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**”

GitHub Username: dsvoronin

Grind.FM radiostation

Description

Grind.FM is the first radiostation for gamers in Russia. Listen to the radio stream and podcasts with game developers and famous gamers. Follow and be notified with latest station news. Watch videos, records from twitch livestream.

Intended User

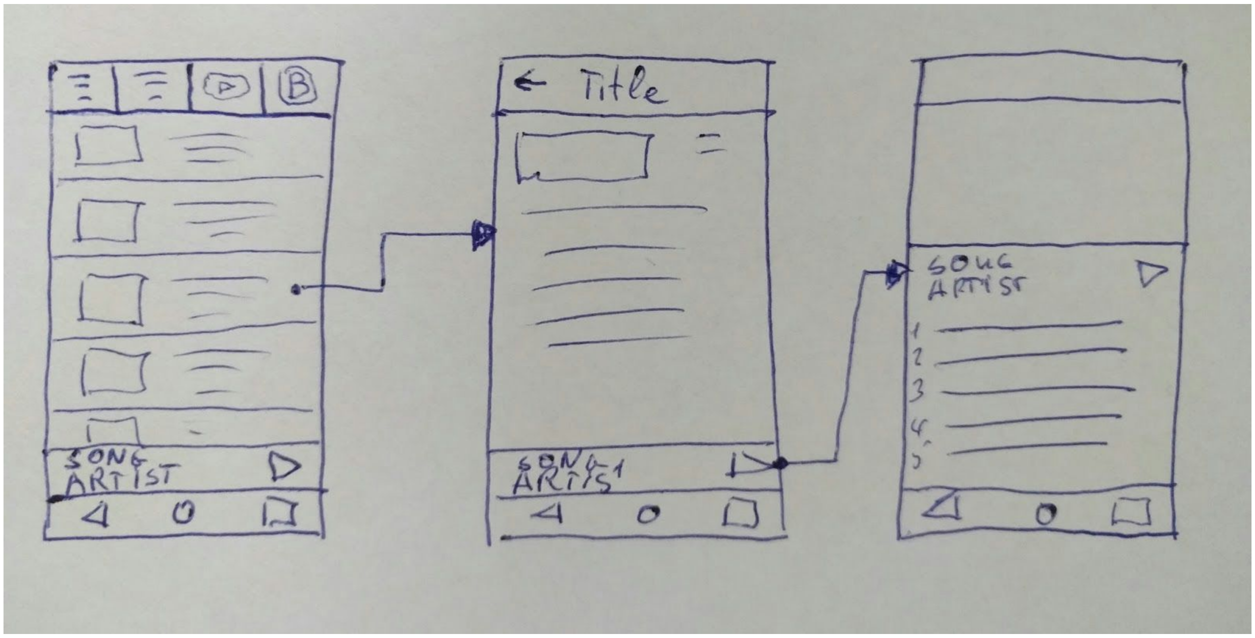
Grind.FM radio listeners and Goha.ru forums community(these projects are very close to each other).

Features

- Radio stream
- Podcasts (+ optional notifications)
- Grind.FM news (+ optional notifications)
- Goha.ru news
- Videos list with youtube followup

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.



Main screen of the app consists of:

1. Tabbar - root navigation between: Grind.fm News, Goha.ru News(probably mixed with grind), Podcasts, VideoRecords, VK social webview.
2. Content - list of news/podcasts/videos/webview -> details view (master detail for tablets)
3. BottomSheet with player, by swiping out user can reveal last played songs list

Key Considerations

How will your app handle data persistence?

App will persist data for all main lists, share it via content provider.
Sync adapter will be used to periodically sync content(once a day, the source never updated more frequently). User can't add or edit any data, so it will be read(sync)-only behavior.

Describe any corner cases in the UX.

To be able to control music player from every place in the app i choose bottom sheet like in Google Play Music app.
Master - detail pattern will be used for any data screens in table layout.

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

Picasso will be used for image loading. It's easy to configure and takes care of image caching and recycling in adapter.
OkHttp as a modern httpclient, basically for not bother myself about compatibility issues.
Retrofit to generalize web interfaces. And wrap networking.
Support design library is a must for creating material UI elements easily as my app will conform material design guidelines.
Crashlytics to be aware of user crashes and some basic analytics for the app.

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

First of all i have an old (2012) project of this app, the first app i ever created for android.
The goal for this project is to reimagine all the stuff i've done back in the days. So:

- Create a branch on github and upgrade all the libraries, tools etc. See if it breaks something and fix it.
- Add all the libraries for the new project.

Task 2: Implement UI for Each Activity and Fragment

I can develop ui and backend layers separate from each other, so first step will be redesign an app to conform material guidelines and mocks i created.
If something missed i can provide temporarily mocks for data.

Task 3: Create new backend layer

Build SyncAdapter, Authenticator(probably stub one), ContentProviders for new data sources.
Don't forget about validation!
Errors will be send to crashlytics.
For some time they will live side to side with old ones.
I can fully test it from now on.

Task 4: Link UI with new backend layer

With help of loaders and content resolvers - provide data from new backend layer to UI.
On this moment if i can remove old code.

Task 5: Rebuild music player service from scratch.

Old code is ugly. There are some new helpers like MediaSessionCompat .
Not sure about accessability here. How can i provide non-audio version?

Task 6: Notifications

Notifications will be especially usable for this app:

- Music player service notification with controls and current song.
- Optional news and podcasts notifications, like: "hey new event on grind.fm!", these must be groupable.

Task 7: Widget

Last one: widget with radio stream controls.

Task 8: Google Play Services considirations

Not sure if i need some of these.
Closest candidates are:

- **google cast** to be able to cast radio stream to tv or speakers
- App invites for better "share an app" flow

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**”