

Pratap Surendran Ganesan

iMovie

SLOGAN:	3
STATEMENT OF PURPOSE:	3
LIST OF SELLING POINTS:	3
Example Use case:	3
Research:	3
IMDB:	3
SONY REWARDS:	4
QuickThoughts - Earn Rewards	6
TodoMovies 4	6
iMovies vs Sony Rewards vs IMDB vs QuickThoughts vs TodoMovies 4	7
REST BASED API	8
List of currently airing movies:	8
Sample response:	8
DESIGN PRINCIPLES:	9
CONTRAST:	9
REPETITION:	9
ALIGNMENT:	10
PROXIMITY:	10
DEFERENCE:	10
CLARITY:	11
DEPTH:	11
Visual Use case(Wireframe)	11
Use case to view Detail page	11
2) Use case to Redeem points.	15
3) Use case to share movie	15
Implementation of multiple view:	16
Adaptive Layout(Size classes):	16

SNO	DATE	DESCRIPTION
1	01/08/2017	Initial Draft
2	0810/2017	Final version

iMovie

SLOGAN:

Know everything about the latest released and trending movies.

STATEMENT OF PURPOSE:

iMovie provides the platform to stay updated with the latest released and trending movies, and to get rewarded by sharing your favorite movies in Facebook and by liking those movies in iMovie.

LIST OF SELLING POINTS:

A way to get rewarded with points for the entertainment. Earn your points while you like and share your list of movies in Facebook. Grab free vouchers and coupons with your earned reward points. Let the world know your wish list and you know the list of best movies out in the world; thereby, without missing the best entertainment.

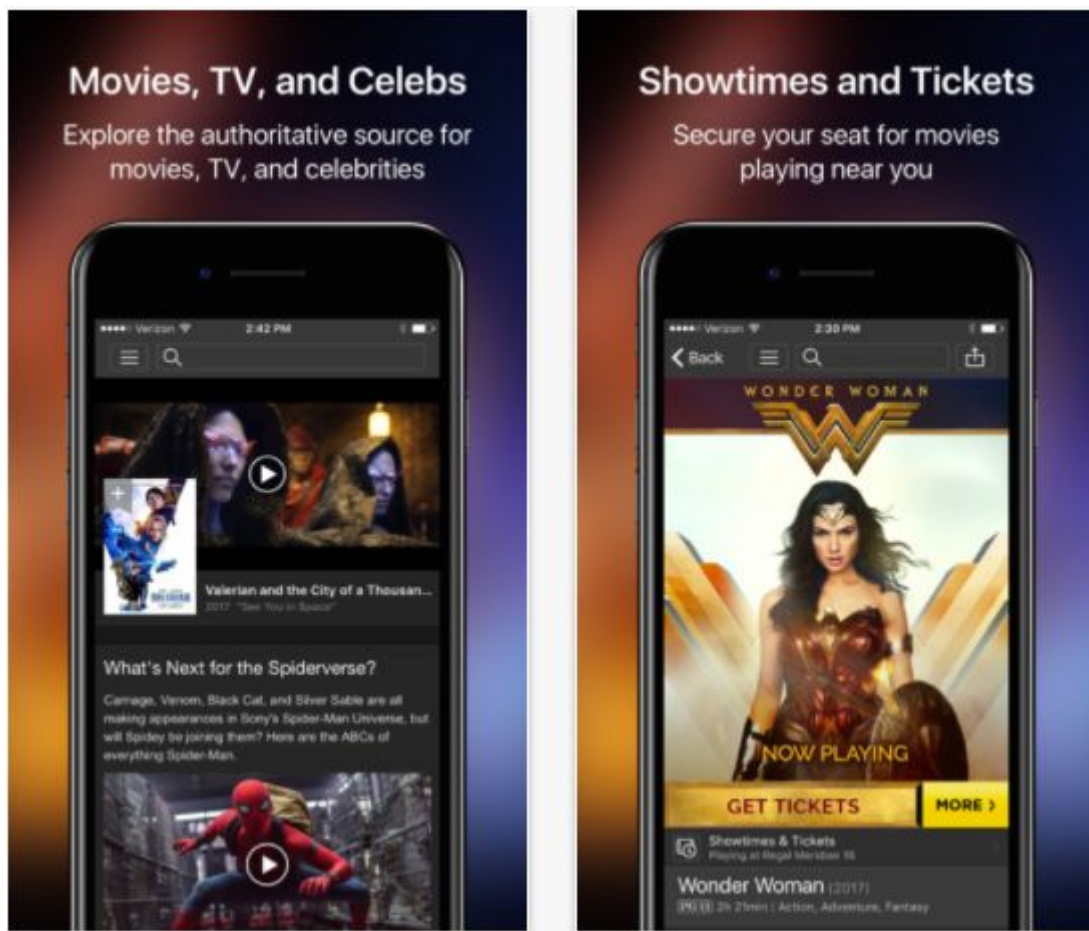
Example Use case:

- 1) User access the list of currently airing movies by accessing iMovies mobile application.
- 2) User earns the reward points by Liking the movie.
- 3) User earns the reward points by playing advertisement and Youtube URL.
- 4) User Can see the reward points for their favorite shows
- 5) User redeems the earned points using redeem button

Research:

IMDB:

The world's most popular and authoritative source for movie, TV, and celebrity content.



<https://itunes.apple.com/us/app/imdb/id342792525?mt=8>

SONY REWARDS:

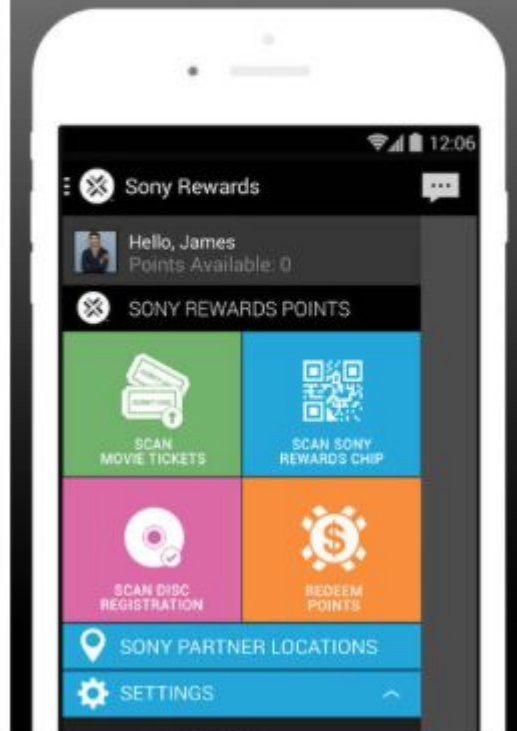
Earn points on the go. Just for doing the things you already do... like going to a movie, playing a game, or buying the Sony products you love.

<https://itunes.apple.com/us/app/sony-rewards/id908307582?mt=8>

The deets on all things Sony.

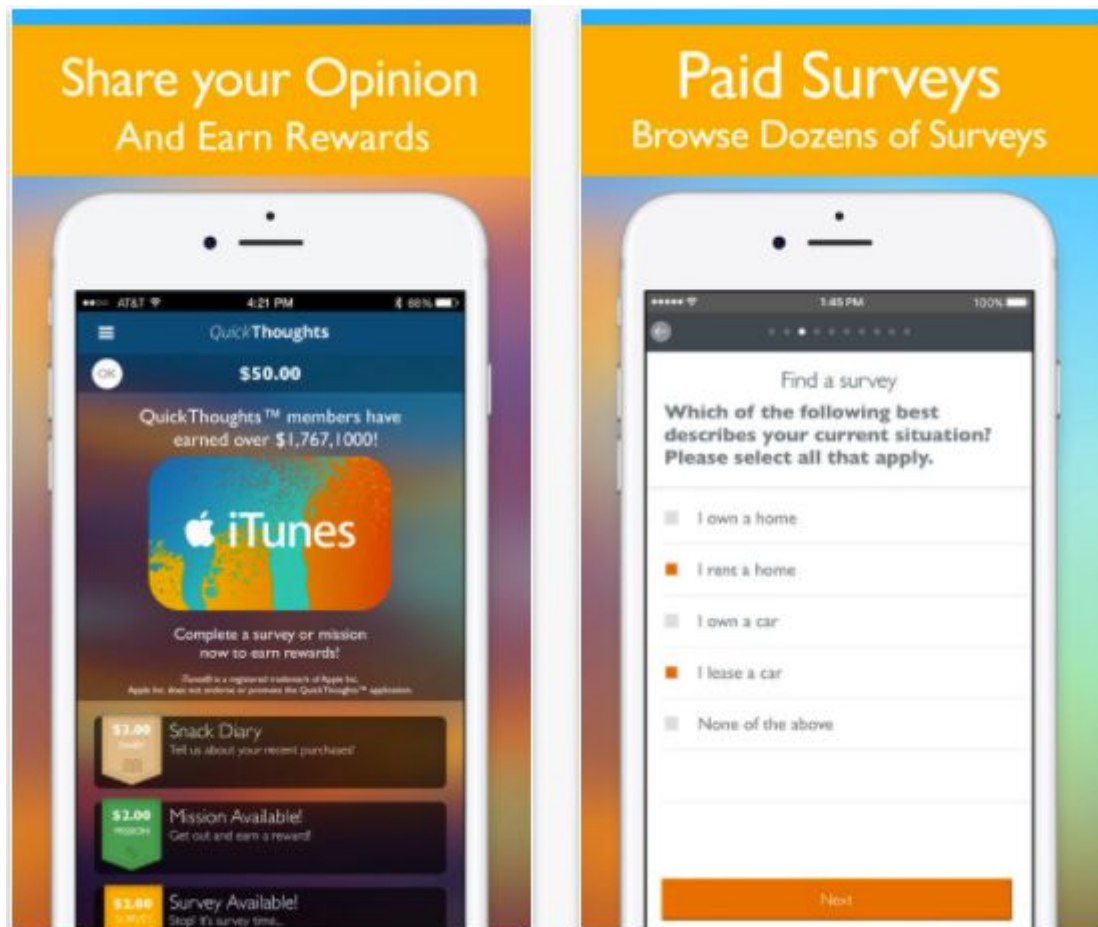


Rewards & account info,
wherever you are.



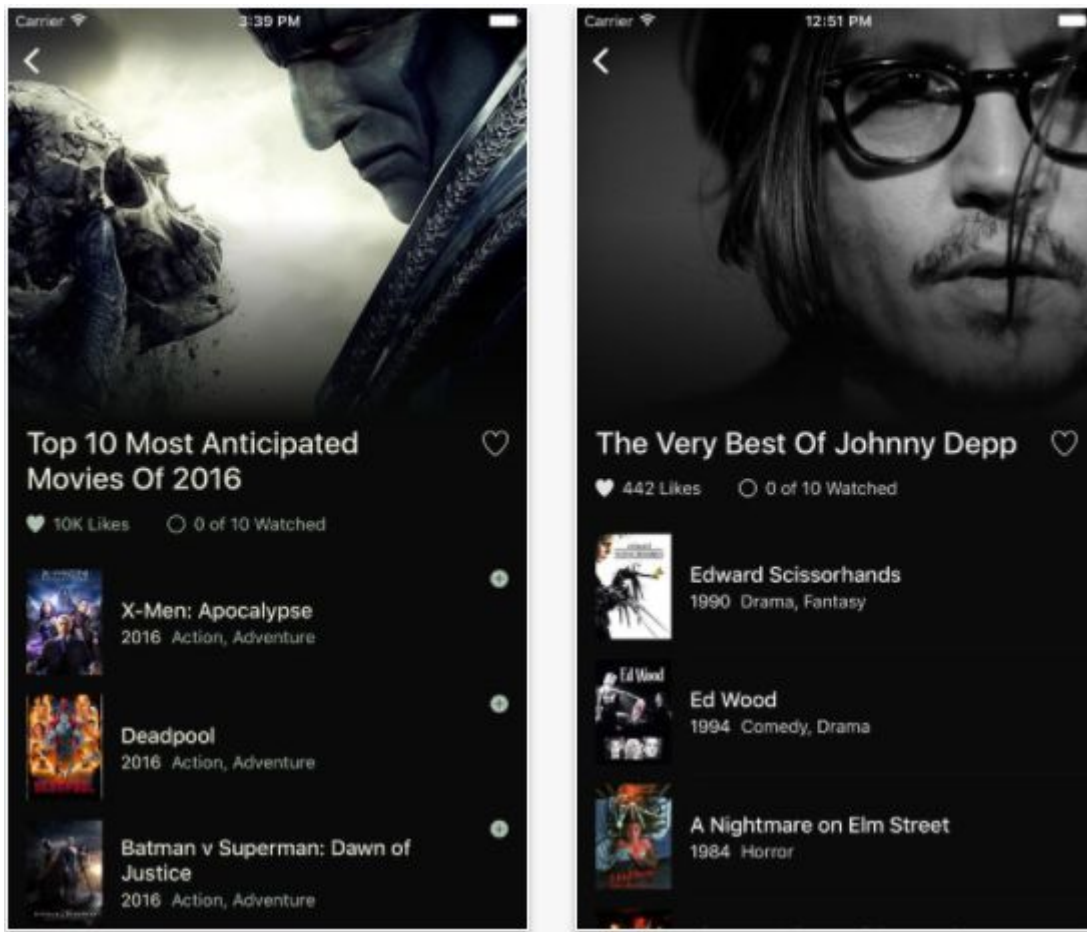
QuickThoughts - Earn Rewards

QuickThoughts rewards you with FREE iTunes Gift Cards for sharing your thoughts via quick surveys and local survey events! Earn rewards on the go and redeem them for free \$10 gift cards!



TodoMovies 4

TodoMovies is the best way to manage movies you want to watch/watched, create custom lists.



iMovies vs Sony Rewards vs IMDB vs QuickThoughts vs TodoMovies 4

iMOVIES	SONY REWARDS	IMDB	QuickThoughts	TodoMovies 4
Earn points by liking movies, playing youtube, advertisement	Multiple way to earn points	No reward point system	Earn points by participating in survey	No reward point system
Currently airing shows	No currently airing shows	No currently airing shows	No currently airing shows	No currently airing shows
Targets only Movies.	Movies, TV Shows, News, Hardware products	Movies, TV Shows, Cinema tickets.	Only survey.	Targets only movies

Trending Movies	No Trending Movies	Popular TV shows & movies	No Trending shows	No Trending movies
Redeem any kind of coupons	Discount only on sony products	No reward points system	Redeem gift coupons	No reward points system
Maintains the favorite list	No Favorite list	No Favorite list	No favorite list	Maintains the favorite list

REST BASED API

File - MovieAPI.swift

Scenes: AiringTableViewController, TrendingViewController.

REST based API [API documentation](#) provides all the necessary information about the movie.

List of currently airing movies:

https://api.themoviedb.org/3/movie/now_playing?api_key=e93538bb421ea6554bf10471e433142e&language=en-US&page=1

Sample response:

```
{
  "results": [
    {
      "vote_count": 1969,
      "id": 315635,
      "video": false,
      "vote_average": 7.4,
      "title": "Spider-Man: Homecoming",
      "popularity": 109.259031,
      "poster_path": "/c24sv2weTHPsmDa7jEMN0m2P3RT.jpg",
      "original_language": "en",
      "original_title": "Spider-Man: Homecoming",
      "genre_ids": [
        28,
        12,
        878
      ],
      "backdrop_path": "/vc8bCGjdVp0UbMNLzHnHSLRbBWQ.jpg",
    }
  ]
}
```



```

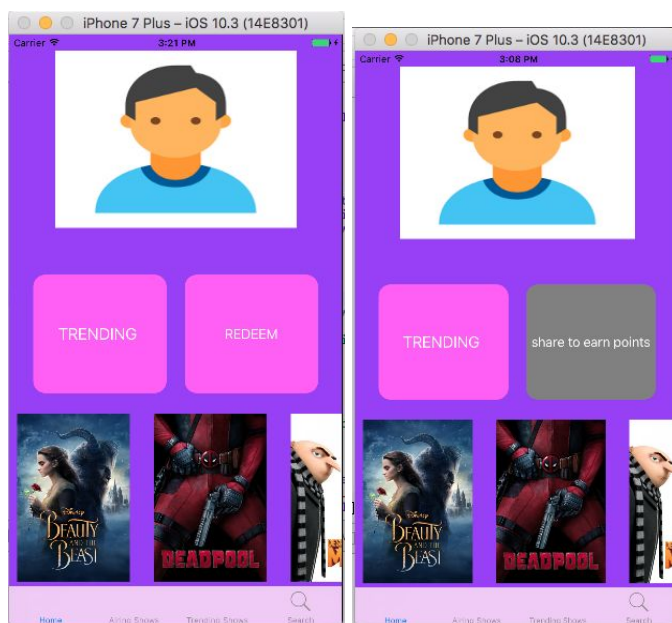
"adult": false,
"overview": "Following the events of Captain America: Civil War, Peter Parker, with the help of his mentor Tony Stark, tries to balance his life as an ordinary high school student in Queens, New York City, with fighting crime as his superhero alter ego Spider-Man as a new threat, the Vulture, emerges.",
"release_date": "2017-07-05"
}

```

DESIGN PRINCIPLES:

CONTRAST:

- 1) Eligibility of the user to redeem is differentiated with button color and the button text. If the user has no points to redeem, "Redeem" button is changed to disabled button "share to earn points"



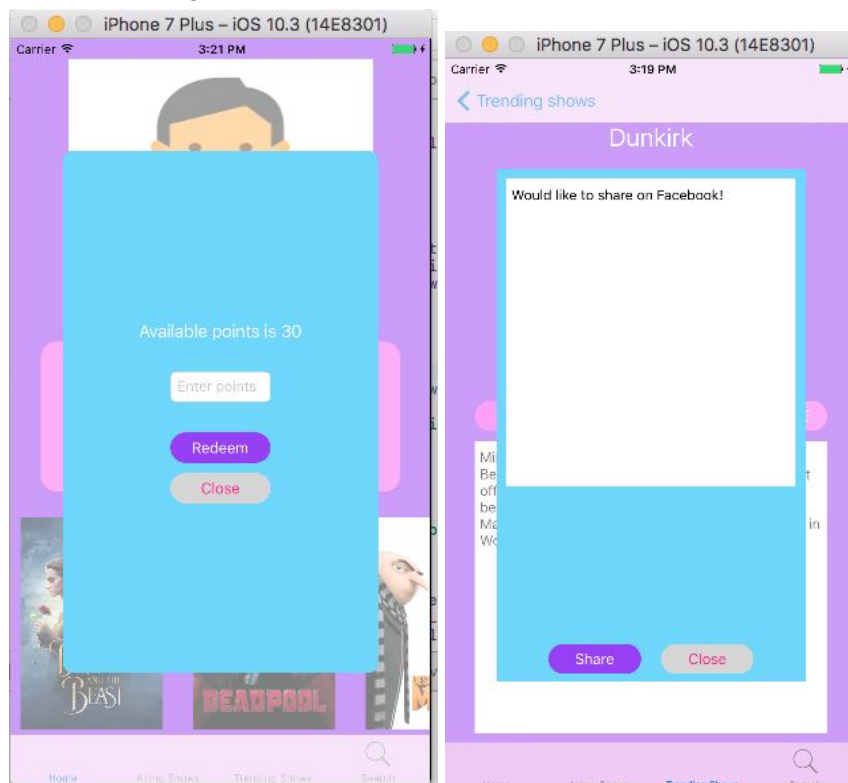
- 2) Vivid contrast is provided by creating a strong primary color Purple, Magenta, Blue, Grey
- 3) Like and dislike status is differentiated with contrast color.

REPETITION:

- 1) The color palette consists of four main colors that are repeated throughout the design of the application
- 2) Buttons are consistently round in shape. The look and feel of the button is used across all the screens
- 3) Design of the movie detail page is same in different navigation flows (search, trending, airing)

ALIGNMENT:

- 1) Most screens have center alignment.
- 2) Used size classes to take care of the alignment in landscape and portrait orientation
- 3) Sharing and Redemption popup and its content have similar center alignment.



PROXIMITY:

Group of data are naturally created with UIView ensuring the border and contrast to the background color. No buttons in the same screen representing different flow share the same color.

DEFERENCE:

- 1) Sharing and Redemption flow are created using popover ; thereby, reducing the extra screens
- 2) The Tab bar is less prominent focussing the user on the main task, while still allowing easy navigation to other features of the application
- 3) collectionView is used for trending shows, where a large set of data is expected
- 4) Carousel view to display the list of favorite list, ensures user to let them have a glance on their favorite list.

CLARITY:

- 1) Tab bar and buttons describes the clear idea about different features of the application
- 2) Greyed out button to inform the user that they do not have eligible points to redeem
- 3) Carousel to describe about the favorite list.

DEPTH:

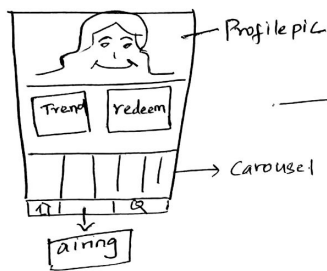
Sharing and redemption popup reveals part of the screen behind reminding the user that they were in the middle of feature flow.

Popup shows the user points while redemption.

Visual Use case(Wireframe)

Share Movies

Home Screen



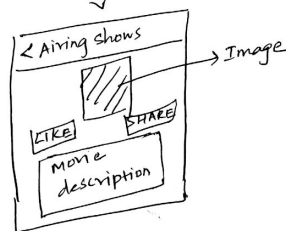
Airing Screen



Popup overlay

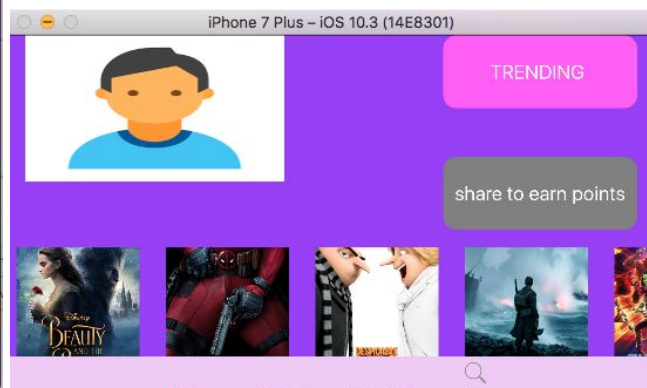
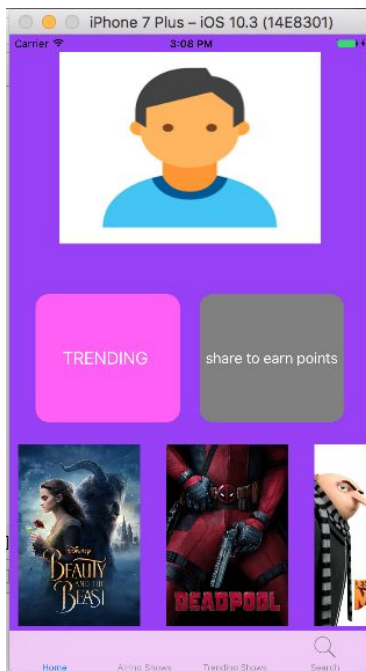


Detail view



1) Use case to view Detail page

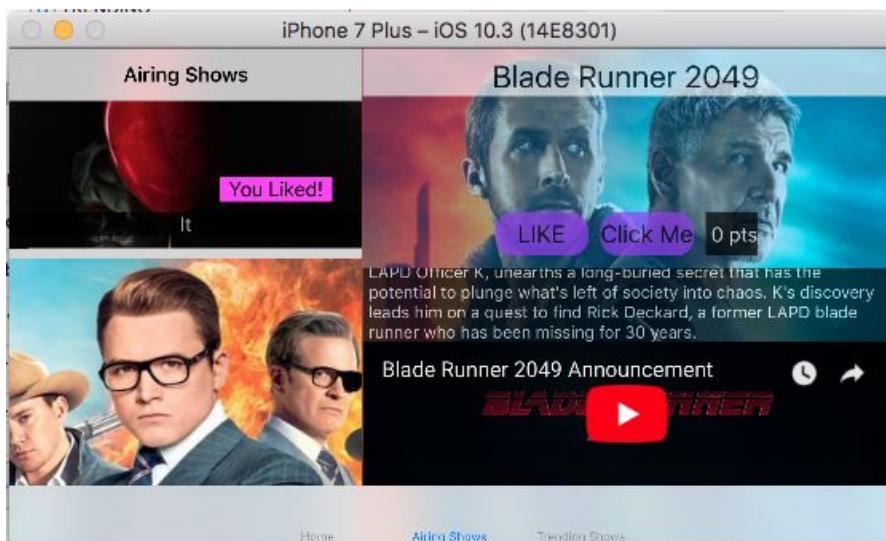
Step 1) Launch Home screen



Step 2: Navigate to Airing show(Table view in combination of split view)

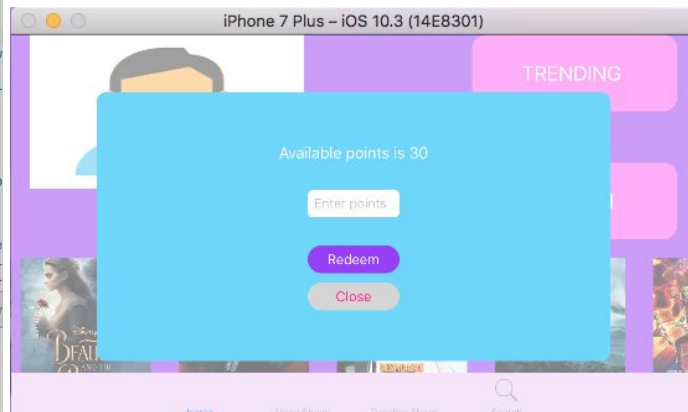
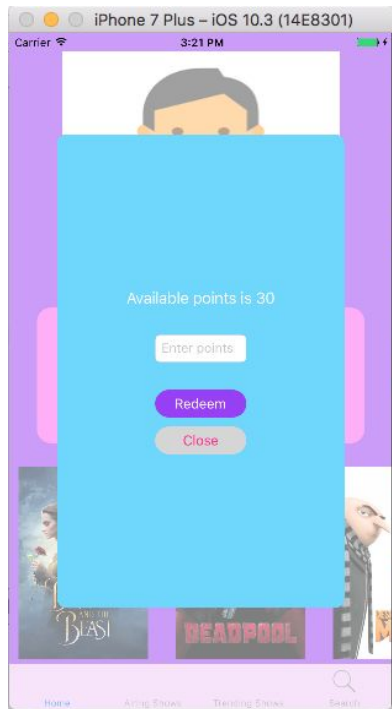


Step 3: Navigate to detail view



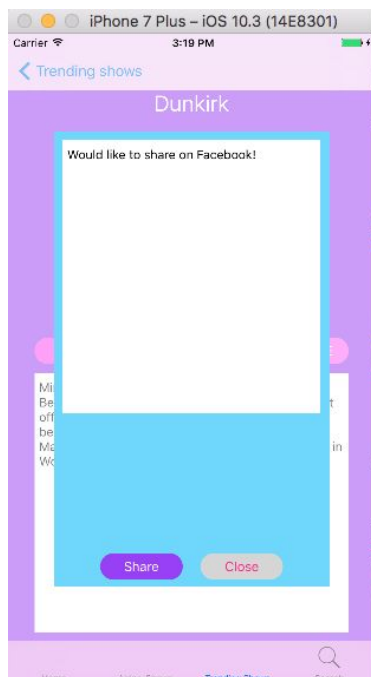
2) Use case to Redeem points.

Step 1: Tap on Redeem button to redeem points



3) Use case to share movie

Step 1) Navigate to detail page and tap on share button

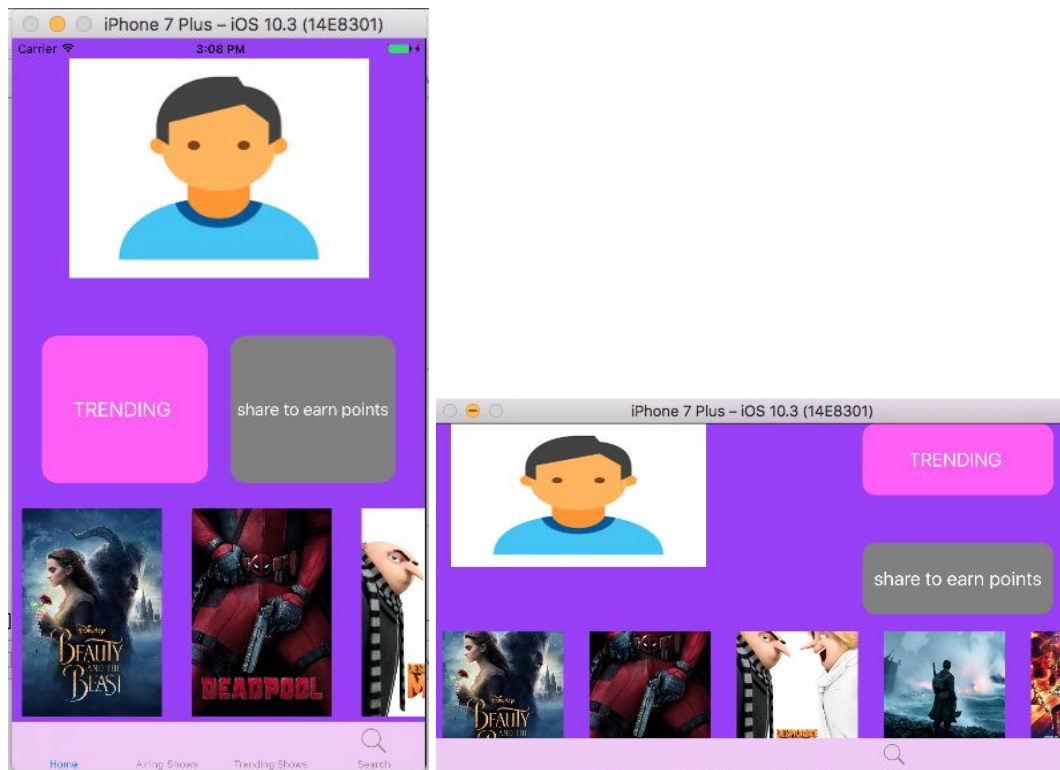


Implementation of multiple view:

- 1) Collection view - To view the trending shows from API, a better layout to view a large collection of set that are trending in social network.
- 2) Split view : To view the currently airing show , a better view for certain amount of data set, enabling the user set to navigate to the different set while viewing the detail page simultaneously.
- 3) Collection view - To View the carousel list of favorite shows in the home page, a better way to have quick glance of user favorite list.
- 4) UIView - To incorporate the image , share and cancel buttons while sharing the movie in the Facebook.
- 5) Table View - Search Movies Functionality.

Adaptive Layout(Size classes):

- 1) Home Screen: Placement of Buttons and Image.



2) Show Detail screen: Length of the buttons and color of the screen background

Cocoa Framework:

AVFoundation was used to implement Cocoa Framework.

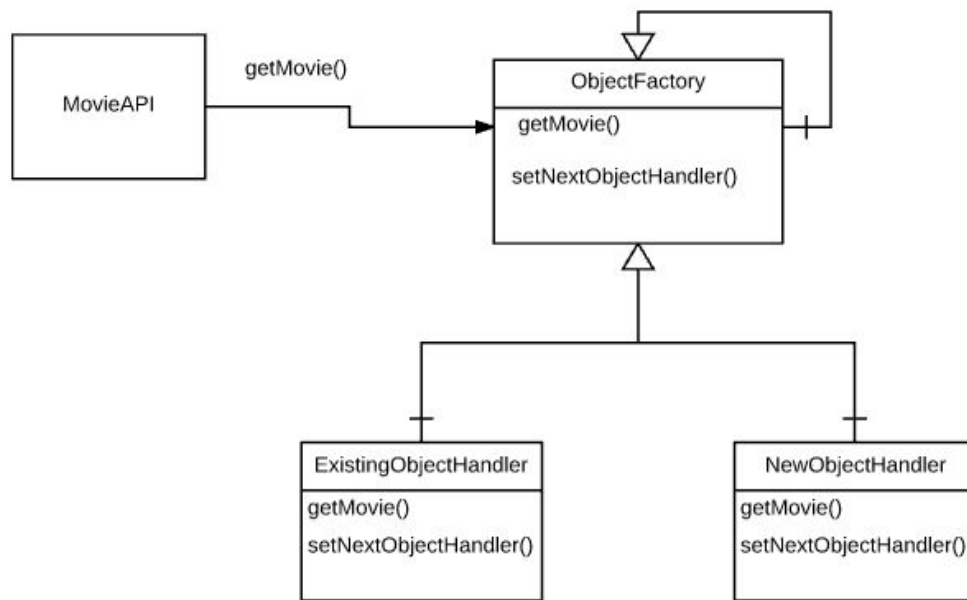
Business Use case:

Advertisement video will be played using AVPlayer, and the user gets points for viewing the video.

Design Pattern:

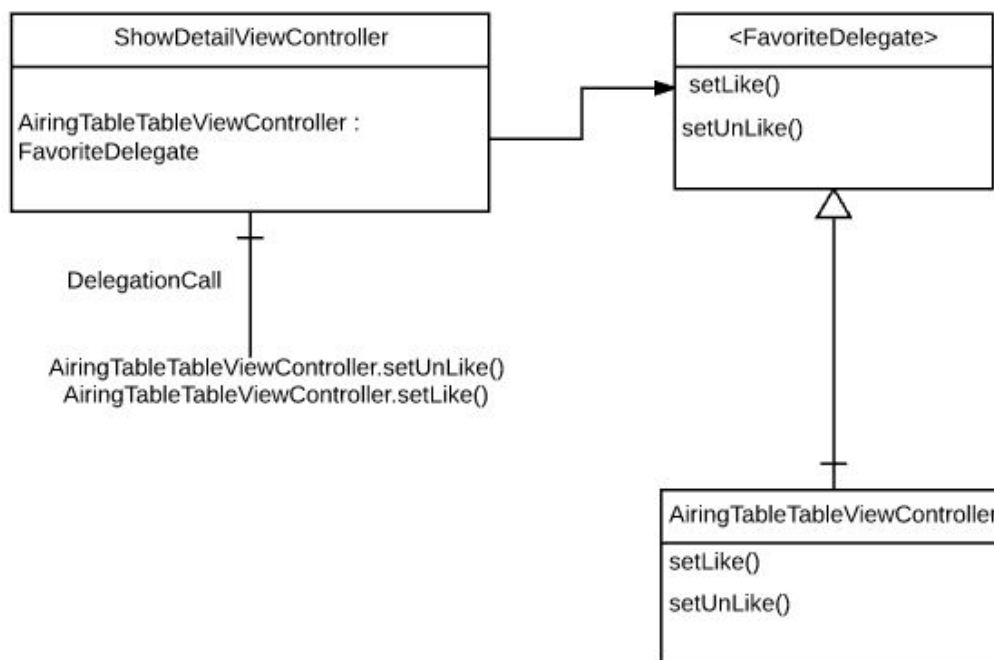
Chain of Responsibility:

System uses chain of responsibility to create movie object. The same movie object is being made to reuse from the object factory if the object is already exists for the same movie. First, ExistingObjectHandler handles the request if the condition satisfies and returns the movie object from the factory, or NewObjectHandler takes the request and creates a new object.



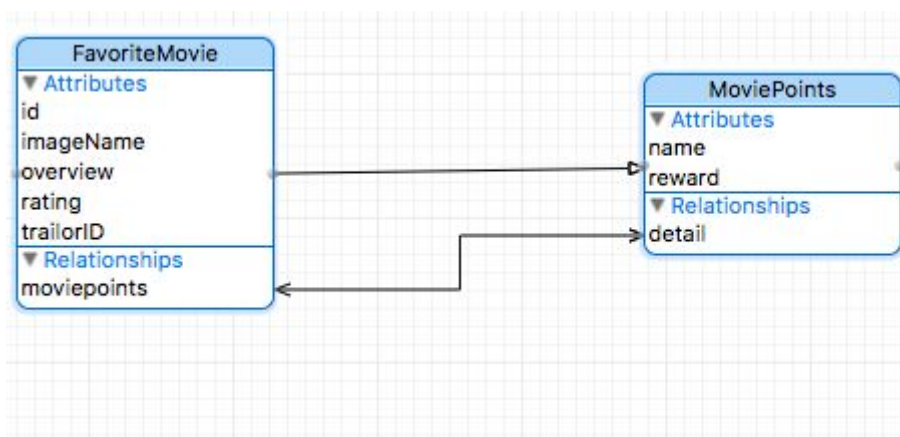
Delegation:

Delegation has been implemented in table view controller. When the movie like status has been changed , it is made to reflect in the table view cell when we navigate back to the table view cell.



Persistent Data(Core Data)

One to One relation



MoviePoints - Parent
FavoriteMove - Chid.

All the Favorite Movie are stored in Coredata. Points earned through these Favorite movie are also stored in the Database.

The above model were implemented using cascade style. Neither cannot exists alone.In other words, the points are only stored for the favorite movie.

CRUD operation are also implemented (FavoriteManager.swift inside the Model)

Unit Test:

Test components: Movie.

Dependency Injection: Movie object has been testing using MockMovieAPI. Dependency injection has been implemented using the technique constructor/method injection where MockMovieAPI has been passed to Movie object, and Movie object has made to interact with MockMovieAPI.

Please also note that MovieAPI has also been used to initiate the correct working behaviour for other cases.

Both MovieAPI and MockMovieAPI conformed to NetworkProtocol

Test Case:

- a) testReuseExistingMovie - Test chain of Responsibility, and check if the objects have been reused
- b) testCreateNewMovie - Test chain of Responsibility, and check if the objects have been created for new movie
- c) testToLoadImageAndTrailer - Test if the request for image and youtube URL has been initiated correctly at correct place
- d) testMovieObject - Test if the movie object has been created with correct data , and attributes can be accessed after creating movie object

Functionality:

Mock :

Test Components: UserDetails

TestCase:

- a) testIncrementUserPoint - Test to check if the user points gets incremented.
- b) testNegativeUserPoint - Test to check if the user points should not be negative.