ICISTS-KAIST   
Application System Documentation

Version 0.2

작성자 : 박준우

작성일 : 2016.03.27

Contents

[Contents 2](#_Toc446819480)

[1. 개요 3](#_Toc446819481)

[머리말 3](#_Toc446819482)

[프로젝트 시작 경위 3](#_Toc446819483)

[개발 진행 과정 3](#_Toc446819484)

[저작권 및 코드 공개 3](#_Toc446819485)

[코드 수정 시 유의사항 3](#_Toc446819486)

[2. 코드 구성 4](#_Toc446819487)

[프로그램의 핵심 로직, App 4](#_Toc446819488)

[Registration 4](#_Toc446819489)

[Session 4](#_Toc446819490)

[Statistics 4](#_Toc446819491)

[Policy 4](#_Toc446819492)

[MVC Pattern 5](#_Toc446819493)

[View 5](#_Toc446819494)

[3. 관리자 페이지 사용안내 6](#_Toc446819495)

[Export 기능 이용한 참가자 통계 관리 6](#_Toc446819496)

[매년 참가자 모집 시작 전 확인할 사항 6](#_Toc446819497)

[패스워드 분실 7](#_Toc446819498)

[4. 서버 관리 7](#_Toc446819499)

[Server 7](#_Toc446819500)

[Deployment 7](#_Toc446819501)

[DB 관리 7](#_Toc446819502)

[Screen 8](#_Toc446819503)

1. 개요

## 머리말

본 문서는 ICISTS 내부적으로 사용법에 대해 안내하고, 프로젝트의 시작 경위와 개발 진행 과정 등을 기록하기 위해 만들어졌다. 때문에 일반적인 Software Documentation에 없을 내용도 다수 포함한다. 때문에 외부에 유출되지 않도록 당부한다.

## 프로젝트 시작 경위

ICISTS는 연 300여명의 참가자를 관리하는데 어려움을 겪어 왔다. 기존의 DMS(Delegate Management System)는 필요한 기능이 자동화 되어 있지 않아 합격/불합격 통보를 하기 위해서 300여개의 참가자 별 페이지에 들어가 상태 변경을 해주는 등의 불편함이 있었다. 이는 참가자 관리 인력의 불필요한 인력 낭비로 이어졌다. 때문에 3-4년 전부터 DMS를 개편하거나 새로운 프로그램이 필요하다는 의견은 꾸준히 있었다.

ICISTS 공홈을 기존의 XE 기반 서버에서 Squarespace를 이용한 새로운 공홈으로 변경했으므로 굳이 XE의 모듈로 만들어진 기존 DMS 구조를 따라갈 필요가 없었고, 차라리 외주를 맡겨 제대로 오래 쓸 수 있는 프로그램을 개발하기로 하였다.

## 개발 진행 과정

2015년 3월 말부터 SPARCS와 협의를 시작해, 4월 말에 계약서를 작성하였다. 개발은 SPARCS의 이창원, 조성원, 박준우가 진행하였고, 2015년 8월 18일 최종 종료했다.

개발 과정에서 양쪽의 미숙함으로 어려움이 많았다. 우선 정확하고 빠른 개발을 위해서는 발주자(ICISTS)가 구체적으로 원하는 바를 Process Diagram이나 문서로 제공했어야 하나 그러지 못하였다. 필요한 기능을 정리한 목록을 작성했지만 소프트웨어 외주에 대한 경험이 부족해 계약 시작 때 적시하지 못한 기능들을 추후에 덧붙여야만 했다. 발주자도 약속한 기일 내에 필요한 인력을 구하지 못해 Early 기간에 참가자 합격/불합격 통지가 늦어지기도 하였다.

이에 대한 ICISTS의 이의제기에 따라 7월 21일 150만원이었던 계약금액을 130만원으로 낮추기로 하였고, 2015년도 행사 진행을 위한 마무리 작업 후 개발은 종료하였다.

## 저작권 및 코드 공개

이 코드는 모두 오픈소스로 github.com에 공개되어 있다. 주소는: <https://github.com/sparcs-kaist/icists>이다. 깃허브를 통해 소스코드 관리, 이슈트랙킹, 버전 컨트롤을 하고 있다.

## 코드 수정 시 유의사항

모든 코드 수정은 Git을 통해 기록이 남아 있어야 추후에 이에 따른 버그가 생겼을 때 파악이 쉽다. 웬만하면 코드 수정은 클론 된 다른 서버에서 진행하고, 문제가 없을 시에 메인 서버에서 git pull을 하는 방식으로 진행되도록 한다. ㅅ

# 2. 코드 구성

본 프로그램은 Python 기반 web framework인 Django 1.8을 이용해 만들어졌다. 설치가 필요한 플러그인 등은 requirements.txt에 명시되어 있다. ~~icists라는 이름의 프로젝트로 시작되었고, 프로젝트 이름 바꾸는 건 골치 아픈 일이니 넘어가도록 한다.~~ 여러가지 디렉토리 이름이 오히려 코드를 헷갈리게 하는 경우가 있을 수 있다.

manage.py가 들어 있는 directory를 “PROJECT\_DIR”라고 하는데 icists라는 이름의 directory가 러시아 인형처럼 반복되기 때문에 헷갈리기 쉽다. 파일 레퍼런스는 이 PROJECT\_DIR를 기준으로 하게 된다. PROJECT\_DIR의 위치는 settings.py에 명시되어 있고, 흔히는 manage.py가 위치한 곳으로 보면 된다.

PROJECT\_DIR/icists/settings.py 파일에 주요 설정 사항이 결정되어 있다.

## 프로그램의 핵심 로직, App

**App은 본 프로그램의 핵심적인 로직이 모두 들어가 있는 폴더이다. PROJECT\_DIR/icists/app directory 내에 registration, session, statistics, policy 총 세개의 app이 구성되어 있다.** 각 app의 directory 안에 여러 파일들이 있는데, 실제 코드는 views.py, urls.py, models.py, admin.py, forms.py 5개의 파일에 작성 되었다. 참가자 관리를 위한 핵심적인 코드는 대부분 registration app에 있다.

### Registration

크게 ‘application’ 부분과 ‘participation’ 부분으로 나눌 수 있다. Application이 이전 DMS 버전에서 얘기하던 1차신청서이고, participation이 2차 신청서이다.

### Session

Session은 로그인/회원가입 관리를 위한 app이다. 이를 위한 DB 테이블 등이 위치한다.

### Statistics

Statistics는 사용되지 않는 app이다. 건드릴 일도 없고, 수정할 일도 없을 것이다. 우리가 필요한 통계는 admin 페이지 통해 추출 가능하다. 개발 과정에서 폐기된 부분이다.

### Policy

Policy에 해당하는 페이지는 없다. 다만 참가신청 기간 설정, 참가비 설정 등을 위해서 별도의 DB Table을 구성하기 위해 만들었다.

## MVC Pattern

MVC는 Model-View-Controller의 약자이다. 널리 알려진 프로그램 아키텍쳐 패턴이고, Django를 비롯한 웹 프레임워크에서 쓰이는 방식이기도 하다. 요지는 사용자의 request를 받아 데이터를 조작하고, response로 변환할 때 까지의 과정을 세가지 컴포넌트로 구분한다는 것이다.

Django의 경우 각각 다음 파일에 해당된다:  
Model : models.py에 정의된 데이터 구조  
 + settings.py에 명시된 DB 시스템  
View : HTML Templates, CSS, Javascript etc…  
Controller : views.py 에 포함된 Python 함수

## View

상황에 따라 HTML 파일을 수정해야 하는 경우가 있을 것이다. 현재 사용되는 HTML파일은 모두 PROJECT\_DIR/icists/templates 폴더에 있으며 app별로 나뉘어져 있다. Registration app을 위한 HTML파일은 다음과 같다.

welcome.html : 약관 동의 시 application 작성 페이지로 이동  
app\_form.html : application 작성 페이지  
draft.html : application 작성 후, 제출 전 페이지  
app\_closed.html : 지원기간이 아닐 때 접속하면 나오는 페이지  
status.html : 제출 후 나타나는 페이지. Application status와 payment status 표시  
cancel.html : application 제출 이후에 제출 취소 버튼 클릭 시 이 페이지로 이동  
financial.html : financial aid 제출 위한 페이지  
participation.html : participation form (2차 지원서) 작성 페이지

나열된 HTML 파일은 base.html🡪registration/base.html🡪registration/[\*\*\*].html 순으로 extend 한다. 즉, 나열된 모든 파일은 base.html을 포함하고 있다.

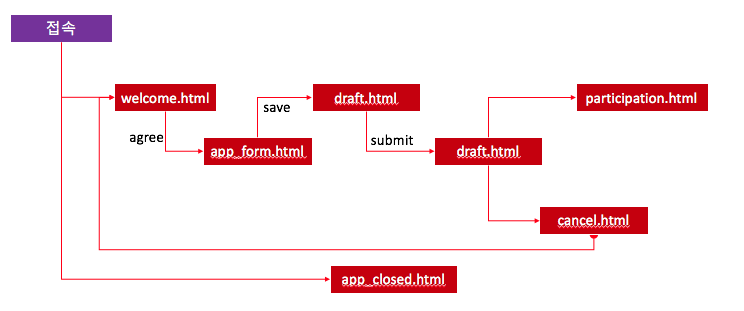


Figure 1. Process Diagram

# 3. 관리자 페이지 사용안내

[url:http://application.icists.org]/admin에서 관리자 페이지에 접근할 수 있다. 이 페이지를 통해 참가자 관리에 필요한 대부분의 작업을 수행한다. 참가자 관리의 거의 대부분 항목은 결국 DB 관리다.

참가자 별 정보가 저장되어 있는 테이블은 Users(사용자들; Django에서 기본 제공되는 테이블이기 때문에 한국어로 번역되어 표시된다), Applications, Participants 세 가지이다. Application Form 작성을 통해 참가신청을 할 때 Application object가 생성되고, 합격통보 후 Participation Form을 작성할 때 Participation object가 생성되어 one-to-one mapping 된다.



Figure 2. Django Admin Main Page – App 별로 구분하여 존재하는 DB 테이블의 목록을 보여준다.

## Export 기능 이용한 참가자 통계 관리

Admin 페이지에 들어가면 각 테이블을 리스트 뷰로 볼 수 있는데, 할 수 있는 operation이 제한되어 있다. 때문에 해당 테이블을 excel 파일로 export 하는 기능이 구현되어 있다.

이 기능을 사용했을 때 출력되는 컬럼의 목록은 apps/[\*\*\*]/admin.py에 있다. “[TableName]Resource” 라는 이름의 클래스가 가지는 attribute 중 fields 리스트를 수정하여 컬럼 목록을 바꿀 수 있다. 또한 출력시 포맷을 선택할 수 있는데 .xls를 선택하면 문제없이 사용할 수 있다.



Figure 3. export

## 매년 참가자 모집 시작 전 확인할 사항

매년 참가자 모집 전에 확인할 사항이 몇가지 있다. Admin 페이지에서 다음 object를 수정해야 한다:

1. Policy 앱의 Configuration: Configuration은 현재의 참가신청 status를 나타낸다. 새로운 object를 생성하지 말고, 하나만 존재하도록 한다. Early/Regular/Late 참가 신청을 열고 닫을 수 있다.
2. Price: Price는 매년 새로운 object를 생성해서 사용하도록 한다.
3. Registration 앱의 Essay Topics 와 Project Topics

Admin 페이지에서 수정하는 내용 이외에도 코드 수정으로 고쳐야 할 부분들이 있다:

1. Paypal 계정 및 입금 계좌명 : PROJECT\_DIR/icists/templates/registration/participation.html 에서 수정 가능하다.
2. 페이지 제목 등에 나타나는 연도 표기 (2015->2016) : base.html 확인
3. Financial Aid Form 질문

## 패스워드 분실

본 프로그램은 사용자가 알아서 패스워드 분실에 대처하는 로직이 없다. 따라서 관리자가 직접 변경해주어야 한다. python manage.py shell 에 접속하면 Python code를 입력할 수 있다. 여기서 임의의 패스워드로 변경한 다음 참가자에게 직접 새로운 패스워드로 변경하라고 하면 된다. 다음 코드는 이메일이 [icists@icists.org](mailto:icists@icists.org)인 계정의 패스워드를 passw0rd로 변경하는 방법이다.

$ python manage.py shell  
>>> from django.contrib.auth.models import User  
>>> u = User.objects.get(email=‘icists@icists.org’)  
>>> u.set\_password(‘passw0rd’)  
>>> exit()

# 4. 서버 관리

## Server

현재 application.icists.org 서버는 아마존 웹 서비스 (AWS)에서 돌아가고 있다. 매달 소액의 사용료를 사용하고 있으며, Ubuntu instance로 생성되었다. icists.org 도메인은 예전부터 cafe24에서 관리하고 있다.

## Deployment

이상적이지는 않지만, 현재는 Apache나 Nginx 같은 웹 엔진이 아니라 Django의 자체 개발용 서버 (python manage.py runserver 0.0.0.0:80)를 통해 서버가 운영되고 있다. 명령 실행을 screen command 위에서 했기 때문에 서버가 꺼지지 않는 한 계속 돌아간다. 개발 이후 Production 단계에서는 Nginx나 Apache 등을 사용해 돌리는 게 정석이나 그렇게 하려면 추가적인 웹서버 setup이 필요해 이대로 계속 돌리고 있다. (<http://www.djangobook.com/en/2.0/chapter12.html>)

## DB 관리

개발 과정에서는 sqlite3 를 DB엔진으로 썼으나 지금은 mysql로 돌아가고 있다. DB 파일은 리눅스 커맨드 창에서 Mysql 프로그램을 실행해 접근할 수 있다. Mysql 계정/비번은 카페에 따로 기록해 두었다. 필요한 자료 접근 및 수정은 admin 페이지 통해서 다 할 수 있으니 Mysql 통한 접근이 필요한 일은 거의 없을 것이다.

다만 혹시나 모를 경우를 대비해 Mysql DB를 일주일 단위로 backup 하도록 서버 내에서 cron script가 실행되고 있다. DB 백업 파일은 /home/icists/backup에 있고, cron 설정은 /etc/crontab 파일에 있다. 현재는 주 단위로 backup이 되고 있지만, 참가신청을 받기 시작하면 일 단위로 backup을 하는 것이 좋을 것이다. 설정은 *sudo vi /etc/crontab* 으로 바꾸면 된다.

## Screen

Screen은 리눅스 서버에서 하나의 ‘가상 스크린’을 만들어 주는 기능이다. 우리처럼 서버를 항시 돌려야 하는데, 백그라운드에서 돌리면 화면에 출력되는 내용을 볼 수 없기 때문에 Screen을 사용한다. Screen을 사용하면 로그인 후, Django 서버를 열고, 그 가상 스크린을 별도로 저장해 둔 후 로그아웃하면 셸 화면이 사라져도 서버 프로세스는 종료되지 않는다.

구글하면 사용법은 다 나온다.

<http://seungigi.blogspot.kr/2012/03/screen.html> (한국어)

<http://www.tecmint.com/screen-command-examples-to-manage-linux-terminals/> (영어)