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

Computer Science Department

CS392 Software Engineering 2

Workshop #1

Quality Assessment For Local Banking Apps

First Semester 1443/2021

Section:371

Group Members' Names	Students IDs	University Email
Reema Saeed Alqahtani	440018811	rsmalqahtani11@sm.imamu.edu.sa
Abeer Mohammed Aldosari	440023513	amsaldosari13@sm.imamu.edu.sa
Mudhi Abdulaziz Alqahtani	439021022	Madalqahtani22@sm.imamu.edu.sa
Razan Saad Alabdulkrim	440021710	rsaalabdukrim@sm.imamu.edu.sa

Supervisor Name : Dr.Eng Lamees Alhazzaa

Contents

1.Introduction

Maintaining a good quality code is as important as writing the code. The quality of code can be measured by using tools or seeking review by someone experienced. Code quality is very important for the successful implementation of any program, Code quality is measures of how well code can communicate between developers. When a developer fails to heed the code quality, it can sometimes lead to rewrite in the code, this can eventually enhance the cost of the software because the maintenance in the software lifetime it's expensive.

1. 1 Purpose

The purpose of this workshop is to describe the quality assessment of local banking applications by using Android applications.
In this workshop, we will discuss and explain the quality of the code in the Al-Rajhi application mobile with the Android system, through which we will analyze the program code from several aspects, namely:
Optimization, Readability, Maintainability, Compatibility, Security, Understandability, Documentation.

1.2 Goals

Ensuring maintainability :
Because software is difficult and expensive to maintain, software must be developed so that developers will be able to use and modify existing code in the future.Maintainability is achieved by organizing the code and placing appropriate comments so that anyone not involved in the project is able to read the code and understand what it is doing.

Improving code :
The code must be well developed to comply with the standards of good code:
It does what it should, has a consistent style, is easy to read and understand, is well documented, is testable, and is free from repetitions.

Improving programmer :
The discovery and debugging of programming errors by developers who have sufficient previous experience helps programmers to increase their knowledge of programming when they know their errors and know how to correct and develop them to avoid them when writing code in the coming times.

Achieving safety :
Constantly checking the code base and looking for vulnerabilities to avoid being exploited by outside parties.

1.3 Lessons Learned

During this project, we learned several things that I will mention as points:

- We learned how to analyze the code superficially, and how to write well the names of variables and methods.
- We learned that every language has rules and basics for writing code, like Java uses CapelCase.
- We learned when repeating code errors, whether writing wrongly or incorrectly defining variables may cause a very high cost to fix.
- We learned that comments should be written with each method to know the reason of this method and to avoid wasting a lot of time to understand the program for other programmers.
- We learned what GitHub is, how to use it, and how to use it in our future projects.
- We learned that the permission can be dangerous to the system, and we did not expect its severity until the research.
- We learned that security has risks, and the level of risk varies from high to medium.

2. Mobile App Analysis results

- At first, our group planned to meet in an application such as Microsoft Teams and Zoom to decide which banking app we would choose to analyze it from many aspects.
- Then we chose the Al-Rajhi application and the reason for choosing this App is because all our bank accounts are from Al-Rajhi Bank and we want to know the technical aspects of this program to analyze it from many aspects of our safety and reliability, and so on.
- After our application was chosen, we had to download the Ubuntu system to access the Android application package (APK) and take a copy of this program to upload it in Mobfs for analyze it from many aspects in, but we encountered some problems in downloading this system, including:

- The first problem we faced was downloading the Docker in Ubuntu there were many problems, including:

The space was not enough to download the docker [Figure 1], so we added space in Ubuntu but somehow the page was Frozen and forced us to delete it and download the Ubuntu again and then, we had another problem in Downloading the Ubuntu says, "No bootable medium found!" As you can see in [Figure 2], so we look how to fix this problem and we found how to solve it in YouTube, so we were able to download the docker successfully.

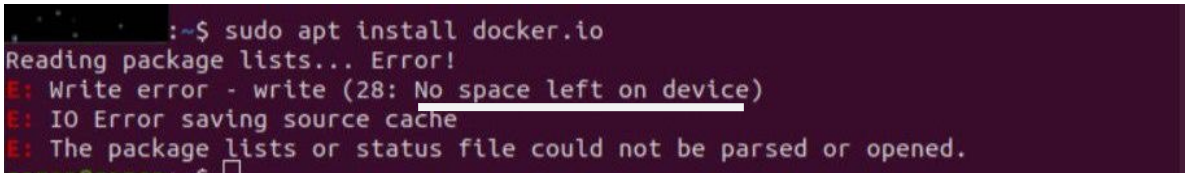


Figure 1

Figure 2

Link YouTube that solve the problem: https://youtu.be/vN-T8soPnLU

- After downloading and solving the problems we encountered while downloading the docker, then we searched for the Android play store to copy the link of the Al-Rajhi application we had a problem with downloading Android application package (Apk) we could not download the same website used by Dr. Sultan in the video, the problem was appeared to us that the page had been deleted or gone or old versions like that [Figure 3&4] and we tried to use many websites Apk but we could not able to download it.

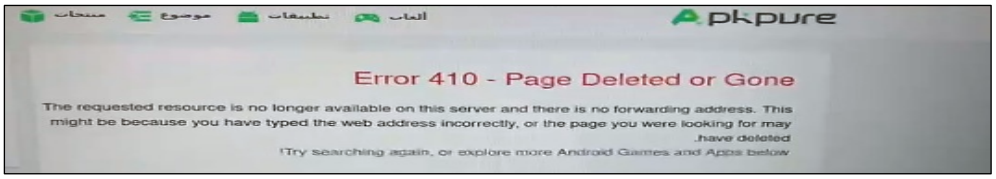


Figure 3

Apk denied download: https://m.apkpure.com/ar/

Figure 4

Apk denied download: https://apkdownload.com/

After a long and continuous search for a site that allows us to download the Al-Rajhi version, we found this site in the image below and after that the apk was successfully downloaded.

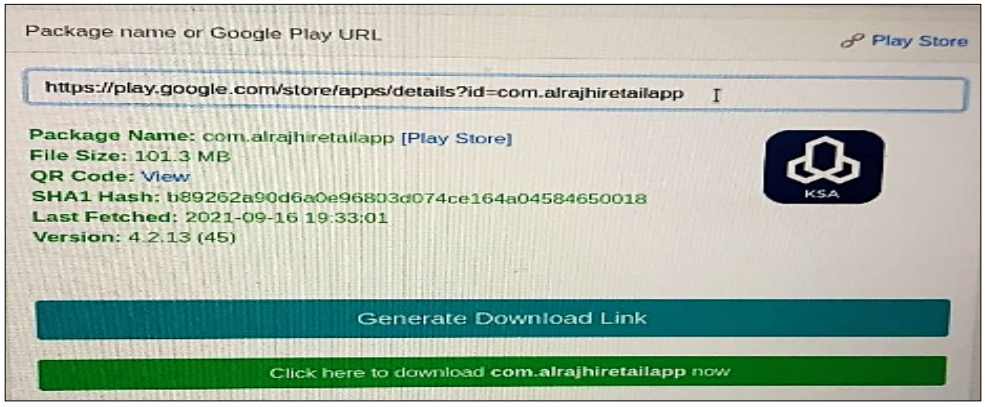


Figure 5

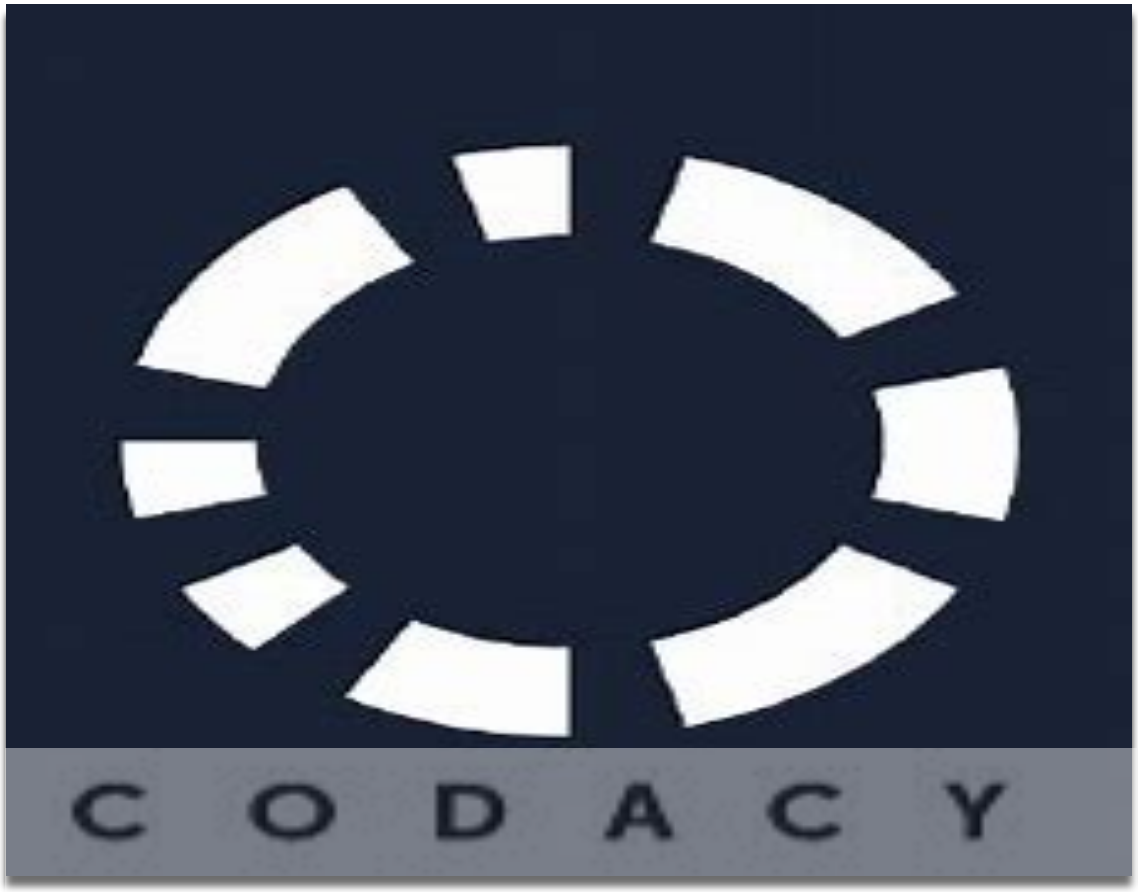
successful Apk download: https://apps.evozi.com/apk-downloader/

3. Discussion Results

3.1 Codacy-Github

It's a code quality analysis tool that helps developers with static analysis, **Quality settings,** cyclomatic complexity, code coverage **and Security checks**, Codacy can be used to enforce quality standards on your code and save time in code review, Codacy integrates with Github and some of the features of Codacy:

- **Support Multilanguage**
- Make sure your code quality is standardized across all teams and projects by applying code patterns
- security and performance checks early in the process to prevent your product from being affected by vulnerabilities.
- Identify exactly which lines of code are being covered by your test.



3.1.1 Code Style

We found 177,805 **issues** and all of them were medium and minor.

- **Avoid unused local variable.**

```
17
18  @Override // com.aurelhubert.ahbottomnavigation.s.a
19  public final Object map(Object obj) {
20      Integer num = (Integer) obj;
21      return Integer.valueOf(this.a.getColor(y.AHBottomNavigationBehavior_Params_accentColor, b.d(this.b, t.colorBottomNavigationAccent)));
22  }
```

```
291                 }
292                 this.H.setContentView(f0Var);
293             } else {
294                 ViewGroup viewGroup = (ViewGroup) this.H.getContentView();
295                 View view2 = this.p;
296                 if (view2 != null) {
```

```
169         private int d;
170         private boolean e;
171
172         public /* synthetic */ d(h hVar, n20 n20, u10 u10, int i, boolean z, a aVar) {
173             this(n20, u10, i, z);
174         }
```

This problem has occurred 255 times.

➔ I noticed that there are a lot of variables that are defined and have an original value that were not used, and this is not good at all for the cleanliness of the code, and because it wastes a lot of time to understand it and make it unused, and this affects the quality of the code.

- **The class or method name is not meaningful.**

```
11  import vigqyno.C0201;
12
13  /* compiled from: SwipeToDismissListener */
14  public class g implements View.OnTouchListener {
15      private final View a;
16      private int b;
```

```
20      /* access modifiers changed from: package-private */
21      /* renamed from: r6$a */
22      /* compiled from: ViewGroupUtilsApi14 */
23      public static class a extends LayoutTransition {
24          public boolean isChangingLayout() {
25              return true;
```

```
313         return as.h((double) (((float) l.intValue()) / 10.0f));
314     }
315
316     public String E() {
317         return m(49419, C0201.m82(38142), C0201.m82(38143), C0201.m82(38144), C0201.m82(38145), C0201.m82(38146), C0201.m
    1.m82(38150), C0201.m82(38151), C0201.m82(38152), C0201.m82(38153));
318     }
```

This problem has occurred 708 times.

➔ There are a lot of classes and methods whose names are meaningless and not appropriate at all. It is not possible to know this class for any purpose that was written, so on the other hand, I don't know what the purpose is of naming these incomprehensible names, but it greatly affects the quality of the code.

➢ Missing Javadoc for methods and variables

```
59      public void setPageWithoutAnimation(d dVar, int i) {
60      }
61
62      public void addView(d dVar, View view, int i) {
63          dVar.X(view, i);
64      }
```

```
820         return z2;
821     }
822
823     public void C0(Fragment fragment) {
824         if (K) {
825             Log.v(C0201.m82(18719), C0201.m82(18718) + fragment);
```

```
30      public static final String MARKETPLACE_PAYMENT_ID = null;
31      public static final String MARKETPLACE_PAYMENT_PASSWORD = null;
32      public static final String METHOD_AUTH = null;
33      public static final String MORTGAGE_SIT_URL = null;
34      public static final String ONE_TIME_PASSWORD = null;
35      public static final String PASSPHRASE = null;
```

This problem has occurred 86,279 times.

➔ This problem, I almost assumed that it existed in all code writing. Most written code has no comments on it to explain the method or variable, what is the purpose of using this method or variable and why?

3.1.2 Error prone

We found 8956 **issues** and all of them were medium.

➢ Document empty method body

```
575         this.g = z2 && getContext().getApplicationInfo().targetSdkVersion < 19;
576     }
577
578     public void setShowingForActionMode(boolean z2) {
579     }
580
```

```
440     }
441
442     @Override // defpackage.e3
443     public void onNestedPreScroll(View view, int i2, int i3, int[] iArr) {
444     }
445
```

```
60          return this.a;
61      }
62
63      public void l() {
64      }
65
```

This problem has occurred 828 times.

➔ There are several reasons for a method not to have a method body:

- Unintended omission of the programmer
- It is not supported yet
- The method is an intentionally blank override.

➔ This error may cause a problem with the results or the occurrence of unexpected results. In the second case, the programmer had to set the UnsupportedOperationException. Finally, a comment should have been made and explain why the blank is override.

- **Classes should not have non-constructor methods with the same name as the class**

```
24          e(resources);
25      }
26
27      private f d() {
28          return new f(this.d);
29      }
```

```
18          private boolean b = false;
19          private boolean c = false;
20
21          public final a a(LocationRequest locationRequest) {
22              if (locationRequest != null) {
23                  this.a.add(locationRequest);
```

```
78                  return -1;
79              }
80
81              public int b(int i) {
82                  if (c(i, 4)) {
83                      return this.a.getInt(i);
```

This problem has occurred 1118 times.

- ➔ This error was repeated many and many times and reached up to 1118 times, and this error is against the rules of Java Class.
- ➔ And this could lead other programmers to get confused with non-constructer methods with the constructor's name.
- ➔ anyway, it will takes 30 minutes to fix this problem, and with so many same errors it would be really expensive.

➢ Avoid unnecessary if..then..else statements when returning Booleans

```
79          obtainNoHistory.recycle();
80          int actionMasked = motionEvent.getActionMasked();
81          boolean z = (actionMasked == 1 || actionMasked == 3) ? false : true;
82          if (!e2 || !z) {
83              return false;
84          }
```

```
843         if (weakReference2 != null) {
844             view = weakReference2.get();
845         }
846         if (actionMasked != 2 || view == null || this.x || this.v == 1 || coordinatorLayout.G(view, (int) motionEvent.getX(), (int) motionEvent.getY()) || this.
    w == null || Math.abs(((float) this.H) - motionEvent.getY()) <= ((float) this.w.y())) {
847             return false;
848         }
```

```
23              return false;
24          }
25          BooleanResult booleanResult = (BooleanResult) obj;
26          if (this.resultValue != booleanResult.getValue() || !this.myStatus.equals(booleanResult.getStatus())) {
27              return false;
28          }
```

This problem has occurred 372 times.

➔ Avoid unnecessary if-then-else statements when returning a Boolean. The conditional test result can be returned instead. Avoid unnecessary comparisons in Boolean expressions, they serve no purpose and affect readability.

➔ This error affects the readability and also, in my opinion, gives the impression that the programmer is inexperienced, and because this error is repeated about 400 times, it will cost a lot to repair.

3.1.3 Performance

We found 4470 **issues** and all of them were medium.

➢ Useless parentheses

```
89    public final int hashCode() {
90        long doubleToLongBits = Double.doubleToLongBits(this.d);
91        long doubleToLongBits2 = Double.doubleToLongBits(this.e);
92        return ((((((((((int) (doubleToLongBits ^ (doubleToLongBits >>> 32))) + 31) * 31) + ((int) (doubleToLongBits2 ^ (doubleToLongBits2 >>> 32)))) * 31) + Float.floatToIn
tBits(this.f)) * 31) + this.c) * 31) + this.g;
93    }
94
```

```
19          this.c = new byte[i3];
20          for (int i4 = 0; i4 < i3; i4++) {
21              int i5 = iArr[i4];
22              this.c[i4] = (byte) (((((i5 >> 16) & 255) + ((i5 >> 7) & 510)) + (i5 & 255)) / 4);
23          }
24      }
```

```
753         if (this.z) {
754             T(this.i, ((-this.g.e(this.k)) + ((float) getHeight())) * f2, z2);
755         } else {
756             T(((-this.g.e(this.k)) + ((float) getWidth())) * f2, this.j, z2);
757         }
758         P();
```

This problem has occurred 1622 times.

➔ As we can see, When redundant useless parentheses are used, it is assumed that the increase in this is what makes the code clear and understandable and arranges the programming operations, but I think no, because when I tried to understand the code in general, I encountered difficulty understanding it because a lot of parentheses makes the code misleading and the result is wrong in the piece of code, which reduces the efficiency and performance of the code.

One of the solutions that can be used are:

- In each writing part of the code, a test was made to perform this code and prove the correctness of its work, and this increases performance
- Reduce as much as possible the unwanted parentheses and put only what we need to reduce the error of the code

➢ Avoid instantiating new objects inside loops

```
177         int i3 = 0;
178         int i4 = 100;
179         while (i3 < 1001 && i4 == 100) {
180             ArrayList arrayList = new ArrayList();
181             if (!d0 || (H = t().H(100)) == null) {
182                 i2 = 0;
```

```
223                 int i6 = 53760 + i5;
224                 if (i6 == 53768 || i6 == 53769) {
225                     if (iArr5.length - 1 >= i4 + i3) {
226                         short[] sArr = new short[i3];
227                         for (int i7 = 0; i7 < i3; i7++) {
228                             sArr[i7] = (short) iArr5[i4 + i7];
```

```
110     private static File a(File file, String... strArr) {
111         for (String str : strArr) {
112             if (str != null) {
113                 file = new File(file, str);
114             }
115         }
```

This problem has occurred 1454 times.

➔ The error here is that an object has been defined or created within loop, which could result in a memory being allocated each time. Actually, it is not a big problem if the program is small, but if the program is large of course the memory space it will be huge, then the problem will be a very expensive.

➔ So the program should instantiating new object OUTSIDE the loops.

➢ Avoid unnecessary return statements

```
136             }
137             this.c.b(this.a, dVar);
138         }
139         return;
140         sb.append(r0);
141         sb.append(this.a.b());
```

```
444             }
445             throw new C0080ab(C0201.m82(13469));
446         }
447         return;
448         com.huawei.hianalytics.ab.bc.ef.ab.fg(r0, str);
449         if (ec2 == null) {
```

This problem has occurred 2 times.

➔ The error here is big and very expensive, because the return will ends the method, even if there is a code below the return.

3.2 MOBFs

3.2.1 Application Permission

android.permission.SYSTEM_ALERT_WINDOW	dangerous	display system-level alerts	Allows an application to show system-alert windows. Malicious applications can take over the entire screen of the phone.

android.permission.SYSTEM_ALERT_WINDOW (abbreviated SAW):
To use this permission, each app requests the SYSTEM_ALERT_WINDOW permission and is installed through the Play Store. The app must obtain consent from the user "by manually agreeing to this permission", except version 6.0.5 or higher automatically download without the user's permission. This permission allows other apps to have a look at the app, by allowing the app to be displayed on top of any other app without the user's knowledge. Thus, it has a huge potential of harmful methods such as,

- displaying fraudulent ads
- phishing scams
- click-jacking
- and overlay windows
- which are common with banking Trojans

❖ Ransomware can also be used to prevent users from accessing their devices by creating a fixed screen.

The way these attacks lead to something like this:
1- The user downloads the malicious application.

➢ Thus, it uses the SYSTEM_ALERT_WINDOW permission to display messages at the top of the user interface.

2- The screen asks the user for important personal information, and the user discloses this information for the purpose of not disturbing and making the screen disappear.

➢ Thus, this makes a number of attacks.

3- Once the malicious application is installed on your device, it will be waiting for the user to start the application.
4- When you run the application, it will ask you to enter information in a fake screen, and then it will start sending the data to the attacker's server.
Be careful when granting this permission, because it means that Your consent to download malware and steal your personal information requested within the application. When you download any application, make sure that it is the actual application you want, how do you know? You can go to the website of the parent company that invented the application - download it in the Play Store and download it directly from there. If this malicious app gets your consent, you may run the risk of scams, scams, and more.

〉 Users can see what apps the permission have using the instructions in the table below (instructions may not apply to all devices).

Version	How to review apps with Draw Over Other Apps permission
Android 7	Settings > Apps > “Gear symbol” > Special Access > Draw over other apps
Android 6	Settings > Apps > “Gear symbol” > Draw over other apps
Android 5	Settings > Apps > Select app and look for “draw over other apps”

android.permission.CAMERA	dangerous	take pictures and videos	Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.

android.permission.CAMERA :

This permission allows the app to launch the camera and take photos and videos .

When a user grants access to their camera, the app can do the following:

- Access to both front and back camera.
- Take photos and videos without telling you.
- Upload the photos/videos it takes instantly.
- Turn on real-time face recognition to detect facial features or expressions.

AVG, a security software company, warned against allowing applications to access the camera, noting that a malicious application can secretly launch the camera and record what is happening around you, and this is a violation of user privacy, so users' confidential data should be protected and not used for commercial or other purposes, I think this will of course affect the number of users of the app, and users may turn to other, more secure banking apps. In my opinion, this issue can be resolved by disabling Allow the app to access the camera from the settings and activating it only when needed. Disabling the permissions may affect the performance of the program as it may cause the application to lose some of its functionality.

android.permission.RECORD_AUDIO	dangerous	record audio	Allows application to access the audio record path.

android.permission.RECORD_AUDIO:

In my opinion, any application capable of accessing the audio recording path is a violation of the user's privacy for any purpose, because the voice is considered one of the user's private data. This data is recorded, processed and stored, and this leads to the application's lack of authentication and inefficiency, and on the other hand, not only from The application company deals with this data only, but the application company uses it for other purposes that were profitable for them, such as advertisements, or it was another purpose, and this may expose the user to a violation in the event that a hacker attacked and stole or leaked this data.

I think this is one of the solutions that can be applied using it

- An app that uses a dangerous permission must ask the user for consent each time to access the audio path or after it takes permission from the user asks the user not to be reminded again after the permission is granted
- The application company writes in its own terms, immediately after downloading the application, that the audio path is accessed, and the user agrees to these terms to absolve the application company of the violation, and the user agrees to these terms and knows that

android.permission.READ_CONTACTS	dangerous	read contact data	Allows an application to read all of the contact (address) data stored on your phone. Malicious applications can use this to send your data to other people.

android.permission.READ_CONTACTS:

As shown above, the status of requesting permission to read contact is dangerous, and this situation is not good at all, because when the user consents to the application's access to the user's contact, it is dangerous for the user because it affects his privacy and user’s contact privacy, other than that the application may read the contacts Every time he needs and may have private information of the user's other contacts, he may display a lot of information about the user and his contacts users' information

- We can make the app get permission every time it needs access to a user's contact
- Inform the user of the item during download that the application may allow him to access the user's contacts
- There is an option we can turn off allow access to contacts

android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.

android.permission.ACCESS_FINE_LOCATION :

Determining your geographical location through the global navigation system that operates via satellite and is needed by navigation applications, maps and prayer times, for example. Be careful when granting this permission, because it means that your geographic location can be detected by the responsible app, after obtaining the location information permission, apps can get your location at any time. And if a malicious app gets your location information, you may run the risk of scams, scams, and more. In my opinion, this problem can be resolved by enabling or disabling location services as needed and regularly monitoring and managing the permissions granted to the application to protect the privacy and security of user information, as there is a feature to allow access to the location while using the application only, and undertakes not to share the user's location information with another application If not authorized.

android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.

android.permission.ACCESS_COARSE_LOCATION:

This service allows the API to use Wi-Fi and cellular data to determine a device's location with an accuracy equivalent to city block, "which means it identifies the location very clearly."
In general, three options appear when using the program or application to allow locating:

1) Allow Always On selecting “Allow Always”.
2) Allow only while using the app, means that it shares user location only when the application is used.
3) or Deny While clicking “Deny”, means the application cannot use your location.

Certainly, the program will be able to access the database to determine the location, which may allow malicious applications to determine your location perfectly.
I think developers can make periodic fixes to avoid security holes and, of course, to avoid malicious applications! Also, users can use the second option to avoid problems as much as possible.

android.permission.ACCESS_BACKGROUND_LOCATION	dangerous	access location in background	Allows an app to access location in the background.

android.permission.ACCESS_BACKGROUND_LOCATION:

Applications that access your location in the background may make malicious applications access your location easily, so you can reconsider allowing the location to be determined according to your need for this application, for example:

1) If background access to your location is critical, this can be allowed.
2) In case there is no need, you can just close the feature or remove the application.

I believe the developer's job is to make it clear to the user that the application is being used and determine its location in the background, and to give the user the choice whether to allow this or not to avoid problems that may occur to the device from battery drain or even malicious applications.
And I think that the developer should develop or refactor the location access logic so that the apps is only active when users used it.

3.2.2 Network Security

1.2.2.1Manifest Analysis

- **Clear text traffic is Enabled For App :**

The app use cleartext network traffic, such as cleartext HTTP and MediaPlayer. you should avoiding cleartext traffic because is the lack of confidentiality and protections against tampering; because the network attacker can eavesdrop on transmitted data and also modify it without being detected.

- **Launch Mode of Activity is not standard :**

It is possible for other applications to read the contents of the calling Intent (abstract description of an operation to be performed). It's so dangerous because sensitive information is included in an Intent.

- **Activity-Alias is not Protected :**
  An Activity-Alias is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.
- **Service is not Protected :**

Because we have an encryption problem and we didn't get other information on this point.

- **Broadcast Receiver :**

1-not Protected [android:exported=true]:-

A Broadcast Receiver is discovered to be shared with other apps on the device, making it available to any other application.

2-Protected by permission [android:exported=true]:-

A Broadcast Receiver is discovered to be shared with other apps on the device, making it available to any other application. Protected by a permission, which is mean it's not defined in the analyzed application. but the protection level of the permission should be checked. even if it is set to normal or dangerous, a malicious application can request and obtain the permission and interact with the component.

Permission: com.google.android.c2dm.permission.SEND is held by Google Play services. This prevents other apps from invoking the broadcast receiver.

-what do we mean by **android:exported** whether it true or false??

true: events delivered by the same or different apps can be received by the broadcast receiver.

false : events sent by the same application can be received by the broadcast receiver.

3.2.2.2 Code Analysis

4	App can read/write to External Storage. Any App can read data written to External Storage.	high	CVSS V2: 5.5 (medium) CWE: CWE-276 Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2	defpackage/wc2.java com/reactnative/ivpusic/imagepicker/d.java defpackage/t40.java com/huawei/hms/update/a/d.java io/invertase/firebase/storage/RNFirebaseStorage.java com/learnium/RNDeviceInfo/RNDeviceModule.java defpackage/yc2.java com/christopherdro/htmltopdf/RNHTMLtoPDFModule.java defpackage/z60.java com/RNFetchBlob/d.java com/wansbrough/RCTCamera/RCTCameraModule.java com/reactnative/ivpusic/imagepicker/PickerModule.java com/RNFetchBlob/Utils/a.java com/reactnative/ivpusic/imagepicker/a.java com/reactnativecommunity/webview/RNCWebViewModule.java

App can read/write to External Storage. Any App can read data written to **External Storage**:

What do we mean about External Storage?
Every Android-compatible device Supports a shared External and internal Storage. These files are world readable and can be modified by the user. This storage might be removable (like an SD card) unlike internal (non-removable). it is possible that the USB capacity is large, the user can modify it. The file will be created, storing the data in a clear text file in external storage once the activity is requested. Also, you must know that the files are stored outside the application file. It will not be removed when the user uninstalls the application. It's important to note that the external storage can be acccess by an attacker that allow to control of the application in some cases.

Let look in deep of External Storage it’s have 2 types Public and private files:-
1- Private files: they are files specific to your application that are considered part of the application (but they are readable and written). They are considered private files but in fact they can be viewed and accessed by other applications on the device, they do not have strict protection by Android.
2- Public files: These files are not considered specific to a specific application and mean freely shared, any other files found on the external storage.

In my own opinion, the app should use the directories inside the internal storage because they are (non-removable) and will save sensitive information that other apps shouldn't have access to.

As you see from the file…

```
406.          String str;
407.          boolean exists = file.exists();
408.          String r2 = C0201.m82(32167);
409.          if (exists || file.mkdirs()) {
410.              String format = String.format(C0201.m82(32170), new SimpleDateFormat(C0201.m82(32169)).format(new Date()));
411.              if (i2 == 1) {
412.                  str = String.format(C0201.m82(32171), format);
413.              } else if (i2 == 2) {
414.                  str = String.format(C0201.m82(32172), format);
415.              } else {
416.                  Log.e(r2, C0201.m82(32174) + i2);
417.                  return null;
418.              }
419.              return new File(String.format(C0201.m82(32173), file.getPath(), File.separator, str));
420.          }
421.          Log.e(r2, C0201.m82(32168) + file.getAbsolutePath());
422.          return null;
423.      }
424.
425.      private File getOutputMediaFile(int i2) {
426.          String str;
427.          if (i2 == 1) {
428.              str = Environment.DIRECTORY_PICTURES;
429.          } else if (i2 == 2) {
430.              str = Environment.DIRECTORY_MOVIES;
431.          } else {
432.              Log.e(C0201.m82(32176), C0201.m82(32175) + i2);
433.              return null;
434.          }
435.          return getOutputFile(i2, Environment.getExternalStoragePublicDirectory(str));
436.      }
437.
438.      public static ReactApplicationContext getReactContextSingleton() {
439.          return _reactContext;
440.      }
441.
442.      private File getTempMediaFile(int i2) {
443.          String r0 = C0201.m82(32177);
444.          try {
445.              String format = new SimpleDateFormat(C0201.m82(32178)).format(new Date());
446.              File cacheDir = _reactContext.getCacheDir();
447.              if (i2 == 1) {
```

```
115.              String str4;
116.              String r7;
117.              boolean equals = str.equals(C0201.m82(24101));
118.              String r1 = C0201.m82(24102);
119.              if (equals) {
120.                  str4 = Environment.DIRECTORY_PICTURES;
121.                  r7 = C0201.m82(24103);
122.                  str3 = C0201.m82(24104);
123.              } else if (str.equals(C0201.m82(24105))) {
124.                  str4 = Environment.DIRECTORY_MOVIES;
125.                  r7 = C0201.m82(24106);
126.                  str3 = C0201.m82(24107);
127.              } else {
128.                  str2 = r1;
129.                  str3 = str2;
130.                  String str5 = r1 + String.valueOf(System.currentTimeMillis()) + str3;
131.                  if (Build.VERSION.SDK_INT >= 23) {
132.                      return new File(Environment.getExternalStoragePublicDirectory(str2), str5);
133.                  }
134.                  return File.createTempFile(str5, str3, getReactApplicationContext().getExternalFilesDir(null));
135.              }
136.              r1 = r7;
137.              str2 = str4;
138.              String str52 = r1 + String.valueOf(System.currentTimeMillis()) + str3;
139.              if (Build.VERSION.SDK_INT >= 23) {
140.              }
141.          }
142.
143.          private Intent getFileChooserIntent(String str) {
144.              String r0 = str.isEmpty() ? C0201.m82(24108) : str;
145.              if (str.matches(C0201.m82(24109))) {
146.                  r0 = getMimeTypeFromExtension(str.replace(C0201.m82(24110), C0201.m82(24111)));
147.              }
148.              Intent intent = new Intent(C0201.m82(24112));
149.              intent.addCategory(C0201.m82(24113));
150.              intent.setType(r0);
151.              return intent;
152.          }
153.
154.          private String getMimeTypeFromExtension(String str) {
155.              if (str != null) {
156.                  return MimeTypeMap.getSingleton().getMimeTypeFromExtension(str);
```

```
78.      }
79.
80.      public File d(Context context, String str, int i, int i2, int i3) throws IOException {
81.          int i4;
82.          int i5;
83.          Bitmap decodeFile = BitmapFactory.decodeFile(str);
84.          int width = decodeFile.getWidth();
85.          int height = decodeFile.getHeight();
86.          int attributeInt = new ExifInterface(str).getAttributeInt(C0201.m82(2478), 1);
87.          Matrix matrix = new Matrix();
88.          matrix.postRotate((float) c(attributeInt));
89.          float f = ((float) width) / ((float) height);
90.          float f2 = (float) i;
91.          float f3 = (float) i2;
92.          if (f2 / f3 > 1.0f) {
93.              i5 = (int) (f3 * f);
94.              i4 = i2;
95.          } else {
96.              i4 = (int) (f2 / f);
97.              i5 = i;
98.          }
99.          Bitmap createBitmap = Bitmap.createBitmap(Bitmap.createScaledBitmap(decodeFile, i5, i4, true), 0, 0, i5, i4, matrix, true);
100.         File externalFilesDir = context.getExternalFilesDir(Environment.DIRECTORY_PICTURES);
101.         if (!externalFilesDir.exists()) {
102.             Log.d(C0201.m82(2479), C0201.m82(2480));
103.             externalFilesDir.mkdirs();
104.         }
105.         File file = new File(externalFilesDir, UUID.randomUUID() + C0201.m82(2481));
106.         BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(new FileOutputStream(file));
107.         createBitmap.compress(Bitmap.CompressFormat.JPEG, i3, bufferedOutputStream);
108.         bufferedOutputStream.close();
109.         decodeFile.recycle();
110.         createBitmap.recycle();
111.         return file;
112.     }
113. }
```

5	The App uses an insecure Random Number Generator.	warning	CVSS V2: 7.5 (high) CWE: CWE-330 Use of Insufficiently Random Values OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-6	defpackage/f30.java defpackage/sh1.java defpackage/g30.java defpackage/v52.java

The APP uses an insecure Random Number Generator :

It is difficult for any computer to produce true random numbers, so we replace them with software that produces random number generators: a set of random numbers or values that appear as if they were randomly generated.

When this program generates guess values, the quality of the numbers varies according to the type of algorithm used.

It is resilient against prediction attacks, but is insufficient in a security context because it relies on unexpected numbers, so it is possible for an attacker to guess the values, and using that guess, the attacker will impersonate another user or access sensitive information.

In my opinion to fix this issue, the CWE crypto team must identify vulnerabilities related to randomness and unpredictability, as well as relationships between randomness and crypto primitives through planning, comprehension, or other scenarios. But it may not be a reliable solution if some details are neglected when analyzing the problem. As you see from the file…

```
3.  import android.graphics.Color;
4.  import com.google.android.gms.maps.model.b;
5.  import com.google.android.gms.maps.model.p;
6.  import com.google.android.gms.maps.model.t;
7.  import com.google.android.gms.maps.model.v;
8.  import java.util.HashMap;
9.  import java.util.HashSet;
10. import java.util.Random;
11. import vigqyno.C0201;
12.
13. /* renamed from: v52  reason: default package */
14. /* compiled from: KmlStyle */
15. public class v52 extends u42 {
16.     private final HashMap<String, String> d = new HashMap<>();
17.     private final HashSet<String> e = new HashSet<>();
18.     private boolean f = true;
19.     private boolean g = true;
20.     private String h;
21.     private double i = 1.0d;
22.     private String j = null;
23.     private boolean k = false;
24.     private boolean l = false;
25.     private boolean m = false;
26.     private float n = 0.0f;
27.
28.     public static int g(int i2) {
29.         Random random = new Random();
30.         int red = Color.red(i2);
31.         int green = Color.green(i2);
32.         int blue = Color.blue(i2);
33.         if (red != 0) {
34.             red = random.nextInt(red);
35.         }
36.         if (blue != 0) {
37.             blue = random.nextInt(blue);
38.         }
39.         if (green != 0) {
40.             green = random.nextInt(green);
41.         }
42.         return Color.rgb(red, green, blue);
43.     }
44.
45.     private static String h(String str) {
```

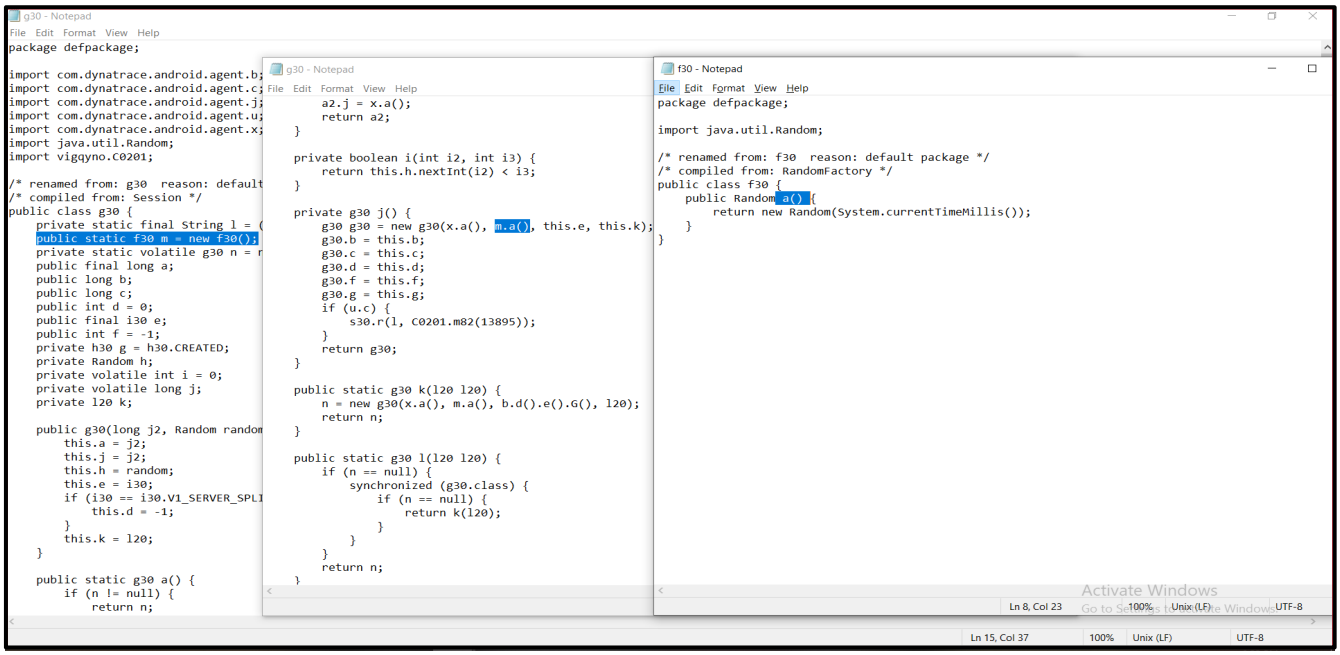
```java
package defpackage;

import java.util.Random;

/* renamed from: f30  reason: default package */
/* compiled from: RandomFactory */
public class f30 {
    public Random a() {
        return new Random(System.currentTimeMillis());
    }
}
```

```java
package defpackage;

import com.dynatrace.android.agent.b;
import com.dynatrace.android.agent.c;
import com.dynatrace.android.agent.j;
import com.dynatrace.android.agent.u;
import com.dynatrace.android.agent.x;
import java.util.Random;
import vigqyno.C0201;

/* renamed from: g30  reason: default package */
/* compiled from: Session */
public class g30 {
    private static final String l = (u.b + C0201.m82(13894));
    public static f30 m = new f30();
    private static volatile g30 n = null;
    public final long a;
    public long b;
    public long c;
    public int d = 0;
    public final i30 e;
    public int f = -1;
    private h30 g = h30.CREATED;
    private Random h;
    private volatile int i = 0;
    private volatile long j;
    private l20 k;

    public g30(long j2, Random random, i30 i30, l20 l20) {
        this.a = j2;
        this.j = j2;
        this.h = random;
        this.e = i30;
        if (i30 == i30.V1_SERVER_SPLITTING) {
            this.d = -1;
        }
        this.k = l20;
    }
```

4. design

4.1 High Coupling

High coupling is bad programming design most of the time because it reduces flexibility and code reusability, makes changes more difficult, and means having program modules that depend on each other. We are finding high coupling in code:

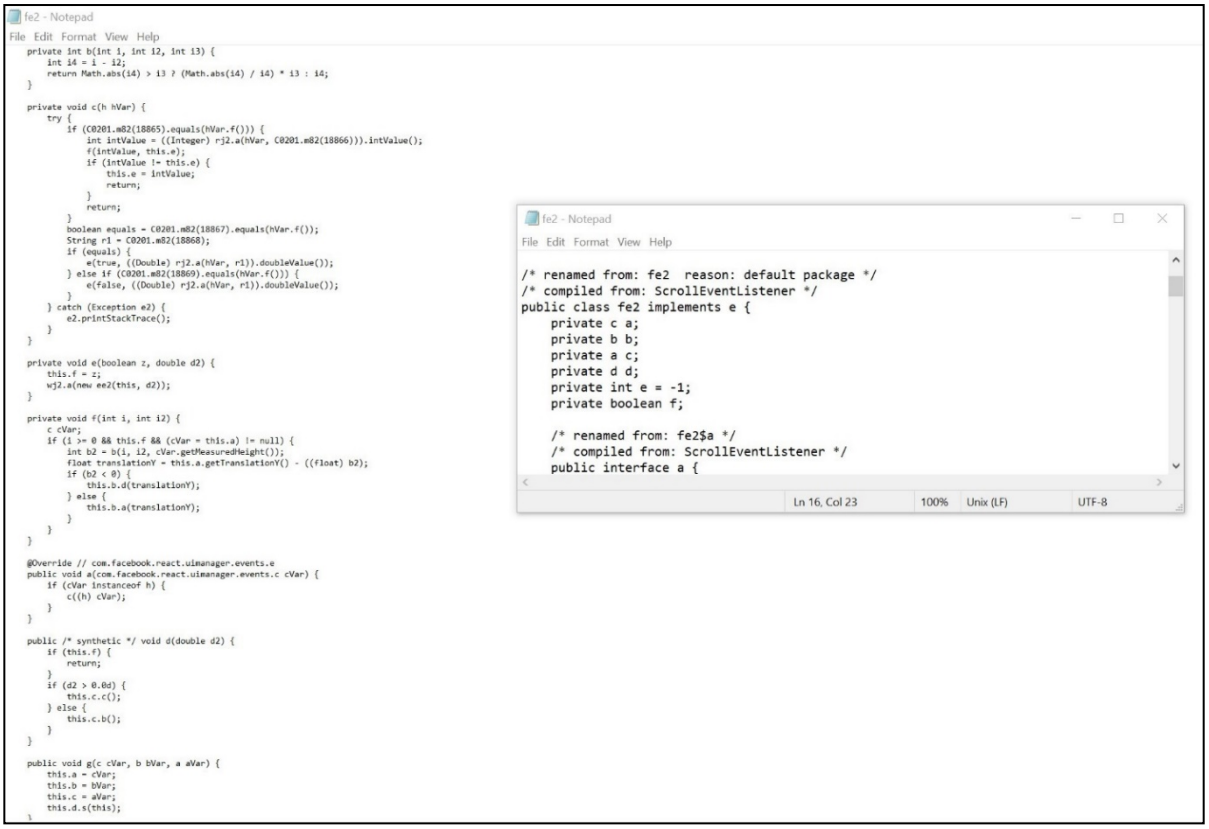


- Class g30 has an object m from class f30 which is declared public.
- Class g30 has a method j() that calls m.a() from class f30.

In the above program the g30 class is dependents on f30 class, In the above program g30 class is high coupled with f30 class it means if any change in the f30 class requires g30 class to change. For example, if f30 class a() method change to RandomNum() method then you have to change the j() method will call m.RandomNum() method instead of calling a() method.

4.2 Low Cohesion

Low cohesion is when a class does a lot of jobs that don't have much in common. results in monolithic classes that are difficult to maintain, understand and reduces re-usability.As you see it in the code:



- Class fe2 has a lot of method as shown has b, c, e, f, a, d and g and there are also more
- Each method described above does a different job using the objects defined above

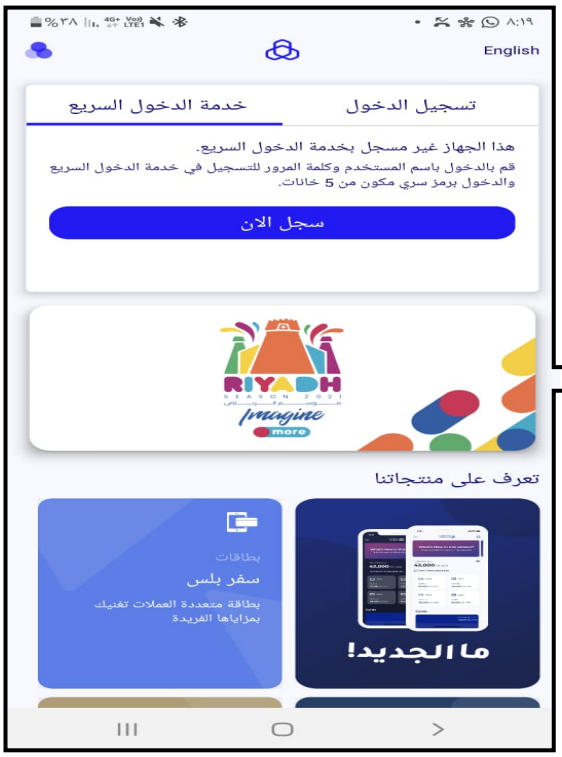
Hence, this class makes it easy to understand and maintain due to the presence of multiple and different methods, and it is even difficult to be reusable.

4.3 Design Interface

One of the advantages of the login interface is that when you click on the "**Remember me**" field, your username or national ID and password are saved to speed up the login process next time.

One of the advantages of the login interface is that the user can log in in two ways:

1- Log in with your national ID or username and password
2- Quick login by entering a 5-digit code



One of the advantages of the user interface is that the user can retrieve the password if they forget it, but its disadvantage is that the user cannot retrieve the username if he forgets it!

One of the disadvantages of the login interface is that if there is a field "**Open a bank account**", what is the meaning of the "**Registration**" field!

One of the advantages of the login page interface it has a **select a theme** feature, a nice feature that enables the user to change the application interfaces in the way that suits him. but disadvantages it's not clear that this is the sign that changes the interfaces, most likely the user will discover this feature after clicking on it to discover what it is.

One of the advantages of the home page interface it displays the **accounts** I owned, but one of disadvantage is that it's small size they can expose the accounts field.

One of the advantages of the home page interface is the presence of ads, such as "**month of financing**: it's to request a financial loan without having to go to a bank branch".

One of the advantages of the home page interface is that the user can use the **hidden mode of the account balance,** this is good for security, if the user opens the account in a public place, he can use this feature.

One of the advantages of the interface of the home page has **Quick Procedures** that it displays the two most important processes for the user **payments** and **transfers** and also shows the Riyadh season "to request a card and this card has advantages written details about it in the application", that facilitates the user to access it quickly to transfer money or view Account payments.

Many good options for the user by adding other services not related to their program to facilitate use

Languages can be changed and that's fine

Displays the last login made by the user without the current login

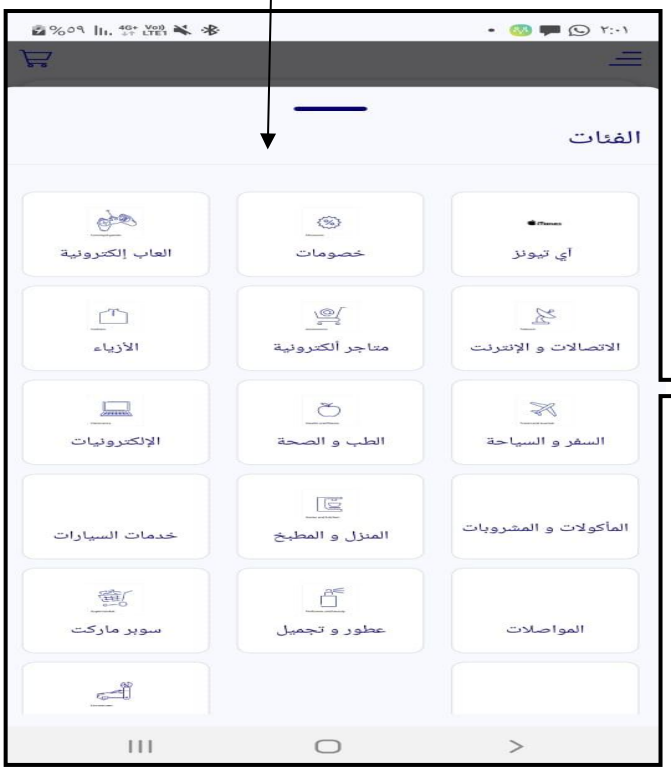
The size of the representative icons for services is small and not quite suitable

The advantages of **help page** interface are that it contains customer **support services**.
-To communicate with the customer if any problem.
-Currency exchange has become more easier to the customer.
-Privacy and security tips to avoid any kind of fraud.
It's good for the user to find all these advantages in one place.

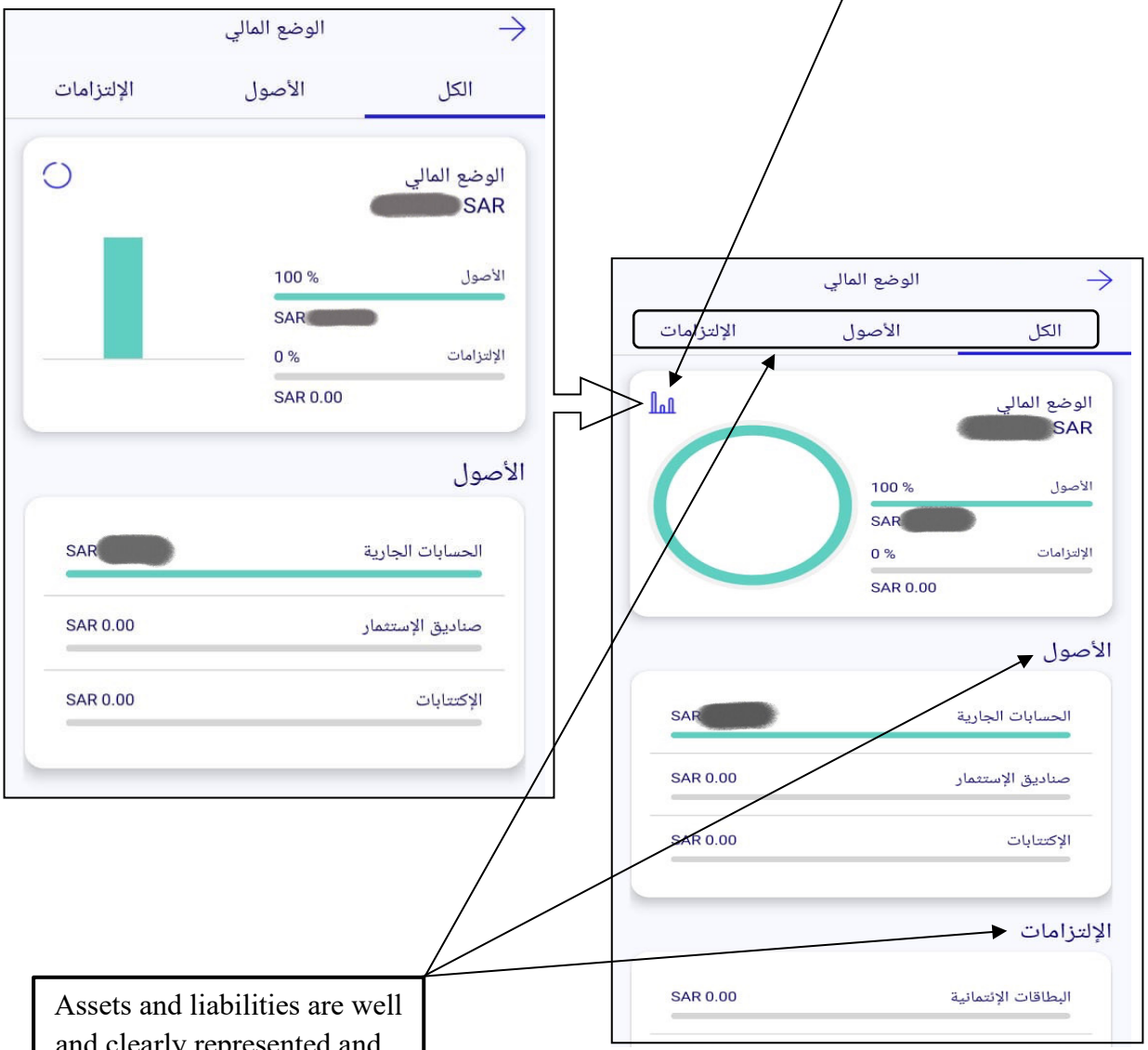
The disadvantages of the categories interface is that it is poorly shaped due to the lack of colors in it, and there is a lack of illustrations for categories such as the category "**Car Services**" and "**Transportation**".

Among the advantages, the user can also search for a specific product or other.

One of the advantages of the shopping interface is the presence of marketing ads.



One of the advantages when displaying financial data is that it may be presented to more than one graphic form to suit each user in understanding the chart



Assets and liabilities are well and clearly represented and grouped in “All” section more generally.

One of the advantages is that it reminds the user that the ID is about to expire

Distribution and selection of colors on the page is suitable and comfortable for the eyes

The way the division divided by the settings page are good

Putting the immediate option that most users are looking for quickly

5 Conclusion

In general, this report shows that there are risks (high score and dangerous) and warnings that threaten the security of the Al Rajhi Bank application .

The system must be protected from any threat that might harm it. This includes the tools, methods, and procedures that must be available to achieve protection from the risks that the system may face from the inside and outside.

Because programming is an ongoing process and requires adjustments from time to time. The code should be well documented because when a number of developers write code for the same program, the complexity increases. With the help of documentation, software developers can reduce the complexity by referring to the documentation of the code. In this way, it helps other developers to understand and use the program code correctly, and documenting the code makes it easier to reuse the code.

6 Reference

Link	page	Link	Page
https://www.codacy.com/product	7-14	https://docs.microsoft.com/en-us/xamarin/android/platform/files/external-storage?tabs=windows	21
https://github.com/marketplace/codacy	7-14	https://github.com/MobSF/owasp-mstg/blob/master/Document/0x05d-Testing-Data-Storage.md#external-storage	21
Android overlay malware and System Alert Window permission explained (nowsecure.com)	15	https://developer.android.com/training/data-storage#filesExternal	21
Manifest.permission | Android Developers	15	owasp-mstg/0x04g-Testing-Cryptography.md at master · MobSF/owasp-mstg · GitHub	22
https://www.liverpoolecho.co.uk/news/liverpool-news/what-really-happens-you-allow-15186255.amp	16	CWE - CWE-330: Use of Insufficiently Random Values (4.6) (mitre.org)	23
https://developer.android.com/training/location/permissions	18	https://www.geeksforgeeks.org/coupling-in-java/	23
https://developer.android.com/	18	https://www.decodejava.com/coupling-cohesion-java.htm	25
https://developers.google.com/	18	https://www.geeksforgeeks.org/cohesion-in-java/v	26
https://www.skcript.com/	19		
What is the use of android:exported="true" in BroadcastReceiver - Stack Overflow	20		
BroadcastReceiver Security Issue - Stack Overflow	20		