

Software-Qualitätssicherung

Übung 1

Inhaltsverzeichnis

0 Disclaimer	2
0.1 Angabe	2
0.2 Allgemeine Informationen	2
0.3 Benennung der Abgaben & Kompilierbarkeit	2
0.4 Verwendung von Git	2
0.5 Entwicklungsumgebung	2
1 Anforderungsanalyse und Review	3
1.1 Review der Anforderungen	3
1.1.1 Typ	3
1.1.2 Überprüfbarkeit (aus Sicht des Auftraggebers)	3
1.1.3 Rückverfolgbarkeit	3
1.2 Design Review der Anwendungsfälle	4
1.3 Risk Based Testing	4
1.3.1 Quality/Feature Matrix	4
1.3.2 Risk-Profile	4
1.4 Planen des Reviewprozesses	5
1.5 Abgabe	5
2 Projektauftrag	6
2.1 Ausgangssituation	6
2.2 Projektbeschreibung	6
2.3 Rollen	7
2.4 Domänenmodell	7
2.5 Anforderungen	8
2.6 Anwendungsfallbeschreibung	9
2.7 Anwendungsfalldiagramme	11

0 Disclaimer

0.1 Angabe

Die Übung ist als aufbauende Übung ausgelegt. Verwenden Sie daher für alle Beispiele den Projektauftrag aus Übung 1 sowie alle bisherigen Übungsergebnisse.

0.2 Allgemeine Informationen

Bei der Übung handelt es sich um eine Einzelarbeit. Alle Abgaben werden von uns auf Plagiate überprüft. Generell werden Plagiate mit 0 Punkten bewertet. Weiters behalten wir es uns vor, Studierende im Verdachtsfall zu einem verpflichtenden Kontrollgespräch einzuladen.

Bedenken Sie, dass wenn sehr viele Studierende gleichzeitig auf das TUWEL zugreifen, der Server überlastet werden kann und es zu längeren Upload-Zeiten bei Ihren Abgaben kommen könnte. Lassen Sie sich deshalb in Ihrem eigenen Interesse nicht bis zur letzten Minute mit Ihrer Abgabe Zeit.

Verspätete Abgaben werden ausnahmslos nicht akzeptiert!

0.3 Benennung der Abgaben & Kompilierbarkeit

Bitte verwenden Sie, sofern mitgeliefert, die Vorlage und beachten Sie, dass ausschließlich richtig benannte Dateien im angegebenen Format bewertet werden.

Achten Sie bei der Benennung der Dateien auch darauf, dass danach noch alle Ihre abgegebenen Quelldateien kompilierbar sind! Sollte Ihre Abgabe nicht kompilieren, wird Sie nicht bewertet.

0.4 Verwendung von Git

Einige Beispiele enthalten ein git-Repository. Halten Sie jeden Ihrer Schritte im git-Repository fest. Sie benötigen dazu lediglich folgende Befehle:

```
1  # Add all untracked files
2  git add -A
3  # Commit all changes with the given message
4  git commit -am "<message>"
5  # Edit last commit message
6  git commit --amend -m "<new_message>"
7  # Show commit log
8  git log
9  # Checkout a specific revision
10 git checkout <revisionHash>
```

Denken Sie immer daran aussagekräftige Commit Messages zu wählen!

Achten Sie darauf, dass Sie am Ende ihrer Arbeit immer alles auf dem master-Branch liegen haben und squashen Sie keine Commits.

Nähere Informationen zur Verwendung von Git sowie den Download erhalten Sie unter:
<https://git-scm.com/>

0.5 Entwicklungsumgebung

Das Beispiel ist auf die Verwendung von IntelliJ IDEA (<https://www.jetbrains.com/idea/>) ausgelegt. Die Entwicklungsumgebung steht Studierenden, unter <https://www.jetbrains.com/student/>, kostenfrei zur Verfügung.

Die Verwendung von IntelliJ IDEA als Entwicklungsumgebung ist nicht zwingend erforderlich. Sie können alternativ auch Eclipse (<https://eclipse.org/>) verwenden oder mit einem Codeeditor wie zum Beispiel Atom (<https://atom.io/>) arbeiten.

1 Anforderungsanalyse und Review

Sie wurden als Mitarbeiter eines mittelständischen Softwareunternehmens mit der Umsetzung einer Applikation für einen kleinen Filmverleih betraut. Der Kunde hat bereits einen Projektauftrag geschrieben, in dem sich leider ein paar kundentypische Fehler eingeschlichen haben. Bevor Ihre Firma mit der Umsetzung beginnen kann, muss der Projektauftrag einem gründlichen Review unterzogen werden.

1.1 Review der Anforderungen

In diesem Teil der Aufgabe sollen Sie die im Projektauftrag definierten Anforderungen systematisch prüfen. Bestimmen Sie zudem den Typ, die Überprüfbarkeit (Messbarkeit & Testbarkeit) sowie die Rückverfolgbarkeit der Anforderungen.

1.1.1 Typ

- Functional - Features, Fähigkeiten, Sicherheitsvorrichtungen
- Usability - Menschliche Faktoren
- Reliability - Häufigkeit eines Ausfalls, Vorhersehbarkeit, Wiederherstellbarkeit
- Performance - Antwortzeiten, Durchsatz
- Supportability - Anpassungsfähigkeiten, Wartung, Konfigurationen

1.1.2 Überprüfbarkeit (aus Sicht des Auftraggebers)

- Ist die Anforderung durch z.B. einen Test verifizierbar?
- Kann der Test bereits vor der Implementierung realisiert werden?
- Ist das Ergebnis des Tests leicht festzustellen? Messbar? Auch ohne Software?
- Manuell oder auch automatisiert testbar?
- Wenn ja, wie? Wenn nein, warum nicht? Änderungsvorschläge festhalten.

Mögliche Ergebnisse:

- Ja, aber in der Beschreibung steht "X": Widerspruch!
- Ja, durch einen Test realisierbar: vor dem Test muss die Datenbank in den Zustand X gebracht werden.
- Nein, "...leicht änderbar." ist nicht testbar. Anforderung wäre besser überprüfbar durch ändern auf "...durch Konfigurationsdatei zur Laufzeit änderbar."
- Ja, jedoch nicht automatisierbar, Test muss manuell an der laufenden Applikation durchgeführt werden.
- Nein, mit Umformulierung: "Das System muss X in ≤ 2 Sekunden ausführen"
- ...

1.1.3 Rückverfolgbarkeit

- Wird die Anforderung in der Projektbeschreibung verlangt? Liegt sie im relevanten Bereich des Systems?
- Anforderungen sollen keine technischen Entwurfs- oder Implementierungsvorschläge enthalten!
- Ist das Verhalten für alle Normalfälle und alle relevanten Fehlerfälle in der Projektbeschreibung spezifiziert?

Mögliche Ergebnisse:

- Nein, die Anforderung ist nicht in der Projektbeschreibung spezifiziert.
- Die Anforderung ist technisch viel zu genau ausgeführt!
- Rückverfolgbarkeit ist gegeben, aber es ist kein Fehlerfall spezifiziert.
- ...

1.2 Design Review der Anwendungsfälle

Jede funktionale Anforderung wird in ein oder mehrere Anwendungsfälle übergeführt. Änderungen in den Anforderungen ergeben Änderungen in den Anwendungsfällen, diese sollen jetzt einem Review unterzogen werden. Die vorhandenen Anwendungsfälle wurden durch die Akteure und deren Operationen, welche in den Anforderungen und der Projektbeschreibung vorhanden sind, gebildet.

Gehen Sie die Anforderungen, wie in der Vorlesung erklärt, systematisch durch und markieren Sie 3 verschiedene Arten von Wörter (dies dient lediglich als Hilfestellung und ist optional):

- Akteure, z.B. Kunde
- Entitäten, z.B. Ticket
- Operationen, z.B. verkauft

Überprüfen Sie jede vorhandene Anwendungsfallbeschreibung auf folgende Qualitätskriterien:

- Ist jeder Anwendungsfall einer Anforderung zugeordnet?
- Ist jeder Anwendungsfall in der Projektbeschreibung zu finden?
- Ist jeder Anwendungsfall notwendig, plausibel, korrekt und vollständig?
- Ist jeder Anwendungsfall mit dem richtigen Akteur verbunden?
- Sind die Assoziationen zu anderen Anwendungsfällen sinnvoll?
- Passen die Diagramme zu den Anwendungsfällen?
- Stimmen die Anwendungsfallbeschreibungen mit den Diagrammen überein?

Mögliche Ergebnisse:

- Anwendungsfall hat keine zugehörige Anforderung.
- Anwendungsfall hat zugehörige Anforderung: 23
- Anwendungsfall ist unvollständig (z.B. Akteur fehlt, weiterer Anwendungsfall ist notwendig).
- Fehler in Schritt 3 der Szenario Beschreibung: Anwendungsfall X existiert nicht
- Fehler in den Vorbedingungen: X ist keine gültige Vorbedingung!
- ...

1.3 Risk Based Testing

In dieser Übungsaufgabe ist zu überprüfen, ob konkrete Testfälle für die wichtigsten Anforderungen (laut Risk-Based-Testing) vorhanden sind.

1.3.1 Quality/Feature Matrix

In dieser Aufgabe geht es um das Einschätzen von Komponenten und das Herausfiltern, welche für den Projekterfolg am riskantesten sind. Bewerten Sie jede Anforderung mit einer Risikoeinschätzung (Quality/Feature Matrix). Welche Risiken gefährden die Anforderungen laut Qualitätsmerkmalen (Nicht-funktionale Anforderungen)?

Risikoeinschätzung:

- **C**ritical
- **I**mportant
- **L**ess important

Tragen Sie Ihre Risikoeinschätzungen in die Matrix ein, verwenden Sie dazu den jeweiligen Anfangsbuchstaben (C, I, L) sowie eine farbliche Kodierung.

Hinweis: Bitte füllen Sie für eine vollständige Bewertung jede Zelle der Matrix aus!

1.3.2 Risk-Profile

- Welche 5 Anforderungen sollten intensiv getestet werden, d.h. welche 5 Anforderungen haben ein großes Risiko-Profil in der Quality/Feature Matrix?
- Kann man jede dieser Anforderungen testen? Wie würden die Negativtests lauten? Geben Sie eine Kurzbeschreibung und den Typ der Testfälle an.

1.4 Planen des Reviewprozesses

Zusätzlich zu den Reviews soll im Zuge der Übung ein Plan eines Reviewprozesses erstellt werden. Hier miteinbezogen sollen werden:

- **Art des Reviews:** Software Requirements Review, Preliminary Design Review, Critical Design Review, In-Process Review, Management Review, Inspection, Code Walkthrough, Technical Review
- **Ressourcen:** Was soll reviewed werden? Wo werden Schwerpunkte gesetzt?
- **Rollen:** Moderator, Leser, Gutachter, Schreiber, Author
- **Zeit:** Zu welchem Zeitpunkt bzw. in welcher Phase des Projektes soll welches Review durchgeführt werden? Wann soll mit der Vorbereitung/Nacharbeit begonnen werden?
- **Kompetenzen:** Welche Rollen, Ressourcen sollen bei welchem Review zur Verfügung stehen?

1.5 Abgabe

Folgende Dateien sind abzugeben:

- QSVU_UEbung1__<Matrikelnummer>__<Nachname>__<Vorname>.pdf

Bitte verwenden Sie, sofern mitgeliefert, die Vorlage und beachten Sie, dass ausschließlich richtig benannte Dateien im angegebenen Format bewertet werden.

Bitte laden Sie ihre Abgabe rechtzeitig hoch. Verspätete Abgaben werden ausnahmslos nicht akzeptiert! Beachten Sie unbedingt den **Disclaimer**.

2 Projektauftrag

2.1 Ausgangssituation

Der Filmverleih "Quality Blockbusters" erlebt gerade eine kleine Renaissance und darf sich über regen Kundenzuwachs freuen. Da das Unternehmen bisher sehr klein war, wurden Kundendaten sowie die zum Verleih stehenden Filme mittels analoger Karteikarten verwaltet. Außerdem war bisher ein Mitarbeiter für alle Tätigkeiten verantwortlich. Durch die rasche Expansion des Filmverleihs wird die weitere händische Abarbeitung der Aufgaben als zu aufwendig angesehen. Aus diesem Grund hat sich die Geschäftsführung entschieden, das Kundenverwaltungs- sowie das Filmverwaltungs- und das Verleihsystem mit einer Applikation zu managen. Zudem sollen mehrere neue Mitarbeiter eingestellt werden, die wahrscheinlich jeweils unterschiedliche Berechtigungen in der neuen Applikation benötigen.

2.2 Projektbeschreibung

Es wird eine Applikation benötigt, mit deren Hilfe es den Mitarbeitern des Unternehmens möglich ist, Kunden sowie Filme zu verwalten und Verleihvorgänge durchzuführen. Das System soll auch einfache Buchungsbelege (mit Kundennummer, Name, Datum und Buchungsvorgang) ausgeben können.

Im System werden verschiedene Benutzerrollen unterschieden. Jeder Mitarbeiter kann im System Filme und Kunden anzeigen und die jeweiligen Daten einsehen. Kunden können sich bei einem Mitarbeiter registrieren und ab dann jederzeit Filme ausleihen. Administratoren haben uneingeschränkte Rechte im System. Sie können auch die Stammdaten der Filme verwalten. Der Kassier ist für den Ausleihvorgang verantwortlich, er kann Rechnungen erstellen und Filme an den Kunden verleihen. Die letzte Rolle ist der Controller. Er soll die Möglichkeit haben, uns wichtige Informationen über einzelne Filme (z.B. wie rentabel sie sind) zu geben. Zudem kann der Controller statistische Informationen über Kunden einsehen. Der Kunde selbst hat in diesem System keine Rechte. Im ersten Schritt muss noch kein Berechtigungskonzept umgesetzt sein. Am wichtigsten ist es, dass es möglichst schnell möglich ist, Daten zu erfassen und Filme an Kunden auszuleihen.

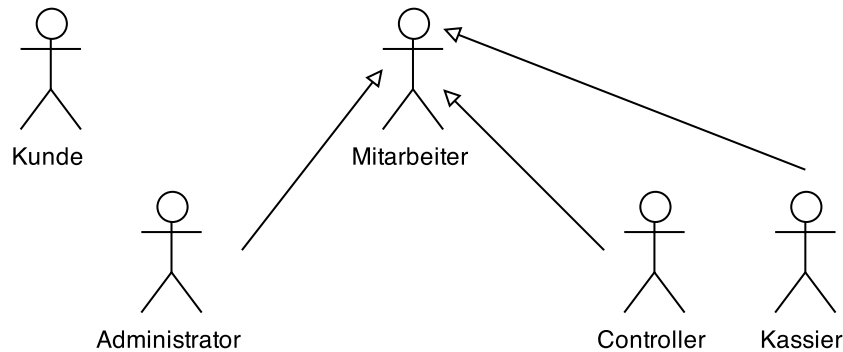
Die Kosten für das Ausleihen der Filme lassen sich einfach berechnen. Jeder Film hat einen Basispreis den dieser Film pro Tag kostet. Zudem werden Filme unterschiedlichen Genres zugeordnet. Je nach Genre werden die Filme um einen bestimmten Faktor teurer oder billiger. Derzeit verleihen wir Filme folgender Genres: Klassiker (x0.9), Kinderfilme(x0.75), Horror (x1.1), SciFi (x1.15) und Fantasy (x1.25).

Die Kunden sammeln im Laufe der Zeit sogenannte "Videopoints". Durch diese ist es möglich, einen Rabatt zu erhalten. Sobald ein Kunde mehr als 10 Videopoints hat, erhält er einen Rabatt von 2% auf die aktuelle Rechnung. Hat ein Kunde mehr als 20 Videopoints, erhält er einen Rabatt von 5% auf die aktuelle Rechnung. Nachdem die Rechnung bezahlt ist, müssen dem Kunden natürlich die entsprechende Anzahl an Videopoints abgebucht werden. Videopoints erhält man für das Ausleihen von Filmen. Immer wenn eine Rechnung bezahlt wird, werden dem Kunden pro 2 € Umsatz 1 Videopoint gutgeschrieben, Stammkunden wird weiters ein Videopoint pro Buchung gutgeschrieben. Es ist dabei unerheblich ob der Kunde bei dieser Bestellung Videopoints verwendet hat oder nicht. Zusätzlich zu dem Rabatt, den die Kunden über das Einlösen der Videopoints erhalten, erhalten Stammkunden immer einen Rabatt von 1.5%.

Da es immer wieder vorkommt, dass Kunden nach Empfehlungen fragen, muss es sehr einfach sein, den kompletten Datenbestand zu filtern und zu sortieren. Außerdem soll es möglich sein, bei jedem Film eine Bewertung von 0.0 (Sehr schlecht) bis 5.0 (Sehr gut) abzuspeichern. Diese Bewertung wird von unseren Mitarbeitern händisch eingegeben, es soll aber auch die Möglichkeit geben, für einzelne Filme die Bewertung aus einer Onlinedatenbank zu laden. Am besten wäre es, wenn die Datenbank von <https://www.themoviedb.org/> verwendet werden würde. Unter Umständen soll die Datenbank aber später getauscht werden können.

Die gesamte Applikation soll möglichst benutzerfreundlich gestaltet werden, das heißt, dass jede Kernfunktionalität (Anlegen/Bearbeiten von Kunden, Anlegen/Bearbeiten von Filmen) mit maximal 3 Mausklicks erreichbar sein muss. Außerdem muss auf dem Buchungsbeleg eine klar erkennbare Abgrenzung zwischen Gesamtkostenberechnung, Videopoints und Rabatten herrschen. Auch soll klar erkennbar sein, welche Kosten pro Filme angefallen sind und welche Filme ausgeborgt oder zurückgegeben wurden. Auch soll es grafisch aufbereitete Fehlermeldungen geben, wenn der Benutzer durch die ausgeführte Aktion Daten verlieren könnte oder ein Fehler im Programm auftritt (Onlineservice für Bewertungen ist nicht erreichbar, Datenbankverbindung schlägt fehl, usw.).

2.3 Rollen



1) Administrator

Administratoren haben alle Rechte im System. Ihre Hauptaufgabe ist es, neue Filme anzulegen und die Stammdaten existierender Filme zu bearbeiten. Der Administrator hat auch die Möglichkeit Stammkunden zu ernennen und Verleihvorgänge abzuwickeln.

2) Kassier

Der Kassier ist der Mitarbeiter der im Ladengeschäft steht. Er ist die Schnittstelle zwischen der Applikation und dem Kunden. Er legt neue Kunden, an bearbeitet die Stammdaten von existierenden Kunden und führt Verleihvorgänge durch.

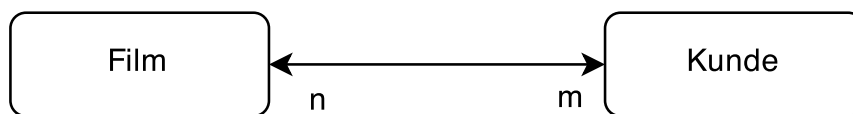
3) Controller

Der Controller kann statistische Auswertungen über Filme erstellen, um dann gegebenenfalls die Preise der Filme anzupassen. Auch muss er die Kundendaten einsehen können, um gute Kunden zu Stammkunden machen zu können.

4) Kunde

Der Kunde leiht bei unseren Mitarbeitern Filme aus oder bittet diese seine gespeicherten Daten zu ändern. Auch kann er die Mitarbeiter nach Empfehlungen fragen oder Sie nach bestimmten Filmen suchen lassen.

2.4 Domänenmodell



1) Filme

Filme haben neben einer eindeutigen Nummer zur Identifizierung auch einen Titel und gegebenenfalls einen Subtitel. Wir führen ausschließlich Filme in den Sprachen Deutsch, Englisch, Französisch und Spanisch. Zudem wird bei jedem Film das Genre und die Altersfreigabe gespeichert. Auch das Erscheinungsjahr, der Regisseur und das Jahr der Veröffentlichung müssen zu dem jeweiligen Film gespeichert werden.

2) Kunden

Kunden werden anhand einer eindeutigen Nummer identifiziert. Es wird ihr Vor- sowie ihr Nachname gespeichert. Zur Kontaktaufnahme muss es möglich sein, Telefonnummer, E-Mail-Adresse und Anschrift des Kunden im System zu hinterlegen. Zur Überprüfung der Altersfreigabe muss aus rechtlichen Gründen auch das Geburtsdatum des Kunden gespeichert werden. Die aktuellen Videopoints sowie der Stammkundenstatus werden ebenfalls beim Kunden vermerkt.

2.5 Anforderungen

1. **Anforderung: Anmeldung**
Mitarbeiter (Administratoren, Kassiere, Controller) sollen sich im System anmelden können.
2. **Anforderung: Kunden Stammdaten**
Es sollen Kunden im System angelegt und bearbeitet werden können.
3. **Anforderung: Film Stammdaten**
Es sollen Filme im System angelegt und bearbeitet werden können.
4. **Anforderung: Ausleihvorgang**
Es soll möglich sein, im Zuge einer Buchung mehrere Filme auszuleihen und zurückzugeben.
5. **Anforderung: Buchungsbeleg**
Es soll bei jedem Verleihvorgang ein Buchungsbeleg (Zeitpunkt, Kundenname, Kundennummer, Gesamtkostenabrechnung, Videopoints, Rabatte, Ausleihen und Rückgaben) erstellt werden.
6. **Anforderung: Barcodescanner**
Das Suchen von Kunden und Filmen während des Ausleihvorgangs soll, zur Vereinfachung, mittels Barcodescanner und Barcodes auf Filmen und Entleihausweis funktionieren.
7. **Anforderung: Stammkunden anlegen**
Kunden sollen von Mitarbeitern zu Stammkunden gemacht werden können. Stammkunden erhalten einen Rabatt von 1.5% auf jede Buchung.
8. **Anforderung: Kostenberechnung nach Tagen**
Die Kosten für einen Film werden nach Tagen berechnet, nach der Formel: $\text{Basispreis} * \text{Tage}$
Danach werden noch etwaige Rabatte und Preismodifikationen durchgeführt.
9. **Anforderung: Preismodifikation nach Genre**
Je nach Genre werden die Filme um einen bestimmten Faktor teurer oder billiger.
Klassiker (x0.9), Kinderfilme(x0.75), Horror (x1.1), SciFi (x1) und Fantasy (x1.25).
10. **Anforderung: Rabatte**
Stammkunden erhalten 1.5% Rabatt, für 10 Videopoints erhalten Kunden 2% Rabatt, für 20 Videopoints erhalten Kunden 6% Rabatt. Die Videopoints müssen danach entsprechend reduziert werden.
Die Rabatte für Stammkunden und Videopoints sind kumulativ.
11. **Anforderung: Videopoints**
Pro 2 € Rechnungssumme wird dem Kunden ein Videopoint gutgeschrieben. Stammkunden wird weiters ein Videopoint pro Buchung gutgeschrieben.
12. **Anforderung: Prüfung der Altersfreigabe**
Beim Verleih von Filmen muss die Freigabe (FSK 0, 6, 12, 16 18 und RATED 21) geprüft werden.
13. **Anforderung: Benutzerfreundlichkeit - Applikation**
Die gesamte Applikation soll möglichst benutzerfreundlich gestaltet sein. Jede Kernfunktionalität (Anlegen/Bearbeiten von Kunden und Filmen) muss mit maximal 3 Mausklicks erreichbar sein.
14. **Anforderung: Übersichtlicher Buchungsbeleg**
Auf dem Buchungsbeleg soll eine klare Trennung zwischen Gesamtkostenberechnung, Videopoints, Rabatten, ausgeliehenen Filmen und zurückgegebenen Filmen herrschen.
15. **Anforderung: Werbung**
Mitarbeiter sollen die Möglichkeit haben, einen Werbenewsletter an alle Kunden zu versenden.
16. **Anforderung: Datenverlust**
Der Benutzer muss immer gewarnt werden, wenn er bei der gewählten Aktion Daten verlieren kann.
17. **Anforderung: Fehlermeldungen**
Bei Fehlern in der Anwendung soll dem Benutzer eine passende Fehlermeldung ausgegeben werden.
18. **Anforderung: Backup**
Jeden Tag um 23:55 Uhr soll ein komplettes Backup des Systems erstellt werden.
19. **Anforderung: Funktionstüchtigkeit**
Die System muss immer fehlerfrei laufen.
20. **Anforderung: Filmbewertungen**
Das System muss auf Wunsch Bewertungen für Filme von 0.0 (Sehr schlecht) bis 5.0 (Sehr gut) über ein Onlineservice laden können.
21. **Anforderung: Erweiterbarkeit**
Das System muss so programmiert sein, dass es einfach ist, im Nachhinein Programmteile auszutauschen (Beispielsweise ein anderes Onlineservice für die Filmbewertungen).
22. **Anforderung: Filtern**
Die max. Dauer für das Filtern von Stammdaten (1000 Filme, 250 Kunden) beträgt 0.5 Sekunden.

2.6 Anwendungsfallbeschreibung

1. Kunden verwalten	
Primäre Aktoren:	Mitarbeiter
Vorbedingungen:	Der Mitarbeiter ist im System angemeldet.
Nachbedingungen:	Die Veränderungen am Kunden wurden gespeichert.
Szenario:	Ein neuer Kunde kommt in das Geschäftslokal und möchte etwas ausleihen, dazu muss zuerst ein Mitarbeiter den Kunden im System anlegen. Sollte der Kunde dannach noch seine persönlichen Daten ändern wollen, so kann er das jederzeit bei einem Mitarbeiter tun.

2. Filme verwalten	
Primäre Aktoren:	Administrator
Vorbedingungen:	Der Administrator ist im System angemeldet.
Nachbedingungen:	Die Veränderungen am Film wurden gespeichert.
Szenario:	Sobald ein neuer Film erscheint, der für den Filmverleih interessant ist, kauft die Geschäftsführung den Film. Der Administrator trägt anschließend den neuen Film in das System ein. Ab diesem Zeitpunkt kann der Film entliehen werden. Es besteht außerdem die Möglichkeit, die Daten des Films nachträglich anzupassen. Die häufigste Anpassung ist hier sicher der Preis des Films.

3. Film verleihen	
Primäre Aktoren:	Kassier
Vorbedingungen:	Der Kassier ist im System angemeldet. Es existiert ein Kunde. Es existiert ein Film.
Nachbedingungen:	Der Film wurde der Ausleihe des gewählten Kunden hinzugefügt. Ein Buchungsbeleg wurde erstellt.
Szenario:	Ein Kunde kommt mit dem Film, den er ausleihen will und seinem Entleihausweis zum Schalter. Der Mitarbeiter startet den Entleihvorgang und gibt die Kundennummer und die Filmnummer im System ein (um den Vorgang zu beschleunigen, kann er dazu auch den Barcodescanner nutzen). Während eines Entleihvorgangs können immer auch Filme zurückgegeben werden (siehe dazu: 4. Film zurückgeben). Sobald der Entleihvorgang abgeschlossen ist, wird die Buchungsbestätigung finalisiert.

4. Film zurückgeben	
Primäre Akteure:	Kassier
Vorbedingungen:	Der Kassier ist im System angemeldet. Es existiert ein Kunde. Es existiert ein Film. Ein Kunde hat einen Film in der Ausleihe.
Nachbedingungen:	Der Film wurde der Ausleihe des gewählten Kunden entfernt. Ein Buchungsbeleg wurde erstellt. Die Videopoints des Kunden wurden aktualisiert.
Szenario:	Ein Kunde kommt mit dem Film, den er zurückgeben will und seinem Entleihausweis zum Schalter. Der Mitarbeiter startet den Rückgabevorgang und gibt die Kundennummer im System ein (um den Vorgang zu beschleunigen kann er dazu auch den Barcodescanner nutzen). Danach wählt er den Film aus, der zurückgegeben werden soll. Während eines Rückgabevorgangs können immer auch Filme entliehen werden (siehe dazu: 3. Film verleihen). Sobald der Rückgabevorgang abgeschlossen ist, wird die Buchungsbestätigung finalisiert.

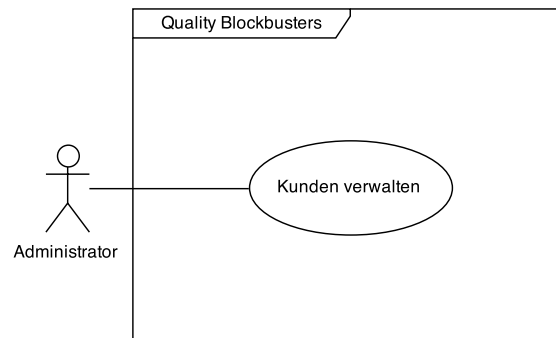
5. Filmstatistik auswerten	
Primäre Akteure:	Controller
Vorbedingungen:	Der Controller ist im System angemeldet. Es wurden bereits Filme an Benutzer verliehen.
Nachbedingungen:	Der Controller hat einen Überblick über die meist ausgeliehenen Filme und Genres.
Szenario:	Der Controller möchte herausfinden, welche Filme im Preis zu hoch oder zu niedrig angesetzt sind, er kann sich dazu Statistiken über die Entleihvorgänge der Filme anzeigen lassen. Über die Auswertung kann er feststellen, welche Filme im Preis angepasst werden sollten.

6. Kundenstatistik auswerten	
Primäre Akteure:	Controller
Vorbedingungen:	Der Controller ist im System angemeldet. Es wurden bereits Filme an Benutzer verliehen.
Nachbedingungen:	Der Controller hat einen Überblick über die Kunden die am meisten Filme ausleihen.
Szenario:	Der Controller möchte die wichtigsten/besten Kunden herausfinden und kann sich dazu Statistiken über das Entleihverhalten der Kunden anzeigen lassen. Über die Auswertung kann er feststellen, welcher Kunde zu einem Stammkunden gemacht werden sollte.

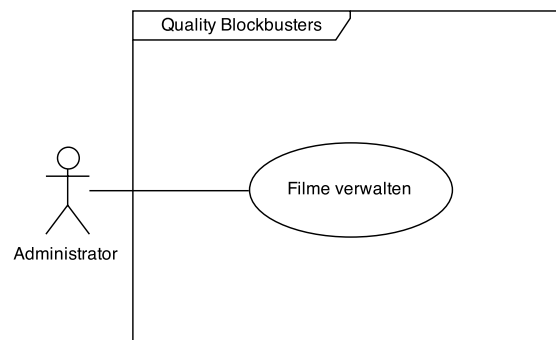
2.7 Anwendungsfalldiagramme

Das Anwendungsfalldiagramm stellt die Anwendungsfälle hier aus Sicht der primären Aktoren dar.

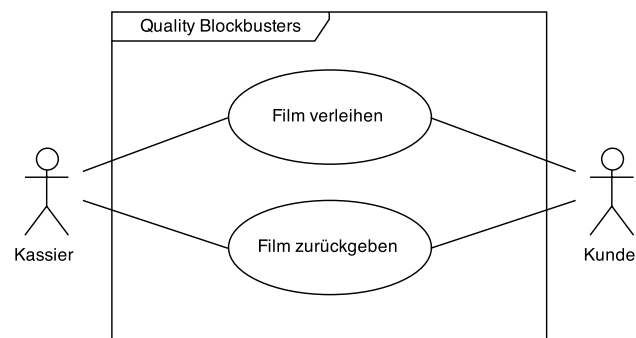
1) Mitarbeiter



2) Administrator



3) Kassier



4) Controller

