OOPT Stage 2050

<Construct>

Software Modeling & Analysis

소프트웨어 모델링 및 분석

보고서

Team. T2

201411278 서희진

201411304 이지수

201411317 조민규

201213189 박성규

Stage 2050. Construct

Activity 2151. Implement Class & Methods Definitions

[Client]

<model - Version>

Type	Class
Name	Version
Purpose	Version 의 정보를 모아두는 클래스
Overview (Class)	getCtgNum,, getChoiceNum, getConsNum, addCategory, addCase, getCaseNum, getCtgList, getWriter, setWriter, setChoiceNum, setConsNum, setCtgList, setCaseNum, getVerNum, addChoice, getFile, setFile, getOpinionList, setOpinionList 를 가지고있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional R.5.1, R.5.2, R.5.3, R.6, R.7
Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.7
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Version 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : AddVer,AnalyzeFile,MakeCase,DelVer,DownloadFile
Input (Method)	String verNum
Output (Method)	-

Abstract Operation (Method)	Version 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : AddVer,AnalyzeFile,MakeCase,DelVer,DownloadFile
Input (Method)	String verNum, File file
Output (Method)	-
Abstract Operation (Method)	Version 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getCtgNum()
Purpose	Category 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Int this.cctgList.size
Abstract Operation (Method)	*CtgList.size* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceNum()
Purpose	Choice 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int choiceNum

Abstract Operation (Method)	*choiceNum* 을 반환.
Exceptional Courses of Events	-

Type	Method
Name	getConsNum()
Purpose	Constraints 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int consNum
Abstract Operation (Method)	*consNum* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addCategory()
Purpose	CategoryList 에 Category 객체를 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Category ctg
Output (Method)	-
Abstract Operation (Method)	*CtgList* 에 *Cateogry* 객체를 추가한다.
Exceptional Courses of Events	-

Type	Method
Name	addCase()
Purpose	Test Case 의 수를 하나 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-

Abstract Operation (Method)	케이스의 숫자를 추가한다.
Exceptional Courses of Events	-

Type	Method
Name	addCase()
Purpose	Test Case 의 수를 caseNum 만큼 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	int caseNum
Output (Method)	-
Abstract Operation (Method)	기존의 case 의 개수에 caseNum 만큼 추가한다.
Exceptional Courses of Events	-

Type	Method
Name	getCaseNum()
Purpose	Test Case 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Int caseNum
Abstract Operation (Method)	*CaseNum 의 개수를 반환한다.*
Exceptional Courses of Events	-

Type	Method
Name	getCtgList()
Purpose	CategoryList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	LinkedList ctgList
Abstract Operation (Method)	*ctgList* 를 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	getWriter()
Purpose	작성자를 가져온다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	-
Output (Method)	String writer
Abstract Operation (Method)	*writer*을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setWriter()
Purpose	작성자를 저장한다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	String writer
Output (Method)	-
Abstract Operation (Method)	*writer*을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setChoiceNum()
Purpose	Choice 의 개수를 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Int choiceNum
Output (Method)	-
Abstract Operation (Method)	*choiceNum* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setConsNum()
Purpose	Cons 의 개수를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.53 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Int consNum
Output (Method)	-
Abstract Operation (Method)	*consNum* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setCtgList()
Purpose	CtgList 를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	LinkedList<Category> ctgList
Output (Method)	-
Abstract Operation (Method)	*CtgList* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setCaseNum()
Purpose	CtgList 를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	int caseNum
Output (Method)	-
Abstract Operation (Method)	*caseNum* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getVerNum()
Purpose	verNum 를 가져온다
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	-
Output (Method)	String verNum
Abstract Operation (Method)	verNum 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	setVerNum()
Purpose	verNum 를 저장한다
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	String verNum
Output (Method)	-
Abstract Operation (Method)	VerNum 을 저장한다.

Exceptional Courses of Events	-

Type	Method
Name	addChoice()
Purpose	choiceNum 을 증가한다
Cross Reference	UseCase : R.5.1, R.5.1, R.5.2 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Choice 의 숫자를 증가한다.
Exceptional Courses of Events	-

Type	Method
Name	getFile()
Purpose	File 을 가져온다.
Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	-
Output (Method)	File file
Abstract Operation (Method)	*File file* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setFile()
Purpose	File 을 저장한다.
Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	File file
Output (Method)	-
Abstract Operation (Method)	*File file* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getOpinionList()
Purpose	OpinionList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.8, R.9 Functional Requirements : R.5.1, R.8, R.9
Input (Method)	-
Output (Method)	LinkedList<Opinion> opinionList
Abstract Operation (Method)	OpinionList 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	setOpinionList()
Purpose	OpinionList 를 저장한다.
Cross Reference	UseCase : R.5.1, R.8, R.9 Functional Requirements : R.5.1, R.8, R.9
Input (Method)	LinkedList<Opinion> opinionList
Output (Method)	-
Abstract Operation (Method)	OpinionList 를 저장한다.
Exceptional Courses of Events	-

<model - Category>

Type	Class
Name	Category
Purpose	Category 의 정보를 모아두는 클래스
Overview (Class)	ctgName,choiceList,singleList,errorList 를 가지고있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7
Exceptional Courses of Events	-

Type	Method
Name	Category()
Purpose	Category 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7
Input (Method)	String ctgName
Output (Method)	-
Abstract Operation (Method)	Category 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	addChoice()
Purpose	ChoiceList 에 Choice 객체를 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7
Input (Method)	Choice choice
Output (Method)	-
Abstract Operation (Method)	Choice 가 error 가 아니고 single 이 아니면 *ChoiceList.add(choice)*
Exceptional Courses of Events	-

Type	Method
Name	getCtgName()
Purpose	CtgName 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	String ctgName
Abstract Operation (Method)	*CtgName* 을 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceList()
Purpose	chocieList 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	LinkedList<Choice> choiceList
Abstract Operation (Method)	*choiceList* 을 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	addSingle()
Purpose	SingleList 에 single 을 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String single
Output (Method)	-
Abstract Operation (Method)	*singleList.add(single)*
Exceptional Courses of Events	-

Type	Method
Name	addError()
Purpose	ErrorList 에 error 를 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String error
Output (Method)	-
Abstract Operation (Method)	*errorList.add(error*
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	getSingleNum()
Purpose	SingleList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*singleList.size()*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getErrorNum()
Purpose	ErrorList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*errorList.size()*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceNum()
Purpose	ChoiceList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*choiceList.size()*를 반환한다.
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	getSingleList()
Purpose	SingleList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArryList<String> singleList
Abstract Operation (Method)	*singleList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getErrorList()
Purpose	ErrorList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> errorList
Abstract Operation (Method)	*errorList* 를 반환한다.
Exceptional Courses of Events	-

<model – Choice>

Type	Class
Name	Choice
Purpose	Choice 의 정보를 모아두는 클래스
Overview (Class)	choiceName,cosNum,consList,hasProp,hasSingle,hasError,hasIf,hasElse,propList,ifList,elseList 을 가지고 있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Exceptional Courses of Events	-

Type	Method
Name	Choice()
Purpose	Choice 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String choiceName
Output (Method)	-
Abstract Operation (Method)	Choice 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceName()
Purpose	Choice 의 이름을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	String choiceName
Abstract Operation (Method)	*ChoiceName* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addCons()
Purpose	ConsList 에 Constraints 객체를 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Constraints cons
Output (Method)	-
Abstract Operation (Method)	*consList.add(cons)* *consNum++*
Exceptional Courses of Events	-

Type	Method
Name	addIf()
Purpose	IfList 에 ifString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String ifString
Output (Method)	-
Abstract Operation (Method)	[Choice 객체가 Error Constraints 를 가지고 있지 않거나 Single Constraints 를 가지고 있지 않는 경우] *ifList.add(ifString)* *hasIf* 에 *true* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	addElse()
Purpose	elseList 에 elseString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String elseString
Output (Method)	-
Abstract Operation (Method)	*elseList.add(elseString)* *hasElse* 에 *true* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	addProp()
Purpose	propList 에 propString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String propString
Output (Method)	-

Abstract Operation (Method)	[Choice 객체가 Error Constraints 를 가지고 있지 않거나 Single Constraints 를 가지고 있지 않는 경우] *propList.add(propString)* *hasProp* 에 *true* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsList()
Purpose	consList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Array<Constraints> consList
Abstract Operation (Method)	*consList* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isProp()
Purpose	hasProp 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasProp
Abstract Operation (Method)	hasProp 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isSingle()
Purpose	hasSingle 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-

Output (Method)	Boolean hasSingle
Abstract Operation (Method)	hasSingle 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isError()
Purpose	hasError 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasError
Abstract Operation (Method)	hasError 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsProp()
Purpose	hasProp 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasProp
Abstract Operation (Method)	hasProp 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsSingle()
Purpose	hasSingle 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasSingle

Abstract Operation (Method)	hasSingle 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsError()
Purpose	hasError 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasError
Abstract Operation (Method)	hasError 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsElse()
Purpose	hasElse 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasElse
Abstract Operation (Method)	hasElse 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getPropList()
Purpose	propList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> propList

Abstract Operation (Method)	PropList 를 ArrayList<String> 형태로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getifList()
Purpose	ifList 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> ifList
Abstract Operation (Method)	ifList 을 ArrayList<String> 형태로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isIf()
Purpose	hasIf 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasIf
Abstract Operation (Method)	hasIf 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isElse()
Purpose	hasElse 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasElse
Abstract Operation (Method)	hasElse 을 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	getElseList()
Purpose	elseList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList elseList
Abstract Operation (Method)	*elseList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsNum()
Purpose	Constraints 의 개수를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int consNum
Abstract Operation (Method)	*consNum* 을 반환한다.
Exceptional Courses of Events	-

<model - Constraints>

Type	Class
Name	Constraints
Purpose	Constraints 의 정보를 모아두는 클래스
Overview (Class)	consName 을 가지고 있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Exceptional Courses of Events	N/A

Type	Method
Name	Constraints()
Purpose	Constraints 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String consName
Output (Method)	-
Abstract Operation (Method)	Constraints 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsName()
Purpose	ConsName 을 반환하는 메소드.
Overview (Class)	ConsName 을 반환한다.
Cross Reference	
Input (Method)	N/A
Output (Method)	String
Abstract Operation (Method)	ConName 을 반환한다.
Exceptional Courses of Events	N/A

<model - Opinion>

Type	Class
Name	Opinion
Purpose	Opinion 정보를 모아두는 클래스
Overview (Class)	OpinionNum, opinionID, date, content 을 가지고 있다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Exceptional Courses of Events	-

Type	Method
Name	Opinion()
Purpose	Opinion 의 객체를 생성하는 함수
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String content
Output (Method)	-
Abstract Operation (Method)	Opinion 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getOpinionID()
Purpose	OpinionID 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String opinionID
Abstract Operation (Method)	*OpinionID* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setOpinionID()
Purpose	OpinionID 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String opinionID
Output (Method)	-
Abstract Operation (Method)	*opinionID 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	getDate()
Purpose	date 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String date
Abstract Operation (Method)	*date* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setDate()
Purpose	date 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String date
Output (Method)	void
Abstract Operation (Method)	*date 를 저장한다.*
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	getContent()
Purpose	content 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String content
Abstract Operation (Method)	*content* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setContent()
Purpose	content 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String content
Output (Method)	-
Abstract Operation (Method)	*content 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	setOpinionNum()
Purpose	OpinionNum 을 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	Int opinionNum
Output (Method)	void
Abstract Operation (Method)	*opinionNum 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	getOpinionNum()
Purpose	opinionNum 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	Int opinionNum
Abstract Operation (Method)	*opinionNum* 을 반환한다.
Exceptional Courses of Events	-

<view – MainUI>

Type	Class
Name	MainUI
Purpose	프로그램의 GUI 를 제어하는 클래스
Overview (Class)	프로그램의 GUI 를 제어한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	setPList()
Purpose	Server 로부터 가져온 정보를 통해 ProjectList 를 재설정한다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4 Functional Requirements : R.1, R.2, R.3, R.4
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Server 로부터 가져온 정보를 통해 ProjectList 를 재설정하고 *projectListPanel* 을 재출력한다.
Exceptional Courses of Events	-

Type	Method
Name	setVList()
Purpose	Server 로부터 가져온 정보를 통해 VersionList 를 재설정한다.

Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Server 로부터 가져온 정보를 통해 VersionList 를 재설정하고 *versionListPanel* 을 재출력한다.
Exceptional Courses of Events	-

Type	Method
Name	setOList()
Purpose	Server 로부터 가져온 정보를 통해 OpinionList 를 재설정한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Server 로부터 가져온 정보를 통해 OpinionList 를 재설정하고 *opinionListPanel* 을 재출력한다.
Exceptional Courses of Events	-

<controller – Main>

Type	Class
Name	Main
Purpose	프로그램을 시작하기 위한 클래스
Overview (Class)	
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

<controller – MainController>

Type	Class
Name	MainController
Purpose	프로그램의 수행을 제어하는 클래스
Overview (Class)	User, Project, Version, Case, Server 와의 통신 등을 관리한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	MainController()
Purpose	MainController 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	MainController 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	int()
Purpose	프로그램을 초기화 하고 Server 와 연결한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	InetAddress 인 *address* 를 설정하고 Socket 을 연결한다. Client 와 AdminUser 와 AdminVersion 과 AdminProject 의 객체를 생성한다.
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	getClient()
Purpose	Client 의 객체를 가져온다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	Client client
Abstract Operation (Method)	*client* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getAdminU()
Purpose	AdminUser 의 객체를 가져온다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	AdminUser adminU
Abstract Operation (Method)	*adminU* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getAdminP()
Purpose	Project 의 객체를 가져온다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	AdminProject adminP
Abstract Operation (Method)	*adminP* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getAdminV()
Purpose	AdminVersion 의 객체를 가져온다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	AdminVersion adminV
Abstract Operation (Method)	*adminV* 를 반환한다.
Exceptional Courses of Events	-

<controller - AdminUser>

Type	Class
Name	AdminUser
Purpose	User 와 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	Server 와 통신하여 Login, ChangePW, Logout 을 한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	AdminUser()
Purpose	AdminUser 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Client client
Output (Method)	-
Abstract Operation (Method)	AdminUser 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	login()
Purpose	Login 한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	String ID, String PW
Output (Method)	Boolean
Abstract Operation (Method)	Server 로 Login 요청과 ID 와 PW 를 전송한다. Server 로부터 *msg* 를 수신 받아 Login 성공 여부를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	chgPW()
Purpose	Password 를 새로운 Password 로 변경한다.
Cross Reference	UseCase : R.0, R.1, R.2 Functional Requirements : R.0, R.1, R.2
Input (Method)	String PW, String PW1, String PW2
Output (Method)	Int
Abstract Operation (Method)	*isSame(PW1,PW2);* Server 로 ChgPW 요청과 *PW, PW1* 을 전송한다. Server 로부터 *msg* 를 수신 받아 chgPW 결과의 경우의 수를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isSame()
Purpose	새로 입력한 Password1 과 Password2 가 일치하는지 확인한다.
Cross Reference	UseCase : R.0, R.1, R.2 Functional Requirements : R.0, R.1, R.2
Input (Method)	String PW1, String PW2
Output (Method)	boolean
Abstract Operation (Method)	*PW1* 과 *PW2* 가 일치하는지 확인한다.
Exceptional Courses of Events	-

Type	Method
Name	exit()
Purpose	시스템을 종료한다.
Cross Reference	UseCase : R.10 Functional Requirements : R.10
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Server 에 *Exit* 요청을 전송한다. 프로그램을 종료한다.
Exceptional Courses of Events	-

Type	Method
Name	getID()
Purpose	ID 를 가져온다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9 Functional Requirements : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9
Input (Method)	-
Output (Method)	String ID
Abstract Operation (Method)	*ID* 를 반환한다.
Exceptional Courses of Events	-

<controller – AdminProject>

Type	Class
Name	AdminProject
Purpose	Project 와 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	HashMap ProjectList 와 Client 객체와 HashSet projectNameSet 을 가지고 있다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Exceptional Courses of Events	-

Type	Method
Name	AdminProject()
Purpose	AdminProject 의 객체를 생성한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	Client client
Output (Method)	-
Abstract Operation (Method)	AdminProject 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	addProject()
Purpose	Project 를 추가한다.
Cross Reference	UseCase : R.3 Functional Requirements : R.3
Input (Method)	String projectName
Output (Method)	boolean
Abstract Operation (Method)	Server 로 AddProject 요청과 *projectName* 을 전송한다. Server 로부터 *msg* 를 수신 받아 addProject 의 결과를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	delProject
Purpose	Project 를 삭제한다.
Cross Reference	UseCase : R.4 Functional Requirements : R.4
Input (Method)	String projectName
Output (Method)	Boolean
Abstract Operation (Method)	Server 로 DelProject 요청과 *projectName* 을 전송한다. Server 로부터 *msg* 를 수신 받아 delProject 의 결과를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	rcvProjectList()
Purpose	Server 로부터 ProjectList 를 수신한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	-
Output (Method)	HashSet projectNameSet
Abstract Operation (Method)	Server 로 rcvProjectList 요청을 전송한다. Server 로부터 *ProjectList* 를 수신 받아 *projectNameSet* 에 저장하고 *projectNameSet* 을 반환한다.
Exceptional Courses of Events	-

<controller – AdminVer>

Type	Class
Name	AdminVer
Purpose	Version 과 Opinion 과 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	Ver 과 Opinion 과 관련된 List 를 관리한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9
Exceptional Courses of Events	N/A

Type	Method
Name	AdminVer()
Purpose	AdminVer 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9
Input (Method)	Client client
Output (Method)	-
Abstract Operation (Method)	AdminVer 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	AdminVer()
Purpose	AdminVer 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.7, R.8, R.9
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	AdminVer 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	addVer()
Purpose	txt 확장자인지 확인하는 메소드
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	File file
Output (Method)	boolean
Abstract Operation (Method)	파일 이름을 string 으로 받아와 마지막 확장자 부분이 .txt 인지 아닌지를 확인하여 Boolean 값을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	analyzeFile()
Purpose	파일을 분석하여 배열에 분류하는 메소드
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String filename
Output (Method)	Version ver
Abstract Operation (Method)	파일을 받아와 []를 기준으로 [] 사이에 있는 문자열을 배열에 분류한다.
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	analyzeFile()
Purpose	파일을 분석하여 각각의 속성에 맞는 Version 객체를 만들어 내는 메소드
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String filename
Output (Method)	Version ver
Abstract Operation (Method)	txt 파일에 있는 구분자를 기준으로 Category 와 Choice 그리고 Constraints 를 구분하는 것을 기준으로 한다. Category 의 경우는 :을 기준으로 분리하고 Choice 의 경우는 .을 기준으로 분리한다. 각각의 Constraints 는 []을 기준으로 분리하고 있는데 property 가 있는 경우에는 해당 choice 의 propList 에 추가한 후 Category 에 추가하기 위하여 addFlag 가 True 로 바뀐다. If 의 경우도 해당 choice 의 ifList 에 추가한 후 Category 에 추가하기 위하여 addFlag 가 True 로 바뀐다. 하지만 Single 이나 Error 와 같은 경우에는 Case 를 세는 경우에서 제외되어야 하므로 addFlag 가 False 가 되어 Category 의 choice 로 add 되지 않는다. 그리하여 각 Category 와 그에 대한 Choice 그리고 그 안의 Constraints 를 가지는 Version 객체를 반환한다.

Type	Method
Name	makeCase()
Purpose	
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String filename
Output (Method)	Version ver
Abstract Operation (Method)	Version 객체를 가져와 먼저 count 와 single Error 의 개수를 세고 전체 테스트 케이스를 makeTmpFile 안에서 조합한다.
Exceptional Courses of Events	-

Type	Method
Name	countSingleError()
Purpose	Single 과 error 의 개수를 합하여 반환한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Version ver
Output (Method)	
Abstract Operation (Method)	Version 객체를 가져와 먼저 count 와 single Error 의 개수를 세고 전체 테스트 케이스를 makeTmpFile 안에서 조합한다. 만약 특정 카테고리의 choice 가 없다면 Nothing Choice 를 추가하여 준다. 그리고 해당 카테고리가 If 만을 가지고 있는 경우에는 아무것도 선택하지 않는 경우가 발생할 수 있으므로 Nothing Choice 를 추가하여 준다.
Exceptional Courses of Events	-

Type	Method
Name	makeTmpFile (),
Purpose	Single 과 error 의 개수를 temp.txt 에 출력한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Version ver, BufferedWriter bw
Output (Method)	
Abstract Operation (Method)	Version 객체를 가져와 먼저 count 와 single Error 의 개수를 세고 전체 테스트 케이스를 makeTmpFile 안에서 조합한다.
Exceptional Courses of Events	-

Type	Method
Name	makeTmpFile (),
Purpose	Single 과 error 의 개수를 합하여 반환한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Int[] pointerArr, LinkedList<Choice>choiceListArr, int totalSize, int idx, Version ver, BufferedWriter bw
Output (Method)	
Abstract Operation (Method)	여기서 재귀적으로 각각의 모든 Chocie 들을 출력한 후 그 경우에 모아있는 ifList 에 해당하는 propList 가 모두 존재한다면 txt 파일에 작성한 후 testCase 의 개수를 늘려준다. 추가된 Nothing 의 경우도 추가적으로 예외처리를 해준다.
Exceptional Courses of Events	-

Type	Method
Name	uploadVer()
Purpose	projectName, Version 객체를 서버에 전송 요청을 하는 메소드
Cross Reference	UseCase : R.5.1 Functional Requirements : R.5.1
Input (Method)	String projectName, Version ver
Output (Method)	Boolean
Abstract Operation (Method)	Txt 확장자의 파일이므로 projectName, Version 객체를 전송하여 확인 메시지를 통해 Boolean 을 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	delVer()
Purpose	서버에서 projectName 과 verNum 을 삭제 요청을 하는 메소드
Cross Reference	UseCase : R.6 Functional Requirements : R.6
Input (Method)	String projectName, String verNum
Output (Method)	Boolean
Abstract Operation (Method)	서버에 projectName, verNum 삭제 명령을 보내고, 서버에서 삭제가 되었는지 여부를 받아 boolean 으로 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	addOpinion()
Purpose	서버에 매개변수 추가 요청을 하는 메소드
Cross Reference	UseCase : R.8 Functional Requirements : R.8
Input (Method)	Opinion opinion, String projectName, String verNum
Output (Method)	-
Abstract Operation (Method)	서버에 projectName, verNum, opinion 추가 요청을 한다.
Exceptional Courses of Events	N/A

Type	Method
Name	delOpinion()
Purpose	서버에 매개변수로 받은 정보들의 삭제 요청을 하는 메소드
Cross Reference	UseCase : R.9 Functional Requirements : R.9
Input (Method)	String projectName, String verNum, String opinionNum
Output (Method)	Boolean
Abstract Operation (Method)	서버에 projectName, verNum, opinionNum 의 정보를 보낸 후 삭제 시도를 한다. 삭제 성공 여부를 boolean 으로 반환한다..
Exceptional Courses of Events	N/A

Type	Method
Name	downloadFile()
Purpose	매개변수와 일치하는 파일을 받아와 저장하는 메소드
Cross Reference	UseCase : R.7 Functional Requirements : R.7
Input (Method)	String projectName, String verNum
Output (Method)	File file
Abstract Operation (Method)	서버에 매개변수를 전송해서 일치하는 파일을 받아와 설정한 경로에 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	rcvVerList()
Purpose	서버로부터 projectName 에 해당하는 VerList 를 가져오는 메소드
Cross Reference	UseCase : R.5.1, R.6, R.7 Functional Requirements : R.5.1, R.6, R.7
Input (Method)	String projectName
Output (Method)	LinkedList<Version>
Abstract Operation (Method)	서버에서 projectName 에 해당하는 verList 를 LinkedList<Version>형태로 저장하고 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	rcvOpinionList()
Purpose	ProjectName 과 verNum 와 일치하는 opinionList 를 가져오는 메소드
Cross Reference	UseCase : R.8, R9 Functional Requirements : R.8, R9
Input (Method)	String projectName, String verNum
Output (Method)	LinkedList<Opinion>
Abstract Operation (Method)	매개변수로 받은 projectName 과 verNum 을 서버에 전송하여, 일치하는 opinionList 를 서버로부터 받아오고 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isContain()
Purpose	배열에 String str 이 존재하는지를 boolean 으로 표시하는 메소드
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String str, ArrayList<String> arr
Output (Method)	boolean
Abstract Operation (Method)	Input 으로 받은 배열에 Input 으로 받은 String str 이 존재하는지를 boolean 형태로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getTmpVer()
Purpose	TempVer 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Version
Abstract Operation (Method)	TempVer 을 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	setTmpVer()
Purpose	tempVer 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Version tempVer
Output (Method)	-
Abstract Operation (Method)	tempVer 을 저장한다.
Exceptional Courses of Events	-

<controller – AdminCase>

Type	Class
Name	AdminCase
Purpose	Case 와 관련된 정보를 저장하는 클래스
Overview (Class)	Case 와 관련된 정보를 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Exceptional Courses of Events	N/A

Type	Method
Name	AdminCase()
Purpose	AdminCase 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	AdminCase 의 객체를 생성한다.

Exceptional Courses of Events	-

Type	Method
Name	getChoiceList()
Purpose	ChoiceList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<Choice>
Abstract Operation (Method)	ChoiceList 를 ArrayList<Choice>로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setChoiceList()
Purpose	ChoiceList 를 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	ArrayList<Choice> choiceList
Output (Method)	-
Abstract Operation (Method)	choiceList 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	isProp()
Purpose	Property 의 존재 유무를 boolean 으로 반환한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Property 의 존재 유무를 boolean 으로 반환한다.
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	setHasProp()
Purpose	hasProp 를 true 로 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Boolean hasProp
Output (Method)	-
Abstract Operation (Method)	hasProp 를 true 로 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	isSingle()
Purpose	Single 의 존재 유무를 boolean 으로 반환한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Single 의 존재 유무를 boolean 으로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setHasSingle()
Purpose	hasSingle 를 true 로 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Boolean hasSinle
Output (Method)	-
Abstract Operation (Method)	hasSingle 를 true 로 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	isError()
Purpose	Error 의 존재 유무를 boolean 으로 반환한다.

Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Error 의 존재 유무를 boolean 으로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setHasError()
Purpose	hasError 를 true 로 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Boolean hasError
Output (Method)	-
Abstract Operation (Method)	hasError 를 true 로 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getPropList()
Purpose	Property List 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String>
Abstract Operation (Method)	propList 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setPropList()
Purpose	Input 값 propList 를 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	ArrayList<String> propList

Output (Method)	-
Abstract Operation (Method)	Input 값 propList 를 저장한다.
Exceptional Courses of Events	-

<controller – Client>

Type	Class
Name	Client
Purpose	Server 와 통신하는 클래스
Overview (Class)	Server 와 msg, file, obj 를 주고받는다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	Client()
Purpose	Cllient 객체를 생성하는 메소드
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Socket sock
Output (Method)	-
Abstract Operation (Method)	Client 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	sendToServer()
Purpose	서버에 메시지를 보내는 메소드
Cross Reference	UseCase : All Functional Requirements : All

Input (Method)	String msg
Output (Method)	-
Abstract Operation (Method)	서버에 보낸 msg 를 출력하고, 소켓을 비운다.
Exceptional Courses of Events	-

Type	Method
Name	rcvFromServer()
Purpose	서버의 소켓을 받는 메소드.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	String
Abstract Operation (Method)	서버에서 소켓을 받아서 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	sendFile()
Purpose	파일을 보내기 위한 메소드
Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	path 에서의 "test.txt"를 찾아와서 Checksum 을 수행한 뒤, 전 byteFile 로 인코딩 한 후 전송한다.
Exceptional Courses of Events	-

Type	Method
Name	rcvFile()
Purpose	파일을 받기 위한 메소드

OOPT Stage 2050 <Construct>

Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	-
Output (Method)	File
Abstract Operation (Method)	서버로부터 Byte 배열을 받아와서 decode 를 수행하고, 파일을 IOutFile 에 저장하고 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	SendObjToServer()
Purpose	Object 를 서버에 전송하기 위한 메소드
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Object object
Output (Method)	-
Abstract Operation (Method)	Opinion 객체 리스트 또는 Version 객체 리스트 또는 프로젝트 리스트를 서버에 전송한다.
Exceptional Courses of Events	-

Type	Method
Name	rcvObjFromServer()
Purpose	Object 를 서버에서 받아오기 위한 메소드
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	Object object
Abstract Operation (Method)	Opinion 객체 리스트, Version 객체 리스트, 프로젝트 리스트를 서버로부터 받아와서 object 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	sendByte()
Purpose	객체들을 서버에 보내는 메소드
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Object object
Output (Method)	-
Abstract Operation (Method)	Opinion 객체 리스트, Version 객체 리스트, 프로젝트 리스트를 Byte 형태로 서버에 전송한다.
Exceptional Courses of Events	-

Type	Method
Name	rcvByte()
Purpose	객체들을 서버에 보내는 메소드
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Object object
Output (Method)	-
Abstract Operation (Method)	Opinion 객체 리스트, Version 객체 리스트, 프로젝트 리스트를 Byte 형태로 서버에서 받는다.
Exceptional Courses of Events	-

[Server]

<model – Version>

Type	Class
Name	Version
Purpose	Version 의 정보를 모아두는 클래스
Overview (Class)	getCtgNum,, getChoiceNum, getConsNum, addCategory, addCase, getCaseNum, getCtgList, getWriter, setWriter, setChoiceNum, setConsNum, setCtgList, setCaseNum, getVerNum, addChoice, getFile, setFile, getOpinionList, setOpinionList 를 가지고있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional R.5.1, R.5.2, R.5.3, R.6, R.7
Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.7
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Version 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : AddVer,AnalyzeFile,MakeCase,DelVer,DownloadFile
Input (Method)	String verNum
Output (Method)	-
Abstract Operation (Method)	Version 의 객체를 생성한다.

Exceptional Courses of Events	-

Type	Method
Name	Version()
Purpose	Version 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6, R.7 Functional Requirements : AddVer,AnalyzeFile,MakeCase,DelVer,DownloadFile
Input (Method)	String verNum, File file
Output (Method)	-
Abstract Operation (Method)	Version 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getCtgNum()
Purpose	Category 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Int this.cctgList.size
Abstract Operation (Method)	*CtgList.size* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceNum()
Purpose	Choice 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int choiceNum
Abstract Operation (Method)	*choiceNum* 을 반환.

Exceptional Courses of Events	-

Type	Method
Name	getConsNum()
Purpose	Constraints 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int consNum
Abstract Operation (Method)	*consNum* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addCategory()
Purpose	CategoryList 에 Category 객체를 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Category ctg
Output (Method)	-
Abstract Operation (Method)	*CtgList* 에 *Cateogry* 객체를 추가한다.
Exceptional Courses of Events	-

Type	Method
Name	addCase()
Purpose	Test Case 의 수를 하나 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	케이스의 개수를 추가한다.

Exceptional Courses of Events	-

Type	Method
Name	addCase()
Purpose	Test Case 의 수를 caseNum 만큼 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	int caseNum
Output (Method)	-
Abstract Operation (Method)	기존의 case 의 개수에 caseNum 만큼 추가한다.
Exceptional Courses of Events	-

Type	Method
Name	getCaseNum()
Purpose	Test Case 의 개수를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Int caseNum
Abstract Operation (Method)	*CaseNum 의 개수를 반환한다.*
Exceptional Courses of Events	-

Type	Method
Name	getCtgList()
Purpose	CategoryList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	LinkedList ctgList
Abstract Operation (Method)	*ctgList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getWriter()
Purpose	작성자를 가져온다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	-
Output (Method)	String writer
Abstract Operation (Method)	*writer* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setWriter()
Purpose	작성자를 저장한다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	String writer
Output (Method)	-
Abstract Operation (Method)	*writer* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setChoiceNum()
Purpose	Choice 의 개수를 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Int choiceNum
Output (Method)	-
Abstract Operation (Method)	*choiceNum* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setConsNum()
Purpose	Cons 의 개수를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.53 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Int consNum
Output (Method)	-
Abstract Operation (Method)	*consNum* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setCtgList()
Purpose	CtgList 를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	LinkedList<Category> ctgList
Output (Method)	-
Abstract Operation (Method)	*CtgList* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setCaseNum()
Purpose	CtgList 를 저장한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	int caseNum
Output (Method)	-
Abstract Operation (Method)	*caseNum* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getVerNum()
Purpose	verNum 를 가져온다
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	-
Output (Method)	String verNum
Abstract Operation (Method)	verNum 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	setVerNum()
Purpose	verNum 를 저장한다
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	String verNum
Output (Method)	-
Abstract Operation (Method)	VerNum 을 저장한다.

Exceptional Courses of Events	-

Type	Method
Name	addChoice()
Purpose	choiceNum 을 증가한다
Cross Reference	UseCase : R.5.1, R.5.1, R.5.2 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Choice 의 숫자를 증가한다.
Exceptional Courses of Events	-

Type	Method
Name	getFile()
Purpose	File 을 가져온다.
Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	-
Output (Method)	File file
Abstract Operation (Method)	*File file* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setFile()
Purpose	File 을 저장한다.
Cross Reference	UseCase : R.3, R.5.1, R.7 Functional Requirements : R.3, R.5.1, R.7
Input (Method)	File file
Output (Method)	-
Abstract Operation (Method)	*File file* 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getOpinionList()
Purpose	OpinionList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.8, R.9 Functional Requirements : R.5.1, R.8, R.9
Input (Method)	-
Output (Method)	LinkedList<Opinion> opinionList
Abstract Operation (Method)	OpinionList 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	setOpinionList()
Purpose	OpinionList 를 저장한다.
Cross Reference	UseCase : R.5.1, R.8, R.9 Functional Requirements : R.5.1, R.8, R.9
Input (Method)	LinkedList<Opinion> opinionList
Output (Method)	-
Abstract Operation (Method)	OpinionList 를 저장한다.
Exceptional Courses of Events	-

<model – Category>

Type	Class
Name	Category
Purpose	Category 의 정보를 모아두는 클래스
Overview (Class)	ctgName,choiceList,singleList,errorList 를 가지고있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7

Exceptional Courses of Events	-

Type	Method
Name	Category()
Purpose	Category 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7
Input (Method)	String ctgName
Output (Method)	-
Abstract Operation (Method)	Category 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	addChoice()
Purpose	ChoiceList 에 Choice 객체를 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3,R.6,R.7 Functional Requirements : R.5.1, R.5.2, R.5.3 R.6,R.7
Input (Method)	Choice choice
Output (Method)	-
Abstract Operation (Method)	Choice 가 error 가 아니고 single 이 아니면 *ChoiceList.add(choice)*
Exceptional Courses of Events	-

Type	Method
Name	getCtgName()
Purpose	CtgName 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	String ctgName
Abstract Operation (Method)	*CtgName* 을 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceList()
Purpose	chocieList 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	LinkedList<Choice> choiceList
Abstract Operation (Method)	*choiceList* 을 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	addSingle()
Purpose	SingleList 에 single 을 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String single
Output (Method)	-
Abstract Operation (Method)	*singleList.add(single)*
Exceptional Courses of Events	-

Type	Method
Name	addError()
Purpose	ErrorList 에 error 를 추가한다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String error
Output (Method)	-
Abstract Operation (Method)	*errorList.add(error*

Exceptional Courses of Events	-

Type	Method
Name	getSingleNum()
Purpose	SingleList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*singleList.size()*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getErrorNum()
Purpose	ErrorList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*errorList.size()*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceNum()
Purpose	ChoiceList 의 크기를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int size
Abstract Operation (Method)	*choiceList.size()*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getSingleList()
Purpose	SingleList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArryList<String> singleList
Abstract Operation (Method)	*singleList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getErrorList()
Purpose	ErrorList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> errorList
Abstract Operation (Method)	*errorList* 를 반환한다.
Exceptional Courses of Events	-

<model – Choice>

Type	Class
Name	Choice
Purpose	Choice 의 정보를 모아두는 클래스
Overview (Class)	choiceName,cosNum,consList,hasProp,hasSingle,hasError,hasIf,hasElse,propList,ifList,elseList 을 가지고 있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Exceptional Courses of Events	-

Type	Method
Name	Choice()
Purpose	Choice 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String choiceName
Output (Method)	-
Abstract Operation (Method)	Choice 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getChoiceName()
Purpose	Choice 의 이름을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	String choiceName
Abstract Operation (Method)	*ChoiceName* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addCons()
Purpose	ConsList 에 Constraints 객체를 추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	Constraints cons
Output (Method)	-
Abstract Operation (Method)	*consList.add(cons)* *consNum++*
Exceptional Courses of Events	-

Type	Method
Name	addIf()
Purpose	IfList 에 ifString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String ifString
Output (Method)	-
Abstract Operation (Method)	[Choice 객체가 Error Constraints 를 가지고 있지 않거나 Single Constraints 를 가지고 있지 않는 경우] *ifList.add(ifString)* *hasIf* 에 *true* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	addElse()
Purpose	elseList 에 elseString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String elseString
Output (Method)	-
Abstract Operation (Method)	*elseList.add(elseString)* *hasElse* 에 *true* 를 저장한다.

Exceptional Courses of Events	-

Type	Method
Name	addProp()
Purpose	propList 에 propString 을추가한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String propString
Output (Method)	-
Abstract Operation (Method)	[Choice 객체가 Error Constraints 를 가지고 있지 않거나 Single Constraints 를 가지고 있지 않는 경우] *propList.add(propString)* *hasProp* 에 *true* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsList()
Purpose	consList 를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Array<Constraints> consList
Abstract Operation (Method)	*consList* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isProp()
Purpose	hasProp 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasProp

Abstract Operation (Method)	hasProp 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isSingle()
Purpose	hasSingle 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasSingle
Abstract Operation (Method)	hasSingle 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isError()
Purpose	hasError 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasError
Abstract Operation (Method)	hasError 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsProp()
Purpose	hasProp 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasProp

Abstract Operation (Method)	hasProp 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsSingle()
Purpose	hasSingle 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasSingle
Abstract Operation (Method)	hasSingle 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsError()
Purpose	hasError 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasError
Abstract Operation (Method)	hasError 을 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	setIsElse()
Purpose	hasElse 을 저장한다..
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasElse

Abstract Operation (Method)	hasElse 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getPropList()
Purpose	propList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> propList
Abstract Operation (Method)	PropList 를 ArrayList<String> 형태로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getifList()
Purpose	ifList 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList<String> ifList
Abstract Operation (Method)	ifList 을 ArrayList<String> 형태로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	isIf()
Purpose	hasIf 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasIf
Abstract Operation (Method)	hasIf 을 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	isElse()
Purpose	hasElse 을 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	Boolean hasElse
Abstract Operation (Method)	hasElse 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getElseList()
Purpose	elseList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	ArrayList elseList
Abstract Operation (Method)	*elseList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsNum()
Purpose	Constraints 의 개수를 가져온다
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	-
Output (Method)	int consNum
Abstract Operation (Method)	*consNum* 을 반환한다.
Exceptional Courses of Events	-

<model - Constraints>

Type	Class
Name	Constraints
Purpose	Constraints 의 정보를 모아두는 클래스
Overview (Class)	consName 을 가지고 있다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Exceptional Courses of Events	N/A

Type	Method
Name	Constraints()
Purpose	Constraints 의 객체를 생성하는 함수
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3 Functional Requirements : R.5.1, R.5.2, R.5.3
Input (Method)	String consName
Output (Method)	-
Abstract Operation (Method)	Constraints 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getConsName()
Purpose	ConsName 을 반환하는 메소드.
Overview (Class)	ConsName 을 반환한다.
Cross Reference	
Input (Method)	N/A
Output (Method)	String
Abstract Operation (Method)	ConName 을 반환한다.
Exceptional Courses of Events	N/A

<model – Opinion>

Type	Class
Name	Opinion
Purpose	Opinion 정보를 모아두는 클래스
Overview (Class)	OpinionNum, opinionID, date, content 을 가지고 있다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Exceptional Courses of Events	-

Type	Method
Name	Opinion()
Purpose	Opinion 의 객체를 생성하는 함수
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String content
Output (Method)	-
Abstract Operation (Method)	Opinion 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	getOpinionID()
Purpose	OpinionID 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String opinionID
Abstract Operation (Method)	*OpinionID* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setOpinionID()
Purpose	OpinionID 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String opinionID
Output (Method)	-
Abstract Operation (Method)	*opinionID 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	getDate()
Purpose	date 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String date
Abstract Operation (Method)	*date* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setDate()
Purpose	date 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String date
Output (Method)	void
Abstract Operation (Method)	*date 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	getContent()
Purpose	content 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	String content
Abstract Operation (Method)	*content* 을 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setContent()
Purpose	content 를 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	String content
Output (Method)	-
Abstract Operation (Method)	*content 를 저장한다.*
Exceptional Courses of Events	-

Type	Method
Name	setOpinionNum()
Purpose	OpinionNum 을 저장한다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	Int opinionNum
Output (Method)	void
Abstract Operation (Method)	*opinionNum 를 저장한다.*
Exceptional Courses of Events	-

Type	Method

Name	getOpinionNum()
Purpose	opinionNum 를 가져온다.
Cross Reference	UseCase : R.8, R.9 Functional Requirements : R.8, R.9
Input (Method)	-
Output (Method)	Int opinionNum
Abstract Operation (Method)	*opinionNum* 을 반환한다.
Exceptional Courses of Events	-

<controller - MainServer>

Type	Class
Name	MainServer
Purpose	Server 의 수행을 제어하는 클래스
Overview (Class)	초기화, saveObject, Exit 한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	--

Type	Method
Name	MainServer()
Purpose	MainServer 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	Client 와 연결을 시도한다.
Exceptional Courses of Events	-

Type	Method
Name	init()

OOPT Stage 2050 <Construct>

Purpose	Server 를 초기화한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	boolean isConnected
Output (Method)	boolean isConnected
Abstract Operation (Method)	Client 와 연결을 시도하고 *isConnected* 를 반환한다.
Exceptional Courses of Events	N/A

Type	Method
Name	SaveObject()
Purpose	데이터를 저장한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	*userList* 와 *projectList* 를 "*object.dat*"로 저장한다. .
Exceptional Courses of Events	N/A

Type	Method
Name	ReadObject()
Purpose	데이터를 읽어온다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	"*object.dat*"를 읽어와서 각각에 해당하는 정보를 *userList* 와 *projectList* 에 저장한다.
Exceptional Courses of Events	N/A

Type	Method
Name	exit()
Purpose	종료하는 메소드
Cross Reference	UseCase : R.10 Functional Requirements : R.10
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	종료한다.
Exceptional Courses of Events	-

<controller - Server>

Type	Class
Name	Server
Purpose	Client 와 통신하는 클래스
Overview (Class)	Client 와 msg,file,obj 를 주고받는다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	Server()
Purpose	Server 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Socket sock-
Output (Method)	-
Abstract Operation (Method)	Server 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	sendACK()
Purpose	Client 로 msg 를 전송한다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4, R.5.1, R.6., R.7, R.8, R.9 Functional Requirements : R.1, R.2, R.3, R.4, R.5.1, R.6., R.7, R.8, R.9
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	PrintWriter 를 이용해 "*/ACK*"를 전송한다.
Exceptional Courses of Events	-

Type	Method
Name	sendREJ()
Purpose	Client 로 msg 를 전송한다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4, R.5.1, R.6., R.7, R.8, R.9 Functional Requirements : R.1, R.2, R.3, R.4, R.5.1, R.6., R.7, R.8, R.9
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	PrintWriter 를 이용해 "*/REJ*"를 전송한다.
Exceptional Courses of Events	-

Type	Method
Name	sendFile1()
Purpose	Client 로 object 를 전송한다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	Object object
Output (Method)	-
Abstract Operation (Method)	Client 에 Object object 를 encoding 해서 보낸다.
Exceptional Courses of Events	N/A

Type	Method
Name	rcvFile1()
Purpose	Client 로부터 object 를 받는다.
Cross Reference	UseCase : R.5.1, R.6, R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	-
Output (Method)	Object obj
Abstract Operation (Method)	Client 로부터 받아온 byte 를 decoding 해서 반환한다.
Exceptional Courses of Events	N/A

<controller - ServerThread>

Type	Class
Name	ServerThread
Purpose	Client 로부터 msg 를 수신하는 클래스
Overview (Class)	Client 로부터 msg 를 수신하고 분석한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	ServerThread()
Purpose	ServerThread 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	Socket sock, AdminProject adminP
Output (Method)	-
Abstract Operation (Method)	ServerThread 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	analyzeMSG()
Purpose	Client 로부터 받은 msg 를 분석한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	String msg
Output (Method)	-

OOPT Stage 2050 <Construct>

stract Operation (Method)	[msg == "/LGIN"] : Client 에서 받아온 ID 와 PW 를 매개변수로 실행한 AdminUser 의 login() 함수의 결과에 따라 Server 의 sendACK() 또는 sendREJ()를 호출한다. [msg == "/CHPW"] : Client 에서 받아온 PW 와 newPW 를 매개변수로 실행한 AdminUser 의 chgPW() 함수의 결과에 따라 Server 의 sendACK() 또는 sendREJ()를 호출한다. [msg == "/ADDP"] : AdminUser 의 chkAuth()를 호출하고 그 결과에 따라 AdminUser 의 addProject() 함수를 Client 에서 받아온 projectName 를 매개변수로 실행하고 Server 의 sendACK()를 호출하거나 sendREJ()를 호출한다. [msg == "/DELP"] : AdminUser 의 chkAuth()를 호출하고 그 결과에 따라 AdminUser 의 delProject() 함수를 Client 에서 받아온 projectName 를 매개변수로 실행하고 Server 의 sendACK()를 호출하거나 sendREJ()를 호출한다 [msg == "/ADDV"] : ID, version 과 Client 에서 받아온 projectNamed 을 매개변수로 실행한 AdminVer 의 addVer() 함수를 실행하고 AdminProject 의 setVerList()를 호출한 후 Server 의 sendACK()를 호출한다. [msg == "/DELV"] : AdminUser 의 getID()의 반환값인 ID 와 Client 에서 받아온 projectName 을 매개변수로 하는 AdminProject 의 getVerList()의 반환값인 tmpList 를 매개변수로 AdminVer 의 chkAuth 를 수행하고 그 결과에 따라 Server 의 sendACK()를 호출하거나 sendREJ()를 호출한다. [msg == "/ADDO"] : Client 에서 받아온 projectName,verNum,obj 를 매개변수로 AdminVer 의 addOpinion() 함수를 실행하고 AdminProject 의 setVerList()를 호출한 후 Server 의 sendACK()를 호출한다. [msg == "/DELO"] : AdminUser 의 getID()의 반환값인 ID 와 Client 에서 받아온 projectName 을 매개변수로 하는 AdminProject 의 getVerList()의 반환값인 tmpList 를 매개변수로 AdminVer 의 chkAuth 를 수행하고 그 결과에 따라 AdminVer 의 delOpinion()을 수행하고 Server 의 sendACK()를 호출하거나 sendREJ()를 호출한다. [msg == "/RCVP"] : AdminProject 의 sendProjectList()를 호출한다. [msg == "/RCVV"] : Client 에서 받아온 projectName 을 매개변수로 AdminProject 의 sendVerList()를 호출한다. [msg == "/RCVO"] : Client 에서 받아온 projectName 을 매개변수로 호출한 AdminProject 의 getVerList()의 매개변수인 OpinionList 로 AdminVersion 의 sendOpinionList()를 호출한다. [msg == "/DOWN"] : Client 에서 보내온 projectName 과 verNum 에 해당하는 객체를 서버에서 client 에 전송한다. [msg == "/EXIT" : 종료한다.

Exceptional Courses of Events	N/A

Type	Method
Name	run()
Purpose	Thread 를 돌리면서 Client 에서 msg 를 수신한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	N/A
Output (Method)	-
Abstract Operation (Method)	Server 에서 수신하는 msg 를 계속 분석한다.
Exceptional Courses of Events	N/A

<controller - AdminUser>

Type	Class
Name	AdminUser
Purpose	User 와 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	Clinet 와 통신하여 Login, ChangePW, chkAuth, chkPW, Logout 을 한다.
Cross Reference	UseCase : All Functional Requirements : All
Exceptional Courses of Events	-

Type	Method
Name	AdminUser()
Purpose	AdminUser 의 객체를 생성한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	AdminUser 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	login()
Purpose	Login 한다.
Cross Reference	UseCase : All Functional Requirements : All
Input (Method)	String ID, String PW
Output (Method)	Boolean status
Abstract Operation (Method)	*chkPW(flag,ID,PW)*를 수행하고 return 값을 *status* 에 저장한다. [status == true] *userList.put(ID,PW)* AdminUser 객체의 *ID* 와 *PW* 에 Input 값을 저장한다. *status* 를 반환한다. [status == false] *status* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	chgPW()
Purpose	Password 를 새로운 Password 로 변경한다.
Cross Reference	UseCase : R.0, R.1, R.2 Functional Requirements : R.0, R.1, R.2
Input (Method)	String PW, String newPW
Output (Method)	boolean status
Abstract Operation (Method)	*chkPW(flag,PW, newPW)*를 수행하고 return 값을 status 에 저장한다. [status == true] *userList.replace(ID,PW,newPW)* AdminUser 객체의 *PW* 에 *newPW* 를 저장한다. *status* 를 반환한다. [status == false] *status* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	ChkAuth()
Purpose	User 가 Admin 인지 확인한다.
Cross Reference	UseCase : R.1, R.3, R.4 Functional Requirements : R.1, R.3, R.4
Input (Method)	-
Output (Method)	boolean
Abstract Operation (Method)	AdminUser 객체의 ID 가 "ADMIN"이라면 *true* 를 반환한다. 아니라면 *false* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	chkPW()
Purpose	PW 가 동일하면 true, 다르면 false 를 반환하는 메소드
Cross Reference	UseCase : R.1, R.2 Functional Requirements : R.1, R.2
Input (Method)	boolean flag, String str1, String str2
Output (Method)	boolean
Abstract Operation (Method)	[flag == true] Login UserList 에서 User 의 Password 를 가져온다. [UserList 에 ID 가 존재하지 않을 경우] *false* 를 반환한다. [UserList 의 Password 와 일치하지 않을 경우] *false* 를 반환한다. [나머지 경우] *true* 를 반환한다. [flag == false] chgPW [UserList 의 Password 와 일치하지 않을 경우] *false* 를 반환한다. [나머지 경우] *true 를 반환한다.*
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	getID()
Purpose	ID 의 값을 가져온다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9 Functional Requirements : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9
Input (Method)	-
Output (Method)	String ID
Abstract Operation (Method)	*ID* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setID()
Purpose	ID 의 값을 저장한다.
Cross Reference	UseCase : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9 Functional Requirements : R.1, R.2, R.3, R.4, R.5.1, R.6, R.8,R.9
Input (Method)	String ID
Output (Method)	-
Abstract Operation (Method)	AdminUser 객체의 ID 에 *ID* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getPW()
Purpose	PW 의 값을 가져온다.
Cross Reference	UseCase : R.1, R.2 Functional Requirements : R.1, R.2
Input (Method)	-
Output (Method)	String PW
Abstract Operation (Method)	*PW*를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	SetPW()
Purpose	PW 의 값을 저장한다.
Cross Reference	UseCase : R.1, R.2 Functional Requirements : R.1, R.2
Input (Method)	String pW
Output (Method)	-
Abstract Operation (Method)	AdminUser 객체의 *PW*에 *pW*를 저장한다.
Exceptional Courses of Events	-

<controller - AdminProject>

Type	Class
Name	AdminProject
Purpose	Project 와 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	Clinet 와 통신하여 addProject, delProject, sendProjectList, sendVerList 를 한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Exceptional Courses of Events	-

Type	Method
Name	AdminProject()
Purpose	AdminProject 의 객체를 생성한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	-
Output (Method)	-
Abstract Operation (Method)	AdminProject 의 객체를 생성한다.
Exceptional Courses of Events	-

Type	Method
Name	addProject()
Purpose	projectList 에 project 를 추가하는 메소드
Overview (Class)	Input 으로 받은 projectName 을 projectList 에 추가한다.
Cross Reference	UseCase : R.3 Functional Requirements : R.3
Input (Method)	String projectName
Output (Method)	Hashmap projectList
Abstract Operation (Method)	*projectList* 에 *projectName* 을 추가한다. *projectList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	delProject
Purpose	projectList 에서 project 를 제거한다.
Cross Reference	UseCase : R.4 Functional Requirements : R.4
Input (Method)	String projectName
Output (Method)	-
Abstract Operation (Method)	*projectList* 에서 *projectName* 을 제거한다.
Exceptional Courses of Events	-

Type	Method
Name	sendProjectList()
Purpose	Client 에 ProjectList 를 전송한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	Server server
Output (Method)	-
Abstract Operation (Method)	Client 에 projectList 를 전송하기 위해 projectList 의 keySet 인 hashSet 을 생성한다.
Exceptional Courses of Events	-`

Type	Method
Name	sendVerList()
Purpose	Client 에 VersionList 를 전송한다.
Cross Reference	UseCase : R.5.1, R.6 , R.8, R.9 Functional Requirements : R.5.1, R.6, R.8, R.9
Input (Method)	Server server
Output (Method)	-
Abstract Operation (Method)	Client projectList 를 전송하기 위해 projectList 의 key 인 projectName 을 이용해서 VersionList 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	getProjectList()
Purpose	projectList 를 가져온다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	-
Output (Method)	HashMap projectList
Abstract Operation (Method)	*ProjectList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	setProjectList()
Purpose	projectList 를 저장한다.
Cross Reference	UseCase : R.3, R.4 Functional Requirements : R.3, R.4
Input (Method)	HashMap<String, LinkedList<Version> projectList
Output (Method)	-
Abstract Operation (Method)	AdminProject 객체의 *projectList* 에 *projectList* 를 저장한다.
Exceptional Courses of Events	-

Type	Method
Name	getVerList()
Purpose	VersionList 를 가져온다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6 , R.8, R.9 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.8, R.9
Input (Method)	String projectName
Output (Method)	LinkedList verList
Abstract Operation (Method)	projectList 의 key 값인 *projectName* 에 해당하는 *verList* 를 가져온다.
Exceptional Courses of Events	-

Type	Method
Name	setVerList()
Purpose	VerList 를 projectList 에 저장한다.
Cross Reference	UseCase : R.5.1, R.5.2, R.5.3, R.6 , R.8, R.9 Functional Requirements : R.5.1, R.5.2, R.5.3, R.6, R.8, R.9
Input (Method)	String projectName, LinkedList<Version> verList
Output (Method)	-
Abstract Operation (Method)	projectList 의 key 값인 *projectName* 에 해당하는 *verList* 에 *verList* 를 저장한다.
Exceptional Courses of Events	-

<controller – AdminVer>

Type	Class
Name	AdminVer
Purpose	Version, Opinion 과 관련된 모든 Event 를 담당하는 클래스
Overview (Class)	Client 와 통신하여 chkAuth, addVer, delVer, addOpinion, delOpinion, downloadFile, setVerList, sendOpinionList 를 한다.
Cross Reference	UseCase : R.5.1, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.6, R.7, R.8, R.9
Exceptional Courses of Events	N/A

Type	Method
Name	AdminVer()
Purpose	AdminVer 의 객체를 생성한다.
Cross Reference	UseCase : R.5.1, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.6, R.7, R.8, R.9
Input (Method)	Server server
Output (Method)	-
Abstract Operation (Method)	AdminVer 의 객체를 생성한다.
Exceptional Courses of Events	-

OOPT Stage 2050 <Construct>

Type	Method
Name	chkAuth()
Purpose	DelVer 과 delOpinion 을 위한 권한의 승인 여부를 알려주는 메소드
Cross Reference	UseCase : R.6, R.9 Functional Requirements : R.6, R.9
Input (Method)	Boolean flag, String ID, LinkedList tmpList, String str1, String str2
Output (Method)	Boolean status
Abstract Operation (Method)	DelVer 과 delOpinion 의 권한 승인 여부를 boolean 으로 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addVer()
Purpose	VerList 에 version 객체를 추가하는 메소드
Cross Reference	UseCase : R.5.1 Functional Requirements : R.5.1
Input (Method)	*String id, Version version, LinkedList<Version> verList*
Output (Method)	LinkedList tmpVerList
Abstract Operation (Method)	*tmpVerList* 에 *version* 을 추가한다 *opinionNum* 을 증가하고, input 으로 받은 *id* 를 저장한다. *tmpVerList* 를 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	delVer()
Purpose	VerList 에서 Ver 객체를 제거하는 메소드
Cross Reference	UseCase : R.6 Functional Requirements : R.6
Input (Method)	-
Output (Method)	LinkedList tmpVerList
Abstract Operation (Method)	*tmpVerList* 에서 *tmpVer* 을 제거하고 *tmpVerList* 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	addOpinion()
Purpose	OpinionList 에 Opinion 객체를 추가하는 메소드.
Cross Reference	UseCase : R.8 Functional Requirements : R.8
Input (Method)	String id, LinkedList verList, String verNum, Opinion opinion
Output (Method)	LinkedList tmpVerList
Abstract Operation (Method)	Input 값 Opinion 객체에 id 와 opinionNum 을 저장하고, VerList 내의 OpinionList 에 Opinion 객체를 추가하고 tmpVerList 를 반환한다.

Exceptional Courses of Events	-

Type	Method
Name	delOpinion()
Purpose	OpinionList 에서 Opinion 객체를 제거하는 메소드
Cross Reference	UseCase : R.9 Functional Requirements : R.9
Input (Method)	-
Output (Method)	LinkedList tmpVerList
Abstract Operation (Method)	OpinionList 의 마지막 리스트에서 Opinion 객체를 삭제하고 tmpVerList 를 반환한다.
Exceptional Courses of Events	-

Type	Method
Name	downloadFile()
Purpose	Input verNum 와 동일한 File 을 반환하는 메소드
Cross Reference	UseCase : R.7 Functional Requirements : R.7
Input (Method)	String verNum, LinkedList verList
Output (Method)	File file
Abstract Operation (Method)	Input 으로 받은 verList 에서 Input verNum 와 같은 file 을 받아와서 반환한다.

Exceptional Courses of Events	N/A

Type	Method
Name	setVerList()
Purpose	VerList 를 저장하는 메소드
Overview (Class)	Input 으로 받는 verList 를 저장한다.
Cross Reference	UseCase : R.5.1, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.6, R.7, R.8, R.9
Input (Method)	LinkedList verList
Output (Method)	-
Abstract Operation (Method)	input 으로 받는 verList 를 AdminVer 객체 verList 에 저장한다.
Exceptional Courses of Events	-

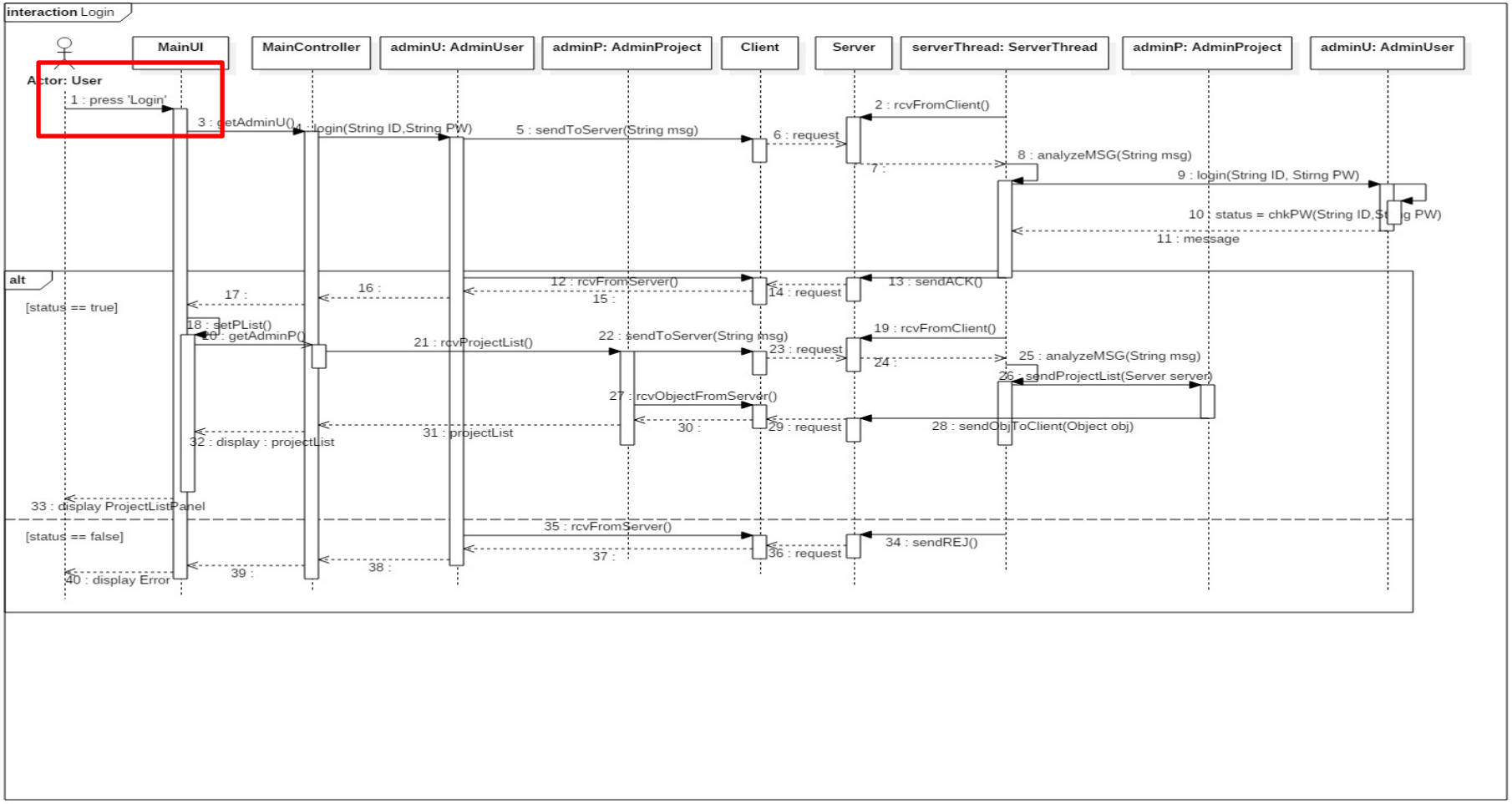
Type	Method
Name	sendOpinionList()
Purpose	OpinionList 를 클라이언트로 전송하는 메소드
Cross Reference	UseCase : R.5.1, R.6, R.7, R.8, R.9 Functional Requirements : R.5.1, R.6, R.7, R.8, R.9
Input (Method)	LinkedList verList, String verNum
Output (Method)	-

OOPT Stage 2050 <Construct>

Abstract Operation (Method)	OpinionList 를 서버에서 클라이언트로 전송한다.
Exceptional Courses of Events	-

Activity 2152. Implement Windows

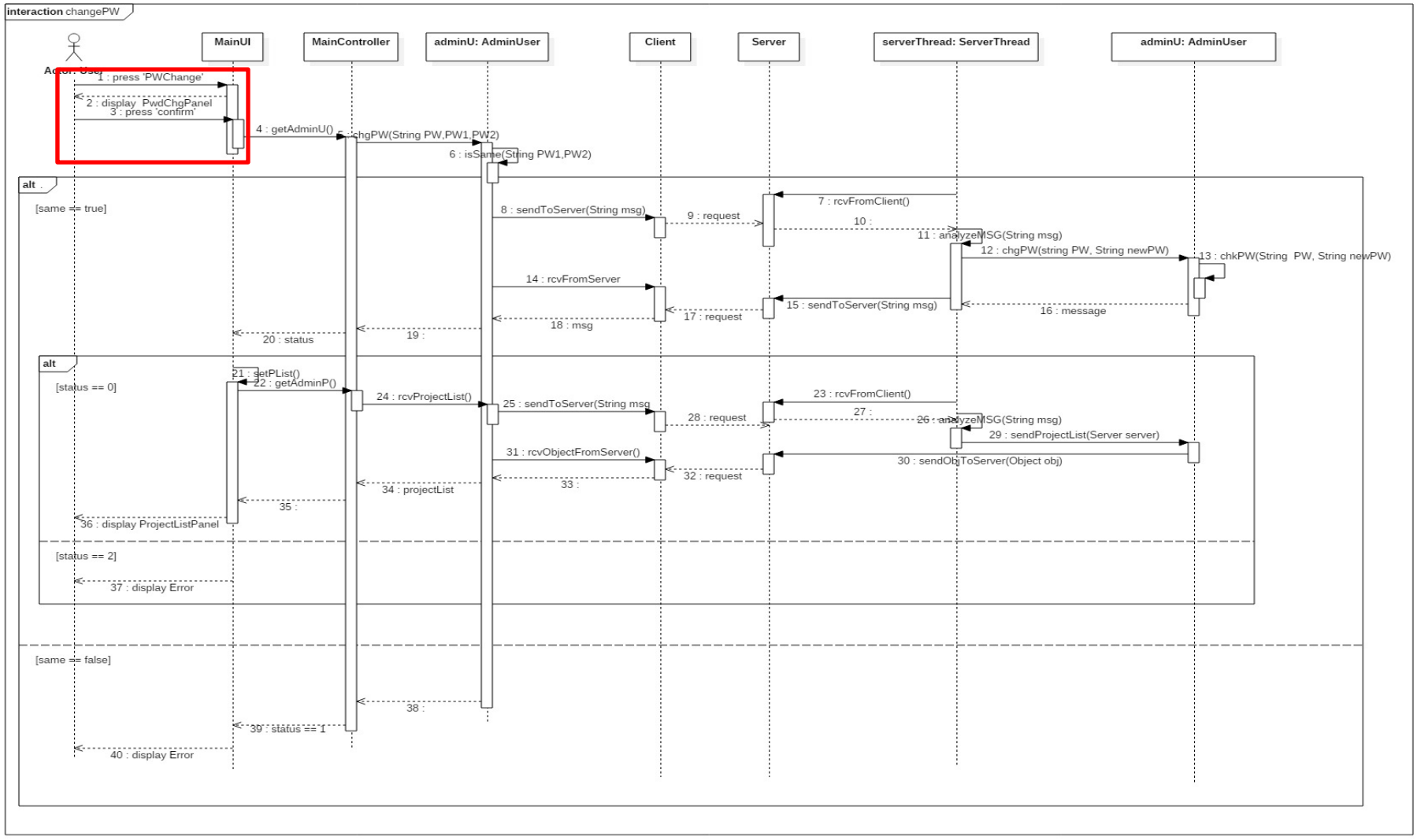
1. Login



OOPT Stage 2050 <Construct>

Name	Press 'Start'
Responsibilities	Window-1 GUI 의 시작버튼을 누른다
Type	GUI
Cross Reference	Login
Notes	GUI 의 시작버튼을 누른다
Pre-Conditions	올바른 ID 와 PW 를 입력한 상태
Post-Conditions	Window-2 로 이동한다.

2. ChangePW

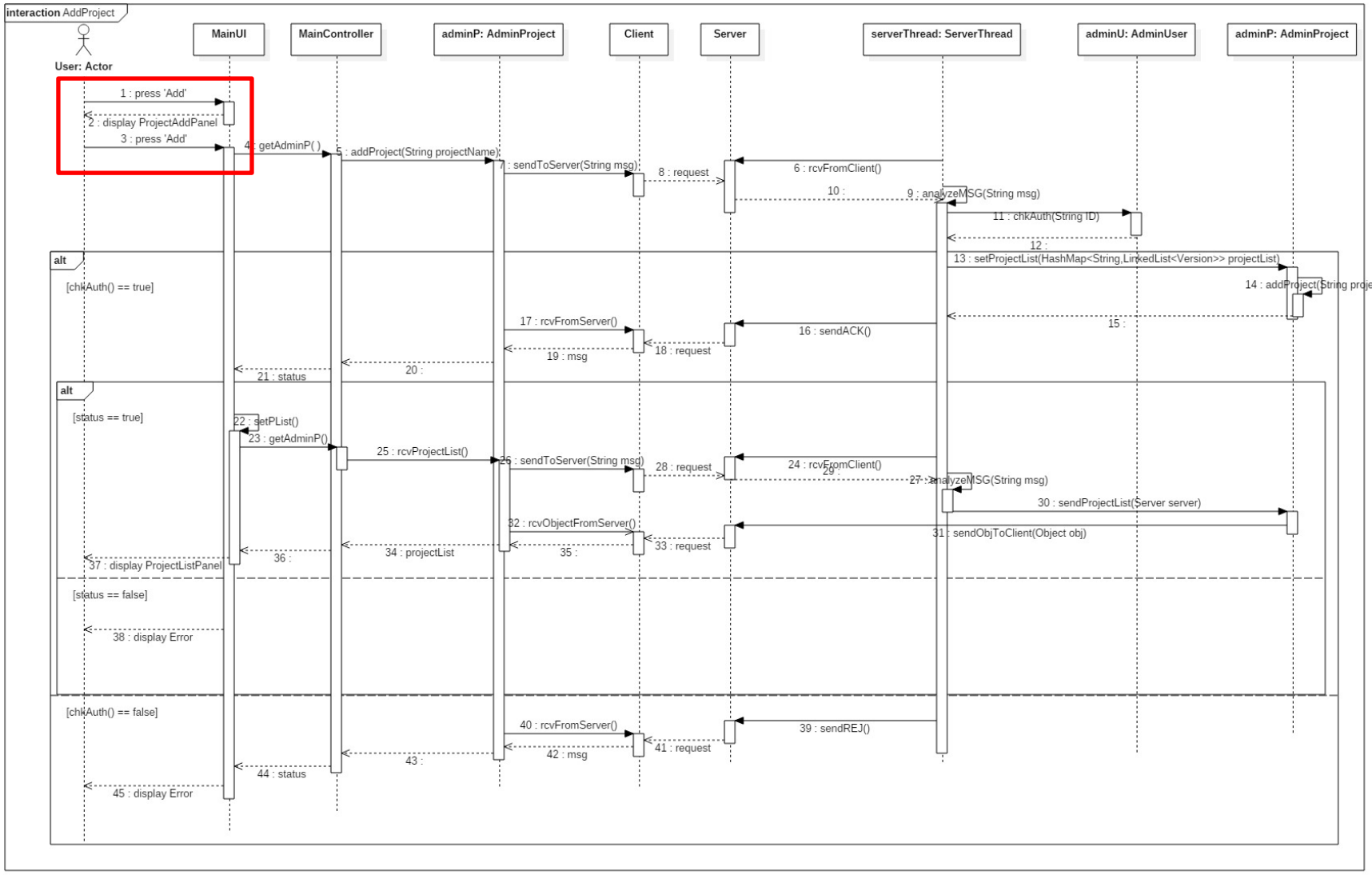

interaction changePW
MainUI
MainController
adminU: AdminUser
Client
Server
serverThread: ServerThread
adminU: AdminUser
1 : press 'PWChange'
2 : display PwdChgPanel
3 : press 'confirm'
4 : getAdminU()
5 : chgPW(String PW,PW1,PW2)
6 : isSame(String PW1,PW2)
alt
[same == true]
8 : sendToServer(String msg)
9 : request
7 : rcvFromClient()
10 :
11 : analyzeMSG(String msg)
12 : chgPW(string PW, String newPW)
13 : chkPW(String PW, String newPW)
14 : rcvFromServer
15 : sendToServer(String msg)
16 : message
17 : request
18 : msg
19 :
20 : status
alt
[status == 0]
21 : getPList()
22 : getAdminP()
24 : rcvProjectList()
25 : sendToServer(String msg
23 : rcvFromClient()
27 :
26 : analyzeMSG(String msg)
28 : request
29 : sendProjectList(Server server)
30 : sendObjToServer(Object obj)
31 : rcvObjectFromServer()
32 : request
33 :
34 : projectList
35 :
36 : display ProjectListPanel
[status == 2]
37 : display Error
[same == false]
38 :
39 : status == 1
40 : display Error

OOPT Stage 2050 <Construct>

Name	Press 'pwdChange
Responsibilities	Window-2 GUI 의 비밀번호 변경 버튼을 누른다
Type	GUI
Cross Reference	ChangePW
Notes	Window-2 GUI 의 비밀번호 변경 버튼을 누른다
Pre-Conditions	Login 에 성공한 상태
Post-Conditions	Window-3 로 이동한다

Name	Press 'confirm
Responsibilities	Window-3 GUI 의 확인버튼을 누른다.
Type	GUI
Cross Reference	ChangePW
Notes	Window-3 GUI 의 확인버튼을 누른다.
Pre-Conditions	올바른 현재 비밀번호와 새 비밀번호와 새 비밀번호 확인을 입력한 상태.
Post-Conditions	User 의 정보를 변경 후 Window-2 로 돌아간다.

3. AddProject

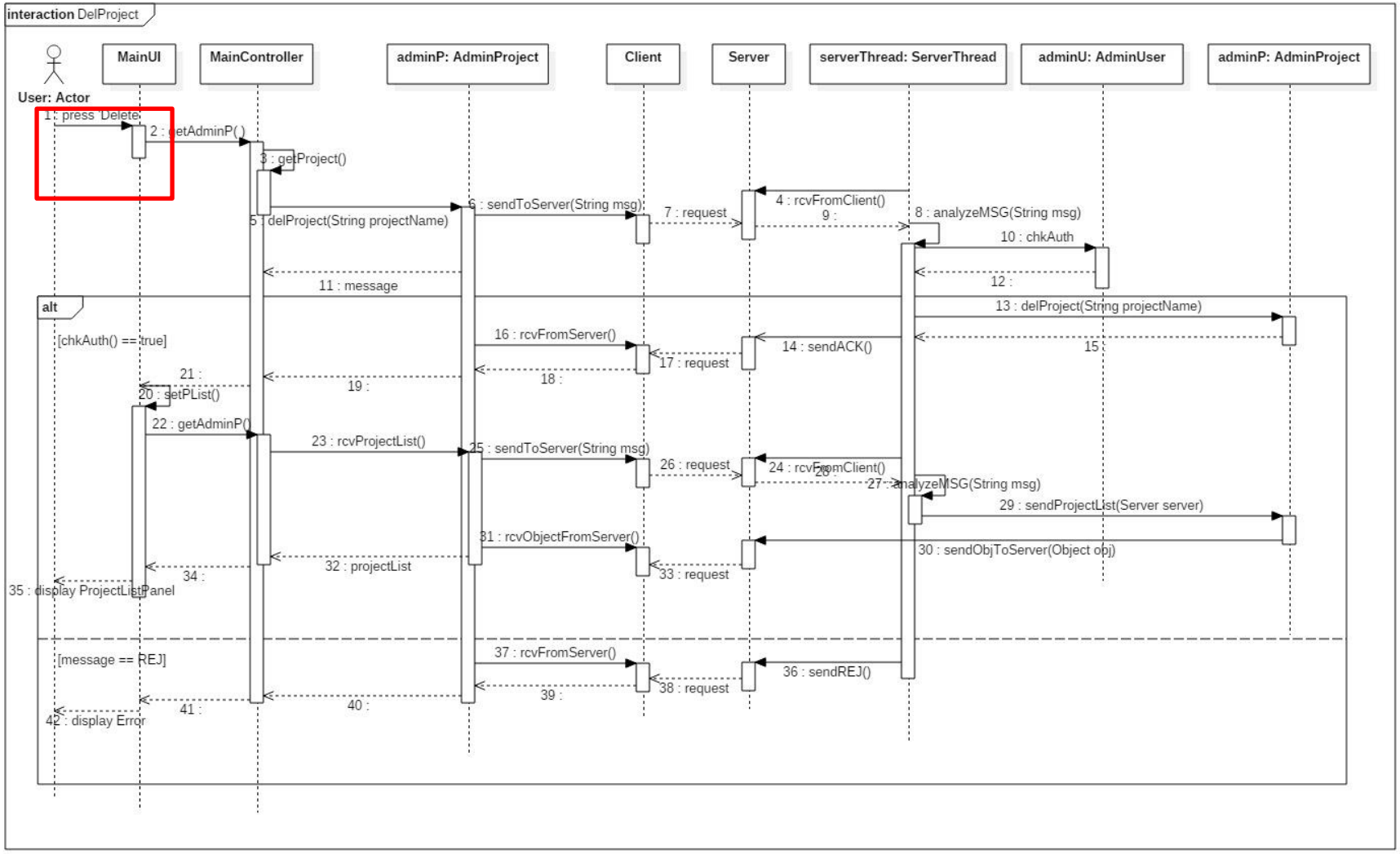


OOPT Stage 2050 <Construct>

Name	Press 'Add'
Responsibilities	Window-2 GUI 의 추가 버튼을 누른다.
Type	GUI
Cross Reference	AddProject
Notes	Window-2 GUI 의 추가 버튼을 누른다.
Pre-Conditions	관리자의 Login 이 성공한 상태
Post-Conditions	Window-4 로 이동한다

Name	Press 'Add'
Responsibilities	Window-4 GUI 의 추가 버튼을 누른다.
Type	GUI
Cross Reference	AddProject
Notes	indow-4 GUI 의 추가 버튼을 누른다.
Pre-Conditions	관리자의 Login 이 성공한 상태, 올바른 프로젝트 명을 입력한 상태
Post-Conditions	Project 를 추가하고 Window-2 로 이동한다

4. DelProject

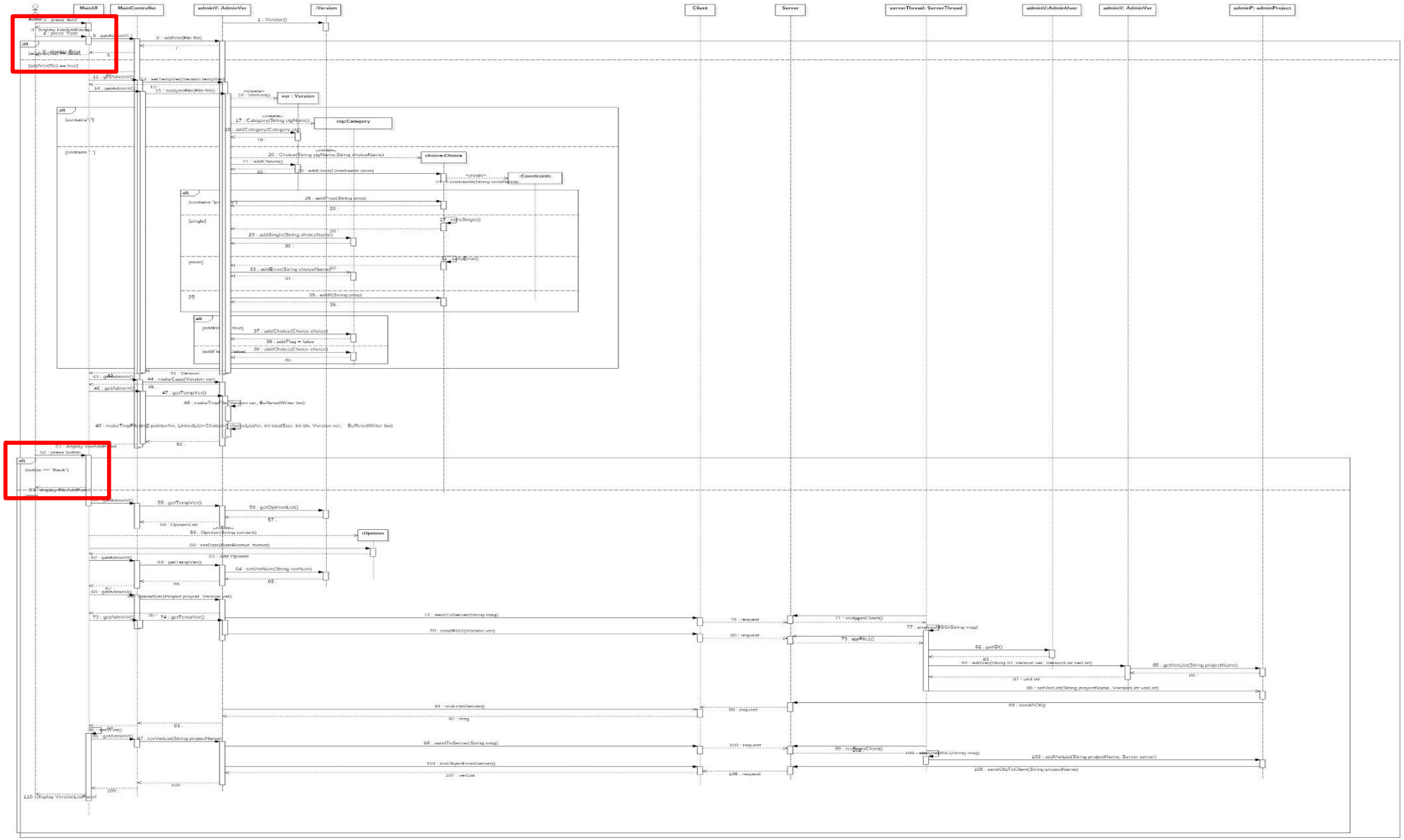


OOPT Stage 2050 <Construct>

Name	Press 'Delete'
Responsibilities	Window-2 GUI 의 삭제하고자 하는 project 의 삭제버튼을 누른다.
Type	GUI
Cross Reference	DelProject
Notes	Window-2 GUI 의 삭제하고자 하는 project 의 삭제버튼을 누른다.
Pre-Conditions	관리자의 Login 이 성공한 상태
Post-Conditions	Project 를 삭제하고 정보를 반영해서 Window-2 를 재출력한다.

OOPT Stage 2050 <Construct>

5. AddVer



OOPT Stage 2050 <Construct>

Name	Press "Add"
Responsibilities	Window-5 GUI 에서 추가 버튼을 누른다.
Type	GUI
Cross Reference	AddVer
Notes	Window-5 GUI 에서 추가 버튼을 누른다.
Pre-Conditions	Login 을 성공한 상태
Post-Conditions	Window-7 로 이동한다.

Name	Press "Add"
Responsibilities	Window-7 GUI 에서 추가 버튼을 누른다.
Type	GUI
Cross Reference	AddVer
Notes	Window-7 GUI 에서 추가 버튼을 누른다.
Pre-Conditions	Login 을 성공한 상태
Post-Conditions	File dialog 를 출력하고 분석할 file 을 선택한다.

Name	Press "Test"
Responsibilities	Window-7 GUI 에서 테스트 버튼을 누른다.
Type	GUI
Cross Reference	AddVer

OOPT Stage 2050 <Construct>

Notes	Window-7 GUI 에서 테스트 버튼을 누른다.
Pre-Conditions	분석할 file 을 성공적으로 추가한 상태
Post-Conditions	File 을 사용할 수 있는 정보로 가공하고 Test Case 를 성공적으로 조합한 후 Window-8 로 이동한다

Name	Press "Back"
Responsibilities	Window-8 GUI 에서 뒤로 버튼을 누른다.
Type	GUI
Cross Reference	AddVer
Notes	Window-8 GUI 에서 뒤로 버튼을 누른다.
Pre-Conditions	분석이 끝나고 성공적으로 Window-8 로 이동한 상태
Post-Conditions	Window-7 로 이동한다.

Name	Press "Lower"
Responsibilities	Window-8 GUI 에서 하위 Version 추가 버튼을 누른다.
Type	GUI
Cross Reference	AddVer
Notes	Window-8 GUI 에서 하위 Version 추가 버튼을 누른다.
Pre-Conditions	분석이 끝나고 성공적으로 Window-8 로 이동한 상태
Post-Conditions	하위 Version 을 추가하고 Window-5 로 이동한다.

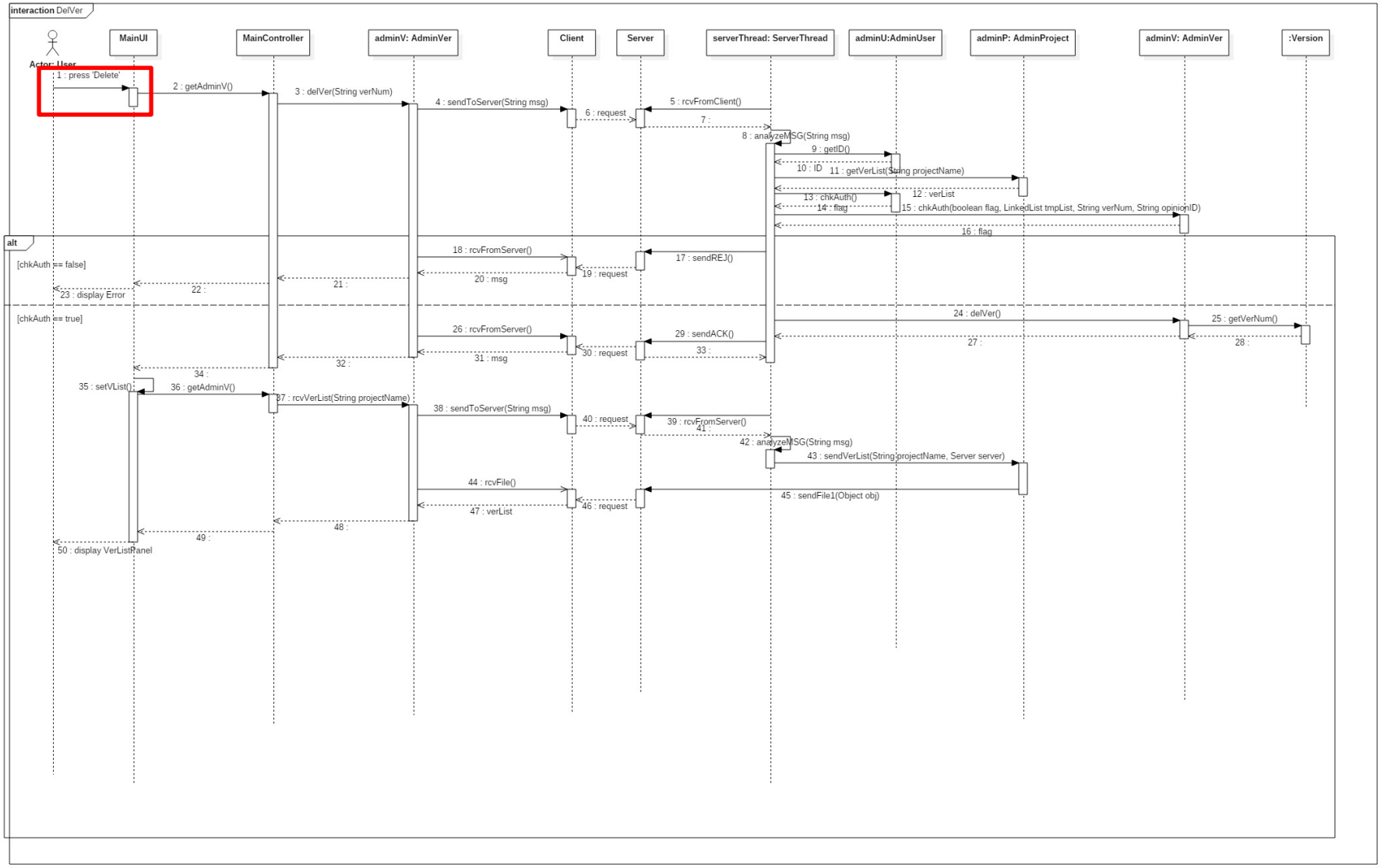
Name	Press "Higher"
Responsibilities	Window-8 GUI 에서 상위 Version 추가 버튼을 누른다.

OOPT Stage 2050 <Construct>

Type	GUI
Cross Reference	AddVer
Notes	Window-8 GUI 에서 상위 Version 추가 버튼을 누른다.
Pre-Conditions	분석이 끝나고 성공적으로 Window-8 로 이동한 상태
Post-Conditions	하위 Version 을 추가하고 Window-5 로 이동한다.

OOPT Stage 2050 <Construct>

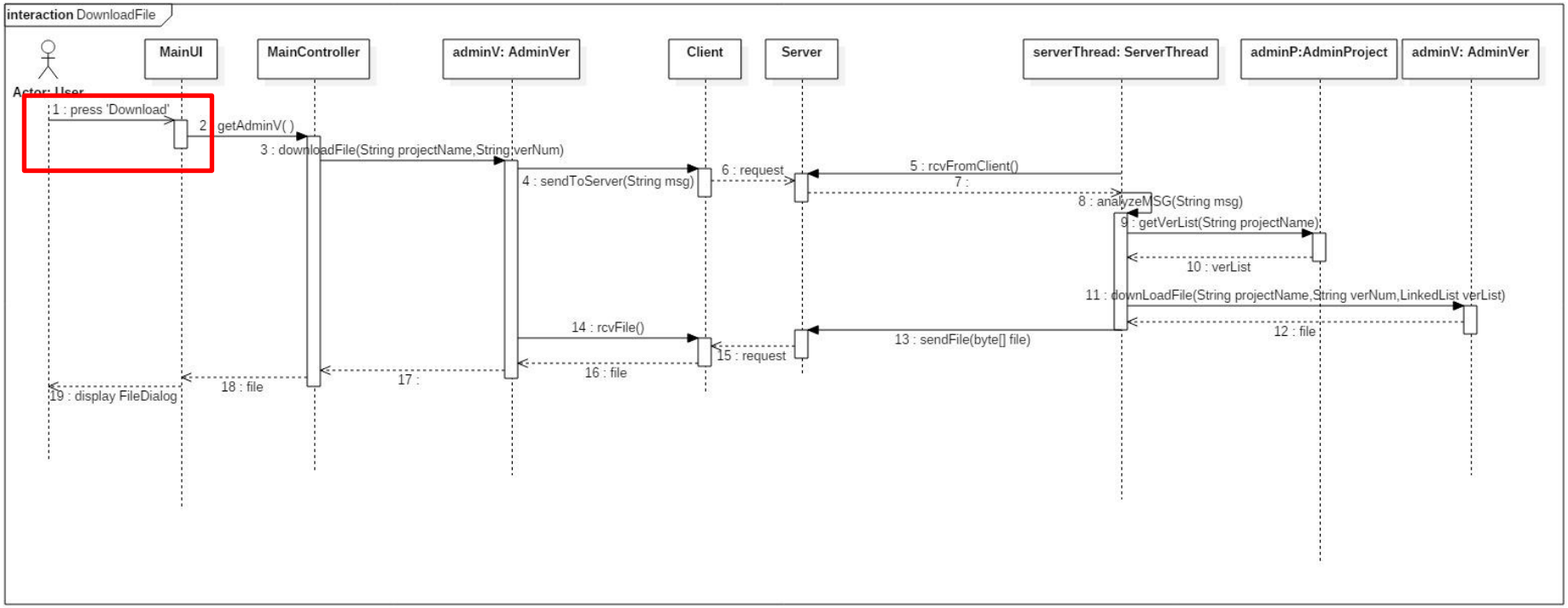
6. DelVer



OOPT Stage 2050 <Construct>

Name	Press 'Delete'
Responsibilities	Window-5 GUI 의 삭제하고자 하는 Version 의 삭제버튼을 누른다.
Type	GUI
Cross Reference	DelVer
Notes	Window-5 GUI 의 삭제하고자 하는 Version 의 삭제버튼을 누른다.
Pre-Conditions	Version 을 추가한 User 가 Login 에 성공한 상태
Post-Conditions	Version 을 삭제하고 정보를 반영해서 Window-5 를 재출력한다.

7. DownloadFile

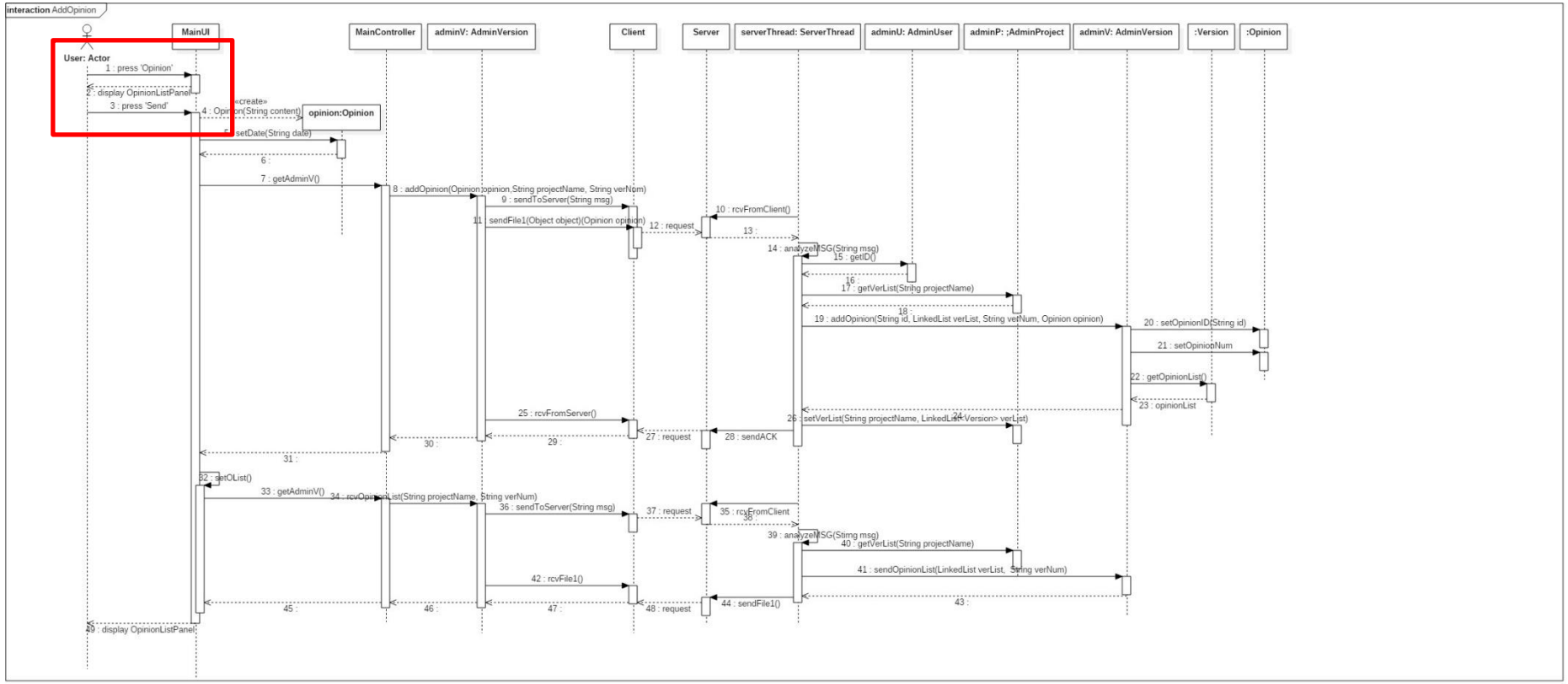


OOPT Stage 2050 <Construct>

Name	Press 'Dowload'
Responsibilities	Window-5 GUI 의 내려 받고자 하는 Version 의 다운로드 버튼을 누른다.
Type	GUI
Cross Reference	DownloadFile
Notes	Window-5 GUI 의 내려 받고자 하는 Version 의 다운로드 버튼을 누른다.
Pre-Conditions	Version List 에 Version 이 존재하는 상태
Post-Conditions	File dialog 를 출력하고 선택한 경로에 해당하는 Version 의 File 을 내려 받는다.

OOPT Stage 2050 <Construct>

8. AddOpinion



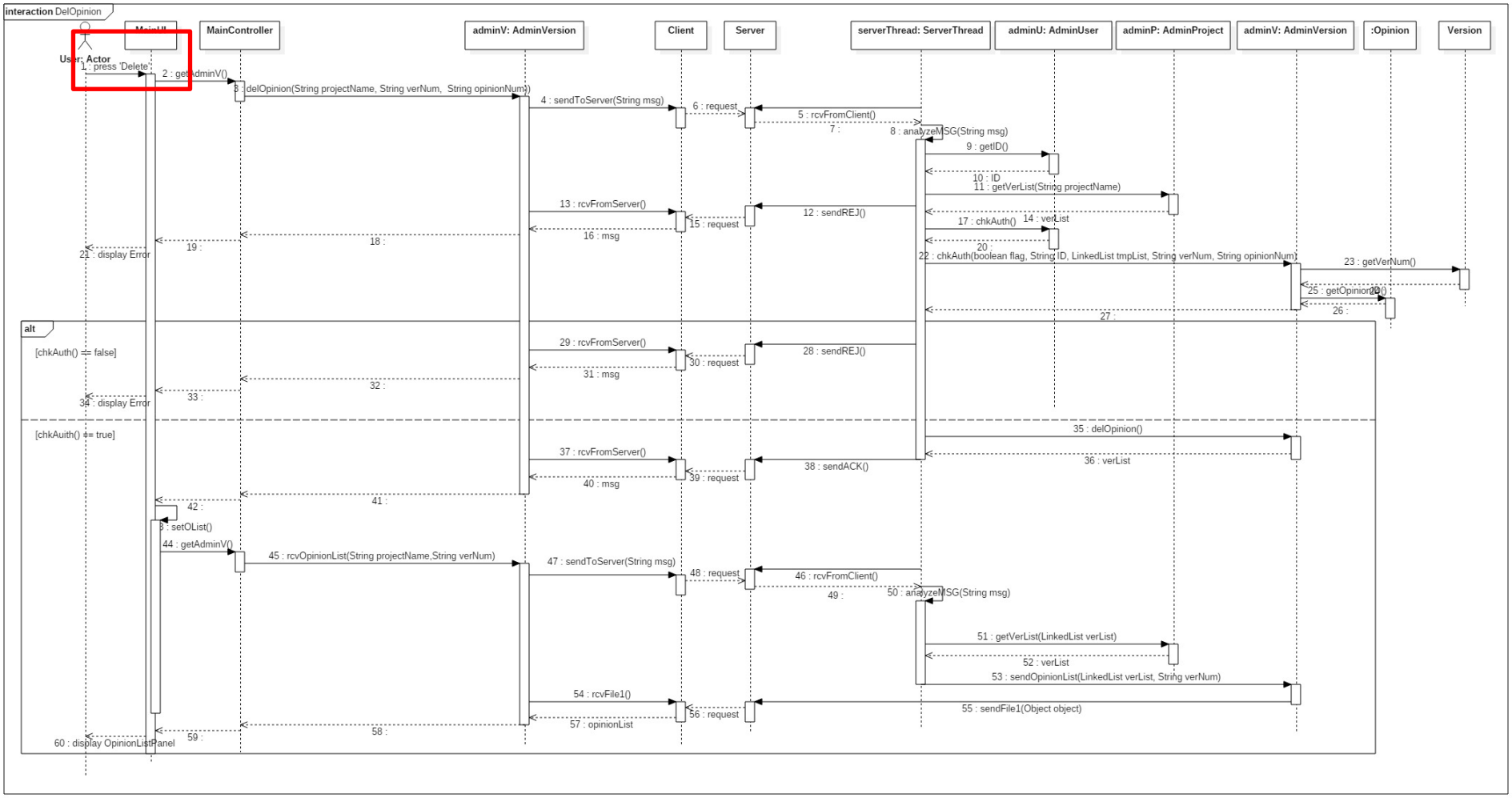
OOPT Stage 2050 <Construct>

Name	Press 'Opinion
Responsibilities	Window - 5 GUI 의 Opinion 을 작성할 Version 작성자의 프로필 버튼을 누른다.
Type	GUI
Cross Reference	AddOpinion
Notes	Window - 5 GUI 의 Opinion 을 작성할 Version 작성자의 프로필 버튼을 누른다.
Pre-Conditions	존재하는 Version 을 선택한 상태
Post-Conditions	Window-6 로 이동한다.

Name	Press 'Send'
Responsibilities	Window -6 GUI 의 Opinion 내용을 작성하고 전송 버튼을 누른다.
Type	GUI
Cross Reference	AddOpinion
Notes	Window -6 GUI 의 Opinion 내용을 작성하고 전송 버튼을 누른다.
Pre-Conditions	Opinion 의 내용을 작성한 상태
Post-Conditions	작성한 내용의 Opinion 을 추가하고 Window-6 를 재출력한다,.

OOPT Stage 2050 <Construct>

9. DelOpinion

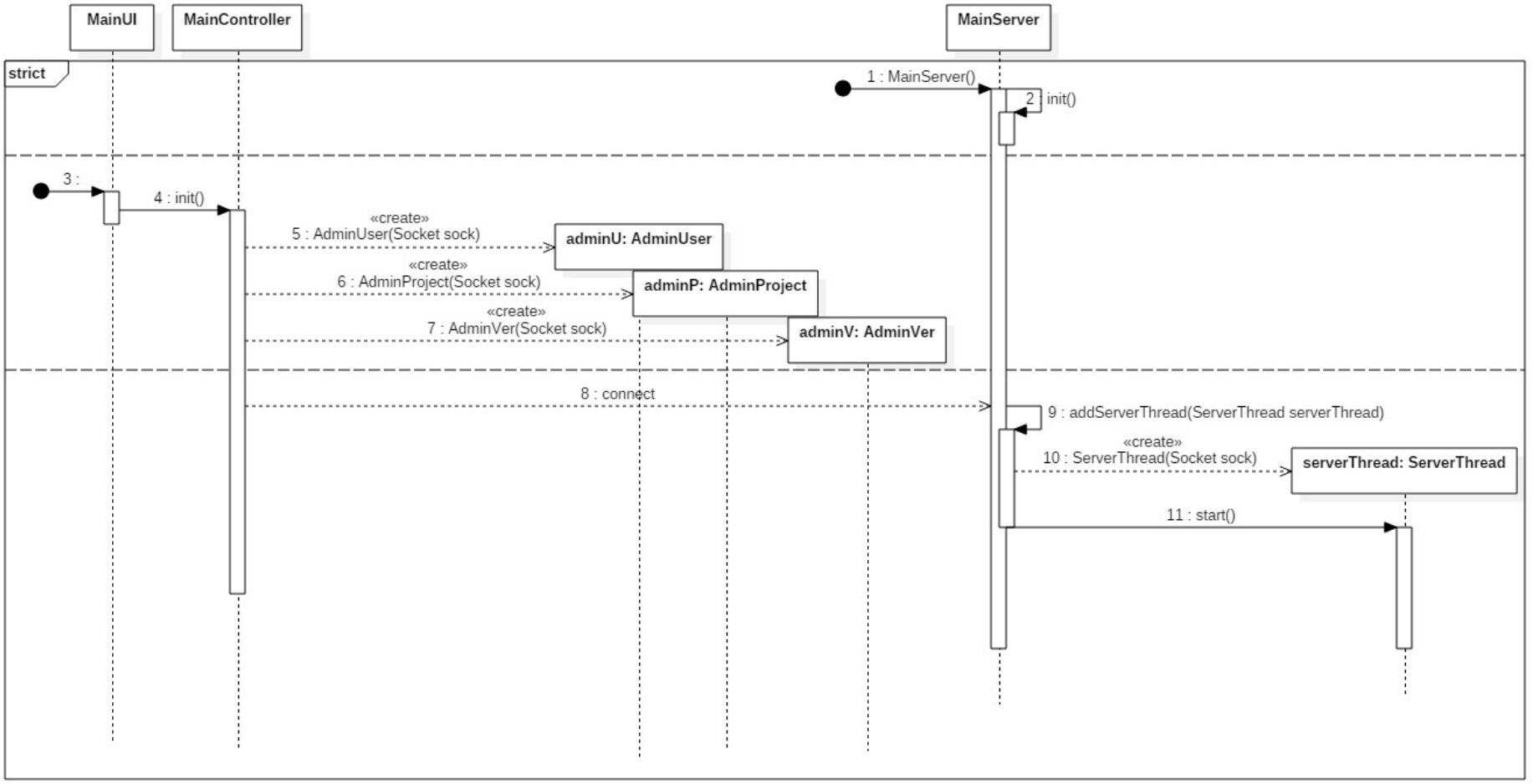


OOPT Stage 2050 <Construct>

Name	Press 'Delete'
Responsibilities	Window-6 GUI 의 삭제하고자 하는 Opinoin 의 삭제버튼을 누른다.
Type	GUI
Cross Reference	DelOpinion
Notes	삭제하고자 하는 Opinon 의 삭제버튼을 누른다.
Pre-Conditions	Opinion List 에 Opinion 이 존재하는 상태 Opinion 을 추가한 User 가 Login 에 성공한 상태
Post-Conditions	Opinion 을 삭제하고 정보를 반영해서 Window-6 을 재출력한다.

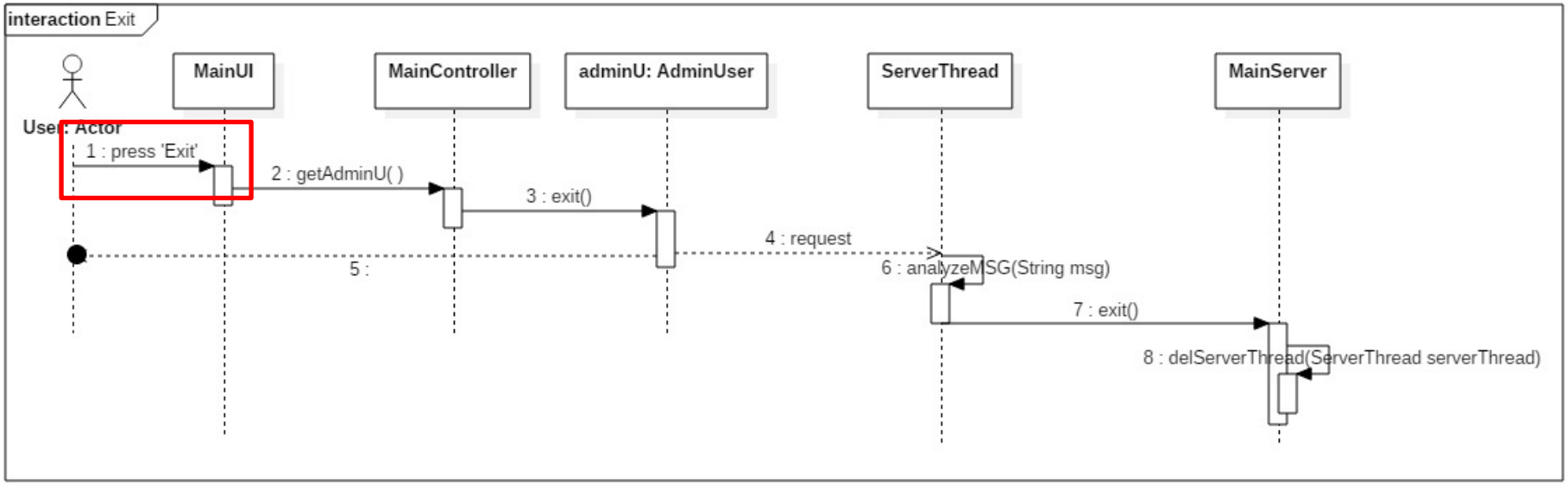
OOPT Stage 2050 <Construct>

10. Init



OOPT Stage 2050 <Construct>

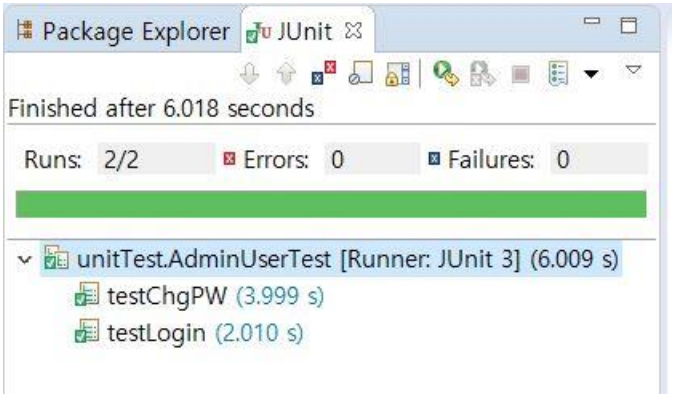
11. Exit



Name	Press 'Exit'
Responsibilities	Window-1 GUI 의 종료 버튼을 누른다.
Type	GUI
Cross Reference	Exit
Notes	Window-1 GUI 의 종료 버튼을 누른다.
Pre-Conditions	N/A
Post-Conditions	프로그램을 종료한다.

Activity 2155. Write Unit Test Code

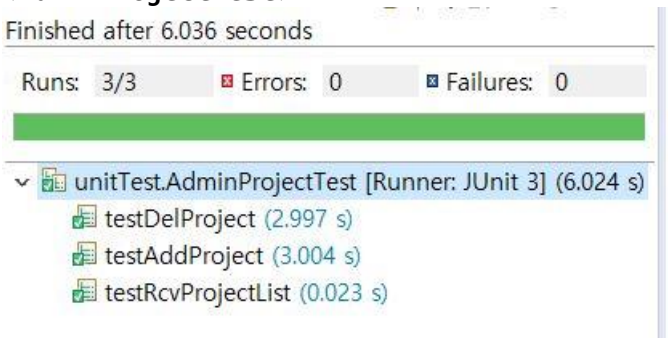
<AdminUserTest>




```
@Override
	protected void setUp() throws Exception {
		mc.init();
	}

public void testLogin() {

		final String ID_1 = "201411317";
		final String PW_1 = "1013";
		assertEquals(true, mc.getAdminU().login(ID_1, PW_1));

		final String ID_2 = "201411317";
		final String PW_2 = "0000";
		assertEquals(false, mc.getAdminU().login(ID_2, PW_2));

		final String ID_3 = "201311304";
		final String PW_3 = "0000";
		assertEquals(false, mc.getAdminU().login(ID_3, PW_3));
	}

	public void testChgPW() {

		mc.getAdminU().login("201411317", "1013");
		final String PW_1 = "1013";
		final String PW1_1 = "1012";
		final String PW2_1 = "1012";
		assertEquals(0, mc.getAdminU().chgPW(PW_1, PW1_1, PW2_1));

		mc.getAdminU().login("201411317", "1012");
		final String PW_2 = "1012";
		final String PW1_2 = "1111";
		final String PW2_2 = "0000";
		assertEquals(1, mc.getAdminU().chgPW(PW_2, PW1_2, PW2_2));

		mc.getAdminU().login("201411317", "1012");
		final String PW_3 = "0000";
```

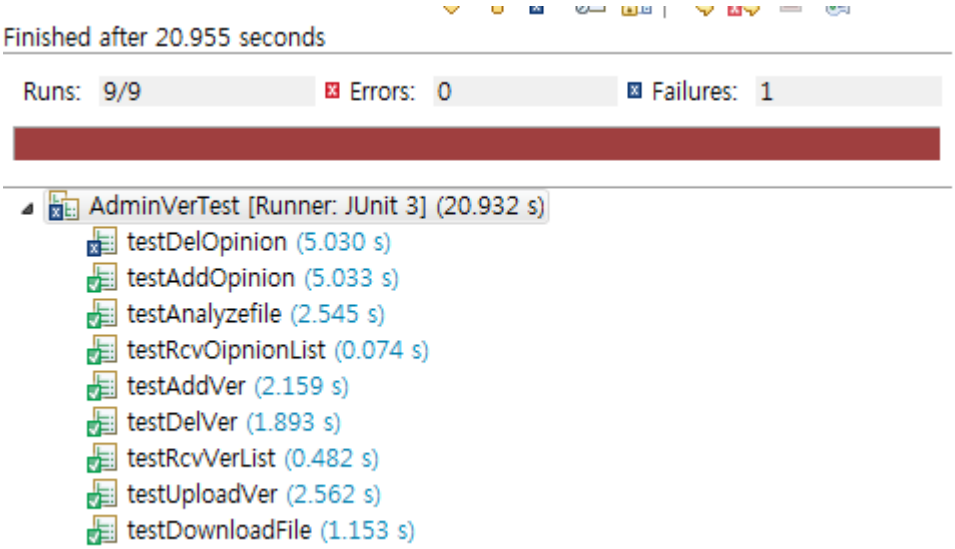
```
            final String PW1_3 = "1111";
            final String PW2_3 = "1111";
            assertEquals(2, mc.getAdminU().chgPW(PW_3, PW1_3, PW2_3));
      }
```

<AdminProjectTest>




```
      @Override
      protected void setUp() throws Exception {
            mc.init();
      }

      public void testAddProject() {

            final String projectName_1 = "Test1";
            mc.getAdminU().login("ADMIN", "0000");
            assertEquals(true, mc.getAdminP().addProject(projectName_1));

            final String projectName_2 = "Test2";
            mc.getAdminU().login("201411317", "1013");
            assertEquals(false,
mc.getAdminP().addProject(projectName_2));

      }

      public void testDelProject() {

            final String projectName_1 = "Test1";
            mc.getAdminU().login("ADMIN", "0000");
            assertEquals(true, mc.getAdminP().delProject(projectName_1));

            final String projectName_2 = "Test2";
            mc.getAdminU().login("201411317", "1013");
            assertEquals(false,
mc.getAdminP().addProject(projectName_2));

      }

      public void testRcvProjectList() {
            assertNotNull(mc.getAdminP().rcvProjectList());

      }

}
```

OOPT Stage 2050 <Construct>
<AdminVertest>



delOpinion fail..

```
public void testAddVer() {

		mc.getAdminU().login("ADMIN", "0000");
		File file_1 = new File("test.txt");
		assertEquals(true, mc.getAdminV().addVer(file_1));


		mc.getAdminU().login("ADMIN", "0000");
		File file_2 = new File("test2.png");
		assertEquals(false, mc.getAdminV().addVer(file_2));

		mc.getAdminU().login("ADMIN", "0000");
		File file_3 = new File("test3.png");
		assertEquals(false, mc.getAdminV().addVer(file_3));

	}

	public void testAnalyzefile() throws IOException {

		mc.getAdminU().login("ADMIN", "0000");
		String fileName = "test.txt";
		assertNotNull(mc.getAdminV().analyzeFile(fileName));
	}


	public void testUploadVer() {

		mc.getAdminU().login("ADMIN", "0000");
		String projectName_1 = "Test2";
		File file_1 = new File("test.txt");
		Version ver_1 = new Version("3.0", file_1);
		assertEquals(true, mc.getAdminV().uploadVer(projectName_1,
ver_1));
```

```
            mc.getAdminU().login("ADMIN", "0000");
            String projectName_2 = "Test4";
            File file_2 = new File("test.txt");
            Version ver_2 = new Version("1.0", file_2);
            assertEquals(false, mc.getAdminV().uploadVer(projectName_2,
ver_2));


            mc.getAdminU().login("ADMIN", "0000");
            String projectName_3 = "Test1";
            File file_3 = new File("test.txt");
            Version ver_3 = new Version("5.0", file_3);
            assertEquals(false, mc.getAdminV().uploadVer(projectName_3,
ver_3));
      }

      public void testDelVer() {

            mc.getAdminU().login("ADMIN", "0000");
            String projectName_1 = "ABC";
            String verNum_1 = "1.0";

            mc.getAdminP().addProject("ABC");
            mc.getAdminV().addVer(new File("test.txt"));
            assertEquals(true, mc.getAdminV().delVer(projectName_1,
verNum_1));


            mc.getAdminU().login("201411278", "1013");
            String projectName_2 = "Test1";
            String verNum_2 = "1.0";
            assertEquals(false, mc.getAdminV().delVer(projectName_2,
verNum_2));
      }

      public void testAddOpinion() {

            mc.getAdminU().login("ADMIN", "0000");
            String content = "갸갸갸갸갹";
            Opinion opinion = new Opinion(content);
            String projectName = "Test1";
            String verNum = "1.0";
            assertEquals(true, mc.getAdminV().addOpinion(opinion,
projectName, verNum));

      }

      public void testDelOpinion() {

            mc.getAdminU().login("ADMIN", "0000");
            String projectName_1 = "ABC";
            String verNum_1 = "1.0";
            String opinionNum_1 = "1";
```

```
            mc.getAdminP().addProject("ABC");

            mc.getAdminV().addVer(new File("text.txt"));
            mc.getAdminV().addOpinion(new Opinion("abcde"),
projectName_1,  verNum_1);

            assertEquals(true, mc.getAdminV().delOpinion(projectName_1,
verNum_1, opinionNum_1));


            System.out.println("==============");
            mc.getAdminU().login("201411278", "1111");
            String projectName = "Test1";
            String verNum = "1.0";
            String opinionNum = "1";

            assertEquals(false, mc.getAdminV().delOpinion(projectName,
verNum, opinionNum));

      }

      public void testDownloadFile() {

            String projectName = "Test1";
            String verNum = "1.0";
            assertNotNull(mc.getAdminV().downloadFile(projectName,
verNum));

      }

      public void testRcvVerList() {


            final String projectName_1 = "Test1";
            assertNotNull(mc.getAdminV().rcvVerList(projectName_1));

      }

      public void testRcvOipnionList() {


            final String projectName_1 = "Test1";
            final String verNum_1 = "1.0";
            assertNotNull(mc.getAdminV().rcvOpinionList(projectName_1,
verNum_1));

      }
```

Activity 2163. System Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass / Fail
1 - 1	Login Test	정상적으로 Login 이 되는지 확인한다.	Login	R.1	P
1 - 2	Login Test	존재하지 않는 ID 를 입력했을 때 알림이 뜨는지 확인한다.	Login	R.1	P
1 - 3	Login Test	일치하지 않는 Password 를 입력했을 때 알림이 뜨는지 확인한다	Login	R.1	P
2 - 1	PW Change Test	정상적으로 Password 가 변경되는지 확인한다.	ChangePW	R.2	P
2 -2	PW Change Test	새로 입력한 2 개의 Password 가 일치하지 않으면 알림이 뜨는지 확인한다.	ChangePW	R.2	P
2 -3	PW Change Test	현재의 PW 와 입력 받은 PW 가 일치하지 않으면 알림이 뜨는지 확인한다.	ChangePW	R.2	P
3 - 1	Project Add Test	정상적으로 project 가 추가되는지 확인한다.	AddProject	R.3	P
3 - 2	Project Add Test	Admin 이 아닌 User 가 project 를 추가하려고 하면 알림이 뜨는지 확인한다.	AddProject	R.3	P
4 – 1	Project Delete Test	정상적으로 project 가 삭제되는지 확인한다.	DelProject	R.4	P
4 - 2	Project Delete Test	Admin 이 아닌 User 가 project 를 삭제하려고 하면 알림이 뜨는지 확인한다.	DelProject	R.4	P
5 -1	Version Add Test	하위 Version 이 정상적으로 추가되는지 확인한다.	AddVer	R.5.1	P
5 - 2	Version Add Test	상위 Version 이 정상적으로 추가되는지 확인한다.	AddVer	R.5.1	P
5 – 3	Version Add Test	Test Specification 파일의 확장자가 txt 가 아니면 알림을 띄우는지 확인한다.	AddVer	R.5.1	P
5 - 4	Version Add Test	Version 을 추가하면서 작성한 comment 가 해당 Version 의 공지 Opinion 으로 들어가는지 확인한다.	AddVer	R.5.1	P
5 - 4	File Analyze Test	Category 와 Choice 와 Constraints 를 잘 분석하는 지 확인한다.	AnalyzeFile	R.5.2	P
5 - 5	Case Make Test	Test Case 를 정상적으로 조합하고 개수를 잘 계산하는 지 확인한다.	MakeCase	R.5.3	P
6 – 1	Version Delete Test	정상적으로 Version 이 삭제되는지 확인한다.	DelVer	R.6	P
6 - 2	Version Delete Test	(Version 을 추가했거나 Admin 인 User)가 아닌 User 가 Version 을 삭제하려고 하면 알림이 뜨는지 확인한다.	DelVer	R.6	P
7 - 1	File Download Test	File 이 정상적으로 download 되는지 확인한다.	DownloadFile	R.7	P
8 - 1	Opinion Add Test	Opinion 이 정상적으로 추가되는지 확인한다.	AddOpinion	R.8	P

9 - 1	Opinion Delete Test	Opinion 을 삭제했을 때 Opinion 목록에서 삭제되는지 확인한다.	DelOpinion	R.9	P
9 -2	Opinion Delete Test	Opinion 작성자가 아니면 알림 창이 뜨는지 확인한다.	DelOpinion	R.9	F
10 - 1	Exit Test	프로그램이 정상적으로 종료되는지 확인한다.	Exit	R.10	P
11-1	Analyze File Performance Test	Test Case 를 분석하는 작업이 5 초 이내로 수행되는지 확인한다.	Non-Functional		P
12-1	FileUpload Performance Test	File 의 업로드 작업이 5 초이내로 수행되는지 확인한다.	Non-Functional		p