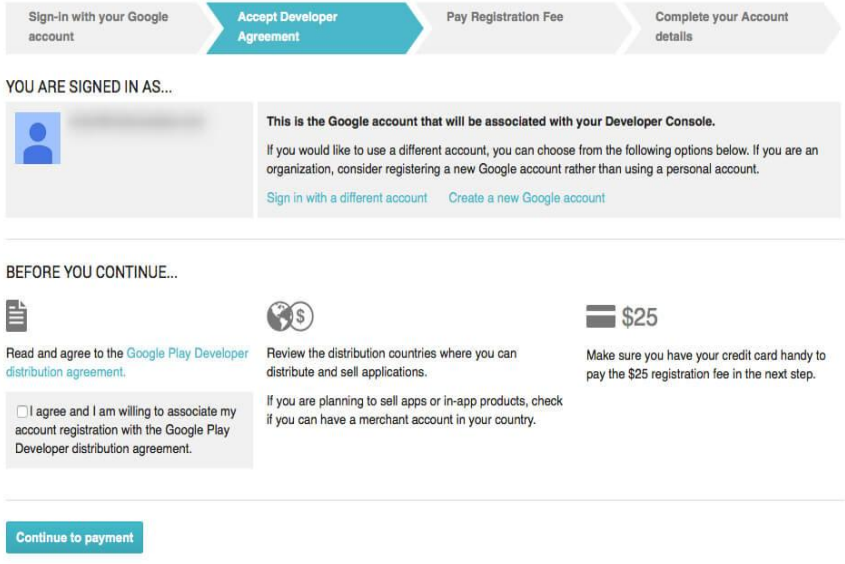
Apps Publishing

Step 1: Create a Google Developer account

- Do at the beginning of the app development process
  - Register a Google Developer Account to publish apps on the Play Market
  - Use any of current Google accounts or create another one to sign up for a Google Developer Account
  - A private or corporate account is fine
  - Any app is transferable to another account
- Creation process
  - Signing the Google Play Developer distribution agreement
  - Adding some personal information
  - Paying a **one-time** registration fee of **$25**
    - Usually, it takes no more than two days to get approval from Google
  - Information can be add or updated later



Step 2: Add a Merchant Account

- If you plan to sell paid apps or in-app purchases, you have to create a Google Merchant Account
  - You may need to provide additional information like a valid photo id (NID/Passport/ResidentCard/DrivingLicence/etc.)
  - Why?
- There you can manage app sales and your monthly payouts, as well as analyze sales reports
- Once you finish creating the Merchant profile, the developer account gets automatically linked to it

- **Note:** You don't need a merchant account for earning from ads
  - For monetizing from ads, you need
    - AdMob account to create and provide ads in your app
    - AdSense account for getting payments and accumulates payments from other platforms (i.e. website, youtube)
  - Payments from ads and sales/in-app purchase are paid by different transactions

Step 3: Prepare the Documents

- Paperwork always requires much effort, especially when it comes to any kind of legal documents
  - It is highly recommended starting to prepare the End User License Agreement (EULA) and Privacy Policy in advance
  - You can take the documents from similar apps as references and create your own based on them, or ask a lawyer to make everything from scratch.
- EULA is an agreement between you as an owner and a user of your product
- It contains:
  - What the users can do with the app, and what they aren't allowed to do
  - Licensing fees
  - Intellectual property information, etc.
  - Terms of Use or Terms and Conditions explain
    - What services you offer the users and how you expect them to behave in return
    - Though Google doesn’t demand Terms of Use, it’s better to publish them
- You can create one document, adding there Privacy Policy and Terms of Use chapters
- Pay special attention to include in the Privacy Policy the following information:
  - A complete list of personal data that is collected, processed and used through the app
  - Technical information that is collected about the device and the installed OS
  - Functional features of the app, its paid and free functionality
  - Place of registration of the company and/or location of the copyright holder of the application
- The chosen legal system and legislation that will be applied in resolving disputes and regulating legal relations
- The terms of subscription
  - Citizenship (residence) of the overwhelming majority of application users
  - Age criteria, the presence of specific content

Step 4: Study Google Developer Policies

- These documents explain how apps need to be developed, updated, and promoted to support the store's high-quality standards
- If Google decides that your product violates some policy chapters, it may be rejected, blocked, or even deleted from the Play Store
- Besides, numerous and repetitive violations may lead to the developer account termination
- **Read also**: Why Google and Apple May Remove Your App and How to Deal With That
- So study all the available information carefully about:
  - Restricted content definition
  - Store listing and promotion
  - Impersonation and intellectual property
  - Rules for monetization and ads
    - **Whatever is displayed in an ad is also consider as the content of your app**
  - Privacy, security and deception regulation
  - Spam and minimum functionality
- Google is constantly working on its policies, and it’s important to monitor the changes and stay up to date even after your app is released

Step 5: Technical Requirements

- You went through the development process, endless testing, and bug fixing, and finally, the “X-day” comes.
- Before moving on to the upload process, you need to check the following things:
  - Unique Bundle ID
    - The package name should be suitable over the life of your application.
    - You cannot change it after the distribution. You can set the package name in the application's manifest file.
- Signed App Release With a Signing Certificate
  - Every application should be digitally signed with a developer's certificate
  - The certificate is used to identify the author of an app and can’t be generated again
- The App Size
  - Google set the limit size of the uploaded file: 100MB for Android 2.3 and higher (API level 9-10, 14 and higher) and 50MB for lower Android versions.
  - If your app exceeds this limit, you can always switch to APK Expansion Files.
- The File Format
  - Two possible release formats are accepted by Google: app bundle and .apk. However, .aab is the preferred one. To use this format, you need to enroll in app signing by Google Play.
- You may learn more about app file technical requirements in the Developer Documents, Prepare for the release guide.

Step 6: Creating the App on the Google Console

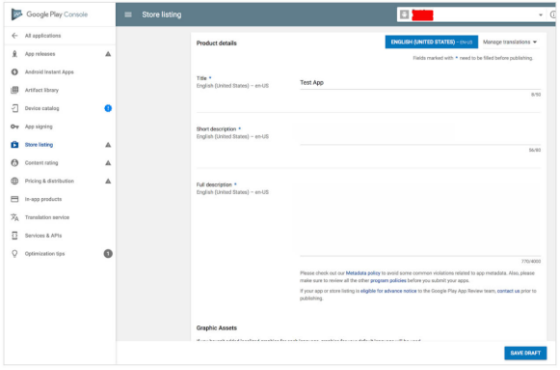
- Now you have the file that is ready for uploading. It’s time to get to the fun part.
- select *Create Application* from the app developer console
- Choose the app’s default language from the drop-down menu
- Add a brief app description (you can change it later)
- Tap on *Create*
- After this, you will be taken to the store entry page, where we will add the complete data about the app.

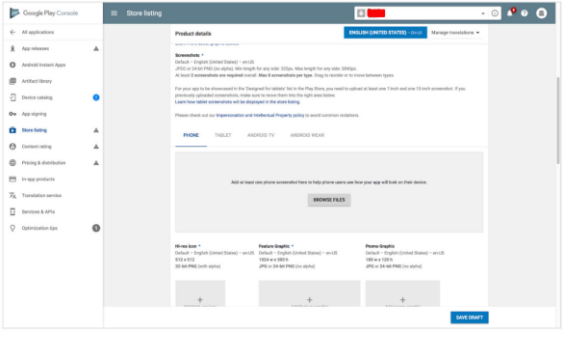
Publish an App

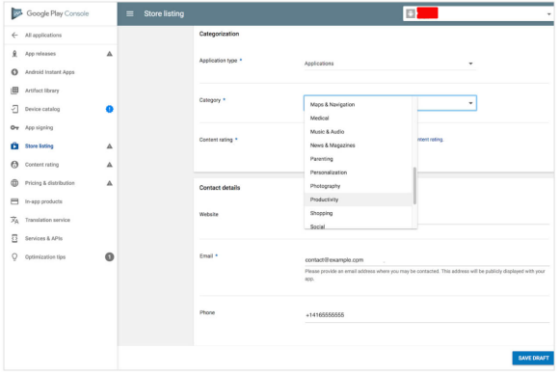
- There are few things that you need to provide with your application while publishing it
- Such as:
  - ✓ Listing details
  - ✓ Store Graphics
  - ✓ Screenshots (Minimum 2/3)
  - ✓ Videos (Optional)

Step 7: Store Listing

- **App Name :-** Give suitable app name for your application.
- **App Description :-** Write suitable description for your app, not exceeding 4000 characters.
- **Category :-** It is required that you select an appropriate main category for you application
    - You can add another four categories as well (used for indexing)
- **Application Type :-** It is also required that you set a category for your application. This could be either set **as application or game**
- **Organization Name :-** It is also required that you mention the name of organization while filling up the details.
- **Support information :-** This includes URL , Email, Phone number etc.







Store Graphics

- **App Icon**
  - ✓ 512 x 512 px
  - ✓ 32-bit PNG
  - ✓ Max size 1024KB
  - ✓ Transparency allowed

- **Feature graphic** This is also optional but is required to get your app featured anywhere. This image has the following constraints
  - ✓ 1024 x 500 px
  - ✓ 24-bit PNG
  - ✓ **No Transparency**

Screenshots

- Screenshots are important because people will see screenshot before installing app on phone. So it must convey your main functionality.

- **Constraints for the screenshots** are
  - ✓ 5/6 additional optional
  - ✓ 480 x 800 px, or 480 x 854 px
  - ✓ 72dpi, RGB, flattened
  - ✓ No transparency
  - ✓ High quality JPEG or 24-bit PNG
  - ✓ No borders
  - ✓ Can show status bar

Video (Optional)

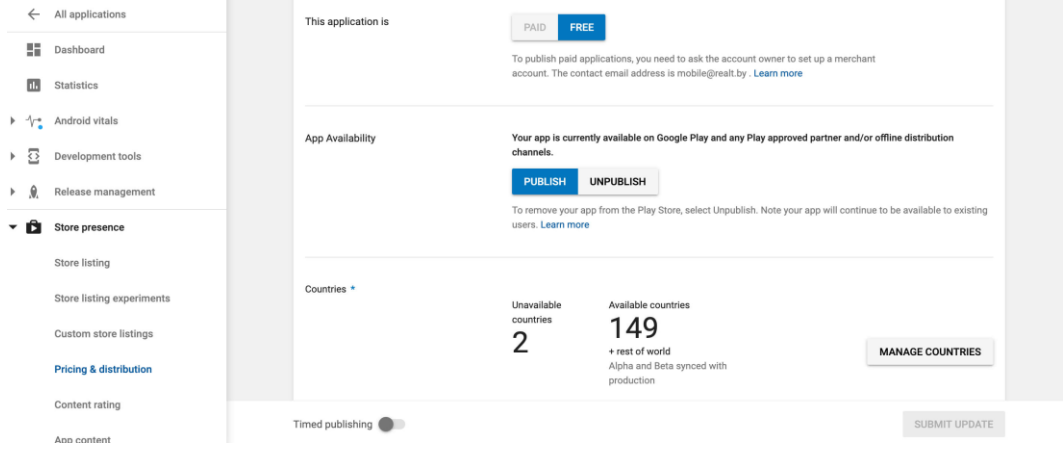
- Finally you can even show how your app works through a video
- This can greatly increase the chances of popularity of your app as the viewer can get synopses of your application
- Google recommends the length of the video to be anything between 30 seconds to 2 minutes

Step 8: Content Rating

- In order not to be marked as an Unrated App (that may lead to app removal), pass a rating questionnaire.
  - You can easily find this section on the left-side menu.
- The information provided in the questionnaire must be accurate.
- Any misrepresentation of your app's content might lead to suspension or removal of the Play Store account.
- Click on *Save Questionnaire* once you complete the survey
- Click on *Calculate Rating*
- In the end, click on *Apply Rating* to confirm the rating and move forward with the pricing & distribution plan

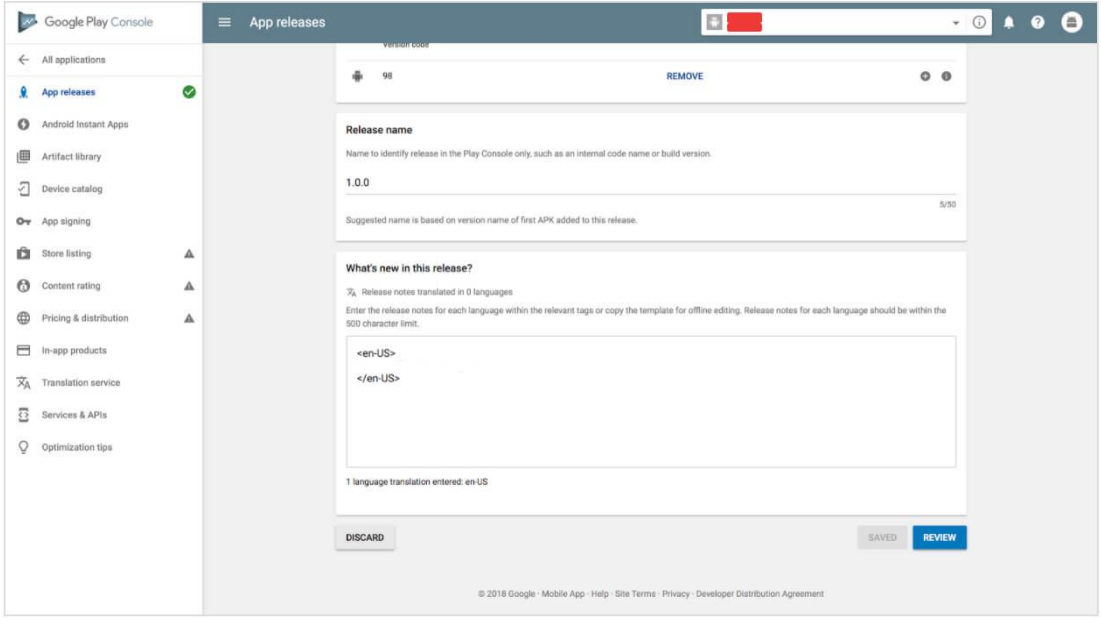
Step 9: Pricing the Application

- In the *Pricing and distribution* section, you need to fill the following information:
  - Whether your app is free or paid
  - Where the app will be available (just choose the countries from the list)
  - Whether your app will be available only on the specific devices
  - Whether the app has sensitive content and is not suitable for children under the age of 13
  - Whether your app contains ads
- Remember that you can change your paid app to a free one later, but you cannot do the vice versa. If you decide later that you want to distribute it for money, you'll have to create another app.



Step 10: Upload APK/AAB and Send for Review

- Finally, you are ready to upload your app file.
- Go to the *App Releases* section on the left panel
  - Three options for publishing the app: Production, Beta and Alpha tracks.
- The Alpha version assumes closed testing and is available only to those who you invite as testers
- The Beta version means that anyone can join your testing program and send feedback to you.
- Recommended starting with Alpha or Beta versions
  - After passing the review process, your app will not be available to everyone on the Play Store.



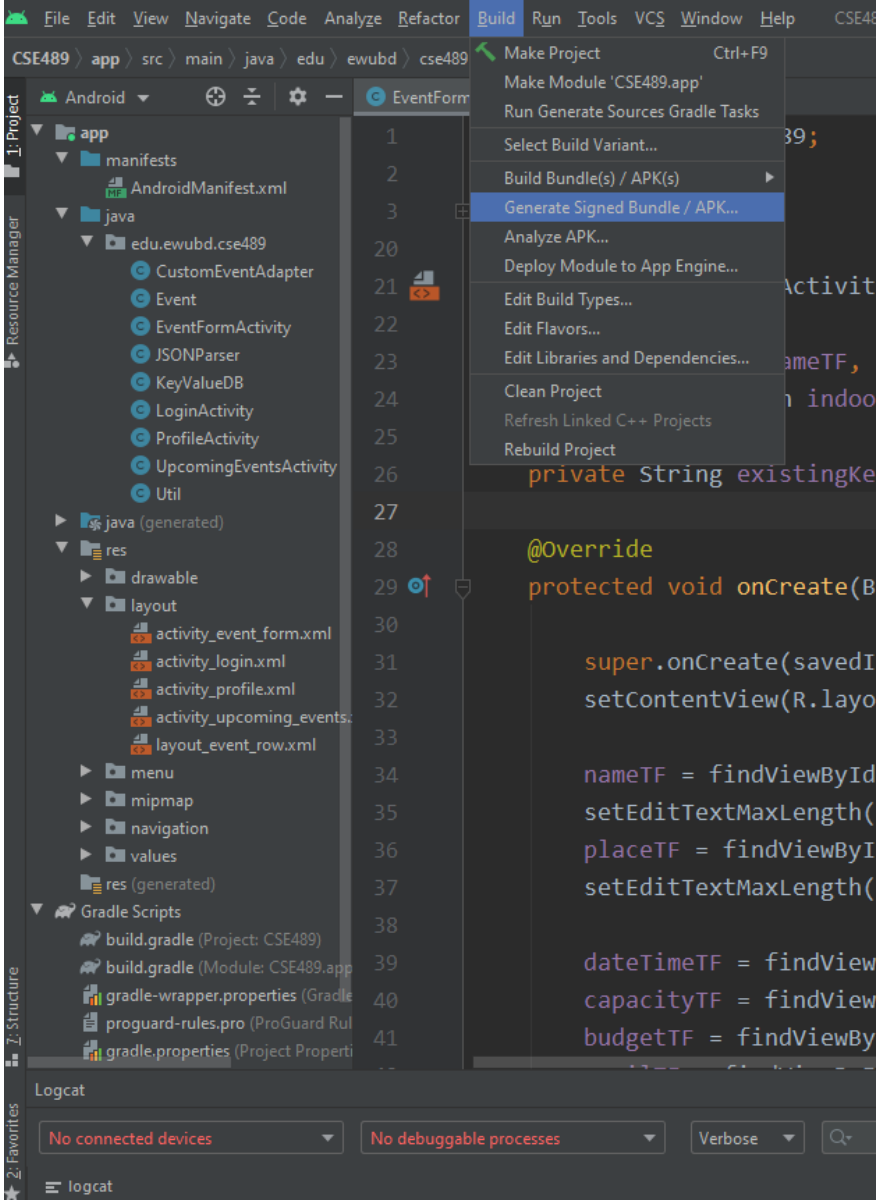
Final Check

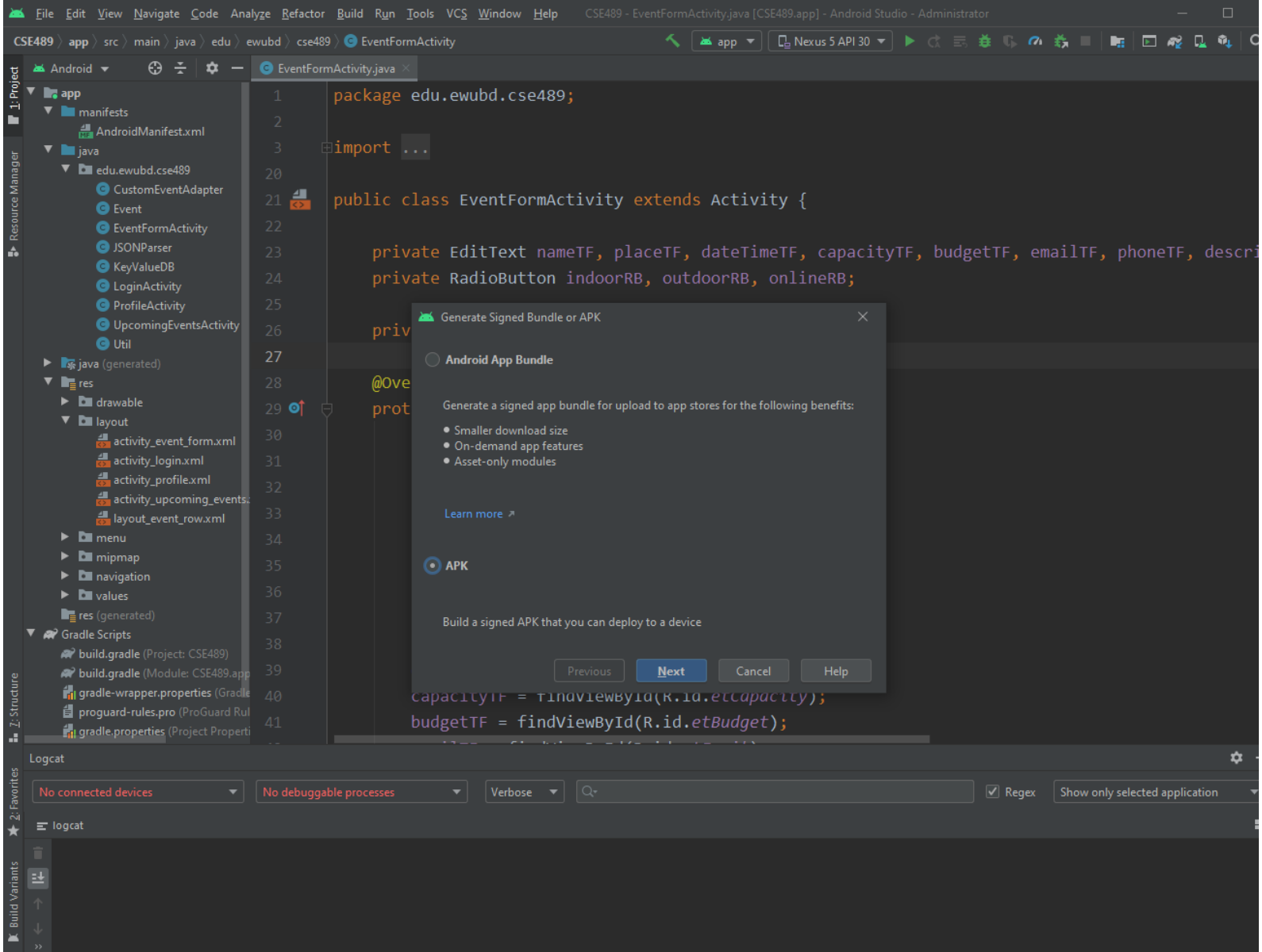
- Apart from this you need to set some other parameters as well. Like

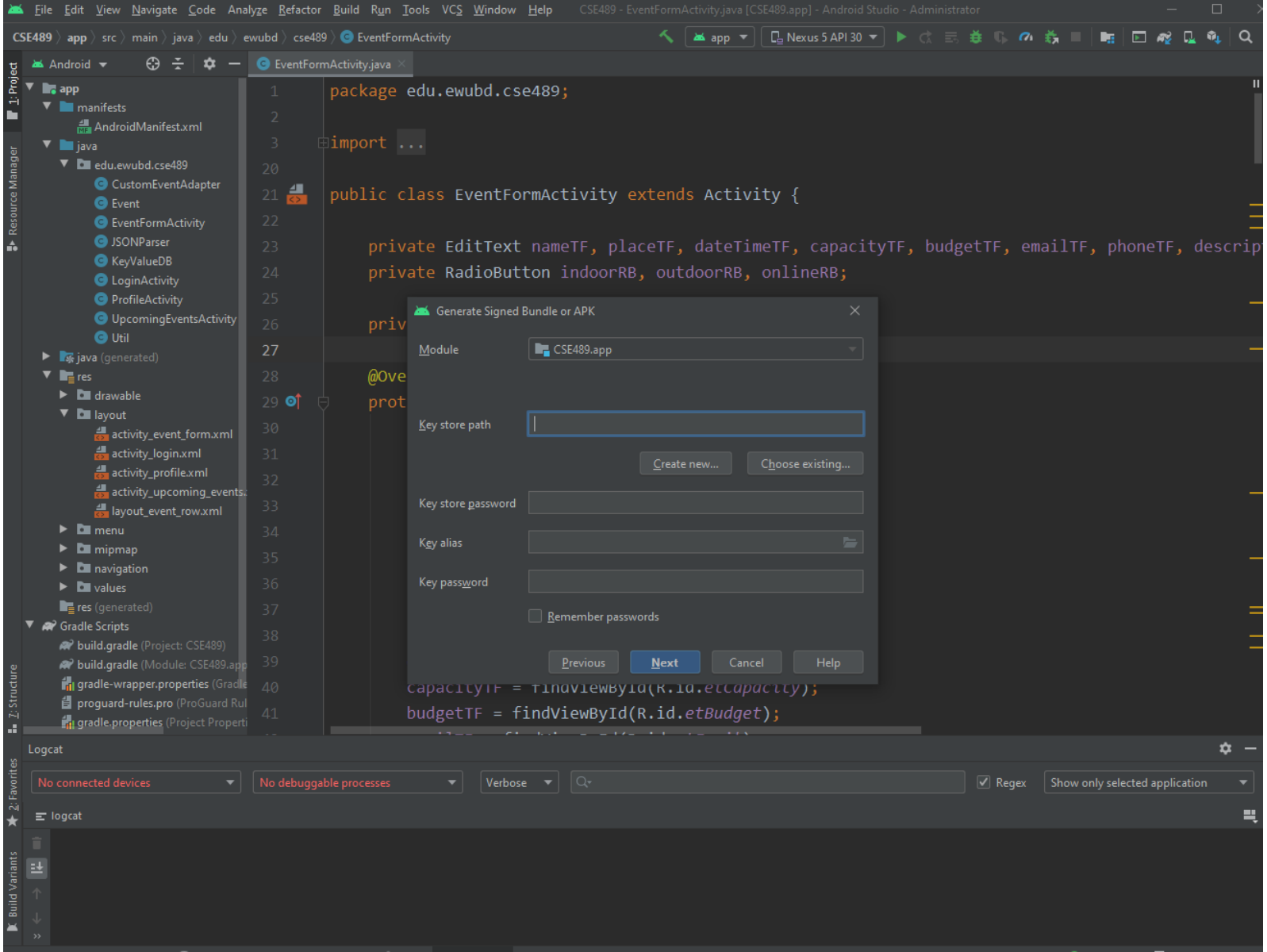
  - ✓ **Size- the size of the app should not exceed 50 Mb**
  - ✓ **Pricing- you need to decide whether your app should be free or priced.**
  - ✓ **Country distribution**- you get to decide which country or territories your app should be distributed.
  - ✓ **Rating-** This is the maturity rating required of your app. You can set is as everyone, low maturity, medium maturity, high maturity.

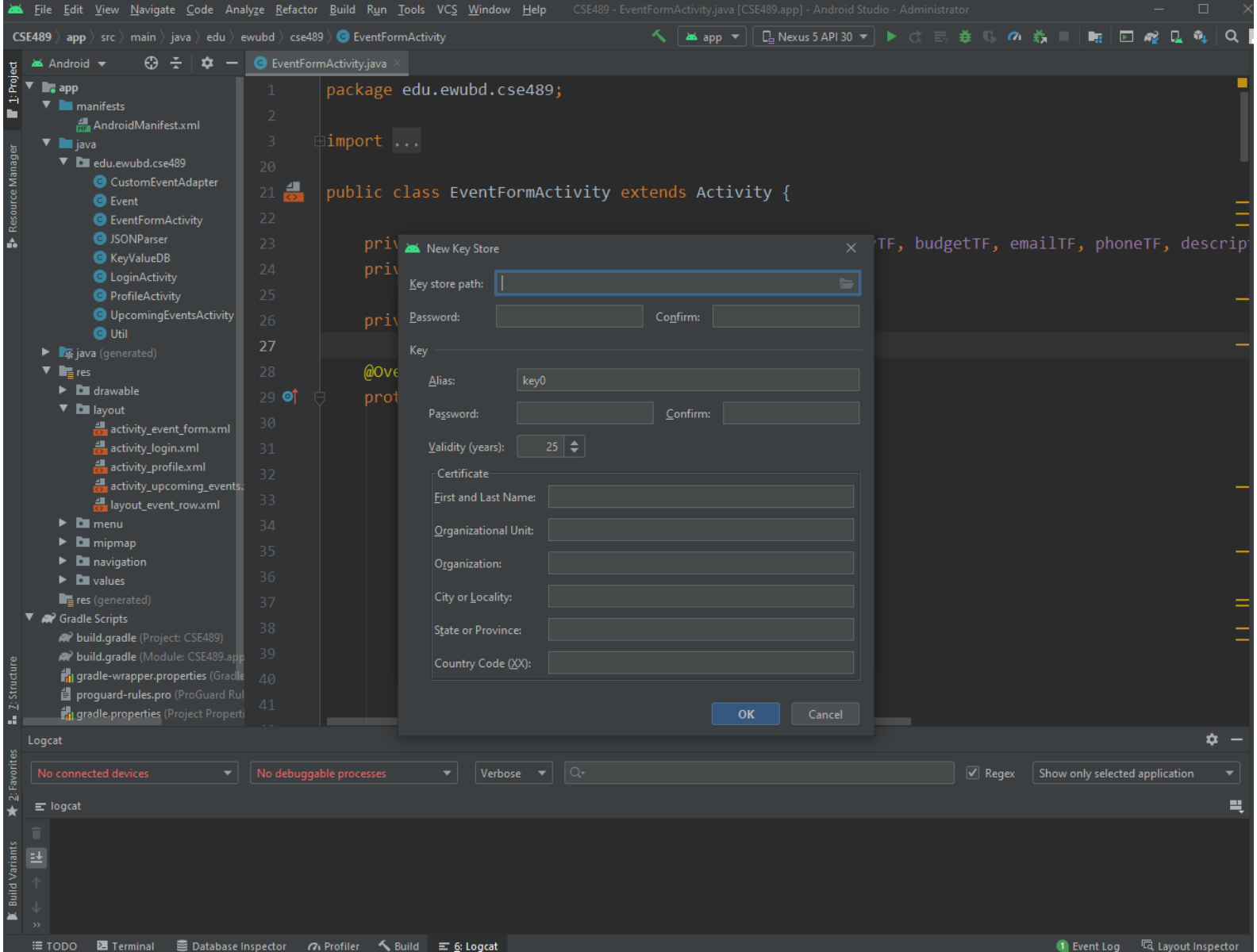
- **At least once you need to read Google play store policies** □
- This APK is then uploaded. This includes
  - ✓ Optimization
  - ✓ Building
  - ✓ signing with your release key
  - ✓ Final testing.
- **All this and you are good to go.**

Exporting APK from Android Studio


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CSE489 app src main java edu ewubd cse489
Make Project Ctrl+F9
Make Module 'CSE489.app'
Run Generate Sources Gradle Tasks
Select Build Variant...
Build Bundle(s) / APK(s)
Generate Signed Bundle / APK...
Analyze APK...
Deploy Module to App Engine...
Edit Build Types...
Edit Flavors...
Edit Libraries and Dependencies...
Clean Project
Refresh Linked C++ Projects
Rebuild Project
Android
app
manifests
AndroidManifest.xml
java
edu.ewubd.cse489
CustomEventAdapter
Event
EventFormActivity
JSONParser
KeyValueDB
LoginActivity
ProfileActivity
UpcomingEventsActivity
Util
java (generated)
res
drawable
layout
activity_event_form.xml
activity_login.xml
activity_profile.xml
layout_event_row.xml
menu
mipmap
navigation
values
res (generated)
Gradle Scripts
build.gradle (Project: CSE489)
gradle-wrapper.properties
proguard-rules.pro
gradle.properties
private String existingKe
@Override
protected void onCreate(B
super.onCreate(savedI
setContentView(R.layo
nameTF = findViewById
setEditTextMaxLength(
placeTF = findViewByI
setEditTextMaxLength(
dateTimeTF = findView
capacityTF = findView
budgetTF = findViewBy
Logcat
No connected devices
No debuggable processes
Verbose
logcat


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CSE489 - EventFormActivity.java [CSE489.app] - Android Studio - Administrator
CSE489 app src main java edu ewubd cse489 EventFormActivity
app
Nexus 5 API 30
Android
EventFormActivity.java
app
manifests
AndroidManifest.xml
java
edu.ewubd.cse489
CustomEventAdapter
Event
EventFormActivity
JSONParser
KeyValueDB
LoginActivity
ProfileActivity
UpcomingEventsActivity
Util
java (generated)
res
drawable
layout
activity_event_form.xml
activity_login.xml
activity_profile.xml
activity_upcoming_events
layout_event_row.xml
menu
mipmap
navigation
values
res (generated)
Gradle Scripts
build.gradle (Project: CSE489)
build.gradle (Module: CSE489.app
gradle-wrapper.properties (Gradle
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Properti
package edu.ewubd.cse489;
import ...
public class EventFormActivity extends Activity {
private EditText nameTF, placeTF, dateTimeTF, capacityTF, budgetTF, emailTF, phoneTF, descri
private RadioButton indoorRB, outdoorRB, onlineRB;
priv
@Ove
prot
budgetTF = findViewById(R.id.etBudget);
Generate Signed Bundle or APK
Android App Bundle
Generate a signed app bundle for upload to app stores for the following benefits:
Smaller download size
On-demand app features
Asset-only modules
Learn more
APK
Build a signed APK that you can deploy to a device
Previous
Next
Cancel
Help
Logcat
No connected devices
No debuggable processes
Verbose
Regex
Show only selected application
logcat


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CSE489 - EventFormActivity.java [CSE489.app] - Android Studio - Administrator
CSE489 app src main java edu ewubd cse489 EventFormActivity
app
Nexus 5 API 30
Android
EventFormActivity.java
app
manifests
AndroidManifest.xml
java
edu.ewubd.cse489
CustomEventAdapter
Event
EventFormActivity
JSONParser
KeyValueDB
LoginActivity
ProfileActivity
UpcomingEventsActivity
Util
java (generated)
res
drawable
layout
activity_event_form.xml
activity_login.xml
activity_profile.xml
activity_upcoming_events.
layout_event_row.xml
menu
mipmap
navigation
values
res (generated)
Gradle Scripts
build.gradle (Project: CSE489)
build.gradle (Module: CSE489.app
gradle-wrapper.properties (Gradle
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Properti
1 package edu.ewubd.cse489;
3 import ...
21 public class EventFormActivity extends Activity {
23 private EditText nameTF, placeTF, dateTimeTF, capacityTF, budgetTF, emailTF, phoneTF, descrip
24 private RadioButton indoorRB, outdoorRB, onlineRB;
26 priv
27
28 @Ove
29 prot
40 capacityTF = findViewById(R.id.etCapacity);
41 budgetTF = findViewById(R.id.etBudget);
Generate Signed Bundle or APK
Module
CSE489.app
Key store path
Create new...
Choose existing...
Key store password
Key alias
Key password
Remember passwords
Previous
Next
Cancel
Help
Logcat
No connected devices
No debuggable processes
Verbose
Regex
Show only selected application
logcat
1: Project
Resource Manager
7: Structure
2: Favorites
Build Variants


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
CSE489 - EventFormActivity.java [CSE489.app] - Android Studio - Administrator
CSE489 app src main java edu ewubd cse489 EventFormActivity
app
Nexus 5 API 30
Android
EventFormActivity.java
app
manifests
AndroidManifest.xml
java
edu.ewubd.cse489
CustomEventAdapter
Event
EventFormActivity
JSONParser
KeyValueDB
LoginActivity
ProfileActivity
UpcomingEventsActivity
Util
java (generated)
res
drawable
layout
activity_event_form.xml
activity_login.xml
activity_profile.xml
activity_upcoming_events.
layout_event_row.xml
menu
mipmap
navigation
values
res (generated)
Gradle Scripts
build.gradle (Project: CSE489)
build.gradle (Module: CSE489.app
gradle-wrapper.properties (Gradle
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Properti
1 package edu.ewubd.cse489;
2
3 import ...
20
21 public class EventFormActivity extends Activity {
22
23 priv
24 priv
25
26 priv
27
28 @Ove
29 prot
30
31
32
33
34
35
36
37
38
39
40
41
TF, budgetTF, emailTF, phoneTF, descrip
New Key Store
Key store path:
Password:
Confirm:
Key
Alias:
key0
Password:
Confirm:
Validity (years):
25
Certificate
First and Last Name:
Organizational Unit:
Organization:
City or Locality:
State or Province:
Country Code (XX):
OK
Cancel
Logcat
No connected devices
No debuggable processes
Verbose
Regex
Show only selected application
logcat
TODO
Terminal
Database Inspector
Profiler
Build
6: Logcat
Event Log
Layout Inspector

END