

Technische Hochschule Köln, Fakultät für Informatik und Ingenieurwissenschaften

Datenbanken 2 Praktikum SS2022 MS01 Gruppe_47

Projektdokumentation

Projektname:

Bücherwurm

Ein interaktiver online Buchhandel

Ein Datenbankenprojekt von:

Dennis Kliewer (AI)

Jesaja Storm (AI)

Caner Özer (WI)

Leo Petersberg (WI)

Teamleiter:

Sven Heiter (AI)

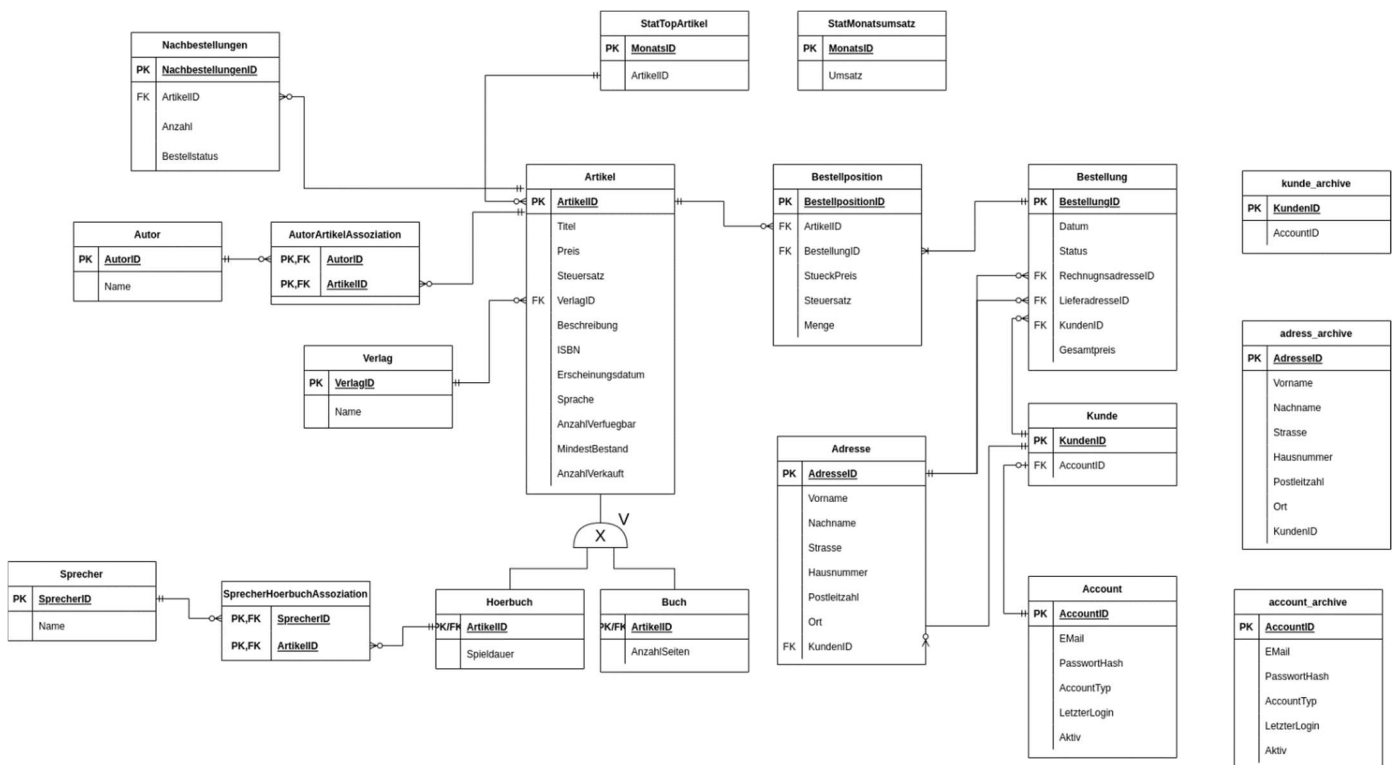
Betreuerin:

Frau Petra Riemer

Projektidee

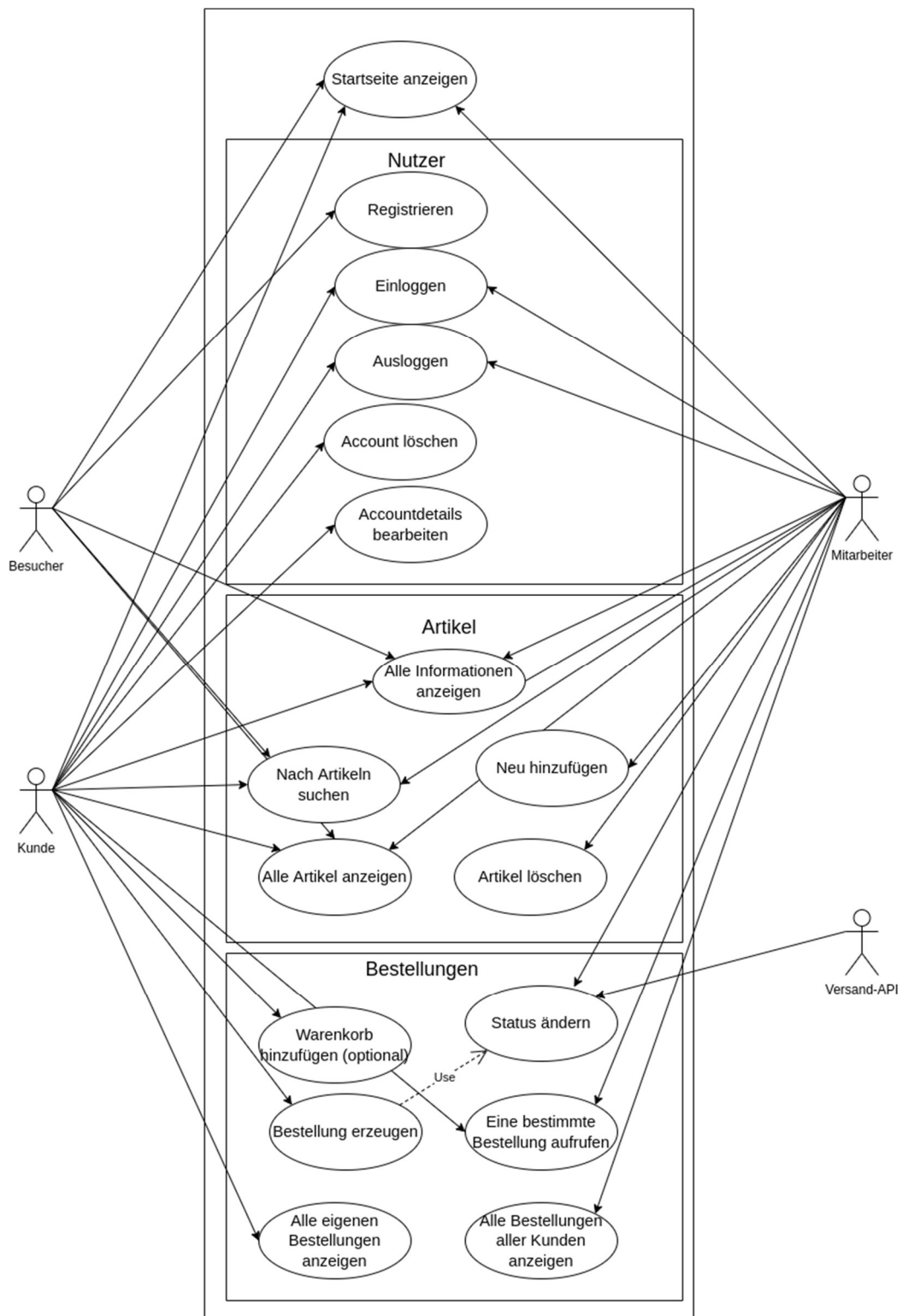
Mit „Bücherwurm“ verfolgen wir die Idee der Realisierung eines Online Buchhändlers in Form einer HTTPS Webseite. Auf dieser soll Kunden ermöglicht werden ganz einfach und bequem auf ihren Endgeräten Bücher, sowie Hörbücher bestellen zu können. Dabei kann jeder sich das Angebot unseres Sortiments anschauen und gezielt nach ihrem Wunschartikel suchen. Um einen Artikel zu kaufen ist es notwendig sich mit seinem selbsterstellten Benutzerprofil anzumelden. Über das personalisierte Benutzerportal bekommen Kunden Einsicht zu Details ihrer aktuellen und bereits abgeschlossenen Bestellungen. Aber auch ****Datenbankadministratoren/dem Personal**** soll die Benutzung des Systems leichtgemacht werden. Deshalb werden sowohl Datenbanktrigger, als auch Funktionen zum Einsatz kommen, um Prozesse, wie den Such- und Bestellvorgang des Kunden, als auch die Werksseitige, an der Marktnachfrage orientierten, Nachbestellung von Artikeln größtenteils zu automatisieren. Mehr zu diesen erfolgt in der weiteren Projektdokumentation.

Erweitertes Entity-Relationship-Model



Unser Use Case Diagramm

Wie sie ausfolgendem Diagramm entnehmen können, haben wir uns für 3 Benutzergruppen entschieden. Zum einen der Besucher der Bücherwurmseite, der unser Sortiment durchstöbern kann, dann der Kunde, der letzteren um die Kauffunktion erweitert und letztlich der Mitarbeiter, der volle



Einsichtnahme genießt, jedoch nicht kaufen kann.

Datenbanktrigger- und Funktionen

Art	Name in der Datenbank	Funktionalität	Erarbeitet von
Trigger	add_price_to_order	Den Preis einer Bestellposition zur Bestellung hinzufügen BEI: Erstellung einer Bestellposition	Jesaja
Trigger	add_to_stock_order	Waren werden der Nachbestellungstabelle hinzugefügt, um wieder über (+20%) den Mindestbestand gehoben zu werden BEI: zu kleiner Stückzahl durch einen Kauf	Caner
Trigger	account_logging	Gibt alle Inserts und Updates auf der Konsole aus BEI: Erstellen oder Bearbeiten der Tabelle Account	Leo
Trigger	reduce_stock	Den Bestand bei Bestellung verringern BEI: Aufgeben einer Bestellung (Bestellstatus von „editierbar“ „offen“)	Sven
Trigger	archiveDeletedAccount	Von Benutzern entfernte Accounts werden archiviert BEI: Löschung eines Accounts durch den Kunden	Dennis
Funktion	adjust_minimum_stock	Mindestbestand der Artikel anhand der letzten Verkaufszahlen des letzten Monats erhöhen/vermindern	Jesaja
Funktion	create_surrogate_key	Alle Kunden mit offener Bestellung ausgeben lassen	Caner
Funktion	send_emails	Sendet Marketing E-Mails an Kunden BEI : langer Inaktivität	Leo
Funktion	show_period_revenue	Den Umsatz eines bestimmten Zeitraumes zurückgeben	Sven
Funktion	show_statistics	Statistiken ausgeben lassen	Dennis
View	Buchinformationen	Gibt alle Daten rund um ein Buch aus (Artikel, Buch, Autor, Verlag)	Jesaja

Mögliche zukünftige Features:

-Warenkorb

-View für alle Details rund um ein Hoerbuch

Gefundene Probleme und ihre Lösung:

Problem 1:

Sobald wir eine Bestellung erzeugt haben, wurde sie direkt als „offen“ markiert. Dadurch konnten wir nicht zwischen einer Bestellung unterscheiden, der noch Bestellpositionen hinzugefügt werden sollen und einer Bestellung, die tatsächlich aufgegeben wurde.

Lösung zu Problem 1:

Die Einführung des Status „editierbar“, dadurch konnten wir Bestellungen erkennen, die noch vom Kunden verändert werden und solchen die von Kunden aufgegeben wurden.