|  |  |  |
| --- | --- | --- |
| AU-Test Engineering Apprentice Managemen | [Titel des Dokuments] | REV 000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Verfasser: | Mueller Kilian | Datum: | 15.03.2023 |

Projektdokumentation

# Aufgabenstellung

Die Aufgabe war eine Anwendung für einen Skiverleih zu programmieren. Man soll den Ausleih-   
und Retourbuchungen anwenden können und diese anzeigen.

# Ansatz

Ich habe mich für die Entwicklung in Blazor entschieden.   
Es soll eine Lager-Seite geben, die alle Artikel im Lager anzeigt, dazu kommt noch, dass man sieht welcher Artikel momentan verfügbar oder bereits ausgeliehen ist.

Die nächste Seite zeigt alle Leihbuchungen an mit Artikel und Kunden. Über einen Button kann man eine neue Buchung erstellen. Beim Drücken des Buttons wird man auf die Erstellungs-Seite weitergeleitet. Hier kann man den Kunden und den Artikel für die Buchung auswählen. Nachdem hinzufügen, wird man wieder auf die Buchungs-Seite weitergeleitet.

# Datenbank

Für die Datenbank werden folgende Tabellen erstellt:

* Articles
* Customers
* Employees(AspNetUsers)
* Categories
* LoanStatuses
* OnLoans

  
BILD VON DATENBANKDIAGRAM

Articles: Artikel im Sortiment

Customers: bereits eingetragene Kunden

Employees: Mitarbeiter, die das Programm bedienen

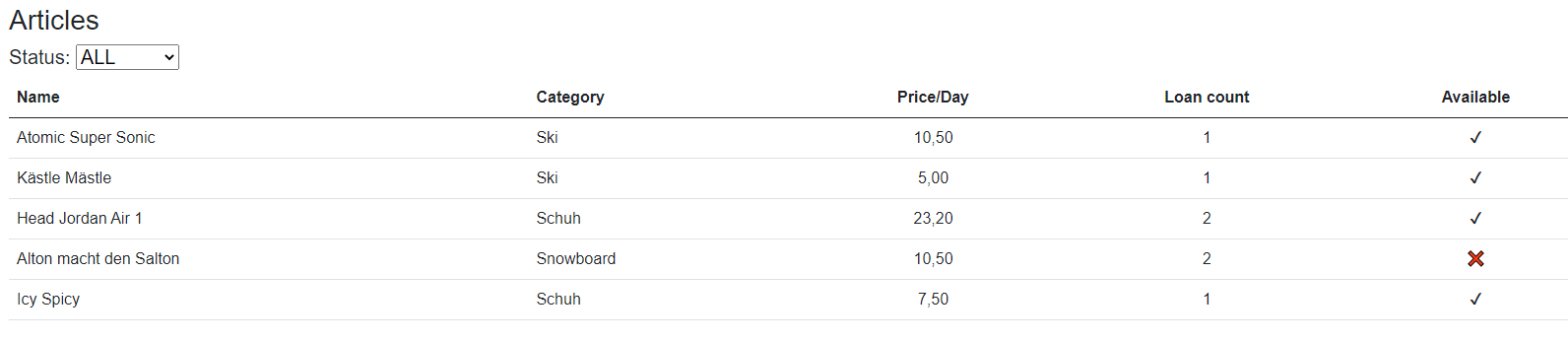
Categories: Kategorien für Artikel

LoanStatuses: Welcher Status der Artikel hat(verliehen/verfügbar)

OnLoans: Leihbuchungen

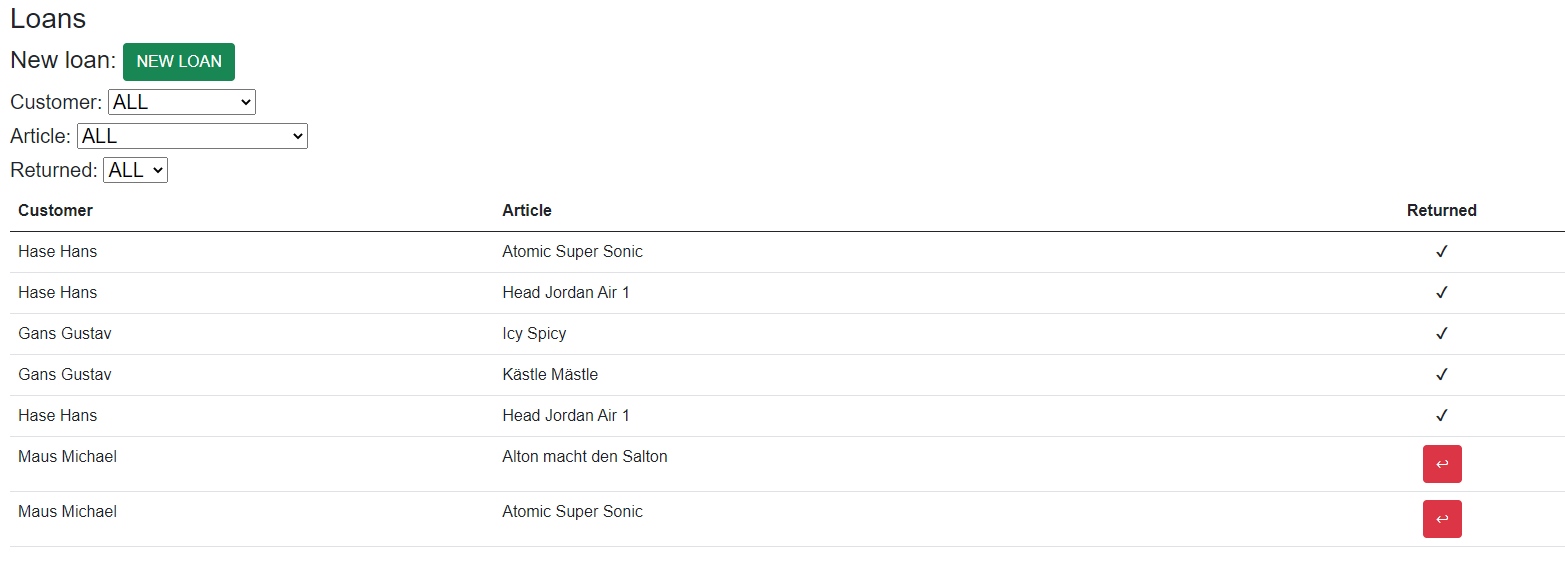
# UserInterface

**Artikel-Seite:**



Zeigt den Namen, Kategorie des Artikels, Leihpreis p. Tag, Verleihungshäufigkeit und Verfügbarkeit an.  
Mit dem Dropdown kann man nach der Verfügbarkeit filtern. Wenn man z.B. „Available“ auswählt, werden nur die verfügbaren Artikel ausgewählt.

**Buchungs-Seite:**



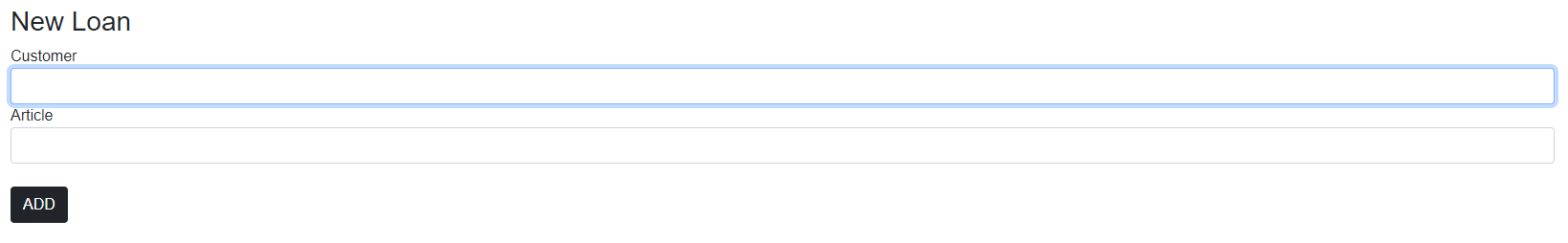
Zeigt den Kunden, der den Artikel ausleiht, den ausgeliehen Artikel und ob dieser bereits zurückgegeben wurde oder nicht.

Über die Drop-Downs kann man nach den Kunden bzw. Arikel und Status filtern.

Wenn man auf den Return-Button klickt, wird die Leihbuchung auf Zurückgegeben gesetzt.

Wenn man auf den „New-Loan“-Button klickt wird man auf die Erstellungsseite weitergeleitet.

**Erstellungsseite:**



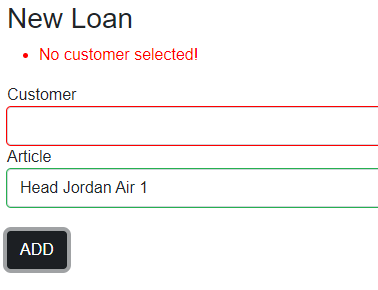
Hier kann man über Drop-Downs den jeweiligen Kunden und Artikel auswählen, beim Betätigen des „ADD“-Buttons wird eine neue Leihbuchung erstellt und man wird wieder auf die Buchungs-Seite weitergeleitet. Hier wird die neue Buchung angezeigt.

# Funktions-Test

**Erstellen von neuer Leihbuchung:**

1. **Kunde nicht ausgewählt**

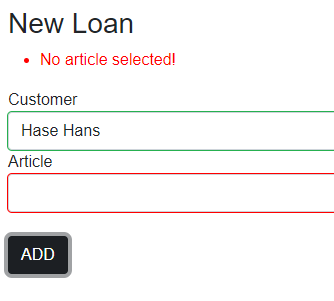
Buchung wird nicht hinzugefügt und eine Fehlermeldung erscheint.



🡪Test-Erfolgreich: Fehlermeldung kommt und Buchung wurde nicht hinzugefügt

1. **Artikel nicht ausgewählt**

Buchung wird nicht hinzugefügt und eine Fehlermeldung erscheint.



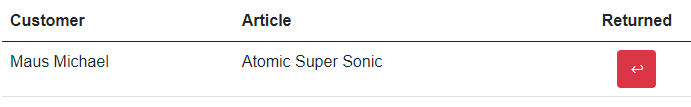
🡪Test-Erfolgreich: Fehlermeldung kommt und Buchung wurde nicht hinzugefügt

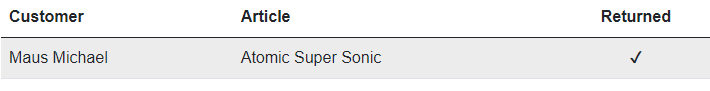
**Buchungsseite:**

1. **Leihbuchung zurückgeben**

Wenn man auf den Return-Button klickt, sollte die Leihbuchung auf Returned gesetzt werden:

Vorher:

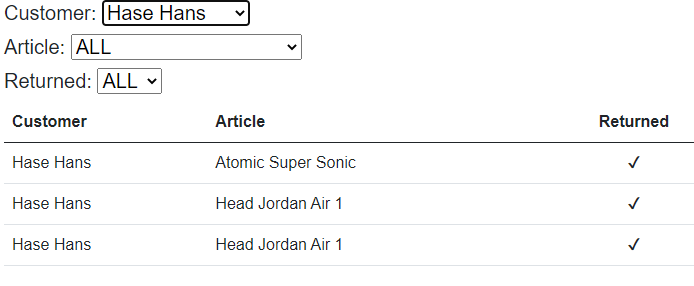


Nachher: 

🡪Test-Erfolgreich: Buchung ist zurückgegeben

1. **Nach Kunden filtern**

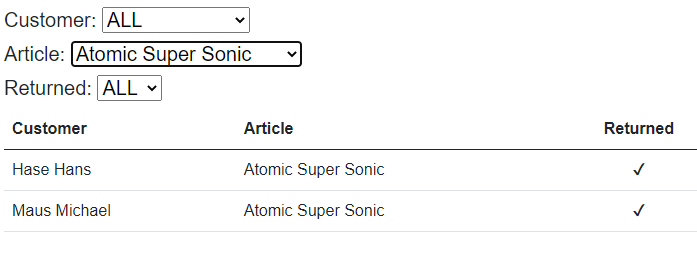
Beim Auswählen eines Kunden sollten nur Buchungen von diesem angezeigt werden



🡪Test-Erfolgreich: Nur Buchungen des ausgewählten Kunden werden angezeigt

1. **Nach Artikel filtern**

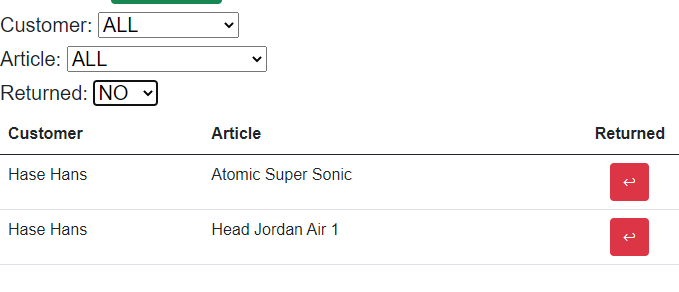
Beim Auswählen eines Artikel sollten nur Buchungen mit diesem Artikel angezeigt werden



🡪Test-Erfolgreich: Nur Buchungen des ausg. Artikel werden ang.

1. **Nach Status filtern**

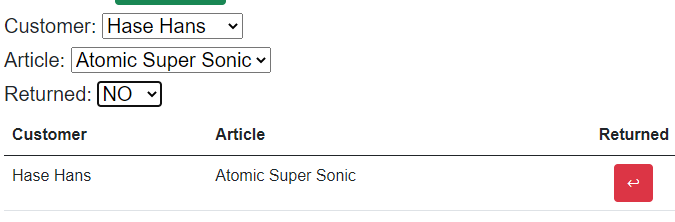
Beim Auswählen des Status der Buchung sollten nur Buchungen mit dem ausgewähltem Status angezeigt werden



🡪Test-Erfolgreich: nur Buchungen mit den ausg. Status werden ang.

1. **Filtern mit Kombination**

Beim Filtern mit mehreren Parametern sollten nur die Buchungen angezeigt werden, die mit den Parametern übereinstimmen.

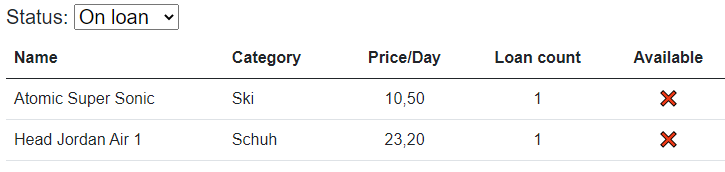


🡪Test-Erfolgreich: nur Buchungen die den ausg. Parametern entsprechen werden ang.

**Lagerseite:**

1. **Filtern nach Status**

Beim Auswählen des jeweiligen Status sollten nur die Artikel angezeigt werden die, den Status haben.

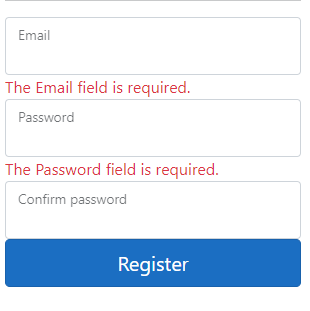


🡪Test-Erfolgreich: nur Buchungen mit den ausg. Status werden ang.

**Register:**

1. **Leerer Input**

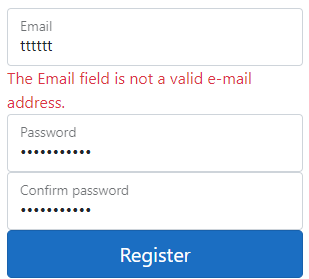
Wenn man sofort auf Register drückt, sollten mehrere Fehlermeldungen angezeigt werden.



🡪Test-Erfolgreich: Register schlägt fehl

1. **Falsche E-Mailadresse**

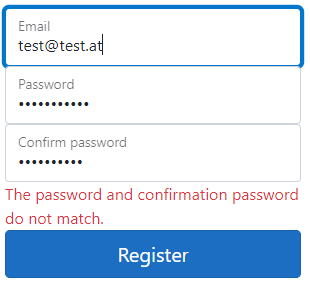
Wenn man eine E-Mailadresse eingibt, die dem Format nicht entspricht sollte, eine Fehlermeldung angezeigt werden.



🡪Test-Erfolgreich: Register schlägt fehl

1. **Nicht übereinstimmende Passwörter**

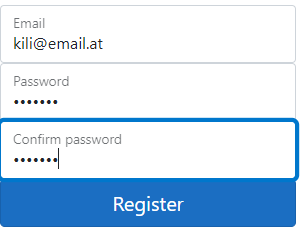
Wenn die Passwörter nicht übereinstimme, sollte eine Fehlermeldung angezeigt werden.

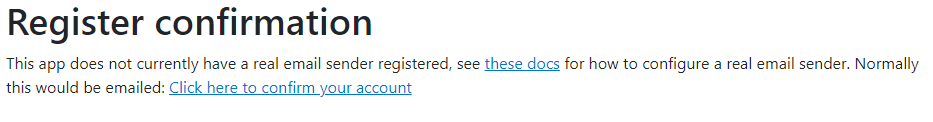


🡪Test-Erfolgreich: Register schlägt fehl

1. **User Regestrierung funktioniert**

Wenn man eine richtige E-Mail und Passwort eingebt und bestätigt

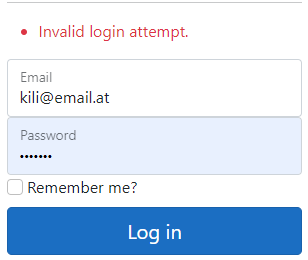




🡪Test-Erfolgreich: Neuer User regestriert

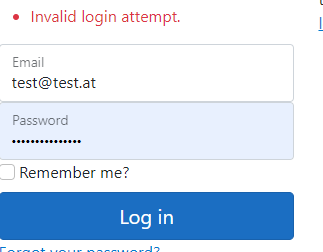
**Login:**

1. **Richtige E-Mail falsches Passwort**



🡪Test-Erfolgreich: Login schlägt fehl

1. **Falsche Email richtiges Passwort von anderem User**



🡪Test-Erfolgreich: Login schlägt fehl

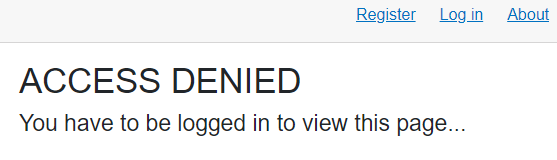
1. **Richtige E-Mail und Passwort**



🡪Test-Erfolgreich: User wird eingeloggt

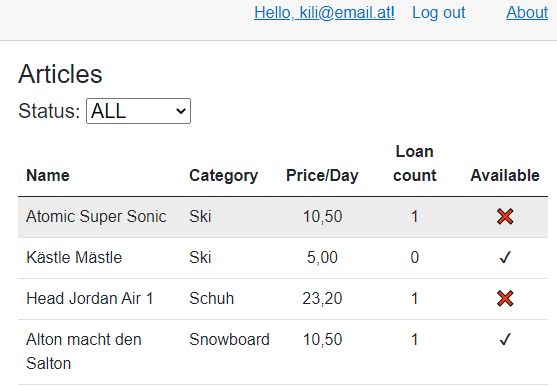
**Seitenauthentifizierung:**

1. **User nicht eingeloggt**



🡪Test-Erfolgreich: Kein Zugang zur Seite

1. **User ist eingeloggt**



🡪Test-Erfolgreich: Zugang auf Seite