

CHAPTER 13: MOBILE APPLICATION DEVELOPMENT

13.1 Introduction

This unit specifies competencies required to develop computer program. It involves Identifying Mobile application concepts, identifying mobile application development environment, identifying Application Design Issues, actual Development of mobile application, testing of the developed mobile application and Publishing and Commercializing the developed Application.

13.2 Performance Report

- Identified Mobile application development platforms
- Identified Mobile application development frameworks and tools
- Installed and Configured Appropriate mobile development software
- Built the project prototype into a debuggable APK that can be installed to an emulator or Android powered device.
- Tested and Debugged the Application
- Published the Application

13.3 Learning Outcome

13.3.1 List of the Learning Outcomes

These are the key learning outcomes, which make up workplace function:

- Identify mobile application concepts
- Identify mobile application development environment
- Identify application design issues
- Develop mobile application
- Test the developed mobile application
- Publish and commercialize the developed application

13.3.2 Learning Outcome 1: Identify mobile application concepts

13.3.2.1 Learning Activities

The following are the performance criteria:

- Definition of Mobile application is done
- Types of mobile applications are identified
- Mobile application development platforms are identified
- Mobile application development approaches are identified
- Reasons for mobile application development are identified.

Trainees to demonstrate knowledge in relation to:

- Definition of Mobile application

- Types of Mobile applications: Hybrid, Native
- Mobile application development approaches: Native, Hybrid Native, Hybrid web, Progressive web
- Reasons for mobile application

13.3.2.2 Information Sheet

A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer.

There are 3 types of apps:

1. Native apps

iOS on Objective-C or Swift

Android on Java

Windows Phone on Net

2. Hybrid apps for all platforms altogether with Xamarin, React Native, Ionic, and Angular Mobile Sencha Touch etc.

3. Web apps as responsive versions of website to work on any mobile device.

Read: Types of application: <https://thinkmobiles.com/blog/popular-types-of-apps/>

Read: Approaches of apps: <https://www.telerik.com/docs/default-source/whitepapers/choose-right-approach-mobile-app-developmentbb581d10116543e79a9febdb187fd0a3.pdf?sfvrsn=0>

Following are the reasons for building a mobile application:

Generate additional sales, reduce cost to serve, reduce marketing cost, enhance the customer experience, broaden your market coverage, create stickiness, deliver speed and get feedback.

Read: Reason to build mobile app:

<http://www.matchboard.com.au/MatchBoard/Forms/Article/App%20Appeal%20%2010%20Great%20Reasons%20to%20build%20a%20Mobile%20App.aspx>

13.3.2.3 Self-Assessment

- i. Give 5 reasons to build successful mobile applications?
- ii. Define mobile application.
- iii. Explain types of mobile application?
- iv. Language, which is supported by Android for application development?
 - A. PHP
 - B. VB.NET
 - C. Java
 - D. C++
- v. Each application can have zero or more activities. True or False? Justify in short.
- vi. Android is based on Linux for the following reason.

- A. Portability
 - B. Security
 - C. Networking
 - D. All of these
- vii. What was the main reason for replacing the Java VM with the Dalvik VM when the project began?
- A. Java VM ran too slow
 - B. Java VM was too complicated to configure
 - C. Java virtual machine was not free
 - D. There was not enough memory capability
- viii. From a phone manufacturer's point of view, what makes Android so great?
- A. Aside from some specific drivers, it provides everything to make a phone work
 - B. It allows them to compete with Apple's iPhone
 - C. It makes the hardware work better
 - D. It allows users to create apps, generating revenue for the companies

13.3.2.4 Tools, Equipment, Supplies and Materials

Software, Android SDK, Integrated Development Environment (IDE)

11.3.2.5 References

- <https://mashable.com/article/build-mobile-apps/>
- <http://www.matchboard.com.au/MatchBoard/Forms/Article/App%20Appeal%20%2010%20Great%20Reasons%20to%20build%20a%20Mobile%20App.aspx>
- <https://www.telerik.com/docs/default-source/whitepapers/choose-right-approach-mobile-app-developmentbb581d10116543e79a9febdb187fd0a3.pdf?sfvrsn=0>
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc, 2015
- Beginning Android Programming With Android Studio (4th edition) authored by Jerome DiMarzio published by Wrox;, 2016

13.3.3 Learning Outcome 2: Identify mobile application development environment

13.3.3.1 Learning Activities

The following are the performance criteria:

- Mobile Application Architecture and Design is identified
- Mobile application development frameworks and tools are identified
- Techniques and methodologies for mobile application development are introduced

Trainees to demonstrate knowledge in relation to:

- Definition of Mobile Application Development Architecture
- Mobile Application Development Architecture: Stack, Linux Kernel, DVM – Dalvik virtual Machine, SDK
- Reference Architecture; Model view presenter, Wildlife

- Mobile development frameworks; Native script, Flutter, React Native
- Mobile application development tools: Integrated Development Environment (IDE), Graphic User Interface (GUI), Emulator, Android SDK

13.3.3.2 Information Sheet

Mobile app architecture is a set of techniques and patterns used to develop fully structured mobile applications based on industry and vendor specific standards. While formulating the app architecture, the procedures that work on the wireless mobile device like smartphones and tablets are also taken into consideration.

An application stack is a suite or set of application programs that help in performing a certain task.

The Linux kernel is a free and open-source, monolithic, Unix-like operating system kernel. The Linux family of operating systems is based on this kernel and deployed on both traditional computer systems.

Dalvik is a discontinued process virtual machine in Google's Android operating system that executes applications written for Android.

A Software Development Kit (SDK) is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

Reference architecture is a document or set of documents to which a project manager or other interested party can refer for best practices.

Read: Model view: <https://www.raywenderlich.com/7026-getting-started-with-mvp-model-view-presenter-on-android>

Watch: Introduction to model view: <https://youtu.be/NNZO9zA0yEY>

Native script this has been ranked as the top frameworks for the development of the hybrid apps. Telerik, a Bulgaria based software company, which supports this framework and the creator is Progress. The apps using the Native Scripts utilize the same APIs similar to using Xcode or the Android Studio. You will get sufficient help in the form of tutorials from the official website.

Flutter was launched by Google and is an open-source mobile application development SDK. It is a popular cross-platform app development and is written in the Dart language.

React Native is basically an open-source framework offering ample support to the IDEs and the other mobile app development tools.

Read: Mobile Development framework: <https://www.mindinventory.com/blog/mobile-app-development-framework-2019/>

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

The graphical user interface (GUI) is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

A mobile emulator is a resource for emulating or simulating a mobile device or smartphone environment. It allows developers to test URLs or other technologies on a mobile device's operating system and display interface.

Read: Mobile applications:

<https://www.cs.cmu.edu/~bam/uicourse/830spring09/BFeiginMobileApplicationDevelopment.pdf>

13.3.3.3 Self-Assessment

- i. Explain mobile architecture?
- ii. Explain briefly mobile development architecture?
- iii. What is emulator?
- iv. What is the name of the program that converts Java byte code into Dalvik byte code?
 - A. Mobile Interpretive Compiler (MIC)
 - B. Dex compiler
 - C. Dalvik Converter
 - D. Android Interpretive Compiler (AIC)
- v. Required folder when Android project is created.
 - A. Build
 - B. Build/
 - C. Bin
 - D. Bin/
- vi. Which of the following is not a component of an APK file?
 - A. Dalvik executable
 - B. Native Libraries
 - C. Resources
 - D. All of these are components of the APK
- vii. Select a component which is NOT part of Android architecture.
 - A. Android framework
 - B. Libraries
 - C. Linux kernel
 - D. Android document

13.3.3.4 Tools, Equipment, Supplies and Materials

Software, Emulator, Mobile Device, Android SDK

13.3.3.5 References

- <https://www.techopedia.com/definition/6010/application-stack>
- <https://www.cs.cmu.edu/~bam/uicourse/830spring09/BFeiginMobileApplicationDevelopment.pdf>
- <https://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>
- <https://www.techopedia.com/definition/30676/mobile-emulator>
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc., 2015
- Beginning Android Programming With Android Studio (4th edition) authored by Jerome DiMarzio published by Wrox, 2016

13.3.4 Learning Outcome 3: Identify application design issues

13.3.4.1 Learning Activities

The following are the performance criteria:

- Mobile development lifecycle is explained
- Overarching Design principles and Guidelines are explained
- Mobile application navigation patterns are identified
- User interface design is explained

Trainees to demonstrate knowledge in relation to:

- Mobile development lifecycle, Setup, Develop, Test and Debug, Publish
- Overarching Design principles and Guidelines, Platform, Customer Benefit, Device, Scalability etc.
- Mobile application Navigation Patterns - Hamburger Menu, Tab bar, Gesture based

11.3.4.2 Information Sheet

The lifecycle of mobile development is largely no different than the SDLC for web or desktop applications.

Read: Mobile development lifecycle: <https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/introduction-to-mobile-sdlc>

Hamburger Menu: Screen space is a precious commodity on mobile, and the hamburger menu (or side drawer) is one of the most popular mobile navigation patterns for helping you save it. The drawer panel allows you to hide the navigation beyond the left edge of the screen and reveal it only upon a user's action.

The tab bar pattern is inherited from desktop design. It usually contains relatively few destinations, and those destinations are of similar importance and require direct access from anywhere in the app.

Read: Mobile application navigation patterns:

<https://www.smashingmagazine.com/2017/05/basic-patterns-mobile-navigation/>

13.3.4.3 Self-Assessment

- i. Explain lifecycle of mobile development?
- ii. Explain briefly navigation patterns and give example?
- iii. Universal front end to database applications that connects to back end through Internet, are the
 - A. Web browsers
 - B. Applications
 - C. Tools
 - D. Software programs
- iv. _____ is largely no different than the SDLC for web or desktop applications. The lifecycle of mobile development
 - A. Android application
 - B. Mobile development
 - C. None of the above
 - D. Both of the above

13.3.4.4 Tools, Equipment, Supplies and Materials

Graphic user interface GUI, Buildfire, Ionic, software,

13.3.4.5 References

- <https://www.smashingmagazine.com/2017/05/basic-patterns-mobile-navigation/>
- <https://www.corephp.com/blog/mobile-app-development-lifecycle/>
- <https://medium.com/@soulless/top-8-mobile-navigation-menu-design-for-your-inspiration-8a2d925bffc0>
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc 2015
- Beginning Android Programming With Android Studio authored by Jerome DiMarzio published by Wrox; 4 edition 2016

13.3.5 Learning Outcome 4: Develop mobile application

13.3.5.1 Learning Activities

The following are the performance criteria:

- Appropriate mobile development software is installed
- Creation of the project structure is done by project wizard
- Configuration of the AndroidManifest.XML file is done

- Resources are defined in XML
- Framework components are defined
- SQL lite database is introduced
- Configuration of the Google play SDK is done
- Project prototype is created as per the scope
- Build the project prototype into a debuggable APK
- That can be installed to an emulator or Android powered device

Trainees to demonstrate knowledge in relation to:

- Mobile Application development software: Integrated Development Environment (IDE) , Android SDK Androidmanifest.XML Configuration, Resources defined in XML, Res/Layout, Res/Menu, Res/Value, Res/Drawable
- Framework component: Activity, Services, Broadcast receiver
- Content provider: SDK Configuration, Building and setting up of the Application

11.3.5.2 Information Sheet

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest presents essential information about the application to the Android system; information the system must have before it can run any of the application's code.

Read: App manifest overview

<https://developer.android.com/guide/topics/manifest/manifest-intro>

<https://stuff.mit.edu/afs/sipb/project/android/docs/guide/topics/manifest/manifest-intro.html>

XML files that define a user interface **layout**. They are saved in res/layout/ and accessed from the **R.layout** class.

XML files that define application **menus**, such as an Options Menu, Context Menu, or Sub Menu. They are saved in res/menu/ and accessed from the **R.menu** class.

Read: Android resources: https://www.tutorialspoint.com/android/android_resources.htm

An activity is the entry point for interacting with the user. It represents a single screen with a user interface. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails.

A service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes.

A broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast

announcements. Because broadcast receivers are another well-defined entry into the app, the system can deliver broadcasts even to apps that aren't currently running.

A content provider manages a shared set of app data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your app can access. Through the content provider, other apps can query or modify the data if the content provider allows it.

Read: SDK Configuration:

https://www.ntu.edu.sg/home/ehchua/programming/android/Android_HowTo.html

Read: Building your first app: <https://developer.android.com/training/basics/firstapp>

13.3.5.3 Self-Assessment

- i. What is activity?
- ii. What is service?
- iii. Explain briefly Resources defined in XML
- iv. Differentiate XML and SDK?
- v. How Long Does It Take To Build An App?
- vi. What Is A Resource?
- vii. Create Android program convert Temperature from Celsius to Fahrenheit and Fahrenheit to Celsius?
- viii. What happen if we all execute () more than once in Async Task
 - A. Nothing will happen
 - B. It will create those many threads
 - C. It will throw an error
 - D. It will throw an execution
- ix. ADB is located in platform-tools folder under android-SDK
 - A. True
 - B. False
- x. Can we create a different layout files for portrait and landscape mode?
 - A. One cannot create different layout files for portrait and landscape mode
 - B. One can create different layout files for portrait and landscape mode
- x. A task can have multiple apps?
 - A. True
 - B. False

13.3.5.4 Tools, Equipment, Supplies and Materials

Software, Computer, Mobile device, Graphic user interface, Emulator

13.3.5.5 References

- https://www.tutorialspoint.com/android/android_resources.htm
- <https://developer.android.com/guide/components/fundamentals>

- https://www.ntu.edu.sg/home/ehchua/programming/android/Android_HowTo.html
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc, 2015
- Beginning Android Programming With Android Studio (4th edition) authored by Jerome DiMarzio published by Wrox, 2016

13.3.6 Learning Outcome 5: Test and developed mobile application

13.3.6.1 Learning Activities

The following are the performance criteria:

- Testing techniques and procedures are identified
- Debugging techniques are identified
- Debugging of the application is done

Trainees to demonstrate knowledge in relation to:

- Testing techniques and procedures: Usability testing, Installation testing, Cloud testing etc.
- Definition of debugging
- Debugging techniques

13.3.6.2 Information Sheet

Mobile application testing is a process by which application software developed for handheld mobile devices is tested for its functionality, usability and consistency. Mobile application testing can be an automated or manual type of testing.

Key challenges for mobile application testing

- **Must be downloadable:** The application must be obtainable for the particular platform, generally from an app store.
- **Diversity in mobile platforms/OSes:** There are different mobile operating systems in the market. The major ones are Android, iOS, and Windows Phone. Each operating system has its own limitations.
- **Device availability:** Access to the right set of devices when there is an ever-growing list of devices and operating system versions is a constant mobile application testing challenge. Access to devices can become even more challenging if testers are spread across different locations.
- **Mobile network operators:** There are over 400 mobile network operators in the world; some are CDMA, some GSM, and others use less common network standards like FOMA, and TD-SCDMA.
- **Scripting:** The variety of devices makes executing a test script (scripting) a key challenge. As devices differ in keystrokes, input methods, menu structure and display properties single script does not function on every device.

- **Test method:** There are two main ways of testing mobile applications: testing on real devices or testing on emulators. Emulators often miss issues that can only be caught by testing on real devices, but because of the multitude of different devices in the market, real devices can be expensive to purchase and time-consuming to use for testing.
- **Compatibility:** It is necessary to test the compatibility; suppose an application can work on the high resolution and it doesn't work on the lower resolution.
- **Variety of mobile devices:** Mobile devices differ in screen input methods (QWERTY, touch, normal) with different hardware capabilities.

Testing applications on mobile devices is more challenging than testing web apps on the desktop due to:

- Different range of mobile devices with different screen sizes and hardware configurations like a hard keypad, virtual keypad (touch screen) and trackball etc.
- Wide varieties of mobile devices like HTC, Samsung, Apple and Nokia.
- Different mobile operating systems like Android, Symbian, Windows, Blackberry and IOS.
- Different versions of operation system like iOS 5.x, iOS 6.x, BB5.x, BB6.x etc.
- Different mobile network operators like GSM and CDMA.
- Frequent updates – (like Android- 4.2, 4.3, 4.4, iOS-5.x, 6.x) – with each update a new testing cycle is recommended to make sure no application functionality is impacted.

Types of mobile application testing

- **Functional testing** ensures that the application is working as per the requirements. Most of the tests conducted for this is driven by the user interface and call flow.
- **Laboratory testing**, usually carried out by network carriers, is done by simulating the complete wireless network. This test is performed to find out any glitches when a mobile application uses voice and/or data connection to perform some functions.
- **Performance testing** is undertaken to check the performance and behavior of the application under certain conditions such as low battery, bad network coverage, low available memory, simultaneous access to the application's server by several users and other conditions. Performance of an application can be affected from two sides: the application's server side and client's side. Performance testing is carried out to check both.
- **Memory leakage testing:** Memory leakage happens when a computer program or application is unable to manage the memory it is allocated resulting in poor performance of the application and the overall slowdown of the system. As mobile devices have significant constraints of available memory, memory leakage testing is crucial for the proper functioning of an application.
- **Interrupt testing:** An application while functioning may face several interruptions like incoming calls or network coverage outage and recovery. The different types of interruptions are:
 - Incoming and outgoing SMS and MMS
 - Incoming and outgoing calls
 - Incoming notifications

- Battery removal
- Cable insertion and removal for data transfer
- Network outage and recovery
- Media player on/off
- Device power cycle
- An application should be able to handle these interruptions by going into a suspended state and resuming afterwards.
- **Usability testing** is carried out to verify if the application is achieving its goals and getting a favorable response from users. This is important as the usability of an application is its key to commercial success (it is nothing but user friendliness). Another important part of usability testing is to make sure that the user experience is uniform across all devices. This section of testing hopes to address the key challenges of the variety of mobile devices and the diversity in mobile platforms/OS, which is also called device fragmentation. One key portion of this type of usability testing is to be sure that there are no major errors in the functionality, placement, or sizing of the user interface on different devices.
- **Installation testing:** Certain mobile applications come pre-installed on the device whereas others have to be installed by the store. Installation testing verifies that the installation process goes smoothly without the user having to face any difficulty. This testing process covers installation, updating and uninstalling of an application
- **Certification testing:** To get a certificate of compliance, each mobile device needs to be tested against the guidelines set by different mobile platforms.
- **Security testing** checks for vulnerabilities to hacking, authentication and authorization policies, data security, session management and other security standards.
- **Location testing:** Connectivity changes with network and location.
- **Outdated software testing:** Not everyone regularly updates their operating system. Some Android users might not even have access to the newest version. Professional testers can test outdated software.
- **Load testing:** When many users all attempt to download, load, and use an app or game simultaneously, slow load times or crashes can occur causing many customers to abandon your app, game, or website. In-country human testing done manually is the most effective way to test load.
- **Black box testing** doesn't include the internally coding logic of the application. The tester tests the application with functionality without peering with internally structure of the application. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.
- **Crowdsourced testing:** In recent years, crowdsourced testing has become popular as companies can test mobile applications faster and cheaper using a global community of testers. Due to growing diversity of devices and operating systems as well as localization needs, it is difficult to comprehensively test mobile applications with small in-house testing teams. A global community of testers provides easy access to different devices and platforms. A globally distributed team can also test it in multiple locations and under different network conditions. Finally, localization issues can be tested by hiring testers in

required geographies. Since real users using real devices test the application, this is more likely to find issues faced by users under real world conditions.

Basic Difference Between Mobile And Desktop Application Testing: Few obvious aspects that set mobile app testing apart from the desktop testing

- On the desktop, the application is tested on a central processing unit. On a mobile device, the application is tested on handsets like Samsung, Nokia, Apple, and HTC.
- Mobile device screen size is smaller than a desktop.
- Mobile devices have less memory than a desktop.
- Mobiles use network connections like 2G, 3G, 4G or WIFI where desktop use broadband or dial-up connections.
- The automation tool used for desktop application testing might not work on mobile applications.

Read: Mobile application development and testing:

<https://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/>

The debugger allows you to control the execution of your application by:

- Setting breakpoints.
- Suspending launched applications.
- Stepping through your code.
- Examining the contents of the variables and view them on an emulator.

The debug types available include the following (android studio):

- Auto (C or C++ code in your project)
- Java
- Native: (Available only with C/C++ code.)
- Dual: (Available only with C/C++ code.) Select if you want to switch between debugging both Java and native code.

Read: Debugging mobile application:

https://docs.oracle.com/cd/E37975_01/doc.111240/e24475/debugging.htm#ADFMF1047

<https://www.wavemaker.com/learn/hybrid-mobile/debugging-mobile-apps/>

<https://developer.android.com/studio/debug>

Read: Case study on debugging:

<https://www.objc.io/issues/19-debugging/debugging-case-study/>

13.3.6.3 Self-Assessment

- i. What is debugging?
- ii. How is mobile application testing different for different mobile?
- iii. You are to prepare a mobile application of online grocery shopping (android). What are the key parameters for testing and debugging?

- iv. _____ ensures that the application is working as per the requirements. Most of the tests conducted for the user drives this interface and call flow.
- A. Black box testing
 - B. Functional testing
 - C. Location testing
 - D. None of above
- v. _____ connectivity changes with network and location.
- A. Location testing
 - B. Installation testing
 - C. Testing
 - D. Program testing

13.3.6.4 Tools, Equipment, Supplies and Materials

Computer, software

11.3.6.5 References

- <https://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/>
- https://docs.oracle.com/cd/E37975_01/doc.111240/e24475/debugging.htm#ADFMF1047
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc., 2015
- Beginning Android Programming With Android Studio (4th edition) authored by Jerome DiMarzio published by Wrox, 2016

13.3.7 Learning Outcome 6: Publish and commercialize the developed application

13.3.7.1 Learning Activities

The following are the performance criteria:

- Application distribution through application stores is done
- Monetizing applications through mobile money APIs is done.
- Routine upgrading, and patching of the application is done.

Trainees to demonstrate knowledge in relation to:

- Application distribution through application stores
- Monetizing applications through mobile money APIs
- Upgrading and patching of the application

13.3.7.2 Information Sheet

The idea of an enterprise application store arose in response to consumer-oriented app stores, such as Apple's App Store and Google Play. By deploying corporate software through an app store, an information technology (IT) department can manage desktop, mobile, cloud and Web app software licenses (EULAs) and maintain a level of control over security.

API monetization is the process by which businesses create revenue from their application programming interfaces (APIs). APIs are the cornerstone of what is widely seen as the next iteration of business development, where having well-developed APIs establishes and maintains relationships in a digital economy. APIs are the wholesale version of a web presence, allowing others to access and integrate your data and resources into their public or private sites and applications.

Read: API Monetizing: <https://www.redhat.com/en/topics/api/what-is-api-monetization>

An update to an application can be provided to users as a Windows **Installer patch package**. A patch can contain an entire file or only the file bits necessary to update part of a file.

A patch is a software update comprised code inserted (or patched) into the code of an executable program. Typically, a patch is installed into an existing software program. Patches are often temporary fixes between full releases of a software package. Patches may do any of the following:

- Fix a software bug
- Install new drivers
- Address new security vulnerabilities
- Address software stability issues
- Upgrade the software

13.3.7.3 Self-Assessment

- i. What is API?
- ii. Define application distribution through application stores
- iii. _____ is the process by which businesses create revenue from their application programming interfaces (APIs).
 - A. API terminal
 - B. API monetization
 - C. API content
 - D. None of above
- iv. You are mobile developer and your client ask you to cerate for him a e-commerce mobile application and he wanted users to pay through mobile money how will you do it?
- v. _____ is a software update comprised code inserted into the code of an executable program.
 - A. Enterprise application
 - B. Patch
 - C. Update
 - D. Upgrade
 - E. None of above

13.3.7.4 Tools, Equipment, Supplies and Materials

SolarWinds Patch Manager, LANDesk Patch Manager, REST-Assured

11.3.7.5 References

- <https://docs.microsoft.com/en-us/windows/desktop/msi/patching-and-upgrades>
- <https://www.redhat.com/en/topics/api/what-is-api-monetization>
- Android Application Development All-in-One For Dummies authored by Barry A. Burd published by John Wiley & Sons Inc 2015
- Beginning Android Programming With Android Studio authored by Jerome DiMarzio published by Wrox; 4 edition 2016

easytvvet.com