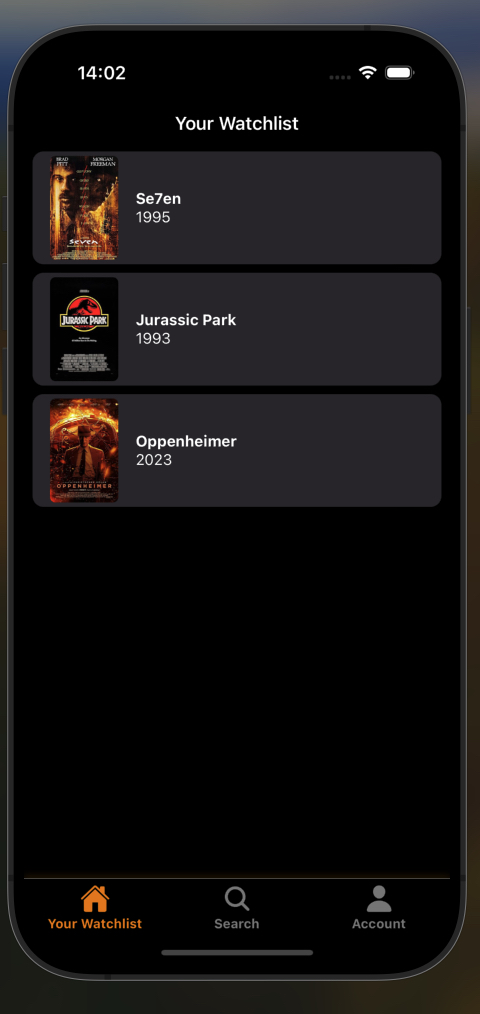
Watchlistpromax von Joel und cyrill

## WatchlistProMax

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



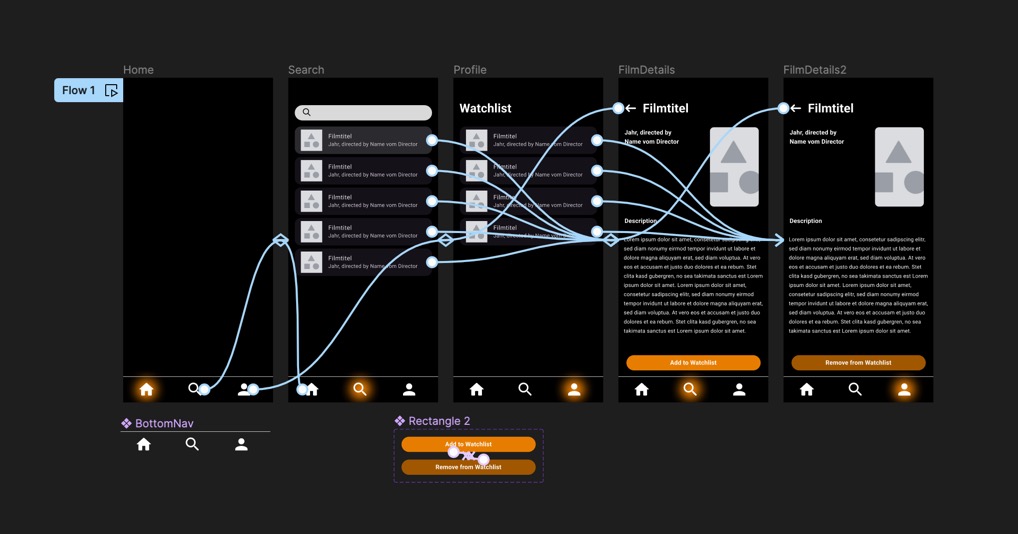
Abstract (Kurzbeschreibung)

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 den Regisseur, die Spiellänge oder die Bewertungen. Ausserdem wird ein Button angezeigt, mit welchem 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 Suchfeld, 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 sieht 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 kann ebenfalls auf den Film draufklicken und 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.

**Wichtig: Mit diesem API Key, hat man nur 1000 Anfragen pro Tag zur Verfügung**

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 eingefügt, in welchem man seine Eingabe tätigen kann. A computer screen shot of a program code

Description automatically generated

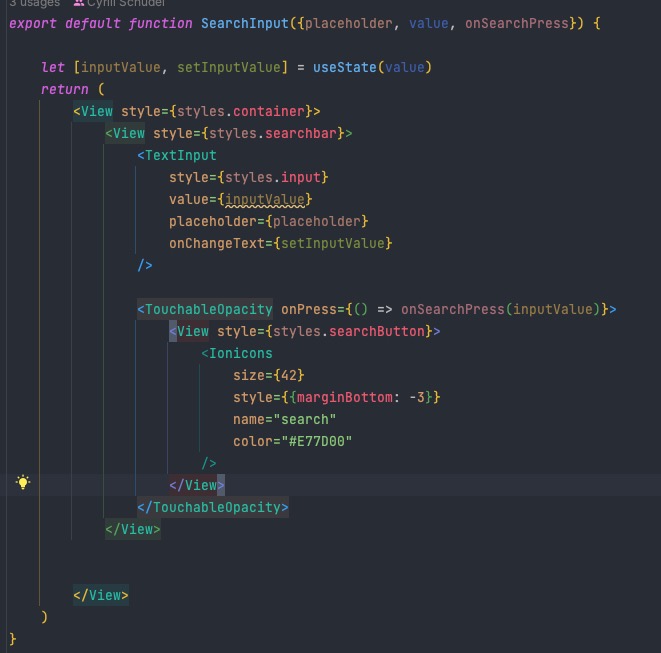
Dieser Code hat die Logik des Search Screens. Man sieht, wie am Anfang ein paar States definiert werden, welche gebraucht werden. Danach beginnen wir auch schon mit dem Request, wo wir nach dem Suchen, was wir im Textfeld eingegeben haben. Falls der Inhalt im Textfeld noch undefined oder ein leerer String 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.

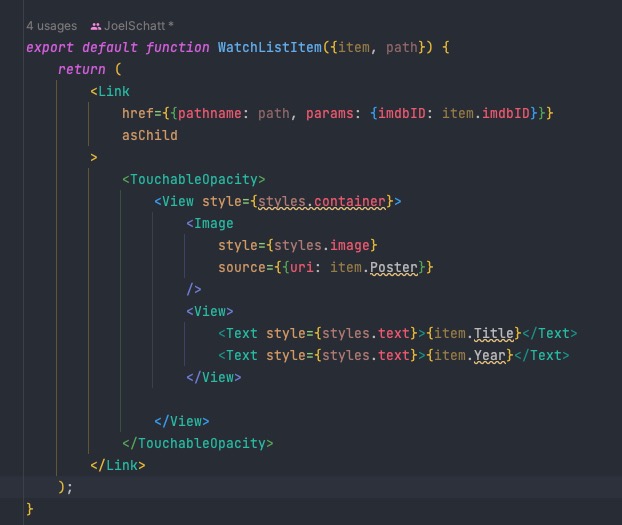
****

Damit man 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. 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.

**Async Storage**

Damit die Watchlist auch nach einem Neustart der Applikation noch vorhanden ist, speichern wir die Items von der Watchliste als String im Async Storage, sobald es per Knopfdruck in die Liste eingefügt werden soll. Falls das item bereits im Async Storage gespeichert ist, wird es hingegen herausgelöscht.

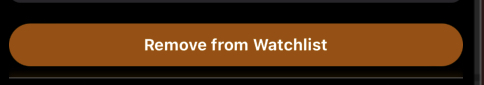
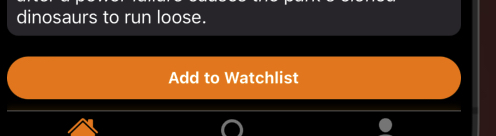
**Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung**

Auf der Home-Seite werden die Items wieder herausgelesen, zu JSON konvertiert und in einer FlatList dargestellt.

**Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung**



Zudem wird dieser Storage dafür verwendet, um den Anfangsstatus des WatchListButtons zu setzen. Wenn der ausgewählte Film bereits in der Watchlist ist, soll «Remove from Watchlist» auf dem Button ersichtlich sein, ansonsten soll «Add to Watchlist» angezeigt werden.

Testing

|  |  |
| --- | --- |
| **Abschnitt** | **Inhalt** |
| ID | Testfall: 00001, ST |
| Anforderungen | * Ich als User möchte nach Filmen suchen können |
| Vorbedingungen | Die 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) 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 soll 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Person / Datum | Erfolgreich | Bemerkungen | |
| ST-001 | Noah & Sameer | Ja | Die Suche bei ihnen lief grösstenteils einwandfrei. Es war aber so, dass wenn man am Ende noch ein Leerzeichen in der Suche drinstehen hatte, dass dann die Suche nicht funktionierte. |
| ST-002 | Noah & Sameer | Ja | Das hin und zurücknavigieren in die Detailansicht und wieder zurück hat sehr gut funktioniert. Noah hat es aber geschafft einen Film zu finden, wo die Detailansicht leider nicht geladen werden konnte. Das liegt wahrscheinlich daran, dass der Film, welcher er herausgesucht hatte ein sehr unbekannter Film ist und vielleicht keine mehr gültige IMDb ID mehr besitzt. Müsste man noch in einem zukünftigen Release noch abfangen. |
| ST-003 | Noah & Sameer | Ja | Noah und Sameer konnten die Filme sehr gut in die Watchlist hinzufügen. Es gab aber ein kleiner Fehler in der Anzeige der Buttons. Teilweise stand auf dem Button “Add to Watchlist” obwohl der Film bereits dort in der Watchlist vorhanden ist. Dieser Bug konnten wir aber dann zum Glück beheben und ist jetzt kein Problem mehr. |
| ST-004 | Noah & Sameer | Ja | Wie auch schon bei Testfall ST-003 funktionierte diese Funktion einwandfrei, aber auch dort gab es den Fehler mit der Anzeige des Buttons. Nichtsdestotrotz wurde dieser Fehler auch behoben und sollte jetzt kein Problem mehr darstellen. |

Fazit

**Was lief gut/schlecht?**

Was sicher gut lief, war unsere Zusammenarbeit. Wir verstanden uns gut und waren auch beide gewillt unseren Beitrag zu diesem Projekt beizusteuern. Beim Verteilen der einzelnen Arbeitspakete gab es auch absolut keine Streitereien oder Unstimmigkeiten, was natürlich einen sauberen Fortschritt sicher stark begünstigt hat. Ebenfalls hatten wir wenige Fehler und Bugs, welche uns Zeit raubten. Somit konnten wir praktisch immer zügig vorwärts arbeiten und zeitgerecht unser Projekt abschliessen.

Wir müssen auch sagen, dass nichts wirklich schlecht lief. Es war eine sehr angenehme Projektarbeit, welche einfach lief und es keine Komplikationen gab.

**Wie seid ihr mit dem Endergebnis zufrieden?**

Mit dem Endergebnis sind wir zufrieden. Wir konnten alle Features, welche wir umsetzen wollten, auch umsetzen. Selbstverständlich sehen wir, dass die App noch viel Verbesserungspotenzial hat. Z. B. einen Screen, wo man sieht, welche Filme man bereits angeschaut hat etc. Trotzdem sind wir zufrieden, vor allem wenn man in Betracht zieht, dass wir vorher keine React Kenntnisse hatten.

**Was habt ihr gelernt?**

In diesen 5 Tagen war eigentlich alles, was mit React zu tun hatte neu. Vor allem am Anfang brauchte es ein wenig Eingewöhnungszeit, aber als wir dann drin waren, lief es gut und es war auch interessant. Es war für uns eine komplett neue Erfahrung eine mobile App zu entwickeln und wir werden dies als spassige Erfahrung in Erinnerung behalten.

**Was war schon alles vorhanden?**

Vorhanden war das Unterrichtbeispiel mit den Star Wars Planeten, welches gewisse Ähnlichkeiten mit teilen unserer Applikation aufweist. Dies haben wir in unsere Applikation eingebaut und nach unseren Bedürfnissen angepasst und gestylt. Ansonsten war die IPA vorhanden, so mussten wir uns nie Gedanken über Filmdaten machen, was natürlich sehr viel Arbeit abgenommen hat. Ansonsten der Rest ist selbst gebaut und selbst implementiert.