

Assignment 3


Due Dec 16 by 11:59pm **Points** 10 **Submitting** a file upload **File Types** zip
Available until Dec 18 at 11:59pm

This assignment was locked Dec 18 at 11:59pm.

Assignment to be done individually. Write an iPhone App that targets **iOS 13.4** that uses the **OMDb API** (<http://www.omdbapi.com/>) web service to get information from the Internet Movie Database. The user will be able to add movie records that have title, release year, IMDb id, plot, and a link to a movie poster. The app will use a navigation controller, a collection view controller, and a view controller. The app must be properly commented. Each method you create must have a brief description preceding it. When complete, compress your project folder into a single zip file and submit it.

IMDb

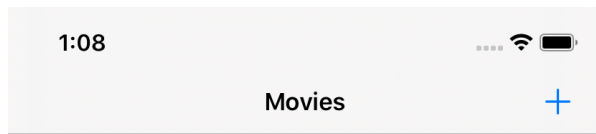
- Need a Mac? See [Student Resources](#).
- Create an iOS Application using the Single View Application template.
 - Product Name: IMDb
 - Organization Name: Mohawk College
 - Organization Identifier: ca.mohawkcollege
 - Language: Swift
 - User Interface: Storyboard
 - Use Core Data: Unchecked
 - Include Unit Tests: Unchecked
 - Include UI Tests: Unchecked
 - Source Control: Create Git repository on my Mac: Unchecked

 Add the following statement of authorship as a comment to the top of MoviesViewController.swift replacing Student Name and 123456789 with your name and student number. **Failure to include a statement of authorship will result in a grade of 0.** These two lines must be the first two lines of the file.

- I, Student Name, student number 123456789, certify that this material is my original work. No other person's work has been used without due acknowledgement and I have not made my work available to anyone else.
- Images for app: [03_imdb_images.zip](#). Configure the AppIcon and add the other image to the asset catalog, it will be used as a default when no movie poster is available..

Screenshots

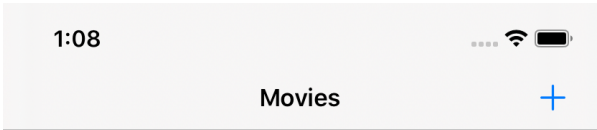
- When the app first launches it will look like this.




•



After the + button is tapped, it will look like this.



•

 After the default image is tapped, and a movie has been searched for, it will look like this.

1:09

.....

< Movies

New Movie

Title

Search

Year

2019

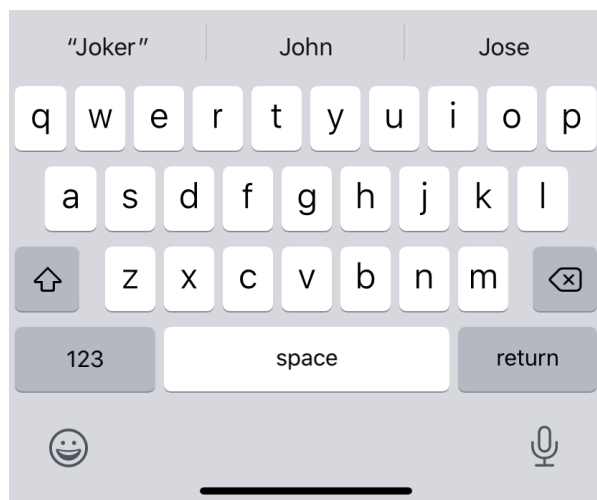
ID

tt7286456

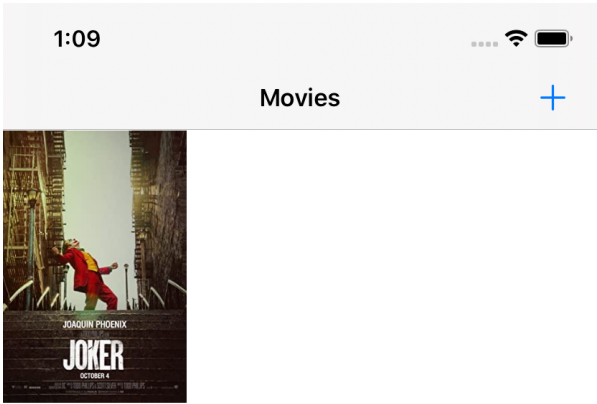
Plot

In Gotham City, mentally troubled comedian Arthur Fleck is disregarded and mistreated by society. He then embarks on a downward spiral of revolution and bloody crime. This path brings him face-to-face with his alter-ego: the Joker.


•




After the back button is tapped, it will look like this.





•

 If a movie that has already been added is tapped, it will look like this.

1:09





 Movies

Metropolis

Title

Search

Year

1927

ID

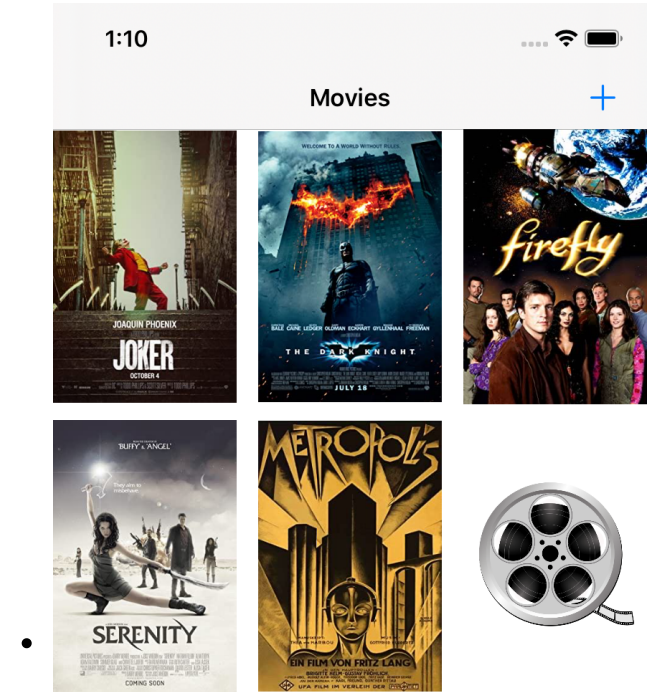
tt0017136

Plot

Sometime in the future, the city of Metropolis is home to a Utopian society where its wealthy residents live a carefree life. One of those is Freder Fredersen. One day, he spots a beautiful woman with a group of children, she and the children quickly disappear. Trying to follow her, he is horrified to find an underground world of workers who apparently run the machinery that keeps the Utopian world above ground functioning. One of the few people above ground who knows about the world below is Freder's father, John Fredersen, who is the founder and master of Metropolis. Freder learns that the woman is called Maria, who espouses the need to join the "hands" - the workers - to the "head" - those in power above - by a mediator who will act as the "heart". Freder wants to help the plight of the workers in their struggle for a better life. But when John learns of what Maria is advocating and that Freder has joined their cause, with the assistance of an old colleague, an inventor called Rotwang, who turns out to be But their nemesis goes to work towards quashing a proposed uprising, with Maria at the centre of their plan. John, unaware that Rotwang has his own agenda., makes plans that include shutting down the machines, with the prospect of unleashing total anarchy both above and below ground.



After several movies have been added, it will look like this.



After an unsuccessful search, it would look like this.



1:44

< Movies New Movie

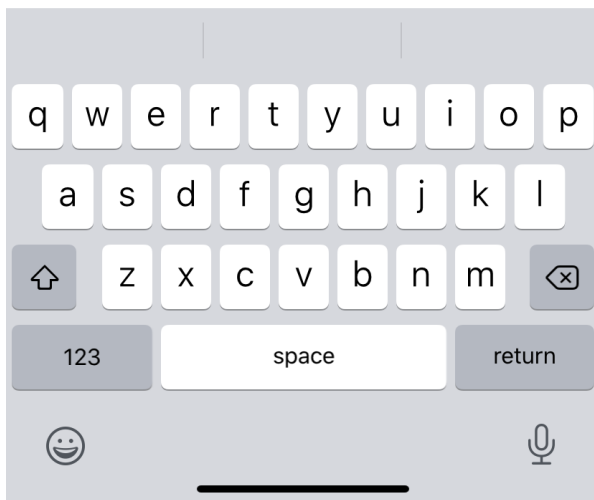
Title Search

Year Not found

ID

Plot

•



- Add a class named Movie.
- The Movie class implementation must adhere to the class diagram.
- It must also conform to the Codable protocol.

Movie
Title : String
Year : String
Plot : String
Poster : String
imdbID : String
init()

- In the initializer, set Title to <Placeholder>.

- Poster is the URL to the movie's poster.
- imdbID is the Internet Movie Database's Id for the movie.

MovieCell

- Add a class named MovieCell that subclasses UICollectionViewCell.
- The class will have one image view outlet property only.

MoviesViewController

- Delete the existing view controller.
- Add a new collection view controller named MoviesViewController to the project.

Storyboard

- Delete the existing view controller.
- Add a new collection view controller to the storyboard.
- Embed the new collection view controller in a navigation controller.
- Set the navigation title to Movies.
- Add an add bar button item to the navigation view.
- Set the cell size width to 128, the height to 192, and estimate size to none.
- Change the collection view cell to MovieCell, add an image view, constrain it to the cell size, and connect it to the movie cell image view outlet.

Code

- Declare an array to hold movies.
- Implement collectionView(_:numberOfItemsInSection:).
- Implement collectionView(_:cellForItemAt:).
 - Retrieve a reusable MovieCell.
 - For now, just display the movie_reel image in the movie cell.

- Implement a method named newMovie that will be connected to the add button on the collection view controller.
 - Create a new movie.
 - Add it to the array.
 - Call insertItems on the collection view to add the new cell.

DetailViewController

- Add a view controller named DetailViewController.
- For the sake of simplicity, you are not required to use AutoLayout. The app will be graded using the iPhone 11 scheme. Make sure the UI looks good on that.
- In the storyboard, setup a segue from the cell to this controller.
- Setup the outlets and connect the UI elements.
 - The title is a text field.

- The year, and ID are displayed in labels.
- The plot is displayed in a text view. Disable editing of the text view.
- Connect the search button to an action method.
- Setup a property for the current movie, which will be populated in the collection view controller.
- Setup an additional property for the poster, since the poster URL will not be displayed to the user.
- When the controller is about to appear.
 - If the movie's title is <Placeholder>, set the navigation title to New Movie.
 - Otherwise, set the navigation title to the movie's title and the other UI elements to their associated movie properties. Also, setup the poster property with the movie's poster property.
- When the search button is pressed.
 - Verify there is user input.
 - The user may enter spaces in movie titles to search for. However, the OMDb API web service is expecting the + (plus character) where spaces would be. Use the String method `replacingOccurrences(of:with:)` to make this change.
 - Build the following URL: `https://www.omdbapi.com/?t=<user_input>&plot=full&apikey=mohawk`, where <user_input> is the scrubbed value entered in the text field.
 - Using a `URLSession`, build a data task that will return JSON.
 - Parse the JSON into an instance of `Movie` and set the UI elements and poster property.
 - If the movie isn't found, the parsing will fail. If this happens, set the year to Not found and clear the other UI elements.
- When the user hits the back button.
 - Check that year isn't blank and also Not found.
 - If a movie has been found, update the current movie object, including the poster URL, which wasn't displayed to the user.

MoviesViewController

- Now that the `DetailViewController` exists, configure the segue to pass the current movie to it.
- ▶ When returning from the detail view controller, reload section 0 of the collection view.
- Update `collectionView(_:cellForItemAt:)`.
 - Not all movies have posters, when they don't the API sets the value to N/A. If the title is <Placeholder> or the poster is N/A, set the image to the default movie_reel.
 - Add the [**ImageStore.swift**](#) from the Photorama app to the project.
 - If there is a poster URL, check the `ImageStore` to see if the image is available, if so, use it.
 - Otherwise, set up a `URLSession` data task with the URL, retrieve the data, and convert it to an image. And store it in the `ImageStore`.