

Blu-ray Kreisel



Eckardt, Marco

Wagner, Christoph

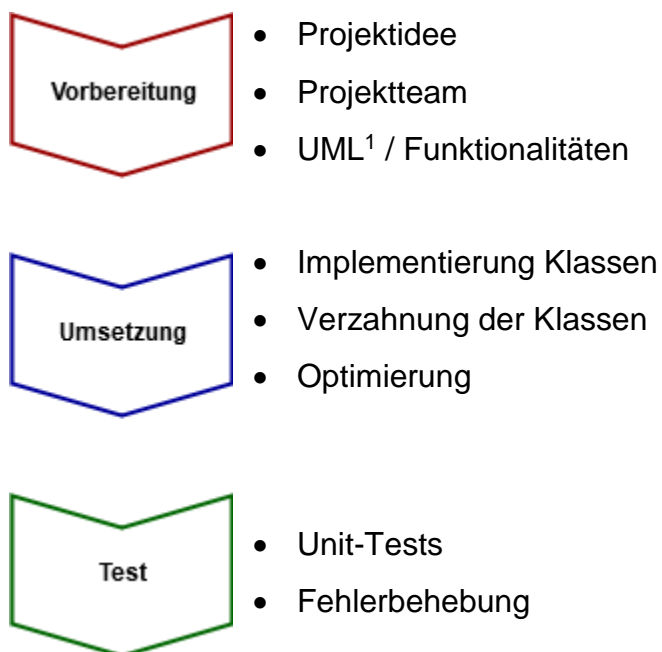
Projektaufgabe

- Logik und Datenmodell für eine Java-basierte Serveranwendung entwickeln
- Wahl eines eigenen Projektthemas
- Team ~4 Personen

Projektteam

- ~~Betcher, Dennis~~
- Eckardt, Marco marco.eckardt@fh-erfurt.de 3. FSBAI
- ~~Huss, Thorsten~~
- Wagner, Christoph christoph.wagner@fh-erfurt.de 3. FSBAI

Projekt Top-Down-Umsetzung



¹ Unified Modeling Language (vereinheitlichte Modellierungssprache)

Inhaltsverzeichnis

Projektaufgabe.....	1
Projektteam.....	1
Projekt Top-Down-Umsetzung.....	1
Inhaltsverzeichnis	2
Die Projektidee.....	3
Struktur: textuelle Beschreibung	4
Struktur: UML - Diagramm	5
Struktur: Javadoc	5
Installationshinweise	6
Ausblick in die Zukunft	6

Die Projektidee

Ziel des Projektes ist die Bereitstellung einer Plattform zum Austausch von Blu-rays unter registrierten Benutzern.

Die Plattform funktioniert auf der Grundlage von Tauschgeschäften ohne den Einsatz einer Währung.

Folgende Funktionen soll die Anwendung bereitstellen:

- Benutzer können sich registrieren
- Benutzer können sich anmelden
- Benutzer können Blu-rays erfassen die getauscht werden sollen
- Benutzer können sich den Blu-ray- Bestand anzeigen lassen
- Benutzer können im Blu-ray- Bestand suchen
- Benutzer können Blu-rays in den Warenkorb legen
- Benutzer können Tauschgeschäfte ausführen
- Benutzer können sich für den Newsletter registrieren

Struktur: textuelle Beschreibung

Die Anwendung wurde im objektorientierten Stil umgesetzt. Diese Art der Implementierung ermöglicht es sich an den Realobjekten und deren Eigenschaften zu orientieren.

Folgende Klassen wurden hierfür angelegt:

- BRK → Blu-ray Kreisel
- CCustomer → Benutzer
- CBluRay → Blu-ray disc
- CBasket → Warenkorb
- CNewsletter → Newsletter

Die Klasse BRK dient als reine Logikklasse und hält somit alle nicht-trivialen Funktionen, die nicht genau einer der anderen Klassen zugeordnet werden können.

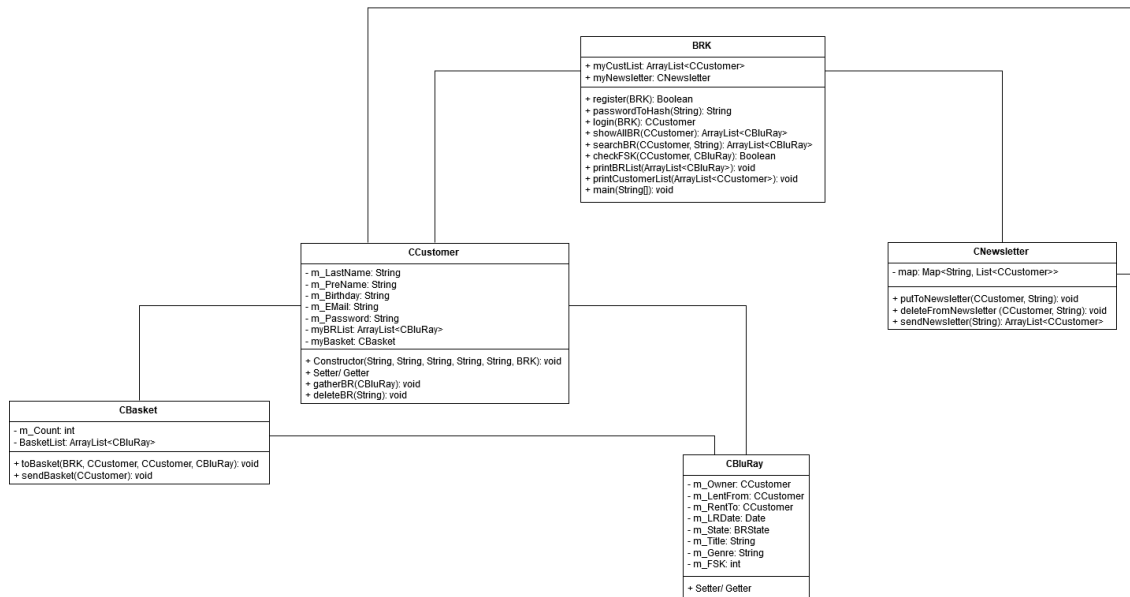
Die Klassen CCustomer, CBluRay, CBasket, CNewsletter sind vereinfachte Abbilder der Realobjekte.

Zur Implementierung der notwendigen Funktionalitäten sind die Klassen miteinander, unter Einhaltung der Schnittstellen, verzahnt.

Struktur: UML - Diagramm

Die Verzahnung der einzelnen Klassen lässt sich aus dem angefügtem UML – Diagramm ableiten.

Des Weiteren sind die Eigenschaften sowie die nicht-trivialen Funktionen der einzelnen Klassen erkennbar.



Struktur: Javadoc²

Durch Einhaltung der Javadoc Standards im Quellcode sind folgende HTML-Dokumentationsdateien entstanden.



javadoc.7z

² Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt

Installationshinweise

Die volle Programmfunktionalität ist von einigen Faktoren abhängig.

So lässt sich die Anwendung nur von registrierten Benutzern ausführen, da diese eine Art Benutzersitzung initiieren.

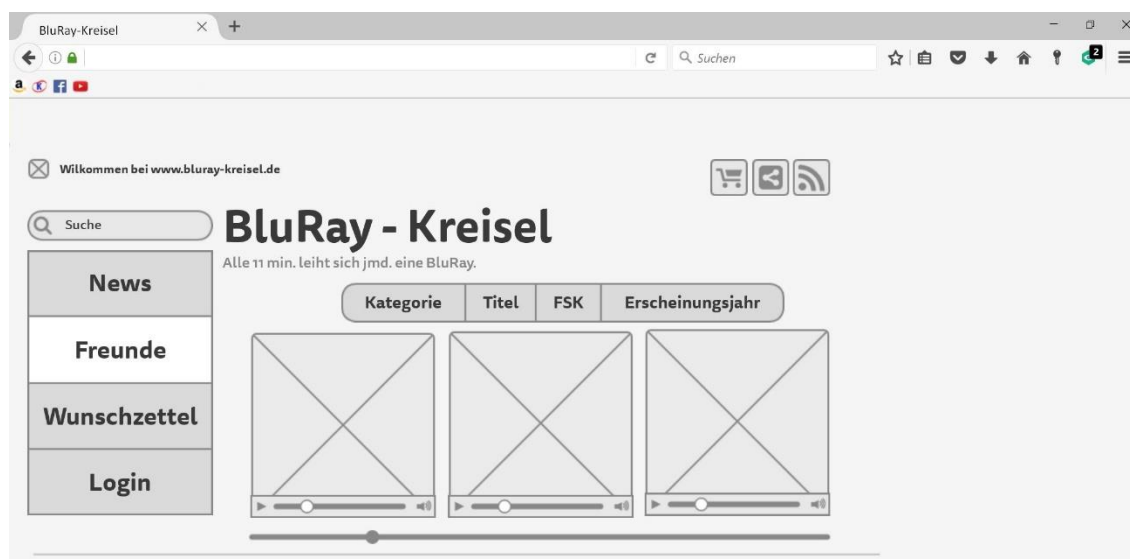
Es sollten jedoch zusätzliche Benutzer in der Anwendung vorhanden sein, so dass eine Suche nach Blu-rays sinnvoll ist.

Ausblick in die Zukunft

Der derzeitige Projektstand lässt den Funktionsumfang gut erkennen.

Allerdings werden weitere Funktionen benötigt um zum Beispiel eine Datenbankanbindung umzusetzen oder eine GUI³ zu erstellen und somit die Interaktion mit dem Benutzer zu ermöglichen.

Erste Mockups zur Umsetzung der GUI existieren bereits.



³ Grafische Benutzeroberfläche