



DEVELOPING OUTSYSTEMS MOBILE APPS

# Movies App Optional Assignment



# Table of Contents

Optional Exercises .....	3
1. Search Movies (Input, search, & refresh data).....	3
2. Movie Images Carousel (Silk UI Carousel & images).....	3
3. Cast and Crew (Blocks & Search).....	4
4. Movie Comments (Silk UI ChatMessage pattern) .....	5
5. Settings (Plugins) .....	6
6. Local Storage and Offline.....	7
List of images.....	8

## About these exercises

Exercises 1-5 are independent and can be completed out of order. The description in parenthesis provides an idea of the type of components that the exercise focuses on.

So, if some of these exercises seem more interesting, you could work on those first and save the others for later.

## Copyright

This material is owned by OutSystems and may only be used in the ways described in this Copyright Notice:

- You may take the temporary copies necessary to read this document
- You may print a single copy of this material for personal use
- You must not change any of this material or remove any part of any copyright notice
- You must not distribute this material in any shape or form

# Optional Exercises

In the following exercises, you will extend the Movies App even further with interesting new Screens and features. You may need to reference additional Entities from the **MoviesData** module.

## 1. Search Movies (Input, search, & refresh data)

Add an input field (**Search** type) to the **HeaderContent** placeholder of the **Movies** Screen. Use the text entered by the user to automatically filter the list of Movies. To refresh the Movies list, use the **OnChange** Event of the **Input** Widget, and inside the associated Client Action use the **Refresh Data** statement. The Movies Screen with search enabled may look like this.

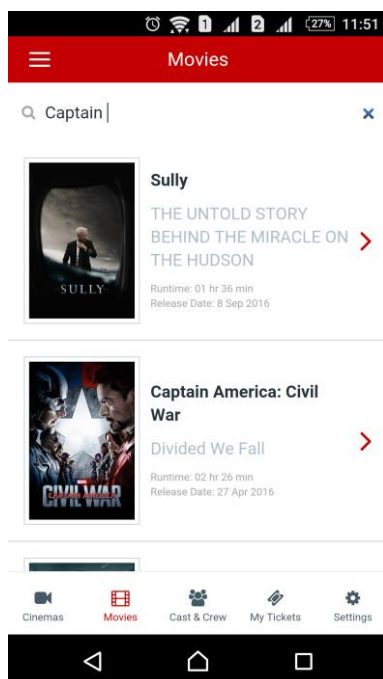


Figure 1. Movies Screen with search field

## 2. Movie Images Carousel (Silk UI Carousel & images)

Add the **Carousel** Widget to the **MovieDetail** Screen, under the **Info** Tab. Fetch the **MovielImages** associated with the current movie using an Aggregate.

Inside the **Carousel** Widget place a **List Item**, and inside it an **Image** with Type set to Binary Data. Bind the **Image Content** property to the BinaryContent attribute of the **MovielImage** Entity, from the Aggregate's result. Limit the maximum number of results of the Aggregate to 5 or 10. The info tab with Movie Carousel may look like the following examples.

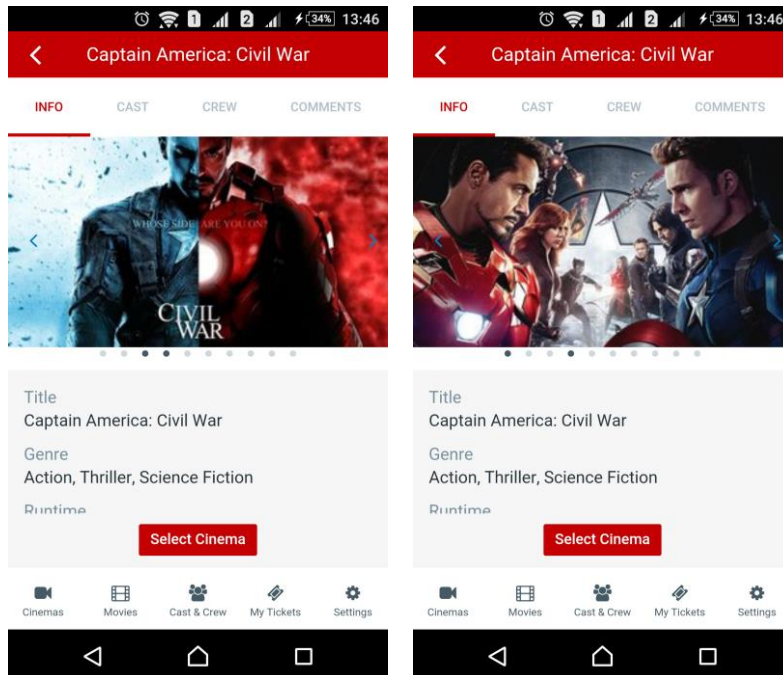


Figure 2. Carousel Samples

### 3. Cast and Crew (Blocks & Search)

Create a **CastAndCrew** Screen with a title of 'Cast & Crew'. Display all the people that are associated as a cast or crew member of any movie in the app. Since that may be quite a few people, improve usability by adding a search field to the top of the screen that reacts automatically (like in Section 1).

You may want to create a **PersonList** Block to display the list of cast and crew members. The list can then be reused in other Screens in the application

The **CastAndCrew** Screen with search enabled may look like the following example.

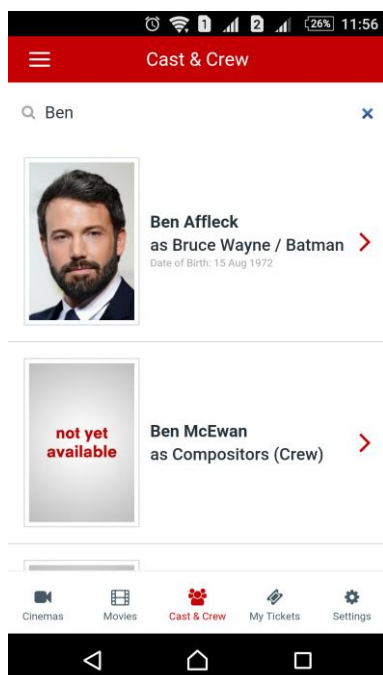


Figure 3. Cast and Crew Screen

Create a **Person** Screen to display more detailed information about each cast or crew member. This Screen will be accessible when the user selects a person from the **CastAndCrew** Screen's list.

In the **MovieDetail** Screen, add lists of cast and crew members to the respective tab. In these tabs, selecting a person takes the user to the **Person** screen. The Cast and Crew tabs may look like this.

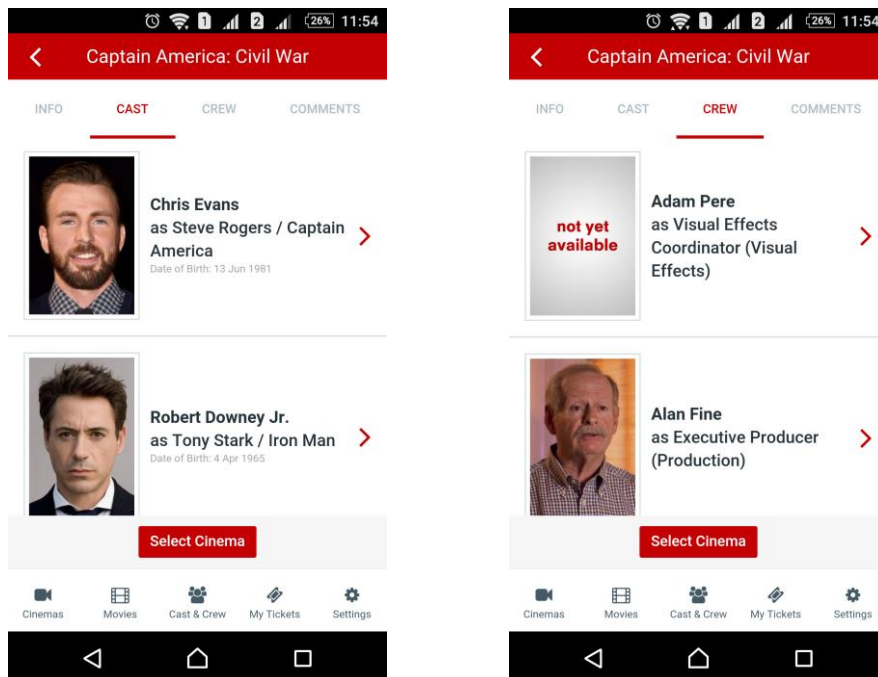


Figure 4. Cast and Crew tabs in MovieDetail Screen

## 4. Movie Comments (Silk UI ChatMessage pattern)

In the **MovieDetail** Screen, implement the **Comments** tab. This tab should display comments submitted by users and also allow logged in users to submit new comments. When the user submits a new message, remember to refresh the Aggregate.

Use the **ChatMessage** Widget to display the comments. The **DisplayOnRight** property allows you to display the user avatar either on the right or on the left side. If the message was written by the current logged in user, display the user's profile picture on the right.

In the sample screenshot below, the **Comments** tab has two comments. The last one is from the logged in user. When a user is not logged in, the input field and button to submit comments should not be displayed. Instead, anonymous users should see a message that login is required to submit comments.

The Comments tabs may look like the following example.

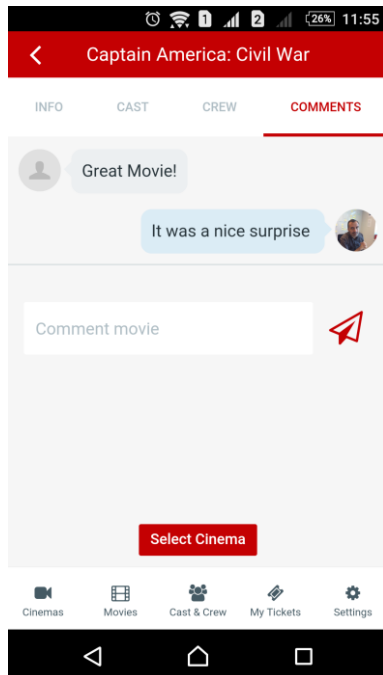


Figure 5. MovieDetail Comments tab

## 5. Settings (Plugins)

Add a **User Avatar** Widget to display the user name and profile picture. The Profile Picture should be stored in the **UserProfile** extension Entity. Using the **Camera** Plugin available in the **Forge**, allow users to change their profile picture by taking a new picture with the device camera.

Use the **UserCreditCard** extension Entity to store the user credit card information. The **Card IO** Plugin available in the **Forge** allows you to scan your credit card information with the device camera.

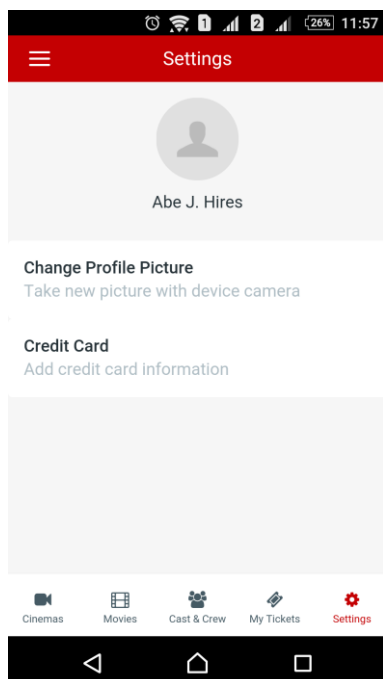


Figure 6. Settings Screen

## 6. Local Storage and Offline

Extend the Movies app to save all tickets, and the ticket information, in the device's Local Storage, as well as in the server, for when the user is online.

Since the Data Model of the app is a bit large and complex, it probably makes sense to create a **LocalTicket** Entity that holds all the information needed to display the ticket, e.g: session, cinema, seat, even though that information is stored in different Database Entities.

If the user is not online, the app should synchronize the information in Local Storage, to the server, when the app gets back online. Remember to set the needed sync configurations in the **OfflineDataSyncConfiguration**.

Also, remember to provide feedback to the user when the device is offline.

## List of images

Figure 1. Movies Screen with search field .....	3
Figure 2. Carousel Samples .....	4
Figure 3. Cast and Crew Screen.....	4
Figure 4. Cast and Crew tabs in MovieDetail Screen.....	5
Figure 5. MovieDetail Comments tab.....	6
Figure 6. Settings Screen .....	6