

# OSP Stage 2040

< Design >

## 그놈! Clone Checker

Project Team

T4

Date

2016-04-12

---

Team Information

201411258 강태준

201411265 김서우

201411321 홍유리

# Contents

**Activity 2041. Define Real Use Cases**

**Activity 2042. Define Reports UI, and Storyboards**

**Activity 2043. Refine System Architecture**

**Activity 2044. Define Interaction Diagrams**

**Activity 2045. Define Design Class Diagrams**

**Activity 2046. Design Traceability Analysis**

## Activity 2041. Define Real Use Cases

Use Case	Display Main
Actor	User
Purpose	프로그램을 시작한다.
Overview	프로그램의 첫 메인 화면을 보여준다.
Type	Primary and Essential
Cross Reference	System Function : R 1.1
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (A) 사용자가 프로그램을 실행한다. 2. (S) 프로그램의 메인 UI를 띄운다.(그림#1)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Input Path
Actor	User
Purpose	유사도 검사를 진행할 대상의 위치 파악.
Overview	유사도 검사를 할 폴더의 경로를 입력한다.
Type	Primary and Essential
Cross Reference	Use Case : Setting Files System Function : R 1.2, R 1.2.1
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (A) 경로를 입력한 뒤(그림#1-A), 입력 버튼(그림#1-B) 클릭. 2. (S) input_path(path) 실행 3. (S) 해당 폴더의 모든 소스파일의 주석을 제거한다. 4. (S) 해당 폴더의 모든 소스파일의 대문자를 소문자로 통합한다. 5. 완료 되면 화면에 알림을 띄운다.(그림#3-A)
Alternative Courses of Events	2. (S) 해당 경로에 오류가 있다면 알림을 띄운다.(그림#2-A)
Exceptional Courses of Events	N/A

Use Case	Setting Files
Actor	System
Purpose	소스코드 파일 정리
Overview	소스코드의 주석을 제거하고, 소문자로 통일한다.
Type	Primary and Essential
Cross Reference	System Function : R 1.2.1
Pre-Requisites	N/A
Typical Courses of Events	<p>(A) : Actor, (S):System</p> <ol style="list-style-type: none"> <li>(S) 파일 개수만큼 setting_files(FILE f1) 을 f1자리의 파일을 바꿔가며 실행</li> <li>(S) setting_files(FILE f1)에서 f1.open() 실행</li> <li>(S) while(buf!=null){ strBuf[i].append(reader.readLine()); strBuf 원소 1개에 소스코드파일 1개의 내용 전체 저장</li> <li>(S) "/" 뒷부분을 다 지움(함수의 매개변수로 "/"가 들어가는 경우를 걸러내기위해 //뒤에 ")가 나오나 검사 후 삭제 "/"가 나온다면 "*"가 나오는부분까지 찾아서 다 지움  -&gt; strBuf 각 원소안에 존재하는 모든 주석 제거</li> <li>(S) while(strBuf[i][j] !=null) { if( 'A' &lt;= strBuf[i][j] &amp;&amp; strBuf[i][j] &lt;= 'Z') strBuf[i][j] +=32} 전체코드에서 모든('A'~'Z')찾아서 +32 -&gt;모든 대문자 소문자로 변경</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Start Analyze
Actor	User
Purpose	유사도 검사 시작
Overview	유사도 검사를 시작한다.
Type	Primary and Essential
Cross Reference	Use Case : Analyze Change Name, Analyze Loop, Analyze Conditional, Analyze Function, Calculate Similarity System Function : R 2.1.0, R 2.2.1, R 2.2.2, R 2.2.3, R 3.1
Pre-Requisites	N/A
Typical Courses of Events	<p>(A) : Actor, (S):System</p> <ol style="list-style-type: none"> <li>(A) Start 버튼(그림 #1 - C) 클릭</li> </ol>

	2. 분석 진행 중이라는 알림(그림 #4 - A)을 띄운다. 3. (S) start() ->start_flow() ->counting() 4. (S) 변수,함수,조건문,반복문의 개수를 세어줌. 5. (S) AnalysisSystem as = new AnalysisSystem(); AnalysisSystem()에서 Sourcecode[] 저장 6. (S) as.analyze_codes() 7. (S) SetupSystem에서 받아온 strBuf를 분석. 8. (S) SourceCode객체, 그 안의 Variable , Function , Loop , Conditional 객체생성
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Change Name
Actor	N/A
Purpose	유사도 분석
Overview	변수명, 함수명을 변경하였는지 검사한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirement : R 2.2.0
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (S) change_name() For(j=0;j<SourceCode_num;j++) If (SourceCode[i].Variable[j].numOfuse == SourceCode[i+1].Variable[j].numOfuse) Check[0]++; 두 소스코드상에서 각각 변수와 함수의 배열을 만들 고, 각 변수,함수의 빈도수를 두 소스코드 사이에서 비교한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Loop
Actor	N/A
Purpose	유사도 분석
Overview	반복문의 유사도를 검사한다.
Type	Primary and Essential and Hidden

Cross Reference	Functional Requirement : R 2.2.1
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (S) analyze_loop() 반복문의 종류변경(for->while,while->for)을 검사한다 {}안의 내용 확인 {}안의 내용은 변수,함수의 이름 변경 말고는 알고리즘 상의 변화이므로 이는 유사도에 반영이 되지 않는다. 따라서, 아스키코드 33번값('!')부터 126번(~) 사이의 값의 개수(알파벳대소문자,숫자,기본적인 특수문자포함)가 각각 일치시 Check[1]++; -> 반복문 안의 body의 내용을 비교
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Conditional
Actor	N/A
Purpose	유사도 분석
Overview	조건문의 유사도를 검사한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirement : R 2.2.2
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) analyze_cond() 반복문과 마찬가지로 조건문(if->switch, switch->if)를 검사한다. 조건문의 body(내용) 부분은 switch나 if나 같기 때문에,body 부분을 아스키코드 33번값('!')부터 126번(~) 사이의 값의 개수(알파벳대소문자,숫자,기본적인 특수문자포함)가 각각 일치시 ->Check[2]++;
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Function
----------	------------------

Actor	N/A
Purpose	유사도 분석
Overview	함수의 분기를 검사한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirement : R 2.2.3
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) analyze_func() 1개의 함수가 2개로 분해되었거나, 2개의 함수를 1개로 합쳐 놓은 것을 검사 소스코드 1개의 모든 함수의 {}안의 내용을 합쳐놓고, 5번과 비슷한 방법으로, 특수문자, 숫자, 알파벳의 개수가 일치시 Check[3]++;
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Calculate Similarity
Actor	System
Purpose	유사도의 계산 및 그놈 파일 찾기
Overview	코드 분석을 바탕으로 유사도를 계산하여 저장하고 이를 바탕으로 중심이 되는 그놈 파일을 찾는다.
Type	Primary and Essential
Cross Reference	Use Case : Analyze Code System Function : R 3.1
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (S) cal_similarity(Check) Check 배열 각각의 원소에 가중치를 적용하여 유사도를 계산한다. 2. (S) write_result() 각 소스코드끼리 어떤 부분이 유사하고 어떤 식으로 유사도를 도출해냈는지 총 분석 결과를 text파일로 저장.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Show X File
----------	-------------

Actor	User
Purpose	그 놈! 파일 확인
Overview	그 놈! 파일의 파일명을 본다.
Type	Primary and Essential
Cross Reference	Use Case : Calculate Similarity System Function : R 3.2
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (A) Show 그 놈_File 버튼 클릭(그림#1-D) 2. (S) showMessage(2); ->Actor에게 X File name 출력(그림#6-A)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

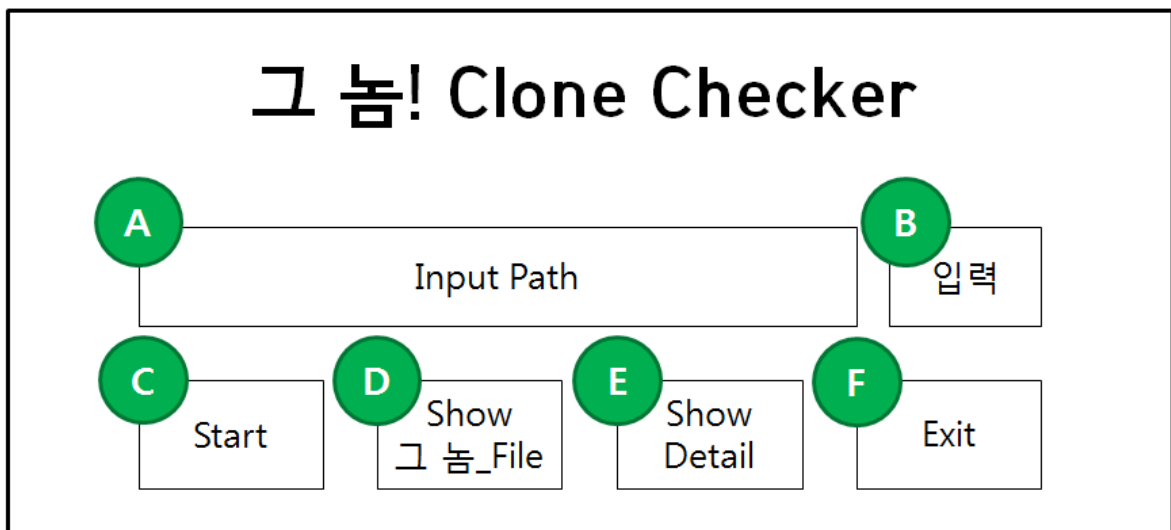
Use Case	Show Detail
Actor	User
Purpose	각각의 유사도 검사 결과 확인
Overview	각각의 유사도 검사 결과를 확인한다.
Type	Primary and Essential
Cross Reference	Use Case : Calculate Similarity System Function : R 3.3
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (A) Show Detail 버튼(그림#1-E) 클릭 2. (S) 유사도 검사의 기록이 담긴 txt 파일을 보여준다.(그림 #7-A)
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Exit
Actor	User
Purpose	프로그램 종료
Overview	프로그램을 종료한다.
Type	Primary and Essential
Cross Reference	System Function : R 3.3
Pre-Requisites	N/A

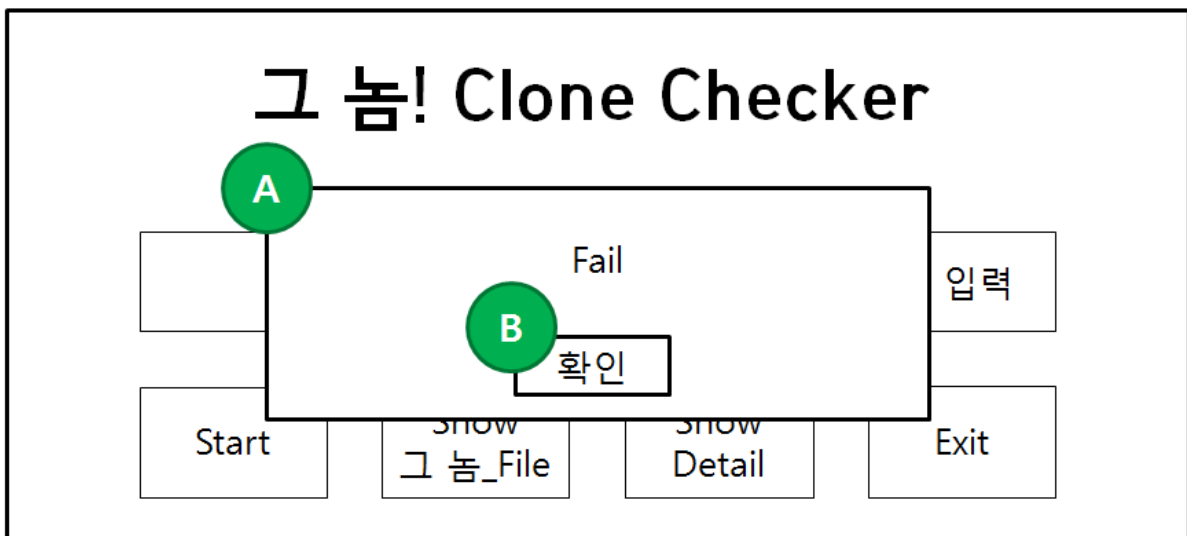


Typical Courses of Events	(A) : Actor, (S):System 1. (A) Exit 버튼(그림#1-F) 클릭 2. (S) 프로그램을 종료할 것인지 다시 물어본다. 3. (A) 확인(그림#7-B) 버튼을 누르면 종료한다.
Alternative Courses of Events	3. (A) 취소(그림#7-C) 버튼을 누르면 메인으로 돌아간다.
Exceptional Courses of Events	N/A

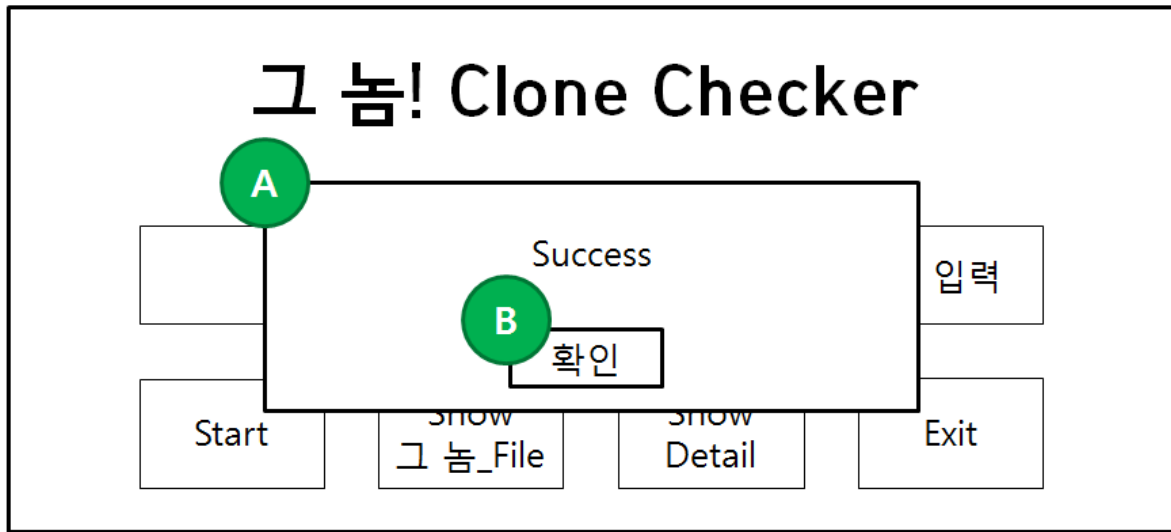
### Activity 2042. Define Reports UI, and Storyboards



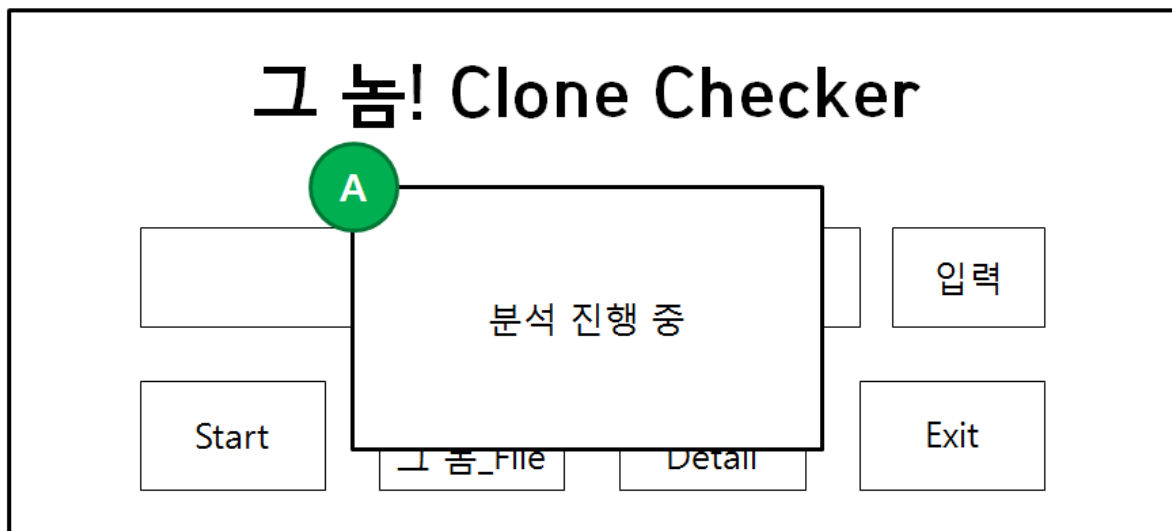
[그림 #1]



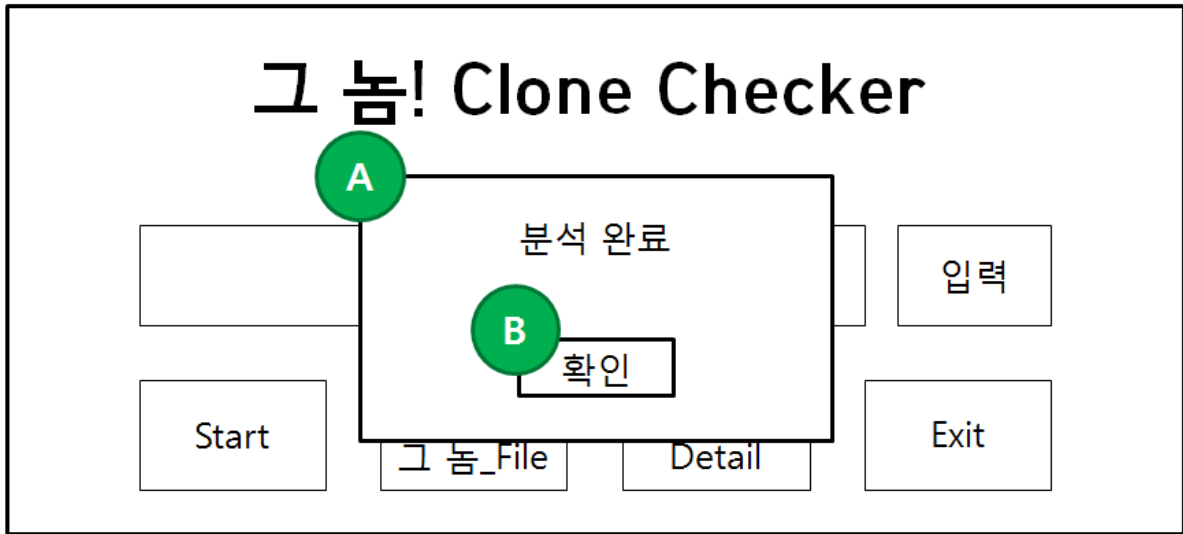
[그림 #2]



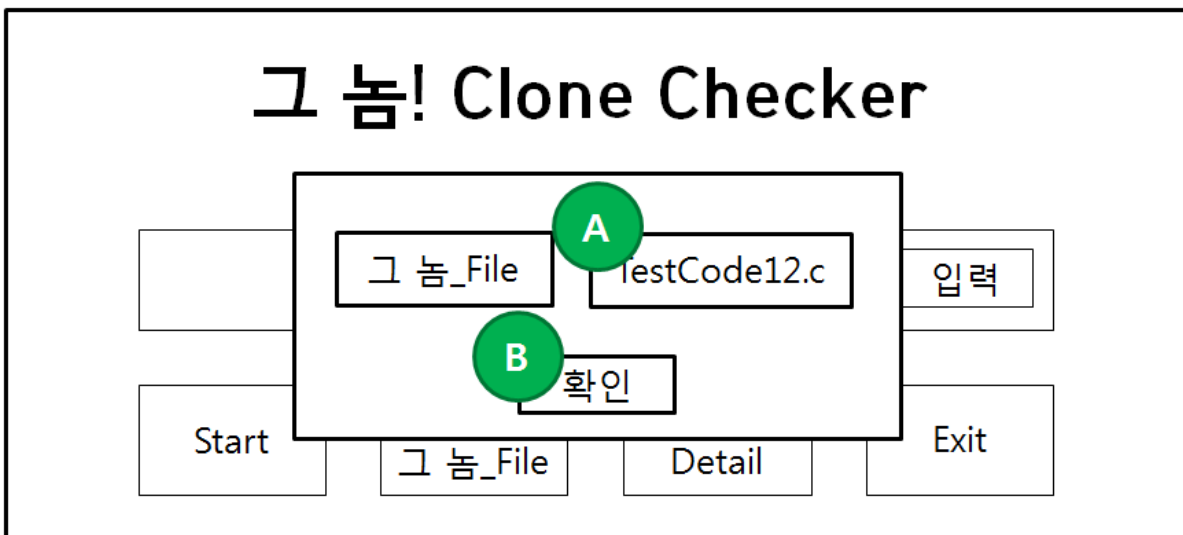
[그림 #3]



[그림 #4]



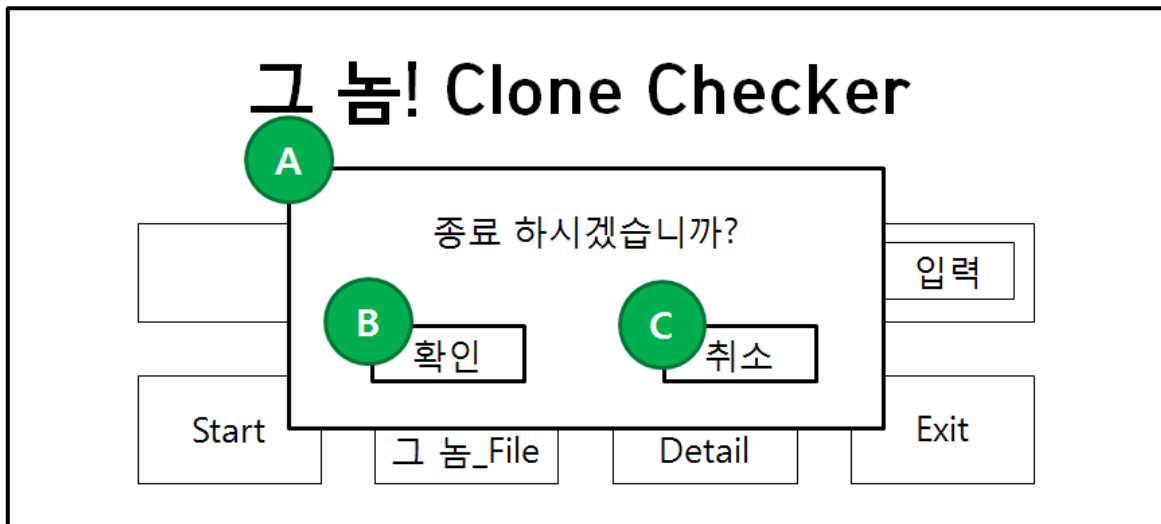
[그림 #5]



[그림 #6]

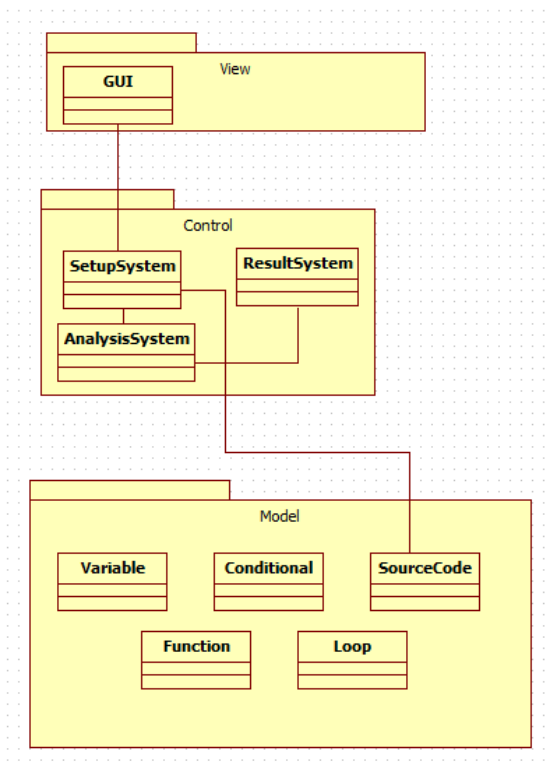


[그림 #7]



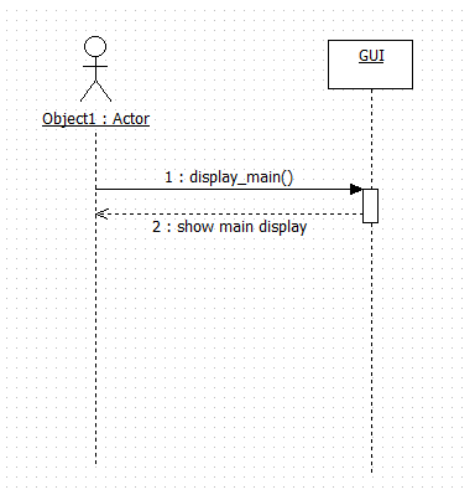
[그림 #8]

## Activity 2043. Refine System Architecture

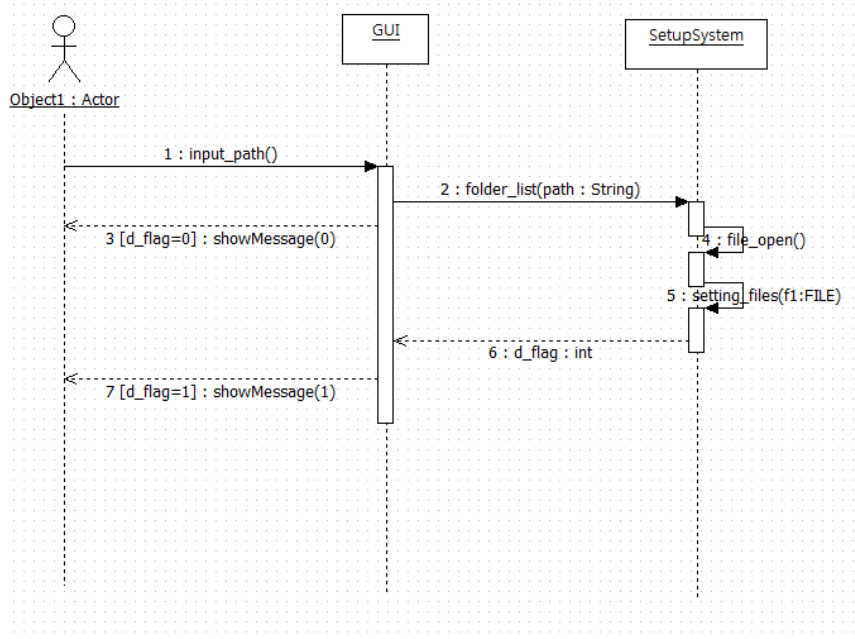


## Activity 2044. Define Interaction Diagrams

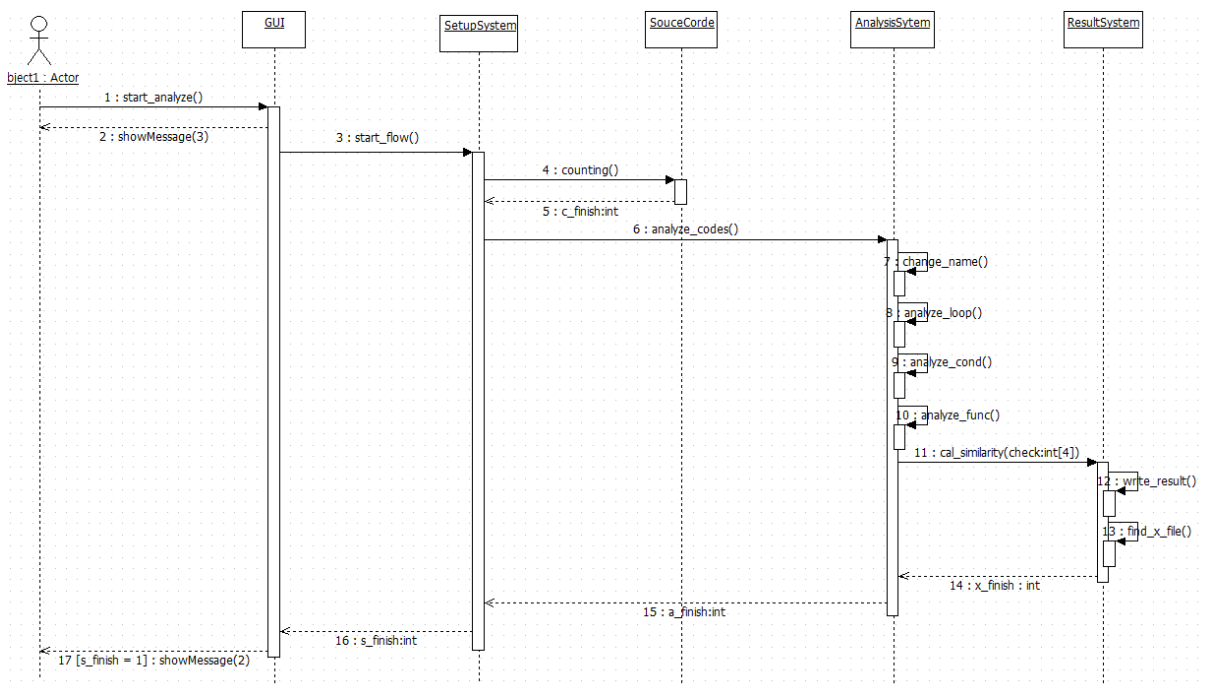
[ Display Main ]



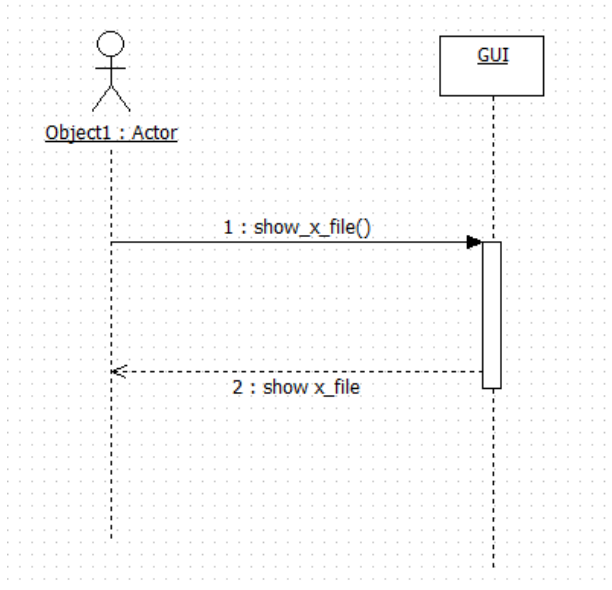
[ Input Path ]



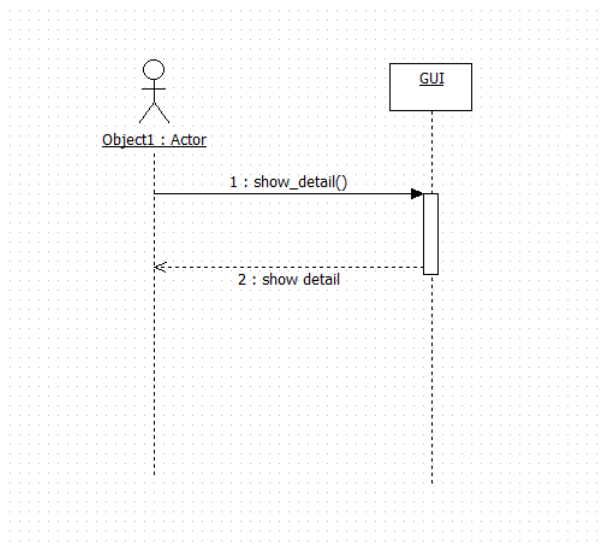
[ Start Analyze ]



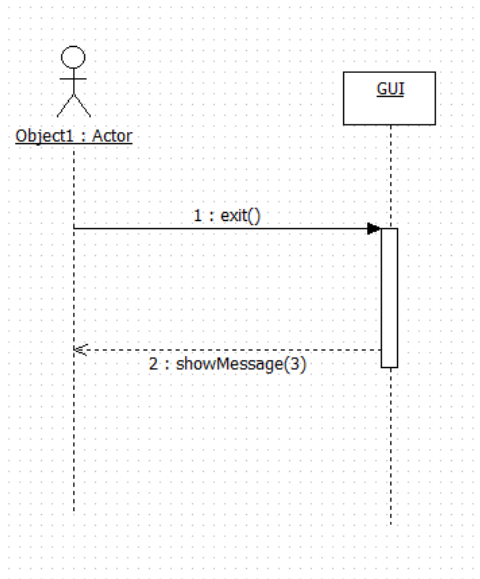
[ Show X File ]



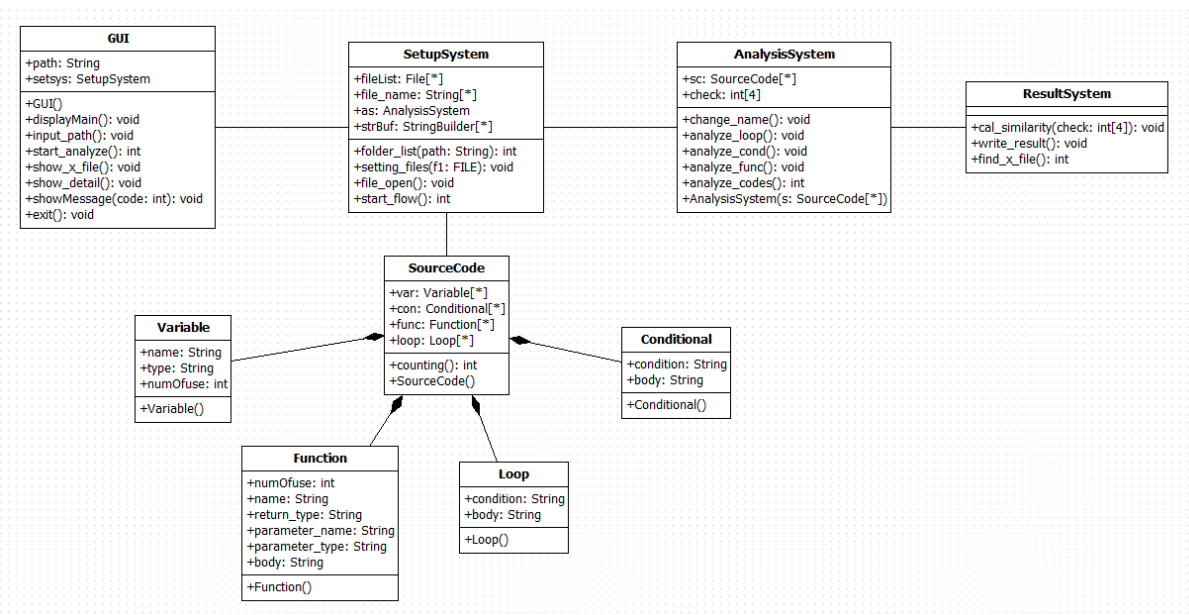
[ Show Detail ]



[ Exit ]



### Activity 2045. Define Design Class Diagrams





## Activity 2046. Design Traceability Analysis

Operation in sequence diagram	Operation in interaction diagram	Method	Class
display_main	display_main()	displayMain():void	GUI
input_path	input_path()	input_path():void	
start_analyze	folder_list(path)	start_analyze():void	
show_x_file	file_open()	show_x_file():void	
show_detail	setting_files(f1)	show_detail():void	
exit	showMessage()	showMessage(code:int):void	
	start_analyze()	exit():void	SetupSystem
	start_flow()	folder_list(path:String):int	
	counting()	setting_files(f1:FILE):void	
	analyze_codes()	file_open():void	AnalsisSystem
	change_name()	start_flow():int	
	analyze_loop()	change_name():void	
	analyze_cond()	analyze_loop():void	
	analyze_func()	analyze_func():void	ResultSystem
	cal_similarity(check)	analyze_codes():int	
	write_result()	cal_similarity(check:int(4)):void	
	find_x_file()	write_result():void	SourceCode
	show_x_file()	find_x_file():int	
	show_detail()	counting():int	
	exit()		