Hochschule Bremen
City University of Applied Sciences

HSB

Cinemagic

Kinoticket-Buchungs-Applikation

Internettechnologien

HOCHSCHULE BREMEN

Projektarbeit im Internationalen Frauenstudiengang Informatik
der Fakultät Elektrotechnik und Informatik

Cinemagic

Kinoticket-Buchungs-Applikation

Eingereicht von: Juohina Khaled | Matrikelnr. 5194447

Kathrin Jüttner | Matrikelnr. 5183520

Im Modul: Internettechnologien

Erarbeitet im: 6. Semester

Abgegeben am: 19. Juli 2024

Prüfer: Prof. Dr. Benjamin Tannert

Inhaltsverzeichnis

1. Projektübersicht

Das vorliegende Projekt stellt eine innovative und benutzerfreundliche Kinoanwendung vor, die mehrere zentrale Funktionen bietet, um den Nutzern ein ultimatives Kinoerlebnis bieten zu können. Die Anwendung ermöglicht es den Nutzern, sich zu registrieren und anzumelden, verfügbare Filmvorführungen zu durchsuchen, Tickets zu buchen sowie Filme zu bewerten.

Die benutzerfreundliche Oberfläche der Anwendung erleichtert die Navigation durch die verschiedenen Funktionen und sorgt für ein nahtloses Nutzungserlebnis. Eine der Hauptfunktionen der Anwendung ist die Anzeige der verfügbaren Filmvorführungen. Nutzer können sich eine umfassende Übersicht der aktuell laufenden und kommenden Filme anzeigen lassen. Der auf der Startseite befindende Slider bietet außerdem die Möglichkeit, sich einen Trailer ansehen zu können. Jeder Film zeigt detaillierte Informationen wie Titel, Beschreibung, Dauer und Vorführungszeiten an. Diese strukturierte und informative Darstellung ermöglicht es den Nutzern, schnell und einfach die gewünschten Filme zu finden und die passenden Vorführungen auszuwählen. Es können lediglich Vorführungen für einen Zeitraum von einer Woche ab dem aktuellen Datum ausgewählt werden.

Ein weiteres zentrales Feature der Anwendung ist das Buchungssystem. Nutzer können Tickets für ausgewählte Filmvorführungen direkt über die Anwendung buchen. Wichtig dabei ist, dass die Plätze, ohne sich eingeloggt zu haben ausgewählt werden, jedoch erst nach der Anmeldung gebucht werden können. Der Buchungsvorgang umfasst die Auswahl der gewünschten Sitzplätze im Saal, wobei ein interaktiver Sitzplan zur Verfügung steht. Dieser Sitzplan erleichtert die Auswahl der besten verfügbaren Plätze und bietet eine visuelle Darstellung des Saals, um den Nutzern eine optimale Platzwahl zu ermöglichen. Um zu vermeiden, dass mehrere Kunden zur gleichen Zeit die gleichen Plätze auswählen können, werden WebSockets eingesetzt.

Die Anwendung unterstützt die Benutzerregistrierung und -anmeldung, um personalisierte Funktionen anzubieten. Nach der Registrierung und Anmeldung haben Nutzer Zugriff auf ihre Buchungshistorie. Zusätzlich zur Buchung und Anmeldung bietet die Anwendung ein Bewertungssystem. Nutzer können Filme, die sie gesehen haben, bewerten. Diese Bewertungen helfen anderen Nutzern bei der Auswahl von Filmen und bieten wertvolles Feedback für das Kino. Somit trägt das Bewertungssystem zur kontinuierlichen Verbesserung des Kinoangebots bei und fördert den Austausch innerhalb der Nutzercommunity.

2. Genutzte Technologien

Für die Entwicklung der Kinoanwendung wurden folgende Technologien verwendet:

2.1 Angular

Angular wurde verwendet, um die Frontend-Anwendung zu entwickeln. Es ermöglicht die Erstellung von Single-Page-Anwendungen (SPAs) mit einer klaren Struktur und wiederverwendbaren Komponenten. In der Kinoanwendung wird Angular genutzt, um die Benutzeroberfläche zu erstellen, Daten zu binden, Formulare zu verarbeiten und HTTP-Anfragen an den Backend-Server zu senden.

2.2 IntelliJ IDEA

IntelliJ IDEA wurde als integrierte Entwicklungsumgebung (IDE) genutzt. Es bietet umfangreiche Tools für die Entwicklung, das Debugging und das Refactoring von Code, was die Produktivität der Entwickler erheblich verbessert hat. In der Kinoanwendung wurde IntelliJ IDEA zur Entwicklung und Verwaltung des gesamten Projekts eingesetzt, sowohl für das Frontend als auch das Backend.

2.3 Angular CLI

Angular CLI (Command Line Interface) wurde zur Verwaltung des Angular-Projekts verwendet. Es bietet Werkzeuge zum Erstellen, Entwickeln und Testen von Angular-Anwendungen und kann über IntelliJ IDEA installiert und verwendet werden. Angular CLI erleichtert die Entwicklung, indem es eine standardisierte Projektstruktur und zahlreiche Befehle zur Automatisierung von Entwicklungsaufgaben bereitstellt.

2.4 MySQL

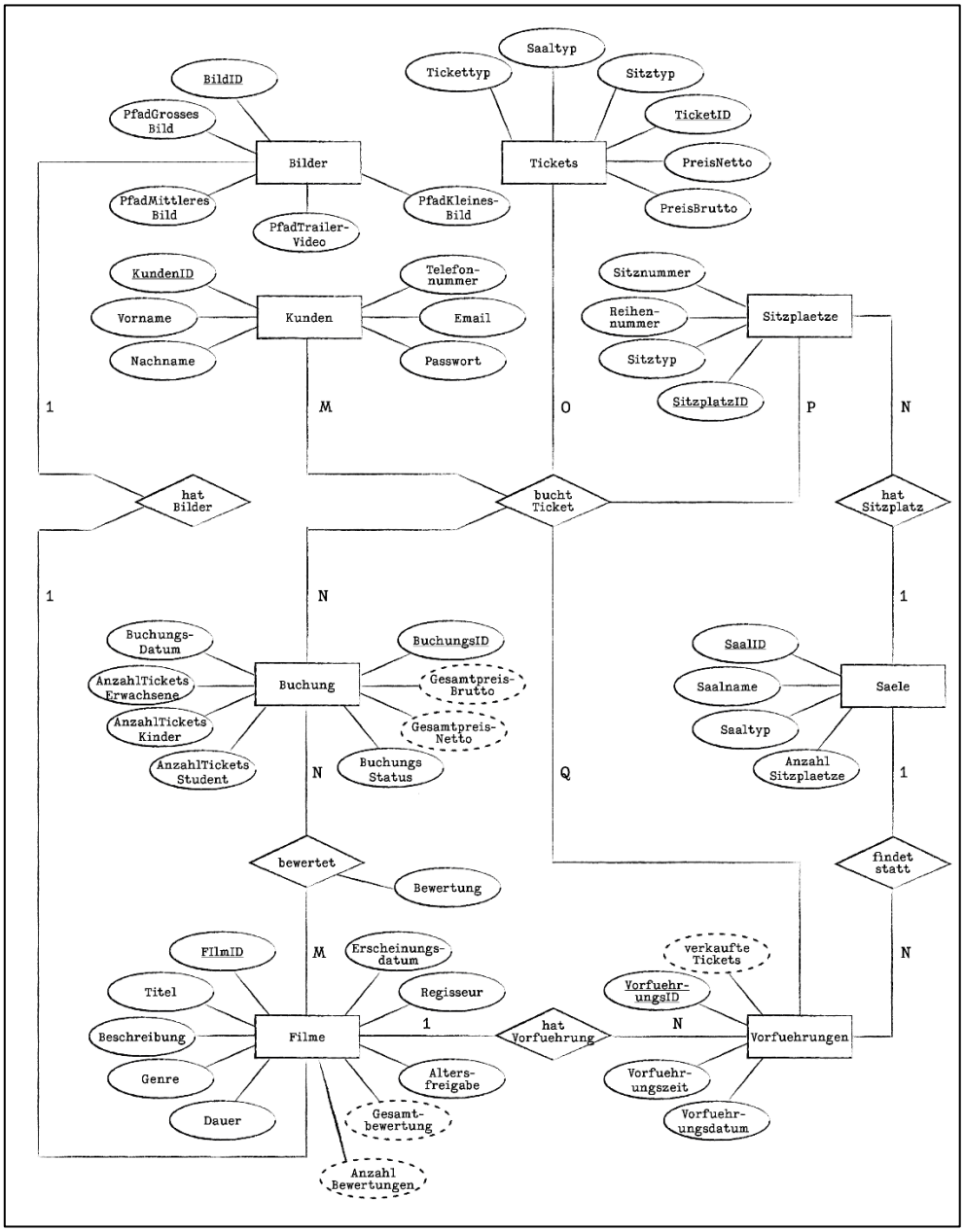
MySQL wurde als Datenbankverwaltungssystem verwendet. Es speichert die Daten der Anwendung, einschließlich Informationen über Kunden, Filme, Vorführungen, Buchungen und Sitzplätze. MySQL ermöglicht die effiziente Verwaltung und Abfrage dieser Daten, was für die Funktionalität der Kinoanwendung entscheidend ist.

2.5 Node.js

Node.js wurde verwendet, um das Backend der Anwendung zu entwickeln. Es ermöglicht die Ausführung von JavaScript-Code auf dem Server und wird für die Verarbeitung von HTTP-Anfragen, den Zugriff auf die MySQL-Datenbank und die Echtzeitkommunikation über WebSockets genutzt. In der Kinoanwendung verwaltet Node.js die Geschäftslogik und stellt APIs für das Frontend bereit.

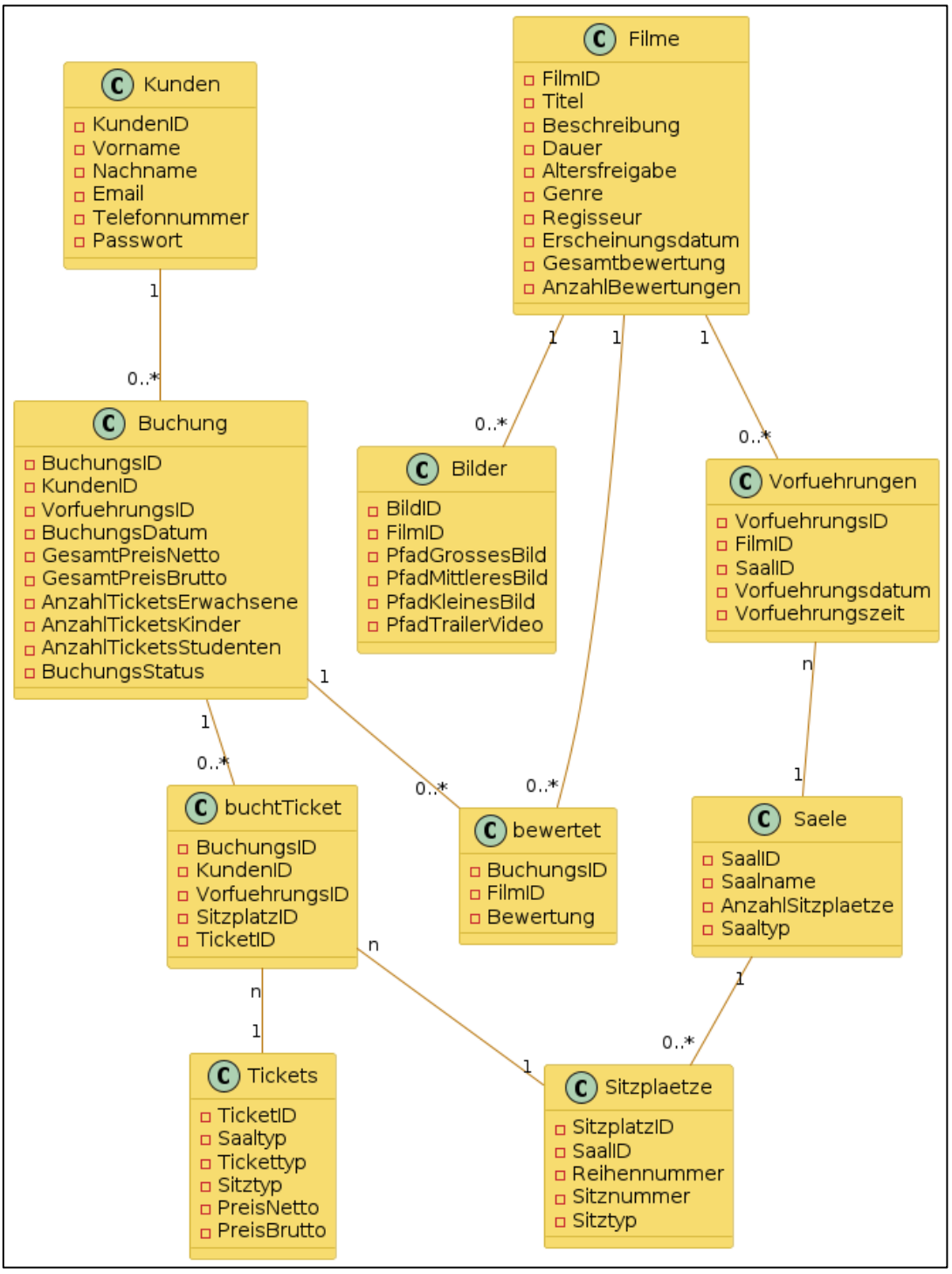
3. ER-Modell

Das vorliegende Entity-Relationsship-Modell repräsentiert die Struktur und den Aufbau der erstellten Datenbank. Es befindet sich in der dritten Normalform und setzt sich aus den Entitäten zusammen. Die Entitäten sind mit einer definierten Menge an Attributen sowie einen Primärschlüssel zur eindeutigen Identifikation versehen. Während Entitäten in einem Rechteck dargstellt werden, finden sich die Attribute in einem Oval wieder. Der Primärschlüssel der jeweiligen Entitäten sind durch einen Unterstrich gekennzeichnet. Die Objekte stehen miteinander in Beziehung, was durch einen Beziehungs-Notation in Form einer Raute und den entsprechenden Kardinalitäten modelliert ist. Die Kardinalität entspricht der möglichen Anzahl von Ausprägungen eines beteiligten Objektes an einer Beziehung. Außerdem legt der Grad deiner Beziehung fest, wie viele Objekte an einer Beziehung beteiligt sind, wobei es sich mindestens um zwei Objekte handeln muss.



4. Klassendiagramm

Aus der Überführung des ER-Modells in ein relationales Datenbank-Modell lässt sich das Klassendiagramm ableiten und stellt die Grundlage für die zu generierenden Tabellen für die Datenbank dar.



5. Datenbank

Die Tabelle **Kunden** speichert Informationen über die Kunden, die sich auf der Plattform registrieren und Buchungen vornehmen. Sie enthält Felder für eine eindeutige Kundenkennung (KundenID), den Vornamen, Nachnamen, die E-Mail-Adresse, Telefonnummer und das Passwort des Kundenkontos.

Die Tabelle **Filme** enthält Informationen über die Filme, die im Kino gezeigt werden. Jedes Feld in dieser Tabelle beschreibt eine bestimmte Eigenschaft des Films. Die Tabelle umfasst eine eindeutige Filmkennung (FilmID), den Titel des Films, eine Beschreibung, die Dauer in Minuten, die Altersfreigabe, das Genre, den Regisseur, das Erscheinungsdatum, die durchschnittliche Bewertung (Gesamtbewertung) und die Anzahl der Bewertungen, die der Film erhalten hat.

Die Tabelle **Bilder** speichert die Bildinformationen, die mit den Filmen verknüpft sind. Jedes Bild hat eine eindeutige Bildkennung (BildID) und ist über die FilmID mit einem Film verknüpft. Diese Tabelle enthält Felder für den Pfad zu verschiedenen Bildgrößen (PfadGrossesBild, PfadMittleresBild, PfadKleinesBild) und den Pfad zum Trailer-Video (PfadTrailerVideo).

Die Tabelle **Vorfuehrungen** enthält Informationen über die Filmvorführungen. Jede Vorführung hat eine eindeutige Vorführungskennung (VorfuehrungsID) und ist mit einem Film (FilmID) und einem Saal (SaalID) verknüpft. Weitere Felder beschreiben das Datum (Vorfuehrungsdatum) und die Uhrzeit (Vorfuehrungszeit) der Vorführung.

Die Tabelle **Saele** speichert Informationen über die Kinosäle. Jeder Saal hat eine eindeutige Saalkennung (SaalID), einen Saalnamen, die Anzahl der Sitzplätze (AnzahlSitzplaetze) und den Typ des Saals (Saaltyp).

Die Tabelle **Sitzplaetze** enthält Informationen über die Sitzplätze in den Sälen. Jeder Sitzplatz hat eine eindeutige Sitzplatzkennung (SitzplatzID) und ist mit einem Saal (SaalID) verknüpft. Weitere Felder beschreiben die Reihennummer (Reihennummer), die Sitznummer (Sitznummer) und den Sitztyp (Sitztyp).

Die Tabelle **Buchung** speichert Informationen über die Buchungen, die von den Kunden vorgenommen werden. Jede Buchung hat eine eindeutige Buchungskennung (BuchungsID) und ist mit einem Kunden (KundenID) und einer Vorführung (VorfuehrungsID) verknüpft. Weitere Felder beschreiben das Buchungsdatum (BuchungsDatum), den Gesamtpreis netto (GesamtPreisNetto), den Gesamtpreis brutto (GesamtPreisBrutto), die Anzahl der Tickets für Erwachsene (AnzahlTicketsErwachsene), Kinder (AnzahlTicketsKinder) und Studenten (AnzahlTicketsStudenten) sowie den Buchungsstatus (BuchungsStatus).

Die Tabelle **buchtTicket** speichert die Sitzplatzreservierungen, die im Rahmen einer Buchung vorgenommen werden. Jedes reservierte Ticket hat eine Buchungskennung (BuchungsID), eine Kundenkennung (KundenID), eine Vorführungskennung (VorfuehrungsID), eine Sitzplatzkennung (SitzplatzID) und eine Ticketkennung (TicketID).

Die Tabelle **Tickets** enthält Informationen über die verschiedenen Tickettypen. Jedes Ticket hat eine eindeutige Ticketkennung (TicketID), den Saaltyp (Saaltyp), den Tickettyp (Tickettyp), den Sitztyp (Sitztyp) sowie den Netto- (PreisNetto) und den Brutto-Preis (PreisBrutto).

Die Tabelle **bewertet** speichert die Bewertungen, die Kunden den Filmen gegeben haben. Jede Bewertung ist durch eine Buchungskennung (BuchungsID) und eine Filmkennung (FilmID) eindeutig identifiziert und enthält das Bewertungsfeld (Bewertung), das die Bewertung des Kunden für den Film speichert.

6. Schwierigkeiten

Zu Beginn des Projekts gab es Schwierigkeiten beim Zugriff auf den MySQL-Server und damit auf die Datenbank. Das Problem entstand, weil zwei Server parallel auf demselben Port gestartet wurden, einmal durch *„ng serve“* und einmal durch *„node server.js“.* Dadurch konnten die Daten aus der Datenbank nicht ordnungsgemäß abgerufen werden. Schließlich wurde das Problem behoben, sodass die Anwendung nun nur noch über *„node server.js“* läuft.

7. Ausblick

Das vorliegende Projekt bietet Potenzial für zukünftige Erweiterungen, die jedoch den Rahmen dieser Projektarbeit überschritten hätten und daher nicht umgesetzt wurden. So könnten beispielsweise Movie-Points bei Buchungen gesammelt werden, die ab einer bestimmten Anzahl ein vergünstigtes Kinoticket ermöglichen. Zudem könnten Rabatt-Codes oder Gutscheine eingeführt werden.

Des Weiteren wäre eine Mitarbeiter- bzw. Admin-Rolle denkbar, die eine Übersicht über gebuchte Filme und Tickets sowie die Verwaltung der Filme, Vorführungen und Säle ermöglicht. Dies könnte das Hinzufügen und Löschen der genannten Entitäten sowie das Abrufen von Statistiken zu gebuchten und bewerteten Filmvorführungen bzw. Filmen umfassen, um einen besseren Überblick und mehr Verwaltungsspielraum zu bieten.

8. Ausführen der Webanwendung

Die Anwendung wird in der Entwicklungsumgebung IntelliJ IDEA geöffnet. Nach dem Öffnen des Projekts kann die Anwendung über das Terminal gestartet werden. Dafür sollten zunächst der Befehl *„ng build“* und anschließend der Befehl *„node server.js“* ausgeführt werden. Dabei muss man mit dem VPN vpn.hs-bremen.de verbunden sein und seine Hochschulzugangsdaten verwenden. Die Anwendung kann dann im Browser über den Link http://localhost:4200/ aufgerufen werden.

Zum Testen der Anwendung wurde der folgende Benutzer angelegt:

- E-Mail: benjamin.tannert@example.com
- Passwort: bt

Quellenangaben

[1] **W3Schools Angular Tutorial:**
W3Schools. (n.d.). *Angular Tutorial*.
Retrieved from W3Schools

[2] **Fetch and Show Data from MySQL Database in Node.js:**
PositronX. (2023). *How to Fetch & Show Data from MySQL Database in Node.js*.
Retrieved from PositronX

[3] **ChatGPT:**
OpenAI. (2024). *ChatGPT: Conversational Model*.
Retrieved from https://www.openai.com/chatgpt

[4] **Udemy:**
Udemy (2024). *Angular - The Complete Guide*
Retrieved from https://www.udemy.com/course/the-complete-guide-to-angular-2/learn/lecture/14466450?start=0#overview