Al Imam Mohammad Ibn Saud Islamic University
College of Computer and Information Sciences
Computer Science Department

CS392 Software Engineering

Workshop Document

(Alinma Bank Report)

First semester 1443/2021

Date : 4/11/2021

Section: 371

Group members' names	Students IDs
Manar.M Alqabbani	440022244
Dana.Z Alyami	440022746
Leen.S Alaboudi	440020514
Mehaf.Kh Allazzam	440021473

Instruction: Dr.Lamees Alhazzaa

Contents

1.Introduction

1.1 Overview

A quality Code Review is a specialized activity that seeks to identify categories of risk within a particular code base so that mitigation techniques may be created, rather than identifying types of flaws that exist inside the code.

1.2 Purpose

The purpose of this report is to describe the code quality for "Alinma Bank".
It will explain the purpose of analyzing the code. This report will be proposed to the company.

1.3 Goal

The goal of this code review is to discover the types of flaws that exist inside Alinma code base and to assist the developers in improving the application's code quality.

1.4 Lessons learned

The first step of writing this report was to install and download the code analysis tool, which was our first experience with it, which proved there were some weaknesses that resulted in the quality being reduced and then we tried to find solutions for each of these problems, we then learned how to apply what we studied by following the classes and finding errors manually, and during analysis and research, we became familiar with different types of security and the problems that fall under them.

2. Scope

2.1 Mobile App Analysis

2.1.1 Steps we follow to extract the mobile app data

- Install docker to use with Mobsf.
- Upload the source code using apk tool.
- Upload the apk file in Mobsf to get code analysis.

2.1.2 Clean Code Analysis

After reviewing the codes, and according to our study of the Quality code, we found some mistakes, which negatively affected the quality of the code, and it will increase the burden on maintenance by increasing the cost and the time work on it, especially since it is a bank application, which requires security and very high standards, because of the sensitivity of the data included in the app.
Some of the mistakes we found

- No documentation, which make the code more difficult to understand.
- they use more than one variable style so it difficult to read.
- no vertical alignment, which make the code more difficult to read.
- didn't use proper data structure, they repeat the line many times
- a lot of identifies are not meaningful so it's not clear the purpose of it
- some functions names are not meaningful so it's difficult to tell what the function do
- The names of some of the classes are meaningless and do not give an idea of the content of the classes
- Classes were not arranged clearly, which made it difficult to access specific classes

```
17.
18.     public static void setLogLevel(String str) {
19.         if ("VERBOSE".equals(str)) {
20.             LOGLEVEL = 2;
21.         } else if ("DEBUG".equals(str)) {
22.             LOGLEVEL = 3;
23.         } else if ("INFO".equals(str)) {
24.             LOGLEVEL = 4;
25.         } else if ("WARN".equals(str)) {
26.             LOGLEVEL = 5;
27.         } else if ("ERROR".equals(str)) {
28.             LOGLEVEL = 6;
29.         }
```

Figure 1 : didn't use proper data structure

```
36.
37.         public static void v(String str, String str2) {
38.             if (2 >= LOGLEVEL) {
39.                 Log.v(str, str2);
40.             }
41.         }
42.
43.         public static void d(String str, String str2) {
44.             if (3 >= LOGLEVEL) {
45.                 Log.d(str, str2);
46.             }
47.         }
48.
49.         public static void i(String str, String str2) {
50.             if (4 >= LOGLEVEL) {
51.                 Log.i(str, str2);
52.             }
53.         }
54.
55.         public static void w(String str, String str2) {
56.             if (5 >= LOGLEVEL) {
57.                 Log.w(str, str2);
58.             }
59.         }
60.
61.         public static void e(String str, String str2) {
62.             if (6 >= LOGLEVEL) {
63.                 Log.e(str, str2);
64.             }
65.         }
66.
67.         public static void v(String str, String str2, Throwable th) {
68.             if (2 >= LOGLEVEL) {
69.                 Log.v(str, str2, th);
70.             }
71.         }
```

Figure 2 : identifies method are not meaningful

```
61. public class InAppBrowser extends CordovaPlugin {
62.     private static final String BEFORELOAD = "beforeload";
63.     private static final String CLEAR_ALL_CACHE = "clearcache";
64.     private static final String CLEAR_SESSION_CACHE = "clearsessioncache";
65.     private static final String CLOSE_BUTTON_CAPTION = "closebuttoncaption";
66.     private static final String CLOSE_BUTTON_COLOR = "closebuttoncolor";
67.     private static final Boolean DEFAULT_HARDWARE_BACK = true;
68.     private static final String EXIT_EVENT = "exit";
69.     private static final int FILECHOOSER_REQUESTCODE = 1;
70.     private static final int FILECHOOSER_REQUESTCODE_LOLLIPOP = 2;
71.     private static final String FOOTER = "footer";
72.     private static final String FOOTER_COLOR = "footercolor";
73.     private static final String HARDWARE_BACK_BUTTON = "hardwareback";
74.     private static final String HIDDEN = "hidden";
75.     private static final String HIDE_NAVIGATION = "hidenavigationbuttons";
76.     private static final String HIDE_URL = "hideurlbar";
77.     private static final String LEFT_TO_RIGHT = "lefttoright";
78.     private static final String LOAD_ERROR_EVENT = "loaderror";
79.     private static final String LOAD_START_EVENT = "loadstart";
80.     private static final String LOAD_STOP_EVENT = "loadstop";
81.     private static final String LOCATION = "location";
82.     protected static final String LOG_TAG = "InAppBrowser";
83.     private static final String MEDIA_PLAYBACK_REQUIRES_USER_ACTION = "mediaPlaybackRequiresUserAction"
84.     private static final String MESSAGE_EVENT = "message";
85.     private static final String NAVIGATION_COLOR = "navigationbuttoncolor";
86.     private static final String NULL = "null";
87.     private static final String SELF = "_self";
88.     private static final String SHOULD_PAUSE = "shouldPauseOnSuspend";
89.     private static final String SYSTEM = "_system";
90.     private static final String TOOLBAR_COLOR = "toolbarcolor";
91.     private static final String USER_WIDE_VIEW_PORT = "useWideViewPort";
92.     private static final String ZOOM = "zoom";
93.     private static final List customizableOptions = Arrays.asList(CLOSE_BUTTON_CAPTION, TOOLBAR_COLOR,
94.     private String[] allowedSchemes;
95.     private String beforeload = "";
```

Figure 3 : use more than one variable style

```
131.
132.     public void onSaveInstanceState(Bundle bundle) {
133.         CordovaPlugin cordovaPlugin = this.activityResultCallback;
134.         if (cordovaPlugin != null) {
135.             bundle.putString("callbackService", cordovaPlugin.getServiceName());
136.         }
137.         PluginManager pluginManager2 = this.pluginManager;
138.         if (pluginManager2 != null) {
139.             bundle.putBundle("plugin", pluginManager2.onSaveInstanceState());
140.         }
141.     }
142.
143.     public void restoreInstanceState(Bundle bundle) {
144.         this.initCallbackService = bundle.getString("callbackService");
145.         this.savedPluginState = bundle.getBundle("plugin");
146.         this.activityWasDestroyed = true;
147.     }
148.
149.     /* access modifiers changed from: private */
150.     public static class ActivityResultHolder {
151.         private Intent intent;
152.         private int requestCode;
153.         private int resultCode;
154.
155.         public ActivityResultHolder(int i, int i2, Intent intent2) {
156.             this.requestCode = i;
157.             this.resultCode = i2;
158.             this.intent = intent2;
159.         }
160.     }
161.
162.     public void onRequestPermissionResult(int i, String[] strArr, int[] iArr) throws JSONException {
163.         Pair<CordovaPlugin, Integer> andRemoveCallback = this.permissionResultCallbacks.getAndRemoveCallback(i);
164.         if (andRemoveCallback != null) {
165.             ((CordovaPlugin) andRemoveCallback.first).onRequestPermissionResult(((Integer) andRemoveCallback.second).intValue(),
166.         }
167.     }
168.
```

Figure 4 : no vertical alignment

2.1.3 Security Analysis

software code is more vulnerable to malicious malware and unauthorized users, because it is the core of any application system. And to avoid that you must check for any vulnerabilities and implement appropriate security measures; otherwise, the entire application may be jeopardized.

Predefined security policies improve speed while also allowing for checks on automated processes to avoid any deployment disasters (such as mistakenly bringing the entire infrastructure down because a problem wasn't spotted in a staging environment).

App Security Score Calculation

Every app is given an ideal score of 100 to begin with.
For every findings with severity high we reduce 15 from the score.
For every findings with severity warning we reduce 10 from the score.
For every findings with severity good we add 5 to the score.
If the calculated score is greater than 100, then the app security score is considered as 100.
And if the calculated score is less than 0, then the app security score is considered as 10.

Risk Calculation

APP SECURITY SCORE	RISK
0 - 15	**Critical**
16 – 40	**High**
41 – 70	**Medium**
71 - 100	**Low**

Our focus on this report will be on **severity high** issues.

Manifest Analysis

Num	Issue	Severity	Description	Solution
1	Clear text traffic is Enabled for App	High	The app intends to use cleartext network traffic, such as cleartext HTTP, FTP stacks, DownloadManager, and MediaPlayer. The default value for apps that target API level 27 or lower is "true". Apps that target API level 28 or higher default to "false". a network attacker can eavesdrop on transmitted data and also modify it without being detected.	Avoid cleartext traffic in order to maintain confidentiality, authenticity, and protections against tampering [1].
2	Broadcast Receiver is not Protected. An intent-filter exists.	High	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported.	Possible solution is to add permission on receiver [2].
3	Content Provider is not Protected.	High	A Content Provider is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.	Make the content provider private, to Restrict the access [3].
4	Broadcast Receiver is Protected by a permission, but the protection level of the permission should be checked.	High	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. It is protected by a permission which is not defined in the analyzed application.	As a result, the protection level of the permission should be checked where it is defined. If it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component. If it is set to signature, only applications signed with the same certificate can obtain the permission [3].

Code Analysis

Num	Issue	Severity	Solution
1	The App uses the encryption mode CBC with PKCS5/PKCS7 padding. This configuration is vulnerable to padding oracle attacks. Shown in the figure(5) blow	High	Change encryption mode to more secure one like GCM [3].
2	App can read/write to External Storage. Any App can read data written to External Storage. Shown in the figure(6) blow	High	1-Encrypt any sensitive data that it writes to external storage [4]. 2-Perform input validation on any data that it reads from external storage [4].
3	This App may request root (Super User) privileges. Shown in the figure(7) blow	High	No solution.

```
public class a {
    public String a(byte[] bArr, byte[] bArr2, String str) throws GeneralSecurityException {
        byte[] bytes = str.getBytes();
        SecretKeySpec secretKeySpec = new SecretKeySpec(bArr, "AES");
        Cipher instance = Cipher.getInstance("AES/CBC/PKCS5Padding");
        instance.init(1, secretKeySpec, new IvParameterSpec(bArr2));
        return new String(Base64.encode(a(bArr2, instance.doFinal(bytes)), 0));
    }
```

Figure 5: issue 1

```
kage com.scottyab.rootbeer;

lic final class Const {
 public static final String[] knownDangerousAppsPackages = {"com.koushikdutta.rommanager", "com.koushikdutta.rommanager.license", "com.dimonvideo.luckyp
 public static final String[] knownRootAppsPackages = {"com.noshufou.android.su", "com.noshufou.android.su.elite", "eu.chainfire.supersu", "com.koushikd
 public static final String[] knownRootCloakingPackages = {"com.devadvance.rootcloak", "com.devadvance.rootcloakplus", "de.robv.android.xposed.installer
 public static final String[] pathsThatShouldNotBeWrtiable = {"/system", "/system/bin", "/system/sbin", "/system/xbin", "/vendor/bin", "/sbin", "/etc"};
 public static final String[] suPaths = {"/data/local/", "/data/local/bin/", "/data/local/xbin/", "/sbin/", "/su/bin/", "/system/bin/", "/system/bin/.ex

 private Const() throws InstantiationException {
     throw new InstantiationException("This class is not for instantiation");
 }
```

Figure 6: issue 2

```
    private void performImageSave() throws JSONException {
        File file = new File(this.filePath);
        File externalStoragePublicDirectory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
        Log.d("SaveImage", "SaveImage dstGalleryFolder: " + externalStoragePublicDirectory);
        try {
            File copyFile = copyFile(file, externalStoragePublicDirectory);
            scanPhoto(copyFile);
            this.callbackContext.success(copyFile.toString());
        } catch (RuntimeException e) {
            CallbackContext callbackContext2 = this.callbackContext;
            callbackContext2.error("RuntimeException occurred: " + e.getMessage());
        }
    }
```

Figure 7: issue 3

2.1.4 Design

2.1.4.1 Graphical User Interface

Taking a look at the program's interface, and from our viewpoint as users, we notice that it is primitive in terms of design, as it does not offer the option of changing the theme so there is only one, however, it is excellent when it comes to arranging information and making it readily accessible. It also allows the user to choose one of the two languages(Arabic, English).

And after logging in, we see that the interface is good, but the way to display the latest transfers and operations is annoying because it is displayed automatically without asking, and there is no search box, which makes it difficult for the user to find any service, also the lack of customer care and no automatic response, which makes it difficult to communicate when there is a problem or request an inquiry. As well as unavailability of common questions and answers that reduce frequent customer inquiries.

2.1.4.2 Internal design

As for the internal design, and after looking into some general codes, we were able to see that the classes vary in terms of cohesion, some have low cohesion and others have high cohesion. There are classes designed with a single, well-focused purpose., while others perform multiple purposes that are not related to each other, which in result makes the class much difficult to maintain (and more frequently changed) than classes with high cohesion. Moreover, we were able to determine that some of the classes were designed with tight coupling which decreased flexibility and re-usability of code, hampered testability, made changes more difficult, etc. While others were designed with loose coupling, which will help you when your application need to change or grow.

3.Discussion

It was expected to find a professional-looking code as this was a banking application belonging to a large organization, but we found repeated, clear, and obvious issues that affected the quality of the code, which impaired our understanding and analysis of the code.Ideally, A company is supposed to set specific, high-quality standards for code writing that it applies to software developers, so that any developer, whether the original creator or the glitch repair team, can understand and maintain it easily and reduce costs associated with error fixing.The Mobfs tool enabled us to identify some security problems that were unexpected due to the high sensitivity and high confidentiality of the included data, which could lead to legal accountability by customers when their data is compromised in any way, so the company must give more attention to application security , and it is important the company addresses both application security and usability balances so that the use of the program does not become a burden for its users.

4.Conclusion

In the process of analyzing the code of the development program and using the tool to assist us, we discovered some issues that affect the code as a whole, and the security in particular because of the extreme confidentiality in the data of the program. As it is the biggest reason that reduces the quality of the code. The purpose of code analysis is to help programmers to raise the level of software quality, fix code errors and develop better programs.

References

[1] Pie, P., Haider, I., Klein, Q. and Ayed, N., 2021. Picasso image loading issue with Android 9.0 Pie. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/53139689/picasso-image-loading-issue-with-android-9-0-pie> [Accessed 3 November 2021].

[2] android, H., 2021. How to set permissions in broadcast sender and receiver in android. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/11770794/how-to-set-permissions-in-broadcast-sender-and-receiver-in-android> [Accessed 3 November 2021].

[3] Mobsf Tool.

[4] FutureLearn. 2021. Internal and external storage security. [online] Available at: <https://www.futurelearn.com/info/courses/secure-android-app-development/0/steps/21599> [Accessed 3 November 2021].