

Johannes Visintini, Philip Bell,
Moritz Nöltner

Vorlesung: Einführung in Software Engineering
Institut für Informatik
Universität Heidelberg

3. Februar 2014

Requirements - Funktional

UseCase: SyncDialog

Description: öffnet sich wenn ein/e Abgleich/Aktualisierung
oder der Use Case „Stichwortsuche (Filmtitel)“ aufgerufen wird

Precondition: Internetverbindung und MovieManager gestartet

Postcondition: Film wird vervollständigt/aktualisiert

Exception: Film in der Online-Datenbank nicht gefunden

UseCase: Stichwortsuche (Filmtitel)

Description: Der User übergibt (über das Titel-Feld) ein Stichwort.
Dann sucht das Programm nach diesem (oder ähnlichem) Titel
in der Online-Datenbank.

Precondition: Internetverbindung und MovieManager gestartet

Postcondition: User bekommt Liste von Filmtiteln

Exception: kein Ergebnis gefunden

TestCases

System Test

- testLoadMatchMovieDataDialog
- testSendSearchQuery
- testReceiveSearchQueryResult
- testShowMovieMatchDetailData
- testChangeFilmDataInMatchMovieDataDialog
- testAdoptFilmDataInMatchMovieDialog
- testTimeoutInMatchMovieDataDialog
- testTimeoutNotificationInMatchMovieDialog
- testAbortInMatchMovieDataDialog

testSendSearchQuery [TestCase]

Name	testSendSearchQuery
Description	checks whether the dialog will properly send out a search query
Comments	0 comments
Use Case	Match movie data to IMDb  
Type	unit
Precondition	the match movie data dialog is opened and a search term is entered
Postcondition	a search query to imdb is sent out and a timeout is started

GUI - Entwurf

The diagram shows a hand-drawn GUI design for a movie search and suggestion dialog, divided into two main panels.

Left Panel:

- Suchbegriff:** A text input field for the search term.
- Aktuelle Daten:** A section containing five data fields:
 - Filmtitel A
 - orig Titel A
 - Filmlänge A
 - ▼ Bewertung A
 - ▼ Kategorie A
- Darsteller:** A label for the actor field, with "Bsp darst 1" as an example. The field is represented by a vertical scrollbar.

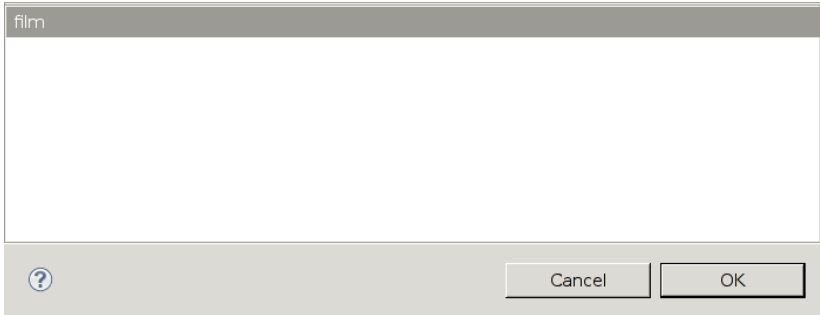
Right Panel:

- ▼ | Vorschläge:** A section header for suggestions.
- Daten des Vorschlags:** A section containing five data fields, each with a checkbox:
 - ☒ Filmtitel B
 - ☒ orig Titel B
 - ☐ Filmlänge B
 - ☐ ▼ Bewertung B
 - ☒ ▼ Kategorie B
- Darsteller:** A label for the actor field, with "Bsp darst 2" and "Bsp darst 3" as examples. The field is represented by a vertical scrollbar.

GUI - Show Loaned Movies Dialog

Loaned Movies

These Movies are currently loaned



GUI - Add Performer Dialog

Create Performer

Step 1: Creat a new Performer



Name:

Gender: ☐ Female
☐ Male

Rating: ▼

GUI - Movie Wizard

Create Movie

Step 1: Create a new Movie

Name:	<input type="text" value="er Karibik"/>	Suchergebnisse:	<input type="text" value="Fluch der Karibik"/>	<input type="button" value="▼"/>
Title:	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Fluch der"/>	
Orig. Title:	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Pirates of"/>	
Time:	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	
Rating:	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	
Category:	<input type="text" value="Action"/>	<input type="checkbox"/>	<input type="text" value="Action"/>	

Implementation - Query-Funktion

```
suggestionDropDown.removeAll();
suggestionDropDown.select(0);
final String nearMovieName = textSearchName.getText();
Thread t = new Thread() {
    private void searchNearMovies(String title)
    {
        MovieQueryService mqs = new MovieQueryService();
        if (mqs.getListOfNearMovies(title) != null)
        {
            nearMovies = mqs.getListOfNearMovies(title);
        }
    }

    public void run()
    {
        searchNearMovies(nearMovieName);
        try {
            this.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
};
t.start();
updateSuggestionDropDown();
setPageComplete(true);
```


ENDE