

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Screen 7](#)

[Screen 8](#)

[Screen 9](#)

[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: Setup Content Provider and SyncAdapter](#)

[Task 4: Handle Error Cases](#)

[Task 5: Add Widget](#)

[Task 6: Add Notifications](#)

[Task 7: Implement Google Play Services](#)

[Task 8: Setting Wallpaper](#)

GitHub Username: [Anuradha Ohm](#)

MovieCorner

Description

This app helps decide movies to watch easily by

1. Searching your favorite movie by title
2. Browsing movies based on Upcoming, Now playing, Top Rated, Most Popular, Kids Movies.

and makes the device more lively with these features!

1. Add widget for top ten box office hits to Home screen
2. Notifications for movies to be released this week (in favorite genre)
3. Set favorite movie poster as device wallpaper

Intended User

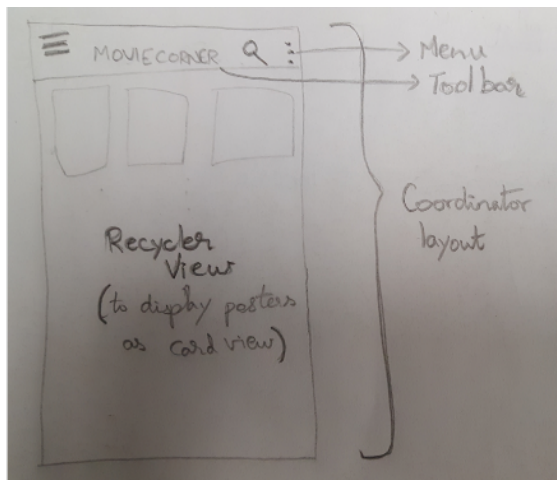
Movie lovers or anyone who wants to decide what movies to watch from anywhere anytime and be up-to-date on the latest top movies

Features

- Search a movie
- Widget for top ten box office hits
- Saves favorite movies
- Sets favorite movie poster as wallpaper
- Notifies based on a chosen genre

User Interface Mocks

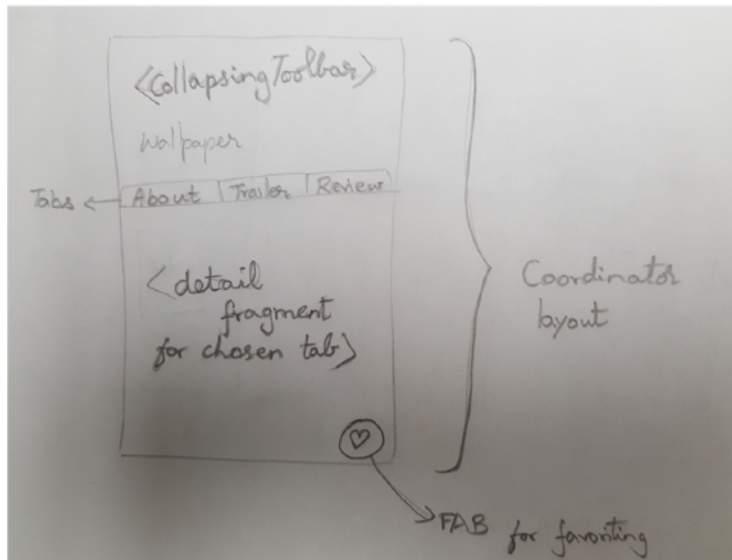
Screen 1



Main Screen of the app.

It has a navigation drawer to display various sort options, favorites, help, share, etc. According to the option selected the movies will be displayed as card views in a recycler view in this screen. Movie can be searched by title using the search widget on the toolbar. Settings - for choosing the genre for notifications will be displayed in the menu. Banner ads will be displayed in this screen.

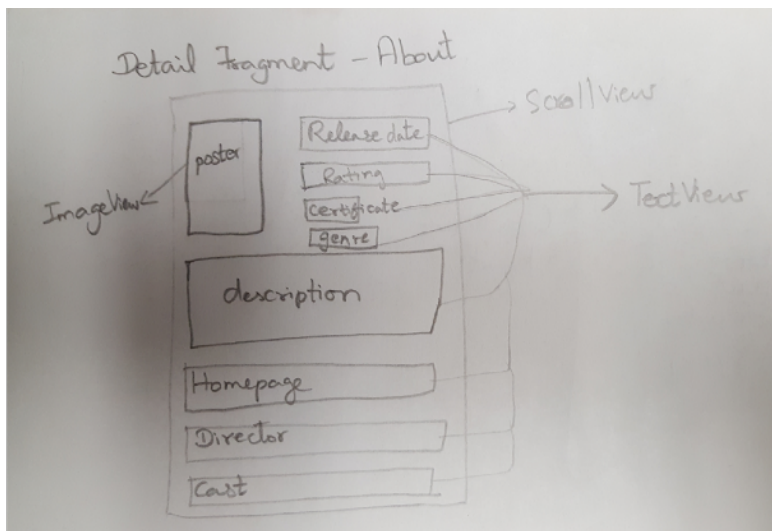
Screen 2



Detail screen

When a movie is selected, its details are shown in the detail activity. Wallpaper is displayed in a collapsible toolbar. The details are shown in tabs-about, trailer and review. All of this is contained in a coordinator layout.

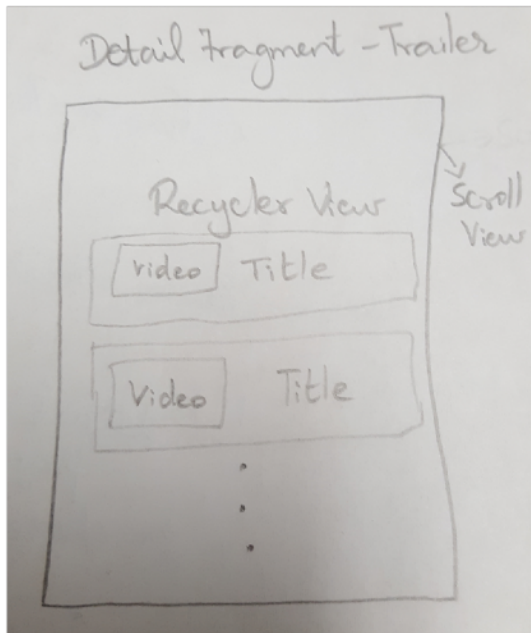
Screen 3



Detail fragment for About tab

When a movie card is selected, the details of the movie - poster, release date, rating, genre, description, homepage, director, etc will be shown in its about tab in a scrollable linear layout.

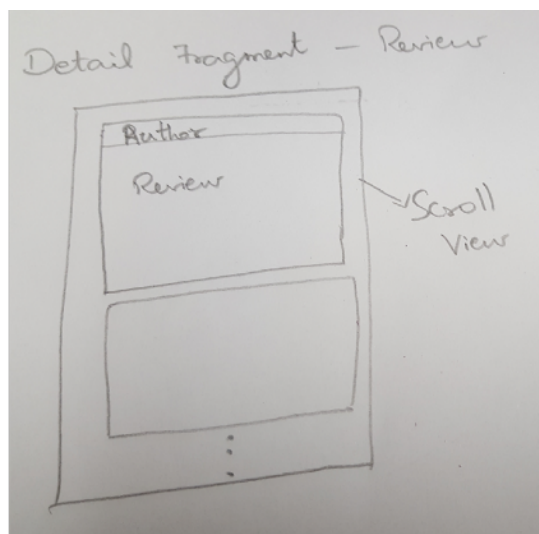
Screen 4



Detail fragment for Trailer tab

When a movie card is selected, the trailers of the movie will be shown in its trailer tab as card view.

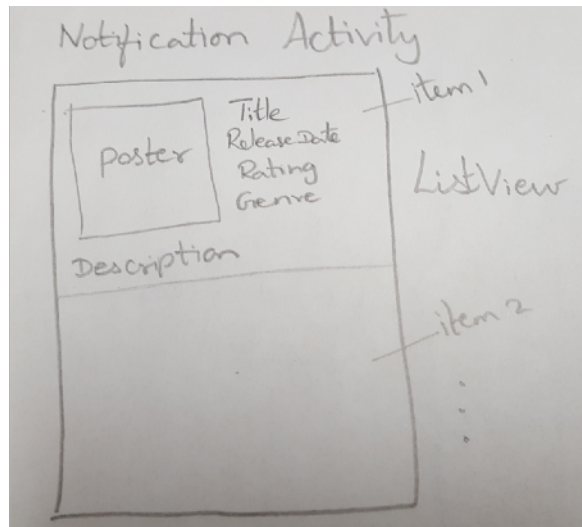
Screen 5



Detail fragment for Review tab

When a movie card is selected, the reviews of the movie will be shown in the review tab as a cardview.

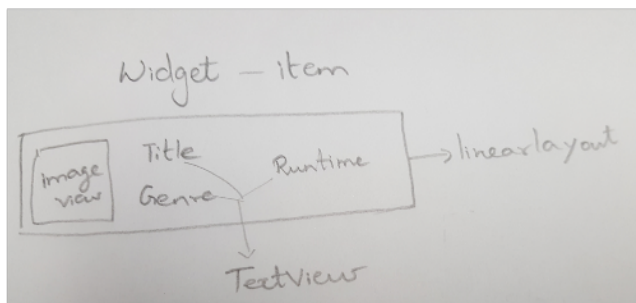
Screen 6



Notification Activity

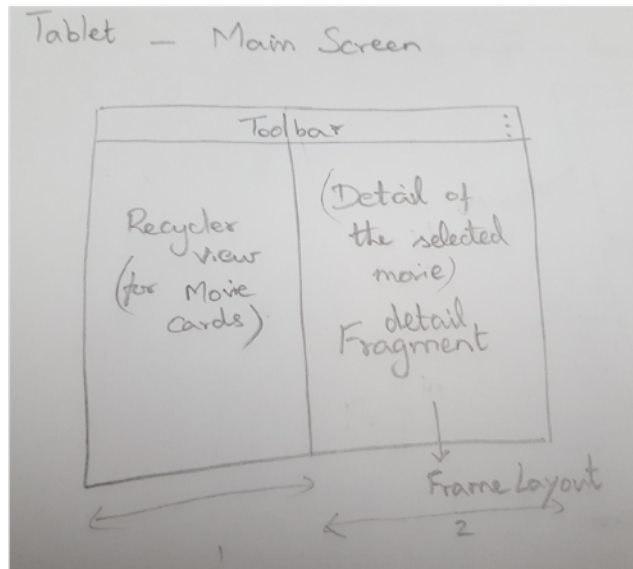
When the notification is selected, user will be directed to this activity. The details of the movies in the notification will be displayed in a list view.

Screen 7



The layout for the widget list. It is a linear layout with a poster, title, genre and runtime.

Screen 8



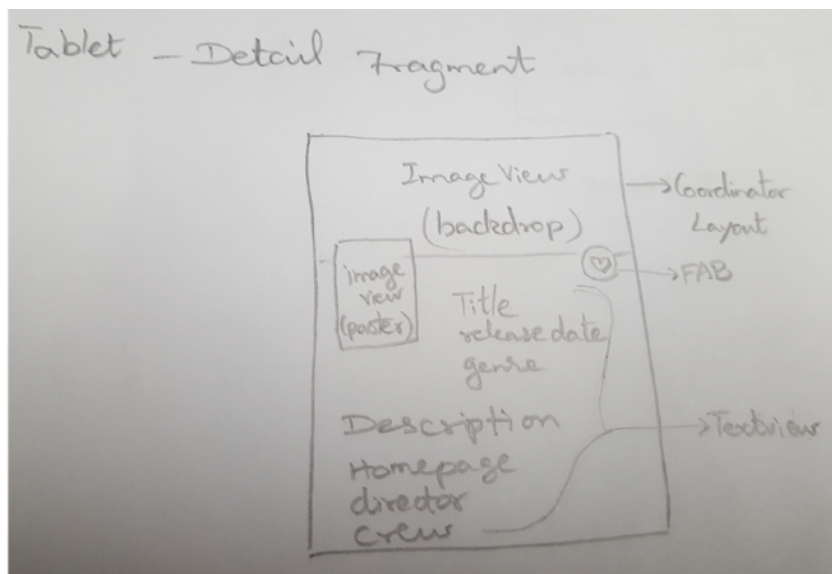
For tablet - Main activity

It is a linear layout with *Fragment*- for displaying the movie cards and *Framelayout* - for the details of the selected movie.

The fragment will be a recycler view with movie cards.

The toolbar will have a navigation drawer to display the sort options, favorites, share, notification activity, etc. There will be a search widget on the toolbar too.

Screen 9



For Tablet - detail fragment

When a movie is selected in the left pane, its details will be displayed in this right pane. It will be a coordinator layout will collapsible toolbar. The wall paper will be displayed in the toolbar. The poster and FAB will overlap some of the wallpaper.

Other than the details of the movie, the [trailer](#) will be displayed as horizontal recycler view and [reviews](#) will be displayed below it.

Key Considerations

How will your app handle data persistence?

Favorited movies will be saved to the SQLite database which will be implemented using content provider (will be coded). So, the favorited movies will be viewable offline. The movies for the widget(box office hits) are stored in the content provider. The last browsed category of movies will be stored in this database too. However, as internet connectivity is required to establish connection to the movie database, search and browsing movies will not load without internet.

Describe any corner cases in the UX.

1. Default category of movies will be the ones currently in theatre.
2. The interval to sync the movies (from internet) is every 6 hours
3. The notifications will be updated once every day
4. Up to 100 movies will be displayed for each category
5. When a poster for a movie is not available, "poster unavailable" might be displayed in that image view
6. For tablets, in two pane mode, a movie has to be selected for its details to be displayed in the detail pane.

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

1. Muzei - for displaying wallpaper
2. Picasso - to handle the loading and caching of images.
3. Materialish-progress - For progress wheel
4. Stetho - for debugging the content provider

Describe how you will implement Google Play Services.

1. Google Play Services - AdMob
Banner ads will be implemented in the main screen.
2. Google Play Services - Analytics
Analytics will be added to see geo information and accordingly add other language movies too. Also could analyze the preferred screen users navigate to and include/exclude features of the app.

Next Steps: Required Tasks

Task 1: Project Setup

- Create an android studio project (Basic Activity with fragment)
- Configure libraries in the app's gradle (muzei, picasso, progress wheel, etc)
- Add internet, network permissions and permissions for using sync adapter to Android Manifest file.
- get API key from www.myapifilms.com and themoviedb.org
- decide on the api's to be used for updating widgets, notifications, browsing various categories and for searching movies.
- collect the icons required and decide on a icon for the app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity with Navigation View - to displays the posters for the movies according to the selected sort order.
- Build UI for movie details screen with Tab layout (3 tabs - for About, Trailers, Reviews)
- Build UI for help page and notifications list page.
- Build UI for details screen on landscape mode.
- Build UI for tablets (two pane mode)
- Create menu items.
- Choose the colors for primary, secondary, text, accent, etc.

Task 3: Setup Content Provider and SyncAdapter

- Decide on the column_names for the data to be stored.
- Code the contract, dbHelper and content provider (including bulk insert)
- Setup Sync Adapter to update content provider every 6 hours or when the category selection changes.

Task 4: Handle Error Cases

- Handle no network error with error string in string.xml.
- Handle error when the data fetch fails or when there are no movies for a search.
- Handle error when no apps to support messaging or emailing (for share options)

Task 5: Add Widget

- Declaring an App Widget in the Manifest and Adding metadata.
- Build UI for widget.

- Setup widget collection using remoteviewsservice.

Task 6: Add Notifications

- Setup inbox style big view for notifications.
- Build UI for notificationsActivity - to be opened when notification is clicked.
- Setup to show notifications for a selected genre or show no notifications at all, based on settings in SettingsActivity.

Task 7: Implement Google Play Services: AdMob, Analytics

- Add play services dependencies to gradle.
- Set the main activity to display banner ads.
- Signin to AdMob account and use the ID so real ads can be displayed when the app is in production (for development test Ad will be used)
- Setup the Google Analytics account.
- Configure analytics in the app and implement the tracker and application class.

Task 8: Setting Wallpaper

- Use Muzei to display a favorited movie's poster as wallpaper.
-