**Tech Talk Overview**

**Grant Saylor, Kyle Smith, Anthony Tran, Jiayi Xu**

**11/6/2020**

**Topic: Android Studio**

***Total Runtime: 15-20 minutes***

* **Subtopics: GitHub integration, Firebase Integration, APIs, Kotlin Visual Editor**

**Presenters:**

* ***Anthony: Kotlin Visual Editor***
* ***Grant: GitHub Integration***
* ***Kyle: APIs***
* ***Jiayi: Firebase Integration***

**Transcript:**

***<Grant Saylor – GitHub in Android Studio>***

**1 min – 5 mins:**

**SLIDE 1 Android Studio GitHub Integration**

**Hi everyone, I’m Grant Saylor and I’m here to talk about the integration of GitHub and Android Studio. <Click Slide>**

**Android Studio is a powerful integrated development environment that is used to write Android applications, as you’ll see in our presentation there are many integrations and features that we have found helpful for our capstone project.**

**SLIDE 2** **Git and GitHub**

**Many of us utilize GitHub to host our projects for version controlling. Typically, GitHub is accessed via Git on the command line.**

**Git is a version control system to track changes in your source code during the development of your application and GitHub is where it’s all stored in the cloud for access**

**SLIDE 3 The Command Line Interface**

**As you’re going to be seeing in this video, versioning your files on the command line requires a sometime arcane knowledge to invoke a specific task. <Play video> The command line is fine and perfectly usable and utilizing the command line for Git is knowledge that is you should know.**

**SLIDE 4 The Graphical User Interface**

**In this next video you’ll see a graphical user interface of the same function. *<show video of GitHub app>* There can be times, such as when a project is large and developed by many where a graphical user interface is preferable, such as the GitHub app.**

**This does all of the same functions as the command line just with a user interface. It is particularly good for avoiding large scale disasters due to it all being presented in an easy to manage form.**

**SLIDE 5 The Power of GitHub and Android Studio Together**

**I’d like to highlight how Android Studio pulls the power of GitHub and integrates it directly in to the IDE. For our project, *Virtual Library,* we are utilizing the Android Studio IDE as our development environment.**

**Android Studio has the ability to utilize GitHub from within the IDE in the form of a graphical user interface, this allows quick access and keeps a constant workflow. <Show video>**

**You can see here that the actions performed are very similar to the GitHub desktop app, just within the IDE. The Android Studio toolbar allows access to every Git command, from pushing to pulling and more.**

**SLIDE 6 What’s the Catch?**

**The great thing about Android Studio is that it’s a free tool and so are its version controlling features.**

**Some IDEs may require a higher paid tier account to gain access to certain features, like version controlling, but with Android Studio it is included in the box.**

**To utilize the feature, you will just need to create GitHub account and connect your local repository to the remote repo.**

**The great thing about GitHub is that it can also be free, you dont need to pay extra for it unless you want some extra features.**

**SLIDE 7 Some Additional Features**

**Speaking of extra features,** **Android Studio allows you to do more than just pushing and pulling. In addition to push/pull, you can also shelve changes, refresh the status of files, browse the local history of the files.**

**All these features together allow you to have a continuous workflow without breaking up the work by switching to and from a command line or outside GUI. Pictured in this screenshot is how to access all of your VCS from within the toolbar in Android Studio.**

**SLIDE 8 Version Controlling In General**

**These version control systems are included in the IDE for a reason, it’s extremely important to the development of your application.**

**As you can see in this diagram, the local files in my file system are then marked by the Android Studio VCS as files to be added to my Git repository, by then committing and pushing it will show up in my remote repository in the cloud.**

**From there my groupmates can access these files and pull them to their local system.**

**We’ve found that with our newly remote work habits due to COVID-19, having a remote repository is vital to the development of our capstone project, without it, sharing our source code would be very difficult, and really this is going to apply in the real world as well.**

**SLIDE 9 How Android Studio Makes VCS Easier**

**Android Studio really does make VCS easier. As you can see in these two mini demos, on the left, with Android Studio there is never a break from the IDE and everything is handled in less than a minute.**

**On the right, without using Android Studio there is a much longer process that requires a lot of typing, window switching and overall distraction from your project.**

**The integration in the IDE itself is very helpful, it’s a feature that Android Studio handles very well.**

**SLIDE 10** **Thank you!**

**Thank you! Really it comes down to that Android Studio provides an elegant solution to version controlling by having it all easily accessible in the IDE. Thank you and I’ll hand it over to my groupmate!**

***<Student Name and Topic>***

**5 mins – 10 mins:**

***<Anthony Tran – Using Android Studio’s Layout Editor>***

**10 mins – 15 mins:**

***Slide 21:***

Hi my name is Anthony and I’m going to talk about Using Android studio’s layout editor.

***Slide 22:***

Android Studio allows programmers to create applications using pre-existing UI elements that comes along with the code that makes up that element. This feature is called Layout Editor

The Layout Editor in Android studio allows users to simultaneously work on Kotlin code while also using the Layout Editor’s convenient interface to help create an application more efficiently

Anything that is changed through the Layout Editor will have the appropriate code added/changed in the Code section of Android Studio

***Slide 23:***

Here is the interface for the Layout editor. There’s a lot going on so I’m going to only focus on a few topics.

First is the Palette. The palettes allows the user to click and drag elements of an android app onto the layout editor rather than to manually code it out. There are elements such as text boxes, buttons and other widgets you can work with.

Next is the Attributes section. The attributes sections allows the user to change many aspects of the android app such as widget size, font size, font color, and other cosmetic details. We’ll come back to this

Lastly, I would like to talk about the view modes. Right now we are in the Layout editor mode. However, the user can change this view.

***Slide 24:***

Here is the dual view mode that allows the user to see both the visual editor AND the actual code as well.

The reason why layout editor is useful for our project is because it allows us to visually see our application while coding and real time.

***Slide 25:***

Like I said before, you can manually change variables of elements in the attribute section. When you Alter parameters in the attribute section <next slide>

***Slides 26:***

It will alter the coded section right away. This is also true when you alter code, the layout editor will change right away as well.

***<Jiayi Xu – Connect Firebase to Android Studio>***

**Slide 27:**

This is Jiayi Xu, and I am here to talk about the Firebase. Firebase is an application development platform designed in particular for mobile app developers. It includes hosting, authentication, data driven triggers and analytics.

For our project, we will design an mobile application, so will use Firebase to manage data rather than use MangoDB.

**Slide 28:**

We can use Firebase to do different things, such as measure user activity, sign in, store and sync app data in realtime across all connected clients, and so on.

Our project will focus on realtime database. For example, allow users use their cell phone to check in or check out books.

**Slide 29:**

To use Firebase, we need to connect to Android Studio. Before connect, few things need to do. Check the version of the Android Studio and must meet these requirements. “build.gradle,” “gradle.wrapper.propities,” and these requirements.

The Google Play icon means the emulator may use system image that include the play store app.

**Slide 30:**

There are two ways to connect Firebase to Android Studio, I will introduce the simple method which we do not need to configure profiles.

First, sign into Firebase using Google account, and create an project for our application.

**Slide 31:**

After this step, click the project, then go to the Realtime Database section and create database. Here shows the definition of the Realtime Database, it is a NoSQL database and can execute quickly.

**Slide 32:**

Now, open the Android Studio, we can find the Firebase assistant in the toolbar, and click Realtime Database.

**Slide 33:**

Follow the steps, connect the app to Firebase, accept changes then it will configure profiles automatically on Android Studio.

**Slide 34:**

The last thing I want to talk about Configure Rules. By default, only authenticated use can read or write data, but we can alter rules from Firebase console. There have 10 rules can be used. The link shows here.

Thank you for listening to our group’s Tec-Talk.