

## How to Use this Template

1. Make a copy [ File → Make a copy... ]
2. Rename this file: “**Capstone\_Stage1**”
3. Replace the text in green

## Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone\_Stage1.pdf**”

---

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** jasensanders

# Folio

## Description

Folio is a easy way to manage all of your DVD and Blu-ray movies. Track what movies you own and in what format. Do you have that in UltraViolet or Disney Movies Anywhere? Folio lets you know what you have and what you don’t without having to dig through all your discs to find out.

Simply enter the UPC code or use the built in barcode scanner and the movie is recorded into your database. Have the movie on DVD, but want it on Bluray? Folio lets you search different stores for the movie you want, in the format you want. Never again will you have to dig through piles of discs to decide which movie to watch, with Folio you will always know.

## Intended User

Do you have a vast collection of DVD and/or Bluray movies? Do you hate digging through them just to find what you want to watch? This app is for you!

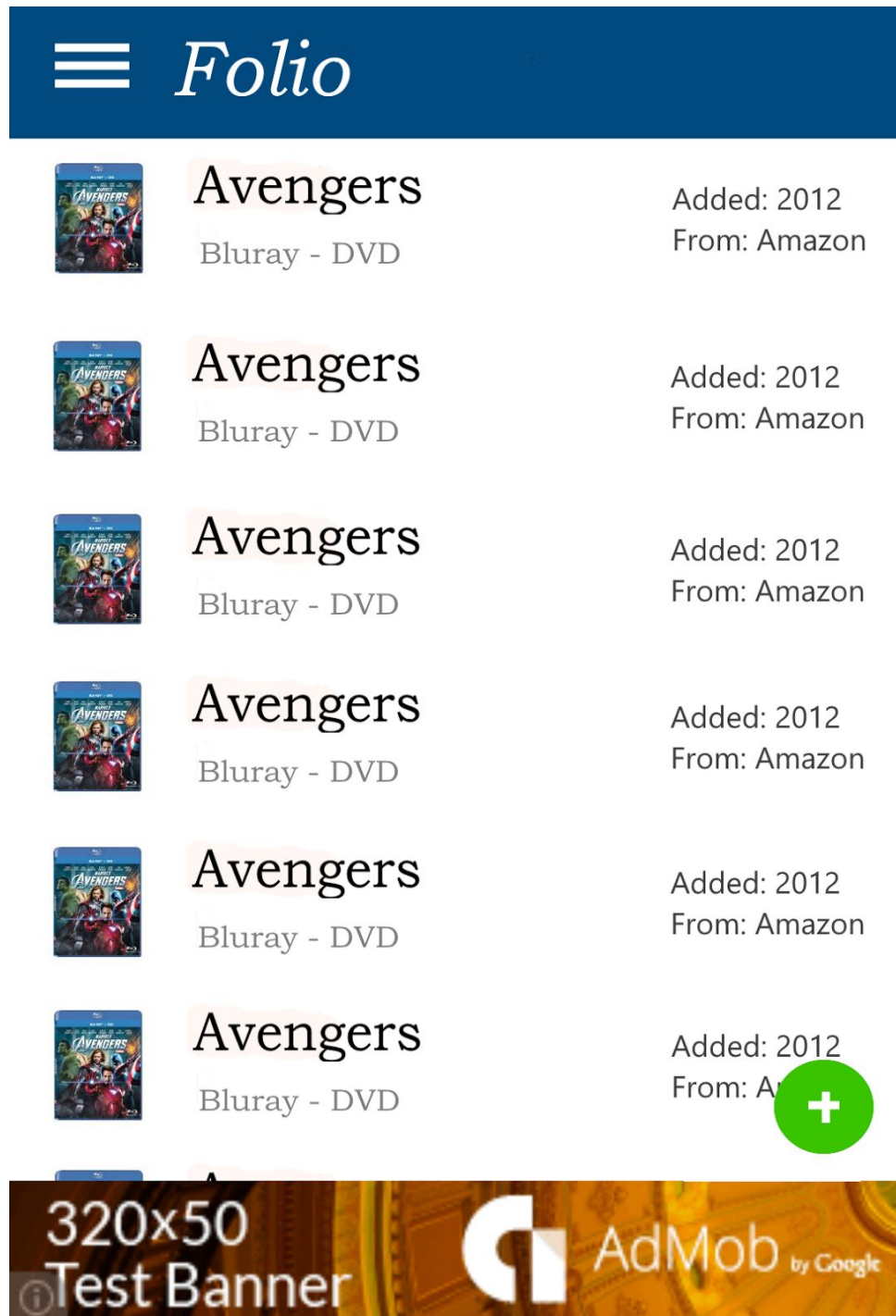
## Features

- View all owned movies in a clean clear interface
- Look up Movie details from online provider using manual entry or barcode scan
- Saves your movie in a database on your phone or tablet
- Track the movie you own and in what format.
- Edit and Update details of each movie bought. What retailer, date purchased, etc.
- Find movies or formats that you do not yet own and Search online retailers for them.

## User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



This is the main screen showing the list of movies you currently own. Also shows the date added and where you bought it from.

## Screen 2



*Formats: Blu-ray, DVD*

*Release date: 2012*

*Purchase Location: Target*

*UPC: 7-86936-85038-3*



*Date Added: 9-15-2016*


### *Overview:*

*All these superhero's now have to defend earth from more than their own egos. Drama will ensue and buildings will fall down, but you will want to watch it anyway.*

This is the details screen for each movie.

## Screen 3

 *Add new* 




*Details:* *Marvel's The Avengers* *2012*


*Formats:* ☐ *Bluray* ☐ *DVD* ☐ *Digial Copy* ☐ *UltraViolet*

☐ *Disney Movies Anywhere*

*Purchase Location:*

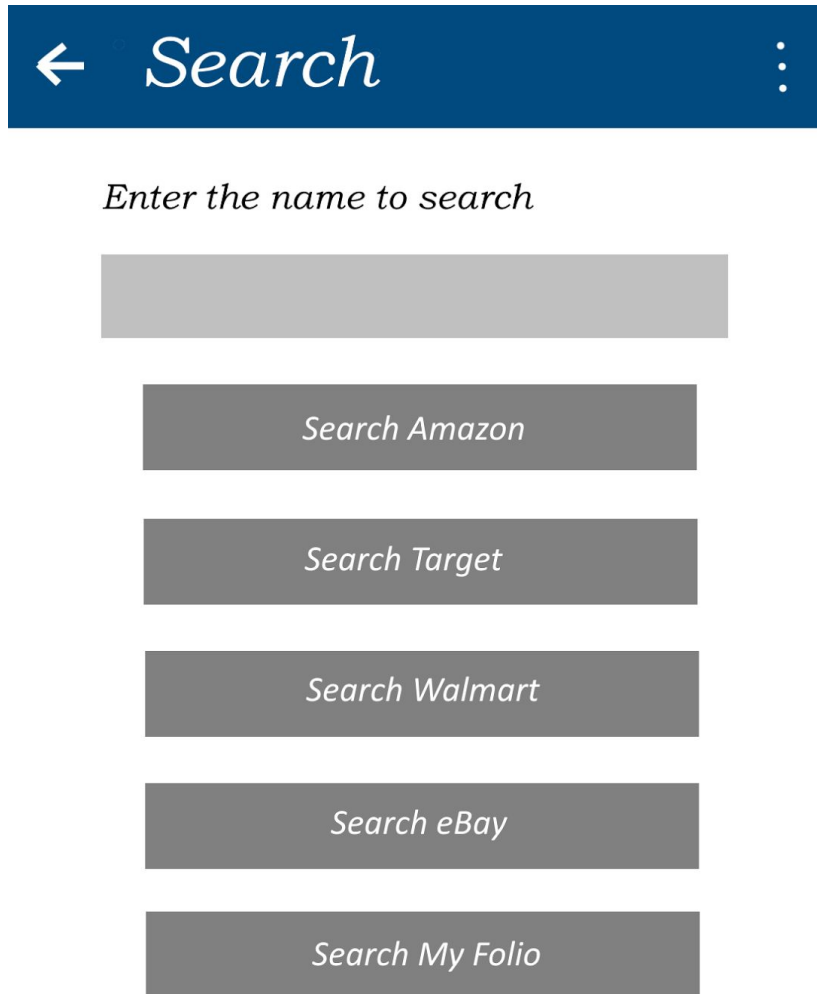
*Purchase Date:*





This is the Add New Movie Screen It allows you to add a movie by Scanning a barcode or typing in the UPC. It also allows you to add other details to the database to help track formats and dates.

## Screen 4

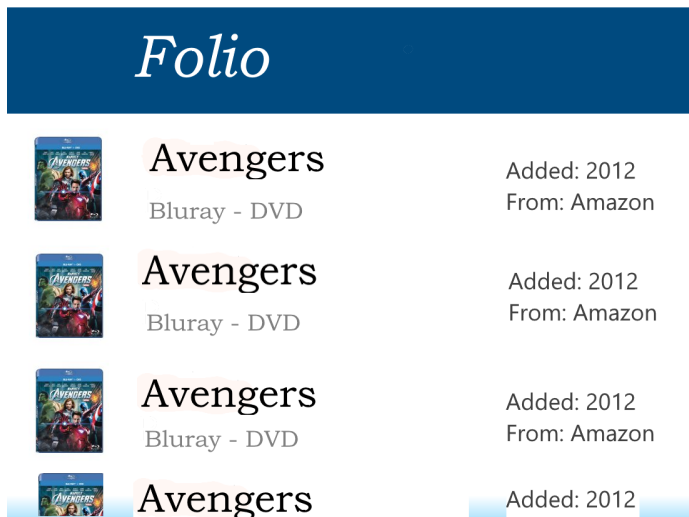


The screenshot shows a mobile application interface for a search screen. At the top is a dark blue header bar with a white left-pointing arrow, the word "Search" in a white serif font, and a white three-dot menu icon on the right. Below the header is a light gray instruction text: "Enter the name to search". Underneath this is a wide, light gray rectangular input field. Below the input field are five dark gray rectangular buttons, each containing white text. The buttons are labeled "Search Amazon", "Search Target", "Search Walmart", "Search eBay", and "Search My Folio" from top to bottom.

This is the Search screen. It allows you to search online from movies you don't have, as well as find a movie you may already have in your Folio



## Screen 6



This is the collections widget preview.

## Key Considerations

### How will your app handle data persistence?

Data Persistence will be through Shared Preferences for settings. A Content Provider will store all the details about each movie in its database. If there is time, the Backup API will be implemented for api 22 and lower devices, with appropriate settings for api 23 and up.

### Describe any corner cases in the UX.

It's possible that the user may go back to the "main screen" from the "add a movie screen" before the intent service is done. However since the intent service is independant of the UI and the database is also independent, the UI on the main screen will simply update when new data is available.

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

Glide library for loading images

Google Mobile Vision library for Barcode scanning

Appcompat Library for Material design that is backwards compatible with older Android versions

Play services admob library for serving ads

Play services Analytics library for Ads and tracking user usage patterns



**Describe how you will implement Google Play Services.**

Play services admob library for serving ads

Play services Analytics library for Ads and tracking user usage patterns.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Project Setup

Load New Nested Scrolling project into Android Studio and Update all SDK components

Build Content Provider:

- Configure DataBaseHelper
- Implement Provider
- Implement Contract class

Build Intent Service to Fetch Data:

- Get the Item data from SearchUPC API and parse
- Get data from TMDB API to fill in details
- Push to content provider

Build elements for Barcode scanner:

### Task 2: Implement UI for Each Activity and Fragment

Choose Robot Font Style and Color theme:

- Design clean Material Interface with bold color and no glare.
- Design Folio Logo or Font Style

Implement UI's:

- Build UI for MainActivity and Setup banner Activity
- Build UI for AddAnItem fragment
- Build UI for DetailsView/RemoveItem fragment
- Build UI and data support for Barcode Scanner

- Build UI for the Search Screen
- Build UI for Widget

### **Task 3: Set up Analytics and Connect to Admob**

Build Utility Class for sending hits to Analytics:

- Extend Application class and add tracker
- Write code to send hits to analytics
- Implement Analytics setup xml file.
- Connect Analytics to Admob in dashboard.

### **Task 4: Test/Handle Error cases and Loss of Connectivity**

Test Error Cases:

- Invalid User Input Errors
- API Error
- No Network connection Errors
- Tablet Layout/ Screen Rotation Errors
- Make Sure all Errors are logged and handled gracefully.

### **Task 5: Implement Backup API for devices API 22 and lower**

Setup Backup for API 23+:

- XML and Manifest Modifications

Setup Backup for API 22-:

- Implement Backup API

Add as many tasks as you need to complete your app.

---

### **Submission Instructions**

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"