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.
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: ijcheung

Karaokyo

Description

Karaokyo is a lyric overlay app. Now you can surf the net or play your Angry Birds and have your lyrics right there at the tips of your fingers.

Intended User

Music and karaoke enthusiasts, self described multitaskers.

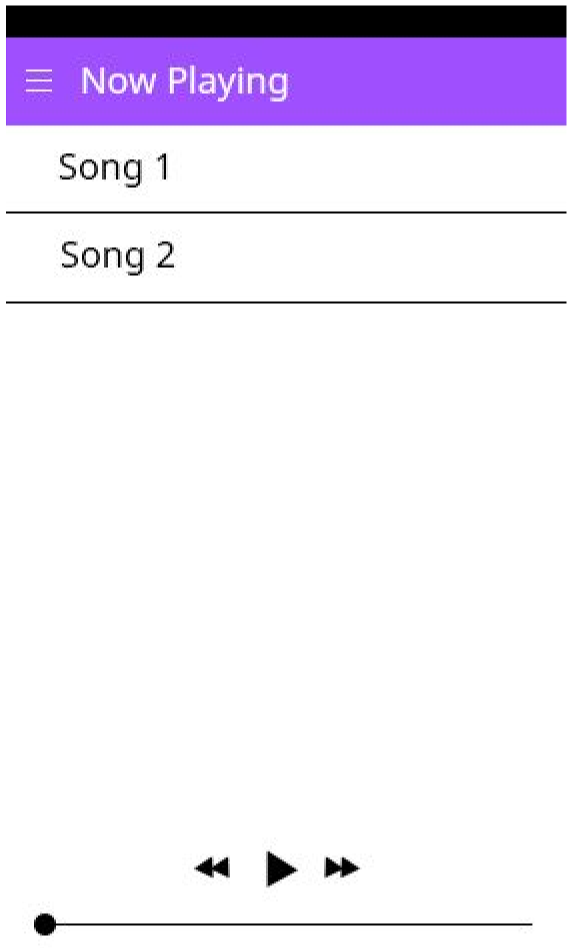
Features

- Download lyrics
- Display lyrics
- Play audio files

User Interface Mocks

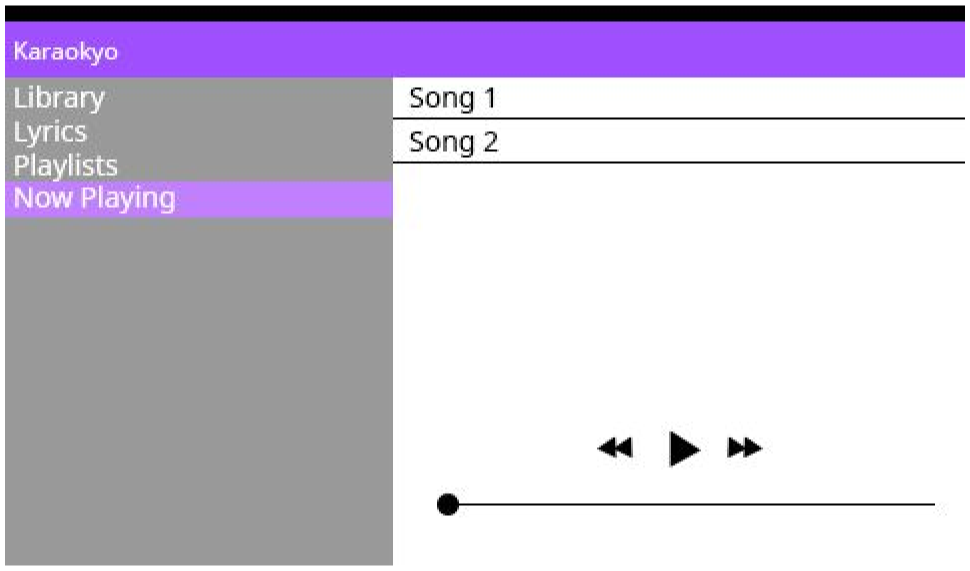
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



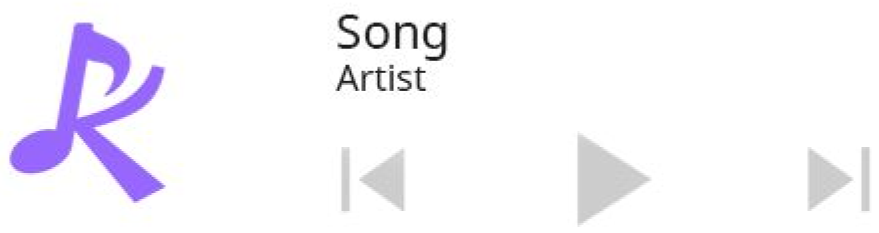
Now Playing Phone

Screen 2



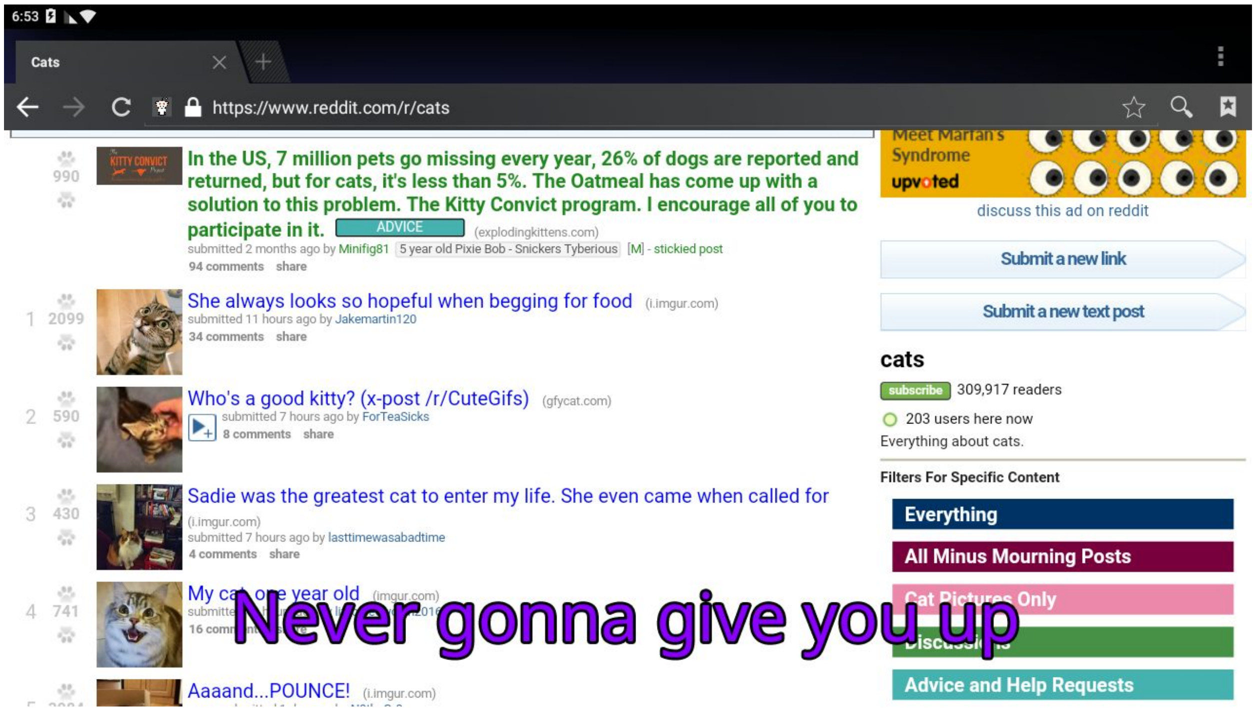
Now Playing Tablet

Screen 3



Widget

Screen 4



990 In the US, 7 million pets go missing every year, 26% of dogs are reported and returned, but for cats, it's less than 5%. The Oatmeal has come up with a solution to this problem. The Kitty Convict program. I encourage all of you to participate in it. ADVICE (explodingkittens.com)
submitted 2 months ago by Minifig81 5 year old Pixie Bob - Snickers Tyberious [M] - stickied post
94 comments share

1 2099 She always looks so hopeful when begging for food (i.imgur.com)
submitted 11 hours ago by Jakemartin120
34 comments share

2 590 Who's a good kitty? (x-post /r/CuteGifs) (gfycat.com)
submitted 7 hours ago by ForTeaSicks
8 comments share

3 430 Sadie was the greatest cat to enter my life. She even came when called for (i.imgur.com)
submitted 7 hours ago by lasttimewasabadtime
4 comments share

4 741 My cat one year old (imgur.com)
16 comments

Aaaand...POUNCE! (i.imgur.com)

cats
subscribe 309,917 readers
203 users here now
Everything about cats.

Never gonna give you up

Lyric overlay on r/cats

Key Considerations

How will your app handle data persistence?

Utilize system content provider for media files
Store lyric files in external storage
Store location of lyrics files in content provider
Store playlists in internal storage
Store current playlist in shared prefs
Store text settings in shared prefs

Describe any corner cases in the UX.

Media playing logic will be contained in a foreground service. The app can be exited and killed and will update its UI based on information from the service.

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

Retrofit - ease of calling the backend lyric repository service

Design Library - retroactive material theme
Google Play Services, Ads - monetization
Google Play Services, Analytics - utilization statistics
Bypass - render Markdown
ambilwarna - color picker preference

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

- Generate new project in Android Studio
- Add libraries to build.gradle and sync

Task 2: Implement UI for Each Activity and Fragment

- MainActivity
- SettingsActivity
- LyricService
- SettingsActivity
- LibraryFragment
- LyricsFragment
- PlaylistFragment
- NowPlayingFragment
- DownloadLyricsActivity
- ViewLyricsFragment
- AudioControllerNotification
- AudioControllerWidget

Task 3: MainActivity

- Generate DrawerLayout
- Configure logic for switching to fragments via drawer

Task 4: Implement Google Play Services

- Ads

    - Create ad id
    - Add to UI
    - Ad retrieval code
- Analytics
    - Create analytics id
    - Setup in code

Task 5: Create Foreground Service

- Create lyrics layout
- Create notification
- Create audio playing logic
    - Accept broadcast intents to control playback
- Connect to necessary activity/fragments
- Create thread to update UI based on playback time

Task 6: Settings Activity

- Generate Settings Activity
- Create default values
    - text color
    - stroke color
    - text size
- Create ColorPickerPreference
- Create SliderPreference
- Add each item to SettingsActivity

Task 7: Databases

- Create databases for lyrics and playlists and corresponding:
    - loaders
    - helpers
    - contracts

Task 8: Connections

- Connect MainActivity to LyricService
    - expose methods to children fragments
- Connect respective fragments to content loader
- Create adapters to render data

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**”