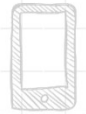
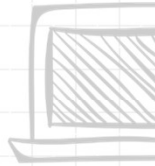
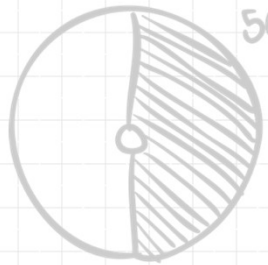


OOPT 3rd Cycle

Team 4 용사여, 일어나시게



Content

- 1. Specification Review
- 2. PMD Result - Bugs
- 3. PMD Result – Vulnerabilities
- 4. After Modification
- 5. Closing

1. *Specification Review*

Stage 1000 Planning → 수정 및 검증 완료

- 수정 사항 : Set Summer Time 제거

Stage 2030 Analysis → 수정 및 검증 완료

- 수정 사항 : Show realtime → Change Type, Bell section → section, Sun mode 설명 추가

Stage 2040 Design → 수정 및 검증 완료

- 수정 사항 : 2030번과 동일

2. PMD Result – Overview

PMD Summary

✓ PMD 결과

> Bugs : 10개 발생

■ "InterruptedException" should not be ignored 3개

- src/main/java/Alarm.java
 - Either re-interrupt this method or throw the "InterruptedException". 34 minutes ago • L228 % T +
 - Bug • Major • Open • Not assigned • 15min effort Comment [cwe, error-handling, multi-threa...](#)
 - Either re-interrupt this method or throw the "InterruptedException". 34 minutes ago • L266 % T +
 - Bug • Major • Open • Not assigned • 15min effort Comment [cwe, error-handling, multi-threa...](#)
- src/main/java/TimeThread.java
 - Either re-interrupt this method or throw the "InterruptedException". 34 minutes ago • L13 % T +
 - Bug • Major • Open • Not assigned • 15min effort Comment [cwe, error-handling, multi-threa...](#)

■ Correctness - Method accesses list or array with constant index 2개

- src/main/java/Sun.java
 - Method Sun.realTimeTask(Sun) accesses list or array with constant index 23 minutes ago • L102 % T +
 - Bug • Major • Open • Not assigned • Comment [correctness](#)
 - Method Sun.init(Sun) accesses list or array with constant index 23 minutes ago • L149 % T +
 - Bug • Major • Open • Not assigned • Comment [correctness](#)

■ "Thread.run()" should not be called directly 2개

- src/main/java/Bei.java
 - Call the method Thread.start() to execute the content of the run() method in a dedicated thread. 24 minutes ago • L34 % T +
 - Bug • Major • Open • Not assigned • 20min effort Comment [cert, cwe, multi-threading](#)
- src/main/java/Main.java
 - Call the method Thread.start() to execute the content of the run() method in a dedicated thread. 24 minutes ago • L10 % T +
 - Bug • Major • Open • Not assigned • 30min effort Comment [cert, cwe, multi-threading](#)

■ Correctness - Class defines fields that are used only as locals 1개

- src/main/java/Alarm.java
 - Class Alarm defines fields that are used only as locals 25 minutes ago • L60 % T +
 - Bug • Major • Open • Not assigned • Comment [correctness](#)

■ Correctness - Class has a circular dependency with other classes 1개

- src/main/java/WatchSystem.java
 - Class WatchSystem has a circular dependency with other classes 26 minutes ago • L12 % T +
 - Bug • Major • Open • Not assigned • Comment [correctness](#)

■ Correctness - Constructor makes call to non-final method 1개

- src/main/java/Sun.java
 - Constructor new Sun(RealTime) makes call to non-final method 27 minutes ago • L74 % T +
 - Bug • Major • Open • Not assigned • Comment [correctness](#)

> Vulnerabilities : 52개 발생

■ Critical : 15개 발생

- ◆ Security - Method returns internal array 11개
- ◆ Security - array is stored directly 2개
- ◆ Handling files is security-sensitive 2개

■ Minor : 37개 발생

> Code Smells : 4.2K개 발생

Findbugs Summary

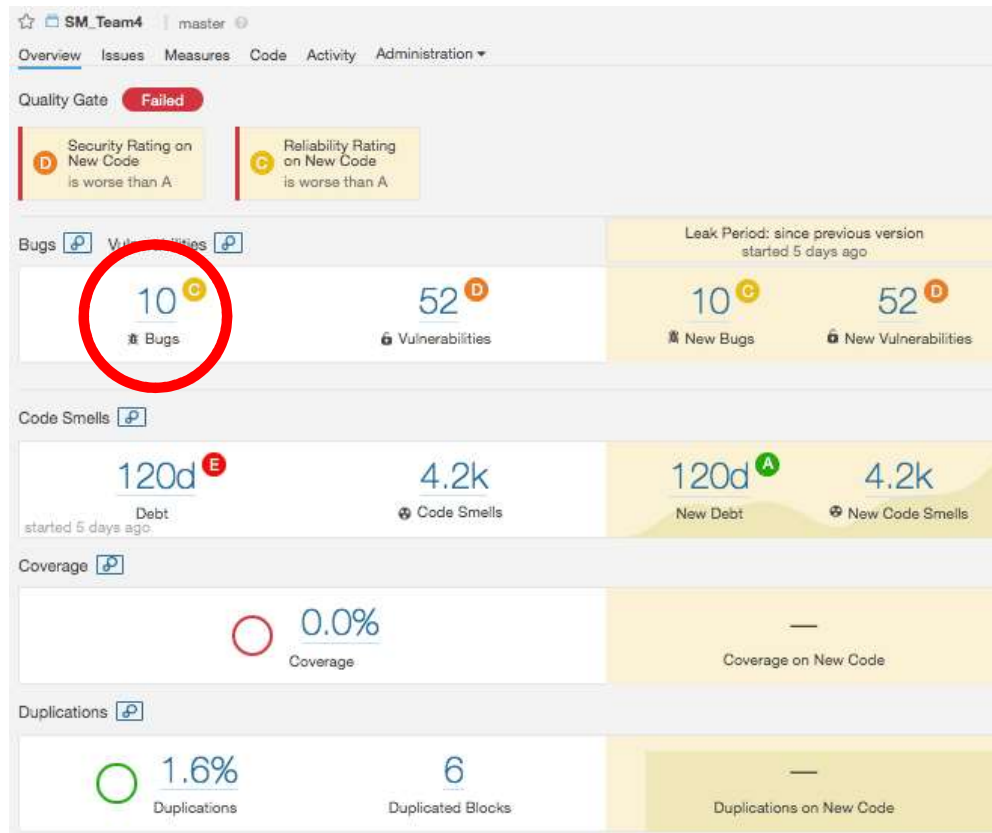
- ✓ Findbugs 결과 14개의 code smell이 확인되었고 발생한 code smell 은 크게 세 종류로 나누어진다.
 - > Return value : call by reference
 - > Storing externally mutable object
 - > Default case missing

1. Return value : call by reference

- src/main/java/Alarm.java
 - Alarm.showAlarm() may expose internal representation by returning displayAlarmData 54 minutes ago • L321 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
 - Alarm.getReserved() may expose internal representation by returning reservedAlarm 54 minutes ago • L327 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
 - Alarm.getFrequency() may expose internal representation by returning frequency 54 minutes ago • L331 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
 - Alarm.getRepeat() may expose internal representation by returning repeat 56 minutes ago • L335 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/ModeSetting.java
 - ModeSetting.showModeSetting() may expose internal representation by returning displayModeData 58 minutes ago • L147 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/RealTime.java
 - RealTime.showRealTime() may expose internal representation by returning displayRealTimeData 59 minutes ago • L132 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/Stopwatch.java
 - Stopwatch.showStopwatch() may expose internal representation by returning displaySWData 59 minutes ago • L82 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/Sun.java
 - Sun.showSun() may expose internal representation by returning displaySunDisplay 1 hour ago • L178 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/Timer.java
 - Timer.showTimer() may expose internal representation by returning displayTimerData 1 hour ago • L166 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)
- src/main/java/WorldTime.java
 - WorldTime.showWorldTime() may expose internal representation by returning displayWTData 1 hour ago • L143 % T +
 - Code Smell • Info • Open • Not assigned • 30min effort Comment [malicious-code](#)

2. PMD Result - Bugs

◆ Before Modification



2. PMD Result - Bugs

Bugs #1~#3 : "InterruptedException" should not be ignored

- Thread.interrupt을 호출해서 re-interrupted로 해결

```
- try { Thread.sleep(3000);} catch(InterruptedException e) {e.printStackTrace();}
+ /* [sonarqube][Bug #2] */
+ try { Thread.sleep(3000);} catch(InterruptedException e) {
+     e.printStackTrace();
+     Thread.currentThread().interrupt();
+ }
```

Bugs #4,5 : Method accesses list of array with constant index

- 배열에 직접 접근이 아닌 접근자를 이용하는 것으로 대체

```
- if(this.currTime.getTimeInMillis() >= this.sun[1].getTimeInMillis()){
+ /* [sonarqube][Bug #4] */
+ if(this.currTime.getTimeInMillis() >= this.getSun(1).getTimeInMillis()){
```

Bugs #6,7 : "Thread.run()" should not be called directly

- Thread.start() 하는 것으로 대체

```
- this.run();
+ if(this.getState() == State.NEW)
+     this.start();
```

Bugs #8 : Class defined fields that are used only as locals

- 객체변수를 지역변수로 바꿔서 해결

```
- this.currTime = this.realTime.requestRealTime();
+ /* [sonarqube][Bug #8] */
+ Calendar currTime = this.realTime.requestRealTime();
+ //this.currTime = this.realTime.requestRealTime();
```

Bugs #10 : Constructor makes call to non-final method

- 생성자에서 실행시키는 메소드는 final 메소드로 변경

```
- public void initSun(){
+ /* [sonarqube][Bug #10] */
+ public final void initSun(){
```

2. PMD Result - Bugs

- ♦ Limitation – Bug #9(Class has a circular dependency with other classes)

```
public WatchSystem(){
    this.menu = new ArrayList();
}

Class WatchSystem has a circular dependency with other classes ***
Bug Major Open Not assigned

menu.add(new ModeSetting(this));
menu.add(new RealTime());
menu.add(new SettingTime((RealTime)menu.get(1)));
menu.add(new Stopwatch());

/* [sonarqube][Bug #9] */
try{

    menu.add(new Timer());
    menu.add(new Alarm((RealTime)menu.get(1)));
}
catch(Exception e) {
    /* [sonarqube][Use a logger to log this exception] */
    Logger logger = Logger.getLogger(WatchSystem.class.getName());
    logger.log(Level.ALL, "context", e);
    return;
}

this.currMode = 1; // [currMode] 1: Always RealTime
this.maxCnt = 4;

watchGUI = new WatchGUI(this);
watchGUI.setMode(menu.get(1));
watchGUI.designMode(true);
}
```

Bugs <9>

- 문제 발생 위치
 - src/main/java/WatchSystem.java
- 문제 내용
 - Correctness – Class has a circular dependency with other classes
 - ◆ 해당 클래스가 다른 클래스와 순환적으로 의지하고 있다.

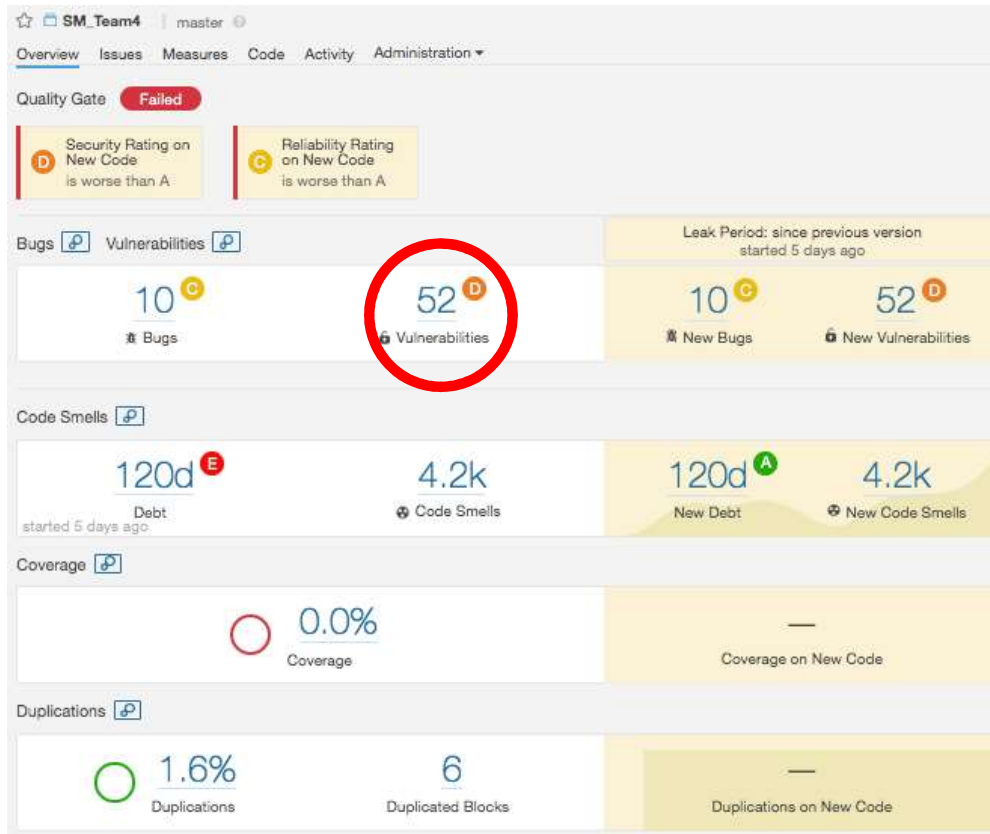
```
public WatchSystem() throws UnsupportedOperationException, IOException, LineUnavailableException{

Class WatchSystem has a circular dependency with other classes *** 4 hours ago L12 %
Bug Major Open Not assigned Comment correctness
```

- 해결 방법
 - 의존성을 없애기 위해 인터페이스를 사용한다.

3. PMD Result – Vulnerabilities

◆ Before Modification



3. PMD Result – Vulnerabilities(Major)

Vulnerabilities #1~#6 : Security – Method returns internal array
- 배열을 발생하는 곳에서 나타남, 복사본을 반환

```
- return displayModeData;  
+ /* [sonarqube][Vuln #1] */  
+ return displayModeData.clone();
```

Vulnerabilities #14, 15 : Handling files is security-sensitive
- 파일 경로를 string으로 하지 않는 방법으로 대체

```
1 부분 : 22-34 줄  
// Make Jar File  
audioInputStream = AudioSystem.getAudioInputStream(bufferedIn);  
}catch(Exception e){  
+ e.printStackTrace();  
+ /* [sonarqube][Vuln #14] */  
+ // IDE Test  
+ /*  
+ System.out.println(getClass().getResource("").getPath()+"sounds/Alarm"+index+".wav");  
+ audioInputStream = AudioSystem.getAudioInputStream(new File(getClass().getResource("").getPath()+"sounds/Alarm"+index+".wav"));  
+ */  
}  
clip = AudioSystem.getClip();  
clip.open(audioInputStream);
```

3. PMD Result – Vulnerabilities(Minor)

Vulnerabilities #1 : Use a logger to log this exception

- e.printStackTrace()를 Logger Class 를 사용하여 출력하게 함

```
+ e.printStackTrace();  
+ Logger logger = Logger.getLogger(Bell.class.getName());  
+ logger.log(Level.ALL, "[Bell] Exception", e);  
+ Thread.currentThread().interrupt();
```

Vulnerabilities #2

: Store copy of "newMode", Return a copy of "new Mode"

- .clone()을 통해 복사본을 반환

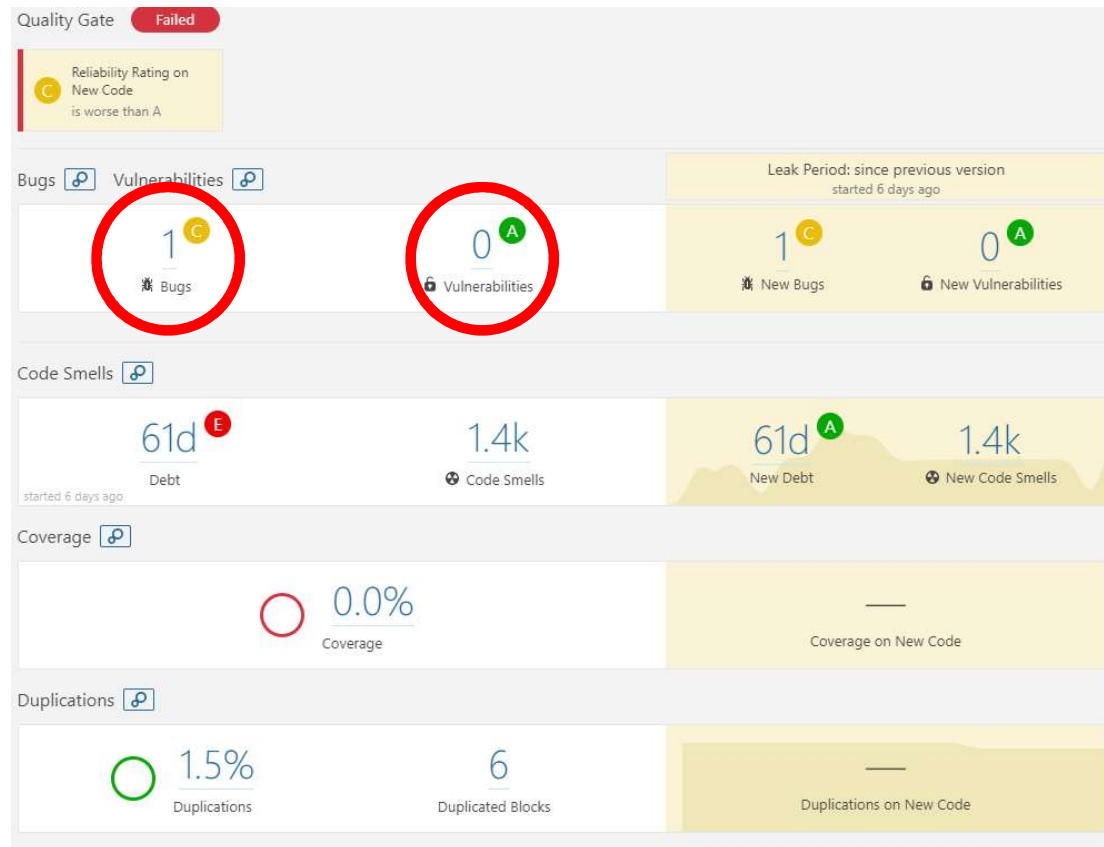
```
- public ArrayList<String> getNewMode() { return newMode; }  
- public void setNewMode(ArrayList<String> newMode) { this.newMode = newMode; }  
- public void setOldMode(ArrayList oldMode) { this.prevModeObject = oldMode; }  
+ /* - public ArrayList getPrevModeObject(){ return this.prevModeObject; }  
+ public ArrayList<String> getNewMode() { return (ArrayList)newMode.clone(); }  
+  
+ /* [sonarqube][Return a copy of] */  
+ public void setNewMode(ArrayList<String> newMode) { this.newMode = (ArrayList)newMode.clone(); }  
+  
+ /* [sonarqube][Store a copy of] */  
+ //public void setOldMode(ArrayList oldMode) { this.prevModeObject = (ArrayList)oldMode.clone(); }  
+  
+ /* [sonarqube][Store a copy of] */  
+ public ArrayList getPrevModeObject(){ return (ArrayList)this.prevModeObject.clone(); }
```

Vulnerabilities #3 : public -> private

- public을 private로 변경하여 해결

```
- public JTextField[] showDate = new JTextField[]{new JTextField(), new JTextField(), new JTextField()};  
+ private JTextField[] showDate = new JTextField[]{new JTextField(), new JTextField(), new JTextField()};
```

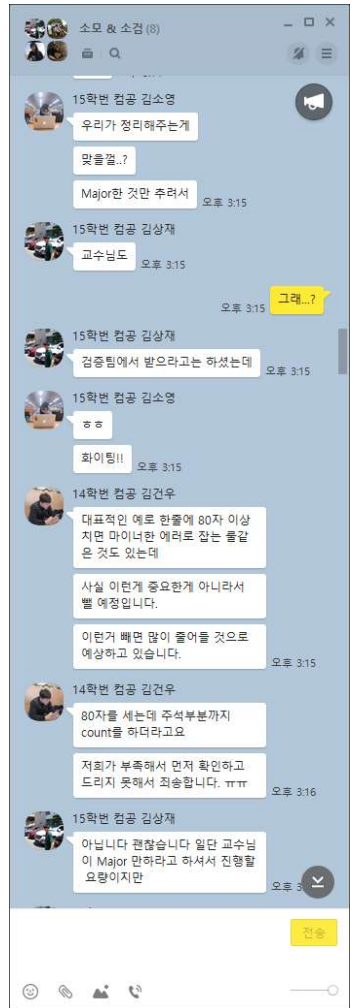
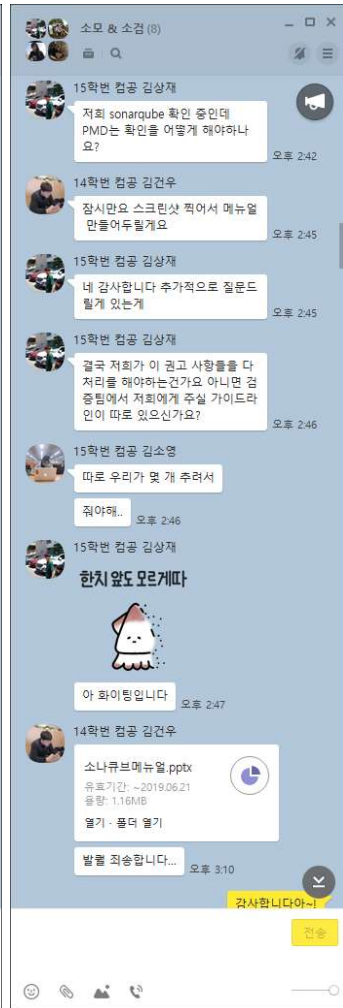
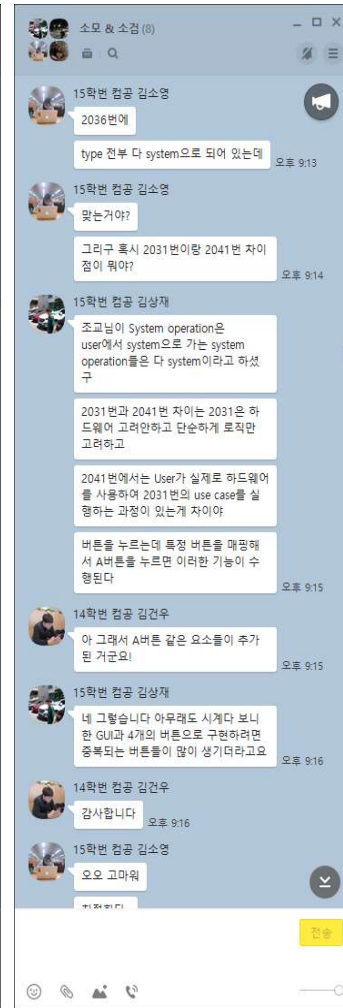
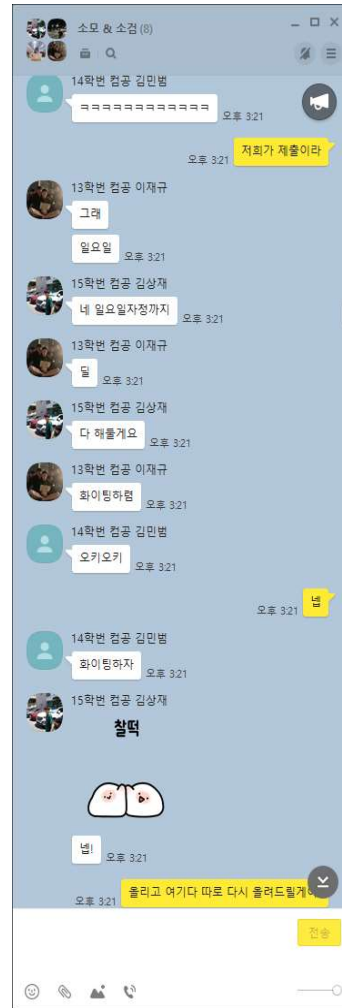
4. After Modification



5. Closing

Cooperation

상호 이해관계 완화,
 추가적인 보고서 자체 제작, 공유
 (SonarQube Manual, Bug report, etc...)



5. Closing

- A line of thought

A : 큰 소프트웨어를 만들기 위해서 공부해야 할 것들이 아직은 많다는 것을 느꼈습니다

B : 지금까지 아무런 설계과정 없이 바로 개발을 하였지만 이번 수업을 통해 설계의 중요성을 깨닫게 되었습니다

C : 힘들었습니다.

D : 스스로의 부족함을 깨닫는 좋은 계기가 되었습니다.

감사합니다

