INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES

DEVI AHILYA VISHWA VIDHYALAYA,

(IIPS)INDORE

Software test plan
On the
Open Source Software
“ZERO Qmart”

Submitted to

Mr. shaligram prajapat sir

Submitted by

Devendra rai (IT-2k19-14)
Harsh joshi(IT-2k19-18)

SOFTWARE TEST PLAN OF “ZERO QMART

Introduction to Programming Languages, IDE’S, Tools and Technologies

Java

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. As mentioned in,

[3] Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, Java has come into evolution by addressing their drawbacks [4] .It is one of the open source projects [5] that could be easily installed in our machine. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability, it has a platform independence feature including any virtual machines(Write Once Read Everywhere), as in no need to write the code for different OS as the Java Compliers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It is more secured as the compilers are designed efficiently to figure out any kind of errors.

IDE’s, Tools and Technologies:

Android Studio

Android Studio [6] is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer’s job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an ‘S’, it automatically senses everything that starts with an ‘S’ and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK’s, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or anyother compatibility issues and to download any component that is required.

AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as itmight be little slower when compared to real device.

SQLite Database:

Android also supports inbuilt database which is Android SQLite to develop any small applications and perform any CRUD (Create, Update, and Delete) operations. As it is not flexible enough to support substantial number of data, for complex applications we are using other external databases.

Eclipse IDE

Android applications could be developed in Eclipse IDE in which we can compile, run, debug and deploy using ADT (Android Development Tools). In this project, I have used Eclipse IDE to create Java SOAP Web Services for connecting the database and used Android Studio to develop the application.

MySQL Database

In this project, I have used MySQL database to store the data. This is one of the popular open source relational database management systems. We can perform all DDL, DML, DCL operations using this database. This also supports different programming language applications. The applications could connect the database using separate ways which includes PHP myadmin WAMP, LAMP, Web Services. To use this database, we should first download, install and configure the MySQL instance in our machine. While configuring, we should give access credentials which could be used further whenever youopen the MySQL shell.

Apache Tomcat Server

It is one of the open source enterprise Java servlet containers which are widely used in many applications to maintain their database. The database is kept in this remote server and could be accessed whenever needed through MySQL shell.

Soap UI

I have used SOAP Web Services in this application to connect the application with database. SOAP Web Services are implemented by writing the service as a method and we should call this method in corresponding Android java files, the page where we want to call and display the details. The web service method could be tested in SOAP UI [7] bypassing the parameters or fields in the database. The service method returns success if it returns the actual result and fails if it does not. In this way, we can test the service before calling it in Android Java source files. The UI is very interactive and easy to use.

Security and Permissions

in Android Security notions in Android are quite high. Whenever a new Android Application is created, a unique user and group ID. This makes the maintenance of the application in an easier way to avoid any security or privacy issues. As the application is created uniquely, it becomes private and no one can access other’s applications. Permissions are another important concept which is included in AndroidManifest.XML configuration file. This is required if the application wants to access the external features.For ex, if the application wants to access the Internet, Camera or it could be any feature,it requires permissions. It is included within the tags as it is an XML file. Permissions are automatically created for the basic applications at the time when we create the application. If the app uses higher level API or SDK we must explicitly mention the permissions inside uses-permissions tag to access the features or components.

**END OF STP