Rooting Checker

Development Document

2022-10-22

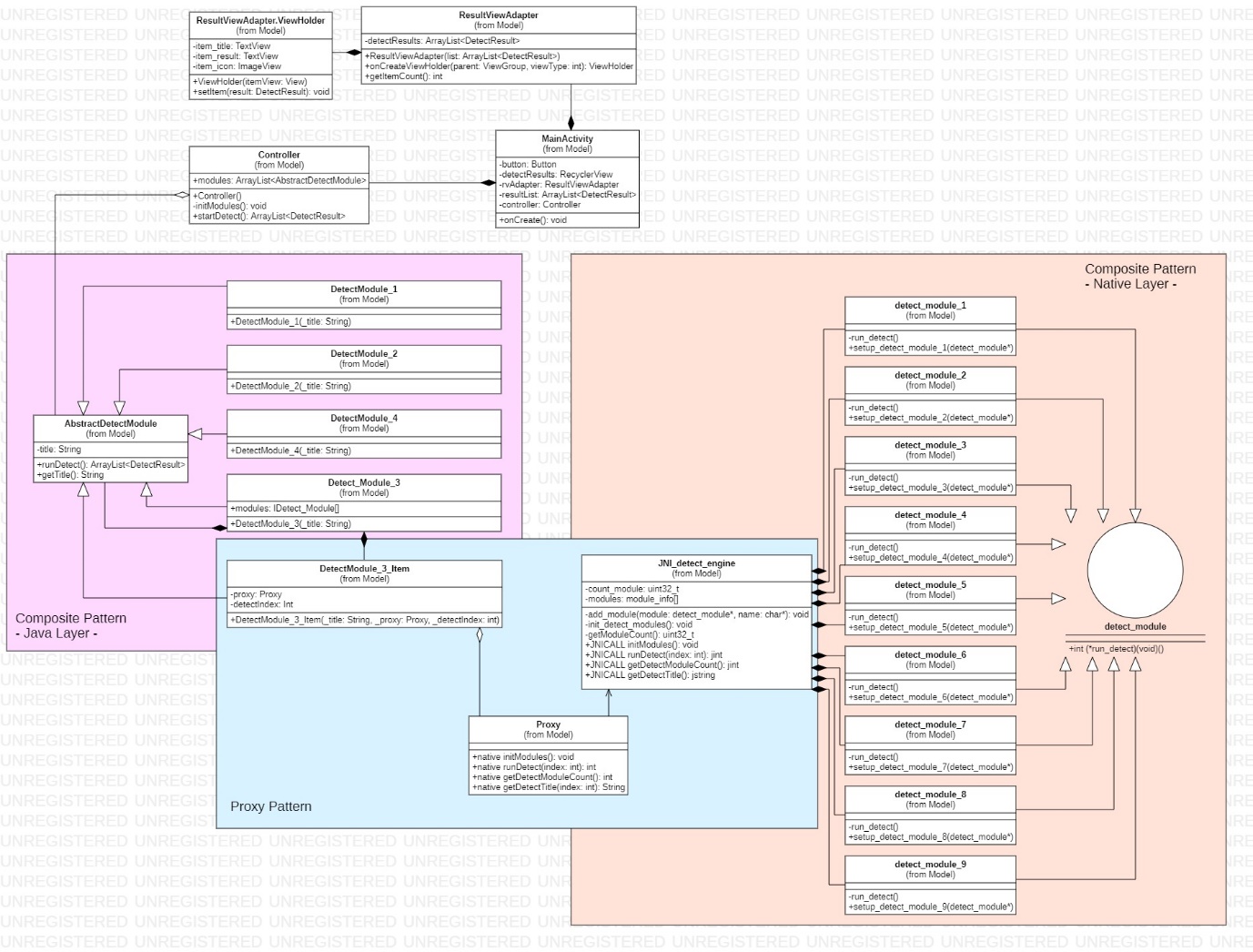
이성용

목차

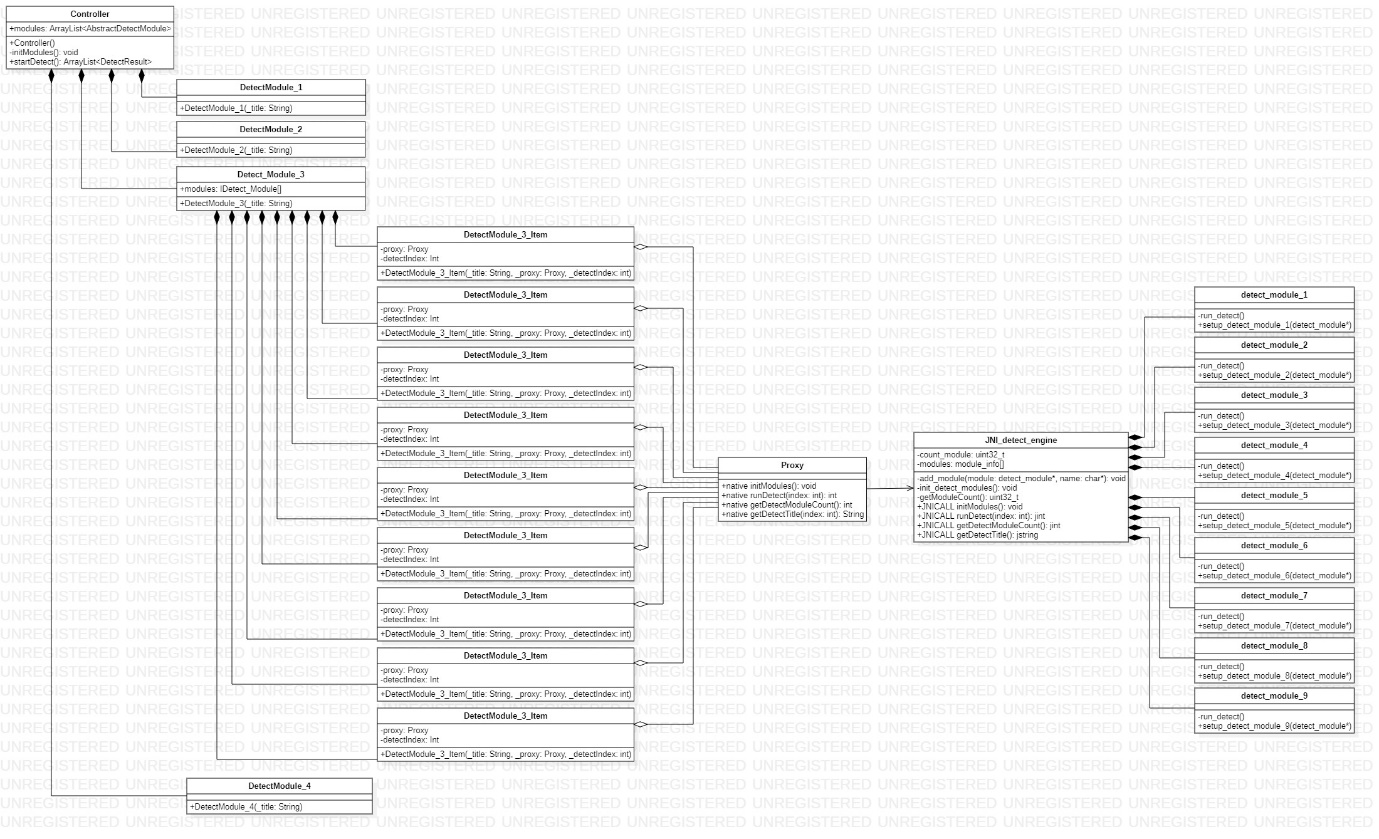
1. 개발 설계
   1. Class Diagram
   2. Dynamic View Class Diagram
   3. Sequence Diagram – Initialization
   4. Sequence Diagram – Start Detect
   5. Description
2. 개발
   1. UI
   2. Class Description
3. 개발 검증
   1. Description
   2. Native Layer Test Design
   3. Test Cases
   4. Test Result

**1. 개발 설계**

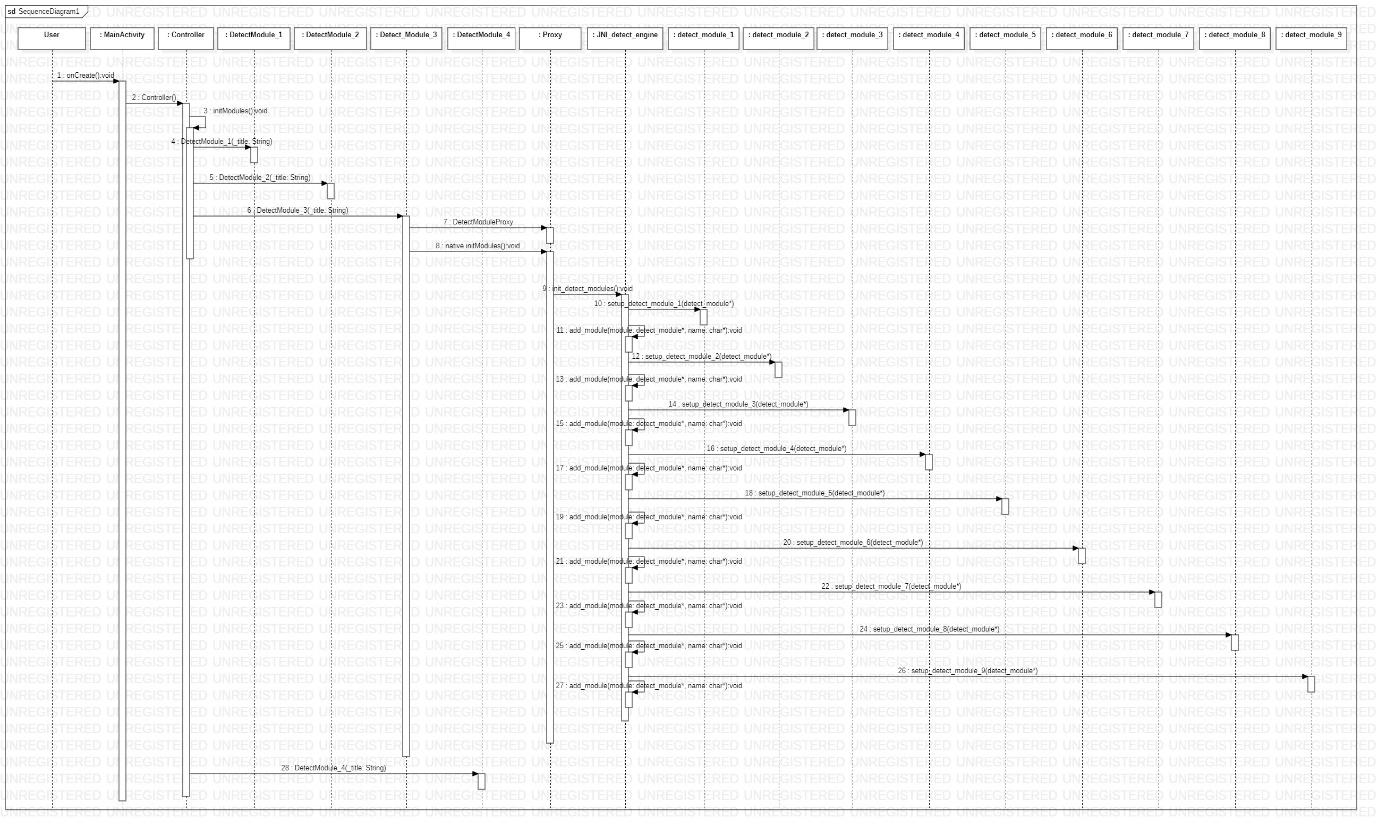
A. Class Diagram



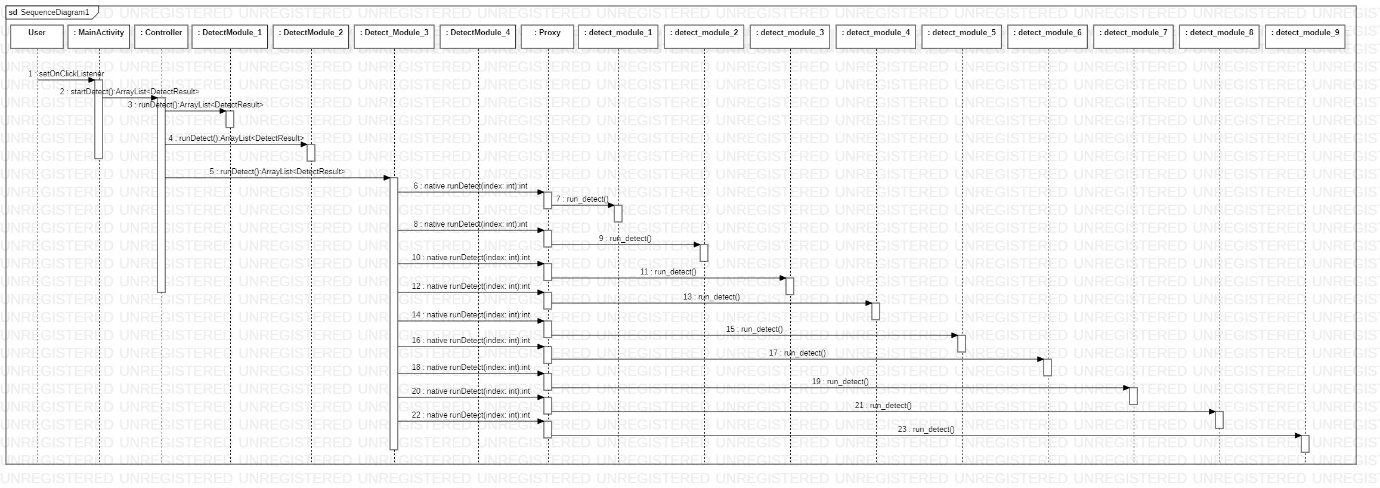
B. Dynamic View Class Diagram



C. Sequence Diagram – Initialization



D. Sequence Diagram – Start Detect



E. Description

Rooting Checker은 연속된 Detect Module이 동작할 수 있도록 설계하였고, Detect Module의 추가 또는 수정이 용이할 수 있도록 중점을 두어 설계를 진행하였다. 각각의 Detect Module은 Rooting을 Check할 수 있는 방법을 구현한 Module이며 Rooting Checker은 Detect Module들이 연속적으로 동작하면서 Rooting을 Check 한다. 이것은 Rooting 확인하는 새로운 방법이 새로 추가될 경우 혹은 방법이 변경될 경우 유지보수가 용이하도록 하였다.

Rooting Checker은 Native Layer의 Detect Module 그리고 Java Layer의 Detect Module을 모두 추가할 수 있으며, 각각은 Composite Pattern을 적용함으로 Detect Module의 추가 또는 수정 뿐 만 아니라, Detect Module들을 카테고리화 시켜 적용할 수도 있다.

Java Layer의 경우 각각의 DetectModule들은 Abstract Class인 “AbstractDetectModule”을 상속받아 구현함으로 Composite Pattern을 구현하였고, Native Layer의 경우 함수 포인터를 응용하여 각각의 DetectModule에서 detect\_module.h를 구현하게하여 Composite Pattern을 구현하였다.

Native Layer 개발자 또는 Java 개발자는 다른 Layer을 바라보지 않고 각 Layer에서 Detect Module을 구현하여 추가할 수 있다. 방법은 아래와 같다.

1. Java Layer
   1. AbstractDetectModule을 상속받아 Detect Module을 구현
   2. Controller Class의 InitModule함수에 신규 Detect Module을 추가
2. Native Layer
   1. detect\_module.h의 구현된 Structure을 구현
   2. JNI\_detect\_engine의 init\_detect\_module에 신규 Detect Module을 추가

Native Layer의 Detect Module들은 Proxy를 사용하여 Java Layer의 Detect Module을 통해 호출할 수 있다. 이것은 Run time 시 Java Layer의 Detect Module들이 Proxy를 통해 Native Layer의 Detect Module들을 호출할 수 있는 관계가 만들어짐으로 가능하게 된다.

Rooting Checker Design의 장점은 Rooting을 Check할 수 있는 Module들을 자유롭게 추가 또는 수정하여 유지보수가 용이하다는 점과 Java 개발자, Native 개발자는 각 Layer에만 집중하여 Detect Module을 개발할 수 있다는 것이다.

**2. 개발**

A. UI

|  |  |
| --- | --- |
| **Start App** | **RUN DETECT** |
|  |  |

B. Class Description

|  |  |  |
| --- | --- | --- |
| **Class** | **Function** | **Description** |
| **Java Layer** | | |
| AbstractDetectModule | | DetectModule의 Abstract Class |
|  | runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| Controller | | User의 명령 수행 |
|  | Controller | 생성자 |
| initModules | Module 집합 초기화 |
| startDetect | Detect 시작 |
| DetectModule\_1 | | Detect Module |
|  | DetectModule\_1 | 생성자 |
| runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| DetectModule\_2 | | Detect Module |
|  | DetectModule\_2 | 생성자 |
| runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| DetectModule\_3 | | Detect Module |
|  | DetectModule\_3 | 생성자 |
| runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| DetectModule\_3\_Item | | Detect Module |
|  | DetectModule\_3\_Item | 생성자 |
| runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| DetectModule\_4 | | Detect Module |
|  | DetectModule\_4 | 생성자 |
| runDetect | Detect 수행 |
| getTitle | Detect Title 반환 |
| DetectModuleProxy | | Java Layer와 Native Layer 연결 Proxy |
|  | native initModules | Native Layer Module 집합 초기화 |
| native runDetect | Detect 수행 |
| native getDetectModuleCount | Native Layer Module 수 반환 |
| native getDetectTitle | Detect Title 반환 |
| DetectResult | | Detect Result 결과 Class |
|  | DetectResult |  |
| ResultViewAdapter | | Result View에 대한 Adapter |
|  | onCreateViewHolder | onCreateViewHolder 함수 |
| onBindViewHolder | onBindViewHolder 함수 |
| getItemCount | View Item 수 반환 |
| ViewHolder | | View Holder Class |
|  | ViewHolder | 생성자 |
| setItem | 각 Item 설정 |
| **Native Layer** | | |
| detect\_engine.c | | Native Layer Module 제어 |
|  | add\_module | Module 추가 |
| init\_detect\_modules | Module 집합 초기화 |
| getModuleCount | Module 수 반환 |
| detect\_module\_1.c | | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_1 | Detect Module 반환 |
| detect\_module\_2.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_2 | Detect Module 반환 |
| detect\_module\_3.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_3 | Detect Module 반환 |
| detect\_module\_4.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_4 | Detect Module 반환 |
| detect\_module\_5.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_5 | Detect Module 반환 |
| detect\_module\_6.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_6 | Detect Module 반환 |
| detect\_module\_7.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_7 | Detect Module 반환 |
| detect\_module\_8.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_8 | Detect Module 반환 |
| detect\_module\_9.c |  | Detect Module |
|  | run\_detect | Detect 수행 |
| setup\_detect\_module\_9 | Detect Module 반환 |

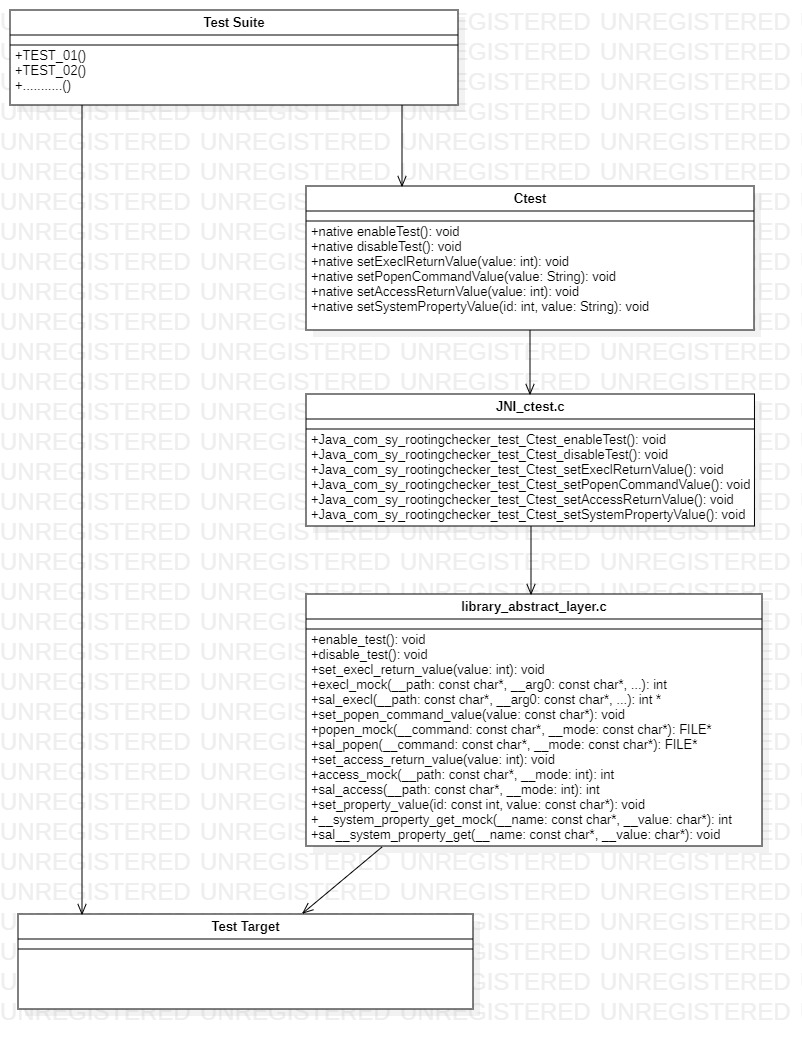
**3. 개발 검증**

A. Description

각 Detect Module에 대해 검증을 수행. Android Framework에서 제공하는 Unit Test를 최대한 활용하여 작성한다. Java Layer의 Detect Module들은 Junit을 활용하여 Unit Test로 작성하고, Native Layer의 Module들은 InstrumentedTest를 활용하여 작성한다.

각 Detect Module들은 Java Native 함수 또는 C Native 함수를 활용하여 작성된 경우가 많다. 따라서, 각 Detect Module들이 Logic을 검증하기 위해서는 각 Layer의 Native 함수들을 Mocking하여 결과값을 제어할 필요가 있다. Java Layer의 경우 Junit과 같이 사용할 수 있는 mockito 또는 PowerMockito를 활용하여 사용하면 된다. 하지만 C로 작성되는 Native Layer의 경우, cpputest 등 Test Framework을 활용할 수 있으나, 본 과제에서는 최대한 개발 Tool인 Android Studio를 활용하고자 Native Layer를 Mocking 하여 Test할 수 있는 Class를 개발하여 검증을 진행한다.

B. Native Layer Test Design



Test Target에서의 구현은 Mocking할 API는 library\_abstract\_layer에 구현된 API로 대체하여 구현한다. 즉, Test Target에서 C Library중 access 함수를 사용한다면 해당 함수는 library\_abstract\_layer의 sal\_access함수로 대체한다. library\_abstract\_layer에서 제공하는 함수들은 테스트 설정 여부에 따라 실제 C Library 함수 또는 Mocking 함수로 변환 제공된다.

Test Suite에서 Test Target에 대해 테스트를 수행하는 절차는 아래와 같다.

1. Ctest를 통해 enableTest를 호출한다.
2. Ctest를 통해 Mocking 함수에서 원하는 값을 설정한다.
3. Test Target의 함수를 호출하여 테스트를 진행한다.

C. Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Case** | | **Description** |
| **C Layer** | | |
| **detect\_module\_1\_test** | | |
| 1 | run\_detect\_Not\_Execute\_su\_NOT\_DETECT | Execute Su 실행되지 않음 |
| 2 | run\_detect\_Execute\_su\_DETECT | Execute Su 실행 |
| **detect\_module\_2\_test** | | |
| 3 | run\_detect\_Exist\_which\_su\_DETECT | Which su 실행 |
| 4 | run\_detect\_Not\_Exist\_which\_su\_NOT\_DETECT | Which su 실행되지 않음 |
| **detect\_module\_3\_test** | | |
| 5 | run\_detect\_Exist\_Su\_Files\_DETECT | Su File들 존재 |
| 6 | run\_detect\_Not\_Exist\_Su\_Files\_NOT\_DETECT | Su File들 존재하지 않음 |
| **detect\_module\_4\_test** | | |
| 7 | run\_detect\_Exist\_Xpose\_Files\_DETECT | Xpose File들 존재 |
| 8 | run\_detect\_Not\_Exist\_Xpose\_Files\_NOT\_DETECT | Xpose File들 존재하지 않음 |
| **detect\_module\_5\_test** | | |
| 9 | run\_detect\_Check\_dev\_keys\_DETECT | Build Tag가 dev-keys |
| 10 | run\_detect\_Check\_dev\_keys\_NOT\_DETECT | Build Tag가 dev-keys가 아님 |
| **detect\_module\_6\_test** | | |
| 11 | run\_detect\_Check\_dev\_keys\_DETECT | Build Tag가 dev-keys |
| 12 | run\_detect\_Check\_dev\_keys\_NOT\_DETECT | Build Tag가 dev-keys가 아님 |
| **detect\_module\_7\_test** | | |
| 13 | run\_detect\_Check\_RO\_DEBUGGABLE\_DETECT | RO\_DEBUGGABLE 1 |
| 14 | run\_detect\_Check\_RO\_DEBUGGABLE\_NOT\_DETECT | RO\_DEBUGGABLE 0 |
| 15 | run\_detect\_Check\_SERVICE\_ADB\_ROOT\_DETECT | SERVICE\_ADB\_ROOT 0 |
| 16 | run\_detect\_Check\_SERVICE\_ADB\_ROOT\_NOT\_DETECT | SERVICE\_ADB\_ROOT 1 |
| 17 | run\_detect\_Check\_RO\_SECURE\_DETECT | RO\_SECURE 1 |
| 18 | run\_detect\_Check\_RO\_SECURE\_NOT\_DETECT | RO\_SECURE 0 |
| 19 | run\_detect\_Check\_SYS\_INITD\_DETECT | SYS\_INITD 0 |
| 20 | run\_detect\_Check\_SYS\_INITD\_NOT\_DETECT | SYS\_INITD 1 |
| 21 | run\_detect\_Check\_RO\_BUILD\_SELINUX\_DETECT | RO\_BUILD\_SELINUX 0 |
| 22 | run\_detect\_Check\_RO\_BUILD\_SELINUX\_NOT\_DETECT | RO\_BUILD\_SELINUX 1 |
| **detect\_module\_8\_test** | | |
| 23 | run\_detect\_Exist\_Magisk\_File\_DETECT | Magisk File 존재 |
| 24 | run\_detect\_Not\_Exist\_Magisk\_File\_NOT\_DETECT | Magisk File 존재하지 않음 |
| **detect\_module\_9\_test** | | |
| 25 | run\_detect\_Exist\_Busybox\_Files\_DETECT | Busybox File 존재 |
| 26 | run\_detect\_Not\_Exist\_Busybox\_Files\_NOT\_DETECT | Busybox File 존재하지 않음 |
| **Java Layer** | | |
| **DetectModule\_1\_UnitTest** | | |
| 1 | runDetect\_ExistSuperuserAPK\_DETECT | SuperuserAPK 존재 |
| 2 | runDetect\_NotExistSuperuserAPK\_NOT\_DETECT | SuperuserAPK 존재하지 않음 |
| 3 | getTitle\_SuperAPK\_Success | getTile 함수 확인 |
| **DetectModule\_2\_UnitTest** | | |
| 4 | runDetect\_Build\_Tag\_Equals\_test\_keys\_DETECT | Build Tag가 test key |
| 5 | runDetect\_Build\_Tag\_Not\_Equals\_test\_keys\_NOT\_DETECT | Build Tag가 test key가 아님 |
| 6 | getTitle\_test\_keys\_Success | getTile 함수 확인 |
| **DetectModule\_4\_UnitTest** | | |
| 7 | runDetect\_CheckMount\_DETECT | Mount RO 존재 |
| 8 | runDetect\_CheckMount\_NOT\_DETECT | Mount RO 존재하지 않음 |
| 9 | getTitle\_test\_keys\_Success | getTile 함수 확인 |

D. Test Result

