

Programming in Android



Session: 14

Google Play Store

Objectives

- ◆ Explain the requirement for Google Play Store
- ◆ Explain the different versions and feature set
- ◆ Explain the share of Android in the market
- ◆ Explain different kinds of devices available in the market
- ◆ Describes the making of an .apk file
- ◆ Explain the process of publishing the .apk file in Google Play Store
- ◆ Explain the best practices to be followed
- ◆ Explain good marketing and promotion strategies

Introduction

- ◆ Google Play Store is Google's market place for Android Applications, music, movies, and books
- ◆ Google Play Store comes pre-installed in all Android Devices
- ◆ It serves as the main source for application for more than 70% of Android users
- ◆ Hence, it is in the best interest of a developer to publish the app on Google Play



Google play

What is the Play Store?

- ◆ Google Play Store (formerly, Android Market) is an online electronic store or digital application distribution platform for Android powered devices
- ◆ It is a robust publishing platform
- ◆ The services offered by Play Store to the users includes downloading free and paid applications, books, movies, music, magazines, and so on
- ◆ Users can also purchase mobile devices such as Google-Nexus, Chromebook, and so on through Play Store
- ◆ The developer will have access to revenue generating tools, such as in-app-billing and application-licensing
- ◆ The developer will also know the sale trends and understand the end users

Market Shares and Targeting the Right Versions

- ◆ The Android market is fragmented. The hardware and the OS version vary from one device to another
- ◆ This enables innovation and rapid progress on the platform
- ◆ However, at the same time, it adds the burden on the developer to maintain compatibility
- ◆ Targeting the latest and greatest may give you better quality application, but the potential market of the application will be reduced drastically



Hardware Requirements

- ◆ The developer needs to try to utilize the least common denominator in terms of hardware
- ◆ Support alternatives when the hardware is not available
- ◆ If the application is using location based services and GPS hardware is not available, the application uses the network location to generate an estimate
- ◆ If GPU acceleration is being used, try to make sure that the application runs properly in a device from two generations earlier (Nexus 4 if Nexus 6 is the current generation)



Operating System Updates

- ◆ Operating System upgrades on the Android platform are much slower than any other platform
- ◆ The user cannot decide when/if he can upgrade to a new OS version. The device will receive the update only if the vendor pushes an update
- ◆ The vendor may also choose to discontinue updates for the device leaving the device forever at the same OS version
- ◆ Only the Nexus devices receive the update on day one
- ◆ Hence, the developer needs to make sure that by targeting the latest version, he is not alienating potential customers
- ◆ Most applications have no particular use of the latest API
- ◆ Unless it is absolutely mandatory to make the switch, try to use the old API

Market Shares

- As of May 2015, the number of devices executing particular version of Android are displayed in the following table:

Android Version	Version No.	API	Share
Lollipop	5.1	22	0.7 %
Lollipop	5.0	21	9 %
Kitkat	4.4	19	39.8 %
JellyBean	4.2	18, 17, 16	39.2 %
Ice Cream Sandwich	4.0	15	5.3 %
Gingerbread	2.3	9	5.7 %
Froyo	2.2	8	0.3 %



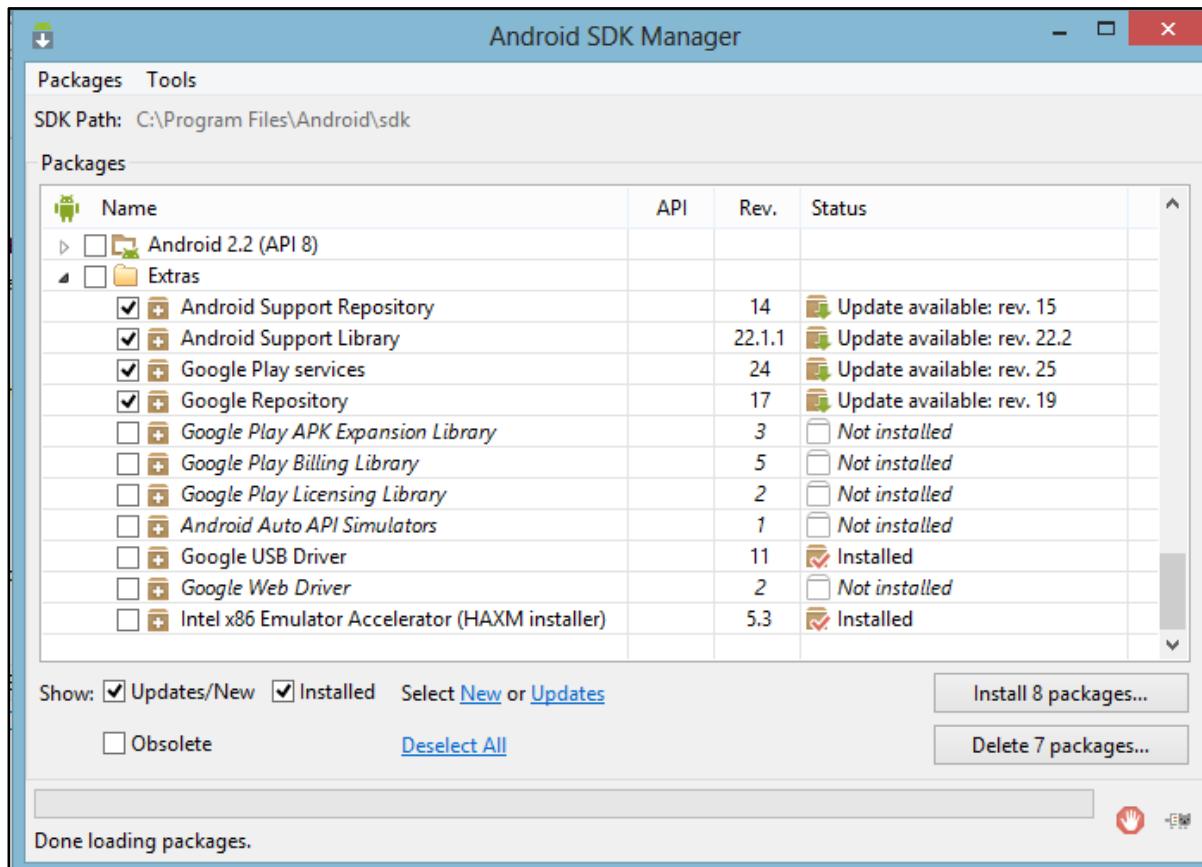
- It must be ensured that the application supports atleast 80% of the market share for it to be successful. In that case, the minimum SDK version suitable is JellyBean.

Support Library

- ◆ In order to ease the issue with variation of API, Android has introduced Support Libraries to use in applications
- ◆ By using these libraries, the developer can continue to use the newer API and still make sure that the application will be compatible with older devices
- ◆ The Library chooses the appropriate implementation for the developer
- ◆ The packages android.support.v4 and android.support.v7 should to be imported for the v4 and v7 support libraries respectively
- ◆ The classes in this library have a prefix of AppCompat with the same API

Adding Support Libraries

- Support Libraries can be added from the SDK Manager as shown in the following figure:



Perquisites to Publishing on the Play Store

- ◆ Certain steps need to be taken before publishing the application to End users
- ◆ Before uploading the application in Android market, the basic points to be noted are as follows:
 - ❖ Testing the application
 - ❖ Checking the application performance
 - ❖ SDK Compatibility

Testing the Application

- ◆ The application developed must be tested well before uploading it to the Play Store so that the uploaded application is not only error/bug free but also does not crash at users' end
- ◆ A stable application and good reviews are key to the success of the application
- ◆ The output varies from one device to another and also from one version of Android to another
- ◆ The application is required to be tested on real devices of different types before being uploaded to the Play Store for the users to download
- ◆ The resolutions supported and the layouts for the supported resolutions need to be thoroughly tested

Preparations 1-2

- ◆ **Collect Material for Release:**

- ◆ This includes preparation of End User License Agreement (EULA) for the application
- ◆ This will help to protect not only the developer and the organization but also the intellectual property rights
- ◆ It also includes the process of obtaining an encrypted key for digitally signing the application and creation of an icon for the application

- ◆ **Arrange Application for Resource:**

- ◆ This includes gathering of all the materials required for configuring the applications which includes configuration changes made to the source code, resource files, and the manifest file of the application
- ◆ Besides this the developer has to clean up the project, update the manifest file setting, update the URLs for servers and services, and so on

Preparations 2-2

- ◆ **Build Application for Resource:**
 - ❖ This includes signing the application and building the application for release
- ◆ **Prepare Remote Server:**
 - ❖ This includes the process of ensuring that if a remote server is used it is secure and configured for use
- ◆ **Test Application for Resource:**
 - ❖ This includes testing the application to ensure that the application works properly under real device and varying conditions

Application Performance and Pre-Publish Tasks

◆ Application Performance

- ❖ The performance of the application is one of the most important concerns to the developer
- ❖ A sluggish application is destined to receive bad reviews followed by less downloads
- ❖ Code optimization tools such as lint and code clean up tools (part of the IDE) can be used to improve the performance of the application

◆ Pre-Publish Tasks

- ❖ Set permissions
- ❖ Establish version, set icon, and application label
- ❖ Set compatibility options
- ❖ Remove log data
- ❖ Export the project and create the key using Eclipse

Setting Permissions

- ◆ **<uses-permission> element**
 - ❖ In this tag, specify only those permissions that are relevant and required for executing the application
- ◆ **android:icon and android:label attributes**
 - ❖ In the <application> element, specify values for these attributes as these are displayed to the user. The default icon needs to be replaced with a proper icon representing the application
- ◆ **android:versionCode and android:versionName attributes**
 - ❖ In the <manifest> element, specify values for these attributes

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"
/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
```

- ◆ The version names have to be specified in the AndroidManifest file that is written within the <manifest> tag
- ◆ The version code starts from 1 and is required to be incremented by 1 for each upload. Version name specifies the release version and always should be in ascending order

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.helloandroid"  
    android:versionCode="1"  
    android:versionName="1.0" >
```

Specifying Icon, Application Label, and SDK Version

- ◆ The icon to be set for the application and the name of the application have to be specified inside the <application> tag in AndroidManifest file
- ◆ The minSdkVersion, specifies the minimum sdk version the application will support
- ◆ The targetSdkVersion, specifies the version to which the application is built for
- ◆ The maxSdkVersion, specifies the maximum sdk version the application can support

```
<uses-sdk android:minSdkVersion="8"  
        android:targetSdkVersion="10"  
        android:maxSdkVersion="16" />
```

Other Pre-Publish Tasks

- ◆ **Feature Restriction**
 - ❖ Features that are essential to the functionality of the application can be specified using the Uses-feature tag
 - ❖ This ensures that the application is not installed on devices that do not have the feature
- ◆ **Compatibility Options**
 - ❖ The Android system will decide on which platform the application will be installed depending on the target specified in the manifest file
 - ❖ The play store automatically identifies the compatible devices and prevents unsupported devices from installing the application
 - ❖ The publisher can change this from the Google Play Developer Console
- ◆ **Remove Log Data**
 - ❖ Remove the calls to logs, as it will display output data in LogCat always
 - ❖ Unnecessary resource files are required to be removed as it will lead to increase in size of the application
 - ❖ This can also lead to security vulnerabilities and leak key information about the source code of the application

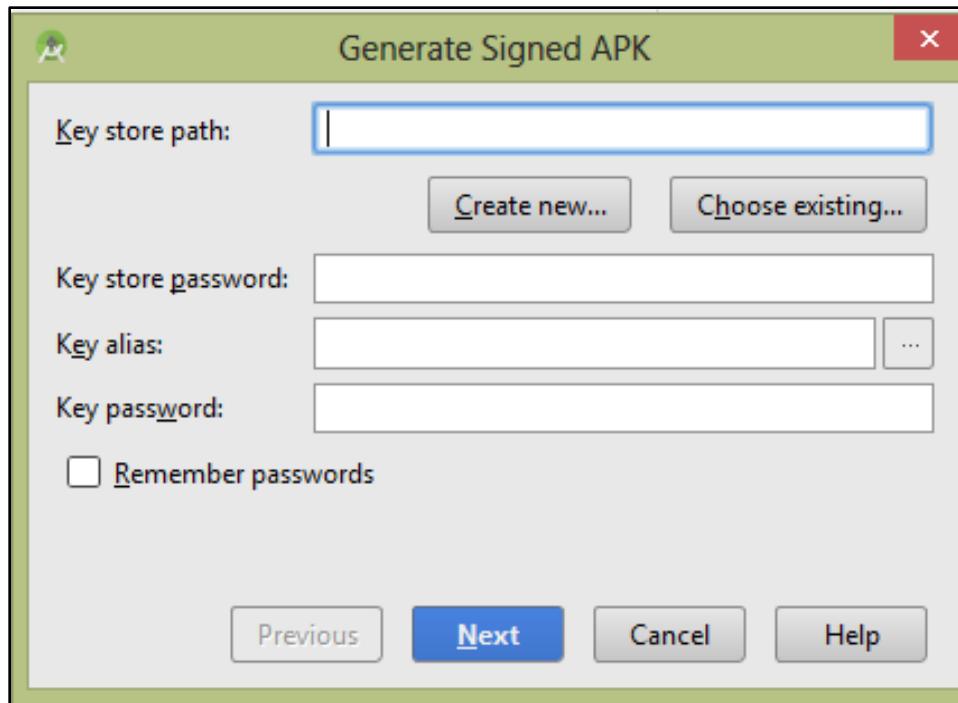
Signing an Application 1-5

- ◆ Before an application can be uploaded to any application store, the application needs to be signed by the developer
- ◆ There is no need for a Certification Authority
- ◆ The developers can generate their own certificates for signing the application
- ◆ The purpose of signing the application is to verify the authenticity of the application and to ensure that it is not a counterfeit or tampered with



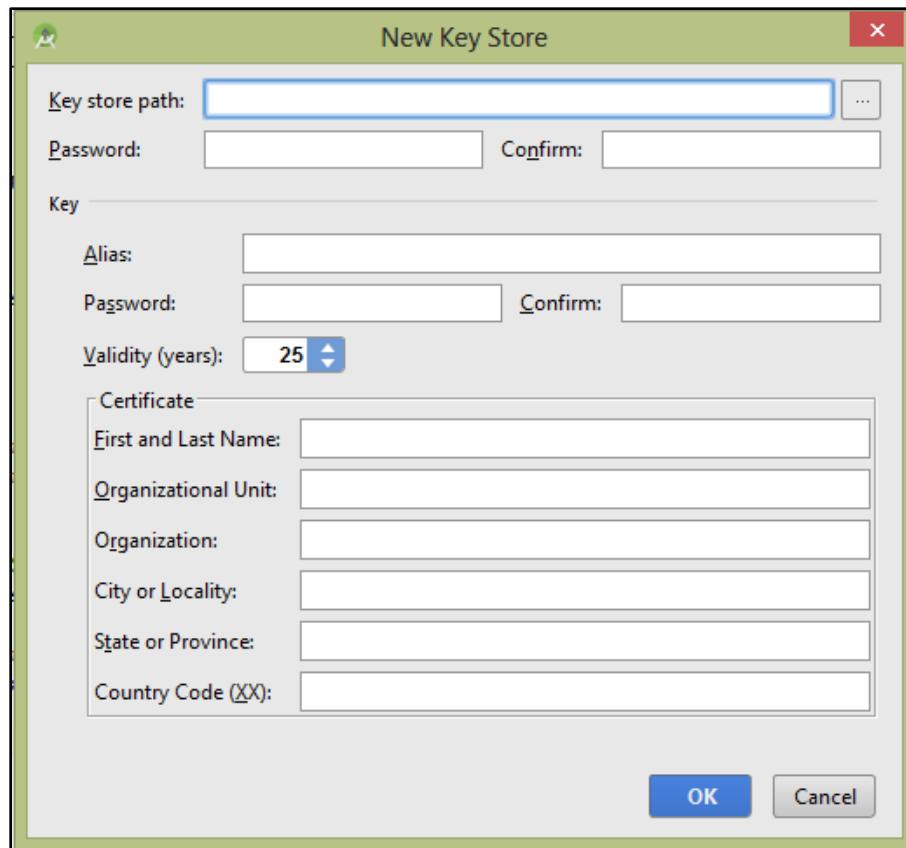
Signing an Application 2-5

- Start Android Studio and open the desired project
- Navigate to the Build Menu and select Generate Signed APK
- The Generate Signed APK dialog box is displayed as shown in the following figure:

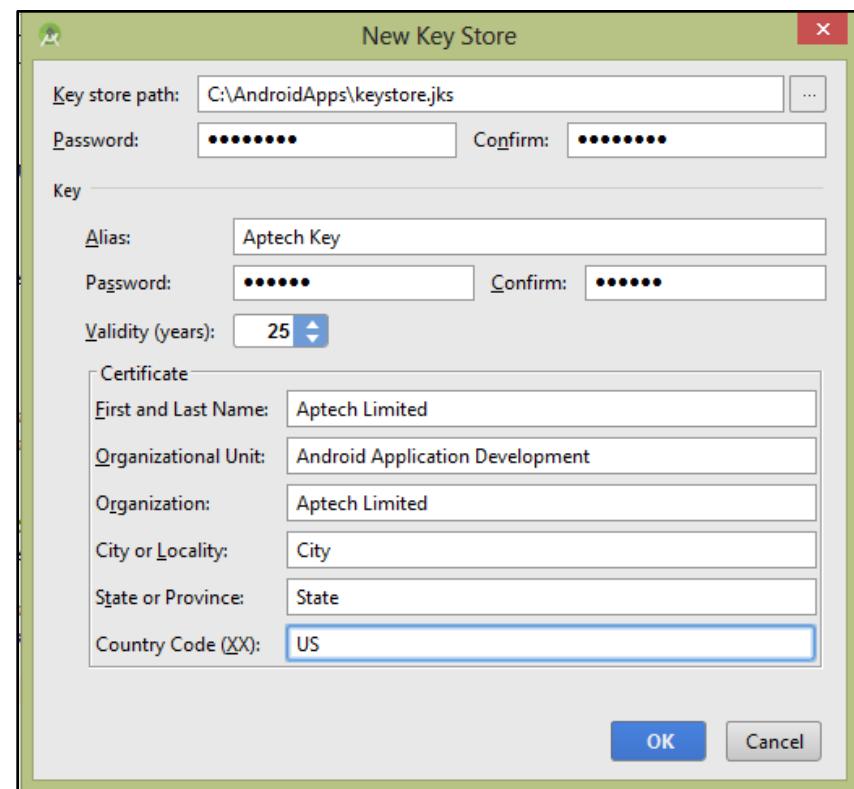


Signing an Application 3-5

- If it is the first time signing, click Create new to open the New Key Store dialog box as shown in the following figure:

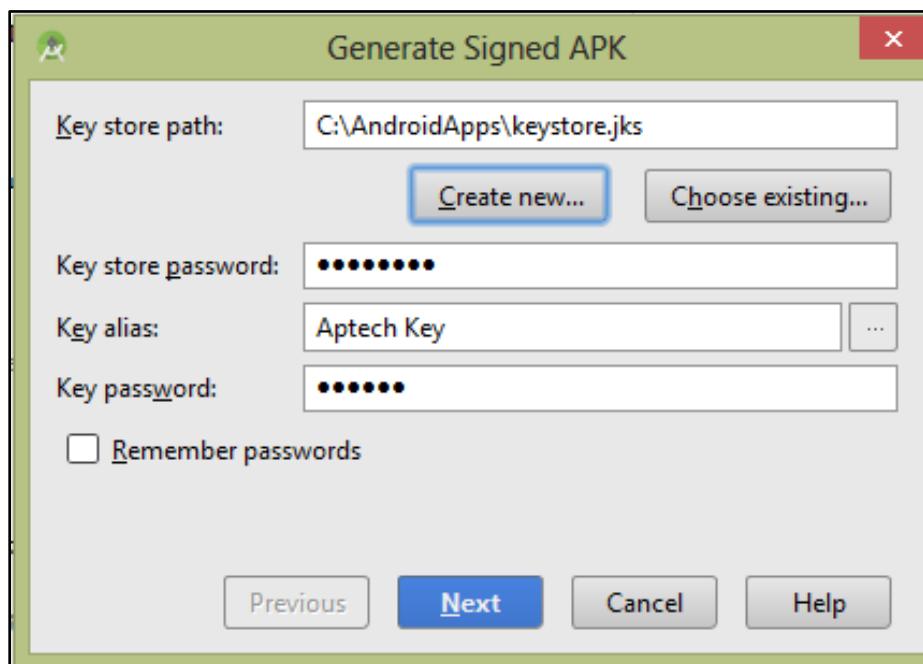


- Enter the details as shown in the following figure:



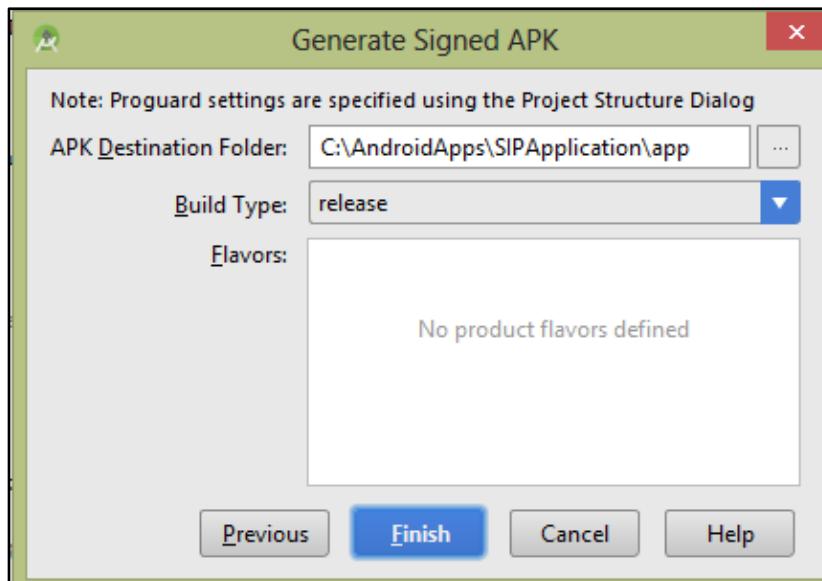
Signing an Application 4-5

- Click OK to return to the Generate Signed APK dialog box with the details already entered as shown in the following figure:



Signing an Application 5-5

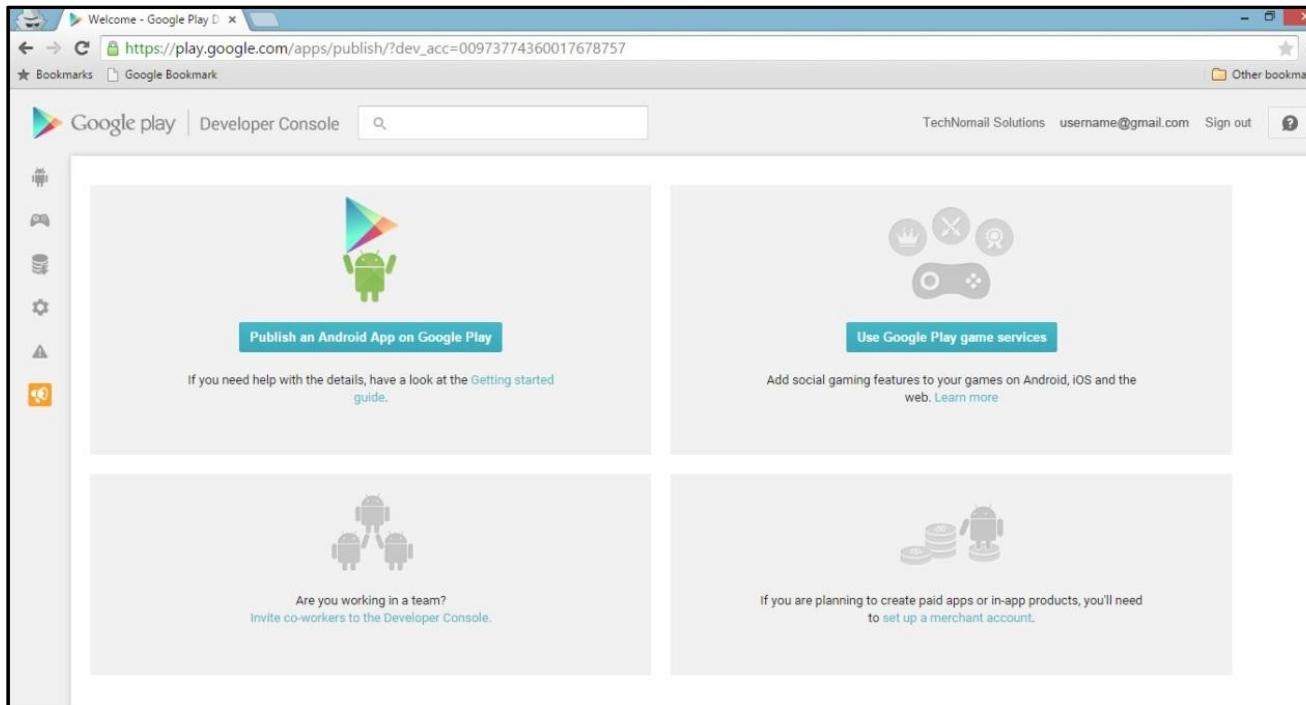
- Click Next to display the Destination dialog box as shown in the following figure:



- Click Finish. A new app-release.apk file is generated by Android Studio in the specified target directory. This .apk file can be renamed and uploaded to the Play Store

Publishing Android Application to the Play Store 1-13

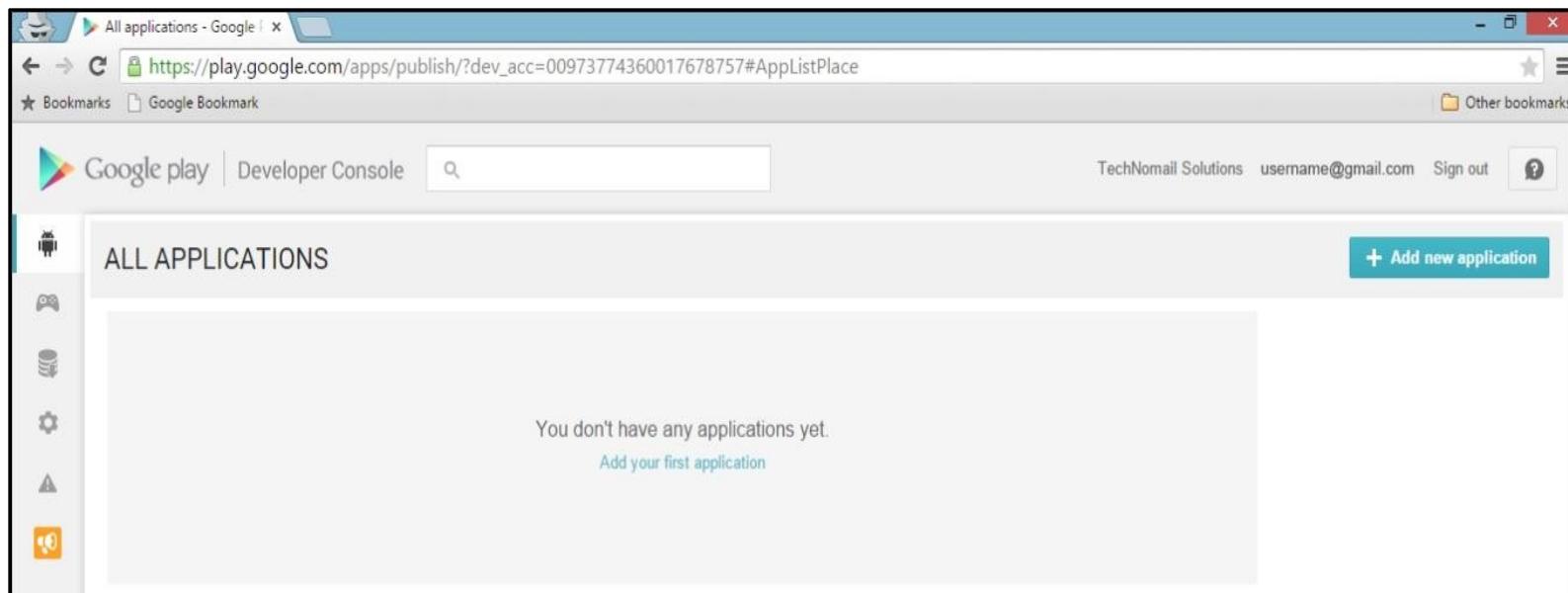
- Create a Signed .apk file
- Sign in to the Play Store Account as shown in the following figure:



- URL - <https://play.google.com/apps/publish/>

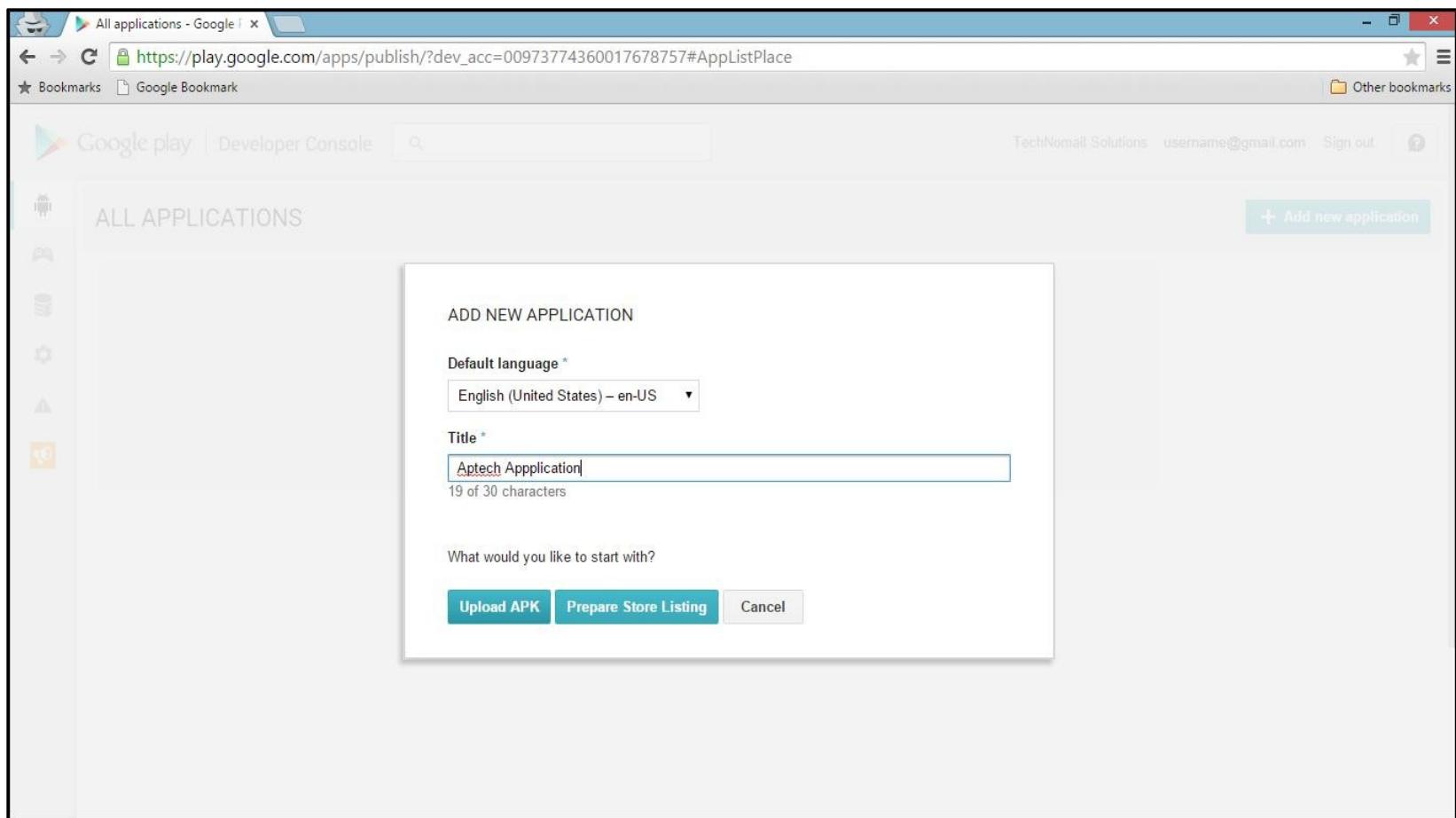
Publishing Android Application to the Play Store 2-13

- Select the 'ALL APPLICATIONS' tab from the left pane. Add your application to play store by clicking Add new application as shown in the following figure:



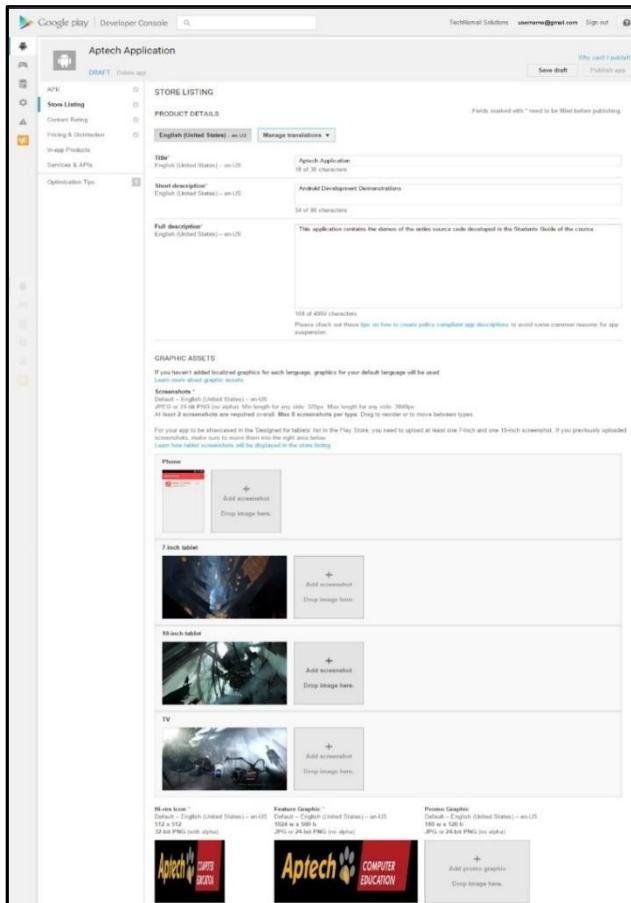
Publishing Android Application to the Play Store 3-13

- The Add New Application screen is displayed as shown in the following figure:



Publishing Android Application to the Play Store 4-13

- Store listing displays a page which allows the developer to enter the description of the application, Promo text, Recent changes, and so on as shown in the following figure:



Publishing Android Application to the Play Store 5-13

- Screen shots of the application is taken using either Emulator or Real device and uploaded as shown in the following figure:

GRAPHIC ASSETS

If you haven't added localized graphics for each language, graphics for your default language will be used.
[Learn more about graphic assets.](#)

Screenshots *
Default – English (United States) – en-US
JPEG or 24-bit PNG (no alpha). Min length for any side: 320px. Max length for any side: 3840px.
At least 2 screenshots are required overall. Max 8 screenshots per type. Drag to reorder or to move between types.

For your app to be showcased in the 'Designed for tablets' list in the Play Store, you need to upload at least one 7-inch and one 10-inch screenshot. If you previously uploaded screenshots, make sure to move them into the right area below.
[Learn how tablet screenshots will be displayed in the store listing.](#)

Phone

+ Add screenshot
Drop image here.

Publishing Android Application to the Play Store 6-13

- The Application type and Category helps the developer to specify the type of application that is getting uploaded as shown in the following figure:

CATEGORIZATION

Application type *	Applications
Category *	Education
Content rating *	Everyone
Learn more about content rating.	
New content rating *	You need to fill a rating questionnaire and apply a content rating .

Publishing Android Application to the Play Store 7-13

- Play Store allows the developer to enter the contact information as shown in the following figure:

CONTACT DETAILS

Website

Email *
Please provide an email address where you may be contacted. This address will be publicly displayed with your app.

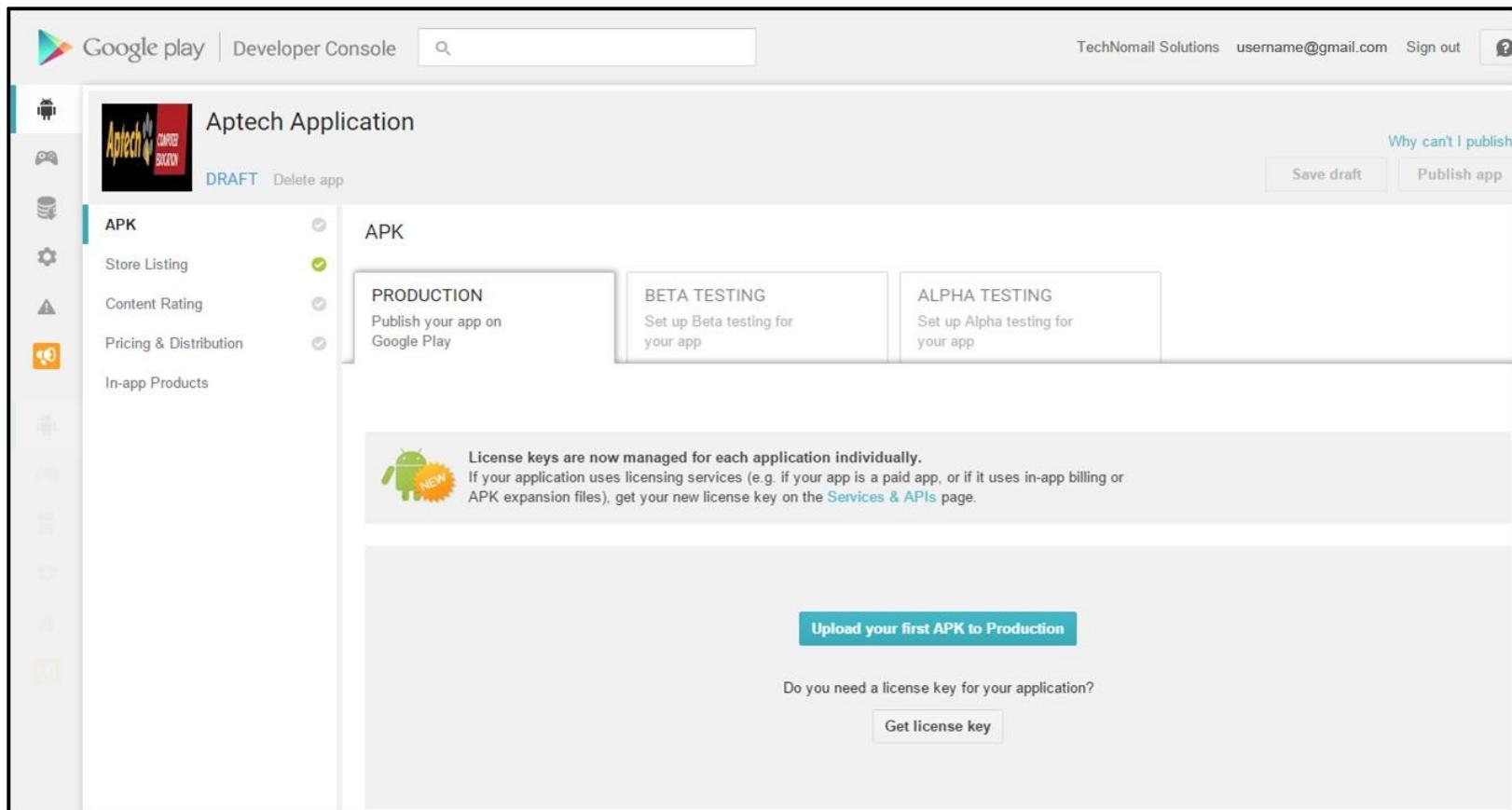
Phone

PRIVACY POLICY *
If you wish to provide a privacy policy URL for this application, please enter it below.

Privacy Policy
 Not submitting a privacy policy URL at this time. [Learn more](#)

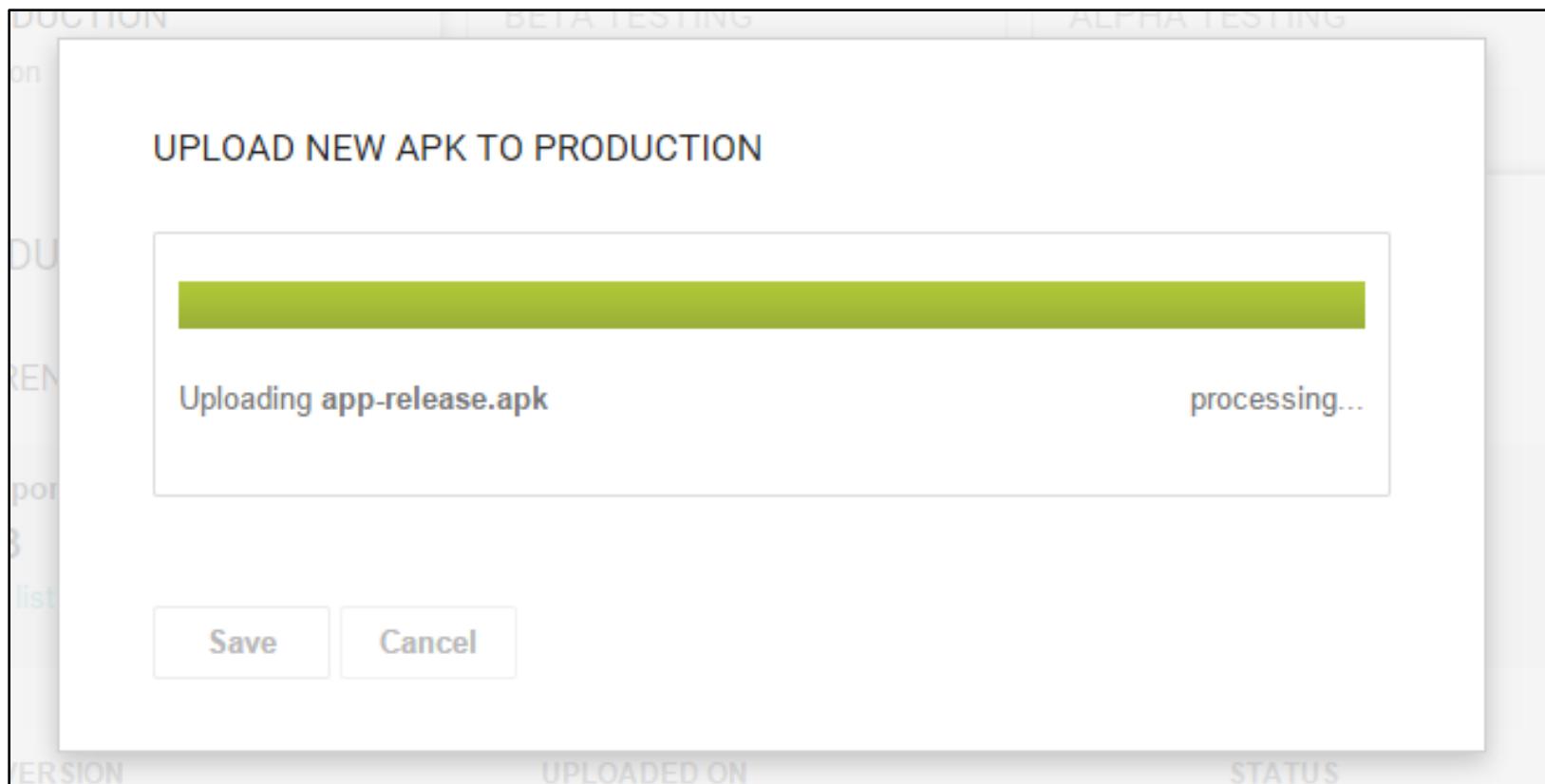
Publishing Android Application to the Play Store 8-13

- Finally, the developer needs to click ‘Upload your first APK to Production’ as shown in the following figure:



Publishing Android Application to the Play Store 9-13

- The developer needs to navigate to the desired folder and select the .apk file and the uploading process starts as shown in the following figure:



Publishing Android Application to the Play Store 10-13

- The developer needs to complete a content rating questionnaire before the application is published. Click the Content Rating tab as shown in the following figure:

The screenshot shows the Google Play Developer Console interface. At the top, there's a navigation bar with the Google Play logo, 'Developer Console', a search bar, and user information ('TechNomail Solutions username@gmail.com Sign out'). Below the navigation is a toolbar with icons for APK, Store Listing, Content Rating (which is highlighted in blue), Pricing & Distribution, In-app Products, Services & APIs, and Optimization Tips. The main content area displays the app details for 'Aptech Application' (aptech.com.sipapplication) in draft mode. A large green checkmark indicates that the APK and Store Listing are ready. The 'Content Rating' tab is selected, showing a sub-section titled 'CONTENT RATING'. It explains the Google Play content rating system and developer responsibilities, which include completing the questionnaire for new apps and updates. It also lists how ratings are used for consumer information and legal filtering. At the bottom, it mentions the Content Rating Guidelines and a 'Continue' button. The IARC logo is visible in the bottom right corner.

Publishing Android Application to the Play Store 11-13

- Finish the questionnaire and click Save questionnaire as shown in the following figure:

The screenshot shows the 'CONTENT RATING' section of the Google Play Store questionnaire. It asks users to complete a questionnaire to calculate their app's rating. The section includes a 'REFERENCE, NEWS, OR EDUCATIONAL' category icon and a note that the app is a news, reference, or educational app. Below this, there are five sections: 'VIOLENCE', 'SEXUALITY', 'LANGUAGE', 'CONTROLLED SUBSTANCE', and 'MISCELLANEOUS'. Each section contains a question with 'Yes' and 'No' radio buttons. A green checkmark is visible in the top right corner of each section header.

CONTENT RATING

Please complete the questionnaire so that we can calculate your app rating.

REFERENCE, NEWS, OR EDUCATIONAL

App is a news, reference, or educational app. [Edit Category](#)

VIOLENCE

Can the app contain violent material? [Learn more](#)
Please note that this question does not refer to user generated content.

Yes No

SEXUALITY

Can the app contain sexual material or nudity (except in a natural or scientific setting)? [Learn more](#)
Please note that this question does not refer to user generated content.

Yes No

LANGUAGE

Can the app contain any potentially offensive language? [Learn more](#)
Please note that this question does not refer to user generated content.

Yes No

CONTROLLED SUBSTANCE

Can the app contain references to or depictions of illegal drugs? [Learn more](#)
Please note that this question does not refer to user generated content.

Yes No

MISCELLANEOUS

Can users of this app interact or exchange content with other users? [Learn more](#)

Yes No

Does the app share user-provided personal information with third parties? [Learn more](#)

Yes No

Does the app share the user's current physical location to other users? [Learn more](#)

Yes No

Does the app allow users to purchase digital goods? [Learn more](#)

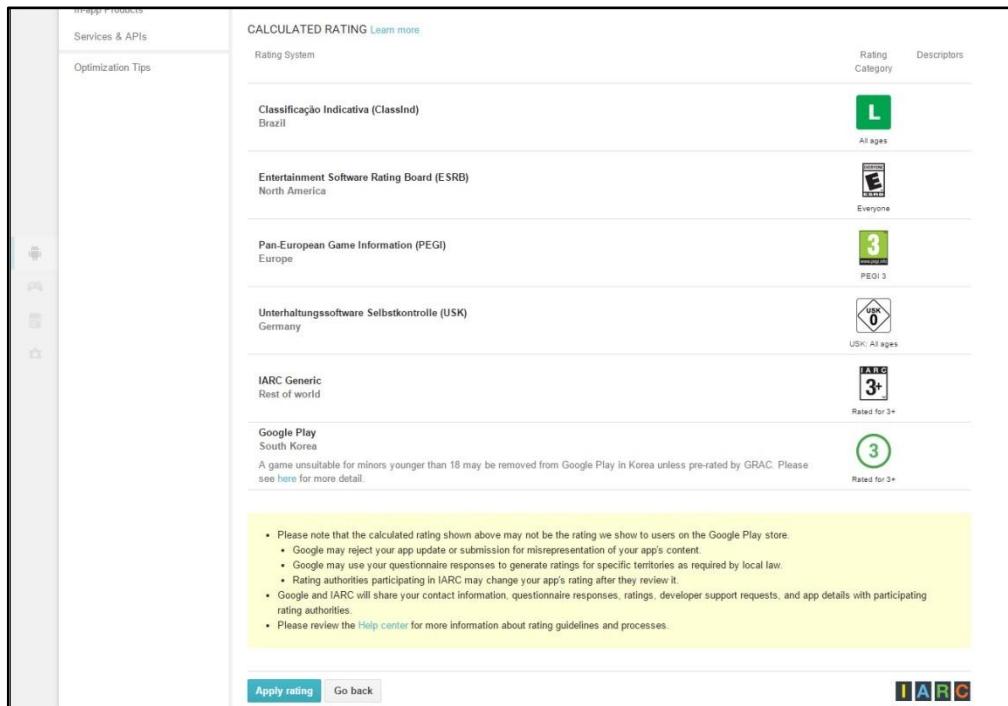
Yes No

[Calculate rating](#) [Save questionnaire](#)

IARC

Publishing Android Application to the Play Store 12-13

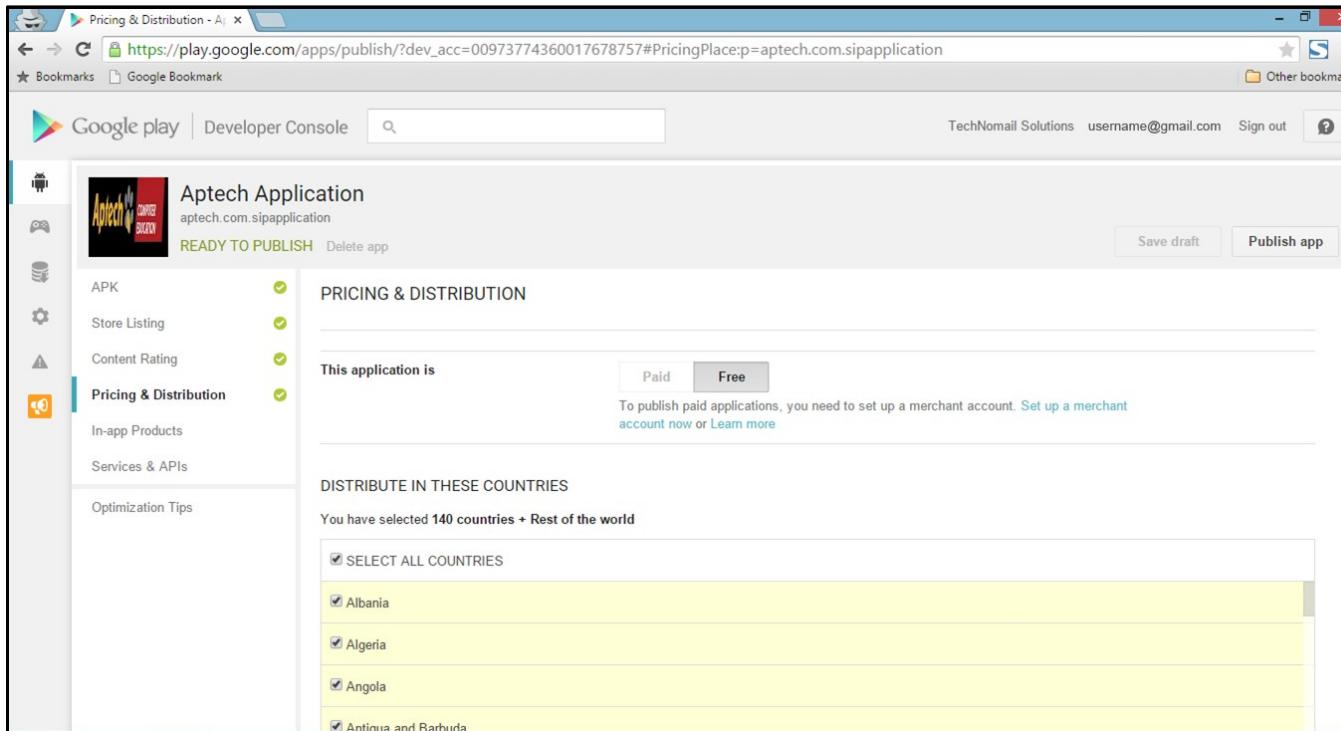
- Click Calculate rating. The confirmation screen is displayed as shown in the following figure:



- Click Apply Rating

Publishing Android Application to the Play Store 13-13

- Click Pricing & Distribution tab in the left pane of the screen as shown in the following figure:



- Once all these steps are completed, the developer is ready to publish the application. Click Publish app. The App is published on the Play Store

◆ Setup a Developer Page

- ❖ Play Store allows the developer to set up a developer page which displays developer/publisher information and promotional material
- ❖ This page can help to increase visibility and priority of the application in search results

◆ Upload and Publish for Testing

- ❖ Play Store allows the developer/publisher to upload and publish app in Alpha or Beta testing phase before making it available to all users
- ❖ Rating is also disabled during these phases
- ❖ Participants can be added from within the organization duri

◆ Revenue Stream

- ❖ Choose the right price
- ❖ Free Trail and Limited Versions



Marketing and Promoting

- ◆ Quality Assurance
- ◆ Customer Support
- ◆ Google Adwords
- ◆ Setup a Website
- ◆ Reputation and Reviews



Summary

- ◆ The Google Play Store (formerly Android market) is known to be the very own repository of Google for Android applications
- ◆ Play Store also sells music, movies, e-Books, and audio books
- ◆ In order to publish applications on the play store, a Google Play Developer Console account is needed. A payment of USD 25 is required for this
- ◆ The application needs to be signed and tested before uploading to the Play Store
- ◆ Marketing and good reputation are key factors for the success of an application
- ◆ In order to maximize the audience, the market shares and distributions need to be researched thoroughly