How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: scneba

App Name: CamEx

Description

The application is intended to showcase Cameroon exclusive talent and culture. The application will include music, movies, fashion and touristic sites. It will use the Youtube api to stream Cameroonian content posted on youtube in order to bring content closer to people interested in Cameroon.

Intended User

The application is open to anybody interested in Cameroon entertainment and culture, especially those who are willing to visit the country..

Features

List the main features of your app. For example:

- Display pictures of music videos, movie trailers, fashion shops and touristic sites.
- Show youtube content of music, movie trailers, fashion shops and touristic sites.
- Add and remove from favourites
- Login with facebook, google, or direct login in order to like and rate app.

User Interface Mocks

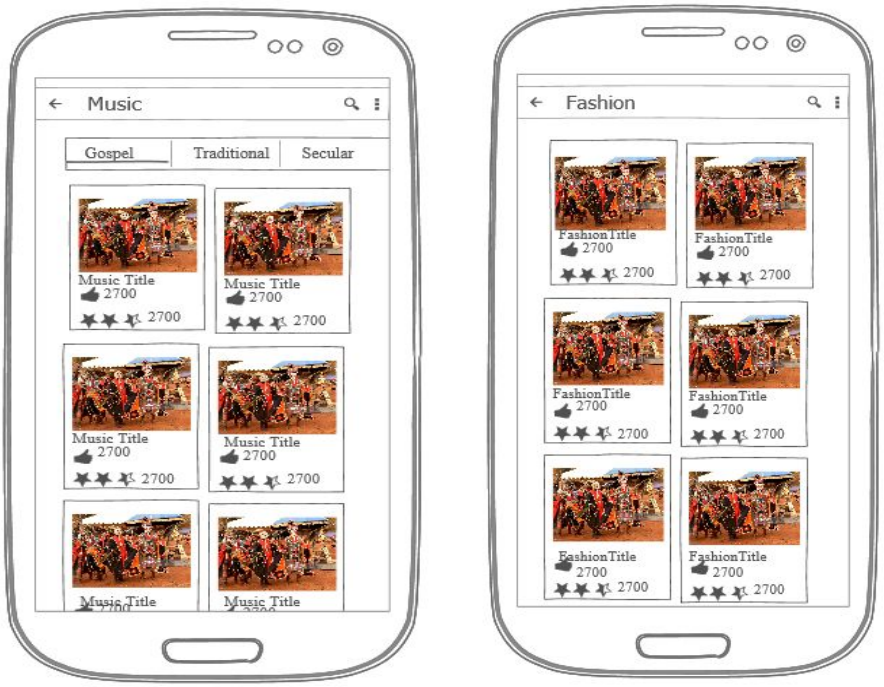
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

A demo of the app can be found here

Screen 1

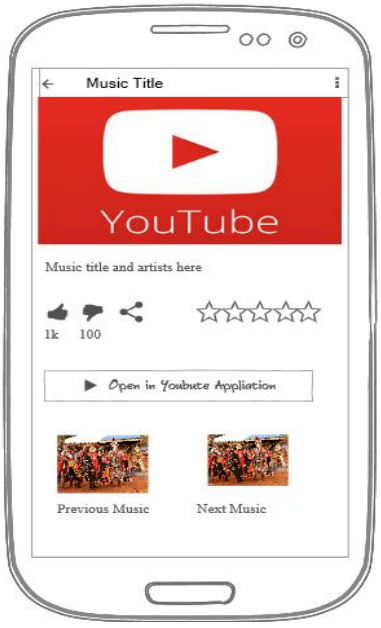
This is the home view of the application containing buttons which will redirect to the details of the four categories of the application.

Screen 2



The two screens display the summary of all four categories of the application(Music, Movies, Fashion, and Tourism). Each card of the recycler will contain a picture, title of view, number of likes, and the average rating for that item.

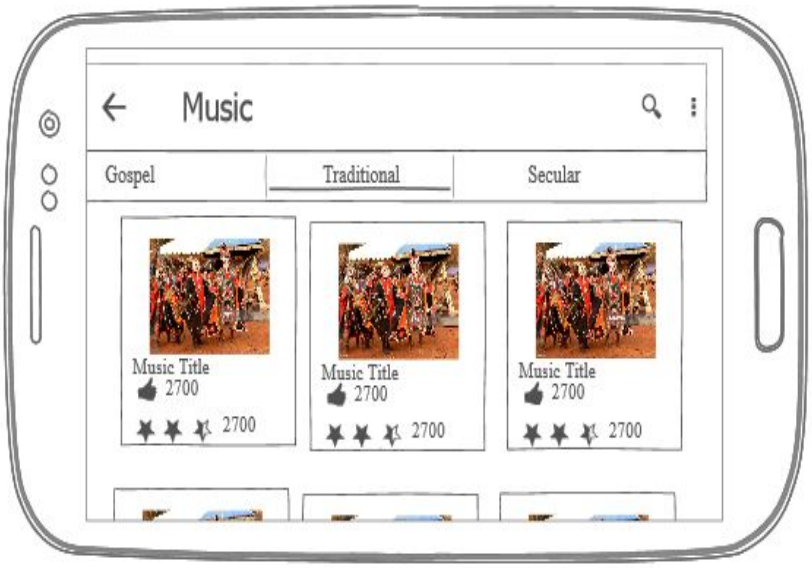
Screen 3



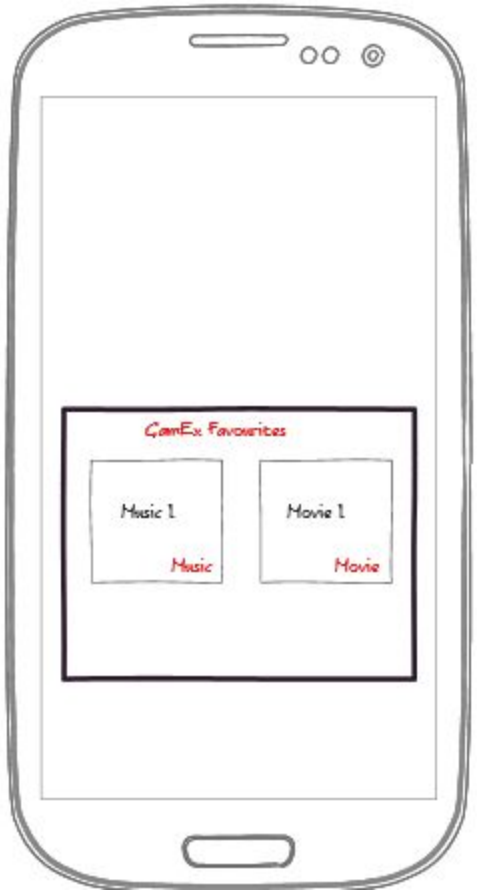
These are the detail screens of the two categories that will be implemented for the Capstone. The youtube video will be shown as above, the user will be able to like, share and rate item. This will require the user to sign into the application.

Screen 4

This screen will allow the user to sign into the application either with facebook, gmail or direct sign up.



These are two tablet views of the application showing the Music page and the music details page.



This is the Widget screen which will show favourites of the user in all categories.

Key Considerations

How will your app handle data persistence?

I will use both Sqlite database and shared preferences to persist data. I will build a Content Provider to mediate between the application and storage.

- The Content Provider will be used by the application to access the Sqlite and display and display favorite content on the UI.
- A Cursorloader will be used to access sqlite data from Sqlite since it is much simpler to use
- Sqlite will be used to store favourite items and shared preferences will be used to store user information.

Describe any edge or corner cases in the UX.

Since the application has organised content, the standard back buttons will work well. I will also add navigation among main categories (music, movies, fashion, and tourism) in each detail activity by adding links in the toolbar dropdown.

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

I will use the following libraries:

- Picasso to display images as it is quite efficient.
- Butter Knife for data binding and onclick listeners as it is a very easy-to-use library.
- Firebase for notifications which work in real time with data addition on the backend. Firebase will be very helpful to provide real time functionality.
- Facebook sdk to allow user signup using facebook.
- Youtube api to play youtube videos in application.

Describe how you will implement Google Play Services or other external services.

I will use google play services to enable user sign up with google account. I will also use Firebase services to enable real time notifications from my backend.

Next Steps: Required Tasks

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

Task 1: Building the Backend

I will start the project by setting up the backend. I will use php with Laravel framework to build a content addition and modification system for the application.

Task 2: Importing relevant libraries.

I will create a new simple project and use gradle to import all the relevant libraries including Picasso and Butter knife libraries.

Task 3: Build Main View.

I will implement the main view here and link up this view to their specific views using intents.

Task 4:Handle “Music” Category

I will then implement the Music view to load music data from backend. The music view has three subcategories, and this will be implemented using android viewpager. I will also implement the detail view to play music using youtube api. The user will be able to like a music, share the music using android shared intent, and also rate the music.
The AsyncTaskLoader will be used to load data from the remote database since it is a short term request.

Task 5: Handle “Movies” Category

At this level, I will implement the details view for Movies category. This view will contain a card of pictures which will link to a detail view on click. The detail view will play the movie trailer, allow user to like, share and rate the movie.

Task 6: Handle User Sign Up

At this level, I will handle user sign up with facebook, google and direct signup with username and email address.

Task 7: Build Notification service with Firebase

Lastly, I will build a real time notification service with Firebase such that, any new music or movie added to database should immediately notify all users.

Task 8: Build the Details Screen

I will use Intent Service here to update likes in the background so as to carry the weight away from the main screen.

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

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]

    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”