

Unit Test Plan [T1] for Network Printer System

201011375 하지용

201211823 유치영

201213220 허인석

201214262 라가영

1#

테스트 기준

2# 테스트 기준

1. 시스템 핵심 기능 관련 프로세스에 테스트 중점
2. 경우의 수가 없는 단순한 프로세스는 테스트 제외
3. 모듈 별 요구사항의 만족 여부, 정상적인 결과의 출력 여부, 잘못된 값 입력 시 예외 처리의 동작의 여부를 확인

2#

개발 환경

2# 개발 환경

1. 개발

Microsoft Visual Studio 2010
gcc

2. 형상 관리

Tortoise SVN / Naver Open Project SVN Server

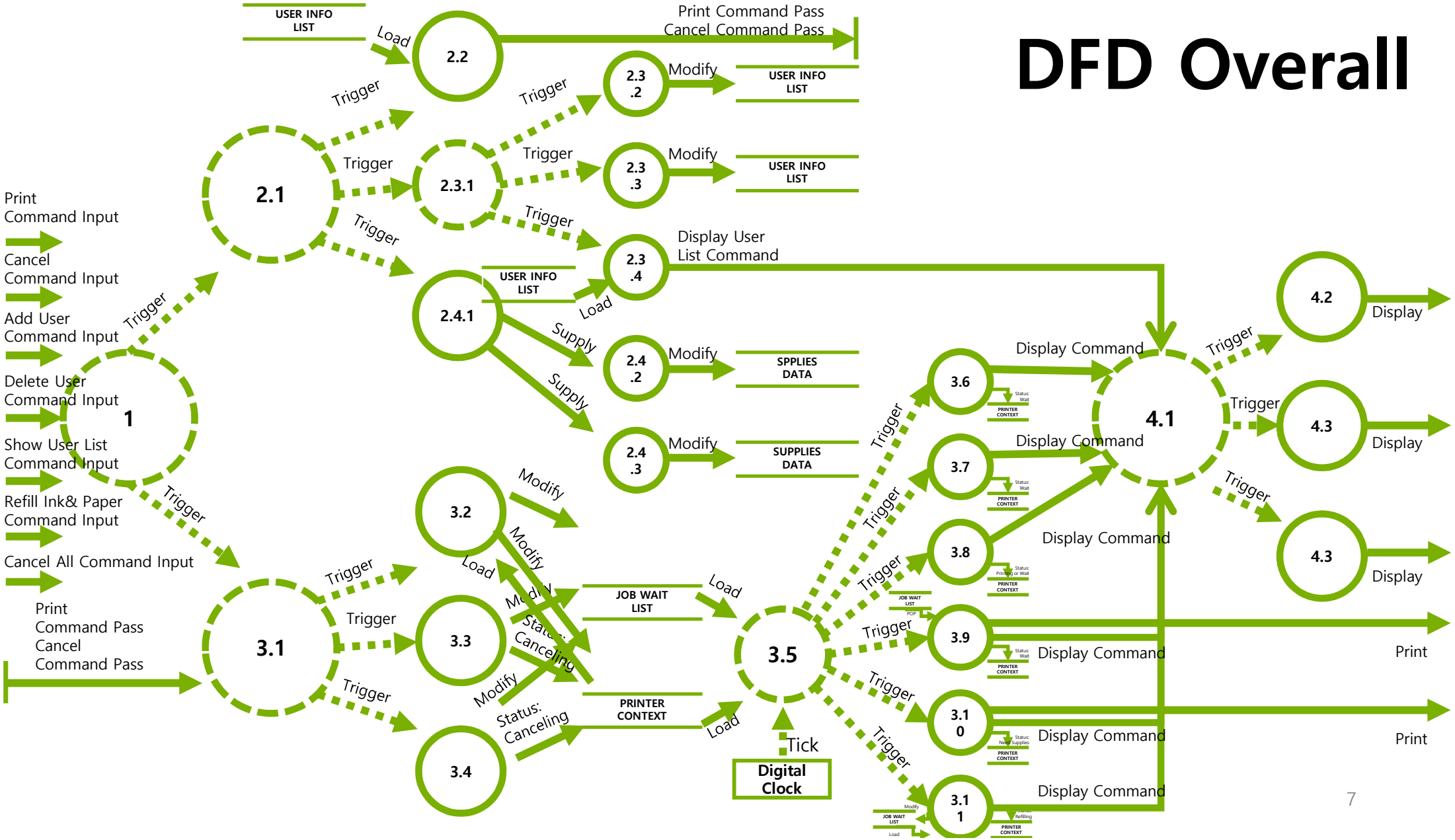
3. 실행 및 테스트 환경

Cygwin

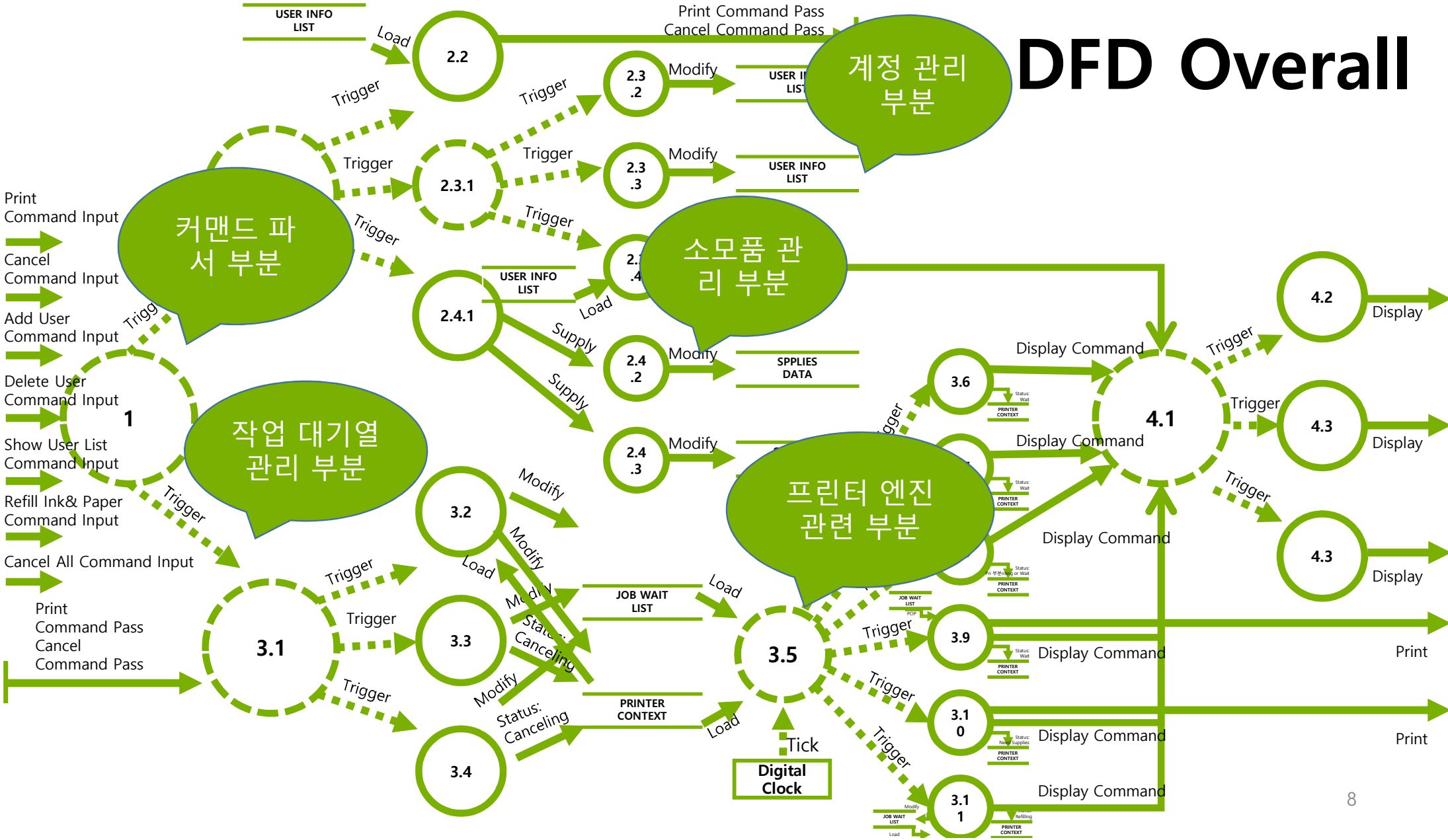
3#

DFD Overall

DFD Overall



DFD Overall



4#

대단위 모듈 설명

4# 대단위 모듈 설명

테스트할 모듈

Identifier.	Feature.
NPS.UTC.1000.	입력된 명령을 분류하여 각 명령을 Admin Sub System과 Print Sub System으로 전달한다..
NPS.UTC.2031.	관리자가 유저를 추가/삭제 및 유저리스트 확인을 할 것인지를 Add User, Delete User, Show User List로 분류하여 각 프로세스를 실행한다..
NPS.UTC.2032.	유저를 추가한다..
NPS.UTC.2033.	유저를 삭제한다..
NPS.UTC.2034.	유저 리스트를 확인한다..
NPS.UTC.2041.	유저가 Ink를 리필 할 것인지, Paper를 리필 할 것인지Supplies Data를 Ink와 Paper로 분류하여 각 프로세스를 실행한다..
NPS.UTC.2042.	유저가 Ink를 리필 할 경우, Supplies Data 내 충전할 Ink량을 수정한다..
NPS.UTC.2043.	유저가 Paper를 리필 할 경우, Supplies Data 내 충전할 Paper량을 수정한다..
NPS.UTC.3010.	명령이 추가 / 삭제, 전체 취소 인지를 확인하며 Add Job, Delete Job, Cancel All로 분류하여 각 프로세스를 실행한다..
NPS.UTC.3020.	Admin Sub System에서 authorize check가 된 인쇄 추가 명령을 받아 Job Wait List과 Print Context를 수정한다..
NPS.UTC.3030.	Admin Sub System에서 인쇄 삭제 명령을 받아 Job Wait List과 Print Context를 수정한다..
NPS.UTC.3040.	Command Parser에서 Cancel All Command를 받아 Job Wait List과 Print Context를 수정한다..

4# 대단위 모듈 설명

테스트할 모듈

NPS.UTC.3050.,	JOB WAIT LIST와 PRINTER CONTEXT의 데이터를 받아와 각 프로세스로 전달한다..
NPS.UTC.3060.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 대기중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait으로 변경한다..
NPS.UTC.3070.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 취소 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait로 변경한다..
NPS.UTC.3080.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 충전이 요구되는 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Need Supplies로 변경한다..
NPS.UTC.3090.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, JOB WAIT LIST에서 최상위 작업을 불러와 출력을 시작한다. 출력 중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Printing으로 변경한다..
NPS.UTC.3100.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 출력중인 작업을 지속한다..
NPS.UTC.3110.,	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 잉크와 용지를 SUPPLIES DATA에서 Paper가 0이 아닐 때 10만큼, Ink가 0이 아닐 때 100만큼 감소 후 같은 량을PRINTER CONEXT의 Ink와 Paper에 증가시킨다. .

5#

단위 모듈 테스트 실행 보고

5# 단위 모듈 테스트 실행 보고

커맨드 파서 관련

Identifier.	Feature.	Pass / Fail Criteria.
NPS.UTC.1000	입력된 명령을 분류하여 각 명령을 Admin Sub System과 Print Sub System으로 전달한다.	
NPS.UTC.1000.00.	user add user a1.	<u>ERROR: not supported command.</u>
NPS.UTC.1000.01.	user del user a1.	<u>ERROR: not supported command.</u>
NPS.UTC.1000.02.	user show user list.	<u>ERROR: not supported command.</u>

5# 단위 모듈 테스트 실행 보고

소모품 충전 관련

NPS.UTC.2041	유저가 Ink를 리필 할 것인지, Paper를 리필 할 것인지Supplies Data를 Ink와 Paper로 분류하여 각	
	프로세스를 실행한다.	
NPS.UTC.2041.00.	전달 역할 등 단순한 프로세스 및 UI구성을 담당 하는 프로세스는 테스트에서 제외.	.
NPS.UTC.2042	유저가 Ink를 리필 할 경우, Supplies Data 내 충전할 Ink량을 수정한다.	
NPS.UTC.2042.00.	KEYWORD_INK == TRUE., <u>inkValue =100.</u>	<u>g_SuppliesData.Ink={100}.</u>
NPS.UTC.2042.01.	KEYWORD_INK == TRUE., <u>inkValue = -100.</u>	<u>ERROR:Invalid Input ink Value.</u>
NPS.UTC.2043	유저가 Paper를 리필 할 경우, Supplies Data 내 충전할 Paper량을 수정한다.	
NPS.UTC.2043.00.	KEYWORD_PAPER == TRUE., <u>paperValue =100.</u>	<u>g_SuppliesData.Paper={100}.</u>
NPS.UTC.2043.01.	KEYWORD_PAPER == TRUE., <u>paperValue = -100.</u>	<u>ERROR:Invalid Input paper Value.</u>
NPS.UTC.2043.02.	KEYWORD_PAPER == TRUE., <u>paperValue = 100.</u> KEYWORD_INK == TRUE., <u>inkValue = 100.</u>	<u>g_SuppliesData.Paper={100}.</u> <u>g_SuppliesData.Ink={100}.</u>
NPS.UTC.2043.03.	KEYWORD_PAPER == TRUE., <u>paperValue = 100.</u> KEYWORD_INK == TRUE., <u>inkValue = -100.</u>	<u>g_SuppliesData.Paper={100}.</u> <u>ERROR:Invalid Input ink Value.</u>

5# 단위 모듈 테스트 실행 보고

소모품 충전 관련

NPS.UTC.2043.04.	KEYWORD_PAPER == TRUE., <u>paperValue</u> = -100., KEYWORD_INK == TRUE., <u>inkValue</u> =100.,	<u>ERROR:Invalid</u> Input paper Value., <u>g_SuppliesData.Ink</u> ={10 0}.,
NPS.UTC.2043.05.	KEYWORD_PAPER == TRUE., <u>paperValue</u> = -100., KEYWORD_INK == TRUE., <u>inkValue</u> = -100.,	" <u>ERROR:Invalid</u> Input paper Value., <u>ERROR:Invalid</u> Input ink Value.,
NPS.UTC.2043.06.	KEYWORD_INK == TRUE., <u>inkValue</u> =100., KEYWORD_PAPER == TRUE., <u>paperValue</u> =100.,	<u>g_SuppliesData.Ink</u> ={10 0}., <u>g_SuppliesData.Paper</u> ={ 100}.,
NPS.UTC.2043.07.	KEYWORD_INK == TRUE., <u>inkValue</u> =100., KEYWORD_PAPER == TRUE.,	<u>g_SuppliesData.Ink</u> ={10 0}., <u>ERROR:Invalid</u> Input
NPC.UTS.2043.08.	KEYWORD_INK == TRUE., <u>inkValue</u> = -100., KEYWORD_PAPER == TRUE., <u>paperValue</u> =100.,	<u>ERROR:Invalid</u> Input ink Value., <u>g_SuppliesData.Paper</u> ={ 100}.,
NPC.UTS.2043.09.	KEYWORD_INK == TRUE., <u>inkValue</u> = -100., KEYWORD_PAPER == TRUE., <u>paperValue</u> = -100.,	<u>ERROR:Invalid</u> Input ink Value., <u>ERROR:Invalid</u> Input paper Value.,

5# 단위 모듈 테스트 실행 보고

작업 관련

NPC.UTS.3010	명령이 추가 / 삭제, 전체 취소 인지를 확인하며 Add Job, Delete Job, Cancel All로 분류하여 각 프로세스를 실행한다.	
NPC.UTS.3010.00.	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외.	.
NPC.UTS.3020	Admin Sub System에서 authorize check가 된 인쇄 추가 명령을 받아 Job Wait List과 Print Context를 수정한다.	
NPC.UTS.3020.00.	<u>AddJob(userIdx, "test.txt");</u> .	<u>g_JobWaitList={test.txt};</u> .
NPC.UTS.3020.01.	<u>AddJob(userIdx1, "test1.txt");</u> <u>AddJob(userIdx2, "test2.txt");</u> .	<u>g_JobWaitList={test.txt1</u> <u>,test.txt2};</u> .
NPC.UTS.3020.02.	<u>GetNodeCount(g_JobWaitList)=5;</u> <u>AddJob(userIdx, "test.txt");</u> .	ERROR:JOB WAIT LIST FULL.
NPC.UTS.3020.03.	<u>AddJob(userIdx, "test.txt");</u> test.txt != EXIST.	ERROR:request file open failed.

5# 단위 모듈 테스트 실행 보고

작업 관련

NPC.UTS.3030	Admin Sub System에서 인쇄 삭제 명령을 받아 Job Wait List과 Print Context를 수정한다	
NPC.UTS.3030.00.	<u>AddJob(userIdx1, "test1.txt");</u> .. <u>AddJob(userIdx2, "test2.txt");</u> .. <u>DeleteJob(userIdx1);</u> ..	<u>g_JobWaitList={test.txt2</u> <u>};</u>
NPC.UTS.3030.01.	<u>GetNodeCount(g_JobWaitList)==0;</u> .. <u>DeleteJob(userIdx);</u> ..	ERROR: UID : 0 Job not found.
NPC.UTS.3030.02.	<u>AddJob(userIdx1, "test1.txt");</u> .. <u>DelelteJob(userIdx1);</u> ..	<u>g_PrinterContext.pCurre</u> <u>ntJob=Printing.</u> <u>g_PrinterContext.pCurre</u> <u>ntJob=NULL.</u>
NPC.UTS.3030.03.	<u>AddJob(userIdx1, "test1.txt");</u> .. <u>AddJob(userIdx2, "test2.txt");</u> .. <u>AddJob(userIdx1, "test2.txt");</u> .. <u>DelelteJob(userIdx1);</u> ..	<u>g_JobWaitList={test.txt2</u> <u>};</u>
NPC.UTS.3030.04.	<u>g_PrinterContext.status == Printing.</u> <u>DeleteJob(currentUserIdx);</u> ..	<u>g_PrinterContext.status</u> <u>=print.</u> <u>g_PrinterContext.status</u> <u>=wait.</u>
NPS.UTS.3040	Command Parser에서 Cancel All Command를 받아 Job Wait List과 Print Context를 수정한다.	
NPS.UTS.3040 00.	<u>AddJob(userIdx1, "test1.txt");</u> .. <u>AddJob(userIdx2, "test2.txt");</u> .. <u>AddJob(userIdx3, "test3.txt");</u> .. <u>CancelAll();</u> ..	<u>g_PrinterContext.status</u> <u>= CancelSignal.</u> <u>DeleteNode(&g_JobWai</u> <u>tlist nJobNode).</u>

5# 유닛 테스트 실행 보고

엔진 관련

NPC.UTS.3050	JOB WAIT LIST와 PRINTER CONTEXT의 데이터를 받아와 각 프로세스로 전달한다	
NPC.UTS.3050.00.	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외.	.
NPC.UTS.3060	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 대기중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait으로 변경한다.	
NPC.UTS.3060.00.	Interfaces는 testing 제외.	.
NPC.UTS.3070	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 취소 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait로 변경한다.	
NPC.UTS.3070.00.	<u>pCurrentJob</u> != NULL.	<u>g_PrinterContext.pCurrentJob</u> = NULL. <u>g_PrinterContext.status</u> = Wait.
NPC.UTS.3070.01.	<u>pCurrentJob</u> = NULL.	<u>g_PrinterContext.status</u> = Wait.
NPC.UTS.3080	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 충전이 요구되는 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Need Supplies로 변경한다.	
NPC.UTS.3080.00.	Interfaces는 testing 제외.	.

5# 유닛 테스트 실행 보고

엔진 관련

NPC.UTS.3090	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, JOB WAIT LIST에서 최상위 작업을 불러와 출력을 시작한다. 출력 중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Printing으로 변경한다	
NPC.UTS.3090.00.	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외.	.
NPC.UTS.3100	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 출력중인 작업을 지속한다.	
NPC.UTS.3100.00.	<u>pCurrentJob.needPaper != pCurrentJob.printedCnt.</u> <u>!feof(pCurrentJob.plnFile) == TRUE.</u>	<u>g_PrinterContext.status = Printing.</u> Printing 출력.
NPC.UTS.3100.01.	<u>pCurrentJob.needPaper != pCurrentJob.printedCnt.</u> <u>!feof(pCurrentJob.plnFile) == FALSE.</u>	<u>g_PrinterContext.status = Wait.</u> Wait 출력.
NPC.UTS.3100.02.	<u>pCurrentJob.needPaper == pCurrentJob.printedCnt.</u>	<u>fclose(pCurrentJob.plnFile).</u> <u>fclose(pCurrentJob.pOutputFile).</u> <u>free(pCurrentJob).</u> <u>g_PrinterContext.pCurrentJob = NULL.</u> <u>g_PrinterContext.status = Wait.</u> Wait 출력

5# 유닛 테스트 실행 보고

엔진 관련

NPC.UTS.3110	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 잉크와 용지를 SUPPLIES DATA에서 Paper가 0이 아닐 때 10만큼, Ink가 0이 아닐 때 100만큼 감소 후 같은 량을PRINTER CONEXT의 Ink와 Paper에 증가시킨다.	
NPC.UTS.3110.00.	<u>g_PrinterContext.ink</u> + <u>refillInk</u> > <u>MAX_SUPPLIES_INK</u> .,	<u>refillInk</u> = <u>refillInk</u> - (<u>g_PrinterContext.ink</u> + <u>refillInk</u> - <u>MAX_SUPPLIES_INK</u>).
NPC.UTS.3110.01.	<u>g_PinterContext.paper</u> + <u>refillPaper</u> > <u>MAX_SUPPLIES_PAPER</u> .,	<u>refillPaper</u> = <u>refillPaper</u> - (<u>g_PrinterContext.paper</u> + <u>refillPaper</u> - <u>MAX_SUPPLIES_PAPAER</u>).
NPC.UTS.3110.02.	<u>(g_SuppliesData.Ink != 0 && g_PrinterContext.ink < MAX_SUPPLIES_INK) (g_SuppliesData.Paper != 0 &&</u>	<u>g_PrinterContext.status</u> = Refilling.,
	<u>g_PrinterContext.paper < MAX_SUPPLIES_PAPER).</u>	
NPC.UTS.3110.03.	<u>(g_SuppliesData.Ink = 0 g_PrinterContext.ink > MAX_SUPPLIES_INK) && (g_SuppliesData.Paper = 0 g_PrinterContext.paper > MAX_SUPPLIES_PAPER).</u>	<u>g_PrinterContext.status</u> = Wait.,

감사합니다