

# OSP Stage 2040

< Design >

## 그놈! Clone Checker

Project Team

T4

Date

2016-06-01

---

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	Actor
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) Actor가 프로그램을 실행한다. 2. (S) 초기화면(그림#1)을 보여준다. 이때 경로 입력(그림#1 - A), Input버튼(그림#1 - B), Exit(그림#1 - F)는 활성화 되어 있고, Start(그림#1 - C), Show X_File(그림#1 - D), Show Detail(그림#1 - E)는 비활성화 되어있는 상태로 프로그램이 실행된다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Input Path
Actor	Actor
Purpose	유사도 검사를 진행할 대상의 위치 파악.
Overview	유사도 검사를 할 폴더의 경로를 입력한다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 1.2, R 1.2.1 UseCase : Setting Files
Pre-Requisites	Start(그림#1 - C), Show X_File(그림#1 - D), Show Detail(그림#1 - E) 버튼이 비활성화 상태여야 한다.
Typical Courses of Events	(A) : Actor, (S):System 1. (A) Actor가 유사도 검사를 진행할 소스 코드들이 있는 폴더의 경로를(그림#1 - A)에 입력한 뒤, Input 버튼(그림#1 - B)을 누른다. 2. (S) 입력한 경로를 바탕으로 유사도 검사 전에 SourceCode 클래스의 객체 배열을 생성하여 소스 코드의 정보를 저장한다. 3. (S) 정보 저장까지 성공하였다면, 경로 탐색 성공 알림창(그림#4 - A)을 띄우고 확인버튼(그림#4 - B)을 누르면, Start 버튼(그림#1 - C)을 활성화시킨다.

Alternative Courses of Events	<p>2. (S) 해당 경로가 잘못되었거나 파일이 없거나, 파일이 1 개라면 알림창(그림#2 - A)을 띄우고, 확인버튼(그림#2 - B)을 누르면, 다시 초기화면(그림#1)로 돌아간다.</p> <p>2. (S) 해당 경로에 .c 파일 외의 다른 파일이 들어있다면 알림창(그림#3 - A)을 띄우고, 확인버튼(그림#3 - B)을 누르면, 다시 초기화면(그림#1)로 돌아간다.</p>
Exceptional Courses of Events	N/A

Use Case	Setting Files
Actor	N/A
Purpose	소스 코드의 정보 저장
Overview	소스 코드의 변수, 함수, 반복문, 조건문에 관련된 내용 SourceCode 클래스의 객체 배열에 저장한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirements : R 1.2.1 UseCase : Input Path
Pre-Requisites	.c 파일만 들어 있는 폴더의 경로를 입력하고, Input 버튼을 누른다
Typical Courses of Events	<p>(A) : Actor, (S):System</p> <ol style="list-style-type: none"> <li>(S) 경로의 파일 리스트를 받아온다.</li> <li>(S) 파일 리스트에서 파일의 이름을 따로 분리하여 저장한다.</li> <li>(S) 파일을 열어 소스 코드의 전체 내용을 하나의 String변수에 저장한다.</li> <li>(S) String 변수에서 소스 코드의 주석에 해당하는 내용을 모두 삭제한다.</li> <li>(S) String 변수의 모든 문자를 소문자로 통일한다.</li> <li>(S) String 변수에서 정규식을 통하여 소스 코드의 구조체 관련 내용을 찾아 파싱, 함수 파싱, 전역 변수 파싱, 반복문 파싱, 조건문 파싱, 함수 지역변수 파싱을 진행하여 SourceCode 내의 변수에 각각 저장한다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Start Analyze
Actor	Actor
Purpose	유사도 검사 시작
Overview	유사도 검사를 시작한다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 2.1, R 2.1.1, R 2.1.2, R 2.1.3, R 2.1.4, R 2.1.5, R 2.1.6 UseCase : Analyze Variable, Analyze Loop, Analyze Conditional, Analyze Function, Make Detail, Find X_File
Pre-Requisites	.c 파일만 들어 있는 폴더의 경로를 입력하여 Input 버튼(그림#1 - B)을 누른 뒤 경로 탐색 성공 알림창(그림#4 - A)이 뜨고, Start 버튼(그림#1 - C)을 누른다.
Typical Courses of Events	(A) : Actor, (S):System 1. (A) Actor가 Start버튼(그림#1 - C)을 누르면, 알림창(그림#5 - A)이 뜬다. 2. (S) SourceCode 객체 배열을 바탕으로 모든 SourceCode 객체 자기 자신과의 변수 유사도 검사, 함수 유사도 검사, 반복문 유사도 검사, 조건문 유사도 검사를 진행하여 double[] self 에 저장한다. 3. (S) 기존에 저장해놓은 SourceCode 객체 배열을 바탕으로 2개의 SourceCode 객체의 변수 유사도 검사를 진행하여 결과를 double[0][] score에 저장한다. 4. (S) 3 에서 검사를 진행한 두 SourceCode 객체 사이의 함수 유사도 검사를 진행하여 결과를 double[1][] score에 저장한다. 5. (S) 3 에서 검사를 진행한 두 SourceCode 객체 사이의 반복문 유사도 검사를 진행하여 결과를 double[2][] score에 저장한다. 6. (S) 3 에서 검사를 진행한 두 SourceCode 객체 사이의 조건문 유사도 검사를 진행하여 결과를 double[3][] score에 저장한다. 7. (S) 2. 3. 4. 5. 6.의 결과를 바탕으로 두 SourceCode 객체 사이의 최종 유사도를 계산하여 double[][] final_result에 저장한다. 8. (S) 2. 3. 4. 5. 6. 7. 의 결과를 바탕으로 두 SourceCode 객체 사이의 유사도 검사의 세부 내용을 .txt 파일에 저장한다. 9. (S) 마지막 두 SourceCode 객체 사이의 3. 4. 5. 6. 7. 8.이 끝난 후, 저장되어 있던 최종 유사도를 바탕으로 X_File을 찾는다. 10. (S) 위의 과정이 끝나면 분석 성공 알림창(그림#6 - A)을 띄

	우고, 확인버튼(그림#6 - B을 누르면, Start 버튼(그림#1 - C)은 비활성화 시키고, Show X_File 버튼(그림#1 - D), Show Detail 버튼(그림#1 - E)을 활성화 시킨다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Variable
Actor	N/A
Purpose	변수의 유사도 분석
Overview	변수에 관한 유사도를 검사한다..
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirements : R 2.1.1 UseCase : Start Analyze Code
Pre-Requisites	Start Analyze Code 가 완료되어야 한다
Typical Courses of Events	(A) : Actor, (S):System 1. (S) 두 SourceCode 객체 안의 변수의 총 개수를 세고, 그 값들의 차이에 따른 유사도 점수를 부여하고, 각 소스 코드의 변수의 총 개수와 점수를 double[] score에 저장한다. 2. (S) 두 SourceCode 객체를 토대로 동일한 자료형 내에서 같은 이름을 갖는 변수의 개수를 모두 세고, 전체 변수 개수에 대한 비율을 계산한다. 그 비율로 점수를 부여하고 두 점수의 평균값을 최종적인 점수로 계산한다. 그 후 동일한 이름을 갖는 변수의 개수와, 최종 점수를 double[] score에 저장한다. 3. (S) 1. 2. 의 결과 점수에 가중치를 적용하여 최종 변수 유사도 점수를 계산하여 double[] score에 저장하여 double[] score를 return한다.
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 Requirements : R 2.1.2 UseCase : Start Analyze Code

Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (S) 두 SourceCode 객체 안의 for문의 총 개수 차이에 따른 유사도 점수를 부여하고, 각 소스 코드의 for문의 총 개수와 점수를 double[] score에 저장한다. 2. (S) 두 SourceCode 객체 안의 while문의 총 개수 차이에 따른 유사도 점수를 부여하고, 각 소스 코드의 while문의 총 개수와 점수를 double[] score에 저장한다. 3. (S) 1. 2 의 결과 점수의 평균 값을 최종 반복문 유사도 점수로 계산하여 double[] score에 저장하여 double[] score를 return 한다.
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 Requirements : R 2.1.4 UseCase : Start Analyze Code
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) 두 SourceCode 객체 안의 cond.if_number의 차이에 따른 유사도 점수를 부여하고, 각 SourceCode 객체의 cond.if_number와 점수를 double[] score에 저장한다. 2. (S) 두 SourceCode 객체 안의 cond.else_number의 차이에 따른 유사도 점수를 부여하고, 각 SourceCode 객체의 cond.else_number와 점수를 double[] score에 저장한다. 3. (S) 두 SourceCode 객체 안의 cond.else_if_number차이에 따른 유사도 점수를 부여하고, 각 SourceCode 객체의 cond.else_if_number와 점수를 double[] score에 저장한다. 4. (S) 두 SourceCode 객체 안의 cond.case_number 차이에 따른 유사도 점수를 부여하고, 각 SourceCode 객체의 cond.case_number와 점수를 double[] score에 저장한다. 5. (S) 1. 2. 3. 4. 의 결과 점수에 가중치를 적용하여 최종 조건문 유사도 점수를 계산하여 double[] score에 저장하여 double[] score를 return한다.

Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Analyze Function
Actor	N/A
Purpose	유사도 분석
Overview	함수에 대한 검사를 한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirements : R 2.1.3 UseCase : Start Analyze Code
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S) : System 1. (S) 두 SourceCode 객체 안의 num_of_func의 차이에 따른 유사도 점수를 부여하고, 각 SourceCode 객체의 num_of_func와 점수를 double[] score에 저장한다. 2. (S) 두 SourceCode 객체에서 이름이 동일한 함수의 개수를 세어 그 개수에 비례하여 점수를 부과하고, 개수와 점수를 double[] score에 저장한다 3. (S) 1. 2. 의 결과 점수의 평균 값을 최종 함수 유사도 점수로 계산하여 double[] score에 저장하여 double[] score를 return 한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Find X_File
Actor	N/A
Purpose	X_File 찾기
Overview	진행된 검사들의 최종 유사도 점수를 바탕으로 X_File 을 찾는다.

Type	Primary and Essential and Hidden
Cross Reference	Functional Requirements : R 2.1.5 UseCase : Start Analyze, Show X_File
Pre-Requisites	경로 안에 들어 있던 SourceCode 객체끼리 1:1 로 유사도 검사를 진행하여 최종 유사도 점수를 계산하고, 세부 내용을 .txt 파일에 작성한다.
Typical Courses of Events	(A) : Actor, (S):System 1. (S) 최종 유사도를 저장한 double[][] final_result의 각 행의 평균 값을 행을 인덱스로 하는 double[] avr에 저장한다. 2. (S) double[] avr의 원소 중 최댓값을 찾는다. 3. (S) 최댓값에 해당하는 double[] avr의 인덱스를 return_index에 저장하여 return한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Make Detail
Actor	N/A
Purpose	최종 유사도 점수 계산 및 유사도 검사 결과의 세부 내용 작성
Overview	가중치를 적용한 최종 유사도 점수를 계산하고, 변수에 대한 유사도 검사, 반복문에 대한 유사도 검사, 조건문에 대한 유사도 검사, 함수에 대한 유사도 검사의 세부 내용을 .txt 파일에 저장한다.
Type	Primary and Essential and Hidden
Cross Reference	Functional Requirements : R 2.1.6 UseCase : Start Analyze, Show Detail
Pre-Requisites	Analyze Variable, Analyze Loop, Analyze Conditional, Analyze Function 을 완료한다.
Typical Courses of Events	(A) : Actor, (S):System 1. (S) 두 SourceCode 객체를 바탕으로 진행한 변수, 반복문, 조건문, 함수 유사도 검사 결과로 가중치를 적용하여 최종 유사도를 계산한다. 2. (S) 두 SourceCode 객체의 변수 유사도 검사 세부 내용을 .txt 파일에 작성한다. 3. (S) 두 SourceCode 객체의 반복문 유사도 검사 세부 내용을 .txt 파일에 작성한다. 4. (S) 두 SourceCode 객체의 조건문 유사도 검사 세부 내용을 .txt 파일에 작성한다.

	<p>5. (S) 두 SourceCode 객체의 함수 유사도 검사 세부 내용을 .txt 파일에 작성한다.</p> <p>6. (S) 두 SourceCode 객체의 최종 유사도를 .txt 파일에 작성한다.</p>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

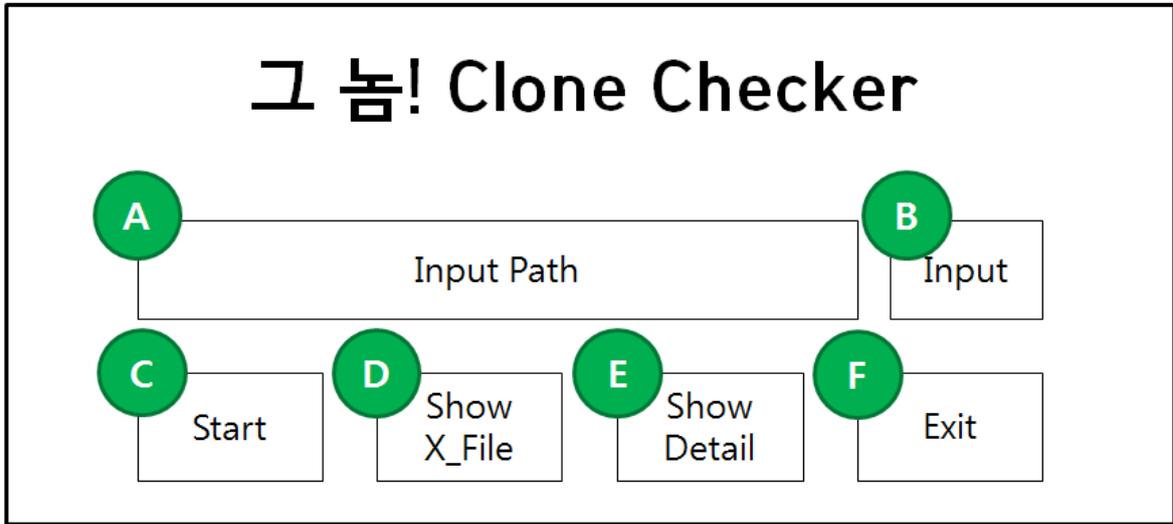
Use Case	Show X_File
Actor	Actor
Purpose	X_File 의 파일명 보여주기
Overview	X_File 의 파일명을 보여준다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 3.1 UseCase : Find X_File
Pre-Requisites	Start Analyze 가 완료되어 Show X_File 버튼(그림#1 - D)이 활성화 상태여야 한다.
Typical Courses of Events	<p>(A) : Actor, (S):System</p> <ol style="list-style-type: none"> <li>1. (A) Actor가 Show X_File 버튼(그림#1 - D)을 누른다.</li> <li>2. (S) X_File의 파일명(그림#7 - A)을 보여준다.</li> <li>3. (A) Actor가 확인 버튼(그림#7 - B)을 누른다.</li> <li>4. (S) Show X_File 버튼(그림#1 - D)을 비활성화 시킨다.</li> </ol>
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Show Detail
Actor	Actor
Purpose	검사의 세부 내용 파일(.txt) 의 내용 보여주기
Overview	검사의 세부 내용 파일(.txt) 의 내용을 보여준다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 3.2 UseCase : Make Detail
Pre-Requisites	Start Analyze 가 완료되어 Show Detail 버튼(그림#1 - E)이 활성화 상태여야 한다.
Typical Courses of Events	<p>(A) : Actor, (S):System</p> <ol style="list-style-type: none"> <li>1. (A) Actor가 Show Detail 버튼(그림#1 - E)을 누른다.</li> <li>2. (S) 유사도 검사의 세부 내용이 담긴 .txt 파일의 내용(그림#8 - A)을 보여준다.</li> </ol>

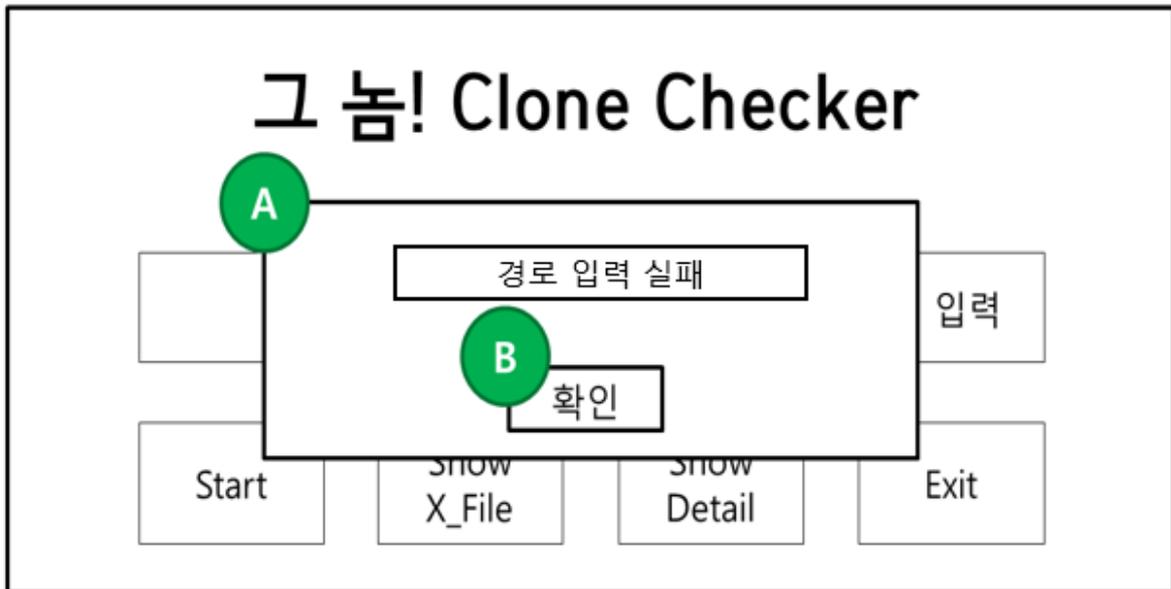
	3. (A) Actor가 열려 있던 .txt 파일(그림#8 - A)을 닫는다. 4. (S) Show Detail 버튼(그림#1 - E)을 비활성화 시킨다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Exit
Actor	Actor
Purpose	프로그램 종료
Overview	프로그램을 종료한다.
Type	Primary and Essential
Cross Reference	Functional Requirements : R 4.1
Pre-Requisites	N/A
Typical Courses of Events	(A) : Actor, (S):System 1. (A) Actor가 Exit 버튼(그림#1 - F)을 누른다. 2. (S) 종료 확인 창(그림#9 - A)을 보여준다. 3. (A) Actor가 확인 버튼(그림#9 - B)을 누른다. 4. (S) 프로그램을 종료한다.
Alternative Courses of Events	3. (A) Actor 가 취소 버튼(그림#9 - C)을 누른다. 4. (S) Exit 버튼을 누르기 전의 화면(그림#1)으로 돌아간다.
Exceptional Courses of Events	N/A

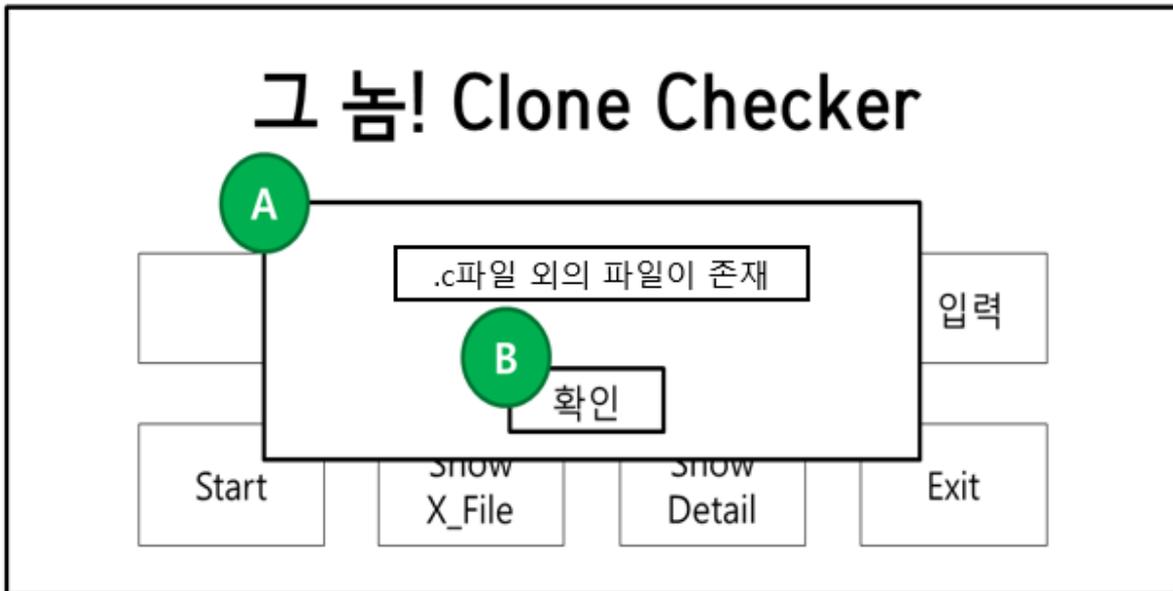
Activity 2042. Define Reports UI, and Storyboards



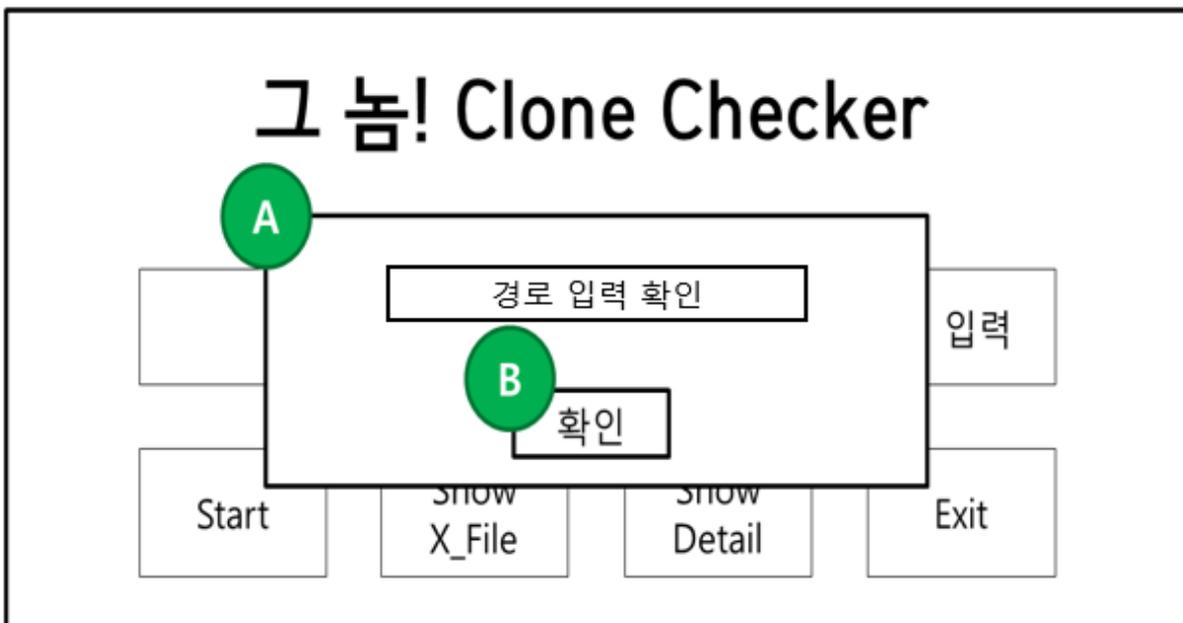
[그림 #1]



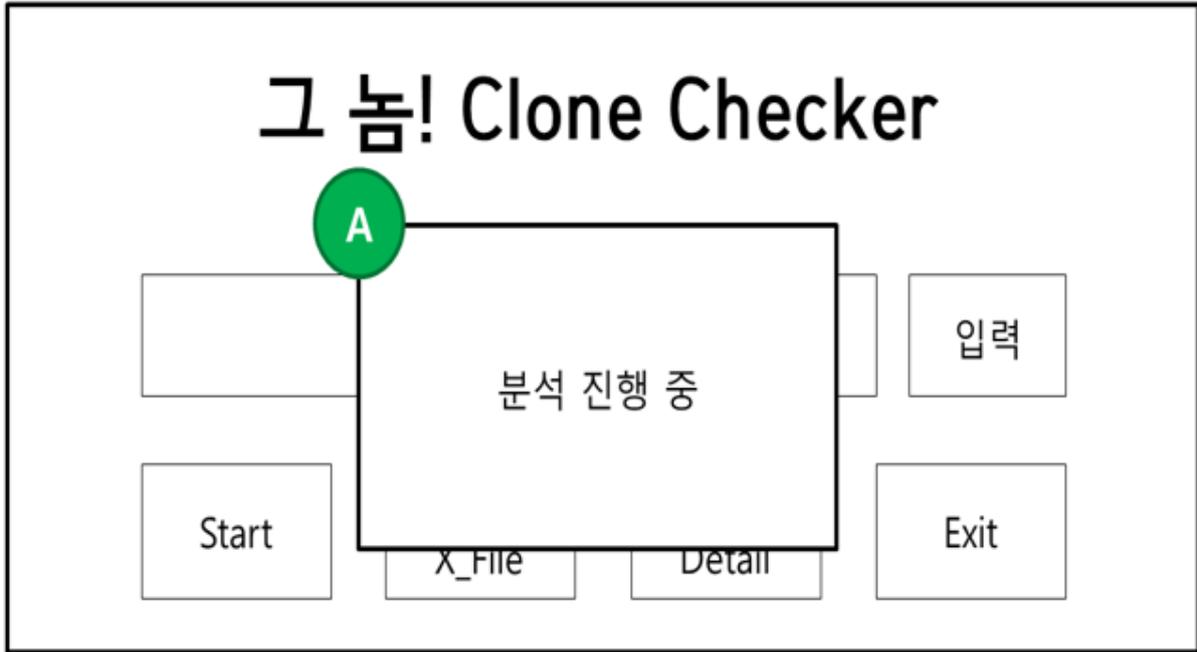
[그림 #2]



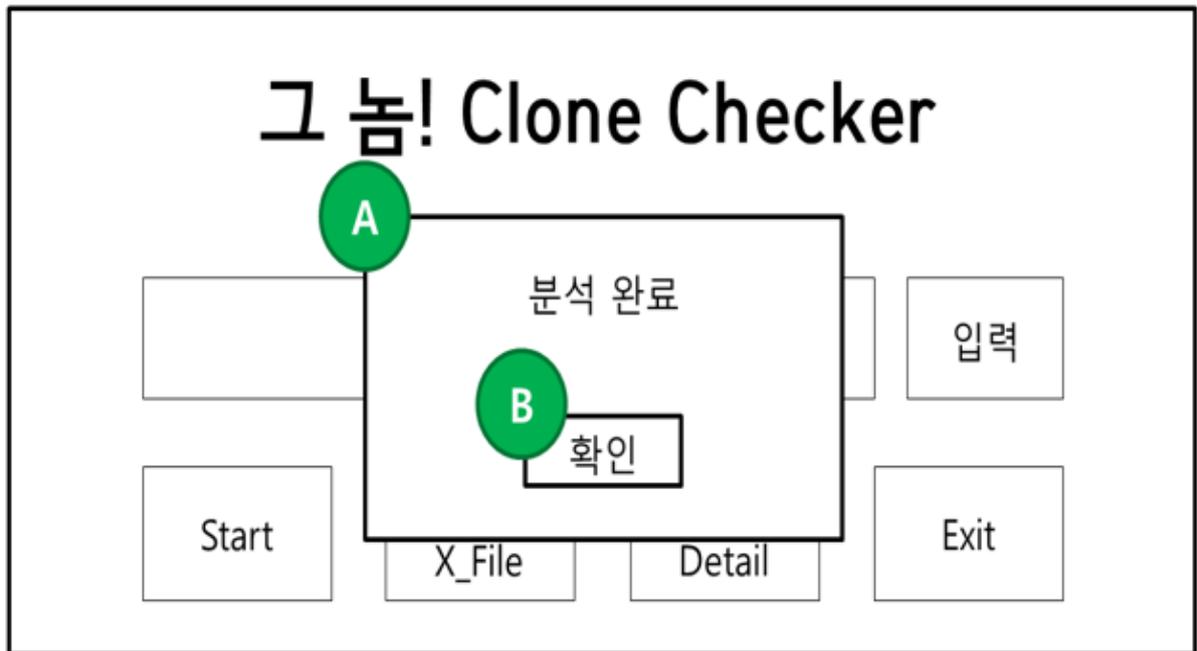
[그림 #3]



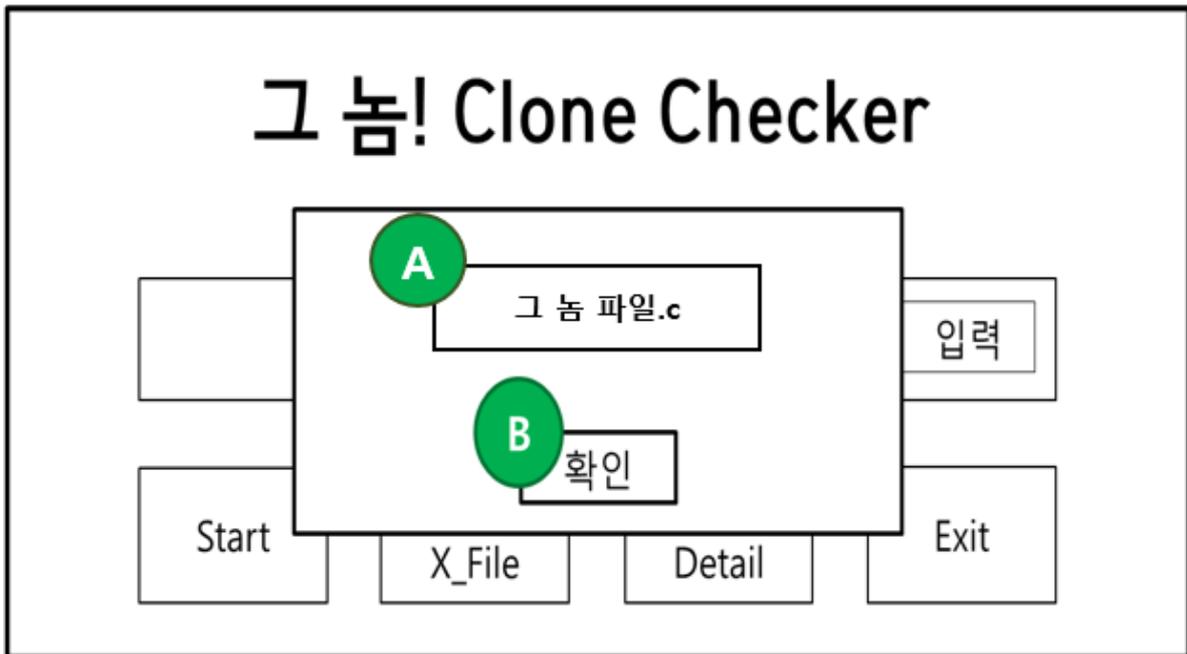
[그림 #4]



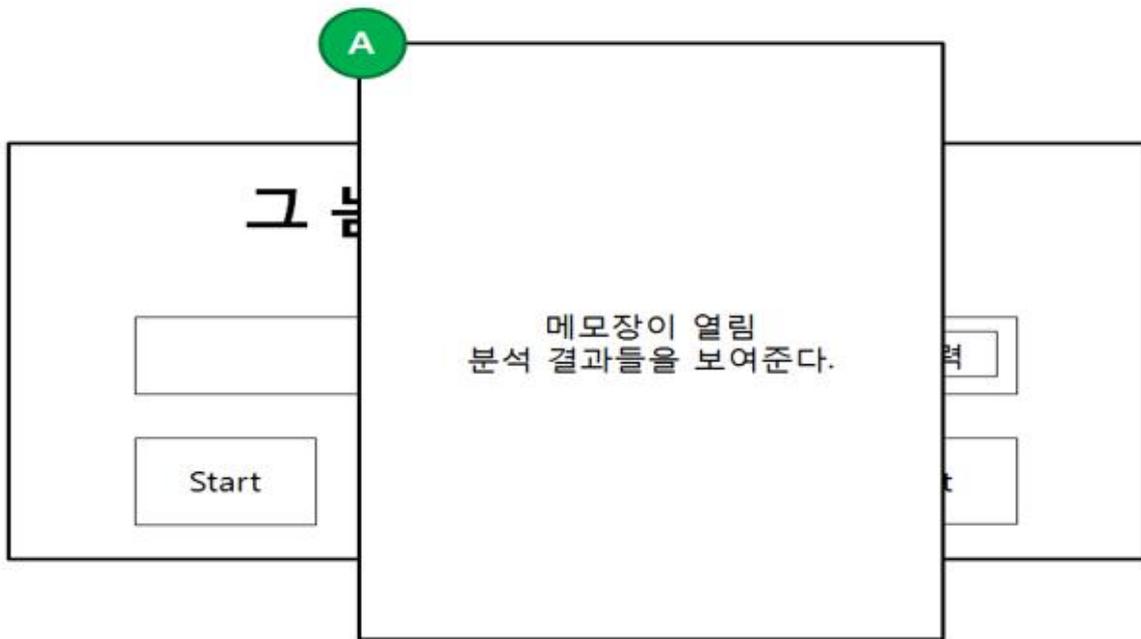
[그림 #5]



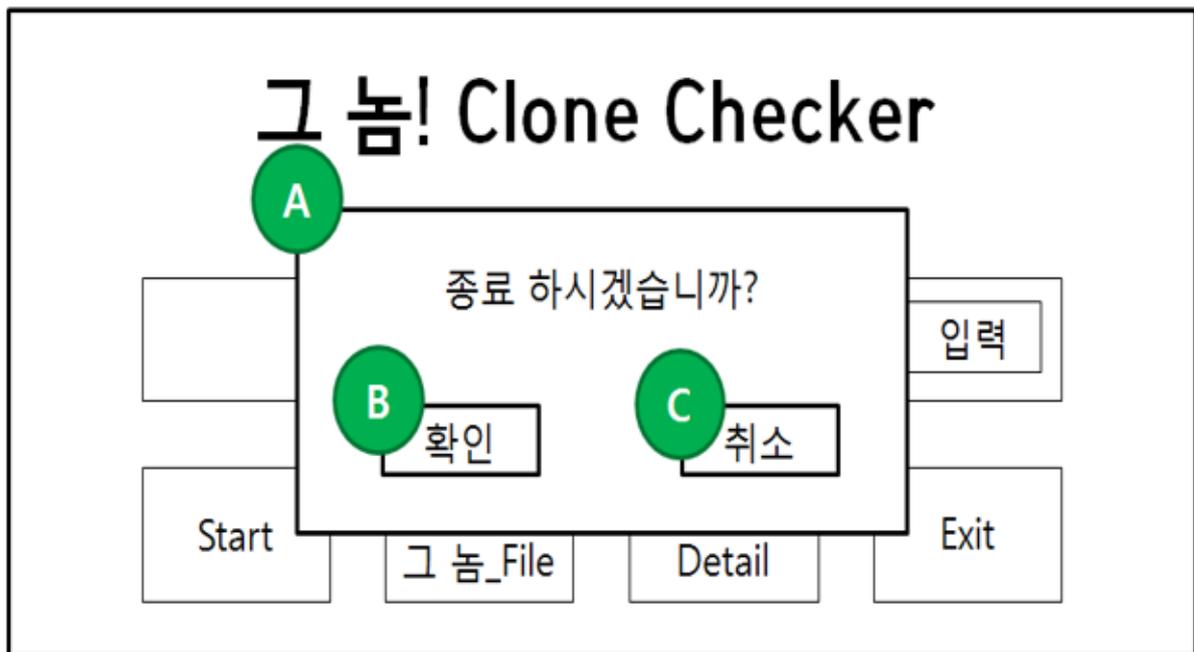
[그림 #6]



[그림 #7]



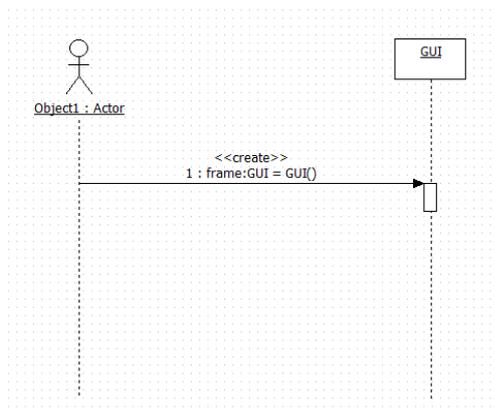
[그림 #8]



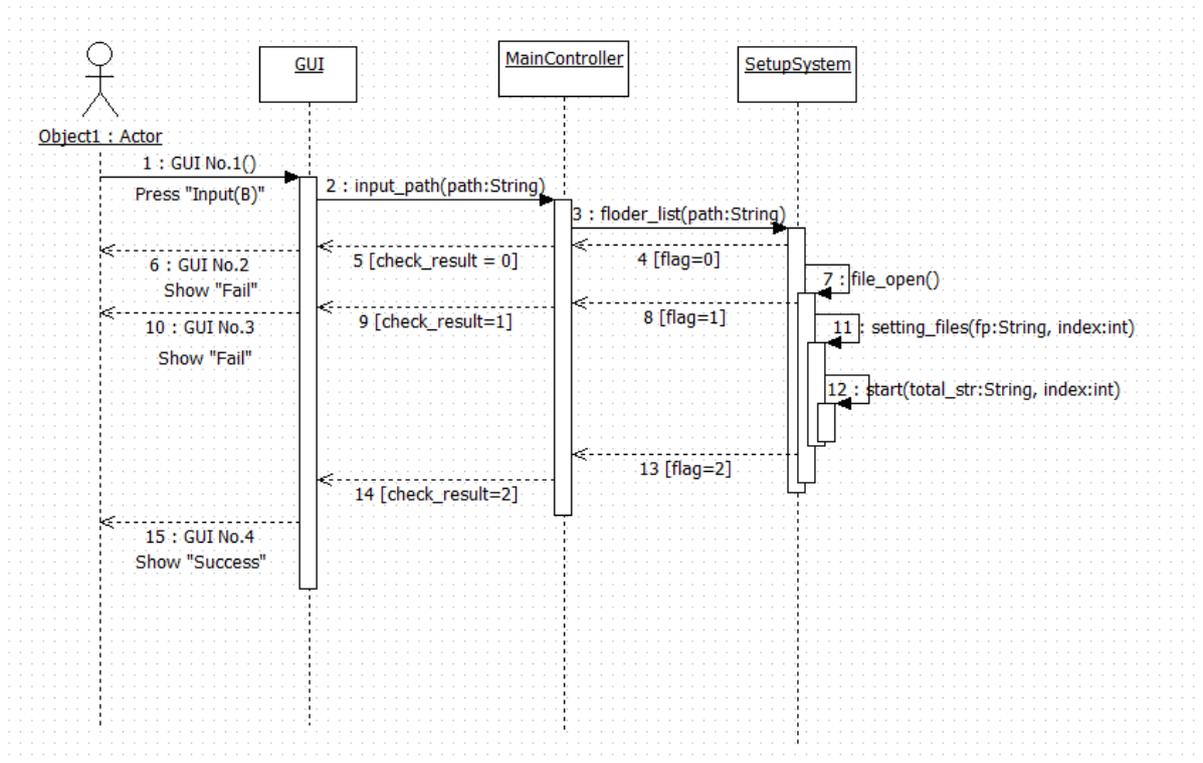
[그림 #9]

## Activity 2044. Define Interaction Diagrams

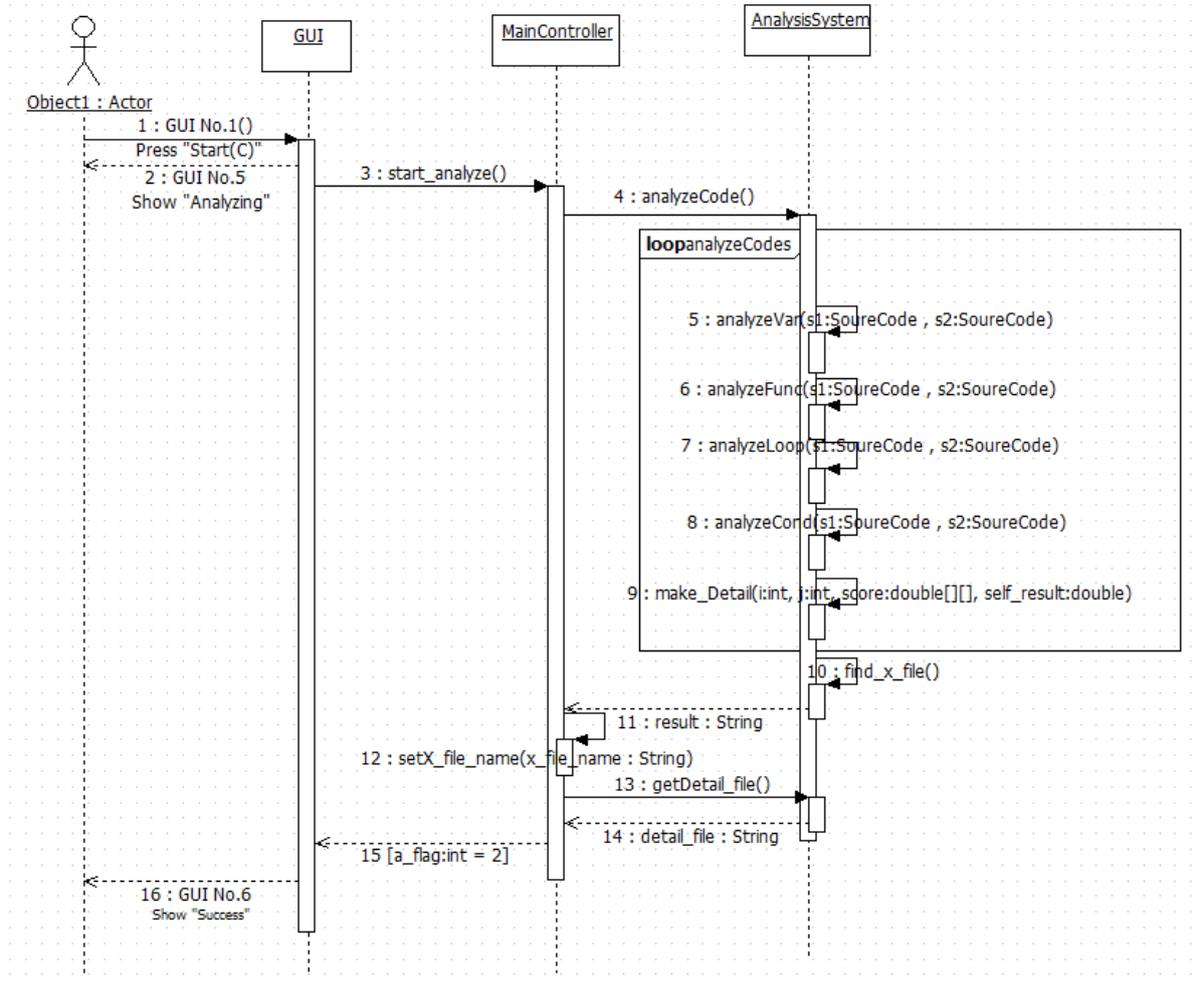
[ Display Main ]



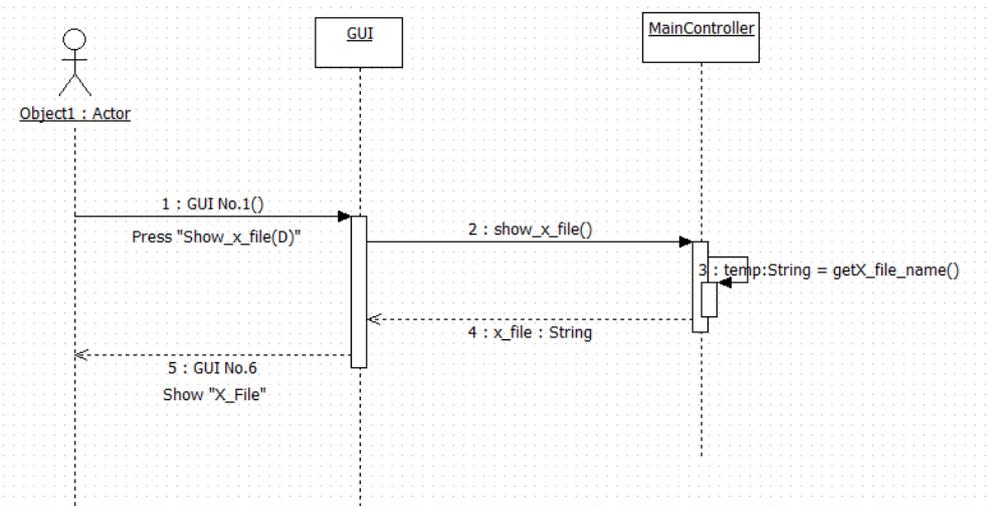
[ Input Path ]



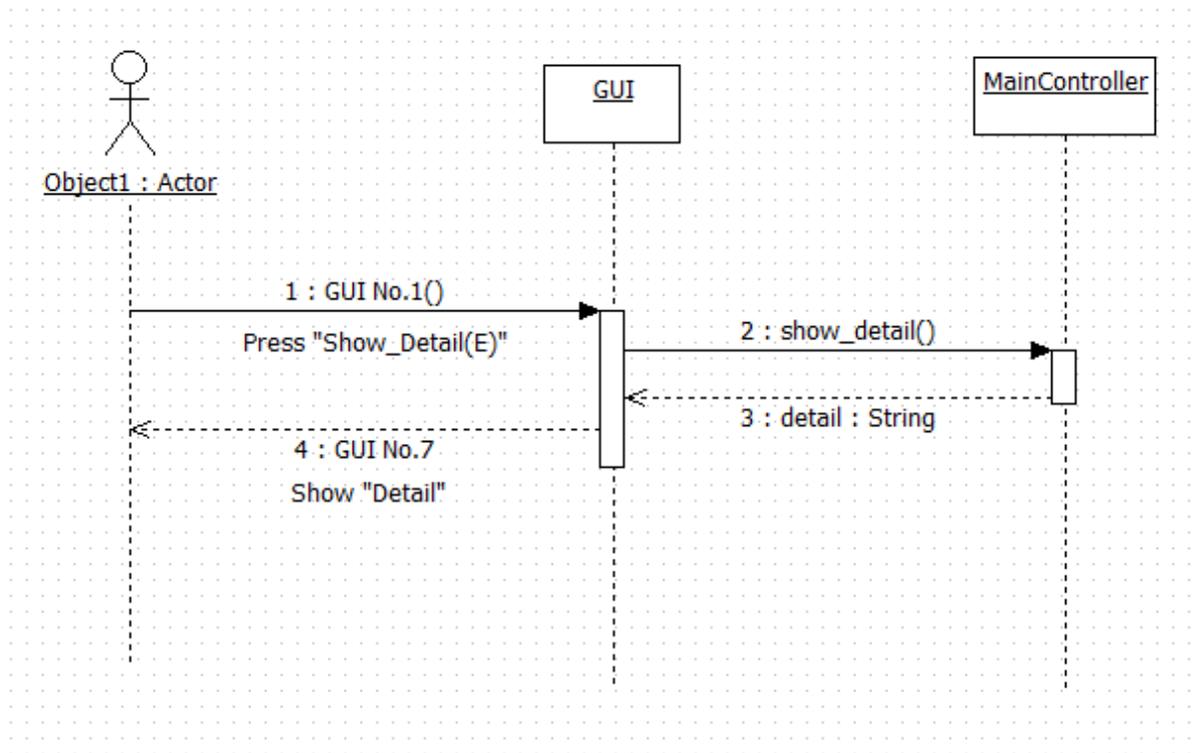
[ Start Analyze ]



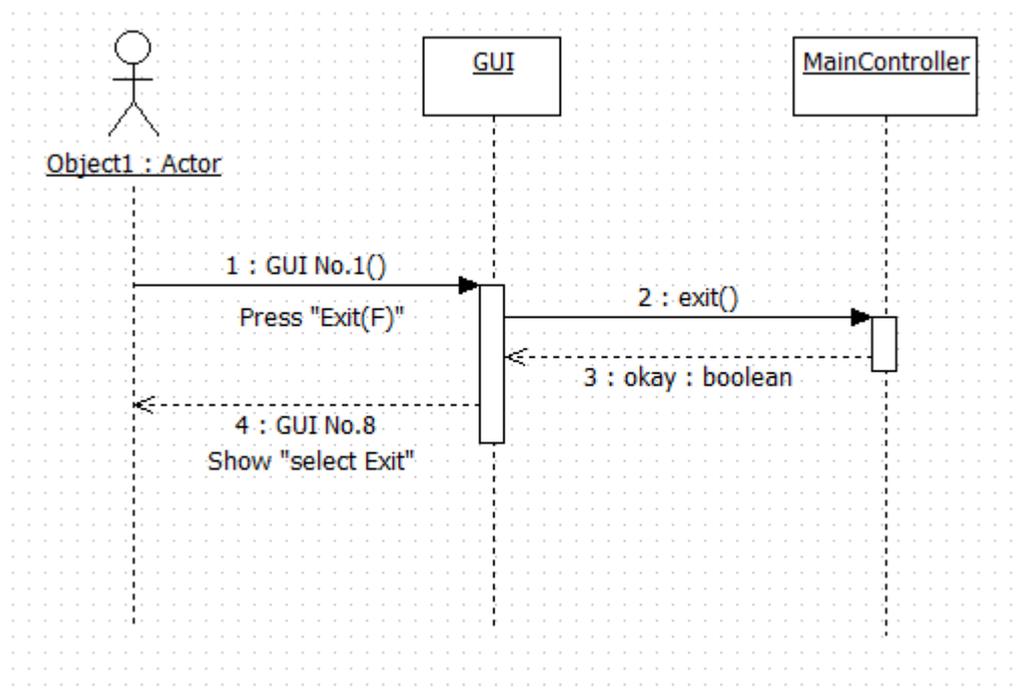
[ Show X File ]



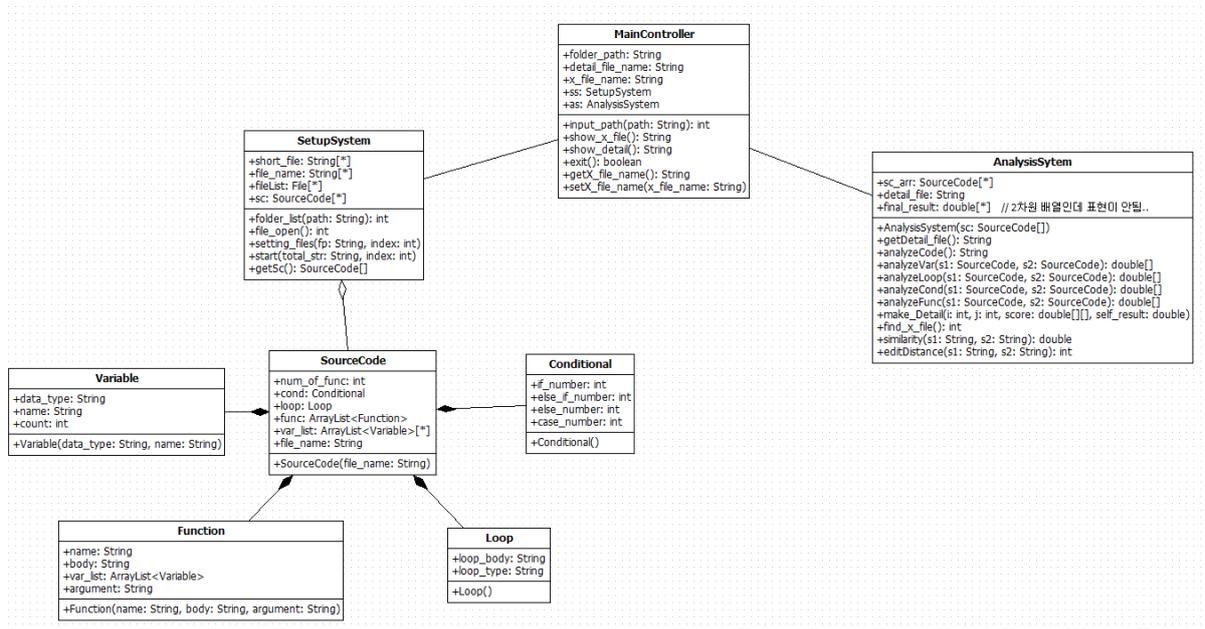
## [ Show Detail ]



## [ Exit ]



## Activity 2045. Define Design Class Diagrams



## Activity 2046. Design Traceability Analysis

