

# Framework for rapid in-house development of web applications for higher education institutions in Poland

Janina Mincer-Daszkiewicz<sup>1</sup>

<sup>1</sup>Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw, Banacha 2, 02-097 Warszawa, [jmd@mimuw.edu.pl](mailto:jmd@mimuw.edu.pl)

## Keywords

Software framework, web application, Django, Python, RAD, USOS, USOS API.

## 1. ABSTRACT

**USOS** [2] is a *student management information system* developed in-house for higher education institutions (HEIs) in Poland, from the MUCI consortium. It consists of a desktop application developed in Oracle technology for the university administration, and a suite of web based applications, for handling admission, course registration, grades, student diploma theses, social aid, course surveys and many other services, dedicated for students and university teachers. They were developed in the last ten years in such technologies as PHP, Smarty, JavaScript and Ajax. Recently the system designers decided to rewrite some of them from scratch. They chose **Django** as the framework for the new software. While designing the **software architecture** they recognized functionalities which might be shared among various applications and decided to gather them in a separate framework on which some of the future web applications for USOS might be based. In such a way **Django USOS framework** was born. This framework may be used not only by the team of USOS developers but also by programmers developing local applications for the universities from the consortium, either in-house or with the help of independent suppliers.

## 2. INTRODUCTION

USOS is used by about 40 HEIs in Poland. The general procedures are the same in all institutions, since they are regulated by law, the *Act on Higher Education* and accompanying regulations. However, specific procedures may vary, being ruled by local terms of study. USOS tries to adapt to local requirements, but some differences might be so radical that they may require separate solutions. It means that architecture of USOS applications and used technology should allow for rapid development of local software solutions and adjustments.

### 2.1. USOS API

USOS API [1] is a set of REST-full methods giving access to the university data by the standard API. USOS API thus plays a role of an **application server** which implements business logic and makes it available to other modules of the software system. Some USOS web applications use methods of USOS API. New modules and subsystems are based on functionality of USOS API, old modules will be rewritten in the new architecture in the future. USOS API is publicly available and may be used by any software vendor (see <http://apps.usos.edu.pl/developers/api/>). However, only trustful ones will obtain a special administration key giving unrestricted access to the data. Other applications will have to ask a user to log in and grant access to the given range of data.

USOS API can be used to build any type of application, web based, mobile, or desktop, in almost any programming language. However, the range of possibilities is restricted by the API – methods of data access designed by API developers. This may be too restrictive for many applications which need unlimited access to the data. On the other hand, many of them will be web-based and will not need flexibility of USOS API. It means that USOS API is a useful tool for independent developers, but does not meet all needs.

## 2.2. Django USOS framework

Django has been recently chosen by USOS designers as the framework for the new versions of web applications from the USOS suite. While designing the **software architecture** they recognized pieces of software which might be shared among various applications and decided to gather them in a framework on which future web applications with access to USOS database might be based. **Django USOS framework** has been designed to support **rapid application development (RAD)**. Local programmers from HEIs may use it to develop web applications for their specific needs.

## 3. DESIGN AND IMPLEMENTATION OF DJANGO USOS

Django USOS is written in Python and based on Django framework. It has direct access to USOS central Oracle database. It contains libraries, components and plug-ins delivering many services to the client applications. The list of delivered services contains:

- a. Template which defines layout of pages and their constant elements (such as title bar, left menu), the general appearance of the application and some elements of the graphic interface, such as info boxes, message boxes, hints, icons.
- b. Mechanism supporting different language versions of the application.
- c. Mechanism supporting error handling, in particular common error messages.
- d. Data models which transform Oracle tables into Django objects (for most popular tables).
- e. Authentication based on CAS.
- f. Mechanism for building software package and installing software at the server.
- g. Mechanism for defining user privileges.
- h. Some useful extensions to Django template language.
- i. Libraries for connecting to USOS API, building database queries, supporting caching, and system configuration.
- j. Template of an administration module with interface for administrators delivering functions such as acquisition of user session and temporarily shutting down service.
- k. Simple search mechanism and mailing service.

The software package comes with an application template, which is ready to use when the new project is created. Such application is a basis for building other specific functionalities.

Django USOS framework has already been used to build two applications: APD (Archive of Diplomas for uploading and reviewing diploma theses) and USOSmail (mailing service for groups of users defined in USOS, like students of a particular study program or students attending a particular course). In January 2013 authorities of the Faculty of Mathematics, Informatics, and Mechanics asked for the new application which would support staff evaluation. EVA (EVALuation) server was installed in a day, giving the basic functionality delivered by Django USOS. The main functionality required from the dean was ready in a month. In a month and a half the service was delivered to the end users. This has been a real test of Django USOS as the framework supporting the RAD approach.

## 4. CONCLUSIONS

We deliver not only ready to use applications, but also tools, which can be used for rapid in-house development of software for HEIs using USOS. Django USOS framework delivers functionalities needed in many web applications based on USOS Oracle database. APD, USOSmail and EVA are examples of its usability. Other examples coming from independent developers are expected soon.

This paper is based on the Master Thesis of Łukasz Karniewski, supervised by Janina Mincer-Daszkiewicz. All programming work was done by Łukasz.

## 5. REFERENCES

- [1] Mincer-Daszkiewicz J. (2012). USOS API – how to open universities to Web 2.0 community by data sharing. EUNIS 2012, Vila Real. <http://www.eunis.pt/index.php/programme/full-programme>.
- [2] USOS website (2013). *University Study Oriented System*. <http://usos.edu.pl>.

## 6. AUTHORS' BIOGRAPHIES



Janina Mincer-Daszkiewicz graduated the University of Warsaw, Poland, majoring in computer science, and obtained a Ph.D. degree in math from the same university. She is an associate professor in Computer Science at the Faculty of Mathematics, Informatics and Mechanics at the University of Warsaw. Her main fields of research include operating systems, distributed systems, performance evaluation and software engineering. Since 1999, she leads a project for the development of a student management information system USOS, which is used in about 40 Polish Higher Education Institutions from the MUCI consortium. In 2008, she started the Mobility Project with RS3G. Janina takes active part in many nation-wide projects in Poland.