

SOFTWAREENTWICKLUNGSPRAKTIKUM

MOVIE GENIE

Software-Entwicklungspraktikum (SEP)

Sommersemester 2012

G r o b e n t w u r f



Auftraggeber

Technische Universität Braunschweig

Institut für Informationssysteme

Mühlenpfordtstraße 23, 2.OG

D-38106 Braunschweig

Betreuer: Silviu Homocanu, Dr. Christoph Lofi

Auftragnehmer:

| Name | E-Mail-Adresse |
|----------------------|--|
| André-Dominik Müller | andre-dominik.mueller@tu-braunschweig.de |
| Frederik Kanning | f.kanning@tu-braunschweig.de |
| Jasper Bergner | j.bergner@tu-braunschweig.de |
| Jewgeni Rose | j.rose@tu-braunschweig.de |
| Martyna Zurek | m.zurek@tu-braunschweig.de |
| Patrick Hill | p.hill@tu-braunschweig.de |
| Philipp Tresp | p.tresp@tu-braunschweig.de |
| Tatiana Reiß | tatiana.reiss@rocketmail.com |

Braunschweig, 23.05.2012

Versionsübersicht

| Version | Datum | Autor | Status | Kommentar |
|---------|------------|---|----------------|--|
| 0.1 | 08.05.2012 | Jasper Bergner | in Bearbeitung | Eintragung aller Kapitel |
| 0.2 | 09.05.2012 | Patrick Hill, Jewgeni Rose | in Bearbeitung | Einleitung Kapitel 1 und Aktivitätsdiagramm hinzugefügt |
| 0.3 | 10.05.2012 | Tatiana Reiß, André- Dominik Müller | in Bearbeitung | Sequenzdiagramme hinzu- gefügt |
| 0.4 | 12.05.2012 | Frederik Kanning | in Bearbeitung | Komponentendiagramm eingefügt, Kapitel 3 bearbeitet |
| 0.5 | 12.05.2012 | Martyna Zurek, Philip Tresp | in Bearbeitung | Einleitung Kapitel 2, Kapi- tel 2 bearbeitet |
| 0.6 | 13.05.2012 | Frederik Kanning, Martyna Zurek, Philip Tresp | in Bearbeitung | Kapitel 3 überarbeitet, Statecharts hinzugefügt |
| 0.7 | 14.05.2012 | Jewgeni Rose, Jasper Bergner | in Bearbeitung | Kapitel 4 bearbeitet, Kor- rektur der restlichen Kapi- tel |
| 0.8 | 15.05.2012 | Tatiana Reiß, André- Dominik Müller | in Bearbeitung | weitere Sequenzdiagramme hinzugefügt |
| 0.9 | 17.05.2012 | siehe Auftragnehmer | in Bearbeitung | Abgabe Betreuer |
| 1.0 | 23.05.2012 | siehe Auftragnehmer | | Abgabe SSE |

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 5 |
| 1.1 | Projektdetails | 5 |
| 2 | Analyse der Produktfunktionen | 8 |
| 2.1 | Analyse von Funktionalität /F100/: Movie Search | 10 |
| 2.2 | Analyse von Funktionalität /F200/: View Movie Details | 12 |
| 2.3 | Analyse von Funktionalität /F300/: Feedback | 13 |
| 2.4 | Analyse von Funktionalität /F400/: Speicherung der Fimbewertung des Benutzers | 14 |
| 3 | Resultierende Softwarearchitektur | 16 |
| 3.1 | Komponentenspezifikation | 16 |
| 3.2 | Schnittstellenspezifikation | 19 |
| 3.3 | Protokolle für die Benutzung der Komponenten | 20 |
| 4 | Verteilungsentwurf | 24 |
| 5 | Glossar | 25 |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 1.1 | Aktivitätsdiagramm Movie Genie | 7 |
| 2.1 | Sequenzdiagramm: Übersicht aller Funktionen | 9 |
| 2.2 | Sequenzdiagramm /F100/ Movie Search | 11 |
| 2.3 | Sequenzdiagramm /F200/ View Movie Details | 12 |
| 2.4 | Sequenzdiagramm /F300/ Feedback | 13 |
| 2.5 | Sequenzdiagramm /F400/ Speicherung der Filmbewertung des Benutzers | 15 |
| 3.1 | Komponentendiagramm Movie Genie | 16 |
| 3.2 | Protokoll-Statechart Movie Genie | 20 |
| 3.3 | Protokoll-Statechart Movie Search | 21 |
| 3.4 | Protokoll-Statechart User Profile | 22 |
| 3.5 | Protokoll-Statechart Text Analysis | 23 |

1 Einleitung



Movie Genie ist eine klassische, webbasierte Client-Server-Anwendung. Das bedeutet, dass unser Server für das Ausführen des Programms verantwortlich ist, und der Benutzer auf der Clientseite Informationen abrufen – in unserem Fall das Erfragen von Filmen durch **"verbose queries"**. Dafür steht dem Benutzer eine Webseite mit einer grafischen Oberfläche im **"Web 2.0"**-Stil zur Verfügung, die er über jeden gängigen Webbrowser mit JavaScript-Unterstützung und einer Internetanbindung erreichen kann.

Für die Umsetzung der Aufgabenstellung verwenden wir eine 3-Schichten-Referenzarchitektur, welche in Kapitel 3 noch im Detail erklärt wird. Grundsätzlich ist unsere grafische Oberfläche die Applikationsschicht bzw. Benutzungsschnittstelle, die dem Benutzer die Kommunikation mit **"Movie Genie"** erlaubt, und ihm zugleich die Ergebnisse präsentiert. Diese baut auf der Serviceschicht auf, die die Funktionalitäten realisiert, die aus der Analyse herauskommen. Das heißt also, dass hier die Anfrage des Benutzers analysiert und mit unseren vorhandenen Film- und Reviewdaten abgeglichen wird. Die Produktfunktionen werden im zweiten Kapitel noch genauer betrachtet und analysiert. Die dritte und unterste Schicht implementiert unseren Datenbankzugriff, wo die persistenten Anwendungsdaten gespeichert werden.

Programmiert wird **"Movie Genie"** in der Programmiersprache Java (Version 7). Für die Benutzeroberfläche verwenden wir das Framework Vaadin, womit wir eine Java Server Page (JSP) erstellen.

1.1 Projektdetails

Die Abbildung 1.1 beschreibt den Workflow des Programms durch ein Aktivitätsdiagramm. Wie dort gezeigt, ist nach dem Aufruf der **"Movie Genie"**-Website eine kurze Authentifizierung mit Benutzername und Passwort notwendig, um auf die Anwendung zugreifen zu können. Ist diese abgeschlossen, erhält der Benutzer die Möglichkeit seine Suchanfragen, in Form einer **"verbose query"** (z.B. **"I'd like to watch an action movie without cars."**), einzugeben und daraufhin den Suchvorgang durch das Betätigen eines Buttons zu starten. Nun prüft **"Movie Genie"** als erstes, ob die Anfrage **"ungültig"** ist, d.h. ob diese aus unsinnigen, zu unklaren oder unverständlichen Informationen besteht (ein Beispiel hierfür wäre eine Eingabe wie: **"bu!rHg2!"**). Sollte dies der Fall sein, wird der Benutzer gebeten seine Suchanfrage neu zu formulieren. Andernfalls bereitet **"Movie Genie"** die Anfrage auf die Analyse vor, indem er überprüft, ob womöglich

bereits Informationen, wie vorherige Suchanfragen der Session und kurzfristig sowie langfristig gespeicherte Filmbewertungen, vorhanden sind. Diese Informationen werden seiner Anfrage hinzugefügt, und an die Analyse-Komponente weitergegeben. Hier werden nun alle Informationen verarbeitet und mit allen dazu passenden Daten aus unserer Datenbank abgeglichen. Nach dem Hinzufügen dieser Informationen, fährt Movie Genie fort, indem er über seine Komponenten zur Sprach- und Textanalyse die Eingabe zerlegt und nach Nomen und den dazugehörigen Adjektiven filtert. Daraufhin stellt er eine Verbindung zwischen den zusammengehörigen Nomen und Adjektiven her, wodurch er auf die Polarität dieses Satzgefüges schließen kann, sprich ob etwas wie z.B. "good acting" positiv oder negativ gemeint ist (Komplexere Sprachgebilde, wie z.B. Ironie können leider nicht erkannt und berücksichtigt werden). Nun stellt Movie Genie noch fest ob etwas negiert wurde (z.B. durch not oder without) und wandelt die aus diesen Schritten gewonnene Informationen in eine Suchanfrage um. Daraufhin kann er dem Benutzer das Ergebnis in Form einer Liste mit Filmdaten (Titel, Genre, Postervorschau, IMDb-Bewertungen) und einer Begründung, warum der Film vom "Movie Genie" ausgesucht wurde, anzeigen. Dieser kann das Ergebnis begutachten und sich einzelne Filme in einer Detailansicht in einem neuen Tab anzeigen lassen, die zusätzlich zu den bisherigen Informationen Zitierungen der Reviews und Quellenangaben enthalten. Die Detailansicht kann der Benutzer jederzeit schließen, oder auch schon parallel dazu zur ursprünglichen Ergebnisliste zurückkehren. Überdies kann der Benutzer Filme, die er bereits gesehen hat oder anderweitig kennt, mit "like" oder "don't like" bewerten. Außerdem hat der Benutzer die Möglichkeit, die Bewertungen der Filme dauerhaft zu speichern. Die gespeicherten Bewertungen fließen dann in zukünftige Suchanfragen mit ein und helfen bei der Optimierung des Suchergebnisses. Gefällt dem Benutzer die vom "Movie Genie" getroffene Auswahl nicht, kann er seine Suche durch zusätzliche Eingaben weiter spezifizieren und eingrenzen. Ihm steht es ebenfalls frei, die letzte Iteration seiner Suchanfragen rückgängig zu machen (und diese auch wiederherzustellen) oder gleich die komplette Suche zu löschen. Den ganzen Ablauf kann der Benutzer so oft er möchte wiederholen. Natürlich kann der Benutzer das Fenster oder den Browser jederzeit schließen, was die Anwendung "Movie Genie" beendet. Diese Option wurde aus Gründen der Übersicht im Aktivitätsdiagramm nicht dargestellt.

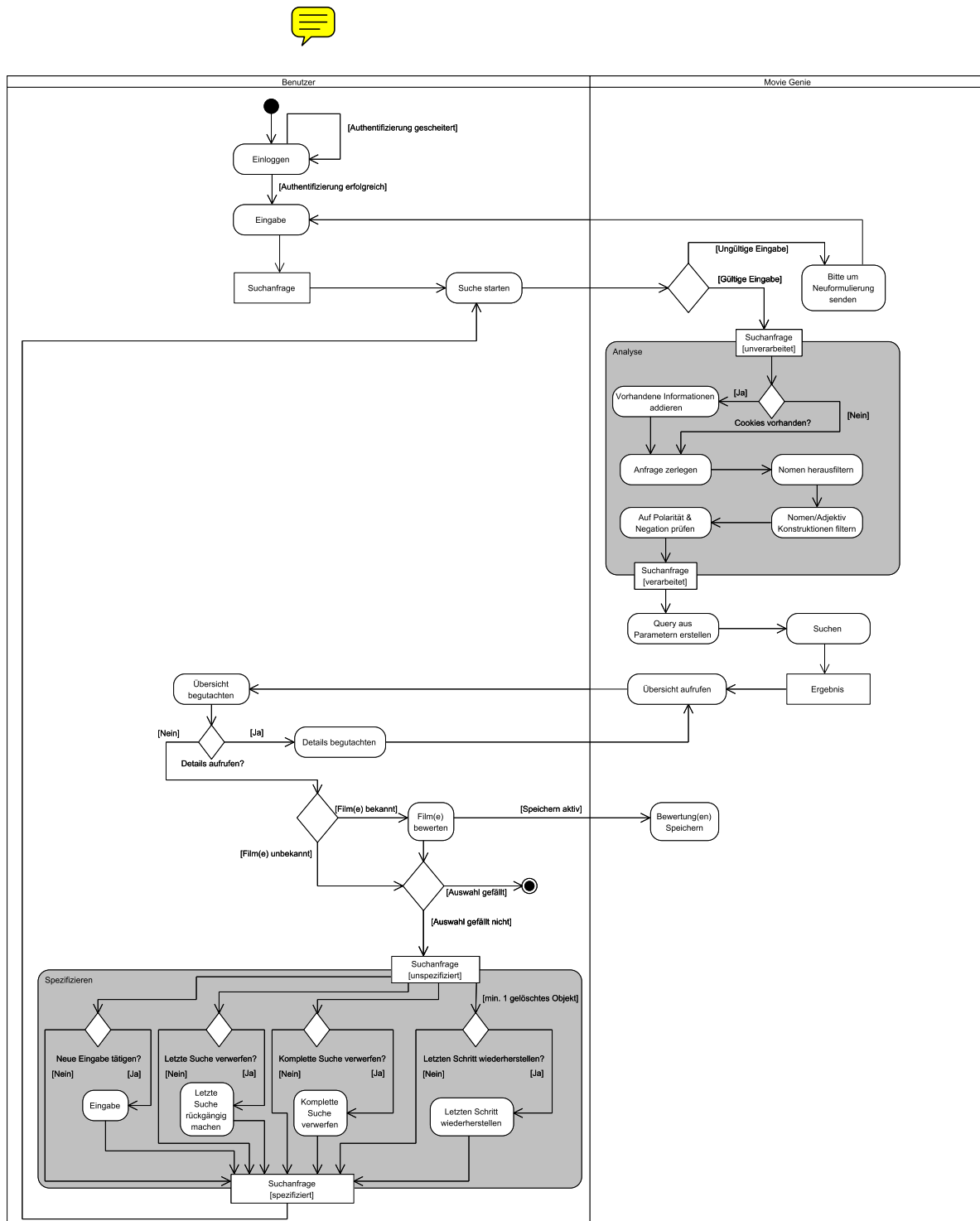


Abbildung 1.1: Aktivitätsdiagramm Movie Genie

2 Analyse der Produktfunktionen

In diesem Kapitel werden die Produktfunktionen und soweit möglich nicht-funktionalen Anforderungen aus dem Pflichtenheft anhand von Sequenzdiagrammen beschrieben. Dabei erhält jede Funktion ein eigenes Unterkapitel, in der sie zunächst im Hinblick auf ihre Verteilung auf die Architektur und für die Realisierung nötigen Datenstruktur untersucht wird. Anschließend beschreiben wir die Funktion verbal und zeigen ihre Interaktion mit den Komponenten vom **"Movie Genie"** in einem Sequenzdiagramm. In diesem sind die einzelnen Komponenten und Methoden so enthalten, wie sie auch später im Entwurf dargestellt werden. Der Benutzer ist hierbei der Anwender vom Movie Genie, der das Programm über das Webinterface (Webseite) aufruft und seine Sucheingaben tätigt.

Bevor wir jede Funktionalität einzeln in den Unterkapiteln analysieren, stellen wir zuerst eine Übersicht vom **"Movie Genie"** in einem etwas abstrakteren Sequenzdiagramm dar. Diese Übersicht soll den Ablauf der Funktionalitäten verdeutlichen. Sie wird in Abbildung 2.0 gezeigt und im nachfolgenden Abschnitt beschrieben.

Um mit **"Movie Genie"** zu kommunizieren, muss der Benutzer zuerst die Webseite mit Hilfe eines Webbrowsers öffnen. Anschließend muss er sich mit einem Passwort anmelden. Konnte sich der Benutzer erfolgreich einloggen, zeigt **"Movie Genie"** die Startseite an, sodass der Benutzer eine Suchanfrage stellen und absenden kann. Besucht der Benutzer die Seite zum ersten Mal, wird mit seinem Einverständnis ein kleines Benutzerprofil angelegt, worin seine Filmbewertungen zu einzelnen Filmen verwaltet werden. Diese kann er sich jederzeit anzeigen lassen und auch ändern bzw. zurücksetzen. Ansonsten werden seine gespeicherten Bewertungen vom **"Movie Genie"** auch noch automatisch zum Optimieren des Ergebnisses verwendet, indem ähnliche Filme zu seinen mit **"like"** bewerteten hinzugezogen werden, und entsprechend mit **"don't like"** gemieden. Um das Benutzerprofil kümmert sich die Komponente **"User Profile"**, wie in Abbildung 2.0 dargestellt. Besitzt der Benutzer bereits ein Profil, wird dieses vom **"Movie Genie"** geladen, und für die Suchanfrage herangezogen. Anschließend wird die Suchanfrage von der **"Movie Search"**-Komponente bearbeitet. An der Bearbeitung dieser sind weiterhin noch andere Komponenten beteiligt. In Kapitel 2.1 wird /F100/ noch detailliert anhand eines Sequenzdiagramms erläutert. Nach Bearbeitung und Auswertung der Suchanfrage wird dem Benutzer eine von **"Movie Search"** erstellte Filmliste angezeigt. Möchte der Benutzer mehr über einen Film aus der Filmliste erfahren, kann er seine Details aufrufen. Dafür öffnet **"Movie Genie"** einen neuen Tab, und lädt die Filmdetails. Nachfolgend werden diese dem Benutzer angezeigt. Hat der

Benutzer genug über die Details herausgefunden, kann er die Ansicht wieder schließen. Danach kann er Filme durch ein positives oder negatives Feedback bewerten ("like" vs. "don't like"), dabei kann er bestimmen, ob die Bewertung langfristig im Benutzerprofil gespeichert, oder nur für die laufende Suche berücksichtigt werden soll. Durch das Betätigen eines entsprechenden Buttons wird "Movie Genie" aufgefordert die Filmliste zu aktualisieren. Dafür kontaktiert er "Movie Search", welche diese aktualisiert und darauf an die "Movie Genie"-Komponente (Benutzungsschnittstelle) übergibt. Diese zeigt die Liste dem Benutzer an.

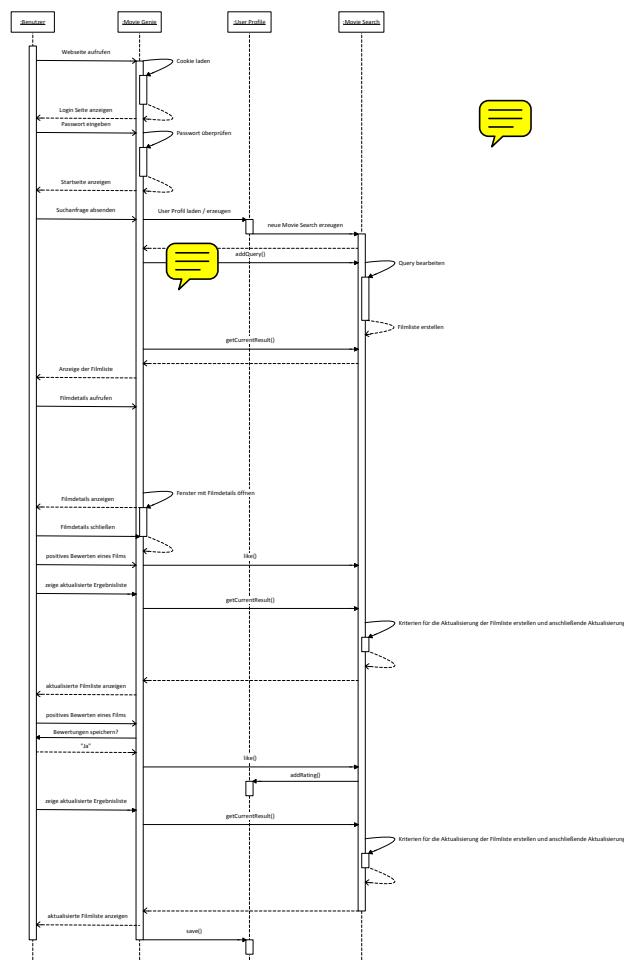


Abbildung 2.1: Sequenzdiagramm: Übersicht aller Funktionen

2.1 Analyse von Funktionalität /F100/: Movie Search

"Movie Search" ist die grundlegende Funktionalität vom "Movie Genie". Die Interaktion der Komponenten, die an dieser Funktionalität beteiligt sind, wird im folgenden Sequenzdiagramm in Abbildung 2.2 verdeutlicht.

Die Abbildung zeigt, wie der Benutzer seine Suchanfrage in der Benutzungsschnittstelle "Movie Genie" eingeben kann. Die Benutzungsschnittstelle sendet die Suchanfrage an "Movie Search" weiter und initialisiert das Profil des Benutzers, wie im Diagramm 2.0 beschrieben. "Movie Search" zerlegt nun diese Anfrage unseren Kriterien nach (scharfe / unscharfe Kriterien). Nachdem eine neue "Text Analysis"-Komponente initialisiert wurde, kann diese die von "Movie Search" zerlegte Suchanfrage bearbeiten, indem sie nicht nur Nominalwörter, Adjektive und die Polarität der Adjektive bestimmt, sondern auch Negationen im Satz auswertet. Diese Informationen werden mit Hilfe von Noun Phrases gewonnen, d.h. dass die vom Benutzer eingegebene "verbose query" in verschiedene Satzglieder getrennt wird. Weiterhin bestimmt die Komponente adverbiale Bereichsbestimmungen und Zeitangaben, wie z.B. "between 2005 and 2010". Diese Ergebnisse sowie die Satzglieder liefert sie zurück an die "Movie Search"-Komponente, welche diese in scharfe und weiche Suchkriterien, wie im Pflichtenheft beschrieben, aufteilt. Nach der Aufteilung tätigt die "Movie Search" selbständig eine Sortierung anhand scharfer Kriterien. Ist diese getan, interagiert sie anschließend mit der "Feature Index"-Komponente, welche die Filme nach weichen Kriterien raussortiert und danach die Ergebnismenge an "Movie Search" übergibt. Die Ergebnismenge wird anhand eines Rankings erstellt. Dabei wird für die oberste Stelle der Film festgelegt, welcher prozentual den Kriterien (scharfen und weichen) aus der "verbose query" am höchsten entspricht. Als Antwort auf die vom "Movie Genie" gesendete Nachricht getCurrentResult() wird die Ergebnismenge verschickt. Zuletzt zeigt "Movie Genie" die Ergebnisliste dem Benutzer grafisch an.

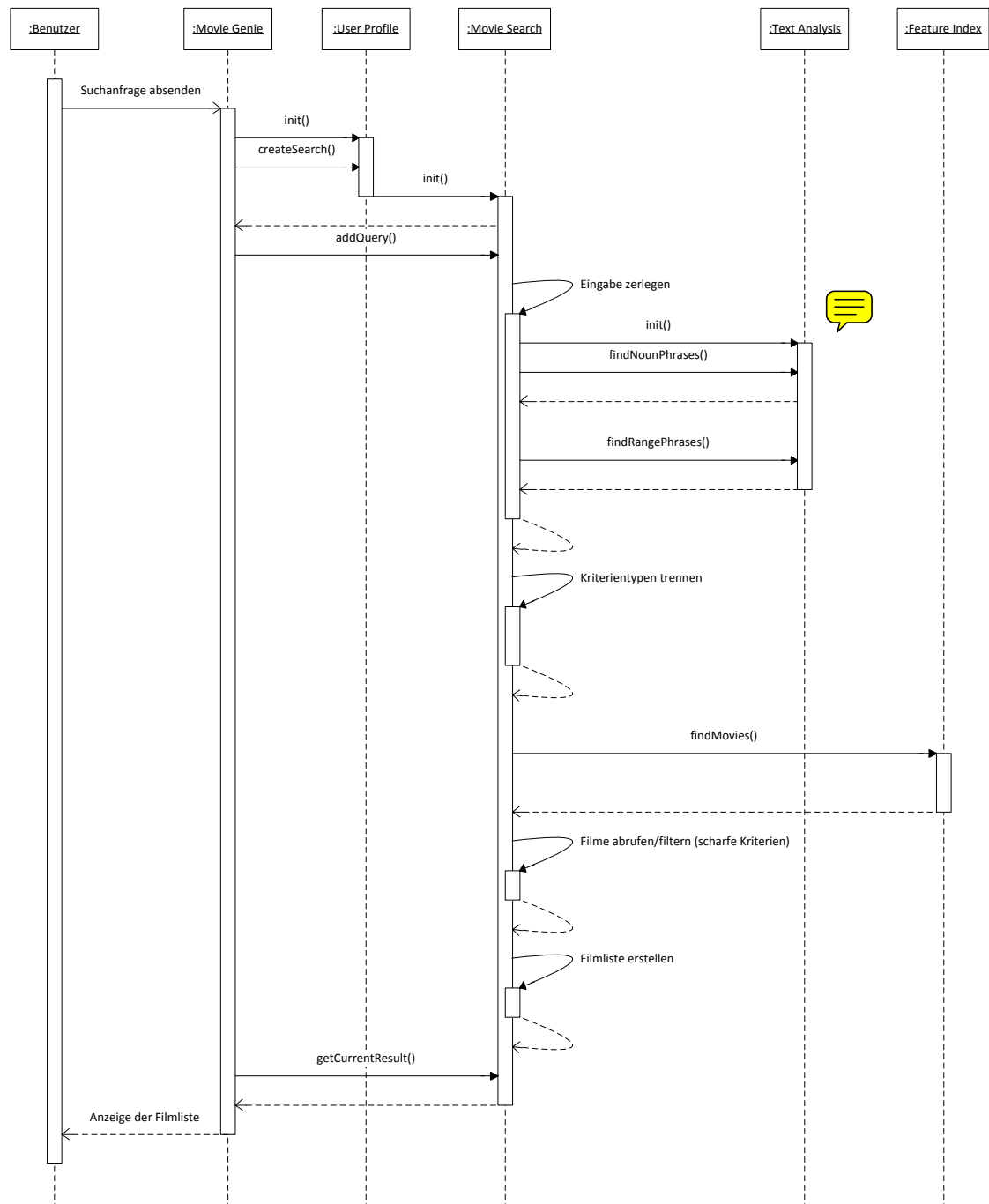


Abbildung 2.2: Sequenzdiagramm /F100/ Movie Search

2.2 Analyse von Funktionalität /F200/: View Movie Details

Das in Abbildung 2.3 abgebildete Sequenzdiagramm zeigt den Ablauf der Betrachtung einer Filmdetailseite.

Hierfür muss der Benutzer einen Film aus der Ergebnisliste, welche ihm im "Movie Search"-Vorgang ausgegeben wurde, wählen und die Filmdetails durch das Klicken auf den verlinkten Filmtitel aufrufen. Dieser Aufruf bedeutet für die "Movie Genie"-Komponente, dass sie erneut eine Verbindung zur "Movie Search" aufbaut, indem sie die Methode `getMovie()` abrufen. Die Kommunikation zwischen "Movie Search" und "Movie Genie" ist synchron und es erfolgt keine Verzögerung. "Movie Search" ruft die Filmdaten des erfragten Films auf und liefert diese umgehend an "Movie Genie". Um die Filmdaten abzurufen, wurde für die "Movie Search"-Komponente ein weiterer Aktivierungsbalken erstellt, welcher den Abrufvorgang darstellt. Nachdem Movie Genie die Filmdetails erhalten hat, macht er diese für den Benutzer sichtbar.

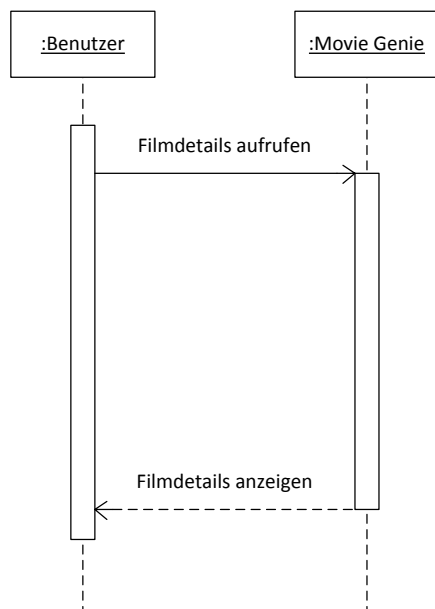


Abbildung 2.3: Sequenzdiagramm /F200/ View Movie Details

2.3 Analyse von Funktionalität /F300/: Feedback

Das Sequenzdiagramm in Abbildung 2.4 zeigt eine beispielhafte Abfolge der Interaktionen zwischen den an der Speicherung der Filmbewertungen des Benutzers beteiligten Komponenten. Nachdem der Benutzer einen Film aus der Ergebnisliste ausgewählt hat (Funktionalität /F100/ Movie Search), bewertet er diesen positiv. Dabei informiert die "Movie Genie"- die "Movie Search"-Komponente, indem sie die Methode `like()` aufruft. "Movie Search" reagiert synchron und stellt Kriterien für die Aktualisierung der Ergebnisliste der Suche zusammen, damit die Bewertung in diese einfließt. Dafür wird außerdem die "Feature Index"-Komponente kontaktiert, welche die Filme zu den zusammengestellten Kriterien liefert. Nachdem die Filme von "Feature Index" gefunden wurden, wird die Filmliste von "Movie Search" aktualisiert. Beim Aufruf `getCurrentResult()` wird "Movie Search" informiert und sendet die aktualisierte Liste an "Movie Genie". Diese zeigt dem Benutzer nach Empfang die Filmliste an. Auch besteht die Möglichkeit für den Benutzer einen Film negativ zu bewerten. Die Bewertung erfolgt analog zur im Beispiel gezeigten positiven Bewertung mit dem Unterschied, dass der "Movie Genie" die Methode `dislike()` aufruft.

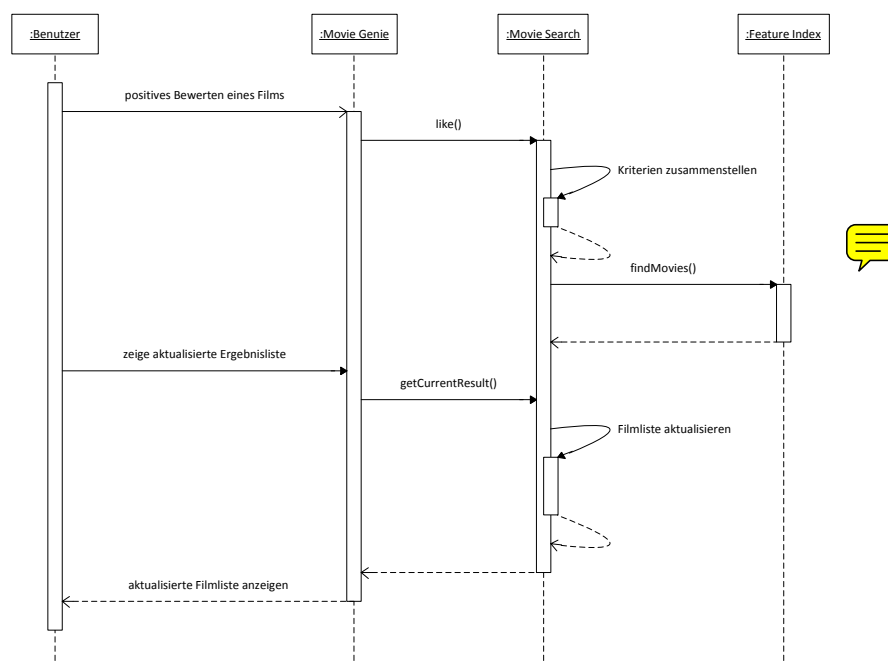


Abbildung 2.4: Sequenzdiagramm /F300/ Feedback

2.4 Analyse von Funktionalität /F400/: Speicherung der Fimbewertung des Benutzers

Diese Funktionalität speichert ein durch den Benutzer abgegebenes Filmfeedback. Dabei erfolgt eine Interaktion zwischen Komponenten aus allen drei Schichten der Architektur. Diese Interaktion wird im Sequenzdiagramm in Abbildung 2.5 dargestellt. Der Benutzer gibt wie bereits in /F300/ beschrieben ein positives oder negatives Feedback ab ("like" oder "dislike"). In Abbildung 2.5 wird der Film positiv bewertet. Eine negative Bewertung erfolgte analog. Nachdem der "Movie Genie" die Information über die positive Bewertung erhält, fragt er den Benutzer, ob diese langfristig gespeichert oder nur für die aktuelle Suche berücksichtigt werden soll. Der Benutzer entscheidet sich für die langfristige Speicherung. In diesem Fall sendet "Movie Genie" durch die Methode like() die positive Bewertung an "Movie Search". Diese Komponente agiert mit dem "User Profile", so dass die Bewertung dem Profil hinzugefügt wird. In "Movie Search" erfolgt weiterhin eine Zusammenstellung der Kriterien und erneute Auswertung der Suche des Benutzers, wobei die abgegebene Bewertung berücksichtigt wird. Die "Feature Index"-Komponente findet also zu den Kriterien passende Filme und übergibt diese an "Movie Search", welche die Filmliste umgehend aktualisiert und an die "Movie Genie"-Komponente überträgt. Diese stellt dem Benutzer eine aktualisierte Filmliste zur Verfügung, in welcher seine Bewertung berücksichtigt wurde und sendet alle aktuellen Änderungen des Profils an die zuständige Komponente "User Profile", welche die Änderungen speichert.

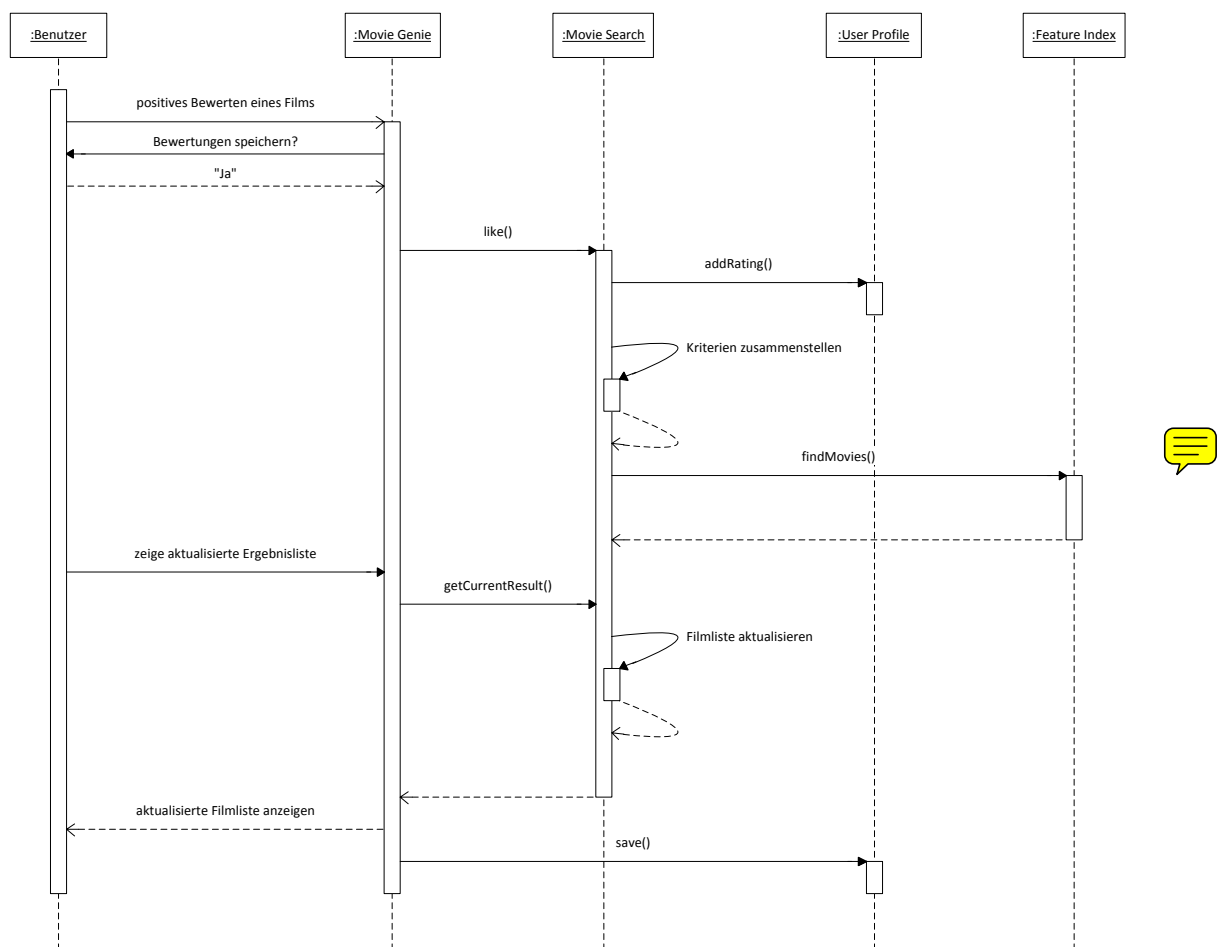


Abbildung 2.5: Sequenzdiagramm /F400/ Speicherung der Filmbewertung des Benutzers

3 Resultierende Softwarearchitektur

Aufgrund der Analyse wurde deutlich, dass sich die Komponenten grob in die drei Aufgabebereiche Präsentation, Anwendungslogik und Datenzugriff einteilen lassen. Wir implementieren daher eine Drei-Schichten-Architektur, die im folgenden genauer erläutert wird.

3.1 Komponentenspezifikation

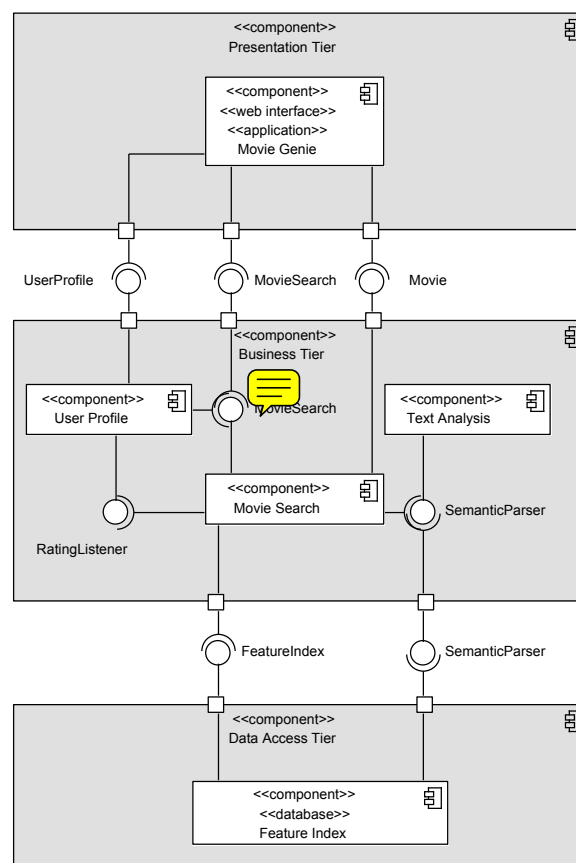


Abbildung 3.1: Komponentendiagramm Movie Genie

Movie Genie, /K10/

Diese Komponente steuert die Interaktion zwischen Benutzer und "Movie Search". Durch sie werden Benutzereingaben an die "Movie Search" weitergeleitet, und die Ergebnisse, die diese

Komponente liefert, graphisch aufbereitet. Insbesondere werden in der Komponente sämtliche GUI-Prozesse wie etwa Anzeige von Dialogen, Ergebnislisten etc. abgewickelt. Die Komponente initialisiert zudem das **"User Profile"** und die **"Movie Search"**.

Movie Search, /K20/

Diese Komponente bereitet die Suchanfragen des Benutzers datenbankgerecht auf, indem sie die Eingabetexte analysiert, in Datenbank-Anfragen umwandelt und darauf basierende Filme zurückliefert. Anfragen können zudem durch Filmbewertungen sowie **"Undo/Redo"**-Aktionen ausgelöst werden. Dazu hält sie alle bisherigen Eingabeschritte vor.

Die Komponente extrahiert zunächst die wesentlichen Phrasen aus dem Eingabetext mit der Text Analysis Komponente, trennt scharfe und unscharfe Kriterien, und erstellt daraus zusammen mit ggf. vorhandenen Benutzerbewertungen eine Liste von Filmen. Die unscharfen Kriterien werden an die **"Feature Index"**-Komponente weitergegeben, die dazu passende Filme liefert. Eingabetexte, die durch die Komponente nicht in Datenbankabfragen umgewandelt werden können, werden als Fehler zurückgemeldet.

An der **"Movie Search"**-Schnittstelle kann zudem direkt eine Sammlung von Filmbewertungen eingespeist werden, die schon bei der ersten Anfrage berücksichtigt wird. Umgekehrt werden an der Komponente ausgelöste Filmbewertungen an die **"RatingListener"**-Schnittstelle gemeldet, sodass **"User Profile"** diese Art der Eingabe unmittelbar verfolgen kann.

Text Analysis, /K30/

Die Komponente zerlegt Eingabetexte in einzelne grammatikalische Objekte, sodass Adjektive und deren zugehörige Substantive, alleinstehende Substantiv-Ausdrücke sowie Adverbialkonstruktionen wie etwa **"movies before 2000"** identifiziert werden. Ferner wird die Polarität der Adjektive bewertet, Negationen werden ebenfalls berücksichtigt.

Der Eingabetext wird in Nominalphrasen zerlegt, sodass z.B. **"beautiful tree"** in dem Eingabesatz **"I'm seeing a beautiful tree"** als Nominalphrase erkannt wird. Zudem trennt die Komponente in diesen Phrasen Substantive (**"tree"**) und Adjektive (**"beautiful"**). Den Adjektiven wird zudem eine Polarität (in diesem Fall positiv, **"beautiful"**) zugeordnet. Falls vor der Phrase eine Negation steht, wird dies ebenfalls berücksichtigt. Ebenso werden bestimmte den Substantiven zugeordnete Adverbialkonstruktionen, die Bereichsangaben von Zahlen enthalten, extrahiert (z.B. **"movies before/from/to/between 2000"**).

User Profile, /K40/

Diese Komponente verwaltet Benutzerprofile, d.h. eine Liste von Filmbewertungen, die ein Benutzer getroffen hat. Über **"User Profile"** erzeugt sie vorkonfigurierte **"Movie Search"**-Komponenten. Umgekehrt wird die Komponente über neue und zurückgenommene Bewertungen per **"RatingListener"** informiert.

Feature Index, /K50/

Die Komponente zerlegt Film-Reviews mit der "**Text Analysis**"-Komponente. Überflüssige Substantive und Adjektive werden herausgefiltert, aus dem übrigen Ergebnis wird ein Index erstellt, der über die gewonnen Wörter und deren Polarität ein Auffinden von Filmen und der Quellreviews der erkannten Schlüsselwörter ermöglicht. Der Großteil der Komponente wird aus Benutzbarkeitsgründen offline, d.h. vor der eigentlichen Movie Genie Anwendung ausgeführt, da rechenintensive Prozesse für den Aufbau des Index nötig sind.

3.2 Schnittstellenspezifikation



| Schnittstelle | Aufgabenbeschreibung |
|-----------------------------|--|
| /S10/ MovieSearch | Anfrageverarbeitung |
| init() | lädt eine Liste von Filmbewertungen eines Benutzers im voraus, und meldet eine UserProfile-Komponente an |
| addQuery() | Benutzeranfragen (Verbose Queries) auswerten und auf Ergebnis anwenden |
| getCurrentResult() | erstellt Liste von Filmen aufgrund aller aktuell angewendeten Eingaben |
| getExplanation() | liefert alle aus den Benutzereingaben erstellten Kriterien zurück |
| getMovie() | liefert einen einzelnen Film anhand der ID aus der Datenbank |
| undo() | macht die letzte Benutzeraktion rückgängig |
| redo() | wendet zuletzt rückgängig gemachte Aktion wieder an |
| reset() | nach dem Aufruf wird keine bisher gemachte Benutzeraktion im Ergebnis berücksichtigt |
| /S20/ Movie | Zugriff auf Daten eines Films |
| like() | bewertet den Film positiv |
| dislike() | bewertet den Film negativ |
| resetRating() | entfernt die Bewertung |
| /S30/ SemanticParser | Grammatikalische Zerlegung und beschränkte Auswertung von Volltext |
| init() | initialisiert den Parser mit einem Volltext |
| findNounPhrases() | liefert alle im Text enthaltenen Nominalphrasen, wertet Negationen aus, bestimmt Adjektive und deren Polarität |
| findRangePhrases() | wie findNounPhrases(), bestimmt jedoch zusätzlich adverbiale Bereichsbestimmungen (z.B. "between 2 und 5") |
| /S40/ FeatureIndex | findet Filme aufgrund von Schlüsselwörtern und deren Polarität |
| findMovies() | findet alle indizierten Filme, denen die angegebene Schlüsselwörter/Polaritäten-Kombination zugeordnet ist |

| Schnittstelle | Aufgabenbeschreibung |
|-----------------------------|---|
| /S50/ UserProfile | verwaltet ein Benutzerprofil mit Filmbewertungen |
| init() | versucht, ein Profil anhand einer bestimmten ID zu laden; falls nicht möglich, wird ein neues Profil initialisiert |
| save() | speichert alle aktuellen Änderungen des Profils |
| createSearch() | erzeugt eine neue Movie Search-Komponente mit den geladenen Bewertungen |
| /S60/ RatingListener | überwacht alle von einem Benutzer getätigten Filmbewertungen |
| addRating() | meldet die getätigte Filmbewertung unter Angabe des Films und Wertung |
| removeRating() | nimmt die angegebene Filmbewertung zurück |

3.3 Protokolle für die Benutzung der Komponenten

Die Arbeitsweise der Komponenten wird im Folgenden anhand von Protokoll-Statecharts dargestellt.

Movie Genie

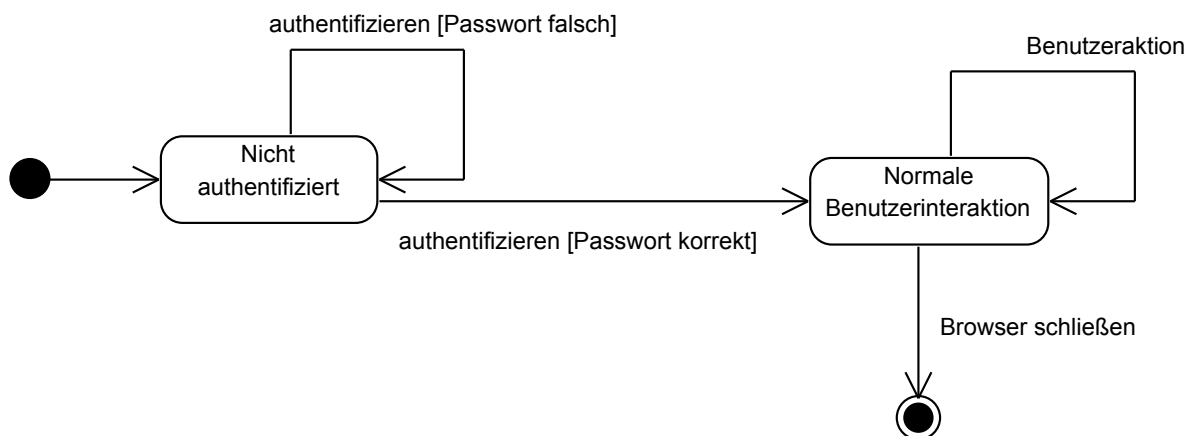


Abbildung 3.2: Protokoll-Statechart Movie Genie

Der Benutzer muss sich zunächst an der Anwendung authentifizieren. Danach beginnt die eigentliche Benutzersitzung, in der der Benutzer alle anderen Funktionen ausführen kann, bis er/sie den Browser schließt.

"Wiederverwendung": Da diese Komponente die Benutzungsschnittstelle zur Verfügung stellt, ist sie sehr spezifisch auf Filmsuchen gemäß Lasten- und Pflichtenheft ausgelegt und daher nicht wiederverwendbar.

Movie Search

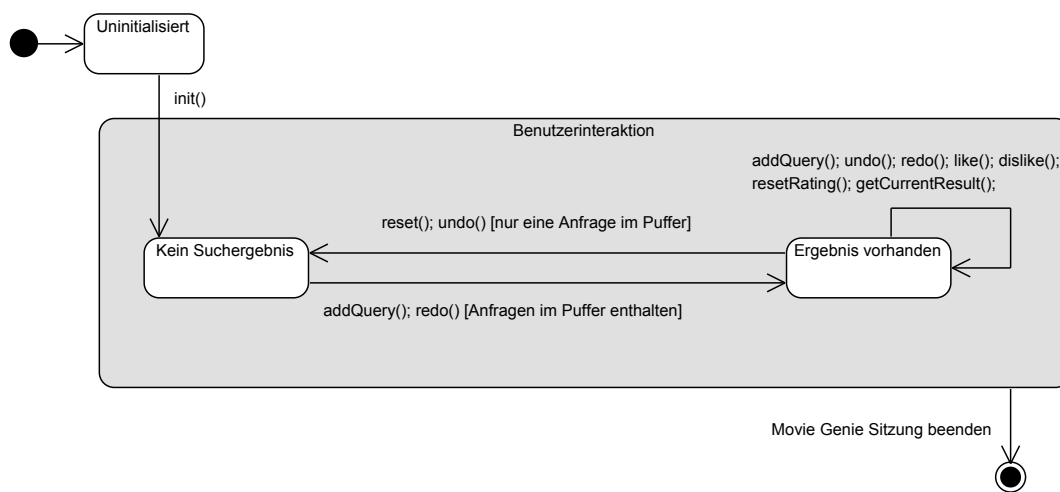


Abbildung 3.3: Protokoll-Statechart Movie Search

Die Komponente muss zunächst mit einem (ggf. leeren) Benutzerprofil von Filmbewertungen initialisiert werden, um vom Benutzer in der aktuellen Sitzung ausgelöste Bewertungen speichern zu können. Bevor eine Ergebnisliste angezeigt (*getCurrentResult()*) und mit diesem Ergebnis gearbeitet werden kann (Filme bewerten, Aktionen rückgängig machen), muss durch eine Anfrage (*addQuery()*) die Erstellung ebendieser Liste ausgelöst werden. Ferner kann die Liste durch Zurücksetzen oder eine Undo-Aktion, wenn der Aktionspuffer nur noch ein Element enthält, wieder geleert werden (analog mit *redo()* befüllt werden). Modifikationen der Liste werden durch weitere *addQuery()*-Anfragen ausgelöst, leeren diese jedoch nicht. Eine Instanz der Komponente existiert während der gesamten Sitzung und wird am Ende dieser zerstört.

Wiederverwendung: Diese Komponente ist ebenfalls recht spezifisch, aber unabhängig vom GUI, und lässt sich daher mit beliebigen Benutzungsschnittstellen verwenden. Eine Wiederverwendung ist aber nicht geplant.

User Profile

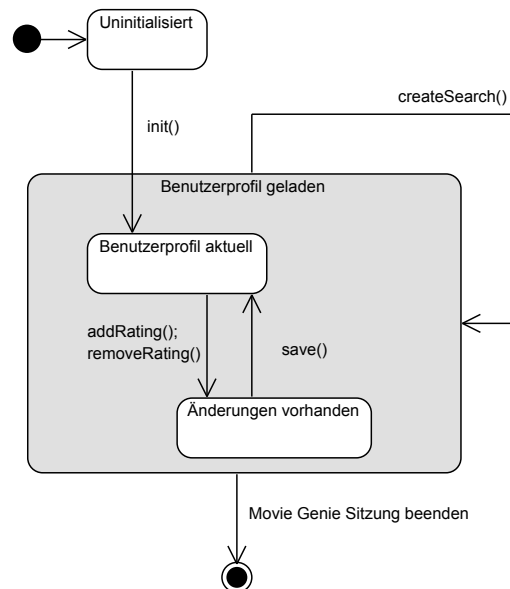


Abbildung 3.4: Protokoll-Statechart User Profile

Die Komponente wird, falls vorhanden, mit einer Benutzer-ID initialisiert, ansonsten wird ein neues Profil angelegt. Falls eine ID angegeben wurde, werden zuerst alle Filmbewertungen von diesem Benutzer geladen. Danach ist das Profil abruf- und bearbeitbar. Änderungen durch Filmbewertungen, führen dazu, dass die persistente Version des Profils nicht mehr aktuell ist, was mit *save()* behoben werden kann, aber nicht muss. Unabhängig davon können mit den aktuellen Profildaten beliebige "Movie Search" Komponenten initialisiert werden.

Wiederverwendung: Diese Komponente ist speziell auf die Zusammenarbeit mit den angrenzenden Komponenten ausgelegt und verwaltet sehr spezielle Daten (positive/negative Bewertungen eines Films), ist daher nicht wiederverwendbar.

Text Analysis

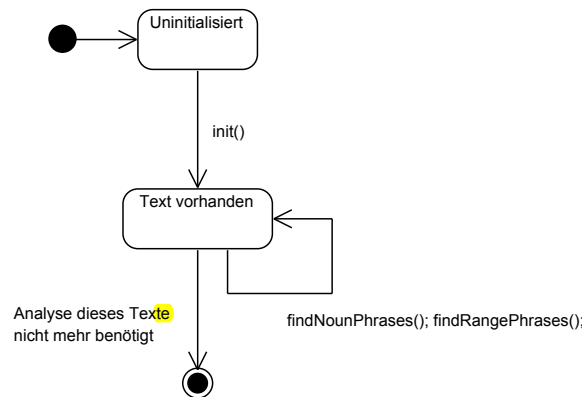



Abbildung 3.5: Protokoll-Statechart Text Analysis

Für jeden zu untersuchenden Text wird eine Instanz dieser Komponente mit dem jeweiligen Text initialisiert. Danach können die angebotenen Analysefunktionen beliebig oft ausgeführt werden, bis die Instanz nicht mehr benötigt wird.

Wiederverwendung: Die Komponente ist allgemein auf Zerlegung von englischen Texten in Nominalphrasen und deren Polaritätsbestimmung ausgelegt, kann daher potenziell in beliebigen Anwendungen verwendet werden, die diese Funktionalität benötigen. Dies ist jedoch nicht konkret geplant.

Feature Index

Diese Komponente stellt  nur einen Suchdienst zur Verfügung, der zu einer Eingabe gehörige Filme auffindet. Es gibt dabei kein einzuhaltendes Protokoll, die Suche (und damit einzige Funktion) kann jederzeit ausgeführt werden. Die Instanz existiert, solange Movie Genie ausgeführt wird.

Wiederverwendung: Da die Komponente das Auffinden von Filmen anhand unscharfer Kriterien (Schlüsselwörter und Polarität) ermöglicht, ansonsten aber vom restlichen System entkoppelt ist, kann sie auch in anderen Anwendungen mit ähnlicher Funktionalität wiederverwendet werden, was jedoch nicht geplant ist.

4 Verteilungsentwurf

Bei "Movie Genie" handelt es sich um keine typische verteilte Anwendung, bei der verschiedene Systembestandteile auf unterschiedlichen Rechnern laufen und miteinander interagieren müssen. Sämtliche Komponenten werden auf unserem Server ausgeführt, und vom Benutzer über eine Webseite aufgerufen. Aus diesem Grund haben wir hier auf ein Verteilungsdiagramm verzichtet.

5 Glossar

Feature Index - Dies ist eine Komponente in unserer Architektur, die bereits im Vorfeld alle Reviewdaten auswerten soll und dem Ergebnis einen Index zuweist. Anhand des Indexes und der erkannten Schlüsselwörter können später Filme zugeordnet werden.

JSP - Die Abkürzung bedeutet Java Server Pages und ist eine Webprogrammiersprache zur einfachen dynamischen Erzeugung von HTML- und XML- Ausgaben eines Webserver. Noun phrases - Eine noun phrases besteht aus einem Artikel, einem oder mehreren Adjektiven und einem Nomen. Ein Beispiel dafür wäre: "**the red ball**". Mit Hilfe einer noun phrases lassen sich die Reviewdaten und Suchanfragen analysieren. Aus der Suchanfrage werden sie herausgefiltert um sie dann mit den Reviewdaten abzugleichen. Aus diesem Prozess können dann Filme gefunden werden, die diesen Übereinstimmungen entsprechen.

Preprocessing - Dies ist eine Datenvorverarbeitung, die zu offline Zeiten durchgeführt werden kann. Wir nutzen dies, um die die Reviewdaten bereits im Vorfeld zu analysieren. Dadurch wird der Aufwand der Anwendung stark reduziert.

scharfe Kriterien - Sie sollen eindeutige Filmdaten charakterisieren. Dazu gehören zum Beispiel das Genre, der Schauspieler, der Regisseur oder der Titel eines Films.

unscharfe Kriterien - Zu den unscharfen Kriterien gehören Daten, die nicht direkt aus den Filmdaten abgeleitet werden können. Dazu gehört zum Beispiel die Aussage: "**good plot**".

Verbose Queries - Die verbose Queries unterscheiden sich von Keyword Queries (Stichwortabfragen), die in Suchmaschinen sonst üblich sind, vor allem dadurch, dass man seine Fragen so formulieren kann, als ob man mit einem anderen Menschen kommunizierte. Ein Beispiel dafür ist: "**I am looking for a movie with a good plot**". Sie beinhalten also nicht nur ein einzelnes Stichwort, sondern darüber hinaus über positiv / negativ beschriebene Nomen und verschiedene Adjektive. Diese Zusätzlichen Informationen können dazu genutzt werden, eine Suchanfrage detaillierter und genauer zu gestalten.