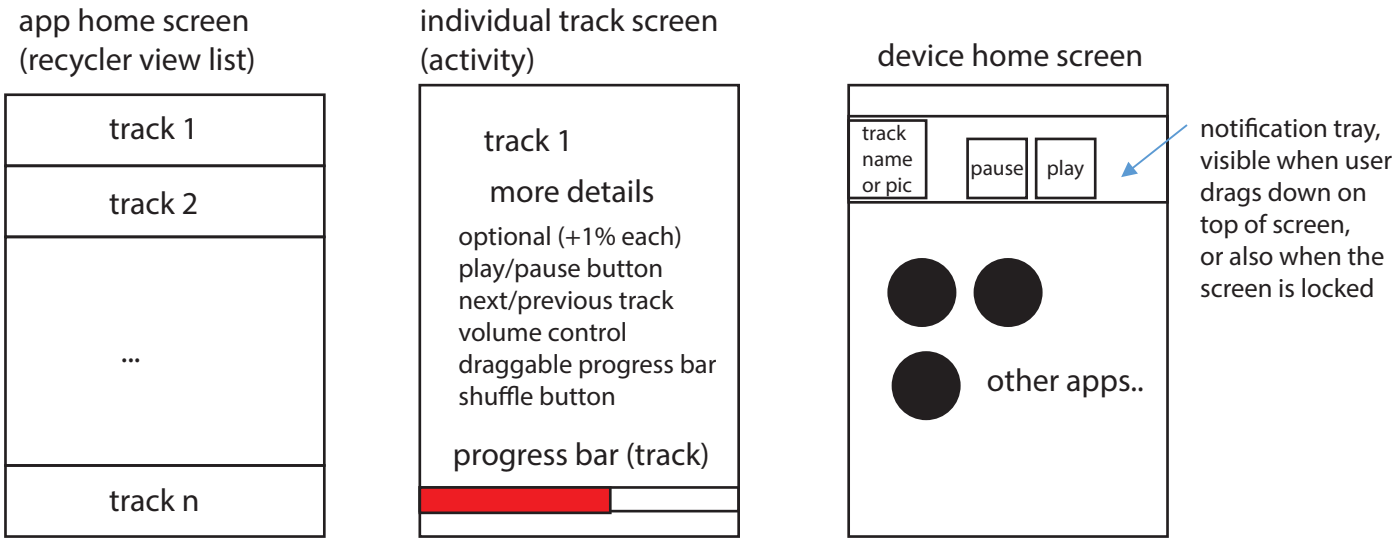
Assignment 2 CS230 Developing Mobile Apps
Fall 2019, Professor: Russell Butler, Johnson 114A, Office Hours MWF 9:00am-11:00am
Due Wednesday October 30, 2019, worth 20% of final grade
you may work in teams of 2 or less

The audio player
The goal of this assignment is to design and implement an audio player using Android best practices
Requirements:
1 (10%) - display list of .mp3 meta-data (song title, album, artist name, etc.) using RecyclerView
2 (20%) - select .mp3 file from list, open new activity/fragment, play audio and display progress bar of track length
3 (30%) - when user exits activity 1 or 2, audio will continue because it is played from a Service
4 (20%) - if app is running, notification present in device's notification tray, with controls to play/pause track
5 (20%) - audio volume responds to device controls (increasing phone volume increases app's volume)

Bonus:
(+5%): a search bar in activity from (1) where user enters text to filter recycler view contents according to matching strings (string can come from any part of track meta-data (artist, title, album, etc.)
(+5%): read the tracks from the user's music folder on their device (instead of just the tracks provided)
(+5%): all the features in "individual track screen" below

Main concepts to be learned:

Service, Foreground Service: https://developer.android.com/guide/components/services
Notifications: https://developer.android.com/guide/topics/ui/notifiers/notifications
Media Player: https://developer.android.com/guide/topics/media-apps/audio-app/building-an-audio-app
Client/Server pattern:
https://developer.android.com/guide/topics/media-apps/audio-app/building-a-mediabrowserservice



The audio player app is similar to the email app (assignment 1) in many ways (recyclerView that opens into another activity with more details). The main new features are the service, that keeps the media playing even when the user exits the app, the use of notifications, so the user can control the player from outside the app, and the client/server pattern, through MediaController, MediaBrowser, and MediaBrowserService