Case Study Car Audio

By Sunny Sur

Date of start	11.09.2023
Tech Case Demo	Demo
Figma	Figma
Video showcase	Duo App showcase
Git Personal	Git personal
Git Duo	Git Duo
Date of submission	08.10.2023
Authors	Sunny Sur
Duo case	Car Audio

Fontys
University of Applied Sciences

Content

Project Overview

Introduction

Our project is called the car audio project and for the next 4 weeks me and Christiyan intend to solve a problem that could possibly prevent fines and accidents on the road. Staying connected while on the road in this day and age is a necessity. Drivers unfortunately often need to take their hands off the wheel to control various inputs on their phone, such as changing music, using the gps or a call. This case study explores the implementation of a car audio app feature that addresses this safety concern by introducing a bluetooth button on the steering wheel.

Problem Statement

The main problem at hand is that drivers have to frequently take their hands off the steering wheel to interact with a music app on the phone. It’s a significant risk that diverts the users attention from the road. Our team split this main problem into multiple other smaller research questions which have made our research easier. You can read about this in more detail at the user research part of the case study. Our attempt to solve this problem is meant to allow users to control the audio system without compromising any safety.

Objectives

The main objectives of this project are ensuring the users safety by developing something that minimizes distraction by allowing audio control while both hands remain on the steering wheel.

The second objective we kept in mind is improving the user experience. By looking at comparisons of the already existing music apps our team will try to create a more intuitive and user-friendly interface that makes audio control even more easy and accessible.

Our third objective was making this app feasible and for the most important class of people which are young and inexperienced drivers since they get in the most accidents and trouble with the law when driving. Another important factor is older cars since these don’t have modern safety features and can be more prone to accidents. So we intend to make this app and feature targeted especially towards them. We decided not to focus on modern cars which already have integrated car audio systems with the built in screen since this app will not be as desired by them

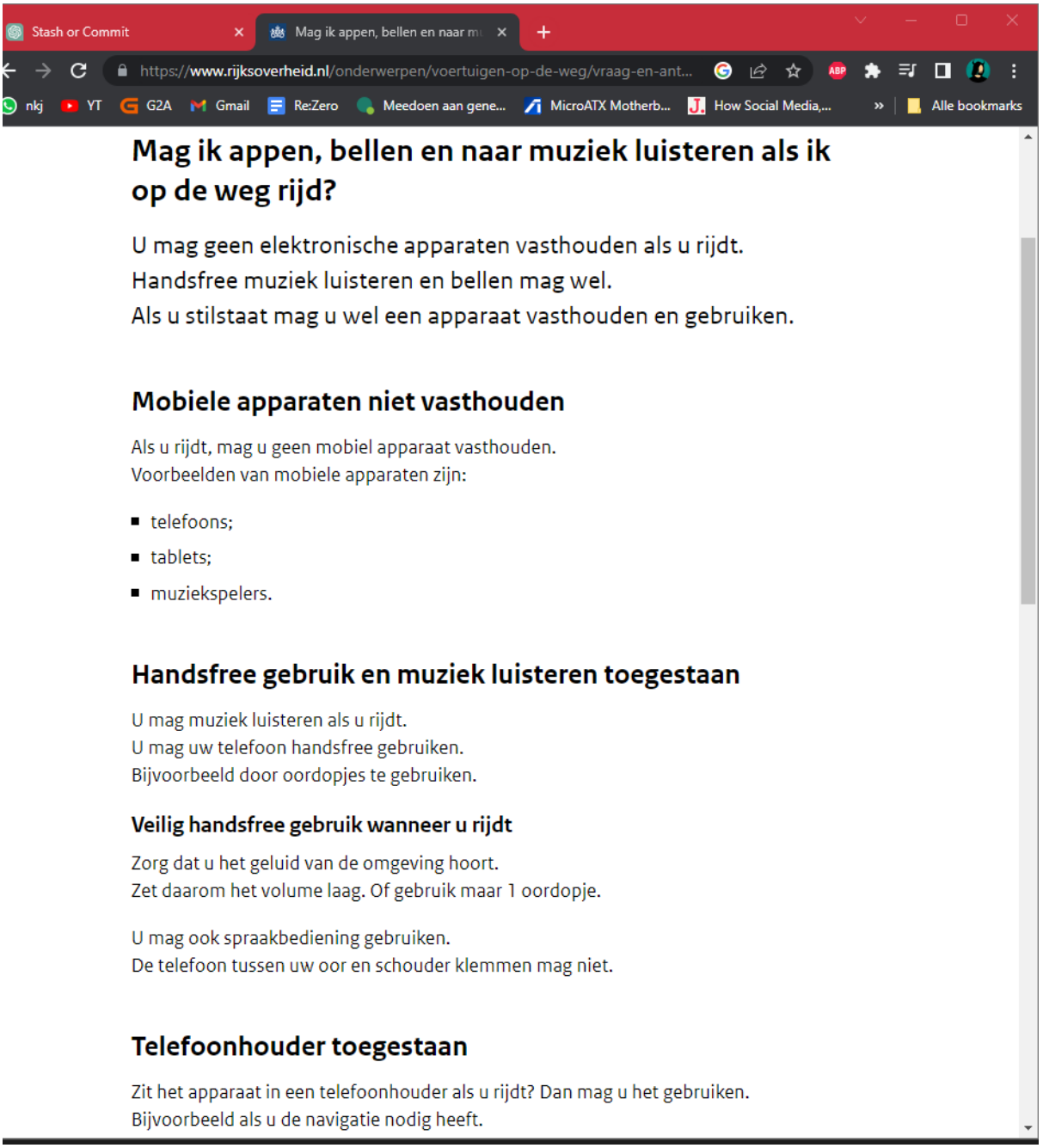
Our last objective is adhering to all the relevant safety regulations and laws. Research will be done about what exactly is allowed when driving and using the phone. If there are any flaws in the law which are still safe (think of using the phone at a redlight) then we intend to use those laws to our project's benefit. While still respecting the law of course. We want to warn the users about possible dangers when driving distracted and what laws they can break by doing so.

Sub questions

- What are the exact laws we can use in order to make our product feasible and as legal as possible?
- What other ways besides voice integration and gestures could we implement safety into our product?
- What are the actual major flaws most used music apps at the market have currently?

Methodology

To answer our first sub question I did a small online research. I researched both European laws about this and Dutch laws since this is where our user research was done. The main website I will use for this will probably be rijksoverheid.nl.



Then our second research question is going to be answered by more online research and potential interviews. This could also be answered in surveys but we decided to not use them since our experience proved that interviews are more valuable. Mostly ChatGPT was used.

Our third research question is also going to be answered by the use of qualitative interviews and market research. Understanding what flaws are most common with music apps will allow us to get rid of these and keep user contentment at a maximum

Summary

The development of our car audio app aimed to get rid of the general problem of the user having to take their hands off of the steering wheel. Providing our users with a seamless audio experience which was meant to also be very customizable. We addressed both technical and design challenges, ensuring a polished and reliable user experience. The main 3 ways we fixed this problem were by the use of gestures, voice integration and a simple UI. Overall the project met most of its goals and we hope that it will make the difference in the future of car audio apps. Additional updates of making things even more functional would be ideal too.

Expected Results

Our general expectations with this app was to reduce the amount of times the user looks at his phone during driving by a considerable amount and with the use of our a/b testing we can safely say that we successfully did it.

Research

User Research

Our team's user research consists of interviews. The main research question was split into some others and a few of these became interview questions. The following seven questions were asked to eight people. The first 2 questions were asked in order to know whether the user fit our target audience. I managed to interview 6 people near the TUe campus.

Do you drive an older car?
This question was asked in order to understand whether the user had a car with an already built in screen and audio system of the car or not.

Do you listen to your own music using your phone or do you listen to your radio?
This question was asked to get a feel if the user actually uses apps like spotify while in the car or not. In case the user listened to the radio he would not be a viable interviewee.

What do you dislike about the music app you use?
This question was asked so we can identify current pain points and enhance user experience.

What do you usually do if you try to use your phone when driving? Do you try to change the song or adjust the volume or repeat. Do you try to change the song or adjust the volume or repeat. Which of these actions do you do the most?
This question was asked to understand how the user exactly interacts with their phone when driving. So we can make these things easier in our app and also ignore any features that could be redundant.

How do you use your phone when driving? Is it in your hand, do you have a holder or do you not interact with it?
This question was asked to understand where the user tends to interact with their phone so we can get a feel for the placement of the bluetooth device and keep in mind what is preferred.

What would you think about a bluetooth device that would be on the steering wheel allowing you to switch songs without much distraction.
This question was asked in order to get direct feedback on our idea and understand if people would actually buy it or not.

How would you ideally like to change your music in your car? Is there any preference you have or do you recommend anything?
This question was asked to get any final recommendations or ideas from the user we might not have thought about.

Key findings

Expected Findings

The general findings we found were aligned with what we expected. These are of course still valuable since they prove to us that the user indeed does have the same pain points that were expected of them. However the true value comes from the key findings we did not expect and these allow us to potentially make more solutions. The pain points which were expected and found in the interviews were the following: taking hands off steering wheel, distracting UI and bad voice control.

Unexpected Findings

The unexpected pain points were that 2 interviewees both have problems with podcasts rather than changing the music. Another pain point an interviewee told us to look at was that if we use our idea that we should look at volkswagen golf since this car has the same button idea as us. Lots of reviews of that car complain about the buttons being accidentally pressed when trying to park. Another pain point was that one of the users ideally didn't wanna change the music at all, he prefers good mood playlists (generated by the app). The last pain point we didn't expect was that someone wanted an easier way of raising volume. We didn't expect this to be a pain point since the radio button could be used as a volume changer but apparently it can still be an issue. We should aim to minimize these problems.

The full list of answers can be found in the Figma document. **Interview questions and answers can be seen HERE**

Market Research

	Spotify	Apple Music	Amazon music	YouTube music	Tidal
Users	551 milion	88 milion	80 milion	50 milion	7 milion
Target Audience	Spotify's target audience includes music enthusiasts of all ages, especially Millennials and Gen Z, who prefer digital music streaming on mobile devices, have diverse musical tastes, and may also be interested in podcasts.	Apple Music's target audience includes a broad range of music lovers, Apple device users, and individuals interested in a seamless integration with the Apple ecosystem.	Amazon Music's target audience includes music enthusiasts, streaming subscribers, and Amazon Prime members looking for a diverse catalog of songs, playlists, and personalized music recommendations.	YouTube Music's target audience is music enthusiasts of all ages who enjoy streaming and discovering a wide range of music genres, artists, and music-related content online.	Tidal's target audience is music enthusiasts and audiophiles who prioritize high-quality audio streaming and exclusive content.
Free service	Yes	No	Yes	Yes	No
Subscription cost	$10.99	$9.99	$9.99	$10.99	$9.99
Available on	Android, IOS	IOS	Android, IOS	Android, IOS	Android, IOS
Voice assistant	Google	Apple Siri	Alexa, Google(?)	Google, Apple Siri	Google
Main selling point	music and podcasts	Apple users Audio app hi res audio	music and podcasts	All the music with videos	HiFi Audio
Integration	Google Home, Waze, Google maps, android wear, garmin wear	Waze	amazon alexa	google voice assistant	Google assistant, Android wear, Garmin
UI complexity	Not complex	A little bit complex	A little bit complex	Not complex	A little complex but easily gets used to
Offline mode	Yes	Yes	Yes	Yes	Yes
Max Audio Quality (higher is better)	320kbps	2304kbps	320kbps	256kbps	5000+kbps
Video support	Yes	No	No	Yes	Yes
Family plans	Yes	Yes	Yes	Yes	Yes
Family plans	Yes	Yes	Yes	Yes	Yes
Free trial period	Yes	Yes	Yes	Yes	Yes
Trial period	90 Days	90 Days	30 Days	30 Days	30 Days
Landscape car mode	Yes	Yes	Yes	Yes	No
Podcasts	Yes	Yes	Yes	Yes	Yes
Lyrics display	Yes	Yes	Yes	Yes	Yes
Equalizer	Yes	Yes	Yes	No	Yes
Wakelocks	Yes	No	No	Yes	No
Colorblind mode	No	No	No	No	No
Voice control	Yes	Subscription	Yes	Yes	No
Voice control by default	No	Subscription	No	Yes	No
Customizable text	No	No	No	No	No
TTS	Yes	Yes	No	Offers third party services	Unknown
Equalizer	Yes	Yes	No	Yes	Yes
Alternative text for image					
Gestures support	No	No	No	Android only	Removed
Gesture customization	No	No	No	No	No

I personally didn't do much for this one. All I did was do everything underneath the colorblind mode. Trends we saw were that Spotify is the most used app for a reason. Most of the features we put in, Spotify already had. It seemed like the most complete app out of the 5 things we researched. What I did do however is look at the exact UI's of all 5 of these apps

and see if I could get the best of what all 5 of these have to offer turning the prototype into a safer mixture of all of them.

Tech Research

Doing Marcel's assignment with the SMS and the other workshops, also my tech case, helped with understanding the basics of Kotlin. I personally researched how to do gestures, use media players and how to do vocal recognition in kotlin. I did this with the use of chatgpt and youtube tutorials.

Ideation

Brainstorming and ideas

	Vocal recognition	Hey "App Name" to start	Phone holder	Steering wheel incl bikes thing	Arduino		Copy the headphones Gesture thing	Large swiping motions
	Vocal interaction	Customize call out/train voice assistant	Voice activation/command	Physical interaction	Swipe gestures while the phone is on the steering wheel		Gestures	Should be customizable
Video https://www.youtube.com/watch?v=V-WCzgjXqkA	Tutorials with safety maybe this	Small warning or awareness of distracted driving	Vocal interaction	Physical interaction	Gestures	Both landscape and vertical UI	Easy navigation	Play/pause/skip/volume buttons
Different skipping presets	Extras	Lower volume when GPS talks	Extras	Car audio Ideas	Simple UI	Large text High contrast and icons	Simple UI	Album cover view on/of button (no distraction)
Potentially dim screen to reduce distraction	Preset playlists both for music and podcast	Screen Reader alternative tag for disabilities	Podcasts	Loopholes with the in the EU	Camera Use	GPS could be a shortcut as an app in the ui (Apple)	Shuffle repeat	Do car mode for different simplified ui
				Phone cannot be in hand	Cameras check	Eye tracking	Body gestures	
	Podcasts	Different UI from music page		Loopholes within the EU	Bluetooth is allowed car phone systems	Looking down at the phone automatically switches to maps	Camera Use	
	UI could show random unwatched podcasts of channels user like	Ask what podcasts the user is interested in when registering		Red light allows changing song or radio	Some EU countries have banned handsfree for younger drivers		Scan mood and suggest playlist	

I made the majority of this Lotus Blossom that allowed us to have every potential feature in one file. (Check Figma for better image quality).

Chosen ideas

In the end we scrapped a lot of ideas since the scope was getting too big. Things we kept from this lotus blossom were mainly the vocal recognition, gestures, simple UI and radio.

The reasons these were chosen is because we had to actually make the features feasible with our current skills. The main idea we wanted to pursue with the button was scrapped since this involved other software we are not specialized in which would take too much time. A way of preventing this would have been starting coding earlier. We barely made it with just these features. A week before coding we decided that the main features should be the focus for now and that the other features like buttons could come later but there was no time for it unfortunately.

Design Process

Prototyping

Our process went as follows: I made a version in figma and Christiyan made one. My focus was on making the user as familiar as possible with the UI and sacrificing a bit of features for it. Basically the UI only has big icons and big text of course but besides that there is a core feature that takes pride in its small range of options. So in my version you can see pre-picked albums, songs and artists. The user themselves know where everything is once they're accustomed to the app. The reason why the user knows this is because he/she is the one that basically decided the layout of the app. This in theory would have offered the user to simply use the app without looking but when my idea was tested we chose to go with Christiyans version. The reason why we went with his version is that it's more accessible then mine. Mine when tested appeared to have a learning curve which we should have avoided.

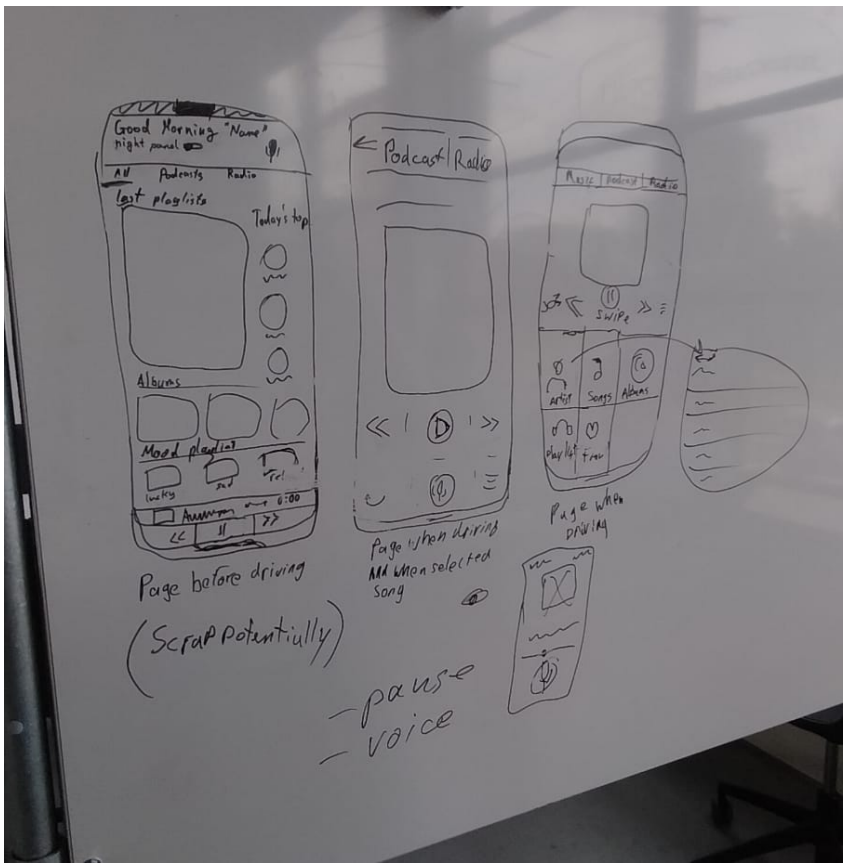
Christians app focuses more on vocal recognition then mine and focuses on having to tap on less stuff in order to listen to music. When we started wireframing the main criticism we got for Christiyans main page was that it was too cluttered and the icons were too small so I tried to make mine have bigger icons

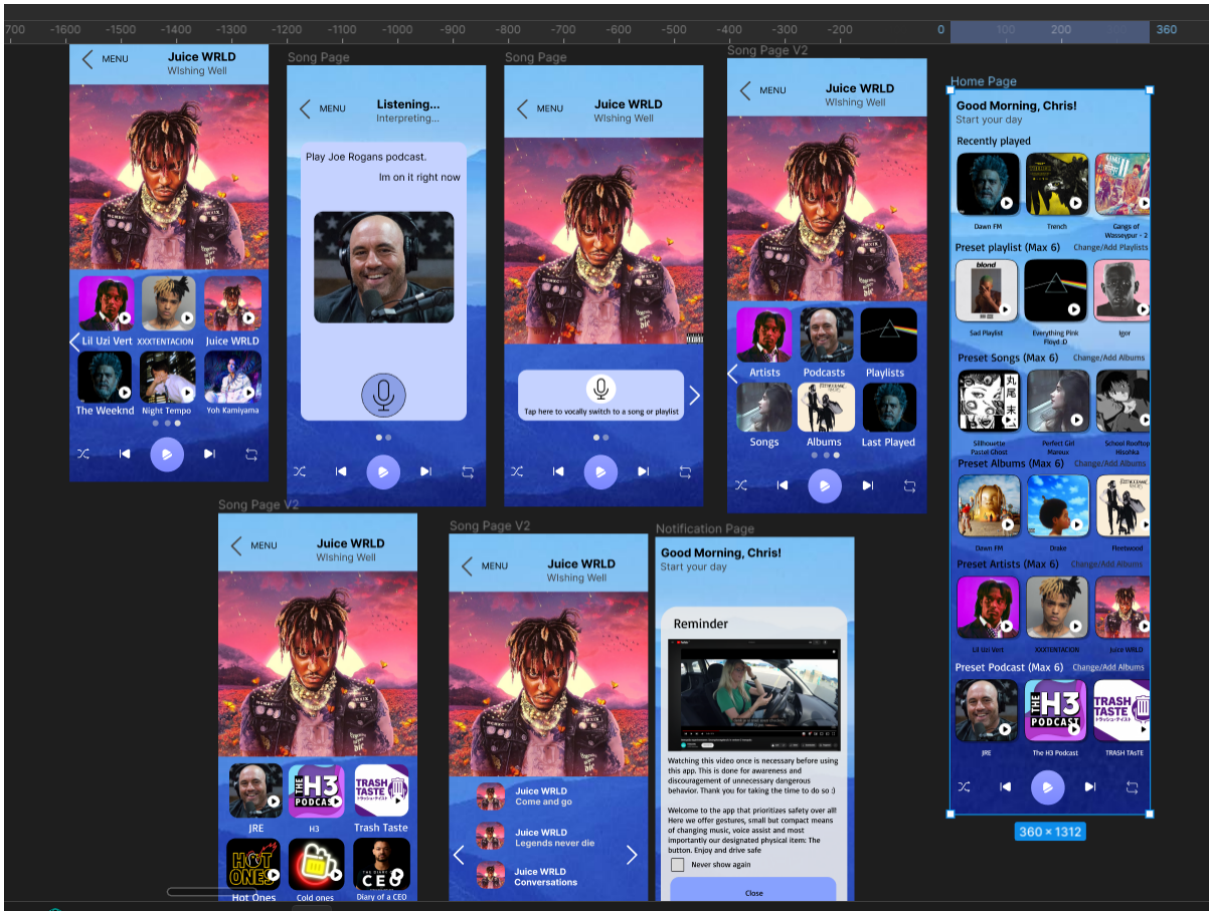
User Testing

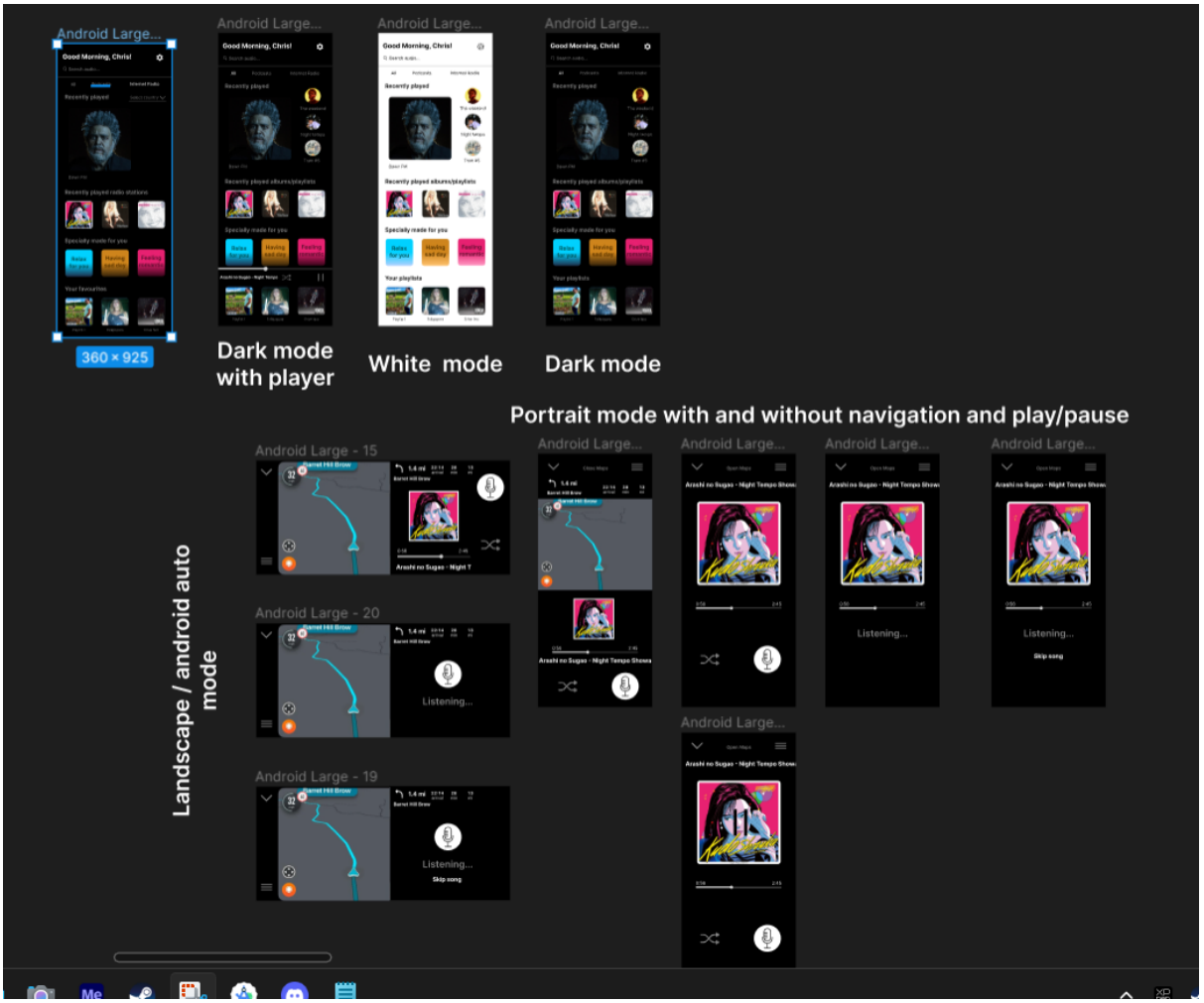
A/b testing I asked Nigel about my prototype and managed to improve minor things in my figma but this didn't really prove to be that useful since we didn't go with my version anyway. I did however still apply the things he asked me to. I recorded Jayden being tested with the setup Christiyan created with the wheel and the game. However I couldn't be there for the entire testing day since I had to leave earlier. The teachers were tested too and our general findings were that Christiyans app was way less distracting than mine. Overall people looked away around 5 times with my app and with Christiyans app they looked at the phone 2 times at max. I believe with more experience my app still could be preferred but that might also not be the case due to the testing.

Test with Jayden (from the front in order to see how distracted he is)
A/B Testing 1 (My prototype)
A/B Testing 2 (Chris prototype test)
Test wil Nigel (My prototype)

Iterations


Good Morning "Name"
night mood
All
Podcasts
Radio
last playlists
Today's top
Albums
Mood playlist
Page before driving
(Scrap Potentially)
Podcast | Radio
Page when driving
AM when selected
Song
SWIPE
Artist
Songs
Albums
Page when driving
_pause
_voice

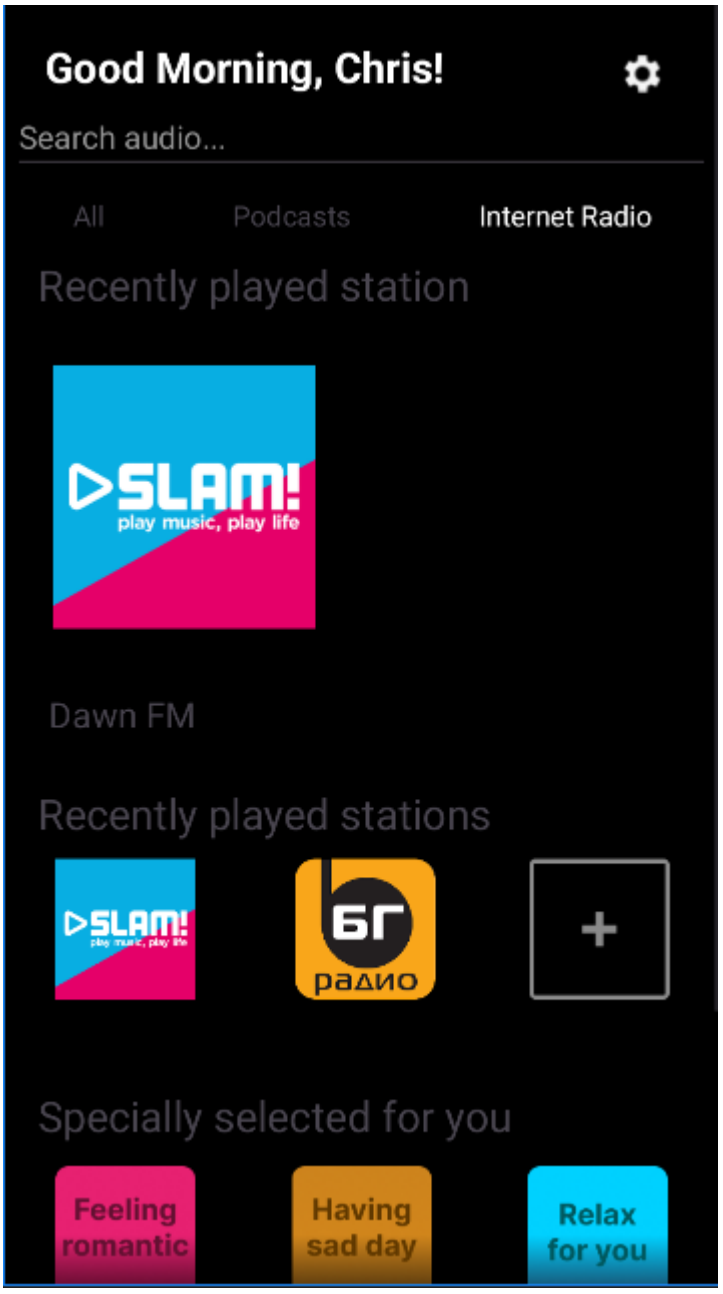

MENU
Juice WRLD
Wishing Well
Lil Uzi Vert
XXXTENTACION
Juice WRLD
The Weeknd
Night Tempo
Yoh Kamiyama
Song Page
Listening...
Interpreting...
Play Joe Rogans podcast.
Im on it right now
Song Page
Tap here to vocally switch to a song or playlist
Song Page V2
Artists
Podcasts
Playlists
Songs
Albums
Last Played
Home Page
Good Morning, Chris!
Start your day
Recently played
Preset playlist (Max 6)
Preset Songs (Max 6)
Preset Albums (Max 6)
Preset Artists (Max 6)
Preset Podcast (Max 6)
Change/Add Albums
360 × 1312
Song Page V2
JRE
H3
Trash Taste
Hot Ones
Cold ones
Diary of a CEO
Song Page V2
Juice WRLD
Come and go
Juice WRLD
Legends never die
Juice WRLD
Conversations
Notification Page
Good Morning, Chris!
Start your day
Reminder
Never show again
Close

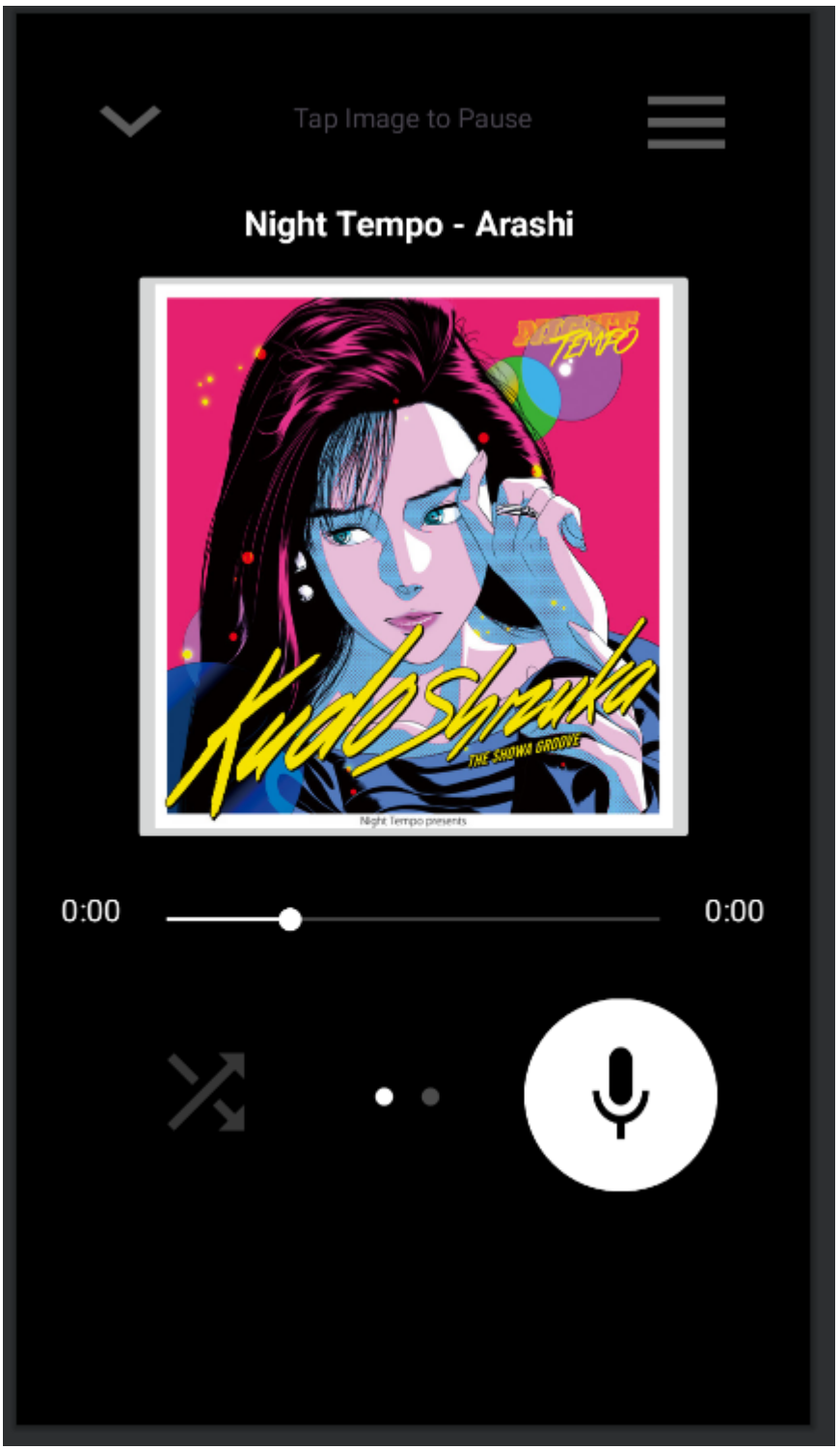


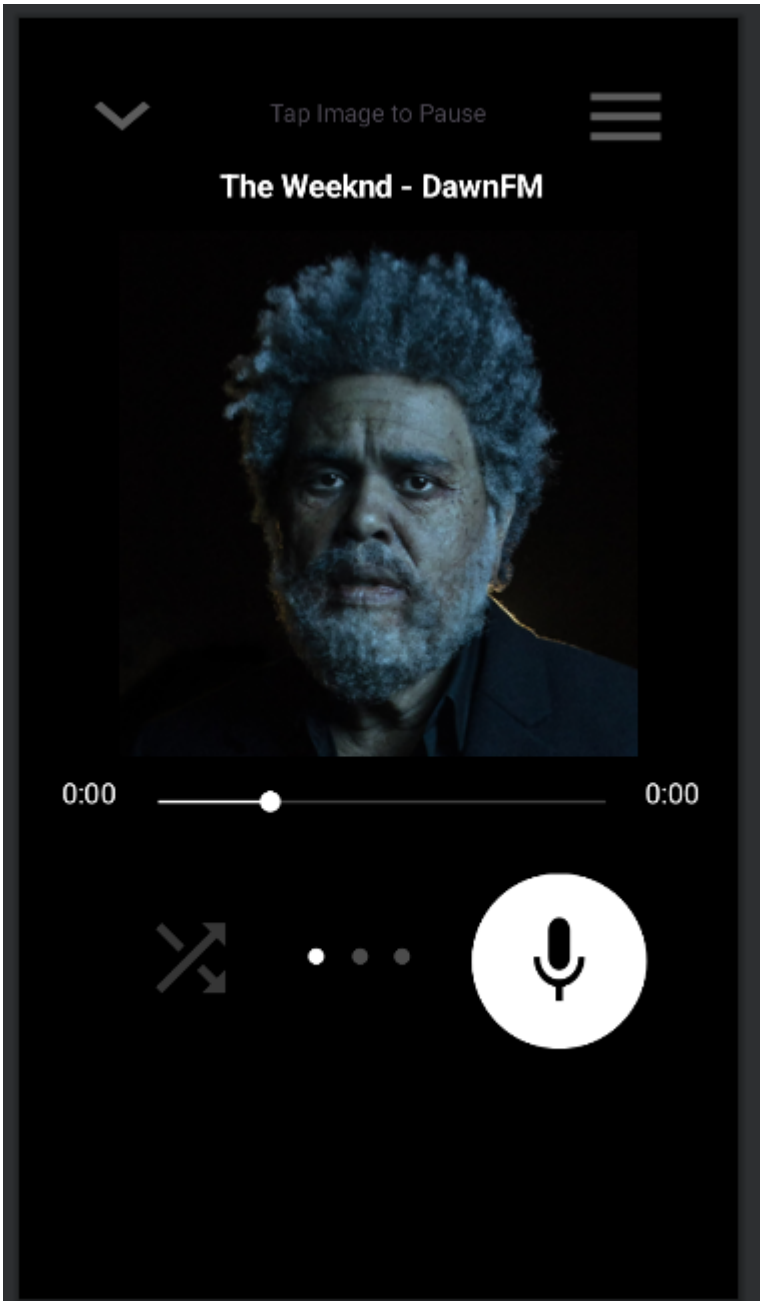
Unfortunately I wasn't able to capture more iterations than this. Lots of things did change but I didn't really make more images then this. For example in the final version we got rid of the map and the landscape options. Other changes were the pause button on the album cover. On my own version of the prototype the things I changed mostly had to do with the add playlist page. Nigel told me that it wasn't really clear that you could add or delete playlists. So I made this more clear. He also said that it wasn't clear how the voice button works so I added text to it.

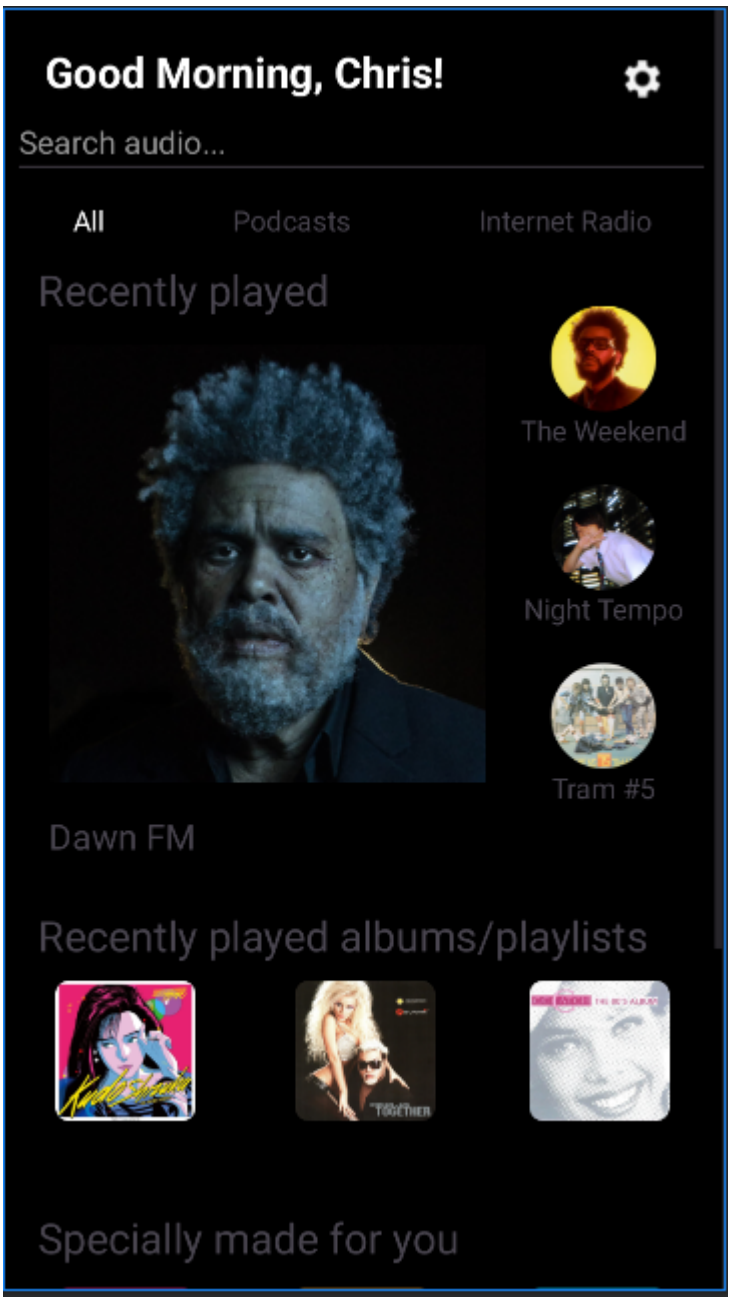
Final Solution

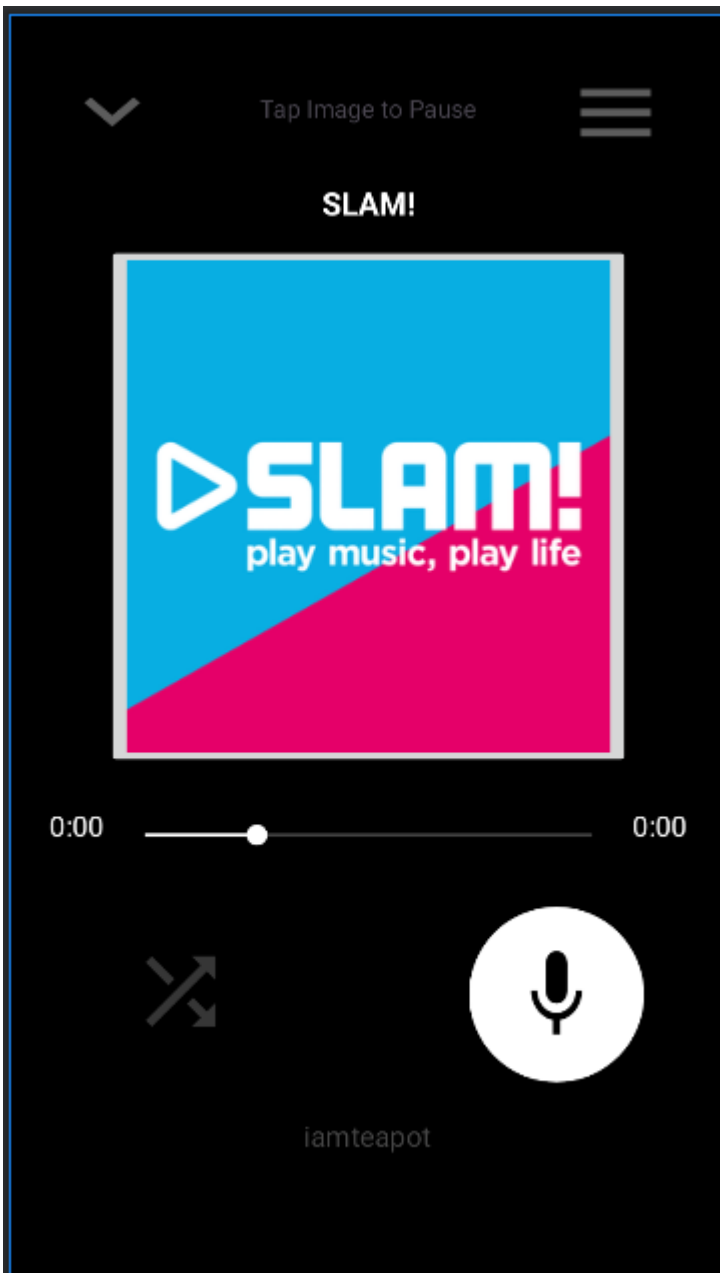
Showcase the Final Design




Tap Image to Pause
Night Tempo - Arashi
Kudo Shinku
THE SHOWA GROOVE
0:00
0:00

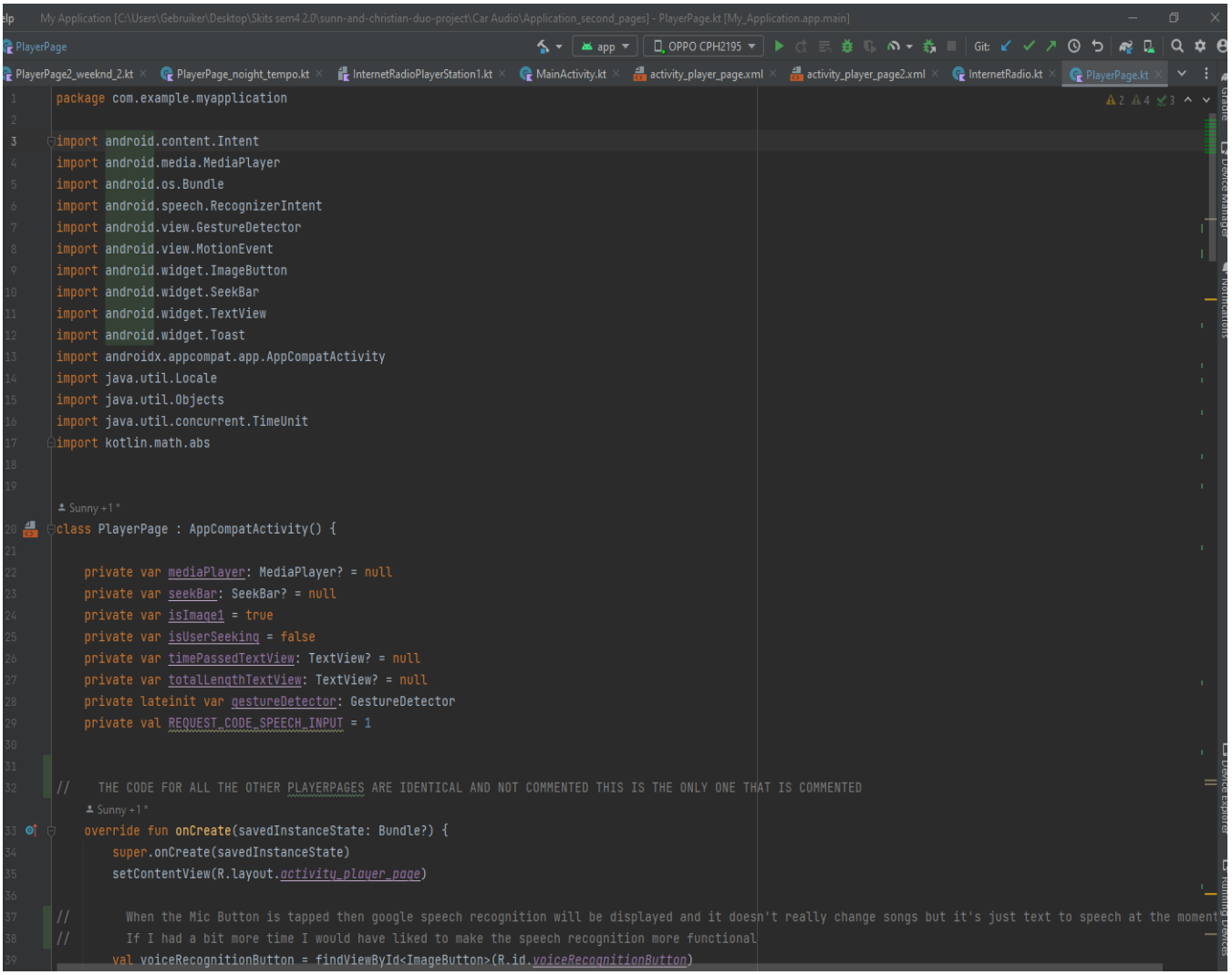

Tap Image to Pause
The Weeknd - DawnFM
0:00
0:00

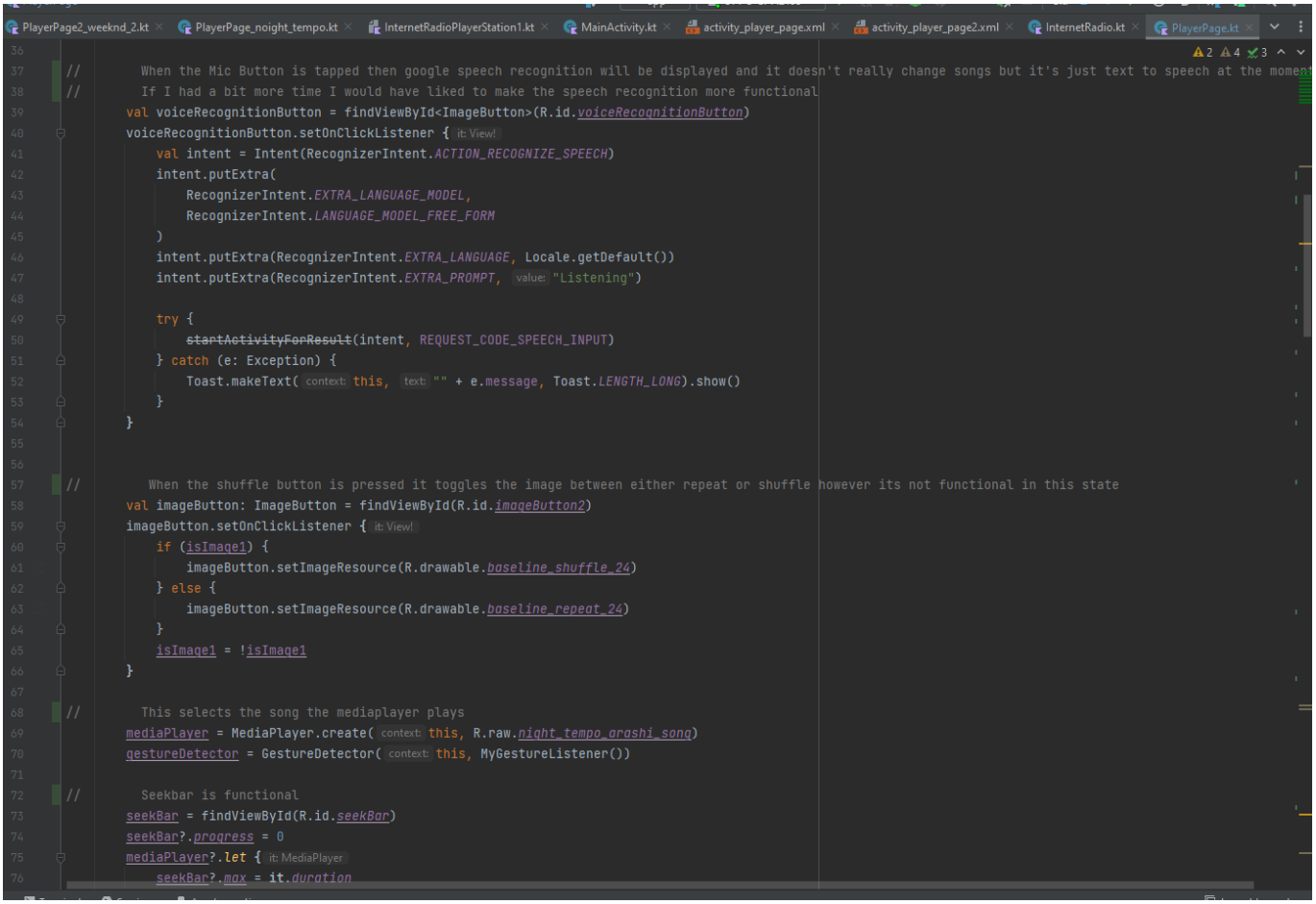

Good Morning, Chris!
Search audio...
All
Podcasts
Internet Radio
Recently played
The Weekend
Night Tempo
Tram #5
Dawn FM
Recently played albums/playlists
Specially made for you

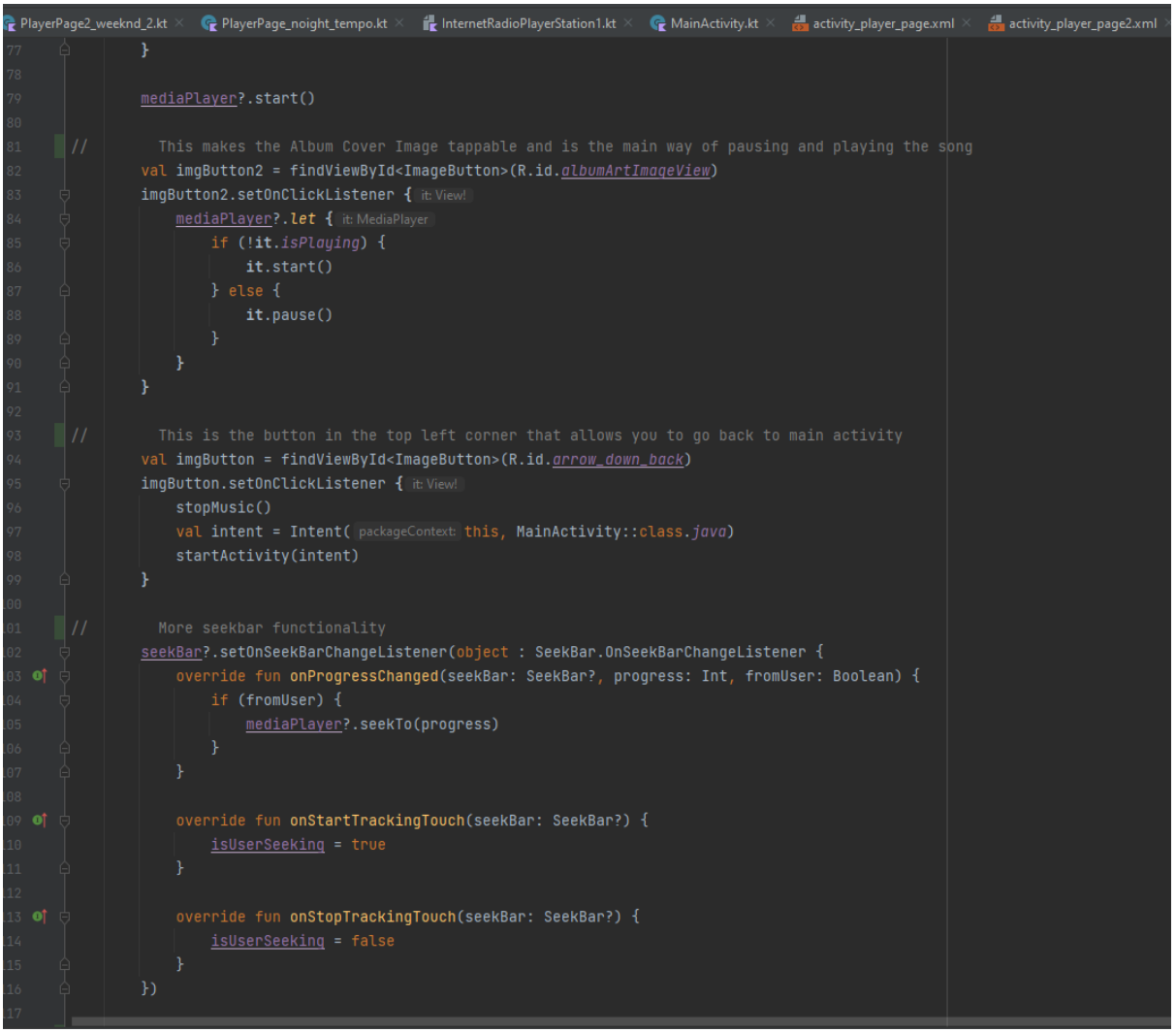


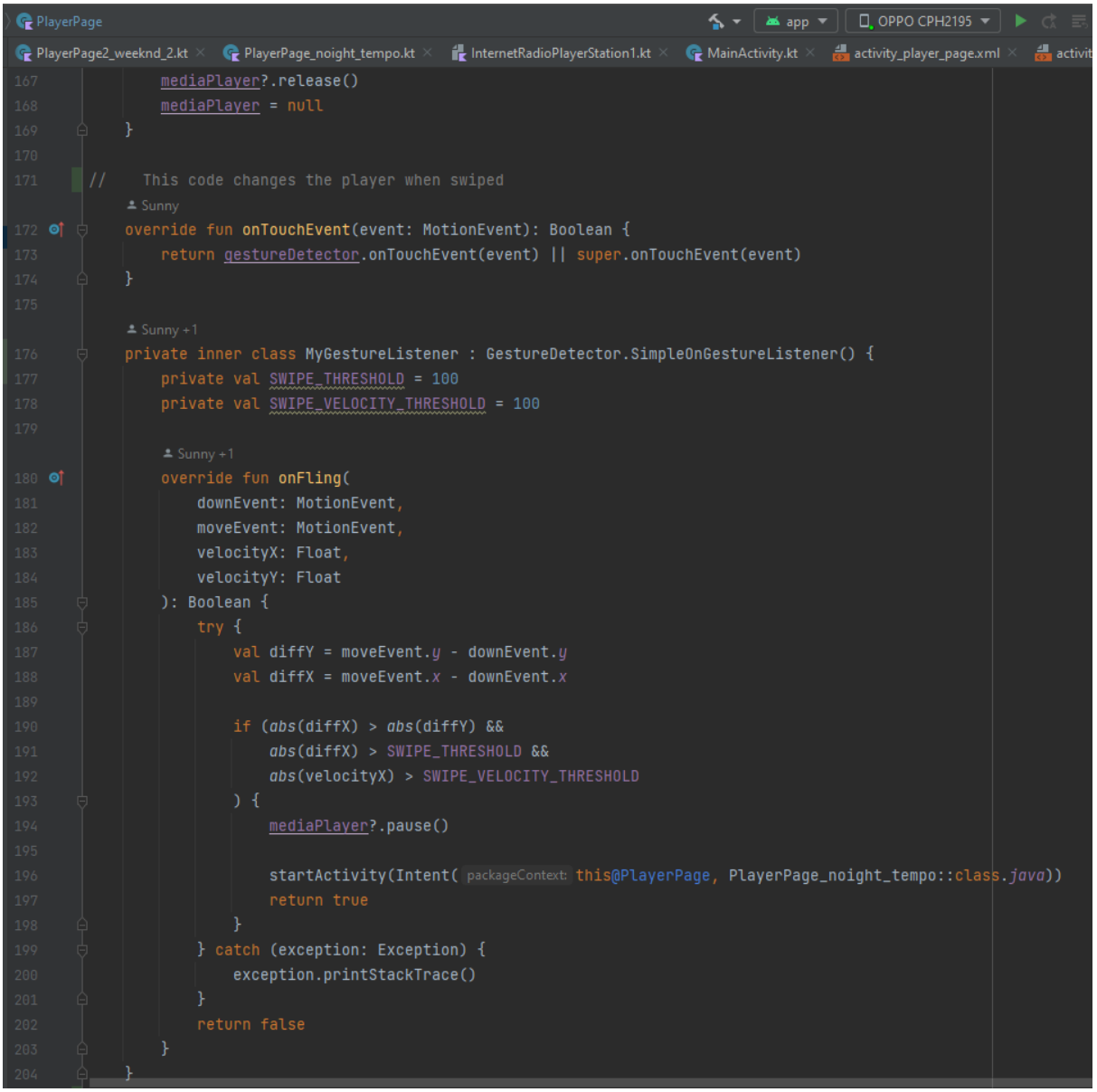
Showcase the Final Code

The following code is all mine, I made 5 Playerpages in Kotlin and XML where most of the code is identical.


My Application [C:\Users\Gebruiker\Desktop\Skits sem4 2.0\sunn-and-christian-duo-project\Car Audio\Application_second_pages] - PlayerPage.kt [My_Application.app.main]
PlayerPage
app
OPPO CPH2195
Git:
PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
activity_player_page2.xml
InternetRadio.kt
PlayerPage.kt
package com.example.myapplication
import android.content.Intent
import android.media.MediaPlayer
import android.os.Bundle
import android.speech.RecognizerIntent
import android.view.GestureDetector
import android.view.MotionEvent
import android.widget.ImageButton
import android.widget.SeekBar
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import java.util.Locale
import java.util.Objects
import java.util.concurrent.TimeUnit
import kotlin.math.abs
Sunny +1 *
class PlayerPage : AppCompatActivity() {
private var mediaPlayer: MediaPlayer? = null
private var seekBar: SeekBar? = null
private var isImage1 = true
private var isUserSeeking = false
private var timePassedTextView: TextView? = null
private var totalLenqthTextView: TextView? = null
private lateinit var gestureDetector: GestureDetector
private val REQUEST_CODE_SPEECH_INPUT = 1
// THE CODE FOR ALL THE OTHER PLAYERPAGES ARE IDENTICAL AND NOT COMMENTED THIS IS THE ONLY ONE THAT IS COMMENTED
Sunny +1 *
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_player_page)
// When the Mic Button is tapped then google speech recognition will be displayed and it doesn't really change songs but it's just text to speech at the moment
// If I had a bit more time I would have liked to make the speech recognition more functional
val voiceRecognitionButton = findViewById<ImageButton>(R.id.voiceRecognitionButton)


PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
activity_player_page2.xml
InternetRadio.kt
PlayerPage.kt
// When the Mic Button is tapped then google speech recognition will be displayed and it doesn't really change songs but it's just text to speech at the moment
// If I had a bit more time I would have liked to make the speech recognition more functional
val voiceRecognitionButton = findViewById<ImageButton>(R.id.voiceRecognitionButton)
voiceRecognitionButton.setOnClickListener { it: View!
val intent = Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
intent.putExtra(
RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM
)
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault())
intent.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "Listening")
try {
startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT)
} catch (e: Exception) {
Toast.makeText(context: this, text: "" + e.message, Toast.LENGTH_LONG).show()
}
}
// When the shuffle button is pressed it toggles the image between either repeat or shuffle however its not functional in this state
val imageButton: ImageButton = findViewById(R.id.imageButton2)
imageButton.setOnClickListener { it: View!
if (isImage1) {
imageButton.setImageResource(R.drawable.baseline_shuffle_24)
} else {
imageButton.setImageResource(R.drawable.baseline_repeat_24)
}
isImage1 = !isImage1
}
// This selects the song the mediaplayer plays
mediaPlayer = MediaPlayer.create(context: this, R.raw.night_tempo_arashi_song)
gestureDetector = GestureDetector(context: this, MyGestureListener())
// Seekbar is functional
seekBar = findViewById(R.id.seekBar)
seekBar?.progress = 0
mediaPlayer?.let { it: MediaPlayer
seekBar?.max = it.duration


PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
activity_player_page2.xml
}
mediaPlayer?.start()
// This makes the Album Cover Image tappable and is the main way of pausing and playing the song
val imgButton2 = findViewById<ImageButton>(R.id.albumArtImageView)
imgButton2.setOnClickListener { it: View!
mediaPlayer?.let { it: MediaPlayer
if (!it.isPlaying) {
it.start()
} else {
it.pause()
}
}
}
// This is the button in the top left corner that allows you to go back to main activity
val imgButton = findViewById<ImageButton>(R.id.arrow_down_back)
imgButton.setOnClickListener { it: View!
stopMusic()
val intent = Intent(packageContext: this, MainActivity::class.java)
startActivity(intent)
}
// More seekbar functionality
seekBar?.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
if (fromUser) {
mediaPlayer?.seekTo(progress)
}
}
override fun onStartTrackingTouch(seekBar: SeekBar?) {
isUserSeeking = true
}
override fun onStopTrackingTouch(seekBar: SeekBar?) {
isUserSeeking = false
}
})

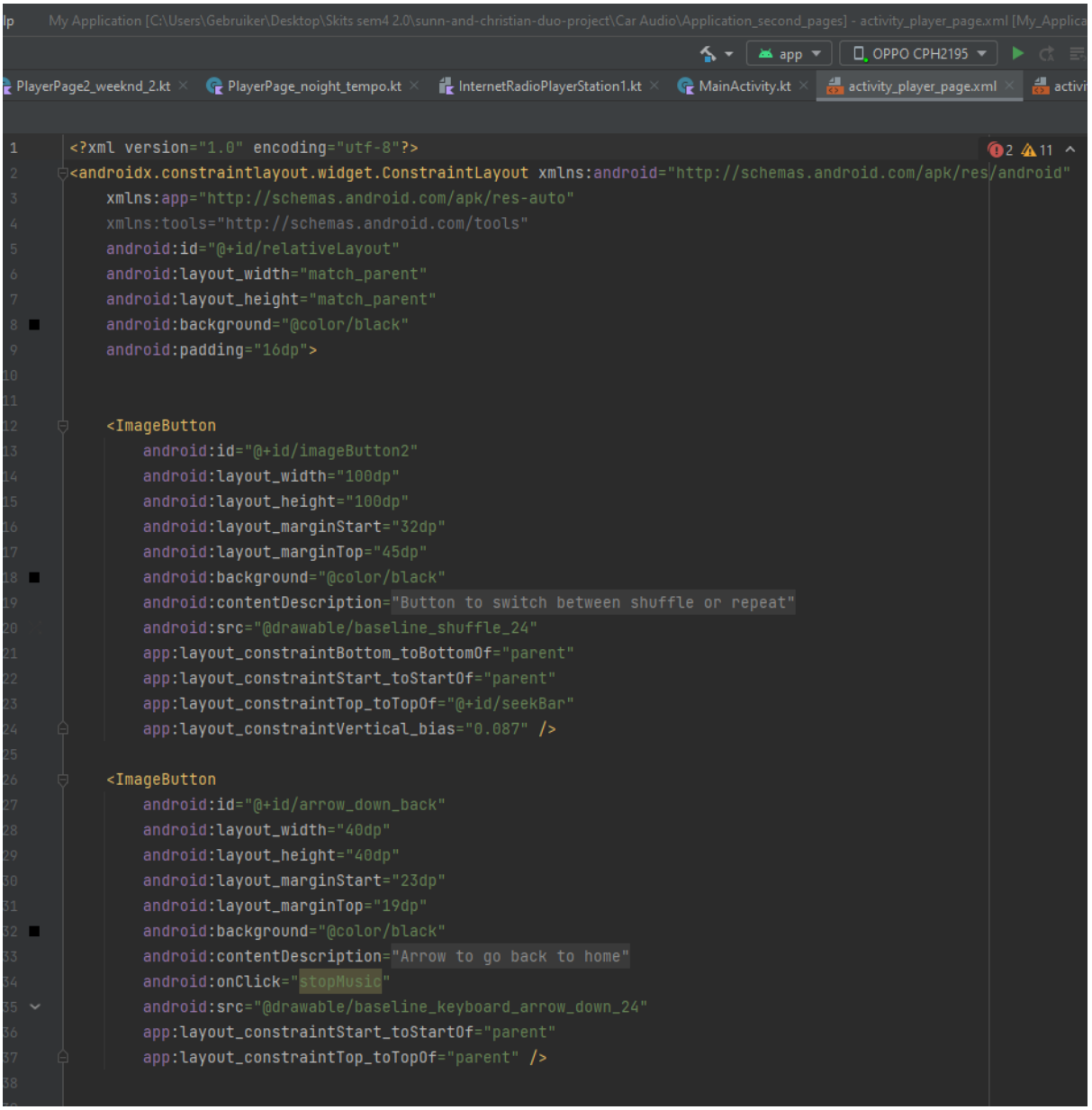

PlayerPage
app
OPPO CPH2195
PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
mediaPlayer?.release()
mediaPlayer = null
}
// This code changes the player when swiped
Sunny
override fun onTouchEvent(event: MotionEvent): Boolean {
return gestureDetector.onTouchEvent(event) || super.onTouchEvent(event)
}
Sunny +1
private inner class MyGestureListener : GestureDetector.SimpleOnGestureListener() {
private val SWIPE_THRESHOLD = 100
private val SWIPE_VELOCITY_THRESHOLD = 100
Sunny +1
override fun onFling(
downEvent: MotionEvent,
moveEvent: MotionEvent,
velocityX: Float,
velocityY: Float
): Boolean {
try {
val diffY = moveEvent.y - downEvent.y
val diffX = moveEvent.x - downEvent.x
if (abs(diffX) > abs(diffY) &&
abs(diffX) > SWIPE_THRESHOLD &&
abs(velocityX) > SWIPE_VELOCITY_THRESHOLD
) {
mediaPlayer?.pause()
startActivity(Intent(packageContext: this@PlayerPage, PlayerPage_noight_tempo::class.java))
return true
}
} catch (exception: Exception) {
exception.printStackTrace()
}
return false
}
}

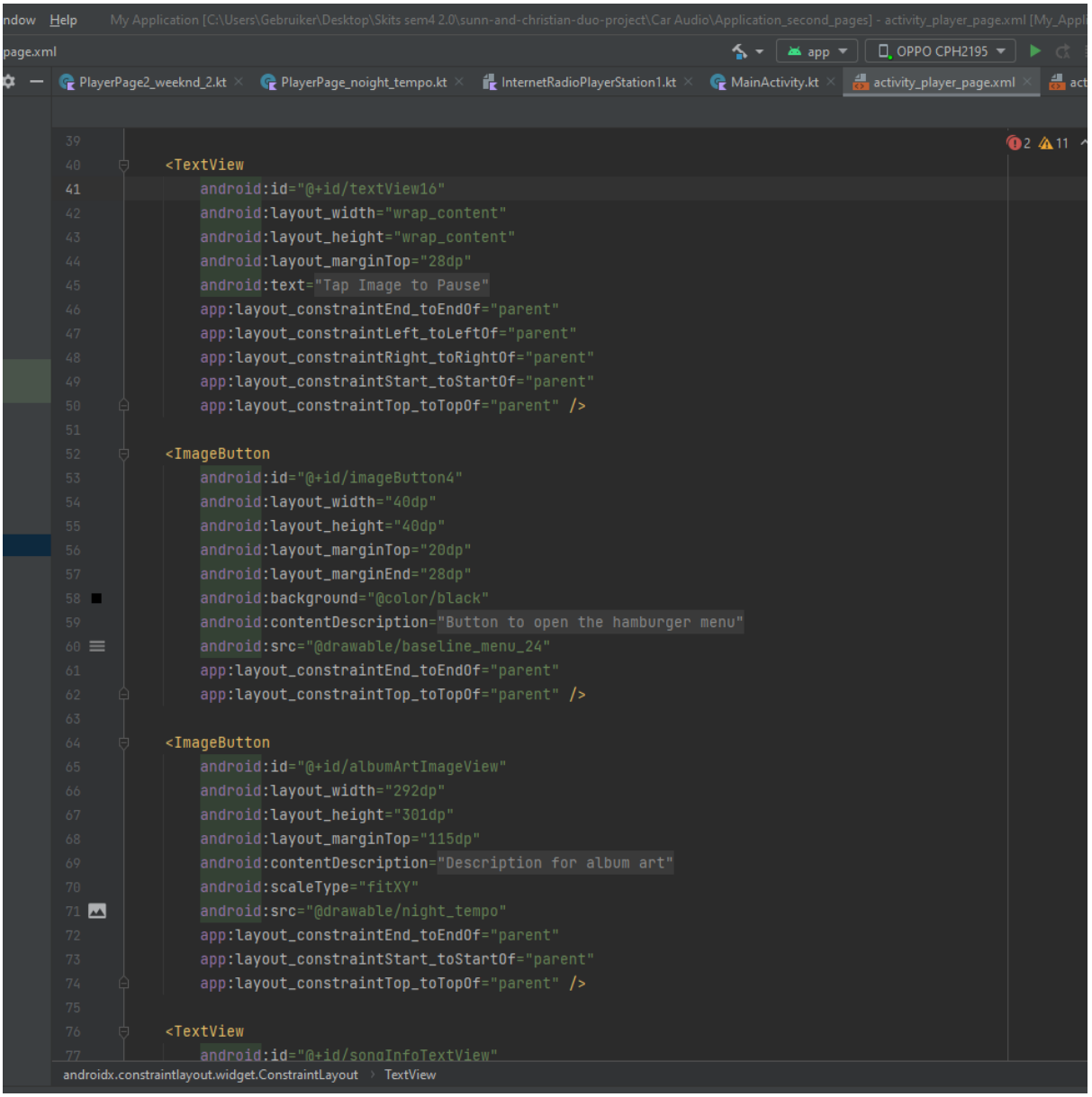
```
    }
//    The vocal recognition
    Sunny
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)

        if (requestCode == REQUEST_CODE_SPEECH_INPUT) {


            if (resultCode == RESULT_OK && data != null) {
                val res: ArrayList<String> =
                    data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS) as ArrayList<String>
                val tvtext = findViewById<TextView>(R.id.tvtext)

                tvtext.setText(Objects.requireNonNull(res)[0])
            }
        }
    }
}
```

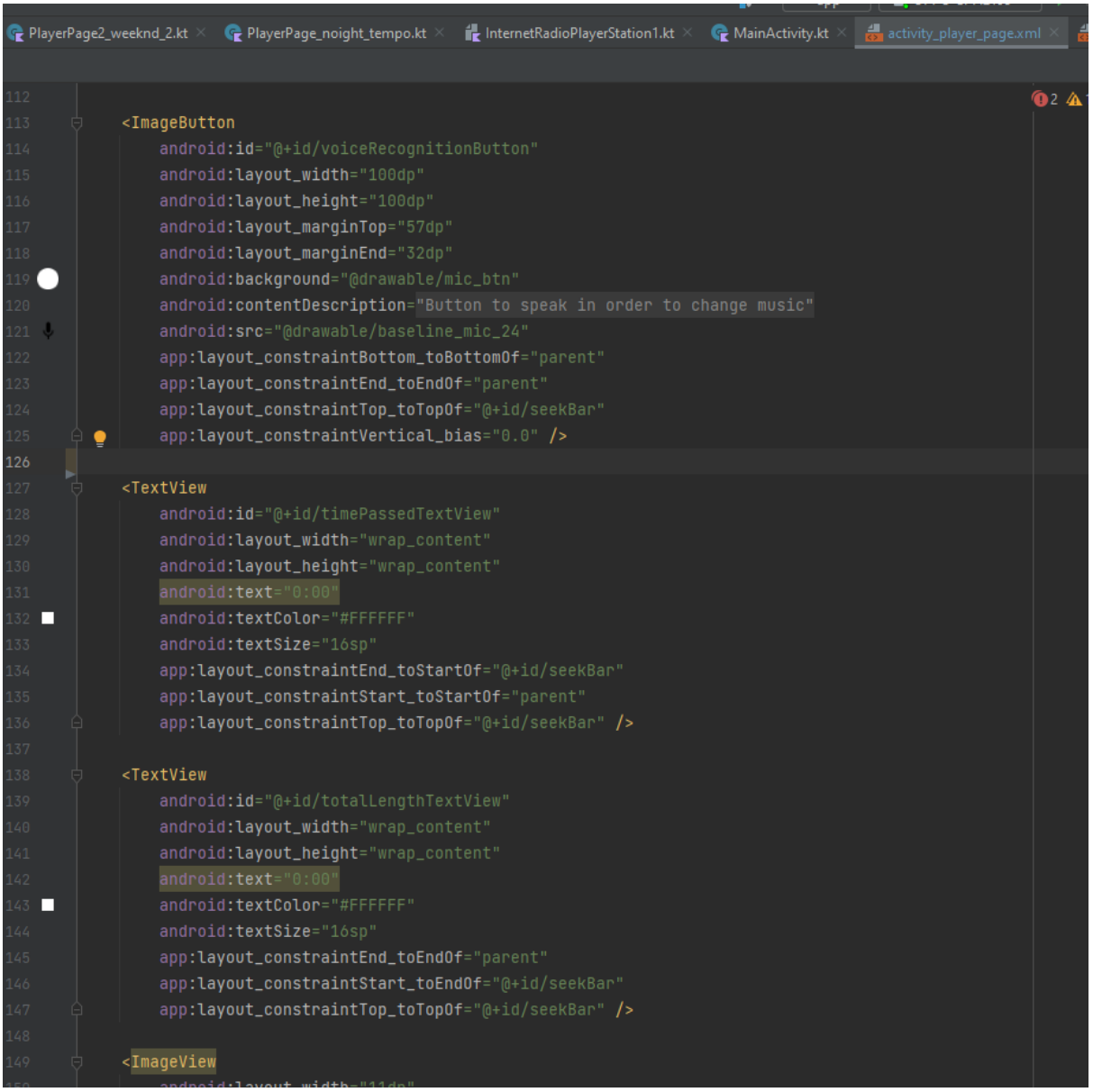

My Application [C:\Users\Gebruiker\Desktop\Skits sem4 2.0\sunn-and-christian-duo-project\Car Audio\Application_second_pages] - activity_player_page.xml [My_Applica
app
OPPO CPH2195
PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/relativeLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/black"
android:padding="16dp">
<ImageButton
android:id="@+id/imageButton2"
android:layout_width="100dp"
android:layout_height="100dp"
android:layout_marginStart="32dp"
android:layout_marginTop="45dp"
android:background="@color/black"
android:contentDescription="Button to switch between shuffle or repeat"
android:src="@drawable/baseline_shuffle_24"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@+id/seekBar"
app:layout_constraintVertical_bias="0.087" />
<ImageButton
android:id="@+id/arrow_down_back"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginStart="23dp"
android:layout_marginTop="19dp"
android:background="@color/black"
android:contentDescription="Arrow to go back to home"
android:onClick="stopMusic"
android:src="@drawable/baseline_keyboard_arrow_down_24"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
2
11


ndow Help
My Application [C:\Users\Gebruiker\Desktop\Skits sem4 2.0\sunn-and-christian-duo-project\Car Audio\Application_second_pages] - activity_player_page.xml [My_Appl
page.xml
app
OPPO CPH2195
PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
2 11
<TextView
android:id="@+id/textView16"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="28dp"
android:text="Tap Image to Pause"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<ImageButton
android:id="@+id/imageButton4"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginTop="20dp"
android:layout_marginEnd="28dp"
android:background="@color/black"
android:contentDescription="Button to open the hamburger menu"
android:src="@drawable/baseline_menu_24"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<ImageButton
android:id="@+id/albumArtImageView"
android:layout_width="292dp"
android:layout_height="301dp"
android:layout_marginTop="115dp"
android:contentDescription="Description for album art"
android:scaleType="fitXY"
android:src="@drawable/night_tempo"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<TextView
android:id="@+id/songInfoTextView"
androidx.constraintlayout.widget.ConstraintLayout
TextView

```
    <TextView
        android:id="@+id/songInfoTextView"
        android:layout_width="264dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="-336dp"
        android:gravity="center"
        android:text="Night Tempo - Arashi "
        android:textColor="@color/white"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/albumArtImageView" />

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="288dp"
        android:layout_height="33dp"
        android:layout_marginStart="49dp"
        android:layout_marginTop="333dp"
        android:layout_marginEnd="49dp"
        android:backgroundTint="#FFFFFF"
        android:foregroundTint="#FFFFFF"
        android:indeterminateTint="#FFFFFF"
        android:max="100"
        android:progress="25"
        android:progressBackgroundTint="#FFFFFF"
        android:progressTint="#FFFFFF"
        android:secondaryProgressTint="#FFFFFF"
        android:thumbTint="#FFFFFF"
        android:tickMarkTint="#FFFFFF"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/songInfoTextView" />

    <ImageButton
        android:id="@+id/voiceRecognitionButton"
roidx.constraintlayout.widget.ConstraintLayout › TextView
```


PlayerPage2_weeknd_2.kt
PlayerPage_noight_tempo.kt
InternetRadioPlayerStation1.kt
MainActivity.kt
activity_player_page.xml
112
113 <ImageButton
114 android:id="@+id/voiceRecognitionButton"
115 android:layout_width="100dp"
116 android:layout_height="100dp"
117 android:layout_marginTop="57dp"
118 android:layout_marginEnd="32dp"
119 android:background="@drawable/mic_btn"
120 android:contentDescription="Button to speak in order to change music"
121 android:src="@drawable/baseline_mic_24"
122 app:layout_constraintBottom_toBottomOf="parent"
123 app:layout_constraintEnd_toEndOf="parent"
124 app:layout_constraintTop_toTopOf="@+id/seekBar"
125 app:layout_constraintVertical_bias="0.0" />
126
127 <TextView
128 android:id="@+id/timePassedTextView"
129 android:layout_width="wrap_content"
130 android:layout_height="wrap_content"
131 android:text="0:00"
132 android:textColor="#FFFFFF"
133 android:textSize="16sp"
134 app:layout_constraintEnd_toStartOf="@+id/seekBar"
135 app:layout_constraintStart_toStartOf="parent"
136 app:layout_constraintTop_toTopOf="@+id/seekBar" />
137
138 <TextView
139 android:id="@+id/totalLengthTextView"
140 android:layout_width="wrap_content"
141 android:layout_height="wrap_content"
142 android:text="0:00"
143 android:textColor="#FFFFFF"
144 android:textSize="16sp"
145 app:layout_constraintEnd_toEndOf="parent"
146 app:layout_constraintStart_toEndOf="@+id/seekBar"
147 app:layout_constraintTop_toTopOf="@+id/seekBar" />
148
149 <ImageView

```
    <ImageView
        android:layout_width="11dp"
        android:layout_height="11dp"
        android:src="@drawable/baseline_circle_25"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.459"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.784" />

    <ImageView
        android:layout_width="11dp"
        android:layout_height="11dp"
        android:src="@drawable/baseline_circle_24"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.524"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.784" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Features

Gestures

Gestures allow the user to interact with the phone without particularly looking at the phone. The gestures basically make the layout of the phone unimportant since you just use large swiping motions and tapping without an exact location on the phone. The problem was letting the user know that the gestures are actually there. At first I made two arrows next to the album cover but It didn't really look good so I changed it to dots underneath the album cover. These show the exact number of songs which you can skip and how often you can swipe. The arrows did not offer this much insight.

Vocal Recognition

What gestures lack in the amount of options to choose from vocal recognition makes up for it. In case the user wants to do something other than simply skipping, reverting or pausing. Such as selecting specific songs they could in theory with the use of our vocal recognition but its not really functional the way it is right now it just detects what ur saying in this version. In a newer version we would have liked to make it functional.

Simple UI

We decided to go with the most minimal UI we could with big icons and high contrast text. White on black is intended to make it as minimally distracting as possible while still offering great visibility and clarity on the features the app has.

Buttons (scrapped)

A main feature we wanted to implement was the use of a physical button like one that would be coded with arduino. Later we realized this was too high of an expectation. We tried to go with another idea of implementing the button idea by making the headphone buttons able to skip when double tapped or pause when tapped once. I tried to do this but in the end I wasn't able to make it feasible. A button would have helped solve this problem by eliminating the need for the user to interact with the screen at all. It would have been the best solution I believe.

Results

Successes and Learnings

What I learned was how to play music into Kotlin android studio, how to make a proper text to speech in it, how to make gestures and generally how to properly make a layout in Kotlin since it was really hard for me at first to understand object oriented programming since I'm used to js html and css this was really different from all of those things. What I also learned is having to start earlier with coding since this was a whole new language. My planning was 1 week of empathizing, 1 week of defining, 1 week of prototyping and 1 week of coding while it might have been better to actually do 2 weeks of coding since it was all so new to me.

Conclusion

The Car Audio project demonstrated a commitment to creating a safer and more user-friendly audio experience for drivers. Despite certain challenges and scope limitations, the project successfully incorporated innovative features and positive results in reducing distractions during driving. The lessons learned from this project provide a solid foundation for future endeavors in mobile app development, emphasizing the importance of user-centric design, technological exploration, and iterative refinement. I believe this project will have helped me for my internship.

Tech case

I had a lot of difficulty finding any tutorial about this techcase and I feel like I didn't do it properly because you still have to do a lot of things manually like actually switching the camera. It's stated in the assignment that both cameras have to take it immediately after each other or simultaneously which I did not manage to do. I looked at Camera2Helper, CameraX but ended up using the normal camera API which was deprecated but it still worked. I first used a youtube tutorial how to make a simple camera and then I got help from chat gpt to display two images at the same time.