

# Assignment Submission System Using Git

ByungHo Lee, HyeonTaek Kong, JooEun Ahn

**Abstract**—Git is considered as one of the most famous version managing tools and getting used to those tool is essential in software development process. If the assignment submission system supports git, it is a good chance for undergraduates to experience and learn git. Prominent Universities overseas have already implemented git for assignment submission system. So, our web application will include the functions that help the professor to manage assignments and the students to easily submit assignments using git. Conclusively, our web application will have a good influence on the students and professors of Hanyang University.

**Index Terms**—git, assignment, submission system, web application

---

## 1 INTRODUCTION

OUR team project theme is to utilize git for effective class assignments submission and supervision for department of Information System, Hanyang Univ. professors, teaching assistants and undergraduate students. What motivated us was that throughout last 3 years in school, especially taking classes including invitation to computer science, operating system, data structure etc, our team members found out several drawbacks in class assignments submission system. Most of the professors tend to prefer hand written assignment rather than online submission via Hanyang portal. The reason why is that hand written assignments are only partially effective for preventing copying other students' assignments. Professors may have already realized; however, hand writing submission system is just a temporary method which does not fit perfectly for professor's purpose, plus it is a big burden for students and teaching assistants. So, our team suggests new online assignment submission system utilizing 'git' for all members of department of Information System. What we mainly expect from our software project is that first of all, since 'git' is one of the most famous version managing tools and getting used to those tools is essential in software development process, it is a good chance for undergraduates to experience and learn 'git'. Prominent Universities overseas have already implemented git for assignment submission system. we would like to add special functions like assignment tracking and auto-testing. Students specify proper branches and those branches will be registered on the main server. Then worker consistently import information from branches to the main server. Based on the imported information, professors are able to monitor assignment submission status and process of all students at a time. this web application will make the assignment submission process much easier.

## 2 REQUIREMENT

Our team members come up with total 4 requirements for this web application. All functions should provide CRUD(create, read, update, delete) functions.

### 2.1 User-friendly program

Our assignment submission application must be easy-to-use. Because our team concentrate on getting more efficiency on managing assignments than original process.

### 2.2 Function to manage accounts

Students and professors will have different registration process.

#### 2.2.1 Student registration

All students need a github account for registration. The system gets authentication token from github and students who registered will assigned to add their personal information such as Hanyang University email address and their student number. Only who verified their Hanyang University email will be able to use this service.

#### 2.2.2 Professor registration

Professors does not need a github account for registration. They only need Hanyang University email address. For registration, the system manager or other professors can approve a user as a professor.

### 2.3 Function to manage courses and assignments for professors

One of the main function of our web application is that professors can easily manage assignment for their students. There are some specific requirements for this function.

#### 2.3.1 Course registration

The professor can register courses. Inside course register page, the professor may fill in the title of the course for students to enable searching and short description.

#### 2.3.2 Assignment registration

The professor can assign assignments to those who register the course. Details are shown below.

Title: The professor can set a title for an assignment. As students will be able to register their github repository for each assignment, the redundancy of assignments could

be handled.

Description: The professor can fill out description of the assignment. Multiple files can be uploaded.

Period setting: The professor can set a deadline for an assignment. When the assignment deadline is over, an email would be sent for professor which includes the statistical data about who did or didnt completed the assignment.

Auto-Testing: The professor can activate auto-testing function to assignment. This function requires test standard input and output results. If this function is activated, the professor need to fill in test inputs and outputs for the assignment or upload a sample file filled with test inputs and outputs. Only the assignment that pass the test can be submitted.

### 2.3.3 Assignment lookup

The professor can ordinarily check who have submitted and passed the test for an assignment

## 2.4 Function to submit assignments for students

The students easily submit their assignment to their professors. There are some specific requirements for this function.

### 2.4.1 Course search

Users can search the course with the title that professor registered. Users also can put the course into my course to easily access for that course

### 2.4.2 Github repository registration

Users can register their remote repository to server which is related to courses. One and different repository for each course.

### 2.4.3 Auto-submission

Students can submit their assignment only by pusing their code to remote repository which is registered. The cron system on our server will repeatedly check assignments in students remote repository every hour. If the assignment exists and passes several tests, the submission will be completed and register to the list of student who complete assignment.

### 2.4.4 Manual submission

There is a defect in auto-submission system if a student doesnt have enough time for waiting as the system checks students repository every hour. If a student wants manual testing, the student has to push code into his github repository and click the test button. Then, the service would fetch from his github repository which is related to course and test for test inputs and make the result.

## 3 DEVELOPMENT ENVIRONMENT

### 3.1 Choice of software development platform

#### 3.1.1 Software Platform Choice and Why

We develop our software on Web, because it provides the best environment for assignment submission for students.

TABLE 1: Team roles

Role	Name
User	ByungHo Lee
Customer	ByungHo Lee
Software Developer	HyeonTaek Kong
Development Manager	JooEun Ahn

## 3.2 Programming Language and Development Environment Choice and Why

### 3.2.1 Angular4

Angular4 is a javascript framework developed by google. Its base is typescript and recently it is widely used in web front development. It is also used in application development because angular 4 is able to provide same user interface.

a)TypeScript Has Great Tools: The biggest selling point of TypeScript is tooling. It provides advanced auto-completion, navigation, and refactoring. Having such tools is almost a requirement for large projects. Without them the fear changing the code puts the code base in a semi-read-only state, and makes large-scale refactorings very risky and costly. TypeScript is not the only typed language that compiles to JavaScript. There are other languages with stronger type systems that in theory can provide absolutely phenomenal tooling. But in practice most of them do not have anything other than a compiler. This is because building rich dev tools has to be an explicit goal from day one, which it has been for the TypeScript team. That is why they built language services that can be used by editors to provide type checking and auto-completion. If you have wondered why there are so many editors with great TypeScript supports, the answer is the language services. The fact that intellisense and basic refactorings are reliable makes a huge impact on the process of writing and especially refactoring code. Although it is hard to measure, I feel that the refactorings that would have taken a few days before now can be done in less than a day. While TypeScript greatly improves the code editing experience, it makes the dev setup more complex, especially comparing to dropping an ES5 script on a page. In addition, you cannot use tools analyzing JavaScript source code (e.g., JSHint), but there are usually adequate replacements.

b)TypeScript is a Superset of Javascript: Since TypeScript is a superset of JavaScript, you dont need to go through a big rewrite to migrate to it. You can do it gradually, one module at a time. Just pick a module, rename the .js files into .ts, then incrementally add type annotations. When you are done with this module, pick the next one. Once the whole code base is typed, you can start tweaking the compiler settings to make it more strict. This process can take some time, but it was not a big problem for Angular, when we were migrating to TypeScript. Doing it gradually

allowed us to keep developing new functionality and fixing bugs during the transition.

c)TypeScript Makes Abstractions Explicit: A good design is all about well-defined interfaces. And it is much easier to express the idea of an interface in a language that supports them. For instance, imagine a book-selling application where a purchase can be made by either a registered user through the UI or by an external system through some sort of an API.

### 3.2.2 Docker

Docker works as a virtual machine and it supplements any weakness of current virtual machine. It takes less time and space because it shares operating system and kernel. Taking images from server, user can use anywhere if they have installed Docker. We will try to synchronize development environment and actual service to eliminate works on my machine problems when collaborating on code.

### 3.2.3 Django

Django is high level python based web framework free, open source so that widely used easy to learn so that development cost is low. Also, Django was designed to help developers take applications from concept to completion as quickly as possible. Django takes security seriously and helps developers avoid many common security mistakes. Some of the busiest sites on the Web leverage Djangos ability to quickly and flexibly scale.

### 3.2.4 MySQL

It is comparably cheap for commercial use. It creates little overhead because it does not matter server performance. Mysql also shows good compatibility with Oracle DB. So any error can be fixed in a short time because it is widely used DBMS and has many references.

### 3.2.5 Cloud Platform

- a)AWS EC2: EC2 works as a server.
- b)AWS Route53: Route53 is used to set up our domain(hyuis.xyz) name server.
- c)AWS RDS: RDS is a kind of database used in server.
- d)AWS Elastic Load Balancer: Elastic Load Balancer is used for auto-scaling.
- e)AWS Elastic Beanstalk: We will utilize elastic beanstalk to integrate and to manage other cloud platforms.

## 3.3 Cost Estimation

AWS costs \$7 a month and RDS(Aurora DB) costs \$29 a month. In total, it approximately costs \$40.

## 3.4 Similar Software in Use

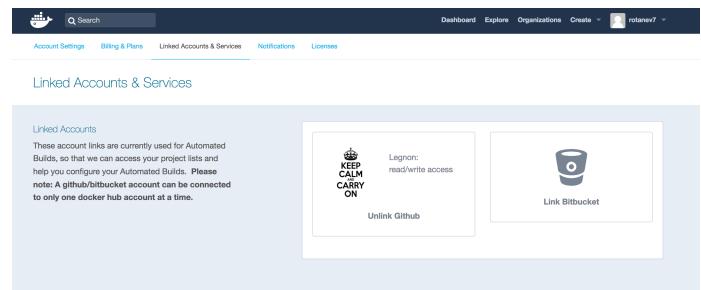
### 3.4.1 Baekjoon Coding

[www.acmicpc.net](http://www.acmicpc.net) is also known as baekjoon coding provides similar function with our web. The purpose of this website is to suggest many code testing problems and users

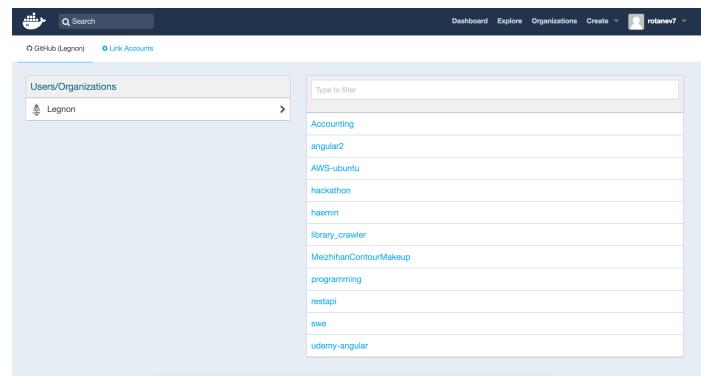
solve them by programming proper codes. Our team got the motivation from this website function code testing. Until now, Hanyang-In portal assignment board did not provide suitable code testing function for computer engineering undergraduates. Students just uploaded their codes and professors or teaching assistants should have runned them one by one. In that sense, our assignment submission system using Github will be a good chance not only for students but also for Professors and TA.

### 3.4.2 Docker Hub

Docker hub is similar to our web in the sense of linking Github repository. Our original plan was to make students type their own repository URL, but it might cause abuse of other persons public repository. So we decided to make students select after searching their repositories.



Example of Docker1

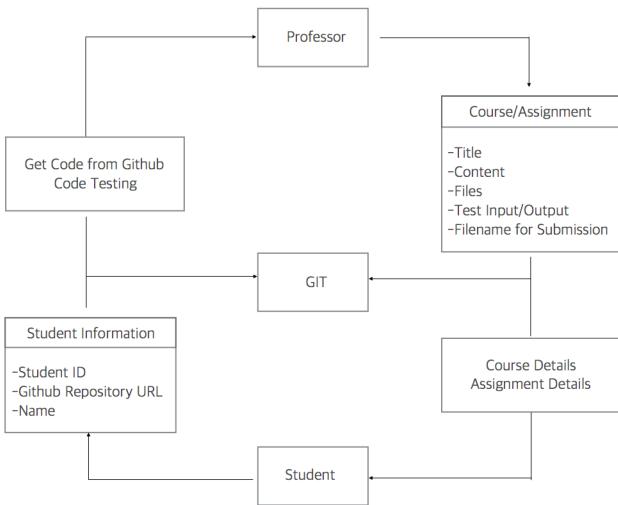


Exmaple of Docker 2

## 4 SPECIFICATIONS

4

### 4.1 Interaction Overview



Interaction overview

#### 4.1.1 From Professors to Web

Professors first need to verify himself with hanyang email account. First time they try to sign up for our web, our web automatically sends verification email to professors so that they can prove themselves. After they log into our web, they can create lectures they teach so that students who take the course can register to conduct assignments. Course information may include title content, semester, title, content and test input and output and deadline. Professors can manage assignments by checking students repository url and see if their source code passed test input and output.

#### 4.1.2 From Web to Students

On our web students can get information for course details and assignment details. First students as well need to verify themselves via hanyang email and register courses they take among lectures professors created.

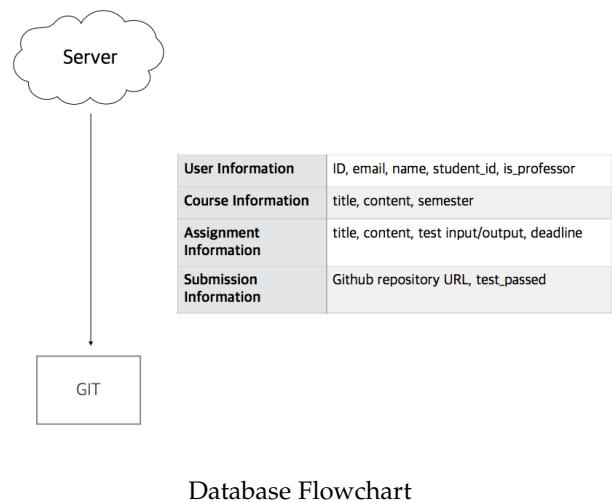
#### 4.1.3 From Students to Github.com

Students create their own repository and repository url then they can submit repository url on our web later on for assignment submission. Students write their source codes that can pass test input and output.

#### 4.1.4 From Github.com to Web

Github provides students source code through each repository url. Since the main purpose of our software program is to give chance for students using github for assignment submission, it is expected that students are acclimatized with utilizing github which is essential for developers.

### 4.2 Database Flowchart



Database Flowchart

From back server to front our web there are several information to bring user information, course information, assignment information and submission information. User information may include each student and professor identification and Hanyang email for verification. After student or professor signs up to our Web, the user data consist of id and password put and built up in our database. By doing this, when a user log in to our web, the set of id and password is on it or not. If the set of id and password are correctly matched the user can access successfully.

#### 4.2.1 User Information

User information definitely includes ID, password and email. Our web is not a open server because only authorized professors and students should have access to each lecture they give or take. Those authorization will be checked via hanyang email and will be monitored by managers.

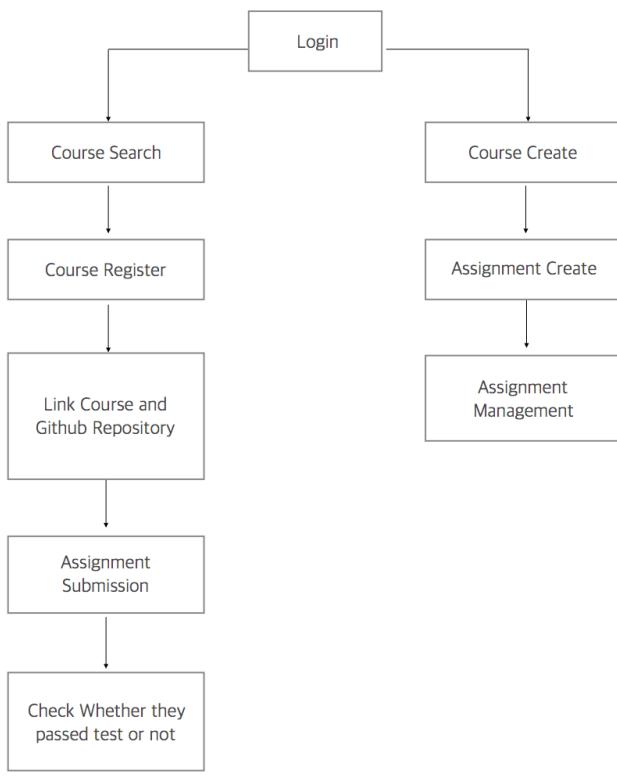
#### 4.2.2 Customer Information

Course information contains title and summary of course. The course should indicate semester, professor and any details like classroom number and time.

#### 4.2.3 Assignment Information

Main target of our web is professors and students in computer engineering major or any other related major. Therefore, most of assignments would contain code testing. Assignment information has files, contents, test input/output and deadlines. Professors should suggest a problem and expected input and output so that students can find the solution and test their codes.

### 4.3 User Interface



User Interface

Both students and professor login verifying their Hanyang email and they will have different interface. Professors can create course which contains course details such as semester, course title and summary of course. Then they can create assignments on each course which contains assignment details such as title, content, deadline, file and test input/output. Lastly, professors can manage assignments by looking at students Github repository URL and they can see if which students passed assignments test input and output with students code. Professors will get email in the end showing results of submitted assignments. After students verify themselves using Hanyang email, they search lectures and register them. Then they can find assignments professors uploaded. After that, when assignments are on the board, students link Github repository URL to the course and they conduct assignment at github.com. After the assignment is submitted students can check whether they passed test or not according to professors test input/output.

#### 4.3.1 Professor Interface

a)Create Course: Professors create lecture course so that students can register course they take. Professors can create multiple courses they teach.

Course Creation	
Course Name	
Content	
Submit	

Overview of how to create course

b)Create Assignment: Professors can create various assignments on each lecture course they created. If they uploaded assignment with test input and output, students need to submit proper codes to get according result .

Assignment Registration	
Course Name	
Course Content	
Test Input / Output Files	
Deadline	
Submit	

Overview of how to create Assignment

c)Assignment Management: Professors can manage assignments through our web. They can take a good look at assignment submission status and test result of students code.

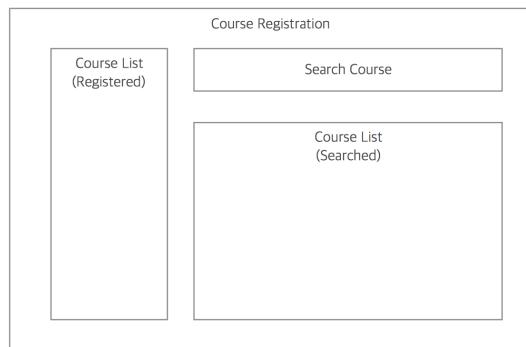
Assignment Management(Professor)	
Course List (Created)	Course Name
Student List and Test Results	

Overview of how to manage Assignment

#### 4.3.2 Student Interface

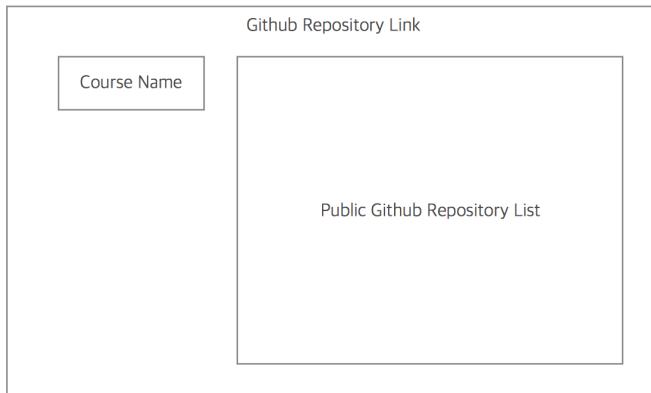
a)Course Search and Register: Students search any course they take among the lecture courses professors created. They

can add multiple lectures into their profile. Only granted students can register courses so that they can browse assignments created. Once they register courses, they can find those anytime they log into our web.



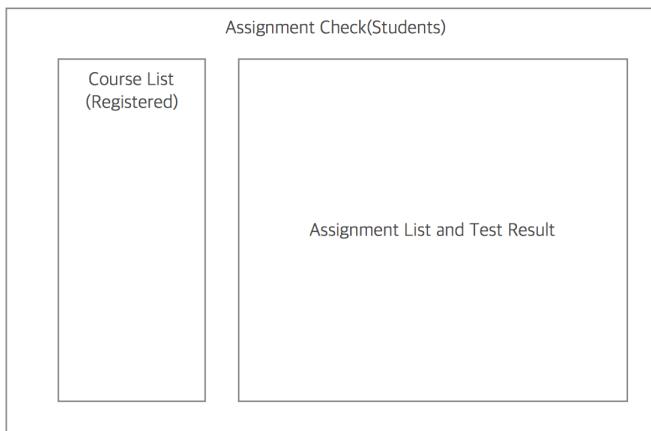
Overview of how to register course

b)Link Course: Professors can manage assignments through our web. They can take a good look at assignment submission status and test result of students code.



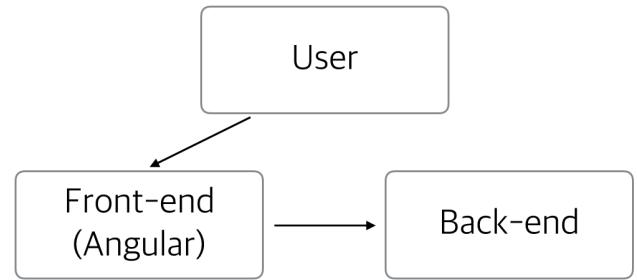
Overview of how to link course

c)Assignment Submission and Check: Students do not submit their assignments on our web. However, they submit assignments on github repository and submit the Url.



Overview of how to submit and check assignment

### 5.1 Interaction Overview



Pictrue 12: Overview of Interaction

Users access to our web through front-end application comprised of Angular. When users send http requests through front-end, those requests are sent to back-end and reply corresponding response. In the response, many kinds of data might be included and since our team implemented REST API using json data type, the response should include data in json data type accordingly. Front-end conduct page rendering based on the response with json data.

### 5.2 Overall Design

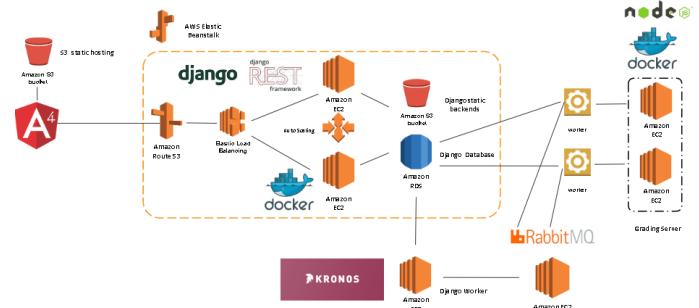
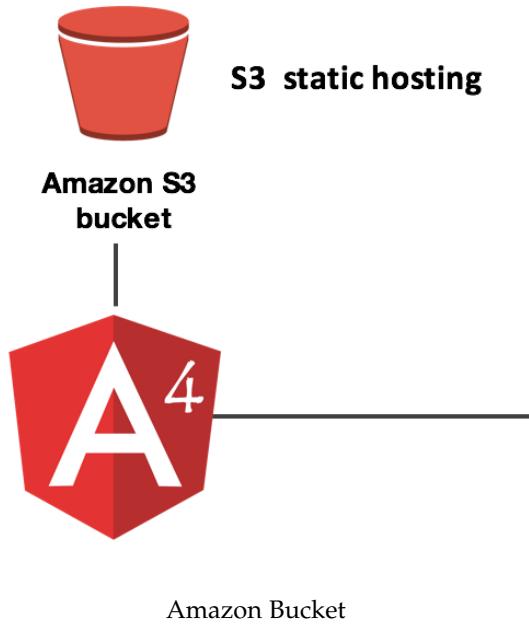


Diagram of Overall Design

I just explained detailed front-end web view flow about how it works. This part is for explaining the detail workflow of whole service.

#### 5.2.1 Frontend

Angular4 and Amazon S3 Bucket Static Hosting



Our service uses angular, we need to serve JavaScript files and a HTML file. For stability and maintenance of the service, we chose static hosting function of AWS S3. We created a bucket for front-end application. We set property as Static website hosting and we also set permissions for users.

#### 5.2.2 Frontend-to-Backend: Route 53

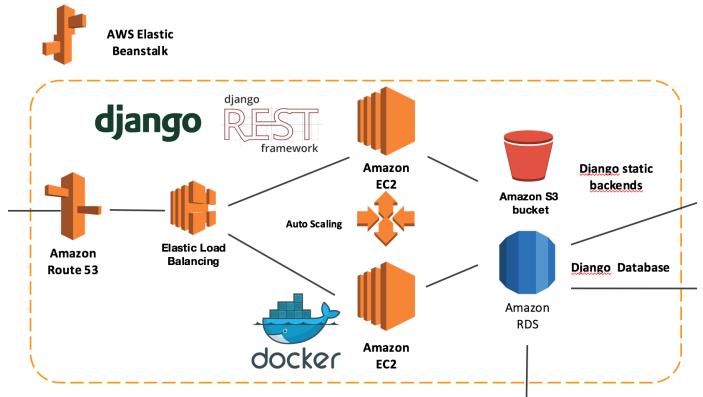


Picture of Route 53

Route53 connects IP address to domain name server. Simply, it makes possible to show what happens at the back on the front. It not only helps to register domain names it also routes server traffic to the resources of our domain. We route our service to AWS EB that contains servers.

#### 5.2.3 Backend

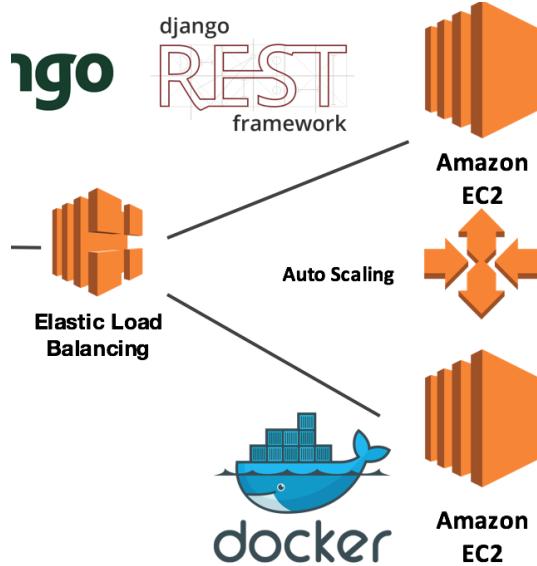
a)Elastic Beanstalk(EB)



Picture of Elastic Beanstalk

AWS Elastic Beanstalk takes control of whole part in dotted line. It has authority to initiate and halt the service. AWS Elastic Beanstalk is the fastest and simplest way to get an application up and running on AWS. We can simply deploy application and the service automatically handles all the details such as resource provisioning, load balancing, auto-scaling, and monitoring. Elastic Beanstalk enables auto-scaling by Elastic Load Balancer to easily support highly variable amounts of traffic.

#### b) Load Balancing and Auto-scaling



Picture of Load Balancing

Load balancing is an essential function when there are variable requests from clients. If a single Amazon EC2 server cannot handle all the requests, Elastic Load Balancer helps you ensure that you have the enough number of Amazon EC2 instances available to handle the load for our server. It creates collections of EC2 instances, called auto-scaling groups. We can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that our group never goes below this size.

c) Amazon EC2 Server using Django and Django



Picture of EC2 Server

Django is a web framework constructed by Python language. As we used angular for our front-end app, there is a need to transfer data between back-end server and front-end application. We chose Representational State Transfer(REST) API to enable it. REST API allow requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations. The server response any information that client requested by json type.

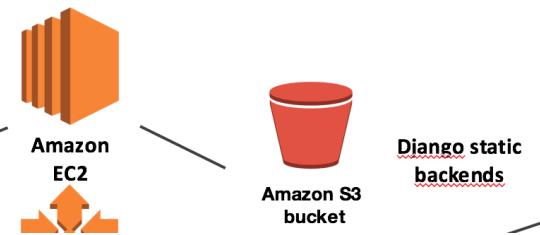
d) Docker



Picture of Docker

As we use auto-scaling for our stability and expandability of our service, there has to be some kind of program that build and deploy our application made by Django as ELB does not handle it automatically. It only increases and decreases the number of servers. We chose Docker. Docker is a software container platform that has a lot of advantages. As it has something in common with virtual machine, though it overcame disadvantages from it, it has a high compatibility which means that it escapes the app dependency matrix.

e) Amazon S3 bucket Django Static Backends

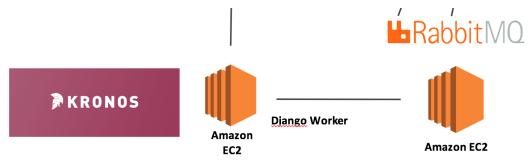


Picture of Amazon S3 bucket and Django Static Backends

We chose AWS S3 to provide static files for the service. Django Static Back-ends is the place to save images up-

loaded by developers. It is linked to Amazon EC2 server and provides static files when they are called.

f)Kronos and Rabbit MQ



Overview of Kronos and Rabbit

EC2 brings submitted codes from student repositories of Github to main server and produce lists of queue to Rabbit MQ. The role of Kronos is to do its fetching job automatically and periodically.

Rabbit MQ basically uses round-robin algorithm. We added prefetch option to ensure the produced queue in list is deleted only after its grading and testing is completed. The grading server will notify that it has finished grading and it will be removed

g)Amazon RDS



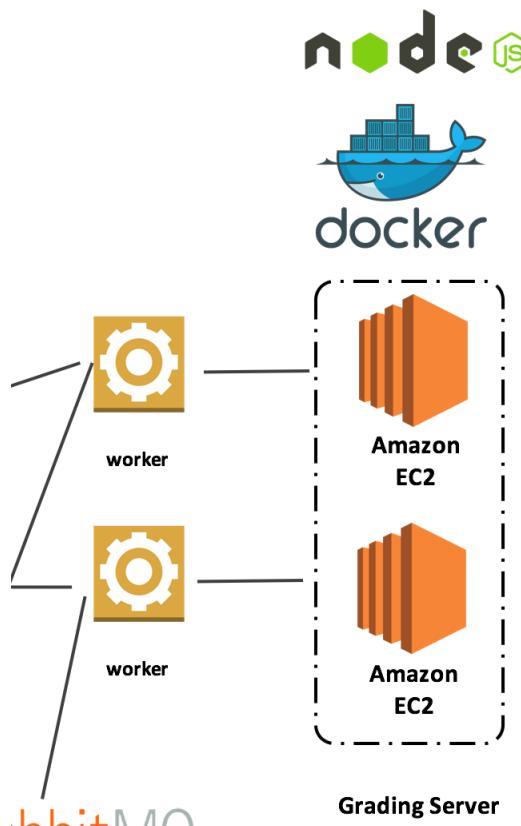
Picture of Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. When you buy a server, you get CPU, memory, storage, and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. So, for example, if you need more CPU, less IOPS, or more storage, you can easily allocate them. Amazon RDS manages backups, software patching, automatic failure detection, and recovery. In order to deliver a managed service experience, Amazon RDS does not provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges. You can have automated backups performed when you need them, or create your own backup snapshot. These backups can be used to restore a database, and the Amazon RDS restore process works reliably and efficiently. You can get high availability with a primary instance and a synchronous secondary instance that you can failover to when problems

occur. In addition to the security in your database package, you can help control who can access your RDS databases by using AWS IAM to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud. In our server, Amazon Relational Database Service saves the mark result of students code and saves the code itself.

We made a route with server that takes assignment code from each students Github repository to store. It interacts with worker connected to grading server and Kronos EC2 which brings code by read-only method from github students repository regularly.

h)Workers, Node and Amazon EC2



Overview of Workers, Node and Amazon EC2

This part is about grading servers. Worker is connected to a grading server and it consumes queue entities from Rabbit MQ. Server takes codes from each worker in order and this code is compiled and run on Docker. From our result from testing demos, a single server can handle up to 10 Dockers at a time. It means that a single grading server can handle approximately 10 workers once at a time. If there are one hundred students and they submitted one hundred different coding assignments, 10 workers bring the code from the RDS server as they consume the list from Rabbit MQ then, deploy the codes to 10 Amazon EC2 servers. What Docker does in grading server is that it deploys CompileBox and compiles codes that workers gave. After grading is completed, it destroys the Docker. CompileBox is a Docker based sandbox to run untrusted code and return the output to your app. It can compile 15 languages. The reason why

we use Docker is to ensure that the system test the code in an isolated environment. This way we do not have to worry about untrusted code possibly damaging our server intentionally or unintentionally. We can use this system to allow student to compile their code right.

### 5.3 Implementation

#### 5.3.1 Front-end Angular App Implementation

a) Modularization As we used Angular for front-end development, it is common that all the working set is divided by component.

```

courses
  course
    assignments
      assignment
      assignment-add
      assignment-edit
        assignment.model.ts
        assignments.component.html
        assignments.component.scss
        assignments.component.spec.ts
        assignments.component.ts
      course.component.html
      course.component.scss
      course.component.spec.ts
      course.component.ts
    course-add
    course-edit
      course.model.ts
      course.service.spec.ts
      course.service.ts
      courses.component.html
      courses.component.scss
      courses.component.ts
      courses.module.ts
      courses.routing.ts
  
```

Overview of Folders

Each folder represents single component. The hierarchy of folder also represents hierarchy of component. Each component contains different and essential functions to render the pages user requested.

b)Guard Guard prevents users from using unintentional work such as registering course before login and sign up. There are two guards in our service, auth and profile guard. Guards work whenever user routes to server. It inspects whether the user login or added profile.

```

path: 'pages',
component: Pages,
canActivate: [AuthGuard, ProfileGuard],
  
```

Picture of Guard

c)Service Service is a collection of working functions to communicate with back-end servers.

```
getCourses(): Observable<Course[]> {
  return this.http.get(`${this.config.BASE_URL}api/v1/course/`,
  {
    headers: new Headers({
      'Content-Type': 'application/json',
      'Authorization': `Token ${localStorage.getItem('auth_token')}`
    })
  })
  .map((res: Response) => <Course[]>res.json());
}
```

Picture of Service

This picture shows how we get the list of courses from DB. It uses RESTful API to get list. The logic of serialization into json type of data is implemented in back-end server. This only sends request and receives the response from server.

When browser(user) sends request, server receives it and get data from DB and serialize the data into json and then it returns response to user. From that data, components render pages.

d)Routing To route our modules interdependent, we use built in module from angular. To make hierarchy of URLs, we had to do like this.

```
export const routes: Routes = [
{
  path: '',
  component: CoursesComponent,
  children: [
    {
      path: 'add',
      component: CourseAddComponent,
    },
    {
      path: ':id',
      component: CourseComponent,
      children: [
        {
          path: 'edit',
          component: CourseEditComponent,
        },
        {
          path: 'assignments',
          component: AssignmentsComponent,
          children: [
            {
              path: 'add',
              component: AssignmentAddComponent,
            },
          ],
        },
      ],
    },
  ],
}
```

Picutre of routing module

At first, it is routing module in courses component. It is reached by `http://base_url/pages/courses/`. There is children array inside constant routes. there is a object which has path of add and component of CourseAddComponent. To access that component, we can simply add `add/` string to URL above. And there is another object which has path of `:id`. It also has component and children array. As it has another children array, it has another hierarchy dependently.

e)Rendering Rendering works as user access the component and the component renders HTML file.

```
<tr>
  <td>Title</td>
  <td>${ assignment?.title }</td>
</tr>
<tr>
  <td>Content</td>
  <td>${ assignment?.content }</td>
</tr>
<tr>
```

Picture of Rendering

This is an example of HTML file. There is special grammar which is called string concatenation. assignment surrounded by braces is variable from component typescript.

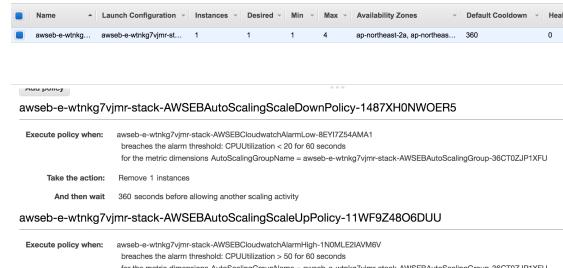
```
ngOnInit() {
  this.route.params.subscribe(
    data => {
      this.id = data['id'];
      this.courseService.getAssignment(this.id).subscribe(
        assignment => {
          this.assignment = assignment;
        },
        err => {
          console.log(err);
        }
      );
      err => {
        console.log(err);
      }
    );
}
```

Picture of Variable assignment

The variable assignment is assigned in the typescript file. It receives data from server through functions implemented in the services such as CourseService.

### 5.3.2 Back-end Server Implementation

a)Auto-scaling AWS provides auto-scaling. We set if percentage of CPU usage is more than 50%, AWS will automatically add 1 EC2 instance.



Picture of Rendering

If CPU utilization is more than 50% for 60seconds, auto scaling group would add 1 instance and CPU utilization is less than 20% for 60seconds, it will remove 1 instance.

b)RDS Rendering works as user access the component and the component renders HTML file.



Picture of RDS

AWS RDS is also used. In this service, we used MySQL5.6. It routes the Kronos EC2 server which fetches code from each students Github repository so that read the data from it.

c)Encryption Those who dont want to share their own assignment, we made encryption algorithm.

```
def encrypt(key, data, size, size_of_chunk=64*1024):
    # initial vector
    iv = os.urandom(16)

    encryptor = AES.new(key, AES.MODE_CBC, iv)
    new_data = struct.pack('<Q', size)
    new_data += iv

    while True:
        chunk = data.read(size_of_chunk)
        if len(chunk) == 0:
            break
        elif len(chunk) % 16 != 0:
            # PAD
            chunk += b' ' * (16 - len(chunk) % 16)
        new_data += encryptor.encrypt(chunk)

    return BytesIO(new_data)
```

Code of Encryption

Its algorithm is CBC, and we used AES of Crypto module for pseudo random function inside. As the user upload file and submit it, the key is stored and outputs the encrypted code. The user has to upload their encrypted code on their Github repository. Then the server will fetch the code from repository and with the key stored in database, it could be decrypted.

d)Kronos To fetch data from Github repository, workers have to periodically and automatically get the job done. In Linux system, cron is a suitable option for this. Django Kronos registers the function in service to cron so that it works automatically.

```
@kronos.register('50 * * * *')
def submit():
    now = datetime.now(timezone.utc)
    assignment_list = SGAssignment.objects.filter(deadline__gte=now)

    for assignment in assignment_list:
        repo_list = assignment.course.sgrepository_set.all()
        for repo in repo_list:
            # already passed
            if SGSubmission.objects.filter(student=repo.student,
                                           assignment=assignment,
                                           is_passed=True).exists():
                continue

            # already working
            if SGSubmission.objects.filter(student=repo.student,
                                           assignment=assignment,
                                           is_working=True).exists():
                continue

            repo_url = [i for i in repo.url.split('/') if i != ""]
            github_repo_name = repo_url.pop()
            github_username = repo_url.pop()
```

Example of Kronos

e)Rabbit MQ To make a queue for grading server, we used Rabbit MQ. This queue produces the entities to be tested and graded. Then worker consumes the entities and gives the work to grading server.

```
connection = pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

channel.queue_declare(queue=_secrets["QUEUE_NAME"], durable=True)
print('worker [x] Waiting for messages. To exit press CTRL+C')

def callback(ch, method, properties, body):
    print("worker [x] Received %r" % body)
    origin_data = json.loads(body)
```

Code of Rabbit MQ

f) Compilebox In grading server, testing should be done. CompileBox is an open source module that compiles and tests the code. It is made by Nodejs and with Docker the code is deployed and tested.

```
#####
# Docker SETUP
#####
apt-get update
apt-get install -y docker.io
ln -sf /usr/bin/docker.io /usr/local/bin/docker
sed -i '$!complete -F _docker docker' /etc/bash_completion.d/docker.io

echo "Docker Setup complete"

#####
# NodeJS setup
#####
apt-get update
apt-get install -y nodejs
apt-get install -y npm
echo "NodeJS setup Complete"

#####
# Start Docker
#####
chmod 777 ..../API/DockerTimeout.sh
chmod 777 ..../API/Payload/script.sh
chmod 777 ..../API/Payload/javaRunner.sh
chmod 777 UpdateDocker.sh

service docker.io restart
./UpdateDocker.sh
```

Code of Compilebox



## 6 USE CASES

13

Sign up to GIT

Already have an GIT account? Sign in!

Username:

Email:

Password:

Repeat:

Example of Student Sign up

First thing users need to do is signing up. In this page, both professors and students should input their ID, password and Hanyang e-mail account.

Sign up to GIT

Already have an GIT account? Sign in!

Username:

Email:

Plz use hanyang email

Password:

Repeat:

Example of Student Sign up2

Referring to figure above, when users enter not hanyang e-mail account but other e-mail accounts, our server rejects users to be signed up.

Sign in to GIT

New to GIT? Sign up!

username:

password:

[Forgot password?](#)

Example of Student Sign In

After users complete signing up, user information including username, email and password is added to database. The last step before logging in is for users to verify Hanyang email account.

Sign in to GIT

New to GIT? Sign up!

username:

password:

User account is disabled.

[Forgot password?](#)

When access is denied

Just like the figure above, it is impossible to log in with unverified email account.

profile-add

is\_prof:

name:

sid:

github:

Example of how to add profile

After logging in, the next page is for adding user profile. In this user profile page, user needs to fill out the form with professor/student identification, name, student number and github identification.

GitHub Register

register github username:

profile-add

is\_prof:

name:

sid:

github:

Example of search in Github

Users are also able to search for their github ID

The screenshot shows a profile-add form with the following fields:

- is\_prof:** A checked checkbox.
- name:** Input field containing "공현택".
- sid:** Input field containing "2013012320".
- github:** Input field containing "Legnon".
- Submit:** A button at the bottom.

Login with Github button checked

Click check button then github ID would be automatically registered on the user profile

The screenshot shows a list of courses with the following columns:

Professor	Course title
공현택(교수)	소프트웨어공학
원영준	asdf
공현택	asdf
공현택	○모글
원영준	asdf
공현택	new

Example of courses

After filling out the profile, user can search courses and register them.

The screenshot shows a search engine interface for courses with the following fields:

- Search term:** Input field containing "소프트".
- Courses:** A table with columns "Professor" and "Course title".

Professor	Course title
공현택(교수)	소프트웨어공학

Searching courses in search engine

On the default screen, users can browse all the courses in the semester, searching engine at the top left help to filter courses by professor name or course title.

The screenshot shows detailed information for a course titled "소프트웨어공학". The table includes:

Category	Detail
Title	소프트웨어공학
Content	소웨공
Professor	공현택(교수)
Year	2017
Semester	0

Picture 22: Information of course example

Detailed information of certain course is shown on the screen if a user selects one of those courses and click it. Course title, contents, professor name and semester info are included in course detail information.

The screenshot shows detailed information for a course titled "소프트웨어공학". The table includes:

Category	Detail
Title	소프트웨어공학
Content	소웨공
Professor	공현택교수
Year	2017
Semester	0
Attachments	tumblr\_nsu2p1NYkT1qagm14o1\_1280.j

Example of register button

In case of student users, they can find register button at the bottom right as shown in the figure above.

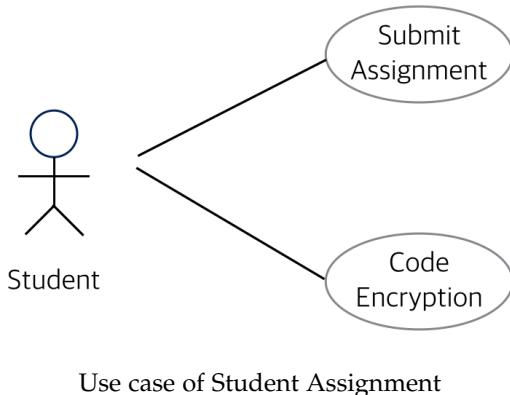
The screenshot shows a registration form with the following fields:

- Semester:** Input field containing "0".
- Attachments:** Input field containing "tumb".
- 신청하기:** A button labeled "신청하기".
- github repository:** Input field containing "submitgit-angular".
- Save:** A green button.
- Cancel:** A red button.

Example of Github register button

If student users click on register button, they can register their own github repository and after saving it, they can register the courses they take. Professors need to verify for the final step of course registration.

#### 6.0.1 Student Assignment use case



Use case of Student Assignment

Course Details of 소프트웨어공학	
Category	Detail
Title	소프트웨어공학
Content	소웨공
Professor	공현택교수
Year	2017
Semester	0
Attachments	tumblr_nsu2p1NYkT1qagm14o1_1280.jpg
Assignment	assignment1
Assignment	테스트

Example of assignment lists

As the procedures for registration is over, new lists for assignments are created.

Assignment Detail of 테스트	
Category	Detail
Title	테스트
Content	asdf
Deadline	2017-05-31T12:00:00Z
File Name	test2
Passed	Not submitted
Allowed Languages	Python Clojure Javascript Scala Bash

**FILE UPLOADER**

파일 선택	선택된 파일 없음	암호화+제출
-------	-----------	--------

Example of information checking assignment

Selecting a certain assignment, student user can check detailed information for the assignment. Detailed information may include title, content, deadline, allowed languages and file name if submitted. File uploader field exists for

students who are reluctant to release their source codes on their github.

#### Assignment Detail of 테스트

Category	Detail
Title	테스트
Content	asdf
Deadline	2017-05-31T12:00:00Z
File Name	test2
Passed	Not submitted
Allowed Languages	Python Clojure Javascript Scala Bash

**FILE UPLOADER**

파일 선택	ERP_중간고사 내용정리.txt	http://submitgit-static.s3.amazonaws.com/upload/ef/2017/05/31/ERP_%E1%84%8C%E1%85%AE%
암호화+제출		

Example of assignment encryption

As the figure above, if the file is uploaded and student user click on encryption button, he or she can download it via url created through encryption. Then student user is required to upload it on github.

#### Assignment Detail of 테스트

Category	Detail
Title	테스트
Content	asdf
Deadline	2017-07-31T12:00:00Z
File Name	test2
Passed	Not submitted
Allowed Languages	Python Clojure Javascript Scala Bash

**FILE UPLOADER**

파일 선택	선택된 파일 없음	암호화
-------	-----------	-----

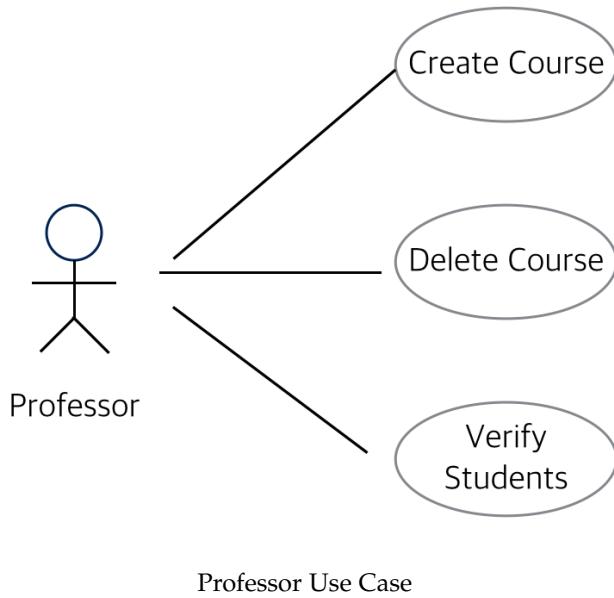
**바로 채점하기**

Example of assignment grading

If assignment submission is completed through github, immediate grading button will be shown on the screen. Technically, our system checks the assignment every one hour but immediate grading button is for students who need prompt feedback and grade for their assignments.

#### 6.0.2 Professor

##### a) Use case diagram Professor Use Case



Professor Use Case

A screenshot of a web form titled 'Create Course'. It includes fields for 'title', 'year', 'semester' (with a dropdown arrow), and 'content'. At the bottom, there is a 'New File' section with a '파일 선택' (File Selection) button and a message '선택된 파일 없음' (No file selected). Buttons for 'Save' and 'Cancel' are at the top right.

Example of course creation form

A screenshot of a web form titled 'profile-add'. It has a checked checkbox labeled 'is\_prof' and a text input field labeled 'name' containing '공현택'. A 'Submit' button is at the bottom.

Example of adding professor's profile

For professor profile, he does not need to write student id or github repository. All he has to do is to check on is\_prof field and write his name.

Courses	
Professor	Course title
공현택교수	소프트웨어공학
원영준교수님	asdf
공현택	asdf
공현택	○ㅁㄹㄴ
원영준교수님	asdf
공현택	new

Example of creating courses

In case of professors, they can create courses they teach in course page.

This screenshot shows a two-step process. The top part is a confirmation dialog asking 'Are you sure you want to delete the course and related assignments?' with '취소' (Cancel) and '확인' (Confirm) buttons. The bottom part is a 'Course Details' page for 'Course Details of 소프트웨어공학'. It lists course information: Title (소프트웨어공학), Content (소웨공), Professor (공현택교수), Year (2017), Semester (0), Attachments (tumblr\_nsu2p1NYkT1qagm14o1\_1280.jpg), Assignment (assignment1), and Assignment (테스트). At the bottom are buttons for '강좌 수정하기' (Edit course), '강좌 삭제하기' (Delete course), and '과제 생성' (Create assignment).

Example of deleting course

Any course will be deleted as professor user clicks on deletion button and confirm his action on confirm dialogue. Extra attention is required during course deletion because deletion of assignments submitted is all accompanied by course deletion.

## Course Details of 소프트웨어공학

Category	Detail
Title	<u>소프트웨어공학</u>
Content	소웨공
Professor	공현택교수
Year	2017
Semester	0
Attachments	tumblr_nsu2p1NYkT1qagm14o1_1280.jpg

localhost:4200 내용:

승인하시겠습니까?



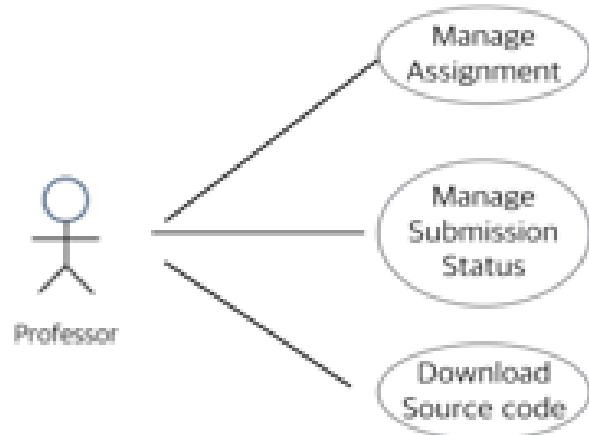
## Examples of acceptance button

When the course is created by professor user and student user register for the course, acceptance button is created like above figure and according screen pops up.

### Example of course creation form

Just like course creation, professors are required to fill out the form for assignment creation, then save button will be activated. All entries are required fields excluding test input and test output.

### *6.0.3 Professor Assignment Use case diagram*

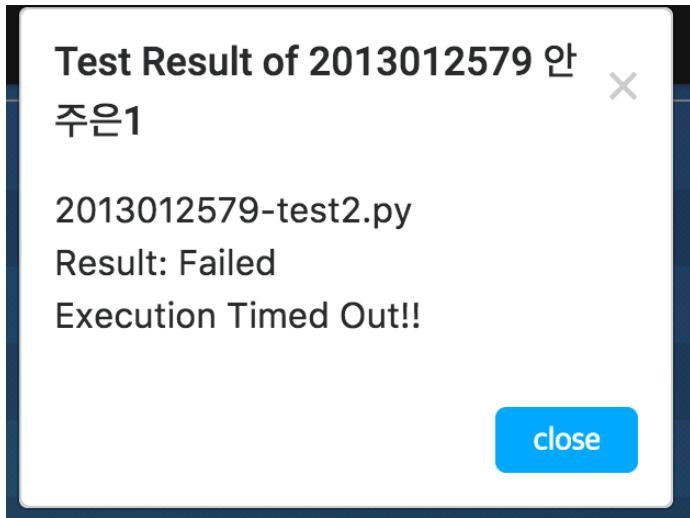


## Use case of Professor Assignment

Assignment Detail of 테스트		
Category	Detail	
Title	테스트	
Content	asdf	
Deadline	2017-07-31T12:00:00Z	
File Name	test2	
Allowed Languages	Python Clojure Javascript Scala Bash	
학생들 결과 조회		
Student	Submission	Check Detail
2013012579 안주은1	Failed	<button>확인하기</button>
2013012579 안주은	Failed	<button>확인하기</button>
2013012320 공현택	Not Submitted	<button>확인하기</button>
과제 수정하기	과제 삭제하기	학생들 과제 다운받기(압축파일)

## Example of checking submission

Student user can check submission status for all students taking the course if the course he or she is taking is registered in his or her profile. For more information, he or she can click check detail button.

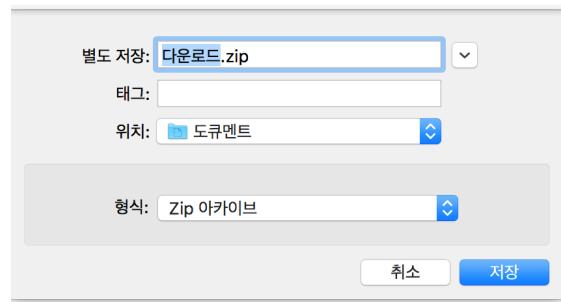


Example of checking submission details

In check detail field, if the student users code fail, it shows the reason of failure with the file name. Code is downloaded by clicking on file name.

Example of assignment amendment

For assignment amendment, professor user should click on assignment correction button then fill out the required entry in the form. Saving button will be activated and correction is completed by hitting the button.



Example of downloading Assignment

When professors click on assignment downloading button, the button will be disabled immediately and codes student submitted will be downloaded in a zip file. Compression will be conducted in EC2 server and the server only sends the route of the file to front-end for download.

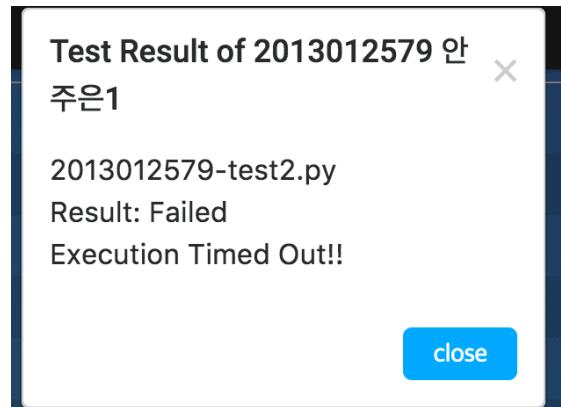
Assignment Detail of 테스트	
Category	Detail
Title	테스트
Content	asdf
Deadline	2017-07-31T12:00:00Z
File Name	test2
Allowed Languages	Python Clojure Javascript Scala Bash

학생들 결과 조회		
Student	Submission	Check Detail
2013012579 안주은1	Failed	확인하기
2013012579 안주은	Failed	확인하기
2013012320 공현택	Not Submitted	확인하기

Example of submission status

Student user can check submission status for all students taking the course if the course he or she is taking is registered in his or her profile. For more information, he or she can click check detail button.



Example of downloading Assignment

In check detail field, if the student users code fail, it shows the reason of failure with the file name. Code is downloaded by clicking on file name.

## 7 SOFTWARE INSTALLATION GUIDE

## 7.1 How to Setup Servers

### 7.1.1 API server

First of all, download the codes via our github, install the requirements using pip. Second, Set secret keys in settings directory. Third, Make Elastic beanstalk environment and Deploy application using eb-cli which is AWS official tools.

### 7.1.2 Django-Worker

First, download the codes via our github on server, install the requirements using pip. Second, Set secret keys in settings directory. Finally, Register tasks with cron written on django enviornment.

### 7.1.3 Rabbit-MQ server and Worker

First, download rabbit-mq server codes via apt-get on server, Second, download supervisord for worker management. Third, download the worker codes via our github. Finally, using supervisord, make worker as a daemon.

### 7.1.4 CompileBox

Download compilebox server codes via our github on server, and deploy node server as daemon using nohup command.

## 7.2 Supported Version of Librariess

- 1)Ubuntu v14.04 LTS
- 2)Docker v1.12.6
- 3)Python v3.5
- 4)Django v1.10.6 & v1.11 LTS
- 5)Django Rest Framework v3.6.2
- 6)Django Kronous v1.0
- 7)TypeScript v2.3
- 8)Angular v4.0
- 9)Node v6.10.3 LTS
- 10)Rabbit-MQ v3.6.8
- 11)MySQL v5.6.27

## 8 DIFFICULTIES

Our team could successfully finish this project, but getting our feet wet to work long-term project, there were several difficulties. First of all, coming up with a new helpful idea or theme for the project was absolutely not easy. We had no idea what to do, how to do and whether we are doing in an efficient way. So, at the beginning of the project, we three members try to gather as much as possible for brainstorming. In the phase of brainstorming, it did not matter any of members had better software development skill. Every single member proposed what he feels uncomfortable in daily life and came up with new assignment submission system using git. Second, in the middle of software development, facing new software platform or skills like making encryption module and messaging queue was not easy. Since our team determined to focus on server distribution, it was critical to highlight auto-scaling. However, for some team members even the concept of auto-scaling was unfamiliar, so it took long time to implement the skill in back-end. For front-end as well, we tried to study angular

4 watching some video clips on Internet, but most of them only focus on theoretical perspective. Advanced skills like connecting it to our server was completely our job. Last but not least, communication problem was the most difficult part. As professor mentioned, the point of software engineering project is to learn communication and cooperation skill. All three members had different software development experience and skills. So what we should have done was to distribute our roles clearly and properly and take much time in discussion and teaching each other. After ideation phase, however, we all depended on remote communication method and it caused some miscommunications and distrust. But the bright side is that we all learned lesson that next time doing project with other members, the most important part is not a difference in development skill but a communication way and skill.

## 8.1 Project Directory

Directory <sup>3</sup>	File names <sup>3</sup>	Module names in use <sup>3</sup>	Etc. <sup>3</sup>
/project/src/app <sup>3</sup>	app.component.ts, app.config.ts, app.module.ts, app.routing.ts, app.service.ts <sup>3</sup>	main component <sup>3</sup>	
/project/src/app/pages <sup>3</sup>	pages.menu.ts, pages.module.ts, pages.routing.ts, pages.component.ts <sup>3</sup>	pages component <sup>3</sup>	used when update left-side menu <sup>3</sup>
/project/src/app/pages/courses <sup>3</sup>	course.model.ts, course.service.ts, courses.component.html, courses.component.scss, courses.component.ts, courses.module.ts, courses.routing.ts <sup>3</sup>	courses component <sup>3</sup>	used when show the list of courses <sup>3</sup>
/project/src/app/pages/courses/course <sup>3</sup>	course.component.html, course.component.ts, course.component.scss <sup>3</sup>	course component <sup>3</sup>	used when show detail of one course <sup>3</sup>
/project/src/app/pages/courses/course-add <sup>3</sup>	course-add.component.html, course-add.component.ts, course-add.component.scss <sup>3</sup>	course-add component <sup>3</sup>	used when adding a course <sup>3</sup>
/project/src/app/pages/courses/course-edit <sup>3</sup>	course-edit.component.html, course-edit.component.ts, course-edit.component.scss <sup>3</sup>	course-edit component <sup>3</sup>	used when editing a course <sup>3</sup>

	course-edit.component.scss <sup>3</sup>		
/project/src/app/pages/courses/course/assignments <sup>3</sup>	assignments.model.ts, assignments.component.html, assignments.component.scss, assignments.component.ts <sup>3</sup>	assignments component <sup>3</sup>	used when read the list of assignments <sup>3</sup>
/project/src/app/pages/courses/course/assignments/assignment <sup>3</sup>	assignment.component.html, assignment.component.scss, assignment.component.ts <sup>3</sup>	assignment component <sup>3</sup>	used when showing the detail information of assignment <sup>3</sup>
/project/src/app/pages/courses/course/assignments/assignment/modal <sup>3</sup>	modal.component.html, modal.component.scss, modal.component.ts <sup>3</sup>	assignment modal component <sup>3</sup>	used when showing the details of result of test <sup>3</sup>
/project/src/app/pages/courses/course/assignments/assignment-add <sup>3</sup>	assignment-add.component.html, assignment-addt.component.scss, assignment-addt.component.ts <sup>3</sup>	assignment add component <sup>3</sup>	used when adding assignment <sup>3</sup>
/project/src/app/pages/courses/course/assignments/assignment-edit <sup>3</sup>	assignment-edit.component.html, assignment-edit.component.scss, assignment-editt.component.ts <sup>3</sup>	assignment edit component <sup>3</sup>	used when editing assignment <sup>3</sup>

/project/src/app/pages/profile <sup>3</sup>	profile.component.h tml, profile.component.s css, profile.component.t s, profile.model.ts, profile.module.ts, profile.routing.ts, profile.service.ts <sup>3</sup>	profile compon ent <sup>3</sup>	compon ent with only service. used to represe nt child compon ents <sup>3</sup>
/project/src/app/pages/profile/profile-add <sup>3</sup>	profile- add.component.ts, profile- add.component.htm l, profile-add component.scss <sup>3</sup>	profile add compon ent <sup>3</sup>	used when adding profile after sign up <sup>3</sup>
/project/src/app/pages/profile/profile- add/modal <sup>3</sup>	modal.component.h tml, modal.component.s css, modal.component.ts <sup>3</sup>	profile modal compon ent <sup>3</sup>	used when check github userna me <sup>3</sup>
/project/src/app/pages/profile/profile-edit <sup>3</sup>	profile- edit.component.ts, profile- edit.component.htm l, profile-edit component.scss <sup>3</sup>	profile edit compon ent <sup>3</sup>	used when editing profile <sup>3</sup>
/project/src/app/pages/register <sup>3</sup>	register.component. html, register.html, register.module.ts, register.routing.ts, register.scss <sup>3</sup>	register compon ent <sup>3</sup>	used when sign up and login <sup>3</sup>
/project/src/app/shared/services <sup>3</sup>	auth.guard.ts, auth.service.ts, profile.guard.ts <sup>3</sup>	shared services <sup>3</sup>	used when prevent from user to use service without login

Directory <sup>2</sup>	File names <sup>2</sup>	Module names in use <sup>2</sup>	Etc. <sup>2</sup>
/project <sup>2</sup>	manage.py <sup>2</sup>	base project <sup>2</sup>	automatically made file by Django to run, manage project <sup>2</sup>
/project/accounts <sup>2</sup>	adapters.py, admin.py, apps.py, models.py, permissions.py, serializers.py, views.py <sup>2</sup>	account app <sup>2</sup>	REST api that manages accounts <sup>2</sup>
/project/portal <sup>2</sup>	admin.py, apps.py, crypto.py, models.py, permissions.py, serializers.py, utils.py, views.py <sup>2</sup>	portal app <sup>2</sup>	app that manages all the transaction occurs throughout the whole service <sup>2</sup>
/project/submitgit <sup>2</sup>	urls.py, wsgi.py <sup>2</sup>	base app <sup>2</sup>	app that connects routes <sup>2</sup>
/project/submitgit/settings <sup>2</sup>	common.py, dev.py, loader.py, prod.py <sup>2</sup>	settings under base app <sup>2</sup>	setting files lies in this folder <sup>2</sup>