

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Test Api](#)

[Task 4: Implements Firebase Login](#)

[Task 5: Persistence](#)

[Task 6: Write App Logic](#)

[Task 7: Create a Widget](#)

[Task 8: Last steps](#)

**GitHub Username:** Utnapishtim86

## Tonight Movie

### Description

Tonight Movie let you find your next favourite film!

With Tonight Movie you can ask for a movie like your favorite ones, join the community and help others or ask for help.

You don't need anymore to ask your friend what to watch tonight!!

### Intended User

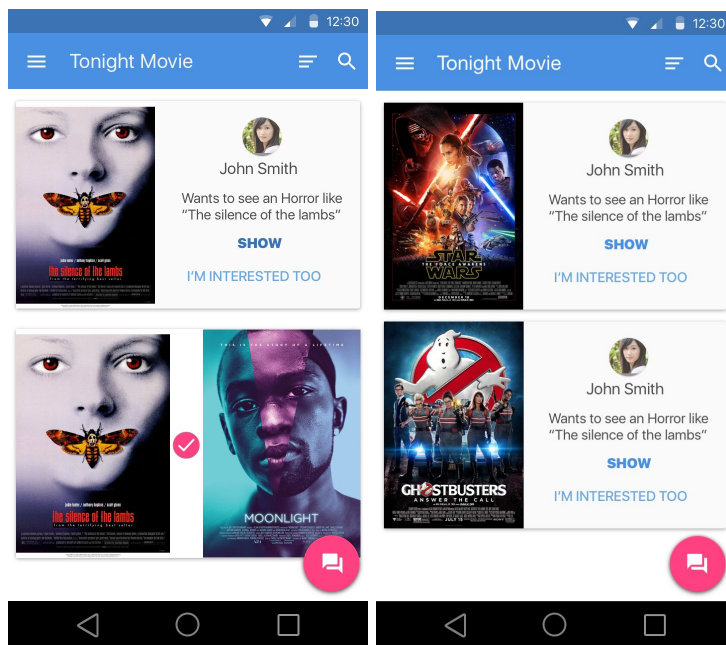
Movies addicted or for a person that want to find a movie to watch tonight

## Features

- Find information about films
- Save your favorites films to the cloud and access them with any mobile devices
- Save your watchlist locally on the device
- Ask the community to find movies to watch
- Help others to find their next favorite film

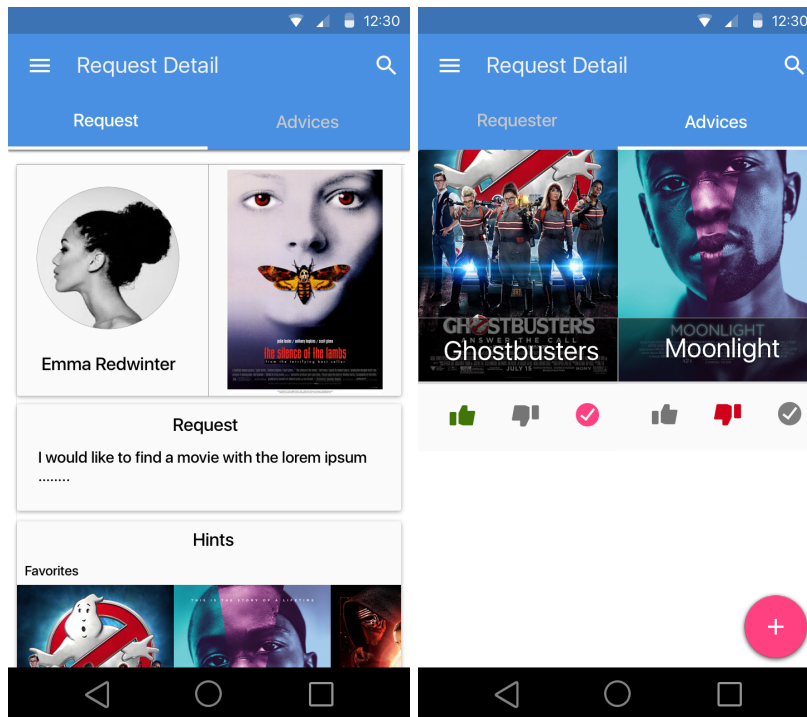
## User Interface Mocks

### Feed



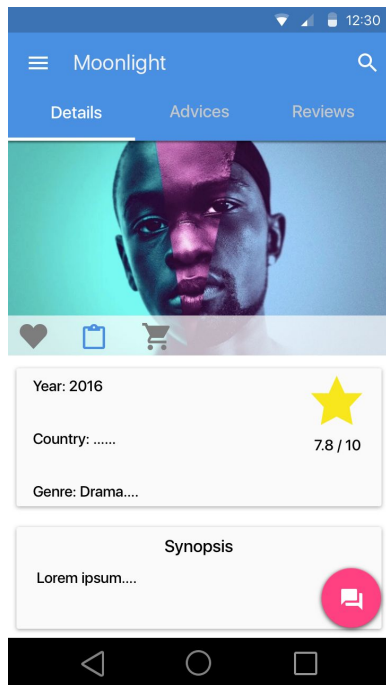
This is the home of the app (after login). The user can click on the fab to ask for a film suggestion. Can also filter the feed or search for a film (action bar)

## Feed card detail



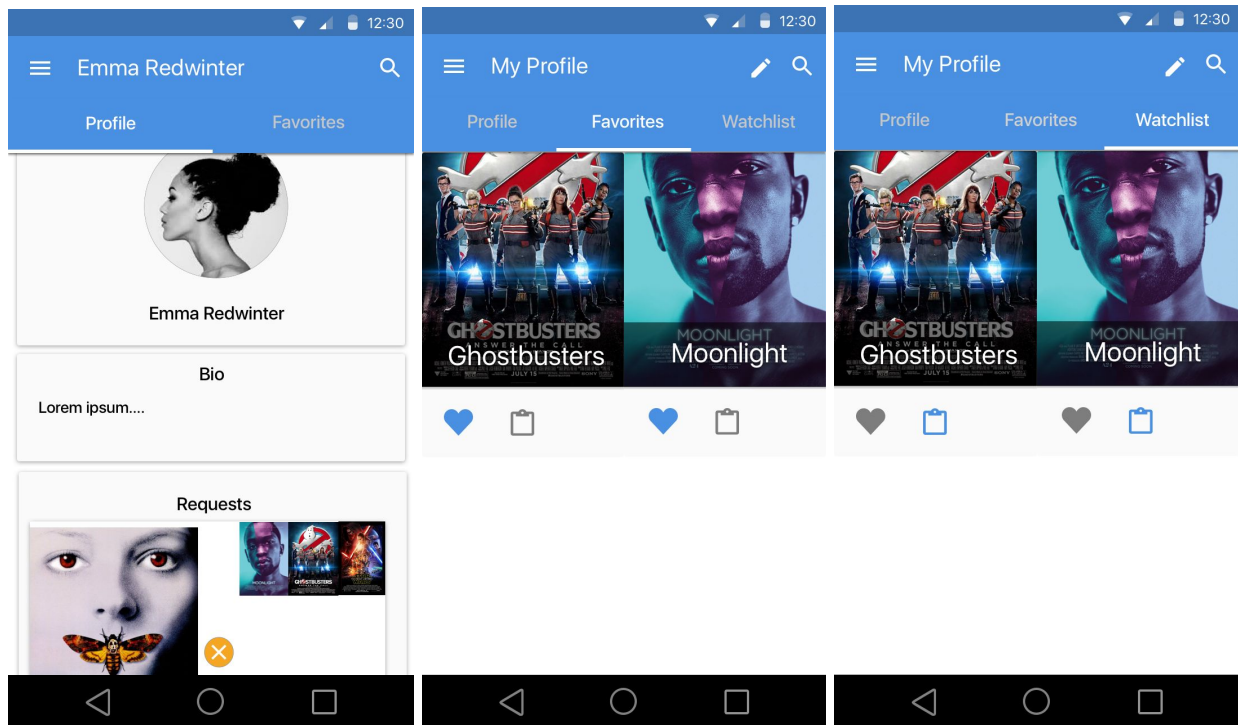
This is the detail of the request. Here the user can see informations about the user, like his favorite movies. The advices tabs are the movies suggested by other users, any user can vote the suggestions, only the requester can accept a suggestion. Other users can add advices clicking on the fab.

## Movie Detail



This page contains the movie details, the suggestions related to this movie and the reviews, clicking on the fab the user can create a suggestion from this movie

## User profile



This is the profile screens. If a user check his profile there is also a “watchlist” tab (showing locally saved watchlist)

## User's content forms

The image displays two side-by-side mobile app screenshots. The left screenshot shows a 'New request' form with fields for Title, Description, Film similar to, and Genre, each with a character count. The right screenshot shows a 'New Advice' form with a movie poster image and a text input field. Both forms have a 'SEND' button at the bottom.

**New request form:**

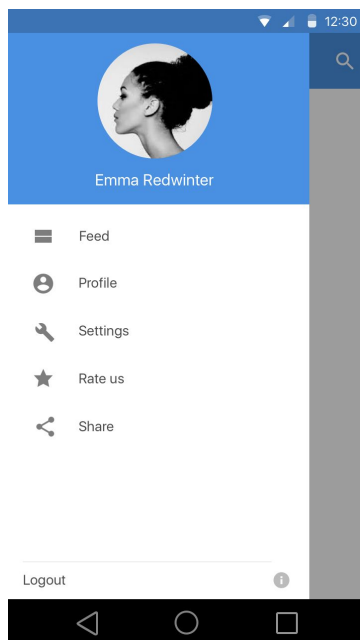
- Title: Lorem ipsum dolor sit amet, consectetur adipiscing elit. (50/50)
- Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam gravida venenatis accumsan. In mi massa, tempus lorem ipsum dolor si (110 / 120)
- Film similar to: Autocomplete textfield
- Genre: Spinner
- SEND button

**New Advice form:**

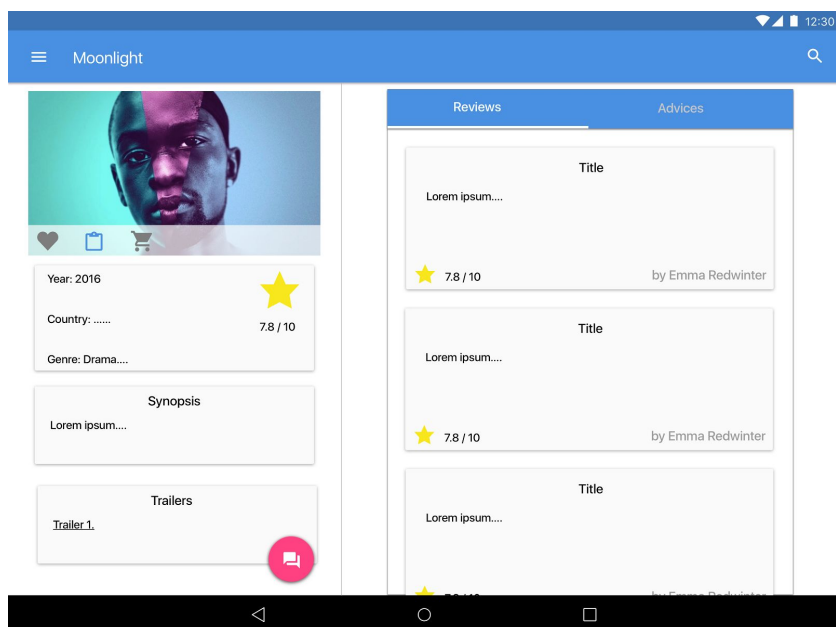
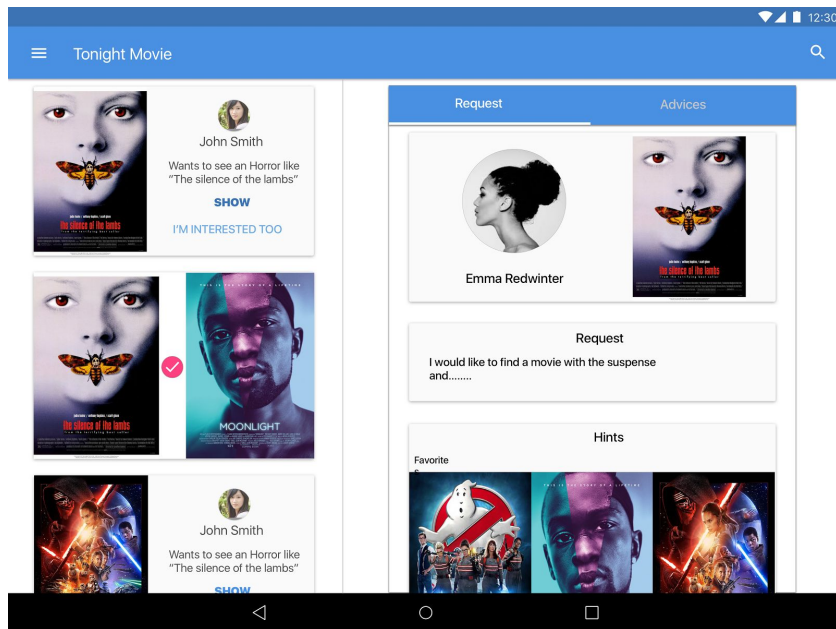
- Movie poster image (Ghostbusters)
- Text input field
- SEND button

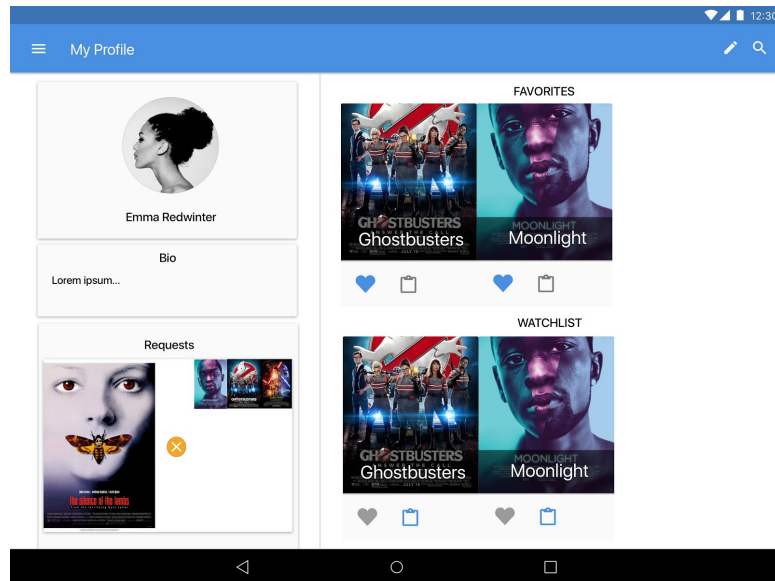
These are the form to create a new request and to create a new advice for a request.

## App wide Navigation Drawer



## Tablet screens





## Key Considerations

How will your app handle data persistence?

There are two types of data persistence. One online, I will use firebase. One offline (user's watchlist), and for that i will use a content provider

Describe any corner cases in the UX.

Navigation will be implemented with a navigation drawer

Describe any libraries you'll be using and share your reasoning for including them.

I will use fresco for the images, retrofit for the communication with a rest api (movie's database), firebase to communicate with the backend. For the content provider I will use Schematic

Describe how you will implement Google Play Services.

I will use google analytics and google adMob (the user will see an ad when he creates a request or accepts a suggestion



## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Project Setup

- Configure Firebase
- Find a good movies api (with reviews)
- Configure libraries

### Task 2: Implement UI for Each Activity and Fragment

- Built any UI from the mockups, using fragment (to simplify the tablet implementation)
- Create dimensions variable in dimen.xml
- Test mockups with fake data

### Task 3: Test Api

- Test api endpoints;
- Implements calls with retrofit

### Task 4: Implements Firebase login

- Write login logic;
- Test login
- Configure Firebase
- Create User's variables app wide (Application class)

### Task 5: Persistence

- Create the content provider to manage user's watchlist

### **Task 6: Write App Logic**

- Create a basic AppCompatActivity, it will manage the navigation drawer and all the logic common to all activities;
- Create test data to test the app;
- Create every activity that extends the basic AppCompatActivity;

### **Task 7: Create a Widget**

- Create a widget with the watchlist of the user

### **Task 8: Last Steps**

- Change Firebase security rules;
- Add proguard;
- Publish App to the app store;