CS683 Project Assignment

Guitarist

Shajee Ur Rehman

1. Overview

As someone that has played guitar for the most part of my life, I often taught my friends or family members upon request, and I tended to be quite good at it. I always felt that it would be extremely beneficial if I could teach multiple people ‘simultaneously’ through an app – thus requiring less physical work: the primary goal of most apps and technology.

My app, which will be called ‘Guitarist’ (subject to change) is for those who wish to learn to play the guitar as complete beginners, and only wish to play it purely as a hobby and not necessarily at a professional level. This means that it will not go into detail about musical theory (which not everyone wants to learn) and instead only focus on the essentials. However, given that it teaches beginner, intermediate and advanced techniques; it is also beneficial for those that already play but wish to further hone their skills.

This way the user gets up to speed with playing the guitar with exceptional clarity and have mastered all chords and certain songs that I have selected that serve as a benchmark for your current progress.

The reason I chose this project is that playing the guitar is my hobby and I found it to be an exciting prospect to create an app related to it; one that is not too complicates as I am new to the Software Development/Mobile Development field.

Furthermore, there are hundreds of thousands of people across the world that have had a fleeting thought of wanting to learn the guitar but do not have access to a good teacher that is convenient and not costly either. At the end of the day this app gives me a potential revenue stream as well.

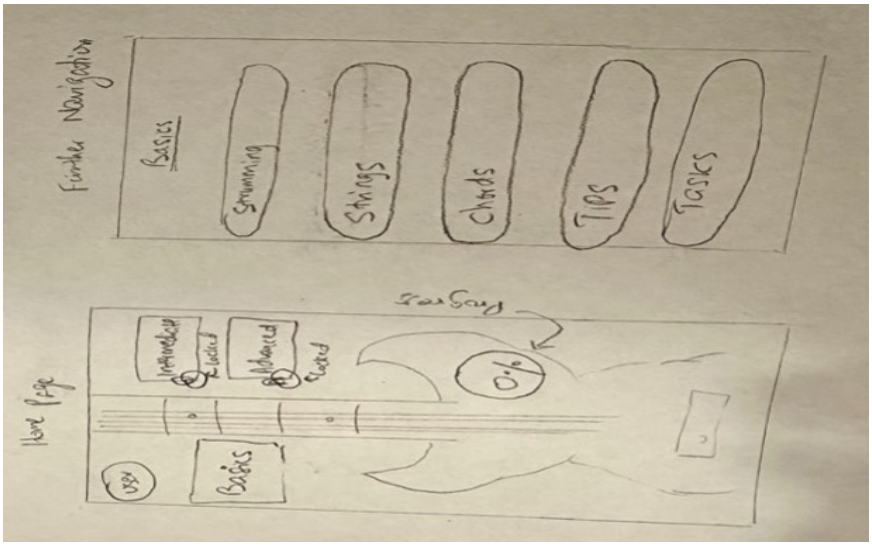
2. Related Work

There are certainly many apps out there that teach the user how to play the guitar for example: Yousician, Justin Guitar, Simply Guitar etc. but I find my method of teaching the guitar to be quite unique and effective as said by those that have experienced it and compared it with other resources. My app is similar to the aforementioned apps in that it will naturally cover overlapping material, contain similar media (audio, video, images etc.) and have a music-centered UI and overall design. However, it will differ from other apps in the manner with which I approach the teaching and subsequently the way in which the media, the layout, the focus and objectives are presented for the user to absorb and learn effectively.

Therefore, I expect to implement key ideas/functionality that separate my approach from that of other guitar teachers. This will include comprehensive explanations of improving your technique, the approach to strumming clearly, holding the pick etc.

From a features perspective: I am looking forward to implementing the overall design, User Interface, playable audio files, short video clips, images, diagrams, tracking progress etc. among other features I think of along the way.

3. Requirement Analysis and Testing

Title	User Interface / Home Screen / Navigation (Essential)
Description	As a Product Owner I want to plan the overall look such that it is appealing to the user
Mockups	
Acceptance tests	When the user opens the app: the main home screen and subsequent screens must follow a guitar-focused theme while being easy to navigate and look appealing.
Test Results	The User Interface has been tested by rotating, navigating without any lags or issue
Status	Completed in Iteration 1

Title	Accessible Media (Essential)
Description	As a Developer I need to incorporate audio, video and image files so that the user can audio visually benefit from my app while making it more attractive.
Mockups	The Screenshots have been shared below.
Acceptance tests	When the user clicks on the relevant media: it must display correctly, in all orientations, devices, screen dimensions.
Test Results	The screen does not switch to landscape for this app, media played despite rotating. So it is working well. I was able to seek through the videos without any problems.
Status	Media was added in Iterations 1 and 2, and then completed in the Final Iteration.

Title	Text and Diagram Explanations (Essential)
Description	As a Developer I want to include text and diagrams for the user to gain understanding
Mockups	The screenshots have been provided below
Acceptance tests	When the user opens a guide: the text should be legible, neat, appealing, no errors. The diagrams should display correctly and sit within the text as desired.
Test Results	ScrollView used for this. The text and image displays with good alignment and legibility.
Status	This was completed in iteration 2 with additional content added until Final Iteration.

Title	Track Progress (Desirable)
Description	As a Developer I want to code different stages (basic, intermediate, advanced) that can only be accessed after each stage is completed.
Mockups	This element may not be included
Acceptance tests	When the user completes a stage, the progress tracker should show the percentage of their journey correctly.
Test Results	This will be performed at a later stage.
Status	Was not implemented since the user is free learn subsequent guides or not.

Title	Social Element (Optional)
Description	As a user, I want a social element that allows me to connect with other guitar players and potentially play music together. Also to share my progress and skills on social media.
Mockups	Not Implemented.
Acceptance tests	The app should be able to correctly utilize network to link different users with one another and also link with social media platforms seamlessly.
Test Results	This will be performed at a later stage.
Status	Was not implemented

4. Design and Implementation

- *For this project I will be following the MVC architecture.*

- *Since the concept of this app is to contain teaching guides within different tabs and these guides are all separate from each other; each tab will be created using a new Activity and not contain fragments. This is because each Activity has the same structure in terms of a ScrollView and its Text and Media content within it*

- *The onclicklistener was used with intents in order to navigate between each activity (opening each tab and its resulting content)*

- *For the UI, I plan to follow a guitar-based theme and have in-app buttons, widgets in the shape of musical notes.*

- *There will possibly be internet connectivity if the social element is implemented (this was not implemented)*

- *I seek to include animations that are brief, relevant and follow the overall theme where necessary to add to the look of the app (videos and diagrams were implemented).*

- *Since the app does not need to store much information from the user's end (unless the social element is implemented) it does not need to include building new lists and storing data and then performing calculations or manipulating it. However, the progress tracker, if implemented, will utilize the number of tasks complete and apply basic arithmetic to determine current progress of the user towards completion.*

- *Currently there is no algorithm involved. However, I may attempt to create an algorithm that can gauge a user's progress through sound they play into a recording and the app determines if they have completed the task or not by comparing it to a benchmark audio. This will be tricky because it will have to rely on variations in the users playing speed, the tuning of their guitar etc. This was too complicated to be implemented during the timeframe of the project but it is definitely something I will look into for the future.*

- *The mediacontroller was used to seek through videos – which is a very good addition for user experience. The videos begin on start of the activity, unless there is a second video, that has to be started by clicking it and pressing play in order to prevent playback overlap of both videos. I tried to use tools to shrink my video sizes for an even faster app experience but that did not work out – ideally the videos should be smaller in size.*

5. Project Structure

Final Iteration:

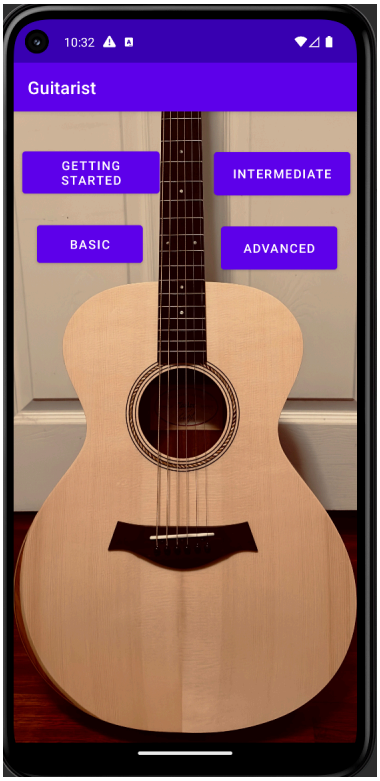
My Guitar teaching app begins with 4 tabs: “Getting Started”, “Basic”, “Intermediate” and “Advanced”. These tabs navigate to their respective screens and contain content that has been included relevant to these categories.

I have completed the UI and overall look of all the screens that have further content. On the Main Screen, “Basic” and “Getting Started” are clickable and will navigate to their respective screens.

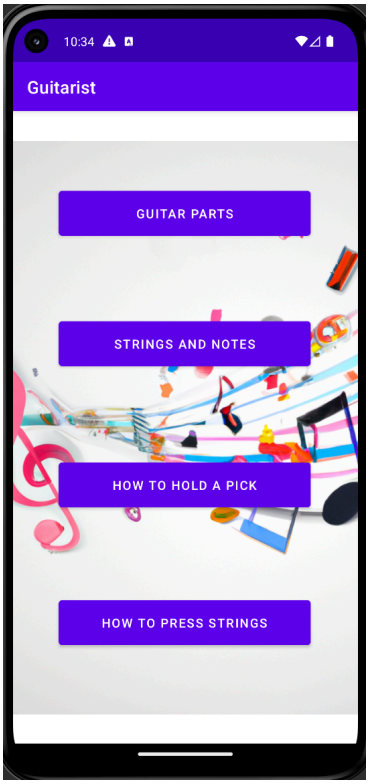
The screen “Getting Started” has got content within each of the tabs. They are clickable and will display their own content which has been created using a new activity. The content within the tabs has a Scroll View that allows the user to scroll through the same screen containing content in the form of Text and/or a Video or an Image. In this case, I have implemented videos in the content within the “Getting Started” screen, and these videos can also be paused, rewind and forwarded as well. They sit well in between all the Text and are well formatted into the layout.

The “Intermediate” and “Advanced” tabs are not clickable. They will not be clickable anyway because the purpose of this project is to demonstrate the implementation of techniques learned during this course: and those will be showcased within the “Getting Started” and “Basic” sections of the app. This is because adding content and data to “Intermediate” and “Advanced” would just be more of the same demonstration.

Below are screenshots of the look of the app:

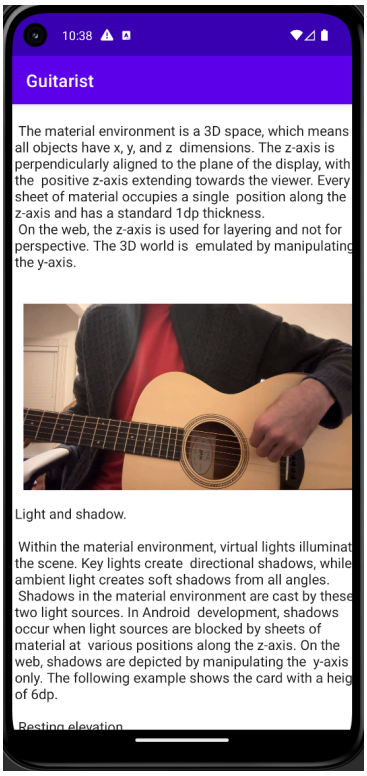


Main Screen



Getting Started

Guitar Parts



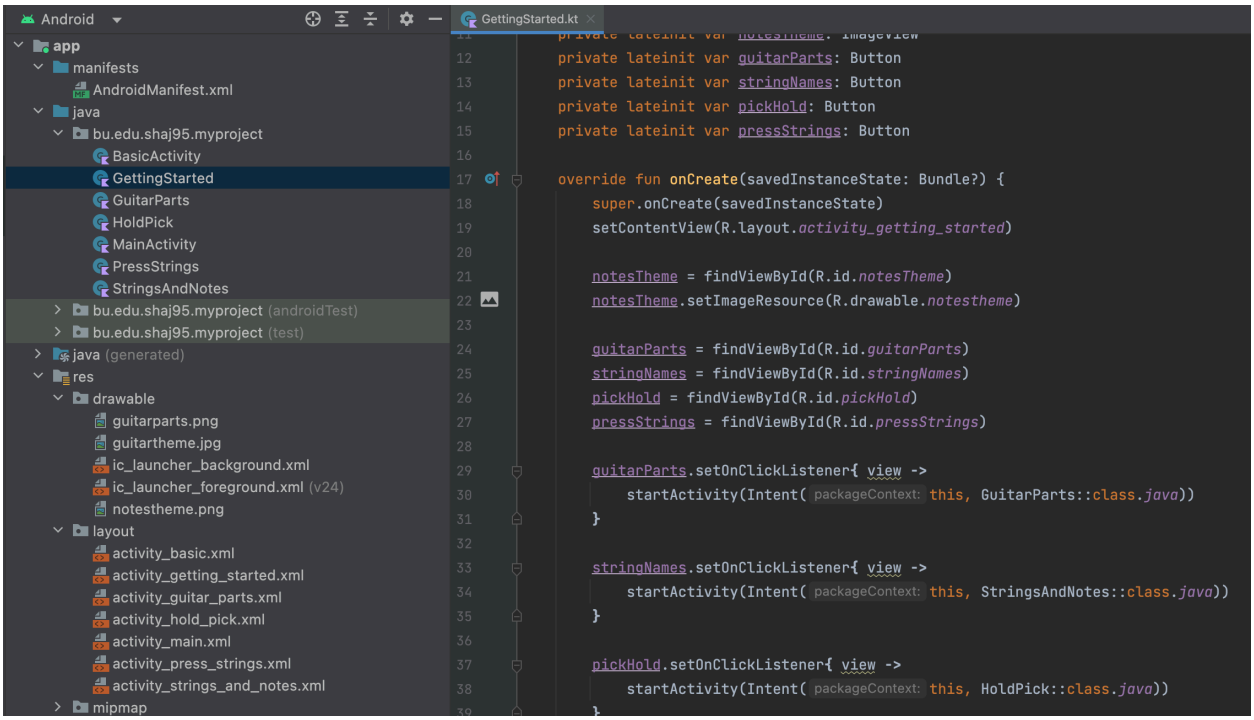
String and Notes

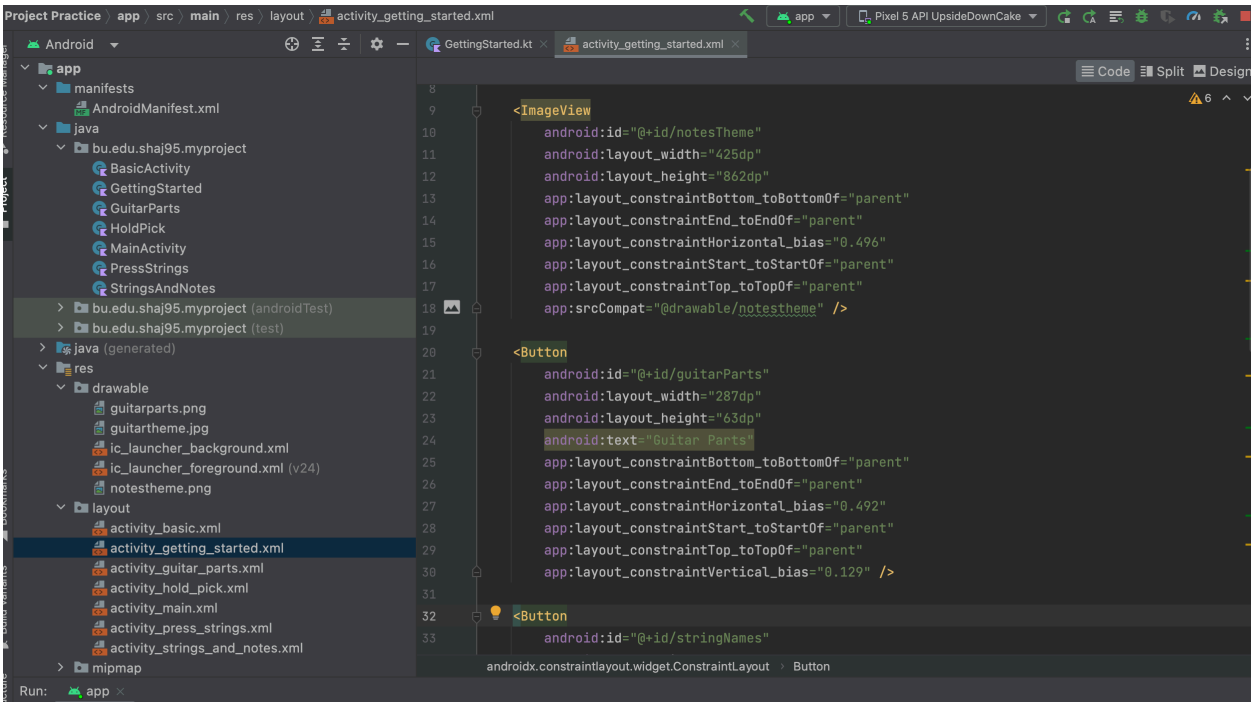
The size and alignment of the tabs on the “Getting Started” and “Basic” screens is more consistent and neater.

Code:

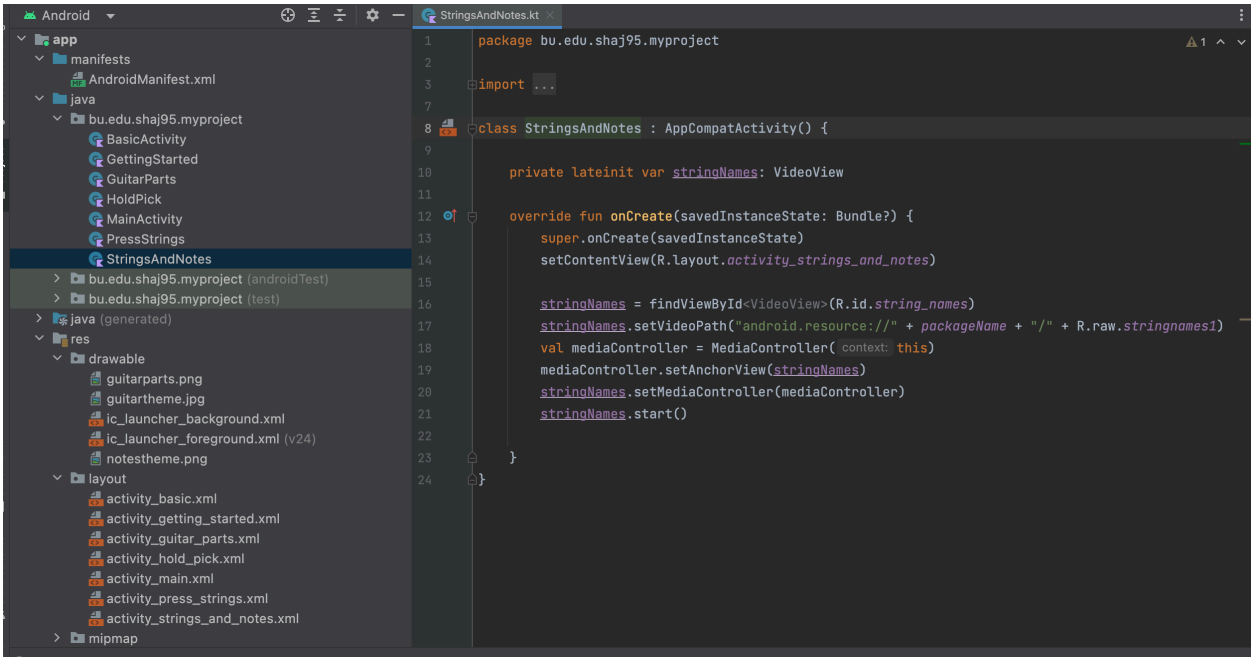
I have used different activities to implement each screen and setup the OnClickListener and an intent pointing to the respective activity in order to navigate between pages. Multiple kotlin Activity files and their respective layout xml have been setup, some containing further intents and listeners. The video and image files have been coded in appropriately based on their saved locations in the directory. This required saving the images in the ‘res/drawable’ folder and videos in the ‘res/raw’ folder. These were then given id in the xml layout files and linked on the activity file in order to compile correctly.

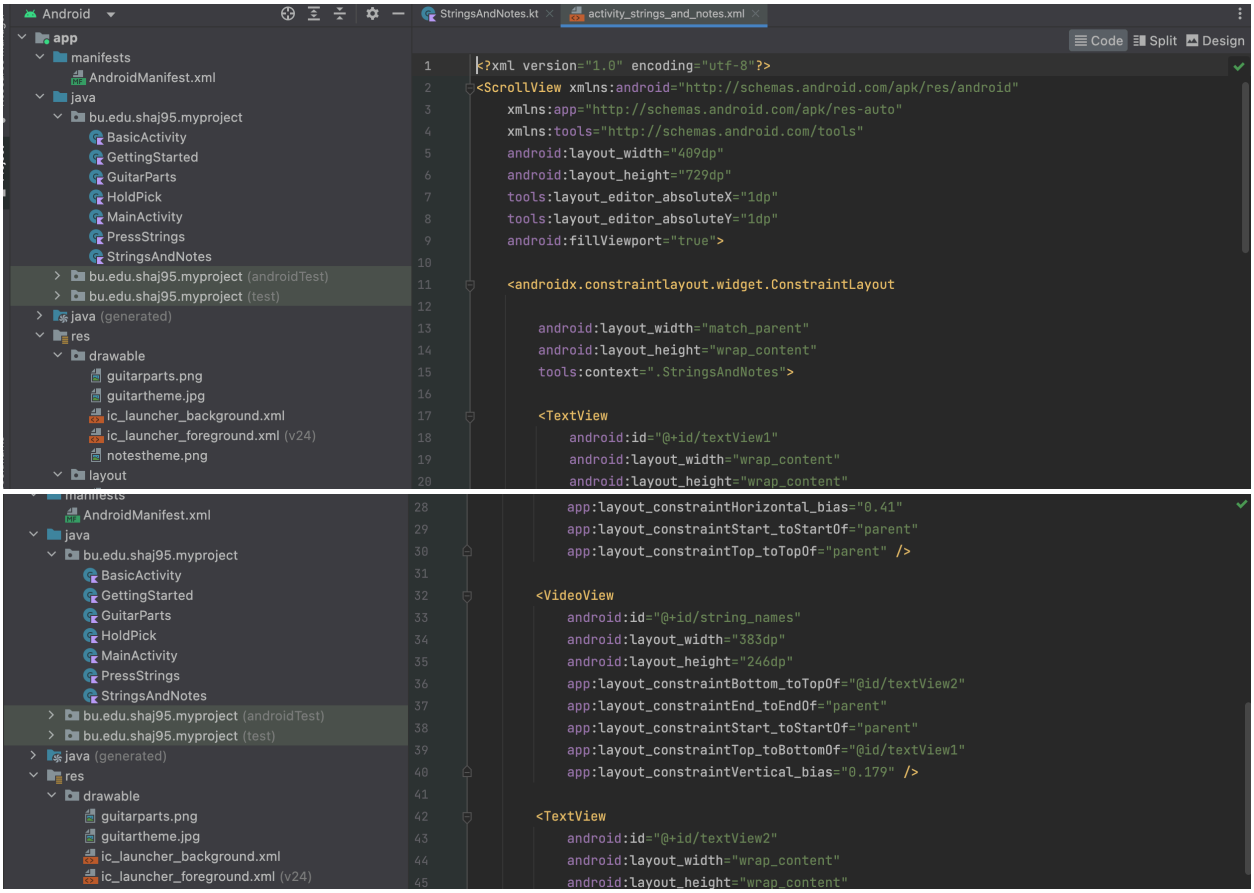
Here is a screenshot of the “Getting Started” Activity and its xml file:


Android
app
manifests
AndroidManifest.xml
java
bu.edu.shaj95.myproject
BasicActivity
GettingStarted
GuitarParts
HoldPick
MainActivity
PressStrings
StringsAndNotes
bu.edu.shaj95.myproject (androidTest)
bu.edu.shaj95.myproject (test)
java (generated)
res
drawable
guitarparts.png
guitartheme.jpg
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
notestheme.png
layout
activity_basic.xml
activity_getting_started.xml
activity_guitar_parts.xml
activity_hold_pick.xml
activity_main.xml
activity_press_strings.xml
activity_strings_and_notes.xml
mipmap
GettingStarted.kt
private lateinit var guitarParts: Button
private lateinit var stringNames: Button
private lateinit var pickHold: Button
private lateinit var pressStrings: Button
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_getting_started)
notesTheme = findViewById(R.id.notesTheme)
notesTheme.setImageResource(R.drawable.notestheme)
guitarParts = findViewById(R.id.guitarParts)
stringNames = findViewById(R.id.stringNames)
pickHold = findViewById(R.id.pickHold)
pressStrings = findViewById(R.id.pressStrings)
guitarParts.setOnClickListener{ view ->
startActivity(Intent(packageContext: this, GuitarParts::class.java))
}
stringNames.setOnClickListener{ view ->
startActivity(Intent(packageContext: this, StringsAndNotes::class.java))
}
pickHold.setOnClickListener{ view ->
startActivity(Intent(packageContext: this, HoldPick::class.java))


Project Practice app src main res layout activity_getting_started.xml
app
Pixel 5 API UpsideDownCake
Android
GettingStarted.kt
activity_getting_started.xml
Code
Split
Design
<ImageView
android:id="@+id/notesTheme"
android:layout_width="425dp"
android:layout_height="862dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.496"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/notestheme" />
<Button
android:id="@+id/guitarParts"
android:layout_width="287dp"
android:layout_height="63dp"
android:text="Guitar Parts"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.492"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.129" />
<Button
android:id="@+id/stringNames"
androidx.constraintlayout.widget.ConstraintLayout
Button
Run:
app

Here is a screenshot of the “Strings and Notes” Activity and its xml file:





6. Timeline

In the first week of the project, all previously mentioned features were in the planning phase. Week 2 focussed on the UI Design. Subsequent to this, from week 3 onwards; we I produced iteration 1, 2 and 3 of the app. The table below indicates the timeline for the three iterations that were implemented from week 3 and onwards.

Iteration	Application Requirements (Essential/Desirable/Optional)	Android Components and Features executed
1	Essentials	Manifest File, Initial Activities, UI
2	Essentials	Adding Media, Content, Videos/Images
3	Desirables (not implemented)	Refining the app and adding more content

7. Future Work (Optional)

As mentioned earlier: Currently there is no algorithm involved. However, I may attempt to create an algorithm that can gauge a user's progress through sound they play into a recording and the app determines if they have completed the task or not by comparing it to a benchmark audio. This will be tricky because it will have to rely on variations in the users playing speed, the tuning of their guitar etc. This was too complicated to be implemented during the timeframe of the project but it is definitely something I will look into for the future.

I would also try to shrink the video file sizes to make the app require less storage space.

8. Project Demo Links

I have uploaded a detailed video presentation to YouTube as I am not familiar with other tools and also used YouTube for a video presentation of another course in a previous semester. This is because at times videos can get too large to upload in some platforms so I used YouTube again due to familiarity. The following is a link to my video presentation: https://youtu.be/52mwOwDjyQc

9. References

None used since the app concept is purely my own and my own content and write up