln zu können, wurden in Java Exceptions-Objekte eingeführt. Durch sie ist es möglich Laufzeitfehler abzufangen und zu behandeln.¹

Aufgrund dieser Gründe verwenden wir Java in unserer Backend-Entwicklung für das Kinoticket-Reservierungs-System.

4.1.2 MariaDB

MariaDB ist ein freies, relationales Open-Source-Datenbankmanagementsystem, das durch eine Abspaltung aus MySQL entstanden ist. Das Projekt wurde von MySQLs früherem Hauptentwickler Michael Widenius initiiert, der auch die Storage-Engine Aria entwickelte, auf welcher MariaDB ursprünglich aufbaute. MariaDB ist die Datenbank, die wir für das Speichern all unserer Daten verwenden, die durch die Kino-Ticket-Reservierung entstehen oder benötigt werden.²

4.2 Frontend

4.2.1 JavaScript

JavaScript ist eine Skriptsprache, die 1995 von Netscape entwickelt wurde und üblicherweise in Webseiten Anwendung findet. Durch JavaScript können sich diese dynamisch aufbauen und sich dem Benutzer anpassen. Das heißt, dass direkt auf das DOM zugegriffen und dieses manipuliert werden kann. Diese Anwendungen nennt man clientseitig, da hier keine Kommunikation mit einem Server stattfindet. Aber auch der serverseitige Einsatz findet statt. Für diesen Einsatz ist vor allem das Framework NodeJS bekannt, auf das im Folgenden noch eingegangen wird. Bekannte Konzepte wie die Klassen, Vererbung oder Typsicherung sind in JavaScript nicht vorhanden.

¹<https://www.programmierenlernenhq.de/was-ist-java-einfuehrung-in-die-programmiersprache-java/>

²<https://mariadb.com/de/>

Diese und weitere Konzepte sind jedoch in der Programmiersprache TypeScript realisiert.³

4.2.2 TypeScript

TypeScript ist eine Programmiersprache, die 2012 veröffentlicht und von Microsoft entwickelt wurde. Die Programmiersprache basiert auf dem ECMAScript-6-Standard. TypeScript ist zu JavaScript sehr ähnlich, wurde allerdings mit bekannten Konzepten, wie Klassen und Vererbung oder der Typsicherung erweitert. So können Fehler oder Ausnahmen schneller gefunden und vorgebeugt werden. Wichtig ist, dass TypeScript, bevor es von dem Browser ausgeführt wird, in JavaScript umgewandelt wird. So ist es auch möglich, JavaScript in TypeScript umzuwandeln. Das wiederum macht es möglich, reines JavaScript in eine TypeScript Datei zu schreiben. Der Vorteil ist, dass jegliche JavaScript Frameworks und Libraries problemlos auch mit TypeScript verwendet werden können.⁴

4.2.3 NodeJS

NodeJS ist eine serverseitige Open-Source-JavaScript-Laufzeitumgebung. Es herrscht ein besonderer Fokus auf der Performance, sodass sich damit hoch performante Webanwendungen entwickeln und ausführen lassen. Des Weiteren sind bereits einige JavaScript-Module in NodeJS enthalten, wobei diese meist nicht ausreichend sind. In der Regel kommen auf diese Module ergänzende weitere Module, wie zum Beispiel die Frontend-Bibliothek React oder der Modul-Bundler Webpack. Diese beiden Module werden im Folgenden näher beschrieben. In unserem Fall nutzen wir NodeJS lediglich im Entwicklungsraum als Development Server.⁵

³Freeman, Adam (2019). Essential TypeScript: From Beginner to Pro. 1. Auflage. New York City: Apress. ISBN: 978-1484249789.

⁴Freeman, Adam (2019). Essential TypeScript: From Beginner to Pro. 1. Auflage. New York City: Apress. ISBN: 978-1484249789.

⁵Springer, Sebastian (2018). Node.js: Das umfassende Handbuch. Serverseitige Web-Applikationen mit JavaScript entwickeln. 3. Auflage. Bonn: Rheinwerk und Rheinwerk Computing. ISBN: 978-3836262552.

4.2.4 React

React ist eine JavaScript-Software Bibliothek, die anfänglich 2011 von Facebook entwickelt wurde. Mit dieser in den letzten Jahren immer populärer werdenden Bibliothek lassen sich Benutzeroberflächen vergleichsweise einfach entwickeln. Ein zentrales Konzept von React ist die Komponenten-bezogene Architektur, welche den Code nachvollziehbar und wartbar macht. Jede einzelne Komponente ist autark und wiederverwendbar. Jede Komponente kann andere Komponenten importieren bzw. sich selbst exportieren. Es gibt zwei verschiedene Weisen eine Komponente aufzubauen: Entweder kann sie aus einer Funktion bestehen oder sie ist klassenorientiert und erbt aus der abstrakten Klasse `React.Component`. Weiter verbreitet sind hier die klassenorientierten Komponenten, die von Facebook bzw. vom React-Entwicklerteam auch empfohlen werden. Technisch gesehen vereint jede Komponente Struktur (HTML), Aussehen (CSS), sowie Funktionalität (JavaScript) und verfügt stets über eine `render`-Funktion. Durch die sogenannte JavaScript beziehungsweise TypeScript Syntax Extension (JSX/TSX), auf die im nächsten Absatz näher eingegangen wird, ist es möglich, JavaScript Code mit HTML zu vermischen.

Zentrale Bausteine von React sind die Props, der State sowie die Lifecycle-Methoden.

Props sind die Kurzform von Properties (zu Deutsch Eigenschaften) und bieten die Möglichkeit, Daten an eine andere Komponente weiter zu leiten. Für die Sub-Komponente ist es nicht möglich, die Props zu verändern. Mit den Props ist es möglich, auf eine andere Seite bzw. URL weiter zu leiten. Änderungen von außen (durch die Props), also von einer anderen Komponente, bewirken in der Regel ein erneutes Rendern (Re-Rendering).

Neben den Props ist das Konzept des States (zu Deutsch: Zustand) von zentraler Bedeutung. Der State stellt ein einfaches JavaScript-Objekt, indem die einzelnen Attribute als eine Art von Variablen dienen, dar. Der State ist ausschließlich innerhalb der Komponente, in der er definiert wurde, sichtbar und kann auch nur von ihr modifiziert werden. In der Regel passiert eine Änderung des States durch die vordefinierte `setState`-Funktion. Auch eine Änderung am State bewirkt in der Regel ein Re-Rendern der Komponente.

Zusätzlich besitzt jede Komponente sogenannte „Lifecycle-Methoden“, die überschrieben werden können und die jeweils zu einem sehr bestimmten Zeitpunkt des Programms aufgerufen werden. Drei äußerst hilfreiche solcher Methoden sind folgende:

- **componentDidMount():** Diese Methode wird aufgerufen, sobald die Komponente geladen (mount) wurde. In dieser Methode bietet es sich beispielsweise an, für die Komponente relevante Daten zu laden und abzuspeichern.
- **componentDidUpdate():** Diese Methode wird aufgerufen, sobald die sich die Komponente in irgendeiner Art und Weise aktualisiert (update).
- **componentWillUnmount():** Diese Methode wird kurz bevor die Komponente nicht mehr genutzt wird (unmount) aufgerufen. Gegebenenfalls ist es hier sinnvoll, Cleanup wie beispielsweise das Beenden von Netzwerkanfragen oder von Timern zu betreiben.⁶

React zeichnet sich zudem mit einer weiteren Eigenschaft aus, die eine hervorragende Rendering-Performance ermöglicht. Operationen werden nicht auf dem tatsächlichen Browser-DOM ausgeführt, sondern auf einem virtuellen DOM. Hintergedanke dieses virtuellen DOMs ist es, dass React bei einer Änderung eine virtuelle Baumstruktur erstellt. Diese wird mit der vorherigen Baumstruktur verglichen, woraufhin ausschließlich die Unterschiede zu Operationen gebündelt und an das Browser-DOM weitergeleitet werden. Es kommt zu einem kleineren Aktualisierungsvorgang, der Basis für eine hohe Performance schafft.⁷

⁶<https://reactjs.org/docs/getting-started.html/>

⁷Zeigermann, Oliver und Nils Hartmann (2016). React: Die praktische Einführung in React, React Router und Redux. 1. Auflage. Heidelberg: Dpunkt.verlag. ISBN: 978-3864903274.

4.2.5 JSX/TSX

JSX wird bei der Verwendung von React empfohlen, ist aber optional. Üblicherweise wird JavaScript in Form von mehreren Skripten in HTML eingebettet. JSX funktioniert allerdings genau andersrum. So werden JavaScript Dateien erzeugt, die HTML einbetten. Normalerweise müsste jedes HTML Element mit `React.createElement` erzeugt werden. JSX macht das hinfällig und ermöglicht HTML Elemente direkt im JavaScript Code zu verwenden. Dadurch ist die Code Gestaltung lesbarer und übersichtlicher. Um nun innerhalb dieses Bereiches auf JavaScript Variablen, States oder Props zugreifen zu können, müssen diese lediglich in geschweifte Klammern gesetzt werden. So wird nämlich angezeigt, dass keine Übersetzung des geklammerten Codes stattfinden soll. Die Umwandlung des Codes übernimmt in diesem Fall ein sogenannter Transpiler. Dieses ist allgemein dafür zuständig, eine Programmiersprache in eine andere zu kompilieren.⁸

4.2.6 React Semantic UI

React Semantic UI ist eine React Library, die vorgestellten Komponenten zur Verfügung stellt. Sie ermöglicht eine übersichtlichere und einfachere Code-Gestaltung, da kleinere Komponenten wie Buttons oder Tabellen importiert werden können. Zur Verfügung steht eine Vielzahl an unterschiedlichen Komponenten, sodass bei der Oberflächenentwicklung nur auf wenige weitere Libraries zugegriffen werden muss. Mithilfe von Semantic UI ist es zusätzlich möglich seine Benutzeroberfläche responsiv und einheitlich zu gestalten.⁹

4.2.7 Axios

Axios ist eine sehr beliebte JavaScript Library mit welcher HTTP Requests ausgeführt werden können, die sowohl für den Browser als auch für NodeJS Plattformen funktionieren. Axios unterstützt alle modernen Browser. Es ist promise-based und erlaubt es uns `async/await` Code zu schreiben um XHR Requests auf sehr einfache Weise auszuführen.¹⁰

⁸<https://de.reactjs.org/docs/introducing-jsx.html>

⁹<https://react.semantic-ui.com/usage>

¹⁰<https://blog.logrocket.com/how-to-make-http-requests-like-a-pro-with-axios/>

4.2.8 Webpack

Webpack hat die Aufgabe, die Abhängigkeiten von Webanwendungen zu analysieren und aus den benötigten Modulen und Erweiterungen eine JavaScript Datei zu erstellen, die der Browser ausführen kann. Hier spricht man von einem Bundle. Von Vorteil ist, dass Webpack auch HTML, CSS und Bilddateien konvertieren kann. Hierbei gilt allerdings, dass bestimmte Loader im Projekt enthalten sein müssen. Um alle Module richtig einzubinden, gibt es eine Konfigurationsdatei, namens `webpack.config.js`, die beliebig angepasst werden kann.

Webpack wird eingesetzt, wenn Webanwendungen zu groß und unübersichtlich werden. Um die Anwendungen gut zu strukturieren, kann man den Code in verschiedene, abhängige Module einteilen. Da der Browser diese Modulsysteme nicht unterstützt, muss hier Webpack verwendet werden, um die benötigten Module, bzw. Dateien zusammenzuführen.¹¹

4.2.9 FTP Server

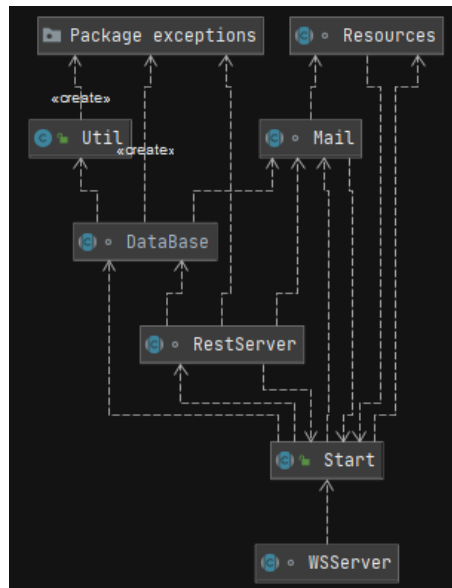
FTP steht für File Transfer Protocol, was so viel bedeutet wie Dateiübertragungsprotokoll. Es handelt sich also um ein Kommunikationsprotokoll. Ein Kommunikationsprotokoll regelt die Bedingungen, unter denen Dateien zwischen mehreren Computern übertragen werden. Unter den Kommunikationsprotokollen gehört FTP zu der Gruppe der Netzwerkprotokolle. Mit dem FTP wird also der Austausch von Daten zwischen mehreren Computern in einem Netzwerk geregelt. Sie können von einem FTP-Server Dateien herunterladen, Dateien hochladen und auch die Ordnerstruktur auf dem Server ansehen und verändern. Meist wird FTP verwendet, um Dateien auf einem HTTP-Server, dem einfachen Webserver, zu organisieren.¹²

¹¹Zammetti, Frank (2020). Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker. 1. Auflage. New York: Apress. ISBN: 978-1484257371.

¹²https://praxistipps.chip.de/was-ist-ftp-und-wie-funktioniert-s-einfach-erklart_41170

5 Backend UML

Es gibt 6 Klassen die alle miteinander in Beziehung stehen.



Backend UML-Klassendiagramm

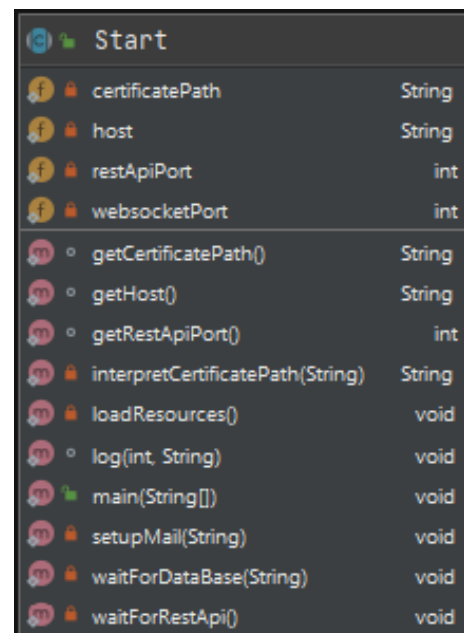
Klasse Start: Einstieg in das Programm

Methoden:

- `getCertificatePath` → Fragt den Path des Zertifikates ab.
- `getHost` → Fragt den Host ab, auf dem der Server läuft.
- `getRestApiPort` → Fragt den Port ab, auf der die REST API arbeitet.
- `interpretCertificatePath` → Interpretiert den mitgegeben Pfad des Zertifikates. Dieser kann entweder direkt den Pfad zu dem Zertifikat enthalten, oder ein Pfad zu einer "last_nginx.conf"-Datei sein.

Falls es die "last_nginx.conf"-Datei ist, wird diese interpretiert und der Speicherort des Zertifikates aus dieser ausgelsen.

- `loadResources` → Lädt alle Ressourcen, die während der Laufzeit benötigt werden.
- `log` → Loggt einen Text in der Konsole (mit farbigem Prefix)
- `main` → Einstieg in das Programm



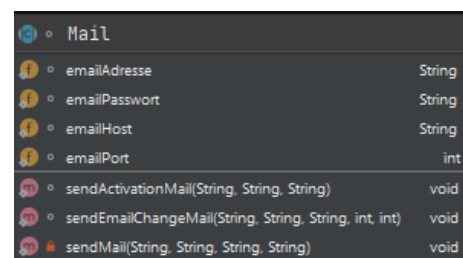
Klasse Start

- `setupMail` → Interpretiert einen EMail-Konfigurations-String. Dieser muss folgendes Format haben: HOST:PORT:EMAILADRESSE:PASSWORT
- `waitForDataBase` → Wartet darauf, dass eine Verbindung zu der Datenbank hergestellt wird. Diese Methode probiert dabei alle 10 Sekunden erneut, die Verbindung herzustellen.
- `waitForRestApi` → Wartet darauf, dass die REST API online geht. Diese Methode probiert dies dabei alle 10 Sekunden erneut.

Die **Klasse Mail** ist für alle Interaktionen mit Emails (senden, empfangen, etc.) zuständig.

Methoden:

- `sendActivationMail` → Sendet eine Aktivierungs-Email an eine Person
- `sendMail` → Sendet eine verschlüsselte Email über den angegebenen SMTP-Server.



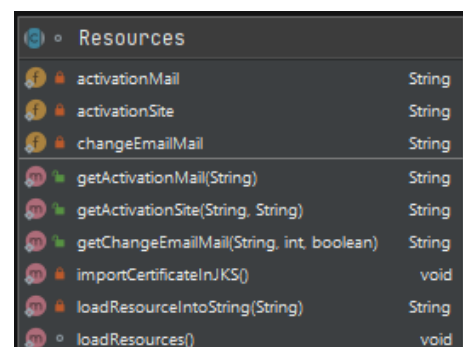
Mail	
emailAdresse	String
emailPasswort	String
emailHost	String
emailPort	int
sendActivationMail(String, String, String)	void
sendEmailChangeMail(String, String, String, int, int)	void
sendMail(String, String, String, String)	void

Klasse Mail

Die **Klasse Resources** ist für die Verwaltung von Ressourcen (Mails, Webseiten, Bildern, etc.) zuständig.

Methoden:

- `getActivationMail` → Gibt den HTML-Code der Aktivierungsmail zurück. Dabei wird ein Platzhalter für den Aktivierungslink durch den mitgegebenen ersetzt.
- `getActivationSite` → Gibt den HTML-Code für die Aktivierungswebseite zurück. Dabei werden die Platzhalter für Title und Untertitel durch die mitgegeben Attribute ersetzt.
- `importCertificateInJKS` → Liest das Zertifikat von dem Pfad ein, dass durch
- `loadResourceIntoString` → Liest eine Datei aus den Ressourcen und kopiert Sie in einen String
- `loadResources` → Lädt alle Ressourcen, die verwendet werden in diese Klasse.



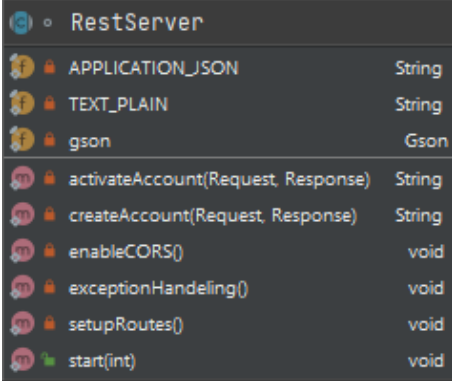
Resources	
activationMail	String
activationSite	String
changeEmailMail	String
getActivationMail(String)	String
getActivationSite(String, String)	String
getChangeEmailMail(String, int, boolean)	String
importCertificateInJKS()	void
loadResourceIntoString(String)	String
loadResources()	void

Klasse Resources

Die **Klasse RestServer** Richtet die REST API ein und steuert sie.

Methoden:

- activateAccount → Aktiviert einen Account mit der DataBase.activateAccount(String)-Methode. Nach dem Ausführen der Methode wird auf die Startseite der Webseite weitergeleitet (mit dem Parameter "activated=true/false")
- createAccount → Erstellt einen Account mit der DataBase.createUser(String, String, String, boolean)-Methode. Der dabei erstellte Aktivierungscode wird dann per Mail.sendActivationMail(String, String, String) versendet.
- enableCORS → ktiviert CORS (Cross-Origin Resource Sharing). Dies wird z.B. für den Flutter Client benötigt um einwandfrei zu funktionieren. CORS erlaubt, dass Ressourcen (wie Bilder) von anderen Servern abgefragt wird.
- exceptionHandling → Richtet die Fehlerbehandlung ein. Hier werden Fehler zu Statuscodes zugeordnet und die passenden Fehlermeldungen an den Nutzer weitergegeben. Ebenfalls werden alle Fehler in der Konsole ausgegeben
- setupRoutes() → Richte alle Routen auf dem Server ein
- start → Startet den Server auf einem bestimmten Port.



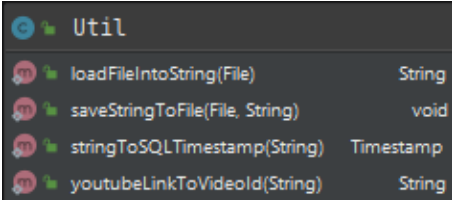
RestServer	
APPLICATION_JSON	String
TEXT_PLAIN	String
gson	Gson
activateAccount(Request, Response)	String
createAccount(Request, Response)	String
enableCORS()	void
exceptionHandling()	void
setupRoutes()	void
start(int)	void

Klasse RestServer

Die **Klasse Util** ist für verschiedene Hilfsmethoden zuständig.

Methoden:

- loadFileIntoString → Liest eine Textdatei und kopiert Sie in einen String.
- saveStringToFile → Speichert einen String in einer Datei ab. Falls diese Datei bereits existiert, wird sie überschrieben. Sollte der Pfad zu der Datei nicht existieren, wird er erstellt. Sollte ein Parameter null sein, wird die Methode direkt abgebrochen.
- youtubeLinkToVideoid → Analysiert den Link aus einem YouTube-Video und findend die Video-Id heraus



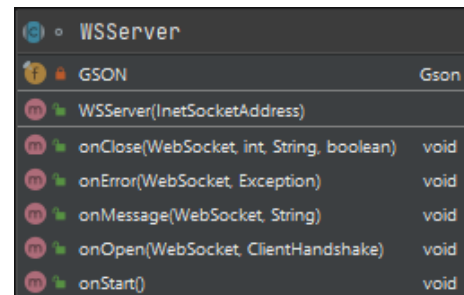
Util	
loadFileIntoString(File)	String
saveStringToFile(File, String)	void
stringToSQLTimestamp(String)	Timestamp
youtubeLinkToVideoid(String)	String

Klasse Util

Die **Klasse WSServer** ist für Implementation eines Websockets, der für die Verteilung von Live-Daten zuständig ist.

Methoden:

WSServer → Erstellt einen neuen Websocket.



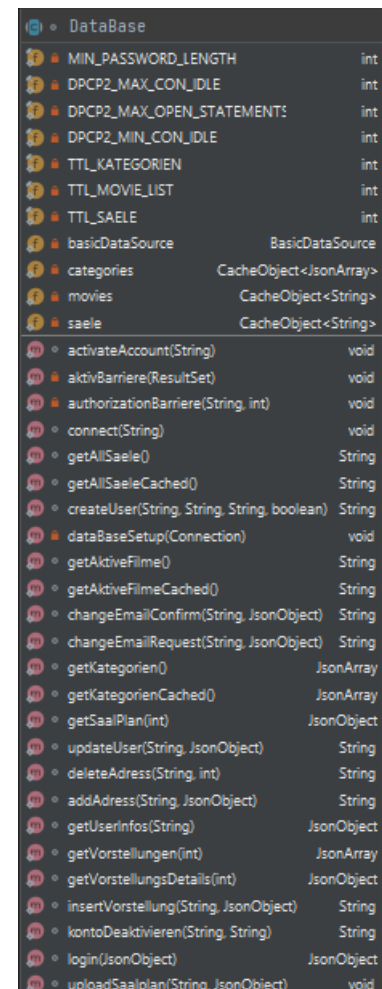
WSServer	
GSON	Gson
WSServer(InetSocketAddress)	
onClose(WebSocket, int, String, boolean)	void
onError(WebSocket, Exception)	void
onMessage(WebSocket, String)	void
onOpen(WebSocket, ClientHandshake)	void
onStart()	void

Klasse WSServer

Die **Klasse Database** ist zuständig für die Verbindung zur Datenbank und für Aktionen, die dort ausgeführt werden.

Methoden:

- activateAccout → Die Methode aktiviert einen Account mit einem Aktivierungsschlüssel.
- addAddress → Fügt einem Benutzer eine Adresse hinzu.
- aktivBarriere → Blockt alle Anfragen, bei denen ein Konto nicht aktiv ist.
- authorizationBarriere → Prüft, ob ein Account autorisiert ist, eine bestimmte Aktion auszuführen
- changeEmailConfirm → Bestätigt die Änderung einer Email-Adresse unter zur Hilfe nahme der beiden Codes, die per changeEmailRequest(String, JsonObject) versendet wurden.
- changeEmailRequest → Fordert an, dass die Email von dem, dem AuthCode zugeordneten, Konto geändert wird.
- createUser → Mit dieser Methode kann ein Benutzerkonto erstellt werden.
- dataBaseSetup → Erstellt die benötigten Tabellen, falls diese noch nicht existieren.
- deleteAddress → Löscht eine Adresse aus dem System.
- getAktiveFilme → Fragt eine Filmübersicht ab.
getAktiveFilmeCached → Fragt eine aktuelle ggf. gecachde Variante von getFragt alle vorhandenen Kinosäle ab.
AktiveFilme() ab.



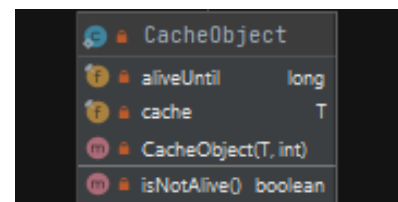
DataBase	
MIN_PASSWORD_LENGTH	int
DPCP2_MAX_CON_IDLE	int
DPCP2_MAX_OPEN_STATEMENTS	int
DPCP2_MIN_CON_IDLE	int
TTL_KATEGORIEN	int
TTL_MOVIE_LIST	int
TTL_SAELE	int
basicDataSource	BasicDataSource
categories	CacheObject<JSONArray>
movies	CacheObject<String>
saele	CacheObject<String>
activateAccount(String)	void
aktivBarriere(ResultSet)	void
authorizationBarriere(String, int)	void
connect(String)	void
getAllSaele()	String
getAllSaeleCached()	String
createUser(String, String, String, boolean)	String
dataBaseSetup(Connection)	void
getAktiveFilme()	String
getAktiveFilmeCached()	String
changeEmailConfirm(String, JsonObject)	String
changeEmailRequest(String, JsonObject)	String
getKategorien()	JSONArray
getKategorienCached()	JSONArray
getSaalPlan(int)	JsonObject
updateUser(String, JsonObject)	String
deleteAddress(String, int)	String
addAddress(String, JsonObject)	String
getUserInfos(String)	JsonObject
getVorstellungen(int)	JSONArray
getVorstellungsDetails(int)	JsonObject
insertVorstellung(String, JsonObject)	String
kontoDeaktivieren(String, String)	String
login(JsonObject)	JsonObject
uploadSaalplan(String, JsonObject)	void

Klasse Database

- `getAllSaele` → Fragt alle vorhandenen Kinosäle ab.
- `getAllSaeleCached` → Hat dieselbe Funktion wie `getAllSaele()`, fragt die Daten aber nicht gewzungenermaßen aus der Datenbank ab, sondern versucht ein gecachtes Objekt zu verwenden.
- `getKategorien` → Fragt alle vorhandenen Kinosäle ab.
- `getKathegorienCached` → Fragt eine aktuelle ggf. gecachde Variante von `getKategorien()` ab.
- `getSaalPlan` → Fragt alle Details zu einem Saal ab.
- `getUserInfos` → Ermöglicht es einem Nutzer seine eigenen Nutzerinfos abzufragen.
- `getVorstellungen` → Fragt alle Vorstellungen zu einem oder allen Filmen ab.
- `getVorstellungenDetails` → Gibt Details zu einer Vorstellung zurück.
- `kontoDeaktivieren` → Diese Methode deaktiviert ein Konto.
- `login` → Führt einen Login durch und gibt sowohl den Namen als auch den Auth Code zurück.
- `updateUser` → Aktualiser die Daten von einem Nutzer.

Ein Cache-Objekt, dass Daten zwischenspeichert

`isNotAlive` → Fragt ab, ob das Objekt noch valide (true) ist oder erneuert (false) werden muss



6 GitHub - Online-Reservierung von Kinotickets



GitHub

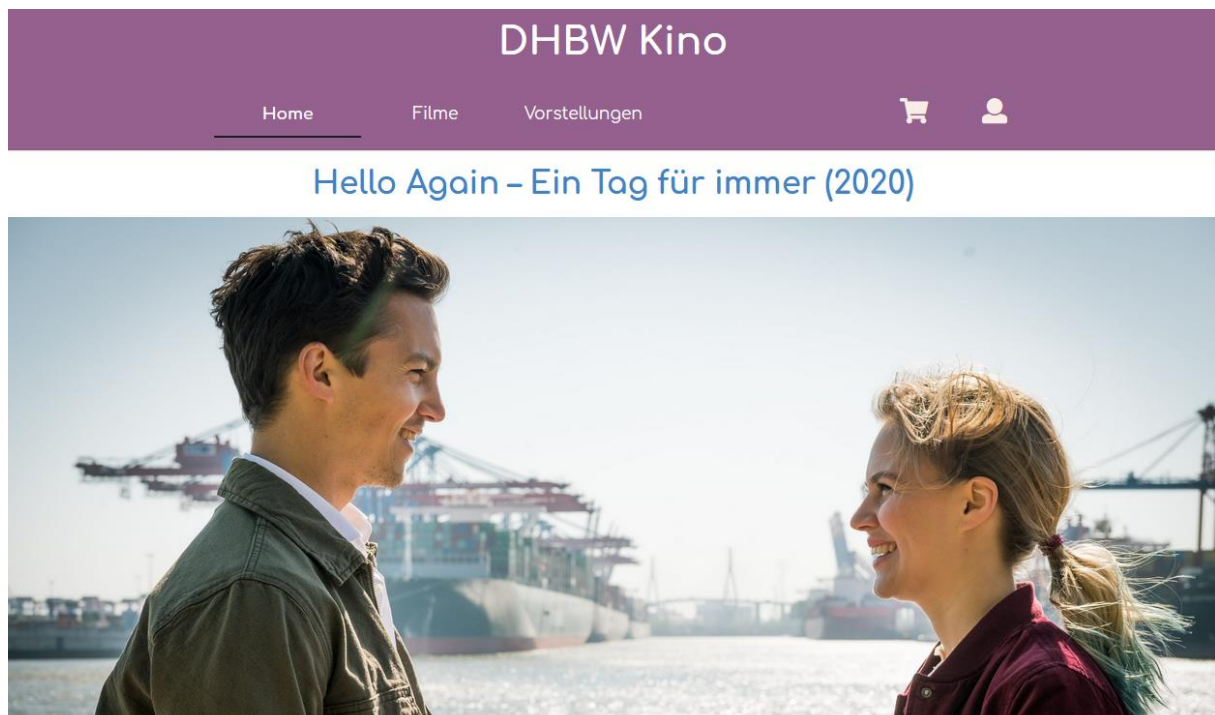
6.1 Frontend

<https://github.com/Niklas00122/kinoticketing>

6.2 Backend

<https://github.com/NoahMoritz/cinema-backend>

7 Frontend Dokumentation



Home_01

Nachdem der Benutzer die Seite <https://dhbw-kino.de/> aufgerufen hat, bekommt dieser die angesagtesten Filme angezeigt.

Unserer Filme

Hello Again – Ein Tag für immer
2020
FSK: 6

Zazie (Alicia von Rittberg) lebt gemeinsam mit ihren Freunden Patrick (Samuel Schneider) und Anton (Edin Hosanovic) in einer WG und in einer Sache sind sich die Mitbewohner einig: Sie sind allesamt nicht für die große Liebe und feste Beziehungen gesc... [mehr lesen](#)

Mehr erfahren

The Outpost – Überleben ist Alles
2020
FSK: 12

Camp Keating ist ein Außenposten des US-Militärs, der nahe der afghanischen Stadt Kamdesh liegt und eigentlich für Frieden und Ordnung in der Region sorgen soll. Doch Camp Keating liegt auch an einer sehr angreifbaren Position, von Bergen umgeben und... [mehr lesen](#)

Mehr erfahren

X-Men: New Mutants
2020
FSK: 16

Rahne Sinclair alias Wolfsbane (Maisie Williams), Illyana Rasputin alias Magik (Anyu Taylor-Joy), Sam Guthrie alias Cannonball (Charlie Heaton) und Roberto da Costa alias Sunspot (Henry Zogo) sind zusammen in einer besonderen Klinik untergebracht, wo... [mehr lesen](#)

Mehr erfahren

After Truth
2020
FSK: 12

Tessa (Josephine Langford) hat noch immer mit der Enthüllung zu kämpfen, dass Hardin (Hero Fiennes Tiffin) zuerst nur im Rahmen einer Wette dafür sorgen wollte, dass sie sich in ihn verliebt. Alle Beteuerungen, dass er sich dann jedoch selbst in sie ... [mehr lesen](#)

Mehr erfahren

Die Boonies - Eine bärenstarke Zeitreise
2020
FSK: 6

Die Bärenbrüder Briar und Bramble reisen gemeinsam mit ihrem Widersacher, dem Holzfäller Vick in die Vergangenheit, als der gelangweilte Bramble einem mysteriösen Schmetterling hinterherjagt und dieser eine Art Portal in die Steinzeit öffnet. Hier mü... [mehr lesen](#)

Mehr erfahren

Home_02

Der Benutzer kann runterscrollen, dann werden ihm weitere Filme angezeigt. Außerdem wird zu jedem Film eine Kurzbeschreibung angezeigt (siehe Home_02).

Home
Filme
Vorstellungen

Suche...

Hello Again – Ein Tag für immer

2020
FSK: 6

Zazie (Alicia von Rittberg) lebt gemeinsam mit ihren Freunden Patrick (Samuel Schneider) und Anton (Edin Hasanovic) in einer WG und in einer Sache sind sich die Mitbewohner einig: Sie sind allesamt nicht für die große Liebe und feste Beziehungen gesc... [mehr lesen](#)

Mehr erfahren

The Outpost – Überleben ist Alles

2020
FSK: 12

Camp Keating ist ein Außenposten des US-Militärs, der nahe der afghanischen Stadt Kamdesh liegt und eigentlich für Frieden und Ordnung in der Region sorgen soll. Doch Camp Keating liegt auch an einer sehr angreifbaren Position, von Bergen umgeben und... [mehr lesen](#)

Mehr erfahren

X-Men: New Mutants

2020
FSK: 16

Rahne Sinclair alias Wolfsbane (Maisie Williams), Ilyana Rasputin alias Magik (Anyu Taylor-Joy), Sam Guthrie alias Cannonball (Charlie Heaton) und Roberto da Costa alias Sunspot (Henry Zaga) sind zusammen in einer besonderen Klinik untergebracht, wo... [mehr lesen](#)

Mehr erfahren

After Truth

2020
FSK: 12

Tessa (Josephine Langford) hat noch immer mit der Enthüllung zu kämpfen, dass Hardin (Hero Fiennes Tiffin) zuerst nur im Rahmen einer Wette dafür sorgen wollte, dass sie sich in ihn verliebt. Alle Beteuerungen, dass er sich dann jedoch selbst in sie ... [mehr lesen](#)

Mehr erfahren

Die Boonies - Eine bärenstarke Zeitreise

2020
FSK: 6

Die Bärenbrüder Briar und Bramble reisen gemeinsam mit ihrem Widersacher, dem Holzfäller Vick in die Vergangenheit, als der gelangweilte Bramble einem mysteriösen Schmetterling hinterherjagt und dieser eine Art Portal in die Steinzeit öffnet. Hier mü... [mehr lesen](#)

Mehr erfahren

Filme_01

Durch das Betätigen des Buttons „Filme“ in der Navigationsleiste bekommt der Benutzer alle Filme angezeigt. Zudem kann der Benutzer nach Filmen suchen. Um Filme zu suchen muss der Benutzer den Filmenamen in die Suchleiste eingeben.

The Outpost - Überleben ist Alles

FSK: 12 Filmstart: 17.09.2020 Laufzeit: 123 Min.

Camp Keating ist ein Außenposten des US-Militärs, der nahe der afghanischen Stadt Kamdesh liegt und eigentlich für Frieden und Ordnung in der Region sorgen soll. Doch Camp Keating liegt auch an einer sehr angreifbaren Position, von Bergen umgeben und nahe der pakistanischen Grenze, weswegen ständig die Gefahr eines Taliban-Angriffs in der Luft liegt. Die US-Soldaten Clint Romesha (Scott Eastwood), Justin Gallegos (Jacob Scipio), Michael Scusa (Scott Alda Coffey) und Josh Kirk (Jack Kesy) werden als Verstärkung nach Camp Keating versetzt, wo sie von Captain Keating (Orlando Bloom) in Empfang genommen werden. Einige Zeit nach ihrer Ankunft soll der Außenposten dann geschlossen werden. Doch die erleichterten Soldaten haben sich zu früh gefreut, denn die Taliban planen vor dem Abzug der Amerikaner noch einen letzten Angriff...

Filstart: 17.09.2020

FSK: 12

Länge: 123

Land: USA



Datum wählen

Montag, 16.11 Mittwoch, 18.11 Donnerstag, 19.11 Dienstag, 24.11 Montag, 30.11

14:01

05:15

09:33

13:30

14:35

Dienstag, 01.12 Sonntag, 06.12

00:30

22:53

Filme_02

Nachdem der Benutzer auf „Mehr Erfahrungen“ in Abbildung Filme_01 geklickt hat, gelangt er auf die nächste Seite (Filme_02). Auf dieser wird dem Benutzer eine Kurzbeschreibung zum Film angezeigt. Durch das betätigen eines der Uhrzeit Buttons in der unteren Bildhälfte kann der Benutzer Tickets zum jeweiligen Film buchen.

Datum wählen

Montag, 16.11

Mittwoch, 18.11

Donnerstag, 19.11

Dienstag, 24.11

Montag, 30.11

14:01

05:15

09:33

13:30

14:35

Dienstag, 01.12

Sonntag, 06.12

00:30

22:53



DHBW Kino
[Impressum](#) [Datenschutzerklärung](#)

Filme_03

Durch das Runterscrollen bekommt der Benutzer einen Trailer zum jeweiligen Film angezeigt.

Home
Filme
Vorstellungen

Datum wählen

Hello Again – Ein Tag für immer

FSK: 6 | 92min.

Romantische Komödie, die "Und täglich grüßt das Murmeltier" mit "Die Hochzeit meines besten Freundes" kreuzt.

Sonntag, 15.11 Montag, 16.11 Dienstag, 17.11 Mittwoch, 18.11

23:05
00:00
13:45
14:30

The Outpost - Überleben ist Alles

FSK: 12 | 123min.

Auf einer wahren Begebenheit basierender Kriegs-Actioner mit Starbesetzung

Montag, 16.11 Mittwoch, 18.11 Donnerstag, 19.11 Dienstag, 24.11

14:01
05:15
09:33
13:30

X-Men: New Mutants

FSK: 16 | 94min.

Horrorlastiges "X-Men"-Spinoff um fünf junge Mutanten, die in einem Krankenhaus gefangen gehalten werden

Mittwoch, 18.11 Donnerstag, 19.11 Donnerstag, 19.11 Donnerstag, 19.11

Vorstellungen

Nachdem der Benutzer „Vorstellungen“ in der Navigationsleiste betätigt hat, gelangt er zur Seite Vorstellungen. Hier kann er nach allen Vorstellungen filtern. Zu jeder Vorstellung wird dem Benutzer eine Kurzbeschreibung des Filmes sowie das Datum und die Uhrzeit der Vorstellung angezeigt. Durch das Betätigen einer Uhrzeit gelangt der Benutzer zur Reservierung des Filmes.

Zurück

14:01 Uhr am 16.11.2020

Tickets reservieren für: The Outpost - Überleben ist Alles

Auf einer wahren Begebenheit basierender Kriegs-Actioner mit Starbesetzung.

- Länge: 123 Minuten
- FSK: 12
- Saal: Kino 1

Reservierung_Film

Bei der Reservierung eines Filmes, wird dem Benutzer erneut eine Kurzbeschreibung des Filmes angezeigt. Als nächstes kann der Benutzer die Sitzplätze aussuchen.

Bitte wählen Sie Ihre Plätze aus.

Leinwand

Gesamtpreis: 0€ inkl. MwSt.

Tickets kaufen

Sitzplatzauswahl

Bei der Sitzplatzauswahl hat der Benutzer die Möglichkeit zwischen einer Loge, Loveseat, Parkett oder Premium Sitzplatz auszuwählen.

Bitte wählen Sie Ihre Plätze aus.

Leinwand

Loge * 1 = 11.5€
Barrierefrei * 1 = 9.5€
Loveseat * 1 = 10€
Premium * 1 = 13.5€
Parkett * 1 = 10€

Gesamtpreis: 54.5€ inkl. MwSt.

Tickets kaufen

Sitzplatzauswahl_Sitzplätze_ausgewählt

Nachdem der Benutzer einen oder mehrere Sitzplätze ausgesucht hat, muss dieser auf den Button Tickets kaufen klicken.

DHBW Kino

[Home](#)
[Filme](#)
[Vorstellungen](#)




Wie möchten Sie fortfahren?

Login



Als Gast kaufen



DHBW Kino



[Impressum](#)
[Datenschutzklärung](#)

Wie_möchten_Sie_fortfahren

Nachdem der Benutzer auf "Tickets kaufen" geklickt hat, bekommt er die Auswahl wie er fortfahren möchte. Entweder loggt er sich ein oder tätigt den Kauf als Gast.

DHBW Kino

[Home](#)
[Filme](#)
[Vorstellungen](#)

Bitte füllen Sie das Formular aus um den Kauf abzuschließen.

Vollständiger Name *

Anrede *

Name

Anrede

Strasse *

Hausnummer *

Postleitzahl *

Stadt *

Straße

Hausnummer

Postleitzahl

Stadt

Präfix

Telefonnummer

Email

+49

Telefonnummer

@

Ihre Bestellung:

The Outpost - Überleben ist Alles

Loge * 1 = 11.5€

Barrierefrei * 1 = 9.5€

Loveseat * 1 = 10€

Premium * 1 = 13.5€

Parkett * 1 = 10€

Gesamtpreis: 54.5€ inkl. MwSt.

Bezahlung wählen.

☒ Barzahlung

Sie können Ihre Tickets bei uns abholen und bar bezahlen.

☐ Paypal

Kostenpflichtig kaufen

Ihre_Bestellung

37

Klickt er auf "Als Gast kaufen" gelangt er auf diese Seite. Hier ist in der unteren Hälfte seine Bestellung aufgelistet und rechts daneben sind seine möglichen Zahlungsarten zu sehen. Doch bevor er als Gast den Kauf abschließen kann muss er das obige Formular zunächst mal ausfüllen.

Bitte füllen Sie das Formular aus um den Kauf abzuschließen.

Vollständiger Name *	Anrede *		
<input type="text" value="Max Müller"/>	<input type="text" value="Herr"/>		
Straße *	Hausnummer *	Postleitzahl *	Stadt *
<input type="text" value="Musterstraße"/>	<input type="text" value="12"/>	<input type="text" value="12345"/>	<input type="text" value="Berlin"/>
Präfix	Telefonnummer	Email	
<input type="text" value="+49"/>	<input type="text" value="123456789"/>	<input type="text" value="Max.Müller@Test.de"/>	

Formular

Ausgefüllt sieht das Formular ungefähr so aus.

DHBW KINO

Hier können Sie Kinotickets reservieren und buchen.

Einloggen

Melden Sie sich mit ihren Zugangsdaten an.

@ Email

Passwort

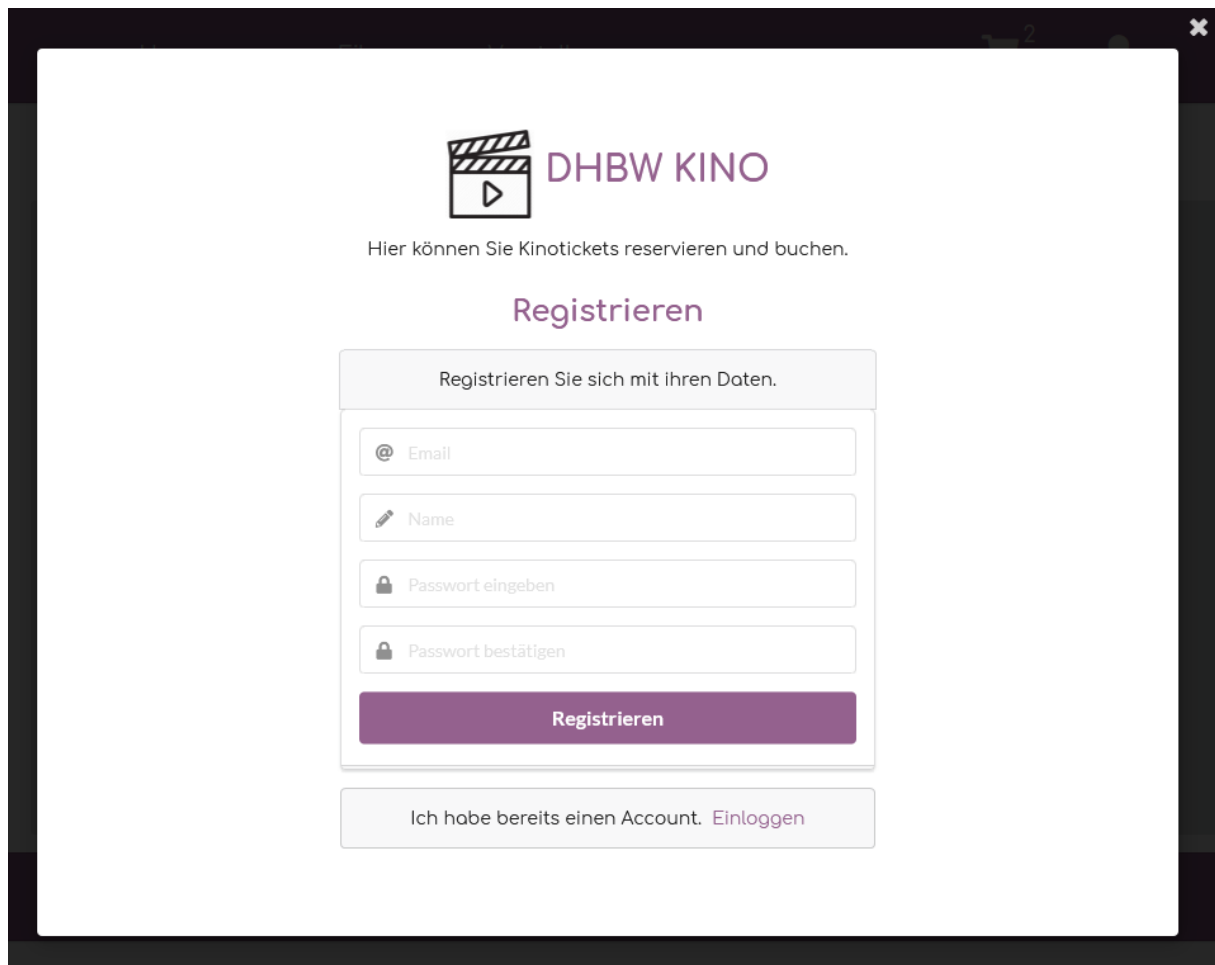
[Passwort vergessen?](#)


Einloggen

Neu bei uns? [Registrieren](#)

Einloggen

Dies ist das Pop-Up, dass sich öffnet, wenn der Benutzer nicht auf "Als Gast kaufen" sondern auf "Login" klickt. Hier wird nach den Zugangsdaten für das Login gefragt. Bevor man sich einloggen kann muss man sich zuerst auf der Seite registrieren.




 **DHBW KINO**


Hier können Sie Kinotickets reservieren und buchen.


Registrieren

Registrieren Sie sich mit ihren Daten.

@ Email

 Name

 Passwort eingeben

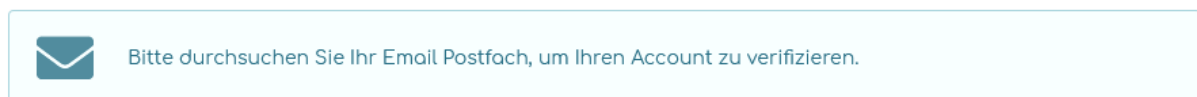
 Passwort bestätigen

Registrieren

Ich habe bereits einen Account. [Einloggen](#)

Registrieren

Über den "Registrieren"-Button gelangt man auf diese Seite. Hier wird nach einer E-Mail, einem Namen und einem Passwort für die Registrierung gefragt.



Einloggen

E-Mail_durchsuchen

Sobald man seine Daten eingetragen hat und auf "Registrieren" gedrückt hat, bekommt man diese Mitteilung und muss seinen Account über die Bestätigungsemail verifizieren.

Willkommen!



Es fehlt nur noch ein Schritt, um Ihr Konto zu bestätigen! Klicken Sie dafür einfach auf den Button.

Konto bestätigen

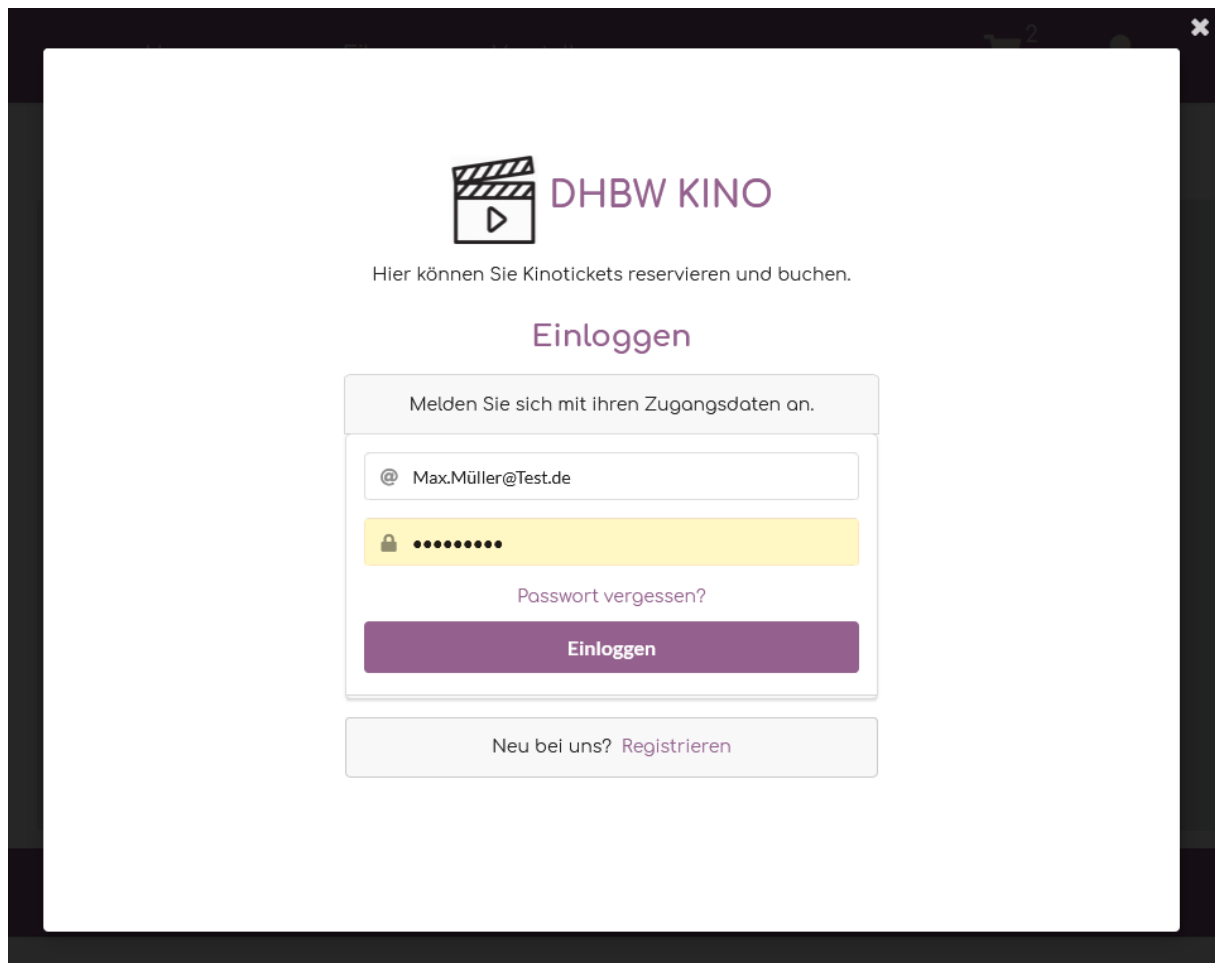
Falls der Button nicht funktioniert, kopieren Sie bitte den folgenden Link in Ihren Browser:

<https://dhbw-kino.de:4567/activate/101bbc70-46db-4a8e-a020-9cdae175462f>

Beste Grüße,
DHBW Kino

Willkommen

Sobald man auf den Link in der Mail geklickt hat wird das Konto aktiviert. Sonst bleibt es deaktiviert.



Einloggen

Nun kann man sich mit seinen Daten einloggen.

<p>Bitte füllen Sie das Formular aus um den Kauf abzuschließen.</p> <p>Adresse hinzufügen</p>	<p>Adresse auswählen.</p> <p>Bitte fügen Sie eine Adresse hinzu.</p>
<p>Ihre Bestellung:</p> <p>The Outpost - Überleben ist Alles</p> <p>Barrierefrei * 1 = 9.5€</p> <p>Loge * 1 = 11.5€</p> <p>Loveseat * 1 = 10€</p> <p>Premium * 1 = 13.5€</p> <p>Parkett * 1 = 10€</p> <p>Gesamtpreis: 54.5€ inkl. MwSt.</p>	<p>Bezahlung wählen.</p> <p><input checked="" type="radio"/> Barzahlung</p> <p>Sie können Ihre Tickets bei uns abholen und bar bezahlen.</p> <p><input type="radio"/> Paypal</p>
<p>Kostenpflichtig kaufen</p>	

Bestellung

Nach dem Betätigen der "Einloggen" Taste gelangt man auch wieder zur Bestellübersicht. "Adresse hinzufügen" drücken.

The Outpost - Überleben ist Alles

Barzahlung

+

Kontaktdaten hinzufügen

Aktuell sind keine Kontaktdaten hinterlegt.

Fügen Sie Kontaktdaten hinzu, um ihren Kauf fortzusetzen.

Kontaktdaten

Nun muss man seine "Kontaktdaten hinzufügen".

Ihre Bestellung

Bezahlung wählen

Vollständiger Name *

Max Müller

Anrede *

Herr

Straße *

Musterstraße

Hausnummer *

12

Postleitzahl *

12345

Stadt *

Berlin

Präfix

+49

Telefonnummer

123456789

⏪

Zurück

💾

Speichern

Kontaktdaten_hinzufügen

Ähnlich wie im Formular zuvor. Nur muss diese mal nicht die E-Mail-Adresse eingegeben werden. Mit dem "Speichern" Knopf werden die Kontaktdaten gespeichert.

Bitte füllen Sie das Formular aus um den Kauf abzuschließen.

Adresse hinzufügen

Adresse auswählen.

● Herr Max Müller, Musterstraße 12 (12345)

Hierhin liefern.

Ihre Bestellung:

The Outpost - Überleben ist Alles

Barrierefrei * 1 = 9.5€

Loge * 1 = 11.5€

Loveseat * 1 = 10€

Premium * 1 = 13.5€

Parkett * 1 = 10€

Gesamtpreis: 54.5€ inkl. MwSt.

Bezahlung wählen.

● Barzahlung

Sie können Ihre Tickets bei uns abholen und bar bezahlen.

○ Paypal

Kostenpflichtig kaufen

Bestellung

Dadurch kann man sie nun beim Bestellvorgang auswählen und muss diese nicht erneut eingeben.

Bitte füllen Sie das Formular aus um den Kauf abzuschließen.

Adresse hinzufügen

Ihre Bestellung:

The Outpost - Überleben ist Alles

Barrierefrei * 1 = 9,5€

Loge * 1 = 11,5€

Loveseat * 1 = 10€

Premium * 1 = 13,5€

Parkett * 1 = 10€

Gesamtpreis: 54.5€ inkl. MwSt.

Adresse auswählen.

● Herr Max Müller, Musterstraße 12 (12345)

Hierhin liefern.

Bezahlung wählen.

☐ Barzahlung

● **Paypal**

Bezahlen Sie bequem per Paypal und wir senden Ihnen die Tickets per Email zu.

PayPal

Debit or Credit Card

SEPA Lastschrift

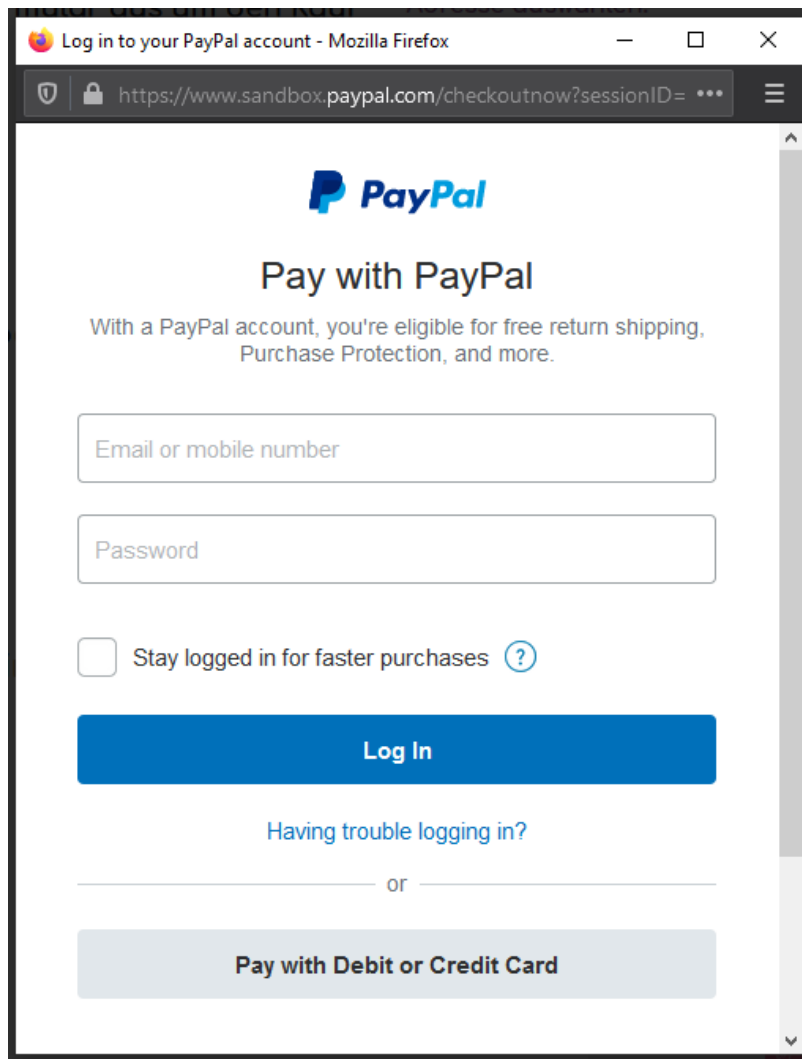
giropay

Sofort

Powered by **PayPal**

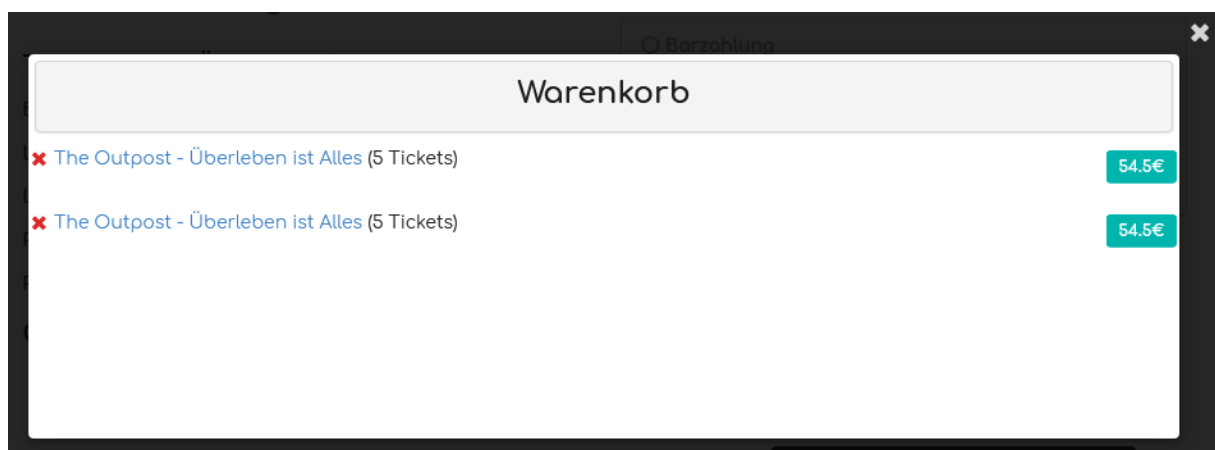
Bezahlung

Neben der Barzahlung ist auch die Bezahlung per PayPal möglich. Klickt man nun auf "PayPal".



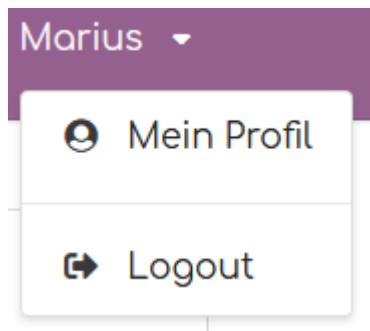
PayPal

Öffnet sich dieses PayPal Pop-Up. Hier kann man sich mit seinem PayPal Account anmelden und seine Tickets im Vorhinein bezahlen.



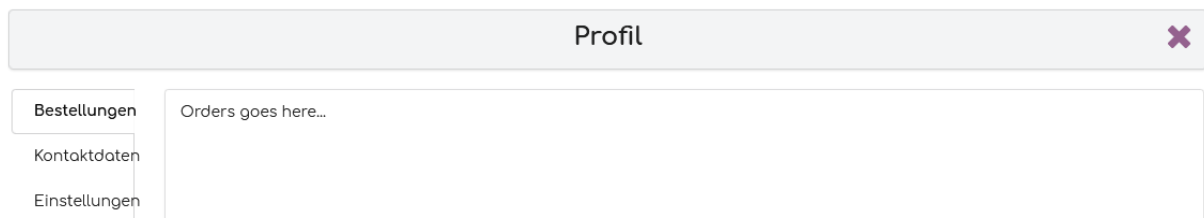
Warenkorb

Wie auf der Homepage zu sehen gibt es auch ein Einkaufswagen-Icon. Diese steht für den Warenkorb in welchem alle noch nicht vollendeten Bestellungen enthalten sind. Außerdem ist es möglich hier diese Bestellungen abubrechen bzw. zu löschen.



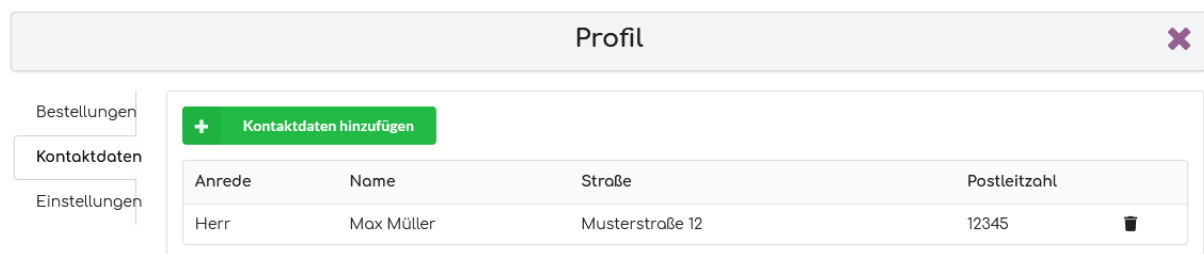
Mein_Profil

Klickt man oben rechts auf seinen Namen öffnet sich diese Drop-Down. Mit dem Logout-Button kann man sich ausloggen. Klickt man auf "Mein Profil" kann man seine persönlichen Informationen einsehen.



Profil

Hier sind zum einen alle vollendeten Bestellungen aufgelistet.



Kontaktdaten

Als auch alle abgespeicherten Kontaktdaten, von welchen mehrere angelegt und auch wieder gelöscht werden können.

Admin Bereich

Markieren Sie Bestellungen als markiert oder fügen Sie neue Vorstellungen hinzu.

[Vorbestellungen einsehen](#) [Vorstellungen anlegen](#)

Film auswählen *

Saal auswählen *

Basis Preis *

X-Men: New Mutants

Kino 1

7

✕ 2020-11-13 09:20

Vorstellung anlegen

DHBW Kino
[Impressum](#) [Datenschutzerklärung](#)

Admin_Bereich

Anhand des Basis Preises werden außerdem die restlichen Preise für besondere Sitzplätze errechnet.

Admin Bereich

Markieren Sie Bestellungen als markiert oder fügen Sie neue Vorstellungen hinzu.

[Vorbestellungen einsehen](#) [Vorstellungen anlegen](#)

Film auswählen *

Saal auswählen *

Basis Preis *

Film

Saal

0

Vorstellungsbeginn

Vorstellung anlegen

Erfolgreich.
Vorstellung wurde erfolgreich angelegt.

DHBW Kino
[Impressum](#) [Datenschutzerklärung](#)

Adminbereich_02

Nachdem der Kunde seine Vorstellung angepasst hat und auf den Button "Vorstellung anlegen" geklickt hat, bekommt er bei erfolgreicher Erstellung der Vorstellung diese Mitteilung.