Deployment and Operations Guide (Runbook)

From Scriber Mobile Application

Milestone 3 – Mobile Team

Software Engineering Project

SWEN 670

March 16, 2021

Version 1.0

Presented by: Sylvia Lopez-Willis, Sompon Boontho, Alex Dattilio, Bertina Lee, Brian Malott, Karim Mansour, Komi Noukafou, Joselitio Ocampo, Ermias Seyoum, Aruand Tako

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| DATE | VERSION | DESCRIPTION | AUTHOR |
| 03/16/2021 | 1.0 | Initial documents | Sylvia Lopez-Willis, Sompon Boontho, Alex Dattilio, Bertina Lee, Brian Malott,  Karim Mansour, Komi Noukafou, Joselitio Ocampo,  Ermias Seyoum,  Aruand Tako |
|  |  |  |  |

**Programmer Guide Approvals**

|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date** |
| Approved by:  "Software Lead" | Karim Mansour | 3/16/2021 |
| Approved by:  "Project Manager" | Sylvia Lopez-Willis | 3/16/2021 |
| Approved by:  “Stakeholder-Dr. Mir Assadullah” |  |  |

Table of Contents

[1 Introduction 5](#_Toc66807659)

[1.1 Purpose 5](#_Toc66807660)

[1.2 Intended audience and reading suggestions 5](#_Toc66807661)

[1.3 Technical project stakeholders 5](#_Toc66807662)

[1.4 References 5](#_Toc66807663)

[1.5 Definitions, Acronyms, and Abbreviations 6](#_Toc66807664)

[2 Mobile Application 6](#_Toc66807665)

[2.2 Features, Packages, Plugins, and Widgets 6](#_Toc66807666)

[2.2.1 Features 6](#_Toc66807667)

[2.2.2 Packages 7](#_Toc66807668)

[2.2.3 Plugins 7](#_Toc66807669)

[2.2.4 Widgets 8](#_Toc66807670)

[3 Software Installation 8](#_Toc66807671)

[3.1 Android Studio 8](#_Toc66807672)

[3.1.1 Prerequisites 8](#_Toc66807673)

[3.1.2 Installation Steps 8](#_Toc66807674)

[3.2 Flutter and Dart 18](#_Toc66807675)

[3.3 GitHub 20](#_Toc66807676)

[5 Testing the Mobile Application 34](#_Toc66807677)

[6 Troubleshooting 36](#_Toc66807678)

[6.1 Issue installing Flutter on MacOS Catalina 36](#_Toc66807679)

[6.2 Android Studio ADB not detected 36](#_Toc66807680)

[6.3   Unable to launch Android Studio AVD Emulator 36](#_Toc66807681)

[6.4 Android Studio - Gradle taking a long time 37](#_Toc66807682)

[Appendices: 37](#_Toc66807683)

[Appendix A - References 37](#_Toc66807684)

# 1 Introduction

## 1.1 Purpose

The Deployment and Operations Guide (Runbook) purpose is to outline the steps needed to deploy the mobile application and be in working order.

## 1.2 Intended audience and reading suggestions

The intended audience for this document is the stakeholders and technical departments connected to public service professionals. Those responsible for deploying the Form Scriber mobile application within their organization will be provided this guide to install all the tools involved and read through the suggested material listed in the sections.

## 1.3 Technical project stakeholders

The table below shows a list of the project stakeholders for the Form Scriber Mobile Application:

Table 1 - Project Stakeholders

|  |  |
| --- | --- |
| Name | Role |
| Dr. Mir Assadullah | Stakeholder (Project Owner) |
| Sylvia Lopez-Willis | Project Manager |
| Karim Mansour | Lead Developer/Backup PM |
| Alex Dattilio | Developer |
| Bertina Lee | Developer |
| Brian Malott | Developer |
| Komi Noukafou | Developer |
| Joselito Ocampo | Developer |
| Ermias Seyoum | Developer |
| Arnaud Tako | Developer |
| Sompon Boontho | Software Tester |

## 1.4 References

Table 2 - Referenced Documents

|  |  |
| --- | --- |
| Title | Reference |
| RaisedButton Class | https://api.flutter.dev/flutter/material/RaisedButton-class.html |
| Adding interactivity to your Flutter app | https://flutter.dev/docs/development/ui/interactive#:~:text=A%20widget%20is%20either%20stateful,A%20stateless%20widget%20never%20changes.&text=A%20widget's%20state%20is%20stored,widget's%20state%20from%20its%20appearance. |
| macOS Install | https://flutter.dev/docs/get-started/install/macos |
| Download for Mac OS | https://git-scm.com/download/mac |
| Install Flutter and Dart plugins | https://flutter.dev/docs/get-started/editor |
| Setting up Flutter on Mac OS Catalina | https://medium.com/@alexandrosbaramilis/setting-up-flutter-on-macos-catalina-d023df8845ae |
| [While launching AVD in emulator showing this “Unable to locate adb”](https://stackoverflow.com/questions/62186550/while-launching-avd-in-emulator-showing-this-unable-to-locate-adb)  [Ask Question](https://stackoverflow.com/questions/ask) | https://stackoverflow.com/questions/62186550/while-launching-avd-in-emulator-showing-this-unable-to-locate-adb%3E |

## 1.5 Definitions, Acronyms, and Abbreviations

Table 3 - Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Acronyms and Abbreviations | Definitions |
| AI | Artificial Intelligence |
| JDK | Java Development Kit |
| GB | Gigabyte |
| iOS | iPhone operating system |
| RAM | Random Access Memory |

# 2 Mobile Application

## 2.2 Features, Packages, Plugins, and Widgets

### 2.2.1 Features

* Usability - Form Scriber can be used in multiple settings where interviews take place using a mobile device.
* Efficiency and time-saving - The User Interface is designed for ease of use for logging in and tapping a button to start and stop the recordings. The sliding screen makes it easy to access the menu options. Report forms can be printed quickly.
* Invaluable - Form Scriber allows professionals to spend quality time with their clients by taking out the burden of hands-on documentation.

### 2.2.2 Packages

* FlutterFire - A package containing a flutter-compatible API for Google Firebase operations, including user authentication.
* Flutter\_auth
* Flutter\_core
* Flutter Plugins – A package containing first-party flutter plugins
* Google\_sign \_in – A plugin for creating a Google user sign-in
* Material - package:flutter/material.dart
* “dialog\_flowtter”
* “firebase\_auth”
* “get\_it”
* “googleapis”
* “googleapis\_auth
* “google\_sign\_in”

### 2.2.3 Plugins

Flutter and Dart plugins were installed in Android Studio.

For Windows and Linux install:

1. Open plugin preferences (**File > Settings > Plugins**).
2. Select **Marketplace**, select the Flutter plugin, and click **Install**.

For Mac OS install:

1. Start Android Studio.
2. Open plugin preferences (**Configure > Plugins** as of v3.6.3.0 or later).
3. Select the Flutter plugin and click **Install**.
4. Click **Yes** when prompted to install the Dart plugin.
5. Click **Restart** when prompted.

### 2.2.4 Widgets

* Landing Page consisted of two RaisedButton widgets that are a part of the Material widget. When the Login and Help buttons are pressed, the buttons are elevated. The raised button was wrapped with a Center widget. The child of the RaisedButton was given a Text widget for “Login” and “Help.” An onPressed function was added when the button is tapped.
* StatelessWidget – for text and title
* StatefulWidget – is dynamic and allows user interaction where the buttons are pressed and tasks are performed.

# 3 Software Installation

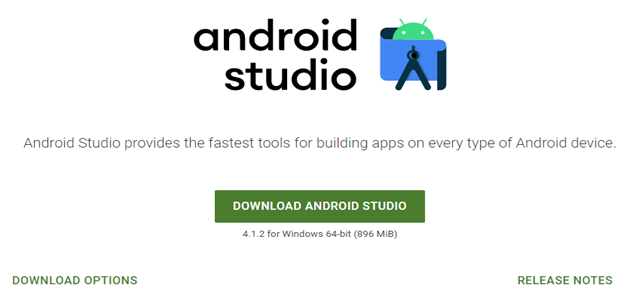
## 3.1 Android Studio

### 3.1.1 Prerequisites

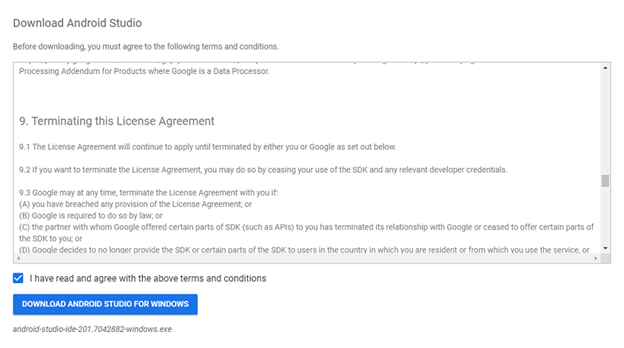
1. Java Development Kit (JDK) is already installed. If not installed, click this link: <https://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html>
2. Previous versions of Android Studio and Android SDK must be uninstalled.
3. Check System requirements: Windows 10, recommended 8GB of RAM, 4 GB of disk space, and 1200 X 800 minimum screen resolution.

### 3.1.2 Installation Steps

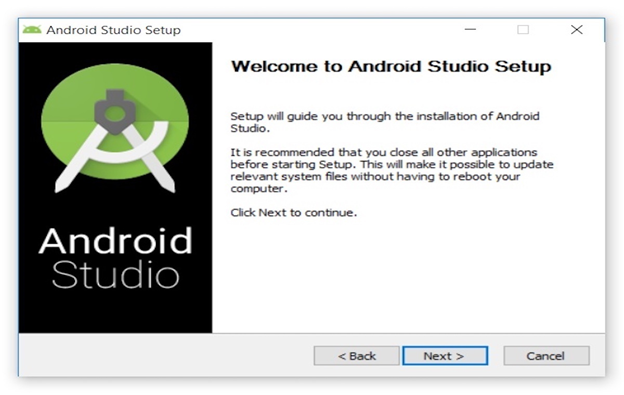
1. To install Android Studio after the required prerequisites are available in the system environment, it can be installed by clicking this link <https://developer.android.com/studio>



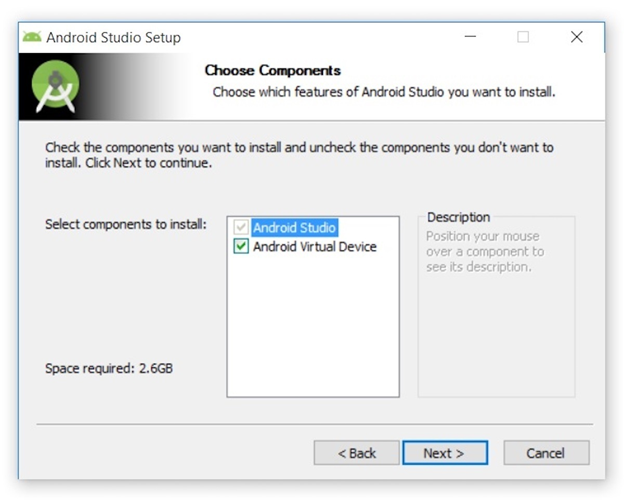
1. Click on “DOWNLOAD ANDROID STUDIO” to download and after selecting “I have read and agree with the above terms and conditions, click Download Android Studio for Windows.



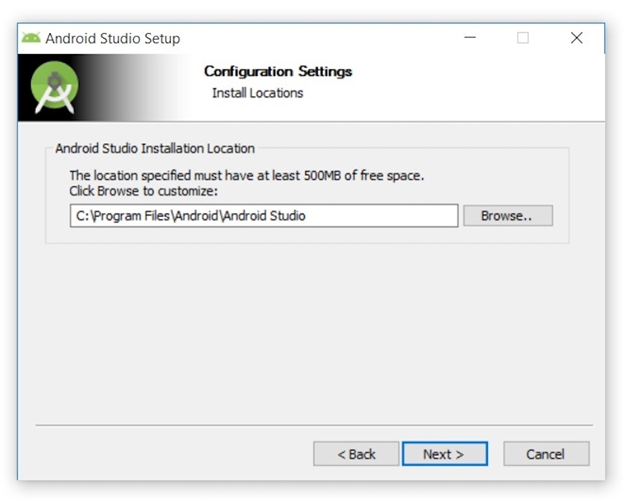
1. After the download finish, it provides this link to set up the Android Studio android-studio-ide-181.5056338-windows.exe
2. Clicking the link will open the Android Studio dialog box to setup.



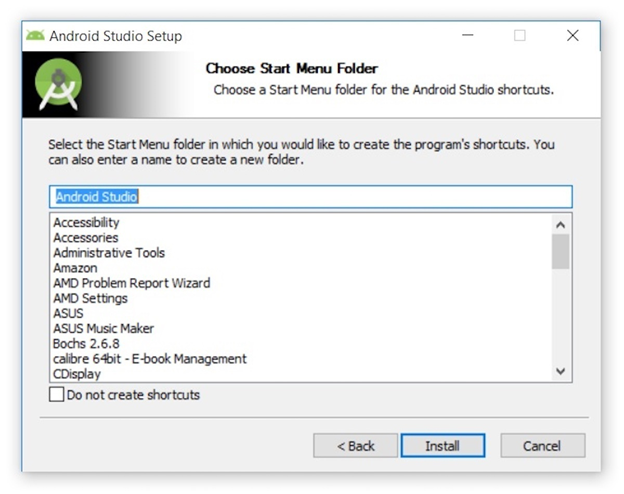
1. The dialog box provides Android Studio and Android Virtual Device components to select and install. In this step, a default setting is recommended. Click next.



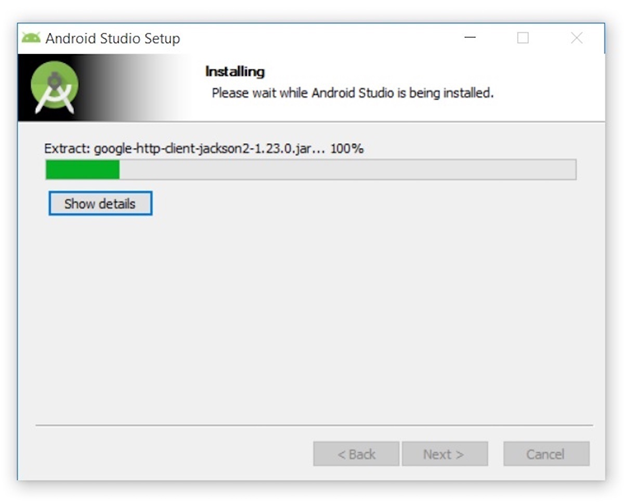
1. Browse and select keep the default or a specific location with more space to store. Click next.



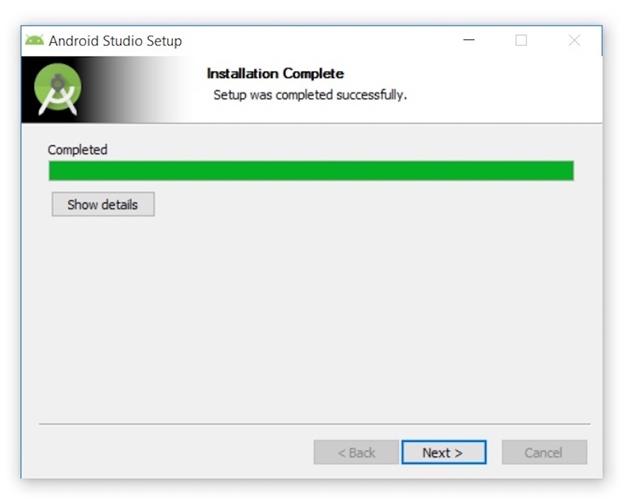
1. In this Choose Start Menu Folder, keep the default folder and click install.



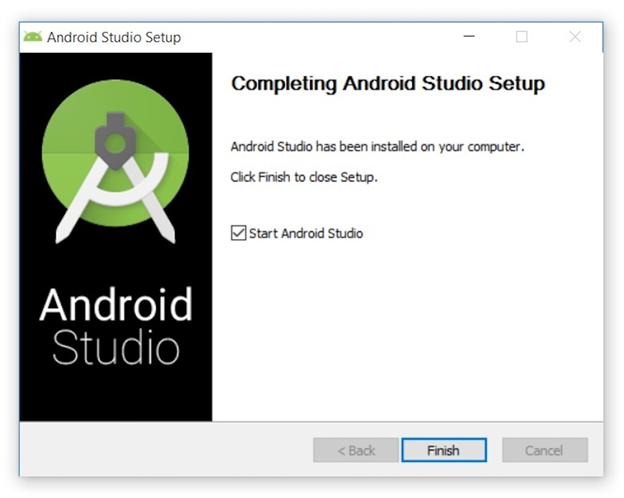
1. Now, the installing panel shows the Android Studio is being installed.



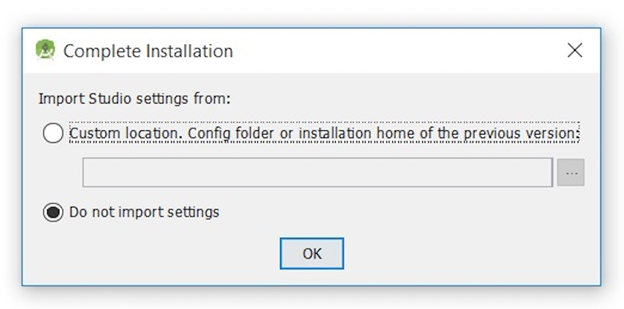
1. When the installation is completed, the dialog flow box displays a “Set was completed successfully.” Click next.



1. Clicking next will show the completed Android Studio Setup. To close and finish the installation, click finish.



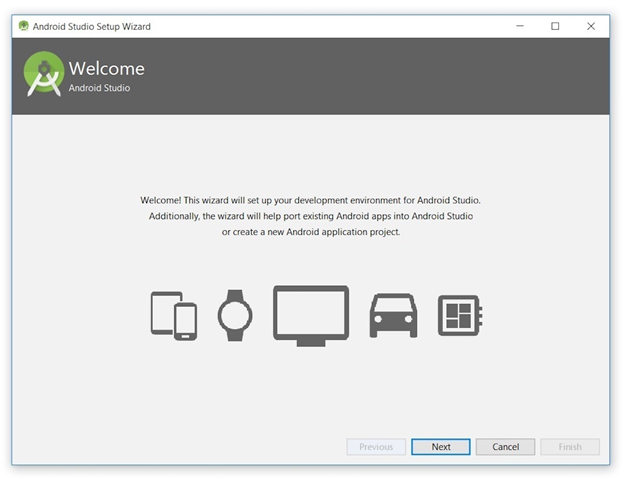
1. When running Android Studio for the first time, it provides choices to import and select “Do not import settings” and click ok.



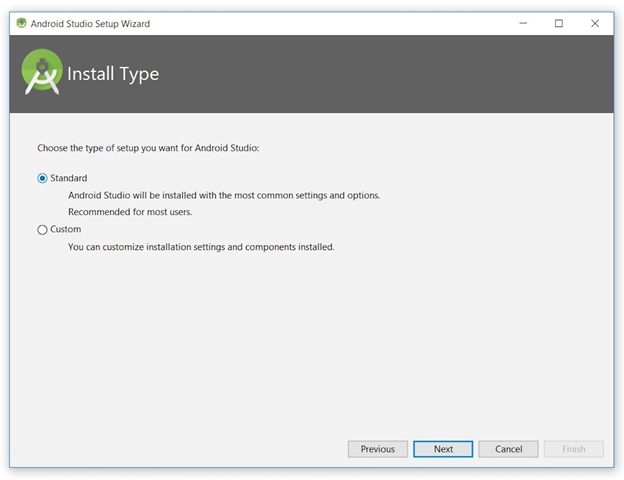
1. Android Studio opens up while downloading SDK components.



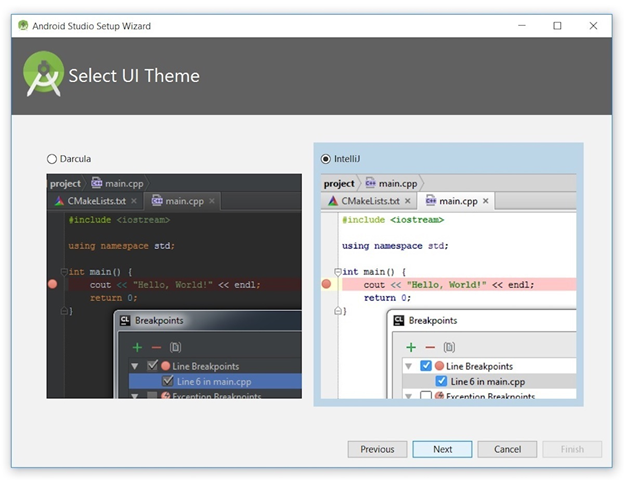
1. Android Studio provides the setup wizard to create a development environment for Android Studio. Click next.



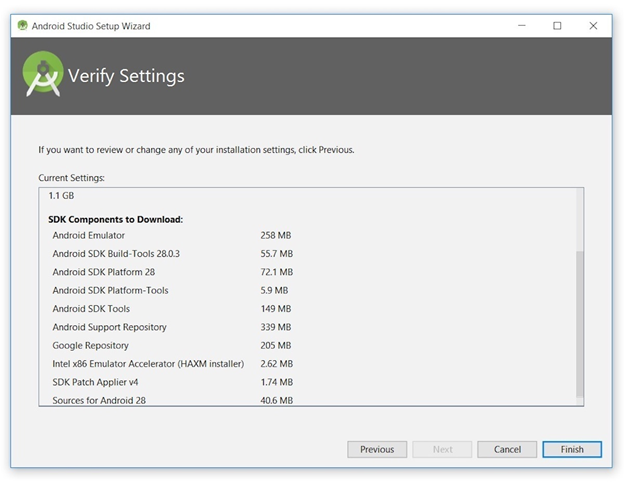
1. Android Studio Provides to select the install type and select standard, and click next.



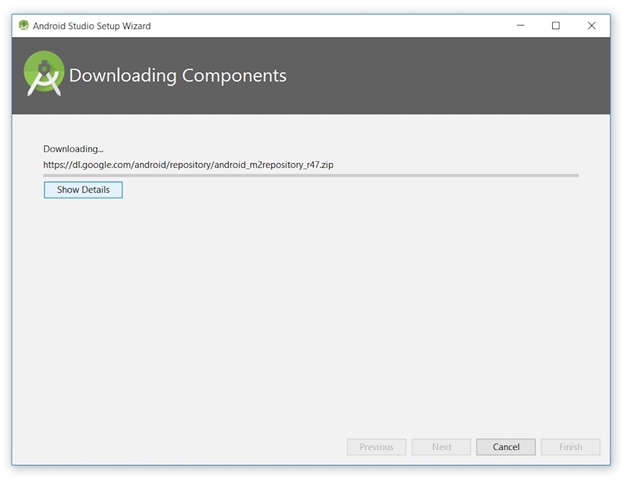
1. Android Studio provides a choice to select for UI Theme. Select IntelliJ and click next.



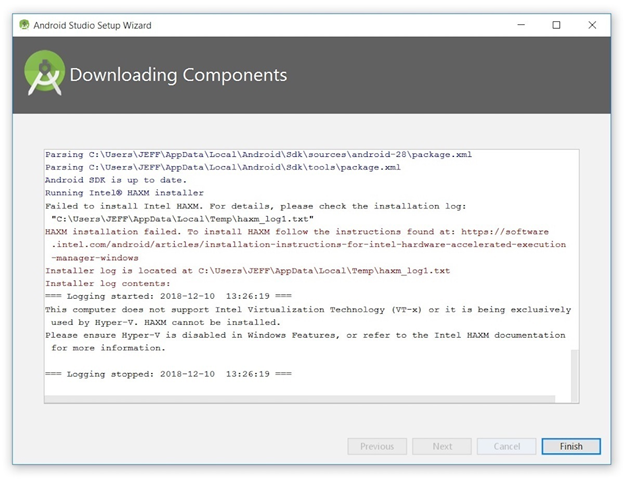
1. Android Studio provides current settings to verify. After verified, click finish.



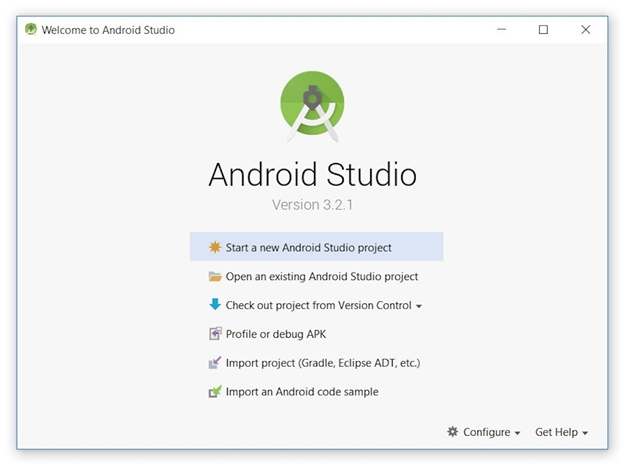
1. At this point, Android Studio starts downloading all the necessary components.



1. Downloading the necessary components continued. When the download is finished, click finish.



1. Finally, the Android Studio dialog box displays welcome to Android Studio.



## 3.2 Flutter and Dart

**Flutter System Requirements**

* Operating Systems: Windows 7 SP1 or later (64-bit), x86-64 based
* Disk Space: 1.64 GB (does not include disk space for IDE/tools).
* Tools: Flutter depends on these tools being available in your environment.
* [Windows PowerShell 5.0](https://docs.microsoft.com/en-us/powershell/scripting/install/installing-windows-powershell) or newer (this is pre-installed with Windows 10)
* [Git for Windows](https://git-scm.com/download/win) 2.x, with the Use Git from the Windows Command Prompt option.
* If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.

1. To get started installing Flutter SDK, download the link below.

<https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_2.0.2-stable.zip>

2. Extract the Flutter zip file and select a specific location in C:\ drive to store the installation.

    Example: C:\src\flutter

3. Locate the file Flutter\_cosole.bat inside the Flutter directory and start it by double-clicking.

* At this point, Flutter is ready to run Flutter commands in the Flutter console
* To update the older version of Flutter.

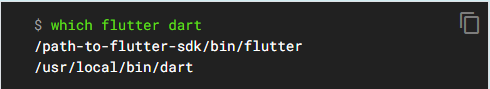
If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

* From the Start search bar, enter ‘env’ and select **Edit environment variables for your account**.
* Under **User variables** check if there is an entry called **Path**:
* If the entry exists, append the full path to flutter\bin using; as a separator from existing values.
* If the entry doesn’t exist, create a new user variable named Path with the full path to flutter\bin as its value.

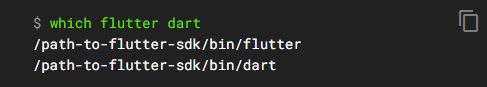
You have to close and reopen any existing console windows for these changes to take effect.

4. Flutter SDK contains Dart command-line programs. To make the Flutter and Dart version compatible, run the following command lines in Flutter console.

* This command line checks Flutter and Dart compatibilities whether they are in the same bin directory or not.



* This command line will update the Flutter and Dart directory to create the same bin directory for both.



5. To check if any dependencies are needed to set up, enter this command line below in the Flutter console.



6. Flutter and Dart installation completed.

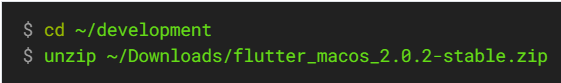
Reference: <https://flutter.dev/docs/get-started/install/windows>

**Flutter and Dart install for macOS:**

* **Operating Systems**: macOS (64-bit)
* **Disk Space**: 2.8 GB (does not include disk space for IDE/tools).
* **Tools**: Flutter uses git for installation and upgrade. Install Xcode first, and this includes git.

Get the Flutter SDK by downloading it from the macOS installation page through the Flutter website, <https://flutter.dev/docs/get-started/install/macos>. Dart SDK comes with the Flutter automatically if using Flutter 1.21 and new.

* flutter\_macos\_2.0.2-stable.zip
* Extract the file by going into the terminal and using the following command:



***Add the flutter tool to the path permanently:***

* Determine the directory where you placed the Flutter SDK. You need this in Step 3.
* Open (or create) the rc file for your shell. Typing echo $SHELL in your Terminal tells you which shell you’re using. If you’re using Bash, edit $HOME/.bash\_profile or $HOME/.bashrc. If you’re using Z shell, edit $HOME/.zshrc. If you’re using a different shell, the file path and filename will be different on your machine.
* Add the following line and change [PATH\_TO\_FLUTTER\_GIT\_DIRECTORY] to be the path where you cloned Flutter’s git repo:

*content\_copy*

$ export PATH**=**"$PATH:[PATH\_TO\_FLUTTER\_GIT\_DIRECTORY]/flutter/bin"

* Run source $HOME/.<rc file> to refresh the current window, or open a new terminal window to source the file automatically.
* Verify that the flutter/bin directory is now in your PATH by running:

*content\_copy*

$ echo $PATH

Verify that the flutter command is available by running:

*content\_copy*

$ which flutter

## 3.3 GitHub

macOS install:

* If Xcode is installed, then Mac will have git. First, check if git is installed by typing the following command into the terminal:

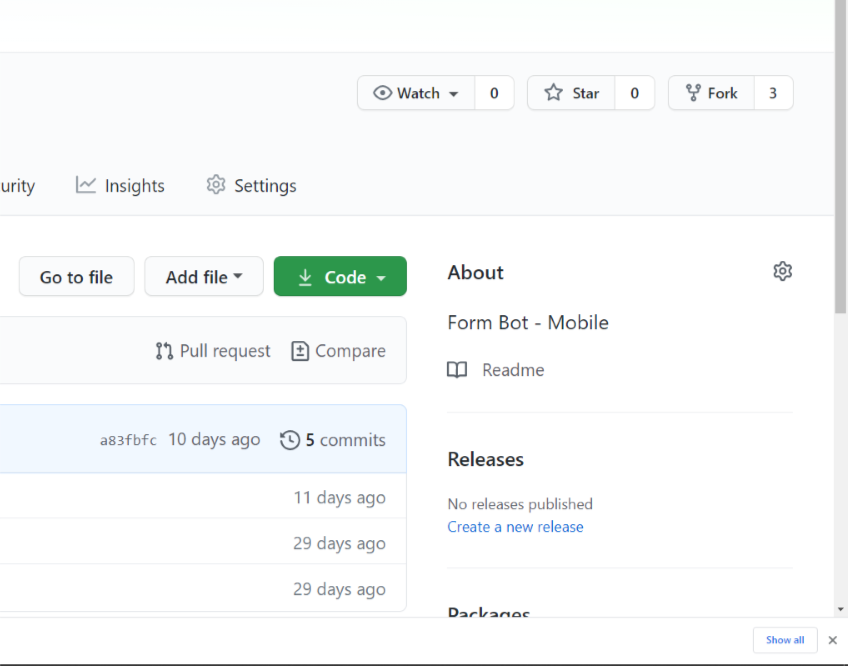


* If installed, then after getting the response in your terminal of the version, no need to install git.
* If git is not installed, then download XCode, or do the following:
* Install homebrew through the terminal type the following command:



Windows: cloning a repository using the command line

* In Github, Go to the repository main page.



* Click the Code button above the files list.

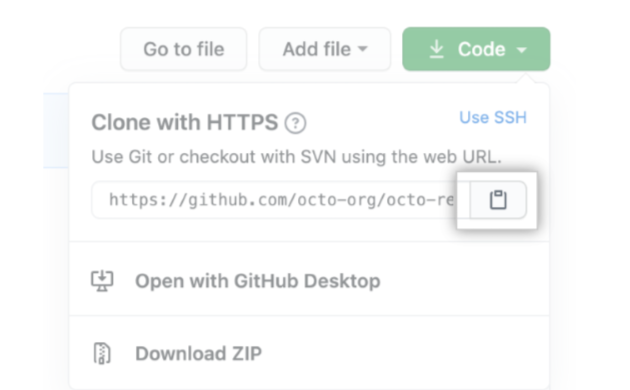


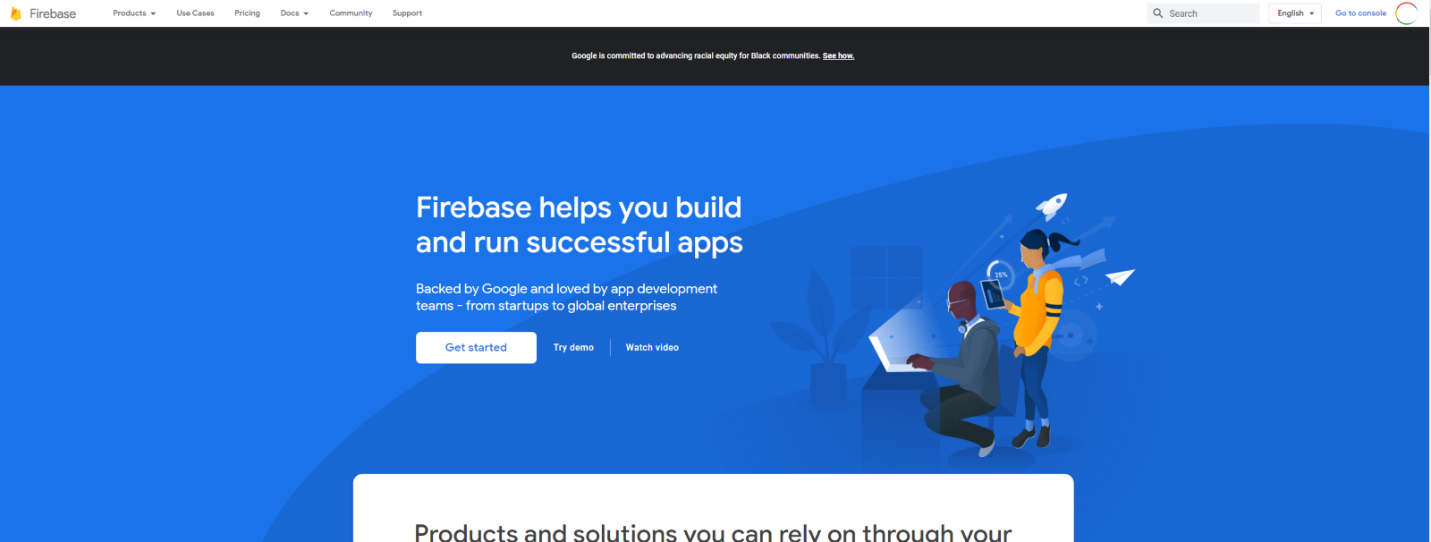
Image Source: https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository

* Click the clipboard icon to copy the link.

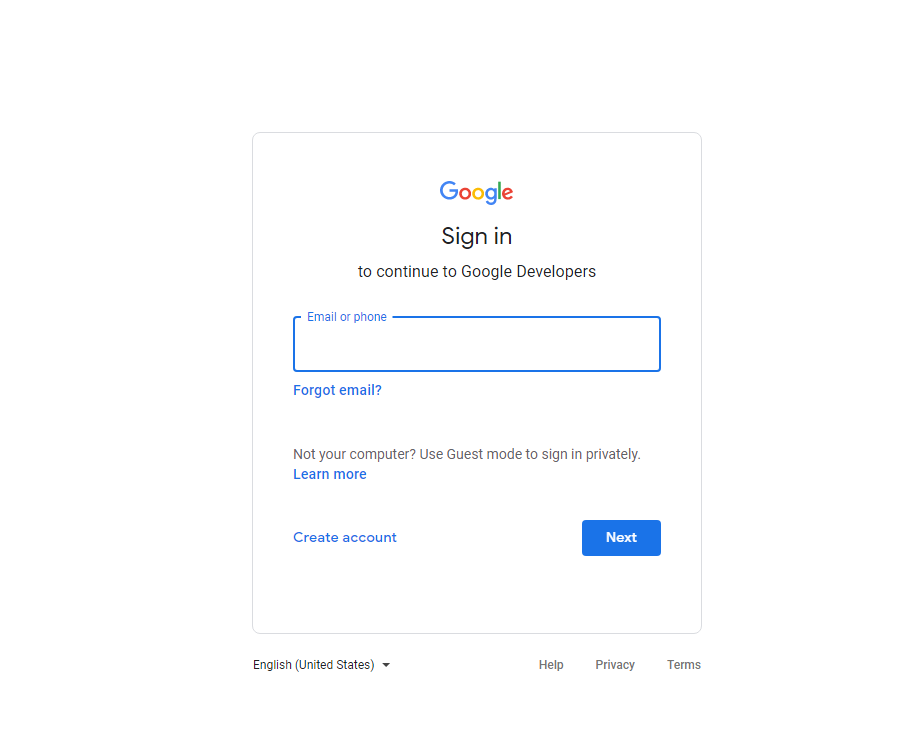
4 Prepare the Mobile Application for Use

4.1 Firebase setup

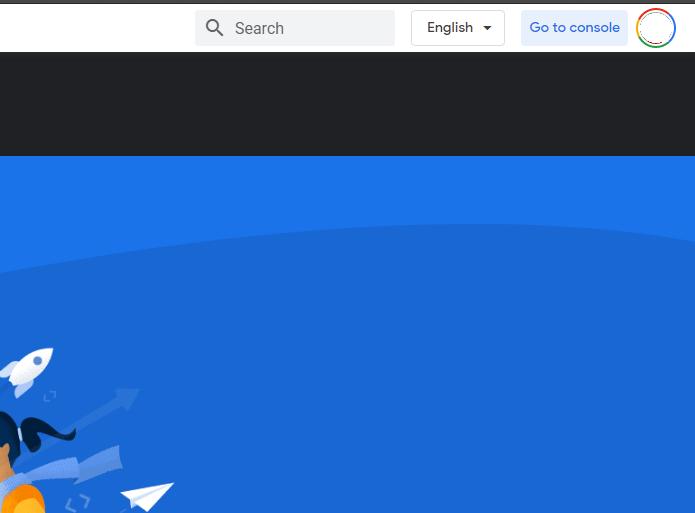
1. Open a browser and go to <https://firebase.google.com/>



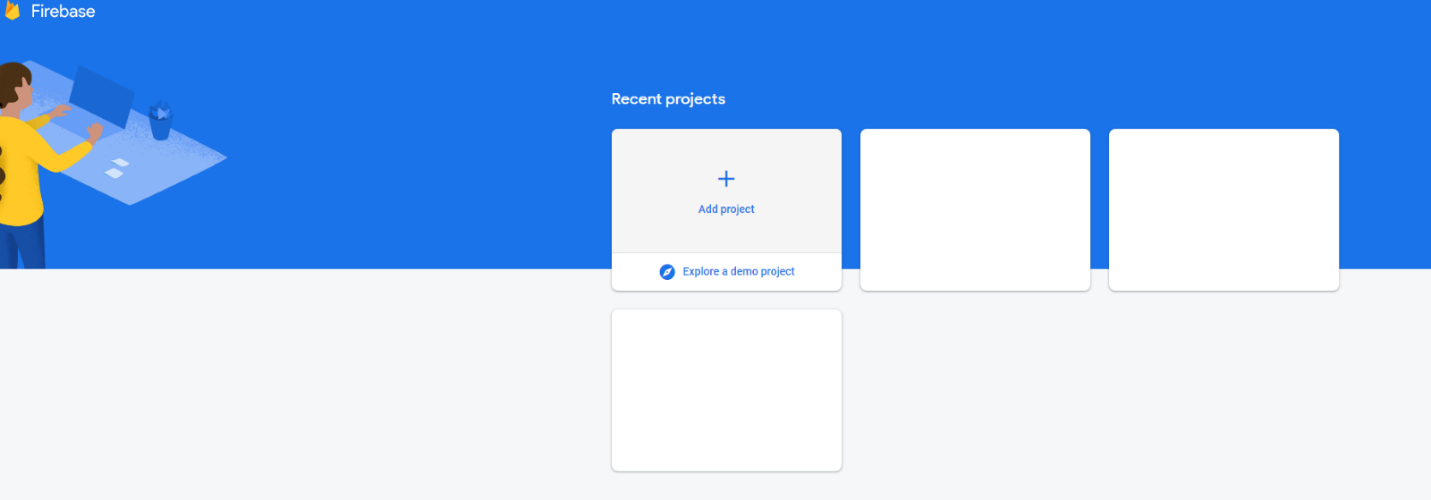
1. Click the circle in the upper right-hand corner of the screen to open the login screen. From here, enter your Google account credentials and log in.



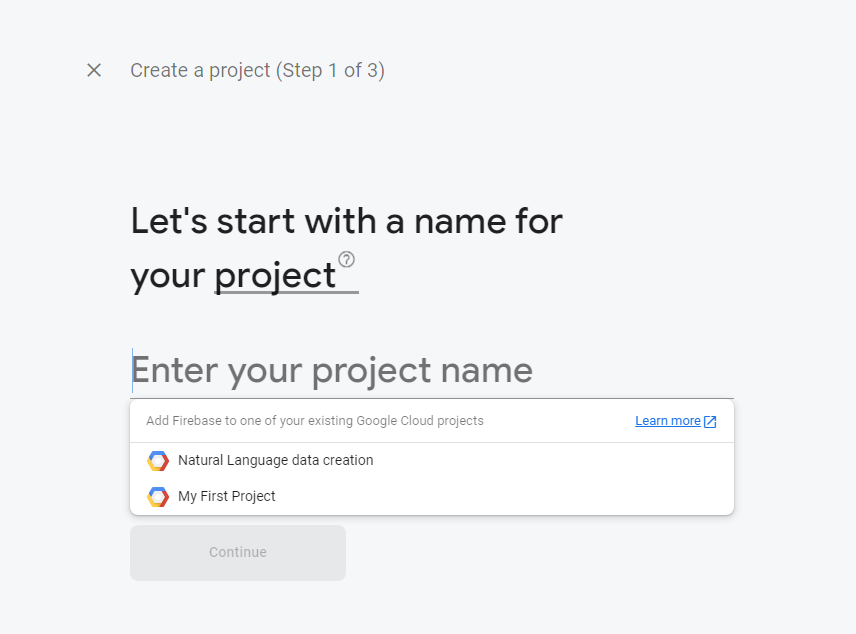
1. Once logged in, click the “Go to console” button in the top right corner of the window



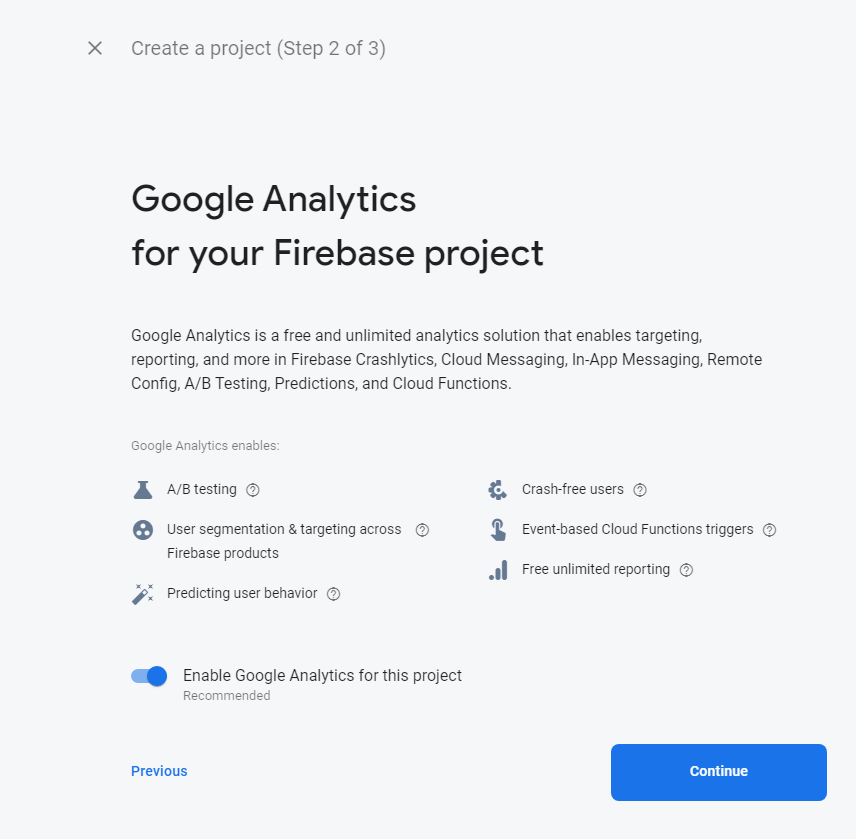
1. Click on the “Add project” button



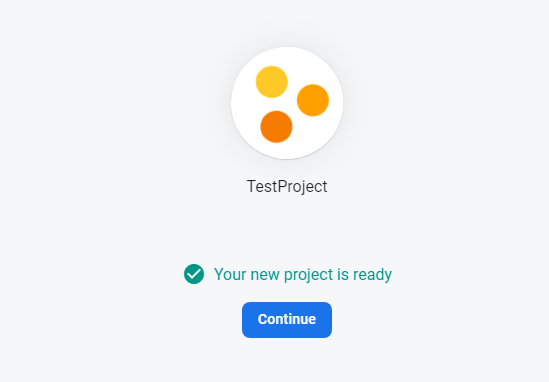
1. Enter the name of your existing Google Cloud Project



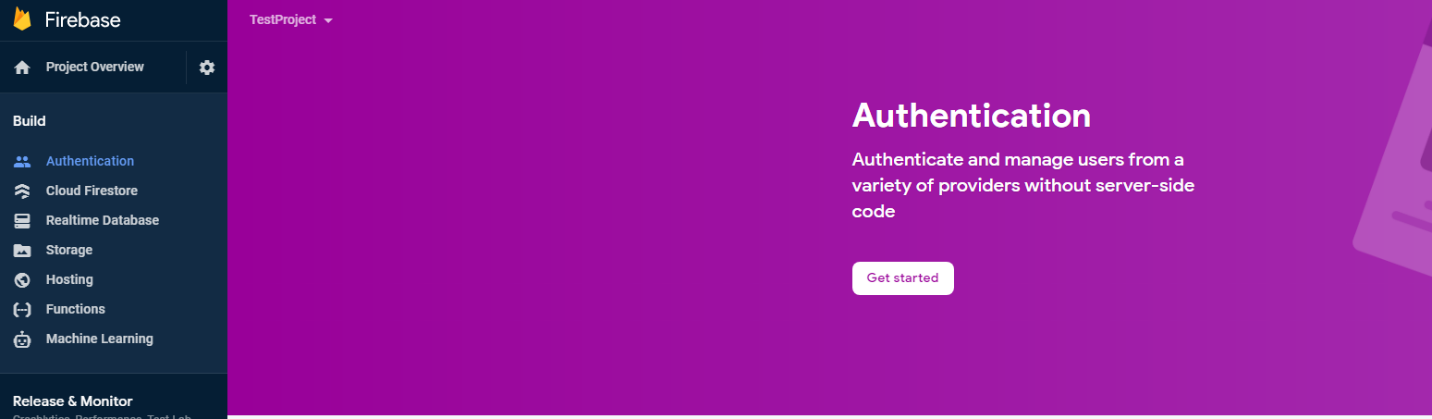
1. Choose whether or not to enable google analytics (we disabled it in this tutorial)



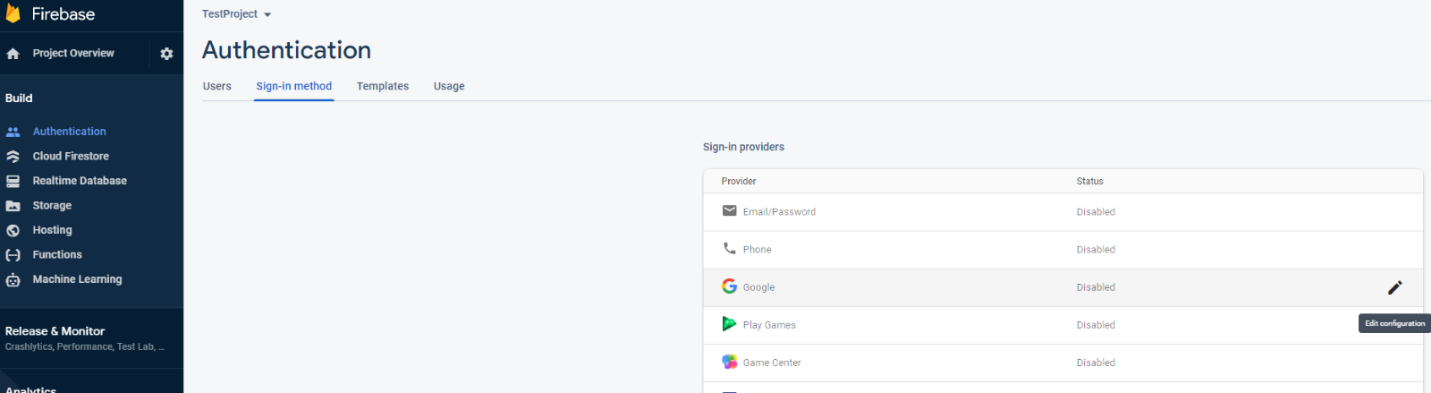
1. Wait until the project is ready



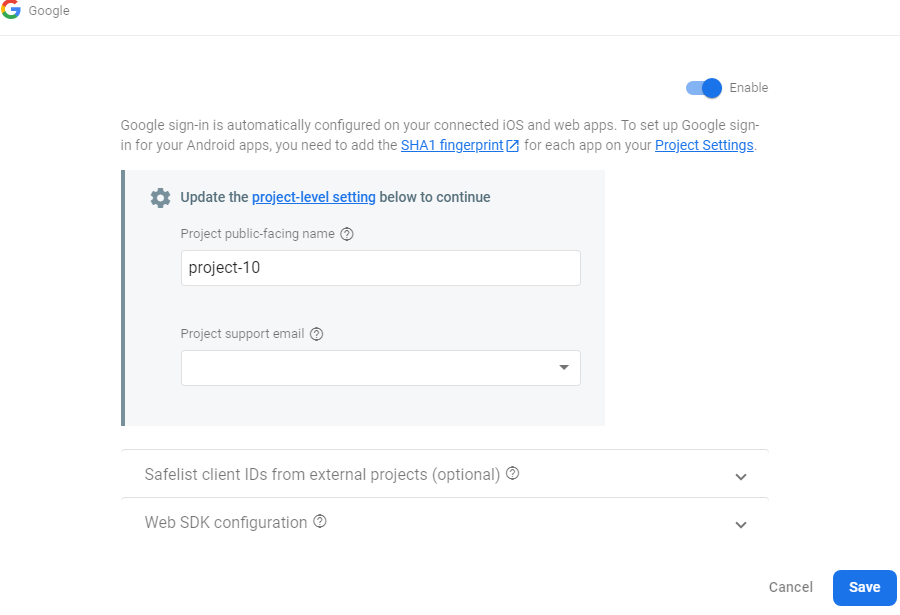
1. Open the authentication tab of the console.



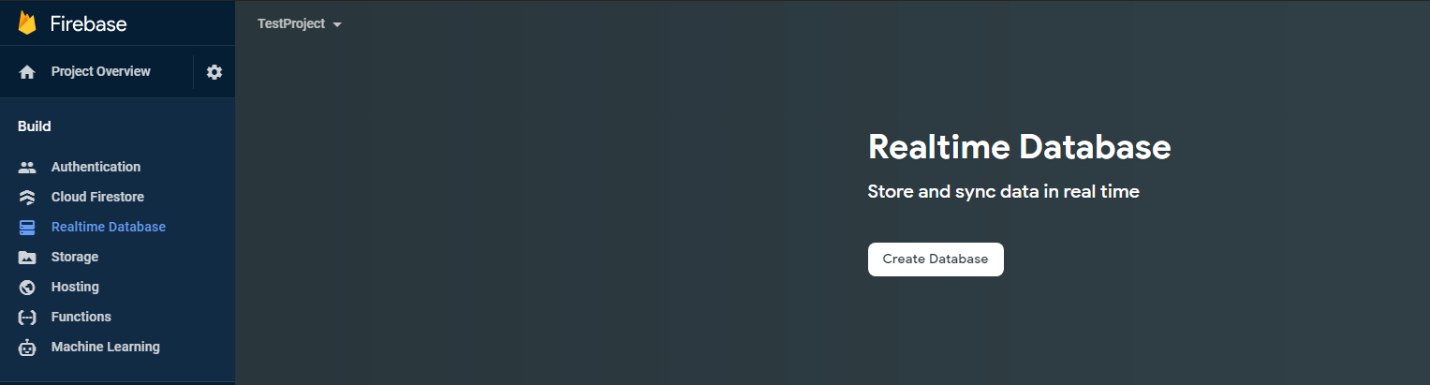
1. Click “Get Started,”  go to the “Sign-in method” tab, and click the “Edit configuration” button next to “Google.”



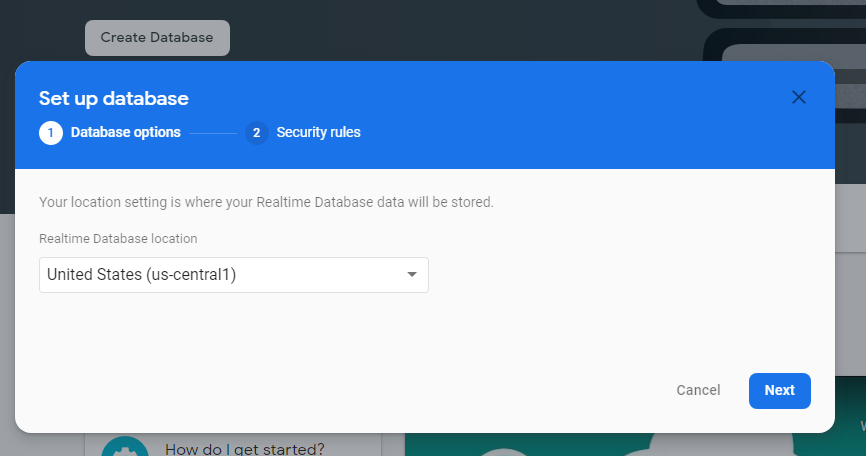
1. Click the enable button, set a “Project public-facing name” and a “Project support email,” and click the “Save” button.



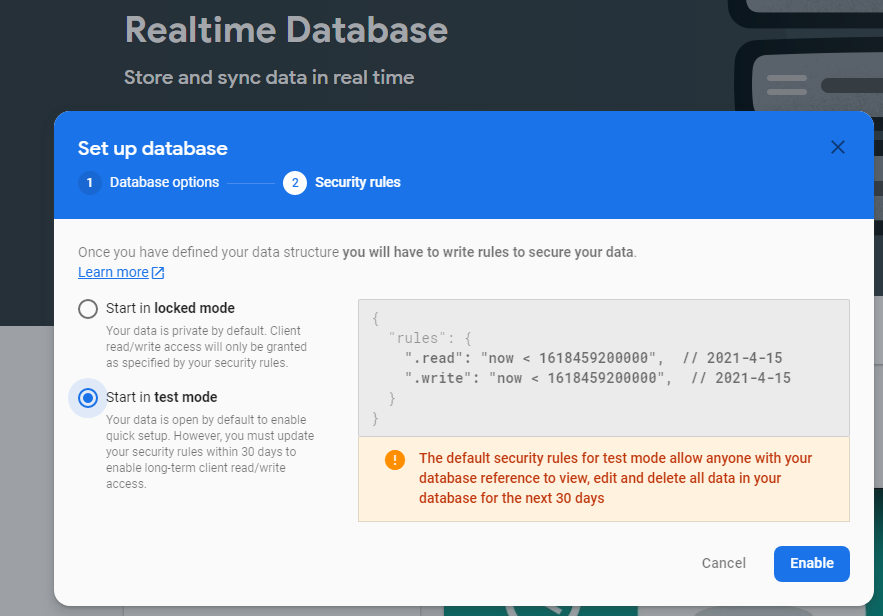
1. Click the “Realtime Database” button in the side menu and click the “Create database” button



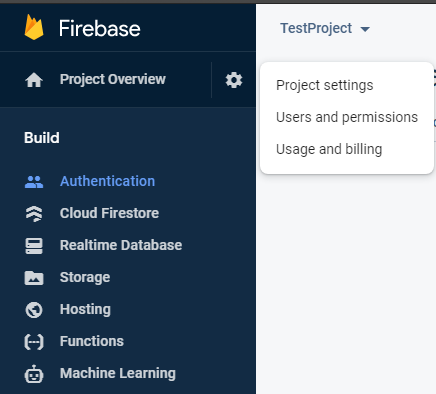
1. Select a region to host the database



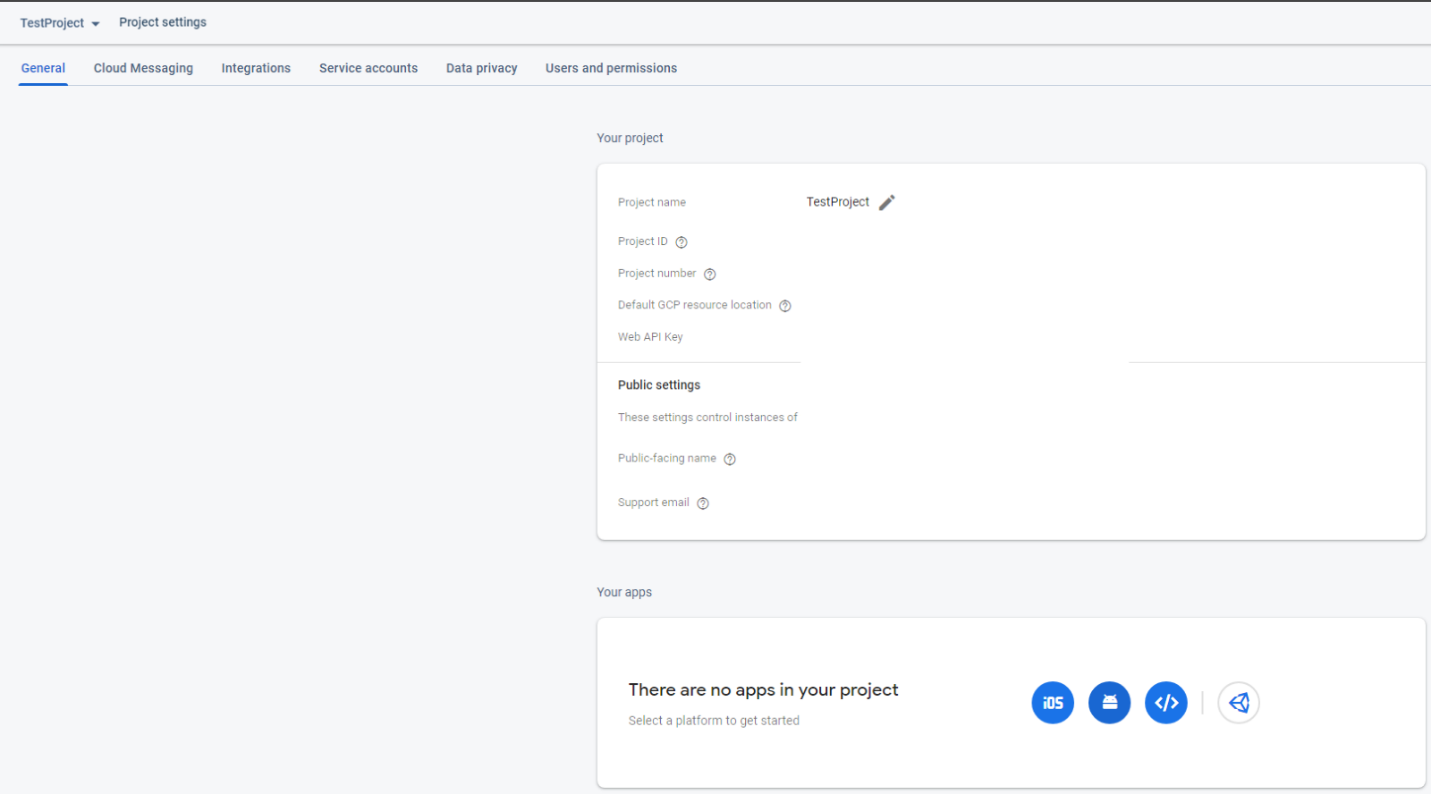
1. Choose to “Start in test mode” for a quicker setup or “Start in locked mode” to make data private by default. We recommend a test mode for initial testing of the system.  Then, click “Enable”



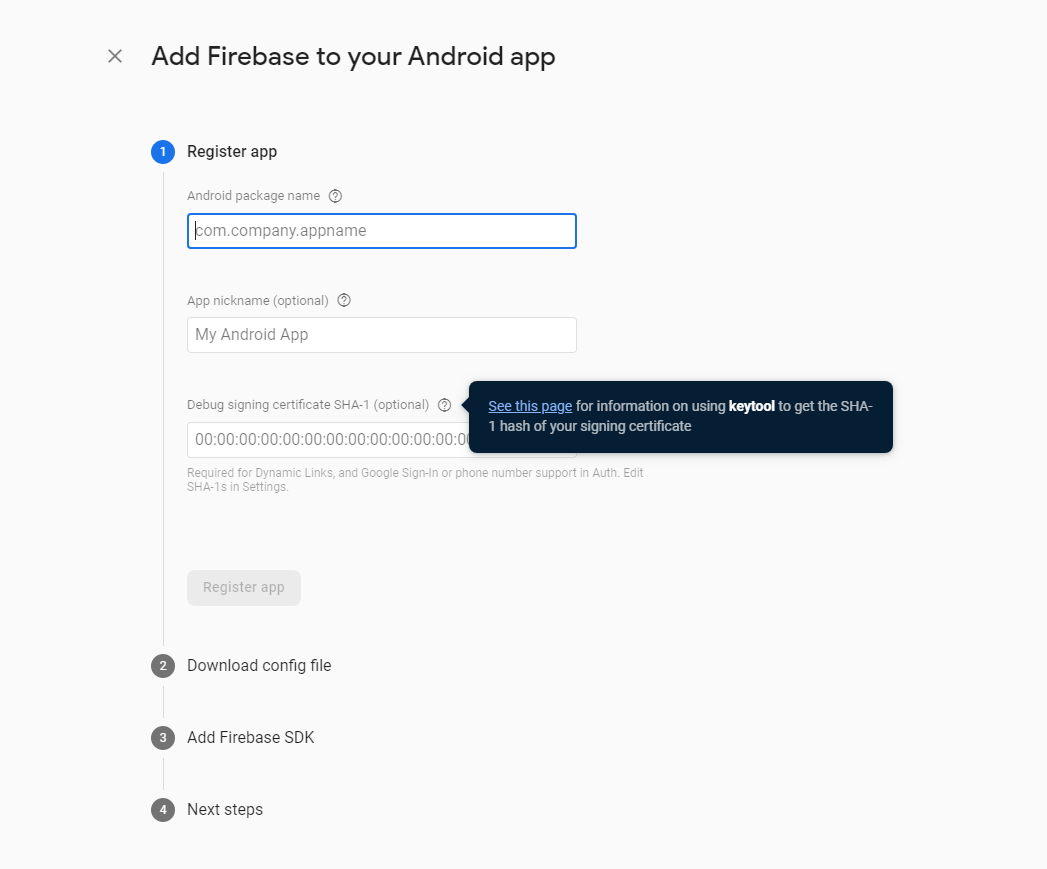
1. Click the gear symbol next to “Project Overview” and click “Project settings” in the pop-up menu.



1. At the bottom of the settings screen, the  “Your apps” button determines what platforms you will be connecting Firebase with. Android and IOS have supported platforms for using the application.

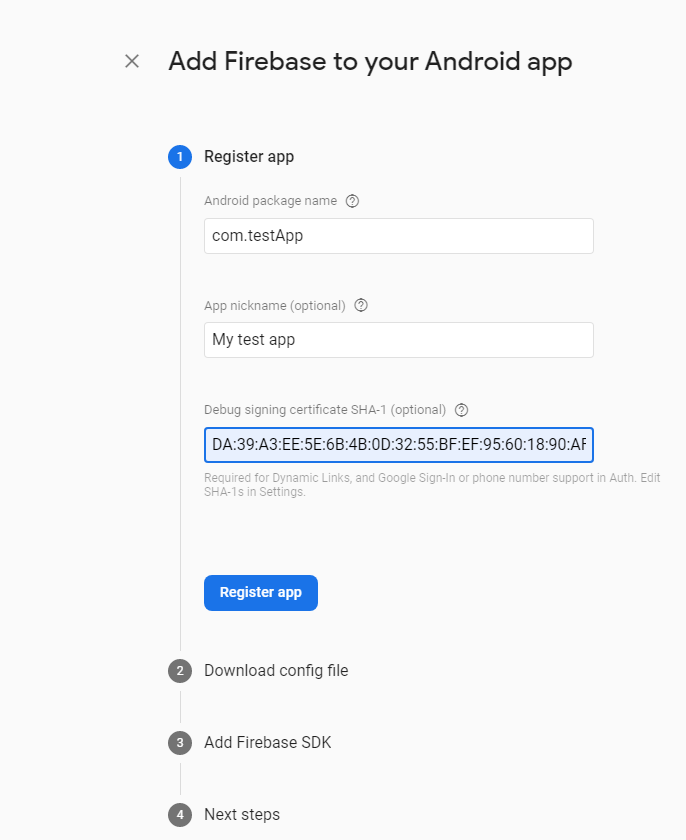


1. The following steps are for setting up Firebase for Android
2. Set the android package name and optionally set the app nickname. The “Debug signing certificate SHA-1 is required, so follow the link that is provided in the pop-up window that appears when you hover over the question mark icon.



b. Follow the steps in the following Google Developer page to generate the SHA-1 certificate. https://developers.google.com/android/guides/client-auth

c. Copy the certificate, paste it in the field on the “Add Firebase to Android app” page, and click “Register App”.  Then, continue through the remaining steps of the process to finish adding Firebase to the Android app.



# 5 Testing the Mobile Application

This section discusses the testing process that is done for the Form Scriber mobile application. A few testing challenges that may come up with testing on the mobile device are the different screen sizes (Help, 2021). With frequent updates to Android and iOS, it is crucial to continue to test the application and make sure it is compatible with the different versions of Android and iOS that will be developed in the future (Help, 2021).

A testing strategy that has been implemented is to take into account to manage quality and performance (Help, 2021).

* Research what mobile devices for Android and iOS are trending in the market for users and test those devices primarily (Help, 2021).
* When testing the application, use emulators for iOS and different Android versions, such as Q and R (Help, 2021). The bullet points below show the different types of emulators to test with.
  + Device Emulators – Android Studio offers a way for the developer or tester to plug in a mobile device and test the application in real-time.
  + Operating system Emulator – When testing on the application through Android Studio on a Mac OS, Xcode should be downloaded and updated. Through Xcode, iOS device emulators can be tested on Android Studio.

**Testing Objective**

The Mobile App testing for this project focuses on UI testing for the end-to-end or the system integration of the product to make sure that the software product that we will deliver has met the Software Requirements Specification (SWS) (Mansour, K. 2021) and performs functionalities as described in the Project Plan (Lopez-Willis, S. 2021) successfully. The testing will cover all of the Mobile Application functionalities, features, use cases, and code coverage for this software project.

Our tests will simulate the user's (for instance, a patient and a physician) interaction with the system, and the system responses to the user’s request for each category accurately and successfully to satisfy the stakeholders and the customer’s expectation. Best of all, our test will assure that the UMGC Software Capstone Project (SWEN670, Spring 2021) will be successful.

**Testing Procedures**

1. Prepare a Windows (version 8 or later) 64-bit operating system, x64 based processor laptop.
2. Install Flutter version 6.3.
3. Install Android Studio version 4.1 with Espresso, Flutter and Dart plugin.
4. Install DevTools (optional).
5. Install Github for collaboration with team members.
6. Install Microsoft Teams for collaboration with team members.
7. Install Git, download it from https://git-scm.com/downloads.
8. Clone the Github repository umgc/umgc.formbot.mobile. Use the Git command as follows: git clone < https://github.com/umgc/umgc.formbot.mobile.git>
9. Load the use case program to be tested into Android Studio.
10. Write the test program to test the use case. See the Android Studio User Guide <https://developer.android.com/studio/test> for reference.
11. Run the test
12. Check the test output, and all tests must pass. If a test failed, resolve the problem.
13. Check the result of the test that passed against the output expectation.
14. Collaborate the test output that deviated from the output expectation with the developers, lead developer, the project manager, and the stakeholders.
15. Document the test results.
16. Write the test report.

The detailed description of the tests and their results will be documented in the Test Report section on milestone 4. Schedule to be released on 3/30/2021.

# 6 Troubleshooting

## 6.1 Issue installing Flutter on MacOS Catalina

Isuue: MacOS Catalina with terminal zsh.

If the Mac OS has terminal zsh (Zshell) and you run into issues installing Flutter. Write the terminal command line ( vim .zprofile) to open the vim editor that will properly work for the install.

**vim .zprofile** if zsh is your default shell) to open the vim editor.

## 6.2 Android Studio ADB not detected

Reinstalling Android SDK after deleting the platform-tools dir doesn’t work for everyone follow these steps, they may help you:

1. In Android Studio go to File>Project Structure (Ctrl + Alt + Shift + S in windows)

2. Project Settings>Project>Project SDK must have a valid Android SDK selected.

3. Start the adb-server daemon with adb.exe start-server (this is to avoid the: adb wasn’t listening on port XXX error message).

4. Emulate a new device (if you’ve added %ANDROID\_HOME%\emulator to your path run emulator.exe -list-avds select one of the virtual devices from the list and run emulator.exe -avd, if not find out your Android SDK path and change the previous commands accordingly).

5. Check if the adb daemon is running and recognize the emulated device: open cmd prompt, run %ANDROID\_HOME%\platform-tools\adb.exe devices and it should list the device you have on at the moment.

## 6.3   Unable to launch Android Studio AVD Emulator

The system may display an error message such as “Cannot find adb.exe” when trying to launch the emulator on Windows.

If the user runs into this error while trying to run the emulator on Windows, there may be a few different causes. One of the most common is not having enough disk space to run the emulator. Freeing up more disk space is the solution to this issue.

One way to free up the disk space would be to:

Go to (C:) --à Users à Admin Address à Android à AVD à Remove unnecessary large files.

Another common cause is adb.exe has been quarantined via an antivirus software. Please check the antivirus quarantine log for more details and to return the file to the filesystem.

If neither one of these solutions fixes the issue, please visit <https://developer.android.com/studio/run/emulator-troubleshooting> for more information.

## 6.4 Android Studio - Gradle taking a long time

If starting the project seems to get stuck here,



Open Powershell.

Navigate to the **android** folder under your project folder (in this case, the project folder is formbot\_app).

Then run the command **./gradlew clean**



# Appendices:

# Appendix A - References

Help, S. T. (2021, February 18). *Mobile App Testing Tutorials (A Complete Guide With 30+ Tutorials)*. Retrieved from Software Testing Help: https://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/