Flixat Design document

Structure

7 screens

- Main popular movies list
- Now playing list customized on user's location
- Search screen
- Preferences
- About screen with attribution to apis
- Detail screen with additional information about the user's selection

Data

- Network responses and other networking handling
- Local data, Room and Room entities
- Repository implementations

Domain

- Repository interfaces
- Domain models

UI

- Single activity, multiple fragments
- ViewModels
- Binding adapters

Library usage

Paging 3

- Paged data is available for main lists and search
- Popular list will use remote mediator to cache network data in case of no connection for offline use

Koin

- Dependency injection

The application will be using MVVM architecture so several other androidx libraries will be used such as ViewModel, Lifecycle, and Navigation components. For networking Retrofit will be used. Coil will be

used for image loading. Databinding will be used to populate the views with data retrieved by the viewmodel.

APIs

Flixat will use tmdb.org for movie data. It will use JustWatch apis for providing watch provider information. Fandango apis for showing local movie theater results based on the user's location.

System Hardware integration

Flixat will use user's location to get now playing information from tmdb specific to the user's country. Location permissions will be handled at the time that the user navigates to the now playing screen.

Storing data

Flixat will use Preferences Datastore for saving preference data that persists across sessions. Flixat will use Room database to cache Popular movie data so that it will be available offline and updated whenever possible.

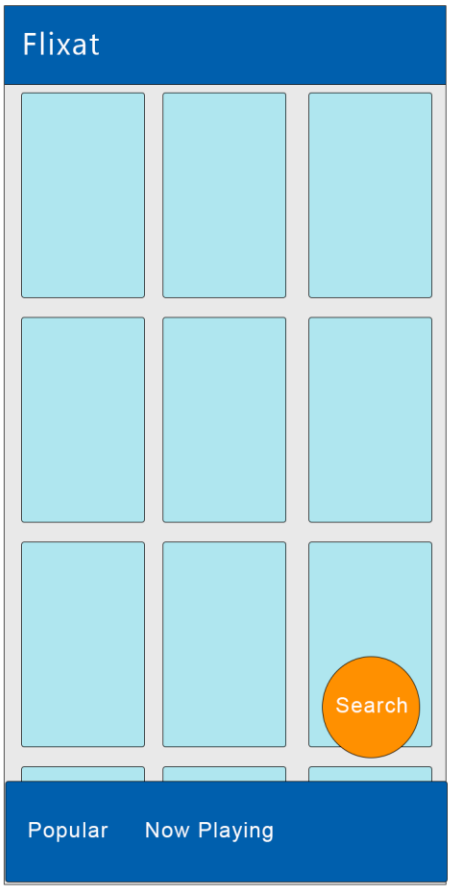
Milestones

The first screens will be based on the initial application structure to make use of the tmdb apis. The popular list and now playing lists will have the same layout so these screens will be implemented first. This will give basic functionality to the application in a usable although basic state. Search functionality will be added next. Then the detail view will be implemented. At this point the application will be usable but more features included in the basic state that can continue to be iterated upon. Preferences will be added, the about screens, then finally the onboarding screen to polish up the package. Testing will be implemented across layers as the application continues to grow so that regressions aren't introduced as more features are added.

Motion Layout

Flixat will use Motion Layout to animate an onboarding screen on the user's first interaction with the application.

Basic mockups of main screens



Main screen

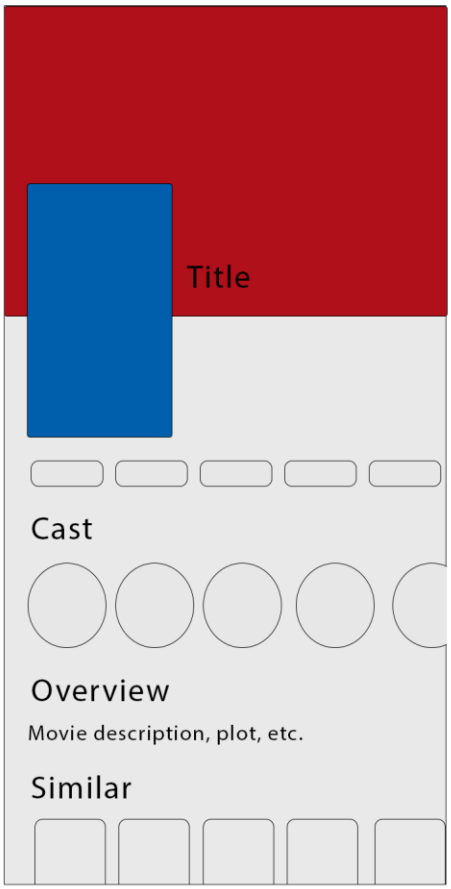
Popular movies from tmdb listed in recycler view with image.

Search fab on main screen

Menu selection of popular movies or now playing in theater in bottom appbar.

Now playing screen

Same layout at popular screen, only showing now in theaters based on user location vs. Popular movies.



Detail screen

Shows detailed information about the movie that was selected. Full screen backdrop, movie image, cast, similar movies, where movies are playing, watch providers if online.