

# Software Requirement Analysis for Network Printing System

Project Team

**T5 Team**

Date

**2015-11-10**

---

## Team Information

201411294 이상혁

201411296 이선명

201411305 이찬규

201411316 정진호

## Table of Contents

1	Introduction _____	6
1.1	Purpose _____	6
1.2	Scope _____	6
1.2.1	Developer Team _____	6
1.2.2	Restrictions _____	6
1.3	Definition, acronyms, and abbreviations _____	6
1.4	Reference _____	8
1.5	Overview _____	8
2	Overall Description _____	8
2.1	Product Perspective _____	8
2.2	Product functions _____	8
2.2.1	인쇄 _____	8
2.2.2	자원 관리 _____	8
2.2.3	상태 디스플레이 _____	8
2.3	User characteristics _____	9
2.4	Constraints _____	9
2.5	Assumptions and dependencies _____	9
3	Structured Analysis _____	10
3.1	System Context Diagram _____	10
3.1.1	Basic System Context Diagram _____	10
3.1.2	Event List _____	10
3.1.3	The System Context Diagram _____	10
3.2	Data Flow Diagram _____	11

3.2.1	DFD level 0	11
3.2.1.1	DFD	11
3.2.1.2	Process Specification	11
3.2.1.2.1	Process 1	11
3.2.1.3	Data Dictionary	12
3.2.2	DFD Level 1	13
3.2.2.1	DFD	13
3.2.2.2	Process Specification	13
3.2.2.2.1	Process 1	13
3.2.2.2.2	Process 2	14
3.2.2.2.3	Process 3	14
3.2.2.2.4	Process 4	14
3.2.2.2.5	Process 5	15
3.2.2.3	Data Dictionary	15
3.2.3	DFD Level 2	16
3.2.3.1	DFD	16
3.2.3.1.1	Management Process	16
3.2.3.1.2	View Process	17
3.2.3.1.3	Printing Process	17
3.2.3.2	Process Specification	18
3.2.3.2.1	Process 1.1	18
3.2.3.2.2	Process 1.2	18
3.2.3.2.3	Process 1.3	18
3.2.3.2.4	Process 2.1	19
3.2.3.2.5	Process 2.2	19
3.2.3.2.6	Process 2.3	19

3.2.3.2.7	Process 3.1	20
3.2.3.2.8	Process 3.2	20
3.2.3.2.9	Process 3.3	20
3.2.3.2.10	Process 3.4	21
3.2.3.2.11	Process 3.5	21
3.2.3.3	Data Dictionary	21
3.2.3.4	State Transition Diagram ( <i>View Controller 2.1</i> )	22
3.2.3.5	State Transition Diagram ( <i>Printing Controller 3.1</i> )	22
3.2.4	DFD Level 3	23
3.2.4.1	DFD	23
3.2.4.1.1	Request Process	23
3.2.4.1.2	H/W Stop Process	23
3.2.4.1.3	Virtual Refill Process	24
3.2.4.2	Process Specification	24
3.2.4.2.1	Process 1.1.1	24
3.2.4.2.2	Process 1.1.2	24
3.2.4.2.3	Process 1.1.3	25
3.2.4.2.4	Process 1.1.4	25
3.2.4.2.5	Process 1.1.5	25
3.2.4.2.6	Process 1.1.6	26
3.2.4.2.7	Process 1.1.7	26
3.2.4.2.8	Process 1.2.1	26
3.2.4.2.9	Process 1.2.2	26
3.2.4.2.10	Process 1.2.3	27
3.2.4.2.11	Process 1.3.1	27
3.2.4.2.12	Process 1.3.2	27

3.2.4.2.13	Process 1.3.3	28
3.2.4.2.14	Process 1.1.4.3	28
3.2.4.2.15	Process 1.1.4.4	28
3.2.4.2.16	Process 1.1.4.5	28
3.2.4.2.17	Process 1.1.4.6	29
3.2.4.2.18	Process 1.1.4.7	29
3.2.4.3	Data Dictionary	29
3.2.4.4	State Transition Diagram ( <i>Auth &amp; Dispatch Controller 1.1.1</i> )	31
3.2.4.5	State Transition Diagram ( <i>H/W Stop Controller 1.2.1</i> )	32
3.2.4.6	State Transition Diagram ( <i>Virtual Refill Controller 1.3.1</i> )	32
3.2.5	Overall DFD	33

## 1 Introduction

### 1.1 Purpose

본 문서는 Network Printing System을 구현하는 가상의 소프트웨어 시스템의 요구사항을 Structured Analysis 기법을 사용하여 분석한 내용을 담고 있다. 이 문서를 통하여 Network Printer System에 필요한 요구사항들 간의 관계를 명확히 표현하고 실제 구현의 토대를 구축한다.

### 1.2 Scope

#### 1.2.1 Developer Team

Class A – T5 Team

#### 1.2.2 Restrictions

하드웨어로 구동되는 모든 사항은 일종의 에뮬레이터(SW 혹은 Console 명령어)를 구현하여 가상으로 구현한다.

### 1.3 Definition, acronyms, and abbreviations

SW: Software

HW: Hardware

NPS: Network Printing System

인쇄요청: 사용자는 프린터에게 자신이 인쇄하고자 하는 파일을 전송한다.

충전: 관리자는 프린터의 자원(잉크, 종이)이 모자라는 경우, 자원 공급을 실행할 수 있다.

사용자 등록/삭제: 관리자는 사용자를 NPS에 등록하거나 삭제할 수 있다.

사용자 인증: NPS는 인쇄 요청을 보내는 사용자가 사용자 데이터베이스에 등록되어 있는지 확인한다.

프린터 출력: 사용자가 전송한 파일을 특정 형태로 인쇄하는 작업이다

잔량 확인: 사용자는 현재 프린터의 자원 상태(용지/잉크)

Cmd Network Response: 사용자나 관리자에게 받은 요청을 처리하고 난 결과로 요청한 사용자나 관리자에게 네트워크를 통해 전달

Data Format Structure: Data Dictionary에서 사용하는 Data Format Structure를 의미하며 모든 수치 데이터는 Big-endian을 기준으로 하며 C 스타일을 참조하여 struct의 내부 변수들만 기술하여 아래와 같은 방식으로 작성한다.

C Style	Format Structure
<pre> <b>struct</b> TestStruct {     <b>int</b> req_id;     <b>int</b> req_type = 1;     <b>int</b> file_size;     <b>char</b> file_data[file_size]; } </pre>	<pre> <b>int</b> req_id; <b>int</b> req_type = 1; <b>int</b> file_size; <b>char</b> file_data[file_size]; </pre>

또한 Request에서 req\_id는 매 요청마다 고유한(매번 다른) 값을 전송하여야 하며 Response에서의 req\_id와 req\_type는 해당 Request의 req\_id와 req\_type를 복사하여 사용한다.

req\_id에는 -1을 사용할 수 없다. (프린터 상태 모니터링용으로 사용됨)

Network 구현에 있어서 TCP(port:6000)를 사용하며 Payload는 Format Structure를 따른다.

Virtual\_Refill\_Interface는 UDP를 사용하며 127.0.0.1:6500을 사용한다.

HW\_Stop 는 UDP를 사용하며 127.0.0.1:6600으로 데이터를 전송한다.

LCD\_Interface는 UDP를 사용하며 127.0.0.1:6700으로 데이터를 전송한다.

PAPER\_MAX\_VALUE : 100

INK\_MAX\_VALUE : 3000

JOB\_MAX\_SIZE: 5

PRINTING\_OR\_WAIT: 0

REMOVING: 1

PACKET\_TYPE\_REQUEST: 0

PACKET\_TYPE\_CANCEL: 1

PACKET\_TYPE\_USER\_CREATION: 2

PACKET\_TYPE\_USER\_REMOVAL: 3

PACKET\_TYPE\_USER\_LISTING: 4

#### 1.4 Reference

<http://dslab.konkuk.ac.kr>

#### 1.5 Overview

이 문서에는 Network Printing System의 Data Flow Diagram, State Transition Diagram, Data Dictionary(+ Data Format Structure), Process Specification이 나타나 있다.

### 2 Overall Description

#### 2.1 Product Perspective

대상 제품은 실제 네트워크 프린터 체계를 가상화하였다. 네트워크를 통하여 사용자와 관리자에게 입력을 받고, 가상의 리필 인터페이스를 이용하여 자원의 상태를 수정 할 수 있으며 프린터는 이러한 입력에 따라서 출력 또는 자원 관리를 수행한다. 그리고 프린터는 자신의 상태를 LCD 화면 또는 네트워크를 통하여 나타내게 된다. 현 제품의 경우 HW로 작동하는 부분은 모두 SW 및 Console 명령어로 가상화하여 기능의 동작 여부를 확인하도록 한다.

#### 2.2 Product functions

##### 2.2.1 인쇄

사용자는 프린터로 인쇄 요청을 하고 프린터는 요청 받은 사항을 출력한다.

##### 2.2.2 자원 관리

프린터는 현재 자신의 상태를 지속적으로 체크하고, 상태를 Network, LCD로 전달 하며 관리자는 프린터의 자원을 충전할 수 있다.

##### 2.2.3 상태 디스플레이



프린터가 반환한 현재 상태 (자원, Job Queue 등)을 LCD 또는 네트워크로 출력해 준다. 이 시스템의 경우 LCD는 Cygwin Console로 가상화 될 것이다.

### 2.3 User characteristics

한 사용자가 중복으로 프린트를 요청할 수 있다.

관리자의 경우 잉크와 종이 충전이 가능하다.

### 2.4 Constraints

모든 명령은 Cygwin의 command로 수행된다.

사용자는 존재하지 않는 파일의 인쇄를 요청 할 수 없다.

### 2.5 Assumptions and dependencies

사용자 입력: 사용자로부터 프린터 요청은 Cygwin 명령으로 대체한다.

출력할 데이터는 사용자로부터 파일(\*.txt)로 직접 입력 받는다. (ex: Cygwin>userA doc.txt)

관리자 입력: 관리자로부터 용지 및 잉크 충전은 Cygwin 명령으로 대체한다. (ex: Cygwin>Admin A4 100 / Cygwin>Admin Ink 100)

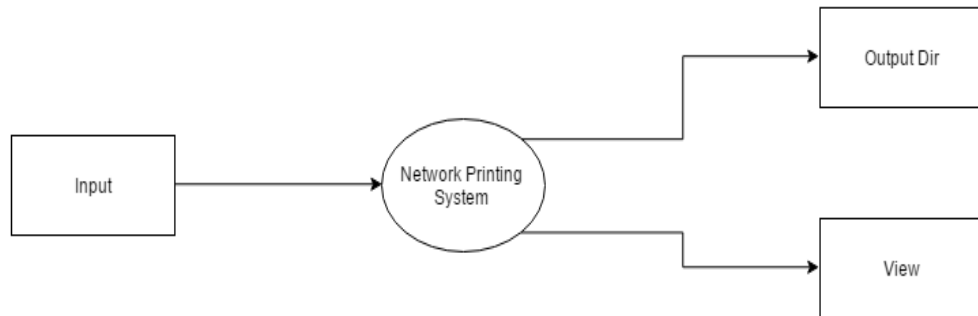
관리자에 의한 사용자 등록/삭제: 관리자로부터 사용자 등록/삭제는 Cygwin 명령으로 대체한다. (ex: Cygwin>Admin Show UserList)

출력물: 출력물 내용을 \*.txt 파일로 저장한다. 하나의 폴더에 모두 저장한다. 저장 이름은 Job별로 날짜와 시간으로 작성한다. (ex: 20150912150000.txt)

### 3 Structured Analysis

#### 3.1 System Context Diagram

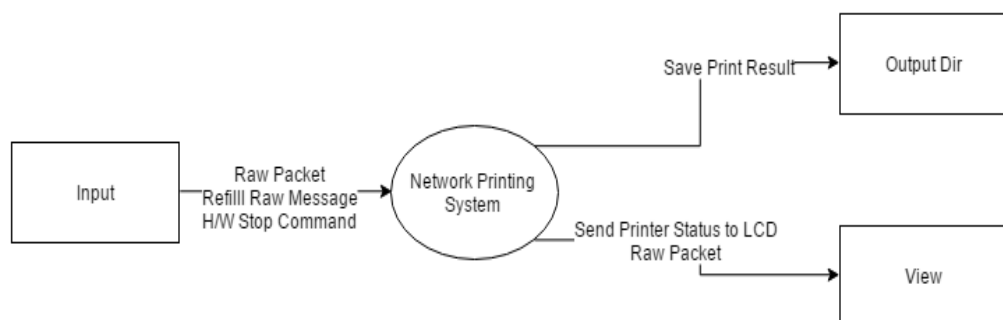
##### 3.1.1 Basic System Context Diagram



##### 3.1.2 Event List

Input / Output Event	Description
<b>Raw Packet</b>	사용자의 요청이나 응답을 담은 데이터
<b>Refill Raw Message</b>	리필 요청 메시지
<b>H/W Stop Command</b>	강제 중지 버튼 클릭 시 발생
<b>Send Printer Status to LCD</b>	LCD에 현재 프린터 상태를 표시
<b>Save Print Result</b>	인쇄 결과물을 저장

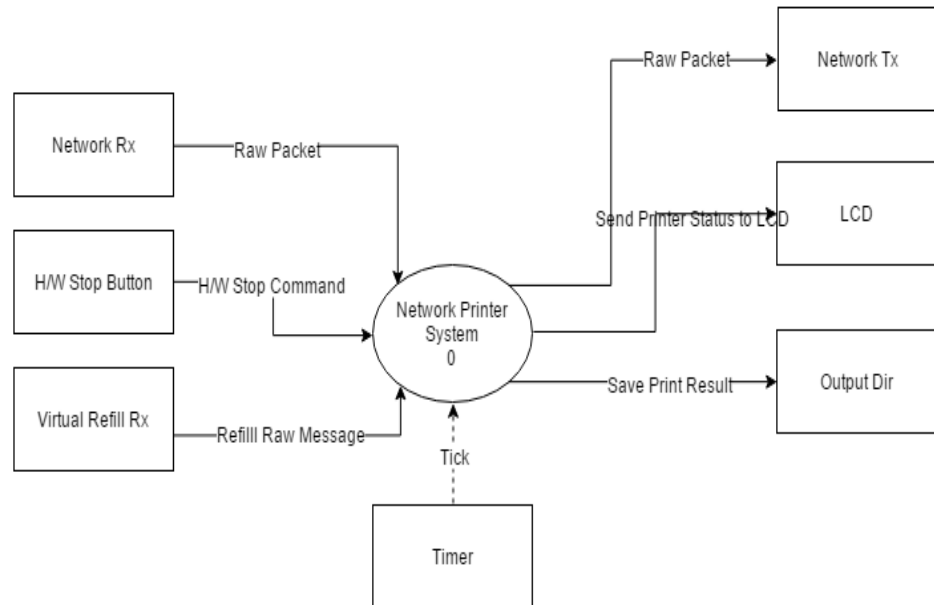
##### 3.1.3 The System Context Diagram



### 3.2 Data Flow Diagram

#### 3.2.1 DFD level 0

##### 3.2.1.1 DFD



##### 3.2.1.2 Process Specification

###### 3.2.1.2.1 Process 1

<b>Reference No</b>	<b>0</b>
<b>Name</b>	Network Printer System
<b>Input</b>	Raw Packet Refill Raw Message H/W Stop Command
<b>Output</b>	Save Print Result Send Printer Status to LCD Raw Packet
<b>Process Description</b>	네트워크 프린터 시스템으로 아래와 같은 내용을 처리함 <ol style="list-style-type: none"> <li>1. Network Rx를 통하여 사용자나 관리자로부 터 명령을 받고 명령에 따른 루틴을 수행 한 이후 결과를 Network Tx를 통하여 요청 자에게 전송함</li> <li>2. H/W Stop 버튼을 통하여 현재 진행중인 인쇄를 중지 할 수 있음</li> </ol>

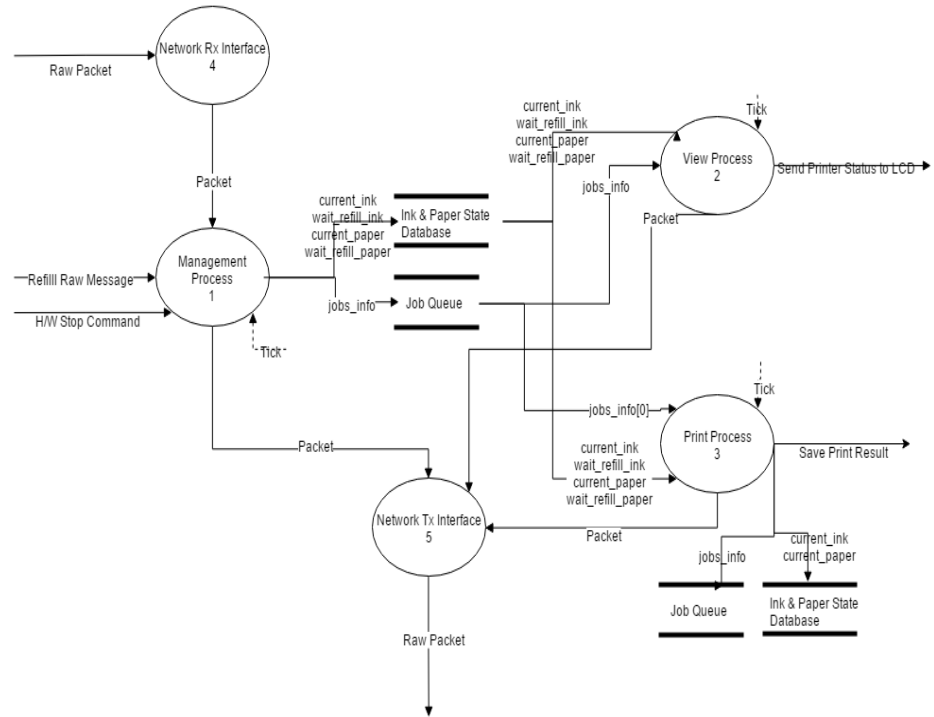
	<ul style="list-style-type: none"> <li>3. Virtual Refill Rx를 통해 잉크나 용지 충전 요청을 받아 Ink와 Paper을 충전함</li> <li>4. 인쇄 결과를 Save Print Result를 통하여 Output Dir에 저장함</li> <li>5. 현재 프린터 상태(Job Queue, Wait-Refill Ink&amp;Paper Value, Current Ink&amp;Paper Value)를 Network Tx와 LCD로 전달함</li> </ul>
--	--

## 3.2.1.3 Data Dictionary

Input / Output Event	Description	Format Structure
<b>Tick</b>	50ms 마다 발생	
<b>Raw Packet</b>	사용자의 요청이나 응답을 담은 데이터	char var=0; int length; char data[length]; // Packet
<b>Refill Message</b> <b>Raw</b>	리필 요청 메시지	// REFILL_TYPE_INK 0 // REFILL_TYPE_PAPER 1 char type int value
<b>H/W Command</b> <b>Stop</b>	강제 중지 버튼 클릭 시 발생	char stop=1
<b>Send Printer Status to LCD</b>	LCD에 현재 프린터 상태를 표시	char *message
<b>Save Print Result</b>	인쇄 결과물을 저장	

### 3.2.2 DFD Level 1

#### 3.2.2.1 DFD



#### 3.2.2.2 Process Specification

##### 3.2.2.2.1 Process 1

<b>Reference No</b>	<b>1</b>
<b>Name</b>	Management Process
<b>Input</b>	<a href="#">Tick</a> <a href="#">Packet</a> <a href="#">H/W Stop Command</a> <a href="#">Refill Raw Message</a>
<b>Output</b>	<a href="#">Packet</a> <a href="#">jobs_info</a> <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Process Description</b>	프린터의 자원을 관리하고 사용자 관리나 인증을 처리하는 프로세스

## 3.2.2.2.2 Process 2

<b>Reference No</b>	<b>2</b>
<b>Name</b>	View Process
<b>Input</b>	<a href="#">jobs_info</a> <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Output</b>	<a href="#">Packet</a> <a href="#">Send Printer Status to LCD</a>
<b>Process Description</b>	LCD나 Network로 현재 프린터 상태를 전달하는 프로세스

## 3.2.2.2.3 Process 3

<b>Reference No</b>	<b>3</b>
<b>Name</b>	Printing Process
<b>Input</b>	<a href="#">jobs_info[0]</a> <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Output</b>	<a href="#">Save Print Result</a> <a href="#">Packet</a> <a href="#">jobs_info</a> <a href="#">current_ink</a> <a href="#">current_paper</a>
<b>Process Description</b>	실제 프린트를 수행하는 프로세스 (First Job만 가지고 작업한다)

## 3.2.2.2.4 Process 4

<b>Reference No</b>	<b>4</b>
<b>Name</b>	Network Rx Interface
<b>Input</b>	<a href="#">Raw Packet</a>
<b>Output</b>	<a href="#">Packet</a>
<b>Process Description</b>	네트워크 수신 인터페이스로 TCP Packet를 수신 받아 내부적으로 사용되는 데이터 형태로 변환한다.

## 3.2.2.2.5 Process 5

<b>Reference No</b>	<b>5</b>
<b>Name</b>	Network Tx Interface
<b>Input</b>	<a href="#">Packet</a>
<b>Output</b>	<a href="#">Raw Packet</a>
<b>Process Description</b>	네트워크 송신 인터페이스로 내부적으로 사용되는 데이터들을 Packet로 가공하여 TCP Packet로 전달한다

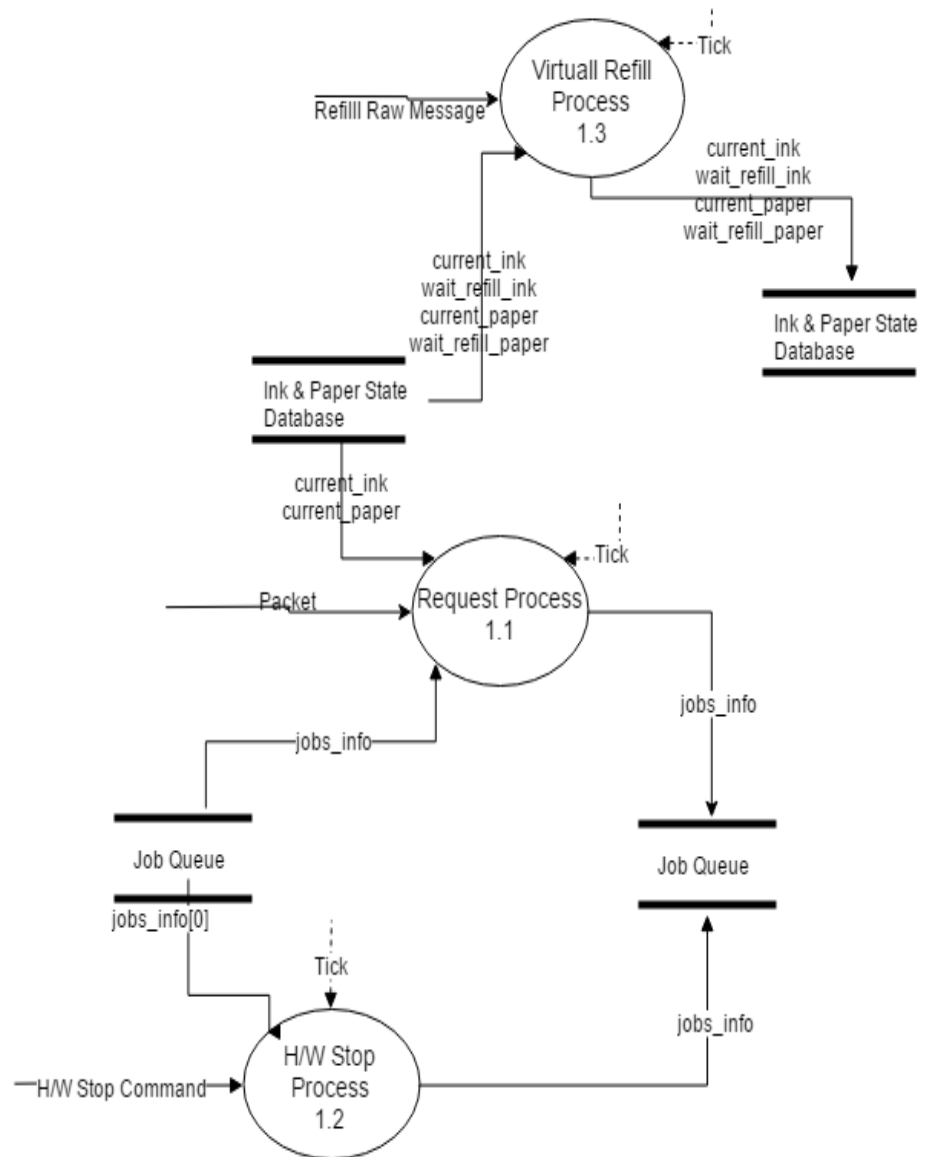
## 3.2.2.3 Data Dictionary

<b>Input / Output Event</b>	<b>Description</b>	<b>Format Structure</b>
<b>Packet</b>	사용자와의 통신을 위한 규격으로 Packet->data는 Packet(~)의 세부 규격을 따른다	int length; char *data; int sock;
<b>jobs_info</b>	인쇄 Job 큐에 저장되는 Job 정보	struct JobInfo { // not exist : -1 int job_id; int state; char user_id[12]; int current_page; int total_page; int file_size; int fd; char *buffer; };  struct JobInfo jobs_info[JOB_MAX_SIZE];
<b>current_ink</b>	현재 잉크 잔량	Int
<b>wait_refill_ink</b>	리필 대기 잉크량	Int
<b>current_paper</b>	현재 종이 잔량	Int
<b>wait_refill_paper</b>	리필 대기 종이량	int

3.2.3 DFD Level 2

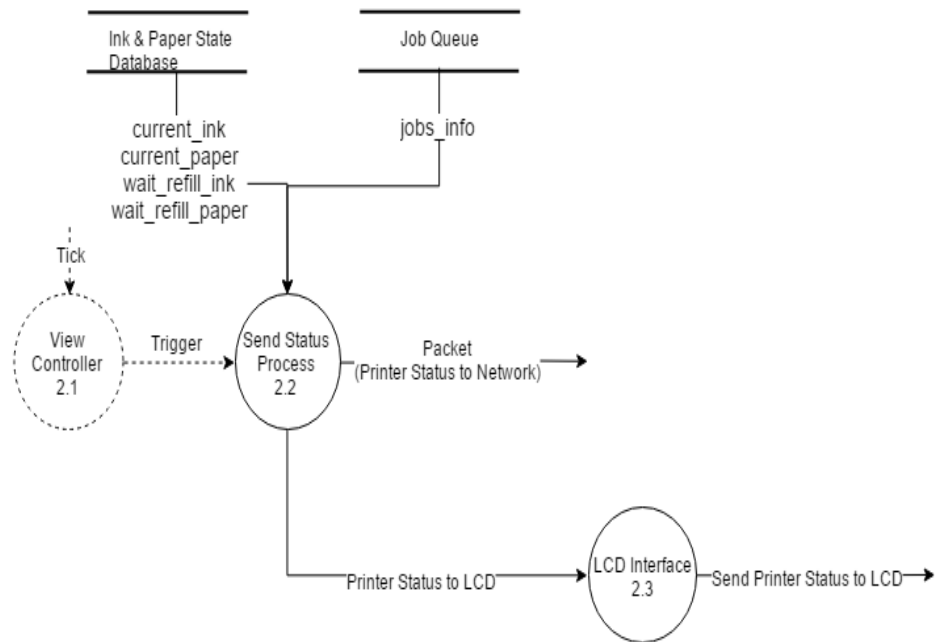
3.2.3.1 DFD

3.2.3.1.1 Management Process

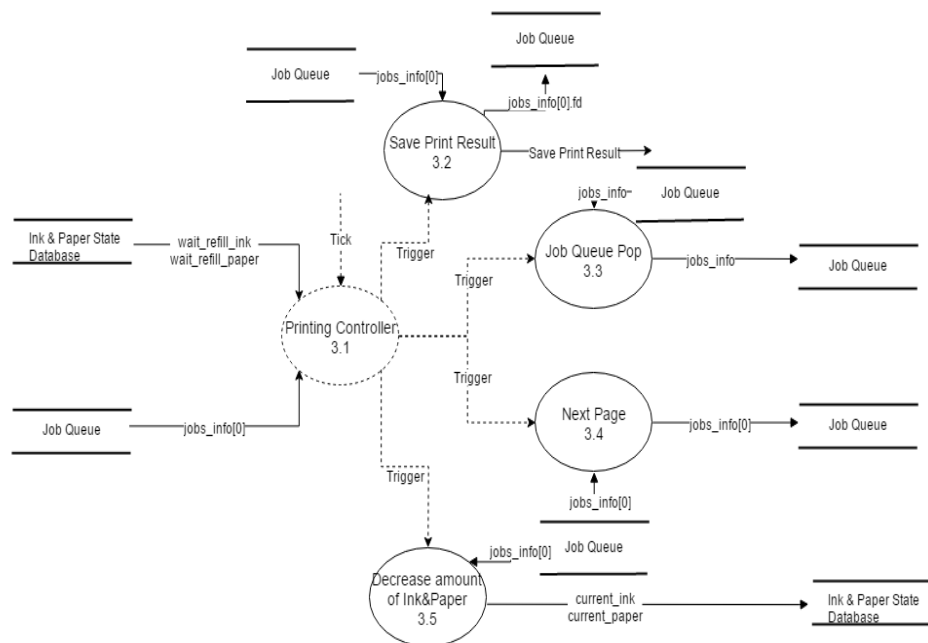




### 3.2.3.1.2 View Process



### 3.2.3.1.3 Printing Process



## 3.2.3.2 Process Specification

## 3.2.3.2.1 Process 1.1

<b>Reference No</b>	<b>1.1</b>
<b>Name</b>	Request Process
<b>Input</b>	<a href="#">Tick</a> <a href="#">Packet</a> <a href="#">current_ink</a> <a href="#">current_paper</a> <a href="#">jobs_info</a>
<b>Output</b>	<a href="#">jobs_info</a>
<b>Process Description</b>	네트워크로부터 받은 사용자의 요청을 처리하는 프로세스

## 3.2.3.2.2 Process 1.2

<b>Reference No</b>	<b>1.2</b>
<b>Name</b>	H/W Stop Process
<b>Input</b>	<a href="#">Tick</a> <a href="#">jobs_info[0]</a> <a href="#">H/W Stop Command</a>
<b>Output</b>	<a href="#">jobs_info</a>
<b>Process Description</b>	하드웨어 Stop을 담당하는 프로세스

## 3.2.3.2.3 Process 1.3

<b>Reference No</b>	<b>1.3</b>
<b>Name</b>	Virtual Refill Process
<b>Input</b>	<a href="#">Tick</a> <a href="#">Refill Raw Message</a> <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Output</b>	<a href="#">current_ink</a> <a href="#">wait_refill_ink</a>

	<a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Process Description</b>	자원 충전을 처리하는 프로세스

## 3.2.3.2.4 Process 2.1

<b>Reference No</b>	2.1
<b>Name</b>	View Controller
<b>Input</b>	<a href="#">Tick</a>
<b>Output</b>	Trigger
<b>Process Description</b>	500ms 마다 Trigger를 발생

## 3.2.3.2.5 Process 2.2

<b>Reference No</b>	2.2
<b>Name</b>	Send Status Process
<b>Input</b>	Trigger <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a> <a href="#">jobs_info</a>
<b>Output</b>	<a href="#">Printer Status to LCD</a> <a href="#">Packet (Printer Status to Network)</a>
<b>Process Description</b>	각종 프린터 상태를 받아 LCD나 네트워크로 전달

## 3.2.3.2.6 Process 2.3

<b>Reference No</b>	2.3
<b>Name</b>	LCD Interface
<b>Input</b>	<a href="#">Printer Status to LCD</a>
<b>Output</b>	<a href="#">Send Printer Status to LCD</a>
<b>Process Description</b>	LCD로 데이터를 전달하는 인터페이스

## 3.2.3.2.7 Process 3.1

<b>Reference No</b>	<b>3.1</b>
<b>Name</b>	Printing Controller
<b>Input</b>	<a href="#">Tick</a> <a href="#">wait_refill_ink</a> <a href="#">wait_refill_paper</a> <a href="#">jobs_info[0]</a>
<b>Output</b>	Trigger
<b>Process Description</b>	인쇄 명령을 수행하는 프로세스

## 3.2.3.2.8 Process 3.2

<b>Reference No</b>	<b>3.2</b>
<b>Name</b>	Save Print Result
<b>Input</b>	Trigger <a href="#">jobs_info[0]</a>
<b>Output</b>	<a href="#">Save Print Result</a> <a href="#">jobs_info[0].fd</a>
<b>Process Description</b>	실제 파일 저장을 수행하는 프로세스  출력물의 1줄은 30글자가 최대이다. (30글자 이상 부분은 출력 하지 않음) 잉크 소모 시 공백은 제외한다.

## 3.2.3.2.9 Process 3.3

<b>Reference No</b>	<b>3.3</b>
<b>Name</b>	Job Queue Pop
<b>Input</b>	Trigger <a href="#">jobs_info</a>
<b>Output</b>	<a href="#">jobs_info</a>
<b>Process Description</b>	첫번째 Job를 제거하는 프로세스

## 3.2.3.2.10 Process 3.4

<b>Reference No</b>	<b>3.4</b>
<b>Name</b>	Next Page
<b>Input</b>	Trigger <a href="#">jobs_info[0]</a>
<b>Output</b>	<a href="#">jobs_info[0]</a>
<b>Process Description</b>	인쇄중인 페이지를 다음 장으로 변경하는 프로세스

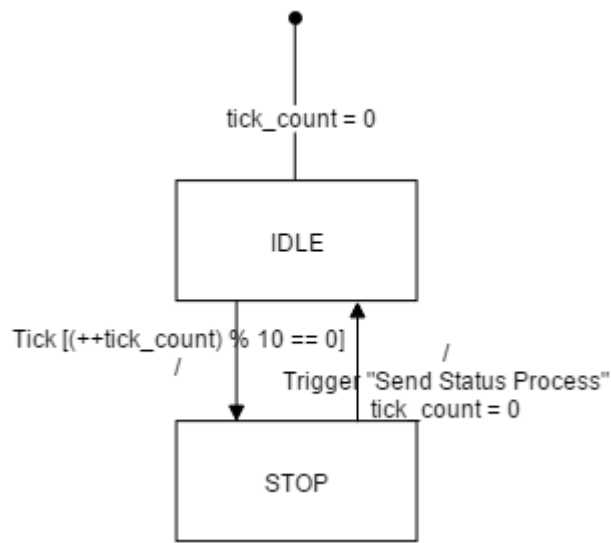
## 3.2.3.2.11 Process 3.5

<b>Reference No</b>	<b>3.5</b>
<b>Name</b>	Decrease amount of Ink/Paper
<b>Input</b>	Trigger <a href="#">jobs_info[0]</a>
<b>Output</b>	<a href="#">current_ink</a> <a href="#">current_paper</a>
<b>Process Description</b>	Ink&Paper을 소모 처리하는 프로세스

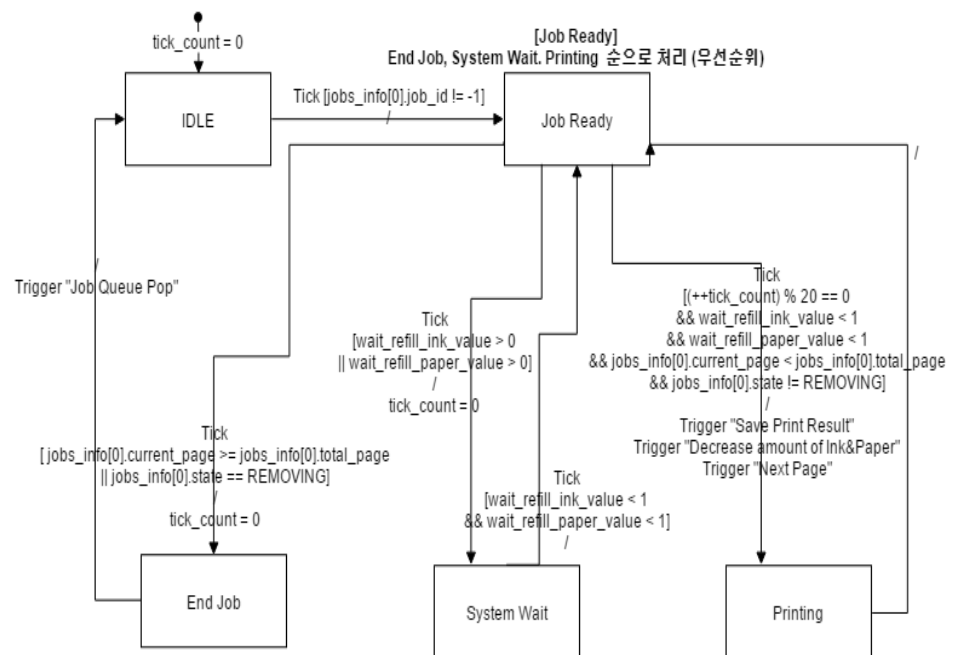
## 3.2.3.3 Data Dictionary

<b>Input / Output Event</b>	<b>Description</b>	<b>Format Structure</b>
<b>Printer Status to LCD</b>	LCD에 현재 프린터 상태를 표시	char *message
<b>Packet (Printer Status to Network)</b>	모든 유저에게 현재 프린터 상태를 전송	char req_type=-1; int req_id=-1; char res_type=0; int length; char message[length];

3.2.3.4 State Transition Diagram (*View Controller 2.1*)



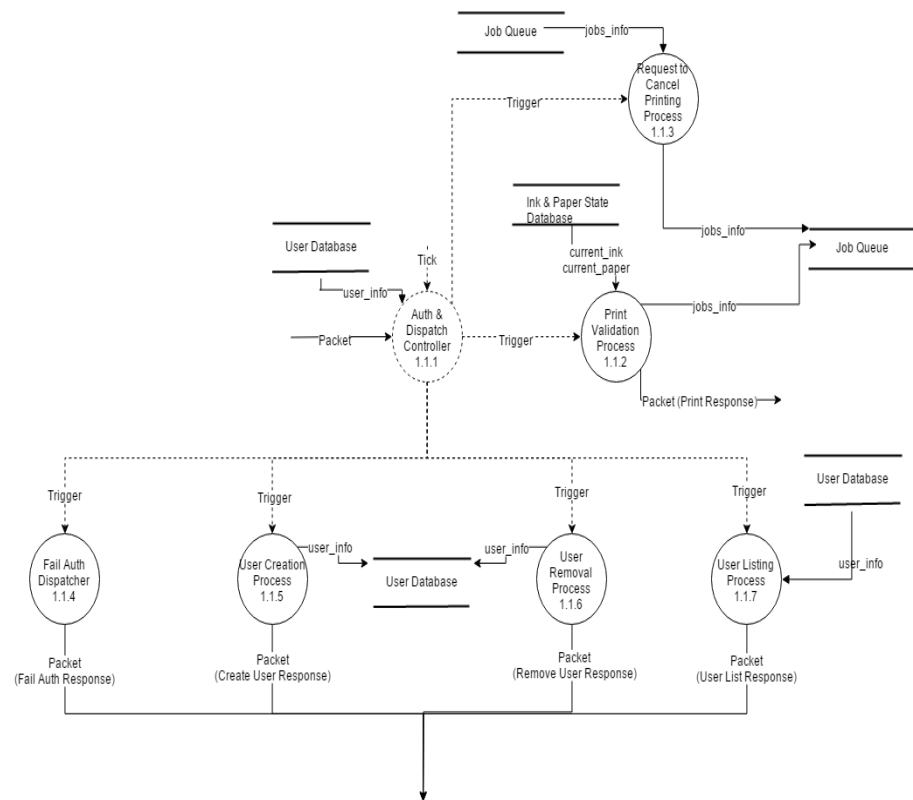
3.2.3.5 State Transition Diagram (*Printing Controller 3.1*)



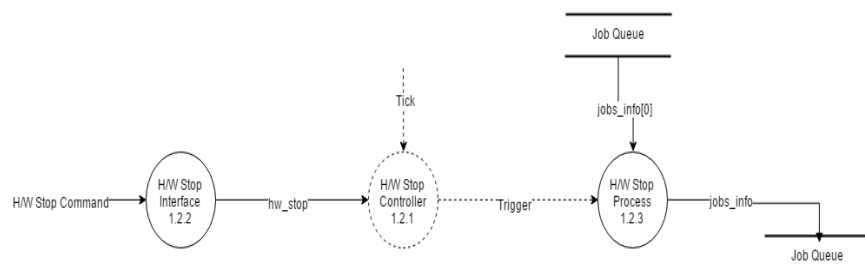
### 3.2.4 DFD Level 3

#### 3.2.4.1 DFD

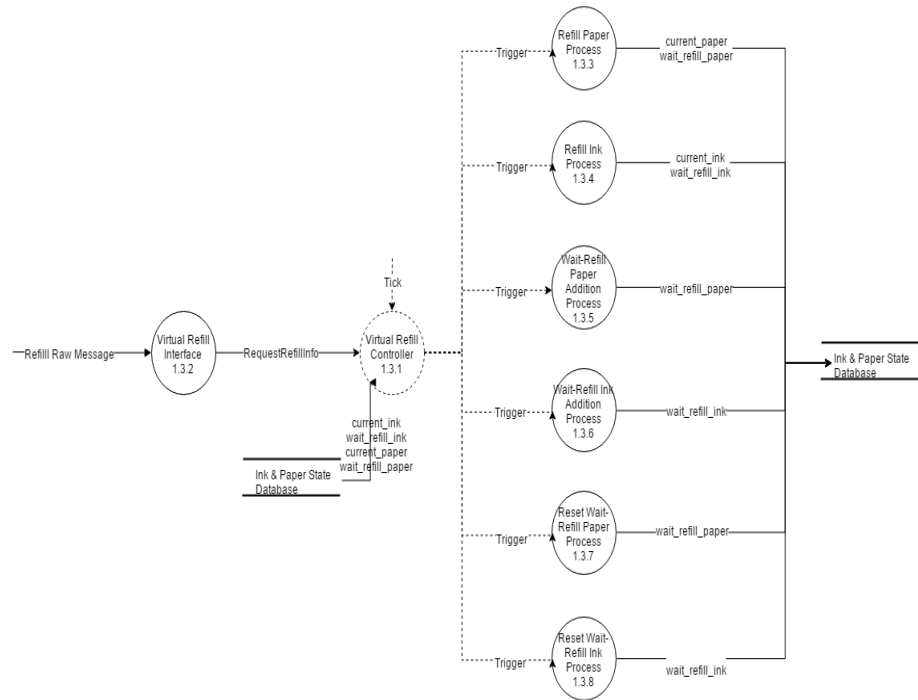
##### 3.2.4.1.1 Request Process



##### 3.2.4.1.2 H/W Stop Process



### 3.2.4.1.3 Virtual Refill Process



### 3.2.4.2 Process Specification

#### 3.2.4.2.1 Process 1.1.1

<b>Reference No</b>	<b>1.1.1</b>
<b>Name</b>	Auth & Dispatch Process
<b>Input</b>	<a href="#">Tick</a> <a href="#">Packet</a> <a href="#">user_info</a>
<b>Output</b>	Trigger
<b>Process Description</b>	사용자, 관리자 인증과 해당 데이터가 어느 프로세스로 가야할지 결정하는 프로세스

#### 3.2.4.2.2 Process 1.1.2

<b>Reference No</b>	<b>1.1.2</b>
<b>Name</b>	Print Validation Process
<b>Input</b>	Trigger <a href="#">current_ink</a> <a href="#">current_paper</a>
<b>Output</b>	<a href="#">jobs_info</a>



	<a href="#">Packet (Print Response)</a>
<b>Process Description</b>	인쇄가 가능한지 체크하는 프로세스로 Job 정보들과 현재 자원 정보를 기반으로 Queue의 사이즈 5 이상이거나 자원이 인쇄에 필요한 만큼 존재하지 않을 경우 Create Job를 수행하지 않는다.

## 3.2.4.2.3 Process 1.1.3

<b>Reference No</b>	<b>1.1.3</b>
<b>Name</b>	Print Cancellation Process
<b>Input</b>	Trigger <a href="#">jobs_info</a>
<b>Output</b>	<a href="#">jobs_info</a>
<b>Process Description</b>	인쇄 취소를 처리하는 프로세스로 대상이 첫번째 Job인 경우 job의 state를 removing로 변경하고 그렇지 않은 경우 job에서 해당 job를 제거한다.

## 3.2.4.2.4 Process 1.1.4

<b>Reference No</b>	<b>1.1.4</b>
<b>Name</b>	Fail Auth Dispatcher
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">Packet (Fail Auth Response)</a>
<b>Process Description</b>	인증 실패시 사용자에게 인증실패를 알림

## 3.2.4.2.5 Process 1.1.5

<b>Reference No</b>	<b>1.1.5</b>
<b>Name</b>	User Creation Process
<b>Input</b>	Trigger <a href="#">user_info</a>
<b>Output</b>	<a href="#">Packet (Create User Response)</a>
<b>Process Description</b>	유저 생성을 처리하는 프로세스

## 3.2.4.2.6 Process 1.1.6

<b>Reference No</b>	<b>1.1.6</b>
<b>Name</b>	User Removal Process
<b>Input</b>	Trigger <a href="#">user_info</a>
<b>Output</b>	<a href="#">Packet (Remove User Response)</a>
<b>Process Description</b>	유저 삭제를 처리하는 프로세스

## 3.2.4.2.7 Process 1.1.7

<b>Reference No</b>	<b>1.1.7</b>
<b>Name</b>	User Listing Process
<b>Input</b>	Trigger <a href="#">user_info</a>
<b>Output</b>	<a href="#">Packet (User List Response)</a>
<b>Process Description</b>	유저 리스트 조회 프로세스

## 3.2.4.2.8 Process 1.2.1

<b>Reference No</b>	<b>1.2.1</b>
<b>Name</b>	H/W Stop Controller
<b>Input</b>	<a href="#">Tick</a> <a href="#">hw_stop</a>
<b>Output</b>	Trigger
<b>Process Description</b>	Stop가 눌렀는지 감시하는 컨트롤러

## 3.2.4.2.9 Process 1.2.2

<b>Reference No</b>	<b>1.2.2</b>
<b>Name</b>	H/W Stop Interface
<b>Input</b>	<a href="#">H/W Stop Command</a>
<b>Output</b>	<a href="#">hw_stop</a>
<b>Process</b>	UDP로부터 Stop 명령을 받는 인터페이스

<b>Description</b>	
--------------------	--

## 3.2.4.2.10 Process 1.2.3

<b>Reference No</b>	<b>1.2.3</b>
<b>Name</b>	H/W Stop Process
<b>Input</b>	Trigger <a href="#">jobs_info[0]</a>
<b>Output</b>	<a href="#">jobs_info</a>
<b>Process Description</b>	Stop 처리를 담당하는 프로세스로 첫번째 Job를 제거한다.

## 3.2.4.2.11 Process 1.3.1

<b>Reference No</b>	<b>1.3.1</b>
<b>Name</b>	Virtual Refill Controller
<b>Input</b>	<a href="#">Tick</a> <a href="#">RequestRefillInfo</a> <a href="#">current_ink</a> <a href="#">wait_refill_ink</a> <a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Output</b>	Trigger
<b>Process Description</b>	자원 충전을 관리하는 컨트롤러

## 3.2.4.2.12 Process 1.3.2

<b>Reference No</b>	<b>1.3.2</b>
<b>Name</b>	Virtual Refill Interface
<b>Input</b>	<a href="#">Refill Raw Message</a>
<b>Output</b>	<a href="#">RequestRefillInfo</a>
<b>Process Description</b>	관리자로부터 자원 충전 요청을 받는 인터페이스

## 3.2.4.2.13 Process 1.3.3

<b>Reference No</b>	<b>1.3.3</b>
<b>Name</b>	Refill Paper Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">current_paper</a> <a href="#">wait_refill_paper</a>
<b>Process Description</b>	용지 충전을 처리하는 프로세스

## 3.2.4.2.14 Process 1.1.4.3

<b>Reference No</b>	<b>1.3.4</b>
<b>Name</b>	Refill Ink Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">current_ink</a> <a href="#">wait_refill_ink</a>
<b>Process Description</b>	잉크 충전을 처리하는 프로세스

## 3.2.4.2.15 Process 1.1.4.4

<b>Reference No</b>	<b>1.3.5</b>
<b>Name</b>	Wait-Refill Paper Addition Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">wait_refill_paper</a>
<b>Process Description</b>	잉크 충전 대기를 추가하는 프로세스

## 3.2.4.2.16 Process 1.1.4.5

<b>Reference No</b>	<b>1.3.6</b>
<b>Name</b>	Wait-Refill Ink Addition Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">wait_refill_ink</a>
<b>Process Description</b>	용지 충전 대기를 추가하는 프로세스

## 3.2.4.2.17 Process 1.1.4.6

<b>Reference No</b>	<b>1.3.7</b>
<b>Name</b>	Reset Wait-Refill Paper Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">wait_refill_paper</a>
<b>Process Description</b>	용지 충전 대기량을 초기화 하는 프로세스

## 3.2.4.2.18 Process 1.1.4.7

<b>Reference No</b>	<b>1.3.8</b>
<b>Name</b>	Reset Wait-Refill Ink Process
<b>Input</b>	Trigger
<b>Output</b>	<a href="#">wait_refill_ink</a>
<b>Process Description</b>	잉크 충전 대기량을 초기화하는 프로세스

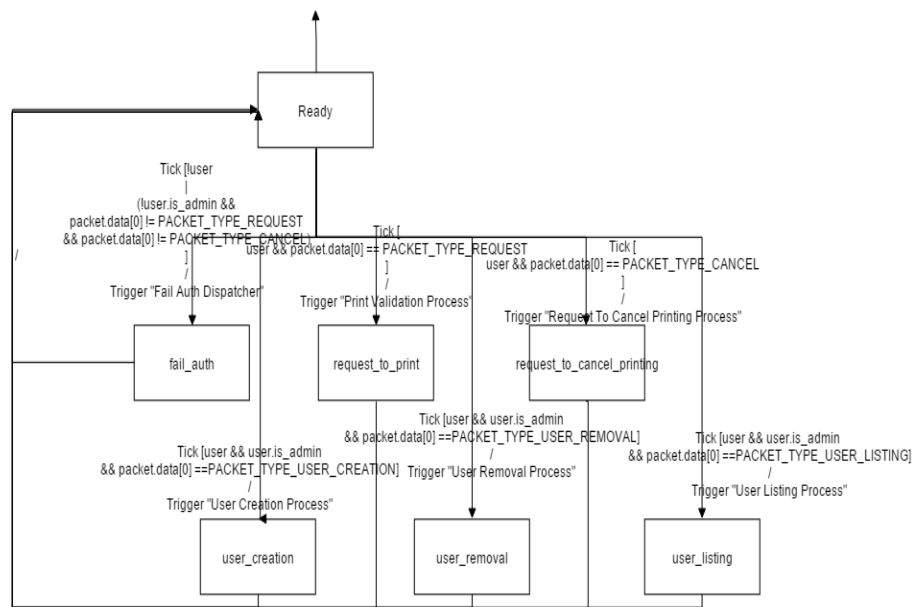
## 3.2.4.3 Data Dictionary

<b>Input / Output Event</b>	<b>Description</b>	<b>Format Structure</b>
<b>user_info</b>	사용자 정보	<pre>struct UserInfo { char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH]; char is_admin; // 0, 1 }  struct UserInfo list[~~];</pre>
<b>Packet (Print Request)</b>	인쇄 요청	<pre>char req_type=PACKET_TYPE_REQUEST; int req_id; char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH]; int file_size; char file_data[file_size];</pre>

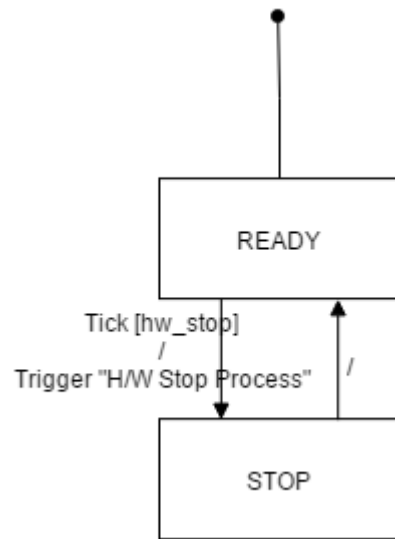
<b>Packet (Print Response)</b>	인쇄 요청 결과	char req_type; int req_id; // 0: 성공 1: 큐x 2: 잉크x 3: 용지x char res_type; // 생성된 Job Id int job_id;
<b>Packet (Cancel Request)</b>	Job 취소	char req_type=PACKET_TYPE_CANCEL; int req_id; char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH]; int job_id;
<b>Packet (Fail Auth Response)</b>	인증 실패	char req_type; int req_id; char res_type=-1;
<b>Packet (Create User Request)</b>	사용자 생성 요청	char req_type = PACKET_TYPE_USER_CREATION; int req_id; char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH]; char t_id[USER_ID_MAX_LENGTH]; char t_pw[USER_PW_MAX_LENGTH];
<b>Packet (Create User Response)</b>	사용자 생성 결과	char req_type; int req_id; // 0: 성공 1: 실패 char res_type;
<b>Packet (Remove User Request)</b>	사용자 삭제 요청	char req_type = PACKET_TYPE_USER_REMOVAL; int req_id; char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH]; char t_id[USER_ID_MAX_LENGTH];
<b>Packet (Remove User Response)</b>	사용자 삭제 결과	char req_type; int req_id; // 0: 성공 1: 실패 char res_type;
<b>Packet (User List)</b>	유저 리스트 조회 요청	char req_type = PACKET_TYPE_USER_LISTING;

<b>Request)</b>	청	int req_id; char user_id[USER_ID_MAX_LENGTH]; char user_pw[USER_PW_MAX_LENGTH];
<b>Packet (User List Response)</b>	유저 리스트 조회 결과	char req_type; int req_id; char res_type; int count; for(count~~) { char user_id[USER_ID_MAX_LENGTH]; char is_admin; }
<b>hw_stop</b>	Stop 버튼 눌렀는지 여부	// 1이면 놀림 (int)1 or 0
<b>RequestRefillInfo</b>	충전 요청	// 0: 잉크 1: 종이 char type; int value; // -1이면 요청 없음

3.2.4.4 State Transition Diagram (Auth & Dispatch Controller 1.1.1)



3.2.4.5 State Transition Diagram (*H/W Stop Controller 1.2.1*)



3.2.4.6 State Transition Diagram (*Virtual Refill Controller 1.3.1*)

