


캡스톤 디자인 I 종합설계 프로젝트

프로젝트 명	SSD Test 자동화 시스템
팀 명	1g's dormitory
문서 제목	계획서

Version	1.1
Date	2018-04-12

이름	권유상
----	-----


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인Ⅰ 수강 학생 중 프로젝트 “SSD Test 자동화 시스템”을 수행하는 팀 “1g's dormitory”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “1g's dormitory”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.


문서 정보 / 수정 내역

수정날짜	대표수정 자	Revision	추가/수정 항 목	내 용
2018-03-09	권유상	1.0	최초 작성	최초 작성
2009-04-10	권유상	1.1	계획서 수정	내용수정 및 추가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

목차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
1.2.1	기존에 개발된 시스템 현황 및 한계점	4
1.2.2	개발할 시스템의 필요성	4
2	개발 목표 및 내용	5
2.1	목표	5
2.2	연구/개발 내용	5
2.2.1	Host Server & SSD Test Server	5
2.2.2	Database	6
2.2.3	Web	6
2.3	개발 결과	6
2.3.1	기능 요구사항	6
2.3.2	Use Case Diagram	7
2.3.3	시스템 설계도	7
2.3.4	비 기능 요구사항	8
2.4	기대효과 및 활용방안	8
3	배경 기술	8
3.1	기술적 요구사항	8
3.2	현실적 제한 요소 및 그 해결 방안	9
3.2.1	하드웨어	9
3.2.2	소프트웨어	9
4	개발 일정 및 자원 관리	9
4.1	개발 일정	9
4.2	일정별 주요 산출물	10

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

1 개요

1.1 프로젝트 개요

최근 저장매체 시장이 HDD에서 SSD로 바뀌고 있다. SSD가 HDD에 비하여 속도가 빠르고, 전력 소비도 적으며, 소음 또한 작기때문에 SSD시장이 더욱 활성화될 것으로 예상 된다. 이에 따라서 여러 회사에서는 SSD 제품을 출시하고 있고, 현재 산학과제를 함께 진행하기로한 회사인 Fadu 또한 SSD 제품을 개발하고 있기 때문에, 이 제품들을 검증하기 위한 시스템이 필수적이다. 현재 나와있는 Test Tool들은 대부분 local machine에서 테스트하기 때문에 대량으로 테스트하기에는 비효율적이다. 이러한 비효율성을 해결하기 위해, Fadu에서는 Web상에서 SSD 테스트를 제어하여 장소와 시간에 제약 없이 테스트를 할 수 있는 시스템을 만드는 것을 산학과제로 다루기로 했다.

1.2 추진 배경 및 필요성


1.2.1 기존에 개발된 시스템 현황 및 한계점

현재는 SSD 테스트를 할 때, 하나의 Local Machine(PC 또는 장비)에서 SSD Benchmark Tool을 실행하기 때문에 비용(인력과 시간)이 크다. 또한 테스트 결과를 볼 수는 있지만 테스트 목적에 맞는 정보 또는 회사에서 필요한 정보들 외에도 많은 정보가 함께 있기 때문에 비효율적이다. 그리고 Log File들을 Database에 저장하기는 하지만 수많은 로그들을 Database의 저장된 값으로만 비교/분석하기는 힘들며, Firmware/Test Script 별 SSD Test 이력관리가 어렵다.

1.2.2 개발할 시스템의 필요성

1. 한 PC 당 하나씩 테스트를 진행하게 되면 수량이 많아 졌을 때 많은 시간을 소비하게 된다. 한 사람이 여러 대의 PC 를 제어하여 한번에 다수의 SSD 를 테스트 한다면 시간과 인력을 줄일 수 있다.
2. SSD Test 결과에 불필요한 정보까지 포함되어 있다. 테스트의 결과들 중 필요한 정보만 뽑아서 보여 준다면 테스트 결과를 더 잘 활용할 수 있게 된다.
3. 테스트 이력을 다양한 기준(시간, 제품, Firmware 등)에 따라 분류하여 저장해 놓으면 테스트 결과의 비교/분석이 쉬워진다.
4. 테스트가 회사 내부에서만 가능하다면 시간 낭비가 크다. 테스트에 많은 시간이 소요되는데 그것을 회사 외부에서도 웹페이지를 통해 할 수 있다면 시간을 절약할 수 있다.

따라서 위와 같이 효율성을 높여, 더 많은 SSD를 빠르고 편리하게 Test해 줄 수 있는 System이 필요하다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

2 개발 목표 및 내용

2.1 목표


기존에 있는 SSD Test System에 대한 구조와 이해를 바탕으로 보다 효율적인 Test System을 개발한다. 이를 위한 주요 목표는 다음과 같다.

1. 원격으로 SSD Test 가 실행되고 결과값을 원격으로 받아볼 수 있는 시스템
2. 동시에 여러 PC 의 SSD Test 를 수행하고 비동기식으로 결과값을 받아볼 수 있는 시스템
3. 원하는 테스트 결과 및 테스트 제어를 Web 으로 할 수 있는 시스템
4. Firmware 버전관리와 Test Script 의 관리가 테스트와 마찬가지로 원격으로 동시에 여러 SSD 에 적용 가능한 시스템
5. 지금까지의 테스트 결과를 관리할 수 있는 시스템

2.2 연구/개발 내용

2.2.1 Host Server & SSD Test Server

- Host Server는 중앙 서버로 각 테스트 장비(PC)와, Database, Web을 연결해 주는 역할이고, SSD Test Server (이하 Test Server) 는 각각의 테스트 장비에 있는 서버로, 테스트를 실행하고 Host Server와의 통신을 담당하는 역할이다.
1. Local PC 내에서의 SSD Test Tool 을 찾아서 이 Tool 에 대한 I/O 를 정리한다.
 - Local PC에서 Test Tool이 올바르게 실행되는지 확인하고 어떤 방식으로 테스트 하는지 확인하기 위함이다.
 2. SSD Test Tool 의 데이터를 네트워크로 전송 가능한 json 형태로 가공한다.
 3. SSD Test PC 와 Host PC 를 네트워크로 연결하고 기본적인 명령어(GET, POST 등)를 테스트 한다.
 4. json 형태로 가공한 데이터를 송/수신이 가능한 서버를 개발한다.
 - Node.js 의 express framework 를 사용해서 서버를 개발한다.
 - Test Server 에서는 테스트 결과값을 json 데이터로 송신하고 post 를 통해 명령어를 수신한다.
 - Host Server 에서는 post 를 통해 테스트 결과값을 수신하고 json 형식의 데이터로 web 으로 전송한다.
 5. 수신한 json 형태의 message 로 원격으로 테스트를 실행할 수 있도록 Test Server 를 개발한다.
 6. 현재까지의 내용을 1:n 상황에서도 작동할 수 있는 Host Server 를 개발한다.
 - 1대의 local 컴퓨터에서 내린 명령이 n대의 컴퓨터에서 작동하도록 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

2.2.2 Database

1. Web에서 받는 테스트 script에 대한 데이터와 Test Server에서 받는 테스트 결과에 대한 데이터를 저장할 스키마를 정의한다. (mongo db 사용)
2. Remote Firmware Update를 위한 repository를 생성한다
3. Test Server로 Script 정보를 전송하고 전송된 정보로 테스트할 수 있도록 Test Server를 수정한다.
 - Script 정보란 script를 구성하는 config 값들을 말한다.
4. Web에서 script에 대한 정보를 받아서 저장할 수 있게 Host Server를 구성한다
5. Firmware를 원격에서 SSD Test PC로 전송하고 Firmware Update가 작동하게 Test Server와 Host Server를 수정한다.


2.2.3 Web

1. 테스트 결과를 볼 수 있는 웹페이지를 만든다. (node.js의 react framework를 사용)
2. Html, node.js(react), css 로 만들어진 Web 과 node.js(express)로 만들어진 Host Server 가 서로 통신할 수 있도록 만든다.
3. 테스트 결과를 Web 에서 보여줄 수 있도록 Host Server 에서 database 의 자료를 Web 으로 전송하는 기능을 추가한다.
4. Database 에 저장된 결과 로그 값들을 받아서 그래프나 테이블 형식으로 visualize 해서 보여줄 수 있도록 구현한다.

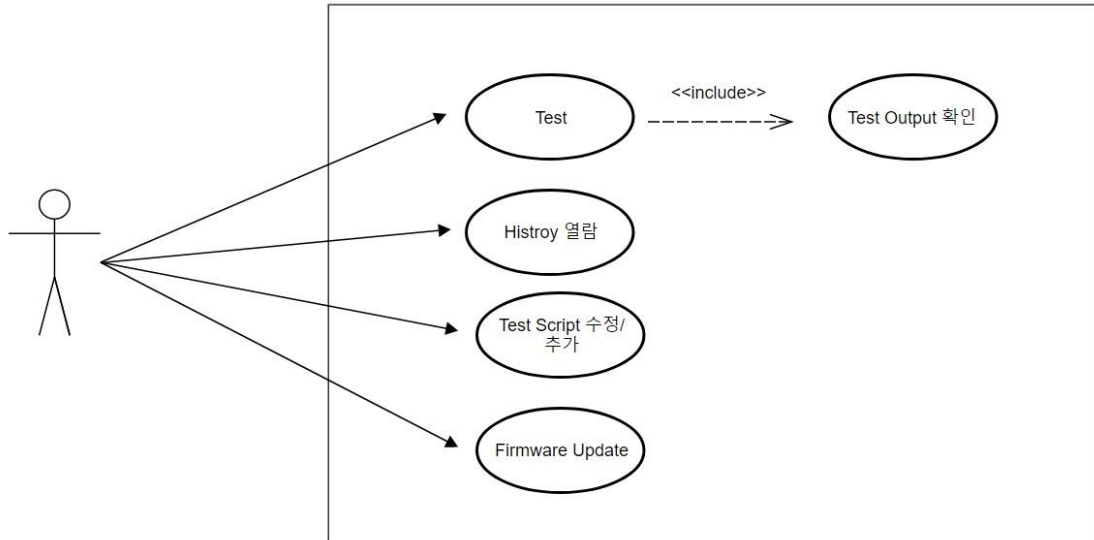
2.3 개발 결과

2.3.1 기능 요구사항

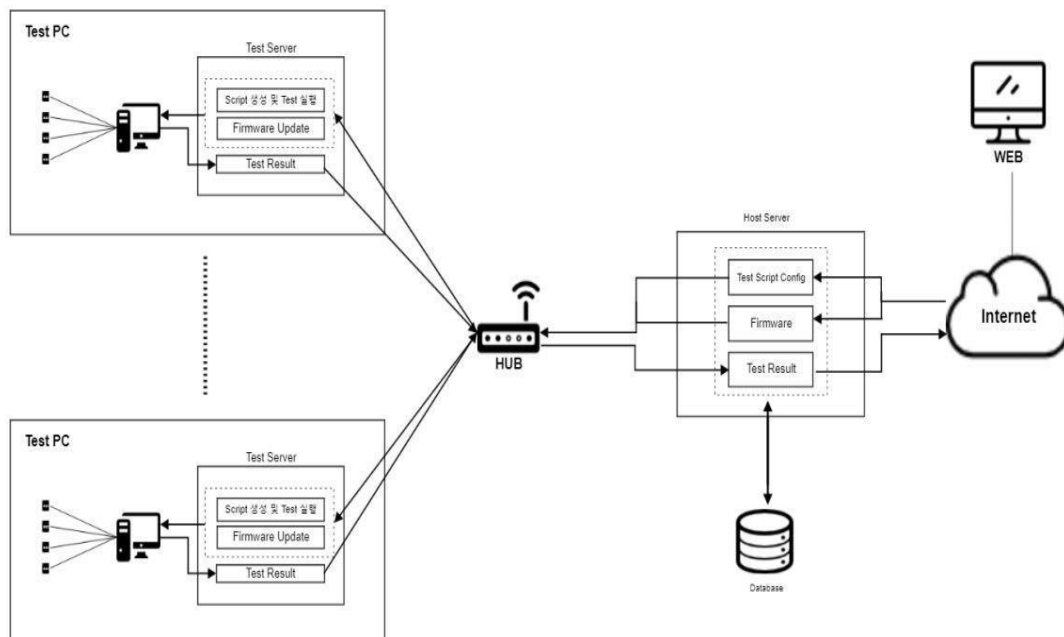
1. 사용자는 시간과 장소에 상관 없이 인터넷이 연결되어 있다면, 웹 페이지를 통해 테스트를 실행하고 결과를 받아볼 수 있고 Test script 를 추가, 삭제, 변경 할 수 있다.
2. 사용자는 다양한 SSD Test tool 을 선택하여 하나의 웹페이지에서 실행할 수 있다.
3. 사용자는 Database 에 저장된 테스트 결과를 비교/분석하기 쉽게 열람할 수 있다.
4. 사용자는 장비에 있는 각각의 SSD 에 대해 원격으로 Firmware update 가 가능하다.


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

2.3.2 Use Case Diagram



2.3.3 시스템 설계도



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

2.3.4 비 기능 요구사항

1. 속도

- SSD Test를 할 때, 사용하는 리소스가 크기 때문에, 테스트 외의 Performance에 영향이 최소화 되어야 한다. 따라서 Performance에 영향이 최소화 되도록 가벼운 Test Server를 구현하여 Overhead를 최소화 하는 방안으로 작업한다.


2.4 기대효과 및 활용방안

- 여러 개의 SSD를 한 PC로 테스트 할 수 있게 된다.
 - 한 대의 local PC로 다른 여러 PC들의 테스트를 컨트롤 할 수 있게 하여, 하나씩 테스트하는 것보다 시간이 감소하게 된다.
- 회사 내부가 아닌 다른 곳에서도 SSD 테스트가 가능하게 된다.
 - 회사 내부에서만 가능하던 테스트가 인터넷이 되는 어느 곳이든 가능해져서 효율성이 높아 진다.
- 테스트뿐만 아니라 Firmware Update도 테스트 방식과 같게 작동하게 된다.
- 과거 테스트 결과들과 필요한 테스트 결과를 볼 수 있게 된다.
 - 여러 테스트 결과 중 회사에서 필요한 정보만 볼 수 있게 되고, 과거 테스트 결과를 저장해 놓고 필요할 때 언제든지 볼 수 있게 된다.

3 배경 기술

3.1 기술적 요구사항

- 프로젝트 개발환경 (개발환경의 OS는 Fadu의 요구사항)
 - 1) SSD Test Server
 - OS : Centos 6.9 server
 - Server : node.js(^v6.9.0 LTS)
 - Framework : express(^v4.0.0 LTS)
 - 2) Host Server
 - OS : Centos 6.7 server
 - Server : node.js(^v6.9.0 LTS)
 - Framework : express(^v4.0.0 LTS)
 - Database : MongoDB
 - Proxy : Nginx
 - 3) Client(Web)
 - Browser : Chrome
 - Language : HTML + JavaScript(node.js)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

Framework : react(^v16.3.1)

- 프로젝트 버전관리
 - 1) GitHub(public)
 - 2) Slack

3.2 현실적 제한 요소 및 그 해결 방안

3.2.1 하드웨어

- SSD 가 장착된 어떤 PC 라도 모두 Test tool 을 사용하는 것은 가능하지만 여러 개의 SSD 를 테스트 하려면 PC 에 걸리는 부하가 크다. 그래서 SSD 를 테스트만을 위한 장비가 있는데 이것은 현재 회사 내부적으로 사용하기 때문에 장비를 이용하여 시험해 보는 것은 어려운 상황이다. 하지만 산학과제 기간 중 새로운 장비가 들어온다고 하여, 최종적으로는 장비에서도 시험을 해볼 수 있을 것으로 예상된다.


3.2.2 소프트웨어

- SSD 테스트 자동화 시스템에서 현존하는 여러 Test tool들을 자동화하는 것을 목표로 하고 있는데 그 Test tool들 중 하나인 nbio라는 것은 회사 내부적으로 만든 것이어서 nbio와 관련된 것들은 외부로 유출되면 안되는 상황이다. 그래서 일단 산학과제 기간에는 fio라는 오픈소스 tool을 이용하여 진행하기로 했다.

4 개발 일정 및 자원 관리

4.1 개발 일정

항목	세부내용	3 월	4 월	5 월	6 월	비고
요구사항분석	요구 분석					
	SRS 작성					
관련분야연구	주요 기술 연구					
	관련 시스템 분석					
설계	시스템 설계					
구현	코딩 및 모듈 테스트					
테스트	시스템 테스트					

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	SSD Test 자동화 시스템	
	팀 명	1g's dormitory	
	Confidential Restricted	Version 1.1	2018-04-12

4.2 일정별 주요 산출물

마일스톤	개요	시작일	종료일
계획서 발표	SSD Test PC 및 Host server PC 정하기 개발 환경 완성 (개발 PC OS 설치) 산출물 : 1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표	~	2018-03-08
설계 완료	시스템 설계 완료 산출물 : 1. 시스템 설계 사양서	2018-03-09	2018-03-23
1 차 중간 보고	개발/연구내용 1~6 구현 완료 산출물 : 1. 프로젝트 1 차 중간 보고서 2. 차분 구현 소스 코드	2018-03-24	2018-04-12
구현 완료	개발/연구내용 6~14 구현 완료 산출물 : 1. 프로젝트 2 차 중간 보고서 2. 차분 구현 소스 코드	2018-04-13	2018-05-18
테스트	시스템 통합 테스트 산출물: 시스템 테스트 결과 보고서	2018-05-18	2018-05-29
최종 보고서	최종 보고		2018-06-01