PROJECT DOCUMENT

System Model (Sequence Diagram Document

AOSP 조

조원 : 김재현, 이송무, 임진현

지도교수: 조은선

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2020/05/15	Design Pattern	이송무
2	2020/05/15	Testing Application	김재현
3	2020/05/16	Error handling	임진현

Table of Contents

List of Figure

1. Introduction

1.1. Objective

이 문서는 개선된 안드로이드 시스템의 시스템 모델(시퀀스 다이어그램)에 대한 내용을 기술하고 있다. 요구사항 명세 단계에서 작성한 유스케이스 다이어그램을 기반으로 각 유스케이스의 상세한 내부 동작 흐름을 시퀀스 다이어그램으로 모델링한다.

2. Use Case Diagram

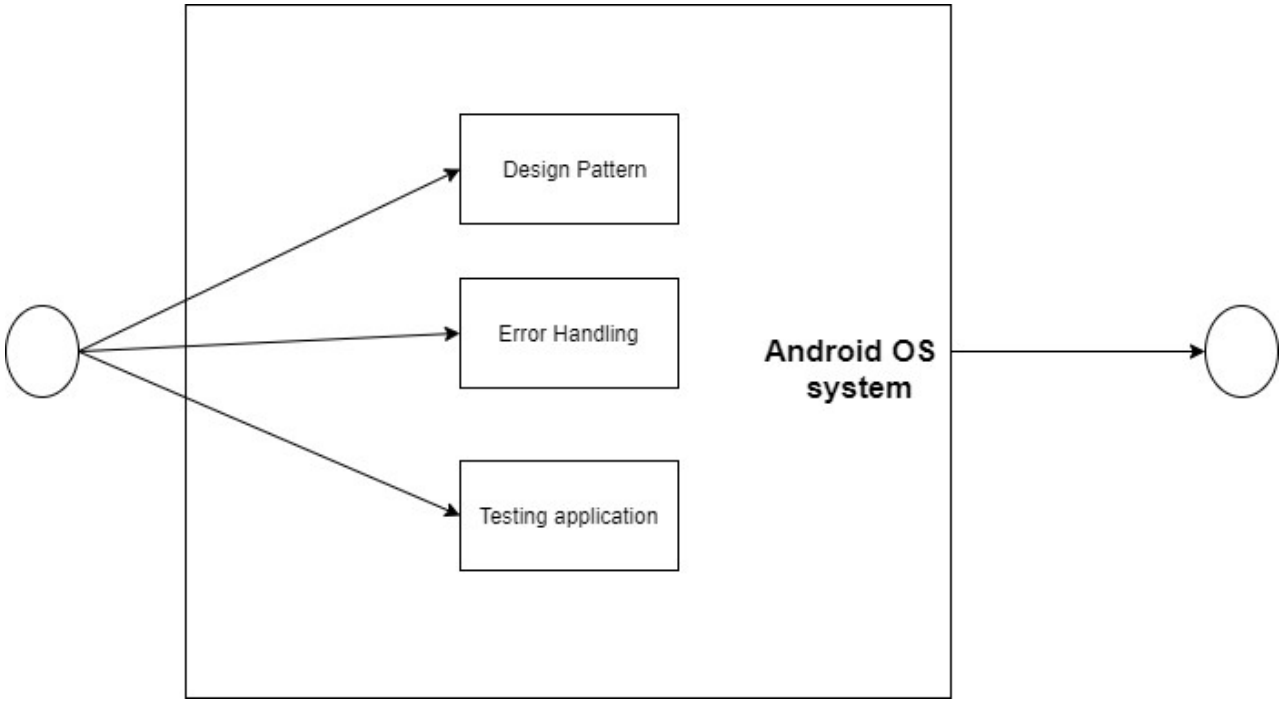


Figure 1 – Use Case Diagram

3. Sequence Diagram

3.1. Design Pattern

Design Pattern 는 안드로이드 시스템의 android.view.WindowManager.java 에서 성능이 의심되는클래스의 디자인 패턴을 개선한것이다. System 이 LayoutParams 객체를 호출하면, LayoutParams 가 내부 클래스인 Builder 클래스를 호출하여 Layout 관련 변수들을 initialize 를 하고, build() 메소드를 마지막에 호출하면 initialized 변수들을 포함하는 LayoutParams

객체를 리턴해준다.

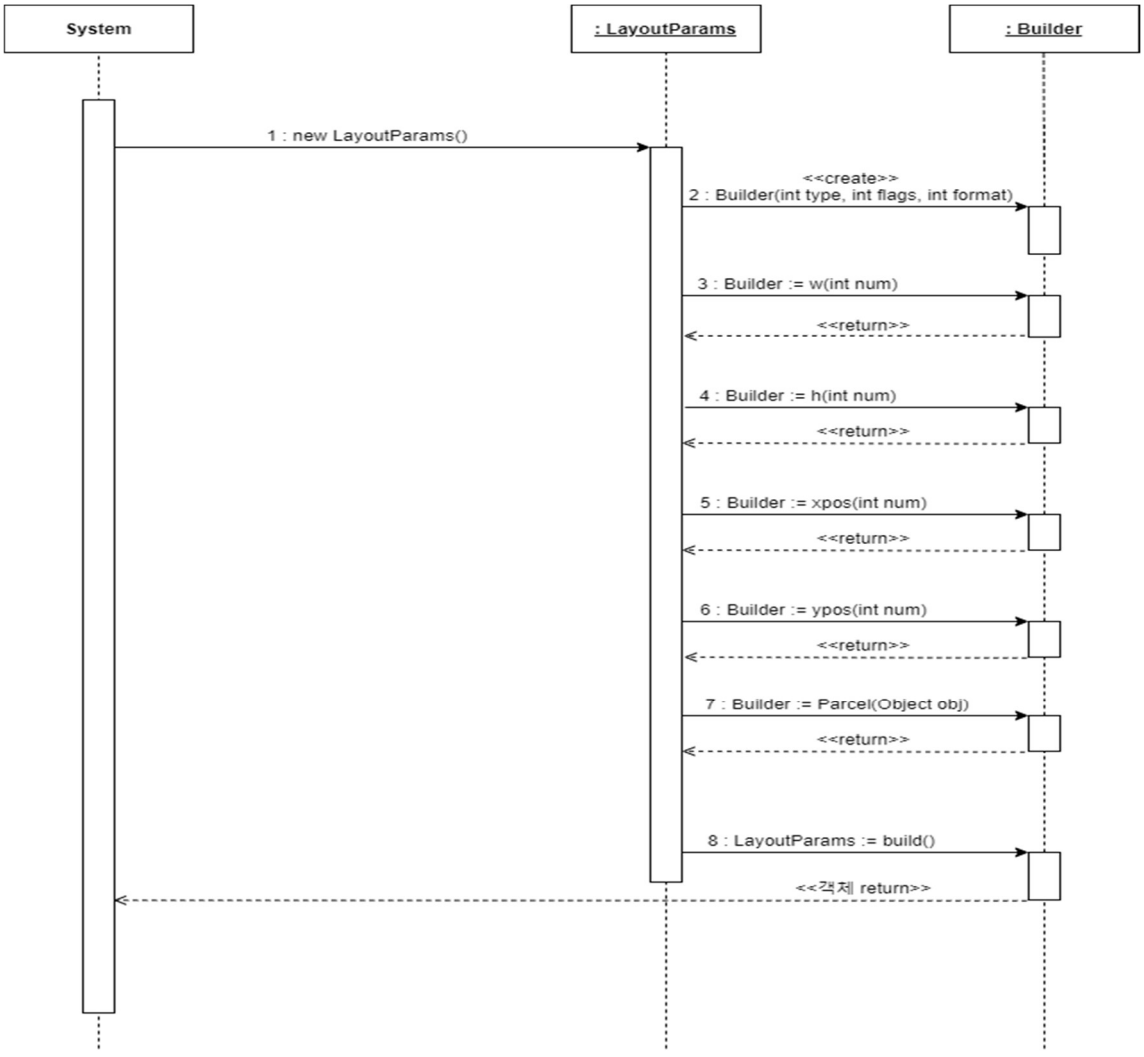


Figure 2 – Design Pattern Sequence Diagram

1. System 이 새로운 LayoutParams 객체를 생성한다.

2. LayoutParams 객체는 필수 인자들을 매개변수로 하는 Builder 생성자를 호출한다.

3. Builder 클래스의 w 메소드는 주어진 num 인자 값으로, 변수 w 를 초기화 시킨 후, Builder 클래스의 변수로 리턴한다.

4. Builder 클래스의 h 메소드는 주어진 num 인자 값으로, 변수 h 를 초기화 시킨 후, Builder 클래스의 변수로 리턴한다.

5. Builder 클래스의 xpos 메소드는 주어진 num 인자 값으로, 변수 xpos 를 초기화 시킨 후, Builder 클래스의 변수로 리턴한다.

6. Builder 클래스의 ypos 메소드는 주어진 num 인자 값으로, 변수 ypos 를 초기화 시킨 후, Builder 클래스의 변수로 리턴한다.

7. Builder 클래스의 Parcel 메소드는 주어진 obj 객체로, 객체 obj 를 초기화 시킨 후, Builder 클래스의 변수로 리턴한다.

8. Builder 클래스의 build 메소드는 LayoutParams 의 형태로 객체를 시스템에 리턴해준다.

3.2. Testing Application

테스팅 어플리케이션은 AOSP(Android Open Source Project) 에서의 성능 개선을 한 클래스 코드 들을 수정을 한 것을 포팅 혹은 빌드된 안드로이드 기기에서 얼마나 빠른 성능 속도가 개선이 되어 있는 지를 측정을 하는 것이다. 또한 서로 간의 결합성을 낮추기 위하여 안드로이드 MVVM 패턴 변형 적용

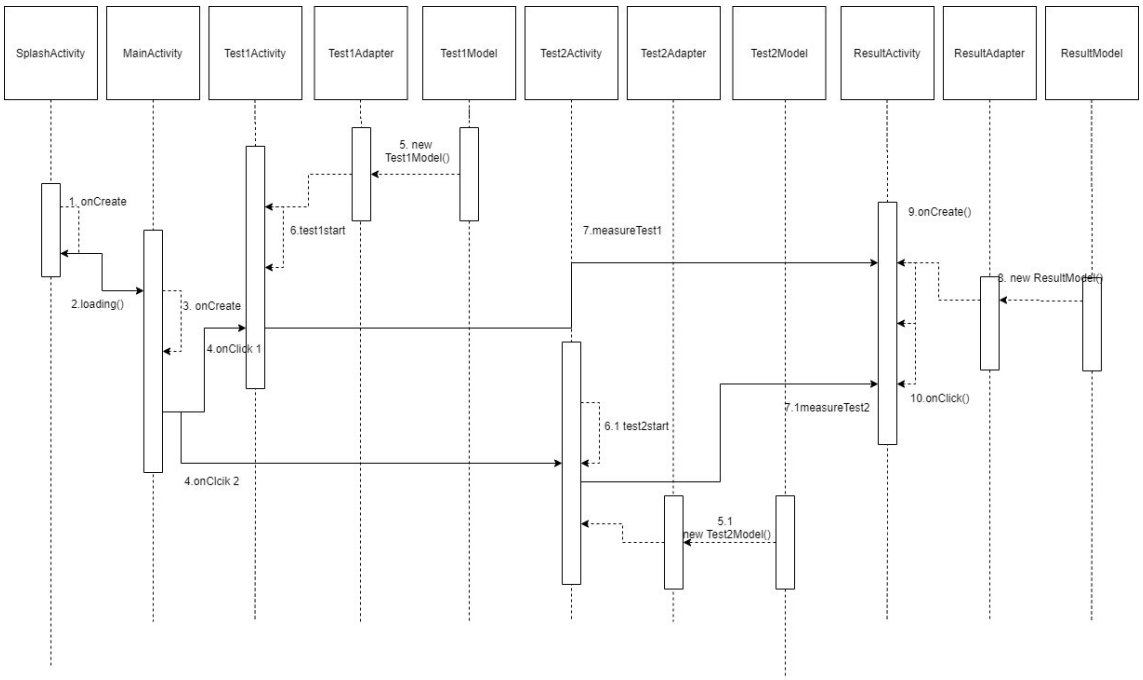


Figure 3 – Testing Application Sequence Diagram

1. SplashActivity Android Lifecycle 이용 초기화

2. SplashActivity -> MainActivity Intent 를 통한 액티비티 이동

3. MainActivity Android Lifecycle 이용 초기화

4. MainActivity -> Test1Activity Intent 를 통한 액티비티 이동

4.1. MainActivity -> Test2Activity Intent 를 통한 액티비티 이동

5. Test1Model data class instance 생성

5.1. Test2Model data class instance 생성

6. test1Start() Test1Activity 측정 항목 테스트 시작

6.1. test2Start() Test2Activity 측정 항목 테스트 시작

7. measureTest1() test1 측정 항목 ResultActivity intent 로 데이터 실어서 액티비티 이동

7.1. measureTest2() test2 측정 항목 ResultActivity intent 로 데이터 실어서 액티비티 이동

8. ResultModel data class instance 생성

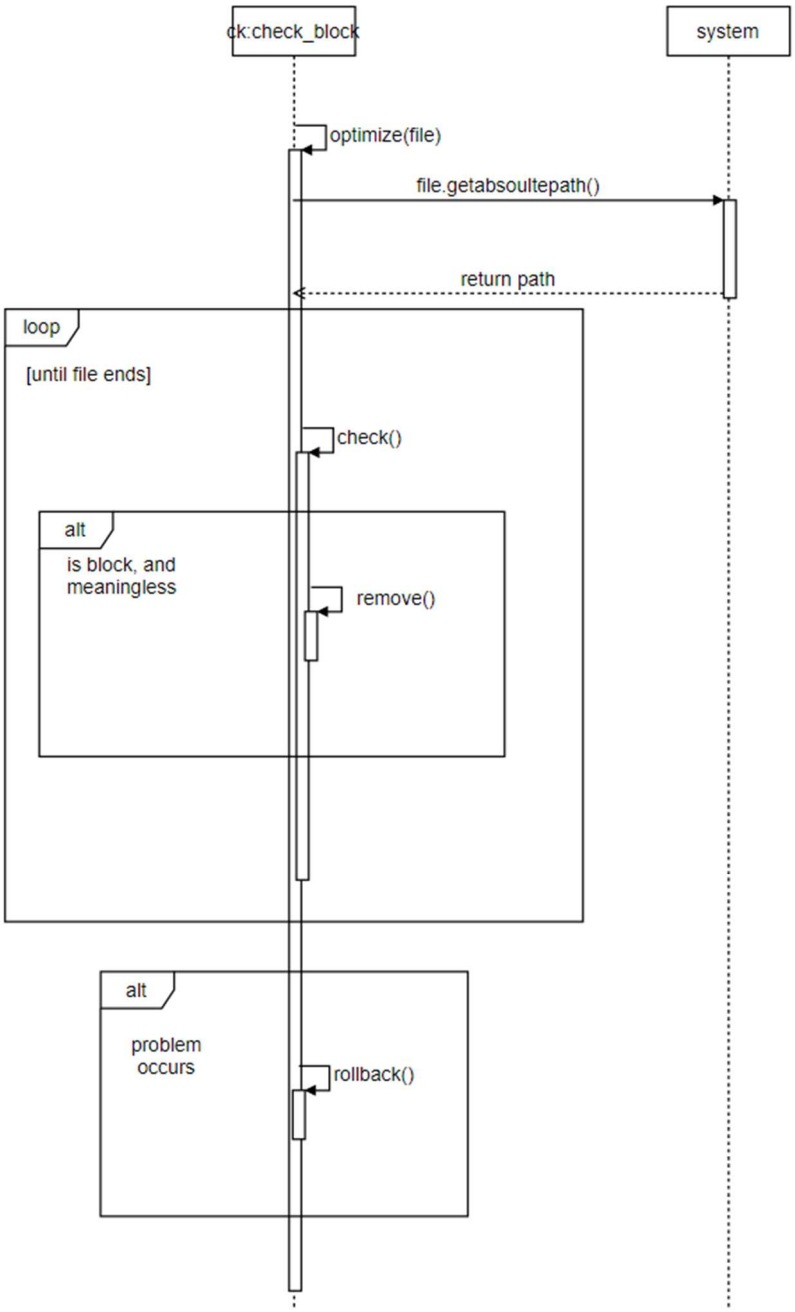
9. ResultActivity Android LifeCycle 을 이용한 초기화

10. onClick 테스트 마무리 및 결과 표시

3.3. Error Handling

별 의미없는 try catch 문을 찾아 없애는것을

수행해주는 class



1. ck 클래스에서 optimize(file) 메소드 호출

2. 파일경로를 받아 파일을 가져오기위해 file.absoultepath 사용

3. 파일이 끝날때까지 check loop

4. try 문을 발견했고, 별의미 없으면 remove

5. 문제발생시 rollback