



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 II

종합설계 프로젝트

프로젝트 명	Unity & Web 프로젝트
팀 명	코드 아카데미
문서 제목	결과보고서

Version	1.3
Date	2017-12-06

팀원	홍용현(팀장)
	정찬형
	최준형
	한용민



CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인II 수강 학생 중 프로젝트 "Unity & Web 프로젝트"를 수행하는 팀 "코드 아카데미"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "코드 아카데미"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	캡스톤2결과보고서-Unity&Web프로젝트.doc
원안작성자	정찬형
수정작업자	정찬형, 최준형, 한용민, 홍용현

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2017-12-01	정찬형	1.0	최초 작성	전반적인 작성
2017-12-05	최준형	1.1	내용 작성	내용 작성
2017-12-05	한용민	1.2	내용 추가	리팩토링 내용 보강
2017-12-06	홍용현	1.3	검토 및 추가	보고서 최종 검토 및 테스트 내용 보강

 국민대학교 컴퓨터공학부 캡스톤 디자인 II	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	Confidential Restricted	Version 1.3	2017-12-06

목 차

1	개요	4
1.1	프로젝트 개요	4
2	개발 내용 및 결과물	4
2.1	목표	4
2.1.1	코드 리팩토링	4
2.1.2	데이터 베이스 구조	4
2.1.3	성능 테스트	5
2.2	연구/개발 내용 및 결과물	6
2.2.1	리팩토링 결과	6
2.2.2	데이터 베이스 구조 개선	7
2.2.3	성능 테스트 결과	8
2.2.4	현실적 제한 요소 및 그 해결 방안	10
2.3	기대효과	10
3	참고 문헌	10
4	부록	10
4.1	프로젝트에 대한 기술 문서	10

 국민대학교 컴퓨터공학부 캡스톤 디자인 II	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	Confidential Restricted	Version 1.3	2017-12-06

1 개요

1.1 프로젝트 개요

본 캡스톤 팀은 개발자를 대상으로 하는 '웹 애플리케이션 개발 능력', '유니티 개발 능력' 코딩 테스트를 개발한다. 기존 알고리즘 테스트와 달리 특정 개발도구 사용 능력까지 확인할 수 있어 회사에서 원하는 인재를 쉽게 채용할 수 있다.

2 개발 내용 및 결과물

2.1 목표

캡스톤1에서 진행한 프로젝트를 보완 및 강화하여 실상용화에 보다 가깝게 만들고, 테스트를 통해 어느 정도의 성능을 낼 수 있는지 확인 후 해결 방안을 모색한다.

2.1.1 코드 리팩토링


리팩토링 목표

1. 중복 코드와 필요 없는 코드를 제거하여 성능을 개선
2. 변수명 규칙을 통일하여 가독성을 높임
3. 절대경로를 상대경로로 바꾸어 호환성 개선
4. 포트번호 오류 수정

2.1.2 데이터 베이스 구조

기존 데이터베이스 구조는 사용자가 하나의 언어로 하나의 문제만 풀 수 있는 구조였다.

사용자가 시험을 볼 때 여러 개의 언어를 선택할 수 있도록 하고 여러 개의 문제를 풀 수 있도록 하여 전체적인 데이터베이스의 구조 개선을 목표로 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 II	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	Confidential Restricted	Version 1.3	2017-12-06

2.1.3 성능 테스트

성능을 테스트하여 실제로 서비스를 하는 데에 있어서 얼마나 가능성이 있는지를 알아보는 것을 목표로 한다.

테스트 전 주요 진행

1. 기존에는 Build 실행 시 Docker 핸들링 문제로 일시정지를 사용하여 서버에 영향이 있었으나 Job마다 개별적으로 Sleep을 하도록 변경하여 부하를 대폭 감소 시켰다
2. 변수명, 함수명의 정리와 불필요한 코드 삭제 (리팩토링)
3. 고정경로를 개선된 경로로 수정 (리팩토링)
4. Ajax, Board Test의 성능을 소폭 향상 (리팩토링)
5. 시연과 대회를 포함한 총 3건의 안정성 테스트를 진행
고정IP를 할당 후 10~30시간 원거리 구동
 - 캡스톤 1차 발표 시연
 - 국민대 공학 페스타 본선 시연
 - 일산 킨텍스 전시회 시연
6. 결과 url 오류, 표기 오류 등 버그 수정


테스트 환경

OS – Ubuntu

1. Laptop1
Intel i7-5600U, 4GB
2. Laptop2
Intel i7-7700 (4core 8thread), 8GB
3. PC
Ryzen 7 1700x (8core 16thread), 16GB
4. Virtual Box
7core 14thread, 12GB

테스트 방법

1. 변수를 다르게 설정한 여러 테스트 환경을 준비
2. 일괄 테스트를 위해 Run Job n개와 Test job 1개를 사용
3. Docker 유지시간을 기존의 약 1분에서 5분으로 늘려 여러 개의 Docker를 동시에 유지시키고 서버가 안정적으로 구동되는지를 확인
4. 하나의 환경마다 5회 이상의 반복 테스트를 진행
5. Web Test 중 부하가 가장 큰 Board Test의 Spring 프로젝트를 사용

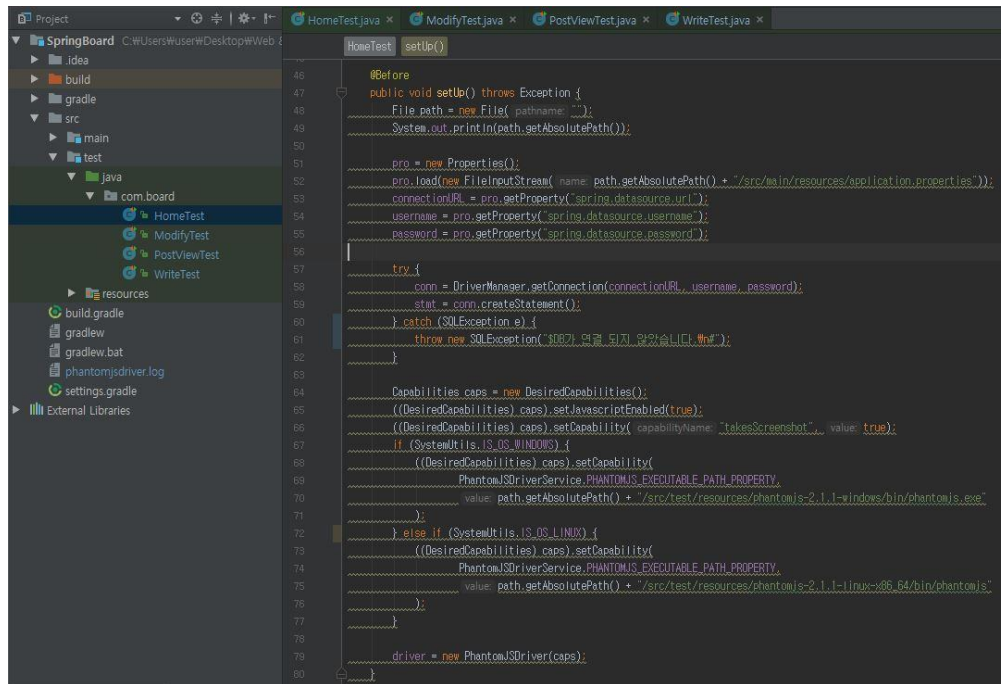
 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 II </div>	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	Confidential Restricted	Version 1.3	2017-12-06

2.2 연구/개발 내용 및 결과물

2.2.1 리팩토링 결과

1. 중복 코드와 필요 없는 코드 제거 결과

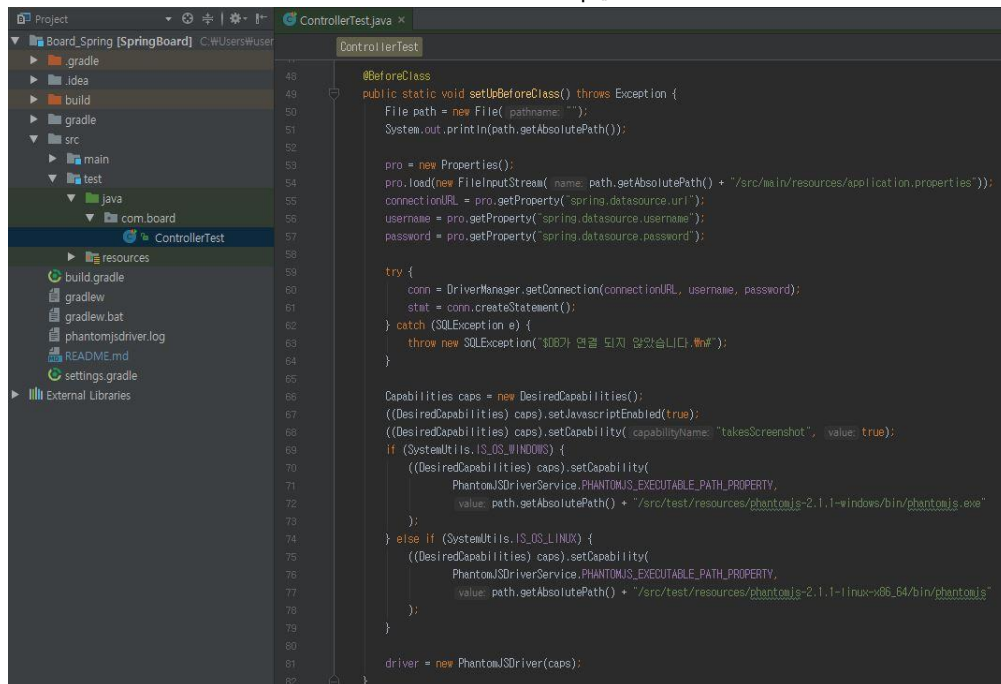
수정 전 코드



```

46  @Before
47  public void setUp() throws Exception {
48      File path = new File( "pathname");
49      System.out.println(path.getAbsolutePath());
50
51      pro = new Properties();
52      pro.load(new FileInputStream( name: path.getAbsolutePath() + "/src/main/resources/application.properties"));
53      connectionURL = pro.getProperty("spring.datasource.url");
54      username = pro.getProperty("spring.datasource.username");
55      password = pro.getProperty("spring.datasource.password");
56
57      try {
58          conn = DriverManager.getConnection(connectionURL, username, password);
59          stat = conn.createStatement();
60      } catch (SQLException e) {
61          throw new SQLException("DB가 연결 되지 않았습니다. #e#");
62      }
63
64      Capabilities caps = new DesiredCapabilities();
65      ((DesiredCapabilities) caps).setJavascriptEnabled(true);
66      ((DesiredCapabilities) caps).setCapability( "takesScreenshot", value: true);
67      if (SystemUtils.IS_OS_WINDOWS) {
68          ((DesiredCapabilities) caps).setCapability(
69              PhantomJSDriverService.PHANTOMJS_EXECUTABLE_PATH_PROPERTY,
70              value: path.getAbsolutePath() + "/src/test/resources/phantomjs-2.1.1-windows/bin/phantomjs.exe"
71          );
72      } else if (SystemUtils.IS_OS_LINUX) {
73          ((DesiredCapabilities) caps).setCapability(
74              PhantomJSDriverService.PHANTOMJS_EXECUTABLE_PATH_PROPERTY,
75              value: path.getAbsolutePath() + "/src/test/resources/phantomjs-2.1.1-linux-x86_64/bin/phantomjs"
76          );
77      }
78
79      driver = new PhantomJSDriver(caps);
80  }
  
```

4개의 테스트 파일과 모든 테스트 파일에 setUp() 함수의 중복을 확인할 수 있다.



```

46  @BeforeClass
47  public static void setUpBeforeClass() throws Exception {
48      File path = new File( "pathname");
49      System.out.println(path.getAbsolutePath());
50
51      pro = new Properties();
52      pro.load(new FileInputStream( name: path.getAbsolutePath() + "/src/main/resources/application.properties"));
53      connectionURL = pro.getProperty("spring.datasource.url");
54      username = pro.getProperty("spring.datasource.username");
55      password = pro.getProperty("spring.datasource.password");
56
57      try {
58          conn = DriverManager.getConnection(connectionURL, username, password);
59          stat = conn.createStatement();
60      } catch (SQLException e) {
61          throw new SQLException("DB가 연결 되지 않았습니다. #e#");
62      }
63
64      Capabilities caps = new DesiredCapabilities();
65      ((DesiredCapabilities) caps).setJavascriptEnabled(true);
66      ((DesiredCapabilities) caps).setCapability( "takesScreenshot", value: true);
67      if (SystemUtils.IS_OS_WINDOWS) {
68          ((DesiredCapabilities) caps).setCapability(
69              PhantomJSDriverService.PHANTOMJS_EXECUTABLE_PATH_PROPERTY,
70              value: path.getAbsolutePath() + "/src/test/resources/phantomjs-2.1.1-windows/bin/phantomjs.exe"
71          );
72      } else if (SystemUtils.IS_OS_LINUX) {
73          ((DesiredCapabilities) caps).setCapability(
74              PhantomJSDriverService.PHANTOMJS_EXECUTABLE_PATH_PROPERTY,
75              value: path.getAbsolutePath() + "/src/test/resources/phantomjs-2.1.1-linux-x86_64/bin/phantomjs"
76          );
77      }
78
79      driver = new PhantomJSDriver(caps);
80  }
  
```

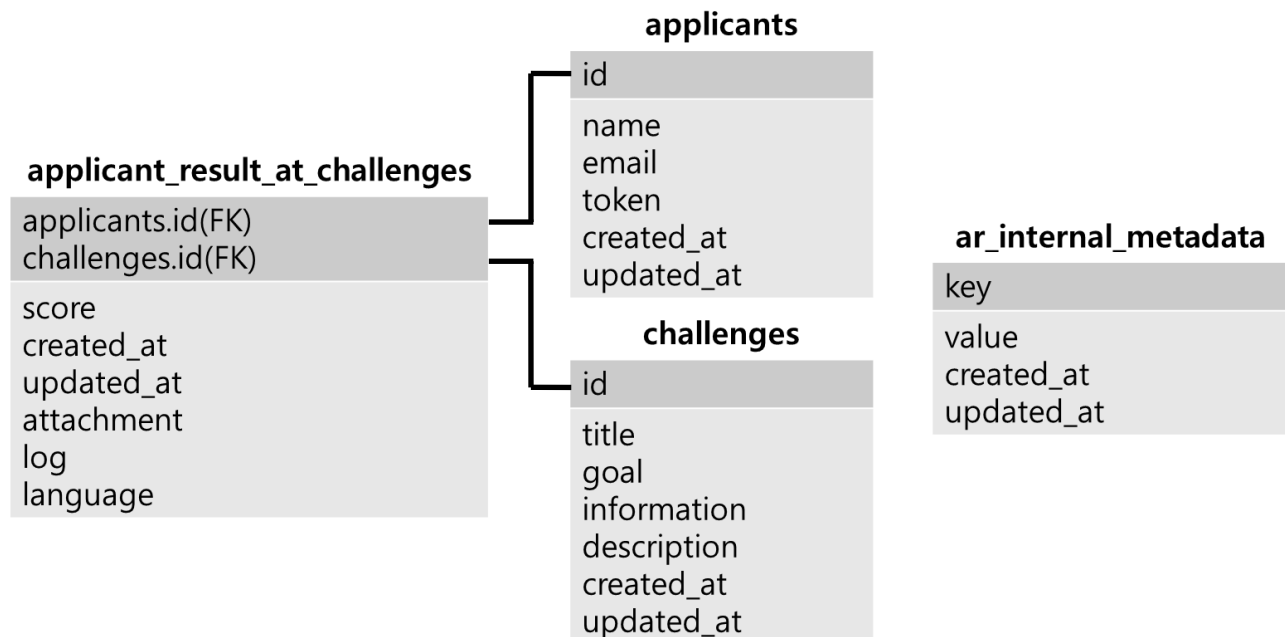
4개의 테스트 파일 통합과 setUp() 함수를 setUpBeforeClass() 함수로 변경하여 성능 개선

2.2.2 데이터 베이스 구조 개선

변경 전 구조

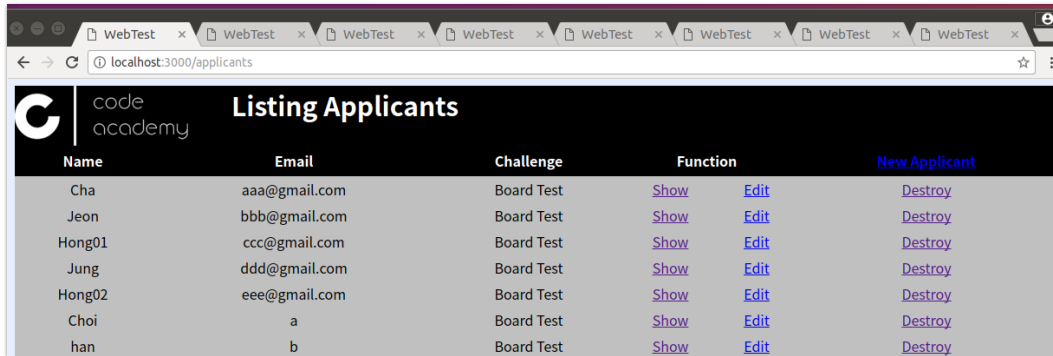


변경 후 구조



결과 테이블을 추가하여 목표로 하던 여러 개의 언어로 여러 개의 문제를 풀 수 있는 구조로 변경하였다.

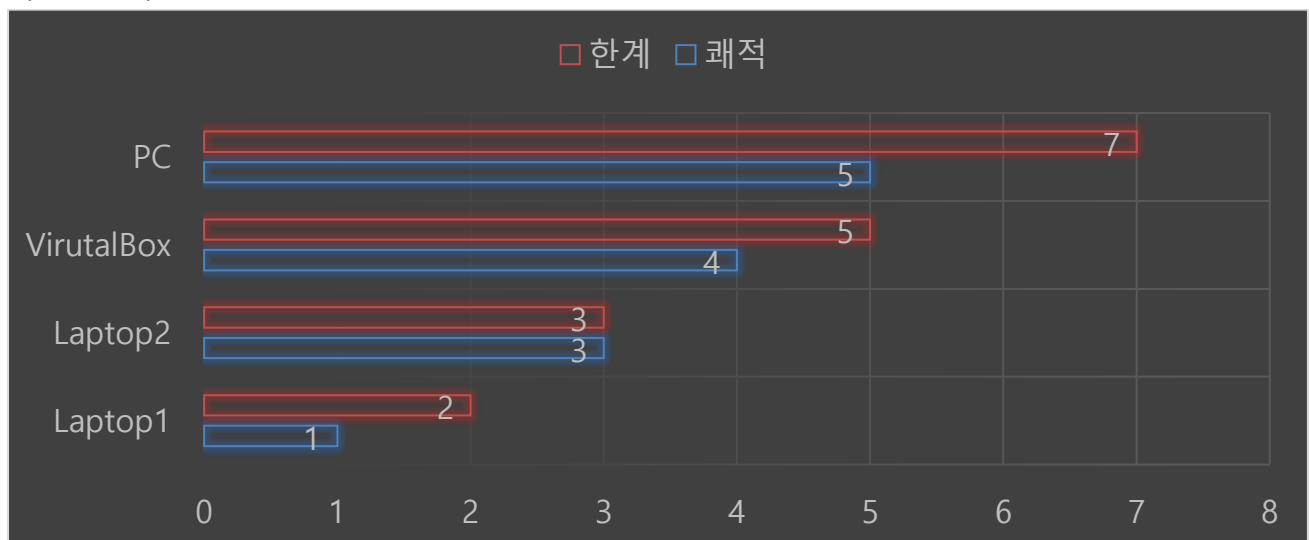
2.2.3 성능 테스트 결과



Name	Email	Challenge	Function	New Applicant
Cha	aaa@gmail.com	Board Test	Show Edit Destroy	
Jeon	bbb@gmail.com	Board Test	Show Edit Destroy	
Hong01	ccc@gmail.com	Board Test	Show Edit Destroy	
Jung	ddd@gmail.com	Board Test	Show Edit Destroy	
Hong02	eee@gmail.com	Board Test	Show Edit Destroy	
Choi	a	Board Test	Show Edit Destroy	
han	b	Board Test	Show Edit Destroy	

Board Test의 응시자를 생성하여 관리자 시점의 Build를 다수 진행하였고 각각의 Docker가 정상 작동(Run) 하는지 확인하였다. 그리고 서버 상태를 파악하기 위해 항상 Test 1개를 마지막으로 실행.

테스트 결과




다수의 Docker가 정상 구동되는 모습

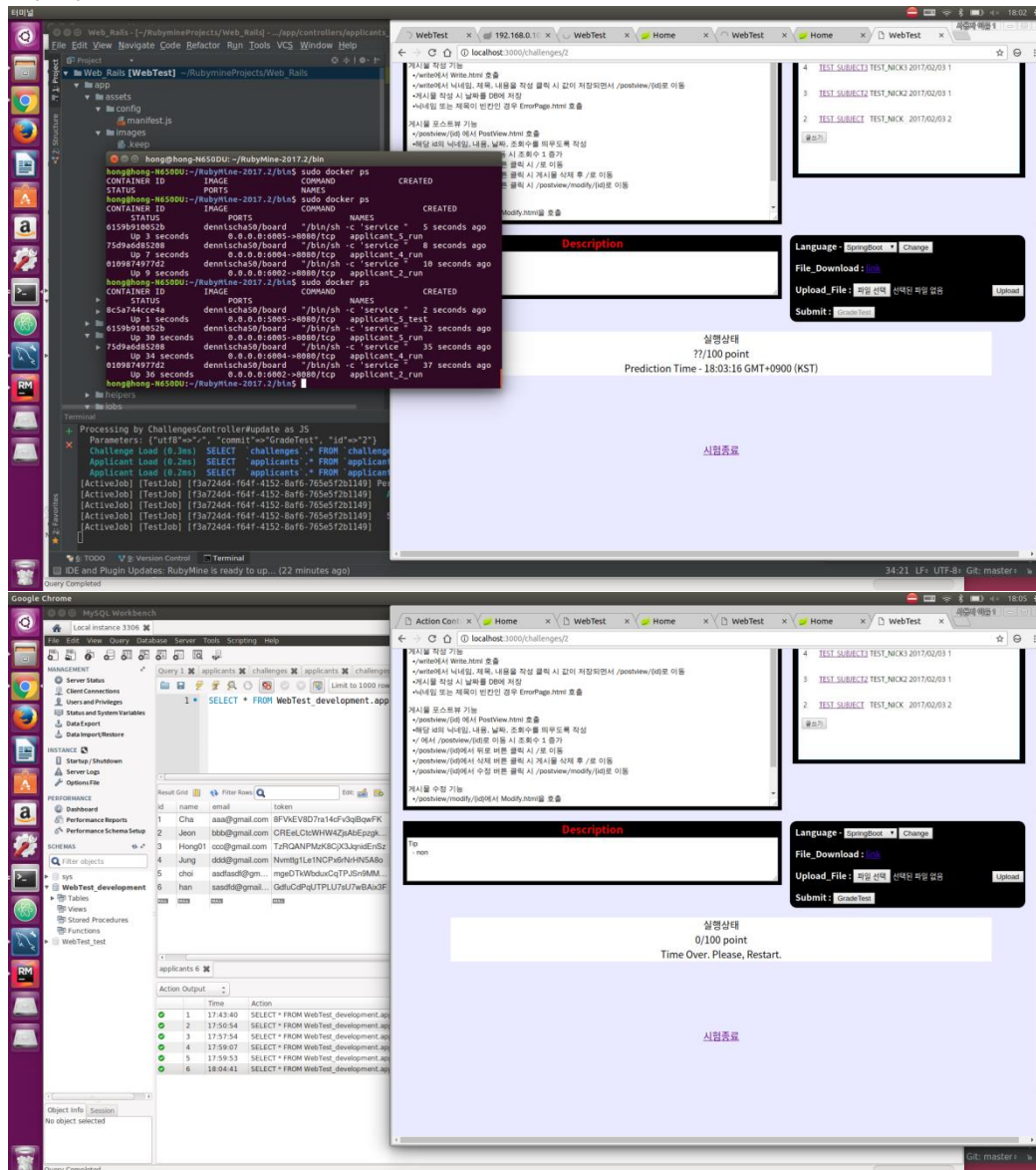
```

hong@hong-VirtualBox:~/다운로드$ sudo docker ps
[sudo] password for hong:
CONTAINER ID        IMAGE               COMMAND             NAMES                  CREATED
023f791206df        dennischa50/board  "/bin/sh -c 'servi..."  applicant_7_test       7 seconds ago
9f524b6be7c3        dennischa50/board  "/bin/sh -c 'servi..."  applicant_4_run        11 seconds ago
7ad9304f4534        dennischa50/board  "/bin/sh -c 'servi..."  applicant_3_run        14 seconds ago
e7c18e249148        dennischa50/board  "/bin/sh -c 'servi..."  applicant_2_run        17 seconds ago
925afd1c1c77        dennischa50/board  "/bin/sh -c 'servi..."  applicant_1_run        20 seconds ago


```


	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	캡스톤 디자인 II	Confidential Restricted	Version 1.3
			2017-12-06

Laptop2 – Docker 4개 실행 시 오류발생 화면



The screenshot displays a laptop screen with a Docker management interface (likely Docker Desktop or a similar tool) and a web application. The Docker interface shows four containers running, with their status and resource usage visible. The web application, titled 'WebTest', shows a 'Description' tab with a list of challenges and a 'Submit' button. The challenges are listed with their IDs and names, and the 'Submit' button is labeled 'Submit'.

 국민대학교 컴퓨터공학부 캡스톤 디자인 II	결과보고서		
	프로젝트 명	Unity & Web 프로젝트	
	팀 명	코드 아카데미	
	Confidential Restricted	Version 1.3	2017-12-06

2.2.4 현실적 제한 요소 및 그 해결 방안

1. 가용 스레드를 초과할 경우 서버에서 에러를 반환하는 현상이 발생
절반 정도 Docker 자체가 종료되어 테스트 진행이 불가능해지는 사례가 발견
2. 코어와 스레드가 많을수록 쾌적한 서버 진행과 한계치의 간극이 발생
이때 PC는 노트북과 달리 오류가 발생하기보다는 서버 자체가 부하를 견디지 못하고 극단적으로 느려지는 현상이 발생
3. 위의 결과를 통하여 서버의 안정성은 시스템의 성능에 비례하며 구동 가능한 Docker의 개수가 증가함을 알 수 있다.
4. 시스템의 성능에 따라 Docker 사용 개수를 제한하고 대기열을 만들 필요가 있어 보인다.

2.3 기대효과

성능테스트를 통하여 실제로 서비스가 가능한 여부에 대해 알아보는 것을 중점적으로 프로젝트를 진행하였다. 결과를 보면 서버 환경에 따라 안정성을 충분히 늘릴 수 있을 것이라 판단하였다. 현재는 웹사이트와 게임만을 테스트 하는데 회사가 요구하는 문제를 만들어 콘텐츠를 늘리고, 안정성을 좀 더 강화한다면 상용화 하는 데에 큰 문제는 없을 것이라고 본다.

3 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	서적	스프링 부트 코딩 공작소	길벗	2016	크레이그 월즈	
2	서적	가장 빨리 만나는 Docker	길벗	2014	이재홍	

4 부록

4.1 프로젝트에 대한 기술 문서

<https://github.com/CapstoneDesign08-2>