Watchlistpromax von Joel und cyrill

## WatchlistProMax

### Version 1.1.0, Datum: 12. März 2024

Hier kannst du einen Printscreen deiner App einfügen.

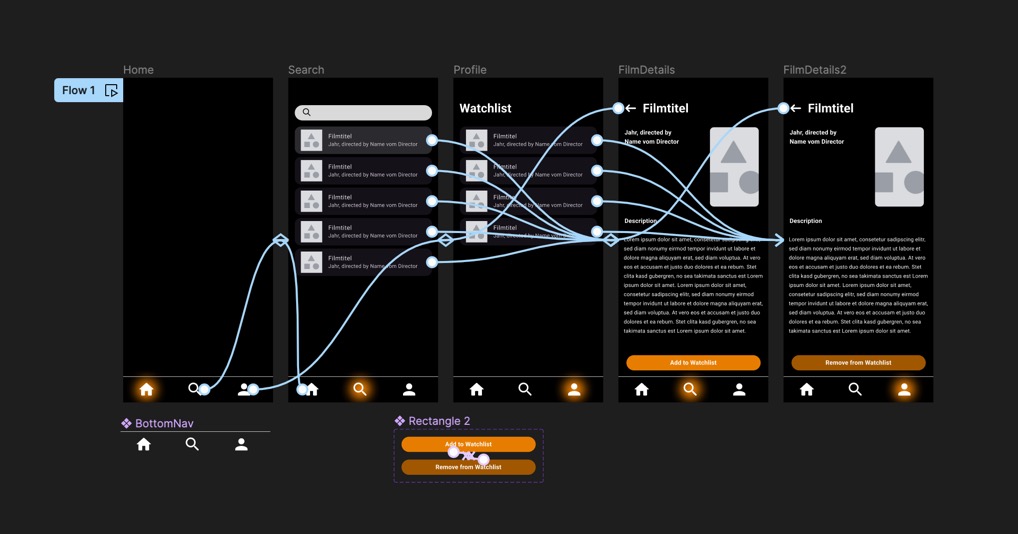
Abstract (Kurzbeschreibung)

Mit der WatchlistProMax App kann man ganze einfach einen Überblick über seine Watchlist bekommen. Man kann mithilfe von einem Suchfeld die OMDB API durchsuchen. Die Ergebnisse werden dann aufgelistet angezeigt und man sieht das Poster, den Titel und das Jahr, in welchem der Film herausgekommen ist. Mit einem Click auf den Film öffnet sich eine Detailpage, wo man noch weitere Informationen einsehen kann, wie der Regisseur, der Cast oder die Bewertungen. Ausserdem wird ein Button angezeigt, mit welchen man mit einem Click, den Film in seine Watchlist hinzufügen kann. Diese Watchlist kann man dann im Home von dieser App ganz übersichtlich überblicken und durchstöbern und auch schon wie beim Suchen, kann man die Detailansicht des in der Watchlist stehenden Filmes anschauen.

user stories

* Ich als User möchte nach Filmen suchen können
* Ich als User möchte bei den Ergebnissen das Filmposter sehen können
* Ich als User möchte eine Detailansicht eines Filmes ansehen können
* Ich als User möchte einen Film in meine Watchlist hinzufügen können
* Ich als User möchte einen Film aus der Watchlist entfernen können
* Ich als User möchte auf meine Watchlist auch offline zugreifen können

Mockups



**Search Screen:** Im Search Screen ist ganz oben ein Textfeld, mit welchem man die Filme durchsuchen kann. Nach der Suche werden die Ergebnisse unten aufgelistet. Man sieht das Poster, den Filmtitel und das Jahr in welchem der Film herausgekommen ist. Man kann durch diese Liste durchscrollen. Ebenfalls kann man jeden Film anklicken und man wird zu der Detailansicht weitergeleitet.

**Detailansicht:** In der Detailansicht sich man das Poster in einer noch grösseren Auflösung und sonstige Interessante Fakten über den ausgewählten Film, wie z. B. der Regisseur, Lauflänge, Genre, IMDb Rating etc.

**Watchlist:** In der Watchlist Ansicht werden alle Filme aufgelistet, welche man zu der Watchlist hinzugefügt hat. Wie auch schon beim Search Screen werden hier die Filme rudimentär angezeigt (Poster, Title und Jahr) und man ebenfalls auf den Film draufklicken und man landet auf der Detailansicht des Filmes.

**Unterschiede zwischen dem Mockup und der App:**

* Der Home Screen ist jetzt der Watchlist Screen
* Bei Account wäre jetzt der Platz, um Mandantenfähigkeit zu implementieren und ein Onlinefeature möglich zu machen etc.
* Einige Effekte und Buttons sind teilweise ein wenig anders platziert als im Mockup.

Technische Realisierung

**API:**

Ein Teil des Projektes ist die Kommunikation mit OMDB (Open Movie Database). Die API liefert und all die Informationen über die Filme, welche wir für diese Applikation brauchen.

Man kann nach einem Titel suchen, dann bekommt man verschiedene Treffer oder man sucht nach einer IMDb ID, bei welchen man logischerweise dann nur einen Treffer zurückbekommt.

Suche nach einem Film mit dem Titel Matrix: <https://www.omdbapi.com/?s=matrix&page=1&apikey=e69bd20e>

Suche nach dem Film mit der IMDb ID “tt0133093”:

<https://www.omdbapi.com/?i=tt0133093&apikey=e69bd20e>

**Search Funktion:**

Um nach Filmen suchen zu können, wurde ein TextInput eingefgt, in welchem man seinen Eingabe tätigen kann. **A computer screen shot of a program code

Description automatically generated**

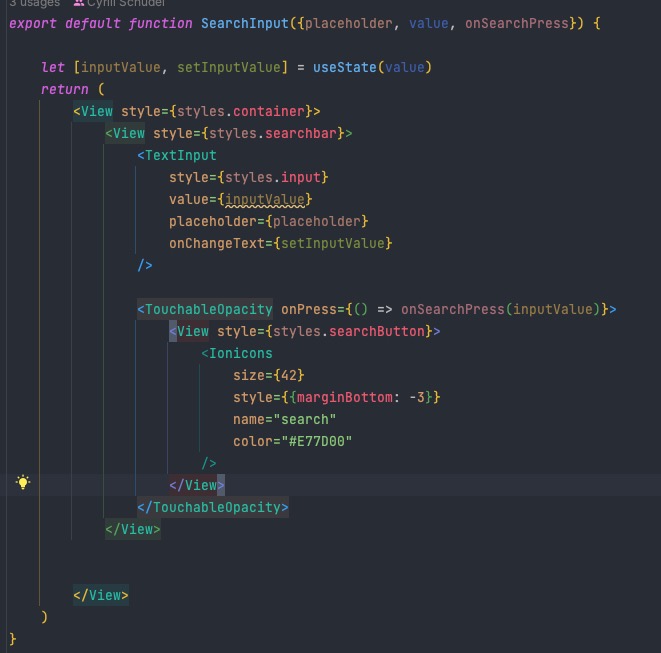
Dieser Code hat die Logik des Search Screens. Man sieht, wie am Anfang ein paar States definiert werden, welche gebraucht werden. Danach beginnen wir auch schon mit dem Request, wo wir nach dem Suchen, was wir im Textfeld eingeben werden. Falls der inhalt im Textfeld noch undefinded oder ein leerer String ist, wird aus der Funktion mit einem early return der Request verhindert. Falls der Request fehlschlägt, wird ein Fehler geworfen. Wenn der Request erfolgreich war, wird das was in der Response unter .Search steht, in den State movies geschrieben.

Somit haben wir schon einmal alle unsere Filme, jetzt müssen wir diese nur noch anzeigen.

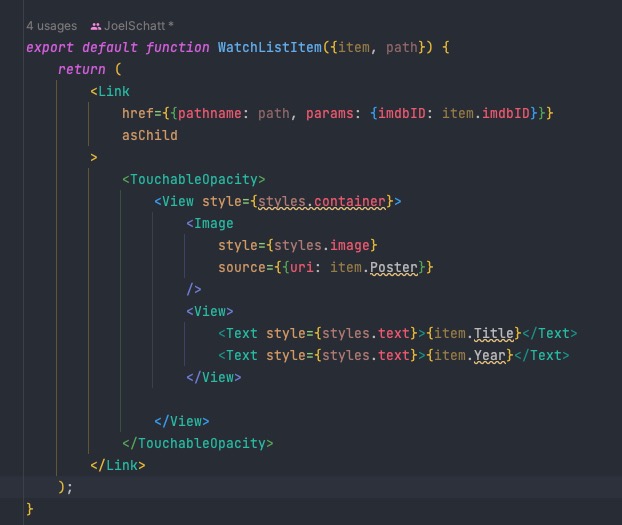
****

Damit man überhaupt auf der Seite eine Searchbar sieht, haben wir eine SearchInput Komponente gebaut, welcher wir einen Placeholder und auch noch eine kleine Logik für den Fall mitgeben, wenn der Search Button gedrückt wird, weitergeben. In der Logik, wird der Wert aus dem Inputfeld ausgelesen und nachher dann die loadMovies Methode aufgerufen, welche dann den Request auf die API macht.

Weiter unten verwenden wir eine Flatlist, um nachher dann alle Filme auf Flatlist Elemente zu mappen und dann anzuzeigen.



Für den SearchInput Component geben wir einen Placeholder, den value und onSearchPress mit. Der Wert im Parameter Placeholder, wird der Placeholder im TextInput. Value wird übergeben und dann gleich gesetzt mit dem UseState inputValue, welcher dann wiederrum, bei jedem ändern des values im TextInput wieder aktualisiert wird. Bei onSearchPress wird ein bisschen Funktionslogik übergeben (Siehe vorheriges Bild) welche dann beim Drücken des Buttons ausgeführt wird, und dabei wird der Wert des aktuellen TextInputs übergeben und dann durch die Logik verarbeitet. In der Logik wird dann der Request auf der API auf Basis dieses Values ausgeführt.

****

Im Watchlistitem geben wir zuerst den Film weiter, welchen wir Anzeigen wollen, sowie aber auch den Path, auf die nächste Stack Navigation, falls der User sich dazu entscheiden sollte darauf zu klicken. Man sieht auch gleich, dass alles in einem Link Tag eingebettet ist, welches die Stack Navigation und weiterleitung auf die Detailseite erst möglich macht. Danach ist es einfach ein “Button”, mit einem Poster, Titel und dem Release Year des Filmes, auf welchen man draufklicken kann.

Testing

In diesem Kapitel definiert ihr die manuellen Tests, die durchgeführt werden. Es müssen mind. 4 manuelle Tests definiert werden.

Hier ein Beispiel eines Tests:

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall: 00001, ST |
| Anforderungen | * Ich als User möchte nach Filmen suchen können |
| Vorbedingungen | Das App muss laufen |
| Ablauf | 1. Navigiere in der Bottom Navigation zu Search 2. Klicke auf das Textinput Feld 3. Gebe “Back to the future” ein 4. Klicke auf das orangene Lupen Icon |
| Erwartetes Resultat | Es sollen Filme auftauchen, welche im Titel «Back to the future» drinnstehen haben. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | ST-002 |
| Anforderungen | * Ich als User möchte eine Detailansicht eines Filmes ansehen können |
| Vorbedingungen | Es soll unter search nach einem beliebigen Film gesucht werden und die Ergebnisse sollen sichtbar sein. |
| Ablauf | 1. Klicke auf den Film 2. (Ergebnis 1) sollte Scrollable sein 3. Klicke oben Links auf den Pfeil zurück 4. (Ergebnis 2) |
| Erwartetes Resultat | Ergebnis 1:  Ergebnis2: |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | ST-003 |
| Anforderungen | * Ich als User möchte einen Film in meine Watchlist hinzufügen können |
| Vorbedingungen | Mann soll sich in der Detailansicht eines Filmes befinden |
| Ablauf | 1. Klicke auf den orangenen Button «Add to Watchlist» 2. Klicke in der Bottom Navigation auf Home |
| Erwartetes Resultat | Der Film sollte jetzt in der Watchlist angezeigt werden. |

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | ST-004 |
| Anforderungen | * Ich als User möchte einen Film aus der Watchlist entfernen können |
| Vorbedingungen | App soll laufen und es sol mind. ein Film in der Watchlist vorhanden sein. |
| Ablauf | 1. Klicke in der Bottom Navigation auf Home 2. Klicke auf einen beliebigen Film, der in deiner Watchlist ist. 3. Klicke in der Detailansicht auf den Button «Remove from Watchlist» |
| Erwartetes Resultat | Der Film sollte jetzt von der Watchlist verschwunden sein |

Testprotokoll

In diesem Kapitel wird die Testausführung protokolliert. Welche Tests wurden wann von wem mit welchem Resultat durchgeführt.

Hier ein Beispiel eines Eintrages:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen |
| ST-001 | Noah || Sameer | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| ST-002 | Noah || Sameer | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| ST-003 | Noah || Sameer | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| ST-004 | Noah || Sameer | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |

fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.