

CAPSTONE_STAGE 1

[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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

GitHub Username: ArcMax

Swimming Techniques

Description

Techniques to improve swimming skills at one place. Learn to swim with our app which provides complete set to master swimming strokes. App provides with water aerobics which helps to get fit helps to tone body & to get bikini body.

Intended User

Anyone who loves swimming, master the techniques & who would like to be fit.

Features

- List of different techniques.
- Sharing techniques with friends.
- Sharing nearby pools information with friends.

User Interface Mocks

Screen 1



Description:

First screen list up all techniques for freestyle stroke videos.
On selecting sort button on top right shows list of different techniques.
On selecting other techniques sorts & displays videos.

Screen 2



Description:

Detail screen of selected video.

Description of techniques described in video.

Share button to share this particular video.

Key Considerations

How will your app handle data persistence?

Will build a Content Provider for data persistence.

Describe any corner cases in the UX.

On back pressed, goes back to swimming technique screen.

On Share pressed, menu to share, ex: whatsapp

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

Picasso to handle the loading and caching of images.

Next Steps: Required Tasks

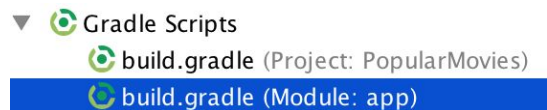
This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Image Library - Picasso

How to Setup Picasso

You'll need to modify the build.gradle file for your app. These modifications will happen in the build.gradle file for your module's directory, *not* the project root directory (it is the file highlighted in blue in the screenshot below).



In your app/build.gradle file, add:

```
repositories {  
    mavenCentral()  
}
```

Next, add **compile 'com.squareup.picasso:picasso:2.5.2'** to your dependencies block.

Using Picasso To Fetch Images and Load Them Into Views

You can use Picasso to easily load album art thumbnails into your views using:

[Picasso.with\(context\).load\("http://i.imgur.com/DvpvklR.png"\).into\(imageView\);](#)

Picasso will handle loading the images on a background thread, image decompression and caching the images.

API Hints

Fetch swim techniques with Youtube API

Task 2: Implement UI for Each Activity and Fragment

User Experience

In this stage you'll add additional functionality to the app

Add more information to details view:

You'll allow users to view and play videos.