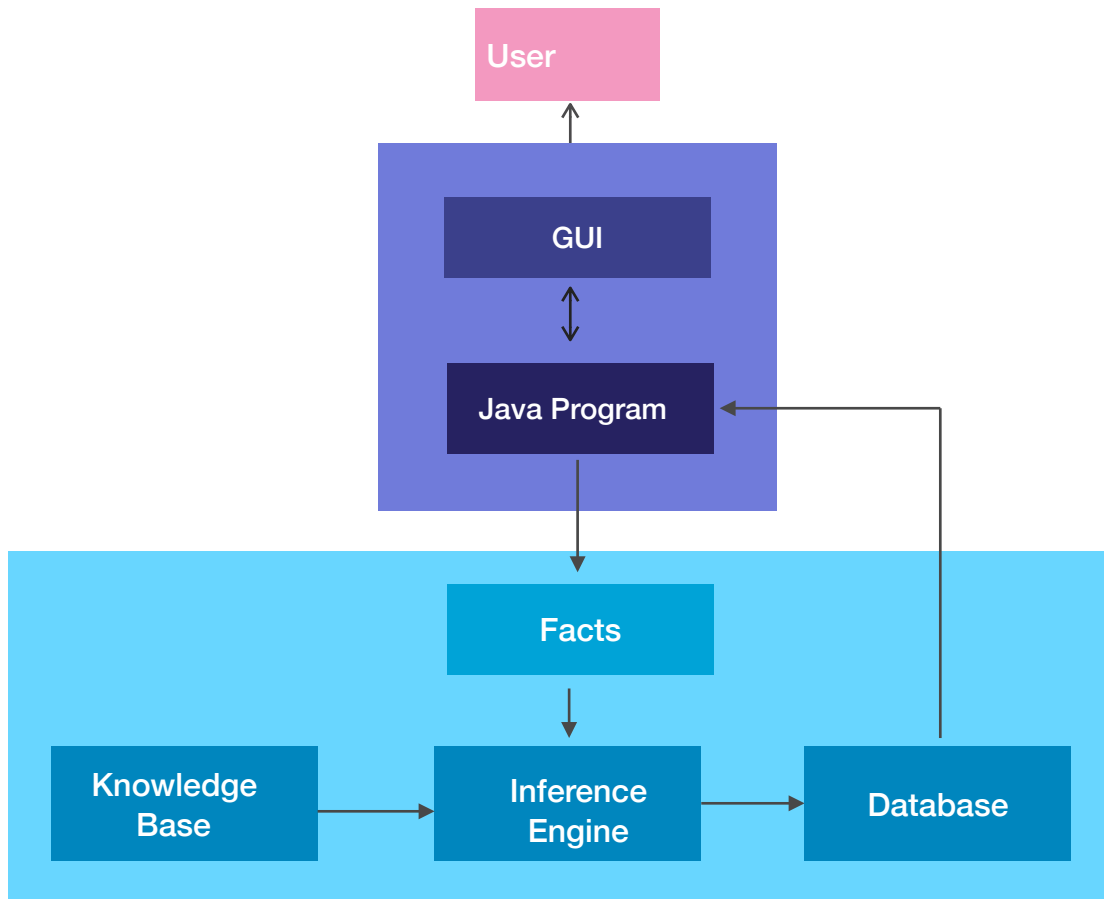# Implementation

A rule-based expert system is made up of five parts: knowledge base, database, inference engine, explanation facilities and user interface. CLIPS[1] provide knowledge base, database and inference engine parts of the rule-based expert system. Android application provides the explanation facilities and user interface which access the CLIPS[1] expert system. The flowchart of the application is as follows:

```
                          ┌──────────┐
                          │   User   │
                          └──────────┘
                               ↑
        ┌──────────────────────┼──────────────────┐
        │              ┌────────────────┐          │
        │              │      GUI       │          │
        │              └────────────────┘          │
        │                      ↕                    │
        │              ┌────────────────┐           │
        │              │  Java Program  │◄──────────┼───┐
        │              └────────────────┘           │   │
        └──────────────────────┼───────────────────┘   │
                               ↓                         │
   ┌───────────────────────────────────────────────────┼─────┐
   │                   ┌──────────────┐                 │     │
   │                   │    Facts     │                 │     │
   │                   └──────────────┘                 │     │
   │                          ↓                          │     │
   │ ┌───────────┐    ┌──────────────┐    ┌──────────┐  │     │
   │ │ Knowledge │───►│  Inference   │───►│ Database │──┘     │
   │ │   Base    │    │    Engine    │    │          │        │
   │ └───────────┘    └──────────────┘    └──────────┘        │
   └──────────────────────────────────────────────────────────┘
```

The knowledge base contains all the expert rules which is pre saved in .clp file by using CLIPS IDE. This file can be loaded again in the CLIPS[1] environment using load() command. The database stores all the facts. Facts can inserted using assert() command. The inference engine is in-built in CLIPS[1]. The user interface is designed in android using XML. Software used to develop the Android application is Android Studio[7]. The CLIPS[1] is integrated in the application using CLIPS4Android[5] library

which is based on CLIPS JNI[2]. The testing is performed using Android Emulator provided by the Android Studio[7]. The working of the application as follows:

1. The user launches the application.
2. The GUI showed in figure 4 appears.
3. When the user presses the Evaluate button, the button listener is executed.
4. In the button listener, first all the values for variables (like API level and Security Patch Level) is collected.
5. A new CLIPS[1] environment is created and expert rules are loaded in the environment.
6. The values of variables (API level, Security Patch Level, Device Lock Set Status, Root Access Status and Number of Applications with dangerous permissions) collected is added to the CLIPS[1] environment as facts.

Figure 4. Initial GUI after app launch.   Figure 5. Screen after evaluation process
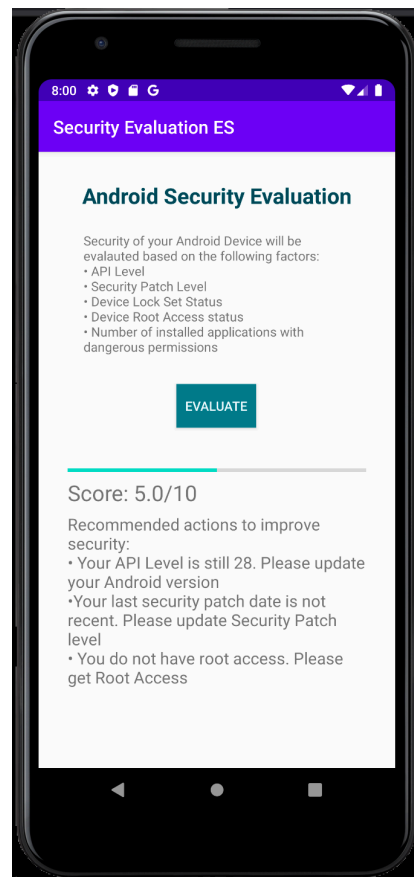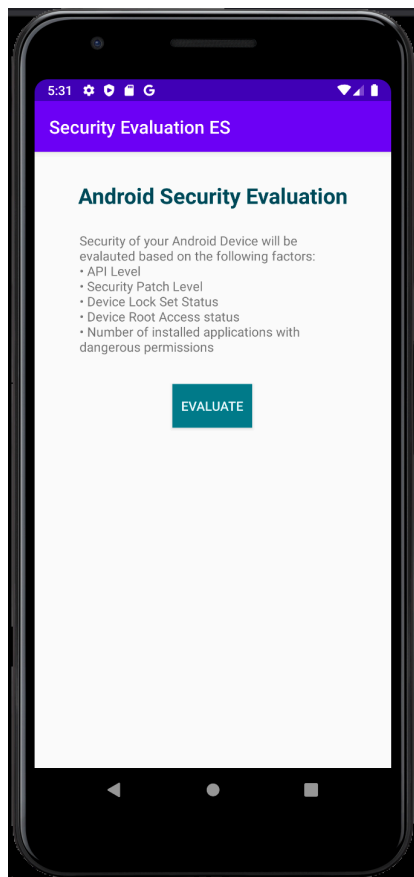
7. Then the inference engine in CLIPS[1] is run.

8. The respective rules from knowledge base will be fired and new facts will be generated in the database. These facts will hold all the values of risk rate factors like Android Security Risk, Software Security Risk, Hardware Security Risk and Application Security Risk.
9. The java program extracts relevant facts from the CLIPS[1] environment.
10. The final android security evaluation is displayed on the UI to the user along with recommendations to improve security. The figure 5 shows the final screen after evaluation.

Some of the rules in the knowledge base are:
1. IF LockSetStatus true AND RootAccessStatus true
   THEN HardwareSecurityRisk 0.0
2. IF LockSetStatus false AND RootAccessStatus true
   THEN HardwareSecurityRisk 50.0
3. IF LockSetStatus true AND RootAccessStatus false
   THEN HardwareSecurityRisk 50.0
4. IF LockSetStatus false AND RootAccessStatus false
   THEN HardwareSecurityRisk 100.0

The above rules show how Hardware Security Risk is calculated.

Similar expert rules can be formed to develop security evaluation expert system for other platforms like Windows machine, iPhone, etc. This application can be run on any android smartphone with API level 21 or more.

User's Guide

To run this application, a user need an android device or an android emulator. There are various android emulators available. But the AVD manager that comes with Android Studio[7] allows the user to create multiple virtual android devices with different specifications.

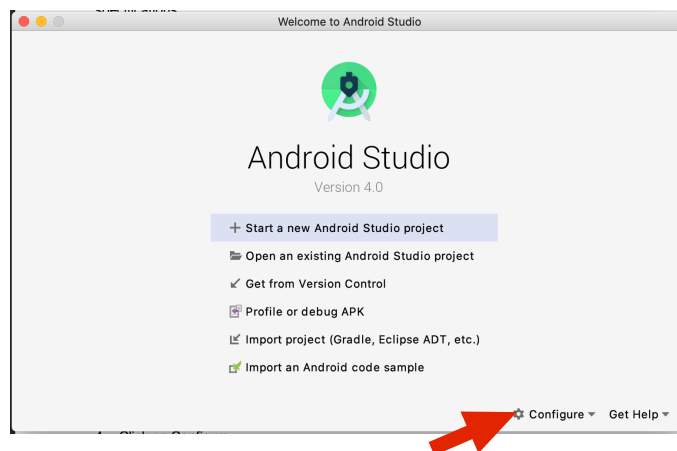Steps to install and run application on android smartphone:
1. Copy the apk provided to your android smartphone.
2. Make sure your device allows installing applications from unknown source.
3. Install the application on your device from the apk file.
4. Launch application.
5. Click Evaluation button.
6. The security score will be displayed on the progress bar.
7. The recommendations will appear right below the score.

Steps to install AVD manager on your laptop or pc:
1. Download Android Studio[7] from https://developer.android.com/studio
2. Install Android Studio[7] by following the step mentioned on this website: https://developer.android.com/studio/install

Steps to create new android virtual device:
1. Start Android Studio[7].
2. Click on Configure.

3. Select AVD Manager from the menu.
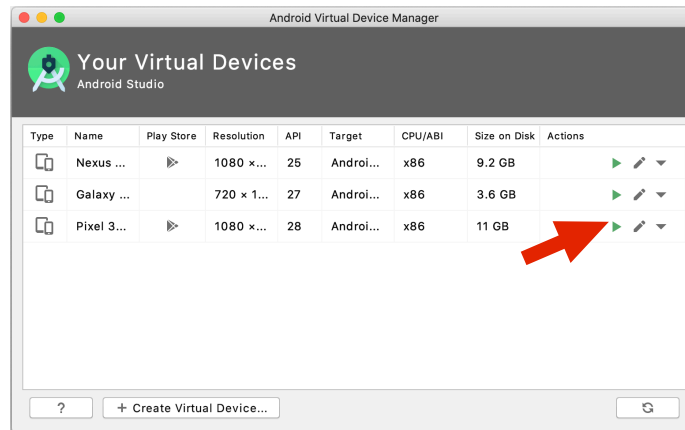4. Click on Create Virtual Device.

5. Create an android smartphone of your choosing.

Steps to install and run application on your virtual android device:
1. Open AVD Mananger.

2. Click on "Launch this AVD in emulator" button.

3. Select and drag the provided apk to the virtual android device.
4. The application will get installed automatically.
5. Launch application.
6. Click Evaluation button.
7. The security score will be displayed on the progress bar.
8. The recommendations will appear right below the score.

References

1. CLIPS http://www.clipsrules.net/
2. CLIPS JNI http://www.clipsrules.net/CLIPSJNI.html
3. Jess https://jess.sandia.gov/
4. JRuleEngine http://jruleengine.sourceforge.net/
5. CLIPS4Android https://github.com/gomezgoiri/CLIPS4Android
6. DROID-CLIPS https://github.com/DrItanium/DROID-CLIPS
7. Android Studio https://developer.android.com/studio